Oracle® Fusion Middleware Application Adapters

Application Adapter for Siebel User's Guide for

12*c* (12.2.1.2.0)

E84214-01

December 2016

Provides information on how to integrate with Siebel systems and develop applications.

Oracle Fusion Middleware Application Adapter 12*c* (12.2.1.2.0) for Siebel User's Guide for Oracle WebLogic Server, 12*c* (12.2.1.2.0)

E84214-01

Primary Author: Stefan Kostial

Contributors: Vikas Anand, Marian Jones, Sunil Gopal, Bo Stern

Contents

Preface ix

Audience ix
Documentation Accessibility ix
Related Documents ix
Conventions x

1 Introduction

Adapter Features 1-1
Oracle Adapter Business Services Engine (BSE) Architecture 1-2
Oracle Adapter J2CA Generic Architecture 1-3
The Siebel Application Model 1-4
Integration with Siebel 1-4
Integrating with Siebel EAI Architecture 1-5
Using Application Explorer with Oracle Application Adapter for Siebel 1-5
BSE Versus Oracle Adapter J2CA Deployment 1-6
Sample Projects 1-6
Quick Start Guide 1-8
Installation 1-8
Copying Third-Party Library Files 1-8
Configuration 1-8
WebLogic Server Deployment and Integration 1-10
Creating Configurations, Targets, and Channels in Application Explorer 1-10
Working With Service Components in the SOA Suite 1-10
Working With Oracle Service Bus 1-10
Other Features 1-11

2 Configuring Oracle Application Server Adapter for Siebel

Starting Application Explorer 2-2
Configuring Repository Settings 2-2
Creating a Repository Configuration 2-3
Creating a Configuration for BSE 2-3
Creating a Configuration for J2CA 2-4
Connecting to a BSE or J2CA Configuration 2-5
Establishing a Connection (Target) for Siebel 2-5
Defining a Target to Siebel 2-6

Connecting to a Defined Target 2-9
Disconnecting From Siebel 2-9
Editing a Target 2-10
Deleting a Target to Siebel 2-10
Viewing Application System Objects 2-10
Viewing Metadata 2-11
Creating XML Schemas 2-12
Siebel Schema Considerations 2-13
Creating an XML Schema for a Siebel Business Object or Business Service 2-13
Creating an XML Schema for a Siebel Business Component or Business Service 2-13
Searching for a Specific Siebel Object 2-15
Returning Fields in a Specified Order 2-15
Using QueryWithView 2-16
Siebel Prerequisites for Working With Integration Objects 2-16
Creating Schemas for Siebel Integration Objects 2-17
Creating a Siebel XDR or XSD Schema for a Siebel Integration Object 2-17
Creating Integration Object (IO) Nodes for Siebel 2-19
Creating an XML Schema for a Siebel Integration Object 2-21
Creating a Service Node for a Siebel Business Service 2-22
Creating and Testing a Web Service (BSE Configurations Only) 2-24
Creating a Web Service 2-24
Testing a Web Service 2-25
Generating WSDL (J2CA Configurations Only) 2-27
Configuring an Event Adapter 2-28
Creating and Modifying a Channel 2-29

3 Oracle WebLogic Server Deployment and Integration

Adapter Integration with Oracle WebLogic Server 3-1
Deployment of Adapter 3-1
Updating Adapter Configuration 3-2
Creating a Managed Connector Factory Object 3-3
Creating Multiple Managed Connector Factory Objects 3-4
Modifying WSDL Files for Additional Connection Factory Values 3-6

4 Integration With BPEL Service Components in the Oracle SOA Suite

Overview 4-1
Deployment of Adapter 4-2
Configuring a New Application Server Connection 4-2
Designing an Outbound BPEL Process for Service Integration (J2CA Configuration) 4-7
Generating WSDL for Request/Response Service 4-8
Creating an Empty Composite for SOA 4-9
Defining a BPEL Outbound Process 4-11
Deploying the BPEL Outbound Process 4-28
Invoking the Input XML Document in the Oracle Enterprise Manager Console 4-31
Testing Outbound BPEL and Mediator Processes 4-33
Designing an Inbound BPEL Process for Event Integration (J2CA Configuration) 4-34
Generating WSDL for Event Integration 4-34

Creating an Empty Composite for SOA 4-41
Defining a BPEL Inbound Process 4-42
Deploying the BPEL Inbound Process 4-48
Triggering an Event in Siebel 4-49
Designing an Outbound BPEL Process for Service Integration (BSE Configuration) 4-75
Generating a WSDL File for Request and Response Services Using a Web Service 4-75
Creating an Empty Composite for SOA 4-77
Defining a BPEL Outbound Process 4-77

5 Integration With Mediator Service Components in the Oracle SOA Suite

Configuring a New Application Server Connection 5-2
Configuring a Mediator Outbound Process (J2CA Configuration) 5-2
Creating an Empty Composite for SOA 5-2
Defining a Mediator Outbound Process 5-3
Deploying the Mediator Outbound Process 5-10
Invoking the Input XML Document in the Oracle Enterprise Manager Console 5-11
Configuring a Mediator Inbound Process (J2CA Configuration) 5-11
Creating an Empty Composite for SOA 5-11
Defining a Mediator Inbound Process 5-11
Configuring a Mediator Outbound Process (BSE Configuration) 5-17
Creating an Empty Composite for SOA 5-18
Defining a Mediator Outbound Process 5-18

6 Integration With BPM Service Components in the Oracle SOA Suite

Overview 6-1
Deployment of Adapter 6-1
Configuring a New Application Server Connection 6-2
Designing an Outbound BPM Process Using Transformations for Service Integration (J2CA Configuration) 6-2
Creating an Empty Composite for BPM 6-2
Defining a BPM Outbound Process 6-3
Adjusting for Known Deployment Issues With 12c 6-25
Deploying the BPM Outbound Process 6-25
Invoking the Input XML Document in the Oracle Enterprise Manager Console 6-26
Designing an Inbound BPM Process Using Transformations for Event Integration (J2CA Configuration) 6-27
Creating an Empty Composite for BPM 6-27
Defining a BPM Inbound Process 6-28
Designing an Outbound BPM Process Using Transformations for Service Integration (BSE Configuration) 6-42
Creating an Empty Composite for BPM 6-42
Defining a BPM Outbound Process 6-42

7 Configuring an Outbound and Inbound Process for Oracle Service Bus Using sbconsole

Overview of Application Adapter Integration with Oracle Service Bus 7-1

Configuring an Outbound Process Using sbconsole (J2CA Configuration) 7-2
Starting Oracle Service Bus and Creating Project Folders 7-2
Setting the Class Path for Application Explorer to Integrate With Oracle Service Bus 7-6
Publishing a WSDL From Application Explorer to Oracle Service Bus 7-6
Configuring a WSDL-based Business Service 7-7
Configuring a File Type Business Service 7-9
Configuring a Pipeline With Proxy Service 7-13
Configuring an Inbound Process Using sbconsole (J2CA Configuration) 7-22
Starting Oracle Service Bus and Creating Project Folders 7-22
Setting the Class Path for Application Explorer to Integrate With Oracle Service Bus 7-22
Publishing a WSDL From Application Explorer to Oracle Service Bus 7-23
Configuring a WSDL-based Proxy Service 7-24
Configuring a File Type Business Service 7-25
Configuring a Pipeline 7-29
Configuring an Outbound Process Using sbconsole (BSE Configuration) 7-36
Starting Oracle Service Bus and Creating Project Folders 7-36
Setting the Class Path for Application Explorer to Integrate With Oracle Service Bus 7-36
Publishing a WSDL From Application Explorer to Oracle Service Bus 7-37
Configuring a File Type Business Service 7-38
Configuring a WSDL-based Business Service 7-38
Configuring a Pipeline With Proxy Service 7-40
Configuring JMS Proxy Services Using Oracle Service Bus (J2CA Configuration) 7-49
Configuring HTTP Proxy Services Using Oracle Service Bus (J2CA Configuration) 7-64

8 Configuring an Outbound and Inbound Process for Oracle Service Bus Using JDeveloper

Configuring an OSB Outbound Process Using JDeveloper (J2CA Configuration) 8-1
Creating a Service Bus Application for OSB 8-2
Defining an OSB Outbound Process 8-3
Deploying the OSB Outbound Process 8-16
Configuring an OSB Inbound Process Using JDeveloper (J2CA Configuration) 8-19
Creating a Service Bus Application for OSB 8-19
Defining an OSB Inbound Process 8-19
Deploying the OSB Inbound Process 8-27
Configuring an OSB Outbound Process Using JDeveloper (BSE Configuration) 8-28
Creating a Service Bus Application for OSB 8-28
Defining an OSB Outbound Process 8-28
Deploying the OSB Outbound Process 8-42
Configuring a JMS Inbound Process Using JDeveloper (J2CA Configuration) 8-42
Configuring a JMS Outbound Process Using JDeveloper (J2CA Configuration) 8-48
Configuring an HTTP Outbound Process Using JDeveloper (J2CA Configuration) 8-55

9 Key Features

Configuring the Logging Feature 9-1
Configuring Log File Management for the J2CA Connector Application 9-1
Configuring Log File Management for Business Services Engine (BSE) 9-8
Configuring the Diagnosibility Feature 9-11

Supporting Protocols 9-12

Configuring the SOA Debugging Feature 9-12

Guidelines for Using the SOA Debugger 9-13

Prerequisite 9-13

Debugging a BPEL Process in Oracle JDeveloper 9-13

Debugging an OSB Process in Oracle JDeveloper 9-26

Exception Filter 9-44

Configuring the Exception Filter 9-44

Credential Mapping for Oracle SOA Suite (BPEL, Mediator, or BPM) 9-55

Configuring Credential Mapping 9-56

Credential Mapping for Oracle Service Bus (OSB) Using JDeveloper 9-60

Configuring Credential Mapping 9-61

10 Troubleshooting and Error Messages

Troubleshooting 10-1

General Usage Notes for the Oracle Application Adapter for Siebel 10-1

Application Explorer 10-2

Siebel 10-3

Oracle Adapter J2CA 10-4

BSE Error Messages 10-4

General Error Handling in BSE 10-5

Adapter-Specific Error Handling 10-5

A Using Siebel Workflows

Overview A-1

Siebel Workflows A-1

Using a Policy to Invoke a Siebel EAI Workflow A-2

Siebel Workflow - Outbound A-2

Siebel Workflow - Inbound A-3

Creating a Siebel Workflow A-3

Creating a Siebel Workflow for an Event Using MQSeries Transport A-4

Creating a Siebel Workflow for an Event Using File Transport A-8

Creating a Siebel Workflow for an Event Using HTTP Transport A-13

Creating a Siebel Workflow for a Service Using MQSeries Transport A-16

Creating a Siebel Workflow for a Service Using File Transport A-21

Creating a Siebel Workflow for a Service Using HTTP Transport A-25

Glossary

Index

Preface

Welcome to *Oracle Fusion Middleware Application Adapter for Siebel User's Guide for Oracle WebLogic Server*. This document provides information on how to integrate with Siebel systems and develop applications.

Audience

This document is intended for system administrators and developers who integrate with Siebel systems and develop applications.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc`.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` or visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` if you are hearing impaired.

Related Documents

For more information, see the following documents in the Oracle Enterprise Repository 12*c* (12.2.1.2.0) documentation set:

- *Oracle Fusion Middleware Application Adapters Installation Guide for Oracle WebLogic Server*
- *Oracle Fusion Middleware Application Adapter Upgrade Guide for Oracle WebLogic Server*
- *Oracle Fusion Middleware Application Adapter Best Practices Guide for Oracle WebLogic Server*
- Oracle's Unified Method (OUM)

  A wealth of additional Governance information can be found within Oracle's Unified Method (OUM). OUM can be used by Oracle employees, Oracle Partner Network Certified Partners or Certified Advantage Partners, and Clients who either participate in the OUM Customer Program or are engaged on projects where Oracle provides consulting services. OUM is a web-deployed toolkit for planning, executing and controlling software development and implementation projects.

For more information about OUM, see the OUM FAQ at

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

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1 Introduction

Oracle WebLogic Server connects to a Siebel system through Oracle Application Adapter for Siebel. Oracle Application Adapter for Siebel provides connectivity and carries out interactions on a Siebel system. This chapter contains the following sections:

> **Note:** Throughout this document, *<ORACLE_HOME>* refers to the 12*c* (12.2.1.0.0) SOA/OSB installed home location.
>
> *<ADAPTER_HOME>* refers to the following:
>
> - For SOA:
>
>   ```
>   <ORACLE_HOME>\soa\soa\thirdparty\ApplicationAdapters
>   ```
>
> - For OSB:
>
>   ```
>   <ORACLE_HOME>\osb\3rdparty\ApplicationAdapters
>   ```

- Section 1.1, "Adapter Features"
- Section 1.2, "The Siebel Application Model"
- Section 1.3, "Integration with Siebel"
- Section 1.4, "Using Application Explorer with Oracle Application Adapter for Siebel"
- Section 1.5, "BSE Versus Oracle Adapter J2CA Deployment"
- Section 1.6, "Sample Projects"
- Section 1.7, "Quick Start Guide"

1.1 Adapter Features

Oracle Application Adapter for Siebel provides a means to exchange real-time business data between Siebel systems and other applications, databases, or external business partner systems. The **adapter** enables external applications for inbound and outbound processing with Siebel.

Oracle Application Adapter for Siebel can be deployed as a J2EE Connector Architecture (J2CA) version 1.0 resource adapter. This deployment is referred to as Oracle Adapter J2CA. It can also be deployed as a Web services servlet and as such is referred to as Oracle Adapter Business Services Engine (BSE).

This section contains the following topics:

- Section 1.1.1, "Oracle Adapter Business Services Engine (BSE) Architecture"
- Section 1.1.2, "Oracle Adapter J2CA Generic Architecture"

Oracle Application Adapter for Siebel uses XML messages to enable non-Siebel applications to communicate and exchange transactions with Siebel using services and events. Services and events are defined as follows:

- Services (also known as outbound processing): Enables applications to initiate a Siebel business event.
- Events (also known as inbound processing): Enables applications to access Siebel data only when a Siebel business event occurs.

To support event functionality, channels are supported. A **channel** represents configured connections to particular instances of back-end or other types of systems.

The channel is the adapter component that receives events in real time from the EIS application. The channel component can be a File reader, an HTTP listener, or an MQ listener. A channel is always EIS specific. The adapter supports multiple channels for a particular EIS, which enables the user to choose the optimal channel component based on deployment requirements

Oracle Application Adapter for Siebel:

- Supports synchronous and asynchronous, bidirectional message interactions for Siebel Business Services, Business Components, and Integration Objects.
- Includes Oracle WebLogic Server Adapter Application Explorer (Application Explorer), a GUI tool that uses the Siebel Object Manager to explore Siebel metadata and build XML schemas or Web services.
- Supports Siebel transports—MQSeries, File, and HTTP. It also supports MSMQ messaging.
- XML schemas for Oracle Adapter J2CA.
- Web services for BSE.

Oracle Application Adapter for Siebel supports all 23 Siebel Industry Applications (SIA) through business objects, business components, business services, and integration objects. Siebel Industry Applications include industry verticals such as insurance, high technology, automotive, communications, media, financial services, life sciences, manufacturing, and consumer goods.

Siebel Industry Applications is tailored to the specific business requirements and processes of a particular industry with additional business logic in the form of business objects, business components, business services, and integration objects. Oracle Application Adapter for Siebel exposes and generates metadata and interacts with these industry-specific objects.

See Also: *Oracle Application Server Adapter Concepts Guide*

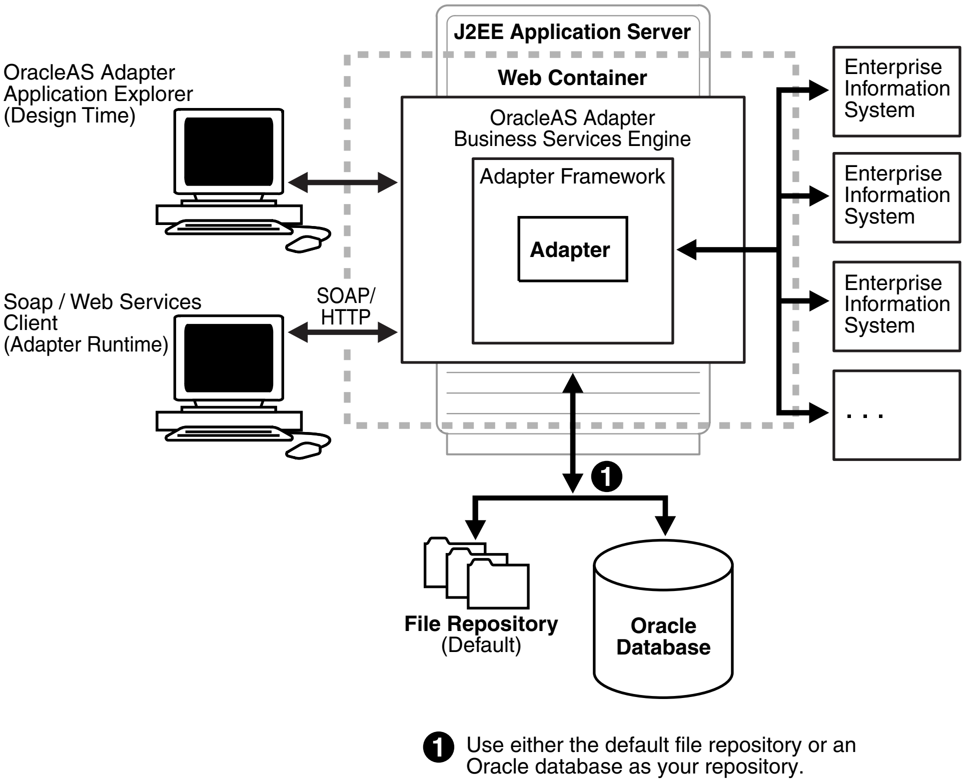
1.1.1 Oracle Adapter Business Services Engine (BSE) Architecture

Figure 1–1 shows the generic architecture for the Oracle Web service adapter for packaged applications. The adapter works with BSE, as deployed to a Web container in a J2EE application server. BSE serves as host to the adapters, enabling Web service requests to the adapters.

Application Explorer, a design-time tool deployed along with BSE, is used to configure adapter connections, browse EIS objects, and configure services. Metadata created while you perform these operations are stored in the repository by BSE.

BSE uses SOAP as a protocol for receiving requests from clients, interacting with the EIS, and sending responses from the EIS back to clients.

Figure 1–1 Oracle Adapter Business Services Engine (BSE) Generic Architecture


J2EE Application Server
Web Container
OracleAS Adapter Application Explorer (Design Time)
OracleAS Adapter Business Services Engine
Adapter Framework
Adapter
Soap / Web Services Client (Adapter Runtime)
SOAP/ HTTP
Enterprise Information System
Enterprise Information System
Enterprise Information System
. . .
1
File Repository (Default)
Oracle Database
1 Use either the default file repository or an Oracle database as your repository.

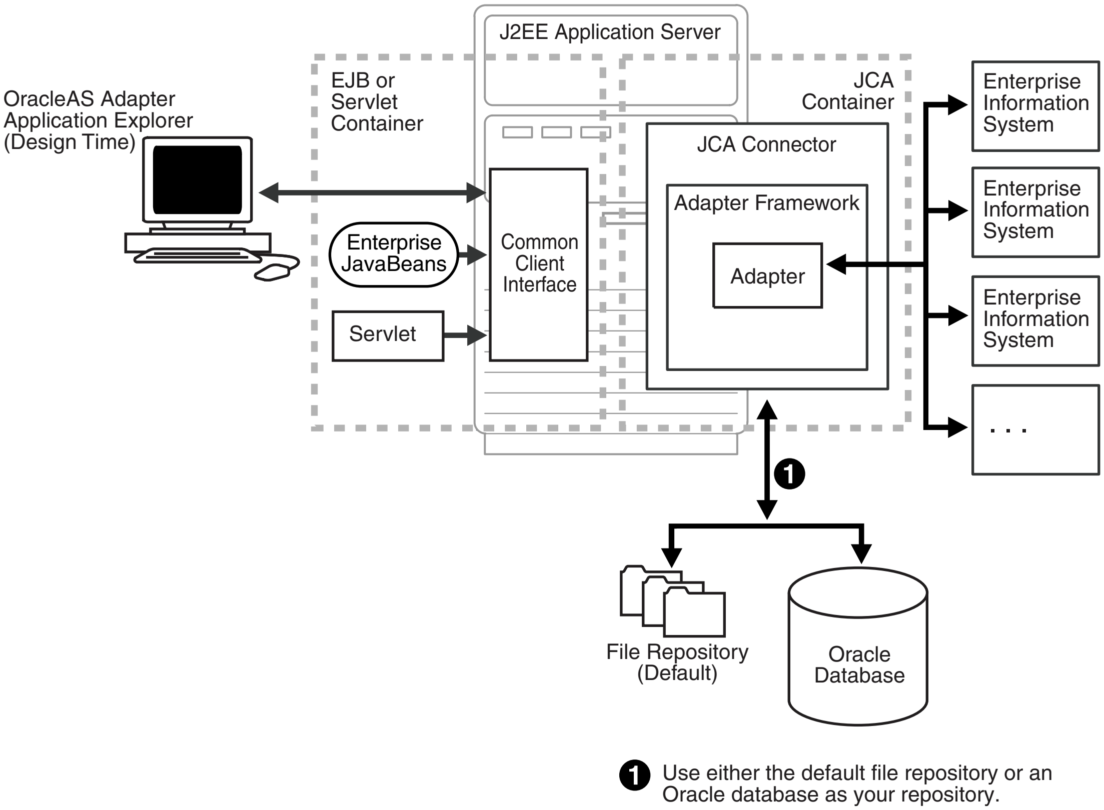
Note: Do not use a file repository for BSE in production environments.

1.1.2 Oracle Adapter J2CA Generic Architecture

Figure 1–2 shows the generic architecture for the Oracle Adapter J2CA for packaged applications. The Oracle Adapter J2CA is deployed to a standard J2CA container and serves as host container to the adapters. The connector is configured with a repository.

Application Explorer, a design tool that works with the connector, is used to configure adapter connections, browse EIS objects, and configure services. Metadata created while you perform these operations are stored in the repository by the connector. The repository can be a file system or an Oracle database. It is deployed as a RAR file and has an associated deployment descriptor called `ra.xml`. You can create multiple connector factories by editing the Oracle WebLogic Server deployment descriptor `ra.xml`. For more information, see Chapter 3, "Oracle WebLogic Server Deployment and Integration".

Figure 1–2 Oracle WebLogic Server Adapter J2CA Generic Architecture


J2EE Application Server
OracleAS Adapter Application Explorer (Design Time)
EJB or Servlet Container
JCA Container
Enterprise JavaBeans
Servlet
Common Client Interface
JCA Connector
Adapter Framework
Adapter
Enterprise Information System
Enterprise Information System
Enterprise Information System
. . .
1
File Repository (Default)
Oracle Database
1 Use either the default file repository or an Oracle database as your repository.

1.2 The Siebel Application Model

The Siebel Enterprise application defines a data abstraction layer that removes dependencies on the underlying database. It accomplishes this by using intermediate Business Components and Business Objects that represent database structures. A Business Component usually represents a table in a database. A Business Object is a group of related business components.

From a given business component, you can navigate the relationships defined for that component to another component. The path you use to traverse component relationships is called the navigation path. For example, if you want to obtain all addresses for a particular account, you can traverse the parent/child relationship between Account and Address to obtain those addresses. By using navigation paths, you can traverse nearly all of the business component relationships defined in the Siebel system.

In Siebel, Integration Objects are similar to Siebel Business Components but describe more complex hierarchal data relationships.

1.3 Integration with Siebel

You can use Oracle Application Adapter for Siebel to initiate a Siebel business process, such as add/update account, or you can use the adapter as part of an integration effort to connect Siebel and non-Siebel systems. Oracle Application Adapter for Siebel is bidirectional and can detect an event from Siebel by receiving a Siebel XML document emitted by Siebel.

This section contains the following topic:

- Section 1.3.1, "Integrating with Siebel EAI Architecture"

When integrating with Siebel using Siebel XML documents, the adapter application developer must use existing Siebel Integration Objects or create new Siebel Integration Objects to use within a Siebel Workflow. The Workflow processes inbound or outbound Siebel XML and uses various transports such as MQSeries, File, and HTTP to exchange transactions with external systems. The Siebel Workflow is usually created by the Siebel administrator or developer using Siebel Workflow Administration screens.

When integrating with Siebel directly using the Java Data Bean or COM Data Interface, Oracle Application Adapter for Siebel does not require a Siebel Integration Object or Siebel Workflow. Instead, it executes Siebel Business Services and Siebel Business Components directly.

The following table lists Siebel objects and processes.

Table 1–1 Siebel Objects and Processes

Siebel Objects	API or Transport	Process
Business Services	Java Data Bean (Siebel Version 6.3-8.0)	Service
	Com Data Interface (Siebel Version 6.01-6.2)	
Business Components	Java Data Bean (Siebel Version 6.3-8.0)	Service
	Com Data Interface (Siebel Version 6.01-6.2)	
Integration Objects	File	Event, Service
	HTTP	Event, Service
	MQSeries	Event, Service
	MQ Read	Service

1.3.1 Integrating with Siebel EAI Architecture

Siebel enables integration with other applications and systems using its Siebel EAI (Enterprise Application Integration) framework and its Business Integration Manager facility. Oracle Application Adapter for Siebel uses the Siebel EAI framework and leverages various integration access methods to provide the greatest amount of flexibility and functionality while working within the Siebel framework.

Oracle Application Adapter for Siebel supports the following integration access methods:

- Siebel Java Data Bean for services involving Siebel Business Components or Siebel Business Services.
- Siebel COM Data Interface for services involving Siebel Business Components or Siebel Business Services.
- Siebel XML for events and services involving Siebel Integration Objects.

1.4 Using Application Explorer with Oracle Application Adapter for Siebel

Application Explorer uses an explorer metaphor for browsing the Siebel system for Business Services, Business Objects, Business Components, and Integration Objects. The explorer enables you to create XML schemas and Web services for the associated object. External applications that access Siebel through Oracle Application Adapter for

Siebel use either XML schemas or Web services to pass data between the external application and the adapter.

Application Explorer uses interfaces provided by Siebel and in-depth knowledge of the Siebel application systems to access and browse business object metadata. After an object is selected, Application Explorer can generate an XML schema or Web service to define the object for use with Oracle Application Adapter for Siebel.

Key features of Application Explorer include:

- The ability to connect to and explore a variety of application systems.
- Access to application system object metadata.
- A point-and-click process for generating XML schemas and Web services.

> **See Also:**
>
> - *Oracle Application Server Adapter Concepts Guide*
> - *Oracle Fusion Middleware Application Adapters Installation Guide for Oracle WebLogic Server*

1.5 BSE Versus Oracle Adapter J2CA Deployment

If you are using Oracle Application Adapter for Siebel with Oracle SOA Suite components (for example, BPEL, Mediator, BPM, or OSB), then note that:

- Only Oracle Adapter J2CA deployment supports inbound integration (event notification) with Oracle SOA Suite components.
- Oracle Adapter J2CA and BSE deployments support outbound integration (request-response service) with Oracle SOA Suite components.

The following two factors explain the differences between deploying BSE and Oracle Adapter J2CA. Understanding these factors can help in selecting a deployment option.

1. BSE has the following advantages:
   - Can be deployed in a separate instance of Oracle WebLogic Server.
   - Provides better distribution of load.
   - Conforms more closely to the Service Oriented Architecture (SOA) model for building applications.
2. Oracle Adapter J2CA does provide slightly better performance than BSE.

1.6 Sample Projects

Sample projects for the Oracle Application Adapter for Siebel that demonstrate outbound and inbound integration scenarios using Oracle BPEL, Mediator, BPM, and OSB tools are packaged with the Application Adapters installation. The following table lists the locations of the sample projects:

Sample Project	Location
Outbound BPEL Process (J2CA)	`<ADAPTER_HOME>\etc\sample\SIEBEL_Samples.zip\SIEBEL_Samples\BPEL\J2CA\Outbound_Project`

Sample Project	Location
Inbound BPEL Process (J2CA)	`<ADAPTER_HOME>\etc\sample\SIEBEL_Samples.zip\SIEBEL_Samples\BPEL\J2CA\Inbound_Project`
Outbound BPEL Process (BSE)	`<ADAPTER_HOME>\etc\sample\SIEBEL_Samples.zip\SIEBEL_Samples\BPEL\BSE\Outbound_Project`
Outbound Mediator Process (J2CA)	`<ADAPTER_HOME>\etc\sample\SIEBEL_Samples.zip\SIEBEL_Samples\Mediator\J2CA\Outbound_Project`
Inbound Mediator Process (J2CA)	`<ADAPTER_HOME>\etc\sample\SIEBEL_Samples.zip\SIEBEL_Samples\Mediator\J2CA\Inbound_Project`
Outbound Mediator Process (BSE)	`<ADAPTER_HOME>\etc\sample\SIEBEL_Samples.zip\SIEBEL_Samples\Mediator\BSE\Outbound_Project`
Outbound BPM Process (J2CA)	`<ADAPTER_HOME>\etc\sample\SIEBEL_Samples.zip\SIEBEL_Samples\BPM\J2CA\Siebel_Sample_J2CA_BPM_Outbound_Project`
Inbound BPM Process (J2CA)	`<ADAPTER_HOME>\etc\sample\SIEBEL_Samples.zip\SIEBEL_Samples\BPM\J2CA\Inbound_Project`
Outbound BPM Process (BSE)	`<ADAPTER_HOME>\etc\sample\SIEBEL_Samples.zip\SIEBEL_Samples\BPM\BSE\Outbound_Project`
Outbound OSB sbconsole Process (J2CA)	`<ADAPTER_HOME>\etc\sample\SIEBEL_Samples.zip\SIEBEL_Samples\OSB\J2CA\Siebel_Sample_J2CA_OSB_Outbound_Project`
Inbound OSB sbconsole Process (J2CA)	`<ADAPTER_HOME>\etc\sample\SIEBEL_Samples.zip\SIEBEL_Samples\OSB\J2CA\Siebel_Sample_J2CA_OSB_Inbound_Project`
Outbound OSB sbconsole Process (BSE)	`<ADAPTER_HOME>\etc\sample\SIEBEL_Samples.zip\SIEBEL_Samples\OSB\BSE\Siebel_Sample_BSE_OSB_Outbound_Project`
Outbound OSB Jdeveloper Process (J2CA)	`<ADAPTER_HOME>\etc\sample\SIEBEL_Samples.zip\SIEBEL_Samples\OSB_Jdeveloper\J2CA\Siebel_Sample_J2CA_OSB_Outbound_Project`
Inbound OSB Jdeveloper Process (J2CA)	`<ADAPTER_HOME>\etc\sample\SIEBEL_Samples.zip\SIEBEL_Samples\OSB_Jdeveloper\J2CA\Siebel_Sample_J2CA_OSB_Inbound_Project`
Outbound OSB Jdeveloper Process (BSE)	`<ADAPTER_HOME>\etc\sample\SIEBEL_Samples.zip\SIEBEL_Samples\OSB_Jdeveloper\BSE\Siebel_Sample_BSE_OSB_Outbound_Project`

1.7 Quick Start Guide

This section enables you to quickly learn the basic steps to install and configure Oracle Application Adapter for Siebel and to use it immediately. It includes the following topics:

- Section 1.7.1, "Installation"
- Section 1.7.2, "Copying Third-Party Library Files"
- Section 1.7.3, "Configuration"
- Section 1.7.4, "WebLogic Server Deployment and Integration"
- Section 1.7.5, "Creating Configurations, Targets, and Channels in Application Explorer"
- Section 1.7.6, "Working With Service Components in the SOA Suite"
- Section 1.7.7, "Working With Oracle Service Bus"
- Section 1.7.8, "Other Features"

1.7.1 Installation

To install Oracle Application Adapter for Siebel, download the Oracle Fusion Middleware Application Adapters installer and complete the installation for SOA/OSB.

For more information on installing the Oracle Fusion Middleware Application Adapters, see the *Oracle Fusion Middleware Application Adapters Installation Guide for Oracle WebLogic Server*.

1.7.2 Copying Third-Party Library Files

Once the adapter installation is completed, copy the required third-party library files for Siebel to the following directories:

```
<ADAPTER_HOME>\lib

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

For more information on encoding settings and prerequisites for Siebel versions 6.2 and lower, see the following topics in Chapter 2, "Configuring Oracle Application Server Adapter for Siebel":

- Encoding Support on UNIX Platforms
- Adding Required Encoding Option (All UNIX Platforms)
- Siebel Connectivity Prerequisites for Versions 6.2 and Lower

1.7.3 Configuration

Navigate to <*ADAPTER_HOME*> and make the following changes:

1. Open `iwafjca.rar\META-INF\ra.xml` and add the following values under the specified config-property-name parameters, as shown in Table 1–2.

Table 1–2

Config-Property-Name	Config-Property-Value
IWayHome	*<ADAPTER_HOME>* For example: - **For SOA:** `C:\12C_soa\soa\soa\thirdparty\ApplicationAdapters` - **For OSB:** `C:\12c_OSB\osb\3rdparty\ApplicationAdapters`
IWayConfig	The name of the configuration. For example: `jca_sample`

2. Open *ibse.war\WEB-INF\web.xml* and add the following values under the specified param-name parameters, as shown in Table 1–3.

Table 1–3

Param-Name	Param-Value
ibseroot	*<ADAPTER_HOME>\ibse.war* For example: - **For SOA:** `C:\12C_soa\soa\soa\thirdparty\ApplicationAdapters\ibse.war` - **For OSB:** `C:\12c_OSB\osb\3rdparty\ApplicationAdapters\ibse.war`
IWay.home	*<ADAPTER_HOME>* For example: - **For SOA:** `C:\12C_soa\soa\soa\thirdparty\ApplicationAdapters` - **For OSB:** `C:\12c_OSB\osb\3rdparty\ApplicationAdapters`
Iway.config	The name of the configuration. For example: `IBSE`

Note: These steps are provided only when configuring a File repository. For more information about configuring a database repository and general configuration information, see Chapter 2, "Configuring Oracle Application Server Adapter for Siebel" and Chapter 3, "Oracle WebLogic Server Deployment and Integration".

1.7.4 WebLogic Server Deployment and Integration

1. Start the WebLogic server and open the WebLogic console.
2. Deploy the adapter components (ibse.war, iwafjca.war, and iwafjca.rar files) and start the deployed adapter components.

For more information on deployment, integration, and target creation, see Chapter 3, "Oracle WebLogic Server Deployment and Integration".

1.7.5 Creating Configurations, Targets, and Channels in Application Explorer

For more information on creating configurations, targets, and channels in Application Explorer, see the following sections in this user's guide:

- Starting Application Explorer: Section 2.1, "Starting Application Explorer"
- Creating a BSE Configuration: Section 2.3.1, "Creating a Configuration for BSE"
- Creating a J2CA Configuration: Section 2.3.2, "Creating a Configuration for J2CA"
- Connecting the Created Configurations: Section 2.3.3, "Connecting to a BSE or J2CA Configuration"
- Creating and Connecting to Targets: Section 2.4, "Establishing a Connection (Target) for Siebel"
- Working with Integration Objects: Section 2.7, "Siebel Prerequisites for Working With Integration Objects", Section 2.8, "Creating Schemas for Siebel Integration Objects", and Section 2.9, "Creating Integration Object (IO) Nodes for Siebel".
- Working With Service Nodes: Section 2.10, "Creating a Service Node for a Siebel Business Service"
- Creating and Testing Web Services: Section 2.11, "Creating and Testing a Web Service (BSE Configurations Only)"
- Generating WSDL Files: Section 2.12, "Generating WSDL (J2CA Configurations Only)"
- Creating and Working With Channels: Section 2.13, "Configuring an Event Adapter"

1.7.6 Working With Service Components in the SOA Suite

Oracle Application Adapter for Siebel integrates with service components in SOA suite such as BPEL, Mediator, and BPM. Required processes are created in JDeveloper and then deployed to the SOA server.

For more information on working with BPEL, Mediator, and BPM service components, see:

- Chapter 4, "Integration With BPEL Service Components in the Oracle SOA Suite"
- Chapter 5, "Integration With Mediator Service Components in the Oracle SOA Suite"
- Chapter 6, "Integration With BPM Service Components in the Oracle SOA Suite"

1.7.7 Working With Oracle Service Bus

Oracle Application Adapter for Siebel integrates with Oracle Service Bus (OSB) to facilitate Web service integration. Required processes are created in the Oracle Service

Bus Console. The process can also be created in JDeveloper and then deployed to the SOA server.

For more information on working with OSB Console, see Chapter 7, "Configuring an Outbound and Inbound Process for Oracle Service Bus Using sbconsole".

For more information on working with OSB Jdeveloper, see Chapter 8, "Configuring an Outbound and Inbound Process for Oracle Service Bus Using JDeveloper".

1.7.8 Other Features

The following is list of other features and their relevant sections in this user's guide:

- Configuring the Exception Filter: Section 9.4, "Exception Filter"
- Configuring Credential Mapping:
  - Section 9.5, "Credential Mapping for Oracle SOA Suite (BPEL, Mediator, or BPM)"
  - Section 9.6, "Credential Mapping for Oracle Service Bus (OSB) Using JDeveloper"

2 Configuring Oracle Application Server Adapter for Siebel

This chapter describes how to configure Oracle Application Adapter for Siebel and create schemas for Siebel Business Objects. It contains the following sections:

- Section 2.1, "Starting Application Explorer"
- Section 2.2, "Configuring Repository Settings"
- Section 2.3, "Creating a Repository Configuration"
- Section 2.4, "Establishing a Connection (Target) for Siebel"
- Section 2.5, "Viewing Application System Objects"
- Section 2.6, "Creating XML Schemas"
- Section 2.7, "Siebel Prerequisites for Working With Integration Objects"
- Section 2.8, "Creating Schemas for Siebel Integration Objects"
- Section 2.9, "Creating Integration Object (IO) Nodes for Siebel"
- Section 2.10, "Creating a Service Node for a Siebel Business Service"
- Section 2.11, "Creating and Testing a Web Service (BSE Configurations Only)"
- Section 2.12, "Generating WSDL (J2CA Configurations Only)"
- Section 2.13, "Configuring an Event Adapter"

Siebel Connectivity Prerequisites for Versions 6.2 and Lower

For Siebel versions 6.2 and lower only, you must perform the following steps to connect to your Siebel system using COM connectivity for a J2CA configuration.

1. Install Siebel thick client on the same system where the adapters are installed.
2. Install the database client (Microsoft SQL Server or Oracle) on the same system.
3. The Siebel .DLL files (iwsiebel.local.dll and iwsiebel.core.dll) in the adapter lib folder must be added to the Application server path.
4. Edit the uagent.cfg file and change the data source parameter value from "local" to "server".

   The uagent.cfg file can be found in the following Siebel thick client folder:

   ```
   c:\sea\client\bin
   ```

5. Edit the data source for SEA MSQl with appropriate parameters.

You can edit a data source in Windows by accessing the Control Panel, Administrative Tools, and Data Sources (ODBC).

6. Use the following target type when creating the adapter target connection:

   ```
   Siebel 6.2 - (Local COM Access Implementation)
   ```

7. Provide the full path to the uagent.cfg file when creating an adapter target connection, for example:

   ```
   c:\sea\client\bin\uagent.cfg
   ```

2.1 Starting Application Explorer

To start Application Explorer:

1. Ensure that Oracle WebLogic Server is started, which is where Application Explorer is deployed.
2. Open the command prompt.
3. Navigate to the following directory:

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

4. Execute *setDomainEnv.cmd* (Windows) or *. ./setDomainEnv.sh* (UNIX/Linux).

   This command sets the class path and other environment variables for Application Explorer in the Oracle WebLogic Server environment. In addition, it allows Application Explorer to access the Oracle WebLogic Server APIs to publish WSDL files to the Oracle Service Bus (OSB) Console.

5. Do not close the command prompt.
6. Navigate to the following directory:

   ```
   <ADAPTER_HOME>\tools\iwae\bin
   ```

7. Execute *ae.bat* (Windows) or *iwae.sh* (UNIX/Linux) to start Application Explorer.

Application Explorer starts. You are ready to define new targets to your Siebel system.

Note: Before you run the **iwae.sh** file on UNIX or Linux platforms, the permissions must be changed. For example:

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

2.2 Configuring Repository Settings

A repository holds information about configuration details, adapter targets, channels, and other configuration information. For more information on how to configure BSE and J2CA repository settings, see the *Oracle Fusion Middleware Application Adapters Installation Guide for Oracle WebLogic Server* (Section 2.7.4 "Configuring the Oracle Database Repository").

2.3 Creating a Repository Configuration

Before you use Application Explorer with Oracle Application Adapter for Siebel, you must create a repository configuration. You can create two kinds of repository configurations, Web services and J2CA, depending on the container to which the adapter is deployed.

This section contains the following topics:

- Section 2.3.1, "Creating a Configuration for BSE"
- Section 2.3.2, "Creating a Configuration for J2CA"
- Section 2.3.3, "Connecting to a BSE or J2CA Configuration"

During design time, the repository is used to store metadata created when using Application Explorer to configure adapter connections, browse EIS objects, configure services, and configure listeners to listen for EIS events. The information in the repository is also referenced at run-time.

Web services and BSE refer to the same type of deployment. For more information, see "Adapter Features" on page 1-1.

2.3.1 Creating a Configuration for BSE

To create a configuration for BSE using Application Explorer, you must first define a new configuration.

This section contains the following topic:

- Section 2.3.1.1, "Defining a New Configuration for BSE"

2.3.1.1 Defining a New Configuration for BSE

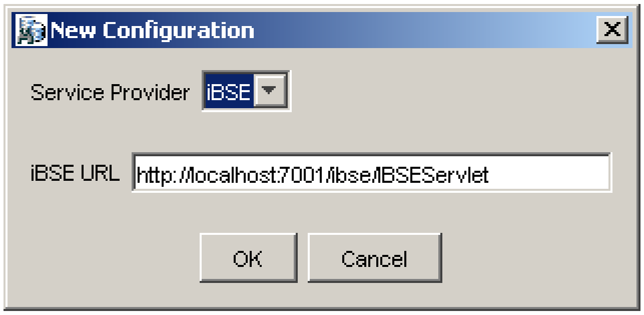
To create a new configuration for BSE:

1. Start the Application Explorer.
2. Right-click **Configurations** and select **New**.

   The New Configuration dialog is displayed.
3. Enter a name for the new configuration (for example, SampleConfig) and click **OK**.

   The New Configuration dialog is displayed, as shown in Figure 2–1.

Figure 2–1 New Configuration Dialog



Service Provider iBSE

iBSE URL http://localhost:7001/ibse/IBSEServlet

4. From the **Service Provider** list, select **iBSE**.

5. In the **iBSE URL** field, accept the default URL or replace it with a different URL using the following format:

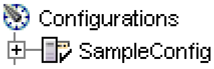
```
http://host name:port/ibse/IBSEServlet
```

   Where *host name* is the system where your Oracle WebLogic Server resides and *port* is the HTTP port number on which the Oracle WebLogic Server is listening.

6. Click **OK**.

   As shown in Figure 2–2, a node representing the new configuration appears beneath the root Configurations node.

Figure 2–2 SampleConfig Node



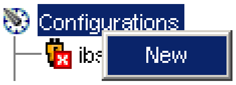
2.3.2 Creating a Configuration for J2CA

To create a configuration for Oracle Adapter J2CA using Application Explorer, you must first define a new configuration.

To define a new configuration for J2CA:

1. Start the Application Explorer.
2. Right-click **Configurations** and select New, as shown in Figure 2–3.

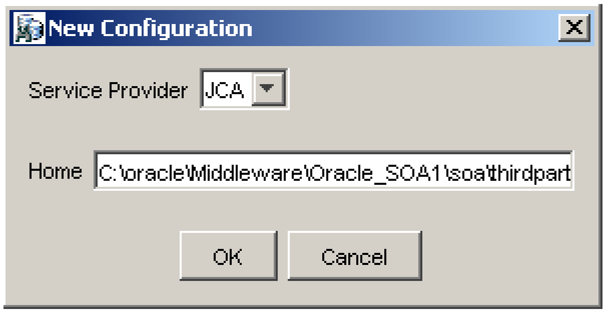
Figure 2–3 Configurations Node



   The New Configuration dialog is displayed, as shown in Figure 2–4.

3. Enter a name for the new configuration (for example, SampleConfig) and click **OK**.

Figure 2–4 New Configuration Dialog



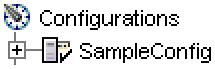
Service Provider JCA

Home C:\oracle\Middleware\Oracle_SOA1\soa\thirdpart

4. From the **Service Provider** list, select JCA.
5. Click **OK**.

   As shown in Figure 2–5, a node representing the new configuration appears beneath the root Configurations node.

Figure 2–5 SampleConfig Node



The Oracle Adapter J2CA configuration folder is stored in a location based on your adapter installation:

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

The `configuration_name` is the name of the configuration you created (for example, SampleConfig).

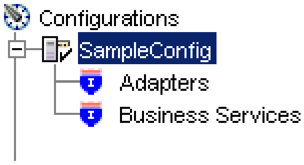
2.3.3 Connecting to a BSE or J2CA Configuration

To connect to a new configuration:

1. Right-click the configuration to which you want to connect, for example, SampleConfig.
2. Select **Connect**.

   Nodes appear for Adapters, Events, and Business Services (also known as Web services). The Business Services node is only available for BSE configurations. If you are connected to a J2CA configuration, then the Business Services node is not shown. As shown in Figure 2–6, the following is an example of a BSE configuration named SampleConfig:

Figure 2–6 The New SampleConfig Configuration That Appears Under The Configurations Node



- Use the **Adapters** node to create inbound interaction with Siebel. For example, you use the Siebel node in the Adapters node to configure a service that updates Siebel.
- Use the **Events** node (available for J2CA configurations only) to configure listeners that listen for events in Siebel.
- Use the **Business Services** node (available for BSE configurations only) to test Web services created in the Adapters node. You can also control security settings for the Web services by using the security features of the Business Services node.

You can now define new targets to Siebel.

2.4 Establishing a Connection (Target) for Siebel

To browse the Siebel Business Services, Business Components, and Integration Objects, you must define a target to Siebel. After you define the target, the parameters are automatically saved.

This section contains the following topics:

- Section 2.4.1, "Defining a Target to Siebel"
- Section 2.4.2, "Connecting to a Defined Target"

- Section 2.4.3, "Disconnecting From Siebel"
- Section 2.4.4, "Editing a Target"
- Section 2.4.5, "Deleting a Target to Siebel"

Important (All UNIX Platforms): Before you attempt to connect to a Siebel target using a BSE or J2CA configuration in a UNIX environment, you must perform the additional steps described in "Adding Required Encoding Option (All UNIX Platforms)" on page 2-6. Failure to add the encoding option as described in this section results in an error and you are not able to connect to the Siebel target. The error message may indicate that the encoding is not supported, for example:

```
Error: Problem activating adapter -- UTF-8 is not supported. Check logs for more
information.

Error: Error getting target [Siebel] -- UTF-8 is not supported.
```

Adding Required Encoding Option (All UNIX Platforms)

Before attempting to connect to a Siebel target, perform the following steps:

1. Add the following Java file encoding option to the **startWebLogic.sh** file:

   ```
   JAVA_OPTIONS="${SAVE_JAVA_OPTIONS} -Dfile.encoding=ISO8859_1"
   ```

   The **startWebLogic.sh** file is located in the following directory:

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

2.4.1 Defining a Target to Siebel

The connection parameters required for defining a Siebel target can be obtained from the eapps.cfg file, which is located in the following directory:

```
drive:\SiebelRoot\SWEApp\BIN
```

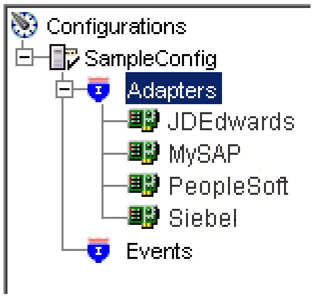
Where Siebelroot is the Siebel installation directory.

When you are working with a J2CA configuration, creating, updating, and deleting a target requires you to restart the Oracle WebLogic Server. In addition, make sure to close Application Explorer before you restart the Oracle WebLogic Server.

To define a target to Siebel:

1. In the left pane, expand the Adapters node, as shown in Figure 2–7.

Figure 2–7 Adapters Node



2. Right-click the **Siebel** node and select **Add Target**.

   The Add Target dialog is displayed. Provide the following information:

   a. In the Name field, enter a name for the new target.

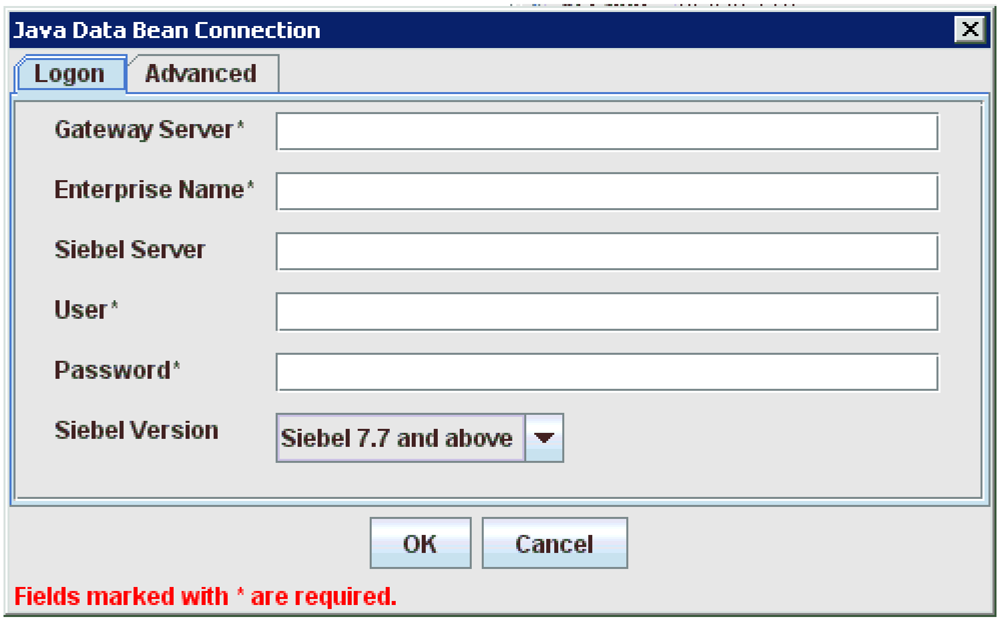
   b. In the Description field, enter a description (optional).

   c. From the Target Type list, select **Java Bean Data Connection** (default).

3. Click **OK**.

   The Java Data Bean Connection dialog is displayed, as shown in Figure 2–8.

Figure 2–8 Java Data Bean Connection Dialog



Siebel Version: Siebel 7.7 and above

Fields marked with * are required.

   Enter the system information as specified in the following steps:

   a. In the **Gateway Server** field, enter the name of the server. To specify a Gateway Server that uses a port other than the default (usually, 2320), add a colon and the port number, for example, *`gateway name`*:*`port number`*.

   b. In the **Enterprise** Name field, enter the appropriate name.

   c. In the **Siebel Server** field, enter the name of your Siebel server. Do not supply a value in this field when connecting to a Siebel 7.7, 7.8, or 8 system.

   d. In the **User** field, enter the user name.

   e. In the **Password** field, enter the password associated with the user name.

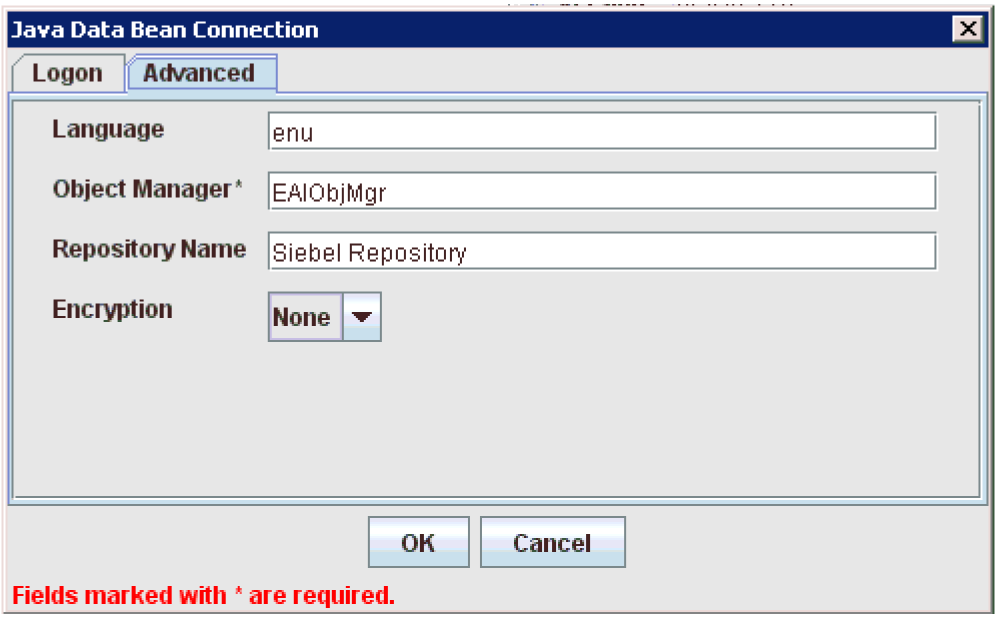
   f. From the Siebel Version list, select **Siebel 7.7 and above** (default) or **Siebel 7.5 and below**.

   g. Click the **Advanced** tab, as shown in Figure 2–9 and verify the following:

      Language

      Object Manager

Figure 2–9 Java Data Bean Connection Dialog Advanced Tab



Language: enu

Object Manager*: EAIObjMgr

Repository Name: Siebel Repository

Encryption: None

Fields marked with * are required.

Object Manager

For Siebel 7.0.3, the default Object Manager is EAIObjMgr. For Siebel 7.7, the default is EAIObjMgr_enu. Siebel 7.7 requires that you add a language extension (for example, _enu) to the end of the Object Manager name. Check with your Siebel Administrator for the specific names that apply to your system.

Repository Name

If no repository is specified, then a full list of objects from all available repositories is returned. If a specified repository is not found, then an empty list of objects is returned.

The configuration parameters supplied are those used by Siebel client applications to connect to the Siebel system. For more information about these parameters, see your Siebel documentation or ask your Siebel system administrator.

Encryption

A new parameter named Encryption is now introduced to the Advanced tab when using the Siebel adapter to create a target during design time. This parameter has two values, None and RSA. The default value is None, where no encryption is performed. By choosing RSA, an RSA-encrypted connection to the object manager specified is established.

To use RSA encryption, the Object Manager must be specified as **SCCObjMgr_enu**.

Note: These parameters are typically found in Siebel configuration files stored under the Siebel server `root/bin/<language>` directory, where `language` is the Siebel code for the language you installed (`enu` for U.S English). For example, for Siebel versions 7 and higher on a Windows platform, for the Siebel Call Center module, these values can be found in the `uagent.cfg` file. Consult your Siebel administrator and your Siebel bookshelf documentation for more information.

---

4. Click **OK**.

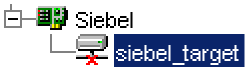
   In the left pane, the target you create appears under the Siebel node.

2.4.2 Connecting to a Defined Target

To connect to a defined target:

1. Expand the **Siebel** node and click the target name to which you want to connect, as shown in Figure 2–10.

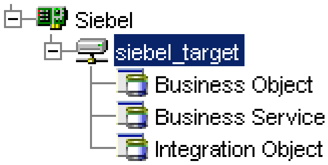
Figure 2–10 Disconnected Siebel Target



2. In the left pane, right-click the target name and select **Connect**.

   The target icon changes, indicating that you are connected to the Siebel system, as shown in Figure 2–11.

Figure 2–11 Siebel Target Node



You can now browse the available Business Objects, Business Services, and Integration Objects in the Siebel system.

2.4.3 Disconnecting From Siebel

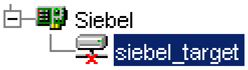
Although you can maintain multiple open connections to different application systems, it is good practice to close connections when not in use.

To disconnect from Siebel:

1. In the left pane, select the target to which you are connected.
2. Right-click the target and select **Disconnect**.

   Disconnecting from the application system drops the target, but the node remains. The SiebelConnection node in the left pane changes to reflect that the target is disconnected, as shown in Figure 2–12.

Figure 2–12 Disconnected Siebel Target

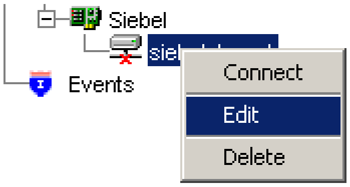


2.4.4 Editing a Target

To edit a target:

1. In the left pane, ensure the target you want to edit is disconnected.
2. Right-click the disconnected target and select **Edit**, as shown in Figure 2–13.

Figure 2–13 Edit Option



   The Edit pane is displayed on the right.

3. Modify the target information.
4. Click **OK**.

2.4.5 Deleting a Target to Siebel

You can delete a target, rather than just disconnecting and closing it. When you delete the target, the node disappears from the list of Siebel targets in the left pane of Application Explorer.

When you delete a target, you must restart the Oracle WebLogic Server to update the repository for run time purposes.

To delete a target:

1. In the left pane, select the target.
2. Right-click the target and select **Delete**.

   A confirmation message is displayed.

3. Click **OK** to delete the target you selected.

   The Siebel connection node disappears from the left pane.

2.5 Viewing Application System Objects

Application Explorer gives you the flexibility to view all Siebel application system objects. One benefit of this flexibility is that you can gain an understanding of the Siebel data structure. You can review parameters, data types, and other attributes of the Siebel data in the right pane.

This section contains the following topic:

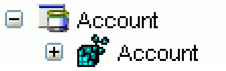
- Section 2.5.1, "Viewing Metadata"

2.5.1 Viewing Metadata

To view metadata:

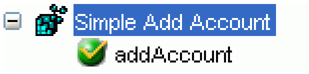
1. Start Application Explorer and connect to your Siebel system.
2. In the left pane, expand the **Business Object** or **Business Service** containing the component for which you want to generate schema.
3. Expand the **Business Object** or **Business Service** node.
4. Expand the **Business Component** or the **Business Service** node to view the objects under it.
   - For a **Business Component**, select the node in which you are interested, for example, Account, as shown in Figure 2–14.

   Figure 2–14 Account Node

   - For a **Siebel Business Service** ,select the object in which you are interested, for example, **addAccount**, as shown in Figure 2–15.

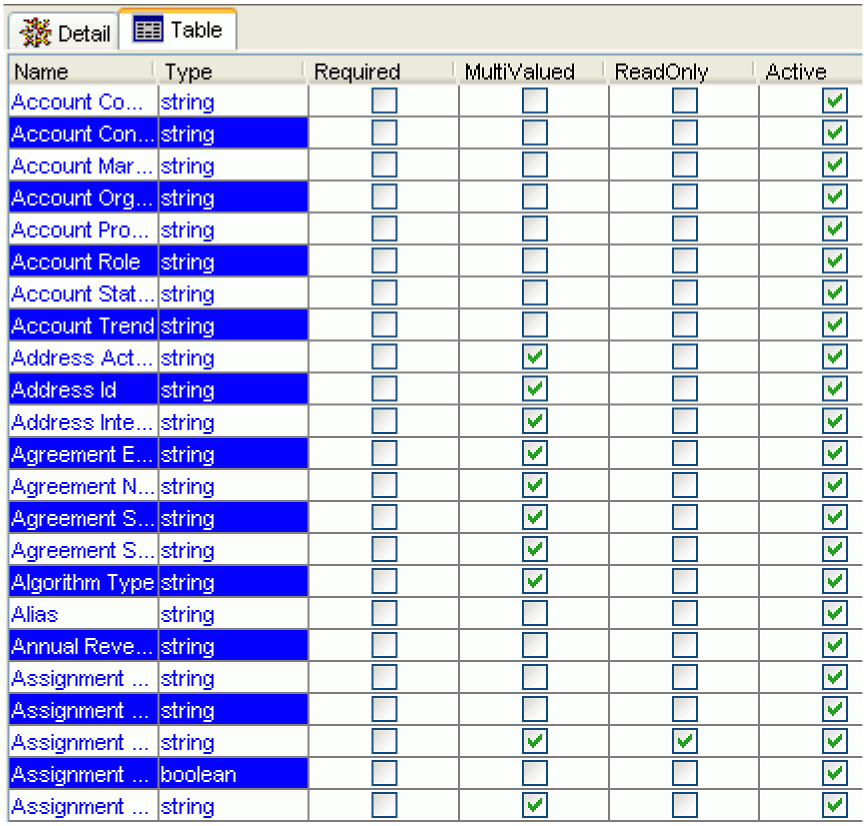
   Figure 2–15 Simple Add Account Node

5. In the right pane, click the ellipsis (...) in the Table row of the properties table.

   The metadata table appears in the right pane, as shown in Figure 2–16.

Figure 2–16 Metadata Table for the Siebel Object



Name	Type	Required	MultiValued	ReadOnly	Active
Account Co...	string	□	□	□	☑
Account Con...	string	□	□	□	☑
Account Mar...	string	□	□	□	☑
Account Org...	string	□	□	□	☑
Account Pro...	string	□	□	□	☑
Account Role	string	□	□	□	☑
Account Stat...	string	□	□	□	☑
Account Trend	string	□	□	□	☑
Address Act...	string	□	☑	□	☑
Address Id	string	□	☑	□	☑
Address Inte...	string	□	☑	□	☑
Agreement E...	string	□	☑	□	☑
Agreement N...	string	□	☑	□	☑
Agreement S...	string	□	☑	□	☑
Agreement S...	string	□	☑	□	☑
Algorithm Type	string	□	☑	□	☑
Alias	string	□	□	□	☑
Annual Reve...	string	□	□	□	☑
Assignment ...	string	□	□	□	☑
Assignment ...	string	□	□	□	☑
Assignment ...	string	□	☑	☑	☑
Assignment ...	boolean	□	□	□	☑
Assignment ...	string	□	☑	□	☑

2.6 Creating XML Schemas

You can create service schemas for Business Services and Business Components using Application Explorer.

This section contains the following topics:

- Section 2.6.1, "Siebel Schema Considerations"
- Section 2.6.2, "Creating an XML Schema for a Siebel Business Object or Business Service"
- Section 2.6.3, "Creating an XML Schema for a Siebel Business Component or Business Service"
- Section 2.6.4, "Searching for a Specific Siebel Object"
- Section 2.6.5, "Returning Fields in a Specified Order"
- Section 2.6.6, "Using QueryWithView"

The following topic describes how to create schemas for the adapter when you deploy Oracle Application Adapter for Siebel for use either in a J2CA environment or a Web services environment. For more information, see "Creating and Testing a Web Service (BSE Configurations Only)" on page 2-24 if you plan to deploy Oracle Application Adapter for Siebel in a Web services environment.

2.6.1 Siebel Schema Considerations

When inserting a record into Siebel, the data can be specified by the user or configured in Siebel to have default values or other system generated values. For example the Account Business Component, Currency Code, by default, has 'USD' and the system fields such as ROW_ID generated by the Siebel system when the record is inserted. The Siebel API does not provide this distinction. Therefore, the Oracle Application Adapter for Siebel can not anticipate what the required fields the user should enter are and what are the required fields that can be filled by Siebel. As a result, the adapter schemas have been modified to have all elements as optional by setting minoccurs=0 for the elements.

Hence, all users must determine which fields are mandatory through Siebel Tools and create a payload (request XML document) for Siebel services (outbound).

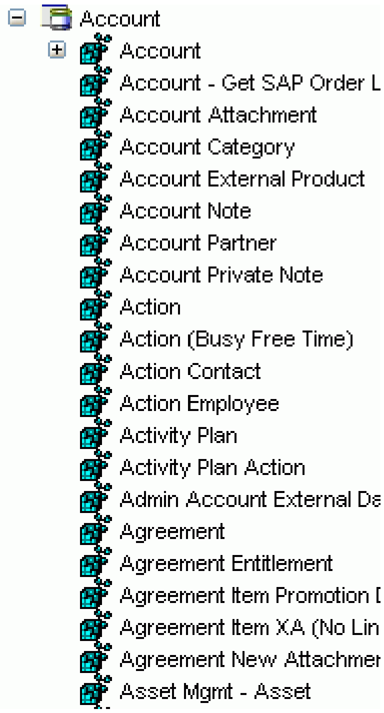
2.6.2 Creating an XML Schema for a Siebel Business Object or Business Service

You create schemas for Siebel Business Service methods (for example, the Add method) and Business Components using Application Explorer. After you create a schema, you can use it to generate service request and response schemas for the Business Service or Business Component.

Siebel Business Objects contain one or more Siebel Business Components. You can view Business Components by clicking the associated Business Object.

For example, the Account Business Object can be expanded to display all available Business Components, as shown in Figure 2–17.

Figure 2–17 Account Business Object



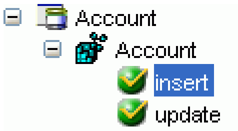
- Account
  - Account
  - Account - Get SAP Order L
  - Account Attachment
  - Account Category
  - Account External Product
  - Account Note
  - Account Partner
  - Account Private Note
  - Action
  - Action (Busy Free Time)
  - Action Contact
  - Action Employee
  - Activity Plan
  - Activity Plan Action
  - Admin Account External Da
  - Agreement
  - Agreement Entitlement
  - Agreement Item Promotion [
  - Agreement Item XA (No Lin
  - Agreement New Attachmer
  - Asset Mgmt - Asset

2.6.3 Creating an XML Schema for a Siebel Business Component or Business Service

To generate service request and response schemas for a Business Component or Business Service:

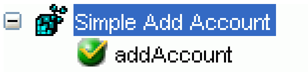
1. Start Application Explorer and connect to your Siebel system.
2. In the left pane, expand the **Business Object** or the **Business Service** node.
3. Expand the **Business Component** or **Business Service** to view the objects under it.
   - For a **Business Component**, expand the Business Object node, then expand the Business Component you want, then expand the node you want, and select the method for which you want to create a schema, as shown in Figure 2–18.

Figure 2–18 Insert Method selected Under the Account Business Object



   - For a **Siebel Business Service**, expand the **Business Service** node containing the object for which you want to create schema, as shown in Figure 2–19.

Figure 2–19 The addAccount Object Under The Add Account Business Service

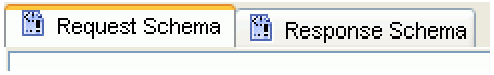


4. Right-click the node and select **Generate Schema.**

   Application Explorer accesses the Siebel repository and builds schemas.

   As shown in Figure 2–20, schema tabs similar to the following appear in the right pane.

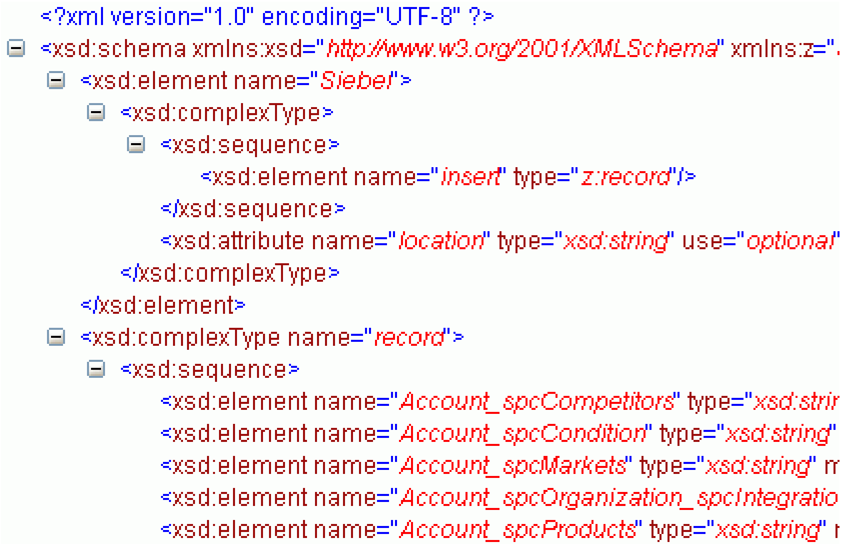
Figure 2–20 Request and Response Schema Tabs



5. To view a schema, click the ellipsis tab corresponding to the schema you want to view.

   The schema appears on the right, as shown in Figure 2–21.

Figure 2–21 XML Schema



```
  <?xml version="1.0" encoding="UTF-8" ?>
⊟ <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:z=".
  ⊟ <xsd:element name="Siebel">
    ⊟ <xsd:complexType>
      ⊟ <xsd:sequence>
           <xsd:element name="insert" type="z:record"/>
        </xsd:sequence>
        <xsd:attribute name="location" type="xsd:string" use="optional"
      </xsd:complexType>
    </xsd:element>
  ⊟ <xsd:complexType name="record">
    ⊟ <xsd:sequence>
        <xsd:element name="Account_spcCompetitors" type="xsd:strir
        <xsd:element name="Account_spcCondition" type="xsd:string"
        <xsd:element name="Account_spcMarkets" type="xsd:string" m
        <xsd:element name="Account_spcOrganization_spcIntegratio
        <xsd:element name="Account_spcProducts" type="xsd:string" r
```

2.6.4 Searching for a Specific Siebel Object

You can use the search function in Application Explorer to locate a Siebel object or node quickly.

1. Start Application Explorer and connect to your Siebel system through a target.
2. Expand the target and select **Business Object**, **Business Service**, or **Integration Object**.
3. In the right pane, move the cursor over Operations and select **Search**.
4. Enter the name of the node or object on which you want to search in the text entry box, for example, **Account**.
5. Click **OK**.

   A list containing the Siebel items that match your search appears.
6. Select the item in which you are interested.

   Application Explorer locates the item in which you are interested.

2.6.5 Returning Fields in a Specified Order

When you create a request document from an XML schema to query the Siebel system, you can limit the expected response to specific fields that are specified in the query. The response contains the fields in the order in which they were specified. If you do not specify a set of fields, then the response document contains the entire set.

For example, the following query returns all fields:

```
<m:Siebel location="S/BO/Account/Account/queryWithView" view="AllView">
  <m:select>
   <m:Name>Yelena*</m:Name>
  </m:select>
</m:Siebel>
```

The following query returns a response that only contains the fields Name, Location and Account Status fields:

```
<m:Siebel location="S/BO/Account/Account/queryWithView" view="AllView">
```

```
  <m:select>
    <m:Name>Yelena*</m:Name>
  </m:select>
  <m:field>Name</m:field>
  <m:field>Location</m:field>
  <m:field>Account Status</m:field>
</m:Siebel>
```

2.6.6 Using QueryWithView

For Business Components, the Oracle Application Adapter for Siebel enables Insert, Update, Delete, and Query. It also enables a method called QueryWithView. The View modes are a visibility feature provided by Siebel.

By using QueryWithView, you can specify a Siebel View mode as a parameter. The API parameters allow different presentations of data depending on the Siebel environment that you configured.

You can use Query except when you want to enable a user to retrieve records based on different view modes. In this case, use QueryWithView. For more information on QueryWithView mode or Siebel "Visibility" concepts, see your Siebel Administrator.

The following levels are available:

- Sales Rep View
- Manager View
- Personal View
- All View
- Organization View
- Group View
- Catalog View
- SubOrganization View

2.7 Siebel Prerequisites for Working With Integration Objects

To create XML schemas for Siebel Integration Objects, you may have to generate XDR schemas first, using the Siebel Tools Schema Wizard.

The XDR schema is used as input to Application Explorer when generating schemas for integration objects. After you generate the XDR schema, Application Explorer uses the XDR file to generate the XML schema.

Please note:

- For **Siebel 7.5 and later**: Generate XSD schemas directly from Siebel tools. These XSD schemas are used to create Web services directly using Application Explorer. After you generate an XSD schema through Siebel tools, use it to create an IO node and Web service.
- For **Siebel 7.0**: You cannot generate XSD schemas directly from Siebel tools; only XDR schemas can be created. Therefore, to create a Web service, Application Explorer must first generate an XSD schema from the XDR schema.
- For releases **before Siebel 6.3**: The Siebel Tools Schema Wizard creates only DTD schemas. You must transform these schemas manually, or by using other tools, into XDR files before Application Explorer can use them as input to create XML

schemas. In addition, you must include the SiebelMessage tag reference in your XDR file.

Oracle Application Adapter for Siebel supports access to Siebel Integration Objects by using Siebel XML to handle events. Using Siebel Integration Objects through supported transports requires Siebel workflows.

2.8 Creating Schemas for Siebel Integration Objects

This section describes how to create schemas for Siebel Integration Objects and contains the following topic:

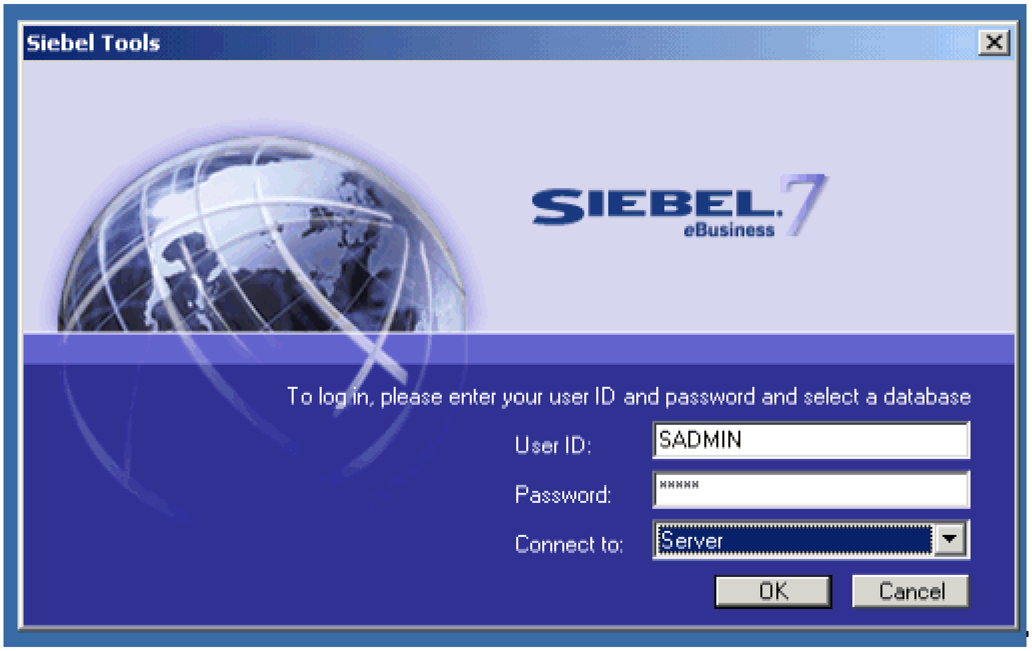
- Section 2.8.1, "Creating a Siebel XDR or XSD Schema for a Siebel Integration Object"

2.8.1 Creating a Siebel XDR or XSD Schema for a Siebel Integration Object

To generate a Siebel XDR or XSD schema:

1. Log on to Siebel Tools, as shown in Figure 2–22.

Figure 2–22 Siebel Tools Menu



Siebel Tools

To log in, please enter your user ID and password and select a database

User ID: SADMIN

Password: *****

Connect to: Server

Perform the following steps:

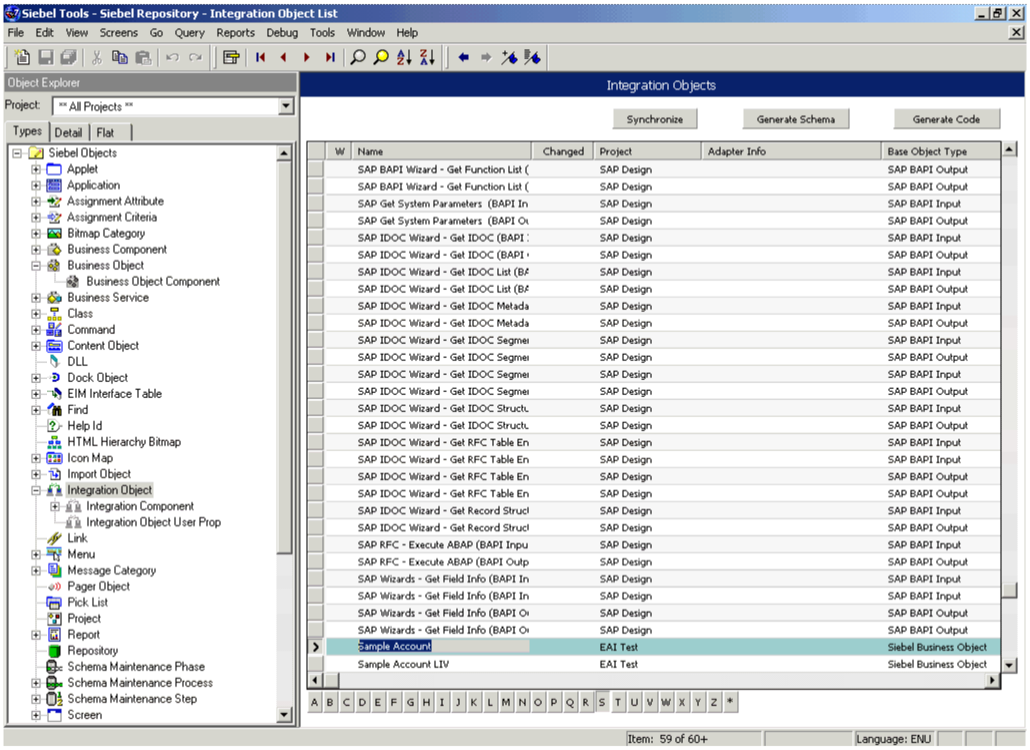
a. Enter your user ID and password.

b. Select a database from the list.

2. Click **OK**.

The Siebel Tools window is displayed, as shown in Figure 2–23. Integration Objects appear in the right pane.

Figure 2–23 Siebel Tools Window



Siebel Tools - Siebel Repository - Integration Object List
File Edit View Screens Go Query Reports Debug Tools Window Help
Object Explorer
Integration Objects
Synchronize
Generate Schema
Generate Code
Types
Detail
Flat
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z *

Project: ** All Projects **

- Siebel Objects
  - Applet
  - Application
  - Assignment Attribute
  - Assignment Criteria
  - Bitmap Category
  - Business Component
  - Business Object
    - Business Object Component
  - Business Service
  - Class
  - Command
  - Content Object
  - DLL
  - Dock Object
  - EIM Interface Table
  - Find
  - Help Id
  - HTML Hierarchy Bitmap
  - Icon Map
  - Import Object
  - Integration Object
    - Integration Component
    - Integration Object User Prop
  - Link
  - Menu
  - Message Category
  - Pager Object
  - Pick List
  - Project
  - Report
  - Repository
  - Schema Maintenance Phase
  - Schema Maintenance Process
  - Schema Maintenance Step
  - Screen

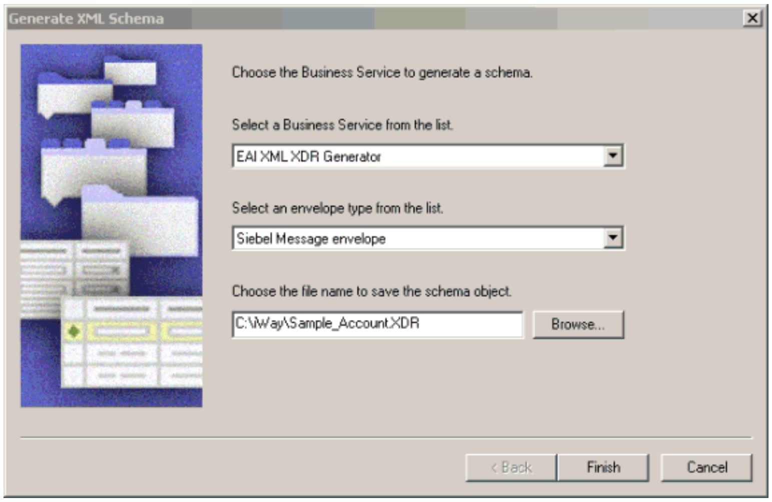
W	Name	Changed	Project	Adapter Info	Base Object Type
	SAP BAPI Wizard - Get Function List (		SAP Design		SAP BAPI Output
	SAP BAPI Wizard - Get Function List (		SAP Design		SAP BAPI Output
	SAP Get System Parameters (BAPI In		SAP Design		SAP BAPI Input
	SAP Get System Parameters (BAPI Ou		SAP Design		SAP BAPI Output
	SAP IDOC Wizard - Get IDOC (BAPI :		SAP Design		SAP BAPI Input
	SAP IDOC Wizard - Get IDOC (BAPI (		SAP Design		SAP BAPI Output
	SAP IDOC Wizard - Get IDOC List (BA		SAP Design		SAP BAPI Input
	SAP IDOC Wizard - Get IDOC List (BA		SAP Design		SAP BAPI Output
	SAP IDOC Wizard - Get IDOC Metada		SAP Design		SAP BAPI Input
	SAP IDOC Wizard - Get IDOC Metada		SAP Design		SAP BAPI Output
	SAP IDOC Wizard - Get IDOC Segmei		SAP Design		SAP BAPI Input
	SAP IDOC Wizard - Get IDOC Segmei		SAP Design		SAP BAPI Output
	SAP IDOC Wizard - Get IDOC Segmei		SAP Design		SAP BAPI Input
	SAP IDOC Wizard - Get IDOC Segmei		SAP Design		SAP BAPI Output
	SAP IDOC Wizard - Get IDOC Structu		SAP Design		SAP BAPI Input
	SAP IDOC Wizard - Get IDOC Structu		SAP Design		SAP BAPI Output
	SAP IDOC Wizard - Get RFC Table En		SAP Design		SAP BAPI Input
	SAP IDOC Wizard - Get RFC Table En		SAP Design		SAP BAPI Input
	SAP IDOC Wizard - Get RFC Table En		SAP Design		SAP BAPI Output
	SAP IDOC Wizard - Get RFC Table En		SAP Design		SAP BAPI Output
	SAP IDOC Wizard - Get Record Struct		SAP Design		SAP BAPI Input
	SAP IDOC Wizard - Get Record Struct		SAP Design		SAP BAPI Output
	SAP RFC - Execute ABAP (BAPI Inpu		SAP Design		SAP BAPI Input
	SAP RFC - Execute ABAP (BAPI Outp		SAP Design		SAP BAPI Output
	SAP Wizards - Get Field Info (BAPI In		SAP Design		SAP BAPI Input
	SAP Wizards - Get Field Info (BAPI In		SAP Design		SAP BAPI Input
	SAP Wizards - Get Field Info (BAPI Ou		SAP Design		SAP BAPI Output
	SAP Wizards - Get Field Info (BAPI Ou		SAP Design		SAP BAPI Output
	Sample Account		EAI Test		Siebel Business Object
	Sample Account LIV		EAI Test		Siebel Business Object

Item: 59 of 60+ Language: ENU

3. To create a schema, select an Integration Object, for example, Sample Account.
4. Click **Generate Schema**.

   The Generate XML Schema wizard is displayed, as shown in Figure 2–24.

Figure 2–24 Generate XML Schema Wizard



Choose the Business Service to generate a schema.

Select a Business Service from the list.
EAI XML XDR Generator

Select an envelope type from the list.
Siebel Message envelope

Choose the file name to save the schema object.
C:\iWay\Sample_Account.XDR

Perform the following steps:

a. From the Select a Business Service list, select **EAI XML XDR Generator** for XDR schemas or EAI XML XSD Generator for XSD schemas (for Siebel 7.5 and later).

b. From the Select an envelope type list, select **Siebel Message envelope**.

c. In the Choose the file name field, specify a file name for the XDR schema and a directory where it can be accessed by Application Explorer.

> **Note:** The XDR or XSD schema file must be saved to a directory on the same computer as Application Explorer.

5. Click **Finish**.
6. Create a workflow to accept incoming XML documents through HTTP and to insert/update Siebel data by using the EAI XML Converter and EAI Siebel Adapter Business Services.

   For more information, see Appendix A, "Using Siebel Workflows".
7. Edit the `eai.cfg` file, which is located in the following directory:

   ```
   <siebel_server>/bin/enu
   ```
8. Add the following line to the `[HTTP Services]` section:

   ```
   [HTTP Services]
   wf = iWayWorkflow
   ```
9. Confirm that the following line is set in the `[EAI_ENU]` section of the `Eapps.cfg` file:

   ```
   [EAI_ENU]
   EnableExtServiceOnly = True
   ```

   The `Eapps.cfg` file is located in the following directory:

   ```
   <siebel_server>/bin
   ```
10. Create a named subsystem using Siebel Server Manager by running the following command, where EAITEST is the name of the workflow that was created in step 6:

    ```
    create named subsystem iWAyWorkflow for subsystem
    EAITransportDataHandlingSubsys with DispatchWorkflowProcess="EAITEST"
    ```

    Now you can use Application Explorer to create Integration Object (IO) nodes for Siebel.

2.9 Creating Integration Object (IO) Nodes for Siebel

This section contains the following topic:

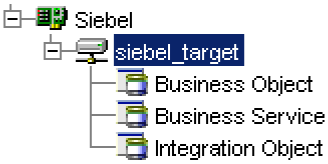
- Section 2.9.1, "Creating an XML Schema for a Siebel Integration Object"

To create an Integration Object node for Siebel, perform the following steps:

1. In Application Explorer, connect to a defined target. For more information on how to connect to a target, see "Connecting to a Defined Target" on page 2-9.

   The X over the icon disappears, indicating that the node target is connected, as shown in Figure 2–25.

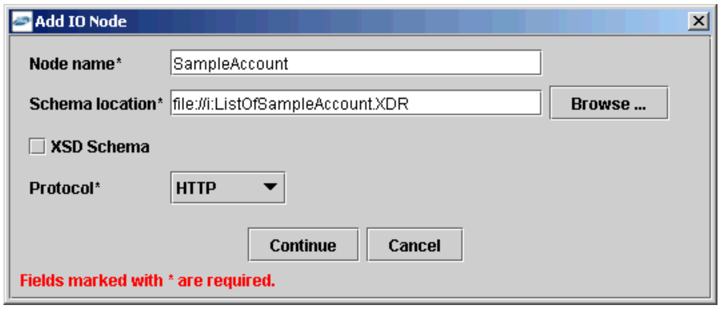
Figure 2–25 Expanded Siebel Node



2. Expand the Integration Object node and select Sample Account.
3. Right-click the **Sample Account** node and select **Add IO Node**.

   The Add IO Node dialog is displayed, as shown in Figure 2–26.

Figure 2–26 Add IO Node Dialog



Add IO Node

Node name* SampleAccount

Schema location* file://i:ListOfSampleAccount.XDR

□ XSD Schema

Protocol* HTTP

Fields marked with * are required.

Please note:

- **For Siebel 7.5 or later:** Generate XSD schemas directly from Siebel tools. You use the XSD schemas when you create Web services in Application Explorer. After you generate an XSD schema through Siebel tools, use it to create an IO node and a Web service.
- **For Siebel 7.0:** You cannot generate XSD schemas directly from Siebel tools; only XDR schemas can be created. Before you create a Web service, you must first generate an XSD schema from the XDR schema using Application Explorer.

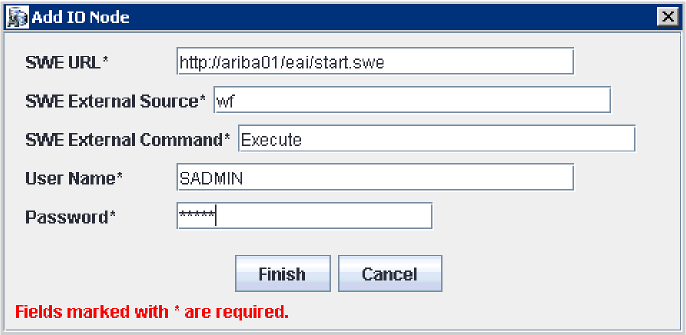
Note: This is the schema file that you generated in Creating Schemas for Siebel Integration Objects on page 2-17.

---

4. Enter a node name, for example SampleAccount in the **Node name** field and a path to the Sample Account XDR or XSD file in the **Schema location** field.
5. If the XSD schema has already been generated, then select XSD Schema. If you are using Siebel-generated XDR schemas, then do not select the XSD schema option.
6. Select a protocol from the **Protocol** list.
7. Click **Continue**.

   The Add IO Node dialog is displayed, as shown in Figure 2–27.

Figure 2–27 Add IO Node Dialog



SWE URL* http://ariba01/eai/start.swe

SWE External Source* wf

SWE External Command* Execute

User Name* SADMIN

Password* *****

Fields marked with * are required.

8. Perform the following steps:

   a. In the SWE URL field, type the Base SWE URL. For example:

      ```
      http://web_server/eai/start.swe
      ```

      Where **web_server** is the name of the Web server that is hosting Siebel SWE.

   b. In the SWE External Source field, type the section within the eai.cfg file to execute, which is the [HTTP Services] section.

      For more information, see step 8 in Creating Schemas for Siebel Integration Objects on page 2-17.

   c. In the SWE External Command field, type the following command exactly as shown:

      ```
      Execute
      ```

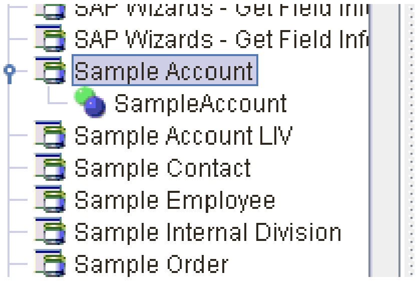
   d. In the User Name and Password fields, type a valid user name and password used to connect to the Siebel SWE.

      The user name and password must have privileges to execute the given workflow.

9. Click **Finish**.

   The new IO node is listed under the Integration Object's Sample Account node, as shown in Figure 2–28.

Figure 2–28 Integration Object's Sample Account Node



You can now create an XML schema.

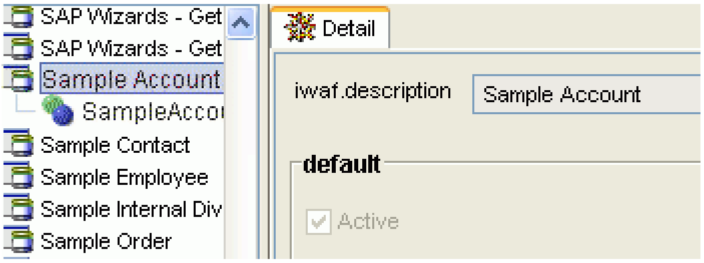
2.9.1 Creating an XML Schema for a Siebel Integration Object

After you create an Integration Object node for Siebel, you can create an XML schema using Application Explorer.

To create an XML schema:

1. In Application Explorer, expand the **Integration Objects** node to browse the Integration Objects in the Siebel system, as shown in Figure 2–29.

Figure 2–29 Siebel Integration Objects Node, Sample Account

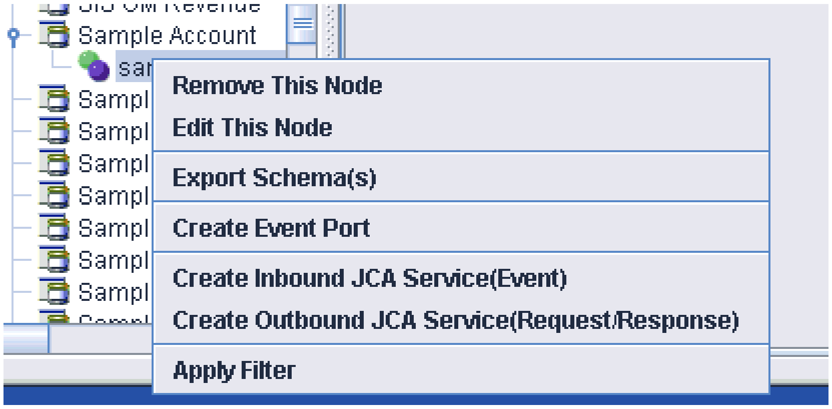


iwaf.description: Sample Account

2. Scroll down and select an Integration Object (for example, SampleAccount).

3. Right-click the created Integration Object node (for example, SampleAccount) and select **Export Schema(s)** from the menu, as shown in Figure 2–30.

Figure 2–30 Export Schema(s) Menu Option



The Select Export Directory dialog is displayed.

The exported event schema must be specified during the channel creation process in the PreParser tab (Schema location field).

4. Click **OK** to save the Schemas.

2.10 Creating a Service Node for a Siebel Business Service

OracleAS Adapter for Siebel enables the addition of a service node for a Business Service that includes methods containing method arguments having hierarchy data types.

Important limitations:

- The adapter supports only Integration Object hierarchy data types.
- Adding a Service node requires that you have previously generated an XSD schema for the Integration Object. For more information on generating XSD schemas for Siebel Integration Objects, see "Creating Schemas for Siebel Integration Objects" on page 2-17.

- Only one of the method arguments for the Business Service method for which you want to add a service node can be a hierarchical data type.
- The method argument `XMLCharEncoding` is not supported. Leave this element blank in the XML payload. If you enter a valid `XMLCharEncoding` value such as UTF-8 or UTF-16, then the following error is received:

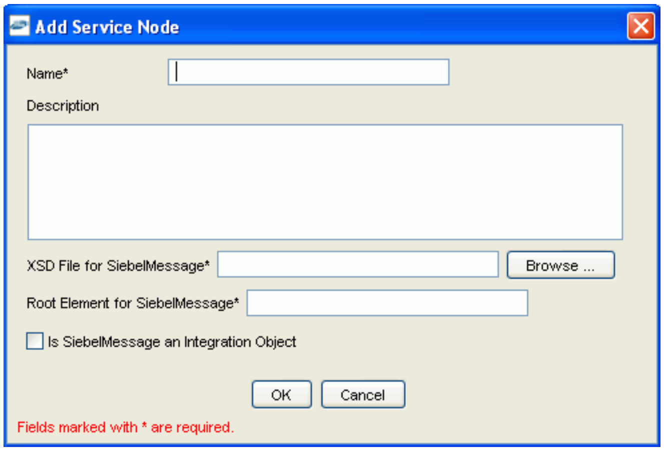
  ```
  Invocation of Service failed.
  ```

To create the service:

1. Select the Business Service node in which you are interested.
2. Right-click the Business Service method argument for which you want to create a service and select **Add Service Node**.

   The Add Service Node dialog is displayed, as shown in Figure 2–31.

Figure 2–31 Add Service Node Dialog



Add Service Node

Name*

Description

XSD File for SiebelMessage*

Root Element for SiebelMessage*

□ Is SiebelMessage an Integration Object

Fields marked with * are required.

3. Perform the following steps:

   a. Provide a service node name.

   b. Enter a description (optional).

   c. Provide the full path (including the file name) to the XSD schema file.

   d. Specify the root element for the XSD schema file. For many XSD schemas for Integration Objects, the root element is SiebelMessage.

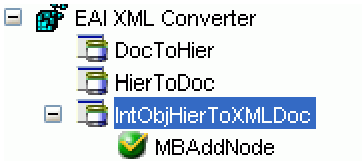
   e. Specify whether the XSD schema is for an Integration Object.

      Important: You must verify that this check box is selected.

4. Click **OK**.

   The Service node is listed under the Business Service object, as shown in Figure 2–32.

Figure 2–32 Service Node Listed Under The Business Service Object



You can right-click this node to create a Web service. The request and response schemas are displayed in the right pane.

The following procedure describes how to create a Web service for a Business Object.

2.11 Creating and Testing a Web Service (BSE Configurations Only)

You can generate a **business service** (also known as a Web service) for Siebel objects you want to use with your adapter after you have properly configured the servlet BSE.

> **Note:** In a J2EE Connector Architecture (J2CA) implementation of adapters, Web services are not available. When the adapters are deployed to use Oracle Adapter J2CA, the Common Client Interface provides integration services using the adapters.

This section contains the following topics:

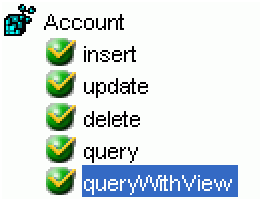
- Section 2.11.1, "Creating a Web Service"
- Section 2.11.2, "Testing a Web Service"

2.11.1 Creating a Web Service

To generate a Web service for a Siebel Business Object:

1. Connect to your Siebel system.
2. Expand a **Business Object** node.
3. Expand the **Business Component** for which you want to create a Web service, as shown in Figure 2–33.

Figure 2–33 Account Business Object with queryWithView method



4. Expand the object and select a method for creating the Web service, for example, QueryWithView under Account.
5. Right-click the node from which you want to create a business service and select **Create Business Service**.

   The Create Web Service dialog is displayed.

   You can add the business object as a method for a new Web service or as a method for an existing one. Perform the following steps:

   a. From the **Existing Service Names** list, select either <new service> or an existing service.

   b. Specify a service name if you are creating a new service. This name identifies the Web service in the list of services under the **Business Services** node.

   c. Enter a description for the service (optional).

   d. Select one of the available licenses.

6. Click **Next**.

   The License and Method dialog is displayed. Perform the following steps:

   a. In the **License** field, select one or more license codes to assign to the Web service. To select more than one, hold down the Ctrl key and click the licenses.

   b. In the **Method Name** field, leave the default method name.

   c. In the **Description** field, enter a brief description of the method (optional).

7. Click **OK**.

   Application Explorer switches the view to the **Business Services** node, and the new Web service appears in the left pane.

8. Right-click the new Web service and select **Save WSDL** from the menu.

   The Save dialog is displayed.

9. Provide a name for the WSDL file and a location to save the WSDL file on your file system.

10. Click **Save**.

2.11.2 Testing a Web Service

After you create a Web service for the Siebel Business Object, test it to ensure it functions properly. Application Explorer includes a test tool for testing a Web service.

This section contains the following topics:

- Section 2.11.2.1, "Testing a Web Service for a Business Object"
- Section 2.11.2.2, "Testing a Web Service for a Business Service"
- Section 2.11.2.3, "Identity Propagation"

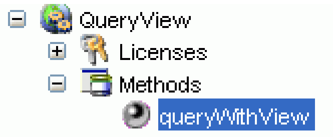
2.11.2.1 Testing a Web Service for a Business Object

1. In the left pane of Application Explorer, expand the **Business Services** node.

2. Expand the **Services** node.

3. As shown in Figure 2–34, select the name of the business service you want to test.

Figure 2–34 Expanded Service Node



4. Expand the **Methods** node under the service and select the method you want to test.

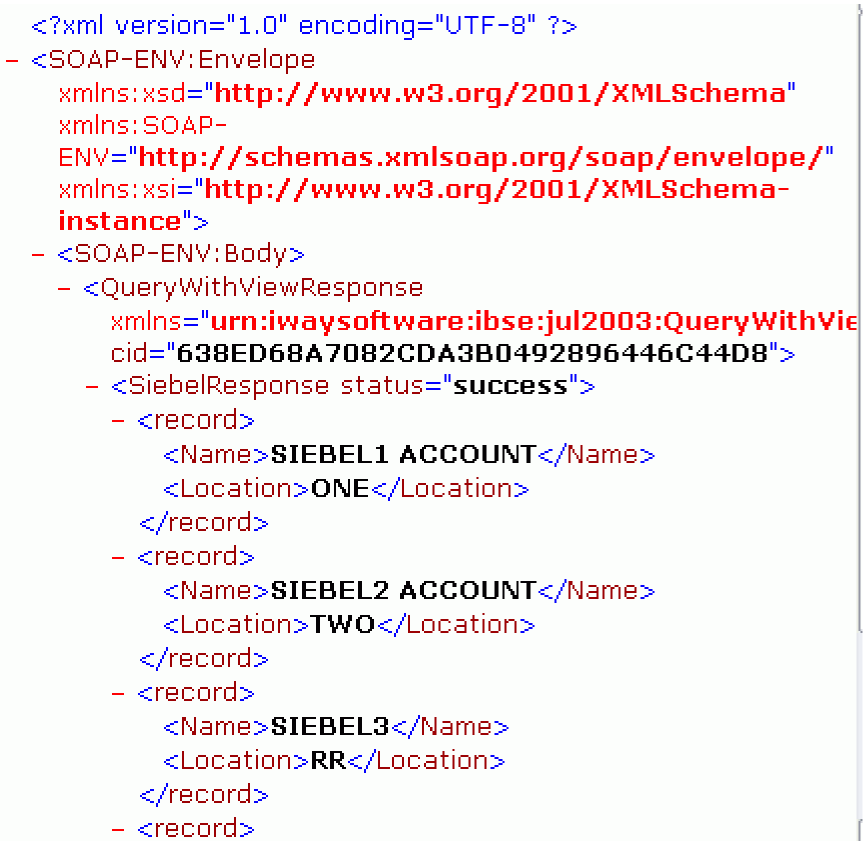
   The test option appears in the right pane.

   If you are testing a Web service that requires XML input, then an input field is displayed.

5. Click **Invoke**.

Application Explorer displays the results in the results pane, as shown in Figure 2–35.

Figure 2–35 XML Results in the Results Pane



```
  <?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>
    - <QueryWithViewResponse
        xmlns="urn:iwaysoftware:ibse:jul2003:QueryWithVie
        cid="638ED68A7082CDA3B0492896446C44D8">
      - <SiebelResponse status="success">
        - <record>
            <Name>SIEBEL1 ACCOUNT</Name>
            <Location>ONE</Location>
          </record>
        - <record>
            <Name>SIEBEL2 ACCOUNT</Name>
            <Location>TWO</Location>
          </record>
        - <record>
            <Name>SIEBEL3</Name>
            <Location>RR</Location>
          </record>
        - <record>
```

2.11.2.2 Testing a Web Service for a Business Service

After you create a Web service for the Siebel Business Service, test it to ensure it functions properly. Application Explorer includes a test tool for testing a Web service.

1. Expand the **Business Services** node.
2. Expand the **Services** node.
3. Select the name of the business service you want to test.
4. Expand the **Methods** node and select the name of the method you want to test.

   The test option appears in the right pane.

   If you are testing a Web service that requires XML input, then an input field is displayed.
5. Provide the appropriate input.
6. Click **Invoke**.

   Application Explorer displays the results in the results pane.

2.11.2.3 Identity Propagation

If you test or execute a Web service using a third party XML editor, for example XMLSPY, then the user name and password values that you specify in the SOAP header must be valid and are used to connect to Siebel. The user name and password values that you provided for Siebel during target creation using Application Explorer are overwritten for this Web service request. The following is a sample SOAP header that is included in the WSDL file for a Web service:

```
<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>
```

You can remove the `<m:disposition>` and `<m:language>` tags from the SOAP header, since they are not required.

2.12 Generating WSDL (J2CA Configurations Only)

The Web Service Definition Language (WSDL) description of a Web service enables you to make the service available to other services within a host server. You use Application Explorer to create both request-response (outbound) and event notification (inbound) JCA services of the adapter.

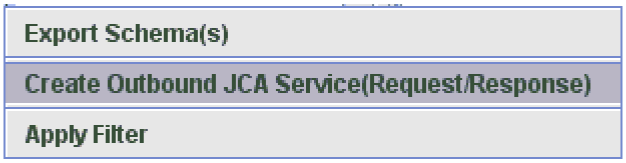
> **Note:** The **Create Inbound JCA Service (Event)** option is only available when the selected node supports events.

To generate a WSDL file for request-response service:

1. Under your connected Siebel target, expand **Business Object**, **Account**, **Account**. Navigate to an object and right-click the object.

   The following menu is displayed, as shown in Figure 2–36.

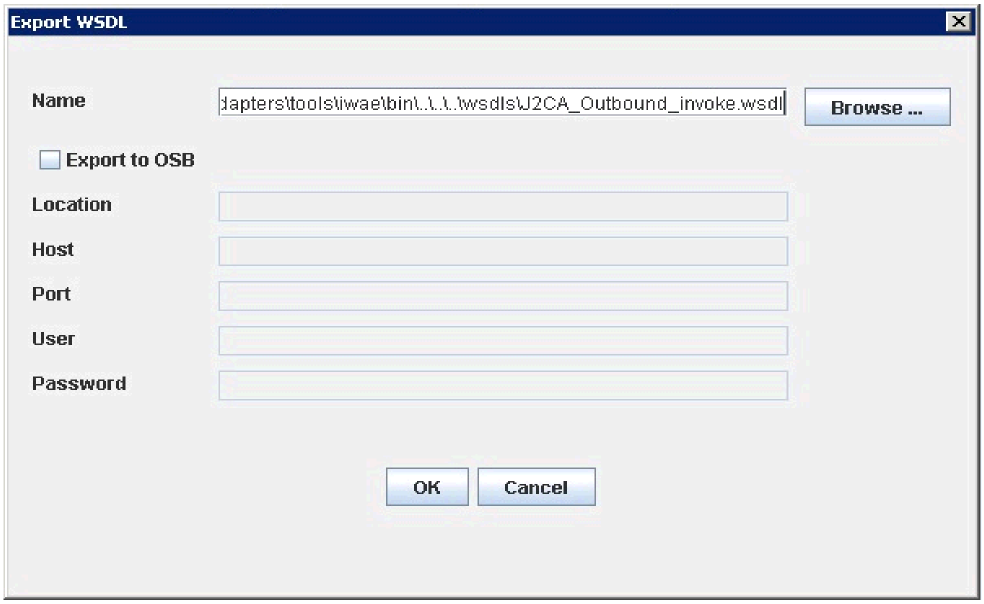
Figure 2–36 Create Outbound JCA Service (Request/Response) Option



2. Select **Create Outbound JCA Service (Request/Response)**.

   As shown in Figure 2–37, the Export WSDL dialog is displayed.

Figure 2–37 Export WSDL Dialog



Export WSDL

Name dapters\tools\iwae\bin\..\..\..\wsdls\J2CA_Outbound_invoke.wsdl

☐ Export to OSB

Location

Host

Port

User

Password

3. Accept the default name or provide a name (for example, J2CA_Outbound) for the file.

   The **.wsdl** file extension is added automatically. By default, the names of WSDL files generated for request-response services end with `_invoke`, while those generated for event notification end with `_receive`.

4. Click **OK**.

   The WSDL file is saved in the specified location.

   The procedure for generating WSDL for event notification is similar to request-response. To generate WSDL for event notification, you must first create a channel for every event.

2.13 Configuring an Event Adapter

Events are generated by a specific business condition being satisfied or triggered in the Siebel system. You can use events to trigger an action in your application. For example, an update to a database can reflect an update to customer information. If your application must perform when this happens, then your application is a consumer of this event.

This section contains the following topic:

- Section 2.13.1, "Creating and Modifying a Channel"

After you create a connection to your application system, you can add events using Application Explorer. To configure an event, you must create a channel.

Note: If you are using a J2CA configuration, then you must create a new channel for every different event object and select this channel when you generate WSDL. Creating a channel is required for J2CA configurations only. For example, if you are working with the Account and Contact Siebel objects, then two separate channels are required for this purpose.

A channel represents configured connections to particular instances of back-end systems. A channel binds one or more event ports to a particular listener managed by the adapter. For more information, see "Creating and Modifying a Channel" on page 2-29.

Please note that adding IO node functionality is not applicable in event configurations.

2.13.1 Creating and Modifying a Channel

This section contains the following topics:

- Section 2.13.1.1, "Creating an HTTP Channel"
- Section 2.13.1.2, "Creating an MQ Series Channel"
- Section 2.13.1.3, "Creating a File Channel"
- Section 2.13.1.4, "Editing a Channel"
- Section 2.13.1.5, "Deleting a Channel"

The following procedure describes how to create a channel for your event. All defined event ports must be associated with a channel.

When you create, modify, or delete a channel, you must restart the Oracle WebLogic Server to recognize the change and update the repository for run time purposes. After successfully creating the channel and inbound WSDL file, close Application Explorer before you restart the Oracle WebLogic Server.

> **Note:** If you are planning to integrate Oracle Application Adapter for Siebel with BPM, BPEL, Mediator, or OSB inbound process components, then do not start the channel. The channel is managed by the run-time server after the BPM, BPEL, Mediator, or OSB process component is deployed. If you start the channel from Application Explorer for testing and debugging purposes, then stop it before run-time (when working with BPM, BPEL, Mediator, or OSB process components).

Three channel types are available:

- HTTP
- MQ Series
- File

> **Note:** Channels can be configured only on the system where the Oracle Application Adapter for Siebel is installed.

2.13.1.1 Creating an HTTP Channel

To create an HTTP channel:

1. Click the **Events** node.

   The Events window is displayed. The adapters that appear in the left pane support events.

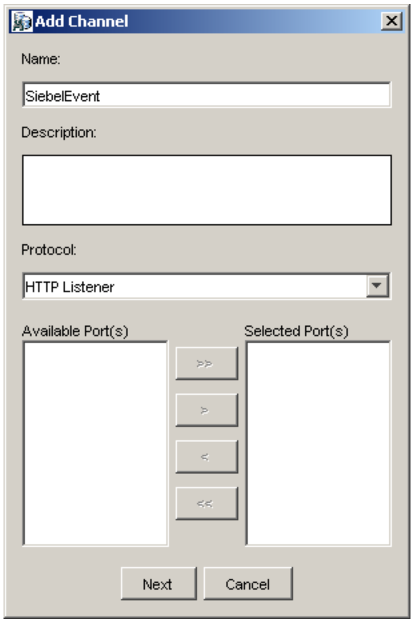
2. In the left pane, expand the **Siebel** node.

   The ports and channels nodes appear.

3. Right-click **channels** and select **Add channel**.

   The Add Channel dialog is displayed, as shown in Figure 2–38.

Figure 2–38 Add Channel DIalog



Add Channel

Name: SiebelEvent

Description:

Protocol: HTTP Listener

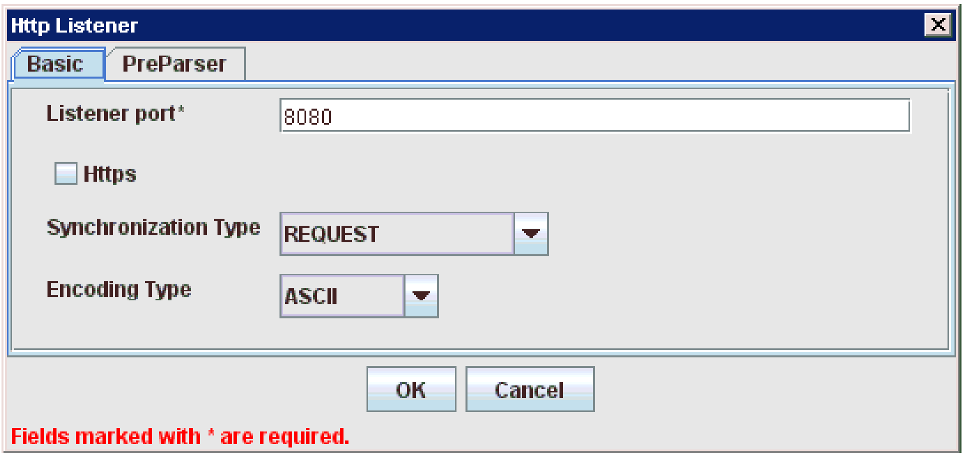
Perform the following steps:

   a. Enter a name for the channel, for example, NewChannel.
   b. Enter a brief description.
   c. From the Protocol list, select **HTTP Listener**.

4. Click **Next**.

   The Http Listener dialog is displayed, as shown in Figure 2–39.

Figure 2–39 Http Listener Dialog



Http Listener

Listener port* 8080

□ Https

Synchronization Type REQUEST

Encoding Type ASCII

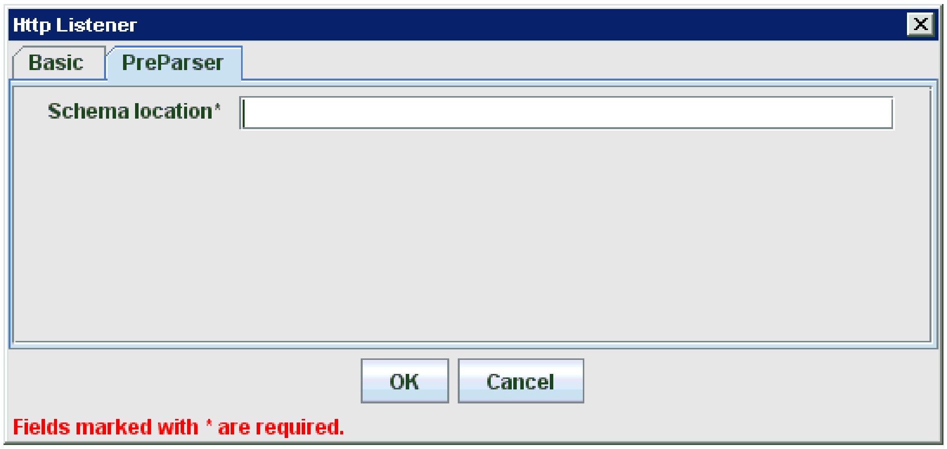
Fields marked with * are required.

5. Enter the system information as specified in the following table:

Parameter	Description
Listener port	Port on which to listen for Siebel event data.
Https	For a secure HTTP connection, select the **Https** check box. This option is currently not supported.
Synchronization Type	Select **REQUEST_RESPONSE** from the Synchronization Type list, which is the recommended option.
Encoding Type	Choose an encoding type to be used from the Encoding Type list. By default, ASCII is selected.

6. Click the **PreParser** tab, as shown in Figure 2–40.

Figure 2–40 PreParser Tab



Schema location*

Fields marked with * are required.

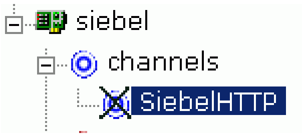
7. Specify the location of the schema file that was generated for the Integration Object node using the **Export Schema(s)** option in Application Explorer.

> **Note:** During run time, the Oracle Application Adapter for Siebel adds the namespace to the Siebel published document using the schema that is specified in the PreParser tab. If the Schema location field in the PreParser tab is left blank, then BPEL, BPM, OSB, and Mediator processes do not work properly as the Siebel published documents do not contain any namespaces.

8. Click **OK**.

   A summary is displayed, which provides the channel description, channel status, and available ports. All the information is associated with the channel you created. The channel also appears under the channels node in the left pane, as shown in Figure 2–41.

Figure 2–41 Inactive SiebelHTTP Node



An X over the icon indicates that the channel is currently disconnected. You must start the channel to activate your event configuration.

9. Right-click the channel and select **Start**.

   The channel you created becomes active. The X over the icon in the left pane disappears.

10. To stop the channel, right-click the channel and select **Stop**.

2.13.1.2 Creating an MQ Series Channel

To create an MQ Series channel:

1. Click the **Events** node.

   The Events window is displayed. The adapters that appear in the left pane support events.

2. In the left pane, expand the **Siebel** node.

   The ports and channels nodes appear.

3. Right-click the **channels** node and select **Add channel**.

   The Add a new channel pane is displayed. Perform the following steps:

   a. Enter a name for the channel, for example, NewChannel.

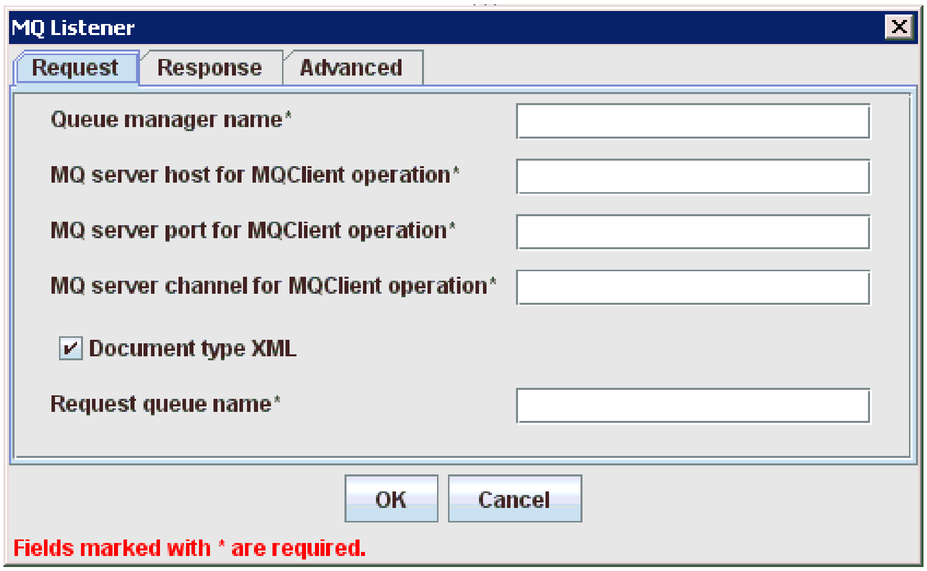
   b. Enter a brief description.

   c. From the Protocol list, select **MQ Series Listener**.

4. Click **Next**.

   The MQ Listener dialog is displayed, as shown in Figure 2–42.

Figure 2–42 MQ Listener Dialog



Queue manager name*

MQ server host for MQClient operation*

MQ server port for MQClient operation*

MQ server channel for MQClient operation*

☑ Document type XML

Request queue name*

Fields marked with * are required.

5. Enter the system information as specified in the following steps:

   a. In the **Request** tab, enter values for the following parameters:

Parameter	Description
Queue manager name	The host on which the MQ Server is located (MQ Client only).
MQ server host for MQClient operation	Port on which the host database is listening.

MQ server port for MQClient operation	The number to connect to an MQ Server queue manager (MQ client only).
	`REQUEST`
	`REQUEST_RESPONSE`
	`REQUEST_ACK`

MQ server channel for MQClient operation	The case-sensitive name of the channel that connects with the remote MQ Server queue manager (MQ client only). The default channel name for MQSeries is SYSTEM.DEF.SVRCONN.
Document type XML	Leave the default selection.

Request queue name	Queue where the message is routed and where request documents are received. The name of the queue is case-sensitive and conforms to the following format:
	`Host\queue type$\qName`
	Host
	Is the system name where the MQ Series queuing system is running.
	queue type
	Private queues are queues that are not published in Active Directory and appear only on the local computer where they reside. Private queues are accessible only by Message Queuing applications that recognize the full path name or format name of the queue.
	qName
	Is the name of the queue where messages are placed, for example,
	`iwaykxc1\Private$\siebel`

b. In the **Response** tab, enter values for the following parameters:

Parameter	Definition
Synchronization Type	Select **REQUEST_RESPONSE** from the Synchronization Type list, which is the recommended option.

c. In the **Advanced** tab, enter values for the following parameters.

Parameter	Definition
Message wait interval (msec)	The interval (in milliseconds) when to check for new input. The default is 3 seconds. Optional.
Mode of operation	Choose Sequential or Threaded. ■ Sequential indicates single processing of requests. ■ Threaded indicates processing of multiple requests simultaneously.
Thread limit	If you selected threaded processing, then indicate the maximum number of requests that can be processed simultaneously.

6. Click **OK**.

   A summary is displayed, which provides the channel description, channel status, and available ports. All the information is associated with the channel you created. The channel also appears under the channels node in the left pane

   An X over the icon indicates that the channel is currently disconnected. You must start the channel to activate your event configuration.

7. Right-click the channel and select **Start**.

   The channel you created becomes active. The X over the icon in the left pane disappears.

8. To stop the channel, right-click the channel and select **Stop**.

2.13.1.3 Creating a File Channel

To create a File channel:

1. Click the **Events** node.

   The Events window is displayed. The adapters that appear in the left pane support events.

2. In the left pane, expand the **Siebel** node.

   The ports and channels nodes appear.

3. Right-click the **channels** node and select **Add Channel**.

   The Add Channel dialog is displayed. Perform the following steps:

   a. Enter a name for the channel, for example, NewChannel.

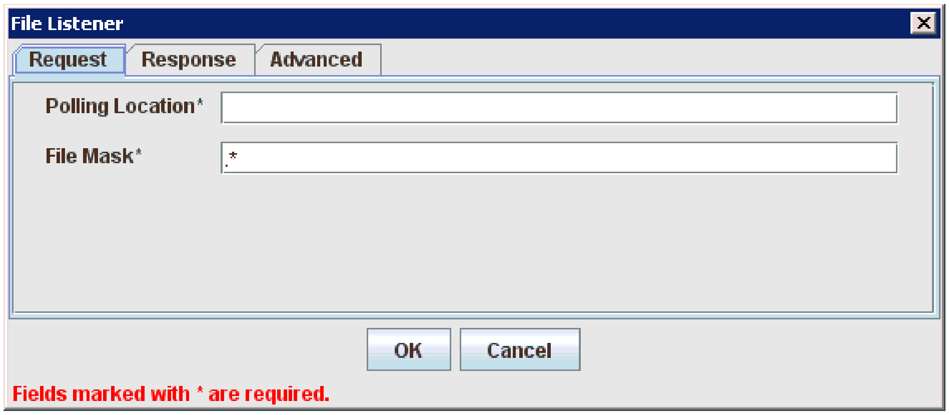
   b. Enter a brief description.

   c. From the Protocol list, select **File Listener**.

4. Click **Next**.

   The File Listener dialog is displayed, as shown in Figure 2–43.

Figure 2–43 File Listener Dialog



File Listener

Polling Location*

File Mask* .*

Fields marked with * are required.

5. Enter the system information as specified in the following steps:

   a. In the **Request** tab, enter values for the following parameters:

Parameter	Description
Polling Location	The target file system location for the Siebel XML file.
File Mask	The file name to be used for the output file generated by this operation.

b. In the **Response** tab, enter values for the following parameters:

Parameter	Definition
Synchronization Type	Select **REQUEST_RESPONSE** from the Synchronization Type list, which is the recommended option.
Response/Ack Directory	Directory where responses or acknowledgments are sent.

c. In the **Advanced** tab, enter values for the following parameters:

Parameter	Definition
Error Directory	Directory to which documents with errors are written.
Poll interval (msec)	The interval (in milliseconds) when to check for new input. The default is 3 seconds. Optional.
Processing Mode	Choose Sequential or Threaded. ■ Sequential indicates single processing of requests. ■ Threaded indicates processing of multiple requests simultaneously.
Thread limit	If you selected threaded processing, then indicate the maximum number of requests that can be processed simultaneously.

6. Click **OK**.

   A summary is displayed, which provides the channel description and channel status. All the information is associated with the channel you created. The channel also appears under the channels node in the left pane.

   An X over the icon indicates that the channel is currently disconnected. You must start the channel to activate your event configuration.

7. Right-click the channel and select **Start**.

   The channel you created becomes active.

   The X over the icon in the left pane disappears.

8. To stop the channel, right-click the channel and select **Stop**.

2.13.1.4 Editing a Channel

To edit a channel:

1. In the left pane, select the channel you want to edit.
2. Right-click the channel and select **Edit**.

   The Edit channels dialog is displayed.

3. Make the required changes to the channel configuration and click **OK**.

2.13.1.5 Deleting a Channel

To delete a channel:

1. In the left pane, select the channel you want to delete.
2. Right-click the channel and select **Delete**.

   The channel disappears from the list in the left pane.

3

Oracle WebLogic Server Deployment and Integration

This chapter describes Oracle WebLogic Server (OracleWLS) deployment and integration with Oracle Application Adapter for Siebel. It contains the following sections:

- Section 3.1, "Adapter Integration with Oracle WebLogic Server"
- Section 3.2, "Deployment of Adapter"
- Section 3.3, "Updating Adapter Configuration"

See Also:

- *Oracle Application Server Adapter Concepts Guide*

3.1 Adapter Integration with Oracle WebLogic Server

Oracle Application Adapter for Siebel is deployed within an OracleWLS container during installation. All client applications run within the OracleWLS environment. In a J2CA deployment, the Common Client Interface (CCI) integrates an OracleWLS client application with a resource adapter.

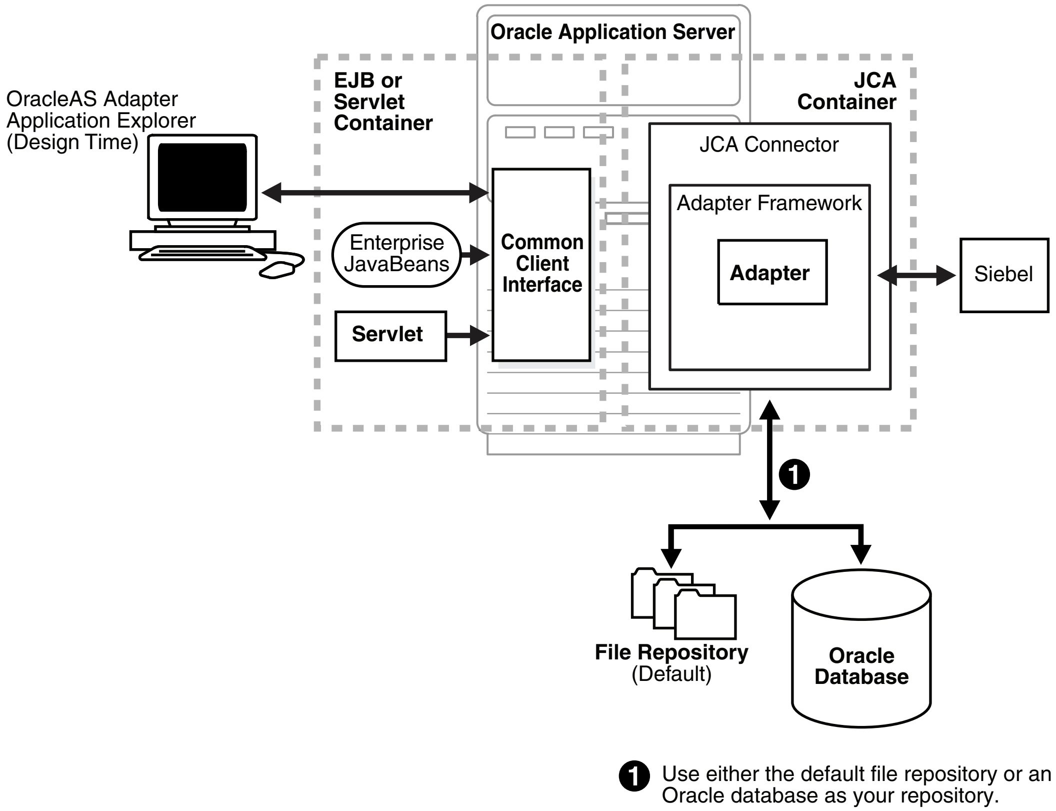
See Also:

- *Oracle Application Server Adapter Concepts Guide*

3.2 Deployment of Adapter

Figure 3–1 shows deployment of the Connector to the Oracle WebLogic Server. In a run-time service scenario, an Enterprise Java Bean, Servlet, or Java program client makes CCI calls to J2CA resource adapters. The adapters process the calls as requests and send them to the EIS. The EIS response is then sent back to the client.

Figure 3–1 Oracle Adapter J2CA Architecture



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

See Also:

- *Oracle Application Server Adapter Concepts Guide*

3.3 Updating Adapter Configuration

This section contains the following topics:

- Section 3.3.1, "Creating a Managed Connector Factory Object"
- Section 3.3.2, "Creating Multiple Managed Connector Factory Objects"
- Section 3.3.3, "Modifying WSDL Files for Additional Connection Factory Values"

During the J2CA deployment of OracleAS Adapter for Siebel, OracleWLS generates a deployment descriptor called `ra.xml`, located in:

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

Your installation contains more than one file named `ra.xml`. The OracleWLS deployment descriptor that is described in this section is located in the directory specified above.

Note: Multiple managed connection factories are supported only for outbound processing (services).

3.3.1 Creating a Managed Connector Factory Object

The ra.xml descriptor provides OracleWLS-specific deployment information for resource adapters. For example, the default jca_sample configuration in Application Explorer is represented in the ra.xml file as follows:

```
<?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>IWAFJCA10</display-name>
  <vendor-name>IWAY Software</vendor-name>
  <spec-version>1.0</spec-version>
  <eis-type>IWAF</eis-type>
  <version>1.0</version>
  <license>
    <license-required>false</license-required>
  </license>
  <resourceadapter>

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

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

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

<connection-impl-class>com.ibi.afjca.cci.IWAFConnection</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>
```

The parameters defined in the ra.xml file are described in the following table:

Parameter Name	Description
IWayHome	The base installation directory for the OracleWLS packaged application adapter.
IWayConfig	The adapter configuration name as defined in Application Explorer. For example, Oracle Application Adapter for Siebel has a preconfigured `jca_sample` configuration in Application Explorer.
IWayRepoURL	The URL to use when opening a connection to the database. This is necessary only when using an Oracle database as the repository.
IWayRepoUser	User name to use when connecting to the database. This is necessary only when using an Oracle database as the repository.
IWayRepoPassword	Password. If provided, then it overwrites configuration. This is necessary only when using an Oracle database as the repository.
loglevel	It overwrites the level set by the ManagedConnectorFactory property.

3.3.2 Creating Multiple Managed Connector Factory Objects

To establish multiple managed connector factory objects, you must edit the `weblogic-ra.xml` file and add more `<connection-instance>` nodes. This file is located in:

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

For example, the first jca_configuration in Application Explorer is represented in the `weblogic-ra.xml` file as follows:

```
<?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-fac
tory-interface>
            <connection-instance>
                <jndi-name>eis/OracleJCAAdapter/DefaultConnection</jndi-name>
            </connection-instance>
        </connection-definition-group>
    </outbound-resource-adapter>
</weblogic-connector>
```

To create multiple managed connector factory objects, you must add new `<connection-instance>` nodes in the file. For example:

```
<?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-fac
tory-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>
```

If you do not specify a `<property>` element in the `<connection-instance>` section, then the value is taken from the `ra.xml` file. You can specify the default properties in the `ra.xml` file and then override them as required in the `weblogic-ra.xml` file. In addition, note that the J2CA configuration (for example, jca_sample2) must already be created in Application Explorer.

> **Note:** When you modify the `ra.xml` and `weblogic-ra.xml` files, the Oracle WebLogic Server must be restarted. If the Oracle WebLogic Server is already running, then stop the Oracle WebLogic Server and then restart it.
>
> In addition, the `iwafjca.rar` file must be redeployed in the Oracle WebLogic Administration Console to activate these changes.

3.3.3 Modifying WSDL Files for Additional Connection Factory Values

Application Explorer generates the J2CA properties file using the default connection factory name `eis/OracleJCAAdapter/DefaultConnection`. If you created additional connection factories, then the WSDLs generated for the additional configuration and connection factory must be changed to reflect the location field of the `jca:address` section in the J2CA properties file. The default J2CA properties file for the Oracle Application Adapter for Siebel with a configuration of `isdsrv2_conn2` is shown in the following example.

Notice that the J2CA properties file has the following default connection factory: `eis/OracleJCAAdapter/DefaultConnection`

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

The connection factory value must be changed to the following: `eis/OracleJCAAdapter/DefaultConnection1`

For example:

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

Note that only the value for the location field in the `jca:address` section should be modified. Do not modify any other field or section.

4 Integration With BPEL Service Components in the Oracle SOA Suite

Oracle Application Adapter for Siebel integrates seamlessly with Business Process Execution Language (BPEL) Process Manager to facilitate Web service integration. Oracle BPEL Process Manager is based on the Service-Oriented Architecture (SOA). It consumes adapter services exposed as Web Service Definition Language (WSDL) documents.

This chapter contains the following topics:

- Section 4.1, "Overview"
- Section 4.2, "Deployment of Adapter"
- Section 4.3, "Configuring a New Application Server Connection"
- Section 4.4, "Designing an Outbound BPEL Process for Service Integration (J2CA Configuration)"
- Section 4.5, "Designing an Inbound BPEL Process for Event Integration (J2CA Configuration)"
- Section 4.6, "Designing an Outbound BPEL Process for Service Integration (BSE Configuration)"

4.1 Overview

To integrate with Oracle BPEL Process Manager, Oracle Application Adapter for Siebel must be deployed in the same WLS container as Oracle BPEL Process Manager. The underlying adapter services must be exposed as WSDL files, which are generated during design time in Oracle Adapter Application Explorer (Application Explorer) for both request-response (outbound) and event notification (inbound) services of the adapter. For more information, see "Generating WSDL (J2CA Configurations Only)" on page 2-27.

The generated WSDL files are used to design the appropriate BPEL processes for inbound or outbound adapter services. A completed BPEL process must be successfully compiled in Oracle JDeveloper and deployed to a BPEL server. Upon deployment to the BPEL server, every newly built process is automatically deployed to the Oracle Enterprise Manager console, where you run, monitor, administer BPEL processes, and listen to adapter events.

4.2 Deployment of Adapter

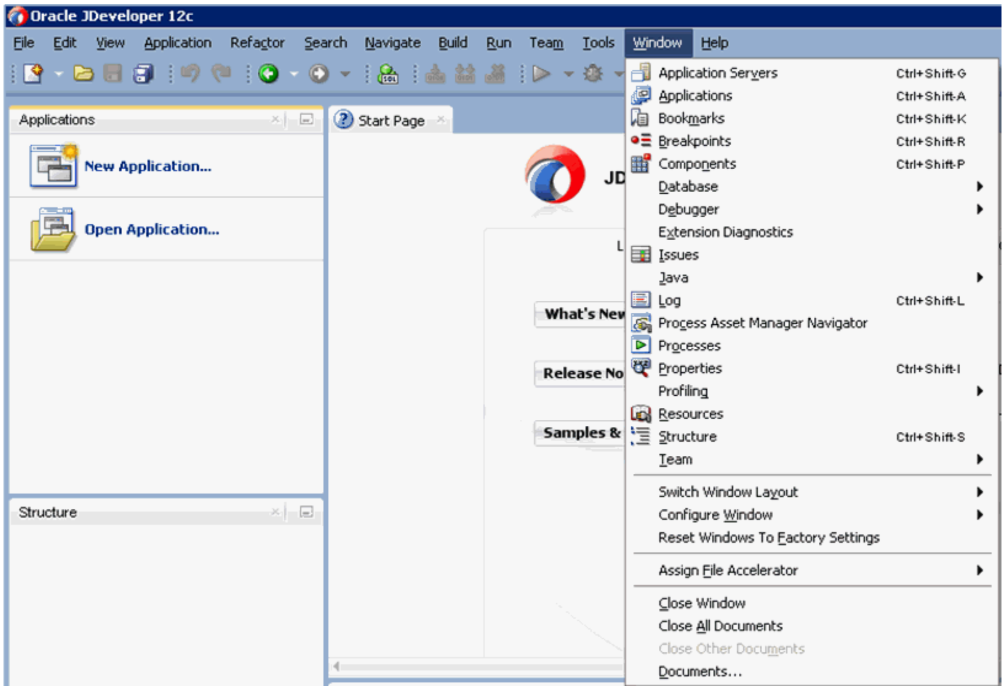
During installation, Oracle Application Adapter for Siebel is deployed as a J2CA 1.0 resource adapter within the WLS container. The adapter must be deployed in the same WLS container as Oracle BPEL Process Manager.

4.3 Configuring a New Application Server Connection

To configure a new Application Server connection in Oracle JDeveloper:

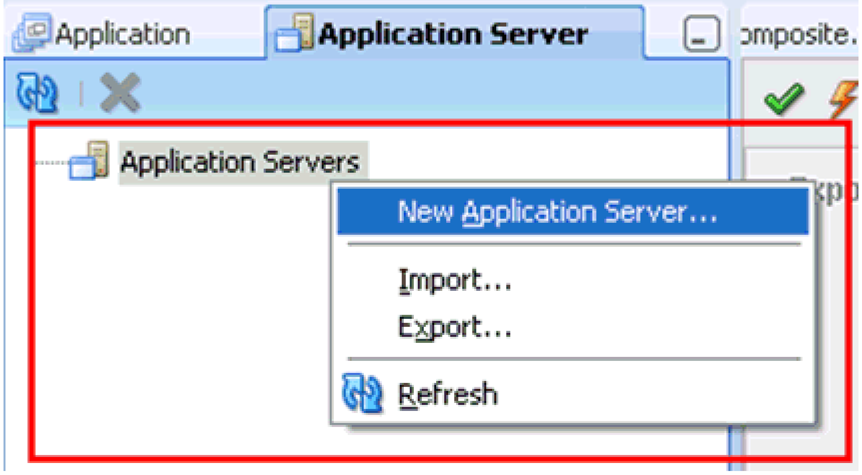
1. Open **Oracle JDeveloper** on your system.
2. From the menu bar, click **Window** and select **Application Server Navigator**, as shown in Figure 4–1.

Figure 4–1 Application Server Navigator



The Application Server tab is displayed, as shown in Figure 4–2.

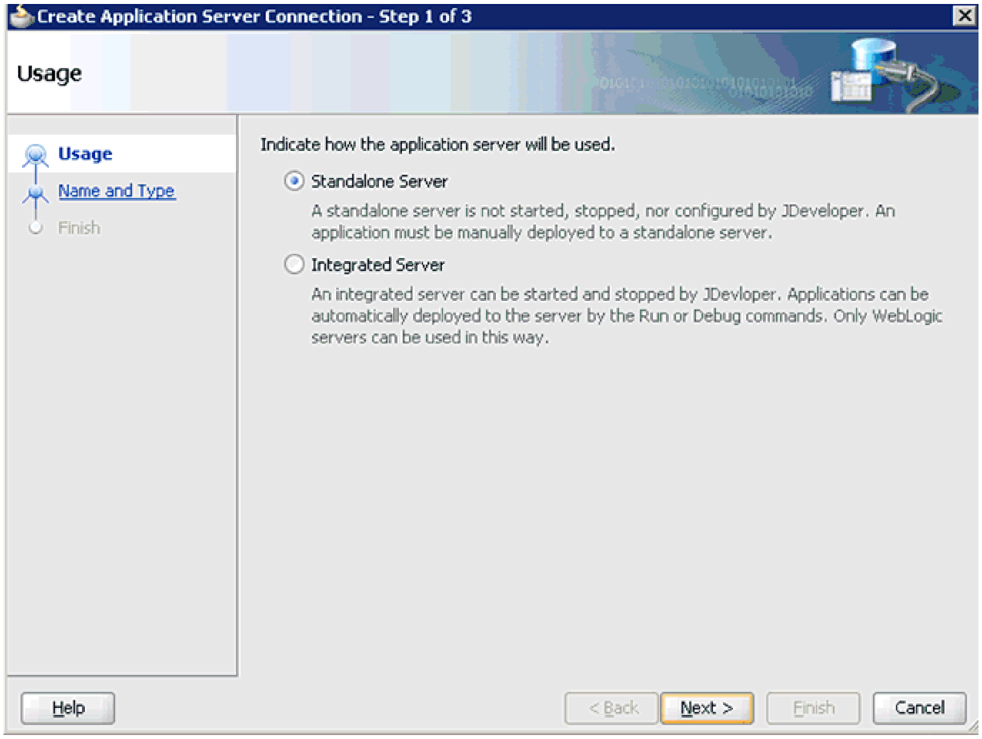
Figure 4–2 Application Server Tab



3. Right-click **Application Servers** and select **New Application Server**.

   The Create Application Server Connection Wizard is displayed, as shown in Figure 4–3.

Figure 4–3 Create Application Server Connection Wizard



Indicate how the application server will be used.

● Standalone Server

A standalone server is not started, stopped, nor configured by JDeveloper. An application must be manually deployed to a standalone server.

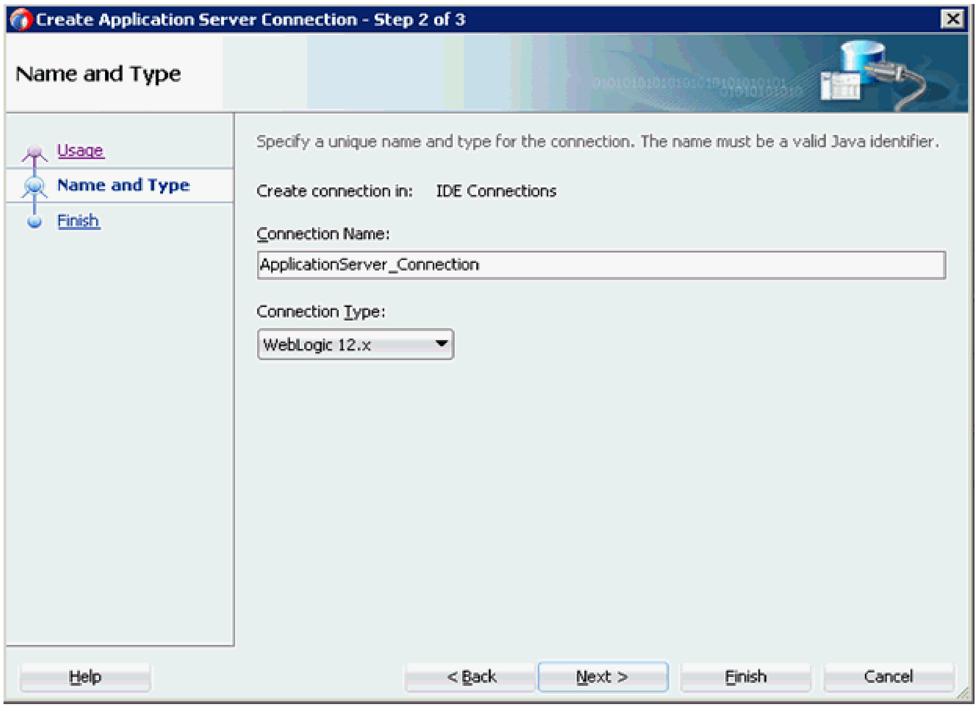
○ Integrated Server

An integrated server can be started and stopped by JDevloper. Applications can be automatically deployed to the server by the Run or Debug commands. Only WebLogic servers can be used in this way.

4. Accept the default selection (Standalone Server) and click **Next**.

   The Name and Type page is displayed, as shown in Figure 4–4.

Figure 4–4 Name and Type Page



Specify a unique name and type for the connection. The name must be a valid Java identifier.

Create connection in: IDE Connections

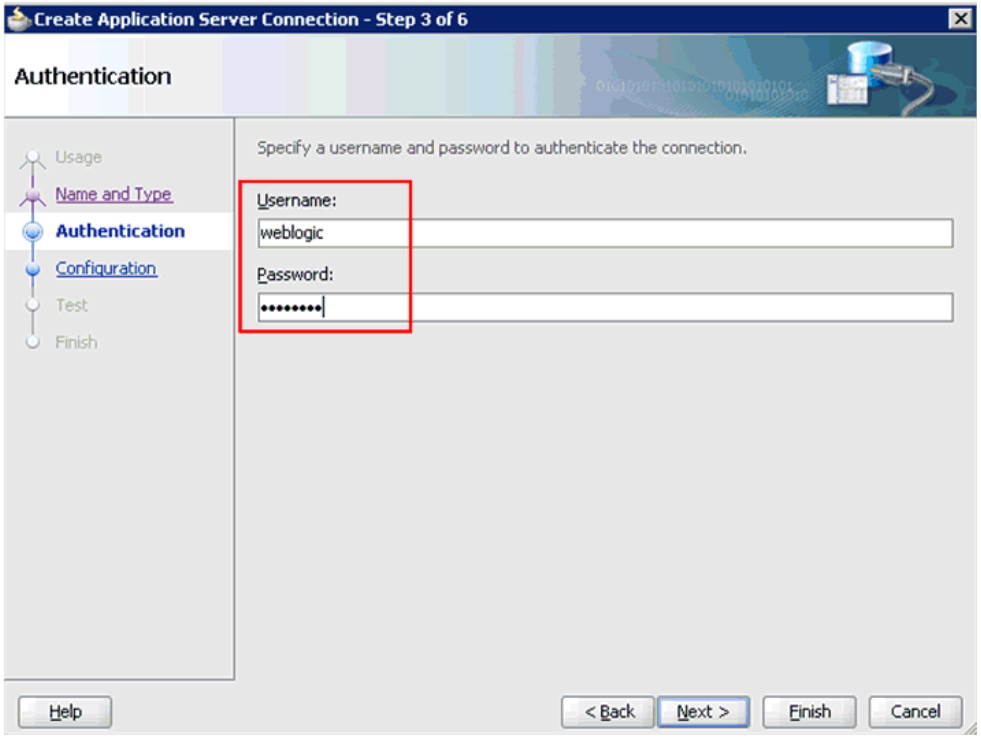
Connection Name: ApplicationServer_Connection

Connection Type: WebLogic 12.x

5. Specify a new name for the Application Server connection and click **Next**.

   The Authentication page is displayed, as shown in Figure 4–5.

Figure 4–5 Authentication Page



Specify a username and password to authenticate the connection.

Username: weblogic

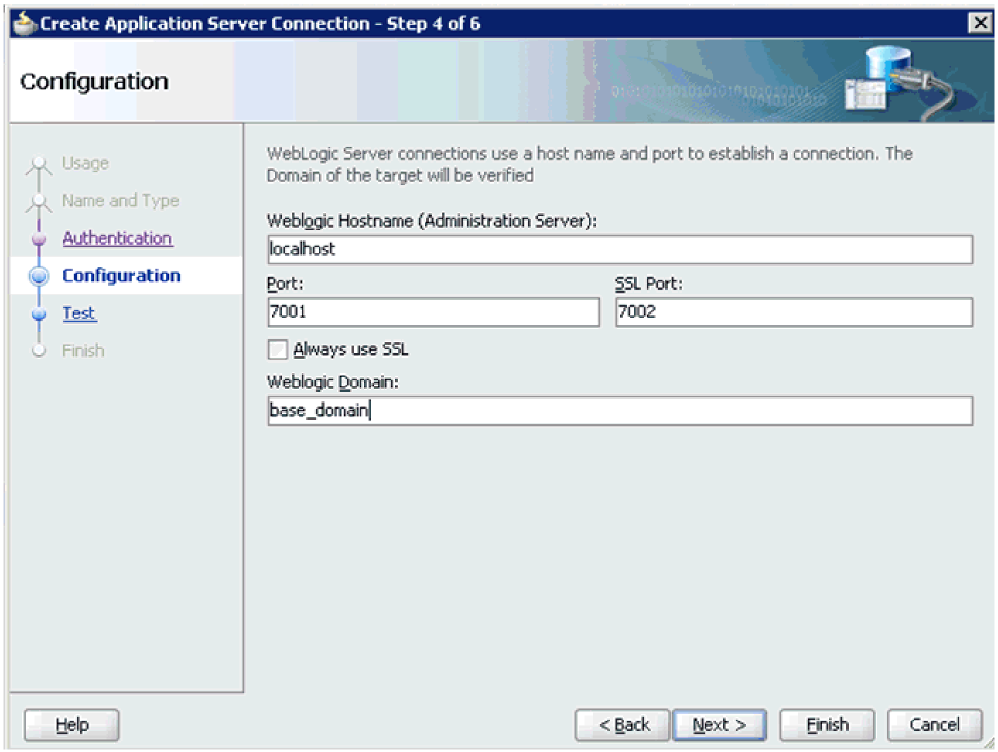
Password: ••••••••

6. Specify a valid user name (for example, weblogic) and a password (for example, welcome1) for your new connection.

7. Click **Next**.

   The Configuration page is displayed, as shown in Figure 4–6.

Figure 4–6 Configuration Page



Create Application Server Connection - Step 4 of 6

Configuration

WebLogic Server connections use a host name and port to establish a connection. The Domain of the target will be verified

Weblogic Hostname (Administration Server): localhost

Port: 7001

SSL Port: 7002

☐ Always use SSL

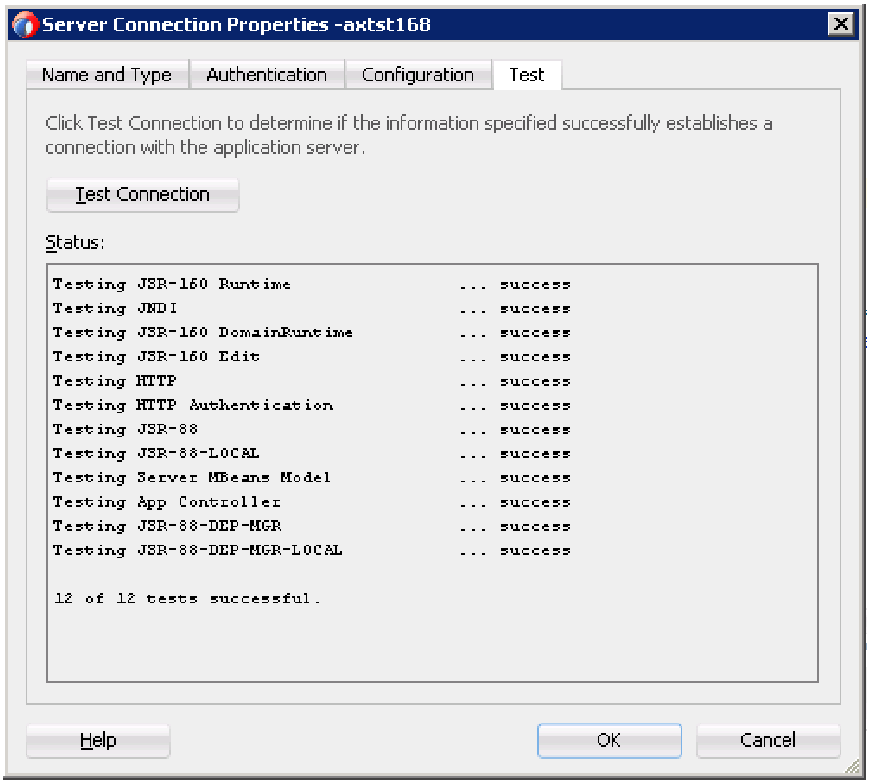
Weblogic Domain: base_domain

8. Specify the Oracle WebLogic host name (for example, localhost), which is the system IP where the process must deploy and Oracle WebLogic domain (for example, base_domain).

9. Click **Next**.

   The Test page is displayed, as shown in Figure 4–7.

Figure 4–7 Test Page



Click Test Connection to determine if the information specified successfully establishes a connection with the application server.

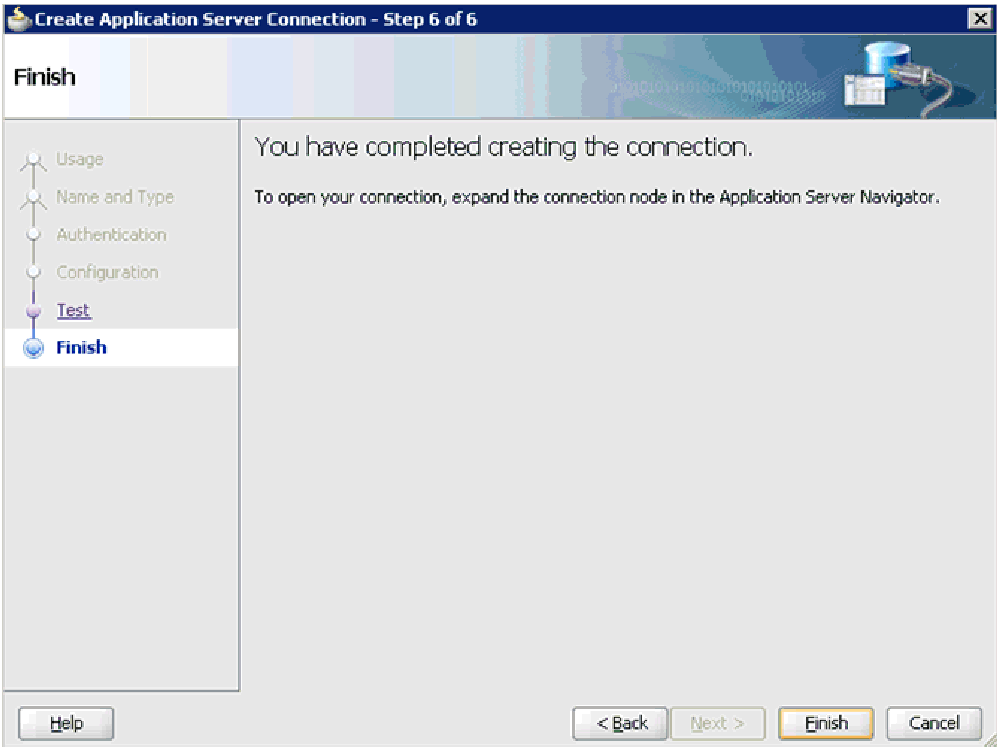
Status:

```
Testing JSR-160 Runtime                      ... success
Testing JNDI                                 ... success
Testing JSR-160 DomainRuntime                ... success
Testing JSR-160 Edit                         ... success
Testing HTTP                                 ... success
Testing HTTP Authentication                  ... success
Testing JSR-88                               ... success
Testing JSR-88-LOCAL                         ... success
Testing Server MBeans Model                  ... success
Testing App Controller                       ... success
Testing JSR-88-DEP-MGR                       ... success
Testing JSR-88-DEP-MGR-LOCAL                 ... success

12 of 12 tests successful.
```

10. Click **Test Connection**.
11. Make sure that the test status is successful.
12. Click **Next**.

    The Finish page is displayed, as shown in Figure 4–8.

Figure 4–8 Finish Page



You have completed creating the connection.

To open your connection, expand the connection node in the Application Server Navigator.

13. Click **Finish**.

    The new Application Server connection is listed in the left pane (Application Server tab).

4.4 Designing an Outbound BPEL Process for Service Integration (J2CA Configuration)

This section describes how to design an outbound BPEL process for service integration.

A sample project has been provided for this outbound use case scenario in the following folder of the Application Adapters installation:

```
<ADAPTER_HOME>\etc\sample\SIEBEL_Samples.zip\SIEBEL_Samples\BPEL\J2CA\Outbound_
Project
```

The following tools are required to complete your outbound design-time configuration:

- Oracle Adapter Application Explorer (Application Explorer)
- Oracle JDeveloper BPEL Designer (JDeveloper)

> **Note:** The examples in this chapter demonstrate the use of JDeveloper.

This section contains the following topics:

- Section 4.4.1, "Generating WSDL for Request/Response Service"

- Section 4.4.2, "Creating an Empty Composite for SOA"
- Section 4.4.3, "Defining a BPEL Outbound Process"
- Section 4.4.4, "Deploying the BPEL Outbound Process"
- Section 4.4.5, "Invoking the Input XML Document in the Oracle Enterprise Manager Console"
- Section 4.4.6, "Testing Outbound BPEL and Mediator Processes"

4.4.1 Generating WSDL for Request/Response Service

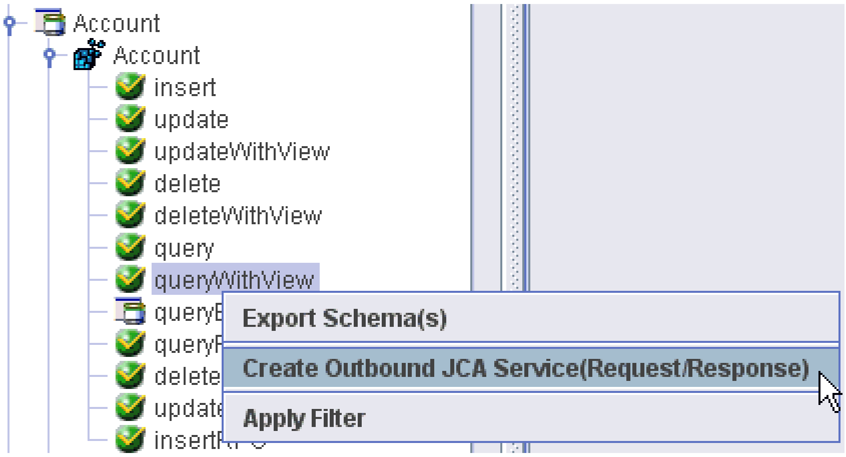
Before you design a BPEL process, you must generate the respective WSDL file using Application Explorer. Perform the following steps to generate a WSDL for the request/response service:

1. Start **Application Explorer** and connect to a defined Siebel target or create a new target.

   For more information on starting the Application Explorer and on connecting a target, see Section 2.1, "Starting Application Explorer" on page 2-2 and Connecting to a Defined Target on page 2-9.

2. Expand the Siebel target to which you are connected.
3. As shown in Figure 4–9, expand **Business Object**, **Account**, and then **Account**.

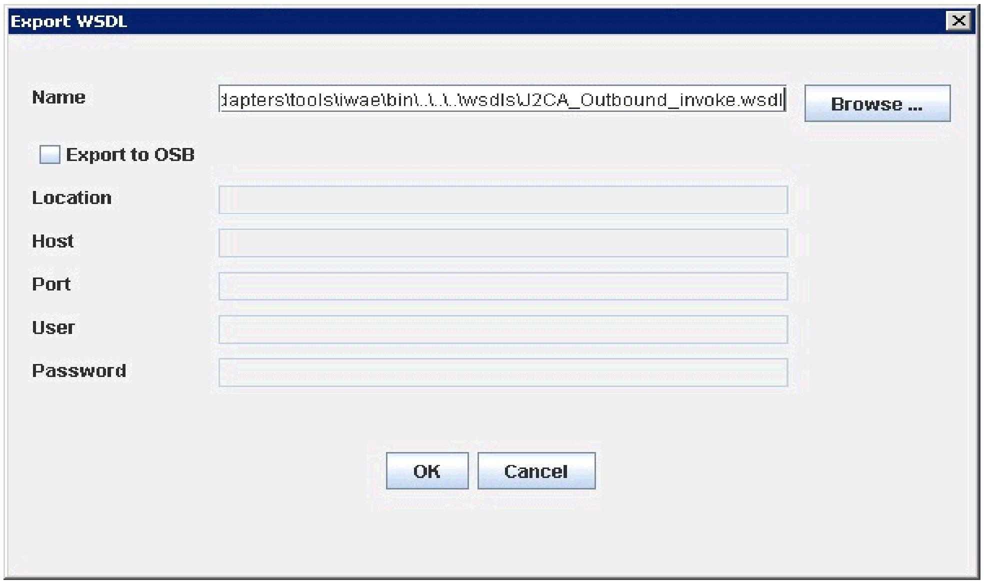
Figure 4–9 Create Outbound JCA Service (Request/Response) Option



4. Right-click **queryWithView**, and then select **Create Outbound JCA Service (Request/Response)**.

   The Export WSDL dialog is displayed, as shown in Figure 4–10.

Figure 4–10 Export WSDL Dialog


Export WSDL

Name: \adapters\tools\iwae\bin\..\..\..\wsdls\J2CA_Outbound_invoke.wsdl

☐ Export to OSB

Location

Host

Port

User

Password

5. Accept the default name or provide a name (for example, J2CA_Outbound) for the file.

   The .wsdl file extension is added automatically. By default, the names of WSDL files generated for request-response services end with _invoke.

6. Click **OK**.

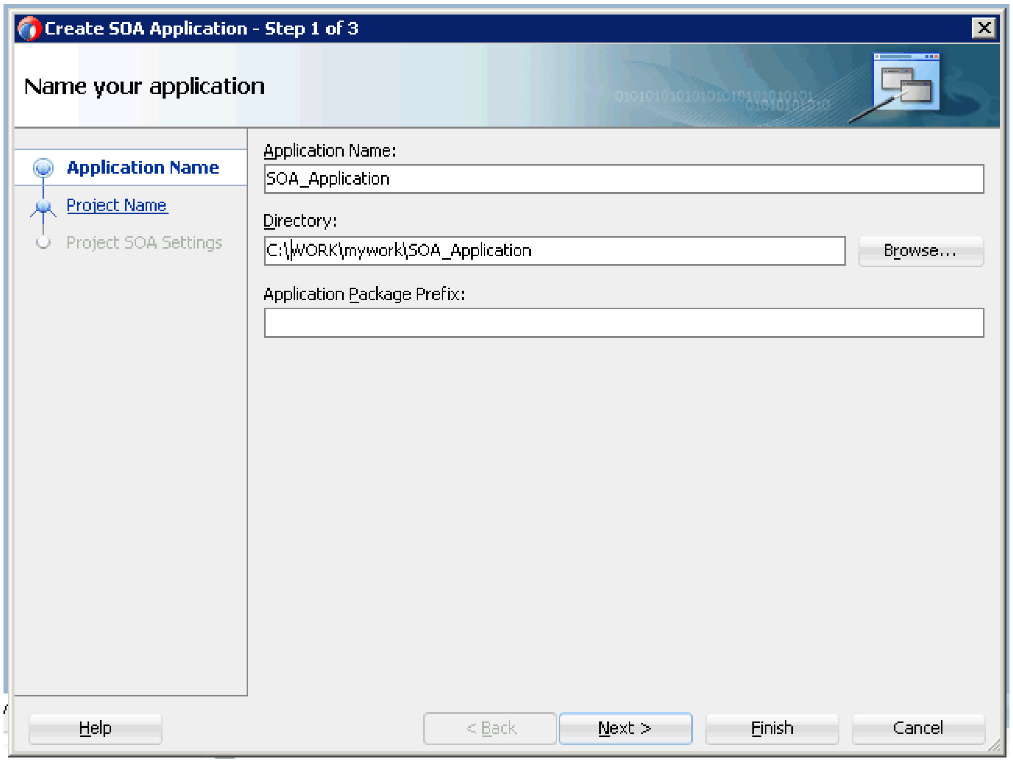
You can now create an empty composite for SOA, which is the first step that is required to define a BPEL outbound process in Oracle JDeveloper.

4.4.2 Creating an Empty Composite for SOA

Perform the following steps to create an empty composite for SOA:

1. Create a new SOA application.
2. Enter a name for the new SOA Application and click **Next**, as shown in Figure 4–11.

Figure 4–11 Name Your Application Page



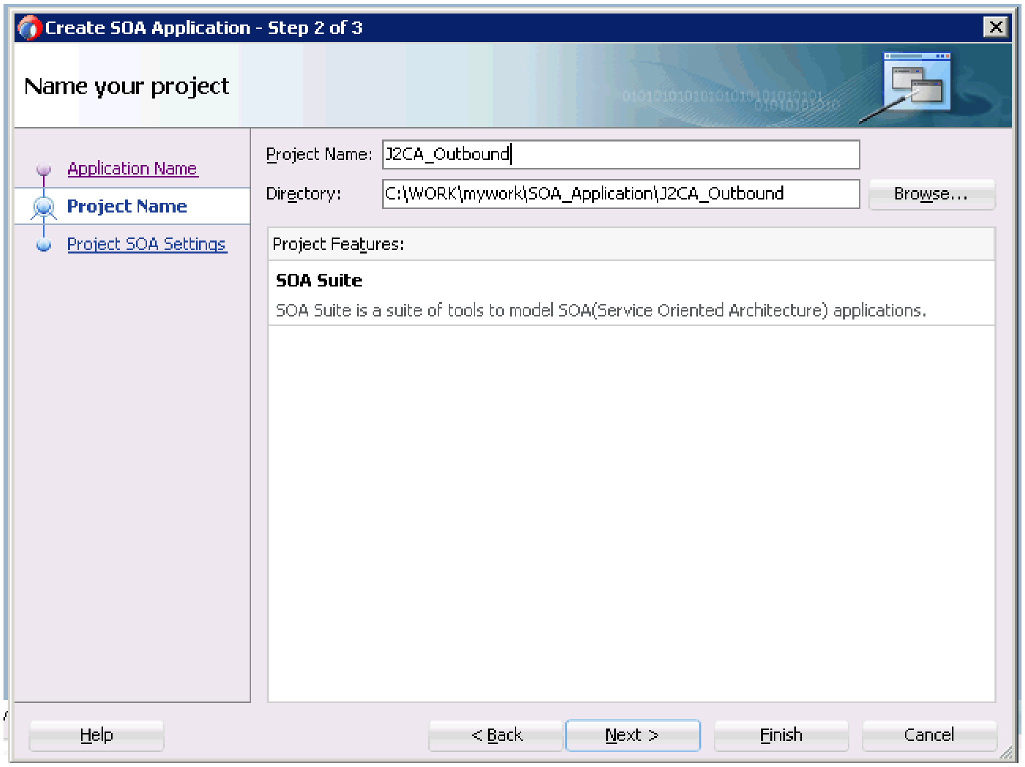
Application Name: SOA_Application

Directory: C:\WORK\mywork\SOA_Application

Application Package Prefix:

The Name your project page is displayed, as shown in Figure 4–12.

Figure 4–12 Name Your Project Page



Project Name: J2CA_Outbound

Directory: C:\WORK\mywork\SOA_Application\J2CA_Outbound

Project Features:

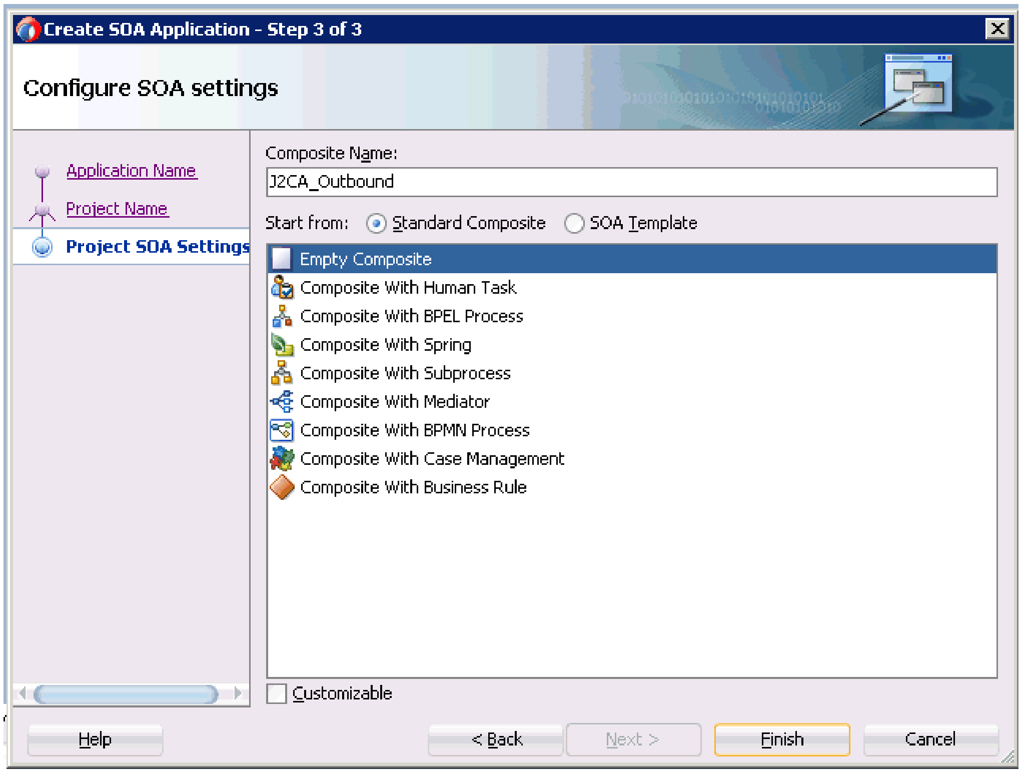
SOA Suite

SOA Suite is a suite of tools to model SOA(Service Oriented Architecture) applications.

3. Enter a project name and click **Next**.

   The Configure SOA settings page is displayed, as shown in Figure 4–13.

Figure 4–13 Configure SOA Settings Page


Create SOA Application - Step 3 of 3
Configure SOA settings
Application Name
Project Name
Project SOA Settings
Empty Composite
Composite With Human Task
Composite With BPEL Process
Composite With Spring
Composite With Subprocess
Composite With Mediator
Composite With BPMN Process
Composite With Case Management
Composite With Business Rule
Help
< Back
Next >
Finish
Cancel

Composite Name: J2CA_Outbound

Start from: ● Standard Composite ○ SOA Template

□ Customizable

4. From the Composite Template list, select **Empty Composite** and click **Finish**.

4.4.3 Defining a BPEL Outbound Process

This section describes how to define a BPEL outbound process, which consists of the following topics:

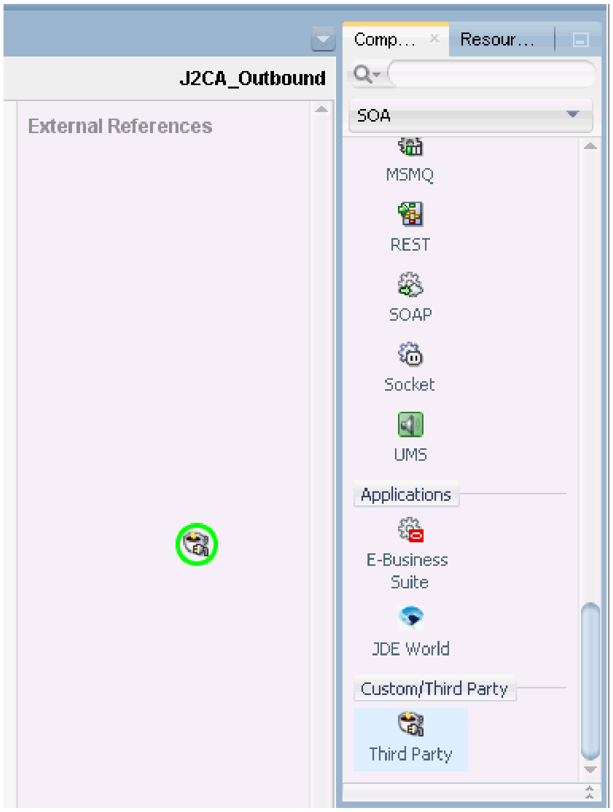
- Section 4.4.3.1, "Configuring a Third Party Adapter Service Component"
- Section 4.4.3.2, "Configuring an Outbound BPEL Process Component"
- Section 4.4.3.3, "Adjusting for Known Deployment Issues With 12c"

4.4.3.1 Configuring a Third Party Adapter Service Component

Perform the following steps to create a third party adapter service component:

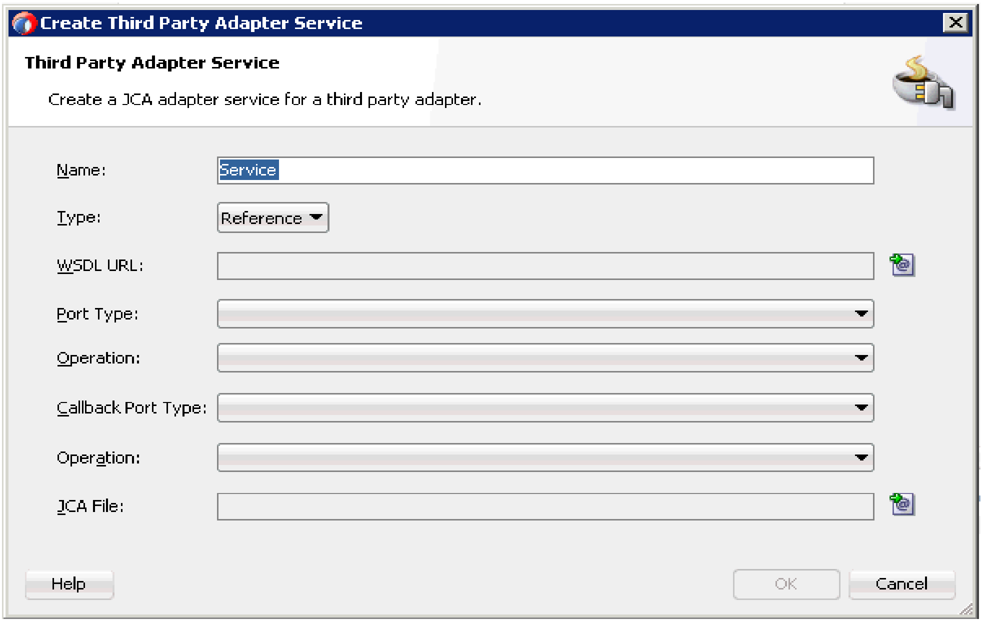
1. Drag and drop the **Third Party Adapter** component from the Service Adapters pane to the External References pane, as shown in Figure 4–14.

Figure 4–14 Third Party Adapter Component



The Create Third Party Adapter Service dialog is displayed, as shown in Figure 4–15.

Figure 4–15 Create Third Party Adapter Service Dialog


Create Third Party Adapter Service

Third Party Adapter Service

Create a JCA adapter service for a third party adapter.

Name: Service

Type: Reference

WSDL URL:

Port Type:

Operation:

Callback Port Type:

Operation:

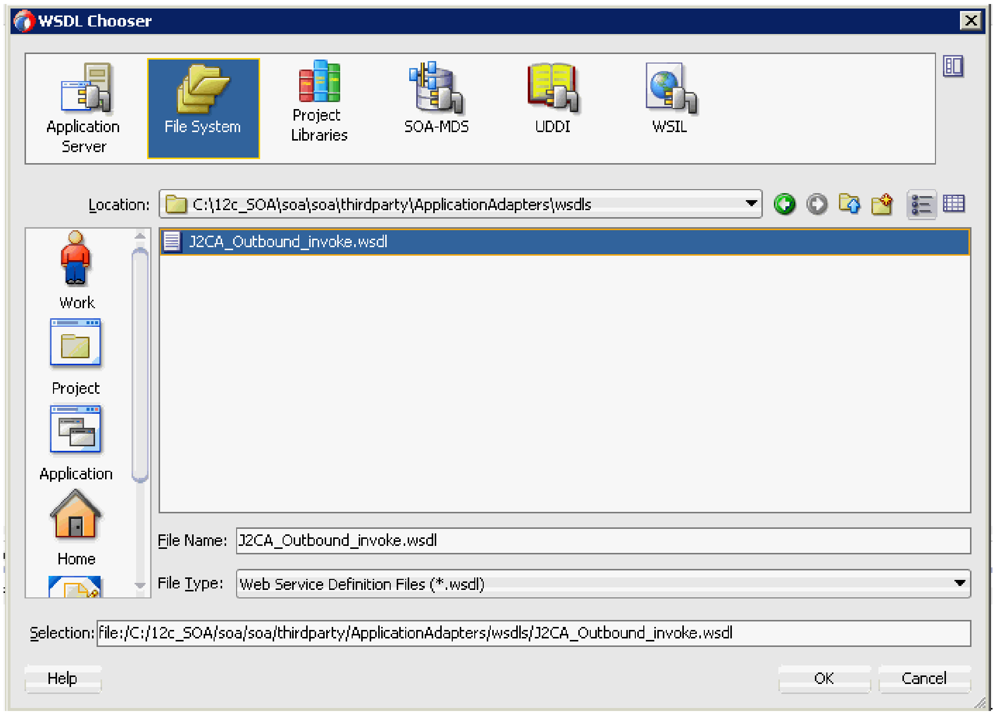
JCA File:

2. Ensure that **Reference** is selected from the Type list (default).

3. Click the **Find existing WSDLs** icon, which is located to the right of the WSDL URL field.

   The WSDL Chooser dialog is displayed, as shown in Figure 4–16.

Figure 4–16 WSDL Chooser Dialog



WSDL Chooser

Location: C:\12c_SOA\soa\soa\thirdparty\ApplicationAdapters\wsdls

J2CA_Outbound_invoke.wsdl

File Name: J2CA_Outbound_invoke.wsdl

File Type: Web Service Definition Files (*.wsdl)

Selection: file:/C:/12c_SOA/soa/soa/thirdparty/ApplicationAdapters/wsdls/J2CA_Outbound_invoke.wsdl

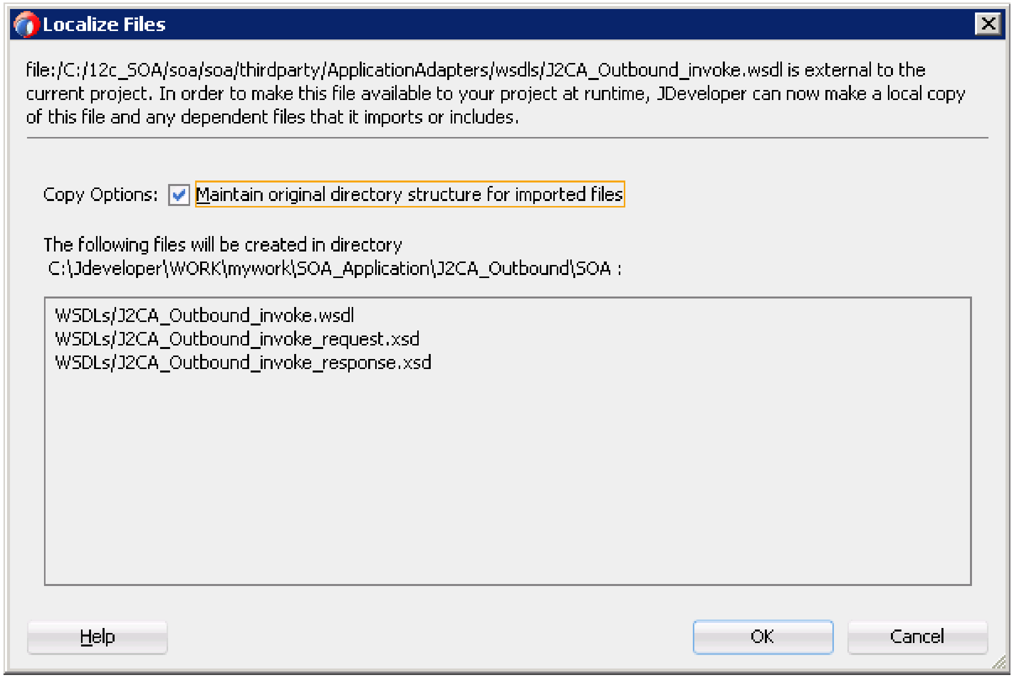
4. Browse and select an outbound WSDL file from the following directory:

   `<ADAPTER_HOME>\wsdls`

5. Click **OK**.

   The Localize Files dialog is displayed, as shown in Figure 4–17.

Figure 4–17 Localize Files Dialog



Localize Files

file:/C:/12c_SOA/soa/soa/thirdparty/ApplicationAdapters/wsdls/J2CA_Outbound_invoke.wsdl is external to the current project. In order to make this file available to your project at runtime, JDeveloper can now make a local copy of this file and any dependent files that it imports or includes.

Copy Options: ☑ Maintain original directory structure for imported files

The following files will be created in directory
C:\Jdeveloper\WORK\mywork\SOA_Application\J2CA_Outbound\SOA :

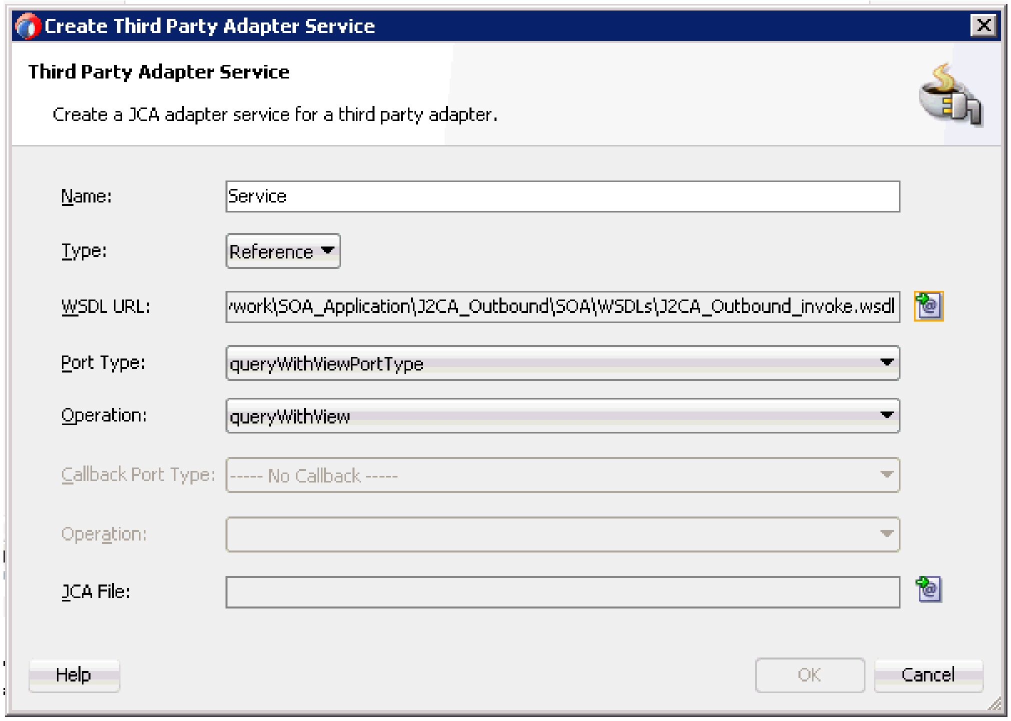
```
WSDLs/J2CA_Outbound_invoke.wsdl
WSDLs/J2CA_Outbound_invoke_request.xsd
WSDLs/J2CA_Outbound_invoke_response.xsd
```

6. Click **OK**.

   The outbound WSDL file and associated request and response XML schema files (.xsd) are imported to the project folder that has been created.

   You are returned to the Create Third Party Adapter Service dialog, as shown in Figure 4–18.

Figure 4–18 Create Third Party Adapter Service Dialog



Create Third Party Adapter Service

Third Party Adapter Service

Create a JCA adapter service for a third party adapter.

Name: Service

Type: Reference

WSDL URL: work\SOA_Application\J2CA_Outbound\SOA\WSDLs\J2CA_Outbound_invoke.wsdl

Port Type: queryWithViewPortType

Operation: queryWithView

Callback Port Type: ----- No Callback -----

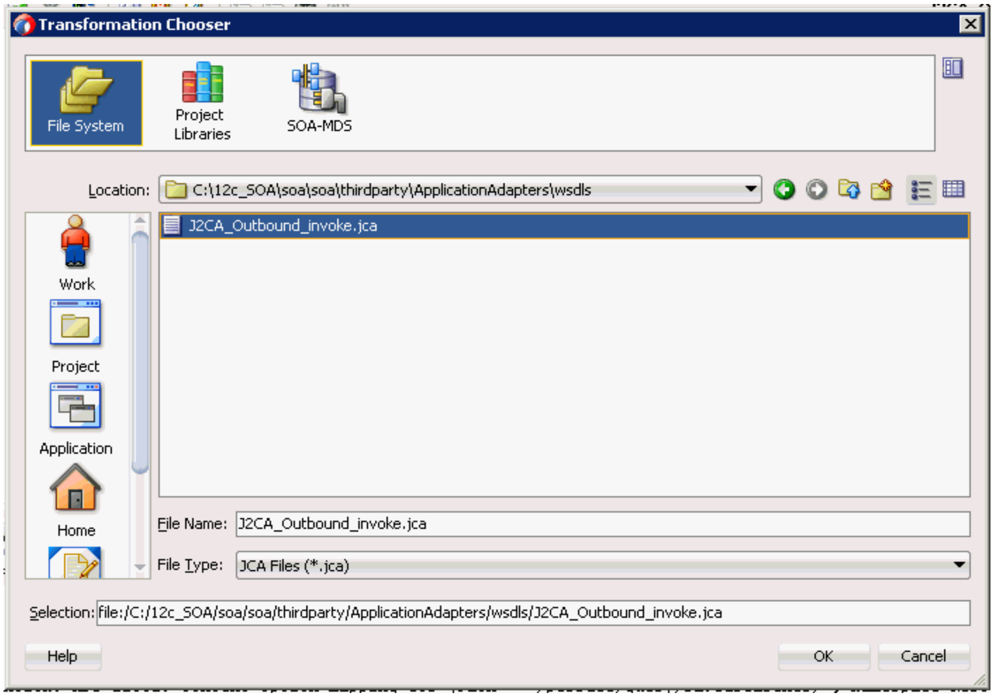
Operation:

JCA File:

7. Click the **Find JCA file** icon, which is located to the right of the JCA File field.

   The Transformation Chooser dialog is displayed, as shown in Figure 4–19.

Figure 4–19 Transformation Chooser Dialog



Location: C:\12c_SOA\soa\soa\thirdparty\ApplicationAdapters\wsdls

J2CA_Outbound_invoke.jca

File Name: J2CA_Outbound_invoke.jca

File Type: JCA Files (*.jca)

Selection: file:/C:/12c_SOA/soa/soa/thirdparty/ApplicationAdapters/wsdls/J2CA_Outbound_invoke.jca

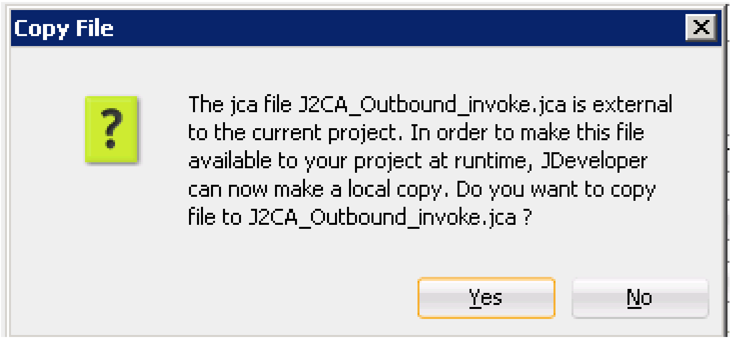
8. Browse and select the JCA properties file from the following directory:

   `<ADAPTER_HOME>\wsdls`

9. Click **OK**.

   The Copy File message is displayed, as shown in Figure 4–20.

Figure 4–20 Copy File Message



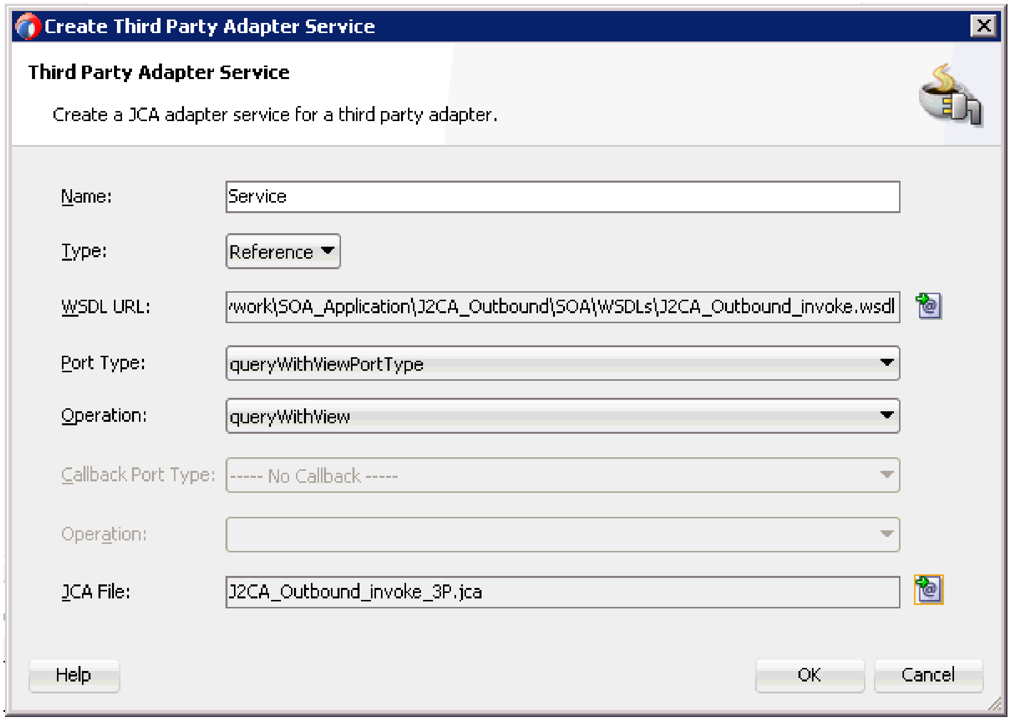
The jca file J2CA_Outbound_invoke.jca is external to the current project. In order to make this file available to your project at runtime, JDeveloper can now make a local copy. Do you want to copy file to J2CA_Outbound_invoke.jca ?

10. Click **Yes**.

    A copy of the JCA properties file is made in the project folder.

    You are returned to the Create Third Party Adapter Service dialog, as shown in Figure 4–21.

Figure 4–21 Create Third Party Adapter Service Dialog



Create Third Party Adapter Service

Third Party Adapter Service

Create a JCA adapter service for a third party adapter.

Name: Service

Type: Reference

WSDL URL: ·work\SOA_Application\J2CA_Outbound\SOA\WSDLs\J2CA_Outbound_invoke.wsdl

Port Type: queryWithViewPortType

Operation: queryWithView

Callback Port Type: ----- No Callback -----

Operation:

JCA File: J2CA_Outbound_invoke_3P.jca

11. Click **OK**.

    The third party adapter service component is created and displayed in the External References pane.

    You are now ready to configure an outbound BPEL process component.

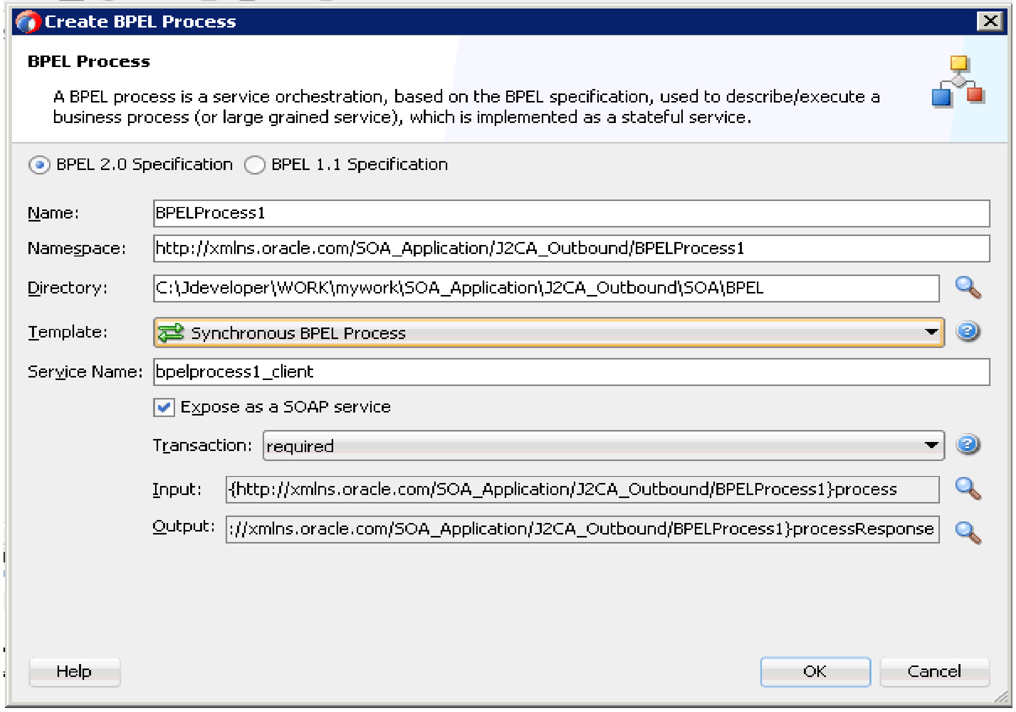
4.4.3.2 Configuring an Outbound BPEL Process Component

Perform the following steps to configure an outbound BPEL process component:

1. Drag and drop the **BPEL Process** component from the Components pane to the Components pane.

   The Create BPEL Process dialog is displayed, as shown in Figure 4–22.

Figure 4–22 Create BPEL Process Dialog



Create BPEL Process

BPEL Process

A BPEL process is a service orchestration, based on the BPEL specification, used to describe/execute a business process (or large grained service), which is implemented as a stateful service.

● BPEL 2.0 Specification ○ BPEL 1.1 Specification

Name: BPELProcess1

Namespace: http://xmlns.oracle.com/SOA_Application/J2CA_Outbound/BPELProcess1

Directory: C:\Jdeveloper\WORK\mywork\SOA_Application\J2CA_Outbound\SOA\BPEL

Template: Synchronous BPEL Process

Service Name: bpelprocess1_client

☑ Expose as a SOAP service

Transaction: required

Input: {http://xmlns.oracle.com/SOA_Application/J2CA_Outbound/BPELProcess1}process

Output: ://xmlns.oracle.com/SOA_Application/J2CA_Outbound/BPELProcess1}processResponse

2. In the Name field, enter a name to identify the new outbound BPEL process component or leave it to the default value.

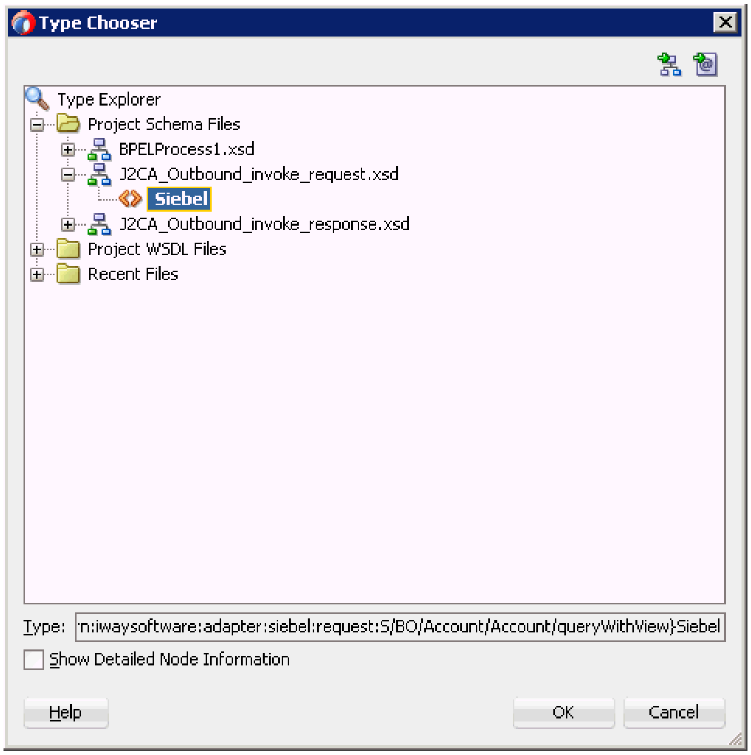
   By default, the BPEL 2.0 Specification option is selected.

3. From the Template list, select **Synchronous BPEL Process**.

4. Click the **Browse** icon, which is located to the right of the Input field to select the associated XML request schema file.

   The Type Chooser dialog is displayed, as shown in Figure 4–23.

Figure 4–23 Type Chooser Dialog



Type: n:iwaysoftware:adapter:siebel:request:S/BO/Account/Account/queryWithView}Siebel

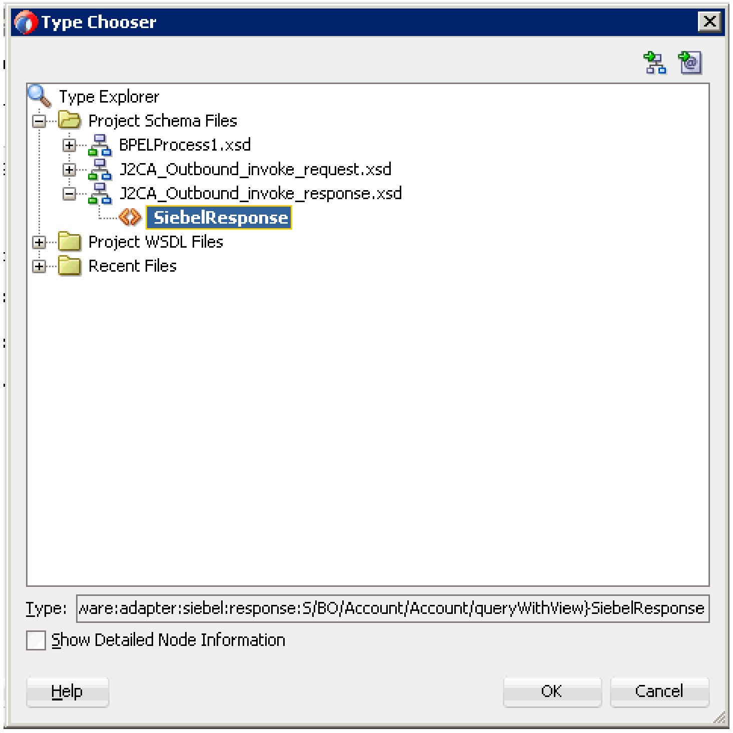
□ Show Detailed Node Information

5. Expand **Project Schema Files**, **J2CA_Outbound_invoke_request.xsd**, and select **Siebel**.
6. Click **OK**.

   You are returned to the Create BPEL Process dialog.
7. Click the **Browse** icon, which is located to the right of the Output field to select the associated XML response schema file.

   The Type Chooser dialog is displayed, as shown in Figure 4–24.

Figure 4–24 Type Chooser Dialog



Type: ware:adapter:siebel:response:S/BO/Account/Account/queryWithView}SiebelResponse

□ Show Detailed Node Information

8. Expand **Project Schema Files**, **J2CA_Outbound_invoke_response.xsd**, and select **SiebelResponse**.

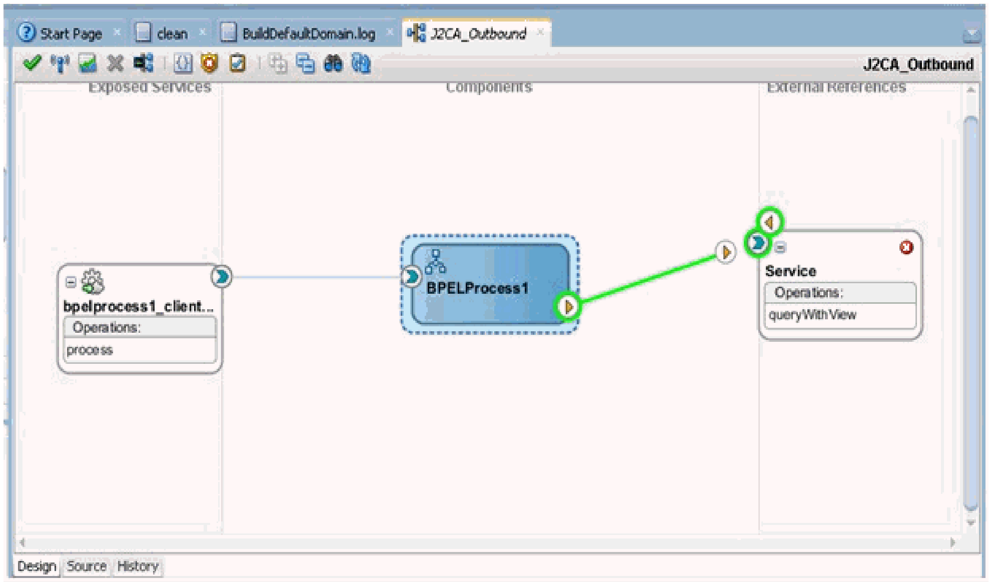
9. Click **OK**.

   You are returned to the Create BPEL Process dialog.

10. Click **OK**.

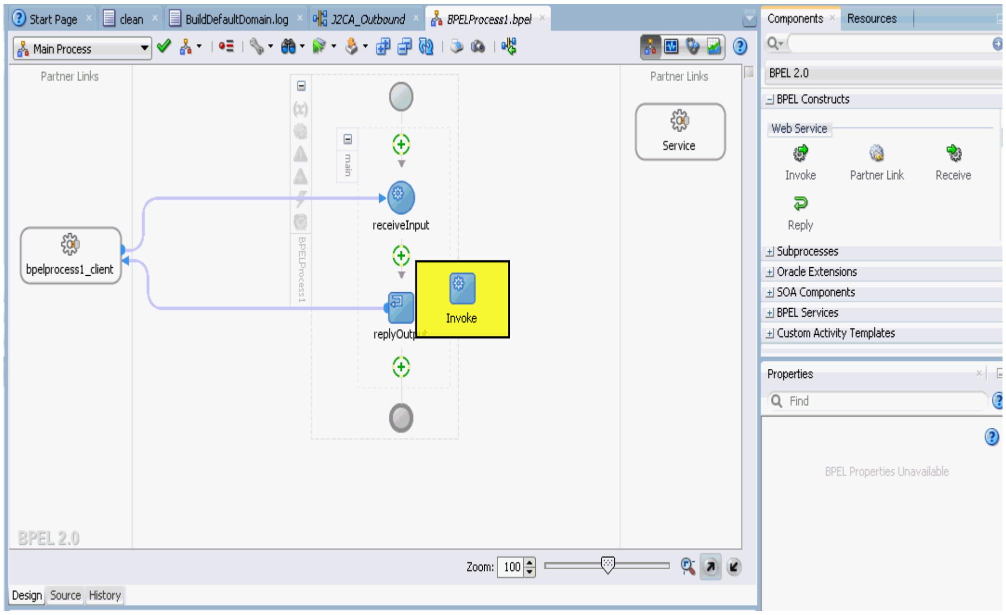
11. Create a connection between the outbound BPEL process component and the third party adapter service component, as shown in Figure 4–25.

Figure 4–25 Created Connection



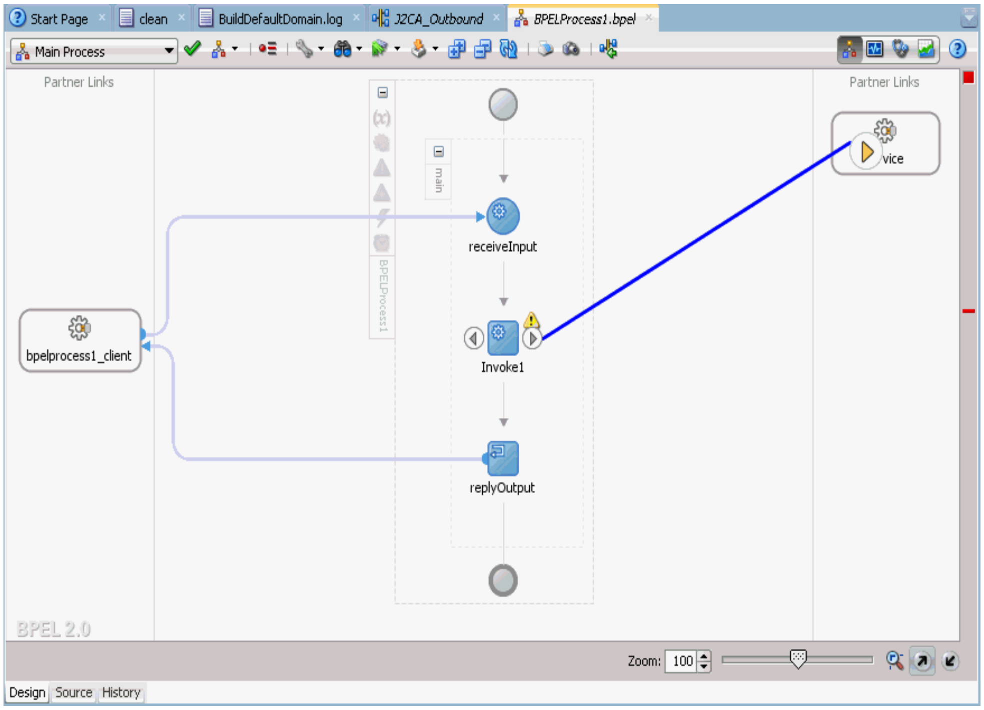
12. Double-click the outbound BPEL process component in the Components pane.
13. Drag and drop the **Invoke** activity component under BPEL Constructs - Web Service, to the Components pane and place it between the **receiveInput** activity component and the **replyOutput** activity component, as shown in Figure 4–26.

Figure 4–26 Invoke Activity Component



14. Create a connection between the new Invoke activity component Service and the third party adapter service component (Service), as shown in Figure 4–27.

Figure 4–27 Created Connection



The Edit Invoke dialog is displayed.

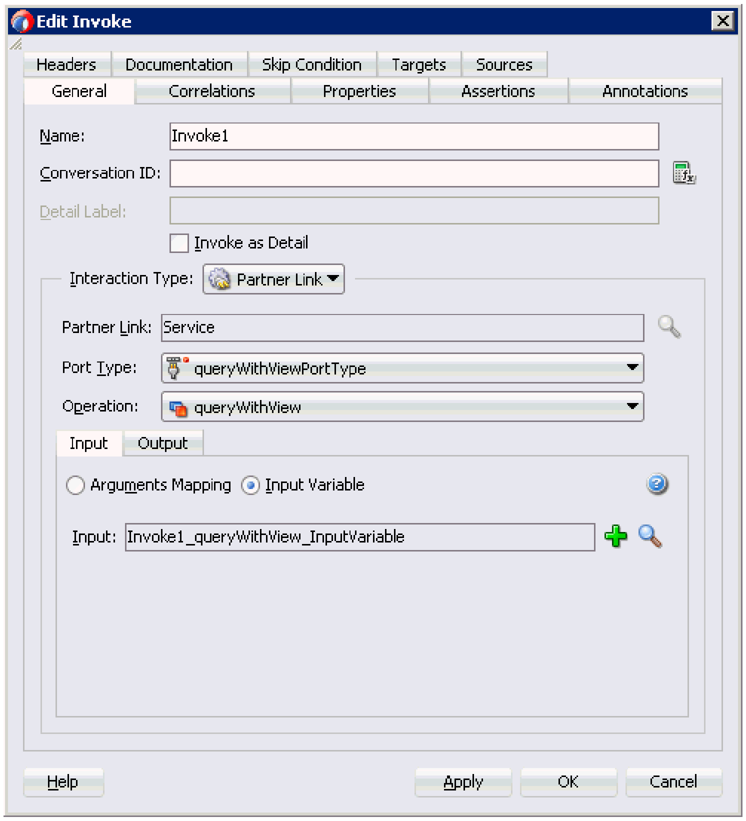
15. Click the **Plus sign** icon, which is located to the right of the Input field to configure a new input variable.

    The Create Variable dialog is displayed.

16. Accept the default values that are provided for the new input variable and click **OK**.

    You are returned to the Edit Invoke dialog, as shown in Figure 4–28.

Figure 4–28 Edit Invoke Dialog



Edit Invoke

Name: Invoke1

Conversation ID:

Detail Label:

□ Invoke as Detail

Interaction Type: Partner Link

Partner Link: Service

Port Type: queryWithViewPortType

Operation: queryWithView

○ Arguments Mapping ● Input Variable

Input: Invoke1_queryWithView_InputVariable

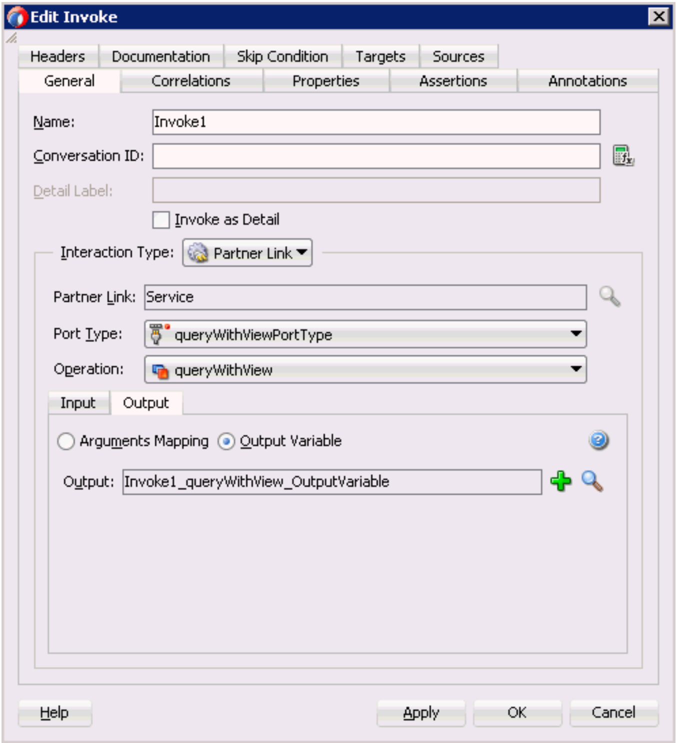
17. Select the **Output** tab and click the **Plus sign** icon, which is located to the right of the Output field to configure a new output variable.

    The Create Variable dialog is displayed.

18. Accept the default values that are provided for the new output variable and click **OK**.

    You are returned to the Edit Invoke dialog, as shown in Figure 4–29.

Figure 4–29 Edit Invoke Dialog



Name: Invoke1

Conversation ID:

Detail Label:

☐ Invoke as Detail

Interaction Type: Partner Link

Partner Link: Service

Port Type: queryWithViewPortType

Operation: queryWithView

Input | Output

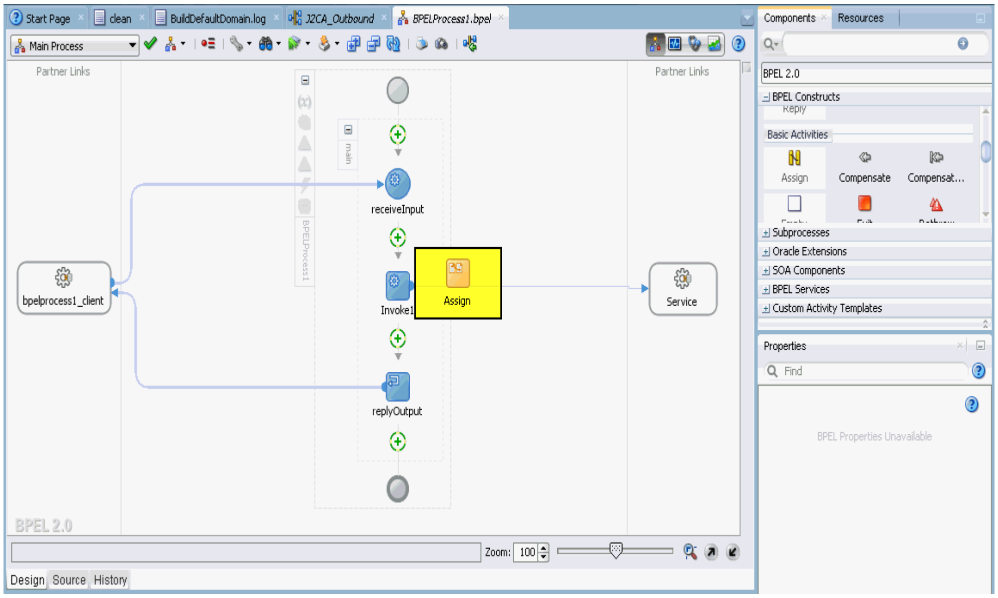
○ Arguments Mapping ● Output Variable

Output: Invoke1_queryWithView_OutputVariable

19. Click **Apply** and then **OK**.

20. Drag and drop the **Assign** activity under BPEL Constructs - Basic Activities component, to the Components pane and place it between the Receive activity component (receiveInput) and the Invoke activity component (Invoke1), as shown in Figure 4–30.

Figure 4–30 Assign Activity Component



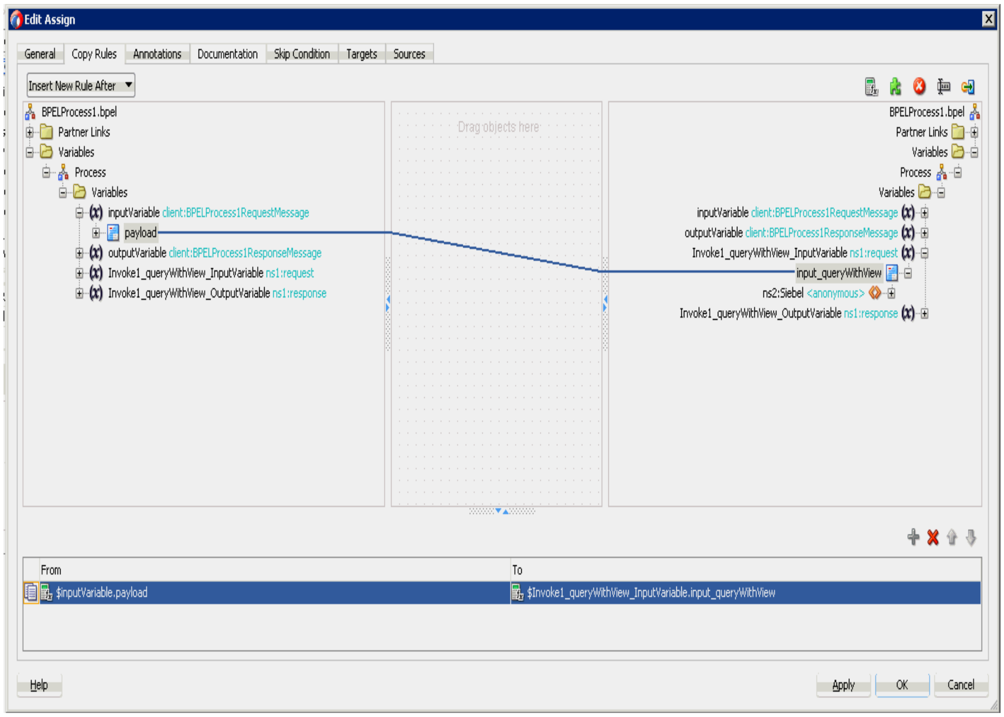
21. Double-click the new Assign activity component (**Assign1**).

The Edit Assign dialog is displayed.

22. In the left pane, under Variables, expand **InputVariable**, and then select **payload**.
23. In the right pane, under Variables, expand **Invoke1_queryWithView_InputVariable**, and then select **input_queryWithView**.
24. Drag and map the **payload** variable to the **input_queryWithView** variable.

    The mapped variables are populated in the highlighted area as shown in Figure 4–31.

Figure 4–31 Edit Assign Dialog

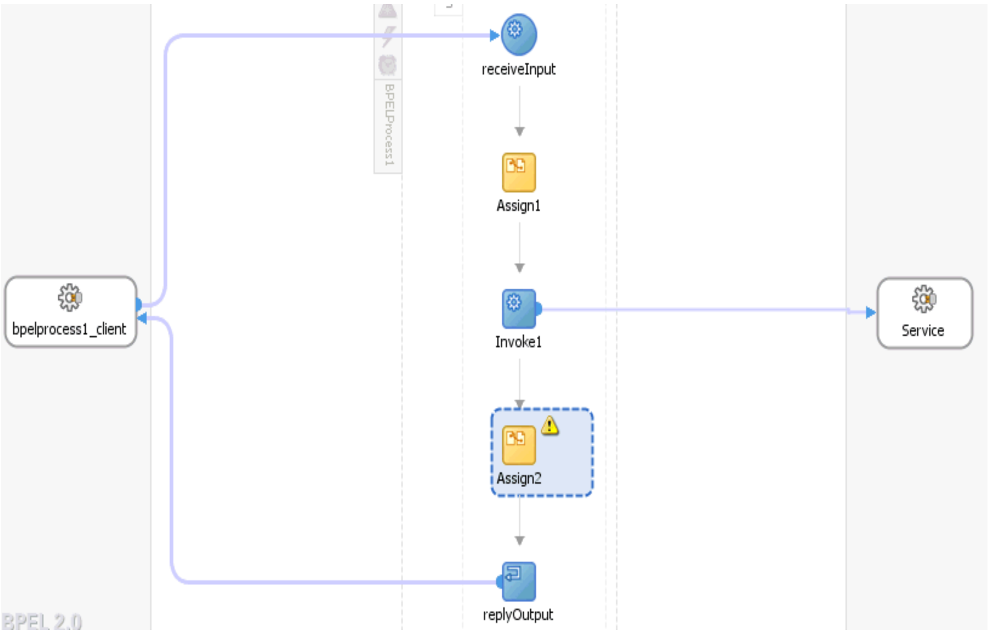


Edit Assign
General
Copy Rules
Annotations
Documentation
Skip Condition
Targets
Sources
Insert New Rule After
BPELProcess1.bpel
Partner Links
Variables
Process
Variables
inputVariable client:BPELProcess1RequestMessage
payload
outputVariable client:BPELProcess1ResponseMessage
Invoke1_queryWithView_InputVariable ns1:request
Invoke1_queryWithView_OutputVariable ns1:response
Drag objects here
BPELProcess1.bpel
Partner Links
Variables
Process
Variables
inputVariable client:BPELProcess1RequestMessage
outputVariable client:BPELProcess1ResponseMessage
Invoke1_queryWithView_InputVariable ns1:request
input_queryWithView
ns2:Siebel <anonymous>
Invoke1_queryWithView_OutputVariable ns1:response
Help
Apply
OK
Cancel

From	To
$inputVariable.payload	$Invoke1_queryWithView_InputVariable.input_queryWithView

25. Click **Apply** and then **OK**.
26. Drag and drop the **Assign** activity component to the Components pane and place it between the Invoke activity (Invoke1) and the Reply activity (replyOutput).
27. Double-click the new Assign activity component (**Assign2**), as shown in Figure 4–32.

Figure 4–32 New Assign Activity Component



The Edit Assign dialog is displayed.

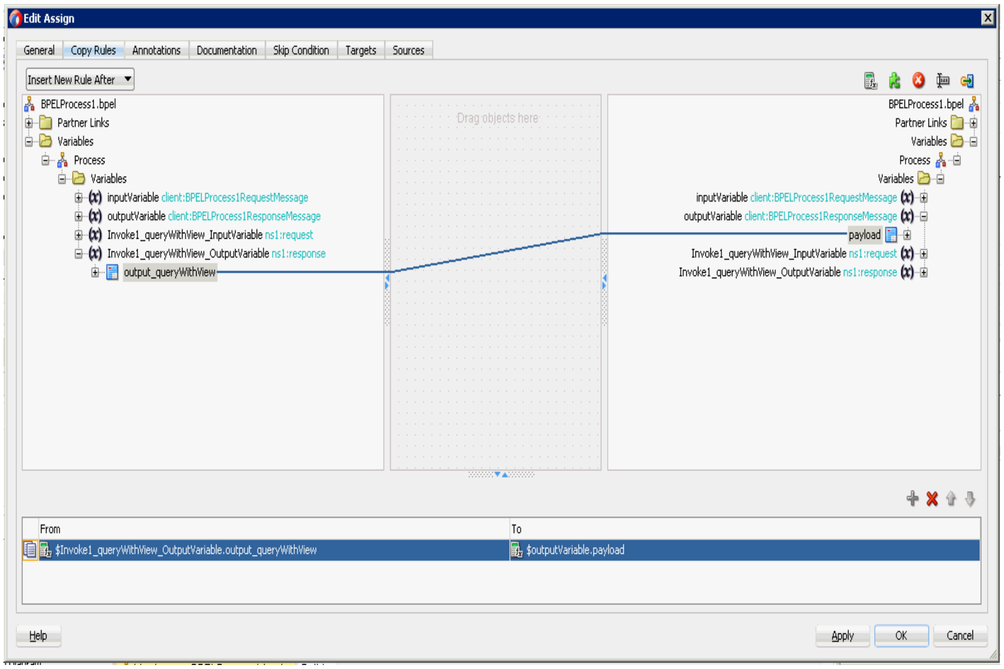
28. In the left pane, under Variables, expand **Invoke1_queryWithView_OutputVariable**, and then select **output_queryWithView**.

29. In the right pane, under Variables, expand **outputVariable** and select **payload**.

30. Drag and map the **output_queryWithView** variable to the **payload** variable.

    The mapped variables are populated in the highlighted area as shown in Figure 4–33.

Figure 4–33 Edit Assign Dialog

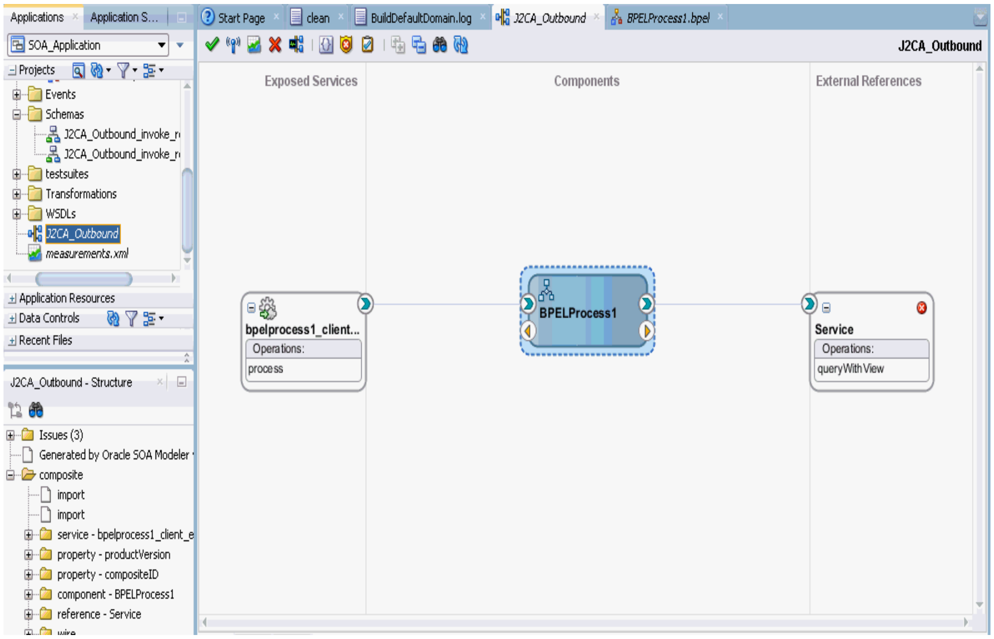

Edit Assign
General
Copy Rules
Annotations
Documentation
Skip Condition
Targets
Sources
Insert New Rule After
BPELProcess1.bpel
Partner Links
Variables
Process
Variables
inputVariable client:BPELProcess1RequestMessage
outputVariable client:BPELProcess1ResponseMessage
Invoke1_queryWithView_InputVariable ns1:request
Invoke1_queryWithView_OutputVariable ns1:response
output_queryWithView
Drag objects here
BPELProcess1.bpel
Partner Links
Variables
Process
Variables
inputVariable client:BPELProcess1RequestMessage
outputVariable client:BPELProcess1ResponseMessage
payload
Invoke1_queryWithView_InputVariable ns1:request
Invoke1_queryWithView_OutputVariable ns1:response
Help
Apply
OK
Cancel

From	To
$Invoke1_queryWithView_OutputVariable.output_queryWithView	$outputVariable.payload

31. Click **Apply** and then **OK**.

    You are returned to the Activity component pane, as shown in Figure 4–34.

Figure 4–34 Activity Component Pane



32. Click the **Save All** icon in the menu bar to save the new outbound BPEL process component that was configured.

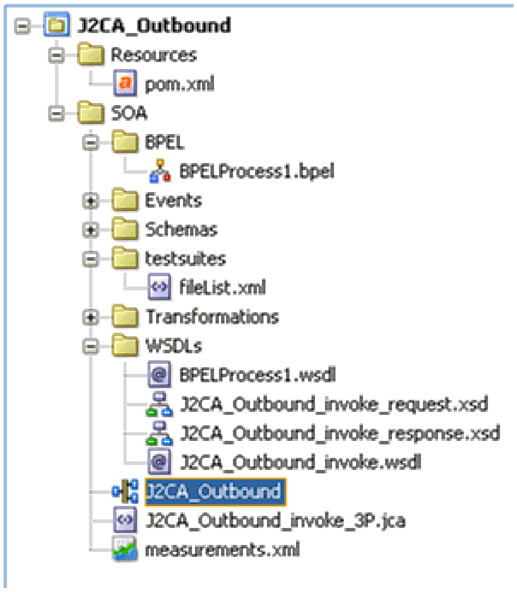
    You are now ready to deploy the BPEL outbound process.

4.4.3.3 Adjusting for Known Deployment Issues With 12c

Perform the following steps to adjust for known deployment issues with 12c.

1. Double-click **J2CA_Outbound** (created BPEL process) of the created process, as shown in Figure 4–35.

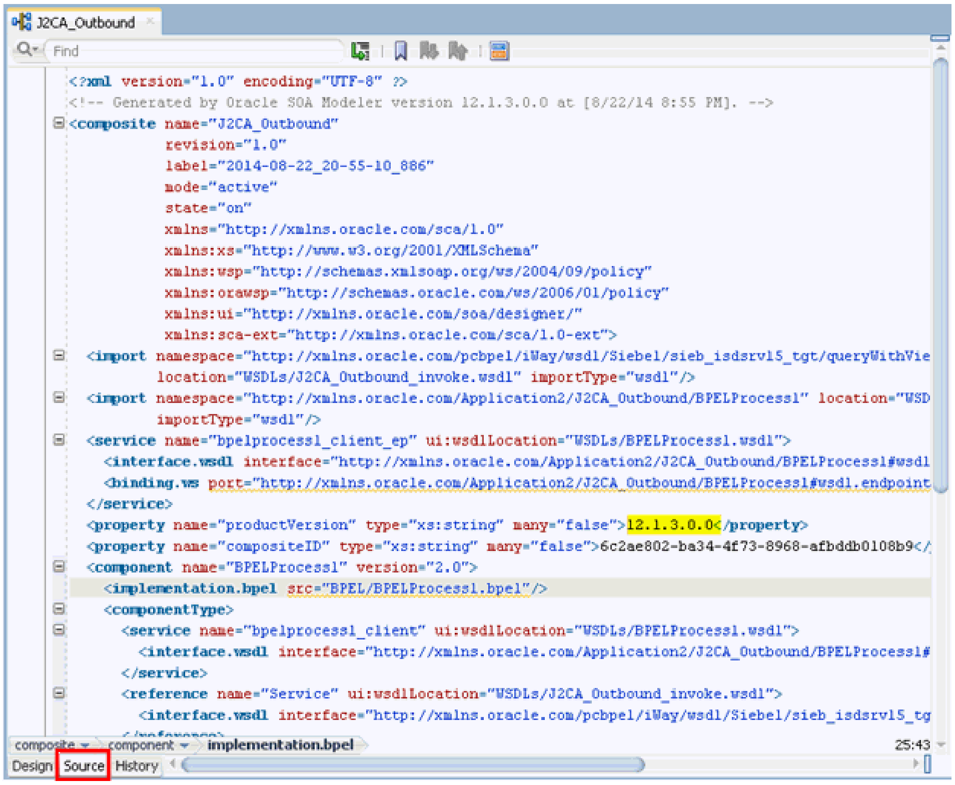
Figure 4–35 J2CA_Outbound Node



```
J2CA_Outbound
    Resources
        pom.xml
    SOA
        BPEL
            BPELProcess1.bpel
        Events
        Schemas
        testsuites
            fileList.xml
        Transformations
        WSDLs
            BPELProcess1.wsdl
            J2CA_Outbound_invoke_request.xsd
            J2CA_Outbound_invoke_response.xsd
            J2CA_Outbound_invoke.wsdl
        J2CA_Outbound
        J2CA_Outbound_invoke_3P.jca
        measurements.xml
```

2. Click the **Source** tab below the opened process, as shown in Figure 4–36.

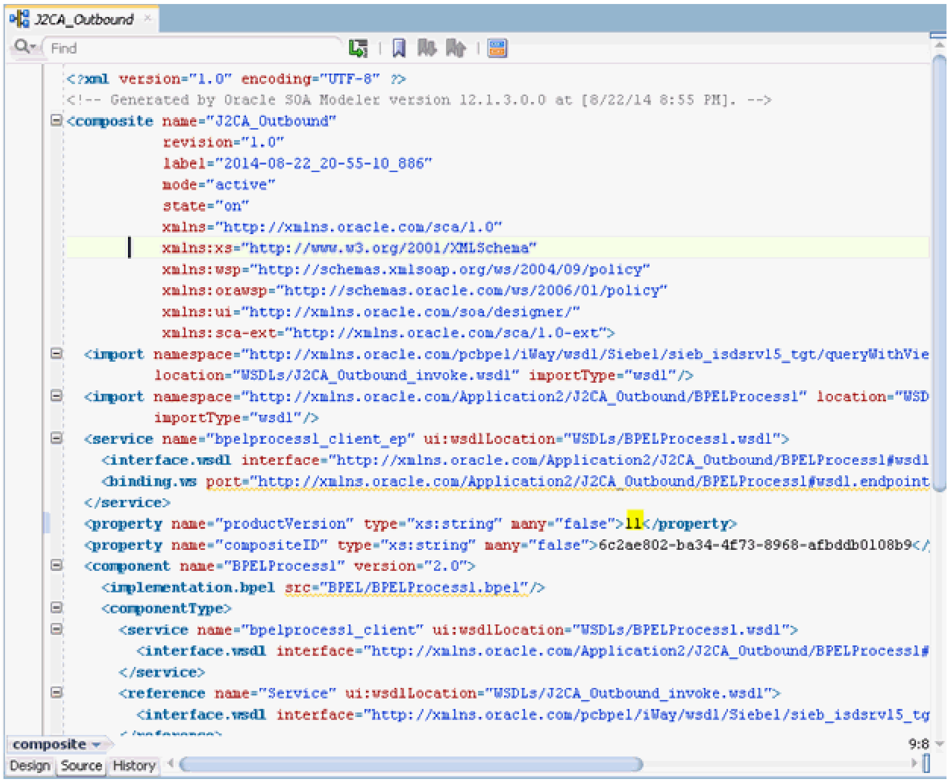
Figure 4–36 Source Tab



```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- Generated by Oracle SOA Modeler version 12.1.3.0.0 at [8/22/14 8:55 PM]. -->
<composite name="J2CA_Outbound"
           revision="1.0"
           label="2014-08-22_20-55-10_886"
           mode="active"
           state="on"
           xmlns="http://xmlns.oracle.com/sca/1.0"
           xmlns:xs="http://www.w3.org/2001/XMLSchema"
           xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"
           xmlns:orawsp="http://schemas.oracle.com/ws/2006/01/policy"
           xmlns:ui="http://xmlns.oracle.com/soa/designer/"
           xmlns:sca-ext="http://xmlns.oracle.com/sca/1.0-ext">
  <import namespace="http://xmlns.oracle.com/pcbpel/iWay/wsdl/Siebel/sieb_isdsrv15_tgt/queryWithVie
          location="WSDLs/J2CA_Outbound_invoke.wsdl" importType="wsdl"/>
  <import namespace="http://xmlns.oracle.com/Application2/J2CA_Outbound/BPELProcess1" location="WSD
          importType="wsdl"/>
  <service name="bpelprocess1_client_ep" ui:wsdlLocation="WSDLs/BPELProcess1.wsdl">
    <interface.wsdl interface="http://xmlns.oracle.com/Application2/J2CA_Outbound/BPELProcess1#wsdl
    <binding.ws port="http://xmlns.oracle.com/Application2/J2CA_Outbound/BPELProcess1#wsdl.endpoint
  </service>
  <property name="productVersion" type="xs:string" many="false">12.1.3.0.0</property>
  <property name="compositeID" type="xs:string" many="false">6c2ae802-ba34-4f73-8968-afbddb0108b9</
  <component name="BPELProcess1" version="2.0">
    <implementation.bpel src="BPEL/BPELProcess1.bpel"/>
    <componentType>
      <service name="bpelprocess1_client" ui:wsdlLocation="WSDLs/BPELProcess1.wsdl">
        <interface.wsdl interface="http://xmlns.oracle.com/Application2/J2CA_Outbound/BPELProcess1#
      </service>
      <reference name="Service" ui:wsdlLocation="WSDLs/J2CA_Outbound_invoke.wsdl">
        <interface.wsdl interface="http://xmlns.oracle.com/pcbpel/iWay/wsdl/Siebel/sieb_isdsrv15_tg
```

3. Change the productVersion property value from `12.1.3.0.0` to `11`, as shown in Figure 4–37.

Figure 4–37 Property Value



```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- Generated by Oracle SOA Modeler version 12.1.3.0.0 at [8/22/14 8:55 PM]. -->
<composite name="J2CA_Outbound"
           revision="1.0"
           label="2014-08-22_20-55-10_886"
           mode="active"
           state="on"
           xmlns="http://xmlns.oracle.com/sca/1.0"
           xmlns:xs="http://www.w3.org/2001/XMLSchema"
           xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"
           xmlns:orawsp="http://schemas.oracle.com/ws/2006/01/policy"
           xmlns:ui="http://xmlns.oracle.com/soa/designer/"
           xmlns:sca-ext="http://xmlns.oracle.com/sca/1.0-ext">
  <import namespace="http://xmlns.oracle.com/pcbpel/iWay/wsdl/Siebel/sieb_isdsrv15_tgt/queryWithVie
          location="WSDLs/J2CA_Outbound_invoke.wsdl" importType="wsdl"/>
  <import namespace="http://xmlns.oracle.com/Application2/J2CA_Outbound/BPELProcess1" location="WSD
          importType="wsdl"/>
  <service name="bpelprocess1_client_ep" ui:wsdlLocation="WSDLs/BPELProcess1.wsdl">
    <interface.wsdl interface="http://xmlns.oracle.com/Application2/J2CA_Outbound/BPELProcess1#wsdl
    <binding.ws port="http://xmlns.oracle.com/Application2/J2CA_Outbound/BPELProcess1#wsdl.endpoint
  </service>
  <property name="productVersion" type="xs:string" many="false">11</property>
  <property name="compositeID" type="xs:string" many="false">6c2ae802-ba34-4f73-8968-afbddb0108b9</
  <component name="BPELProcess1" version="2.0">
    <implementation.bpel src="BPEL/BPELProcess1.bpel"/>
    <componentType>
      <service name="bpelprocess1_client" ui:wsdlLocation="WSDLs/BPELProcess1.wsdl">
        <interface.wsdl interface="http://xmlns.oracle.com/Application2/J2CA_Outbound/BPELProcess1#
      </service>
      <reference name="Service" ui:wsdlLocation="WSDLs/J2CA_Outbound_invoke.wsdl">
        <interface.wsdl interface="http://xmlns.oracle.com/pcbpel/iWay/wsdl/Siebel/sieb_isdsrv15_tg
```

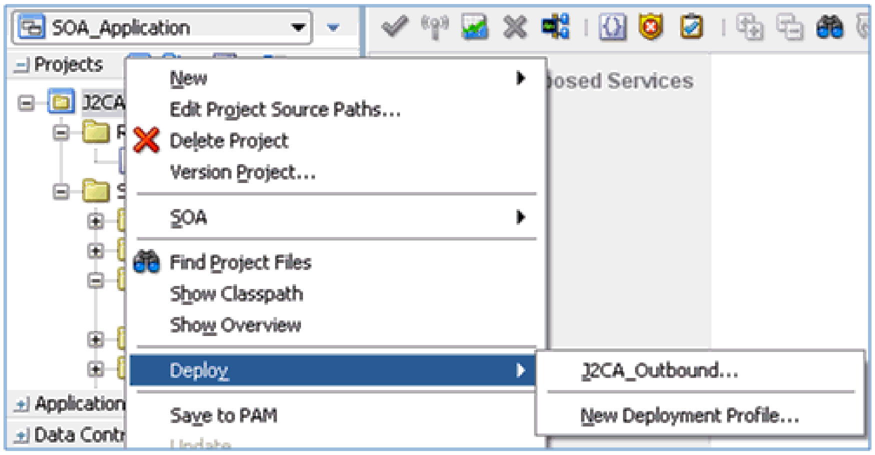
4. Save the changes and proceed to deploy the project.

4.4.4 Deploying the BPEL Outbound Process

Perform the following steps to deploy the BPEL outbound process.

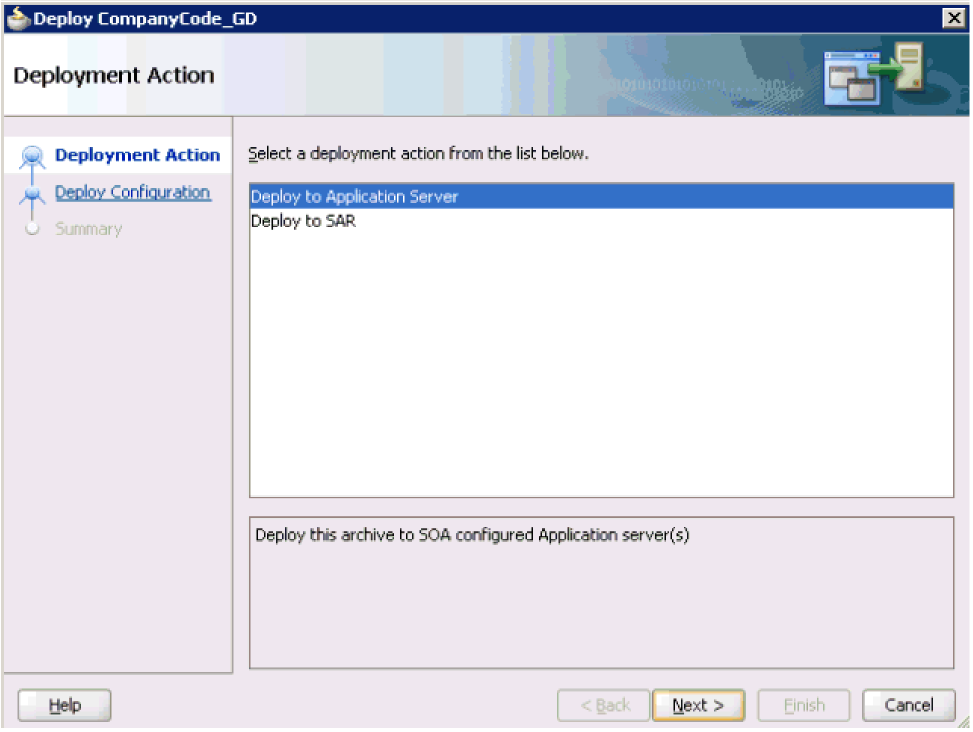
1. Right-click the project name in the left pane, select **Deploy**, and then click **J2CA_Outbound**, as shown in Figure 4–38.

Figure 4–38 J2CA_Outbound Option



The Deployment Action page is displayed, as shown in Figure 4–39.

Figure 4–39 Deployment Action Page



Select a deployment action from the list below.

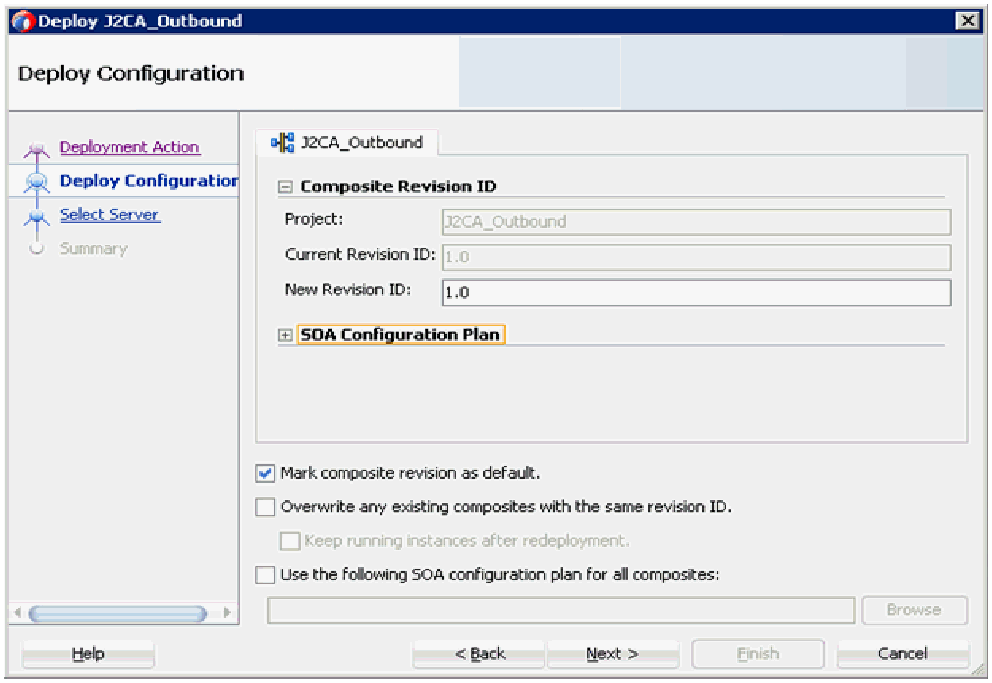
- Deploy to Application Server
- Deploy to SAR

Deploy this archive to SOA configured Application server(s)

2. Ensure that **Deploy to Application Server** is selected.
3. Click **Next**.

   The Deploy Configuration page is displayed, as shown in Figure 4–40.

Figure 4–40 Deploy Configurations Page



Project: J2CA_Outbound

Current Revision ID: 1.0

New Revision ID: 1.0

☑ Mark composite revision as default.

□ Overwrite any existing composites with the same revision ID.

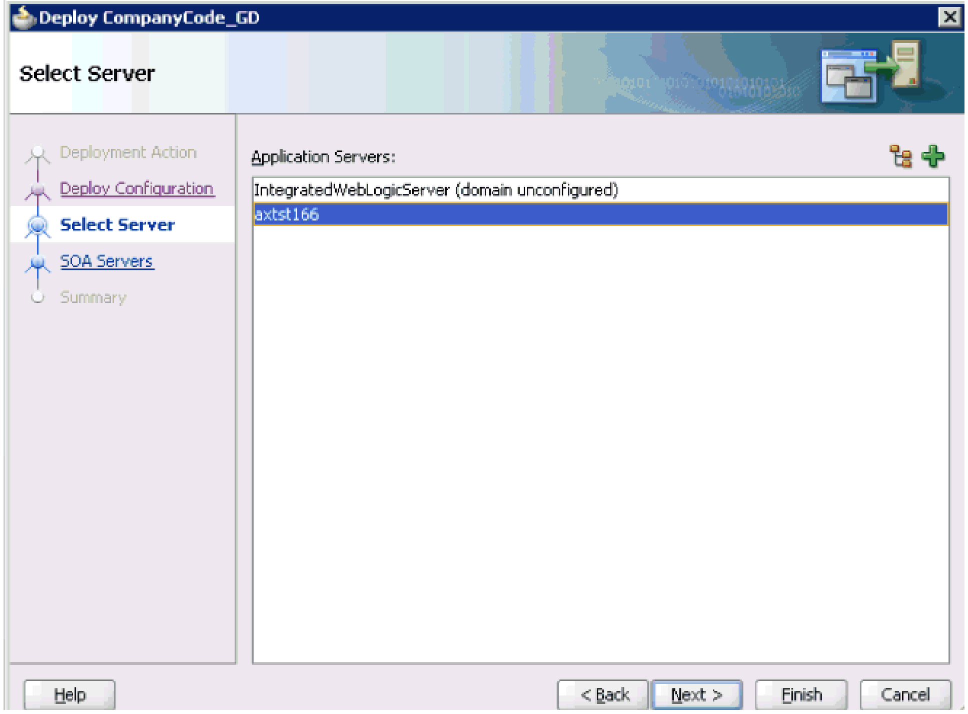
□ Keep running instances after redeployment.

□ Use the following SOA configuration plan for all composites:

4. Leave the default values selected and click **Next**.

The Select Server page is displayed, as shown in Figure 4–41.

Figure 4–41 Select Server Page



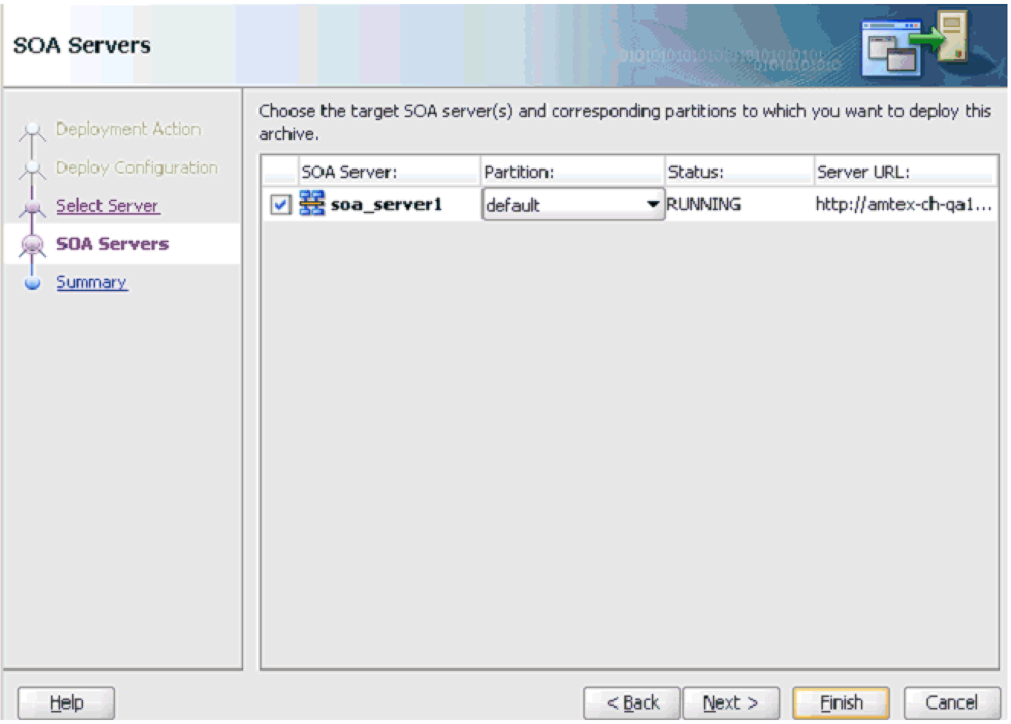
Application Servers:

IntegratedWebLogicServer (domain unconfigured)
axtst166

5. Select an available application server that was configured and click **Next**.

   The SOA Servers page is displayed, as shown in Figure 4–42.

Figure 4–42 SOA Servers Page



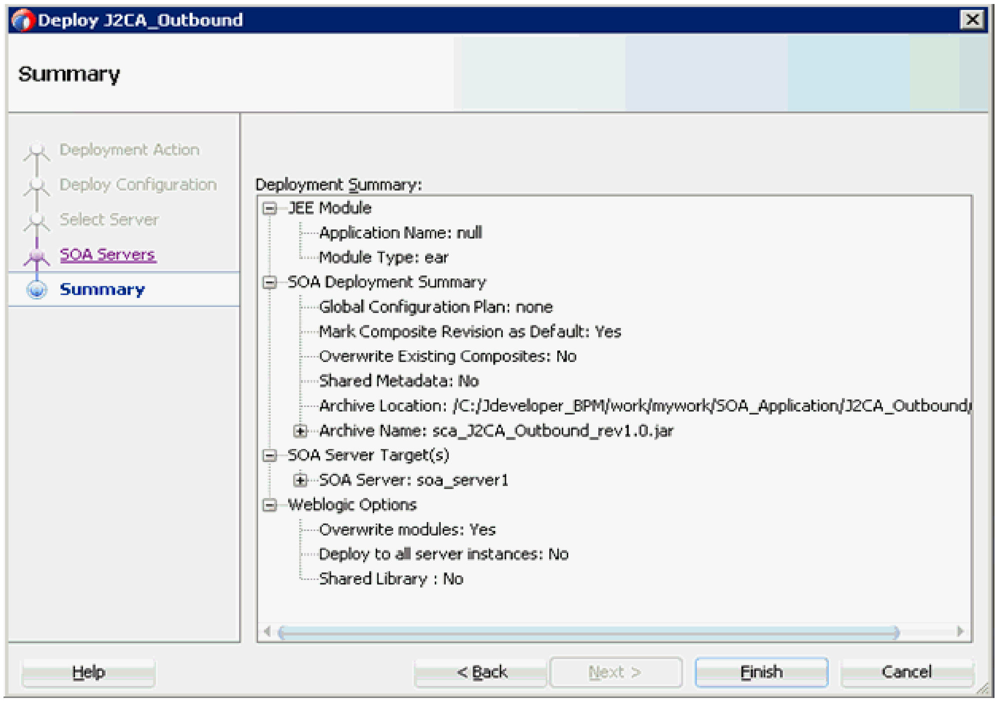
Choose the target SOA server(s) and corresponding partitions to which you want to deploy this archive.

	SOA Server:	Partition:	Status:	Server URL:
☑	soa_server1	default	RUNNING	http://amtex-ch-qa1...

6. Select a target SOA server and click **Next**.

   The Summary page is displayed, as shown in Figure 4–43.

Figure 4–43 Summary Page



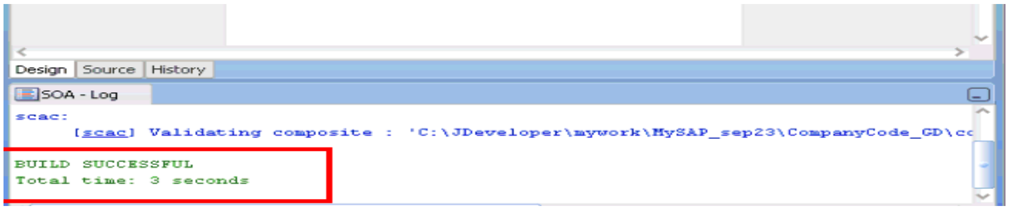
Deployment Summary:

```
⊟ JEE Module
    Application Name: null
    Module Type: ear
⊟ SOA Deployment Summary
    Global Configuration Plan: none
    Mark Composite Revision as Default: Yes
    Overwrite Existing Composites: No
    Shared Metadata: No
    Archive Location: /C:/Jdeveloper_BPM/work/mywork/SOA_Application/J2CA_Outbound/
  ⊞ Archive Name: sca_J2CA_Outbound_rev1.0.jar
⊟ SOA Server Target(s)
  ⊞ SOA Server: soa_server1
⊟ Weblogic Options
    Overwrite modules: Yes
    Deploy to all server instances: No
    Shared Library : No
```

7. Review and verify all the available deployment information for your project and click **Finish**.

   The process is deployed successfully, as shown in Figure 4–44.

Figure 4–44 Successful Deployment Message



```
scac:
     [scac] Validating composite : 'C:\JDeveloper\mywork\MySAP_sep23\CompanyCode_GD\cc
BUILD SUCCESSFUL
Total time: 3 seconds
```

4.4.5 Invoking the Input XML Document in the Oracle Enterprise Manager Console

Perform the following steps to invoke the input XML document in the Oracle Enterprise Manager console.

1. Logon to the Oracle Enterprise Manager console.

Note: For customers using 12*c* (12.2.1.1.0) and 12c (12.2.1.2.0), perform the following steps:

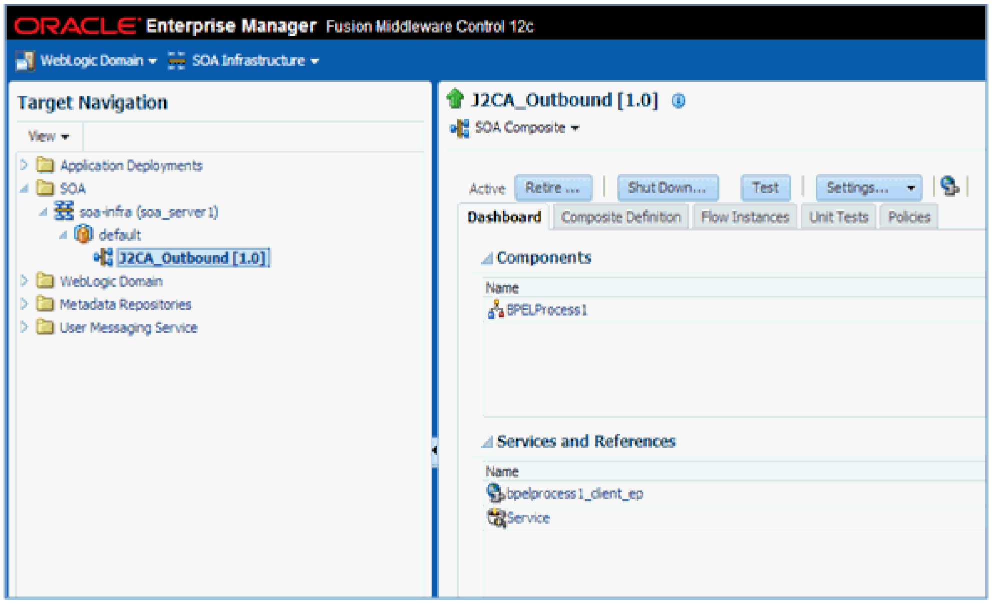
2. Click **Target Navigation** in the left pane, expand **SOA**, and then select **soa-infra (soa_server1)**.

3. Click the **Deployed Composites** tab, which will list all of the deployed composites. Click on the available project (for example, J2CA_Outbound).

Skip to **Step 4** in this procedure.

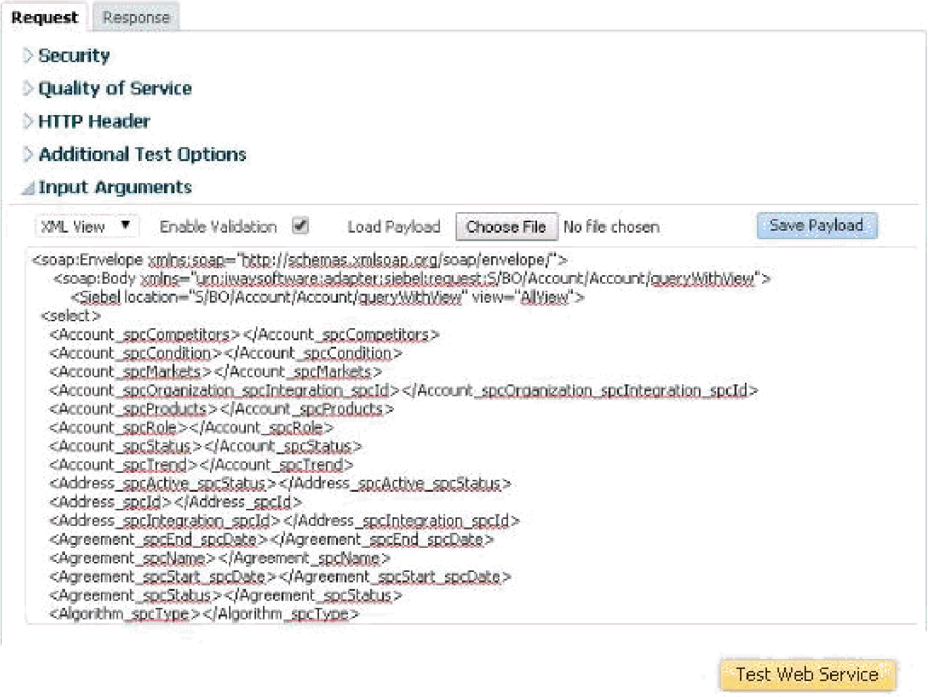
2. Expand **SOA**, select **soa-infra (soa_server1)**, and then click **Default**.
3. Select an available project (for example, J2CA_Outbound) and click **Test** as shown in Figure 4–45.

Figure 4–45 Test Button



4. Click the **Request** tab.
5. Select **XML View** from the list, as shown in Figure 4–46.

Figure 4–46 Input Arguments List

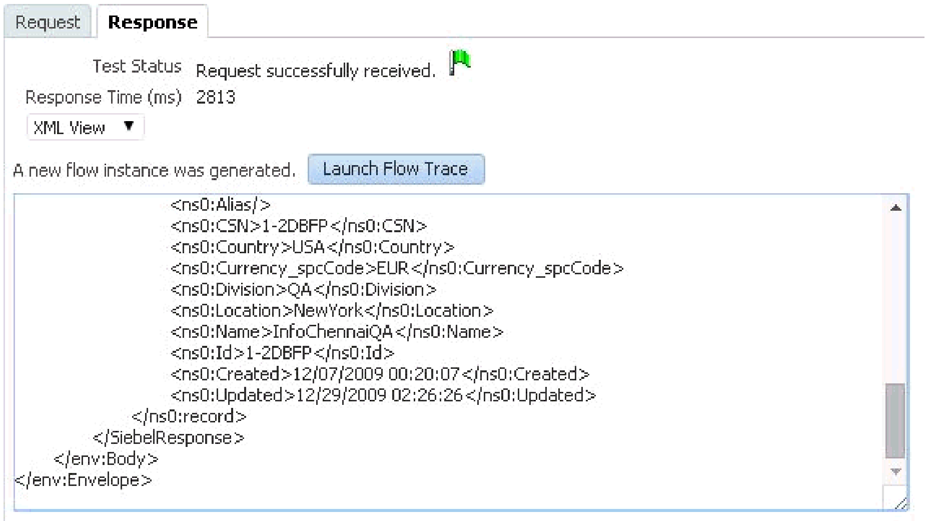


```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body xmlns="urn:iwaysoftware:adapter:siebel:request:S/BO/Account/Account/queryWithView">
      <Siebel location="S/BO/Account/Account/queryWithView" view="AllView">
 <select>
  <Account_spcCompetitors></Account_spcCompetitors>
  <Account_spcCondition></Account_spcCondition>
  <Account_spcMarkets></Account_spcMarkets>
  <Account_spcOrganization_spcIntegration_spcId></Account_spcOrganization_spcIntegration_spcId>
  <Account_spcProducts></Account_spcProducts>
  <Account_spcRole></Account_spcRole>
  <Account_spcStatus></Account_spcStatus>
  <Account_spcTrend></Account_spcTrend>
  <Address_spcActive_spcStatus></Address_spcActive_spcStatus>
  <Address_spcId></Address_spcId>
  <Address_spcIntegration_spcId></Address_spcIntegration_spcId>
  <Agreement_spcEnd_spcDate></Agreement_spcEnd_spcDate>
  <Agreement_spcName></Agreement_spcName>
  <Agreement_spcStart_spcDate></Agreement_spcStart_spcDate>
  <Agreement_spcStatus></Agreement_spcStatus>
  <Algorithm_spcType></Algorithm_spcType>
```

6. Provide an appropriate input XML document in the Input Arguments area and click **Test Web Service**.

   The output response is received in the Oracle Enterprise Manager console, as shown in Figure 4–47.

Figure 4–47 Received Output Response



Test Status Request successfully received.

Response Time (ms) 2813

A new flow instance was generated.

