

Oracle® Fusion Middleware Application Adapters

Application Adapter for J.D. Edwards OneWorld
ユーザーズ・ガイド for 12c (12.2.1.2.0)

E85900-01

2016 年 12 月

J.D. Edwards OneWorld システムとの統合方法およびアプリケーションの開発方法について説明します。

Oracle Fusion Middleware Oracle WebLogic Server 12c (12.2.1.2.0) Application Adapter 12c (12.2.1.2.0) for J.D. Edwards OneWorld ユーザーズ・ガイド

E85900-01

Copyright © 2001, 2016, Oracle and/or its affiliates. All rights reserved.

原本著者 : Stefan Kostial

協力者 : Vikas Anand, Marian Jones, Sunil Gopal, Bo Stern

このソフトウェアおよび関連ドキュメントの使用と開示は、ライセンス契約の制約条件に従うものとし、知的財産に関する法律により保護されています。ライセンス契約で明示的に許諾されている場合もしくは法律によって認められている場合を除き、形式、手段に関係なく、いかなる部分も使用、複写、複製、翻訳、放送、修正、ライセンス供与、送信、配布、発表、実行、公開または表示することはできません。このソフトウェアのリバース・エンジニアリング、逆アセンブル、逆コンパイルは互換性のために法律によって規定されている場合を除き、禁止されています。

ここに記載された情報は予告なしに変更される場合があります。また、誤りが無いことの保証はいたしかねます。誤りを見つけた場合は、オラクルまでご連絡ください。

このソフトウェアまたは関連ドキュメントを、米国政府機関もしくは米国政府機関に代わってこのソフトウェアまたは関連ドキュメントをライセンスされた者に提供する場合は、次の通知が適用されます。

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

このソフトウェアまたはハードウェアは様々な情報管理アプリケーションでの一般的な使用のために開発されたものです。このソフトウェアまたはハードウェアは、危険が伴うアプリケーション (人的傷害を発生させる可能性があるアプリケーションを含む) への用途を目的として開発されていません。このソフトウェアまたはハードウェアを危険が伴うアプリケーションで使用する場合、安全に使用するために、適切な安全装置、バックアップ、冗長性 (redundancy)、その他の対策を講じることは使用者の責任となります。このソフトウェアまたはハードウェアを危険が伴うアプリケーションで使用したこと起因して損害が発生しても、Oracle Corporation およびその関連会社は一切の責任を負いかねます。

Oracle および Java はオラクルおよびその関連会社の登録商標です。その他の社名、商品名等は各社の商標または登録商標である場合があります。

Intel、Intel Xeon は、Intel Corporation の商標または登録商標です。すべての SPARC の商標はライセンスのもとに使用し、SPARC International, Inc. の商標または登録商標です。AMD、Opteron、AMD ロゴ、AMD Opteron ロゴは、Advanced Micro Devices, Inc. の商標または登録商標です。UNIX は、The Open Group の登録商標です。

このソフトウェアまたはハードウェア、そしてドキュメントは、第三者のコンテンツ、製品、サービスへのアクセス、あるいはそれらに関する情報を提供することがあります。オラクル社およびその関連会社は、第三者のコンテンツ、製品、サービスに関して一切の責任を負わず、いかなる保証もいたしません。オラクル社およびその関連会社は、第三者のコンテンツ、製品、サービスへのアクセスまたは使用によって損失、費用、あるいは損害が発生しても、一切の責任を負いかねます。

目次

はじめに	ix
対象読者	ix
ドキュメントのアクセシビリティについて	ix
関連ドキュメント	ix
表記規則	x
1 概要	
アダプタの機能	1-1
J.D. Edwards OneWorld の概念	1-2
J.D. Edwards OneWorld との統合	1-2
J.D. Edwards OneWorld からの内部リスナーの伝播	1-3
J.D. Edwards OneWorld の相互運用性のフレームワーク	1-3
アダプタのアーキテクチャ	1-5
Oracle Adapter アプリケーション・エクスプローラ (アプリケーション・エクスプローラ) ...	1-6
リソース・アダプタ	1-6
Oracle WebLogic Server Adapter Business Services Engine (BSE) アーキテクチャ	1-6
Oracle WebLogic Server Adapter 汎用 J2CA アーキテクチャ	1-7
ビジネス関数の処理	1-8
BSE と Oracle Adapter J2CA デプロイメントの差異	1-8
サンプル・プロジェクト	1-9
クイック・スタート・ガイド	1-10
インストール	1-10
サード・パーティ・ライブラリ・ファイルのコピー	1-11
構成	1-11
WebLogic Server のデプロイメントおよび統合	1-12
アプリケーション・エクスプローラでの構成、ターゲット、チャンネルの作成	1-12
SOA Suite のサービス・コンポーネントの使用	1-13
Oracle Service Bus の使用	1-13
その他の機能	1-13
2 Oracle Application Adapter for J.D. Edwards OneWorld の構成	
アプリケーション・エクスプローラの起動	2-1
リポジトリ設定の構成	2-2
リポジトリ構成の作成	2-2

BSE 用の構成の作成.....	2-2
J2CA 用の構成の作成	2-3
BSE または J2CA 構成への接続	2-4
J.D. Edwards OneWorld 用の接続 (ターゲット) の確立	2-5
J.D. Edwards OneWorld に対するターゲットの定義.....	2-5
XML スキーマの作成.....	2-10
リクエスト・スキーマおよびレスポンス・スキーマの作成.....	2-10
WSDL の生成 (J2CA 構成のみ)	2-11
アウトバウンド相互作用のための WSDL の生成.....	2-12
Web サービスの作成およびテスト (BSE 構成のみ).....	2-13
Web サービスの作成.....	2-13
Web サービスのテスト.....	2-14
伝播の識別.....	2-14
イベント・アダプタの構成.....	2-15
チャンネルの作成および編集.....	2-15
J.D. Edwards OneWorld Event Listener	2-29
J.D. Edwards OneWorld Event Listener の構成.....	2-30
ランタイムの概要.....	2-32
アウトバウンドおよびインバウンド処理用 JDE.INI ファイルの変更.....	2-32

3 Oracle WebLogic Server のデプロイメントおよび統合

Oracle WebLogic Server とのアダプタの統合	3-1
アダプタのデプロイメント.....	3-1
アダプタ構成の更新.....	3-2
管理対象コネクタ・ファクトリ・オブジェクトの作成.....	3-3
複数の管理対象コネクタ・ファクトリ・オブジェクトの作成.....	3-4
追加の接続ファクトリ値用の WSDL ファイルの変更.....	3-6

4 Oracle SOA Suite での BPEL サービス・コンポーネントの統合

概要.....	4-1
アダプタのデプロイメント.....	4-2
新規のアプリケーション・サーバー接続の構成.....	4-2
サービス統合のためのアウトバウンド BPEL プロセスの設計 (J2CA 構成).....	4-7
リクエスト/レスポンス型のサービス用の WSDL の生成.....	4-8
SOA 用の空のコンポジットの作成	4-9
BPEL アウトバウンド・プロセスの定義.....	4-11
BPEL アウトバウンド・プロセスのデプロイ	4-28
Oracle Enterprise Manager Console での入力 XML ドキュメントの起動.....	4-32
アウトバウンド BPEL およびアウトバウンド・メディアータ・プロセスのテスト.....	4-34
イベント統合のためのインバウンド BPEL プロセスの設計 (J2CA 構成).....	4-34
イベント統合のための WSDL の生成.....	4-35
SOA 用の空のコンポジットの作成	4-40
BPEL インバウンド・プロセスの定義.....	4-41
BPEL インバウンド・プロセスのデプロイ	4-47

J.D. Edwards OneWorld でのイベントのトリガー.....	4-48
サービス統合のためのアウトバウンド BPEL プロセスの設計 (BSE 構成).....	4-53
Web サービスを使用したリクエスト / レスポンス型サービス用の WSDL ファイルの生成...	4-53
SOA 用の空のコンポジットの作成	4-54
BPEL アウトバウンド・プロセスの定義.....	4-54
5 Oracle SOA Suite でのメディエータ・サービス・コンポーネントとの統合	
新規のアプリケーション・サーバー接続の構成.....	5-2
メディエータ・アウトバウンド・プロセスの構成 (J2CA 構成).....	5-2
SOA 用の空のコンポジットの作成	5-2
メディエータ・アウトバウンド・プロセスの定義.....	5-3
メディエータ・アウトバウンド・プロセスのデプロイ.....	5-10
Oracle Enterprise Manager Console での入力 XML ドキュメントの起動.....	5-11
メディエータ・インバウンド・プロセスの構成 (J2CA 構成).....	5-11
SOA 用の空のコンポジットの作成	5-11
メディエータ・インバウンド・プロセスの定義.....	5-11
メディエータ・アウトバウンド・プロセスの構成 (BSE 構成).....	5-17
SOA 用の空のコンポジットの作成	5-18
メディエータ・アウトバウンド・プロセスの定義.....	5-18
6 Oracle SOA Suite での BPM サービス・コンポーネントの統合	
概要.....	6-1
アダプタのデプロイメント.....	6-2
新規のアプリケーション・サーバー接続の構成.....	6-2
トランスフォーメーションを使用したサービス統合のためのアウトバウンド BPM プロセスの設計 (J2CA 構成).....	6-2
BPM 用の空のコンポジットの作成.....	6-2
BPM アウトバウンド・プロセスの定義.....	6-3
12c でのデプロイメントに関する既知の問題への対応.....	6-25
BPM アウトバウンド・プロセスのデプロイ	6-25
Oracle Enterprise Manager Console での入力 XML ドキュメントの起動.....	6-26
トランスフォーメーションを使用したイベント統合のためのインバウンド BPM プロセスの設計 (J2CA 構成).....	6-27
BPM 用の空のコンポジットの作成.....	6-27
BPM インバウンド・プロセスの定義.....	6-28
トランスフォーメーションを使用したサービス統合のためのアウトバウンド BPM プロセスの設計 (BSE 構成).....	6-42
BPM 用の空のコンポジットの作成.....	6-43
BPM アウトバウンド・プロセスの定義.....	6-43
7 sbconsole を使用した Oracle Service Bus 用のアウトバウンドおよびインバウンド・プロセスの構成	
Application Adapter と Oracle Service Bus の統合の概要.....	7-1
sbconsole を使用したアウトバウンド・プロセスの構成 (J2CA 構成).....	7-2

Oracle Service Bus の起動およびプロジェクト・フォルダの作成.....	7-2
Oracle Service Bus 統合のためのアプリケーション・エクスプローラのクラス・パスの設定.....	7-6
アプリケーション・エクスプローラから Oracle Service Bus への WSDL の公開.....	7-6
WSDL ベースのビジネス・サービスの構成.....	7-7
ファイル・タイプ・ビジネス・サービスの構成.....	7-9
パイプラインとプロキシ・サービスの構成.....	7-13
sbconsole を使用したインバウンド・プロセスの構成 (J2CA 構成).....	7-22
Oracle Service Bus の起動およびプロジェクト・フォルダの作成.....	7-22
Oracle Service Bus 統合のためのアプリケーション・エクスプローラのクラス・パスの設定.....	7-22
イベント統合のための WSDL の生成.....	7-23
WSDL ベースのプロキシ・サービスの構成.....	7-23
ファイル・タイプ・ビジネス・サービスの構成.....	7-26
パイプラインの構成.....	7-29
sbconsole を使用したアウトバウンド・プロセスの構成 (BSE 構成).....	7-36
Oracle Service Bus の起動およびプロジェクト・フォルダの作成.....	7-36
Oracle Service Bus 統合のためのアプリケーション・エクスプローラのクラス・パスの設定.....	7-36
アプリケーション・エクスプローラから Oracle Service Bus への WSDL の公開.....	7-37
ファイル・タイプ・ビジネス・サービスの構成.....	7-38
WSDL ベースのビジネス・サービスの構成.....	7-38
パイプラインとプロキシ・サービスの構成.....	7-40
Oracle Service Bus を使用した JMS プロキシ・サービスの構成 (J2CA 構成).....	7-49
Oracle Service Bus を使用した HTTP プロキシ・サービスの構成 (J2CA 構成).....	7-64

8 JDeveloper を使用した Oracle Service Bus 用のアウトバウンドおよびインバウンド・プロセスの構成

JDeveloper を使用した OSB アウトバウンド・プロセスの構成 (J2CA 構成).....	8-1
OSB 用のサービス・バス・アプリケーションの作成.....	8-2
OSB アウトバウンド・プロセスの定義.....	8-3
OSB アウトバウンド・プロセスのデプロイ.....	8-16
JDeveloper を使用した OSB インバウンド・プロセスの構成 (J2CA 構成).....	8-18
OSB 用のサービス・バス・アプリケーションの作成.....	8-18
OSB インバウンド・プロセスの定義.....	8-18
OSB インバウンド・プロセスのデプロイ.....	8-26
JDeveloper を使用した OSB アウトバウンド・プロセスの構成 (BSE 構成).....	8-27
OSB 用のサービス・バス・アプリケーションの作成.....	8-27
OSB アウトバウンド・プロセスの定義.....	8-27
OSB アウトバウンド・プロセスのデプロイ.....	8-41
JDeveloper を使用した JMS インバウンド・プロセスの構成 (J2CA 構成).....	8-41
JDeveloper を使用した JMS アウトバウンド・プロセスの構成 (J2CA 構成).....	8-47
JDeveloper を使用した HTTP アウトバウンド・プロセスの構成 (J2CA 構成).....	8-54

9 主要な機能

ロギング機能の構成.....	9-1
J2CA コネクタ・アプリケーションのログ・ファイル管理の構成.....	9-1
Business Services Engine (BSE) のログ・ファイル管理の構成.....	9-8
診断機能の構成.....	9-11
サポートされるプロトコル.....	9-12
SOA デバッグ機能の構成.....	9-13
SOA デバッガの使用のガイドライン.....	9-13
前提条件.....	9-13
Oracle JDeveloper での BPEL プロセスのデバッグ.....	9-13
Oracle JDeveloper での OSB プロセスのデバッグ.....	9-27
例外フィルタ.....	9-45
例外フィルタの構成.....	9-45
Oracle SOA Suite (BPEL、メディエータまたは BPM) の資格証明マッピング.....	9-56
資格証明マッピングの構成.....	9-57
Oracle Service Bus (OSB) の資格証明マッピング.....	9-61
資格証明マッピングの構成.....	9-62

10 トラブルシューティングおよびエラー・メッセージ

トラブルシューティング.....	10-1
アプリケーション・エクスプローラ.....	10-2
J.D. Edwards One World.....	10-3
Oracle Adapter J2CA.....	10-4
BSE エラー・メッセージ.....	10-4
BSE での一般的なエラー処理.....	10-4
アダプタ特定のエラー処理.....	10-5

A アウトバウンドおよびインバウンド処理用 J.D. Edwards OneWorld の構成

アウトバウンドおよびインバウンド処理用 JDE.INI ファイルの変更.....	A-1
GenJava 開発ツールの使用 (アウトバウンド処理).....	A-2
GenJava の実行.....	A-2
J.D. Edwards OneWorld イベントのトリガー.....	A-4
アウトバウンド・スケジューラ・サブシステム処理 (R00460) の起動.....	A-6
サブシステム処理の検証.....	A-7
イベントをトリガーするための P4210(受注オーダー) の構成.....	A-9
構成ステップの検証.....	A-11

用語集

索引

はじめに

Oracle WebLogic Server のための *Oracle Fusion Middleware Application Adapters Application Adapter for J.D. Edwards OneWorld ユーザーズ・ガイド*へようこそ。このドキュメントでは、J.D. Edwards OneWorld システムとの統合方法およびアプリケーションの開発方法について説明します。

対象読者

このドキュメントは、J.D. Edwards OneWorld システムとの統合およびアプリケーションの開発を行うシステム管理者や開発者を対象にしています。

ドキュメントのアクセシビリティについて

オラクルのアクセシビリティについての詳細情報は、Oracle Accessibility Program の Web サイト (<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>) を参照してください。

Oracle Support へのアクセス

サポートをご契約のお客様には、My Oracle Support を通して電子支援サービスを提供しています。詳細情報は (<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info>) か、聴覚に障害のあるお客様は (<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs>) を参照してください。

関連ドキュメント

詳細は、Oracle Enterprise Repository 12c (12.2.1.2.0) キュメント・セット内の次のドキュメントを参照してください。

- *Oracle Fusion Middleware Oracle WebLogic Server Application Adapters インストール・ガイド*
- *Oracle Fusion Middleware Oracle WebLogic Server Application Adapter アップグレード・ガイド*
- *Oracle Fusion Middleware Oracle WebLogic Server Application Adapter ベスト・プラクティス・ガイド*
- Oracle Unified Method (OUM)

Oracle Unified Method(OUM) では、その他の管理情報を豊富に提供しています。OUM は、オラクル社の従業員、パートナー・ネットワーク認定パートナーや認定アドバンテージ・パートナーとともに、OUM 顧客プログラム、またはオラクル社によるコンサルティング・サービス提供プロジェクトにご参加いただいているお客様にもご利用いただけます。OUM はソフトウェア開発および実装プロジェクトの計画、実施、管理用の Web デプロイ型ツールキットです。

OUM の詳細は、次の URL の OUM FAQ を参照してください。

http://my.oracle.com/portal/page/myo/ROOTCORNER/KNOWLEDGEAREAS1/BUSINESS_PRACTICE/Methods/Learn_about_OUM.html

表記規則

このドキュメントでは次の表記規則を使用します。

規則	意味
太字	太字は、操作に関連する Graphical User Interface 要素、または本文中で定義されている用語および用語集に記載されている用語を示します。
斜体	イタリックは、ユーザーが特定の値を指定するプレースホルダー変数を示します。
固定幅フォント	固定幅フォントは、段落内のコマンド、URL、サンプル内のコード、画面に表示されるテキスト、または入力するテキストを示します。

Oracle WebLogic Server は、Oracle Application Adapter for J.D. Edwards OneWorld を使用して J.D. Edwards OneWorld システムに接続します。Oracle Application Adapter for J.D. Edwards OneWorld を使用すると、J.D. Edwards OneWorld との接続が可能になり、J.D. Edwards OneWorld システム上で相互作用を実行できます。

注意： このドキュメントでは、<ORACLE_HOME> は、12c (12.2.1.0.0) SOA/OSB のインストール先であるホーム・ディレクトリを指します。

<ADAPTER_HOME> は、次を指します。

- SOA の場合：

<ORACLE_HOME>\soa\soa\thirdparty\ApplicationAdapters

- OSB の場合：

<ORACLE_HOME>\osb\3rdparty\ApplicationAdapters

この章の内容は以下のとおりです。

- 1.1 項「アダプタの機能」
- 1.2 項「J.D. Edwards OneWorld の概念」
- 1.3 項「J.D. Edwards OneWorld との統合」
- 1.4 項「アダプタのアーキテクチャ」
- 1.5 項「BSE と Oracle Adapter J2CA デプロイメントの差異」
- 1.6 項「サンプル・プロジェクト」
- 1.7 項「クイック・スタート・ガイド」

1.1 アダプタの機能

Oracle Application Adapter for J.D. Edwards OneWorld は、J.D. Edwards システムと他のアプリケーション、データベースまたは外部ビジネス・パートナー・システムとの間でリアルタイムのビジネス・データを交換する手段を提供します。この[アダプタ](#)により、J.D. Edwards とのインバウンド処理およびアウトバウンド処理が可能になります。

Oracle Application Adapter for J.D. Edwards OneWorld は、J2EE Connector Architecture (J2CA) 1.0 リソース・アダプタとしてデプロイできます。このデプロイを Oracle Adapter J2CA と呼びます。また、Oracle Adapter Business Services Engine (BSE) と呼ばれるように、Web サービス・サーブレットとしてもデプロイできます。

Oracle Application Adapter for J.D. Edwards OneWorld は、XML メッセージを使用することにより、非 J.D. Edwards OneWorld アプリケーションがサービスおよびイベントを使用して J.D. Edwards OneWorld と通信およびトランザクション交換ができるようにします。サービスおよびイベントは、次のように説明されます。

- サービス: この機能を使用して、アプリケーションは J.D. Edwards OneWorld ビジネス・イベントを開始します。
- イベント: この機能を使用して、アプリケーションは J.D. Edwards OneWorld ビジネス・イベントの発生時にのみ J.D. Edwards OneWorld データにアクセスします。

イベント機能をサポートするため、チャンネルがサポートされます。[チャンネル](#)とは、バックエンドまたは他のタイプのシステムの特定インスタンスへの構成済の接続のことです。

チャンネルは、エンタープライズ情報システム (EIS) アプリケーションからリアルタイムでイベントを受信するアダプタ・コンポーネントです。チャンネル・コンポーネントは、ファイル・リーダー、HTTP リスナー、TCP/IP リスナー、または FTP リスナーのいずれかになります。チャンネルは、常に EIS に固有です。アダプタでは、特定の EIS に対して複数のチャンネルがサポートされるため、ユーザーは、デプロイメント要件に基づいて最適なチャンネル・コンポーネントを選択できます。

Oracle Application Adapter for J.D. Edwards OneWorld は、次のものを提供します。

- J2CA 1.0 および 1.5 リソース・アダプタ用の XML スキーマおよび WSDL
- BSE 用 Web サービス

関連項目: [Oracle Application Server Adapter コンセプト・ガイド](#)

1.2 J.D. Edwards OneWorld の概念

Oracle Application Adapter for J.D. Edwards OneWorld を使用して、住所録や購買オーダー、受注オーダーなどの J.D. Edwards OneWorld マスター・ビジネス関数をコールできます。また、J.D. Edwards OneWorld を J.D. Edwards OneWorld 以外のシステムと接続するための統一作業の一部として、このアダプタを使用できます。

Oracle Application Adapter for J.D. Edwards OneWorld は、XML ドキュメントを受信したり、J.D. Edwards OneWorld ThinNet API を介して XML ドキュメントを J.D. Edwards OneWorld に渡すことで 1 つ以上の J.D. Edwards OneWorld マスター・ビジネス関数 (MBF) を実行することができます。

1.3 J.D. Edwards OneWorld との統合

この項では、次のトピックについて説明します。

- [1.3.1 項「J.D. Edwards OneWorld からの内部リスナーの伝播」](#)
- [1.3.2 項「J.D. Edwards OneWorld の相互運用性のフレームワーク」](#)

J.D. Edwards OneWorld では、相互運用性を提供するために複数のメソッドと技術をサポートしています。次の 3 つのエントリ・ポイントがサポートされています。

- フラット・ファイル
- データベース表
- マスター・ビジネス関数 (MBF) 対話型コール

リクエストを J.D. Edwards OneWorld に送信できるように Oracle AS アダプタを構成します。アダプタでは、XML ドキュメントに埋め込まれている J.D. Edwards OneWorld マスター・ビジネス関数 (MBF) のリクエストを処理し、バックエンド J.D. Edwards OneWorld システムに転送します。結果レスポンス情報が返され、その後のルーティング用に処理されます。

Oracle Application Adapter for J.D. Edwards OneWorld では、クライアントから XML リクエスト・ドキュメントを受信し、ターゲットのエンタープライズ情報システム (EIS) で特定の機能をコールできます。Oracle Application Adapter for J.D. Edwards OneWorld はリクエスト・メッセージのコンシューマとして動作し、レスポンスを返します。アダプタにより、次の機能が実行されます。

- 旧システム、他の EIS または EIS 以外のクライアントからのリクエストの受信
- XML リクエスト・ドキュメントを EIS 固有の書式に変換
 - リクエスト・ドキュメントはリクエスト XML スキーマに準拠しています。スキーマは EIS のメタデータに基づいています。
- EIS の基礎となる機能のコールおよびそのレスポンスの待機
- EIS 固有のデータ形式から XML ドキュメントへのレスポンスの変換
 - レスポンス・ドキュメントは、アダプタによって生成されるレスポンス XML スキーマに準拠しています。
 - スキーマは EIS のメタデータに基づいています。

アダプタが J.D. Edwards OneWorld からメッセージを受信できるように、チャンネルを構成できます。チャンネルが受信する情報は XML レコードのビルドに使用され、さらなる処理のために指定した配置に転送されます。

チャンネルは EIS 固有のメッセージのコンシューマで、レスポンスを返す場合と返さない場合があります。チャンネルにより、次の機能が実行されます。

- EIS クライアントからのメッセージの受信
- EIS 固有のメッセージ書式の XML フォーマットへの変換

1.3.1 J.D. Edwards OneWorld からの内部リスナーの伝播

J.D. Edwards OneWorld リスナーと外部システムの統合は、リバースの場合を除き、アウトバウンド処理と同様です。データ・エクスポート制御表では、トランザクションを外部システムと統合させる必要があるかどうかの決定が保持されます。トランザクションを統合させる必要がある場合、MBF により、未編集のトランザクション表に対するすべての追加、変更および削除についてログ処理が行われます。トランザクション情報が表に書き込まれると、レコードのキーが MBF からサブシステム・データ・キューに送信されます。

サブシステム・データ・キューでは、汎用アウトバウンド・サブシステム・バッチ処理を起動して新規レコードの処理をトリガーし、すべてのインバウンド・トランザクションを処理します。J.D. Edwards アウトバウンド・サブシステムがデータ・エクスポート制御表にアクセスし、構成済の外部サブスクライバの実行を決定します。

1.3.2 J.D. Edwards OneWorld の相互運用性のフレームワーク

J.D. Edwards OneWorld を使用すると、相互運用性フレームワークを利用したシステム統合が可能です。アダプタでは、フレームワークを使用し、様々な統合アクセス・メソッドを利用して最大限の柔軟性と機能を提供します。

Oracle Application Adapter for J.D. Edwards OneWorld では、次の統合アクセス・メソッドがサポートされています。

- J.D. Edwards OneWorld ThinNet API
- J.D. Edwards OneWorld XML
- J.D. Edwards OneWorld の未編集のトランザクション表 (Z 表)

図 1-1 に、アウトバウンド処理フレームワークを示します。

アダプタでは J.D. Edwards OneWorld ThinNet API を使用して、J.D. Edwards OneWorld アプリケーションと通信します。ThinNet API を使用すると、アダプタで、1 つ以上の MBF を単一の作業単位 (UOW) で実行できます。いずれかの MBF が失敗した場合、部分的な更新を避けるために UOW 全体が失敗します。アダプタにより MBF が実行されるため、データ検証、ビジネス・ルールおよび基礎となるデータベースとの通信は J.D. Edwards OneWorld アプリケーションにより処理されます。

図 1-1 J.D. Edwards OneWorld アウトバウンド処理

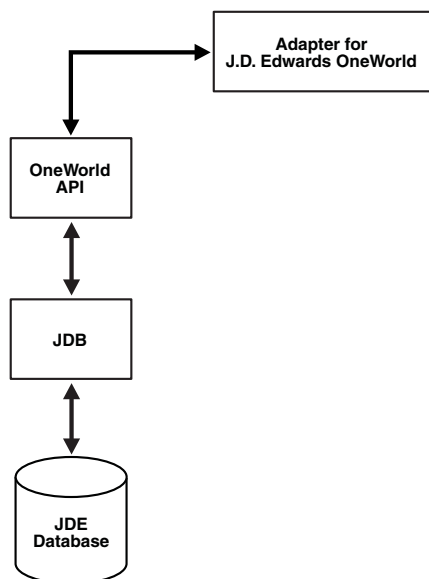
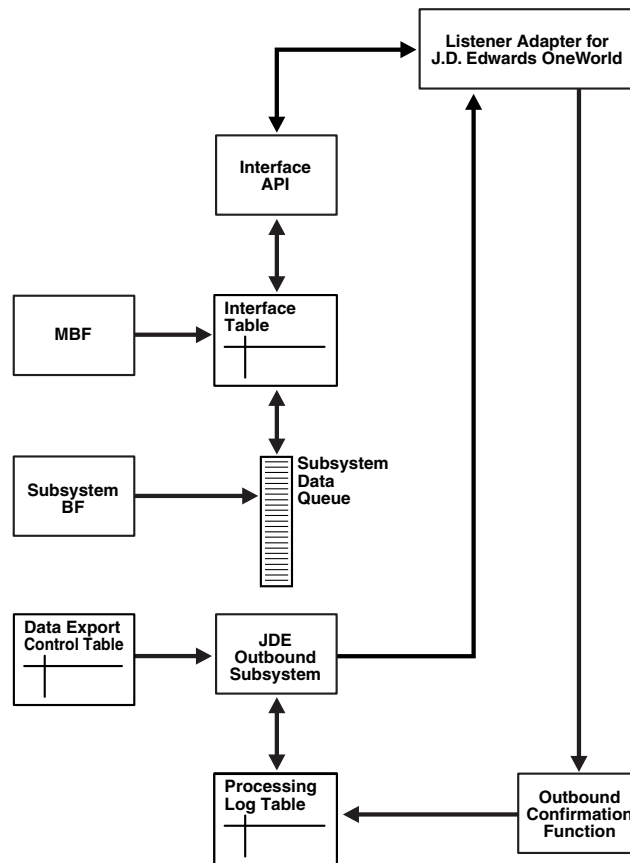


図 1-2 に、インバウンド処理フレームワークを示します。

図 1-2 J.D. Edwards OneWorld インバウンド処理



アウトバウンド処理では、指定の MBF が J.D. Edwards OneWorld 環境で実行されるとイベントを開始します。MBF により、イベントの必須情報が適切なインタフェース表へ書き込まれ、イベントの発生がサブシステム Batch Function(BF) に通知されます。サブシステム BF により、サブシステム・データ・キューにイベントに関するエントリが配置されます。

J.D. Edwards OneWorld アウトバウンド・サブシステムはデータ・キュー・エントリを受信すると、通知のために外部処理用のデータ・エクスポート制御表を検索します。J.D. Edwards OneWorld アウトバウンド・サブシステムにより、通知付きで Oracle Application Adapter for J.D. Edwards OneWorld リスナーがコールされます。リスナーにより通知がジェネレータに渡されます。ジェネレータは、J.D. Edwards OneWorld ThinNet API を使用し、インタフェース表から適切な情報を取得します。

1.4 アダプタのアーキテクチャ

Oracle Application Adapter for J.D. Edwards OneWorld は、次のいずれかのコンポーネントとともにアプリケーション・エクスプローラを使用します。

- Oracle WebLogic Server Adapter Business Services Engine (BSE)
- Enterprise Connector for J2EE Connector Architecture (J2CA)

この項では、次のトピックについて説明します。

- 1.4.1 項「Oracle Adapter アプリケーション・エクスプローラ (アプリケーション・エクスプローラ)」
- 1.4.2 項「リソース・アダプタ」

- 1.4.3 項「Oracle WebLogic Server Adapter Business Services Engine (BSE) アーキテクチャ」
- 1.4.4 項「Oracle WebLogic Server Adapter 汎用 J2CA アーキテクチャ」
- 1.4.5 項「ビジネス関数の処理」

1.4.1 Oracle Adapter アプリケーション・エクスプローラ (アプリケーション・エクスプローラ)

アプリケーション・エクスプローラは、データベース接続の構成や Web サービスおよびイベントの作成に使用します。それを、BSE または Enterprise Connector for J2EE Connector Architecture (J2CA) とともに Web サービス環境で動作するように構成できます。J2CA 環境で動作する場合、コネクタでは Web サービスのかわりにアダプタを使用して迅速な統合サービスを行うため、Common Client Interface (CCI) を使用します。

BSE および J2CA 用コネクタは、アプリケーション・エクスプローラおよびアダプタ付属のアプリケーション・サーバーにデプロイされます。

アプリケーション・エクスプローラでは、ビジネス関数を J.D. Edwards システムで参照するためにエクスプローラ・メタファを使用します。アプリケーション・エクスプローラを使用すると、関連付けられたビジネス関数用の XML スキーマおよび Web サービスを作成できます。

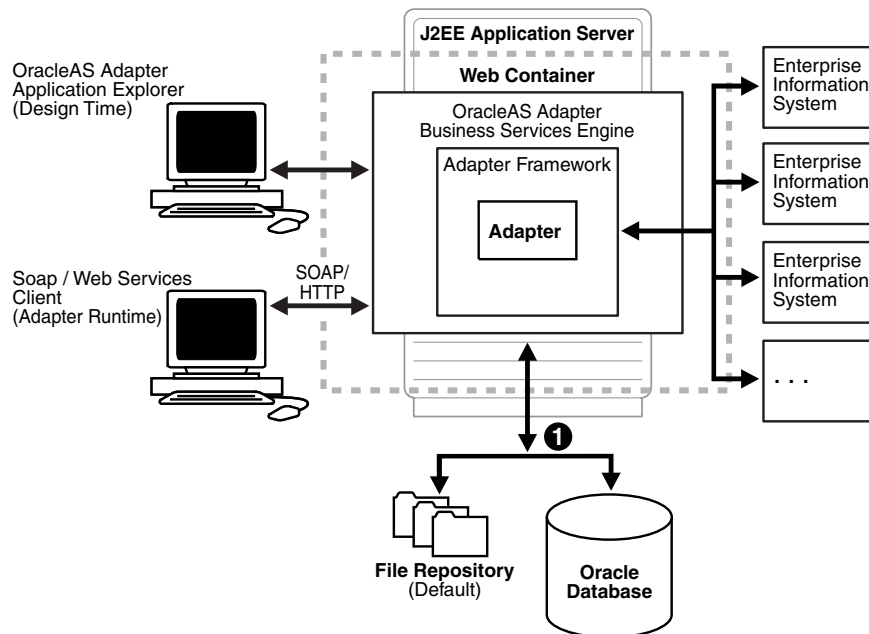
1.4.2 リソース・アダプタ

Oracle Application Adapter for J.D. Edwards OneWorld は、リソース・アダプタとも呼ばれる J2CA ベースのコンポーネントです。リソース・アダプタを使用すると、本来相互通信できるように設計されていなかったアプリケーションに接続できます。アダプタは双方向であるため、エンタープライズ情報システム (EIS) にリクエストを送信でき、EIS で発生したイベントについても通知を受信できます。

1.4.3 Oracle WebLogic Server Adapter Business Services Engine (BSE) アーキテクチャ

図 1-3 に、パッケージされているアプリケーションの Oracle Web サービス・アダプタ用の汎用アーキテクチャを示します。このアダプタは、J2EE アプリケーション・サーバーで Web コンテナにデプロイすることにより、BSE と連携します。

図 1-3 Oracle Adapter Business Services Engine (BSE) アーキテクチャ



① Use either the default file repository or an Oracle database as your repository.

注意： BSE用のファイル・リポジトリは本番環境では使用しないでください。

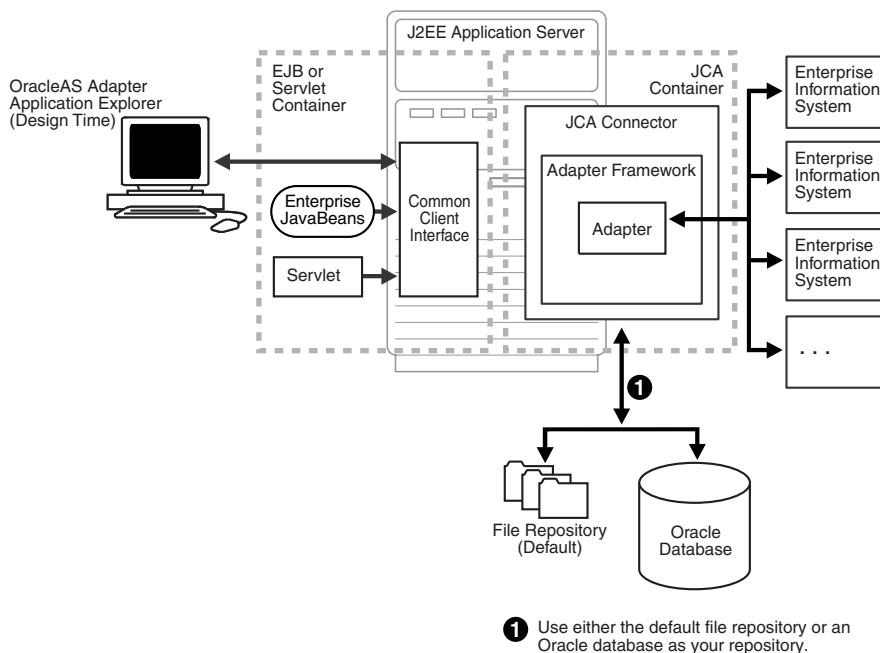
BSEとともにデプロイされるデザインタイム・ツールのアプリケーション・エクスプローラを使用して、アダプタ接続の構成、EISオブジェクトの参照、サービスの構成およびEISイベントをリスニングするリスナーの構成ができます。これらの操作の実行中に作成されるメタデータは、BSEによりリポジトリ内に格納されます。

BSEは、クライアントからリクエストの受信、EISとの対話、EISからのレスポンスのクライアントへの返信にSOAPをプロトコルとして使用します。

1.4.4 Oracle WebLogic Server Adapter 汎用 J2CA アーキテクチャ

図 1-4 に、パッケージされているアプリケーションの Oracle J2CA アダプタ用の汎用アーキテクチャを示します。J2CA コネクタは標準の J2CA コンテナにデプロイされ、アダプタへのホスト・コンテナとして機能します。コネクタは、リポジトリで構成されます。

図 1-4 Oracle Adapter 汎用 J2CA アーキテクチャ



アプリケーション・エクスプローラは、コネクタとともに動作する設計ツールで、これを使用して、アダプタ接続の構成、EIS オブジェクトの参照、サービスの構成、および EIS イベントをリスニングするリスナーの構成を行います。これらの操作中に作成されるメタデータは、コネクタによりリポジトリに格納されます。リポジトリにはファイル・システムまたは Oracle データベースを使用できます。それは RAR ファイルとしてデプロイされ、`ra.xml` と呼ばれる関連デプロイメント・ディスクリプタを持っています。Oracle WebLogic Server デプロイメント・ディスクリプタの `ra.xml` を編集することにより、複数のコネクタ・ファクトリを作成できます。詳細は、[第 3 章「Oracle WebLogic Server のデプロイメントおよび統合」](#)を参照してください。

1.4.5 ビジネス関数の処理

Oracle Application Adapter for J.D. Edwards OneWorld では、J.D. Edwards ThinNet API を介して、J.D. Edwards OneWorld ビジネス関数を処理できます。API を使用すると、複雑で非実用的なバッチ処理を作成する必要がありません。さらに、リスナーが HTTP、TCP またはファイル接続を介して定義されているため、IBM MQSeries などのトランスポート層は必要ありません。

Oracle Application Adapter for J.D. Edwards OneWorld を介して J.D. Edwards OneWorld にアクセスする外部アプリケーションは、XML スキーマまたは Web サービスを使用して、外部アプリケーションとアダプタとの間でデータを渡します。[第 2 章「Oracle Application Adapter for J.D. Edwards OneWorld の構成」](#)では、アプリケーション・エクスプローラを使用して、アダプタを使用した J.D. Edwards マスター・ビジネス関数 (MBF) 用 XML スキーマと Web サービスを作成する方法について説明します。

1.5 BSE と Oracle Adapter J2CA デプロイメントの差異

Oracle Application Adapter for J.D. Edwards OneWorld を Oracle SOA Suite コンポーネント (たとえば、BPEL、メディエータ、BPM または OSB) とともに使用する場合、次の点に注意してください。

- Oracle Adapter J2CA デプロイメントのみ、Oracle SOA Suite コンポーネントとのインバウンド統合 (イベント通知) がサポートされます。
- Oracle Adapter J2CA and BSE デプロイメントでは、Oracle SOA Suite とのアウトバウンド統合 (リクエスト / レスポンス型サービス) がサポートされます。

BSE と Oracle Adapter J2CA のデプロイメント間の差異には、次の 2 つの要素があります。これらの要素を理解することは、デプロイメント・オプションの選択に役立ちます。

1. BSE には次の利点があります。
 - Oracle WebLogic Server の個別のインスタンスでデプロイできます。
 - 優れた負荷分散を提供します。
 - アプリケーションの構築について、Service Oriented Architecture(SOA) モデルにより厳密に準拠します。
2. Oracle Adapter J2CA の方が BSE よりもわずかに高いパフォーマンスが得られます。

1.6 サンプル・プロジェクト

Oracle BPEL、メディエータ、BPM、OSB ツールを使用したアウトバウンドおよびインバウンドの統合シナリオを実演する Oracle Application Adapter for J.D. Edwards OneWorld 用のサンプル・プロジェクトが、Application Adapter とともにインストールされます。次の表にサンプル・プロジェクトの場所を示します。

サンプル・プロジェクト	場所
アウトバウンド BPEL プロセス (J2CA)	<ADAPTER_HOME>\etc\sample\JDEdwards_Samples.zip\JDEdwards_Samples\BPEL\J2CA\Outbound_Project
インバウンド BPEL プロセス (J2CA)	<ADAPTER_HOME>\etc\sample\JDEdwards_Samples.zip\JDEdwards_Samples\BPEL\J2CA\Inbound_Project
アウトバウンド BPEL プロセス (BSE)	<ADAPTER_HOME>\etc\sample\JDEdwards_Samples.zip\JDEdwards_Samples\BPEL\BSE\Outbound_Project
アウトバウンド・メディエータ・プロセス (J2CA)	<ADAPTER_HOME>\etc\sample\JDEdwards_Samples.zip\JDEdwards_Samples\Mediator\J2CA\Outbound_Project
インバウンド・メディエータ・プロセス (J2CA)	<ADAPTER_HOME>\etc\sample\JDEdwards_Samples.zip\JDEdwards_Samples\Mediator\J2CA\Inbound_Project
アウトバウンド・メディエータ・プロセス (BSE)	<ADAPTER_HOME>\etc\sample\JDEdwards_Samples.zip\JDEdwards_Samples\Mediator\BSE\Outbound_Project
アウトバウンド BPM プロセス (J2CA)	<ADAPTER_HOME>\etc\sample\JDEdwards_Samples.zip\JDEdwards_Samples\BPM\J2CA\Outbound_Project
インバウンド BPM プロセス (J2CA)	<ADAPTER_HOME>\etc\sample\JDEdwards_Samples.zip\JDEdwards_Samples\BPM\J2CA\Inbound_Project
アウトバウンド BPM プロセス (BSE)	<ADAPTER_HOME>\etc\sample\JDEdwards_Samples.zip\JDEdwards_Samples\BPM\BSE\Outbound_Project

サンプル・プロジェクト	場所
アウトバウンド OSB sbconsole プロセス (J2CA)	<ADAPTER_HOME>\etc\sample\JDEdwards_Samples.zip\JDEdwards_Samples\OSB\J2CA\JDEdwards_Sample_J2CA_OSB_Outbound_Project
インバウンド OSB sbconsole プロセス (J2CA)	<ADAPTER_HOME>\etc\sample\JDEdwards_Samples.zip\JDEdwards_Samples\OSB\J2CA\JDEdwards_Sample_J2CA_OSB_Inbound_Project
アウトバウンド OSB sbconsole プロセス (BSE)	<ADAPTER_HOME>\etc\sample\JDEdwards_Samples.zip\JDEdwards_Samples\OSB\BSE\JDEdwards_Sample_BSE_OSB_Outbound_Project
アウトバウンド OSB Jdeveloper プロセス (J2CA)	<ADAPTER_HOME>\etc\sample\JDEdwards_Samples.zip\JDEdwards_Samples\OSB_Jdeveloper\J2CA\JDEdwards_Sample_J2CA_OSB_Outbound_Project
インバウンド OSB Jdeveloper プロセス (J2CA)	<ADAPTER_HOME>\etc\sample\JDEdwards_Samples.zip\JDEdwards_Samples\OSB_Jdeveloper\J2CA\JDEdwards_Sample_J2CA_OSB_Inbound_Project
アウトバウンド OSB Jdeveloper プロセス (BSE)	<ADAPTER_HOME>\etc\sample\JDEdwards_Samples.zip\JDEdwards_Samples\OSB_Jdeveloper\BSE\JDEdwards_Sample_BSE_OSB_Outbound_Project

1.7 クイック・スタート・ガイド

この項では、Oracle Application Adapter for J.D. Edwards OneWorld をインストールして構成し、すぐに使用するための基本的な方法について簡単に説明します。次のトピックが含まれています。

- [1.7.1 項「インストール」](#)
- [1.7.2 項「サード・パーティ・ライブラリ・ファイルのコピー」](#)
- [1.7.3 項「構成」](#)
- [1.7.4 項「WebLogic Server のデプロイメントおよび統合」](#)
- [1.7.5 項「アプリケーション・エクスプローラでの構成、ターゲット、チャンネルの作成」](#)
- [1.7.6 項「SOA Suite のサービス・コンポーネントの使用」](#)
- [1.7.7 項「Oracle Service Bus の使用」](#)
- [1.7.8 項「その他の機能」](#)

1.7.1 インストール

Oracle Application Adapter for J.D. Edwards OneWorld をインストールするには、Oracle Fusion Middleware Application Adapter インストーラをダウンロードし、SOA/OSB のインストールを完了します。

Oracle Fusion Middleware Application Adapter のインストールの詳細は、*Oracle Fusion Middleware Oracle WebLogic Server Application Adapters インストレーション・ガイド*を参照してください。

1.7.2 サード・パーティ・ライブラリ・ファイルのコピー

アダプタのインストールが完了した後、J.D. Edwards OneWorld の必要なサード・パーティ・ライブラリ・ファイルを次のディレクトリにコピーします。

```
<ADAPTER_HOME>\lib
```

```
<ORACLE_HOME>\user_projects\domains\base_domain\lib
```

Oracle Fusion Middleware Application Adapter のインストールの詳細は、*Oracle Fusion Middleware Oracle WebLogic Server Application Adapters インストール・ガイド*を参照してください。

1.7.3 構成

<ADAPTER_HOME> にナビゲートし、次の変更を行います。

1. `iwafjca.rar\META-INF\ra.xml` を開き、表 1-1 に示すように、指定された `config-property-name` パラメータの下に次の値を追加します。

表 1-1

Config-Property-Name	Config-Property-Value
<code>IWayHome</code>	<ADAPTER_HOME> たとえば、次のようになります。 <ul style="list-style-type: none"> ■ SOA の場合: C:\12C_soa\soa\soa\thirdparty\ApplicationAdapters ■ OSB の場合: C:\12c_OSB\osb\3rdparty\ApplicationAdapters
<code>IWayConfig</code>	構成の名前。たとえば、次のようになります。 jca_sample

2. `ibse.war\WEB-INF\web.xml` を開き、表 1-2 に示すように、指定された `param-name` パラメータの下に次の値を追加します。

表 1-2

Param-Name	Param-Value
<code>ibseroot</code>	<ADAPTER_HOME>\ibse.war たとえば、次のようになります。 <ul style="list-style-type: none"> ■ SOA の場合: C:\12C_soa\soa\soa\thirdparty\ApplicationAdapters\ibse.war ■ OSB の場合: C:\12c_OSB\osb\3rdparty\ApplicationAdapters\ibse.war

表 1-2 (続き)

Param-Name	Param-Value
<i>IWay.home</i>	<ADAPTER_HOME> たとえば、次のようになります。 <ul style="list-style-type: none"> ■ SOA の場合 : C:\12C_soa\soa\soa\thirdparty\ApplicationAdapters ■ OSB の場合 : C:\12c_OSB\osb\3rdparty\ApplicationAdapters
<i>IWay.config</i>	構成の名前。たとえば、次のようになります。 IBSE

注意： これらのステップは、ファイル・リポジトリを構成する場合にのみ使用できます。データベース・リポジトリの構成に関する詳細および一般的な構成情報は、[第 2 章「Oracle Application Adapter for J.D. Edwards OneWorld の構成」](#) および [第 3 章「Oracle WebLogic Server のデプロイメントおよび統合」](#) を参照してください。

1.7.4 WebLogic Server のデプロイメントおよび統合

1. WebLogic サーバーを起動し、WebLogic コンソールを開きます。
2. アダプタ・コンポーネント (ibse.war、iwafjca.war、iwafjca.rar ファイル) をデプロイし、デプロイしたアダプタ・コンポーネントを起動します。

デプロイメント、統合およびターゲットの作成の詳細は、[第 3 章「Oracle WebLogic Server のデプロイメントおよび統合」](#) を参照してください。

1.7.5 アプリケーション・エクスプローラでの構成、ターゲット、チャネルの作成

アプリケーション・エクスプローラでの構成、ターゲット、チャネルの作成の詳細は、次に示すこのユーザー・ガイドの各項を参照してください。

- アプリケーション・エクスプローラの起動 : [2.1 項「アプリケーション・エクスプローラの起動」](#)
- BSE 構成の作成 : [2.3.1 項「BSE 用の構成の作成」](#)
- J2CA 構成の作成 : [2.3.2 項「J2CA 用の構成の作成」](#)
- 作成した構成の接続 : [2.3.3 項「BSE または J2CA 構成への接続」](#)
- ターゲットの作成および接続 : [2.4 項「J.D. Edwards OneWorld 用の接続\(ターゲット\)の確立」](#)
- Web サービスの作成およびテスト : [2.7 項「Web サービスの作成およびテスト \(BSE 構成のみ\)」](#)
- WSDL ファイルの生成 : [2.6 項「WSDL の生成 \(J2CA 構成のみ\)」](#)
- チャネルの作成および使用 : [2.8 項「イベント・アダプタの構成」](#)

1.7.6 SOA Suite のサービス・コンポーネントの使用

Oracle Application Adapter for J.D. Edwards OneWorld は、BPEL、メディエータ、BPM などの SOA Suite のサービス・コンポーネントと統合されています。必要なプロセスを JDeveloper で作成してから、SOA サーバーにデプロイします。

BPEL、メディエータ、BPM サービス・コンポーネントの使用の詳細は、次を参照してください。

- [第 4 章「Oracle SOA Suite での BPEL サービス・コンポーネントの統合」](#)
- [第 5 章「Oracle SOA Suite でのメディエータ・サービス・コンポーネントとの統合」](#)
- [第 6 章「Oracle SOA Suite での BPM サービス・コンポーネントの統合」](#)

1.7.7 Oracle Service Bus の使用

Oracle Application Adapter for J.D. Edwards OneWorld は、Oracle Service Bus (OSB) と統合されており、Web サービスの簡易な統合を可能にします。必要なプロセスを Oracle Service Bus Console で作成します。また、プロセスを JDeveloper で作成してから、SOA サーバーにデプロイすることもできます。

OSB Console の使用の詳細は、[第 7 章「sbconsole を使用した Oracle Service Bus 用のアウトバウンドおよびインバウンド・プロセスの構成」](#)を参照してください。

OSB JDeveloper の使用の詳細は、[第 8 章「JDeveloper を使用した Oracle Service Bus 用のアウトバウンドおよびインバウンド・プロセスの構成」](#)を参照してください。

1.7.8 その他の機能

以下に、その他の機能、およびそれらの機能に関連するこのユーザー・ガイドの項を示します。

- [例外フィルタの構成 : 9.4 項「例外フィルタ」](#)
- [資格証明マッピングの構成 :](#)
 - [9.5 項「Oracle SOA Suite \(BPEL、メディエータまたは BPM\) の資格証明マッピング」](#)
 - [9.6 項「Oracle Service Bus \(OSB\) の資格証明マッピング」](#)

Oracle Application Adapter for J.D. Edwards OneWorld の構成

この章では、Oracle Adapter アプリケーション・エクスプローラ (アプリケーション・エクスプローラ) を使用して、J.D. Edwards OneWorld システムに接続するターゲットを定義し、システム・オブジェクトを表示し、XML スキーマおよび Web サービスを作成する方法について説明します。また、イベント・アダプタの構成方法についても説明します。

この章の内容は以下のとおりです。

- 2.1 項「アプリケーション・エクスプローラの起動」
- 2.2 項「リポジトリ設定の構成」
- 2.3 項「リポジトリ構成の作成」
- 2.4 項「J.D. Edwards OneWorld 用の接続 (ターゲット) の確立」
- 2.5 項「XML スキーマの作成」
- 2.6 項「WSDL の生成 (J2CA 構成のみ)」
- 2.7 項「Web サービスの作成およびテスト (BSE 構成のみ)」
- 2.8 項「イベント・アダプタの構成」
- 2.9 項「ランタイムの概要」
- 2.10 項「アウトバウンドおよびインバウンド処理用 JDE.INI ファイルの変更」

2.1 アプリケーション・エクスプローラの起動

アプリケーション・エクスプローラを起動するには、次のステップを実行します。

1. アプリケーション・エクスプローラがデプロイされている Oracle WebLogic Server が起動していることを確認します。
2. コマンド・プロンプトを開きます。
3. 次のディレクトリに移動します。
`<ADAPTER_HOME>\user_projects\domains\base_domain\bin`
4. `setDomainEnv.cmd` (Windows) または `../setDomainEnv.sh` (UNIX、Linux) を実行します。

このコマンドは、Oracle WebLogic Server 環境でアプリケーション・エクスプローラ用のクラス・パスおよびその他の環境変数を設定します。また、これにより、アプリケーション・エクスプローラは、Oracle WebLogic Server API にアクセスして、WSDL ファイルを Oracle Service Bus (OSB) Console に公開できます。

5. コマンド・プロンプトを開けたままにします。
6. 次のディレクトリに移動します。

```
<ADAPTER_HOME>\tools\iwae\bin
```
7. `ae.bat` (Windows) または `iwae.sh` (UNIX、Linux) を実行してアプリケーション・エクスプローラを起動します。

アプリケーション・エクスプローラが起動します。これで、J.D. Edwards OneWorld システムに新規ターゲットを定義する準備ができました。

注意: `iwae.sh` ファイルを UNIX または Linux プラットフォームで実行する前に、権限を変更する必要があります。たとえば、次のようになります。

```
chmod +x iwae.sh
```

2.2 リポジトリ設定の構成

リポジトリには、構成の詳細、アダプタのターゲット、チャンネルに関する情報、およびその他の構成情報が格納されます。BSE および J2CA リポジトリ設定を構成する方法の詳細は、Oracle Fusion Middleware Oracle WebLogic Server Application Adapters インストール・ガイド (データベース・リポジトリの構成に関する項) を参照してください。

2.3 リポジトリ構成の作成

アプリケーション・エクスプローラを Oracle Application Adapter for J.D. Edwards OneWorld とともに使用する前に、リポジトリ構成を作成する必要があります。アダプタのデプロイ先のコンテナに基づき、Web サービスと J2CA の 2 種類のリポジトリ構成を作成できます。デザインタイム中に、リポジトリは、アプリケーション・エクスプローラを使用したアダプタ接続の構成、EIS オブジェクトの参照、サービスの構成、および EIS イベントのリスニングのためのリスナー構成の際、作成されたメタデータの格納に使用されます。リポジトリ内の情報は実行時でも参照されます。

この項では、次のトピックについて説明します。

- [2.3.1 項「BSE 用の構成の作成」](#)
- [2.3.2 項「J2CA 用の構成の作成」](#)
- [2.3.3 項「BSE または J2CA 構成への接続」](#)

Web サービスと BSE は同じタイプのデプロイメントを参照します。詳細は、1-1 ページの「[アダプタの機能](#)」を参照してください。

2.3.1 BSE 用の構成の作成

アプリケーション・エクスプローラを使用して BSE 用のリポジトリ構成を作成するには、先に新規構成を定義する必要があります。

この項では、次のトピックについて説明します。

- [2.3.1.1 項「BSE 用の新規構成の定義」](#)

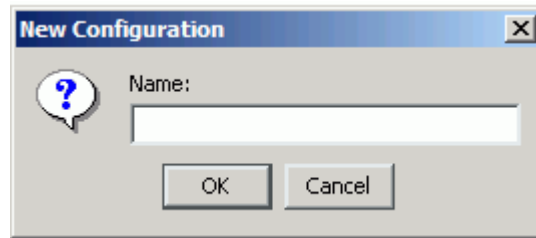
2.3.1.1 BSE 用の新規構成の定義

BSE 用の新規構成を定義するには、次のステップを実行します。

1. 「**構成**」を右クリックし、「**新規**」を選択します。

図 2-1 に示すように、「新規構成」ダイアログが表示されます。

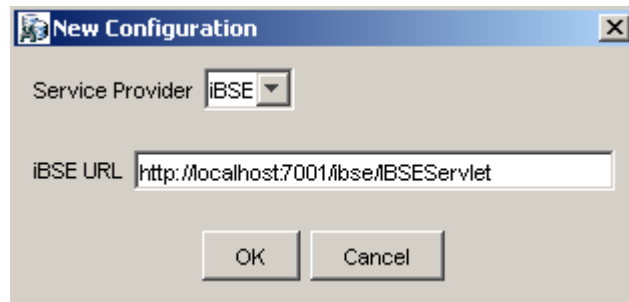
図 2-1 「新規構成」ダイアログ



2. 新規構成の名前を入力し(「myConfig」など)、「OK」をクリックします。

図 2-2 に示すように、「新規構成」ダイアログが表示されます。

図 2-2 「新規構成」ダイアログ



3. 「サービス・プロバイダ」リストから、「iBSE」を選択します。
4. 「iBSE URL」フィールドで、デフォルトの URL を受け入れるか、または次のフォーマットを使用して別の URL と置き換えます。

`http://host name:port/ibse/IBSEServlet`

ここで、*host name* は Oracle WebLogic Server が存在するシステムで、*port* は管理されている Oracle WebLogic Server 用の HTTP ポートです(「soa_server1」など)。

5. 「OK」をクリックします。

図 2-3 に示すように、新規構成を示すノードがルートの「構成」ノードの下に表示されます。

図 2-3 「構成」ノード



2.3.2 J2CA 用の構成の作成

アプリケーション・エクスプローラを使用して J2CA 用の構成を作成するには、先に新規構成を作成する必要があります。

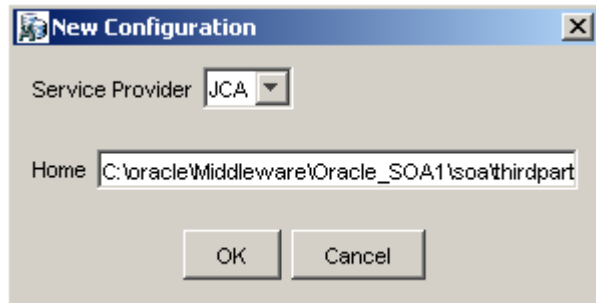
J2CA 用の新規構成を定義するには、次のステップを実行します。

1. 「構成」を右クリックし、「新規」を選択します。

「新規コンフィギュレーション」ダイアログが表示されます。

2. 図 2-4 に示すように、新規構成の名前を入力し (「myConfig」など)、「OK」をクリックします。

図 2-4 「新規構成」ダイアログ



3. 「サービス・プロバイダ」リストから、「JCA」を選択します。
 4. 「OK」をクリックします。
- 図 2-5 に示すように、新規構成を示すノードがルートの「構成」ノードの下に表示されます。

図 2-5 「構成」ノード



Oracle Adapter J2CA 構成フォルダが、アダプタのインストールに基づいて次の場所に格納されます。

```
<ADAPTER_HOME>\config\configuration_name
```

configuration_name は、作成した構成の名前です (「SampleConfig」など)。

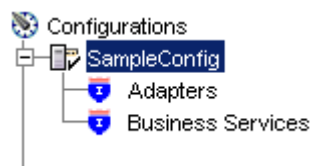
2.3.3 BSE または J2CA 構成への接続

新規構成に接続するには、次のステップを実行します。

1. 「SampleConfig」など、接続する構成を右クリックします。
2. 「接続」を選択します。

「アダプタ」、「イベント」および「ビジネス・サービス」(Web サービスとしても知られる) のノードが表示されます。「ビジネス・サービス」ノードは、BSE 構成の場合にのみ表示されます。J2CA 構成に接続している場合は、「ビジネス・サービス」ノードは表示されません。図 2-6 に、「SampleConfig」という名前の BSE 構成の例を示します。

図 2-6 「構成」ノードの下に表示される新しい「SampleConfig」構成



- 「アダプタ」ノードを使用して、J.D. Edwards OneWorld とのインバウンド相互作用を作成します。たとえば、「アダプタ」ノード内の「J.D. Edwards OneWorld」ノードを使用して、J.D. Edwards OneWorld を更新するサービスを構成します。
- 「イベント」ノード (J2CA 構成の場合のみ) を使用して、J.D. Edwards OneWorld でイベントをリスニングするリスナーを構成します。
- 「ビジネス・サービス」ノード (BSE 構成の場合のみ) を使用して、「アダプタ」ノードで作成された Web サービスをテストします。「ビジネス・サービス」ノードのセキュリティ機能を使用して、Web サービスのセキュリティ設定を制御することもできます。

これで、J.D. Edwards OneWorld に新規ターゲットを定義できます。

2.4 J.D. Edwards OneWorld 用の接続 (ターゲット) の確立

アプリケーション定義の一部には、アダプタ用のターゲットの追加が含まれています。アプリケーション・エクスプローラでのターゲットの設定には、ターゲットに固有の情報が必要です。

この項では、次のトピックについて説明します。

- [2.4.1 項「J.D. Edwards OneWorld に対するターゲットの定義」](#)

利用できるマスター・ビジネス関数 (MBF) を参照するには、使用するシステムに対してターゲットを最初に定義する必要があります。ターゲットを定義すると、その後は自動的に保存されます。アプリケーション・エクスプローラを起動するたび、または切断後に、システムに接続する必要があります。

アプリケーション・エクスプローラを起動すると、インストールされているアダプタに基づいて、アプリケーション・エクスプローラでサポートされているアプリケーション・システムが (ノードとして) 左ペインに表示されます。

2.4.1 J.D. Edwards OneWorld に対するターゲットの定義

この項では、次のトピックについて説明します。

- [2.4.1.1 項「定義済の J.D. Edwards OneWorld ターゲットへの接続」](#)
- [2.4.1.2 項「J.D. Edwards OneWorld からの切断」](#)
- [2.4.1.3 項「ターゲットの編集」](#)
- [2.4.1.4 項「ターゲットの削除」](#)

アプリケーション・システムに初めて接続する場合、新規ターゲットを定義する必要があります。

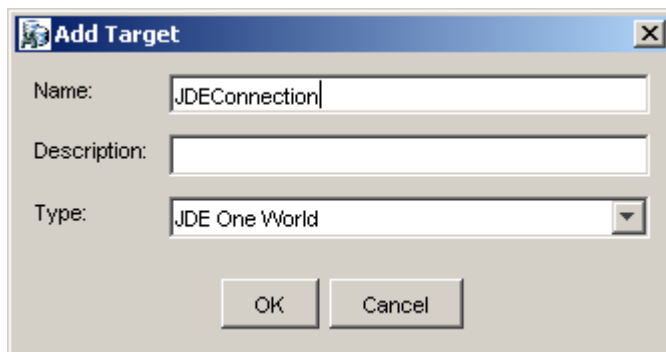
ターゲットを定義したときは、Oracle WebLogic Server を再起動してランタイム用リポジトリを更新する必要があります。

注意： 新規ターゲットを作成する前に、J.D. Edwards OneWorld システムに必要なライブラリ・ファイルを取得し、Oracle Application Adapter for J.D. Edwards OneWorld をデプロイする適切な場所にそれらをコピーする必要があります。詳細は、*Oracle Fusion Middleware Oracle WebLogic Server Application Adapters インストレーション・ガイド*を参照してください。

ターゲットを定義するには、次のステップを実行します。

1. 左ペインで、「アダプタ」ノードを展開します。
インストールされているアダプタに基づいて、アプリケーション・エクスプローラでサポートされているアプリケーション・システムがノードとして表示されません。
2. 「JDEdwards」ノードを右クリックし、「ターゲットの追加」を選択します。
☒ 2-7 に示すように、「ターゲットの追加」ダイアログが表示されます。

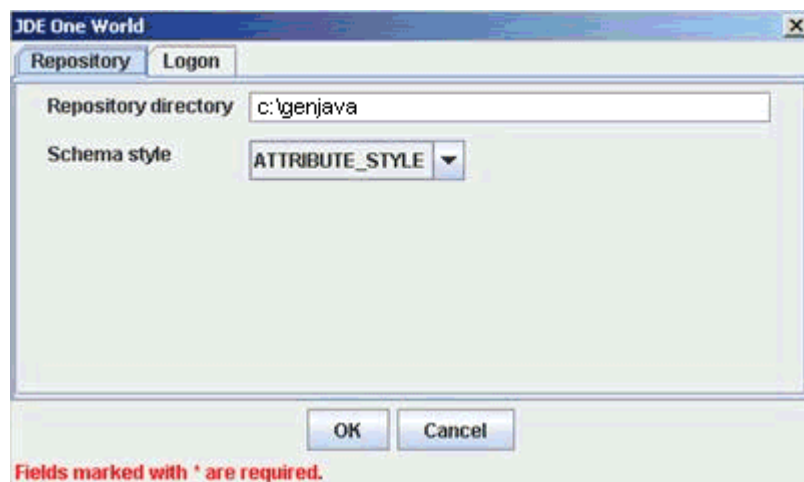
図 2-7 「ターゲットの追加」ダイアログ



次のステップを実行します。

- a. 「名前」フィールドに、「JDEConnection」などのわかりやすい名前を入力します。
 - b. 「説明」フィールドに、ターゲットの説明を入力します (オプション)。
 - c. 「タイプ」リストから、「JDE One World」を選択します。
3. 「OK」をクリックします。
☒ 2-8 に示すように、「JDE One World」ダイアログが表示されます。

図 2-8 「JDE One World」 ダイアログ



- a. 「リポジトリ」タブで、「リポジトリ・ディレクトリ」フィールドに GenJava リポジトリへのパスを入力します。

これは、J.D. Edwards OneWorld ビジネス関数にアクセスするための Java ラッパーの場所で、GenJava 開発ツールによって作成されます。これは前提条件ステップで、アプリケーション・エクスプローラを使用して新規ターゲットを作成する前に実行する必要があることに注意してください。

注意： スキーマを生成するには、GenJava リポジトリが必要です。J.D. Edwards OneWorld マスター・ビジネス関数リポジトリのビルドに関する包括的な詳細は、*J.D. Edwards OneWorld Xe 相互運用ガイド* を参照してください。GenJava プログラムの使用の詳細は、[付録 A 「アウトバウンドおよびインバウンド処理用 J.D. Edwards OneWorld の構成」](#) の [GenJava 開発ツールの使用 \(アウトバウンド処理\)](#) を参照してください。

- b. 「スキーマ・スタイル」リストで、「ELEMENT_STYLE」または「ATTRIBUTE_STYLE」を選択します。
- c. 「ログオン」タブをクリックし、次の表の情報に従ってターゲット・タイプの適切な情報を入力します。図 2-9 に示すように、アスタリスク付きのフィールドは必須です。

図 2-9 「ログオン」タブ

Repository	Logon
User id*	<input type="text"/>
User password*	<input type="text"/>
JDE Environment*	<input type="text"/>
Server IP address*	<input type="text"/>
Server port *	<input type="text"/>
User role	*ALL

パラメータ	説明
ユーザー ID*	J.D. Edwards OneWorld の有効なユーザー ID。
ユーザー・パスワード*	ユーザー ID に関連付けられたパスワード。
JDE 環境 *	J.D. Edwards OneWorld 環境 (例 : DU7333)。このパラメータの詳細は、J.D. Edwards OneWorld のドキュメントを参照するか、J.D. Edwards OneWorld システム管理者にお問い合わせください。
サーバー IP アドレス *	J.D. Edwards OneWorld が実行されているサーバーの名前。JDEOW などのサーバー名または IP アドレスを指定できます。
サーバー・ポート *	6009 など、サーバーがリスニングしているポート番号。
ユーザー・ロール	*ALL と指定します。

4. 「OK」をクリックします。

新規ターゲットの「JDEConnection」が、「JDEdwards」ノードの下に表示されます。

2.4.1.1 定義済の J.D. Edwards OneWorld ターゲットへの接続

ターゲットに接続するには、次のステップを実行します。

1. 「サービス・アダプタ」ノードを展開します。
2. 「JDEdwards」ノードを展開します。
3. JDEdwards ノードの下にターゲット名 (「JDEConnection」 など) をクリックします。
4. 右側の「ログオン」タブをクリックします。
「ログオン」タブに、接続パラメータに入力した値が表示されます。
5. 接続パラメータを確認します。
6. ターゲット名を右クリックし、「接続」を選択します。

図 2-10 に示すように、「x」アイコンが表示されなくなります。これは、ノードが接続状態であることを示します。

図 2-10 JDEdwards のターゲット・ノード



2.4.1.2 J.D. Edwards OneWorld からの切断

ターゲットから切断するには、次のステップを実行します。

1. 「アダプタ」ノードを展開します。
2. 「JDEdwards」ノードを展開します。
3. 接続先のターゲット (「JDEConnection」など) を右クリックし、「切断」を選択します。

「x」アイコンが表示されます。これは、ノードが切断状態であることを示します。

2.4.1.3 ターゲットの編集

ターゲットを編集するには、次のステップを実行します。

1. 左ペインで、編集するターゲットの接続が切断されていることを確認します。
2. ターゲットを右クリックし、「編集」を選択します。
既存の接続パラメータを編集するためのウィンドウが表示されます。
3. ターゲットの情報を変更します。
4. 「OK」をクリックします。

ターゲットを編集したときは、Oracle WebLogic Server を再起動してランタイム用にリポジトリを更新する必要があります。

2.4.1.4 ターゲットの削除

ターゲットを切断して閉じるかわりに削除できます。ターゲットを削除すると、エクスプローラの左ペインで J.D. Edwards OneWorld ターゲットのリストからそのノードが消えます。

接続を削除したときは、Oracle WebLogic Server を再起動してランタイム用にリポジトリを更新する必要があります。

ターゲットを削除するには、次のステップを実行します。

1. 「アダプタ」ノードを展開します。
2. 「JDEdwards」ノードを展開します。
3. 接続先のターゲット (「JDEConnection」など) を右クリックし、「削除」を選択します。

使用可能な接続のリストからそのノードが消えます。

アプリケーション・システム・オブジェクトの表示方法の詳細は、*J.D. Edwards OneWorld XE 相互運用ガイド* リリースを参照してください。

2.5 XMLスキーマの作成

MBF を実行するには、アダプタが J.D. Edwards OneWorld ThinNet API を介してリクエスト・ドキュメントを受信する必要があります。エージェントにより、リクエストが処理され、結果を示す XML レスポンス・ドキュメントが送信されます。アプリケーション・エクスプローラで、XML リクエスト・スキーマおよび XML レスポンス・スキーマの両方が作成されます。

この項では、次のトピックについて説明します。

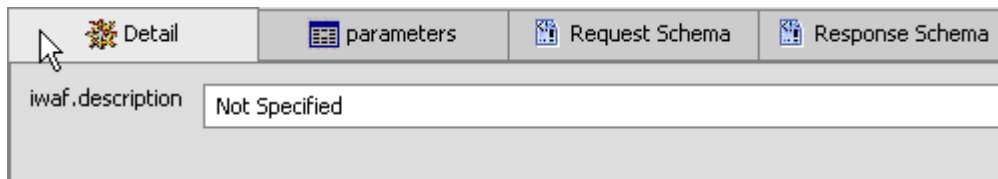
- 2.5.1 項「リクエスト・スキーマおよびレスポンス・スキーマの作成」

2.5.1 リクエスト・スキーマおよびレスポンス・スキーマの作成

次の手順では、J.D. Edwards OneWorld ビジネス関数用のリクエスト・スキーマおよびレスポンス・スキーマの作成方法について説明します。アプリケーション・エクスプローラを使用すると、この関数用の XML スキーマを作成できます。

1. 2-8 ページの「定義済の J.D. Edwards OneWorld ターゲットへの接続」の説明に従って、J.D. Edwards OneWorld ターゲットに接続します。
2. 「サービス」ノードを展開します。
3. スキーマを作成する MBF のノードを展開します。
4. 図 2-11 に示すように、MBF の下にあるノードを展開してから選択します。

図 2-11 「詳細」タブ



5. 図 2-12 に示すように、「パラメータ」タブをクリックして、パラメータ情報を表示します。

図 2-12 「パラメータ」タブ

Field	Type	MaxLength
szLedgerType	String	3
szUnitsLedg...	String	3
cRetainedEa...	Char	1
cLedgerReq...	Char	1
cIntercompa...	Char	1
cRestateme...	Char	1
szCurrency...	String	4
cDirectBalan...	Char	1

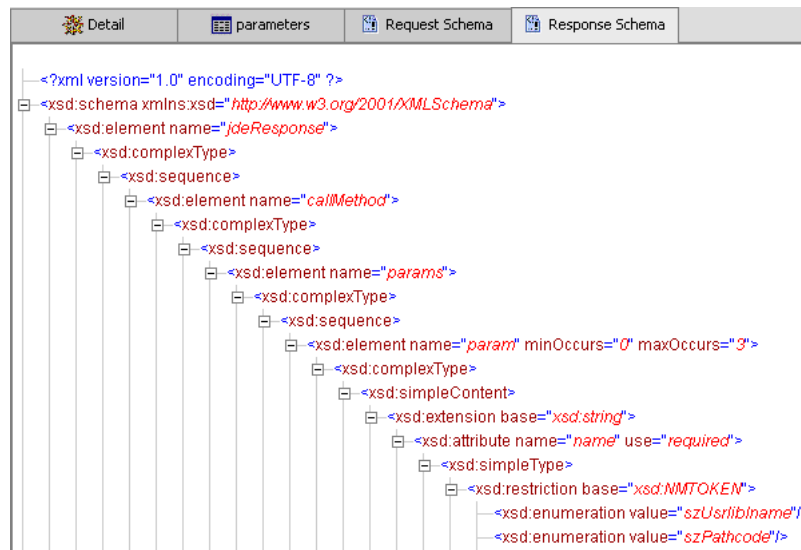
6. 図 2-13 に示すように、「リクエスト・スキーマ」をクリックして、リクエスト・スキーマ情報を表示します。

図 2-13 「リクエスト・スキーマ」タブ



7. 図 2-14 に示すように、「レスポンス・スキーマ」をクリックして、レスポンス・スキーマ情報を表示します。

図 2-14 「レスポンス・スキーマ」タブ



2.6 WSDL の生成 (J2CA 構成のみ)

リクエスト/レスポンス (アウトバウンド) サービスの WSDL (Web Service Definition Language) を生成する手順は、アダプタのイベント通知 (インバウンド) J2CA サービスの WSDL を生成する手順とは異なります。

この項では、次のトピックについて説明します。

- 2.6.1 項「アウトバウンド相互作用のための WSDL の生成」

2.6.1 アウトバウンド相互作用のための WSDL の生成

リクエスト/レスポンス型サービス用の WSDL ファイルを生成するには、次のステップを実行します。

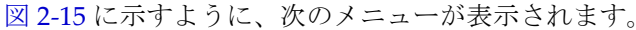
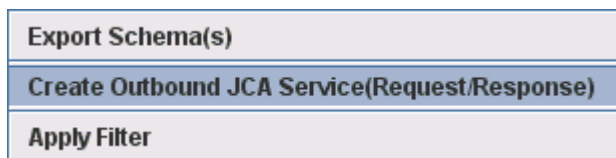
1. 「アプリケーション・エクスプローラ」を起動し、定義済の J.D. Edwards OneWorld ターゲットに接続します。
2. 「サービス」、「CALLBSFN」、「アドレス帳」の順に展開します。「GetEffectiveAddress」を選択します。
3. 「GetEffectiveAddress」を右クリックします。

 図 2-15 に示すように、次のメニューが表示されます。

図 2-15 アウトバウンド JCA サービスの作成 (リクエスト/レスポンス) オプション



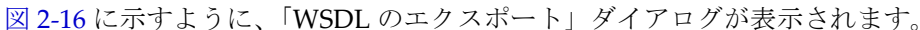
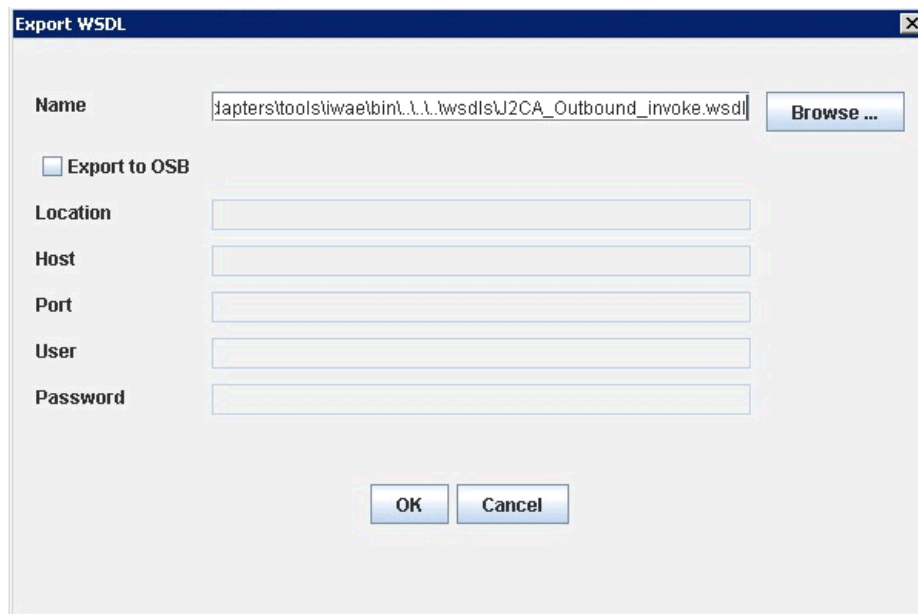
4. アウトバウンド JCA サービスの作成 (リクエスト/レスポンス) を選択します。

 図 2-16 に示すように、「WSDL のエクスポート」ダイアログが表示されます。

図 2-16 「WSDL のエクスポート」ダイアログ



5. ファイルのデフォルト名を受け入れます。
 .wsdl ファイル拡張子が自動的に追加されます。デフォルトでは、リクエスト/レスポンス型のサービス用に生成された WSDL ファイルの名前の末尾には「_invoke」が付き、イベント通知用に生成されたものには「_receive」が付き
 ます。
6. 「OK」をクリックします。
 WSDL ファイルが指定した場所に保存されます。

2.7 Web サービスの作成およびテスト (BSE 構成のみ)

アプリケーション・エクスプローラを使用して、Web サービス ([ビジネス・サービス](#)とも呼ばれます) を生成できます。ビジネス関数リポジトリを検索したり、アダプタで使用する関数用の Web サービスを生成できます。

この項では、次のトピックについて説明します。

- [2.7.1 項「Web サービスの作成」](#)
- [2.7.2 項「Web サービスのテスト」](#)
- [2.7.3 項「伝播の識別」](#)

次の手順では、BusinessUnitExistenceCheck という例を使用します。

注意： J2EE Connector Architecture(J2CA) 実装では、Web サービスは使用できません。J2CA を使用するようにアダプタがデプロイされている場合、Common Client Interface(CCI) により、統合サービスが提供されます。

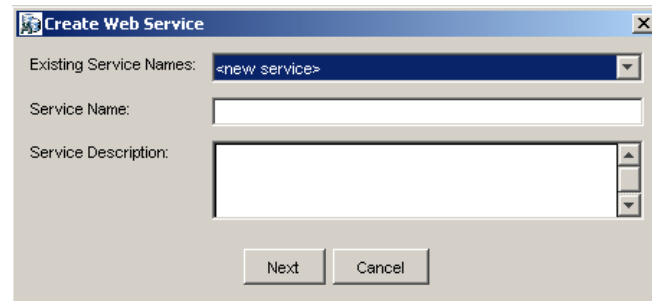
2.7.1 Web サービスの作成

ビジネス関数用の Web サービスを作成するには、次のステップを実行します。

1. 「JDEdwards」ノード、サービスノードの順に展開します。
2. 「CALLBSFN」、「アドレス帳」の順に展開します。
3. 「GetEffectiveAddress」を右クリックし、「Web サービスの作成」を選択します。

[図 2-17](#) に示すように、「Web サービスの作成」ダイアログが表示されます。

図 2-17 「Web サービスの作成」ダイアログ



新規または既存の Web サービスのメソッドとして、ビジネス関数を追加できます。

- a. 「既存のサービス名」リストから、<新規サービス>または既存のサービスを選択します。
 - b. 新規サービスを作成する場合は、「サービス名」フィールドで、サービス名を指定します。この名前により、「ビジネス・サービス」ノードの下のサービスのリストで Web サービスが識別されます。
 - c. サービスに関する説明を入力します (オプション)。
4. 「次へ」をクリックします。
次のステップを実行します。

- a. 「ライセンス名」フィールドで、Web サービスに割り当てるライセンス・コードを1つ以上選択します。
 - b. 「メソッド名」フィールドで、デフォルトのメソッド名のままにします。
 - c. 「説明」フィールドに、メソッドの簡単な説明を入力します (オプション)。
 - d. 「DTD ディレクトリ」フィールドに、Web サービスを保存する場所を指定します。デフォルト以外の場所を選択する場合、「参照」をクリックして、目的の場所へ移動します。
5. 「OK」をクリックします。
- アプリケーション・エクスプローラで、「ビジネス・サービス」ノードに表示が切り替わり、新規の Web サービスが左ペインに表示されます。
6. 新規 Web サービスを右クリックし、メニューから「WSDL の保存」を選択します。「保存」ダイアログが表示されます。
 7. WSDL ファイルの名前、およびファイル・システム上に WSDL ファイルを保存する場所を指定します。
 8. 「保存」をクリックします。

2.7.2 Web サービスのテスト

Web サービスを作成した後は、テストを行い、適切に機能することを確認できます。Web サービスのテスト用のテスト・ツールは装備されています。

Web サービスをテストするには、次のステップを実行します。

1. 「ビジネス・サービス」ノードをクリックして、Web サービスにアクセスします。
2. 「サービス」ノードを展開します。
3. テストするビジネス・サービスの名前を選択します。
ビジネス・サービスの名前は、右ペインにリンクとして表示されます。
4. 右ペインで、指定するビジネス・サービスのリンクをクリックします。
テスト・オプションが右ペインに表示されます。XML 入力を必要とする Web サービスをテストする場合は、入力フィールドが表示されます。
5. 適切な入力項目を入力します。
6. 「起動」をクリックします。
アプリケーション・エクスプローラで結果が表示されます。

2.7.3 伝播の識別

XMLSPY などのサード・パーティの XML エディタを使用して Web サービスをテストまたは実行する場合は、SOAP ヘッダーで指定するユーザー名およびパスワードの値が有効で、J.D. Edwards OneWorld への接続に使用されるものと同じである必要があります。アプリケーション・エクスプローラを使用したターゲットの作成中に J.D. Edwards OneWorld 用に指定したユーザー名およびパスワードの値は、この Web サービスのリクエスト用のもので上書きされます。次に示すのは、Web サービス用の WSDL ファイルに含まれるサンプルの SOAP ヘッダーです。

```
<SOAP-ENV:Header>
  <m:ibsinfo xmlns:m="urn:schemas-iwaysoftware-com:iwse">
    <m:service>String</m:service>
    <m:method>String</m:method>
    <m:license>String</m:license>
```

```

    <m:disposition>String</m:disposition>
    <m:Username>String</m:Username>
    <m>Password>String</m>Password>
    <m:language>String</m:language>
  </m:ibsinfo>
</SOAP-ENV:Header>

```

<m:disposition> および <m:language> タグは必要でないため、SOAP ヘッダーから削除できます。

2.8 イベント・アダプタの構成

データベースまたはアプリケーション・システムのアクティビティによって、イベントが生成されます。イベントを使用して、アプリケーションのアクションをトリガーできます。たとえば、顧客情報への更新を反映してデータベースが更新されるようにすることができます。この時点でアプリケーションを実行する必要がある場合、アプリケーションはこのイベントのコンシューマです。

この項では、次のトピックについて説明します。

- [2.8.1 項「チャンネルの作成および編集」](#)
- [2.8.2 項「J.D. Edwards OneWorld Event Listener」](#)
- [2.8.3 項「J.D. Edwards OneWorld Event Listener の構成」](#)

アプリケーション・システムへの接続を作成した後、アプリケーション・エクスプローラを使用してイベントを追加できます。イベントを作成するには、チャンネルを作成する必要があります。

注意： J2CA 構成を使用している場合は、各オブジェクトに対して新規のチャンネルを作成し、WSDL を正生成するときにこのチャンネルを選択する必要があります。チャンネルの作成が必要なのは J2CA 構成の場合のみです。

チャンネルは、バックエンド・システムの特定のインスタンスへの構成済の接続を表します。詳細は、2-15 ページの「[チャンネルの作成および編集](#)」を参照してください。

2.8.1 チャンネルの作成および編集

この項では、イベントのチャンネルを作成する方法について説明します。次のトピックで構成されています。

- [2.8.1.1 項「HTTP チャンネルの作成」](#)
- [2.8.1.2 項「TCP チャンネルの作成」](#)
- [2.8.1.3 項「ファイル・チャンネルの作成」](#)
- [2.8.1.4 項「チャンネルの編集」](#)
- [2.8.1.5 項「チャンネルの削除」](#)

チャンネルを作成、変更または削除する場合、ランタイム目的のリポジトリの変更および更新を認識するように、Oracle WebLogic Server を再起動する必要があります。チャンネルおよびインバウンド WSDL ファイルが正常に作成されてから、アプリケーション・サーバーを再起動する前にアプリケーション・エクスプローラを閉じます。

注意： Oracle Application Adapter for J.D. Edwards OneWorld を BPM、BPEL、メディエータまたは OSB インバウンド・プロセス・コンポーネントと統合する予定の場合は、チャンネルを起動しないでください。チャンネルは、BPM、BPEL、メディエータまたは OSB プロセス・コンポーネントがデプロイされた後、ランタイム・サーバーによって管理されます。テストまたはデバッグ目的でチャンネルをアプリケーション・エクスプローラから起動する場合は、ランタイム前に停止してください (BPM、BPEL、メディエータまたは OSB プロセス・コンポーネントと連携する場合)。

次の3つのチャンネル・タイプを使用できます。

- HTTP
- TCP
- ファイル

注意： チャンネルは、Oracle Application Adapter for J.D. Edwards OneWorld をインストールするシステムにのみ構成できます。

2.8.1.1 HTTP チャンネルの作成

HTTP チャンネルを作成するには、次のステップを実行します。


1. 「イベント」ノードをクリックします。
2. 「JDEdwards」ノードを展開します。
ポートおよびチャンネル・ノードが左ペインに表示されます。
3. 「チャンネル」を右クリックし、「チャンネルの追加」を選択します。
 図 2-18 に示すように、「チャンネルの追加」ダイアログが表示されます。

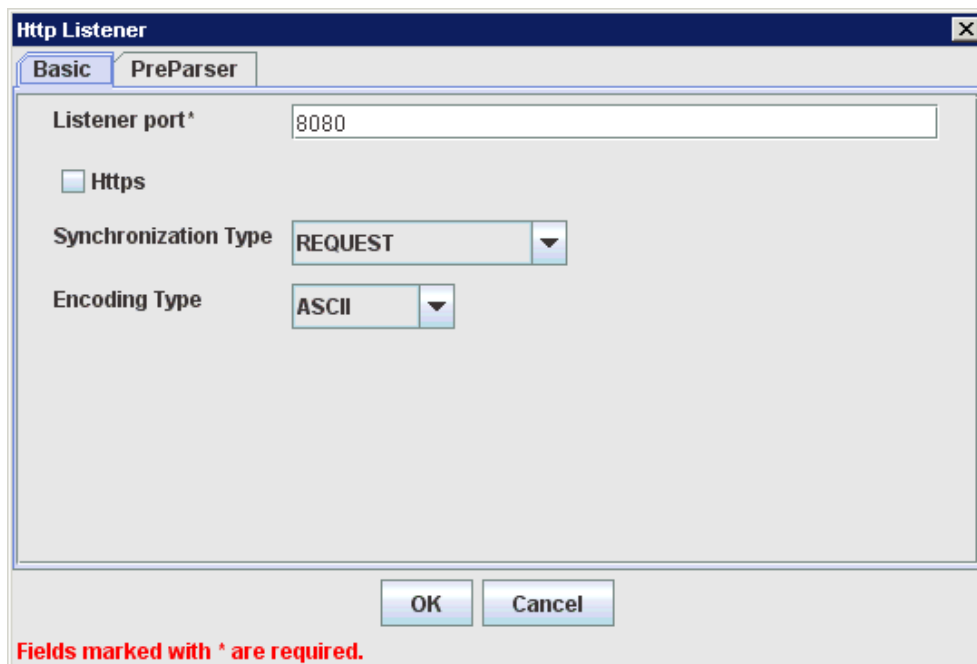
図 2-18 「チャンネルの追加」ダイアログ

The screenshot shows a standard Windows-style dialog box titled "Add Channel". It has a close button in the top right corner. The dialog is divided into several sections: "Name:" with a text input field containing "JDE_Channel1"; "Description:" with a larger empty text area; "Protocol:" with a dropdown menu currently set to "HTTP Listener"; and a port selection section with two empty list boxes labeled "Available Port(s)" and "Selected Port(s)", and four arrow buttons (>>, >, <, <<) between them. At the bottom of the dialog are two buttons: "Next" and "Cancel".

次の情報を入力します。

- a. 「JDE_Channel1」など、チャンネルの名前を入力します。
 - b. 簡単な説明を入力します。
 - c. 「プロトコル」リストから、「HTTP リスナー」を選択します。
4. 「次へ」をクリックします。
- 図 2-19 に示すように、「HTTP リスナー」ダイアログが表示されます。

図 2-19 「HTTP リスナー」ダイアログの「基本」タブ



5. 次の表で指定されているようにシステム情報を入力します。

パラメータ	説明
リスナー・ポート	J.D. Edwards OneWorld イベント・データをリスニングするポート
Https	安全な HTTP 接続を確保するには、「https」チェック・ボックスを選択します。 このオプションは現在サポートされていません。
同期タイプ	次のいずれかの同期オプションを選択します。 <ul style="list-style-type: none"> ■ REQUEST_RESPONSE ■ REQUEST_ACK 重要: 同期タイプが「REQUEST」に設定されている場合、J.D. Edwards OneWorld チャンネルは機能しません。
エンコーディング・タイプ	リストから使用するエンコーディング・タイプを選択します。デフォルトでは、「ASCII」が選択されています。

6. 図 2-20 に示すように、事前パーサー・タブをクリックします。

図 2-20 「HTTP リスナー」ダイアログの事前パーサー・タブ

The screenshot shows the 'Http Listener' dialog box with the 'PreParser' tab selected. The fields are as follows:

- User id* (required)
- User password* (required)
- JDE Environment* (required)
- Application
- Server IP address* (required)
- Server port ^
- User role
- Schema Location
- Schema style: ELEMENT_STYLE (dropdown menu)

Buttons: OK, Cancel

Fields marked with * are required.

7. 次の表で指定されているようにシステム情報を入力します。

パラメータ	説明
ユーザー ID	J.D. Edwards OneWorld の有効なユーザー ID。
ユーザー・パスワード	J.D. Edwards OneWorld ユーザー ID に関連付けられたパスワード。
JDE 環境	J.D. Edwards OneWorld 環境 (例: DU7333)。このパラメータの詳細は、J.D. Edwards OneWorld のドキュメントを参照するか、J.D. Edwards OneWorld システム管理者にお問い合わせください。
アプリケーション	J.D. Edwards OneWorld 環境に定義されたアプリケーション。
サーバー IP アドレス	J.D. Edwards OneWorld が実行されているサーバーの名前。JDEOW などのサーバー名または IP アドレスを指定できます。
サーバー・ポート	6009 など、サーバーがリスニングしているポート番号。
ユーザー・ロール	*ALL と指定します。
スキーマの場所	イベント出力から生成された XML スキーマ (.xsd ファイル) の場所。たとえば、次のようになります。 <code><ADAPTER_HOME>\config\configuration_name\schemas\JDEdwards\target_name\jde-schema.xsd</code>
スキーマ・スタイル	次のオプションから 1 つ選択します。 <ul style="list-style-type: none"> ■ ELEMENT_STYLE(デフォルト) ■ ATTRIBUTE_STYLE <p>詳細は、4-35 ページの 4.5.1 項「イベント統合のための WSDL の生成」を参照してください。</p>

8. 「OK」をクリックします。

チャンネルの説明、ステータスおよび使用可能なポートが記載されているサマリー・ページが表示されます。これらの情報はすべて作成したチャンネルと関連付けられています。

チャンネルは、左ペインのチャンネル・ノードの下に表示されます。

アイコン上の「X」は、そのチャンネルが現在切断されていることを示します。イベント構成をアクティブ化するにはチャンネルを再起動する必要があります。

9. チャンネルを右クリックし、「開始」を選択します。

作成したチャンネルがアクティブになります。左ペインのアイコン上の「X」が消えます。

10. チャンネルを停止するには、チャンネルを右クリックし、「停止」を選択します。

2.8.1.2 TCP チャンネルの作成

TCP チャンネルを作成するには、次のステップを実行します。

1. 「イベント」ノードをクリックします。

2. 「JDEdwards」ノードを展開します。

ポートおよびチャンネル・ノードが左ペインに表示されます。

3. 「チャンネル」を右クリックし、「チャンネルの追加」を選択します。

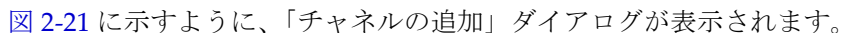
 図 2-21 に示すように、「チャンネルの追加」ダイアログが表示されます。

図 2-21 「チャンネルの追加」ダイアログ

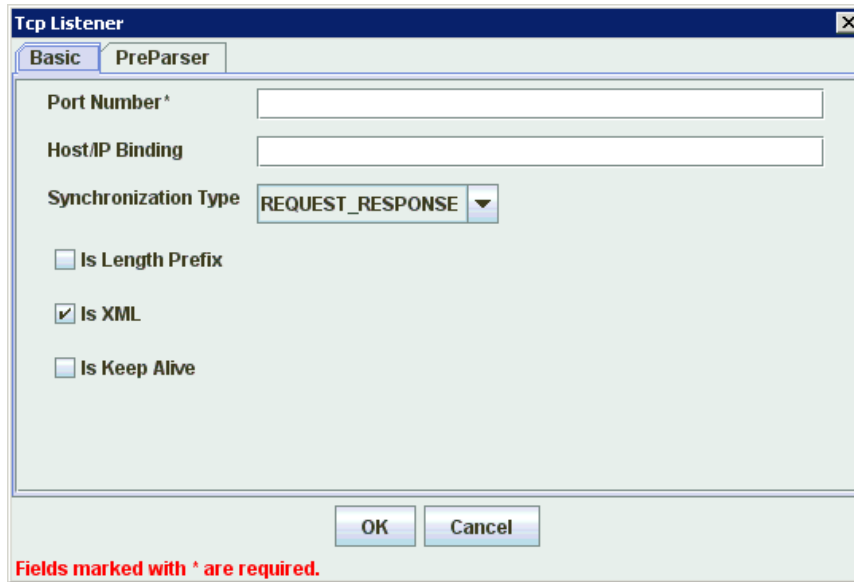
The screenshot shows a Windows-style dialog box titled "Add Channel". It has a close button in the top right corner. The dialog is divided into several sections:

- Name:** A text input field containing "JDE_Channel2".
- Description:** A larger empty text area.
- Protocol:** A dropdown menu currently showing "TCP Listener".
- Available Port(s):** An empty list box on the left.
- Selected Port(s):** An empty list box on the right.
- Between the two list boxes are four arrow buttons: ">>", ">", "<", and "<<".
- At the bottom are two buttons: "Next" and "Cancel".

次の情報を入力します。

- a. 「JDE_Channel2」など、チャンネルの名前を入力します。
 - b. 簡単な説明を入力します。
 - c. 「プロトコル」リストから、「TCP リスナー」を選択します。
4. 「次へ」をクリックします。
- 図 2-22 に示すように、「TCP リスナー」ダイアログが表示されます。

図 2-22 「TCP リスナー」ダイアログの「基本」タブ



5. 次の表で指定されているようにシステム情報を入力します。

パラメータ	説明
ポート番号	ホスト・データベースがリスニングしているポート。
ホスト /IP バインディング	データベースが常駐するシステムの名前または URL。
同期タイプ	次のいずれかの同期オプションを選択します。 <ul style="list-style-type: none"> ■ REQUEST_RESPONSE ■ REQUEST_ACK 重要: 同期タイプが「REQUEST」に設定されている場合、J.D. Edwards OneWorld チャンネルは機能しません。
接頭辞の長さ	XML フォーマットではないデータを戻す J.D. Edwards OneWorld イベントの場合。TCP/IP イベント・アプリケーションでは、TCP/IP ポートにデータを書き込むとき、データに 4 バイトのバイナリ長フィールドの接頭辞を付ける必要があります。
XML	XML フォーマットでデータを戻す J.D. Edwards OneWorld イベントの場合。事前パーサーは必要ありません。
キープ・アライブ	イベント・トランザクションとチャンネル間の連続通信を維持します。

6. 図 2-23 に示すように、事前パーサー・タブをクリックします。

図 2-23 「TCP リスナー」ダイアログの事前パーサー・タブ

7. 次の表で指定されているようにシステム情報を入力します。

パラメータ	説明
ユーザー ID	J.D. Edwards OneWorld の有効なユーザー ID。
ユーザー・パスワード	J.D. Edwards OneWorld ユーザー ID に関連付けられたパスワード。
JDE 環境	J.D. Edwards OneWorld 環境 (例: DU7333)。このパラメータの詳細は、J.D. Edwards OneWorld のドキュメントを参照するか、J.D. Edwards OneWorld システム管理者にお問い合わせください。
アプリケーション	J.D. Edwards OneWorld 環境に定義されたアプリケーション。
サーバー IP アドレス	J.D. Edwards OneWorld が実行されているサーバーの名前。JDEOW などのサーバー名または IP アドレスを指定できます。
サーバー・ポート	6009 など、サーバーがリスニングしているポート番号。
ユーザー・ロール	*ALL と指定します。
スキーマの場所	イベント出力から生成された XML スキーマ (.xsd ファイル) の場所。たとえば、次のようになります。 <code><ADAPTER_HOME>\config\configuration_name\schemas\JDEdwards\target_name\jde-schema.xsd</code> <p>詳細は、4-35 ページの 4.5.1 項「イベント統合のための WSDL の生成」を参照してください。</p>
スキーマ・スタイル	次のオプションから 1 つ選択します。 <ul style="list-style-type: none"> ■ ELEMENT_STYLE(デフォルト) ■ ATTRIBUTE_STYLE

8. 「OK」をクリックします。

チャンネルの説明、ステータスおよび使用可能なポートが記載されているサマリー・ページが表示されます。これらの情報はすべて作成したチャンネルと関連付けられています。

チャンネルは、左ペインのチャンネル・ノードの下に表示されます。

アイコン上の「X」は、そのチャンネルが現在切断されていることを示します。イベント構成をアクティブ化するにはチャンネルを再起動する必要があります。

9. チャンネルを右クリックし、「開始」を選択します。

作成したチャンネルがアクティブになります。左ペインのアイコン上の「X」が消えます。

10. チャンネルを停止するには、チャンネルを右クリックし、「停止」を選択します。

2.8.1.3 ファイル・チャンネルの作成

ファイル・チャンネルを作成するには、次のステップを実行します。

1. 「イベント」ノードをクリックします。

2. 「JDEdwards」ノードを展開します。

ポートおよびチャンネル・ノードが左ペインに表示されます。

3. 「チャンネル」を右クリックし、「チャンネルの追加」を選択します。

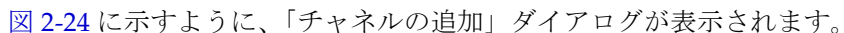
 図 2-24 に示すように、「チャンネルの追加」ダイアログが表示されます。

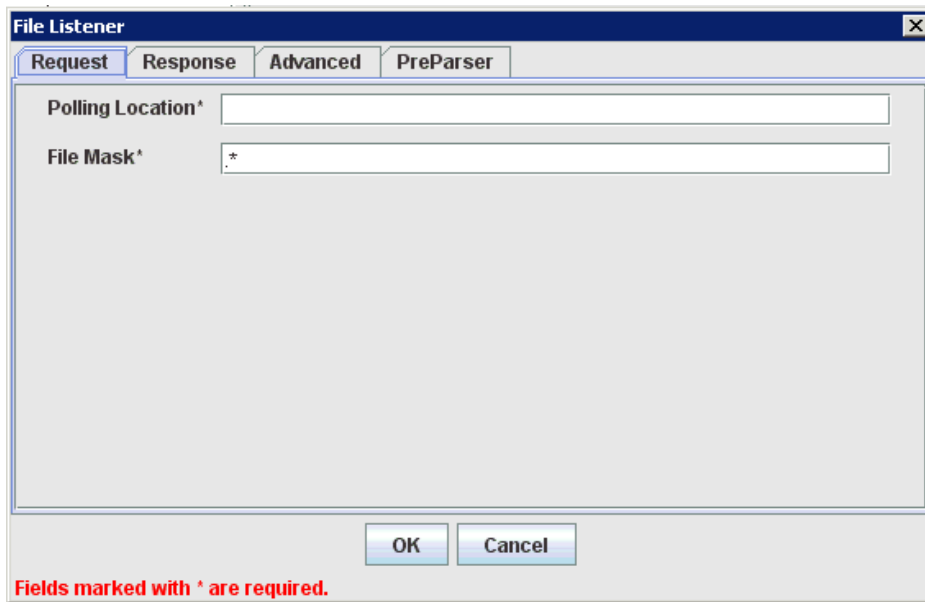
図 2-24 「チャンネルの追加」ダイアログ

The screenshot shows a standard Windows-style dialog box titled "Add Channel". It has a close button in the top right corner. The dialog is divided into several sections: "Name:" with a text input field containing "JDE_Channel3"; "Description:" with a larger empty text area; "Protocol:" with a dropdown menu currently set to "File Listener"; and a port selection section at the bottom. This section consists of two empty list boxes, "Available Port(s)" on the left and "Selected Port(s)" on the right, with four arrow buttons (>>, >, <, <<) positioned between them. At the very bottom of the dialog are two buttons: "Next" and "Cancel".

次の情報を入力します。

- a. 「JDE_Channel3」など、チャンネルの名前を入力します。
 - b. 簡単な説明を入力します。
 - c. 「プロトコル」リストから、ファイル・リスナーを選択します。
4. 「次へ」をクリックします。
- 図 2-25 に示すように、ファイル・リスナー・ダイアログが表示されます。

図 2-25 ファイル・リスナー・ダイアログ

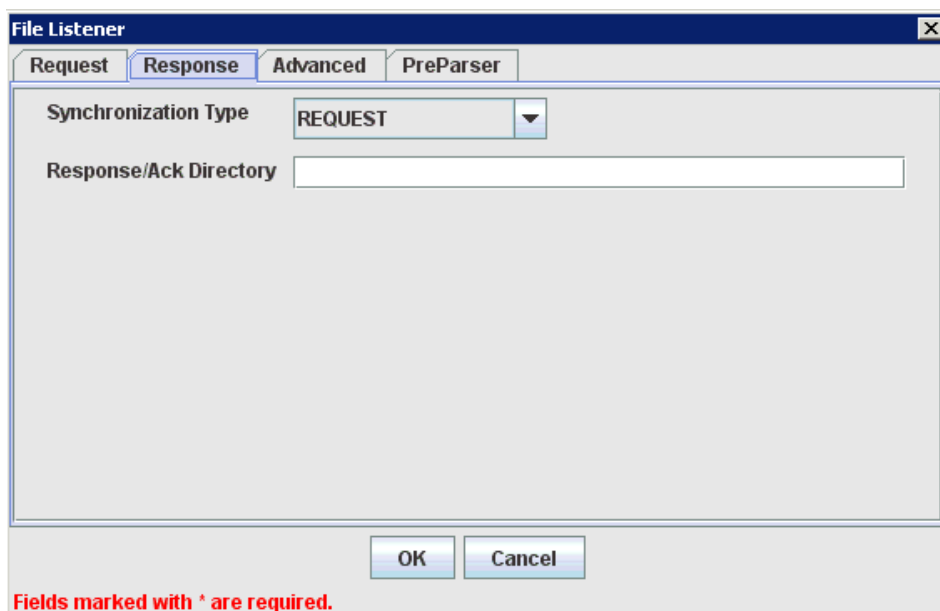


5. 次の表で指定されているように、「リクエスト」タブにシステム情報を入力します。

パラメータ	説明
ポーリング場所	J.D. Edwards OneWorld XML ファイルのターゲット・ファイル・システムの場所。
ファイル・マスク	操作で生成される出力ファイルに使用されるファイル名。

6. 図 2-26 に示すように、「レスポンス」タブをクリックします。

図 2-26 ファイル・リスナー・ダイアログの「レスポンス」タブ

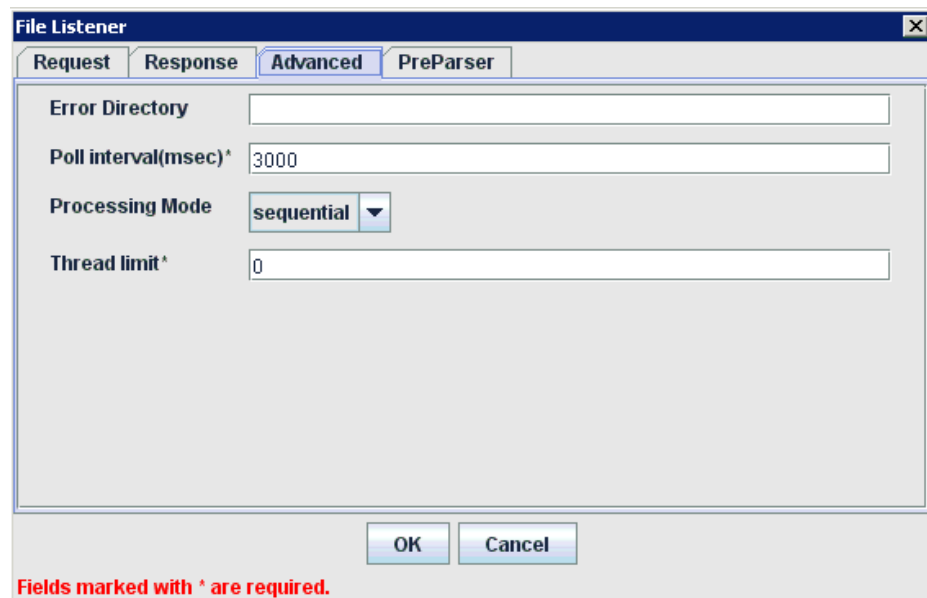


7. 次の表で指定されているように、「レスポンス」タブにシステム情報を入力します。

パラメータ	説明
同期タイプ	次のいずれかの同期オプションを選択します。 <ul style="list-style-type: none"> REQUEST_RESPONSE REQUEST_ACK <p>重要: 同期タイプが「REQUEST」に設定されている場合、J.D. Edwards OneWorld チャンネルは機能しません。</p>
レスポンス /Ack ディレクトリ	J.D. Edwards OneWorld XML ファイルのターゲット・ファイル・システム の場所。

8. [図 2-27](#) に示すように、「詳細」タブをクリックします。

図 2-27 ファイル・リスナー・ダイアログの「詳細」タブ



9. 次の表で指定されているように、「詳細」タブにシステム情報を入力します。

パラメータ	説明
エラー・ディレクトリ	エラーのあるドキュメントが書き込まれるディレクトリ
ポーリング間隔	新規入力をチェックする間隔 (ミリ秒)。デフォルトは3秒です。 オプション。
処理モード	「順次」は、リクエストを1つずつ処理することを示します。 「スレッド」は、複数のリクエストを同時に処理することを示します。
スレッド制限	スレッド処理を選択した場合、同時に処理可能なリクエストの最大数を示します。

10. [図 2-28](#) に示すように、事前パーサー・タブをクリックします。

図 2-28 ファイル・リスナー・ダイアログの事前パーサー・タブ

11. 次の表で指定されているようにシステム情報を入力します。

パラメータ	説明
ユーザー ID	J.D. Edwards OneWorld の有効なユーザー ID。
ユーザー・パスワード	J.D. Edwards OneWorld ユーザー ID に関連付けられたパスワード。
JDE 環境	J.D. Edwards OneWorld 環境 (例: DU7333)。このパラメータの詳細は、J.D. Edwards OneWorld のドキュメントを参照するか、J.D. Edwards OneWorld システム管理者にお問い合わせください。
アプリケーション	J.D. Edwards OneWorld 環境に定義されたアプリケーション。
サーバー IP アドレス	J.D. Edwards OneWorld が実行されているサーバーの名前。JDEOW などのサーバー名または IP アドレスを指定できます。
サーバー・ポート	6009 など、サーバーがリスニングしているポート番号。
ユーザー・ロール	*ALL と指定します。
スキーマの場所	イベント出力から生成された XML スキーマ (.xsd ファイル) の場所。たとえば、次のようになります。 <code><ADAPTER_HOME>\config\configuration_name\schemas\JDEdwards\target_name\jde-schema.xsd</code>
スキーマ・スタイル	次のオプションから 1 つ選択します。 <ul style="list-style-type: none"> ■ ELEMENT_STYLE (デフォルト) ■ ATTRIBUTE_STYLE

12. 「OK」をクリックします。

チャンネルの説明、ステータスおよび使用可能なポートが記載されているサマリー・ページが表示されます。これらの情報はすべて作成したチャンネルと関連付けられています。

チャンネルは、左ペインのチャンネル・ノードの下に表示されます。

アイコン上の「X」は、そのチャンネルが現在切断されていることを示します。イベント構成をアクティブ化するにはチャンネルを再起動する必要があります。

13. チャンネルを右クリックし、「開始」を選択します。

作成したチャンネルがアクティブになります。左ペインのアイコン上の「X」が消えます。

14. チャンネルを停止するには、チャンネルを右クリックし、「停止」を選択します。**2.8.1.4 チャンネルの編集**

チャンネルを編集するには、次のステップを実行します。

1. 左ペインで、編集するチャンネルを見つけます。**2. チャンネルを右クリックし、「編集」を選択します。**

チャンネルの編集ペインが表示されます。

3. チャンネル構成に必要な変更を行い、「終了」をクリックします。**2.8.1.5 チャンネルの削除**

チャンネルを削除するには、次のステップを実行します。

1. 左ペインで、削除するチャンネルを見つけます。**2. チャンネルを右クリックし、「削除」を選択します。**

確認ダイアログが表示されます。

3. 選択したチャンネルを削除するには、「OK」をクリックします。

チャンネルが左ペインのリストから消えます。

2.8.2 J.D. Edwards OneWorld Event Listener

Oracle Application Adapter for J.D. Edwards OneWorld Event Listener は、ビジネス・イベントへの J.D. Edwards OneWorld 承認アクセスの提供専用設計されています。J.D. Edwards OneWorld Event Listener は、J.D. Edwards OneWorld ビジネス関数とともに実行し、J.D. Edwards OneWorld アプリケーション・システムによってコールする専用アプリケーションです。

J.D. Edwards OneWorld アプリケーション・システムでは、目的のイベントのみのイベント情報を取得するために必要な情報を Event Listener に提供します。J.D. Edwards OneWorld 環境の構成の詳細は、次のガイドを参照してください。『*J.D. Edwards OneWorld 相互運用ガイド*』

J.D. Edwards OneWorld Event Listener は、J.D. Edwards OneWorld アプリケーションから直接コールされ、Z ファイル・レコード識別子が渡されます。次に、この識別子により、処理用サーバーに渡されるリクエスト・ドキュメントが生成されます。サーバーは、J.D. Edwards OneWorld システムからイベント情報を受信し、他のアプリケーション・システムとの統合用の情報を伝播します。

2.8.3 J.D. Edwards OneWorld Event Listener の構成

J.D. Edwards OneWorld Event Listener は、基本インストールの一部としてインストールされます。J.D. Edwards OneWorld Adapter は、適切なディレクトリに自動的にインストールされます。統合サーバーが J.D. Edwards OneWorld アプリケーション・サーバーと同じコンピュータにインストールされていない場合は、J.D. Edwards OneWorld Event Listener を構成する必要があります。

J.D. Edwards OneWorld Event Listener は、J.D. Edwards OneWorld 環境で構成されたとおりに、専用トランザクション用に J.D. Edwards OneWorld により起動されます。

J.D. Edwards OneWorld Event Listener には、次のコンポーネントがあります。

- リスナー・イベント・スタブ (IWOEvent.dll)。これは、\etc\jde ディレクトリにあります。たとえば、次のようになります。

```
<ADAPTER_HOME>\etc\jde\iwoevent.dll
```

ファイル拡張子は、オペレーティング・システムによって異なります。

- **Windows** の場合、イベント・スタブは iwoevent.dll になります。
- **Sun Solaris** の場合、イベント・スタブは libiwoevent.so になります。
- **HP-UX** の場合、イベント・スタブは libiwoevent.sl になります。
- **AS/400** の場合、イベント・スタブは iwaysav.sav になります。
- **IBM AIX** の場合、イベント・スタブは libiwoevent.so になります。
- リスナー構成ファイル (iwoevent.cfg)。これは、ユーザーが作成する必要があります。

J.D. Edwards OneWorld Event のリスナー終了は、インバウンド Oracle Application Adapter for J.D. Edwards OneWorld による処理用統合サーバーに、J.D. Edwards OneWorld アウトバウンド・トランザクション表内のレコードのキー・フィールドを渡す関数です。J.D. Edwards OneWorld Event Listener は、J.D. Edwards OneWorld Enterprise Server の下にデプロイされます。J.D. Edwards OneWorld Event Listener 用 Java クラスは IWOEvent と呼ばれ (ファイル拡張子はオペレーティング・システムによって異なります)、大文字/小文字が区別されます。

1. JDE Enterprise Sever の JDE 構造の下に、Outbound というフォルダを作成します。たとえば、次のようになります。

```
\\JDEdwards\E812\DDP\Outbound
```

2. iwoevent.dll ファイルを、新しい Outbound フォルダにコピーします。
3. 環境変数 IWOEVENT_HOME を作成し、iwoevent.dll ファイルを含むディレクトリを指すようにします。
 - Windows の場合: IWOEVENT_HOME をシステム環境変数に追加します。
 - UNIX の場合: 次のコマンドを起動スクリプトに追加します。

```
export IWOEVENT_HOME =/directory_name
```

4. J.D. Edwards OneWorld Server で、定義済ディレクトリ IWOEVENT_HOME に、iwoevent.cfg ファイルを作成します。

J.D. Edwards OneWorld Event Listener では、関連付けられたアダプタでイベントを正確に開始するための接続情報が必要です。この情報は、iwoevent.cfg ファイルに含まれています。このファイルを作成し、接続情報をそれに追加する必要があります。J.D. Edwards OneWorld Event Listener では、関連付けられた統合サーバーが正確に機能するための接続情報が必要です。この情報は、

`iwoevent.cfg` ファイルに含まれています。`iwoevent.cfg` ファイルには、次の3つの異なるセクションがあります。

■ 共通

構成ファイルの共通セクションには、基本的な構成オプションが含まれています。現在は、トレース・オプションのみがサポートされています。

トレース・オプションを選択するには、「**on**」または「**off**」を選択します。

```
common.trace=on|off
```

「on」はトレースをオンに設定し、「off」はトレースをオフに設定します。「off」がデフォルト値です。

■ 別名

構成ファイルの別名セクションには、トランザクションを特定のサーバーに送信するために必要な接続情報が含まれています。現在、Oracle Application Adapter for J.D. Edwards OneWorld では、構成ファイル内に 100 個のエントリ (別名) をサポートしています。

これらのエントリに対する別名値は、次のとおりです。

```
Alias.aliasname={ipaddress|dsn}:port, trace={on|off}
```

各要素の説明は次のとおりです。

`aliasname` は、接続に指定された記号名です。

`ipaddress|dsn` は、Oracle Application Adapter for J.D. Edwards OneWorld を含むサーバーの IP アドレスまたは DSN 名です (必須)。

`port` は、TCP チャネル構成内で Oracle Application Adapter for J.D. Edwards OneWorld 用に定義されたポートです (必須)。

`trace={on|off}` は、特定の別名のトレースをオンに設定します。

■ トランザクション

構成ファイルのトランザクション・セクションには、J.D. Edwards OneWorld トランザクションを指定のサーバーにルーティングするために必要なトランザクション情報が含まれています。

特定の J.D. Edwards OneWorld トランザクションが別名に定義されていない場合、すべての別名に送信されます。これらのエントリに対するトランザクション値は、次のとおりです。

```
trans.jdeTransactionName=alias1,alias2,aliasn
```

`jdeTransactionName` はアウトバウンド・トランザクション用の JDE 定義の名前、`alias1,alias2,aliasn` はトランザクションが送信される別名のリストです。

次に示すのは、接続情報を提供する `iwoevent.cfg` のエントリのサンプルです。

```
common.trace=on
```

```
alias.edamcs1=172.1.1.1:3694
```

```
alias.edamcs1t=172.1.1.1:3694, trace=on
```

```
alias.edamcs2=222.2.2.2:1234
```

```
trans.JDES00OUT=edamcs1t,edamcs2
```

```
trans.JDEP00OUT=edamcs1
```

5. 定義済ディレクトリ `IWOEVENT_HOME` の下の `iwoevent.cfg` ファイルに指定された別名を使用して、フォルダを作成します。たとえば、次のようになります。

```
\\JDEdwards\E812\DDP\Outbound\edamcs1
```

2.9 ランタイムの概要

J.D. Edwards OneWorld で J.D. Edwards OneWorld Event Listener を起動した後に、リスナーは `iwoevent.cfg` (大文字/小文字が区別されます) という構成ファイルにアクセスします。構成ファイルの情報に基づいて、リスナーは、イベント通知を統合サーバーに送信します。すべてのログ情報は、`iwoevent.log` というファイルに保存されます。`iwoevent.log` ファイルは、`iwoevent.dll` および `iwoevent.cfg` ファイルが存在するアウトバウンド・フォルダに作成されます。

2.10 アウトバウンドおよびインバウンド処理用 JDE.INI ファイルの変更

この項では、XML コール・オブジェクト・カーネル (アウトバウンドおよびインバウンド処理) の `JDE.INI` ファイルに必要な設定について説明します。

`JDE.INI` ファイルを開き、`[JDENET_KERNEL_DEF6]` および `[JDENET_KERNEL_DEF15]` セクションを次のように変更します。

```
[JDENET_KERNEL_DEF6]
krnlName=CALL OBJECT KERNEL
dispatchDLLName=XMLCallObj.dll
dispatchDLLFunction=_XMLTransactionDispatch@28
maxNumberOfProcesses=1
numberOfAutoStartProcesses=1
```

```
[JDENET_KERNEL_DEF15]
krnlName=XML TRANSACTION KERNEL
dispatchDLLName=XMLTransactions.dll
dispatchDLLFunction=_XMLTransactionDispatch@28
maxNumberOfProcesses=1
numberOfAutoStartProcesses=1
```

アンダースコア (`_`) および `@28` を含むパラメータは、Windows NT オペレーティング・システムのみで使用されます。他のオペレーティング・システムの場合は、パラメータを次の表の値に置き換えます。

オペレーティング・システム	Call Object のディスパッチ DLLName	XML Trans のディスパッチ DLLName
AS400	XMLCALLOBJ	XMLTRANS
HP9000B	libxmlcallobj.sl	libxmltransactions.lo
Sun または RS6000	libxmlcallobj.so	Libxmltransactions.so

注意： J.D. Edwards OneWorld のバージョン B7333(XE) のインストールには、`[JDENET_KERNEL_DEF15]` が含まれていません。そのため、バージョン B7333(XE) を使用する場合は、`jde.ini` ファイルに手動で追加する必要があります。J.D. Edwards OneWorld の他のすべてのバージョンでは、インストールで `[JDENET_KERNEL_DEF15]` が含まれます。

Oracle WebLogic Server のデプロイメント および統合

この章では、Oracle WebLogic Server (OracleWLS) の Oracle Application Adapter for J.D. Edwards OneWorld とのデプロイメントおよび統合について説明します。次のトピックが含まれています。

- 3.1 項「Oracle WebLogic Server とのアダプタの統合」
- 3.2 項「アダプタのデプロイメント」
- 3.3 項「アダプタ構成の更新」

関連項目：

- *Oracle Application Server Adapter* コンセプト・ガイド

3.1 Oracle WebLogic Server とのアダプタの統合

Oracle Application Adapter for J.D. Edwards OneWorld は、インストール中に OracleWLS コンテナ内にデプロイされます。すべてのクライアント・アプリケーションは OracleWLS 環境で動作します。J2CA デプロイメントでは、Common Client Interface(CCI) インタフェースにより、OracleWLS クライアント・アプリケーションはリソース・アダプタと統合されます。

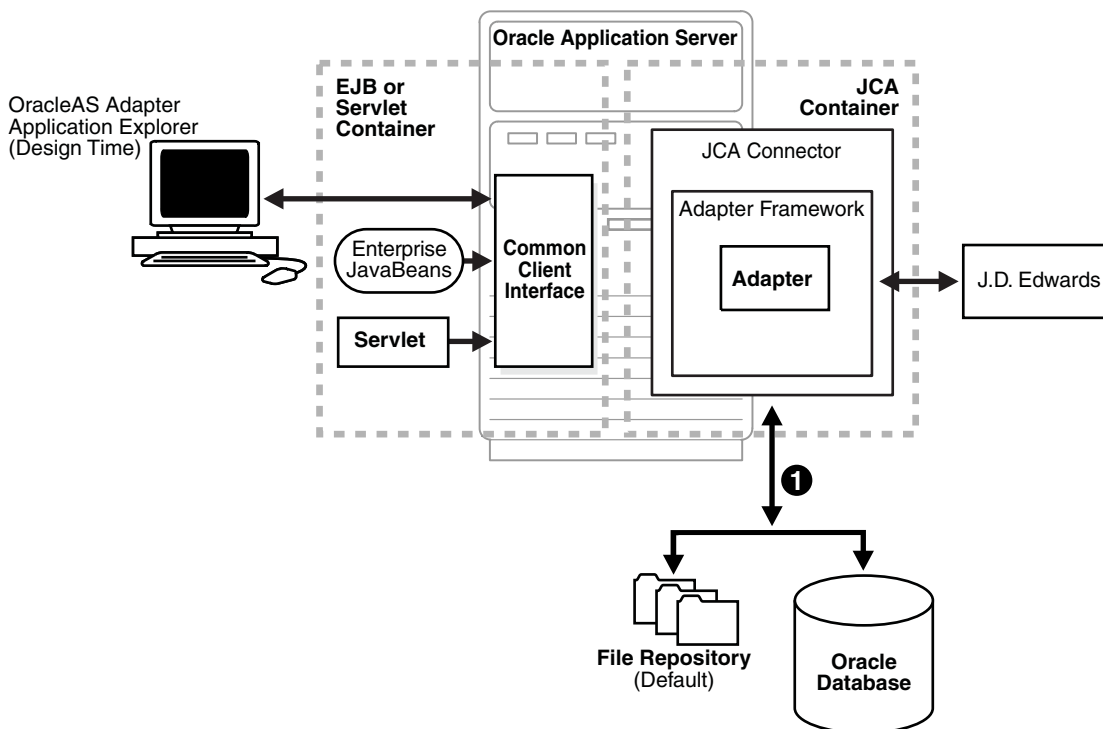
関連項目：

- *Oracle Application Server Adapter* コンセプト・ガイド

3.2 アダプタのデプロイメント

図 3-1 に、Oracle Application Server への J2CA コネクタのデプロイメントを示します。ランタイム・サービス・シナリオでは、Enterprise Java Bean、サーブレットまたは Java プログラム・クライアントによって、J2CA リソース・アダプタへの CCI コールが行われます。アダプタはこれらのコールをリクエストとして処理し、EIS に送信します。その後、EIS のレスポンスがクライアントに返信されます。

図 3-1 Oracle Application Server J2CA アーキテクチャ



① Use either the default file repository or an Oracle database as your repository.

関連項目：

- [Oracle Application Server Adapter コンセプト・ガイド](#)

3.3 アダプタ構成の更新

この項では、次のトピックについて説明します。

- [3.3.1 項「管理対象コネクタ・ファクトリ・オブジェクトの作成」](#)
- [3.3.2 項「複数の管理対象コネクタ・ファクトリ・オブジェクトの作成」](#)
- [3.3.3 項「追加の接続ファクトリ値用の WSDL ファイルの変更」](#)

Oracle Application Adapter for J.D. Edwards OneWorld の J2CA デプロイメント中、OracleWLS により、次の場所に ra.xml と呼ばれるデプロイメント・ディスクリプタが生成されます。

```
<ADAPTER_HOME>\iwafjca.rar\META-INF
```

インストールには、ra.xml というファイルが 1 つ以上含まれています。この項で説明されている OracleWLS デプロイメント・ディスクリプタは、前述のディレクトリ内にあります。

注意： 複数の管理対象接続ファクトリは、アウトバウンド処理 (サービス) の場合のみサポートされています。

3.3.1 管理対象コネクタ・ファクトリ・オブジェクトの作成

ra.xml ディスクリプタには、リソース・アダプタに関する OracleWLS 固有のデプロイメント情報が記載されます。たとえば、アプリケーション・エクスプローラ内のデフォルトの jca_sample 構成は、ra.xml ファイル内では次のように表されます。

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE connector PUBLIC "-//Sun Microsystems, Inc.//DTD Connector 1.0//EN"
'http://java.sun.com/dtd/connector_1_0.dtd'>
<connector>
  <display-name>IWAYJCA10</display-name>
  <vendor-name>IWAY Software</vendor-name>
  <spec-version>1.0</spec-version>
  <eis-type>IWAY</eis-type>
  <version>1.0</version>
  <license>
    <license-required>>false</license-required>
  </license>
  <resourceadapter>

  <managedconnectionfactory-class>com.ibi.afjca.spi.IWAYManagedConnectionFactory</ma
nagedconnectionfactory-class>

  <connectionfactory-interface>javax.resource.cci.ConnectionFactory</connectionfacto
ry-interface>

  <connectionfactory-impl-class>com.ibi.afjca.cci.IWAYConnectionFactory</connectionf
actory-impl-class>
    <connection-interface>javax.resource.cci.Connection</connection-interface>

  <connection-impl-class>com.ibi.afjca.cci.IWAYConnection</connection-impl-class>
    <transaction-support>NoTransaction</transaction-support>
    <config-property>
      <config-property-name>AdapterName</config-property-name>
      <config-property-type>java.lang.String</config-property-type>
      <config-property-value></config-property-value>
    </config-property>
    <config-property>
      <config-property-name>Config</config-property-name>
      <config-property-type>java.lang.String</config-property-type>
      <config-property-value></config-property-value>
    </config-property>
    <config-property>
      <config-property-name>IWayHome</config-property-name>
      <config-property-type>java.lang.String</config-property-type>
      <config-property-value>C:\oracle\Middleware\Oracle_
SOA1\soa\thirdparty\ApplicationAdapters</config-property-value>
    </config-property>
    <config-property>
      <config-property-name>IWayConfig</config-property-name>
      <config-property-type>java.lang.String</config-property-type>
      <config-property-value>jca_sample</config-property-value>
    </config-property>
    <config-property>
      <config-property-name>IWayRepoDriver</config-property-name>
      <config-property-type>java.lang.String</config-property-type>
      <config-property-value></config-property-value>
    </config-property>
    <config-property>
      <config-property-name>IWayRepoURL</config-property-name>
      <config-property-type>java.lang.String</config-property-type>
```

```

    <config-property-value></config-property-value>
  </config-property>
<config-property>
  <config-property-name>IWayRepoUser</config-property-name>
  <config-property-type>java.lang.String</config-property-type>
  <config-property-value></config-property-value>
</config-property>
<config-property>
  <config-property-name>IWayRepoPassword</config-property-name>
  <config-property-type>java.lang.String</config-property-type>
  <config-property-value></config-property-value>
</config-property>
<config-property>
  <config-property-name>LogLevel</config-property-name>
  <config-property-type>java.lang.String</config-property-type>
  <config-property-value>DEBUG</config-property-value>
</config-property>
<authentication-mechanism>
  <authentication-mechanism-type>BasicPassword</authentication-mechanism-type>

<credential-interface>javax.resource.spi.security.PasswordCredential</credential-i
nterface>
  </authentication-mechanism>
  <reauthentication-support>true</reauthentication-support>
</resourceadapter>
</connector>

```

ra.xml ファイルで定義されているパラメータを次の表に示します。

パラメータ名	説明
IWayHome	OracleWLS がパッケージされているアプリケーション・アダプタのベース・インストール・ディレクトリ。
IWayConfig	アプリケーション・エクスプローラで定義されているアダプタ構成名。たとえば、Oracle Application Adapter for J.D. Edwards OneWorld には、アプリケーション・エクスプローラで jca_sample 構成が事前定義されています。
IWayRepoURL	データベースへの接続を開くときに使用される URL。これは Oracle データベースをリポジトリとして使用する場合にのみ必要です。
IWayRepoUser	データベースに接続するときに使用されるユーザー名。これは Oracle データベースをリポジトリとして使用する場合にのみ必要です。
IWayRepoPassword	パスワード。指定された場合は、構成を上書きします。これは Oracle データベースをリポジトリとして使用する場合にのみ必要です。
loglevel	ManagedConnectorFactory プロパティによって設定されるレベルを上書きします。

3.3.2 複数の管理対象コネクタ・ファクトリ・オブジェクトの作成

複数の管理対象コネクタ・ファクトリ・オブジェクトを作成するには、weblogic-ra.xml ファイルを編集し、さらに <connection-instance> ノードを追加します。このファイルは次の場所にあります。

```
<ADAPTER_HOME>\iwafjca.rar\META-INF
```

たとえば、アプリケーション・エクスプローラ内の最初の `jca_configuration` は `weblogic-ra.xml` ファイル内では次のように表現されます。

```
<?xml version="1.0"?>
<weblogic-connector xmlns="http://www.bea.com/ns/weblogic/90">
  <enable-access-outside-app>true</enable-access-outside-app>
  <enable-global-access-to-classes>true</enable-global-access-to-classes>
  <outbound-resource-adapter>
    <default-connection-properties>
      <pool-params>
        <initial-capacity>0</initial-capacity>
      </pool-params>
      <transaction-support>LocalTransaction</transaction-support>
    </default-connection-properties>
    <connection-definition-group>

<connection-factory-interface>javax.resource.cci.ConnectionFactory</connection-factory-interface>
  <connection-instance>
    <jndi-name>eis/OracleJCAAdapter/DefaultConnection</jndi-name>
  </connection-instance>
</connection-definition-group>
</outbound-resource-adapter>
</weblogic-connector>
```

複数の管理対象コネクタ・ファクトリ・オブジェクトを作成するには、新規の `<connection-instance>` ノードをファイルに追加する必要があります。たとえば、次のようになります。

```
<?xml version="1.0"?>
<weblogic-connector xmlns="http://www.bea.com/ns/weblogic/90">

  <enable-access-outside-app>true</enable-access-outside-app>
  <enable-global-access-to-classes>true</enable-global-access-to-classes>

  <outbound-resource-adapter>
    <default-connection-properties>
      <pool-params>
        <initial-capacity>0</initial-capacity>
      </pool-params>
      <transaction-support>LocalTransaction</transaction-support>
    </default-connection-properties>
    <connection-definition-group>

<connection-factory-interface>javax.resource.cci.ConnectionFactory</connection-factory-interface>
  <connection-instance>
    <jndi-name>eis/OracleJCAAdapter/DefaultConnection</jndi-name>
  </connection-instance>
  <connection-instance>
    <jndi-name>eis/OracleJCAAdapter/DefaultConnection1</jndi-name>
    <connection-properties>
      <properties>
        <property>
<name>IWayHome</name>
<value>C:\oracle\Middleware\Oracle_SOA1\soa\thirdparty\ApplicationAdapters</value>
        </property>
        <property>
<name>IWayConfig</name>
<value>jca_sample2</value>
        </property>
```

```

        <property>
<name>IWayRepoURL</name>
<value></value>
        </property>
        <property>
<name>IWayRepoUser</name>
<value></value>
        </property>
        <property>
<name>IWayRepoPassword</name>
<value></value>
        </property>
        <property>
<name>LogLevel</name>
<value>Debug</value>
        </property>
    </properties>
</connection-properties>
</connection-instance>
</connection-definition-group>
</outbound-resource-adapter>
</weblogic-connector>

```

<connection-instance> セクションで <property> 要素を指定しない場合、ra.xml ファイルから値が取得されます。ra.xml ファイルでデフォルトのプロパティを指定し、weblogic-ra.xml ファイルで必要に応じてそれらをオーバーライドできます。また、J2CA 構成 (jca_sample2 など) が、アプリケーション・エクスペローラ内で作成済である必要があることに注意してください。

注意： ra.xml および weblogic-ra.xml ファイルを変更するとき、アプリケーション・サーバーを再起動する必要があります。アプリケーション・サーバーがすでに実行されている場合は、アプリケーション・サーバーを停止してから再起動してください。

また、iwafjca.rar ファイルを Oracle WebLogic 管理コンソールに再デプロイして、これらの変更をアクティブ化する必要もあります。

3.3.3 追加の接続ファクトリ値用の WSDL ファイルの変更

アプリケーション・エクスペローラでは、eis/OracleJCAAdapter/DefaultConnection という名前のデフォルトの接続ファクトリを使用して J2CA プロパティ・ファイルが生成されます。追加の接続ファクトリを作成した場合は、J2CA プロパティ・ファイル内の jca:address セクションのロケーション・フィールドを反映するように、追加の構成および接続ファクトリ用に生成された WSDL を変更する必要があります。isdsrv2_conn2 の構成が含まれている Oracle Application Adapter for J.D. Edwards OneWorld 用のデフォルトの J2CA プロパティ・ファイルを、次の例に示します。

J2CA プロパティ・ファイルに、次のデフォルトの接続ファクトリが含まれていることがわかります。

```

eis/OracleJCAAdapter/DefaultConnection
<jca:address location="eis/OracleJCAAdapter/DefaultConnection"
    ConnectionSpec="com.ibi.afjca.cci.IWAFConnectionSpec"
    cs.AdapterName="JDEdwards" cs.Config="isdsrv2_conn2"
    UIConnectionName="Connection1"/>

```

この接続ファクトリ値を次の値に変更する必要があります。
eis/OracleJCAAdapter/DefaultConnection1

たとえば、次のようになります。

```
<jca:address location="eis/OracleJCAAdapter/DefaultConnection1"
             ConnectionSpec="com.ibi.afjca.cci.IWAFConnectionSpec"
             cs.AdapterName="JDEdwards" cs.Config="isdsrv2_conn2"
             UIConnectionName="Connection1"/>
```

`jca:address` セクションのロケーション・フィールドの値のみ変更可能です。その他のフィールドまたはセクションは変更しないでください。

Oracle SOA Suite での BPEL サービス・コンポーネントの統合

Oracle Application Adapter for J.D. Edwards OneWorld には、Business Process Execution Language (BPEL) Process Manager がシームレスに統合されており、Web サービスの統合を容易にします。Oracle BPEL Process Manager は、サービス指向アーキテクチャ (SOA) に基づいています。これは、Web Service Definition Language (WSDL) ドキュメントとして公開されているアダプタ・サービスを使用します。

この章の内容は以下のとおりです。

- 4.1 項「概要」
- 4.2 項「アダプタのデプロイメント」
- 4.3 項「新規のアプリケーション・サーバー接続の構成」
- 4.4 項「サービス統合のためのアウトバウンド BPEL プロセスの設計 (J2CA 構成)」
- 4.5 項「イベント統合のためのインバウンド BPEL プロセスの設計 (J2CA 構成)」
- 4.6 項「サービス統合のためのアウトバウンド BPEL プロセスの設計 (BSE 構成)」

4.1 概要

Oracle BPEL Process Manager に統合するには、Oracle Application Adapter for J.D. Edwards OneWorld が Oracle BPEL Process Manager と同じ WLS コンテナにデプロイされている必要があります。また、基になるアダプタ・サービスは、アダプタのリクエスト/レスポンス (アウトバウンド) とイベント通知 (インバウンド) の両方のサービス用に Oracle Application Adapter アプリケーション・エクスプローラ (アプリケーション・エクスプローラ) でデザインタイム中に生成される WSDL ファイルとして公開される必要があります。詳細は、[第 2 章「Oracle Application Adapter for J.D. Edwards OneWorld の構成」](#) を参照してください。

生成された WSDL ファイルは、インバウンドまたはアウトバウンドのアダプタ・サービスに対して適切な BPEL プロセスの設計に使用されます。完成した BPEL プロセスは、Oracle JDeveloper で正常にコンパイルされ、BPEL サーバーにデプロイされる必要があります。BPEL サーバーへのデプロイメント時に、新規構築されたプロセスはすべて自動的に Oracle Enterprise Manager Console にデプロイされ、BPEL プロセスの実行、監視、管理、およびアダプタ・イベントのリスニングを行うことができます。

4.2 アダプタのデプロイメント

インストール中、Oracle Application Adapter for J.D. Edwards OneWorld は、WLS コンテナ内の J2CA 1.0 リソース・アダプタとしてデプロイされます。アダプタは、Oracle BPEL Process Manager と同じ WLS コンテナ内にデプロイされる必要があります。

関連項目： *Oracle Application Server Adapter* コンセプト・ガイド

4.3 新規のアプリケーション・サーバー接続の構成

Oracle JDeveloper で新規のアプリケーション・サーバー接続を構成するには、次のステップを実行します。

1. システムで **Oracle JDeveloper** を開きます。
2. 図 4-1 に示すように、メニュー・バーで、「ウィンドウ」をクリックし、「アプリケーション・サーバー・ナビゲータ」を選択します。

図 4-1 アプリケーション・サーバー・ナビゲータ

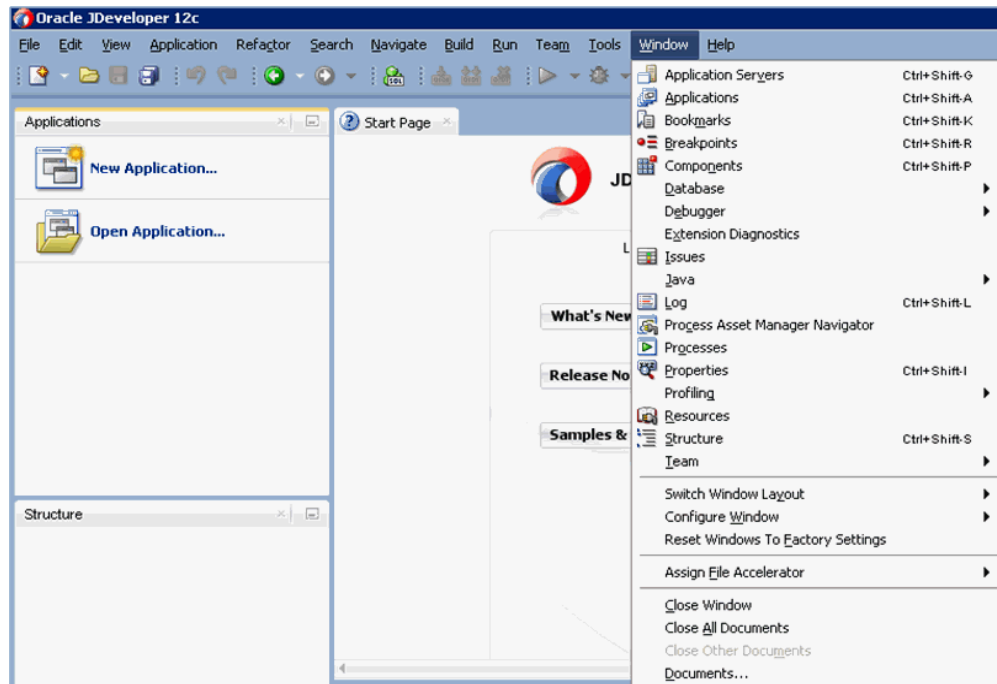
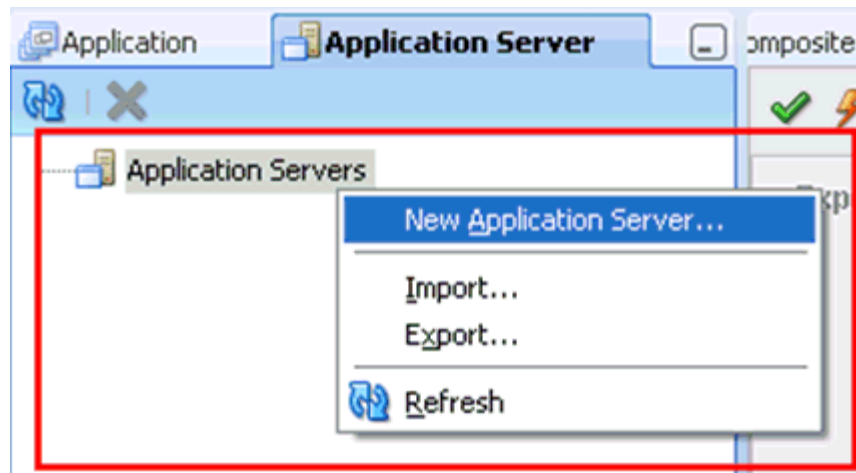


図 4-2 に示すように、「アプリケーション・サーバー」タブが表示されます。

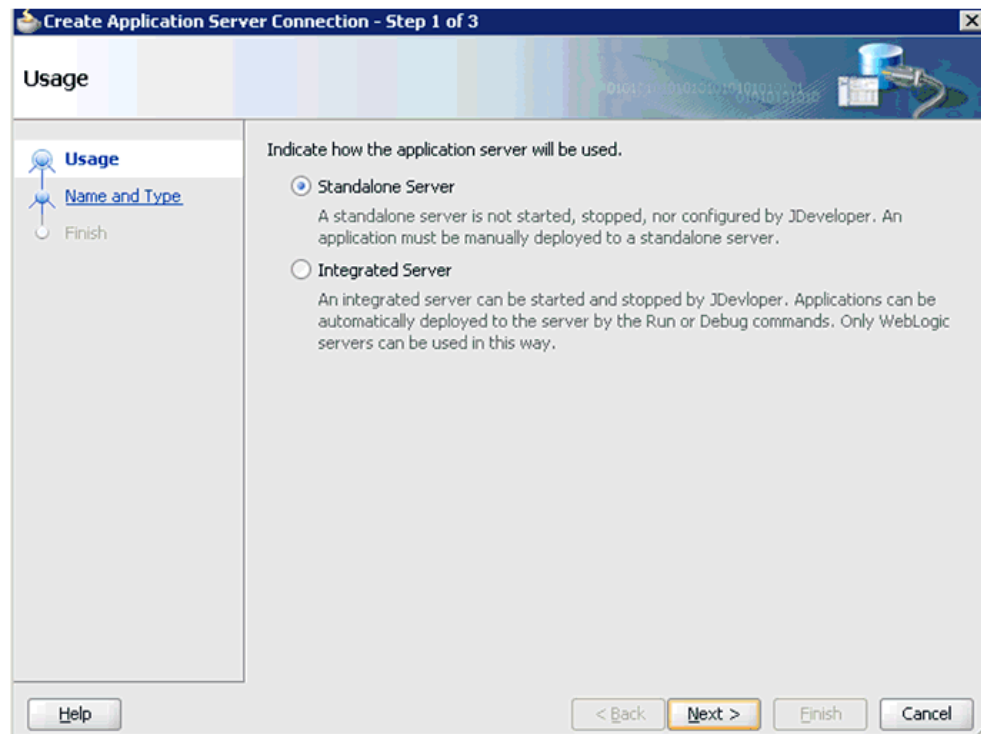
図 4-2 「アプリケーション・サーバー」タブ



3. 「アプリケーション・サーバー」を右クリックし、「新規アプリケーション・サーバー」を選択します。

図 4-3 に示すように、アプリケーション・サーバー接続の作成ウィザードが表示されます。

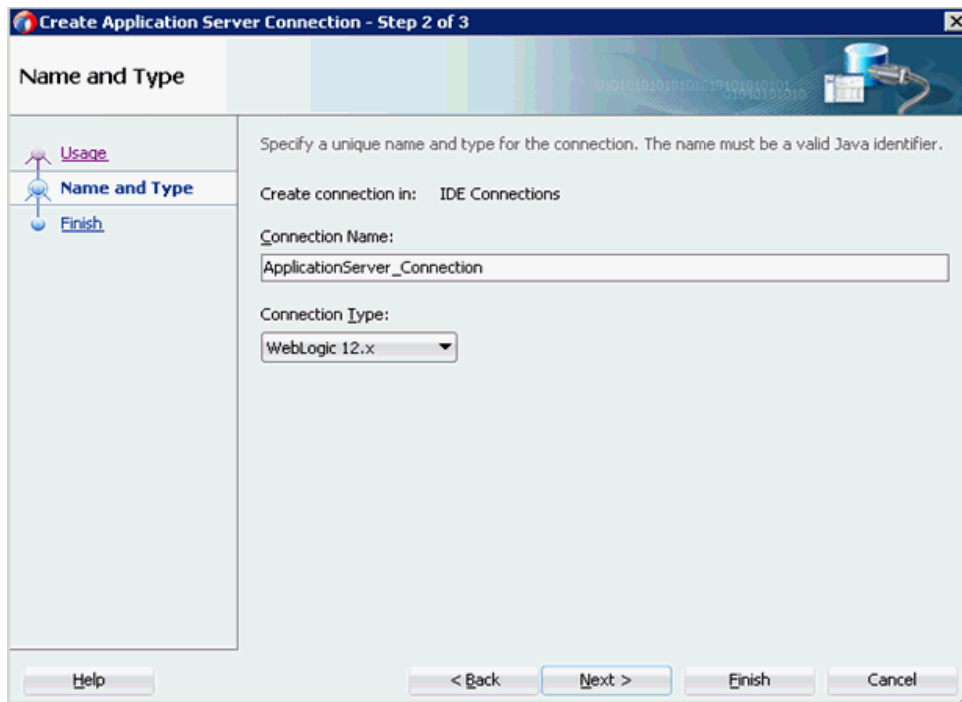
図 4-3 アプリケーション・サーバー接続の作成ウィザード



4. デフォルト選択 (スタンドアロン・サーバー) を受け入れ、「次へ」をクリックします。

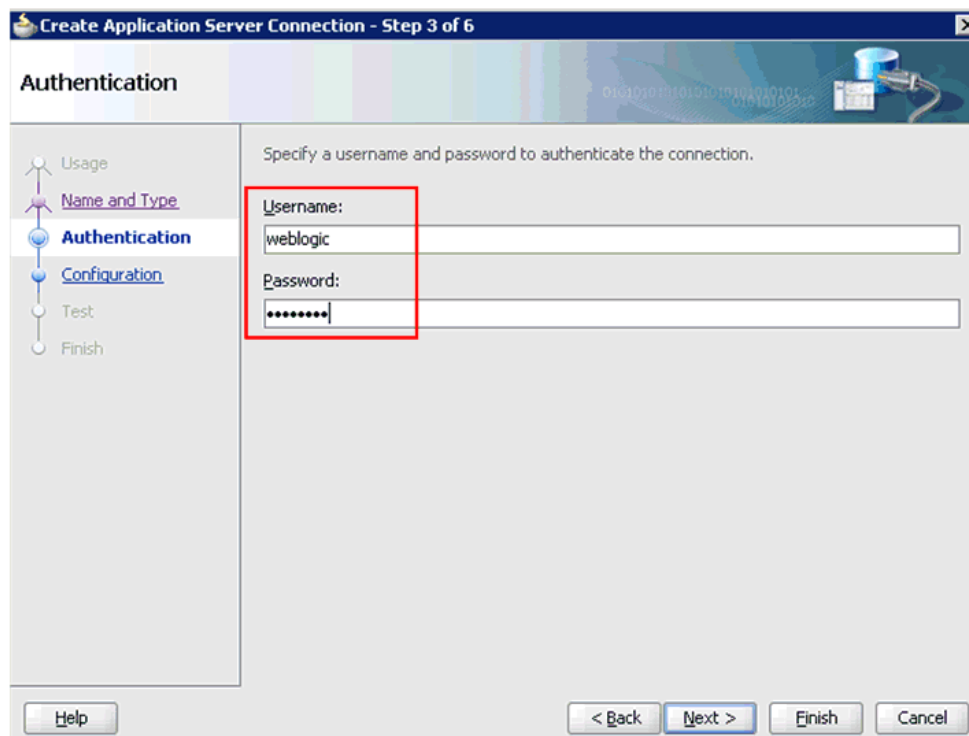
図 4-4 に示すように、「名前とタイプ」ページが表示されます。

図 4-4 「名前とタイプ」 ペイン



5. アプリケーション・サーバー接続の名前を指定し、「次へ」をクリックします。
 図 4-5 に示すように、「認証」 ページが表示されます。

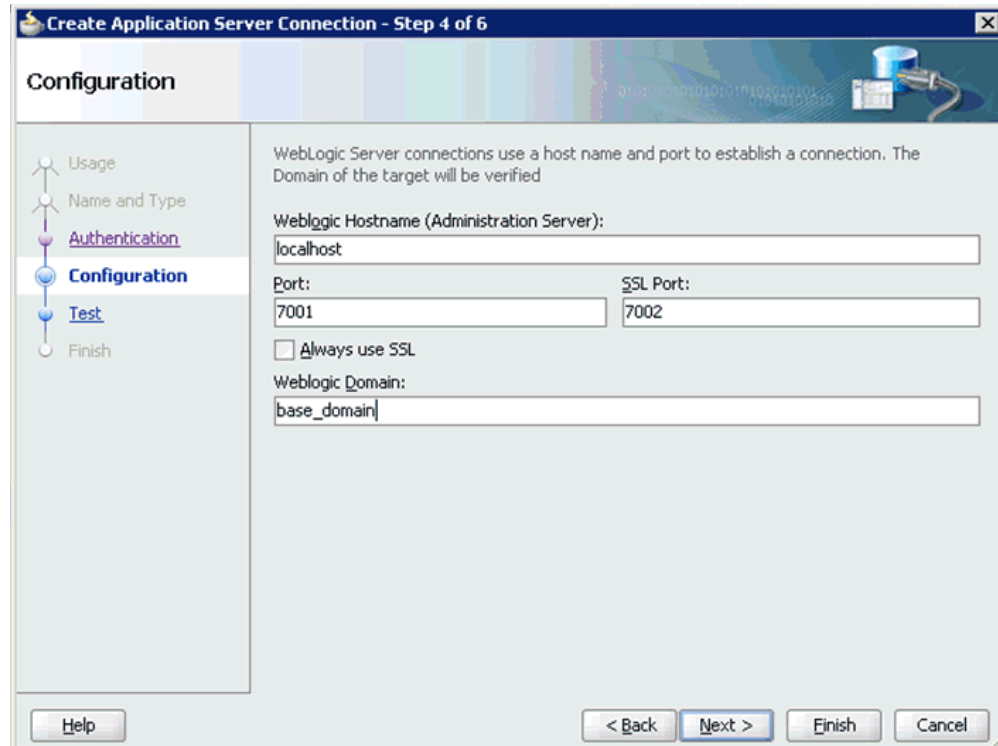
図 4-5 「認証」 ページ



- 新規の接続用のユーザー名 (「weblogic」など) およびパスワード (「welcome1」など) を指定します。
- 「次へ」をクリックします。

図 4-6 に示すように、「構成」ページが表示されます。

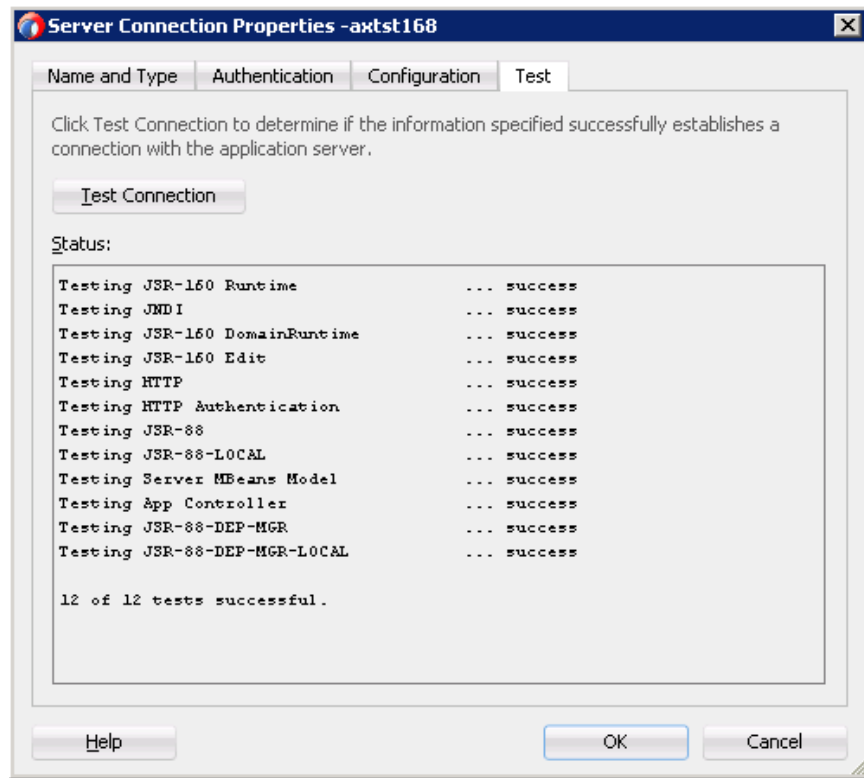
図 4-6 「構成」ページ



- Oracle WebLogic のホストの名前を指定します (「localhost」など)。これは、プロセスのデプロイが必要なシステムの IP および Oracle WebLogic のドメインです (「base_domain」など)。
- 「次へ」をクリックします。

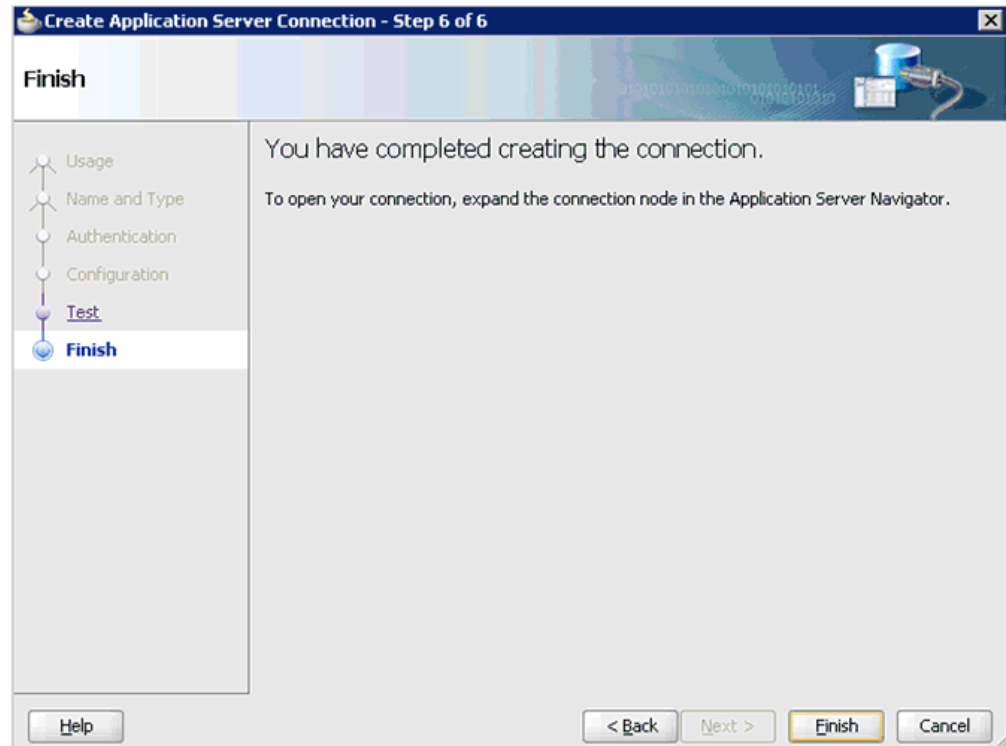
図 4-7 に示すように、「テスト」ページが表示されます。

図 4-7 「テスト」 ページ



10. 「接続のテスト」をクリックします。
11. テストのステータスが成功であることを確認します。
12. 「次へ」をクリックします。
図 4-8 に示すように、「終了」ページが表示されます。

図 4-8 「終了」 ページ



13. 「終了」をクリックします。

新規のアプリケーション・サーバー接続が左ペイン (「アプリケーション・サーバー」タブ) に表示されます。

4.4 サービス統合のためのアウトバウンド BPEL プロセスの設計 (J2CA 構成)

この項では、サービス統合のためのアウトバウンド BPM プロセスの設計方法について説明します。

このアウトバウンド・ユース・ケース・シナリオ用のサンプル・プロジェクトが、次に示す Application Adapter のインストール・フォルダに用意されています。

```
<ADAPTER_HOME>\etc\sample\JDEdwards_Samples.zip\JDEdwards_Samples\BPEL\J2CA\Outbound_Project
```

アダプタのデザインタイム構成を完了するには、次のツールが必要です。

- Oracle Adapter アプリケーション・エクスプローラ (アプリケーション・エクスプローラ)
- Oracle JDeveloper BPEL デザイナ (JDeveloper)

注意： この章に記載されている例では、Oracle JDeveloper が使用されています。

この項では、次のトピックについて説明します。

- [4.4.1 項「リクエスト / レスポンス型のサービス用の WSDL の生成」](#)

- 4.4.2 項「SOA 用の空のコンポジットの作成」
- 4.4.3 項「BPEL アウトバウンド・プロセスの定義」
- 4.4.4 項「BPEL アウトバウンド・プロセスのデプロイ」
- 4.4.5 項「Oracle Enterprise Manager Console での入力 XML ドキュメントの起動」
- 4.4.6 項「アウトバウンド BPEL およびアウトバウンド・メディエータ・プロセスのテスト」

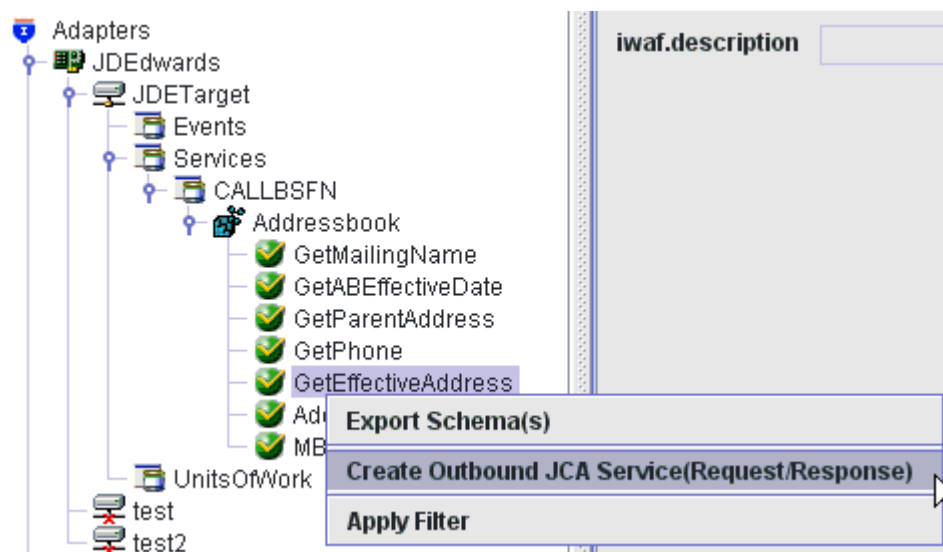
BPEL プロセスを設計する前に、アプリケーション・エクスプローラを使用して WSDL を生成する必要があります。詳細は、4-8 ページの「リクエスト/レスポンス型のサービス用の WSDL の生成」を参照してください。アプリケーション・エクスプローラで生成された WSDL は、BPEL プロセスの構成中に使用されます。

4.4.1 リクエスト/レスポンス型のサービス用の WSDL の生成

リクエスト/レスポンス型のサービス用の WSDL を生成するには、次のステップを実行します。

1. 「アプリケーション・エクスプローラ」を起動し、定義済の J.D. Edwards OneWorld ターゲットに接続します。
詳細は、2-5 ページの「J.D. Edwards OneWorld に対するターゲットの定義」を参照してください。
2. 「サービス」、「CALLBSFN」、アドレス帳の順に展開します。

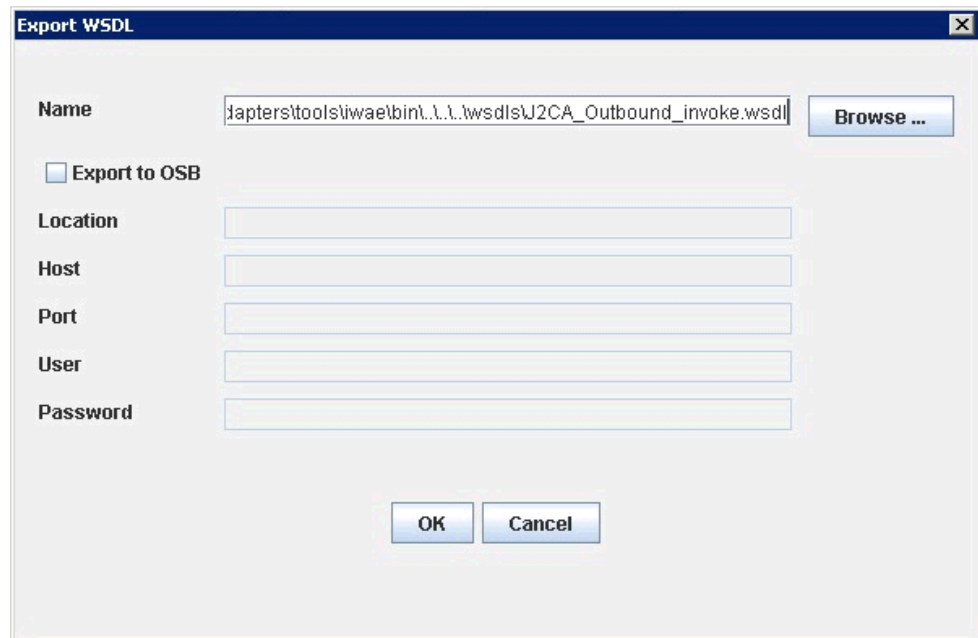
図 4-9 アウトバウンド JCA サービスの作成 (リクエスト/レスポンス) オプション



3. 図 4-9 に示すように、「GetEffectiveAddress」を右クリックし、アウトバウンド JCA サービスの作成 (リクエスト/レスポンス) を選択します。

図 4-10 に示すように、「WSDL のエクスポート」ダイアログが表示されます。

図 4-10 「WSDL のエクスポート」ダイアログ



4. ファイルのデフォルト名を受け入れます。

.wsdl ファイル拡張子が自動的に追加されます。デフォルトでは、リクエスト/レスポンス型のサービス用に生成された WSDL ファイルの名前の末尾には「_invoke」が付きます。

5. 「OK」をクリックします。

これで、Oracle JDeveloper での BPEL アウトバウンド・プロセスの定義に必要な最初のステップである、新規の SOA アプリケーションを作成する準備ができました。

4.4.2 SOA 用の空のコンポジットの作成

SOA 用に空のコンポジットを作成するには、次のステップを実行します。

1. 新規の SOA アプリケーションを作成します。
2. 図 4-11 に示すように、新規の SOA アプリケーションの名前を入力し、「次へ」をクリックします。

図 4-11 「アプリケーションの名前付け」ページ

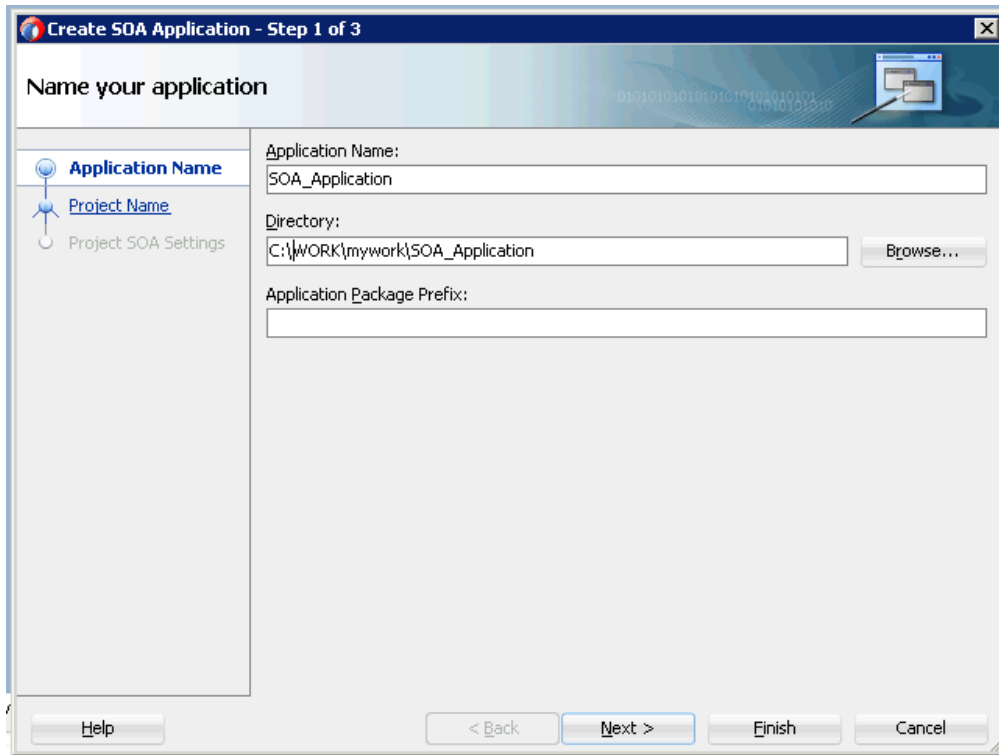
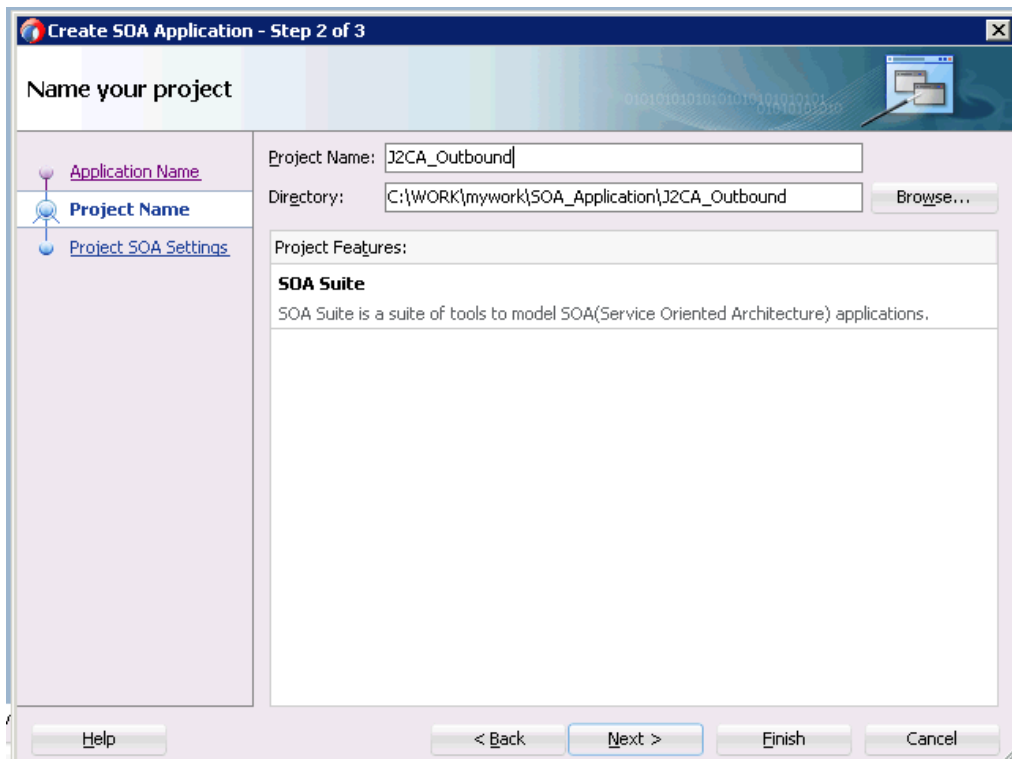


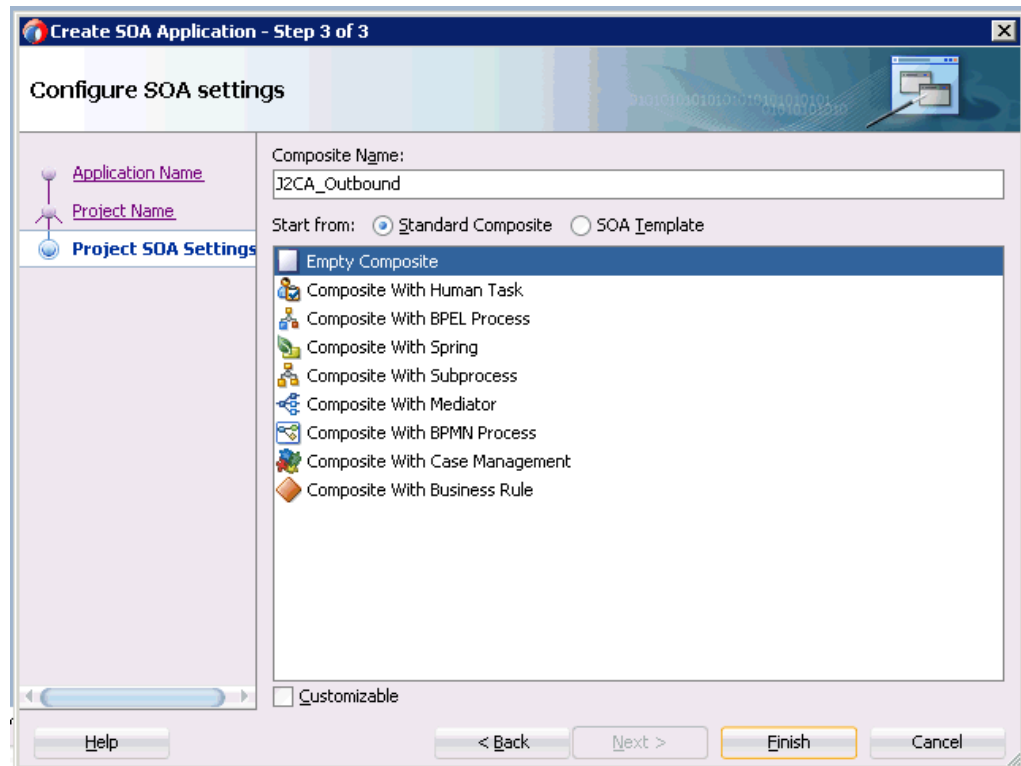
図 4-12 に示すように、「プロジェクトの名前付け」ページが表示されます。

図 4-12 「プロジェクトの名前付け」ページ



- プロジェクト名を入力し、「次へ」をクリックします。
 図 4-13 に示すように、「SOA 設定の構成」ページが表示されます。

図 4-13 「SOA 設定の構成」ページ



- 「コンポジット・テンプレート」リストで「空のコンポジット」を選択し、「終了」をクリックします。

4.4.3 BPEL アウトバウンド・プロセスの定義

この項では、BPEL アウトバウンド・プロセスの定義方法について説明します。次のトピックで構成されています。

- 4.4.3.1 項「サード・パーティ・アダプタ・サービス・コンポーネントの構成」
- 4.4.3.2 項「アウトバウンド BPEL プロセス・コンポーネントの構成」
- 4.4.3.3 項「12c でのデプロイメントに関する既知の問題への対応」

4.4.3.1 サード・パーティ・アダプタ・サービス・コンポーネントの構成

サード・パーティ・アダプタ・サービス・コンポーネントを作成するには、次のステップを実行します。

- 図 4-14 に示すように、「サービス・アダプタ」ペインから「サード・パーティ・アダプタ」コンポーネントを「外部参照」ペインにドラッグ・アンド・ドロップします。

図 4-14 「サード・パーティ・アダプタ」コンポーネント

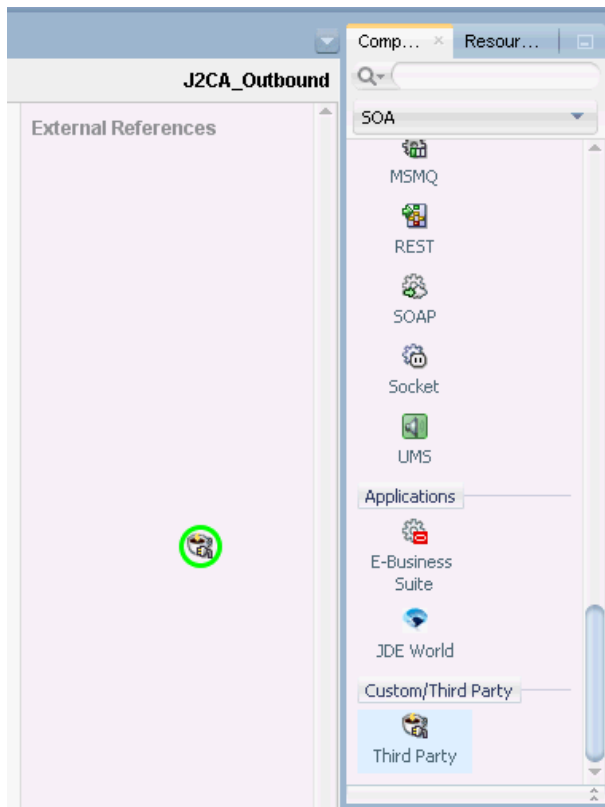
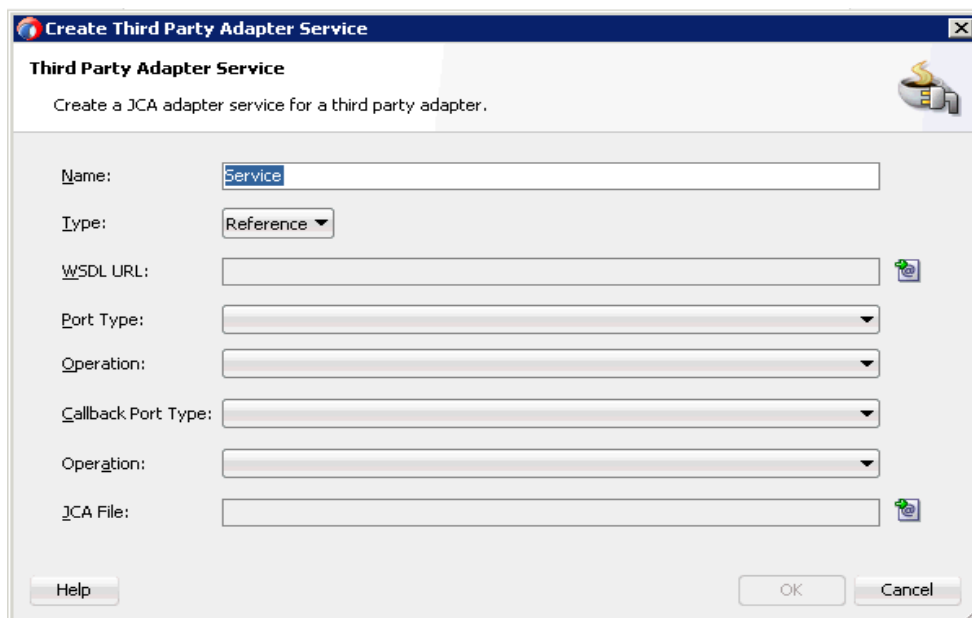


図 4-15 に示すように、「サード・パーティ・アダプタ・サービスの作成」ダイアログが表示されます。

図 4-15 「サード・パーティ・アダプタ・サービスの作成」ダイアログ

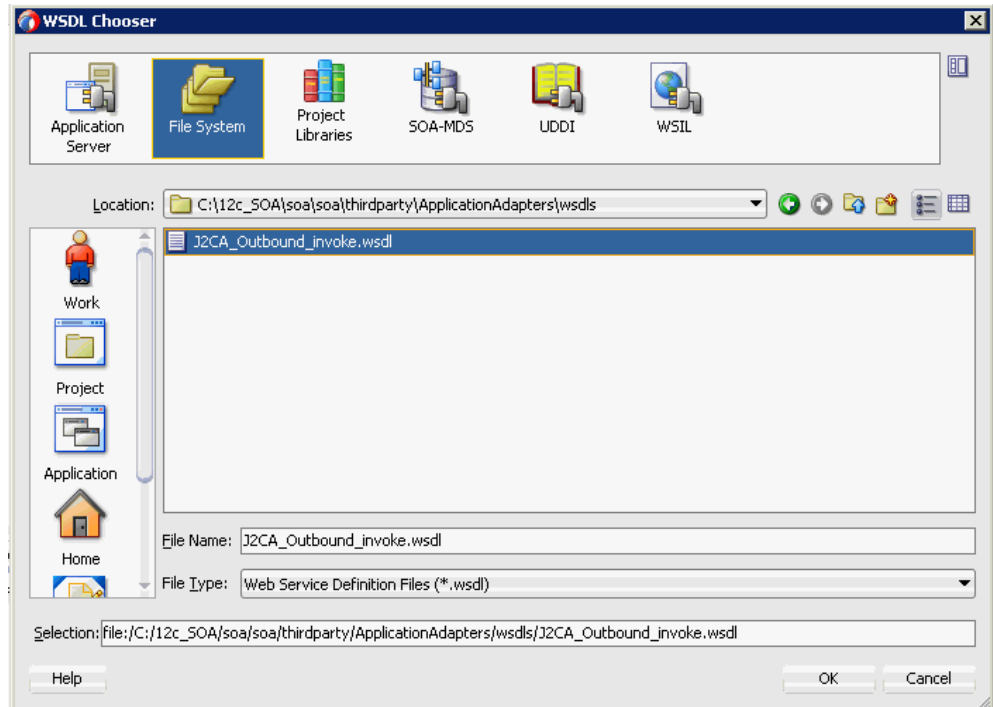


2. 「タイプ」リストで「参照」が選択されていることを確認します (デフォルト)。

3. 「WSDL URL」フィールドの右の「既存の WSDL を検索します。」アイコンをクリックします。

図 4-16 に示すように、「WSDL チューザ」ダイアログが表示されます。

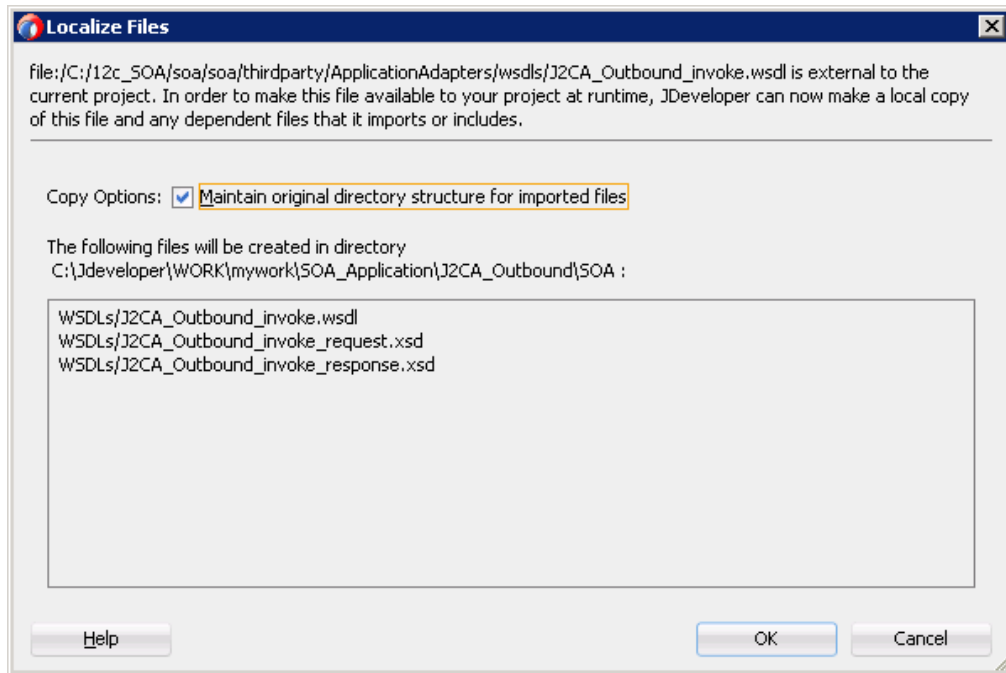
図 4-16 WSDL チューザ」ダイアログ



4. 次のディレクトリまでブラウズし、アウトバウンド WSDL ファイルを選択します。
<ADAPTER_HOME>\wsdl
5. 「OK」をクリックします。

図 4-17 に示すように、「ファイルのローカライズ」ダイアログが表示されます。

図 4-17 「ファイルのローカライズ」 ダイアログ

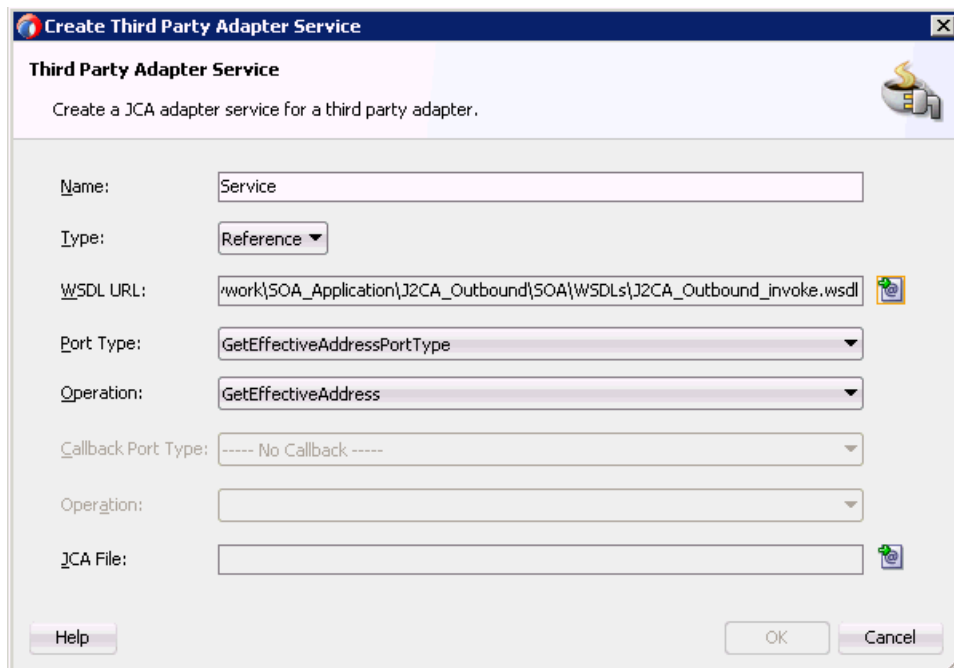


6. 「OK」をクリックします。

アウトバウンド WSDL ファイルおよび関連リクエスト/レスポンス XML スキーマ・ファイル (.xsd) が、作成済のプロジェクト・フォルダにインポートされます。

図 4-18 に示すように、「サード・パーティ・アダプタ・サービスの作成」ダイアログに戻ります。

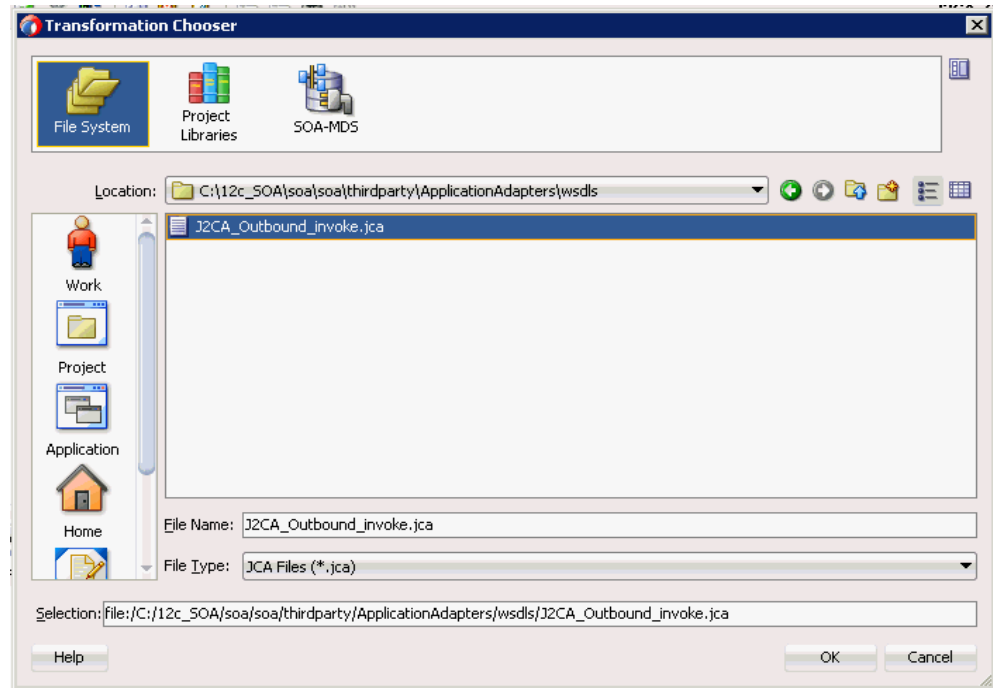
図 4-18 「サード・パーティ・アダプタ・サービスの作成」ダイアログ



- 「JCA ファイル」フィールドの右の「JCA ファイルを検索します。」アイコンをクリックします。

図 4-19 に示すように、変換チューザ・ダイアログが表示されます。

図 4-19 変換チューザ・ダイアログ



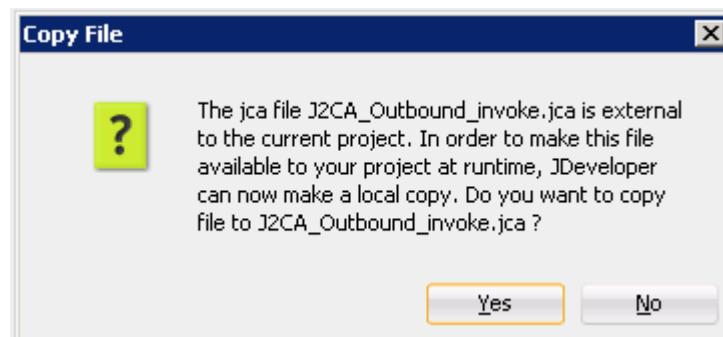
- 次のディレクトリまでブラウズし、JCA プロパティ・ファイルを選択します。

<ADAPTER_HOME>\wsdls

- 「OK」をクリックします。

図 4-20 に示すように、「ファイルのコピー」メッセージが表示されます。

図 4-20 「ファイルのコピー」メッセージ

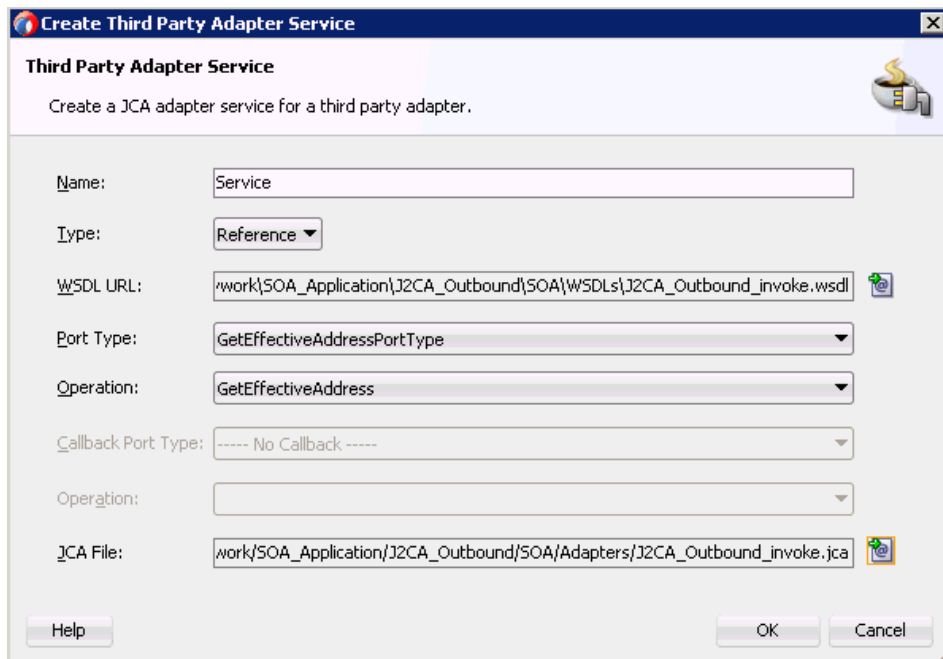


- 「はい」をクリックします。

JCA プロパティ・ファイルのコピーがプロジェクト・フォルダ内に作成されます。

図 4-21 に示すように、「サード・パーティ・アダプタ・サービスの作成」ダイアログに戻ります。

図 4-21 「サード・パーティ・アダプタ・サービスの作成」ダイアログ



11. 「OK」をクリックします。

サード・パーティ・アダプタ・サービス・コンポーネントが作成され、「外部参照」ペインに表示されます。

これで、アウトバウンド BPEL プロセス・コンポーネントを構成する準備ができました。

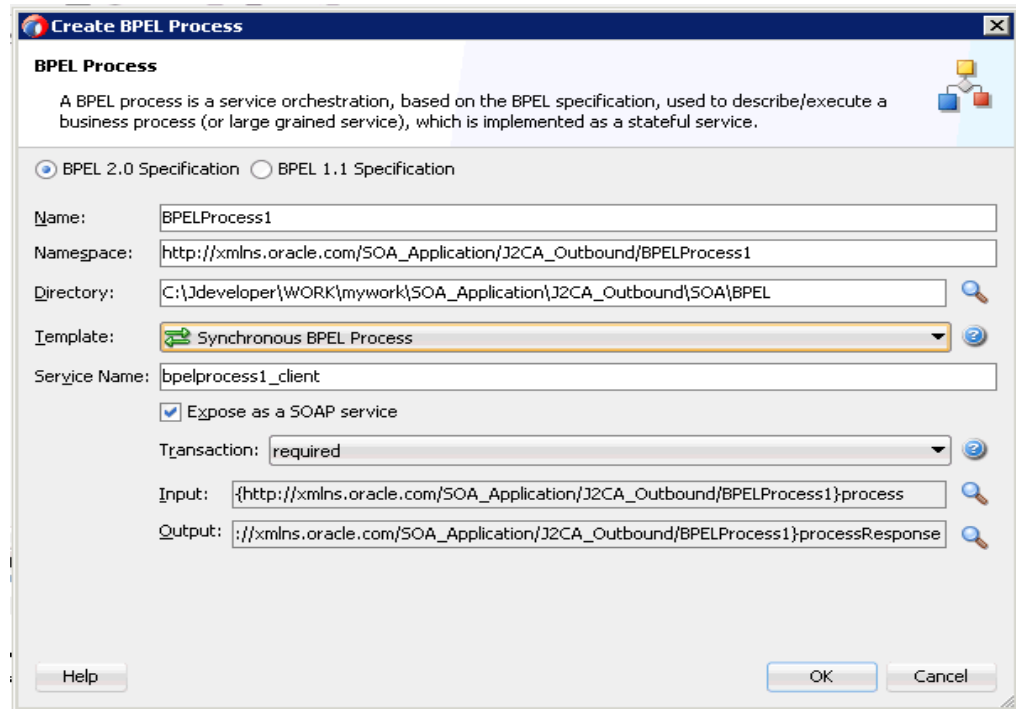
4.4.3.2 アウトバウンド BPEL プロセス・コンポーネントの構成

アウトバウンド BPEL プロセス・コンポーネントを構成するには、次のステップを実行します。

1. 「コンポーネント」ペインから「BPEL プロセス」コンポーネントを「コンポーネント」ペインにドラッグ・アンド・ドロップします。

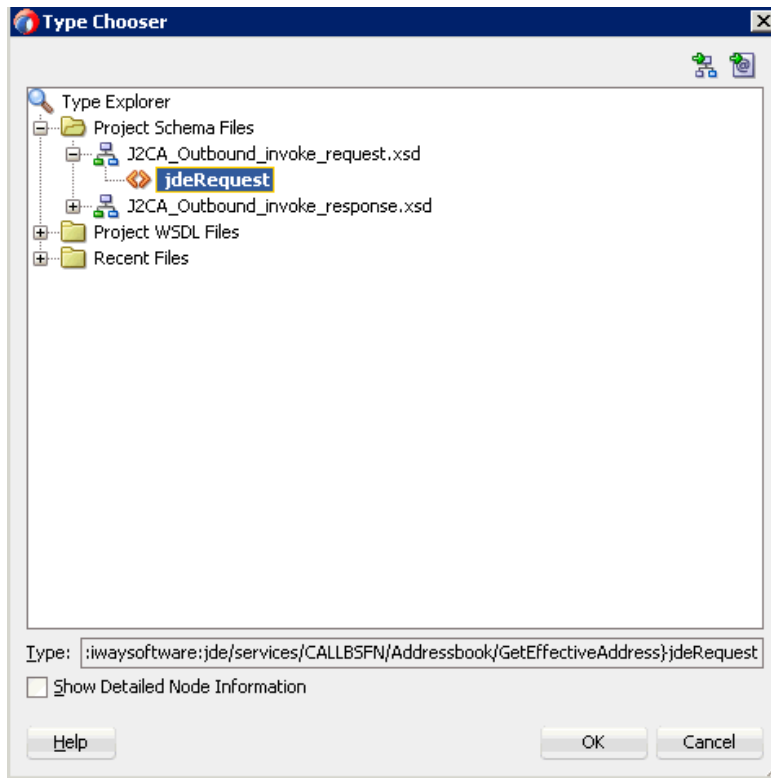
図 4-22 に示すように、「BPEL プロセスの作成」ダイアログが表示されます。

図 4-22 「BPEL プロセスの作成」 ダイアログ



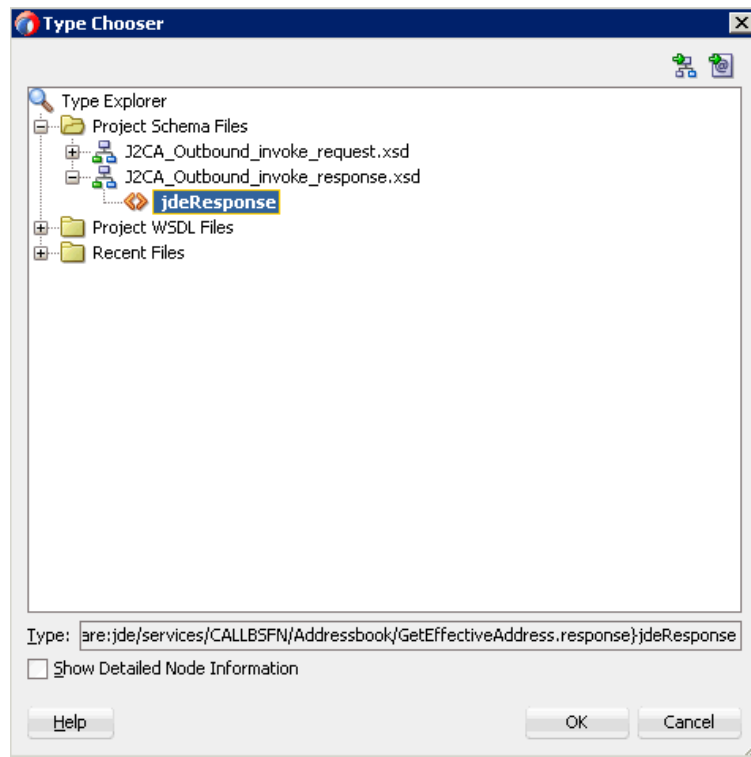
2. 「名前」フィールドで、新規のアウトバウンド BPEL プロセス・コンポーネントの名前を入力するか、デフォルト値をそのまま使用します。
デフォルトでは、「BPEL 2.0 仕様」オプションが選択されます。
3. 「テンプレート」リストで、「同期 BPEL プロセス」を選択します。
4. 関連する XML リクエスト・スキーマ・ファイルの選択用の「入力」フィールドの右の「参照」アイコンをクリックします。
図 4-23 に示すように、「タイプ・チューザ」ダイアログが表示されます。

図 4-23 「タイプ・チューザ」ダイアログ



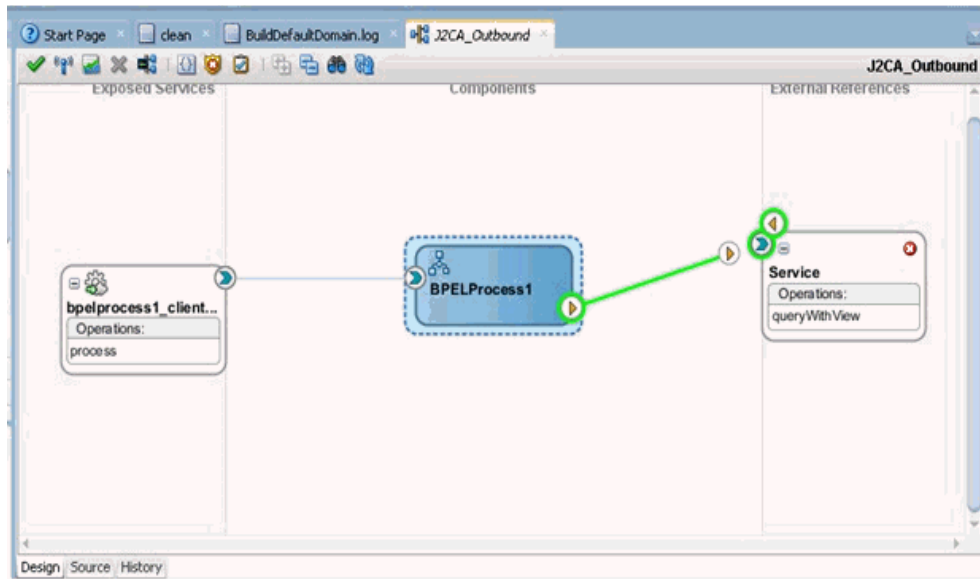
5. 「プロジェクトのスキーマ・ファイル」、「J2CA_Outbound_invoke_request.xsd」の順に展開し、「jdeRequest」を選択します。
6. 「OK」をクリックします。
「BPEL プロセスの作成」ダイアログに戻ります。
7. 関連する XML レスポンス・スキーマ・ファイルの選択用の「出力」フィールドの右の「参照」アイコンをクリックします。
図 4-24 に示すように、「タイプ・チューザ」ダイアログが表示されます。

図 4-24 「タイプ・チューザ」ダイアログ



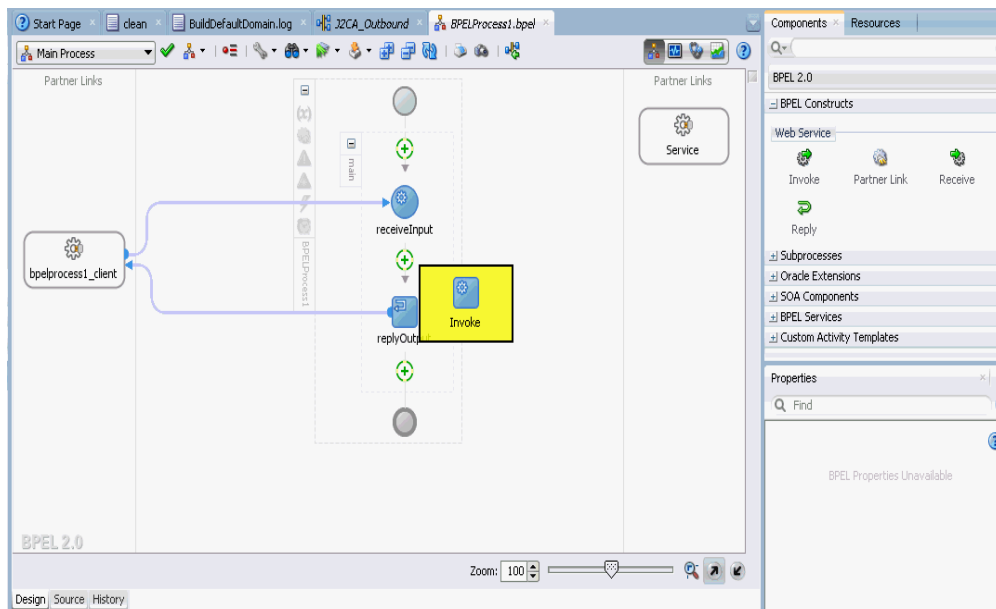
8. 「プロジェクトのスキーマ・ファイル」、「J2CA_Outbound_invoke_response.xsd」の順に展開し、「jdeResponse」を選択します。
9. 「OK」をクリックします。
「BPEL プロセスの作成」ダイアログに戻ります。
10. 「OK」をクリックします。
11. 図 4-25 に示すように、アウトバウンド BPEL プロセス・コンポーネントとサード・パーティ・アダプタ・サービス・コンポーネント間の接続を作成します。

図 4-25 作成された接続



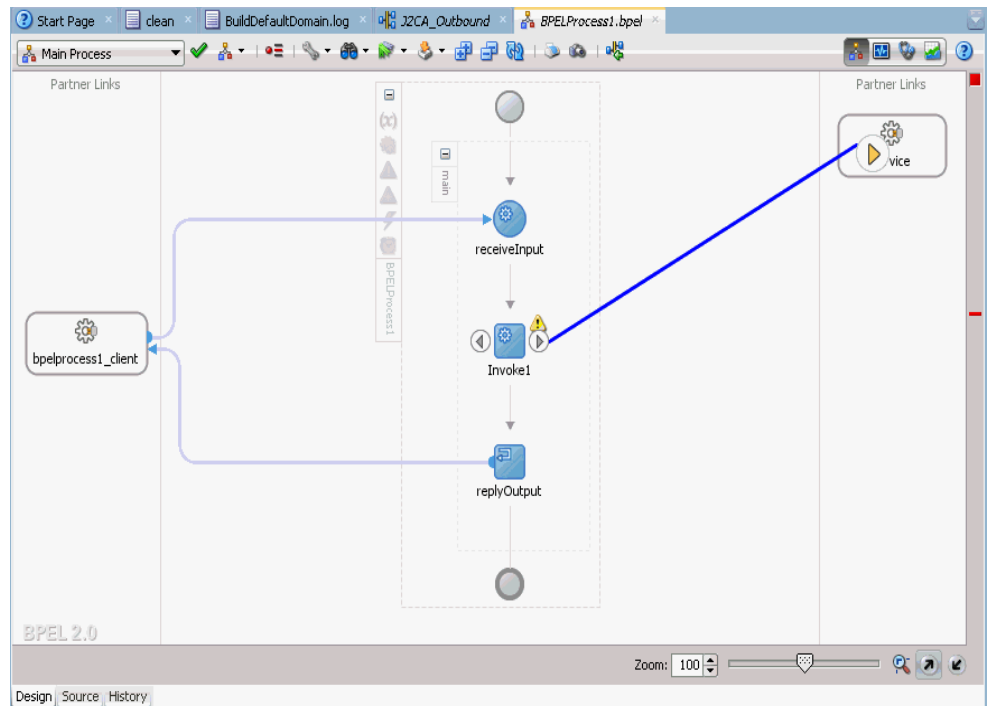
12. 「コンポーネント」 ペインで、アウトバウンド BPEL プロセス・コンポーネントをダブルクリックします。
13. 図 4-26 に示すように、「BPEL コンストラクト」の「Web サービス」の下の「Invoke」アクティビティ・コンポーネントを「コンポーネント」ペインにドラッグ・アンド・ドロップし、「receiveInput」アクティビティ・コンポーネントと「replyOutput」アクティビティ・コンポーネントの間に置きます。

図 4-26 「Invoke」アクティビティ・コンポーネント



14. 図 4-27 に示すように、新規の「Invoke」アクティビティ・コンポーネント (Service) とサード・パーティ・アダプタ・サービス・コンポーネント (Service) の間に接続を作成します。

図 4-27 作成された接続



「起動の編集」ダイアログが表示されます。

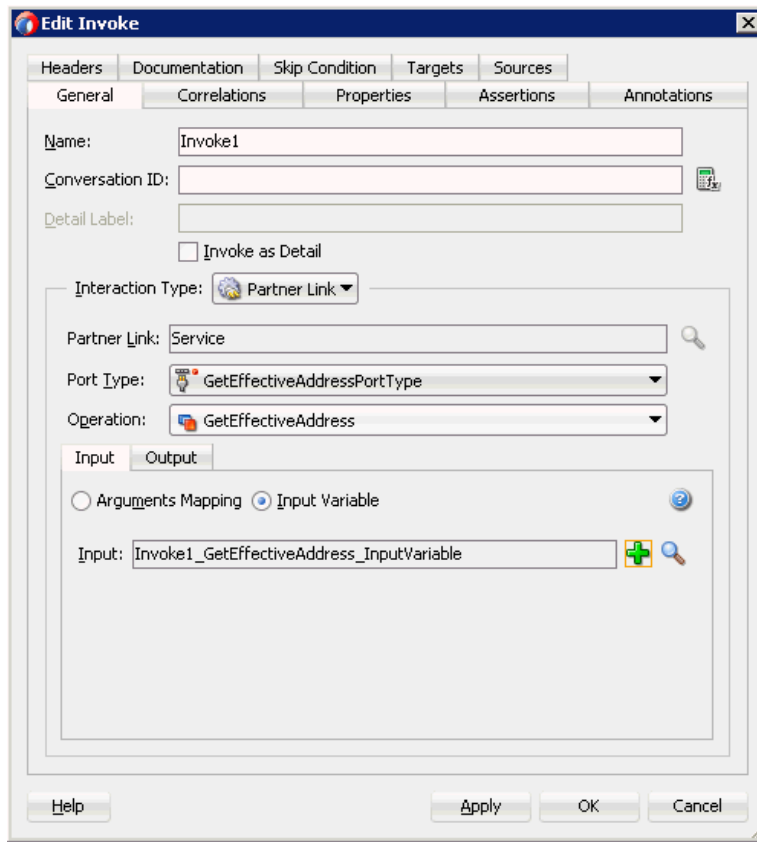
15. 新規入力変数の構成用の「入力」フィールドの右の**プラス記号**アイコンをクリックします。

「変数の作成」ダイアログが表示されます。

16. 新規の入力変数に示されたデフォルト値を受け入れ、「**OK**」をクリックします。

図 4-28 に示すように、「起動の編集」ダイアログに戻ります。

図 4-28 「起動の編集」ダイアログ



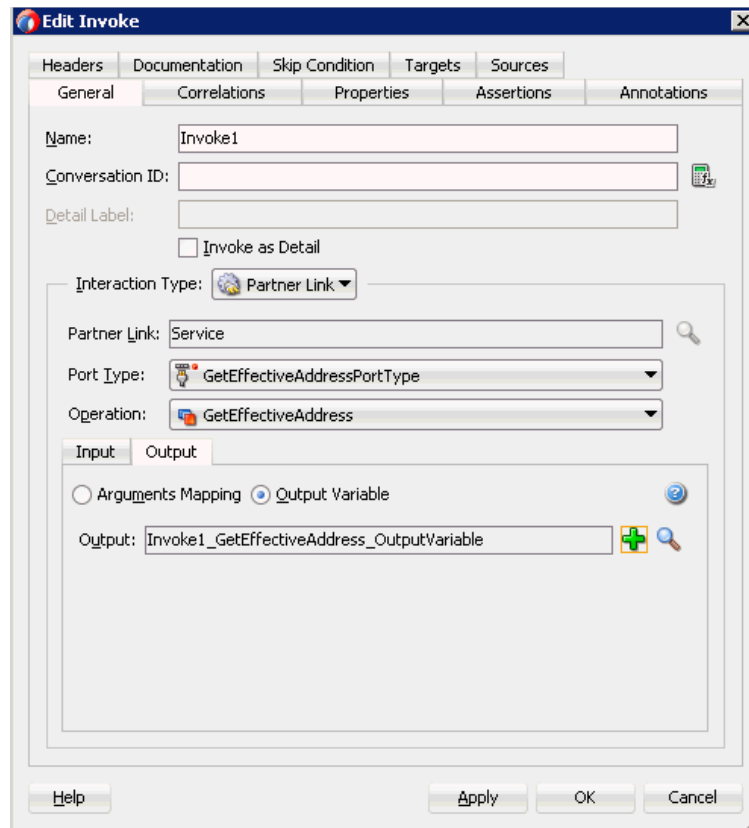
17. 「出力」タブを選択し、新規出力変数の構成用の「出力」フィールドの右の**プラス記号**アイコンをクリックします。

「変数の作成」ダイアログが表示されます。

18. 新規の出力変数に示されたデフォルト値を受け入れ、「**OK**」をクリックします。

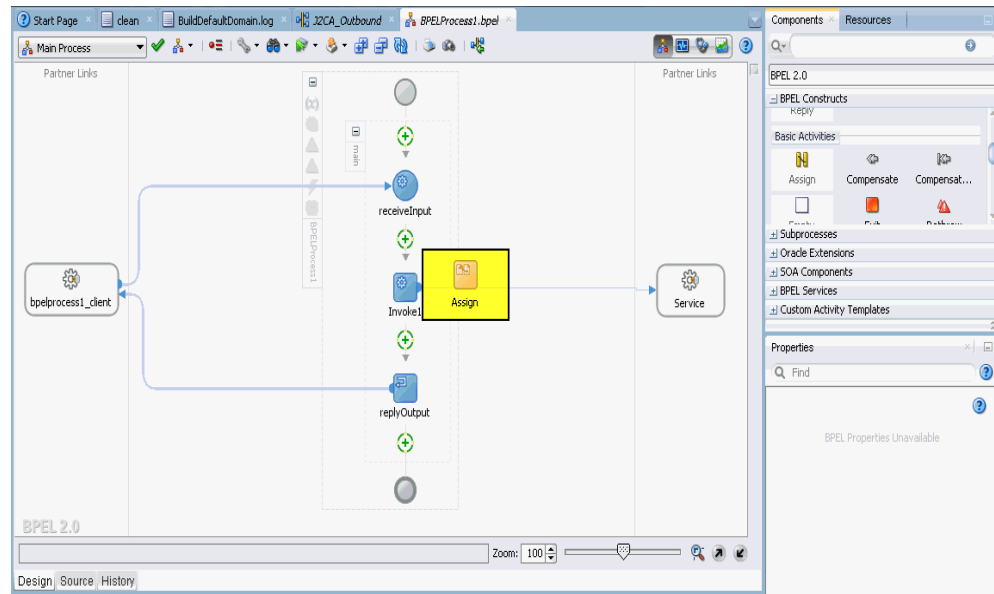
図 4-29 に示すように、「起動の編集」ダイアログに戻ります。

図 4-29 「起動の編集」ダイアログ



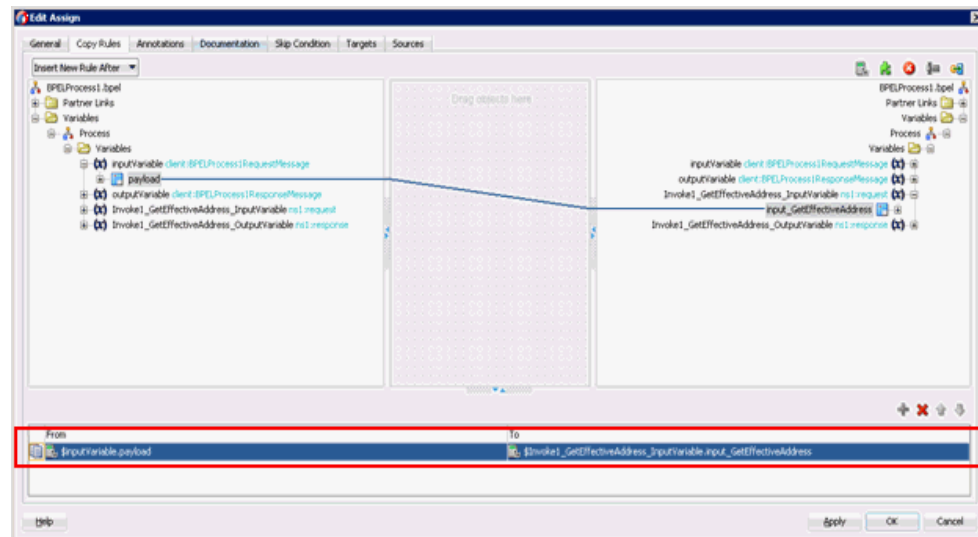
19. 「適用」、「OK」の順にクリックします。
20. 図 4-30 に示すように、「BPEL コンストラクト」の「基本アクティビティ」の下の「Assign」アクティビティを「コンポーネント」ペインにドラッグ・アンド・ドロップし、「Receive」アクティビティ・コンポーネント (receiveInput) と「Invoke」アクティビティ・コンポーネント (Invoke1) の間に置きます。

図 4-30 「Assign」 アクティビティ・コンポーネント



21. 新規の「Assign」アクティビティ・コンポーネント (**Assign1**) をダブルクリックします。
「割当ての編集」ダイアログが表示されます。
22. 左ペインの「変数」の下で、「InputVariable」を展開し、「payload」を選択します。
23. 右ペインの「変数」の下で、「Invoke1_GetEffectiveAddress_InputVariable」を展開し、「input_GetEffectiveAddress」を選択します。
24. 「payload」変数を「input_GetEffectiveAddress」変数にドラッグしてマップします。
図 4-31 に示すように、マップされた変数は強調表示された領域に移入されます。

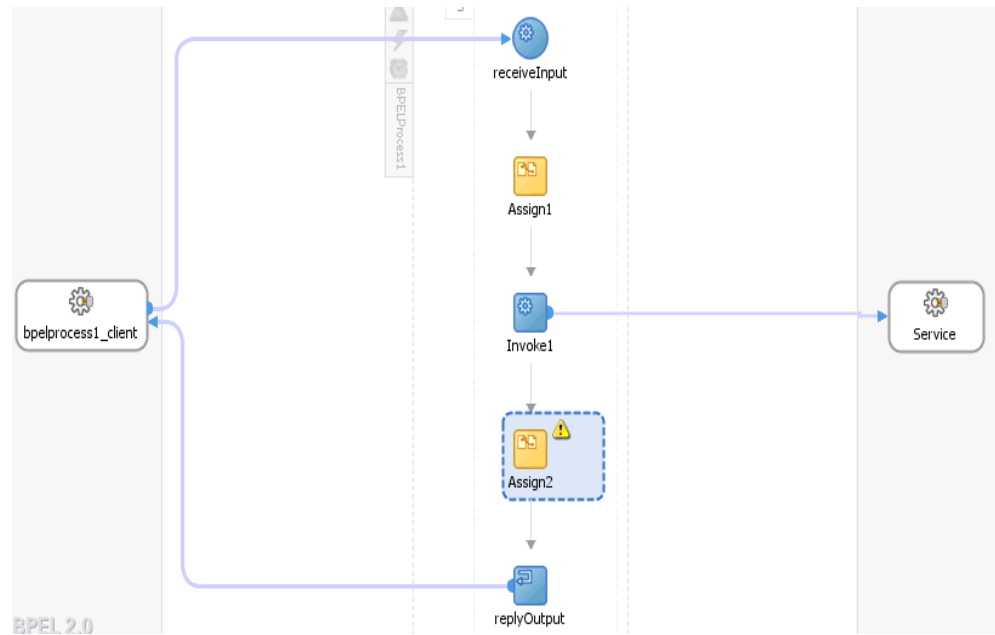
図 4-31 「割当ての編集」ダイアログ



25. 「適用」、「OK」の順にクリックします。

26. 「Assign」 アクティビティ・コンポーネントを「コンポーネント」 ペインにドラッグ・アンド・ドロップし、「Invoke」 アクティビティ (Invoke1) と「Reply」 アクティビティ (replyOutput) の間に置きます。
27. 図 4-32 に示すように、新規の「Assign」 アクティビティ・コンポーネント (Assign2) をダブルクリックします。

図 4-32 新規の「Assign」 アクティビティ・コンポーネント

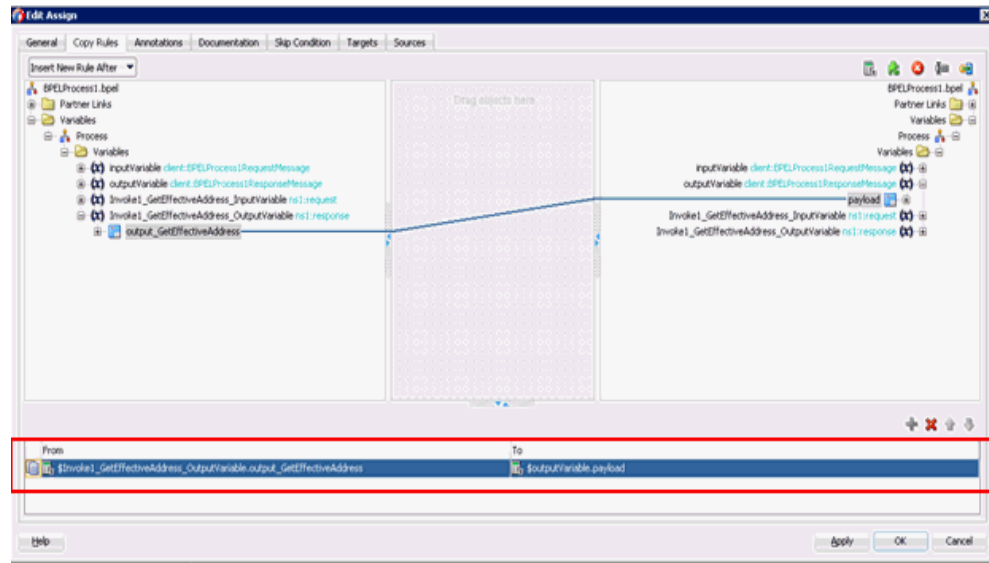


「割当ての編集」 ダイアログが表示されます。

28. 左ペインの「変数」の下で、「Invoke1_GetEffectiveAddress_OutputVariable」を展開し、「output_GetEffectiveAddress」を選択します。
29. 右ペインの「変数」の下で、「outputVariable」を展開し、「payload」を選択します。
30. 「output_GetEffectiveAddress」変数を「payload」変数にドラッグしてマップします。

図 4-33 に示すように、マップされた変数は強調表示された領域に移入されます。

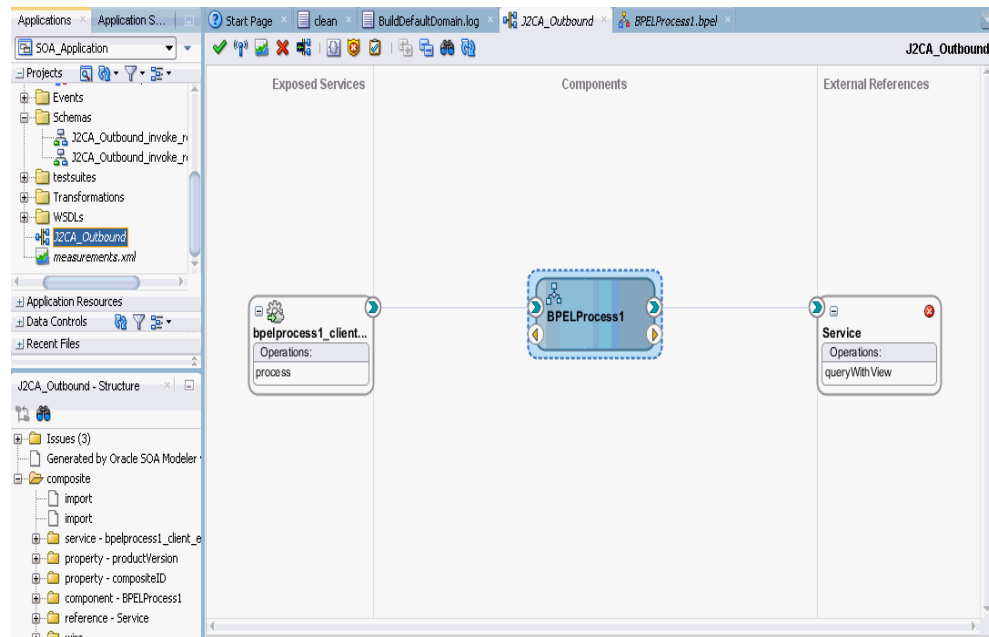
図 4-33 「割当ての編集」ダイアログ



31. 「適用」、「OK」の順にクリックします。

図 4-34 に示すように、「アクティビティ」コンポーネント・ペインに戻ります。

図 4-34 「アクティビティ」コンポーネント・ペイン



32. メニュー・バーで「すべて保存」アイコンをクリックし、構成された新規のアウトバウンド BPEL プロセス・コンポーネントを保存します。

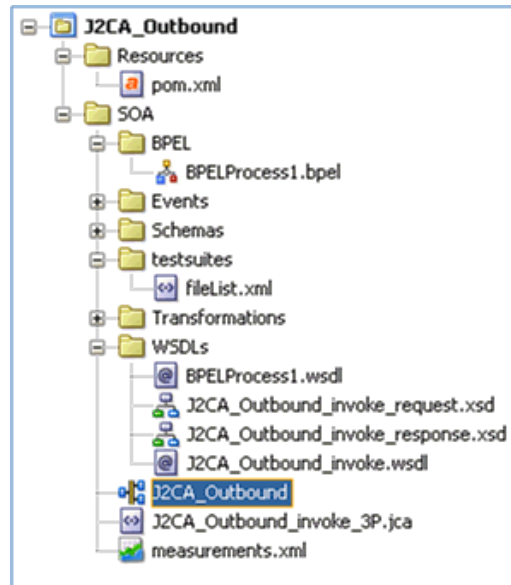
これで、BPEL アウトバウンド・プロセスをデプロイする準備ができました。

4.4.3.3 12c でのデプロイメントに関する既知の問題への対応

12c でのデプロイメントに関する既知の問題に対応するには、次のステップを実行します。

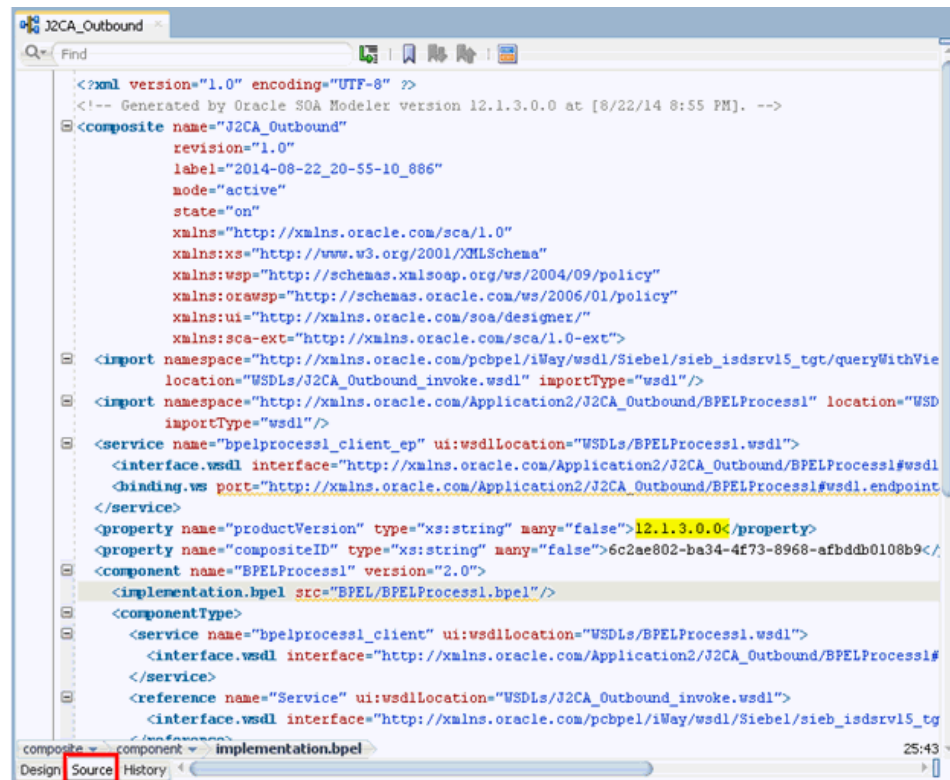
1. 図 4-35 に示すように、作成されたプロセスの「J2CA_Outbound」(作成された BPEL プロセス)をダブルクリックします。

図 4-35 「J2CA_Outbound」ノード



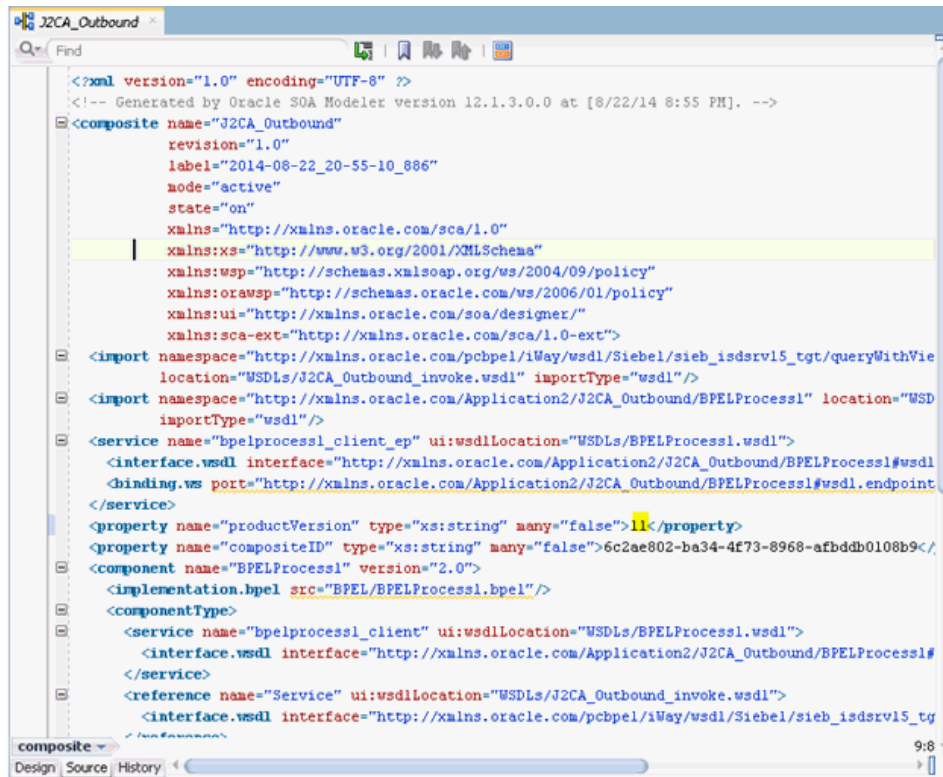
2. 図 4-36 に示すように、開いているプロセスの下の「ソース」タブをクリックします。

図 4-36 「ソース」タブ



3. 図 4-37 に示すように、「productVersion」プロパティ値を「12.1.3.0.0」から「11」に変更します。

図 4-37 プロパティ値



4. 変更を保存し、プロジェクトのデプロイに進みます。

4.4.4 BPEL アウトバウンド・プロセスのデプロイ

BPEL アウトバウンド・プロセスをデプロイするには、次のステップを実行します

1. 図 4-38 に示すように、左ペインでプロジェクト名を右クリックし、「デプロイ」を選択して、「J2CA_Outbound」をクリックします。

図 4-38 「J2CA_Outbound」オプション

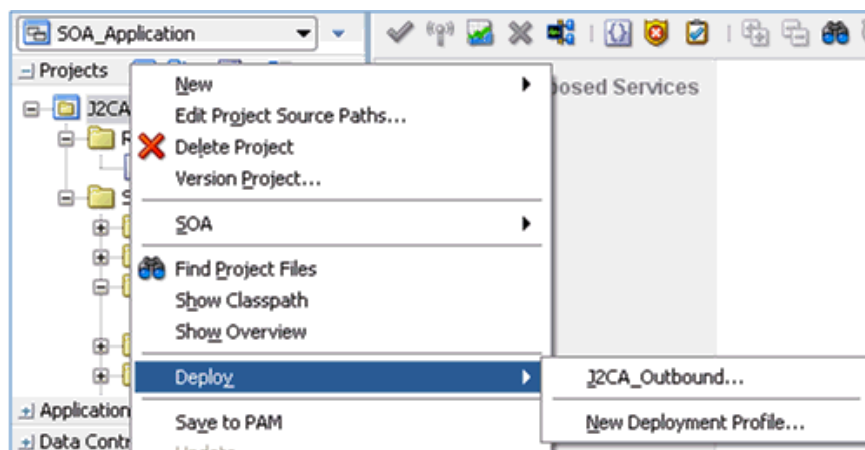
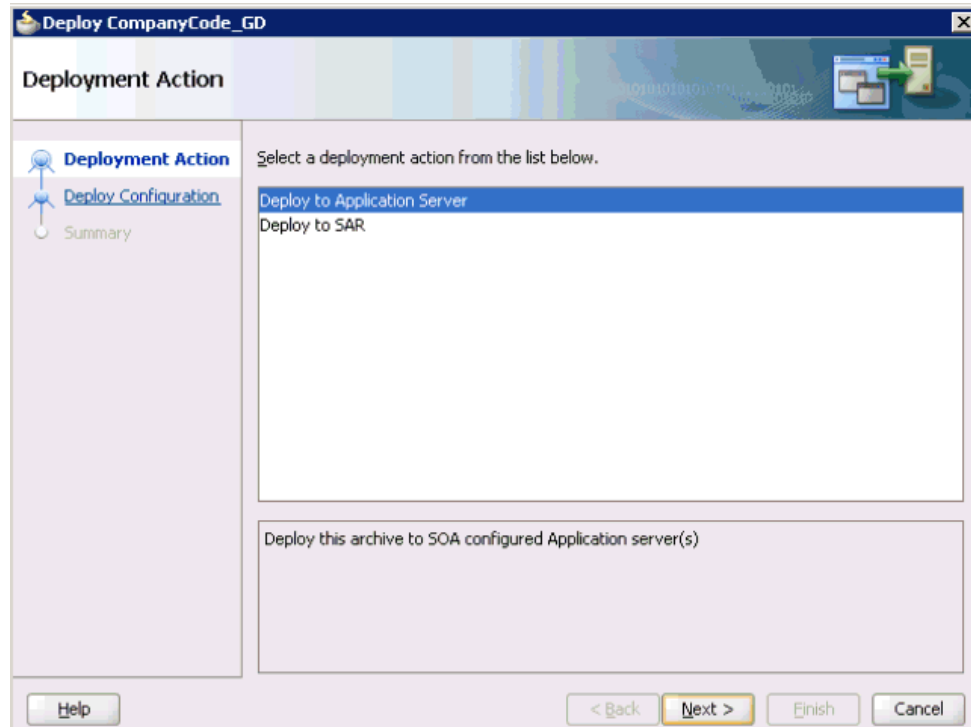


図 4-39 に示すように、「デプロイメント・アクション」ページが表示されます。

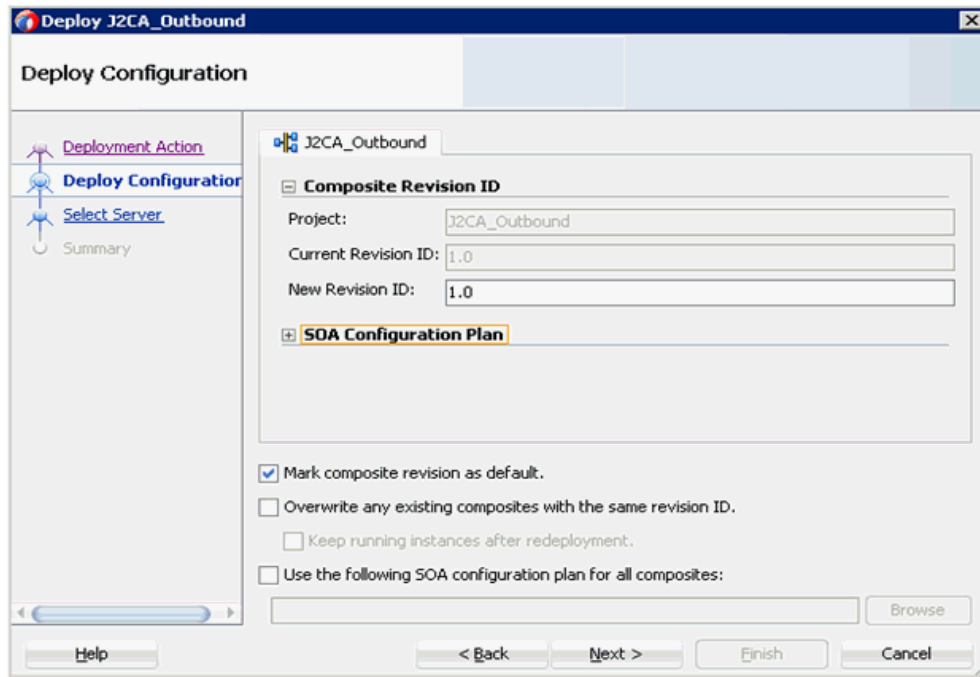
図 4-39 「デプロイメント・アクション」ページ



2. 「アプリケーション・サーバーにデプロイ」が選択されていることを確認します。
3. 「次へ」をクリックします。

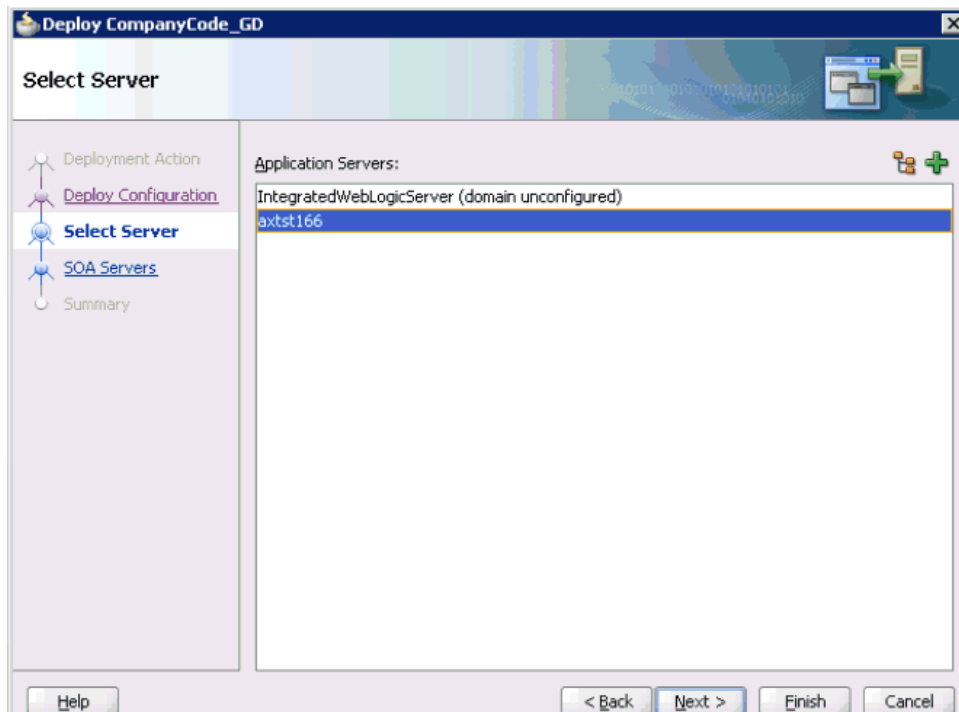
図 4-40 に示すように、「構成のデプロイ」ページが表示されます。

図 4-40 「構成のデプロイ」 ページ



4. デフォルト値を選択したままにし、「次へ」をクリックします。
☒ 4-41 に示すように、「サーバーの選択」ページが表示されます。

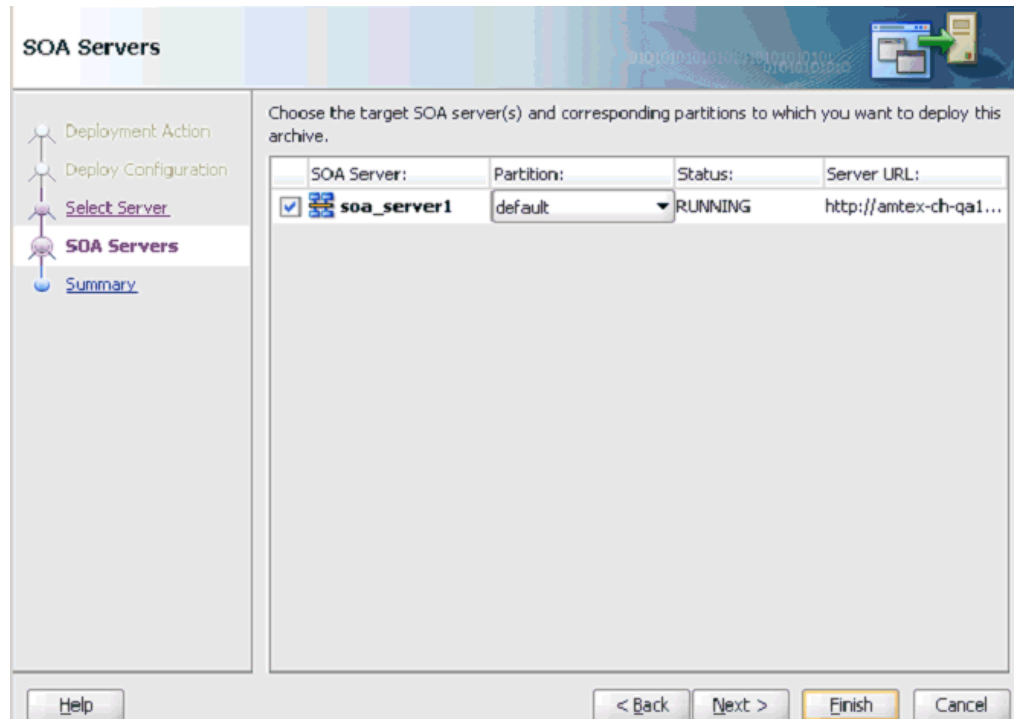
図 4-41 「サーバーの選択」 ページ



5. 構成済の使用可能なアプリケーション・サーバーを選択し、「次へ」をクリックします。

図 4-42 に示すように、「SOA サーバー」ページが表示されます。

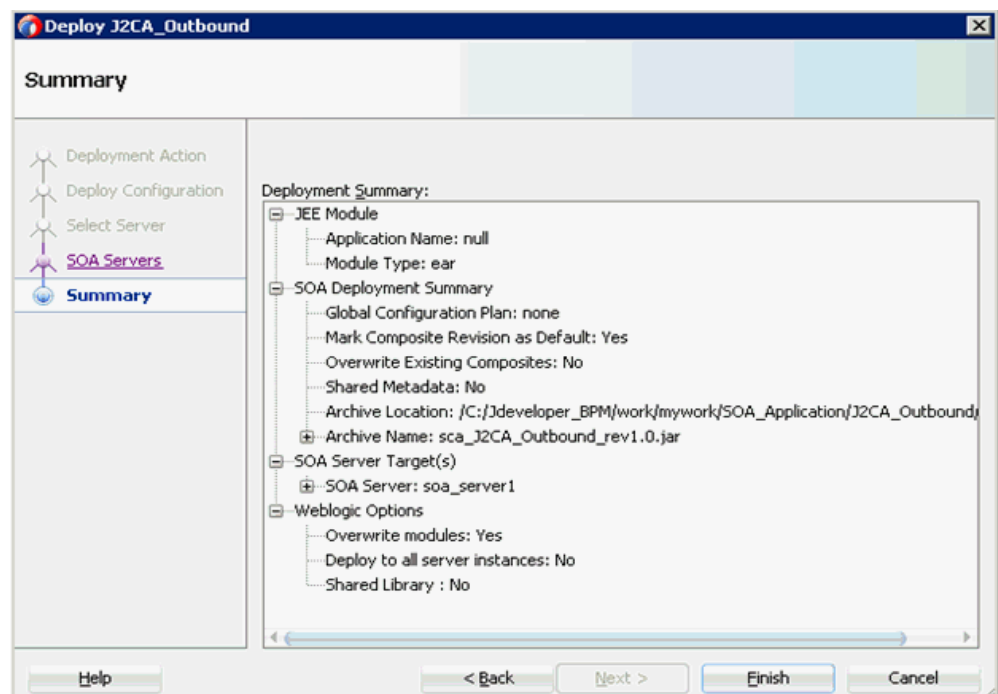
図 4-42 「SOA サーバー」ページ



6. ターゲットの SOA サーバーを選択し、「次へ」をクリックします。

図 4-43 に示すように、「サマリー」ページが表示されます。

図 4-43 「サマリー」ページ



- プロジェクトに使用可能なすべてのデプロイメント情報をレビューおよび確認し、「終了」をクリックします。

図 4-44 に示すように、プロセスが正常にデプロイされます。

図 4-44 デプロイメントの成功を伝えるメッセージ



4.4.5 Oracle Enterprise Manager Console での入力 XML ドキュメントの起動

Oracle Enterprise Manager Console で入力 XML ドキュメントを起動するには、次のステップを実行します

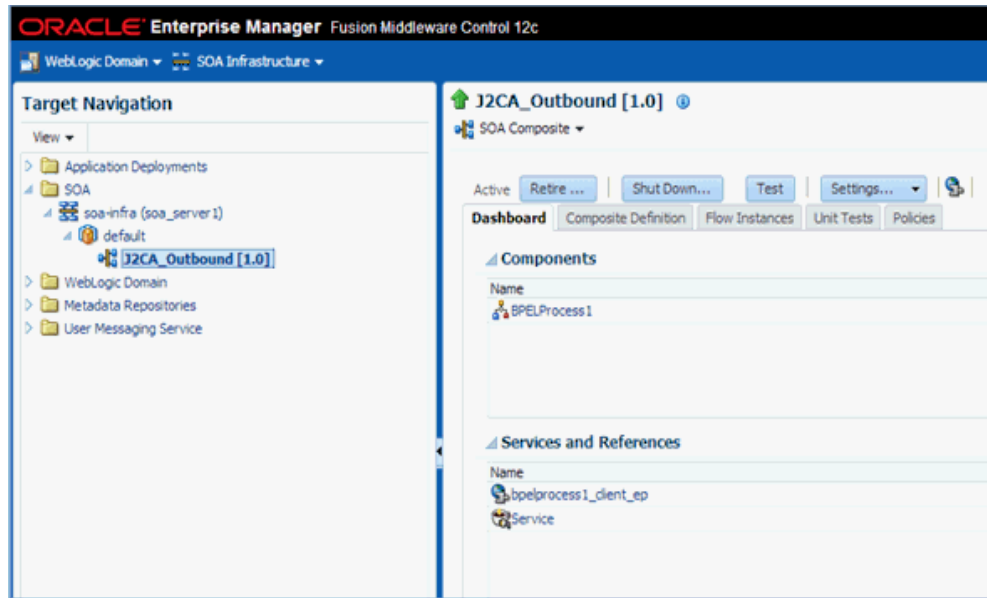
- Oracle Enterprise Manager Console にログオンします。

注意: 12c (12.2.1.1.0) を使用しているお客様は、次のステップを実行します。

- 左ペインの「ターゲット・ナビゲーション」をクリックし、「SOA」を展開し、「soa-infra (soa_server1)」を選択します。
 - 「デプロイ済コンポジット」タブをクリックすると、デプロイされたすべてのコンポジットが表示されます。使用可能なプロジェクト (「J2CA_Outbound」など) をクリックします。
- この手順のステップ 4 にスキップします。
-

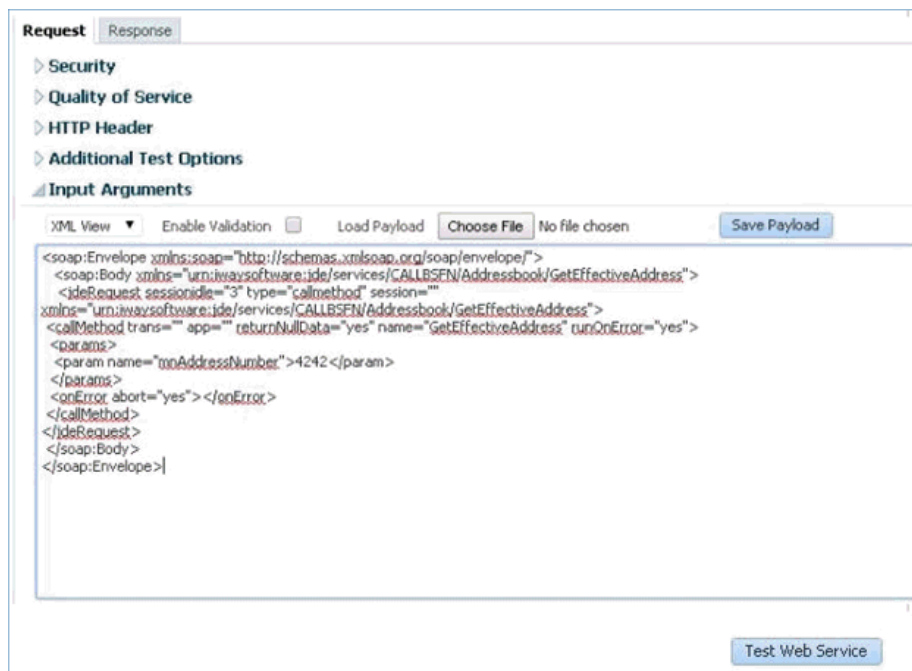
- 「SOA」を展開し、「soa-infra (soa_server1)」を選択して、「デフォルト」をクリックします。
- 図 4-45 に示すように、使用可能なプロジェクト (「J2CA_Outbound」など) を選択し、「テスト」をクリックします。

図 4-45 「テスト」 ボタン



4. 「リクエスト」 タブをクリックします。
5. 図 4-46 に示すように、リストから「XML 表示」を選択します。

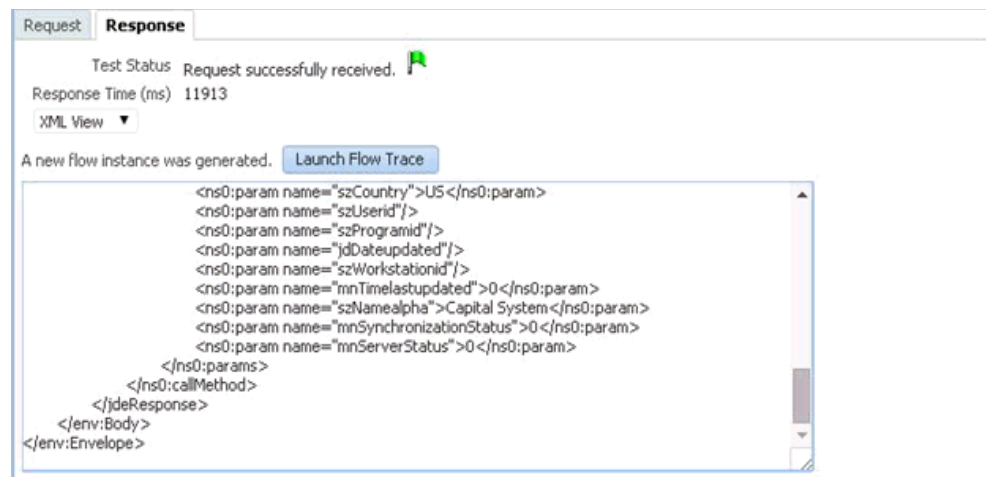
図 4-46 「入力引数」 リスト



6. 「入力引数」領域で適切な入力 XML ドキュメントを指定し、「Web サービスのテスト」をクリックします。

図 4-47 に示すように、Oracle Enterprise Manager Console に出力レスポンスが表示されます。

図 4-47 受信した出力のレスポンス



4.4.6 アウトバウンド BPEL およびアウトバウンド・メディエータ・プロセスのテスト

Oracle Enterprise Manager Console でアウトバウンド BPEL プロセスまたはアウトバウンド・メディエータ・プロセスをテストするとき、これらのコンソールで生成された XML エンベロープは使用しないでください。代わりに、これらを削除し、ネームスペース修飾の WSDL に準拠する、スキーマから生成された XML ペイロードを使用します。

Enterprise Manager Console を使用してメディエータ・データ・フローをテストできます。メディエータ・データ・フローおよび相互作用を作成すると、Web サービスが作成され、Oracle Application Server に登録されます。メディエータ・アウトバウンド・プロセスの作成の詳細は、第 5 章「Oracle SOA Suite でのメディエータ・サービス・コンポーネントとの統合」を参照してください。

4.5 イベント統合のためのインバウンド BPEL プロセスの設計 (J2CA 構成)

この項は、Oracle Application Adapter for J.D. Edwards OneWorld を J.D. Edwards OneWorld と統合してイベント・データを受信する方法を示しています。デザインタイムおよびランタイムの構成手順については、次の各項で説明します。

このインバウンド・ユース・ケース・シナリオ用のサンプル・プロジェクトが、次に示す Application Adapter のインストール・フォルダに用意されています。

```
<ADAPTER_HOME>\etc\sample\JDEdwards_Samples.zip\JDEdwards_Samples\BPEL\J2CA\Inbound_Project
```

アダプタのデザインタイム構成を完了するには、次のツールが必要です。

- Oracle Adapter アプリケーション・エクスプローラ (アプリケーション・エクスプローラ)
- Oracle JDeveloper BPEL デザイナ (JDeveloper)

注意： この章に記載されている例では、Oracle JDeveloper が使用されています。

この項では、次のトピックについて説明します。

- 4.5.1 項「イベント統合のための WSDL の生成」
- 4.5.2 項「SOA 用の空のコンポジットの作成」
- 4.5.3 項「BPEL インバウンド・プロセスの定義」
- 4.5.4 項「BPEL インバウンド・プロセスのデプロイ」
- 4.5.5 項「J.D. Edwards OneWorld でのイベントのトリガー」

BPEL プロセスを設計する前に、アプリケーション・エクスプローラを使用して、各 WSDL ファイルを生成する必要があります。詳細は、4-35 ページの「イベント統合のための WSDL の生成」を参照してください。

4.5.1 イベント統合のための WSDL の生成

Oracle JDeveloper を使用して BPEL プロセスを設計する前に、アプリケーション・エクスプローラを使用してインバウンド相互作用のための WSDL を生成するとき、各 J2CA イベントに個別のチャンネルを作成し、そのチャンネルを選択する必要があります。

注意： 2つ以上のイベントで同じチャンネルが共有されている場合、正しい BPEL プロセスにイベント・メッセージが配信されないことがあります。

この項では、次のトピックについて説明します。

- 4.5.1.1 項「アプリケーション・エクスプローラでのチャンネルの作成」
- 4.5.1.2 項「イベント通知のための WSDL の生成 (コマンド・プロンプトのみ)」

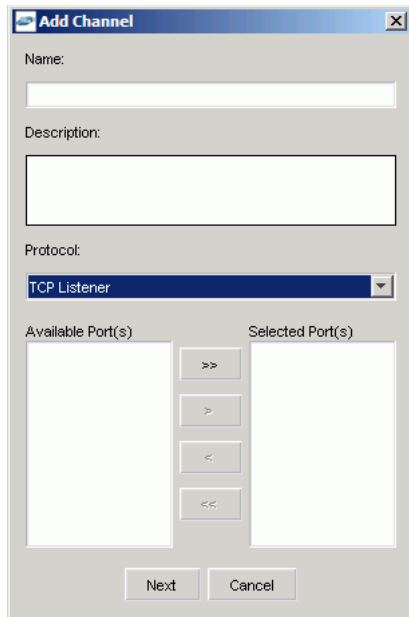
4.5.1.1 アプリケーション・エクスプローラでのチャンネルの作成

チャンネルを作成するには、次のステップを実行します。

1. アプリケーション・エクスプローラで、「JDEdwards」ノードを展開します。
2. 「チャンネル」ノードを右クリックし、「チャンネルの追加」を選択します。

図 4-48 に示すように、「チャンネルの追加」ダイアログが表示されます。

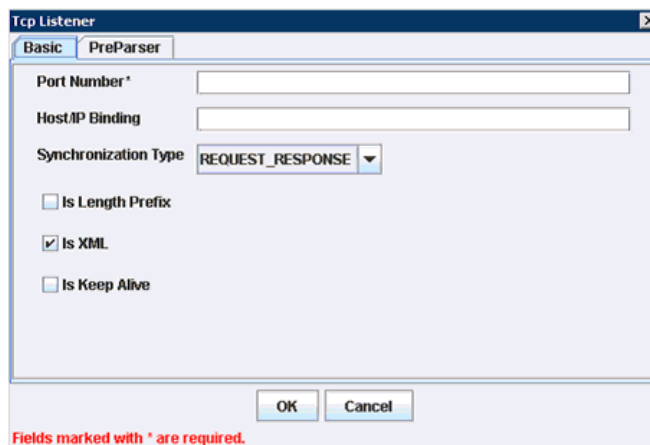
図 4-48 「チャンネルの追加」ダイアログ



3. 「名前」フィールドに、チャンネルのわかりやすい名前を入力します。
4. 「説明」フィールドに説明を入力します (オプション)。
5. 「プロトコル」リストから、チャンネルのプロトコルを選択します。
6. 「次へ」をクリックします。

図 4-49 に示すように、選択したリスナーのダイアログが表示されます。

図 4-49 「TCP リスナー」ダイアログ



7. 「ポート番号」フィールドに、チャンネルのポート番号を入力します。
8. ホスト /IP バインディング・フィールドに、サーバーの場所を入力します。
9. 「同期タイプ」リストから、同期タイプを選択します。
10. XML フォーマットでないデータを送信するイベントの場合、**接頭辞の長さ**を選択します。TCP/IP イベント・アプリケーションでは、TCP/IP ポートにデータを書き込むとき、データに 4 バイトのバイナリ長フィールドの接頭辞を付ける必要があります。

11. XML フォーマットでデータを戻すイベントの場合、「XML」を選択します。事前パーサーは必要ありません。
12. イベント・トランザクションとチャンネル間の連続通信を維持する場合、**キープ・アライブ**を選択します。
13. 図 4-50 に示すように、**事前パーサー・タブ**をクリックします。

図 4-50 事前パーサー・タブ

表に基づいて、値を入力します。

パラメータ	説明
ユーザー ID*	J.D. Edwards OneWorld の有効なユーザー ID。
ユーザー・パスワード*	ユーザー ID に関連付けられたパスワード。
JDE 環境*	ご使用の J.D. Edwards OneWorld 環境。このパラメータの詳細は、J.D. Edwards OneWorld のドキュメントを参照するか、OneWorld システム管理者にお問い合わせください。
アプリケーション	XMLInterop または J.D. Edwards OneWorld 内のアプリケーション名。オプション。
サーバー IP アドレス*	J.D. Edwards OneWorld が実行されているサーバーの名前。JDEOW などのサーバー名または 123.45.67.89 などの IP アドレスを指定できます。
サーバー・ポート*	6009 など、サーバーがリスニングしているポート番号。
ユーザー・ロール	要件に従って、ユーザー・ロールを定義します。

パラメータ	説明
スキーマの場所	<p>イベント出力から生成された XML スキーマ (.xsd ファイル) の場所。たとえば、次のようになります。</p> <pre><ADAPTER_HOME>\config\configuration_name\schemas\JDEdwards\target_name\jde-schema.xsd</pre> <p>詳細は、4-35 ページの 4.5.1 項「イベント統合のための WSDL の生成」 を参照してください。</p>
スキーマ・スタイル	<p>次のオプションから 1 つ選択します。</p> <ul style="list-style-type: none"> ■ ELEMENT_STYLE(デフォルト) ■ ATTRIBUTE_STYLE

「OK」をクリックします。

チャンネルが作成され、「チャンネル」ノードの下に表示されます。アイコン上の「X」は、そのチャンネルが現在切断されていることを示します。

注意： アプリケーション・エクスプローラで作成されたチャンネルは、BPEL PM Server で管理されています。テストおよびデバッグ目的でチャンネルを起動する場合は、ランタイム前に停止してください。

4.5.1.2 イベント通知のための WSDL の生成 (コマンド・プロンプトのみ)

アプリケーション・エクスプローラを使用して J.D. Edwards OneWorld のイベント通知用に WSDL を生成できません。コマンド・プロンプトから WSDL を生成するには、次のステップを実行する必要があります。

選択したノードがイベントをサポートしている場合のみ、インバウンド J2CA サービスを作成できます。

注意： スキーマ検証オプション(ルート、ネームスペース、スキーマ)は、Oracle Application Adapter for J.D. Edwards OneWorld には適用できません。

J.D. Edwards OneWorld のイベント通知のための WSDL ファイルを生成するには、次のステップを実行します。

1. アプリケーション・エクスプローラを使用して、J.D. Edwards イベント・ノードの下にチャンネルを作成します。
2. チャンネルを開始します。
チャンネルの開始後は、Oracle WebLogic Server を再起動しないでください。
3. J.D. Edwards OneWorld からインバウンド・メッセージを送信します。
4. アダプタのインストール先の次のディレクトリにあるログ・ファイルで、インバウンド・メッセージ・ペイロードを取得します。

```
<ADAPTER_HOME>\config\configuration_name\log\iwaf_jca1500.log
```

または、アプリケーション・エクスプローラの「イベント」ノードの下にある「ファイル」プロトコルを使用してポートを作成でき、イベント・メッセージをファイル・システムに配置します。

5. XMLSpy などのサード・パーティ・ツールによって、前のステップで取得した XML ペイロードを使用して XML スキーマ (.xsd ファイル) を作成します。
6. 生成された XML スキーマ (.xsd ファイル) で、次の変更を実行します。

- a. Schemas-jdedwards-com を検索し、iwaysoftware に置換します。

```
<xs:schema
targetNamespace="urn:Schemas-jdedwards-com:trans.response.JDES00UT"
  xmlns="urn:Schemas-jdedwards-com:trans.response.JDES00UT"
  xmlns:xs=http://www.w3.org/2001/XMLSchema elementFormDefault="qualified">
```

これを次のように変更します。

```
<xs:schema
  targetNamespace="urn:iwaysoftware:trans.response.JDES00UT"
  xmlns="urn:iwaysoftware:trans.response.JDES00UT"
  xmlns:xs=http://www.w3.org/2001/XMLSchema elementFormDefault="qualified">
```

- b. 次の構文を切り取ります。

```
<xs:element name="jdeResponse">
<xs:complexType>
</xs:complexType>
</xs:element>
```

- c. 次の行の前に貼り付けます。

```
<xs:element name="transaction">
```

7. アダプタのインストール先の次のディレクトリに XML スキーマ (.xsd ファイル) をコピーします。

```
<ADAPTER_HOME>\config\configuration_name\schemas\JDEdwards\target_name\
```

注意: アプリケーション・エクスプローラの事前パーサー・タブで、(ステップ7に示したように) スキーマ (.xsd) ファイルの場所を指定することによって、作成されたチャネルを編集します。たとえば、次のようになります。

```
<ADAPTER_HOME>\config\configuration_name\schemas\JDEdwards\target_name\jde-schema.xsd
```

8. コマンド・プロンプトを開き、次のベース・ドメイン・ディレクトリに移動します。

```
<ADAPTER_HOME>\user_projects\domains\base_domain\bin
```

9. **setDomainEnv.cmd** (Windows) または **../setDomainEnv.sh** (UNIX、Linux) を実行します。

10. 同じコマンド・プロンプトで、次のディレクトリに移動します。

```
<ADAPTER_HOME>\tools\iwaeb\bin
```

11. **obadapter.bat** ファイルを実行して、環境を設定します。

12. アダプタのインストール先に基づいて、XML スキーマ (.xsd ファイル) をコピーする次のディレクトリに移動します。

```
<ADAPTER_HOME>\config\configuration_name\schemas\JDEdwards\target_name
```

13. 次のコマンドを入力して、WSDL を生成します。

```
java -Diway.oem=oracle12c
com.iwaysoftware.af.container.tools.wsdl.IWayWSILBrowser adapterhome adapter
target channel schemaPrefix wsdlFileName
```

各要素の説明は次のとおりです。

`adapterhome` は、ApplicationAdapters ホームへのパスです。たとえば、次のようになります。

SOA の場合：

```
<ORACLE_HOME>\soa\soa\thirdparty\ApplicationAdapters
```

OSB の場合：

```
<ORACLE_HOME>\osb\3rdparty\ApplicationAdapters
```

`adapter` は、アダプタの名前です。例：JDEdwards

`target` は、アプリケーション・エクスプローラで作成したアダプタ・ターゲットの名前です。

`channel` は、アプリケーション・エクスプローラで作成したチャンネルの名前です。

`schemaPrefix` は、XSD スキーマの接頭辞です。スキーマ・ファイルは、Java コマンドを実行するディレクトリと同じディレクトリに存在する必要があります。

次のコマンドを実行して、インバウンド WSDL を生成します。

```
java -Diway.oem=oracle12c
com.iwaysoftware.af.container.tools.wsdl.IWayWSILBrowser
C:\12c_SOA\soa\soa\thirdparty\ApplicationAdapters\
JDEdwards JDEConnection JDEchannel jde-schema J2CA_Inbound_receive.wsdl
```

コマンドが実行されると、コマンド・ウィンドウに次のように表示されます。

```
Running Inbound WSDL generation tool...
-> Generating WSDL...
-> Generating files for OEM oracle12c
-> Done.
```

注意： イベント通知サービス用に生成された WSDL ファイルの名前に **_receive** を追加することをお勧めします。これにより、リクエスト/レスポンス・サービス用に生成されたものと簡単に区別できるようになります。

14. アプリケーション・エクスプローラでチャンネルを停止します。

これで、Oracle JDeveloper での BPEL インバウンド・プロセスの定義に必要な最初のステップである、新規の SOA アプリケーションを作成する準備ができました。

4.5.2 SOA 用の空のコンポジットの作成

SOA 用に空のコンポジットを作成するには、次のステップを実行します。

1. 新規の SOA アプリケーションを作成します。
2. 新規の SOA アプリケーションの名前を入力し、「次へ」をクリックします。

「プロジェクトの名前付け」ページが表示されます。

3. プロジェクト名を入力し、「次へ」をクリックします。
「SOA 設定の構成」ページが表示されます。
4. 「コンポジット・テンプレート」リストで「空のコンポジット」を選択し、「終了」をクリックします。
詳細は、4-9 ページの 4.4.2 項「SOA 用の空のコンポジットの作成」を参照してください。

4.5.3 BPEL インバウンド・プロセスの定義

この項では、BPEL インバウンド・プロセスの定義方法について説明します。次のトピックで構成されています。

- 4.5.3.1 項「サード・パーティ・アダプタ・サービス・コンポーネントの作成」
- 4.5.3.2 項「インバウンド BPEL プロセス・コンポーネントの作成」
- 4.5.3.3 項「12c でのデプロイメントに関する既知の問題への対応」

4.5.3.1 サード・パーティ・アダプタ・サービス・コンポーネントの作成

サード・パーティ・アダプタ・サービス・コンポーネントを作成するには、次のステップを実行します。

1. 図 4-51 に示すように、「サービス・アダプタ」ペインから「サード・パーティ・アダプタ」コンポーネントを「公開されたサービス」ペインにドラッグ・アンド・ドロップします。

図 4-51 「サード・パーティ・アダプタ」コンポーネント

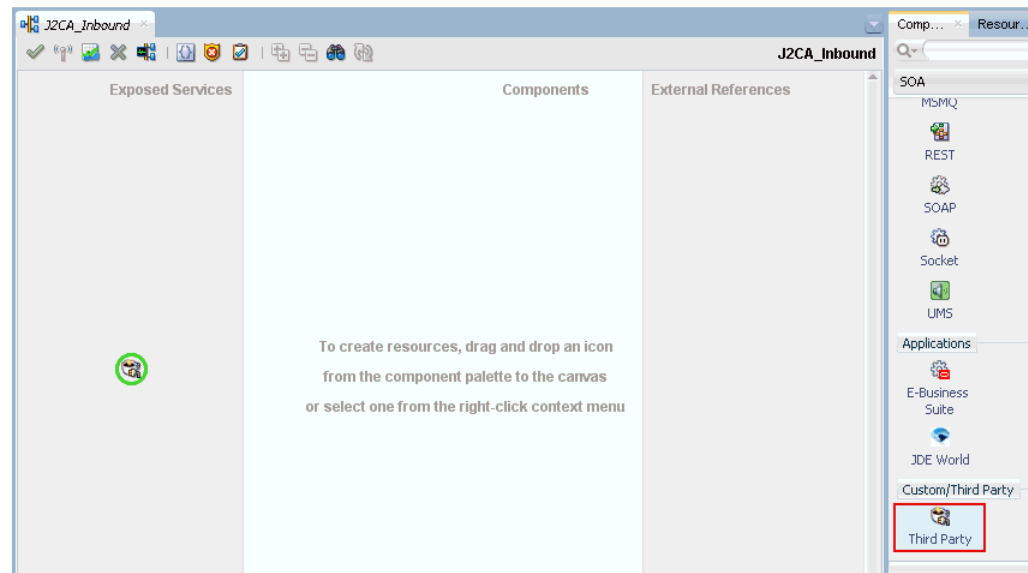
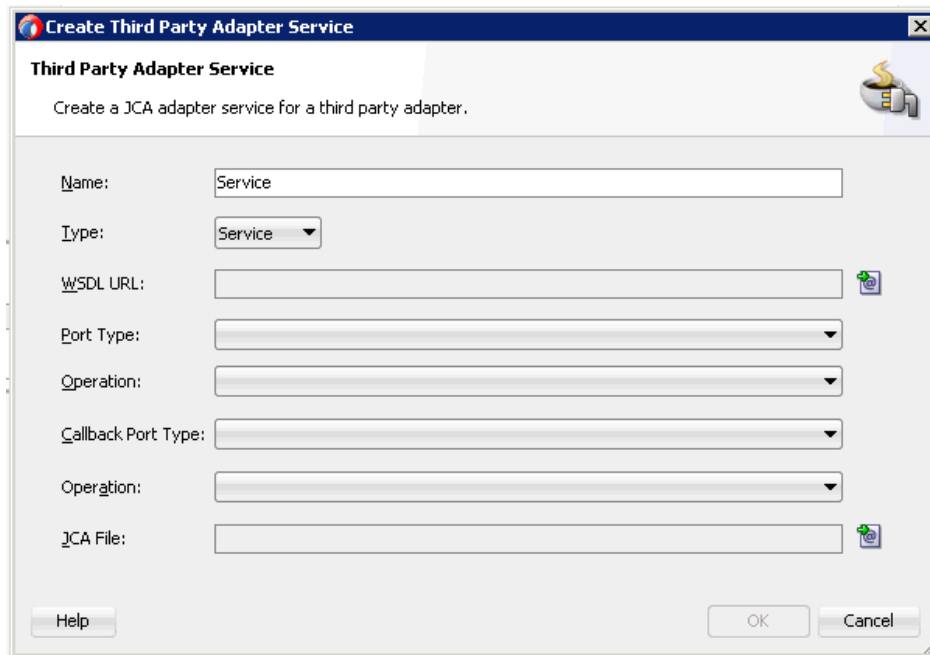


図 4-52 に示すように、「サード・パーティ・アダプタ・サービスの作成」ダイアログが表示されます。

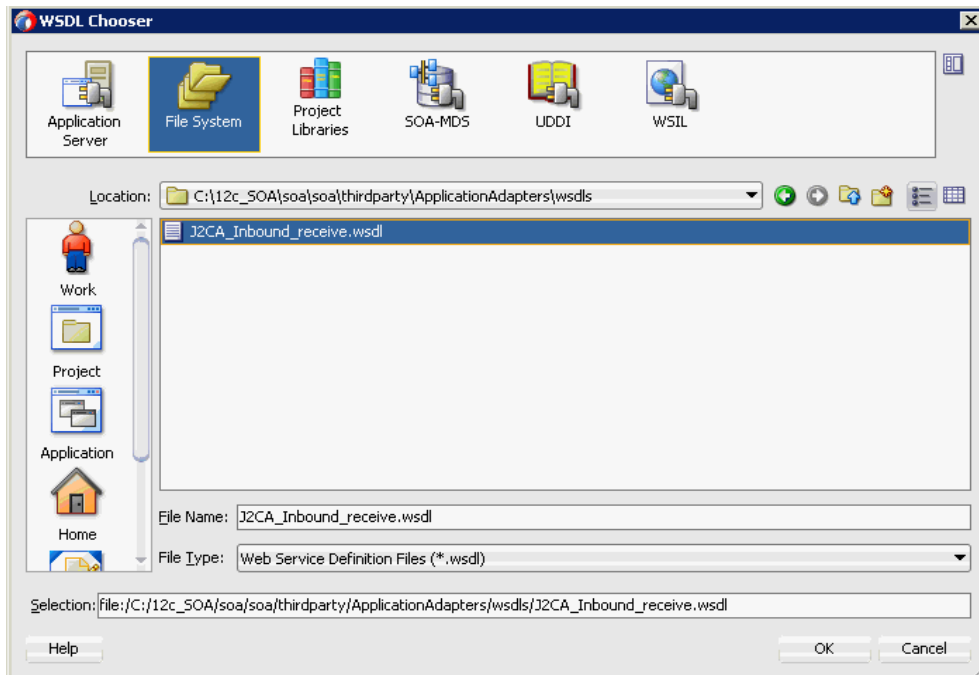
図 4-52 「サード・パーティ・アダプタ・サービスの作成」ダイアログ



2. 「タイプ」リストで「サービス」が選択されていることを確認します (デフォルト)。
3. 「WSDL URL」フィールドの右の「既存の WSDL を検索します。」アイコンをクリックします。

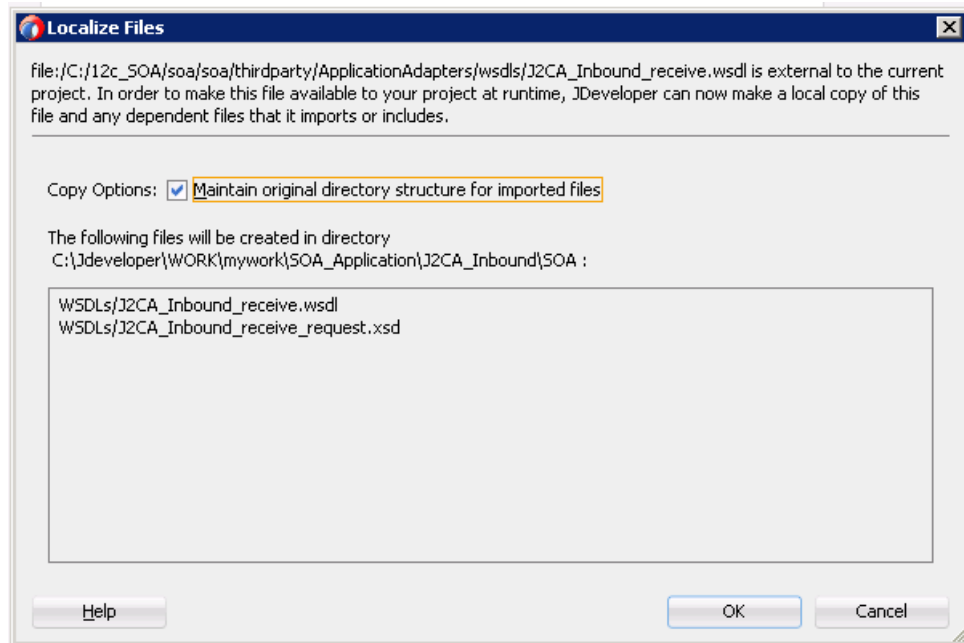
図 4-53 に示すように、「WSDL チューザ」ダイアログが表示されます。

図 4-53 WSDL チューザ」ダイアログ



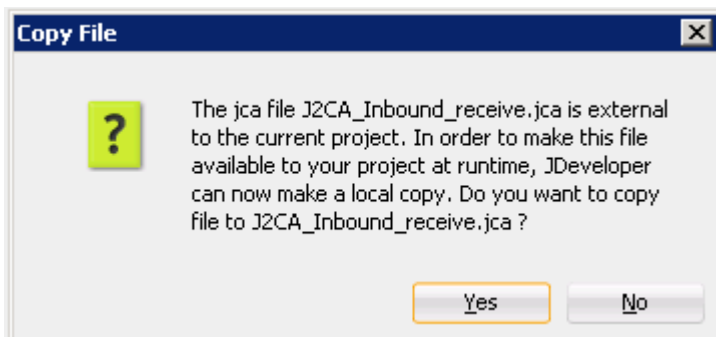
4. 次のディレクトリまでブラウザし、インバウンド WSDL ファイルを選択します。
<ADAPTER_HOME>\wsdls
5. 「OK」をクリックします。
図 4-54 に示すように、「ファイルのローカライズ」ダイアログが表示されます。

図 4-54 「ファイルのローカライズ」ダイアログ



6. 「OK」をクリックします。
インバウンド WSDL ファイルおよび関連リクエスト/レスポンス XML スキーマ・ファイル (.xsd) が、作成済のプロジェクト・フォルダにインポートされます。
「サード・パーティ・アダプタ・サービスの作成」ダイアログに戻ります。
7. 「JCA ファイル」フィールドの右の「JCA ファイルを検索します。」アイコンをクリックします。
変換チューザ・ダイアログが表示されます。
8. 次のディレクトリまでブラウザし、JCA プロパティ・ファイルを選択します。
<ADAPTER_HOME>\wsdls
9. 「OK」をクリックします。
図 4-55 に示すように、「ファイルのコピー」メッセージが表示されます。

図 4-55 「ファイルのコピー」 確認メッセージ

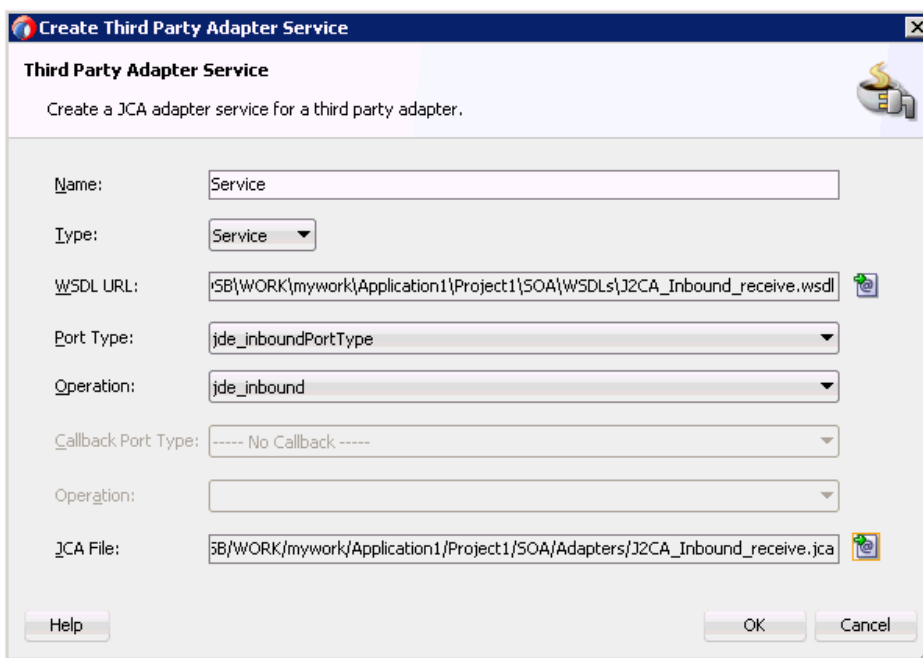


10. 「はい」 をクリックします。

JCA プロパティ・ファイルのコピーがプロジェクト・フォルダ内に作成されます。

図 4-56 に示すように、「サード・パーティ・アダプタ・サービスの作成」 ダイアログログに戻ります。

図 4-56 「サード・パーティ・アダプタ・サービスの作成」 ダイアログ



11. 「OK」 をクリックします。

サード・パーティ・アダプタ・サービス・コンポーネントが作成され、「公開されたサービス」 ペインに表示されます。

これで、インバウンド BPEL プロセス・コンポーネントを構成する準備ができました。

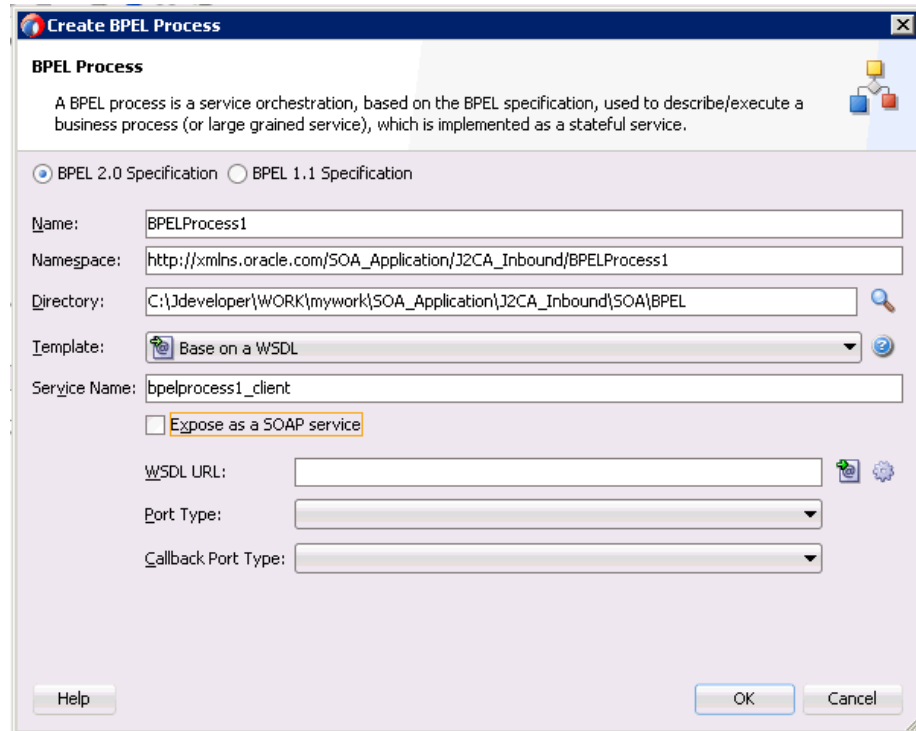
4.5.3.2 インバウンド BPEL プロセス・コンポーネントの作成

インバウンド BPEL プロセス・コンポーネントを作成するには、次のステップを実行します。

1. 「サービス・コンポーネント」 ペインから「BPEL プロセス」 コンポーネントを「コンポーネント」 ペインにドラッグ・アンド・ドロップします。

図 4-57 に示すように、「BPEL プロセスの作成」ダイアログが表示されます。

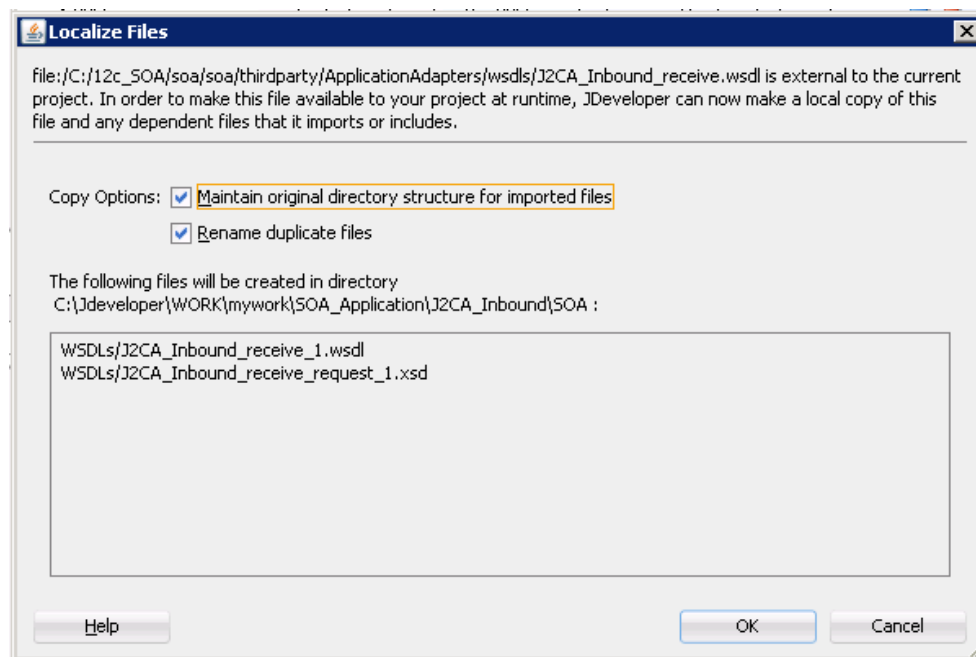
図 4-57 「BPEL プロセスの作成」ダイアログ



2. 「名前」フィールドで、新規のインバウンド BPEL プロセス・コンポーネントの名前を入力するか、デフォルトをそのまま使用します。
デフォルトでは、「BPEL 2.0 仕様」オプションが選択されます。
3. 「テンプレート」リストで「WSDL に基づく」を選択します。
4. 「SOAP サービスとして公開」チェック・ボックスを選択解除します。
5. 「WSDL URL」フィールドの右の「既存の WSDL を検索します。」アイコンをクリックします。
「WSDL チューザ」ダイアログが表示されます。
6. 次のディレクトリからインバウンド WSDL ファイルを選択します。
<ADAPTER_HOME>\wsdl1s
7. 「OK」をクリックします。

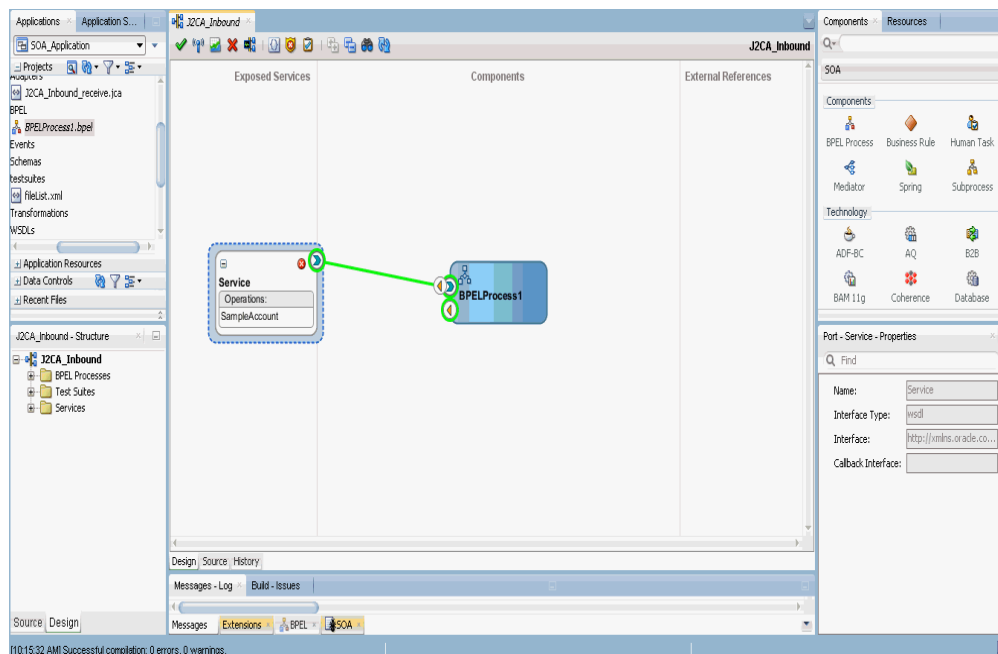
図 4-58 に示すように、「ファイルのローカライズ」ダイアログが表示されます。

図 4-58 「ファイルのローカライズ」 ダイアログ



8. 「重複ファイルの名前変更」 オプションを選択解除します。
9. 「OK」 をクリックします。
「BPEL プロセスの作成」 ダイアログに戻ります。
10. 「OK」 をクリックします。

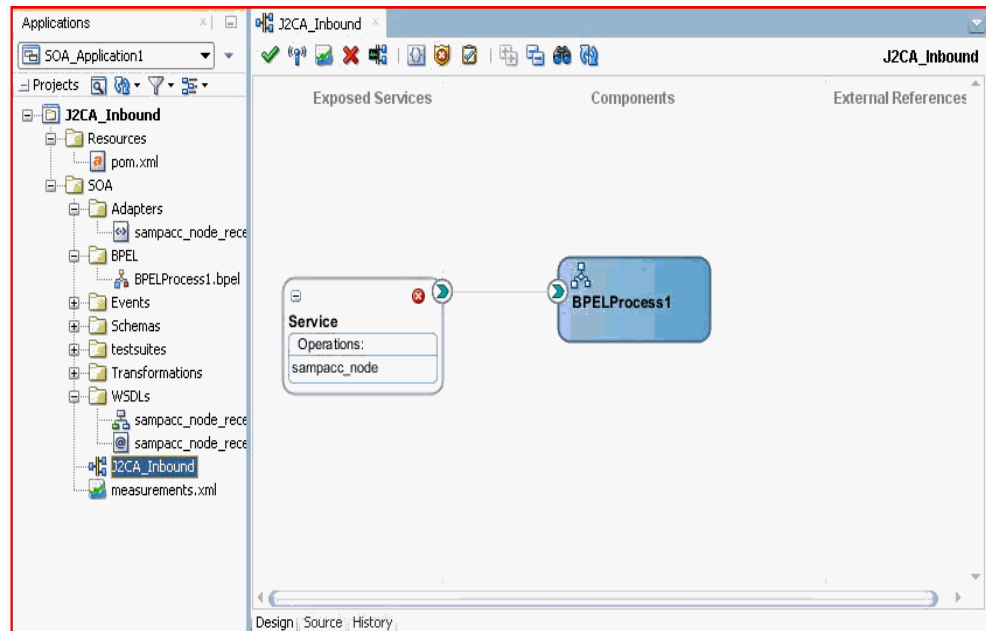
図 4-59 作成された接続



11. 図 4-59 に示すように、サード・パーティ・アダプタ・サービス・コンポーネントとインバウンド BPEL プロセス・コンポーネント間の接続を作成します。

12. 左ペインの「J2CA_Outbound」をダブルクリックします。

図 4-60 「すべて保存」アイコン



13. 図 4-60 に示すように、メニュー・バーで「すべて保存」アイコンをクリックし、構成された新規のインバウンド BPEL プロセス・コンポーネントを保存します。
これで、BPEL インバウンド・プロセスをデプロイする準備ができました。

4.5.3.3 12c でのデプロイメントに関する既知の問題への対応

12c でのデプロイメントに関する既知の問題への対応方法の詳細は、4-26 ページの 4.4.3.3 項「12c でのデプロイメントに関する既知の問題への対応」を参照してください。

4.5.4 BPEL インバウンド・プロセスのデプロイ

BPEL インバウンド・プロセスをデプロイするには、次のステップを実行します

1. 左ペインでプロジェクト名を右クリックし、「デプロイ」を選択して、「J2CA_Inbound」をクリックします。
「デプロイメント・アクション」ページが表示されます。
2. 「アプリケーション・サーバーにデプロイ」が選択されていることを確認します。
3. 「次へ」をクリックします。
「構成のデプロイ」ページが表示されます。
4. デフォルト値を選択したままにし、「次へ」をクリックします。
「サーバーの選択」ページが表示されます。
5. 構成済の使用可能なアプリケーション・サーバーを選択し、「次へ」をクリックします。
「SOA サーバー」ページが表示されます。
6. ターゲットの SOA サーバーを選択し、「次へ」をクリックします。

「サマリー」ページが表示されます。

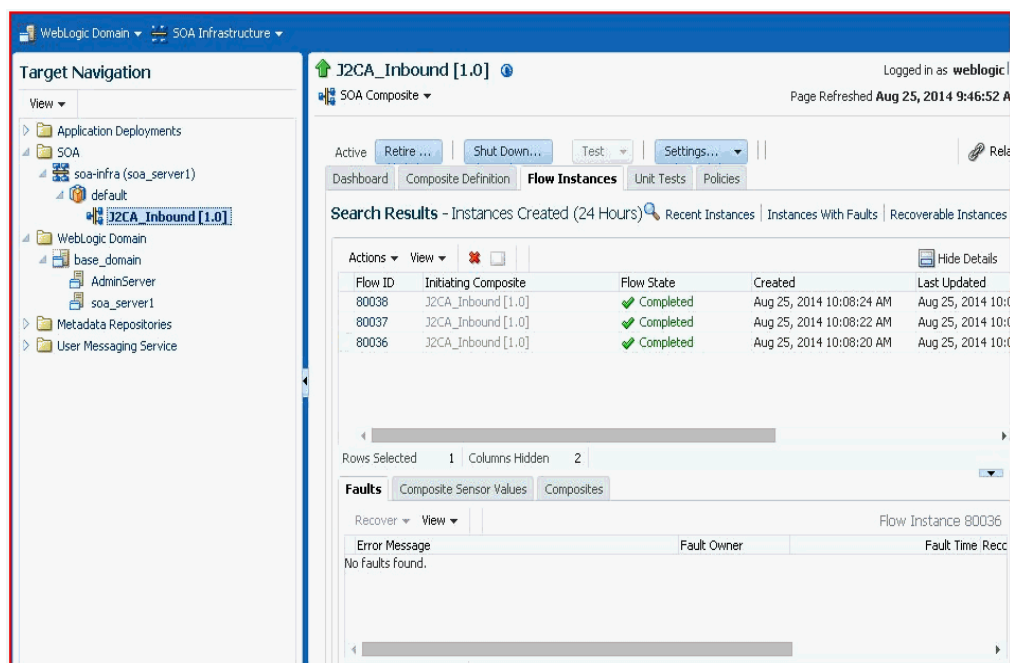
- プロジェクトに使用可能なすべてのデプロイメント情報をレビューおよび確認し、「終了」をクリックします。

プロセスが正常にデプロイされます。

詳細は、4-28 ページの 4.4.4 項「BPEL アウトバウンド・プロセスのデプロイ」を参照してください。

図 4-61 に示すように、イベント・メッセージが J.D. Edwards OneWorld を介してトリガーされると、Oracle Enterprise Manager Console 側でインスタンスが正常に受信されます。

図 4-61 受信されたインスタンス

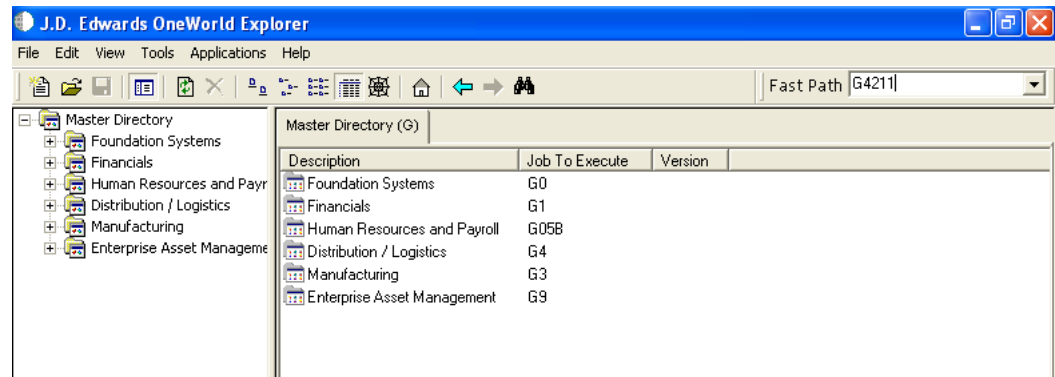


4.5.5 J.D. Edwards OneWorld でのイベントのトリガー

データベースまたはアプリケーション・システムのアクティビティによって、イベントが生成されます。イベントを使用して、アプリケーションのアクションをトリガーできます。J.D. Edwards OneWorld でイベントをトリガーするには、次のステップを実行します。

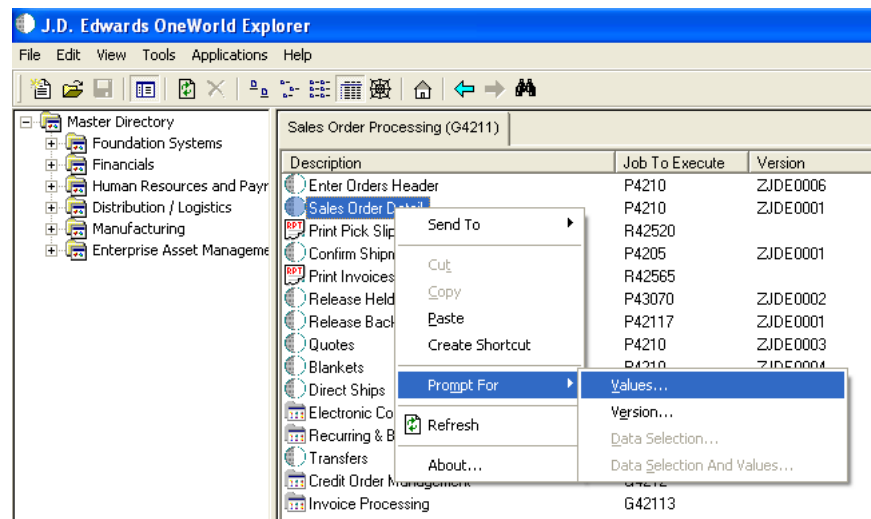
- J.D. Edwards OneWorld システムにログインします。
- 図 4-62 に示すように、J.D. Edwards OneWorld エクスプローラ・ウィンドウのファスト・パス・フィールドで、「G4211」と入力して [Enter] を押します。

図 4-62 J.D. Edwards OneWorld エクスプローラ・ウィンドウ



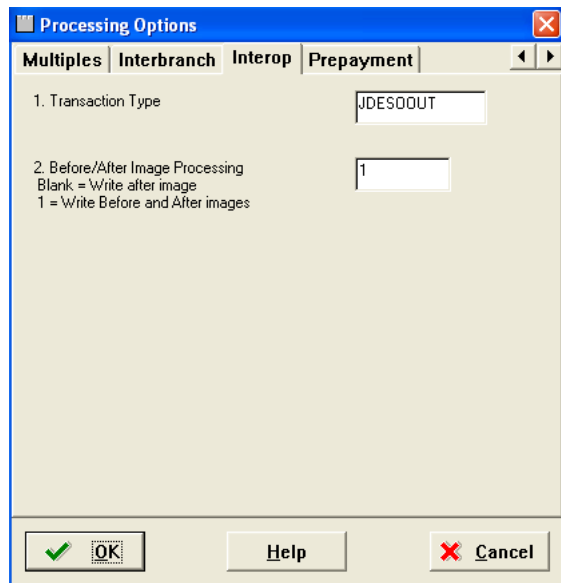
3. 受注オーダー明細 (P4210) を右クリックします。

図 4-63 受注オーダー明細メニュー



4. 図 4-63 に示すように、プロンプト対象、「値」の順に選択します。
 図 4-64 に示すように、処理オプション・ダイアログが表示されます。

図 4-64 処理オプション・ダイアログ

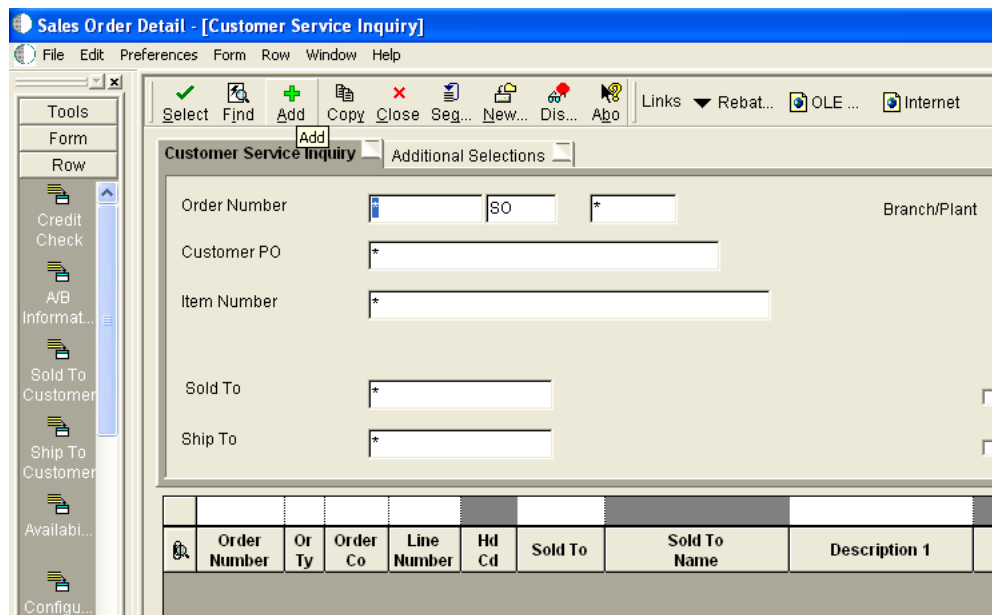


次のステップを実行します。

- a. 相互運用タブをクリックします。
 - b. トランザクション・タイプ・フィールドに「JDES00UT」と入力します。
 - c. イメージの前/後の処理ブランク・フィールドの値が1であることを確認します。
5. 「OK」をクリックします。

図 4-65 に示すように、受注オーダー明細 - (カスタマ・サービス照会) ウィンドウが表示されます。

図 4-65 受注オーダー明細ウィンドウ



6. 「追加」アイコン (左から3番目のアイコン) をクリックします。

7. 図 4-66 に示すように、値を入力します。
別のフィールドに移動するには、キーボードの [Tab] キーを使用します。

図 4-66 値

Quantity Ordered	UoM	Item Number	Ln Ty	Unit Price	Extended Price	Branch/Plant	Location	Lot Number	Lin Num
1			S	0.0000					1

8. 図 4-67 に示すように、「オーダー数量」および「品目番号」の値を入力します。

図 4-67 サンプル値

Quantity Ordered	UoM	Item Number	Ln Ty	Unit Price	Extended Price
1		210	S	0.0000	
			S	0.0000	

9. 図 4-68 に示すように、2 行目の最初のフィールドをクリックし、処理に数秒間を割り当てます。

図 4-68 サンプル値

Quantity Ordered	UoM	Item Number	Ln Ty	Unit Price	Extended Price
1	EA	210	S	5.0000	5.00
			S	0.0000	

10. 「OK」をクリックします。

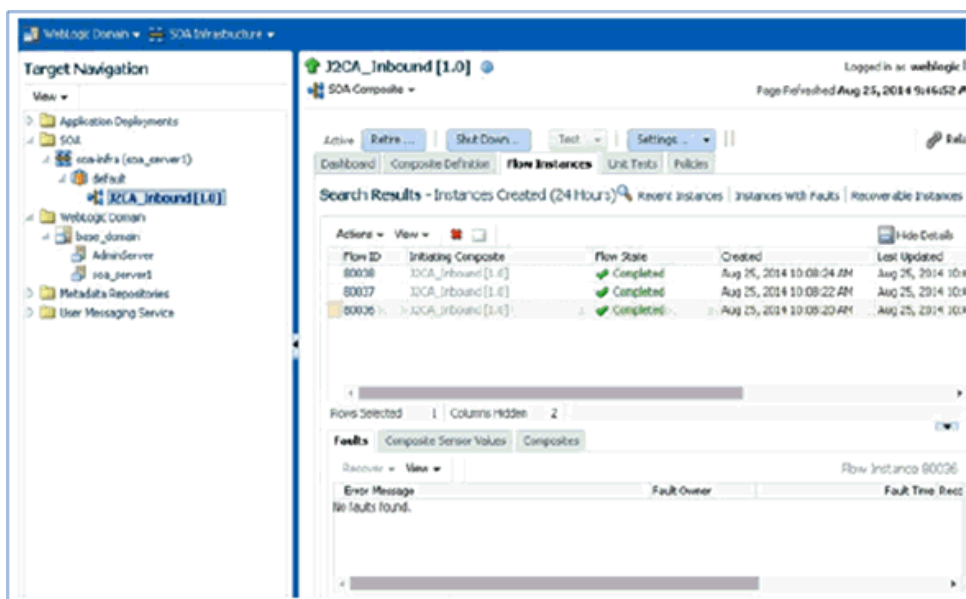
J.D. Edwards OneWorld システムでイベントがトリガーされます。

結果の確認

結果を確認するには、次のステップを実行します。

1. 次の URL を使用して、Oracle Enterprise Manager Console にログインします。
http://localhost:7001/em
2. 「SOA」をクリックし、「soa-infra (soa_server1)」、「default」の順に選択して、「J2CA_Inbound」をクリックします。
3. 「フロー・インスタンス」をクリックします。
図 4-69 に示すように、インスタンスが受信されます。

図 4-69 「フロー・インスタンス」



4.6 サービス統合のためのアウトバウンド BPEL プロセスの設計 (BSE 構成)

この項では、サービス統合のためのアウトバウンド BPM プロセスの設計方法について説明します。

このアウトバウンド・ユース・ケース・シナリオ用のサンプル・プロジェクトが、次に示す Application Adapter のインストール・フォルダに用意されています。

```
<ADAPTER_HOME>\etc\sample\JDEdwards_Samples.zip\JDEdwards_Samples\BPEL\BSE\Outbound_Project
```

アダプタのデザインタイム構成を完了するには、次のツールが必要です。

- Oracle Adapter アプリケーション・エクスプローラ (アプリケーション・エクスプローラ)
- Oracle JDeveloper BPEL デザイナ (JDeveloper)

この項では、次のトピックについて説明します。

- [4.6.1 項「Web サービスを使用したリクエスト/レスポンス型サービス用の WSDL ファイルの生成」](#)
- [4.6.2 項「SOA 用の空のコンポジットの作成」](#)
- [4.6.3 項「BPEL アウトバウンド・プロセスの定義」](#)

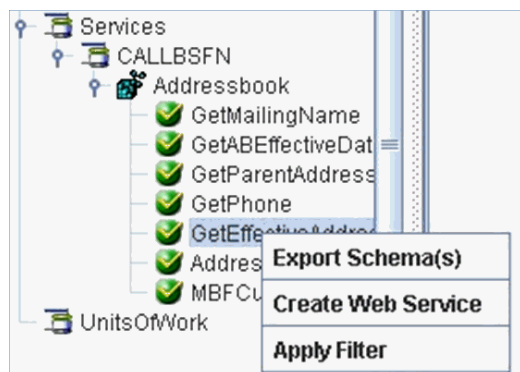
BPEL プロセスを設計する前に、アプリケーション・エクスプローラを使用して、各 WSDL ファイルを生成する必要があります。詳細は、[4.6.1 項「Web サービスを使用したリクエスト/レスポンス型サービス用の WSDL ファイルの生成」](#)を参照してください。

4.6.1 Web サービスを使用したリクエスト/レスポンス型サービス用の WSDL ファイルの生成

Web サービスを使用してリクエスト/レスポンス型サービス用の WSDL ファイルを生成するには、次のステップを実行します。

1. 「アプリケーション・エクスプローラ」を起動し、定義済の J.D. Edwards ターゲット (BSE 構成) に接続します。
ターゲットの定義および J.D. Edwards への接続の詳細は、[2.4.1 項「J.D. Edwards OneWorld に対するターゲットの定義」](#)を参照してください。
2. 接続先の J.D. Edwards ターゲットを展開します。
3. 「サービス」、「CALLBSFN」、アドレス帳の順に展開します。
4. [図 4-70](#) に示すように、「GetEffectiveAddress」を右クリックし、メニューから「Web サービスの作成」を選択します。

図 4-70 「Web サービスの作成」 オプション



「Web サービスの作成」 ダイアログが表示されます。

5. サービス名を入力し、「次へ」をクリックします。
6. 表示される次のダイアログで「OK」をクリックします。

アプリケーション・エクスプローラで、「ビジネス・サービス」ノードに表示が切り替わり、新規の Web サービスが左ペインに表示されます。

7. 新規 Web サービスを右クリックし、メニューから「WSDL の保存」を選択します。
8. WSDL を wsdl フォルダに保存し、「保存」をクリックします。

これで、JDeveloper での BPEL アウトバウンド・プロセスの定義に必要な最初のステップである、SOA 用の空のコンポジットを作成する準備ができました。

4.6.2 SOA 用の空のコンポジットの作成

SOA 用に空のコンポジットを作成するには、次のステップを実行します。

1. 新規の SOA アプリケーションを作成します。
2. SOA アプリケーションの名前を入力し、「次へ」をクリックします。
「プロジェクトの名前付け」ページが表示されます。
3. プロジェクト名を入力し、「次へ」をクリックします。
「SOA 設定の構成」ページが表示されます。
4. 「コンポジット・テンプレート」リストで「空のコンポジット」を選択し、「終了」をクリックします。

詳細は、4-9 ページの 4.4.2 項「SOA 用の空のコンポジットの作成」を参照してください。

4.6.3 BPEL アウトバウンド・プロセスの定義

この項では、BPEL アウトバウンド・プロセス・コンポーネントの構成方法について説明します。

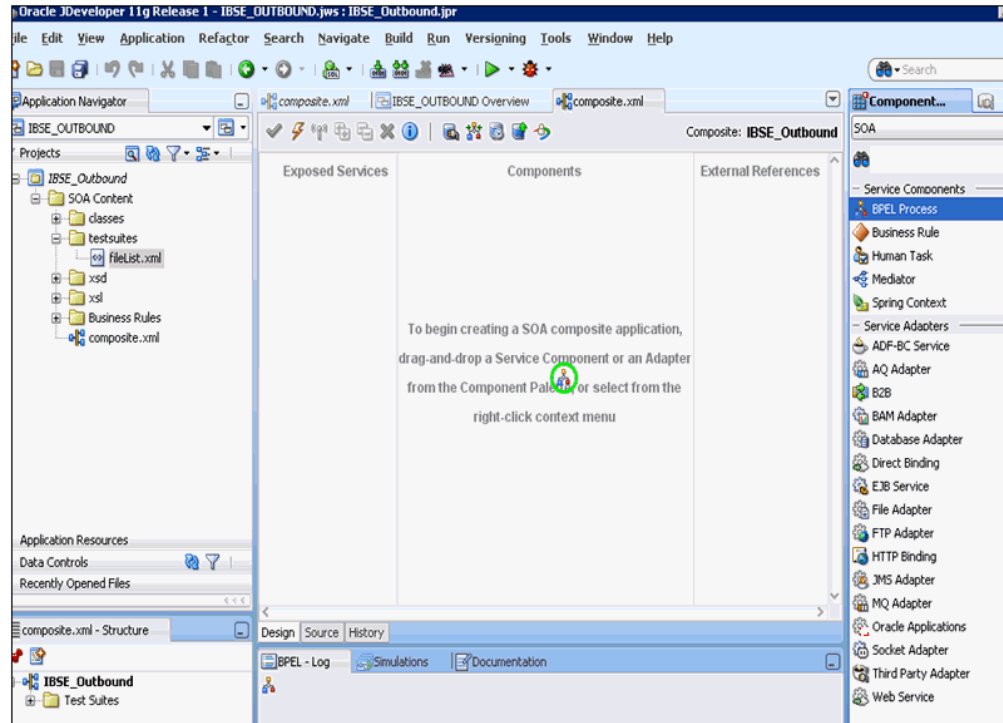
この項では、次のトピックについて説明します。

- 4.6.3.1 項「パートナ・リンクの作成」
- 4.6.3.2 項「作成されたパートナ・リンクを使用した BPEL アクティビティおよびマッピングの作成」

BPEL アウトバウンド・プロセスを定義するには、次のステップを実行します。

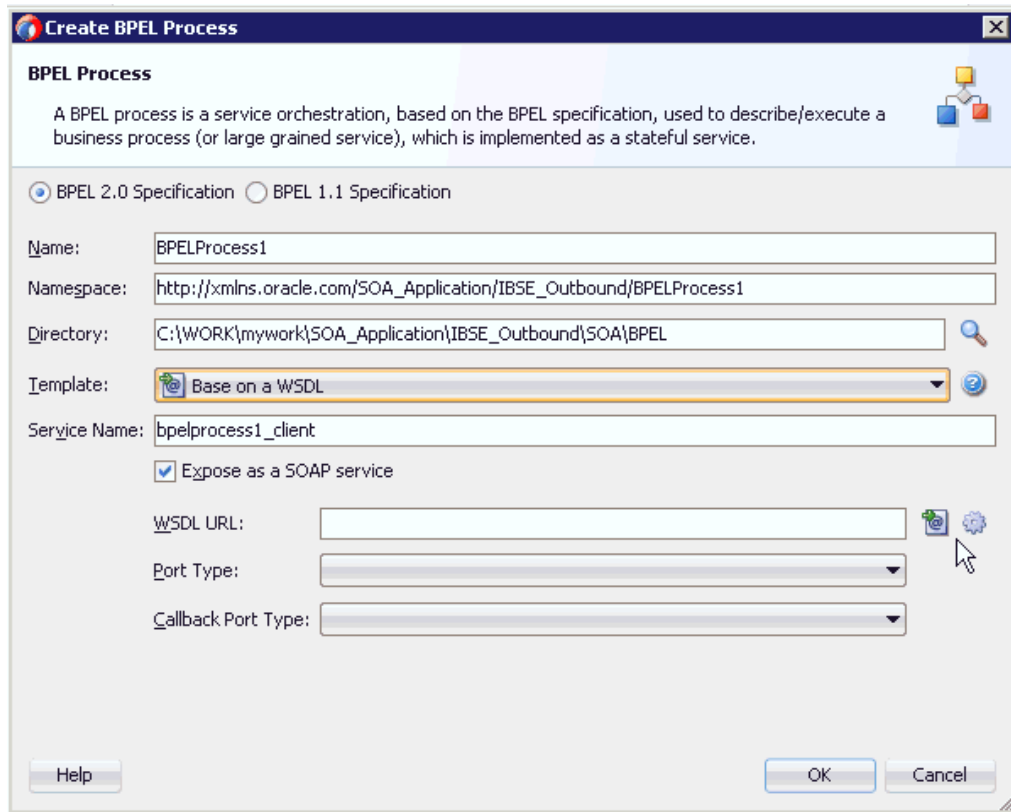
1. 図 4-71 に示すように、「サービス・コンポーネント」ペインから「BPEL プロセス」コンポーネントを「コンポーネント」ペインにドラッグ・アンド・ドロップします。

図 4-71 BPEL プロセス・コンポーネント



2. 「名前」フィールドで、新規のアウトバウンド BPEL プロセス・コンポーネントの名前を入力するか、デフォルト値をそのまま使用します。
デフォルトでは、「BPEL 2.0 仕様」オプションが選択されます。
3. 「テンプレート」ドロップダウン・リストで「WSDL に基づく」を選択します。
4. 図 4-72 に示すように、「WSDL URL」フィールドの右の「既存の WSDL を検索します。」アイコンをクリックします。

図 4-72 「既存の WSDL を検索します。」アイコン



「WSDL チューザ」ダイアログが表示されます。


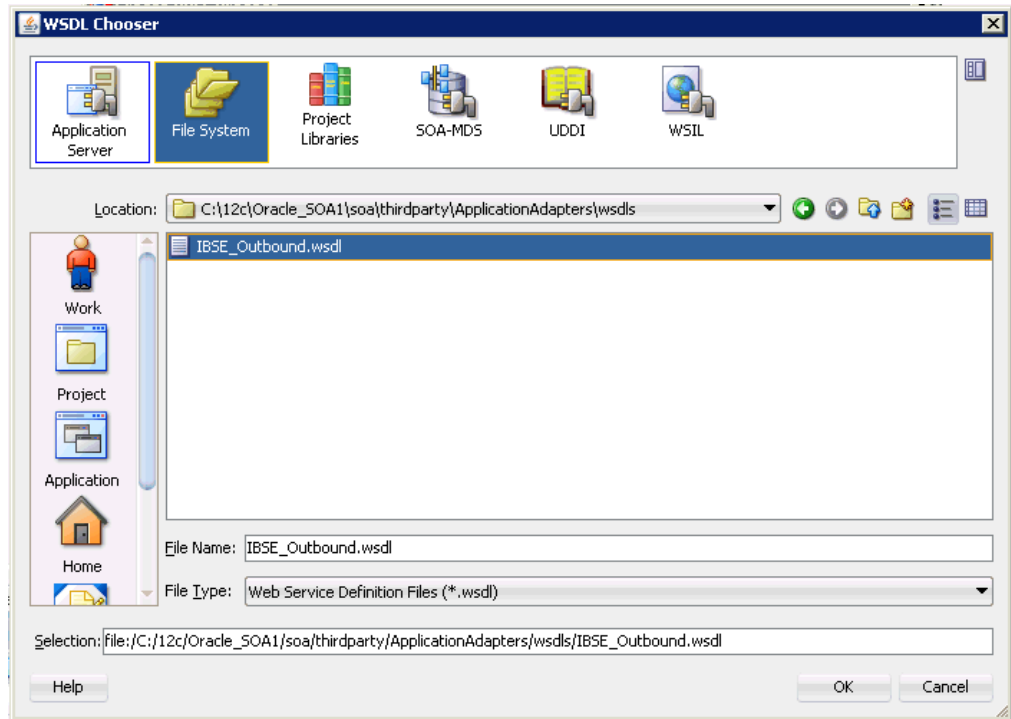
5.  4-73 に示すように、アプリケーション・エクスプローラから WSDL をエクスポートする場所にナビゲートし、WSDL を選択して、「OK」をクリックします。

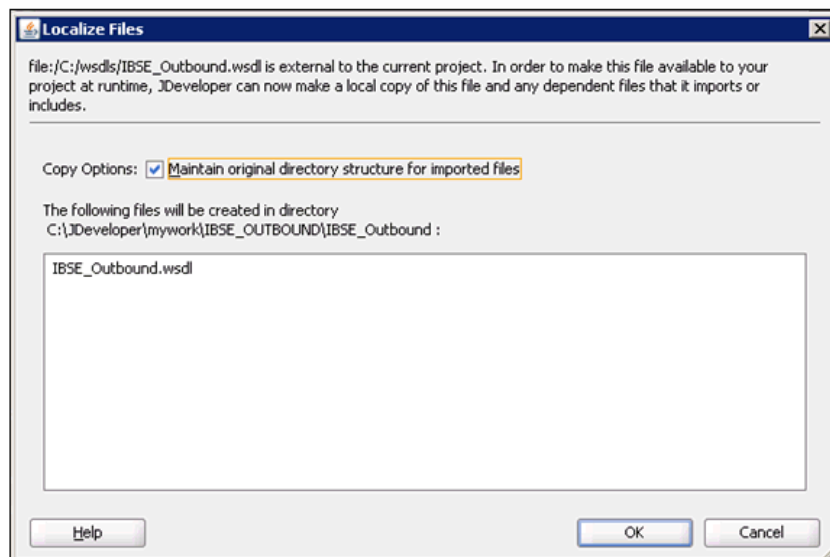
図 4-73 「WSDL チューザ」 ダイアログ



「ファイルのローカライズ」ウィンドウが表示されます。

- 表示された「ファイルのローカライズ」ウィンドウで、「OK」をクリックします。図 4-74 に示すように、これにより、WSDL ファイルがプロジェクト・フォルダにインポートされます。

図 4-74 「ファイルのローカライズ」ウィンドウ



「BPEL プロセスの作成」ウィンドウが表示されます。

- 図 4-75 に示すように、「BPEL プロセス」ペインで、「OK」をクリックします。

図 4-75 「BPEL プロセス」 ペイン

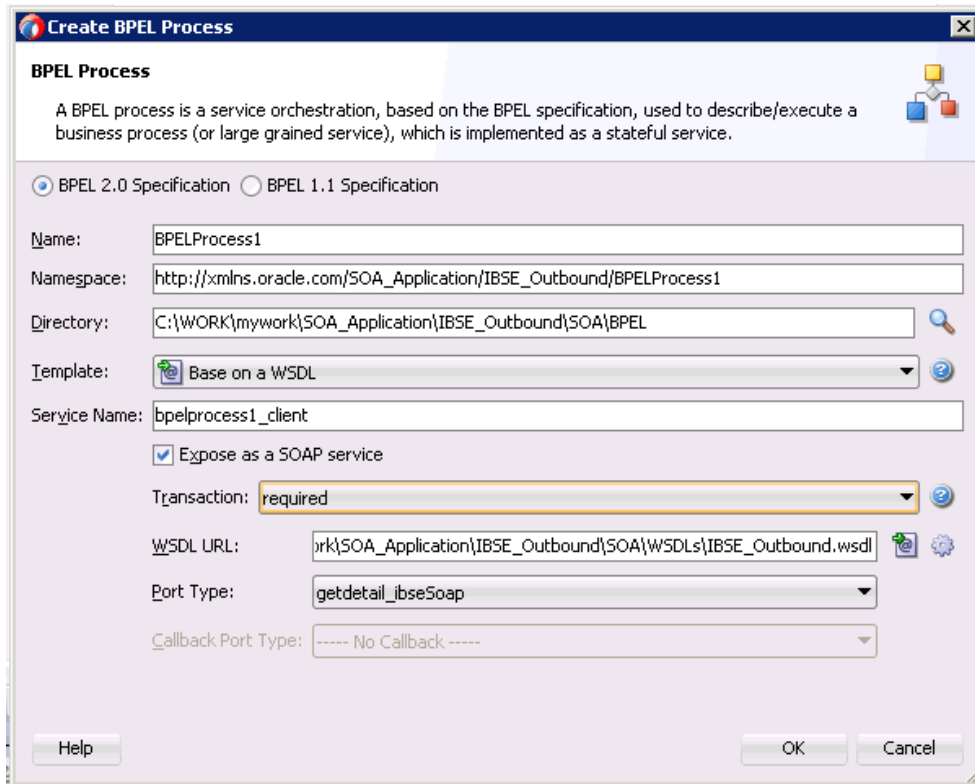
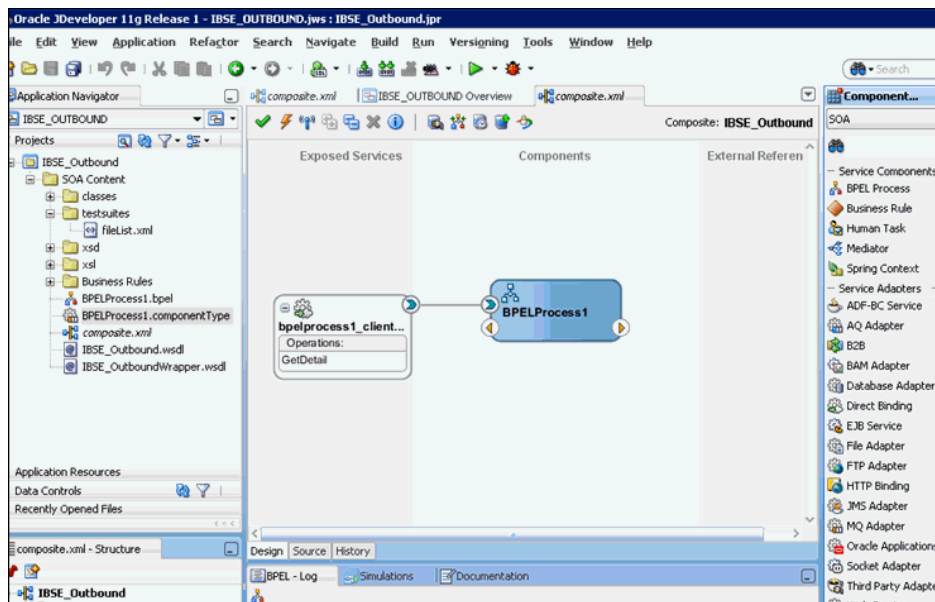


図 4-76 に示すように、BPEL プロセス・コンポーネントが作成されて表示されます。

図 4-76 BPEL プロセス・コンポーネント



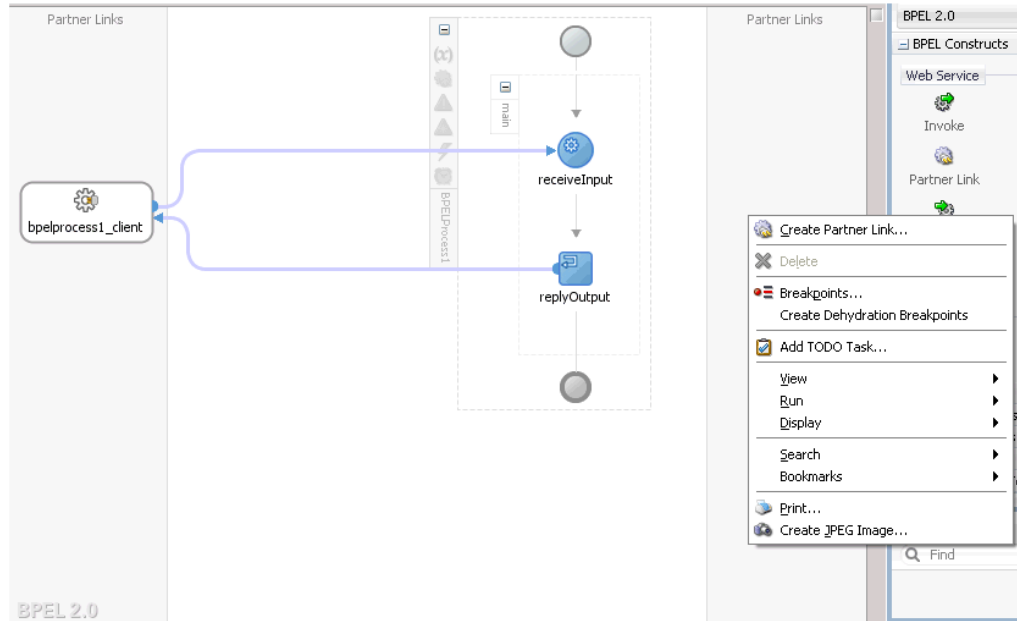
4.6.3.1 パートナ・リンクの作成

この項では、パートナ・リンクを作成する方法について説明します。

パートナー・リンクを作成するには、次のステップを実行します。

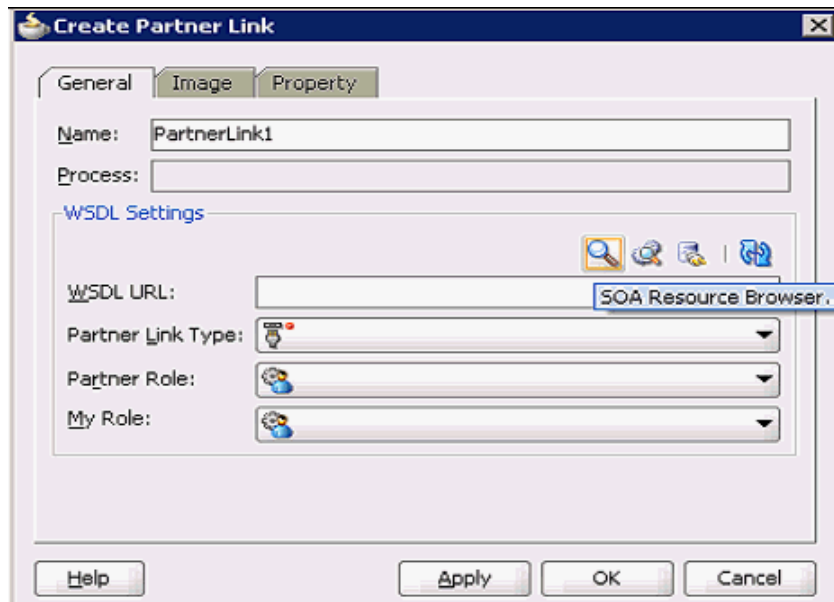
1. 「コンポーネント」ペインで、アウトバウンド BPEL プロセス・コンポーネントをダブルクリックします。
2. 図 4-77 に示すように、「パートナー・リンク」ペインを右クリックし、「パートナー・リンクの作成」を選択します。

図 4-77 パートナ・リンクの作成



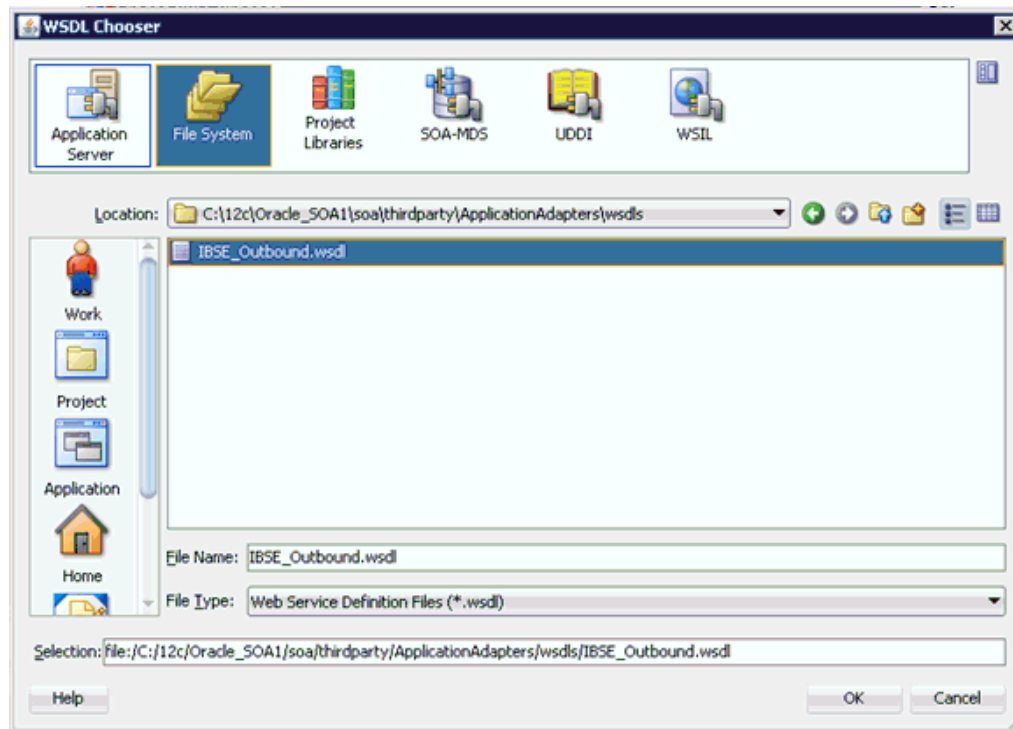
3. 図 4-78 に示すように、表示された「パートナー・リンクの作成」ウィンドウで、適切な名前を指定し、「SOA リソース・ブラウザ」ツールをクリックします。

図 4-78 「SOA リソース・ブラウザ」ツール



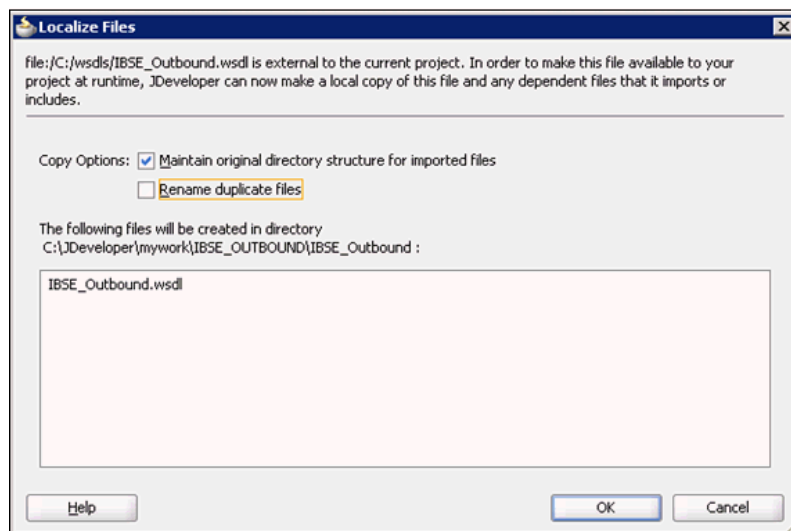
4. 図 4-79 に示すように、表示された「WSDL チューザ」ダイアログで、アプリケーション・エクスプローラから WSDL をエクスポートする場所にナビゲートし、WSDL を選択して、「OK」をクリックします。

図 4-79 WSDL チューザ」ダイアログ



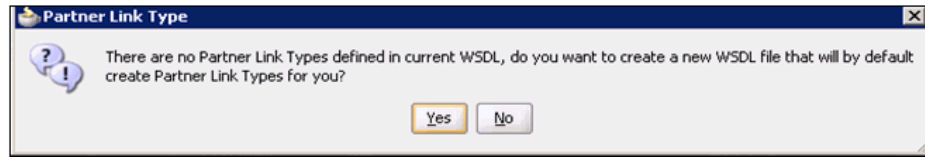
5. 図 4-80 に示すように、表示された「ファイルのローカライズ」ウィンドウで、「重複ファイルの名前変更」チェック・ボックスを選択解除し、「OK」をクリックします。

図 4-80 「ファイルのローカライズ」ウィンドウ



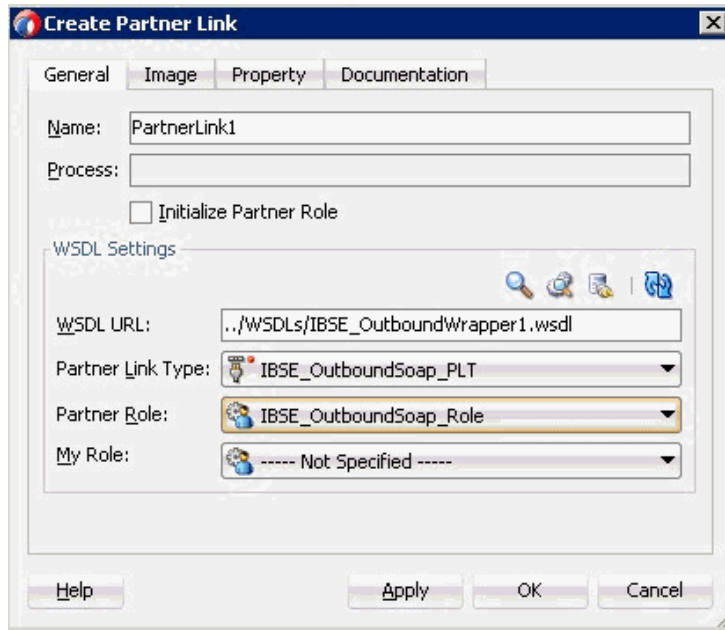
6. 図 4-81 に示すように、表示された「パートナ・リンク・タイプ」ウィンドウで、「はい」をクリックします。

図 4-81 パートナ・リンク・タイプ



7. 表示された「パートナ・リンクの作成」ウィンドウで、「パートナ・ロール」ドロップダウン・リストを展開し、使用可能なパートナ・ロールを選択します。
8. 図 4-82 に示すように、「適用」、「OK」の順にクリックします。

図 4-82 パートナ・リンクの作成



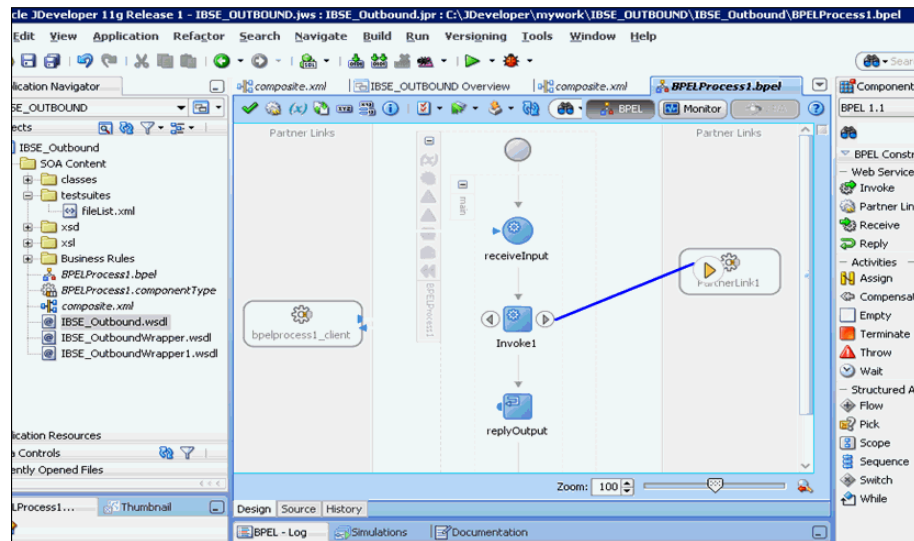
4.6.3.2 作成されたパートナ・リンクを使用した BPEL アクティビティおよびマッピングの作成

この項では、作成されたパートナ・リンクを使用して BPEL アクティビティおよびマッピングを作成する方法について説明します。

作成されたパートナ・リンクを使用して BPEL アクティビティおよびマッピングを作成するには、次のステップを実行します。

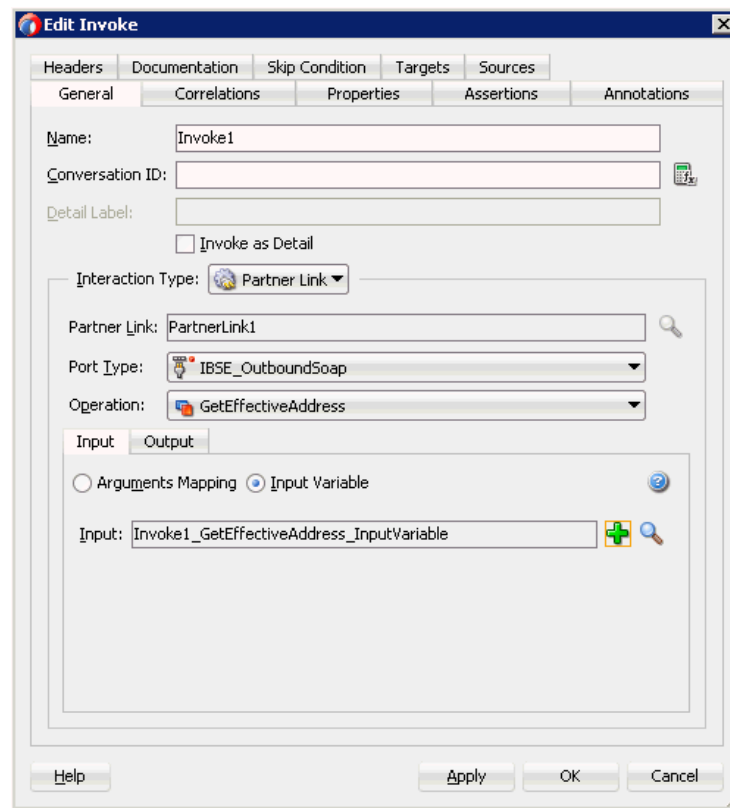
1. 「BPEL コンストラクト」から「Invoke」アクティビティ・コンポーネントを「コンポーネント」ペインにドラッグ・アンド・ドロップします。「receiveInput」アクティビティ・コンポーネントと「replyOutput」アクティビティ・コンポーネントの間に置きます。
2. 図 4-83 に示すように、新規の「Invoke」アクティビティ・コンポーネント (Invoke1) と「Partner Link」コンポーネント (Partner link1) の間に接続を作成します。

図 4-83 パートナ・リンク・コンポーネント



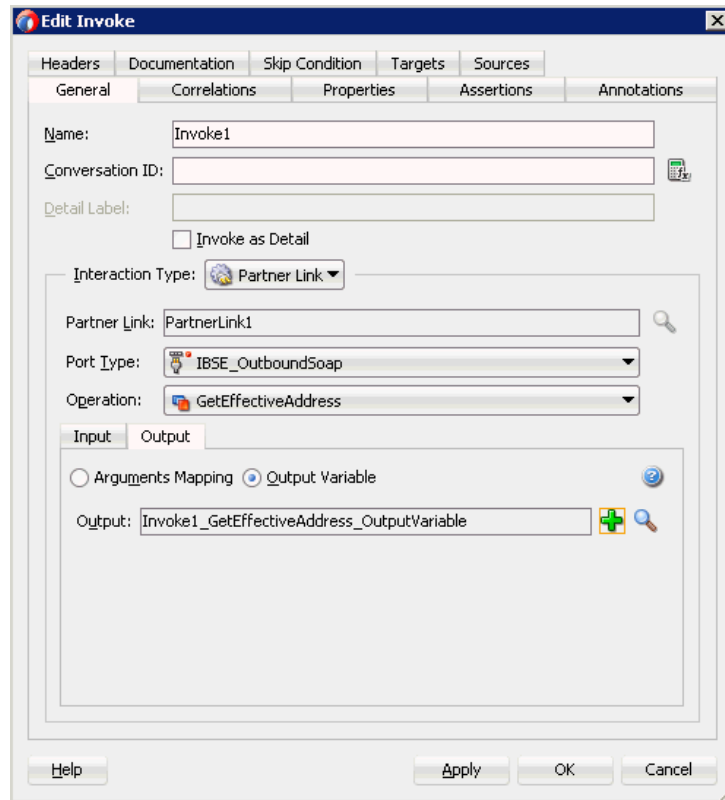
3. 表示された「起動の編集」ウィンドウで、「入力」フィールドの右のプラス (+) アイコンをクリックして、新規入力変数を構成します。
4. 新規の入力変数に示されたデフォルト値を受け入れ、「OK」をクリックします。
5. 図 4-84 に示すように、「出力」フィールドの右のプラス (+) アイコンをクリックして、新規出力変数を構成します。

図 4-84 「起動の編集」ウィンドウ



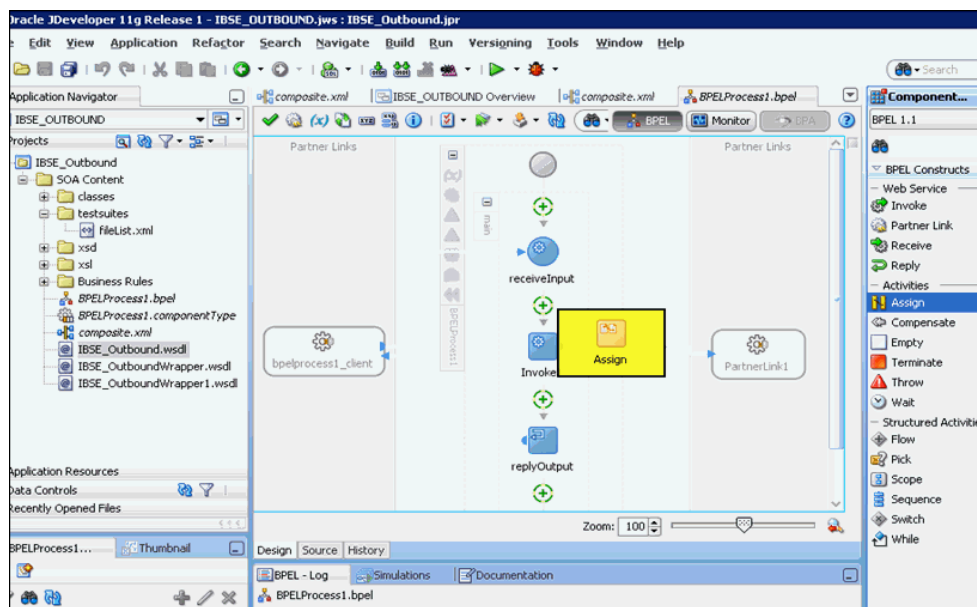
6. 新規の出力変数に示されたデフォルト値を受け入れ、「OK」をクリックします。
7. 図 4-85 に示すように、「適用」、「OK」の順にクリックします。

図 4-85 「起動の編集」ウィンドウ



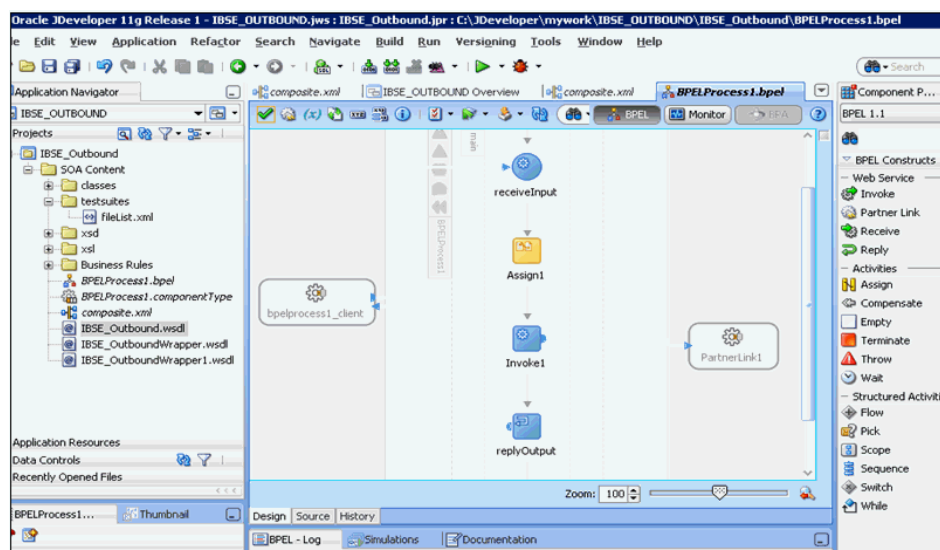
8. 「BPEL コンストラクト」から「Assign」アクティビティ・コンポーネントを「コンポーネント」ペインにドラッグ・アンド・ドロップします。図 4-86 に示すように、「Receive」アクティビティ・コンポーネント (receiveInput) と「Invoke」アクティビティ・コンポーネント (Invoke1) の間に置きます。

図 4-86 「Assign」 アクティビティ・コンポーネント



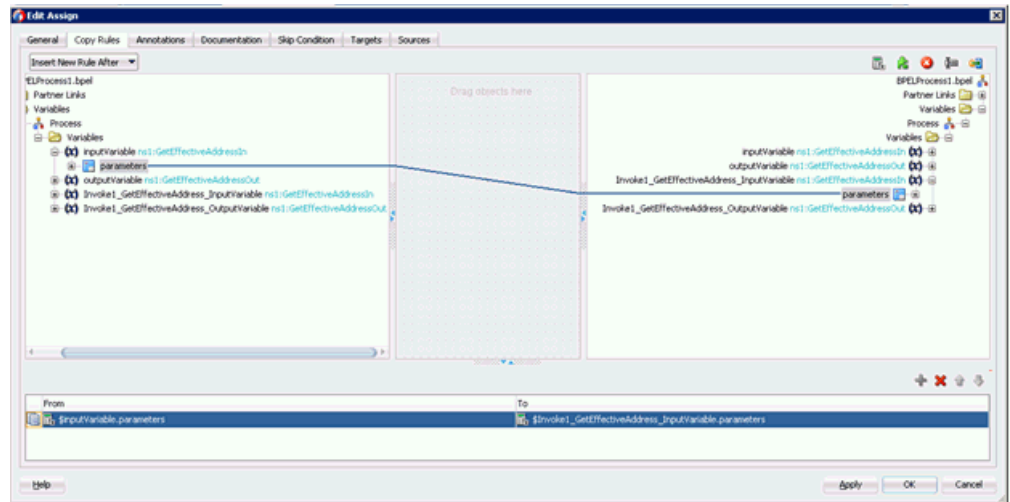
9. 図 4-87 に示すように、新規の「Assign」アクティビティ・コンポーネント (Assign1) をダブルクリックします。

図 4-87 「Assign」 アクティビティ・コンポーネント



10. 左ペインの「変数」の下で、「InputVariable」を展開し、「parameters」を選択します。
11. 右ペインの「変数」の下で、「Invoke1_GetEffectiveAddress_InputVariable」を展開し、「parameters」を選択します。
12. 図 4-88 に示すように、「InputVariable」パラメータを「Invoke1_GetEffectiveAddress_InputVariable」パラメータにドラッグしてマップします。

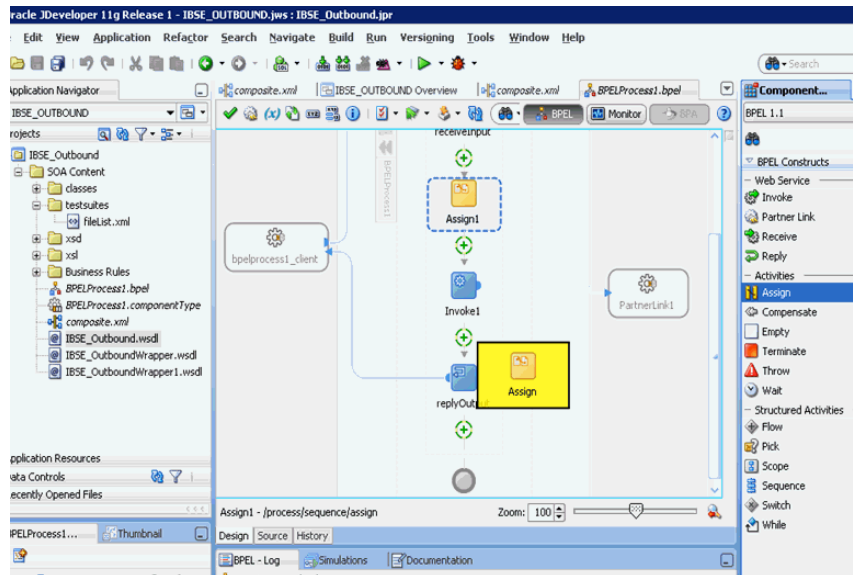
図 4-88 「InputVariable」パラメータ



13. 「適用」、「OK」の順にクリックします。

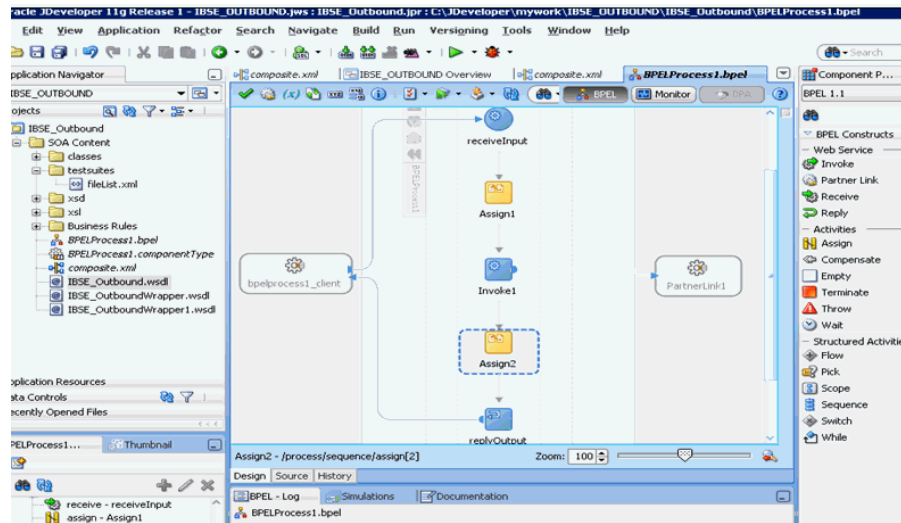
14. 図 4-89 に示すように、「Assign」アクティビティ・コンポーネントを「コンポーネント」ペインにドラッグ・アンド・ドロップし、「Invoke」アクティビティ (Invoke1) と「Reply」アクティビティ (replyOutput) の間に置きます。

図 4-89 「Assign」アクティビティ・コンポーネント



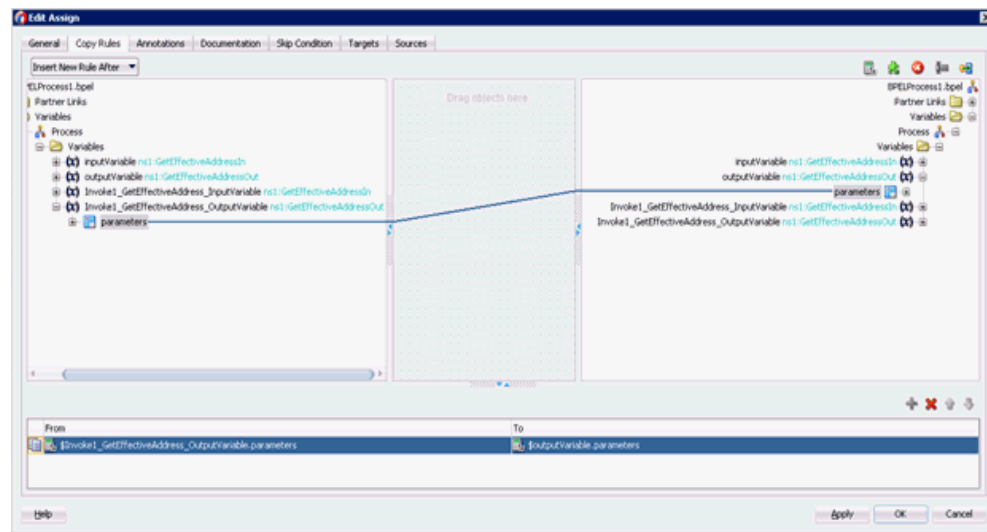
15. 図 4-90 に示すように、新規の「Assign」アクティビティ・コンポーネント (Assign2) をダブルクリックします。

図 4-90 新規の「Assign」アクティビティ・コンポーネント



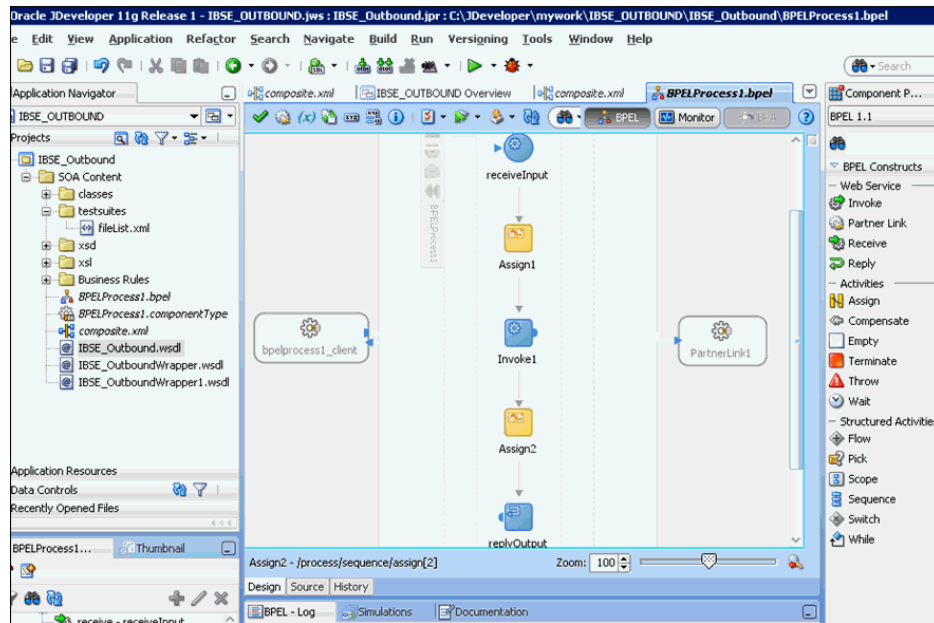
16. 左ペインの「変数」の下で、「Invoke1_GetEffectiveAddress_OutputVariable」を展開し、「parameters」を選択します。
17. 右ペインの「変数」の下で、「outputVariable」を展開し、「parameters」を選択します。
18. 図 4-91 に示すように、「Invoke1_GetEffectiveAddress_OutputVariable」パラメータを「outputVariable」パラメータにドラッグしてマップします。

図 4-91 「outputVariable」パラメータ



19. 「適用」、「OK」の順にクリックします。
- 図 4-92 に示すように、コンポーネント・ペインに戻ります。

図 4-92 コンポーネント・ペイン



- メニュー・バーで「すべて保存」アイコンをクリックし、構成された新規のアウトバウンド BPEL プロセス・コンポーネントを保存します。

これで、BPEL アウトバウンド・プロセスをデプロイする準備ができました。4-28 ページの 4.4.4 項「BPEL アウトバウンド・プロセスのデプロイ」の記載と同じ手順に従います。

デプロイ後、4-32 ページの 4.4.5 項「Oracle Enterprise Manager Console での入力 XML ドキュメントの起動」に従って入力 XML を起動できます。

Oracle SOA Suite でのメディエータ・サービス・コンポーネントとの統合

この章では、Oracle SOA Suite でのメディエータ・サービス・コンポーネントとの統合について説明します。次のトピックが含まれています。

- 5.1 項「新規のアプリケーション・サーバー接続の構成」
- 5.2 項「メディエータ・アウトバウンド・プロセスの構成 (J2CA 構成)」
- 5.3 項「メディエータ・インバウンド・プロセスの構成 (J2CA 構成)」
- 5.4 項「メディエータ・アウトバウンド・プロセスの構成 (BSE 構成)」

この章で示すシナリオには、次の前提条件が必要です。

前提条件

インストールおよび構成に関する要件は次のとおりです。

- Oracle Application Adapter for J.D. Edwards OneWorld が Oracle WebLogic Server にインストールされている必要があります。
- J.D. Edwards OneWorld がインバウンドおよびアウトバウンド処理用に構成されている必要があります。

関連項目： *Oracle Application Server Adapter コンセプト・ガイド*

この章の例では、J.D. Edwards OneWorld とのサービスおよびイベント統合を示すのに必要な構成手順について説明しています。これを使用する前に、次のことについて理解する必要があります。

- サービスおよびイベント用に Oracle Application Adapter for J.D. Edwards OneWorld を構成する方法。詳細は、第 2 章「Oracle Application Adapter for J.D. Edwards OneWorld の構成」を参照してください。
- Oracle JDeveloper を構成する方法。詳細は、第 4 章「Oracle SOA Suite での BPEL サービス・コンポーネントの統合」を参照してください。

メディエータ統合の概要

メディエータは、包括的なアプリケーション統合フレームワークを提供します。Oracle Application Adapter for J.D. Edwards OneWorld をメディエータと使用することにより、カスタム・コードの記述を必要とせずに、シームレスにエンタープライズ・ソフトウェアを統合できます。カスタム・コード・ソリューションではなく機能的なモデリングなので、ソフトウェア再利用が可能になり、ソフトウェアのライフサイクルで発生する複雑さや管理上の問題を軽減します。この統合モデルは、高レベルの統合ロジックと低レベルのプラットフォーム・サービスの 2 つのコンポーネントで構成されています。

Oracle WebLogic Server とのアダプタの統合、メディエータは次の 2 つのステップからなるプロセスです。

1. **デザインタイム**: サービスおよびイベント用に Oracle Application Adapter for J.D. Edwards OneWorld をアプリケーション・エクスペローラで構成します。詳細は、第 2 章「[Oracle Application Adapter for J.D. Edwards OneWorld の構成](#)」を参照してください。統合ロジックは iStudio でモデル化されます。メタデータはリポジトリ内に格納されます。
2. **実行時**: 基になるプラットフォームでは、このメタデータをランタイム・インスタクションとして処理し、特定のアプリケーション間の通信を可能にします。

5.1 新規のアプリケーション・サーバー接続の構成

Oracle JDeveloper での新規アプリケーション・サーバー接続の構成方法の詳細は、4-2 ページの 4.3 項「[新規のアプリケーション・サーバー接続の構成](#)」を参照してください。

5.2 メディエータ・アウトバウンド・プロセスの構成 (J2CA 構成)

この項では、Oracle JDeveloper でメディエータ・プロジェクトを使用して J.D. Edwards OneWorld システム用にメディエータ・アウトバウンド・プロセスを構成する方法について説明します。

このアウトバウンド・ユース・ケース・シナリオ用のサンプル・プロジェクトが、次に示す Application Adapter のインストール・フォルダに用意されています。

```
<ADAPTER_HOME>\etc\sample\JDEdwards_Samples.zip\JDEdwards_Samples\Mediator\J2CA\Outbound_Project
```

この項では、次のトピックについて説明します。

- [5.2.1 項「SOA 用の空のコンポジットの作成](#)
- [5.2.2 項「メディエータ・アウトバウンド・プロセスの定義](#)
- [5.2.3 項「メディエータ・アウトバウンド・プロセスのデプロイ](#)
- [5.2.4 項「Oracle Enterprise Manager Console での入力 XML ドキュメントの起動](#)

前提条件

メディエータ・アウトバウンド・プロセスを設計する前に、アプリケーション・エクスペローラを使用して各 WSDL ファイルを生成する必要があります。詳細は、4-8 ページの 4.4.1 項「[リクエスト / レスポンス型のサービス用の WSDL の生成](#)」を参照してください。

5.2.1 SOA 用の空のコンポジットの作成

SOA 用に空のコンポジットを作成するには、次のステップを実行します。

1. 新規の SOA アプリケーションを作成します。
2. 新規の SOA アプリケーションの名前を入力し、「次へ」をクリックします。
「プロジェクトの名前付け」ページが表示されます。
3. プロジェクト名を入力し、「次へ」をクリックします。
「SOA 設定の構成」ページが表示されます。

4. 「コンポジット・テンプレート」リストで「空のコンポジット」を選択し、「終了」をクリックします。

詳細は、4-9 ページの 4.4.2 項「SOA 用の空のコンポジットの作成」を参照してください。

5.2.2 メディアエータ・アウトバウンド・プロセスの定義

この項では、メディアエータ・アウトバウンド・プロセスの定義方法について説明します。次のトピックで構成されています。

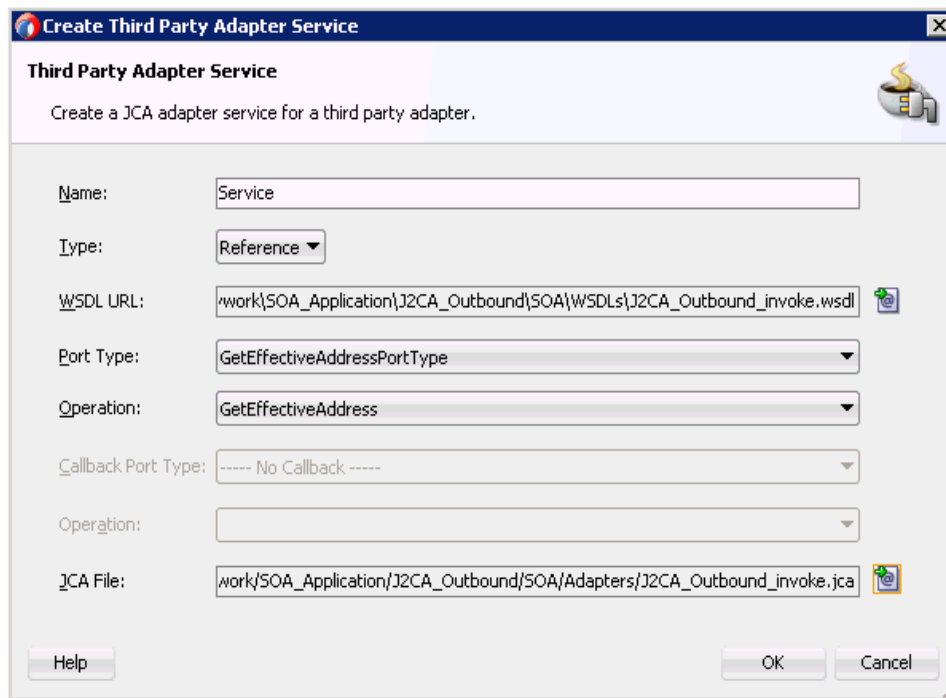
- 5.2.2.1 項「サード・パーティ・アダプタ・サービス・コンポーネントの構成」
- 5.2.2.2 項「アウトバウンド・メディアエータ・プロセス・コンポーネントの構成」
- 5.2.2.3 項「ルーティング・ルールの構成」
- 5.2.2.4 項「12c でのデプロイメントに関する既知の問題への対応」

5.2.2.1 サード・パーティ・アダプタ・サービス・コンポーネントの構成

サード・パーティ・アダプタ・サービス・コンポーネントを作成するには、次のステップを実行します。

1. 「サービス・アダプタ」ペインから「サード・パーティ・アダプタ」コンポーネントを「外部参照」ペインにドラッグ・アンド・ドロップします。
2. サード・パーティ・アダプタ・サービスの名前を入力します。
3. 「タイプ」ドロップダウン・リストで「参照」が選択されていることを確認します (デフォルト)。
4. 「WSDL URL」フィールドの右の「既存の WSDL を検索します。」アイコンをクリックします。
5. 次のディレクトリまでブラウズし、アウトバウンド WSDL ファイルを選択します。
<ADAPTER_HOME>\wsdl.s
6. 「OK」をクリックします。
7. 「OK」をクリックします。
アウトバウンド WSDL ファイルおよび関連リクエスト/レスポンス XML スキーマ・ファイル (.xsd) が、作成済のプロジェクト・フォルダにインポートされます。
8. 「JCA ファイル」フィールドの右の「JCA ファイルを検索します。」アイコンをクリックします。
9. 次のディレクトリまでブラウズし、JCA プロパティ・ファイルを選択します。
<ADAPTER_HOME>\wsdl.s
10. 「OK」をクリックします。
「ファイルのコピー」確認メッセージが表示されます。
11. 「はい」をクリックします。
JCA プロパティ・ファイルのコピーがプロジェクト・フォルダ内に作成されます。

図 5-1 「サード・パーティ・アダプタ・サービスの作成」ダイアログ



12. 「OK」をクリックします。

サード・パーティ・アダプタ・サービス・コンポーネントが作成され、「外部参照」ペインに表示されます。

これで、アウトバウンド・メディエータ・プロセス・コンポーネントを構成する準備ができました。

詳細は、4-41 ページの [4.5.3.1 項「サード・パーティ・アダプタ・サービス・コンポーネントの作成」](#) を参照してください。

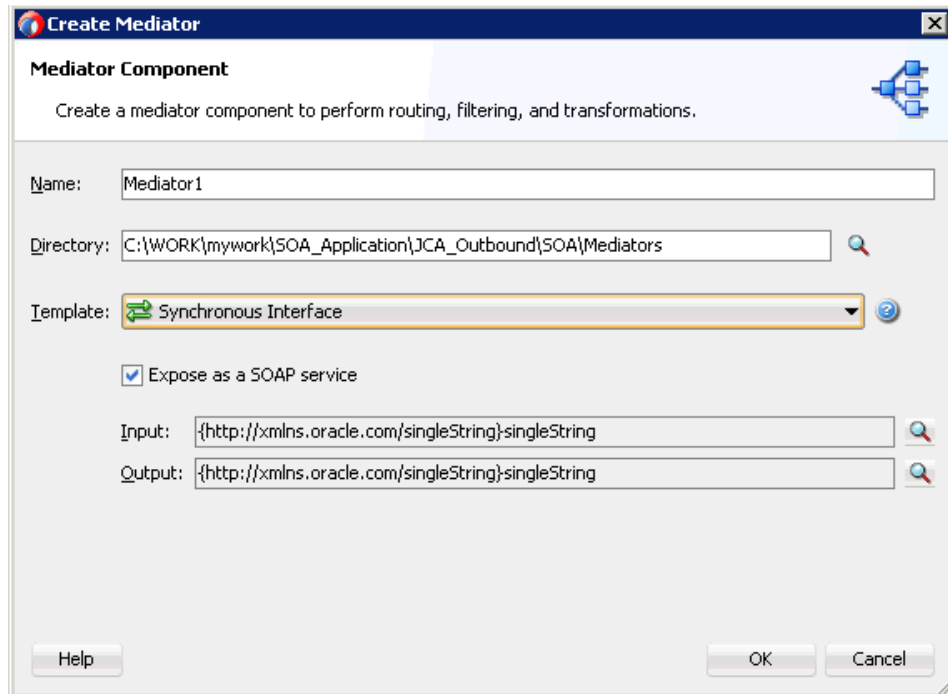
5.2.2.2 アウトバウンド・メディエータ・プロセス・コンポーネントの構成

アウトバウンド・メディエータ・プロセス・コンポーネントを構成するには、次のステップを実行します。

1. 「コンポーネント」ペインから **メディエータ・プロセス・コンポーネント** を「コンポーネント」ペインにドラッグ・アンド・ドロップします。

図 5-2 に示すように、「メディエータの作成」ダイアログが表示されます。

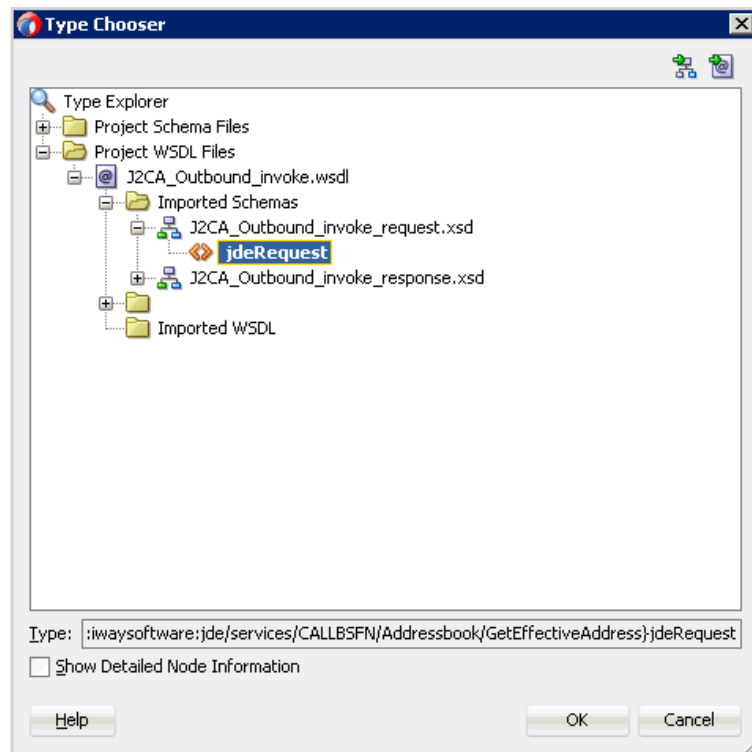
図 5-2 「メディエータの作成」ダイアログ



2. 「名前」フィールドで、新規のアウトバウンド・メディエータ・プロセス・コンポーネントの名前を入力するか、デフォルト値をそのまま使用します。
3. 「テンプレート」ドロップダウン・リストから、「同期インタフェース」を選択します。
4. 関連する XML リクエスト・スキーマ・ファイルの選択用の「入力」フィールドの右の「参照」アイコンをクリックします。

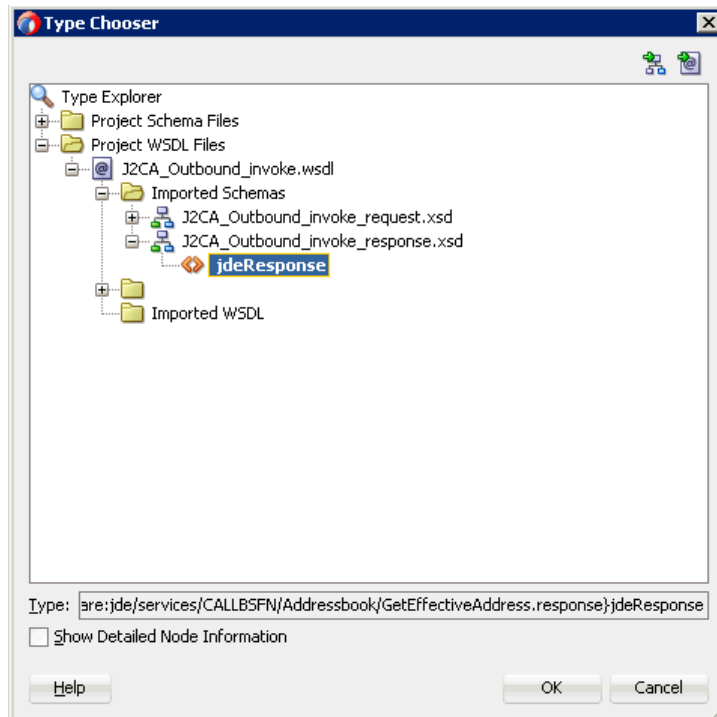
図 5-3 に示すように、「タイプ・チューザ」ダイアログが表示されます。

図 5-3 「タイプ・チューザ」 ダイアログ



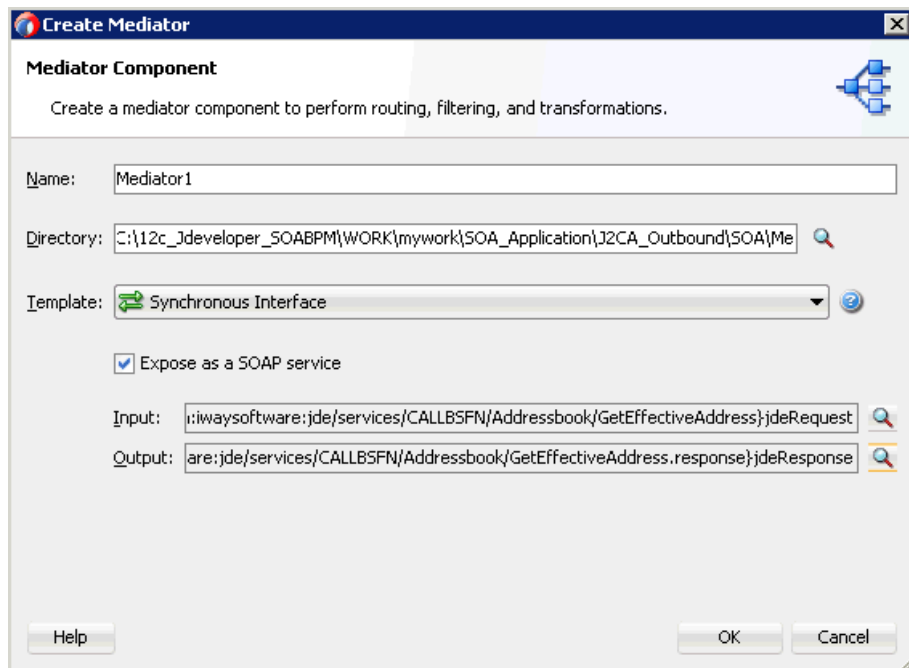
5. 「プロジェクトの WSDL ファイル」、「J2CA_Outbound_invoke.wsdl」、「インポートしたスキーマ」、「J2CA_Outbound_invoke_request.xsd」の順に展開し、「jdeRequest」を選択します。
6. 「OK」をクリックします。
メディアータの作成」ダイアログに戻ります。
7. 関連する XML レスポンス・スキーマ・ファイルの選択用の「出力」フィールドの右の「参照」アイコンをクリックします。
図 5-4 に示すように、「タイプ・チューザ」ダイアログが表示されます。

図 5-4 「タイプ・チューザ」 ダイアログ



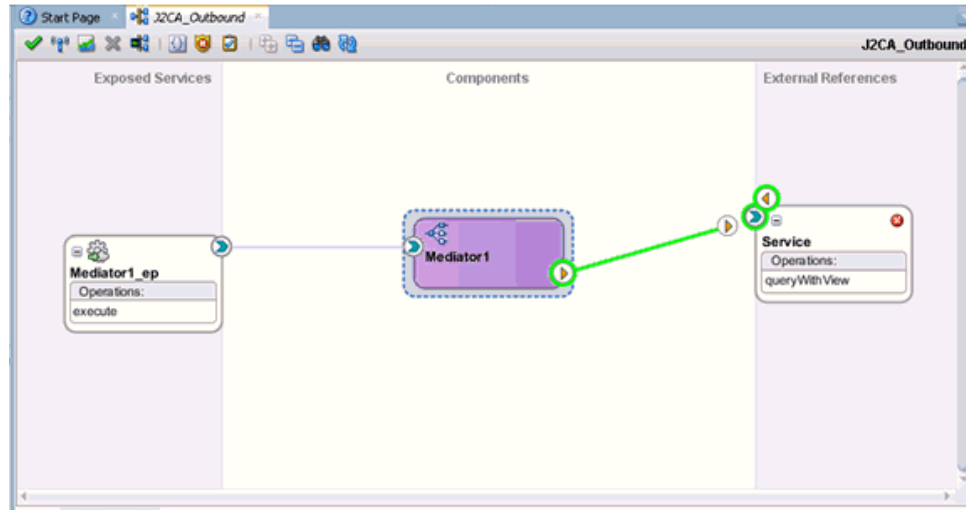
8. 「プロジェクトの WSDL ファイル」、「J2CA_Outbound_invoke.wsdl」、「インポートしたスキーマ」、「J2CA_Outbound_invoke_response.xsd」の順に展開し、「jdeResponse」を選択します。
9. 「OK」をクリックします。
☒ 5-5 に示すように、「メディエータの作成」ダイアログに戻ります。

図 5-5 「メディエータの作成」ダイアログ



10. 「OK」をクリックします。
11. 図 5-6 に示すように、アウトバウンド・メディエータ・プロセス・コンポーネントとサード・パーティ・アダプタ・サービス・コンポーネント間の接続を作成します。

図 5-6 作成された接続



これで、ルーティング・ルールを構成する準備ができました。

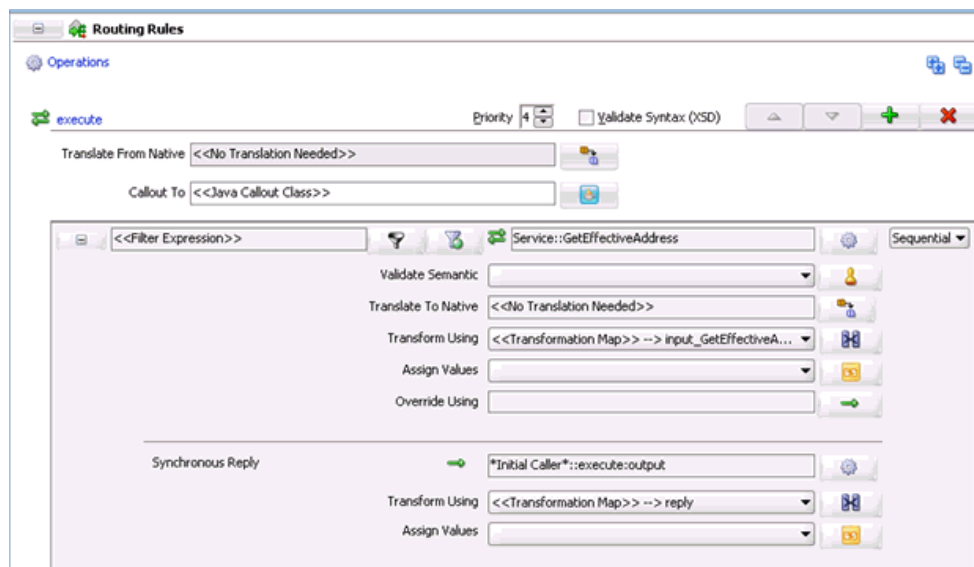
5.2.2.3 ルーティング・ルールの構成

メディエータ・アウトバウンド・プロセス・コンポーネントのルーティング・ルールを構成するには、次のステップを実行します。

1. 「コンポーネント」 ペインで、アウトバウンド・メディエータ・プロセス・コンポーネントをダブルクリックします。

図 5-7 に示すように、「ルーティング・ルール」 ダイアログが表示されます。

図 5-7 「ルーティング・ルール」 ダイアログ



2. 「<<フィルタ式>>」領域で、「次を使用して変換」フィールドの右にあるアイコンをクリックします。

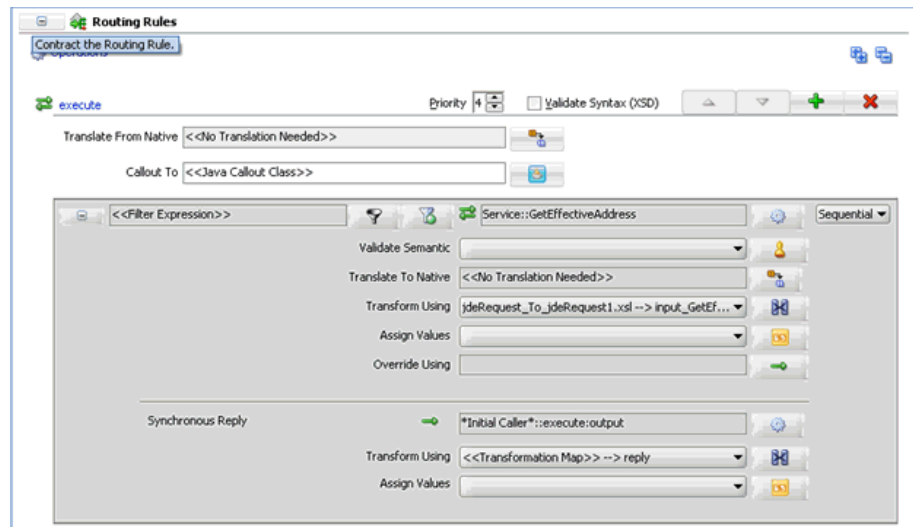
図 5-8 に示すように、「リクエスト・トランスフォーメーション・マップ」ダイアログが表示されます。

図 5-8 「リクエスト・トランスフォーメーション・マップ」ダイアログ



3. 「追加」 (「+」) アイコンをクリックします。
「トランスフォーメーション・マップの作成」ページが表示されます。
4. タイプとして「XSLT」が選択されていることを確認し、「OK」をクリックします。
5. 「OK」をクリックします。
6. 「ns0:jdeRequest」ソース要素を「ns0:jdeRequest」ターゲット要素にマップします。
「自動マップ・プリファレンス」ダイアログが表示されます。
7. デフォルト値を維持し、「OK」をクリックします。
8. 図 5-9 に示すように、「ルーティング・ルール」ダイアログに戻ります。

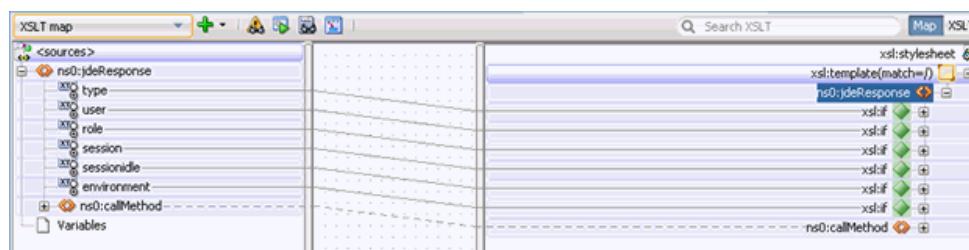
図 5-9 「ルーティング・ルール」ダイアログ



9. 「同期リプライ」領域で、「次を使用して変換」フィールドの右にあるアイコンをクリックします。
「リプライ・トランスフォーメーション・マップ」ダイアログが表示されます。
10. 「追加」 (「+」) アイコンをクリックします。
「トランスフォーメーションの作成」ページが表示されます。

11. タイプとして「XSLT」が選択されていることを確認し、「OK」をクリックします。
マッピング・ページが表示されます。
12. 「OK」をクリックします。
13. 「ns0:jdeResponse」ソース要素を「ns0:jdeResponse」ターゲット要素にマップします。
「自動マップ・プリファレンス」ダイアログが表示されます。
14. デフォルト値を維持し、「OK」をクリックします。
図 5-10 に示すように、マッピングが完成します。

図 5-10 完成したマッピング



15. メニュー・バーの「すべて保存」アイコンをクリックし、構成された新規のアウトバウンド・メディアータ・プロセス・コンポーネントを保存します。

5.2.2.4 12c でのデプロイメントに関する既知の問題への対応

12c でのデプロイメントに関する既知の問題への対応方法の詳細は、4-26 ページの 4.4.3.3 項「12c でのデプロイメントに関する既知の問題への対応」を参照してください。

5.2.3 メディアータ・アウトバウンド・プロセスのデプロイ

メディアータ・アウトバウンド・プロセスをデプロイするには、次のステップを実行します

1. 左ペインでプロジェクト名を右クリックし、「デプロイ」を選択して、「J2CA_Outbound」をクリックします。
「デプロイメント・アクション」ページが表示されます。
2. 「アプリケーション・サーバーにデプロイ」が選択されていることを確認します。
3. 「次へ」をクリックします。
「構成のデプロイ」ページが表示されます。
4. デフォルト値を選択したままにし、「次へ」をクリックします。
「サーバーの選択」ページが表示されます。
5. 構成済の使用可能なアプリケーション・サーバーを選択し、「次へ」をクリックします。
「SOA サーバー」ページが表示されます。
6. ターゲットの SOA サーバーを選択し、「次へ」をクリックします。
「サマリー」ページが表示されます。

7. プロジェクトに使用可能なすべてのデプロイメント情報をレビューおよび確認し、「終了」をクリックします。

詳細は、4-28 ページの [4.4.4 項「BPEL アウトバウンド・プロセスのデプロイ」](#) を参照してください。

5.2.4 Oracle Enterprise Manager Console での入力 XML ドキュメントの起動

詳細は、4-32 ページの [4.4.5 項「Oracle Enterprise Manager Console での入力 XML ドキュメントの起動」](#) を参照してください。

5.3 メディエータ・インバウンド・プロセスの構成 (J2CA 構成)

この項では、Oracle JDeveloper でメディエータ・プロジェクトを使用して J.D. Edwards OneWorld システム用にメディエータ・インバウンド・プロセスを構成する方法について説明します。

このインバウンド・ユース・ケース・シナリオ用のサンプル・プロジェクトが、次に示す Application Adapter のインストール・フォルダに用意されています。

```
<ADAPTER_HOME>\etc\sample\JDEdwards_Samples.zip\JDEdwards_Samples\Mediator\J2CA\Inbound_Project
```

この項では、次のトピックについて説明します。

- [5.3.1 項「SOA 用の空のコンポジットの作成」](#)
- [5.3.2 項「メディエータ・インバウンド・プロセスの定義」](#)

前提条件

メディエータ・インバウンド・プロセスを設計する前に、アプリケーション・エクステンション・プロシージャを使用して各 WSDL ファイルを生成する必要があります。詳細は、4-35 ページの [4.5.1 項「イベント統合のための WSDL の生成」](#) を参照してください。

5.3.1 SOA 用の空のコンポジットの作成

SOA 用に空のコンポジットを作成するには、次のステップを実行します。

1. 新規の SOA アプリケーションを作成します。
2. 新規の SOA アプリケーションの名前を入力し、「次へ」をクリックします。
「プロジェクトの名前付け」ページが表示されます。
3. プロジェクト名を入力し、「次へ」をクリックします。
「SOA 設定の構成」ページが表示されます。
4. 「コンポジット・テンプレート」リストで「空のコンポジット」を選択し、「終了」をクリックします。

詳細は、4-9 ページの [4.4.2 項「SOA 用の空のコンポジットの作成」](#) を参照してください。

5.3.2 メディエータ・インバウンド・プロセスの定義

この項では、メディエータ・インバウンド・プロセスの定義方法について説明します。次のトピックが含まれています。

- [5.3.2.1 項「サード・パーティ・アダプタ・サービス・コンポーネントの構成」](#)

- 5.2.2.2 項「アウトバウンド・メディエータ・プロセス・コンポーネントの構成」
- 5.2.2.3 項「ルーティング・ルール」の構成」
- 5.3.2.4 項「12c でのデプロイメントに関する既知の問題への対応」

5.3.2.1 サード・パーティ・アダプタ・サービス・コンポーネントの構成

サード・パーティ・アダプタ・サービス・コンポーネントを作成するには、次のステップを実行します。

1. 「サービス・アダプタ」 ペインから「サード・パーティ・アダプタ」 コンポーネントを「公開されたサービス」 ペインにドラッグ・アンド・ドロップします。
「サード・パーティ・アダプタ・サービスの作成」 ダイアログが表示されます。
2. サード・パーティ・アダプタ・サービスの名前を入力します。
3. 「タイプ」 ドロップダウン・リストで「サービス」 が選択されていることを確認します (デフォルト)。
4. 「WSDL URL」 フィールドの右の「既存の WSDL を検索します。」 アイコンをクリックします。
「WSDL チューザ」 ダイアログが表示されます。
5. 次のディレクトリまでブラウズし、インバウンド WSDL ファイルを選択します。
<ADAPTER_HOME>\wsdl
6. 「OK」 をクリックします。
「ファイルのローカライズ」 ダイアログが表示されます。
7. 「OK」 をクリックします。
インバウンド WSDL ファイルおよび関連リクエスト/レスポンス・スキーマ・ファイル (.xsd) が、作成済のプロジェクト・フォルダにインポートされます。
「サード・パーティ・アダプタ・サービスの作成」 ダイアログに戻ります。
8. 「JCA ファイル」 フィールドの右の「JCA ファイルを検索します。」 アイコンをクリックします。
変換チューザ・ダイアログが表示されます。
9. 次のディレクトリまでブラウズし、JCA プロパティ・ファイルを選択します。
<ADAPTER_HOME>\wsdl
10. 「OK」 をクリックします。
「ファイルのコピー」 確認メッセージが表示されます。
11. 「はい」 をクリックします。
JCA プロパティ・ファイルのコピーがプロジェクト・フォルダ内に作成されます。
「サード・パーティ・アダプタ・サービスの作成」 ダイアログに戻ります。
12. 「OK」 をクリックします。
サード・パーティ・アダプタ・サービス・コンポーネントが作成され、「公開されたサービス」 ペインに表示されます。
これで、インバウンド・メディエータ・プロセス・コンポーネントを構成する準備ができました。

詳細は、5-3 ページの 5.2.2.1 項「サード・パーティ・アダプタ・サービス・コンポーネントの構成」を参照してください。

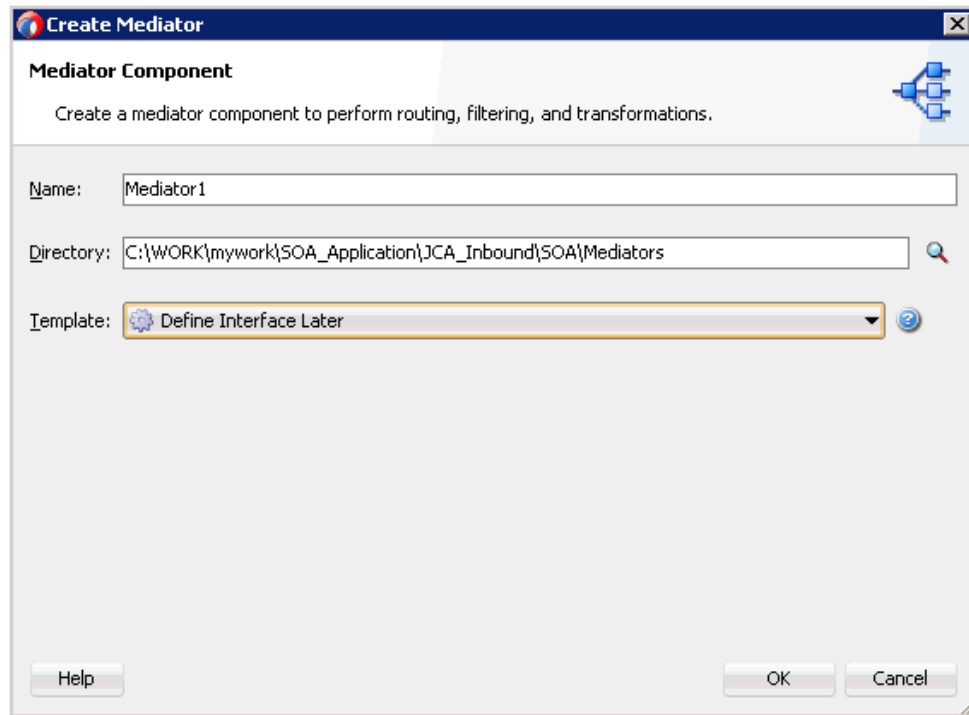
5.3.2.2 ファイル・アダプタを使用したインバウンド・メディエータ・プロセス・コンポーネントの構成

ファイル・アダプタを使用してインバウンド・メディエータ・プロセス・コンポーネントを構成するには、次のステップを実行します

1. 「サービス・コンポーネント」ペインから**メディエータ・プロセス・コンポーネント**を「コンポーネント」ペインにドラッグ・アンド・ドロップします。

図 5-11 に示すように、「メディエータの作成」ダイアログが表示されます。

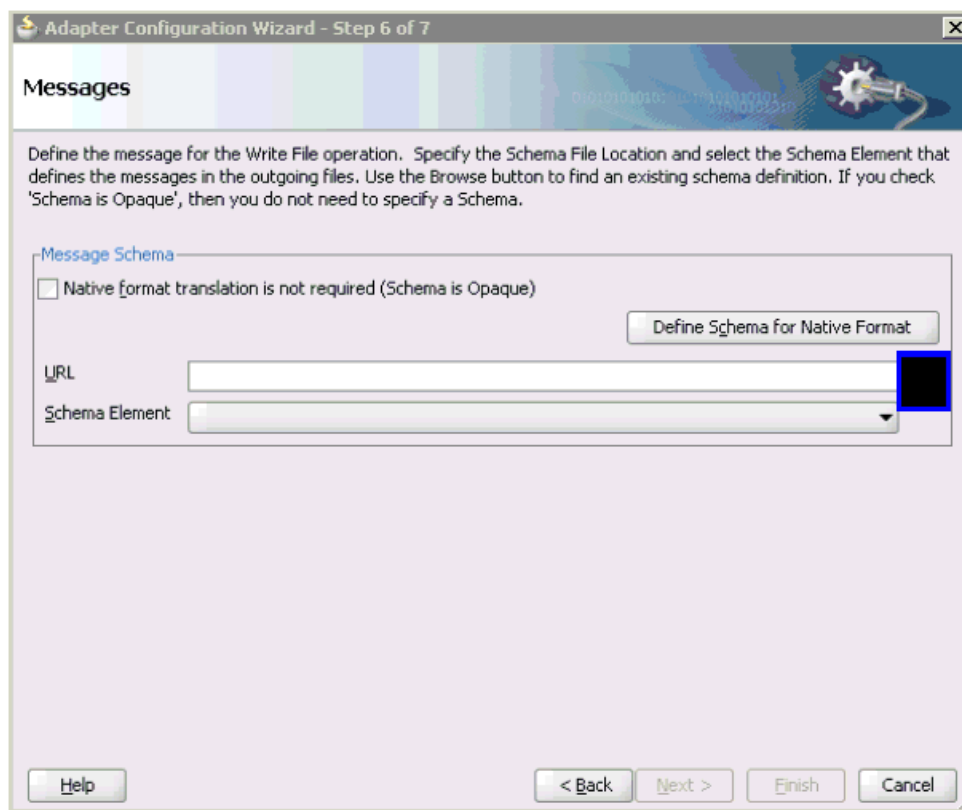
図 5-11 「メディエータの作成」ダイアログ



2. 「名前」フィールドで、新規のインバウンド・メディエータ・プロセス・コンポーネントの名前を入力します。
3. 「テンプレート」ドロップダウン・リストから、「**インタフェースを後で定義**」を選択します。
4. 「OK」をクリックします。
新規メディエータ・プロセス・コンポーネントが「コンポーネント」ペインに追加されます。
5. 「テクノロジ・アダプタ」ペインから「**ファイル**」コンポーネントを「外部参照」ペインにドラッグ・アンド・ドロップします。
ファイル・アダプタ構成ウィザードが表示されます。
6. 新規のファイル・アダプタの名前を入力し、「**次へ**」をクリックします。
「アダプタ・インタフェース」ページが表示されます。

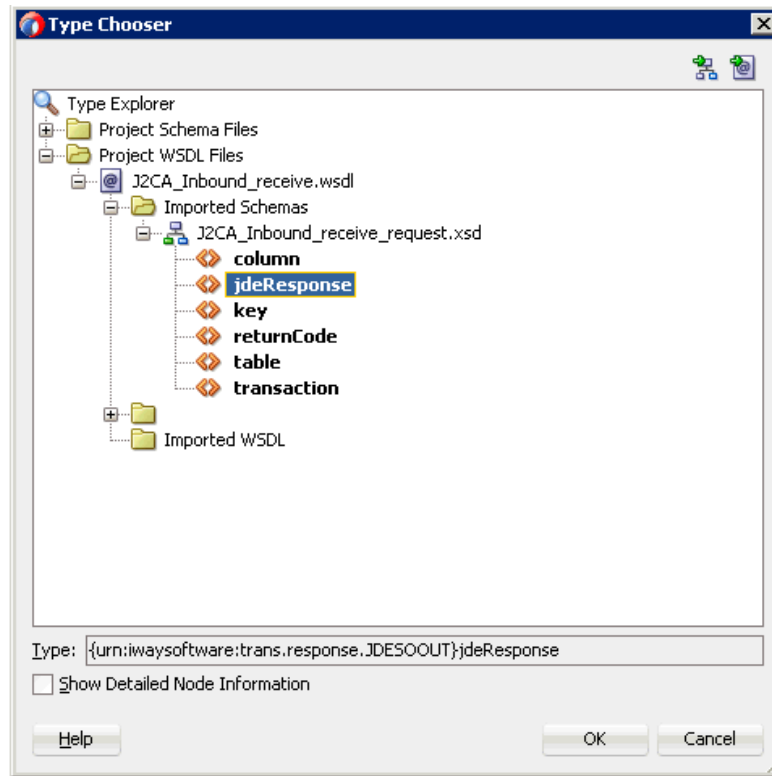
7. 「操作およびスキーマから定義 (後で指定)」オプションが選択されていることを確認します。
8. 「次へ」をクリックします。
「操作」ページが表示されます。
9. 「次へ」をクリックします。
10. 「操作タイプ」オプションのリストから「Write File」を選択し、「操作名」(「Write」など)を指定します。
11. 「次へ」をクリックします。
「ファイル構成」ページが表示されます。
12. 出力ファイルの書き込み先となるファイル・システム上の場所を指定します。
13. 「ファイル命名規則」フィールドで、出力ファイルの名前を指定します。
14. 「次へ」をクリックします。
図 5-12 に示すように、「メッセージ」ページが表示されます。

図 5-12 「メッセージページ」



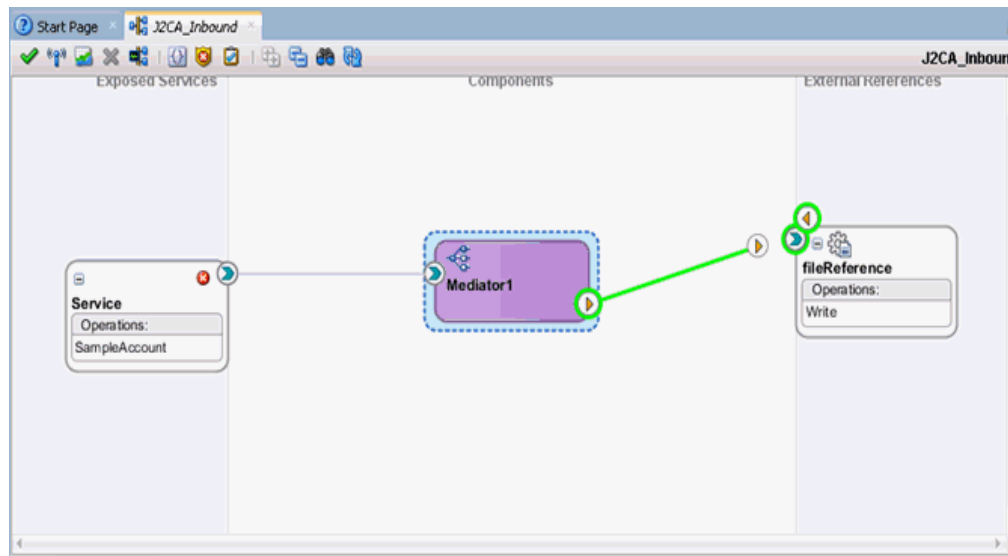
15. 「URL」フィールドの右の「参照」をクリックします。
図 5-13 に示すように、「タイプ・チューザ」ダイアログが表示されます。

図 5-13 「タイプ・チューザ」ダイアログ



16. 「プロジェクトの WSDL ファイル」、「J2CA_Inbound_receive.wsdl」、「インポートしたスキーマ」、「J2CA_Inbound_receive_request.xsd」の順に展開し、「jdeResponse」を選択します。
17. 「OK」をクリックします。
「メッセージ」ページに戻ります。
18. 「次へ」をクリックします。
「終了」ページが表示されます。
19. 「終了」をクリックします。
20. インバウンド・メディエータ・プロセス・コンポーネントとサード・パーティ・アダプタ・サービス・コンポーネント間の接続を作成します。
21. 図 5-14 に示すように、インバウンド・メディエータ・プロセス・コンポーネントとファイル・アダプタ・コンポーネント間の接続を作成します。

図 5-14 作成された接続



これで、ルーティング・ルールを構成する準備ができました。

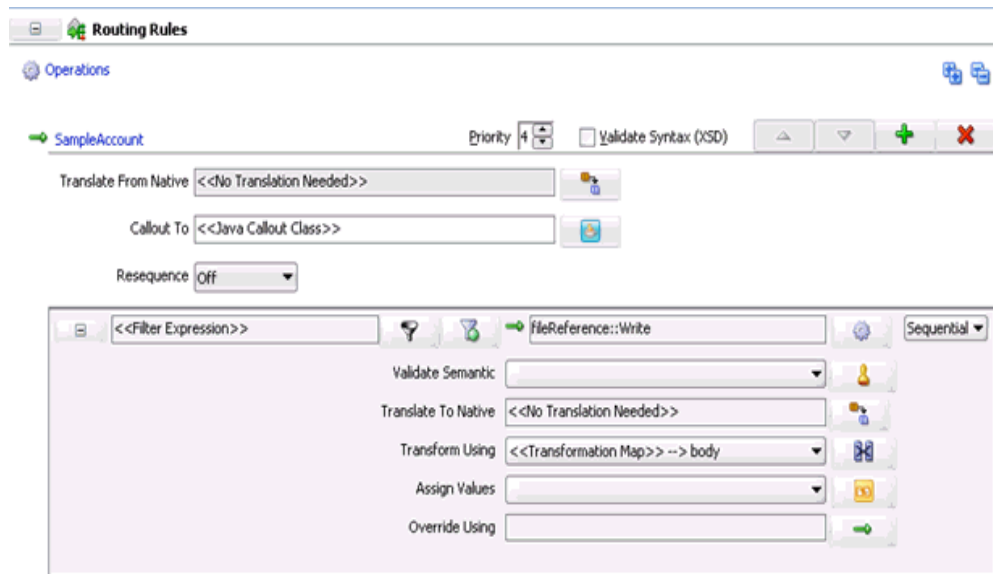
5.3.2.3 ルーティング・ルールの構成

メディエータ・インバウンド・プロセス・コンポーネントのルーティング・ルールを構成するには、次のステップを実行します。

1. 「コンポーネント」 ページで、インバウンド・メディエータ・プロセス・コンポーネントをダブルクリックします。

図 5-15 に示すように、「ルーティング・ルール」 ダイアログが表示されます。

図 5-15 「ルーティング・ルール」 ダイアログ



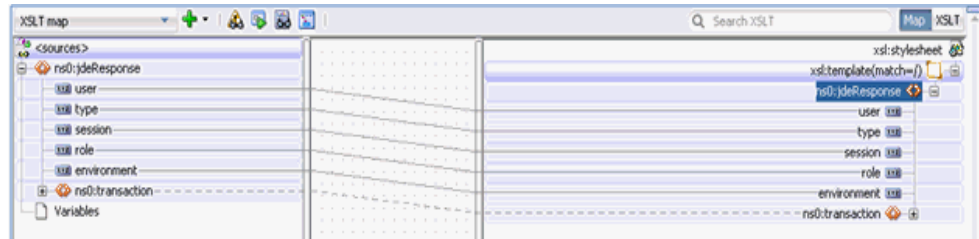
2. 「<<フィルタ式>>」 領域で、「次を使用して変換」 フィールドの右にあるアイコンをクリックします。

「リクエスト・トランスフォーメーション・マップ」 ダイアログが表示されます。

3. 「追加」 (「+」) アイコンをクリックします。タイプとして「XSLT」が選択されていることを確認し、「OK」をクリックします。
4. 「OK」をクリックします。

図 5-16 に示すように、マッピング・ページが表示されます。

図 5-16 マッピング・ページ



5. 「OK」をクリックします。
6. 「ns0:jdeResponse」ソース要素を「ns0:jdeResponse」ターゲット要素にマップします。
「自動マップ・プリファレンス」ダイアログが表示されます。
7. デフォルト値を維持し、「OK」をクリックします。
マッピングが完成しました。
8. メニュー・バーの「すべて保存」アイコンをクリックし、構成された新規のインバウンド・メディエータ・プロセス・コンポーネントを保存します。

5.3.2.4 12c でのデプロイメントに関する既知の問題への対応

12c でのデプロイメントに関する既知の問題への対応方法の詳細は、4-26 ページの 4.4.3.3 項「12c でのデプロイメントに関する既知の問題への対応」を参照してください。

これで、メディエータ・インバウンド・プロセスをデプロイする準備ができました。4-47 ページの 4.5.4 項「BPEL インバウンド・プロセスのデプロイ」の記載と同じ手順に従います。

J.D. Edwards OneWorld を介してイベント・メッセージがトリガーされたら、出力 XML が、ファイル・アダプタ・コンポーネントで指定済の場所で受信されます。J.D. Edwards OneWorld でのイベントのトリガーの詳細は、4-48 ページの 4.5.5 項「J.D. Edwards OneWorld でのイベントのトリガー」を参照してください。

5.4 メディエータ・アウトバウンド・プロセスの構成 (BSE 構成)

この項では、Oracle JDeveloper でメディエータ・プロジェクトを使用して J.D. Edwards OneWorld システム用にメディエータ・アウトバウンド・プロセスを構成する方法について説明します。

このアウトバウンド・ユース・ケース・シナリオ用のサンプル・プロジェクトが、次に示す Application Adapter のインストール・フォルダに用意されています。

```
<ADAPTER_HOME>\etc\sample\JDEdwards_Samples.zip\JDEdwards_Samples\Mediator\BSE\Outbound_Project
```

この項では、次のトピックについて説明します。

- 5.4.1 項「SOA 用の空のコンジットの作成」

- [5.4.2 項「メディアータ・アウトバウンド・プロセスの定義」](#)

前提条件

メディアータ・アウトバウンド・プロセスを設計する前に、アプリケーション・エクスプローラを使用して各 WSDL ファイルを生成する必要があります。詳細は、4-53 ページの [4.6.1 項「Web サービスを使用したリクエスト/レスポンス型サービス用の WSDL ファイルの生成」](#) を参照してください。

5.4.1 SOA 用の空のコンポジットの作成

SOA 用に空のコンポジットを作成するには、次のステップを実行します。

1. 新規の SOA アプリケーションを作成します。
2. SOA アプリケーションの名前を入力し、「次へ」をクリックします。
3. プロジェクト名を入力し(「IBSE_Outbound」など)、「次へ」をクリックします。
4. 「コンポジット・テンプレート」リストで「空のコンポジット」を選択し、「終了」をクリックします。

詳細は、4-54 ページの [4.6.2 項「SOA 用の空のコンポジットの作成」](#) を参照してください。

5.4.2 メディアータ・アウトバウンド・プロセスの定義

この項では、メディアータ・アウトバウンド・プロセスの定義方法について説明します。次のトピックが含まれています。

- [5.4.2.1 項「SOAP サービスの構成」](#)
- [5.4.2.2 項「メディアータ・コンポーネントの作成」](#)
- [5.4.2.3 項「ルーティング・ルールの構成」](#)

5.4.2.1 SOAP サービスの構成

SOAP サービスを構成するには、次のステップを実行します。

1. 「テクノロジ・アダプタ」ペインから「SOAP」ノードを「外部参照」ペインにドラッグ・アンド・ドロップします。
2. SOAP サービスの適切な名前を入力し、「WSDL URL」フィールドの右の「既存の WSDL を検索します。」アイコンをクリックします。
3. 表示された「SOA リソース・ブラウザ」ウィンドウで、「ファイル・システム」タブを選択し、アプリケーション・エクスプローラから WSDL をエクスポートする場所にナビゲートし、WSDL を選択して、「OK」をクリックします。
4. 「Web サービスの作成」ウィンドウで、「OK」をクリックします。
5. 表示された「ファイルのローカライズ」ウィンドウで、「OK」をクリックします。これにより、WSDL ファイルがプロジェクト・フォルダにインポートされます。

Web サービスが作成され、表示されます。

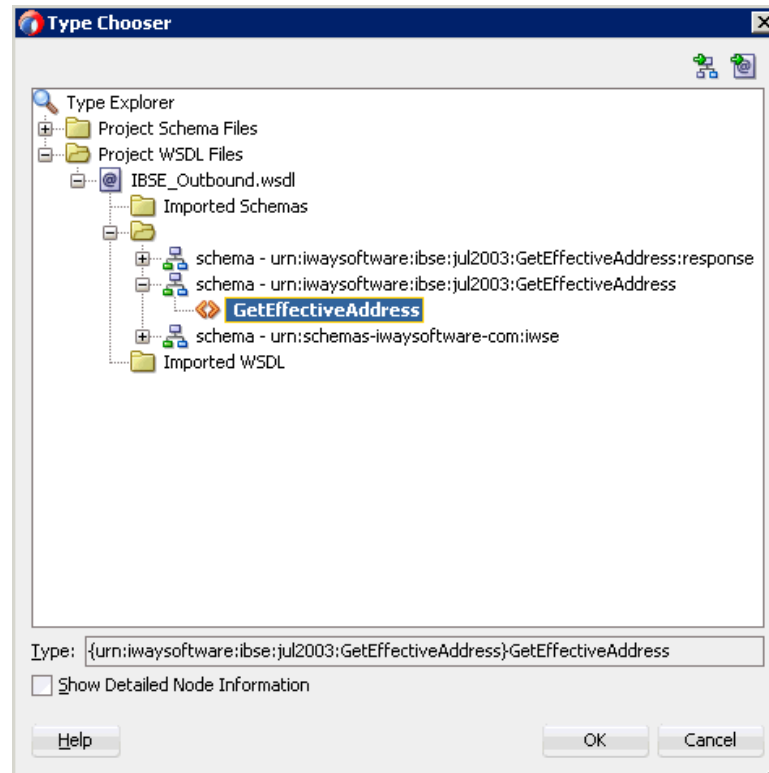
5.4.2.2 メディアータ・コンポーネントの作成

メディアータ・コンポーネントを作成するには、次のステップを実行します。

1. 「コンポーネント」ペインから「メディアータ」コンポーネントを「コンポーネント」ペインにドラッグ・アンド・ドロップします。

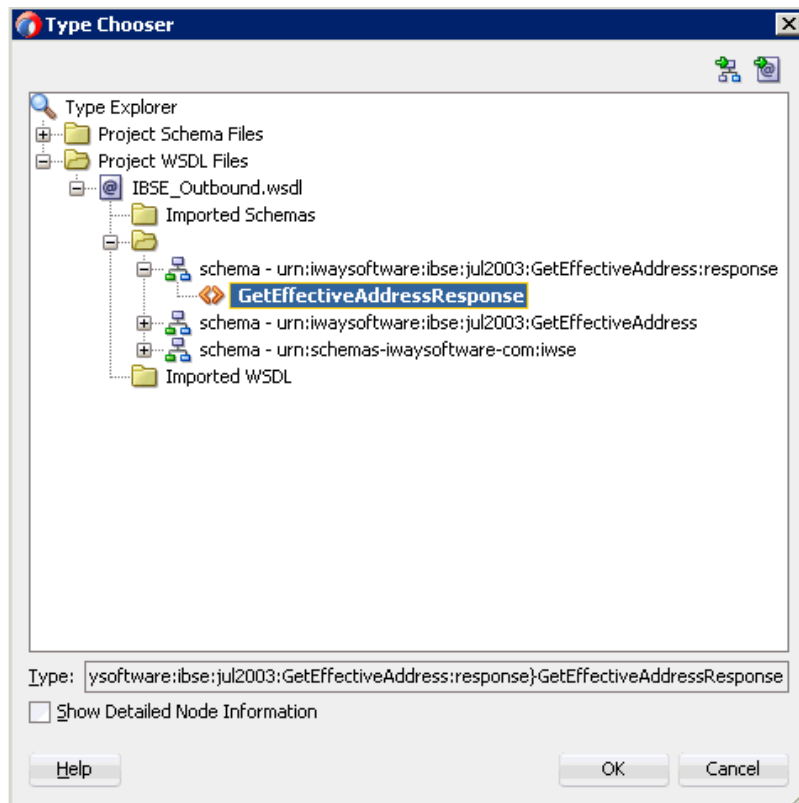
2. 「名前」フィールドで、新規のアウトバウンド・メディエータ・プロセス・コンポーネントの名前を入力します。
3. 「テンプレート」ドロップダウン・リストから、「同期インタフェース」を選択します。
4. 関連する XML リクエスト・スキーマ・ファイルの選択用の「入力」フィールドの右の「参照」アイコンをクリックします。
5. 図 5-17 に示すように、「タイプ・チューザ」ダイアログで、「プロジェクトの WSDL ファイル」を展開し、「IBSE_Outbound.wsdl」を選択して、「GetEffectiveAddress」をクリックします。

図 5-17 「タイプ・チューザ」ダイアログ



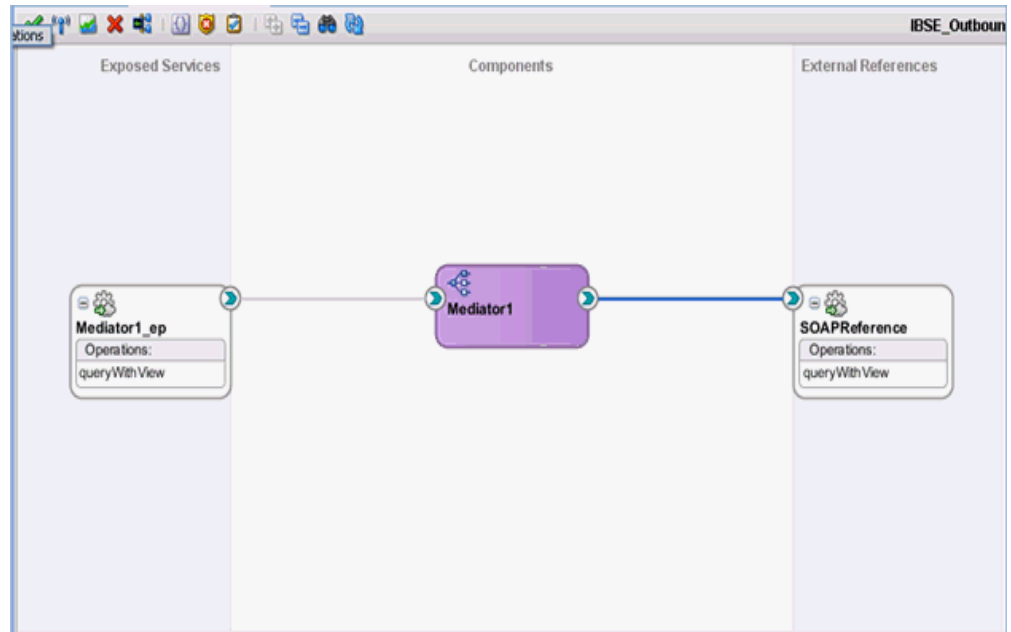
6. 「OK」をクリックします。
7. 関連する XML レスポンス・スキーマ・ファイルの選択用の「出力」フィールドの右の「参照」アイコンをクリックします。
8. 図 5-18 に示すように、「タイプ・チューザ」ダイアログで、「プロジェクトの WSDL ファイル」を展開し、「IBSE_Outbound.wsdl」を選択して、「GetEffectiveAddressResponse」をクリックします。

図 5-18 「タイプ・チューザ」ダイアログ



9. 「OK」をクリックします。
10. 「OK」をクリックします。
メディエータ・コンポーネントが作成され、表示されます。
11. 図 5-19 に示すように、「メディエータ」コンポーネントと「SOAP サービス」コンポーネント間の接続を作成します。

図 5-19 作成された接続

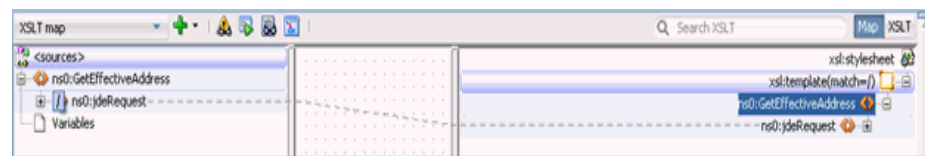


5.4.2.3 ルーティング・ルール構成

ルーティング・ルールを構成するには、次のステップを実行します。

1. 「コンポーネント」ペインで、「メディエータ」コンポーネントをダブルクリックします。
2. 「静的ルーティング」セクションの「<<フィルタ式>>」領域で、「次を使用して変換」フィールドの右にあるアイコンをクリックします。
3. 表示された「リクエスト・トランスフォーメーション・マップ」ウィンドウで、「追加」(「+」)アイコンをクリックします。「トランスフォーメーション・マップの作成」ダイアログ・ボックスでタイプとして「XSLT」が選択されていることを確認し、「OK」をクリックします。
4. 「OK」をクリックします。
5. 図 5-20 に示すように、「ns0:GetEffectiveAddress」ソース要素を「ns0:GetEffectiveAddress」ターゲット要素にマップします。

図 5-20 ソース要素とターゲット要素

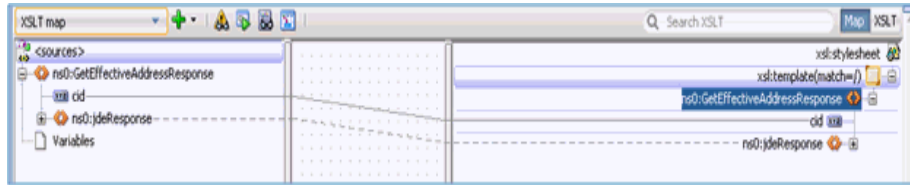


6. 表示された「自動マップ・プリファレンス」ウィンドウで、デフォルト値を維持し、「OK」をクリックします。
7. 「同期リプライ」領域で、「次を使用して変換」フィールドの右にあるアイコンをクリックします。
8. 表示された「リプライ・トランスフォーメーション・マップ」ウィンドウで、「追加」(「+」)アイコンをクリックします。「トランスフォーメーション・マップの

作成」ダイアログ・ボックスでタイプとして「XSLT」が選択されていることを確認し、「OK」をクリックします。

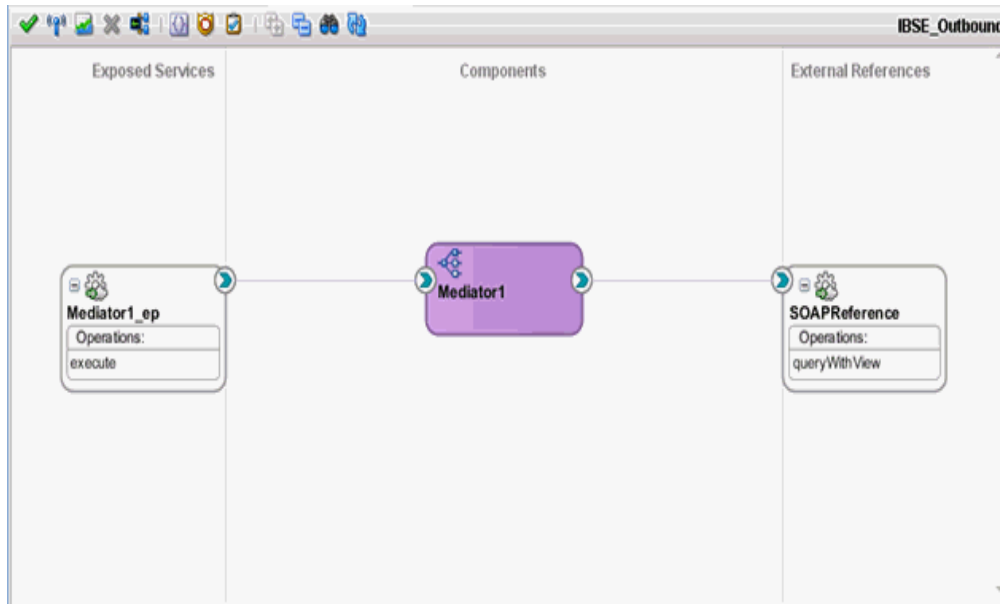
9. 図 5-21 に示すように、「ns0:GetEffectiveAddressResponse」ソース要素を「ns0:GetEffectiveAddressResponse」ターゲット要素にマップします。

図 5-21 ソース要素とターゲット要素



10. 表示された「自動マップ・プリファレンス」ウィンドウで、デフォルト値を維持し、「OK」をクリックします。
11. 左ペインの「composite.xml」をダブルクリックします。
12. 図 5-22 に示すように、メニュー・バーの「すべて保存」アイコンをクリックし、構成された新規のアウトバウンド・メディエータ・コンポーネントを保存します。

図 5-22 「すべて保存」アイコン



これで、メディエータ IBSE アウトバウンド・プロセスをデPLOYする準備ができました。5-10 ページの 5.2.3 項「メディエータ・アウトバウンド・プロセスのデPLOY」の記載と同じ手順に従います。

デPLOY後、5-11 ページの 5.2.4 項「Oracle Enterprise Manager Console での入力 XML ドキュメントの起動」に従って入力 XML を起動できます。

Oracle SOA Suite での BPM サービス・コンポーネントの統合

Oracle Application Adapter for J.D. Edwards OneWorld には、Oracle Business Process Management(BPM) がシームレスに統合されており、Web サービスの統合を容易にします。Oracle BPM は、サービス指向アーキテクチャ (SOA) に基づいています。これは、Web Service Definition Language(WSDL) ドキュメントとして公開されているアダプタ・サービスを使用します。

この章の内容は以下のとおりです。

- 6.1 項「概要」
- 6.2 項「アダプタのデプロイメント」
- 6.3 項「新規のアプリケーション・サーバー接続の構成」
- 6.4 項「トランスフォーメーションを使用したサービス統合のためのアウトバウンド BPM プロセスの設計 (J2CA 構成)」
- 6.5 項「トランスフォーメーションを使用したイベント統合のためのインバウンド BPM プロセスの設計 (J2CA 構成)」
- 6.6 項「トランスフォーメーションを使用したサービス統合のためのアウトバウンド BPM プロセスの設計 (BSE 構成)」

6.1 概要

Oracle BPM を統合するには、Oracle Application Adapter for J.D. Edwards OneWorld が Oracle BPM と同じ WLS コンテナにデプロイされている必要があります。また、基になるアダプタ・サービスは、アダプタのリクエスト/レスポンス (アウトバウンド) とイベント通知 (インバウンド) の両方のサービス用に Oracle Adapter アプリケーション・エクスペローラ (アプリケーション・エクスペローラ) でデザインタイム中に生成される WSDL ファイルとして公開される必要があります。詳細は、2-11 ページの「[WSDL の生成 \(J2CA 構成のみ\)](#)」を参照してください。

生成された WSDL ファイルは、インバウンドまたはアウトバウンドのアダプタ・サービスにそれぞれ適切な BPM プロセスの設計に使用されます。完成した BPM プロセスは、JDeveloper で正常にコンパイルされ、BPM サーバーにデプロイされる必要があります。BPM サーバーへのデプロイメント後すぐに、新規構築されたプロセスはすべて自動的に Oracle Enterprise Manager Console にデプロイされ、BPM プロセスの実行、監視、管理、およびアダプタ・イベントのリスニングを行うことができます。

6.2 アダプタのデプロイメント

インストール中、Oracle Application Adapter for J.D. Edwards OneWorld は、WLS コンテナ内の J2CA 1.0 リソース・アダプタとしてデプロイされます。アダプタは、Oracle BPM と同じ WLS コンテナ内にデプロイされる必要があります。

6.3 新規のアプリケーション・サーバー接続の構成

Oracle JDeveloper での新規アプリケーション・サーバー接続の構成方法の詳細は、4-2 ページの [4.3 項「新規のアプリケーション・サーバー接続の構成」](#) を参照してください。

6.4 トランスフォーメーションを使用したサービス統合のためのアウトバウンド BPM プロセスの設計 (J2CA 構成)

この項では、トランスフォーメーションを使用したサービス統合のためのアウトバウンド BPM プロセスの設計方法について説明します。

このアウトバウンド・ユース・ケース・シナリオ用のサンプル・プロジェクトが、次に示す Application Adapter のインストール・フォルダに用意されています。

```
<ADAPTER_HOME>\etc\sample\JDEdwards_Samples.zip\JDEdwards_Samples\BPM\J2CA\Outbound_Project
```

アウトバウンドのデザインタイム構成を完了するには、次のツールが必要です。

- Oracle Adapter アプリケーション・エクスプローラ (アプリケーション・エクスプローラ)
- Oracle JDeveloper BPM デザイナ (JDeveloper)

注意： この章に記載されている例では JDeveloper が使用されています。

この項では、次のトピックについて説明します。

- [6.4.1 項「BPM 用の空のコンポジットの作成」](#)
- [6.4.2 項「BPM アウトバウンド・プロセスの定義」](#)
- [6.4.3 項「12c でのデプロイメントに関する既知の問題への対応」](#)
- [6.4.4 項「BPM アウトバウンド・プロセスのデプロイ」](#)
- [6.4.5 項「Oracle Enterprise Manager Console での入力 XML ドキュメントの起動」](#)

BPM プロセスを設計する前に、アプリケーション・エクスプローラを使用して、各 WSDL ファイルを生成する必要があります。詳細は、4-8 ページの [4.4.1 項「リクエスト/レスポンス型のサービス用の WSDL の生成」](#) を参照してください。

6.4.1 BPM 用の空のコンポジットの作成

BPM 用に空のコンポジットを作成するには、次のステップを実行します。

1. 新規の BPM アプリケーションを作成します。
2. 新規の BPM アプリケーションの名前を入力し、「[次へ](#)」をクリックします。
「プロジェクトの名前付け」ページが表示されます。

3. プロジェクト名を入力し、プロジェクト機能で「BPM」を選択して、「次へ」をクリックします。
「SOA 設定の構成」ページが表示されます。
4. 「コンポジット・テンプレート」リストで「空のコンポジット」を選択し、「終了」をクリックします。

6.4.2 BPM アウトバウンド・プロセスの定義

この項では、BPM アウトバウンド・プロセスの定義方法について説明します。次のトピックで構成されています。

- 6.4.2.1 項「サード・パーティ・アダプタ・サービス・コンポーネントの構成」
- 6.4.2.2 項「アウトバウンド BPM プロセス・コンポーネントの構成」
- 6.4.2.3 項「書込み操作のファイル・アダプタの作成」

6.4.2.1 サード・パーティ・アダプタ・サービス・コンポーネントの構成

サード・パーティ・アダプタ・サービス・コンポーネントを作成するには、次のステップを実行します。

1. 作成したプロジェクトをダブルクリックして、コンポーネントをロードします。
2. 「カスタム / サード・パーティ」ペインから「サード・パーティ・アダプタ」コンポーネントを「外部参照」ペインにドラッグ・アンド・ドロップします。
「サード・パーティ・アダプタ・サービスの作成」ダイアログが表示されます。
3. サード・パーティ・アダプタ・サービスの名前を入力します。
4. 「タイプ」リストで「参照」が選択されていることを確認します (デフォルト)。
5. 「WSDL URL」フィールドの右の「既存の WSDL を検索します。」アイコンをクリックします。
「WSDL チューザ」ダイアログが表示されます。
6. 次のディレクトリまでブラウズし、アウトバウンド WSDL ファイルを選択します。
<ADAPTER_HOME>\wsdls
7. 「OK」をクリックします。
「ファイルのローカライズ」ダイアログが表示されます。
8. 「OK」をクリックします。
アウトバウンド WSDL ファイルおよび関連リクエスト / レスポンス XML スキーマ・ファイル (.xsd) が、作成済のプロジェクト・フォルダにインポートされます。
「サード・パーティ・アダプタ・サービスの作成」ダイアログに戻ります。
9. 「JCA ファイル」フィールドの右の「JCA ファイルを検索します。」アイコンをクリックします。
「<< トランスフォーメーション・マップ >>」ダイアログが表示されます。
10. 次のディレクトリまでブラウズし、JCA プロパティ・ファイルを選択します。
<ADAPTER_HOME>\wsdls
11. 「OK」をクリックします。
「ファイルのコピー」メッセージが表示されます。

12. 「はい」 をクリックします。

JCA プロパティ・ファイルのコピーがプロジェクト・フォルダ内に作成されます。
「サード・パーティ・アダプタ・サービスの作成」 ダイアログに戻ります。

13. 「OK」 をクリックします。

サード・パーティ・アダプタ・サービス・コンポーネントが作成され、「外部参照」 ペインに表示されます。

これで、アウトバウンド BPM プロセス・コンポーネントを構成する準備ができました。

スクリーン・ショットを含む詳細は、4-11 ページの 4.4.3.1 項「サード・パーティ・アダプタ・サービス・コンポーネントの構成」を参照してください。

6.4.2.2 アウトバウンド BPM プロセス・コンポーネントの構成

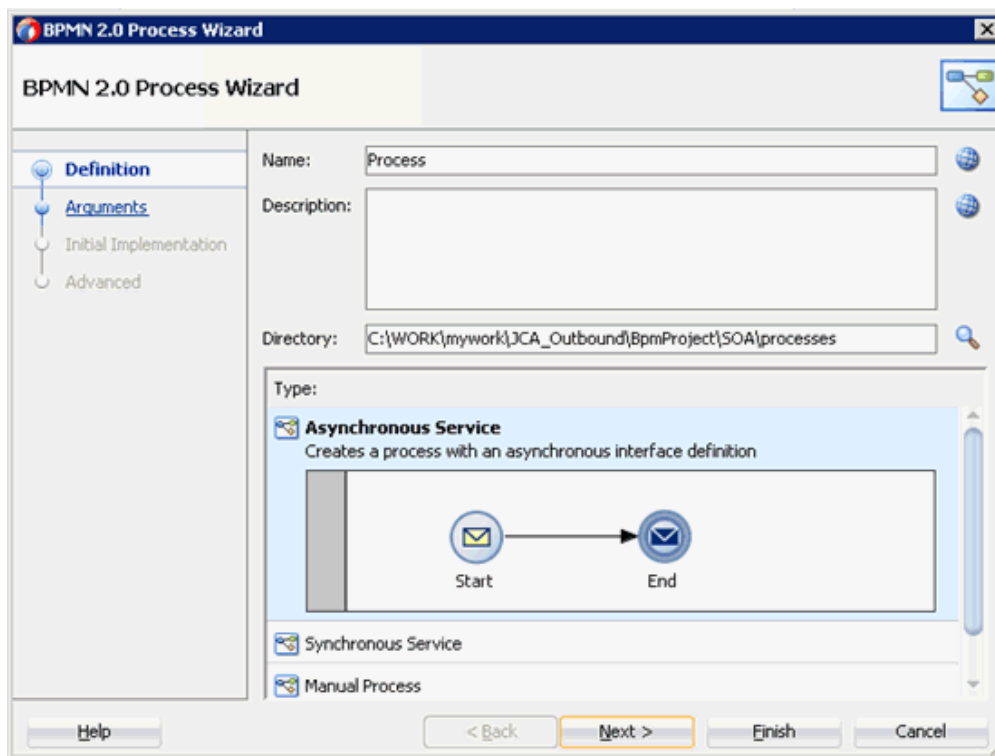
この項では、アウトバウンド BPM プロセス・コンポーネントの構成方法について説明します。

アウトバウンド BPM プロセス・コンポーネントを構成するには、次のステップを実行します。

1. 「コンポーネント」 ペインから「BPMN プロセス」 コンポーネントを「コンポーネント」 ペインにドラッグ・アンド・ドロップします。

図 6-1 に示すように、「BPMN プロセスの作成」 ダイアログが表示されます。

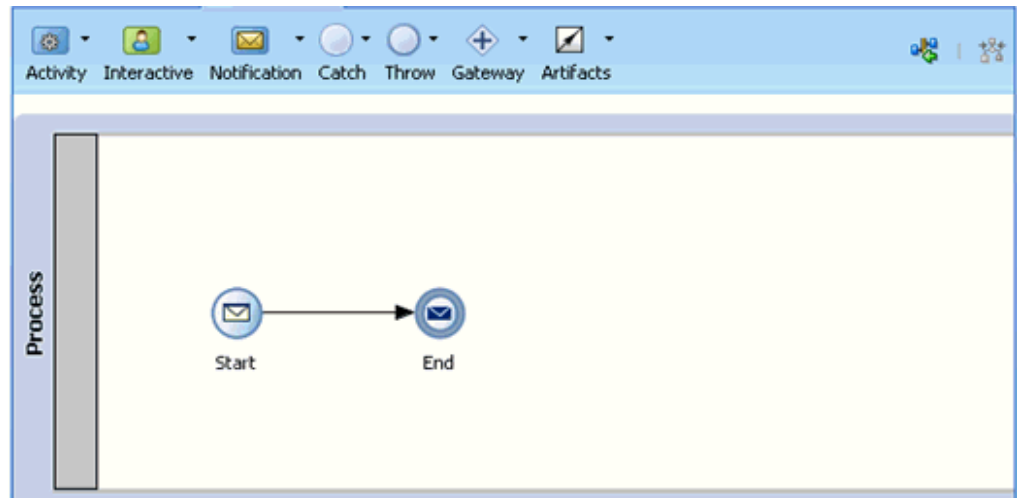
図 6-1 「BPMN プロセスの作成」 ダイアログ



2. 「タイプ」 領域で選択されているデフォルトのオプション (「非同期サービス」) を受け入れ、「終了」 をクリックします。

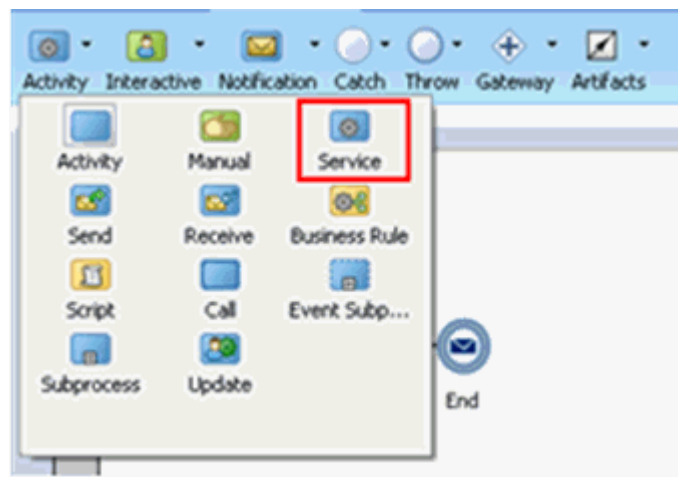
図 6-2 に示すように、BPMN プロセスが表示されます。

図 6-2 BPMN プロセス



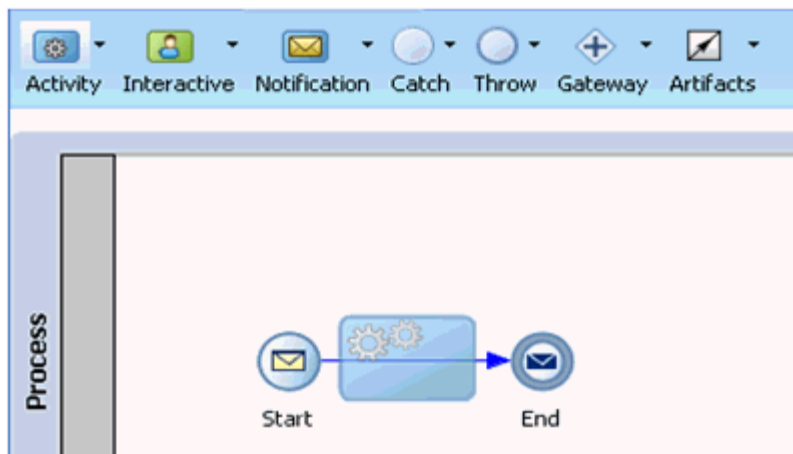
3. 図 6-3 に示すように、「アクティビティ」ドロップダウン・メニューをクリックし、「サービス」を選択します。

図 6-3 「アクティビティ」ドロップダウン・メニュー



4. 図 6-4 に示すように、「Start」と「End」イベント・コンポーネント間の伝送路に「サービス」アイコンをドロップします。

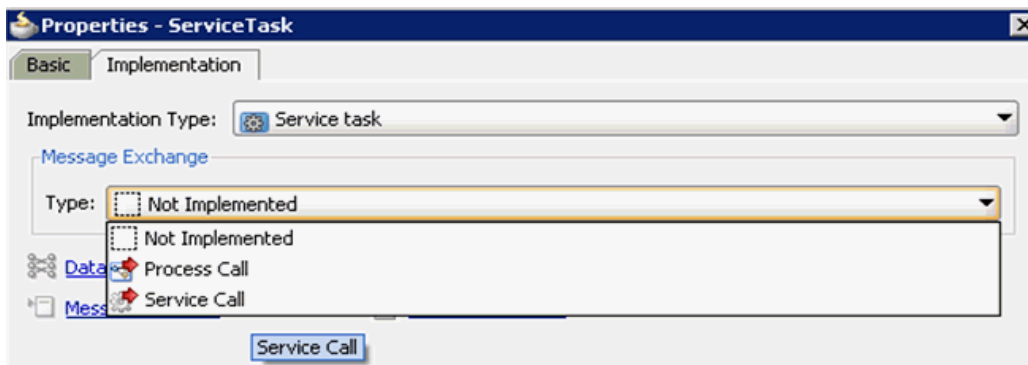
図 6-4 「アクティビティ」アイコン



「プロパティ - ServiceTask」ウィンドウが表示されます。

5. 「実装」タブをクリックします。
6. 図 6-5 に示すように、「メッセージ交換」の「タイプ」リストから「サービス・コール」を選択します。

図 6-5 サービス・コール



7. 図 6-6 に示すように、「サービス」フィールドの右の「参照」アイコンをクリックします。

図 6-6 「参照」アイコン

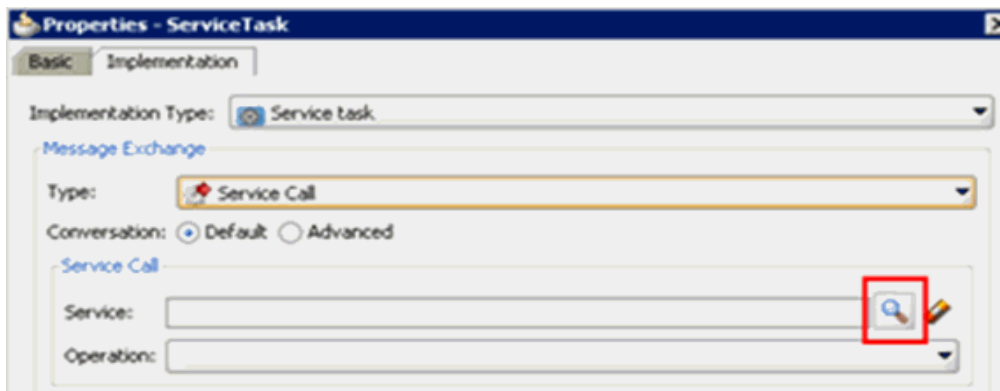
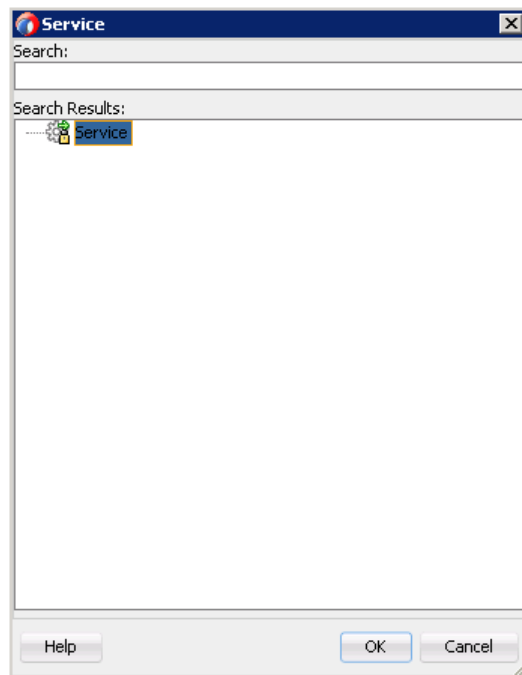


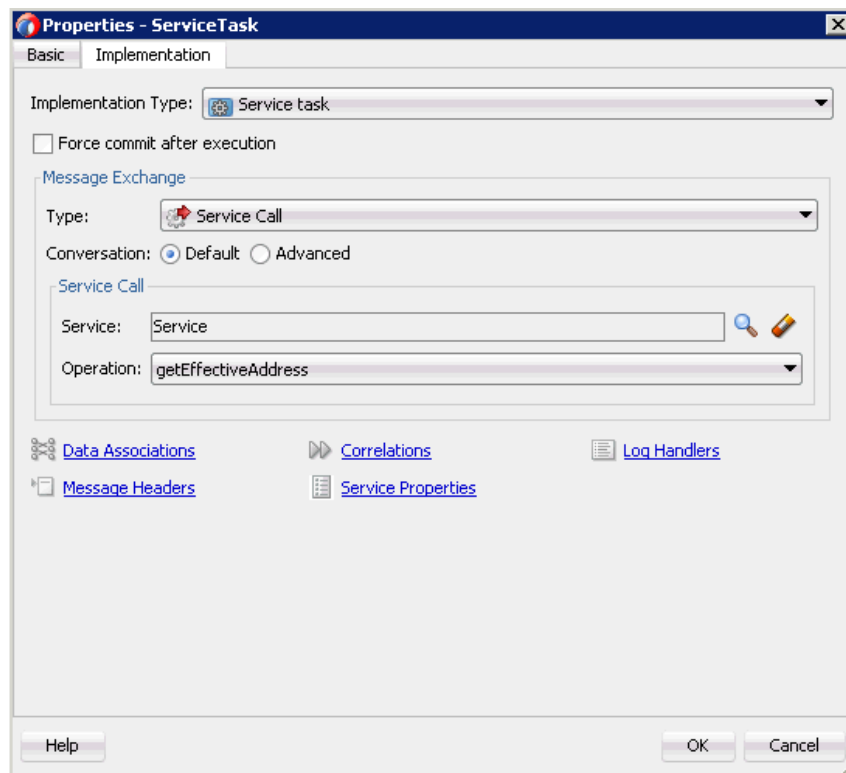
図 6-7 に示すように、「サービス」ダイアログが表示されます。

図 6-7 「サービス」 ダイアログ



- 作成されたサード・パーティ・サービスを選択し、「OK」をクリックします。
図 6-8 に示すように、「プロパティ - ServiceTask」ダイアログに戻ります。

図 6-8 「プロパティ - ServiceTask」ダイアログ



- 「データ・アソシエーション」ハイパーリンクをクリックします。

「データ・アソシエーション」ダイアログが表示されます。

10. 図 6-9 に示すように、左ペインで「プロセス」の下の「データ・オブジェクト」ノードを右クリックし、「新規」を選択します。

図 6-9 「新規」オプション

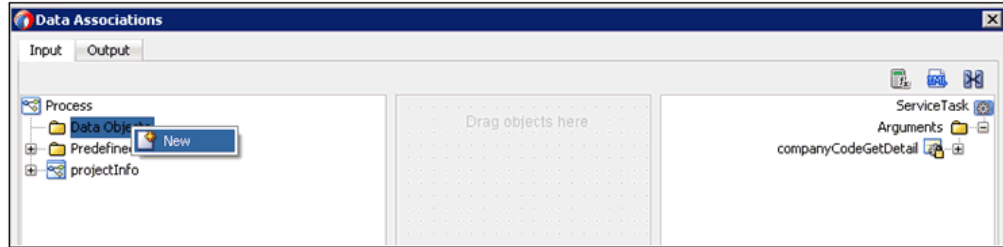
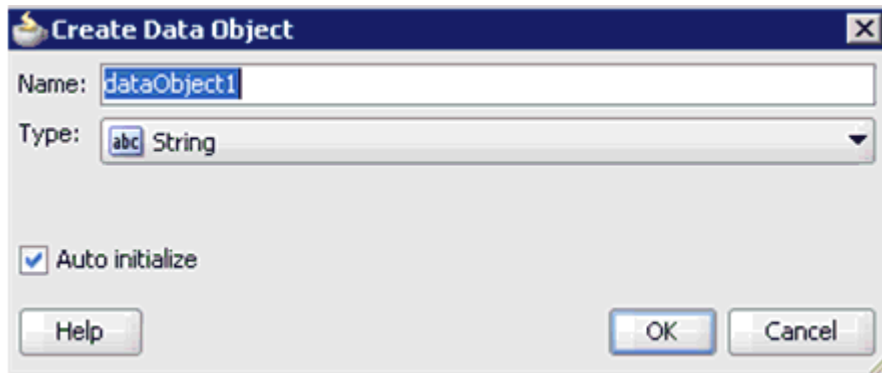


図 6-10 に示すように、「データ・オブジェクトの作成」ダイアログが表示されます。

図 6-10 「データ・オブジェクトの作成」ダイアログ



11. 図 6-11 に示すように、「名前」フィールドに名前を入力し（「Request」など）、「タイプ」フィールドのドロップダウン・ボタンをクリックして、リストから「参照」を選択します。

図 6-11 「データ・オブジェクトの作成」ダイアログ

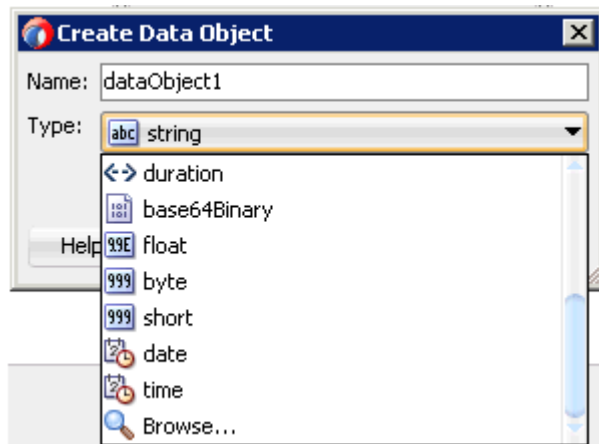
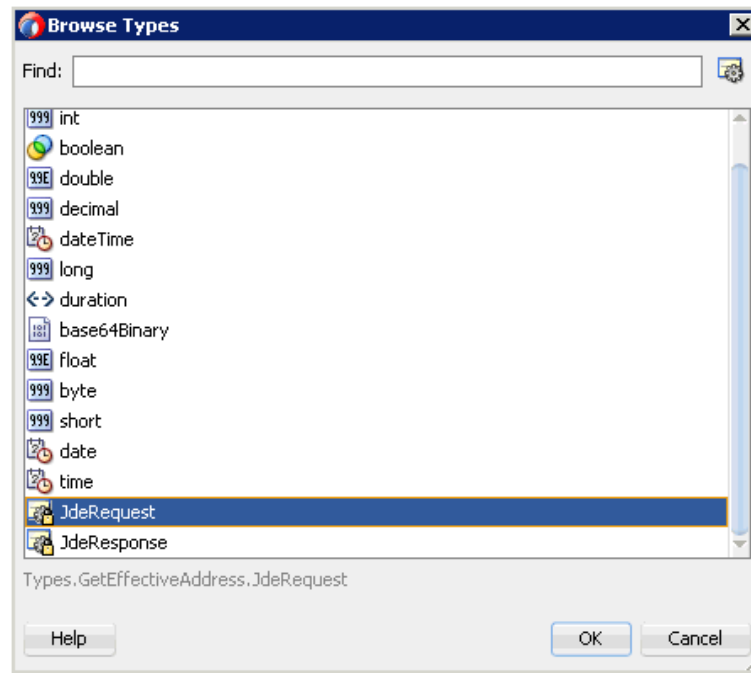


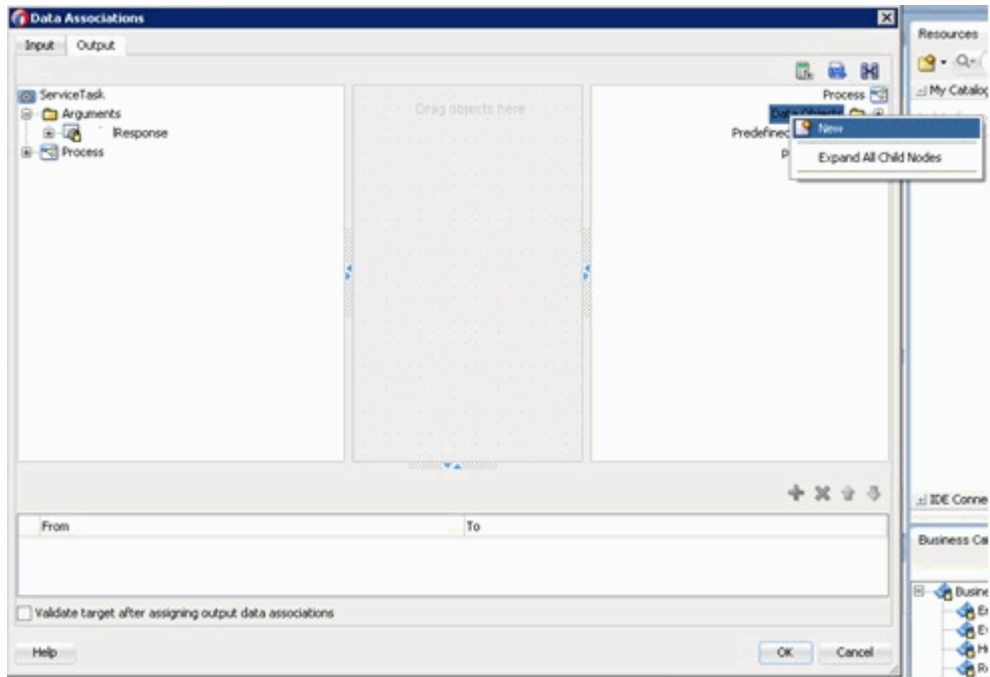
図 6-12 に示すように、「参照タイプ」ダイアログが表示されます。

図 6-12 「参照タイプ」 ダイアログ



12. 最初のコンポーネント (「JdeRequest」など) を選択し、「OK」をクリックします。
「データ・オブジェクトの作成」ダイアログに戻ります。
13. 「OK」をクリックします。
作成されたデータ・オブジェクト (「Request」など) が「データ・アソシエーション」ダイアログの「データ・オブジェクト」ノードの下に表示されます。
14. 図 6-13 に示すように、「出力」タブの右ペインで「データ・オブジェクト」ノードを右クリックし、「新規」を選択して、データ・オブジェクトをもう 1 つ作成します。

図 6-13 「データ・アソシエーション」 ダイアログ

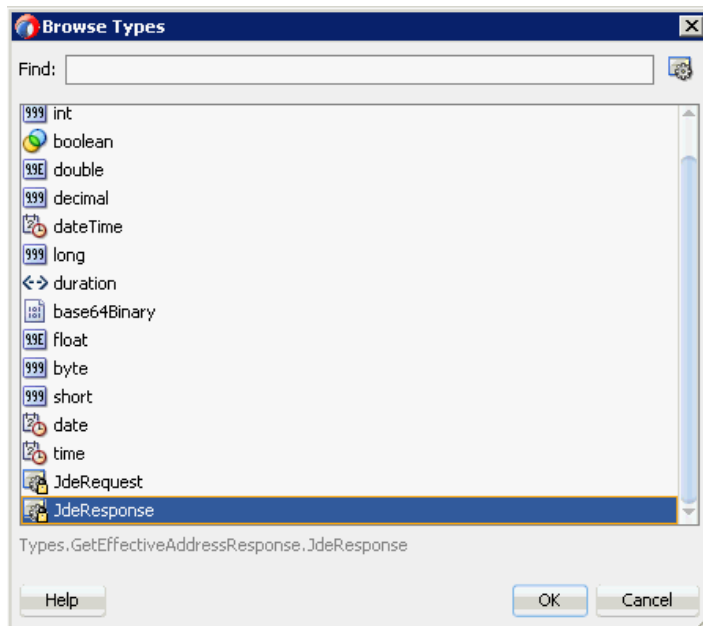


「データ・オブジェクトの作成」ダイアログが表示されます。

15. 「名前」フィールドに名前を入力し(「Response」など)、「タイプ」フィールドのドロップダウン・ボタンをクリックして、リストから「参照」を選択します。

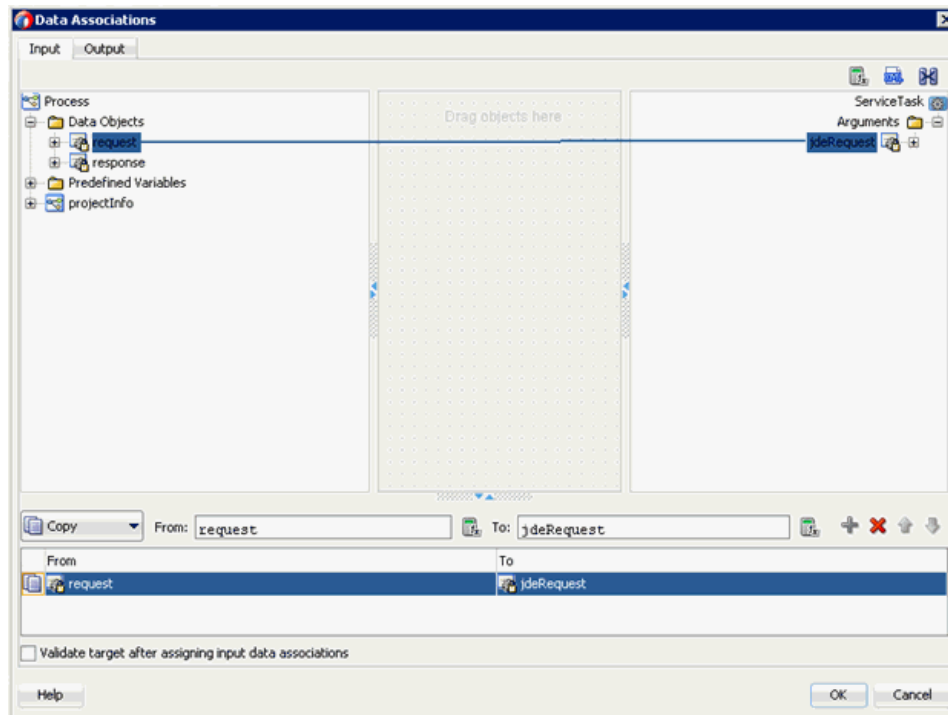
図 6-14 に示すように、「参照タイプ」ダイアログが表示されます。

図 6-14 「参照タイプ」ダイアログ



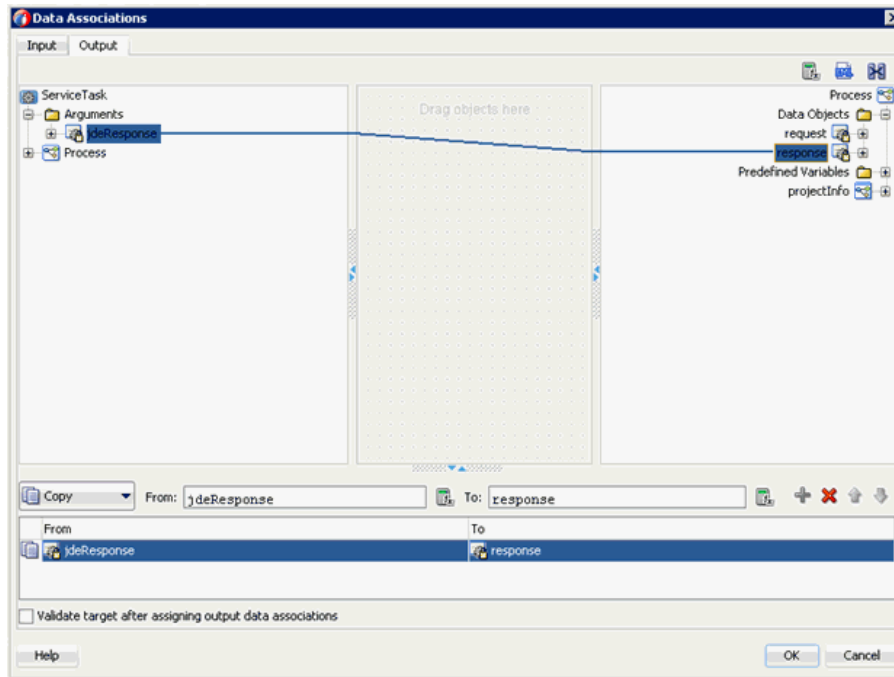
16. 2 番目のコンポーネント (「jdeResponse」など) を選択し、「OK」をクリックします。
「データ・オブジェクトの作成」ダイアログに戻ります。
17. 「OK」をクリックします。
作成されたデータ・オブジェクト (「Response」など) が「データ・アソシエーション」ダイアログの「プロセス」ノードの下に表示されます。
18. 図 6-15 に示すように、「入力」タブの左ペインで「データ・オブジェクト」ノードの下の「Request」データ・オブジェクトを選択し、右ペインの「引数」ノードの下の「JdeRequest」にドラッグして接続します。

図 6-15 「Request」データ・オブジェクト



19. 図 6-16 に示すように、「出力」タブをクリックし、左ペインの「引数」ノードの下の「jdeResponse」を選択し、「データ・オブジェクト」ノードの下の「Response」データ・オブジェクトにドラッグして接続します。

図 6-16 「Response」 データ・オブジェクト



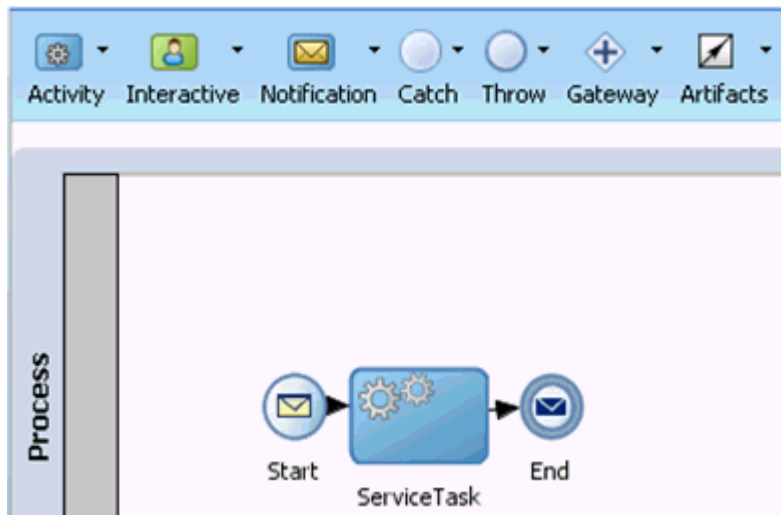
20. 「OK」をクリックします。

「プロパティ - ServiceTask」ダイアログに戻ります。

21. 「OK」をクリックします。

図 6-17 に示すように、「Start」と「End」イベント・コンポーネント間に「サービス・タスク」が作成されます。

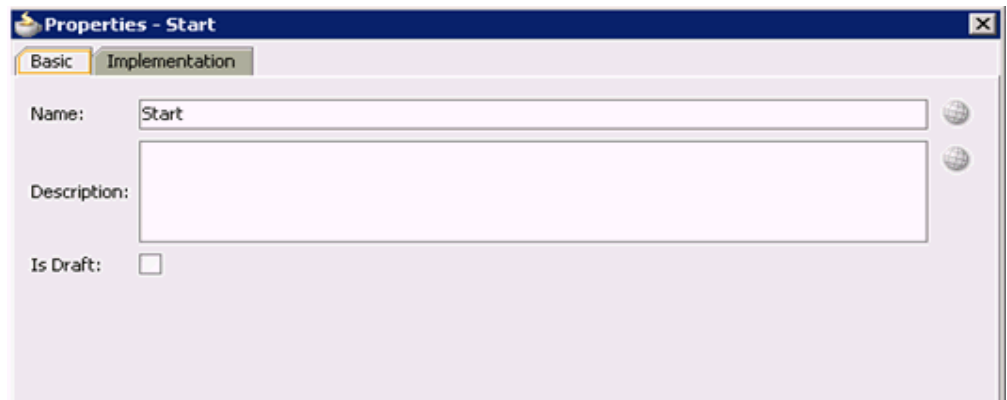
図 6-17 サービス・タスク



22. プロセスを保存し、「Start」イベント・コンポーネントをダブルクリックします。

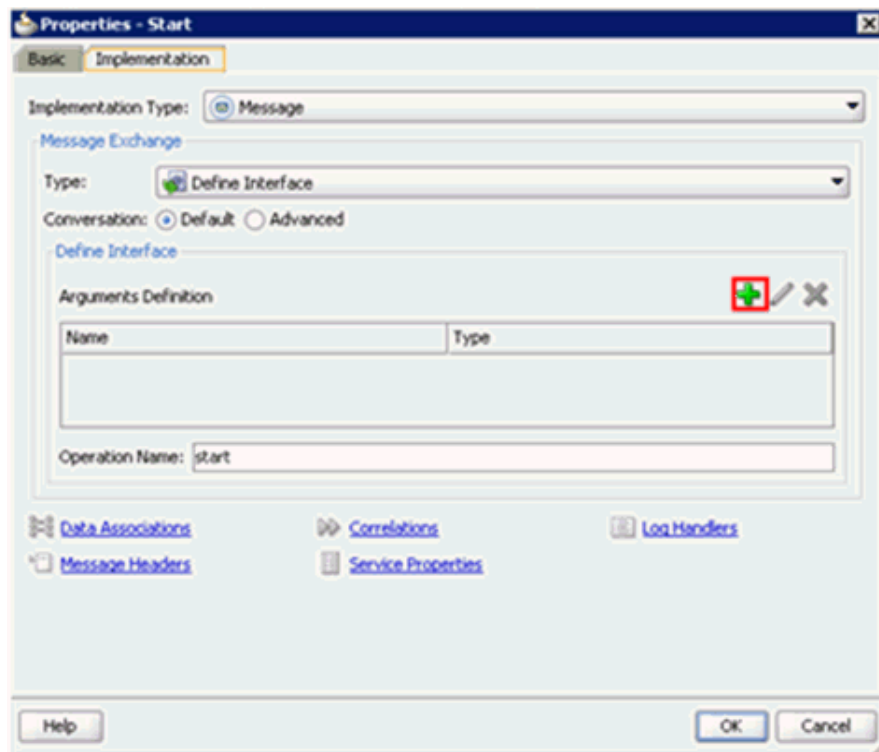
図 6-18 に示すように、「プロパティ - Start」ダイアログが表示されます。

図 6-18 「プロパティ - Start」 ダイアログ



23. 図 6-19 に示すように、「実装」タブをクリックします。

図 6-19 「実装」タブ



24. 「引数の定義」フィールドの右の「プラス」アイコンをクリックします。
「引数の作成」ダイアログが表示されます。
25. 図 6-20 に示すように、「名前」フィールドに名前を入力し (デフォルトでは「argument1」)、 「タイプ」フィールドのドロップダウン・ボタンをクリックして、リストから「参照」を選択します。

図 6-20 「引数の作成」 ダイアログ

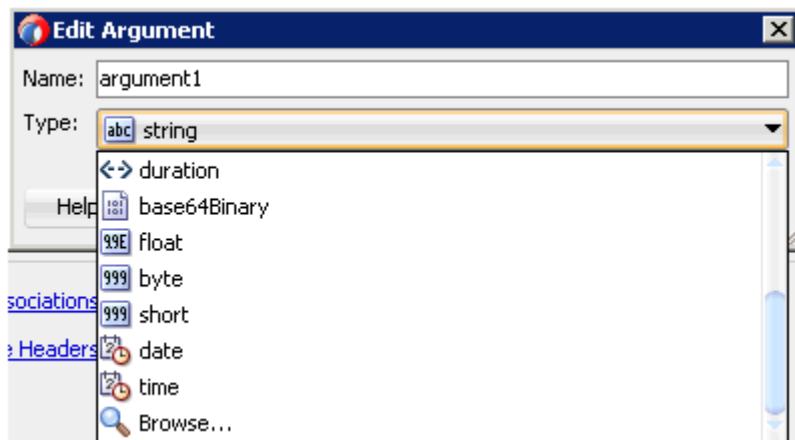
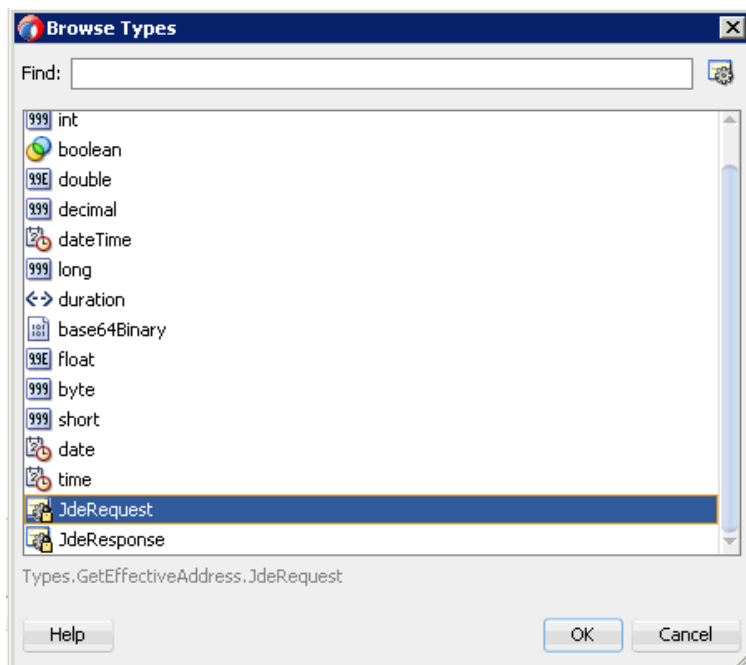


図 6-21 に示すように、「参照タイプ」ダイアログが表示されます。

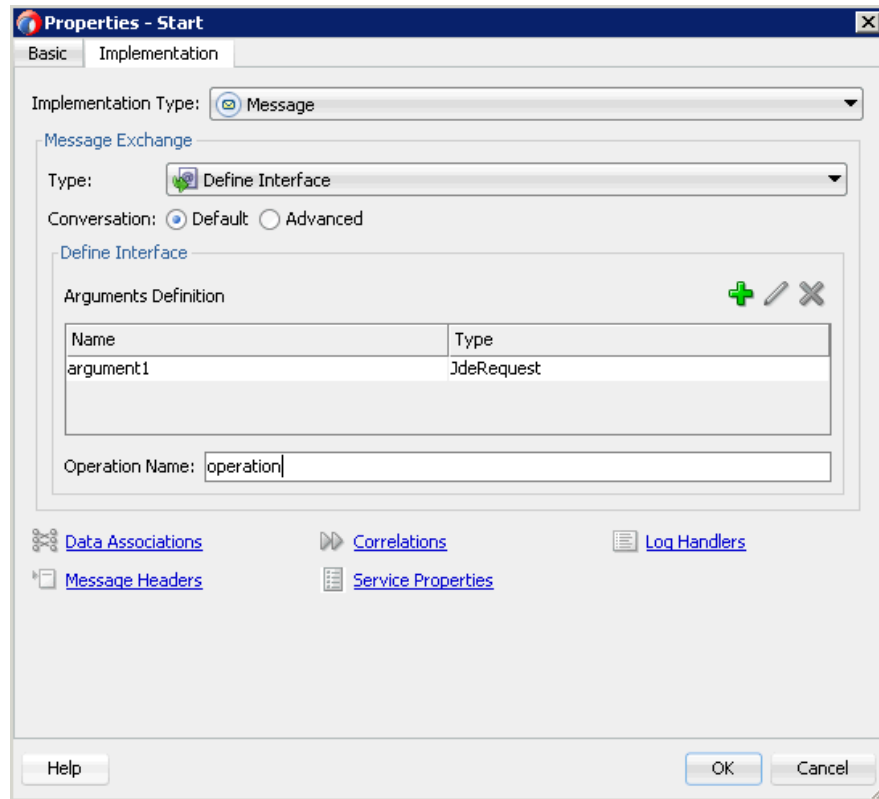
図 6-21 「参照タイプ」ダイアログ



26. 最初のコンポーネント (「JdeRequest」など) を選択し、「OK」をクリックします。
「引数の作成」ダイアログに戻ります。
27. 「OK」をクリックします。
「プロパティ - Start」ダイアログに戻ります。
28. 図 6-22 に示すように、「操作名」フィールドで「start」(デフォルト)を「operation」に変更します。

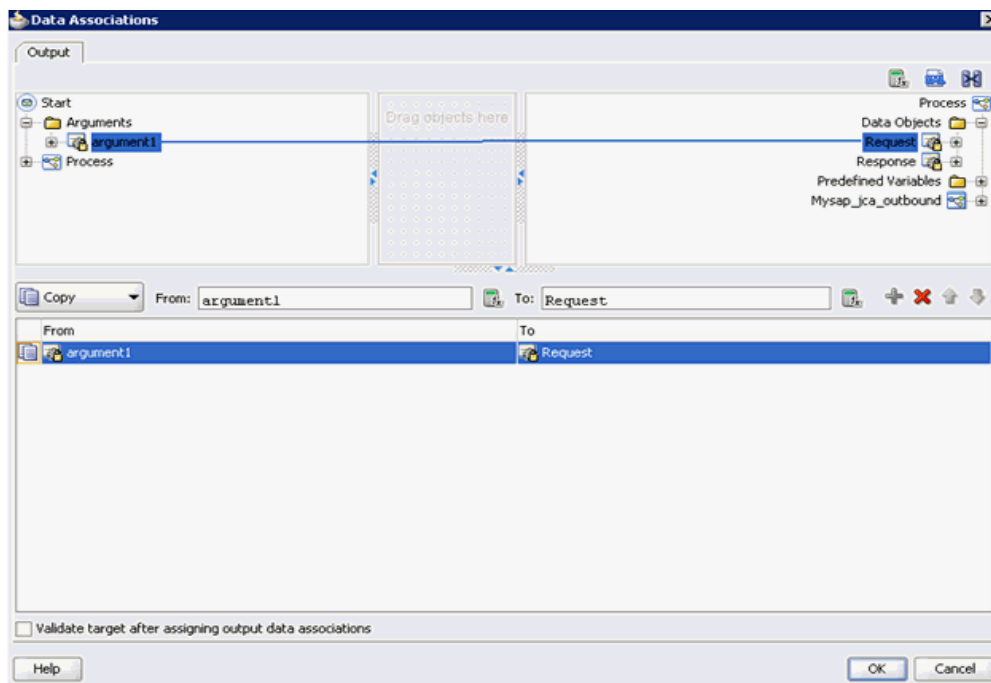
注意: この変更は、古い BPM ペイロードと連携させるために必要です。

図 6-22 「操作名」 フィールド



29. 「データ・アソシエーション」ハイパーリンクをクリックします。
「データ・アソシエーション」ダイアログが表示されます。
30. 左ペインで「引数」ノードの下の「arguments1」を選択し、右ペインの「データ・オブジェクト」ノードの下の「Request」データ・オブジェクトにドラッグして接続します。
31. 図 6-23 に示すように、「OK」をクリックします。

図 6-23 「OK」 ボタン

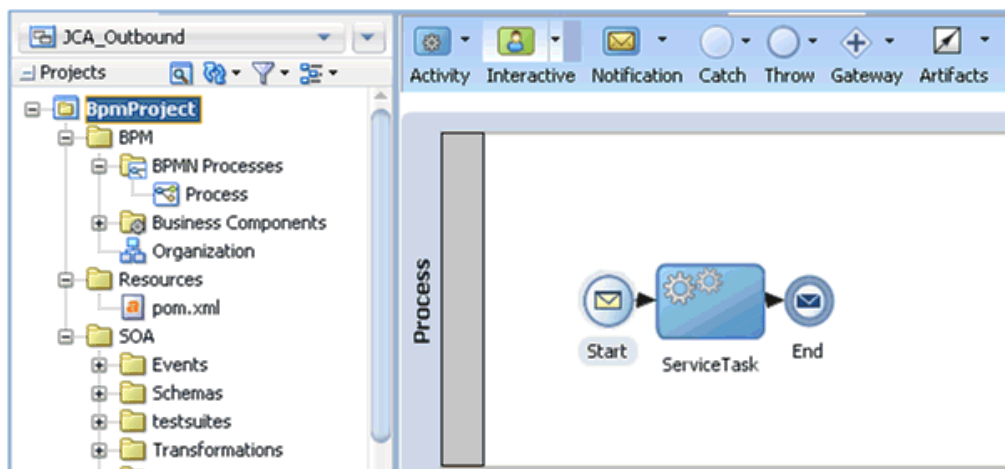


「プロパティ - Start」 ダイアログに戻ります。

32. 「OK」 をクリックします。

図 6-24 に示すように、「プロセス」 ワークスペース領域に戻ります。

図 6-24 「プロセス」 ワークスペース領域



33. 作成したプロジェクトをダブルクリックして、コンポーネントをロードします。
34. メニュー・バーで「すべて保存」アイコンをクリックし、構成された新規のアウトバウンド BPM プロセス・コンポーネントを保存します。
- これで書込み操作作用のファイル・アダプタを作成する準備ができました。

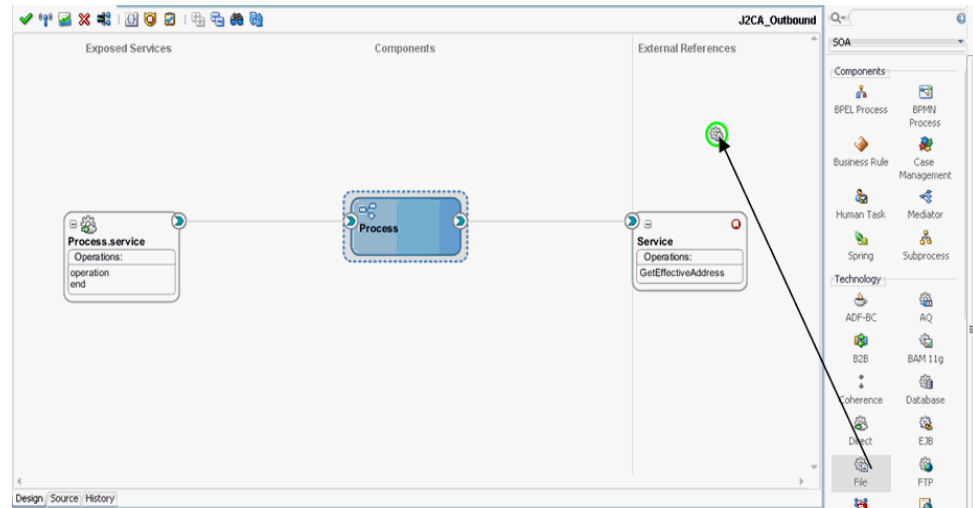
6.4.2.3 書込み操作作用のファイル・アダプタの作成

この項では、書込み操作作用のファイル・アダプタの作成方法について説明します。

書き込み操作のファイル・アダプタを作成するには、次のステップを実行します。

1. 図 6-25 に示すように、「テクノロジー・アダプタ」ペインから「ファイル・アダプタ」コンポーネントを「外部参照」ペインにドラッグ・アンド・ドロップします。

図 6-25 「ファイル・アダプタ」コンポーネント

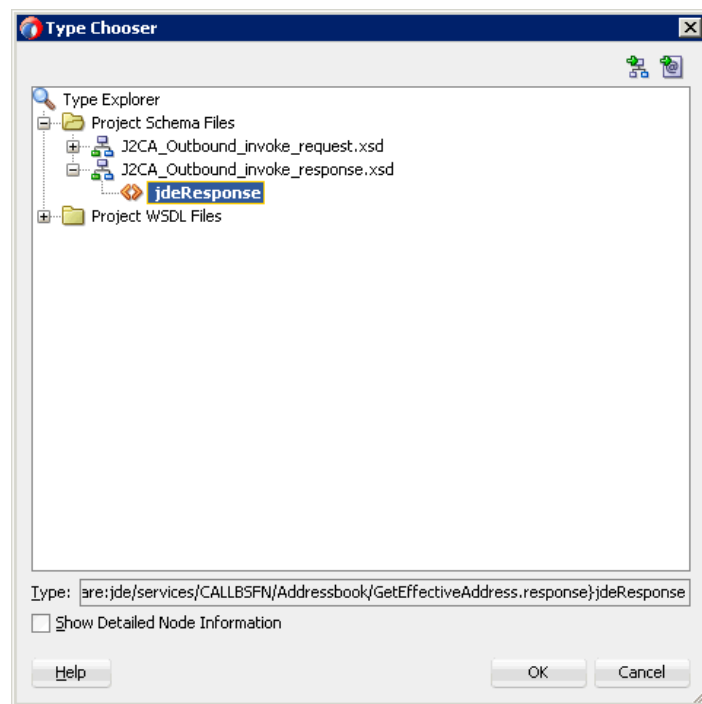


アダプタ構成ウィザードが表示されます。

2. 「参照名」(「FileWrite」など)を入力します。
3. 「次へ」をクリックします。
「アダプタ・インタフェース」ページが表示されます。
4. 「操作およびスキーマから定義(後で指定)」オプションが選択されていることを確認します。
5. 「次へ」をクリックします。
「ファイル・サーバーの接続」ページが表示されます。
6. 「次へ」をクリックします。
「操作」ページが表示されます。
7. 「操作タイプ」オプションのリストから「**Write File**」を選択し、「操作名」(「Write」など)を指定します。
8. 「次へ」をクリックします。
「ファイル構成」ページが表示されます。
9. 出力ファイルの書き込み先となるファイル・システム上の場所を指定します。
10. ファイル・ネーミング規則(po_%SEQ%.txt)フィールドで、出力ファイルの名前を指定します。
11. 「次へ」をクリックします。
「メッセージ」ページが表示されます。
12. 「URL」フィールドの右の「参照」をクリックします。

図 6-26 に示すように、「タイプ・チューザ」ダイアログが表示されます。

図 6-26 「タイプ・チューザ」ダイアログ



13. 「プロジェクトのスキーマ・ファイル」、「J2CA_Outbound_invoke_response.xsd」の順に展開します。

14. 使用可能なスキーマを選択します (「jdeResponse」 など)。

15. 「OK」をクリックします。

「メッセージ」ページに戻ります。

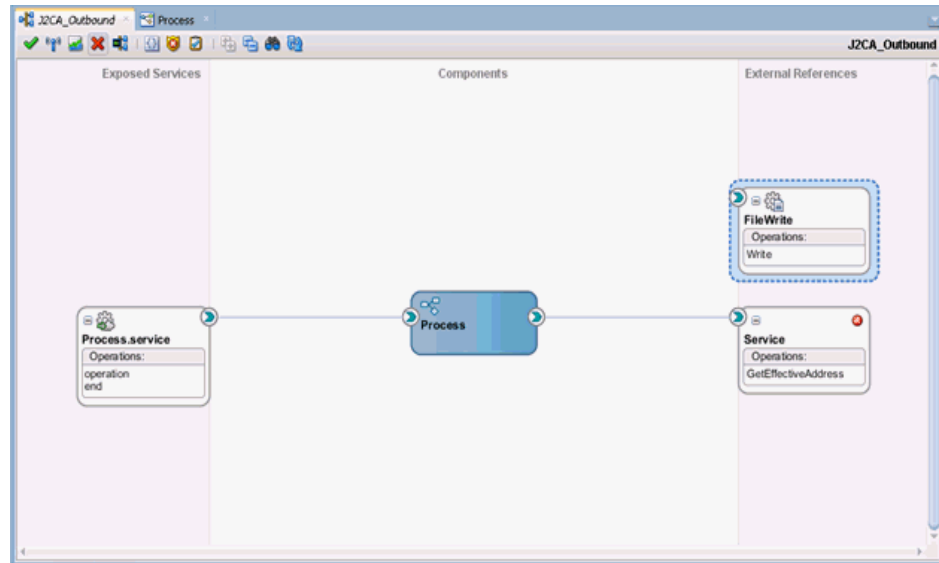
16. 「次へ」をクリックします。

「終了」ページが表示されます。

17. 「終了」をクリックします。

図 6-27 に示すように、「外部参照」ペインでファイル・アダプタ・サービスが作成されます。

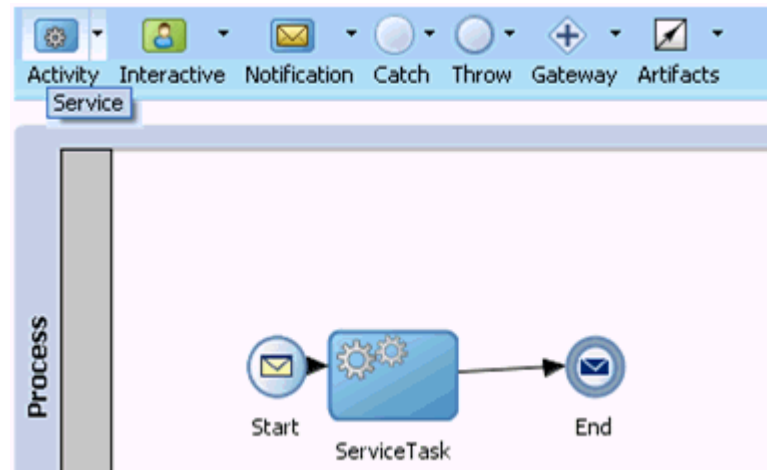
図 6-27 ファイル・アダプタ・サービス



18. BPMN プロセス・コンポーネントをダブルクリックします。

図 6-28 に示すように、BPMN プロセスが表示されます。

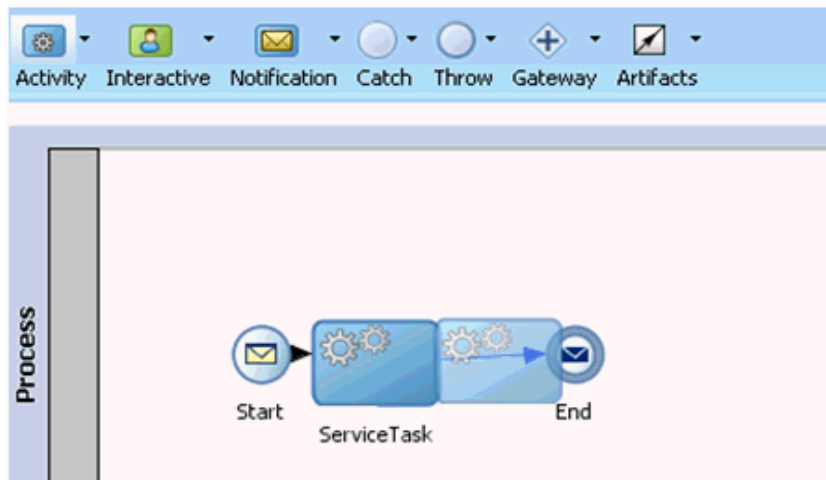
図 6-28 BPMN プロセス



19. 「アクティビティ」アイコンをクリックし、「サービス」を選択します。

20. 図 6-29 に示すように、「サービス・タスク」と「End」イベント・コンポーネント間の伝送路に「サービス」アイコンをドロップします。

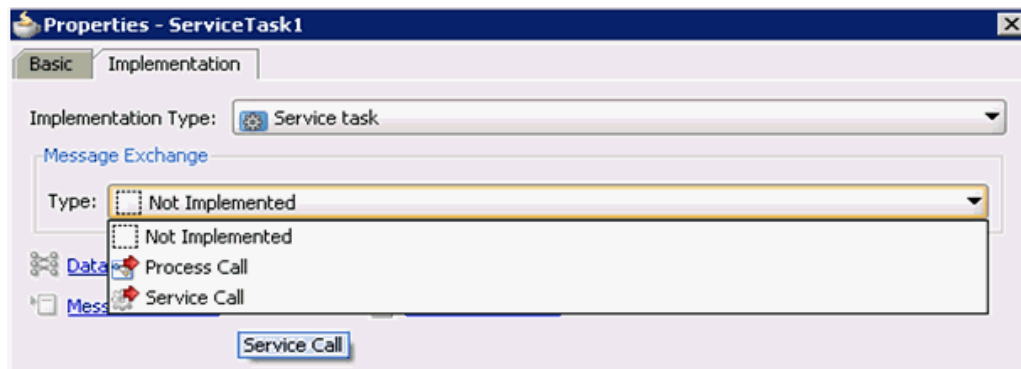
図 6-29 「アクティビティ」アイコン



「プロパティ - ServiceTask1」ダイアログが表示されます。

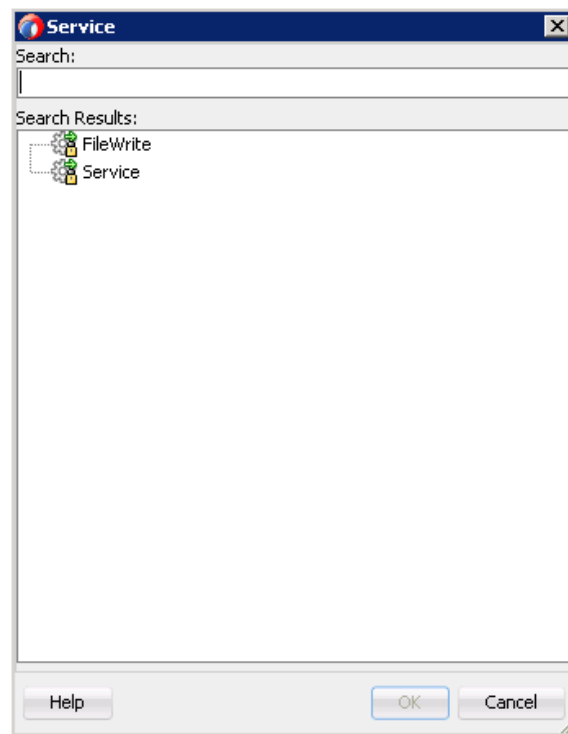
21. 「実装」タブをクリックします。
22. 図 6-30 に示すように、「メッセージ交換」セクションの「タイプ」ドロップダウン・リストから「サービス・コール」を選択します。

図 6-30 サービス・コール



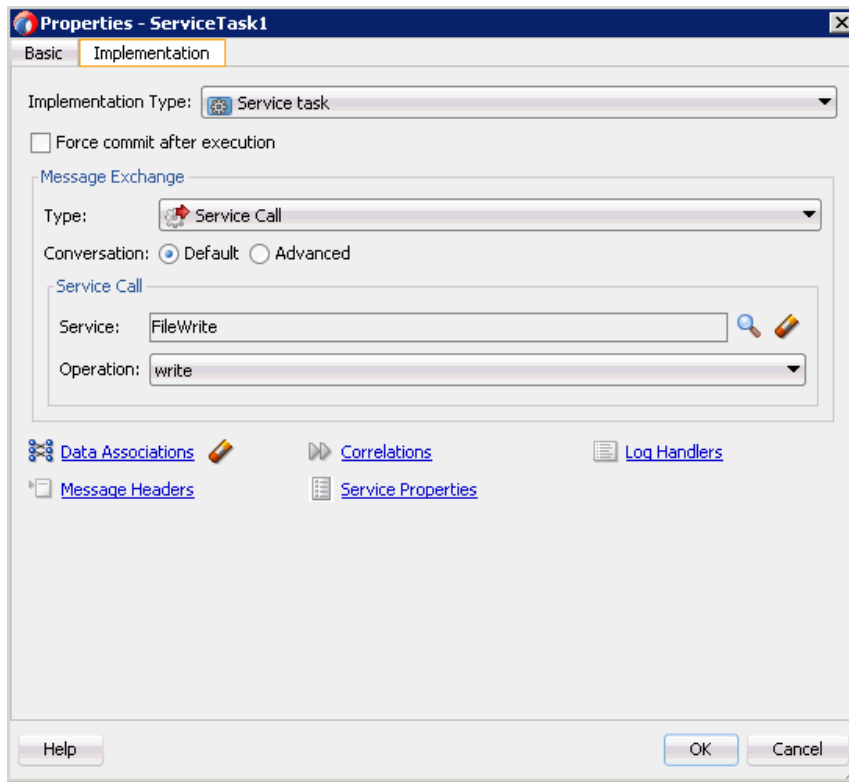
23. 「サービス」フィールドの右の「参照」アイコンをクリックします。
 図 6-31 に示すように、「サービス」ダイアログが表示されます。

図 6-31 「サービス」 ダイアログ



24. 作成された書き込み操作のサービスを選択し (「FileWrite」など)、「OK」をクリックします。
図 6-32 に示すように、「プロパティ - ServiceTask1」ダイアログに戻ります。

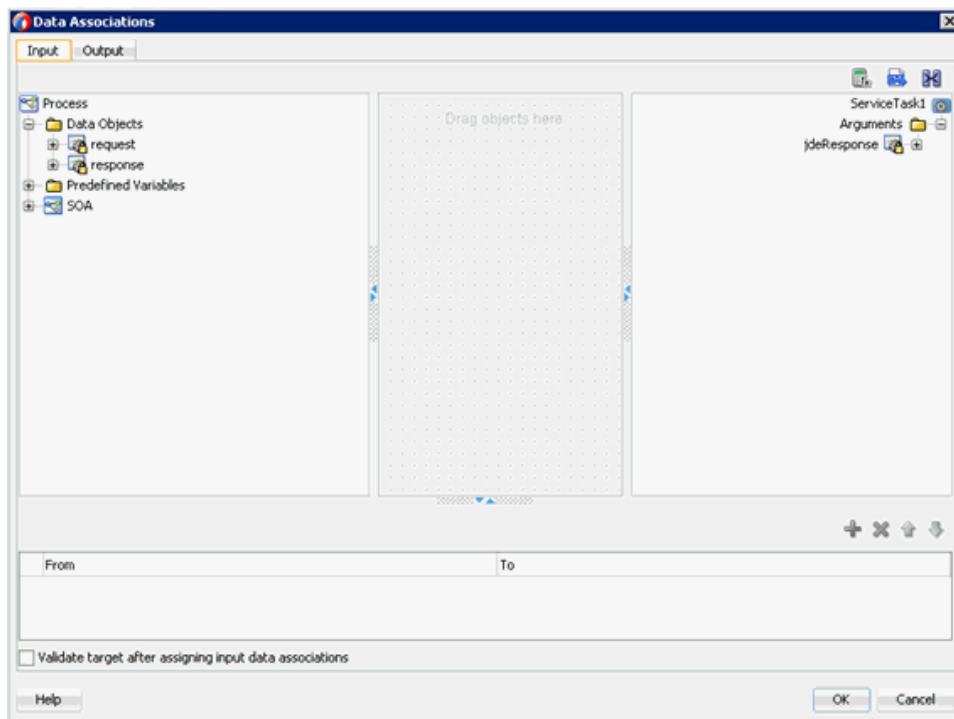
図 6-32 「プロパティ - ServiceTask1」 ダイアログ



25. 「データ・アソシエーション」ハイパーリンクをクリックします。

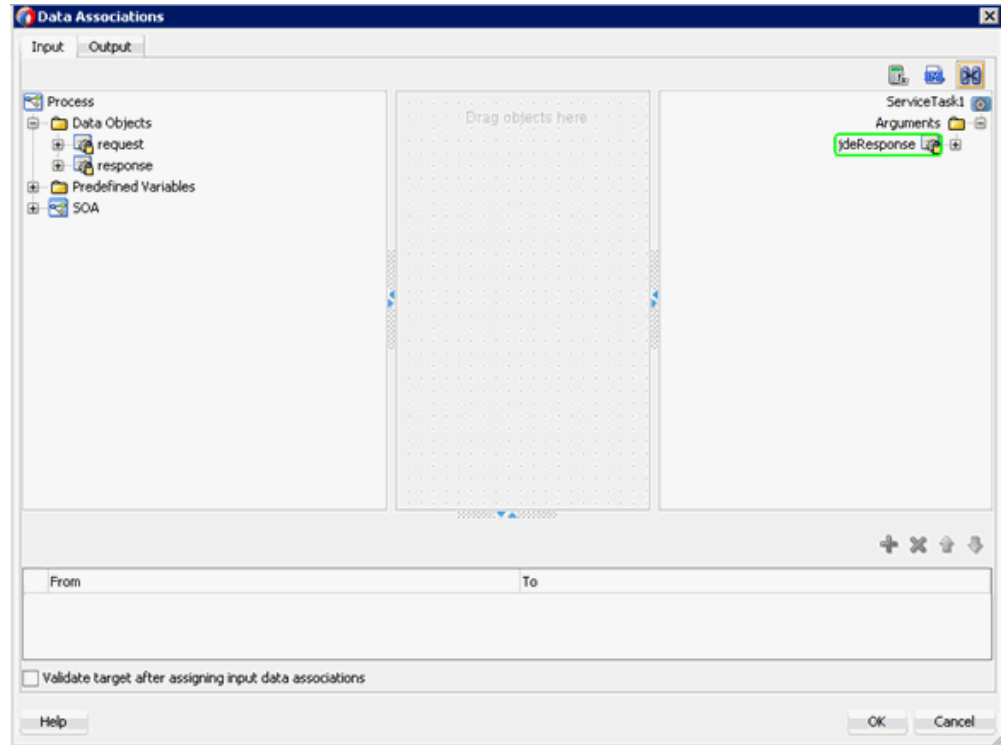
図 6-33 に示すように、「データ・アソシエーション」ダイアログが表示されます。

図 6-33 「データ・アソシエーション」ダイアログ



26. 「入力」タブで、右ペインの上部にある「XSL 変換」アイコンをクリックします。
27. 図 6-34 に示すように、「XSL 変換」アイコンを「jdeResponse」ノードにドラッグ・アンド・ドロップします。

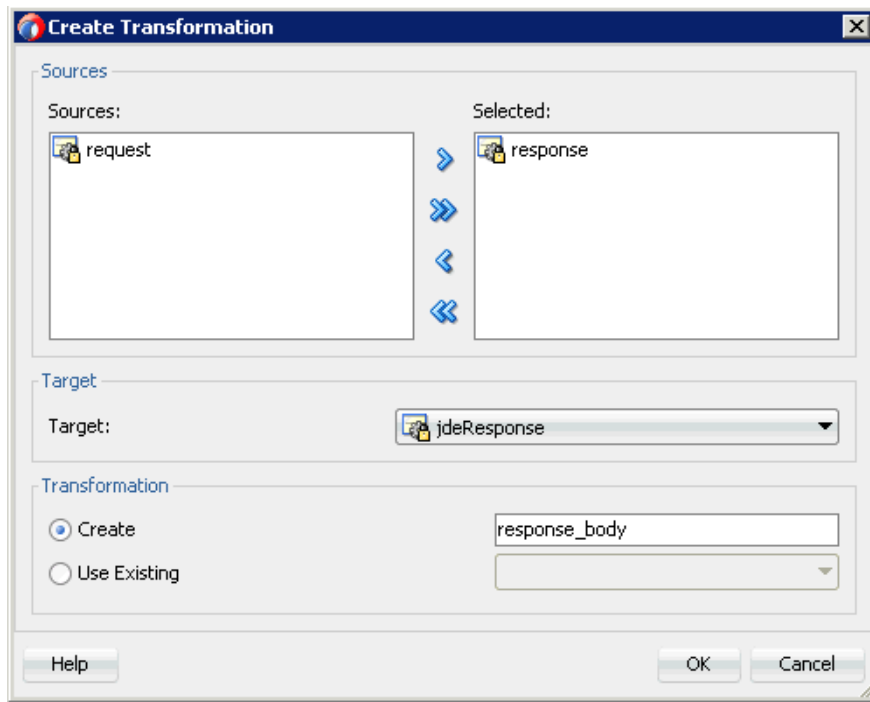
図 6-34 「CompanyCodeJDEResponse」ノード



「トランスフォーメーションの作成」ダイアログが表示されます。

28. 「ソース」セクションで「Response」を選択し、右矢印記号をクリックします。
 図 6-35 に示すように、「Response」オブジェクトが「選択された要素」領域に追加されます。

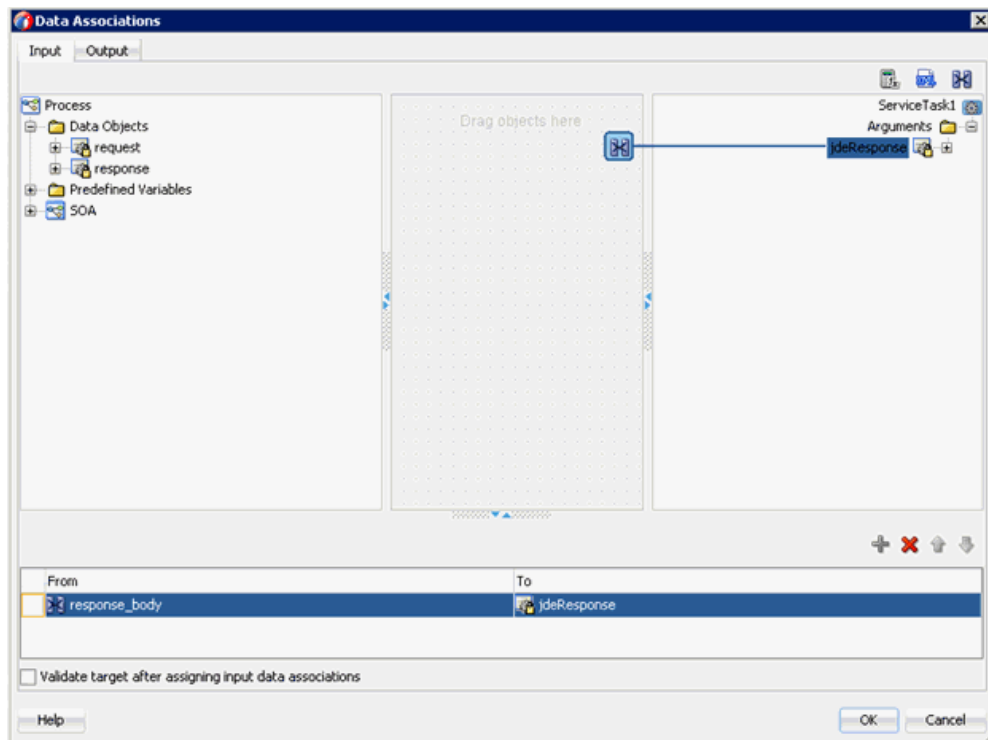
図 6-35 「Response」 オブジェクト



29. 「OK」 をクリックして、「ターゲット」 ドロップダウン・リストで選択されたデフォルト値と「作成」 フィールドのデフォルト名を受け入れます。

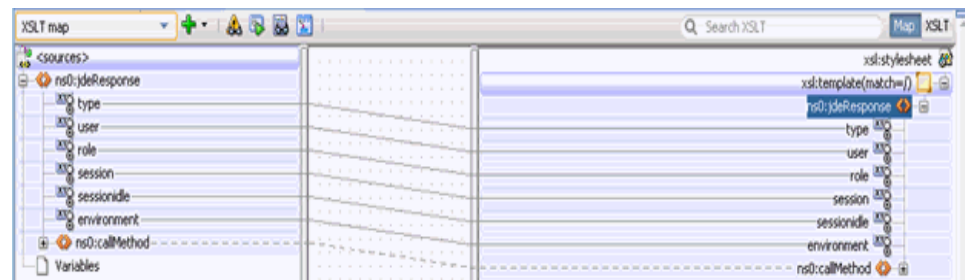
図 6-36 に示すように、XSL 変換が作成された「データ・アソシエーション」ダイアログ・ウィンドウに戻ります。

図 6-36 「データ・アソシエーション」 ダイアログ



30. 「OK」をクリックします。
「プロパティ - ServiceTask1」 ダイアログに戻ります。
31. 「OK」をクリックします。
「Response_body.xml」 タブが表示されます。
32. ソース要素とターゲット要素を自動マップします。
「自動マップ・プリファレンス」 ダイアログが表示されます。
33. デフォルト値を受け入れ、「OK」をクリックします。
図 6-37 に示すように、トランスフォーメーションが完成します。

図 6-37 完成したトランスフォーメーション



34. トランスフォーメーションを保存します。
35. 「プロセス」 ワークスペース領域に戻ります。
「ServiceTask」 コンポーネントと「End」 イベント・コンポーネント間に「ServiceTask1」 コンポーネントが作成されます。
36. メニュー・バーで「すべて保存」アイコンをクリックし、構成された新規のアウトバウンド BPM プロセス・コンポーネントを保存します。
これで、アウトバウンド BPM プロセスをデプロイする準備ができました。

6.4.3 12c でのデプロイメントに関する既知の問題への対応

12c でのデプロイメントに関する既知の問題への対応方法の詳細は、4-26 ページの 4.4.3.3 項「12c でのデプロイメントに関する既知の問題への対応」を参照してください。

6.4.4 BPM アウトバウンド・プロセスのデプロイ

メディアエータ・アウトバウンド・プロセスをデプロイするには、次のステップを実行します

1. 左ペインでプロジェクト名を右クリックし、「デプロイ」を選択して、「J2CA_Outbound」をクリックします。
「デプロイメント・アクション」 ページが表示されます。
2. 「アプリケーション・サーバーにデプロイ」が選択されていることを確認します。
3. 「次へ」をクリックします。
「構成のデプロイ」 ページが表示されます。
4. デフォルト値を選択したままにし、「次へ」をクリックします。
「サーバーの選択」 ページが表示されます。

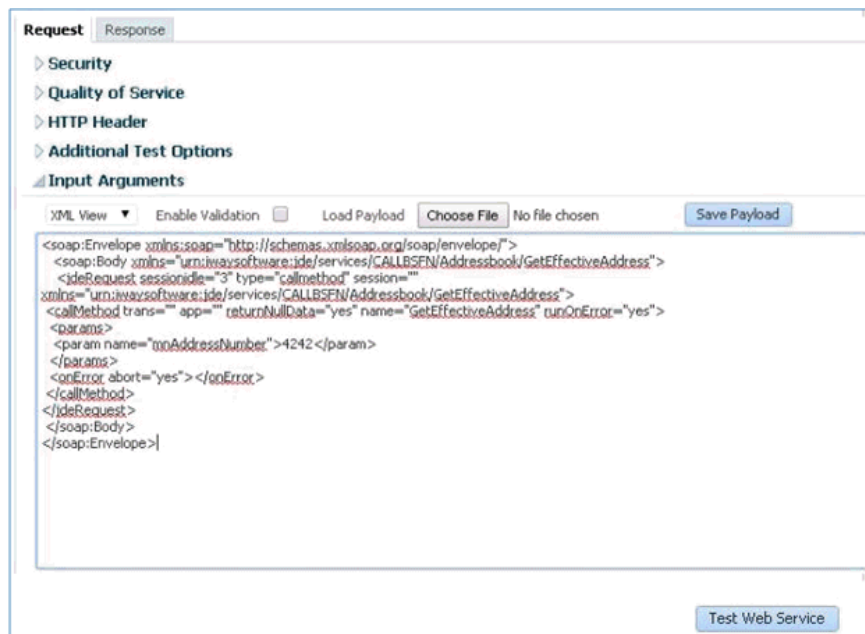
- 構成済の使用可能なアプリケーション・サーバーを選択し、「次へ」をクリックします。
「SOA サーバー」ページが表示されます。
- ターゲットの SOA サーバーを選択し、「次へ」をクリックします。
「サマリー」ページが表示されます。
- プロジェクトに使用可能なすべてのデプロイメント情報をレビューおよび確認し、「終了」をクリックします。
詳細は、4-28 ページの 4.4.4 項「BPEL アウトバウンド・プロセスのデプロイ」を参照してください。

6.4.5 Oracle Enterprise Manager Console での入力 XML ドキュメントの起動

Oracle Enterprise Manager Console で入力 XML ドキュメントを起動するには、次のステップを実行します

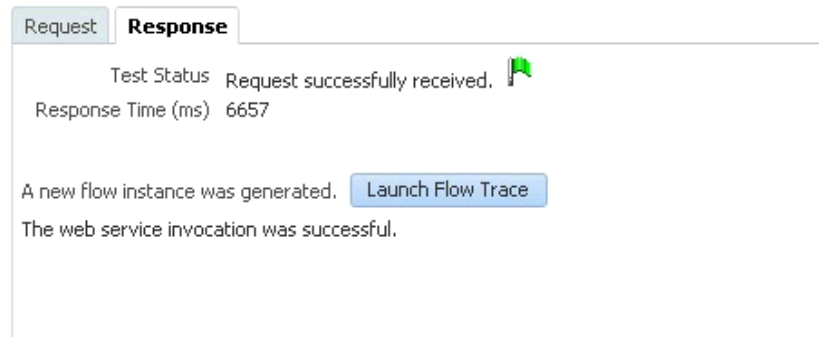
- Oracle Enterprise Manager Console にログオンします。
- 左ペインでドメイン、「SOA」フォルダの順に展開します。
- 使用可能なプロジェクト (「J2CA_Outbound」など) を選択します。
- 「テスト」をクリックします。
- 「リクエスト」タブをクリックします。

図 6-38 「リクエスト」タブ



- 図 6-38 に示すように、「値」フィールドに適切な入力値を入力し、「Web サービスのテスト」をクリックします。
図 6-39 に示すように、Oracle Enterprise Manager Console で起動に成功したことを示すレスポンスが、「レスポンス」タブで受信されます。

図 6-39 受信されたレスポンス



7. ファイル・システムの定義済出力ディレクトリに移動し、受信された XML レスポンス・ドキュメントを開きます。

XML レスポンス・ドキュメントには、値と生成された出力が含まれます。

6.5 トランスフォーメーションを使用したイベント統合のためのインバウンド BPM プロセスの設計 (J2CA 構成)

この項では、Oracle Application Adapter for J.D. Edwards OneWorld を J.D. Edwards OneWorld と統合してイベント・データを受信する方法について説明します。

このインバウンド・ユース・ケース・シナリオ用のサンプル・プロジェクトが、次に示す Application Adapter のインストール・フォルダに用意されています。

```
<ADAPTER_HOME>\etc\sample\JDEdwards_Samples.zip\JDEdwards_Samples\BPM\J2CA\Inbound_Project
```

アダプタのデザインタイム構成を完了するには、次のツールが必要です。

- Oracle Adapter アプリケーション・エクスプローラ (アプリケーション・エクスプローラ)
- Oracle JDeveloper BPM デザイナ (JDeveloper)

注意： この章に記載されている例では JDeveloper が使用されていません。

この項では、次のトピックについて説明します。

- [6.5.1 項「BPM 用の空のコンポジットの作成」](#)
- [6.5.2 項「BPM インバウンド・プロセスの定義」](#)

BPM プロセスを設計する前に、アプリケーション・エクスプローラを使用して、各 WSDL ファイルを生成する必要があります。詳細は、4-35 ページの [4.5.1 項「イベント統合のための WSDL の生成」](#) を参照してください。

6.5.1 BPM 用の空のコンポジットの作成

Oracle JDeveloper での新規アプリケーション・サーバー接続の構成方法の詳細は、4-2 ページの [4.3 項「新規のアプリケーション・サーバー接続の構成」](#) を参照してください。

6.5.2 BPM インバウンド・プロセスの定義

この項では、BPM インバウンド・プロセスの定義方法について説明します。次のトピックで構成されています。

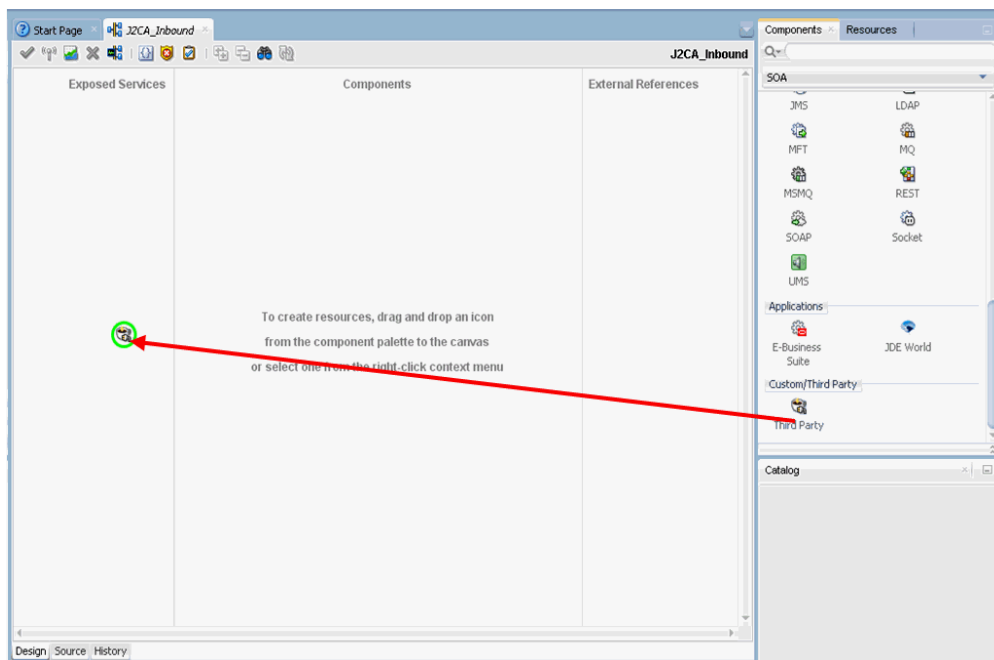
- 6.5.2.1 項「サード・パーティ・アダプタ・サービス・コンポーネントの構成」
- 6.5.2.2 項「インバウンド BPM プロセス・コンポーネントの構成」
- 6.5.2.3 項「書込み操作のファイル・アダプタの作成」
- 6.5.2.4 項「12c でのデプロイメントに関する既知の問題への対応」

6.5.2.1 サード・パーティ・アダプタ・サービス・コンポーネントの構成

サード・パーティ・アダプタ・サービス・コンポーネントを作成するには、次のステップを実行します。

1. 作成したプロジェクトをダブルクリックして、コンポーネントをロードします。
2. 図 6-40 に示すように、「カスタム/サード・パーティ」ペインから「サード・パーティ・アダプタ」コンポーネントを公開された参照ペインにドラッグ・アンド・ドロップします。

図 6-40 「サード・パーティ・アダプタ」コンポーネント



「サード・パーティ・アダプタ・サービスの作成」ダイアログが表示されます。

3. サード・パーティ・アダプタ・サービスの名前を入力します。
4. 「タイプ」リストで「サービス」が選択されていることを確認します (デフォルト)。
5. 「WSDL URL」フィールドの右の「既存の WSDL を検索します。」アイコンをクリックします。

「WSDL チューザ」ダイアログが表示されます。

6. 「ファイルシステム」を選択し、次のディレクトリまでブラウズしてインバウンド WSDL ファイルを選択します。

```
<ADAPTER_HOME>\wsdl
```

7. 「OK」をクリックします。
「ファイルのローカライズ」ダイアログが表示されます。
8. 「OK」をクリックします。
インバウンド WSDL ファイルおよび関連 receive_request XML スキーマ・ファイル (.xsd) が、作成済のプロジェクト・フォルダにインポートされます。
「サード・パーティ・アダプタ・サービスの作成」ダイアログに戻ります。
9. 「JCA ファイル」フィールドの右の「JCA ファイルを検索します。」アイコンをクリックします。
変換チューザ・ダイアログが表示されます。
10. 「ファイルシステム」を選択し、次のディレクトリまでブラウズして JCA プロパティ・ファイルを選択します。
`<ADAPTER_HOME>\wsdl.s`
11. 「OK」をクリックします。
「ファイルのコピー」メッセージが表示されます。
12. 「はい」をクリックします。
JCA プロパティ・ファイルのコピーがプロジェクト・フォルダ内に作成されます。
「サード・パーティ・アダプタ・サービスの作成」ダイアログに戻ります。
13. 「OK」をクリックします。
サード・パーティ・アダプタ・サービス・コンポーネント (matmas) が作成され、公開された参照ペインに表示されます。
これで、インバウンド BPM プロセス・コンポーネントを構成する準備ができました。
詳細は、4-11 ページの 4.4.3.1 項「サード・パーティ・アダプタ・サービス・コンポーネントの構成」を参照してください。

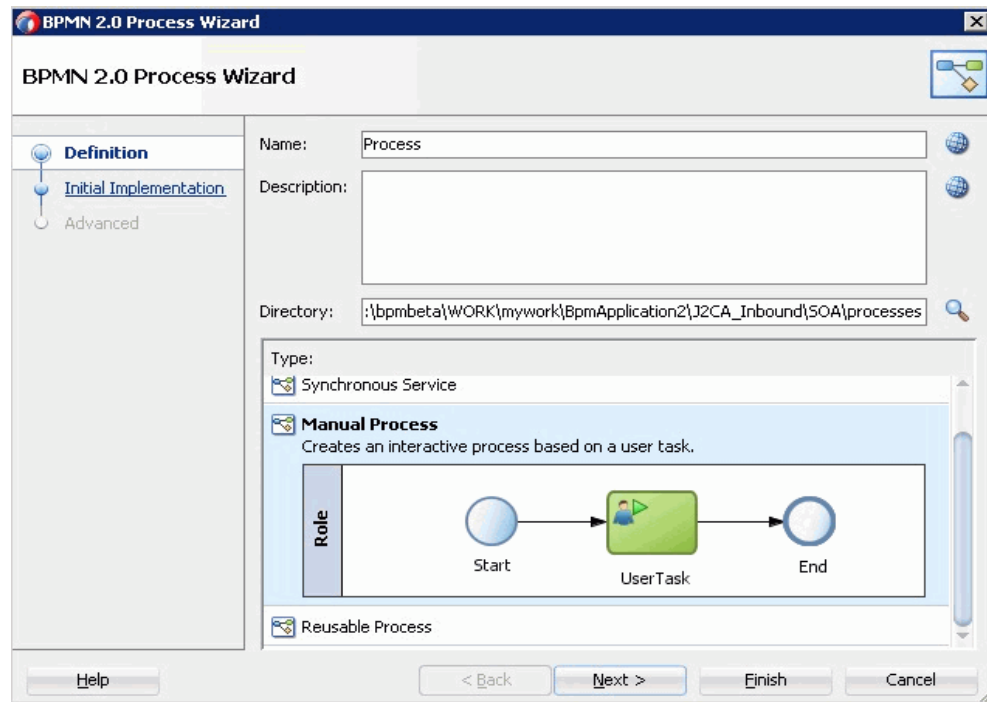
6.5.2.2 インバウンド BPM プロセス・コンポーネントの構成

この項では、インバウンド BPM プロセス・コンポーネントの構成方法について説明します。

インバウンド BPM プロセス・コンポーネントを構成するには、次のステップを実行します。

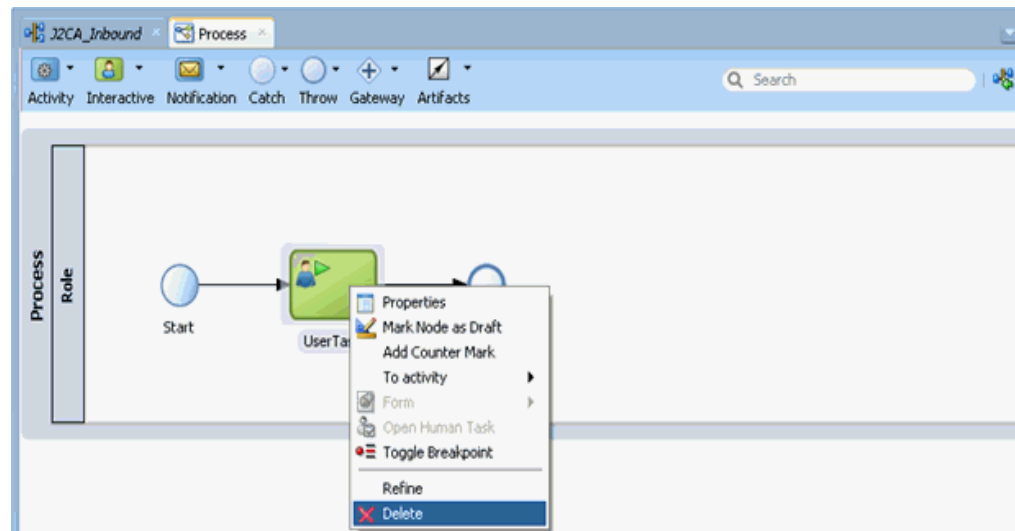
1. 「コンポーネント」ペインから「BPMN プロセス」コンポーネントを「コンポーネント」ペインにドラッグ・アンド・ドロップします。
図 6-41 に示すように、「BPMN プロセスの作成」ダイアログが表示されます。

図 6-41 「BPMN プロセスの作成」ダイアログ



2. 「タイプ」セクションで「手動プロセス」を選択します。
3. 「終了」をクリックします。
 6-42 に示すように、BPMN プロセスが表示されます。

図 6-42 BPMN プロセス



4. 「ユーザー・タスク」を右クリックし、メニューから「削除」を選択します。
5. 「Start」イベント・コンポーネントをダブルクリックします。
「プロパティ - Start」ダイアログが表示されます。
6. 「実装」タブをクリックします。

7. 「実装タイプ」リストから「メッセージ」を選択します。
8. 「メッセージ交換」の「タイプ」ドロップダウン・リストから「インタフェースの使用」を選択します。
9. 図 6-43 に示すように、「参照」フィールドの右の「参照」アイコンをクリックします。

図 6-43 「参照」アイコン

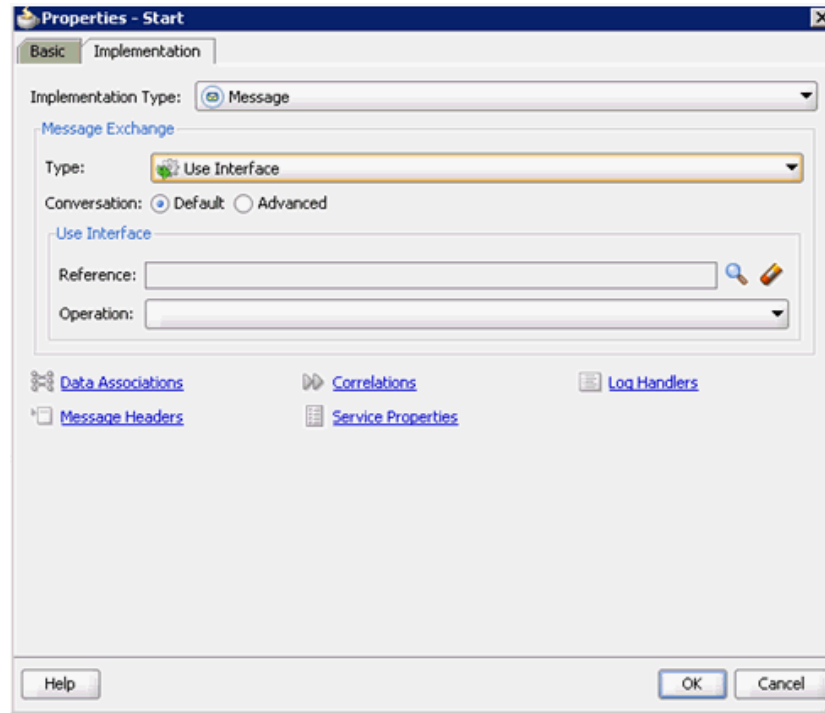
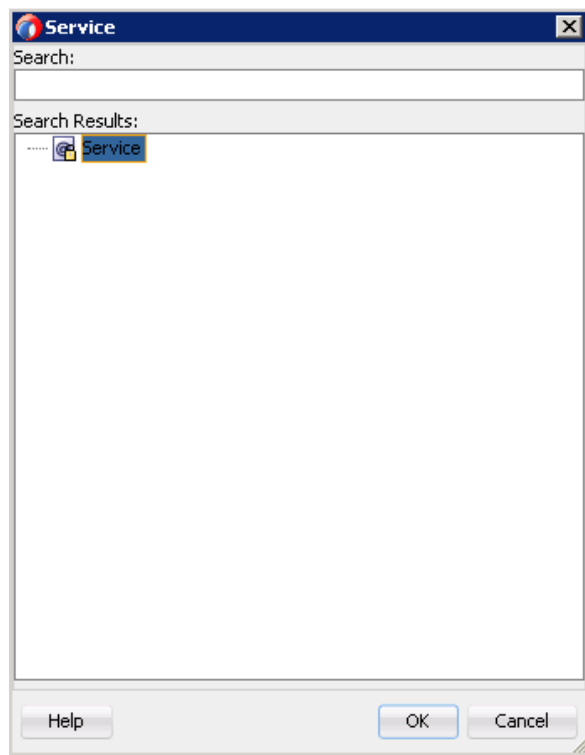


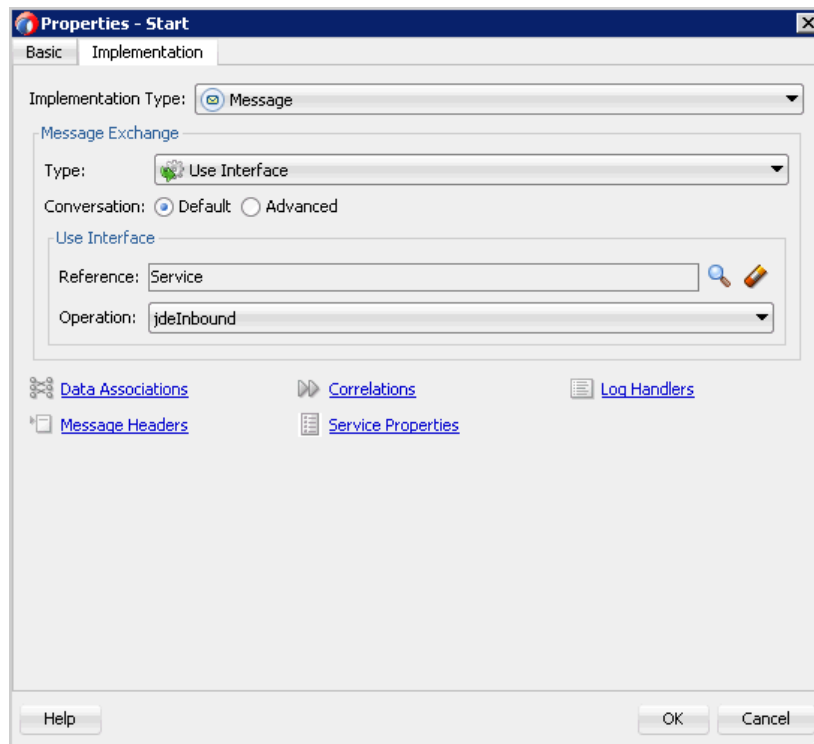
図 6-44 に示すように、「サービス」ダイアログが表示されます。

図 6-44 「サービス」 ダイアログ



10. 作成されたサード・パーティ・サービスを選択し、「OK」をクリックします。
図 6-45 に示すように、「プロパティ - Start」ダイアログに戻ります。

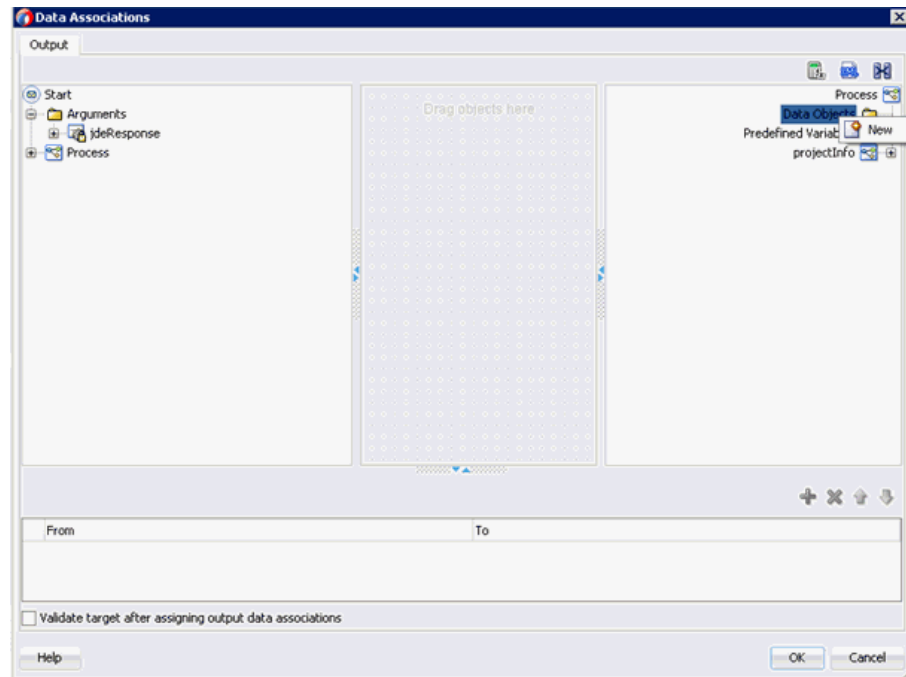
図 6-45 「プロパティ - Start」ダイアログ



11. 「データ・アソシエーション」アイコンをクリックします。

図 6-46 に示すように、「データ・アソシエーション」ダイアログが表示されます。

図 6-46 「データ・アソシエーション」ダイアログ



12. 右ペインで「データ・オブジェクト」ノードを右クリックし、「新規」を選択します。

「データ・オブジェクトの作成」ダイアログが表示されます。

13. 図 6-47 に示すように、「名前」フィールドに名前を入力し、「タイプ」フィールドのドロップダウン・ボタンをクリックして、リストから「参照」を選択します。

図 6-47 「データ・オブジェクトの作成」ダイアログ

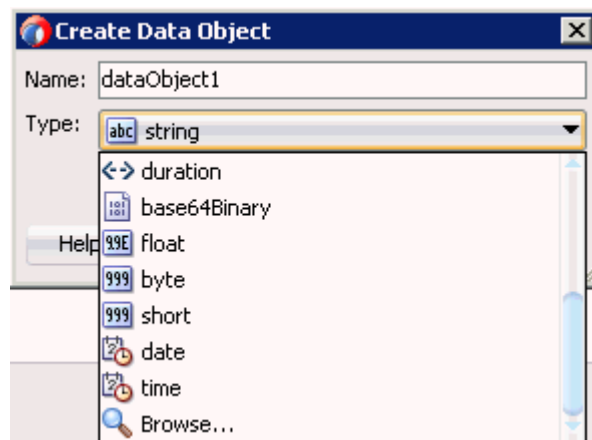
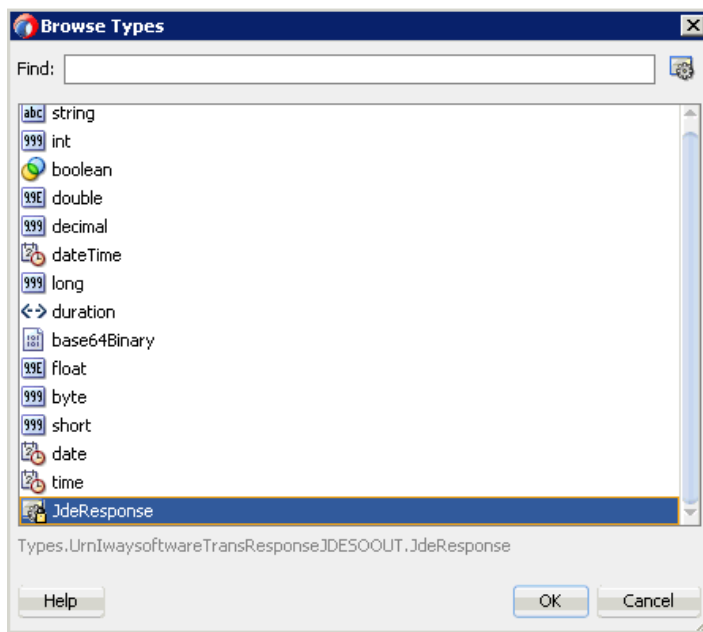


図 6-48 に示すように、「参照タイプ」ダイアログが表示されます。

図 6-48 「参照タイプ」 ダイアログ



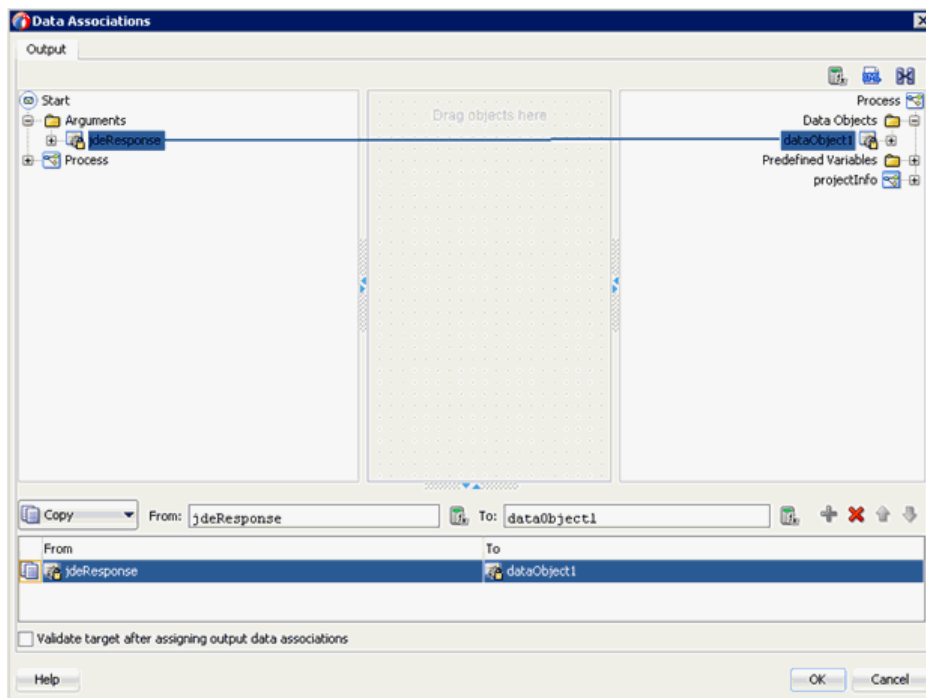
14. コンポーネントを選択し、「OK」をクリックします。

「データ・オブジェクトの作成」ダイアログに戻ります。

15. 「OK」をクリックします。

図 6-49 に示すように、作成されたデータ・オブジェクトが「データ・アソシエーション」ダイアログの「データ・オブジェクト」ノードの下に表示されます。

図 6-49 「データ・アソシエーション」 ダイアログ



16. 左ペインで「Start」ノードの下の「jdeResponse」を選択し、右ペインの「データ・オブジェクト」にドラッグします。
17. 「OK」をクリックします。
「プロパティ - Start」ダイアログに戻ります。
18. 「OK」をクリックします。
「プロセス」ワークスペース領域に戻ります。
19. 作成したプロジェクトをダブルクリックして、コンポーネントをロードします。
20. メニュー・バーで「すべて保存」アイコンをクリックし、構成された新規のインバウンド BPM プロセス・コンポーネントを保存します。
これで書込み操作作用のファイル・アダプタを作成する準備ができました。

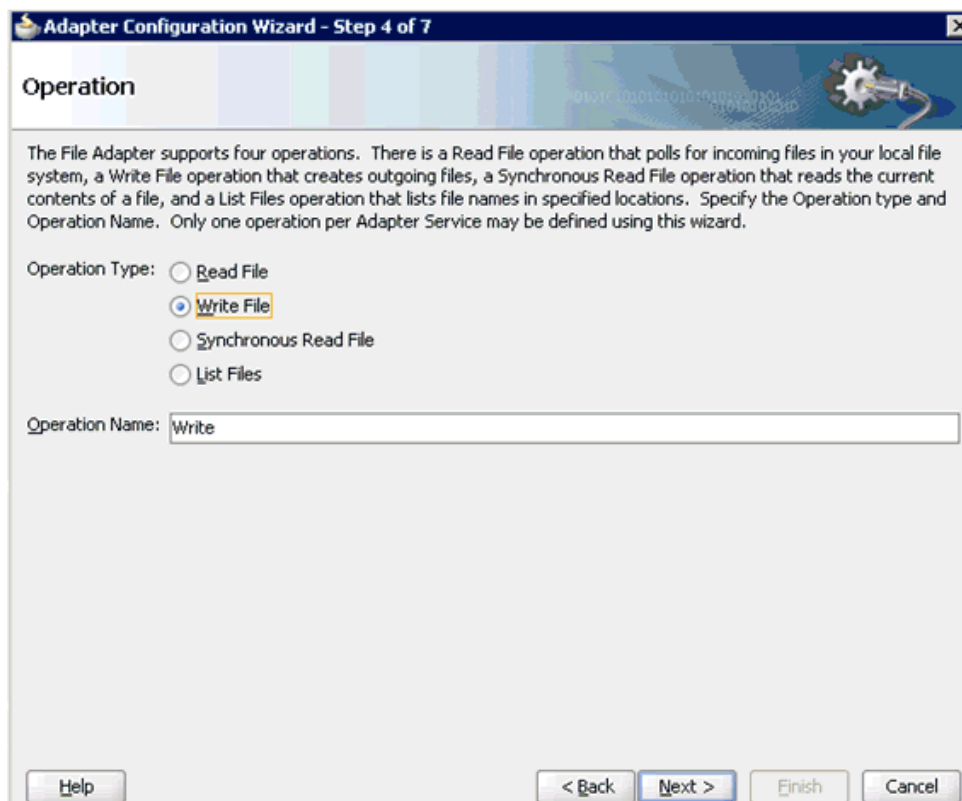
6.5.2.3 書込み操作作用のファイル・アダプタの作成

この項では、書込み操作作用のファイル・アダプタの作成方法について説明します。

書込み操作作用のファイル・アダプタを作成するには、次のステップを実行します。

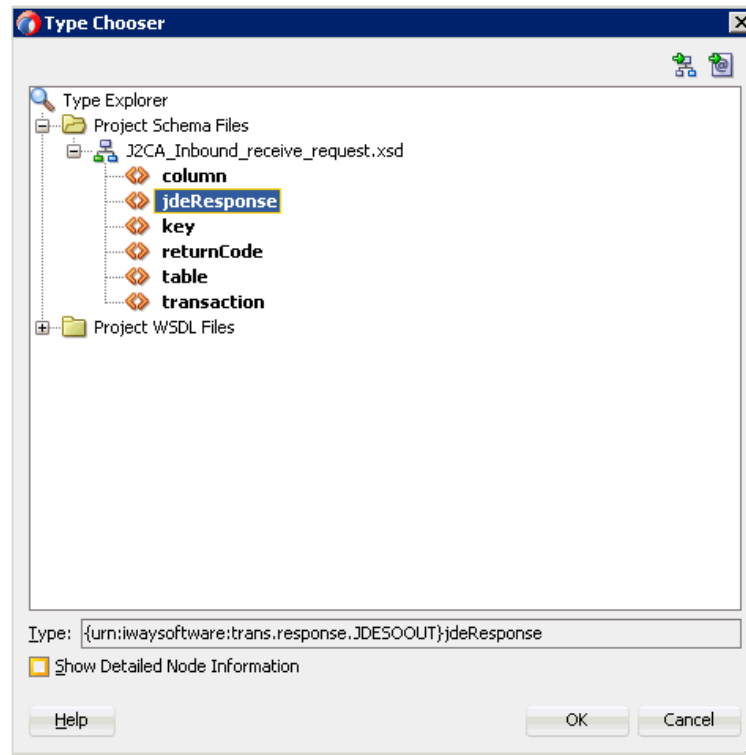
1. 「テクノロジー・アダプタ」ペインから「ファイル・アダプタ」コンポーネントを「外部参照」ペインにドラッグ・アンド・ドロップします。
アダプタ構成ウィザードが表示されます。
2. 「名前」フィールドで、新規のファイル・アダプタの名前を入力し、「次へ」をクリックします。
「アダプタ・インタフェース」ページが表示されます。
3. 「操作およびスキーマから定義 (後で指定)」オプションが選択されていることを確認します。
4. 「次へ」をクリックします。
「ファイル・サーバーの接続」ページが表示されます。
5. 「次へ」をクリックします。
図 6-50 に示すように、「操作」ページが表示されます。

図 6-50 「操作」 ページ



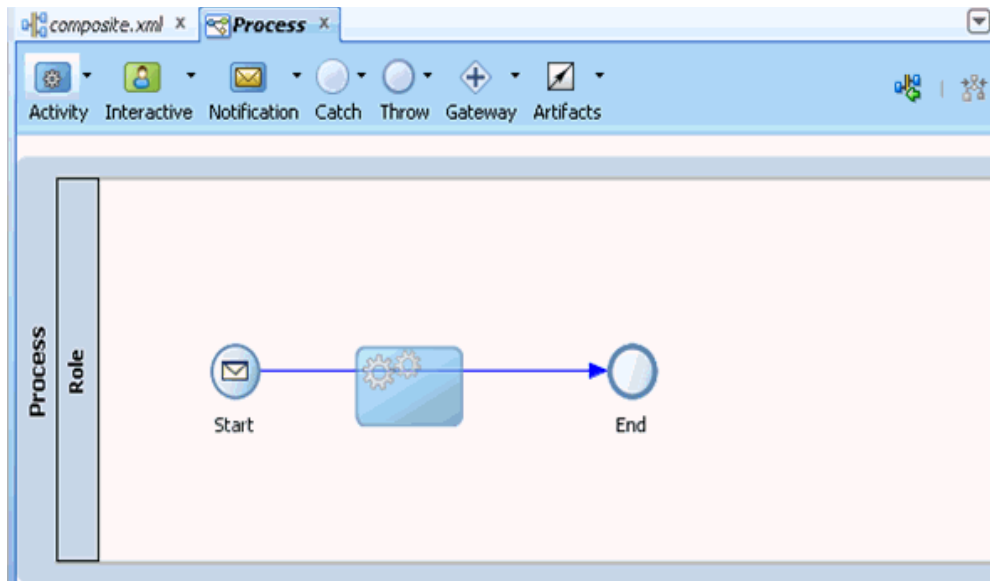
6. 「操作タイプ」オプションのリストから「**Write File**」を選択し、「操作名」(「Write」など)を指定します。
7. 「次へ」をクリックします。
「ファイル構成」ページが表示されます。
8. 出力ファイルの書き込み先となるファイル・システム上の場所を指定します。
9. ファイル・ネーミング規則 (po_%SEQ%.txt) フィールドで、出力ファイルの名前を指定します。
10. 「次へ」をクリックします。
「メッセージ」ページが表示されます。
11. 「URL」フィールドの右の「参照」をクリックします。
図 6-51 に示すように、「タイプ・チューザ」ダイアログが表示されます。

図 6-51 「タイプ・チューザ」 ダイアログ



12. 「プロジェクトのスキーマ・ファイル」、「J2CA_Inbound_receive_request.xsd」の順に展開します。
13. 使用可能なスキーマを選択します。
14. 「OK」をクリックします。
「メッセージ」ページに戻ります。
15. 「次へ」をクリックします。
「終了」ページが表示されます。
16. 「終了」をクリックします。
「外部参照」ペインでファイル・アダプタ・サービスが作成されます。
17. BPMN プロセス・コンポーネントをダブルクリックします。
BPMN プロセスが表示されます。
18. 「アクティビティ」アイコンをクリックし、「サービス」を選択します。
19. 図 6-52 に示すように、「Start」と「End」イベント・コンポーネント間の伝送路に「サービス」アイコンをドロップします。

図 6-52 「アクティビティ」アイコン

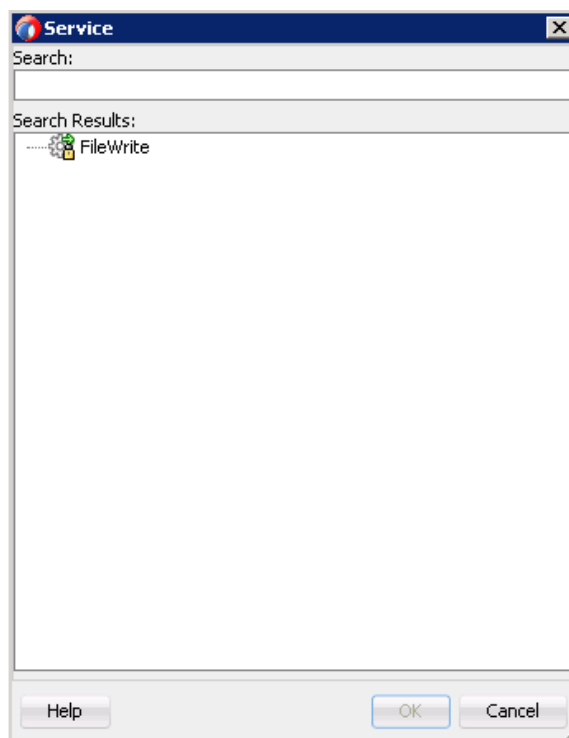


「プロパティ - ServiceTask」ダイアログが表示されます。

20. 「実装」タブをクリックします。
21. 「実装タイプ」リストから「サービス・タスク」を選択します。
22. 「メッセージ交換」の「タイプ」リストから「サービス・コール」を選択します。
23. 「サービス」フィールドの右の「参照」アイコンをクリックします。

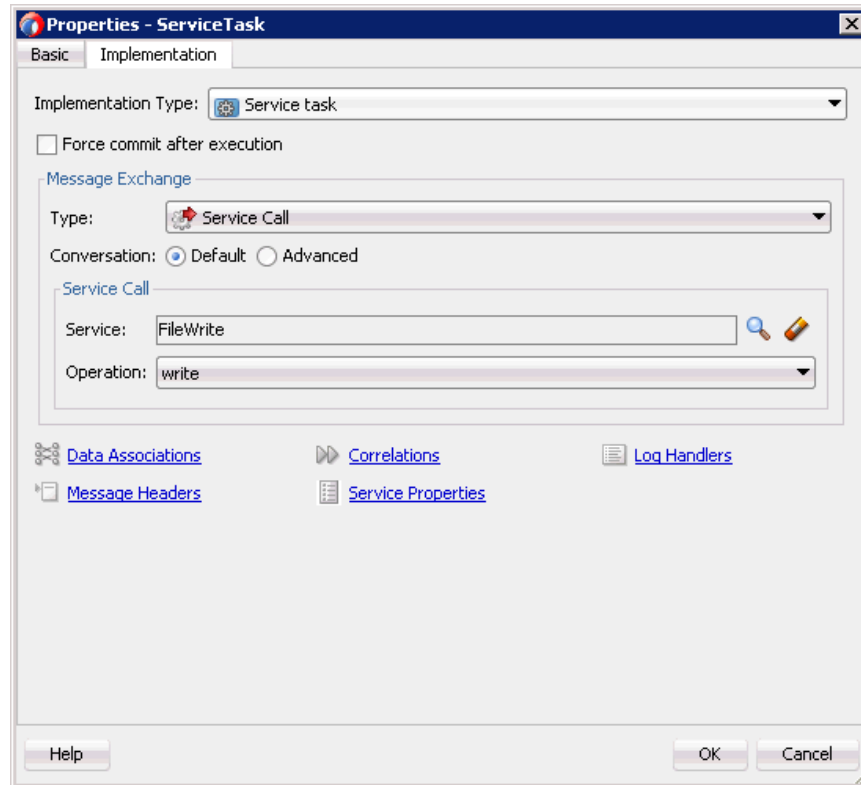
図 6-53 に示すように、「タイプ」ダイアログが表示されます。

図 6-53 「タイプ」ダイアログ



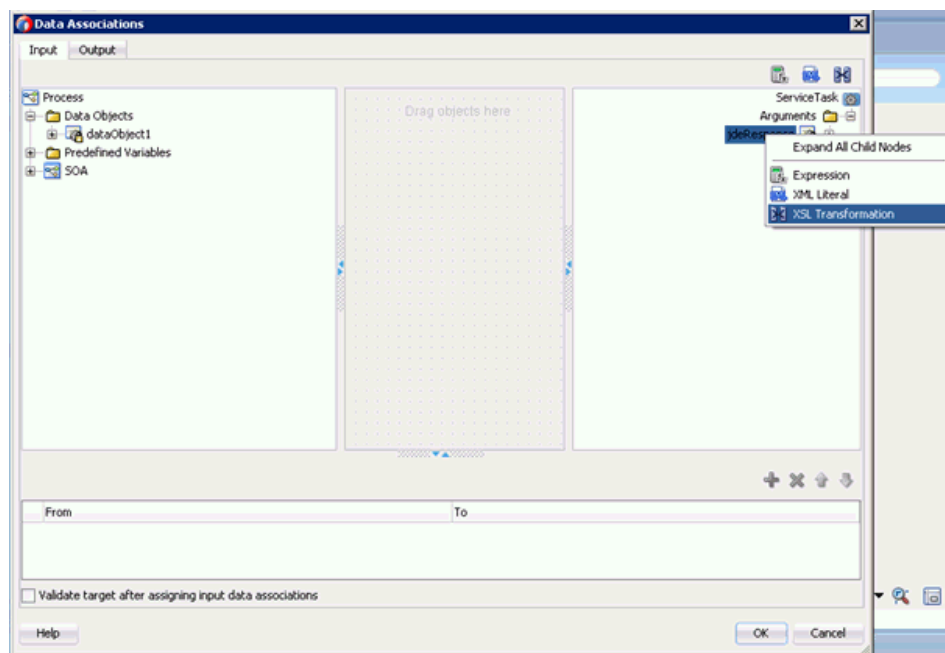
24. 作成された書き込み操作のサービスを選択し、「OK」をクリックします。
図 6-54 に示すように、「プロパティ - ServiceTask」ダイアログに戻ります。

図 6-54 「プロパティ - ServiceTask」ダイアログ



25. 「データ・アソシエーション」ハイパーリンクをクリックします。
「データ・アソシエーション」ダイアログが表示されます。
26. 図 6-55 に示すように、右ペインで「jdeResponse」引数を右クリックし、「XSL 変換」を選択します。

図 6-55 XSL 変換

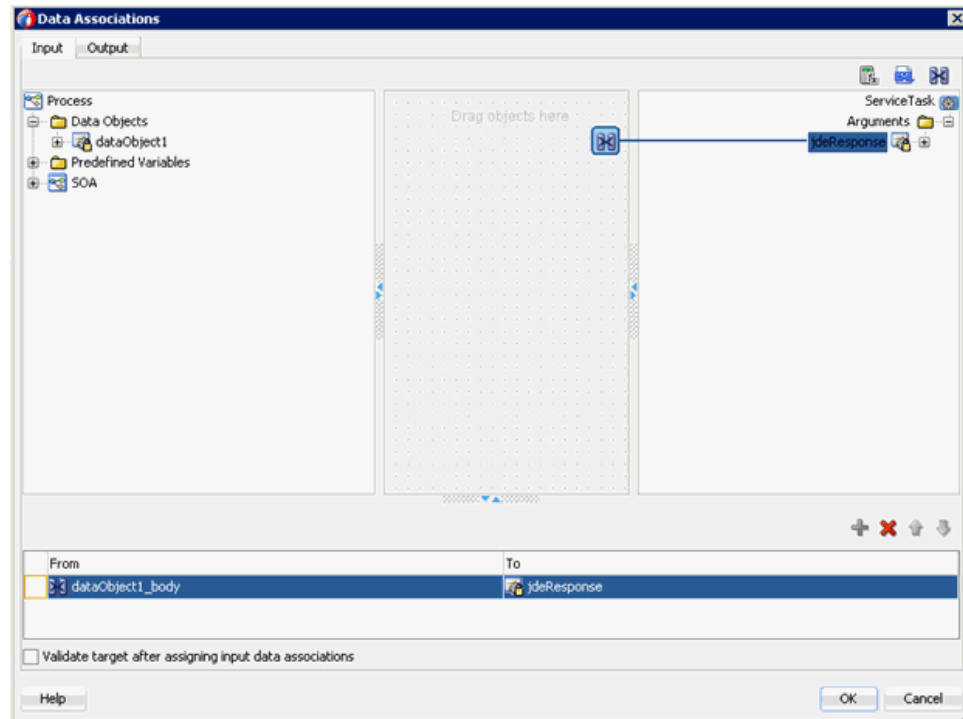


「トランスフォーメーションの作成」ダイアログが表示されます。

27. 「ソース」領域で作成されたデータ・オブジェクトを選択し、右矢印アイコンをクリックします。作成されたデータ・オブジェクトが「選択された要素」領域に追加されます。
28. 「OK」をクリックします。

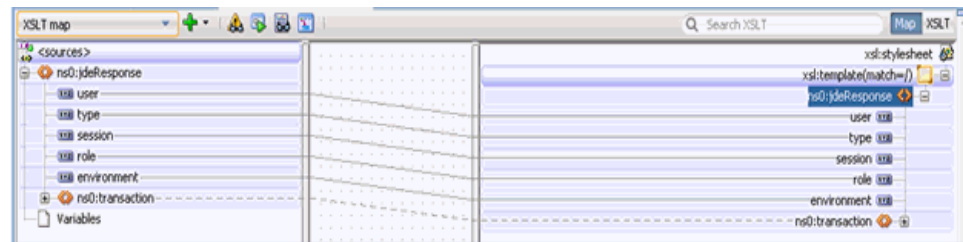
図 6-56 に示すように、「データ・アソシエーション」ダイアログに戻ります。

図 6-56 「データ・アソシエーション」 ダイアログ



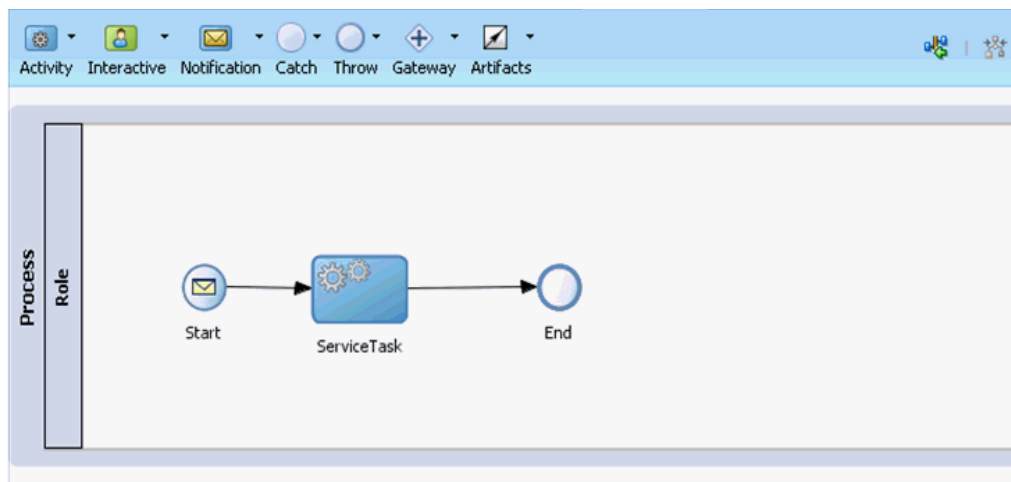
29. 「OK」をクリックします。
「プロパティ - ServiceTask」ダイアログに戻ります。
30. 「OK」をクリックします。
「dataobject1_body.xml」タブが表示されます。
31. ソース要素とターゲット要素を自動マップします。
「自動マップ・プリファレンス」ダイアログが表示されます。
32. デフォルト値を受け入れ、「OK」をクリックします。
図 6-57 に示すように、トランスフォーメーションが完成します。

図 6-57 完成したトランスフォーメーション



33. トランスフォーメーションを保存します。
34. 図 6-58 に示すように、「プロセス」ワークスペース領域に戻ります。

図 6-58 「プロセス」ワークスペース領域



「Start」イベント・コンポーネント間に「ServiceTask」が作成されます。

35. メニュー・バーで「すべて保存」アイコンをクリックし、構成された新規のインバウンド BPM プロセス・コンポーネントを保存します。

6.5.2.4 12c でのデプロイメントに関する既知の問題への対応

12c でのデプロイメントに関する既知の問題への対応方法の詳細は、4-26 ページの 4.4.3.3 項「12c でのデプロイメントに関する既知の問題への対応」を参照してください。

これで、インバウンド BPM プロセスをデプロイする準備ができました。4-47 ページの 4.5.4 項「BPEL インバウンド・プロセスのデプロイ」の記載と同じ手順に従います。

J.D. Edwards OneWorld でのイベントのトリガー方法の詳細は、4-48 ページの 4.5.5 項「J.D. Edwards OneWorld でのイベントのトリガー」を参照してください。

6.6 トランスフォーメーションを使用したサービス統合のためのアウトバウンド BPM プロセスの設計 (BSE 構成)

この項では、Oracle JDeveloper で BPM プロジェクトを使用して J.D. Edwards OneWorld システム用に BPM アウトバウンド・プロセスを構成する方法について説明します。

このアウトバウンド・ユース・ケース・シナリオ用のサンプル・プロジェクトが、次に示す Application Adapter のインストール・フォルダに用意されています。

```
<ADAPTER_HOME>\etc\sample\JDEdwards_Samples.zip\JDEdwards_Samples\BPM\BSE\Outbound_Project
```

アウトバウンドのデザインタイム構成を完了するには、次のツールが必要です。

- Oracle Adapter アプリケーション・エクスプローラ (アプリケーション・エクスプローラ)
- Oracle JDeveloper BPM デザイナ (JDeveloper)

この項では、次のトピックについて説明します。

- 6.6.1 項「BPM 用の空のコンポジットの作成」

- 6.6.2 項「BPM アウトバウンド・プロセスの定義」

前提条件

BPM アウトバウンド・プロセスを設計する前に、アプリケーション・エクスプローラを使用して各 WSDL ファイルを生成する必要があります。詳細は、4-53 ページの 4.6.1 項「Web サービスを使用したリクエスト / レスポンス型サービス用の WSDL ファイルの生成」を参照してください。

6.6.1 BPM 用の空のコンポジットの作成

SOA 用に空のコンポジットを作成するには、次のステップを実行します。

1. 新規の BPM アプリケーションを作成します。
2. BPM アプリケーションの名前を入力し、「次へ」をクリックします。
3. 「プロジェクト名」フィールドに名前を入力し、「次へ」をクリックします。
4. 「コンポジット・テンプレート」リストで「空のコンポジット」を選択し、「終了」をクリックします。

詳細は、4-40 ページの 4.5.2 項「SOA 用の空のコンポジットの作成」を参照してください。

6.6.2 BPM アウトバウンド・プロセスの定義

この項では、BPM アウトバウンド・プロセスの定義方法について説明します。次のトピックが含まれています。

- 6.6.2.1 項「Web サービス・コンポーネントの構成」
- 6.6.2.2 項「BPM プロセス・コンポーネントの構成」
- 6.6.2.3 項「書込み操作のファイル・アダプタの作成」

6.6.2.1 Web サービス・コンポーネントの構成

Web サービス・コンポーネントを構成するには、次のステップを実行します。


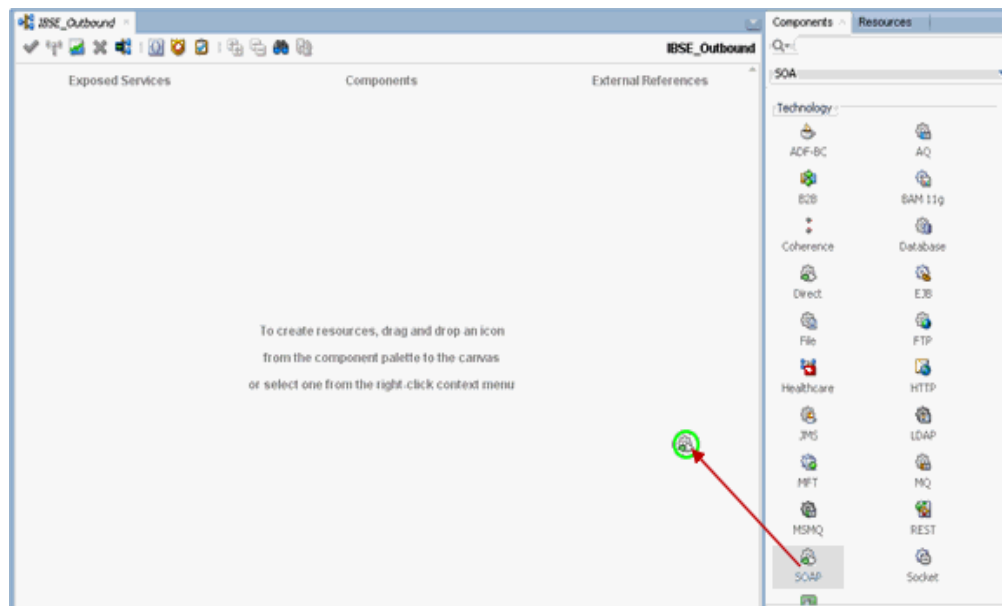
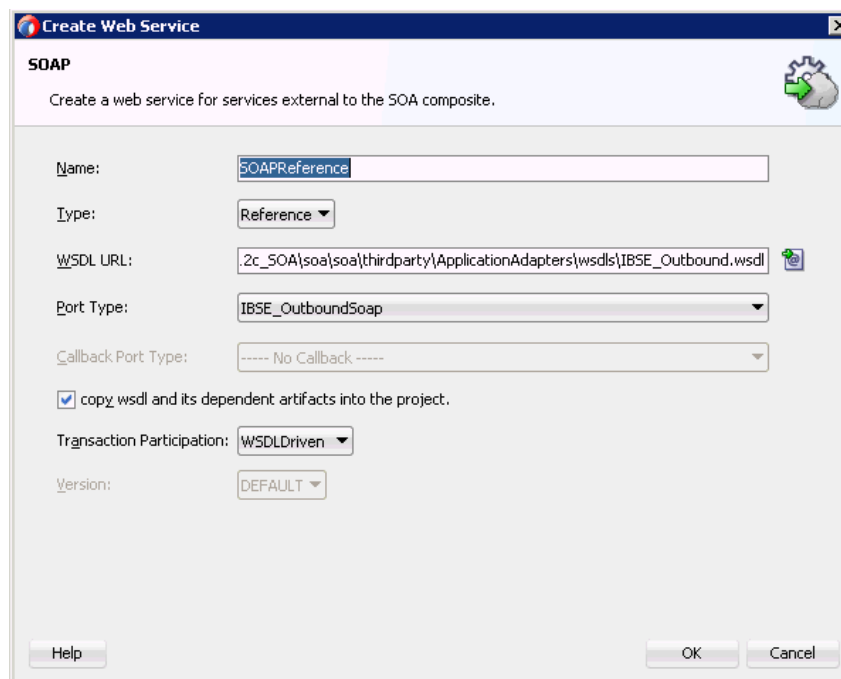
1. 作成したプロジェクトをダブルクリックして、コンポーネントをロードします。
2.  6-59 に示すように、「テクノロジー・アダプタ」ペインから「Web サービス」ノードを「外部参照」ペインにドラッグ・アンド・ドロップします。

図 6-59 「Web サービス」 ノード



3. Web サービスの適切な名前を入力し、「WSDL URL」フィールドの右の「既存の WSDL を検索します。」アイコンをクリックします。
4. 表示された「WSDL チューザ」ウィンドウで、アプリケーション・エクスプローラから WSDL をエクスポートする場所にナビゲートし、WSDL を選択します。
5. 「OK」をクリックします。
6. 図 6-60 に示すように、「Web サービス」 ペインで、「OK」をクリックします。

図 6-60 「Web サービス」 ペイン



7. 表示された「ファイルのローカライズ」ウィンドウで、「OK」をクリックします。これにより、WSDL ファイルがプロジェクト・フォルダにインポートされます。

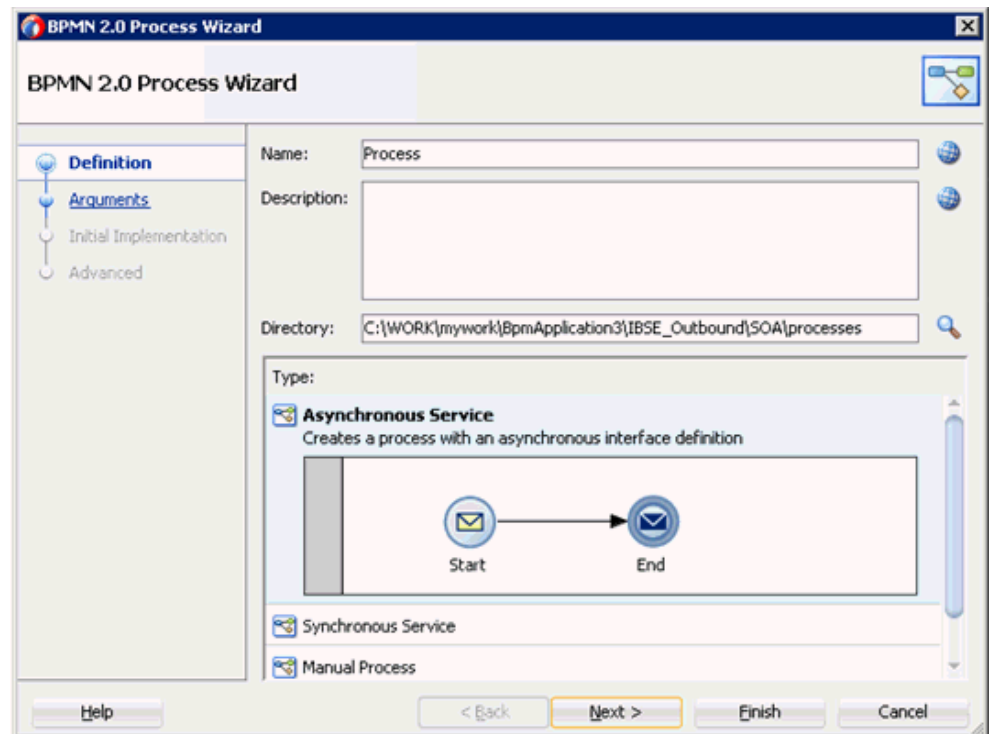
6.6.2.2 BPM プロセス・コンポーネントの構成

この項では、アウトバウンド BPM プロセス・コンポーネントの構成方法について説明します。

BPM コンポーネントを構成するには、次のステップを実行します。

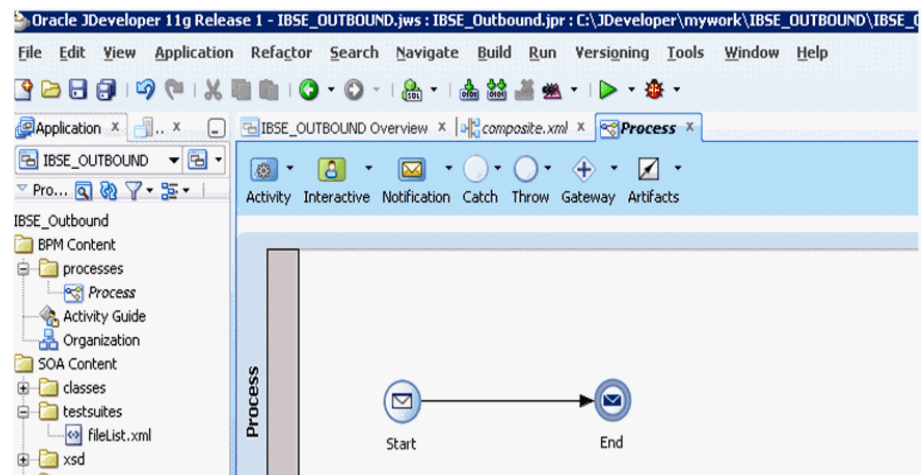
1. 「コンポーネント」ペインから「BPMN プロセス」コンポーネントを「コンポーネント」ペインにドラッグ・アンド・ドロップします。
2. 図 6-61 に示すように、「タイプ」領域で選択されているデフォルトのオプション（「非同期サービス」）を受け入れ、「終了」をクリックします。

図 6-61 「タイプ」領域



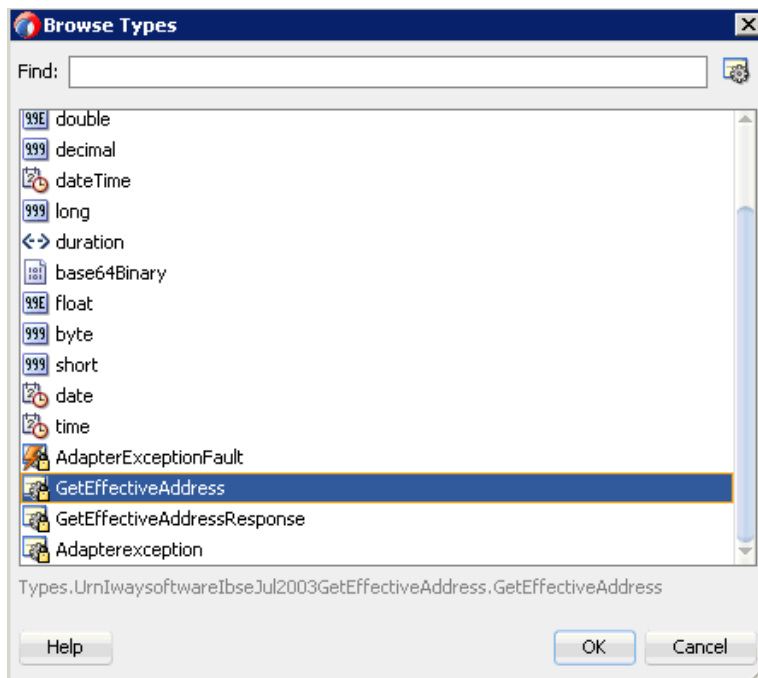
3. 図 6-62 に示すように、「Start」イベント・コンポーネントをダブルクリックします。

図 6-62 「Start」イベント・コンポーネント



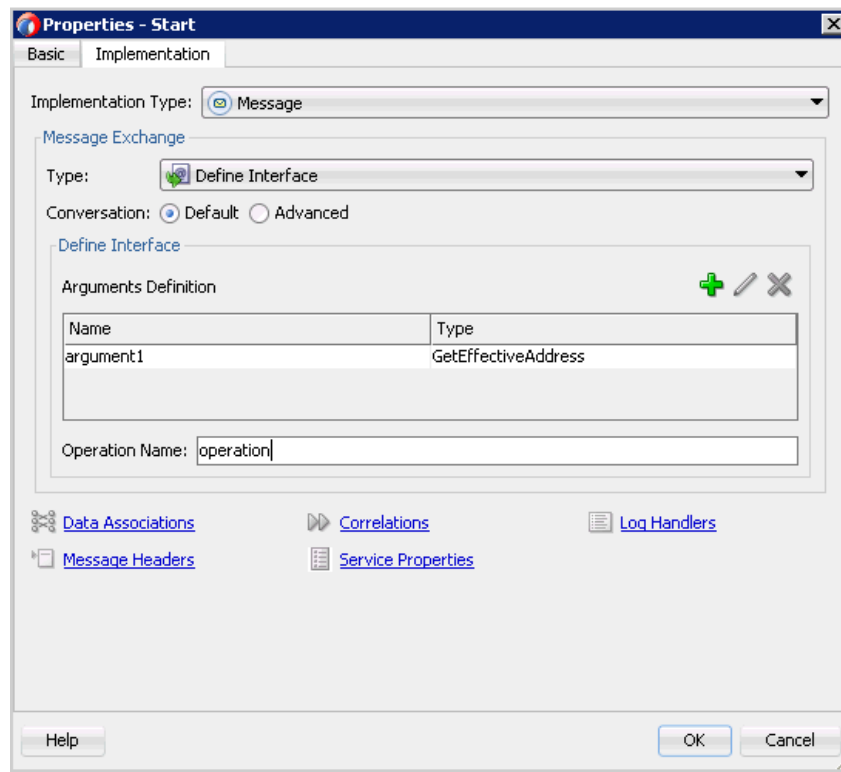
- 表示された「プロパティ - Start」ウィンドウで、「実装」タブをクリックします。
- 「引数の定義」フィールドの右の「プラス」(+) アイコンをクリックします。
「引数の編集」ウィンドウが表示されます。
- 「名前」フィールドに名前を入力し、「タイプ」ドロップダウン・リストをクリックして、「参照」を選択します。
- 図 6-63 に示すように、「Request」コンポーネント (「GetEffectiveAddress」など) を選択し、「OK」をクリックします。

図 6-63 「Request」コンポーネント



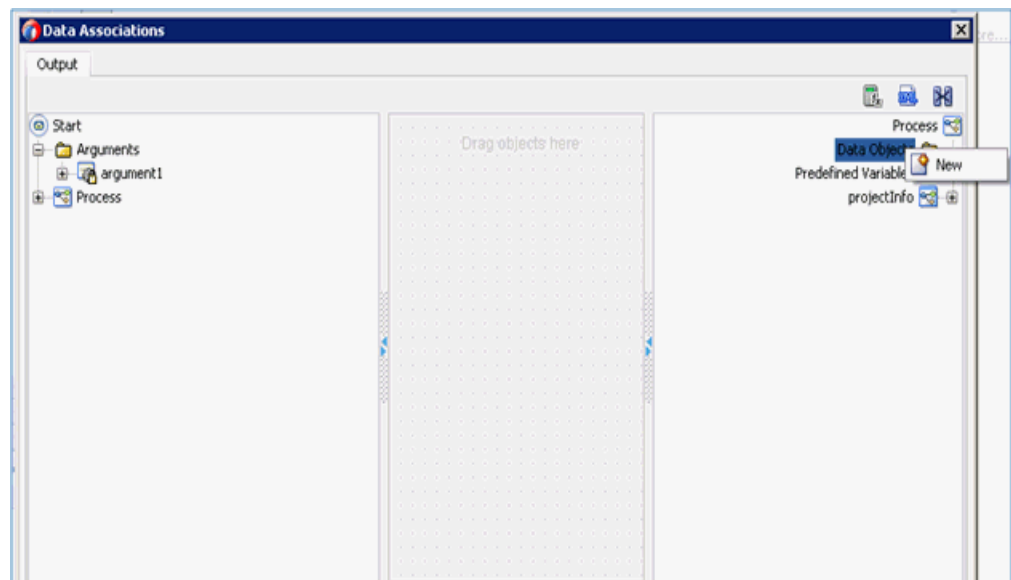
- 表示された「引数の編集」ウィンドウで、「OK」をクリックします。
「プロパティ - Start」ウィンドウが表示されます。
- 「操作名」フィールドで、デフォルト・エントリを「start」から「operation」に変更します。
- 図 6-64 に示すように、「データ・アソシエーション」ハイパーリンクをクリックします。

図 6-64 「プロパティ - Start」 ウィンドウ



11. 図 6-65 に示すように、右ペインで「プロセス」の下の「データ・オブジェクト」ノードを右クリックし、「新規」を選択します。

図 6-65 「データ・オブジェクト」ノード

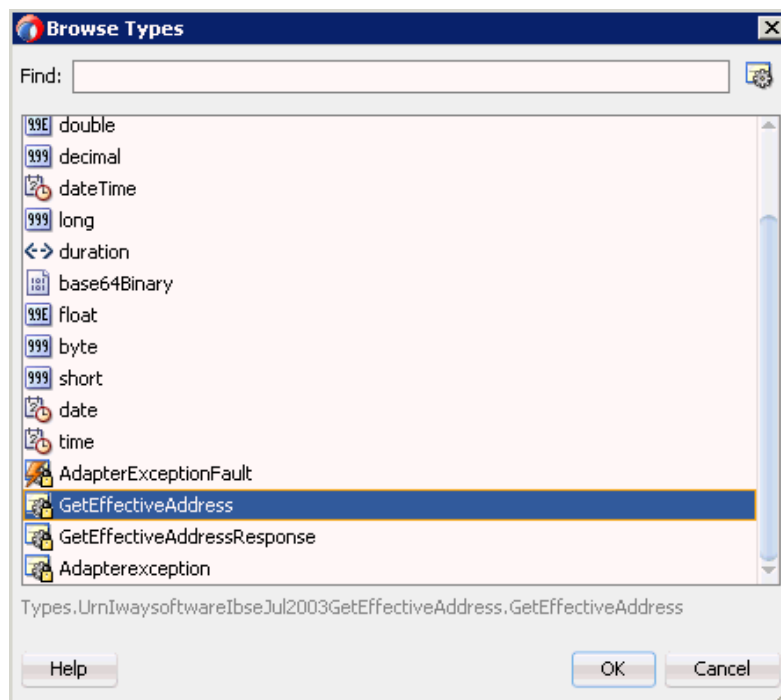


「データ・オブジェクトの作成」ウィンドウが表示されます。

12. 「名前」フィールドに名前を入力し、「タイプ」ドロップダウン・リストをクリックして、「参照」を選択します。

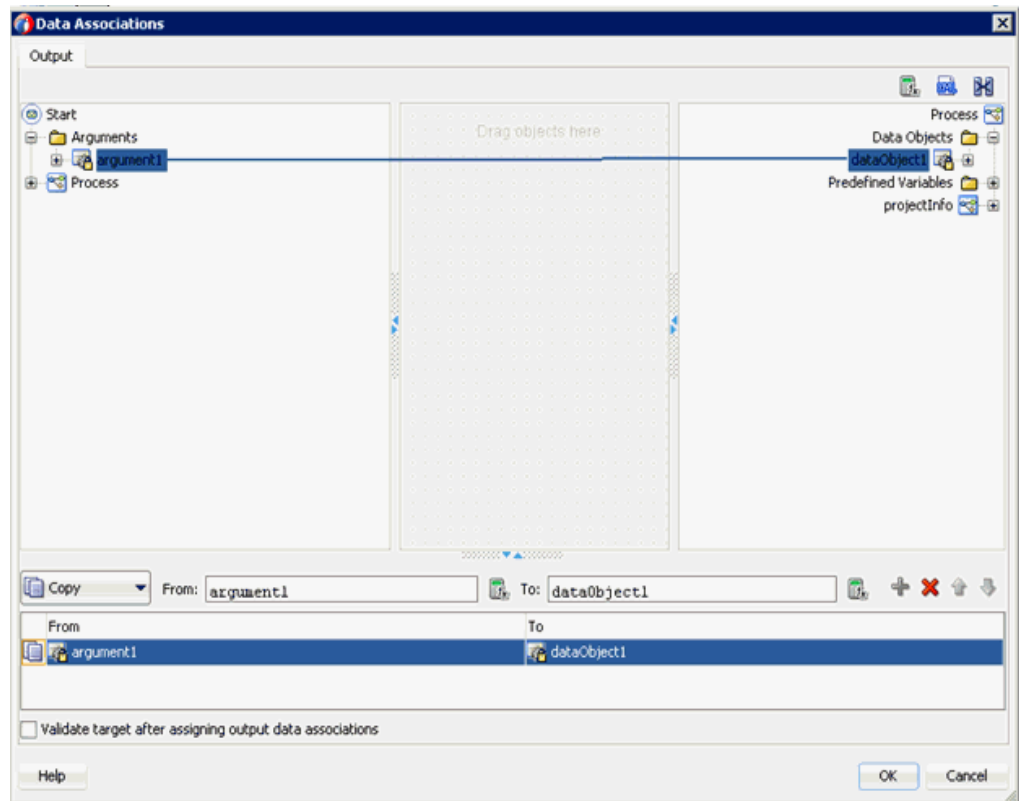
13. 図 6-66 に示すように、「Request」コンポーネント (「GetEffectiveAddress」など) を選択し、「OK」をクリックします。

図 6-66 「Request」コンポーネント



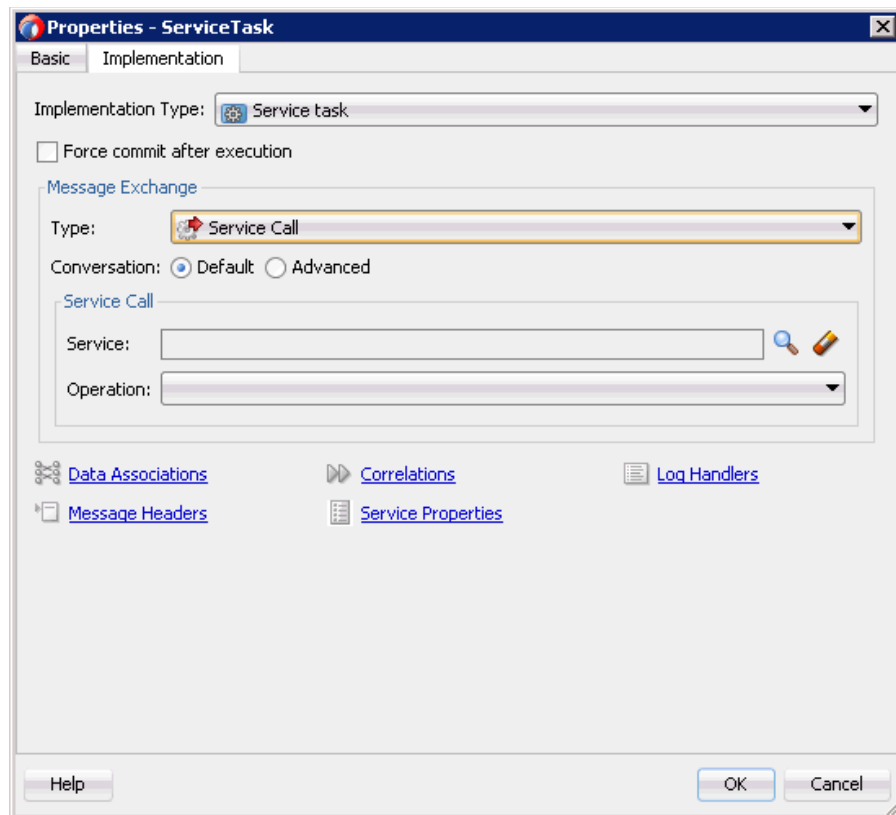
14. 「データ・オブジェクトの作成」ウィンドウで、「OK」をクリックします。
「データ・アソシエーション」ウィンドウが表示されます。
15. 左ペインで「引数」ノードの下の「argument1」を選択し、右ペインの「データ・オブジェクト」ノードの下の「dataObject1」にドラッグして接続します。
16. 図 6-67 に示すように、「OK」をクリックします。

図 6-67 データ・アソシエーション



17. 表示された「プロパティ - Start」ウィンドウで、「OK」をクリックします。
「プロセス」ワークスペース領域に戻ります。
18. 「アクティビティ」ドロップダウン・メニューをクリックし、「サービス」を選択します。
19. 「Start」と「End」イベント・コンポーネント間の伝送路に「サービス」アイコンをドロップします。
20. 表示された「プロパティ - ServiceTask」ウィンドウで、「実装」タブをクリックします。
21. 「メッセージ交換」の「タイプ」リストから「サービス・コール」を選択します。
22. 図 6-68 に示すように、「サービス」フィールドの右の「参照」アイコンをクリックします。

図 6-68 「参照」アイコン



「サービス」ウィンドウが表示されます。


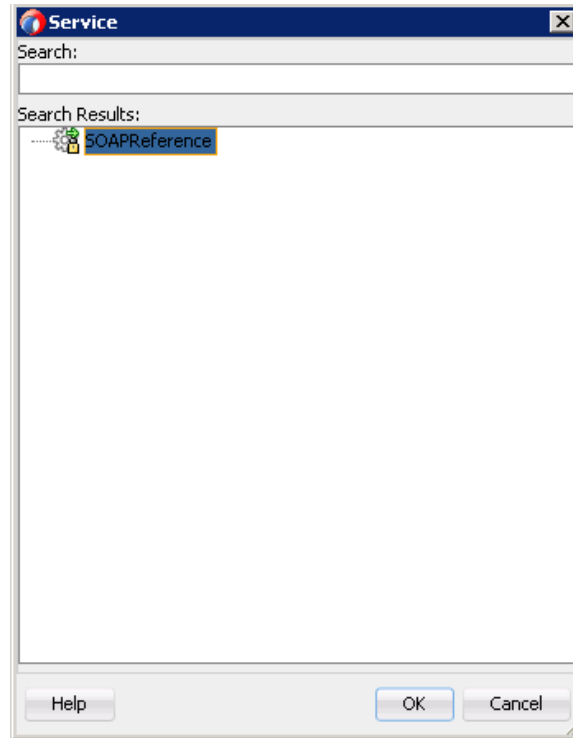
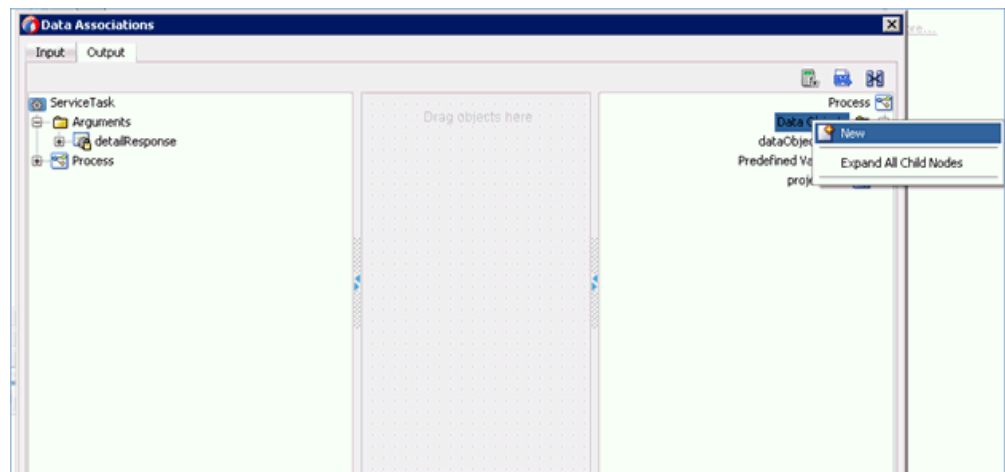
23.  6-69 に示すように、作成された Web サービスを選択し、「OK」をクリックします。

図 6-69 作成された Web サービス



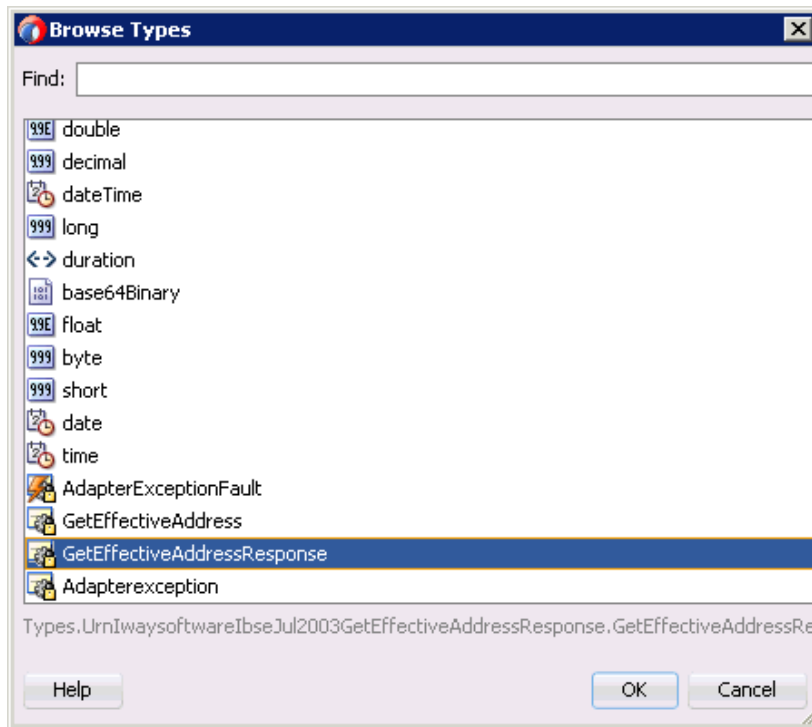
24. 表示された「プロパティ - ServiceTask」ウィンドウで、「データ・アソシエーション」ハイパーリンクをクリックします。
「データ・アソシエーション」ウィンドウが表示されます。
25. 図 6-70 に示すように、「出力」タブの右ペインで「データ・オブジェクト」ノードを右クリックし、「新規」を選択して、レスポンス・データ・オブジェクトを作成します。

図 6-70 「データ・オブジェクト」ノード



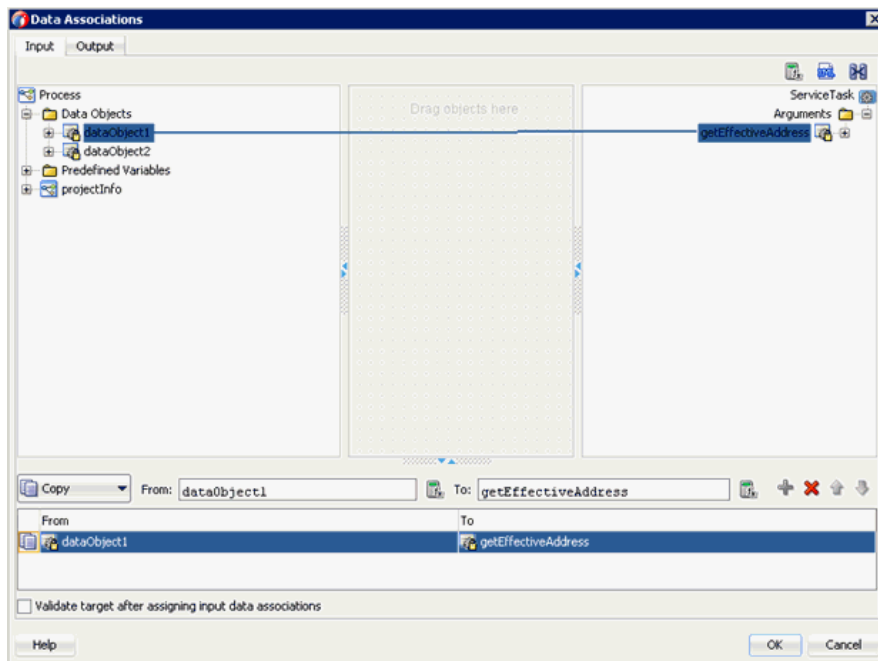
- 「データ・オブジェクトの作成」ウィンドウが表示されます。
26. 「名前」フィールドに名前を入力し、「タイプ」ドロップダウン・リストをクリックして、「参照」を選択します。
 27. 図 6-71 に示すように、「Response」コンポーネント (「GetEffectiveAddressResponse」など) を選択し、「OK」をクリックします。

図 6-71 「Response」コンポーネント



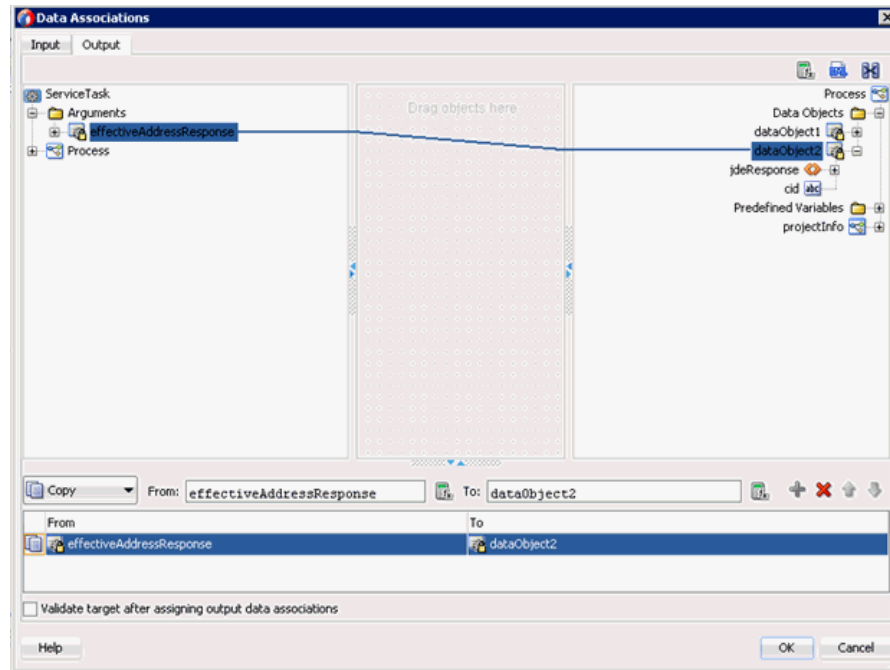
28. 「データ・オブジェクトの作成」ウィンドウで、「OK」をクリックします。
「データ・アソシエーション」ウィンドウが表示されます。
29. 図 6-72 に示すように、「入力」タブの左ペインで「データ・オブジェクト」ノードの下の「dataObject1」を選択し、右ペインの「引数」ノードの下の「getEffectiveAddress」ノードにドラッグして接続します。

図 6-72 データ・アソシエーション



30. 「出力」タブをクリックし、左ペインの「引数」ノードの下の「GetEffectiveAddressResponse」を選択し、「データ・オブジェクト」ノードの下の「dataObject2」にドラッグして接続します。
31. 図 6-73 に示すように、「OK」をクリックします。

図 6-73 「出力」タブ



32. 表示された「プロパティ - ServiceTask」ウィンドウで、「OK」をクリックします。
33. メニュー・バーで「すべて保存」アイコンをクリックし、構成された新規のアウトバウンド BPM プロセス・コンポーネントを保存します。
34. 左ペインの「composite.xml」ノードをダブルクリックします。

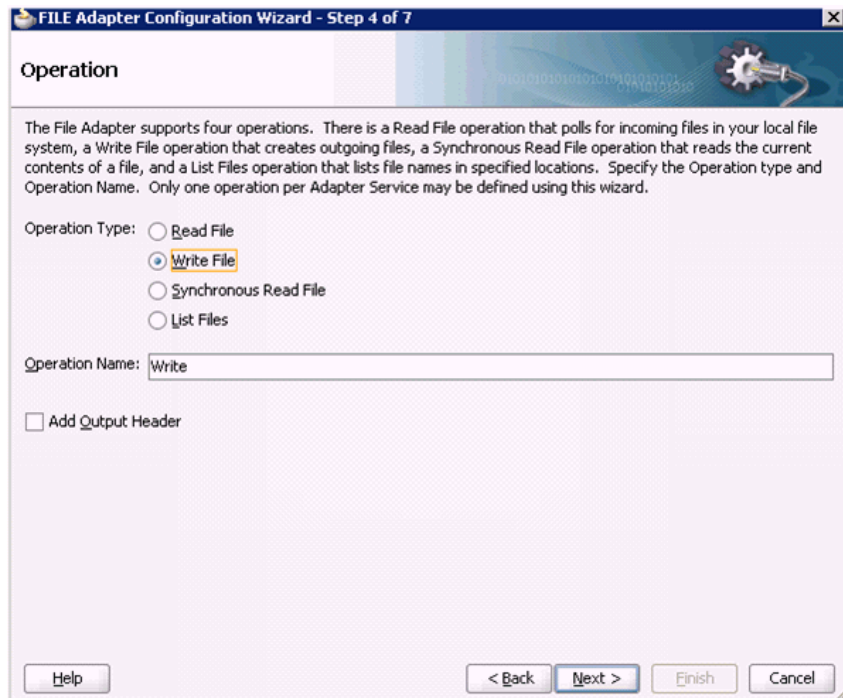
6.6.2.3 書込み操作のファイル・アダプタの作成

この項では、書込み操作のファイル・アダプタの作成方法について説明します。

書込み操作のファイル・アダプタを作成するには、次のステップを実行します。

1. 「テクノロジー・アダプタ」ペインから「ファイル・アダプタ」コンポーネントを「外部参照」ペインにドラッグ・アンド・ドロップし、ファイル・アダプタの名前を指定します。
2. 表示された「アダプタ・インタフェース」ペインで、「操作およびスキーマから定義(後で指定)」オプションが選択されていることを確認し、「次へ」をクリックします。
3. 「次へ」をクリックします。
4. 図 6-74 に示すように、表示された「操作」ペインで、「操作タイプ」オプションのリストから「Write File」を選択し、「次へ」をクリックします。

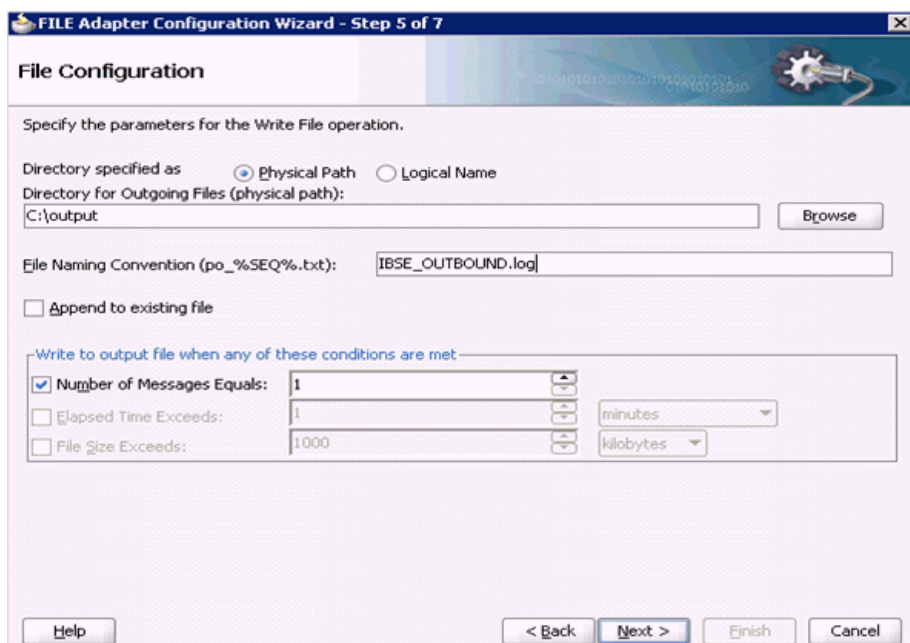
図 6-74 「操作」 ペイン



「ファイル構成」 ペインが表示されます。

5. 「発信ファイルのディレクトリ (物理パス)」フィールドで、出力ファイルの書き込み先となるファイル・システム上の場所を指定します。
6. ファイル・ネーミング規則 (po_%SEQ%.txt) フィールドで、出力ファイルの名前を指定します。
7. 図 6-75 に示すように、「次へ」をクリックします。

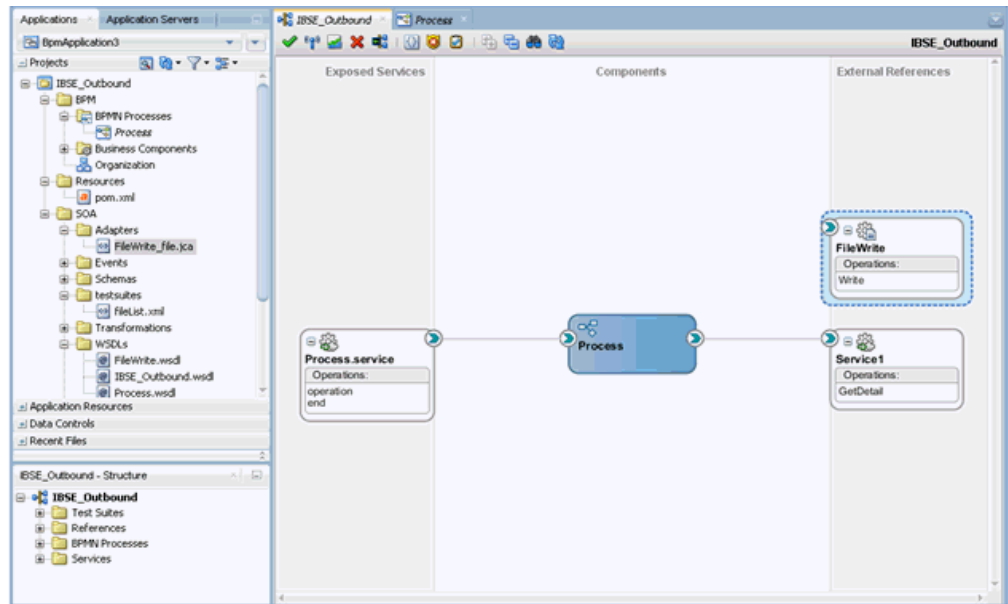
図 6-75 「ファイル構成」 ペイン



「メッセージ」ペインが表示されます。

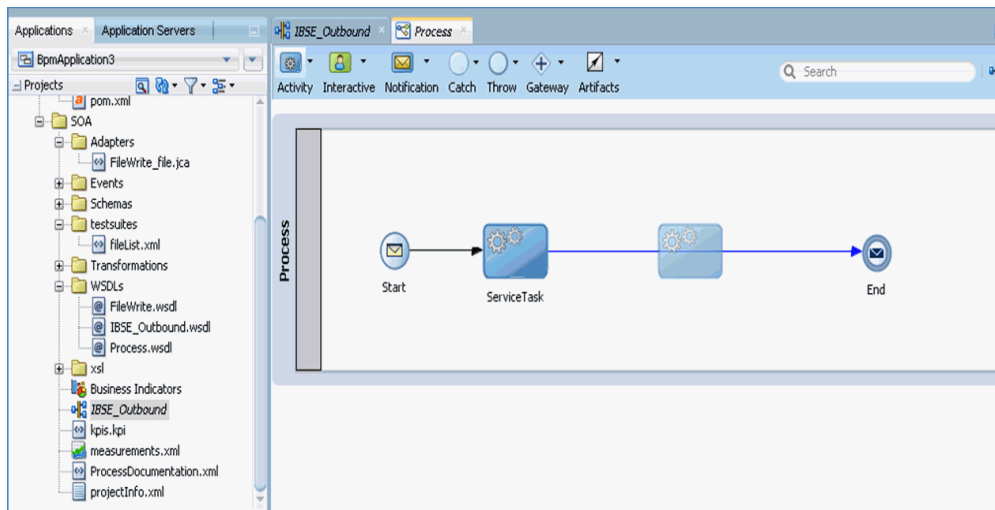
8. 「URL」フィールドの右の「参照」をクリックします。
9. 表示された「タイプ・チューザ」ウィンドウで、「プロジェクトの WSDL ファイル」、「IBSE_Outbound.wsdl」、「インライン・スキーマ」の順に展開し、「GetEffectiveAddressResponse」を選択します。
10. 「OK」をクリックします。
11. 「メッセージ」ペインで、「次へ」をクリックします。
12. 表示された「終了」ペインで、「終了」をクリックします。
13. 図 6-76 に示すように、BPMN プロセス・コンポーネントをダブルクリックします。

図 6-76 「Composite.xml」タブ



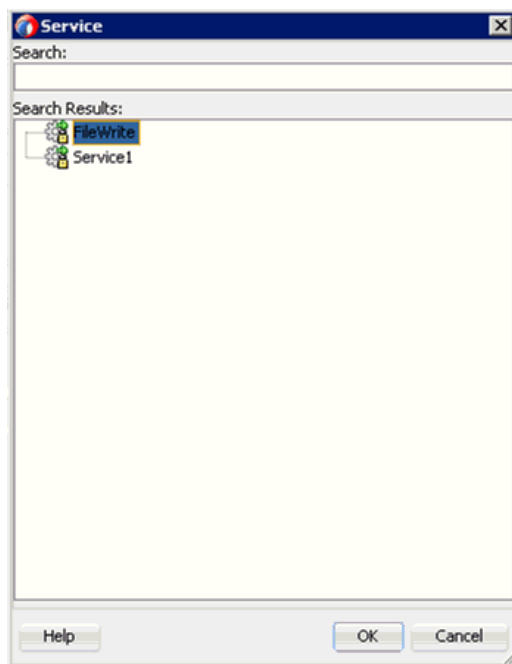
14. 「アクティビティ」アイコンをクリックします。
15. 図 6-77 に示すように、「サービス・タスク」と「End」イベント・コンポーネント間の伝送路に「アクティビティ」アイコンをドロップします。

図 6-77 「アクティビティ」アイコン



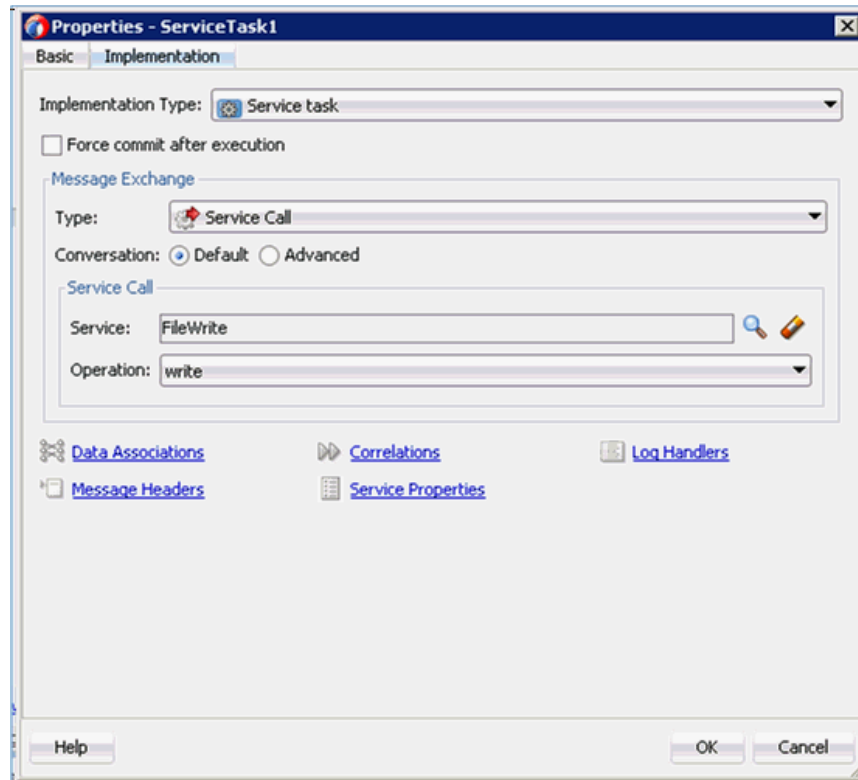
16. 表示された「プロパティ - ServiceTask1」ウィンドウで、「実装」タブをクリックします。
17. 「メッセージ交換」セクションの「タイプ」ドロップダウン・リストから「サービス・コール」を選択します。
18. 「サービス」フィールドの右の「参照」アイコンをクリックします。
19. 図 6-78 に示すように、作成された書込み操作のサービスを選択し、「OK」をクリックします。

図 6-78 「サービス」ウィンドウ



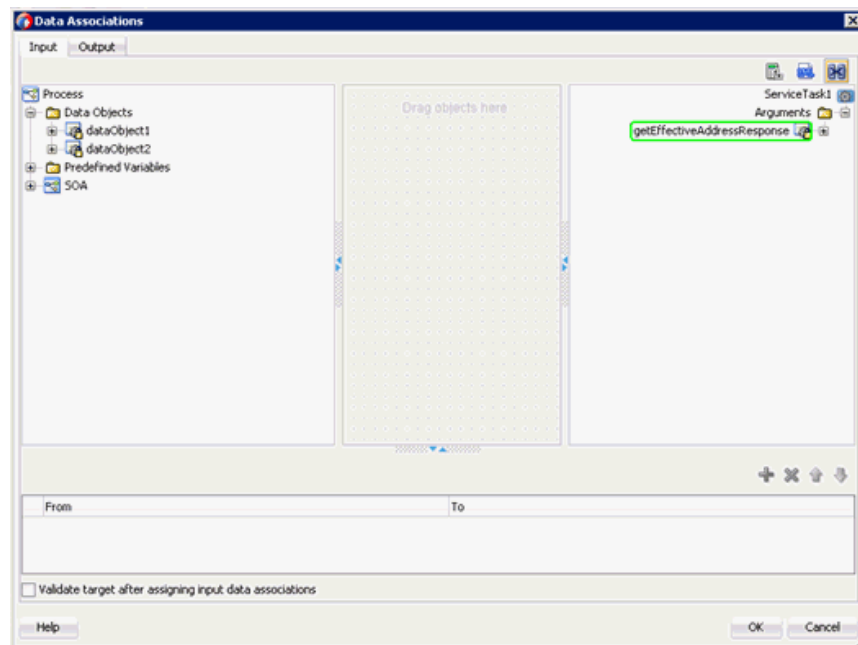
20. 図 6-79 に示すように、「プロパティ - ServiceTask1」ウィンドウで、「データ・アソシエーション」ハイパーリンクをクリックします。

図 6-79 データ・アソシエーション



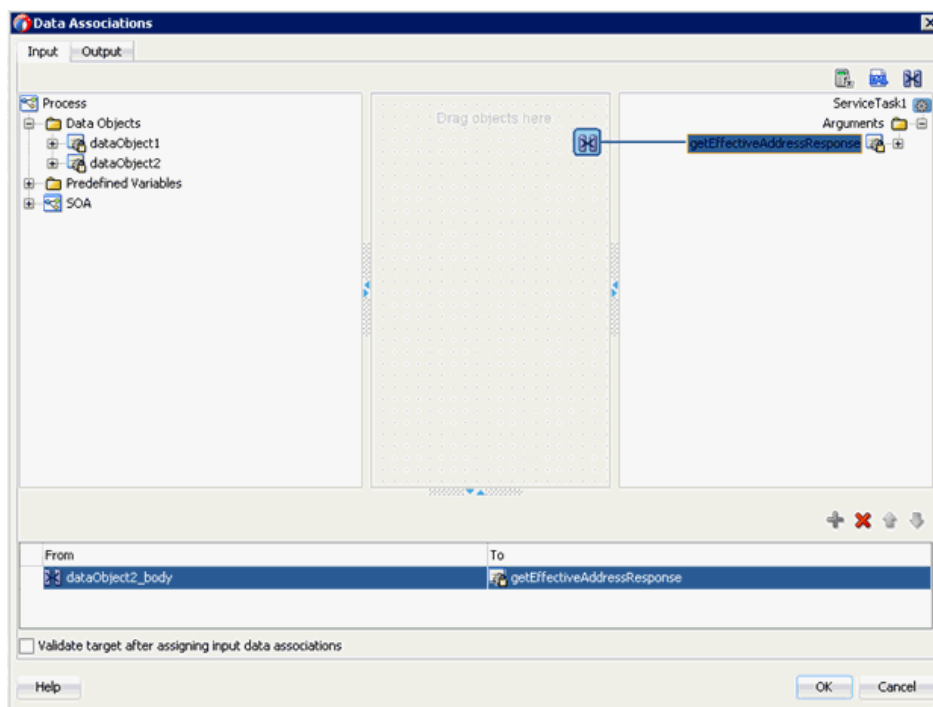
21. 「入力」タブで、右ペインの上部にある「XSL 変換」アイコンをクリックします。
22. 図 6-80 に示すように、「XSL 変換」アイコンを「GetEffectiveAddressResponse」ノードにドラッグ・アンド・ドロップします。

図 6-80 「GetEffectiveAddressResponse」ノード



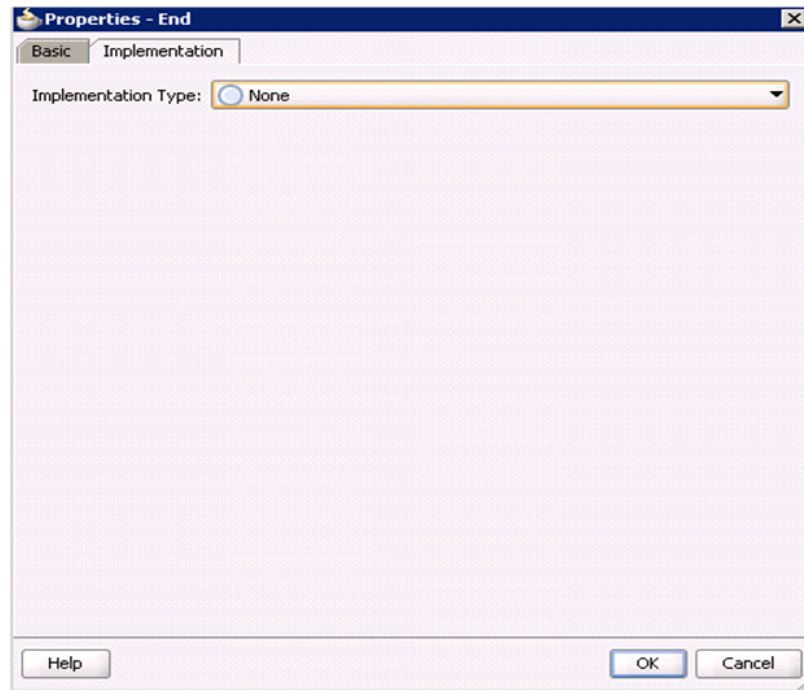
23. 表示された「トランスフォーメーションの作成」ウィンドウで、「ソース」セクションの「dataObject2」を選択し、右矢印記号をクリックします。
24. 「OK」をクリックして、「ターゲット」ドロップダウン・リストで選択されたデフォルト値と「作成」フィールドのデフォルト名を受け入れます。
25. 図 6-81 に示すように、「データ・アソシエーション」ウィンドウで、「OK」をクリックします。

図 6-81 「データ・アソシエーション」ウィンドウ



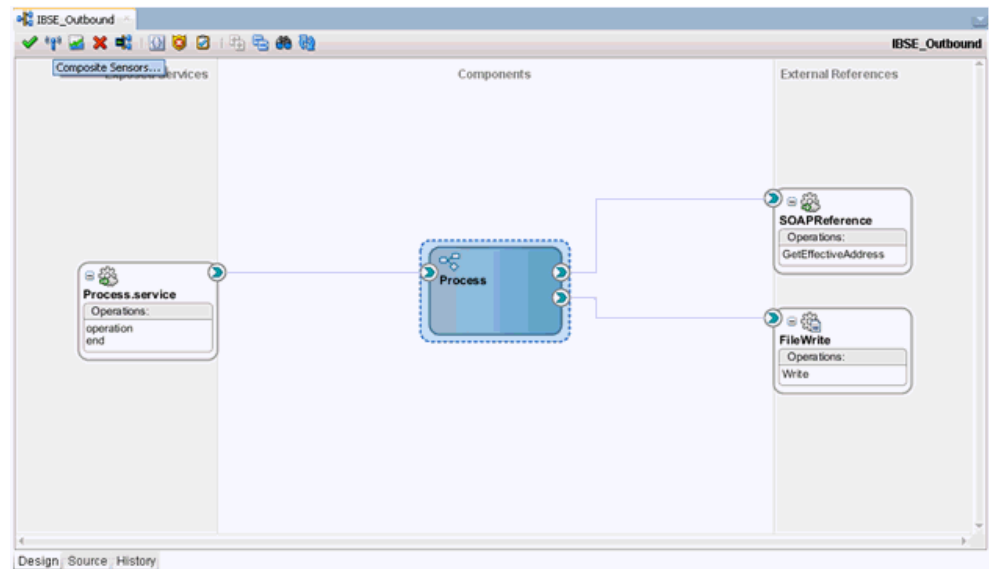
26. 「プロパティ - ServiceTask1」ウィンドウで、「OK」をクリックします。
27. 「response_body.xml」タブで、「ns0:GetEffectiveAddressResponse」ソース要素を「ns0:GetEffectiveAddressResponse」ターゲット要素にマップします。
28. 表示された「自動マップ・プリファレンス」ウィンドウで、デフォルト値を維持し、「OK」をクリックします。
29. 「プロセス」ワークスペース領域に戻り、「End」イベント・コンポーネントをダブルクリックします。
30. 表示された「プロパティ - End」ウィンドウで、「実装」タブをクリックします。
31. 「実装タイプ」ドロップダウン・リストから「なし」を選択します。
32. 図 6-82 に示すように、「OK」をクリックします。

図 6-82 「実装」タブ



33. 図 6-83 に示すように、メニュー・バーで「すべて保存」アイコンをクリックし、構成された新規のアウトバウンド BPM コンポーネントを保存します。

図 6-83 「すべて保存」アイコン



これで、BPM BSE アウトバウンド・プロセスをデプロイする準備ができました。6-25 ページの 6.4.4 項「BPM アウトバウンド・プロセスのデプロイ」の記載と同じ手順に従います。

デプロイ後、6-26 ページの 6.4.5 項「Oracle Enterprise Manager Console での入力 XML ドキュメントの起動」に従って入力 XML を起動できます。

sbconsoleを使用したOracle Service Bus用の アウトバウンドおよびインバウンド・プロセス の構成

注意： リリース 12c (12.2.1.0.0) では、sbconsole を使用した Oracle Service Bus 用のアウトバウンドおよびインバウンド・プロセスの構成方法が変更されています。

sbconsole を使用して Oracle Service Bus 用のプロセスを作成する場合は、『Oracle Fusion Middleware Application Adapters リリース・ノート for 12c (12.2.1.0.0)』の第2章「sbconsole を使用した Oracle Service Bus 用のアウトバウンドおよびインバウンド・プロセスの構成」を参照してください。

Oracle Application Adapter for J.D. Edwards OneWorld は、Oracle Service Bus (OSB) とシームレスに統合されており、Web サービスの簡易な統合を可能にします。OSB は、サービス指向アーキテクチャ (SOA) に基づいており、Web Service Definition Language (WSDL) ドキュメントとして公開されているアダプタ・サービスを使用します。

この章の内容は次のとおりです。

- 7.1 項「Application Adapter と Oracle Service Bus の統合の概要」
- 7.2 項「sbconsole を使用したアウトバウンド・プロセスの構成 (J2CA 構成)」
- 7.3 項「sbconsole を使用したインバウンド・プロセスの構成 (J2CA 構成)」
- 7.4 項「sbconsole を使用したアウトバウンド・プロセスの構成 (BSE 構成)」
- 7.5 項「Oracle Service Bus を使用した JMS プロキシ・サービスの構成 (J2CA 構成)」
- 7.6 項「Oracle Service Bus を使用した HTTP プロキシ・サービスの構成 (J2CA 構成)」

7.1 Application Adapter と Oracle Service Bus の統合の概要

Oracle Service Bus (OSB) と統合するには、Oracle Application Adapter for J.D. Edwards OneWorld が OSB と同じ Oracle WebLogic Server にデプロイされている必要があります。また、基になるアダプタ・サービスは、アダプタのリクエスト/レスポンス (アウトバウンド) とイベント通知 (インバウンド) の両方のサービス用に Oracle Adapter アプリケーション・エクスプローラ (アプリケーション・エクスプローラ) でデザインタイム中に生成される WSDL ファイルとして公開される必要があります。

7.2 sbconsole を使用したアウトバウンド・プロセスの構成 (J2CA 構成)

この項では、J2CA 構成で sbconsole を使用してアウトバウンド・プロセスを構成する方法について説明します。

このアウトバウンド・ユース・ケース・シナリオ用のサンプル・プロジェクトが、次に示す Application Adapter のインストール・フォルダに用意されています。

```
<ADAPTER_HOME>\etc\sample\JDEdwards_Samples.zip\JDEdwards_Samples\OSB\J2CA\JDEdwards_Sample_J2CA_OSB_Outbound_Project
```

この項では、次のトピックについて説明します。

- 7.2.1 項「Oracle Service Bus の起動およびプロジェクト・フォルダの作成」
- 7.2.2 項「Oracle Service Bus 統合のためのアプリケーション・エクスプローラのクラス・パスの設定」
- 7.2.3 項「アプリケーション・エクスプローラから Oracle Service Bus への WSDL の公開」
- 7.2.4 項「WSDL ベースのビジネス・サービスの構成」
- 7.2.5 項「ファイル・タイプ・ビジネス・サービスの構成」
- 7.2.6 項「パイプラインとプロキシ・サービスの構成」

7.2.1 Oracle Service Bus の起動およびプロジェクト・フォルダの作成

この項では、Oracle Service Bus (OSB) を起動してプロジェクト・フォルダを作成する方法について説明します。

Oracle Service Bus を起動してプロジェクト・フォルダを作成するには、次のステップを実行します。

1. 構成済の Oracle WebLogic Server ドメインの Oracle WebLogic Server を起動します。
2. Web ブラウザで次の URL を入力して、Oracle Service Bus コンソールを起動します。

```
http://hostname:port/sbconsole
```

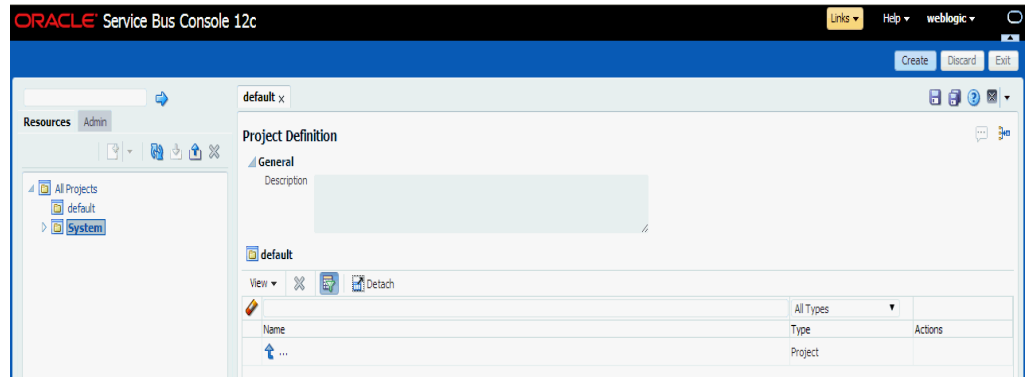
ここで、*hostname* は Oracle WebLogic Server が稼働しているマシンの名前で、*port* は使用しているドメイン用のポートです。

Oracle Service Bus コンソールのログオン・ページが表示されます。

3. 有効なユーザー名およびパスワードを使用して、Oracle Service Bus コンソールにログオンします。

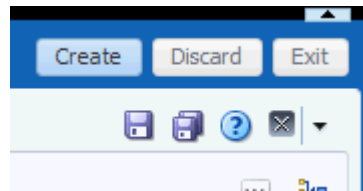
図 7-1 に示すように、Oracle Service Bus コンソールのホーム・ページが表示されます。

図 7-1 Oracle Service Bus コンソールのホーム・ページ



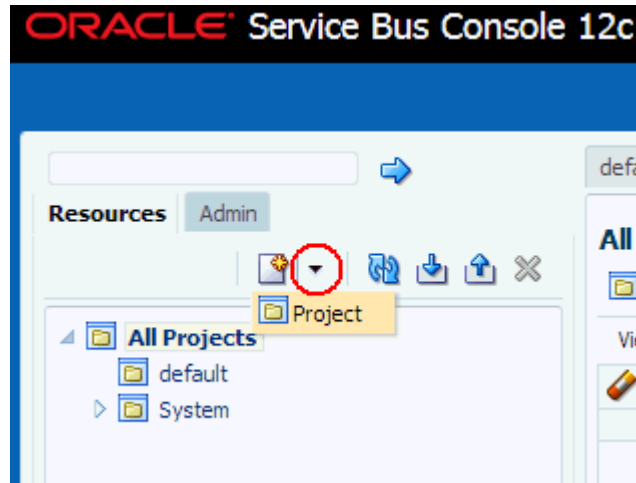
4. 図 7-2 に示すように、Oracle Service Bus セッションの右ペインで「作成」をクリックします。

図 7-2 Oracle Service Bus セッション



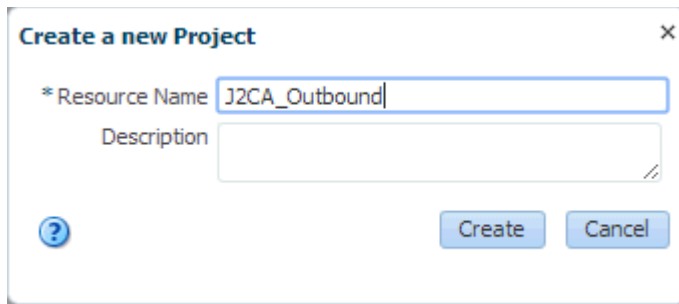
5. 図 7-3 に示すように、「すべてのプロジェクト」を選択し、左ペインの下矢印をクリックして、「プロジェクト」を選択します。

図 7-3 「すべてのプロジェクト」フォルダ



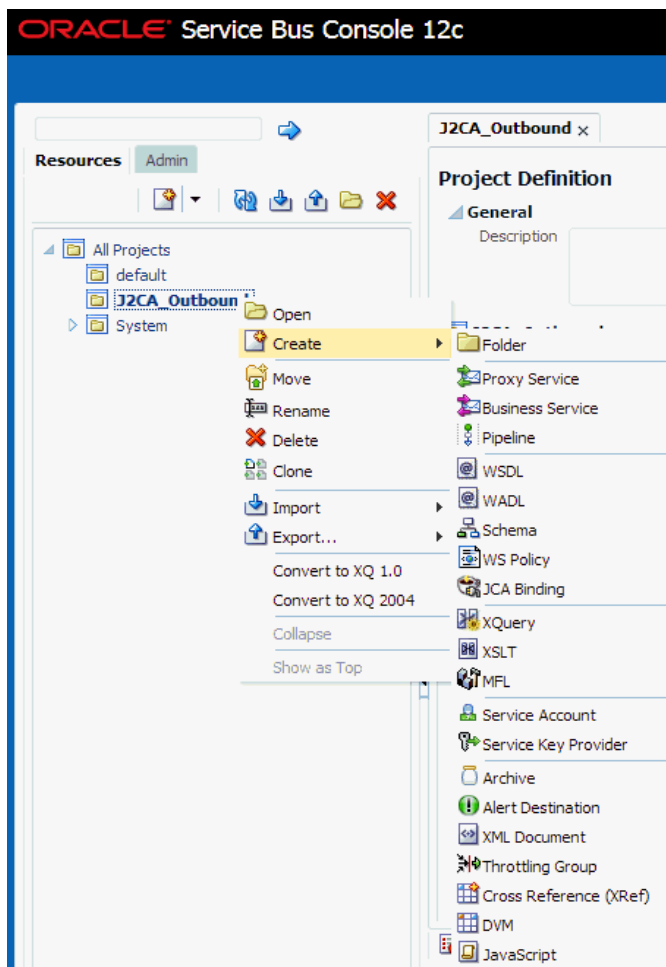
- 図 7-4 に示すように、「新規プロジェクトの作成」ウィンドウが表示されます。

図 7-4 「新規プロジェクトの作成」 ウィンドウ



6. 「リソース名」フィールドに新規プロジェクトの有効な名前 (「J2CA_Outbound」など) を入力し、「作成」をクリックします。
新規プロジェクトが正常に作成され、リストに表示されます。
7. 図 7-5 に示すように、新規に作成されたプロジェクトを右クリックし、「作成」を選択して、「フォルダ」をクリックします。

図 7-5 「作成」オプション



「新規フォルダの作成」ウィンドウが表示されます。

8. 「リソース名」フィールドに「**Business Service**」と入力し、「作成」をクリックします。

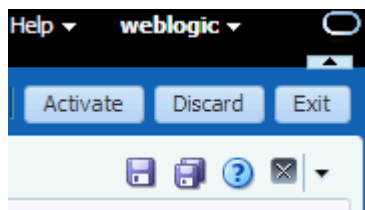
- ステップ7および8を繰り返して、「Proxy Service」という名前のフォルダと「Wsdls」という名前のフォルダを作成します。
図7-6に示すように、「Business Service」フォルダ、「Proxy Service」フォルダ、「Wsdls」フォルダが左ペインのプロジェクト・ノードの下に表示されます。

図7-6 プロジェクト・ノード



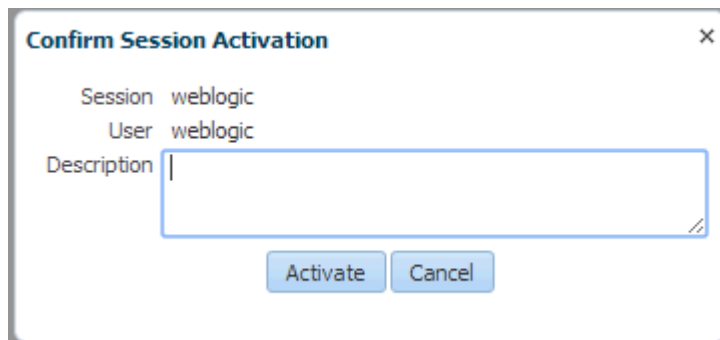
- 図7-7に示すように、Oracle Service Bus セッションの右ペインで「アクティブ化」をクリックします。

図7-7 「アクティブ化」ボタン



- 図7-8に示すように、「セッションのアクティブ化の確認」ページで、「アクティブ化」をクリックして変更を保存します。

図7-8 「セッションのアクティブ化の確認」ウィンドウ



7.2.2 Oracle Service Bus 統合のためのアプリケーション・エクスプローラのクラス・パスの設定

アプリケーション・エクスプローラを起動し、これを使用して WSDL を Oracle Service Bus (OSB) コンソール (プロジェクトまたはフォルダ) に公開する前に、OSB ユーザーは次のステップを実行する必要があります。

1. コマンド・プロンプト・ウィンドウを開きます。
2. 次のディレクトリに移動します。

```
<ORACLE_HOME>\user_projects\domains\base_domain\bin
```
3. **setDomainEnv.cmd** (Windows) または **./setDomainEnv.sh** (UNIX/Linux) を実行します。

このコマンドは、OSB Console を WSDL を公開するためのアプリケーション・エクスプローラから Oracle WebLogic Server API へのアクセス用クラス・パスを設定します。
4. コマンド・プロンプト・ウィンドウを開けたままにします。
5. 次のディレクトリに移動します。

```
<ADAPTER_HOME>\tools\iwae\bin
```
6. **ae.bat** (Windows) または **iwae.sh** (UNIX、Linux) を実行してアプリケーション・エクスプローラを起動します。

これで、アプリケーション・エクスプローラから OSB Console に WSDL を公開する準備ができました。

7.2.3 アプリケーション・エクスプローラから Oracle Service Bus への WSDL の公開

アプリケーション・エクスプローラから Oracle Service Bus に WSDL を公開するには、次のステップを実行します。

1. アプリケーション・エクスプローラを起動し、J2CA 構成および J.D. Edwards OneWorld ターゲットに接続します。

詳細は、[第 2 章「Oracle Application Adapter for J.D. Edwards OneWorld の構成」](#)を参照してください。
2. 接続先の J.D. Edwards OneWorld ターゲットを展開します。
3. メソッドを右クリックし、メニューから **アウトバウンド JCA サービスの作成 (リクエスト/レスポンス)** を選択します。

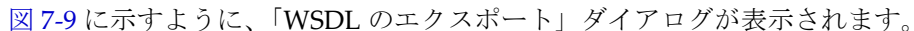
 **図 7-9** に示すように、「WSDL のエクスポート」ダイアログが表示されます。

図 7-9 「WSDL のエクスポート」ダイアログ

4. 「名前」フィールドに、WSDL ファイルのデフォルトのファイル名が表示されます。デフォルトを受け入れるか、任意の名前を指定します。
5. OSB にエクスポート・オプションを選択します。
6. 「ロケーション」フィールドで、WSDL ドキュメントの公開先となる Oracle Service Bus 内のフォルダ名を入力します。

このロケーションは、Oracle Service Bus のプロジェクト名と、オプションで 1 つ以上のフォルダ名で構成されます。プロジェクト名およびフォルダ名はいずれもスラッシュ文字の「/」で区切る必要があります。

7. 「ホスト」フィールドで、Oracle Service Bus がインストールされているマシンの名前を入力します。
8. 「ポート」フィールドで、Oracle Service Bus により使用されるポートを入力します。
9. 「ユーザー」フィールドで、Oracle Service Bus へのアクセス用のユーザー名を入力します。
10. 「パスワード」フィールドで、Oracle Service Bus へのアクセス用のパスワードを入力します。
11. 「OK」をクリックします。

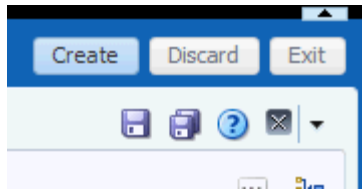
WSDL が「WSDL のエクスポート」ダイアログで指定されている場所に公開され、Oracle Service Bus でビジネス・サービスまたはプロキシ・サービスとともに使用できるようになります。

7.2.4 WSDL ベースのビジネス・サービスの構成

WSDL ベースのビジネス・サービスを構成するには、次のステップを実行します。

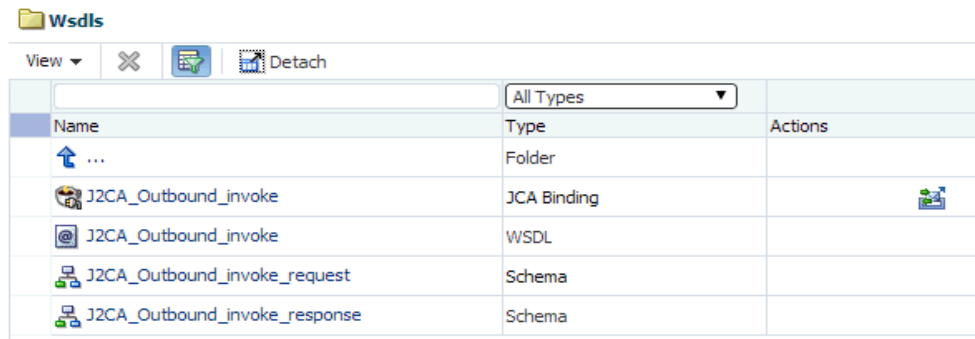
1. 図 7-10 に示すように、Oracle Service Bus コンソールを開き、Oracle Service Bus セッションの右ペインで「作成」をクリックします。

図 7-10 「作成」 ボタン



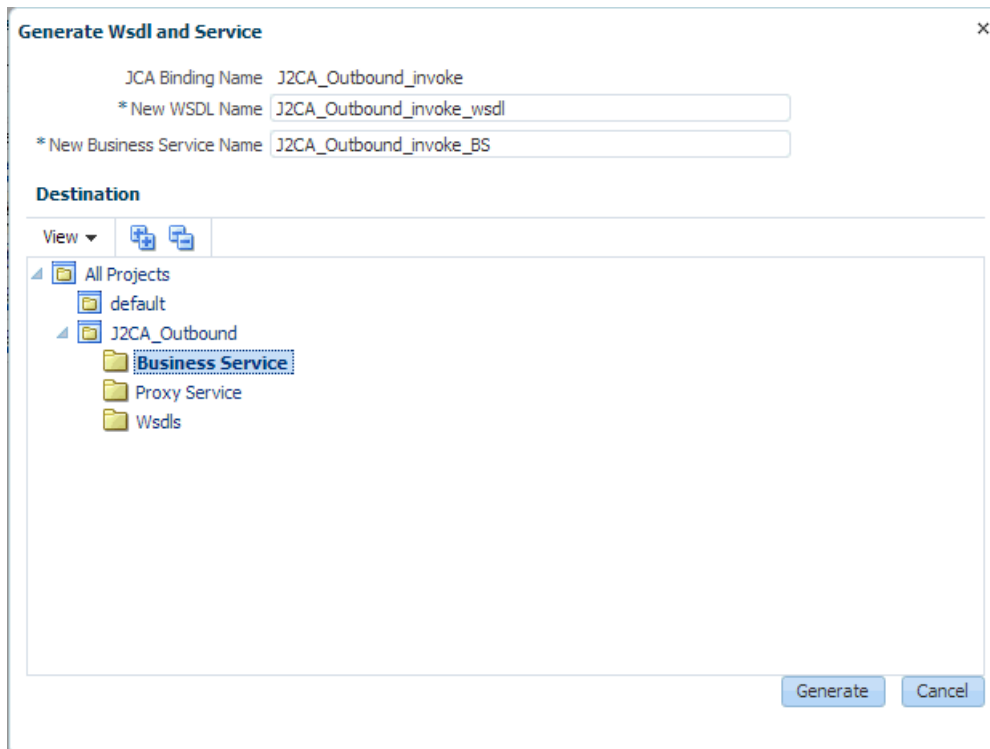
2. 図 7-11 に示すように、左ペインで作成された WSDL フォルダ (「Wsdls」など) をダブルクリックし、右ペインでエクスポートされた WSDL がリストに表示されていることを確認します。

図 7-11 「Wsdls」 フォルダ



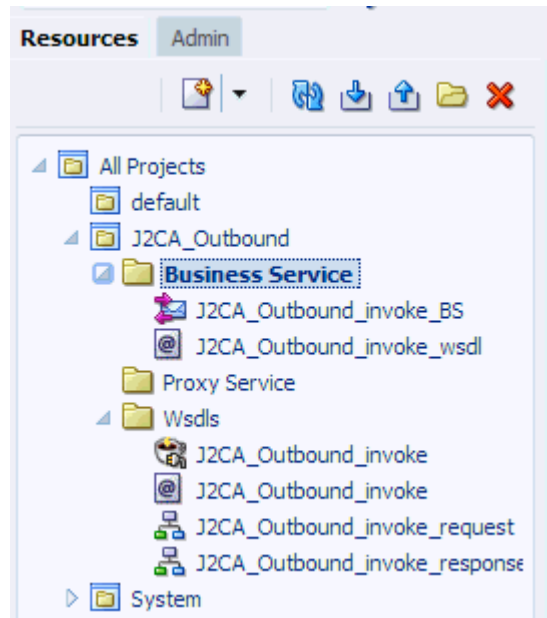
3. 「アクション」列で、JCA バインディングに関連するアイコンをクリックします。
図 7-12 に示すように、「WSDL とサービスの生成」ウィンドウが表示されます。

図 7-12 「WSDL とサービスの生成」ウィンドウ



4. 新規 WSDL の名前および新規ビジネス・サービスの名前を対応するフィールドに入力します。
5. 「宛先」領域で、使用可能なプロジェクトを選択し、ビジネス・サービスに指定されているサブフォルダを選択します。
6. 「生成」をクリックします。
7. 図 7-13 に示すように、プロジェクト・フォルダの下にある「**Business Service**」を展開し、生成された WSDL およびビジネス・サービスがリストに表示されていることを確認します。

図 7-13 「Business Service」フォルダ

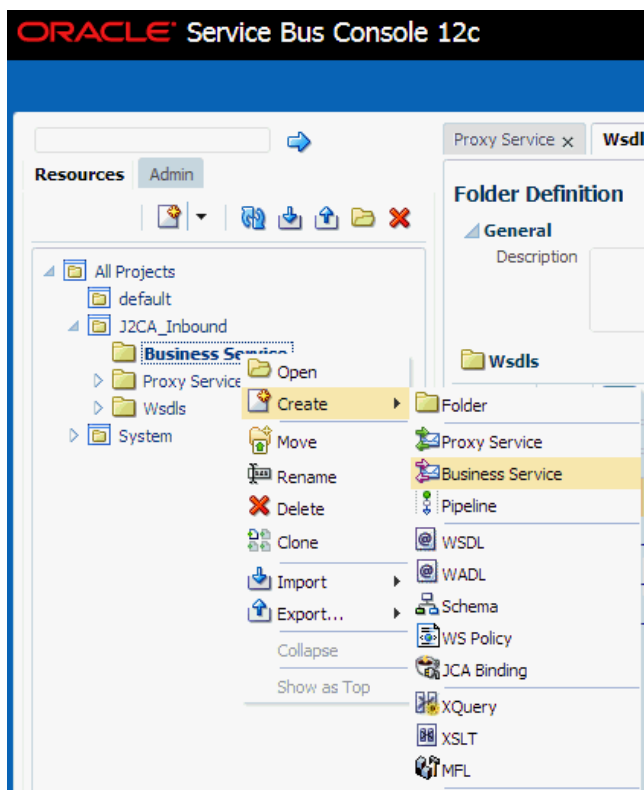


7.2.5 ファイル・タイプ・ビジネス・サービスの構成

ファイル・タイプ・ビジネス・サービスを構成するには、次のステップを実行します。

1. 図 7-14 に示すように、左ペインで作成した「**Business Service**」フォルダを右クリックし、「作成」を選択して、「ビジネス・サービス」をクリックします。

図 7-14 「Business Service」フォルダ



「ビジネス・サービスの作成」ウィンドウが表示されます。

2. 図 7-15 に示すように、「リソース名」フィールドにビジネス・サービスの名前を入力し、「サービス定義」の下にある「トランスポート」セクションで「ファイル」オプションを選択して、「次へ」をクリックします。

図 7-15 サービス定義

Create Business Service

Create Type Transport

Create Service

*Resource Name File_Out

Description

Service Definition

WSDL Based Service

Name

Path

Port/Binding

Transport file

Back Next Create Cancel

- 「サービス・タイプ」セクションで、「メッセージ・サービス」を選択します。デフォルトでは、「リクエスト・タイプ」は「XML」に設定されており、「レスポンス・タイプ」は「なし」に設定されています。図 7-16 に示すように、「次へ」をクリックします。

図 7-16 サービス・タイプの構成ページ

Create Business Service

Create Type Transport

Service Type

WSDL Based Service

Any SOAP Service

Any XML Service

Messaging Service

Request Type XML

Schema Name

Path

Element/Type

Response Type None

Back Next Create Cancel

- 「エンドポイント URI」フィールドで、ファイル・システム上の送信先フォルダへのパスを入力します。

5. 図 7-17 に示すように、「作成」をクリックします。

図 7-17 「トランスポート」 ページ

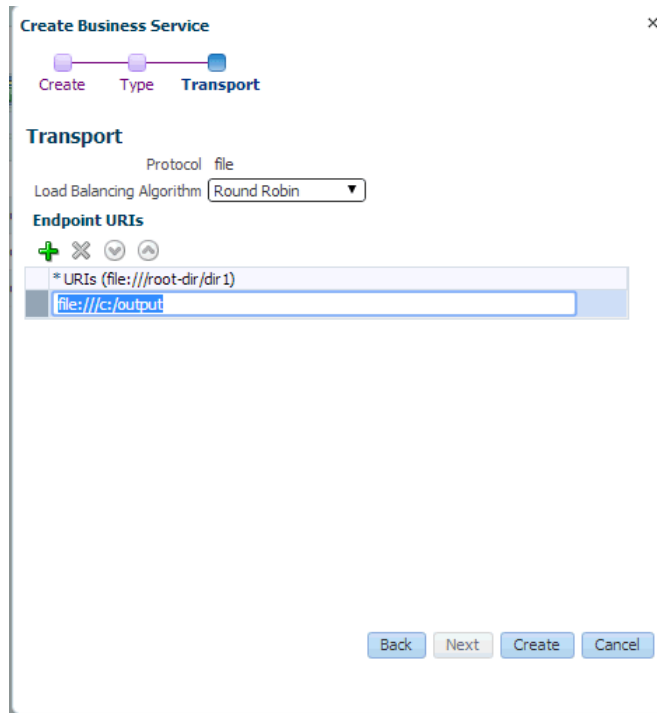
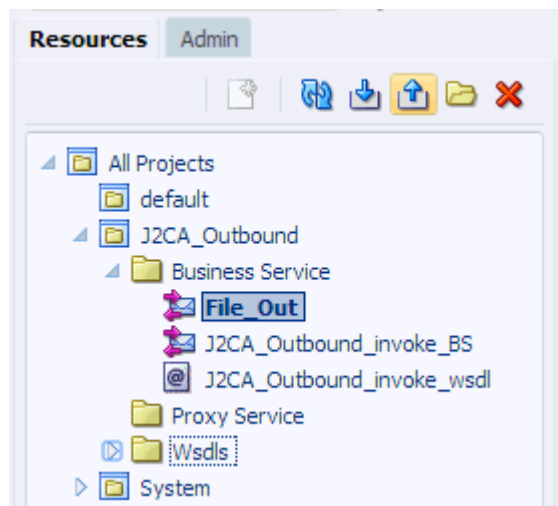


図 7-18 に示すように、「File_Out」ビジネス・サービスが作成され、「Business Service」の下に表示されます。

図 7-18 「File_Out」ビジネス・サービス



6. 図 7-19 に示すように、「File_Out」をダブルクリックし、左ペインで「トランスポートの詳細」をクリックして、受信される出力ファイルの接頭辞および接尾辞を入力します。

図 7-19 トランスポートの詳細

Business Service Definition

Configuration | SLA Alert Rules

General
Transport
Transport Detail
Message Handling
Performance

Transport Detail

Prefix

Suffix

Request encoding

7. 図 7-20 に示すように、右隅の「保存」または「すべて保存」アイコンをクリックします。

図 7-20 「保存」/「すべて保存」アイコン

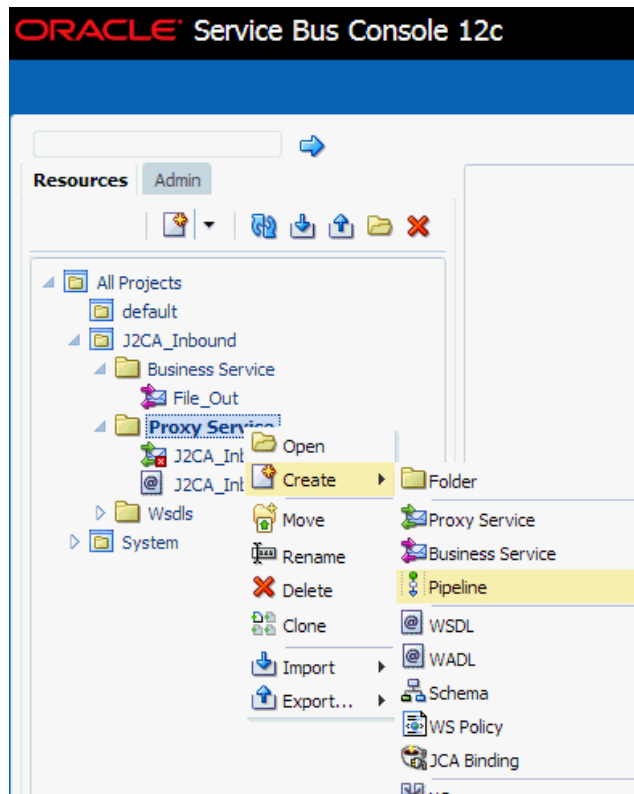


7.2.6 パイプラインとプロキシ・サービスの構成

パイプラインを構成するには、次のステップを実行します。

1. 図 7-21 に示すように、「Proxy Service」フォルダを右クリックし、「作成」を選択して、「パイプライン」をクリックします。

図 7-21 「パイプライン」オプション



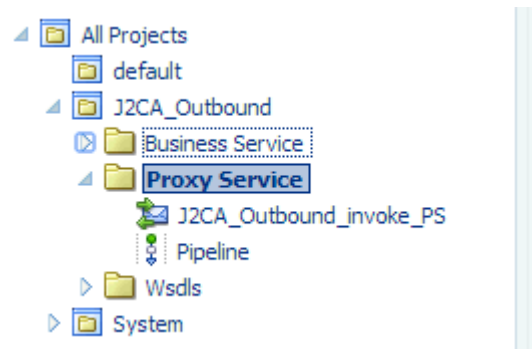
「パイプラインの作成」ウィンドウが表示されます。

2. 「パイプライン名」フィールドに名前を入力します。デフォルトでは、「プロキシ・サービスとして公開」が選択されています。図 7-22 に示すように、プロキシ・サービス名を必要に応じて変更し、「トランスポート」を「ファイル」に設定して、「作成」をクリックします。

図 7-22 「パイプラインの作成」ウィンドウ

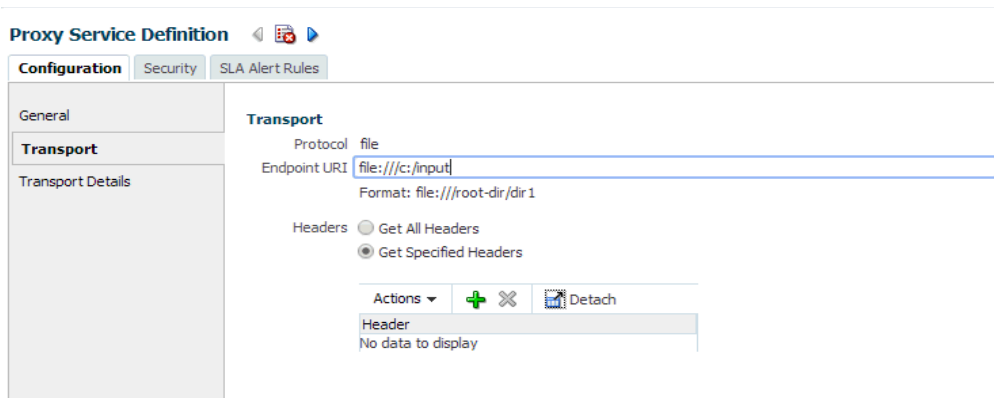
図 7-23 に示すように、作成されたパイプラインとプロキシ・サービスが「Proxy Service」の下に表示されます。

図 7-23 「Pipeline」ノード



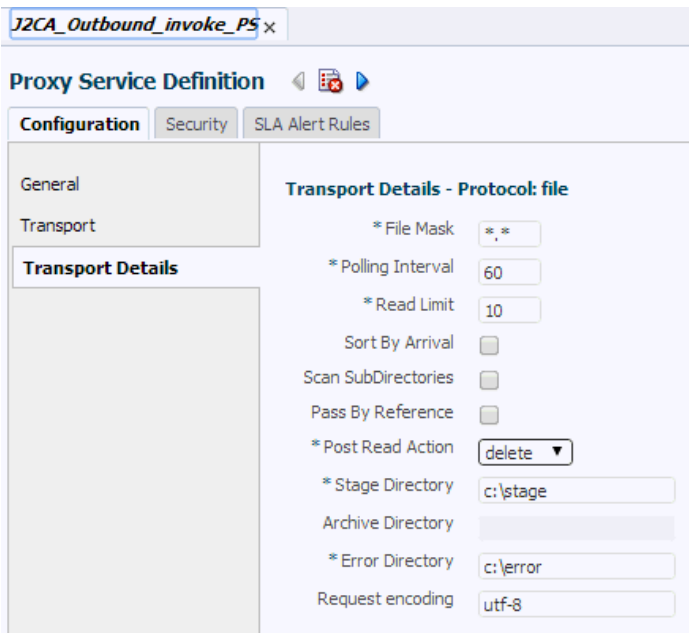
- 作成されたプロキシ・サービスをダブルクリックし、左ペインで「トランスポート」をクリックします。図 7-24 に示すように、「エンドポイント URI」フィールドに入力場所を指定します。

図 7-24 トランスポート



4. 図 7-25 に示すように、左ペインで「トランスポートの詳細」をクリックし、「ステージ・ディレクトリ」フィールドと「エラー・ディレクトリ」フィールドに場所を指定します。

図 7-25 トランスポートの詳細



5. 図 7-26 に示すように、右隅の「すべて保存」アイコンをクリックします。

図 7-26 「すべて保存」アイコン



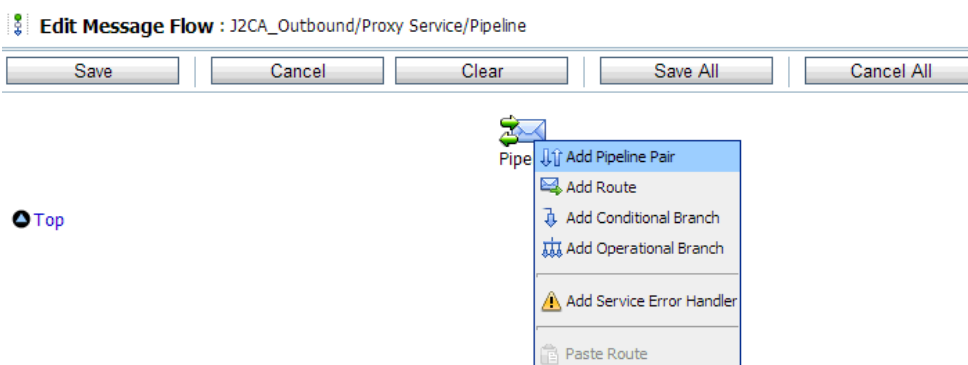
6. 図 7-27 に示すように、「Pipeline」ノードをダブルクリックし、右ペインで「メッセージ・フローを開く」アイコンをクリックして、メッセージ・フローを開きます。

図 7-27 「メッセージ・フローを開く」アイコン



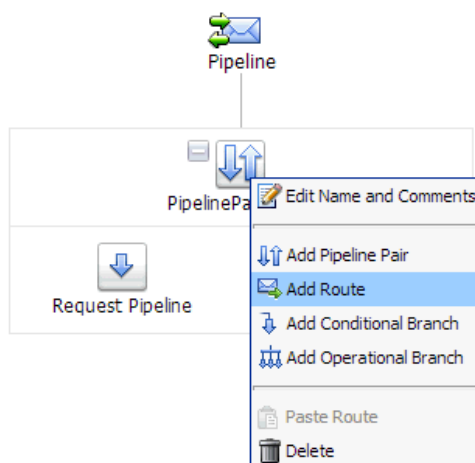
7. 図 7-28 に示すように、「プロキシ・サービス」アイコンをクリックし、メニューから「パイプライン・ペアの追加」を選択します。

図 7-28 「パイプライン・ペアの追加」オプション



8. 図 7-29 に示すように、PipelinePairNode1 アイコンをクリックし、メニューから「ルートへの追加」を選択します。

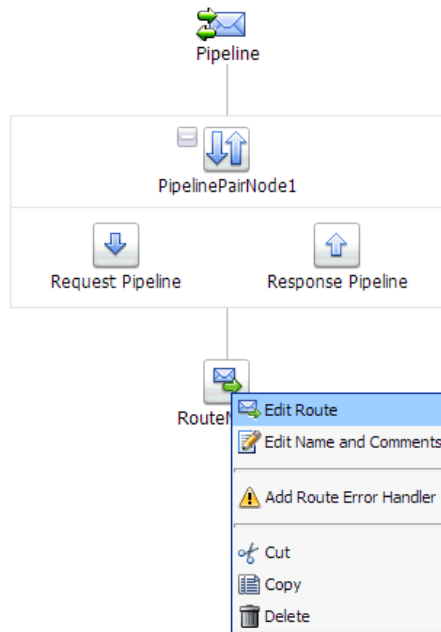
図 7-29 「ルートへの追加」オプション



RouteNode1 アイコンが PipelinePairNode1 アイコンの下に追加されます。

9. 図 7-30 に示すように、RouteNode1 アイコンをクリックし、メニューから「ルートへの編集」を選択します。

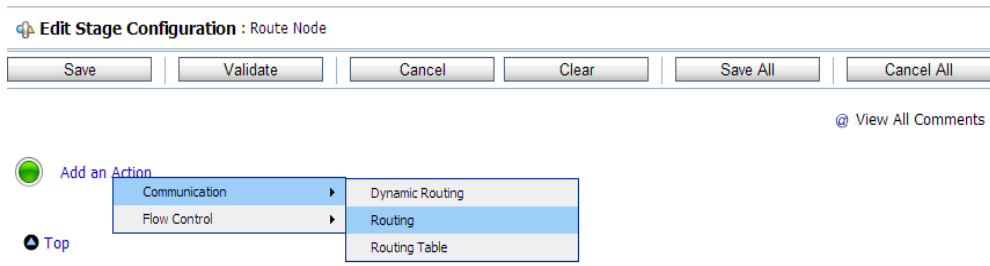
図 7-30 「ルート編集」オプション



「ステージ構成の編集」ワークスペース領域が表示されます。

10. 図 7-31 に示すように、「アクションの追加」をクリックし、「通信」を選択して、「ルーティング」をクリックします。

図 7-31 「ステージ構成の編集」ワークスペース領域



11. 図 7-32 に示すように、<サービス>をクリックします。

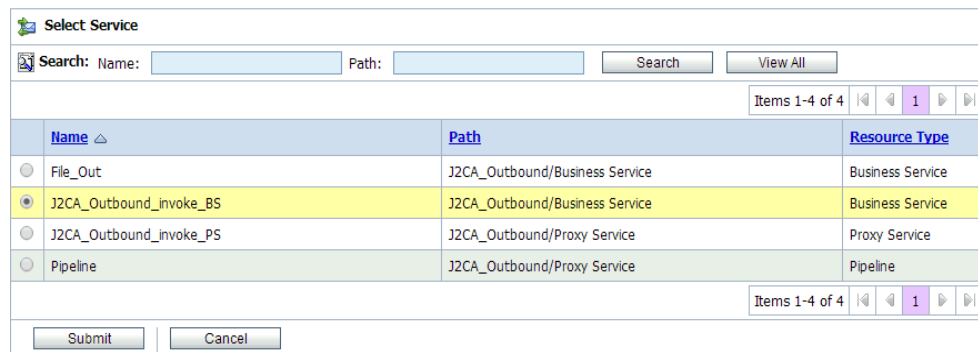
図 7-32 アクション



「サービスの選択」ダイアログが表示されます。

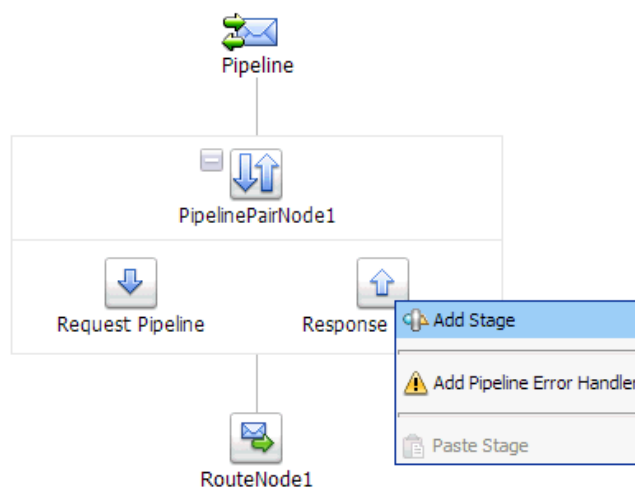
12. 図 7-33 に示すように、J.D. Edwards OneWorld 用に構成された WSDL タイプ・ビジネス・サービスを選択し、「発行」をクリックします。

図 7-33 「サービスの選択」ダイアログ



13. リストから操作属性として J.D. Edwards OneWorld ビジネス・オブジェクトの名前を選択し、「保存」をクリックします。
14. 図 7-34 に示すように、「レスポンス・パイプライン」アイコンをクリックし、メニューから「ステージの追加」を選択します。

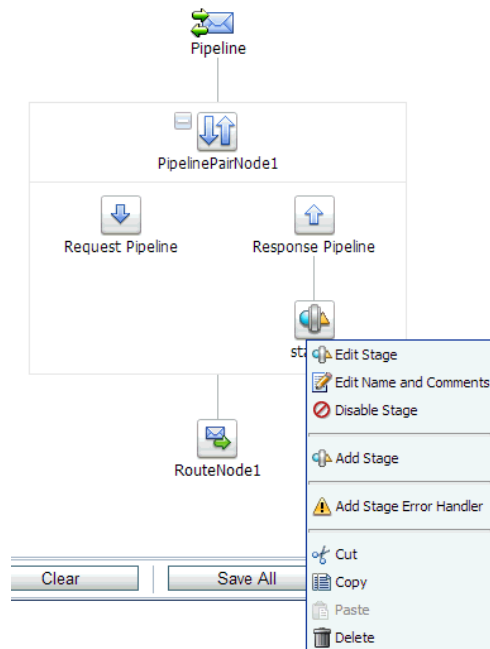
図 7-34 「レスポンス・パイプライン」アイコン



「ステージ 1」アイコンが「レスポンス・パイプライン」アイコンの下に追加されます。

15. 図 7-35 に示すように、「ステージ 1」アイコンをクリックし、メニューから「ステージの編集」を選択します。

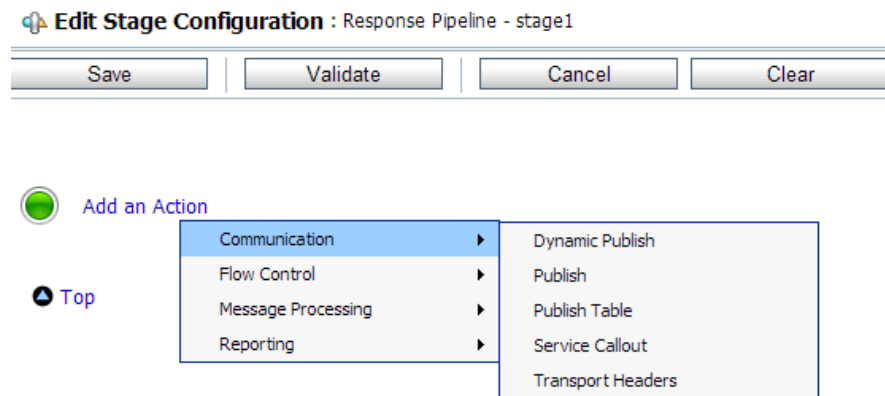
図 7-35 「ステージの編集」オプション



「ステージ構成の編集」ワークスペース領域が表示されます。

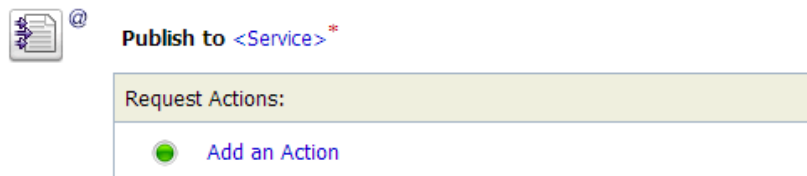
- 図 7-36 に示すように、「アクションの追加」をクリックし、「通信」を選択して、「パブリッシュ」をクリックします。

図 7-36 「ステージ構成の編集」ワークスペース領域



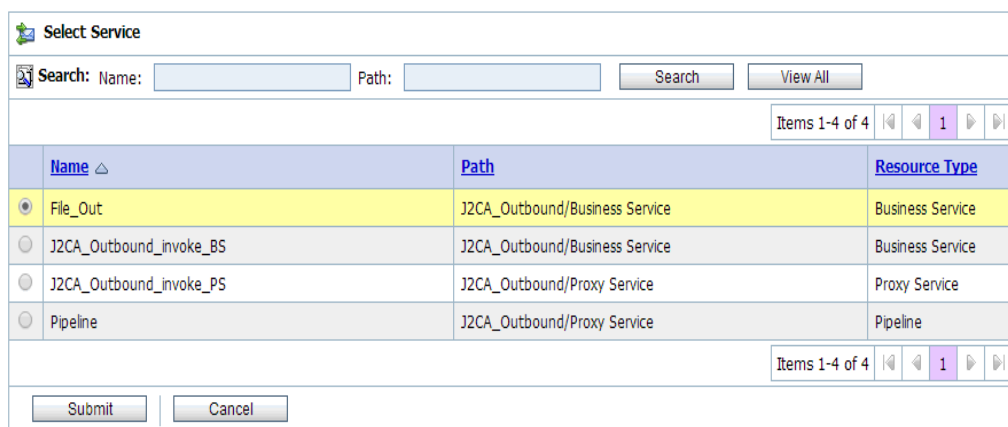
- 図 7-37 に示すように、<サービス>をクリックします。

図 7-37 < サービス > アクション



18. 図 7-38 に示すように、「サービスの選択」ダイアログで、ファイル・タイプ・ビジネス・サービスを選択し、「発行」をクリックします。

図 7-38 「サービスの選択」ダイアログ



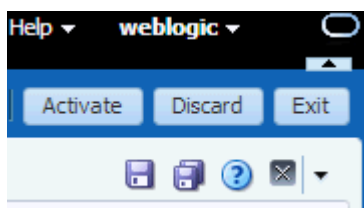
19. 図 7-39 に示すように、「すべて保存」をクリックします。

図 7-39 「すべて保存」ボタン



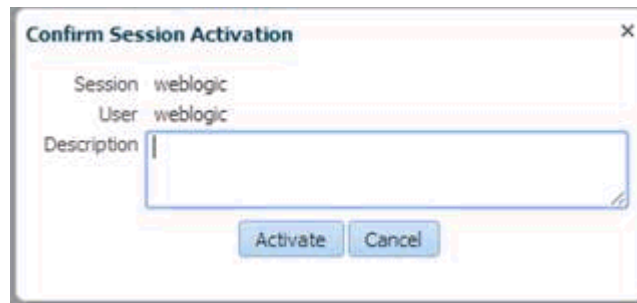
20. 図 7-40 に示すように、Oracle Service Bus セッションの右ペインで「アクティブ化」をクリックします。

図 7-40 「アクティブ化」ボタン



21. 図 7-41 に示すように、「アクティブ化」をクリックして変更を保存します。

図 7-41 セッションのアクティブ化の確認



22. 入力 XML ファイルをコピーして構成済の入力フォルダ (「C:\input」など) に貼り付けます。構成済の出力場所 (「C:\output」など) で出力が受信されたことを確認します。

7.3 sbconsole を使用したインバウンド・プロセスの構成 (J2CA 構成)

この項では、J2CA 構成で sbconsole を使用してインバウンド・プロセスを構成する方法について説明します。

このインバウンド・ユース・ケース・シナリオ用のサンプル・プロジェクトが、次に示す Application Adapter のインストール・フォルダに用意されています。

```
<ADAPTER_HOME>\etc\sample\JDEdwards_Samples.zip\JDEdwards_Samples\OSB\J2CA\JDEdwards_Sample_J2CA_OSB_Inbound_Project
```

この項では、次のトピックについて説明します。

- 7.3.1 項「Oracle Service Bus の起動およびプロジェクト・フォルダの作成」
- 7.3.2 項「Oracle Service Bus 統合のためのアプリケーション・エクスペローラのクラス・パスの設定」
- 7.3.3 項「イベント統合のための WSDL の生成」
- 7.3.4 項「WSDL ベースのプロキシ・サービスの構成」
- 7.3.5 項「ファイル・タイプ・ビジネス・サービスの構成」
- 7.3.6 項「パイプラインの構成」

7.3.1 Oracle Service Bus の起動およびプロジェクト・フォルダの作成

Oracle Service Bus の起動およびプロジェクト・フォルダの作成の詳細は、7.2.1 項「Oracle Service Bus の起動およびプロジェクト・フォルダの作成」(7-2 ページ)を参照してください。

7.3.2 Oracle Service Bus 統合のためのアプリケーション・エクスペローラのクラス・パスの設定

Oracle Service Bus 統合のためのアプリケーション・エクスペローラのクラス・パスの設定の詳細は、7.2.2 項「Oracle Service Bus 統合のためのアプリケーション・エクスペローラのクラス・パスの設定」(7-6 ページ)を参照してください。

7.3.3 イベント統合のための WSDL の生成

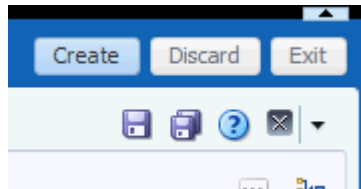
アプリケーション・エクスプローラを使用して J.D. Edwards OneWorld のイベント通知用にインバウンド WSDL を公開できません。コマンド・プロンプトから WSDL を生成するには、4-35 ページの 4.5.1 項「イベント統合のための WSDL の生成」を参照してください。

7.3.4 WSDL ベースのプロキシ・サービスの構成

ファイル・システムからインバウンド WSDL を選択して WSDL ベースのプロキシ・サービスを構成するには、次のステップを実行します。

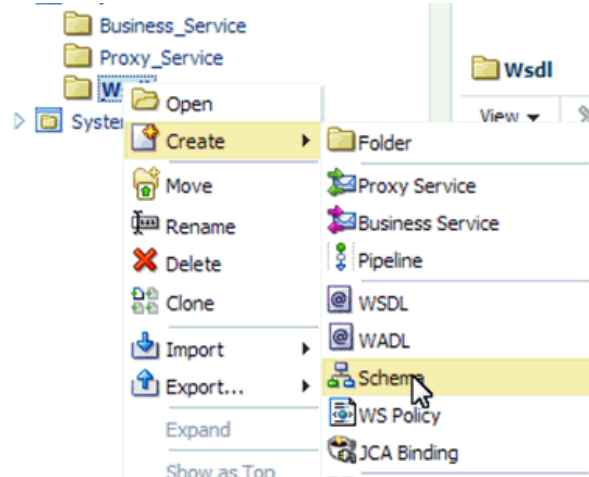
1. 図 7-42 に示すように、Oracle Service Bus コンソールを開き、Oracle Service Bus セッションの右ペインで「作成」をクリックします。

図 7-42 「作成」ボタン



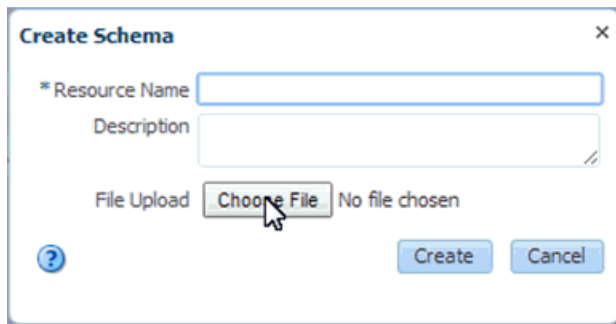
2. 図 7-43 に示すように、WSDL フォルダを右クリックし、「作成」を選択して、「スキーマ」をクリックします。

図 7-43 「スキーマ」オプション



3. 図 7-44 に示すように、表示されたウィンドウで「ファイルの選択」をクリックし、使用可能なスキーマ・ファイル (「J2CA_Inbound_receive_request.xsd」など) を選択します。

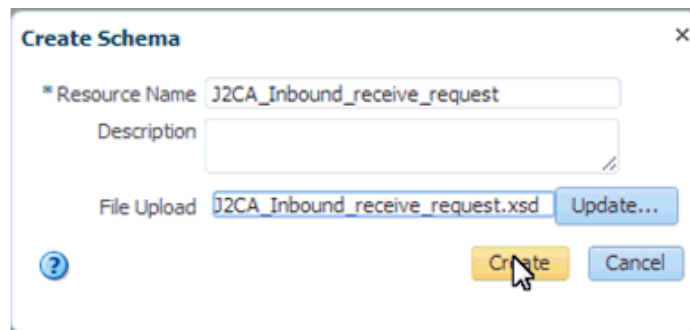
図 7-44 「ファイルの選択」 ボタン



「リソース名」はデフォルトで追加されます。

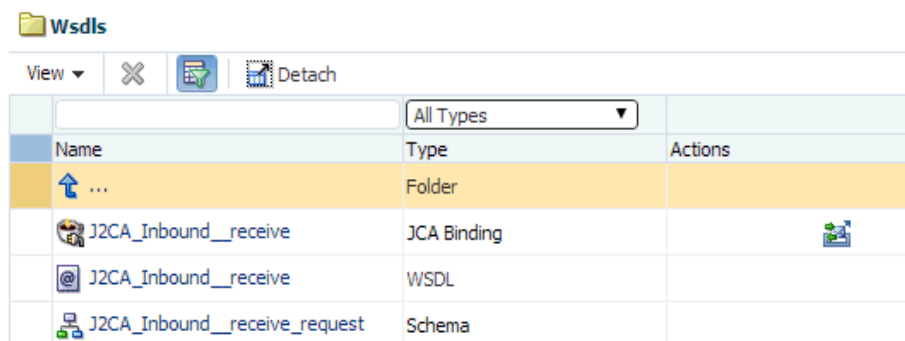
4. 図 7-45 に示すように、「作成」をクリックします。

図 7-45 「作成」 ボタン



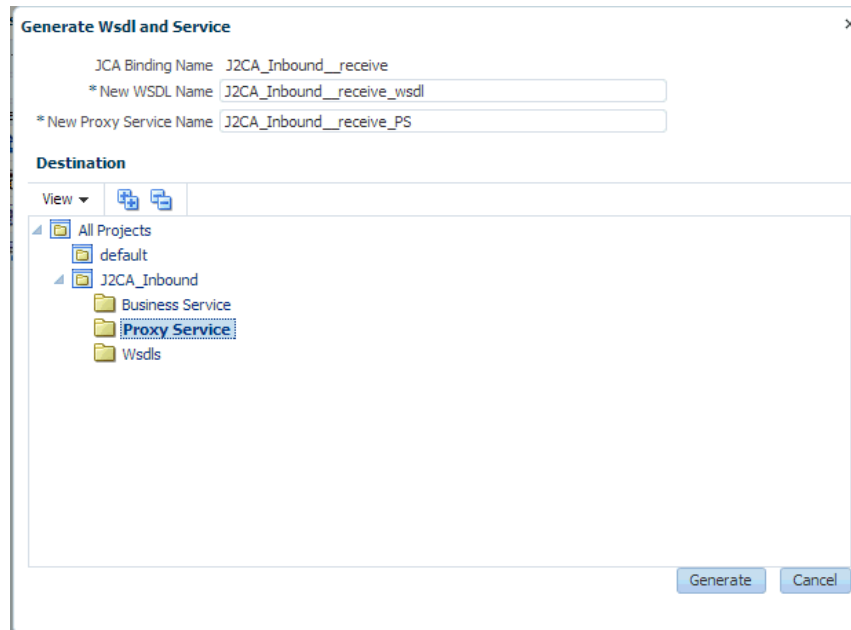
5. WSDL フォルダを右クリックし、「作成」を選択して、「WSDL」をクリックします。前のステップを繰り返して、WSDL ファイルを選択します。
6. WSDL フォルダを右クリックし、「作成」を選択して、「JCA バインド」をクリックします。前のステップを繰り返して、JCA ファイルを選択します。
7. 図 7-46 に示すように、左ペインで作成された WSDL フォルダ (「Wsdl」など) をダブルクリックし、右ペインで WSDL がリストに表示されていることを確認します。

図 7-46 エクスポートされた WSDL



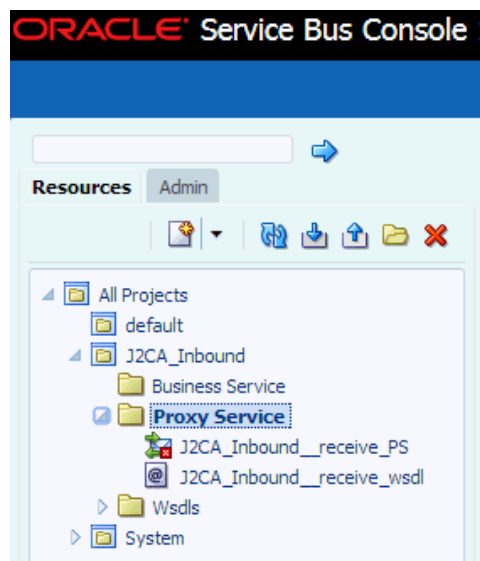
8. 「アクション」列で、JCA バインディングに関連するアイコンをクリックします。
図 7-47 に示すように、「WSDL とサービスの生成」ページが表示されます。

図 7-47 「WSDL とサービスの生成」 ページ



9. 新規 WSDL の名前および新規プロキシ・サービスの名前を対応するフィールドに入力します。
10. 「宛先」領域で、プロキシ・サービスに指定されている使用可能なプロジェクトおよびサブフォルダを選択します。
11. 「生成」をクリックします。
12. 図 7-48 に示すように、「プロジェクト・エクスプローラ」の下にある「Proxy Service」を展開し、生成された WSDL およびプロキシ・サービスがリストに表示されていることを確認します。

図 7-48 生成された WSDL

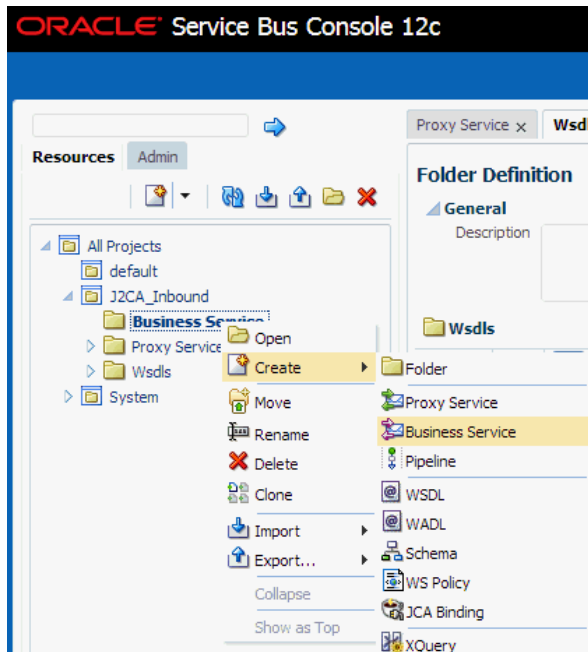


7.3.5 ファイル・タイプ・ビジネス・サービスの構成

ファイル・タイプ・ビジネス・サービスを構成するには、次のステップを実行します。

1. 図 7-49 に示すように、左ペインで作成した「Business Service」フォルダを右クリックし、「作成」を選択して、「ビジネス・サービス」をクリックします。

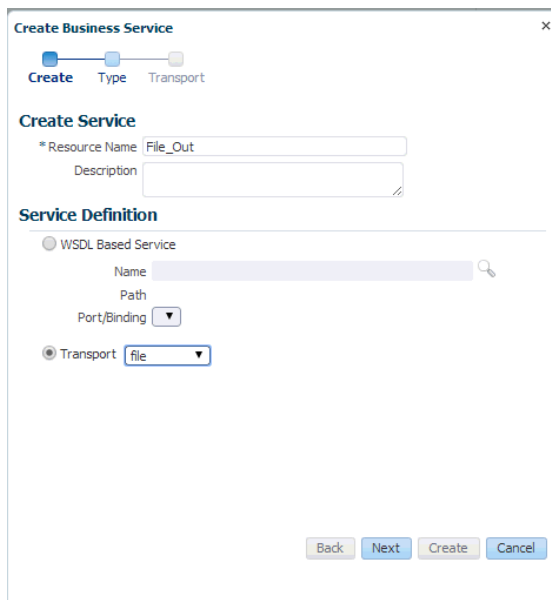
図 7-49 「Business Service」フォルダ



「ビジネス・サービスの作成」ウィンドウが表示されます。

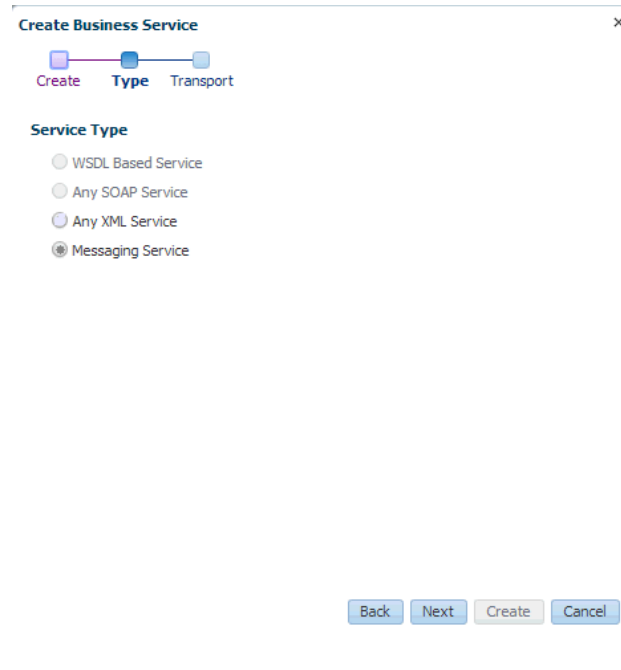
2. 図 7-50 に示すように、「リソース名」フィールドにビジネス・サービスの名前を入力し、「サービス定義」領域の「トランスポート」ドロップダウン・リストから「ファイル」オプションを選択します。

図 7-50 「ビジネス・サービスの作成」ウィンドウ



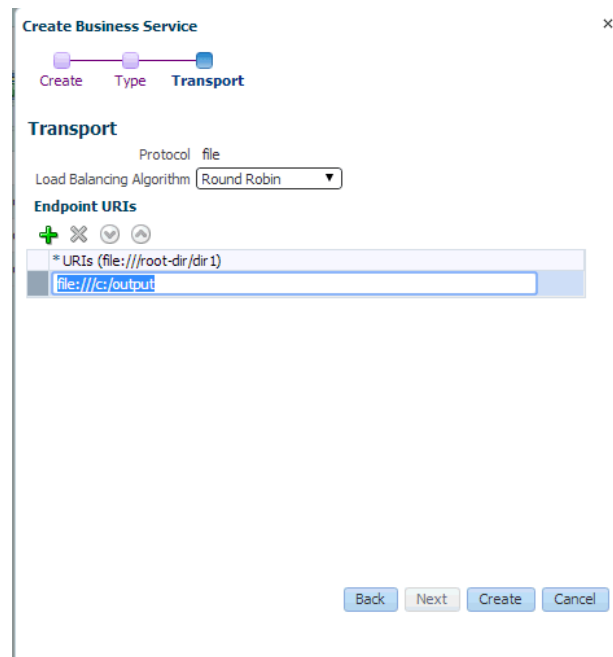
3. 「次へ」をクリックします。
4. 図 7-51 に示すように、「サービス・タイプ」領域で、サービス・タイプとして「メッセージ・サービス」を選択します。

図 7-51 「サービス・タイプ」領域



5. 「次へ」をクリックします。
図 7-52 に示すように、「トランスポート」ページが表示されます。

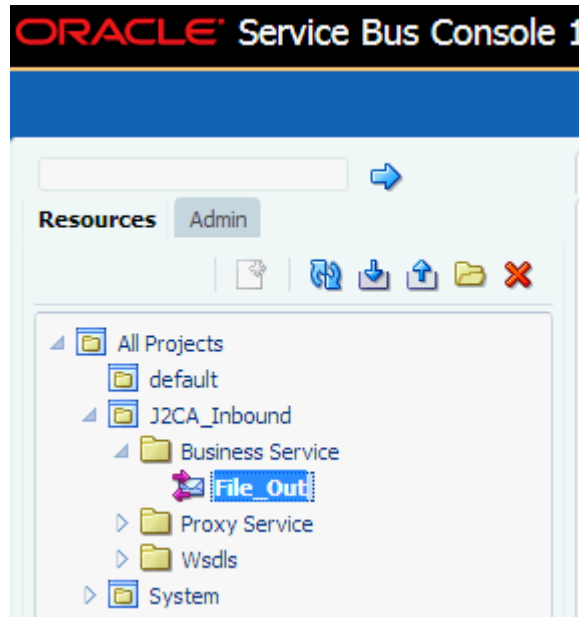
図 7-52 「トランスポート」ページ



- 「エンドポイント URI」フィールドで、ファイル・システム上の送信先フォルダへのパスを入力し、「作成」をクリックします。

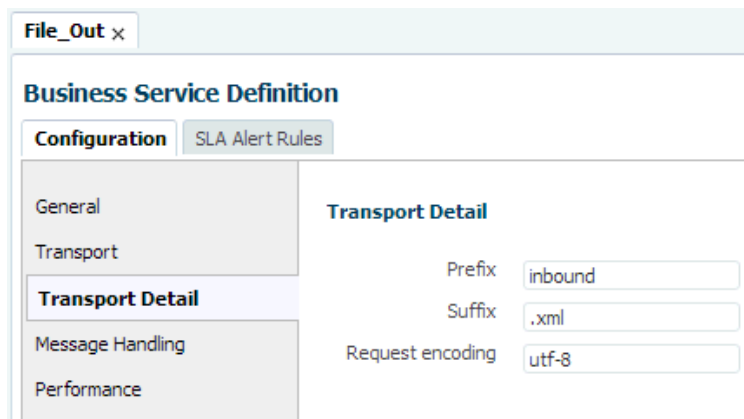
図 7-53 に示すように、「File_Out」ビジネス・サービスが作成され、「Business Service」の下に表示されます。

図 7-53 「File_Out」ビジネス・サービス



- 図 7-54 に示すように、「File_Out」をダブルクリックし、左ペインで「トランスポートの詳細」をクリックして、受信される出力ファイルの接頭辞および接尾辞を入力します。

図 7-54 「トランスポートの詳細」ページ



- 図 7-55 に示すように、右隅の「保存」または「すべて保存」アイコンをクリックします。

図 7-55 保存アイコン

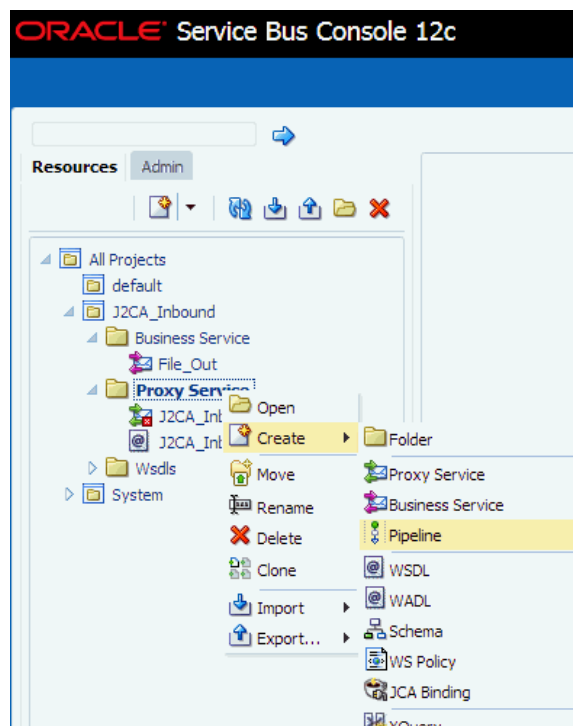


7.3.6 パイプラインの構成

パイプラインを構成するには、次のステップを実行します。

1. 図 7-56 に示すように、作成したプロキシ・サービスを右クリックし、「作成」を選択して、「パイプライン」をクリックします。

図 7-56 「パイプライン」オプション



「パイプラインの作成」ウィンドウが表示されます。

2. 図 7-57 に示すように、「パイプライン名」フィールドに名前を入力し、サービス・タイプとして「WSDL ベース・サービス」を選択します。

図 7-57 「パイプラインの作成」ウィンドウ

3. 「検索」アイコンをクリックし、図 7-58 に示すように、表示された「検索と選択: WSDL リソース」ウィンドウで、「J2CA_Inbound_receive_wsdl」を選択し、「OK」をクリックします。

図 7-58 「検索と選択: WSDL リソース」ウィンドウ

Name	Path	Namespace
J2CA_Inbound__receive_wsdl	J2CA_Inbound/Proxy Service	http://xmlns.oracle...
J2CA_Inbound__receive	J2CA_Inbound/Wsdl	http://xmlns.oracle...

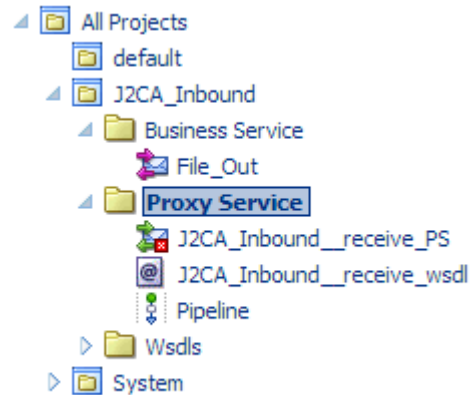
「パイプラインの作成」ウィンドウが開きます。

4. 図 7-59 に示すように、「プロキシ・サービスとして公開」のチェック・ボックスを選択解除し、「作成」をクリックします。

図 7-59 「パイプラインの作成」ウィンドウ

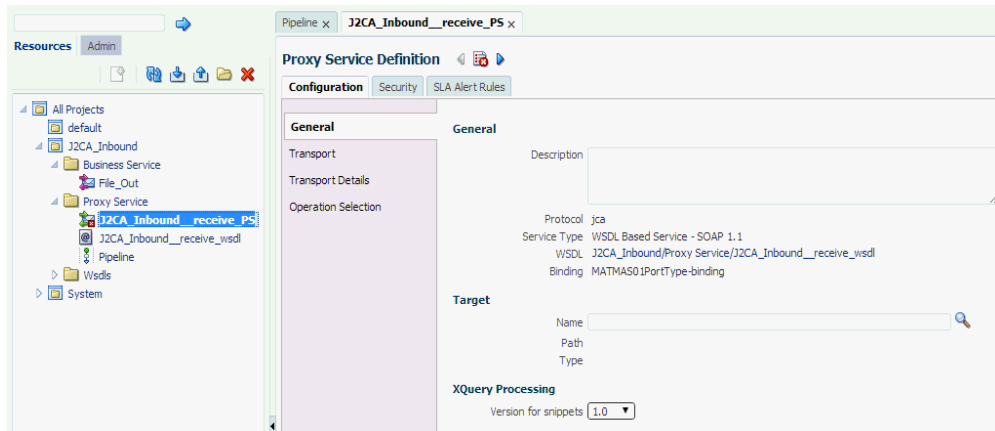
図 7-60 に示すように、パイプラインが作成され、「Proxy Service」の下に表示されます。

図 7-60 プロキシ・サービス・パイプライン



5. 図 7-61 に示すように、左ペインで「Proxy Service」の下にある「J2CA_Inbound_receive_PS」ノードをダブルクリックし、右ペインで「対象」領域の「検索」アイコンをクリックします。

図 7-61 「プロキシ・サービス定義」 ウィンドウ

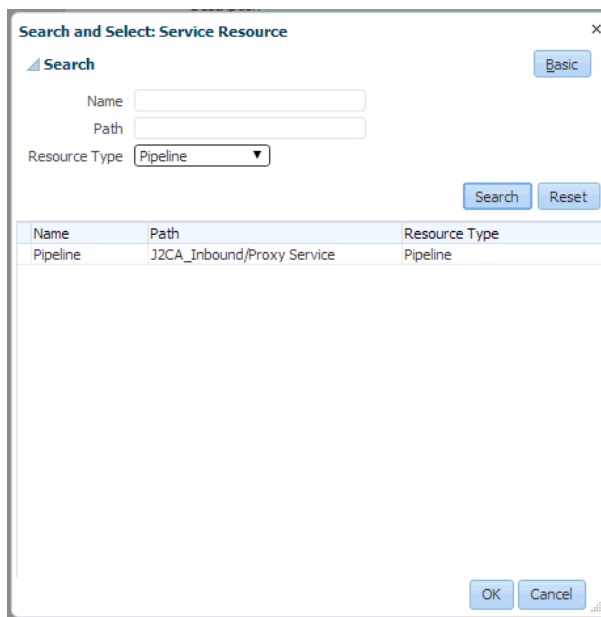


「検索と選択: サービス・リソース」 ウィンドウが表示されます。

- 「リソース・タイプ」 ドロップダウン・リストから「パイプライン」を選択し、「検索」 ボタンをクリックします。

図 7-62 に示すように、パイプラインが表示されます。

図 7-62 「検索と選択: サービス・リソース」 ウィンドウ



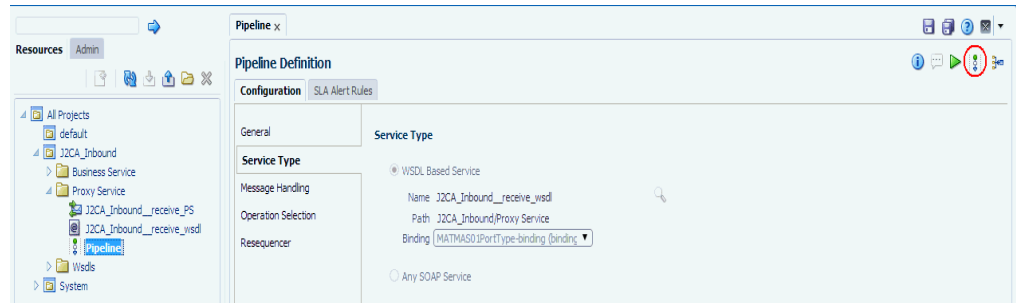
- パイプラインを選択し、「OK」 をクリックします。
- 図 7-63 に示すように、右隅の「保存」または「すべて保存」 アイコンをクリックします。

図 7-63 「保存」アイコンと「すべて保存」アイコン



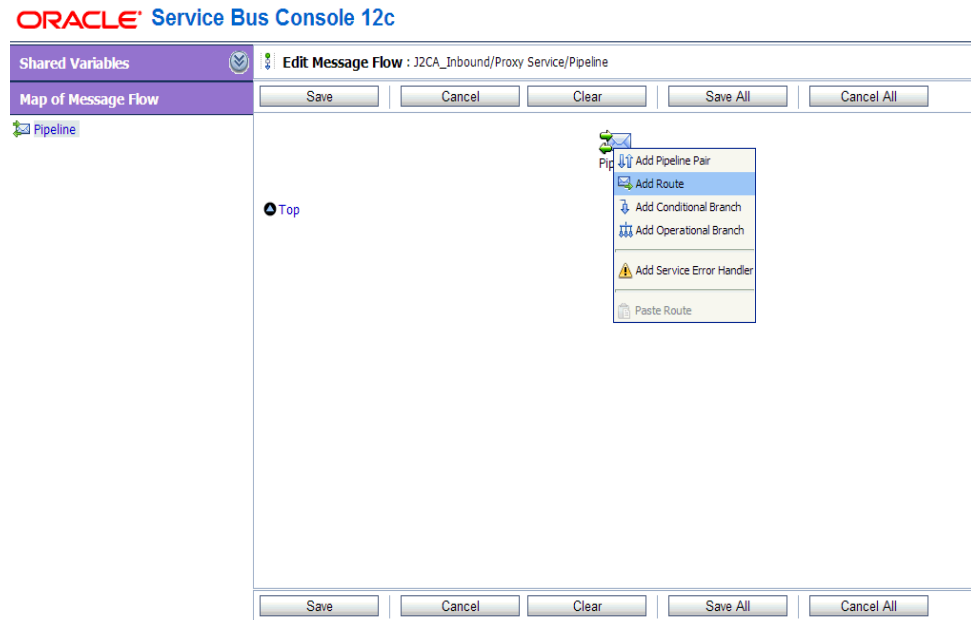
9. 図 7-64 に示すように、左ペインで「Proxy Service」フォルダの下にある「Pipeline」をダブルクリックし、右ペインで下向きのアイコンをクリックしてメッセージ・フローを開きます。

図 7-64 メッセージ・フロー



10. 図 7-65 に示すように、表示されたプロキシ・サービス・アイコンをクリックし、メニューから「ルートの追加」を選択します。

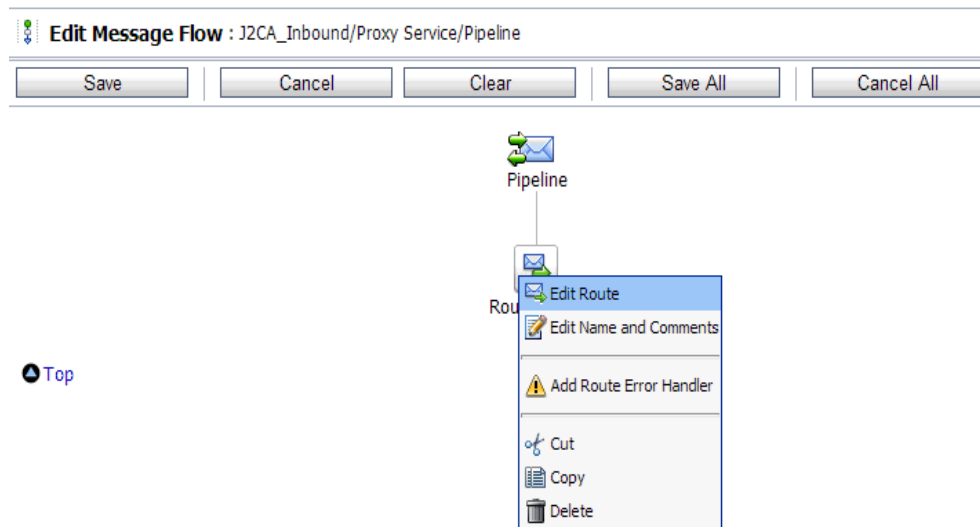
図 7-65 「ルートの追加」オプション



RouteNode1 アイコンが追加されます。

11. 図 7-66 に示すように、RouteNode1 アイコンをクリックし、メニューから「ルートの編集」を選択します。

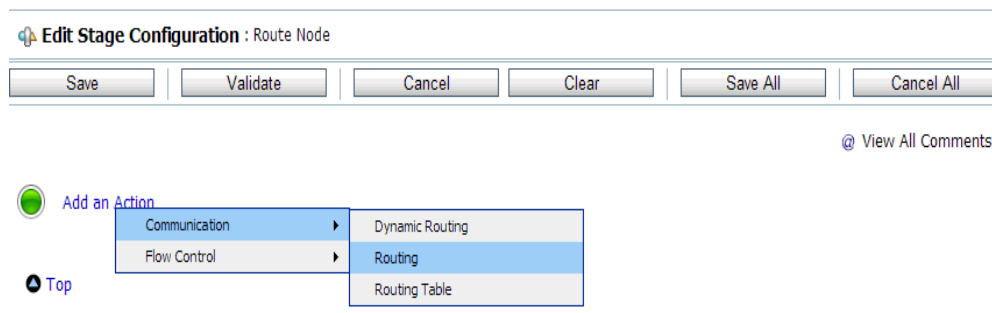
図 7-66 「ルート編集」オプション



「ステージ構成の編集」ワークスペース領域が表示されます。

12. 図 7-67 に示すように、「アクションの追加」をクリックし、メニューから「通信」を選択して、「ルーティング」をクリックします。

図 7-67 「ステージ構成の編集」ワークスペース



13. 図 7-68 に示すように、<サービス>をクリックします。

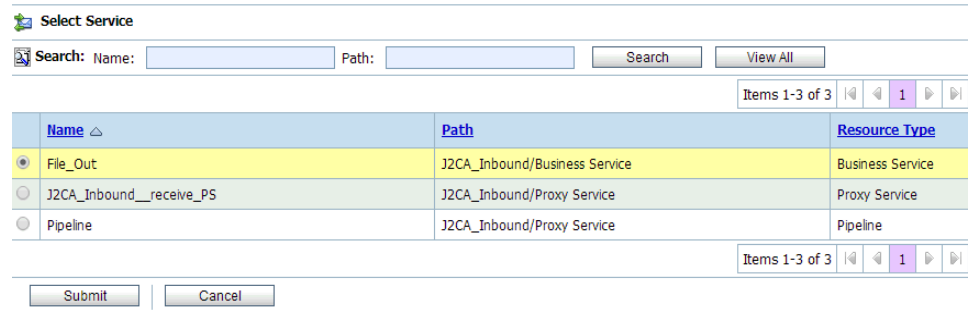
図 7-68 サービスへのルーティング・アクション



「サービスの選択」ダイアログが表示されます。

14. 図 7-69 に示すように、「File_Out」ビジネス・サービスを選択し、「発行」をクリックします。

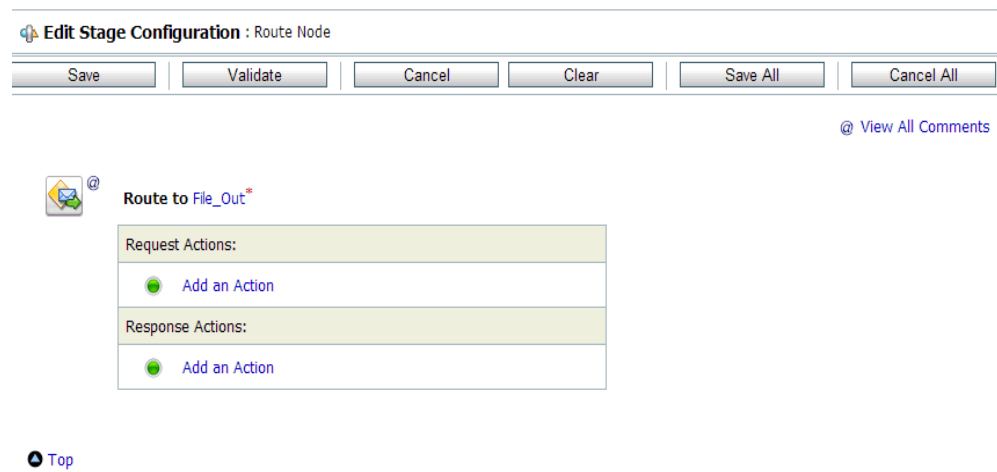
図 7-69 「サービスの選択」ダイアログ



「ステージ構成の編集」ワークスペース領域に戻ります。

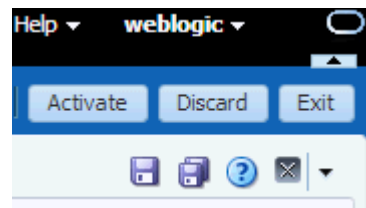
15. 図 7-70 に示すように、「すべて保存」をクリックします。

図 7-70 「ステージ構成の編集」ワークスペース領域



16. 図 7-71 に示すように、Oracle Service Bus セッションの右ペインで「アクティブ化」をクリックします。

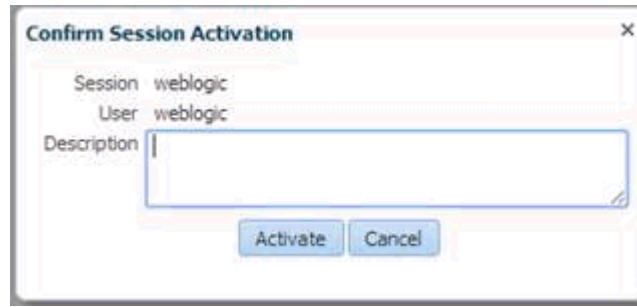
図 7-71 「アクティブ化」ボタン



「セッションのアクティブ化の確認」ウィンドウが表示されます。

17. 図 7-72 に示すように、「アクティブ化」をクリックして変更を保存します。

図 7-72 「セッションのアクティブ化の確認」 ウィンドウ



18. J.D. Edwards OneWorld システムからのイベントをトリガーし、出力が構成済の出力ロケーションで受信されたことを確認します。

イベントのトリガーの詳細は、4-48 ページの 4.5.5 項「J.D. Edwards OneWorld でイベントのトリガー」を参照してください。

7.4 sbconsole を使用したアウトバウンド・プロセスの構成 (BSE 構成)

この項では、BSE 構成で sbconsole を使用してアウトバウンド・プロセスを構成する方法について説明します。

このアウトバウンド・ユース・ケース・シナリオ用のサンプル・プロジェクトが、次に示す Application Adapter のインストール・フォルダに用意されています。

```
<ADAPTER_HOME>\etc\sample\JDEdwards_Samples.zip\JDEdwards_Samples\OSB\BSE\JDEdwards_Sample_BSE_OSB_Outbound_Project
```

この項では、次のトピックについて説明します。

- 7.4.1 項「Oracle Service Bus の起動およびプロジェクト・フォルダの作成」
- 7.4.2 項「Oracle Service Bus 統合のためのアプリケーション・エクスプローラのクラス・パスの設定」
- 7.4.3 項「アプリケーション・エクスプローラから Oracle Service Bus への WSDL の公開」
- 7.4.4 項「ファイル・タイプ・ビジネス・サービスの構成」
- 7.4.5 項「WSDL ベースのビジネス・サービスの構成」
- 7.4.6 項「パイプラインとプロキシ・サービスの構成」

7.4.1 Oracle Service Bus の起動およびプロジェクト・フォルダの作成

Oracle Service Bus の起動およびプロジェクト・フォルダの作成の詳細は、7.2.1 項「Oracle Service Bus の起動およびプロジェクト・フォルダの作成」(7-2 ページ)を参照してください。

7.4.2 Oracle Service Bus 統合のためのアプリケーション・エクスプローラのクラス・パスの設定

Oracle Service Bus 統合のためのアプリケーション・エクスプローラのクラス・パスの設定の詳細は、7.2.2 項「Oracle Service Bus 統合のためのアプリケーション・エクスプローラのクラス・パスの設定」(7-6 ページ)を参照してください。

7.4.3 アプリケーション・エクスプローラから Oracle Service Bus への WSDL の公開

この項では、アプリケーション・エクスプローラ (BSE 構成) から Oracle Service Bus に WSDL を公開する方法について説明します。

1. アプリケーション・エクスプローラを起動し、BSE 構成および J.D. Edwards OneWorld ターゲットに接続します。
2. 「サービス」、「CALLBSFN」、「アドレス帳」ビジネス・オブジェクトの順に展開します。
3. 「GetEffectiveAddress」を右クリックし、メニューから「Web サービスの作成」を選択します。

図 7-73 に示すように、「Web サービスの作成」ダイアログが表示されます。

図 7-73 「Web サービスの作成」ダイアログ

4. サービス名を入力し、「次へ」をクリックします。
5. 表示される次のダイアログで「OK」をクリックします。
アプリケーション・エクスプローラで、「ビジネス・サービス」ノードに表示が切り替わり、新規の Web サービスが左ペインに表示されます。
6. 新規 Web サービスを右クリックし、メニューから「WSDL のエクスポート」を選択します。

図 7-74 に示すように、「WSDL のエクスポート」ダイアログが表示されます。

図 7-74 「WSDL のエクスポート」ダイアログ

7. 「名前」フィールドに、WSDL ファイルのデフォルトのファイル名が表示されず。デフォルトを受け入れるか、任意の名前を指定します。
8. 「ロケーション」フィールドで、WSDL ドキュメントの公開先となる場所を入力します。

このロケーションは、Oracle Service Bus のプロジェクト名と、オプションで1つ以上のフォルダ名で構成されます。プロジェクト名およびフォルダ名はいずれもスラッシュ文字の「/」で区切る必要があります。
9. 「ホスト」フィールドで、Oracle Service Bus が稼働しているマシンの名前を入力します。
10. 「ポート」フィールドで、使用しているドメインのポートを入力します。
11. 「ユーザー」フィールドで、Oracle Service Bus へのアクセス用のユーザー名を入力します。
12. 「パスワード」フィールドで、Oracle Service Bus へのアクセス用のパスワードを入力します。
13. 「OK」をクリックします。

WSDL が「WSDL のエクスポート」ダイアログで指定されている場所に公開され、Oracle Service Bus でビジネス・サービスまたはプロキシ・サービスとともに使用できるようになります。

7.4.4 ファイル・タイプ・ビジネス・サービスの構成

ファイル・タイプ・ビジネス・サービスの構成の詳細は、[7.2.5 項「ファイル・タイプ・ビジネス・サービスの構成」](#) (7-9 ページ) を参照してください。

7.4.5 WSDL ベースのビジネス・サービスの構成

この項では、Oracle Service Bus コンソールを使用した WSDL タイプ・ビジネス・サービスの構成方法について説明します。

WSDL ベースのビジネス・サービスを構成するには、次のステップを実行します。

1. 左ペインで「Business Service」フォルダを右クリックし、「ビジネス・サービス」を選択します。

[図 7-75](#) に示すように、「ビジネス・サービスの作成」ウィンドウが表示されます。

図 7-75 「ビジネス・サービスの作成」ウィンドウ

Create Business Service

Create Type Transport

Create Service

* Resource Name BSE_Outbound_BS

Description

Service Definition

WSDL Based Service

Name

Path

Port/Binding

Transport http

Back Next Create Cancel

2. ビジネス・サービスの名前を入力し、「サービス定義」領域で「WSDL ベース・サービス」オプションを選択して、検索アイコンをクリックします。

図 7-76 に示すように、「検索と選択: WSDL リソース」ウィンドウが表示されます。

図 7-76 「検索と選択: WSDL リソース」ウィンドウ

Search and Select: WSDL Resource

Name

Path

Namespace

Search Reset

Name	Path	Namespace
BSE_Outbound_invoke	BSE_Outbound/...	urn:schemas-iwa...

OK Cancel

3. 「検索」ボタンをクリックし、BSE アウトバウンド WSDL を選択して、「OK」をクリックします。

「ビジネス・サービスの作成」ウィンドウに戻ります。

4. 「次へ」をクリックします。
5. 図 7-77 に示すように、デフォルト値を受け入れ、「作成」ボタンをクリックします。

図 7-77 「ビジネス・サービスの作成」ウィンドウ

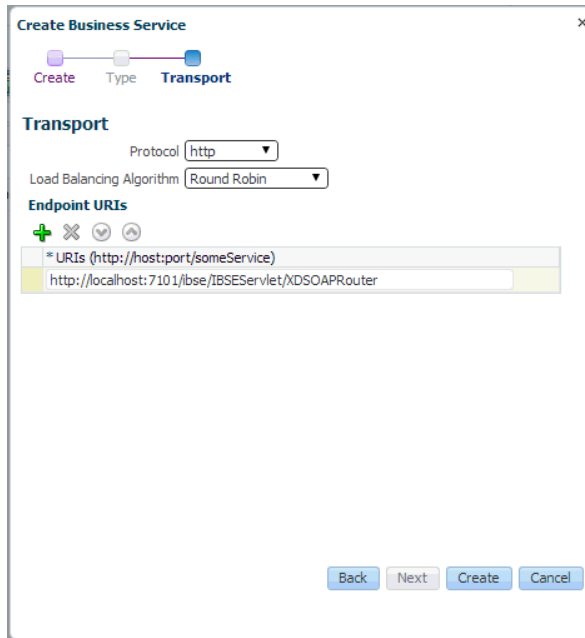
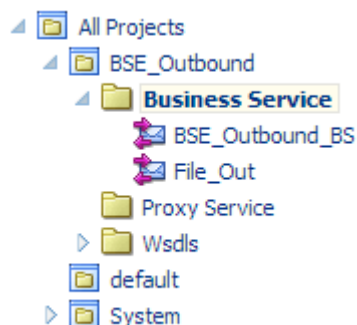


図 7-78 に示すように、作成された WSDL ベースのビジネス・サービスが「Business Service」フォルダの下に表示されます。

図 7-78 WSDL ベースのビジネス・サービス

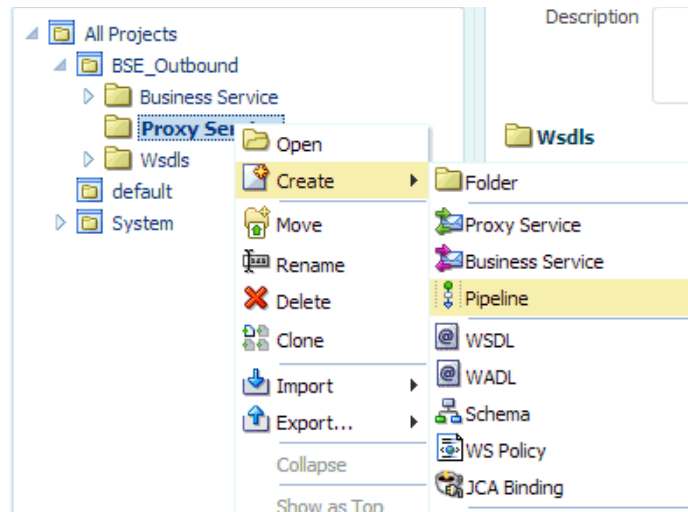


7.4.6 パイプラインとプロキシ・サービスの構成

この項では、Oracle Service Bus コンソールを使用したプロキシ・サービスの構成方法について説明します。

1. 図 7-79 に示すように、「Proxy Service」フォルダを右クリックし、「作成」を選択して、「パイプライン」をクリックします。

図 7-79 「パイプライン」オプション



「パイプライン」の作成ウィンドウが表示されます。

2. 「パイプライン名」フィールドに名前を入力します。デフォルトでは、「プロキシ・サービスとして公開」が選択されています。図 7-80 に示すように、プロキシ・サービス名を必要に応じて変更し、「トランスポート」をファイルに設定して、「作成」をクリックします。

図 7-80 「パイプラインの作成」ウィンドウ

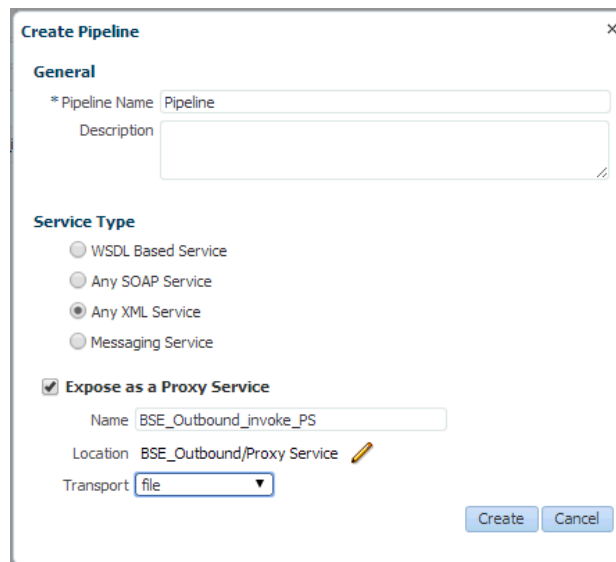
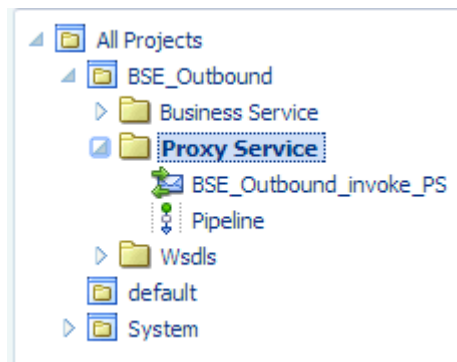


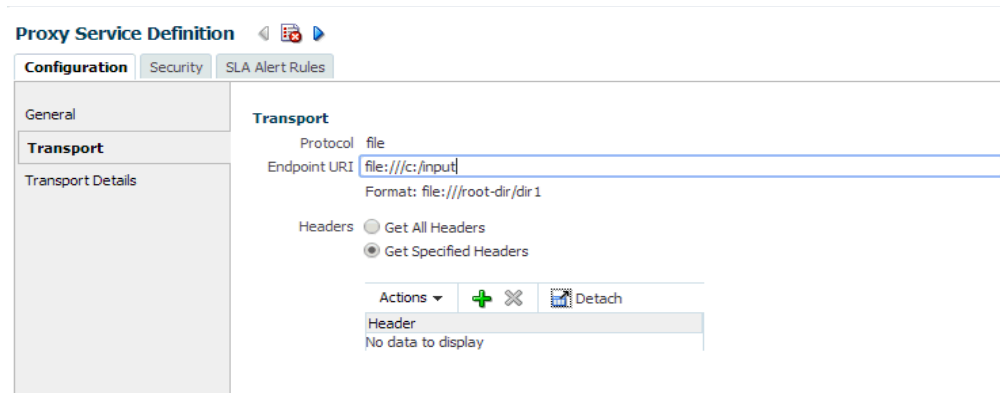
図 7-81 に示すように、作成されたパイプラインとプロキシ・サービスが「Proxy Service」の下に表示されます。

図 7-81 「Pipeline」ノード



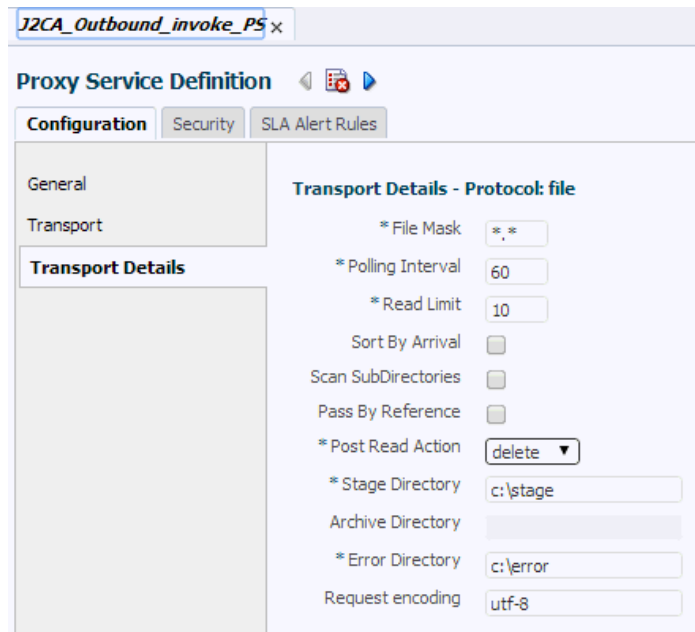
- 作成されたプロキシ・サービスをダブルクリックし、左ペインで「トランスポート」をクリックします。図 7-82 に示すように、「エンドポイント URI」フィールドに「入力場所」を指定します。

図 7-82 トランスポート



- 図 7-83 に示すように、左ペインで「トランスポートの詳細」をクリックし、「ステージ・ディレクトリ」フィールドと「エラー・ディレクトリ」フィールドに場所を指定します。

図 7-83 トランスポートの詳細



5. 図 7-84 に示すように、右隅の「すべて保存」アイコンをクリックします。

図 7-84 「すべて保存」アイコン



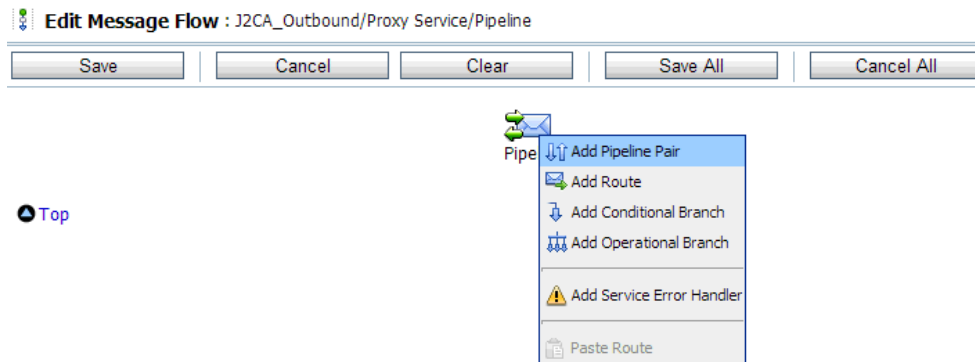
6. 図 7-85 に示すように、「Pipeline」ノードをダブルクリックし、右ペインで「メッセージ・フローを開く」アイコンをクリックして、メッセージ・フローを開きます。

図 7-85 「メッセージ・フローを開く」アイコン



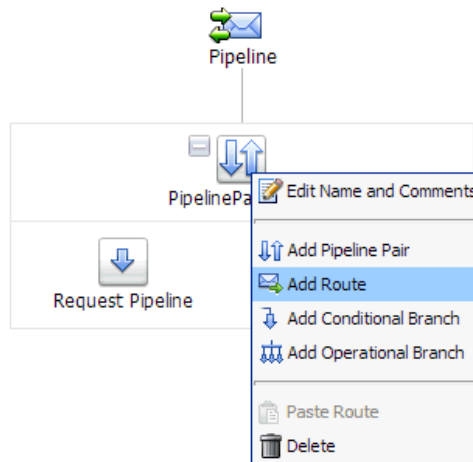
7. 図 7-86 に示すように、「プロキシ・サービス」アイコンをクリックし、メニューから「パイプライン・ペアの追加」を選択します。

図 7-86 「パイプライン・ペアの追加」オプション



8. 図 7-87 に示すように、**PipelinePairNode1** アイコンをクリックし、メニューから「ルートの追加」を選択します。

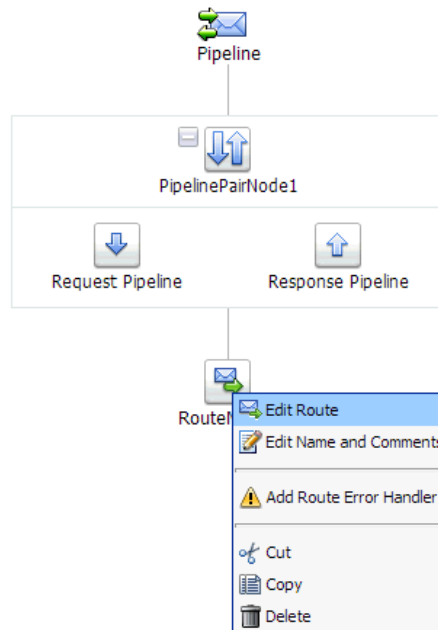
図 7-87 「ルートの追加」オプション



RouteNode1 アイコンが PipelinePairNode1 アイコンの下に追加されます。

9. 図 7-88 に示すように、RouteNode1 アイコンをクリックし、メニューから「ルートの編集」を選択します。

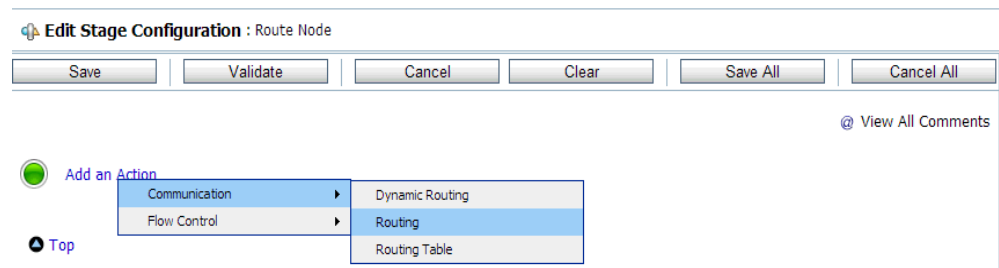
図 7-88 「ルートの編集」オプション



「ステージ構成の編集」ワークスペース領域が表示されます。

10. 図 7-89 に示すように、「アクションの追加」をクリックし、「通信」を選択して、「ルーティング」をクリックします。

図 7-89 「ステージ構成の編集」ワークスペース領域



11. 図 7-90 に示すように、<サービス>をクリックします。

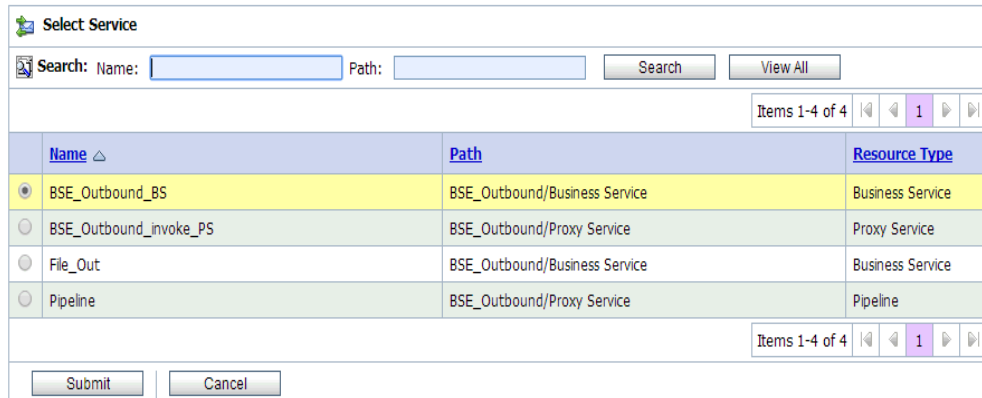
図 7-90 アクション



「サービスの選択」ダイアログが表示されます。

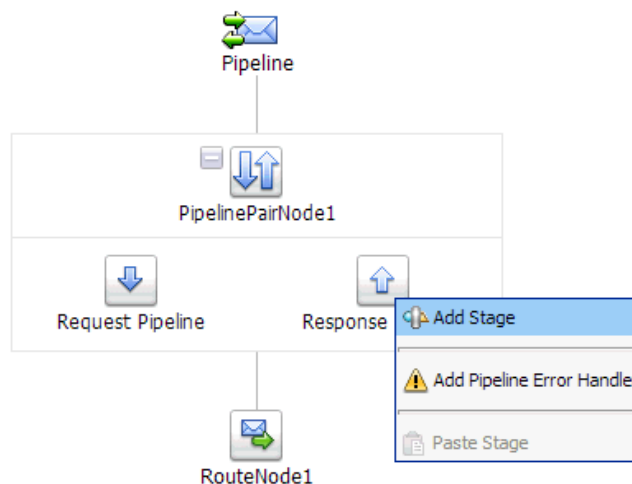
12. 図 7-91 に示すように、J.D. Edwards OneWorld 用に構成された WSDL タイプ・ビジネス・サービスを選択し、「発行」をクリックします。

図 7-91 「サービスの選択」ダイアログ



13. リストから操作属性として J.D. Edwards OneWorld ビジネス・オブジェクトの名前を選択し、「保存」をクリックします。
14. 図 7-92 に示すように、「レスポンス・パイプライン」アイコンをクリックし、メニューから「ステージの追加」を選択します。

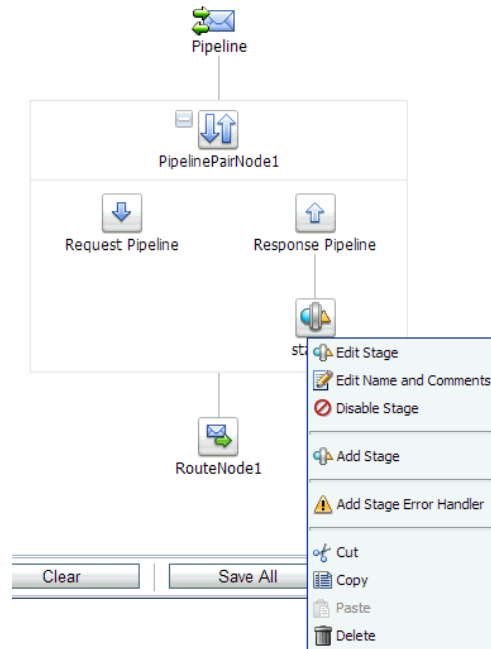
図 7-92 「レスポンス・パイプライン」アイコン



「ステージ 1」アイコンが「レスポンス・パイプライン」アイコンの下に追加されます。

15. 図 7-93 に示すように、「ステージ 1」アイコンをクリックし、メニューから「ステージの編集」を選択します。

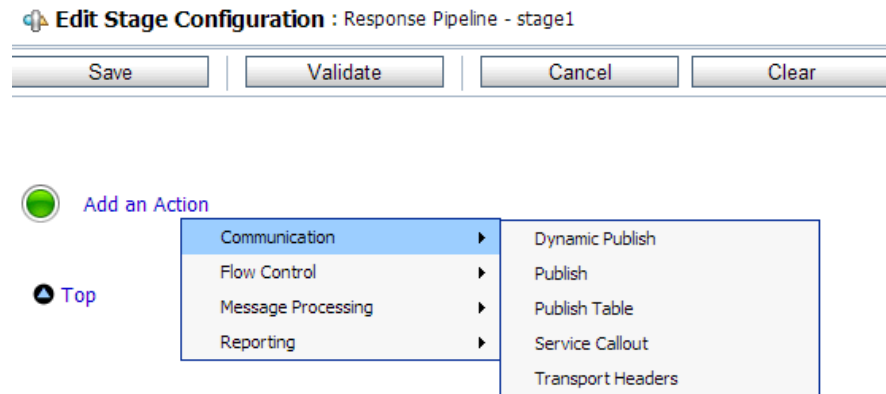
図 7-93 「ステージの編集」オプション



「ステージ構成の編集」ワークスペース領域が表示されます。

16. 図 7-94 に示すように、「アクションの追加」をクリックし、「通信」を選択して、「パブリッシュ」をクリックします。

図 7-94 「ステージ構成の編集」ワークスペース領域



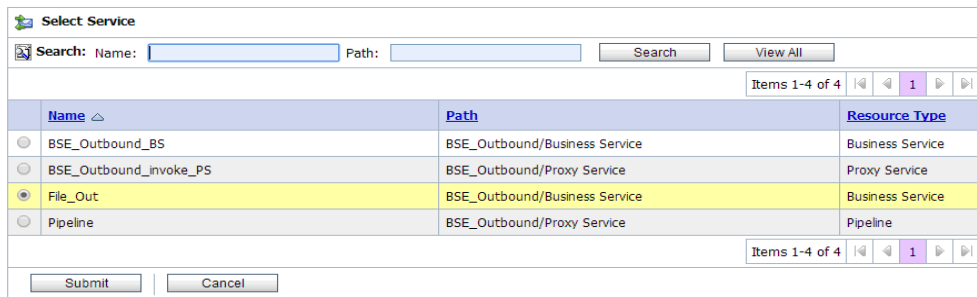
17. 図 7-95 に示すように、<サービス>をクリックします。

図 7-95 <サービス> アクション



18. 図 7-96 に示すように、「サービスの選択」ダイアログで、ファイル・タイプ・ビジネス・サービスを選択し、「発行」をクリックします。

図 7-96 「サービスの選択」ダイアログ



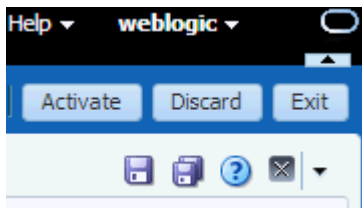
19. 図 7-97 に示すように、「すべて保存」をクリックします。

図 7-97 「すべて保存」ボタン



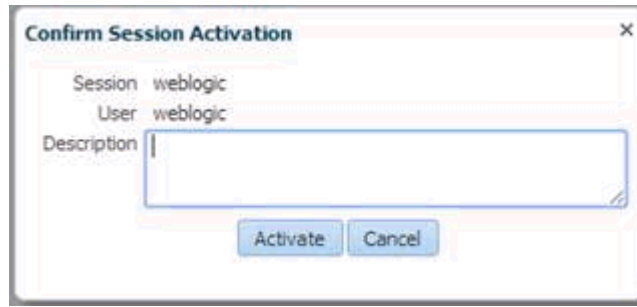
20. 図 7-98 に示すように、Oracle Service Bus セッションの右ペインで「アクティブ化」をクリックします。

図 7-98 「アクティブ化」ボタン



21. 図 7-99 に示すように、「アクティブ化」をクリックして変更を保存します。

図 7-99 セッションのアクティブ化の確認



22. 入力 XML ファイルをコピーして構成済の入力フォルダ (「C:\input」など) に貼り付けます。

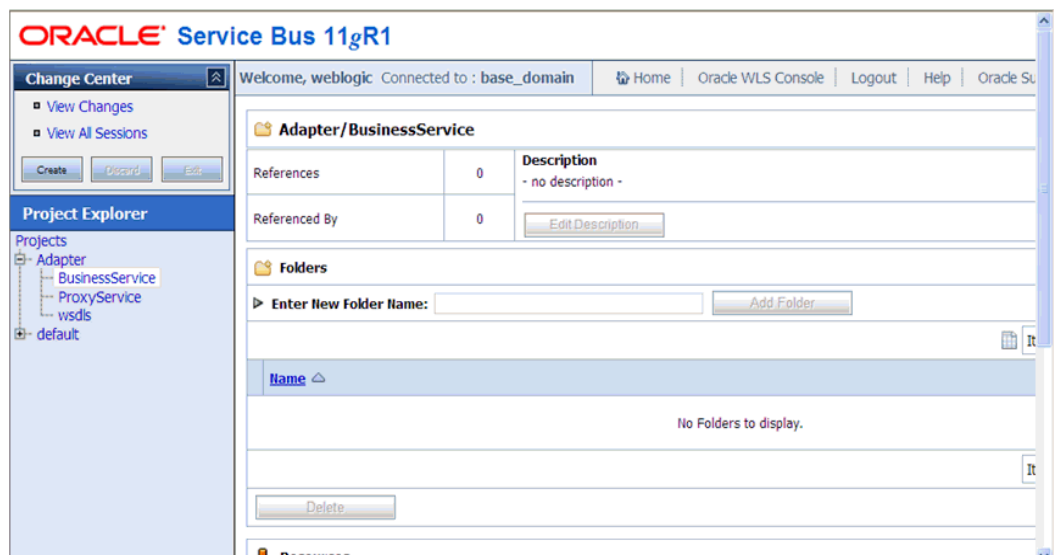
構成済の出力場所 (「C:\output」など) で出力が受信されたことを確認します。

7.5 Oracle Service Bus を使用した JMS プロキシ・サービスの構成 (J2CA 構成)

この項では、J2CA 構成で Oracle Service Bus を使用して JMS プロキシ・サービスを構成する方法について説明します。

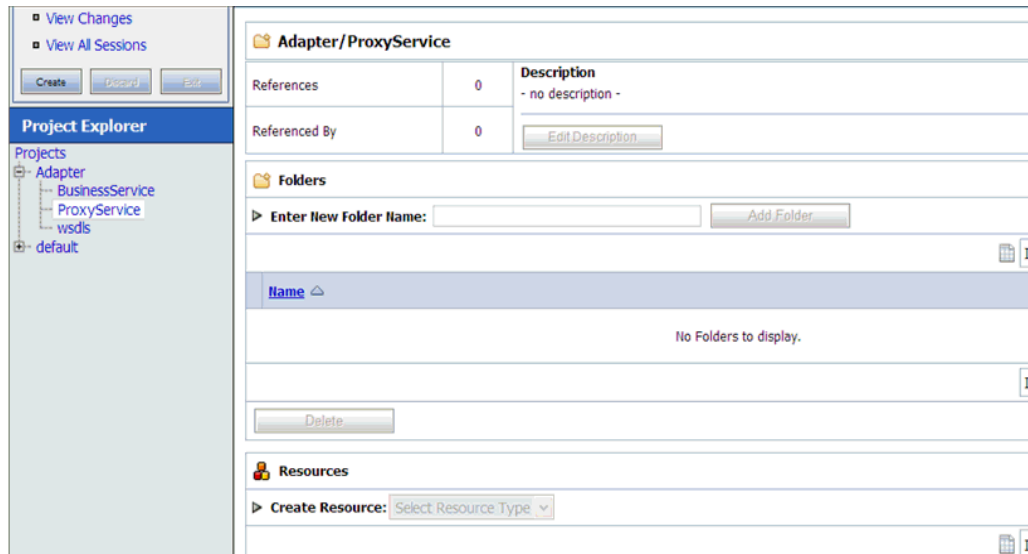
1. Oracle Service Bus を起動し、必要なプロジェクト・フォルダを作成します。
詳細は、7-2 ページの 7.2.1 項「Oracle Service Bus の起動およびプロジェクト・フォルダの作成」を参照してください。
2. WSDL を生成してアプリケーション・エクスプローラから作成済のプロジェクト・フォルダに公開します。公開済の WSDL を使用して、ビジネス・サービスを作成します。
詳細は、7-6 ページの 7.2.3 項「アプリケーション・エクスプローラから Oracle Service Bus への WSDL の公開」を参照してください。
3. 図 7-100 に示すように、Service Bus コンソール・ページが開きます。

図 7-100 Service Bus コンソール



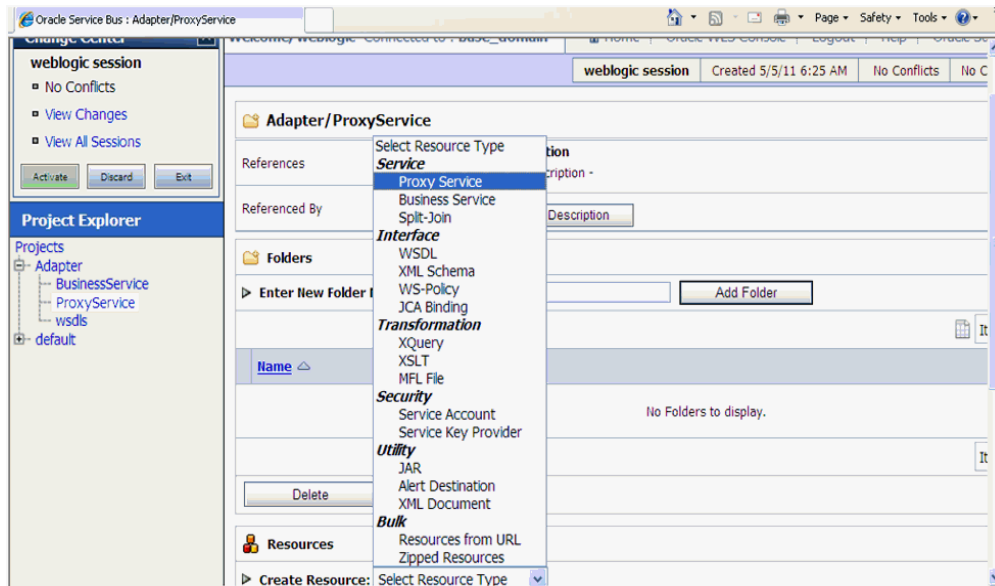
4. 図 7-101 に示すように、左ペインで「ProxyService」プロジェクト・フォルダを選択し、「作成」をクリックします。

図 7-101 プロキシ・サービス



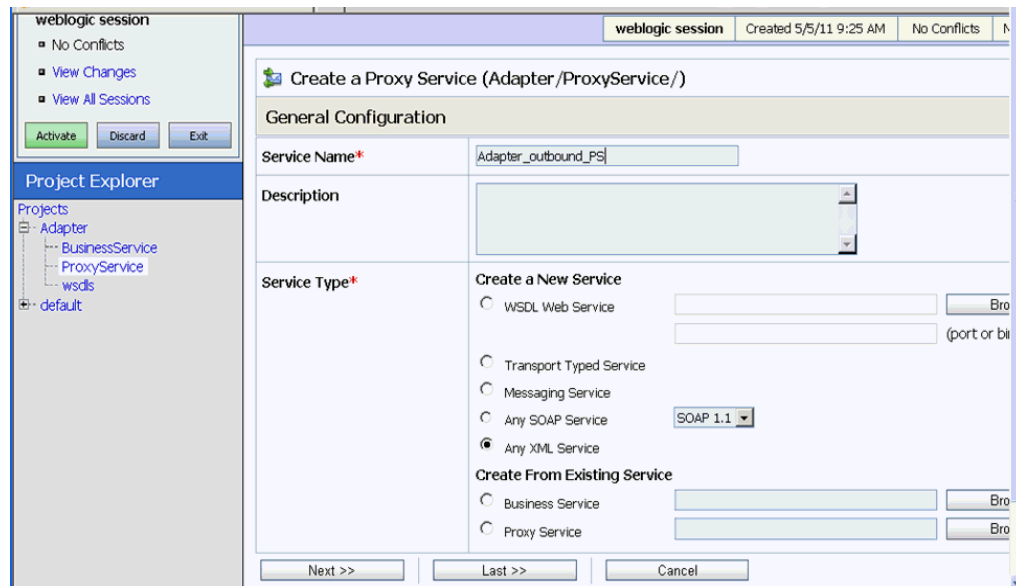
5. 図 7-102 に示すように、右ペインで「リソースの作成」リストから「プロキシ・サービス」を選択します。

図 7-102 「リソースの作成」メニュー



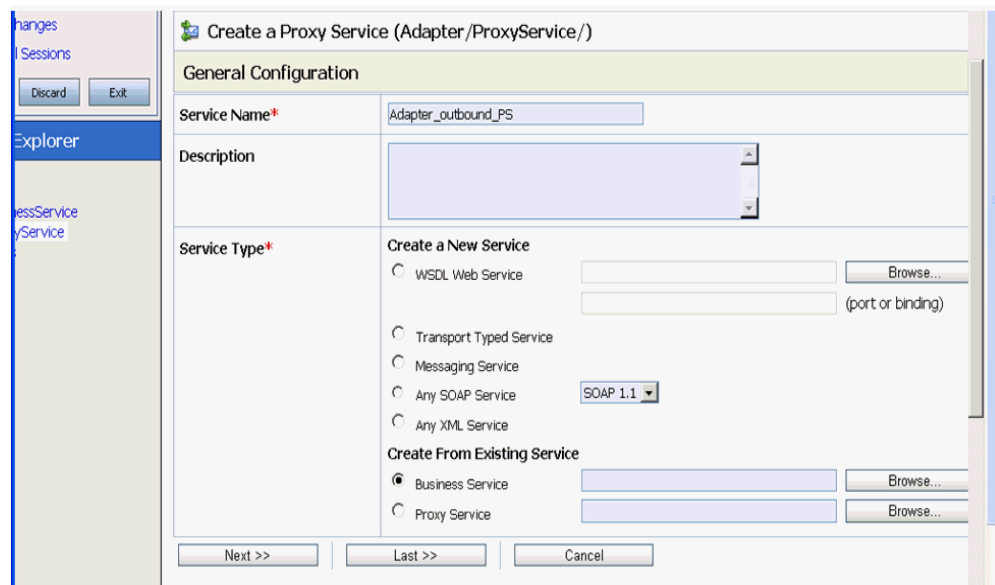
6. 図 7-103 に示すように、「サービス名」フィールドに適切な名前を入力します。

図 7-103 サービス名



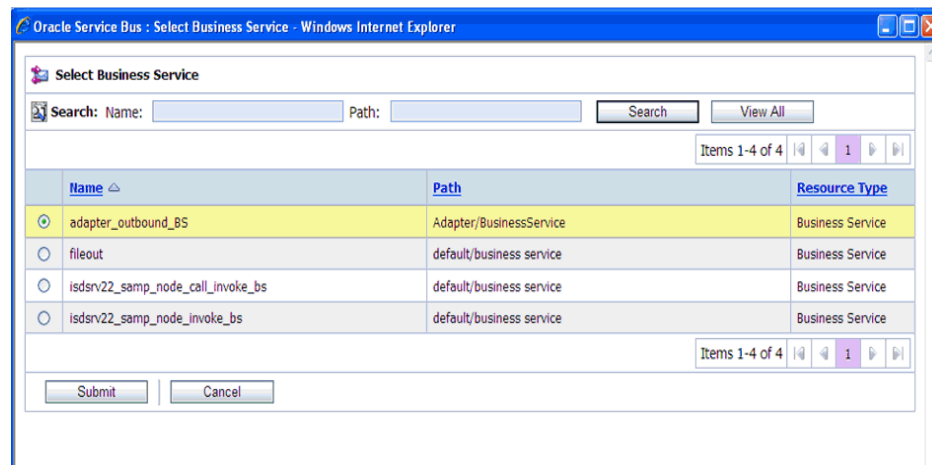
7. 図 7-104 に示すように、「サービス・タイプ」セクションで、「既存のサービスから作成」の下にある「ビジネス・サービス」ラジオ・ボタンを選択し、「参照」をクリックします。

図 7-104 ビジネス・サービス



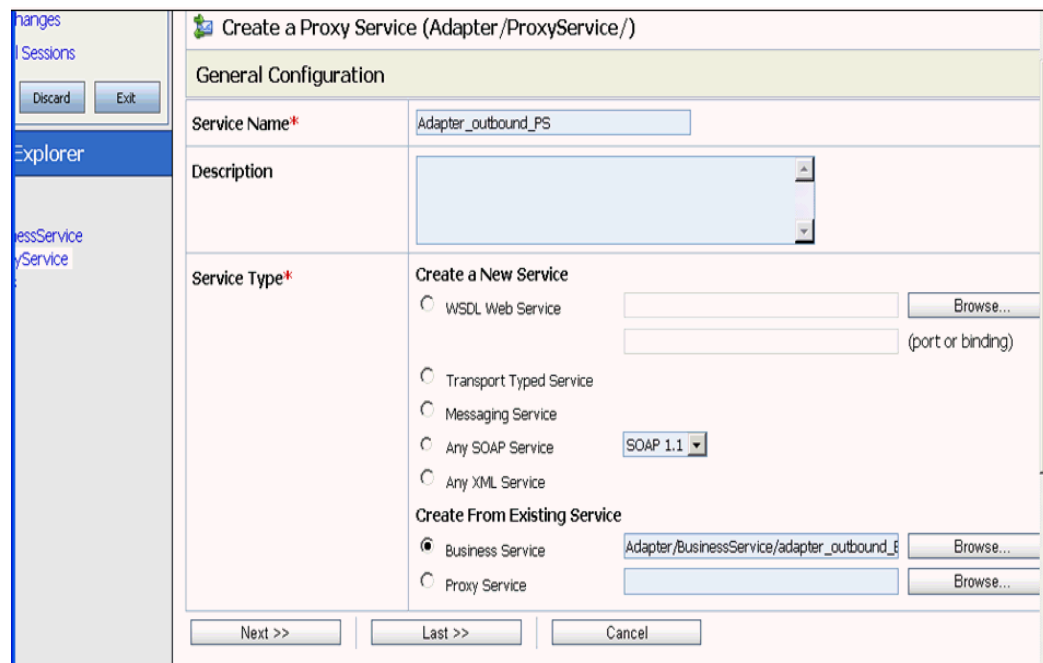
8. 図 7-105 に示すように、既存のビジネス・サービスを選択し、「発行」をクリックします。

図 7-105 既存のビジネス・サービス



9. 図 7-106 に示すように、「次へ」をクリックします。

図 7-106 次



10. 図 7-107 に示すように、「プロトコル」リストから「jms」を選択し、「次へ」をクリックします。

図 7-107 「プロトコル」 リスト

View Changes
View All Sessions
Activate Discard Exit

Project Explorer
jects
Adapter
BusinessService
ProxyService
wsdis
default

Edit a Proxy Service (Adapter/ProxyService/Adapter_outbound_PS)

Transport Configuration

Protocol* jms

Endpoint URI* Format: jms://((host:port),((host:port)*))((host:port?)/Factory/IndName/QueueIn
jms://localhost:8001/weblogic.jms.XAConnectionFactory/Adapter_outbound_PSRequest

Get All Headers
 Yes
 No
Header Add
HEADER ACTION
There are no headers configured.

<< Prev. Next >> Last >> Cancel

11. 図 7-108 に示すように、次のパラメータを指定します。
- 「宛先タイプ」セクションで、「キュー」を選択します。
 - 「レスポンスが必要」チェック・ボックスを選択します。
 - 「レスポンス・メッセージ・タイプ」セクションで、「テキスト」を選択します。
 - 「レスポンス URI」フィールドに、「トランスポート構成」で使用したエンドポイント URI を入力し、「Request」を「Response」に変更します。

次に例を示します。

```
jms://localhost:8001/weblogic.jms.XAConnectionFactory/Adapter_outbound_PSResponse
```

図 7-108 プロキシ・サービスの編集

Edit a Proxy Service (Adapter/ProxyService/Adapter_outbound_PS)

JMS Transport Configuration

Destination Type
 Queue
 Topic

Is Response Required

Response Pattern
 JMSCorrelationID
 JMSMessageID

Response Message Type
 Bytes
 Text

Dispatch Policy
default

Request Encoding
UTF-8

Response Encoding
UTF-8

Client Response Timeout
300

Response URI
tionFactory/Adapter_outbound_PSResponse

JMS Service Account
 Browse...

Advanced Settings
<< Prev. Next >> Last >> Cancel

12. 「次へ」をクリックします。

図 7-109 に示すように、「操作選択構成」 ペインが表示されます。

図 7-109 「操作選択構成」 ペイン

13. 「SOAP 本体タイプ」が選択されていることを確認し、「次へ」をクリックします。

14. 図 7-110 に示すように、「トランザクションが必要」ボックスを有効にし、「次へ」をクリックします。

図 7-110 メッセージ処理

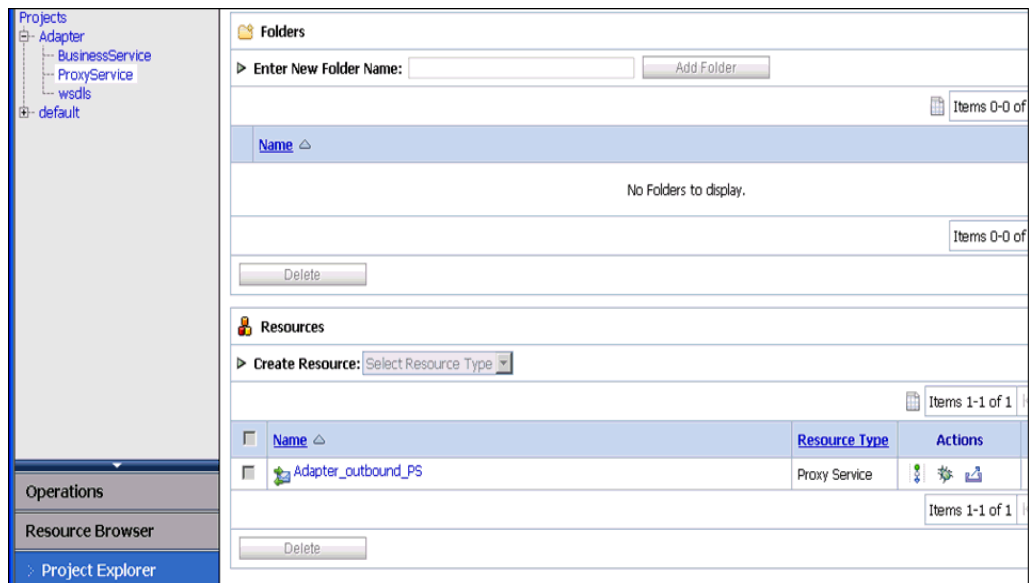
15. 図 7-111 に示すように、「保存」をクリックします。

図 7-111 保存

Use SSL	Disabled
Is Response Required	Enabled
Request Encoding	UTF-8
Response Encoding	UTF-8
Response Pattern	JMSCorrelationID
JNDI Timeout	0
Response URI	jms://localhost:8001/weblogic.jms.XAConnectionFactory/Adapter_outbound_BSResponse
Response Message Type	Text
Client Response Timeout	300
Is XA Required	False
Operation Selection Configuration	
Selection Algorithm	SOAP Body Type
Message Handling Configuration	
Transaction Required	Enabled
Same Transaction For Response	Disabled
Content Streaming	Disabled
<input data-bbox="462 760 602 783" type="button" value=" << Prev. "/> <input data-bbox="651 760 790 783" type="button" value=" Save "/> <input data-bbox="839 760 979 783" type="button" value=" Cancel "/>	

図 7-112 に示すように、作成したプロキシ・サービスが保存されます。

図 7-112 プロキシ・サービス

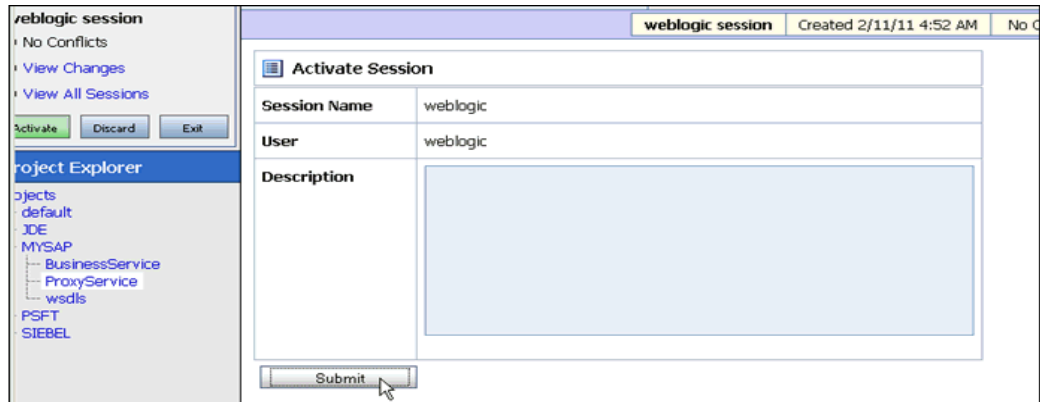


The screenshot shows the Oracle Service Bus console interface. On the left, the 'Project Explorer' pane shows a tree structure under 'Adapter' including 'BusinessService', 'ProxyService', 'wsdl', and 'default'. The main workspace is divided into 'Folders' and 'Resources' sections. The 'Resources' section is active, showing a 'Create Resource' dropdown set to 'Select Resource Type'. Below this is a table with the following data:

Name	Resource Type	Actions
Adapter_outbound_PS	Proxy Service	[Refresh] [Refresh] [Refresh]

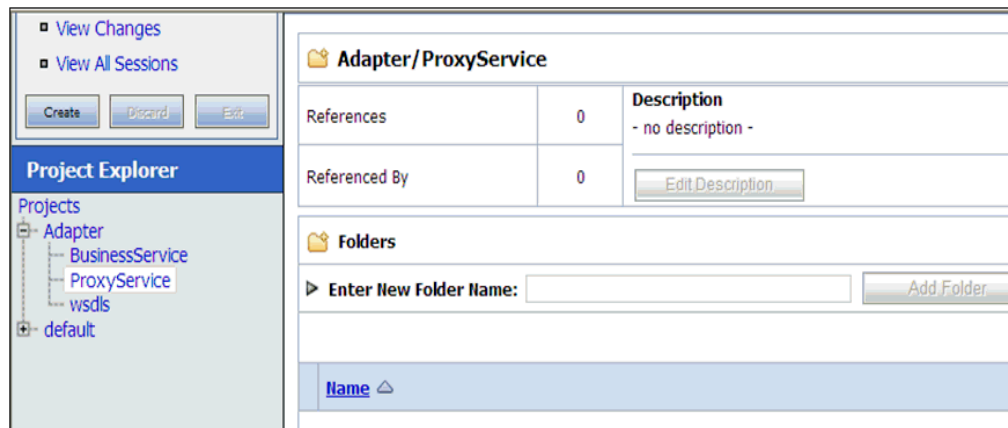
16. 図 7-113 に示すように、左ペインで「アクティブ化」をクリックし、「発行」をクリックします。

図 7-113 セッションのアクティブ化



17. 図 7-114 に示すように、左ペインで、「プロジェクト」フォルダの下にある「ProxyService」をクリックします。

図 7-114 アダプタ /ProxyService



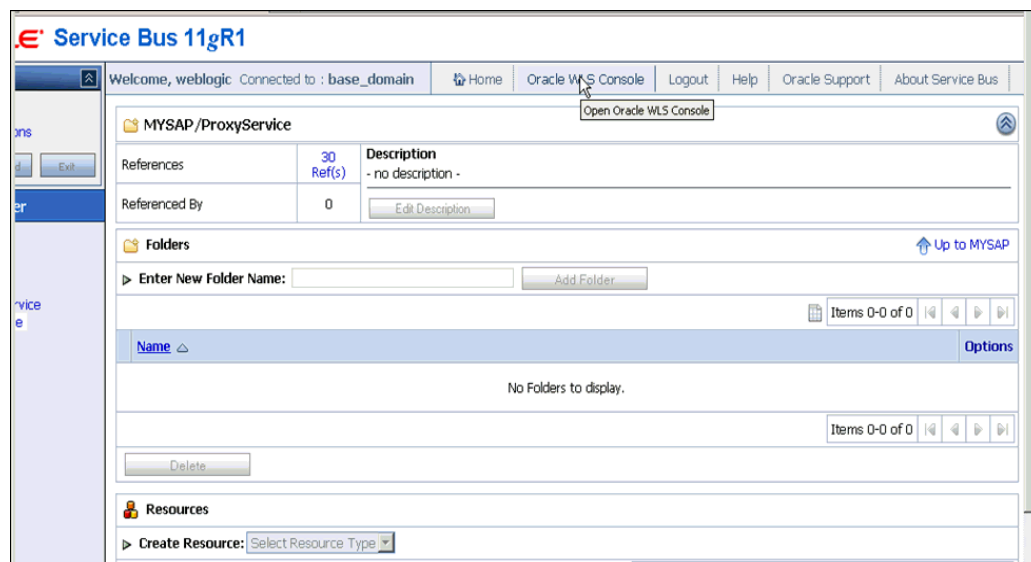
18. 図 7-115 に示すように、作成されたプロキシ・サービスに対する「テスト・コンソールの起動」アイコンをクリックします。

図 7-115 「テスト・コンソールの起動」アイコン



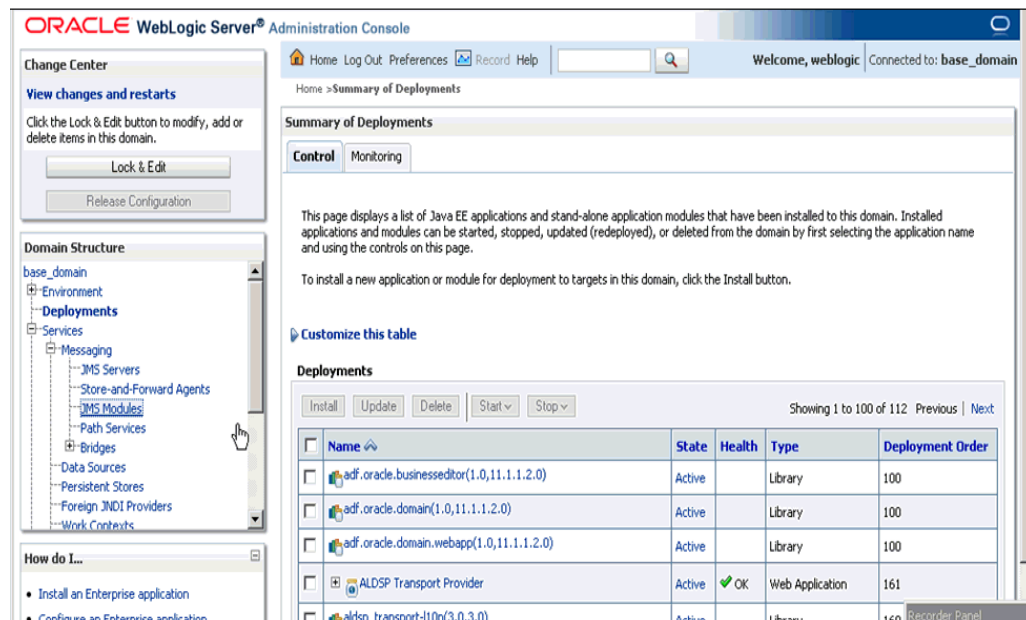
19. 「ペイロード」の入力値を指定し、「直接呼出し」ボックスを選択解除して、「実行」をクリックします。
20. レスポンス・ドキュメントをレビューし、「閉じる」をクリックします。
21. 図 7-116 に示すように、「Oracle WLS コンソール」タブをクリックします。

図 7-116 ProxyService



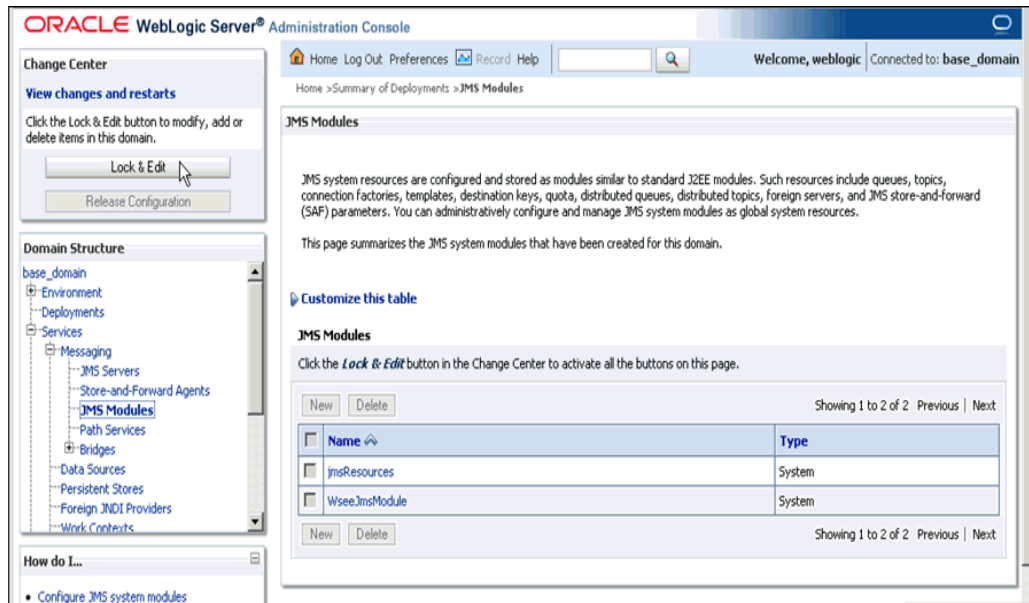
22. 図 7-117 に示すように、Oracle WLS コンソールで、「サービス」を展開し、「メッセージング」を展開して、「JMS モジュール」をクリックします。

図 7-117 Oracle WLS コンソール



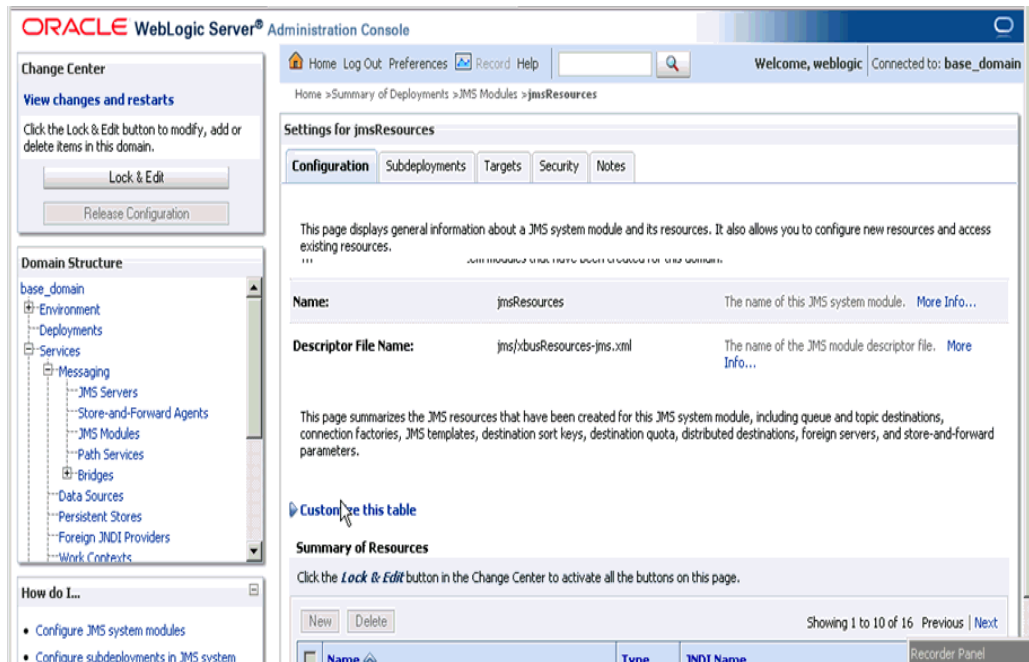
23. 図 7-118 に示すように、「jmsResources」をクリックします。

図 7-118 JMS モジュール



24. 図 7-119 に示すように、「ロックして編集」をクリックします。

図 7-119 構成設定



25. 図 7-120 に示すように、適切なリクエスト・リンク (「Adapter_outbound_PSRequest」など) をクリックします。

図 7-120 Adapter_outbound_PSRequest

The screenshot shows the Oracle WebLogic Administration Console interface. On the left, there are panels for 'How do I...' (with links to configure JMS system modules, subdeployments, and resources) and 'System Status' (showing health of running servers: Failed (0), Critical (0), Overloaded (0), Warning (0), OK (2)). The main area displays a 'Summary of Resources' table for the resource 'Adapter_outbound_PSRequest-1531625329'.

Name	Type	JNDI Name
Adapter_outbound_PSRequest-1531625329	Queue	Adapter_outbound_PSRequest
Adapter_outbound_PSResponse2015668417	Queue	Adapter_outbound_PSResponse
QueueIn	Queue	QueueIn
QueueIn.Quota	Quota	N/A
TemporaryTplmt	Template	N/A
weblogic.wsb.jms.transporttask.QueueConnectionFactory	Connection Factory	weblogic.wsb.jms.transporttask.QueueG
wli.reporting.jmsprovider.ConnectionFactory	Connection Factory	wli.reporting.jmsprovider.ConnectionFact
wli.reporting.jmsprovider.queue	Queue	wli.reporting.jmsprovider.queue
wli.reporting.jmsprovider_error.queue	Queue	wli.reporting.jmsprovider_error.queue
wli.reporting.purge.queue	Queue	wli.reporting.purge.queue

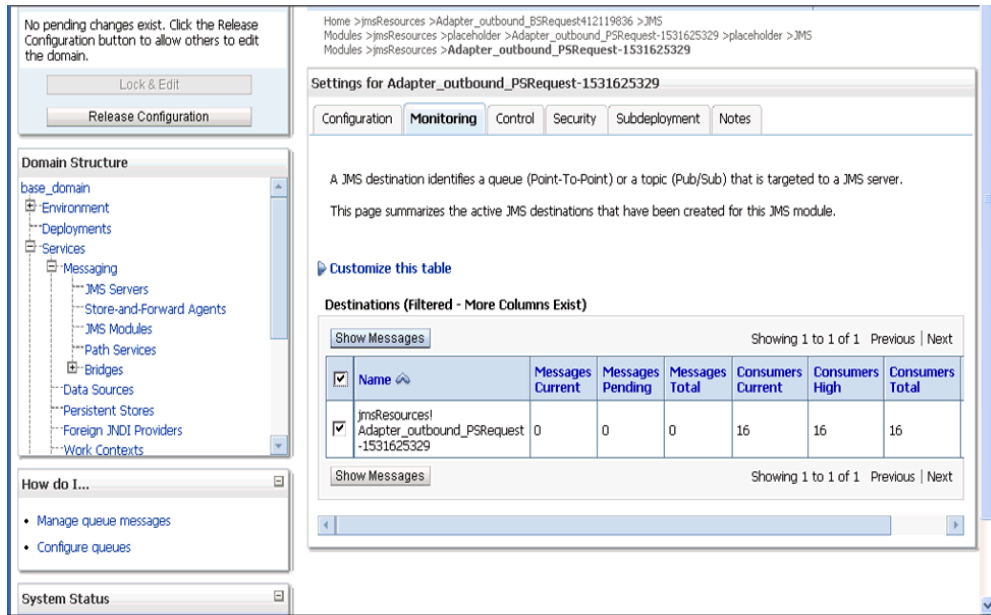
26. 図 7-121 に示すように、「モニタリング」タブをクリックします。

図 7-121 「モニタリング」タブ

The screenshot shows the Oracle WebLogic Administration Console interface. The 'Monitoring' tab is selected for the resource 'Adapter_outbound_PSRequest-1531625329'. The 'General' sub-tab is active, showing configuration parameters for the queue. The 'Name' is 'Adapter_outbound_PSRequest-1531625329', the 'JNDI Name' is 'Adapter_outbound_PSRequest', and the 'Template' is 'None'. A 'Save' button is visible at the top of the configuration area.

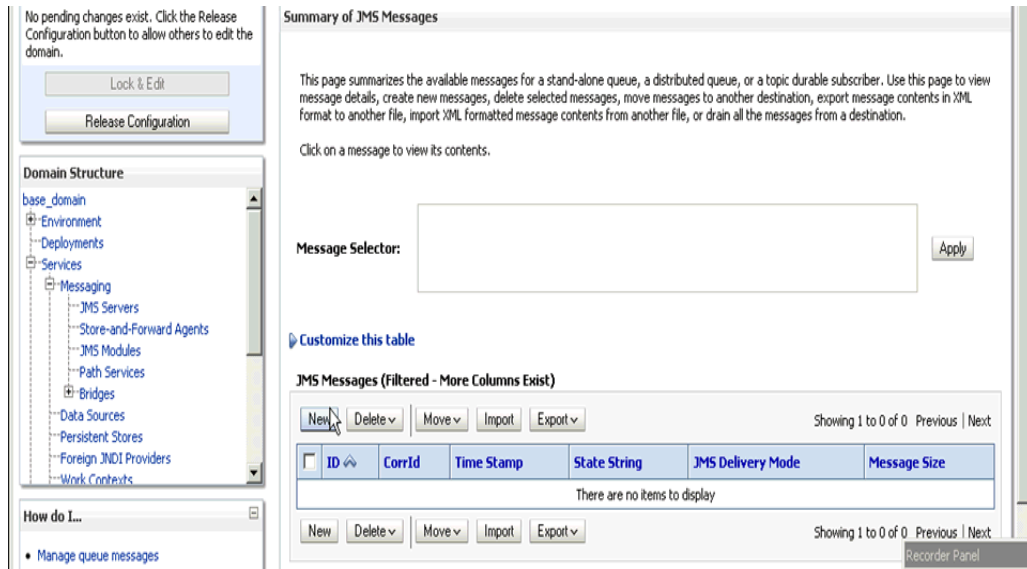
27. 図 7-122 に示すように、チェック・ボックスを選択し、「メッセージの表示」をクリックします。

図 7-122 アダプタ設定



28. 図 7-123 に示すように、「新規」をクリックします。

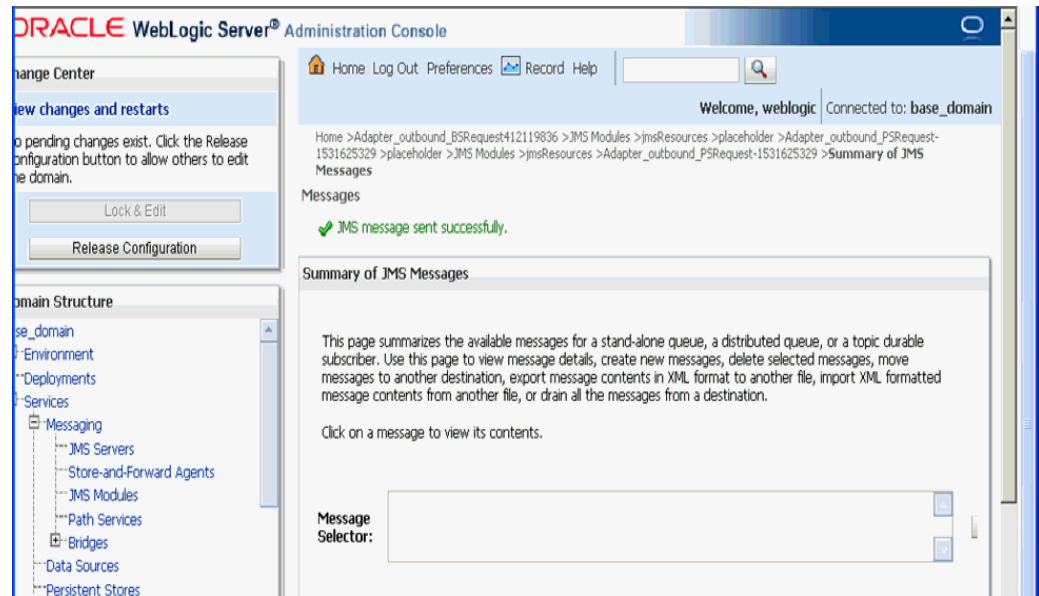
図 7-123 JMS メッセージ



29. 「本文」フィールドに入力ペイロードを入力し、「OK」をクリックします。

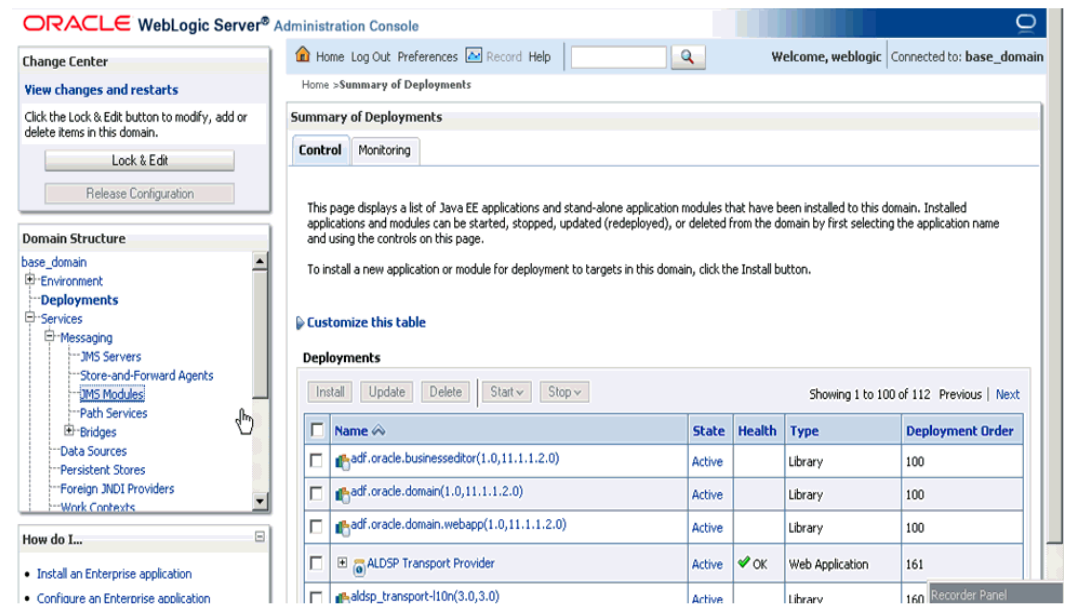
図 7-124 に示すように、成功メッセージが表示されます。

図 7-124 JMS の成功メッセージ



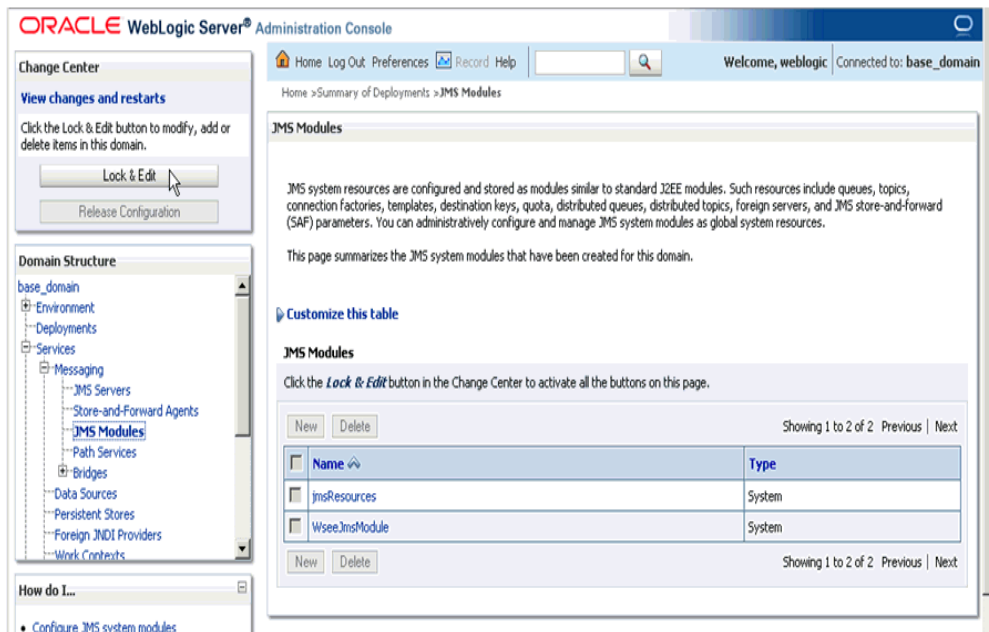
30. 図 7-125 に示すように、Oracle WLS コンソールで、「サービス」を展開し、「メッセージング」を展開して、「JMS モジュール」をクリックします。

図 7-125 JMS モジュール



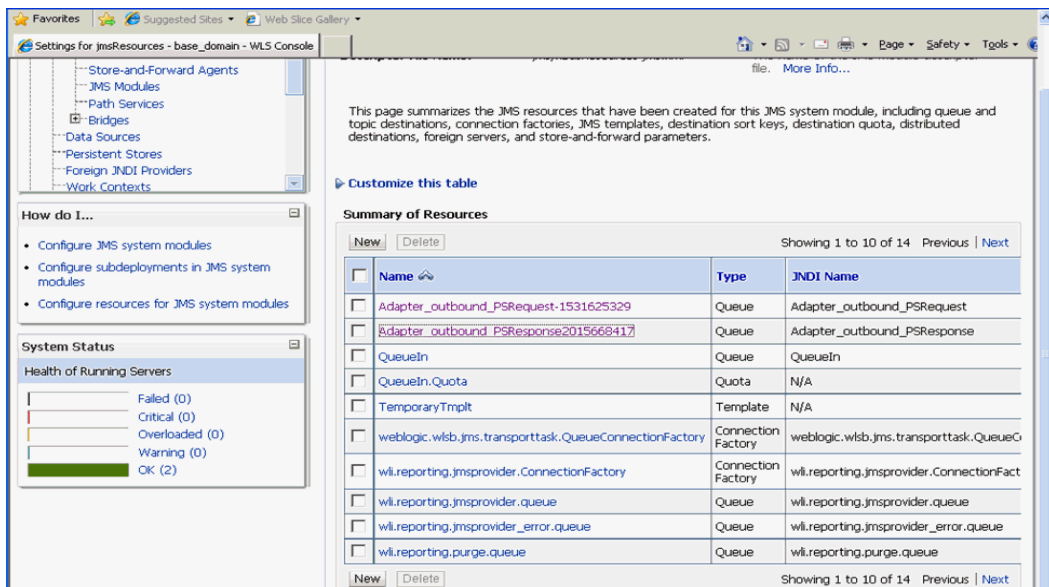
31. 図 7-126 に示すように、「jmsResources」をクリックします。

図 7-126 jmsResources



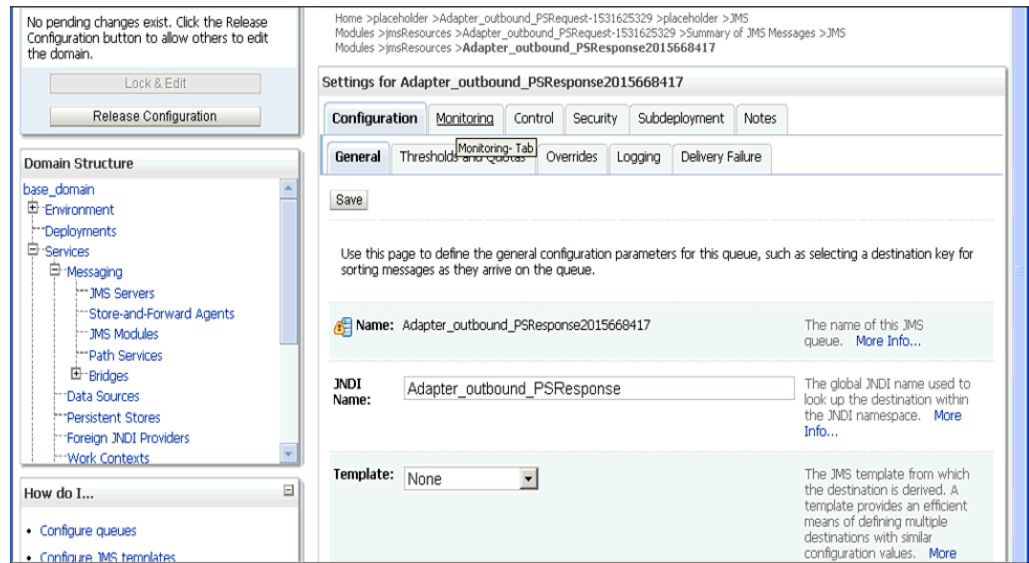
32. 図 7-127 に示すように、適切なレスポンス・リンク (「Adapter_outbound_PSResponse」など) をクリックします。

図 7-127 リソースのサマリー



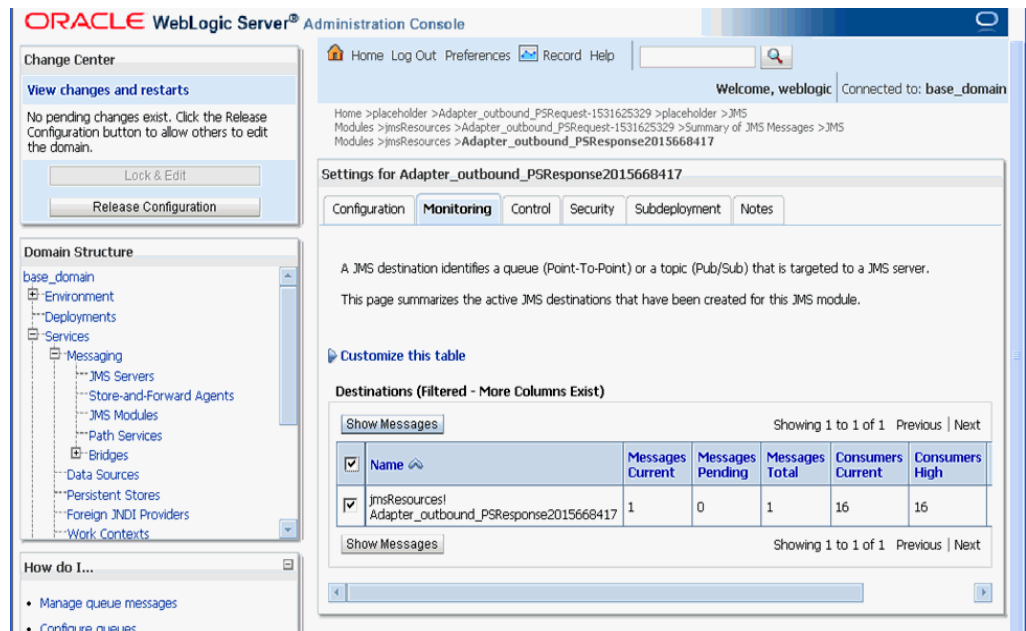
33. 図 7-128 に示すように、「モニタリング」タブをクリックします。

図 7-128 「モニタリング」タブ



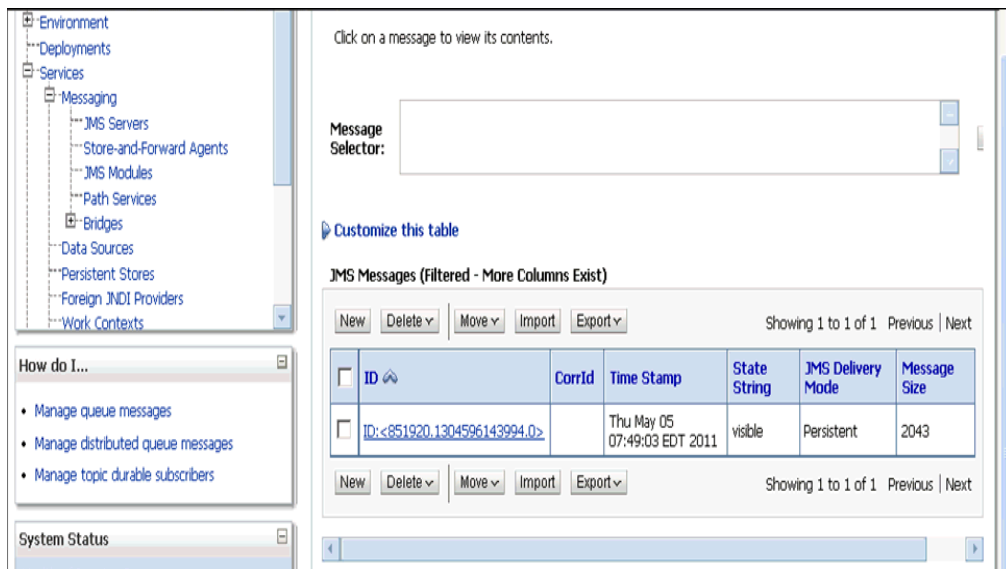
34. 図 7-129 に示すように、チェック・ボックスを選択し、「メッセージの表示」をクリックします。

図 7-129 宛先のメッセージ



35. 図 7-130 に示すように、「ID」リンクをクリックします。

図 7-130 JMS メッセージ



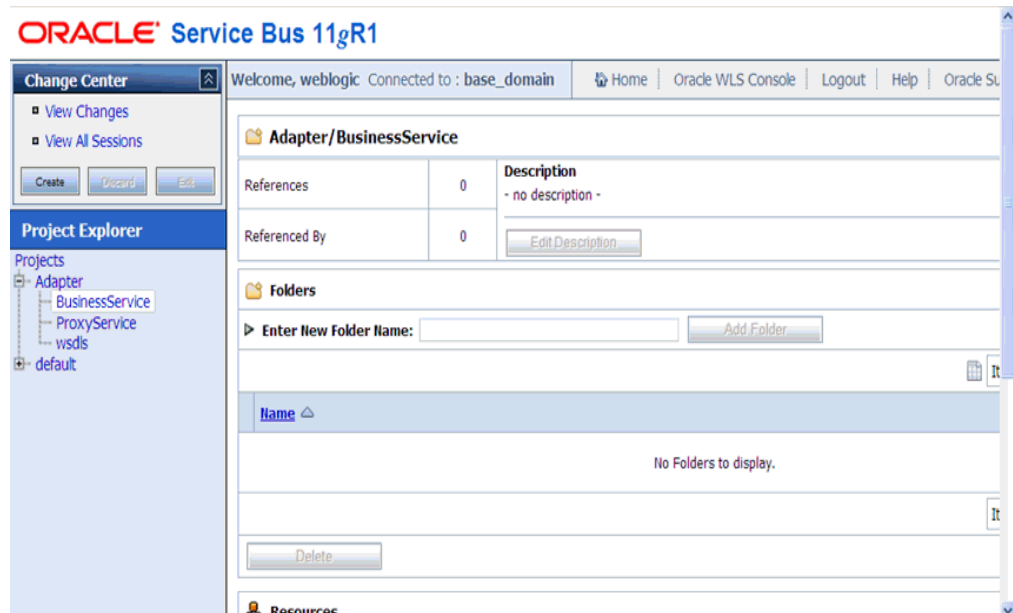
レスポンス・ドキュメントが「テキスト」フィールドの下に表示されます。

7.6 Oracle Service Bus を使用した HTTP プロキシ・サービスの構成 (J2CA 構成)

この項では、J2CA 構成で Oracle Service Bus を使用して HTTP プロキシ・サービスを構成する方法について説明します。

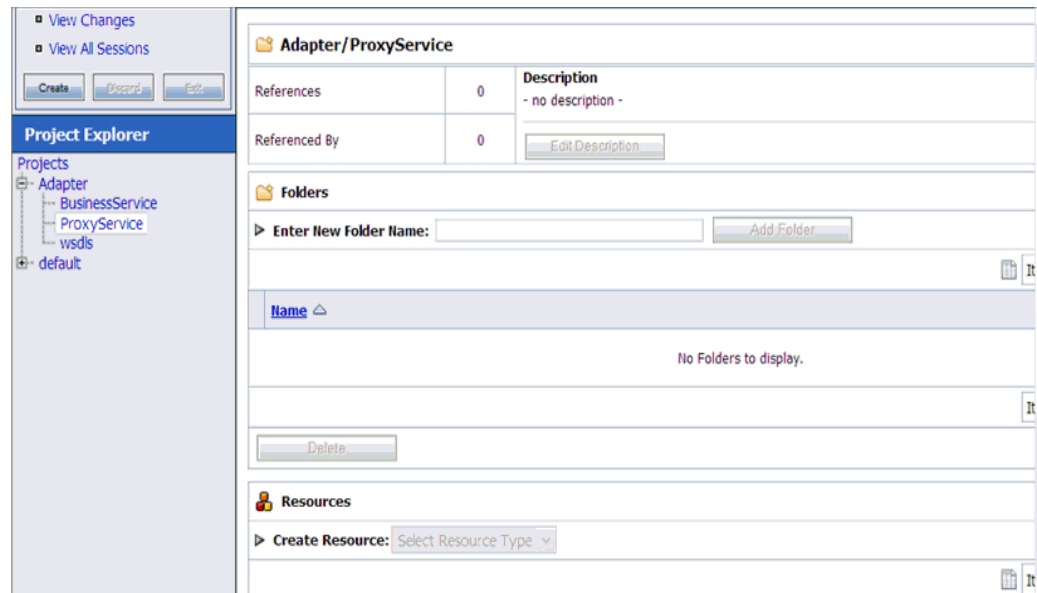
1. Oracle Service Bus を起動し、必要なプロジェクト・フォルダを作成します。
詳細は、7-2 ページの [7.2.1 項「Oracle Service Bus の起動およびプロジェクト・フォルダの作成」](#) を参照してください。
2. WSDL を生成してアプリケーション・エクスプローラから作成済のプロジェクト・フォルダに公開し、公開済の WSDL を使用してビジネス・サービスを作成します。
詳細は、7-6 ページの [7.2.3 項「アプリケーション・エクスプローラから Oracle Service Bus への WSDL の公開」](#) を参照してください。
3. [図 7-131](#) に示すように、Service Bus コンソール・ページが開きます。

図 7-131 Service Bus コンソール・ページ



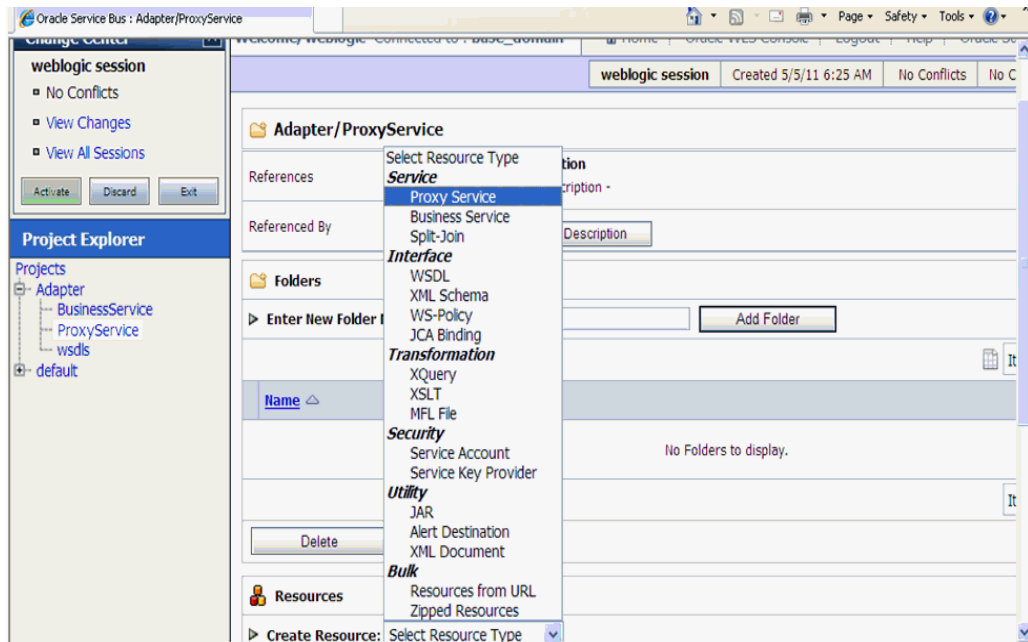
4. 図 7-132 に示すように、「プロジェクト・エクスプローラ」で「**ProxyService**」プロジェクト・フォルダを選択し、「作成」をクリックします。

図 7-132 プロジェクト・エクスプローラ



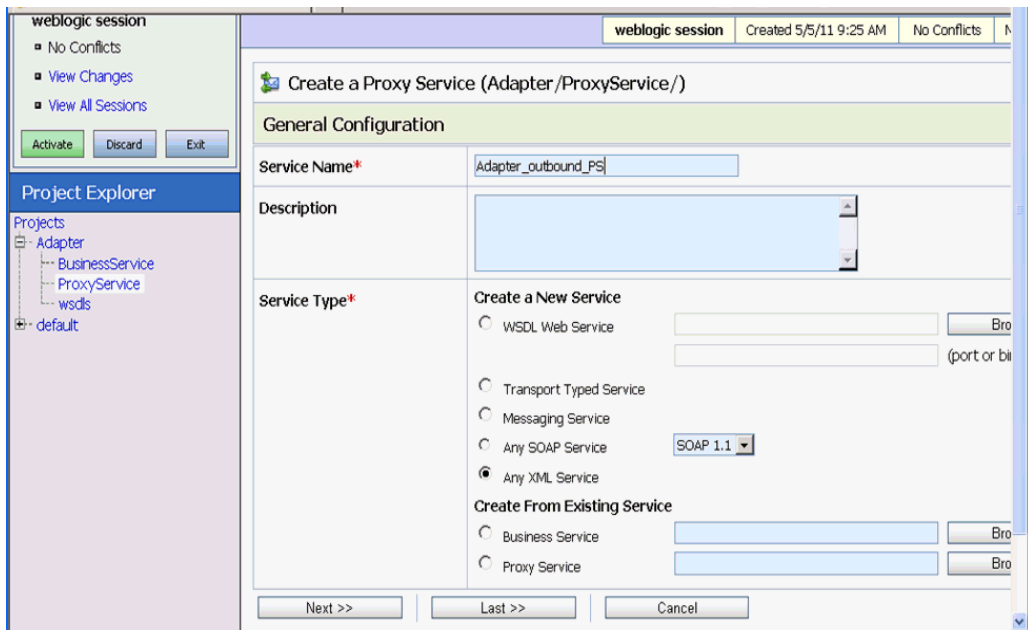
5. 図 7-133 に示すように、右ペインの「リソースの作成」リストで「**プロキシ・サービス**」を選択します。

図 7-133 プロキシ・サービス



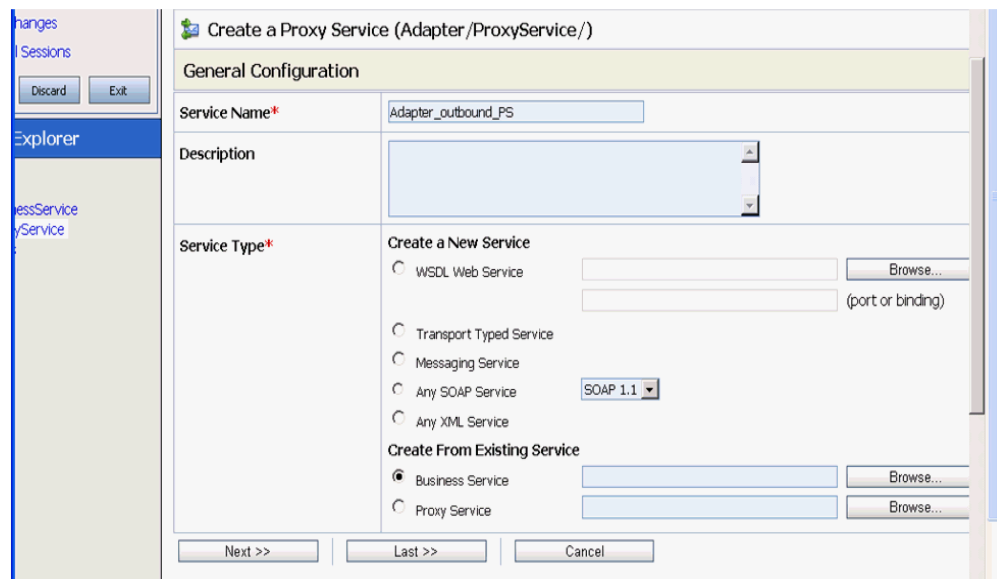
6. 図 7-134 に示すように、「サービス名」フィールドに適切な名前を入力します。

図 7-134 サービス名



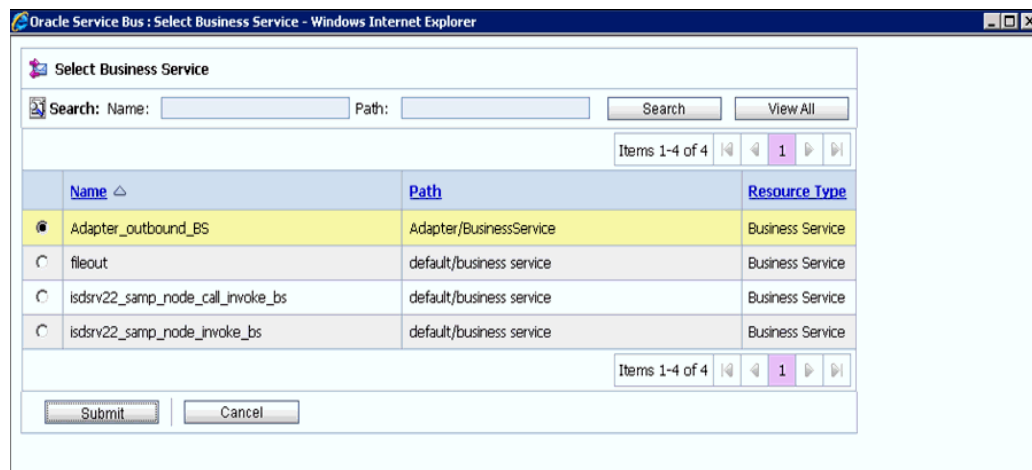
7. 図 7-135 に示すように、「サービス・タイプ」セクションで、「既存のサービスから作成」の下にある「ビジネス・サービス」ラジオ・ボタンを選択し、「参照」をクリックします。

図 7-135 全般的な構成



8. 図 7-136 に示すように、既存のビジネス・サービスを選択し、「発行」をクリックします。

図 7-136 ビジネス・サービス



9. 図 7-137 に示すように、「次へ」をクリックします。

図 7-137 一般的な構成

The screenshot shows the 'Create a Proxy Service (Adapter/ProxyService/)' dialog box. The 'General Configuration' section is active. The 'Service Name' field contains 'Adapter_outbound_PS'. The 'Service Type' section has two sub-sections: 'Create a New Service' and 'Create From Existing Service'. Under 'Create From Existing Service', the 'Business Service' radio button is selected, and the text field contains 'Adapter/BusinessService/Adapter_outbc'. At the bottom, the 'Next >>' button is highlighted.

10. 図 7-138 に示すように、「プロトコル」リストで「http」を選択し、「次へ」をクリックします。

図 7-138 トランスポート構成

The screenshot shows the 'Create a Proxy Service (Adapter/ProxyService/Adapter_outbound_PS)' dialog box, now on the 'Transport Configuration' tab. The 'Protocol' dropdown is set to 'http'. The 'Endpoint URI' field contains '/Adapter/ProxyService/Adapter_outbound_PS'. The 'Get All Headers' section has the 'No' radio button selected. At the bottom, the 'Next >>' button is highlighted.

11. 図 7-139 に示すように、「次へ」をクリックします。

図 7-139 HTTP トランスポート構成

The screenshot shows the 'Create a Proxy Service (Adapter/ProxyService/Adapter_outbound_PS)' configuration wizard. The 'Project Explorer' on the left shows the project structure: Adapter > BusinessService > ProxyService > wsdl > default. The main configuration area is titled 'HTTP Transport Configuration' and includes the following settings:

- HTTPS required:**
- Authentication:**
 - None
 - Basic
 - Client Certificate
 - Custom Authentication (See Advanced Settings)
- Dispatch Policy:** default
- Request Encoding:** [Empty text box]
- Response Encoding:** [Empty text box]

At the bottom, there is an 'Advanced Settings' section with navigation buttons: '<< Prev.', 'Next >>', 'Last >>', and 'Cancel'.

12. 図 7-140 に示すように、「次へ」をクリックします。

図 7-140 操作選択構成

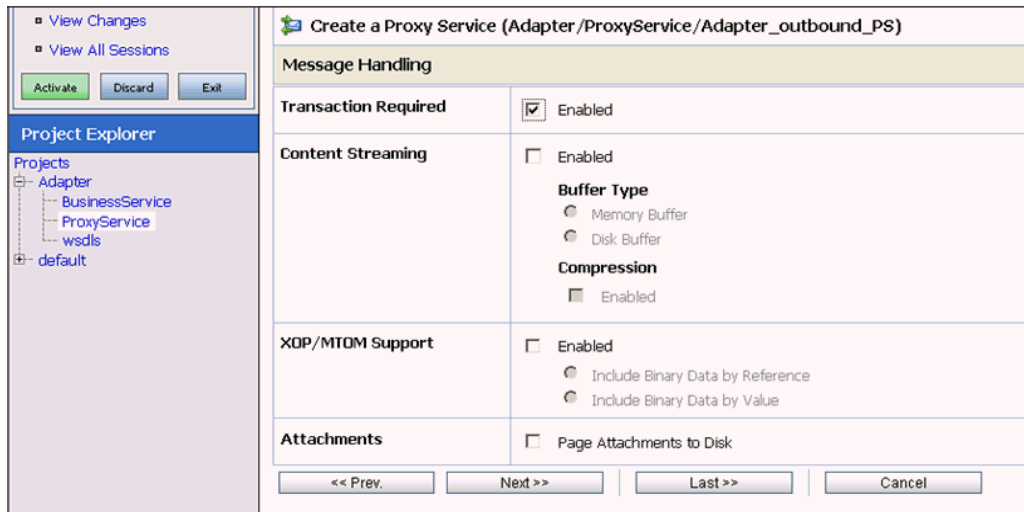
The screenshot shows the 'Create a Proxy Service (Adapter/ProxyService/Adapter_outbound_PS)' configuration wizard. The 'Project Explorer' on the left shows the project structure: Adapter > BusinessService > ProxyService > wsdl > default. The main configuration area is titled 'Operation Selection Configuration' and includes the following settings:

- Enforce WS-I Compliance:**
- Selection Algorithm:**
 - Transport Header
 - SOAPAction Header
 - WS-Addressing
 - SOAP Header
 - SOAP Body Type

At the bottom, there are navigation buttons: '<< Prev.', 'Next >>', 'Last >>', and 'Cancel'.

13. 図 7-141 に示すように、「トランザクションが必要」チェック・ボックスを有効にし、「次へ」をクリックします。

図 7-141 メッセージ処理



14. 図 7-142 に示すように、「保存」をクリックします。

図 7-142 保存

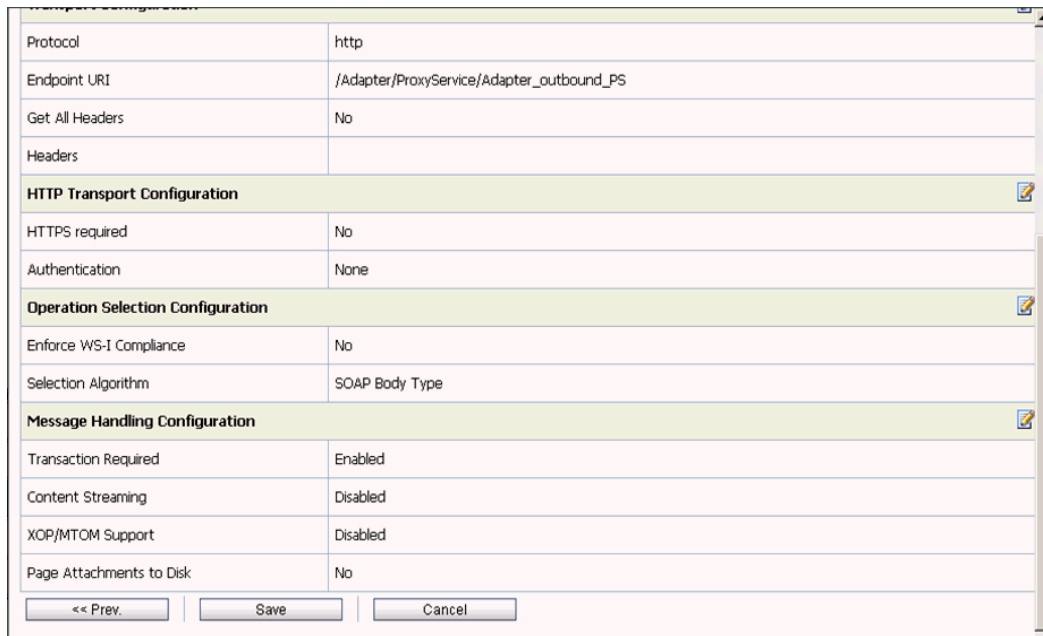
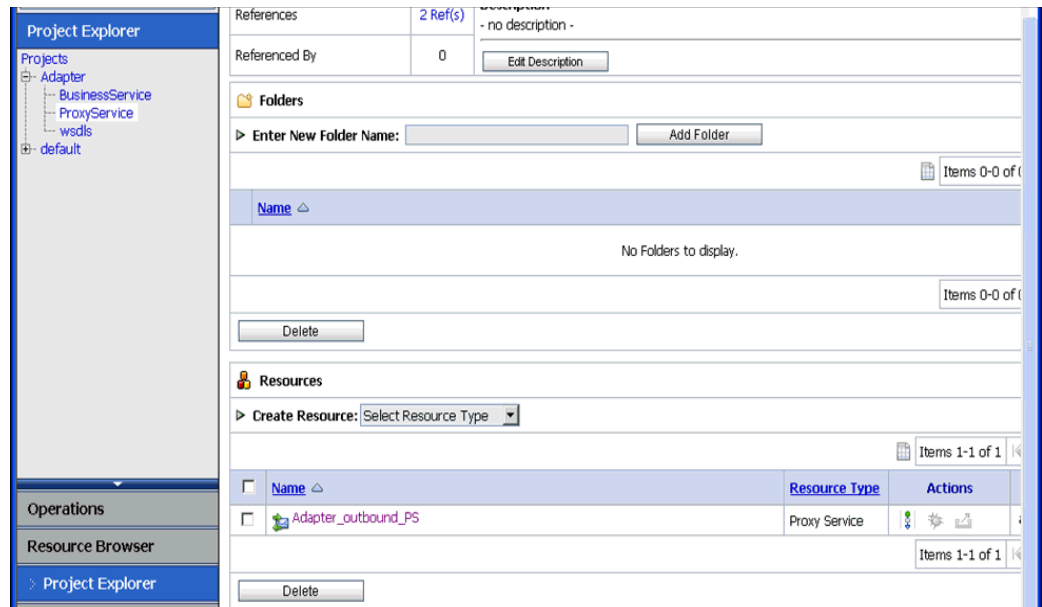


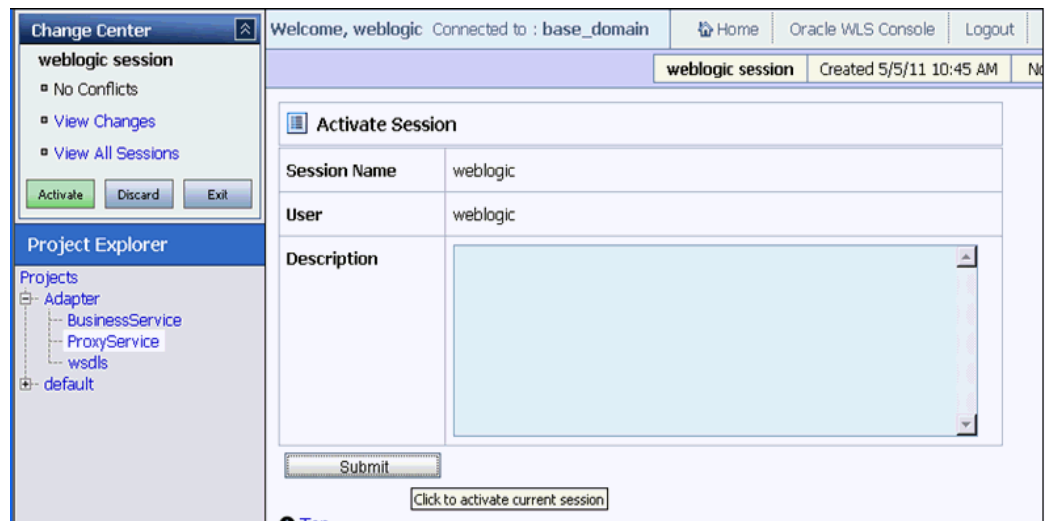
図 7-143 に示すように、作成したプロキシ・サービスが保存されます。

図 7-143 プロキシ・サービス



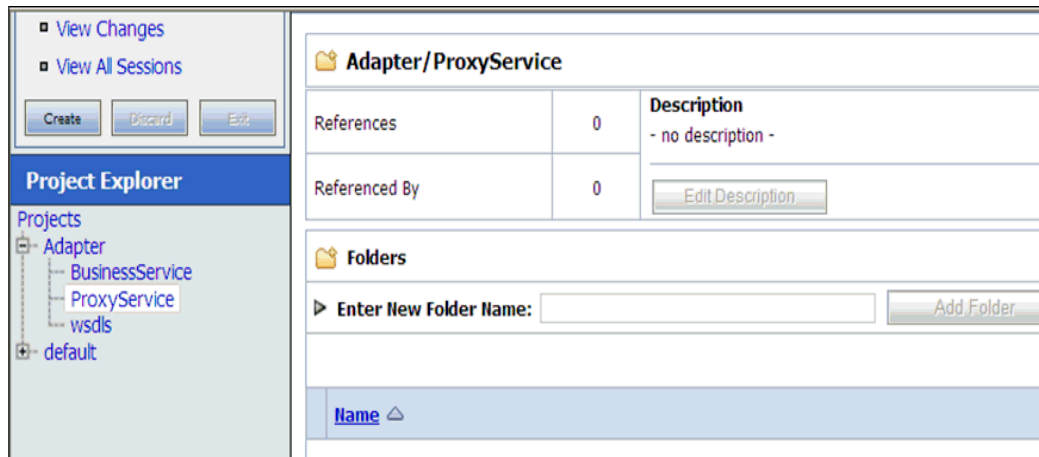
15. 図 7-144 に示すように、左ペインで「アクティブ化」をクリックし、右ペインで「発行」をクリックします。

図 7-144 セッションのアクティブ化



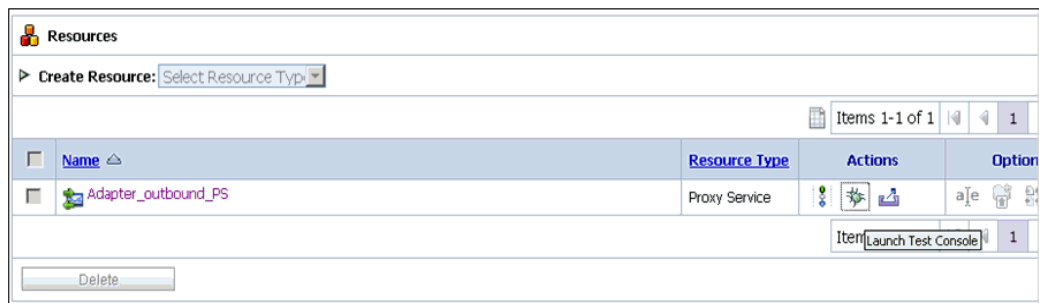
16. 図 7-145 に示すように、左ペインで「プロジェクト」フォルダの「ProxyService」をクリックします。

図 7-145 ProxyService



17. 図 7-146 に示すように、作成されたプロキシ・サービスに対する「テスト・コンソールの起動」アイコンをクリックします。

図 7-146 テスト・コンソールの起動



18. 「直接呼出し」チェック・ボックスを選択解除し、「ペイロード」の入力値を指定して、「実行」をクリックします。
19. 「レスポンス・ドキュメント」をレビューします。

JDeveloperを使用したOracle Service Bus用の アウトバウンドおよびインバウンド・プロセス の構成

Oracle Application Adapter for J.D. Edwards OneWorld には、Oracle JDeveloper がシームレスに統合されており、Web サービスの統合を容易にします。

この章の内容は以下のとおりです。

- 8.1 項「JDeveloper を使用した OSB アウトバウンド・プロセスの構成 (J2CA 構成)」
- 8.2 項「JDeveloper を使用した OSB インバウンド・プロセスの構成 (J2CA 構成)」
- 8.3 項「JDeveloper を使用した OSB アウトバウンド・プロセスの構成 (BSE 構成)」
- 8.4 項「JDeveloper を使用した JMS インバウンド・プロセスの構成 (J2CA 構成)」
- 8.5 項「JDeveloper を使用した JMS アウトバウンド・プロセスの構成 (J2CA 構成)」
- 8.6 項「JDeveloper を使用した HTTP アウトバウンド・プロセスの構成 (J2CA 構成)」

8.1 JDeveloper を使用した OSB アウトバウンド・プロセスの 構成 (J2CA 構成)

この項では、J2CA 構成で Oracle JDeveloper を使用して J.D. Edwards OneWorld システム用に OSB アウトバウンド・プロセスを構成する方法について説明します。

このアウトバウンド・ユース・ケース・シナリオ用のサンプル・プロジェクトが、次に示す Application Adapter のインストール・フォルダに用意されています。

```
<ADAPTER_HOME>\etc\sample\JDEdwards_Samples.zip\JDEdwards_Samples\OSB_
Jdeveloper\J2CA\JDEdwards_Sample_J2CA_OSB_Outbound_Project
```

この項では、次のトピックについて説明します。

- 8.1.1 項「OSB 用のサービス・バス・アプリケーションの作成」
- 8.1.2 項「OSB アウトバウンド・プロセスの定義」
- 8.1.3 項「OSB アウトバウンド・プロセスのデプロイ」

前提条件

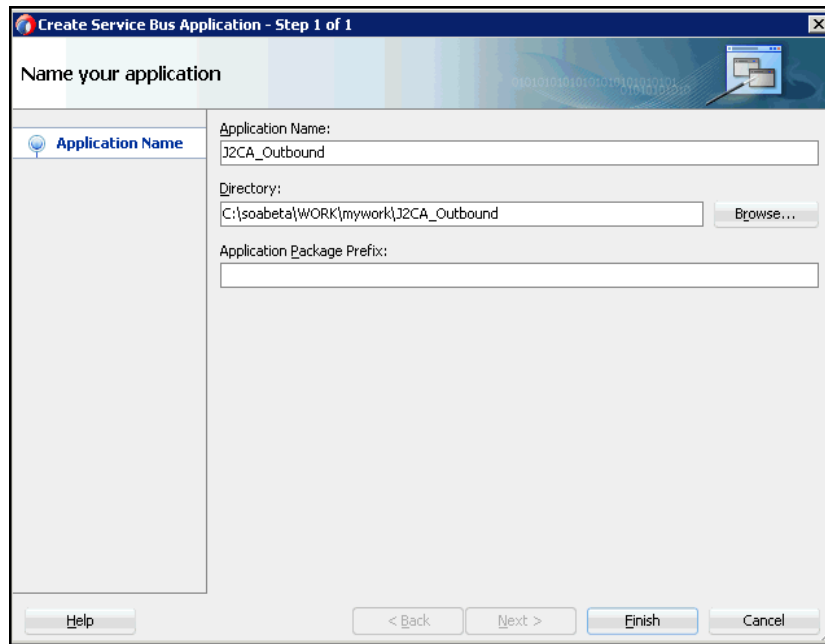
OSB アウトバウンド・プロセスを設計する前に、アプリケーション・エクスペローラを使用して各 WSDL ファイルを生成する必要があります。詳細は、4-8 ページの [4.4.1 項「リクエスト/レスポンス型のサービス用の WSDL の生成」](#) を参照してください。

8.1.1 OSB 用のサービス・バス・アプリケーションの作成

OSB 用のサービス・バス・アプリケーションを作成するには、JDeveloper で次のステップを実行します。

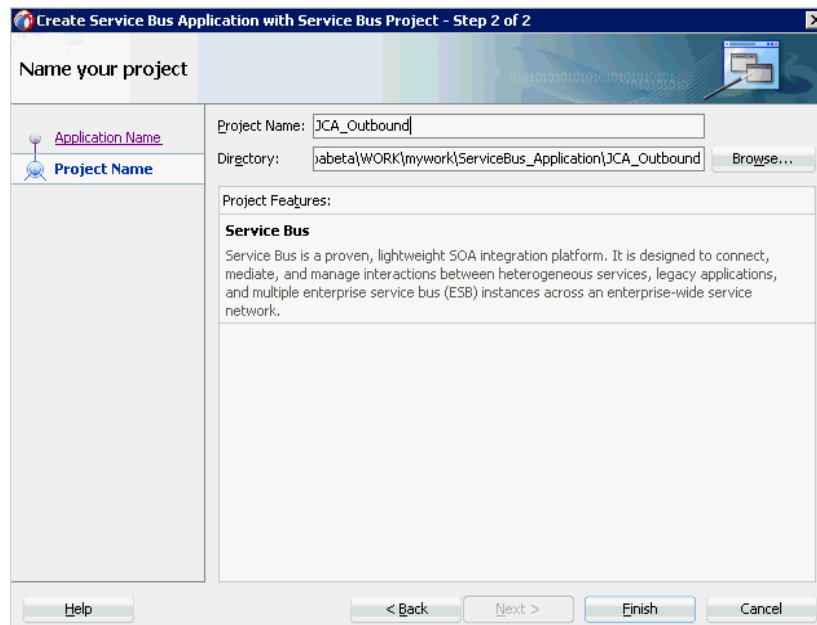
1. 新規の OSB アプリケーションを作成します。
2. 図 8-1 に示すように、OSB アプリケーションの名前を入力し (「J2CA_Outbound」など)、「終了」をクリックします。

図 8-1 「アプリケーションの名前付け」 ペイン



3. 図 8-2 に示すように、プロジェクト名を入力し (「JCA_Outbound」など)、「終了」をクリックします。

図 8-2 「プロジェクトの名前付け」 ペイン



8.1.2 OSB アウトバウンド・プロセスの定義

この項では、OSB アウトバウンド・プロセスの定義方法について説明します。次のトピックが含まれています。

- 8.1.2.1 項「サード・パーティ・アダプタ・サービス・コンポーネントの構成」
- 8.1.2.2 項「ファイル・トランスポート・タイプ・ビジネス・サービスの構成」
- 8.1.2.3 項「プロキシ・サービスとパイプラインの作成」
- 8.1.2.4 項「ルーティング・ルールの構成」

8.1.2.1 サード・パーティ・アダプタ・サービス・コンポーネントの構成

ビジネス・サービスとともにサード・パーティ・アダプタ・サービス・コンポーネントを作成するには、次のステップを実行します。

1. 図 8-3 に示すように、「Service Bus」コンポーネント・ペインから「サード・パーティ・アダプタ」コンポーネントを「外部サービス」ペインにドラッグ・アンド・ドロップします。

図 8-3 「サード・パーティ・アダプタ」コンポーネント

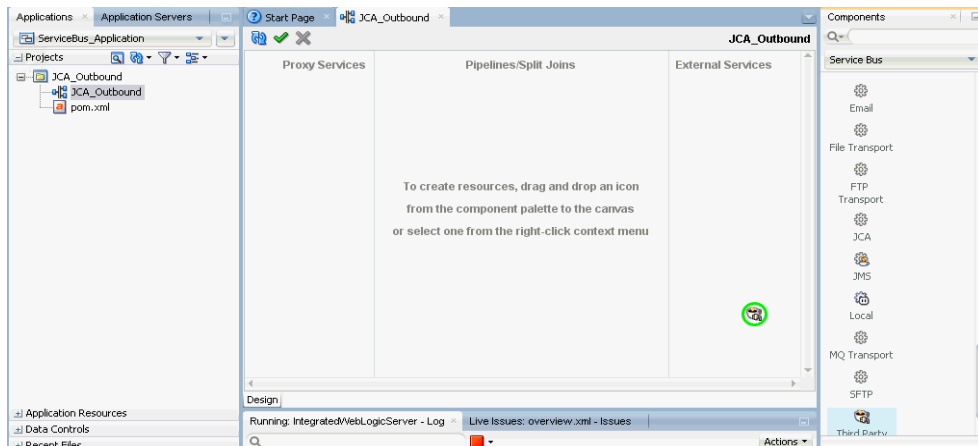
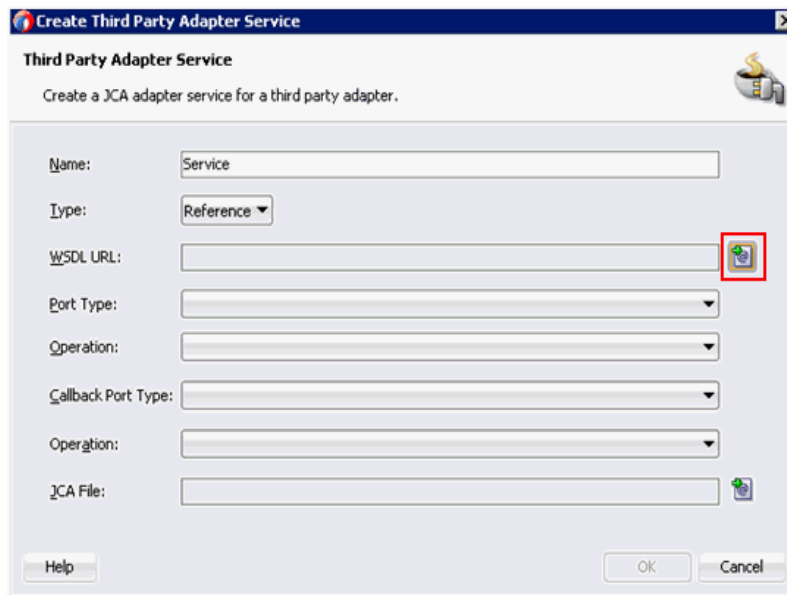


図 8-4 に示すように、「サード・パーティ・アダプタ・サービスの作成」ダイアログが表示されます。

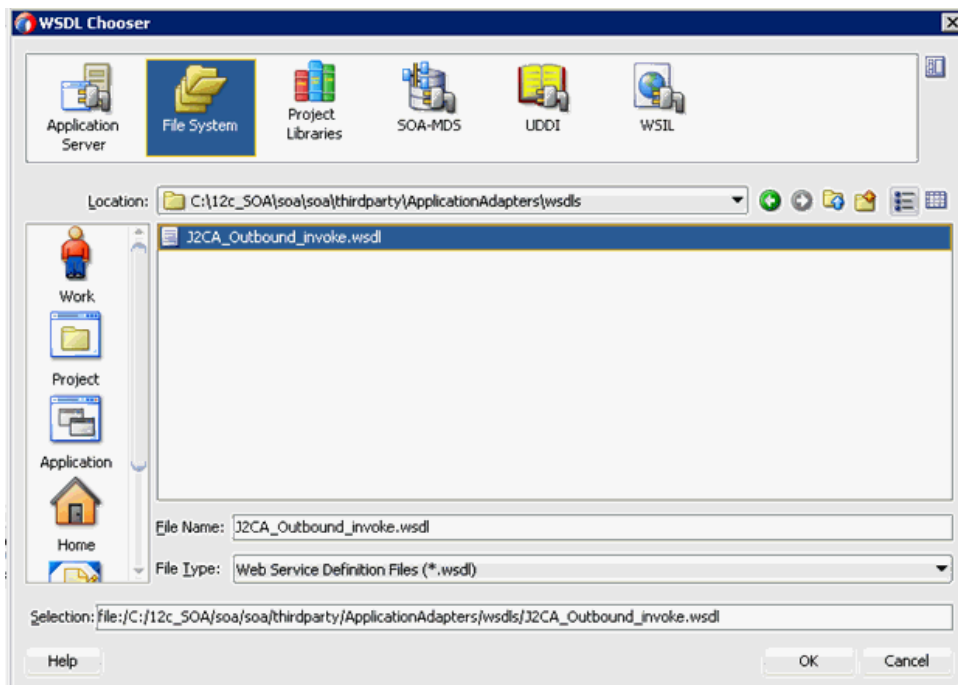
図 8-4 「サード・パーティ・アダプタ・サービスの作成」ペイン



2. サード・パーティ・アダプタ・サービスの適切な名前を入力します。この名前はビジネス・サービス名として使用されます。
3. 「タイプ」ドロップダウン・リストで「参照」が選択されていることを確認します (デフォルト)。
4. 「WSDL URL」フィールドの右の「既存の WSDL を検索します。」アイコンをクリックします。

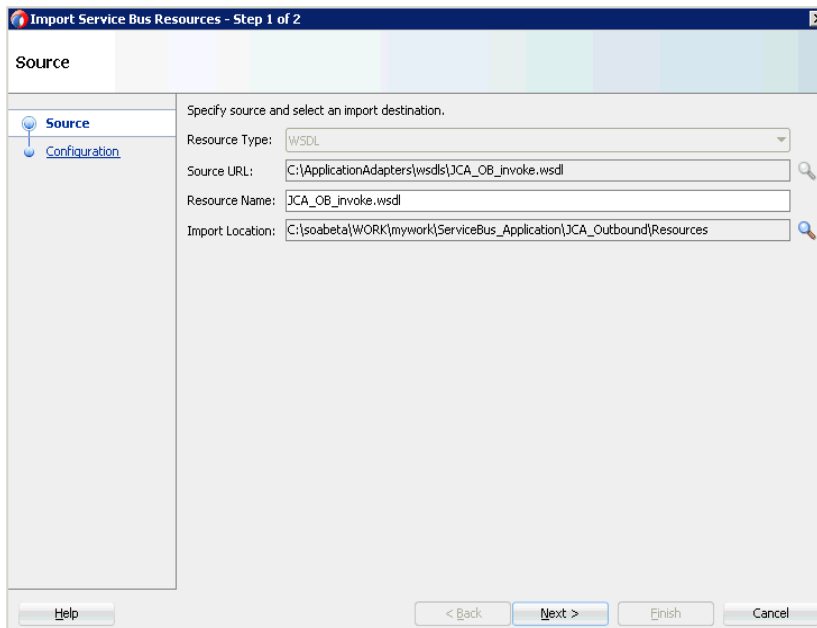
図 8-5 に示すように、「WSDL チューザ」ダイアログが表示されます。

図 8-5 WSDL チューザ」ダイアログ



5. 「ファイルシステム」タブを選択し、WSDL ディレクトリをブラウズしてアウトバウンド WSDL ファイルを選択します。
6. 「OK」をクリックします。
「Service Bus リソースのインポート」ダイアログが表示されます。
7. 図 8-6 に示すように、「次へ」をクリックします。

図 8-6 「ソース」ペイン



8. 「構成」ペインで、「終了」をクリックします。

「サード・パーティ・アダプタ・サービスの作成」ダイアログに戻ります。

9. 「JCA ファイル」フィールドの右の「JCA ファイルを検索します。」アイコンをクリックします。

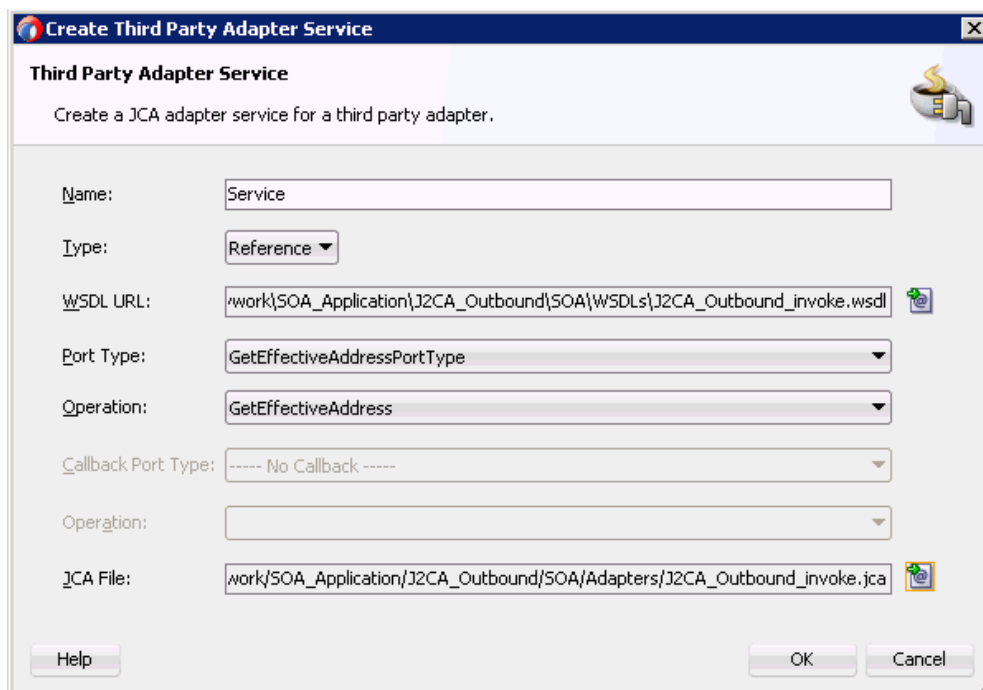
変換チューザ・ダイアログが表示されます。

10. WSDL ディレクトリから JCA プロパティ・ファイルを選択します。
11. 「OK」をクリックします。「ファイルのコピー」メッセージが表示されます。
12. 「はい」をクリックします。

JCA プロパティ・ファイルのコピーがプロジェクト・フォルダ内に作成されます。

図 8-7 に示すように、「サード・パーティ・アダプタ・サービスの作成」ダイアログに戻ります。

図 8-7 「サード・パーティ・アダプタ・サービスの作成」ダイアログ



13. 「OK」をクリックします。

ビジネス・サービス・コンポーネントが「外部サービス」ペインに作成されます。

8.1.2.2 ファイル・トランスポート・タイプ・ビジネス・サービスの構成

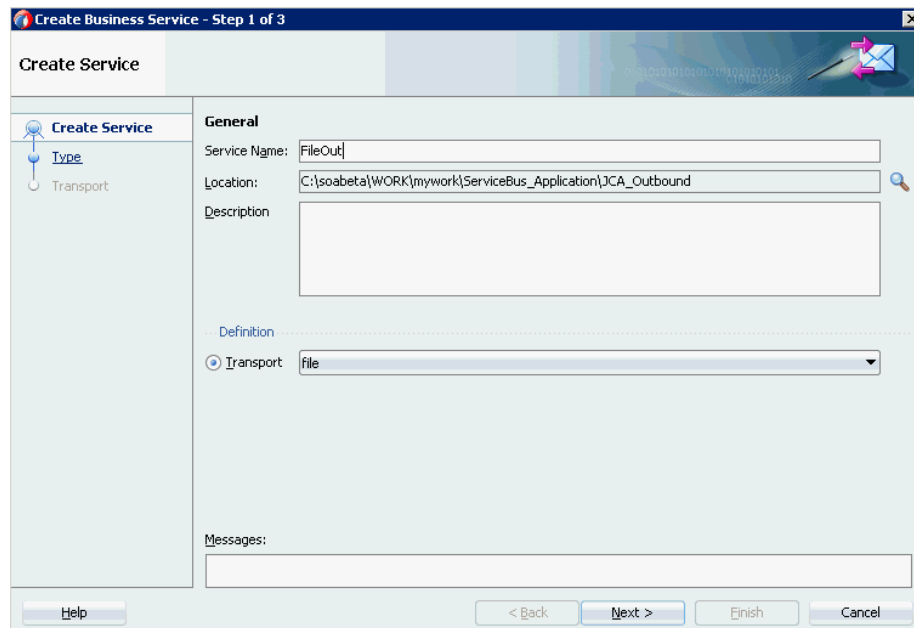
ファイル・トランスポート・ビジネス・サービスを作成するには、次のステップを実行します。

1. 「拡張」ペインから「ファイル・トランスポート」コンポーネントを「外部サービス」ペインにドラッグ・アンド・ドロップします。

「ビジネス・サービスの作成」ダイアログが表示されます。

2. 図 8-8 に示すように、「サービス名」フィールドにビジネス・サービスの任意の名前を入力し(「FileOut」など)、「次へ」をクリックします。

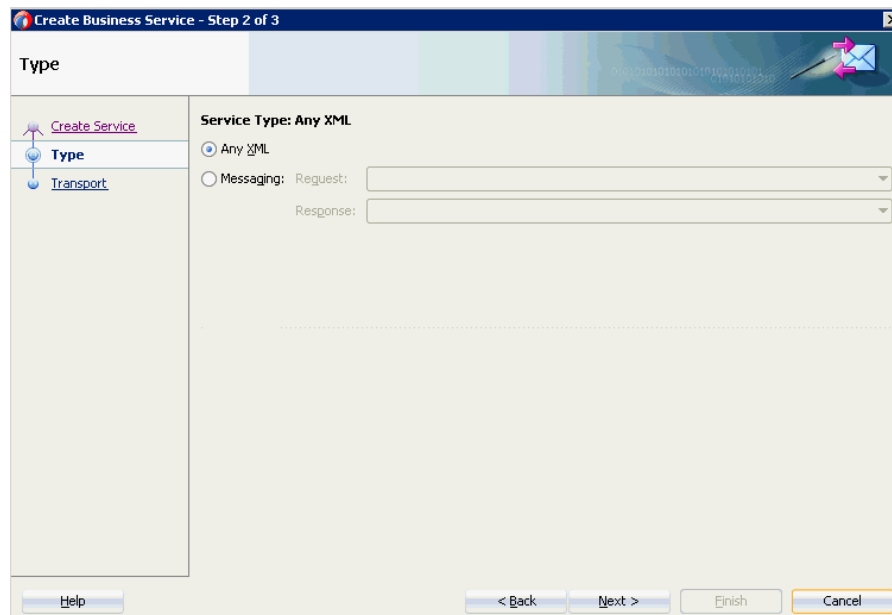
図 8-8 「サービスの作成」 ペイン



「タイプ」 ペインが表示されます。「任意の XML」 オプションがデフォルトで選択されています。

3. 図 8-9 に示すように、「次へ」をクリックします。

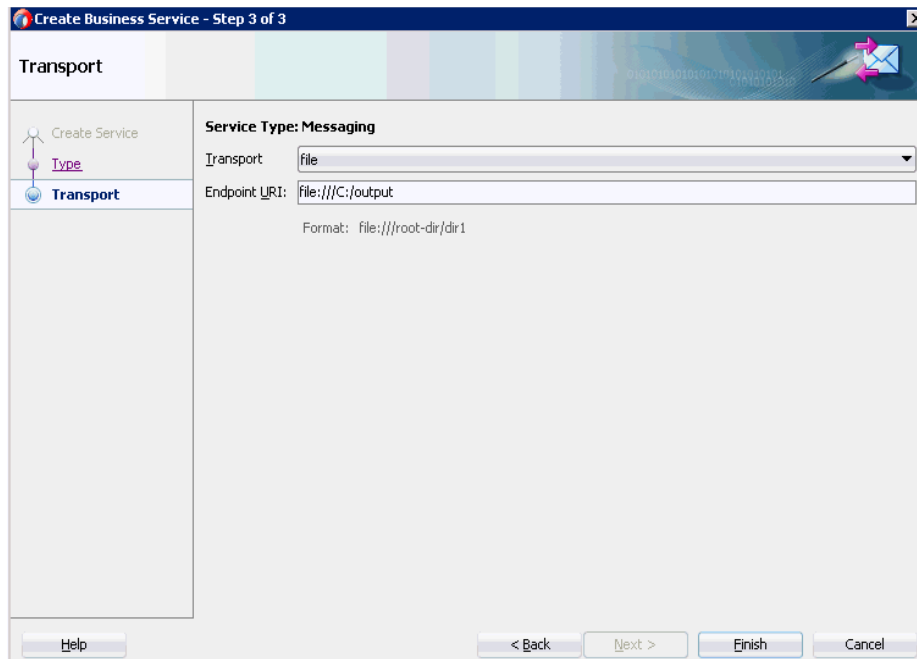
図 8-9 「タイプ」 ペイン



「トランスポート」 ペインが表示されます。

4. 図 8-10 に示すように、「エンドポイント URI」 フィールドに出力場所を指定し (「c:/output」 など)、「終了」をクリックします。

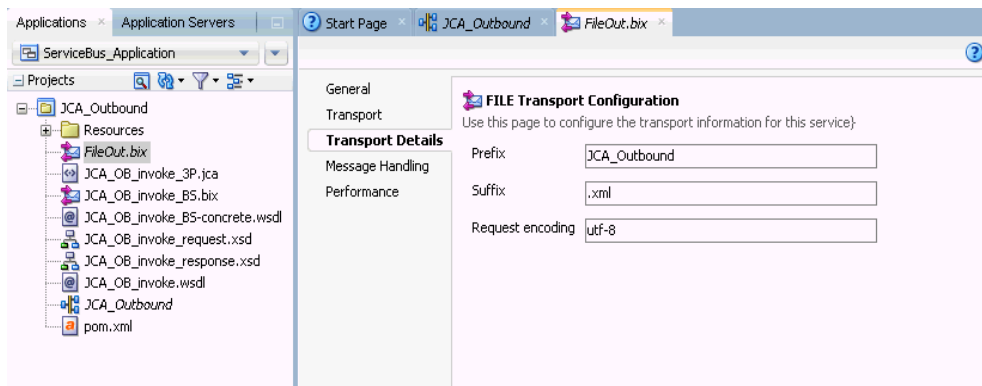
図 8-10 「トランスポート」 ペイン



ファイル・トランスポート・ビジネス・サービスの「FileOut」が作成され、表示されます。

- 作成されたビジネス・サービスの「FileOut」をダブルクリックし、図 8-11 に示すように、「トランスポートの詳細」タブの「接頭辞」フィールドと「接尾辞」フィールドに値を入力します。

図 8-11 トランスポートの詳細

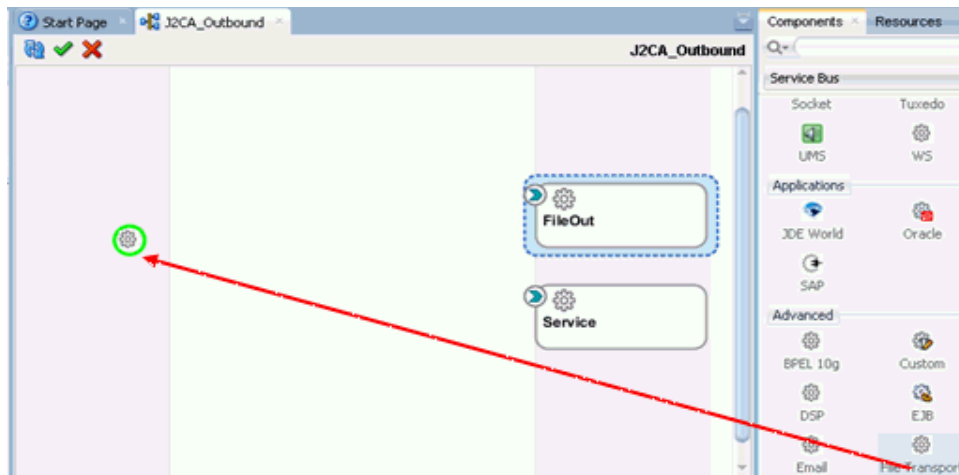


8.1.2.3 プロキシ・サービスとパイプラインの作成

プロキシ・サービスとパイプラインを作成するには、次のステップを実行します。

- 図 8-12 に示すように、「拡張」コンポーネント・ペインから「ファイル・トランスポート」コンポーネントを「プロキシ・サービス」ペインにドラッグ・アンド・ドロップします。

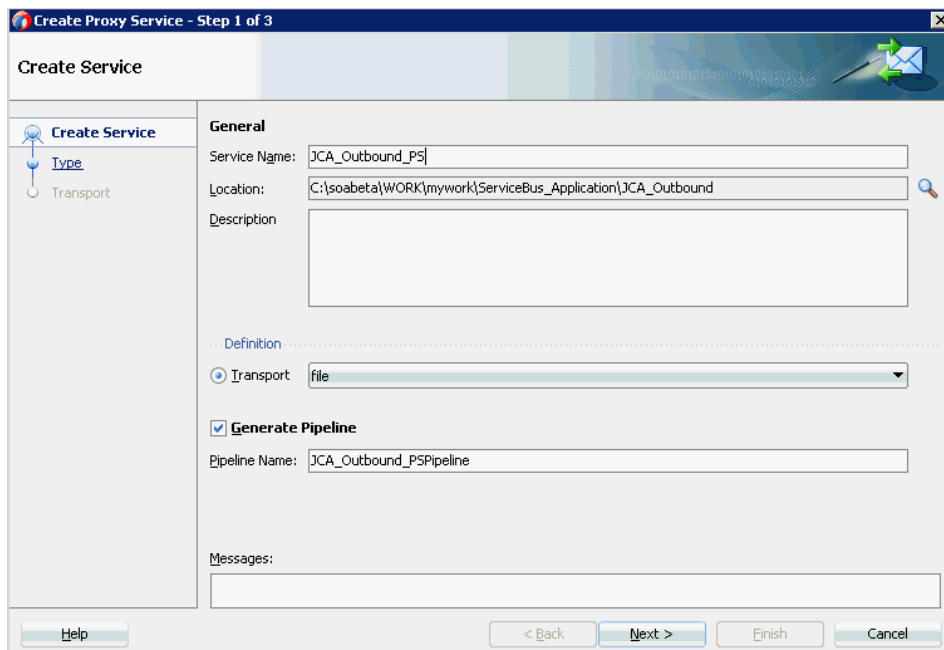
図 8-12 「ファイル・トランスポート」コンポーネント



「プロキシ・サービスの作成」 ペインが表示されます。

2. 「サービス名」 フィールドにプロキシ・サービスの任意の名前を入力します (「JCA_Outbound_PS」など)。デフォルトで、「パイプラインの生成」が選択されています。
3. 図 8-13 に示すように、「次へ」をクリックします。

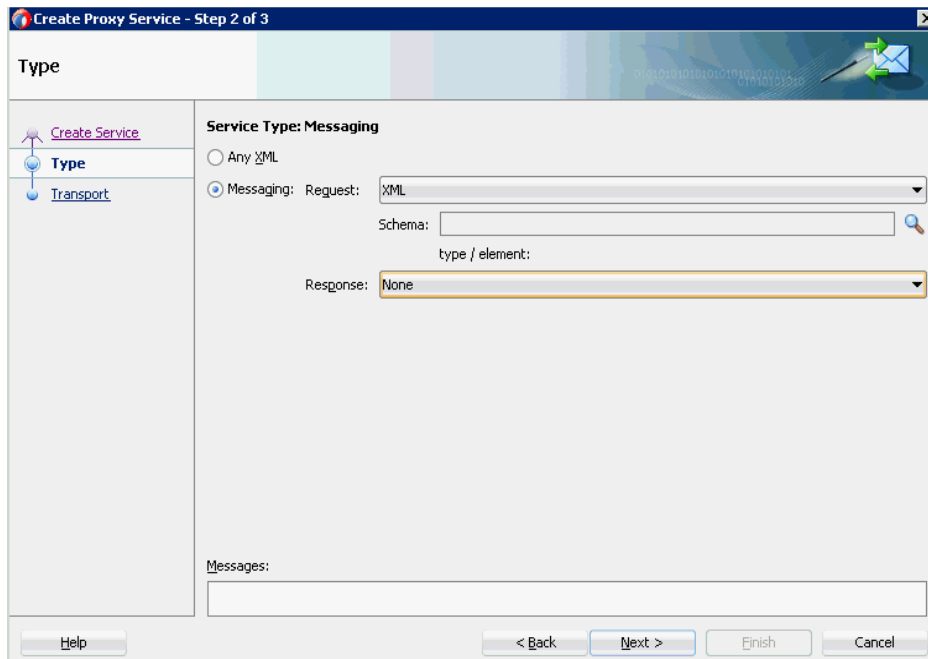
図 8-13 「サービスの作成」ペイン



「タイプ」 ペインが表示されます。

4. 図 8-14 に示すように、「メッセージング」オプションを選択し、「リクエスト」を「XML」に、「レスポンス」を「なし」に設定して、「次へ」をクリックします。

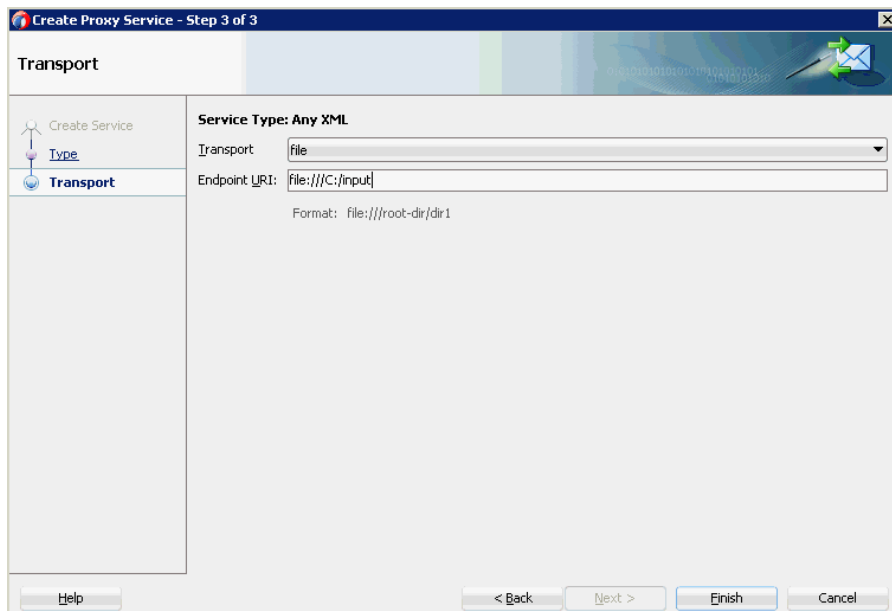
図 8-14 「タイプ」 ペイン



「トランスポート」 ウィンドウが表示されます。

5. 図 8-15 に示すように、「エンドポイント URI」 フィールドに入力場所を指定し（「c:/input」 など）、「終了」をクリックします。

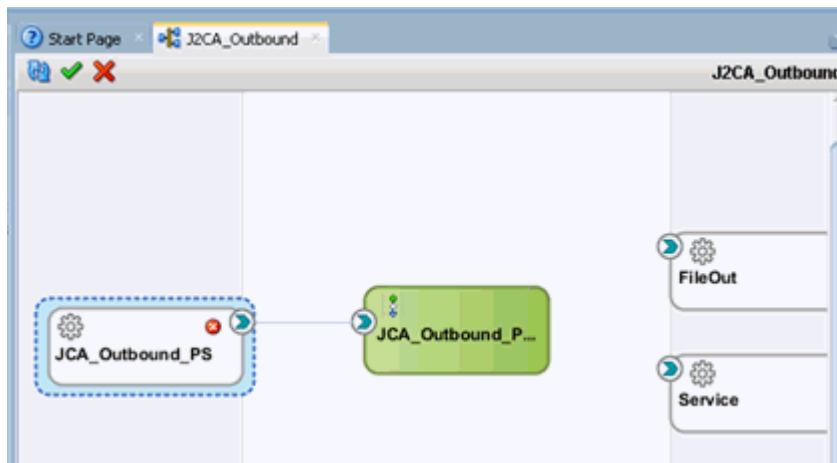
図 8-15 「トランスポート」 ウィンドウ



プロキシ・サービスとパイプラインが作成され、表示されます。

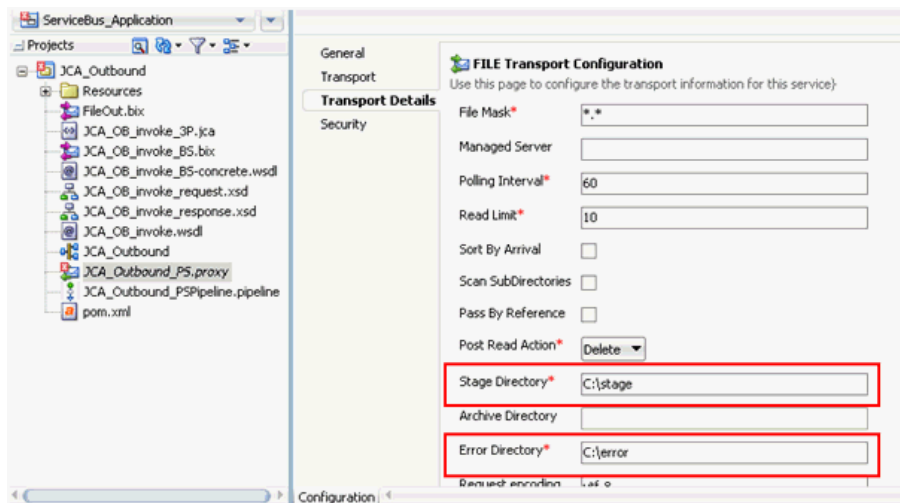
6. 図 8-16 に示すように、作成されたプロキシ・サービス（「JCA_Outbound_PS」 など）をダブルクリックします。

図 8-16 プロキシ・サービスの編集



7. 図 8-17 に示すように、表示されたプロキシ・サービスの構成ページで、「トランスポートの詳細」を選択し、「ステージ・ディレクトリ」と「エラー・ディレクトリ」の値を入力します。

図 8-17 ファイル・トランスポート構成



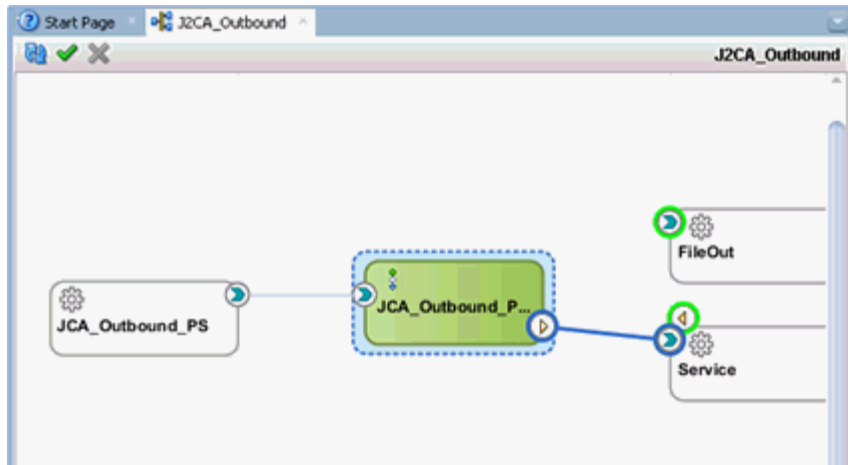
8. プロキシ・サービスの構成ページを保存して閉じます。

8.1.2.4 ルーティング・ルール of 構成

ルーティング・ルールを構成するには、次のステップを実行します。

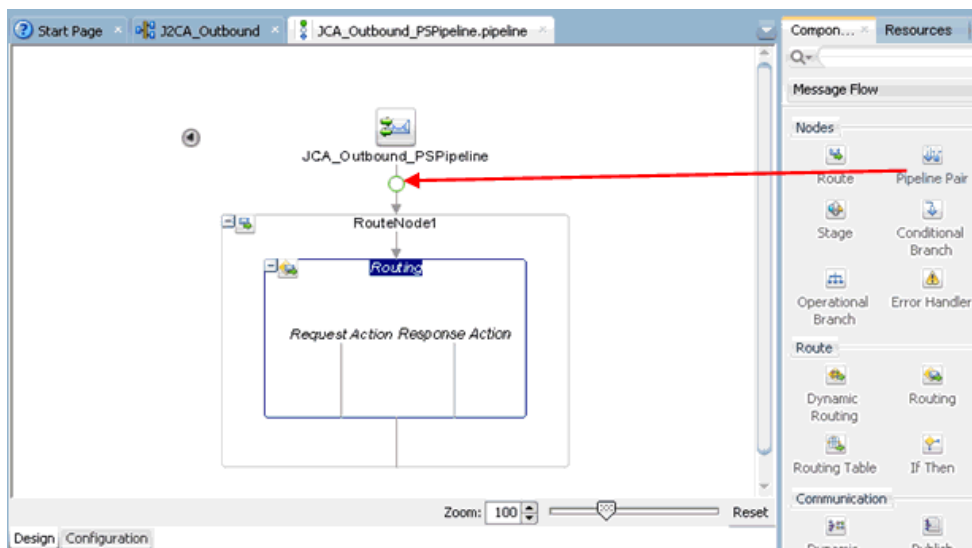
1. 図 8-18 に示すように、パイプラインをビジネス・サービス (「Service」など) に接続します。

図 8-18 ビジネス・サービス・パイプライン



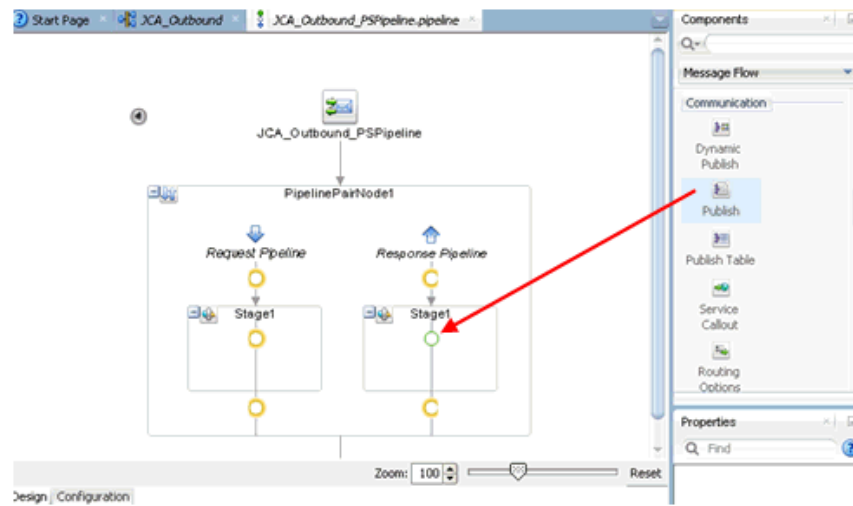
2. 「パイプライン/分割結合」 ペインでパイプライン (「JCA_Outbound_PSPipeline」など) をダブルクリックします。
パイプラインの構成ページが表示されます。
3. 図 8-19 に示すように、「ノード」 ペインから「パイプライン・ペア」 ノードをパイプライン (「JCA_Outbound_PSPipeline」など) の下の領域にドラッグ・アンド・ドロップします。

図 8-19 「パイプライン・ペア」 ノード



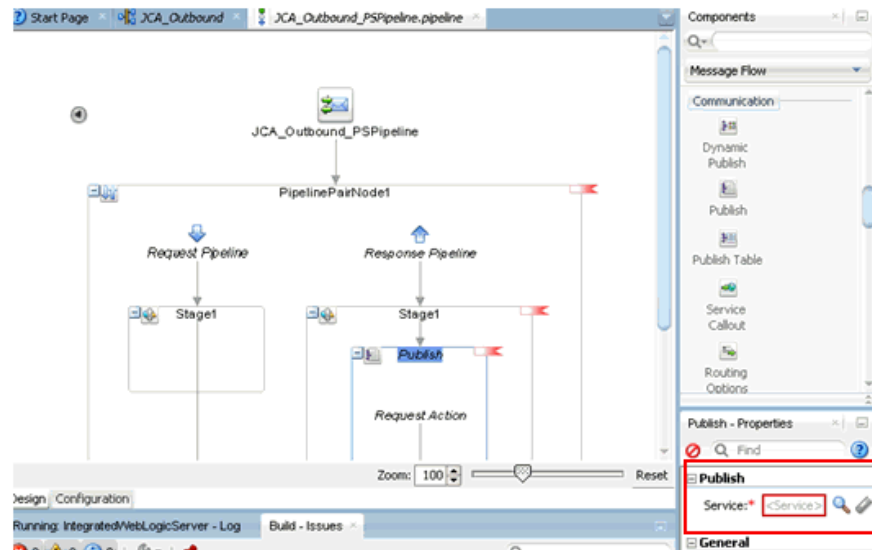
4. 図 8-20 に示すように、「通信」 ペインから「パブリッシュ」 ノードを「レスポンス・パイプライン」の「ステージ 1」の下の領域にドラッグ・アンド・ドロップします。

図 8-20 「パブリッシュ」ノード



5. 図 8-21 に示すように、パブリッシュのプロパティの右ペインで「サービス」フィールドの右の参照アイコンをクリックします。

図 8-21 「参照」アイコン



6. 図 8-22 に示すように、表示された「リソース・チューザ」ウィンドウで、「Fileout.bix」ファイル・トランスポート・ビジネス・サービスを選択し、「OK」をクリックします。

図 8-22 リソース・チューザ

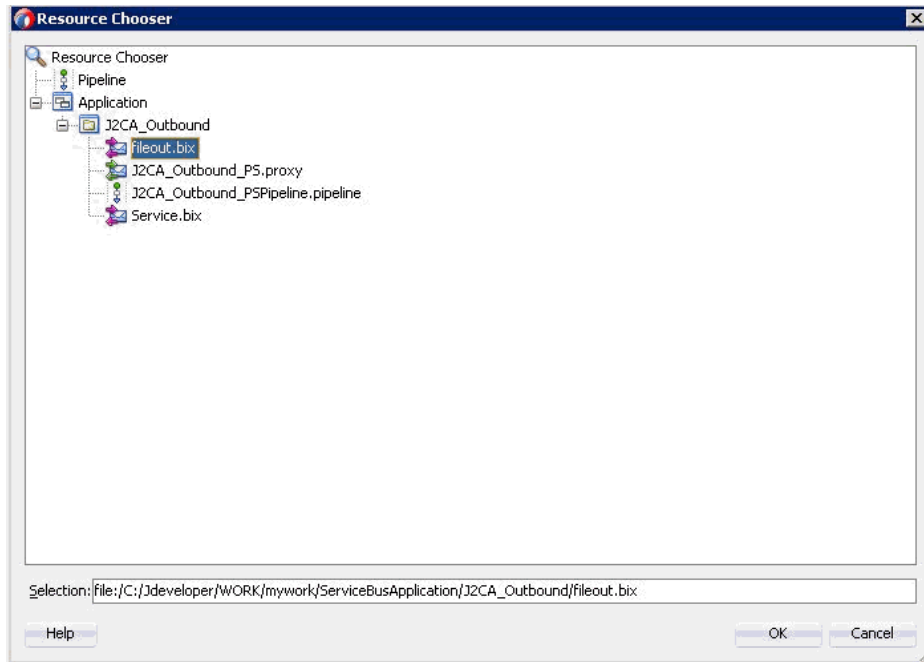
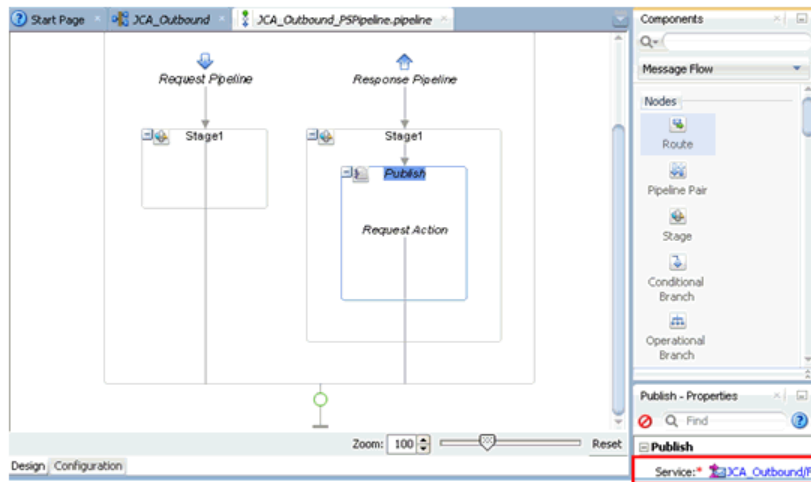


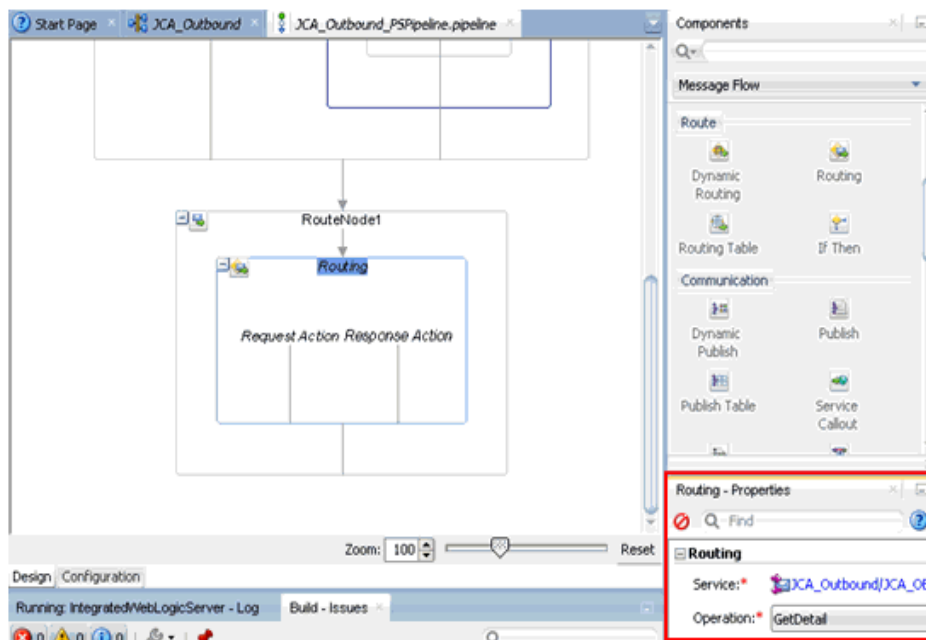
図 8-23 に示すように、選択したサービスが右ペインの「パブリッシュ」ペインで構成されます。

図 8-23 「パブリッシュ」ペイン



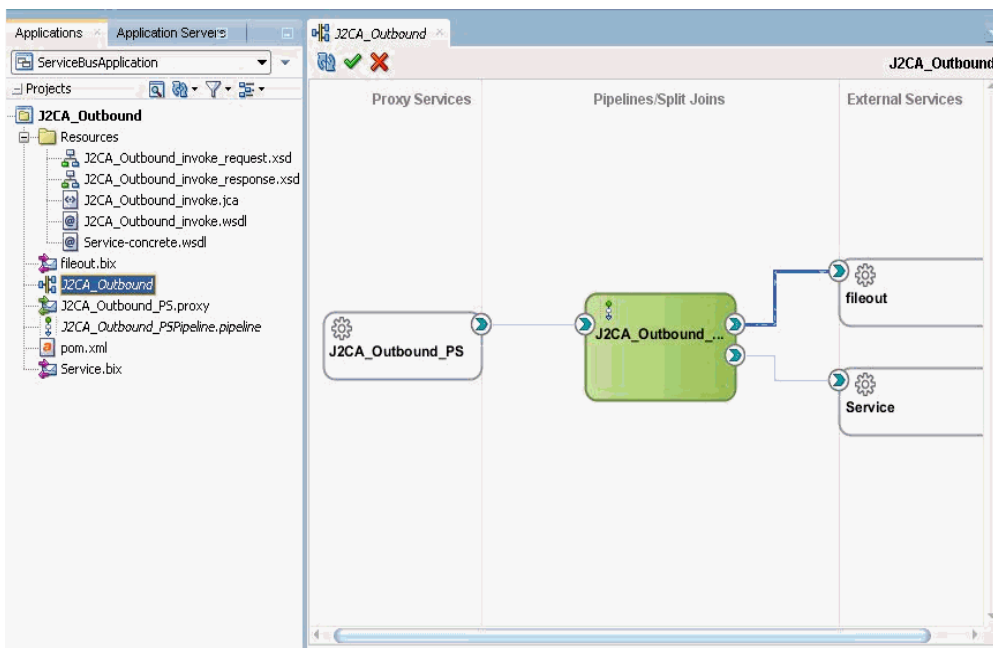
7. 図 8-24 に示すように、「ルーティング」をクリックしてサービスが適切に選択されていることを確認します。

図 8-24 パイプラインの構成



8. パイプラインの構成ページを保存して閉じます。
9. 図 8-25 に示すように、overview.xml ファイル (「J2CA_Outbound」など) をダブルクリックし、メニュー・バーの「すべて保存」をクリックして OSB プロセスを保存します。

図 8-25 「すべて保存」アイコン

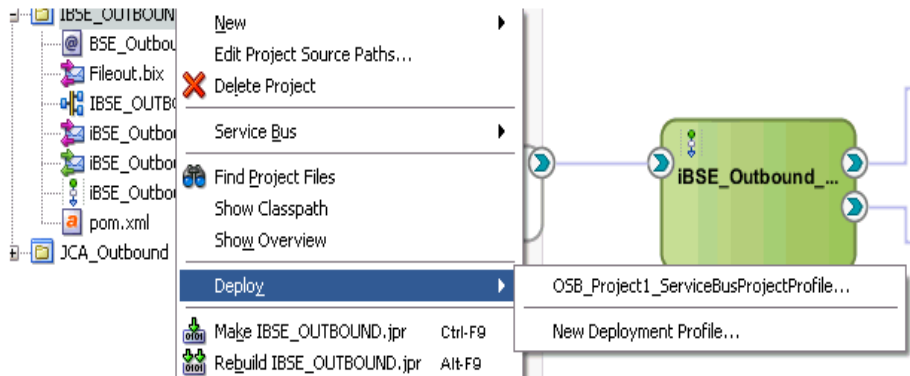


8.1.3 OSB アウトバウンド・プロセスのデプロイ

OSB アウトバウンド・プロセスをデプロイするには、次のステップを実行します。

1. 図 8-26 に示すように、OSB プロジェクトを右クリックし、「デプロイ」を選択して、「OSB_Project1_ServiceBusProjectProfile...」を選択します。

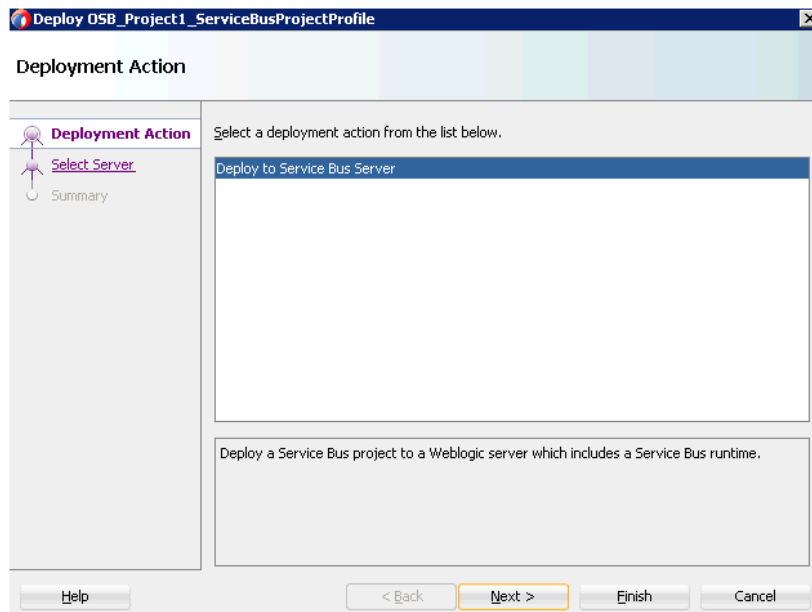
図 8-26 「デプロイ」オプション



「デプロイメント・アクション」ページが表示されます。

2. 図 8-27 に示すように、「次へ」をクリックします。

図 8-27 「デプロイメント・アクション」ページ



「サーバーの選択」ページが表示されます。

3. 図 8-28 に示すように、構成済の使用可能なアプリケーション・サーバーを選択し、「次へ」をクリックします。

図 8-28 「サーバーの選択」 ページ

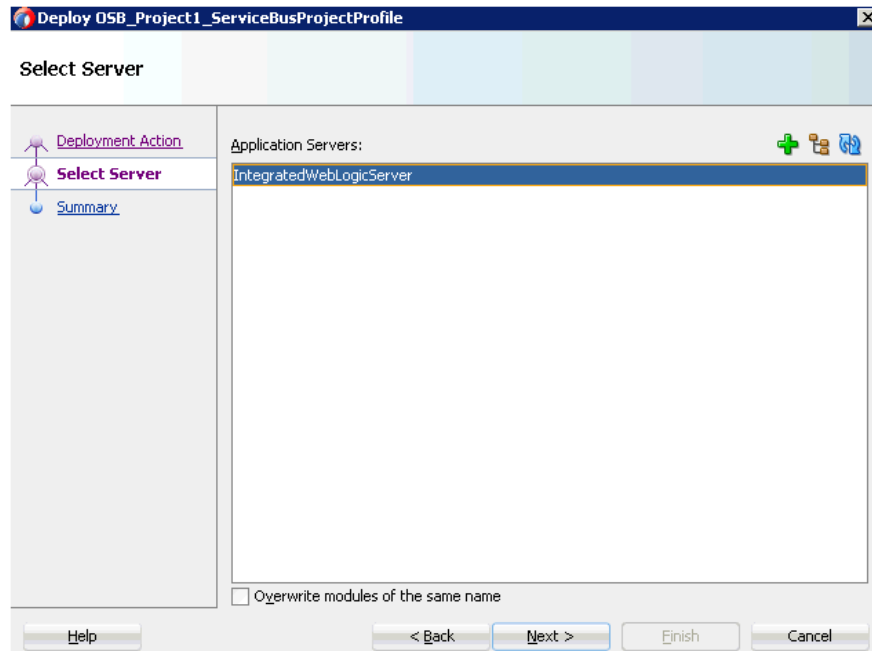
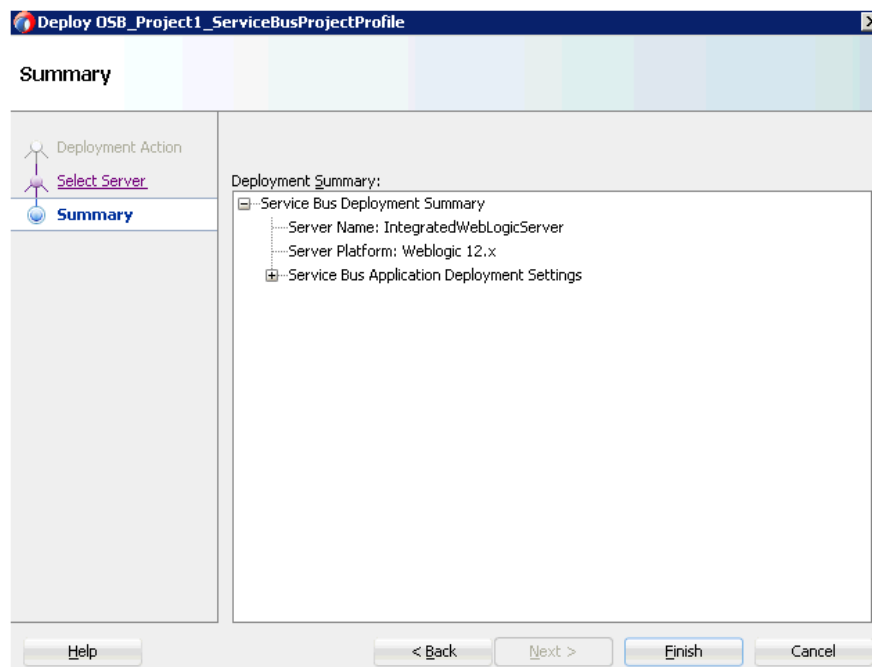


図 8-29 に示すように、「サマリー」 ページが表示されます。

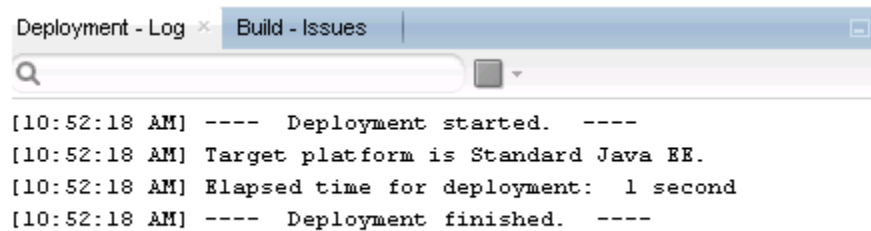
図 8-29 「サマリー」 ページ



4. プロジェクトに使用可能なすべてのデプロイメント情報をレビューおよび確認し、「終了」をクリックします。

図 8-30 に示すように、プロセスが正常にデプロイされます。

図 8-30 デプロイメントの成功を伝えるメッセージ



```

Deployment - Log x Build - Issues
[10:52:18 AM] ---- Deployment started. ----
[10:52:18 AM] Target platform is Standard Java EE.
[10:52:18 AM] Elapsed time for deployment: 1 second
[10:52:18 AM] ---- Deployment finished. ----

```

5. 入力 XML ファイルをコピーして構成済の入力フォルダ (「C:\input」など) に貼り付けます。

構成済の出力場所 (「C:\output」など) で出力が受信されたことを確認します。

8.2 JDeveloper を使用した OSB インバウンド・プロセスの構成 (J2CA 構成)

この項では、J2CA 構成で Oracle JDeveloper を使用して J.D. Edwards OneWorld システム用に OSB インバウンド・プロセスを構成する方法について説明します。

このインバウンド・ユース・ケース・シナリオ用のサンプル・プロジェクトが、次に示す Application Adapter のインストール・フォルダに用意されています。

```
<ADAPTER_HOME>\etc\sample\JDEdwards_Samples.zip\JDEdwards_Samples\OSB_
JDeveloper\J2CA\JDEdwards_Sample_J2CA_OSB_Inbound_Project
```

この項では、次のトピックについて説明します。

- [8.2.1 項「OSB 用のサービス・バス・アプリケーションの作成」](#)
- [8.2.2 項「OSB インバウンド・プロセスの定義」](#)
- [8.2.3 項「OSB インバウンド・プロセスのデプロイ」](#)

前提条件

OSB インバウンド・プロセスを設計する前に、アプリケーション・エクスペローラを使用して各 WSDL ファイルを生成する必要があります。詳細は、4-35 ページの [4.5.1 項「イベント統合のための WSDL の生成」](#) を参照してください。

8.2.1 OSB 用のサービス・バス・アプリケーションの作成

OSB インバウンド・プロセスを構成するには、OSB 用のサービス・バス・アプリケーションを作成する必要があります。詳細は、8-2 ページの [8.1.1 項「OSB 用のサービス・バス・アプリケーションの作成」](#) を参照してください。

8.2.2 OSB インバウンド・プロセスの定義

この項では、OSB インバウンド・プロセスの定義方法について説明します。次のトピックが含まれています。

- [8.2.2.1 項「サード・パーティ・アダプタ・サービス・コンポーネントの構成」](#)
- [8.2.2.2 項「パイプラインの作成」](#)
- [8.2.2.3 項「ファイル・トランスポート・タイプ・ビジネス・サービスの構成」](#)

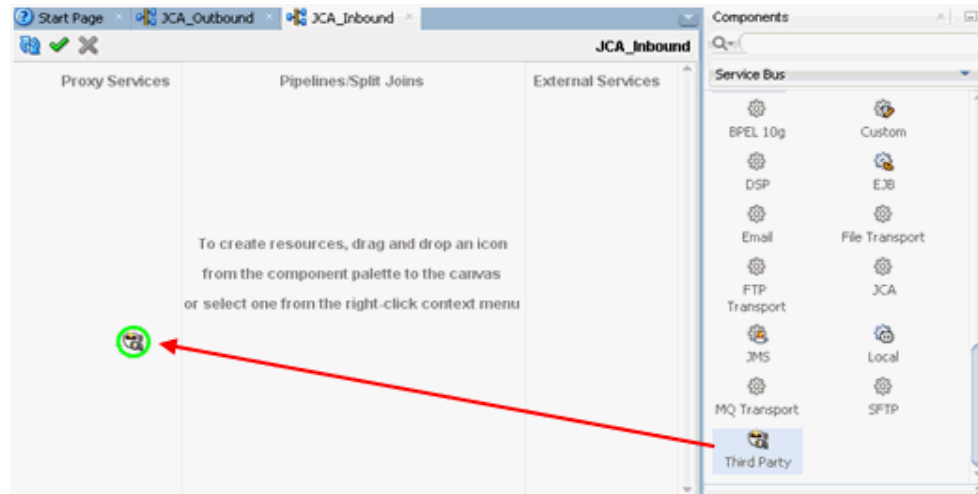
■ 8.2.2.4 項「ルーティング・ルール」の構成

8.2.2.1 サード・パーティ・アダプタ・サービス・コンポーネントの構成

サード・パーティ・アダプタ・サービス・コンポーネントを作成するには、次のステップを実行します。

1. 図 8-31 に示すように、「Service Bus」コンポーネント・ペインから「サード・パーティ」アダプタ・コンポーネントを「プロキシ・サービス」にドラッグ・アンド・ドロップします。

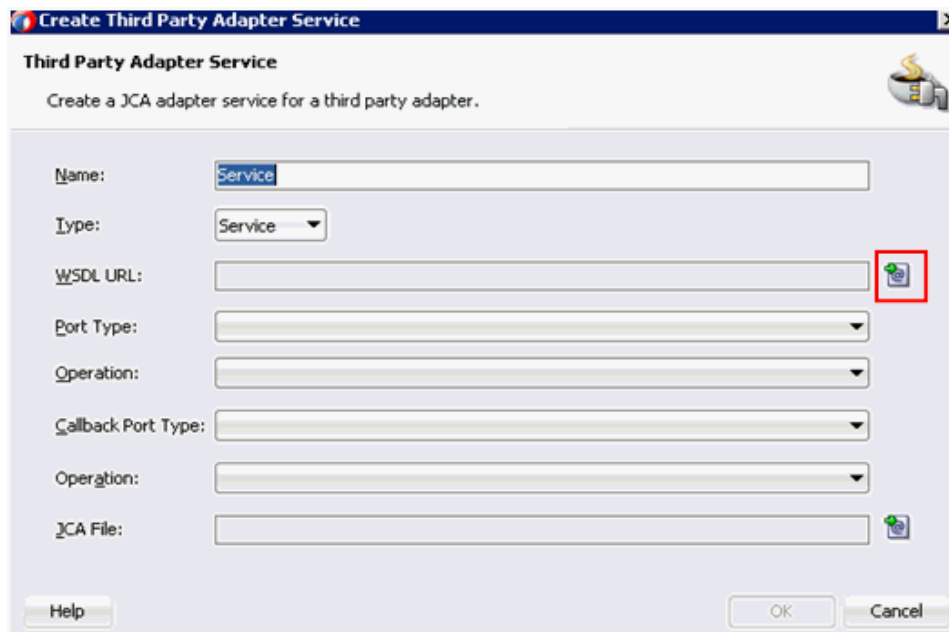
図 8-31 サード・パーティ・アダプタ・サービス・コンポーネント



「サード・パーティ・アダプタ・サービスの作成」ダイアログが表示されます。

2. サード・パーティ・アダプタ・サービスの任意の名前を入力するか、デフォルト値をそのまま使用します。
3. 「タイプ」ドロップダウン・リストで「サービス」が選択されていることを確認します (デフォルト)。
4. 図 8-32 に示すように、「WSDL URL」フィールドの右の「既存の WSDL を検索します。」アイコンをクリックします。

図 8-32 「サード・パーティ・アダプタ・サービス」 ダイアログ



「WSDL チューザ」 ダイアログが表示されます。

5. 「ファイルシステム」 フォルダを選択し、WSDL ディレクトリをブラウザしてインバウンド WSDL ファイルを選択します。

6. 「OK」 をクリックします。

「Service Bus リソースのインポート」 ダイアログが表示されます。

7. 「次へ」 をクリックします。

8. 「構成」 ウィンドウで、「終了」 をクリックします。

「サード・パーティ・アダプタ・サービスの作成」 ダイアログに戻ります。

9. 「JCA ファイル」 フィールドの右の「JCA ファイルを検索します。」 アイコンをクリックします。

変換チューザ・ダイアログが表示されます。

10. WSDL ディレクトリから JCA プロパティ・ファイルを選択します。

11. 「OK」 をクリックします。

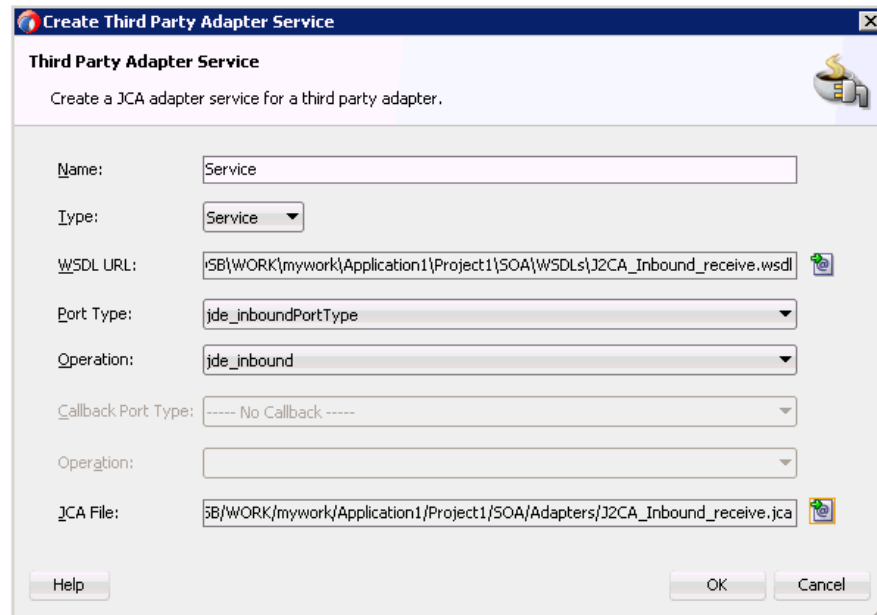
「ファイルのコピー」 メッセージが表示されます。

12. 「はい」 をクリックします。

JCA プロパティ・ファイルのコピーがプロジェクト・フォルダ内に作成されます。

図 8-33 に示すように、「サード・パーティ・アダプタ・サービスの作成」 ダイアログに戻ります。

図 8-33 「サード・パーティ・アダプタ・サービスの作成」ダイアログ



13. 「OK」をクリックします。

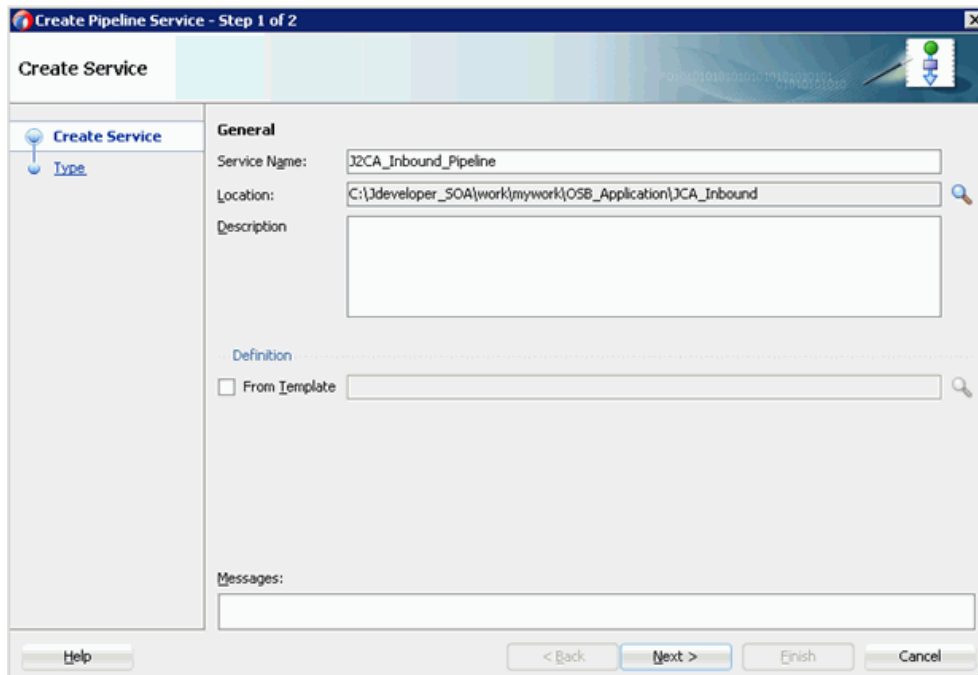
サード・パーティ・アダプタ・サービス・コンポーネントが「プロキシ・サービス」ペインに作成されます。

8.2.2.2 パイプラインの作成

インバウンド・プロキシ・サービスとパイプラインを生成するには、次のステップを実行します。

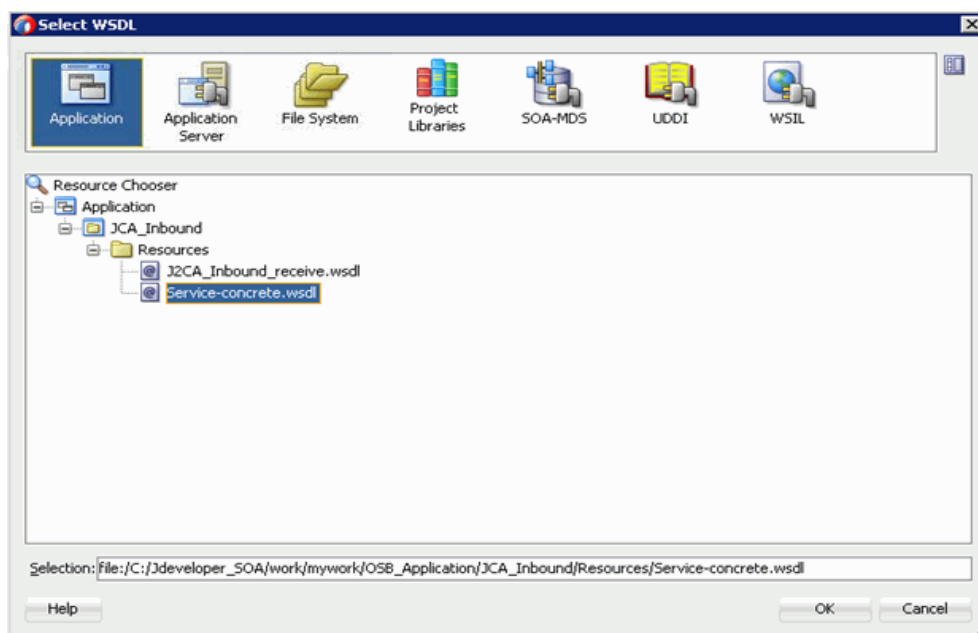
1. 「Service Bus」で、「リソース」をクリックします。
2. 「パイプライン」を「パイプライン/分割結合」ペインにドラッグ・アンド・ドロップします。
3. 図 8-34 に示すように、パイプラインの名前を入力し、「次へ」をクリックします。

図 8-34 「サービスの作成」 ページ



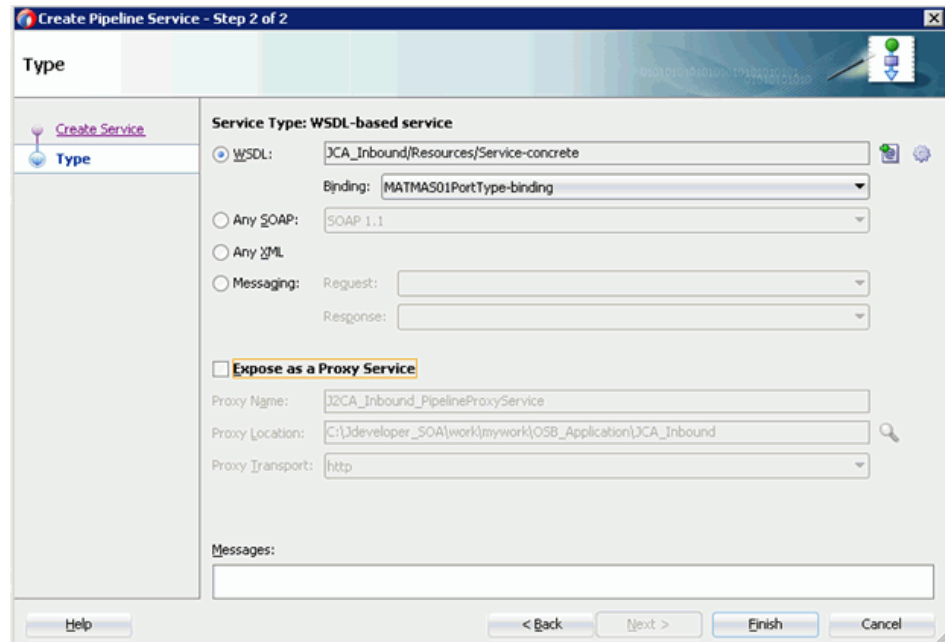
4. 「パイプライン・サービスの作成」 ウィンドウで、「WSDL」 オプションを選択し、WSDL URL をクリックします。
5. 図 8-35 に示すように、WSDL チューザ・ウィンドウで「アプリケーション」を選択し、適切な OSB プロジェクトの「service-concrete.wsdl」を選択して、「OK」をクリックします。

図 8-35 「WSDL の選択」 ページ



6. 図 8-36 に示すように、「プロキシ・サービスとして公開」 チェック・ボックスを選択解除し、「終了」をクリックします。

図 8-36 「タイプ」 ページ



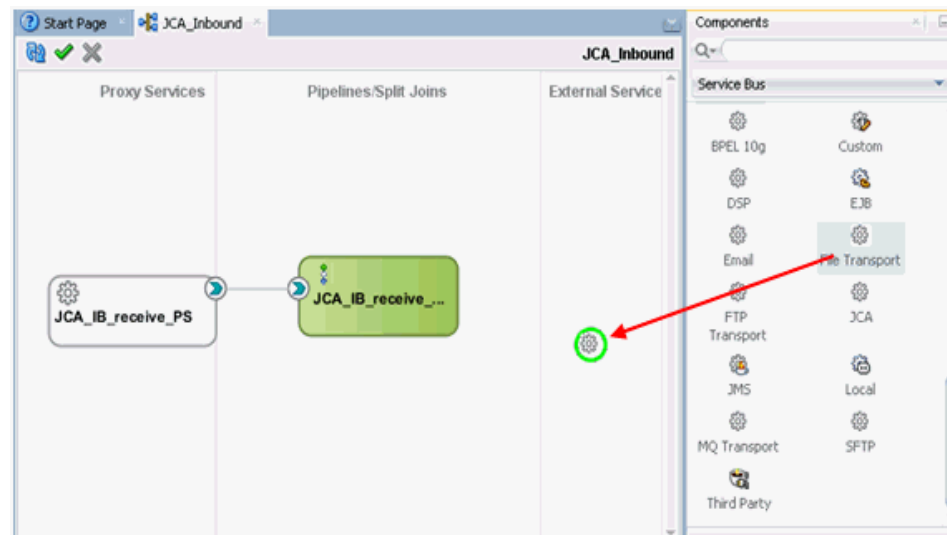
7. 「プロキシ・サービス」を「パイプライン/分割結合」ペインにドラッグ・アンド・ドロップします。

8.2.2.3 ファイル・トランスポート・タイプ・ビジネス・サービスの構成

ファイル・トランスポート・タイプ・ビジネス・サービスを作成するには、次のステップを実行します。

1. 図 8-37 に示すように、「拡張」ペインから「ファイル・トランスポート」コンポーネントを「外部サービス」ペインにドラッグ・アンド・ドロップします。

図 8-37 「ファイル・トランスポート」ノード



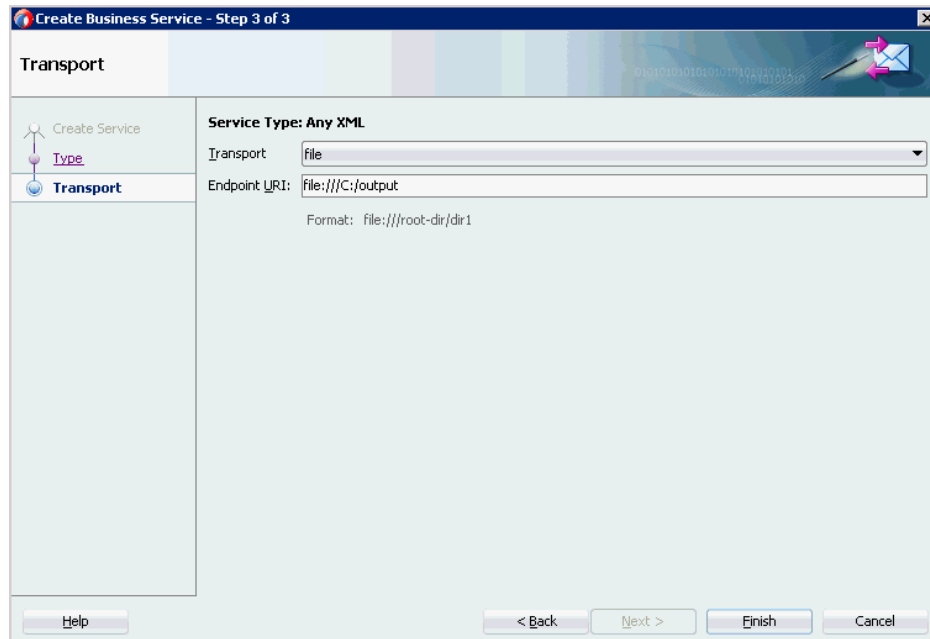
「ビジネス・サービスの作成」ダイアログが表示されます。

2. 「サービス名」フィールドにビジネス・サービスの任意の名前を入力し (「FileOut」など)、「次へ」をクリックします。

表示される「タイプ」ウィンドウで、「任意の XML」オプションがデフォルトで選択されています。

3. 「次へ」をクリックします。
4. 図 8-38 に示すように、表示された「トランスポート」ウィンドウで、「エンドポイント URI」フィールドに出力場所を指定し(「c:\output」など)、「終了」をクリックします。

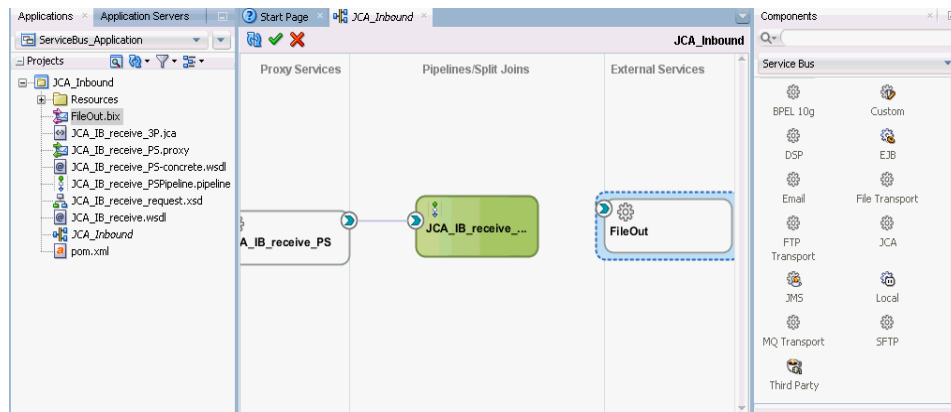
図 8-38 「トランスポート」 ペイン



「FileOut」ビジネス・サービスが作成されます。

5. 図 8-39 に示すように、「FileOut」ビジネス・サービスをダブルクリックします。

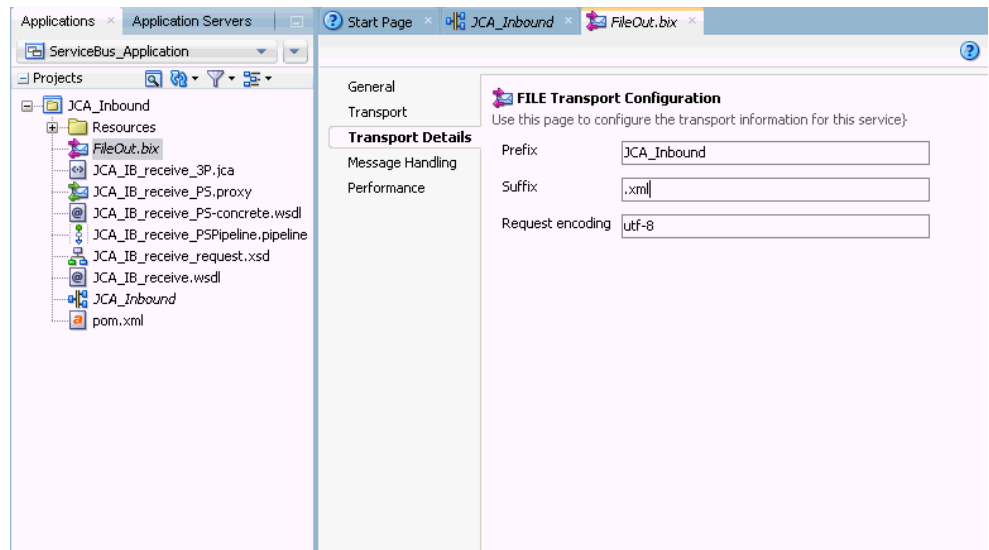
図 8-39 「FileOut」ビジネス・サービス



構成ページが表示されます。

6. 図 8-40 に示すように、「トランスポートの詳細」タブにナビゲートし、「接頭辞」フィールドと「接尾辞」フィールドに値を入力します。

図 8-40 ファイル・トランスポート構成



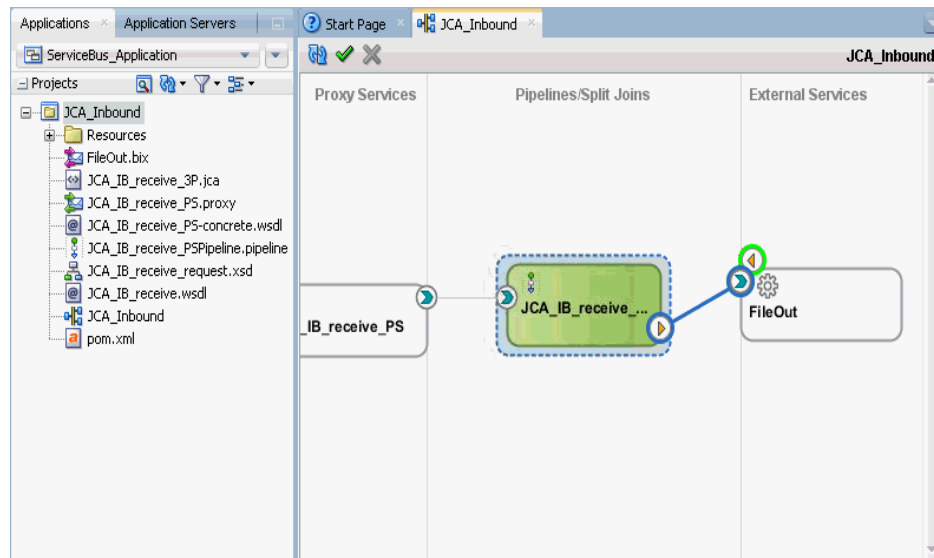
7. 構成ページを保存して閉じます。

8.2.2.4 ルーティング・ルール of 構成

ルーティング・ルールを構成するには、次のステップを実行します。

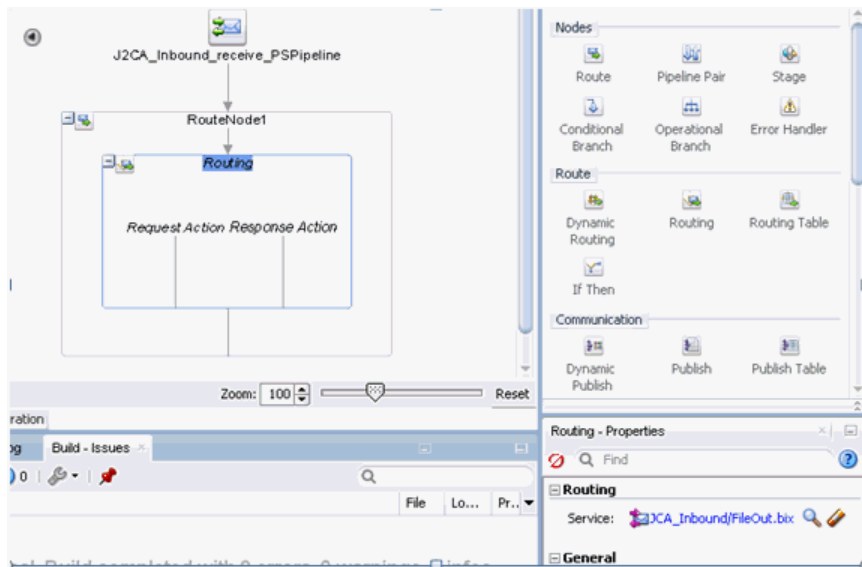
1. 図 8-41 に示すように、パイプライン (「JCA_IB_receive_PSPipeline」など) とファイル・タイプ・ビジネス・サービス (「FileOut」など) の間に接続を作成します。

図 8-41 プロキシと「FileOut」のマッピング



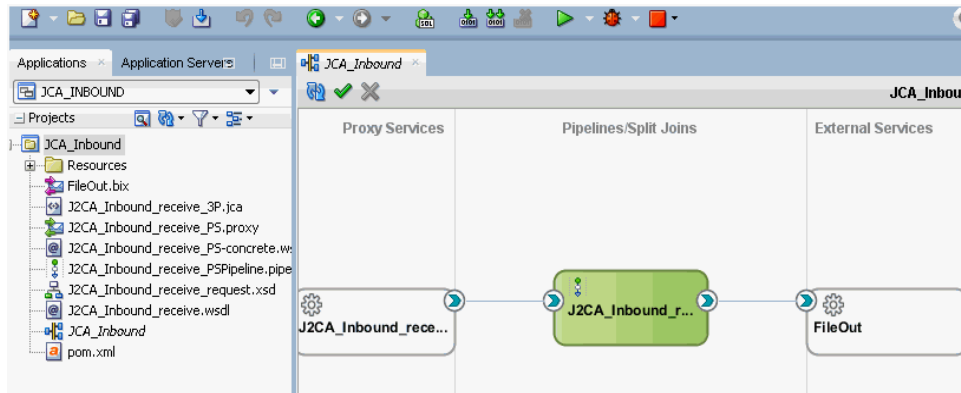
2. パイプライン (「J2CA_Inbound_receive_PSPipeline」など) をダブルクリックします。
3. 図 8-42 に示すように、「ルーティング」ペインをクリックし、ファイル・タイプ・ビジネス・サービス (「FileOut」など) が「サービス」フィールドに適切に構成されていることを確認します。

図 8-42 「ルーティング」ペイン



4. パイプラインの構成ページを保存して閉じます。
5. 図 8-43 に示すように、overview.xml ファイル (「JCA_Inbound」など) をダブルクリックし、メニュー・バーの「すべて保存」をクリックして OSB プロセスを保存します。

図 8-43 すべて保存



8.2.3 OSB インバウンド・プロセスのデプロイ

作成した OSB インバウンド・プロセスをデプロイするには、8-16 ページの 8.1.3 項「OSB アウトバウンド・プロセスのデプロイ」のステップ 1～4 を参照してください。

OSB インバウンド・プロセスが正常にデプロイされた後、J.D. Edwards OneWorld システムからのイベントをトリガーし、構成済の出力場所 (「C:\output」など) で出力が受信されたことを確認します。

イベントのトリガーの詳細は、4-48 ページの 4.5.5 項「J.D. Edwards OneWorld でのイベントのトリガー」を参照してください。

8.3 JDeveloper を使用した OSB アウトバウンド・プロセスの構成 (BSE 構成)

この項では、BSE 構成で Oracle JDeveloper を使用して J.D. Edwards OneWorld システム用に OSB アウトバウンド・プロセスを構成する方法について説明します。

このアウトバウンド・ユース・ケース・シナリオ用のサンプル・プロジェクトが、次に示す Application Adapter のインストール・フォルダに用意されています。

```
<ADAPTER_HOME>\etc\sample\JDEdwards_Samples.zip\JDEdwards_Samples\OSB\BSE\JDEdwards_Sample_BSE_OSB_Outbound_Project
```

この項では、次のトピックについて説明します。

- [8.3.1 項「OSB 用のサービス・バス・アプリケーションの作成」](#)
- [8.3.2 項「OSB アウトバウンド・プロセスの定義」](#)
- [8.3.3 項「OSB アウトバウンド・プロセスのデプロイ」](#)

前提条件

OSB アウトバウンド・プロセスを設計する前に、アプリケーション・エクスプローラを使用して各 WSDL ファイルを生成する必要があります。詳細は、4-53 ページの [4.6.1 項「Web サービスを使用したリクエスト / レスポンス型サービス用の WSDL ファイルの生成」](#) を参照してください。

8.3.1 OSB 用のサービス・バス・アプリケーションの作成

OSB アウトバウンド・プロセスを構成するには、OSB 用のサービス・バス・アプリケーションを作成する必要があります。詳細は、8-2 ページの [8.1.1 項「OSB 用のサービス・バス・アプリケーションの作成」](#) を参照してください。

8.3.2 OSB アウトバウンド・プロセスの定義

この項では、OSB アウトバウンド・プロセスの定義方法について説明します。次のトピックが含まれています。

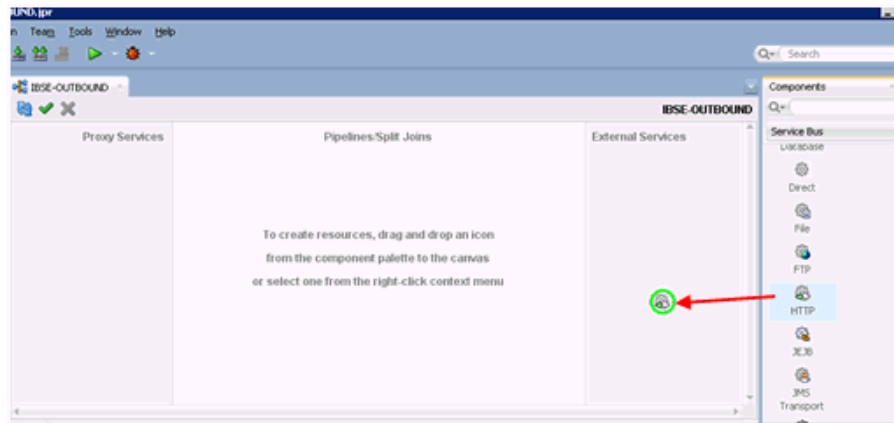
- [8.3.2.1 項「WSDL ベースのビジネス・サービスの構成」](#)
- [8.3.2.2 項「プロキシ・サービスとパイプラインの作成」](#)
- [8.3.2.3 項「ファイル・トランスポート・タイプ・ビジネス・サービスの構成」](#)
- [8.3.2.4 項「ルーティング・ルールの構成」](#)

8.3.2.1 WSDL ベースのビジネス・サービスの構成

WSDL ベースのビジネス・サービスを構成するには、次のステップを実行します。

1. [図 8-44](#) に示すように、「テクノロジー」コンポーネント・ペインから「HTTP」コンポーネントを「外部サービス」領域にドラッグ・アンド・ドロップします。

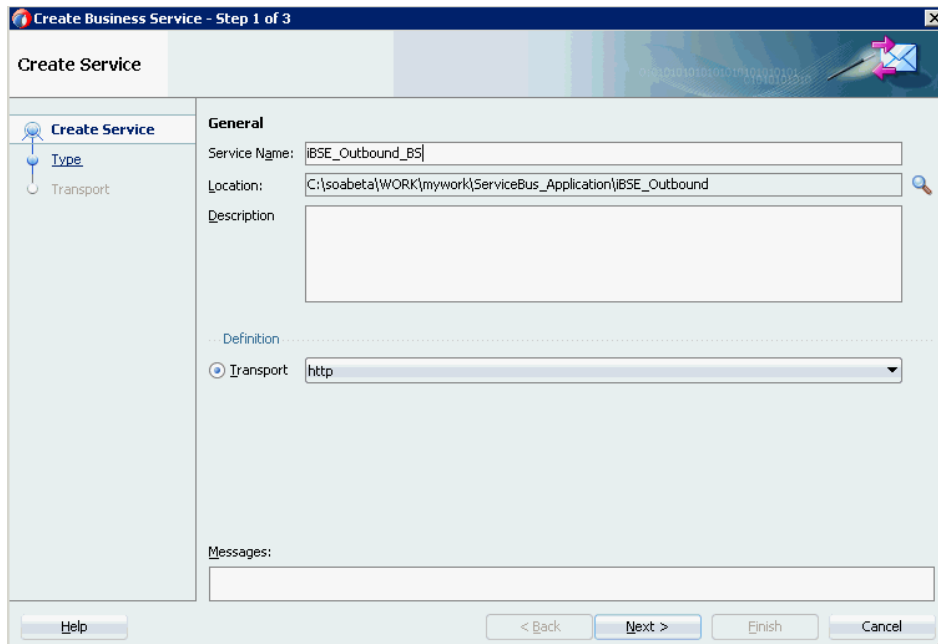
図 8-44 HTTP コンポーネント



「ビジネス・サービスの作成」ウィンドウが表示されます。

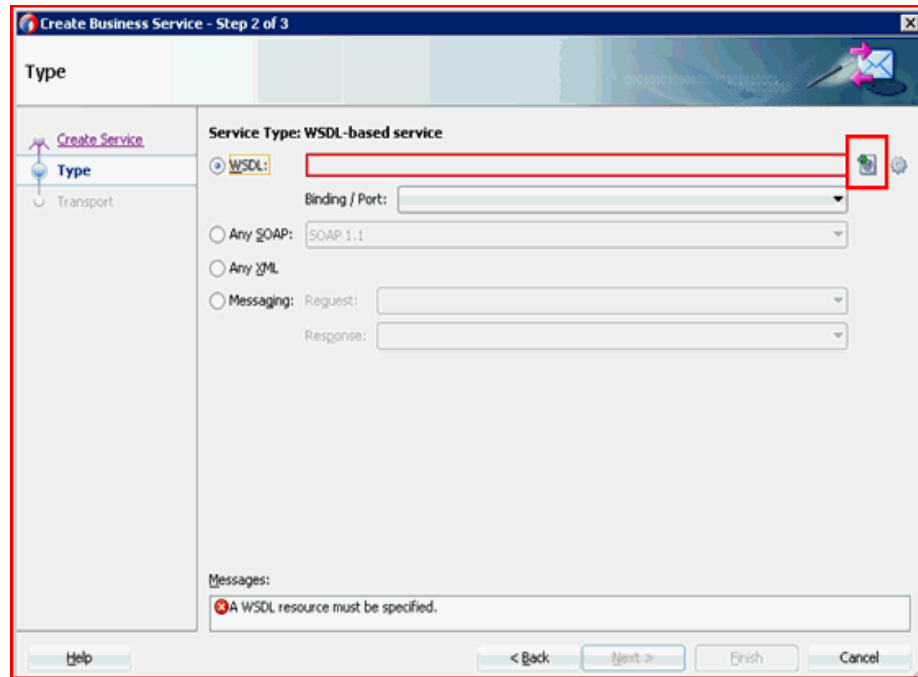
2. 図 8-45 に示すように、「サービス名」フィールドにビジネス・サービスの任意の名前を入力し、「次へ」をクリックします。

図 8-45 ビジネス・サービスの作成



3. 図 8-46 に示すように、表示された「サービス・タイプ」ウィンドウで、「WSDL」オプションを選択し、「WSDL の選択」アイコンをクリックします。

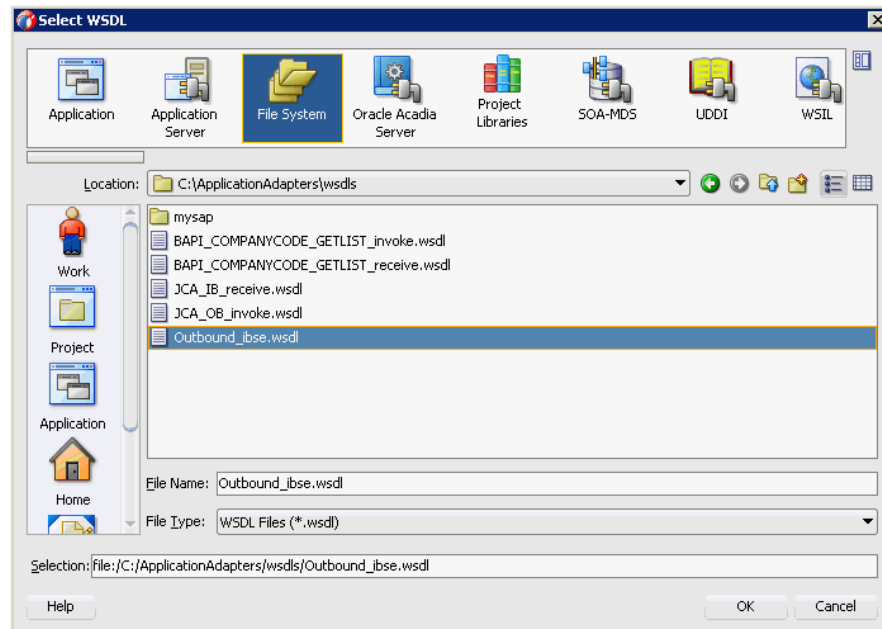
図 8-46 「タイプ」 ペイン



「WSDL の選択」 ウィンドウが表示されます。

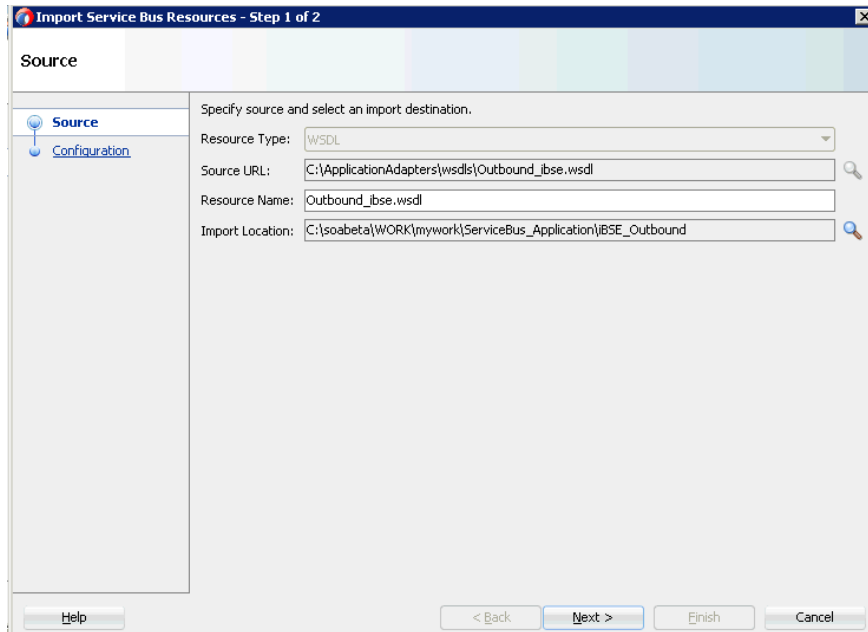
4. 図 8-47 に示すように、「ファイルシステム」フォルダ・アイコンを選択し、WSDL の場所をブラウズして iBSE WSDL ファイルを選択し、「OK」をクリックします。

図 8-47 「WSDL の選択」 ウィンドウ



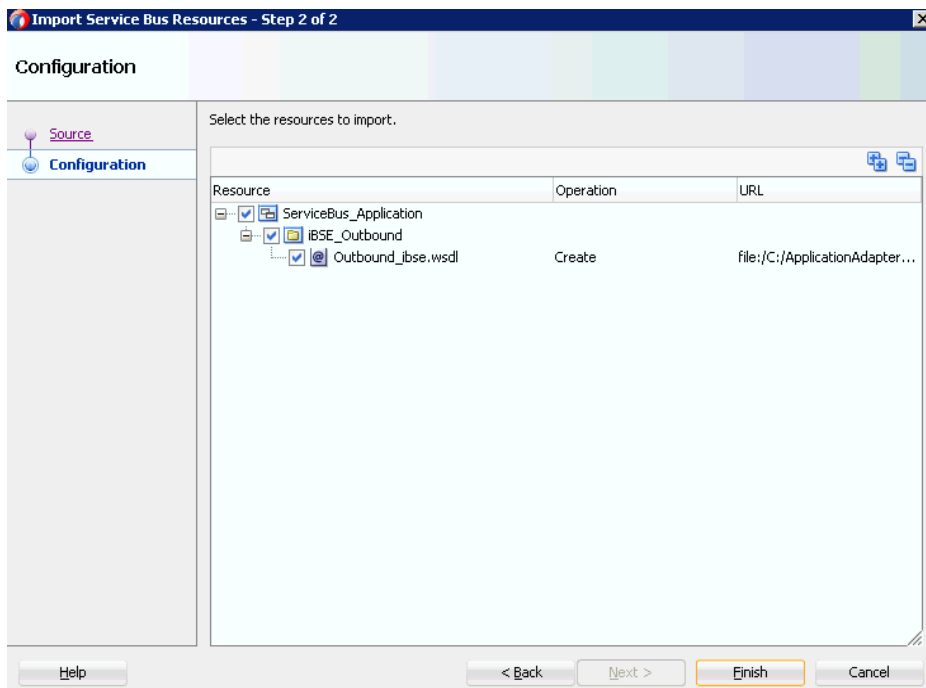
5. 図 8-48 に示すように、表示された「ソース」ペインで、「次へ」をクリックします。

図 8-48 「ソース」 ペイン



6. 図 8-49 に示すように、表示された「構成」ペインで、「終了」をクリックします。

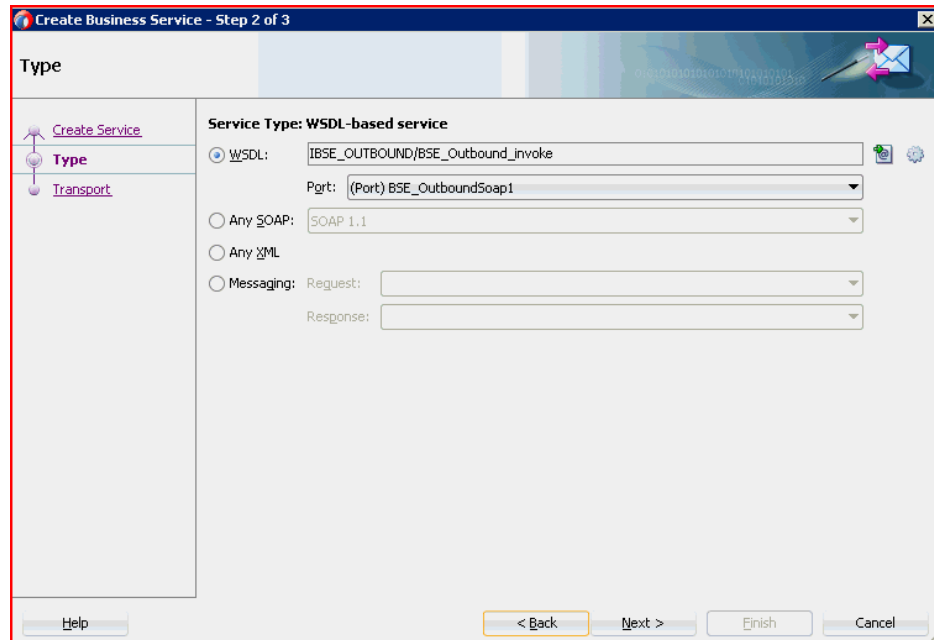
図 8-49 「構成」 ペイン



「ビジネス・サービスの作成」ウィンドウに戻ります。

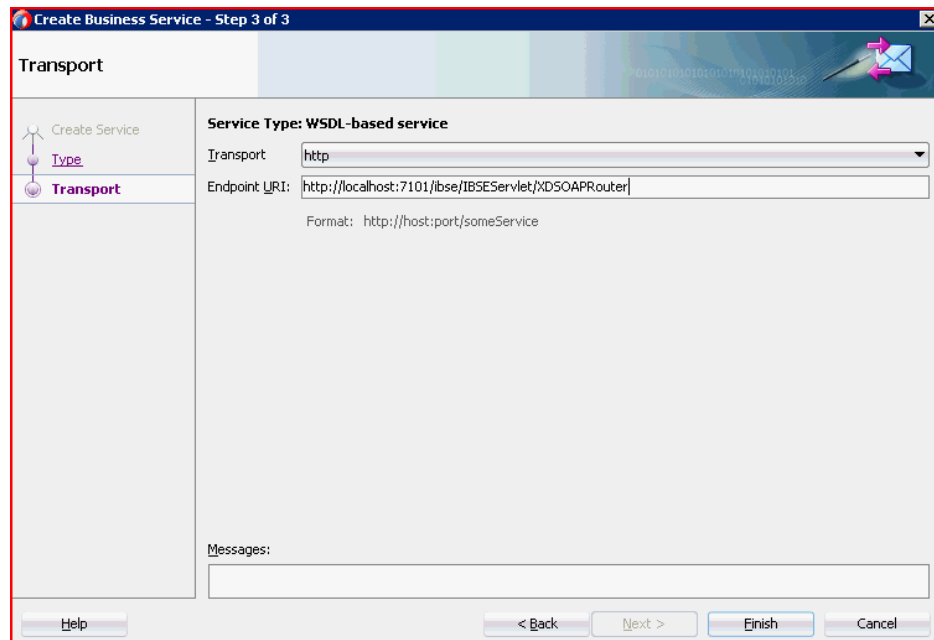
7. 図 8-50 に示すように、表示された「タイプ」ペインで、「次へ」をクリックします。

図 8-50 「タイプ」 ペイン



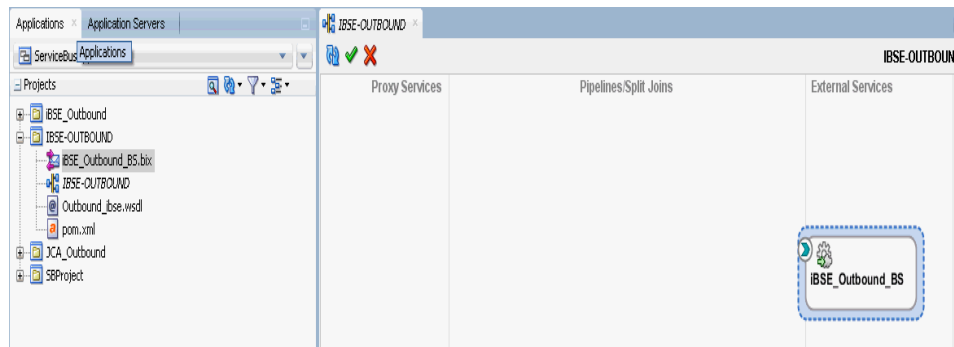
8. 図 8-51 に示すように、表示された「トランスポート」ウィンドウで、ホスト名やポート番号が異なる場合には「エンドポイント URI」フィールドを変更し、「終了」をクリックします。

図 8-51 「トランスポート」 ペイン



- 図 8-52 に示すように、ビジネス・サービスが作成され、「外部サービス」ペインに表示されます。

図 8-52 「外部サービス」 ペイン

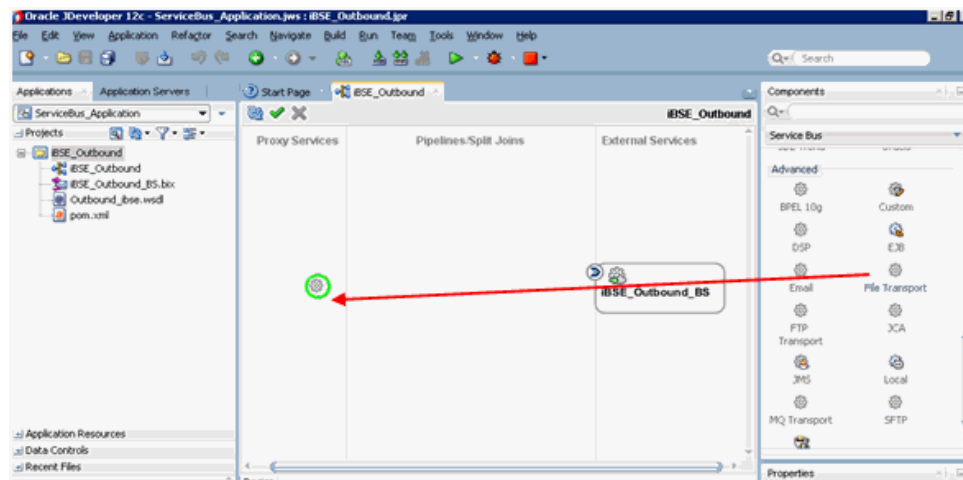


8.3.2.2 プロキシ・サービスとパイプラインの作成

プロキシ・サービスとパイプラインを作成するには、次のステップを実行します。

1. 図 8-53 に示すように、「拡張」コンポーネント・ペインから「ファイル・トランスポート」コンポーネントを「プロキシ・サービス」ペインにドラッグ・アンド・ドロップします。

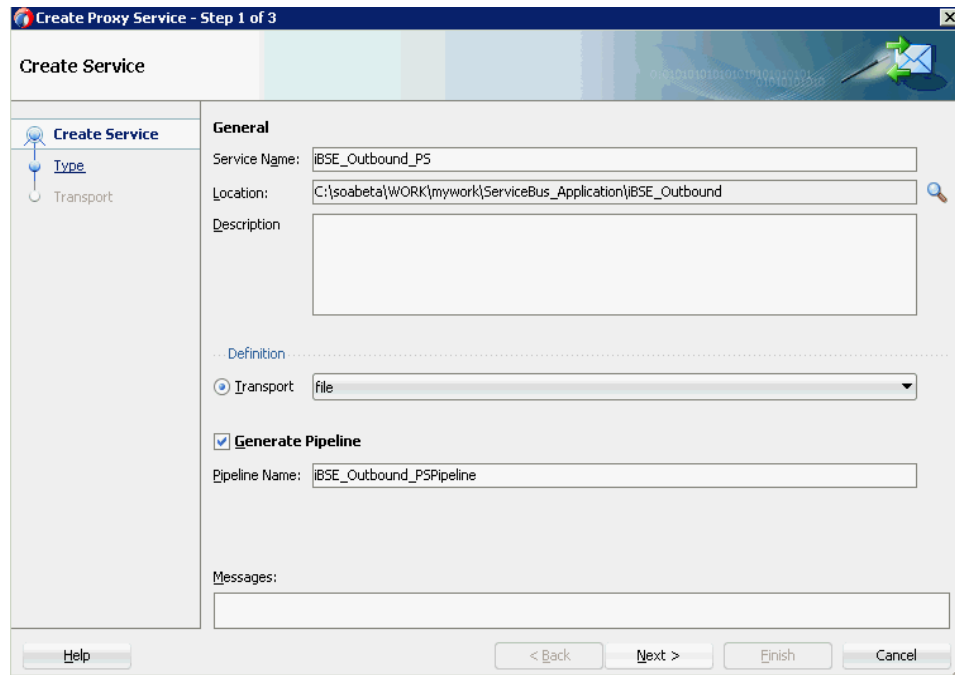
図 8-53 「ファイル・トランスポート」コンポーネント



「プロキシ・サービスの作成」ペインが表示されます。

2. 「サービス名」フィールドにプロキシ・サービスの任意の名前を入力します (「JCA_Outbound_PS」など)。デフォルトで、「パイプラインの生成」が選択されています。
3. 図 8-54 に示すように、「次へ」をクリックします。

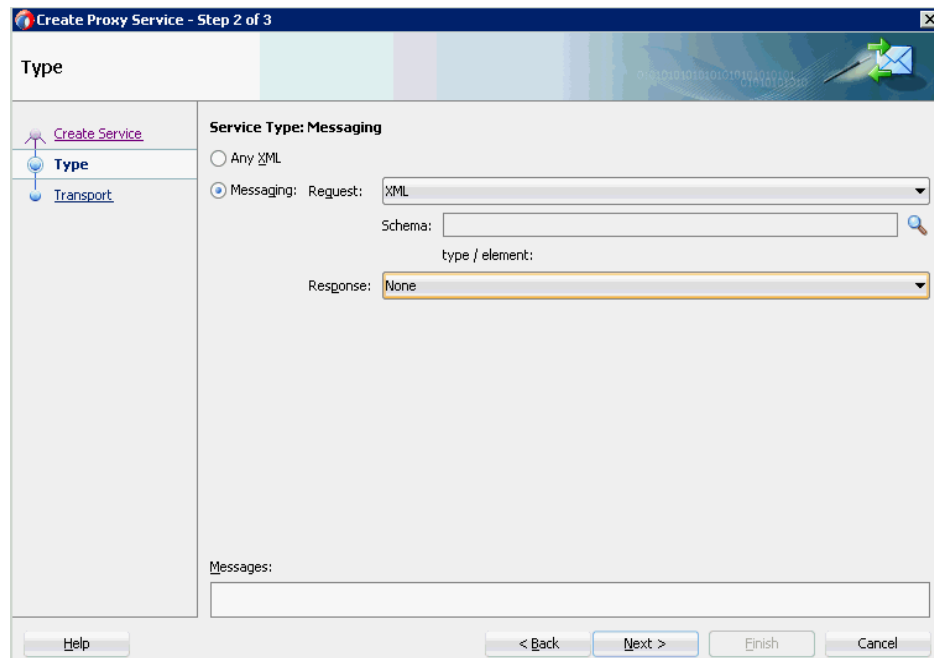
図 8-54 「サービスの作成」 ペイン



「タイプ」 ペインが表示されます。

4. 図 8-55 に示すように、「メッセージング」オプションを選択し、「リクエスト」を「XML」に、「レスポンス」を「なし」に設定して、「次へ」をクリックします。

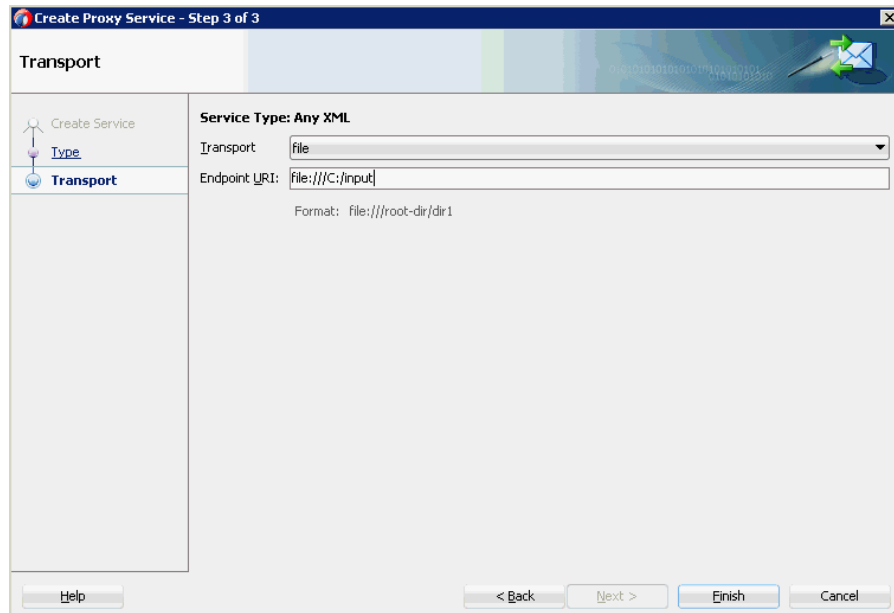
図 8-55 「タイプ」 ペイン



「トランスポート」ウィンドウが表示されます。

5. 図 8-56 に示すように、「エンドポイント URI」フィールドに入力場所を指定し（「c:/input」など）、「終了」をクリックします。

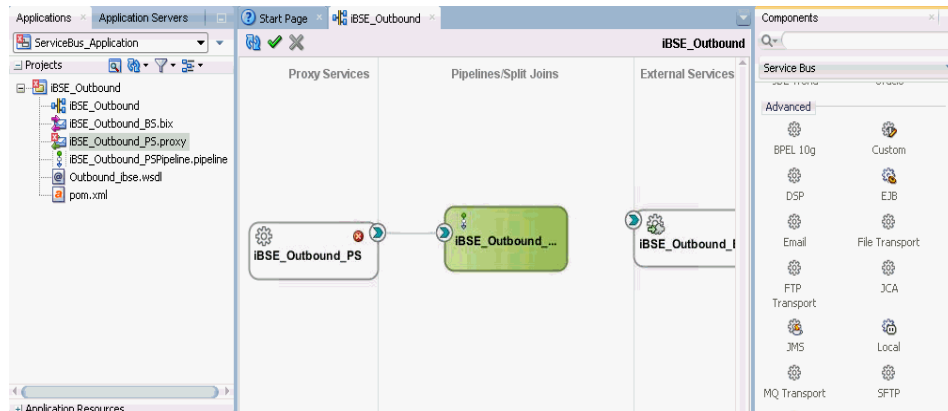
図 8-56 「トランスポート」 ウィンドウ



プロキシ・サービスとパイプラインが作成され、表示されます。

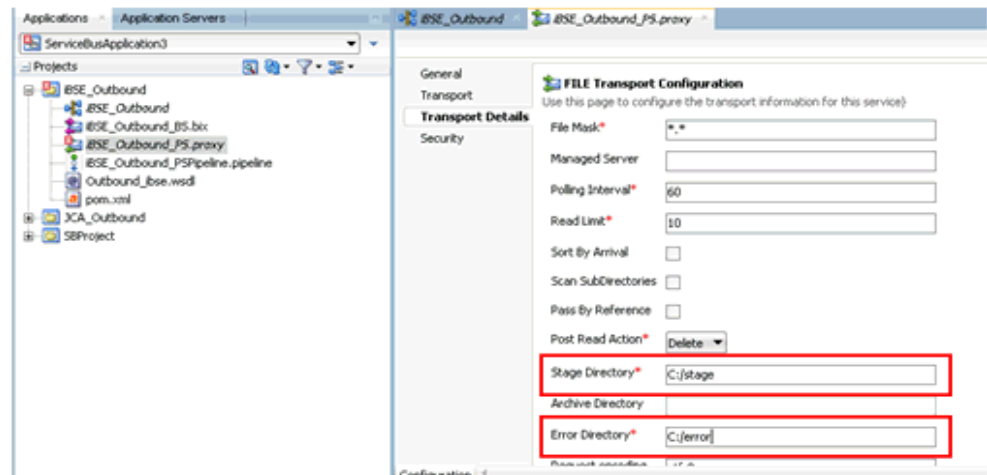
- 図 8-57 に示すように、作成されたプロキシ・サービス (「iBSE_Outbound_PS」など) をダブルクリックします。

図 8-57 プロキシ・サービスの編集



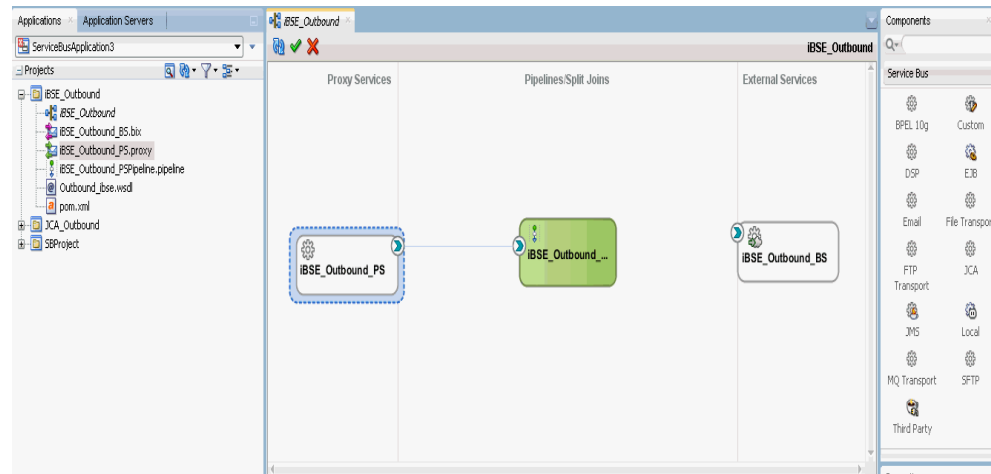
- 図 8-58 に示すように、表示されたプロキシ・サービスの構成ページで、「トランスポートの詳細」を選択し、「ステージ・ディレクトリ」と「エラー・ディレクトリ」の値を入力します。

図 8-58 ファイル・トランスポート構成



8. プロキシ・サービスの構成ページを保存して閉じます。
9. overview.xml ファイル (「iBSE_Outbound」など) をダブルクリックします。
☒ 図 8-59 に示すように、プロキシ・サービスが更新されて表示されます。

図 8-59 プロキシ・サービス

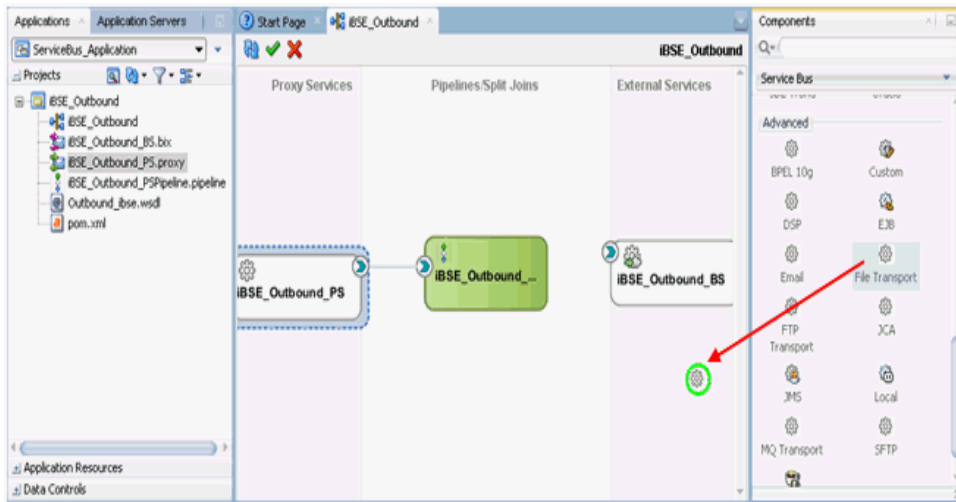


8.3.2.3 ファイル・トランスポート・タイプ・ビジネス・サービスの構成

ファイル・トランスポート・タイプ・ビジネス・サービスを作成するには、次のステップを実行します。

1. ☒ 図 8-60 に示すように、「拡張」ペインから「ファイル・トランスポート」コンポーネントを「外部サービス」ペインにドラッグ・アンド・ドロップします。

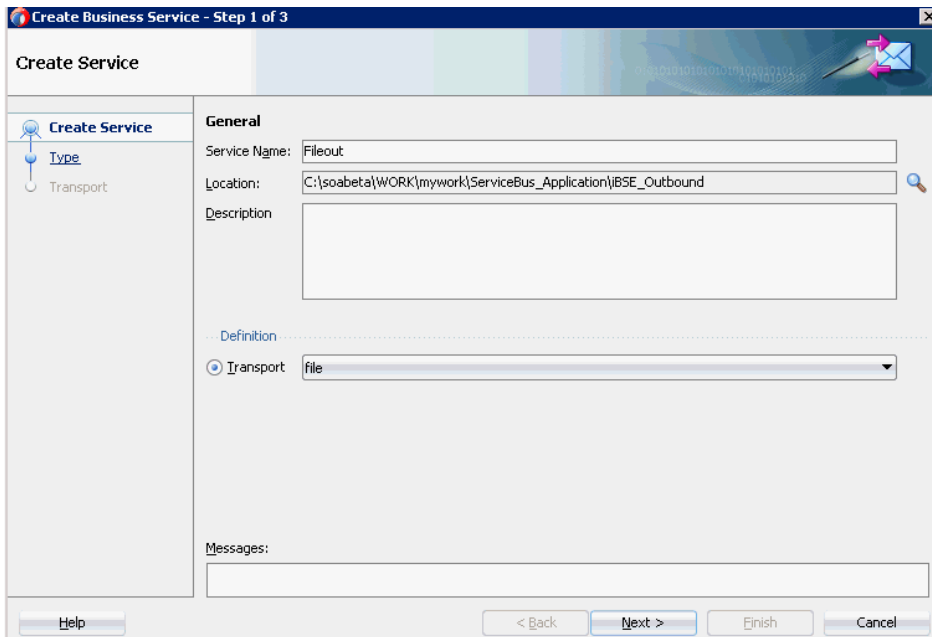
図 8-60 「ファイル・トランスポート」 コンポーネント



「ビジネス・サービスの作成」 ダイアログが表示されます。

2. 図 8-61 に示すように、「サービス名」フィールドにビジネス・サービスの任意の名前を入力し(「FileOut」など)、「次へ」をクリックします。

図 8-61 「サービスの作成」 ペイン



「タイプ」 ペインが表示されます。「任意の XML」 オプションがデフォルトで選択されています。

3. 「次へ」 をクリックします。
「トランスポート」 ペインが表示されます。
4. 図 8-62 に示すように、「エンドポイント URI」 フィールドに出力場所を指定し(「c:/output」 など)、「終了」 をクリックします。

図 8-62 「トランスポート」 ペイン

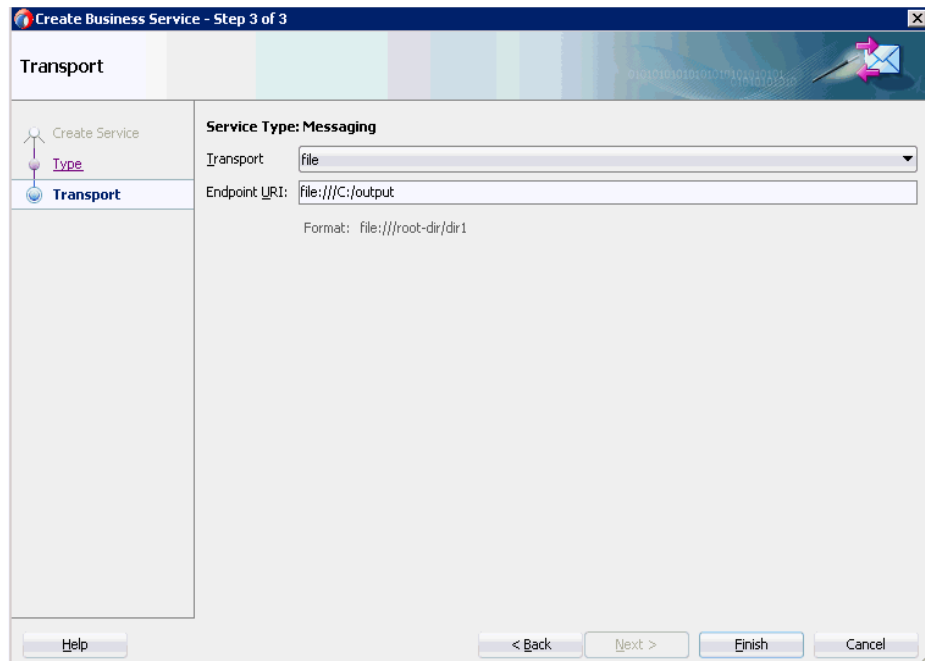
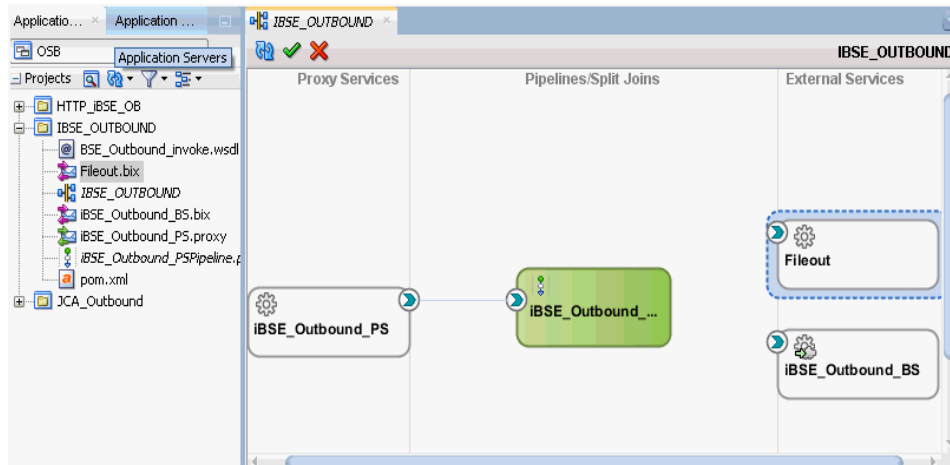


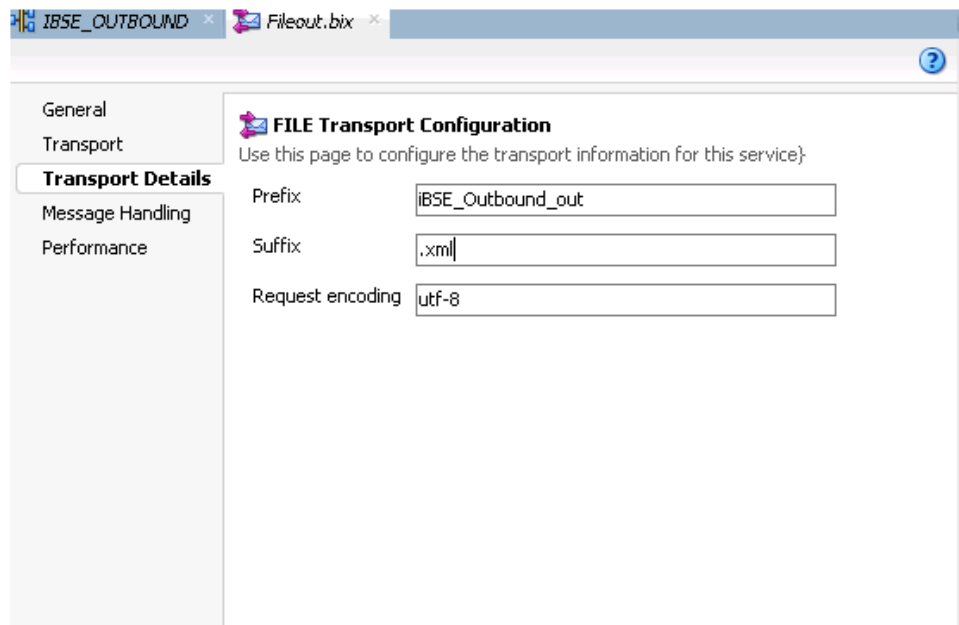
図 8-63 に示すように、ファイル・トランスポート・ビジネス・サービスの「FileOut」が作成され、表示されます。

図 8-63 「FileOut」 ビジネス・サービス



5. 作成されたビジネス・サービスの「FileOut」をダブルクリックし、図 8-64 に示すように、「トランスポートの詳細」タブの「接頭辞」フィールドと「接尾辞」フィールドに値を入力します。

図 8-64 トランスポートの詳細



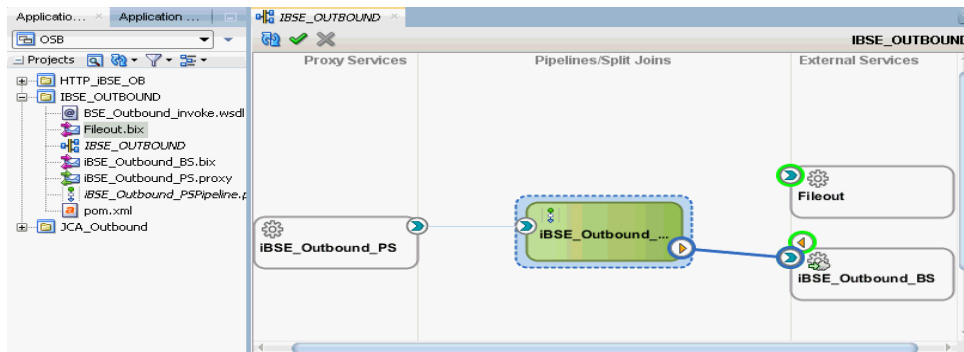
6. 構成ページを保存して閉じ、overview.xml (「iBSE_Outbound」など) をダブルクリックします。

8.3.2.4 ルーティング・ルール構成

ルーティング・ルールを構成するには、次のステップを実行します。

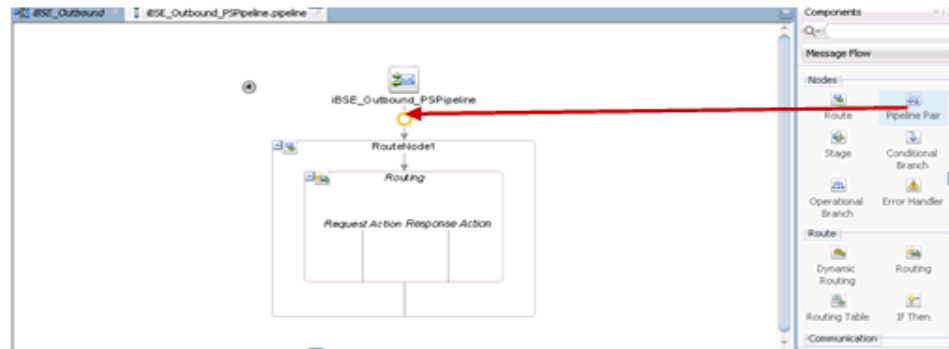
1. 図 8-65 に示すように、パイプライン・コンポーネント (「iBSE_Outbound_PSPipeline」など) と WSDL ベースのビジネス・サービス (「iBSE_Outbound_BS」など) の間に接続を作成します。

図 8-65 パイプライン・コンポーネント



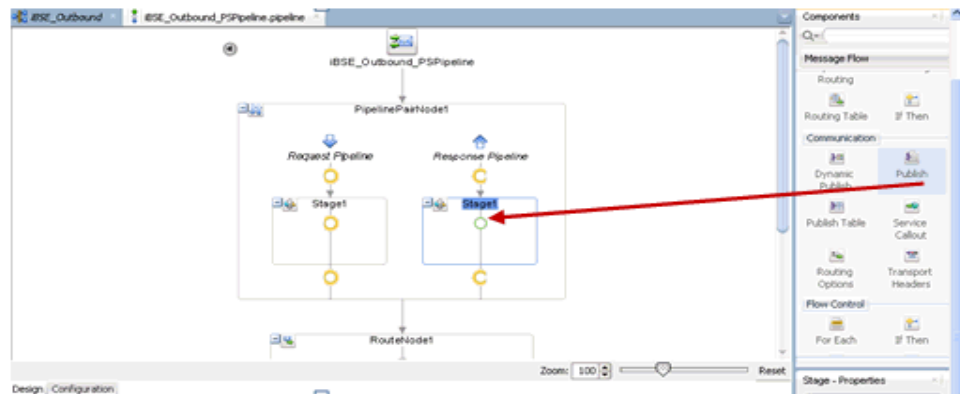
2. 「パイプライン/分割結合」ペインでパイプライン・コンポーネント (「iBSE_Outbound_PSPipeline」など) をダブルクリックします。
3. 図 8-66 に示すように、「ノード」ペインから「パイプライン・ペア」ノードをパイプライン (「iBSE_Outbound_PSPipeline」など) と RouteNode1 の間の領域にドラッグ・アンド・ドロップします。

図 8-66 「パイプライン・ペア」ノード



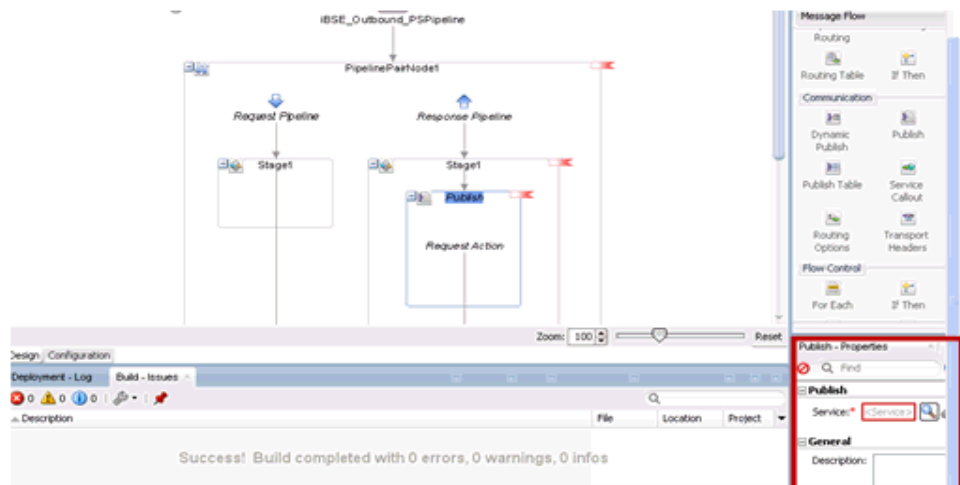
4. 図 8-67 に示すように、「通信」ペインから「パブリッシュ」ノードを「レスポンス・パイプライン」の「ステージ1」の下の領域にドラッグ・アンド・ドロップします。

図 8-67 「パブリッシュ」ノード



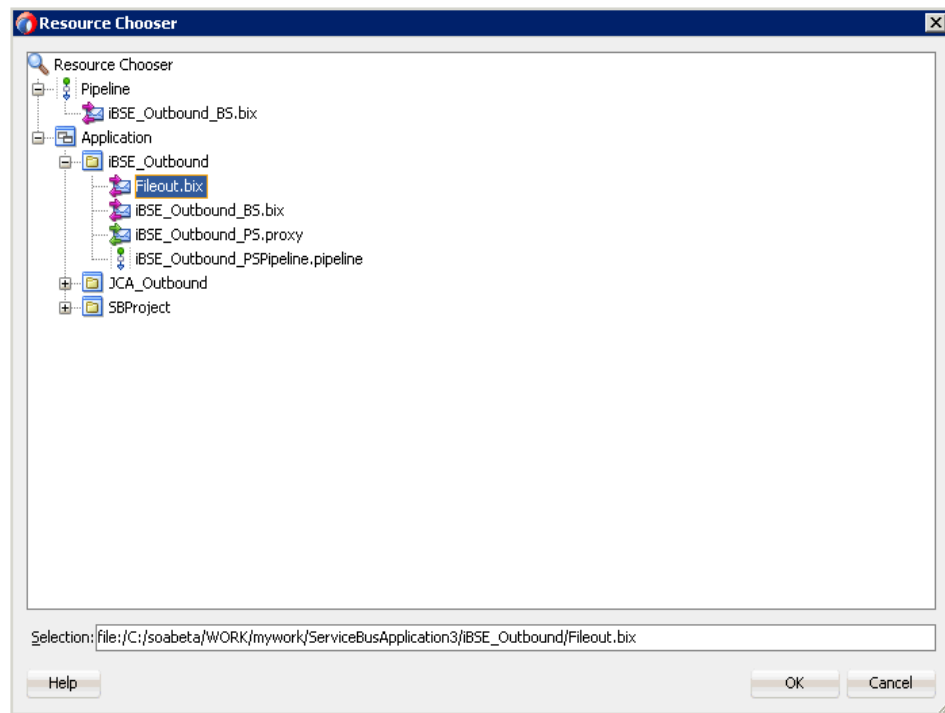
5. 図 8-68 に示すように、パブリッシュのプロパティの右ペインで「サービス」フィールドの右の参照アイコンをクリックします。

図 8-68 「参照」アイコン



- 図 8-69 に示すように、表示された「リソース・チューザ」ウィンドウで、「Fileout.bix」ファイル・トランスポート・ビジネス・サービスを選択し、「OK」をクリックします。

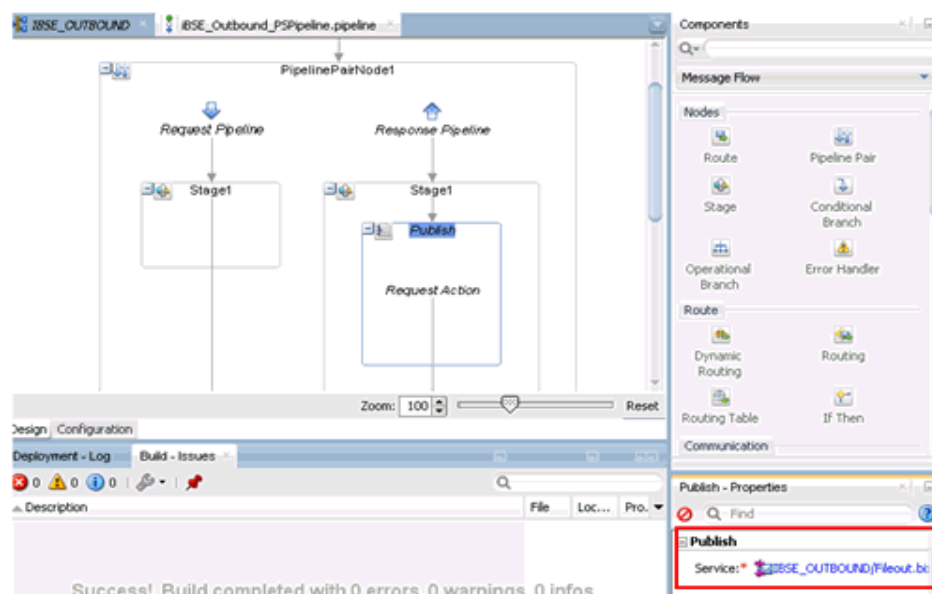
図 8-69 リソース・チューザ



パイプラインの構成ページに戻ります。

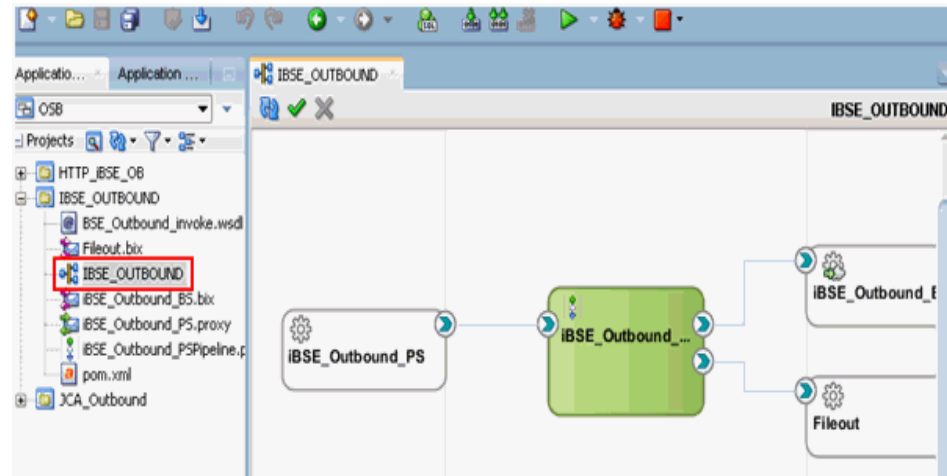
図 8-70 に示すように、選択したサービスが右ペインの「パブリッシュ」ペインで構成されます。

図 8-70 「パブリッシュ」ペイン



7. パイプラインの構成ページを保存して閉じます。
8. 図 8-71 に示すように、overview.xml ファイル (「iBSE_Outbound」など) をダブルクリックし、メニュー・バーの「すべて保存」をクリックして OSB プロセスを保存します。

図 8-71 「すべて保存」アイコン



8.3.3 OSB アウトバウンド・プロセスのデプロイ

作成した OSB アウトバウンド・プロセスをデプロイし、入力 XML ドキュメントを起動するには、8.1.3 項「OSB アウトバウンド・プロセスのデプロイ」を参照してください。

8.4 JDeveloper を使用した JMS インバウンド・プロセスの構成 (J2CA 構成)

この項では、J2CA 構成で Oracle JDeveloper を使用して J.D. Edwards OneWorld システム用に JMS インバウンド・プロセスを構成する方法について説明します。

1. JMS プロセスを設計する前に、アプリケーション・エクスプローラを使用して、各 WSDL ファイルを生成する必要があります。詳細は、4-35 ページの 4.5.1 項「イベント統合のための WSDL の生成」を参照してください。
2. Oracle JDeveloper を起動し、OSB 用のサービス・バス・アプリケーションを作成します。詳細は、8-2 ページの 8.1.1 項「OSB 用のサービス・バス・アプリケーションの作成」を参照してください。
3. サード・パーティ・アダプタ・サービス・コンポーネントを作成します。詳細は、8-19 ページの 8.2.2.1 項「サード・パーティ・アダプタ・サービス・コンポーネントの構成」を参照してください。
4. JCA バインド・ファイルからプロキシ・サービスとパイプラインを作成します。詳細は、8-21 ページの 8.2.2.2 項「パイプラインの作成」を参照してください。
5. JMS トランスポート・ビジネス・サービスを作成し、次のステップを実行します。
 - a. 図 8-72 に示すように、「テクノロジー」コンポーネント・ペインから「JMS トランスポート」コンポーネントを「外部サービス」ペインにドラッグ・アンド・ドロップします。

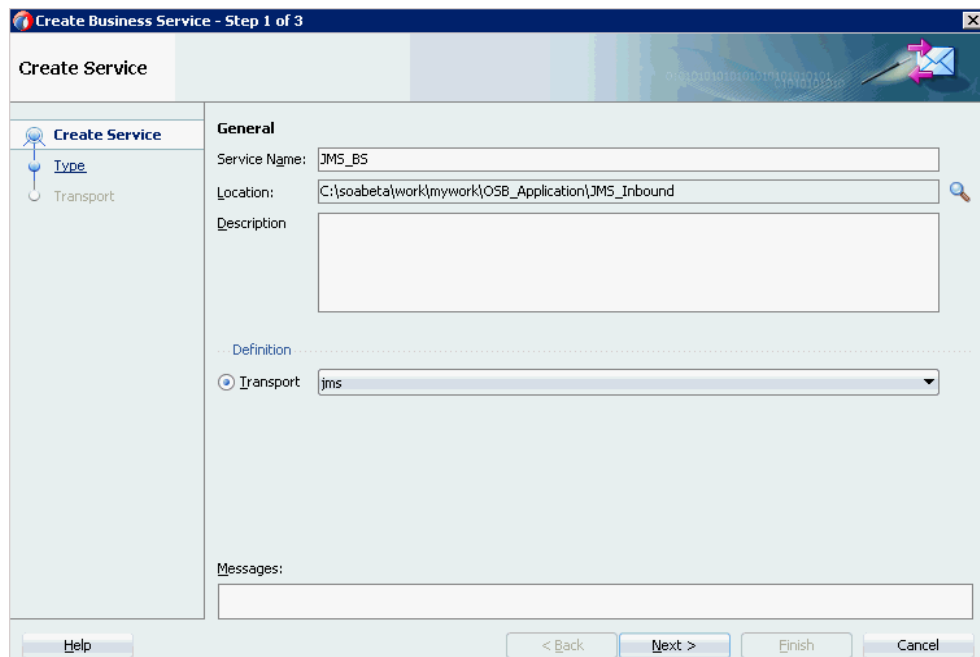
図 8-72 「JMS トランスポート」 コンポーネント



「ビジネス・サービスの作成」ダイアログが表示されます。

- b. 図 8-73 に示すように、「サービス名」フィールドにビジネス・サービスの任意の名前を入力し (「JMS_BS」など)、「次へ」をクリックします。

図 8-73 「サービスの作成」 ペイン

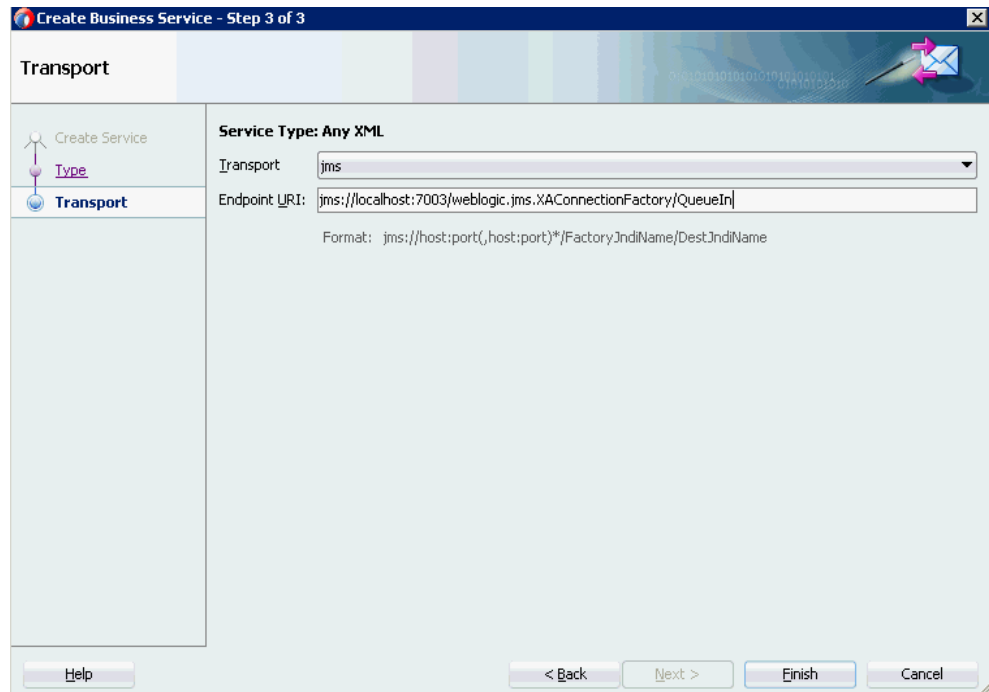


- c. 表示された「タイプ」ウィンドウで、「任意の XML」を選択し、「次へ」をクリックします。

「トランスポート」ウィンドウが表示されます。

- d. 図 8-74 に示すように、「エンドポイント URI」フィールドで「DestJndiName」を「QueueIn」に置き換えて適切なホスト名とポート番号に変更し (「jms://localhost:7003/weblogic.jms.XAConnectionFactory/QueueIn」など)、「終了」をクリックします。

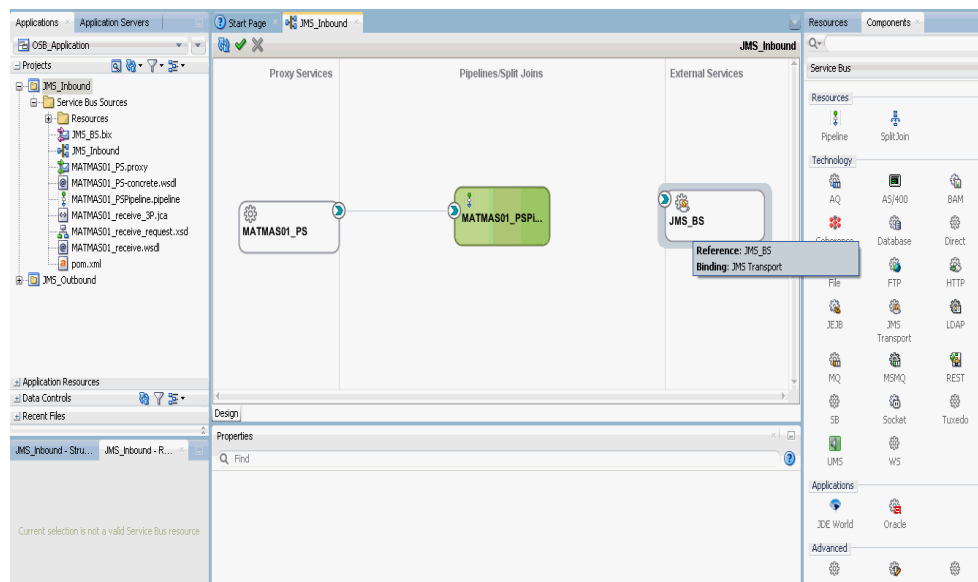
図 8-74 「トランスポート」 ウィンドウ



JMS ビジネス・サービスが作成され、表示されます。

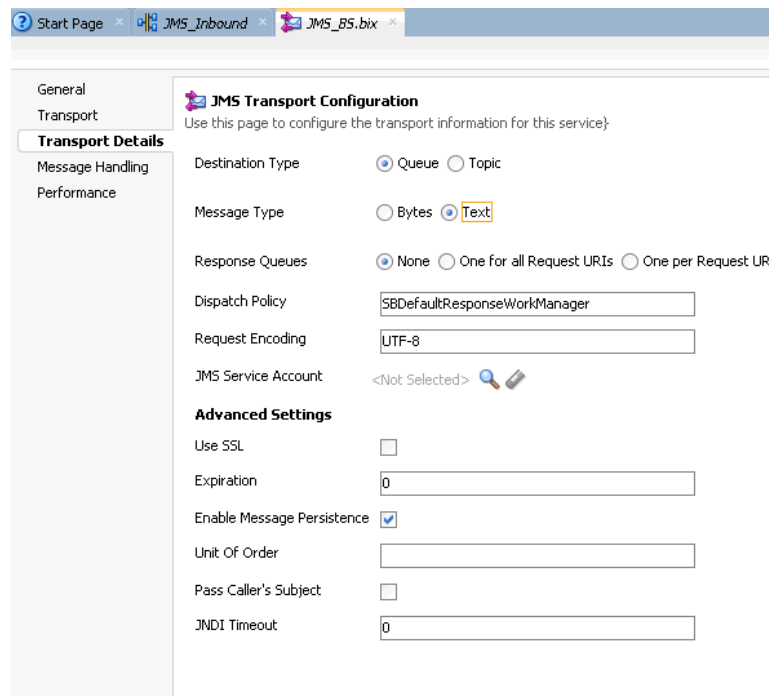
- e. 図 8-75 に示すように、「JMS_BS」をダブルクリックします。

図 8-75 JMS ビジネス・サービス



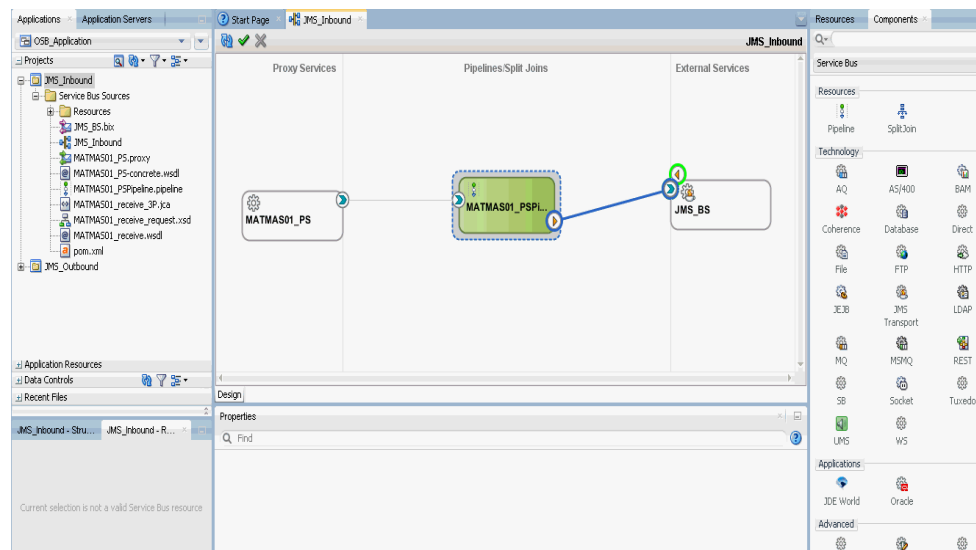
- f. 図 8-76 に示すように、表示されたビジネス・サービスの構成ページにおいて、「トランスポートの詳細」タブで次のパラメータを指定します。

図 8-76 JMS トランスポート構成



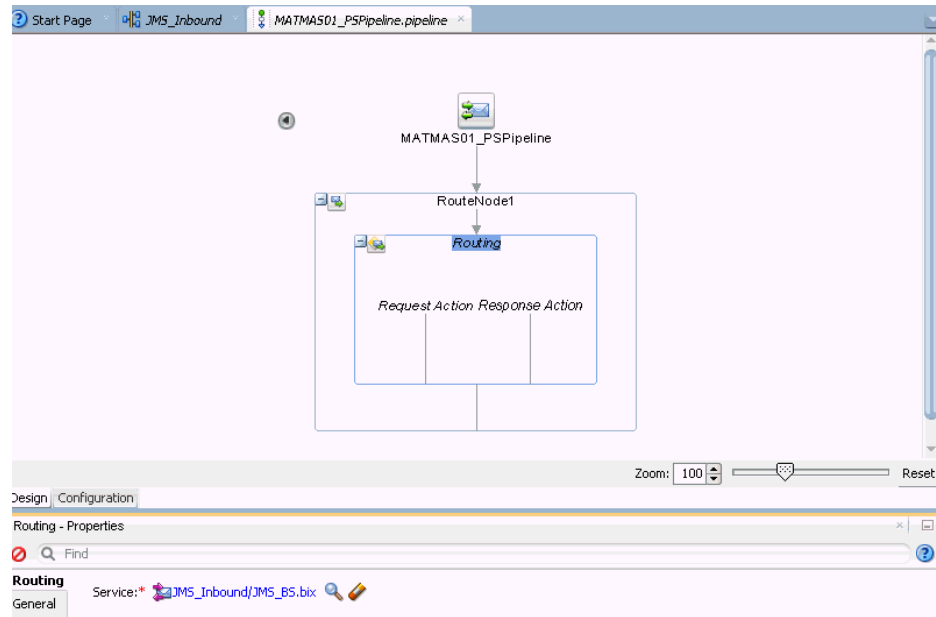
- g. 「宛先タイプ」セクションで、「キュー」を選択します。
- h. 「メッセージ・タイプ」セクションで、「テキスト」を選択します。
6. ビジネス・サービスの構成ページを保存して閉じます。
7. 図 8-77 に示すように、パイプライン (「xxxx_PSPipeline」など) と JMS ビジネス・サービス (「JMS_BS」など) の間に接続を作成します。

図 8-77 「構成」ページ



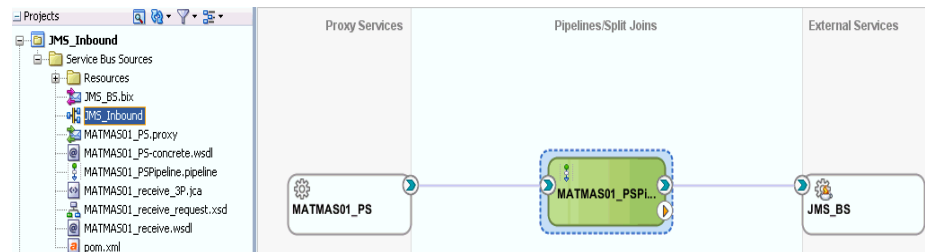
8. パイプラインをダブルクリックします。
図 8-78 に示すように、パイプラインの構成ページが表示されます。

図 8-78 パイプラインの構成



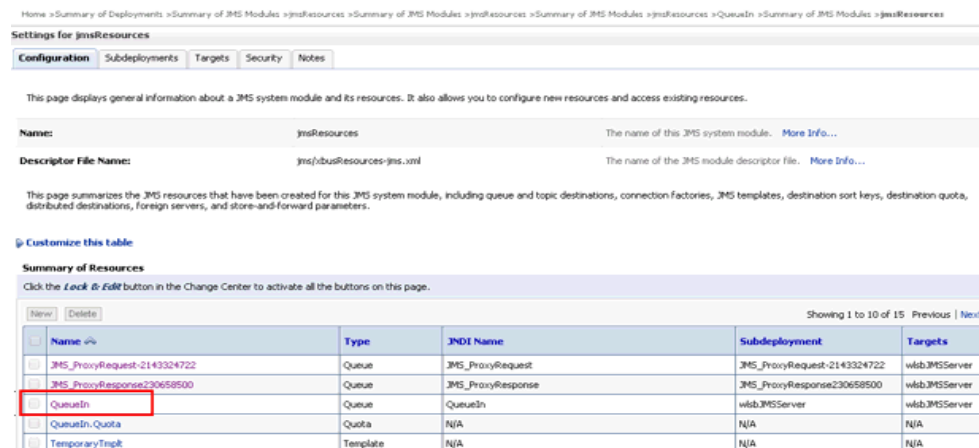
9. 詳細が適切に構成されていることを確認してから、パイプラインの構成ページを保存して閉じます。
コンポジット・エディタ・ウィンドウに戻ります。
10. 図 8-79 に示すように、メニュー・バーで「すべて保存」をクリックし、OSB JMS プロセスを保存します。

図 8-79 「すべて保存」アイコン



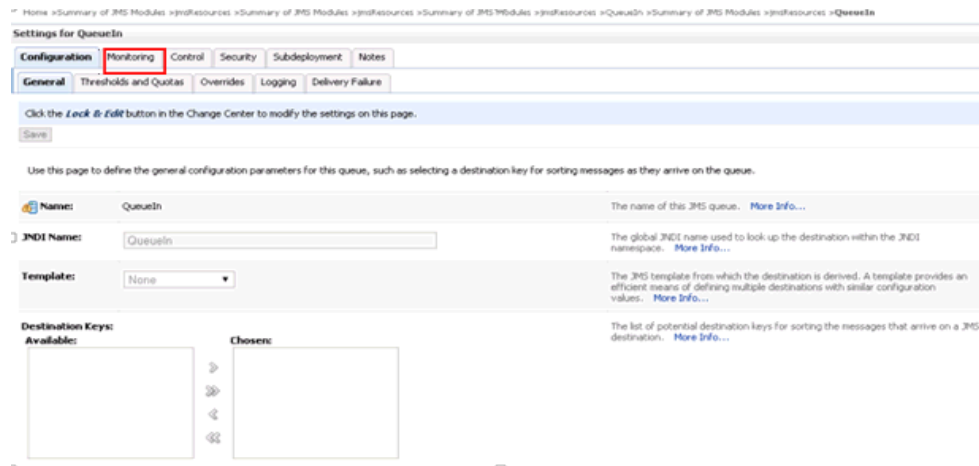
11. OSB JMS インバウンド・プロセスをデプロイします。詳細は、8-26 ページの 8.2.3 項「OSB インバウンド・プロセスのデプロイ」を参照してください。
12. プロセスが正常にデプロイされた後、イベント・メッセージをトリガーします。詳細は、4-48 ページの 4.5.5 項「J.D. Edwards OneWorld でのイベントのトリガー」を参照してください。
13. Oracle WLS コンソールにログオンします。
14. Oracle WLS コンソールで、「サービス」を展開し、「メッセージング」をクリックし、「JMS モジュール」を選択して、「jmsResources」をクリックします。
15. 図 8-80 に示すように、適切なレスポンス・リンク（「QueueIn」など）をクリックします。

図 8-80 「QueueIn」 レスポンス・リンク



16. 図 8-81 に示すように、「モニタリング」タブをクリックします。

図 8-81 「モニタリング」タブ



17. 図 8-82 に示すように、チェック・ボックスを選択し、「メッセージの表示」ボタンをクリックします。

図 8-82 「メッセージの表示」ボタン



18. 適切な日時の ID リンクをクリックします。

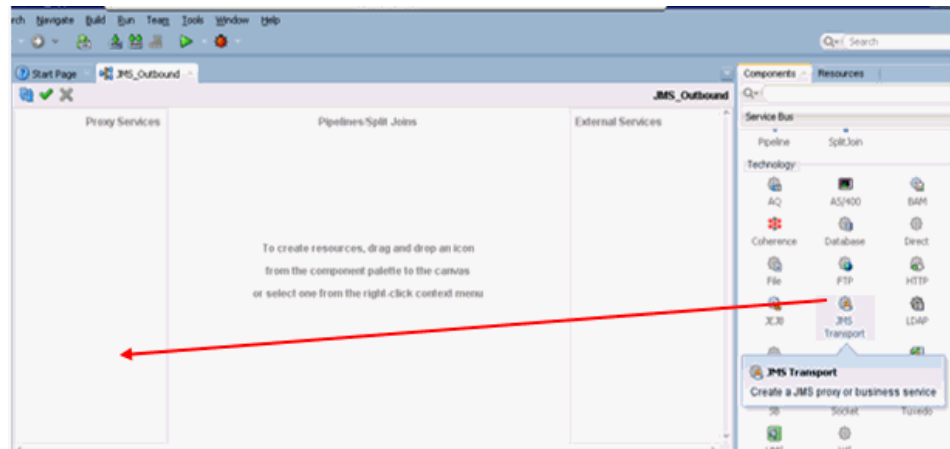
レスポンス・ドキュメントが「テキスト」フィールドの下に表示されます。

8.5 JDeveloper を使用した JMS アウトバウンド・プロセスの構成 (J2CA 構成)

この項では、J2CA 構成で Oracle JDeveloper を使用して J.D. Edwards OneWorld システム用に JMS アウトバウンド・プロセスを構成する方法について説明します。

1. JMS プロセスを設計する前に、アプリケーション・エクスプローラを使用して、各 WSDL ファイルを生成する必要があります。詳細は、4-8 ページの [4.4.1 項「リクエスト/レスポンス型のサービス用の WSDL の生成」](#) を参照してください。
2. Oracle JDeveloper を起動し、OSB 用のサービス・バス・アプリケーションを作成します。詳細は、8-2 ページの [8.1.1 項「OSB 用のサービス・バス・アプリケーションの作成」](#) を参照してください。
3. サード・パーティ・アダプタ・サービス・コンポーネントを作成します。詳細は、8-3 ページの [8.1.2.1 項「サード・パーティ・アダプタ・サービス・コンポーネントの構成」](#) を参照してください。
4. JCA バインド・ファイルから WSDL ベースのビジネス・サービスを作成します。詳細は、8-3 ページの [8.1.2.1 項「サード・パーティ・アダプタ・サービス・コンポーネントの構成」](#) を参照してください。
5. JMS プロキシ・サービスとパイプラインを作成し、次のステップを実行します。
 - a. [図 8-83](#) に示すように、「テクノロジー」コンポーネント・ペインから「JMS トランスポート」コンポーネントを「プロキシ・サービス」ペインにドラッグ・アンド・ドロップします。

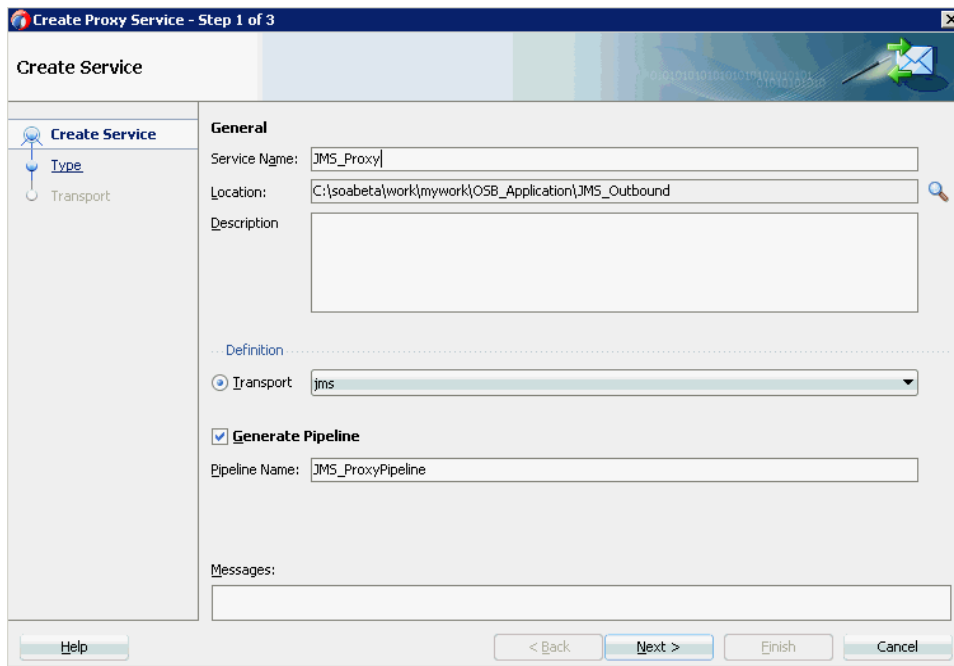
図 8-83 「JMS トランスポート」コンポーネント



「ビジネス・サービスの作成」ダイアログが表示されます。

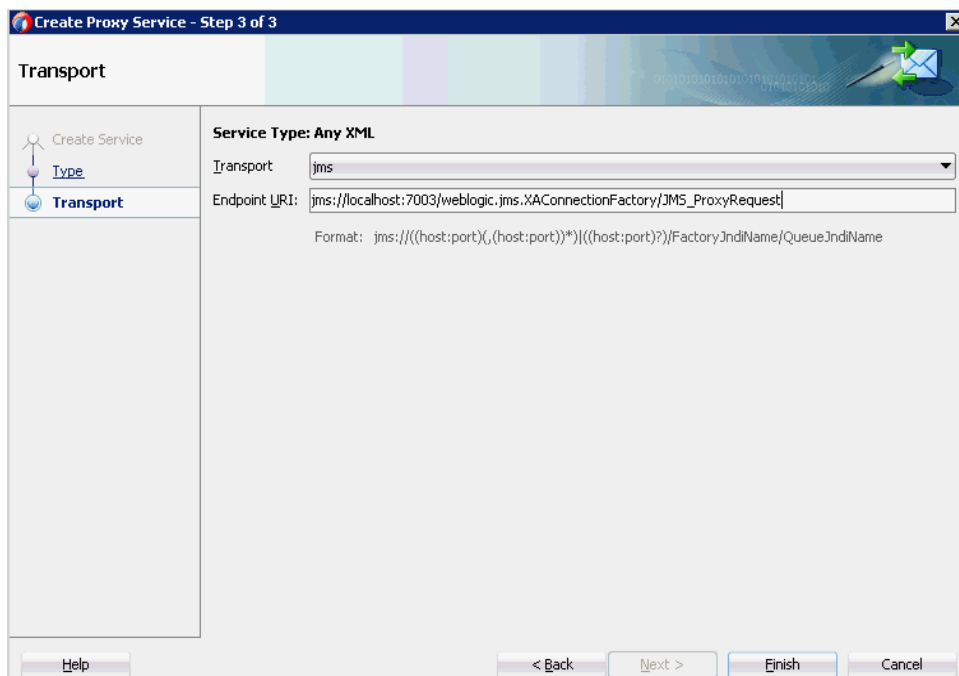
- b. 「サービス名」フィールドにプロキシ・サービスの任意の名前を入力します（「JMS_Proxy」など）。デフォルトで、「パイプラインの生成」が選択されています。
- c. [図 8-84](#) に示すように、「次へ」をクリックします。

図 8-84 「プロキシ・サービスの作成」 ペイン



- d. 表示された「タイプ」ウィンドウで、「任意の XML」を選択し、「次へ」をクリックします。
「トランスポート」ウィンドウが表示されます。
- e. 図 8-85 に示すように、「エンドポイント URI」フィールドを置き換えて適切なホスト名とポート番号に変更し
(「jms://localhost:7003/weblogic.jms.XAConnectionFactory/JMS_ProxyRequest」など)、「終了」をクリックします。

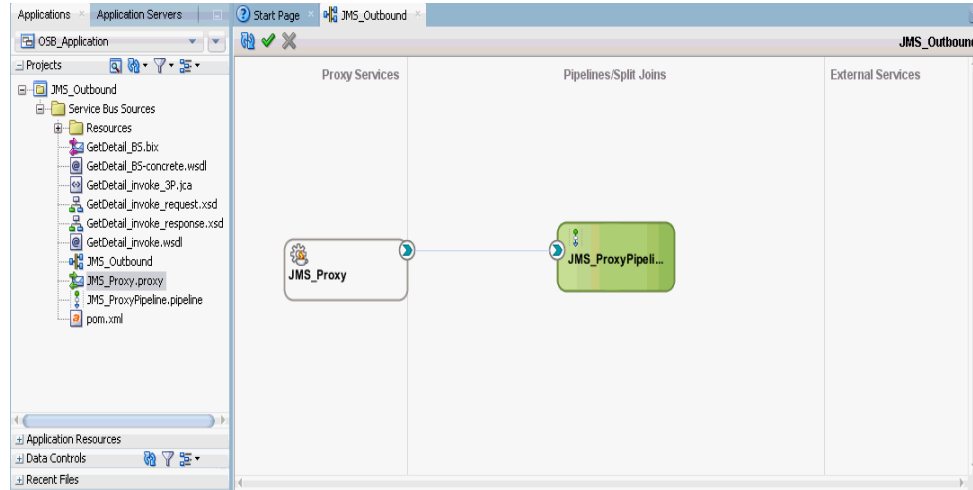
図 8-85 「トランスポート」ウィンドウ



JMS プロキシ・サービスとパイプラインが作成され、表示されます。

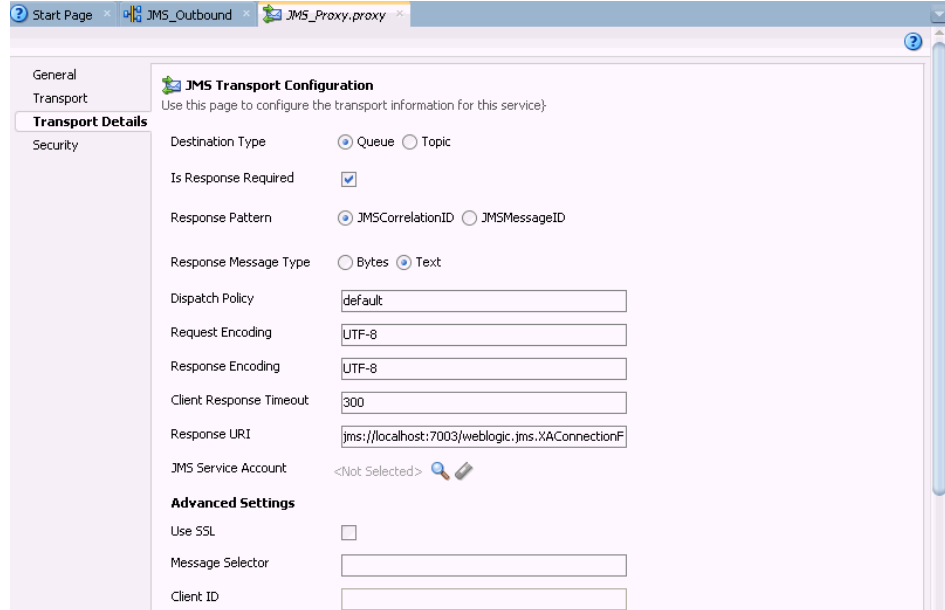
- f. 図 8-86 に示すように、作成されたプロキシ・サービス (「JMS_Proxy」など) をダブルクリックします。

図 8-86 JMS プロキシ・サービス



- g. 図 8-87 に示すように、表示されたプロキシ・サービスの構成ページにおいて、「トランスポートの詳細」タブで次のパラメータを指定します。

図 8-87 JMS トランスポート構成

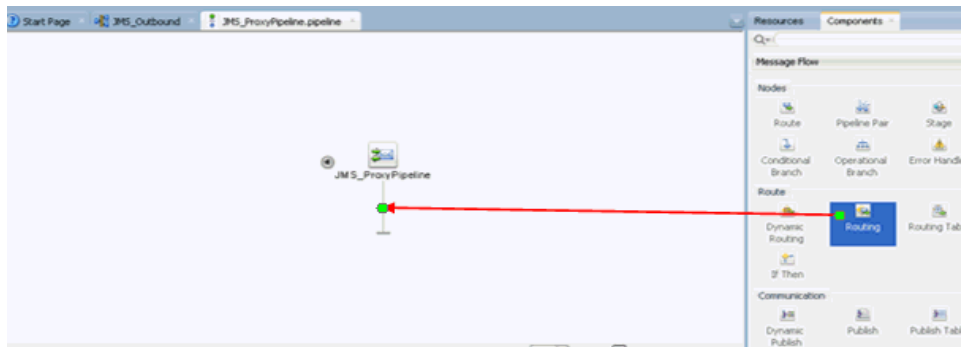


- h. 「宛先タイプ」セクションで、「キュー」を選択します。
- i. 「レスポンスが必要」チェック・ボックスを選択します。
- j. 「レスポンス・メッセージ・タイプ」セクションで、「テキスト」を選択します。
- k. 「レスポンス URI」フィールドに、JMS トランスポート構成で使用したエンドポイント URI を入力し、「Request」を「Response」に変更します。次に例を示します。

jms://localhost:7003/weblogic.jms.XAConnectionFactory/JMS_ProxyResponse

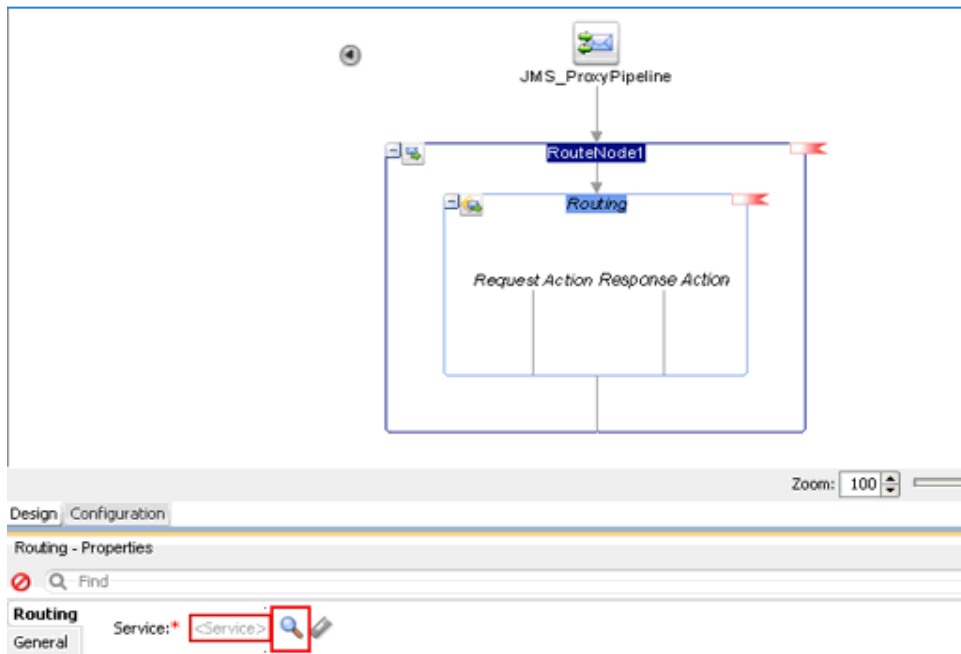
6. プロキシ・サービスの構成ページを保存して閉じます。
7. ルーティング・ルールを構成し、次のステップを実行します。
 - a. 「パイプライン/分割結合」ペインでパイプライン (「JMS_ProxyPipeline」など) をダブルクリックします。
パイプラインの構成ページが表示されます。
 - b. 図 8-88 に示すように、「ルート」セクションから「ルーティング」コンポーネントをパイプライン (「JMS_ProxyPipeline」など) の下の領域にドラッグ・アンド・ドロップします。

図 8-88 「ルーティング」コンポーネント



- c. 図 8-89 に示すように、パイプラインの構成ページで、「ルーティング」を選択し、「ルーティングプロパティ」ペインの「サービス」フィールドの右の参照アイコンをクリックします。

図 8-89 サービスの参照



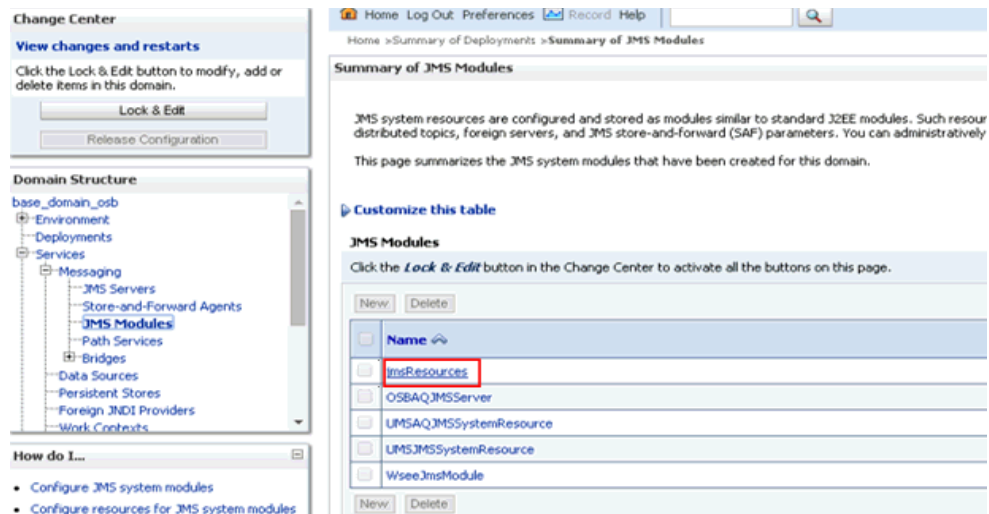
- d. 表示された「リソース・チューザ」ウィンドウで、WSDL ベースのビジネス・サービス (「xxxxx_BS.bix」など) を選択し、「OK」をクリックします。
パイプラインの構成ページに戻ります。
- e. パイプラインの構成ページを保存して閉じます。
コンポジット・エディタ・ウィンドウに戻ります。
- f. 図 8-90 に示すように、メニュー・バーで「すべて保存」をクリックし、OSB JMS プロセスを保存します。

図 8-90 「トランスポート」ウィンドウ



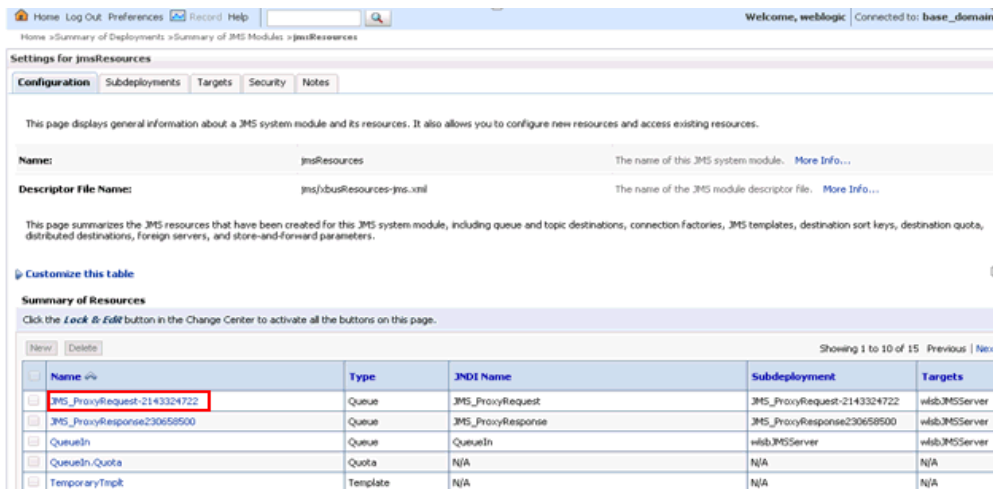
8. OSB JMS アウトバウンド・プロセスをデプロイします。詳細は、8-16 ページの 8.1.3 項「OSB アウトバウンド・プロセスのデプロイ」を参照してください。
9. プロセスが正常にデプロイされた後、Oracle WLS コンソールにログオンします。
10. 図 8-91 に示すように、Oracle WLS コンソールで、「サービス」を展開し、「メッセージング」をクリックし、「JMS モジュール」を選択して、「jmsResources」をクリックします。

図 8-91 JMS リソース



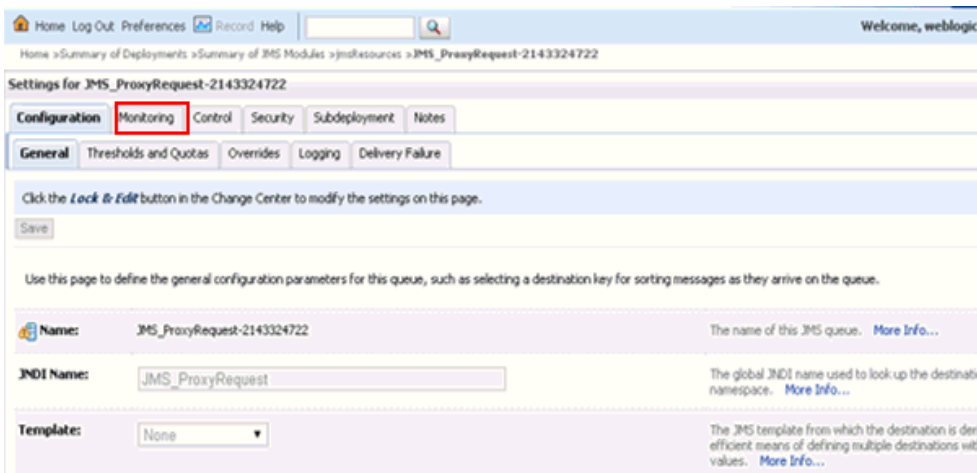
11. 図 8-92 に示すように、適切なリクエスト・リンク (「JMS_ProxyRequest」など) をクリックします。

図 8-92 「JMS_ProxyRequest」 リンク



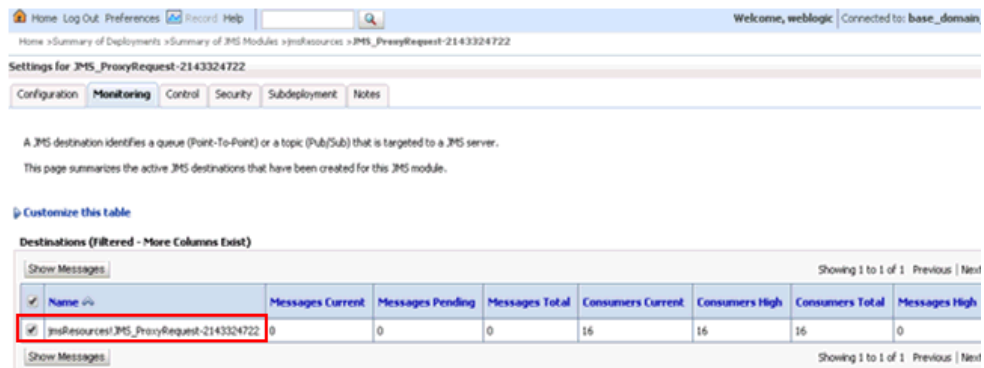
12. 図 8-93 に示すように、「モニタリング」タブをクリックします。

図 8-93 「モニタリング」タブ



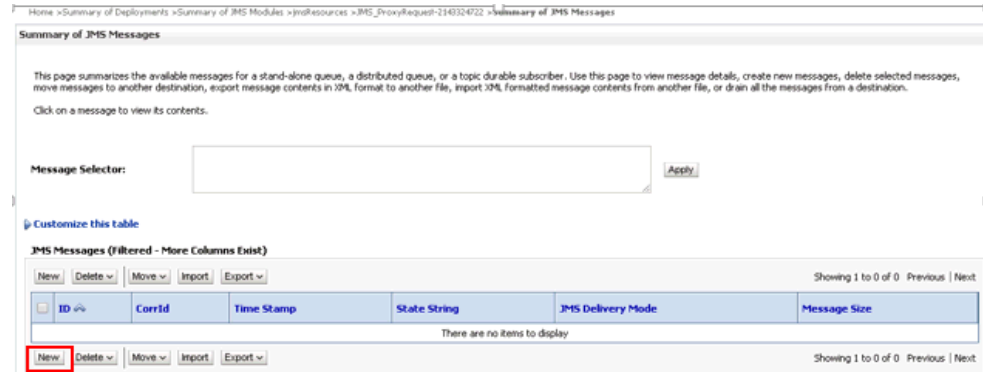
13. 図 8-94 に示すように、チェック・ボックスを選択し、「メッセージの表示」ボタンをクリックします。

図 8-94 「メッセージの表示」ボタン



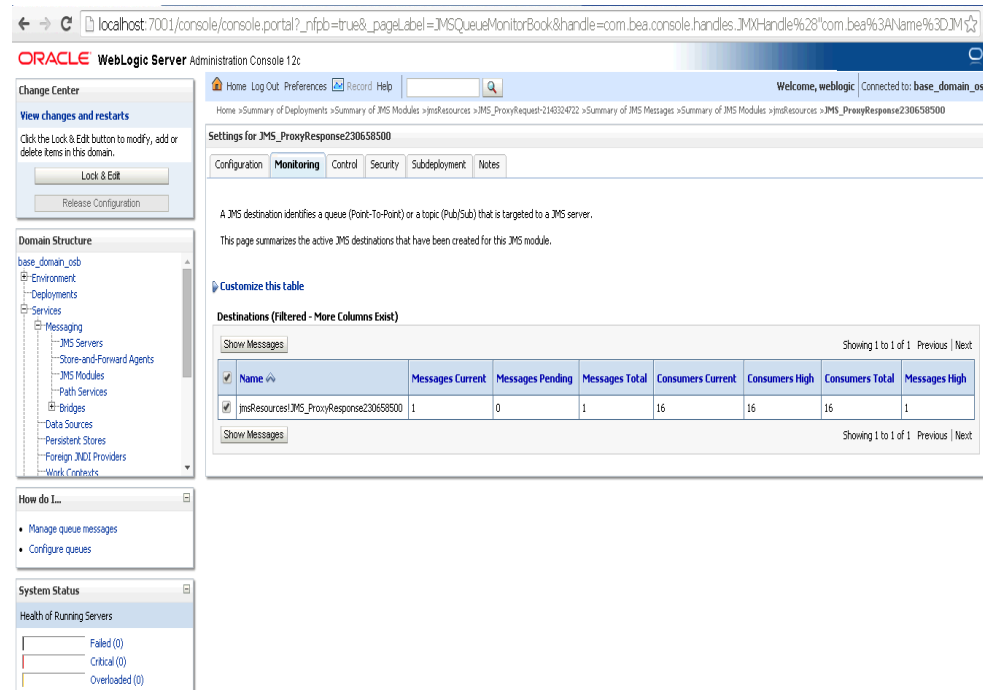
14. 図 8-95 に示すように、「新規」をクリックします。

図 8-95 JMS メッセージ



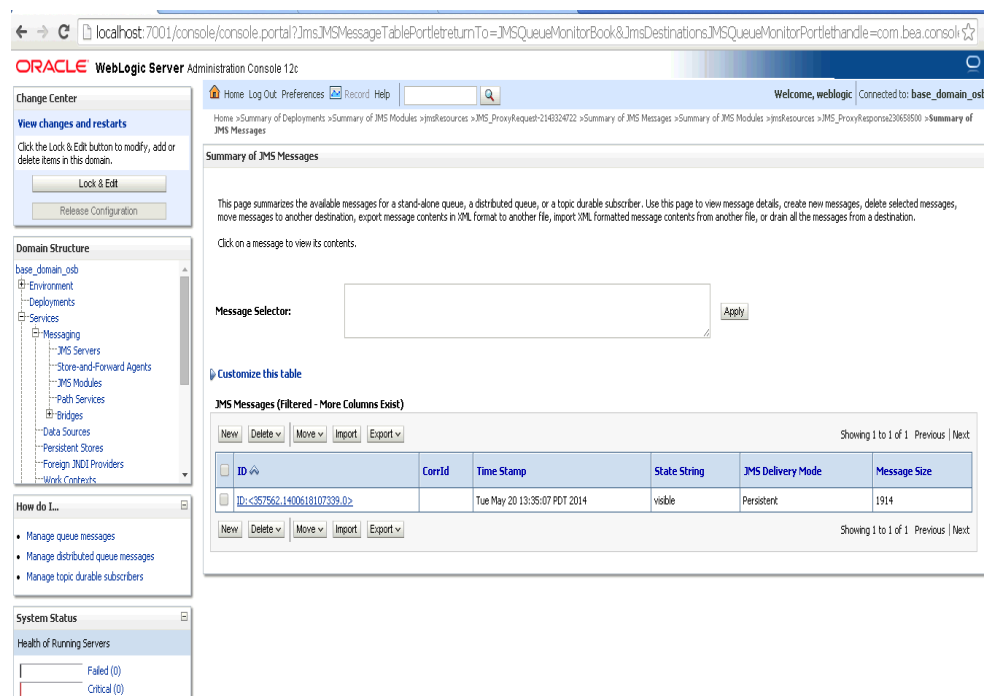
15. 「本文」フィールドに入力ペイロードを入力し、「OK」をクリックします。
16. Oracle WLS コンソールで、「サービス」を展開し、「メッセージング」をクリックし、「JMS モジュール」を選択して、「jmsResources」をクリックします。
17. 適切なレスポンス・リンク（「JMS_ProxyResponse」など）をクリックします。
18. 「モニタリング」タブをクリックします。
19. 図 8-96 に示すように、チェック・ボックスを選択し、「メッセージの表示」をクリックします。

図 8-96 宛先のメッセージ



20. 図 8-97 に示すように、適切な日時の ID リンクをクリックします。

図 8-97 「JMS メッセージのサマリー」 ウィンドウ



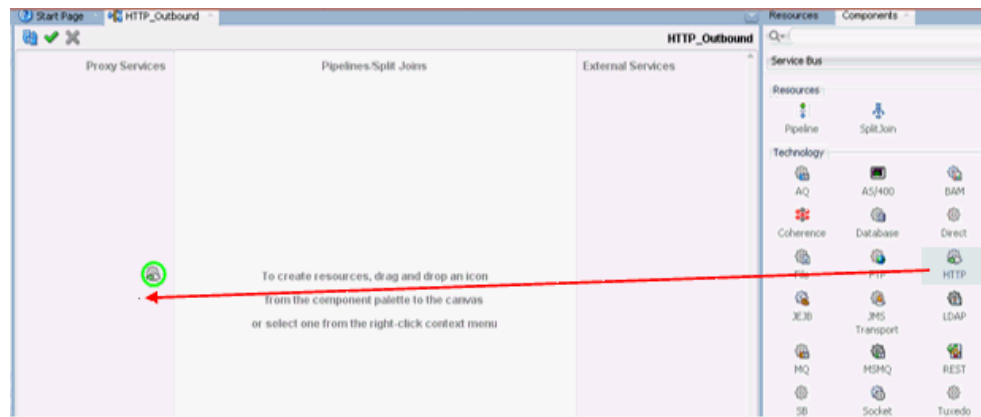
レスポンス・ドキュメントが「テキスト」フィールドの下に表示されます。

8.6 JDeveloper を使用した HTTP アウトバウンド・プロセスの構成 (J2CA 構成)

この項では、J2CA 構成で Oracle JDeveloper を使用して J.D. Edwards OneWorld システム用に HTTP アウトバウンド・プロセスを構成する方法について説明します。

1. HTTP アウトバウンド・プロセスを設計する前に、アプリケーション・エクスペローラを使用して各 WSDL ファイルを生成する必要があります。詳細は、4-8 ページの 4.4.1 項「リクエスト/レスポンス型のサービス用の WSDL の生成」を参照してください。
2. Oracle JDeveloper を起動し、OSB 用のサービス・バス・アプリケーションを作成します。詳細は、8-2 ページの 8.1.1 項「OSB 用のサービス・バス・アプリケーションの作成」を参照してください。
3. サード・パーティ・アダプタ・サービス・コンポーネントを作成します。詳細は、8-3 ページの 8.1.2.1 項「サード・パーティ・アダプタ・サービス・コンポーネントの構成」を参照してください。
4. HTTP プロキシ・サービスとパイプラインを作成し、次のステップを実行します。
 - a. 図 8-98 に示すように、「テクノロジー」コンポーネント・ペインから「HTTP」コンポーネントを「プロキシ・サービス」ペインにドラッグ・アンド・ドロップします。

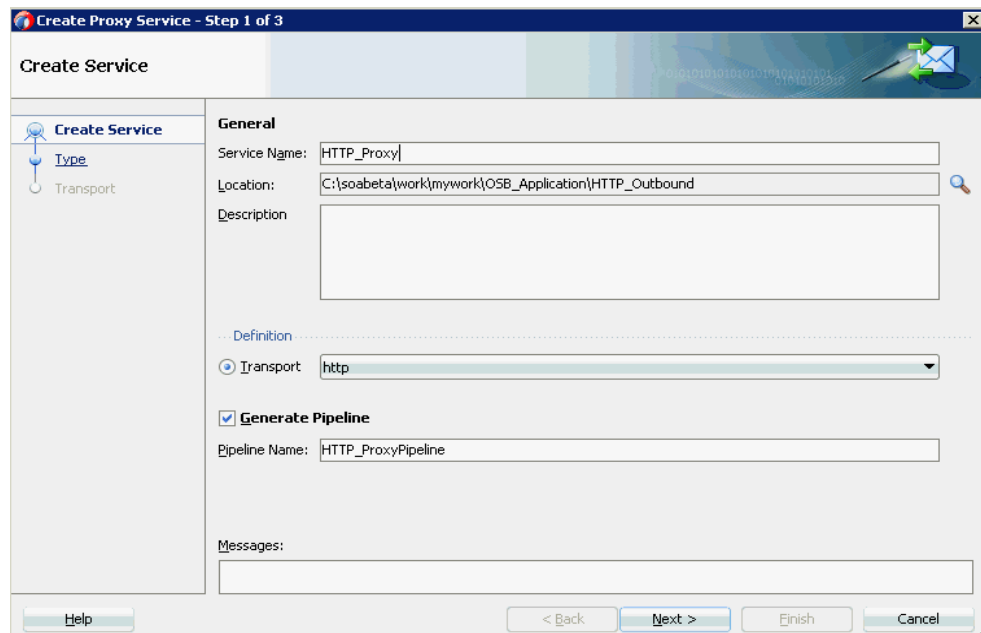
図 8-98 HTTP コンポーネント



「プロキシ・サービスの作成」ダイアログが表示されます。

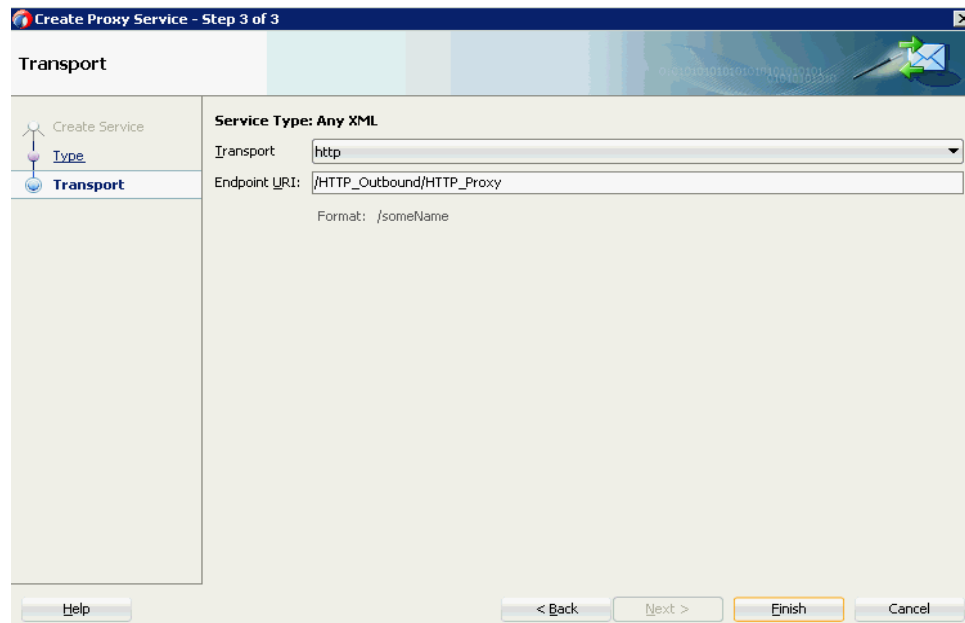
- b. 「サービス名」フィールドにプロキシ・サービスの任意の名前を入力します (「HTTP_Proxy」など)。デフォルトで、「パイプラインの生成」が選択されています。
- c. 図 8-99 に示すように、「次へ」をクリックします。

図 8-99 「プロキシ・サービスの作成」ペイン



- d. 表示された「タイプ」ウィンドウで、「任意の XML」を選択し、「次へ」をクリックします。
「トランスポート」ウィンドウが表示されます。
- e. 図 8-100 に示すように、デフォルト値をそのまま使用し、「終了」をクリックします。

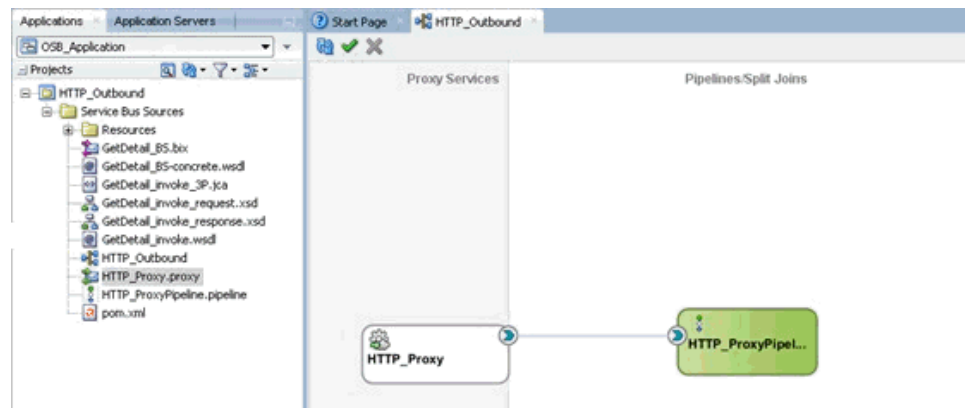
図 8-100 「トランスポート」ウィンドウ



HTTP プロキシ・サービスとパイプラインが作成され、表示されます。

- f. 図 8-101 に示すように、「パイプライン/分割結合」ペインで作成されたパイプライン(「HTTP_ProxyPipeline」など)をダブルクリックします。

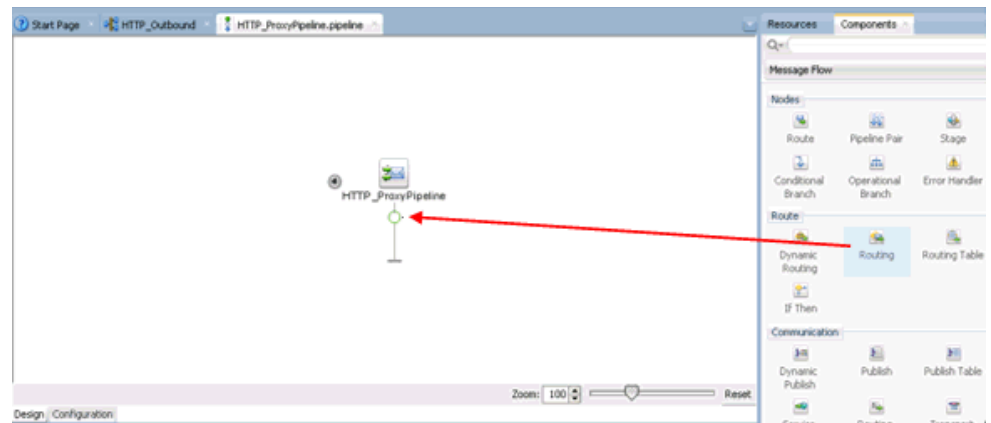
図 8-101 プロキシ・サービス



パイプラインの構成ページが表示されます。

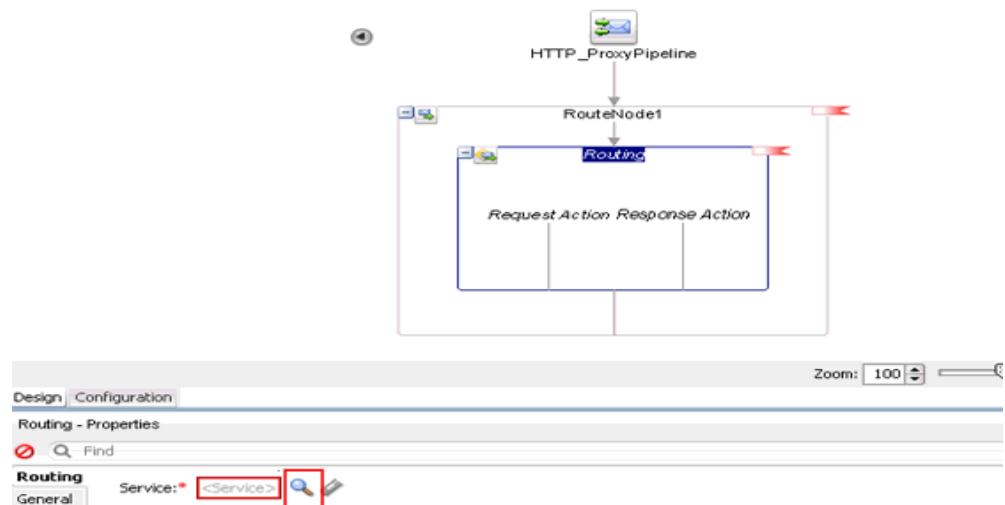
5. ルーティング・ルールを構成し、次のステップを実行します。
 - a. 図 8-102 に示すように、「ルート」セクションから「ルーティング」コンポーネントをパイプライン(「HTTP_ProxyPipeline」など)の下領域にドラッグ・アンド・ドロップします。

図 8-102 「ルーティング」コンポーネント



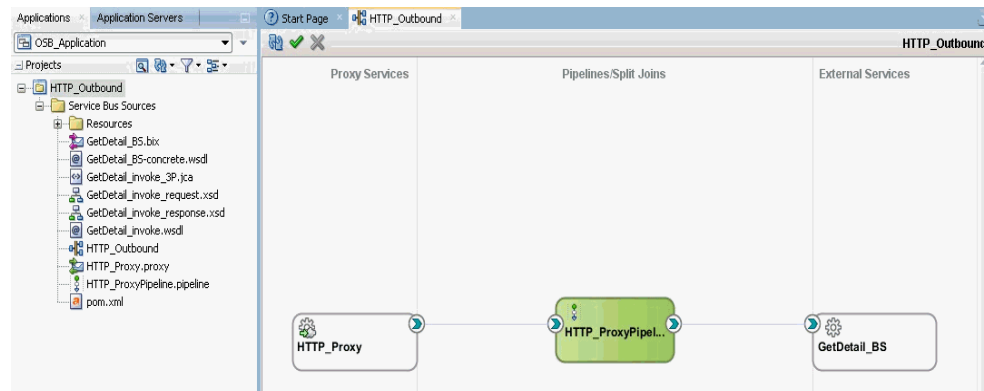
- b. 図 8-103 に示すように、パイプラインの構成ページで、「ルーティング」を選択し、「ルーティングプロパティ」ペインの「サービス」フィールドの右の参照アイコンをクリックします。

図 8-103 サービスの参照



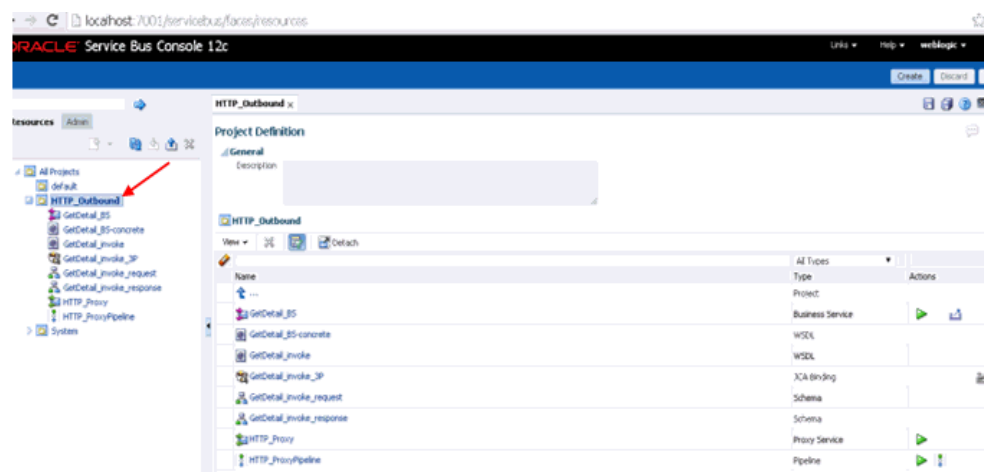
- c. 表示された「リソース・チューザ」ウィンドウで、WSDL ベースのビジネス・サービス (「xxxxx_BS.bix」など) を選択し、「OK」をクリックします。
パイプラインの構成ページに戻ります。
- d. パイプラインの構成ページを保存して閉じます。
コンポジット・エディタ・ウィンドウに戻ります。
- e. 図 8-104 に示すように、メニュー・バーで「すべて保存」をクリックし、OSB HTTP プロセスを保存します。

図 8-104 「トランスポート」ウィンドウ



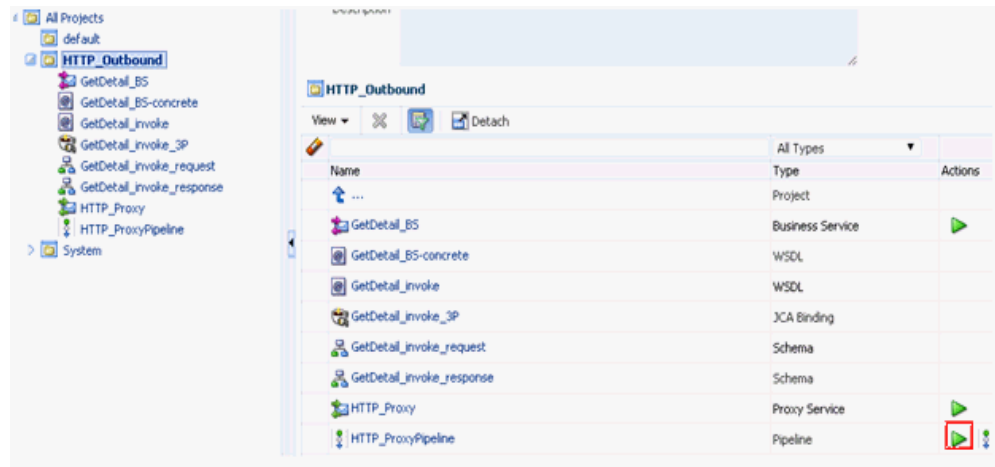
6. OSB HTTP アウトバウンド・プロセスをデプロイします。詳細は、8-16 ページの 8.1.3 項「OSB アウトバウンド・プロセスのデプロイ」を参照してください。
7. プロセスが正常にデプロイされた後、Service Bus コンソールにログオンします。
8. 図 8-105 に示すように、Service Bus コンソールで、デプロイされた HTTP アウトバウンド・プロジェクト (「HTTP_Outbound」など) をクリックします。

図 8-105 Service Bus コンソール



9. 図 8-106 に示すように、作成されたパイプラインに対する OSB コンソールのテスト・アイコンをクリックします。

図 8-106 OSB コンソールのテスト・アイコン



- 表示された OSB コンソールのテスト・ページで、入力 XML を指定し、「実行」ボタンをクリックします。

表示された OSB コンソールのテスト・ページで、レスポンスが受信されます。

主要な機能

この章では、Oracle Application Adapter for J.D. Edwards OneWorld の主要な機能について説明します。この章の内容は以下のとおりです。

- 9.1 項「ロギング機能の構成」
- 9.2 項「診断機能の構成」
- 9.3 項「SOA デバッグ機能の構成」
- 9.4 項「例外フィルタ」
- 9.5 項「Oracle SOA Suite (BPEL、メディエータまたは BPM) の資格証明マッピング」
- 9.6 項「Oracle Service Bus (OSB) の資格証明マッピング」

9.1 ロギング機能の構成

Oracle 12c (12.2.1.0.0) では、J2CA および BSE アダプタ・ログは、次の場所で入手可能な `{server-name}-diagnostic.log` ファイルの Oracle ログで更新されます。

```
<ORACLE_HOME>\user_projects\domains\base_domain\servers\<server_Name>\logs
```

注意： J2CA 用アプリケーション・エクスペローラのログ・ファイルは、`<ADAPTER_HOME>\config\xxxxxxx\log` フォルダの下に作成されます。xxxxxxx はアプリケーション・エクスペローラで作成された J2CA 構成の名前です。アプリケーション・エクスペローラでの各 J2CA 構成には、名前付き J2CA 構成フォルダの下に対応するログ・フォルダが存在します。

この項では、ロギング機能の構成方法について説明します。次のトピックが含まれています。

- 9.1.1 項「J2CA コネクタ・アプリケーションのログ・ファイル管理の構成」
- 9.1.2 項「Business Services Engine (BSE) のログ・ファイル管理の構成」

9.1.1 J2CA コネクタ・アプリケーションのログ・ファイル管理の構成

J2CA コネクタ・アプリケーションのログ・ファイル管理は、次の場所で定義されるログ出力によって制御されます。

```
<ORACLE_HOME>\user_projects\domains\base_
domain\config\fmwconfig\servers\${server-name}\logging.xml
```

新しいログ出力が EM コンソールから管理される場合、このファイルに追加する必要があります。

次に例を示します。

```
<logger name='oracle.soa.adapter.iwaf' level='NOTIFICATION:1'  
useParentHandlers='true'/>  
<logger name='oracle.soa.adapter.iwaf.classloader' level='NOTIFICATION:1'  
useParentHandlers='true'/>  
<logger name='oracle.soa.adapter.iwaf.connection' useParentHandlers='true'/>  
<logger name='oracle.soa.adapter.iwaf.connection.IAEAdapter'  
useParentHandlers='true'/>  
<logger name='oracle.soa.adapter.iwaf.connection.Sample'  
useParentHandlers='true'/>  
<logger name='oracle.soa.adapter.iwaf.connection.JDEdwards'  
useParentHandlers='true'/>  
<logger name='oracle.soa.adapter.iwaf.inbound' useParentHandlers='true'/>  
<logger name='oracle.soa.adapter.iwaf.inbound.IAEAdapter'  
useParentHandlers='true'/>  
<logger name='oracle.soa.adapter.iwaf.inbound.Sample' useParentHandlers='true'/>  
<logger name='oracle.soa.adapter.iwaf.inbound.JDEdwards'  
useParentHandlers='true'/>  
<logger name='oracle.soa.adapter.iwaf.outbound' useParentHandlers='true'/>  
<logger name='oracle.soa.adapter.iwaf.outbound.IAEAdapter'  
useParentHandlers='true'/>  
<logger name='oracle.soa.adapter.iwaf.outbound.Sample' useParentHandlers='true'/>  
<logger name='oracle.soa.adapter.iwaf.outbound.JDEdwards'  
useParentHandlers='true'/>  
<logger name='oracle.soa.adapter.iwaf.transaction' useParentHandlers='true'/>  
<logger name='oracle.soa.adapter.iwaf.transaction.IAEAdapter'  
useParentHandlers='true'/>  
<logger name='oracle.soa.adapter.iwaf.transaction.Sample'  
useParentHandlers='true'/>  
<logger name='oracle.soa.adapter.iwaf.transaction.JDEdwards'  
useParentHandlers='true'/>  
<logger name='oracle.soa.adapter.iwaf.IAEAdapter' useParentHandlers='true'/>  
<logger name='oracle.soa.adapter.iwaf.Sample' useParentHandlers='true'/>  
<logger name='oracle.soa.adapter.iwaf.JDEdwards' useParentHandlers='true'/>
```

これにより、「oracle.soa.adapter.iwaf」の下にあるすべてのログ出力のロギング・レベルが、Oracle によるデフォルトの設定レベルである「NOTIFICATION:1 (INFO)」に設定されます。

次のステップを実行して、すべてのログ出力のロギング・レベルを EM コンソールから構成することもできます。

1. 構成済の Oracle WebLogic Server ドメインの Oracle WebLogic Server を起動します。
2. Web ブラウザで次の URL を入力して、Oracle WebLogic Server Enterprise Manager Console を起動します。

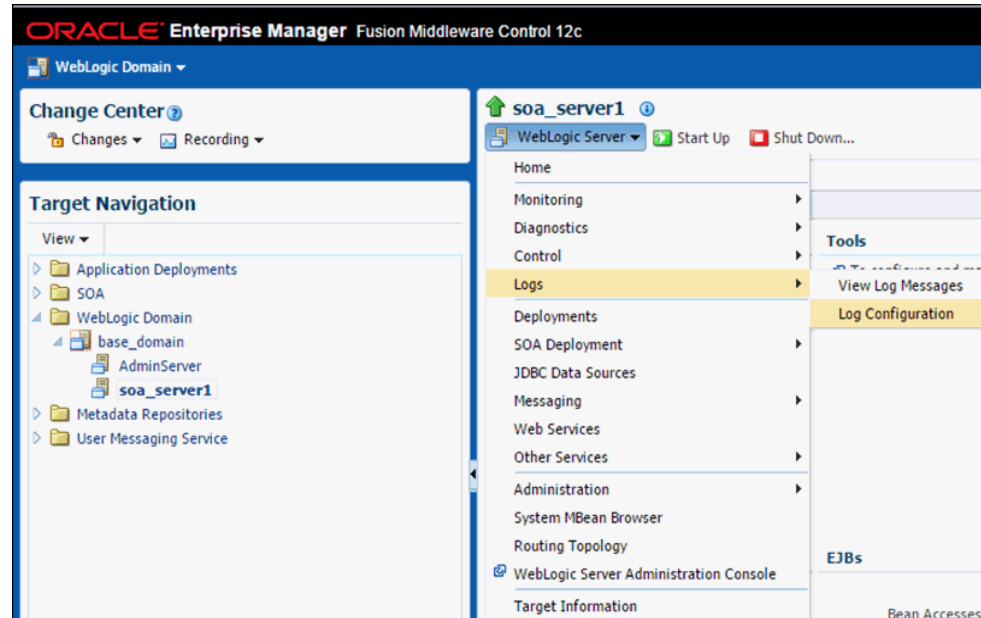
```
http://host name:port/em
```

ここで、host name は Oracle WebLogic Server が稼働しているシステムの名前で、port は稼働している Oracle WebLogic Server のポートです。Oracle WebLogic Server のデフォルト・ポートは 7001 です。ただし、この値はインストールによって異なる場合があります。

3. 管理者権限を持つアカウントを使用して、Oracle WebLogic Server 管理コンソールにログインします。

4. 「ターゲット・ナビゲーション」ペインの下で、「WebLogic ドメイン」をクリックし、作成したドメインを選択して、適切なサーバー（管理対象サーバーまたは統合サーバー）をクリックします。
5. 図 9-1 に示すように、「サーバー」ペインで、「WebLogic Server」を展開し、「ログ」を選択して、「ログ構成」をクリックします。

図 9-1 「ログ構成」オプション



6. 図 9-2 に示すように、「ログ・レベル」タブの下で、「oracle.soa.adapter.iwaf」が表示されるまで Oracle ルート・ログ出力を展開します。

図 9-2 「ログ・レベル」タブ

Log Configuration
Use this page to configure basic and advanced log configuration settings.

Log Levels | Log Files | QuickTrace

This page allows you to configure the log level for both persistent loggers and active runtime loggers. Persistent loggers are loggers that are saved in a configuration file when a component is started. The log levels for these loggers are persisted across component restarts. Runtime loggers are automatically created during runtime and become active when an EJB module is deployed. Log levels for runtime loggers are not persisted across component restarts.

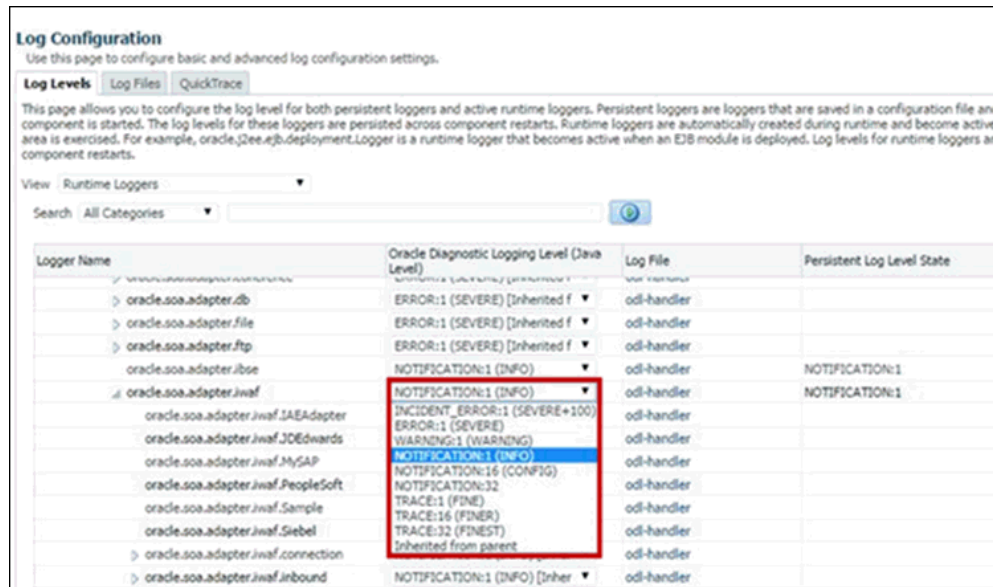
View: Runtime Loggers

Search: All Categories

Logger Name	Oracle Diagnostic Logging Level (Java Level)	Log File	Persistent Log Level State
oracle.soa.adapter.db	ERROR:1 (SEVERE) [Inherited from ...]	odl-handler	
oracle.soa.adapter.file	ERROR:1 (SEVERE) [Inherited from ...]	odl-handler	
oracle.soa.adapter.ftp	ERROR:1 (SEVERE) [Inherited from ...]	odl-handler	
oracle.soa.adapter.ibse	NOTIFICATION:1 (INFO)	odl-handler	NOTIFICATION:1
oracle.soa.adapter.iwaf	NOTIFICATION:1 (INFO)	odl-handler	NOTIFICATION:1
oracle.soa.adapter.iwaf.IAEdapter	NOTIFICATION:1 (INFO) [Inherited from ...]	odl-handler	
oracle.soa.adapter.iwaf.JDEdwards	NOTIFICATION:1 (INFO) [Inherited from ...]	odl-handler	
oracle.soa.adapter.iwaf.MySAP	NOTIFICATION:1 (INFO) [Inherited from ...]	odl-handler	
oracle.soa.adapter.iwaf.PeopleSoft	NOTIFICATION:1 (INFO) [Inherited from ...]	odl-handler	
oracle.soa.adapter.iwaf.Sample	NOTIFICATION:1 (INFO) [Inherited from ...]	odl-handler	
oracle.soa.adapter.iwaf.Siebel	NOTIFICATION:1 (INFO) [Inherited from ...]	odl-handler	
oracle.soa.adapter.iwaf.connection	NOTIFICATION:1 (INFO) [Inherited from ...]	odl-handler	
oracle.soa.adapter.iwaf.inbound	NOTIFICATION:1 (INFO) [Inherited from ...]	odl-handler	

7. 図 9-3 に示すように、「Oracle Diagnostic Logging レベル (Java レベル)」列で、「oracle.soa.adapter.iwaf」ドロップダウン・リストから必要なログ・レベルを選択します。

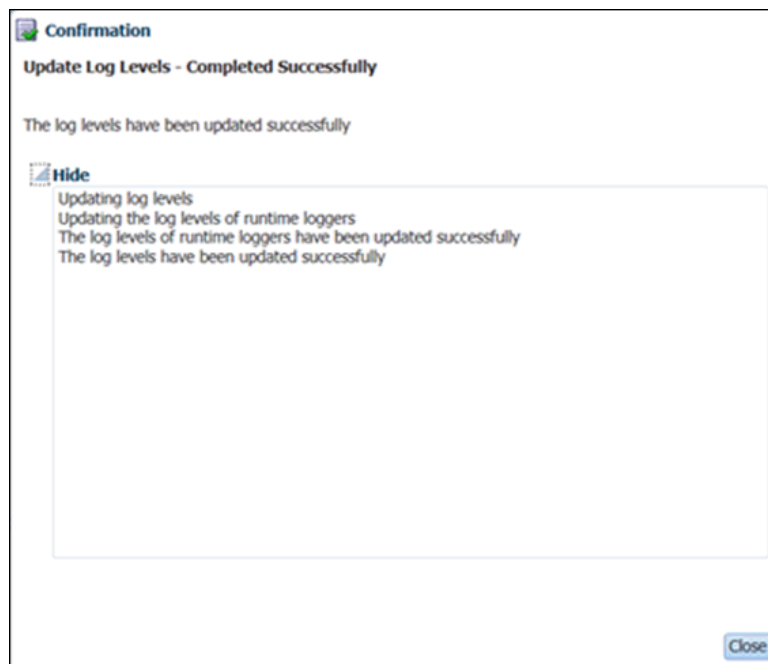
図 9-3 「Oracle Diagnostic Logging レベル (Java レベル)」列



8. 「適用」をクリックします。

図 9-4 に示すように、更新が正常に完了したことを示す確認メッセージが表示されます。

図 9-4 確認メッセージ



9. 「閉じる」をクリックします。

次の表は、「ログ・レベル」プロパティが `{server-name}-diagnostic.log` で EM コンソール内の対応する「ログ・レベル」プロパティ設定に基づいてどのように更新されるかを示しています。

表 9-1 「ログ・レベル」プロパティ設定

EM コンソールで設定された「ログ・レベル」	<code>{server-name}-diagnostic.log</code> で更新された「ログ・レベル」
ERROR:1	ERROR
WARNING:1	WARNING
NOTIFICATION:1	NOTIFICATION
NOTIFICATION:16	NOTIFICATION
NOTIFICATION:32	NOTIFICATION
TRACE:1	NOTIFICATION
TRACE:16	NOTIFICATION
TRACE:32	TRACE:32

注意： EM コンソールでログ・レベルを TRACE:32 に設定すると、`{server-name}-diagnostic.log` のログ・レベルが TRACE:32 に更新されて、ログに「FINEST」の詳細 (アダプタに渡された入力、アダプタから受信されたレスポンスなど) が表示されます。

開発環境およびテスト環境の場合、「TRACE:32」は、すべてのログ詳細が表示される、望ましいログ・レベルです。本番環境の場合、望ましいログ・レベルは「ERROR」です。

ログ・メッセージはディスク・ファイルに書き込まれ、ファイル・パスは「ログ・ファイル」タブで参照できます。「ログ・ファイル」表の「ハンドラ名」は、「ログ・レベル」表の「ログ・ファイル」の名前に対応しています。図 9-5 に示すように、「oracle.soa.adapter」の下階層にあるすべてのログ出力は、現在「odl-handler」によって処理されています。

図 9-5 「ログ構成」ペイン

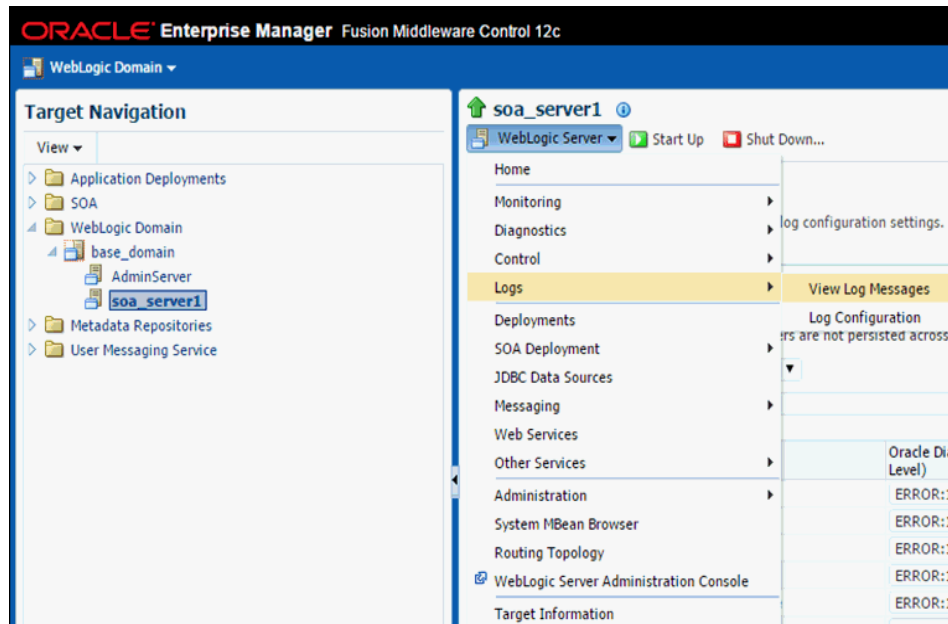
The screenshot shows the 'Log Configuration' page with the 'Log Files' tab selected. It contains a table with the following data:

Handler Name	Log Path	Log File Format
em-log-handler	<code>\${domain.home}/servers/\${weblogic.Name}/sysman/log/emoms.log</code>	Oracle Diagnostics Logging
em-trc-handler	<code>\${domain.home}/servers/\${weblogic.Name}/sysman/log/emoms.trc</code>	Oracle Diagnostics Logging
odl-handler	<code>\${domain.home}/servers/\${weblogic.Name}/logs/\${weblogic.Name}-diagnostic.log</code>	Oracle Diagnostics Logging
owsm-message-ha...	<code>\${domain.home}/servers/\${weblogic.Name}/logs/owsm/msglogging/diagnostic.log</code>	Oracle Diagnostics Logging
soa-tracking-trc-h...	<code>\${domain.home}/servers/\${weblogic.Name}/logs/\${weblogic.Name}-soa-tracking.trc</code>	Oracle Diagnostics Logging

ログは、次の場所で入手できる `{server-name}-diagnostic.log` で更新されます。
`<ORACLE_HOME>\user_projects\domains\base_domain\servers\<server_Name>\logs`

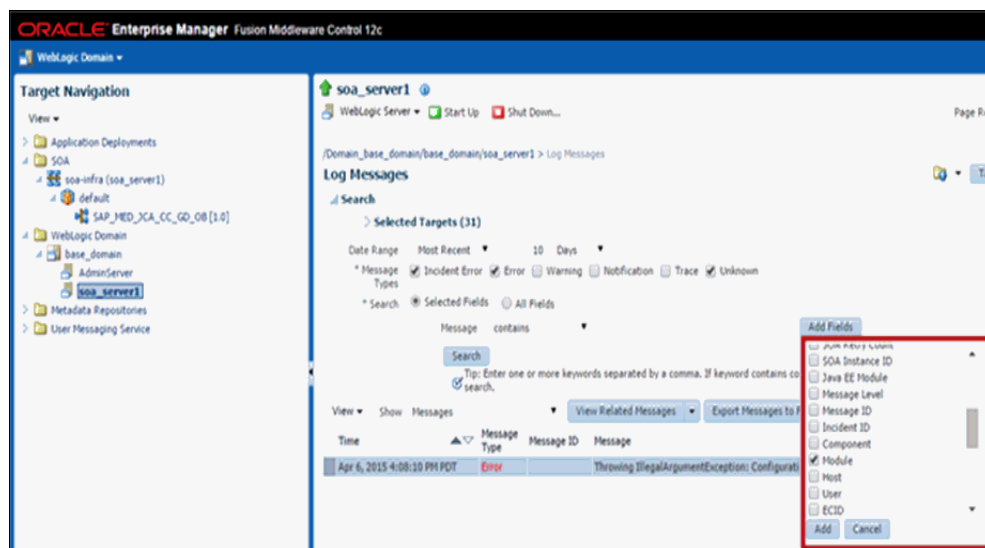
10. 図 9-6 に示すように、「サーバー」 ペインで、「WebLogic Server」 ドロップダウン・リストをクリックし、「ログ」 を選択し、「ログ・メッセージの表示」 をクリックすると、EM コンソールにログ・メッセージが表示されます。

図 9-6 「ログ・メッセージの表示」 オプション



11. 「ログ・メッセージ」 ペインで、必要な検索基準を入力します。図 9-7 に示すように、対象のログ出力の名前が含まれる「モジュール」 フィールドを検索基準に追加することもできます。

図 9-7 「モジュール」 フィールド



12. 「追加」 をクリックします。
13. 図 9-8 に示すように、「モジュール」 フィールドに対象のログ出力の名前を入力し、必要に応じて追加のメッセージ・タイプ (「警告」、「通知」、「トレース」 など) を選択して、「検索」 をクリックします。

図 9-8 「ログ・メッセージ」 ペイン

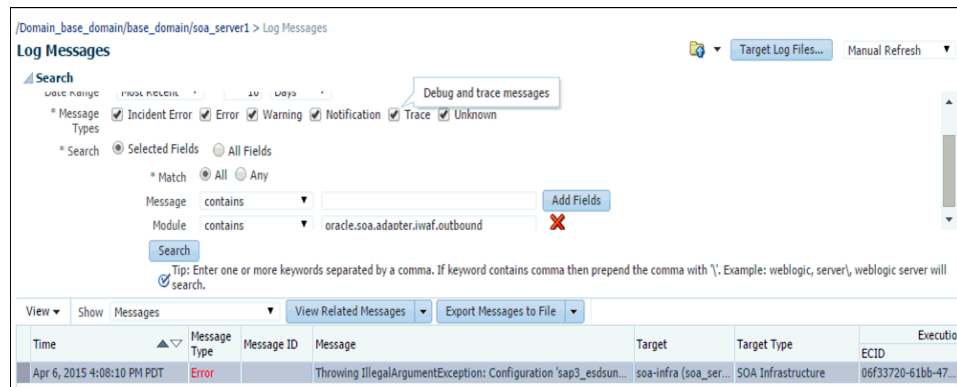
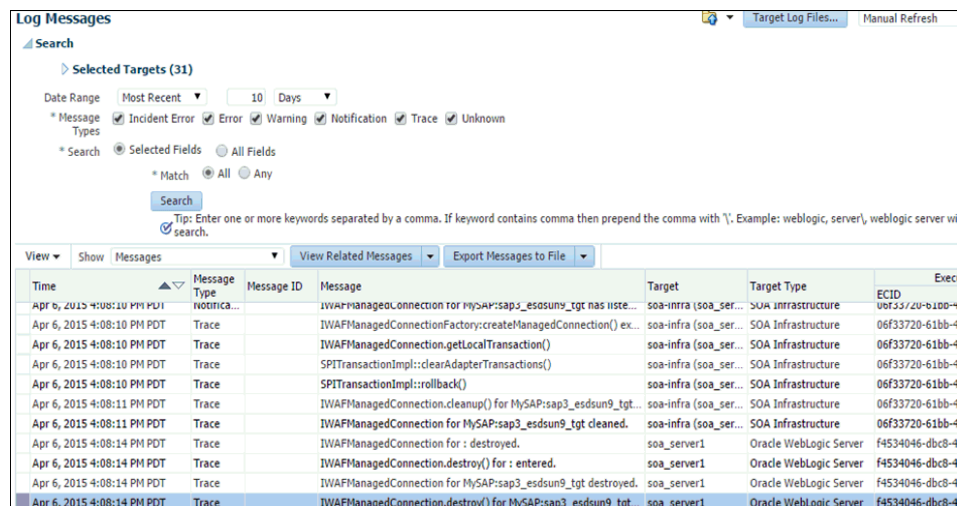


図 9-9 に示すように、指定されたログ出力のメッセージが、検索基準の下の表に表示されます。

図 9-9 ログ出力のメッセージ



14. 表の任意の行を選択します。図 9-10 に示すように、同じ詳細を取得するには、「関連メッセージの表示」ドロップダウン・リストをクリックし、「/ECID」(実行コンテキスト ID)を選択します。

図 9-10 「ECID」 オプション

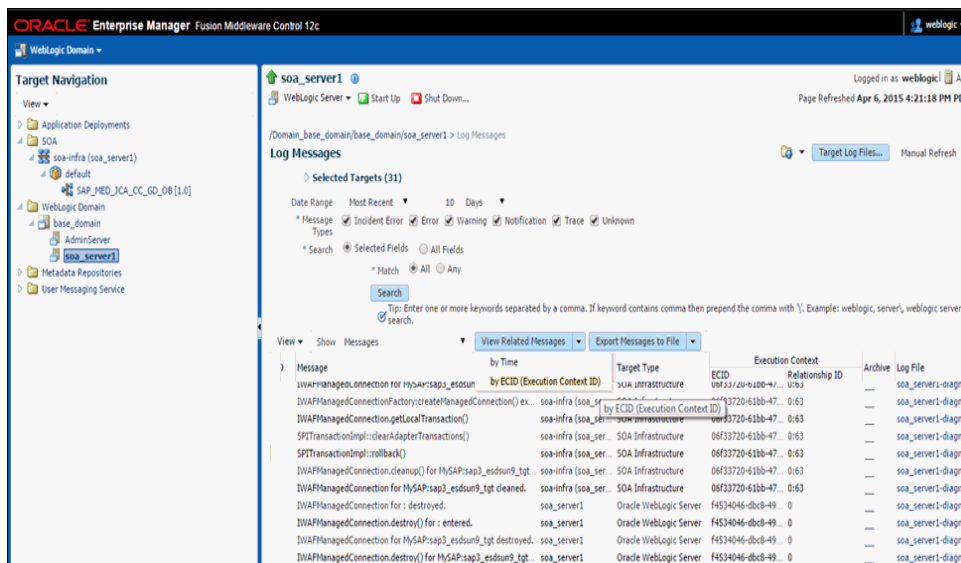
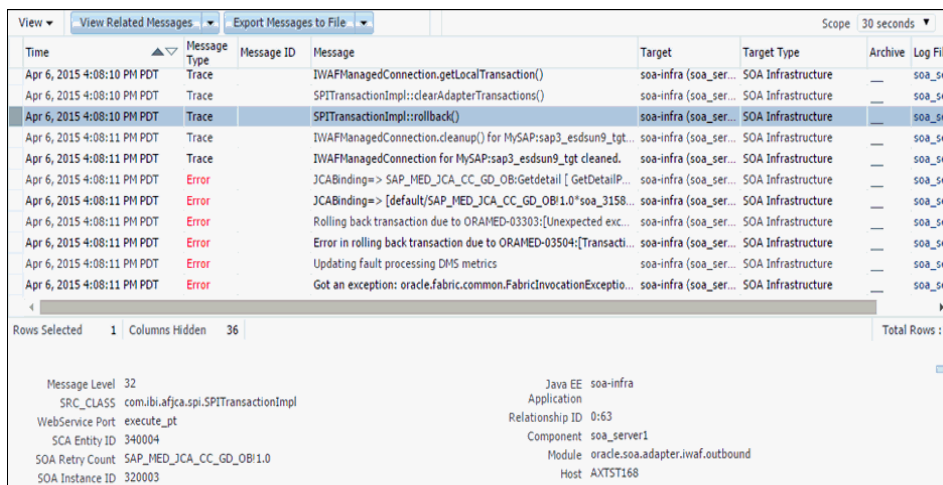


図 9-11 に示すように、詳細が表示されます。

図 9-11 メッセージの詳細



9.1.2 Business Services Engine (BSE) のログ・ファイル管理の構成

J2CA と同様に、BSE の場合も、ログ・ファイル管理は次の場所で定義されるログ出力によって制御されます。

```
<ORACLE_HOME>\user_projects\domains\base_
domain\config\fmwconfig\servers\${server-name}\logging.xml
```

次の構文により、「oracle.soa.adapter.ibse」の下にあるすべてのログ出力のロギング・レベルが、Oracle によるデフォルトの設定レベルである「NOTIFICATION:1 (INFO)」に設定されます。

```
<logger name='oracle.soa.adapter.ibse' level='NOTIFICATION:1'
useParentHandlers='true' />
```

次のステップを実行して、すべてのログ出力のロギング・レベルを EM コンソールから構成することもできます。

- 9.1.1 項「J2CA コネクタ・アプリケーションのログ・ファイル管理の構成」の説明に従って、ステップ 1 から 5 を繰り返します。
- 図 9-12 に示すように、「ログ・レベル」タブの下で、ログ出力名「oracle.soa.adapter.ibse」が表示されるまで Oracle ルート・ログ出力を展開します。

図 9-12 「ログ・レベル」タブ

Logger Name	Oracle Diagnostic Logging Level (Java Level)	Log File	Persistent Log Level State
oracle.soa.adapter.aq	ERROR:1 (SEVERE) [Inherited f ▼]	odi-handler	
oracle.soa.adapter.coherence	ERROR:1 (SEVERE) [Inherited f ▼]	odi-handler	
oracle.soa.adapter.db	ERROR:1 (SEVERE) [Inherited f ▼]	odi-handler	
oracle.soa.adapter.file	ERROR:1 (SEVERE) [Inherited f ▼]	odi-handler	
oracle.soa.adapter.ftp	ERROR:1 (SEVERE) [Inherited f ▼]	odi-handler	
oracle.soa.adapter.ibse	NOTIFICATION:1 (INFO) ▼	odi-handler	NOTIFICATION:1
oracle.soa.adapter.iwaf	TRACE:32 (FINEST) ▼	odi-handler	ERROR:1
oracle.soa.adapter.jdeworld	ERROR:1 (SEVERE) [Inherited f ▼]	odi-handler	
oracle.soa.adapter.jms	ERROR:1 (SEVERE) [Inherited f ▼]	odi-handler	
oracle.soa.adapter.ldap	ERROR:1 (SEVERE) [Inherited f ▼]	odi-handler	
oracle.soa.adapter.mq	ERROR:1 (SEVERE) [Inherited f ▼]	odi-handler	

- 図 9-13 に示すように、「Oracle Diagnostic Logging レベル (Java レベル)」列で、「oracle.soa.adapter.ibse」ドロップダウン・リストから必要なログ・レベルを選択します。

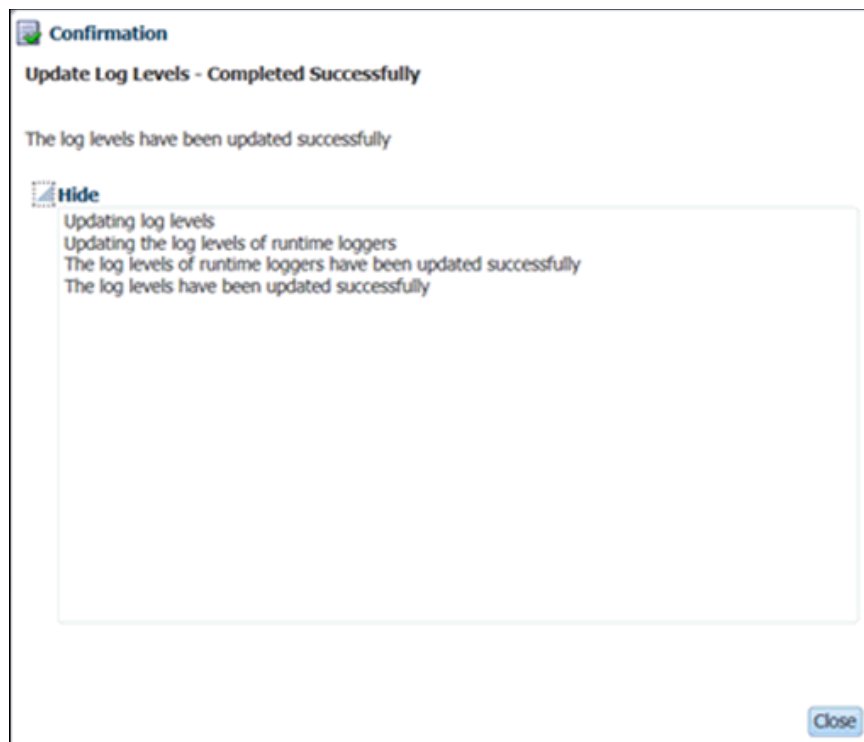
図 9-13 「Oracle Diagnostic Logging レベル (Java レベル)」列

Logger Name	Oracle Diagnostic Logging Level (Java Level)	Log File	Persistent Log Level State
oracle.soa.adapter.aq	ERROR:1 (SEVERE) [Inherited f ▼]	odi-handler	
oracle.soa.adapter.coherence	ERROR:1 (SEVERE) [Inherited f ▼]	odi-handler	
oracle.soa.adapter.db	ERROR:1 (SEVERE) [Inherited f ▼]	odi-handler	
oracle.soa.adapter.file	ERROR:1 (SEVERE) [Inherited f ▼]	odi-handler	
oracle.soa.adapter.ftp	ERROR:1 (SEVERE) [Inherited f ▼]	odi-handler	
oracle.soa.adapter.ibse	NOTIFICATION:1 (INFO) ▼	odi-handler	NOTIFICATION:1
oracle.soa.adapter.iwaf	INCIDENT_ERROR:1 (SEVERE+100)	odi-handler	ERROR:1
oracle.soa.adapter.jdeworld	ERROR:1 (SEVERE) [Inherited f ▼]	odi-handler	
oracle.soa.adapter.jms	WARNING:1 (WARNING)	odi-handler	
oracle.soa.adapter.jms	NOTIFICATION:1 (INFO)	odi-handler	
oracle.soa.adapter.jms	NOTIFICATION:16 (CONFIG)	odi-handler	
oracle.soa.adapter.jms	NOTIFICATION:32	odi-handler	
oracle.soa.adapter.jms	TRACE:1 (FINE)	odi-handler	
oracle.soa.adapter.jms	TRACE:16 (FINER)	odi-handler	
oracle.soa.adapter.jms	TRACE:32 (FINEST)	odi-handler	
oracle.soa.adapter.jms	Inherited from parent	odi-handler	

- 「適用」をクリックします。

図 9-14 に示すように、更新が正常に完了したことを示す確認メッセージが表示されます。

図 9-14 確認メッセージ



5. 「閉じる」をクリックします。

次の表は、「ログ・レベル」プロパティが `{server-name}-diagnostic.log` で EM コンソール内の対応する「ログ・レベル」プロパティ設定に基づいてどのように更新されるかを示しています。

表 9-2 「ログ・レベル」プロパティ設定

EM コンソールで設定された「ログ・レベル」	<code>{server-name}-diagnostic.log</code> で更新された「ログ・レベル」
ERROR:1	ERROR
WARNING:1	WARNING
NOTIFICATION:1	NOTIFICATION
NOTIFICATION:16	NOTIFICATION
NOTIFICATION:32	NOTIFICATION
TRACE:1	NOTIFICATION
TRACE:16	NOTIFICATION
TRACE:32	TRACE:32

ログの検証については、[9.1.1 項「J2CA コネクタ・アプリケーションのログ・ファイル管理の構成」](#)に示す J2CA ロギング機能のステップ 11 から 18 を参照してください。

9.2 診断機能の構成

この項では、Oracle Fusion Middleware Application Adapters for Oracle WebLogic Server の診断機能を構成および使用方法について説明します。次のトピックが含まれています。

- 9.2.1 項「サポートされるプロトコル」

診断機能は、アダプタのエンドポイント・ヘルス・ステータスを取得して (利用可能な場合)、対応するアラートを Oracle アダプタ・フレームワークに提供します。これにより、そのアラートを EM コンソールに表示できるようになります。

注意： 診断機能でサポートされているのはインバウンド・アダプタ・プロセスのみです。

次の項に進む前に、インバウンド・プロセスがデプロイされていることを確認してください。

この情報をアダプタについて表示するには、次のステップを実行します。

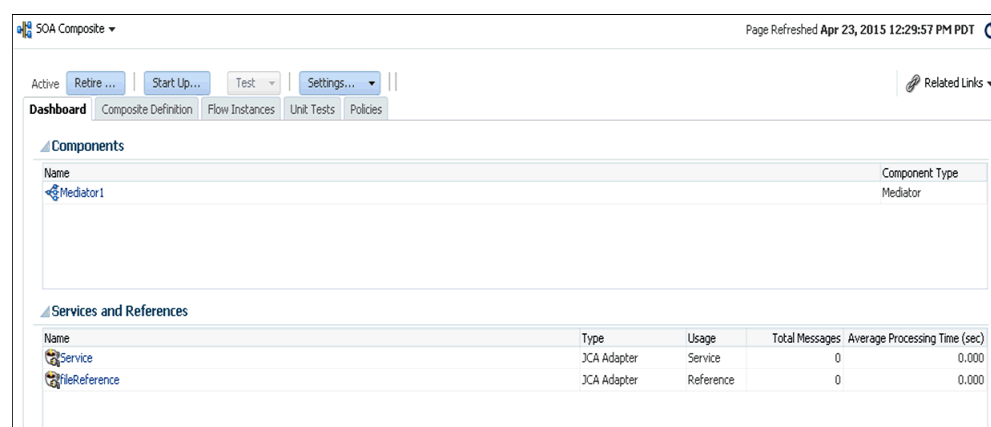
1. Oracle WebLogic Server を起動し、Web ブラウザで次の URL を入力して、Oracle WebLogic Server Enterprise Manager Console を起動します。

`http://host name:port/em`

ここで、*host name* は Oracle WebLogic Server が稼働しているシステムの名前で、*port* は稼働している Oracle WebLogic Server のポートです。

2. 管理者権限を持つアカウントを使用して、Oracle WebLogic Server 管理コンソールにログインします。
3. 右ペインで、「SOA」、「soa-Infra (server_name)」、「デフォルト」、デプロイ済インバウンド・プロセスの順に展開します。
4. プロセスをクリックし、[図 9-15](#) に示すように、右ペインの「サービスと参照」セクションに表示されているサービスを選択します。

図 9-15 「サービスと参照」セクション



5. 「アダプタ・レポート」タブで、「レポートの有効化」チェック・ボックスを選択し、「スナップショット・レポート」を展開して、EIS の停止時間の詳細を表示します。[図 9-16](#) に示すように、「エラー・サマリー」をクリックするとスタック・トレースが表示されます。

図 9-16 スタック・トレース

The screenshot shows the SOA Composite console interface. The top navigation bar includes 'Dashboard', 'Policies', 'Properties', and 'Adapter Reports'. The 'Adapter Reports' section is active, showing 'Diagnosis Reports' with 'Enable reports' checked. Below this, there are sections for 'Configuration Reports' and 'Monitoring Reports'. A table displays real-time monitoring statistics for the endpoint, with columns for Node, EIS Connection Status, Managed Connections (Currently Open, Average Number Used, Currently Free, Maximum Pool Size), and Most Recent Time Stamp (Last Message Publication, Last Service Activation). Below the monitoring table is the 'Snapshot Reports' section, which includes a 'Write to Database' checkbox and a 'Retrieve Data' section with a 'Recent time period' dropdown set to 'Last*' and '24' hours. A table titled 'EIS Connection Downtime' shows a single entry for 'soa_server1' with a 'Connection reset' error summary highlighted in a red box. The table columns are Server Name, Start EIS Downtime, End EIS Downtime, Error Summary, Total EIS Connection Downtime (Seconds), and Data Location.

Node	EIS Connection Status	Currently Open	Average Number Used	Currently Free	Maximum Pool Size	Last Message Publication	Last Service Activation
No data to display							

Server Name	Start EIS Downtime	End EIS Downtime	Error Summary	Total EIS Connection Downtime (Seconds)	Data Location
soa_server1	Apr 10, 2015 4:12:20 AM	Apr 10, 2015 4:14:04 AM	Connection reset	103	

9.2.1 サポートされるプロトコル

この項では、Oracle Fusion Middleware Oracle WebLogic Server Application Adapter for J.D. Edwards OneWorld でサポートされるプロトコルについて説明します。次のトピックで構成されています。

- 9.2.1.1 項「Oracle Fusion Middleware Oracle WebLogic Server Application Adapter for J.D. Edwards OneWorld のエンドポイント・ステータス」

9.2.1.1 Oracle Fusion Middleware Oracle WebLogic Server Application Adapter for J.D. Edwards OneWorld のエンドポイント・ステータス

Oracle Fusion Middleware Oracle WebLogic Server Application Adapter for J.D. Edwards OneWorld (インバウンド) で使用できるプロトコルは次のとおりです。

- HTTP
- TCP
- ファイル (サポートされていません)

HTTP と TCP のリスニング・プロトコル・アダプタは、ソケットでリスニングします。そのため、その性質上、ネットワークの反対側で何かが動作しているかどうかは、リクエストが受信されるまでわかりません。リクエストが受信された場合も、そのリクエストの送信元を確実に特定することはできません。

これらのプロトコルでは、リクエストが受信されると、送信元は EIS と見なされ、EIS が使用されます。EIS は、リクエストの受信時に通信エラーが発生していないかどうかを確認します。ただし、この確認はごく基本的なものにとどまります。

一方、Oracle Fusion Middleware Oracle WebLogic Server Application Adapter for J.D. Edwards OneWorld では、TCP または HTTP のリクエストが受信されると、J.D. Edwards OneWorld サーバーに接続して詳細な情報が取得されます。これにより、アダプタでより正確なステータスを取得して、実際のサーバーと通信できるかどうかを確認できます。

9.3 SOA デバッグ機能の構成

この項では、Oracle Application Adapter for J.D. Edwards OneWorld の SOA デバッグ機能を構成および使用方法について説明します。次のトピックが含まれています。

- 9.3.1 項「SOA デバッガの使用のガイドライン」
- 9.3.2 項「前提条件」
- 9.3.3 項「Oracle JDeveloper での BPEL プロセスのデバッグ」
- 9.3.4 項「Oracle JDeveloper での OSB プロセスのデバッグ」

注意： 現在 SOA デバッグ機能がサポートされているのは J2CA 構成のみです。BSE 構成では使用できません。

SOA については、この機能の説明には BPEL プロセスを使用します。メディエータ・プロセスと BPM プロセスについても同様です。

9.3.1 SOA デバッガの使用のガイドライン

この項では、SOA デバッガの使用のガイドラインについて説明します。

1. SOA デバッガに接続できるクライアントは一度に 1 つのみです。
2. アダプタ・エンドポイント・エラーは、Oracle JDeveloper の SOA デバッガには表示されません。これらのエラーはログ・ファイルに記録されます。

注意： SOA デバッガは現在、BPEL、メディエータ、BPM、OSB の各プロセスで、開発モードでのみ使用できます。

9.3.2 前提条件

Oracle JDeveloper で *IntegratedWebLogicServer* ドメインと BPEL プロセスがすでに作成されていることを確認してください。

9.3.3 Oracle JDeveloper での BPEL プロセスのデバッグ

この項では、Oracle JDeveloper で BPEL プロセスをデバッグする方法について説明します。次のトピックが含まれています。

- 9.3.3.1 項「Oracle JDeveloper でのアウトバウンド BPEL プロセスのデバッグ」
- 9.3.3.2 項「Oracle JDeveloper でのインバウンド BPEL プロセスのデバッグ」

9.3.3.1 Oracle JDeveloper でのアウトバウンド BPEL プロセスのデバッグ


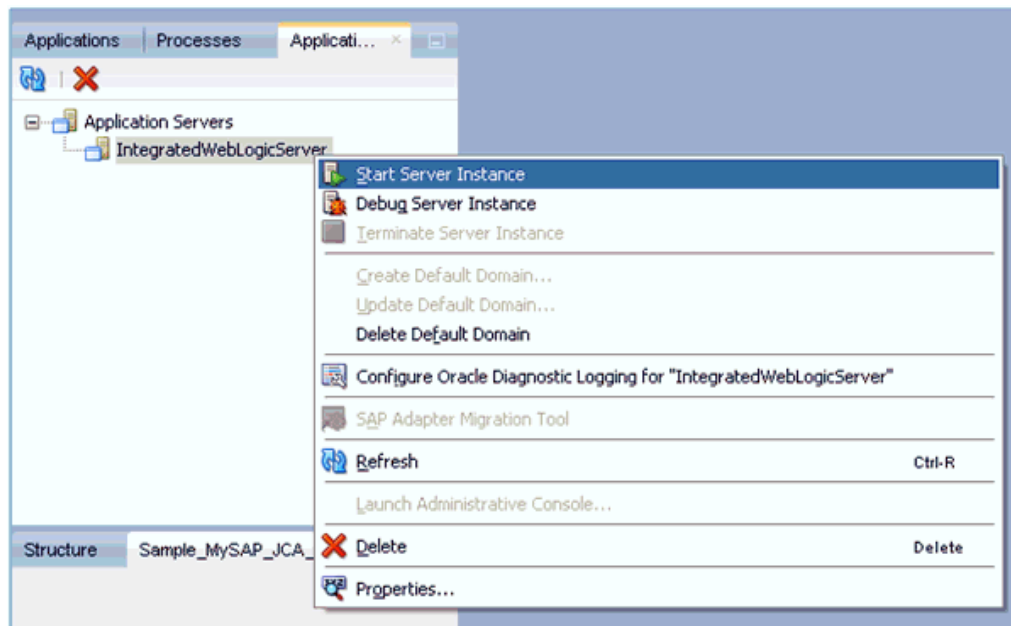
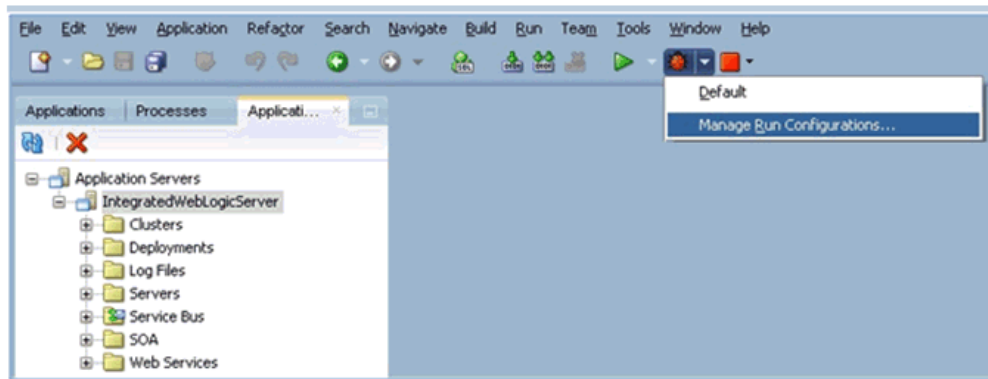
1. Oracle JDeveloper を開きます。
2. *IntegratedWeblogicServer* ドメインを起動します。
 - a. 左ペインの「アプリケーション・サーバー」タブをクリックします。
 - b.  9-17 に示すように、「アプリケーション・サーバー」ノードの下の「IntegratedWeblogicServer」を右クリックし、コンテキスト・メニューから「サーバー・インスタンスの起動」を選択します。

図 9-17 「サーバー・インスタンスの起動」



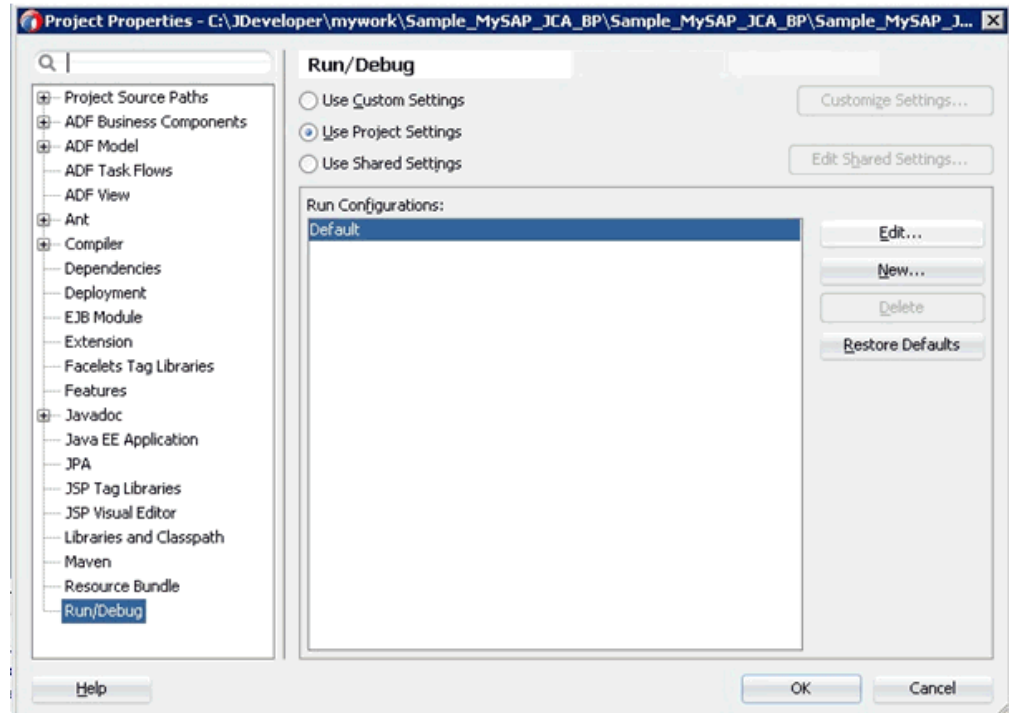
3. デバッグ環境を設定します。
 - a. 図 9-18 に示すように、「デバッグ」アイコンの横の下矢印をクリックし、コンテキスト・メニューから「実行構成の管理」を選択します。

図 9-18 「実行構成の管理」



- b. または、プロジェクトを右クリックし、「プロジェクト・プロパティ」を選択します。
- 図 9-19 に示すように、「プロジェクト・プロパティ」ダイアログが表示されます。

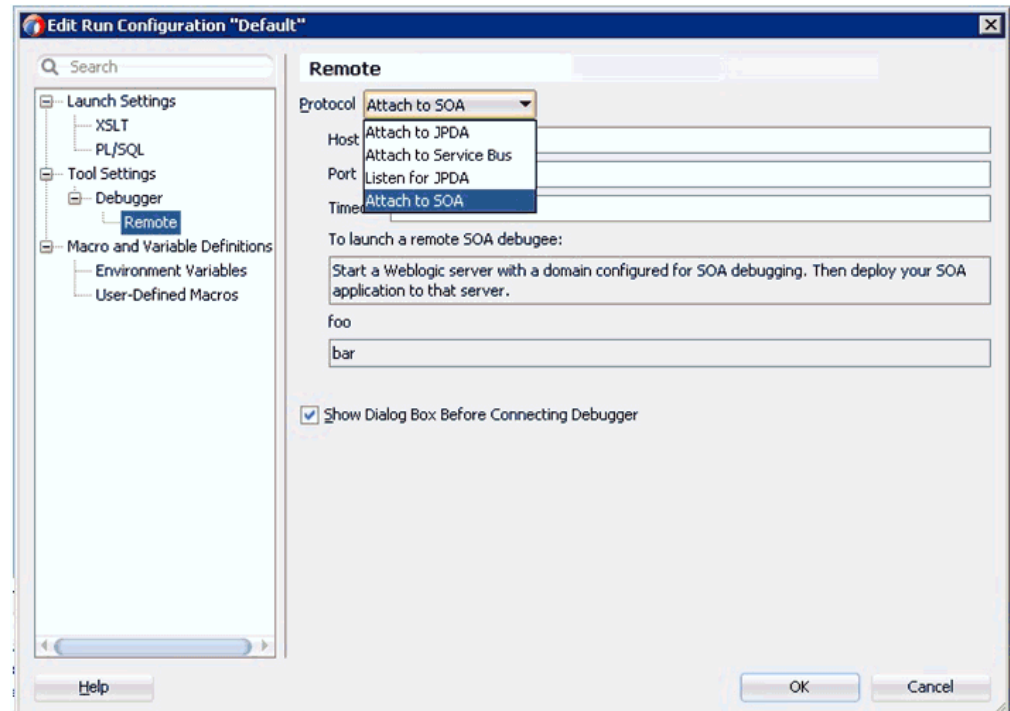
図 9-19 「プロジェクト・プロパティ」ダイアログ



- c. 左ペインで「実行/デバッグ」を選択し、「実行/デバッグ」ペイン(「実行構成」領域)で「編集」をクリックします。

図 9-20 に示すように、「実行構成の編集」ダイアログが表示されます。

図 9-20 「実行構成の編集」ダイアログ




- d. 左ペインで、「ツール設定」、「デバッガ」の順に展開し、「リモート」をクリックします。
 - e. 「プロトコル」リストから、「SOA にアタッチ」を選択します。
 - f. 「ホスト」と「ポート」はデフォルト値のままにしておきます。
 - g. 「OK」をクリックします。
4. プロジェクトをデプロイします。
 - a. プロジェクトを右クリックし、「デプロイ」を選択します。
「デプロイメント・アクション」ダイアログが表示されます。
デプロイするアプリケーションを選択し、「次へ」をクリックします。
「構成のデプロイ」ダイアログが表示されます。
 - b. 「次へ」をクリックします。
「アプリケーション・サーバー」ダイアログが表示されます。
 - c. 「IntegratedWebLogicServer」を選択し、「終了」をクリックしてデプロイを完了します。
 - d. 次のステップに進む前に、プロジェクトのデプロイが完了し、エラーや問題が発生していないことを確認してください。
 5. BPEL プロセスを SOA デバッガに接続します。
 - a.  9-21 に示すように、左ペインの「アプリケーション」タブで、既存の BPEL プロセスのコンポジット XML またはコンポジット・プロジェクトを右クリックし、コンテキスト・メニューから「デバッグ」を選択します。

図 9-21 デバッグする BPEL プロセスを選択

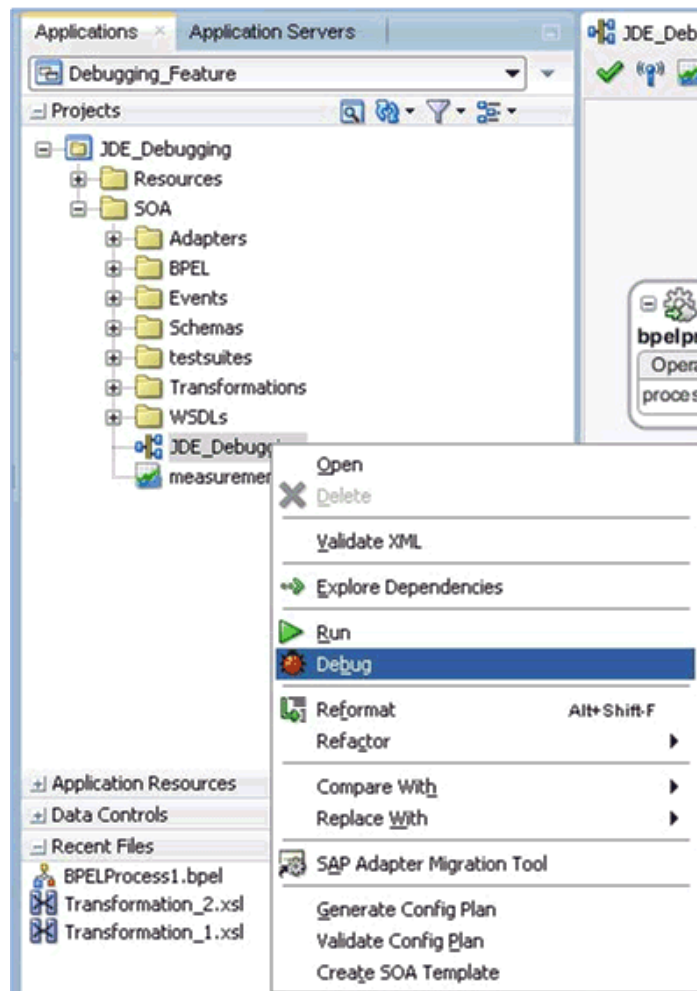
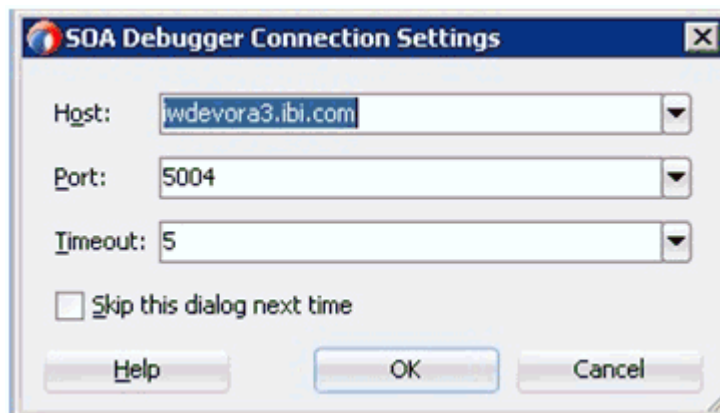


図 9-22 に示すように、「SOA デバッガ接続設定」ダイアログが表示されます。

図 9-22 「SOA デバッガ接続設定」ダイアログ



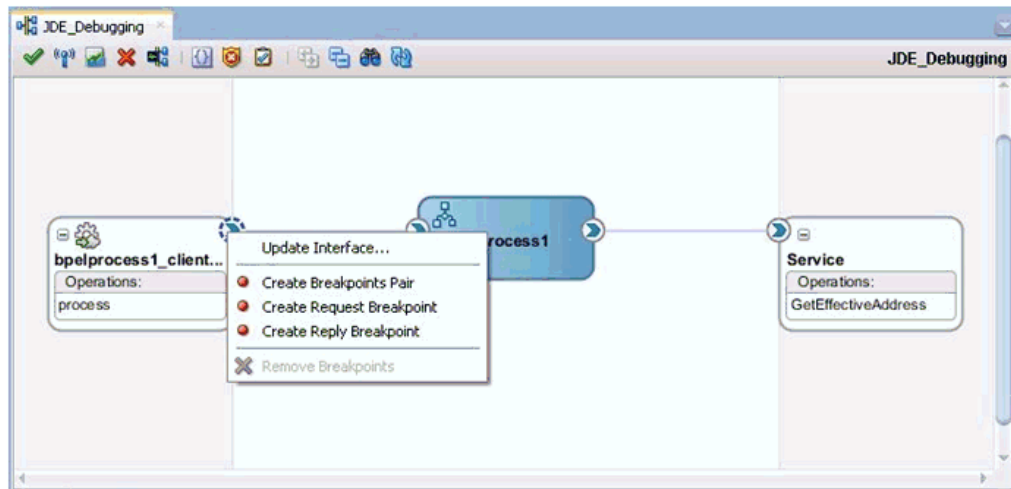
b. 「OK」をクリックします。

BPEL プロセスが SOA デバッガに接続されると、デバッグ・ログに次のメッセージが表示されます。

デバッガは iwdevora3.ibi.com 5004 でリモート・プロセスに接続しようとしています。
デバッガは iwdevora3.ibi.com 5004 でリモート・プロセスに接続しました。
デバッガ・プロセスの仮想マシンは SOA デバッガです。

6. ブレークポイントを設定してデバッグを開始します。
 - a. 図 9-23 に示すように、コンポーネントを右クリックし、設定するブレークポイント・タイプを選択します。

図 9-23 ブレークポイントの選択



「ブレークポイント・ペアの作成」: リクエスト/リプライ (アウトバウンド/インバウンド) 相互作用に対してはこのブレークポイント・タイプを設定します。リクエストとリプライの両方が重要となるシナリオに便利です。

「リクエスト・ブレークポイントの作成」: リクエスト (アウトバウンド) 相互作用に対してはこのブレークポイント・タイプを設定します。リクエストのみが重要となるシナリオに便利です。

「リプライ・ブレークポイントの作成」: リプライ (インバウンド) 相互作用に対してはこのブレークポイント・タイプを設定します。リプライのみが重要となるシナリオに便利です。

「WS デバッグの開始」: デバッグ・セッションを開始するにはこのブレークポイント・タイプを設定します。デバッグ・セッションには、たとえば、Web サービスから BPEL プロセスへの開始 SOAP リクエストから、アダプタ参照バインディング・コンポーネントまでが含まれます。

- b. ブレークポイントを設定したら、図 9-24 に示すように、右側のハンドルを右クリックして「WS デバッグの開始」を選択します。

図 9-24 「WS デバッグの開始」

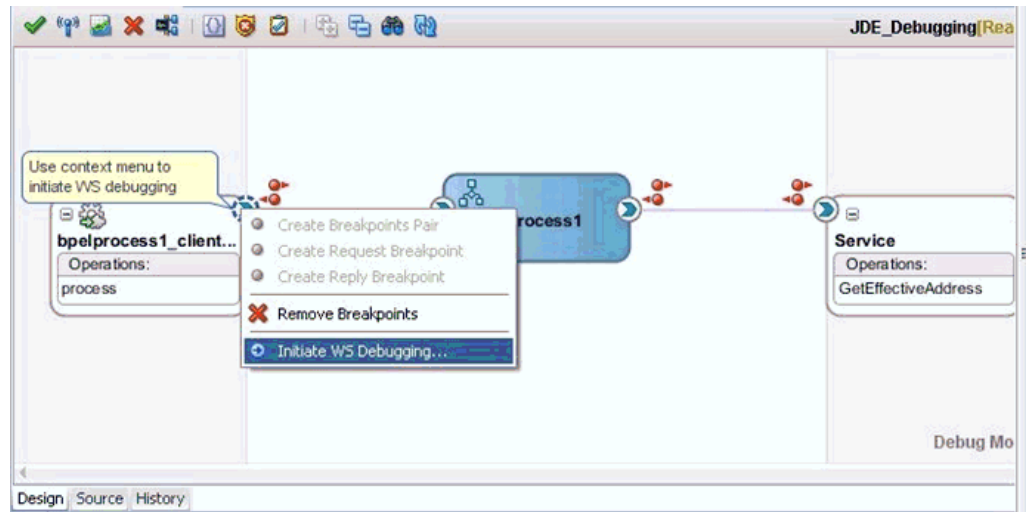
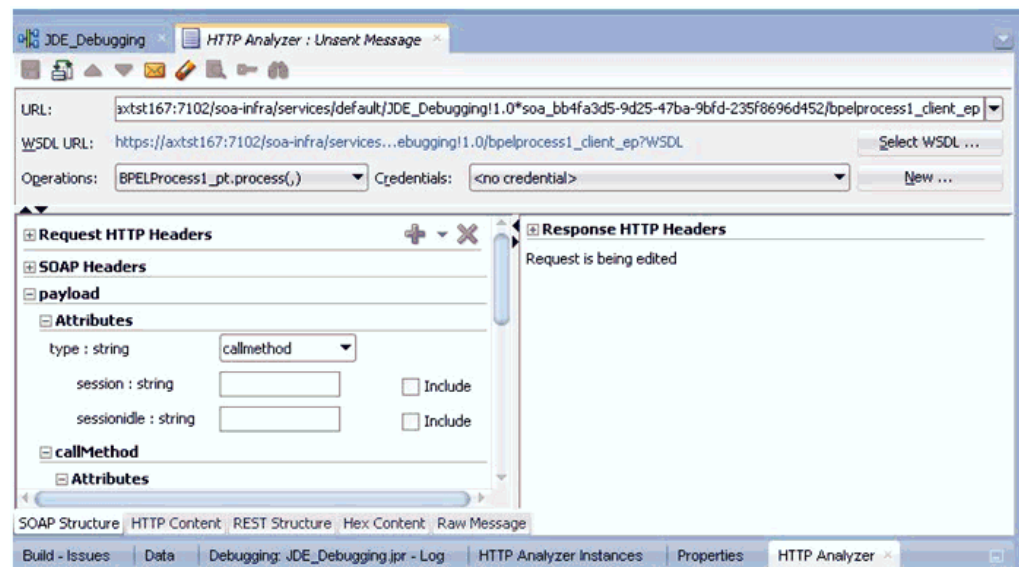


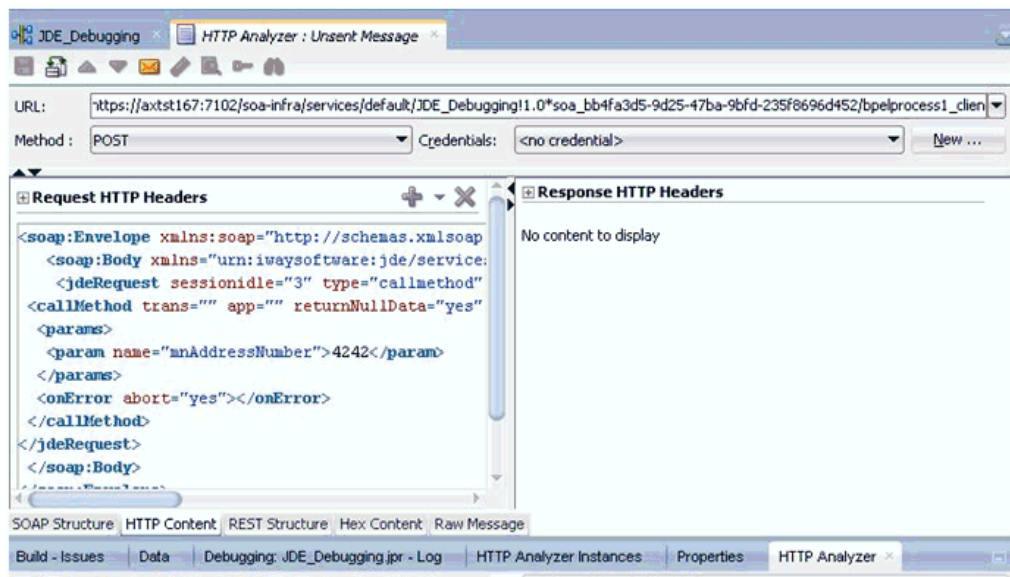
図 9-25 に示すように、「HTTP アナライザ」ダイアログが表示されます。

図 9-25 「HTTP アナライザ」ダイアログ



- c. 下のタブから「HTTP コンテンツ」を選択します。ここで、図 9-26 に示すように、ペイロードをコピーして本文に貼り付けます。

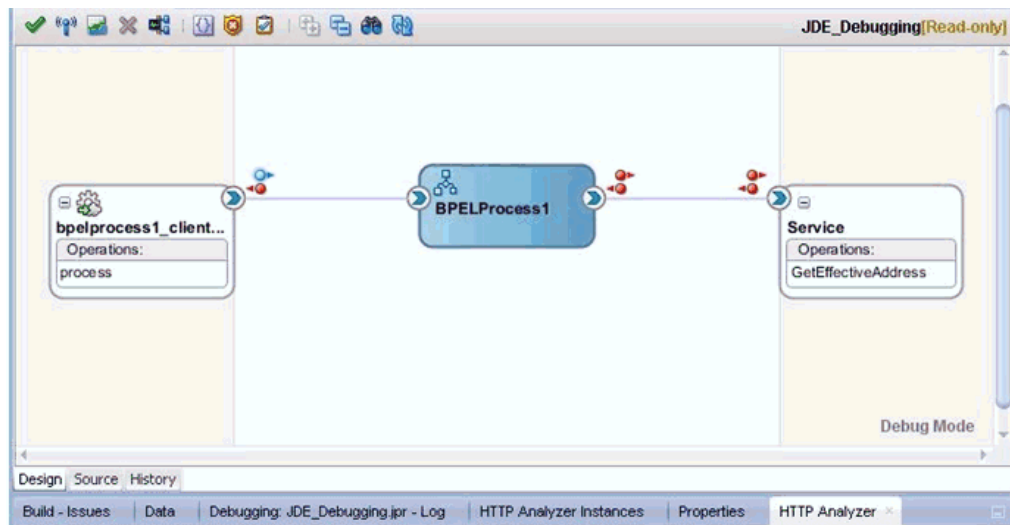
図 9-26 「HTTP コンテンツ」を選択



d. 「リクエストの送信」をクリックします。

図 9-27 に示すように、指定したブレークポイントで BPEL プロセスが停止し、青く点滅します。

図 9-27 BPEL プロセスがブレークポイントで停止



e. 図 9-28 に示すように、使用可能な「ステップ」オプションを使用してデバッグ・プロセスをステップ実行します。

図 9-28 「ステップ」オプション



アイコン	説明
	デバッグ・セッションを終了または連結解除します。
	フレームをステップ実行します。 これにより、次のブレークポイント (ブレークポイントが設定された BPEL プロセスの受信アクティビティなど) に移動します。ブレークポイントがない場合は、すべてのフレームをステップ実行して最初のフレームに戻ります。 [F8] を押してフレームをステップ実行することもできます。
	次の有効な位置までトレース実行します。 これは、新しいフレームである場合もあれば、同じフレームの別の位置である場合もあります。 [F7] を押してフレームをトレース実行することもできます。
	フレームをステップ・アウトします。 このオプションは、BPEL の Scope アクティビティまたは Sequence アクティビティを処理するためにのみ使用されます。スコープの処理が完了すると、プロセスの次のスコープまたはアクティビティで一時停止します。[Shift] + [F7] を押すこともできます。
	ステップ操作を再開します。 [F9] を押して再開することもできます。

7. リクエストのペイロードとヘッダー情報を表示します。

- a. 「ウィンドウ」をクリックし、「デバッガ」、「データ」の順に選択します。

Oracle Application Adapter for J.D. Edwards OneWorld のリクエスト・ブレークポイントで停止したときのヘッダー情報のサンプルを図 9-29 に示します。

図 9-29 ヘッダー情報

Name	Value	Type	Declared Type	Hex Value	Address	ID	Count
completi...	null						
ecid	cb27e7f5-b76...						
masterCo...	null						
MESH_ME...	null						
nativePayl...							
replyToPo...	null						
transaction	Name=[EJB c...						
transport...	192.168.128....						
XID	BEA1-64D03C...						

Oracle Application Adapter for J.D. Edwards OneWorld のレスポンス・ペイロードのサンプルを図 9-30 に示します。

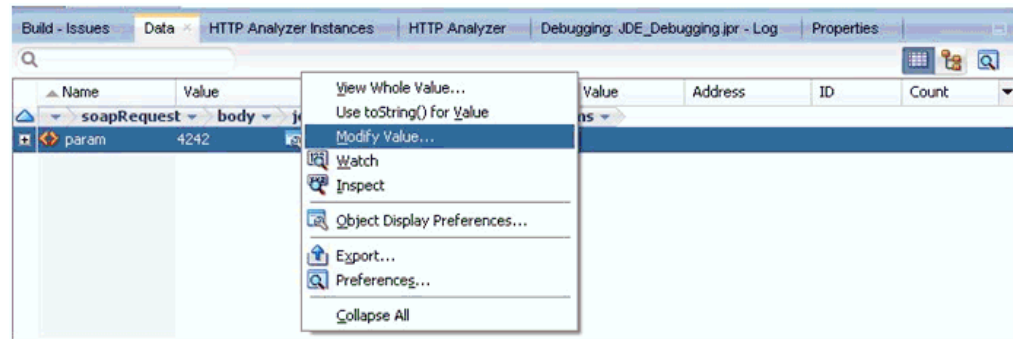
図 9-30 レスポンス・ペイロードのサンプル



注意： 図 9-30 のペイロードの表示は、画面サイズによって制限されています。実際には、BPEL プロセスのブレイクポイントでは完全なペイロードが表示されます。画面をスクロールすると、ペイロードのすべての要素を表示できます。

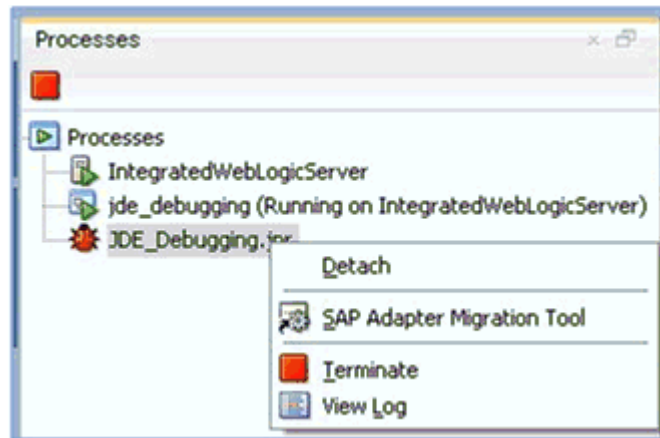
8. リクエスト・ペイロードのコンテンツを変更します。
 - a. 図 9-31 に示すように、SOAP リクエストを展開し、変更するフィールドを右クリックして、コンテキスト・メニューから「値の変更」を選択します。

図 9-31 「値の変更」



9. デバッグ・セッションを終了または連結解除します。
 - a. 「ウィンドウ」、「プロセス」の順にクリックします。図 9-32 に示すように、「プロセス」タブでプロセスを右クリックし、「連結解除」または「終了」を選択します。

図 9-32 デバッグ・セッションの連結解除



- b. 次のいずれかのオプションを選択します。

「連結解除」: デバッグ・プロセスを終了せずに SOA デバッガを削除します。

「終了」: デバッグ・プロセスを終了します。

9.3.3.2 Oracle JDeveloper でのインバウンド BPEL プロセスのデバッグ

1. インバウンド BPEL プロセスをデプロイします。

- a. プロジェクトを右クリックし、「デプロイ」を選択します。

「デプロイメント・アクション」ダイアログが表示されます。

- b. 「アプリケーション・サーバーへのデプロイ」を選択し、「次へ」をクリックします。

「構成のデプロイ」ダイアログが表示されます。

- c. 「次へ」をクリックします。

「サーバーの選択」ダイアログが表示されます。

- d. デプロイするサーバーを選択し、「終了」をクリックします。

- e. 次のステップに進む前に、プロジェクトのデプロイが完了し、エラーや問題が発生していないことを確認してください。

2. BPEL プロセスを SOA デバッガに接続します。

- a. 図 9-33 に示すように、インバウンド・コンポジットを右クリックし、「デバッグ」を選択します。

図 9-33 デバッグする BPEL プロセスを選択

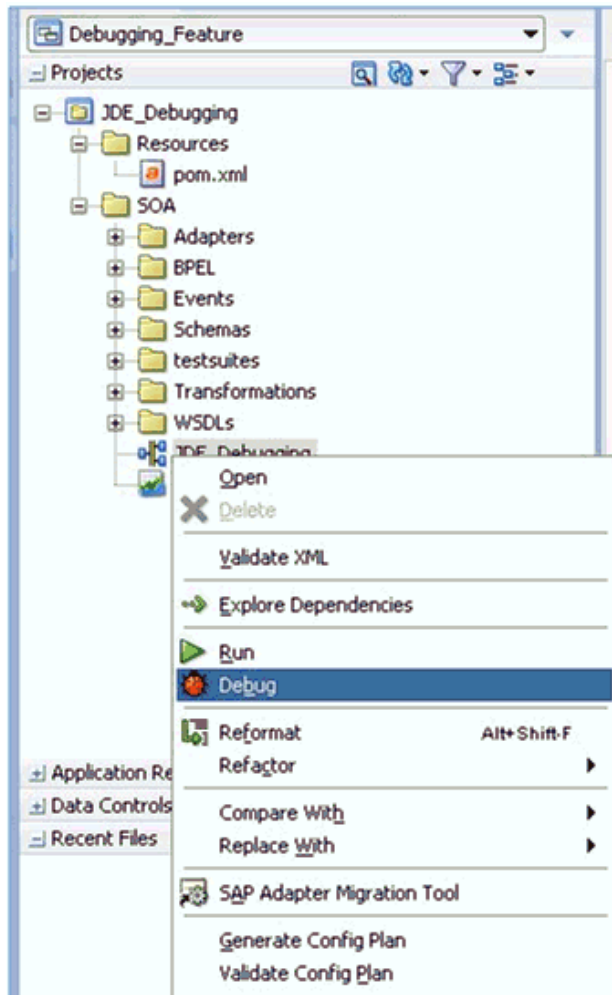
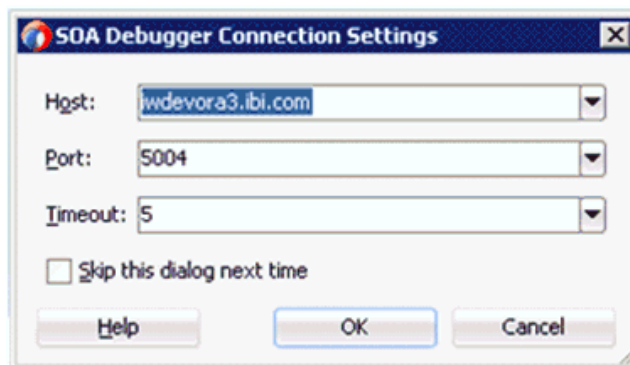


図 9-34 に示すように、「SOA デバッガ接続設定」ダイアログが表示されます。

図 9-34 「SOA デバッガ接続設定」ダイアログ



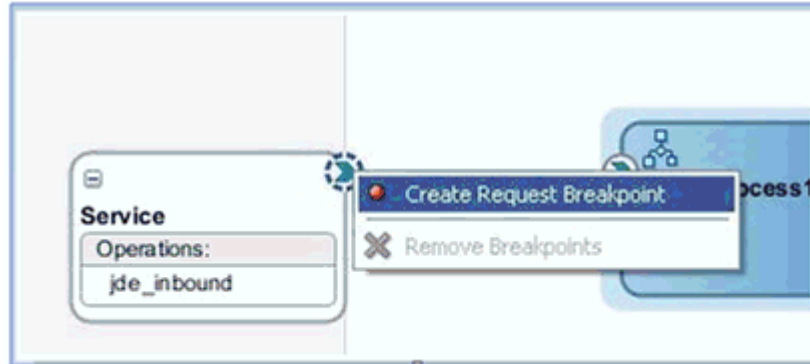
- b. 「OK」をクリックします。

BPEL プロセスが SOA デバッガに接続されると、デバッグ・ログに次のメッセージが表示されます。

デバッガは iwdevora3.ibi.com 5004 でリモート・プロセスに接続しようとしています。
デバッガは iwdevora3.ibi.com 5004 でリモート・プロセスに接続しました。
デバッガ・プロセスの仮想マシンは SOA デバッガです。

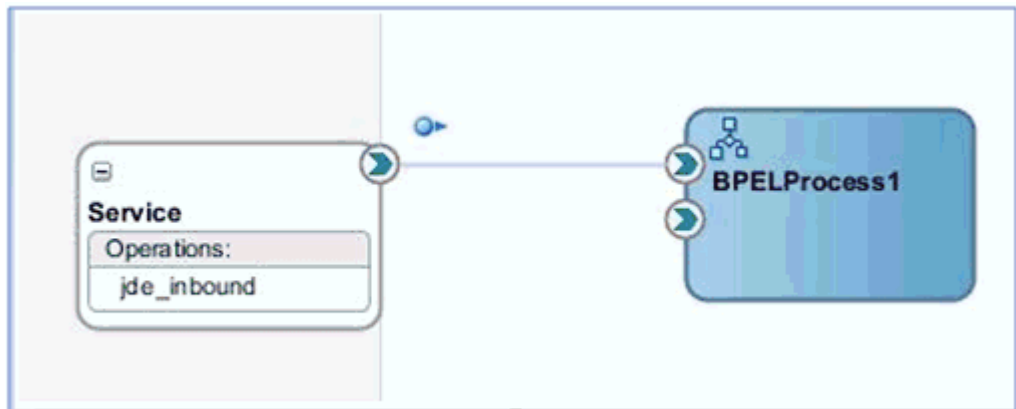
3. 図 9-35 に示すように、ブレークポイントを設定します。

図 9-35 ブレークポイントの設定



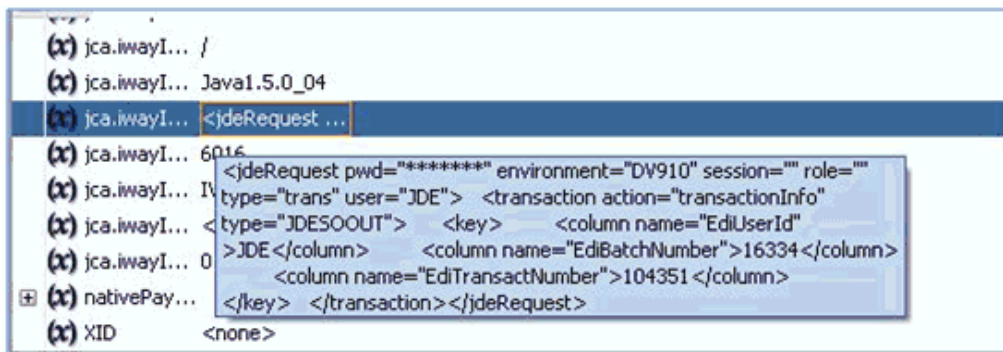
4. ブレークポイントが設定されて、インバウンド・プロセスから (J.D. Edwards OneWorld からトリガーされるなどして) メッセージが受信されると、図 9-36 に示すように、指定したブレークポイントでプロセスが停止します。

図 9-36 ブレークポイントでプロセスを停止



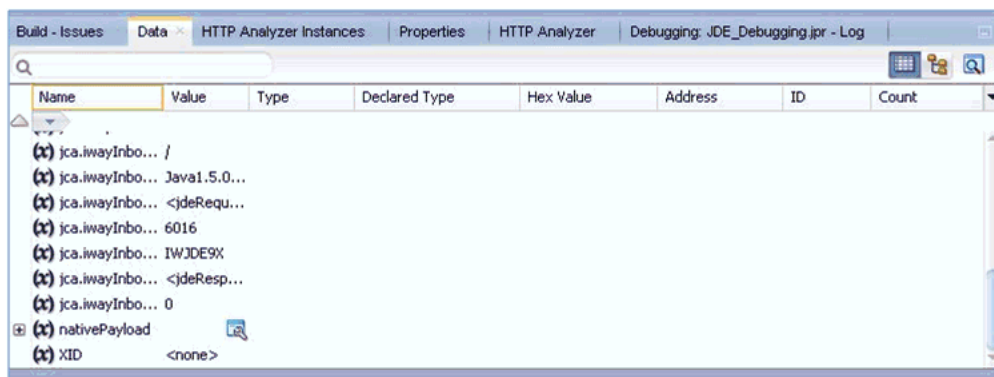
5. リクエストのペイロードとヘッダー情報を表示します。
 - a. 図 9-37 に示すように、「ウィンドウ」をクリックし、「デバッガ」、「データ」の順に選択します。

図 9-37 デバッガのデータ



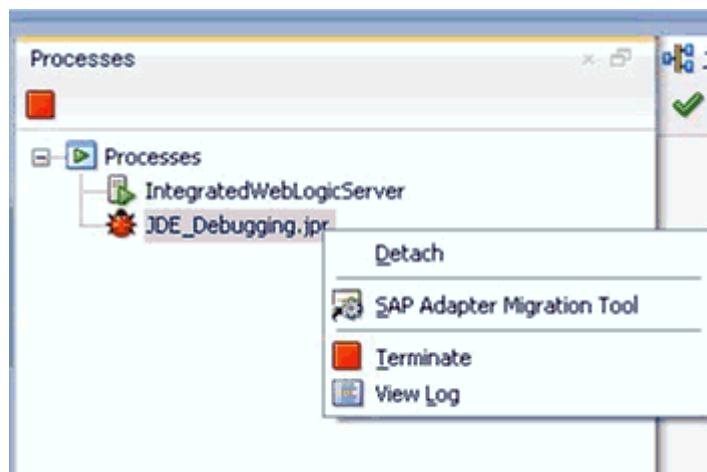
- b. インバウンド・リクエスト・ブレイクポイント (SalesOrder) に対して表示されるヘッダー情報を 図 9-38 に示します。

図 9-38 インバウンド・リクエスト・ブレイクポイント (SalesOrder)



6. デバッグ・セッションを終了または連結解除します。
 - a. 「ウィンドウ」、「プロセス」の順にクリックします。 図 9-39 に示すように、「プロセス」タブでプロセスを右クリックし、「連結解除」または「終了」を選択します。

図 9-39 デバッグ・セッションの連結解除



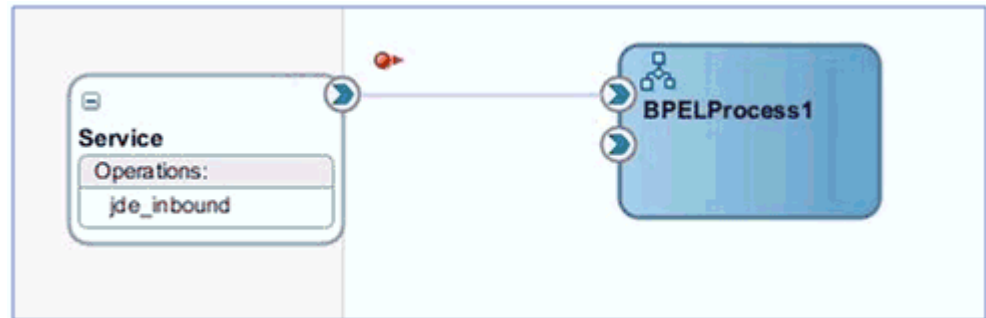
- b. 次のいずれかのオプションを選択します。

「連結解除」: デバッグ・プロセスを終了せずに SOA デバッガを削除します。

「終了」: デバッグ・プロセスを終了します。

図 9-40 に示すように、プロセスが連結解除されて表示されます。

図 9-40 連結解除された BPEL プロセス



9.3.4 Oracle JDeveloper での OSB プロセスのデバッグ

この項では、Oracle JDeveloper で OSB プロセスをデバッグする方法について説明します。次のトピックが含まれています。

- 9.3.4.1 項「前提条件」
- 9.3.4.2 項「Oracle JDeveloper でのアウトバウンド OSB プロセスのデバッグ」
- 9.3.4.3 項「Oracle JDeveloper でのインバウンド OSB プロセスのデバッグ」

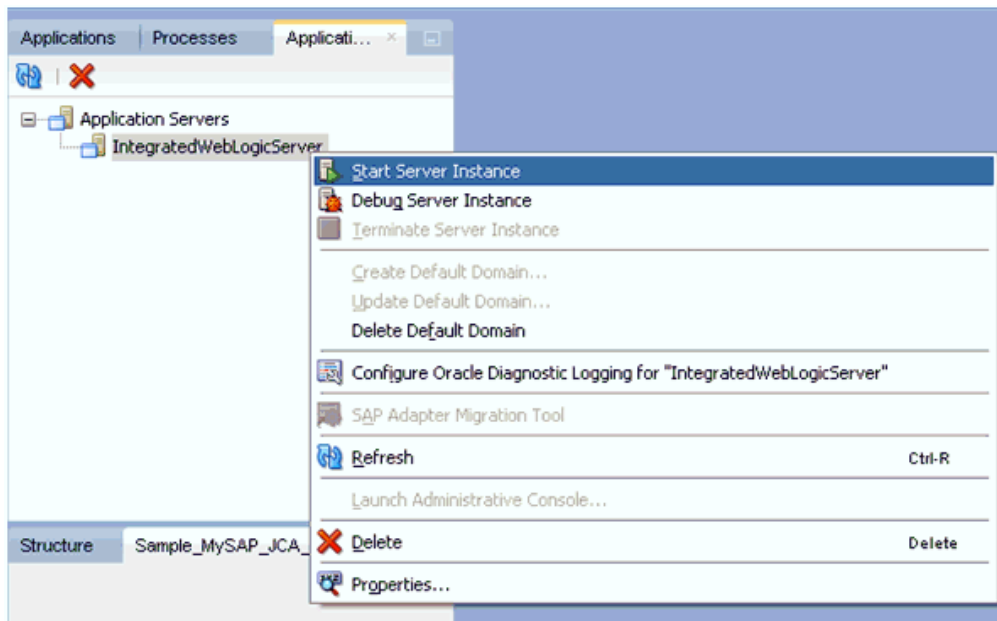
9.3.4.1 前提条件

Oracle JDeveloper で、*IntegratedWebLogicServer* ドメインと、ファイル出力を含む OSB プロセスがすでに作成されていることを確認してください。

注意: アプリケーション名とプロジェクト名が同じであることを確認してください。

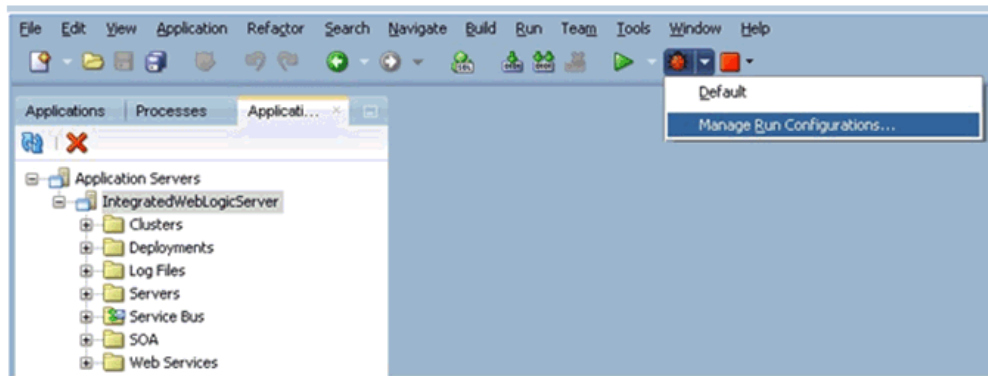
1. Oracle JDeveloper を開き、*IntegratedWeblogicServer* または *startWebLogic.cmd* を起動します。
 - a. 左ペインの「アプリケーション・サーバー」タブをクリックします。
 - b. 図 9-41 に示すように、「アプリケーション・サーバー」ノードの下の「*IntegratedWeblogicServer*」を右クリックし、コンテキスト・メニューから「サーバー・インスタンスの起動」を選択します。

図 9-41 「サーバー・インスタンスの起動」



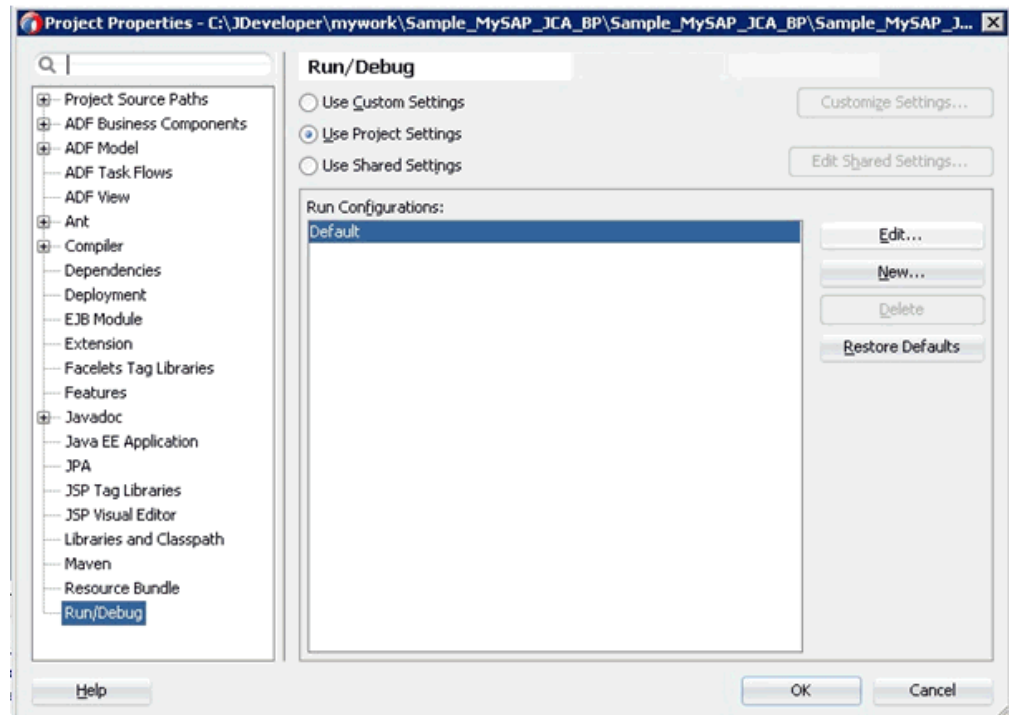
- c. または、`startWebLogic.cmd` を使用して、コマンド・プロンプトから Oracle WebLogic Server を起動します。
2. デバッグ環境を設定します。
 - a. 図 9-42 に示すように、「デバッグ」アイコンの横の下矢印をクリックし、コンテキスト・メニューから「実行構成の管理」を選択します。

図 9-42 「実行構成の管理」



- b. または、プロジェクトを右クリックし、「プロジェクト・プロパティ」を選択します。
- 図 9-43 に示すように、「プロジェクト・プロパティ」ダイアログが表示されます。

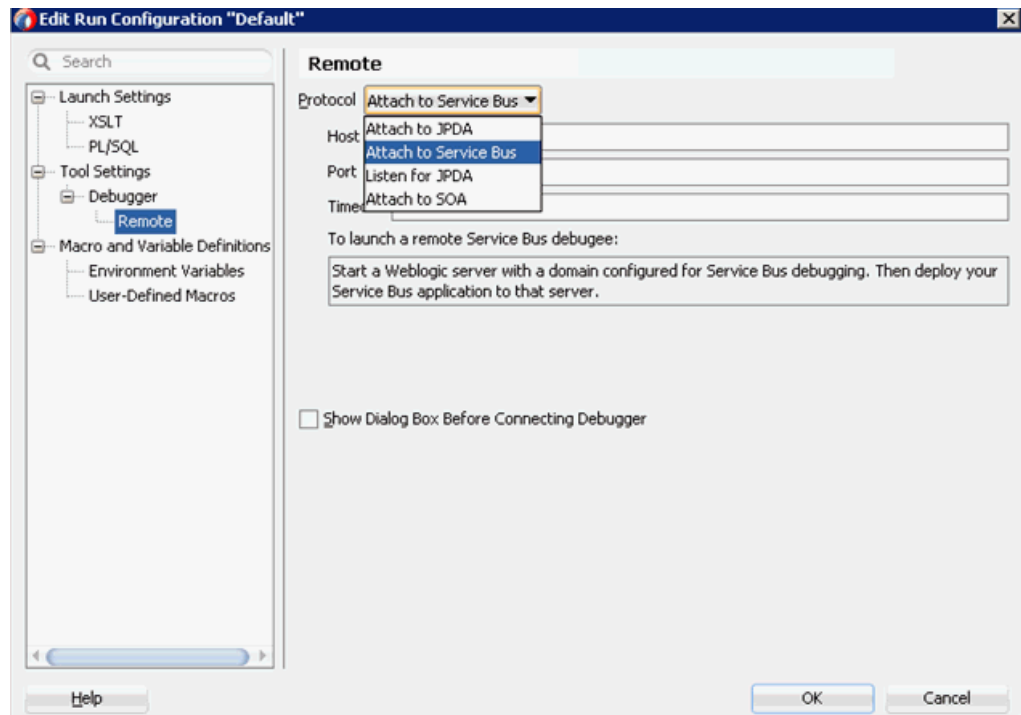
図 9-43 「プロジェクト・プロパティ」ダイアログ



- c. 左ペインで「実行/デバッグ」を選択し、「実行/デバッグ」ペイン（「実行構成」領域）で「編集」をクリックします。

図 9-44 に示すように、「実行構成の編集」ダイアログが表示されます。

図 9-44 「実行構成の編集」ダイアログ



- d. 左ペインで、「ツール設定」、「デバッガ」の順に展開し、「リモート」をクリックします。
- e. 「プロトコル」リストから、「Service Bus にアタッチ」を選択します。
- f. 「ホスト」と「ポート」はデフォルト値のままにしておきます。
- g. 「OK」をクリックします。

9.3.4.2 Oracle JDeveloper でのアウトバウンド OSB プロセスのデバッグ

1. プロジェクトをデプロイします。
 - a. プロジェクトを右クリックし、「デプロイ」を選択します。
「デプロイメント・アクション」ダイアログが表示されます。
 - b. デプロイするアプリケーションを選択し、「次へ」をクリックします。
「構成のデプロイ」ダイアログが表示されます。
 - c. 「次へ」をクリックします。
「アプリケーション・サーバー」ダイアログが表示されます。
 - d. 「IntegratedWebLogicServer」を選択し、「終了」をクリックしてデプロイを完了します。
 - e. 次のステップに進む前に、プロジェクトのデプロイが完了し、エラーや問題が発生していないことを確認してください。
2. OSB プロセスを SOA デバッガに接続します。
 - a. [図 9-45](#) に示すように、左ペインの「アプリケーション」タブで、既存の OSB プロセスのコンポジット XML またはコンポジット・プロジェクトを右クリックし、コンテキスト・メニューから「デバッグ」を選択します。

図 9-45 デバッグする OSB プロセスを選択

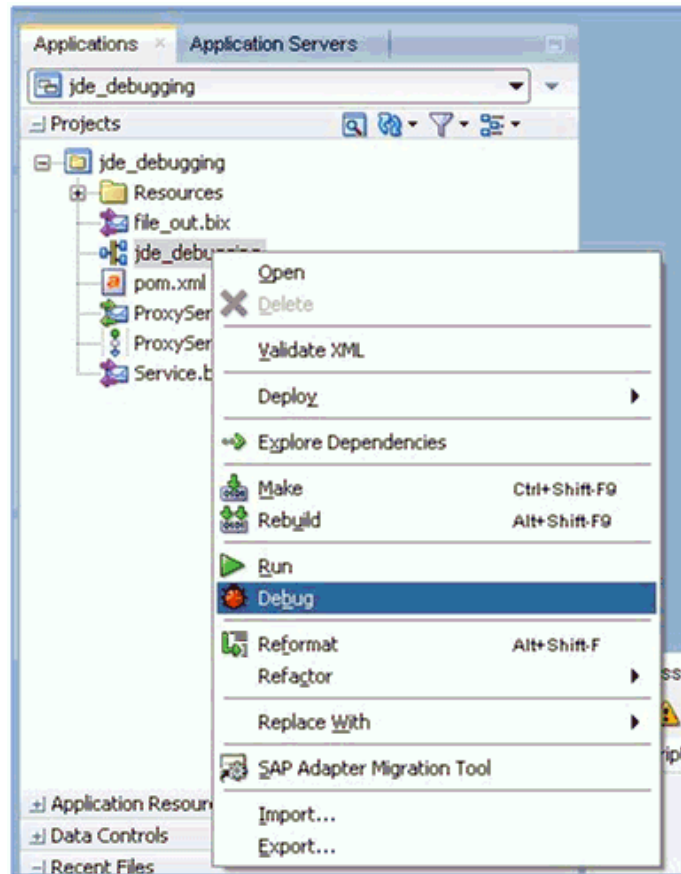
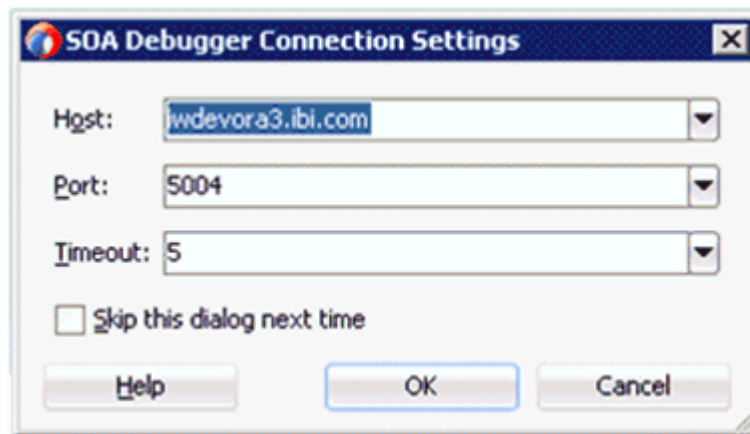


図 9-46 に示すように、「SOA デバッガ接続設定」ダイアログが表示されます。

図 9-46 「SOA デバッガ接続設定」ダイアログ



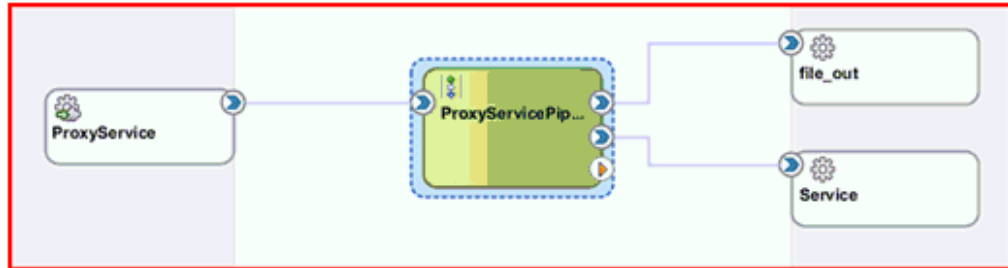
- b. 「OK」をクリックします。

OSB プロセスが SOA デバッガに接続されると、デバッグ・ログに次のメッセージが表示されます。

デバッガは iwdevora3.ibi.com 5004 でリモート・プロセスに接続しようとしています。
 デバッガは iwdevora3.ibi.com 5004 でリモート・プロセスに接続しました。
 デバッガ・プロセスの仮想マシンは SOA デバッガです。

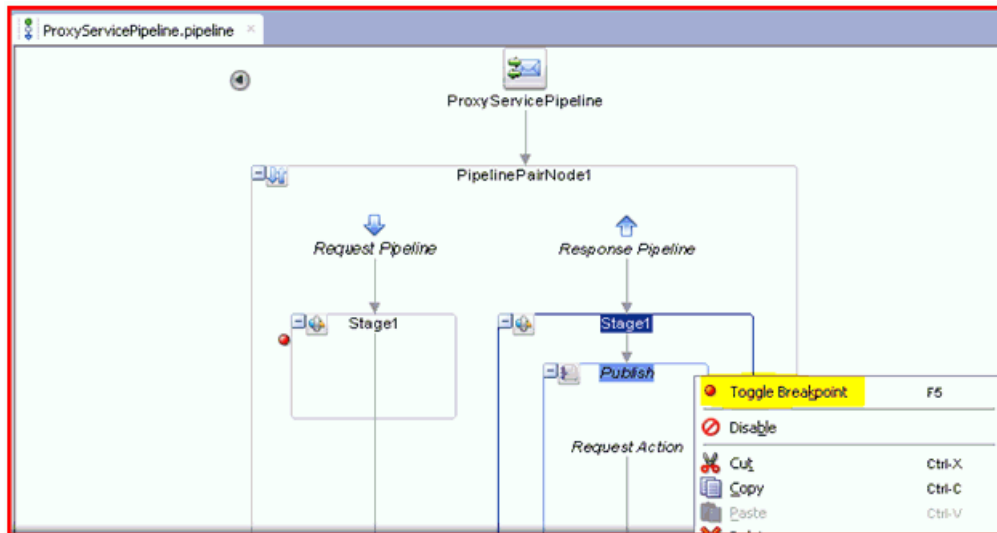
3. ブレークポイントを設定します。
 - a. 図 9-47 に示すように、パイプラインをダブルクリックしてエディタで開きます。

図 9-47 ProxyServicePipeline



- b. ブレークポイントを追加するノードが表示されるまでアクションを展開します。この例では、図 9-48 に示すように、PipelinePairNode1 を展開し、「ステージ 1」ノード（「リクエスト・パイプライン」の下）を右クリックして、コンテキスト・メニューから「ブレークポイントの切替え」を選択します。

図 9-48 「ブレークポイントの切替え」



このステップを「パブリッシュ」ノード（「レスポンス・パイプライン」の下）に対して繰り返します。

ノードの横に、ブレークポイントが設定されたことを示す赤いアイコンが表示されます。

注意： ブレークポイントを無効にするには、ノードを右クリックして「ブレークポイントの無効化」を選択します。

ブレークポイントを削除するには、ノードを右クリックして再度「ブレークポイントの切替え」を選択します。

4. デバッグを開始します。

- a. 図 9-49 に示すように、「アプリケーション・ナビゲータ」でパイプラインを右クリックし、「デバッグ」を選択します。

図 9-49 「デバッグ」を選択

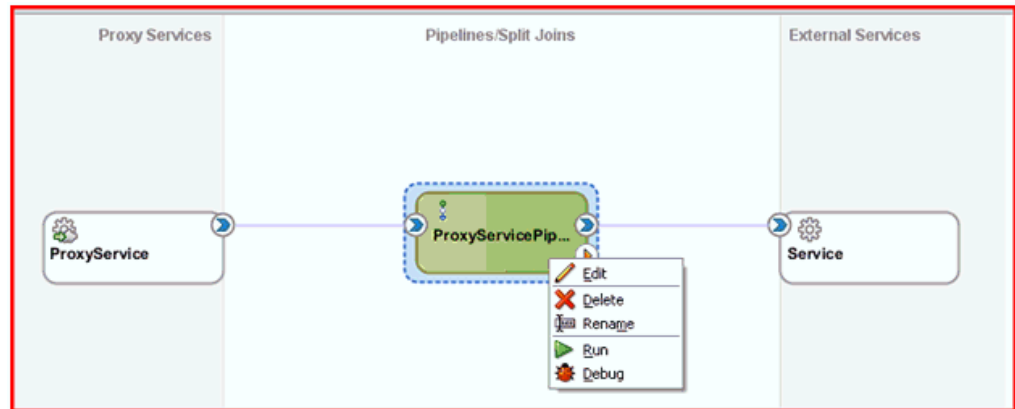
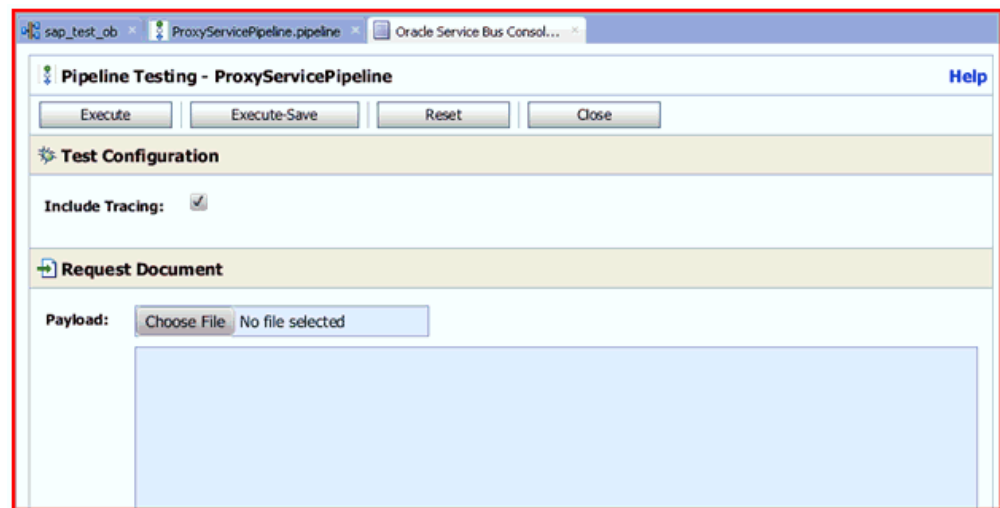


図 9-50 に示すように、プロセスが統合サーバーにデプロイされて、「構成のテスト」ペインが表示されます。

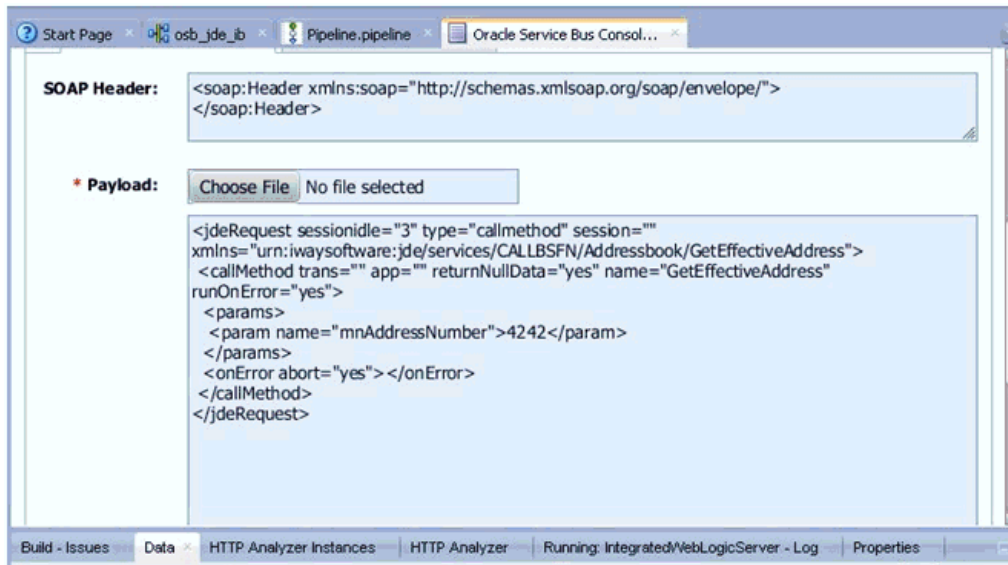
図 9-50 「構成のテスト」ペイン



注意： 現在実行されているドメインがない場合は、「デフォルト・ドメインの作成」ダイアログが表示されます。統合サーバーの接続情報を入力し、「OK」をクリックします。このプロセスには数分かかる場合があります。

- b. 図 9-51 に示すように、「構成のテスト」ペインで、「リクエスト・ドキュメント」領域にテスト・データを入力し、必要に応じて追加の入力を構成します。

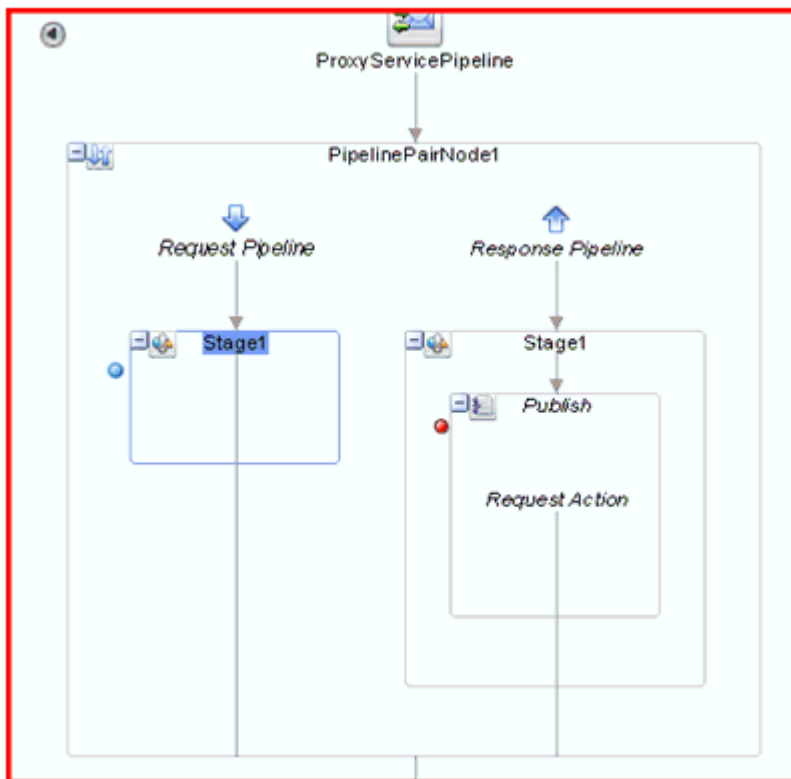
図 9-51 リクエスト・ドキュメントのテスト



- c. 「実行」をクリックします。

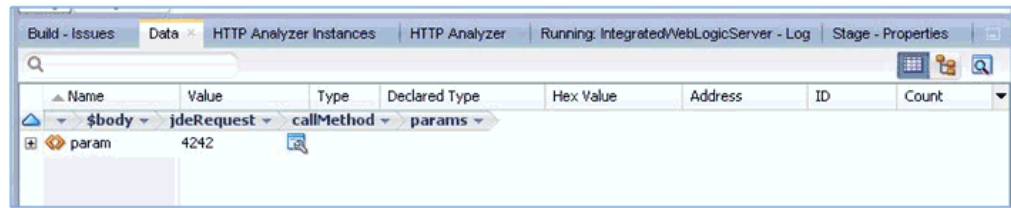
「構成のテスト」ペインでコマンドが実行されますが、図 9-52 に示すように、指定したブレークポイントで OSB プロセスが停止し、青く点滅します。

図 9-52 OSB プロセスがブレークポイントで停止



- d. 図 9-53 に示すように、下部の「データ」タブで、入力が渡されたことを確認します。






図 9-53 「データ」タブ



- e. 図 9-54 に示すように、使用可能な「ステップ」オプションを使用してデバッグ・プロセスをステップ実行します。

図 9-54 「ステップ」オプション



アイコン	説明
	デバッグ・セッションを終了または連結解除します。
	フレームをステップ実行します。 これにより、次のブレークポイント (ブレークポイントが設定された OSB プロセスの受信アクティビティなど) に移動します。ブレークポイントがない場合は、すべてのフレームをステップ実行して最初のフレームに戻ります。 [F8] を押してフレームをステップ実行することもできます。
	次の有効な位置までトレース実行します。 これは、新しいフレームである場合もあれば、同じフレームの別の位置である場合もあります。 [F7] を押してフレームをトレース実行することもできます。
	フレームをステップ・アウトします。 このオプションは、OSB の Scope アクティビティまたは Sequence アクティビティを処理するためにのみ使用されます。スコープの処理が完了すると、プロセスの次のスコープまたはアクティビティで一時停止します。[Shift] + [F7] を押すこともできます。
	ステップ操作を再開します。 [F9] を押して再開することもできます。

- f. 「ステップ実行」を使用して、次のブレークポイント (この例では「レスポンス・パイプライン」) に移動します。

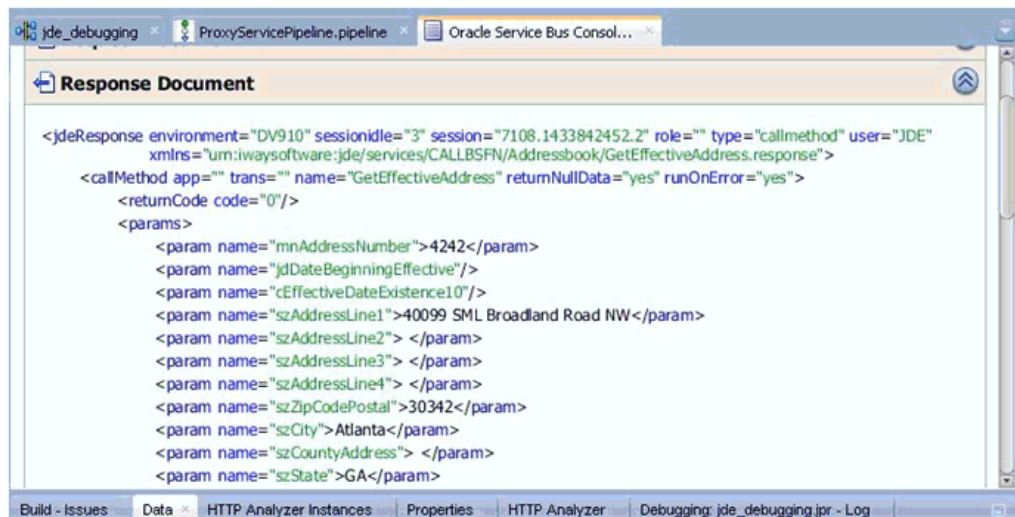
図 9-55 に示すように、「データ」タブで出力を確認できます。

図 9-55 「データ」タブの出力



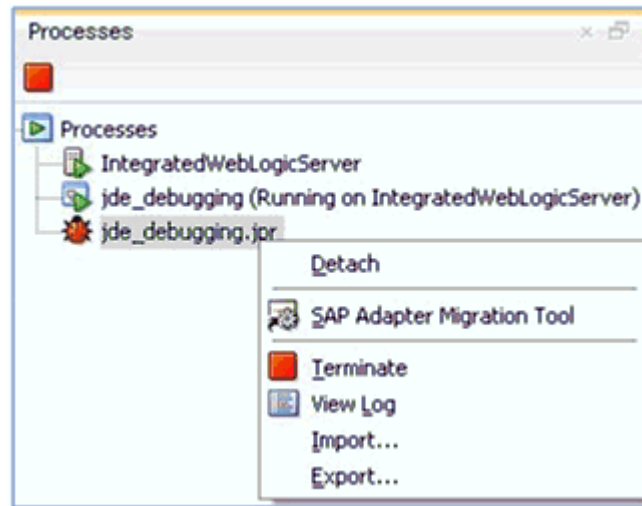
また、図 9-56 に示すように、「Oracle Service Bus コンソール」でレスポンスを確認することもできます。

図 9-56 「Oracle Service Bus コンソール」



5. デバッグ・セッションを終了または連結解除します。
 - a. 「ウィンドウ」、「プロセス」の順にクリックします。図 9-57 に示すように、「プロセス」タブでプロセスを右クリックし、「連結解除」または「終了」を選択します。

図 9-57 デバッグ・セッションの連結解除



- b. 次のいずれかのオプションを選択します。

「連結解除」: デバッグ・プロセスを終了せずに SOA デバッガを削除します。

「終了」: デバッグ・プロセスを終了します。

9.3.4.3 Oracle JDeveloper でのインバウンド OSB プロセスのデバッグ

1. インバウンド OSB プロセスを作成します。

アプリケーション名とプロジェクト名が同じであることを確認してください。

2. プロジェクトをデプロイします。

- a. プロジェクトを右クリックし、「デプロイ」を選択します。

「デプロイメント・アクション」ダイアログが表示されます。

- b. デプロイするアプリケーションを選択し、「次へ」をクリックします。

「構成のデプロイ」ダイアログが表示されます。

- c. 「次へ」をクリックします。

「アプリケーション・サーバー」ダイアログが表示されます。

- d. 「IntegratedWebLogicServer」を選択し、「終了」をクリックしてデプロイを完了します。

- e. 次のステップに進む前に、プロジェクトのデプロイが完了し、エラーや問題が発生していないことを確認してください。

3. OSB プロセスを SOA デバッガに接続します。

- a. 図 9-58 に示すように、左ペインの「アプリケーション」タブで、既存の OSB プロセスのコンポジット XML またはコンポジット・プロジェクトを右クリックし、コンテキスト・メニューから「デバッグ」を選択します。

図 9-58 デバッグする OSB プロセスを選択

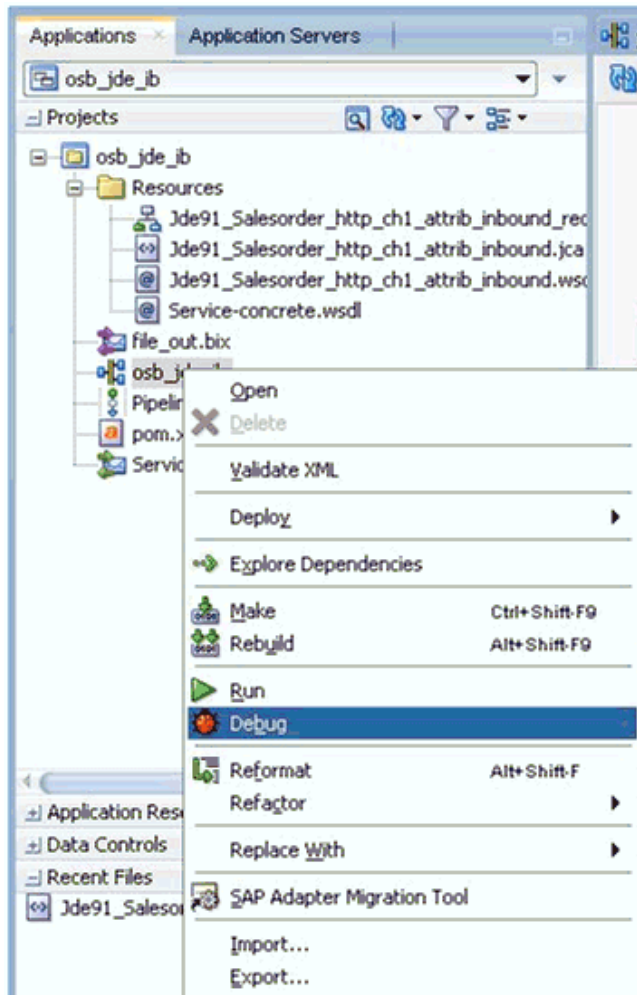
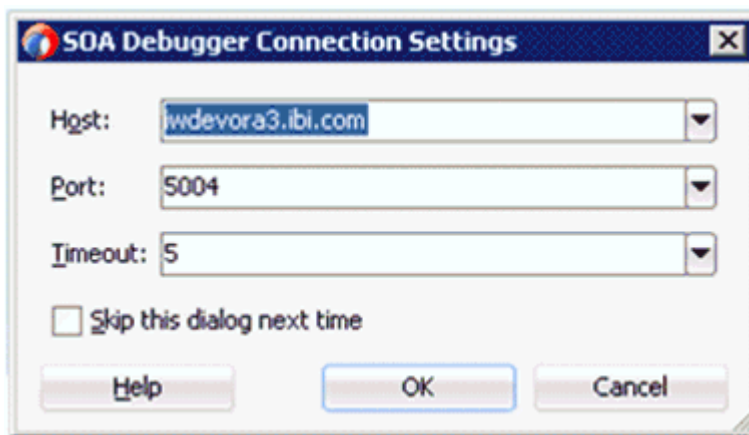


図 9-59 に示すように、「SOA デバッガ接続設定」ダイアログが表示されます。

図 9-59 「SOA デバッガ接続設定」ダイアログ



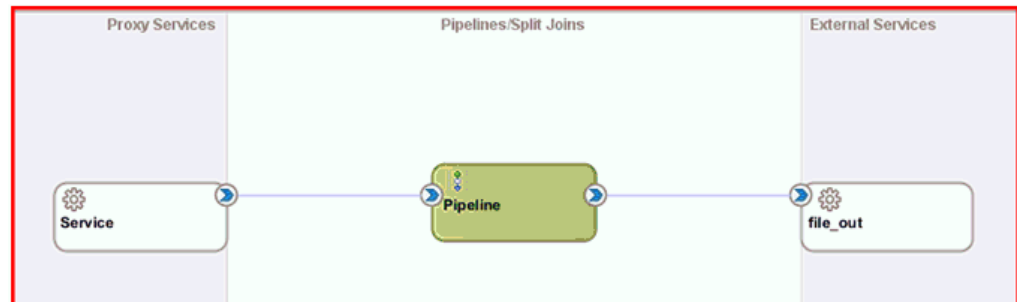
b. 「OK」をクリックします。

OSB プロセスが SOA デバッガに接続されると、デバッグ・ログに次のメッセージが表示されます。

デバッガは iwdevora3.ibi.com 5004 でリモート・プロセスに接続しようとしています。
 デバッガは iwdevora3.ibi.com 5004 でリモート・プロセスに接続しました。
 デバッガ・プロセスの仮想マシンは SOA デバッガです。

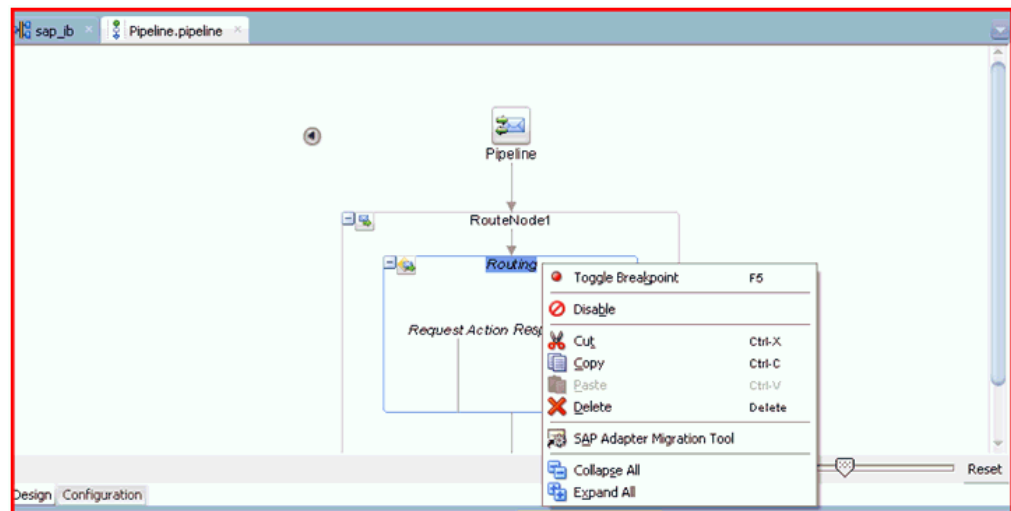
4. ブレークポイントを設定します。
 - a. 図 9-60 に示すように、パイプラインをダブルクリックしてエディタで開きます。

図 9-60 ProxyServicePipeline



- b. ブレークポイントを追加するノードが表示されるまでアクションを展開します。この例では、「ルーティング」が表示されるまで展開します。図 9-61 に示すように、「ルーティング」ノードを右クリックし、コンテキスト・メニューから「ブレークポイントの切替え」を選択します。

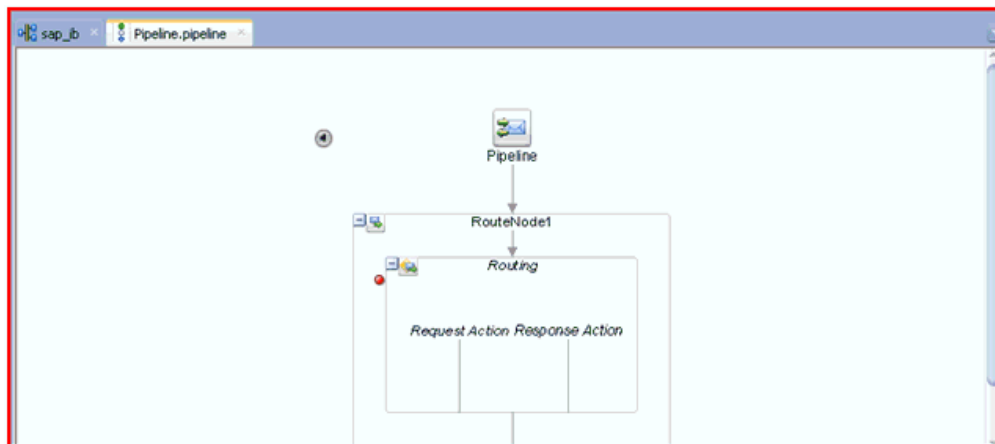
図 9-61 「ブレークポイントの切替え」



このステップを、ブレークポイントを追加する各ノードに対して繰り返します。この例では、ブレークポイントを設定するのは「ルーティング」ノードのみです。

図 9-62 に示すように、ブレークポイントが設定されたことを示す赤いアイコンがノードの横に表示されます。

図 9-62 設定されたブレークポイントの赤いアイコン



注意： ブレークポイントを無効にするには、ノードを右クリックして「ブレークポイントの無効化」を選択します。

ブレークポイントを削除するには、ノードを右クリックして再度「ブレークポイントの切替え」を選択します。

5. デバッグを開始します。
 - a. 図 9-63 に示すように、「アプリケーション・ナビゲータ」でパイプラインを右クリックし、「デバッグ」を選択します。

図 9-63 「デバッグ」を選択

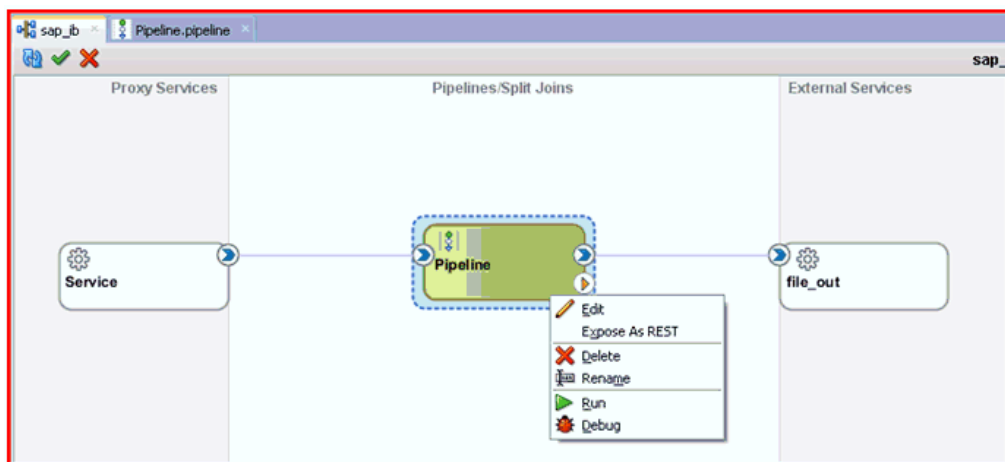


図 9-64 に示すように、プロセスが統合サーバーにデプロイされて、「構成のテスト」ペインが表示されます。

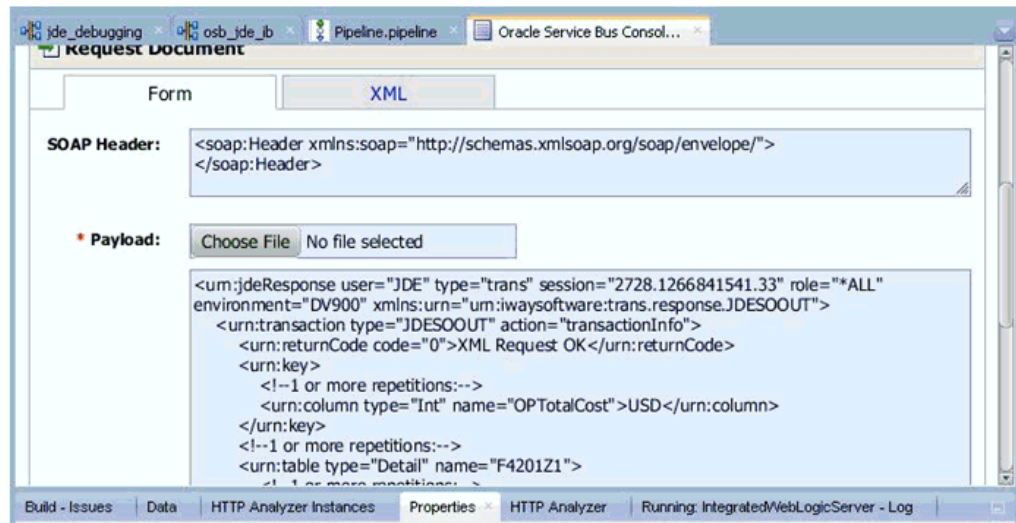
図 9-64 「構成のテスト」ペイン



注意： 現在実行されているドメインがない場合は、「デフォルト・ドメインの作成」ダイアログが表示されます。統合サーバーの接続情報を入力し、「OK」をクリックします。このプロセスには数分かかる場合があります。

- b. 図 9-65 に示すように、「構成のテスト」ペインで、「リクエスト・ドキュメント」領域にテスト・データを入力し、必要に応じて追加の入力を構成します。

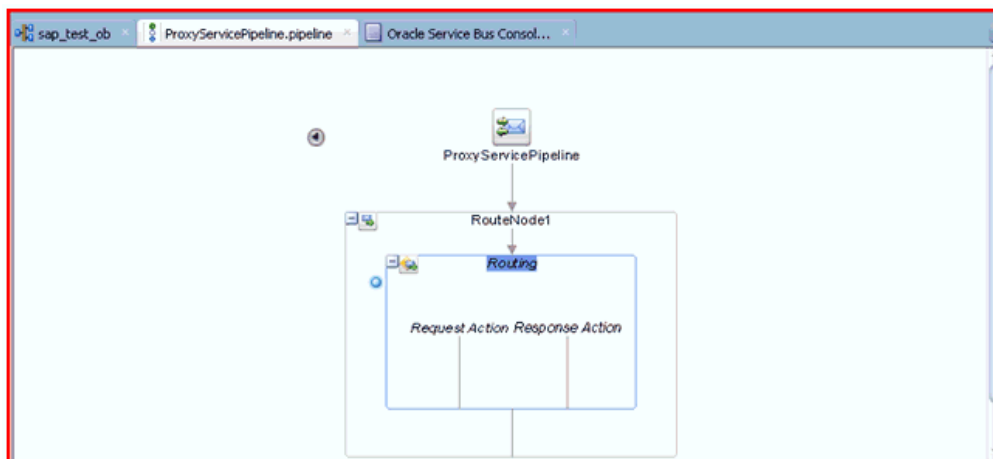
図 9-65 リクエスト・ドキュメントのテスト



- c. 「実行」をクリックします。

「構成のテスト」ペインでコマンドが実行されますが、図 9-66 に示すように、指定したブレークポイントで OSB プロセスが停止し、青く点滅します。






図 9-66 OSB プロセスがブレークポイントで停止



- d. 図 9-67 に示すように、使用可能な「ステップ」オプションを使用してデバッグ・プロセスをステップ実行します。

図 9-67 「ステップ」オプション



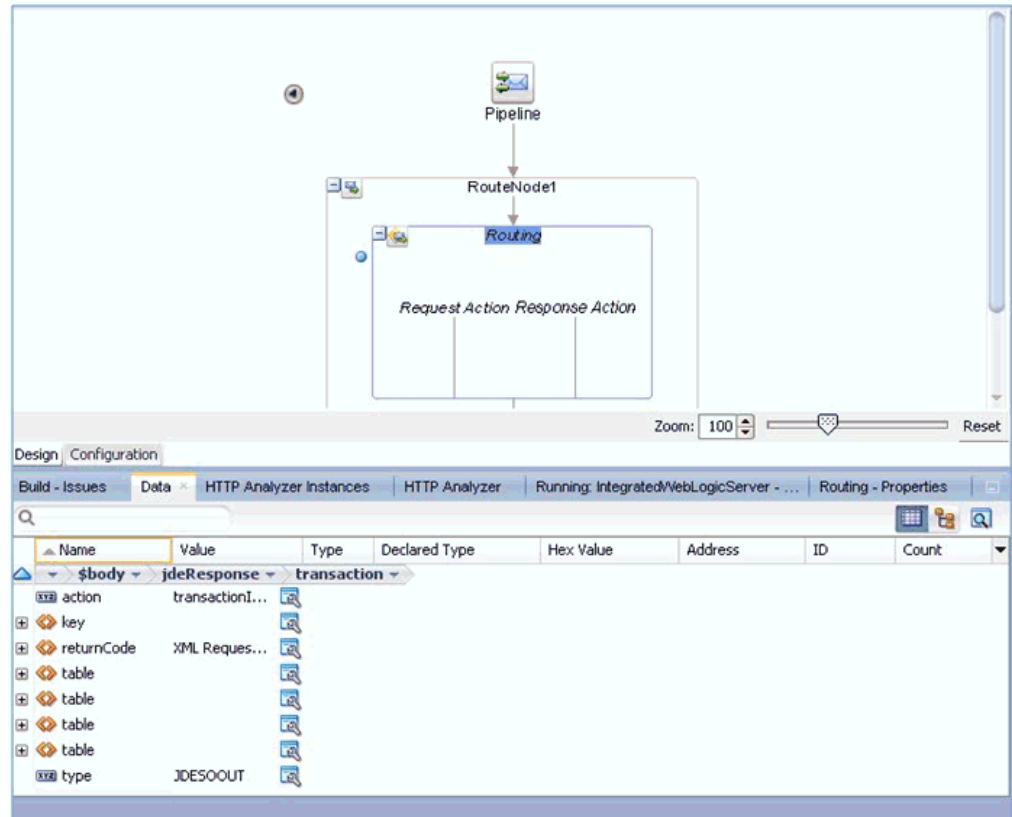
アイコン	説明
	デバッグ・セッションを終了または連結解除します。
	フレームをステップ実行します。 これにより、次のブレークポイント (ブレークポイントが設定された OSB プロセスの受信アクティビティなど) に移動します。ブレークポイントがない場合は、すべてのフレームをステップ実行して最初のフレームに戻ります。 [F8] を押してフレームをステップ実行することもできます。
	次の有効な位置までトレース実行します。 これは、新しいフレームである場合もあれば、同じフレームの別の位置である場合もあります。 [F7] を押してフレームをトレース実行することもできます。
	フレームをステップ・アウトします。 このオプションは、OSB の Scope アクティビティまたは Sequence アクティビティを処理するためにのみ使用されます。スコープの処理が完了すると、プロセスの次のスコープまたはアクティビティで一時停止します。[Shift] + [F7] を押すこともできます。
	ステップ操作を再開します。 [F9] を押して再開することもできます。

- e. 「ステップ実行」を使用して、プロセスの実行を完了します。

注意： この例にはブレークポイントが1つしかないため、「ステップ実行」を使用するとプロセスの実行が完了します。

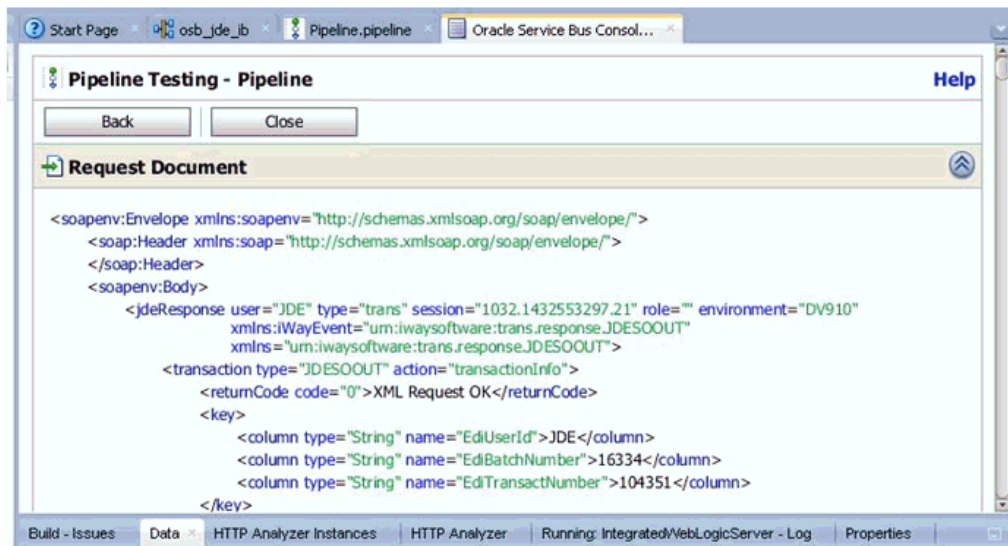
図 9-68 に示すように、「データ」タブでレスポンス・ドキュメントを確認できます。

図 9-68 「データ」タブの出力



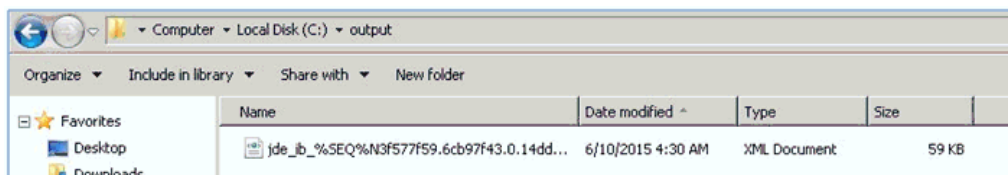
また、図 9-69 に示すように、「Oracle Service Bus コンソール」でレスポンスを確認することもできます。

図 9-69 「Oracle Service Bus コンソール」



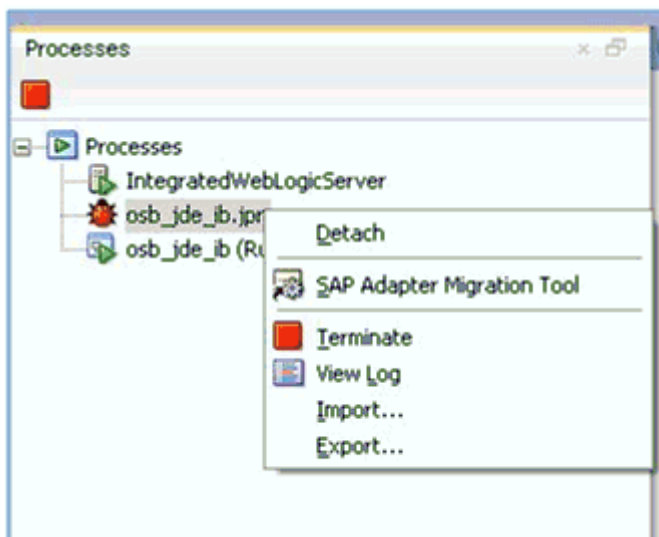
出力は、図 9-70 に示すように、構成済の出力場所にもあります。

図 9-70 構成済の出力場所



6. デバッグ・セッションを終了または連結解除します。
 - a. 「ウィンドウ」、「プロセス」の順にクリックします。図 9-71 に示すように、「プロセス」タブでプロセスを右クリックし、「連結解除」または「終了」を選択します。

図 9-71 デバッグ・セッションの連結解除



- b. 次のいずれかのオプションを選択します。

「連結解除」: デバッグ・プロセスを終了せずに SOA デバッガを削除します。

「終了」: デバッグ・プロセスを終了します。

9.4 例外フィルタ

この項では、Oracle Application Adapter for J.D. Edwards OneWorld の例外フィルタ機能を構成する方法について説明し、サンプルのテスト・シナリオを示します。

この項では、次のトピックについて説明します。

- 9.4.1 項「例外フィルタの構成」

例外フィルタは、J2CA 構成を使用するアウトバウンド・プロセスについてのみサポートされます。この機能は、BSE 構成、および J2CA 構成を使用するインバウンド・プロセスについてはサポートされていません。

例外フィルタでは、生成された例外をフィルタするために、`com.ibi.afjca.oracle.AdapterExceptionFilter` クラスを使用します。このクラスにより、例外はフィルタされ、次のカテゴリに分類されます。

- PCRetriableResourceException
- PCResourceException

次の例外は、フォルト・ポリシー・ファイルに表示されます。

- PCRetriableResourceException - リモート・フォルト。
- PCResourceException - バインディング・フォルト。

9.4.1 例外フィルタの構成

例外フィルタの構成は、次のステップおよびトピックで構成されています。

1. 9.4.1.1 項「WSDL ファイルの生成」
2. 9.4.1.2 項「例外フィルタ機能を備えた BPEL プロセスの作成」
3. 9.4.1.3 項「フォルト・ポリシー・ファイルおよびフォルト・バインディング・ファイルの作成」
4. 9.4.1.4 項「12c でのデプロイメントに関する既知の問題への対応」
5. 9.4.1.5 項「例外フィルタ機能を備えた BPEL プロセスのデプロイおよびテスト」

9.4.1.1 WSDL ファイルの生成

WSDL ファイルを生成するには、次のステップを実行します。

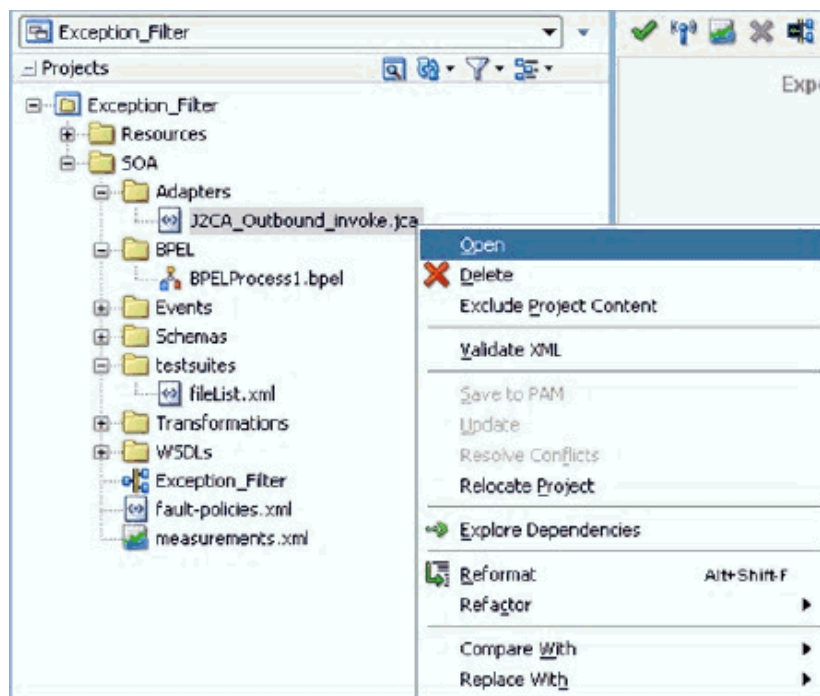
1. アプリケーション・エクスプローラを開き、J2CA 構成を作成します。
詳細は、2-3 ページの「[J2CA 用の構成の作成](#)」を参照してください。
2. PeopleSoft アダプタのターゲットを作成してから、そのターゲットに接続します。
詳細は、2-5 ページの「[J.D. Edwards OneWorld 用の接続 \(ターゲット\) の確立](#)」を参照してください。
3. 適切なオブジェクトの WSDL を生成します。
詳細は、2-11 ページの「[WSDL の生成 \(J2CA 構成のみ\)](#)」を参照してください。

9.4.1.2 例外フィルタ機能を備えた BPEL プロセスの作成

例外フィルタ機能を備えた BPEL プロセスを作成するには、次のステップを実行します。

1. JDeveloper を開き、新規の SOA アプリケーションを作成します。
詳細は、4-9 ページの 4.4.2 項「SOA 用の空のコンポジットの作成」を参照してください。
2. 新規の SOA プロジェクト (Exception_Filter など) を作成します。
3. サード・パーティ・アダプタ・サービス・コンポーネントを作成します。
詳細は、4-11 ページの 4.4.3.1 項「サード・パーティ・アダプタ・サービス・コンポーネントの構成」を参照してください。
サード・パーティ・アダプタ・サービス・コンポーネントが作成されると、WSDL ファイル (対応するスキーマおよび JCA ファイルを含む) が JDeveloper プロジェクトにインポートされます。
詳細は、4-11 ページの 4.4.3 項「BPEL アウトバウンド・プロセスの定義」を参照してください。
4. インポートされた JCA ファイルを変更します。
 - a. 図 9-72 に示すように、インポートされた JCA ファイルを右クリックし、メニューの「開く」を選択します。

図 9-72 「アプリケーション・ナビゲータ」タブ



- b. <interaction-spec> 要素に、ExceptionFilter プロパティを追加します。次に例を示します。

```
<interaction-spec className="com.ibi.afjca.cci.IWAFInteractionSpec">
  <property name="FunctionName" value="PROCESS"/><property
  name="ExceptionFilter"
  value="com.ibi.afjca.oracle.AdapterExceptionFilter"/></interaction-spec>
```

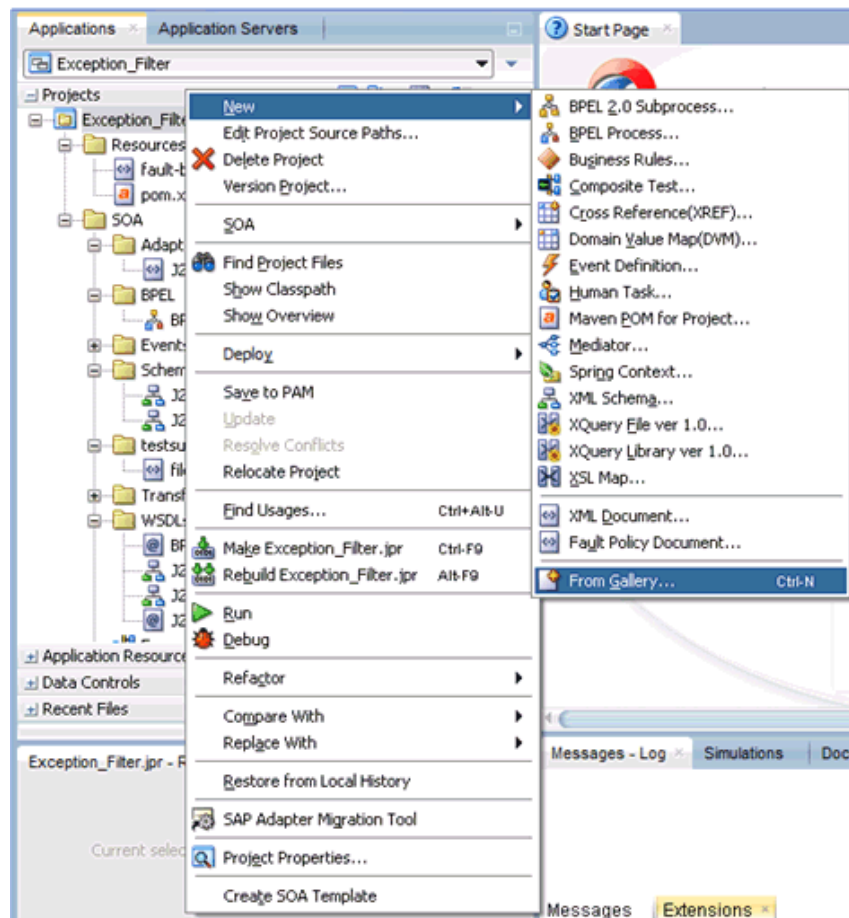

- c. 変更された JCA ファイルを保存します。
5. サード・パーティ・アダプタ・サービス・コンポーネントが作成され、JCA ファイルが変更されたら、BPEL プロセス作成の残りの部分を続行します。
- 詳細は、4-11 ページの 4.4.3 項「BPEL アウトバウンド・プロセスの定義」を参照してください。

9.4.1.3 フォルト・ポリシー・ファイルおよびフォルト・バインディング・ファイルの作成

フォルト・バインディング・ファイルを作成するには、次のステップを実行します。

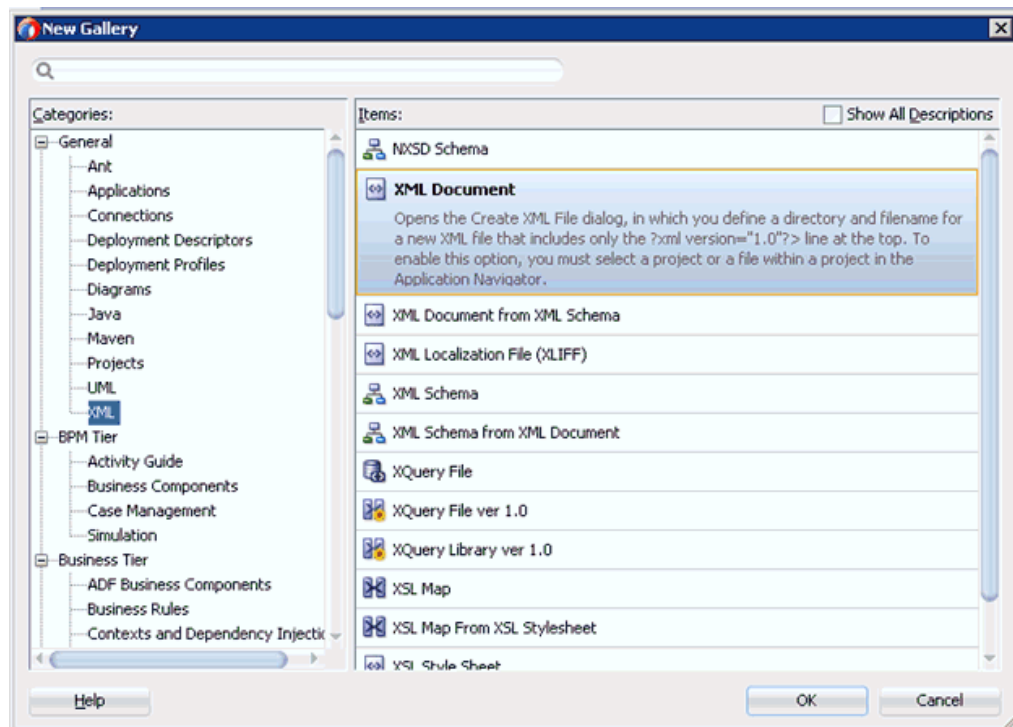
1. 図 9-73 に示すように、作成された SOA プロジェクト (Exception_Filter など) を右クリックして、「新規」を選択し、「ギャラリーから」をクリックします。

図 9-73 「アプリケーション」タブ



「新規ギャラリー」ダイアログが表示されます。図 9-74 に示すように、「一般」カテゴリの下の「XML」をクリックします。

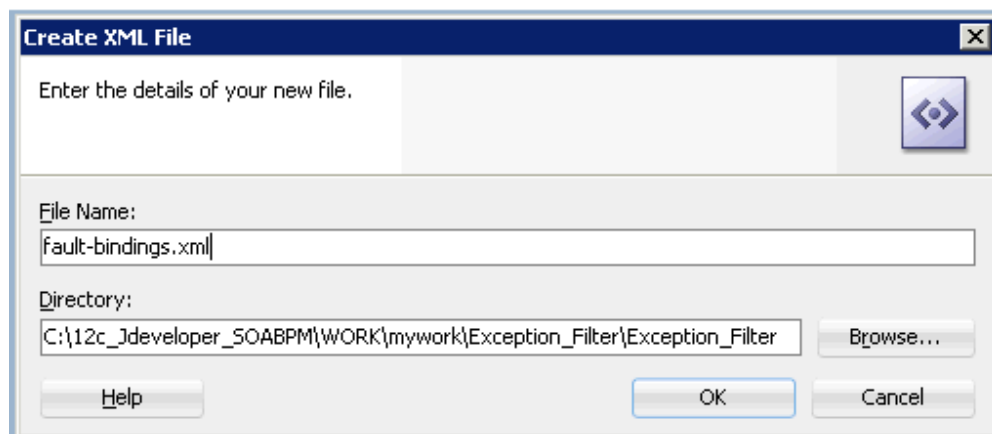
図 9-74 「新規ギャラリー」 ダイアログ



- 「項目」の下の「XML 文書」を選択し、「OK」をクリックします。

図 9-75 に示すように、XML ファイルの作成ダイアログが表示されます。

図 9-75 XML ファイルの作成ダイアログ



- 「ファイル名」フィールドに「**fault-bindings.xml**」と入力し、「OK」をクリックします。
- fault-bindings.xml** ファイルに、適切なフォルト・バインディング機能を追加します。

サンプルの **fault-bindings.xml** ファイルを表示するには、9-49 ページの「[サンプルの Fault-Bindings.xml ファイル](#)」を参照してください。

注意： <name> 要素内のパラメータは、作成された BPEL プロセスの名前です。

5. **fault-bindings.xml** ファイルを保存します。

サンプルの Fault-Bindings.xml ファイル

```
<?xml version="1.0" encoding="UTF-8" ?>
<faultPolicyBindings version="2.0.1"
xmlns="http://schemas.oracle.com/bpel/faultpolicy"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<component faultPolicy="bpelFaultHandling">
<name>BPELProcess1</name>
</component>

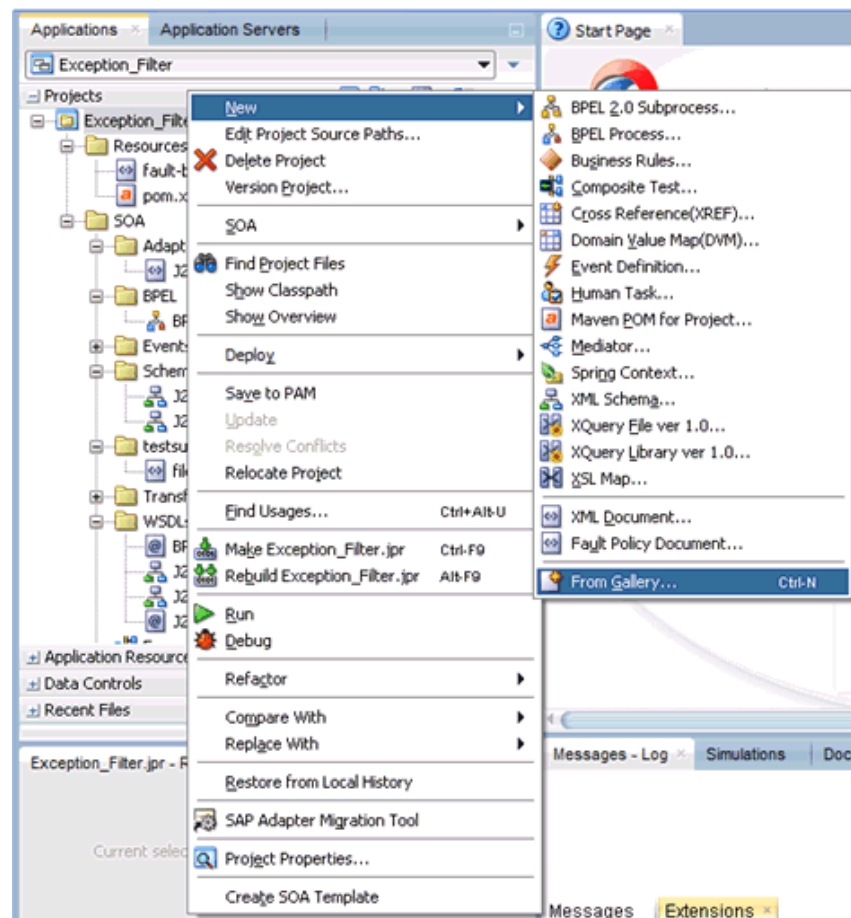
</faultPolicyBindings>
```

フォルト・ポリシー・ファイルの作成

フォルト・ポリシー・ファイルを作成するには、次のステップを実行します。

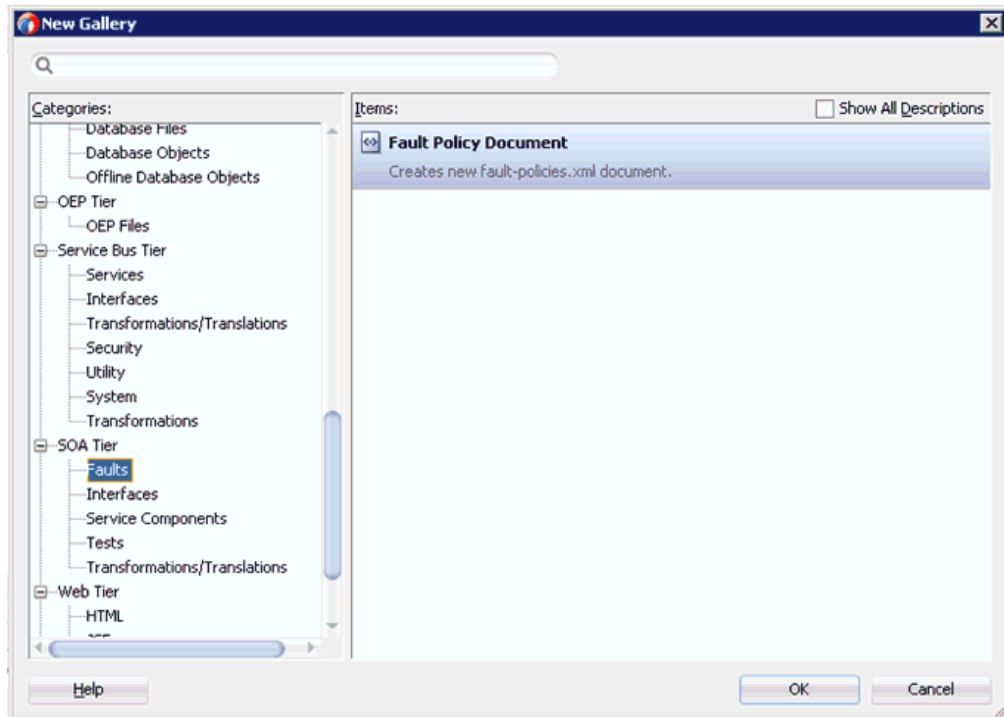
1. 図 9-76 に示すように、作成された SOA プロジェクト (Exception_Filter など) を右クリックして、「新規」を選択し、「ギャラリーから」をクリックします。

図 9-76 「アプリケーション」タブ



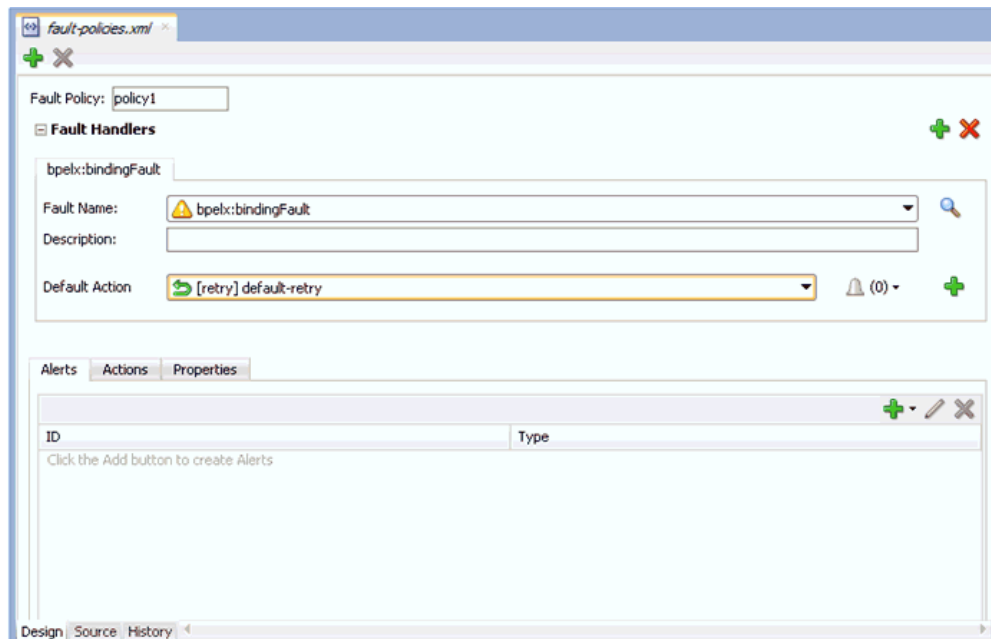
「新規ギャラリー」ダイアログが表示されます。図 9-77 に示すように、「SOA 層」カテゴリの下の「フォルト」を選択します。

図 9-77 「新規ギャラリー」 ダイアログ



- 「項目」の下の「フォルト・ポリシー・ドキュメント」を選択し、「OK」をクリックします。
- 図 9-78 に示すように、「fault-policies.xml」タブで、「フォルト名」ドロップダウン・リストから「**bpelx:bindingFault**」を選択し、「デフォルト・アクション」ドロップダウン・リストから「**[retry] default-retry**」を選択します。

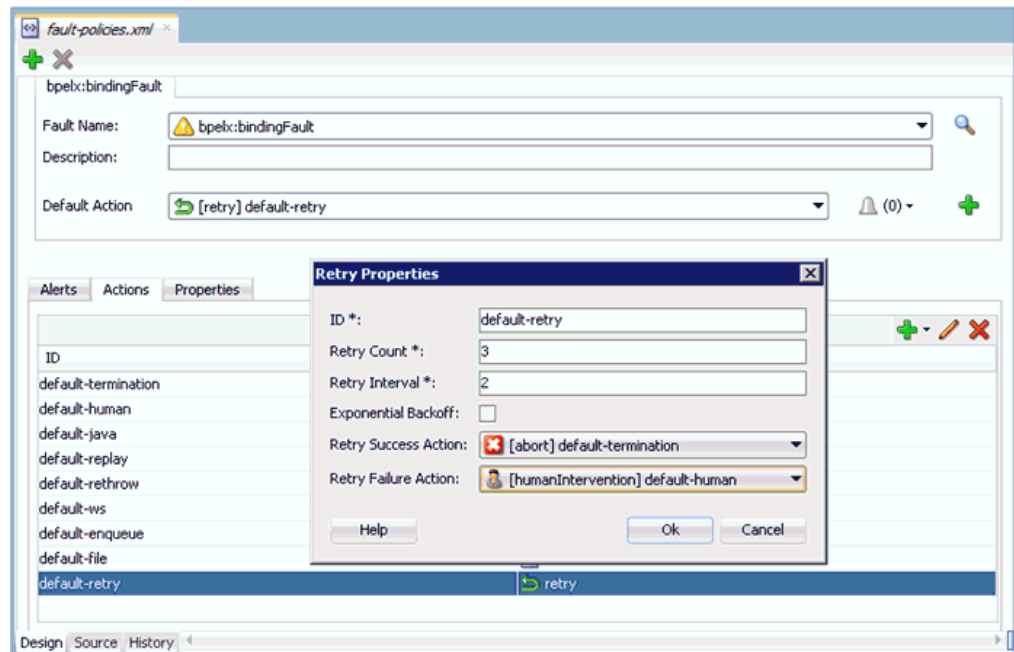
図 9-78 「Fault-policies.xml」タブ



- 「アクション」タブをクリックし、「**default-retry**」をダブルクリックします。

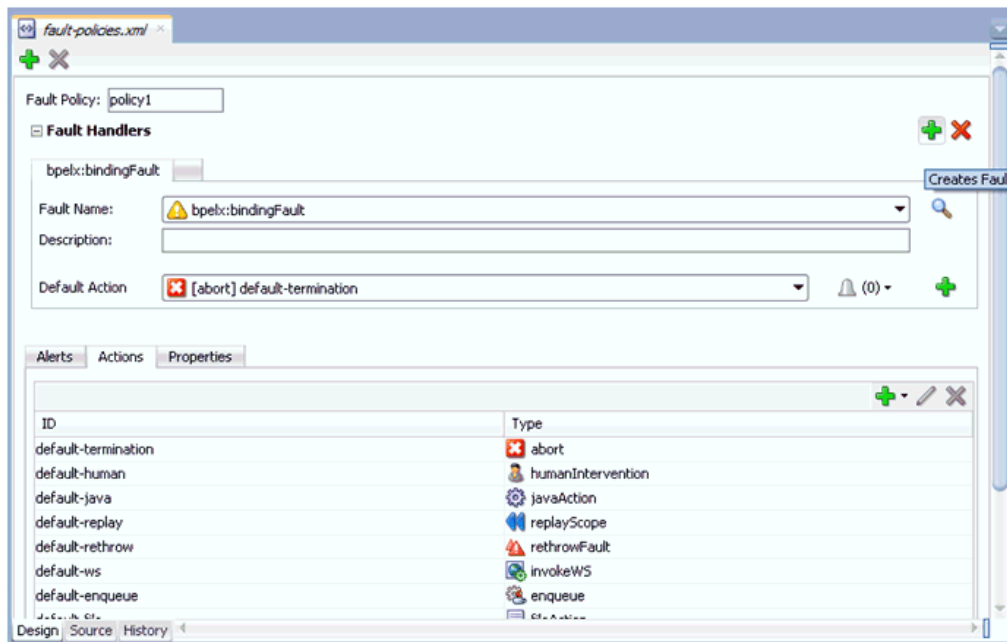
図 9-79 に示すように、「再試行プロパティ」ダイアログ・ボックスが表示されます。

図 9-79 「再試行プロパティ」ダイアログ・ボックス



5. 「再試行成功アクション」ドロップダウン・リストから「**[abort] default-termination**」を選択し、「再試行失敗アクション」ドロップダウン・リストから「**[humanIntervention] default-human**」を選択します。
6. 「OK」をクリックします。
7. 図 9-80 に示すように、別のフォルト・ハンドラを作成するには、「追加」をクリックします。

図 9-80 「Fault-policies.xml」 タブ



- 「fault-policies.xml」タブで、「フォルト名」ドロップダウン・リストから「**bpelx:remoteFault**」を選択し、「デフォルト・アクション」ドロップダウン・リストから「**[abort] default-termination**」を選択します。
- 図 9-81 に示すように、「アクション」タブで、「追加」をクリックし、「**retry**」を選択します。

図 9-81 「アクション」タブ

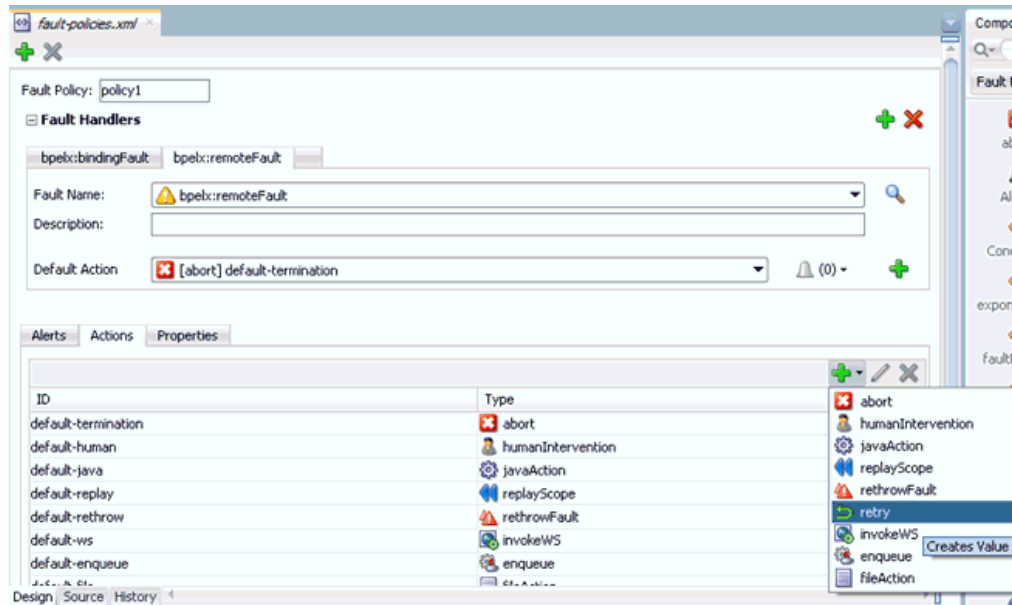
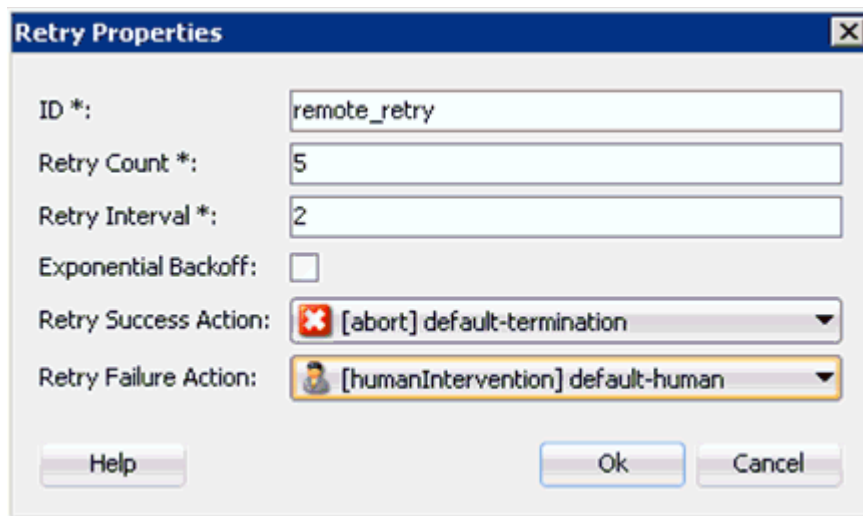


図 9-82 に示すように、「再試行プロパティ」ダイアログが表示されます。

図 9-82 「再試行プロパティ」ダイアログ・ボックス

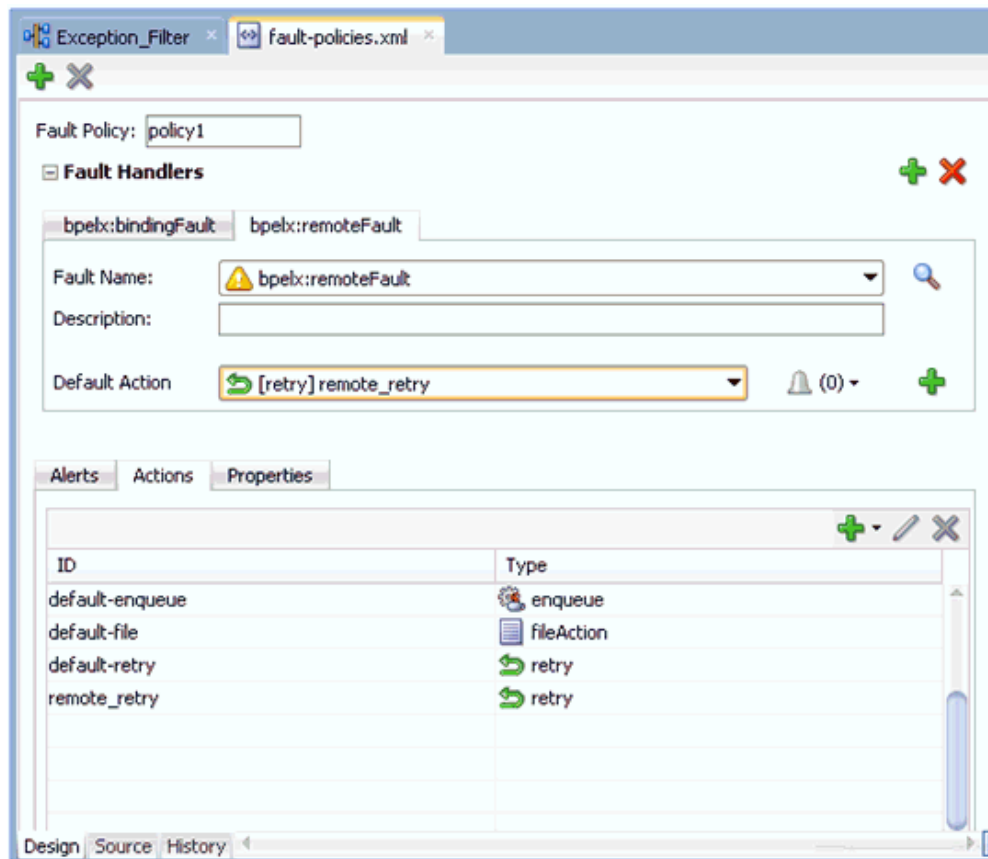


10. 「ID」、「再試行数」および「再試行間隔」の各フィールドに値を指定します。
11. 「再試行成功アクション」ドロップダウン・リストから「**[abort] default-termination**」を選択し、「再試行失敗アクション」ドロップダウン・リストから「**[humanIntervention] default-human**」を選択します。
12. 「OK」をクリックします。

作成された再試行 ID が「アクション」タブに表示されます。

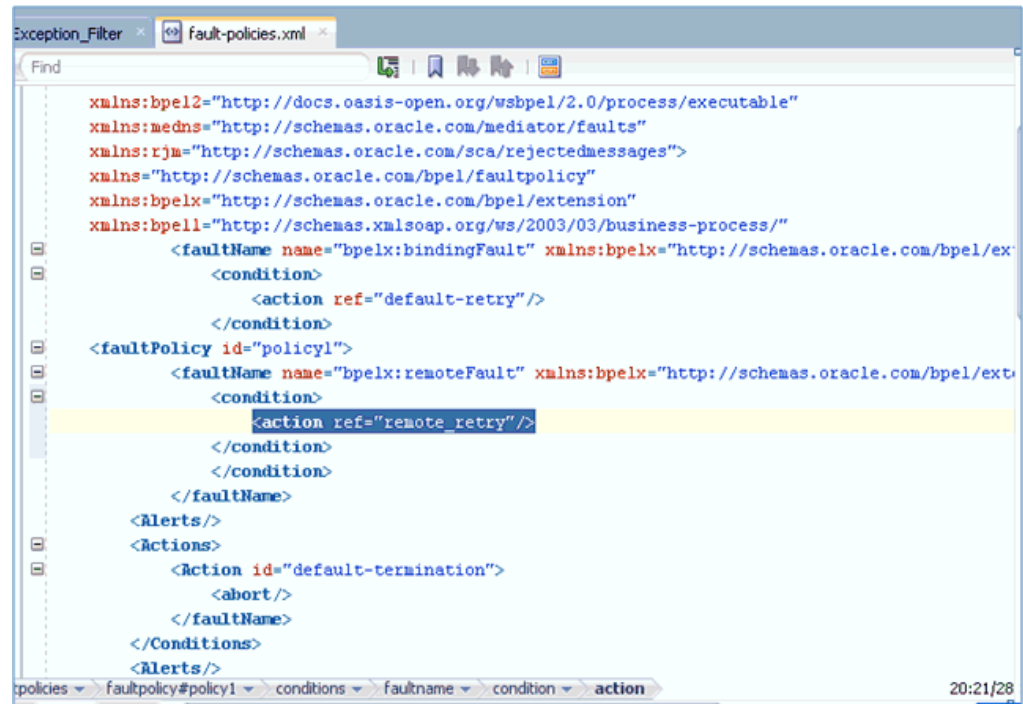
図 9-83 に示すように、「デフォルト・アクション」ドロップダウン・リストから、新規に作成された再試行 ID (remote_retry など) を選択します。

図 9-83 「Fault-policies.xml」 タブ



13. 「すべて保存」 をクリックします。
14. 図 9-84 に示すように、「ソース」 タブをクリックして、フォルト・ポリシーが正しく追加されていることを確認します。

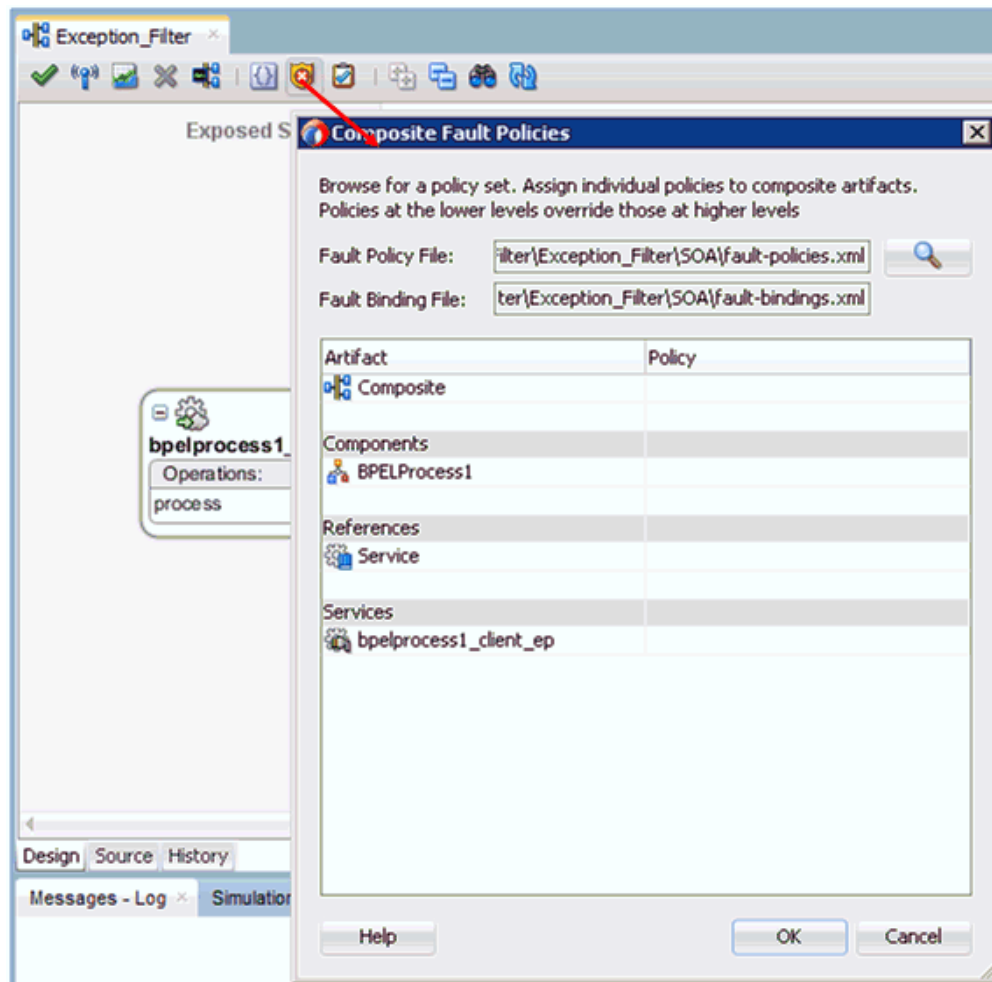
図 9-84 「ソース」タブ



15. 「Exception_Filter」プロジェクトをダブルクリックし、「コンポジット・フォルト・ポリシーの編集」をクリックします。

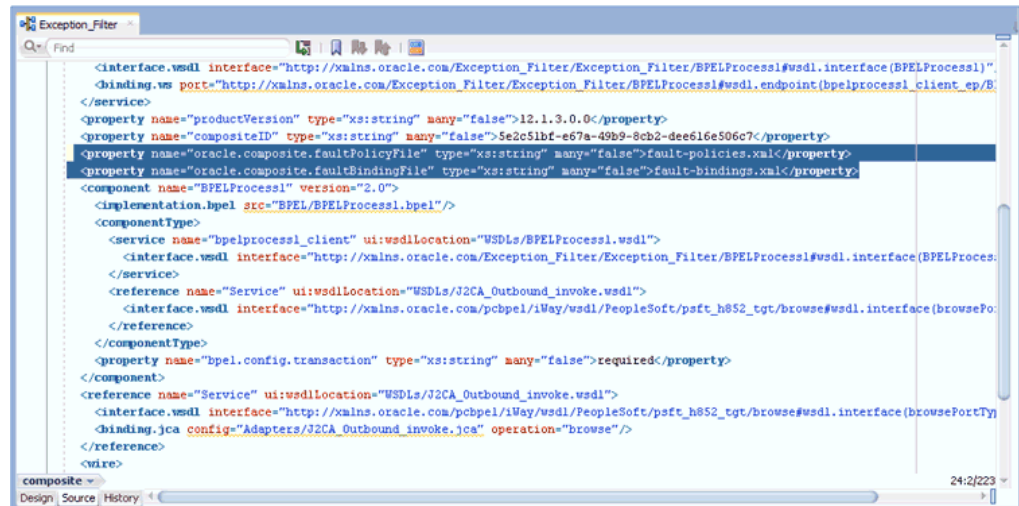
「コンポジット・フォルト・ポリシー」ウィンドウが表示されます。図 9-85 に示すように、フォルト・ポリシーおよびフォルト・バインディングが正しく選択されていることを確認します。

図 9-85 「コンポジット・フォルト・ポリシー」 ウィンドウ



16. 「すべて保存」をクリックします。
17. 図 9-86 に示すように、「ソース」タブをクリックして、*fault-bindings.xml* ファイルおよび *fault-policies.xml* ファイルが正しく追加されていることを確認します。

図 9-86 「ソース」タブ



9.4.1.4 12c でのデプロイメントに関する既知の問題への対応

12c でのデプロイメントに関する既知の問題への対応方法の詳細は、4-26 ページの 4.4.3.3 項「12c でのデプロイメントに関する既知の問題への対応」を参照してください。

9.4.1.5 例外フィルタ機能を備えた BPEL プロセスのデプロイおよびテスト

例外フィルタ機能を備えた BPEL プロセスをデプロイおよびテストするには、次のステップを実行します。

1. 作成された BPEL プロセスをデプロイします。

詳細は、4-28 ページの 4.4.4 項「BPEL アウトバウンド・プロセスのデプロイ」を参照してください。

2. (サーバーが実行中の) システムをネットワークから切断して、通信エラーをシミュレートします。
3. 有効な入力を使用して、デプロイされた BPEL プロセスを起動します。

詳細は、4-32 ページの 4.4.5 項「Oracle Enterprise Manager Console での入力 XML ドキュメントの起動」を参照してください。

4. プロセス ID を選択します。

fault-policies.xml ファイルの構成に基づいて、再試行または中断されている BPEL プロセスを監視できます。

9.5 Oracle SOA Suite (BPEL、メディエータまたは BPM) の資格証明マッピング

この項では、Oracle SOA Suite (BPEL、メディエータまたは BPM) を使用する構成において、Oracle Application Adapter for J.D. Edwards OneWorld 用に資格証明マッピング機能を構成する方法について説明します。サンプルのテスト・シナリオも含まれます。この項では、次のトピックについて説明します。

■ 9.5.1 項「資格証明マッピングの構成」

資格証明マッピングは、J2CA 構成を使用するアウトバウンド・プロセスについてのみサポートされます。この機能は、BSE 構成、および J2CA 構成を使用するインバウンド・プロセスについてはサポートされていません。

注意： J2CA コネクタは、4つのアプリケーション・アダプタ (SAP R/3、PeopleSoft、Siebel および J.D. Edwards OneWorld) すべてに共通です。資格証明マッピングが必要な場合は、特定のインスタンスでアプリケーション・アダプタが1つのみ使用されていることを確認します。たとえば、あるアダプタ・インスタンスでは、J.D. Edwards OneWorld アプリケーション・アダプタのみを使用できます。資格証明マッピングは、個々のアダプタ・レベルでは構成できません。2つのアダプタに対して資格証明マッピングを使用する必要がある場合は、両方のアダプタが、独立した2つのアダプタ・インスタンスで実行されている必要があります。

ユーザー資格証明を J2CA リソース・アダプタに渡すには、Oracle WebLogic Server ユーザー資格証明から EIS ユーザー資格証明 (J.D. Edwards OneWorld アダプタ) への資格証明マップを作成します。次に、BPEL、メディアータ、BPM のいずれかの Web サービスに資格証明ポリシーを関連付けて、Oracle WebLogic Server ユーザー資格証明を使用して Web サービスを起動します。これらの資格証明は、EIS ユーザー資格証明にマップされてから J2CA コンテナに渡され、J2CA コンテナによって EIS アダプタ (J.D. Edwards OneWorld) との接続に使用されます。

9.5.1 資格証明マッピングの構成

この項では、資格証明マッピングの構成について説明します。次のステップおよびトピックで構成されています。

1. アダプタをデプロイします。
詳細は、第3章「Oracle WebLogic Server のデプロイメントおよび統合」を参照してください。
2. Oracle WebLogic Server 資格証明を EIS 資格証明に関連付けます。
詳細は、9-57 ページの 9.5.1.1 項「Oracle WebLogic Server 資格証明の EIS 資格証明との関連付け」を参照してください。
3. WSDL ファイルを生成します。
詳細は、9-60 ページの 9.5.1.2 項「WSDL ファイルの生成」を参照してください。
4. アウトバウンド・プロセスを作成およびデプロイします。
詳細は、9-60 ページの 9.5.1.3 項「アウトバウンド・プロセスの作成およびデプロイ」を参照してください。
5. 起動して、EIS 資格証明が渡されたことを確認します。
詳細は、9-61 ページの 9.5.1.4 項「EIS 資格証明の検証」を参照してください。

9.5.1.1 Oracle WebLogic Server 資格証明の EIS 資格証明との関連付け

Oracle WebLogic Server 資格証明を EIS 資格証明に関連付けるには、次のステップを実行します。


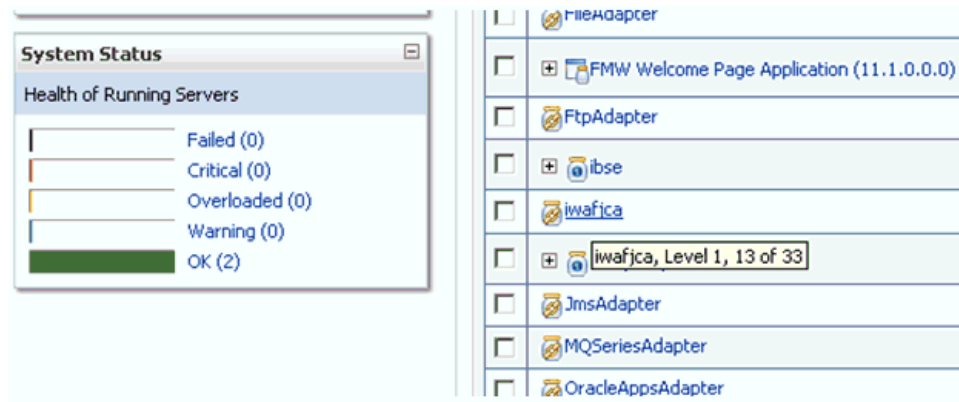
1. Oracle WebLogic Server 管理コンソールにログインします。
2.  9-87 に示すように、左ペインの「ドメイン構造」セクションで、「デプロイメント」をクリックします。

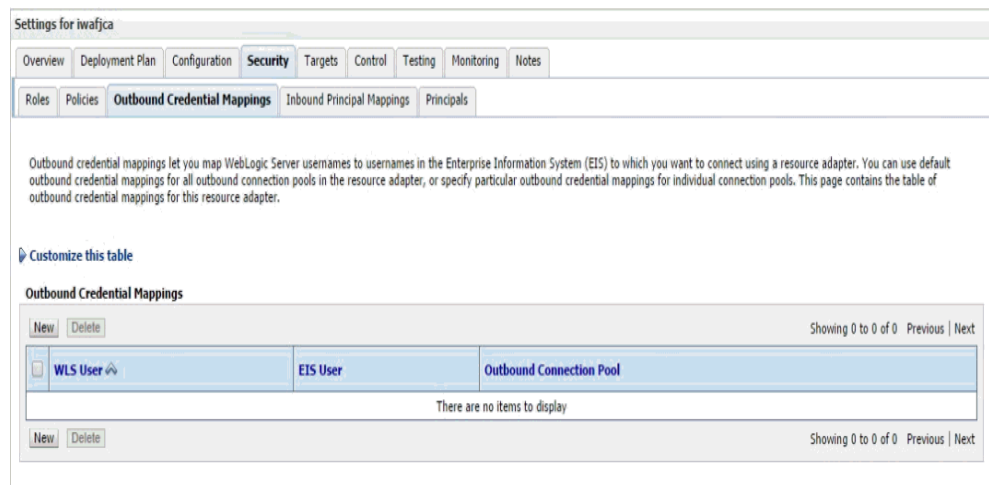
図 9-87 「デプロイメント」 ページ



3. 「iwafjca」 リソース・アダプタをクリックします。

図 9-88 に示すように、「iwafjca の設定」 ページが表示されます。

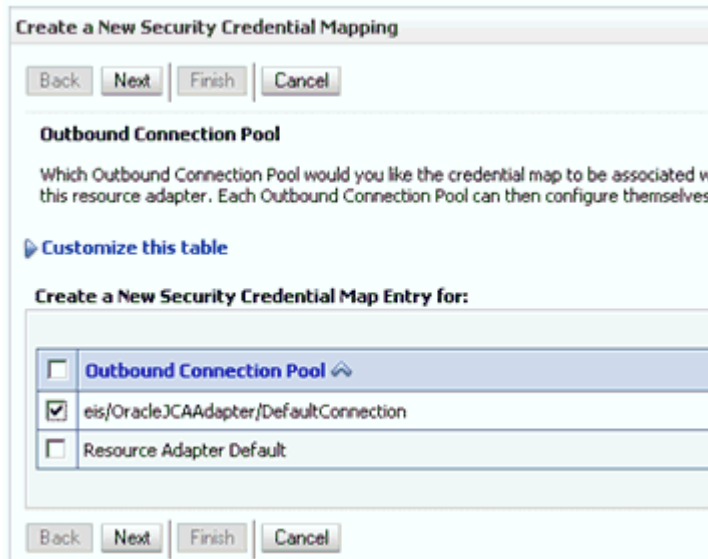
図 9-88 「iwafjca の設定」 ページ



4. 「セキュリティ」 タブの下の「アウトバウンド資格証明マッピング」 タブをクリックし、「新規」 をクリックします。

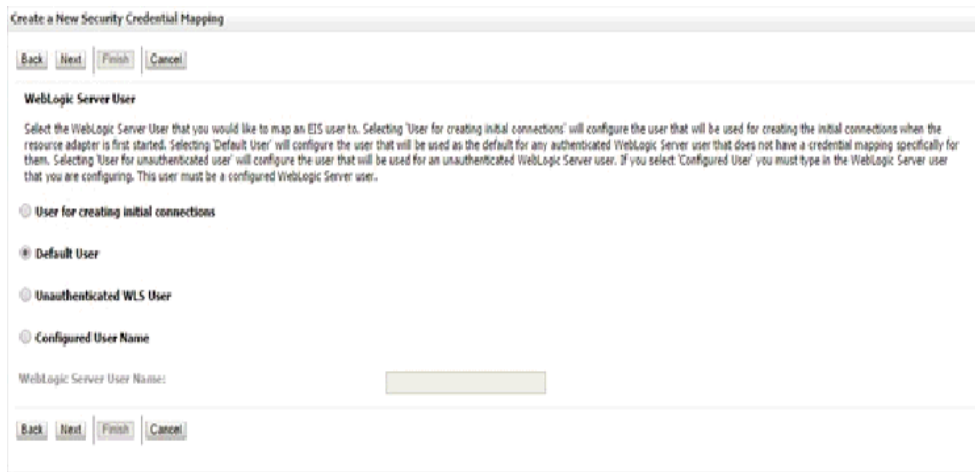
図 9-89 に示すように、「新しいセキュリティ資格証明マッピングの作成」 ページが表示されます。

図 9-89 「新しいセキュリティ資格証明マッピングの作成」 ページ



5. アウトバウンド接続プールを選択します。
次に例を示します。
eis/OracleJCAAdapter/DefaultConnection
6. 「次」をクリックします。
図 9-90 に示すように、「WebLogic Server ユーザー」ページが表示されます。

図 9-90 「WebLogic Server ユーザー」 ページ



7. 「デフォルト・ユーザー」を選択し、有効な Oracle WebLogic Server ユーザー名を入力して、「次」をクリックします。
図 9-91 に示すように、「EIS ユーザー名およびパスワード」ページが表示されます。

図 9-91 「EIS ユーザー名およびパスワード」 ページ

8. EIS のユーザー名およびパスワードを入力し、「終了」をクリックします。
これで、Oracle WebLogic Server ユーザーの資格証明は、EIS ユーザー (J.D. Edwards OneWorld) にマップされました。マッピングは、J2CA サービスの起動前に自動的に起動されます。

9.5.1.2 WSDL ファイルの生成

WSDL ファイルを生成するには、次のステップを実行します。

1. アプリケーション・エクスプローラを開き、J2CA 構成を作成します。
詳細は、2-3 ページの [2.3.2 項「J2CA 用の構成の作成」](#) を参照してください。
2. J.D. Edwards OneWorld アダプタのターゲットを作成してから、そのターゲットに接続します。
詳細は、2-5 ページの [2.4 項「J.D. Edwards OneWorld 用の接続 \(ターゲット\) の確立」](#) を参照してください。
3. 適切なオブジェクトの WSDL を生成します。
詳細は、2-11 ページの [2.6 項「WSDL の生成 \(J2CA 構成のみ\)」](#) を参照してください。

9.5.1.3 アウトバウンド・プロセスの作成およびデプロイ

この項では、アウトバウンド・プロセスの構成方法について説明します。デモンストレーションの目的で、BPEL アウトバウンド・プロセスを使用します。ただし、メディエータおよび BPM アウトバウンド・プロセスにも同じステップが適用されます。

メディエータ・アウトバウンド・プロセスの作成の詳細は、[第 5 章「Oracle SOA Suite でのメディエータ・サービス・コンポーネントとの統合」](#) を参照してください。

BPM アウトバウンド・プロセスの作成の詳細は、[第 6 章「Oracle SOA Suite での BPM サービス・コンポーネントの統合」](#) を参照してください。

BPEL アウトバウンド・プロセスを作成するには、次の項を参照してください。

- 4.4.2 項「SOA 用の空のコンポジットの作成」
- 4.4.3 項「BPEL アウトバウンド・プロセスの定義」
- 4.4.4 項「BPEL アウトバウンド・プロセスのデプロイ」

9.5.1.4 EIS 資格証明の検証

入力 XML を起動し、EIS ターゲットの資格証明が、この項の説明に従って WebLogic 管理コンソールでデフォルト・ユーザー用に構成された資格証明でオーバーライドされることを確認してください。

1. 有効な入力を含む、デプロイされた BPEL アウトバウンド・プロセスを起動します。
詳細は、4-32 ページの 4.4.5 項「Oracle Enterprise Manager Console での入力 XML ドキュメントの起動」を参照してください。
2. J2CA ログ・ファイルを確認し、ユーザー資格証明が Oracle WebLogic Server を介して EIS に渡されたことを示す暗号化パスワードを見つけます。

次に例を示します。

```
FINEST IWAFManagedConnectionFactory com.ibi.afjca.Util
getPasswordCredential(78) InLoop:
User-iwayqa:Password-ENCR(3109311731831131382333215315332323192322731773172)
FINEST IWAFManagedConnectionFactory com.ibi.afjca.Util
getPasswordCredential(90) Use the system PasswordCredential:
User-iwayqa:Password-ENCR(3109311731831131382333215315332323192322731773172)
```

9.6 Oracle Service Bus (OSB) の資格証明マッピング

この項では、Oracle Service Bus (OSB) を使用する構成において、Oracle Application Adapter for J.D. Edwards OneWorld 用に資格証明マッピング機能を構成する方法について説明します。サンプルのテスト・シナリオも含まれます。この項では、次のトピックについて説明します。

- 9.6.1 項「資格証明マッピングの構成」

資格証明マッピングは、J2CA 構成を使用するアウトバウンド・プロセスについてのみサポートされます。この機能は、BSE 構成、および J2CA 構成を使用するインバウンド・プロセスについてはサポートされていません。

注意： J2CA コネクタは、4つのアプリケーション・アダプタ (SAP R/3、PeopleSoft、Siebel および J.D. Edwards OneWorld) すべてに共通です。資格証明マッピングが必要な場合は、特定のインスタンスでアプリケーション・アダプタが1つのみ使用されていることを確認します。たとえば、あるアダプタ・インスタンスでは、J.D. Edwards OneWorld アプリケーション・アダプタのみを使用できます。資格証明マッピングは、個々のアダプタ・レベルでは構成できません。2つのアダプタに対して資格証明マッピングを使用する必要がある場合は、両方のアダプタが、独立した2つのアダプタ・インスタンスで実行されている必要があります。

ユーザー資格証明を iWay J2CA リソース・アダプタに渡すには、Oracle WebLogic Server ユーザー資格証明から EIS ユーザー資格証明 (J.D. Edwards OneWorld アダプタ) への資格証明マップを作成します。次に、資格証明ポリシーを Web サービスに関連付けて、Oracle WebLogic Server ユーザー資格証明を使用して Web サービスを起動します。これらの資格証明は、EIS ユーザー資格証明にマップされてから iWay J2CA コンテナに渡され、iWay J2CA コンテナによって EIS アダプタ (J.D. Edwards OneWorld) との接続に使用されます。

9.6.1 資格証明マッピングの構成

資格証明マッピングの構成は、次のステップおよびトピックで構成されています。

1. アダプタをデプロイします。
詳細は、第3章「Oracle WebLogic Server のデプロイメントおよび統合」を参照してください。
2. Oracle WebLogic Server 資格証明を EIS 資格証明に関連付けます。
詳細は、9.6.1.1 項「Oracle WebLogic Server 資格証明の EIS 資格証明との関連付け」を参照してください。
3. WSDL ファイルを生成します。
詳細は、9.6.1.2 項「WSDL ファイルの生成」を参照してください。
4. Oracle Service Bus (OSB) アウトバウンド・プロセスを作成します。
詳細は、9-65 ページの 9.6.1.3 項「Oracle Service Bus (OSB) アウトバウンド・プロセスの作成」を参照してください。

9.6.1.1 Oracle WebLogic Server 資格証明の EIS 資格証明との関連付け

Oracle WebLogic Server 資格証明を EIS 資格証明に関連付けるには、次のステップを実行します。

1. Oracle WebLogic Server 管理コンソールにログインします。
2. 図 9-92 に示すように、左ペインの「ドメイン構造」セクションで、「デプロイメント」をクリックします。

図 9-92 「ドメイン構造」セクション

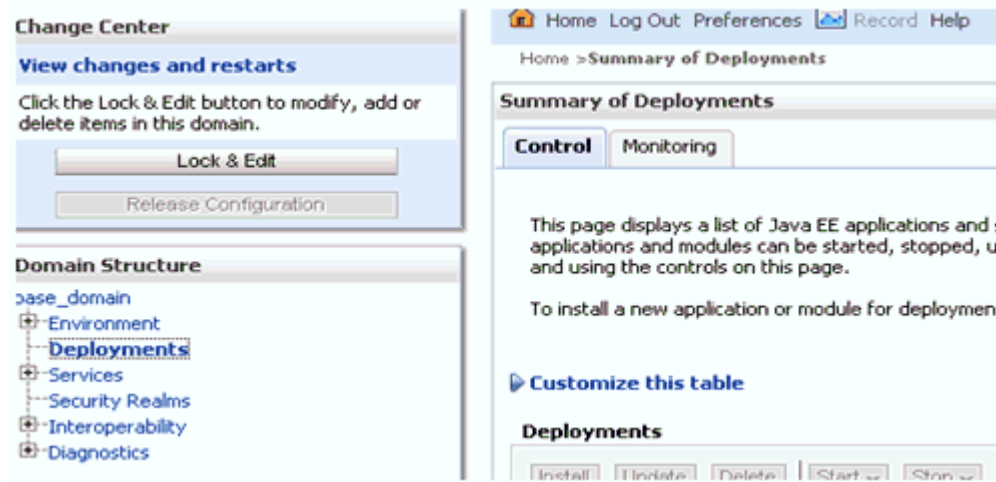
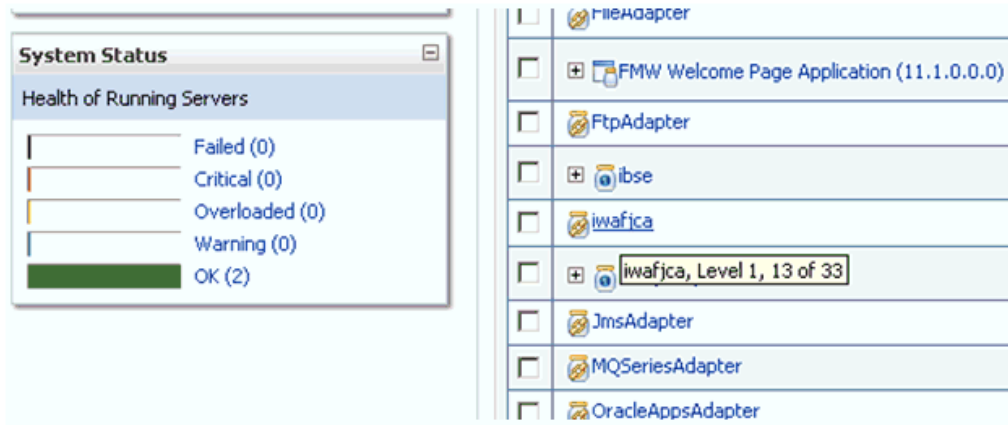


図 9-93 に示すように、「デプロイメント」ページが表示されます。

図 9-93 「デプロイメント」 ページ



3. 「iwafjca」リソース・アダプタをクリックします。
 図 9-94 に示すように、「iwafjca の設定」ページが表示されます。

図 9-94 「iwafjca の設定」 ページ



4. 「セキュリティ」タブの下の「資格証明マッピング」タブをクリックし、「新規」をクリックします。
 図 9-95 に示すように、「新しいセキュリティ資格証明マッピングの作成」ページが表示されます。

図 9-95 「新しいセキュリティ資格証明マッピングの作成」 ページ

5. アウトバウンド接続プールを選択します。
次に例を示します。
`eis/OracleJCAAdapter/DefaultConnection`
6. 「次」をクリックします。
図 9-96 に示すように、「WebLogic Server ユーザー」 ページが表示されます。

図 9-96 「WebLogic Server ユーザー」 ページ

7. 「構成済みユーザー名」を選択し、有効な Oracle WebLogic Server ユーザー名を入力して、「次」をクリックします。
図 9-97 に示すように、「EIS ユーザー名およびパスワード」 ページが表示されます。

図 9-97 「EIS ユーザー名およびパスワード」 ページ

8. EIS のユーザー名およびパスワードを入力し、「終了」をクリックします。
これで、Oracle WebLogic Server ユーザーの資格証明は、EIS ユーザー (J.D. Edwards OneWorld) にマップされました。マッピングは、J2CA サービスの起動前に自動的に起動されます。

9.6.1.2 WSDL ファイルの生成

WSDL ファイルを生成するには、次のステップを実行します。

1. Oracle Service Bus (OSB) との統合のためのアプリケーション・エクスプローラのクラス・パスを設定します。
詳細は、7-6 ページの [7.2.2 項「Oracle Service Bus 統合のためのアプリケーション・エクスプローラのクラス・パスの設定」](#) を参照してください。
2. アプリケーション・エクスプローラを開き、J2CA 構成を作成します。
詳細は、2-3 ページの [2.3.2 項「J2CA 用の構成の作成」](#) を参照してください。
3. J.D. Edwards OneWorld アダプタのターゲットを作成してから、そのターゲットに接続します。
詳細は、2-5 ページの [2.4 項「J.D. Edwards OneWorld 用の接続 \(ターゲット\) の確立」](#) を参照してください。
4. 適切なオブジェクトの WSDL を生成します。
詳細は、4-8 ページの [4.4.1 項「リクエスト/レスポンス型のサービス用の WSDL の生成」](#) を参照してください。

9.6.1.3 Oracle Service Bus (OSB) アウトバウンド・プロセスの作成

Oracle Service Bus (OSB) アウトバウンド・プロセスの作成の詳細は、8-3 ページの [8.1.2 項「OSB アウトバウンド・プロセスの定義」](#) を参照してください。

1. [図 9-98](#) に示すように、OSB プロジェクトを右クリックし、「新規」を選択し、「サービス・アカウント」をクリックして、サービス・アカウントを構成します。

図 9-98 「サービス・アカウント」オプションの選択

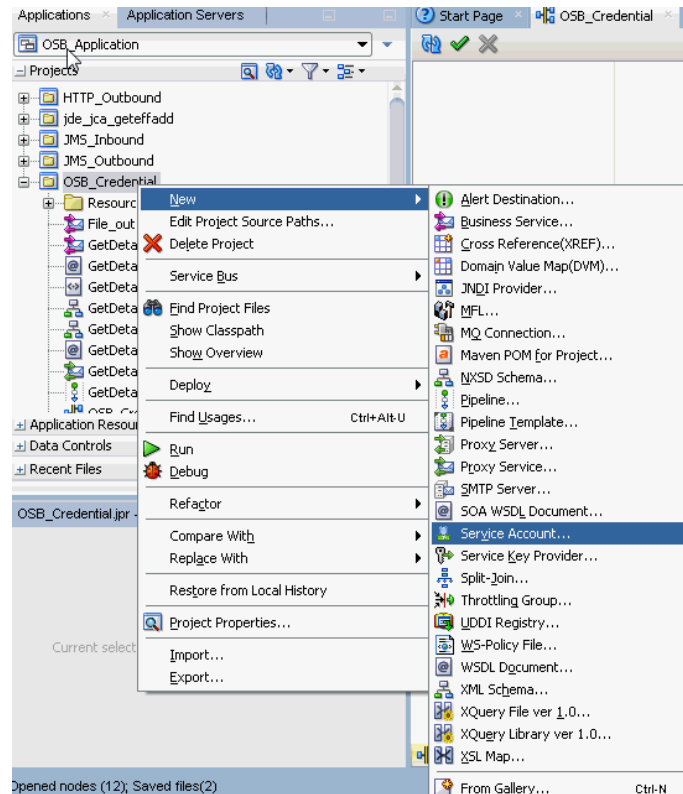
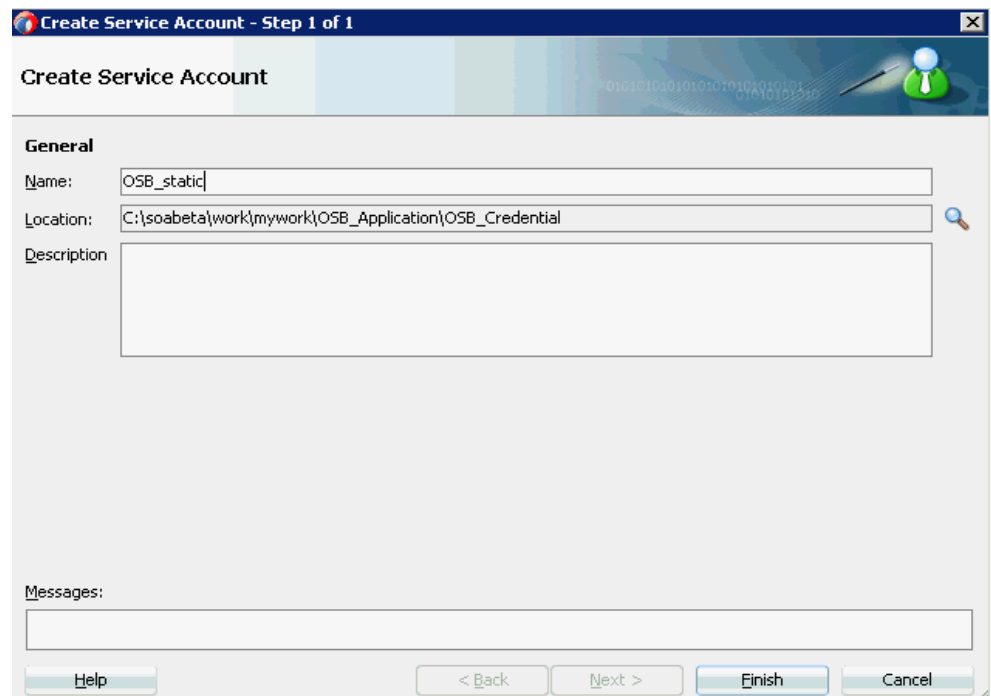


図 9-99 に示すように、「サービス・アカウントの作成」ペインが表示されます。

図 9-99 「サービス・アカウントの作成」ペイン

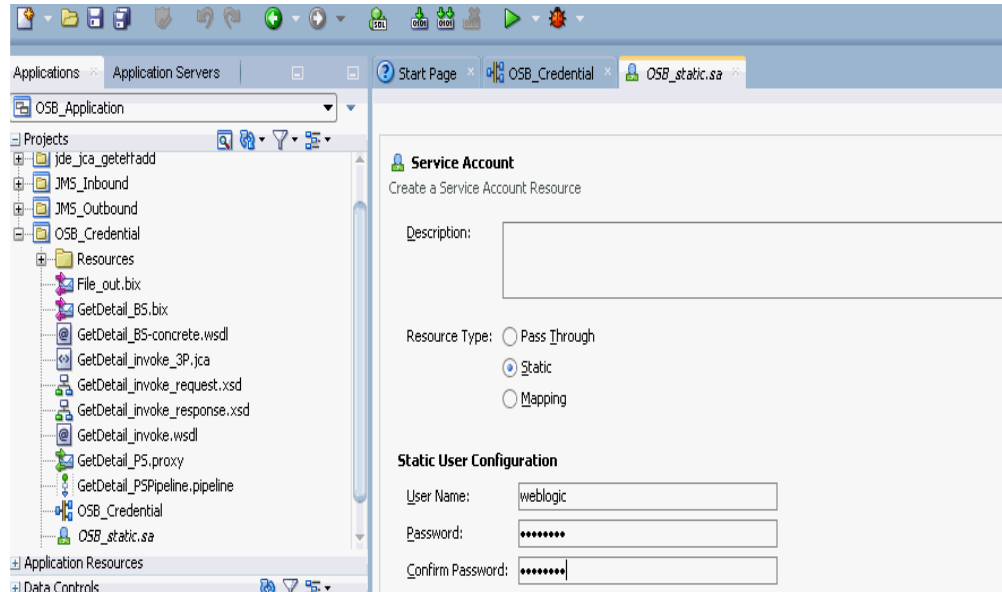


2. サービス・アカウントの名前を入力し、「終了」をクリックします。

サービス・アカウントの構成ページが表示されます。

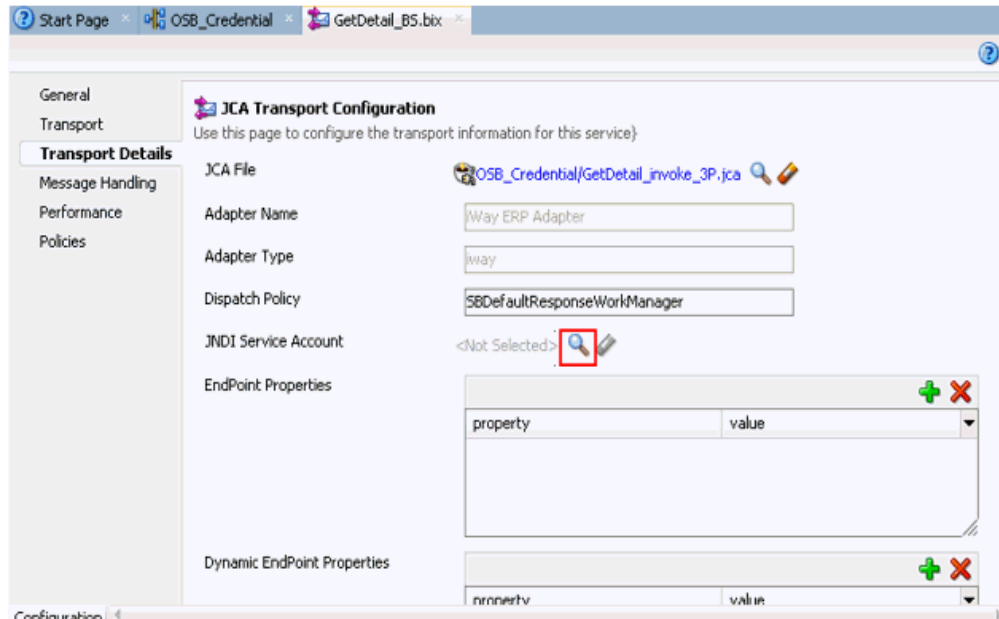
3. 「リソース・タイプ」セクションで、「静的」を選択します。
4. 図 9-100 に示すように、Oracle WebLogic Server 用の有効なユーザー名およびパスワードを指定します。

図 9-100 サービス・アカウントの構成ページ



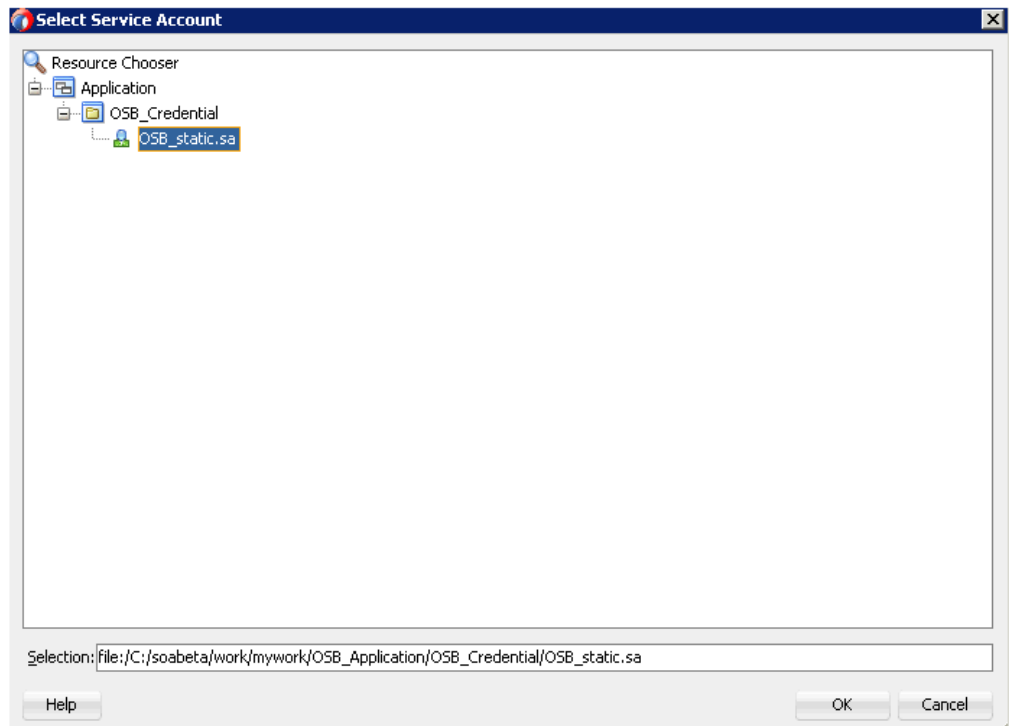
5. 構成ページを保存して閉じます。
6. コンポジット・エディタ・ウィンドウで、ステップ 3 で作成された WSDL ベースのビジネス・サービスをダブルクリックします。
WSDL ベースのビジネス・サービスの構成ページが表示されます。
7. 図 9-101 に示すように、「トランスポートの詳細」タブを選択します。

図 9-101 「トランスポートの詳細」タブ



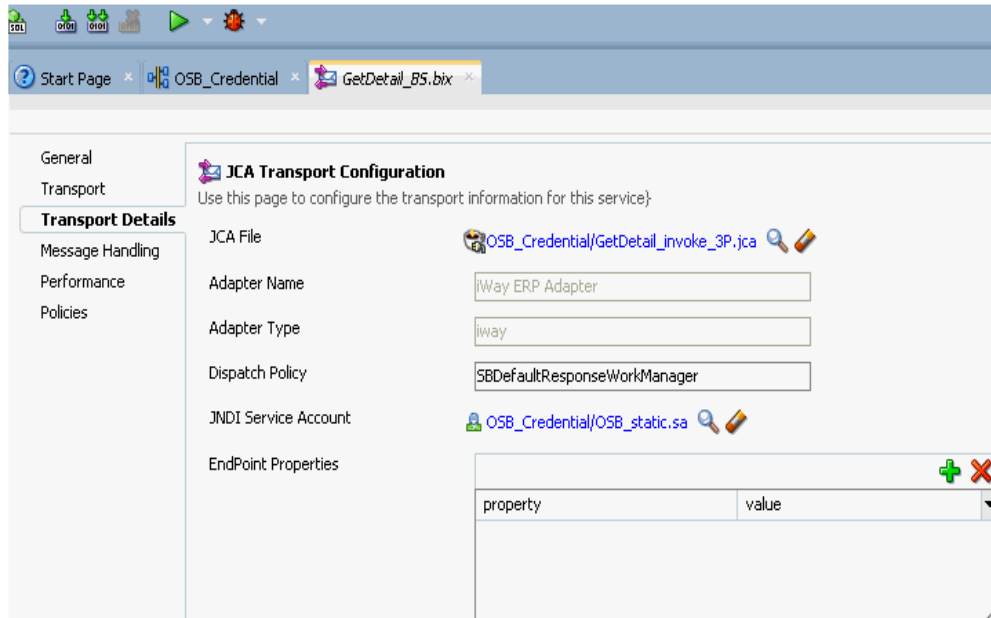
8. 「JNDI サービス・アカウント」セクションで、「参照」アイコンをクリックします。
「サービス・アカウントの選択」ウィンドウが表示されます。
9. 図 9-102 に示すように、作成されたサービス・アカウントを選択し、「OK」をクリックします。

図 9-102 サービス・アカウントの選択



10. 図 9-103 に示すように、構成ページを保存して閉じます。

図 9-103 ビジネス・サービスの構成ページ



11. OSB プロセスをデプロイします。
- 詳細は、8-16 ページの 8.1.3 項「OSB アウトバウンド・プロセスのデプロイ」を参照してください。
12. プロセスが正常にデプロイされた後、有効な入力 XML ファイルをコピーして構成済の入力フォルダに貼り付け、構成済の出力場所で出力が受信されたことを確認します。
13. J2CA ログ・ファイルを確認し、ユーザー資格証明が Oracle WebLogic Server を介して EIS に渡されたことを示す暗号化パスワードを見つけます。

次に例を示します。

```
FINEST IWAFManagedConnectionFactory com.ibi.afjca.Util
getPasswordCredential(78) InLoop:
User-iwayqa:Password-ENCR(3189319731831132182333215323332323192322731773252)
FINEST IWAFManagedConnectionFactory com.ibi.afjca.Util
getPasswordCredential(90) Use the system PasswordCredential:
User-iwayqa:Password-ENCR(3109313331831131702333215320132323192322731773236)
```

トラブルシューティングおよび エラー・メッセージ

この章では、J.D. Edwards OneWorld に接続する場合の制約事項および回避策について説明します。次のトピックが含まれています。

- 10.1 項「トラブルシューティング」
- 10.2 項「BSE エラー・メッセージ」

この章に記載しているアダプタ固有のエラーは、Oracle Adapter J2CA または Oracle Adapter Business Services Engine(BSE) のいずれかの構成でアダプタを使用した場合に発生する可能性があります。

10.1 トラブルシューティング

このトピックでは、J.D. Edwards OneWorld に関するトラブルシューティング情報について説明します。次のトピックで構成されています。

- 10.1.1 項「アプリケーション・エクスプローラ」
- 10.1.2 項「J.D. Edwards One World」
- 10.1.3 項「Oracle Adapter J2CA」

トラブルシューティングに関連する可能性があるログ・ファイル情報は、アダプタのインストール先に基づく次の場所にあります。

- Oracle Adapter J2CA トレース情報は次のディレクトリにあります。

```
<ADAPTER_HOME>\config\configuration_name\log
```

- BSE トレース情報は次のディレクトリにあります。

```
<ORACLE_HOME>\user_projects\domains\base_domain\servers\soa_  
server1\stage\ibse\ibse.war\ibselogs
```

- アプリケーション・エクスプローラのログ・ファイルは次のディレクトリにあります。

```
<ADAPTER_HOME>\tools\iwaeb\bin
```

10.1.1 アプリケーション・エクスプローラ

このトピックでは、アプリケーション・エクスプローラを使用する場合に発生する可能性がある様々なエラー・タイプについて説明します。

エラー	ソリューション
<p>アプリケーション・エクスプローラから Oracle Application Adapter for J.D. Edwards OneWorld に接続できません。</p> <p>アダプタのアクティブ化で問題が発生しました。(J.D. Edwards OneWorld への接続に失敗しました。システムの可用性およびコンフィギュレーション・パラメータを確認してください : ...) 詳細はログを確認してください。</p>	<p>次の点を確認します。</p> <ul style="list-style-type: none"> ■ J.D. Edwards OneWorld が実行されていること。 ■ J.D. Edwards OneWorld のユーザー ID およびパスワードが適切であること。 ■ ポート番号が適切です。
<p>次のエラー・メッセージが表示されます。</p> <pre>java.lang.IllegalStateException: java.lang.Exception: J.D. Edwards OneWorld システムへのログオン・エラー</pre>	<p>J.D. Edwards OneWorld に無効な接続情報が指定されているか、または誤った JAR ファイルが lib ディレクトリに存在します。</p>
<p>J.D. Edwards OneWorld がアプリケーション・エクスプローラの「アダプタ」ノード・リストに表示されません。</p>	<p>J.D. Edwards OneWorld JAR ファイルが lib ディレクトリに追加されていることを確認してください。</p>
<p>実行時にログオン失敗エラーが発生します。</p>	<p>J.D. Edwards OneWorld システムへの接続用のパスワードが、ターゲットの作成時、またはアプリケーション・エクスプローラの「編集」オプションで指定されていない場合、J.D. Edwards OneWorld には接続できません。接続パスワードは repository.xml には保存されません。アプリケーション・エクスプローラの「編集」オプションを使用してパスワードを更新し、アプリケーション・サーバーを再起動します。</p>
<p>ae.bat (iaexplorer.exe ではない) をアクティブ化してアプリケーション・エクスプローラを起動すると次の例外が発生します。</p> <pre>java.lang.ClassNotFoundException: org.bouncycastle.jce.provider.BouncyCastleProvider</pre>	<p>これは無害な例外です。アダプタの機能には影響しません。次の URL から BouncyCastle ファイルをダウンロードします。</p> <pre>ftp://ftp.bouncycastle.org/pub</pre>

エラー	ソリューション
<p>Solaris 環境でアプリケーション・エクスプローラを起動できません。次の例外がコンソールに表示されます。</p> <pre>javax.resource.ResourceException: IWFManagedConnectionFactory: License violation at com.ibi.afjca.spi.IWFManagedConnectionFactory.createConnectionFactory(IWFManagedConnectionFactory.java:98) at com.iwaysoftware.iwae.common.JCATransport.getConnectionFactory(JCATransport.java:133) at com.iwaysoftware.iwae.common.JCATransport.initJCA(JCATransport.java:69) at com.iwaysoftware.iwae.common.JCATransport.<init>(JCATransport.java:62) at com.iwaysoftware.iwae.common.AdapterClient.<init>(AdapterClient.java:85) at com.ibi.bse.ConfigWorker.run(ConfigWorker.java:41) at java.lang.Thread.run(Thread.java:534)</pre> <p>接続ファクトリを作成できませんでした。</p>	<p>JAVACMD がユーザーのシステムで設定されていません。アプリケーション・エクスプローラを起動する前に、次のように JAVACMD をエクスポートします。</p> <p>JAVACMD=/<jdk_home>/bin/java。ここで、<jdk_home> は、システム上で JDK がインストールされているディレクトリです。</p>

10.1.2 J.D. Edwards One World

エラー	原因	ソリューション
アクション・コードが無効です。	受注オーダー・リクエストでアクション・コードが「H」と表示されている場合、アクション・コードは無効です。	次の文字を使用します。 <ul style="list-style-type: none"> ■ 問合せには「I」。 ■ 変更には「C」。 ■ 削除には「D」。 ■ 新規レコードの追加には「A」。
アドレス番号が無効です。	アドレス番号が住所録マスター・ファイル (F0101) に存在しません。	住所録の改訂プログラム (PO1051) を使用し、アドレス番号を入力します。入力した番号が正しいことを確認してください。
レコードが無効です。	ADD 関数の場合、処理レコードはすでに存在しています。また、INQUIRY 関数、CHANGE 関数または DELETE 関数の場合、処理レコードは存在しません。	以前追加したレコードに対して問合せ、変更または削除を試行している場合は、本番ライブラリにデータベース問題が生じている可能性があります。データ処理の部署に連絡してください。
事業所品目レコードが存在しません。	指定した事業所にはこの品目の事業所品目レコード (F4102) が存在しません。	事業所を訂正するか、事業所品目情報 (P41026) にこの品目の事業所品目レコードを入力してください。
&1 が有効な値と一致しません。	&1 は、データ・ディクショナリでこのフィールドに対して指定されている有効な値と一致しません。	有効な値を入力してください。

エラー	原因	ソリューション
日付が範囲外です。	最終サービス日付および検査日付が、サービス契約の有効期間内である必要があります。	日付を、サービス契約の有効期間の開始日以降から終了日以前の間に変更してください。
Jde.net タイムアウト例外	ネット・タイムアウトが無効な値に設定されています。	jde.ini の [NETWORK QUEUE SETTINGS] で、ネット・タイムアウトが 180 に設定されていることを確認してください。次に例を示します。 JDENETTimeout=180
EnterpriseOne バージョン 8.10 に接続できません。	必要なライブラリ・ファイルが存在しません。	バージョン B7333 には、Kernel.jar および Connector.jar が必要です。 EnterpriseOne バージョン 8.10 には、Kernel.jar および Connector.jar に加えて、jdeutil.jar および log4j.jar が必要です。

10.1.3 Oracle Adapter J2CA

エラー	ソリューション
Oracle Adapter J2CA 構成への接続を試行すると、アプリケーション・エクスプローラに次のメッセージが表示されます。 JCA を初期化できませんでした	次の例に示すように、右ペインの「詳細」タブで、「ホーム」フィールドで正しいディレクトリが指定されていることを確認します。 <ADAPTER_HOME>\tools\iwaeb\bin\..\..\..\

10.2 BSE エラー・メッセージ

この項では、BSE を使用して Web サービスを処理する場合に発生する可能性がある様々なエラー・タイプについて説明します。

この項では、次のトピックについて説明します。

- 10.2.1 項「BSE での一般的なエラー処理」
- 10.2.2 項「アダプタ特定のエラー処理」

10.2.1 BSE での一般的なエラー処理

BSE は、アダプタ・フレームワークへの SOAP ゲートウェイと一部のアダプタのエンジンの両方の役割を果たします。デザインタイムとランタイムの両方で、アダプタを使用する Web サービスが実行されるときに様々な条件により BSE でエラーが発生することがあります。これらの条件とその結果発生するエラーは、特定のアダプタにかかわらず同様に公開されるものと、使用されるアダプタによって異なった形態で公開されるものがあります。このトピックでは、アダプタ特定のもので、いくつかの一般的なエラー条件の場合に発生するエラーについて説明します。通常、BSE 内の SOAP ゲートウェイ (エージェント) は、SOAP リクエスト・メッセージを Web サービスに必要なアダプタに渡します。エラーが発生した場合、公開方法は、アダプタと、アダプタで使用される API またはインタフェースによって異なります。いくつかのシナリオで、SOAP ゲートウェイで SOAP フォルトが生成されます。一般に、BSE 内の SOAP エージェントが無効な SOAP リクエストを受信するたびに、SOAP フォルト要素が SOAP レスポンスで生成されます。SOAP フォルト要素にはフォルト・ストリングおよびフォルト・コード要素が含まれます。フォルト・コードには、SOAP エージェント・エラーの記述が含まれます。次の SOAP レスポンス・ドキュメントは、BSE が無効な SOAP リクエストを受信したときに生成されます。

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <faultcode>SOAP-ENV:Client</faultcode>
      <faultstring>パラメータ・ノードがありません</faultstring>
    </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

この例では、BSE は、この Web サービスに必須の SOAP リクエスト・メッセージで要素を受信しませんでした。

10.2.2 アダプタ特定のエラー処理

この項では、次のトピックについて説明します。

- 10.2.2.1 項「無効な SOAP リクエスト」
- 10.2.2.2 項「Oracle WebLogic Server Adapter リクエストからの空の結果」
- 10.2.2.3 項「エラー・ログイン」
- 10.2.2.4 項「Oracle WebLogic Server Adapter リクエストからの空の結果」
- 10.2.2.4 項「Oracle WebLogic Server Adapter リクエストからの空の結果」

アダプタでランタイム中に例外が発生すると、BSE 内の SOAP エージェントは生成された SOAP レスポンス内に SOAP フォルト要素を含めます。SOAP フォルト要素には、フォルト・コード要素およびフォルト・ストリング要素が含まれます。フォルト・ストリングには、アダプタのターゲット・システムからのネイティブ・エラー記述が含まれます。アダプタはターゲット・システムのインタフェースおよび API を使用するの、例外が発生するかどうかはターゲット・システムのインタフェースまたは API によるエラー条件の処理方法によって決定されます。BSE 内の SOAP エージェントによって SOAP リクエスト・メッセージがアダプタに渡される時、リクエストがそのサービスの WSDL に基づいて無効な場合、アダプタは SOAP フォルトを起こす例外が発生することがあります。

アダプタに発生するエラー条件をすべて予測するのはほとんど不可能ですが、次では、アダプタにおける一般的なエラー条件の処理方法と、Web サービスのコンシューマ・アプリケーションへの公開方法について説明します。

10.2.2.1 無効な SOAP リクエスト

Oracle WebLogic Server Adapter で、実行される Web サービスの WSDL に準拠しない SOAP リクエスト・メッセージが受信されると、次の SOAP レスポンスが生成されます。

```
<?xml version="1.0" encoding="ISO-8859-1"
  ?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Body>
  <SOAP-ENV:Fault>
    <faultcode>SOAP-ENV:Server</faultcode>
    <faultstring>RPC サーバー接続に失敗しました : 接続が拒否されました : 接続</faultstring>
  </SOAP-ENV:Fault>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

10.2.2.2 Oracle WebLogic Server Adapter リクエストからの空の結果

Oracle WebLogic Server Adapter で、ターゲット・システム内のレコードと一致しないで渡された入力パラメータを使用して SOAP リクエストを実行する場合、次の SOAP レスポンスが生成されます。

注意： このアダプタの条件では、SOAP フォルトは発生しません。

```
<SOAP-ENV:Envelope xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/1999/XMLSchema">
  <SOAP-ENV:Body>
    <m:RunDBQueryResponse xmlns:m="urn:schemas-iwaysoftware-com:iwse"
xmlns="urn:schemas-iwaysoftware-com:iwse"
cid="2A3CB42703EB20203F91951B89F3C5AF">
      <RunDBQueryResult run="1" />
    </m:RunDBQueryResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

10.2.2.3 エラー・ログイン

Oracle WebLogic Server Adapter で、リクエスト内の無効な SOAP ログを実行すると、次の SOAP レスポンスが生成されます。

```
[2004-07-19T16:28:56:718Z] DEBUG (SOAP1) W.SOAP1.2: POST received
[2004-07-19T16:28:56:718Z] DEBUG (SOAP1) W.SOAP1.2: in XDSOAPHTTPWorker agentName
is [XDSOAPRouter]
[2004-07-19T16:28:56:718Z] DEBUG (SOAP1) W.SOAP1.2: before parse:
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<SOAP-ENV:Header>
<m:ibsinfo xmlns:m="urn:schemas-iwaysoftware-... [861]
[2004-07-19T16:28:56:718Z] ERROR (SOAP1) W.SOAP1.2: Attempting string, no encoding
recognized in document
[2004-07-19T16:28:56:734Z] DEEP (SOAP1) W.SOAP1.2: parse complete in 16 msecs
[2004-07-19T16:28:56:859Z] DEEP (SOAP1) W.SOAP1.2: ST_NODICT
[2004-07-19T16:28:56:859Z] DEEP (SOAP1) W.SOAP1.2: ST_FINISH
[2004-07-19T16:28:56:859Z] DEBUG (SOAP1) extractControl - edaDoc: false
[2004-07-19T16:28:56:859Z] DEBUG (SOAP1) now: 2004-07-19T16:28:56Z expires:
2004-07-20T16:28:56Z
[2004-07-19T16:28:56:859Z] DEBUG (SOAP1) W.SOAP1.2: checking for cached agent
[2004-07-19T16:28:56:859Z] DEBUG (SOAP1) W.SOAP1.2: pushagent: adding agent
com.ibi.iwse.XDSOAPRouter
[2004-07-19T16:28:56:875Z] DEBUG (SOAP1) W.SOAP1.2: inside worker the soap Action
is [B0100033.GetEffectiveAddressRequest#test##]
[2004-07-19T16:28:56:875Z] DEBUG (SOAP1) W.SOAP1.2: precedence: 1
[2004-07-19T16:28:56:875Z] DEBUG (SOAP1) W.SOAP1.2: precedence: 1
[2004-07-19T16:28:56:875Z] DEBUG (SOAP1) W.SOAP1.2: numagents: 1
[2004-07-19T16:28:56:890Z] DEBUG (SOAP1) W.SOAP1.2: running agent 1 name
com.ibi.iwse.XDSOAPRouter document 1
[2004-07-19T16:28:56:890Z] INFO (manager) MGR00X01: Adding active worker:
W.SOAP1.2
[2004-07-19T16:28:56:890Z] DEBUG (SOAP1) W.SOAP1.2: <?xml version="1.0"
encoding="UTF-8" ?>
<SOAP-ENV:Envelope xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
```

```

xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Header>
    <m:ibsinfo xmlns:m="urn:schemas-iwaysoftware-com:iwse">
      <m:service>B0100033</m:service>
      <m:method>GetEffectiveAddress</m:method>
      <m:license>test</m:license>
      <m:Username>user</m:Username>
      <m>Password>password</m>Password>
    </m:ibsinfo>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <m:GetEffectiveAddress
xmlns:m="urn:iwaysoftware:ibse:jul2003:GetEffectiveAddress">
      <m:jdeRequest type="callmethod">
        <m:callMethod name="GetEffectiveAddress">
          <m:params>
            <m:param name="mnAddressNumber">12345</m:param>
          </m:params>
          <m:onError/>
        </m:callMethod>
      </m:jdeRequest>
    </m:GetEffectiveAddress>
  </SOAP-ENV:Body>
  <SOAPAction agentName="XDSOAPRouter"
cid="1FF3D44E0B0AFB2A4E9538ED42B71437">B0100033.GetEffectiveAddressRequest#test##<
/SOAPAction>
</SOAP-ENV:Envelope>
[2004-07-19T16:28:56:890Z] DEBUG (SOAP1) W.SOAP1.2: business method:
m:GetEffectiveAddress
[2004-07-19T16:28:56:906Z] DEBUG (SOAP1) W.SOAP1.2: input:
[2004-07-19T16:28:56:906Z] DEBUG (SOAP1) W.SOAP1.2: <?xml version="1.0"
encoding="UTF-8" ?><jdeRequest xmlns:xsd="http://www.w3.org/2001/XMLSchema"
type="callmethod" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><callMethod
name="GetEffectiveAddress"><params><param name="mnAddressNumber">12345</param>
</params><onError/></callMethod></jdeRequest>
[2004-07-19T16:28:58:234Z] DEBUG (SOAP1) W.SOAP1.2: Agent returned success
[2004-07-19T16:28:58:234Z] INFO (manager) MGR00X02: Removing active worker:
W.SOAP1.2
[2004-07-19T16:28:58:234Z] DEBUG (SOAP1) W.SOAP1.2: doing docTran, docVal,
listTran for agent(1)
[2004-07-19T16:28:58:250Z] DEBUG (SOAP1) W.SOAP1.2: sendToAll reply to XDReply:
[protocol=http */null]
[2004-07-19T16:28:58:250Z] DEBUG (SOAP1) W.SOAP1.2: preemitters from doc: null
[2004-07-19T16:28:58:250Z] DEBUG (SOAP1) W.SOAP1.2: no preemitters, emitting
contents of doc, usestream=false encoding=UTF-8
[2004-07-19T16:28:58:250Z] DEBUG (SOAP1) W.SOAP1.2: writeEntity, len: 670 data:
<?xml version="1.0" encoding="UTF-8" ?><SOAP-ENV:Envelope
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><SOAP-ENV:Body><GetEffective
AddressResponse xmlns="urn:iwaysoftware:ibse:jul2003:GetEffectiveAddress:response"
cid="1FF3D44E0B0AFB2A4E9538ED42B71437"><jdeResponse user="USER" type="callmethod"
session="" environment="DV7333"><returnCode code="12">Environment
&apos;DV7333&apos; could not be initialized for user, check user, pwd and
environment attribute
values</returnCode></jdeResponse></GetEffectiveAddressResponse></SOAP-ENV:Body></S
OAP-ENV:Envelope>

```

```

[2004-07-19T16:28:58:250Z] DEBUG (SOAP1) W.SOAP1.2: writeString: HTTP/1.0
[2004-07-19T16:28:58:250Z] DEBUG (SOAP1) W.SOAP1.2: writeString: 200
[2004-07-19T16:28:58:250Z] DEBUG (SOAP1) W.SOAP1.2: writeString: OK
[2004-07-19T16:28:58:250Z] DEBUG (SOAP1) W.SOAP1.2: writeString: Content-Type:
[2004-07-19T16:28:58:250Z] DEBUG (SOAP1) W.SOAP1.2: writeString: text/xml
[2004-07-19T16:28:58:250Z] DEBUG (SOAP1) W.SOAP1.2: writeString: Content-Length:
[2004-07-19T16:28:58:265Z] DEBUG (SOAP1) W.SOAP1.2: writeString: 670
[2004-07-19T16:28:58:265Z] INFO (SOAP1) W.SOAP1.2: W0000X13: Ended message
processing, rc=0
[2004-07-19T16:28:58:265Z] DEEP (SOAP1) W.SOAP1.2: storing used socket
[2004-07-19T16:28:58:265Z] DEBUG (SOAP1) W.SOAP1.2: entering waitforDocument
[2004-07-19T16:29:03:875Z] DEEP (SOAP1) W.SOAP1.2: cleanup: closing sockets(0)

```

10.2.2.4 Oracle WebLogic Server Adapter リクエストからの空の結果

Oracle WebLogic Server Adapter で、ターゲット・システム内のレコードと一致しない、渡された入力パラメータを使用して SOAP リクエストが実行されると、次の SOAP レスポンスが生成されます。

注意： このアダプタの条件では、SOAP フォルトは発生しません。

```

[2004-07-19T16:27:05:640Z] DEBUG (SOAP1) W.SOAP1.2: POST received
[2004-07-19T16:27:05:640Z] DEBUG (SOAP1) W.SOAP1.2: in XDSOAPHTTPWorker agentName
is [XDSOAPRouter]
[2004-07-19T16:27:05:640Z] DEBUG (SOAP1) W.SOAP1.2: before parse:
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<SOAP-ENV:Body>
<m:GetEffectiveAddress xmlns:m="urn:iwaysoftwar... [590]
[2004-07-19T16:27:05:640Z] ERROR (SOAP1) W.SOAP1.2: Attempting string, no encoding
recognized in document
[2004-07-19T16:27:05:640Z] DEEP (SOAP1) W.SOAP1.2: parse complete in 0 msec
[2004-07-19T16:27:05:781Z] DEEP (SOAP1) W.SOAP1.2: ST_NODICT
[2004-07-19T16:27:05:781Z] DEEP (SOAP1) W.SOAP1.2: ST_FINISH
[2004-07-19T16:27:05:781Z] DEBUG (SOAP1) extractControl - edaDoc: false
[2004-07-19T16:27:05:781Z] DEBUG (SOAP1) now: 2004-07-19T16:27:05Z expires:
2004-07-20T16:27:05Z
[2004-07-19T16:27:05:781Z] DEBUG (SOAP1) W.SOAP1.2: inside isAsync() the soap
Action is ["B0100033.GetEffectiveAddressRequest#test##"]
[2004-07-19T16:27:05:781Z] DEBUG (SOAP1) W.SOAP1.2: inside isAsync() the soap
Action is [B0100033.GetEffectiveAddressRequest#test##]
[2004-07-19T16:27:05:781Z] DEBUG (SOAP1) W.SOAP1.2: checking for cached agent
[2004-07-19T16:27:05:796Z] DEBUG (SOAP1) W.SOAP1.2: pushagent: adding agent
com.ibi.iwse.XDSOAPRouter
[2004-07-19T16:27:05:796Z] DEBUG (SOAP1) W.SOAP1.2: inside worker the soap Action
is [B0100033.GetEffectiveAddressRequest#test##]
[2004-07-19T16:27:05:796Z] DEBUG (SOAP1) W.SOAP1.2: precedence: 1
[2004-07-19T16:27:05:796Z] DEBUG (SOAP1) W.SOAP1.2: precedence: 1
[2004-07-19T16:27:05:796Z] DEBUG (SOAP1) W.SOAP1.2: numagents: 1
[2004-07-19T16:27:05:812Z] DEBUG (SOAP1) W.SOAP1.2: running agent 1 name
com.ibi.iwse.XDSOAPRouter document 1
[2004-07-19T16:27:05:812Z] INFO (manager) MGR00X01: Adding active worker:
W.SOAP1.2
[2004-07-19T16:27:05:812Z] DEBUG (SOAP1) W.SOAP1.2: <?xml version="1.0"
encoding="UTF-8" ?>
<SOAP-ENV:Envelope xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"

```



```

xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Body>
    <m:GetEffectiveAddress
xmlns:m="urn:iwaysoftware:ibse:jul2003:GetEffectiveAddress">
      <m:jdeRequest type="callmethod">
        <m:callMethod name="GetEffectiveAddress">
          <m:params>
            <m:param name="mnAddressNumber">12345</m:param>
          </m:params>
          <m:onError/>
        </m:callMethod>
      </m:jdeRequest>
    </m:GetEffectiveAddress>
  </SOAP-ENV:Body>
  <SOAPAction agentName="XDSOAPRouter"
cid="9F71FEA4C932CD8786F7388D7EF293A1">B0100033.GetEffectiveAddressRequest#test##<
/SOAPAction>
</SOAP-ENV:Envelope>
[2004-07-19T16:27:05:812Z] DEBUG (SOAP1) W.SOAP1.2: business method:
m:GetEffectiveAddress
[2004-07-19T16:27:05:828Z] DEBUG (SOAP1) W.SOAP1.2: input:
[2004-07-19T16:27:05:828Z] DEBUG (SOAP1) W.SOAP1.2: <?xml version="1.0"
encoding="UTF-8" ?><jdeRequest xmlns:xsd="http://www.w3.org/2001/XMLSchema"
type="callmethod" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><callMethod
name="GetEffectiveAddress"><params><param name="mnAddressNumber">12345</param>
</params><onError/></callMethod></jdeRequest>
[2004-07-19T16:27:07:843Z] DEBUG (SOAP1) W.SOAP1.2: Agent returned success
[2004-07-19T16:27:07:843Z] INFO (manager) MGR00X02: Removing active worker:
W.SOAP1.2
[2004-07-19T16:27:07:843Z] DEBUG (SOAP1) W.SOAP1.2: doing docTran, docVal,
listTran for agent(1)
[2004-07-19T16:27:07:859Z] DEBUG (SOAP1) W.SOAP1.2: sendToAll reply to XDReply:
[protocol=http */null]
[2004-07-19T16:27:07:859Z] DEBUG (SOAP1) W.SOAP1.2: preemitters from doc: null
[2004-07-19T16:27:07:859Z] DEBUG (SOAP1) W.SOAP1.2: no preemitters, emitting
contents of doc, usestream=false encoding=UTF-8
[2004-07-19T16:27:07:859Z] DEBUG (SOAP1) W.SOAP1.2: writeEntity, len: 643 data:
<?xml version="1.0" encoding="UTF-8" ?><SOAP-ENV:Envelope
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><SOAP-ENV:Body><GetEffective
AddressResponse xmlns="urn:iwaysoftware:ibse:jul2003:GetEffectiveAddress:response"
cid="9F71FEA4C932CD8786F7388D7EF293A1"><jdeResponse user="JDE" type="callmethod"
environment="DV7333"><callMethod name="GetEffectiveAddress"><returnCode code="2"/>
<params><param
name="mnAddressNumber">12345</param></params></callMethod></jdeResponse></GetEffec
tiveAddressResponse></SOAP-ENV:Body></SOAP-ENV:Envelope>
[2004-07-19T16:27:07:859Z] DEBUG (SOAP1) W.SOAP1.2: writeString: HTTP/1.0
[2004-07-19T16:27:07:859Z] DEBUG (SOAP1) W.SOAP1.2: writeString: 200
[2004-07-19T16:27:07:859Z] DEBUG (SOAP1) W.SOAP1.2: writeString: OK
[2004-07-19T16:27:07:859Z] DEBUG (SOAP1) W.SOAP1.2: writeString: Content-Type:
[2004-07-19T16:27:07:859Z] DEBUG (SOAP1) W.SOAP1.2: writeString: text/xml
[2004-07-19T16:27:07:875Z] DEBUG (SOAP1) W.SOAP1.2: writeString: Content-Length:
[2004-07-19T16:27:07:875Z] DEBUG (SOAP1) W.SOAP1.2: writeString: 643
[2004-07-19T16:27:07:875Z] INFO (SOAP1) W.SOAP1.2: W0000X13: Ended message
processing, rc=0
[2004-07-19T16:27:07:875Z] DEEP (SOAP1) W.SOAP1.2: storing used socket

```

```
[2004-07-19T16:27:07:875Z] DEBUG (SOAP1) W.SOAP1.2: entering waitForDocument
[2004-07-19T16:27:12:781Z] DEEP (SOAP1) W.SOAP1.2: cleanup: closing sockets(0)
```

10.2.2.5 無効なコール・メソッド

無効なコールが Oracle WebLogic Server Adapter に実行されると、次の SOAP レスポンスが生成されます。

```
[2004-07-19T16:24:34:859Z] DEBUG (SOAP1) W.SOAP1.2: POST received
[2004-07-19T16:24:34:859Z] DEBUG (SOAP1) W.SOAP1.2: in XDSOAPHTTPWorker agentName
is [XDSOAPRouter]
[2004-07-19T16:24:34:859Z] DEBUG (SOAP1) W.SOAP1.2: before parse:
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<SOAP-ENV:Body>
<m:GetEffectiveAddress xmlns:m="urn:iwaysoftwar... [581]
[2004-07-19T16:24:34:859Z] ERROR (SOAP1) W.SOAP1.2: Attempting string, no encoding
recognized in document
[2004-07-19T16:24:34:859Z] DEEP (SOAP1) W.SOAP1.2: parse complete in 0 msec
[2004-07-19T16:24:34:875Z] DEEP (SOAP1) W.SOAP1.2: ST_NODICT
[2004-07-19T16:24:34:875Z] DEEP (SOAP1) W.SOAP1.2: ST_FINISH
[2004-07-19T16:24:34:875Z] DEBUG (SOAP1) extractControl - edaDoc: false
[2004-07-19T16:24:34:875Z] DEBUG (SOAP1) now: 2004-07-19T16:24:34Z expires:
2004-07-20T16:24:34Z
[2004-07-19T16:24:34:875Z] DEBUG (SOAP1) W.SOAP1.2: inside isAsync() the soap
Action is ["B0100033.GetEffectiveAddressRequest#test##"]
[2004-07-19T16:24:34:875Z] DEBUG (SOAP1) W.SOAP1.2: inside isAsync() the soap
Action is [B0100033.GetEffectiveAddressRequest#test##]
[2004-07-19T16:24:34:875Z] DEBUG (SOAP1) W.SOAP1.2: checking for cached agent
[2004-07-19T16:24:34:875Z] DEBUG (SOAP1) W.SOAP1.2: pushagent: adding agent
com.ibi.iwse.XDSOAPRouter
[2004-07-19T16:24:34:875Z] DEBUG (SOAP1) W.SOAP1.2: inside worker the soap Action
is [B0100033.GetEffectiveAddressRequest#test##]
[2004-07-19T16:24:34:890Z] DEBUG (SOAP1) W.SOAP1.2: precedence: 1
[2004-07-19T16:24:34:890Z] DEBUG (SOAP1) W.SOAP1.2: precedence: 1
[2004-07-19T16:24:34:890Z] DEBUG (SOAP1) W.SOAP1.2: numagents: 1
[2004-07-19T16:24:34:890Z] DEBUG (SOAP1) W.SOAP1.2: running agent 1 name
com.ibi.iwse.XDSOAPRouter document 1
[2004-07-19T16:24:35:031Z] INFO (manager) MGR00X01: Adding active worker:
W.SOAP1.2
[2004-07-19T16:24:35:031Z] DEBUG (SOAP1) W.SOAP1.2: <?xml version="1.0"
encoding="UTF-8" ?>
<SOAP-ENV:Envelope xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<SOAP-ENV:Body>
<m:GetEffectiveAddress
xmlns:m="urn:iwaysoftware:ibse:jul2003:GetEffectiveAddress">
<m:jdeRequest type="callmethod">
<m:callMethod name="GetAddress">
<m:params>
<m:param name="mnAddressNumber">34518</m:param>
</m:params>
<m:onError/>
</m:callMethod>
</m:jdeRequest>
</m:GetEffectiveAddress>
</SOAP-ENV:Body>
```

```

    <SOAPAction agentName="XDSOAPRouter"
cid="4C0AD8398CB7A5B4DED18057D963AA44">B0100033.GetEffectiveAddressRequest#test##<
/SOAPAction>
</SOAP-ENV:Envelope>
[2004-07-19T16:24:35:031Z] DEBUG (SOAP1) W.SOAP1.2: business method:
m:GetEffectiveAddress
[2004-07-19T16:24:35:031Z] DEBUG (SOAP1) W.SOAP1.2: input:
[2004-07-19T16:24:35:031Z] DEBUG (SOAP1) W.SOAP1.2: <?xml version="1.0"
encoding="UTF-8" ?><jdeRequest xmlns:xsd="http://www.w3.org/2001/XMLSchema"
type="callmethod" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><callMethod
name="GetAddress"><params><param name="mnAddressNumber">34518</param>
</params><onError/></callMethod></jdeRequest>
[2004-07-19T16:24:36:781Z] DEBUG (SOAP1) W.SOAP1.2: Agent returned success
[2004-07-19T16:24:36:781Z] INFO (manager) MGR00X02: Removing active worker:
W.SOAP1.2
[2004-07-19T16:24:36:781Z] DEBUG (SOAP1) W.SOAP1.2: doing docTran, docVal,
listTran for agent(1)
[2004-07-19T16:24:36:781Z] DEBUG (SOAP1) W.SOAP1.2: sendToAll reply to XDReply:
[protocol=http */null]
[2004-07-19T16:24:36:781Z] DEBUG (SOAP1) W.SOAP1.2: preemitters from doc: null
[2004-07-19T16:24:36:781Z] DEBUG (SOAP1) W.SOAP1.2: no preemitters, emitting
contents of doc, usestream=false encoding=UTF-8
[2004-07-19T16:24:36:781Z] DEBUG (SOAP1) W.SOAP1.2: writeEntity, len: 595 data:
<?xml version="1.0" encoding="UTF-8" ?><SOAP-ENV:Envelope
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><SOAP-ENV:Body><GetEffective
AddressResponse xmlns="urn:iwayssoftware:ibse:jul2003:GetEffectiveAddress:response"
cid="4C0AD8398CB7A5B4DED18057D963AA44"><jdeResponse user="JDE" type="callmethod"
environment="DV7333"><callMethod name="GetAddress"><returnCode code="99"><params>
</params></callMethod></jdeResponse></GetEffectiveAddressResponse></SOAP-ENV:Body>
</SOAP-ENV:Envelope>
[2004-07-19T16:24:36:796Z] DEBUG (SOAP1) W.SOAP1.2: writeString: HTTP/1.0
[2004-07-19T16:24:36:796Z] DEBUG (SOAP1) W.SOAP1.2: writeString: 200
[2004-07-19T16:24:36:796Z] DEBUG (SOAP1) W.SOAP1.2: writeString: OK
[2004-07-19T16:24:36:796Z] DEBUG (SOAP1) W.SOAP1.2: writeString: Content-Type:
[2004-07-19T16:24:36:796Z] DEBUG (SOAP1) W.SOAP1.2: writeString: text/xml
[2004-07-19T16:24:36:796Z] DEBUG (SOAP1) W.SOAP1.2: writeString: Content-Length:
[2004-07-19T16:24:36:796Z] DEBUG (SOAP1) W.SOAP1.2: writeString: 595
[2004-07-19T16:24:36:796Z] INFO (SOAP1) W.SOAP1.2: W0000X13: Ended message
processing, rc=0
[2004-07-19T16:24:36:796Z] DEEP (SOAP1) W.SOAP1.2: storing used socket
[2004-07-19T16:24:36:812Z] DEBUG (SOAP1) W.SOAP1.2: entering waitForDocument
[2004-07-19T16:24:42:671Z] DEEP (SOAP1) W.SOAP1.2: cleanup: closing sockets(0)

```

アウトバウンドおよびインバウンド処理用 J.D. Edwards OneWorld の構成

J.D. Edwards OneWorld を使用すると、マスター・ビジネス関数 (MBF) のインバウンド機能を指定できます。

この章では、J.D. Edwards OneWorld でアウトバウンドおよびインバウンド・トランザクション処理を有効にする方法と、XML をサポートするために `jde.ini` ファイルを変更する方法について説明します。次のトピックが含まれています。

- [A.1 項「アウトバウンドおよびインバウンド処理用 JDE.INI ファイルの変更」](#)
- [A.2 項「GenJava 開発ツールの使用 \(アウトバウンド処理\)」](#)
- [A.3 項「J.D. Edwards OneWorld イベントのトリガー」](#)

A.1 アウトバウンドおよびインバウンド処理用 JDE.INI ファイルの変更

この項では、XML コール・オブジェクト・カーネル (アウトバウンドおよびインバウンド処理) の JDE.INI ファイルに必要な設定について説明します。

JDE.INI ファイルを開き、**[JDENET_KERNEL_DEF6]** および **[JDENET_KERNEL_DEF15]** セクションを次のように変更します。

```
[JDENET_KERNEL_DEF6]
krnlName=CALL OBJECT KERNEL
dispatchDLLName=XMLCallObj.dll
dispatchDLLFunction=_XMLTransactionDispatch@28
maxNumberOfProcesses=1
numberOfAutoStartProcesses=1
```

```
[JDENET_KERNEL_DEF15]
krnlName=XML TRANSACTION KERNEL
dispatchDLLName=XMLTransactions.dll
dispatchDLLFunction=_XMLTransactionDispatch@28
maxNumberOfProcesses=1
numberOfAutoStartProcesses=1
```

アンダースコア (`_`) および `@28` を含むパラメータは、Windows NT オペレーティング・システムのみで使用されます。他のオペレーティング・システムの場合は、パラメータを次の表の値に置き換えます。

オペレーティング・システム	Call Object のディスパッチ DLLName	XML Trans のディスパッチ DLLName
AS400	XMLCALLOBJ	XMLTRANS
HP9000B	libxmlcallobj.sl	libxmltransactions.lo
Sun または RS6000	libxmlcallobj.so	Libxmltransactions.so

注意： J.D. Edwards のバージョン B7333(XE) のインストールには、**[JDENET_KERNEL_DEF15]** が含まれていません。そのため、バージョン B7333(XE) を使用する場合は、**jde.ini** ファイルに手動で追加する必要があります。J.D. Edwards の他のすべてのバージョンでは、インストールで **[JDENET_KERNEL_DEF15]** が含まれます。

A.2 GenJava 開発ツールの使用 (アウトバウンド処理)

この項では、J.D. Edwards ビジネス関数へのアクセス用の Java ラッパーを作成するための GenJava 開発ツールの使用方法について説明します。Oracle Application Adapter for J.D. Edwards OneWorld では、これらのラッパーを使用して J.D. Edwards ビジネス関数をコールします。

この項では、次のトピックについて説明します。

■ A.2.1 項「GenJava の実行」

J.D. Edwards には、J.D. Edwards ビジネス関数を Java クラス・ファイルとして外部に公開するのに使用できる GenJava という Java 生成ツールが用意されています。通常、GenJava ツールは、J.D. Edwards システム管理者が実行します。

GenJava の操作時には、CALLBSFN などのビジネス関数のライブラリを指定する必要があります。GenJava は、ビジネス関数の関連 Java クラス・ファイルおよび関連データ構造を作成します。また、GenJava は、ビジネス関数をコンパイルし、Java ドキュメントを生成し、これらを 2 つの .JAR ファイルにパッケージ化します。1 つ目の .JAR ファイルには Java クラスが含まれ、2 つ目の .JAR ファイルには Java ドキュメントが含まれます。

たとえば、GenJava に指定したビジネス関数ライブラリが CALLBSFN である場合、`<install>\system\classes` ディレクトリまたは GenJava によってリダイレクトされた任意のユーザー指定ディレクトリに、次のファイルがあります。

- JDEJAVA_CALLBSFN.xml
- JDEJAVA_CALLBSFNInterop.jar
- JDEJAVA_CALLBSFNInteropDoc.jar

これらが生成されたら、これらのライブラリ・ファイルを CLASSPATH に追加する必要があります。

また、GenJava は、これらのビジネス関数用の Pure Java インタフェースを生成することにより、J.D. Edwards ビジネス関数にアクセスすることもできます。GenJava は、シック・クライアントまたはデプロイメント・サーバーから生成できます。

A.2.1 GenJava の実行

GenJava は、`<install>\system\bin32` ディレクトリにあります。GenJava はコマンドラインから実行します。GenJava には、ラッパーを生成するのに使用できるコマンド・オプションが 2 つあります。

GenJava のコマンド・オプション 1

次のコマンドは、カテゴリ 1 (マスター・ビジネス関数)、カテゴリ 2 (主要ビジネス関数)、カテゴリ 3 (補助ビジネス関数) およびカテゴリ - (未分類のビジネス関数) の Java ラッパーを CALLBSFN ライブラリに生成します。

```
GenJava /Cat 1 /Cat 2 /Cat 3 /Cat - CALLBSFN
```

GenJava のコマンド・オプション 2

GenJava コマンドは、JDEScript ファイルを使用して実行することもでき、J.D. Edwards ログをウィンドウに表示します。ここでは、有効なユーザー ID、パスワードおよび環境を入力する必要があります。

1. エディタを使用して、AddressBook.cmd という新規ファイルを作成し、次のコマンドを入力します。

```
define library CALLBSFN
login
library CALLBSFN
interface AddressBook
import B0100031
import B0100019
import B0100032
import B0100002
import B0100033
build
logout
```

2. 次の GenJava コマンドを実行します。

```
GenJava /cmd .\AddressBook.cmd
```

3. GenJava は、スクリプト・ファイルに記述されるすべてのビジネス関数用として Java で次のラッパー・ファイルを生成します。

- CALLBSFNInterop.jar
- CALLBSFNInteropDoc.jar
- CALLBSFN.xml

注意: これらのラッパー・ファイルの生成時にエラーが発生する場合は、CLASSPATH の設定が正しいことを確認してください。

4. ラッパー・ファイルをリポジトリ・ディレクトリにコピーします。

GenJava を実行するシステムの CLASSPATH に次のファイルが追加されていることを確認します。

- *base_JAR.jar*
- *jdeNet_JAR.jar*
- *system_JAR.jar*
- *connector.jar*
- *xalan.jar*
- *xerces.jar*

これらのファイルは次のディレクトリにあります。

```
<JDE_EnterpriseOne_Home>\System\classes\
```

さらに、JDK のインストールの `\bin` ディレクトリが Java パスに含まれていることを確認します。たとえば、次のようになります。

```
PATH
x:\E900\system\jdk\bin
```

GenJava の使用方法の詳細は、*J.D. Edwards EnterpriseOne Tools 8.98 コネクタ・ガイド* を参照してください。

A.3 J.D. Edwards OneWorld イベントのトリガー

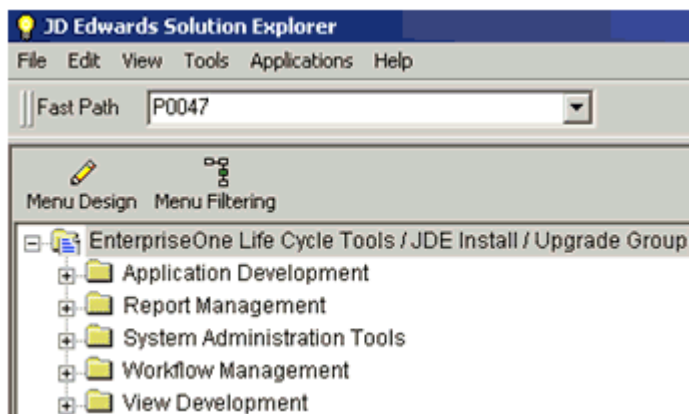
この項では、次のトピックについて説明します。

- [A.3.1 項「アウトバウンド・スケジューラ・サブシステム処理 \(R00460\) の起動」](#)
- [A.3.2 項「サブシステム処理の検証」](#)
- [A.3.3 項「イベントをトリガーするための P4210\(受注オーダー\) の構成」](#)
- [A.3.4 項「構成ステップの検証」](#)

サード・パーティへのインバウンド・データのフローは、データ・エクスポート制御アプリケーションを介して制御されます。トランザクション・タイプとオーダー・タイプごとに、様々な関数名およびライブラリを使用して 1 つ以上のレコードを定義できます。

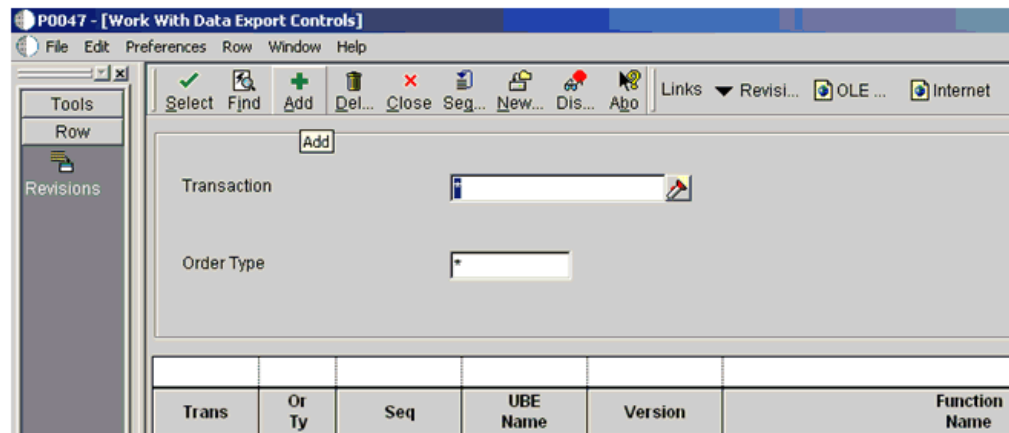
1. [図 A-1](#) に示すように、ファスト・パス・フィールドに「P0047」と入力し、**[Enter]** を押します。

図 A-1 JD Edwards Solution Explorer



[図 A-2](#) に示すように、データ・エクスポート制御の処理ウィンドウが表示されます。

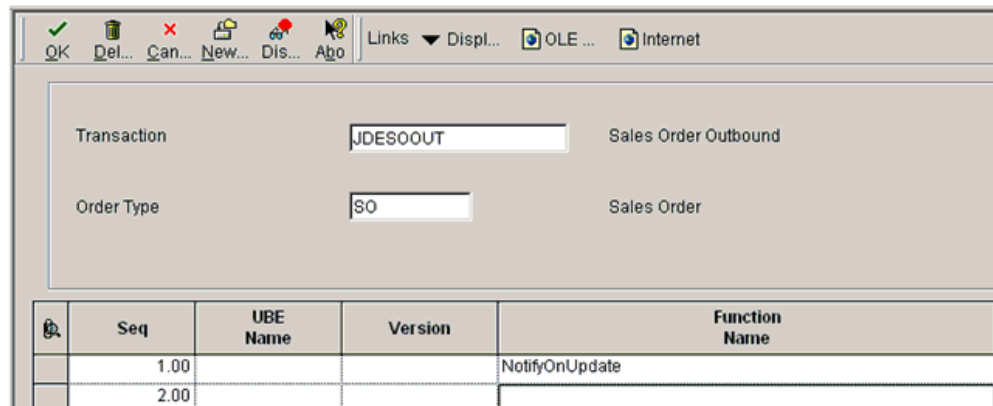
図 A-2 データ・エクスポート制御の処理ウィンドウ



2. 「追加」をクリックします。

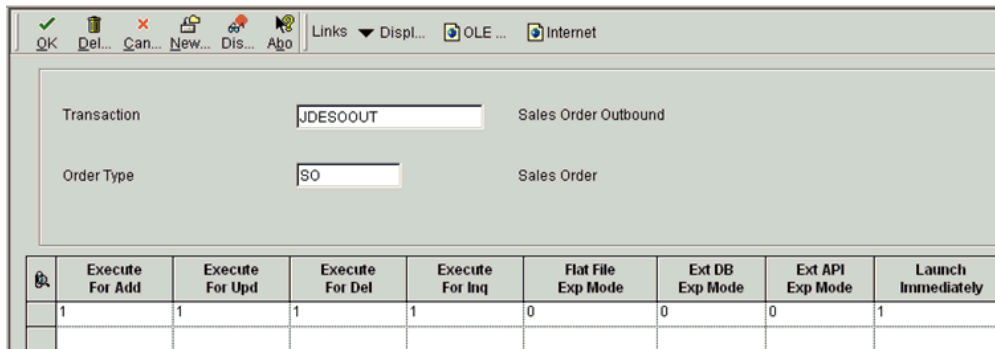
データ・エクスポート制御の改訂ウィンドウが表示されます。図 A-3 に示すように、順序 (Seq) 番号は新規行ごとに自動的に増えていきます。

図 A-3 データ・エクスポート制御の改訂ウィンドウ



3. 次のステップを実行します。
 - a. 「取引」フィールドに「JDES00UT」と入力します。
 - b. 「オーダー・タイプ」フィールドに「SO」と入力します。
 - c. 関数名列の最初の行に「NotifyOnUpdate」と入力します。
 - d. 次のように、関数ライブラリ列の最初の行に、iwoevent.dll ファイルの場所への絶対パスを入力します。
D:\JDEdwards\E812\DDP\Outbound\iwoevent.dll
 - e. 追加 / 挿入に関する通知が必要な場合、追加用に実行列の最初の行に、「1」と入力します。
 - f. 更新、削除および問合せについても同じように判断し、適切な列に「1」と入力します。
 - g. すぐに起動列に「1」と入力し、Outbound Subsystem バッチ処理 (R00460) を起動します。
4. 図 A-4 に示すように、「OK」をクリックします。

図 A-4 データ・エクスポート制御の改訂ウィンドウ



A.3.1 アウトバウンド・スケジューラ・サブシステム処理 (R00460) の起動

各トランザクション・タイプおよびオーダー・タイプに対する 1 つ以上のレコードの定義が終了したら、アウトバウンド・スケジューラ・サブシステム処理を手動で起動する必要があります。

1. 図 A-5 に示すように、ファスト・パス・フィールドに「BV」と入力し、[Enter] を押します。

図 A-5 JD Edwards Solution Explorer

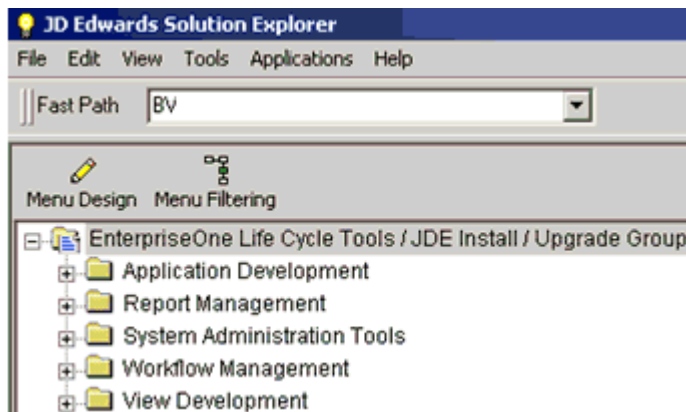
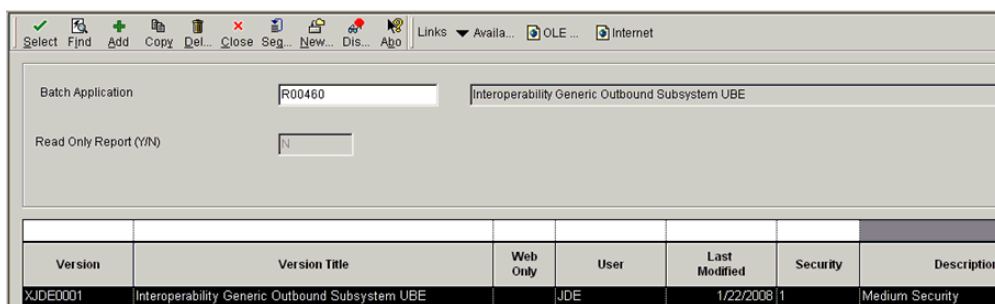


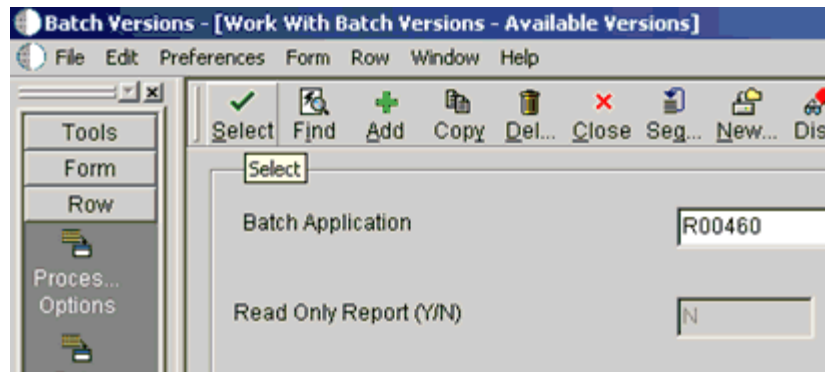
図 A-6 に示すように、バッチ・バージョンの処理 - 使用可能なバージョン・ウィンドウが表示されます。

図 A-6 バッチ・バージョンの処理 - 使用可能なバージョン・ウィンドウ



2. 図 A-7 に示すように、バッチ・アプリケーション・フィールドに「R00460」と入力し、検索をクリックします。

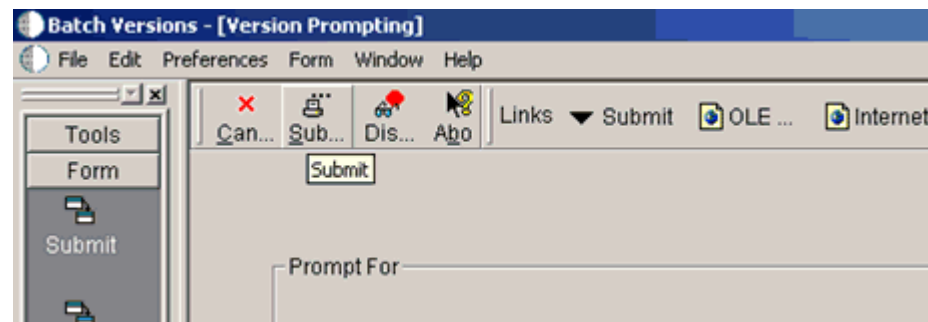
図 A-7 バッチ・アプリケーション・フィールド



3. インタオペラビリティ汎用送信サブシステム UBE(XJDE0001) を選択し、「選択」をクリックします。

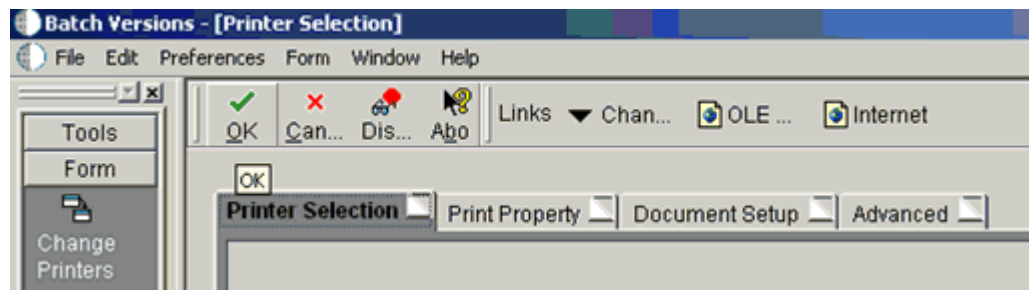
図 A-8 に示すように、バージョン・プロンプト・ウィンドウが表示されます。

図 A-8 バージョン・プロンプト・ウィンドウ



4. 図 A-9 に示すように、「送信」をクリックします。

図 A-9 「OK」オプションの選択



5. 最後の画面に移動し、「OK」をクリックします。

A.3.2 サブシステム処理の検証

この項では、アウトバウンド・スケジューラ・サブシステム処理の検証方法について説明します。

1. 図 A-10 に示すように、ファスト・パス・フィールドに「WSJ」と入力し、[Enter]を押します。

図 A-10 JD Edwards Solution Explorer

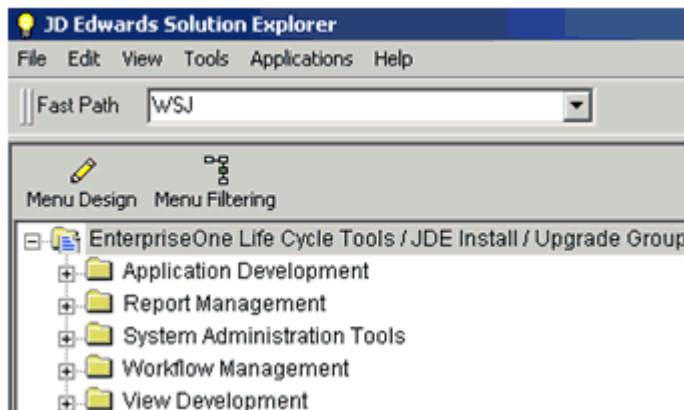
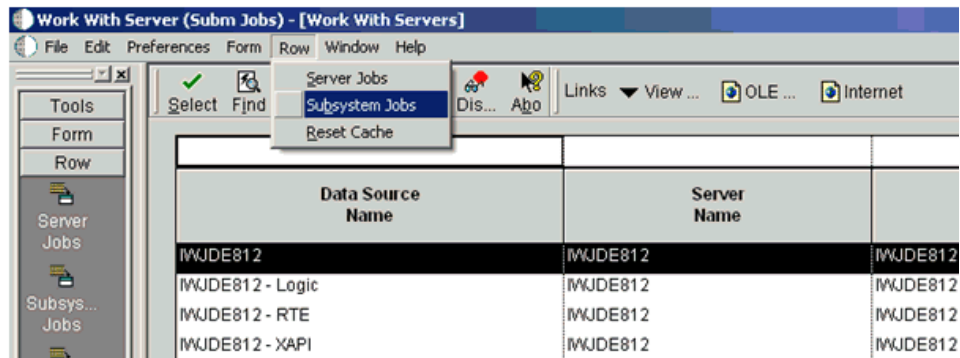


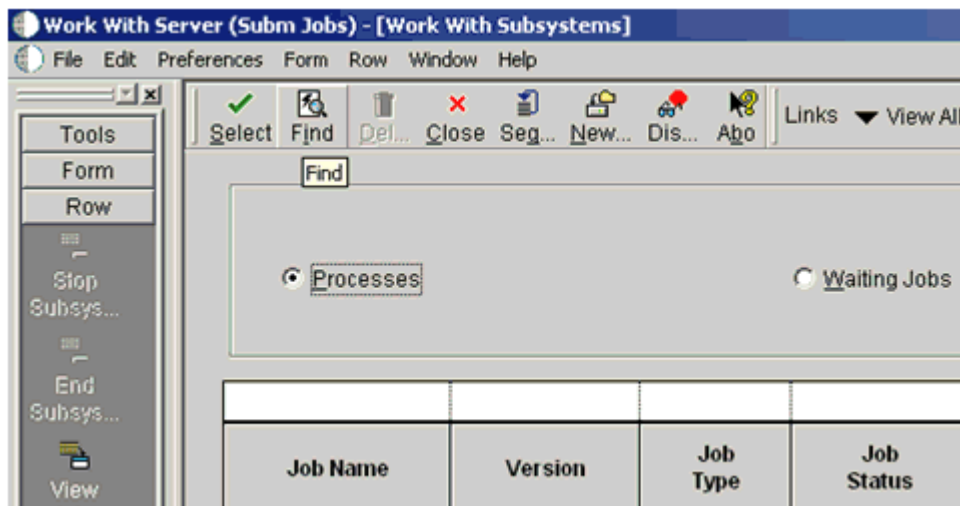
図 A-11 に示すように、サーバーの処理 (Subm Jobs) ウィンドウが表示されます。

図 A-11 サーバーの処理 (Subm Jobs) ウィンドウ



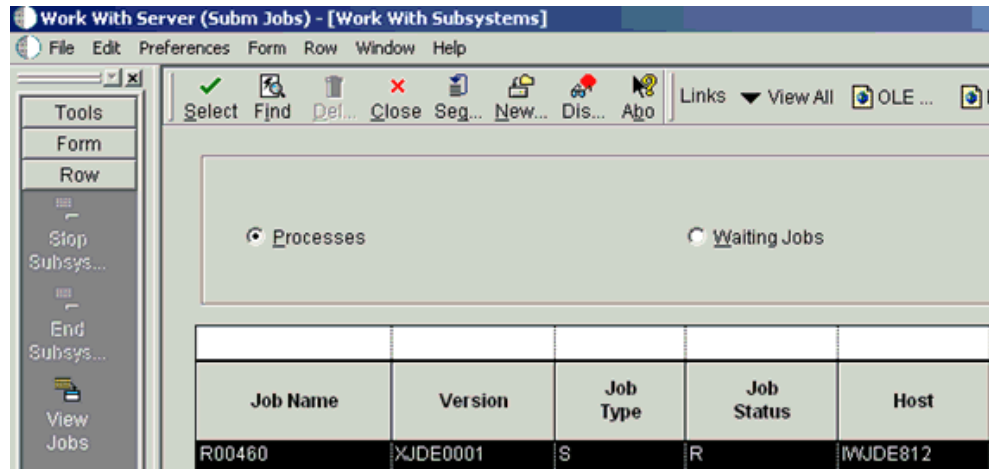
2. 対応するサーバーを表から選択します。
3. 図 A-12 に示すように、メニュー・バーから「ロー」をクリックし、サブシステム・ジョブを選択します。

図 A-12 検索オプションの選択



4. 図 A-13 に示すように、**検索**をクリックします。

図 A-13 「作業状況」列



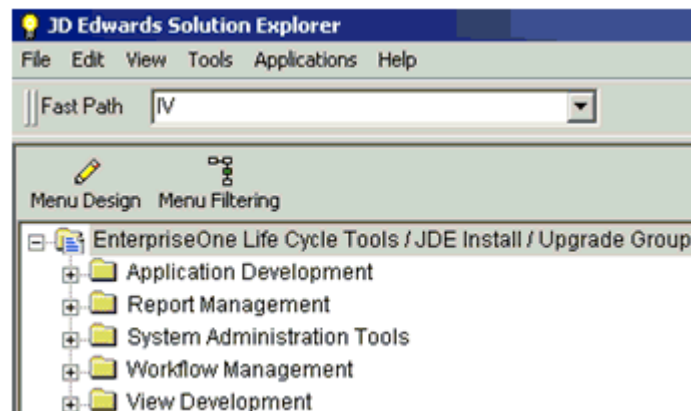
5. 「作業状況」列に「R」がリストされていることを確認します。

A.3.3 イベントをトリガーするための P4210(受注オーダー)の構成

この項では、イベントをトリガーするために P4210(受注オーダー)を構成する方法について説明します。

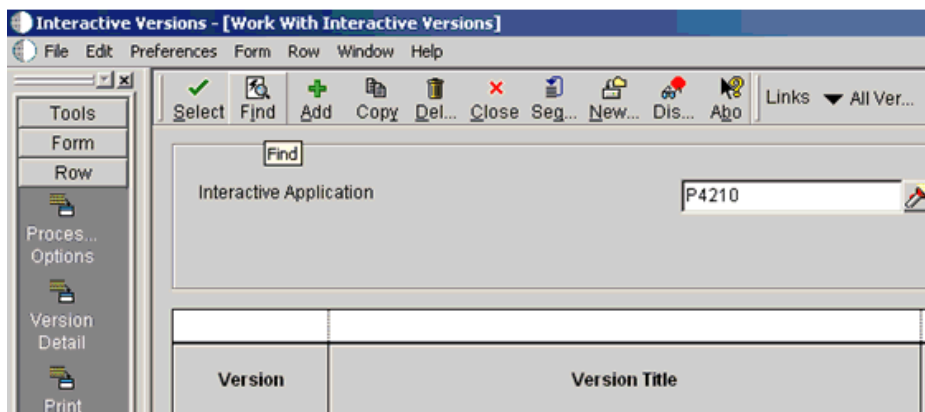
1. 図 A-14 に示すように、ファスト・パス・フィールドに「IV」と入力し、[Enter]を押します。

図 A-14 JD Edwards Solution Explorer



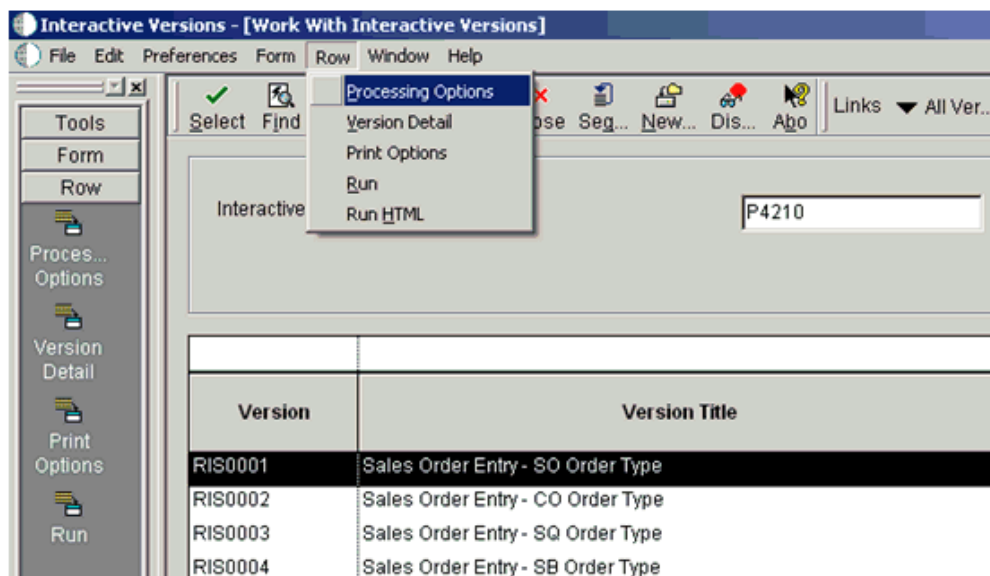
- 図 A-15 に示すように、対話型バージョン・ウィンドウが表示されます。

図 A-15 対話型バージョン・ウィンドウ



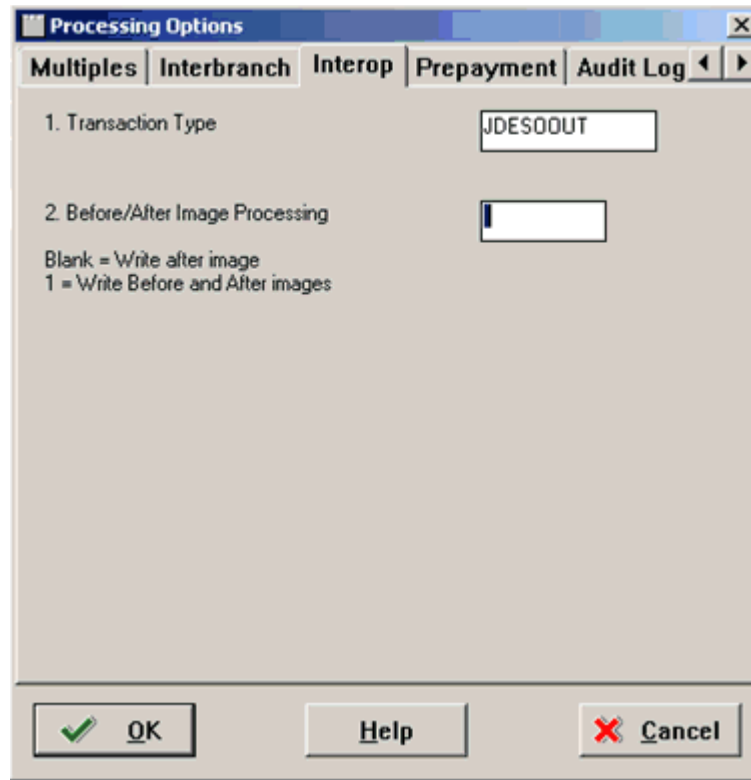
2. 図 A-16 に示すように、対話型アプリケーション・フィールドに「P4210」と入力し、**検索**をクリックします。

図 A-16 対話型アプリケーション・フィールド



3. 表からドキュメント・バージョン (「RIS0001 - Sales Order Entry - SO Order Type」など) を選択します。
4. メニュー・バーから「ロー」をクリックし、**処理オプション**を選択します。
図 A-17 に示すように、処理オプション・ダイアログが表示されます。

図 A-17 処理オプション・ダイアログ



5. 相互運用タブをクリックします。
6. トランザクション・タイプ・フィールドに「JDESOOUT」と入力します。
7. 受注オーダーを追加します。

A.3.4 構成ステップの検証

この項では、F0046 を更新して構成ステップを検証する方法について説明します。

1. 図 A-18 に示すように、ファスト・パス・フィールドに「P0046」と入力し、[Enter] を押します。

図 A-18 JD Edwards Solution Explorer

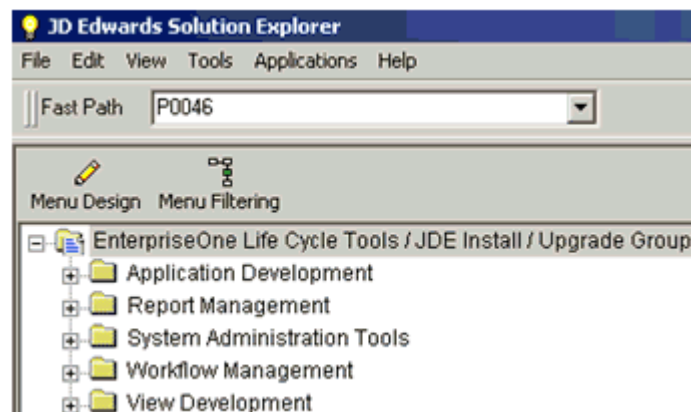
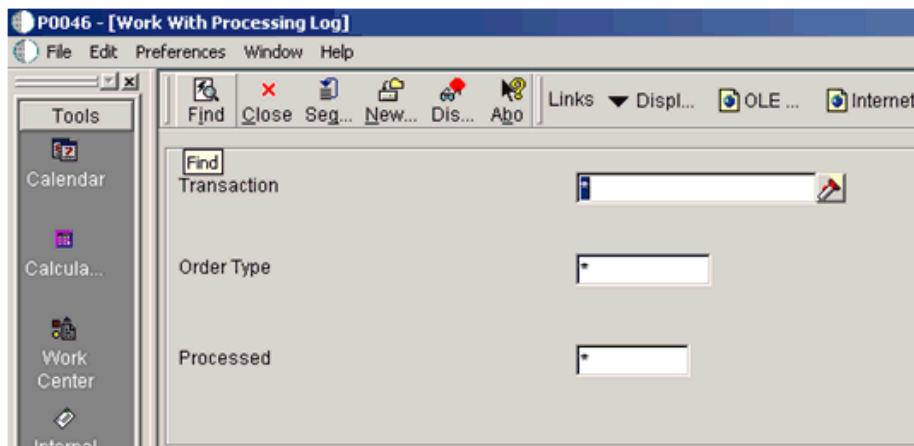


図 A-19 に示すように、P0046 - 処理ログの処理ウィンドウが表示されます。

図 A-19 処理ログの処理ウィンドウ



2. 検索をクリックします。

図 A-20 に示すように、次のデータが表示されます。

図 A-20 データの表示テーブル

User ID	Batch Number	Transaction Number	Line Number	Trans	Or Ty	Seq	UBE Name	Version	S P
JDE	15147	103322	1.000	JDES00UT	SO	1.00			N
JDE	15148	103323	1.000	JDES00UT	SO	1.00			N
JDE	15149	103324	1.000	JDES00UT	SO	1.00			N
JDE	15150	103325	1.000	JDES00UT	SO	1.00			N
JDE	15151	103326	1.000	JDES00UT	SO	1.00			N
JDE	15152	103327	1.000	JDES00UT	SO	1.00			N
JDE	15153	103328	1.000	JDES00UT	SO	1.00			N
JDE	15154	103329	1.000	JDES00UT	SO	1.00			N
JDE	15155	103330	1.000	JDES00UT	SO	1.00			N
JDE	15156	103331	1.000	JDES00UT	SO	1.00			N
JDE	15157	103332	1.000	JDES00UT	SO	1.00			N
JDE	15158	103333	1.000	JDES00UT	SO	1.00			N
JDE	15159	103334	1.000	JDES00UT	SO	1.00			N
JDE	15160	103335	1.000	JDES00UT	SO	1.00			N
JDE	15163	103452	1.000	JDES00UT	SO	1.00			N
JDE	15164	103453	1.000	JDES00UT	SO	1.00			N
JDE	15165	103454	1.000	JDES00UT	SO	1.00			N
JDE	15166	103455	1.000	JDES00UT	SO	1.00			N

3. 対応するトランザクションを検索します。

次のディレクトリに、iwoevent.log ファイルが作成されます。

```
\\iwJDE812\JDEdwards\E812\DDP\system\bin32
```

iwoevent.log ファイルは、iwoevent.dll および iwoevent.cfg ファイルがあるアウトバウンド・フォルダに作成されます。次に、イベント・ログ・ファイルの例を示します。

```
Event call begin...
Server time      : Tue May 27 07:23:55 2008
userId          : JDE
batchNumber     : 15205
transactionNumber: 103494
lineNumber      : 1.000000
transactionType : JDES00UT
sequenceNumber  : 1.000000
Request xml:
=====
<? xml version="1.0" encoding="UTF-8"?><jde><request><connection><dsn /><user
```



```
/><password /><sp><proc>JDES00OUT</proc><data><ediUserId>JDE  
</ediUserId><ediBatchNumber>15205  
</ediBatchNumber><ediTransactionNumber>103494  
</ediTransactionNumber></data></sp></connection></request></jde>  
=====
```

用語集

アダプタ

1つの電子的インタフェースを別の電子的インタフェースに適応させることにより(機能を損なうことなく)一般的な接続性を提供します。

エージェント

リスナーおよびドキュメントにおけるサービス・プロトコルをサポートします。

ビジネス・サービス

Web サービスとしても知られます。Web サービスは、独立型でモジュール化された機能で、オープン標準を使用してネットワーク間で公開およびアクセスが可能です。コンポーネント別のインタフェースの実装で、実行可能なエンティティです。

チャンネル

バックエンド・システムの特定のインスタンスへの構成済接続を表します。1つのチャンネルは、アダプタで管理される特定のリスナーへの1つ以上のイベント・ポートをバインドします。

リスナー

クライアント・アプリケーションからのリクエストを受け入れるコンポーネントです。

ポート

アダプタにより公開される特定のビジネス・オブジェクトを特定の配置と関連付けます。この場合の配置とは、プロトコルおよびイベント・データの場所を定義する URL です。ポートはイベント使用のエンドポイントを定義します。

索引

A

Application Adapter, 1-1 ~ 1-8
 BPEL Process Manager との統合, 6-1
 構成, 3-3
 デプロイ, 1-5, 3-1 ~ 3-3, 4-2
 統合, 5-1
 トラブルシューティング, 10-1 ~ 10-11

B

batch.log ファイル, 2-29
BPEL Process Manager
 OracleAS Adapter for J.D. Edwards OneWorld および, 4-1
 アダプタとの統合, 6-1
 アダプタの統合および, 4-1
BPEL デザイナ, 4-1, 6-1
BSE (OracleAS Adapter Business Services Engine), 1-5
 構成, 2-2
 トラブルシューティング, 10-4 ~ 10-11
BSE デプロイメント, 1-5

C

CCI コール, 3-1

D

DSN(データ・ソース名), 2-31

E

Eclipse 「JDeveloper」を参照
EIS(エンタープライズ情報システム), 1-3
EJB (Enterprise Java Beans), 3-1
Enterprise Connector for J2EE Connector Architecture (J2CA), 1-5, 2-2
Enterprise Java Beans (EJB), 3-1

G

GenJava プログラム, 2-7
GenJava リポジトリ, 2-7

I

「iBSE URL」フィールド, 2-3
InterConnect
 アダプタの統合および, 5-1
IP アドレス, 2-8, 2-31
IWayConfig パラメータ, 3-4
IWayHome パラメータ, 3-4
IWayRepoPassword パラメータ, 3-4
IWayRepoURL パラメータ, 3-4
IWayRepoUser パラメータ, 3-4
iwoevent.cfg ファイル, 2-30 ~ 2-31
iwoevent.cfg ファイルのトランザクション・セクション, 2-30
iwoevent.cfg ファイルの共通セクション, 2-30
iwoevent.cfg ファイルの別名セクション, 2-30
iwoevent.log ファイル, 2-29
IWOEvent リスナー終了, 2-29

J

J2CA (Enterprise Connector for J2EE Connector Architecture), 1-5, 2-2
 OracleAS Adapter および, 3-1
J2CA 構成
 アプリケーション・エクスペローラおよび, 3-3
J2CA サービス, 4-8 ~ 4-35
J2CA リソース・アダプタ, 1-1, 3-1
JAR ファイル, 4-38, 10-2
Java ファイル, 2-7
Java プログラム・クライアント, 3-1
J.D. Edwards OneWorld
 接続, 2-5 ~ 2-8, 10-2
J.D. Edwards OneWorld Event Listener, 2-29 ~ 2-32
J.D. Edwards OneWorld ThinNet API, 1-4, 1-8, 2-10
J.D. Edwards OneWorld への接続, 10-2, 2-5 ~ 2-8
「JDE One World」ダイアログ・ボックス, 2-6
jde TransactionName, 2-31
「JDEdwards」ノード, 4-35
JDeveloper, 4-7, 6-1
「JDE 環境」パラメータ, 2-8, 4-37

L

Loglevel パラメータ, 3-4

M

ManagedConnectionFactory パラメータ, 3-4
MBF(マスター・ビジネス関数), 1-2 ~ 1-5, 2-5,
A-1
実行, 2-10

O

OneWorld Event Listener, 2-29 ~ 2-32
Oracle Application Server
コネクタのデプロイ, 3-1
Oracle JDeveloper, 4-7, 6-1
Oracle JDeveloper BPEL デザイナ「BPEL デザイナ」、
「JDeveloper」または「Oracle JDeveloper」を参
照
Oracle Unified Method (OUM), ix
OracleAS Adapter
J2CA および, 3-1
インストール・ディレクトリおよび, 3-4
OracleAS Adapter Business Services Engine (BSE),
1-5
構成, 2-2
トラブルシューティング, 10-4 ~ 10-11
OracleAS Adapter for J.D. Edwards OneWorld
BPEL Process Manager および, 4-1
InterConnect および, 5-1
構成, 2-1
デプロイ, 1-1, 4-2
トラブルシューティング, 10-1 ~ 10-11
OracleAS Adapter J2CA
アプリケーション・エクスペローラおよび, 10-4
接続, 10-4
OracleAS Adapter J2CA への接続, 10-4
OracleAS Adapter アプリケーション・エクスペロー
ラ「アプリケーション・エクスペローラ」を参照
OracleAS InterConnect
アダプタの統合および, 5-1
Oracle データベース, 3-4

P

Process Manager 「BPEL Process Manager」を参照

S

SOA (サービス指向アーキテクチャ), 4-1
SOAP エージェント, 10-4 ~ 10-5
SOAP フォルト, 10-4 ~ 10-5
SOAP リクエスト, 10-5 ~ 10-8
エラーおよび, 10-4 ~ 10-6, 10-8
SOAP レスポンス, 10-4 ~ 10-8

T

ThinNet API, 1-4, 1-8, 2-10

W

Web Services Description Language (WSDL), 4-35
生成, 4-8, 4-38
Web サービス, 1-1
作成, 2-13 ~ 2-14
テスト, 2-14
デプロイ, 10-4
統合, 4-1
トラブルシューティング, 10-4 ~ 10-11
リポジトリ・プロジェクトおよび, 2-2
Web サービス名, 2-13
WSDL (Web Services Description Language), 4-35
生成, 4-8, 4-38
WSDL ドキュメント, 4-1
「WSDL のエクスポート」ダイアログ・ボックス, 4-8
WSDL ファイル, 4-1, 6-1
アプリケーション・エクスペローラおよび, 4-35
作成, 4-35

X

XDJdeOutboundAgent, 2-29
XMLInterop パラメータ, 4-37
XML スキーマ, 1-3
作成, 2-10
XML ドキュメント, 1-3
「XML」パラメータ, 4-36
XML フォーマット, 1-3
XML メッセージ, 1-2 ~ 1-3

Z

Z ファイル, 2-29

あ

アウトバウンド・エージェント, 2-29
アウトバウンド処理, 1-4, 2-29
アウトバウンド・トランザクション, 2-29
アクション・コード, 10-3
アクセス方法, 2-29
アダプタ, 1-1 ~ 1-8
BPEL Process Manager との統合, 6-1
構成, 3-3
デプロイ, 1-5, 3-1 ~ 3-3, 4-2
統合, 5-1
トラブルシューティング, 10-1 ~ 10-11
アダプタ構成
更新, 3-3
要件, 5-1
アダプタ構成の更新, 3-3
アダプタ・タイプ, 1-1, 1-6
リソース, 3-1 ~ 3-3
アダプタの構成, 3-3
アダプタのデプロイ, 1-5, 3-1 ~ 3-3, 4-2
アダプタの統合
BPEL Process Manager および, 4-1
InterConnect および, 5-1

アダプタの例外, 10-2 ~ 10-5
アドレス番号, 10-3
アプリケーション・アダプタの統合
 BPEL Process Manager および, 4-1
 InterConnect および, 5-1
アプリケーション・エクスプローラ, 1-5, 2-1, 2-15,
 4-7, 4-34
 J2CA 構成および, 3-3
 OracleAS Adapter J2CA および, 10-4
 WSDL ファイルおよび, 4-35
アプリケーション・システムおよび, 2-5
イベント・ポートおよび, 2-15
スキーマおよび, 2-10
チャンネルおよび, 2-15, 2-29
テストおよび, 10-2
デバッグおよび, 10-2
トラブルシューティング, 10-2
アプリケーション・システム, 2-15
 アプリケーション・エクスプローラおよび, 2-5
 サポート, 2-5
「アプリケーション」パラメータ, 4-37

い

イベント, 1-2, 2-15
 構成, 2-15
イベント・アダプタ, 2-15, 2-29
イベント・データ
 受信, 4-34
イベント統合, 4-34
イベント・メッセージ, 4-35
イベント・リスナー, 2-29 ~ 2-31
インストール・ディレクトリ, 3-4
インストール要件
 統合および, 5-1
インバウンド J2CA サービス, 4-35
インバウンド処理, 1-4, A-1

え

エージェント, 1-3
エラー・メッセージ, 10-2 ~ 10-11
 ターゲット・システムおよび, 10-5
エンタープライズ情報システム (EIS), 1-3

か

外部リスナー, 1-3
管理対象コネクタ・ファクトリ, 3-3
管理対象コネクタ・ファクトリ・オブジェクト, 3-3
 複数, 3-5

き

キーブ・アライブ・パラメータ, 4-36
「既存のサービス名」リスト, 2-13
機能的なモデリング, 5-1
キュー, 1-3

け

検査日付, 10-4

こ

構成
 OracleAS Adapter Business Services Engine (BSE),
 2-2 ~ 2-4
 上書き, 3-4
 接続, 2-4
「構成」ノード, 2-2 ~ 2-3
構成パラメータ, 3-4
 IWayConfig, 3-4
 IWayHome, 3-4
 IWayRepoPassword, 3-4
 IWayRepoURL, 3-4
 IWayRepoUser, 3-4
 Loglevel, 3-4
構成要件
 統合および, 5-1
コネクタ
 Oracle Application Server へのデプロイ, 3-1
コネクタ・ファクトリ, 3-3
コネクタ・ファクトリ・オブジェクト, 3-3
 複数, 3-5

さ

「サーバー IP アドレス」パラメータ, 2-8, 4-37
サーバー名, 2-8
「サーバー・ポート」パラメータ, 2-8, 4-37
サービス, 1-2
 作成, 2-13 ~ 2-14
 テスト, 2-14
サービス契約日付, 10-4
サービス指向アーキテクチャ (SOA), 4-1
サービス名, 2-13
「サービス・プロバイダ」リスト, 2-3 ~ 2-4
サブレット, 3-1
最終サービス日付, 10-4

し

事業所品目情報, 10-3
事業所品目レコード, 10-3
事前パーサー・タブ, 4-37
実行時
 構成, 5-2
シナリオ
 統合, 5-1
住所録の改訂プログラム, 10-3
住所録マスター・ファイル, 10-3
「新規コンフィギュレーション」ダイアログ・ボック
 ス, 2-3 ~ 2-4

す

スキーマ, 1-3
 作成, 2-10
「スキーマ・スタイル」パラメータ, 4-38

スキーマの作成, 2-10
スキーマの生成, 2-10

せ

接続

作成, 2-5 ~ 2-8
トラブルシューティング, 10-2 ~ 10-4

接続情報, 2-30, 10-2

「接続」ダイアログ・ボックス, 2-8

接続パラメータ, 10-2

JDE 環境, 2-8

アプリケーション, 2-8
サーバー IP アドレス, 2-8
サーバー・ポート, 2-8
ホスト名, 2-3
ユーザー ID, 2-8
ユーザー・パスワード, 2-8, 4-37

接頭辞の長さパラメータ, 4-36

「説明」フィールド, 2-6, 2-13

そ

相互運用性フレームワーク, 1-3

アウトバウンド処理, 1-4
インバウンド処理, 1-4
ソフトウェア再利用, 5-1

た

ターゲット

削除, 2-9
接続, 2-6 ~ 2-8
定義, 2-6 ~ 2-8
編集, 2-9
接続の切断, 2-9

ターゲット・システム, 10-5

ターゲットの「タイプ」リスト, 2-6

「ターゲットの追加」ダイアログ・ボックス, 2-6

ターゲットの編集, 2-9

ち

チャンネル, 2-15

開始, 4-38
削除, 2-29
作成, 2-15, 2-16, 4-35 ~ 4-38
停止, 4-38
テストおよび, 4-38
デバッグおよび, 4-38
編集, 2-29
ポートおよび, 2-15

チャンネル構成パラメータ, 4-37

JDE 環境, 4-37

XML, 4-36
アプリケーション, 4-37
キープ・アライブ, 4-36
サーバー IP アドレス, 4-37
サーバー・ポート, 4-37
スキーマ・スタイル, 4-38

接頭辞の長さ, 4-36

ポート番号, 4-36

ホスト, 4-36

ユーザー ID, 4-37

ユーザー・パスワード, 4-37

「チャンネル」ノード, 4-35

チャンネルの作成, 4-35 ~ 4-38

「チャンネルの追加」ダイアログ・ボックス, 4-35

チャンネル「リスナー」も参照

て

データベース

Oracle, 3-4

接続, 3-4

データベース接続

開く, 3-4

データベース表, 1-3

データ・エクスポート制御表, 1-3

データ・キュー, 1-3

データ・ソース名 (DSN), 2-31

データ・ディクショナリ, 10-3

デザインタイム, 4-35

構成, 5-2, 4-7, 4-34

デプロイメント

BSE, 1-5

と

同期アクセス, 1-3

「同期タイプ」リスト, 4-36

統合シナリオ, 5-1

統合要件, 5-1

ドキュメント・タイプ

XML, 1-3

リクエスト, 1-3

レスポンス, 1-3

トラブルシューティング, 10-1 ~ 10-11

BSE, 10-4 ~ 10-11

Web サービス, 10-4 ~ 10-11

アプリケーション・エクスプローラ, 10-2

トランザクション処理, A-1, 1-2

トランザクション表, 1-4

トレース情報, 10-1

トレース設定, 2-31

な

内部リスナー, 1-3

「名前」フィールド, 2-6

の

ノード, 2-2 ~ 2-3, 2-8

JDEdwards, 4-35

構成, 2-2 ~ 2-3

チャンネル, 4-35

ノードのリスト, 10-2

ノード・リスト, 10-2

は

パスワード, 3-4
「パスワード」パラメータ, 10-2
バッチ処理, 1-3
「パラメータ」タブ, 2-10
パラメータ・タイプ
 構成, 3-4
 接続, 2-8, 10-2
 チャンネル構成, 4-37

ひ

ビジネス・イベント, 1-2
ビジネス関数, 1-8
 スキーマの作成, 2-10
ビジネス・サービス
 作成, 2-13 ~ 2-14
 テスト, 2-14
日付範囲, 10-4

ふ

フォルト・コード要素, 10-5
フォルト・ストリング要素, 10-5
フラット・ファイル, 1-3
プロセス
 アウトバウンド, 4-8
「プロトコル」リスト, 4-36
プロパティ, 3-4

へ

別名, 2-31

ほ

ポート, 2-15, 2-31
 チャンネルおよび, 2-15
ポート番号, 2-8
「ポート番号」パラメータ, 4-36
「ホスト」パラメータ, 4-36
ホスト名パラメータ, 2-3

ま

マスター・ビジネス関数 (MBF), 1-2 ~ 1-5, 2-5,
 A-1
 実行, 2-10

む

無効なアドレス番号, 10-3
無効なレコード, 10-3

め

「メソッド名」フィールド, 2-13
メッセージ, 1-2 ~ 1-3
 イベント, 4-35
メディエータ・アウトバウンド・プロセス, 5-2
メディエータ・インバウンド・プロセス, 5-11

ゆ

「ユーザー ID」パラメータ, 2-8, 4-37
「ユーザー・パスワード」パラメータ, 2-8, 4-37
「ユーザー名」パラメータ, 10-2
「ユーザー・パスワード」パラメータ, 4-37

ら

ライセンス, 2-13
ライセンスおよびメソッドに関するダイアログ・ボク
 クス, 2-13
「ライセンス」フィールド, 2-13

り

リクエスト・スキーマ, 2-10
「リクエスト・スキーマ」タブ, 2-10
リクエスト・ドキュメント, 1-3
リスナー, 1-3, 2-29 ~ 2-31
リスナー構成ファイル, 2-30, 2-32
リスナー終了, 2-29
リスナー・タイプ, 1-3
リスナー「チャンネル」も参照
リソース・アダプタ, 1-6, 1-1, 3-1 ~ 3-3
「リポジトリ」タブ, 2-7
リポジトリ・プロジェクト
 Web サービスおよび, 2-2
 作成, 2-2 ~ 2-4
リポジトリ・プロジェクトの作成, 2-2 ~ 2-4

れ

レコード
 処理, 10-3
レコード識別子, 2-29
レスポンス・スキーマ, 2-10
「レスポンス・スキーマ」タブ, 2-11
レスポンス・ドキュメント, 1-3

ろ

「ログオン」タブ, 2-7
ログオン・パラメータ, 10-2
ログ・ファイル, 10-1
ログ・レベル
 上書き, 3-4