```
                    <ns0:Alias/>
                    <ns0:CSN>1-2DBFP</ns0:CSN>
                    <ns0:Country>USA</ns0:Country>
                    <ns0:Currency_spcCode>EUR</ns0:Currency_spcCode>
                    <ns0:Division>QA</ns0:Division>
                    <ns0:Location>NewYork</ns0:Location>
                    <ns0:Name>InfoChennaiQA</ns0:Name>
                    <ns0:Id>1-2DBFP</ns0:Id>
                    <ns0:Created>12/07/2009 00:20:07</ns0:Created>
                    <ns0:Updated>12/29/2009 02:26:26</ns0:Updated>
               </ns0:record>
          </SiebelResponse>
     </env:Body>
</env:Envelope>
```

4.4.6 Testing Outbound BPEL and Mediator Processes

When testing an outbound BPEL process or an outbound Mediator process from the Oracle Enterprise Manager console, do not use the XML envelopes that are generated by these consoles. Instead, remove them and use the XML payloads that are generated from the schemas, which conform to the WSDLs for namespace qualifications.

The Mediator data flows can be tested using the Enterprise Manager console. When creating a Mediator data flow and interactions, the Web services are created and registered with the Oracle Application Server. For more information on creating a Mediator outbound process, see Chapter 5, "Integration With Mediator Service Components in the Oracle SOA Suite".

4.5 Designing an Inbound BPEL Process for Event Integration (J2CA Configuration)

This section describes Siebel event integration.

A sample project has been provided for this inbound use case scenario in the following folder of the Application Adapters installation:

```
<ADAPTER_HOME>\etc\sample\SIEBEL_Samples.zip\SIEBEL_Samples\BPEL\J2CA\Inbound_
Project
```

The following tools are required to complete your adapter design-time configuration:

- Oracle Adapter Application Explorer (Application Explorer)
- Oracle JDeveloper BPEL Designer (JDeveloper)

Note: The examples in this chapter demonstrate the use of Oracle JDeveloper.

This section contains the following topics:

- Section 4.5.1, "Generating WSDL for Event Integration"
- Section 4.5.2, "Creating an Empty Composite for SOA"
- Section 4.5.3, "Defining a BPEL Inbound Process"
- Section 4.5.4, "Deploying the BPEL Inbound Process"
- Section 4.5.5, "Triggering an Event in Siebel"

Before you design a BPEL process, you must generate the respective WSDL file using Application Explorer. For more information, see "Generating WSDL for Event Integration" on page 4-34.

4.5.1 Generating WSDL for Event Integration

You must create a separate channel for every inbound J2CA service and select that channel when you generate WSDL for inbound interaction using Application Explorer.

Note: If two or more events share the same channel, then event messages may not be delivered to the right BPEL process.

This section contains the following topics:

- Section 4.5.1.1, "Creating a Channel"
- Section 4.5.1.2, "Creating an Integration Object Node"

- Section 4.5.1.3, "Generating WSDL for Event Notification"

4.5.1.1 Creating a Channel

You must create a separate channel for every inbound J2CA service and select that channel when you generate WSDL for inbound interaction using Application Explorer.

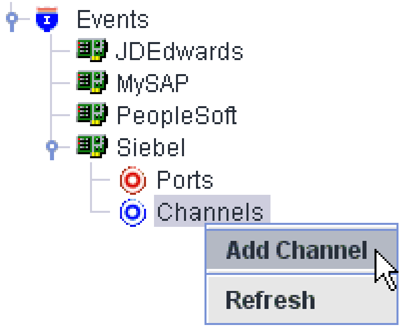
> **Note:** If two or more events share the same channel, then event messages may not be delivered to the right BPEL process.

To create a channel:

1. In the left pane, click **Events**.
2. Expand the **Siebel** node.

   The ports and channels nodes appear in the left pane, as shown in Figure 4–48.

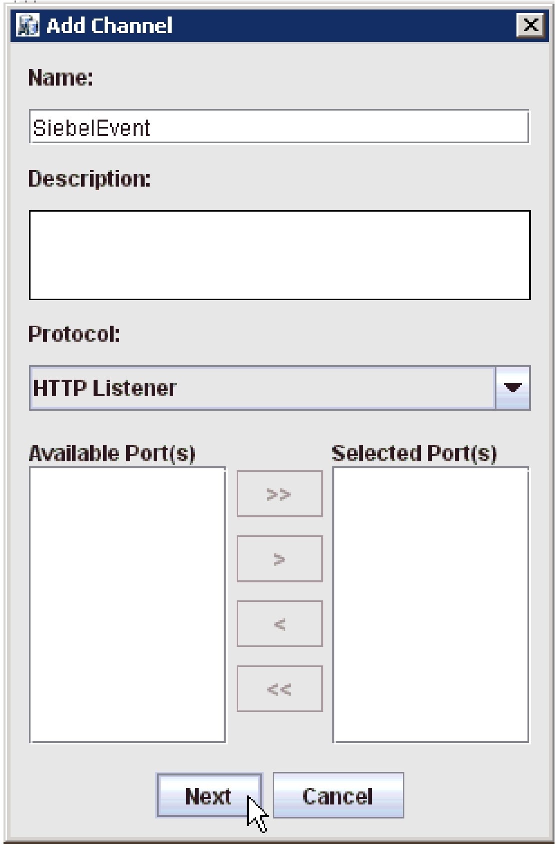
Figure 4–48 Ports and Channels Nodes Under Siebel Node



3. Right-click **Channels** and select **Add Channel**.

   The Add Channel dialog is displayed, as shown in Figure 4–49.

Figure 4–49 Add Channel Dialog



Add Channel

Name: SiebelEvent

Description:

Protocol: HTTP Listener

Perform the following steps:

a. Enter a name for the channel, for example, `SiebelEvent`.

b. Enter a brief description.

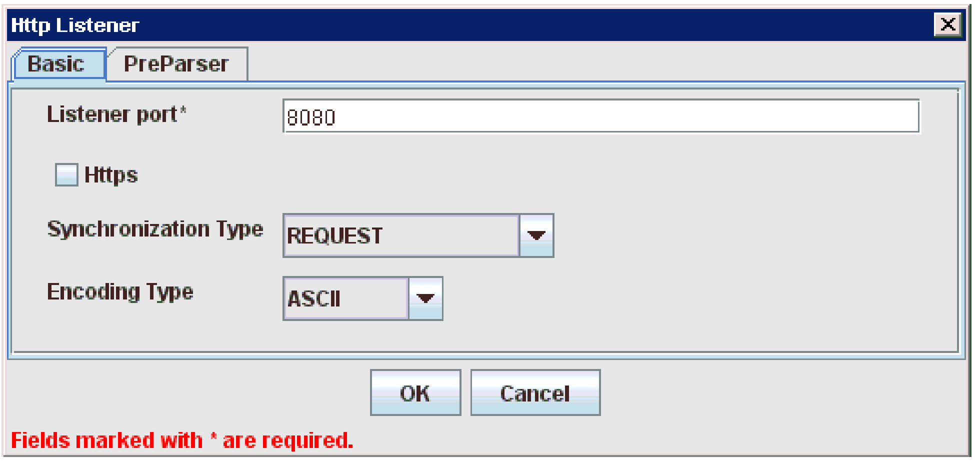
c. From the **Protocol** list, select **HTTP Listener**, **MQ Series Listener**, or **File Listener**.

For demonstration purposes, this procedure uses the HTTP Listener as an example.

4. Click **Next**.

The Basic dialog is displayed, as shown in Figure 4–50.

Figure 4–50 Basic Dialog



Listener port* 8080

□ Https

Synchronization Type REQUEST

Encoding Type ASCII

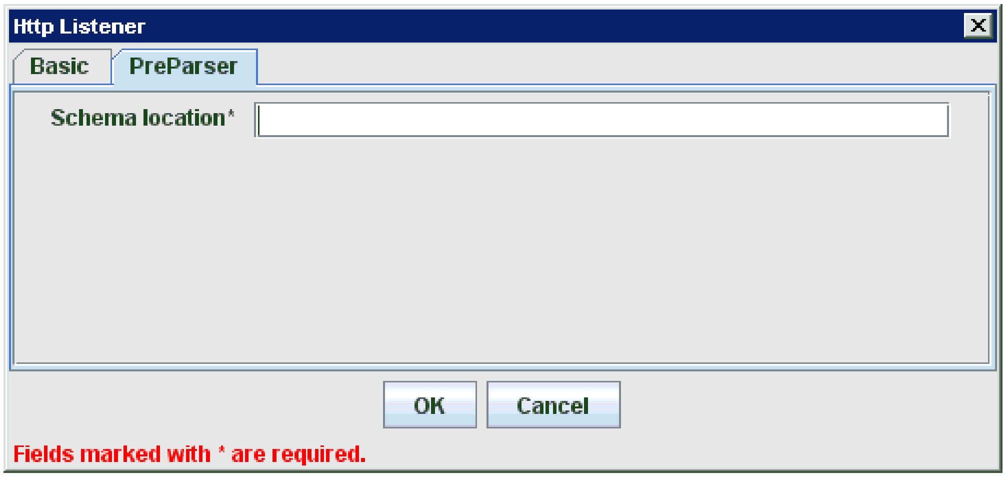
Fields marked with * are required.

5. Enter the system information as specified in the following table:

Parameter	Description
Listener port	Port on which to listen for Siebel event data.
Https	For a secure HTTP connection, select the **Https** check box. This option is currently not supported.
Synchronization Type	Select **REQUEST_RESPONSE** from the list, which is the recommended option.
Encoding Type	Choose an encoding type to be used from the list. By default, ASCII is selected.

6. Click the **PreParser** tab, as shown in Figure 4–51.

Figure 4–51 PreParser Tab



Schema location*

Fields marked with * are required.

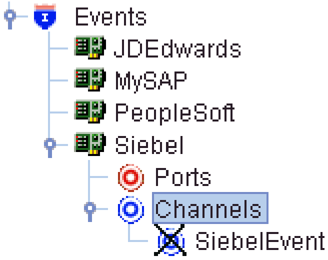
7. Specify the location of the schema file that was generated for the Integration Object node using the **Export Schema(s)** option in Application Explorer.

> **Note:** During run time, the Oracle Application Adapter for Siebel adds the namespace to the Siebel published document using the schema that is specified in the PreParser tab. If the Schema location field in the PreParser tab is empty, then BPEL and Mediator processes do not work properly as the Siebel published documents do not contain any namespaces.

8. Click **OK**.

   As shown in Figure 4–52, the channel is displayed under the channels node in the left pane. An X over the icon indicates that the channel is currently disconnected.

Figure 4–52 New Channel Node



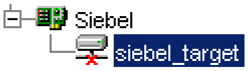
> **Note:** Do not start the channel, as it is managed by BPEL PM Server. If you start the channel for testing and debugging purposes, then stop it before run-time.

You must now create an Integration Object node.

4.5.1.2 Creating an Integration Object Node

1. Start Application Explorer.
2. Expand the **Adapters** node, as shown in Figure 4–53.

Figure 4–53 Disconnected Siebel Target Node, Siebel, Under the Siebel Node



   Perform the following steps:

   a. Expand the **Siebel** node.

      The defined Siebel targets are displayed under the adapter node.

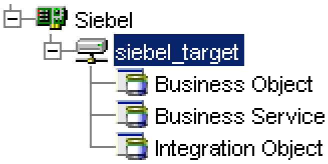
   b. Click the target name, for example, siebel, under the **Siebel** node.

   The Connection dialog displays the values you entered.

3. Verify your connection parameters.
4. Right-click the target name and select **Connect**.

   The x icon disappears, indicating that the node is connected, as shown in Figure 4–54.

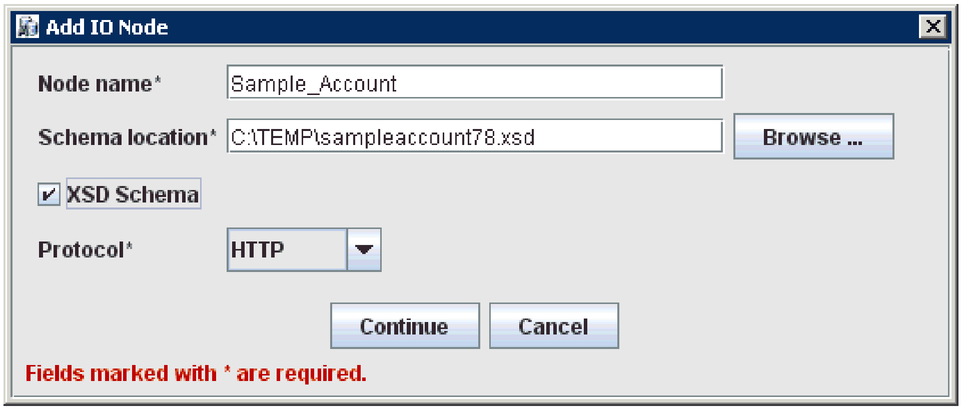
Figure 4–54 Connected Nodes



5. Expand the **Integration Object** node and select **Sample Account**.
6. Right-click the **Sample Account** node and select **Add IO Node**.

   The Add IO Node dialog is displayed, as shown in Figure 4–55.

Figure 4–55 Add IO Node Dialog


Add IO Node

Node name* Sample_Account

Schema location* C:\TEMP\sampleaccount78.xsd

☑ XSD Schema

Protocol* HTTP

Fields marked with * are required.

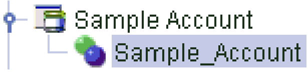
7. Enter a node name (for example, Sample_Account) in the **Node name** field and a path to the Sample Account XSD file in the **Schema location** field.

   Please note:

   - **For Siebel 7.5 or later:** Generate XSD schemas directly from Siebel tools. You use the XSD schemas when you create Web services in Application Explorer. After you generate an XSD schema through Siebel tools, use it to create an IO node and a Web service.
   - **For Siebel 7.0:** You cannot generate XSD schemas directly from Siebel tools; only XDR schemas can be created. Before you create a Web service, you must first generate an XSD schema from the XDR schema using Application Explorer.
8. If the XSD schema has already been generated, then select XSD Schema. If you are using Siebel-generated XDR schemas, then do not select the XSD schema option.
9. Select a protocol (HTTP, FILE, or MQ Series) from the **Protocol** list.
10. Click **Continue**.

    The new Integration Object node is added, as shown inFigure 4–56.

Figure 4–56 Integration Object Node



Note: You must restart the Oracle WebLogic Server after the Integration Object node and channel are created.

4.5.1.3 Generating WSDL for Event Notification

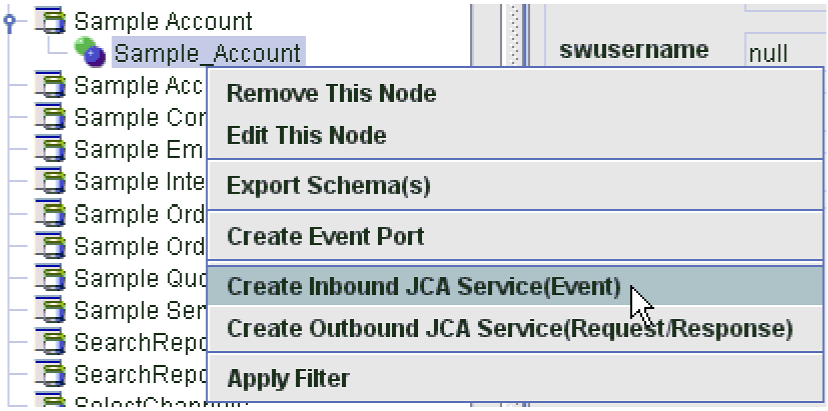
After you create a channel and an associated Integration Object node, you must generate WSDL for the event using Application Explorer.

You must be connected to a Siebel target under the Adapters node in Application Explorer. For detailed information on how to define and connect to a target, see "Establishing a Connection (Target) for Siebel" on page 2-5.

After you connect to a Siebel target, generate WSDL for the event as follows:

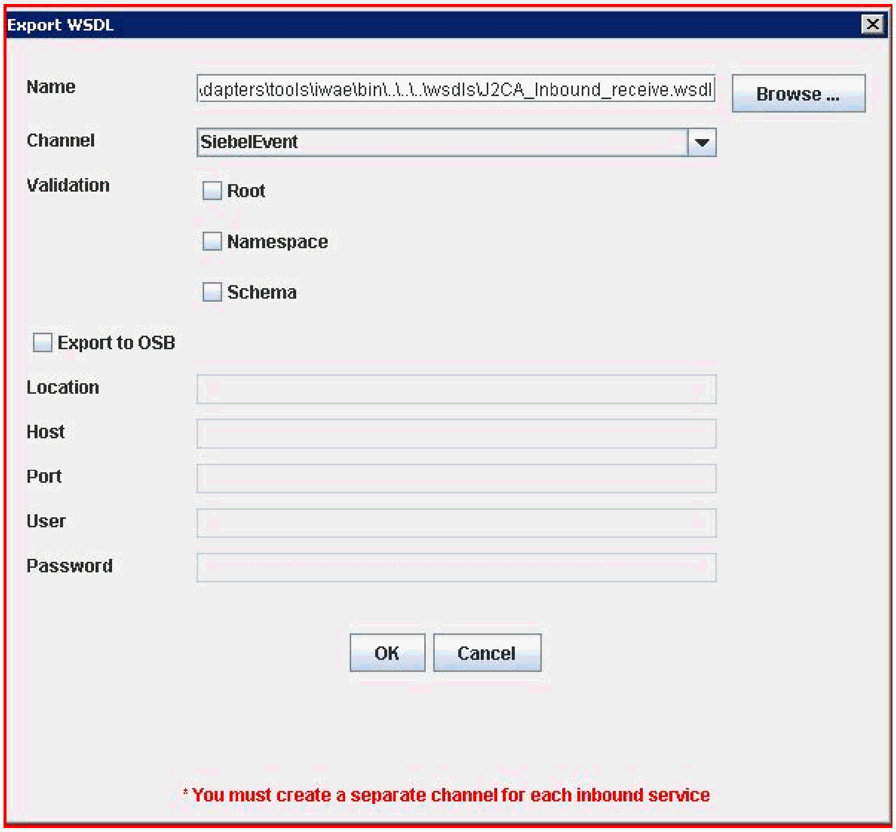
1. Right-click the Integration Object node (for example, Sample_Account), and then select **Create Inbound JCA Service (Event)**, as shown in Figure 4–57.

Figure 4–57 Create Inbound JCA Service (Event) Option Selected in Application Explorer



The Export WSDL dialog is displayed, as shown in Figure 4–58.

Figure 4–58 Export WSDL Dialog



Export WSDL

Name: \dapters\tools\iwae\bin\..\..\..\wsdls\J2CA_Inbound_receive.wsdl

Channel: SiebelEvent

Validation: □ Root □ Namespace □ Schema

□ Export to OSB

Location:

Host:

Port:

User:

Password:

* You must create a separate channel for each inbound service

---

Note: The schema validation options (Root, Namespace, Schema) are not applicable for the Oracle Application Adapter for Siebel.

Perform the following steps:

a. In the **Name** field, specify a name for the WSDL file.

   The .wsdl file extension is added automatically. By default, the names of WSDL files generated for events end with _receive.

b. From the Channel list, select the channel you created for this inbound service (for example, SiebelEvent).

   Important: You must create a separate channel for every inbound service. Verify that the channel is stopped before run-time.

2. Click **OK**.

4.5.2 Creating an Empty Composite for SOA

Perform the following steps to create an empty composite for SOA:

1. Create a new SOA application.
2. Enter a name for the new SOA Application and click **Next**.

   The Name your project page is displayed.
3. Enter a project name and click **Next**.

The Configure SOA settings page is displayed.

4. From the Composite Template list, select **Empty Composite** and click **Finish**.

   For more information, see Section 4.4.2, "Creating an Empty Composite for SOA" on page 4-9.

4.5.3 Defining a BPEL Inbound Process

This section describes how to define a BPEL inbound process, which consists of the following topics:

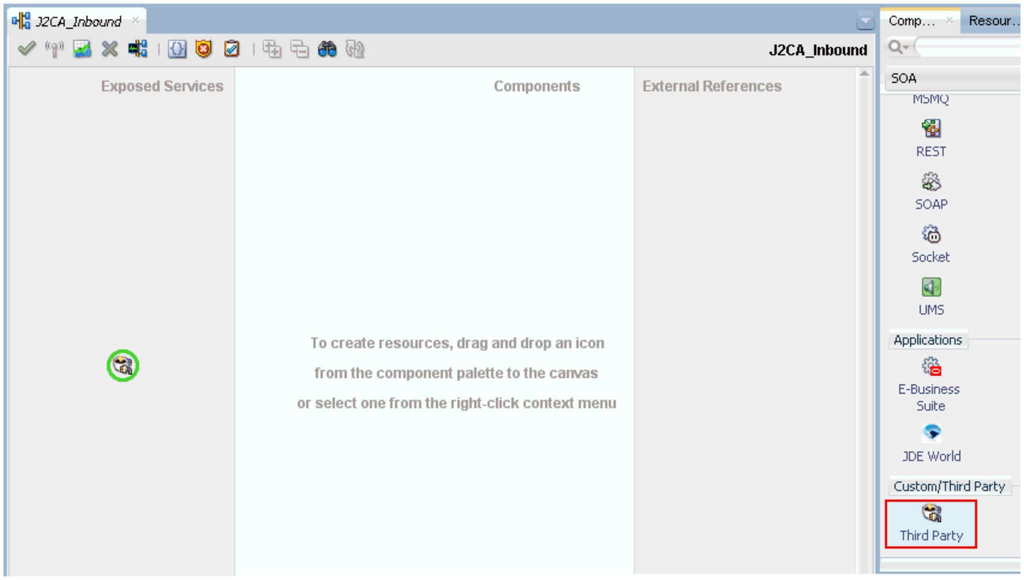
- Section 4.5.3.1, "Creating a Third Party Adapter Service Component"
- Section 4.5.3.2, "Creating an Inbound BPEL Process Component"

4.5.3.1 Creating a Third Party Adapter Service Component

Perform the following steps to create a third party adapter service component:

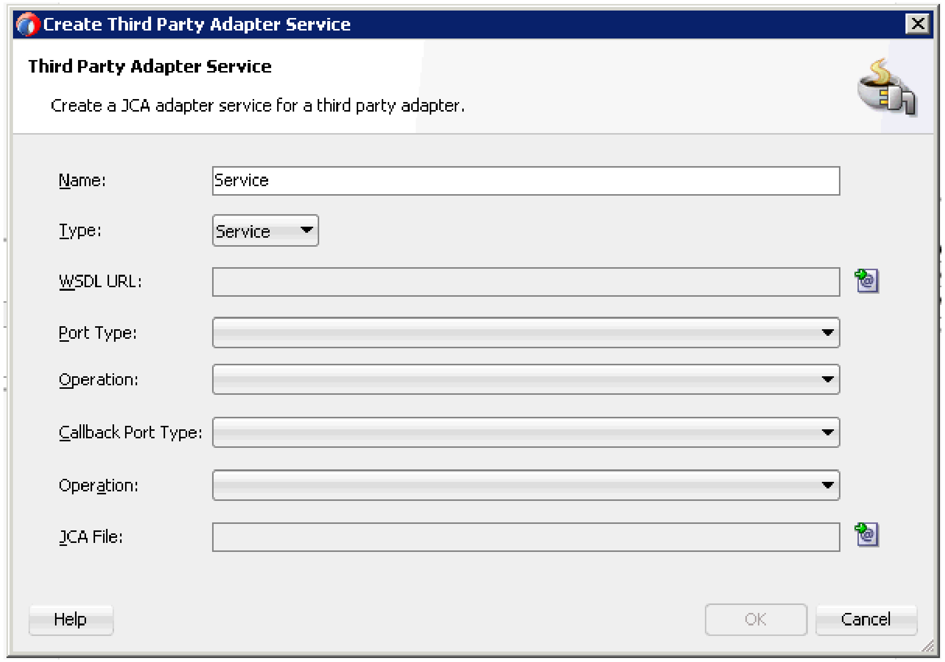
1. Drag and drop the **Third Party Adapter** component from the Service Adapters pane to the Exposed Services pane, as shown in Figure 4–59.

Figure 4–59 Third Party Adapter Component



The Create Third Party Adapter Service dialog is displayed, as shown in Figure 4–60.

Figure 4–60 Create Third Party Adapter Service Dialog



Create Third Party Adapter Service

Third Party Adapter Service

Create a JCA adapter service for a third party adapter.

Name: Service

Type: Service

WSDL URL:

Port Type:

Operation:

Callback Port Type:

Operation:

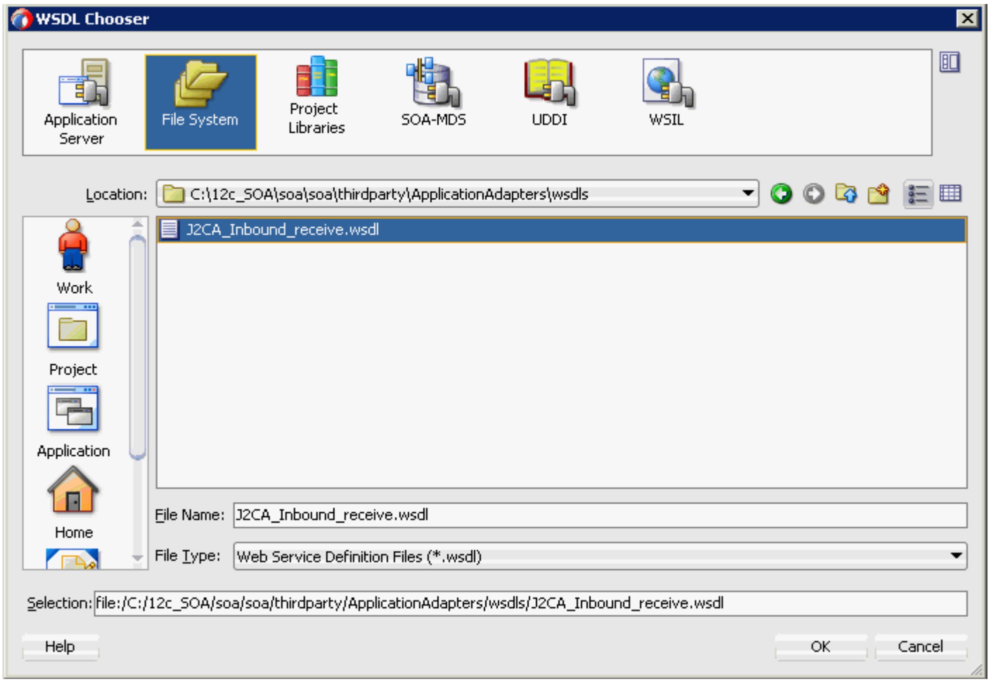
JCA File:

2. Ensure that **Service** is selected from the Type list (default).

3. Click the **Find existing WSDLs** icon, which is located to the right of the WSDL URL field.

   The WSDL Chooser dialog is displayed, as shown in Figure 4–61.

Figure 4–61 WSDL Chooser Dialog



WSDL Chooser

Location: C:\12c_SOA\soa\soa\thirdparty\ApplicationAdapters\wsdls

J2CA_Inbound_receive.wsdl

File Name: J2CA_Inbound_receive.wsdl

File Type: Web Service Definition Files (*.wsdl)

Selection: file:/C:/12c_SOA/soa/soa/thirdparty/ApplicationAdapters/wsdls/J2CA_Inbound_receive.wsdl

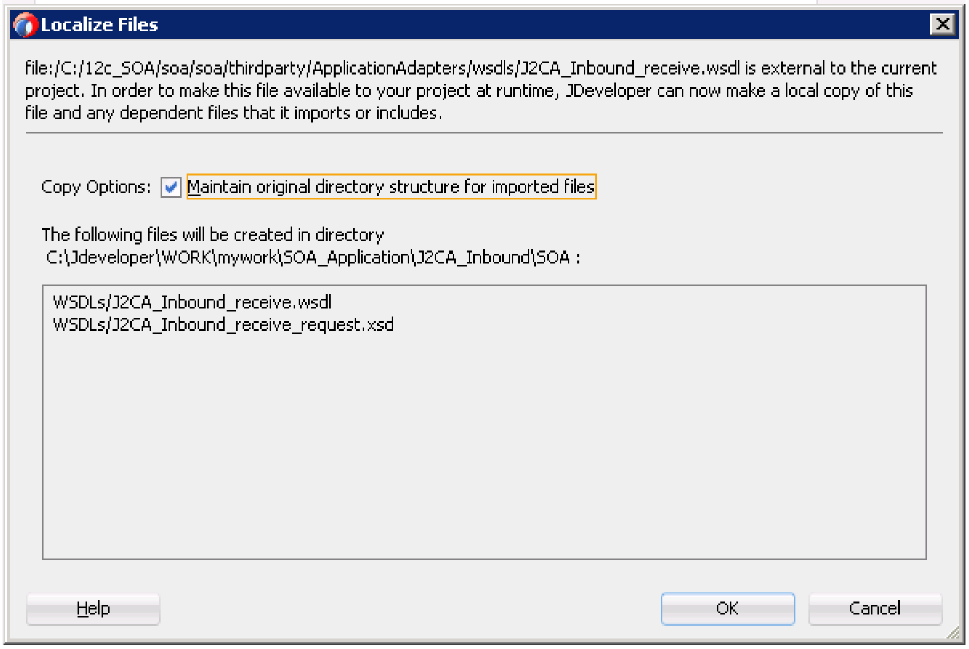
4. Browse and select an inbound WSDL file from the following directory:

```
<ADAPTER_HOME>\wsdls
```

5. Click **OK**.

   The Localize Files dialog is displayed, as shown in Figure 4–62.

Figure 4–62 Localize Files Dialog



file:/C:/12c_SOA/soa/soa/thirdparty/ApplicationAdapters/wsdls/J2CA_Inbound_receive.wsdl is external to the current project. In order to make this file available to your project at runtime, JDeveloper can now make a local copy of this file and any dependent files that it imports or includes.

Copy Options: ☑ Maintain original directory structure for imported files

The following files will be created in directory
C:\Jdeveloper\WORK\mywork\SOA_Application\J2CA_Inbound\SOA :

WSDLs/J2CA_Inbound_receive.wsdl
WSDLs/J2CA_Inbound_receive_request.xsd

6. Click **OK**.

   The inbound WSDL file and associated receive/request XML schema file (.xsd) are imported to the project folder that has been created.

   You are returned to the Create Third Party Adapter Service dialog.

7. Click the **Find JCA file** icon, which is located to the right of the JCA File field.

   The Transformation Chooser dialog is displayed.

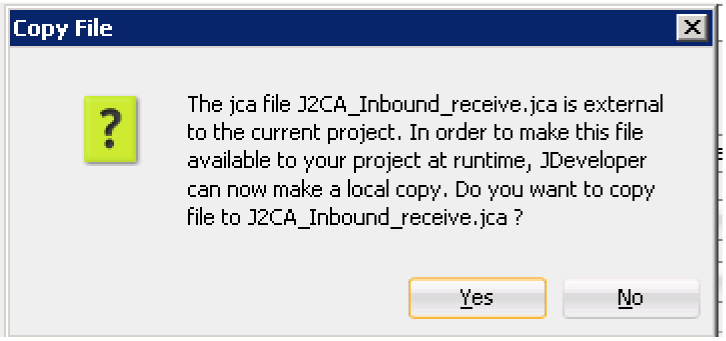
8. Browse and select the JCA properties file from the following directory:

```
<ADAPTER_HOME>\wsdls
```

9. Click **OK**.

   A Copy File message is displayed, as shown in Figure 4–63.

Figure 4–63 Copy File Confirmation Message



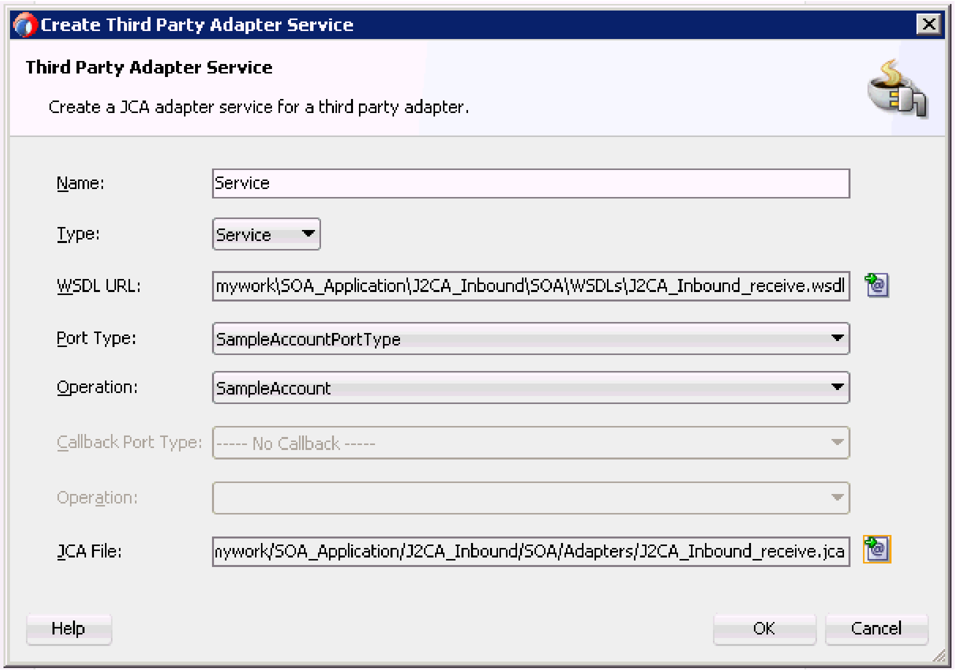
The jca file J2CA_Inbound_receive.jca is external to the current project. In order to make this file available to your project at runtime, JDeveloper can now make a local copy. Do you want to copy file to J2CA_Inbound_receive.jca ?

10. Click **Yes**.

    A copy of the JCA properties file is made in the project folder.

    You are returned to the Create Third Party Adapter Service dialog, as shown in Figure 4–64.

Figure 4–64 Create Third Party Adapter Service Dialog



Third Party Adapter Service

Create a JCA adapter service for a third party adapter.

Name: Service

Type: Service

WSDL URL: mywork\SOA_Application\J2CA_Inbound\SOA\WSDLs\J2CA_Inbound_receive.wsdl

Port Type: SampleAccountPortType

Operation: SampleAccount

Callback Port Type: ----- No Callback -----

Operation:

JCA File: nywork/SOA_Application/J2CA_Inbound/SOA/Adapters/J2CA_Inbound_receive.jca

11. Click **OK**.

    The third party adapter service component is created and displayed in the Exposed Services pane.

    You are now ready to configure an inbound BPEL process component.

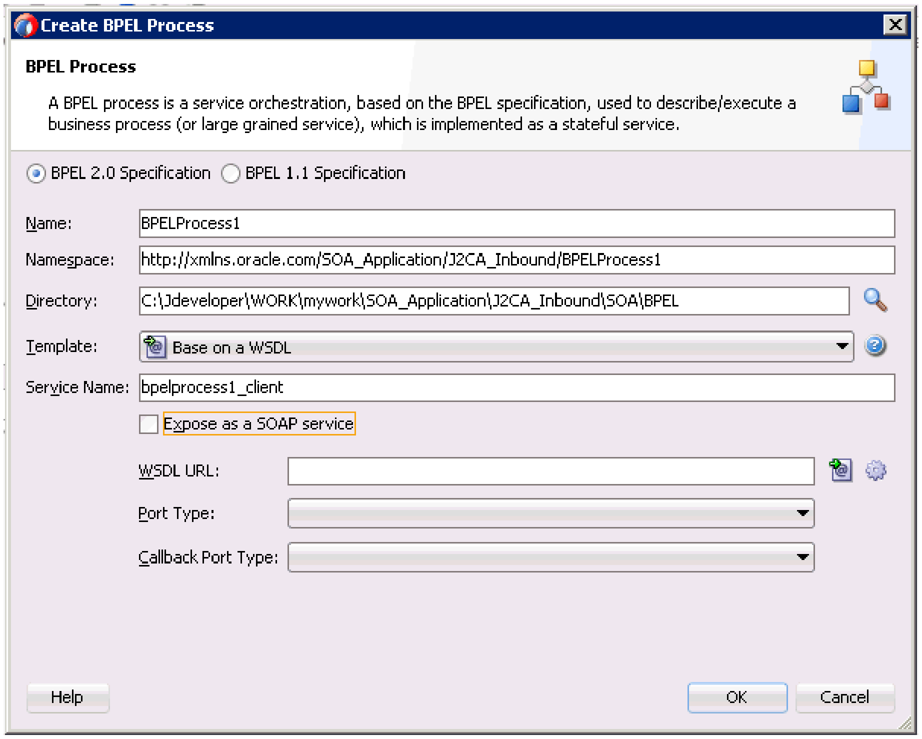
4.5.3.2 Creating an Inbound BPEL Process Component

Perform the following steps to create an inbound BPEL process component:

1. Drag and drop the **BPEL Process** component from the Service Components pane to the Components pane.

   The Create BPEL Process dialog is displayed, as shown in Figure 4–65.

Figure 4–65 Create BPEL Process Dialog



Create BPEL Process

BPEL Process

A BPEL process is a service orchestration, based on the BPEL specification, used to describe/execute a business process (or large grained service), which is implemented as a stateful service.

● BPEL 2.0 Specification ○ BPEL 1.1 Specification

Name: BPELProcess1

Namespace: http://xmlns.oracle.com/SOA_Application/J2CA_Inbound/BPELProcess1

Directory: C:\Jdeveloper\WORK\mywork\SOA_Application\J2CA_Inbound\SOA\BPEL

Template: Base on a WSDL

Service Name: bpelprocess1_client

□ Expose as a SOAP service

WSDL URL:

Port Type:

Callback Port Type:

2. In the Name field, enter a name to identify the new inbound BPEL process component or leave to default.

   By default, the BPEL 2.0 Specification option is selected.

3. From the Template list, select **Base on a WSDL**.
4. Uncheck the **Expose as SOAP service** check box.
5. Click the **Find existing WSDLs** icon, which is located to the right of the WSDL URL field.

   The WSDL Chooser dialog is displayed.

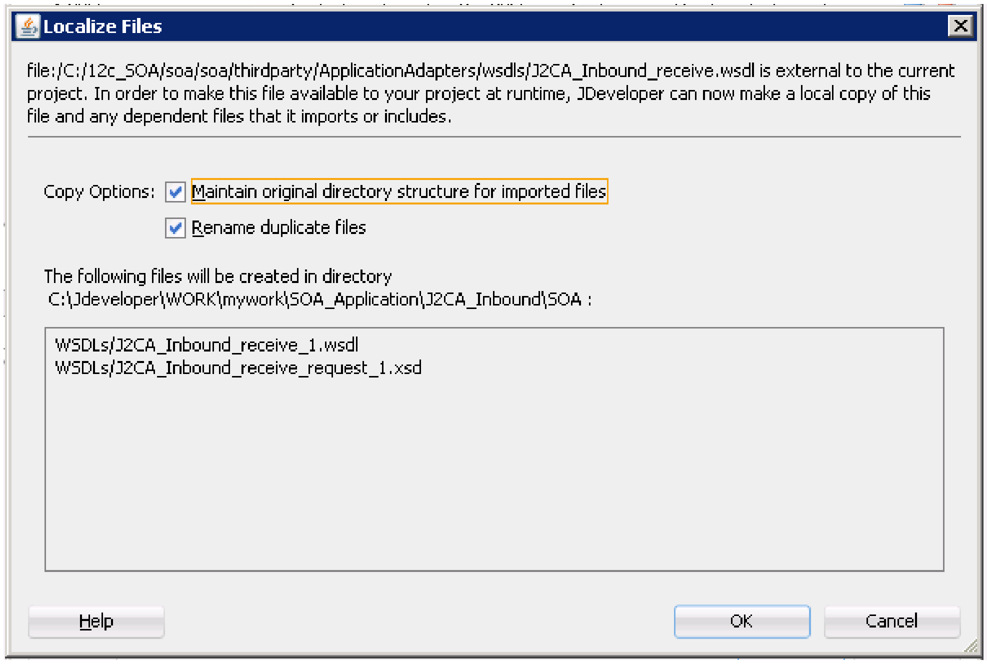
6. Select an inbound WSDL file from the following directory:

   ```
   <ADAPTER_HOME>\wsdls
   ```

7. Click **OK**.

   The Localize Files dialog is displayed, as shown in Figure 4–66.

Figure 4–66 Localize Files Dialog



Localize Files

file:/C:/12c_SOA/soa/soa/thirdparty/ApplicationAdapters/wsdls/J2CA_Inbound_receive.wsdl is external to the current project. In order to make this file available to your project at runtime, JDeveloper can now make a local copy of this file and any dependent files that it imports or includes.

Copy Options: ☑ Maintain original directory structure for imported files

☑ Rename duplicate files

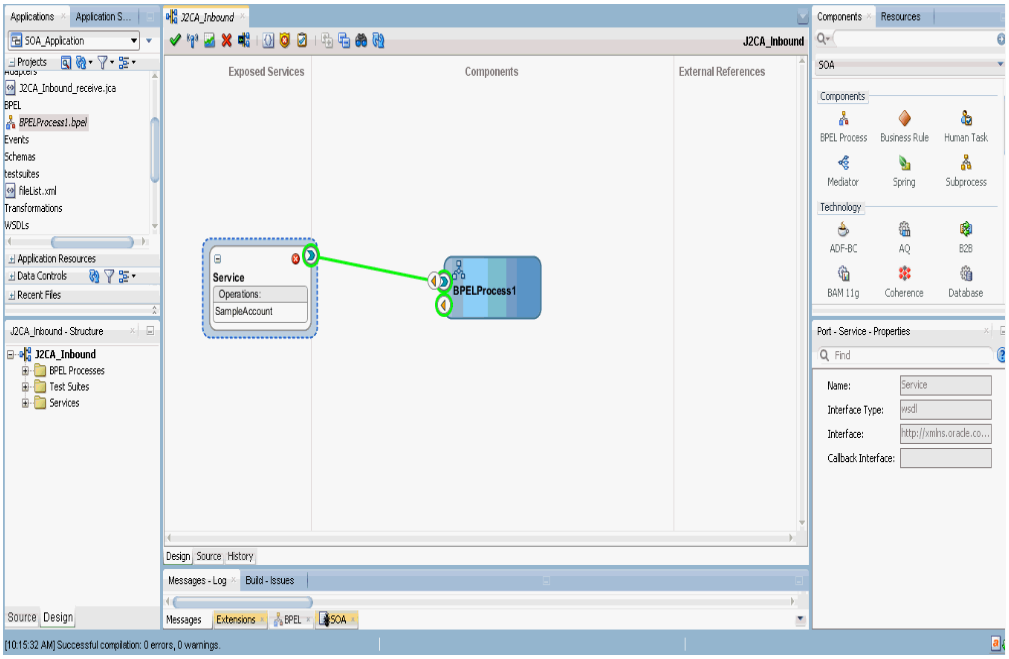
The following files will be created in directory
C:\Jdeveloper\WORK\mywork\SOA_Application\J2CA_Inbound\SOA :

WSDLs/J2CA_Inbound_receive_1.wsdl
WSDLs/J2CA_Inbound_receive_request_1.xsd

8. Uncheck the **Rename duplicate files** option.
9. Click **OK**.

   You are returned to the Create BPEL Process dialog.
10. Click **OK**.

Figure 4–67 Created Connection



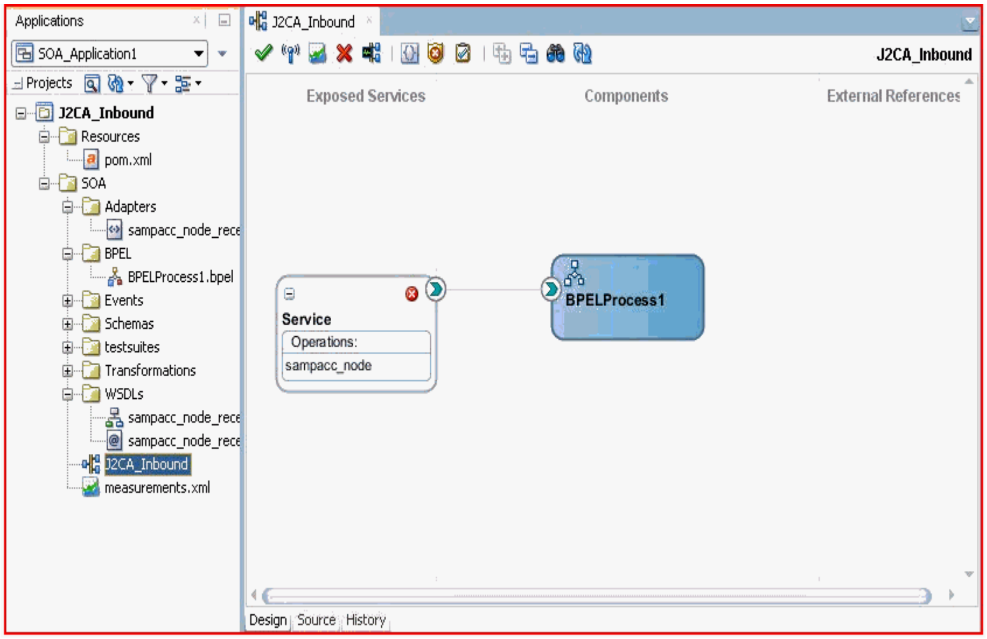
Name: Service
Interface Type: wsdl
Interface: http://xmlns.oracle.co...
Callback Interface:

[10:15:32 AM] Successful compilation: 0 errors, 0 warnings.

11. Create a connection between the third party adapter service component and the inbound BPEL process component, as shown in Figure 4–67.

12. Double-click **J2CA_Outbound** in the left pane.

Figure 4–68 Save All Icon



13. Click the **Save All** icon in the menu bar to save the new inbound BPEL process component that was configured, as shown in Figure 4–68.

    You are now ready to deploy the BPEL inbound process.

4.5.3.3 Adjusting for Known Deployment Issues With 12c

For more information on how to adjust for known deployment issues with 12c, see Section 4.4.3.3, "Adjusting for Known Deployment Issues With 12c" on page 4-26.

4.5.4 Deploying the BPEL Inbound Process

Perform the following steps to deploy the BPEL inbound process.

1. Right-click the project name in the left pane, select **Deploy**, and click **J2CA_ Inbound**.

   The Deployment Action page is displayed.

2. Ensure that **Deploy to Application Server** is selected.

3. Click **Next**.

   The Deploy Configuration page is displayed.

4. Leave the default values selected and click **Next**.

   The Select Server page is displayed.

5. Select an available application server that was configured and click **Next**.

   The SOA Servers page is displayed.

6. Select a target SOA server and click **Next**.

   The Summary page is displayed.

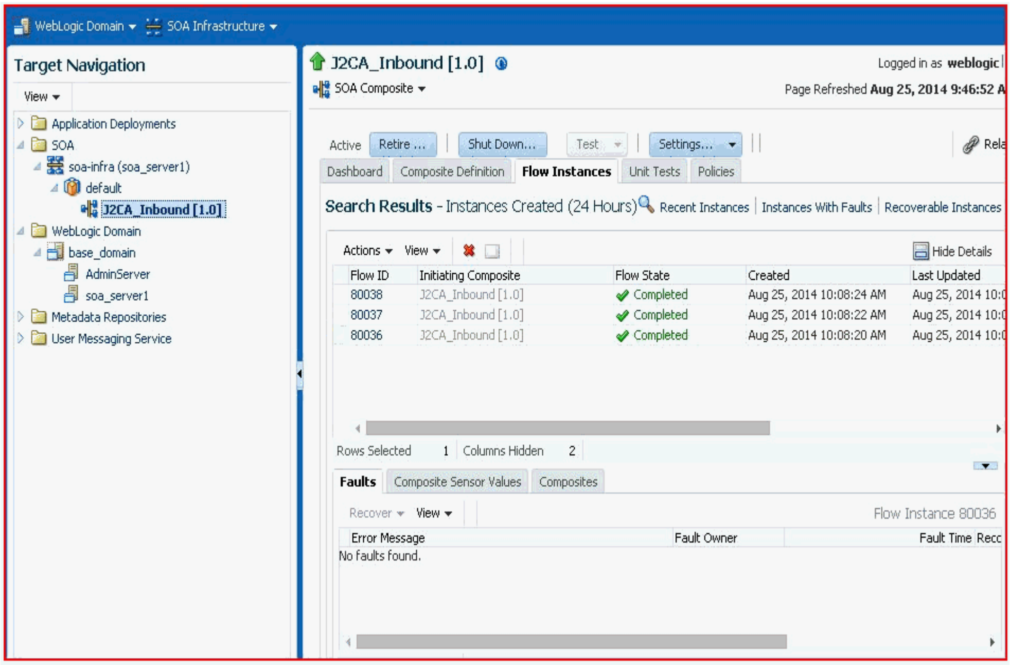
7. Review and verify all the available deployment information for your project and click **Finish**.

   The process is deployed successfully.

   For more information, see Section 4.4.4, "Deploying the BPEL Outbound Process" on page 4-28.

   Once event messages are triggered through Siebel, successful instances are received in the Oracle Enterprise Manager console, as shown in Figure 4–69.

Figure 4–69 Received Instances



Search Results - Instances Created (24 Hours)

Flow ID	Initiating Composite	Flow State	Created	Last Updated
80038	J2CA_Inbound [1.0]	✓ Completed	Aug 25, 2014 10:08:24 AM	Aug 25, 2014 10:(
80037	J2CA_Inbound [1.0]	✓ Completed	Aug 25, 2014 10:08:22 AM	Aug 25, 2014 10:(
80036	J2CA_Inbound [1.0]	✓ Completed	Aug 25, 2014 10:08:20 AM	Aug 25, 2014 10:(

Rows Selected 1 Columns Hidden 2

Flow Instance 80036

Error Message	Fault Owner	Fault Time	Recc
No faults found.			

4.5.5 Triggering an Event in Siebel

This section describes how to trigger an event in Siebel and verify event integration using Oracle Application Adapter for Siebel.

This section contains the following topics:

- Section 4.5.5.1, "Triggering a Siebel Event to Test Event Runtime Integration"
- Section 4.5.5.2, "Triggering an Event in Siebel 7.8 to Test Event Runtime Integration"
- Section 4.5.5.3, "Triggering an Event in Siebel 8.0 to Test Event Runtime Integration"
- Section 4.5.5.4, "Verifying the Results"

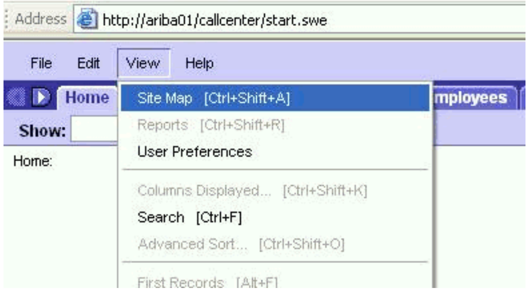
4.5.5.1 Triggering a Siebel Event to Test Event Runtime Integration

To trigger an event in Siebel:

1. As shown in Figure 4–70, start the Siebel Call Center by entering the following URL in a browser:

```
http://host name/callcenter/start.swe
```

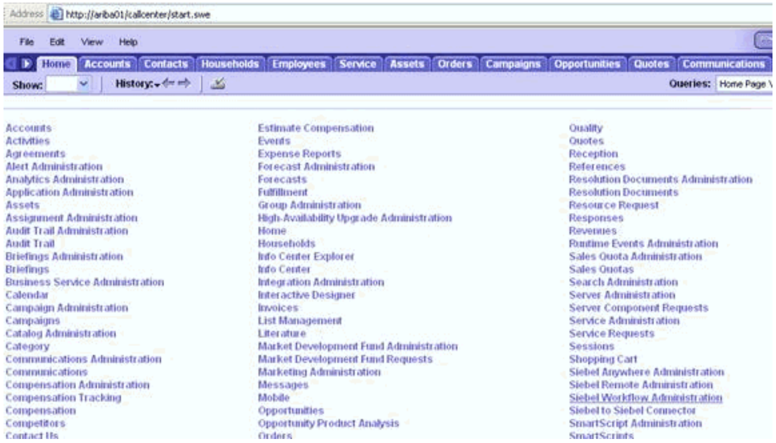
Figure 4–70 Site Map Option Selected Under the View Menu in the Siebel Call Center



2. Click **View** and select **Site Map** from the list.

   The Site Map view is displayed, as shown in Figure 4–71.

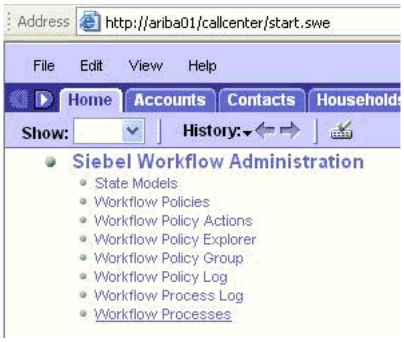
Figure 4–71 Site Map View



3. Click **Siebel Workflow Administration**.

   The Siebel Workflow Administration page is displayed, as shown in Example 4–72.

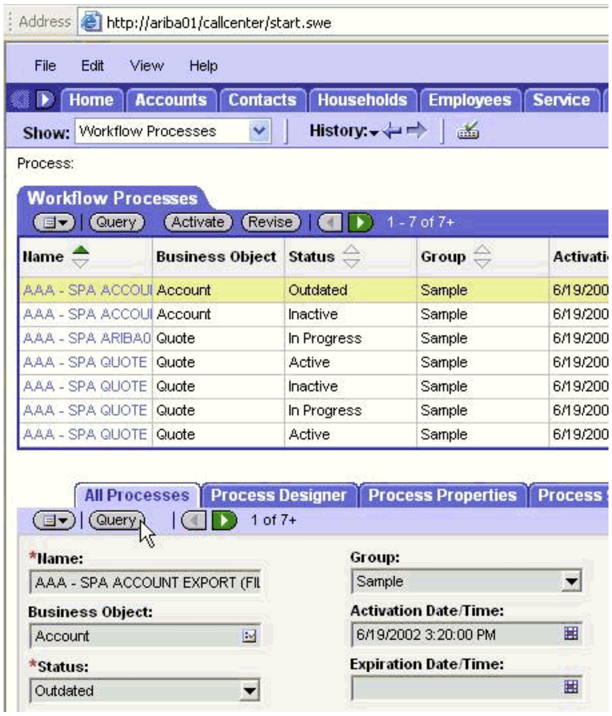
Figure 4–72 Siebel Workflow Administration Page



4. Click **Workflow Processes**.

   The Workflow Processes page is displayed, as shown in Example 4–73.

Figure 4–73 Workflow Processes Page


Address http://ariba01/callcenter/start.swe
File Edit View Help
Home Accounts Contacts Households Employees Service
Show: Workflow Processes
History:
Process:
Workflow Processes
Query Activate Revise 1 - 7 of 7+
All Processes Process Designer Process Properties Process
Query 1 of 7+

Name	Business Object	Status	Group	Activati
AAA - SPA ACCOU	Account	Outdated	Sample	6/19/200
AAA - SPA ACCOU	Account	Inactive	Sample	6/19/200
AAA - SPA ARIBA0	Quote	In Progress	Sample	6/19/200
AAA - SPA QUOTE	Quote	Active	Sample	6/19/200
AAA - SPA QUOTE	Quote	Inactive	Sample	6/19/200
AAA - SPA QUOTE	Quote	In Progress	Sample	6/19/200
AAA - SPA QUOTE	Quote	Active	Sample	6/19/200

*Name: AAA - SPA ACCOUNT EXPORT (FIL

Group: Sample

Business Object: Account

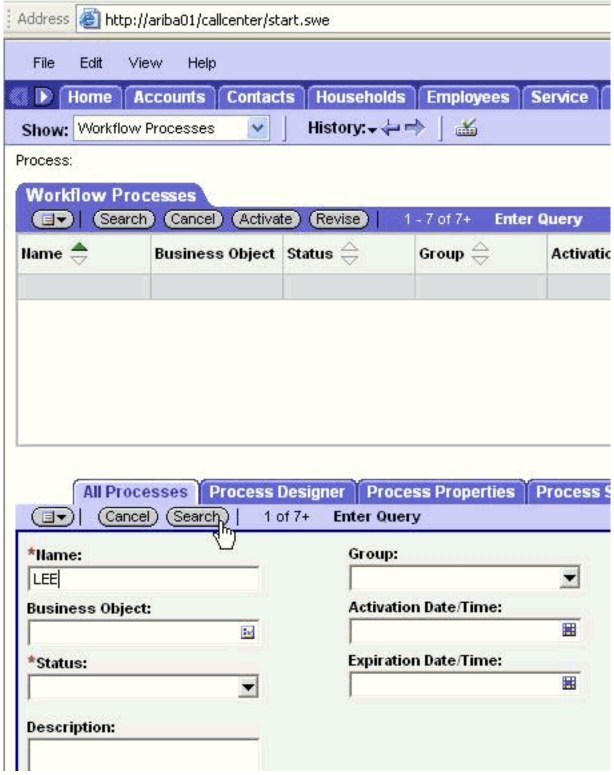
Activation Date/Time: 6/19/2002 3:20:00 PM

*Status: Outdated

Expiration Date/Time:

5. Click **Query** to search for the Workflow needed to trigger a Siebel event, as shown in Figure 4–74.

Figure 4–74 Search Button in Workflow Processes Page


Address http://ariba01/callcenter/start.swe
File Edit View Help
Home Accounts Contacts Households Employees Service
Show: Workflow Processes
History:
Process:
Workflow Processes
Search Cancel Activate Revise 1 - 7 of 7+ Enter Query
Name Business Object Status Group Activatic
All Processes Process Designer Process Properties Process S
Cancel Search 1 of 7+ Enter Query

*Name: LEE

Group:

Business Object:

Activation Date/Time:

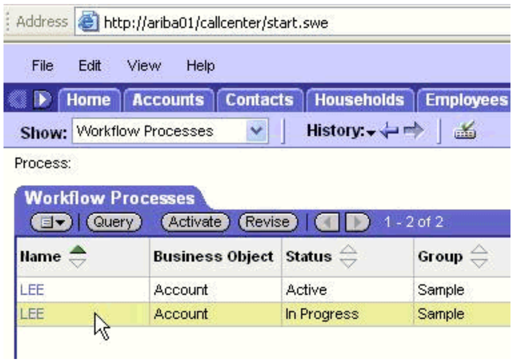
*Status:

Expiration Date/Time:

Description:

6. As shown in Figure 4–75, enter a Siebel workflow name and click **Search**.

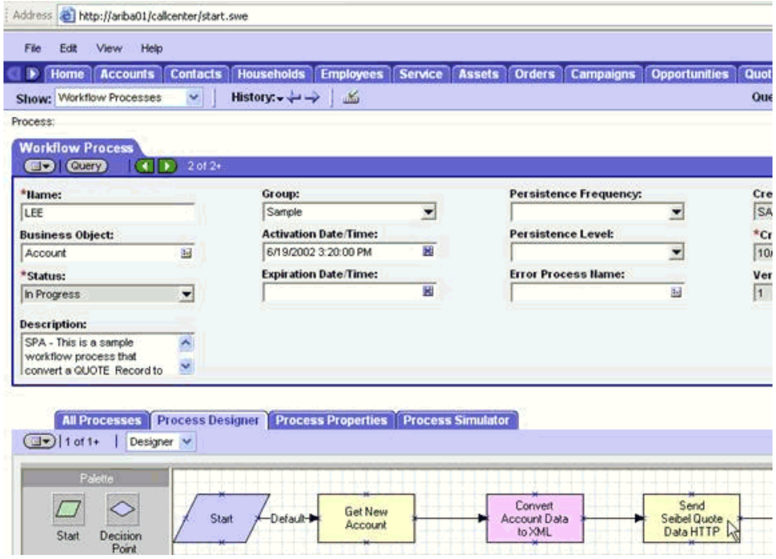
Figure 4–75 Workflow Processes Page



Name	Business Object	Status	Group
LEE	Account	Active	Sample
LEE	Account	In Progress	Sample

7. As shown in Figure 4–76, select the workflow.

Figure 4–76 Process Designer Tab


Address http://ariba01/callcenter/start.swe
File Edit View Help
Home Accounts Contacts Households Employees Service Assets Orders Campaigns Opportunities Quot
Show: Workflow Processes
History:
Que
Process:
Workflow Process
Query
2 of 2+
All Processes Process Designer Process Properties Process Simulator
1 of 1+
Designer
Palette
Start
Decision Point
Start
Default
Get New Account
Convert Account Data to XML
Send Seibel Quote Data HTTP

*Name: LEE

Business Object: Account

*Status: In Progress

Description: SPA - This is a sample workflow process that convert a QUOTE Record to

Group: Sample

Activation Date/Time: 6/19/2002 3:20:00 PM

Expiration Date/Time:

Persistence Frequency:

Persistence Level:

Error Process Name:

Cre SA

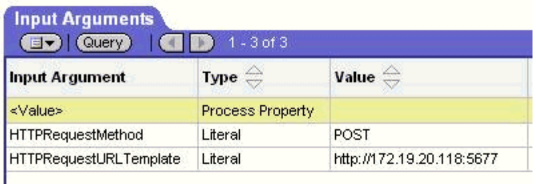
*Cr 10/

Ver 1

8. Click the **Process Designer** tab and double-click the **Send Siebel Quote Data HTTP** workflow element.

   The Input Arguments tab is displayed, as shown in Figure 4–77.

Figure 4–77 Input Arguments Tab

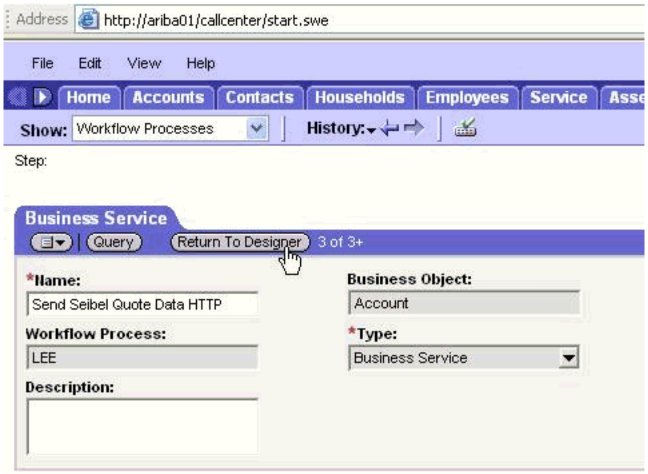


Input Argument	Type	Value
<Value>	Process Property	
HTTPRequestMethod	Literal	POST
HTTPRequestURLTemplate	Literal	http://172.19.20.118:5677

9. Enter the IP address and port for the HTTPRequestURLTemplate input argument.

10. Click **Return To Designer**, as shown in Figure 4–78.

Figure 4–78 Return To Designer Button in Business Service Tab



Address: http://ariba01/callcenter/start.swe

Show: Workflow Processes

*Name: Send Seibel Quote Data HTTP

Business Object: Account

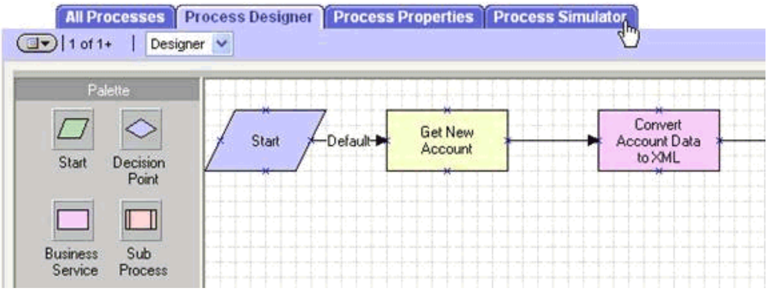
Workflow Process: LEE

*Type: Business Service

Description:

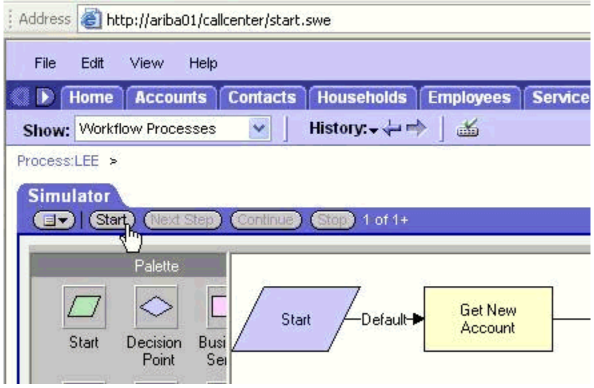
11. Click the Process Simulator tab, as shown in Figure 4–79.

Figure 4–79 Process Simulator Tab



The Simulator tab is displayed, as shown in Figure 4–80.

Figure 4–80 Simulator Tab



Address: http://ariba01/callcenter/start.swe

Show: Workflow Processes

12. Click **Start** then **Continue** to complete the Siebel event triggering process.

4.5.5.2 Triggering an Event in Siebel 7.8 to Test Event Runtime Integration

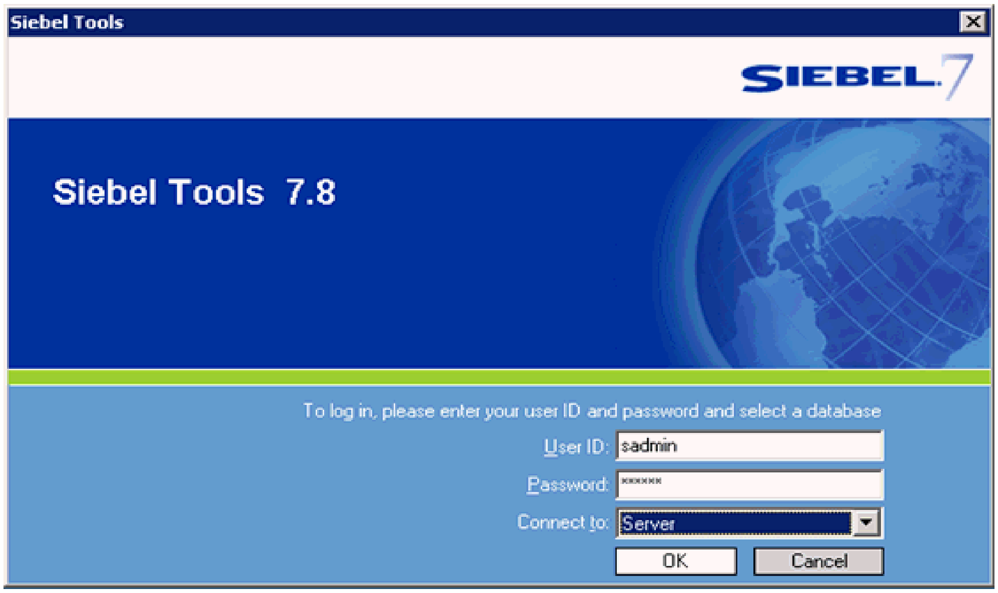
To trigger an event in Siebel 7.8:

1. Log in to Siebel Tools 7.8 by using the following parameters:

```
Username = sadmin

Password = sadmin
```

2. Choose **Server** from the Connect to list and click **OK** as shown in Figure 4–81.

Figure 4–81 Siebel Tools 7.8 Log-in Pane



To log in, please enter your user ID and password and select a database

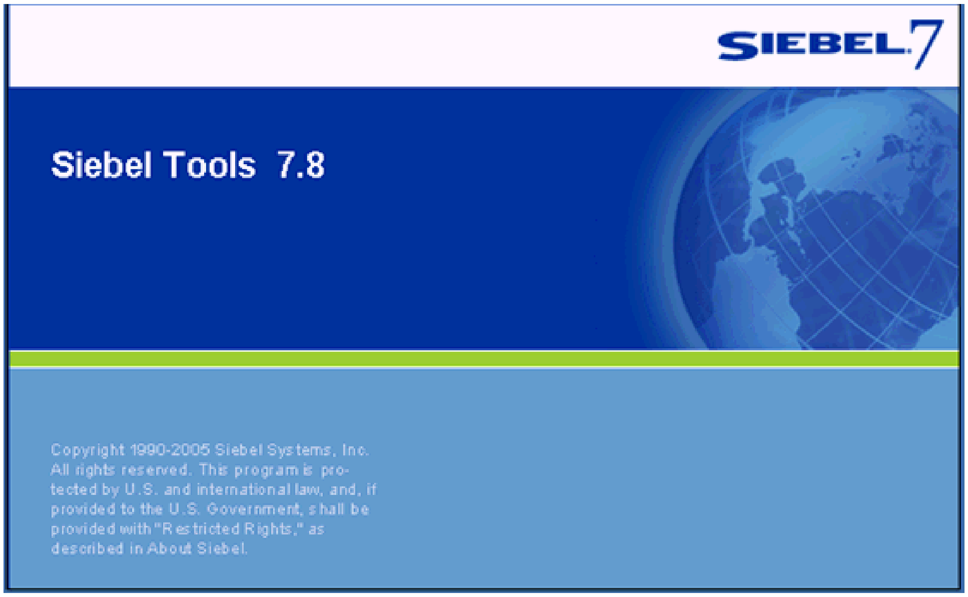
User ID: sadmin

Password: ******

Connect to: Server

You are logged-in to Siebel Tools 7.8, as shown in Figure 4–82.

Figure 4–82 Siebel Tools 7.8 Startup Pane

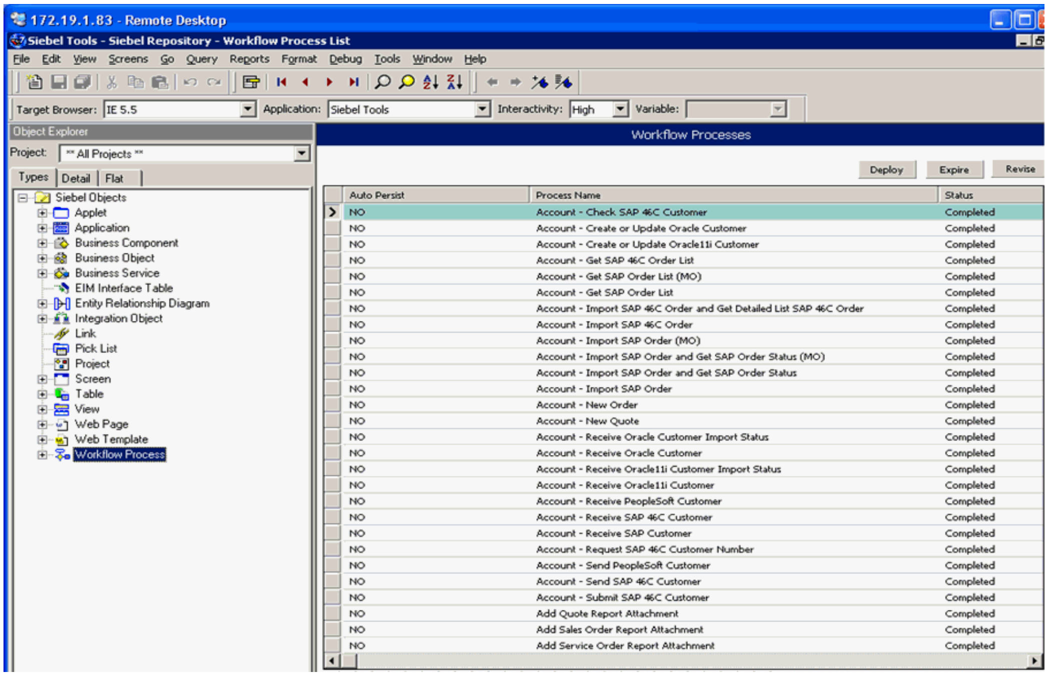


Copyright 1990-2005 Siebel Systems, Inc. All rights reserved. This program is protected by U.S. and international law, and, if provided to the U.S. Government, shall be provided with "Restricted Rights," as described in About Siebel.

3. On the left pane, click on **Workflow Process**.

   The Workflow Processes pane is displayed, as shown in Figure 4–83.

Figure 4–83 Workflow Processes Pane



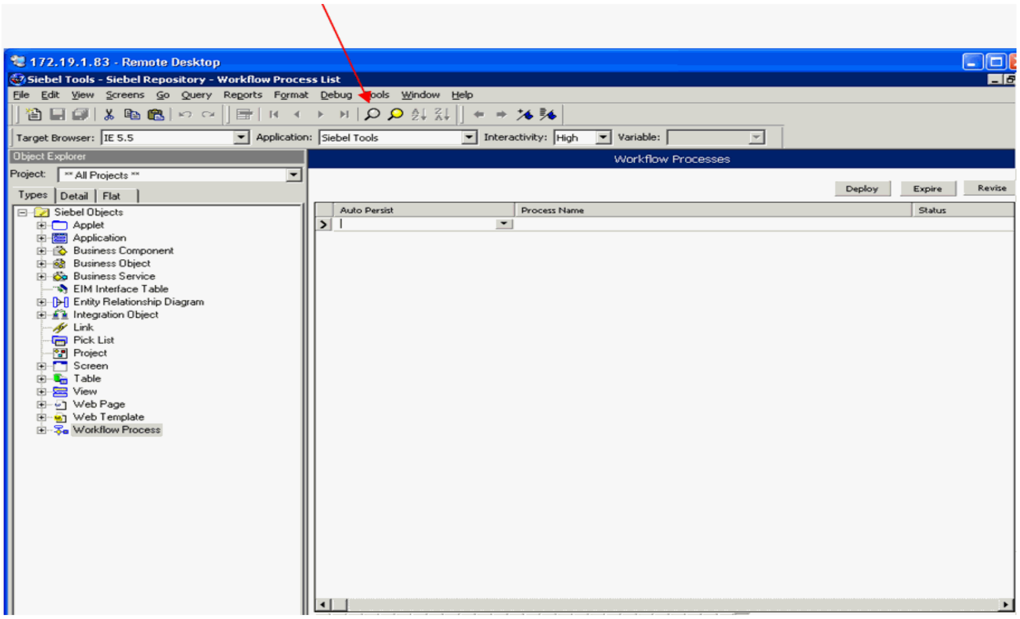
Target Browser: IE 5.5 Application: Siebel Tools Interactivity: High Variable:

Project: ** All Projects **

Auto Persist	Process Name	Status
NO	Account - Check SAP 46C Customer	Completed
NO	Account - Create or Update Oracle Customer	Completed
NO	Account - Create or Update Oracle11i Customer	Completed
NO	Account - Get SAP 46C Order List	Completed
NO	Account - Get SAP Order List (MO)	Completed
NO	Account - Get SAP Order List	Completed
NO	Account - Import SAP 46C Order and Get Detailed List SAP 46C Order	Completed
NO	Account - Import SAP 46C Order	Completed
NO	Account - Import SAP Order (MO)	Completed
NO	Account - Import SAP Order and Get SAP Order Status (MO)	Completed
NO	Account - Import SAP Order and Get SAP Order Status	Completed
NO	Account - Import SAP Order	Completed
NO	Account - New Order	Completed
NO	Account - New Quote	Completed
NO	Account - Receive Oracle Customer Import Status	Completed
NO	Account - Receive Oracle Customer	Completed
NO	Account - Receive Oracle11i Customer Import Status	Completed
NO	Account - Receive Oracle11i Customer	Completed
NO	Account - Receive PeopleSoft Customer	Completed
NO	Account - Receive SAP 46C Customer	Completed
NO	Account - Receive SAP Customer	Completed
NO	Account - Request SAP 46C Customer Number	Completed
NO	Account - Send PeopleSoft Customer	Completed
NO	Account - Send SAP 46C Customer	Completed
NO	Account - Submit SAP 46C Customer	Completed
NO	Add Quote Report Attachment	Completed
NO	Add Sales Order Report Attachment	Completed
NO	Add Service Order Report Attachment	Completed

4. Click on the **New Query** magnifying tool icon with the white glow, as shown in Figure 4–84.

Figure 4–84 New Query Magnifying Tool Icon

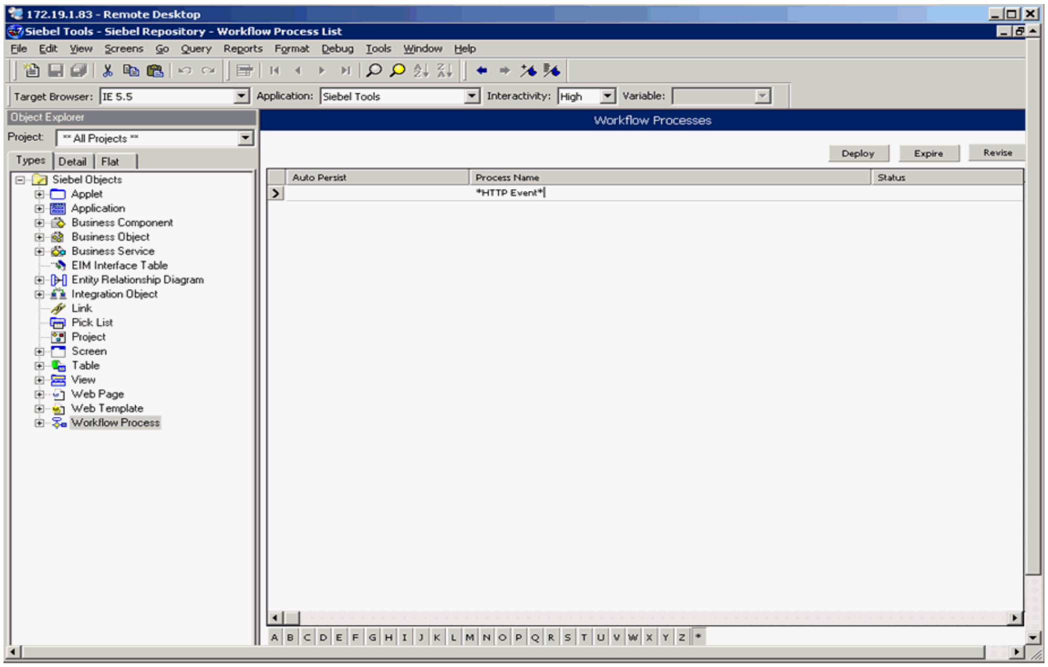


Target Browser: IE 5.5 Application: Siebel Tools Interactivity: High Variable:

Project: ** All Projects **

5. In the Process Name field, enter the name ***HTTP Event*** as shown in Figure 4–85.

Figure 4–85 Process Name Field



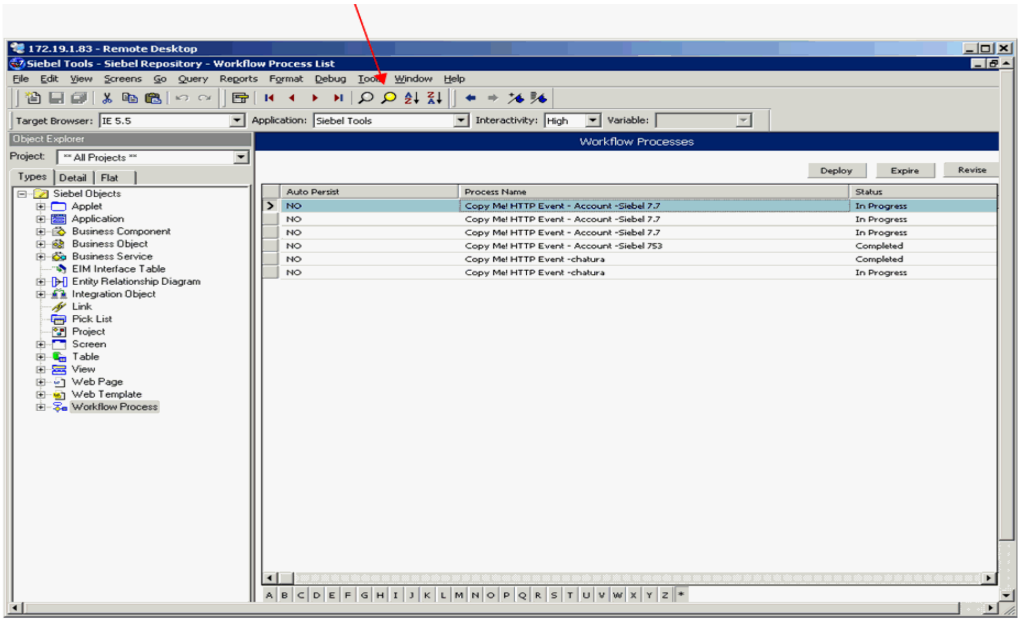
Target Browser: IE 5.5 Application: Siebel Tools Interactivity: High Variable:

Project: ** All Projects **

Auto Persist	Process Name	Status
	HTTP Event	

6. Click on the **Magnifying Tool** icon with the yellow glow, as shown in Figure 4–86.

Figure 4–86 Yellow Magnifying Tool Icon



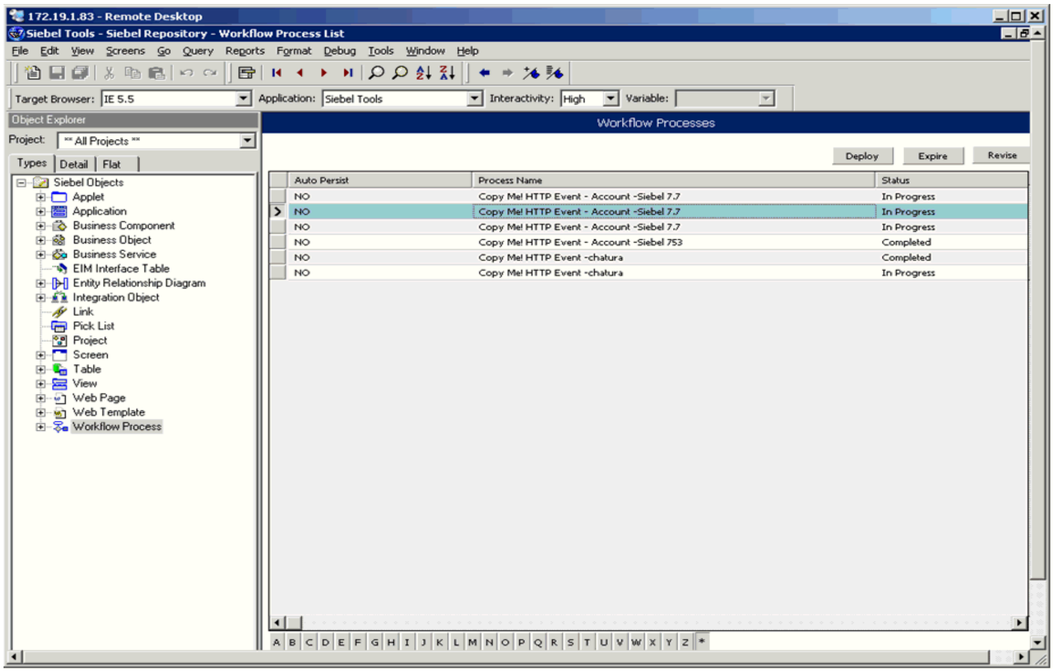
Target Browser: IE 5.5 Application: Siebel Tools Interactivity: High Variable:

Project: ** All Projects **

Auto Persist	Process Name	Status
NO	Copy Me! HTTP Event - Account -Siebel 7.7	In Progress
NO	Copy Me! HTTP Event - Account -Siebel 7.7	In Progress
NO	Copy Me! HTTP Event - Account -Siebel 7.7	In Progress
NO	Copy Me! HTTP Event - Account -Siebel 753	Completed
NO	Copy Me! HTTP Event -chatura	Completed
NO	Copy Me! HTTP Event -chatura	In Progress

7. Click on the line, **Copy Me! HTTP Event - Account -Siebel 7.7**, as shown in Figure 4–87.

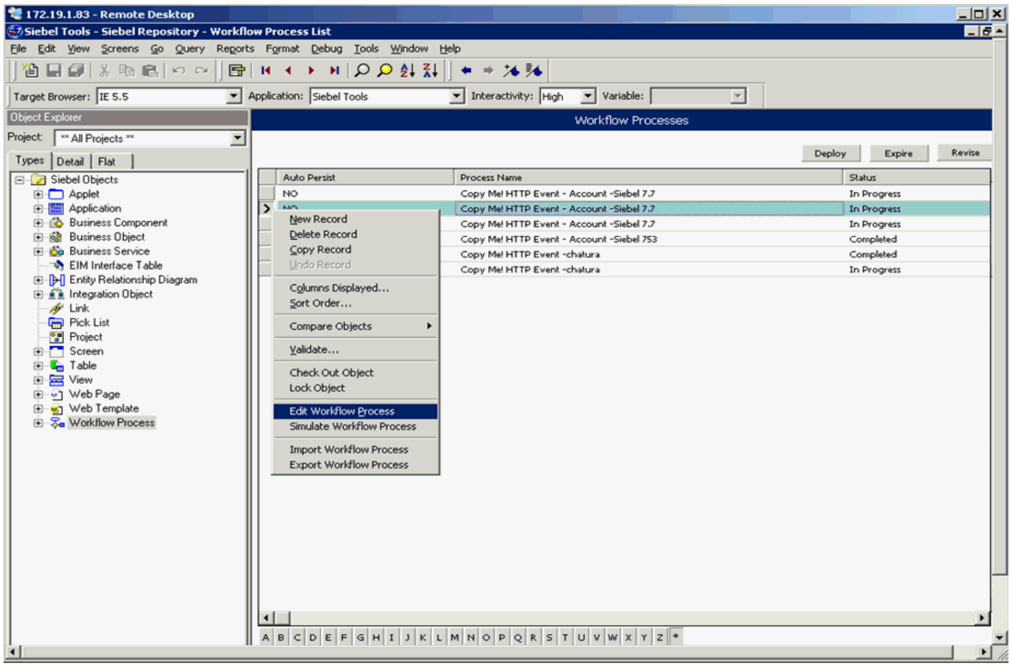
Figure 4–87 Copy Me! HTTP Event - Account -Siebel 7.7



Auto Persist	Process Name	Status
NO	Copy Me! HTTP Event - Account -Siebel 7.7	In Progress
NO	Copy Me! HTTP Event - Account -Siebel 7.7	In Progress
NO	Copy Me! HTTP Event - Account -Siebel 7.7	In Progress
NO	Copy Me! HTTP Event - Account -Siebel 753	Completed
NO	Copy Me! HTTP Event -chatura	Completed
NO	Copy Me! HTTP Event -chatura	In Progress

8. Right-click the arrow next to the selection and select **Edit Workflow Process**, as shown in Figure 4–88.

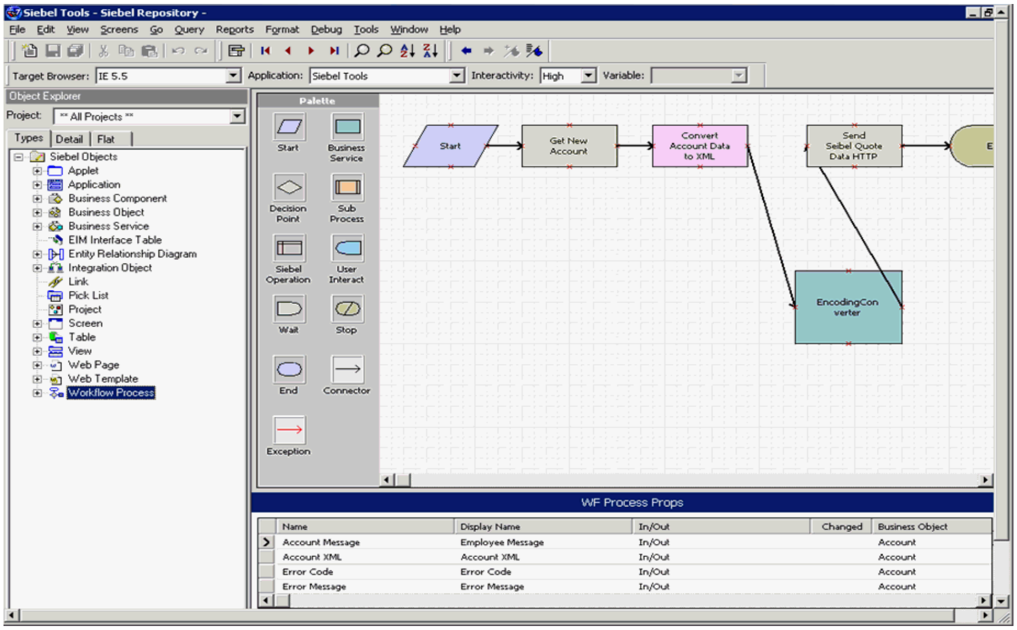
Figure 4–88 Edit Workflow Process Option



Auto Persist	Process Name	Status
NO	Copy Me! HTTP Event - Account -Siebel 7.7	In Progress
NO	Copy Me! HTTP Event - Account -Siebel 7.7	In Progress
	Copy Me! HTTP Event - Account -Siebel 7.7	In Progress
	Copy Me! HTTP Event - Account -Siebel 753	Completed
	Copy Me! HTTP Event -chatura	Completed
	Copy Me! HTTP Event -chatura	In Progress

A diagram is displayed on the right pane, as shown in Figure 4–89.

Figure 4–89 Workflow Process Diagram



Target Browser: IE 5.5 Application: Siebel Tools Interactivity: High Variable:

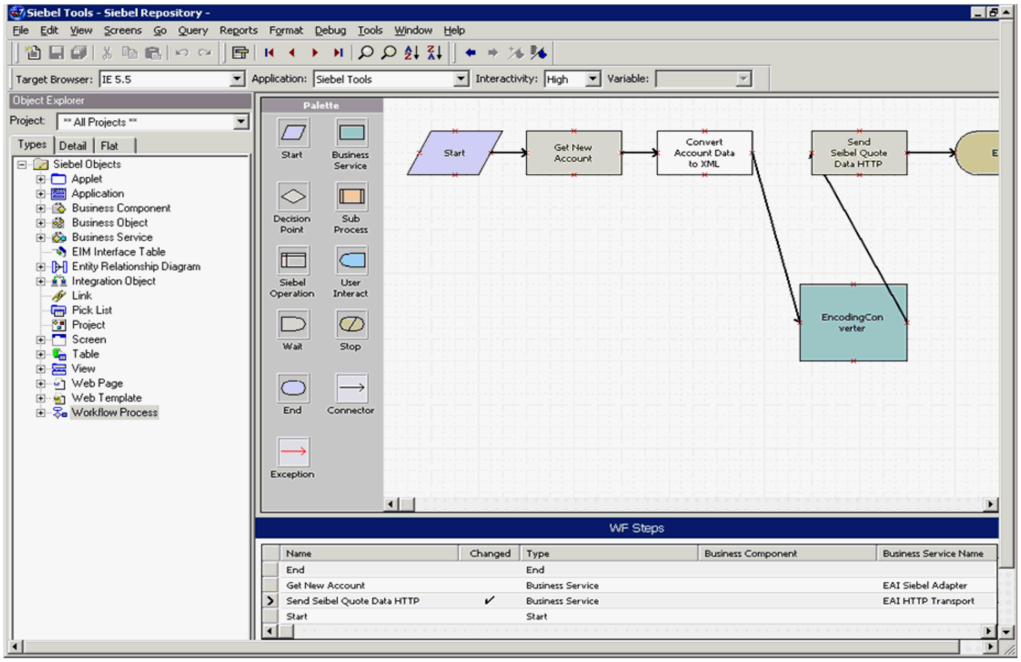
Project: ** All Projects **

WF Process Props

	Name	Display Name	In/Out	Changed	Business Object
>	Account Message	Employee Message	In/Out		Account
	Account XML	Account XML	In/Out		Account
	Error Code	Error Code	In/Out		Account
	Error Message	Error Message	In/Out		Account

9. Click the diagram box entitled, **Send Siebel Quote Data HTTP**, as shown in Figure 4–90.

Figure 4–90 Send Siebel Quote Data HTTP



Target Browser: IE 5.5 Application: Siebel Tools Interactivity: High Variable:

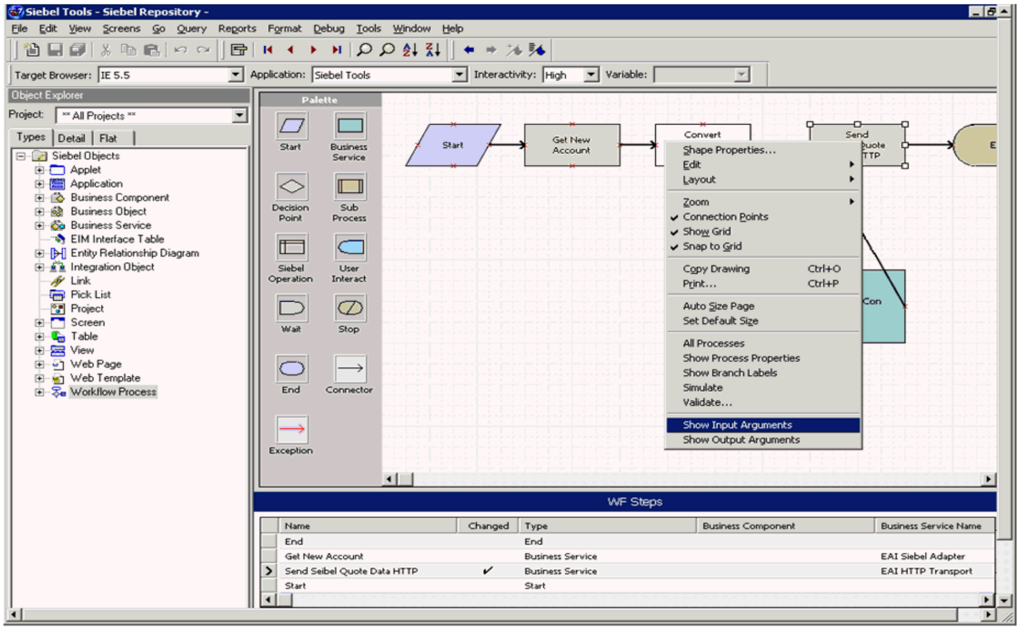
Project: ** All Projects **

WF Steps

	Name	Changed	Type	Business Component	Business Service Name
	End		End		
	Get New Account		Business Service		EAI Siebel Adapter
>	Send Seibel Quote Data HTTP	✓	Business Service		EAI HTTP Transport
	Start		Start		

10. Right-click **Send Siebel Quote Data HTTP** and select **Show Input Arguments**, as shown in Figure 4–91.

Figure 4–91 Show Input Arguments Option

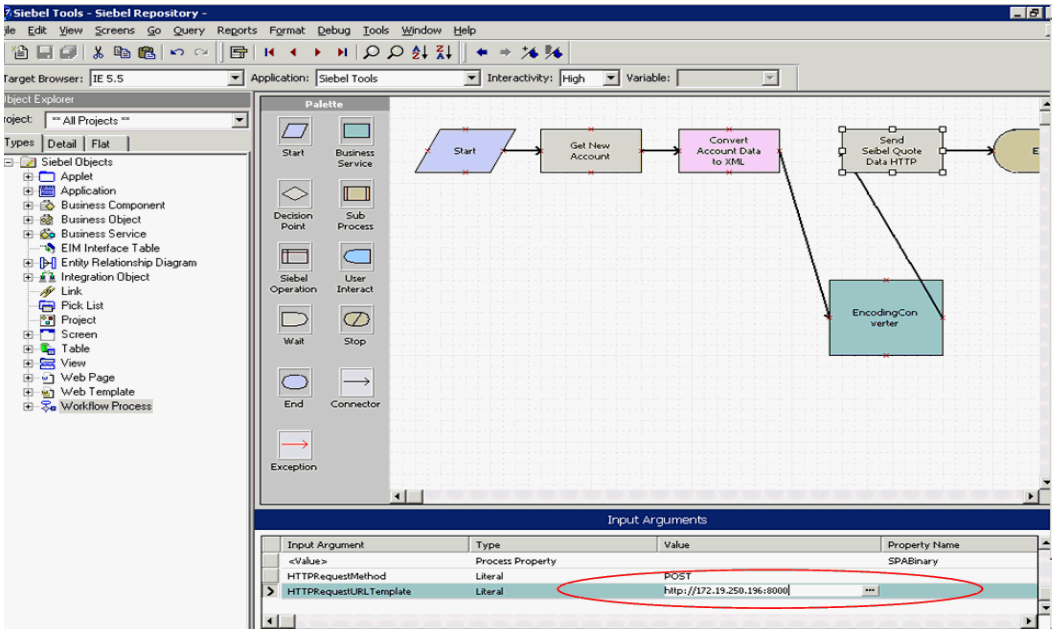


Name	Changed	Type	Business Component	Business Service Name
End		End		
Get New Account		Business Service		EAI Siebel Adapter
Send Seibel Quote Data HTTP	✓	Business Service		EAI HTTP Transport
Start		Start		

11. At the bottom pane, enter the value for **HTTPrequestURLTemplate**, as shown in Figure 4–92, by using the following URL:

```
http://machineIP: portno
```

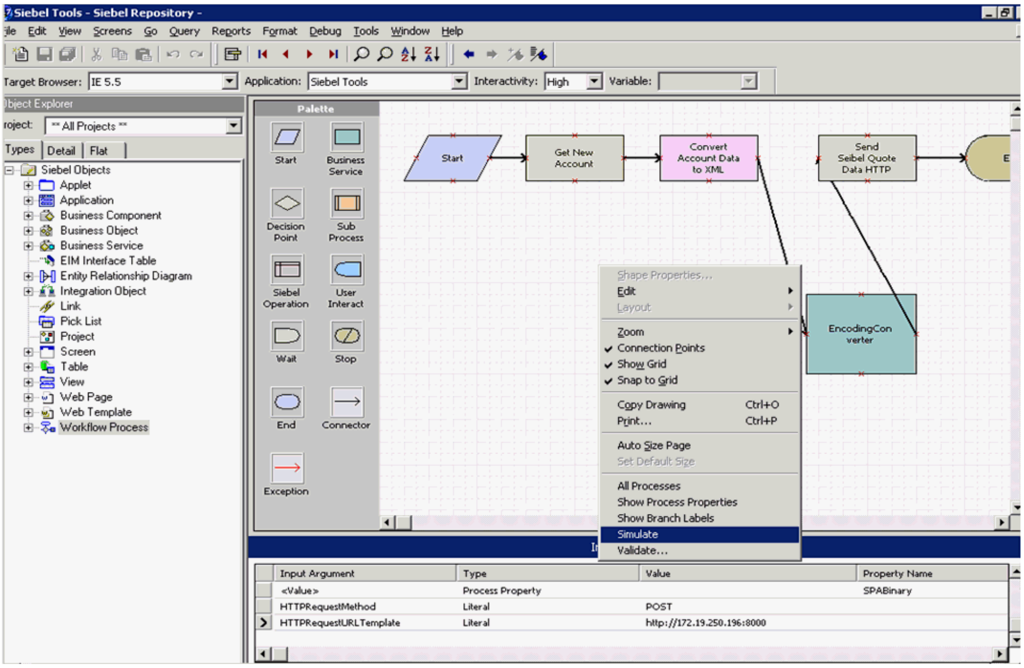
Figure 4–92 HTTPrequestURLTemplate Value



Input Argument	Type	Value	Property Name
<Value>	Process Property		SPABinary
HTTPRequestMethod	Literal	POST	
HTTPRequestURLTemplate	Literal	http://172.19.250.196:8000	

12. Right-click the diagram and select **Simulate**, as shown in Figure 4–93.

Figure 4–93 Simulate Option



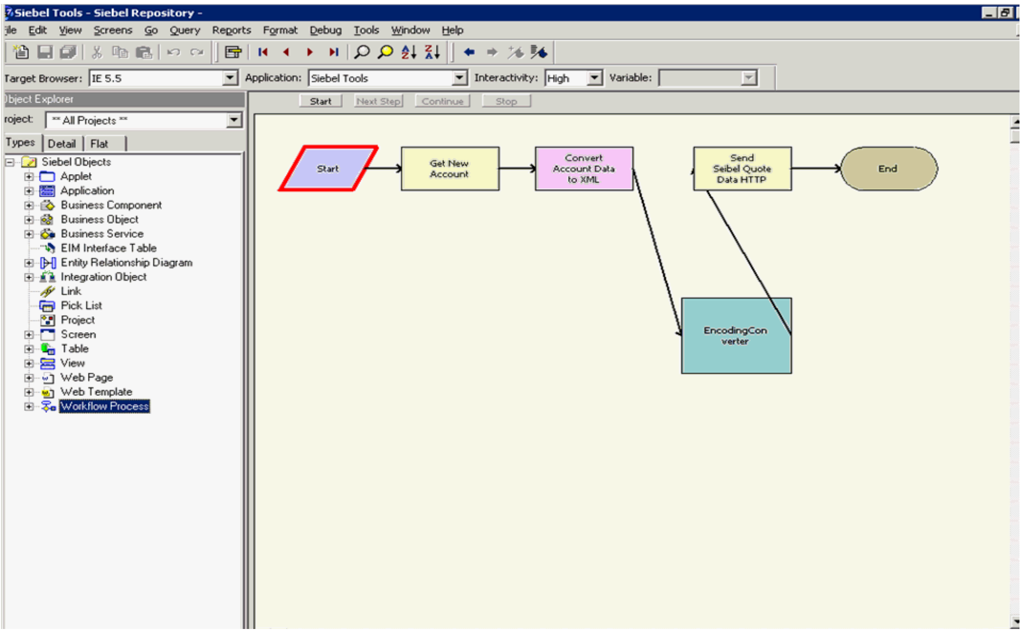
Target Browser: IE 5.5 Application: Siebel Tools Interactivity: High Variable:

roject: ** All Projects **

Input Argument	Type	Value	Property Name
<Value>	Process Property		SPABinary
HTTPRequestMethod	Literal	POST	
HTTPRequestURLTemplate	Literal	http://172.19.250.196:8000	

The Repository diagram is displayed, as shown in Figure 4–94.

Figure 4–94 Repository Diagram

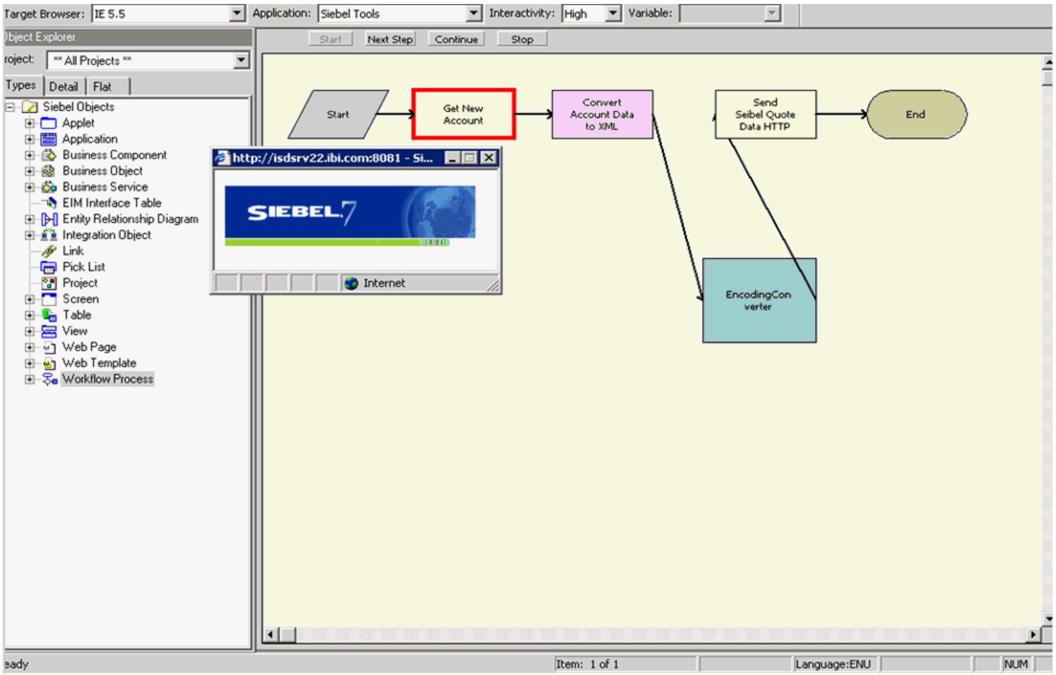


Target Browser: IE 5.5 Application: Siebel Tools Interactivity: High Variable:

roject: ** All Projects **

13. Click **Start** and then minimize the Siebel 7 window that is displayed, as shown in Figure 4–95.

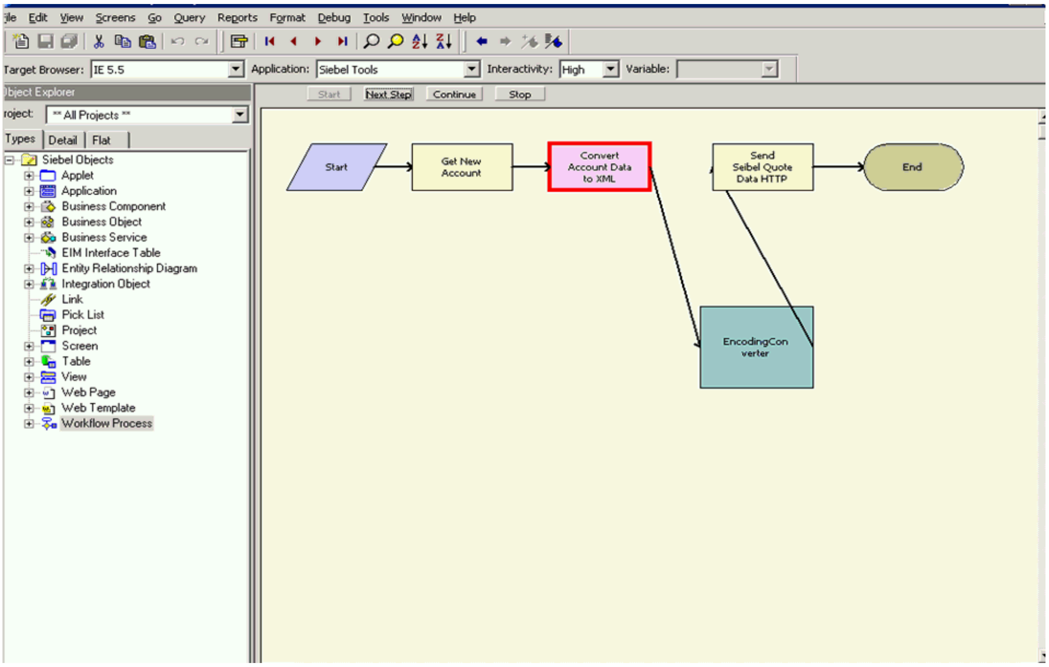
Figure 4–95 Siebel 7 Window



14. Click **Next Step**. The Convert Account Data to XML image is highlighted, as shown in Figure 4–96.

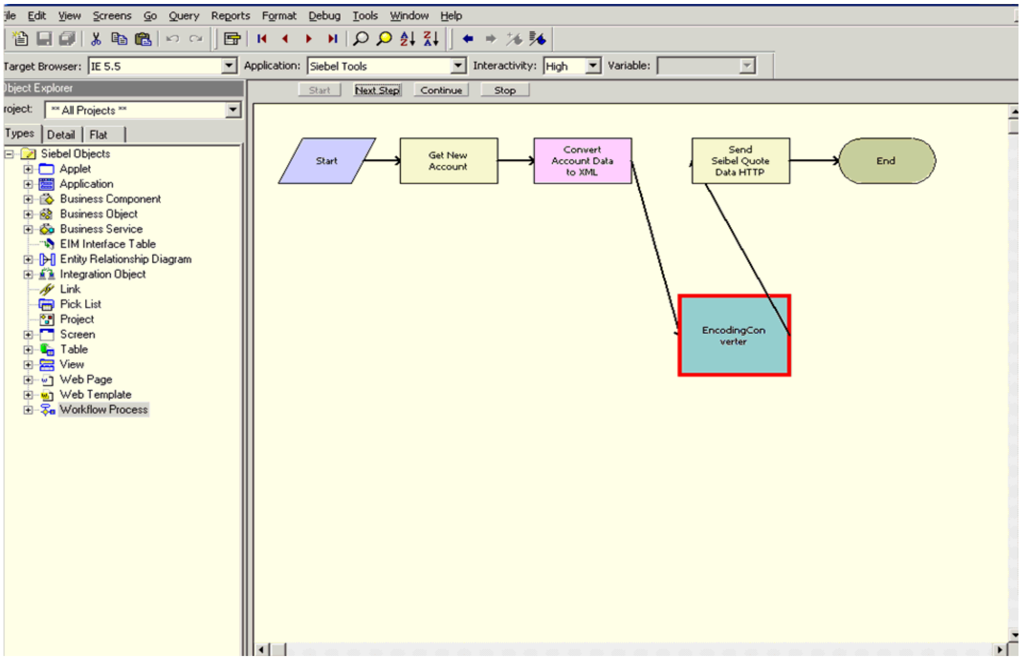
> **Note:** A red outline highlights each diagram image on each step.

Figure 4–96 Convert Data to XML



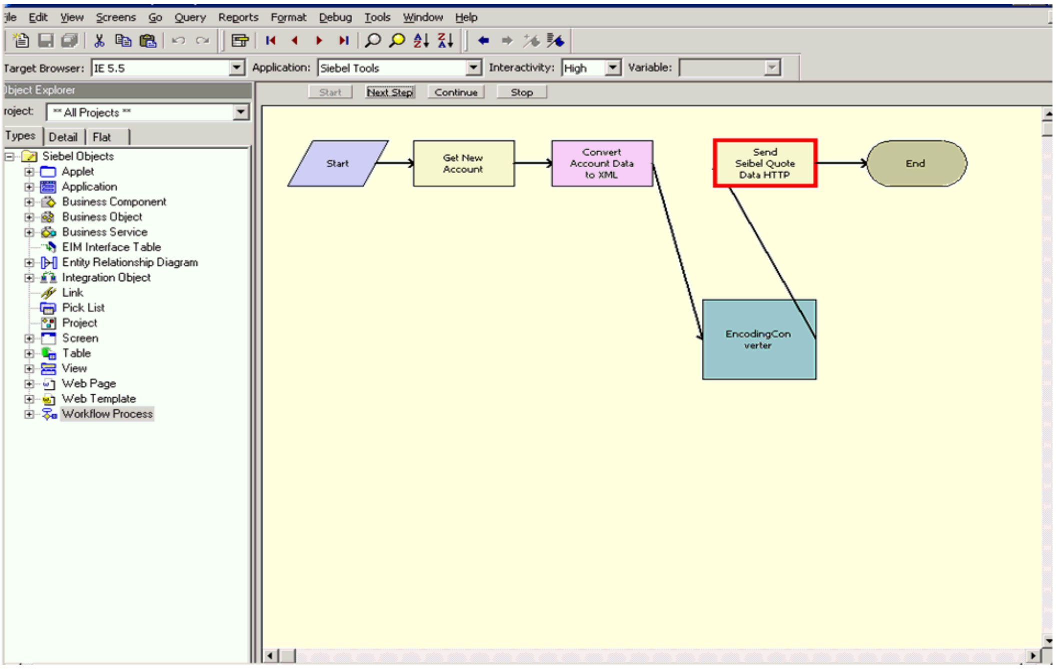
15. Click **Next Step**. The Encoding Converter image is highlighted, as shown in Figure 4–97.

Figure 4–97 Encoding Converter



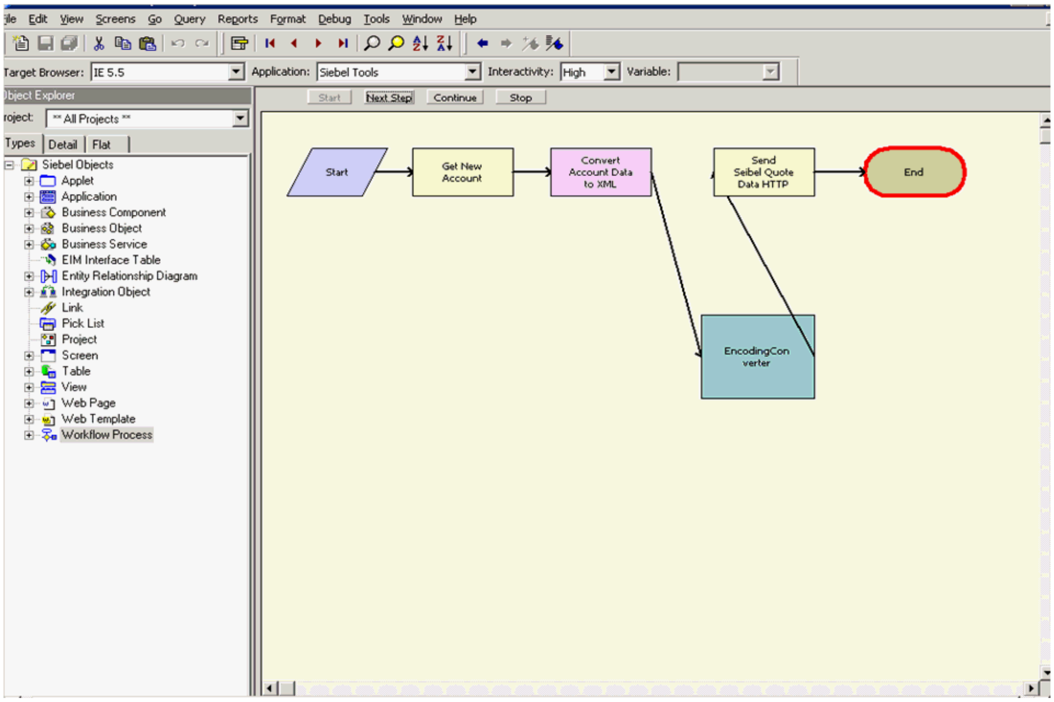
16. Click **Next Step**. The Send Siebel Quote Data HTTP image is highlighted, as shown in Figure 4–98.

Figure 4–98 Send Siebel Quote Data HTTP



17. Click **Next Step**. The End image is highlighted, as shown in Figure 4–99.

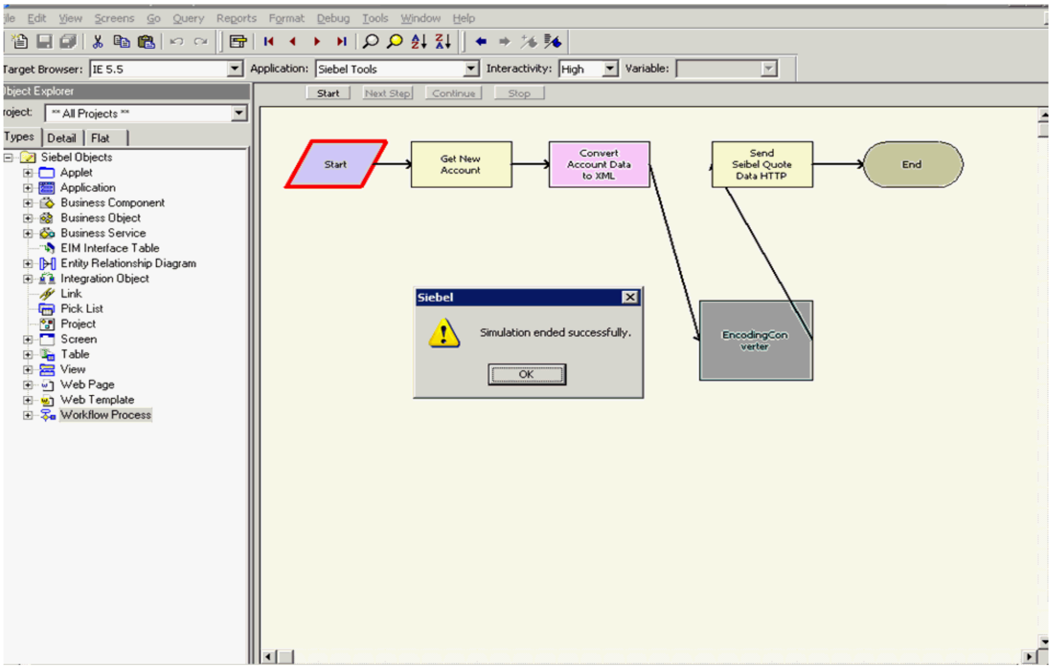
Figure 4–99 End



Target Browser: IE 5.5
Application: Siebel Tools
Interactivity: High

18. Click **Next Step**. A success message is displayed, confirming that triggering has been completed successfully, as shown in Figure 4–100.

Figure 4–100 Success Message



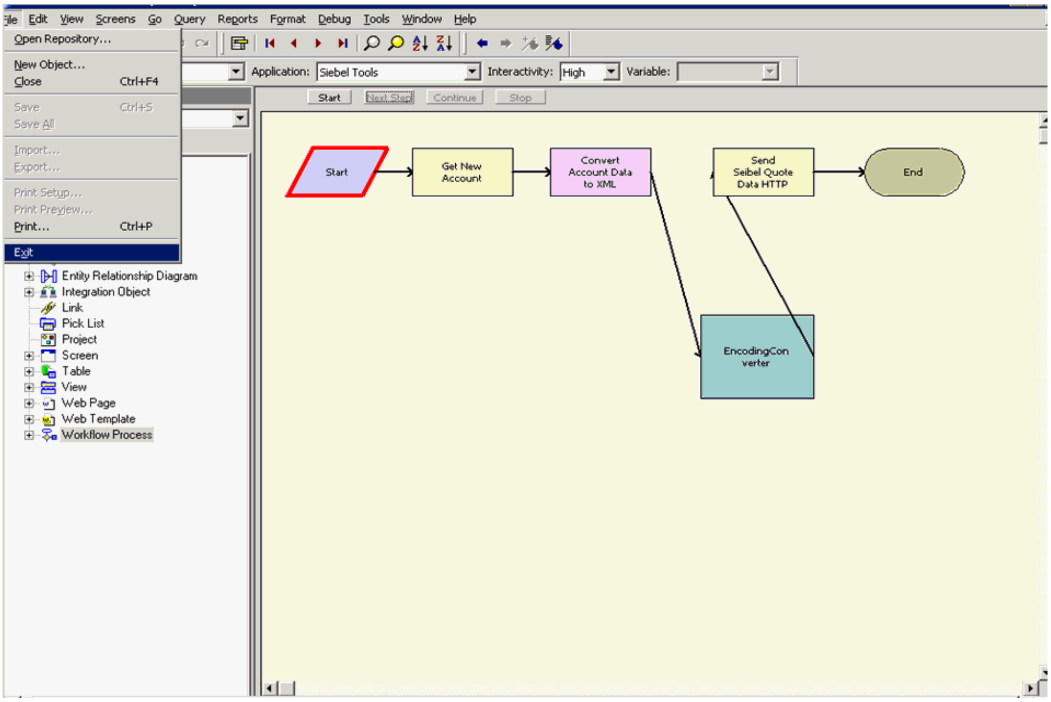
Siebel

Simulation ended successfully.

19. Click **OK**.

20. Click the **File** menu and select **Exit**, as shown in Figure 4–101.

Figure 4–101 Exit Option



4.5.5.3 Triggering an Event in Siebel 8.0 to Test Event Runtime Integration

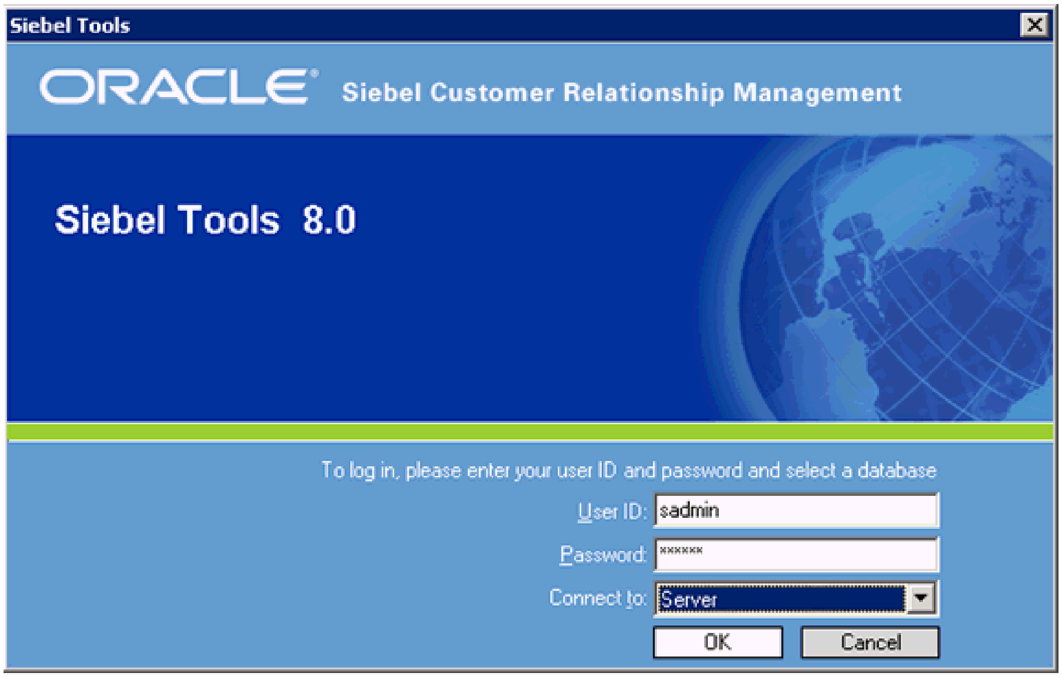
To trigger an event in Siebel 8.0:

1. Log in to Siebel Tools 8.0 by using the following parameters:

```
Username = sadmin

Password = sadmin
```

2. Choose **Server** from the Connect to list and click **OK** as shown in Figure 4–102.

Figure 4–102 Siebel Tools 8.0 Log-in Pane



To log in, please enter your user ID and password and select a database

User ID: sadmin

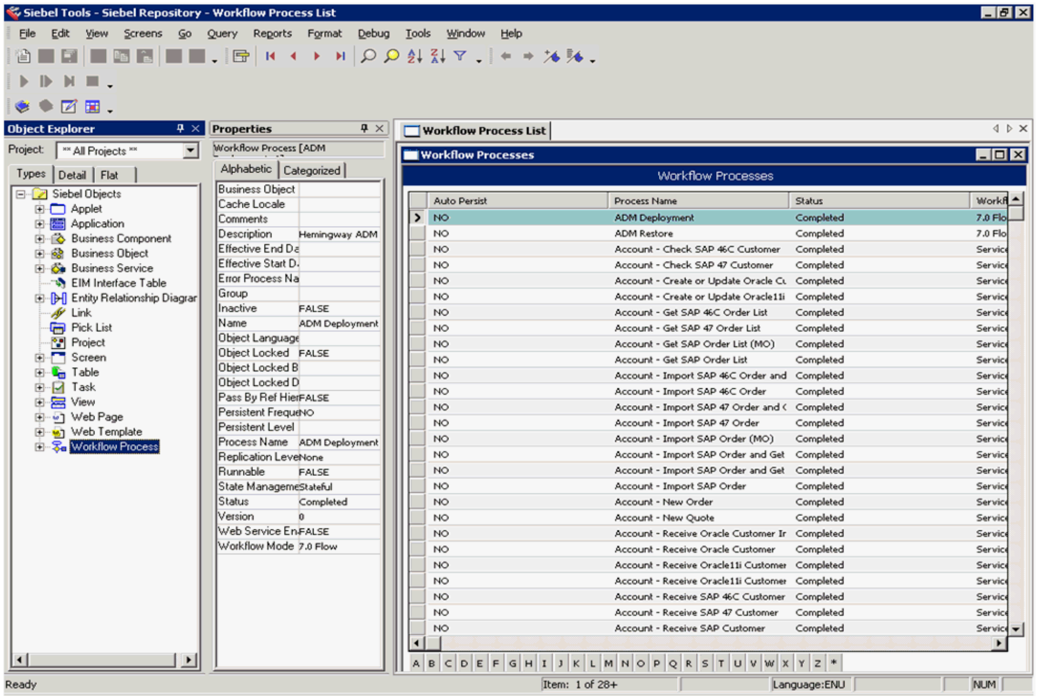
Password: ******

Connect to: Server

3. Click **Workflow Process** on the left pane.

   The Workflow Process List is displayed on the right pane, as shown in Figure 4–103.

Figure 4–103 Workflow Process List



Business Object	
Cache Locale	
Comments	
Description	Hemingway ADM
Effective End Da	
Effective Start D.	
Error Process Na	
Group	
Inactive	FALSE
Name	ADM Deployment
Object Language	
Object Locked	FALSE
Object Locked B	
Object Locked D	
Pass By Ref Hier	FALSE
Persistent Freque	NO
Persistent Level	
Process Name	ADM Deployment
Replication Leve	None
Runnable	FALSE
State Manageme	Stateful
Status	Completed
Version	0
Web Service En	FALSE
Workflow Mode	7.0 Flow

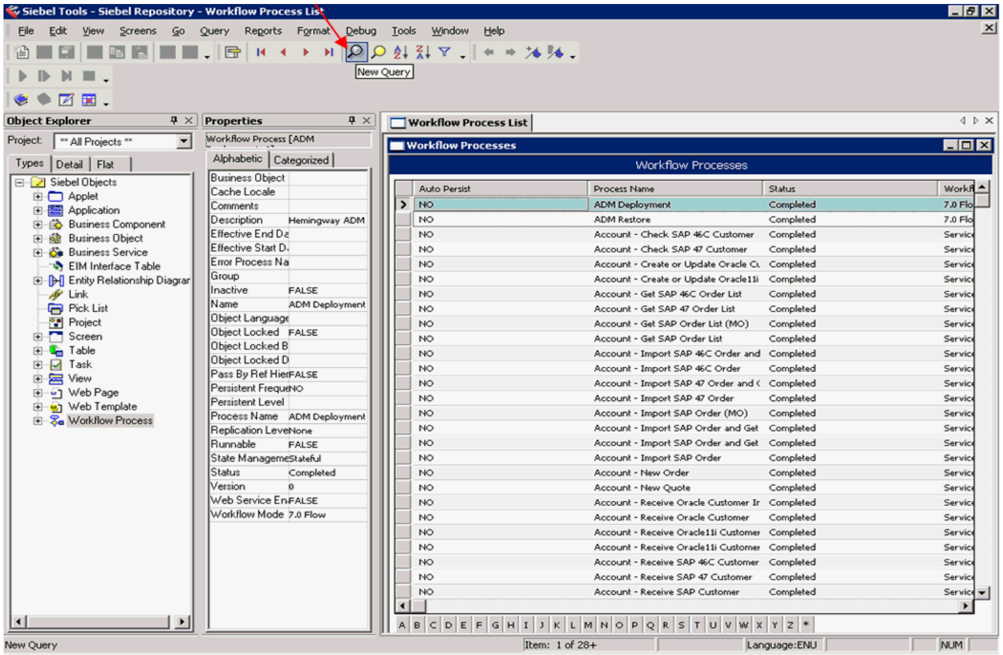
Workflow Processes

Auto Persist	Process Name	Status	Workfl
NO	ADM Deployment	Completed	7.0 Flo
NO	ADM Restore	Completed	7.0 Flo
NO	Account - Check SAP 46C Customer	Completed	Servic
NO	Account - Check SAP 47 Customer	Completed	Servic
NO	Account - Create or Update Oracle Cu	Completed	Servic
NO	Account - Create or Update Oracle11i	Completed	Servic
NO	Account - Get SAP 46C Order List	Completed	Servic
NO	Account - Get SAP 47 Order List	Completed	Servic
NO	Account - Get SAP Order List (MO)	Completed	Servic
NO	Account - Get SAP Order List	Completed	Servic
NO	Account - Import SAP 46C Order and	Completed	Servic
NO	Account - Import SAP 46C Order	Completed	Servic
NO	Account - Import SAP 47 Order and (	Completed	Servic
NO	Account - Import SAP 47 Order	Completed	Servic
NO	Account - Import SAP Order (MO)	Completed	Servic
NO	Account - Import SAP Order and Get	Completed	Servic
NO	Account - Import SAP Order and Get	Completed	Servic
NO	Account - Import SAP Order	Completed	Servic
NO	Account - New Order	Completed	Servic
NO	Account - New Quote	Completed	Servic
NO	Account - Receive Oracle Customer Ir	Completed	Servic
NO	Account - Receive Oracle Customer	Completed	Servic
NO	Account - Receive Oracle11i Customer	Completed	Servic
NO	Account - Receive Oracle11i Customer	Completed	Servic
NO	Account - Receive SAP 46C Customer	Completed	Servic
NO	Account - Receive SAP 47 Customer	Completed	Servic
NO	Account - Receive SAP Customer	Completed	Servic

Item: 1 of 28+ Language:ENU NUM

4. Click the **New Query** magnifying tool icon with the white glow, as shown in Figure 4–104.

Figure 4–104 New Query Magnifying Tool Icon

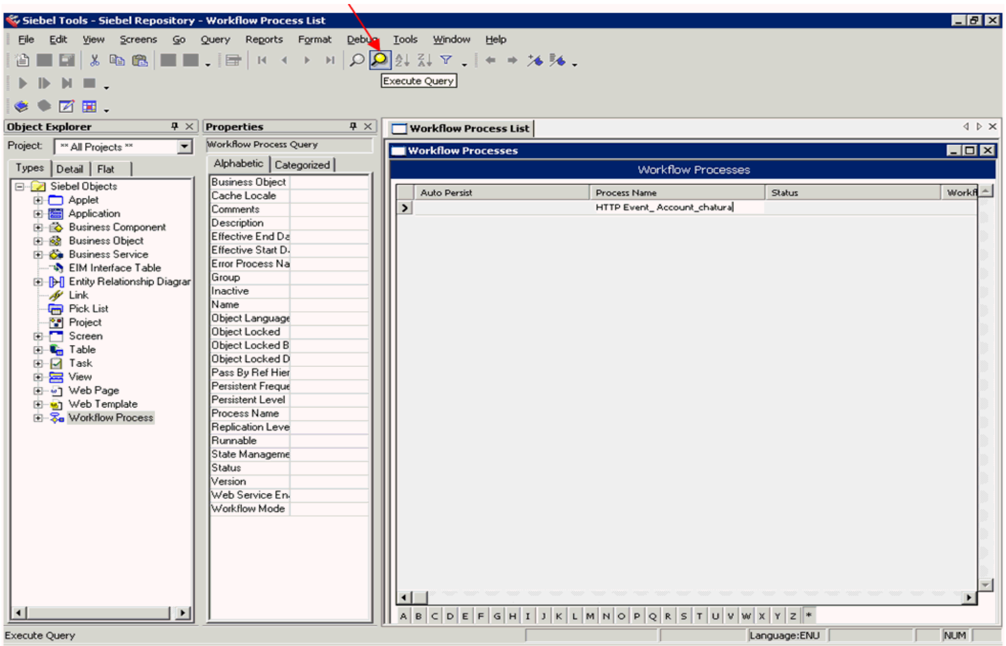


Business Object	
Cache Locale	
Comments	
Description	Hemingway ADM
Effective End Da	
Effective Start D.	
Error Process Na	
Group	
Inactive	FALSE
Name	ADM Deployment
Object Language	
Object Locked	FALSE
Object Locked B	
Object Locked D	
Pass By Ref Hier	FALSE
Persistent Freque	NO
Persistent Level	
Process Name	ADM Deployment
Replication Leve	None
Runnable	FALSE
State Manageme	Stateful
Status	Completed
Version	0
Web Service En	FALSE
Workflow Mode	7.0 Flow

Auto Persist	Process Name	Status	Workfl
NO	ADM Deployment	Completed	7.0 Flo
NO	ADM Restore	Completed	7.0 Flo
NO	Account - Check SAP 46C Customer	Completed	Servic
NO	Account - Check SAP 47 Customer	Completed	Servic
NO	Account - Create or Update Oracle Cu	Completed	Servic
NO	Account - Create or Update Oracle11i	Completed	Servic
NO	Account - Get SAP 46C Order List	Completed	Servic
NO	Account - Get SAP 47 Order List	Completed	Servic
NO	Account - Get SAP Order List (MO)	Completed	Servic
NO	Account - Get SAP Order List	Completed	Servic
NO	Account - Import SAP 46C Order and	Completed	Servic
NO	Account - Import SAP 46C Order	Completed	Servic
NO	Account - Import SAP 47 Order and (	Completed	Servic
NO	Account - Import SAP 47 Order	Completed	Servic
NO	Account - Import SAP Order (MO)	Completed	Servic
NO	Account - Import SAP Order and Get	Completed	Servic
NO	Account - Import SAP Order and Get	Completed	Servic
NO	Account - Import SAP Order	Completed	Servic
NO	Account - New Order	Completed	Servic
NO	Account - New Quote	Completed	Servic
NO	Account - Receive Oracle Customer Ir	Completed	Servic
NO	Account - Receive Oracle Customer	Completed	Servic
NO	Account - Receive Oracle11i Customer	Completed	Servic
NO	Account - Receive Oracle11i Customer	Completed	Servic
NO	Account - Receive SAP 46C Customer	Completed	Servic
NO	Account - Receive SAP 47 Customer	Completed	Servic
NO	Account - Receive SAP Customer	Completed	Servic

5. Enter the process name `HTTP Event_Account_chatura` and click the **Execute Query** magnifying tool icon with the yellow glow to execute the query, as shown in Figure 4–105.

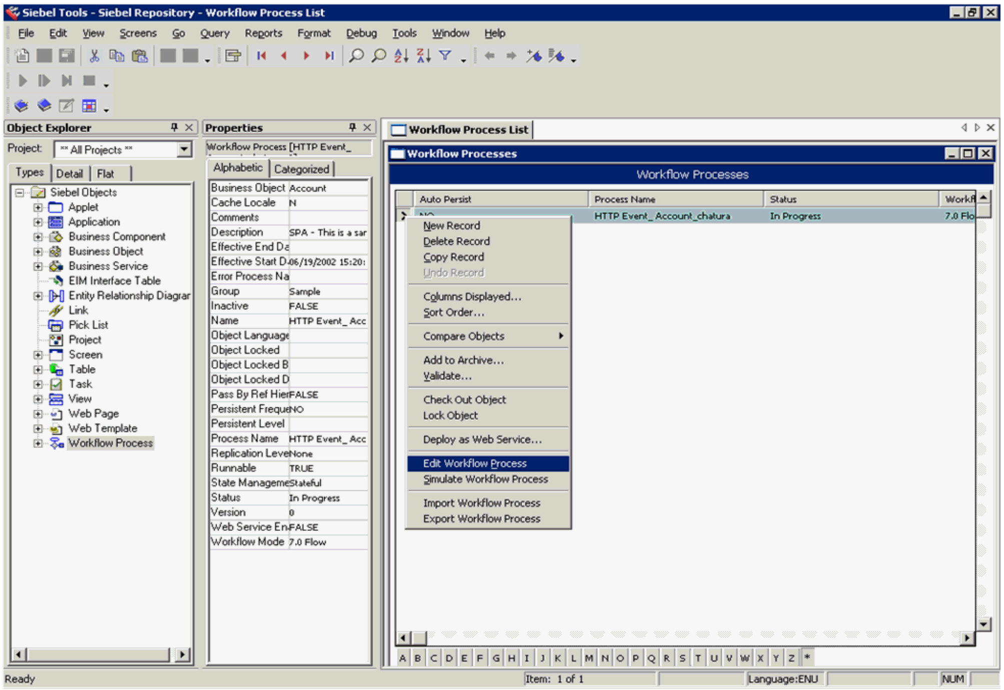
Figure 4–105 Execute Query Icon



Auto Persist	Process Name	Status	Workfl
	HTTP Event_ Account_chatura		

6. Right-click the arrow next to the selected process and select **Edit Workflow Process**, as shown in Figure 4–106.

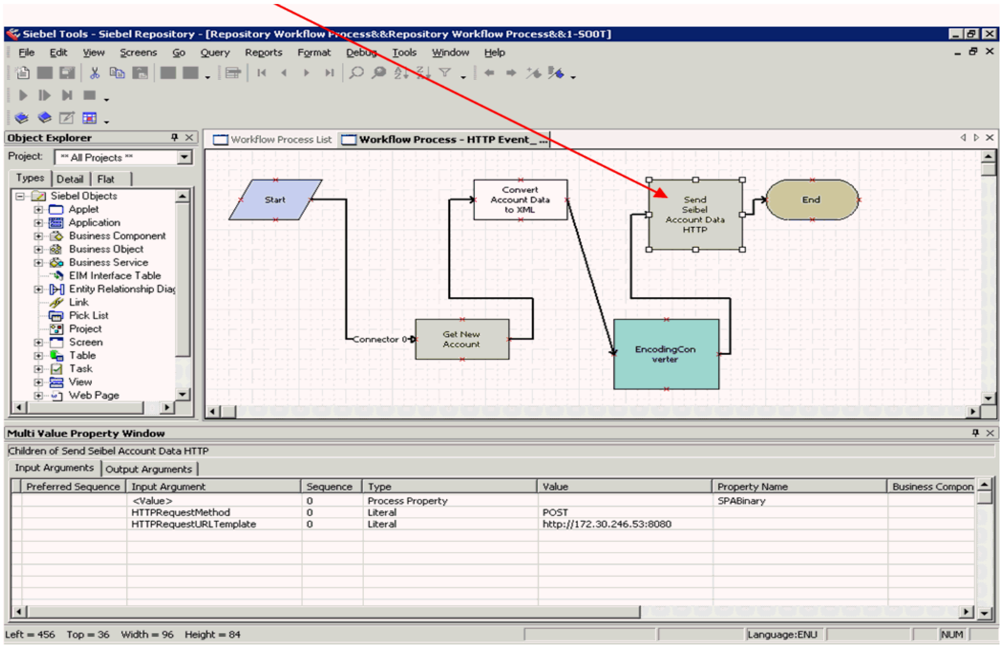
Figure 4–106 Edit Workflow Process Option



Business Object	Account
Cache Locale	N
Comments	
Description	SPA - This is a sar
Effective End Da	
Effective Start D	06/19/2002 15:20:
Error Process Na	
Group	Sample
Inactive	FALSE
Name	HTTP Event_ Acc
Object Language	
Object Locked	
Object Locked B	
Object Locked D	
Pass By Ref Hier	FALSE
Persistent Freque	NO
Persistent Level	
Process Name	HTTP Event_ Acc
Replication Leve	None
Runnable	TRUE
State Manageme	Stateful
Status	In Progress
Version	0
Web Service En	FALSE
Workflow Mode	7.0 Flow

7. Click the **Send Siebel Account Data HTTP** box, as shown in Figure 4–107.

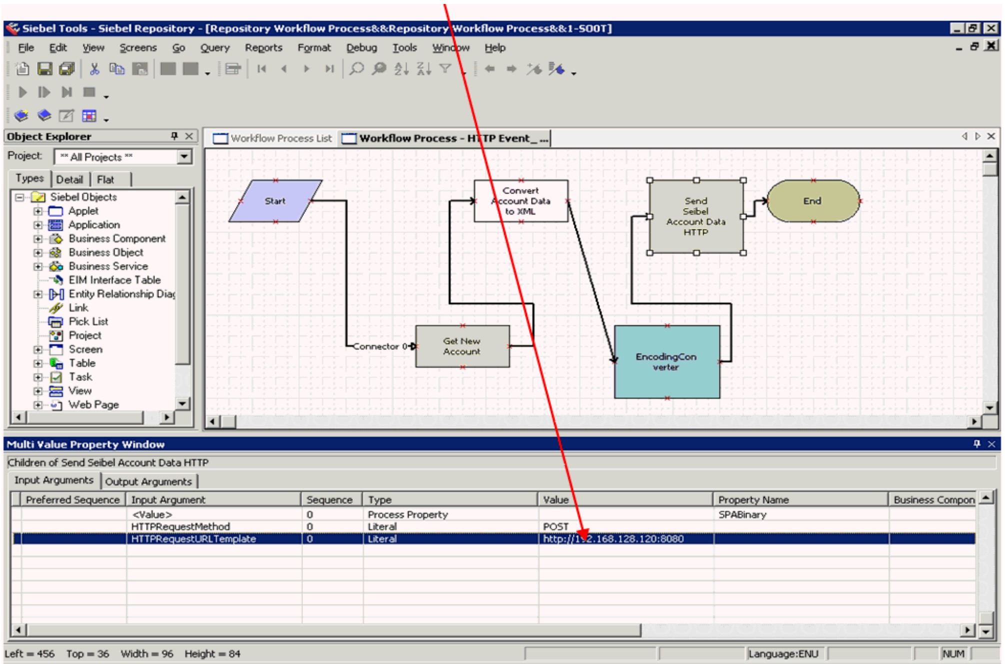
Figure 4–107 Send Siebel Account Data HTTP Box


Siebel Tools - Siebel Repository - [Repository Workflow Process&&Repository Workflow Process&&1-S0OT]
File Edit View Screens Go Query Reports Format Debug Tools Window Help
Object Explorer
Project: ** All Projects **
Types Detail Flat
Siebel Objects
Applet
Application
Business Component
Business Object
Business Service
EIM Interface Table
Entity Relationship Diag
Link
Pick List
Project
Screen
Table
Task
View
Web Page
Workflow Process List
Workflow Process - HTTP Event_...
Start
Convert Account Data to XML
Send Seibel Account Data HTTP
End
Connector 0
Get New Account
EncodingConverter
Multi Value Property Window
Children of Send Seibel Account Data HTTP
Input Arguments
Output Arguments
Left = 456 Top = 36 Width = 96 Height = 84
Language:ENU
NUM

Preferred Sequence	Input Argument	Sequence	Type	Value	Property Name	Business Compon
	<Value>	0	Process Property		SPABinary	
	HTTPRequestMethod	0	Literal	POST		
	HTTPRequestURLTemplate	0	Literal	http://172.30.246.53:8080		

8. In the Multi Value Property Window at the bottom, enter the value for **HTTPRequestURLTemplate** as `http://machineIP:portno` then save the values, as shown in Figure 4–108.

Figure 4–108 HTTPRequestURLTemplate Value


Siebel Tools - Siebel Repository - [Repository Workflow Process&&Repository Workflow Process&&1-500T]
File Edit View Screens Go Query Reports Format Debug Tools Window Help
Object Explorer
Project: ** All Projects **
Types Detail Flat
Siebel Objects
Applet
Application
Business Component
Business Object
Business Service
EIM Interface Table
Entity Relationship Diag
Link
Pick List
Project
Screen
Table
Task
View
Web Page
Workflow Process List
Workflow Process - HTTP Event_...
Start
Convert Account Data to XML
Send Seibel Account Data HTTP
End
Connector 0
Get New Account
EncodingConverter
Left = 456 Top = 36 Width = 96 Height = 84
Language:ENU
NUM

Multi Value Property Window

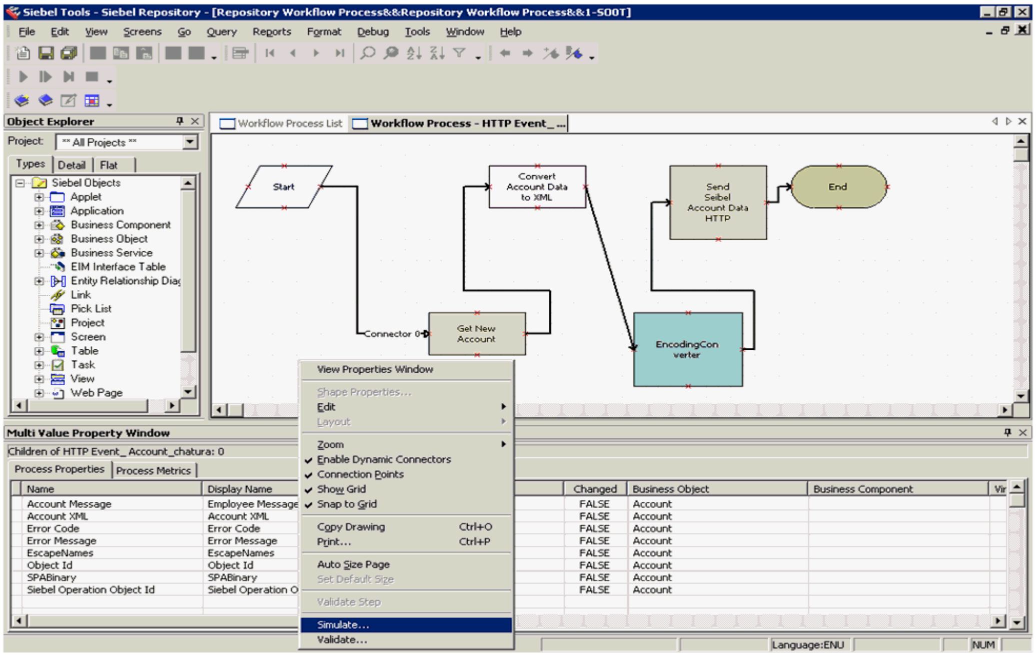
Children of Send Seibel Account Data HTTP

Input Arguments | Output Arguments

Preferred Sequence	Input Argument	Sequence	Type	Value	Property Name	Business Compon
	<Value>	0	Process Property		SPABinary	
	HTTPRequestMethod	0	Literal	POST		
	HTTPRequestURLTemplate	0	Literal	http://192.168.128.120:8080		

9. Right-click the diagram and select **Simulate**, as shown in Figure 4–109.

Figure 4–109 Simulate



Multi Value Property Window

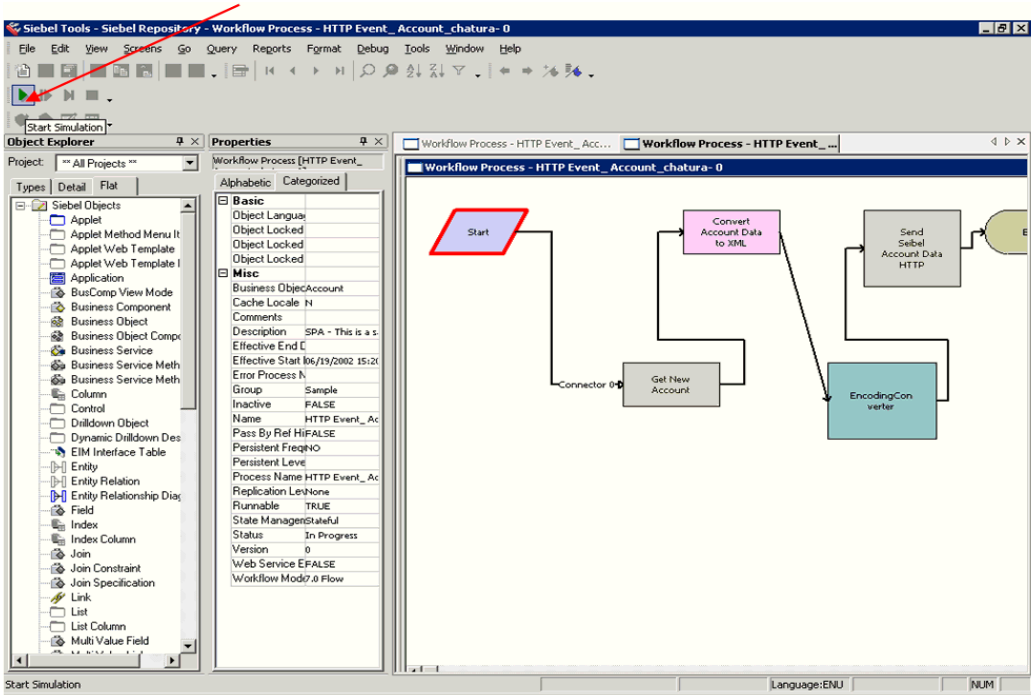
Children of HTTP Event_ Account_chatura: 0

Process Properties | Process Metrics

Name	Display Name	Changed	Business Object	Business Component	Vir
Account Message	Employee Message	FALSE	Account		
Account XML	Account XML	FALSE	Account		
Error Code	Error Code	FALSE	Account		
Error Message	Error Message	FALSE	Account		
EscapeNames	EscapeNames	FALSE	Account		
Object Id	Object Id	FALSE	Account		
SPABinary	SPABinary	FALSE	Account		
Siebel Operation Object Id	Siebel Operation O	FALSE	Account		

10. Click the **Start Simulation** icon, as shown in Figure 4–110.

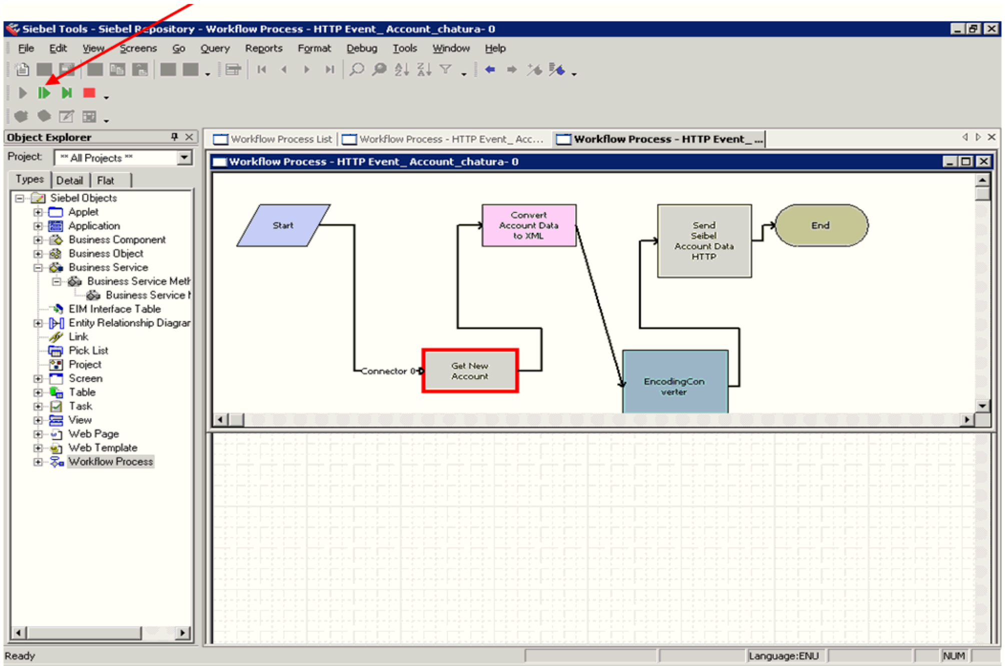
Figure 4–110 Start Simulation



Basic	
Object Langua	
Object Locked	
Object Locked	
Object Locked	
Misc	
Business Objec	Account
Cache Locale	N
Comments	
Description	SPA - This is a s.
Effective End D	
Effective Start	06/19/2002 15:2(
Error Process N	
Group	Sample
Inactive	FALSE
Name	HTTP Event_ Ac
Pass By Ref Hi	FALSE
Persistent Freq	NO
Persistent Leve	
Process Name	HTTP Event_ Ac
Replication Lev	None
Runnable	TRUE
State Managen	Stateful
Status	In Progress
Version	0
Web Service E	FALSE
Workflow Mode	7.0 Flow

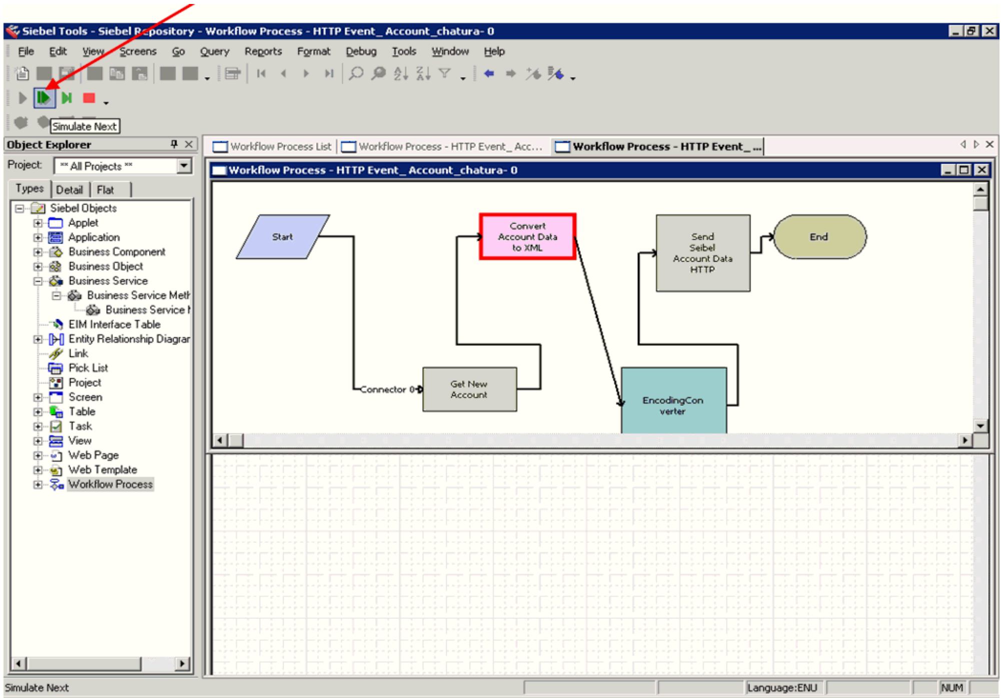
11. Click the **Simulate Next** icon. The Get New Account box is highlighted, as shown in Figure 4–111.

Figure 4–111 Simulate Next


Siebel Tools - Siebel Repository - Workflow Process - HTTP Event_ Account_chatura- 0
File Edit View Screens Go Query Reports Format Debug Tools Window Help
Object Explorer
Project: ** All Projects **
Types Detail Flat
Siebel Objects
Applet
Application
Business Component
Business Object
Business Service
Business Service Meth
Business Service I
EIM Interface Table
Entity Relationship Diagrar
Link
Pick List
Project
Screen
Table
Task
View
Web Page
Web Template
Workflow Process
Workflow Process List
Workflow Process - HTTP Event_ Acc...
Workflow Process - HTTP Event_...
Workflow Process - HTTP Event_ Account_chatura- 0
Start
Convert Application Account Data to XML
Send Seibel Account Data HTTP
End
Connector 0
Get New Account
EncodingConverter
Ready
Language:ENU
NUM

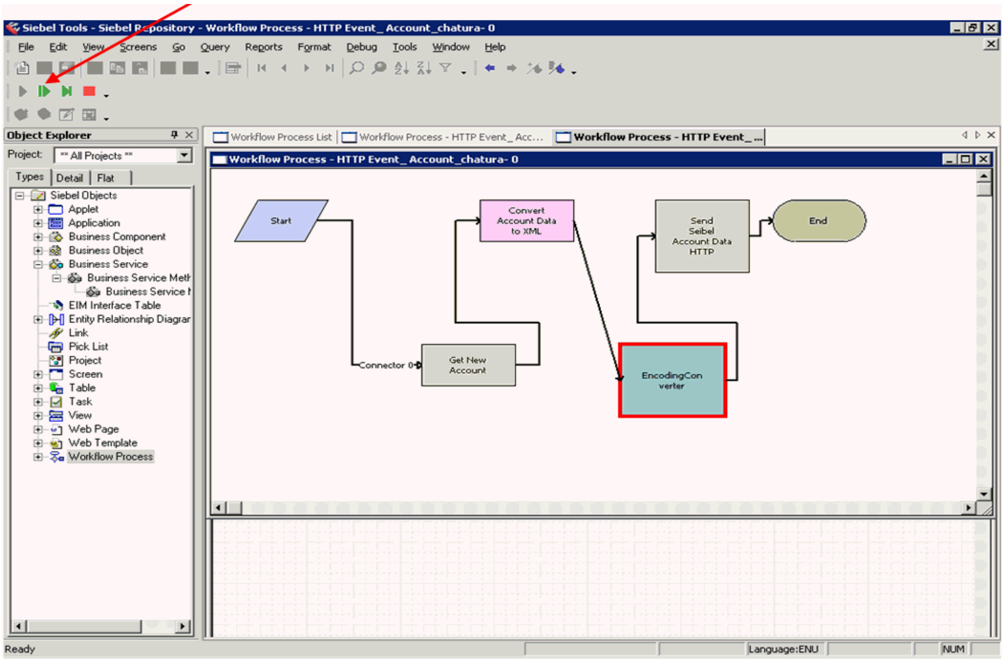
12. Click the **Simulate Next** icon. The Convert Account Data to XML box is highlighted, as shown in Figure 4–112.

Figure 4–112 Convert Account Data to XML


Siebel Tools - Siebel Repository - Workflow Process - HTTP Event_ Account_chatura- 0
File Edit View Screens Go Query Reports Format Debug Tools Window Help
Simulate Next
Object Explorer
Project: ** All Projects **
Types Detail Flat
Siebel Objects
Applet
Application
Business Component
Business Object
Business Service
Business Service Meth
Business Service I
EIM Interface Table
Entity Relationship Diagran
Link
Pick List
Project
Screen
Table
Task
View
Web Page
Web Template
Workflow Process
Workflow Process List
Workflow Process - HTTP Event_ Acc...
Workflow Process - HTTP Event_...
Workflow Process - HTTP Event_ Account_chatura- 0
Start
Convert Account Data to XML
Send Seibel Account Data HTTP
End
Connector 0
Get New Account
EncodingConverter
Simulate Next
Language:ENU
NUM

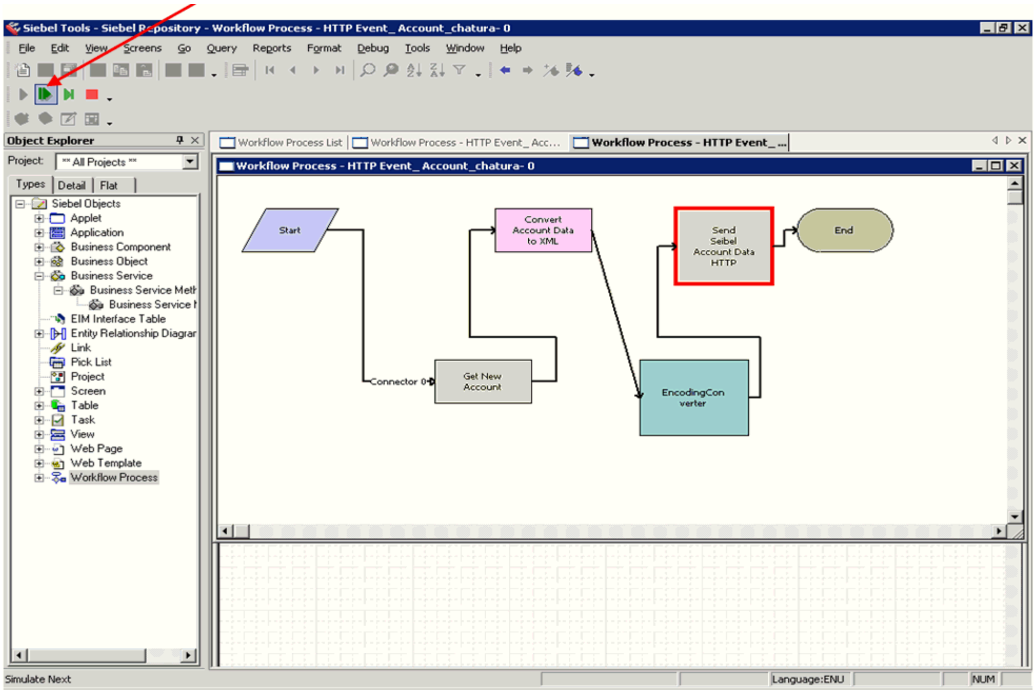
13. Click the **Simulate Next** icon. The Encoding Converter box is highlighted, as shown in Figure 4–113.

Figure 4–113 Encoding Converter


Siebel Tools - Siebel Repository - Workflow Process - HTTP Event_ Account_chatura- 0
File Edit View Screens Go Query Reports Format Debug Tools Window Help
Object Explorer
Project: ** All Projects **
Types Detail Flat
Siebel Objects
Applet
Application
Business Component
Business Object
Business Service
Business Service Meth
Business Service I
EIM Interface Table
Entity Relationship Diagran
Link
Pick List
Project
Screen
Table
Task
View
Web Page
Web Template
Workflow Process
Workflow Process List
Workflow Process - HTTP Event_ Acc...
Workflow Process - HTTP Event_...
Workflow Process - HTTP Event_ Account_chatura- 0
Start
Convert Account Data to XML
Send Seibel Account Data HTTP
End
Connector 0
Get New Account
EncodingConverter
Ready
Language:ENU
NUM

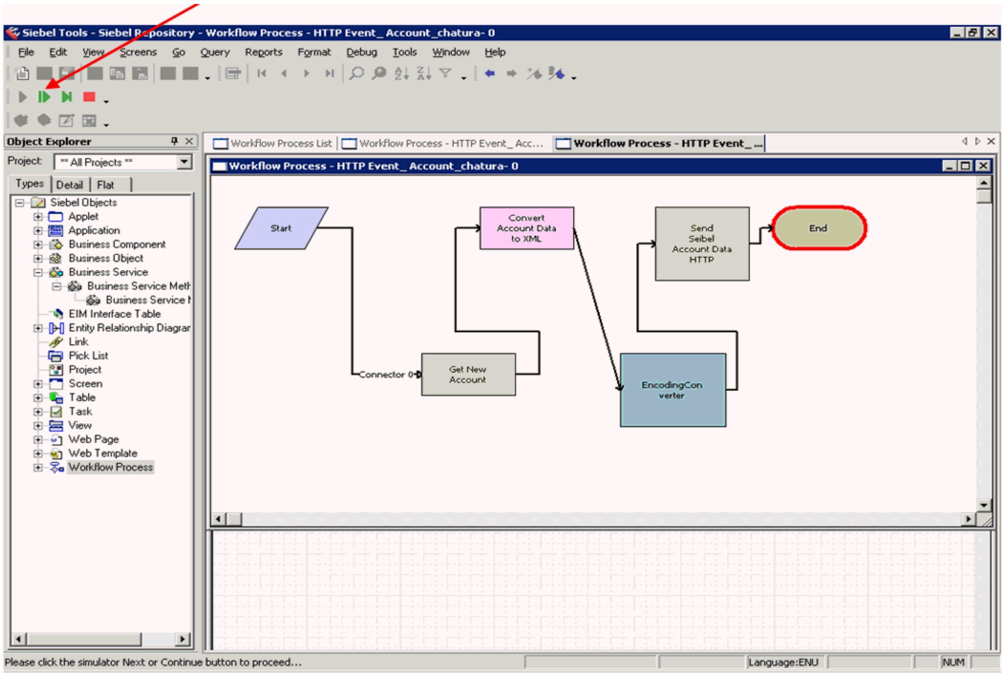
14. Click the **Simulate Next** icon. The Send Siebel Account Data HTTP box is highlighted, as shown in Figure 4–114

Figure 4–114 Send Siebel Account Data HTTP


Siebel Tools - Siebel Repository - Workflow Process - HTTP Event_ Account_chatura- 0
File Edit View Screens Go Query Reports Format Debug Tools Window Help
Object Explorer
Project: ** All Projects **
Types Detail Flat
Siebel Objects
Applet
Application
Business Component
Business Object
Business Service
Business Service Meth
Business Service I
EIM Interface Table
Entity Relationship Diagrar
Link
Pick List
Project
Screen
Table
Task
View
Web Page
Web Template
Workflow Process
Workflow Process List
Workflow Process - HTTP Event_ Acc...
Workflow Process - HTTP Event_...
Workflow Process - HTTP Event_ Account_chatura- 0
Start
Connector 0
Get New Account
Convert Account Data to XML
EncodingConverter
Send Seibel Account Data HTTP
End
Simulate Next
Language:ENU
NUM

15. Click the **Simulate Next** icon. The End image is highlighted as shown in Figure 4–115.

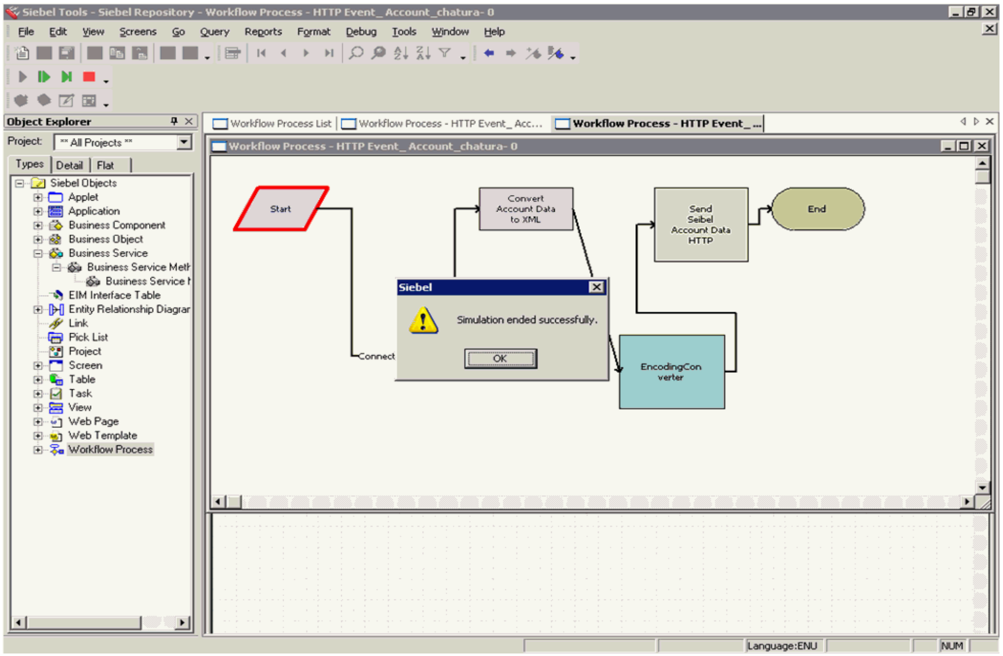
Figure 4–115 Simulate Next


Siebel Tools - Siebel Repository - Workflow Process - HTTP Event_ Account_chatura- 0
File Edit View Screens Go Query Reports Format Debug Tools Window Help
Object Explorer
Project: ** All Projects **
Types Detail Flat
Siebel Objects
Applet
Application
Business Component
Business Object
Business Service
Business Service Meth
Business Service I
EIM Interface Table
Entity Relationship Diagrar
Link
Pick List
Project
Screen
Table
Task
View
Web Page
Web Template
Workflow Process
Workflow Process List
Workflow Process - HTTP Event_ Acc...
Workflow Process - HTTP Event_...
Workflow Process - HTTP Event_ Account_chatura- 0
Start
Connector 0
Get New Account
Convert Account Data to XML
EncodingConverter
Send Seibel Account Data HTTP
End
Language:ENU
NUM

Please click the simulator Next or Continue button to proceed...

16. Click **Next Step** and then click **OK** when the Siebel success message is displayed, as shown in Figure 4–116.

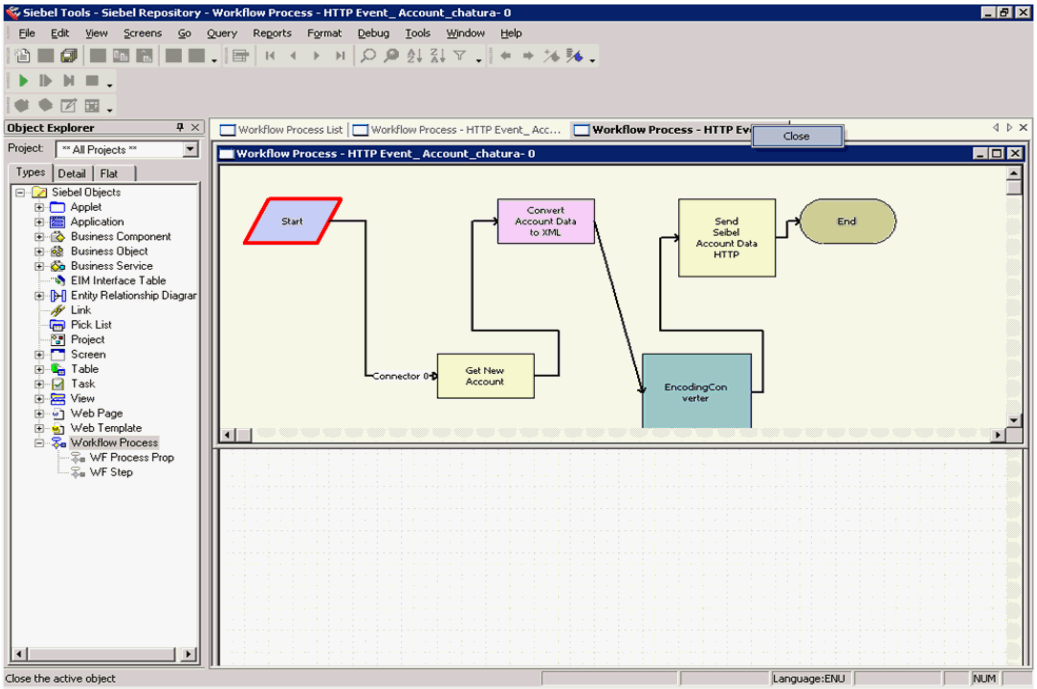
Figure 4–116 Siebel Success Message


Siebel Tools - Siebel Repository - Workflow Process - HTTP Event_ Account_chatura- 0
File Edit View Screens Go Query Reports Format Debug Tools Window Help
Object Explorer
Project: ** All Projects **
Types Detail Flat
Siebel Objects
Applet
Application
Business Component
Business Object
Business Service
Business Service Meth
Business Service I
EIM Interface Table
Entity Relationship Diagrar
Link
Pick List
Project
Screen
Table
Task
View
Web Page
Web Template
Workflow Process
Workflow Process List
Workflow Process - HTTP Event_ Acc...
Workflow Process - HTTP Event_...
Workflow Process - HTTP Event_ Account_chatura- 0
Start
Convert Account Data to XML
Send Seibel Account Data HTTP
End
Connect
EncodingConverter
Siebel
OK
Language:ENU
NUM

Simulation ended successfully.

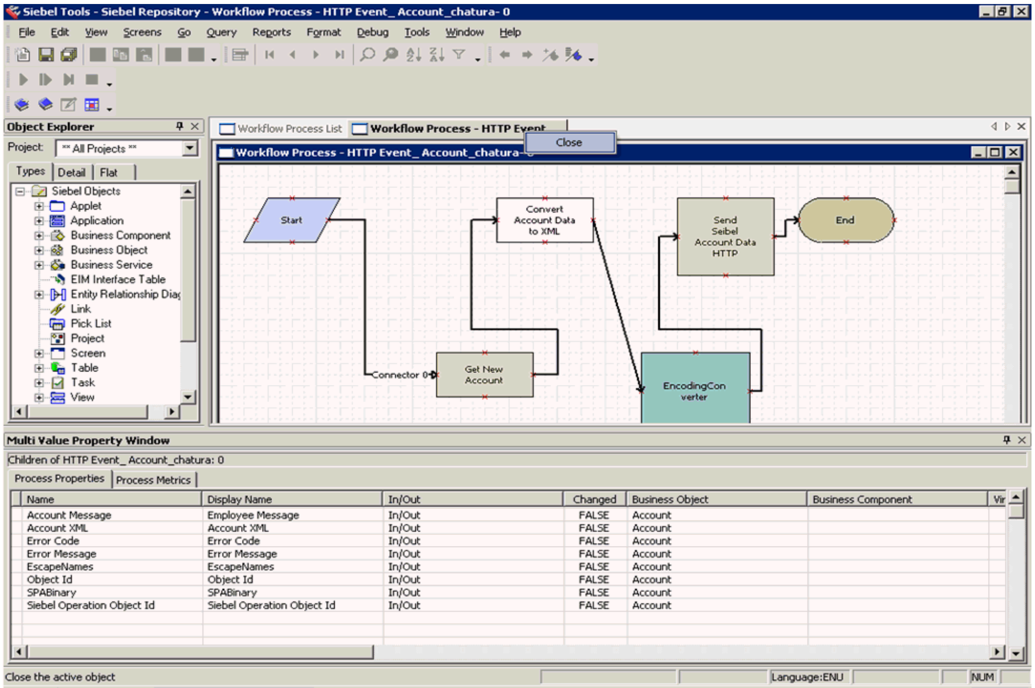
17. Right-click the third **Workflow Process** tab and select **Close**, as shown in Figure 4–117.

Figure 4–117 Workflow Process Tab


Siebel Tools - Siebel Repository - Workflow Process - HTTP Event_ Account_chatura- 0
File Edit View Screens Go Query Reports Format Debug Tools Window Help
Object Explorer
Project: ** All Projects **
Types Detail Flat
Siebel Objects
Applet
Application
Business Component
Business Object
Business Service
EIM Interface Table
Entity Relationship Diagrar
Link
Pick List
Project
Screen
Table
Task
View
Web Page
Web Template
Workflow Process
WF Process Prop
WF Step
Workflow Process List
Workflow Process - HTTP Event_ Acc...
Workflow Process - HTTP Ev
Close
Workflow Process - HTTP Event_ Account_chatura- 0
Start
Convert Account Data to XML
Send Seibel Account Data HTTP
End
Connector 0
Get New Account
EncodingConverter
Close the active object
Language:ENU
NUM

18. Right-click the second **Workflow Process** tab and select **Close**, as shown in Figure 4–118.

Figure 4–118 Workflow Process Tab

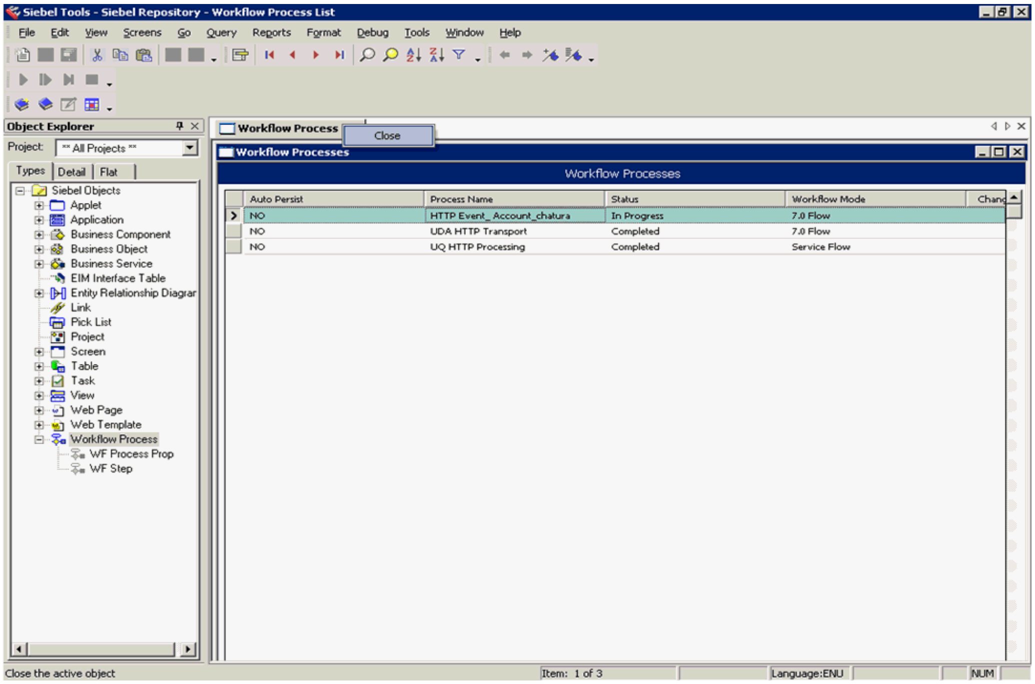


Children of HTTP Event_ Account_chatura: 0

Name	Display Name	In/Out	Changed	Business Object	Business Component	Vir
Account Message	Employee Message	In/Out	FALSE	Account		
Account XML	Account XML	In/Out	FALSE	Account		
Error Code	Error Code	In/Out	FALSE	Account		
Error Message	Error Message	In/Out	FALSE	Account		
EscapeNames	EscapeNames	In/Out	FALSE	Account		
Object Id	Object Id	In/Out	FALSE	Account		
SPABinary	SPABinary	In/Out	FALSE	Account		
Siebel Operation Object Id	Siebel Operation Object Id	In/Out	FALSE	Account		

19. Right-click the remaining **Workflow Process** tab and select **Close**, as shown in Figure 4–119.

Figure 4–119 Workflow Process Tab

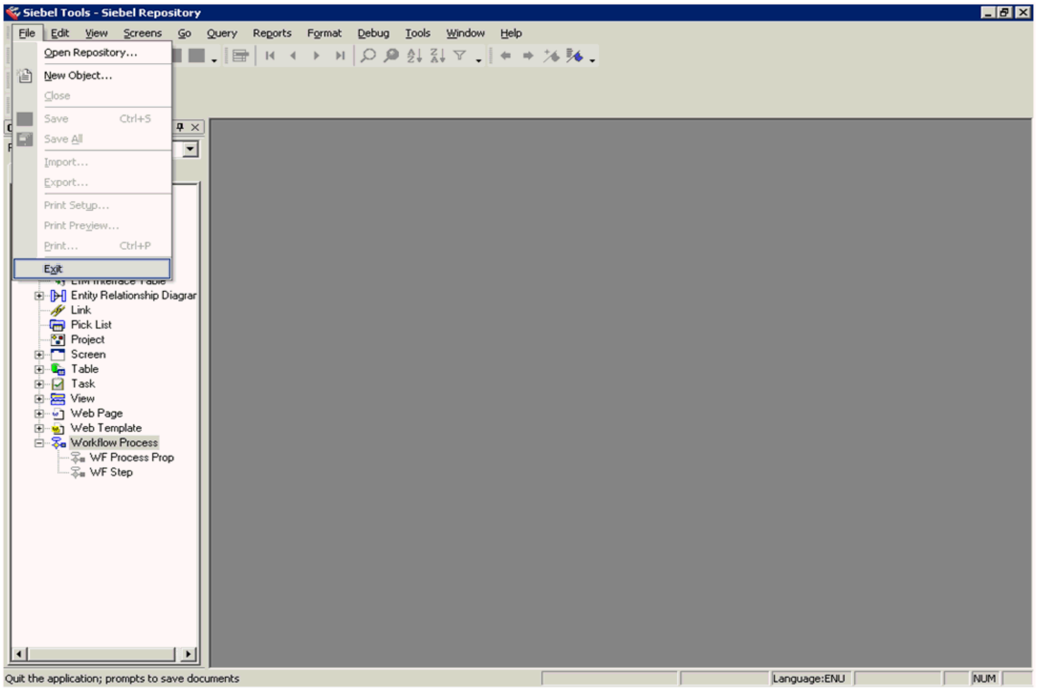


Workflow Processes

Auto Persist	Process Name	Status	Workflow Mode	Chang
NO	HTTP Event_ Account_chatura	In Progress	7.0 Flow	
NO	UDA HTTP Transport	Completed	7.0 Flow	
NO	UQ HTTP Processing	Completed	Service Flow	

20. From the File menu, click **Exit** to close the tool, as shown in Figure 4–120.

Figure 4–120 Exit



4.5.5.4 Verifying the Results

To verify your results:

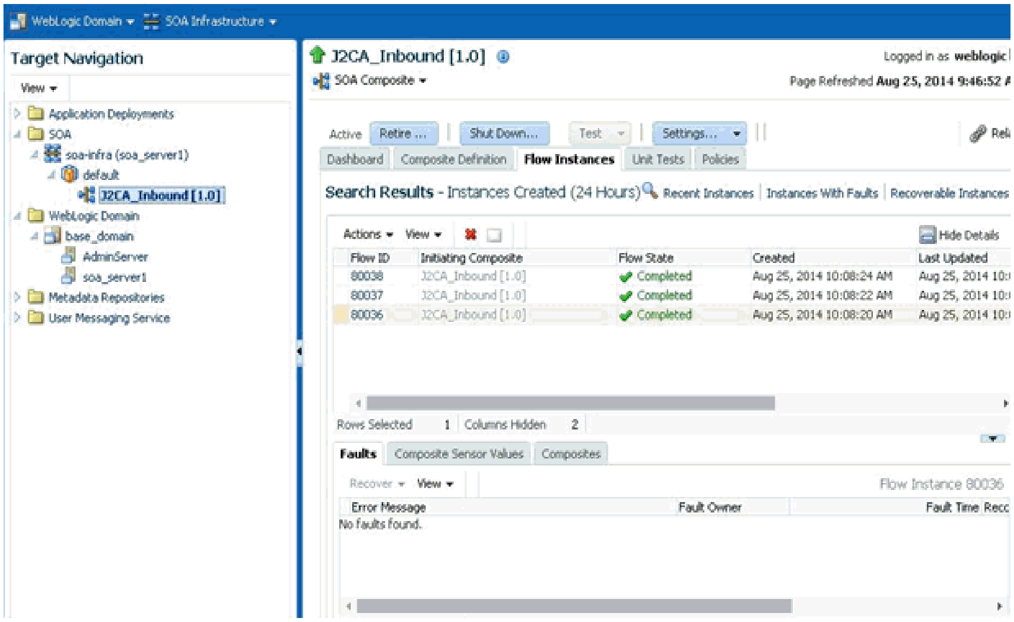
1. Log in to the Oracle Enterprise Manager console by using the following URL:

   `http://localhost:7001/em`

2. Click **SOA**, select **soa-infra (soa_server1)**, **default**, and then click **J2CA_Inbound**.
3. Click **Flow Instances**.

   Instances will be received as shown in Example 4–121.

Figure 4–121 Flow Instances Tab



Flow ID	Initiating Composite	Flow State	Created	Last Updated
80038	J2CA_Inbound [1.0]	✓ Completed	Aug 25, 2014 10:08:24 AM	Aug 25, 2014 10:
80037	J2CA_Inbound [1.0]	✓ Completed	Aug 25, 2014 10:08:22 AM	Aug 25, 2014 10:
80036	J2CA_Inbound [1.0]	✓ Completed	Aug 25, 2014 10:08:20 AM	Aug 25, 2014 10:

Error Message	Fault Owner	Fault Time	Recc
No faults found.			

4.6 Designing an Outbound BPEL Process for Service Integration (BSE Configuration)

This section describes how to design an outbound BPEL process for service integration.

A sample project has been provided for this outbound use case scenario in the following folder of the Application Adapters installation:

```
<ADAPTER_HOME>\etc\sample\SIEBEL_Samples.zip\SIEBEL_Samples\BPEL\BSE\Outbound_
Project
```

The following tools are required to complete your adapter design-time configuration:

- Oracle Adapter Application Explorer (Application Explorer)
- Oracle JDeveloper BPEL Designer (JDeveloper)

This section includes the following topics:

- Section 4.6.1, "Generating a WSDL File for Request and Response Services Using a Web Service"
- Section 4.6.2, "Creating an Empty Composite for SOA"
- Section 4.6.3, "Defining a BPEL Outbound Process"

Before you design a BPEL process, you must generate the respective WSDL file using Application Explorer. For more information, see Section 4.6.1, "Generating a WSDL File for Request and Response Services Using a Web Service".

4.6.1 Generating a WSDL File for Request and Response Services Using a Web Service

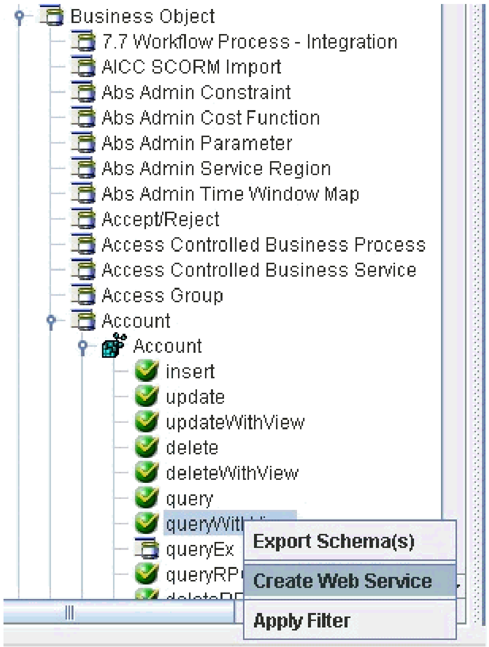
To generate a WSDL file for request and response services using a Web service:

1. Start Application Explorer and connect to a defined Siebel target (BSE configuration).

For more information on defining a target and connecting to Siebel, see Section 2.4.1, "Defining a Target to Siebel".

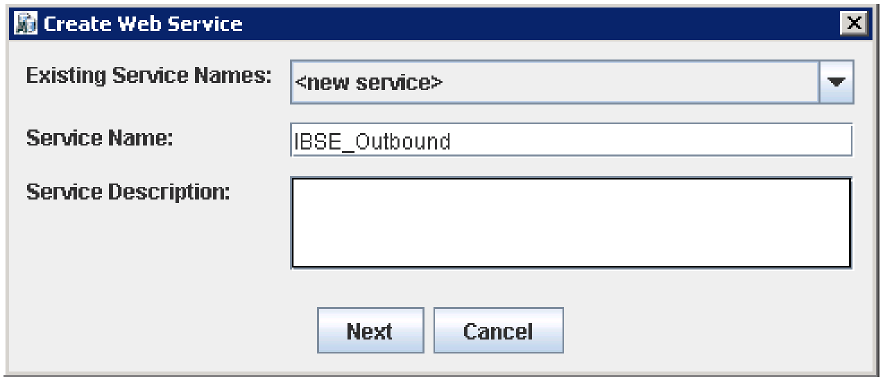
2. Expand the Siebel target to which you are connected.
3. Expand **Business Object**, **Account**, and then **Account**.
4. Right-click **queryWithView**, and then select **Create Web Service** from the menu, as shown in Figure 4–122.

Figure 4–122 queryWithView Node



The Create Web Service dialog is displayed, as shown in Figure 4–123.

Figure 4–123 Create Web Service Dialog


Create Web Service

Existing Service Names: <new service>

Service Name: IBSE_Outbound

Service Description:

5. Enter a service name, and click **Next**.
6. Click **OK** on the next dialog that is displayed.

   Application Explorer switches the view to the Business Services node, and the new Web service is displayed in the left pane.
7. Right-click the new Web service and select **Save WSDL** from the menu.

8. Save the WSDL in the wsdls folder and click **Save**.

   You can now create an empty composite for SOA, which is the first step that is required to define a BPEL outbound process in JDeveloper.

4.6.2 Creating an Empty Composite for SOA

To create an empty composite for SOA:

1. Create a new SOA application.
2. Enter a name for the SOA Application and click **Next**.

   The Name your project page is displayed.
3. Enter a project name and click **Next**.

   The Configure SOA settings page is displayed.
4. From the Composite Template list, select **Empty Composite** and click **Finish**.

   For more information, see Section 4.4.2, "Creating an Empty Composite for SOA," on page 4-9.

4.6.3 Defining a BPEL Outbound Process

This section describes how to configure a BPEL outbound process component.

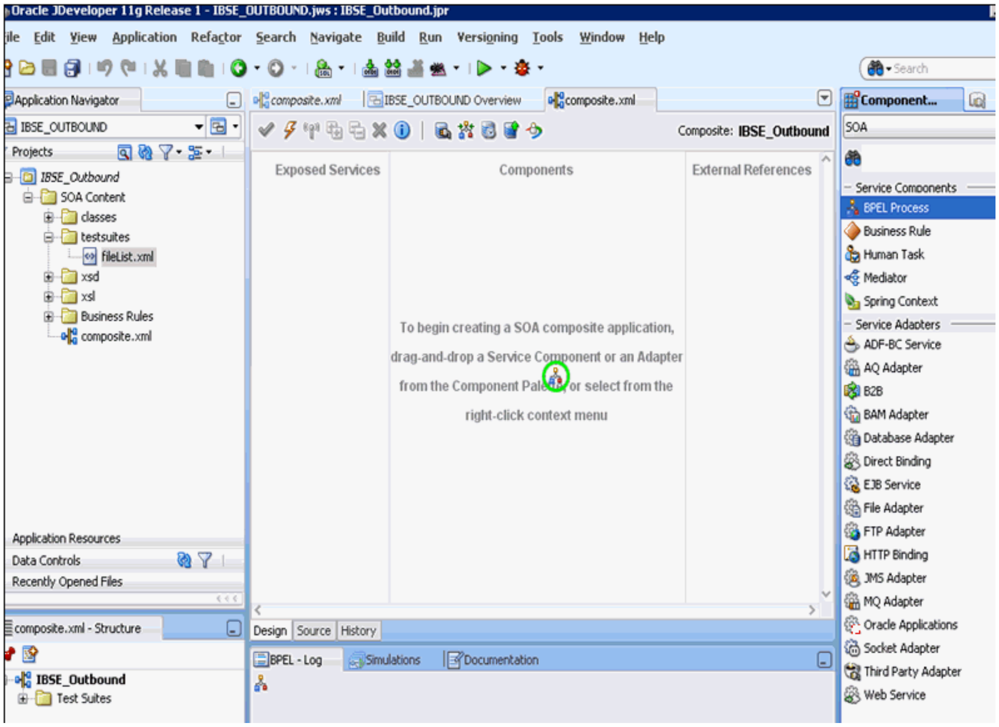
This section includes the following topics:

- Section 4.6.3.1, "Creating a Partner Link"
- Section 4.6.3.2, "Creating BPEL Activities and Mappings With the Created Partner Link"

To define a BPEL outbound process:

1. Drag and drop the **BPEL Process** component from the Service Components pane to the Components pane, as shown in Figure 4–124.

Figure 4–124 BPEL Process Component



To begin creating a SOA composite application, drag-and-drop a Service Component or an Adapter from the Component Palette or select from the right-click context menu

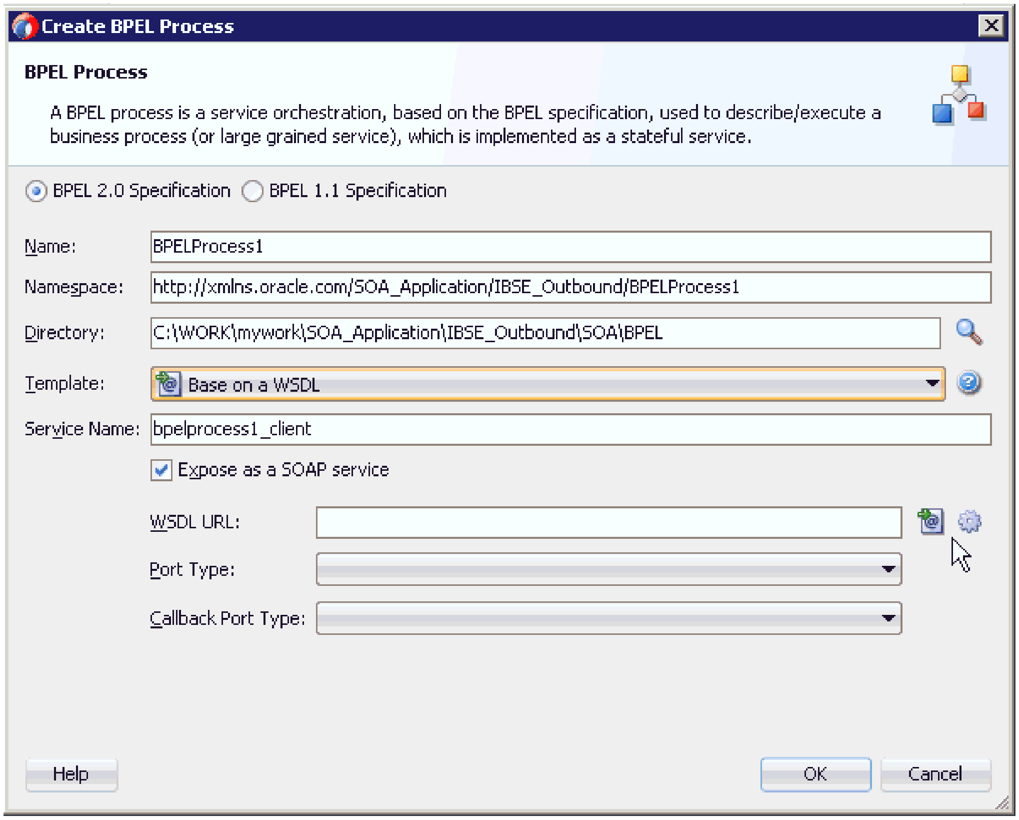
2. In the Name field, enter a name to identify the new outbound BPEL process component or leave it to the default value.

   By default, the BPEL 2.0 Specification option is selected.

3. From the Template drop-down list, select **Base on a WSDL**.

4. Click the **Find existing WSDLs** icon, which is located to the right of the WSDL URL field, as shown in Figure 4–125.

Figure 4–125 Find Existing WSDLs Icon



Create BPEL Process

BPEL Process

A BPEL process is a service orchestration, based on the BPEL specification, used to describe/execute a business process (or large grained service), which is implemented as a stateful service.

● BPEL 2.0 Specification ○ BPEL 1.1 Specification

Name: BPELProcess1

Namespace: http://xmlns.oracle.com/SOA_Application/IBSE_Outbound/BPELProcess1

Directory: C:\WORK\mywork\SOA_Application\IBSE_Outbound\SOA\BPEL

Template: Base on a WSDL

Service Name: bpelprocess1_client

☑ Expose as a SOAP service

WSDL URL:

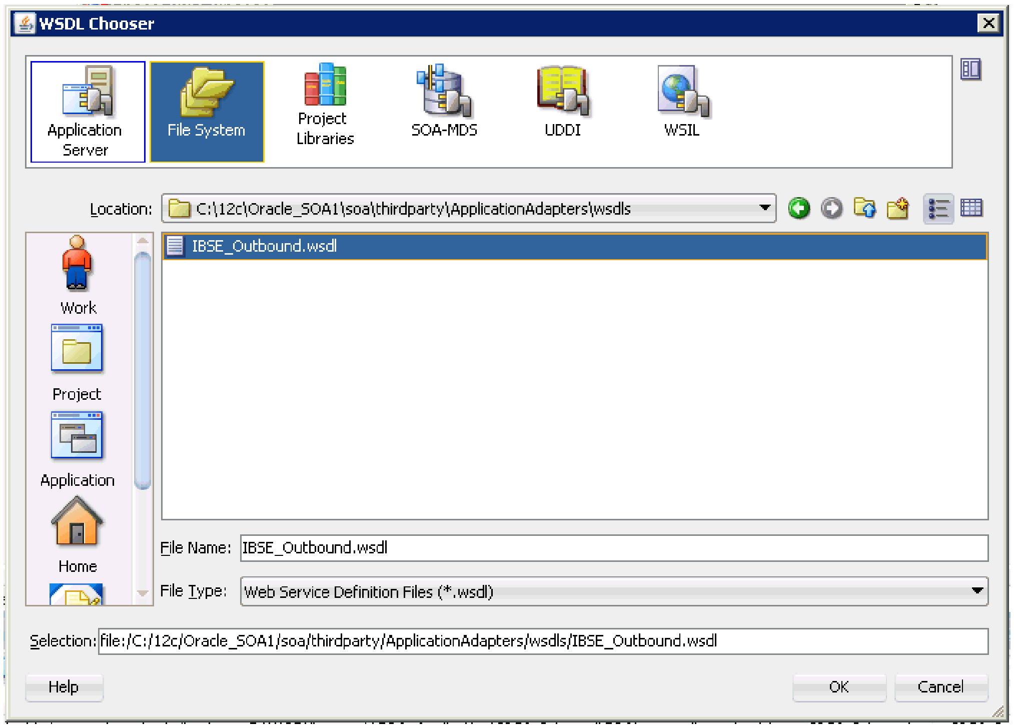
Port Type:

Callback Port Type:

The WSDL Chooser dialog is displayed.

5. Navigate to the location where the WSDL is exported from Application Explorer, select the WSDL, and click **OK**, as shown in Figure 4–126.

Figure 4–126 WSDL Chooser Dialog



Location: C:\12c\Oracle_SOA1\soa\thirdparty\ApplicationAdapters\wsdls

IBSE_Outbound.wsdl

File Name: IBSE_Outbound.wsdl

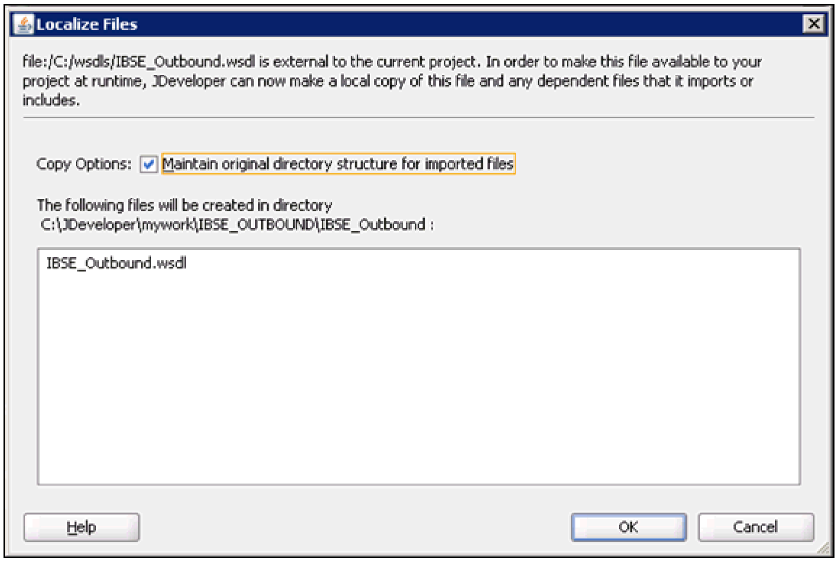
File Type: Web Service Definition Files (*.wsdl)

Selection: file:/C:/12c/Oracle_SOA1/soa/thirdparty/ApplicationAdapters/wsdls/IBSE_Outbound.wsdl

The Localize Files window is displayed.

6. In the displayed Localize Files window, click **OK**. This imports the WSDL file to the project folder, as shown in Figure 4–127.

Figure 4–127 Localize Files Window



file:/C:/wsdls/IBSE_Outbound.wsdl is external to the current project. In order to make this file available to your project at runtime, JDeveloper can now make a local copy of this file and any dependent files that it imports or includes.

Copy Options: ☑ Maintain original directory structure for imported files

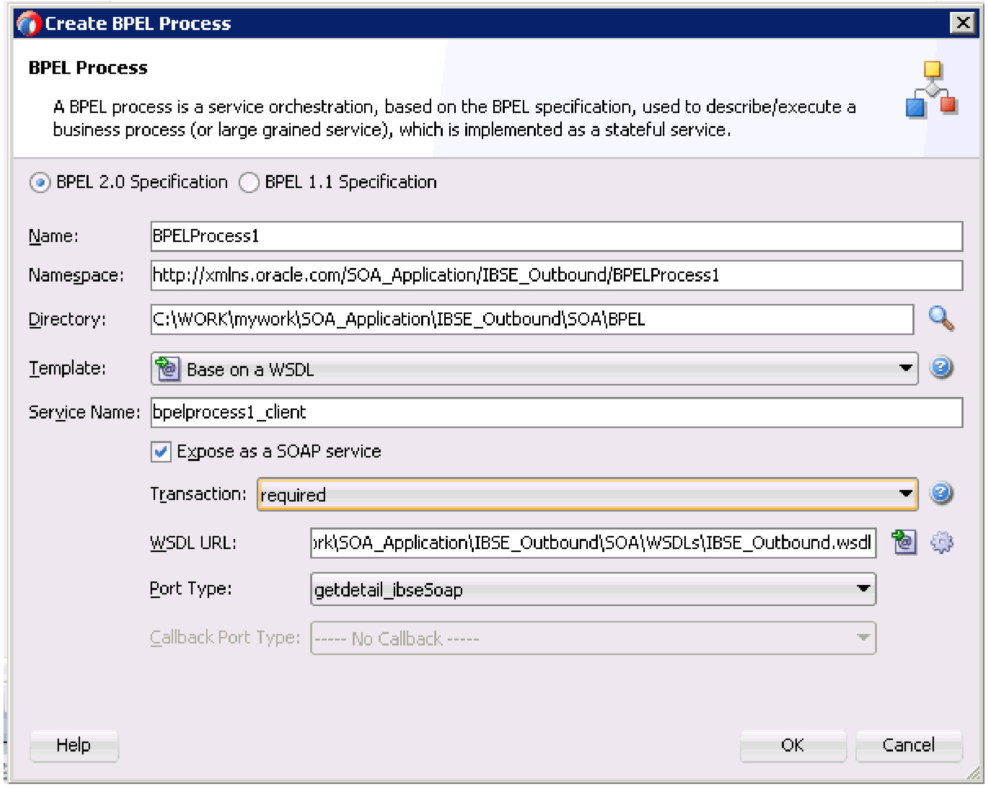
The following files will be created in directory C:\JDeveloper\mywork\IBSE_OUTBOUND\IBSE_Outbound :

IBSE_Outbound.wsdl

The Create BPEL Process window is displayed.

7. In the BPEL Process pane, click **OK**, as shown in Figure 4–128.

Figure 4–128 BPEL Process Pane



BPEL Process

A BPEL process is a service orchestration, based on the BPEL specification, used to describe/execute a business process (or large grained service), which is implemented as a stateful service.

● BPEL 2.0 Specification ○ BPEL 1.1 Specification

Name: BPELProcess1

Namespace: http://xmlns.oracle.com/SOA_Application/IBSE_Outbound/BPELProcess1

Directory: C:\WORK\mywork\SOA_Application\IBSE_Outbound\SOA\BPEL

Template: Base on a WSDL

Service Name: bpelprocess1_client

☑ Expose as a SOAP service

Transaction: required

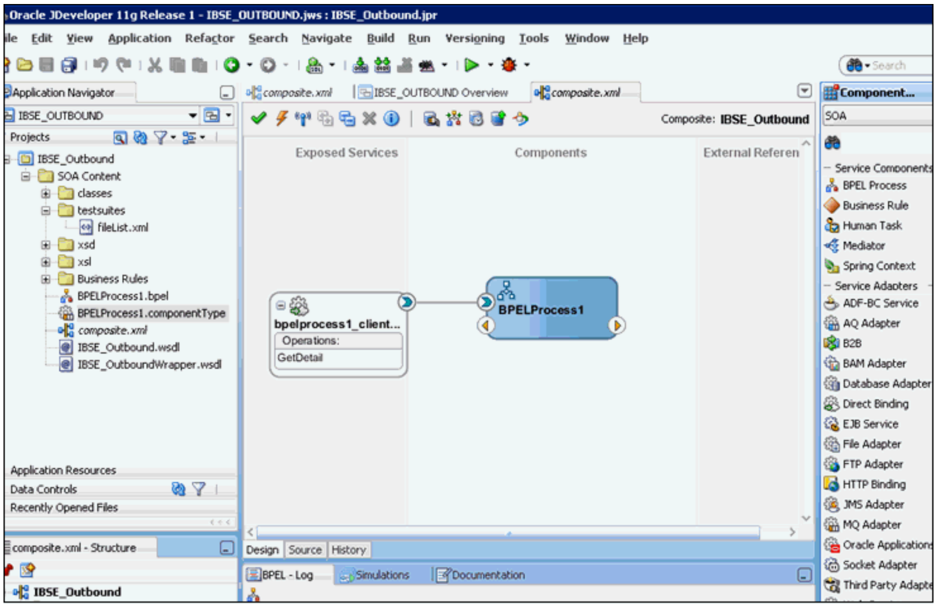
WSDL URL: rk\SOA_Application\IBSE_Outbound\SOA\WSDLs\IBSE_Outbound.wsdl

Port Type: getdetail_ibseSoap

Callback Port Type: ----- No Callback -----

The BPEL Process component is created and displayed, as shown in Figure 4–129.

Figure 4–129 BPEL Process Component



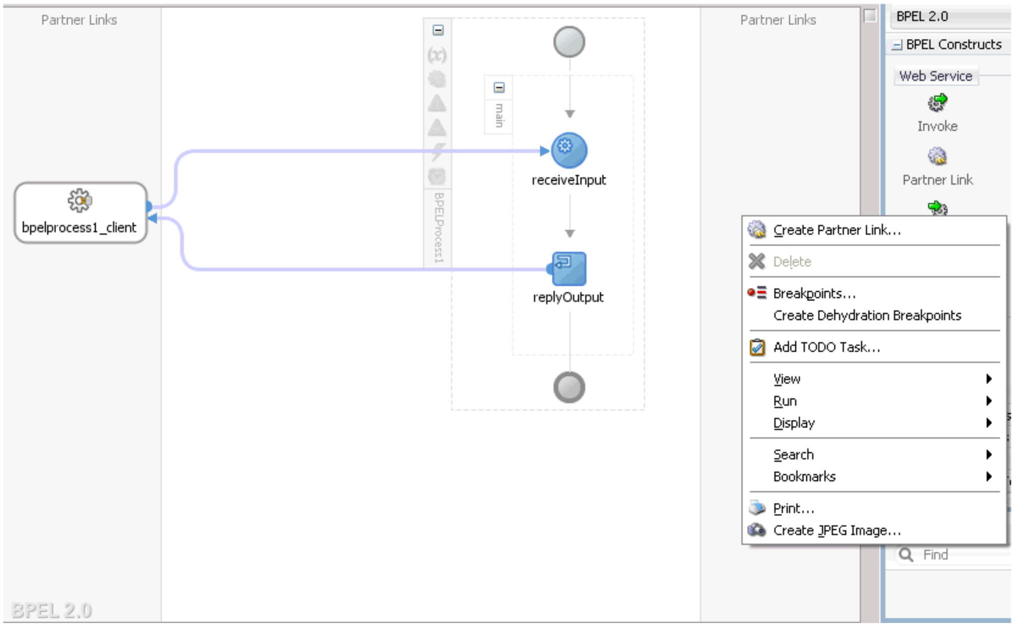
4.6.3.1 Creating a Partner Link

This section describes how to create a partner link.

To create a partner link:

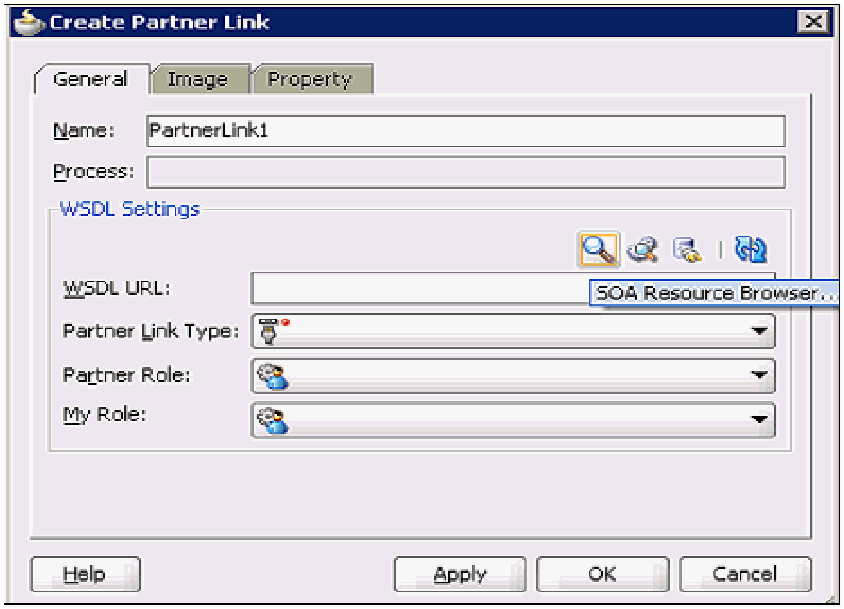
1. Double-click the outbound BPEL process component in the Components pane.
2. Right-click on the **Partner Links** pane and select **Create Partner Link**, as shown in Figure 4–130.

Figure 4–130 Create Partner Link



3. In the displayed Create Partner Link window, provide an appropriate name and click on the SOA Resource Browser tool, as shown in Figure 4–131.

Figure 4–131 SOA Resource Browser Tool



Name: PartnerLink1

Process:

WSDL URL:

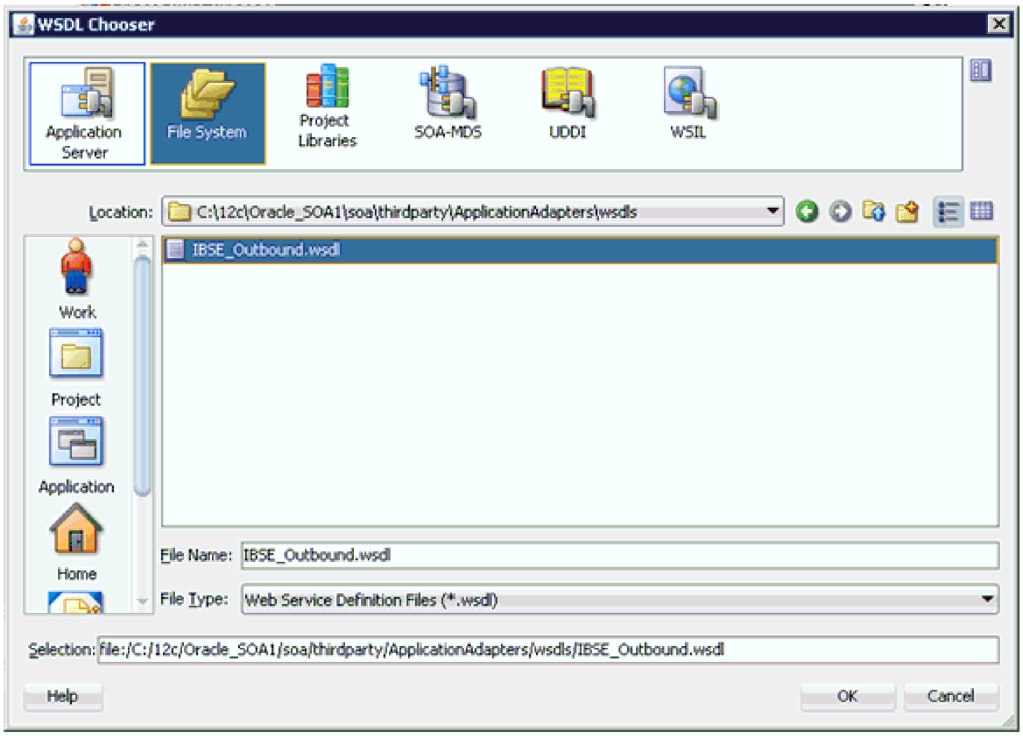
Partner Link Type:

Partner Role:

My Role:

4. In the WSDL Chooser dialog that is displayed, navigate to the location where the WSDL is exported from Application Explorer, select the WSDL, and click **OK**, as shown in Figure 4–132.

Figure 4–132 WSDL Chooser Dialog



Location: C:\12c\Oracle_SOA1\soa\thirdparty\ApplicationAdapters\wsdls

IBSE_Outbound.wsdl

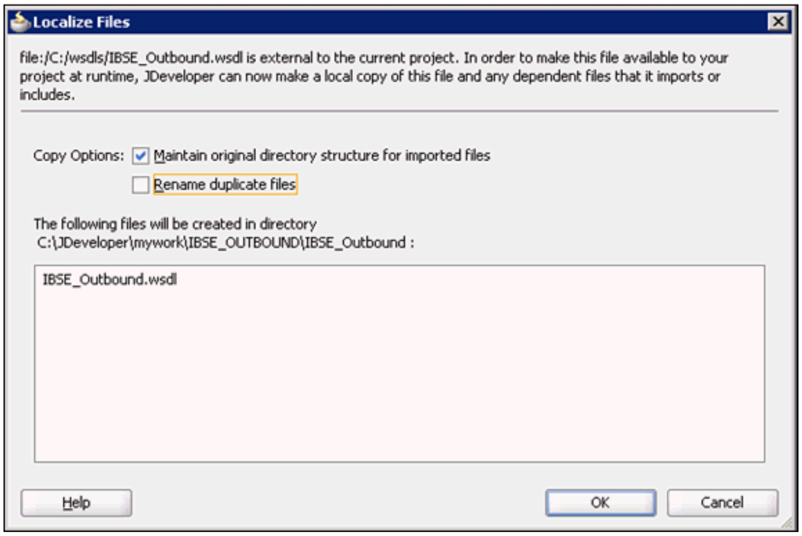
File Name: IBSE_Outbound.wsdl

File Type: Web Service Definition Files (*.wsdl)

Selection: file:/C:/12c/Oracle_SOA1/soa/thirdparty/ApplicationAdapters/wsdls/IBSE_Outbound.wsdl

5. In the displayed Localize Files window, uncheck the **Rename duplicate files** check box and click **OK**, as shown inFigure 4–133.

Figure 4–133 Localize Files Window



file:/C:/wsdls/IBSE_Outbound.wsdl is external to the current project. In order to make this file available to your project at runtime, JDeveloper can now make a local copy of this file and any dependent files that it imports or includes.

Copy Options: ☑ Maintain original directory structure for imported files

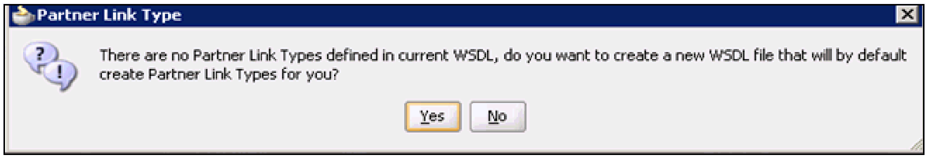
□ Rename duplicate files

The following files will be created in directory C:\JDeveloper\mywork\IBSE_OUTBOUND\IBSE_Outbound :

IBSE_Outbound.wsdl

6. Click **Yes** in the displayed Partner Link Type window, as shown in Figure 4–134.

Figure 4–134 Partner Link Type



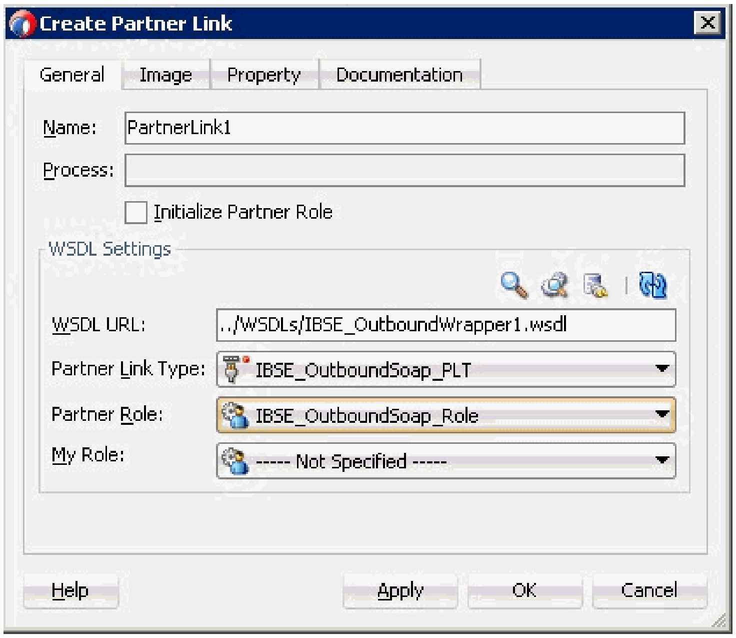
Partner Link Type

There are no Partner Link Types defined in current WSDL, do you want to create a new WSDL file that will by default create Partner Link Types for you?

7. In the displayed Create Partner Link window, expand the **Partner Role** drop-down list and select the available partner role.

8. Click **Apply**, and then **OK**, as shown in Figure 4–135.

Figure 4–135 Create Partner Link



Name: PartnerLink1

Process:

□ Initialize Partner Role

WSDL Settings

WSDL URL: ../WSDLs/IBSE_OutboundWrapper1.wsdl

Partner Link Type: IBSE_OutboundSoap_PLT

Partner Role: IBSE_OutboundSoap_Role

My Role: ----- Not Specified -----

4.6.3.2 Creating BPEL Activities and Mappings With the Created Partner Link

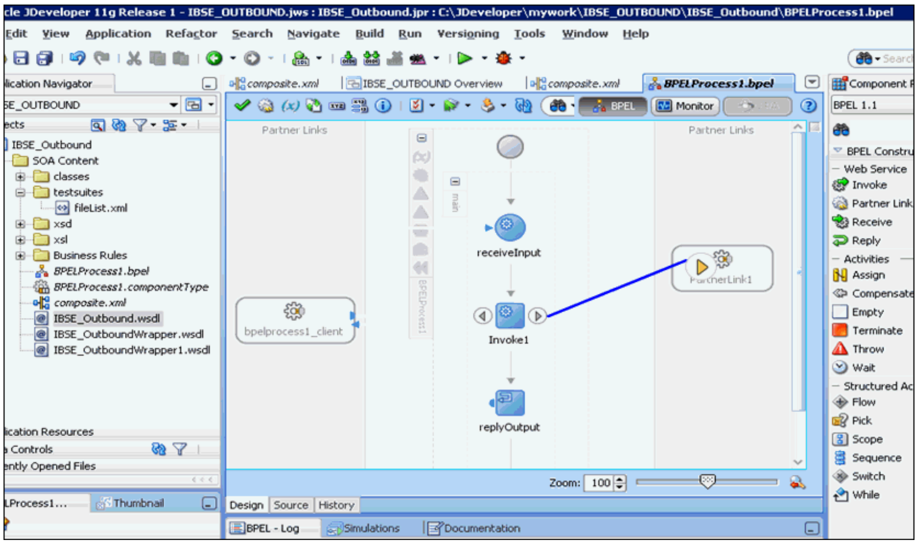
This section describes how to create BPEL activities and mappings with the created partner link.

To create BPEL Activities and map with the created partner link:

1. Drag and drop the **Invoke** activity component from BPEL Constructs to the Components pane. Place it between the **receiveInput** activity component and the **replyOutput** activity component.

2. Create a connection between the new **Invoke** activity component (Invoke1) and the **Partner Link** component (Partner link1), as shown in Figure 4–136.

Figure 4–136 Partner Link Component

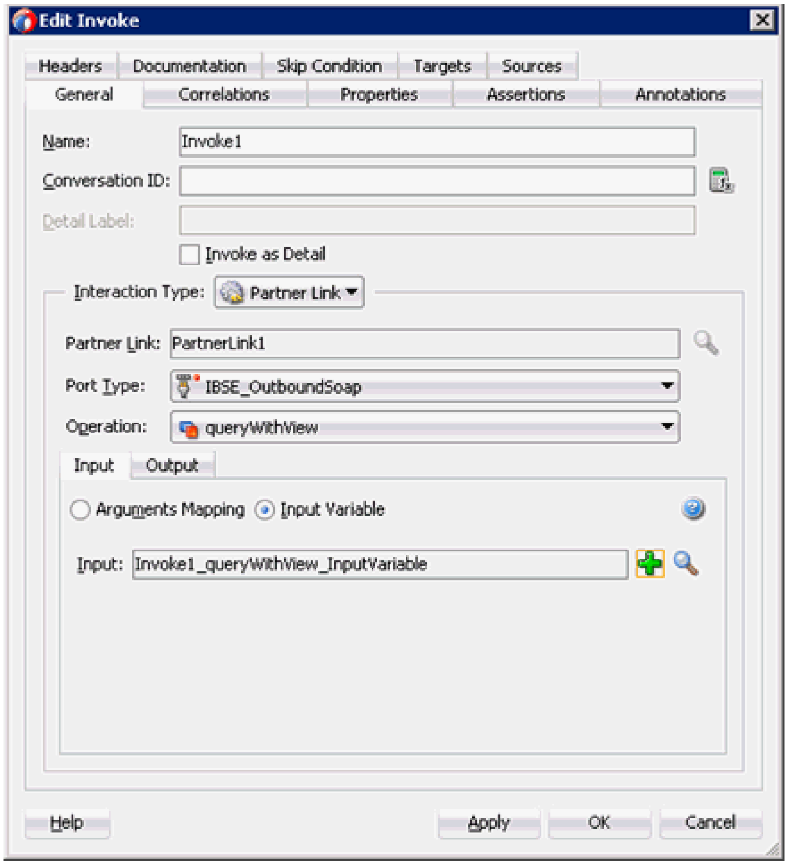


3. In the displayed Edit Invoke window, click the Plus (+) icon, located to the right of the Input field, to configure a new input variable.

4. Accept the default values that are provided for the new input variable and click **OK**.

5. Click the Plus (+) icon, which is located to the right of the Output field, to configure a new output variable, as shown in Figure 4–137.

Figure 4–137 Edit Invoke Window



Name: Invoke1

Conversation ID:

Detail Label:

□ Invoke as Detail

Interaction Type: Partner Link

Partner Link: PartnerLink1

Port Type: IBSE_OutboundSoap

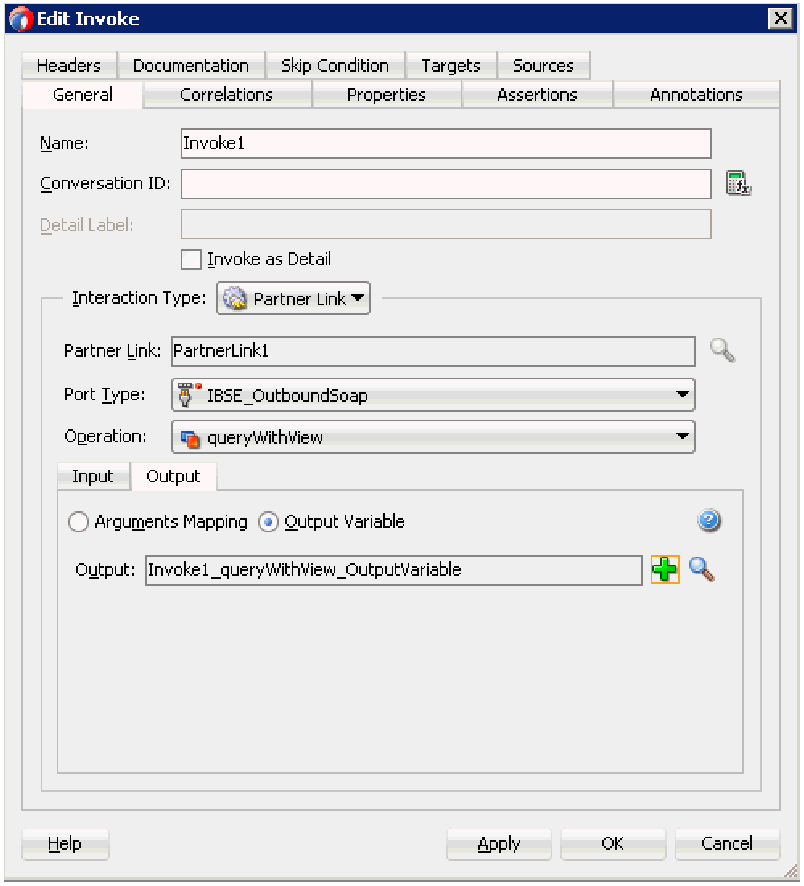
Operation: queryWithView

○ Arguments Mapping ● Input Variable

Input: Invoke1_queryWithView_InputVariable

6. Accept the default values that are provided for the new output variable and click **OK**.
7. Click **Apply** and then **OK**, as shown in Figure 4–138.

Figure 4–138 Edit Invoke Window



Edit Invoke

Headers | Documentation | Skip Condition | Targets | Sources
General | Correlations | Properties | Assertions | Annotations

Name: Invoke1

Conversation ID:

Detail Label:

□ Invoke as Detail

Interaction Type: Partner Link

Partner Link: PartnerLink1

Port Type: IBSE_OutboundSoap

Operation: queryWithView

Input | Output

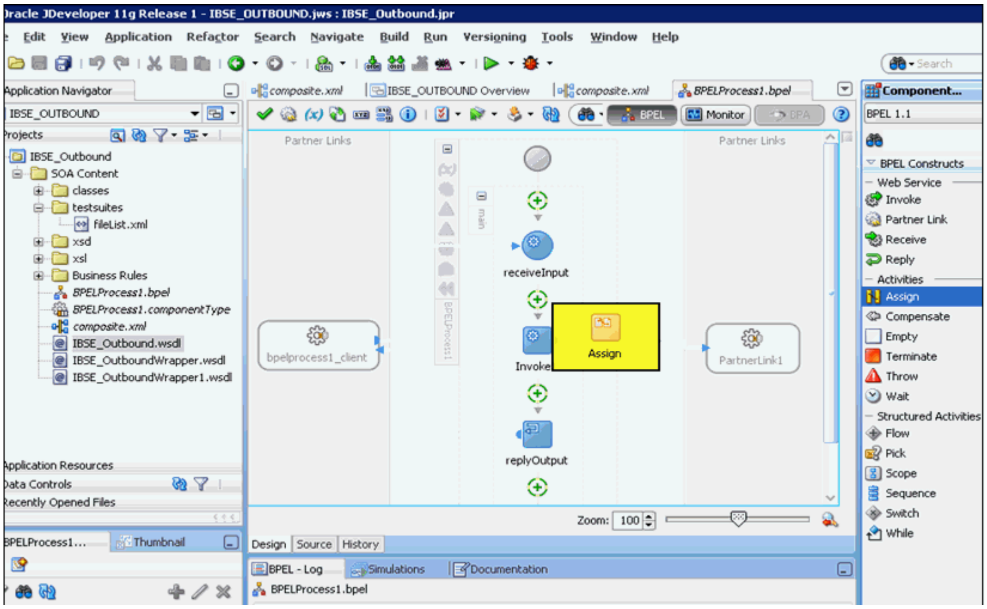
○ Arguments Mapping ● Output Variable

Output: Invoke1_queryWithView_OutputVariable

Help | Apply | OK | Cancel

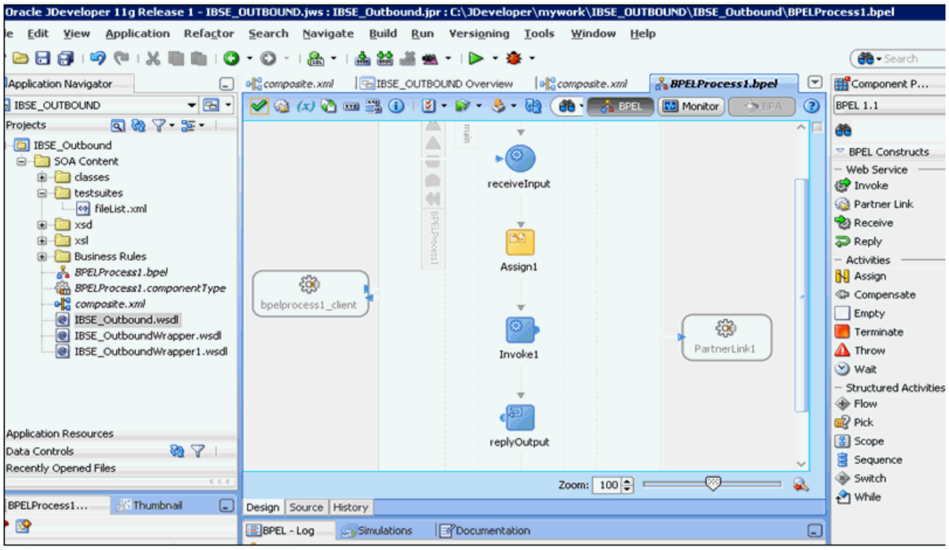
8. Drag and drop the **Assign** activity component from BPEL Constructs to the Components pane. Place it between the **Receive** activity component (receiveInput) and the **Invoke** activity component (Invoke1), as shown in Figure 4–139.

Figure 4–139 Assign Activity Component



9. Double-click the new **Assign** activity component (Assign1), as shown in Figure 4–140.

Figure 4–140 Assign Activity Component

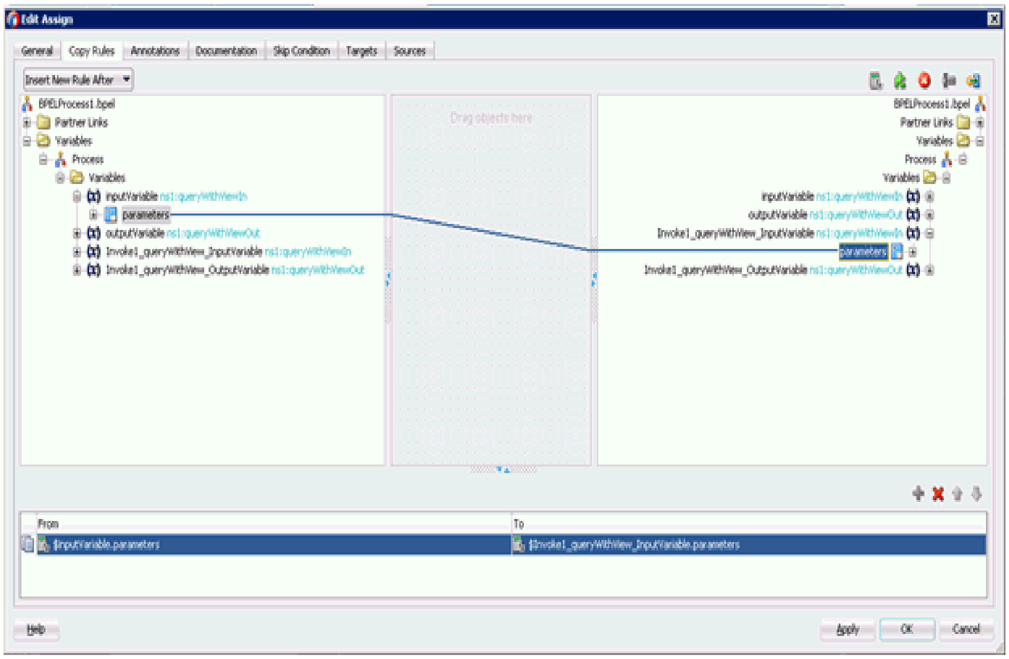


10. In the left pane, under Variables, expand **InputVariable**, and then select **parameters**.

11. In the right pane, under Variables, expand **Invoke1_queryWithView_InputVariable**, and then select **parameters**.

12. Drag and map the **InputVariable** parameters to the **Invoke1_queryWithView_InputVariable** parameters, as shown in Figure 4–141.

Figure 4–141 InputVariable Parameters

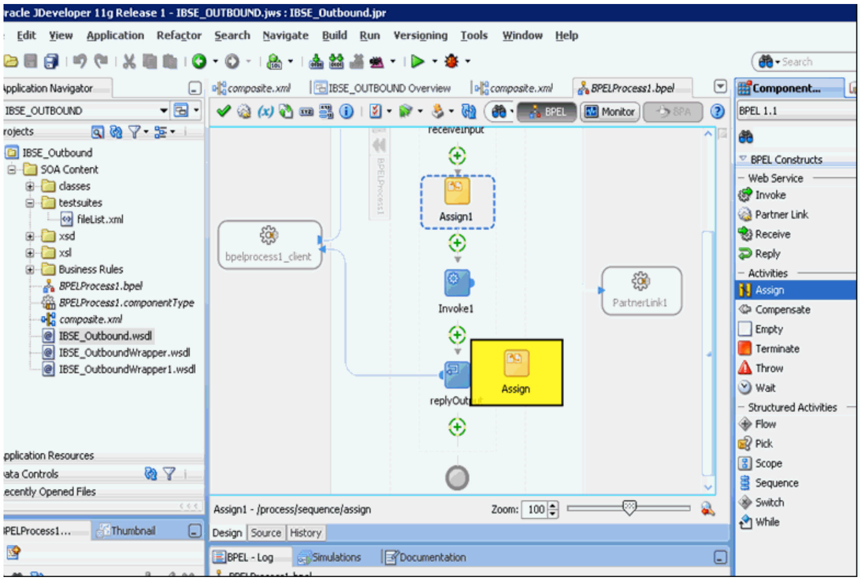

Edit Assign
General
Copy Rules
Annotations
Documentation
Skip Condition
Targets
Sources
Insert New Rule After
BPELProcess1.bpel
Partner Links
Variables
Process
Variables
inputVariable ns1:queryWithViewIn
parameters
outputVariable ns1:queryWithViewOut
Invoke1_queryWithView_InputVariable ns1:queryWithViewIn
Invoke1_queryWithView_OutputVariable ns1:queryWithViewOut
Drag objects here
BPELProcess1.bpel
Partner Links
Variables
Process
Variables
inputVariable ns1:queryWithViewIn
outputVariable ns1:queryWithViewOut
Invoke1_queryWithView_InputVariable ns1:queryWithViewIn
parameters
Invoke1_queryWithView_OutputVariable ns1:queryWithViewOut
Help
Apply
OK
Cancel

From	To
$inputVariable.parameters	$Invoke1_queryWithView_InputVariable.parameters

13. Click **Apply** and then **OK**.

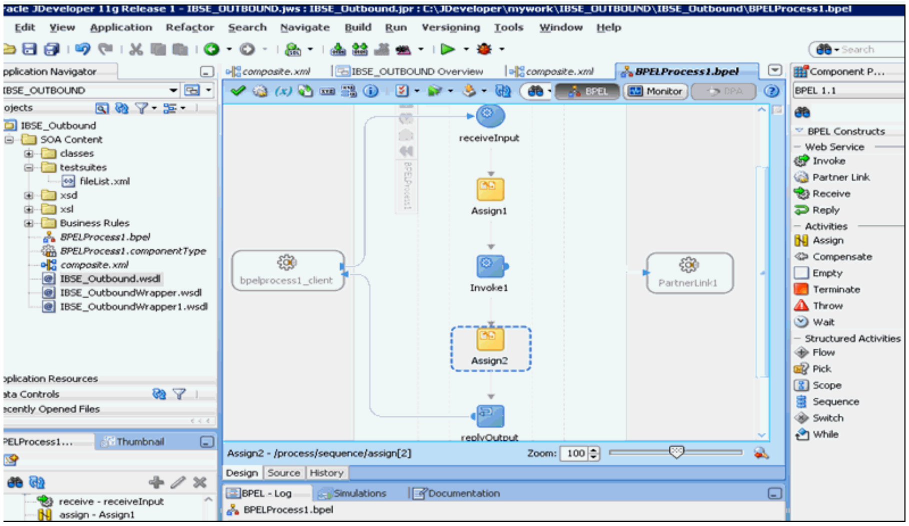
14. Drag and drop the **Assign** activity component to the Components pane and place it between the **Invoke** activity (Invoke1) and the **Reply** activity (replyOutput), as shown in Figure 4–142.

Figure 4–142 Assign Activity Component



15. Double-click the new **Assign** activity component (Assign2), as shown in Figure 4–143.

Figure 4–143 New Assign Activity Component

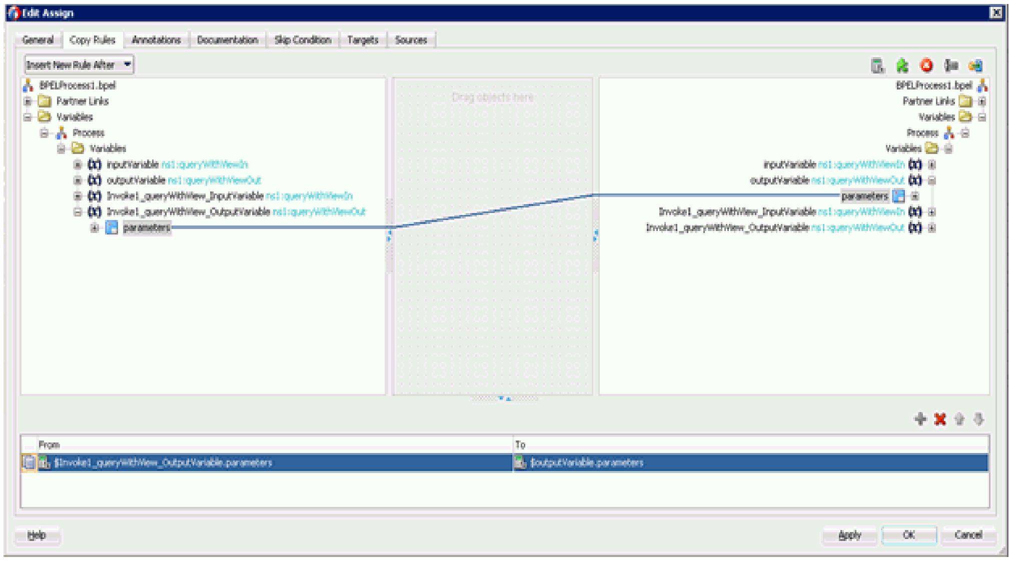


16. In the left pane, under Variables, expand **Invoke1_queryWithView_OutputVariable**, and then select **parameters**.

17. In the right pane, under Variables, expand **outputVariable**, and then select **parameters**.

18. Drag and map the **Invoke1_queryWithView_OutputVariable** parameters to the **outputVariable** parameters, as shown in Figure 4–144.

Figure 4–144 outputVariable Parameters

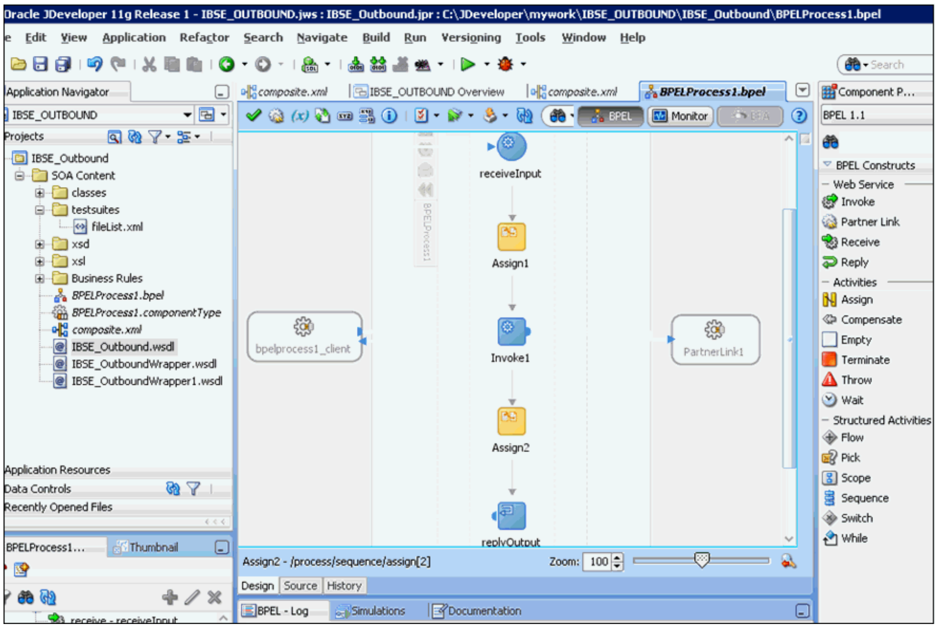


From	To
$Invoke1_queryWithView_OutputVariable.parameters	$outputVariable.parameters

19. Click **Apply** and then **OK**.

    You are returned to the component pane, as shown in Figure 4–145.

Figure 4–145 Component Pane



20. Click the **Save All** icon in the menu bar to save the new outbound BPEL process component that was configured.

    You are now ready to deploy the BPEL Outbound process. You can follow the same procedure as in Section 4.4.4, "Deploying the BPEL Outbound Process" on page 4-28.

    Once deployed you can invoke the input XML, as defined in Section 4.4.5, "Invoking the Input XML Document in the Oracle Enterprise Manager Console" on page 4-31.

5

Integration With Mediator Service Components in the Oracle SOA Suite

This chapter describes integration with Mediator service components in the Oracle SOA Suite. It contains the following sections:

- Section 5.1, "Configuring a New Application Server Connection"
- Section 5.2, "Configuring a Mediator Outbound Process (J2CA Configuration)"
- Section 5.3, "Configuring a Mediator Inbound Process (J2CA Configuration)"
- Section 5.4, "Configuring a Mediator Outbound Process (BSE Configuration)"

The scenarios shown in this chapter require the following prerequisites.

Prerequisites

The following are installation and configuration requirements:

- Oracle Application Adapter for Siebel must be installed on Oracle WebLogic Server.
- Siebel must be configured for inbound and outbound processing.
- OracleAS Technology adapters must be deployed and properly configured.

The examples in this chapter present the configuration steps necessary for demonstrating service and event integration with Siebel. Prior to using this material, you must be familiar with the following:

- How to configure Oracle Application Adapter for Siebel for services and events. For more information, see Chapter 2, "Configuring Oracle Application Server Adapter for Siebel".
- How to configure Oracle JDeveloper. For more information, see Chapter 4, "Integration With BPEL Service Components in the Oracle SOA Suite".
- How to use Siebel workflows. For more information on Siebel design requirements, see Appendix A, "Using Siebel Workflows".

Overview of Mediator Integration

Mediator provides a comprehensive application integration framework. Oracle Application Adapter for Siebel used with Mediator enables you to seamlessly integrate enterprise software, eliminating the need to write custom code. Functional modeling, as opposed to custom coding solutions, allows for software reuse and reduces the complexity and management challenges that arise over the software lifecycle. This integration model consists of two components--high-level integration logic and low-level platform services.

Adapter integration with Oracle WebLogic Server, Mediator is a two-step process:

1. **Design Time:** Oracle Application Adapter for Siebel is configured in Application Explorer for services and events, as described in Chapter 2, "Configuring Oracle Application Server Adapter for Siebel". Integration logic is modeled in iStudio. Metadata are stored in repositories.
2. **Runtime:** The underlying platform treats this metadata as run-time instructions to enable the communication between participating applications.

5.1 Configuring a New Application Server Connection

For more information on how to configure a new Application Server connection in Oracle JDeveloper, see Section 4.3, "Configuring a New Application Server Connection" on page 4-2.

5.2 Configuring a Mediator Outbound Process (J2CA Configuration)

This section describes how to configure a Mediator outbound process to your Siebel system, using a Mediator project in Oracle JDeveloper.

A sample project has been provided for this outbound use case scenario in the following folder of the Application Adapters installation:

```
<ADAPTER_HOME>\etc\sample\SIEBEL_Samples.zip\SIEBEL_
Samples\Mediator\J2CA\Outbound_Project
```

This section contains the following topics:

- Section 5.2.1, "Creating an Empty Composite for SOA"
- Section 5.2.2, "Defining a Mediator Outbound Process"
- Section 5.2.3, "Deploying the Mediator Outbound Process"
- Section 5.2.4, "Invoking the Input XML Document in the Oracle Enterprise Manager Console"

Prerequisites

Before you design a Mediator outbound process, you must generate the respective WSDL file using Application Explorer. For more information, see Section 4.4.1, "Generating WSDL for Request/Response Service" on page 4-8.

5.2.1 Creating an Empty Composite for SOA

Perform the following steps to create an empty composite for SOA:

1. Create a new SOA application.
2. Enter a name for the new SOA Application and click **Next**.

   The Name your project page is displayed.
3. Enter a project name and click **Next**.

   The Configure SOA settings page is displayed.
4. From the Composite Template list, select **Empty Composite** and click **Finish**.

For more information, see Section 4.4.2, "Creating an Empty Composite for SOA" on page 4-9.

5.2.2 Defining a Mediator Outbound Process

This section describes how to define a Mediator outbound process, which consists of the following topics:

- Section 5.2.2.1, "Configuring a Third Party Adapter Service Component"
- Section 5.2.2.2, "Configuring an Outbound Mediator Process Component"
- Section 5.2.2.3, "Configuring the Routing Rules"
- Section 5.2.2.4, "Adjusting for Known Deployment Issues With 12c"

5.2.2.1 Configuring a Third Party Adapter Service Component

Perform the following steps to create a third party adapter service component:

1. Drag and drop the **Third Party Adapter** component from the Service Adapters pane to the External References pane.
2. Enter a name for the third party adapter service.
3. Ensure that **Reference** is selected from the Type drop-down list (default).
4. Click the **Find existing WSDLs** icon, which is located to the right of the WSDL URL field.
5. Browse and select an outbound WSDL file from the following directory:

   ```
   <ADAPTER_HOME>\wsdls
   ```

6. Click **OK**.
7. Click **OK**.

   The outbound WSDL file and associated request and response XML schema files (.xsd) are imported to the project folder that has been created.

8. Click the **Find JCA file** icon, which is located to the right of the JCA File field.
9. Browse and select the JCA properties file from the following directory:

   ```
   <ADAPTER_HOME>\wsdls
   ```

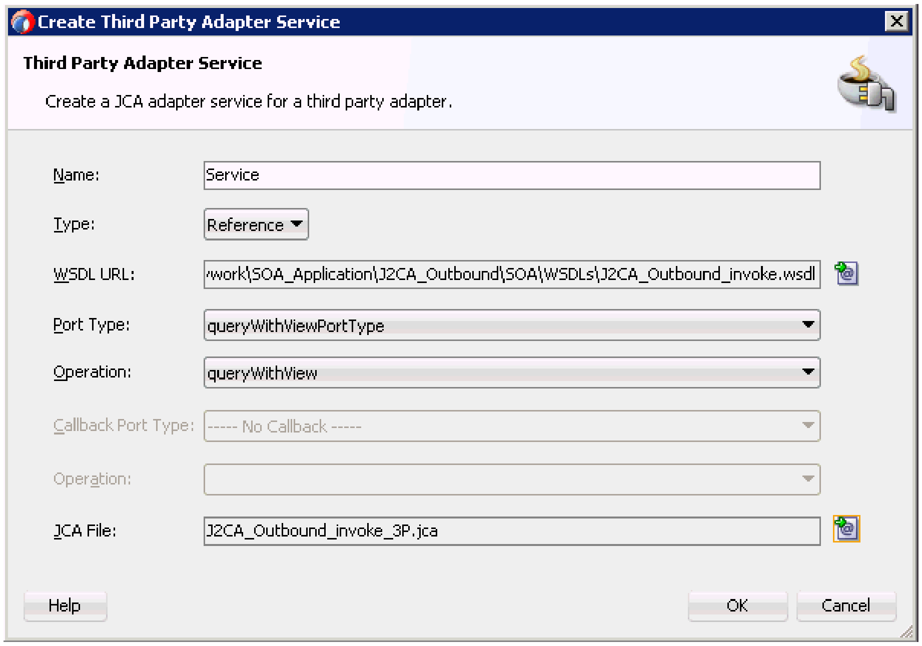
10. Click **OK**.

    A Copy File confirmation message is displayed.

11. Click **Yes**.

    A copy of the JCA properties file is made in the project folder.

Figure 5–1 Create Third Party Adapter Service Dialog



Create Third Party Adapter Service

Third Party Adapter Service

Create a JCA adapter service for a third party adapter.

Name: Service

Type: Reference

WSDL URL: /work\SOA_Application\J2CA_Outbound\SOA\WSDLs\J2CA_Outbound_invoke.wsdl

Port Type: queryWithViewPortType

Operation: queryWithView

Callback Port Type: ----- No Callback -----

Operation:

JCA File: J2CA_Outbound_invoke_3P.jca

12. Click **OK**.

    The third party adapter service component (GetDetail) is created in the External References pane.

    You are now ready to configure an outbound Mediator process component.

    For more information, see Section 6.4.3.1, "Configuring a Third Party Adapter Service Component" on page 6-11.

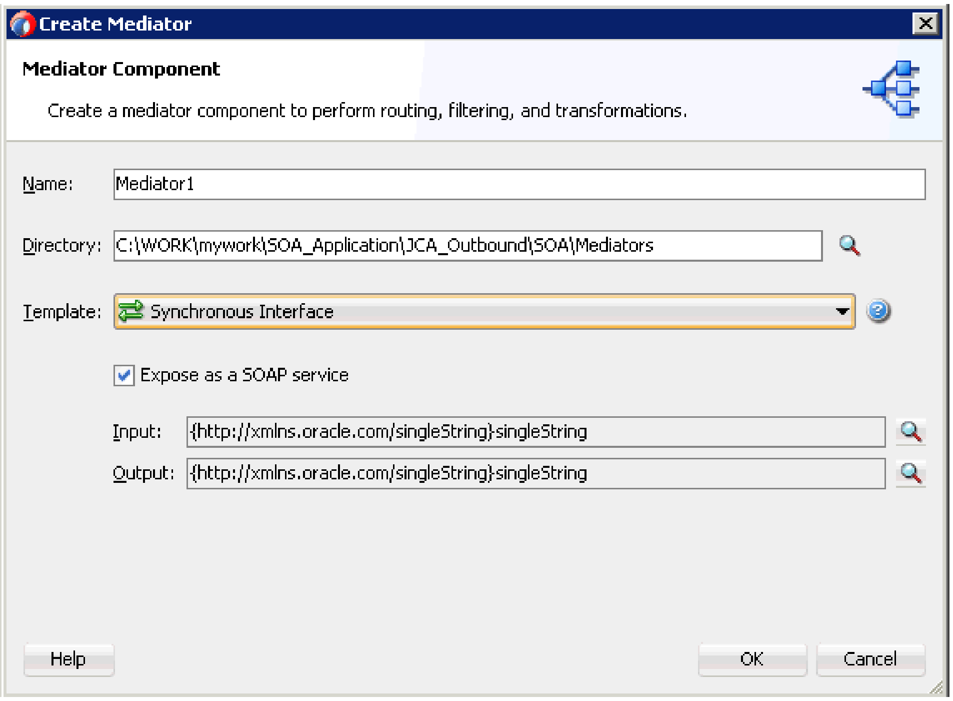
5.2.2.2 Configuring an Outbound Mediator Process Component

Perform the following steps to configure an outbound Mediator process component:

1. Drag and drop the **Mediator Process** component from the Components pane to the Components pane.

   The Create Mediator dialog is displayed, as shown in Figure 5–2.

Figure 5–2 Create Mediator Dialog



Mediator Component

Create a mediator component to perform routing, filtering, and transformations.

Name: Mediator1

Directory: C:\WORK\mywork\SOA_Application\JCA_Outbound\SOA\Mediators

Template: Synchronous Interface

☑ Expose as a SOAP service

Input: {http://xmlns.oracle.com/singleString}singleString

Output: {http://xmlns.oracle.com/singleString}singleString

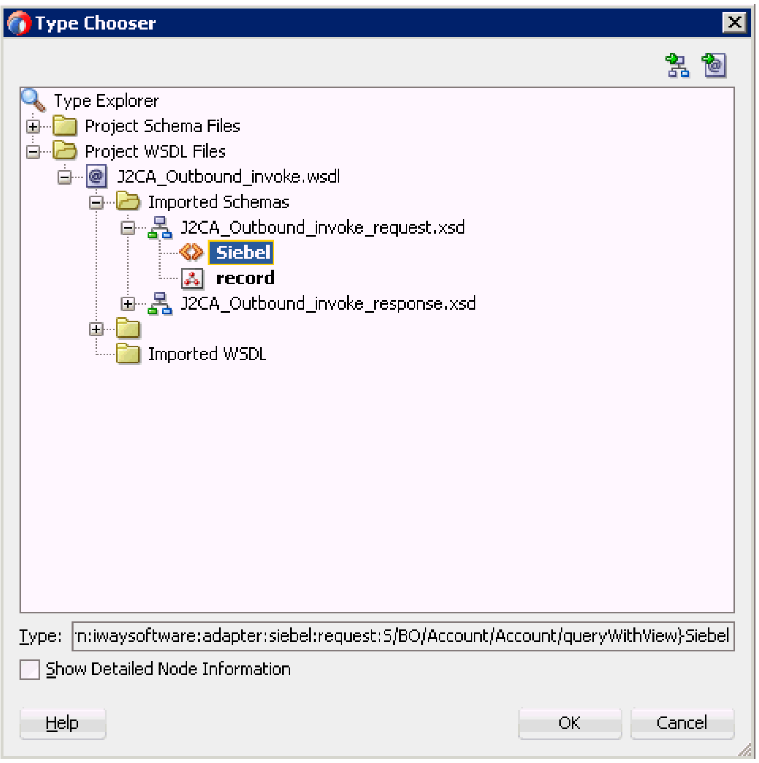
2. In the Name field, enter a name to identify the new outbound Mediator process component or leave it to the default value.

3. From the Template drop-down list, select **Synchronous Interface**.

4. Click the **Browse** icon, which is located to the right of the Input field to select the associated XML request schema file.

   The Type Chooser dialog is displayed, as shown in Figure 5–3.

Figure 5–3 Type Chooser Dialog



Type: 'n:iwaysoftware:adapter:siebel:request:S/BO/Account/Account/queryWithView}Siebel

□ Show Detailed Node Information

5. Expand **Project WSDL Files**, **J2CA_Outbound_invoke.wsdl**, **Imported Schemas**, **J2CA_Outbound_invoke_request.xsd**, and select **Siebel**.

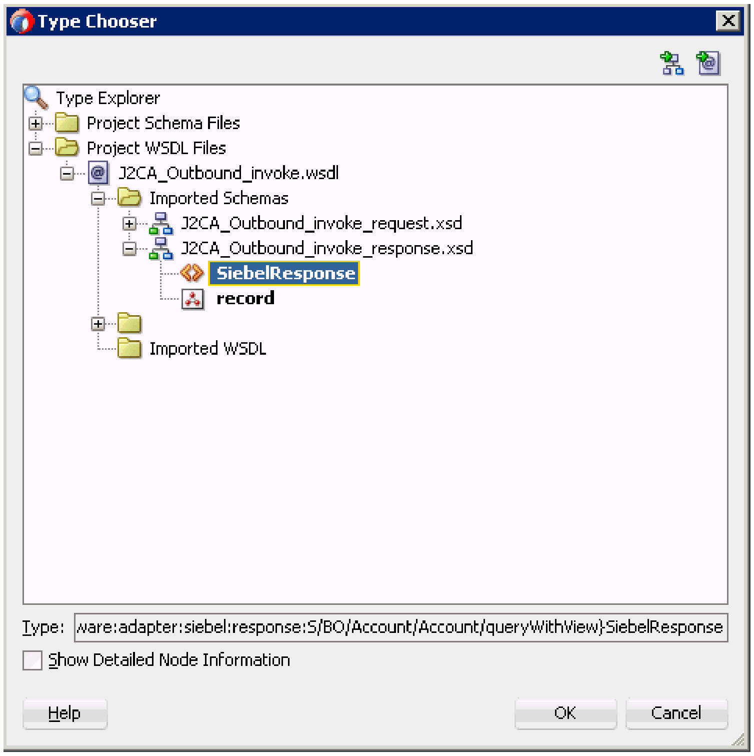
6. Click **OK**.

   You are returned to the Create Mediator dialog.

7. Click the **Browse** icon, which is located to the right of the Output field to select the associated XML response schema file.

   The Type Chooser dialog is displayed, as shown in Figure 5–4.

Figure 5–4 Type Chooser Dialog



Type: ware:adapter:siebel:response:S/BO/Account/Account/queryWithView}SiebelResponse

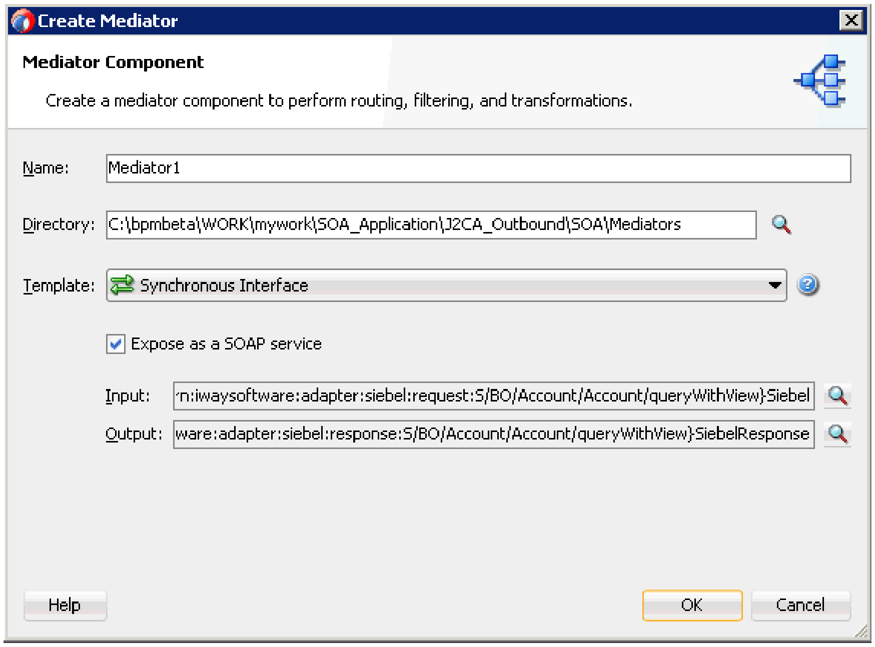
□ Show Detailed Node Information

8. Expand **Project WSDL Files**, **J2CA_Outbound_invoke.wsdl**, **Imported Schemas**, **J2CA_Outbound_invoke_response.xsd**, and select **SiebelResponse**.

9. Click **OK**.

   You are returned to the Create Mediator dialog, as shown in Figure 5–5.

Figure 5–5 Create Mediator Dialog



Mediator Component

Create a mediator component to perform routing, filtering, and transformations.

Name: Mediator1

Directory: C:\bpmbeta\WORK\mywork\SOA_Application\J2CA_Outbound\SOA\Mediators

Template: Synchronous Interface

☑ Expose as a SOAP service

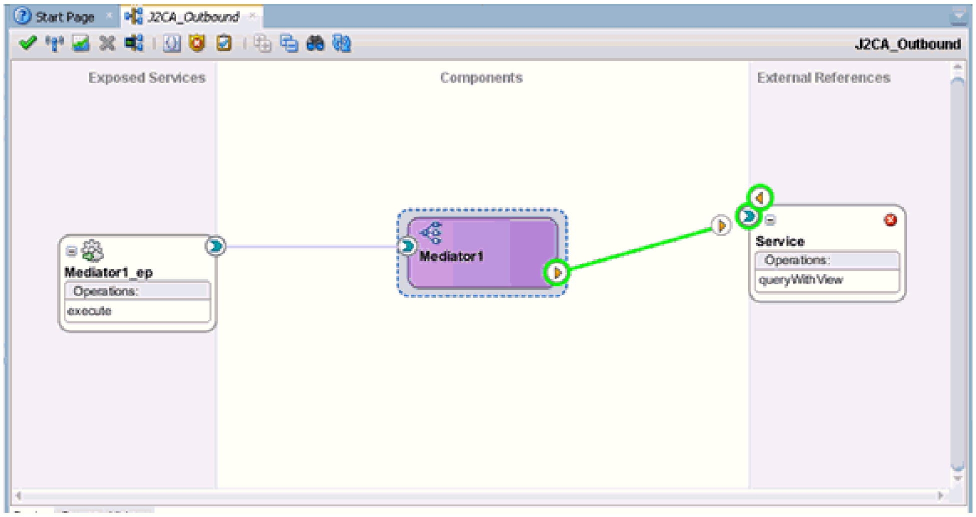
Input: rn:iwaysoftware:adapter:siebel:request:S/BO/Account/Account/queryWithView}Siebel

Output: ware:adapter:siebel:response:S/BO/Account/Account/queryWithView}SiebelResponse

10. Click **OK**.

11. Create a connection between the outbound Mediator process component and the third party adapter service component, as shown in Figure 5–6.

Figure 5–6 Created Connection



You are now ready to configure the routing rules.

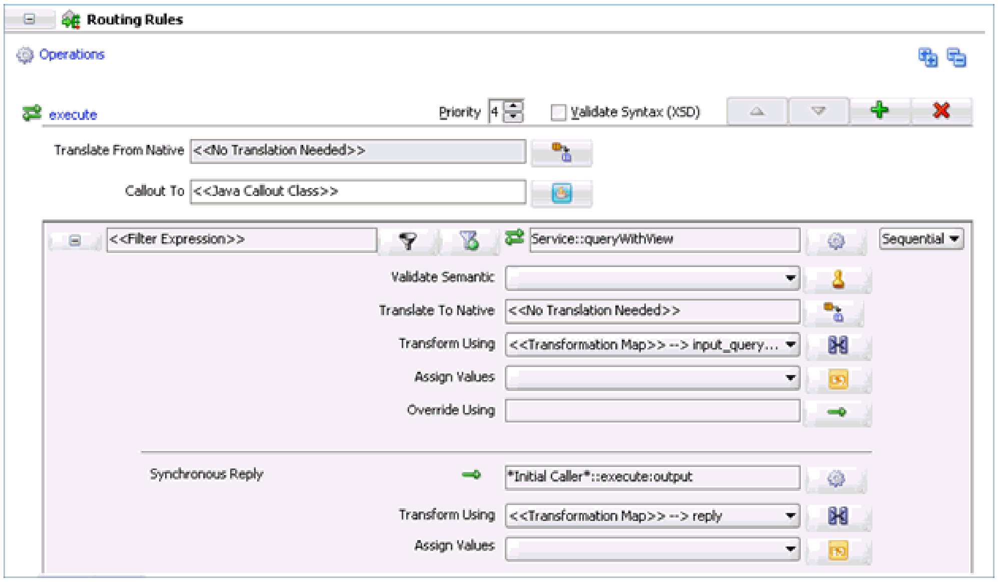
5.2.2.3 Configuring the Routing Rules

Perform the following steps to configure routing rules for the Mediator outbound process component:

1. Double-click the outbound Mediator process component in the Components pane.

   The Routing Rules dialog is displayed, as shown in Figure 5–7.

Figure 5–7 Routing Rules Dialog



Priority 4

□ Validate Syntax (XSD)

Translate From Native <<No Translation Needed>>

Callout To <<Java Callout Class>>

<<Filter Expression>>

Service::queryWithView

Sequential

Validate Semantic

Translate To Native <<No Translation Needed>>

Transform Using <<Transformation Map>> --> input_query...

Assign Values

Override Using

Synchronous Reply *Initial Caller*::execute:output

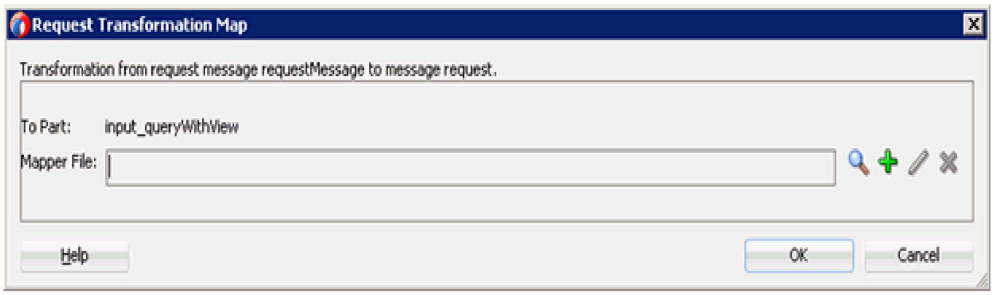
Transform Using <<Transformation Map>> --> reply

Assign Values

2. In the <<Filter Expression>> area, click the icon to the right of the Transform Using field.

   The Request Transformation Map dialog is displayed, as shown in Figure 5–8.

Figure 5–8 Request Transformation Map Dialog



Request Transformation Map

Transformation from request message requestMessage to message request.

To Part: input_queryWithView

Mapper File:

3. Click the Add (+) icon.

   The Create Transformation Map page is displayed.

4. Make sure the Type is selected as **XSLT** and click **OK**.

5. Click **OK**.

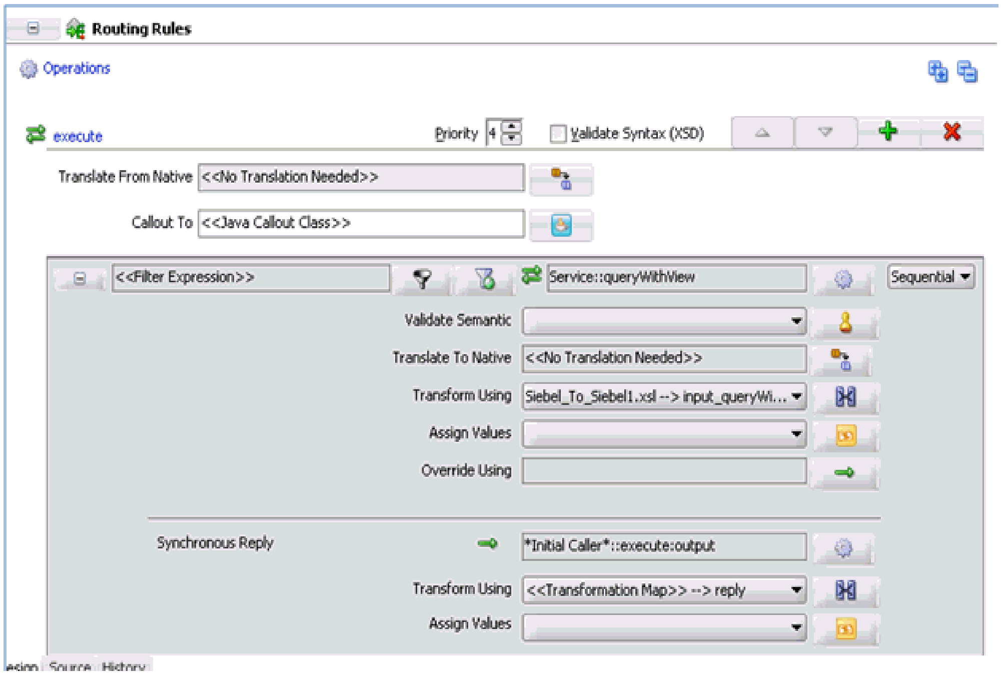
6. Map the **ns0:Siebel** source element to the **ns0:Siebel** target element.

   The Auto Map Preferences dialog is displayed.

7. Retain the default values and click **OK**.

8. Return to the Routing Rules dialog, as shown in Figure 5–9.

Figure 5–9 Routing Rules Dialog



Routing Rules

Operations

execute

Priority 4

□ Validate Syntax (XSD)

Translate From Native <<No Translation Needed>>

Callout To <<Java Callout Class>>

<<Filter Expression>>

Service::queryWithView

Sequential

Validate Semantic

Translate To Native <<No Translation Needed>>

Transform Using Siebel_To_Siebel1.xsl --> input_queryWi...

Assign Values

Override Using

Synchronous Reply

Initial Caller::execute:output

Transform Using <<Transformation Map>> --> reply

Assign Values

9. In the Synchronous Reply area, click the icon to the right of the Transform Using field.

The Reply Transformation Map dialog is displayed.

10. Click the Add (+) icon.

    The create Transformation Page is displayed.

11. Make sure the type is selected as **XSLT** and click **OK**.

    A mapping page is displayed.

12. Click **OK**.

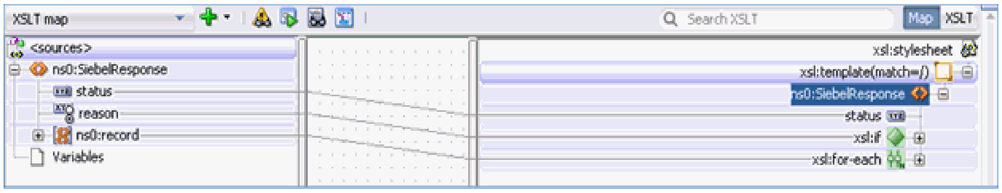
13. Map the **ns0:SiebelResponse** source element to the **ns0:SiebelResponse** target element.

    The Auto Map Preferences dialog is displayed.

14. Retain the default values and click **OK**.

    The mapping is completed, as shown in Figure 5–10.

Figure 5–10 Completed Mapping



15. Click the **Save All** icon in the menu bar to save the new outbound Mediator process component that was configured.

5.2.2.4 Adjusting for Known Deployment Issues With 12c

For more information on how to adjust for known deployment issues with 12c, see Section 4.4.3.3, "Adjusting for Known Deployment Issues With 12c" on page 4-26.

5.2.3 Deploying the Mediator Outbound Process

Perform the following steps to deploy the Mediator outbound process.

1. Right-click the project name in the left pane, select **Deploy**, and then click **J2CA_Outbound**.

    The Deployment Action page is displayed.

2. Ensure that **Deploy to Application Server** is selected.

3. Click **Next**.

    The Deploy Configuration page is displayed.

4. Leave the default values selected and click **Next**.

    The Select Server page is displayed.

5. Select an available application server that was configured and click **Next**.

    The SOA Servers page is displayed.

6. Select a target SOA server and click **Next**.

    The Summary page is displayed.

7. Review and verify all the available deployment information for your project and click **Finish**.

   For more information, see Section 4.4.4, "Deploying the BPEL Outbound Process" on page 4-28.

5.2.4 Invoking the Input XML Document in the Oracle Enterprise Manager Console

For more information, see Section 4.4.5, "Invoking the Input XML Document in the Oracle Enterprise Manager Console" on page 4-31.

5.3 Configuring a Mediator Inbound Process (J2CA Configuration)

This section describes how to configure a Mediator inbound process to your Siebel system, using a Mediator project in Oracle JDeveloper.

A sample project has been provided for this inbound use case scenario in the following folder of the Application Adapters installation:

```
<ADAPTER_HOME>\etc\sample\SIEBEL_Samples.zip\SIEBEL_Samples\Mediator\J2CA\Inbound_
Project
```

This section contains the following topics:

- Section 5.3.1, "Creating an Empty Composite for SOA"
- Section 5.3.2, "Defining a Mediator Inbound Process"

Prerequisites

Before you design a Mediator inbound process, you must generate the respective WSDL file using Application Explorer. For more information, see Section 4.5.1, "Generating WSDL for Event Integration" on page 4-34.

5.3.1 Creating an Empty Composite for SOA

Perform the following steps to create an empty composite for SOA:

1. Create a new SOA application.
2. Enter a name for the new SOA Application and click **Next**.

   The Name your project page is displayed.
3. Enter a project name and click **Next**.

   The Configure SOA settings page is displayed.
4. From the Composite Template list, select **Empty Composite** and click **Finish**.

   For more information, see Section 4.4.2, "Creating an Empty Composite for SOA" on page 4-9.

5.3.2 Defining a Mediator Inbound Process

This section describes how to define a Mediator inbound process, which contains the following topics:

- Section 5.3.2.1, "Configuring a Third Party Adapter Service Component"
- Section 5.3.2.2, "Configuring an Inbound Mediator Process Component With a File Adapter"

- Section 5.3.2.3, "Configuring the Routing Rules"
- Section 5.3.2.4, "Adjusting for Known Deployment Issues With 12c"

5.3.2.1 Configuring a Third Party Adapter Service Component

Perform the following steps to create a third party adapter service component:

1. Drag and drop the **Third Party Adapter** component from the Service Adapters pane to the Exposed Services pane.

   The Create Third Party Adapter Service dialog is displayed.

2. Enter a name for the third party adapter service.
3. Ensure that **Service** is selected from the Type drop-down list (default).
4. Click the **Find existing WSDLs** icon, which is located to the right of the WSDL URL field.

   The WSDL Chooser dialog is displayed.

5. Browse and select an inbound WSDL file from the following directory:

   ```
   <ADAPTER_HOME>\wsdls
   ```

6. Click **OK**.

   The Localize Files dialog is displayed.

7. Click **OK**.

   The inbound WSDL file and associated receive/request schema file (.xsd) are imported to the project folder that has been created.

   You are returned to the Create Third Party Adapter Service dialog.

8. Click the **Find JCA file** icon, which is located to the right of the JCA File field.

   The Transformation Chooser dialog is displayed.

9. Browse and select the JCA properties file from the following directory:

   ```
   <ADAPTER_HOME>\wsdls
   ```

10. Click **OK**.

    The Copy File Confirmation message is displayed.

11. Click **Yes**.

    A copy of the JCA properties file is made in the project folder.

    You are returned to the Create Third Party Adapter Service dialog.

12. Click **OK**.

    The third party adapter service component is created in the Exposed Services pane.

    You are now ready to configure an inbound Mediator process component.

    For more information, see Section 4.5.3.1, "Creating a Third Party Adapter Service Component" on page 4-42.

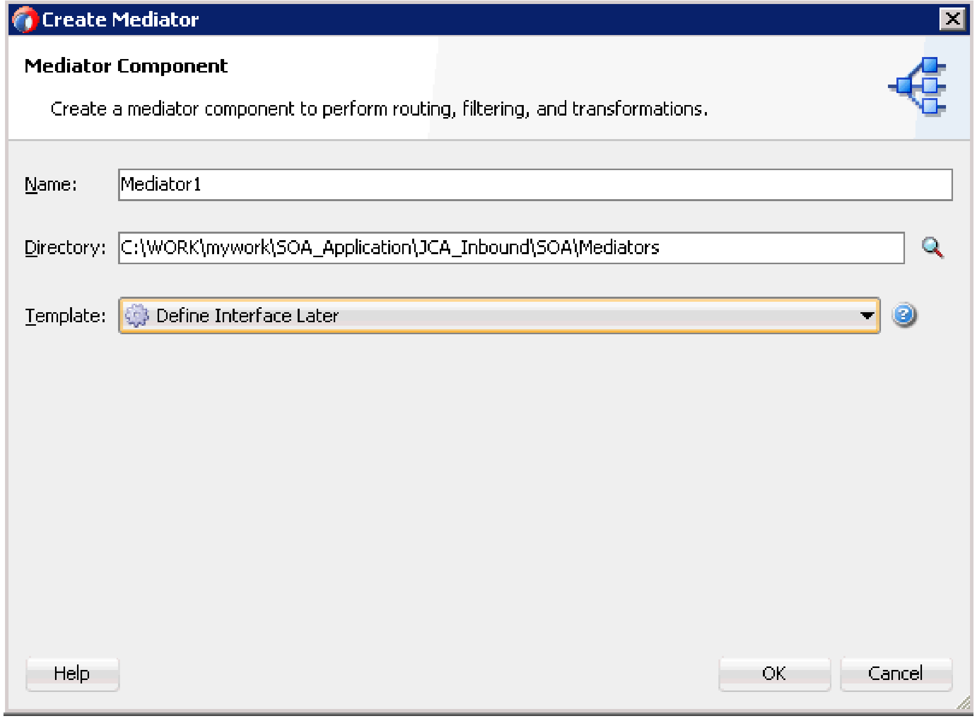
5.3.2.2 Configuring an Inbound Mediator Process Component With a File Adapter

Perform the following steps to configure an inbound Mediator process component with a File adapter.

1. Drag and drop the **Mediator Process** component from the Service Components pane to the Components pane.

   The Create Mediator dialog is displayed, as shown in Figure 5–11.

Figure 5–11 Create Mediator Dialog



Create Mediator

Mediator Component

Create a mediator component to perform routing, filtering, and transformations.

Name: Mediator1

Directory: C:\WORK\mywork\SOA_Application\JCA_Inbound\SOA\Mediators

Template: Define Interface Later

2. In the Name field, enter a name to identify the new inbound Mediator process component.
3. From the Template drop-down list, select **Define Interface Later**.
4. Click the **OK**.

   The new Mediator process component is added to the Components pane.

5. Drag and drop the **File** component from the Technology Adapters pane to the External References pane.

   The File Adapter Configuration Wizard is displayed.

6. Type a name for the new File adapter and click **Next**.

   The Adapter Interface page is displayed.

7. Ensure that the **Define from operation and schema (specified later)** option is selected.
8. Click **Next**.

   The Operation page is displayed.

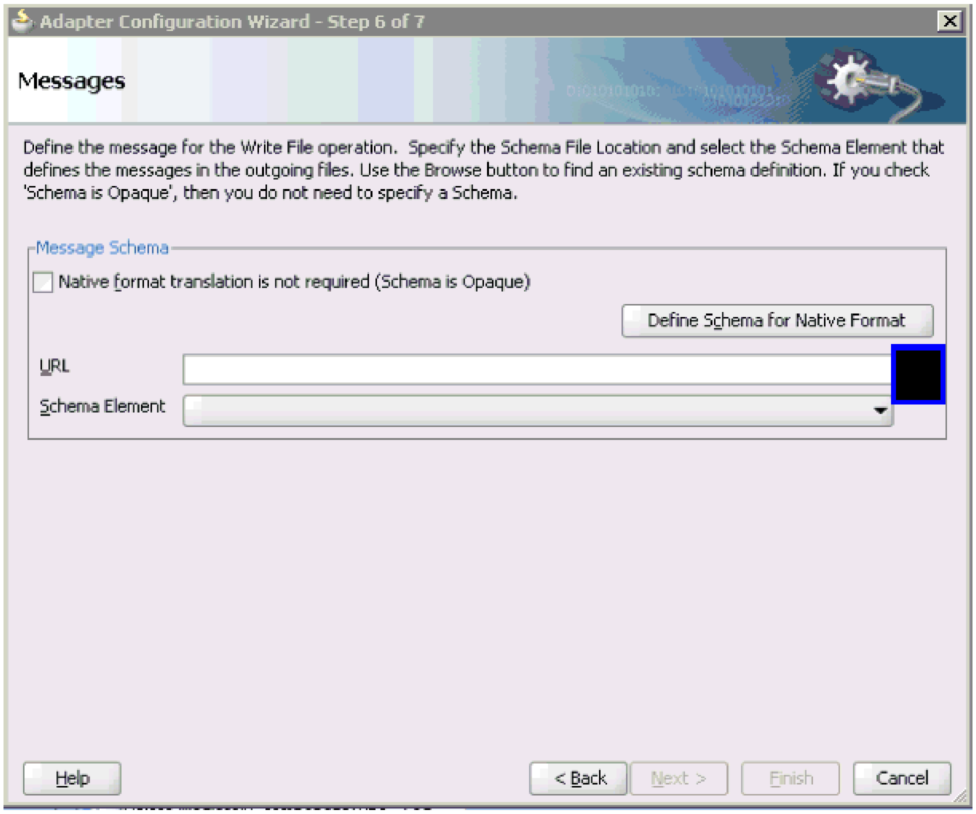
9. Click **Next**.
10. Select **Write File** from the list of Operation Type options and specify an Operation Name (for example, Write).
11. Click **Next**.

The File Configuration page is displayed.

12. Specify a location on your file system where the output file is written.
13. In the File Naming Convention field, specify a name for the output file.
14. Click **Next**.

    The Messages page is displayed, as shown in Figure 5–12.

Figure 5–12 Messages Page



Adapter Configuration Wizard - Step 6 of 7
Messages

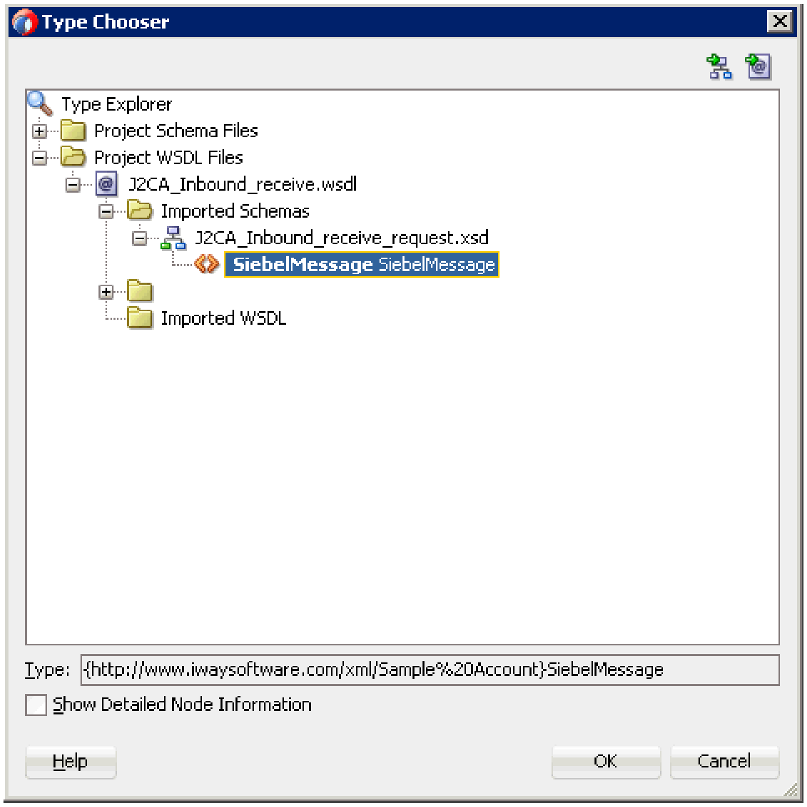
Define the message for the Write File operation. Specify the Schema File Location and select the Schema Element that defines the messages in the outgoing files. Use the Browse button to find an existing schema definition. If you check 'Schema is Opaque', then you do not need to specify a Schema.

Message Schema
□ Native format translation is not required (Schema is Opaque)
Define Schema for Native Format
URL
Schema Element
Help
< Back
Next >
Finish
Cancel

15. Click **Browse**, which is located to the right of the URL field.

    The Type Chooser dialog is displayed, as shown in Figure 5–13.

Figure 5–13 Type Chooser Dialog



Type: {http://www.iwaysoftware.com/xml/Sample%20Account}SiebelMessage

□ Show Detailed Node Information

16. Expand **Project WSDL Files**, **J2CA_Inbound_receive.wsdl**, **Imported Schemas**, **J2CA_Inbound_receive_request.xsd**, and select **SiebelMessage SiebelMessage**.

17. Click **OK**.

    You are returned to the Messages page.

18. Click **Next**.

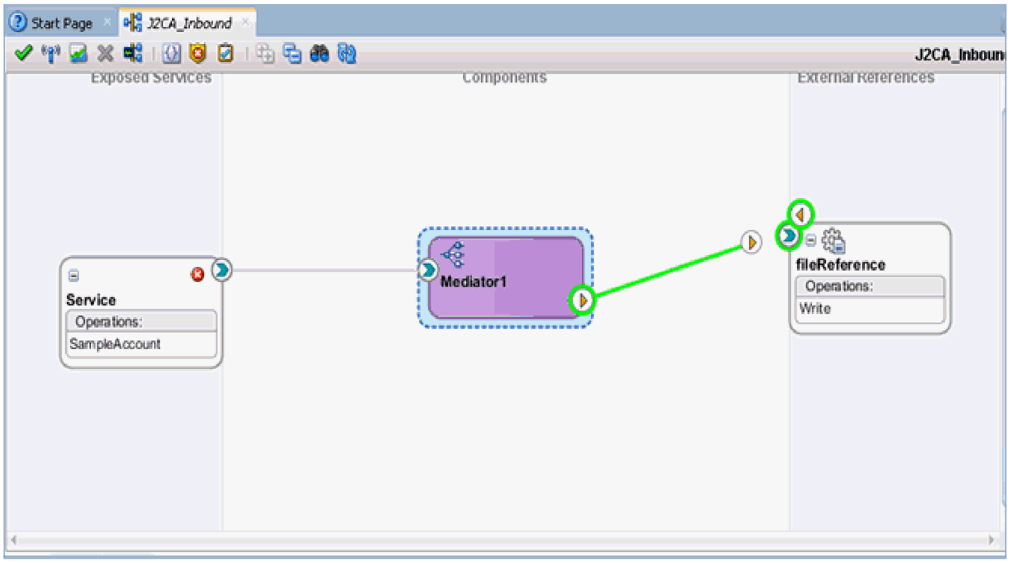
    The Finish page is displayed.

19. Click **Finish**.

20. Create a connection between the inbound Mediator process component and the third party adapter service component.

21. Create a connection between the inbound Mediator process component and the File adapter component, as shown in Figure 5–14.

Figure 5–14 Created Connection



You are now ready to configure the routing rules.

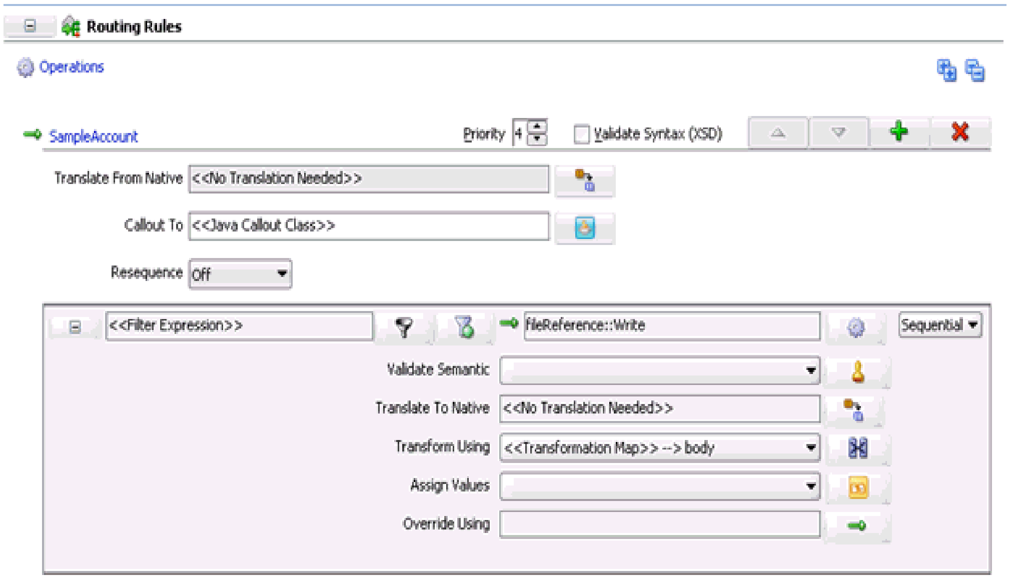
5.3.2.3 Configuring the Routing Rules

Perform the following steps to configure routing rules for the Mediator inbound process component:

1. Double-click the inbound Mediator process component in the Components page.

   The Routing Rules dialog is displayed, as shown in Figure 5–15.

Figure 5–15 Routing Rules Dialog



SampleAccount Priority 4 □ Validate Syntax (XSD)

Translate From Native <<No Translation Needed>>

Callout To <<Java Callout Class>>

Resequence Off

<<Filter Expression>> fileReference::Write Sequential

Validate Semantic

Translate To Native <<No Translation Needed>>

Transform Using <<Transformation Map>> --> body

Assign Values

Override Using

2. In the <<Filter Expression>> area, click the icon to the right of the Transform Using field.

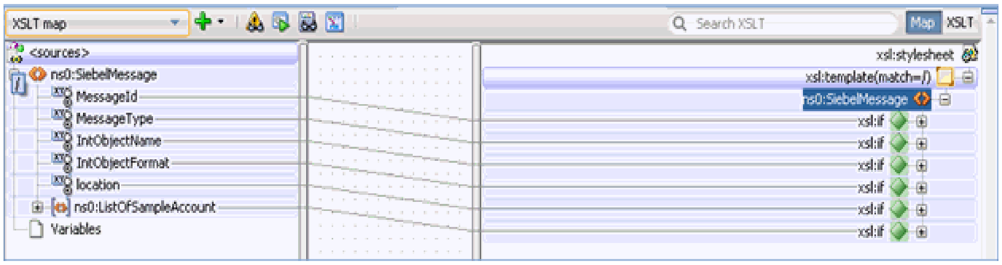
   The Request Transformation Map dialog is displayed.

3. Click the Add (+) icon and ensure that the selected Type is **XSLT**, then click **OK**.

4. Click **OK**.

   The mapping page is displayed, as shown in Figure 5–16.

Figure 5–16 Mapping Page



5. Click **OK**.
6. Map the **ns0:SiebelMessage** source element to the **ns0:SiebelMessage** target element.

   The Auto Map Preferences dialog is displayed.
7. Retain the default values and click **OK**.

   The mapping is now complete.
8. Click the **Save All** icon in the menu bar to save the new inbound Mediator process component that was configured.

5.3.2.4 Adjusting for Known Deployment Issues With 12c

For more information on how to adjust for known deployment issues with 12c, see Section 4.4.3.3, "Adjusting for Known Deployment Issues With 12c" on page 4-26.

You are now ready to deploy the Mediator inbound process. You can follow the same procedure in Section 4.5.4, "Deploying the BPEL Inbound Process" on page 4-48.

Once event messages are triggered through Siebel, output XML is received in the location that was specified for the File adapter component. For more information on triggering events in Siebel, see Section 4.5.5.3, "Triggering an Event in Siebel 8.0 to Test Event Runtime Integration" on page 4-64.

5.4 Configuring a Mediator Outbound Process (BSE Configuration)

This section describes how to configure a Mediator outbound process to your Siebel system, using a Mediator project in Oracle JDeveloper.

A sample project has been provided for this outbound use case scenario in the following folder of the Application Adapters installation:

```
<ADAPTER_HOME>\etc\sample\SIEBEL_Samples.zip\SIEBEL_Samples\Mediator\BSE\Outbound_
Project
```

This section contains the following topics:

- Section 5.4.1, "Creating an Empty Composite for SOA"
- Section 5.4.2, "Defining a Mediator Outbound Process"

  Prerequisites

  Before you design a Mediator outbound process, you must generate the respective WSDL file using Application Explorer. For more information, see Section 4.6.1, "Generating a WSDL File for Request and Response Services Using a Web Service" on page 4-75.

5.4.1 Creating an Empty Composite for SOA

Perform the following steps to create an empty composite for SOA:

1. Create a new SOA application.
2. Enter a name for the SOA Application (for example, IBSE_OUTBOUND), and click **Next**.
3. Enter a project name (for example, IBSE_Outbound), and click **Next**.
4. From the Composite Template list, select **Empty Composite** and click **Finish**.

For more information, see Section 4.4.2, "Creating an Empty Composite for SOA" on page 4-9.

5.4.2 Defining a Mediator Outbound Process

This section describes how to define a Mediator outbound process. The following topics are included:

- Section 5.4.2.1, "Configuring a SOAP Service"
- Section 5.4.2.2, "Creating a Mediator Component"
- Section 5.4.2.3, "Configuring the Routing Rules"

5.4.2.1 Configuring a SOAP Service

Perform the following steps to configure a SOAP Service:

1. Drag and drop the **SOAP** node from the Technology Adapters pane to the External References pane.
2. Enter an appropriate name for the SOAP Service and click on the **Find existing WSDLs** icon, which is located to the right of the WSDL URL field.
3. In the displayed SOA Resource Browser window, select the File system tab and navigate to the location where the WSDL is exported from the Application Explorer, select the WSDL, and click **OK**.
4. In the Create Web Service Window, click **OK**.
5. In the displayed Localize Files window, click **OK**. This imports the WSDL file to the project folder.

   The Web Service is created and displayed.

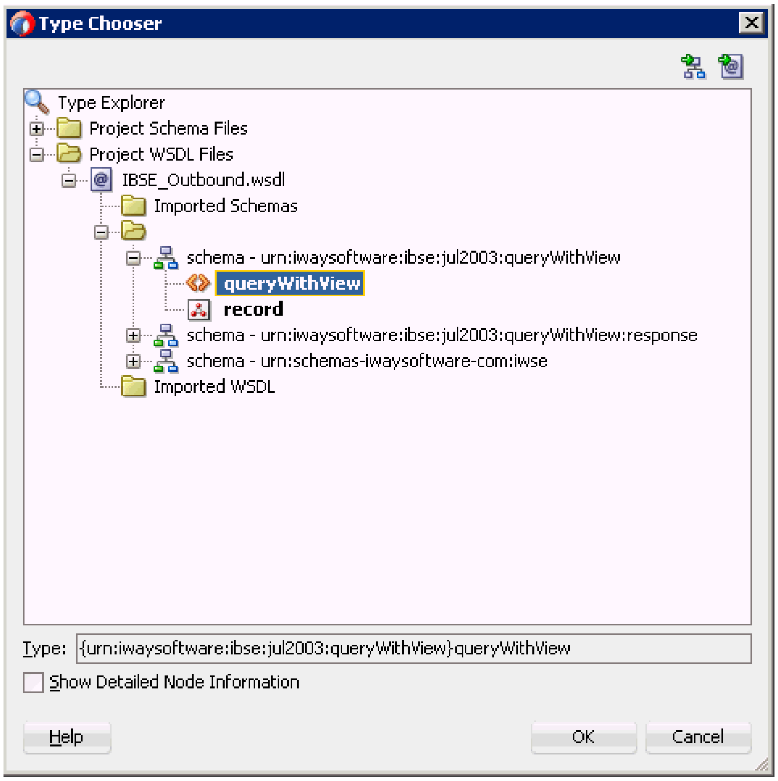
5.4.2.2 Creating a Mediator Component

Perform the following steps to create a Mediator component:

1. Drag and drop the **Mediator** component from the Components pane in to the Components pane.
2. In the Name field, enter a name to identify the new outbound Mediator process component.
3. From the Template drop-down list, select **Synchronous Interface**.
4. Click the **Browse** icon, which is located to the right of the Input field, to select the associated XML request schema file.

5. In the Type Chooser dialog, expand **Project WSDL Files**, select **IBSE_Outbound.wsdl**, and click **queryWithView**, as shown in Figure 5–17.

Figure 5–17 Type Chooser Dialog



Type Chooser
Type Explorer
Project Schema Files
Project WSDL Files
IBSE_Outbound.wsdl
Imported Schemas
schema - urn:iwaysoftware:ibse:jul2003:queryWithView
queryWithView
record
schema - urn:iwaysoftware:ibse:jul2003:queryWithView:response
schema - urn:schemas-iwaysoftware-com:iwse
Imported WSDL

Type: {urn:iwaysoftware:ibse:jul2003:queryWithView}queryWithView

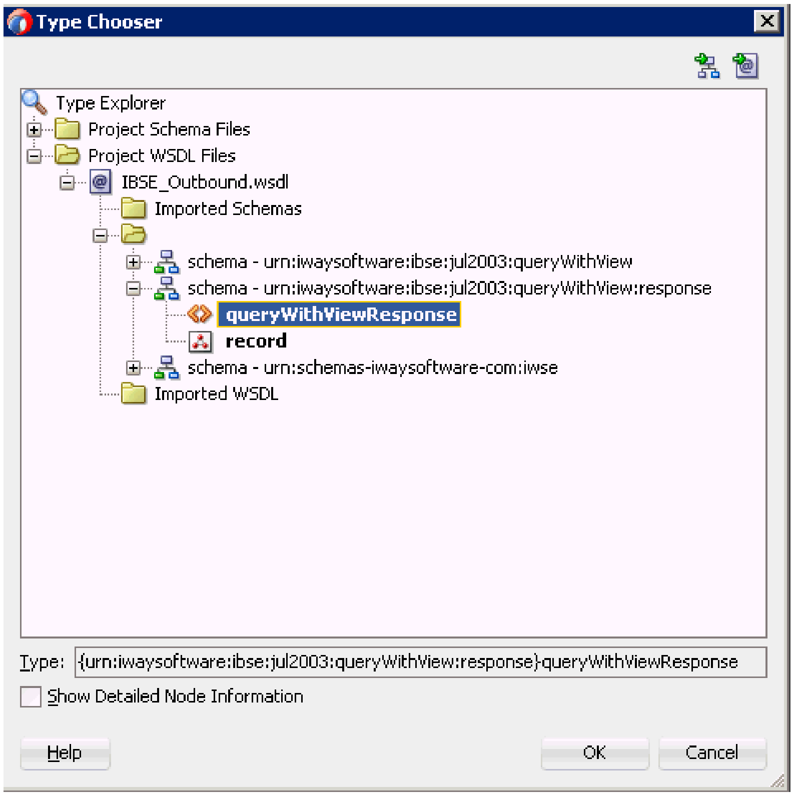
□ Show Detailed Node Information

6. Click **OK**.

7. Click the **Browse** icon, which is located to the right of the Output field, to select the associated XML response schema file.

8. In The Type Chooser dialog, expand **Project WSDL Files**, select **IBSE_Outbound.wsdl**, and click **queryWithViewResponse**, as shown in Figure 5–18.

Figure 5–18 Type Chooser Dialog



Type Chooser
Type Explorer
Project Schema Files
Project WSDL Files
IBSE_Outbound.wsdl
Imported Schemas
schema - urn:iwaysoftware:ibse:jul2003:queryWithView
schema - urn:iwaysoftware:ibse:jul2003:queryWithView:response
queryWithViewResponse
record
schema - urn:schemas-iwaysoftware-com:iwse
Imported WSDL
Show Detailed Node Information
Help
OK
Cancel

Type: {urn:iwaysoftware:ibse:jul2003:queryWithView:response}queryWithViewResponse

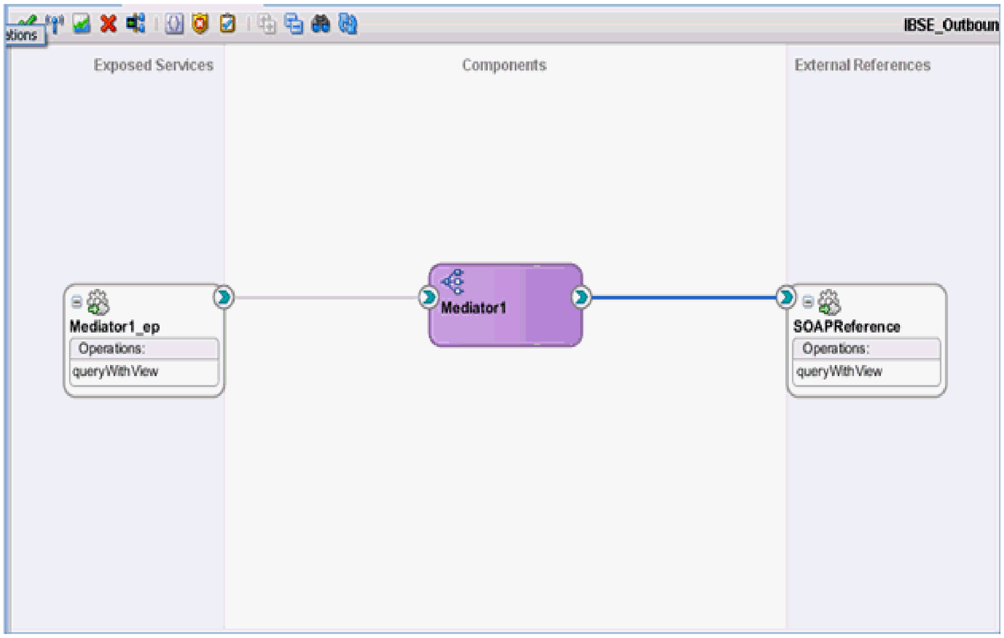
☐ Show Detailed Node Information

9. Click **OK**.
10. Click **OK**.

    The Mediator component is created and displayed.

11. Create a connection between the **Mediator** component and the **SOAP service** component, as shown in Figure 5–19.

Figure 5–19 Created Connection

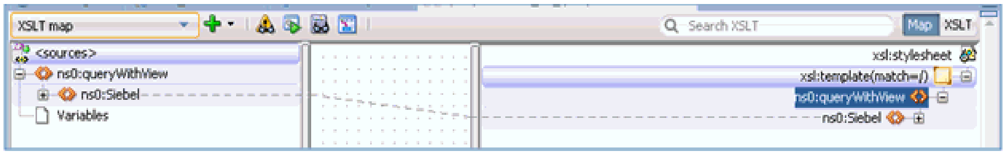


5.4.2.3 Configuring the Routing Rules

Perform the following steps to configure the routing rules:

1. Double-click the **Mediator** component in the Components pane.
2. In the <<Filter Expression>> area of the Static Routing section, click the icon to the right of the Transform Using field.
3. In the displayed Request Transformation Map window, click the Add (+) icon and make sure the selected Type is **XSLT** in the Create Transformation Map dialog box and click **OK**.
4. Click **OK**.
5. Map the **ns0:queryWithView** source element to the **ns0:queryWithView** target element, as shown in Figure 5–20.

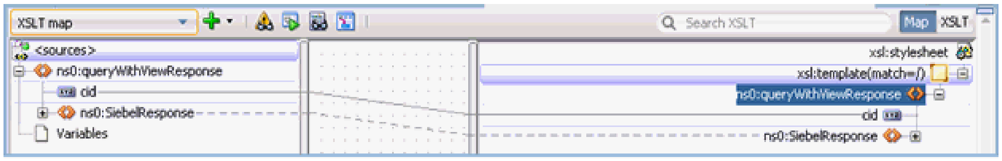
Figure 5–20 GetDetail_To_GetDetail.xsl Tab



6. In the displayed Auto Map Preferences window, retain the default values and click **OK**.
7. In the Synchronous Reply area, click the icon to the right of the Transform Using field.
8. In the displayed Reply Transformation Map window, click the Add (+) icon and make sure the Type is selected as **XSLT** in the Create Transformation Map dialog box, and then click **OK**.

9. Map the **ns0:queryWithViewResponse** source element to the **ns0:queryWithViewResponse** target element, as shown in Figure 5–21.

Figure 5–21 Source and Target Elements

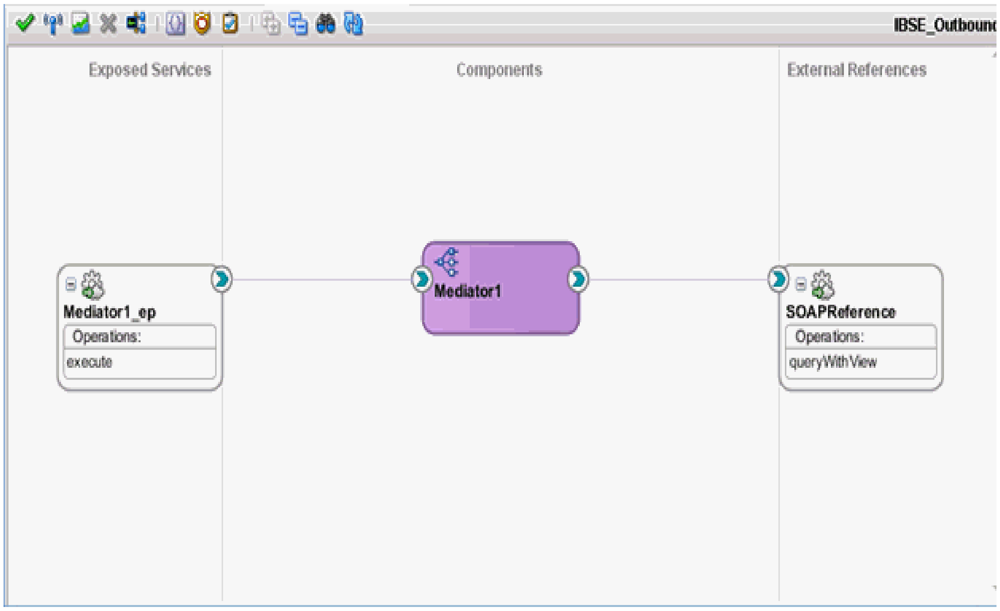


10. In the displayed Auto Map Preferences window, retain the default values and click **OK**.

11. Double-click **composite.xml** in the left pane.

12. Click the **Save All** icon in the menu bar to save the new outbound Mediator component that was configured, as shown in Figure 5–22.

Figure 5–22 Save All Icon



You are now ready to deploy the Mediator IBSE outbound process. You can follow the same procedure found in Section 5.2.3, "Deploying the Mediator Outbound Process" on page 5-10.

Once deployed, you can invoke the input XML, as defined in Section 5.2.4, "Invoking the Input XML Document in the Oracle Enterprise Manager Console" on page 5-11.

6 Integration With BPM Service Components in the Oracle SOA Suite

Oracle Application Adapter for Siebel integrates seamlessly with Oracle Business Process Management (BPM) to facilitate Web service integration. Oracle BPM is based on the Service-Oriented Architecture (SOA). It consumes adapter services exposed as Web Service Definition Language (WSDL) documents.

This chapter contains the following sections:

- Section 6.1, "Overview"
- Section 6.2, "Deployment of Adapter"
- Section 6.3, "Configuring a New Application Server Connection"
- Section 6.4, "Designing an Outbound BPM Process Using Transformations for Service Integration (J2CA Configuration)"
- Section 6.5, "Designing an Inbound BPM Process Using Transformations for Event Integration (J2CA Configuration)"
- Section 6.6, "Designing an Outbound BPM Process Using Transformations for Service Integration (BSE Configuration)"

6.1 Overview

To integrate with Oracle BPM, Oracle Application Adapter for Siebel must be deployed in the same WLS container as Oracle BPM. The underlying adapter services must be exposed as WSDL files, which are generated during design time in Oracle Adapter Application Explorer (Application Explorer) for both request-response (outbound) and event notification (inbound) services of the adapter. For more information, see "Generating WSDL (J2CA Configurations Only)" on page 2-27.

The generated WSDL files are used to design the appropriate BPM processes for inbound or outbound adapter services. A completed BPM process must be successfully compiled in JDeveloper and deployed to a BPM server. Upon deployment to the BPM server, every newly built process is automatically deployed to the Oracle Enterprise Manager console, where you run, monitor, and administer BPM processes, and listen to adapter events.

6.2 Deployment of Adapter

During installation, Oracle Application Adapter for Siebel is deployed as a J2CA 1.0 resource adapter within the WLS container. The adapter must be deployed in the same WLS container as Oracle BPM.

6.3 Configuring a New Application Server Connection

For more information on how to configure a new Application Server connection in Oracle JDeveloper, see Section 4.3, "Configuring a New Application Server Connection" on page 4-2.

6.4 Designing an Outbound BPM Process Using Transformations for Service Integration (J2CA Configuration)

This section describes how to design an outbound BPM process using transformations for service integration.

A sample project has been provided for this outbound use case scenario in the following folder of the Application Adapters installation:

```
<ADAPTER_HOME>\etc\sample\SIEBEL_Samples.zip\SIEBEL_Samples\BPM\J2CA\Siebel_
Sample_J2CA_BPM_Outbound_Project
```

The following tools are required to complete your outbound design-time configuration:

- Oracle Adapter Application Explorer (Application Explorer)
- Oracle JDeveloper BPM Designer (JDeveloper)

> **Note:** The examples in this chapter demonstrate the use of JDeveloper.

This section contains the following topics:

- Section 6.4.1, "Creating an Empty Composite for BPM"
- Section 6.4.2, "Defining a BPM Outbound Process"
- Section 6.4.3, "Adjusting for Known Deployment Issues With 12c"
- Section 6.4.4, "Deploying the BPM Outbound Process"
- Section 6.4.5, "Invoking the Input XML Document in the Oracle Enterprise Manager Console"

Before you design a BPM process, you must generate the respective WSDL file using Application Explorer. For more information, see Section 4.4.1, "Generating WSDL for Request/Response Service" on page 4-8.

6.4.1 Creating an Empty Composite for BPM

Perform the following steps to create an empty composite for BPM:

1. Create a new BPM application.
2. Enter a name for the new BPM application and click **Next**.

   The Name your project page is displayed.
3. Enter a project name, in the project features select **BPM**, and then click **Next**.

   The Configure SOA settings page is displayed.
4. From the Composite Template list, select **Empty Composite** and click **Finish**.

6.4.2 Defining a BPM Outbound Process

This section describes how to define a BPM outbound process, which contains the following topics:

- Section 6.4.2.1, "Configuring a Third Party Adapter Service Component"
- Section 6.4.2.2, "Configuring an Outbound BPM Process Component"
- Section 6.4.2.3, "Creating a File Adapter for the Write Operation"

6.4.2.1 Configuring a Third Party Adapter Service Component

Perform the following steps to create a third party adapter service component:

1. Double-click the created project to load the components.
2. Drag and drop the **Third Party Adapter** component from the Custom/Thirdparty pane to the External References pane.

   The Create Third Party Adapter Service dialog is displayed.
3. Enter a name for the third party adapter service.
4. Ensure that **Reference** is selected from the Type list (default).
5. Click the **Find existing WSDLs** icon, which is located to the right of the WSDL URL field.

   The WSDL Chooser dialog is displayed.
6. Browse and select an outbound WSDL file from the following directory:

   ```
   <ADAPTER_HOME>\wsdls
   ```
7. Click **OK**.

   The Localize Files dialog is displayed.
8. Click **OK**.

   The outbound WSDL file and associated request and response XML schema files (.xsd) are imported to the project folder that has been created.

   You are returned to the Create Third Party Adapter Service dialog.
9. Click the **Find JCA file** icon, which is located to the right of the JCA File field.

   The Transformation Map dialog is displayed.
10. Browse and select the JCA properties file from the following directory:

    ```
    <ADAPTER_HOME>\wsdls
    ```
11. Click **OK**.

    The Copy File message is displayed.
12. Click **Yes**.

    A copy of the JCA properties file is made in the project folder.

    You are returned to the Create Third Party Adapter Service dialog.
13. Click **OK**.

    The third party adapter service component is created and displayed in the External References pane.

    You are now ready to configure an outbound BPM process component.

For more detailed information, including screen shots, see Section 4.4.3.1, "Configuring a Third Party Adapter Service Component" on page 4-11.

6.4.2.2 Configuring an Outbound BPM Process Component

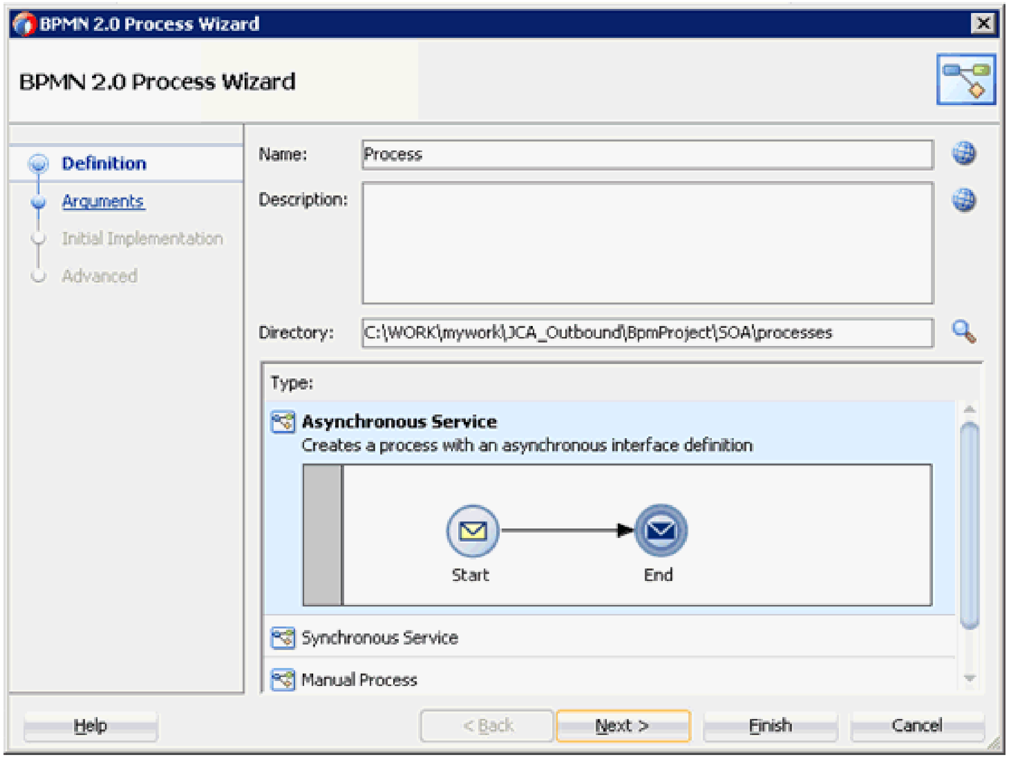
This section describes how to configure an outbound BPM process component.

Perform the following steps to configure an outbound BPM process component:

1. Drag and drop the **BPMN Process** component from the Components pane to the Components pane.

   The Create BPMN Process dialog is displayed, as shown in Figure 6–1.

Figure 6–1 Create BPMN Process Dialog



Name: Process

Description:

Directory: C:\WORK\mywork\JCA_Outbound\BpmProject\SOA\processes

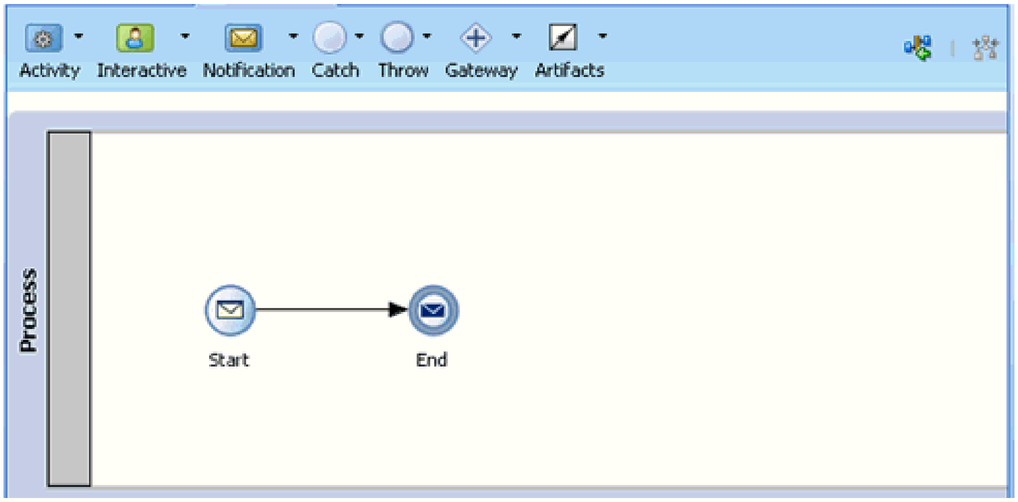
Type:

Asynchronous Service
Creates a process with an asynchronous interface definition

2. Accept the default option that is selected under the Type area (Asynchronous Service) and click **Finish**.

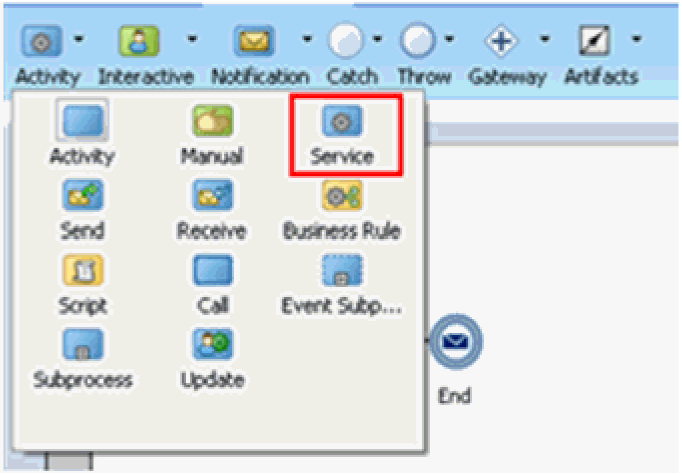
   The BPMN process is displayed, as shown in Figure 6–2.

Figure 6–2 BPMN Process



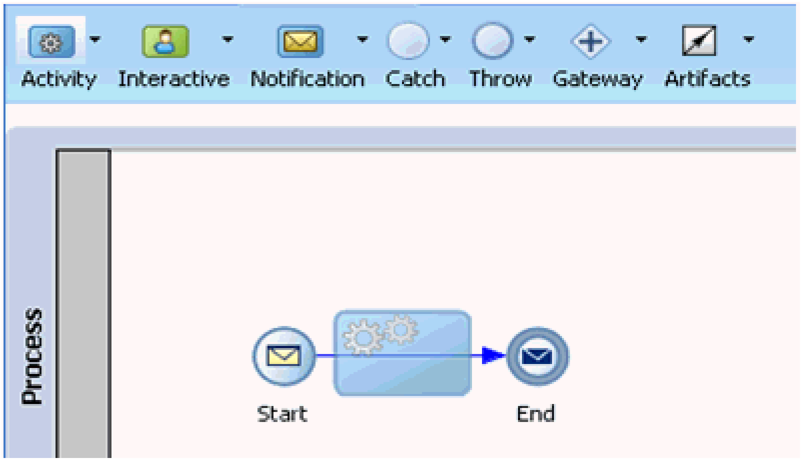
3. Click the **Activity** drop-down menu and select **Service**, as shown in Figure 6–3.

Figure 6–3 Activity Drop-down Menu



4. Drop the Service icon on the wire between the Start and End event components, as shown in Figure 6–4.

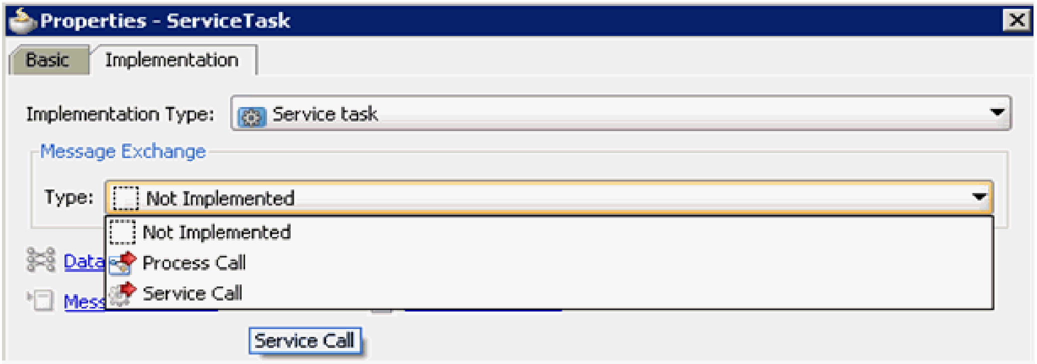
Figure 6–4 Activity Icon



The Properties - ServiceTask window is displayed.

5. Click the **Implementation** tab.
6. Select **Service Call** from the Message Exchange Type list, as shown in Figure 6–5.

Figure 6–5 Service Call

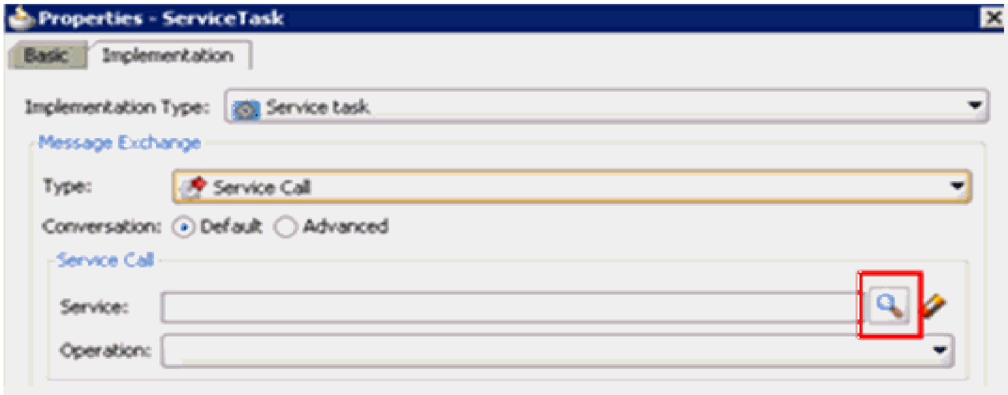


Implementation Type: Service task

Type: Not Implemented

7. Click the Browse icon to the right of the Service field, as shown in Figure 6–6.

Figure 6–6 Browse Icon



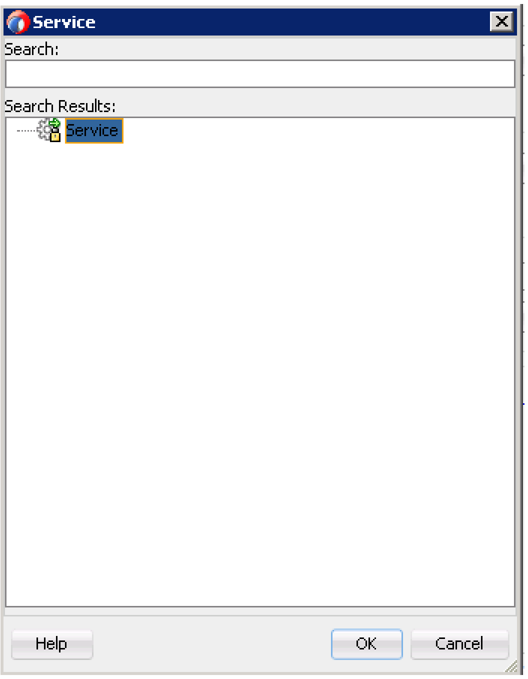
Implementation Type: Service task

Type: Service Call

Conversation: ● Default ○ Advanced

The Service dialog is displayed, as shown in Figure 6–7.

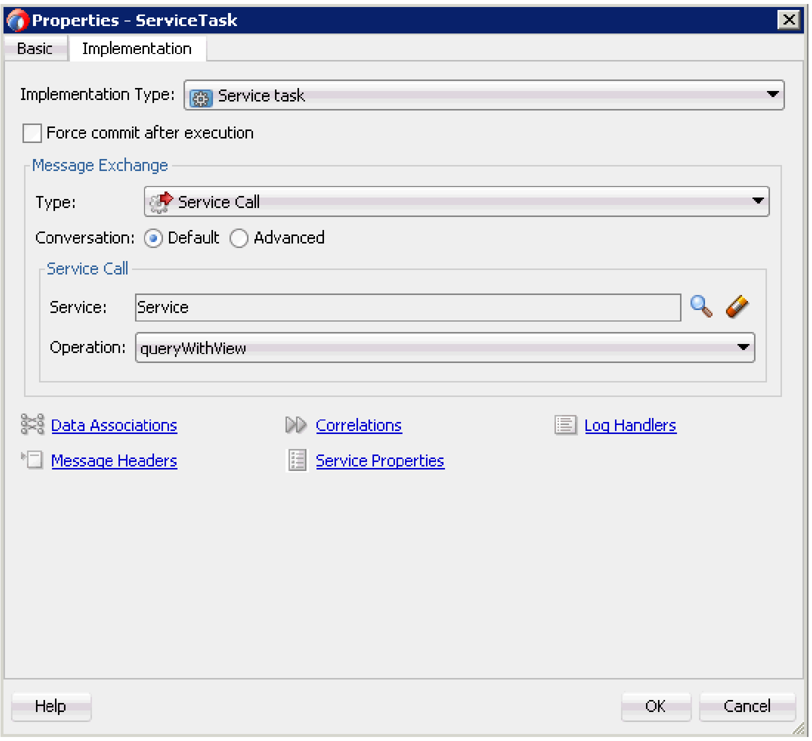
Figure 6–7 Service Dialog



8. Select the Third Party Service that has been created and click **OK**.

   You are returned to the Properties - ServiceTask dialog, as shown in Figure 6–8.

Figure 6–8 Properties - ServiceTask Dialog



Implementation Type: Service task

□ Force commit after execution

Type: Service Call

Conversation: ● Default ○ Advanced

Service: Service

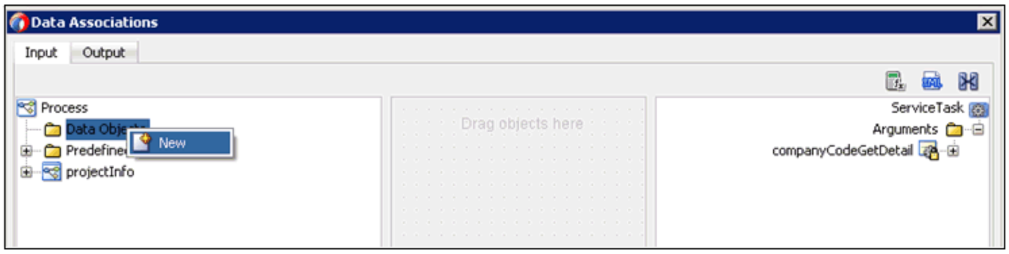
Operation: queryWithView

9. Click the **Data Associations** hyperlink.

   The Data Associations dialog is displayed.

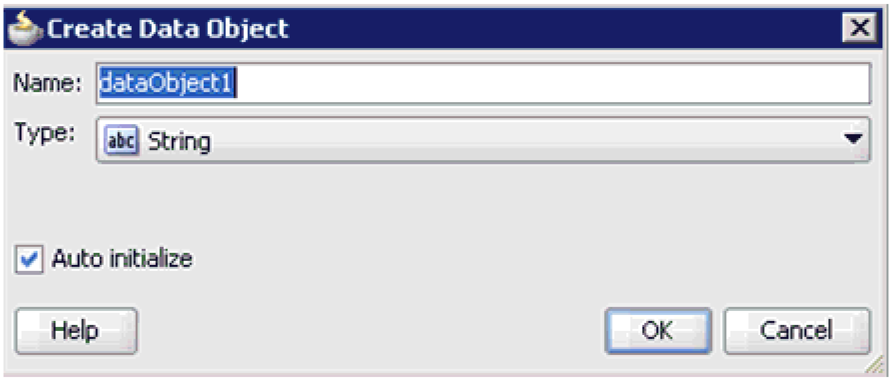
10. Right-click the **Data Objects** node in the left pane under Process, and select **New** as shown in Figure 6–9.

Figure 6–9 New Option



The Create Data Object dialog is displayed, as shown in Figure 6–10.

Figure 6–10 Create Data Object Dialog



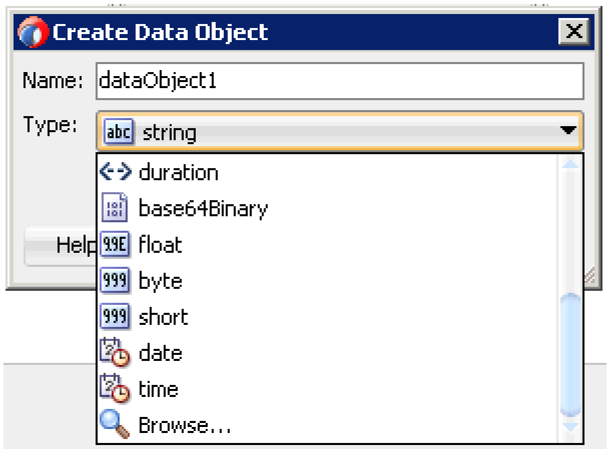
Name: dataObject1

Type: String

☑ Auto initialize

11. Enter a name in the Name field (for example, Request), click the drop-down button in the Type field, and select **Browse** from the list, as shown in Figure 6–11.

Figure 6–11 Create Data Object Dialog

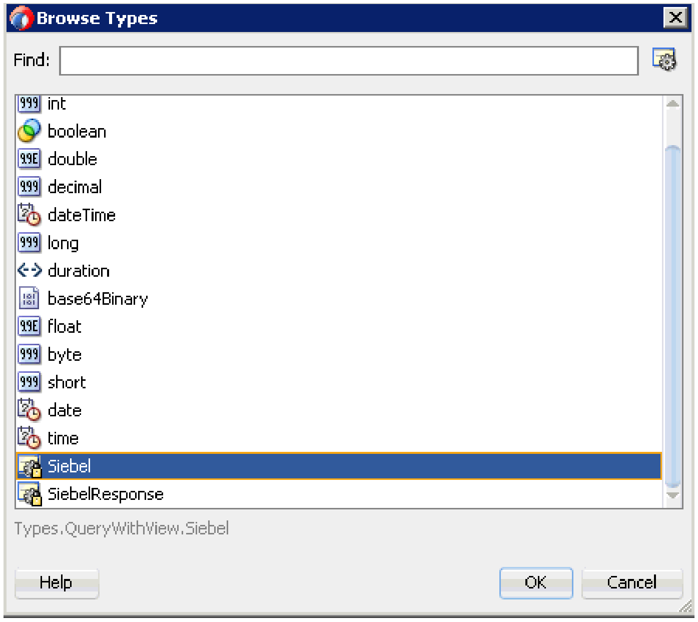


Name: dataObject1

Type: string

The Browse Types dialog is displayed, as shown in Figure 6–12.

Figure 6–12 Browse Types Dialog



12. Select the first component (for example, Siebel) and click **OK**.

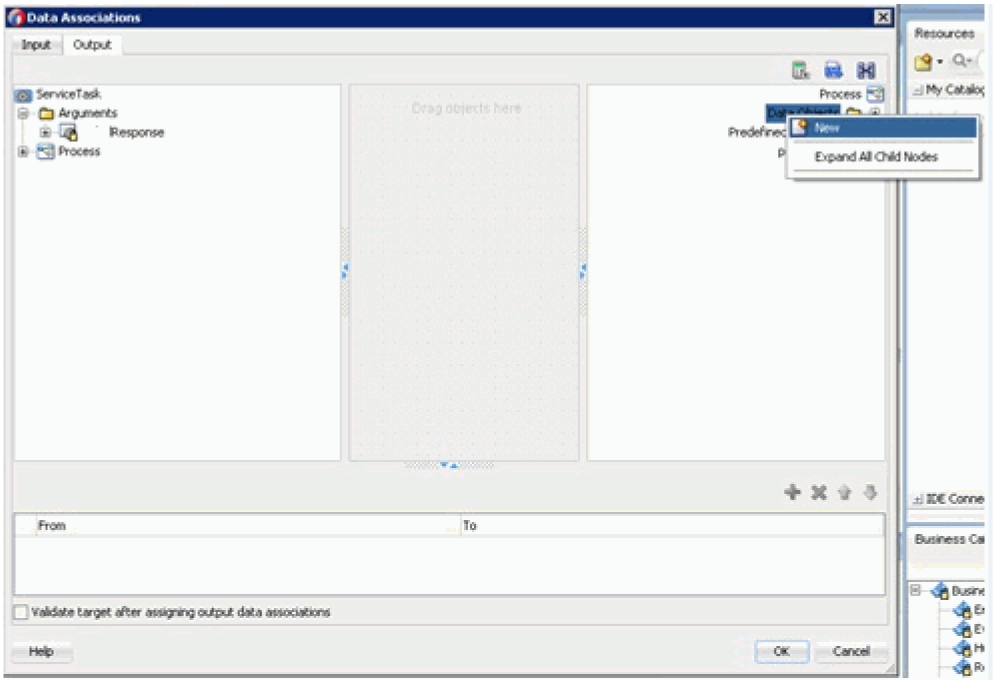
    You are returned to the Create Data Object dialog.

13. Click **OK**.

    The Data Object (for example, Request) that has been created is displayed under the Data Objects node in the Data Associations dialog.

14. Create another Data Object by right-clicking the **Data Objects** node in the right pane of the Output tab and selecting **New**, as shown in Figure 6–13.

Figure 6–13 Data Associations Dialog

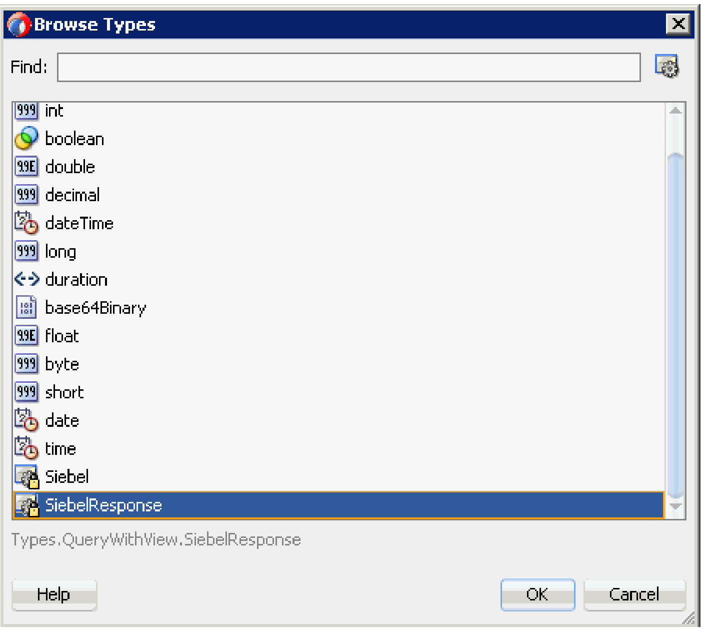


The Create Data Object dialog is displayed.

15. Enter a name in the Name field (for example, Response), and then click the drop-down button in the Type field and select **Browse** from the list.

The Browse Types dialog is displayed, as shown in Figure 6–14.

Figure 6–14 Browse Types Dialog



16. Select the second component (for example, SiebelResponse) and click **OK**.

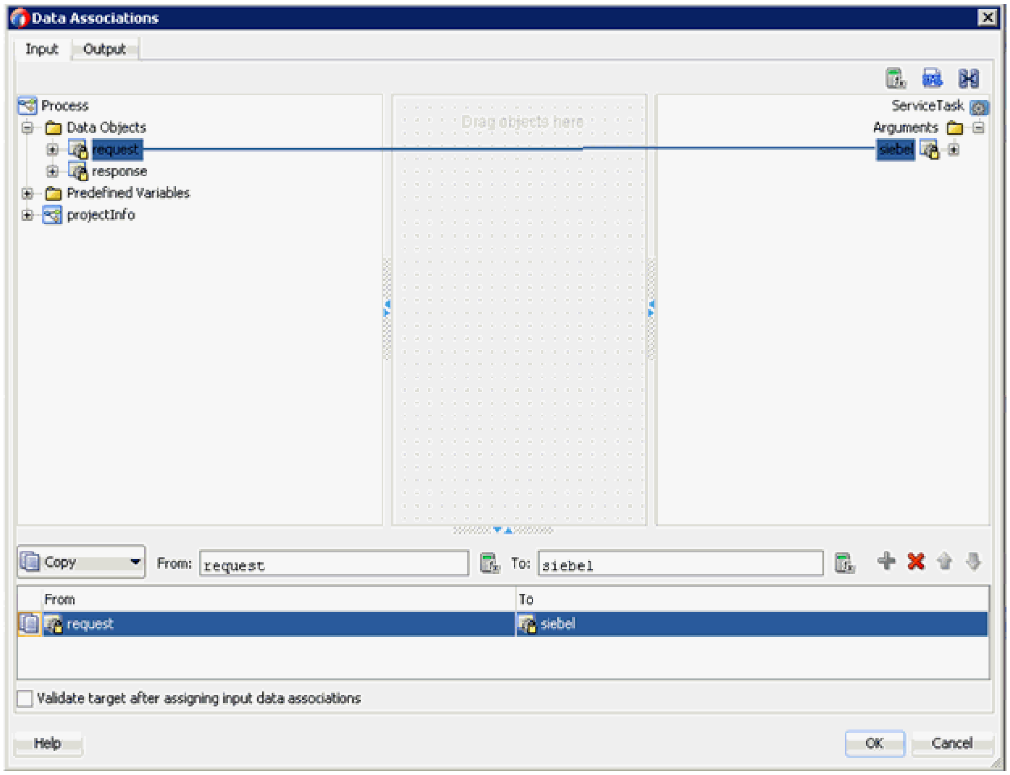
You are returned to the Create Data Object dialog.

17. Click **OK**.

    The Data Object (for example, Response) that has been created is displayed under the Process node in the Data Associations dialog.

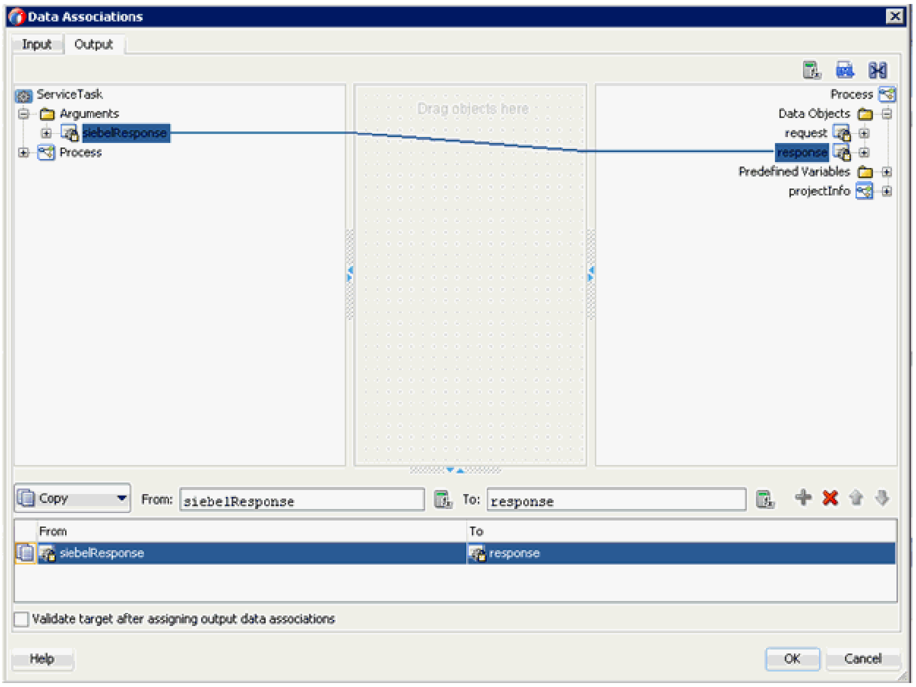
18. Select the **Request** Data Object under the Data Objects node in the left pane of the Input tab and drag and connect it to Siebel under the Arguments node in the right pane, as shown in Figure 6–15.

Figure 6–15 Request Data Object



19. Click on the **Output** tab and select **SiebelResponse** under the Arguments node in the left pane and drag and connect it to the Response Data Object under the Data Objects node, as shown in Figure 6–16.

Figure 6–16 Response Data Object



From: siebelResponse

To: response

From	To
siebelResponse	response

□ Validate target after assigning output data associations

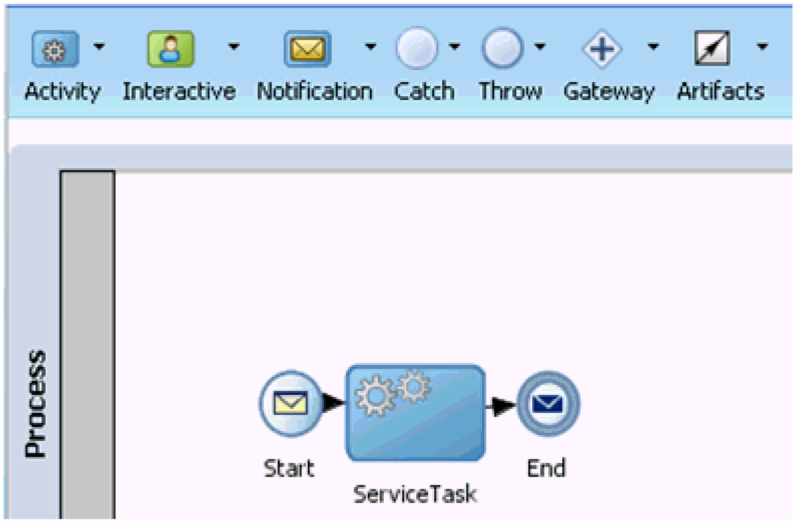
20. Click **OK**.

    You are returned to the Properties - ServiceTask dialog.

21. Click **OK**.

    The Service Task is created between the Start and End Event components, as shown in Figure 6–17.

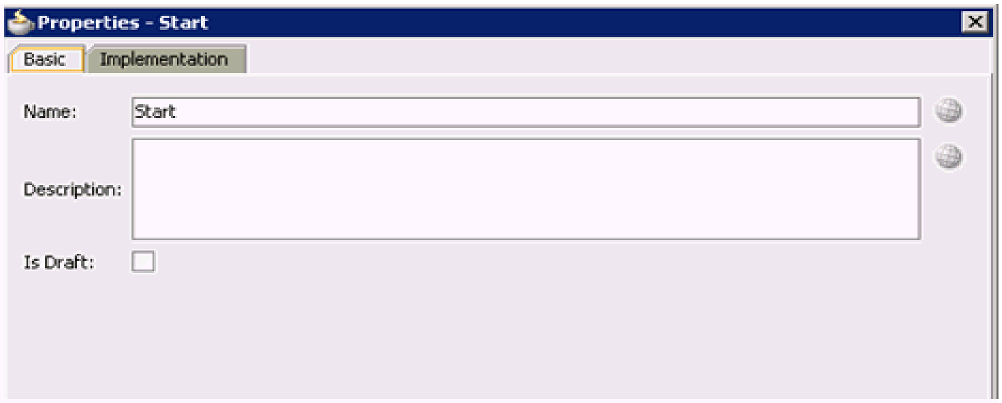
Figure 6–17 Service Task



22. Save the process and double-click the Start event component.

    The Properties - Start dialog is displayed, as shown in Figure 6–18.

Figure 6–18 Properties - Start Dialog



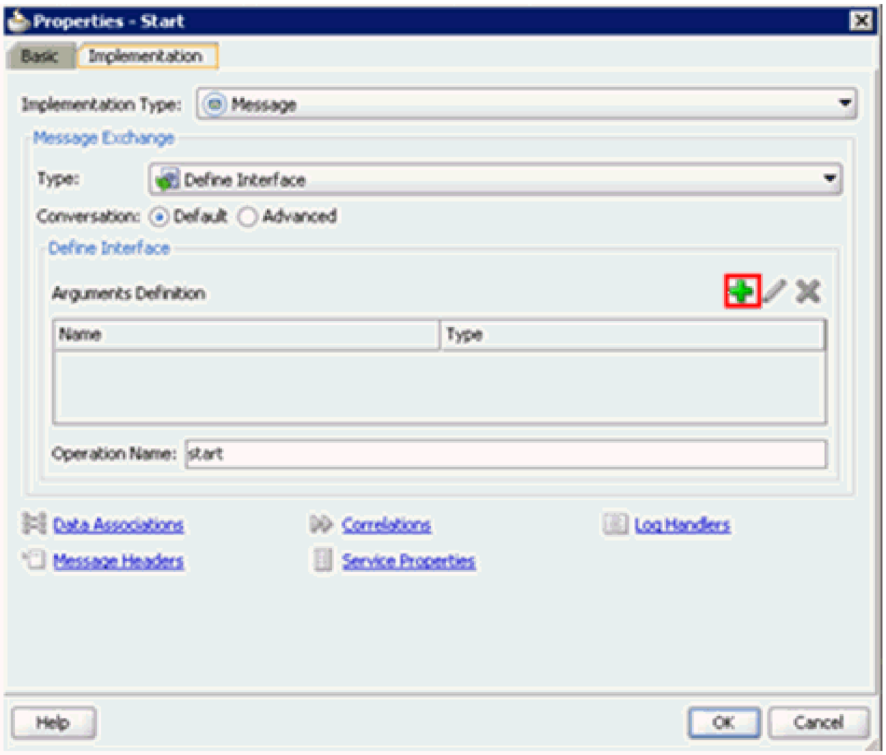
Name: Start

Description:

Is Draft: □

23. Click the **Implementation** tab, as shown in Figure 6–19.

Figure 6–19 Implementation Tab



Implementation Type: Message

Type: Define Interface

Conversation: ● Default ○ Advanced

Name	Type

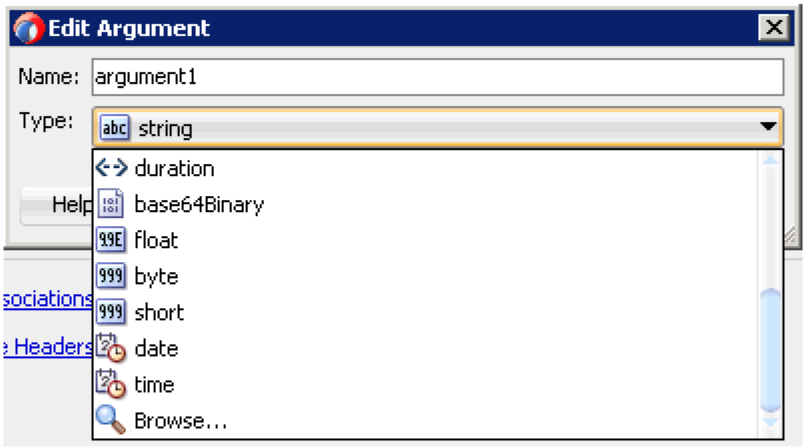
Operation Name: start

24. Click the **Plus** icon to the right of the Arguments Definition field.

    The Create Argument dialog is displayed.

25. Enter a name in the Name field (by default, argument1), and then click the drop-down button in the Type field and select **Browse** from the list, as shown in Figure 6–20.

Figure 6–20 Create Argument Dialog



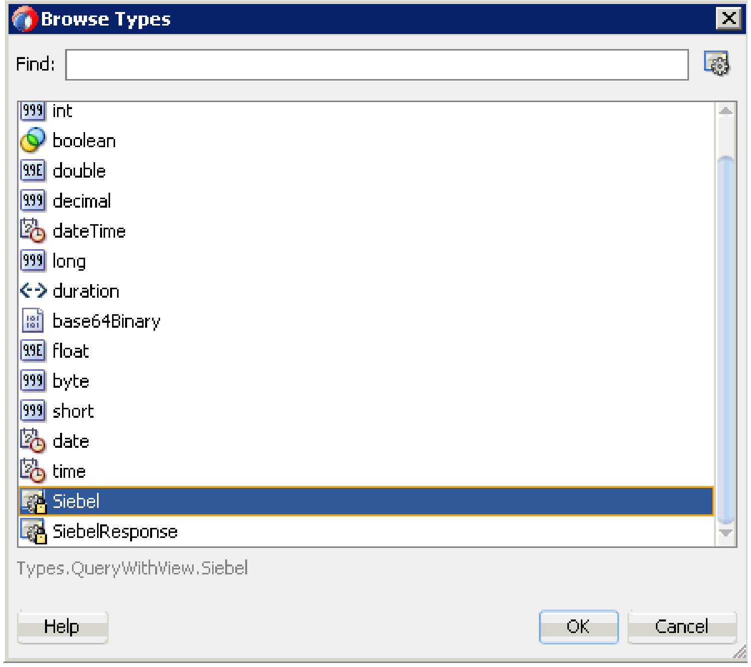
Edit Argument

Name: argument1

Type: string

The Browse Types dialog is displayed, as shown in Figure 6–21.

Figure 6–21 Browse Types Dialog



Browse Types

Find:

Siebel

Types.QueryWithView.Siebel

26. Select the first component (for example, Siebel) and click **OK**.

You are returned to the Create Argument dialog.

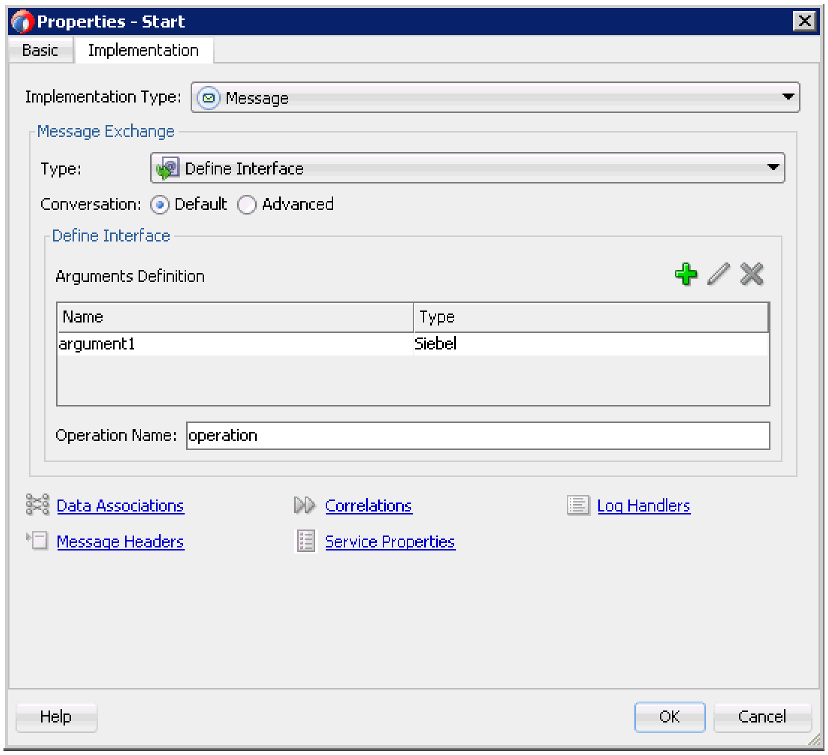
27. Click **OK**.

You are returned to the Properties - Start dialog.

28. In the Operation Name field, change **start** (default) to **operation** as shown in Figure 6–22.

Note: This change is necessary to work with old BPM payloads.

Figure 6–22 Operation Name Field



Properties - Start

Basic | Implementation

Implementation Type: Message

Message Exchange

Type: Define Interface

Conversation: ● Default ○ Advanced

Define Interface

Arguments Definition

Name	Type
argument1	Siebel

Operation Name: operation

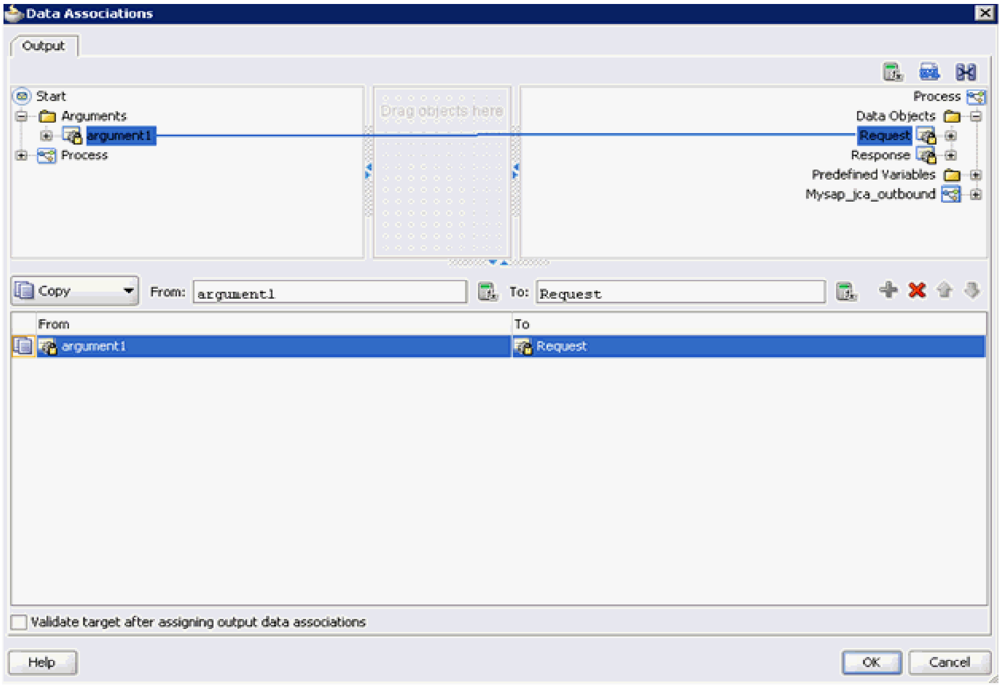
29. Click the **Data Associations** hyperlink.

    The Data Associations dialog is displayed.

30. Select **arguments1** under the Arguments node in the left pane and drag and connect it to the **Request** Data Object under Data Objects in the right pane.

31. Click **OK** as shown in Figure 6–23.

Figure 6–23 OK Button



From: argument1

To: Request

From	To
argument1	Request

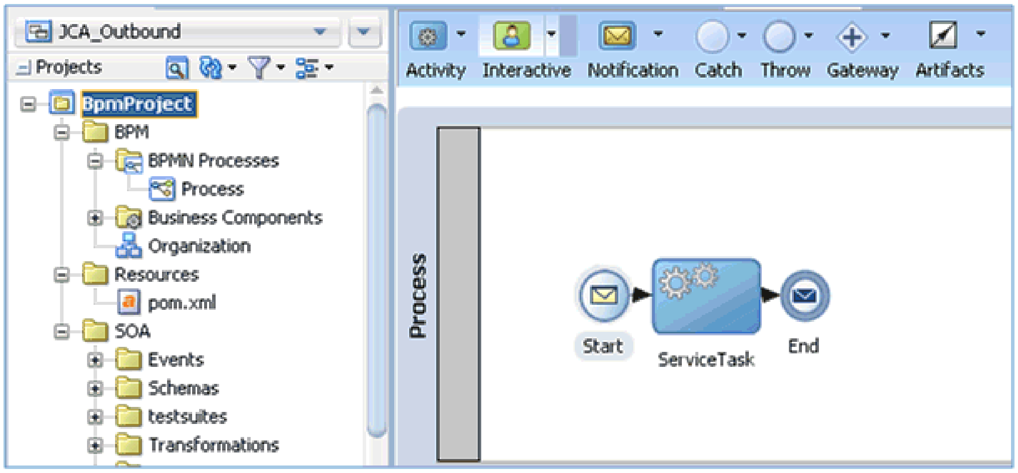
□ Validate target after assigning output data associations

You are returned to the Properties - Start dialog.

32. Click **OK**.

    You are returned to the Process workspace area, as shown in Figure 6–24.

Figure 6–24 Process Workspace Area



33. Double-click the created project to load the components.

34. Click the **Save All** icon in the menu bar to save the new outbound BPM process component that was configured.

    You are now ready to create a File adapter for the write operation.

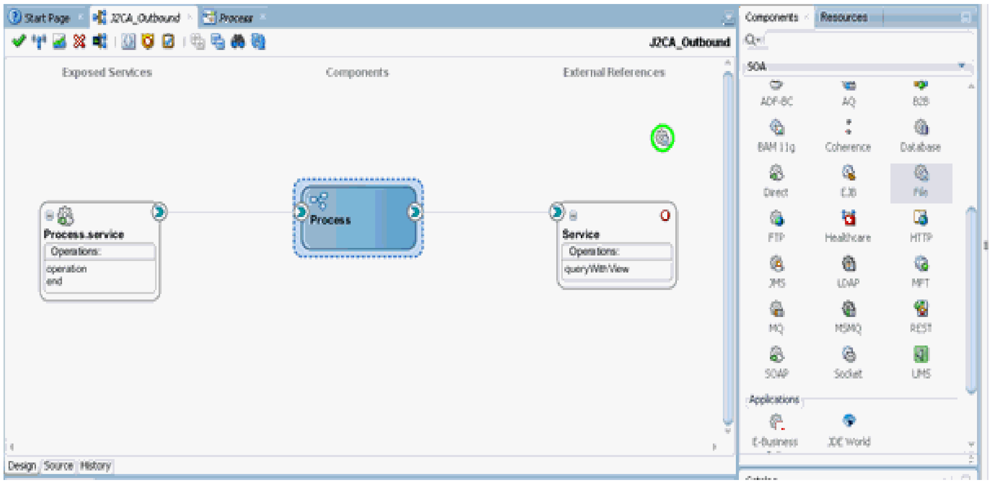
6.4.2.3 Creating a File Adapter for the Write Operation

This section describes how to create a File adapter for the write operation.

Perform the following steps to create a File adapter for the write operation:

1. Drag and drop the **File Adapter** component from the Technology Adapters pane to the External References pane, as shown in Figure 6–25.

Figure 6–25 File Adapter Component



The Adapter Configuration Wizard is displayed.

2. Provide a Reference Name (for example, FileWrite).
3. Click **Next**.

   The Adapter Interface page is displayed.

4. Ensure that the **Define from operation and schema (specified later)** option is selected.
5. Click **Next**.

   The File Server Connection page is displayed.

6. Click **Next**.

   The Operation page is displayed.

7. Select **Write File** from the list of Operation Type options and specify an Operation Name (for example, Write).
8. Click **Next**.

   The File Configuration page is displayed.

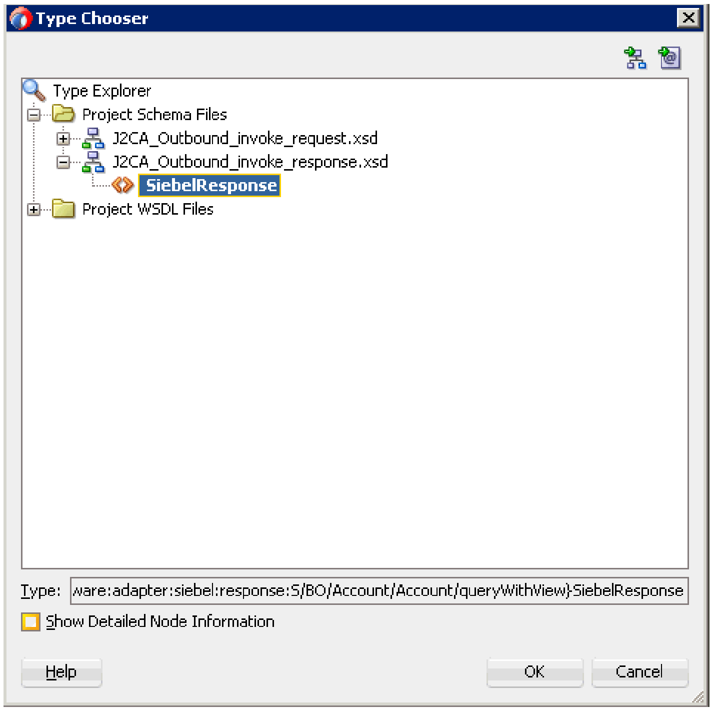
9. Specify a location on your file system where the output file is written.
10. In the File Naming Convention field, specify a name for the output file.
11. Click **Next**.

    The Messages page is displayed.

12. Click **Browse**, which is located to the right of the URL field.

    The Type Chooser dialog is displayed, as shown in Figure 6–26.

Figure 6–26 Type Chooser Dialog


Type Chooser
Type Explorer
Project Schema Files
J2CA_Outbound_invoke_request.xsd
J2CA_Outbound_invoke_response.xsd
SiebelResponse
Project WSDL Files
Show Detailed Node Information
Help
OK
Cancel

Type: ware:adapter:siebel:response:S/BO/Account/Account/queryWithView}SiebelResponse

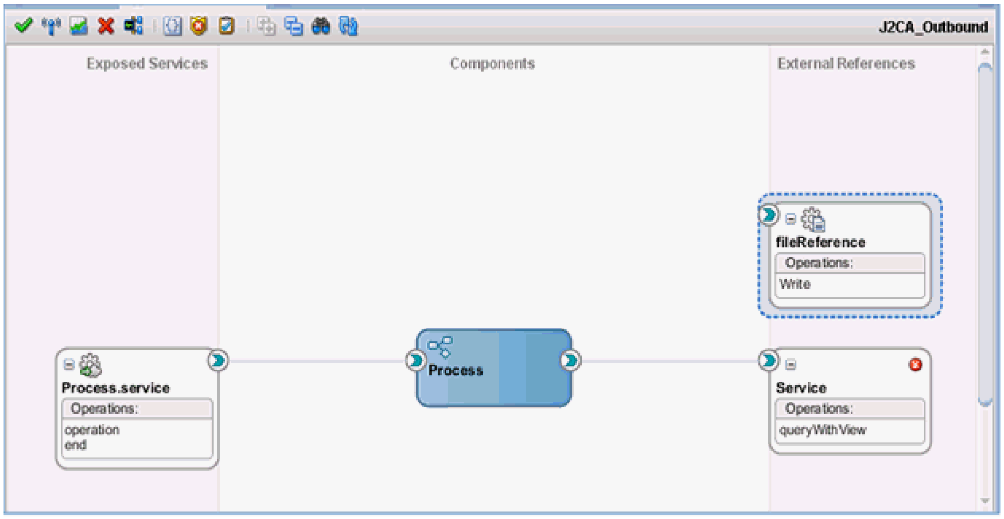
13. Expand **Project Schema Files** and **J2CA_Outbound_invoke_response.xsd**.
14. Select the available schema (for example, SiebelResponse).
15. Click **OK**.

    You are returned to the Messages page.
16. Click **Next**.

    The Finish page is displayed.
17. Click **Finish**.

    The File Adapter service is created in the External References pane, as shown in Figure 6–27.

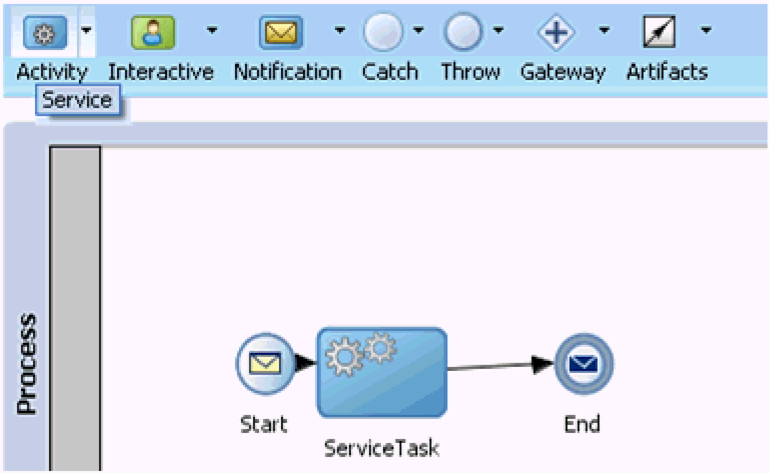
Figure 6–27 File Adapter Service



18. Double-click the BPMN Process component.

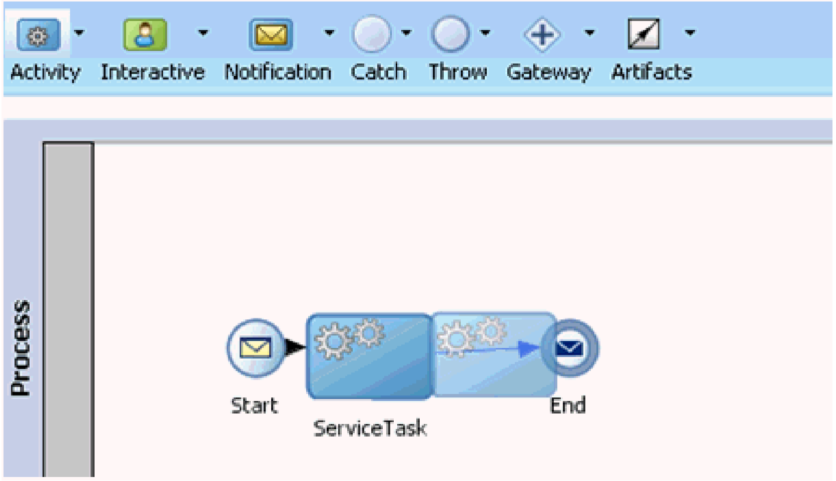
    The BPMN process is displayed, as shown in Figure 6–28.

Figure 6–28 BPMN Process



19. Click the **Activity** icon, and select **Service**.
20. Drop the Service icon on the wire between the Service Task and End event components, as shown in Figure 6–29.

Figure 6–29 Activity Icon

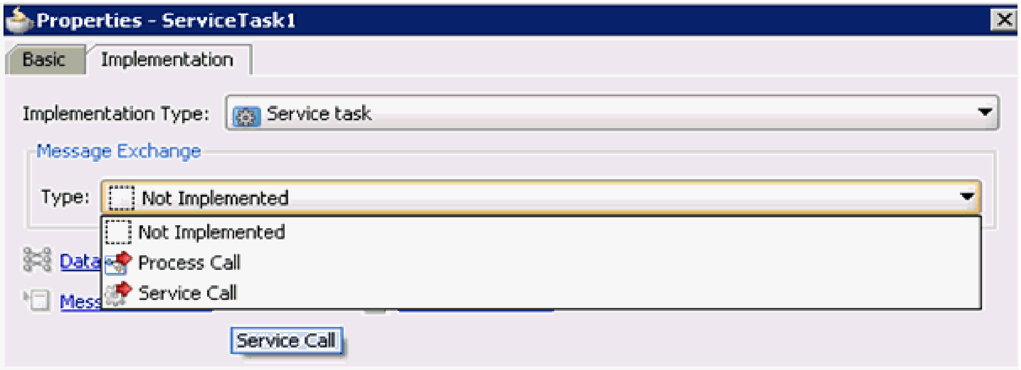


The Properties - ServiceTask1 dialog is displayed.

21. Click the **Implementation** tab.

22. Select **Service Call** from the Type drop-down list in the Message Exchange section, as shown in Figure 6–30.

Figure 6–30 Service Call



Properties - ServiceTask1

Implementation Type: Service task

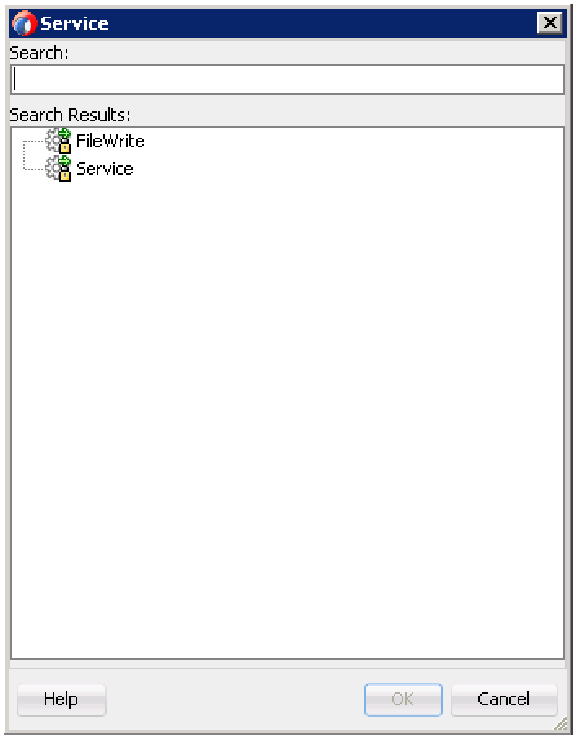
Message Exchange

Type: Not Implemented

23. Click the **Browse** icon to the right of the Service field.

The Service dialog is displayed, as shown in Figure 6–31.

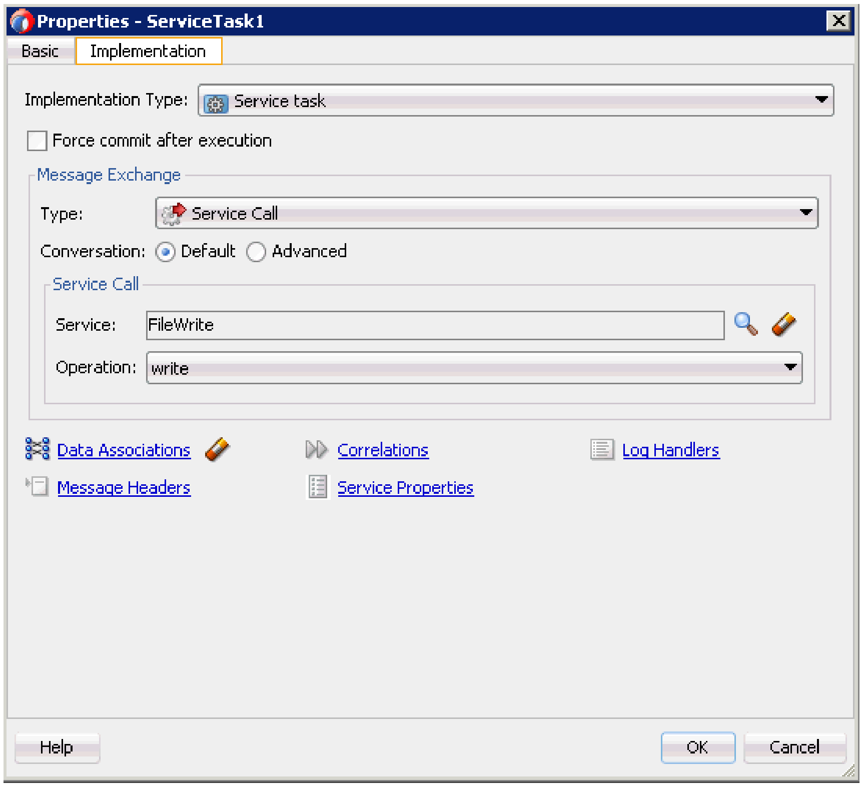
Figure 6–31 Service Dialog



24. Select the service for write operation that has been created (for example, FileWrite) and click **OK**.

    You are returned to the Properties - ServiceTask1 dialog, as shown in Figure 6–32.

Figure 6–32 Properties - ServiceTask1 Dialog



Properties - ServiceTask1

Basic | Implementation

Implementation Type: Service task

□ Force commit after execution

Message Exchange

Type: Service Call

Conversation: ● Default ○ Advanced

Service Call

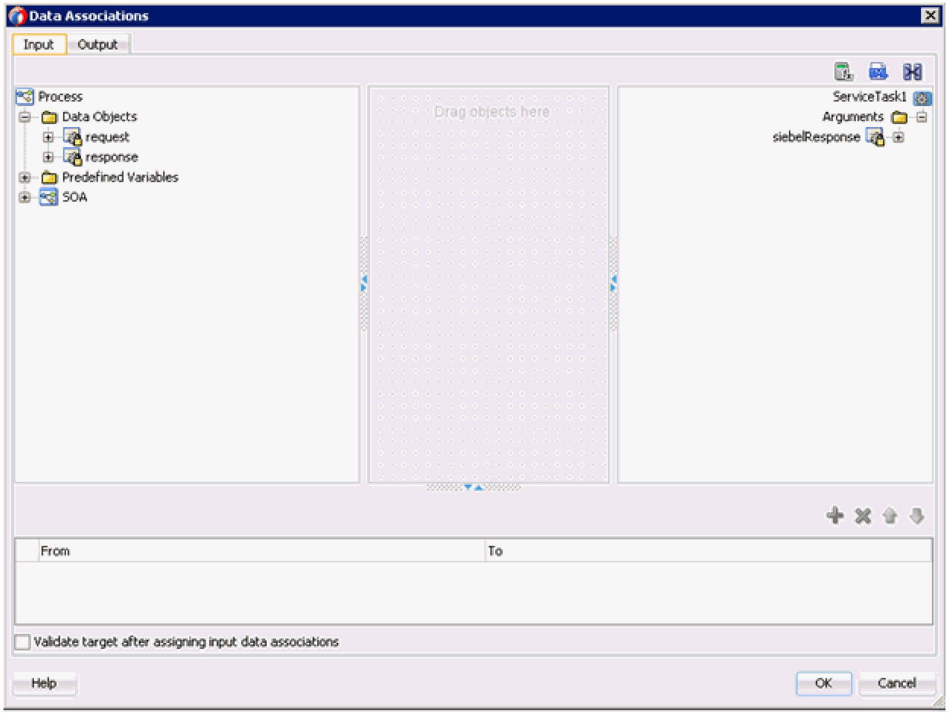
Service: FileWrite

Operation: write

25. Click the **Data Associations** hyperlink.

    The Data Associations dialog is displayed, as shown in Figure 6–33.

Figure 6–33 Data Associations Dialog



Data Associations

Input | Output

Process
- Data Objects
  - request
  - response
- Predefined Variables
- SOA

Drag objects here

ServiceTask1
Arguments
siebelResponse

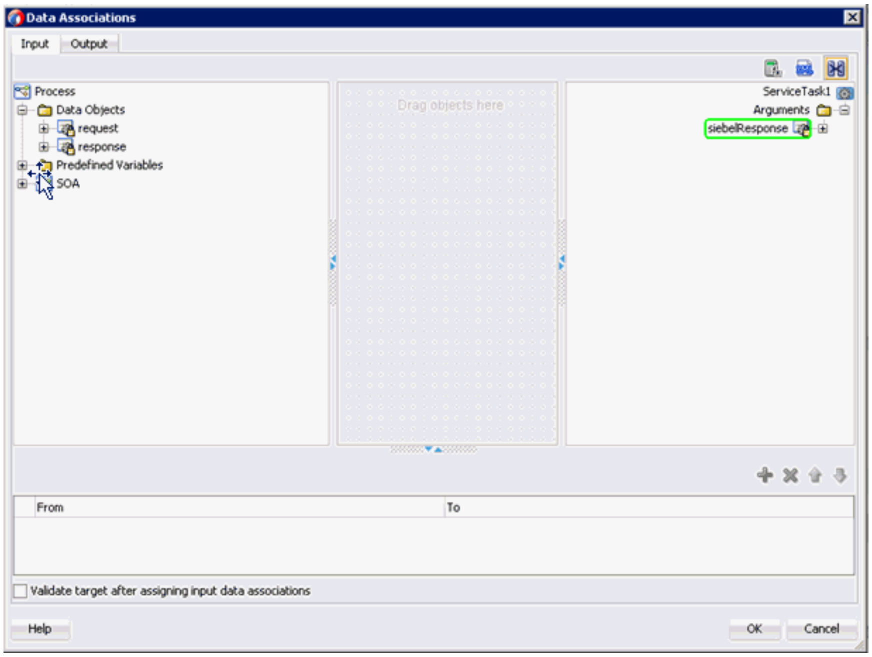
From	To

□ Validate target after assigning input data associations

26. In the Input tab, click the XSL Transformation icon in the top right pane.

27. Drag and drop the XSL Transformation icon to the **SiebelResponse** node, as shown in Figure 6–34.

Figure 6–34 CompanyCodeSiebelResponse Node

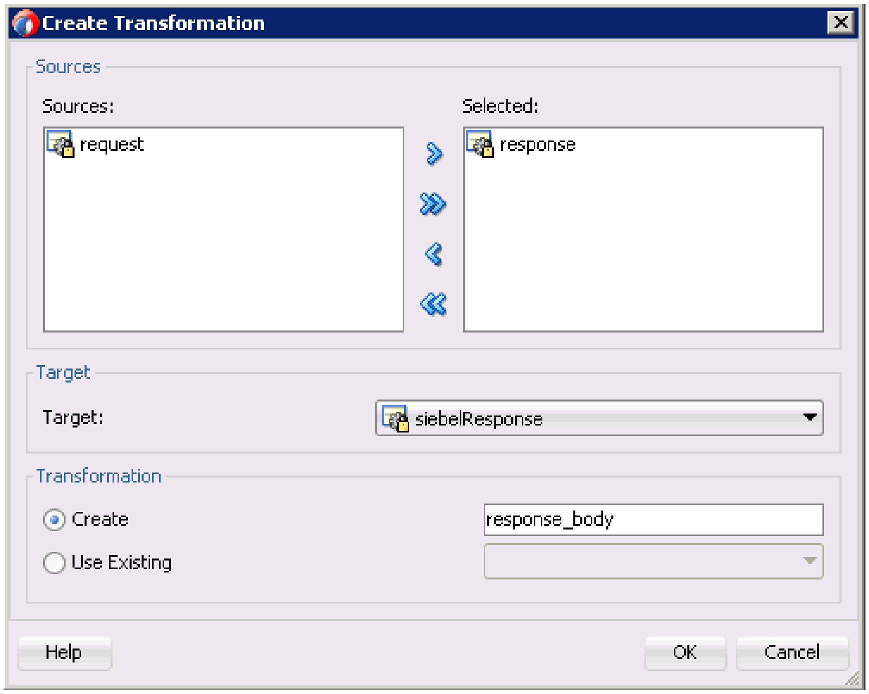


The Create Transformation dialog is displayed.

28. Select **Response** in the Sources section and click the right arrow symbol.

The Response object is added to the Selected elements area as shown in Figure 6–35.

Figure 6–35 Response Object



Create Transformation

Sources

Sources: request

Selected: response

Target

Target: siebelResponse

Transformation

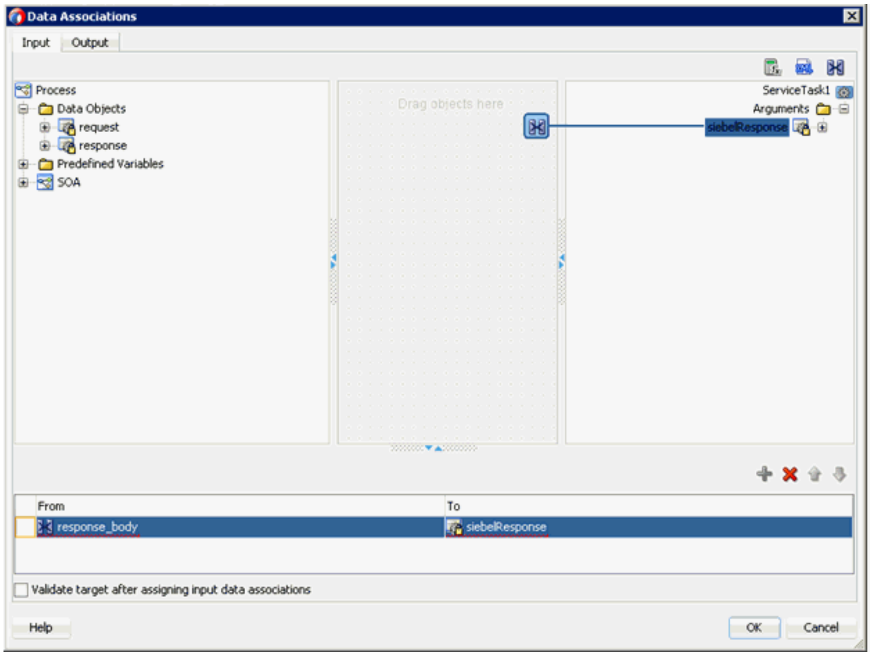
● Create: response_body

○ Use Existing

29. Accept the default value selected in the Target drop-down list and the default name in the Create field by clicking **OK**.

    You are returned to the Data Associations dialog window with the XSL transformation created, as shown in Figure 6–36.

Figure 6–36 Data Associations Dialog



Data Associations

From	To
response_body	siebelResponse

□ Validate target after assigning input data associations

30. Click **OK**.

    You are returned to the Properties - ServiceTask1 dialog.

31. Click **OK**.

    The Response_body.xsl tab is displayed.

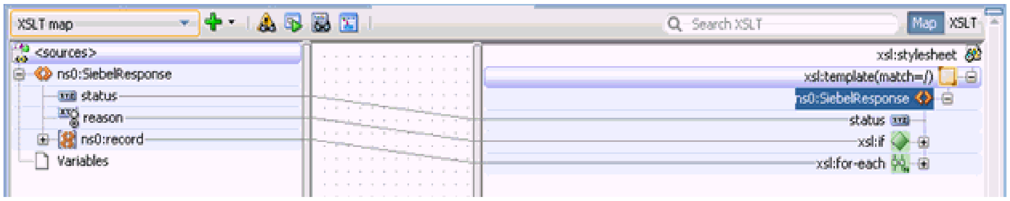
32. Automap the Source and Target elements.

    The Auto Map Preferences dialog is displayed.

33. Accept the default values and click **OK**.

    The transformation is completed, as shown in Figure 6–37.

Figure 6–37 Completed Transformation



34. Save the transformation.

35. Return to the Process workspace area.

    The ServiceTask1 component is created between the ServiceTask component and the End event component.

36. Click the **Save All** icon in the menu bar to save the new outbound BPM process component that was configured.

    You are now ready to deploy the outbound BPM process.

6.4.3 Adjusting for Known Deployment Issues With 12c

For more information on how to adjust for known deployment issues with 12c, see Section 4.4.3.3, "Adjusting for Known Deployment Issues With 12c" on page 4-26.

6.4.4 Deploying the BPM Outbound Process

Perform the following steps to deploy the Mediator outbound process.

1. Right-click the project name in the left pane, select **Deploy**, and then click **J2CA_Outbound**.

   The Deployment Action page is displayed.

2. Ensure that **Deploy to Application Server** is selected.

3. Click **Next**.

   The Deploy Configuration page is displayed.

4. Leave the default values selected and click **Next**.

   The Select Server page is displayed.

5. Select an available application server that was configured and click **Next**.

   The SOA Servers page is displayed.

6. Select a target SOA server and click **Next**.

   The Summary page is displayed.

7. Review and verify all the available deployment information for your project and click **Finish**.

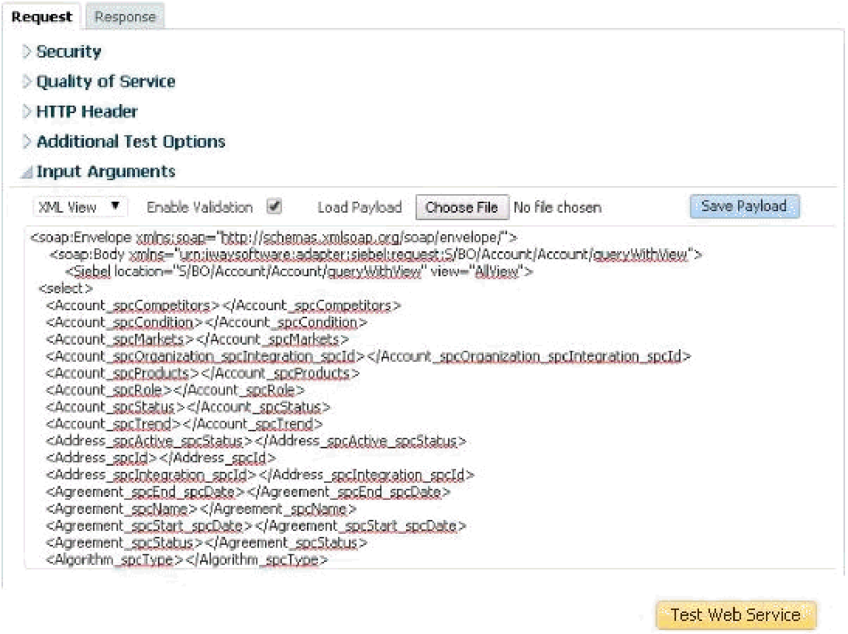
   For more information, see Section 4.4.4, "Deploying the BPEL Outbound Process" on page 4-28.

6.4.5 Invoking the Input XML Document in the Oracle Enterprise Manager Console

Perform the following steps to invoke the input XML document in the Oracle Enterprise Manager console.

1. Logon to the Oracle Enterprise Manager console.
2. Expand your domain in the left pane followed by the **SOA** folder.
3. Select an available project (for example, J2CA_Outbound).
4. Click **Test**.
5. Click the **Request** tab.

Figure 6–38 Request Tab

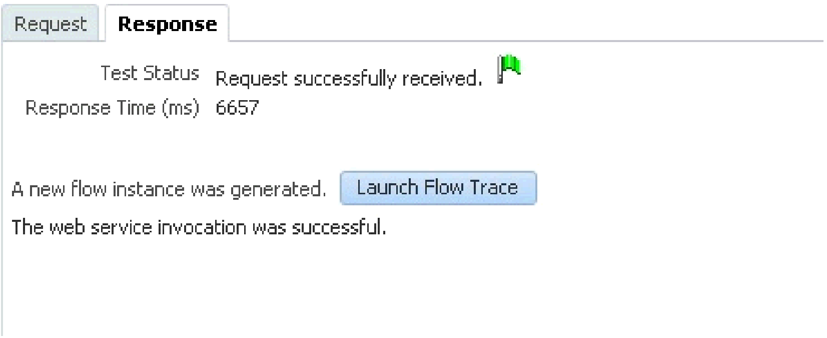


```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body xmlns="urn:iwaysoftware:adapter:siebel:request:S/BO/Account/Account/queryWithView">
      <Siebel location="S/BO/Account/Account/queryWithView" view="AllView">
 <select>
  <Account_spcCompetitors></Account_spcCompetitors>
  <Account_spcCondition></Account_spcCondition>
  <Account_spcMarkets></Account_spcMarkets>
  <Account_spcOrganization_spcIntegration_spcId></Account_spcOrganization_spcIntegration_spcId>
  <Account_spcProducts></Account_spcProducts>
  <Account_spcRole></Account_spcRole>
  <Account_spcStatus></Account_spcStatus>
  <Account_spcTrend></Account_spcTrend>
  <Address_spcActive_spcStatus></Address_spcActive_spcStatus>
  <Address_spcId></Address_spcId>
  <Address_spcIntegration_spcId></Address_spcIntegration_spcId>
  <Agreement_spcEnd_spcDate></Agreement_spcEnd_spcDate>
  <Agreement_spcName></Agreement_spcName>
  <Agreement_spcStart_spcDate></Agreement_spcStart_spcDate>
  <Agreement_spcStatus></Agreement_spcStatus>
  <Algorithm_spcType></Algorithm_spcType>
```

6. Provide an appropriate input value in the Value field and click **Test Web Service**, as shown in Figure 6–38.

   A response is received in the Response tab to indicate that invocation was successful in the Oracle Enterprise Manager console, as shown in Figure 6–39.

Figure 6–39 Received Response



Test Status Request successfully received.

Response Time (ms) 6657

A new flow instance was generated.

The web service invocation was successful.

7. Navigate to the defined output directory on your file system and open the XML response document that was received.

   The XML response document contains the generated output with values.

6.5 Designing an Inbound BPM Process Using Transformations for Event Integration (J2CA Configuration)

This section demonstrates how Oracle Application Adapter for Siebel integrates with Siebel to receive event data.

A sample project has been provided for this inbound use case scenario in the following folder of the Application Adapters installation:

```
<ADAPTER_HOME>\etc\sample\SIEBEL_Samples.zip\SIEBEL_Samples\BPM\J2CA\Inbound_
Project
```

The following tools are required to complete your adapter design-time configuration:

- Oracle Adapter Application Explorer (Application Explorer)
- Oracle JDeveloper BPM Designer (JDeveloper)

> **Note:** The examples in this chapter demonstrate the use of JDeveloper.

This section contains the following topics:

- Section 6.5.1, "Creating an Empty Composite for BPM"
- Section 6.5.2, "Defining a BPM Inbound Process"

Before you design a BPM process, you must generate the respective WSDL file using Application Explorer. For more information, see Section 4.5.1, "Generating WSDL for Event Integration" on page 4-34.

6.5.1 Creating an Empty Composite for BPM

For more information on how to configure a new Application Server connection in Oracle JDeveloper, see Section 4.3, "Configuring a New Application Server Connection" on page 4-2.

6.5.2 Defining a BPM Inbound Process

This section describes how to define a BPM inbound process, which contains the following topics:

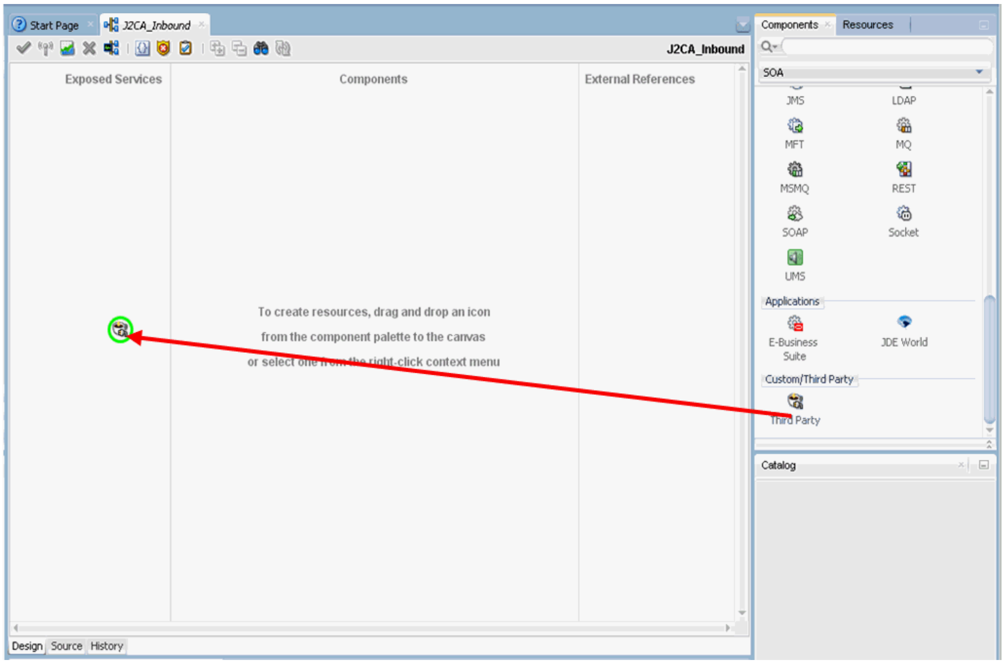
- Section 6.5.2.1, "Configuring a Third Party Adapter Service Component"
- Section 6.5.2.2, "Configuring an Inbound BPM Process Component"
- Section 6.5.2.3, "Creating a File Adapter for the Write Operation"
- Section 6.5.2.4, "Adjusting for Known Deployment Issues With 12c"

6.5.2.1 Configuring a Third Party Adapter Service Component

Perform the following steps to create a third party adapter service component:

1. Double-click the created project to load the components.
2. Drag and drop the **Third Party Adapter** component from the Custom/Thirdparty pane to the Exposed References pane, as shown in Figure 6–40.

Figure 6–40 Third Party Adapter Component



To create resources, drag and drop an icon from the component palette to the canvas or select one from the right-click context menu

The Create Third Party Adapter Service dialog is displayed.

3. Enter a name for the third party adapter service.
4. Ensure that **Service** is selected from the Type list (default).
5. Click the **Find existing WSDLs** icon, which is located to the right of the WSDL URL field.

   The WSDL Chooser dialog is displayed.

6. Select **File System**, and then browse and select an inbound WSDL file from the following directory:

   ```
   <ADAPTER_HOME>\wsdls
   ```

7. Click **OK**.

The Localize Files dialog is displayed.

8. Click **OK**.

   The inbound WSDL file and associated receive_request XML schema file (.xsd) are imported to the project folder that has been created.

   You are returned to the Create Third Party Adapter Service dialog.

9. Click the **Find JCA file** icon, which is located to the right of the JCA File field.

   The Transformation Chooser dialog is displayed.

10. Select **File System**, and then browse and select the JCA properties file from the following directory:

    ```
    <ADAPTER_HOME>\wsdls
    ```

11. Click **OK**.

    The Copy File message is displayed.

12. Click **Yes**.

    A copy of the JCA properties file is made in the project folder.

    You are returned to the Create Third Party Adapter Service dialog.

13. Click **OK**.

    The third party adapter service component (matmas) is created in the Exposed References pane.

    You are now ready to configure an inbound BPM process component.

    For more information, see Section 4.5.3.1, "Creating a Third Party Adapter Service Component" on page 4-42.

6.5.2.2 Configuring an Inbound BPM Process Component

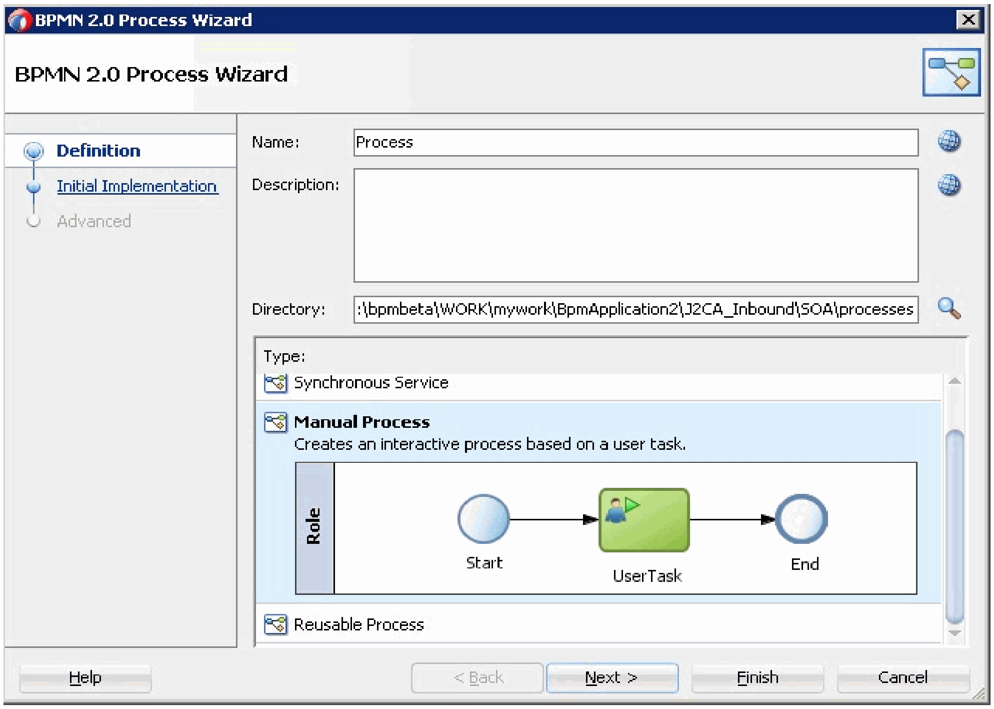
This section describes how to configure an inbound BPM process component.

Perform the following steps to configure an inbound BPM process component:

1. Drag and drop the **BPMN Process** component from the Components pane to the Components pane.

   The Create BPMN Process dialog is displayed, as shown in Figure 6–41.

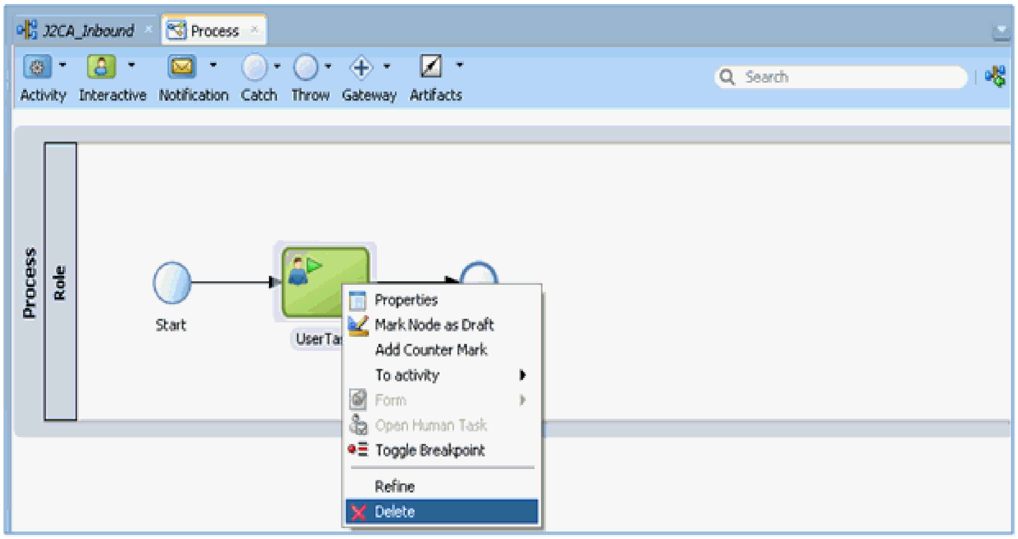
Figure 6–41 Create BPMN Process Dialog



2. Select **Manual Process** in the Type section.
3. Click **Finish**.

   The BPMN process is displayed, as shown in Figure 6–42.

Figure 6–42 BPMN Process

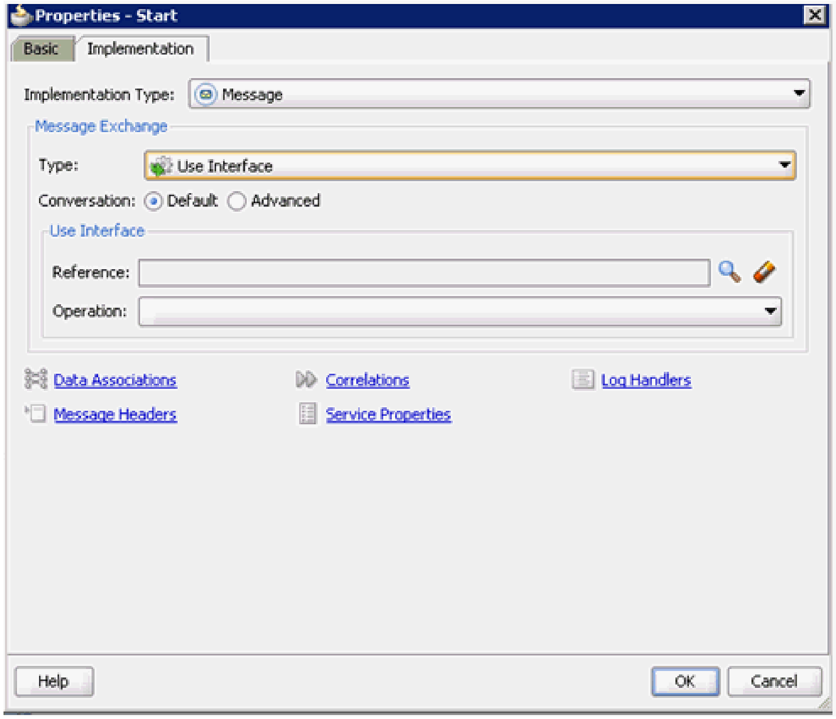


4. Right-click **UserTask** and select **Delete** from the menu.
5. Double-click the **Start** event component.

   The Properties - Start dialog is displayed.
6. Click the **Implementation** tab.

7. Select **Message** from the Implementation Type list.
8. Select **Use Interface** from the Message Exchange Type drop-down list.
9. Click the **Browse** icon to the right of the Reference field, as shown in Figure 6–43.

Figure 6–43 Browse Icon



Implementation Type: Message

Type: Use Interface

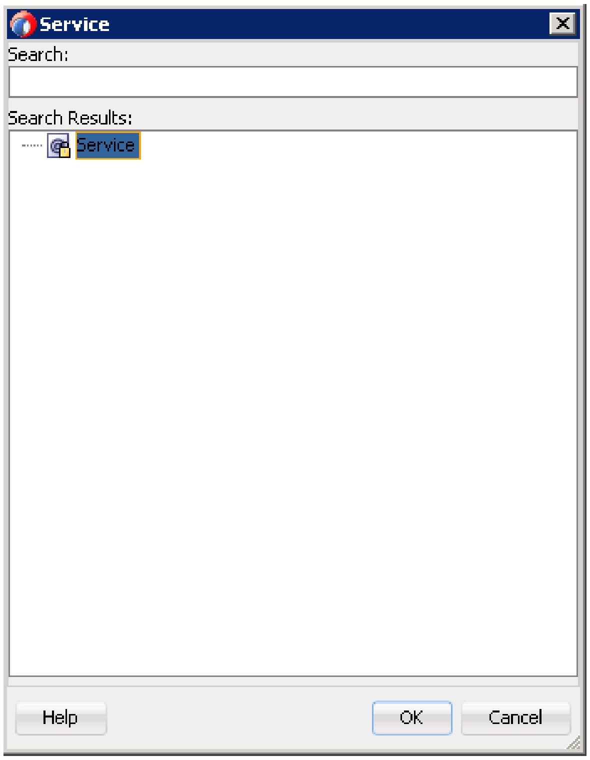
Conversation: ● Default ○ Advanced

Reference:

Operation:

The Service dialog is displayed, as shown in Figure 6–44.

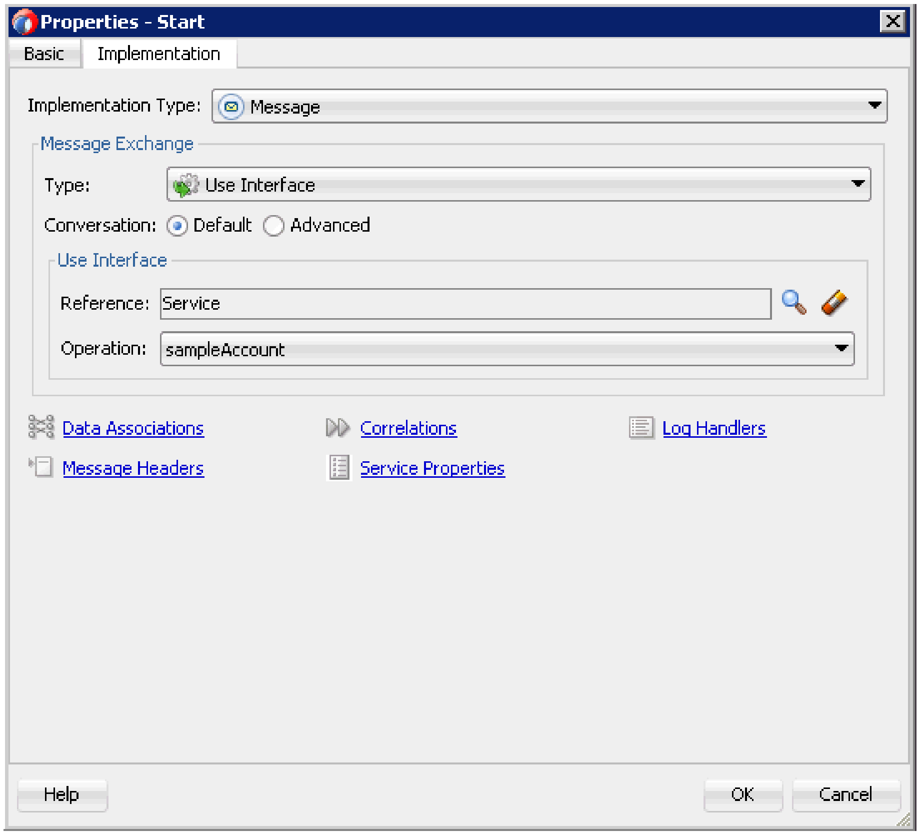
Figure 6–44 Service Dialog



10. Select the Third Party Service that has been created and click **OK**.

You are returned to the Properties - Start dialog, as shown in Figure 6–45.

Figure 6–45 Properties - Start Dialog



Properties - Start

Basic | Implementation

Implementation Type: Message

Message Exchange

Type: Use Interface

Conversation: ● Default ○ Advanced

Use Interface

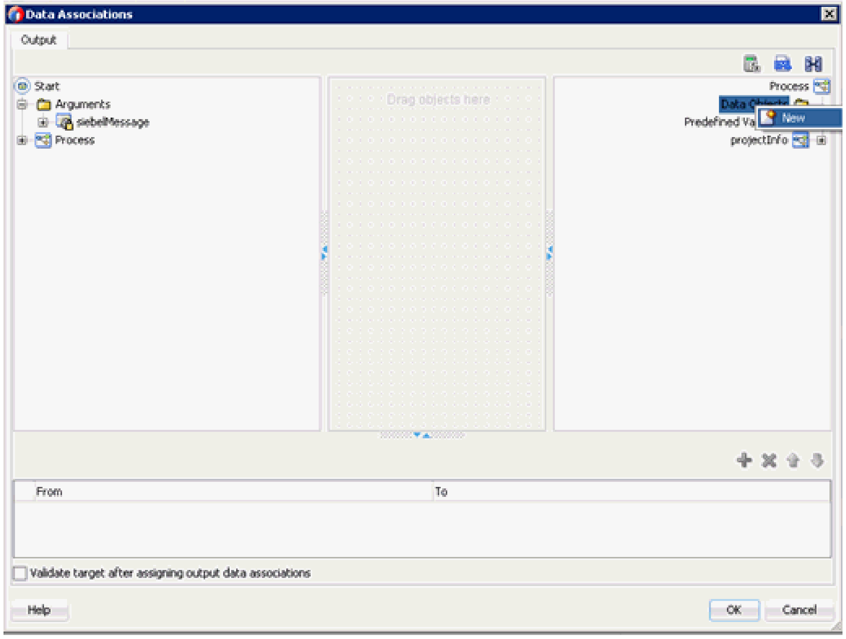
Reference: Service

Operation: sampleAccount

11. Click the **Data Associations** icon.

    The Data Associations dialog is displayed, as shown in Figure 6–46.

Figure 6–46 Data Associations Dialog



Data Associations

Output

Start
Arguments
siebelMessage
Process

Drag objects here

Process
Data Objects
New
Predefined Va
projectInfo

From	To

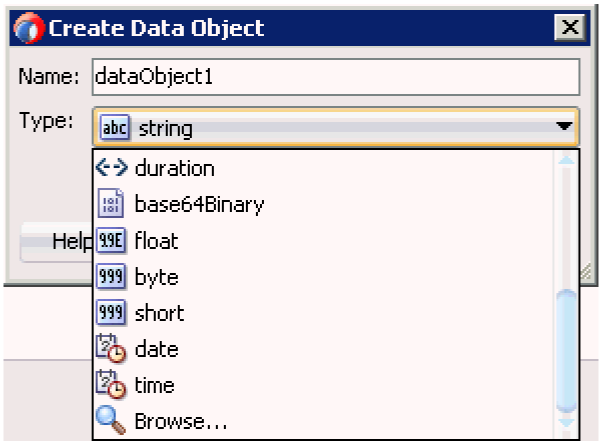
□ Validate target after assigning output data associations

12. Right-click the **Data Object** node in the right pane and select **New**.

    The Create Data Object dialog is displayed.

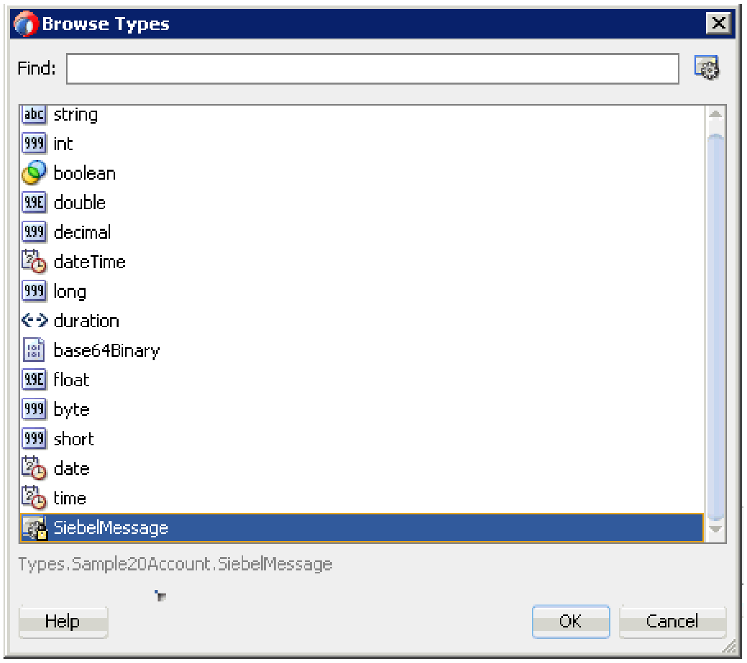
13. Enter a name in the Name field, and then click the drop-down button in the Type field and select **Browse** from the list, as shown in Figure 6–47.

Figure 6–47 Create Data Object Dialog



The Browse Types dialog is displayed, as shown in Figure 6–48.

Figure 6–48 Browse Types Dialog



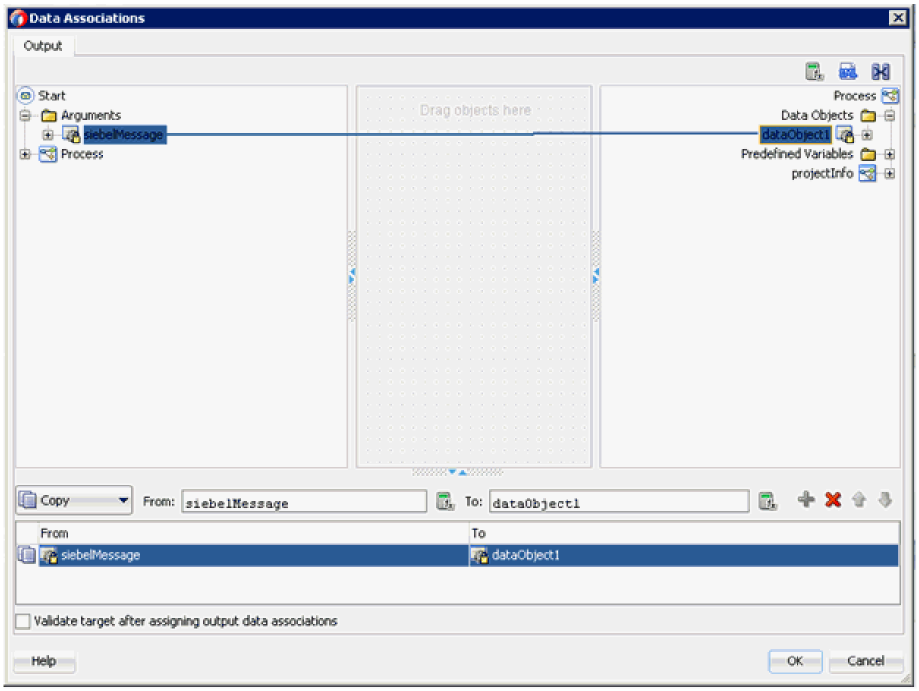
14. Select the component and click **OK**.

    You are returned to the Create Data Object dialog.

15. Click **OK**.

    The Data Object that has been created is displayed under the Data Objects node in the Data Associations dialog, as shown in Figure 6–49.

Figure 6–49 Data Associations Dialog



16. Select and drag the **siebelMessage** Argument under the Start node in the left pane and drag it to the Data Object in the right pane.

17. Click **OK**.

    You are returned to the Properties - Start dialog.

18. Click **OK**.

    You are returned to the Process workspace area.

19. Double-click the created project to load the components.

20. Click the **Save All** icon in the menu bar to save the new inbound BPM process component that was configured.

    You are now ready to create a File adapter for the write operation.

6.5.2.3 Creating a File Adapter for the Write Operation

This section describes how to create a File adapter for the write operation.

Perform the following steps to create a File adapter for the write operation:

1. Drag and drop the **File Adapter** component from the Technology Adapters pane to the External References pane.

   The Adapter Configuration Wizard is displayed.

2. Type a name for the new File adapter in the Name field and click **Next**.

   The Adapter Interface page is displayed.

3. Ensure that the **Define from operation and schema (specified later)** option is selected.

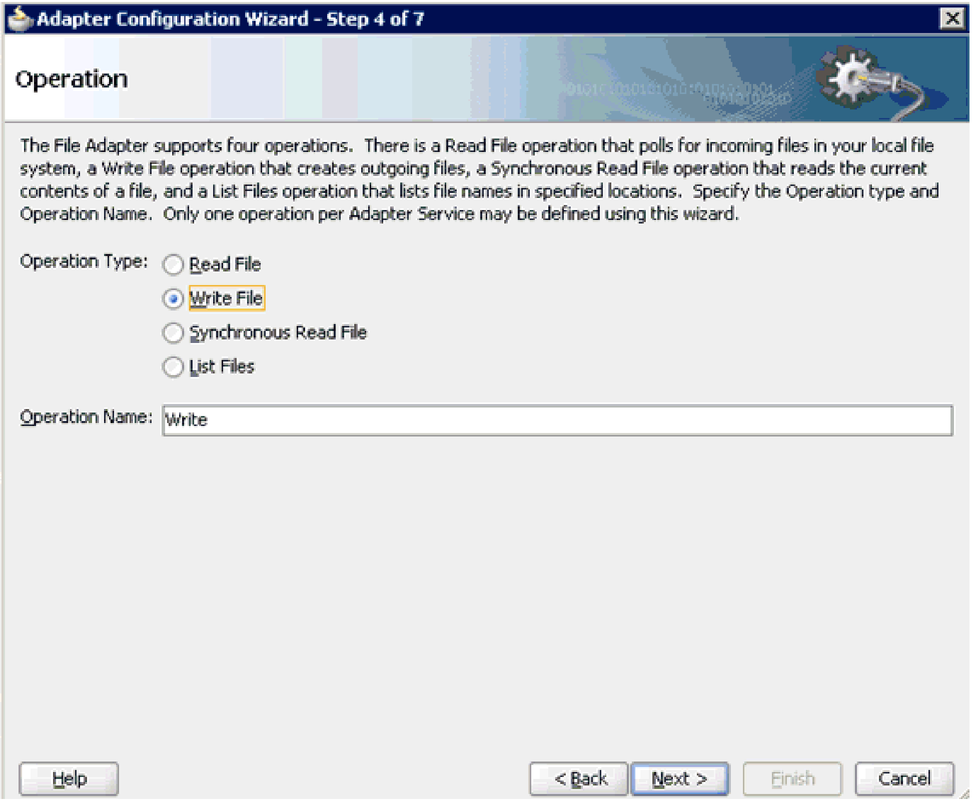
4. Click **Next**.

   The File Server Connection page is displayed.

5. Click **Next**.

   The Operation page is displayed, as shown in Figure 6–50.

Figure 6–50 Operation Page



Adapter Configuration Wizard - Step 4 of 7

Operation

The File Adapter supports four operations. There is a Read File operation that polls for incoming files in your local file system, a Write File operation that creates outgoing files, a Synchronous Read File operation that reads the current contents of a file, and a List Files operation that lists file names in specified locations. Specify the Operation type and Operation Name. Only one operation per Adapter Service may be defined using this wizard.

Operation Type: ○ Read File
● Write File
○ Synchronous Read File
○ List Files

Operation Name: Write

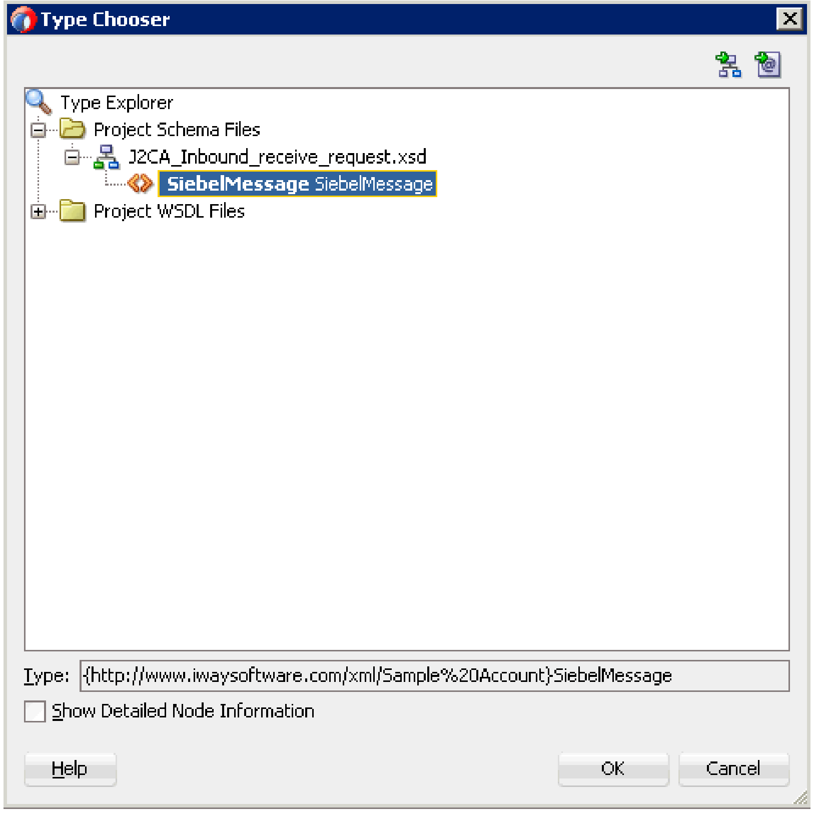
6. Select **Write File** from the list of Operation Type options and specify an Operation Name (for example, Write).
7. Click **Next**.

   The File Configuration page is displayed.
8. Specify a location on your file system where the output file is written.
9. In the File Naming Convention field, specify a name for the output file.
10. Click **Next**.

    The Messages page is displayed.
11. Click **Browse**, which is located to the right of the URL field.

    The Type Chooser dialog is displayed, as shown in Figure 6–51.

Figure 6–51 Type Chooser Dialog



Type: {http://www.iwaysoftware.com/xml/Sample%20Account}SiebelMessage

□ Show Detailed Node Information

12. Expand **Project Schema Files** and **J2CA_Inbound_receive_request.xsd**.
13. Select the available schema.
14. Click **OK**.

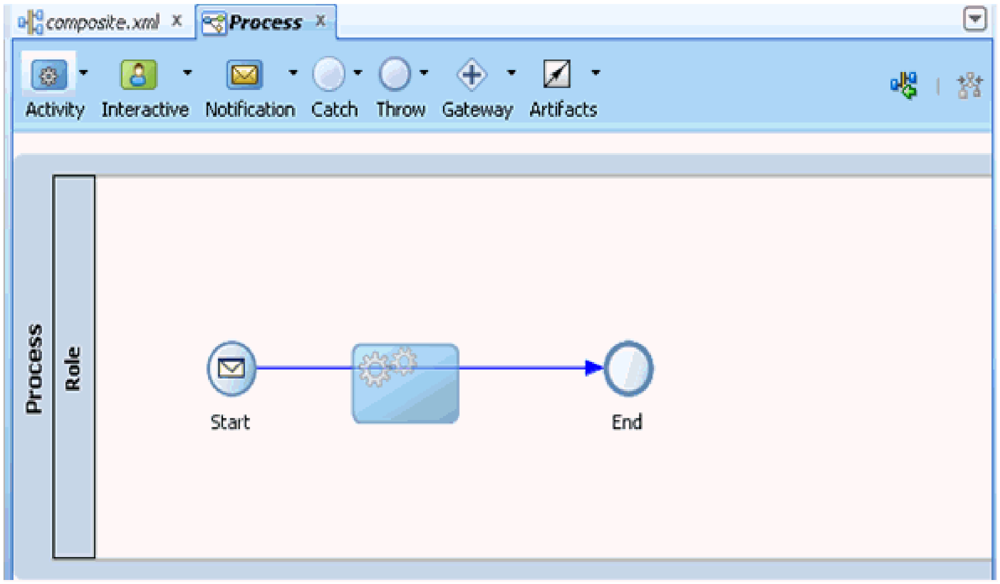
    You are returned to the Messages page.
15. Click **Next**.

    The Finish page is displayed.
16. Click **Finish**.

    The File Adapter service is created in the External References pane.
17. Double-click the BPMN Process component.

    The BPMN process is displayed.
18. Click the **Activity** icon, and select **Service**.
19. Drop the Service icon on the wire between the Start and End event components, as shown in Figure 6–52.

Figure 6–52 Activity Icon

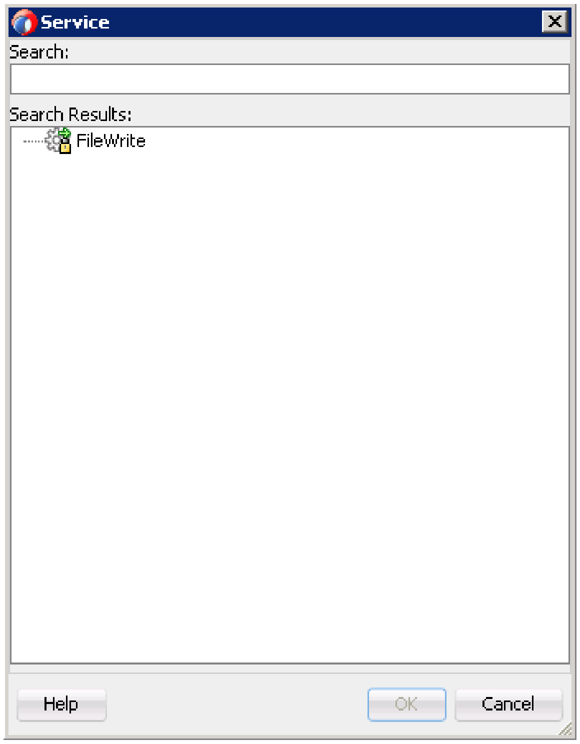


The Properties - ServiceTask dialog is displayed.

20. Click the **Implementation** tab.
21. Select **Service Task** from the Implementation Type list.
22. Select **Service Call** from the Message Exchange Type list.
23. Click the **Browse** icon to the right of the Service field.

    The Type dialog is displayed, as shown in Figure 6–53.

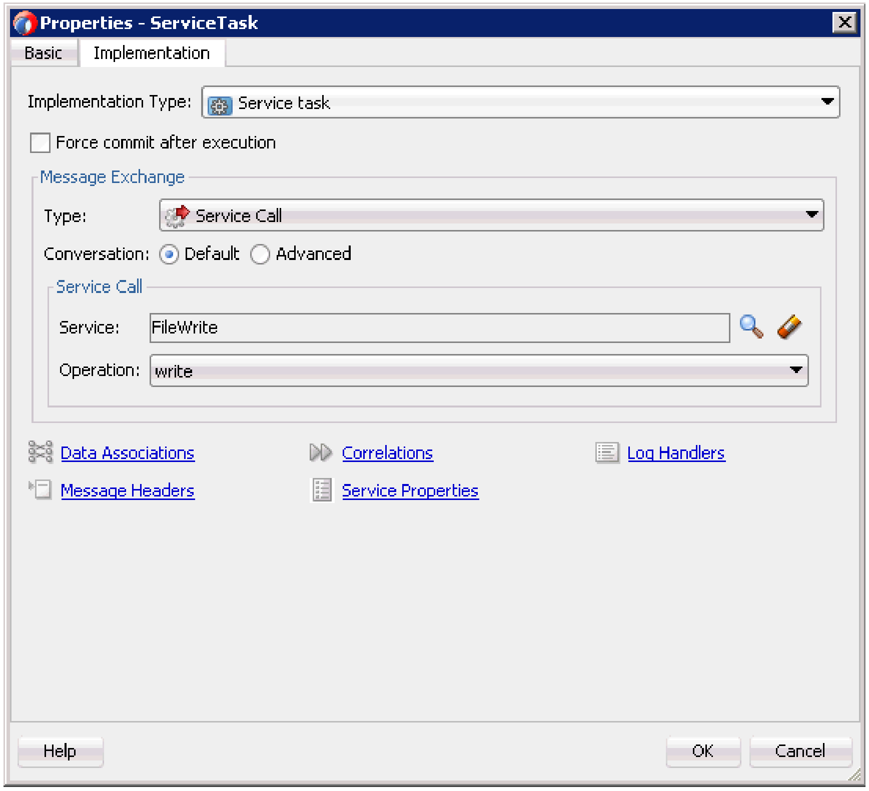
Figure 6–53 Type Dialog



24. Select the service for write operation that has been created and click **OK**.

    You are returned to the Properties - ServiceTask dialog, as shown in Figure 6–54.

Figure 6–54 Properties - ServiceTask Dialog



Properties - ServiceTask

Implementation Type: Service task

□ Force commit after execution

Message Exchange

Type: Service Call

Conversation: ● Default ○ Advanced

Service Call

Service: FileWrite

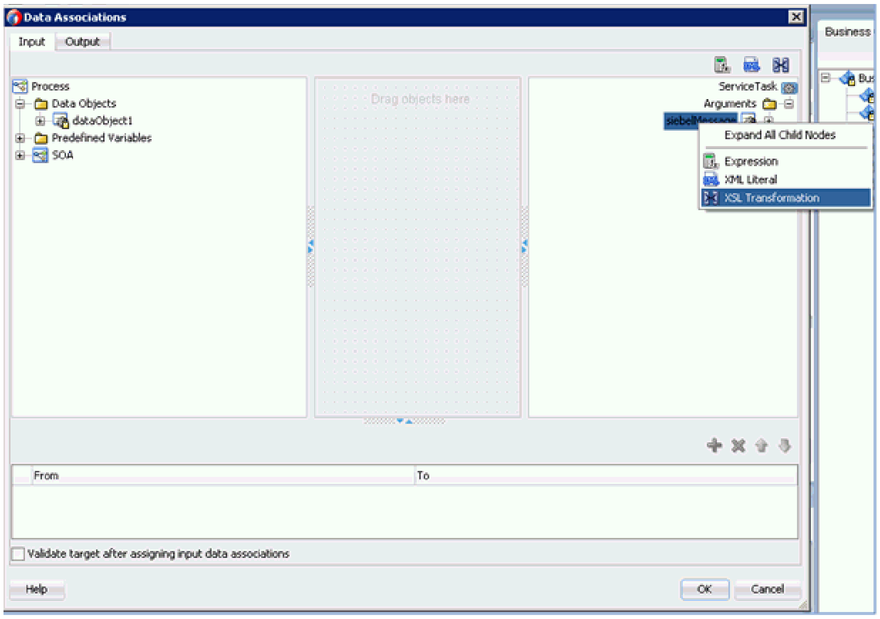
Operation: write

25. Click the **Data Associations** hyperlink.

    The Data Associations dialog is displayed.

26. Right-click the **siebelMessage** argument on the right pane and select **XSL Transformation**, as shown in Figure 6–55.

Figure 6–55 XSL Transformation



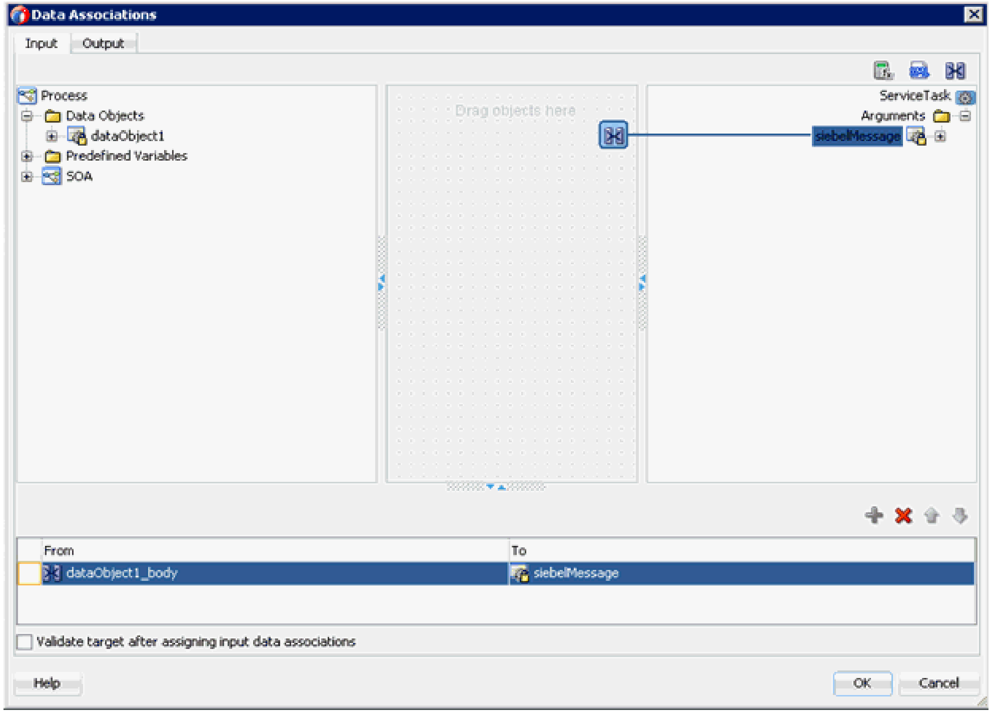
The Create Transformation dialog is displayed.

27. Select the created data object in the Sources area and click the right arrow icon so that the created data object is added to the Selected elements area.

28. Click **OK**.

    You are returned to the Data Associations dialog, as shown in Figure 6–56.

Figure 6–56 Data Associations Dialog



From	To
dataObject1_body	siebelMessage

29. Click **OK**.

You are returned to the Properties - ServiceTask dialog.

30. Click **OK**.

    The dataobject1_body.xsl tab is displayed.

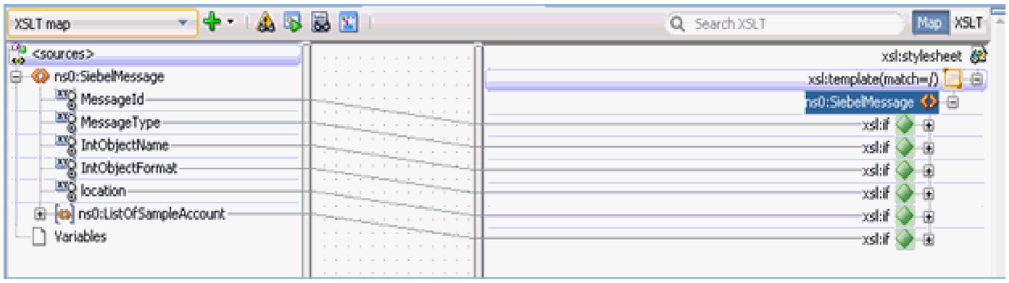
31. Automap the Source and Target elements.

    The Auto Map Preferences dialog is displayed.

32. Accept the default values and click **OK**.

    The transformation is completed, as shown in Figure 6–57.

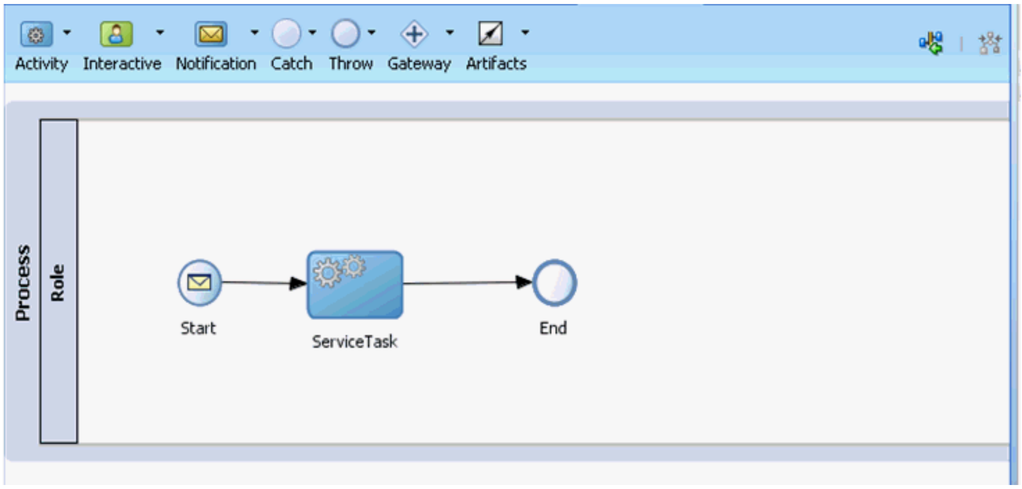
Figure 6–57 Completed Transformation



33. Save the transformation.

34. Return to the Process workspace area, as shown in Figure 6–58.

Figure 6–58 Process Workspace Area



The ServiceTask component is created between the Start event component and the End event component.

35. Click the **Save All** icon in the menu bar to save the new inbound BPM process component that was configured.

6.5.2.4 Adjusting for Known Deployment Issues With 12c

For more information on how to adjust for known deployment issues with 12c, see Section 4.4.3.3, "Adjusting for Known Deployment Issues With 12c" on page 4-26.

You are now ready to deploy the inbound BPM process. You can follow the same procedure that is described in Section 4.5.4, "Deploying the BPEL Inbound Process" on page 4-48. For more information on how to trigger events in Siebel, see Section 4.5.5.3, "Triggering an Event in Siebel 8.0 to Test Event Runtime Integration," on page 4-64.

6.6 Designing an Outbound BPM Process Using Transformations for Service Integration (BSE Configuration)

This section describes how to configure a BPM outbound process to your Siebel system, using a BPM project in Oracle JDeveloper.

A sample project has been provided for this outbound use case scenario in the following folder of the Application Adapters installation:

```
<ADAPTER_HOME>\etc\sample\SIEBEL_Samples.zip\SIEBEL_Samples\BPM\BSE\Outbound_
Project
```

The following tools are required to complete your outbound design-time configuration:

- Oracle Adapter Application Explorer (Application Explorer)
- Oracle JDeveloper BPM Designer (JDeveloper)

This section contains the following topics:

- Section 6.6.1, "Creating an Empty Composite for BPM"
- Section 6.6.2, "Defining a BPM Outbound Process"

Prerequisites

Before you design a BPM outbound process, you must generate the respective WSDL file using Application Explorer. For more information, see Section 4.6.1, "Generating a WSDL File for Request and Response Services Using a Web Service," on page 4-75.

6.6.1 Creating an Empty Composite for BPM

Perform the following steps to create an empty composite for SOA:

1. Create a new BPM application.
2. Enter a name for the BPM Application, and click **Next**.
3. Enter a name in the Project Name field, and click **Next**.
4. From the Composite Template list, select **Empty Composite** and click **Finish**.

   For more information, see Section 4.5.2, "Creating an Empty Composite for SOA" on page 4-41.

6.6.2 Defining a BPM Outbound Process

This section describes how to define a BPM outbound process. It contains the following topics:

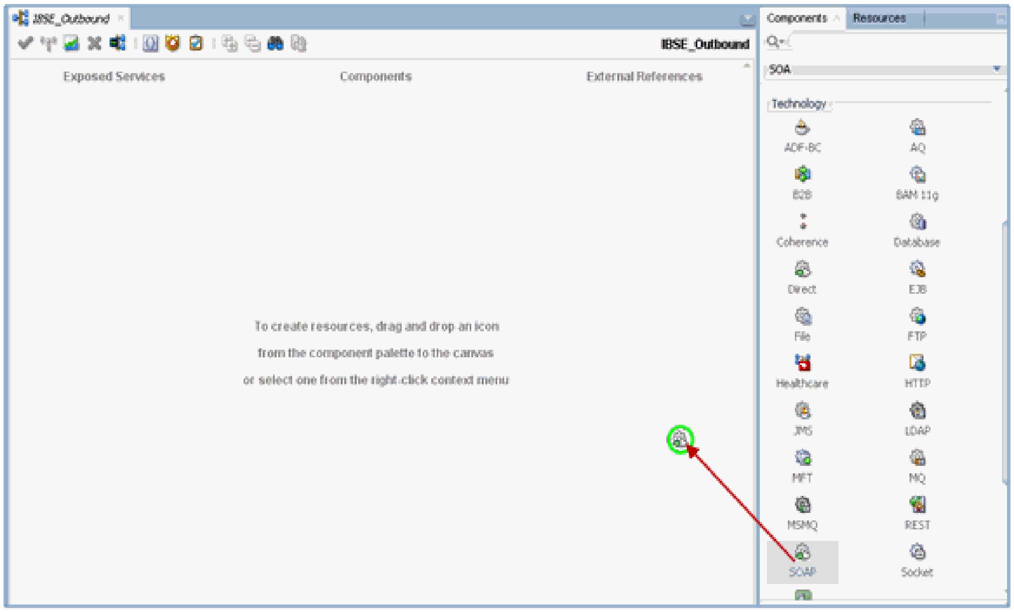
- Section 6.6.2.1, "Configuring a Web Service Component"
- Section 6.6.2.2, "Configuring a BPM Process Component"
- Section 6.6.2.3, "Creating a File Adapter for the Write Operation"

6.6.2.1 Configuring a Web Service Component

Perform the following steps to configure a Web Service component:

1. Double-click the created project to load the components.
2. Drag and drop the **Web Service** node from the Technology Adapters pane to the External References pane, as shown in Figure 6–59.

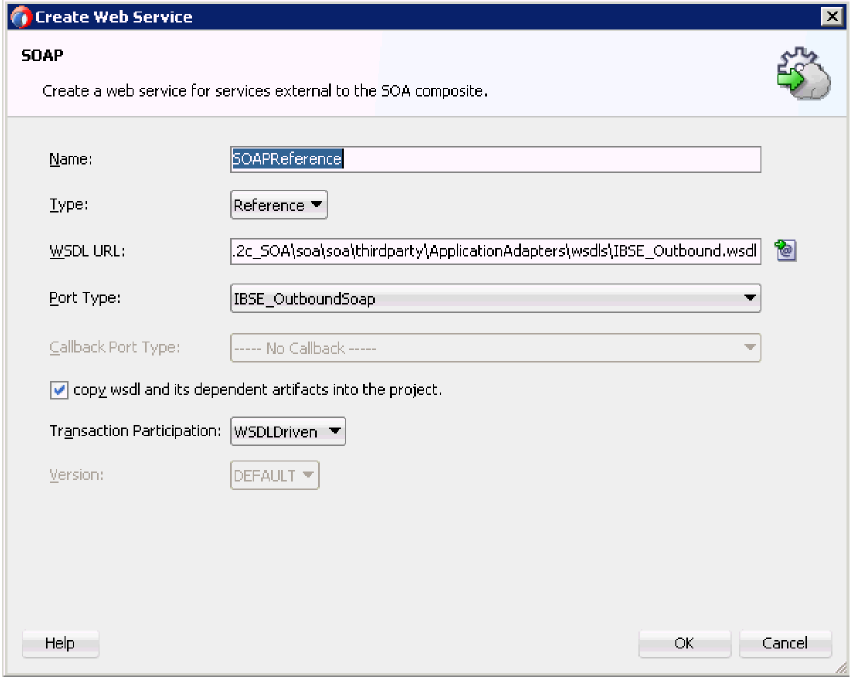
Figure 6–59 Web Service Node



To create resources, drag and drop an icon
from the component palette to the canvas
or select one from the right-click context menu

3. Enter an appropriate name for the Web Service and click on the **Find existing WSDLs** icon, which is located to the right of the WSDL URL field.
4. In the displayed WSDL Chooser window, navigate to the location where the WSDL is exported from the Application Explorer, and select the WSDL.
5. Click **OK**.
6. In the Web Service pane, click **OK**, as shown in Figure 6–60.

Figure 6–60 Web Service Pane



SOAP

Create a web service for services external to the SOA composite.

Name: SOAPReference

Type: Reference

WSDL URL: .2c_SOA\soa\soa\thirdparty\ApplicationAdapters\wsdls\IBSE_Outbound.wsdl

Port Type: IBSE_OutboundSoap

Callback Port Type: ----- No Callback -----

☑ copy wsdl and its dependent artifacts into the project.

Transaction Participation: WSDLDriven

Version: DEFAULT

7. In the displayed Localize Files window, click **OK**.

   This will import the WSDL file to the project folder

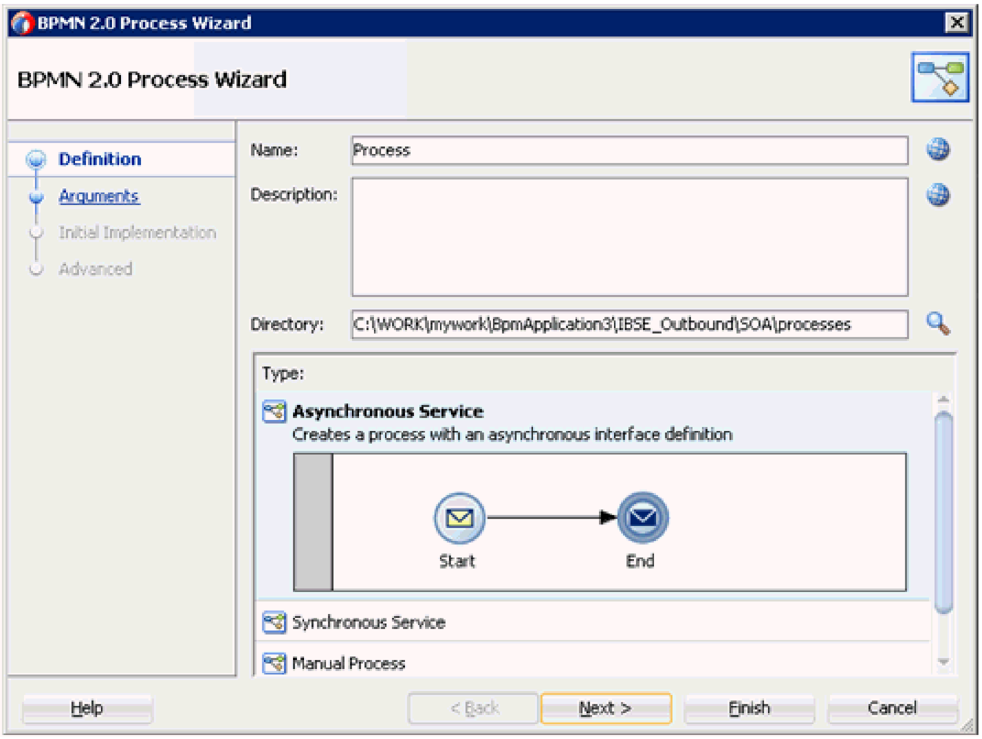
6.6.2.2 Configuring a BPM Process Component

This section describes how to configure an outbound BPM process component.

Perform the following steps to configure a BPM Component:

1. Drag and drop the **BPMN Process** component from the Components pane in to the Components pane.
2. Accept the default option that is selected under the Type area (Asynchronous Service) and click **Finish**, as shown in Figure 6–61.

Figure 6–61 Type Area



Name: Process

Description:

Directory: C:\WORK\mywork\BpmApplication3\IBSE_Outbound\SOA\processes

Type:

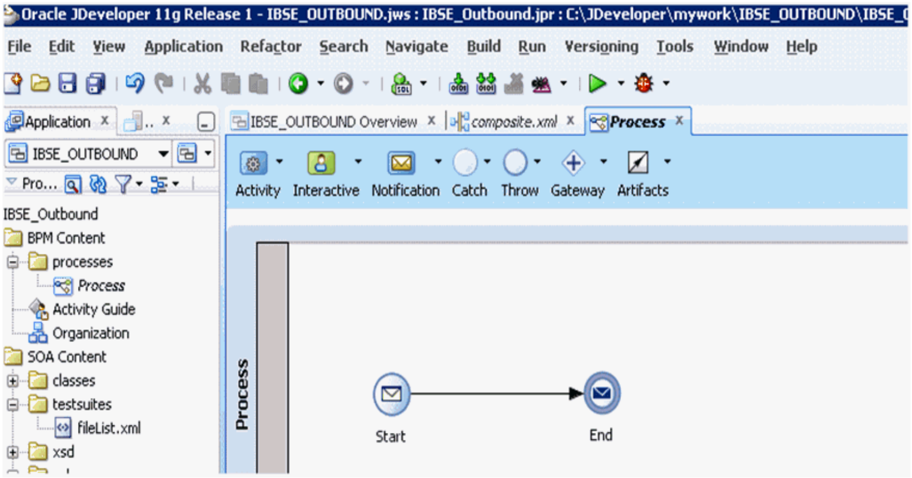
Asynchronous Service
Creates a process with an asynchronous interface definition

Synchronous Service

Manual Process

3. Double click on the Start Event component, as shown in Figure 6–62.

Figure 6–62 Start Event Component



4. In the displayed Properties-start window, click the **Implementation** tab.

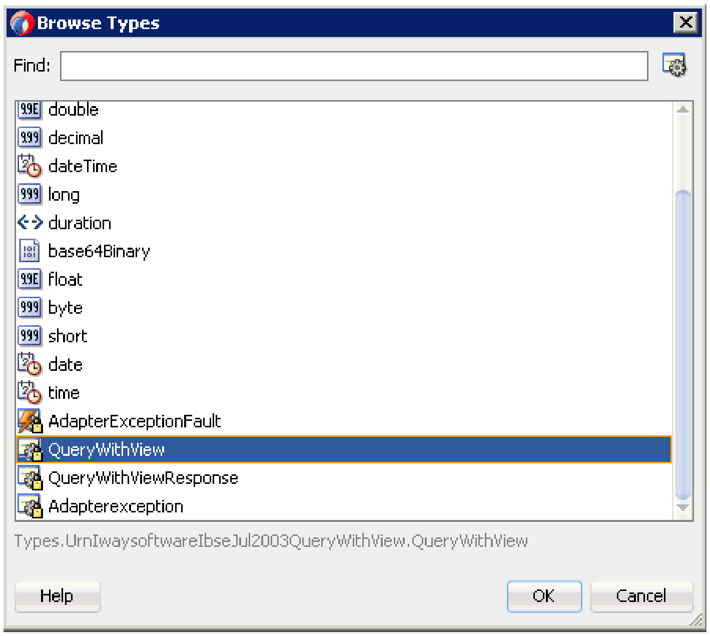
5. Click the Plus (+) icon to the right of the Arguments Definition field.

   The Edit Argument window is displayed.

6. Enter a name in the Name field, and then click the Type drop-down list and select **Browse**.

7. Select the **Request** component (for example, queryWithView), and click **OK**, as shown in Figure 6–63.

Figure 6–63 Request Component



Types.UrnIwaysoftwareIbseJul2003QueryWithView.QueryWithView

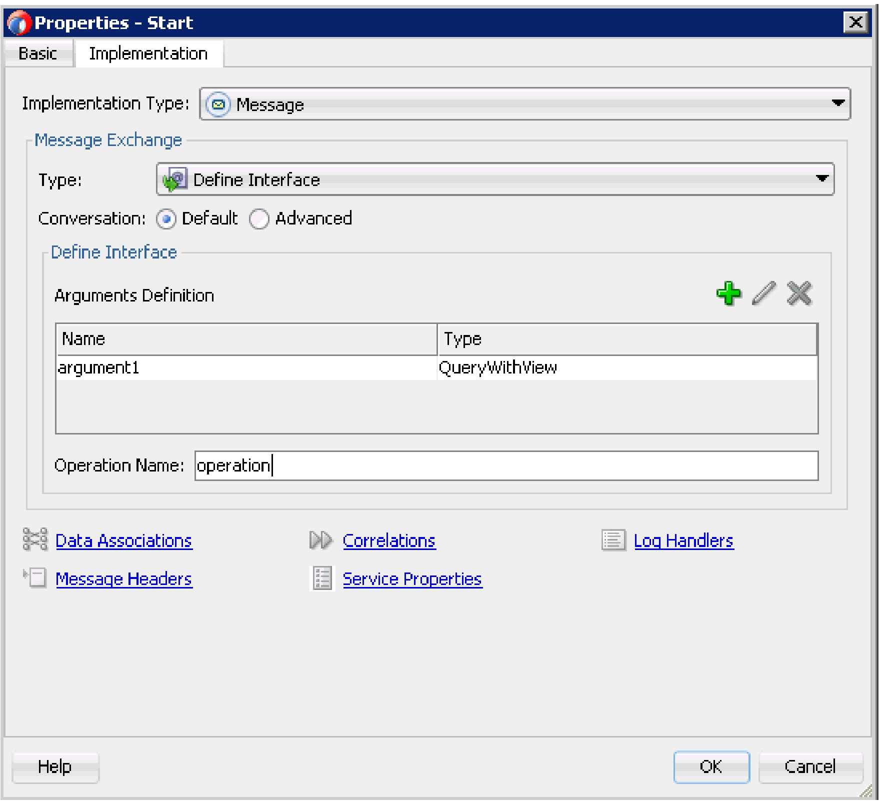
8. In the Edit Argument window that is displayed, click **OK**.

   The Properties - Start window is displayed.

9. In the Operation Name field, change the default entry from **start** to **operation**.

10. Click the **Data Associations** hyperlink, as shown in Figure 6–64.

Figure 6–64 Properties - Start Window



Implementation Type: Message

Type: Define Interface

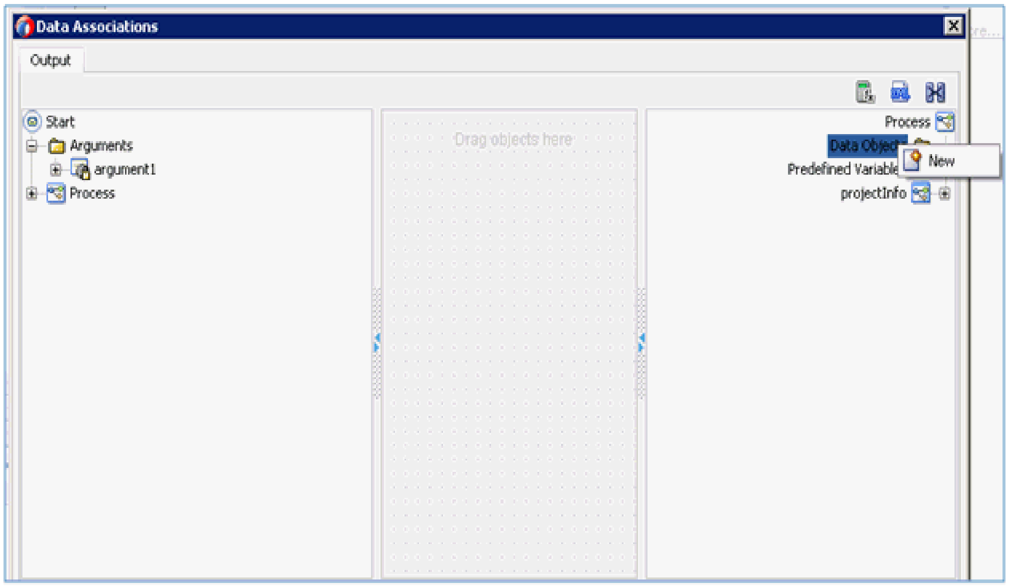
Conversation: ● Default ○ Advanced

Name	Type
argument1	QueryWithView

Operation Name: operation

11. Right-click the **Data Objects** node in the right pane, under Process, and select **New**, as shown in Figure 6–65.

Figure 6–65 Data Objects Node

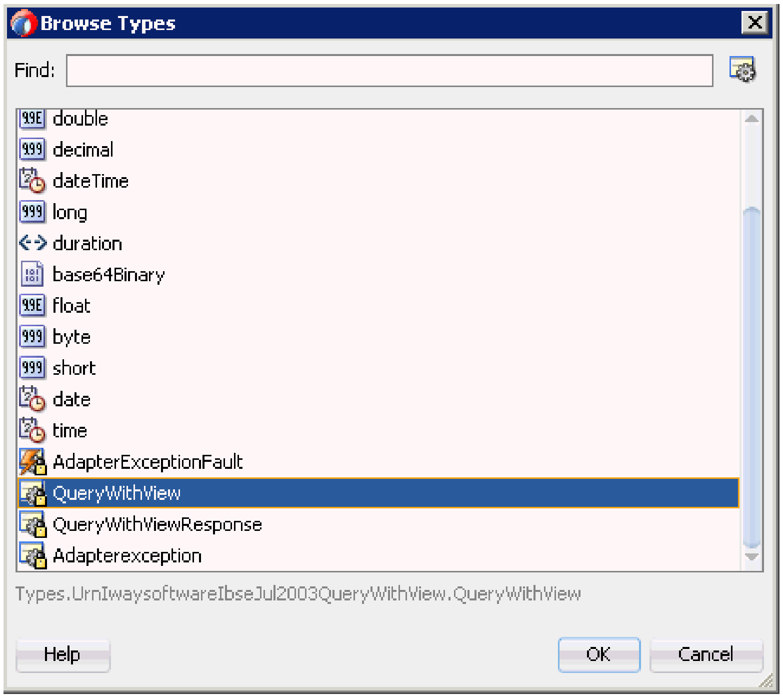


The Create Data Object window is displayed.

12. Enter a name in the Name field, click the Type drop-down list, and select **Browse**.

13. Select the **Request** component (for example, queryWithView) and click **OK**, as shown in Figure 6–66.

Figure 6–66 Request Component



Types.UrnIwaysoftwareIbseJul2003QueryWithView.QueryWithView

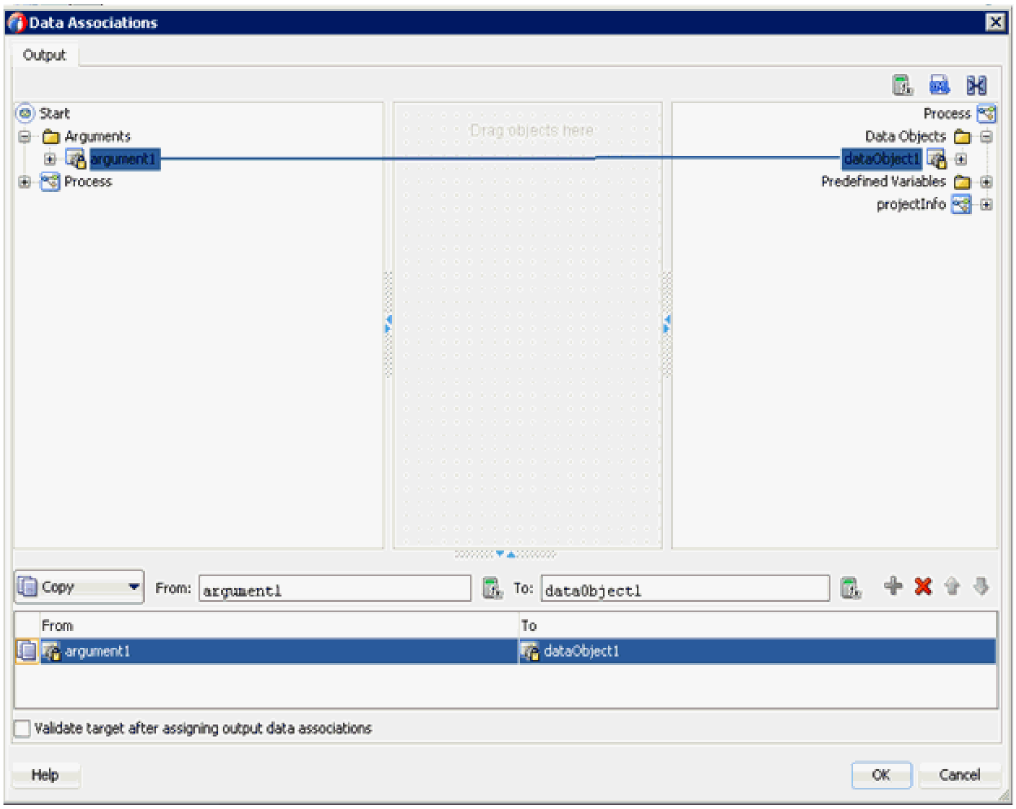
14. In the Create Data Object window, click **OK**.

    The Data Associations window is displayed.

15. Select **argument1** under the Arguments node in the left pane and drag and connect it to **dataObject1**, under Data Objects, in the right pane.

16. Click **OK**, as shown in Figure 6–67.

Figure 6–67 Data Associations



Copy | From: argument1 | To: dataObject1

From	To
argument1	dataObject1

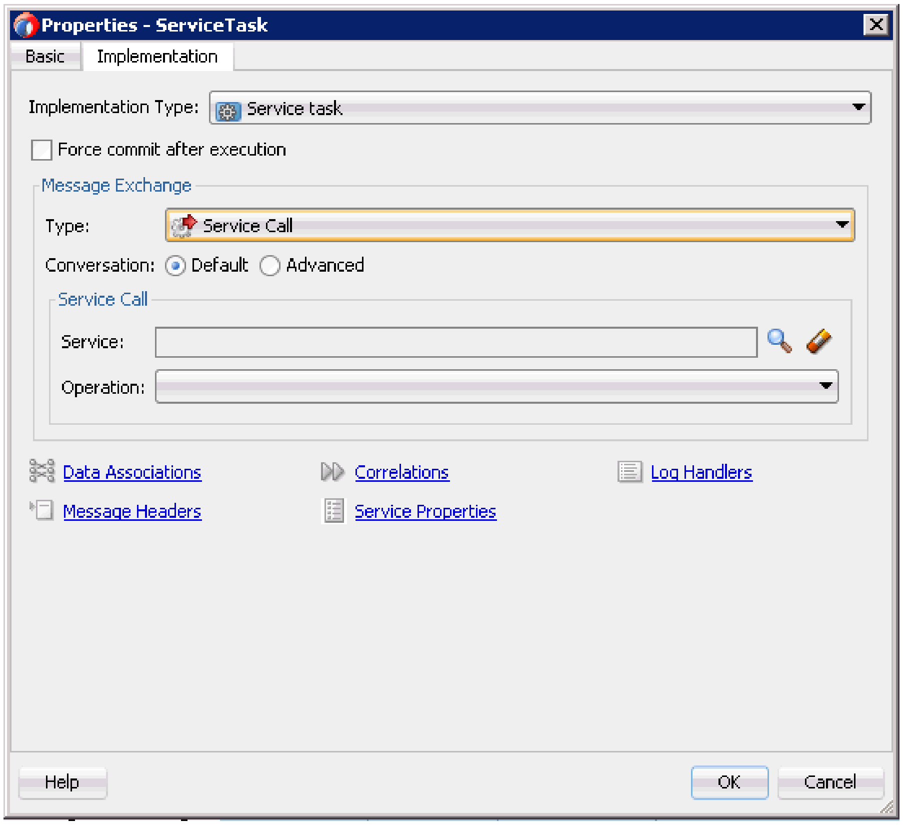
☐ Validate target after assigning output data associations

17. In the Properties - Start window that is displayed, click **OK**.

    You are returned to the Process workspace area.

18. Click the **Activity** drop-down menu and select **Service**.
19. Drop the **Service** icon on the wire between the **Start** and **End** event components.
20. In the displayed Properties - ServiceTask window, click the **Implementation** tab.
21. Select **Service Call** from the Message Exchange Type list.
22. Click the **Browse** icon to the right of the Service field, as shown in Figure 6–68.

Figure 6–68 Browse Icon



Implementation Type: Service task

□ Force commit after execution

Type: Service Call

Conversation: ● Default ○ Advanced

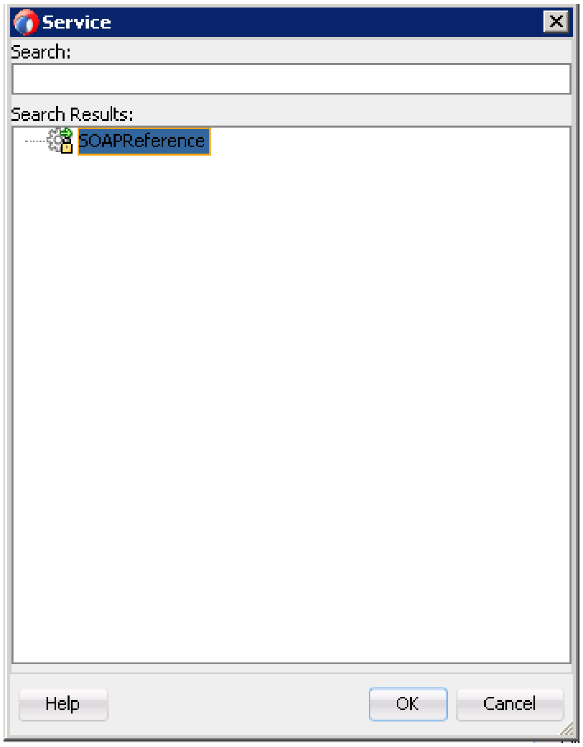
Service:

Operation:

The Service window is displayed.

23. Select the Web Service that has been created and click **OK**, as shown in Figure 6–69.

Figure 6–69 Created Web Service

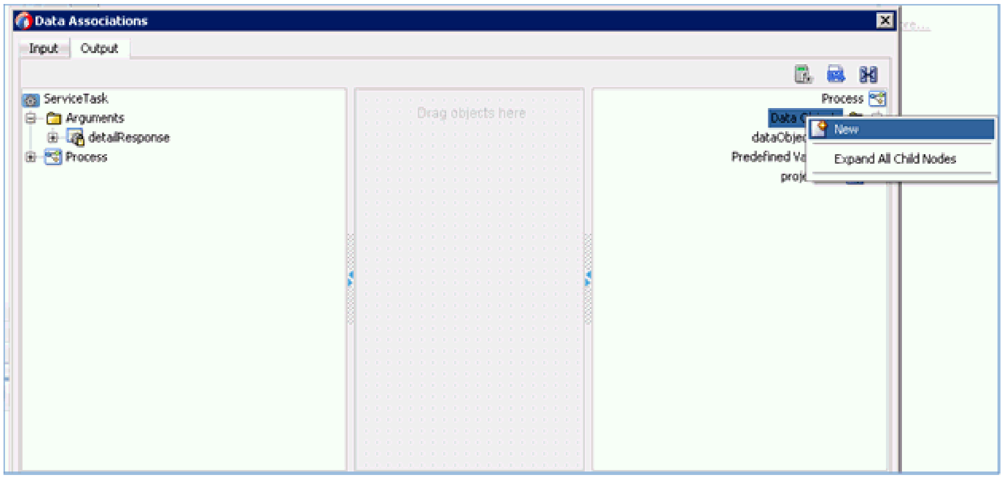


24. In the Properties - ServiceTask window that is displayed, click the **Data Associations** hyperlink.

    The Data Associations window is displayed.

25. Create response Data Object by right-clicking the **Data Objects** node in the right pane of the Output tab and selecting **New**, as shown in Figure 6–70.

Figure 6–70 Data Objects Node

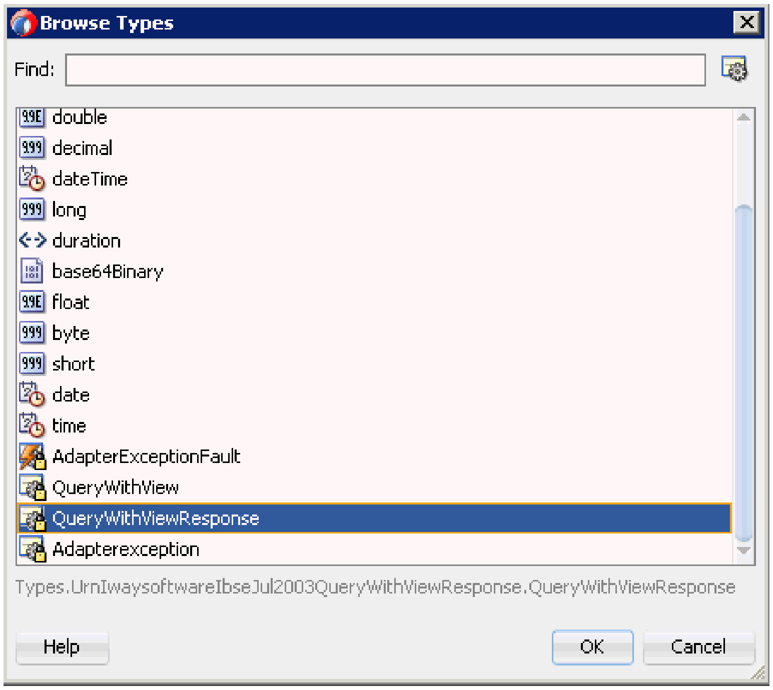


The Create Data Object window is displayed.

26. Enter a name in the Name field, click the Type drop-down list, and select **Browse**.

27. Select the Response component (for example, QueryWithViewResponse) and click **OK**, as shown in Figure 6–71.

Figure 6–71 Response Component (QueryWithViewResponse)



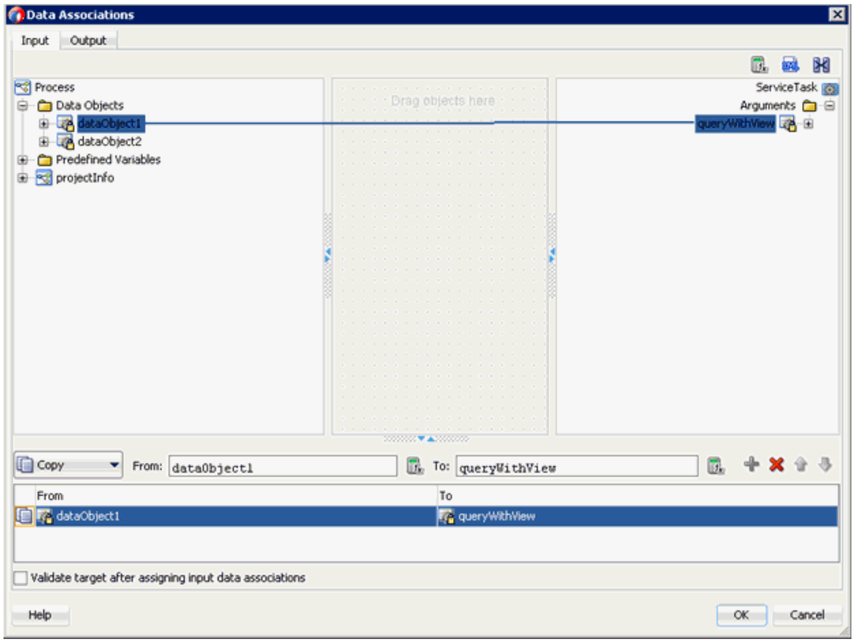
Types.UrnIwaysoftwareIbseJul2003QueryWithViewResponse.QueryWithViewResponse

28. In the Create Data Object window, click **OK**.

    The Data Associations window is displayed.

29. Select **dataObject1**, under the Data Objects node in the left pane of the Input tab, and drag and connect it to the **queryWithView** node, under the Arguments node in the right pane, as shown in Figure 6–72.

Figure 6–72 Data Associations



From: dataObject1
To: queryWithView

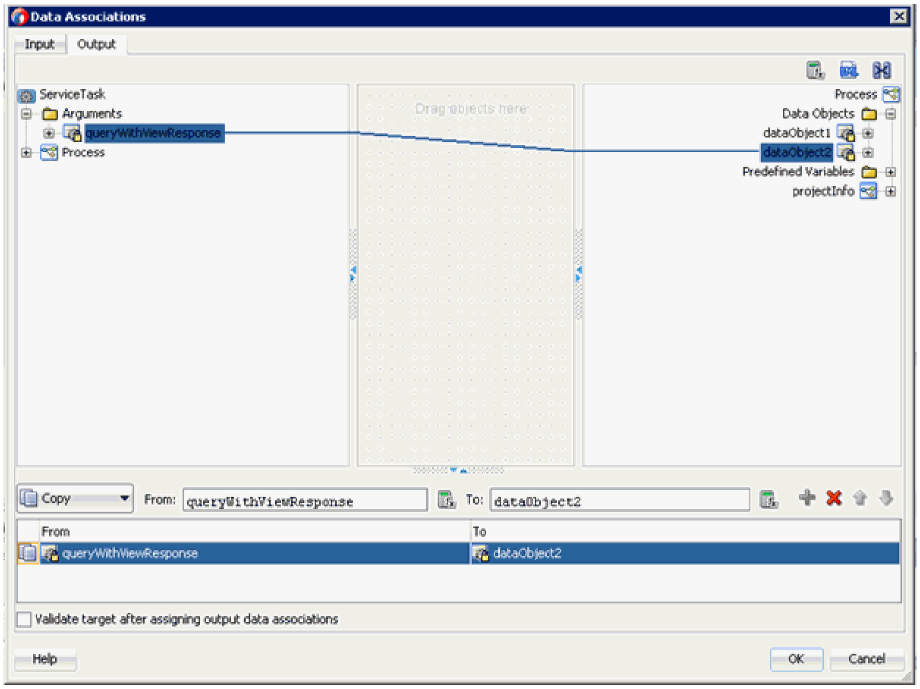
From	To
dataObject1	queryWithView

□ Validate target after assigning input data associations

30. Click on the **Output** tab and select **queryWithViewResponse** under the Arguments node in the left pane and drag and connect it to **dataObject2** under the Data Objects node.

31. Click **OK**, as shown in Figure 6–73.

Figure 6–73 Output Tab



From: queryWithViewResponse To: dataObject2

From	To
queryWithViewResponse	dataObject2

□ Validate target after assigning output data associations

32. In the Properties - ServiceTask window that is displayed, click **OK**.

33. Click the **Save All** icon in the menu bar to save the new outbound BPM process component that was configured.

34. Double-click the **composite.xml** node in the left pane.

6.6.2.3 Creating a File Adapter for the Write Operation

This section describes how to create a File adapter for the write operation.

Perform the following steps to create a File adapter for the write operation:

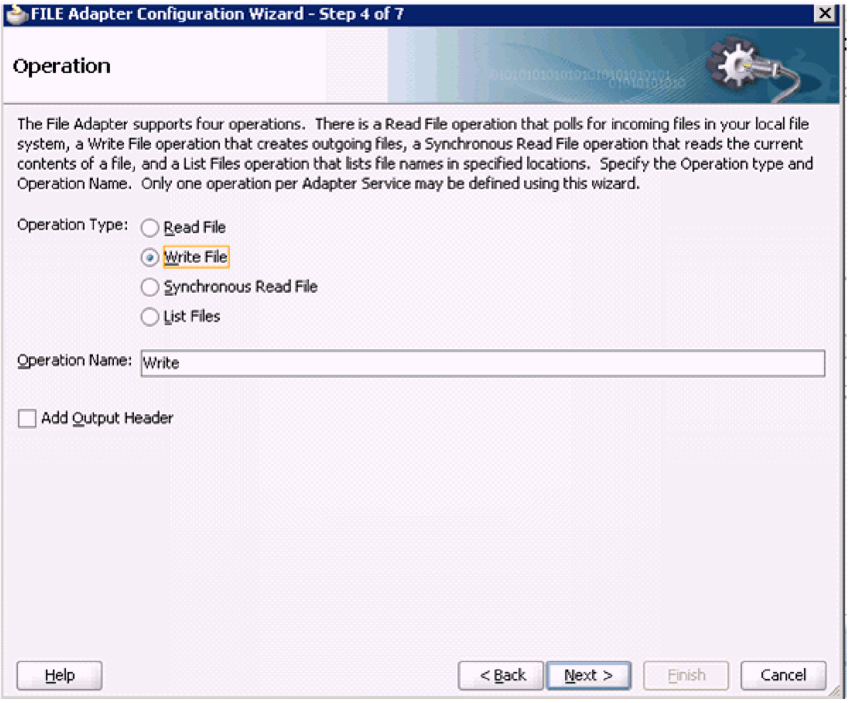
1. Drag and drop the **File Adapter** component from the Technology Adapters pane to the External References pane, and provide a name for the File Adapter.

2. In the Adapter Interface pane that is displayed, ensure that the **Define from operation and schema (specified later)** option is selected, and click **Next**.

3. Click **Next**.

4. In the Operation pane that is displayed, select **Write File** from the list of Operation Type options, and click **Next**, as shown in Figure 6–74.

Figure 6–74 Operation Pane



FILE Adapter Configuration Wizard - Step 4 of 7

Operation

The File Adapter supports four operations. There is a Read File operation that polls for incoming files in your local file system, a Write File operation that creates outgoing files, a Synchronous Read File operation that reads the current contents of a file, and a List Files operation that lists file names in specified locations. Specify the Operation type and Operation Name. Only one operation per Adapter Service may be defined using this wizard.

Operation Type: ○ Read File
● Write File
○ Synchronous Read File
○ List Files

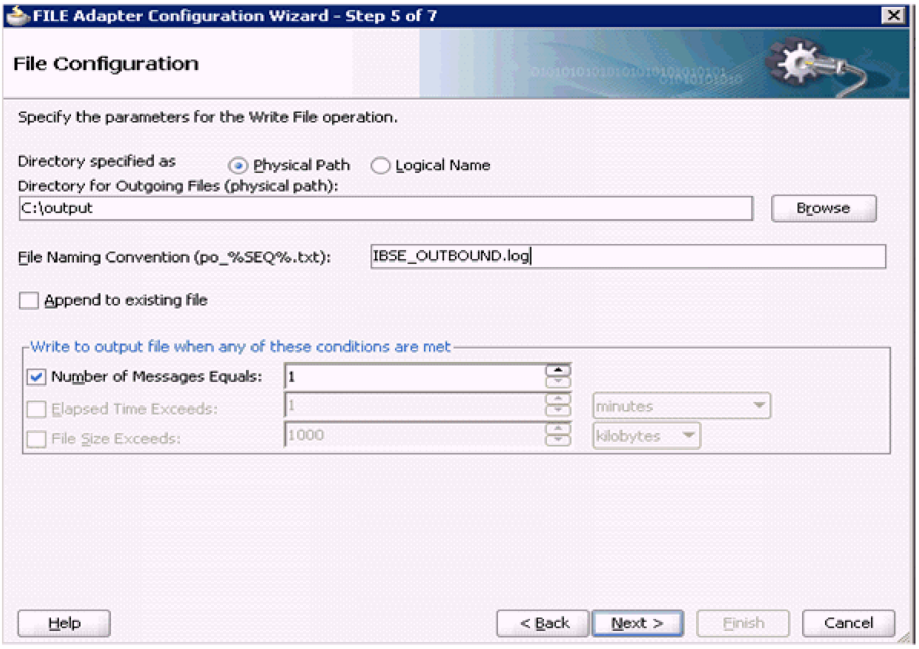
Operation Name: Write

□ Add Output Header

The File Configuration pane is displayed.

5. In the Directory for Outgoing Files (physical path) field, specify a location on your file system where the output file is written.
6. In the File Naming Convention field, specify a name for the output file.
7. Click **Next**, as shown in Figure 6–75.

Figure 6–75 File Configuration Pane



FILE Adapter Configuration Wizard - Step 5 of 7

File Configuration

Specify the parameters for the Write File operation.

Directory specified as ● Physical Path ○ Logical Name

Directory for Outgoing Files (physical path): C:\output

File Naming Convention (po_%SEQ%.txt): IBSE_OUTBOUND.log

□ Append to existing file

Write to output file when any of these conditions are met

☑ Number of Messages Equals: 1

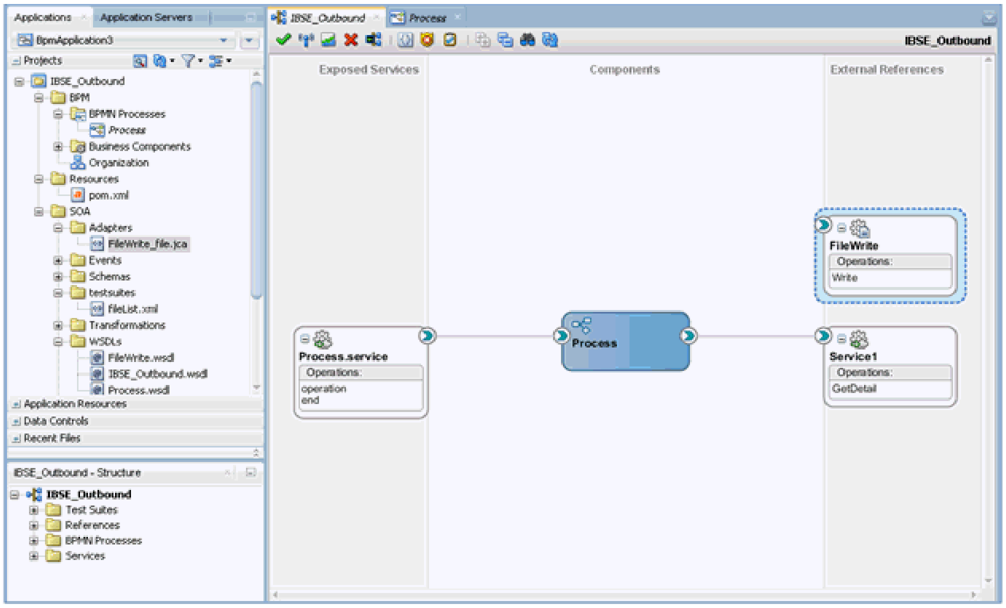
□ Elapsed Time Exceeds: 1 minutes

□ File Size Exceeds: 1000 kilobytes

The Messages pane is displayed.

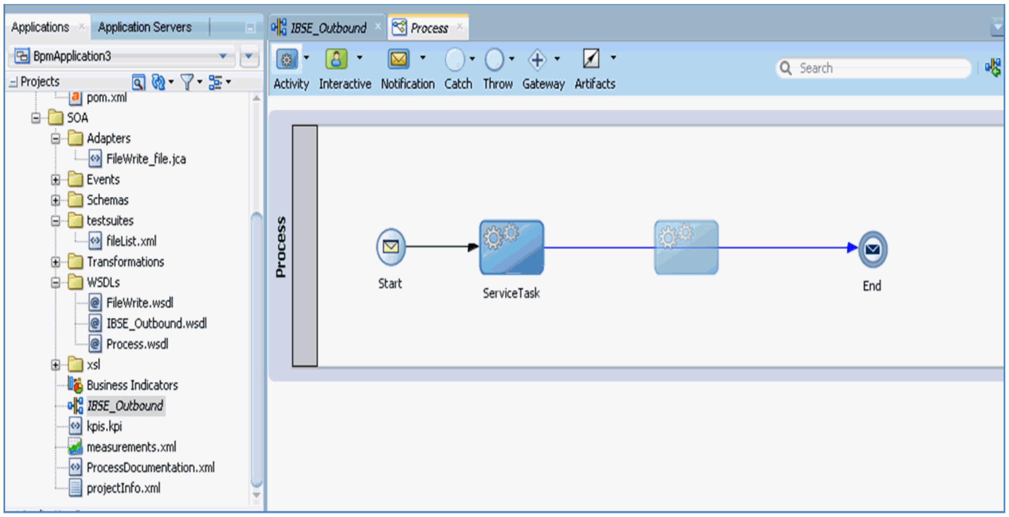
8. Click the **Browse**, which is located to the right of the URL field.
9. In the displayed Type Chooser window, expand **Project WSDL Files**, **IBSE_Outbound.wsdl**, **Inline Schemas** and then select **SiebelResponse**.
10. Click **OK**.
11. In the Messages pane, click **Next**.
12. In the Finish pane that is displayed, click **Finish**.
13. Double-click the **BPMN Process** component, as shown in Figure 6–76.

Figure 6–76 Composite.xml Tab



14. Click the **Activity** icon.
15. Drop the **Activity** icon on the wire between the **Service Task** and **End** event components, as shown in Figure 6–77.

Figure 6–77 Activity Icon



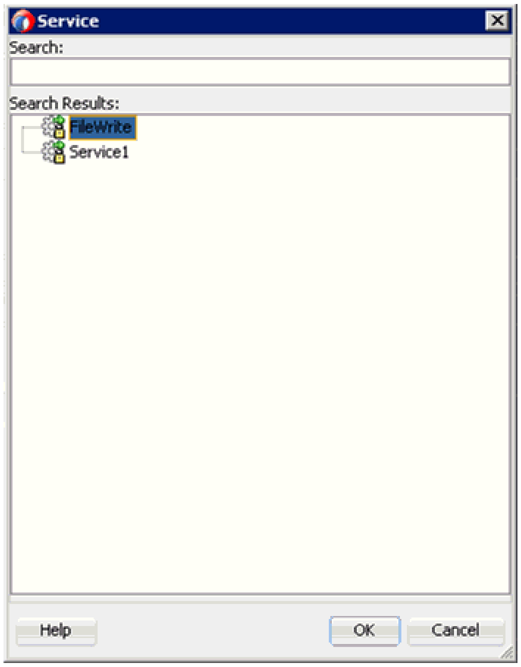
16. In the displayed Properties-ServiceTask1 window, click the **Implementation** tab

17. Select **Service Call** from the Type drop-down list in the Message Exchange section.

18. Click the **Browse** icon to the right of the Service field.

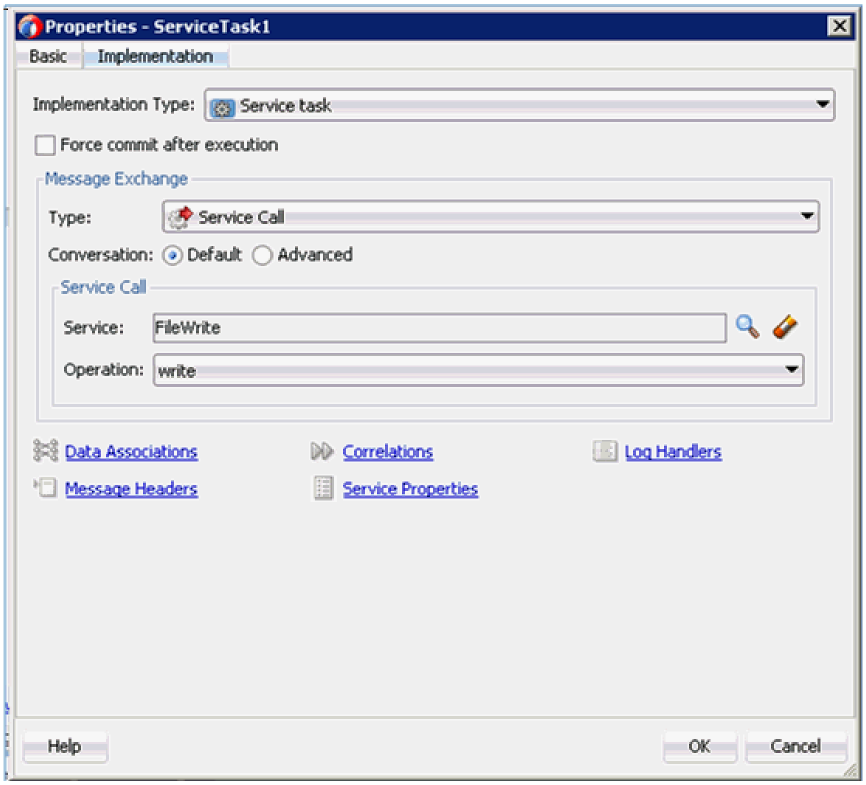
19. Select the service for write operation that has been created and click **OK**, as shown in Figure 6–78.

Figure 6–78 Service Window



20. In the Properties - ServiceTask1 window, click the **Data Associations** hyperlink, as shown in Figure 6–79.

Figure 6–79 Data Associations



Properties - ServiceTask1

Implementation Type: Service task

□ Force commit after execution

Message Exchange

Type: Service Call

Conversation: ● Default ○ Advanced

Service Call

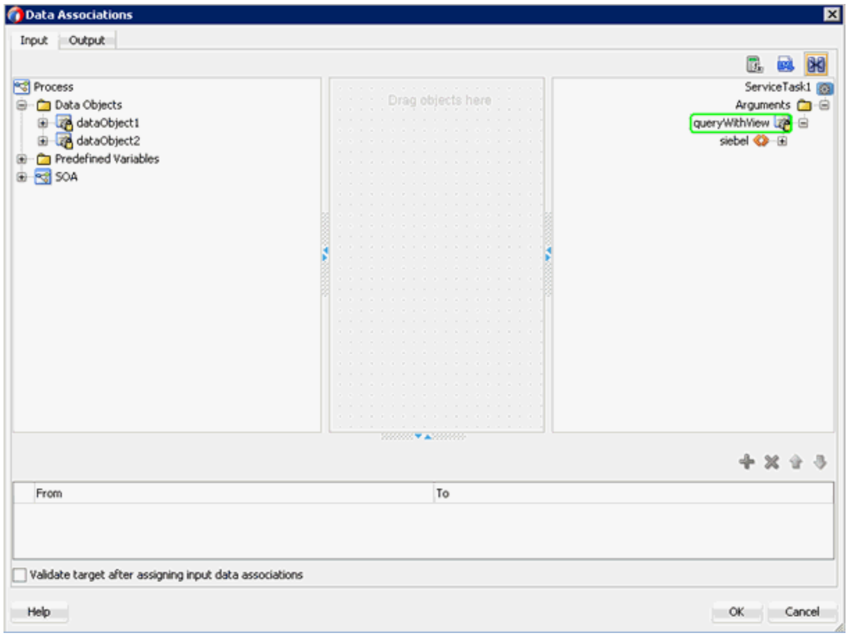
Service: FileWrite

Operation: write

21. In the Input tab, click the **XSL Transformation** icon in the top right pane.

22. Drag and drop the **XSL Transformation** icon to the **queryWithViewResponse** node, as shown in Figure 6–80.

Figure 6–80 QueryWithView Node



Data Associations

From	To

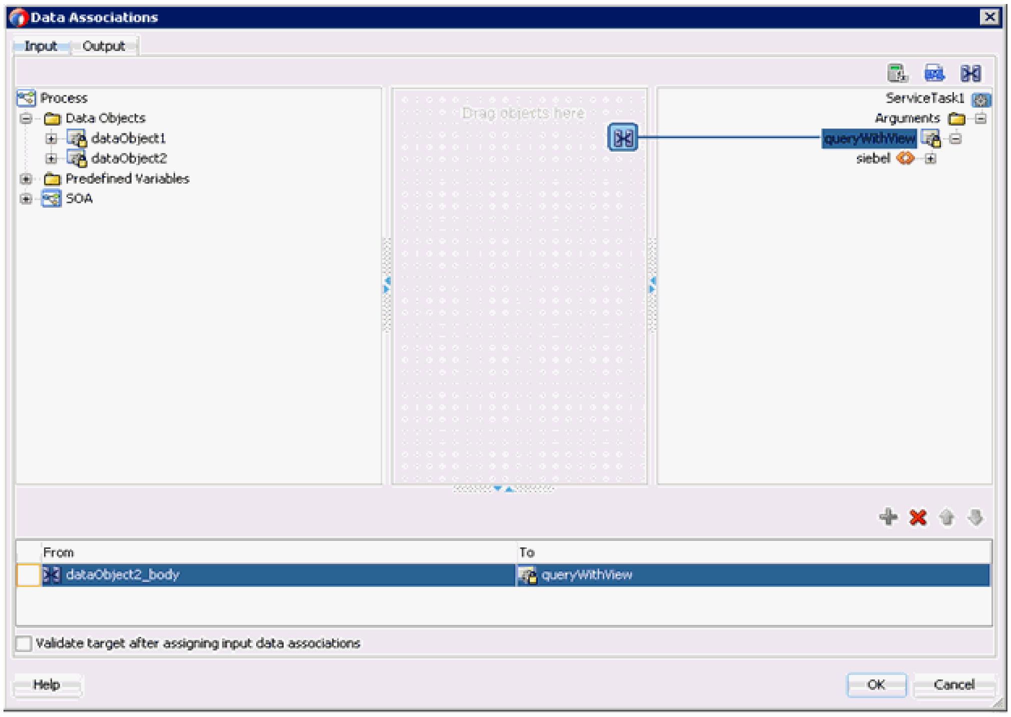
□ Validate target after assigning input data associations

23. In the displayed Create Transformation window, select **dataObject2** in the Sources section and click the right arrow symbol.

24. Accept the default value selected in the Target drop-down list and the default name in the Create field by clicking **OK**.

25. In the Data Associations window, click **OK**, as shown in Figure 6–81.

Figure 6–81 Data Associations Window



From	To
dataObject2_body	queryWithView

□ Validate target after assigning input data associations

26. In the Properties - ServiceTask1 window, click **OK**.

27. In the response_body.xsl tab, map the **ns0:queryWithViewResponse** source element to the **ns0:queryWithViewResponse** target element.

28. In the displayed Auto Map Preferences window, retain the default values and click **OK**.

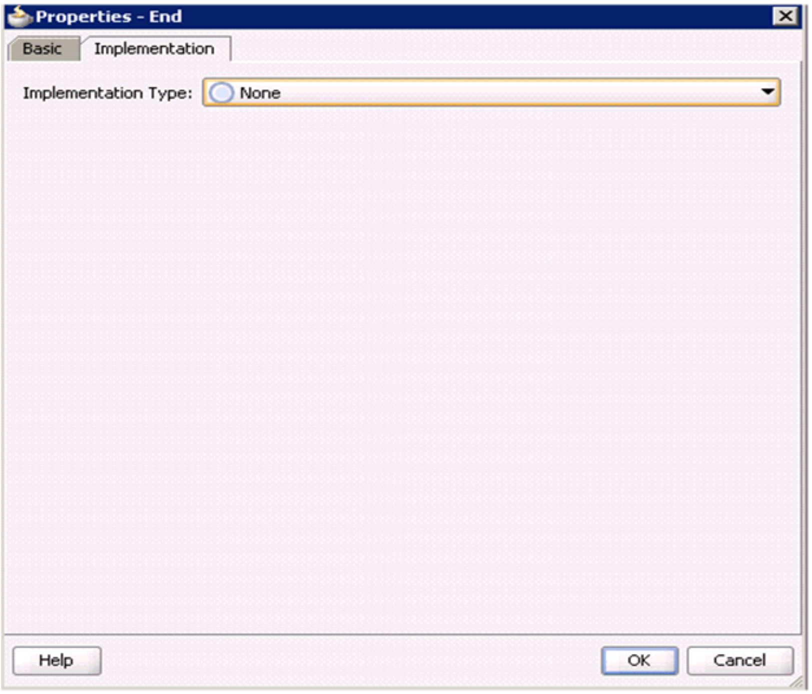
29. Return to the Process workspace area and double-click the **End** event component.

30. In the displayed Properties - End window, click the **Implementation** tab.

31. Select **None** from the Implementation Type drop-down list.

32. Click **OK**, as shown in Figure 6–82.

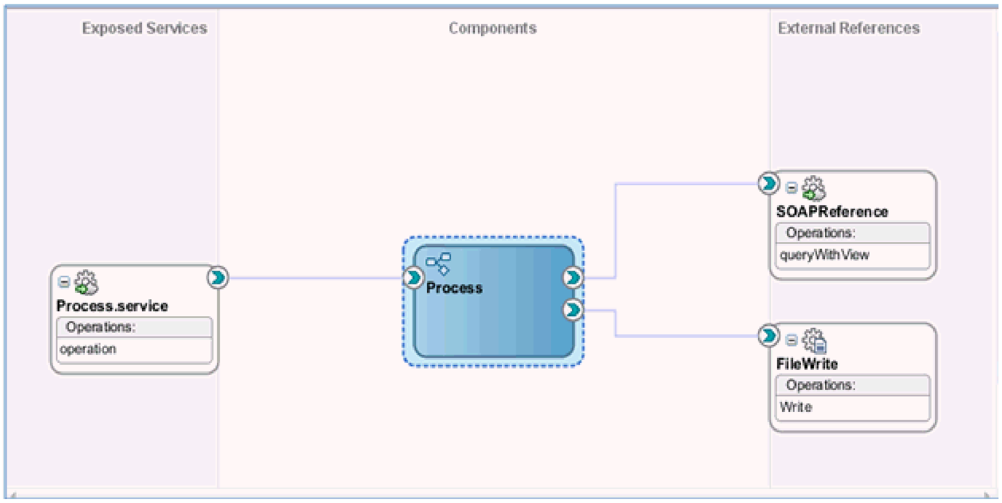
Figure 6–82 Implementation Tab



Implementation Type: None

33. Click the **Save All** icon in the menu bar to save the new outbound BPM component that was configured, as shown in Figure 6–83.

Figure 6–83 Save All Icon



You are now ready to deploy the BPM BSE Outbound process. You can follow the same procedure as Section 6.4.4, "Deploying the BPM Outbound Process" on page 6-25.

Once deployed, you can invoke the input XML as defined in Section 6.4.5, "Invoking the Input XML Document in the Oracle Enterprise Manager Console" on page 6-26.

7

Configuring an Outbound and Inbound Process for Oracle Service Bus Using sbconsole

Note: With Release 12*c* (12.2.1.0.0) configuring an outbound and inbound process for Oracle Service Bus using sbconsole has changed.

If you want to create a process for Oracle Service Bus using sbconsole, see *Chapter 2, Configuring an Outbound and Inbound Process for Oracle Service Bus Using sbconsole* in the *Oracle Fusion Middleware Application Adapters Release Notes for 12c (12.2.1.0.0)*.

Oracle Application Adapter for Siebel integrates seamlessly with Oracle Service Bus (OSB) to facilitate Web service integration. OSB is based on the Service-Oriented Architecture (SOA). It consumes adapter services exposed as Web Service Definition Language (WSDL) documents.

This chapter contains the following sections:

- Section 7.1, "Overview of Application Adapter Integration with Oracle Service Bus"
- Section 7.2, "Configuring an Outbound Process Using sbconsole (J2CA Configuration)"
- Section 7.3, "Configuring an Inbound Process Using sbconsole (J2CA Configuration)"
- Section 7.4, "Configuring an Outbound Process Using sbconsole (BSE Configuration)"
- Section 7.5, "Configuring JMS Proxy Services Using Oracle Service Bus (J2CA Configuration)"
- Section 7.6, "Configuring HTTP Proxy Services Using Oracle Service Bus (J2CA Configuration)"

7.1 Overview of Application Adapter Integration with Oracle Service Bus

To integrate with Oracle Service Bus (OSB), Oracle Application Adapter for Siebel must be deployed in the same Oracle WebLogic Server as OSB. The underlying adapter services must be exposed as WSDL files, which are generated during design time in Oracle Adapter Application Explorer (Application Explorer) for both request-response (outbound) and event notification (inbound) services of the adapter.

7.2 Configuring an Outbound Process Using sbconsole (J2CA Configuration)

This section describes how to configure an outbound process using sbconsole for J2CA configurations.

A sample project has been provided for this outbound use case scenario in the following folder of the Application Adapters installation:

```
<ADAPTER_HOME>\etc\sample\SIEBEL_Samples.zip\SIEBEL_Samples\OSB\J2CA\Siebel_
Sample_J2CA_OSB_Outbound_Project
```

This section includes the following topics:

- Section 7.2.1, "Starting Oracle Service Bus and Creating Project Folders"
- Section 7.2.2, "Setting the Class Path for Application Explorer to Integrate With Oracle Service Bus"
- Section 7.2.3, "Publishing a WSDL From Application Explorer to Oracle Service Bus"
- Section 7.2.4, "Configuring a WSDL-based Business Service"
- Section 7.2.5, "Configuring a File Type Business Service"
- Section 7.2.6, "Configuring a Pipeline With Proxy Service"

7.2.1 Starting Oracle Service Bus and Creating Project Folders

This section describes how to start Oracle Service Bus (OSB) and create project folders.

Perform the following steps to start Oracle Service Bus and create project folders:

1. Start the Oracle WebLogic Server for the Oracle WebLogic Server domain that you have configured.
2. Open the Oracle Service Bus Console in a Web browser by entering the following URL:

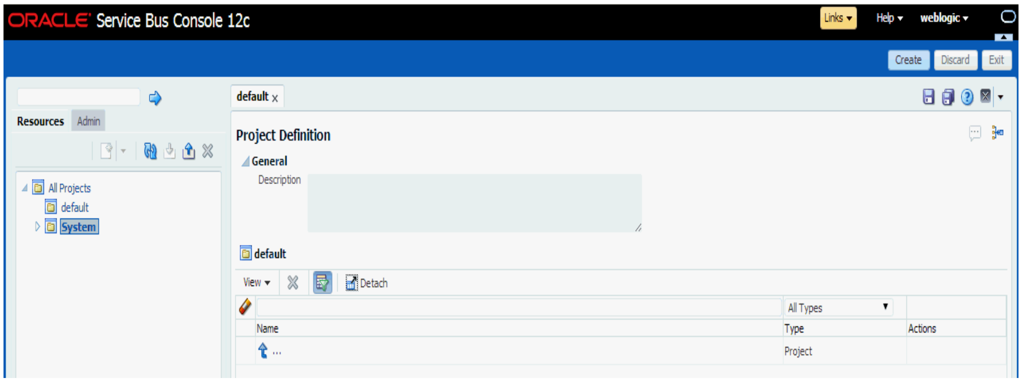
   ```
   http://hostname:port/sbconsole
   ```

   Where `hostname` is the name of the machine where Oracle WebLogic Server is running and `port` is the port for the domain you are using.

   The Oracle Service Bus Console logon page is displayed.
3. Log on to the Oracle Service Bus Console using a valid user name and password.

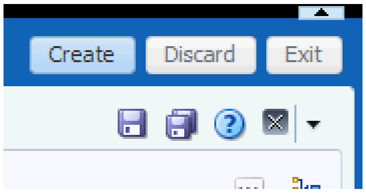
   The Oracle Service Bus Console home page is displayed, as shown in Figure 7–1.

Figure 7–1 Oracle Service Bus Console Home Page



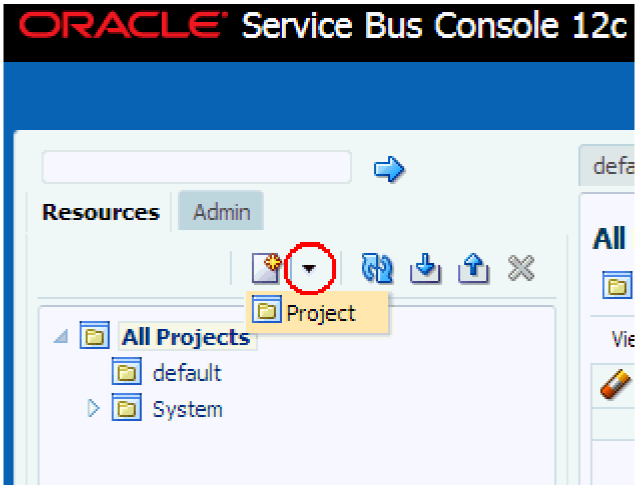
4. Click **Create** in the right pane of the Oracle Service Bus session, as shown in Figure 7–2.

Figure 7–2 Oracle Service Bus Session



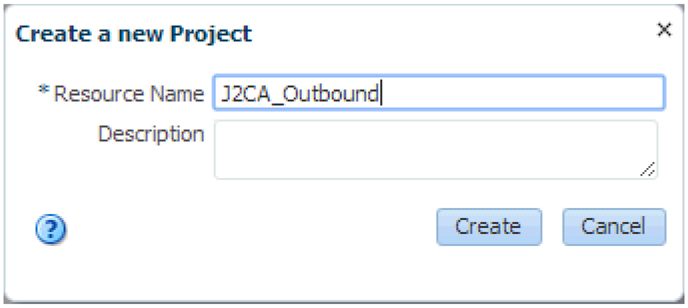
5. Select **All Projects**, click the down arrow in the left pane, and select **Project**, as shown in Figure 7–3.

Figure 7–3 All Projects Folder



The Create a new Project window is displayed, as shown in Figure 7–4.

Figure 7–4 Create New Project Window



Create a new Project

* Resource Name: J2CA_Outbound

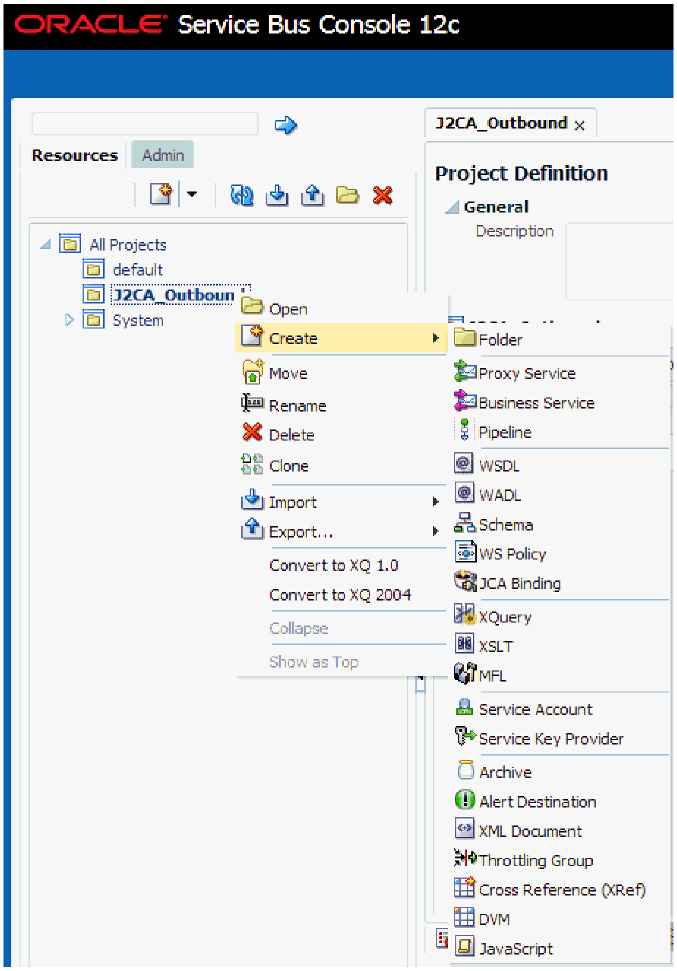
Description:

6. Provide a valid name for the new project (for example, J2CA_Outbound) in the Resource Name field, and click **Create**.

   The new project is successfully created and listed.

7. Right-click the newly created project, select **Create**, and click **Folder**, as shown in Figure 7–5.

Figure 7–5 Create Option



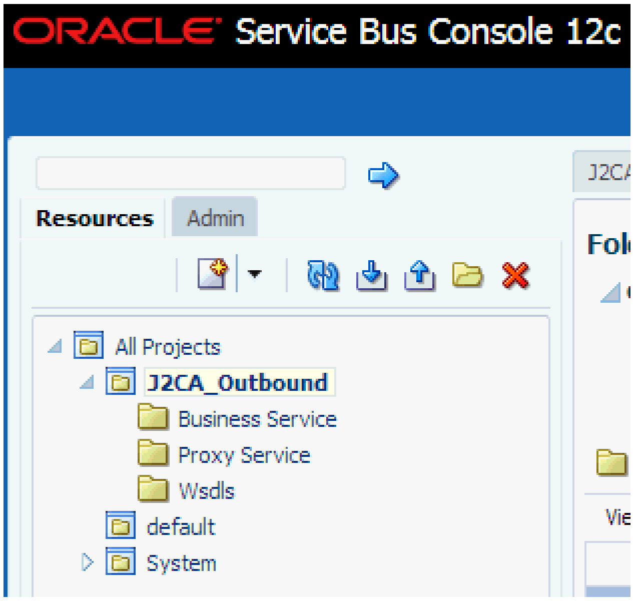
The Create a new Folder window is displayed.

8. In the Resource Name field, type **Business Service** and click **Create**.

9. Repeat steps 7 and 8 to create folders with the names **Proxy Service** and **Wsdls**.

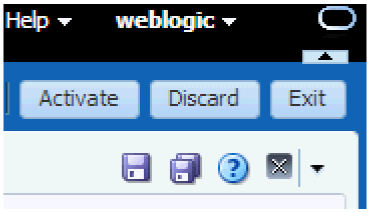
   The Business Service, Proxy Service, and Wsdls folders are listed in the left pane below the project node, as shown in Figure 7–6.

Figure 7–6 Project Node



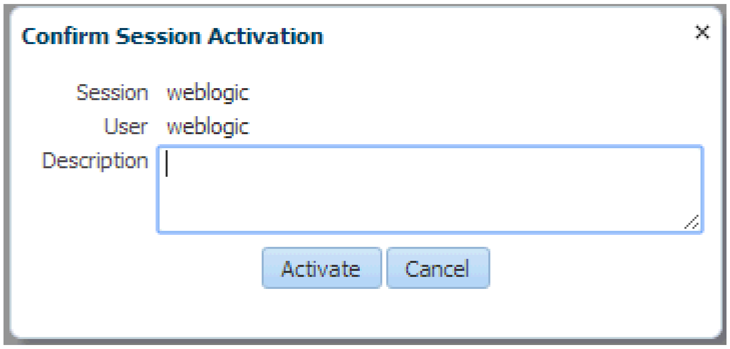
10. Click **Activate** in the right pane of the Oracle Service Bus session, as shown in Figure 7–7.

Figure 7–7 Activate Button



11. In the Confirm Session Activation page, click **Activate** to save the changes, as shown in Figure 7–8.

Figure 7–8 Confirm Session Activation Window



Confirm Session Activation

Session weblogic

User weblogic

Description

7.2.2 Setting the Class Path for Application Explorer to Integrate With Oracle Service Bus

Before starting and using Application Explorer to publish a WSDL directly to the Oracle Service Bus (OSB) Console (project/folder), OSB users must perform the following steps:

1. Open the command prompt window.
2. Navigate to the following directory:

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

3. Execute **setDomainEnv.cmd** (Windows) or **. ./setDomainEnv.sh** (UNIX/Linux).

   This command sets the class path for Application Explorer to access the Oracle WebLogic Server APIs to publish the WSDLs to the OSB Console.

4. Do not close the command prompt window.
5. Navigate to the following directory:

   ```
   <ADAPTER_HOME>\tools\iwae\bin
   ```

6. Execute **ae.bat** (Windows) or **iwae.sh** (UNIX/Linux) to start Application Explorer.

   You are now ready to publish WSDLs from Application Explorer to the OSB Console.

7.2.3 Publishing a WSDL From Application Explorer to Oracle Service Bus

Perform the following steps to publish a WSDL from Application Explorer to Oracle Service Bus:

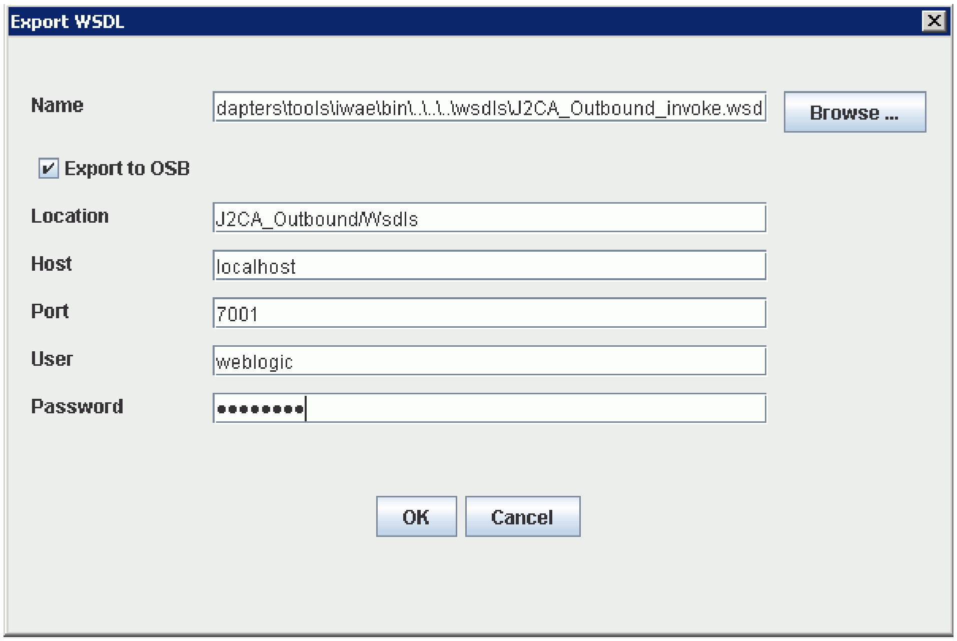
1. Start Application Explorer, connect to a J2CA configuration, and connect to a Siebel target.

   For more information, see Chapter 2, "Configuring Oracle Application Server Adapter for Siebel" on page 2-1.

2. Expand the Siebel target to which you are connected.
3. Expand **Business Object**, **Account**, and then **Account**.
4. Right-click the **queryWithView** method and then select **Create Outbound JCA Service (Request/Response)** from the menu.

   The Export WSDL dialog is displayed, as shown in Figure 7–9.

Figure 7–9 Export WSDL Dialog



Export WSDL

Name: dapters\tools\iwae\bin\..\..\..\wsdls\J2CA_Outbound_invoke.wsd

☑ Export to OSB

Location: J2CA_Outbound/Wsdls

Host: localhost

Port: 7001

User: weblogic

Password: ••••••••

5. In the Name field, a default file name for the WSDL file is provided. You can accept the default or provide your own.
6. Select the **Export to OSB** option.
7. In the Location field, enter the folder name in Oracle Service Bus where you want to publish the WSDL document.

   The location is composed of an Oracle Service Bus project name and optionally, one or more folder names. The project name and any folder names must be separated by a forward slash character "/".
8. In the Host field, enter the name of the machine where Oracle Service Bus is installed.
9. In the Port field, enter the port that is being used by Oracle Service Bus.
10. In the User field, enter your username to access Oracle Service Bus.
11. In the Password field, enter your password to access Oracle Service Bus.
12. Click **OK**.

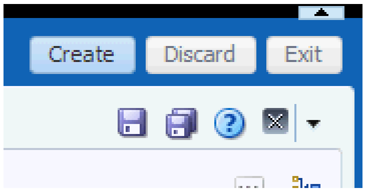
    The WSDL is published to the location specified in the Export WSDL dialog and is now available for use with a Business Service or Proxy Service in Oracle Service Bus.

7.2.4 Configuring a WSDL-based Business Service

Perform the following steps to configure a WSDL-based Proxy Service:

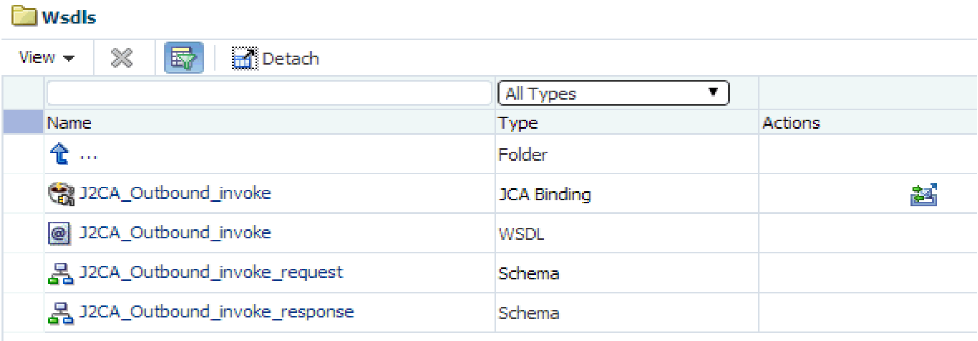
1. Open the Oracle Service Bus Console and click **Create** in the right pane of the Oracle Service Bus session, as shown in Figure 7–10.

Figure 7–10 Create Button



2. Double-click the created WSDL folder in the left pane (for example, Wsdls) and ensure that the exported WSDL is listed in the right pane, as shown in Figure 7–11.

Figure 7–11 Wsdls Folder

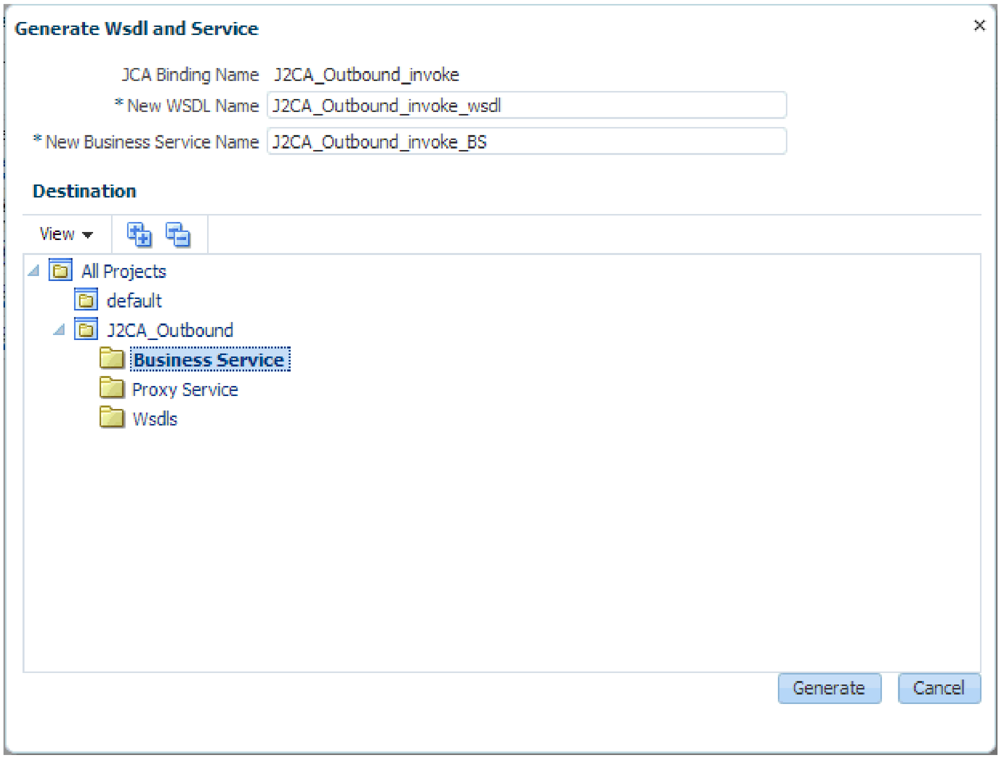


Name	Type	Actions
...	Folder	
J2CA_Outbound_invoke	JCA Binding	
J2CA_Outbound_invoke	WSDL	
J2CA_Outbound_invoke_request	Schema	
J2CA_Outbound_invoke_response	Schema	

3. Click the icon that corresponds to the JCA Binding in the Actions column.

   The Generate WSDL and Service window is displayed, as shown in Figure 7–12.

Figure 7–12 Generate WSDL and Service Window



Generate Wsdl and Service

JCA Binding Name J2CA_Outbound_invoke

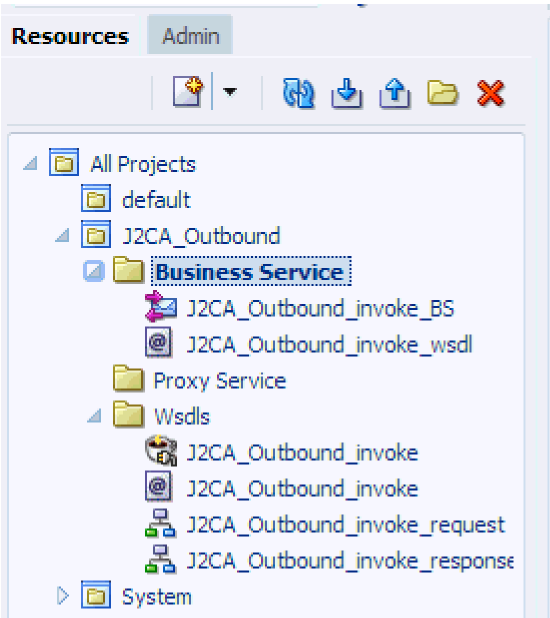
* New WSDL Name J2CA_Outbound_invoke_wsdl

* New Business Service Name J2CA_Outbound_invoke_BS

Destination

4. Provide a new WSDL name and a new Business Service name in the corresponding fields.
5. In the Destination area, select an available project and the sub-folder that is designated for Business Services.
6. Click **Generate**.
7. Expand **Business Service** under the project folder and check if the generated WSDL and Business Service are listed, as shown in Figure 7–13.

Figure 7–13 Business Service Folder

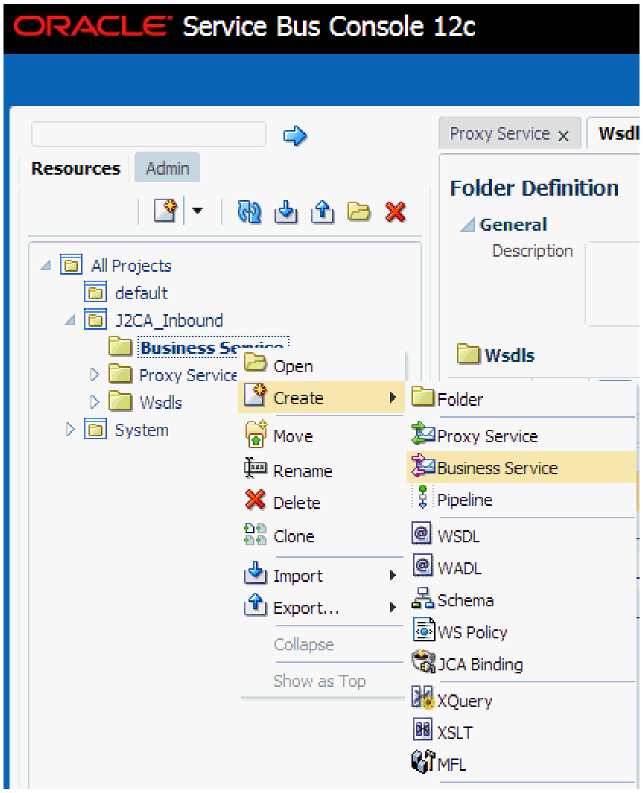


7.2.5 Configuring a File Type Business Service

Perform the following steps to configure a File type Business Service:

1. Right-click the **Business Service** folder you created in the left pane, select **Create**, and click **Business Service** as shown in Figure 7–14.

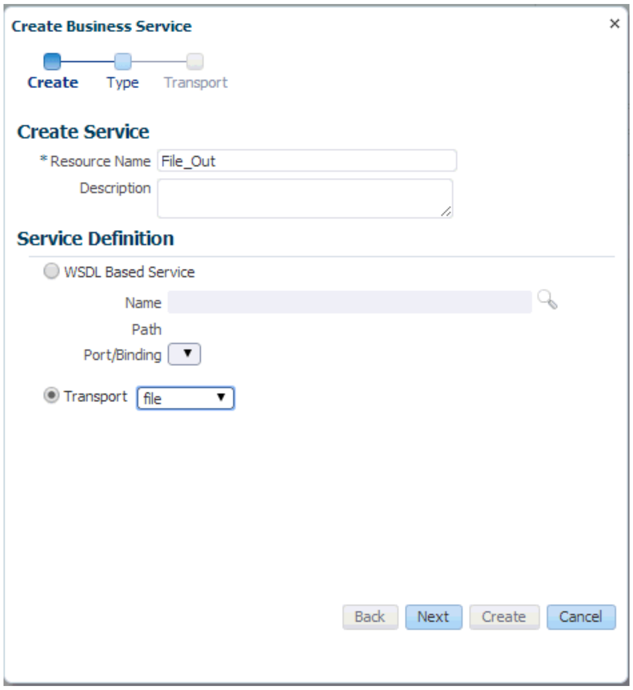
Figure 7–14 Business Service Folder



The Create Business Service window is displayed.

2. In the Resource Name field, provide a name for the Business Service, select the **File** option in the Transport section under Service Definition, and click **Next**, as shown in Figure 7–15.

Figure 7–15 Service Definition



Create Business Service ×

Create — Type — Transport

Create Service

* Resource Name: File_Out

Description:

Service Definition

○ WSDL Based Service

Name:

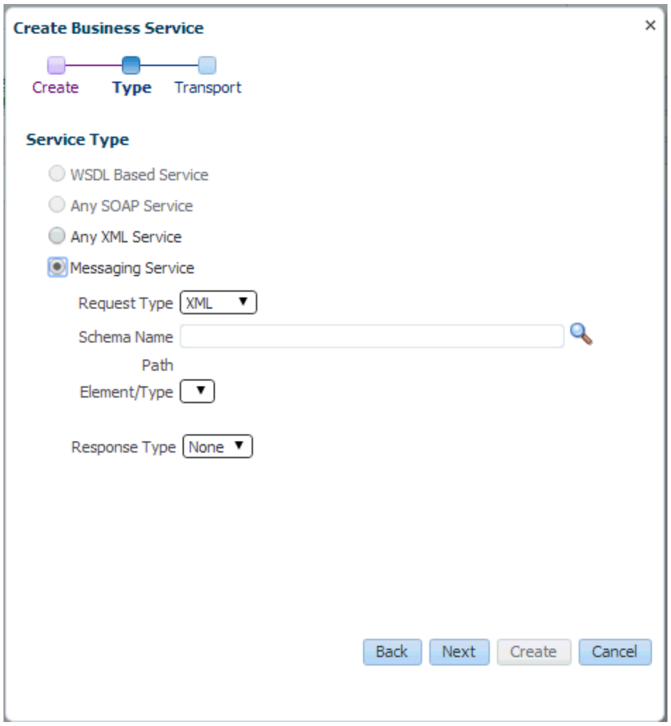
Path

Port/Binding ▼

● Transport: file ▼

3. In the Service Type section, select **Messaging Service**. By default, the Request Type is set to XML, and the Response Type is set to None. Then click **Next**, as shown in Figure 7–16.

Figure 7–16 Service Type Configuration Page



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

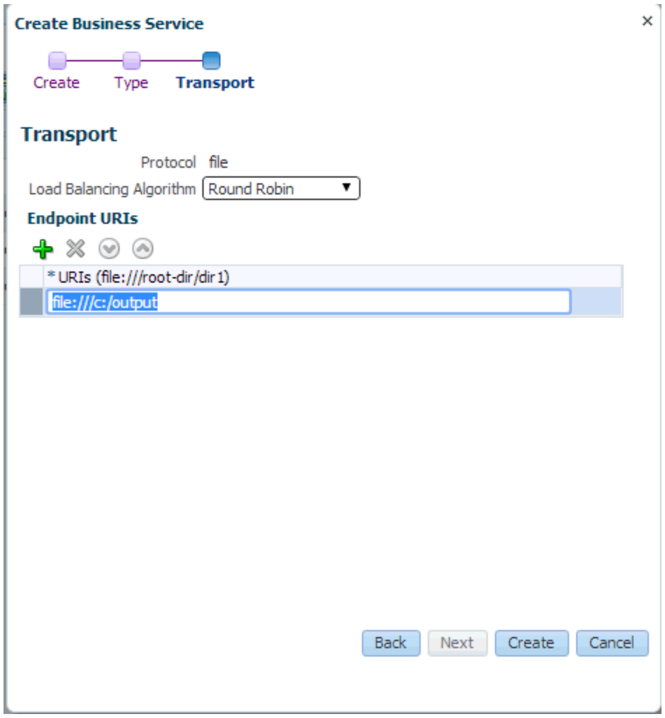
Element/Type ▼

Response Type: None ▼

4. Enter the path to a destination folder on your file system in the Endpoint URI field.

5. Click **Create**, as shown in Figure 7–17.

Figure 7–17 Transport Page



Create Business Service

Transport

Protocol file

Load Balancing Algorithm Round Robin

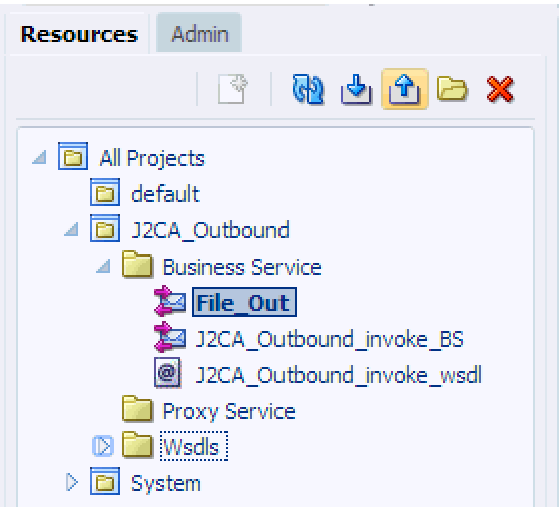
Endpoint URIs

* URIs (file:///root-dir/dir1)

file:///c:/output

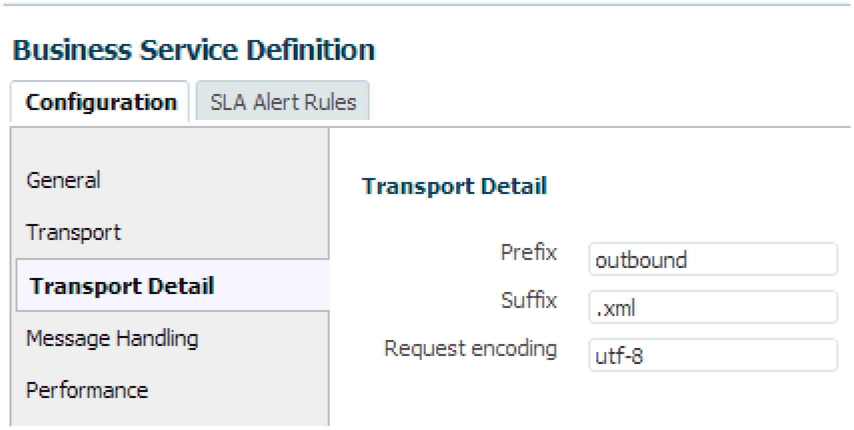
The Business Service **File_Out** is created and listed under Business Service, as shown in Figure 7–18.

Figure 7–18 File_Out Business Service



6. Double-click **File_Out**, click **Transport Detail** in the left pane, and enter the prefix and suffix for the output file to be received, as shown in Figure 7–19.

Figure 7–19 Transport Detail



Business Service Definition

Transport Detail

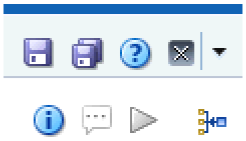
Prefix: outbound

Suffix: .xml

Request encoding: utf-8

7. Click the **Save** or **Save All** icon in the right corner, as shown in Figure 7–20.

Figure 7–20 Save/Save All Icons

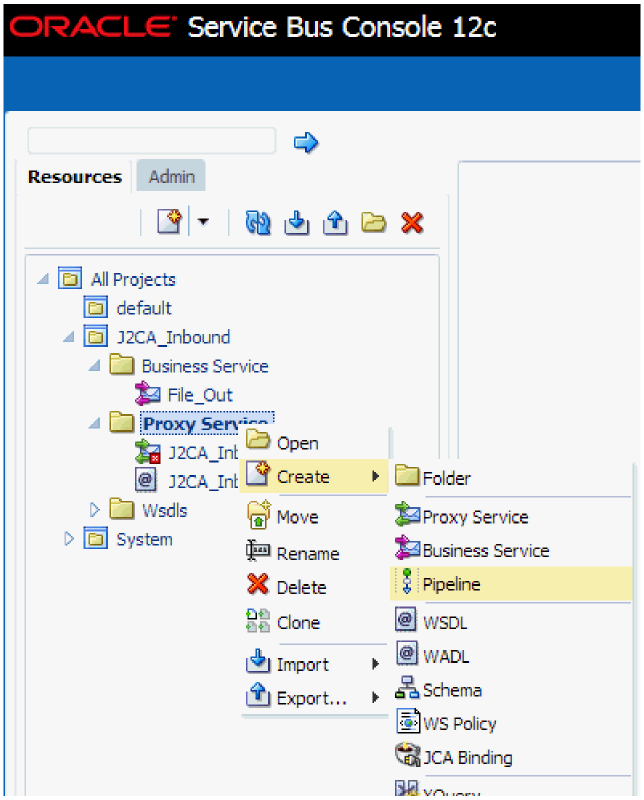


7.2.6 Configuring a Pipeline With Proxy Service

Perform the following steps to configure a Pipeline:

1. Right-click the Proxy Service folder, select **Create** and click **Pipeline**, as shown in Figure 7–21.

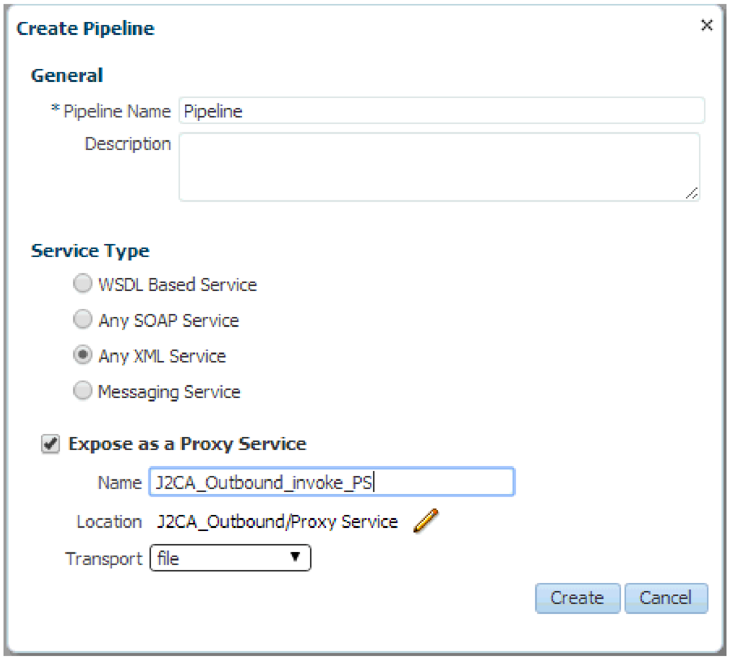
Figure 7–21 Pipeline Option



The Create Pipeline window is displayed.

2. Enter a name in the Pipeline Name field. By default, **Expose as a Proxy Service** is selected. If you wish to change the Proxy Service Name, change it and set Transport as **file**, and click **Create** as shown in Figure 7–22.

Figure 7–22 Create Pipeline Window



Create Pipeline ×

General

* Pipeline Name: Pipeline

Description:

Service Type

○ WSDL Based Service
○ Any SOAP Service
● Any XML Service
○ Messaging Service

☑ Expose as a Proxy Service

Name: J2CA_Outbound_invoke_PS

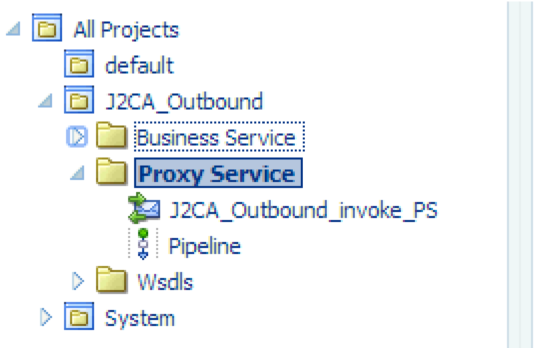
Location: J2CA_Outbound/Proxy Service

Transport: file

Create | Cancel

The created Pipeline and the Proxy Service is listed under Proxy Service, as shown in Figure 7–23.

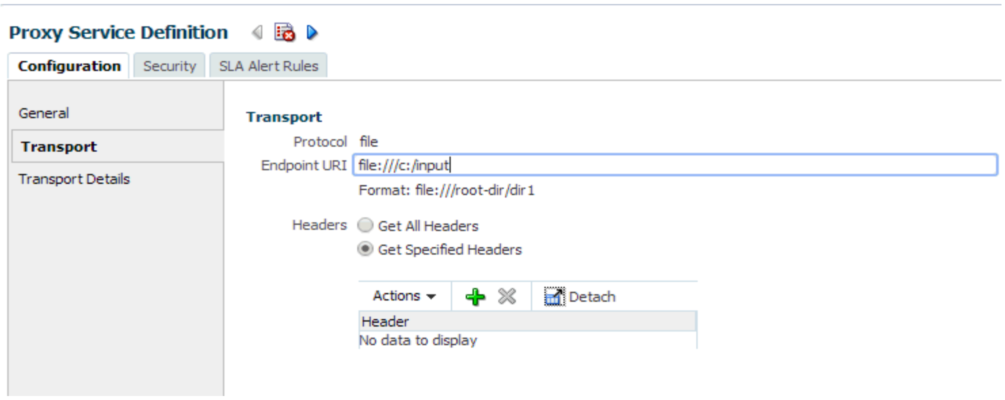
Figure 7–23 Pipeline Node



All Projects
default
J2CA_Outbound
Business Service
Proxy Service
J2CA_Outbound_invoke_PS
Pipeline
Wsdls
System

3. Double-click the created proxy service and click **Transport** in the left pane. Provide the input location in the Endpoint URI field, as shown in Figure 7–24.

Figure 7–24 Transport



Transport

Protocol file

Endpoint URI file:///c:/input

Format: file:///root-dir/dir1

Headers ○ Get All Headers

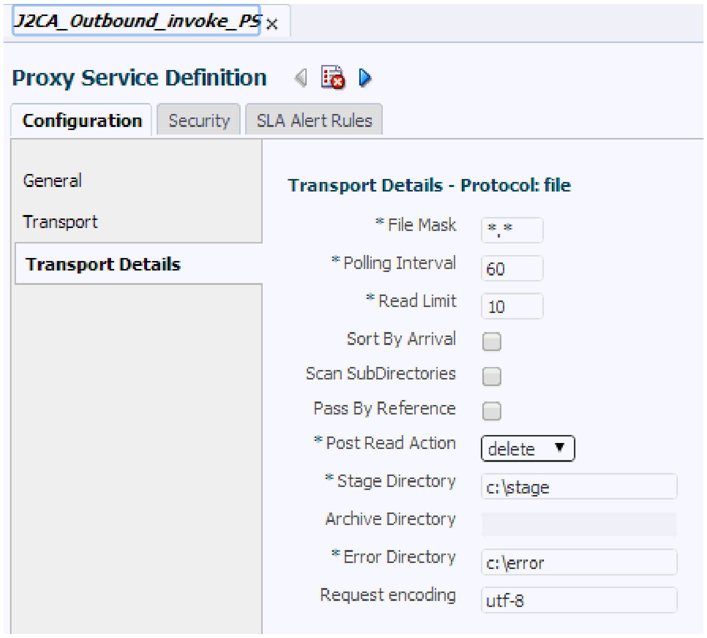
● Get Specified Headers

Header

No data to display

4. Click **Transport Details** in the left pane and provide the location for the Stage Directory and the Error Directory fields, as shown in Figure 7–25.

Figure 7–25 Transport Details



J2CA_Outbound_invoke_PS ×

Proxy Service Definition

Configuration | Security | SLA Alert Rules

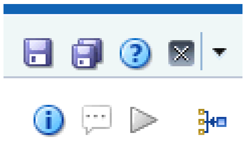
General
Transport
Transport Details

Transport Details - Protocol: file

* File Mask	*.*
* Polling Interval	60
* Read Limit	10
Sort By Arrival	□
Scan SubDirectories	□
Pass By Reference	□
* Post Read Action	delete ▼
* Stage Directory	c:\stage
Archive Directory	
* Error Directory	c:\error
Request encoding	utf-8

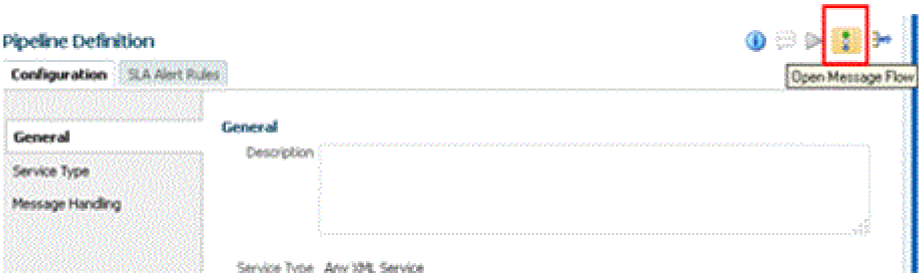
5. Click the **Save All** icon in the right corner, as shown in Figure 7–26.

Figure 7–26 Save All Icon



6. Double-click the **Pipeline** node and click the **Open Message Flow** icon on the right pane to open the message flow, as shown in Figure 7–27.

Figure 7–27 Open Message Flow Icon



Pipeline Definition

Configuration | SLA Alert Rules

Open Message Flow

General
Service Type
Message Handling

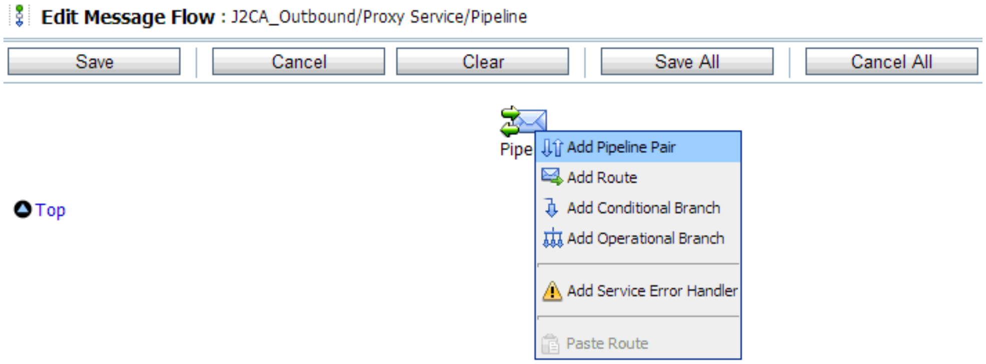
General

Description

Service Type Any XML Service

7. Click the Proxy Service icon and select **Add Pipeline Pair** from the menu, as shown in Figure 7–28.

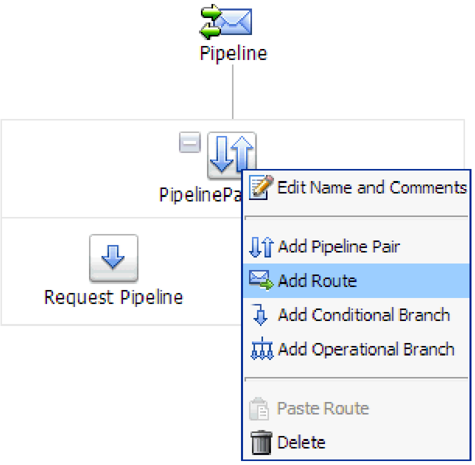
Figure 7–28 Add Pipeline Pair Option



Edit Message Flow : J2CA_Outbound/Proxy Service/Pipeline

8. Click the **PipelinePairNode1** icon and select **Add Route** from the menu, as shown in Figure 7–29.

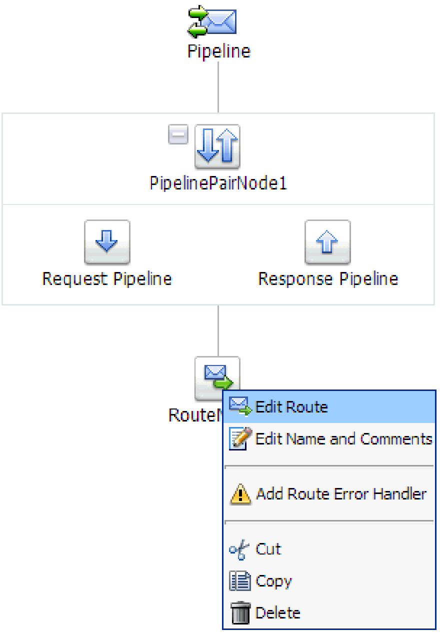
Figure 7–29 Add Route Option



The RouteNode1 icon is added below the PipelinePairNode1 icon.

9. Click the RouteNode1 icon and select **Edit Route** from the menu, as shown in Figure 7–30.

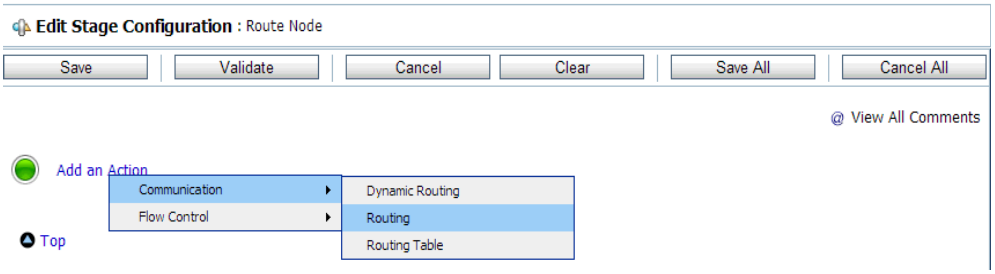
Figure 7–30 Edit Route Option



The Edit Stage Configuration workspace area is displayed.

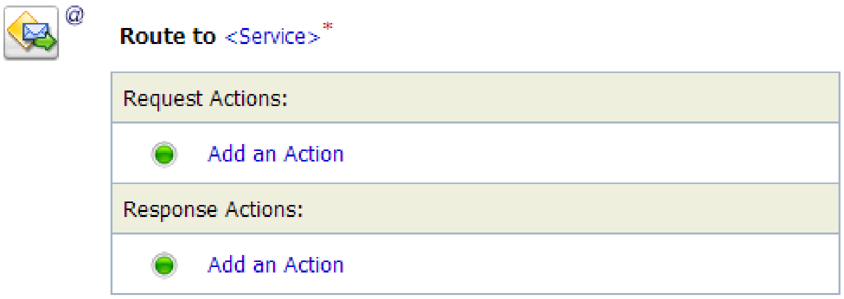
10. Click **Add an Action**, select **Communication** and click **Routing**, as shown in Figure 7–31.

Figure 7–31 Edit Stage Configuration Workspace Area



11. Click **<Service>**, as shown in Figure 7–32.

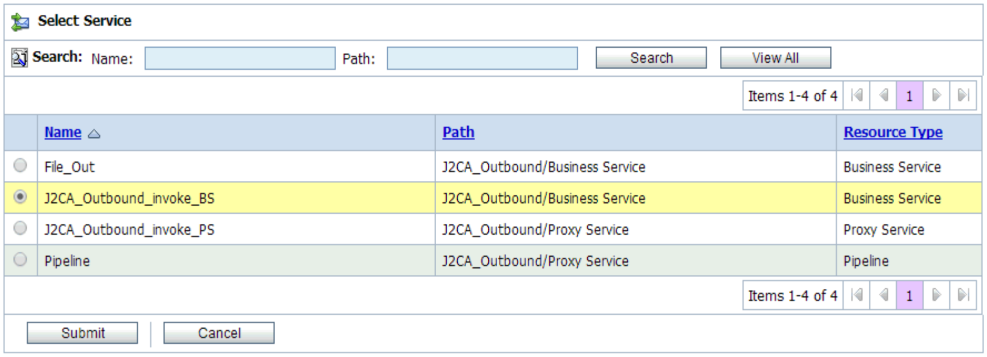
Figure 7–32 Actions



The Select Service dialog is displayed.

12. Select the WSDL type Business Service configured for Siebel and click on **Submit**, as shown in Figure 7–33.

Figure 7–33 Select Service Dialog



Select Service

Search: Name: Path: Search View All

Items 1-4 of 4 1

	Name	Path	Resource Type
○	File_Out	J2CA_Outbound/Business Service	Business Service
●	J2CA_Outbound_invoke_BS	J2CA_Outbound/Business Service	Business Service
○	J2CA_Outbound_invoke_PS	J2CA_Outbound/Proxy Service	Proxy Service
○	Pipeline	J2CA_Outbound/Proxy Service	Pipeline

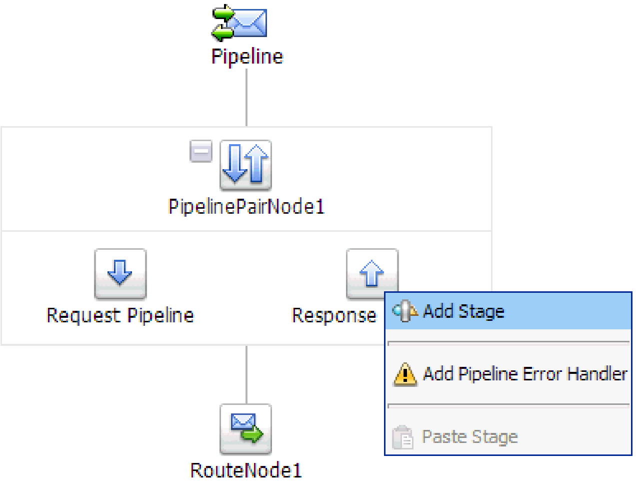
Items 1-4 of 4 1

Submit Cancel

13. Select the name of the Siebel business object (for example, queryWithView) as the operational attribute from the list, and click **Save**.

14. Click the Response Pipeline icon and select **Add Stage** from the menu, as shown in Figure 7–34.

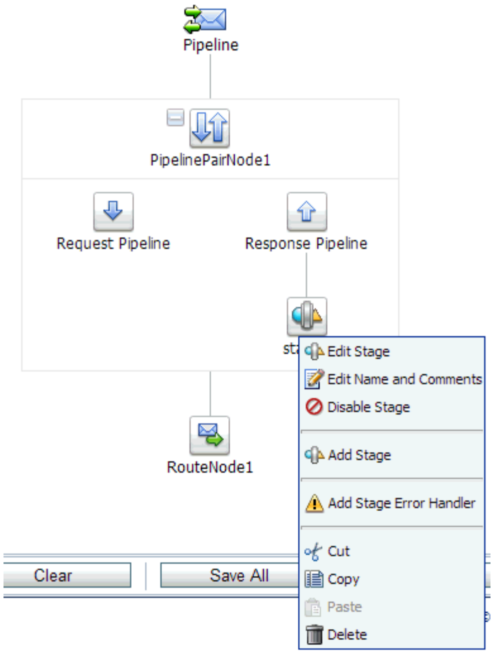
Figure 7–34 Response Pipeline Icon



The Stage1 icon is added below the Response Pipeline icon.

15. Click the Stage1 icon and select **Edit Stage** from the menu, as shown in Figure 7–35.

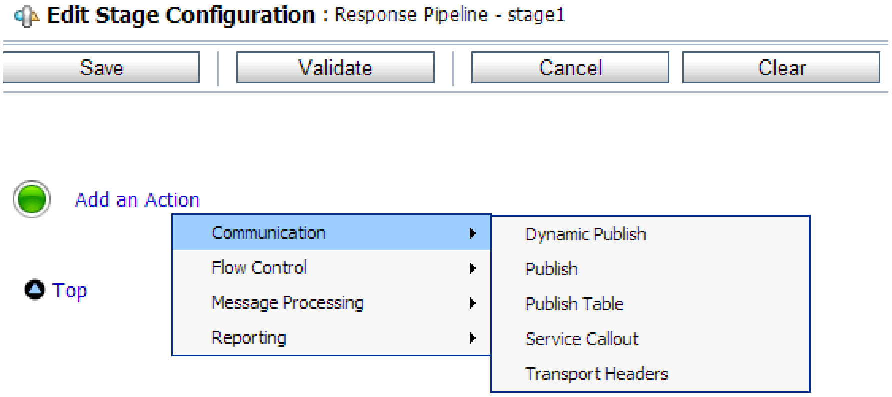
Figure 7–35 Edit Stage Option



The Edit Stage Configuration workspace area is displayed.

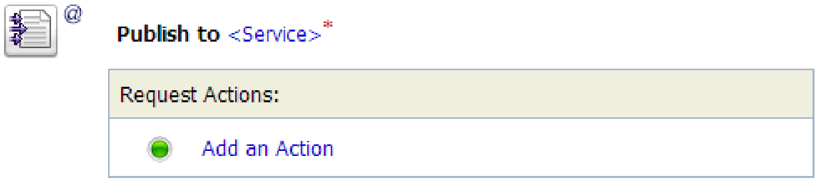
16. Click **Add an Action**, select **Communication**, and then click **Publish**, as shown in Figure 7–36.

Figure 7–36 Edit Stage Configuration Workspace Area



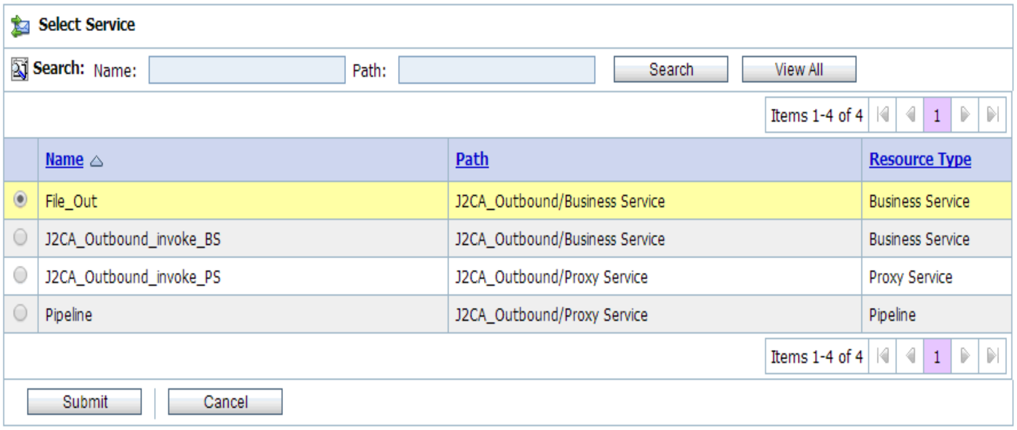
17. Click **<Service>**, as shown in Figure 7–37.

Figure 7–37 <Service> Action



18. In the Select Service dialog, select a File type Business Service and click **Submit**, as shown in Figure 7–38.

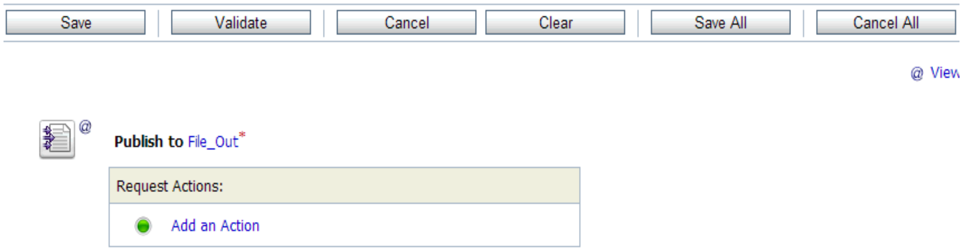
Figure 7–38 Select Service Dialog



	Name	Path	Resource Type
●	File_Out	J2CA_Outbound/Business Service	Business Service
○	J2CA_Outbound_invoke_BS	J2CA_Outbound/Business Service	Business Service
○	J2CA_Outbound_invoke_PS	J2CA_Outbound/Proxy Service	Proxy Service
○	Pipeline	J2CA_Outbound/Proxy Service	Pipeline

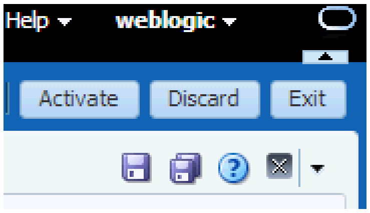
19. Click **Save All**, as shown in Figure 7–39.

Figure 7–39 Save All Button



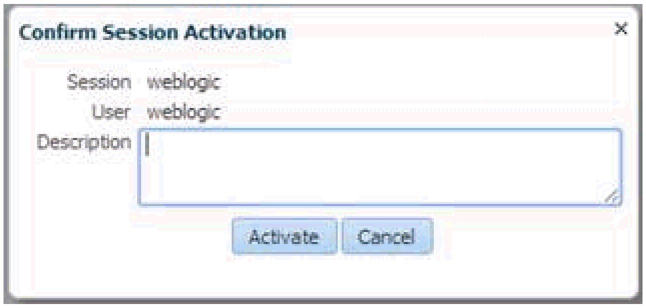
20. Click **Activate** in the right pane of the Oracle Service Bus session, as shown in Figure 7–40.

Figure 7–40 Activate Button



21. Click **Activate** to save the changes, as shown in Figure 7–41.

Figure 7–41 Confirm Session Activation



Confirm Session Activation

Session weblogic
User weblogic
Description

22. Copy and paste an input XML file in the input folder you have configured (for example, C:\input). Output is received in the configured output location (for example, C:\output).

7.3 Configuring an Inbound Process Using sbconsole (J2CA Configuration)

This section describes how to configure an inbound process using sbconsole for J2CA configurations.

A sample project has been provided for this inbound use case scenario in the following folder of the Application Adapters installation:

```
<ADAPTER_HOME>\etc\sample\SIEBEL_Samples.zip\SIEBEL_Samples\OSB\J2CA\Siebel_
Sample_J2CA_OSB_Inbound_Project
```

This section includes the following topics:

- Section 7.3.1, "Starting Oracle Service Bus and Creating Project Folders"
- Section 7.3.2, "Setting the Class Path for Application Explorer to Integrate With Oracle Service Bus"
- Section 7.3.3, "Publishing a WSDL From Application Explorer to Oracle Service Bus"
- Section 7.3.4, "Configuring a WSDL-based Proxy Service"
- Section 7.3.5, "Configuring a File Type Business Service"
- Section 7.3.6, "Configuring a Pipeline"

7.3.1 Starting Oracle Service Bus and Creating Project Folders

For more information on starting Oracle Service Bus and creating project folders, see Section 7.2.1, "Starting Oracle Service Bus and Creating Project Folders" on page 7-2.

7.3.2 Setting the Class Path for Application Explorer to Integrate With Oracle Service Bus

For more information on setting the class path for Application Explorer to integrate with Oracle Service Bus, see Section 7.2.2, "Setting the Class Path for Application Explorer to Integrate With Oracle Service Bus" on page 7-6.

7.3.3 Publishing a WSDL From Application Explorer to Oracle Service Bus

Perform the following steps to publish a WSDL from Application Explorer to Oracle Service Bus:

1. Start Application Explorer, connect to a J2CA configuration, and connect to a Siebel target.

   For more information, see Chapter 2, "Configuring Oracle Application Server Adapter for Siebel" on page 2-1.

2. Create a `Siebel` channel.

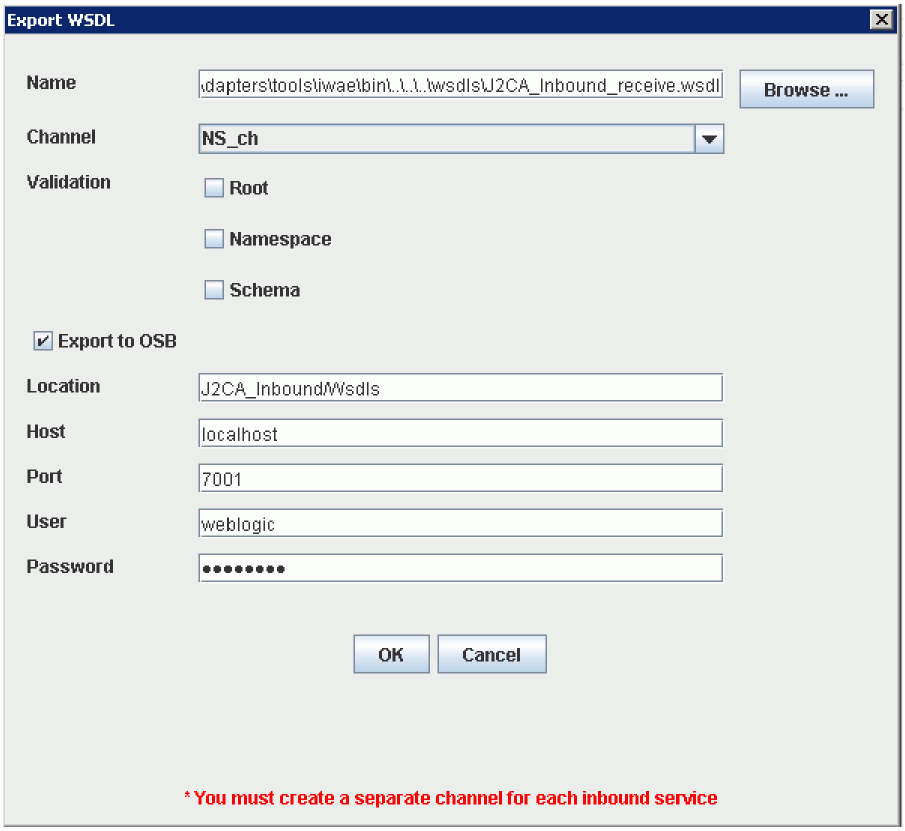
   For more information, see Section 4.5.1.1, "Creating a Channel" on page 4-35.

3. Create an Integration Object Node. For more information see Section 4.5.1.2, "Creating an Integration Object Node" on page 4-38.

4. Right-click the created Integration node and select **Create Inbound JCA Service(Event)** from the menu.

   The Export WSDL dialog is displayed, as shown in Figure 7–42.

Figure 7–42 Export WSDL Dialog



Export WSDL

Name: \dapters\tools\iwae\bin\..\..\..\wsdls\J2CA_Inbound_receive.wsdl

Channel: NS_ch

Validation:
☐ Root
☐ Namespace
☐ Schema

☑ Export to OSB

Location: J2CA_Inbound/Wsdls

Host: localhost

Port: 7001

User: weblogic

Password: ••••••••

* You must create a separate channel for each inbound service

5. In the Name field, a default file name for the WSDL file is provided. You can accept the default or provide your own.

6. From the Channel list, select the channel you created for this inbound service.

7. Three check boxes for Root, Namespace, and Schema validation are also available. Selection of multiple validation options is allowed.

8. Select the **Export to OSB** option.

9. In the Location field, enter the folder name in Oracle Service Bus where you want to publish the WSDL document.

   The location is composed of an Oracle Service Bus project name and optionally, one or more folder names. The project name and any folder names must be separated by a forward slash character "/".

10. In the Host field, enter the name of the machine where Oracle Service Bus is installed.

11. In the Port field, enter the port that is being used by Oracle Service Bus.

12. In the User field, enter your username to access Oracle Service Bus.

13. In the Password field, enter your password to access Oracle Service Bus.

14. Click **OK**.

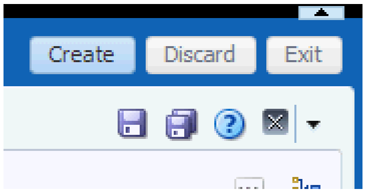
    The inbound WSDL is published to the location specified in the Export WSDL dialog and is now available for use with a Proxy Service in Oracle Service Bus.

7.3.4 Configuring a WSDL-based Proxy Service

Perform the following steps to configure a WSDL-based Proxy Service:

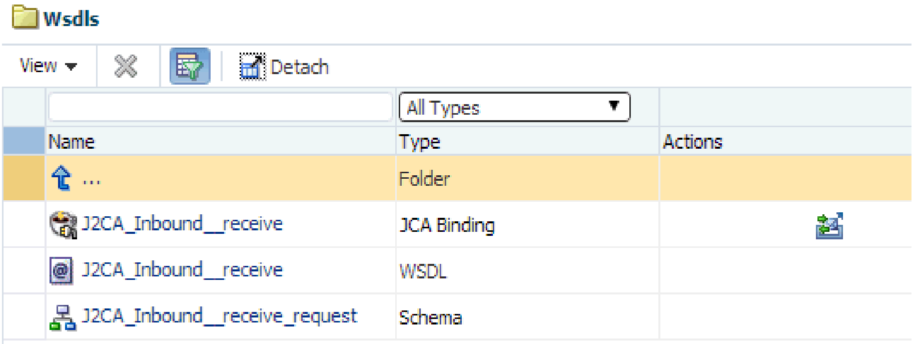
1. Open the Oracle Service Bus Console and click **Create** in the right pane of the Oracle Service Bus session, as shown in Figure 7–43.

Figure 7–43 Create Button



2. Double-click the created WSDL folder in the left pane (for example, Wsdls), and ensure that the exported WSDL is listed in the right pane, as shown in Figure 7–44.

Figure 7–44 Exported WSDL

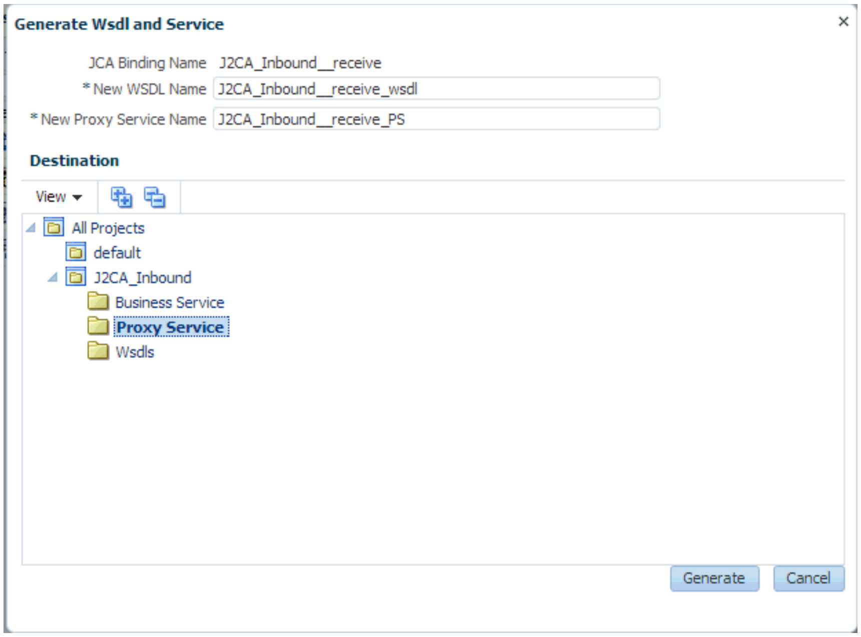


Name	Type	Actions
...	Folder	
J2CA_Inbound__receive	JCA Binding	
J2CA_Inbound__receive	WSDL	
J2CA_Inbound__receive_request	Schema	

3. Click the icon that corresponds to the JCA Binding in the Actions column.

   The Generate WSDL and Service page is displayed, as shown in Figure 7–45.

Figure 7–45 Generate WSDL and Service Page



Generate Wsdl and Service

JCA Binding Name J2CA_Inbound__receive

* New WSDL Name J2CA_Inbound__receive_wsdl

* New Proxy Service Name J2CA_Inbound__receive_PS

Destination

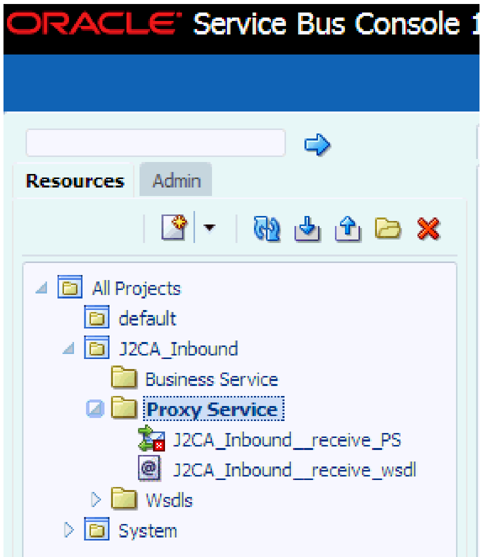
4. Provide a new WSDL name and a new Proxy Service name in the corresponding fields.

5. In the Destination area, select an available project and the sub-folder that is designated for Proxy Services.

6. Click **Generate**.

7. Expand **Proxy Service** under Project Explorer and check if the generated WSDL and Proxy Service are listed, as shown in Figure 7–46.

Figure 7–46 Generated WSDL

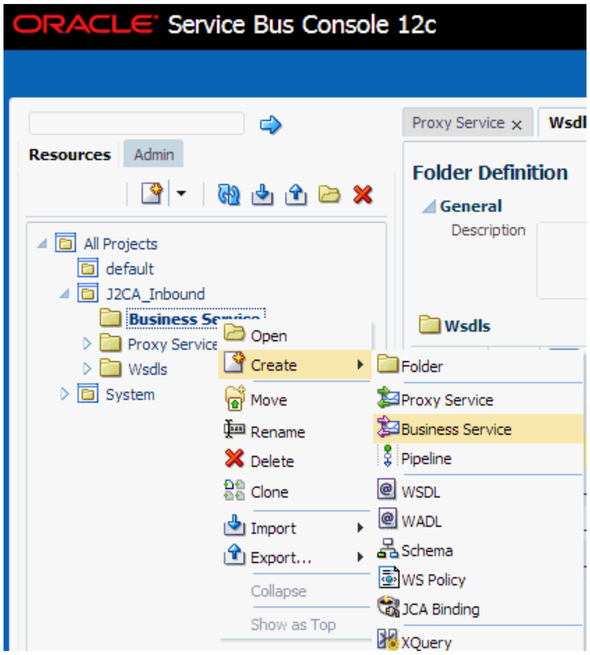


7.3.5 Configuring a File Type Business Service

Perform the following steps to configure a File type Business Service:

1. Right-click the Business Service folder you created in the left pane, select **Create**, and click **Business Service**, as shown in Figure 7–47.

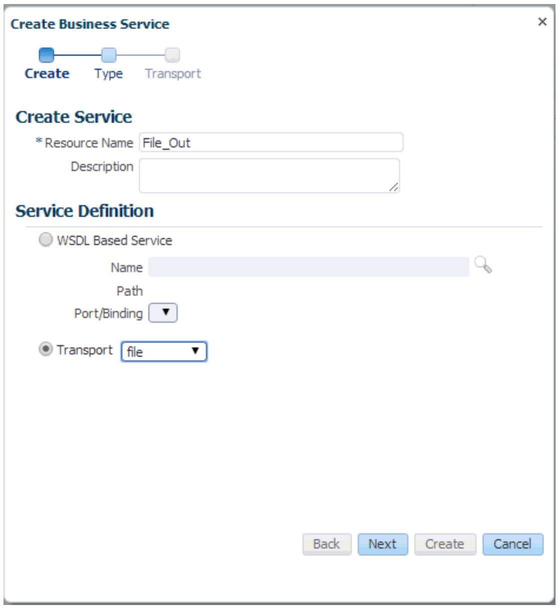
Figure 7–47 Business Service Folder



The Create Business Service window is displayed.

2. In the Resource Name field, provide a name for the Business Service and select the **File** option from the Transport drop-down list in the Service Definition area, as shown in Figure 7–48.

Figure 7–48 Create Business Service Window



Create Service

* Resource Name: File_Out

Description:

Service Definition

○ WSDL Based Service

Name:

Path:

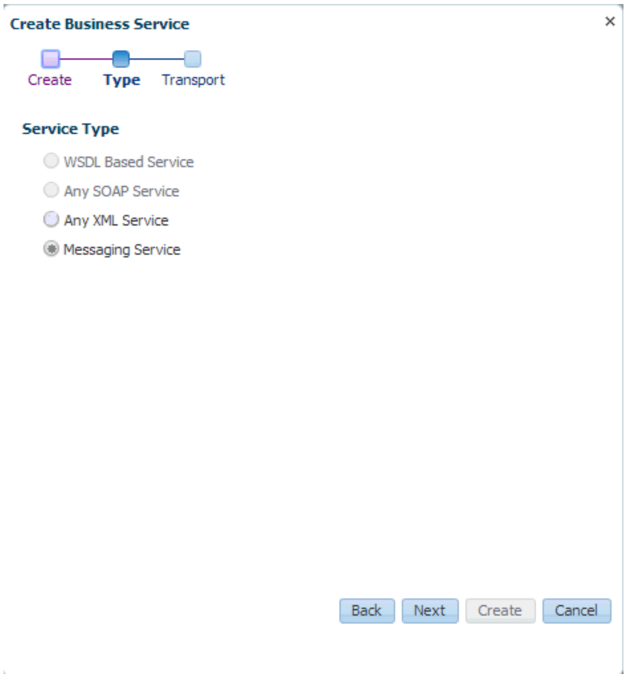
Port/Binding:

● Transport: file

3. Click **Next**.

4. In the Service Type area, select **Messaging Service** as the service type, as shown in Figure 7–49.

Figure 7–49 Service Type Area



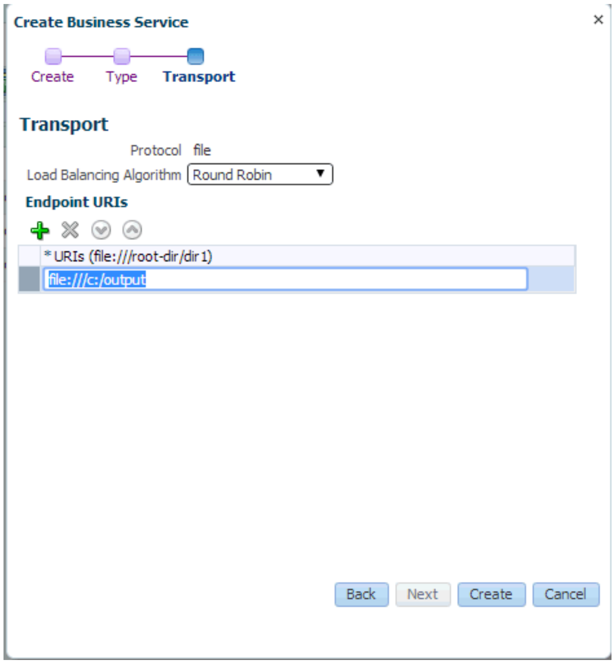
Service Type

○ WSDL Based Service
○ Any SOAP Service
○ Any XML Service
● Messaging Service

5. Click **Next**.

   The Transport page is displayed, as shown in Figure 7–50.

Figure 7–50 Transport Page



Transport

Protocol file

Load Balancing Algorithm Round Robin ▼

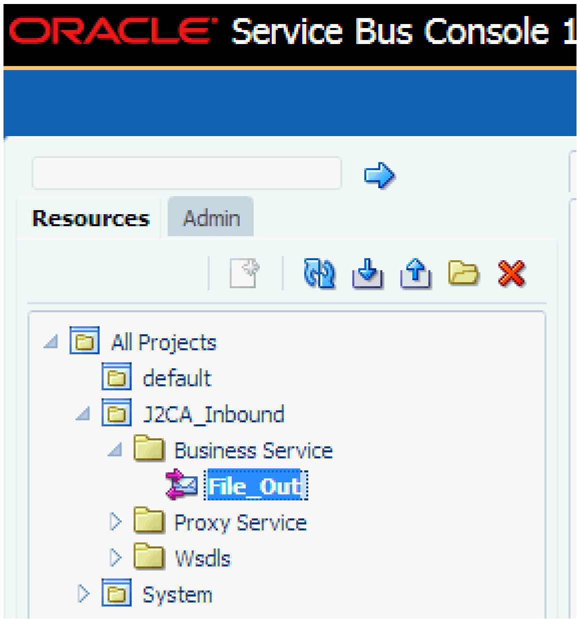
Endpoint URIs

* URIs (file:///root-dir/dir1)
file:///c:/output

6. Enter the path to a destination folder on your file system in the Endpoint URI field and click **Create**.

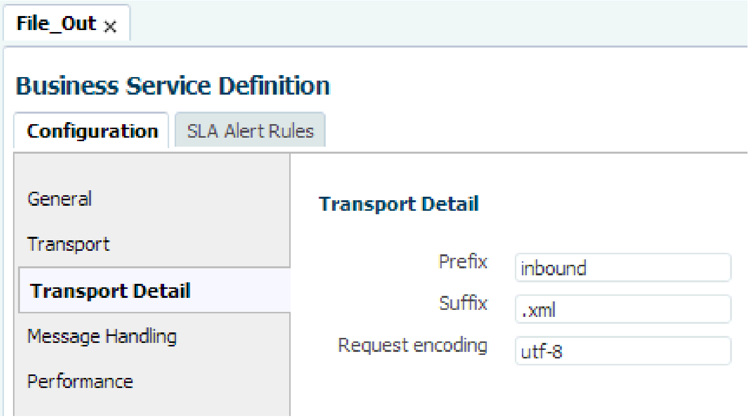
   The Business Service File_Out is created and listed under Business Service, as shown in Figure 7–51.

Figure 7–51 File_Out Business Service



7. Double-click **File_Out**, click **Transport Detail** in the left pane, and enter the prefix and suffix for the output file to be received, as shown in Figure 7–52.

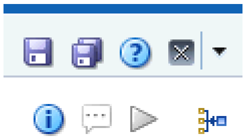
Figure 7–52 Transport Detail Page



Prefix: inbound
Suffix: .xml
Request encoding: utf-8

8. Click the Save or Save All icon in the right corner, as shown in Figure 7–53.

Figure 7–53 Save Icons

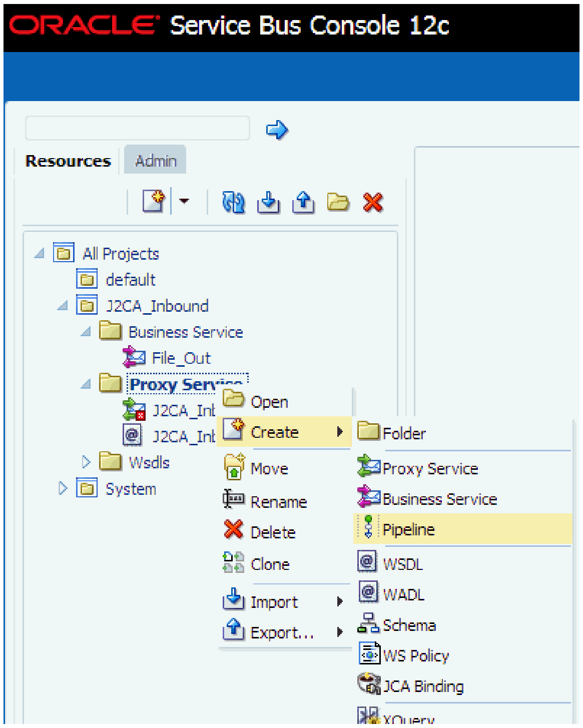


7.3.6 Configuring a Pipeline

Perform the following steps to configure a Pipeline:

1. Right-click the proxy service you created and select **Create**, and then click **Pipeline**, as shown in Figure 7–54.

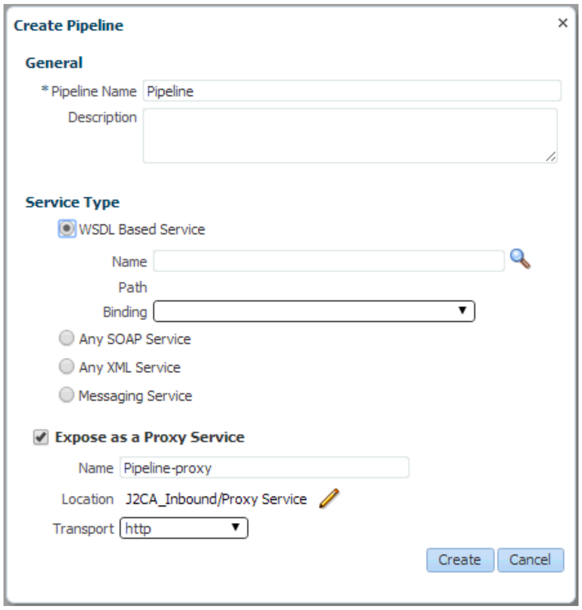
Figure 7–54 Pipeline Option



The Create Pipeline window is displayed.

2. In the Pipeline Name field, enter a name and select the Service Type as **WSDL Based Service**, as shown in Figure 7–55.

Figure 7–55 Create Pipeline Window



Create Pipeline ×

General

* Pipeline Name: Pipeline

Description:

Service Type

● WSDL Based Service

Name:

Path

Binding:

○ Any SOAP Service

○ Any XML Service

○ Messaging Service

☑ Expose as a Proxy Service

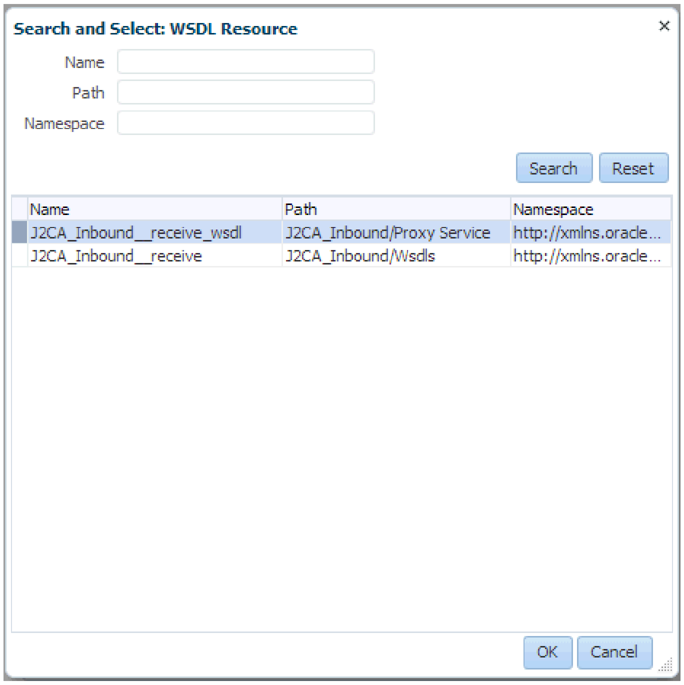
Name: Pipeline-proxy

Location: J2CA_Inbound/Proxy Service

Transport: http

3. Click the Search icon, and in the displayed Search and Select: WSDL Resource window, select **J2CA_Inbound_receive_wsdl**, and click **OK**, as shown in Figure 7–56.

Figure 7–56 Search and Select: WSDL Resource Window



Search and Select: WSDL Resource ×

Name:

Path:

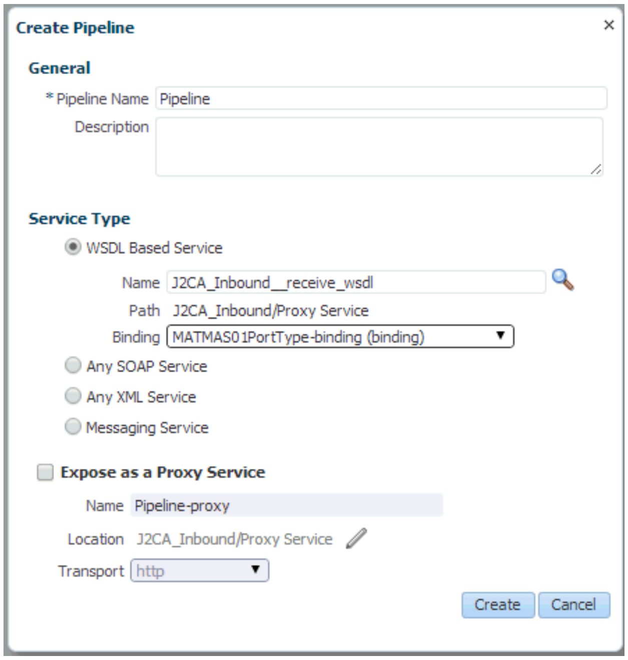
Namespace:

Name	Path	Namespace
J2CA_Inbound__receive_wsdl	J2CA_Inbound/Proxy Service	http://xmlns.oracle...
J2CA_Inbound__receive	J2CA_Inbound/Wsdls	http://xmlns.oracle...

The Create Pipeline window opens.

4. Clear the check box for **Expose as a Proxy Service**, and click **Create**, as shown in Figure 7–57.

Figure 7–57 Create Pipeline Window



Create Pipeline

General

* Pipeline Name: Pipeline

Description:

Service Type

● WSDL Based Service

Name: J2CA_Inbound__receive_wsdl

Path: J2CA_Inbound/Proxy Service

Binding: MATMAS01PortType-binding (binding)

○ Any SOAP Service

○ Any XML Service

○ Messaging Service

□ Expose as a Proxy Service

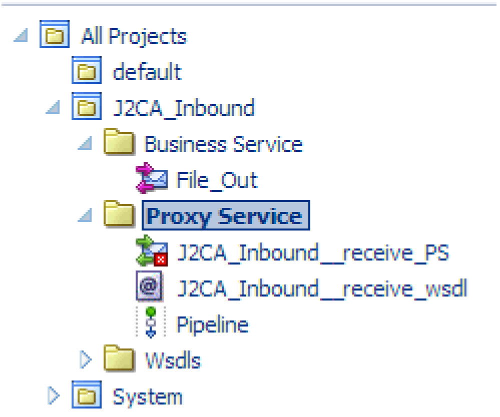
Name: Pipeline-proxy

Location: J2CA_Inbound/Proxy Service

Transport: http

The pipeline is created and listed under Proxy Service, as shown in Figure 7–58.

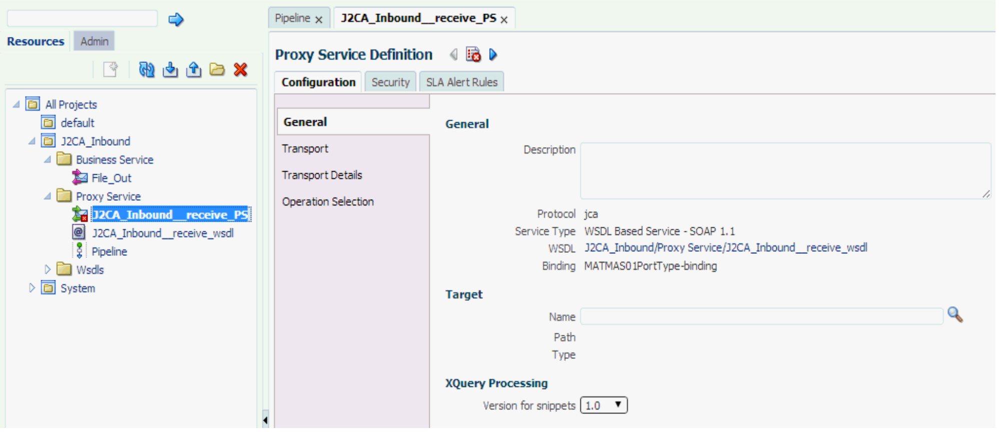
Figure 7–58 Proxy Service Pipeline



All Projects
- default
- J2CA_Inbound
  - Business Service
    - File_Out
  - Proxy Service
    - J2CA_Inbound__receive_PS
    - J2CA_Inbound__receive_wsdl
    - Pipeline
  - Wsdls
- System

5. Double-click the **J2CA_Inbound_receive_PS** node under Proxy Service in the left pane and click the **Search** icon in the Target area in right pane, as shown in Figure 7–59.

Figure 7–59 Proxy Service Definition Window



Protocol jca

Service Type WSDL Based Service - SOAP 1.1

WSDL J2CA_Inbound/Proxy Service/J2CA_Inbound__receive_wsdl

Binding MATMAS01PortType-binding

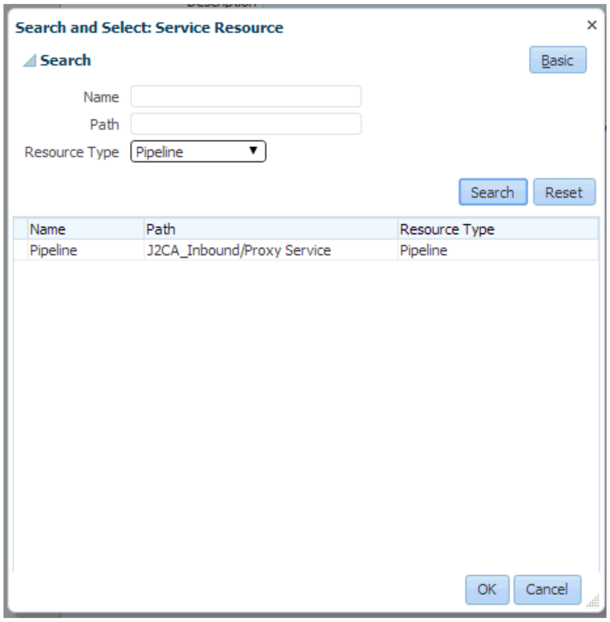
Version for snippets 1.0

The Search and Select: Service Resource window appears.

6. From the Resource Type drop-down list, select **Pipeline** and then click the **Search** button.

   The Pipeline is listed, as shown in Figure 7–60.

Figure 7–60 Search and Select: Service Resource Window



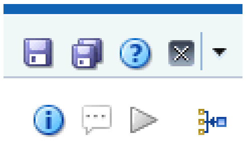
Resource Type Pipeline

Name	Path	Resource Type
Pipeline	J2CA_Inbound/Proxy Service	Pipeline

7. Select the Pipeline and click **OK**.

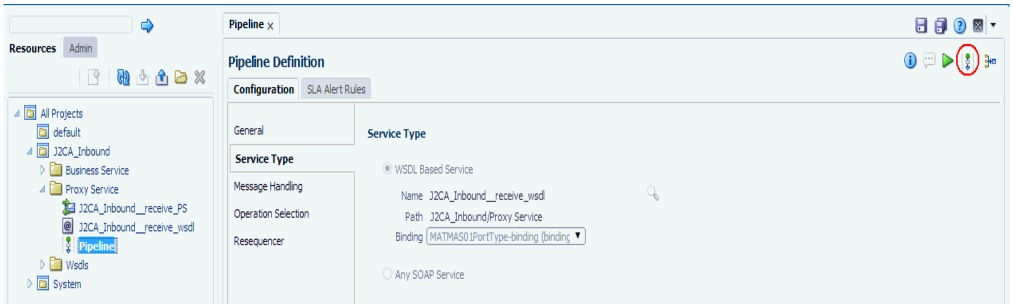
8. Click the Save or Save All icon in the right corner, as shown in Figure 7–61.

Figure 7–61 Save and Save All Icons



9. In the left pane, double-click **Pipeline** under the Proxy Service folder and click the down-pointing icon on the right pane to open the message flow, as shown in Figure 7–62.

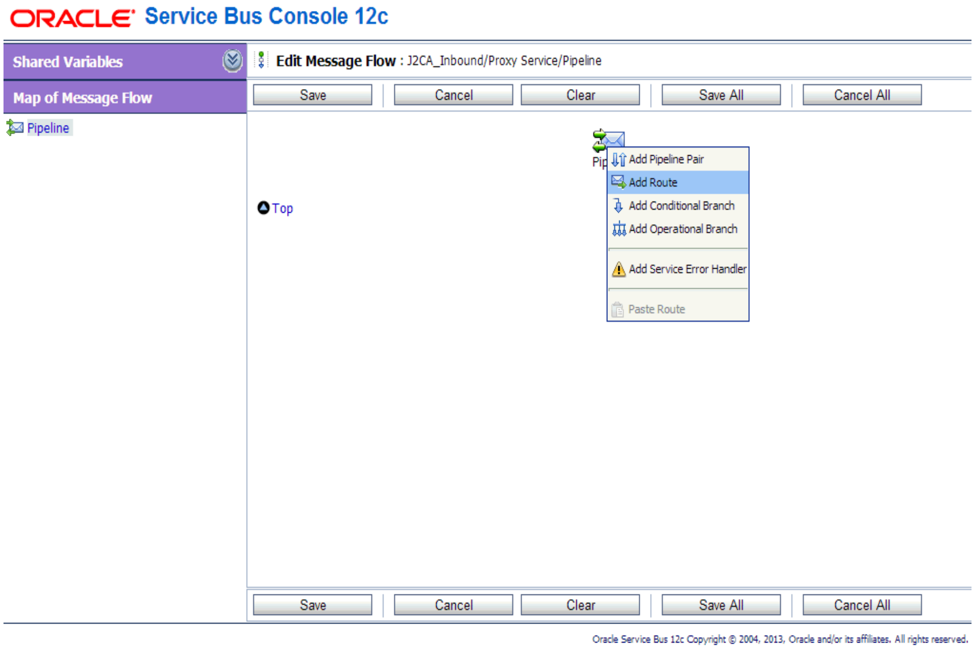
Figure 7–62 Message Flow



Name J2CA_Inbound__receive_wsdl
Path J2CA_Inbound/Proxy Service
Binding MATMAS01PortType-binding (binding

10. Click the displayed Proxy service icon and select **Add Route** from the menu, as shown in Figure 7–63.

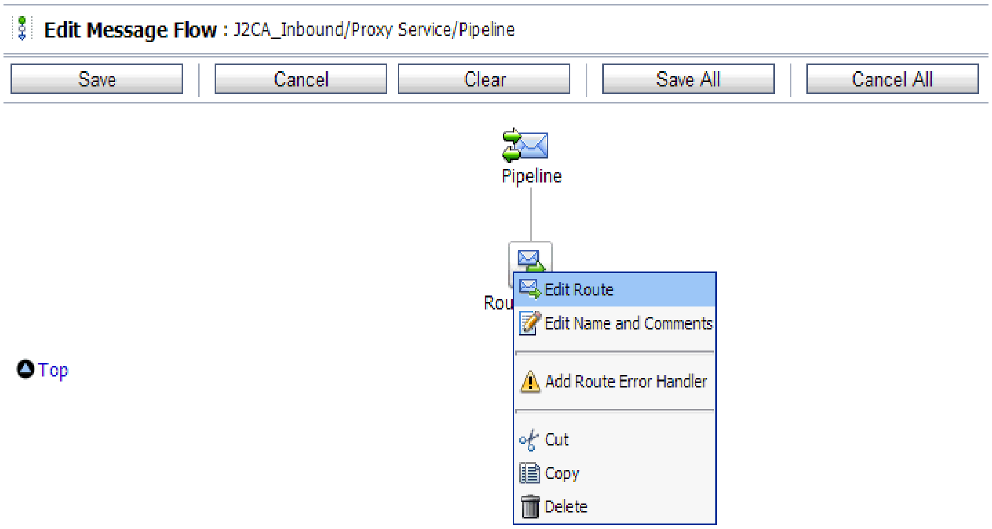
Figure 7–63 Add Route Option


ORACLE Service Bus Console 12c
Shared Variables
Map of Message Flow
Pipeline
Edit Message Flow : J2CA_Inbound/Proxy Service/Pipeline
Save
Cancel
Clear
Save All
Cancel All
Top
Pip
Add Pipeline Pair
Add Route
Add Conditional Branch
Add Operational Branch
Add Service Error Handler
Paste Route
Save
Cancel
Clear
Save All
Cancel All
Oracle Service Bus 12c Copyright © 2004, 2013, Oracle and/or its affiliates. All rights reserved.

The RouteNode1 icon is added.

11. Click the RouteNode1 icon and select **Edit Route** from the menu, as shown in Figure 7–64.

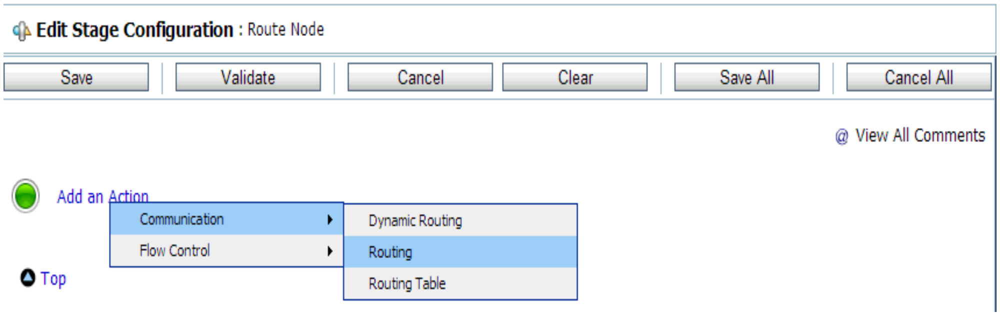
Figure 7–64 Edit Route Option



The Edit Stage Configuration workspace area is displayed.

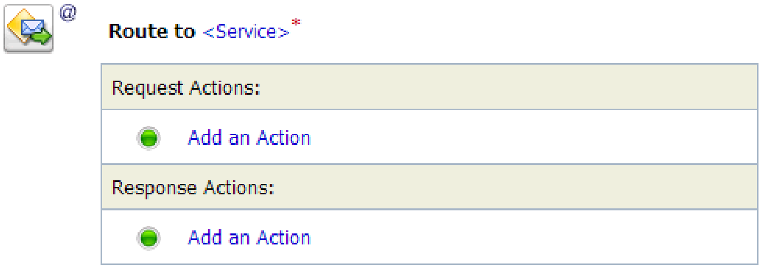
12. Click **Add an Action**, select **Communication** from the menu, and then click **Routing**, as shown in Figure 7–65.

Figure 7–65 Edit Stage Configuration Workspace



13. Click **<Service>**, as shown in Figure 7–66.

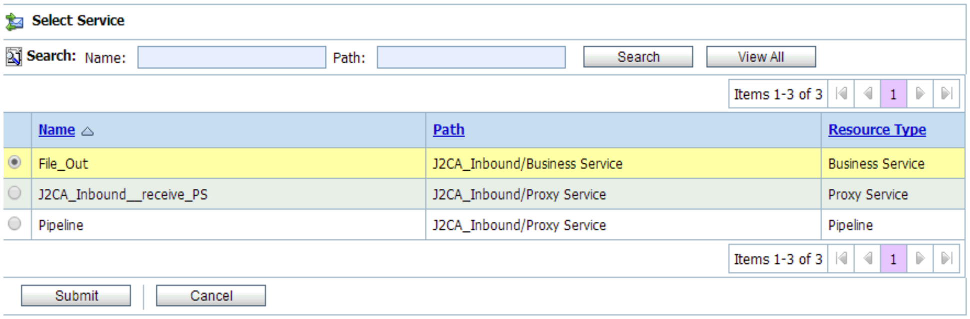
Figure 7–66 Service Route Actions



The Select Service dialog is displayed.

14. Select the **File_Out** Business service and click **Submit** as shown in Figure 7–67.

Figure 7–67 Select Service Dialog

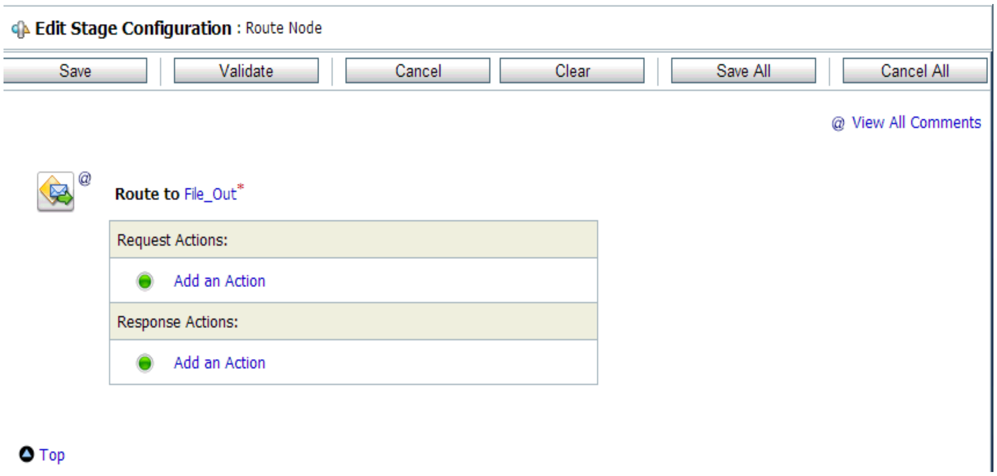


	Name	Path	Resource Type
●	File_Out	J2CA_Inbound/Business Service	Business Service
○	J2CA_Inbound__receive_PS	J2CA_Inbound/Proxy Service	Proxy Service
○	Pipeline	J2CA_Inbound/Proxy Service	Pipeline

You are returned to the Edit Stage Configuration workspace area.

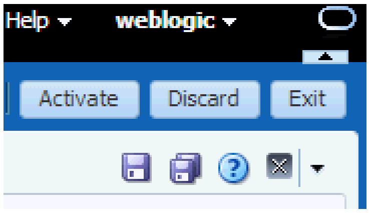
15. Click Save All, as shown inFigure 7–68.

Figure 7–68 Edit Stage Configuration Workspace Area



16. Click **Activate** in the right pane of the Oracle Service Bus session, as shown in Figure 7–69.

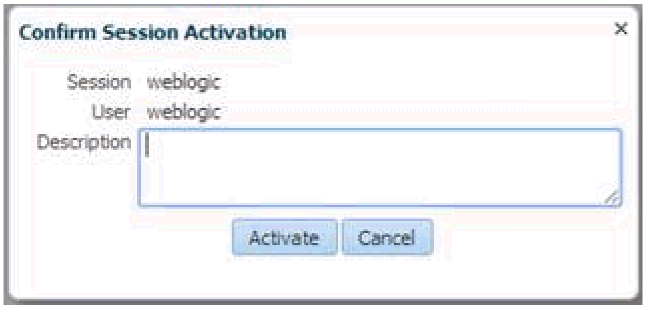
Figure 7–69 Activate Button



The Confirm Session Activation window appears.

17. Click **Activate** to save the changes, as shown in Figure 7–70.

Figure 7–70 Confirm Session Activation Window



Confirm Session Activation

Session weblogic

User weblogic

Description

18. Trigger an event from the Siebel system and check if the output is received in the configured output location.

    For more information on triggering an event, see Section 4.5.5, "Triggering an Event in Siebel" on page 4-49.

7.4 Configuring an Outbound Process Using sbconsole (BSE Configuration)

This section describes how to configure an outbound process using sbconsole for BSE configurations.

A sample project has been provided for this outbound use case scenario in the following folder of the Application Adapters installation:

```
<ADAPTER_HOME>\etc\sample\SIEBEL_Samples.zip\SIEBEL_Samples\OSB\BSE\Siebel_Sample_
BSE_OSB_Outbound_Project
```

This section includes the following topics:

- Section 7.4.1, "Starting Oracle Service Bus and Creating Project Folders"
- Section 7.4.2, "Setting the Class Path for Application Explorer to Integrate With Oracle Service Bus"
- Section 7.4.3, "Publishing a WSDL From Application Explorer to Oracle Service Bus"
- Section 7.4.4, "Configuring a File Type Business Service"
- Section 7.4.5, "Configuring a WSDL-based Business Service"
- Section 7.4.6, "Configuring a Pipeline With Proxy Service"

7.4.1 Starting Oracle Service Bus and Creating Project Folders

For more information on starting Oracle Service Bus and creating project folders, see Section 7.2.1, "Starting Oracle Service Bus and Creating Project Folders" on page 7-2.

7.4.2 Setting the Class Path for Application Explorer to Integrate With Oracle Service Bus

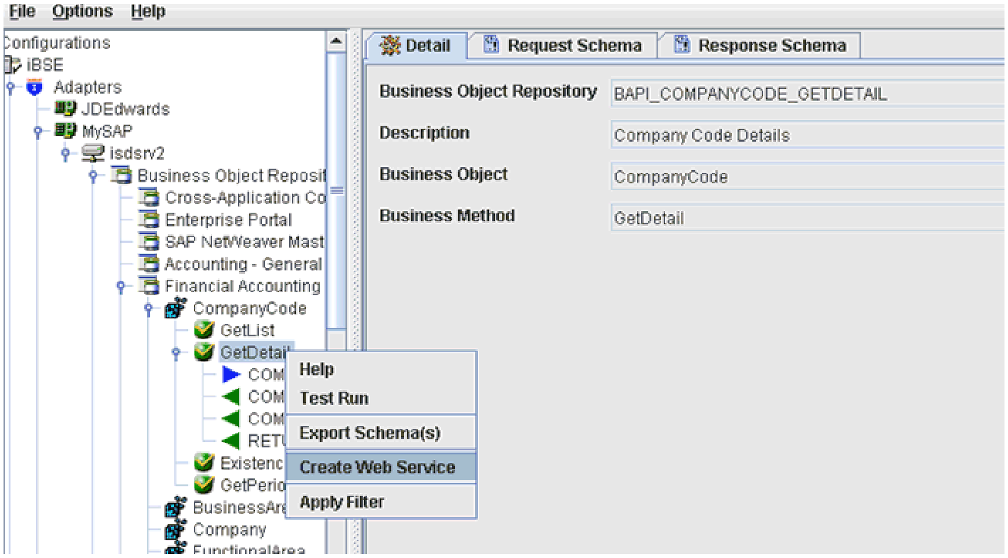
For more information on setting the class path for Application Explorer to integrate with Oracle Service Bus, see Section 7.2.2, "Setting the Class Path for Application Explorer to Integrate With Oracle Service Bus" on page 7-6.

7.4.3 Publishing a WSDL From Application Explorer to Oracle Service Bus

This section describes how to publish a WSDL from Application Explorer (BSE configuration) to Oracle Service Bus.

1. Start Application Explorer, connect to a BSE configuration, and connect to a Siebel target.

Figure 7–71 Create Web Service Option



Business Object Repository: BAPI_COMPANYCODE_GETDETAIL

Description: Company Code Details

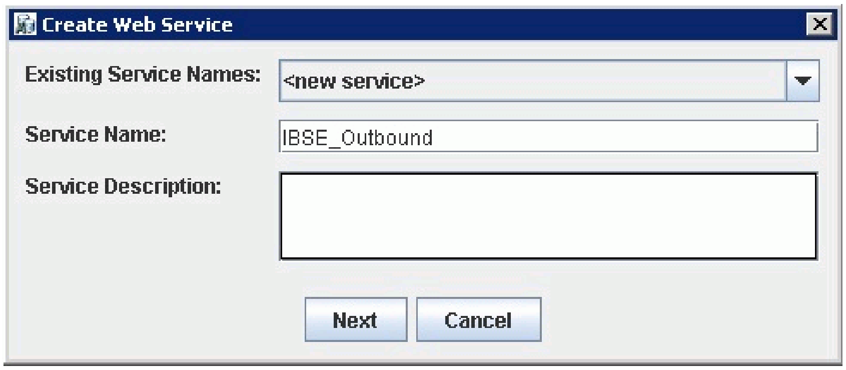
Business Object: CompanyCode

Business Method: GetDetail

2. Expand the **Business Object Repository** node, **Financial Accounting** node, and the **CompanyCode** business object.
3. Right-click the **GetDetail** method and select **Create Web Service** from the menu, as shown in Figure 7–71.

   The Create Web Service dialog is displayed, as shown in Figure 7–72.

Figure 7–72 Create Web Service Dialog



Existing Service Names: <new service>

Service Name: IBSE_Outbound

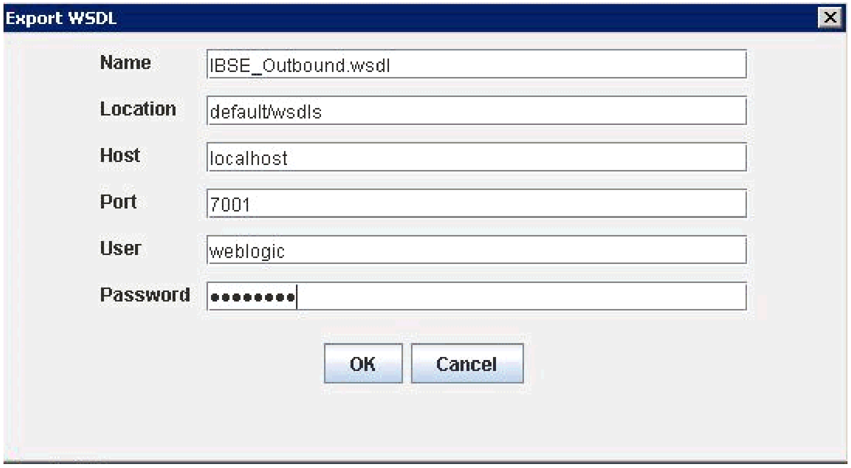
Service Description:

4. Enter a service name and click **Next**.
5. Click **OK** on the next dialog that is displayed.

   Application Explorer switches the view to the Business Services node, and the new Web service appears in the left pane.
6. Right-click the new Web service and select **Export WSDL** from the menu.

The Export WSDL dialog is displayed, as shown in Figure 7–73.

Figure 7–73 Export WSDL Dialog



Export WSDL

Name: IBSE_Outbound.wsdl

Location: default/wsdls

Host: localhost

Port: 7001

User: weblogic

Password: ••••••••

7. In the Name field, a default file name for the WSDL file is provided. You can accept the default or provide your own.
8. In the Location field, enter the location where you want to publish the WSDL document.

   The location is composed of an Oracle Service Bus project name and optionally, one or more folder names. The project name and any folder names must be separated by a forward slash character "/".
9. In the Host field, enter the name of the machine where Oracle WebLogic Server is running.
10. In the Port field, enter the port for the domain you are using.
11. In the User field, enter your username to access Oracle Service Bus.
12. In the Password field, enter your password to access Oracle Service Bus.
13. Click **OK**.

    The WSDL is published to the location specified in the Export WSDL dialog and is now available for use with a Business Service or Proxy Service in Oracle Service Bus.

7.4.4 Configuring a File Type Business Service

For more information on configuring a file type business service, see Section 7.2.5, "Configuring a File Type Business Service" on page 7-9.

7.4.5 Configuring a WSDL-based Business Service

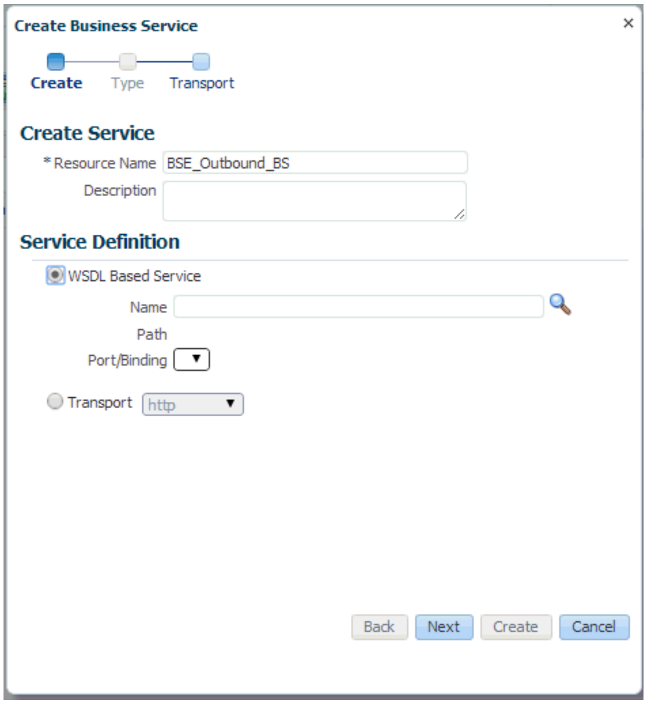
This section describes how to configure a WSDL type Business Service using the Oracle Service Bus Console.

Perform the following steps to configure a WSDL-based Proxy Service:

1. Right-click on the Business Service folder in the left pane and select **Business Service**.

   The Create Business Service window is displayed, as shown in Figure 7–74.

Figure 7–74 Create Business Service Window



Create Business Service

Create Service

* Resource Name: BSE_Outbound_BS

Description:

Service Definition

● WSDL Based Service

Name:

Path:

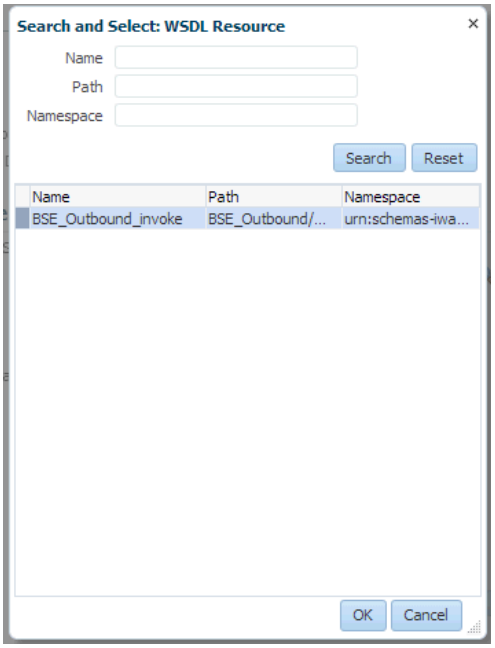
Port/Binding:

○ Transport: http

2. Provide a name for the Business Service, and in Service Definition area, select the WSDL Based Service option and click the search icon.

   The Search and Select: WSDL Resource window is displayed, as shown in Figure 7–75.

Figure 7–75 Search and Select: WSDL Resource Window



Search and Select: WSDL Resource

Name:

Path:

Namespace:

Name	Path	Namespace
BSE_Outbound_invoke	BSE_Outbound/...	urn:schemas-iwa...

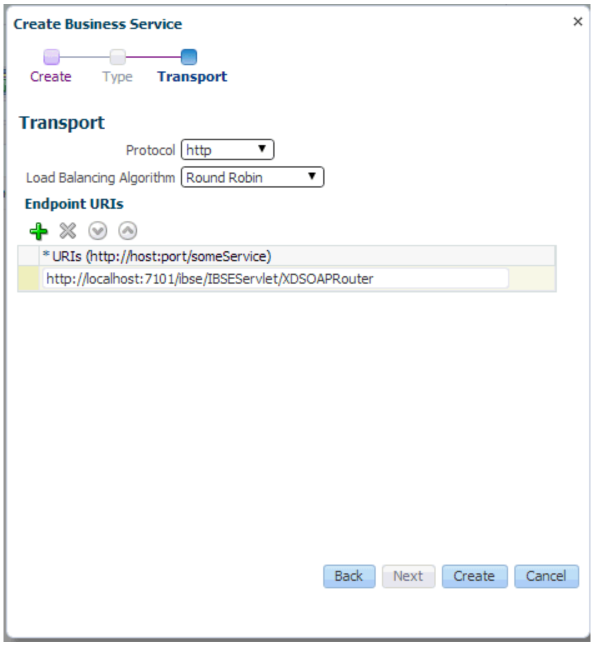
3. Click the **Search** button, select the BSE Outbound WSDL, and click **OK**.

   You are returned to the Create Business Service window.

4. Click **Next**.

5. Accept the default values and click the **Create** button, as shown in Figure 7–76.

Figure 7–76 Create Business Service Window



Transport

Protocol http

Load Balancing Algorithm Round Robin

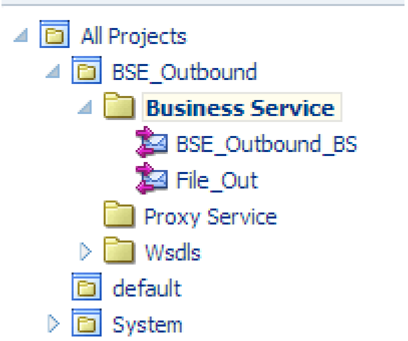
Endpoint URIs

* URIs (http://host:port/someService)

http://localhost:7101/ibse/IBSEServlet/XDSOAPRouter

The created WSDL-based Business Service is listed under the Business Service folder, as shown in Figure 7–77.

Figure 7–77 WSDL-based Business Service



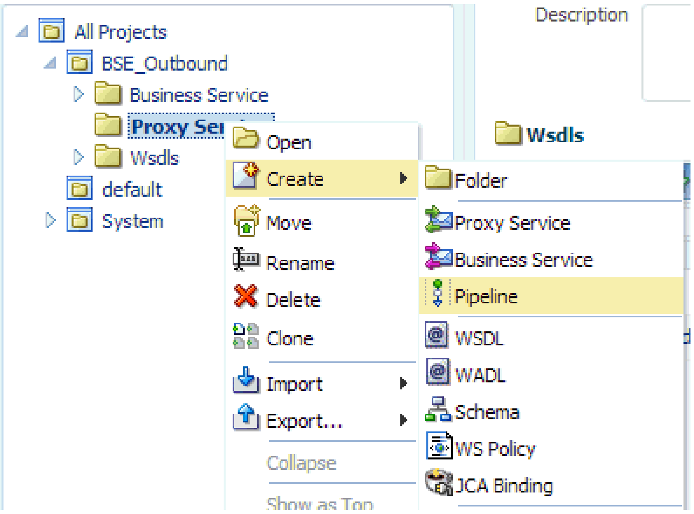
- All Projects
  - BSE_Outbound
    - Business Service
      - BSE_Outbound_BS
      - File_Out
    - Proxy Service
    - Wsdls
  - default
  - System

7.4.6 Configuring a Pipeline With Proxy Service

This section describes how to configure a Proxy Service using the Oracle Service Bus Console.

1. Right-click the Proxy Service folder, select **Create** and click **Pipeline**, as shown in Figure 7–78.

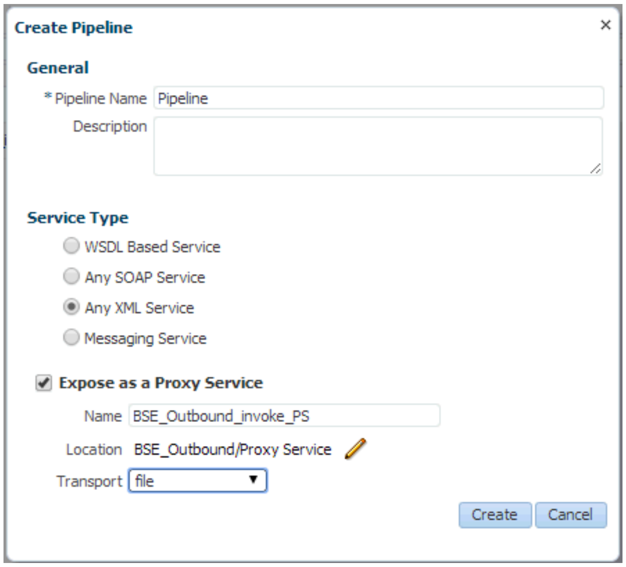
Figure 7–78 Pipeline Option



The Create Pipeline window is displayed.

2. Enter a name in the Pipeline Name field. By default, **Expose as a Proxy Service** is selected. If you wish to change the Proxy Service Name, change it and set Transport to **file**, and click **Create** as shown in Figure 7–79.

Figure 7–79 Create Pipeline Window



Create Pipeline ×

General

* Pipeline Name: Pipeline

Description:

Service Type

○ WSDL Based Service
○ Any SOAP Service
● Any XML Service
○ Messaging Service

☑ Expose as a Proxy Service

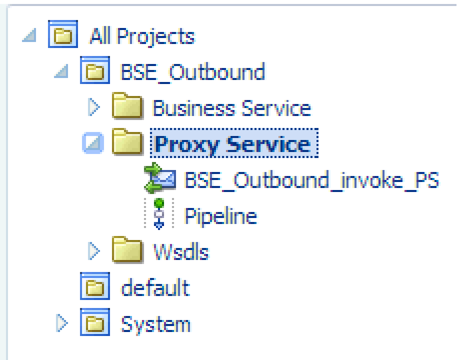
Name: BSE_Outbound_invoke_PS

Location: BSE_Outbound/Proxy Service

Transport: file

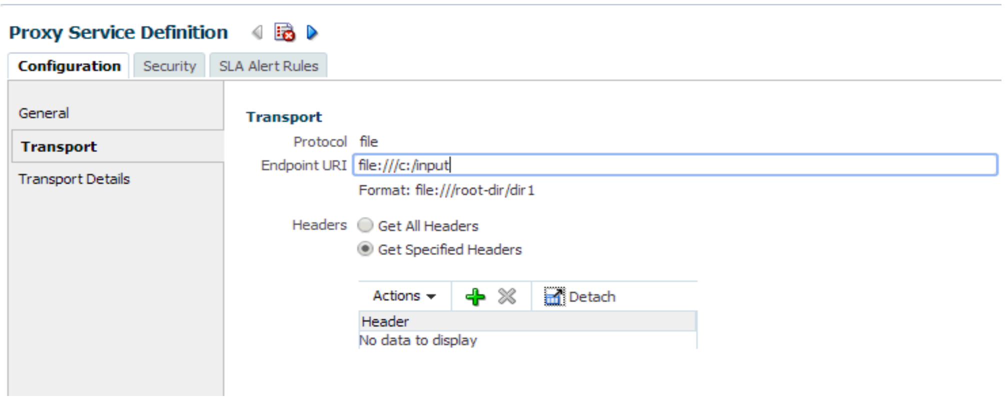
The created Pipeline and the Proxy Service is listed under Proxy Service, as shown in Figure 7–80.

Figure 7–80 Pipeline Node



3. Double-click the created proxy service and click **Transport** in the left pane. Provide the input location in the Endpoint URI field, as shown in Figure 7–81.

Figure 7–81 Transport



Transport

Protocol file

Endpoint URI file:///c:/input

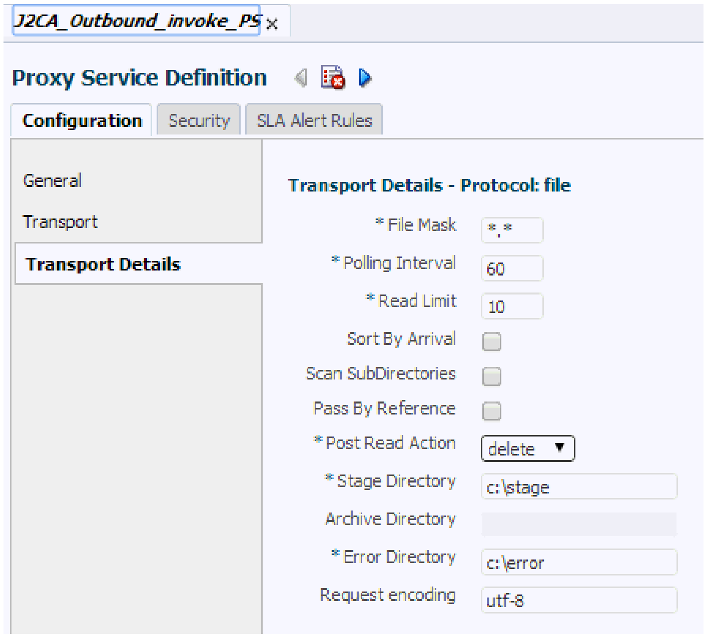
Format: file:///root-dir/dir1

Headers ○ Get All Headers
● Get Specified Headers

No data to display

4. Click **Transport Details** in the left pane and provide the location for the Stage Directory and the Error Directory fields, as shown in Figure 7–82.

Figure 7–82 Transport Details

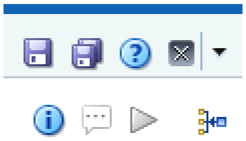


Transport Details - Protocol: file

* File Mask	*.*
* Polling Interval	60
* Read Limit	10
Sort By Arrival	□
Scan SubDirectories	□
Pass By Reference	□
* Post Read Action	delete
* Stage Directory	c:\stage
Archive Directory	
* Error Directory	c:\error
Request encoding	utf-8

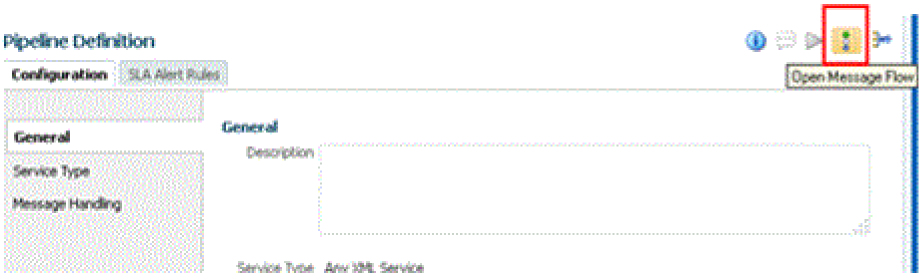
5. Click the **Save All** icon in the right corner, as shown in Figure 7–83.

Figure 7–83 Save All Icon



6. Double-click the **Pipeline** node and click the **Open Message Flow** icon on the right pane to open the message flow, as shown in Figure 7–84.

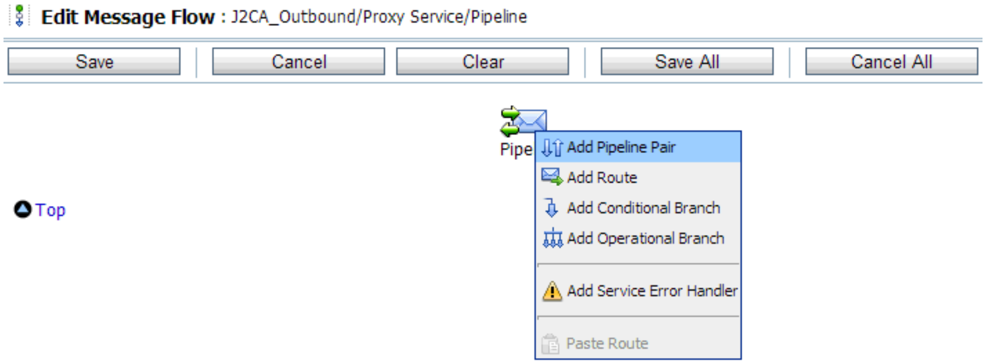
Figure 7–84 Open Message Flow Icon



Service Type Any XML Service

7. Click the Proxy Service icon and select **Add Pipeline Pair** from the menu, as shown in Figure 7–85.

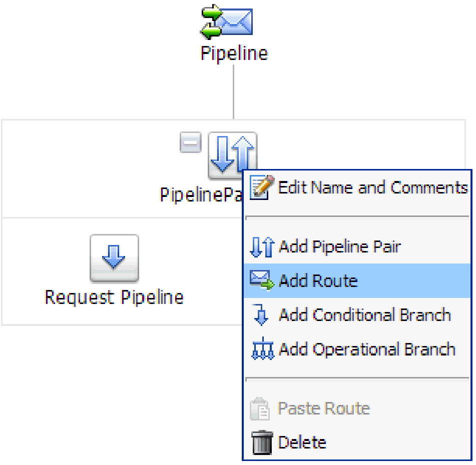
Figure 7–85 Add Pipeline Pair Option



Edit Message Flow : J2CA_Outbound/Proxy Service/Pipeline

8. Click the **PipelinePairNode1** icon and select **Add Route** from the menu, as shown in Figure 7–86.

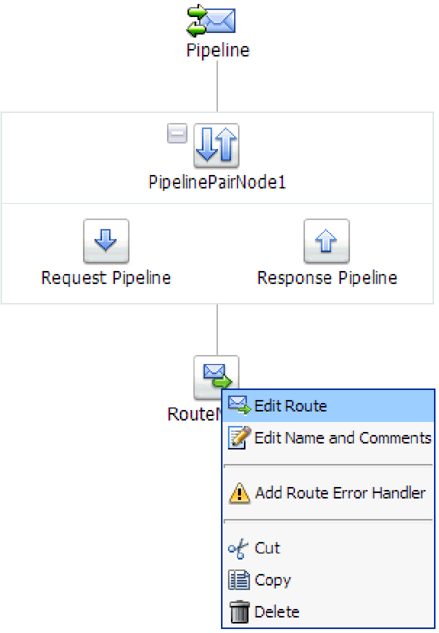
Figure 7–86 Add Route Option



The RouteNode1 icon is added below the PipelinePairNode1 icon.

9. Click the RouteNode1 icon and select **Edit Route** from the menu, as shown in Figure 7–87.

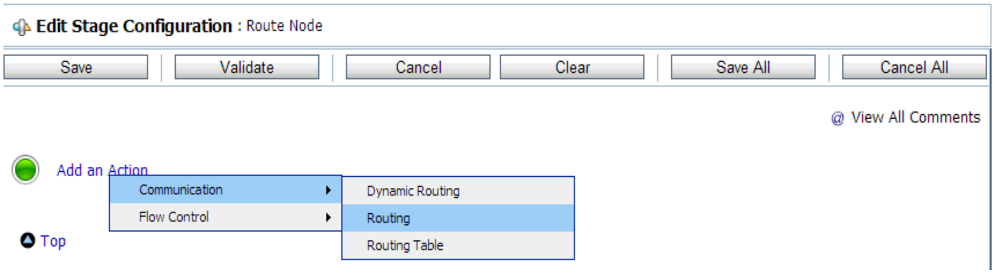
Figure 7–87 Edit Route Option



The Edit Stage Configuration workspace area is displayed.

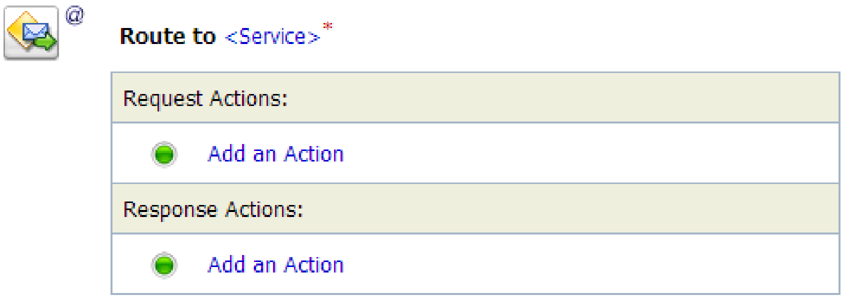
10. Click **Add an Action**, select **Communication** and click **Routing**, as shown in Figure 7–88.

Figure 7–88 Edit Stage Configuration Workspace Area



11. Click **<Service>**, as shown in Figure 7–89.

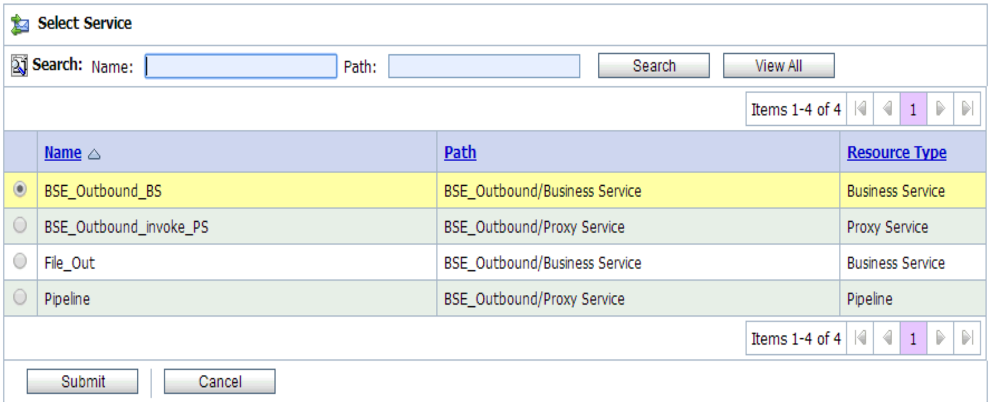
Figure 7–89 Actions



The Select Service dialog is displayed.

12. Select the WSDL type Business Service configured for Siebel and click on **Submit**, as shown in Figure 7–90.

Figure 7–90 Select Service Dialog

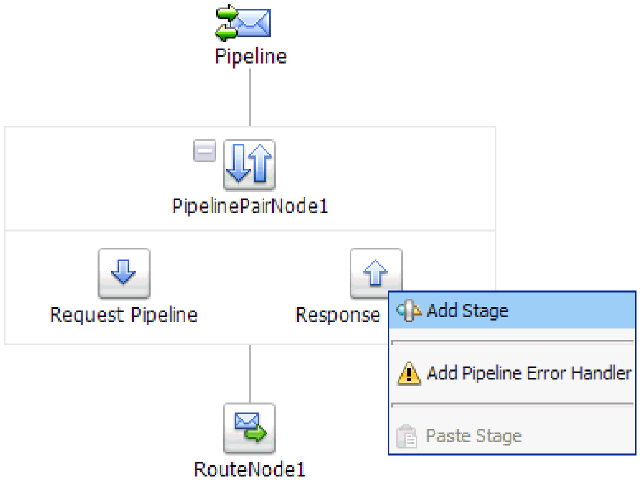


	Name △	Path	Resource Type
●	BSE_Outbound_BS	BSE_Outbound/Business Service	Business Service
○	BSE_Outbound_invoke_PS	BSE_Outbound/Proxy Service	Proxy Service
○	File_Out	BSE_Outbound/Business Service	Business Service
○	Pipeline	BSE_Outbound/Proxy Service	Pipeline

13. Select the name of the Siebel business object (for example, queryWithView) as the operational attribute from the list, and click **Save**.

14. Click the Response Pipeline icon and select **Add Stage** from the menu, as shown in Figure 7–91.

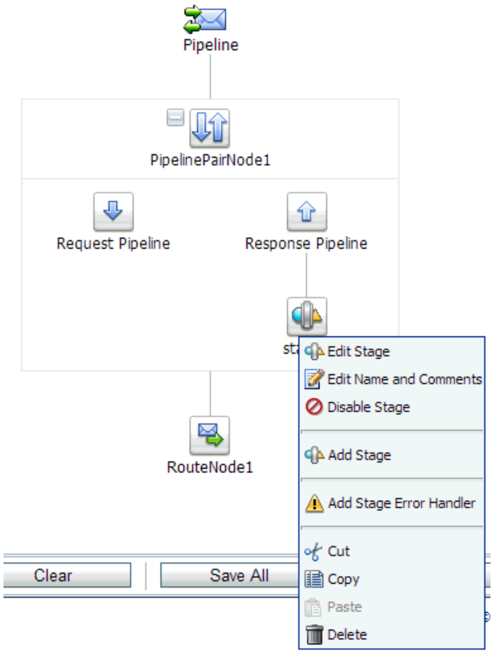
Figure 7–91 Response Pipeline Icon



The Stage1 icon is added below the Response Pipeline icon.

15. Click the Stage1 icon and select **Edit Stage** from the menu, as shown in Figure 7–92.

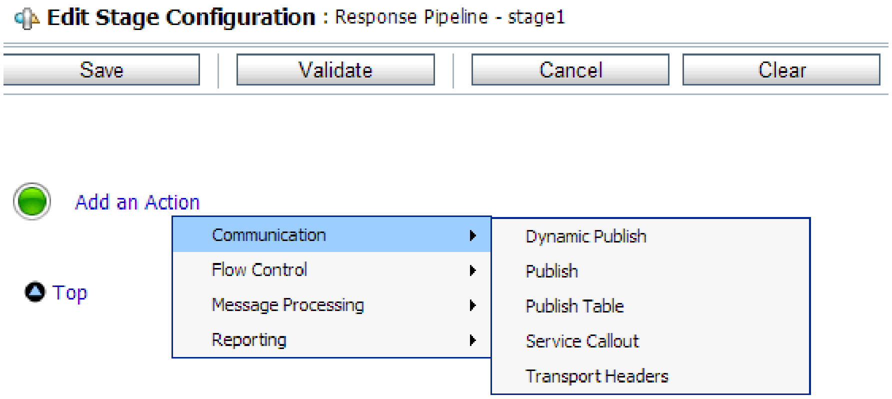
Figure 7–92 Edit Stage Option



The Edit Stage Configuration workspace area is displayed.

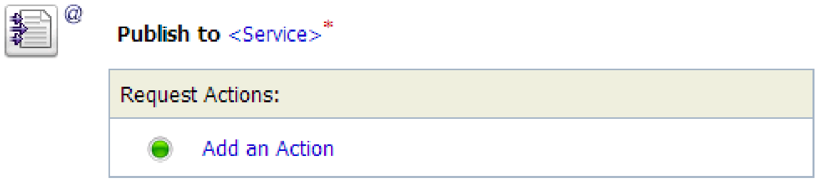
16. Click **Add an Action**, select **Communication**, and then click **Publish**, as shown in Figure 7–93.

Figure 7–93 Edit Stage Configuration Workspace Area



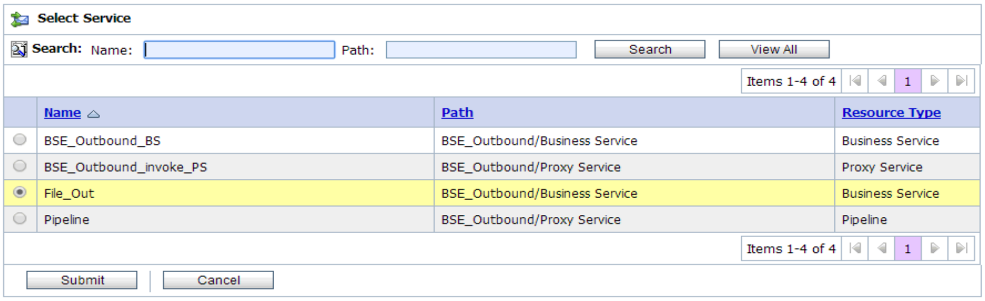
17. Click **<Service>**, as shown in Figure 7–94.

Figure 7–94 <Service> Action



18. In the Select Service dialog, select a File type Business Service and click **Submit**, as shown in Figure 7–95.

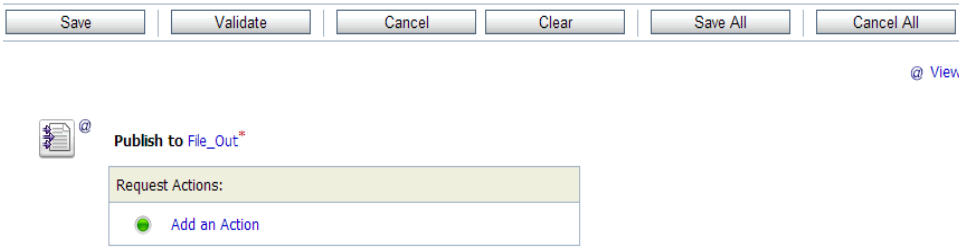
Figure 7–95 Select Service Dialog



	Name	Path	Resource Type
○	BSE_Outbound_BS	BSE_Outbound/Business Service	Business Service
○	BSE_Outbound_invoke_PS	BSE_Outbound/Proxy Service	Proxy Service
●	File_Out	BSE_Outbound/Business Service	Business Service
○	Pipeline	BSE_Outbound/Proxy Service	Pipeline

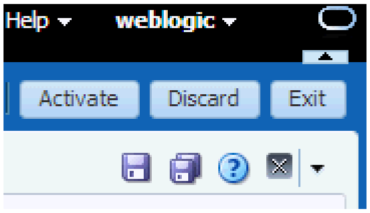
19. Click **Save All**, as shown in Figure 7–96.

Figure 7–96 Save All Button



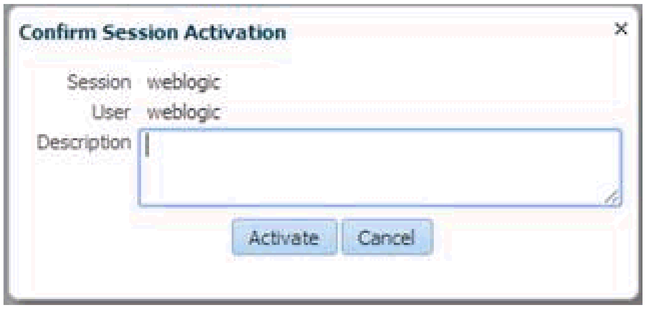
20. Click **Activate** in the right pane of the Oracle Service Bus session, as shown in Figure 7–97.

Figure 7–97 Activate Button



21. Click **Activate** to save the changes, as shown in Figure 7–98.

Figure 7–98 Confirm Session Activation



Session weblogic
User weblogic
Description

22. Copy and paste an input XML file in the input folder you have configured (for example, C:\input).

    Output is received in the configured output location (for example, C:\output).

7.5 Configuring JMS Proxy Services Using Oracle Service Bus (J2CA Configuration)

This section describes how to configure JMS Proxy Services using Oracle Service Bus for a J2CA configuration.

1. Start Oracle Service Bus and create the required project folder.

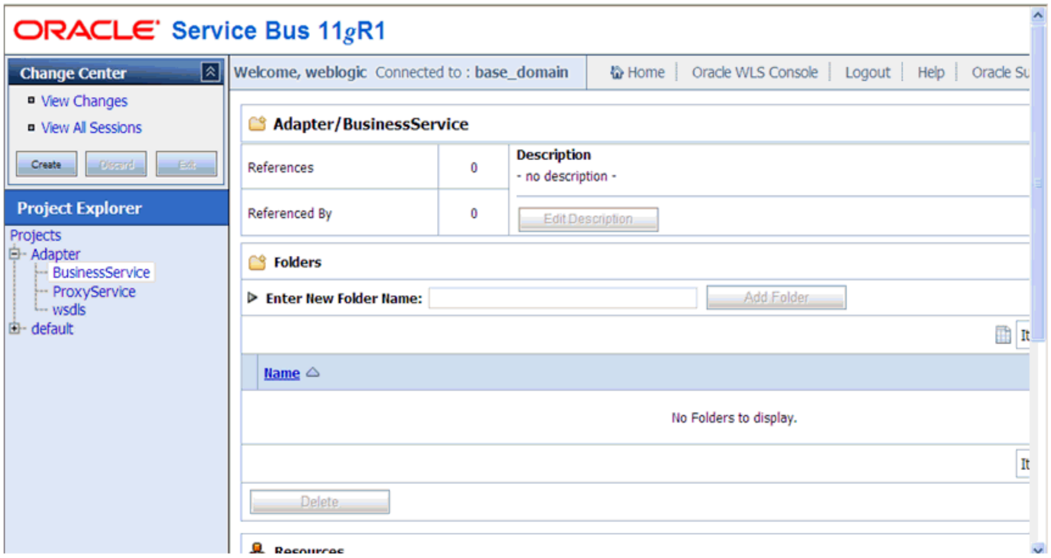
   For more information, see Section 7.2.1, "Starting Oracle Service Bus and Creating Project Folders".

2. Generate and publish the WSDL from Application Explorer to the created project folder. Using the published WSDL, create a Business Service.

   For more information, see Section 7.2.3, "Publishing a WSDL From Application Explorer to Oracle Service Bus".

3. Open the Service Bus Console page, as shown in Figure 7–99.

Figure 7–99 Service Bus Console



Connected to : base_domain

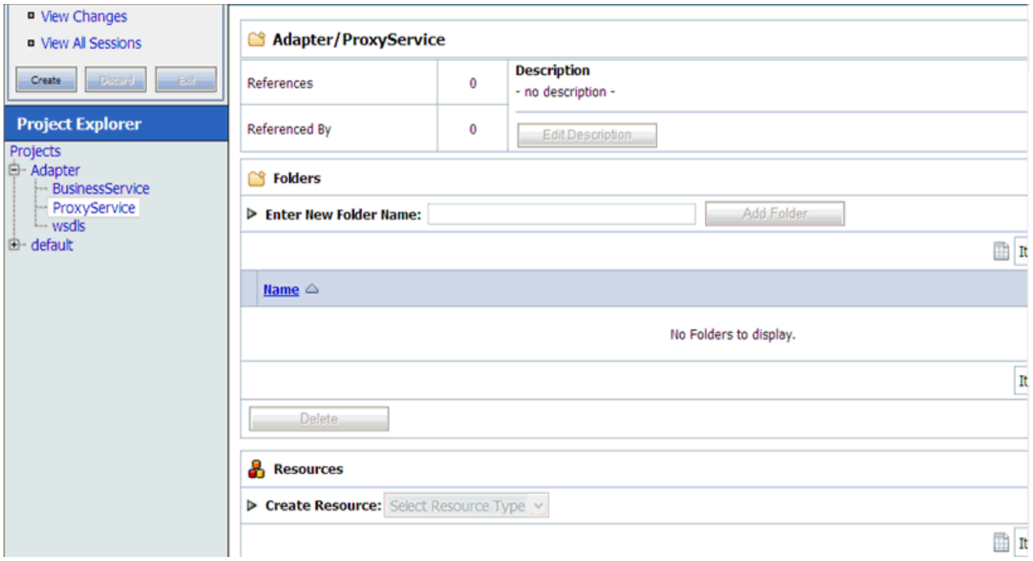
References	0	Description - no description -
Referenced By	0	

Enter New Folder Name:

No Folders to display.

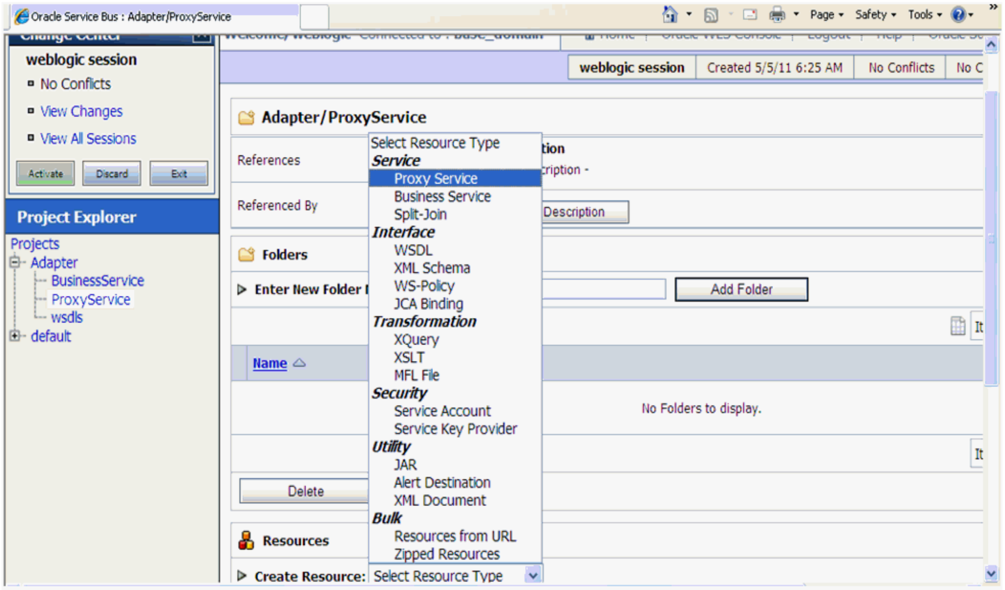
4. Select the ProxyService project folder in the left pane, and click **Create**, as shown in Figure 7–100.

Figure 7–100 Proxy Service



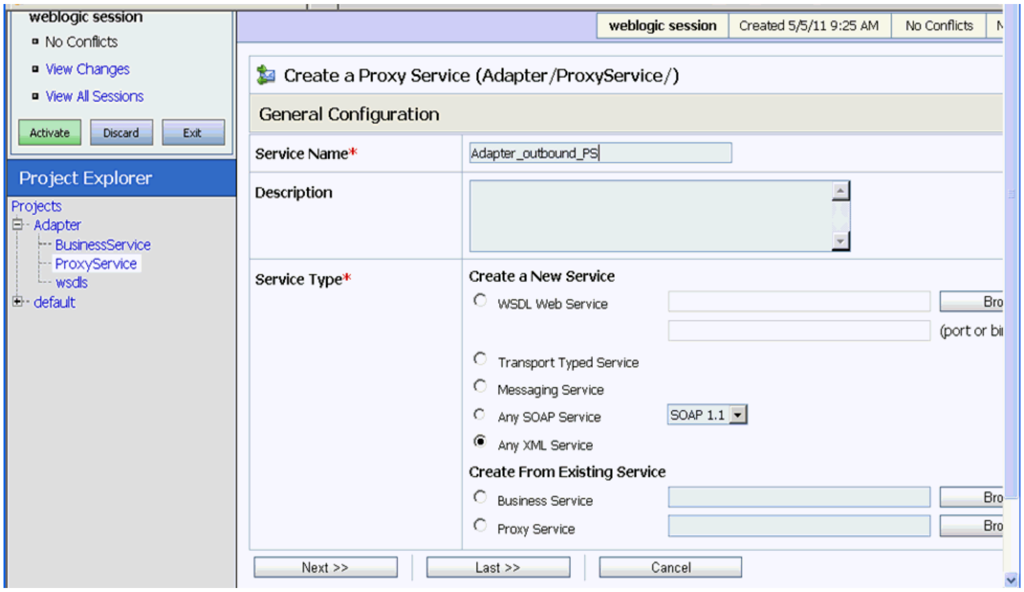
5. In the right pane, select **Proxy Service** from the Create Resource list, as shown in Figure 7–101.

Figure 7–101 Create Resource Menu



6. Enter an appropriate name in the **Service Name** field, as shown in Figure 7–102.

Figure 7–102 Service Name



Create a Proxy Service (Adapter/ProxyService/)

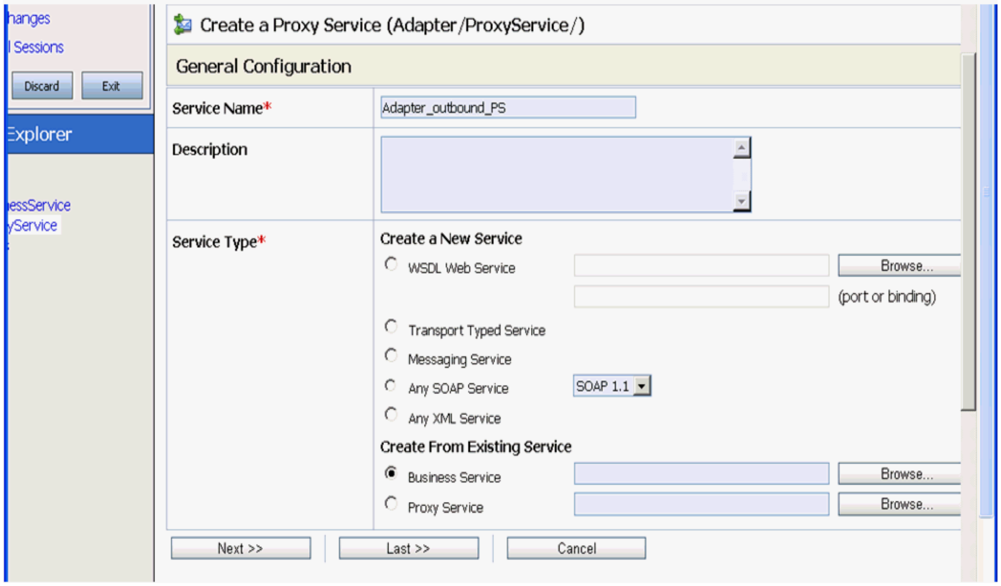
General Configuration

Service Name* Adapter_outbound_PS

Service Type* ● Any XML Service

7. In the Service Type section, under Create From Existing Service, select the **Business Service** radio button and click **Browse**, as shown in Figure 7–103.

Figure 7–103 Business Service



Create a Proxy Service (Adapter/ProxyService/)

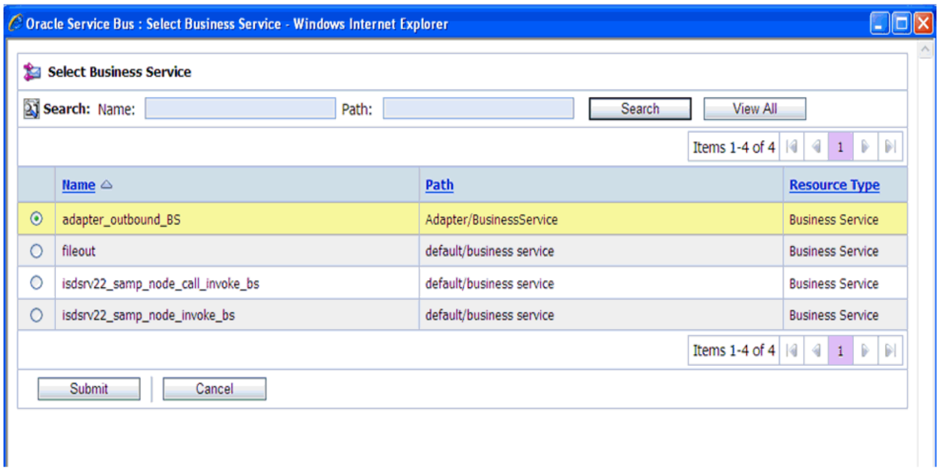
General Configuration

Service Name* Adapter_outbound_PS

Service Type* ● Business Service

8. Select the existing business service and click **Submit**, as shown in Figure 7–104.

Figure 7–104 Existing Business Service


Oracle Service Bus : Select Business Service - Windows Internet Explorer

Select Business Service

Search: Name: Path: Search View All

Items 1-4 of 4

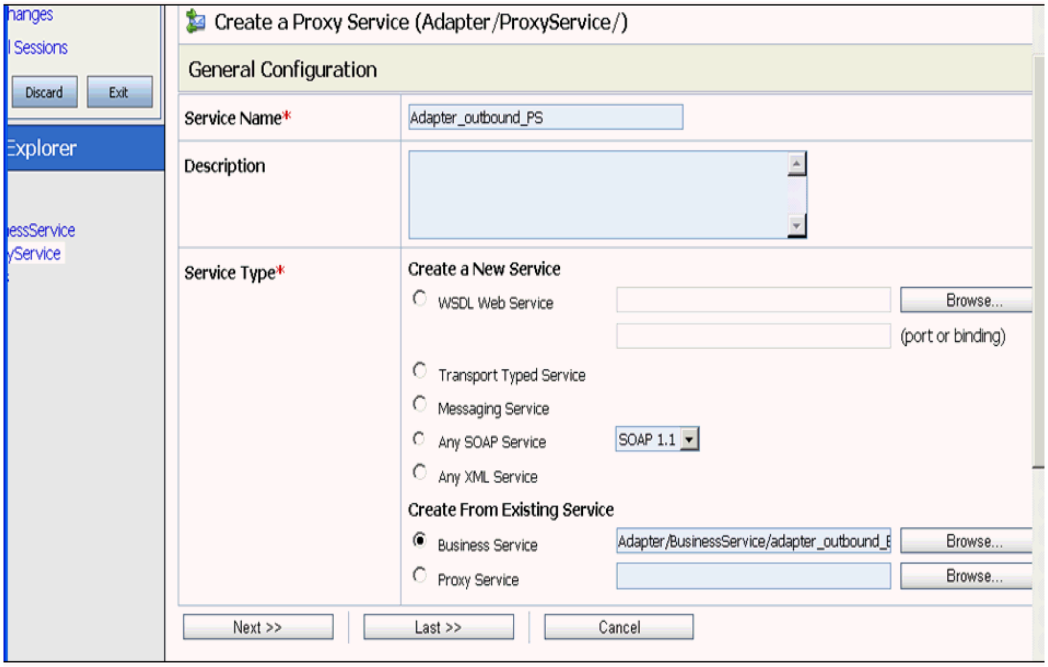
	Name	Path	Resource Type
●	adapter_outbound_BS	Adapter/BusinessService	Business Service
○	fileout	default/business service	Business Service
○	isdsrv22_samp_node_call_invoke_bs	default/business service	Business Service
○	isdsrv22_samp_node_invoke_bs	default/business service	Business Service

Items 1-4 of 4

Submit Cancel

9. Click **Next**, as shown in Figure 7–105.

Figure 7–105 Next



Create a Proxy Service (Adapter/ProxyService/)

General Configuration

Service Name*: Adapter_outbound_PS

Description:

Service Type*:

Create a New Service

○ WSDL Web Service Browse... (port or binding)

○ Transport Typed Service

○ Messaging Service

○ Any SOAP Service SOAP 1.1

○ Any XML Service

Create From Existing Service

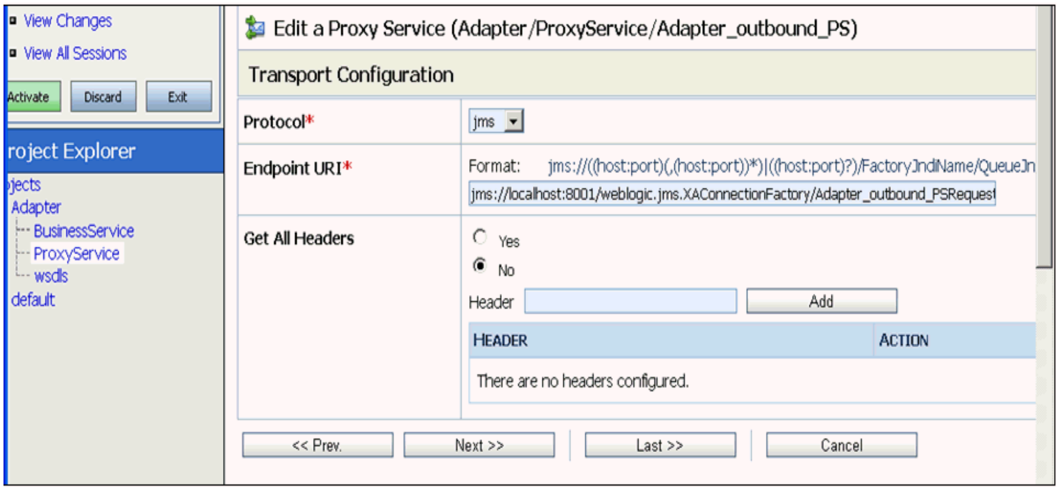
● Business Service Adapter/BusinessService/adapter_outbound_E Browse...

○ Proxy Service Browse...

Next >> Last >> Cancel

10. Select **jms** from the Protocol list and click **Next**, as shown in Figure 7–106.

Figure 7–106 Protocol List



Edit a Proxy Service (Adapter/ProxyService/Adapter_outbound_PS)

Transport Configuration

Protocol* jms

Endpoint URI* Format: jms://((host:port)(,(host:port))*)|((host:port)?)/FactoryJndiName/QueueJn

jms://localhost:8001/weblogic.jms.XAConnectionFactory/Adapter_outbound_PSRequest

Get All Headers ○ Yes ● No

Header

HEADER	ACTION
There are no headers configured.	

11. Provide the following parameters, as shown in Figure 7–107.

    a. Select **Queue** in the Destination Type section.

    b. Enable the **Is Response Required** check box.

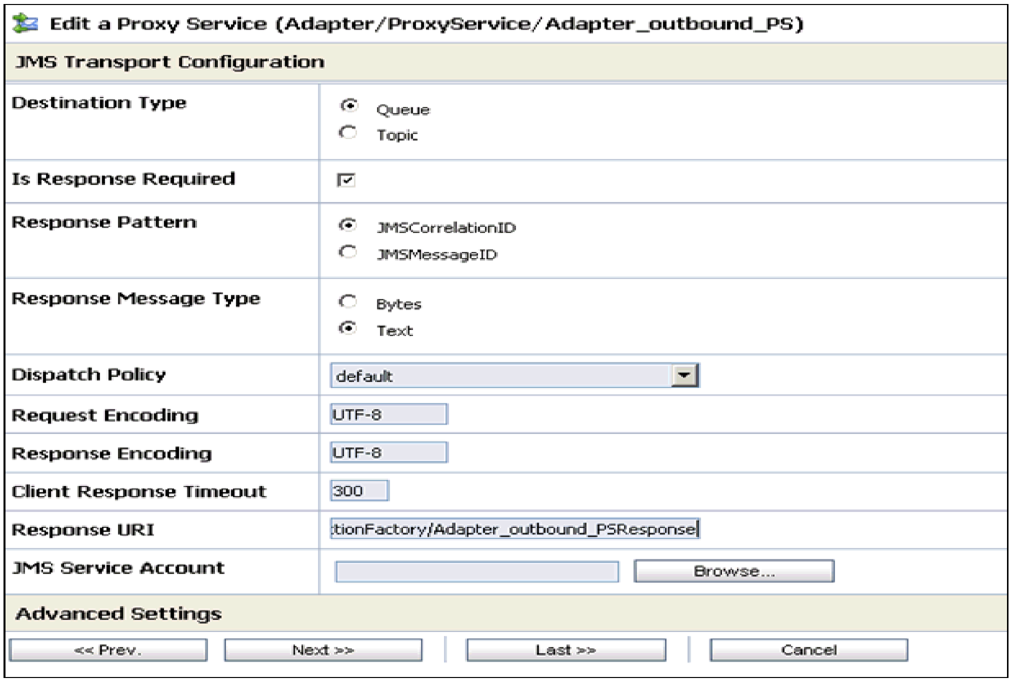
    c. Select **Text** in the Response Message Type section.

    d. In the Response URI field, provide the Endpoint URI used in the Transport Configuration and change `Request` to `Response`.

    For example:

    ```
    jms://localhost:8001/weblogic.jms.XAConnectionFactory/Adap
    ter_outbound_PSResponse
    ```

Figure 7–107 Edit a Proxy Service



Edit a Proxy Service (Adapter/ProxyService/Adapter_outbound_PS)

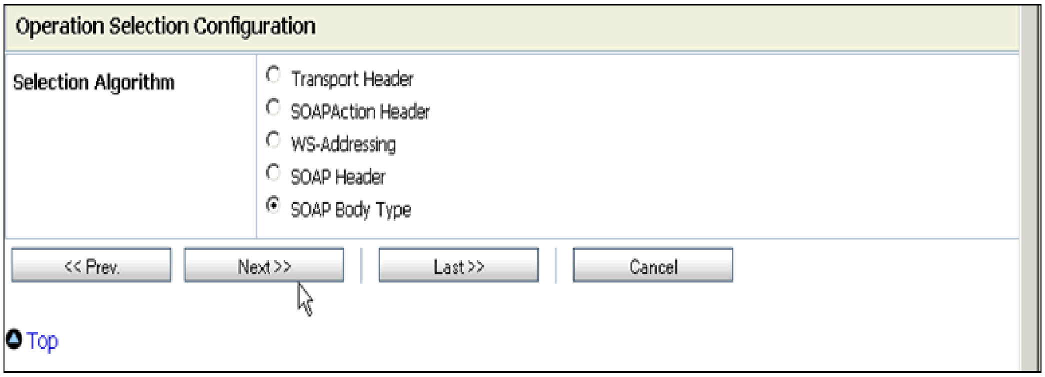
JMS Transport Configuration

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	

12. Click **Next**.

    The Operation Selection Configuration pane appears, as shown in Figure 7–108.

Figure 7–108 Operation Selection Configuration Pane

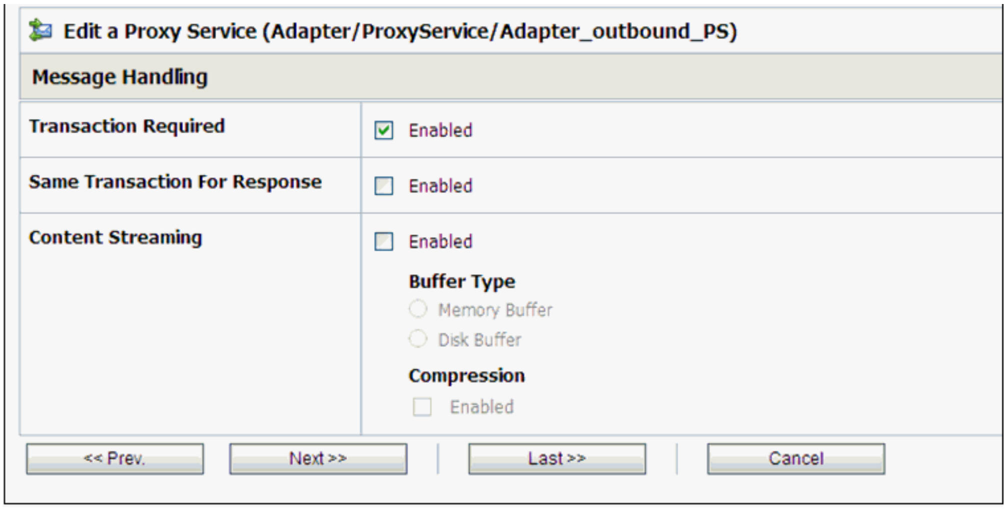


Operation Selection Configuration

Selection Algorithm	○ Transport Header ○ SOAPAction Header ○ WS-Addressing ○ SOAP Header ● SOAP Body Type

13. Ensure the **SOAP Body Type** is selected and click **Next**.
14. Enable the **Transaction Required** box and click **Next**, as shown in Figure 7–109.

Figure 7–109 Message Handling



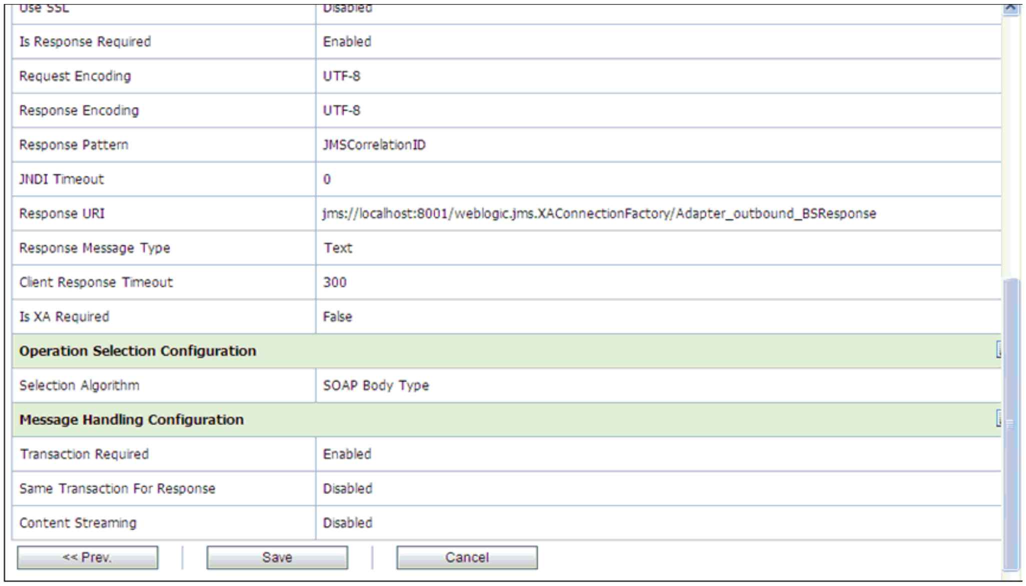
Edit a Proxy Service (Adapter/ProxyService/Adapter_outbound_PS)

Message Handling

Transaction Required	☑ Enabled
Same Transaction For Response	□ Enabled
Content Streaming	□ Enabled **Buffer Type** ○ Memory Buffer ○ Disk Buffer **Compression** □ Enabled

15. Click **Save**, as shown in Figure 7–110.

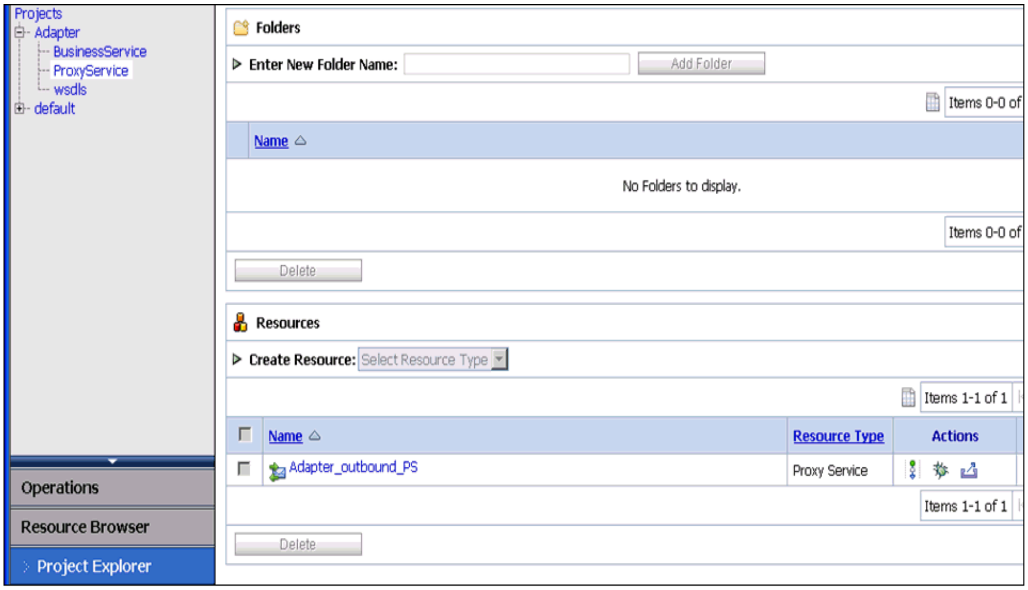
Figure 7–110 Save



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

The created Proxy Service is saved, as shown in Figure 7–111.

Figure 7–111 Proxy Service

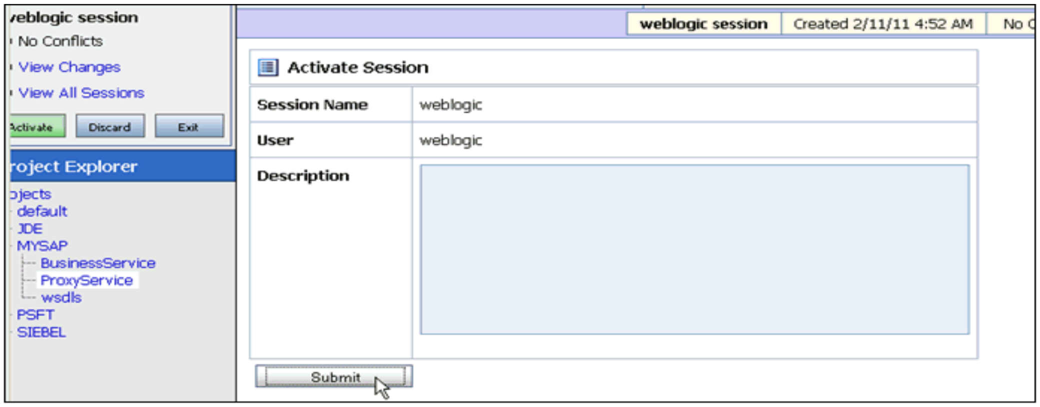


No Folders to display.

□	Name	Resource Type	Actions
□	Adapter_outbound_PS	Proxy Service	

16. In the left pane, click **Activate**, and then **Submit**, as shown in Figure 7–112.

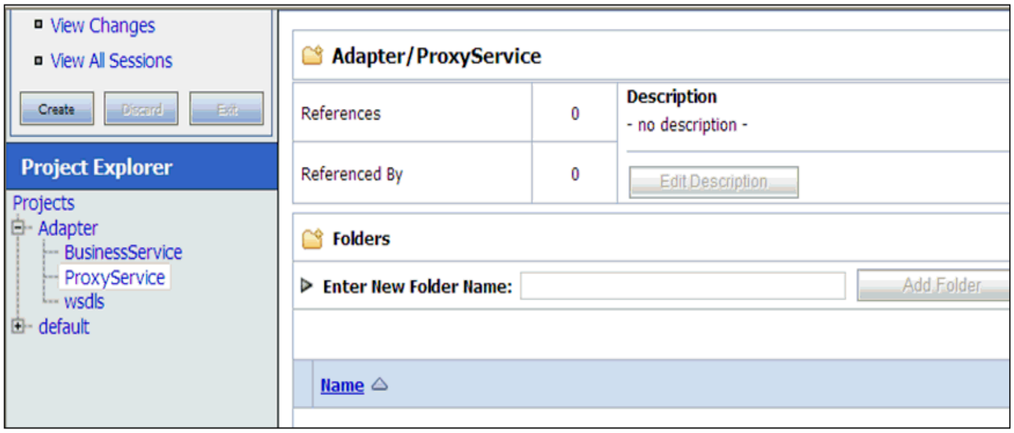
Figure 7–112 Activate Session



Session Name	weblogic
User	weblogic
Description	

17. In the left pane, click **ProxyService** under the Projects folder, as shown in Figure 7–113.

Figure 7–113 Adapter/ProxyService

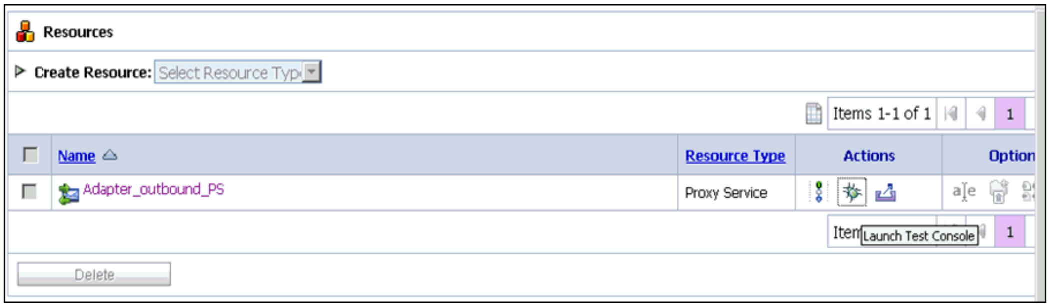


References	0	Description - no description -
Referenced By	0	

Enter New Folder Name:

18. Click the **Launch Test Console** icon for the created Proxy Service, as shown in Figure 7–114.

Figure 7–114 Launch Test Console Icon



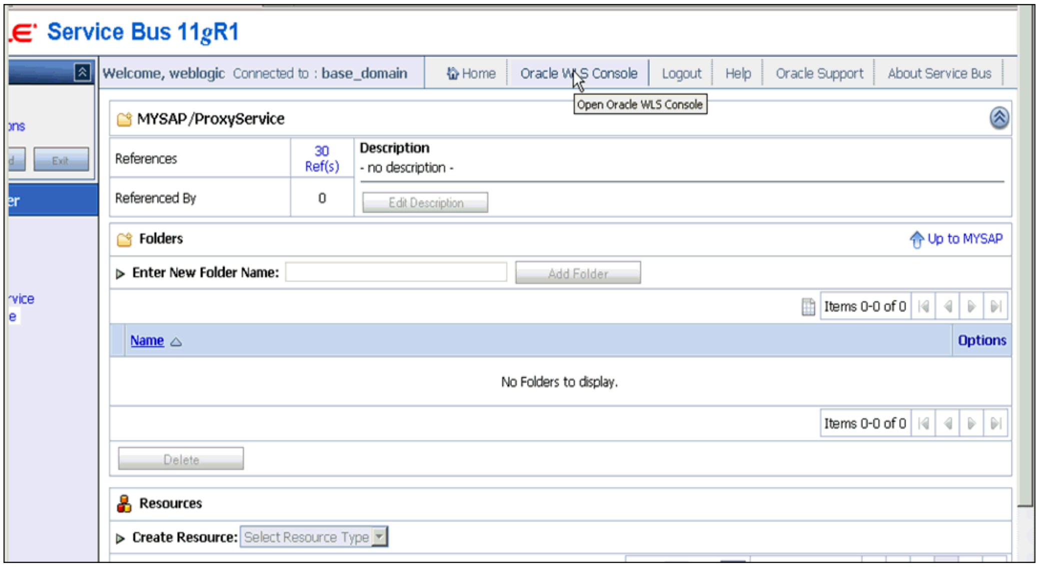
Name	Resource Type	Actions	Option
Adapter_outbound_PS	Proxy Service		

19. Provide the input values for **Payload**, uncheck the **Direct Call** box, and click **Execute**.

20. Review the Response document, and then click **Close**.

21. Click the **Oracle WLS Console** tab, as shown in Figure 7–115.

Figure 7–115 ProxyService



MYSAP/ProxyService

References	30 Ref(s)	Description - no description -
Referenced By	0	

Folders

Enter New Folder Name:

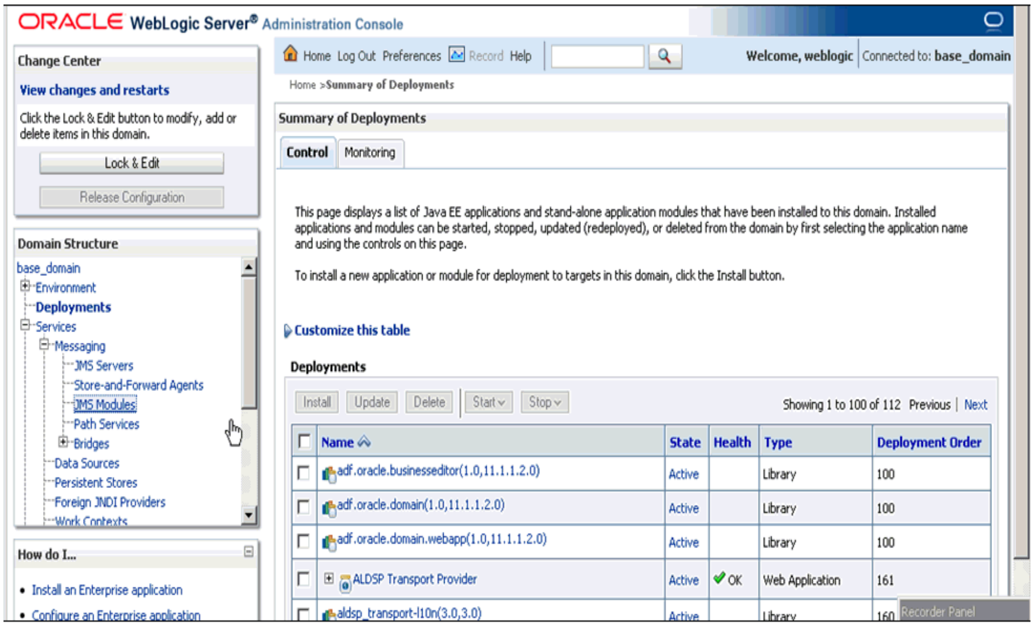
No Folders to display.

Resources

Create Resource: Select Resource Type

22. In the Oracle WLS Console, expand **Services**, expand **Messaging**, and click **JMS Modules**, as shown in Figure 7–116.

Figure 7–116 Oracle WLS Console



Click the Lock & Edit button to modify, add or delete items in this domain.

Summary of Deployments

This page displays a list of Java EE applications and stand-alone application modules that have been installed to this domain. Installed applications and modules can be started, stopped, updated (redeployed), or deleted from the domain by first selecting the application name and using the controls on this page.

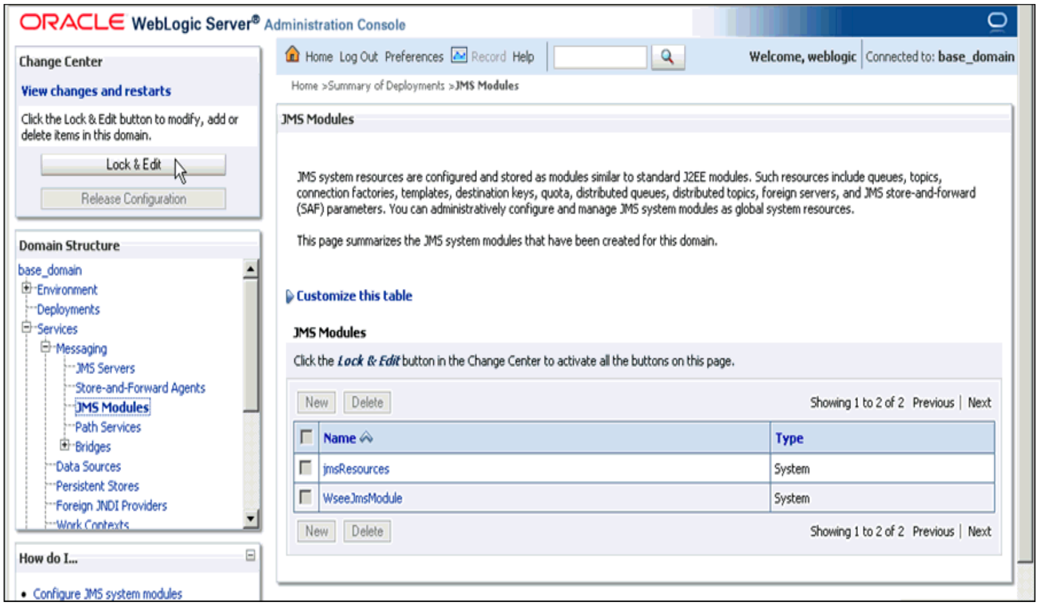
To install a new application or module for deployment to targets in this domain, click the Install button.

Deployments

	Name	State	Health	Type	Deployment Order
□	adf.oracle.businesseditor(1.0,11.1.1.2.0)	Active		Library	100
□	adf.oracle.domain(1.0,11.1.1.2.0)	Active		Library	100
□	adf.oracle.domain.webapp(1.0,11.1.1.2.0)	Active		Library	100
□	ALDSP Transport Provider	Active	✓ OK	Web Application	161
□	aldsp_transport-l10n(3.0,3.0)	Active		Library	160

23. Click **jmsResources**, as shown in Figure 7–117.

Figure 7–117 JMS Modules



Click the Lock & Edit button to modify, add or delete items in this domain.

JMS Modules

JMS system resources are configured and stored as modules similar to standard J2EE modules. Such resources include queues, topics, connection factories, templates, destination keys, quota, distributed queues, distributed topics, foreign servers, and JMS store-and-forward (SAF) parameters. You can administratively configure and manage JMS system modules as global system resources.

This page summarizes the JMS system modules that have been created for this domain.

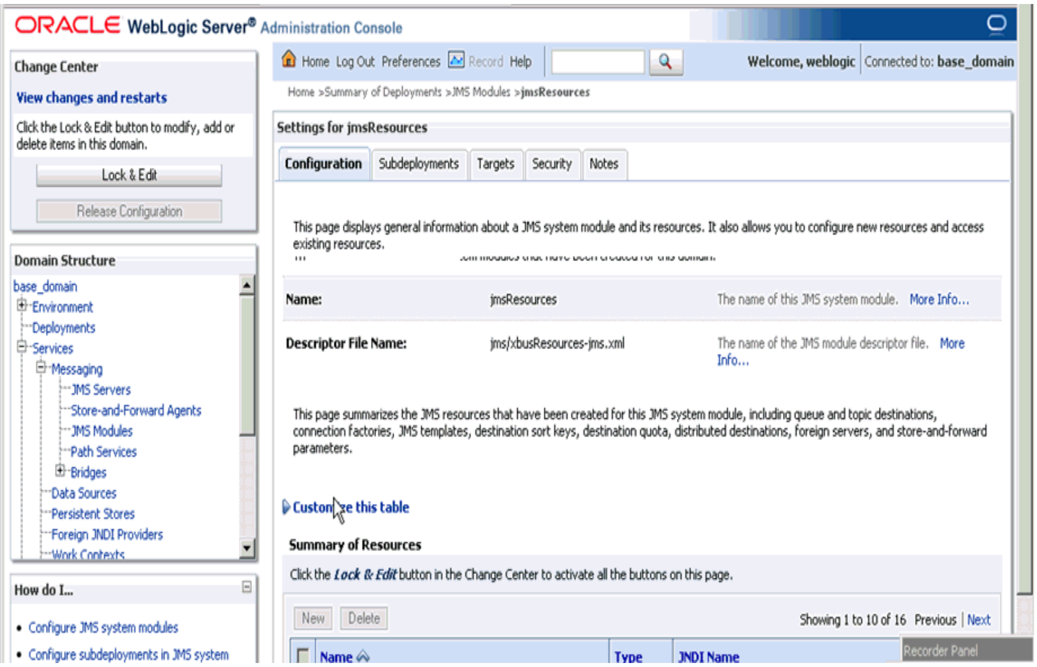
JMS Modules

Click the *Lock & Edit* button in the Change Center to activate all the buttons on this page.

□	Name	Type
□	jmsResources	System
□	WseeJmsModule	System

24. Click **Lock & Edit**, as shown in Figure 7–118.

Figure 7–118 Configuration Settings



Click the Lock & Edit button to modify, add or delete items in this domain.

Settings for jmsResources

This page displays general information about a JMS system module and its resources. It also allows you to configure new resources and access existing resources.

Name:	jmsResources	The name of this JMS system module. More Info...
Descriptor File Name:	jms/xbusResources-jms.xml	The name of the JMS module descriptor file. More Info...

This page summarizes the JMS resources that have been created for this JMS system module, including queue and topic destinations, connection factories, JMS templates, destination sort keys, destination quota, distributed destinations, foreign servers, and store-and-forward parameters.

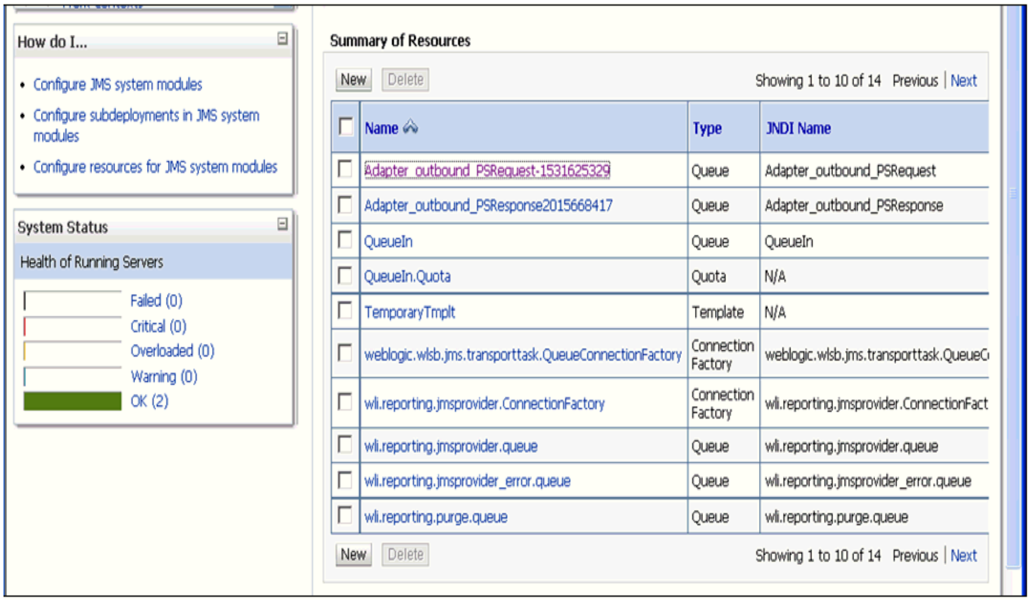
Summary of Resources

Click the *Lock & Edit* button in the Change Center to activate all the buttons on this page.

□	Name	Type	JNDI Name

25. Click the appropriate request link, for example, **Adapter_outbound_PSRequest**, as shown in Figure 7–119.

Figure 7–119 Adapter_outbound_PSRequest


How do I...
Configure JMS system modules
Configure subdeployments in JMS system modules
Configure resources for JMS system modules
System Status
Health of Running Servers
Failed (0)
Critical (0)
Overloaded (0)
Warning (0)
OK (2)
New
Delete
Previous
Next

Summary of Resources

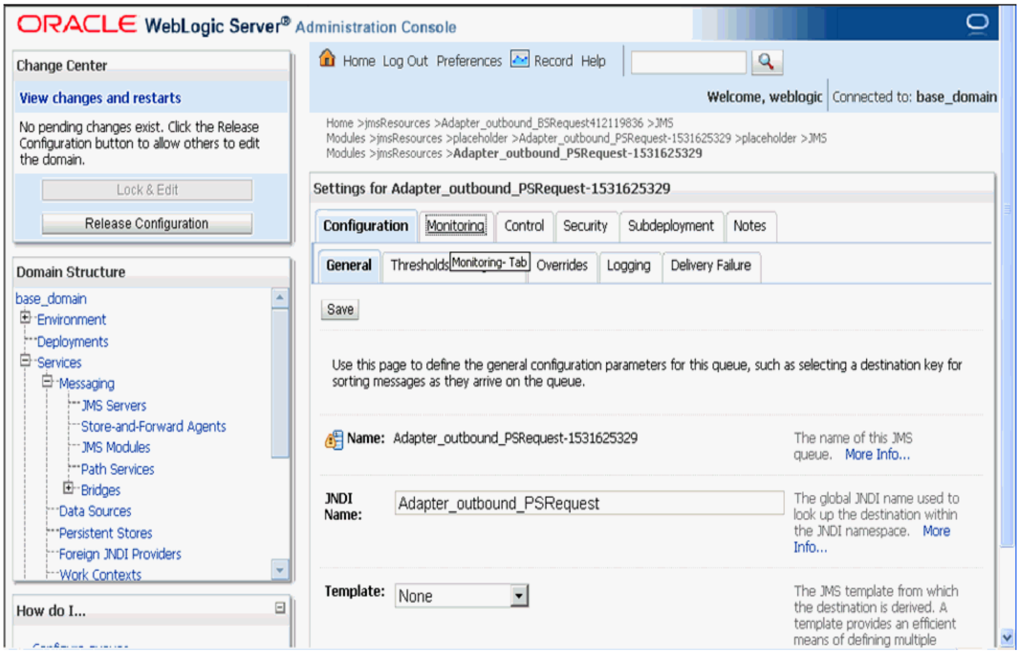
Showing 1 to 10 of 14

☐	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
☐	TemporaryTmplt	Template	N/A
☐	weblogic.wlsb.jms.transporttask.QueueConnectionFactory	Connection Factory	weblogic.wlsb.jms.transporttask.QueueC
☐	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

Showing 1 to 10 of 14

26. Click the **Monitoring** tab, as shown in Figure 7–120.

Figure 7–120 Monitoring Tab



No pending changes exist. Click the Release Configuration button to allow others to edit the domain.

Welcome, weblogic | Connected to: base_domain

Home >jmsResources >Adapter_outbound_BSRequest412119836 >JMS Modules >jmsResources >placeholder >Adapter_outbound_PSRequest-1531625329 >placeholder >JMS Modules >jmsResources >**Adapter_outbound_PSRequest-1531625329**

Settings for Adapter_outbound_PSRequest-1531625329

Use this page to define the general configuration parameters for this queue, such as selecting a destination key for sorting messages as they arrive on the queue.

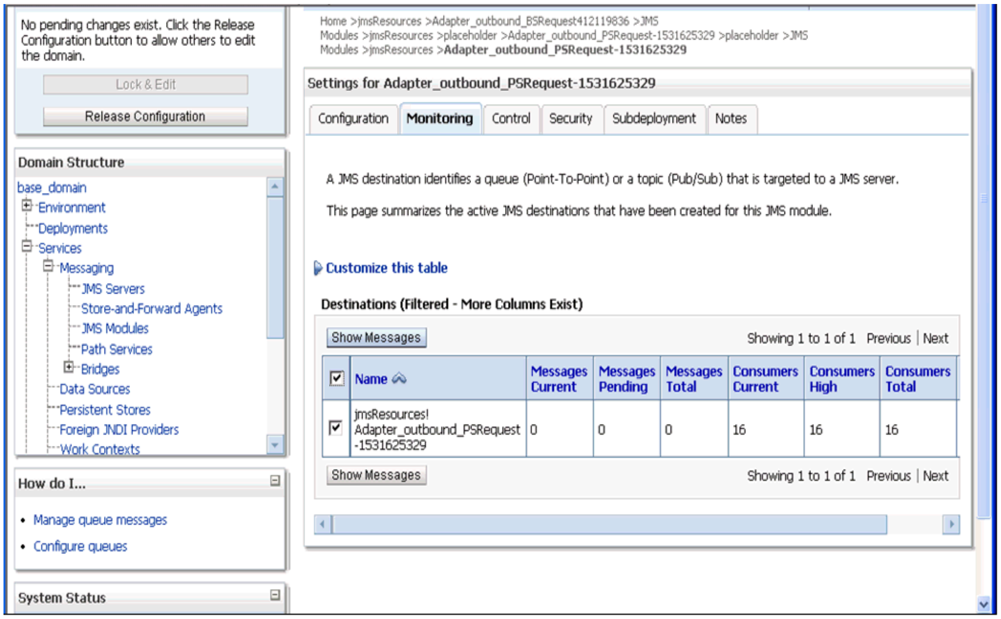
Name: Adapter_outbound_PSRequest-1531625329 — The name of this JMS queue. More Info...

JNDI Name: Adapter_outbound_PSRequest — The global JNDI name used to look up the destination within the JNDI namespace. More Info...

Template: None — The JMS template from which the destination is derived. A template provides an efficient means of defining multiple

27. Enable the check box and click **Show Messages**, as shown in Figure 7–121.

Figure 7–121 Adapter Settings



No pending changes exist. Click the Release Configuration button to allow others to edit the domain.

Lock & Edit

Release Configuration

Domain Structure

How do I...

- Manage queue messages
- Configure queues

System Status

Home >jmsResources >Adapter_outbound_BSRequest412119836 >JMS Modules >jmsResources >placeholder >Adapter_outbound_PSRequest-1531625329 >placeholder >JMS Modules >jmsResources >**Adapter_outbound_PSRequest-1531625329**

Settings for Adapter_outbound_PSRequest-1531625329

A JMS destination identifies a queue (Point-To-Point) or a topic (Pub/Sub) that is targeted to a JMS server.

This page summarizes the active JMS destinations that have been created for this JMS module.

Customize this table

Destinations (Filtered - More Columns Exist)

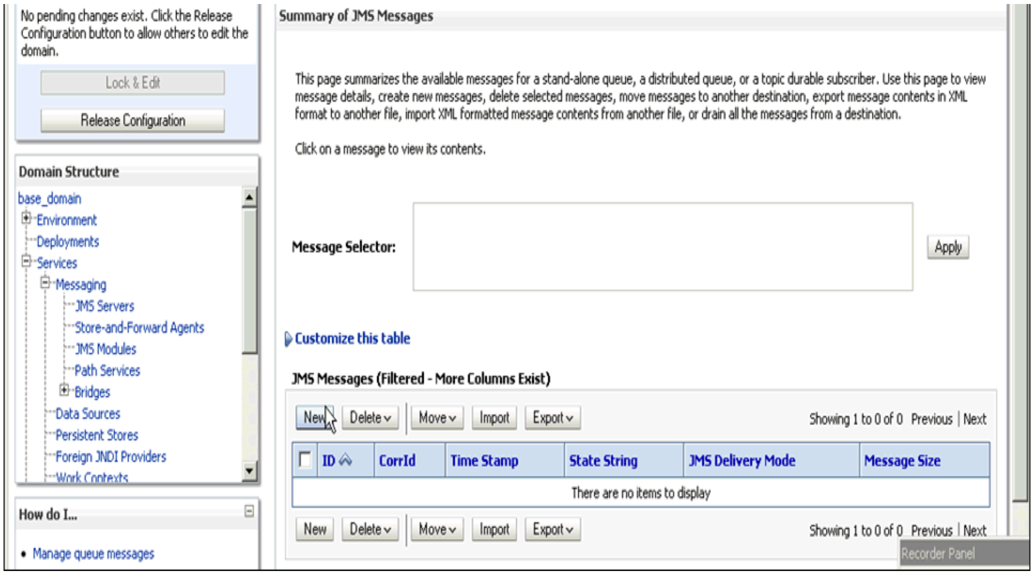
Show Messages Showing 1 to 1 of 1 Previous | Next

☑	Name	Messages Current	Messages Pending	Messages Total	Consumers Current	Consumers High	Consumers Total
☑	jmsResources! Adapter_outbound_PSRequest -1531625329	0	0	0	16	16	16

Show Messages Showing 1 to 1 of 1 Previous | Next

28. Click **New**, as shown in Figure 7–122.

Figure 7–122 JMS Messages



No pending changes exist. Click the Release Configuration button to allow others to edit the domain.

Lock & Edit

Release Configuration

Domain Structure

How do I...

- Manage queue messages

Summary of JMS Messages

This page summarizes the available messages for a stand-alone queue, a distributed queue, or a topic durable subscriber. Use this page to view message details, create new messages, delete selected messages, move messages to another destination, export message contents in XML format to another file, import XML formatted message contents from another file, or drain all the messages from a destination.

Click on a message to view its contents.

Message Selector:

Customize this table

JMS Messages (Filtered - More Columns Exist)

Showing 1 to 0 of 0 Previous | Next

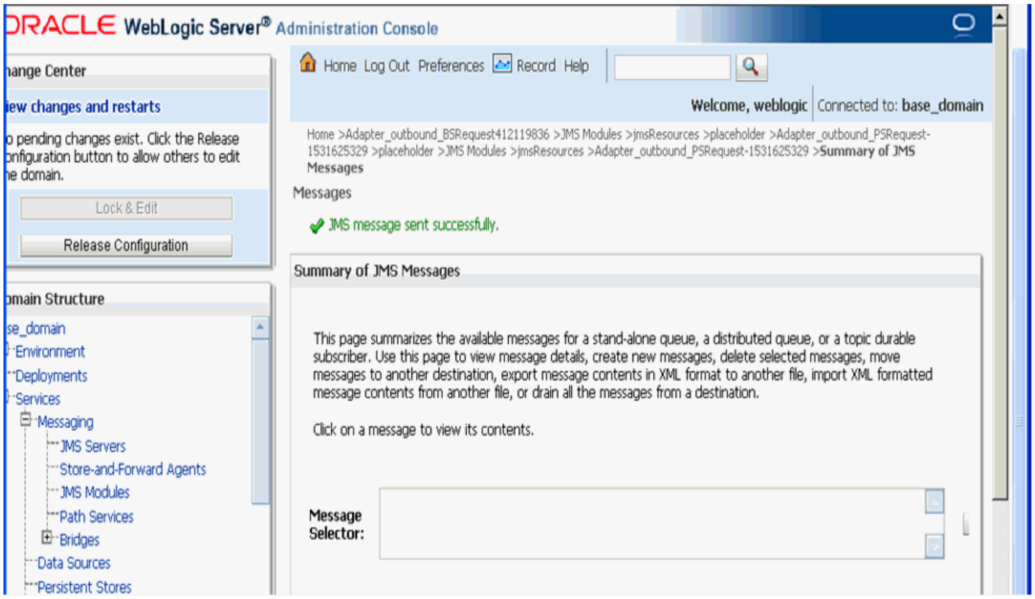
□	ID	CorrId	Time Stamp	State String	JMS Delivery Mode	Message Size
There are no items to display						

Showing 1 to 0 of 0 Previous | Next

29. Provide the input payload in the **Body** field and click **OK**.

    A Success message appears, as shown in Figure 7–123.

Figure 7–123 JMS Success Message



o pending changes exist. Click the Release onfiguration button to allow others to edit he domain.

Home >Adapter_outbound_BSRequest412119836 >JMS Modules >jmsResources >placeholder >Adapter_outbound_PSRequest-1531625329 >placeholder >JMS Modules >jmsResources >Adapter_outbound_PSRequest-1531625329 >**Summary of JMS Messages**

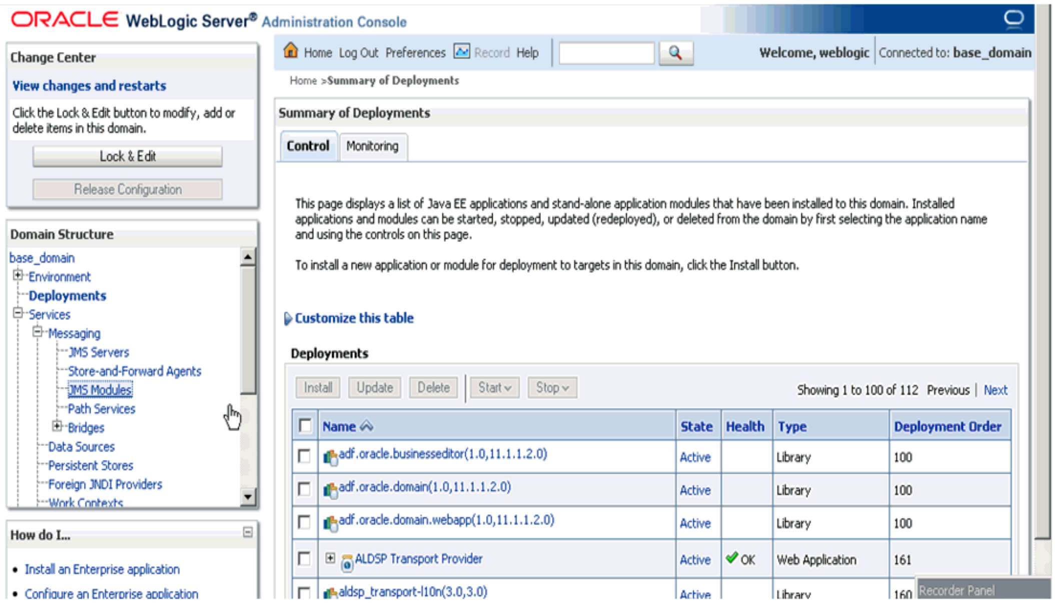
✓ JMS message sent successfully.

This page summarizes the available messages for a stand-alone queue, a distributed queue, or a topic durable subscriber. Use this page to view message details, create new messages, delete selected messages, move messages to another destination, export message contents in XML format to another file, import XML formatted message contents from another file, or drain all the messages from a destination.

Click on a message to view its contents.

30. In the Oracle WLS Console, expand **Services**, expand **Messaging**, and click **JMS Modules**, as shown in Figure 7–124.

Figure 7–124 JMS Modules



Click the Lock & Edit button to modify, add or delete items in this domain.

This page displays a list of Java EE applications and stand-alone application modules that have been installed to this domain. Installed applications and modules can be started, stopped, updated (redeployed), or deleted from the domain by first selecting the application name and using the controls on this page.

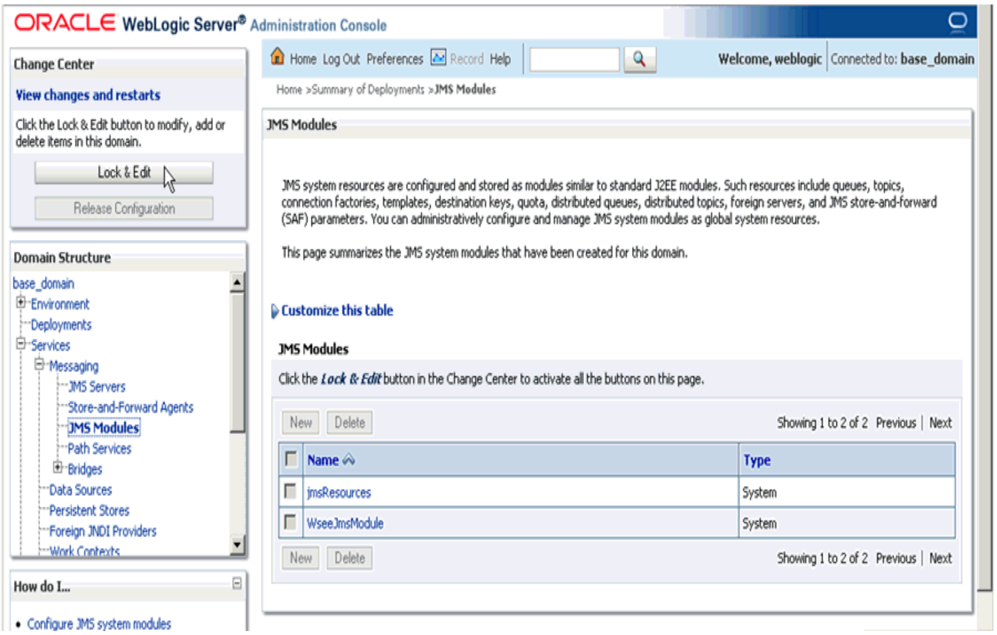
To install a new application or module for deployment to targets in this domain, click the Install button.

Showing 1 to 100 of 112 Previous | Next

□	Name	State	Health	Type	Deployment Order
□	adf.oracle.businesseditor(1.0,11.1.1.2.0)	Active		Library	100
□	adf.oracle.domain(1.0,11.1.1.2.0)	Active		Library	100
□	adf.oracle.domain.webapp(1.0,11.1.1.2.0)	Active		Library	100
□	ALDSP Transport Provider	Active	✓ OK	Web Application	161
□	aldsp_transport-l10n(3.0,3.0)	Active		Library	160

31. Click **jmsResources**, as shown in Figure 7–125.

Figure 7–125 jmsResources



JMS Modules

JMS system resources are configured and stored as modules similar to standard J2EE modules. Such resources include queues, topics, connection factories, templates, destination keys, quota, distributed queues, distributed topics, foreign servers, and JMS store-and-forward (SAF) parameters. You can administratively configure and manage JMS system modules as global system resources.

This page summarizes the JMS system modules that have been created for this domain.

Customize this table

JMS Modules

Click the *Lock & Edit* button in the Change Center to activate all the buttons on this page.

New Delete Showing 1 to 2 of 2 Previous | Next

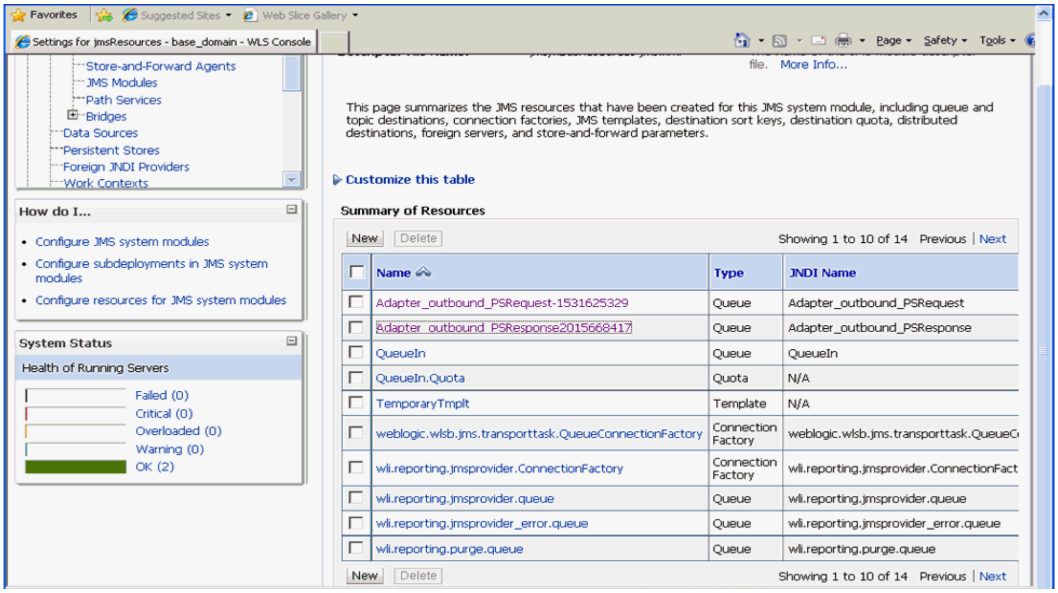
□	Name	Type
□	jmsResources	System
□	WseeJmsModule	System

New Delete Showing 1 to 2 of 2 Previous | Next

ORACLE WebLogic Server® Administration Console
Home Log Out Preferences Record Help
Welcome, weblogic Connected to: base_domain
Home >Summary of Deployments >JMS Modules
Change Center
View changes and restarts
Click the Lock & Edit button to modify, add or delete items in this domain.
Lock & Edit
Release Configuration
Domain Structure
base_domain
Environment
Deployments
Services
Messaging
JMS Servers
Store-and-Forward Agents
JMS Modules
Path Services
Bridges
Data Sources
Persistent Stores
Foreign JNDI Providers
Work Contexts
How do I...
Configure JMS system modules

32. Click the appropriate response link, for example, **Adapter_outbound_PSResponse**, as shown in Figure 7–126.

Figure 7–126 Summary of Resources



file. More Info...

This page summarizes the JMS resources that have been created for this JMS system module, including queue and topic destinations, connection factories, JMS templates, destination sort keys, destination quota, distributed destinations, foreign servers, and store-and-forward parameters.

Customize this table

Summary of Resources

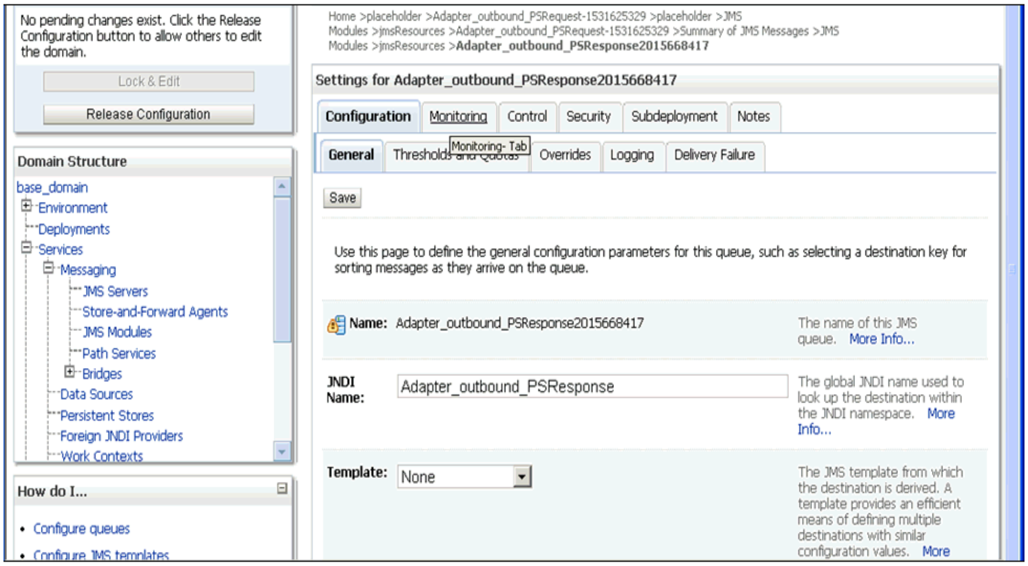
New Delete Showing 1 to 10 of 14 Previous | Next

□	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
□	TemporaryTmplt	Template	N/A
□	weblogic.wlsb.jms.transporttask.QueueConnectionFactory	Connection Factory	weblogic.wlsb.jms.transporttask.QueueC
□	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

New Delete Showing 1 to 10 of 14 Previous | Next

33. Click the **Monitoring** tab, as shown in Figure 7–127.

Figure 7–127 Monitoring Tab



No pending changes exist. Click the Release Configuration button to allow others to edit the domain.

Lock & Edit

Release Configuration

Domain Structure

base_domain
- Environment
- Deployments
- Services
  - Messaging
    - JMS Servers
    - Store-and-Forward Agents
    - JMS Modules
    - Path Services
    - Bridges
  - Data Sources
  - Persistent Stores
  - Foreign JNDI Providers
  - Work Contexts

How do I...
- Configure queues
- Configure JMS templates

Home >placeholder >Adapter_outbound_PSRequest-1531625329 >placeholder >JMS Modules >jmsResources >Adapter_outbound_PSRequest-1531625329 >Summary of JMS Messages >JMS Modules >jmsResources >**Adapter_outbound_PSResponse2015668417**

Settings for Adapter_outbound_PSResponse2015668417

Use this page to define the general configuration parameters for this queue, such as selecting a destination key for sorting messages as they arrive on the queue.

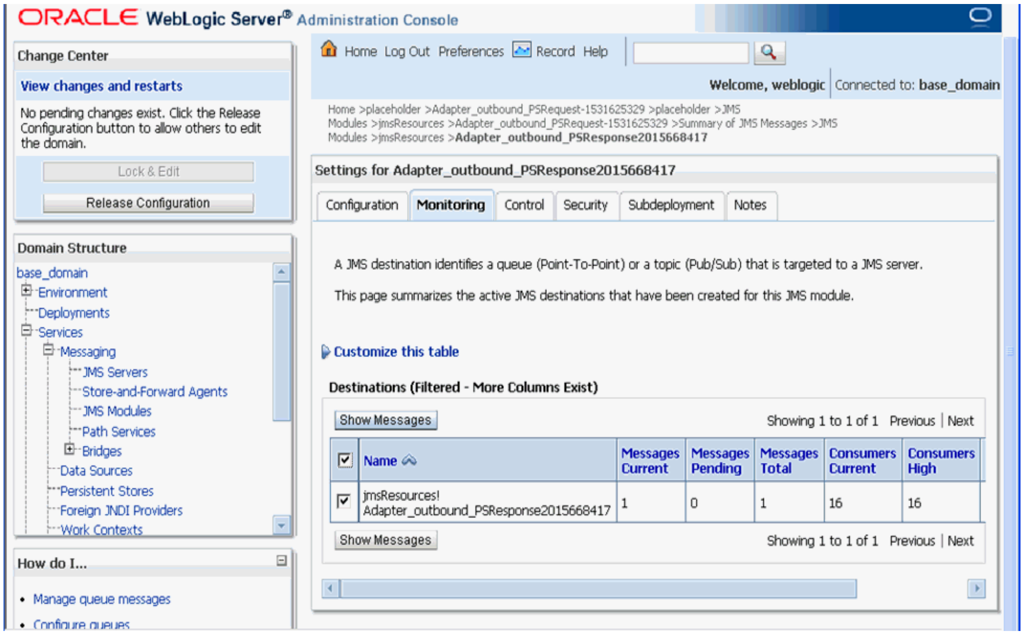
Name: Adapter_outbound_PSResponse2015668417 — The name of this JMS queue. More Info...

JNDI Name: Adapter_outbound_PSResponse — The global JNDI name used to look up the destination within the JNDI namespace. More Info...

Template: None — The JMS template from which the destination is derived. A template provides an efficient means of defining multiple destinations with similar configuration values. More

34. Enable the check box and click **Show Messages**, as shown in Figure 7–128.

Figure 7–128 Destination Messages



ORACLE WebLogic Server® Administration Console
Home Log Out Preferences Record Help

Welcome, weblogic | Connected to: base_domain

Change Center

View changes and restarts

No pending changes exist. Click the Release Configuration button to allow others to edit the domain.

Lock & Edit

Release Configuration

Domain Structure

base_domain
- Environment
- Deployments
- Services
  - Messaging
    - JMS Servers
    - Store-and-Forward Agents
    - JMS Modules
    - Path Services
    - Bridges
  - Data Sources
  - Persistent Stores
  - Foreign JNDI Providers
  - Work Contexts

How do I...
- Manage queue messages
- Configure queues

Home >placeholder >Adapter_outbound_PSRequest-1531625329 >placeholder >JMS Modules >jmsResources >Adapter_outbound_PSRequest-1531625329 >Summary of JMS Messages >JMS Modules >jmsResources >**Adapter_outbound_PSResponse2015668417**

Settings for Adapter_outbound_PSResponse2015668417

A JMS destination identifies a queue (Point-To-Point) or a topic (Pub/Sub) that is targeted to a JMS server.

This page summarizes the active JMS destinations that have been created for this JMS module.

Customize this table

Destinations (Filtered - More Columns Exist)

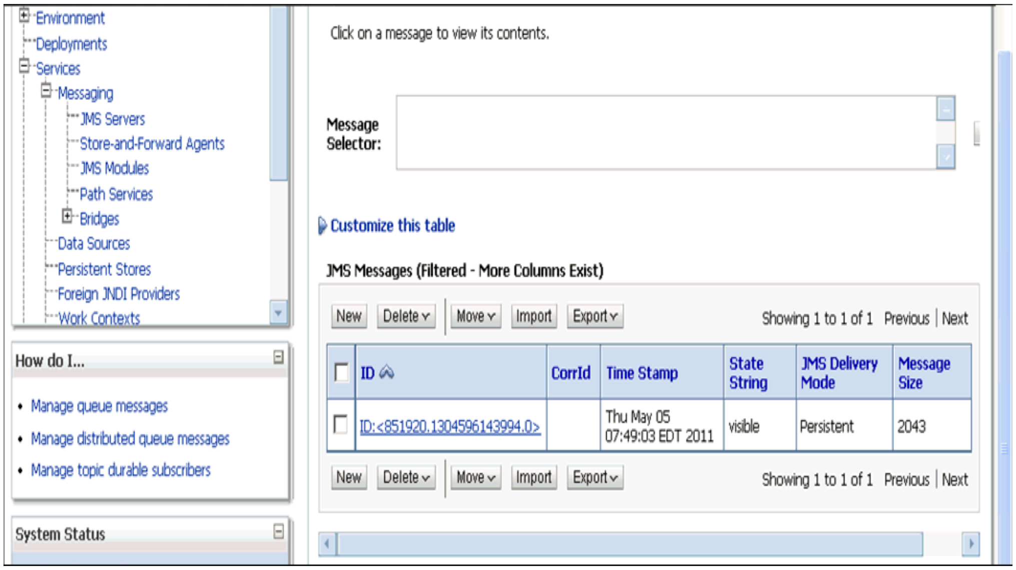
Show Messages — Showing 1 to 1 of 1 Previous | Next

☑	Name	Messages Current	Messages Pending	Messages Total	Consumers Current	Consumers High
☑	jmsResources! Adapter_outbound_PSResponse2015668417	1	0	1	16	16

Show Messages — Showing 1 to 1 of 1 Previous | Next

35. Click the **ID** link, as shown in Figure 7–129.

Figure 7–129 JMS Messages



Click on a message to view its contents.

Message Selector:

JMS Messages (Filtered - More Columns Exist)

□	ID	CorrId	Time Stamp	State String	JMS Delivery Mode	Message Size
□	ID:<851920.1304596143994.0>		Thu May 05 07:49:03 EDT 2011	visible	Persistent	2043

The response document is shown under the Text field.

7.6 Configuring HTTP Proxy Services Using Oracle Service Bus (J2CA Configuration)

This section describes how to configure HTTP Proxy Services using Oracle Service Bus for a J2CA configuration.

1. Start the Oracle Service Bus and create the required project folders.

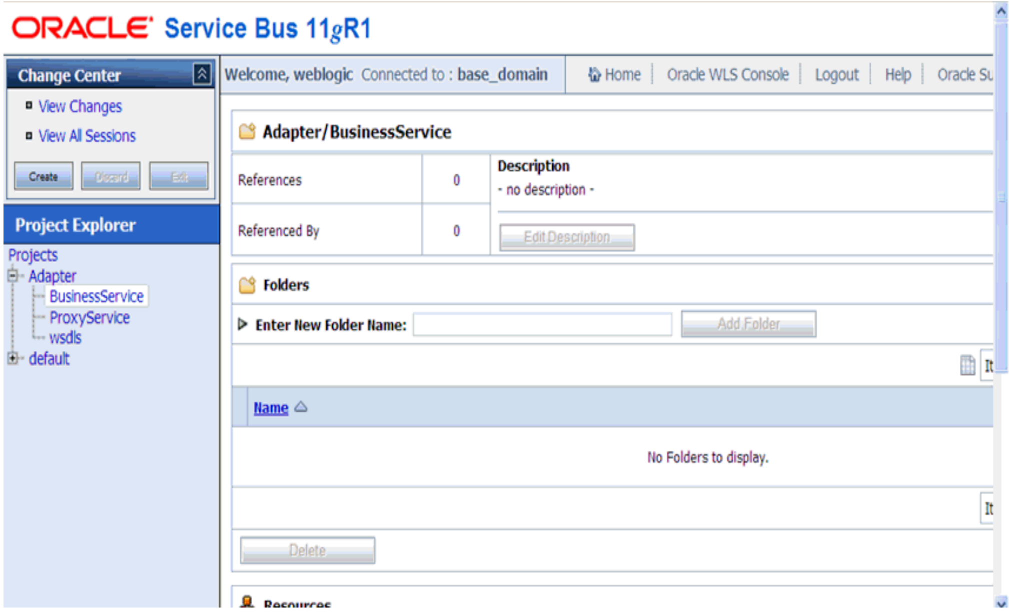
   For more information, see Section 7.2.1, "Starting Oracle Service Bus and Creating Project Folders".

2. Generate and publish the WSDL from Application Explorer to the created project folder, and create a Business Service using the published WSDL.

   For more information, see Section 7.2.3, "Publishing a WSDL From Application Explorer to Oracle Service Bus".

3. Open the Service Bus console page, as shown in Figure 7–130.

Figure 7–130 Service Bus Console Page



Adapter/BusinessService

References	0	Description - no description -
Referenced By	0	

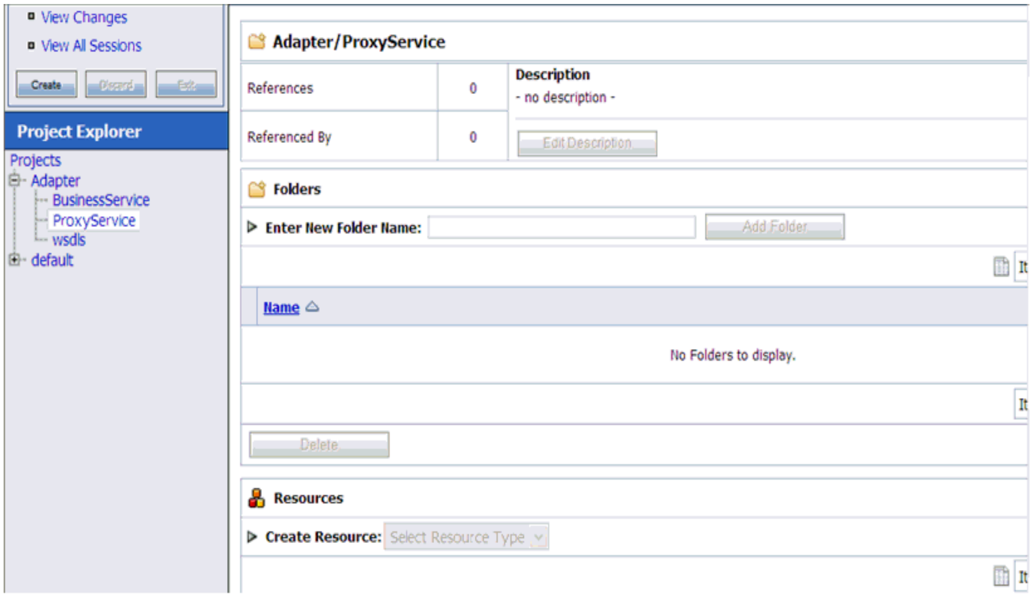
Folders

Enter New Folder Name:

No Folders to display.

4. In the Project Explorer, select the **ProxyService** project folder, and click **Create**, as shown in Figure 7–131.

Figure 7–131 Project Explorer



Adapter/ProxyService

References	0	Description - no description -
Referenced By	0	

Folders

Enter New Folder Name:

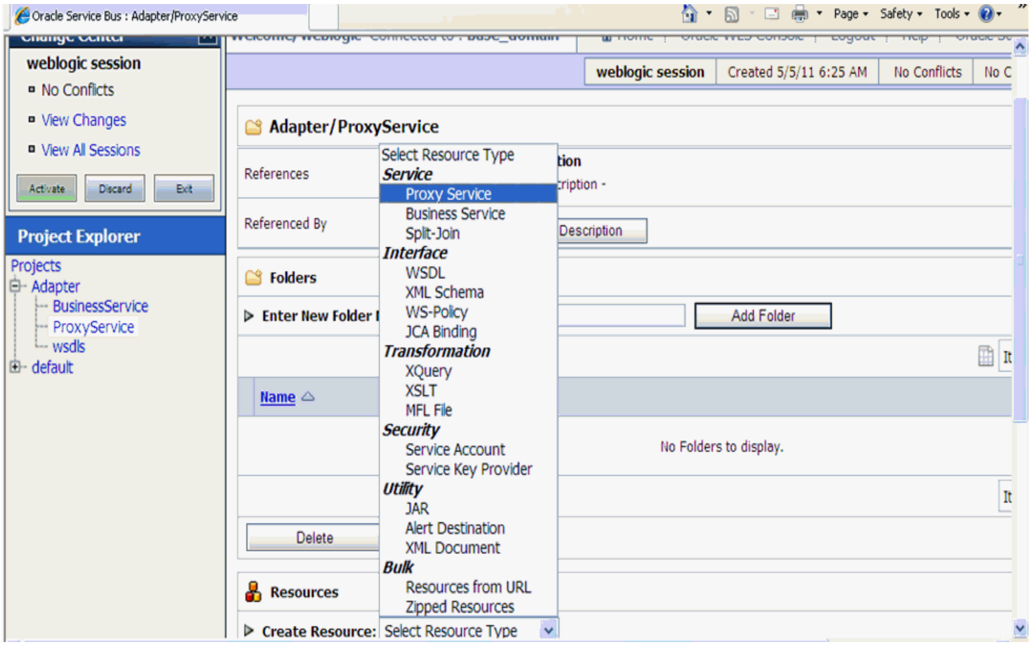
No Folders to display.

Resources

Create Resource:

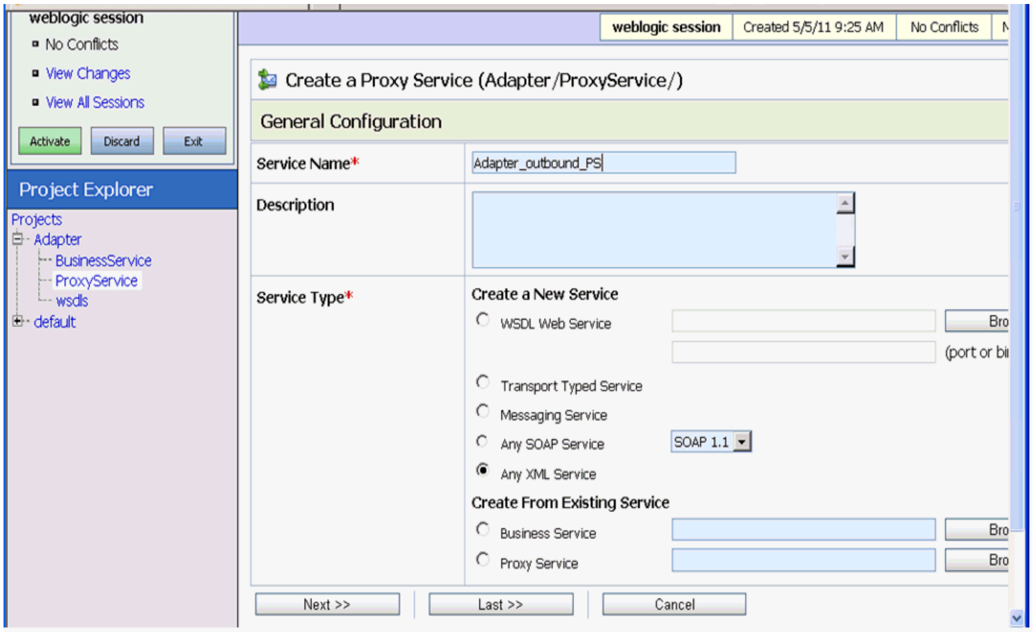
5. In the Create Resource list on the right pane, select **Proxy Service**, as shown in Figure 7–132.

Figure 7–132 Proxy Service



6. In the Service Name field, enter an appropriate name, as shown in Figure 7–133.

Figure 7–133 Service Name


Create a Proxy Service (Adapter/ProxyService/)

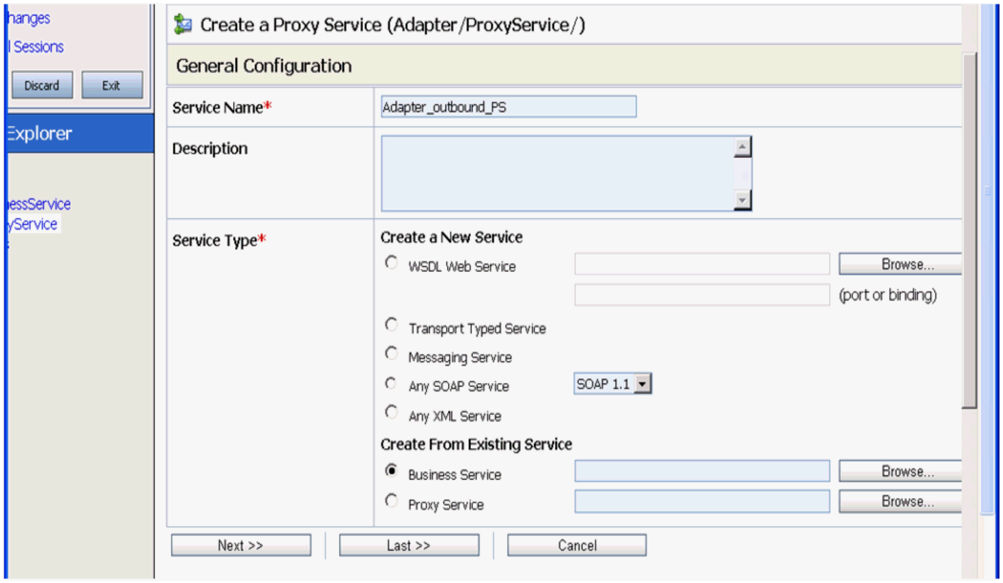
General Configuration

Service Name*: Adapter_outbound_PS

Service Type*: ● Any XML Service

7. In the Service Type section, under Create From Existing Service, select the **Business Service** radio button and click **Browse**, as shown in Figure 7–134.

Figure 7–134 General Configuration



Create a Proxy Service (Adapter/ProxyService/)

General Configuration

Service Name*: Adapter_outbound_PS

Description:

Service Type*:

Create a New Service

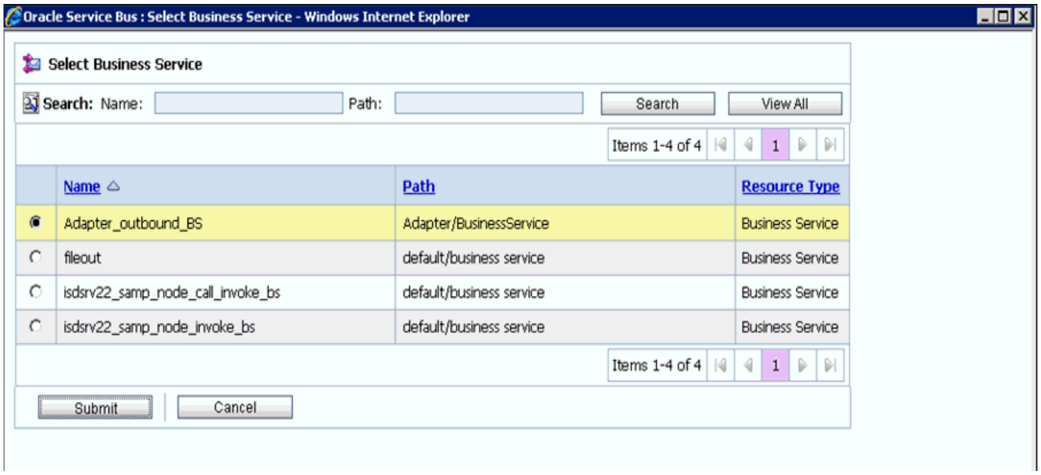
- ○ WSDL Web Service — Browse... (port or binding)
- ○ Transport Typed Service
- ○ Messaging Service
- ○ Any SOAP Service — SOAP 1.1
- ○ Any XML Service

Create From Existing Service

- ● Business Service — Browse...
- ○ Proxy Service — Browse...

8. Select the existing Business Service and click **Submit**, as shown in Figure 7–135.

Figure 7–135 Business Service



Oracle Service Bus : Select Business Service - Windows Internet Explorer

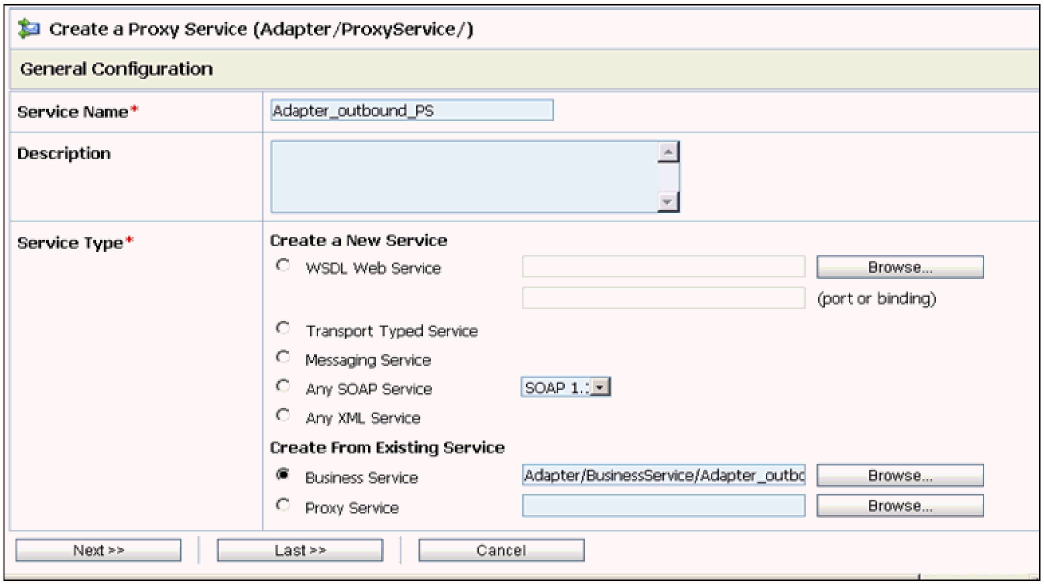
Select Business Service

Search: Name: Path:

	Name	Path	Resource Type
●	Adapter_outbound_BS	Adapter/BusinessService	Business Service
○	fileout	default/business service	Business Service
○	isdsrv22_samp_node_call_invoke_bs	default/business service	Business Service
○	isdsrv22_samp_node_invoke_bs	default/business service	Business Service

9. Click **Next**, as shown in Figure 7–136.

Figure 7–136 General Configuration



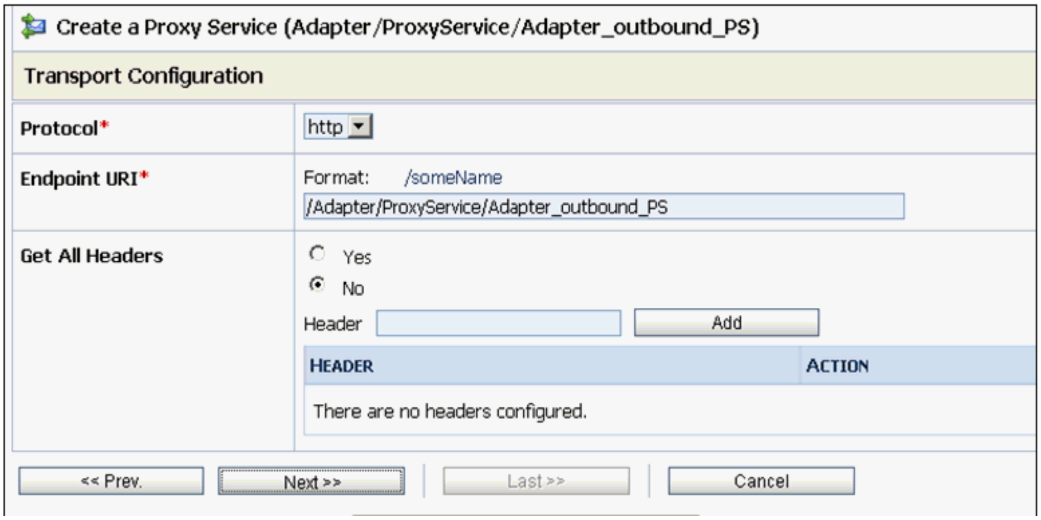
Create a Proxy Service (Adapter/ProxyService/)

General Configuration

Service Name*	Adapter_outbound_PS
Description	
Service Type*	Create a New Service ○ WSDL Web Service (port or binding) ○ Transport Typed Service ○ Messaging Service ○ Any SOAP Service SOAP 1.: ○ Any XML Service Create From Existing Service ● Business Service Adapter/BusinessService/Adapter_outbc ○ Proxy Service

10. Select **http** in the Protocol list and click **Next**, as shown in Figure 7–137.

Figure 7–137 Transport Configuration



Create a Proxy Service (Adapter/ProxyService/Adapter_outbound_PS)

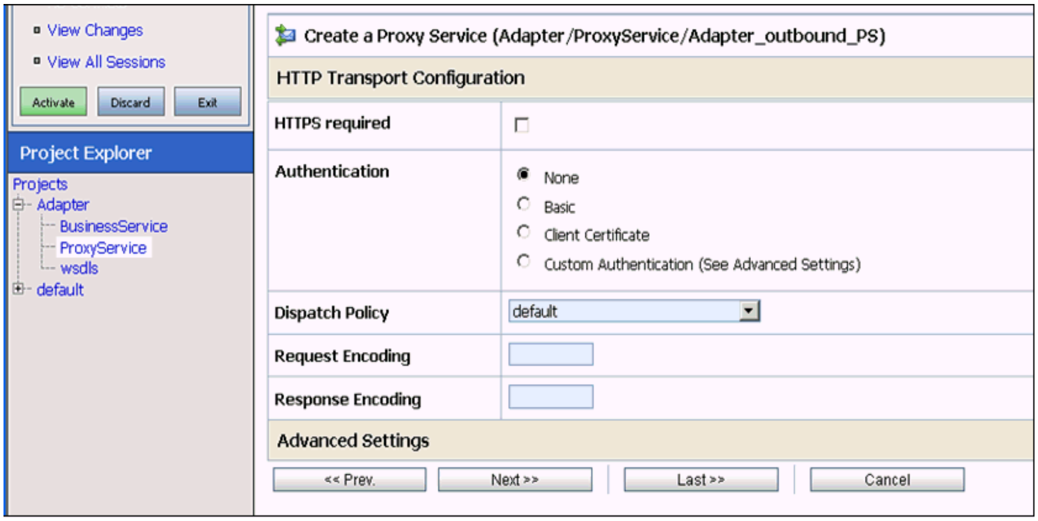
Transport Configuration

Protocol*	http
Endpoint URI*	Format: /someName /Adapter/ProxyService/Adapter_outbound_PS
Get All Headers	○ Yes ● No Header

HEADER	ACTION
There are no headers configured.	

11. Click **Next**, as shown in Figure 7–138.

Figure 7–138 HTTP Transport Configuration



Create a Proxy Service (Adapter/ProxyService/Adapter_outbound_PS)

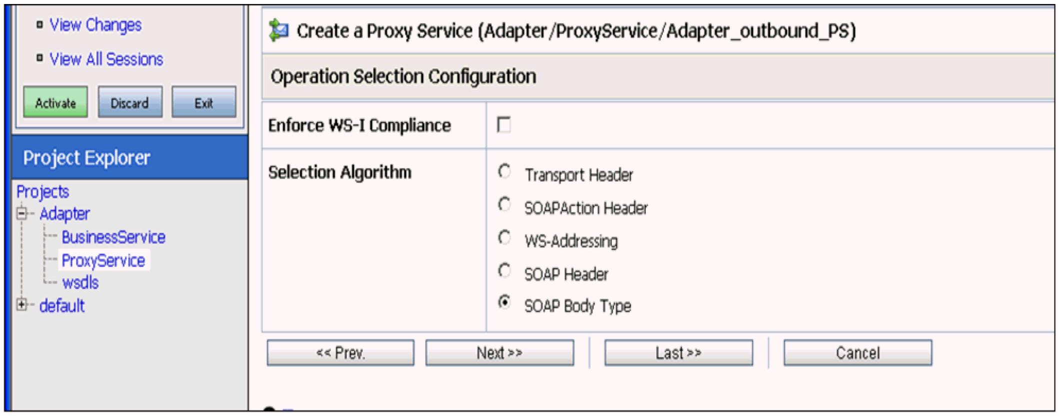
HTTP Transport Configuration

HTTPS required	□
Authentication	● None ○ Basic ○ Client Certificate ○ Custom Authentication (See Advanced Settings)
Dispatch Policy	default
Request Encoding	
Response Encoding	

Advanced Settings

12. Click **Next**, as shown in Figure 7–139.

Figure 7–139 Operation Selection Configuration



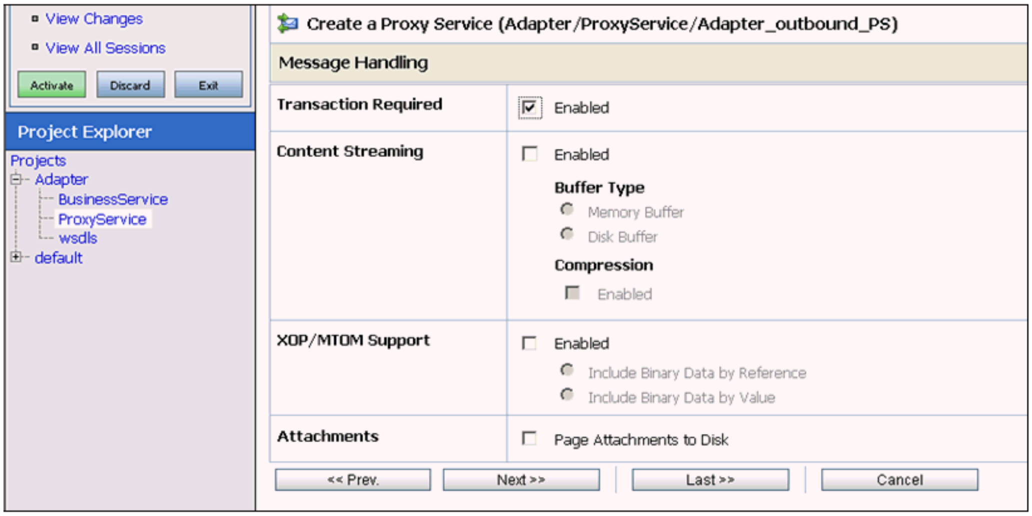
Create a Proxy Service (Adapter/ProxyService/Adapter_outbound_PS)

Operation Selection Configuration

Enforce WS-I Compliance	□
Selection Algorithm	○ Transport Header ○ SOAPAction Header ○ WS-Addressing ○ SOAP Header ● SOAP Body Type

13. Enable the **Transaction Required** check box and click **Next**, as shown in Figure 7–140.

Figure 7–140 Message Handling



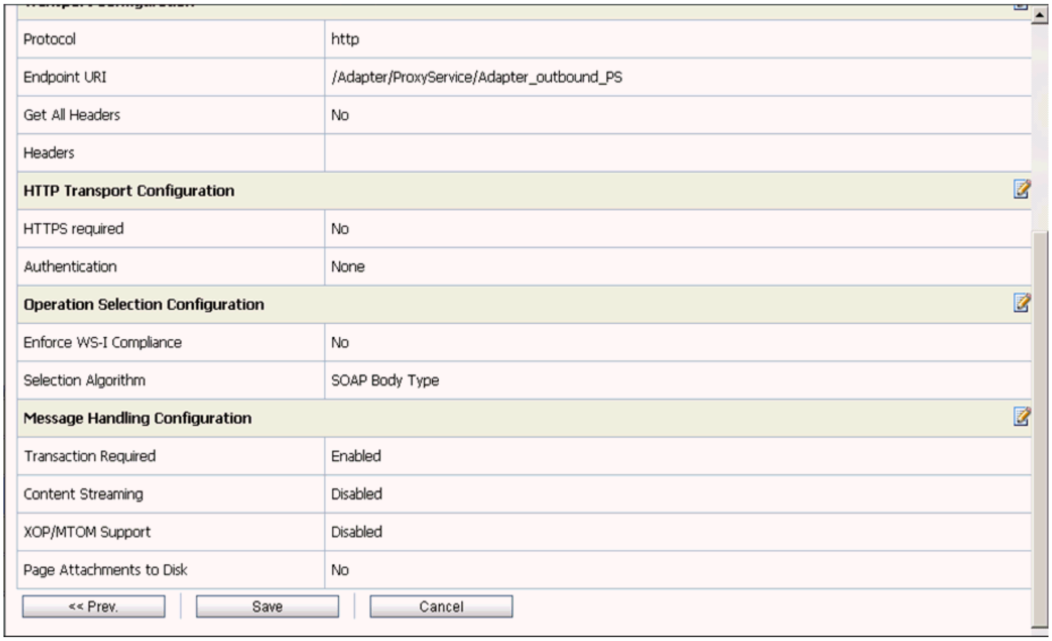
Create a Proxy Service (Adapter/ProxyService/Adapter_outbound_PS)

Message Handling

Transaction Required	☑ Enabled
Content Streaming	□ Enabled Buffer Type ○ Memory Buffer ○ Disk Buffer Compression □ Enabled
XOP/MTOM Support	□ Enabled ○ Include Binary Data by Reference ○ Include Binary Data by Value
Attachments	□ Page Attachments to Disk

14. Click **Save**, as shown in Figure 7–141.

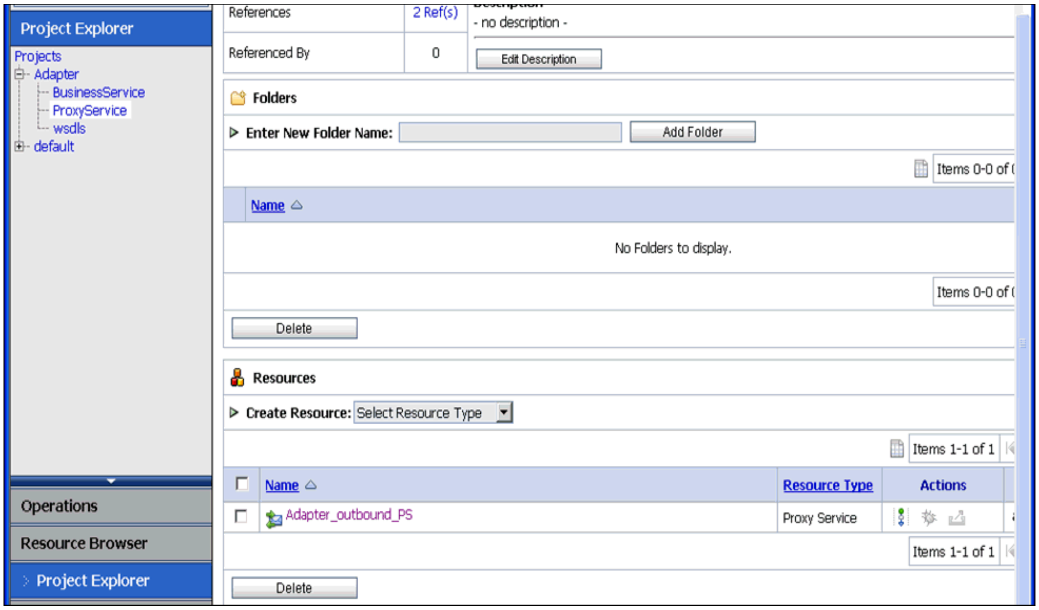
Figure 7–141 Save



Protocol	http
Endpoint URI	/Adapter/ProxyService/Adapter_outbound_PS
Get All Headers	No
Headers	
HTTP Transport Configuration	
HTTPS required	No
Authentication	None
Operation Selection Configuration	
Enforce WS-I Compliance	No
Selection Algorithm	SOAP Body Type
Message Handling Configuration	
Transaction Required	Enabled
Content Streaming	Disabled
XOP/MTOM Support	Disabled
Page Attachments to Disk	No

The created Proxy Service is saved, as shown in Figure 7–142.

Figure 7–142 Proxy Service



References	2 Ref(s)	- no description -
Referenced By	0	

Items 0-0 of (

No Folders to display.

Items 0-0 of (

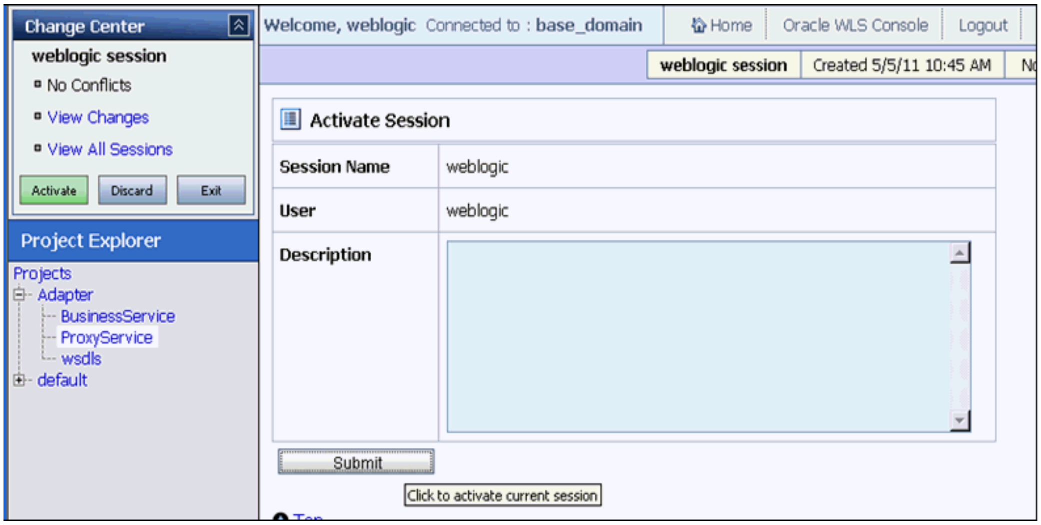
Items 1-1 of 1

Name	Resource Type
Adapter_outbound_PS	Proxy Service

Items 1-1 of 1

15. Click **Activate** in the left pane, and then **Submit** on the right pane, as shown in Figure 7–143.

Figure 7–143 Activate Session



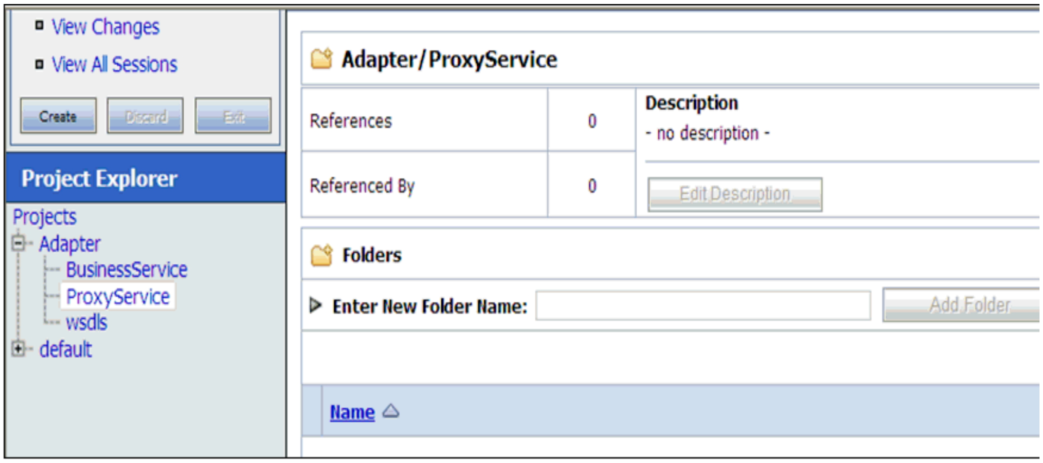
Welcome, weblogic Connected to : base_domain

weblogic session Created 5/5/11 10:45 AM

Session Name	weblogic
User	weblogic
Description	

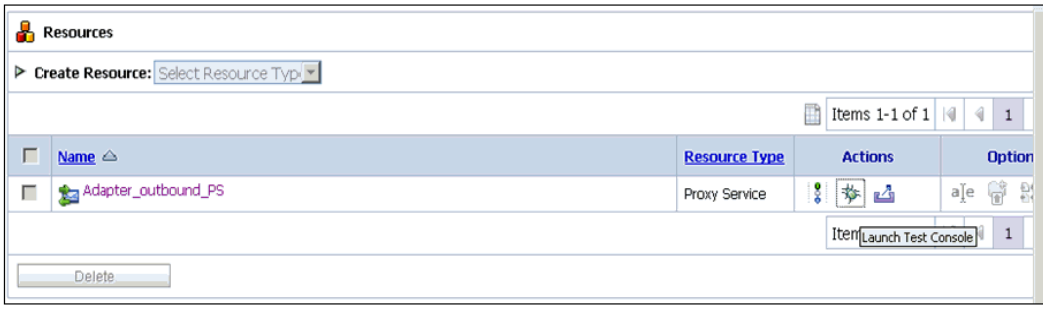
16. Click **ProxyService** in the Projects folder on the left pane, as shown in Figure 7–144.

Figure 7–144 ProxyService



17. Click the **Launch Test Console** icon for the created Proxy Service, as shown in Figure 7–145.

Figure 7–145 Launch Test Console



18. Uncheck the **Direct Call** check box, provide the input values for **Payload**, and click **Execute**.

19. Review the **Response Document**.

8

Configuring an Outbound and Inbound Process for Oracle Service Bus Using JDeveloper

Oracle Application Adapter for Siebel integrates seamlessly with Oracle JDeveloper to facilitate Web service integration.

This chapter contains the following sections:

- Section 8.1, "Configuring an OSB Outbound Process Using JDeveloper (J2CA Configuration)"
- Section 8.2, "Configuring an OSB Inbound Process Using JDeveloper (J2CA Configuration)"
- Section 8.3, "Configuring an OSB Outbound Process Using JDeveloper (BSE Configuration)"
- Section 8.4, "Configuring a JMS Inbound Process Using JDeveloper (J2CA Configuration)"
- Section 8.5, "Configuring a JMS Outbound Process Using JDeveloper (J2CA Configuration)"
- Section 8.6, "Configuring an HTTP Outbound Process Using JDeveloper (J2CA Configuration)"

8.1 Configuring an OSB Outbound Process Using JDeveloper (J2CA Configuration)

This section describes how to configure an OSB outbound process to your Siebel system, using Oracle JDeveloper for J2CA configurations.

A sample project has been provided for this outbound use case scenario in the following folder of the Application Adapters installation:

```
<ADAPTER_HOME>\etc\sample\SIEBEL_Samples.zip\SIEBEL_Samples\OSB_
Jdeveloper\J2CA\Siebel_Sample_J2CA_OSB_Outbound_Project
```

This section includes the following topics:

- Section 8.1.1, "Creating a Service Bus Application for OSB"
- Section 8.1.2, "Defining an OSB Outbound Process"
- Section 8.1.3, "Deploying the OSB Outbound Process"

Prerequisites

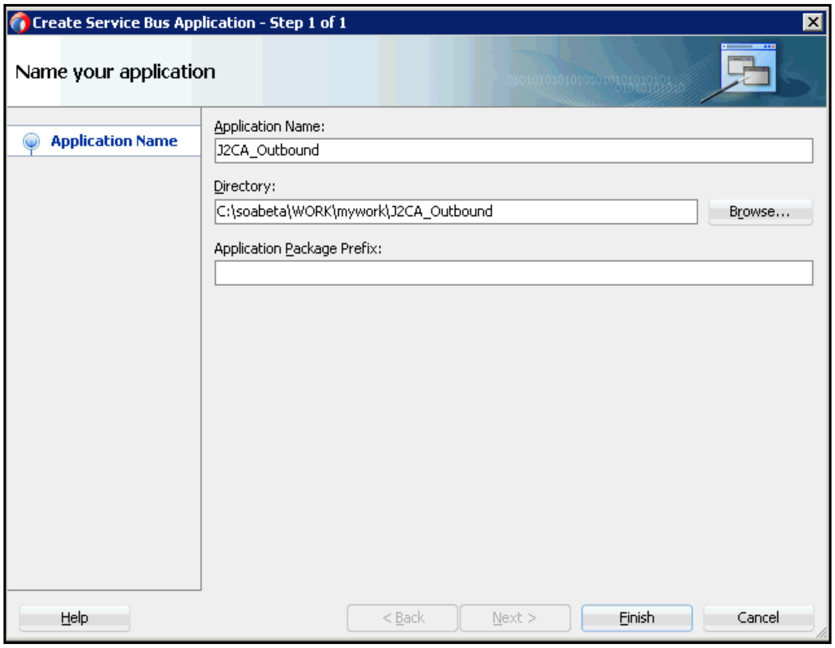
Before you design an OSB outbound process, you must generate the respective WSDL file using Application Explorer. For more information, see Section 4.4.1, "Generating WSDL for Request/Response Service" on page 4-8.

8.1.1 Creating a Service Bus Application for OSB

Perform the following steps in JDeveloper to create a service bus application for OSB.

1. Create a new OSB application.
2. Enter a name for the OSB Application (for example, J2CA_Outbound) and click **Finish**, as shown in Figure 8–1.

Figure 8–1 Name Your Application Pane



Create Service Bus Application - Step 1 of 1

Name your application

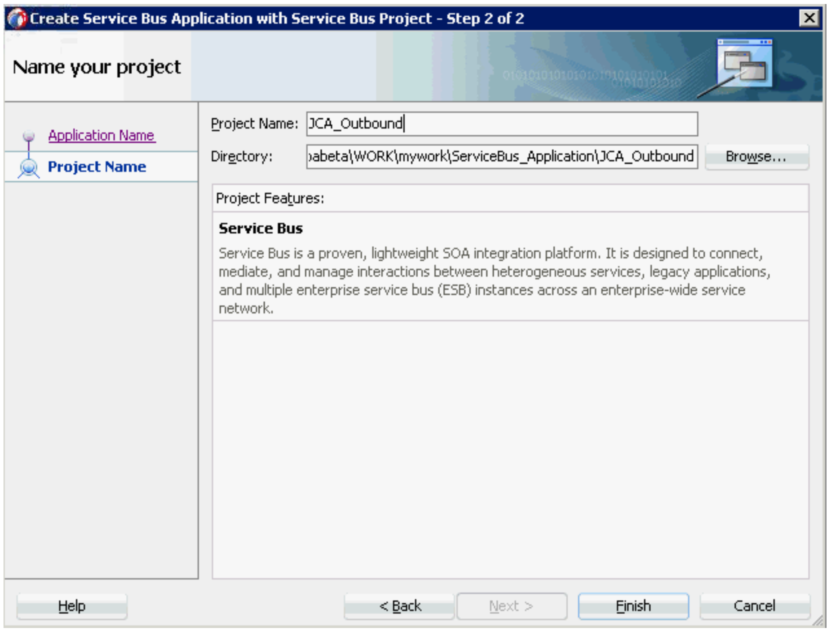
Application Name:
J2CA_Outbound

Directory:
C:\soabeta\WORK\mywork\J2CA_Outbound

Application Package Prefix:

3. Enter a project name (for example, JCA_Outbound), and click **Finish**, as shown in Figure 8–2.

Figure 8–2 Name Your Project Pane



Create Service Bus Application with Service Bus Project - Step 2 of 2

Name your project

Project Name: JCA_Outbound

Directory: ›abeta\WORK\mywork\ServiceBus_Application\JCA_Outbound

Project Features:

Service Bus

Service Bus is a proven, lightweight SOA integration platform. It is designed to connect, mediate, and manage interactions between heterogeneous services, legacy applications, and multiple enterprise service bus (ESB) instances across an enterprise-wide service network.

8.1.2 Defining an OSB Outbound Process

This section describes how to define an OSB outbound process. The following topics are included:

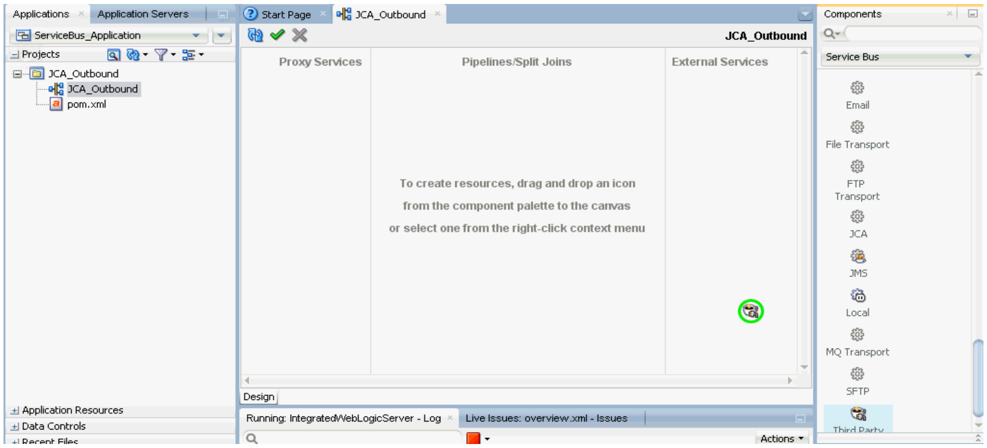
- Section 8.1.2.1, "Configuring a Third-Party Adapter Service Component"
- Section 8.1.2.2, "Configuring a File Transport Type Business Service"
- Section 8.1.2.3, "Creating a Proxy Service With Pipeline"
- Section 8.1.2.4, "Configuring the Routing Rules"

8.1.2.1 Configuring a Third-Party Adapter Service Component

Perform the following steps to create a third party adapter service component along with the Business Service:

1. Drag and drop the **Third Party Adapter** component from the Service Bus Components pane to the External Services pane, as shown in Figure 8–3.

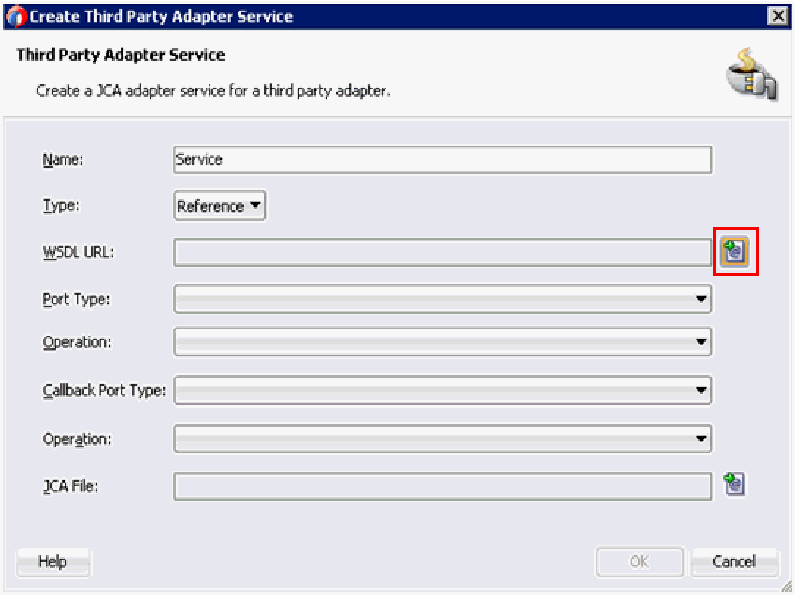
Figure 8–3 Third Party Adapter Component



To create resources, drag and drop an icon from the component palette to the canvas or select one from the right-click context menu

The Create Third Party Adapter Service dialog is displayed, as shown in Figure 8–4.

Figure 8–4 Create Third Party Adapter Service Pane


Create Third Party Adapter Service

Third Party Adapter Service

Create a JCA adapter service for a third party adapter.

Name: Service

Type: Reference

WSDL URL:

Port Type:

Operation:

Callback Port Type:

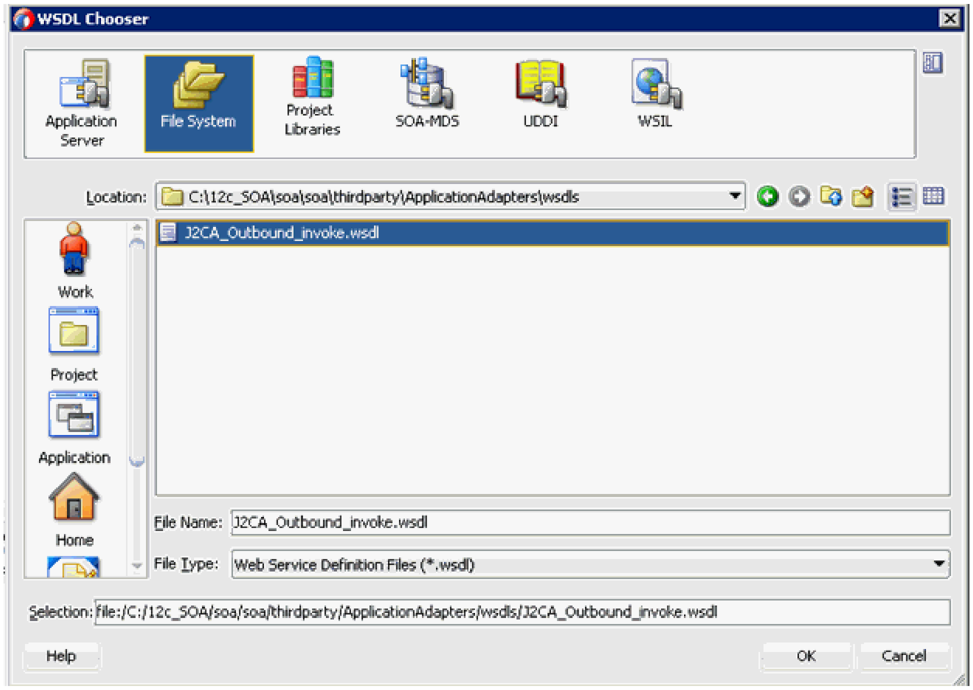
Operation:

JCA File:

2. Enter an appropriate name for the Third Party Adapter Service which will be used as the Business Service name.
3. Ensure that **Reference** is selected from the Type drop-down list (by default).
4. Click the Find existing WSDLs icon, which is located to the right of the WSDL URL field.

   The WSDL Chooser dialog is displayed, as shown in Figure 8–5.

Figure 8–5 WSDL Chooser Dialog



Location: C:\12c_SOA\soa\soa\thirdparty\ApplicationAdapters\wsdls

J2CA_Outbound_invoke.wsdl

File Name: J2CA_Outbound_invoke.wsdl

File Type: Web Service Definition Files (*.wsdl)

Selection: file:/C:/12c_SOA/soa/soa/thirdparty/ApplicationAdapters/wsdls/J2CA_Outbound_invoke.wsdl

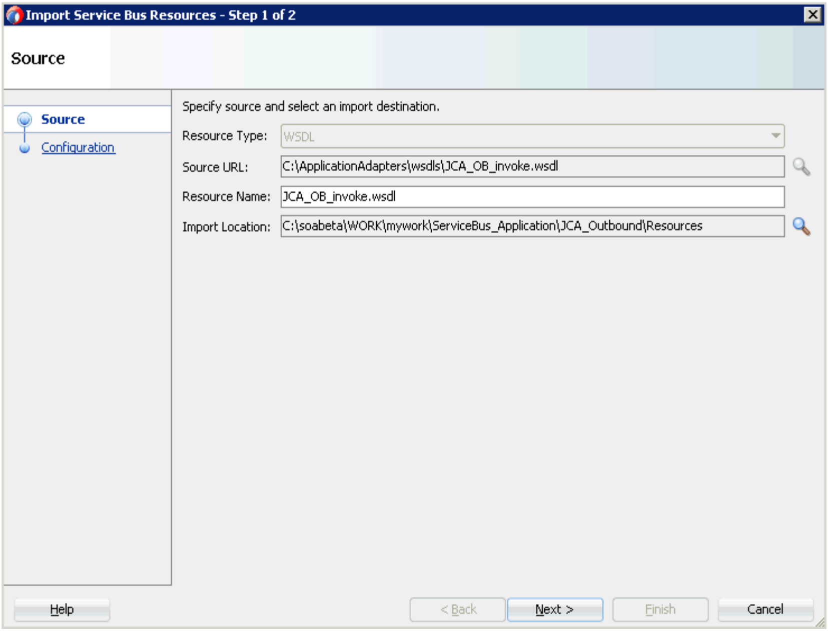
5. Select the **File System** tab, then browse, and select an outbound WSDL file from the WSDL directory.

6. Click **OK**.

   The Import Service Bus Resources dialog is displayed.

7. Click **Next**, as shown in Figure 8–6.

Figure 8–6 Source Pane



Specify source and select an import destination.

Resource Type: WSDL

Source URL: C:\ApplicationAdapters\wsdls\JCA_OB_invoke.wsdl

Resource Name: JCA_OB_invoke.wsdl

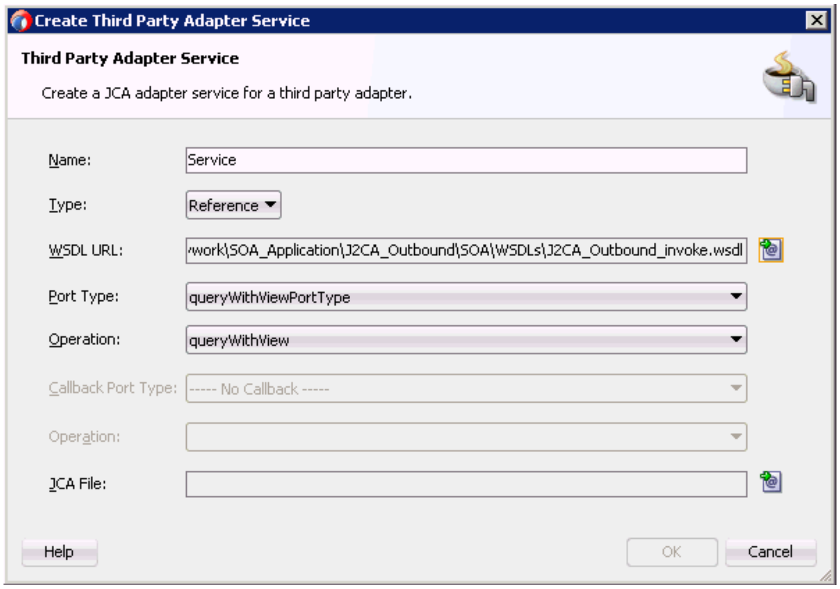
Import Location: C:\soabeta\WORK\mywork\ServiceBus_Application\JCA_Outbound\Resources

8. In the Configuration pane, click **Finish**.

You are returned to the Create Third Party Adapter Service Dialog.

9. Click the Find JCA file icon which is located to the right of the JCA File field, as shown in Figure 8–7.

Figure 8–7 Find JCA File Icon



Create Third Party Adapter Service

Third Party Adapter Service

Create a JCA adapter service for a third party adapter.

Name: Service

Type: Reference

WSDL URL: work\SOA_Application\J2CA_Outbound\SOA\WSDLs\J2CA_Outbound_invoke.wsdl

Port Type: queryWithViewPortType

Operation: queryWithView

Callback Port Type: ----- No Callback -----

Operation:

JCA File:

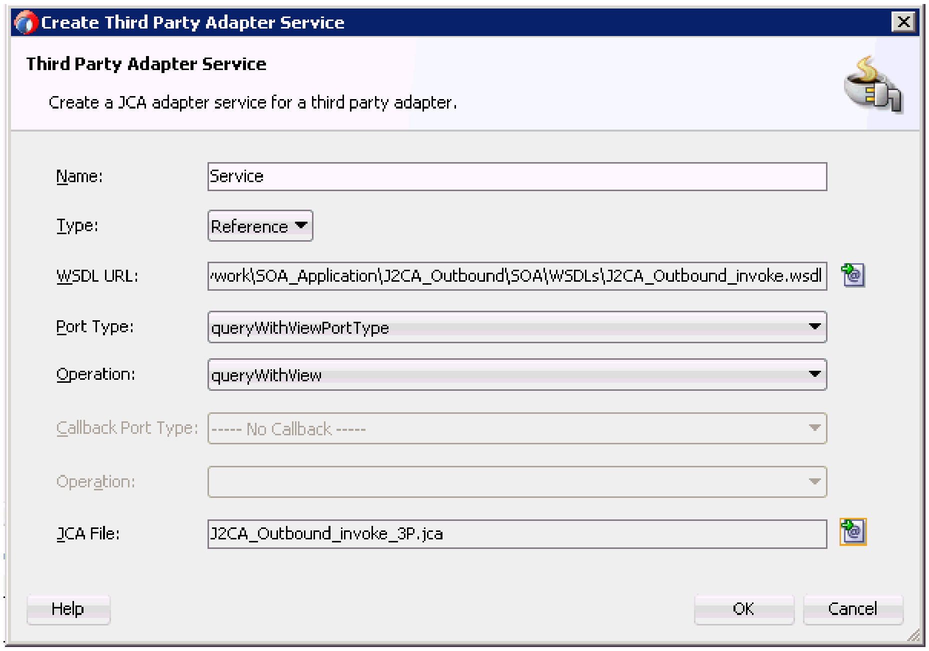
The Transformation Chooser dialog is displayed.

10. Select the JCA properties file from the WSDL directory.
11. Click **OK**. The Copy File message is displayed.
12. Click **Yes**.

    A copy of the JCA properties file is made in the project folder.

    You are returned to the Create Third Party Adapter Service dialog, as shown in Figure 8–8.

Figure 8–8 Create Third Party Adapter Service Dialog



Create Third Party Adapter Service

Third Party Adapter Service

Create a JCA adapter service for a third party adapter.

Name: Service

Type: Reference

WSDL URL: \work\SOA_Application\J2CA_Outbound\SOA\WSDLs\J2CA_Outbound_invoke.wsdl

Port Type: queryWithViewPortType

Operation: queryWithView

Callback Port Type: ----- No Callback -----

Operation:

JCA File: J2CA_Outbound_invoke_3P.jca

13. Click **OK**.

    The Business service component is created in the External Services pane.

8.1.2.2 Configuring a File Transport Type Business Service

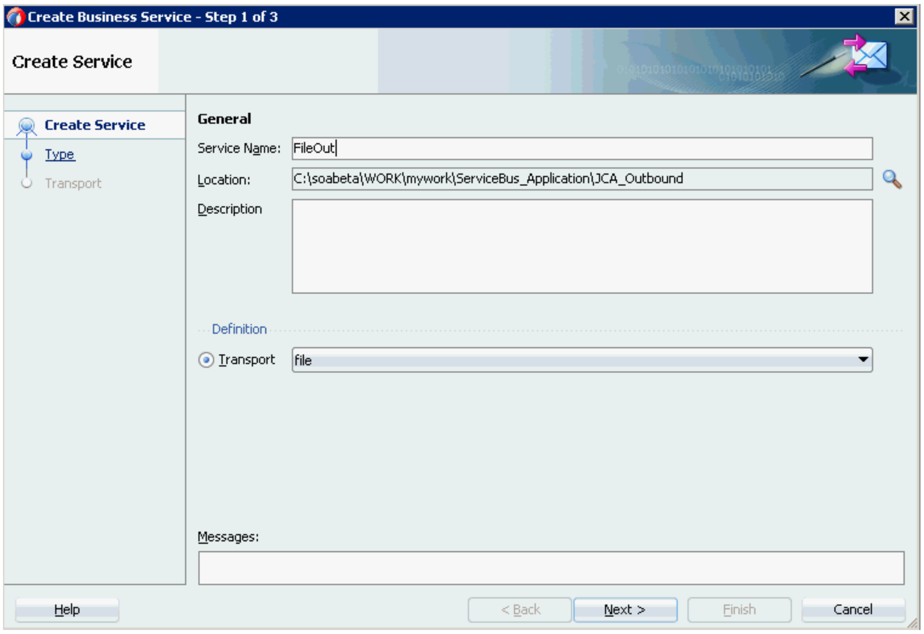
Perform the following steps to create a File Transport Business Service:

1. Drag and drop the **File Transport** component from the Advanced pane to the External Services pane.

   The Create Business Service dialog is displayed.

2. In the Service Name field, enter any name you wish for the Business Service (for example, FileOut), and click **Next**, as shown in Figure 8–9.

Figure 8–9 Create Service Pane



Create Business Service - Step 1 of 3

Create Service

General

Service Name: FileOut

Location: C:\soabeta\WORK\mywork\ServiceBus_Application\JCA_Outbound

Description

Definition

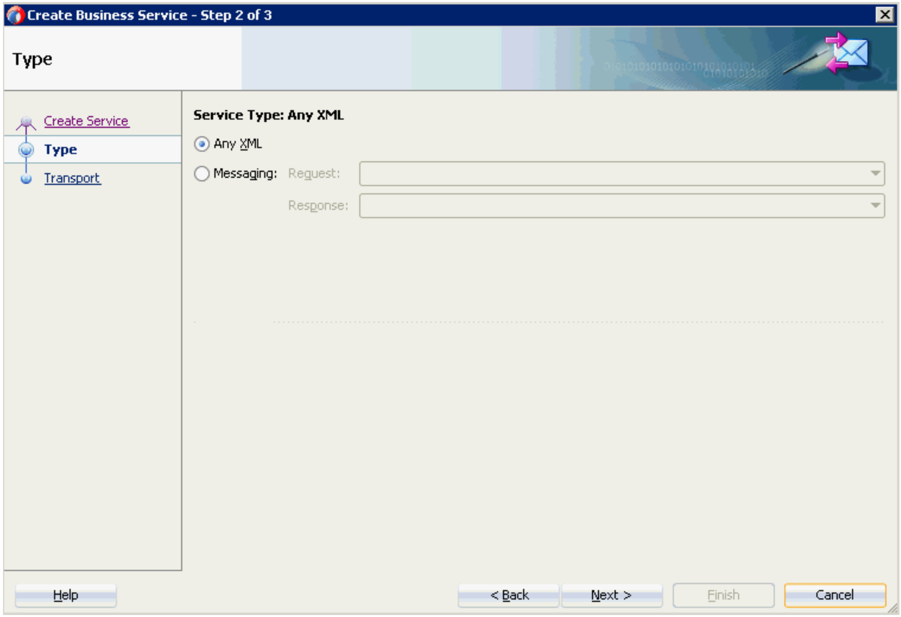
● Transport file

Messages:

The Type pane is displayed. The **Any XML** option is selected by default.

3. Click **Next**, as shown in Figure 8–10.

Figure 8–10 Type Pane



Create Business Service - Step 2 of 3

Type

Service Type: Any XML

● Any XML

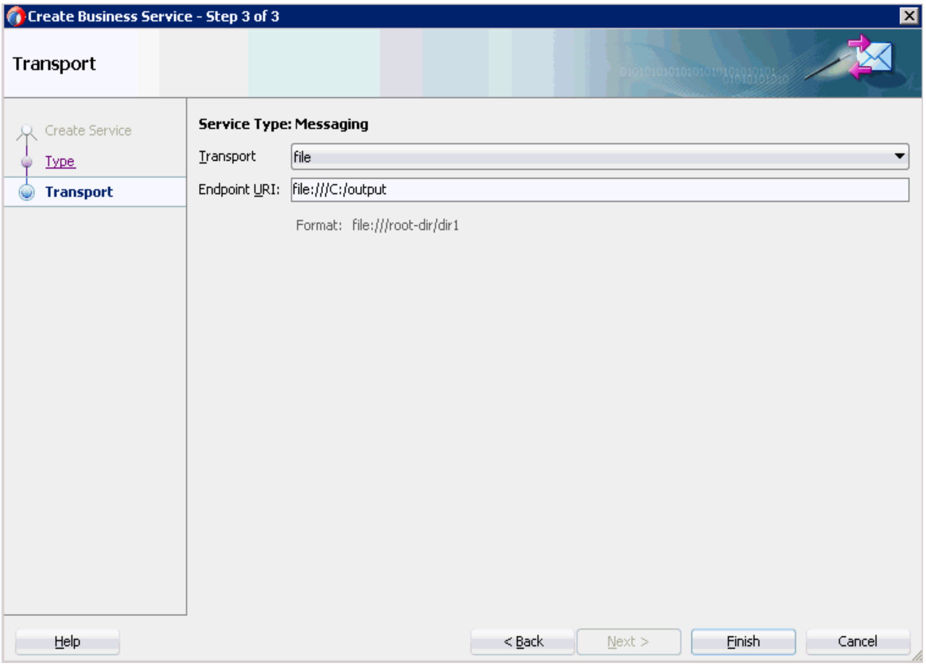
○ Messaging: Request:

Response:

The Transport pane appears.

4. Provide the output location in the Endpoint URI field (for example, c:/output) and click **Finish**, as shown in Figure 8–11.

Figure 8–11 Transport Pane



Create Business Service - Step 3 of 3

Transport

Service Type: Messaging

Transport: file

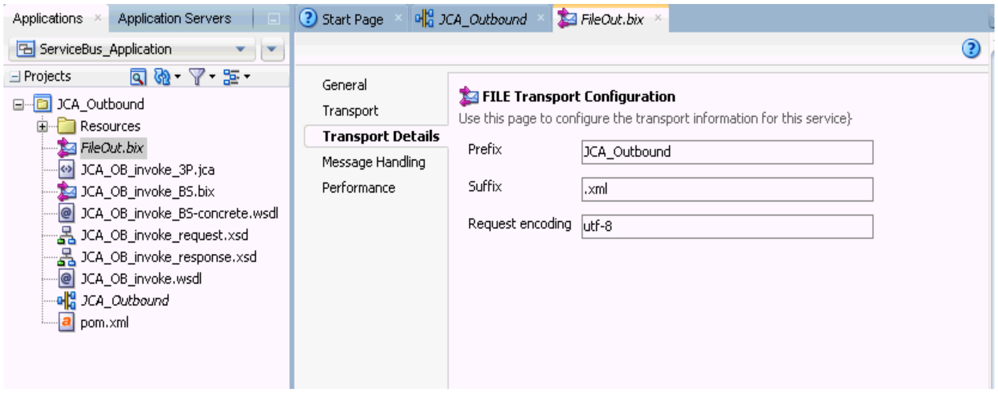
Endpoint URI: file:///C:/output

Format: file:///root-dir/dir1

The File Transport Business service Fileout is created and displayed.

5. Double-click the created Business service **Fileout** and provide the values for the Prefix and Suffix fields in the Transport Details Tab, as shown in Figure 8–12.

Figure 8–12 Transport Details



FILE Transport Configuration

Use this page to configure the transport information for this service}

Prefix: JCA_Outbound

Suffix: .xml

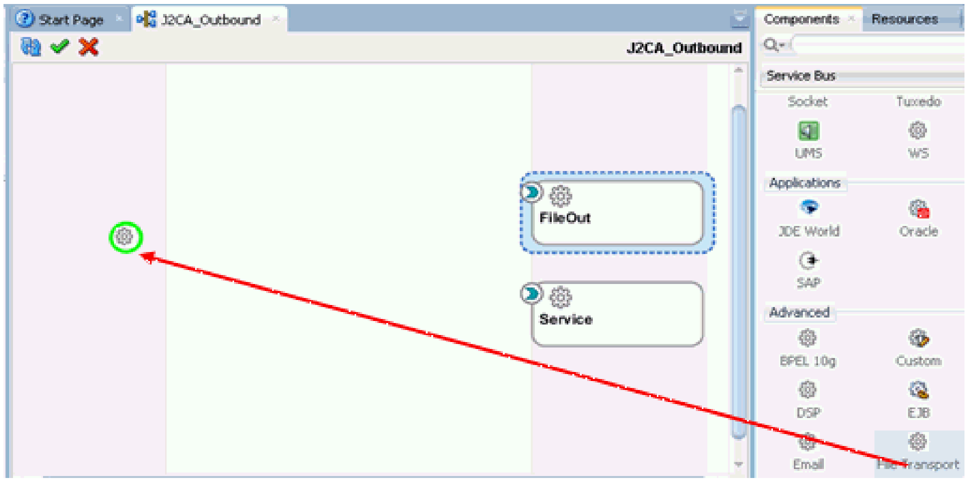
Request encoding: utf-8

8.1.2.3 Creating a Proxy Service With Pipeline

Perform the following steps to create a Proxy Service with Pipeline:

1. Drag and drop the **File Transport** component from the Advanced Components pane to the Proxy Services pane, as shown in Figure 8–13.

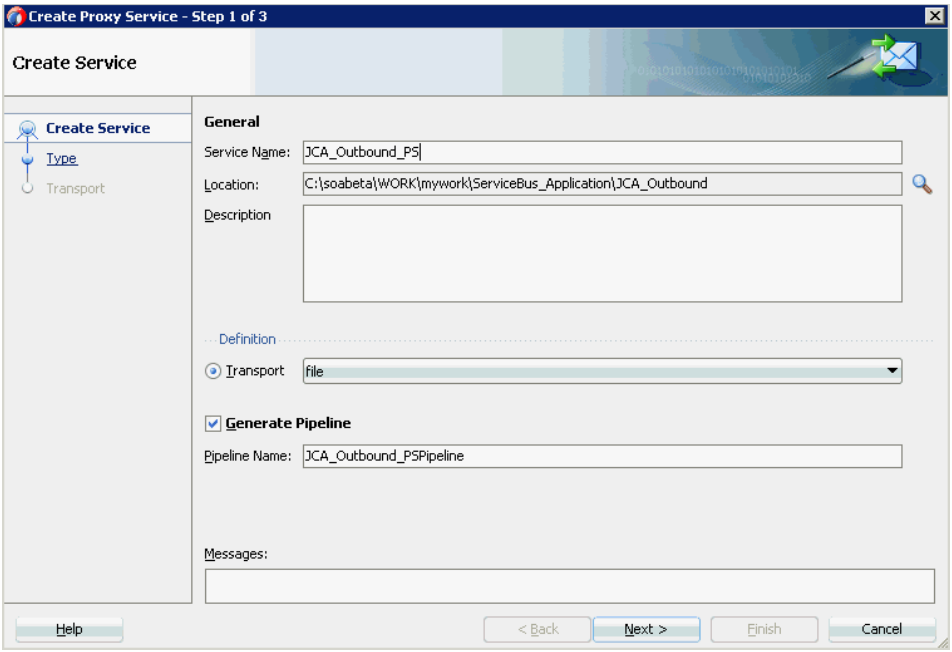
Figure 8–13 File Transport Component



The Create Proxy Service pane is displayed.

2. In the Service Name field, enter any name you wish for the Proxy service (for example, JCA_Outbound_PS). By default, **Generate Pipeline** is selected.
3. Click **Next**, as shown in Figure 8–14.

Figure 8–14 Create Service Pane


Create Proxy Service - Step 1 of 3

Create Service

General

Service Name: JCA_Outbound_PS

Location: C:\soabeta\WORK\mywork\ServiceBus_Application\JCA_Outbound

Description

Definition

● Transport: file

☑ Generate Pipeline

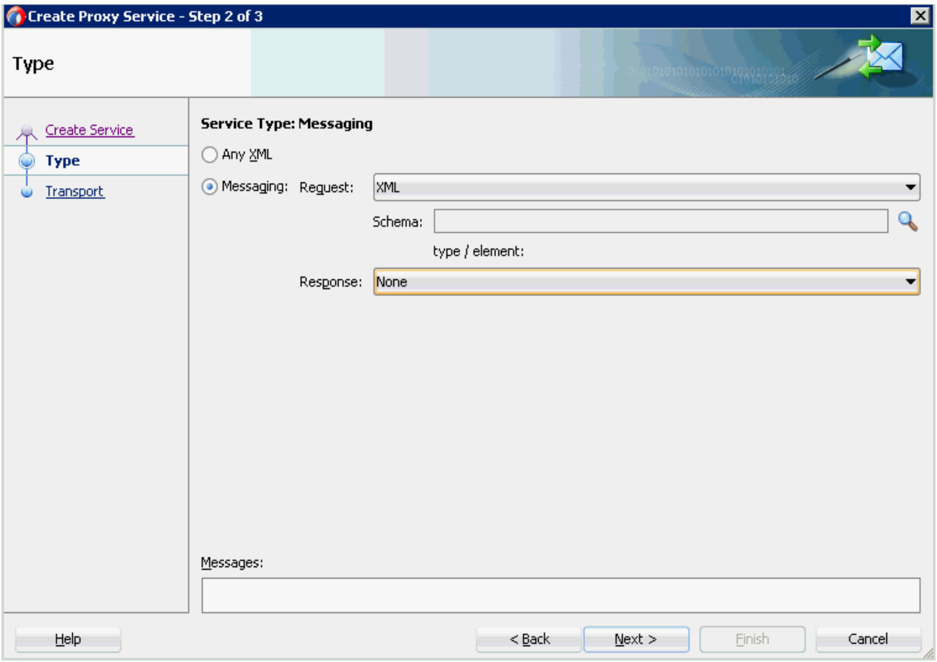
Pipeline Name: JCA_Outbound_PSPipeline

Messages:

The Type pane is displayed.

4. Select the **Messaging** option, set the Request to **XML** and Response as **None**, and then click **Next**, as shown in Figure 8–15.

Figure 8–15 Type Pane



Create Proxy Service - Step 2 of 3

Type

Service Type: Messaging

○ Any XML

● Messaging: Request: XML

Schema:

type / element:

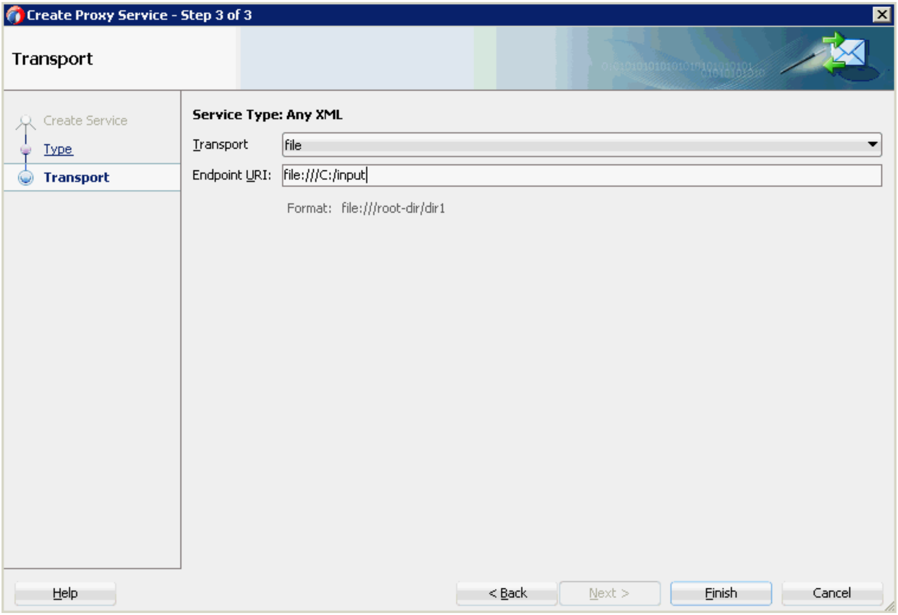
Response: None

Messages:

The Transport window is displayed.

5. Provide the input location in the Endpoint URI field (for example, c:/input) and click **Finish**, as shown in Figure 8–16.

Figure 8–16 Transport Window



Create Proxy Service - Step 3 of 3

Transport

Service Type: Any XML

Transport: file

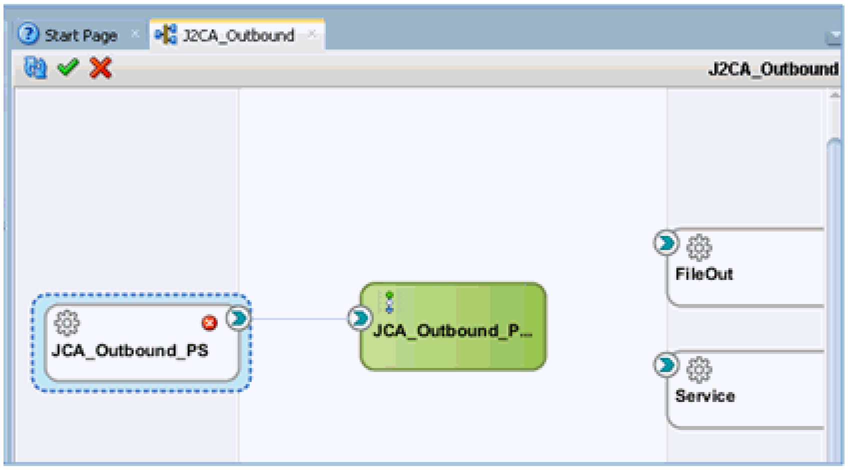
Endpoint URI: file:///C:/input

Format: file:///root-dir/dir1

The Proxy service along with the pipeline is created and displayed.

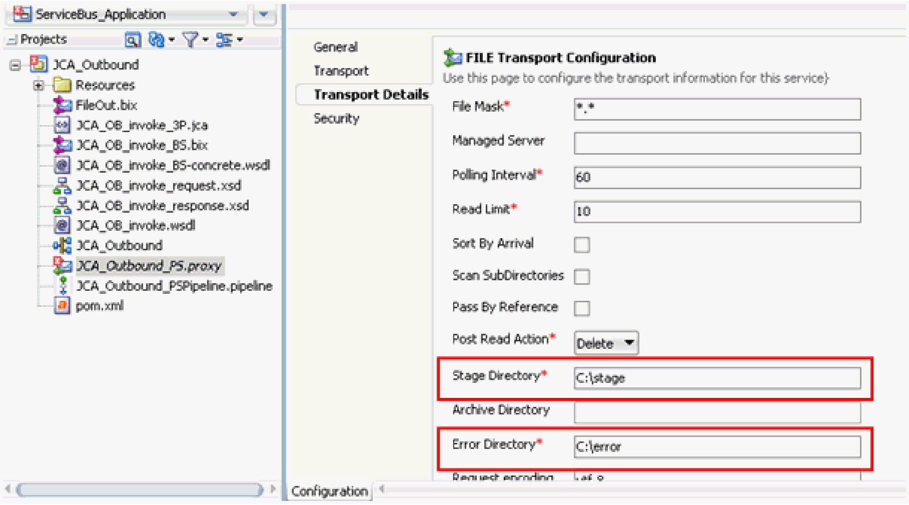
6. Double-click the created Proxy Service (for example: JCA_Outbound_PS), as shown inFigure 8–17.

Figure 8–17 Proxy Service Edit



7. In the displayed Proxy Service configuration page, select **Transport Details** and provide the values for Stage and Error Directory, as shown in Figure 8–18.

Figure 8–18 File Transport Configuration



FILE Transport Configuration

Use this page to configure the transport information for this service)

File Mask*	*.*
Managed Server	
Polling Interval*	60
Read Limit*	10
Sort By Arrival	□
Scan SubDirectories	□
Pass By Reference	□
Post Read Action*	Delete
Stage Directory*	C:\stage
Archive Directory	
Error Directory*	C:\error
Request encoding	utf-8

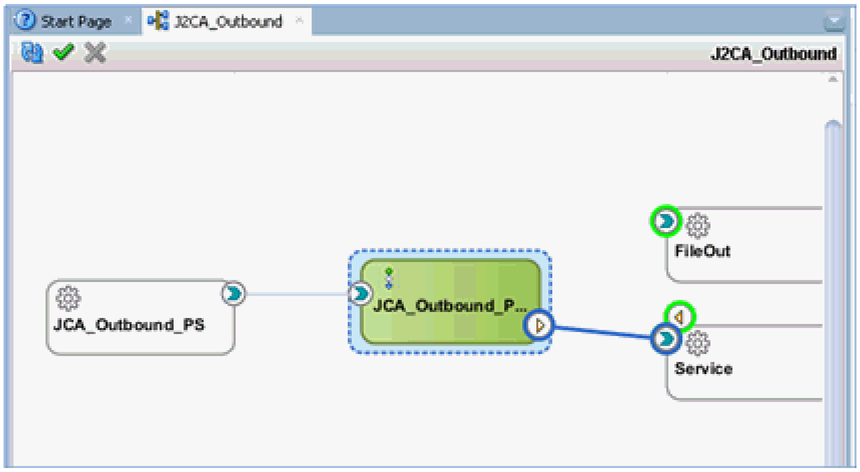
8. Save and close the Proxy Service configuration page.

8.1.2.4 Configuring the Routing Rules

Perform the following steps to configure the routing rules:

1. Connect the Pipeline to the Business Service (for example, Service) as shown in Figure 8–19.

Figure 8–19 Business Service Pipeline

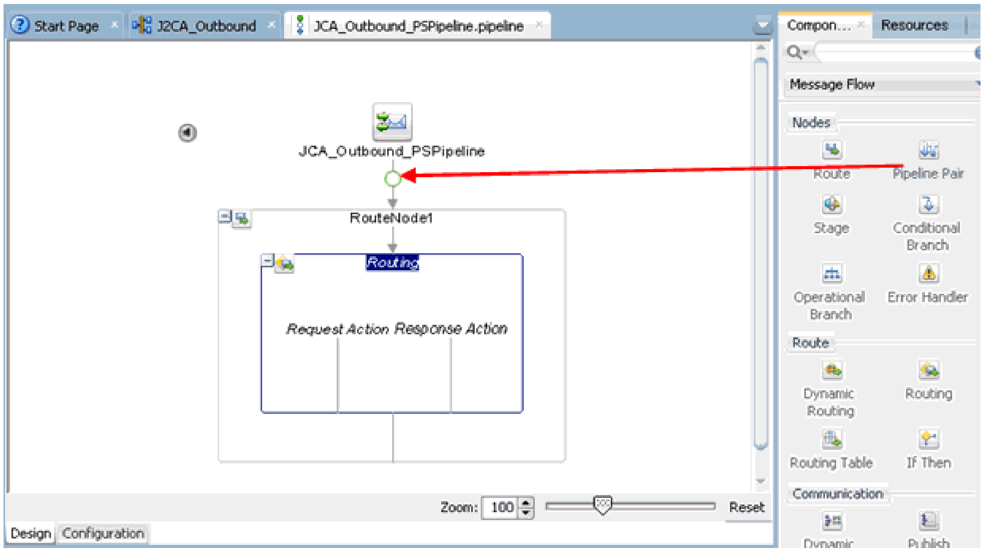


2. Double-click on the pipeline (for example, JCA_Outbound_PSPipeline) in the Pipelines/Split Joins pane.

   The Pipeline configuration page is displayed.

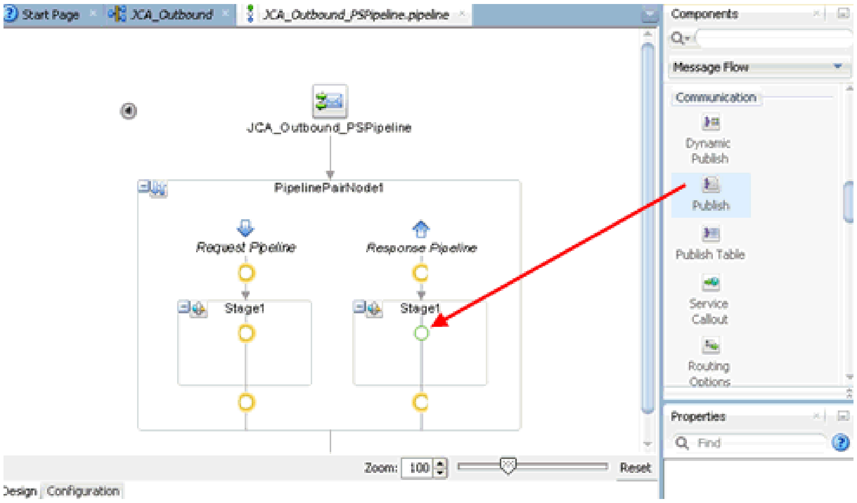
3. Drag and drop the **Pipeline Pair** node from Nodes pane to the area below the Pipeline (for example: JCA_Outbound_PSPipeline), as shown in Figure 8–20.

Figure 8–20 Pipeline Pair Node



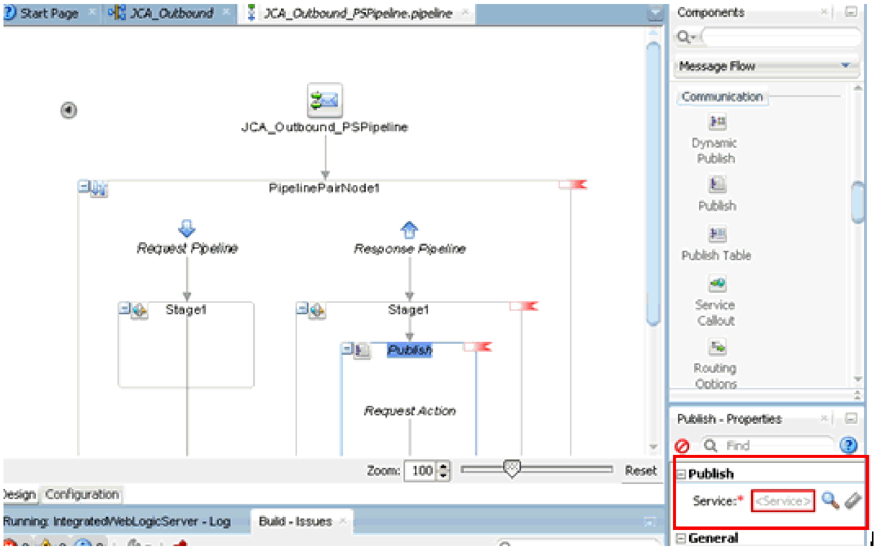
4. Drag and drop the **Publish** node from the Communication pane to the area beneath Stage1 of the Response Pipeline, as shown in Figure 8–21.

Figure 8–21 Publish Node



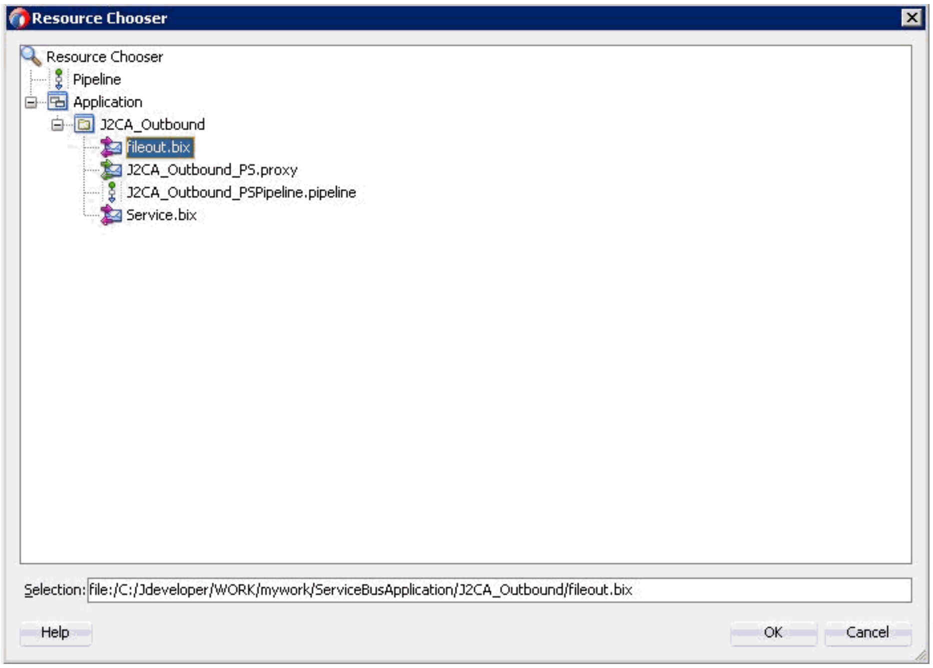
5. Click on the browse icon to the right of the Service field in the right pane of Publish Properties, as shown in Figure 8–22.

Figure 8–22 Browse Icon



6. In the displayed Resource Chooser window, select the **Fileout.bix** File Transport Business service and click **OK**, as shown in Figure 8–23.

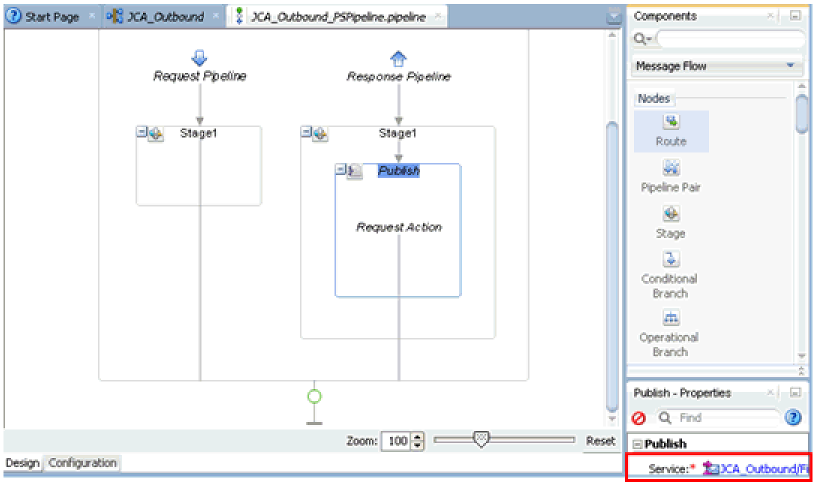
Figure 8–23 Resource Chooser



Selection: file:/C:/Jdeveloper/WORK/mywork/ServiceBusApplication/J2CA_Outbound/fileout.bix

In the right pane, the selected service is configured in the Publish pane, as shown in Figure 8–24.

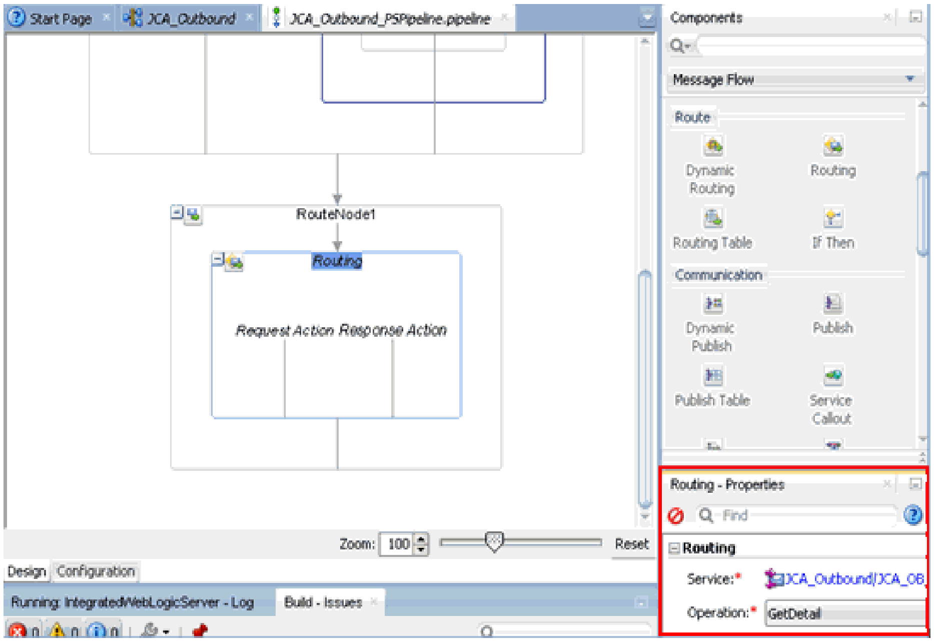
Figure 8–24 Publish Pane



Service:* JCA_Outbound/Fi

7. Click on the Routing to verify the Service is selected properly, as shown in Figure 8–25.

Figure 8–25 Pipeline Configuration



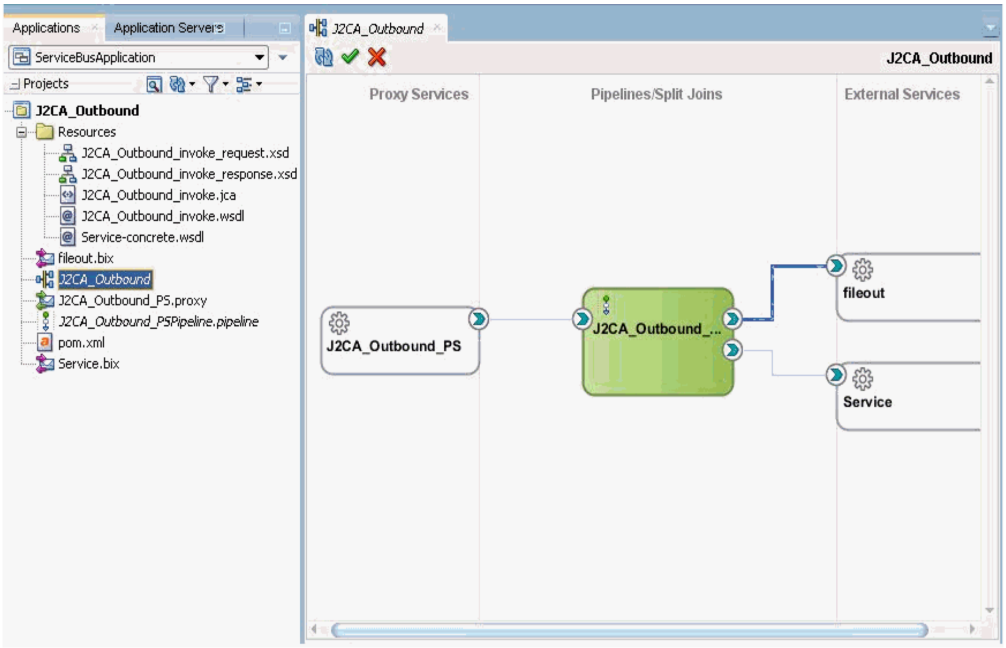
Service:* JCA_Outbound/JCA_OB

Operation:* GetDetail

8. Save and Close the Pipeline configuration page.

9. Double-click the overview.xml file (for example: JCA_Outbound), and click **Save All** in the menu bar to save the OSB process, as shown in Figure 8–26.

Figure 8–26 Save All Icon

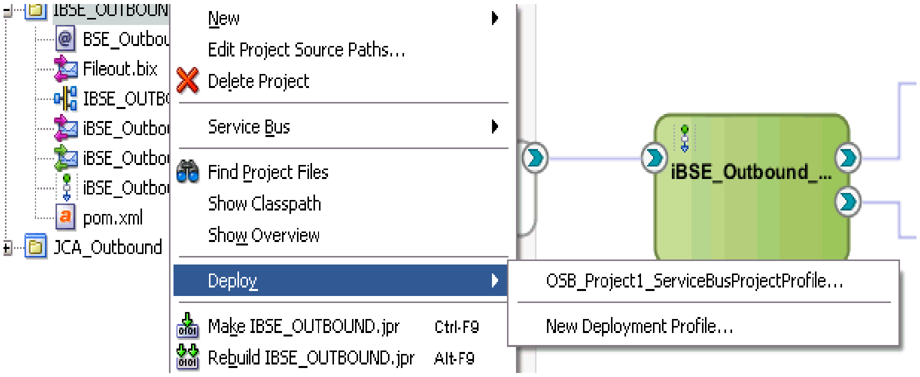


8.1.3 Deploying the OSB Outbound Process

Perform the following steps to deploy the OSB outbound process.

1. Right-click the OSB project, select **Deploy**, and then select **OSB_Project1_ServiceBusProjectProfile...**, as shown in Figure 8–27.

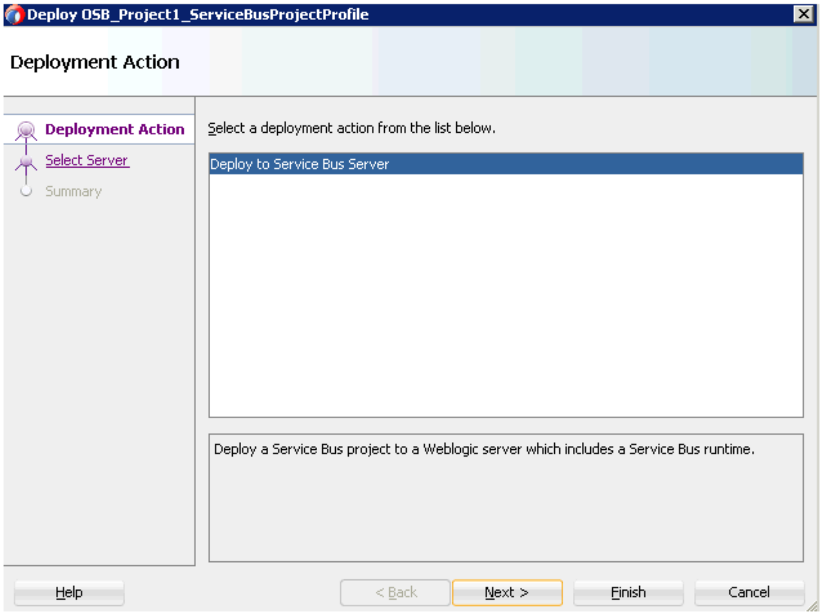
Figure 8–27 Deploy Option



The Deployment Action page is displayed.

2. Click **Next**, as shown in Figure 8–28.

Figure 8–28 Deployment Action Page



Select a deployment action from the list below.

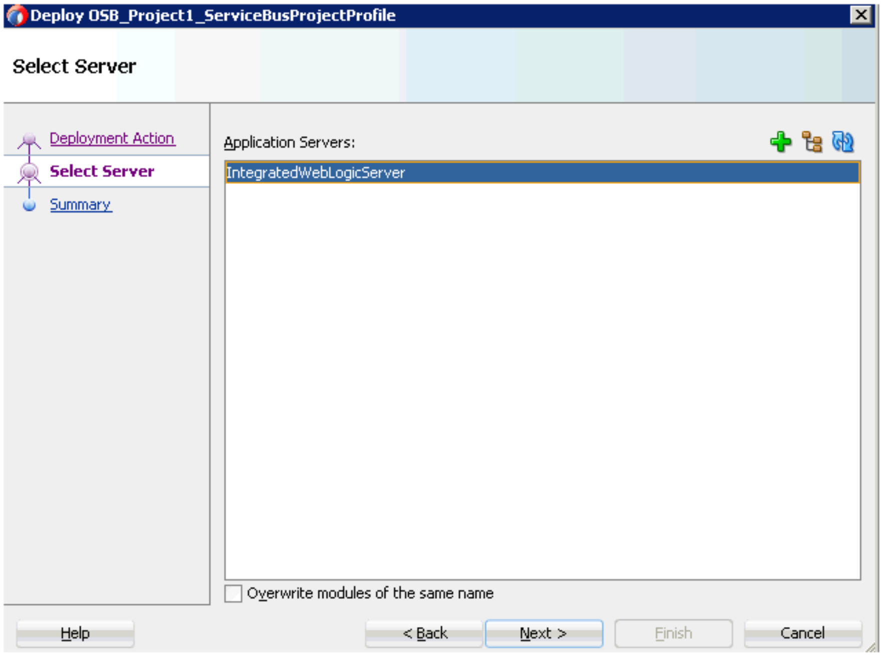
Deploy to Service Bus Server

Deploy a Service Bus project to a Weblogic server which includes a Service Bus runtime.

The Select Server page is displayed.

3. Select an available application server that was configured and click **Next**, as shown in Figure 8–29.

Figure 8–29 Select Server Page



Deploy OSB_Project1_ServiceBusProjectProfile

Select Server

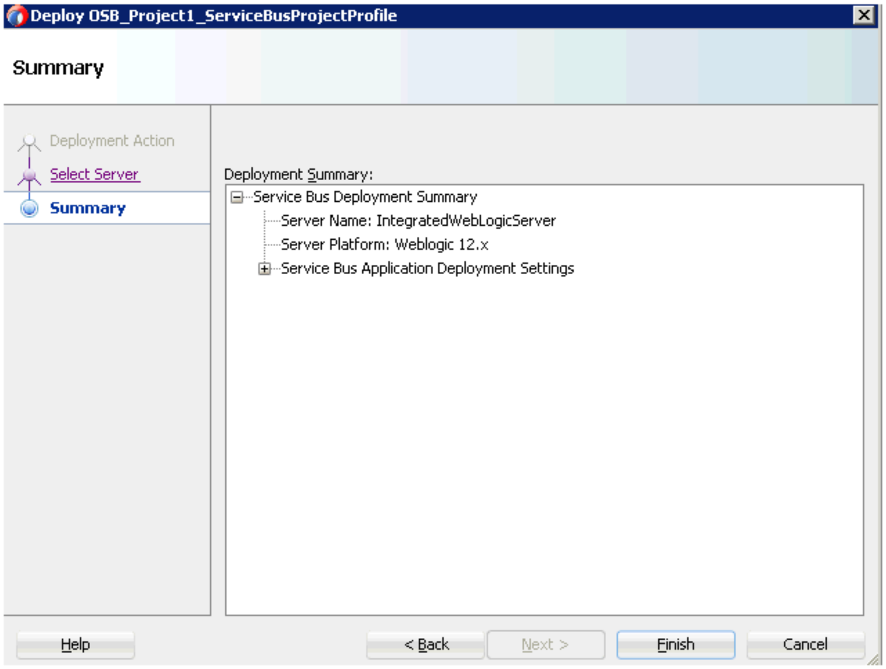
Application Servers:

IntegratedWebLogicServer

☐ Overwrite modules of the same name

The Summary page is displayed, as shown in Figure 8–30.

Figure 8–30 Summary Page



Deploy OSB_Project1_ServiceBusProjectProfile

Summary

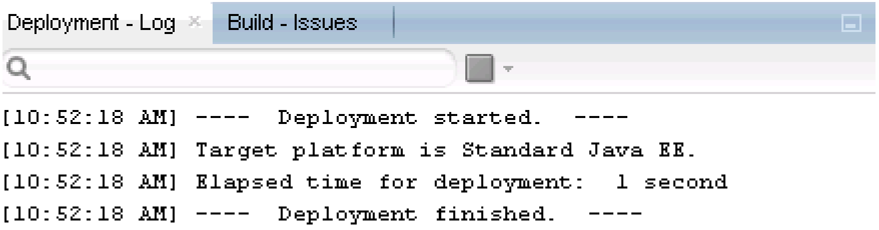
Deployment Summary:

- Service Bus Deployment Summary
  - Server Name: IntegratedWebLogicServer
  - Server Platform: Weblogic 12.x
  - Service Bus Application Deployment Settings

4. Review and verify all the available deployment information for your project and click **Finish**.

   The process is deployed successfully, as shown in Figure 8–31.

Figure 8–31 Successful Deployment Message



```
[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. Copy and paste an input XML file in the input folder you have configured (for example, C:\input).

   The output is received in the configured output location (for example, C:\output).

8.2 Configuring an OSB Inbound Process Using JDeveloper (J2CA Configuration)

This section describes how to configure an OSB inbound process to your Siebel system, using Oracle JDeveloper for J2CA configurations.

A sample project has been provided for this inbound use case scenario in the following folder of the Application Adapters installation:

```
<ADAPTER_HOME>\etc\sample\SIEBEL_Samples.zip\SIEBEL_Samples\OSB_
Jdeveloper\J2CA\Siebel_Sample_J2CA_OSB_Inbound_Project
```

This section includes the following topics:

- Section 8.2.1, "Creating a Service Bus Application for OSB"
- Section 8.2.2, "Defining an OSB Inbound Process"
- Section 8.2.3, "Deploying the OSB Inbound Process"

Prerequisites

Before you design an OSB inbound process, you must generate the respective WSDL file using Application Explorer. For more information, see Section 4.5.1, "Generating WSDL for Event Integration" on page 4-34.

8.2.1 Creating a Service Bus Application for OSB

To configure an OSB inbound process, you must create service bus application for OSB. For more information, see Section 8.1.1, "Creating a Service Bus Application for OSB" on page 8-2.

8.2.2 Defining an OSB Inbound Process

This section describes how to define an OSB inbound process. The following topics are included:

- Section 8.2.2.1, "Configuring a Third-Party Adapter Service Component"
- Section 8.2.2.2, "Creating a Pipeline"
- Section 8.2.2.3, "Configuring a File Transport Type Business Service"

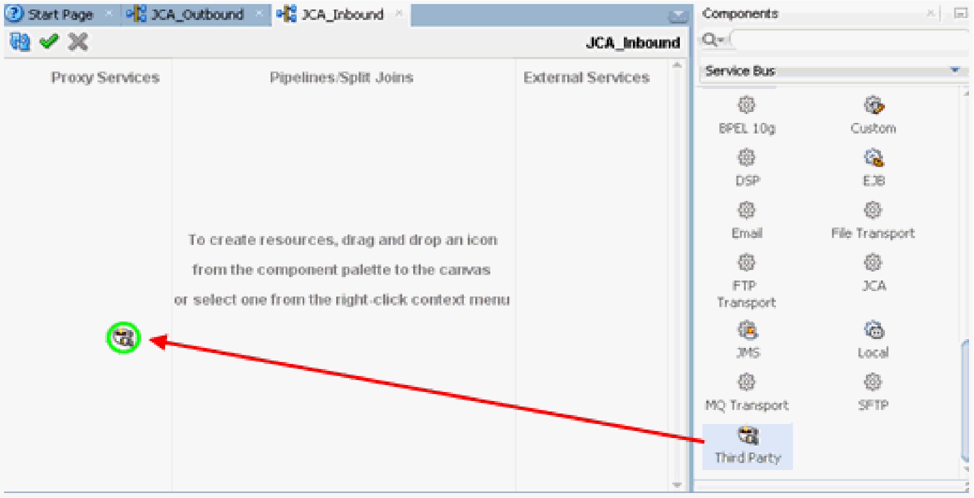
- Section 8.2.2.4, "Configuring the Routing Rules"

8.2.2.1 Configuring a Third-Party Adapter Service Component

Perform the following steps to create a third party adapter service component:

1. Drag and drop the **Third Party** adapter component from the Service Bus Components Pane to the Proxy Services, as shown in Figure 8–32.

Figure 8–32 Third Party Adapter Service Component



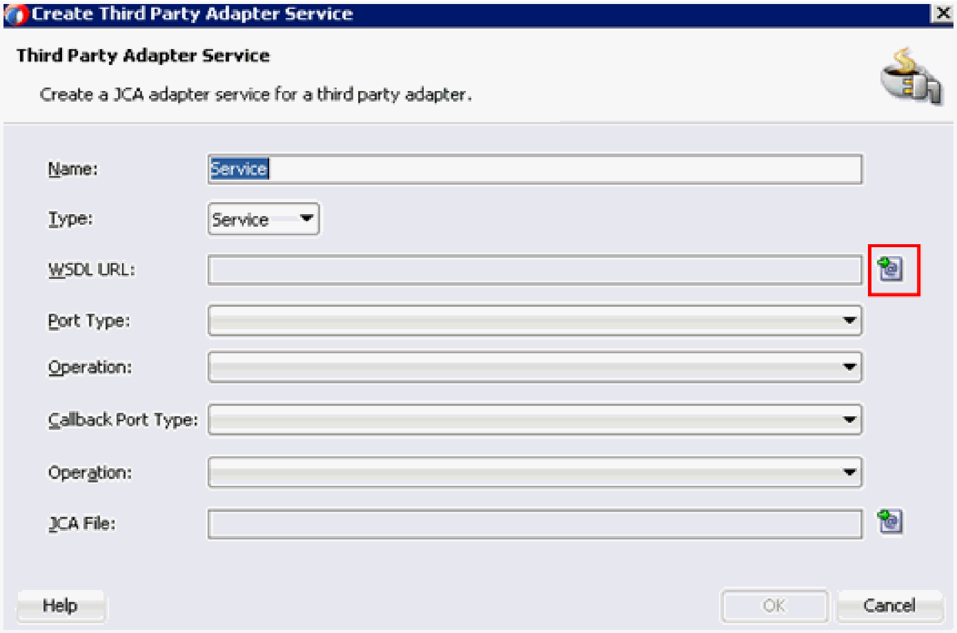
   The Create Third Party Adapter Service dialog is displayed.

2. Enter any name you wish for the Third Party Adapter Service or leave it to the default value.

3. Ensure that **Service** is selected from the Type drop-down list (by default).

4. Click the Find existing WSDLs icon, which is located to the right of the WSDL URL field, as shown in Figure 8–33.

Figure 8–33 Third Party Adapter Service Dialog



Create Third Party Adapter Service

Third Party Adapter Service

Create a JCA adapter service for a third party adapter.

Name: Service

Type: Service

WSDL URL:

Port Type:

Operation:

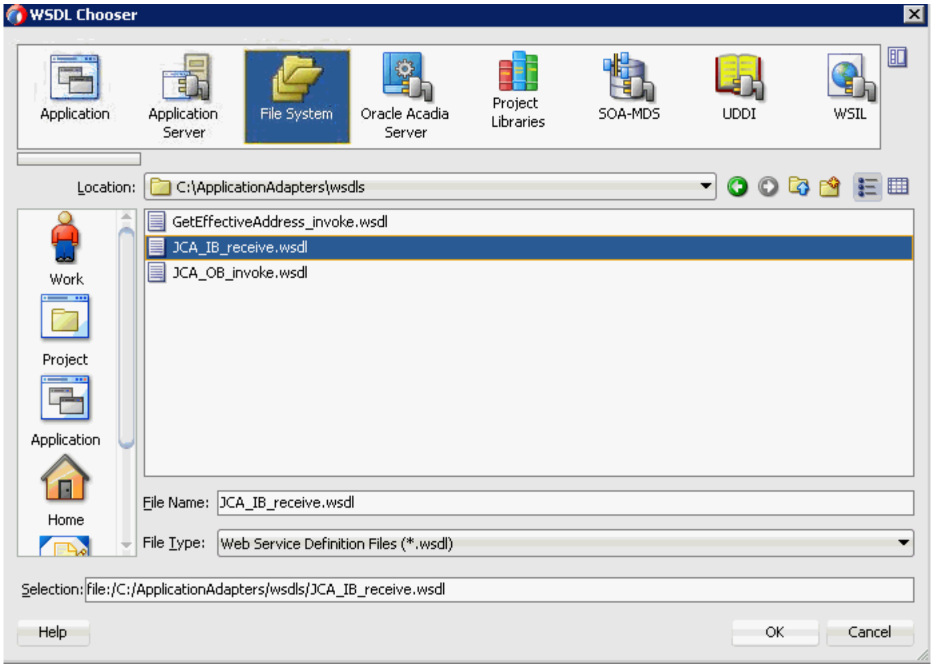
Callback Port Type:

Operation:

JCA File:

The WSDL Chooser dialog is displayed, as shown in Figure 8–34.

Figure 8–34 WSDL Chooser Dialog



Location: C:\ApplicationAdapters\wsdls

GetEffectiveAddress_invoke.wsdl

JCA_IB_receive.wsdl

JCA_OB_invoke.wsdl

File Name: JCA_IB_receive.wsdl

File Type: Web Service Definition Files (*.wsdl)

Selection: file:/C:/ApplicationAdapters/wsdls/JCA_IB_receive.wsdl

5. Select the File system folder, then browse and select an inbound WSDL file from the WSDL directory.
6. Click **OK**.

   The Import Service Bus Resources dialog is displayed.
7. Click **Next**.

8. In the Configuration window, click **Finish**.

   You are returned to the Create Third Party Adapter Service dialog.

9. Click the Find JCA file icon, which is located to the right of the JCA File field.

   The Transformation Chooser dialog is displayed.

10. Select the JCA properties file from the WSDL directory.

11. Click **OK**.

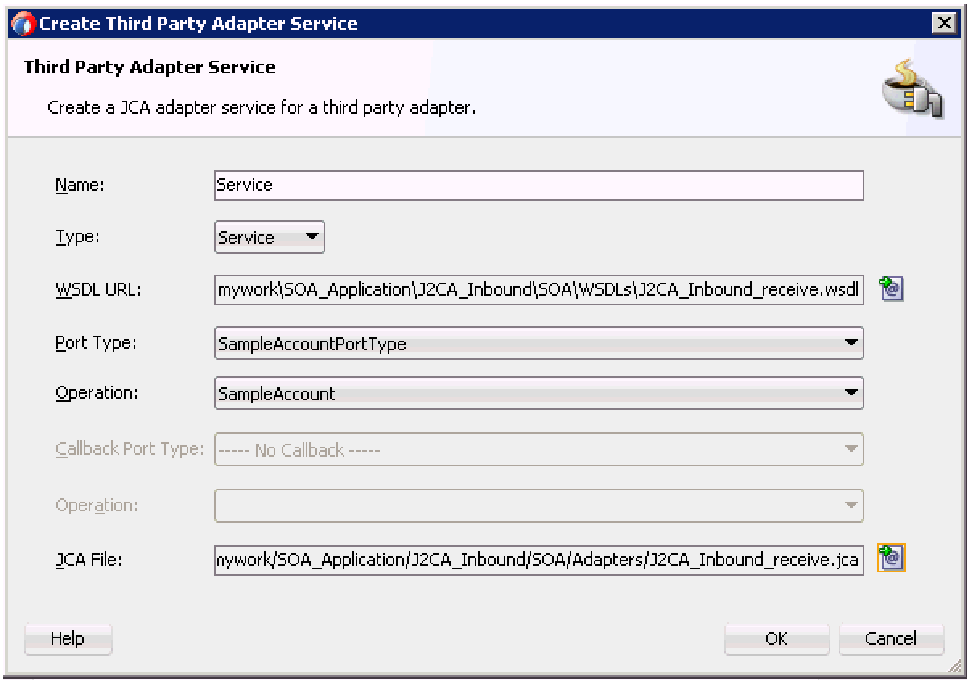
    The Copy File message is displayed.

12. Click **Yes**.

    A copy of the JCA properties file is created in the project folder.

    You are returned to the Create Third Party Adapter Service dialog, as shown in Figure 8–35.

Figure 8–35 Create Third Party Adapter Service Dialog



Create Third Party Adapter Service

Third Party Adapter Service

Create a JCA adapter service for a third party adapter.

Name: Service

Type: Service

WSDL URL: mywork\SOA_Application\J2CA_Inbound\SOA\WSDLs\J2CA_Inbound_receive.wsdl

Port Type: SampleAccountPortType

Operation: SampleAccount

Callback Port Type: ----- No Callback -----

Operation:

JCA File: nywork/SOA_Application/J2CA_Inbound/SOA/Adapters/J2CA_Inbound_receive.jca

13. Click **OK**.

    The third party adapter service component is created in the Proxy Services pane.

8.2.2.2 Creating a Pipeline

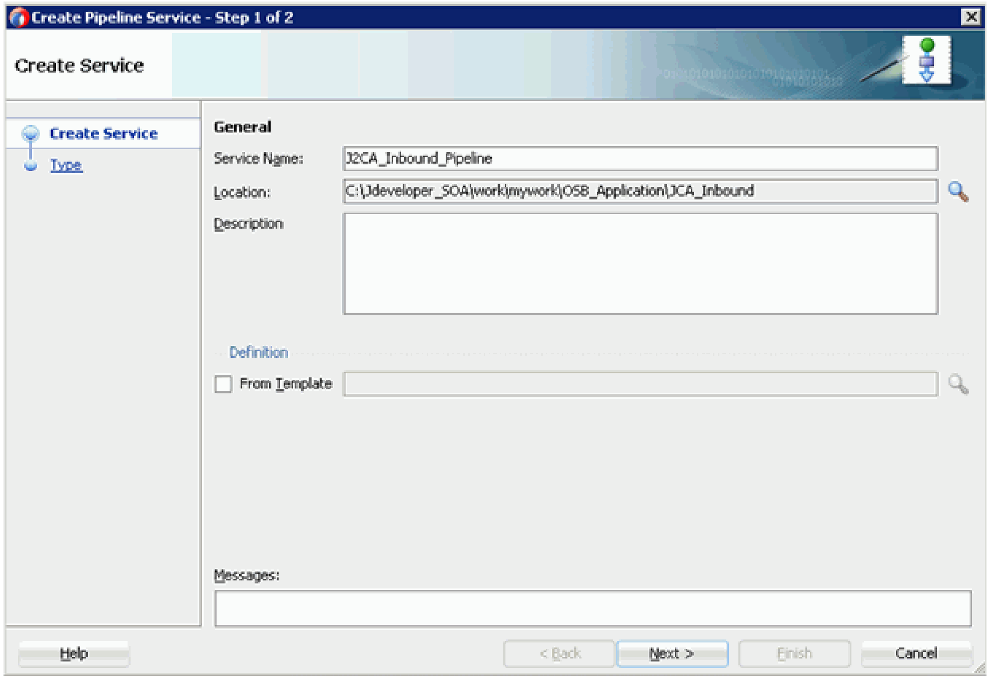
Perform the following steps to generate inbound proxy service with Pipeline:

1. Under Service Bus, click **Resources**.

2. Drag and drop the Pipeline to the Pipelines/Split Joins pane.

3. Provide a name for the Pipeline and click next, as shown in Figure 8–36.

Figure 8–36 Create Service Page



Service Name: J2CA_Inbound_Pipeline

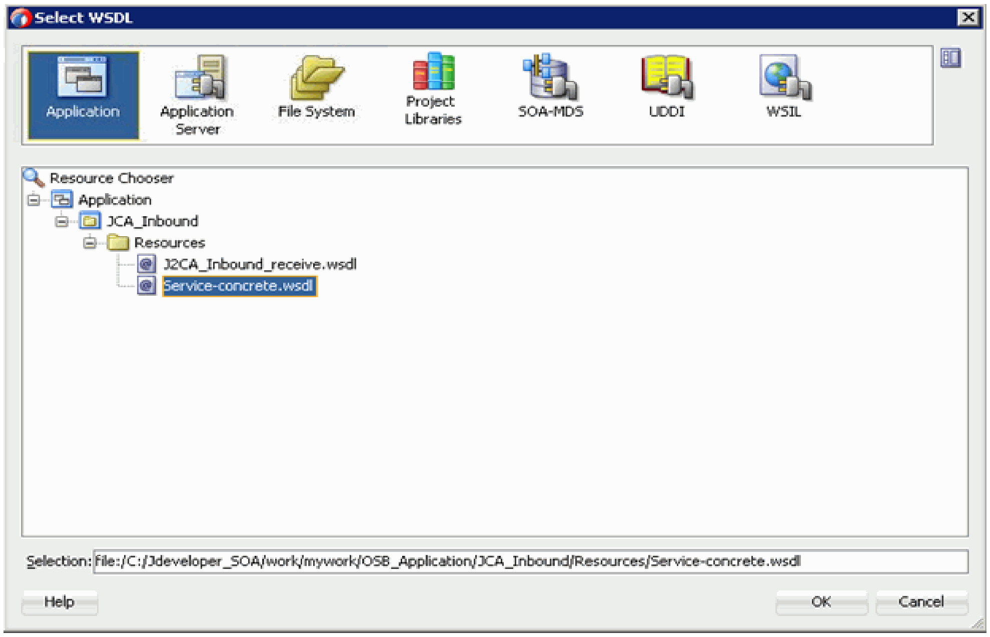
Location: C:\Jdeveloper_SOA\work\mywork\OSB_Application\JCA_Inbound

□ From Template

4. In the Create Pipeline Service window, select the **WSDL** option and click on the WSDL URL.

5. Select **Application** in the WSDL chooser window, then select **service-concrete.wsdl** in the appropriate OSB project, and then click **OK**, as shown in Figure 8–37.

Figure 8–37 Select WSDL Page

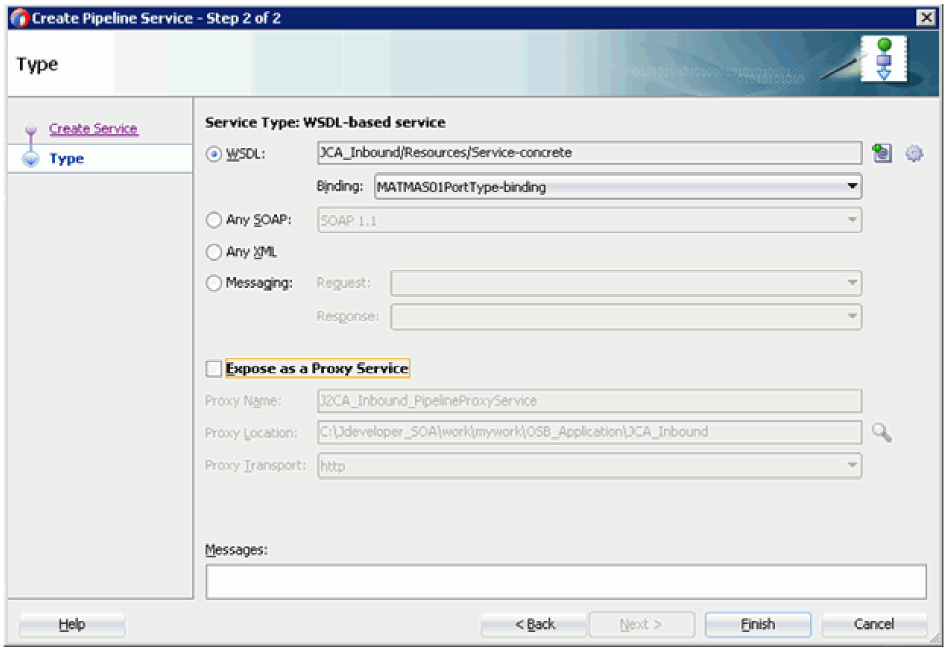


Resource Chooser
- Application
  - JCA_Inbound
    - Resources
      - J2CA_Inbound_receive.wsdl
      - Service-concrete.wsdl

Selection: file:/C:/Jdeveloper_SOA/work/mywork/OSB_Application/JCA_Inbound/Resources/Service-concrete.wsdl

6. Clear the Expose as a Proxy Service check box and click **Finish**, as shown in Figure 8–38.

Figure 8–38 Type Page



Create Pipeline Service - Step 2 of 2

Type

Create Service
Type

Service Type: WSDL-based service

● WSDL: JCA_Inbound/Resources/Service-concrete

Binding: MATMAS01PortType-binding

○ Any SOAP: SOAP 1.1

○ Any XML

○ Messaging: Request: Response:

□ Expose as a Proxy Service

Proxy Name: J2CA_Inbound_PipelineProxyService

Proxy Location: C:\Jdeveloper_SOA\work\mywork\OSB_Application\JCA_Inbound

Proxy Transport: http

Messages:

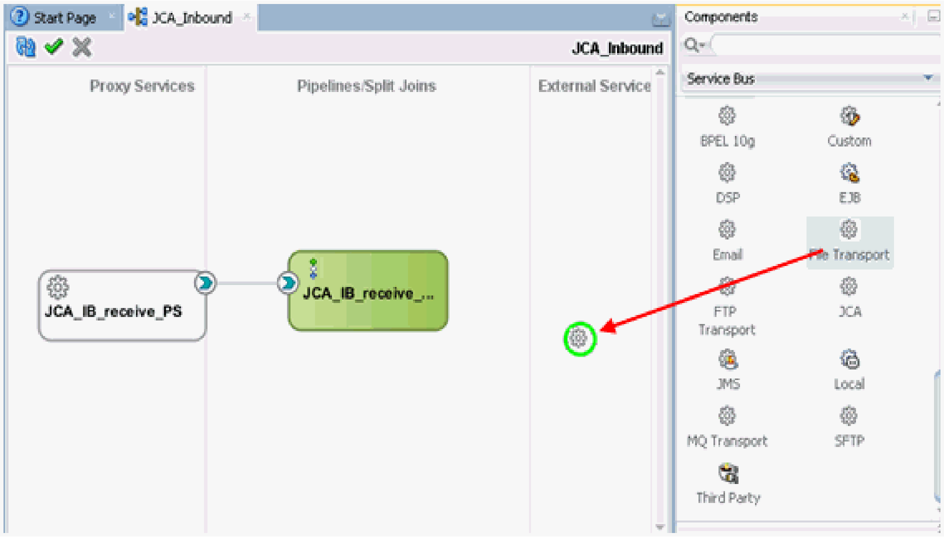
7. Drag and drop the Proxy Service to the Pipelines/Split Joins pane.

8.2.2.3 Configuring a File Transport Type Business Service

Perform the following steps to create the File Transport Type Business Service:

1. Drag and drop the **File Transport** component from the Advanced pane to the External Services pane, as shown in Figure 8–39.

Figure 8–39 File Transport Node



The Create Business Service dialog is displayed.

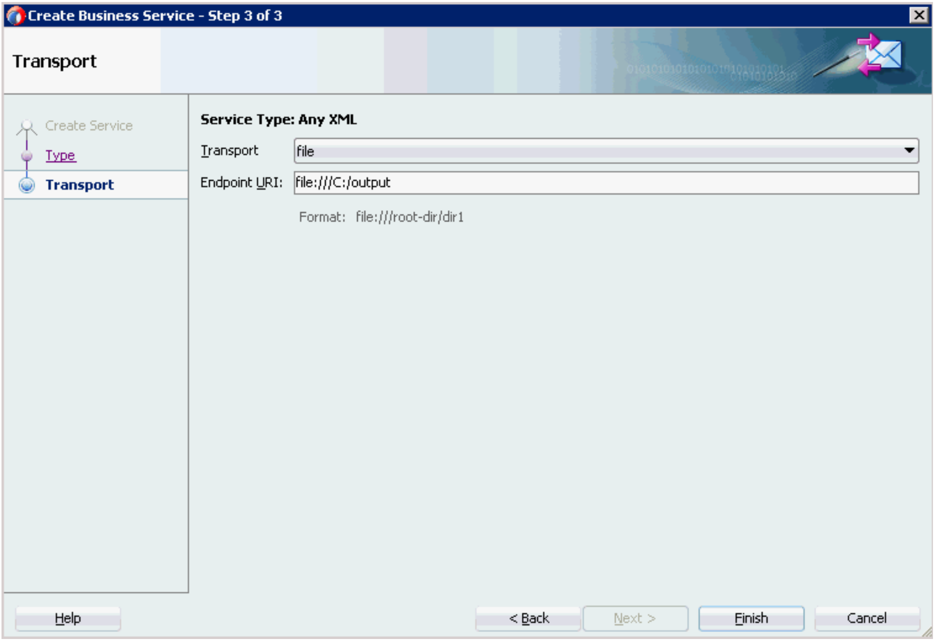
2. In the Service Name field, enter any name you wish for the Business Service (for example, FileOut), and click **Next**.

   In the displayed Type Window, the Any XML option is selected by default.

3. Click **Next**.

4. In the displayed Transport window, provide the output location in the Endpoint URI field (for example, c:\output), and click **Finish**, as shown in Figure 8–40.

Figure 8–40 Transport Pane



Create Business Service - Step 3 of 3

Transport

Service Type: Any XML

Transport: file

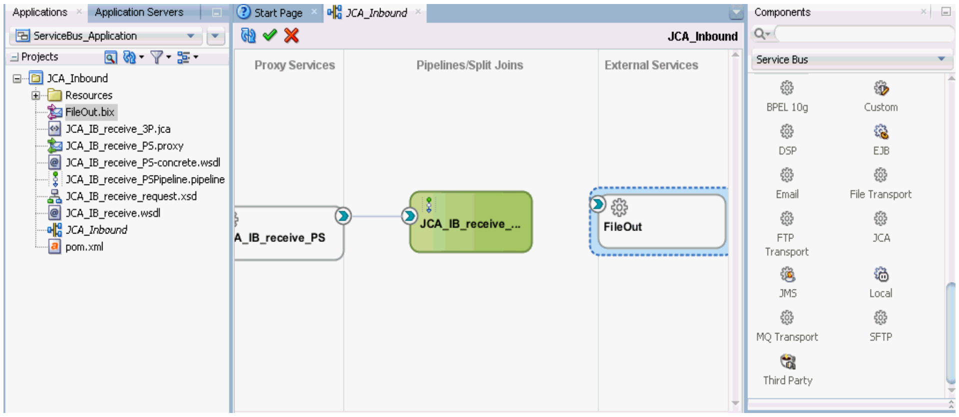
Endpoint URI: file:///C:/output

Format: file:///root-dir/dir1

The FileOut Business service is created.

5. Double-click the FileOut Business service, as shown in Figure 8–41.

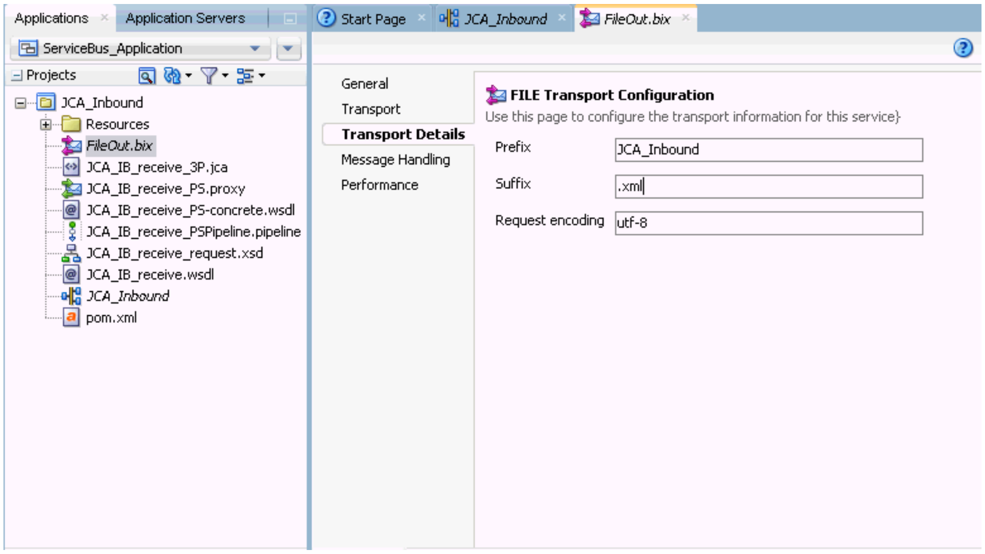
Figure 8–41 FileOut Business Service



The Configuration page is displayed.

6. Navigate to the Transport Details tab and provide the values for the Prefix and Suffix fields, as shown in Figure 8–42.

Figure 8–42 File Transport Configuration



FILE Transport Configuration

Use this page to configure the transport information for this service}

Prefix	JCA_Inbound
Suffix	.xml
Request encoding	utf-8

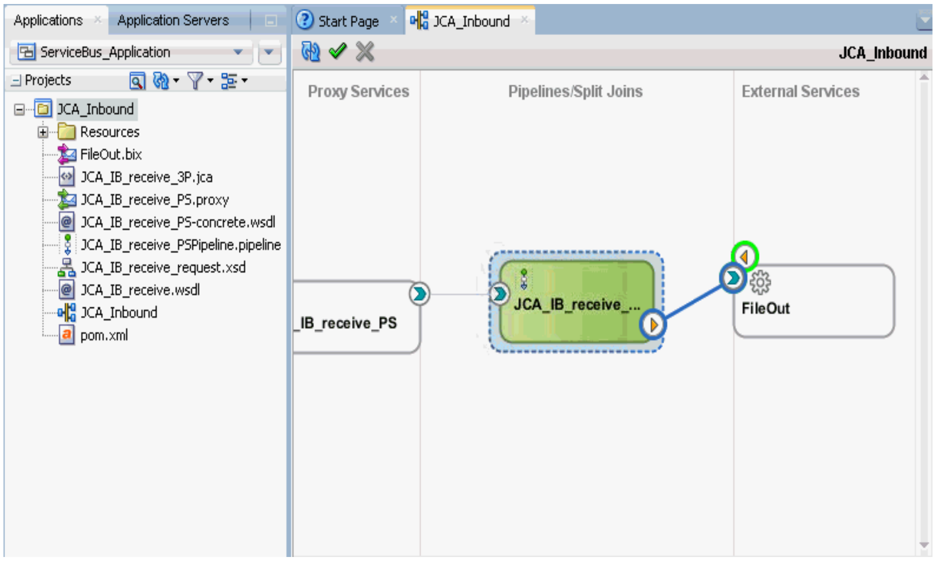
7. Save and close the Configuration page.

8.2.2.4 Configuring the Routing Rules

Perform the following steps to configure the routing rules.

1. Create a connection between the Pipeline (for example, JCA_IB_receive_ PSPipeline) and the File Type Business Service (for example, FileOut), as shown inFigure 8–43.

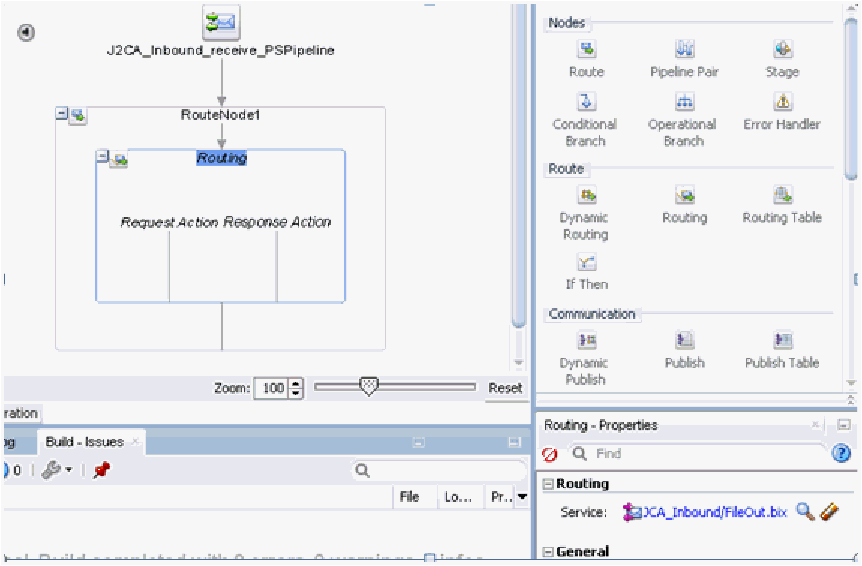
Figure 8–43 Mapping Proxy and FileOut



2. Double-click the Pipeline (for example, J2CA_Inbound_receive_PSPipeline).

3. Click the Routing pane and ensure that the File Type Business Service (for example, FileOut) is properly configured in the Service field, as shown in Figure 8–44.

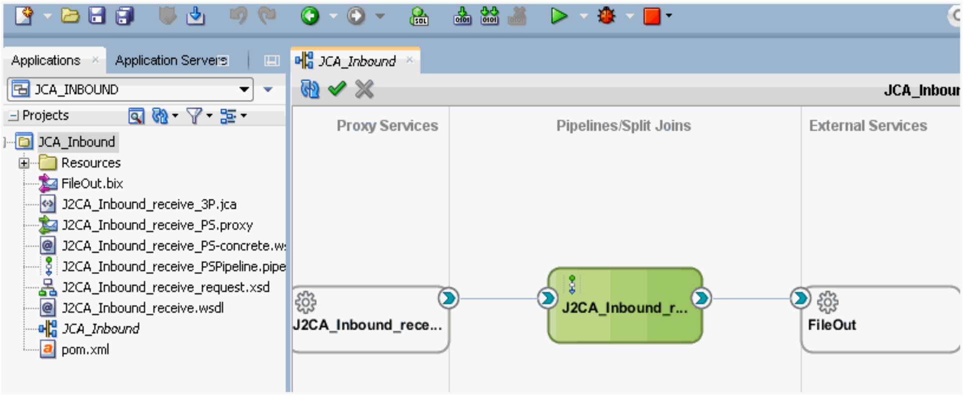
Figure 8–44 Routing Pane



4. Save and close the Pipeline configuration page.

5. Double-click on the overview.xml file (for example, JCA_Inbound) and click **Save All** in the menu bar to save the OSB process, as shown in Figure 8–45.

Figure 8–45 Save All



8.2.3 Deploying the OSB Inbound Process

To deploy the created OSB inbound process, see steps 1 - 4 in Section 8.1.3, "Deploying the OSB Outbound Process" on page 8-16.

Once the OSB inbound process is deployed successfully, trigger an event from the Siebel system and check if the output is received in the configured output location (for example, C:\output).

For more information on triggering an event, see Section 4.5.5, "Triggering an Event in Siebel" on page 4-49.

8.3 Configuring an OSB Outbound Process Using JDeveloper (BSE Configuration)

This section describes how to configure an OSB outbound process to your Siebel system, using Oracle JDeveloper for BSE configurations.

A sample project has been provided for this outbound use case scenario in the following folder of the Application Adapters installation:

```
<ADAPTER_HOME>\etc\sample\SIEBEL_Samples.zip\SIEBEL_Samples\OSB_
Jdeveloper\BSE\Siebel_Sample_BSE_OSB_Outbound_Project
```

This section includes the following topics:

- Section 8.3.1, "Creating a Service Bus Application for OSB"
- Section 8.3.2, "Defining an OSB Outbound Process"
- Section 8.3.3, "Deploying the OSB Outbound Process"

Prerequisites

Before you design an OSB outbound process, you must generate the respective WSDL file using Application Explorer. For more information, see Section 4.6.1, "Generating a WSDL File for Request and Response Services Using a Web Service" on page 4-75.

8.3.1 Creating a Service Bus Application for OSB

To configure an OSB outbound process, you must create a service bus application for OSB. For more information, see Section 8.1.1, "Creating a Service Bus Application for OSB" on page 8-2.

8.3.2 Defining an OSB Outbound Process

This section describes how to define an OSB outbound process. The following topics are included:

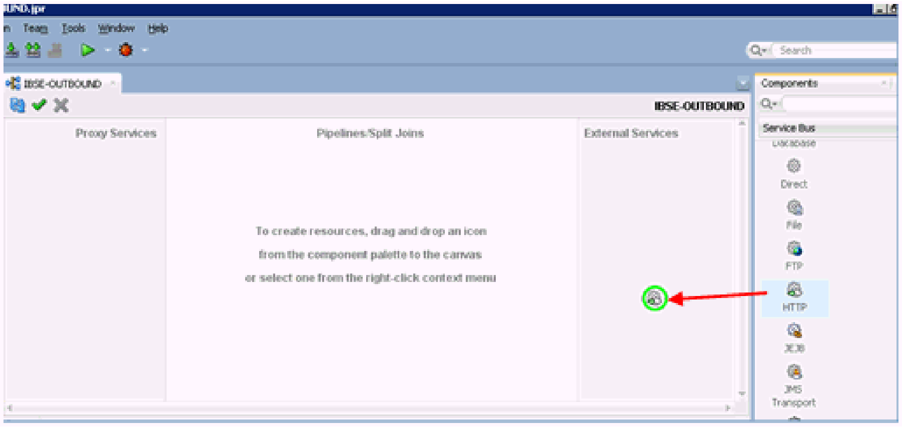
- Section 8.3.2.1, "Configuring a WSDL-based Business Service"
- Section 8.3.2.2, "Creating a Proxy Service With Pipeline"
- Section 8.3.2.3, "Configuring a File Transport Type Business Service"
- Section 8.3.2.4, "Configuring the Routing Rules"

8.3.2.1 Configuring a WSDL-based Business Service

Perform the following steps to configure a WSDL-based Business Service:

1. Drag and drop the **HTTP** component from the Technology Components pane to the External Services area, as shown in Figure 8–46.

Figure 8–46 HTTP Component

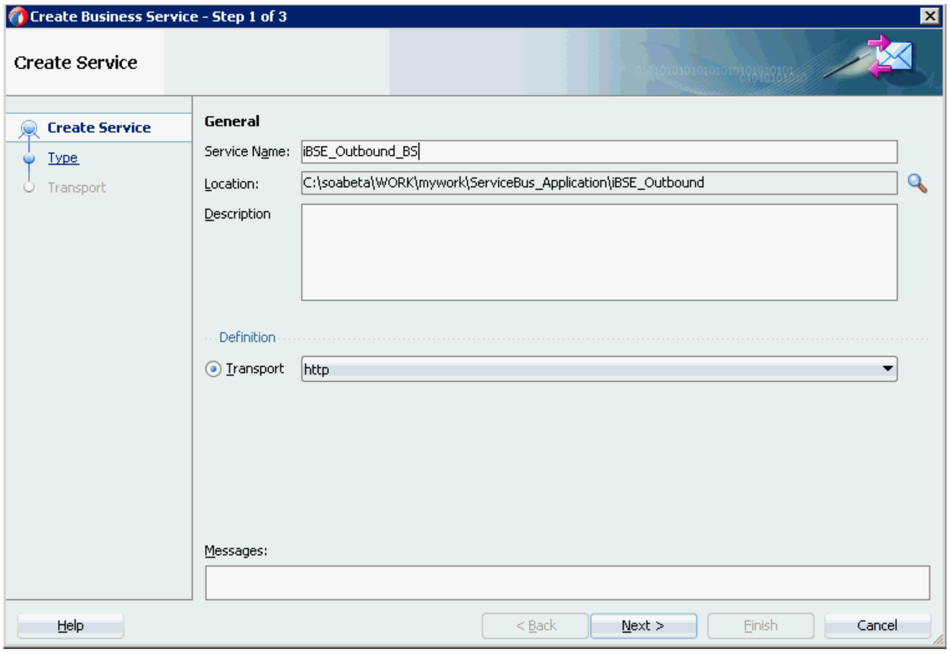


To create resources, drag and drop an icon from the component palette to the canvas or select one from the right-click context menu

The Create Business Service window is displayed.

2. In the Service Name field, enter any name you wish for the Business Service and click **Next**, as shown in Figure 8–47.

Figure 8–47 Create Business Service



Service Name: iBSE_Outbound_BS

Location: C:\soabeta\WORK\mywork\ServiceBus_Application\iBSE_Outbound

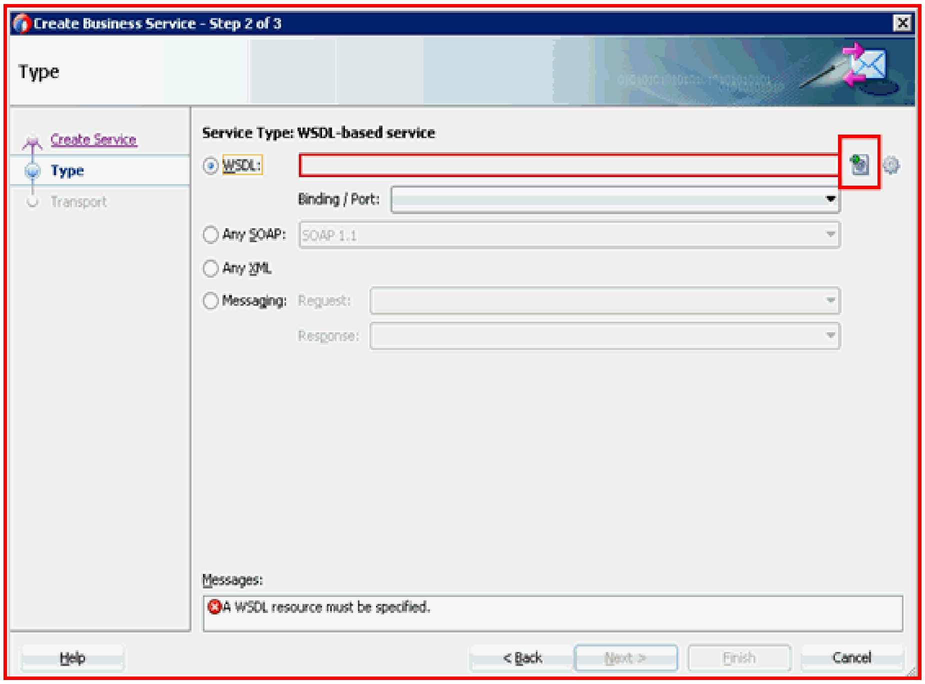
Description

○ Transport http

Messages:

3. In the displayed Service Type window, select the WSDL option and click the **Select WSDL** icon, as shown in Figure 8–48.

Figure 8–48 Type Pane



Create Business Service - Step 2 of 3

Type

Service Type: WSDL-based service

● WSDL:

Binding / Port:

○ Any SOAP: SOAP 1.1

○ Any XML

○ Messaging: Request:

Response:

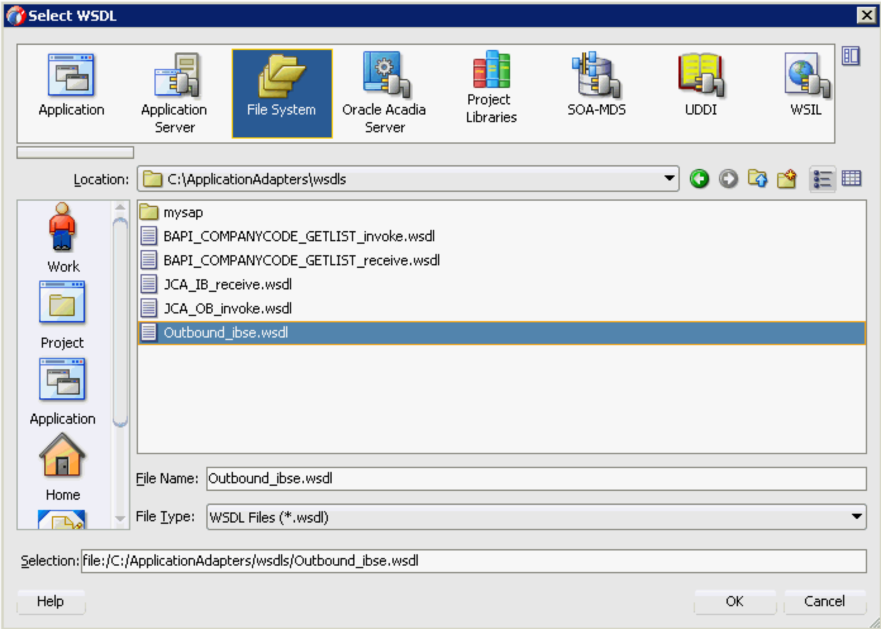
Messages:

A WSDL resource must be specified.

The Select WSDL window is displayed.

4. Select the File System folder icon, browse to the iBSE WSDL file and select it from the WSDL location, and then click **OK**, as shown in Figure 8–49.

Figure 8–49 Select WSDL Window



Select WSDL

Location: C:\ApplicationAdapters\wsdls

mysap
BAPI_COMPANYCODE_GETLIST_invoke.wsdl
BAPI_COMPANYCODE_GETLIST_receive.wsdl
JCA_IB_receive.wsdl
JCA_OB_invoke.wsdl
Outbound_ibse.wsdl

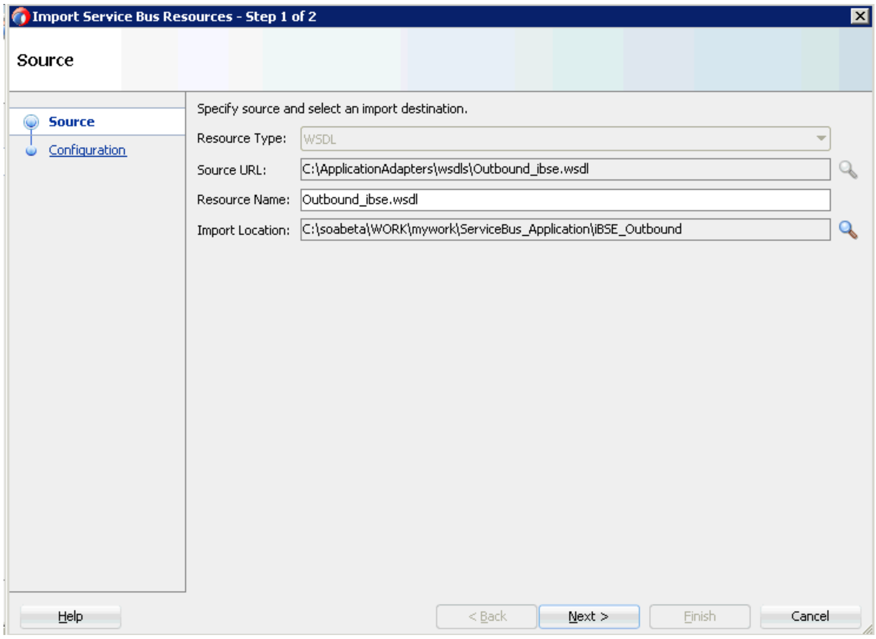
File Name: Outbound_ibse.wsdl

File Type: WSDL Files (*.wsdl)

Selection: file:/C:/ApplicationAdapters/wsdls/Outbound_ibse.wsdl

5. In the displayed Source pane, click **Next**, as shown in Figure 8–50.

Figure 8–50 Source Pane



Specify source and select an import destination.

Resource Type: WSDL

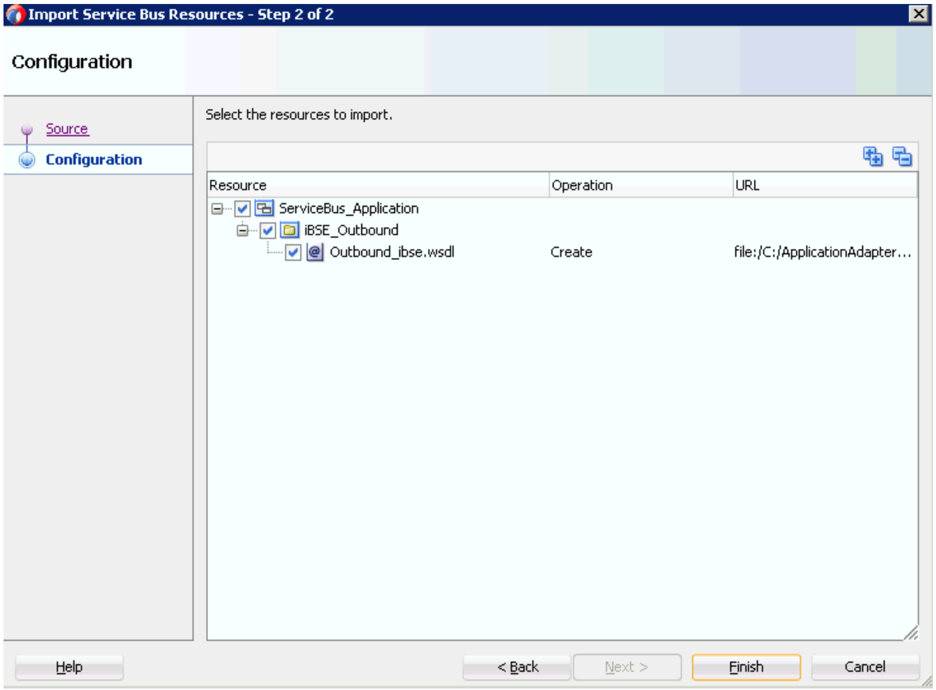
Source URL: C:\ApplicationAdapters\wsdls\Outbound_ibse.wsdl

Resource Name: Outbound_ibse.wsdl

Import Location: C:\soabeta\WORK\mywork\ServiceBus_Application\iBSE_Outbound

6. In the displayed Configuration pane, click **Finish**, as shown in Figure 8–51.

Figure 8–51 Configuration Pane



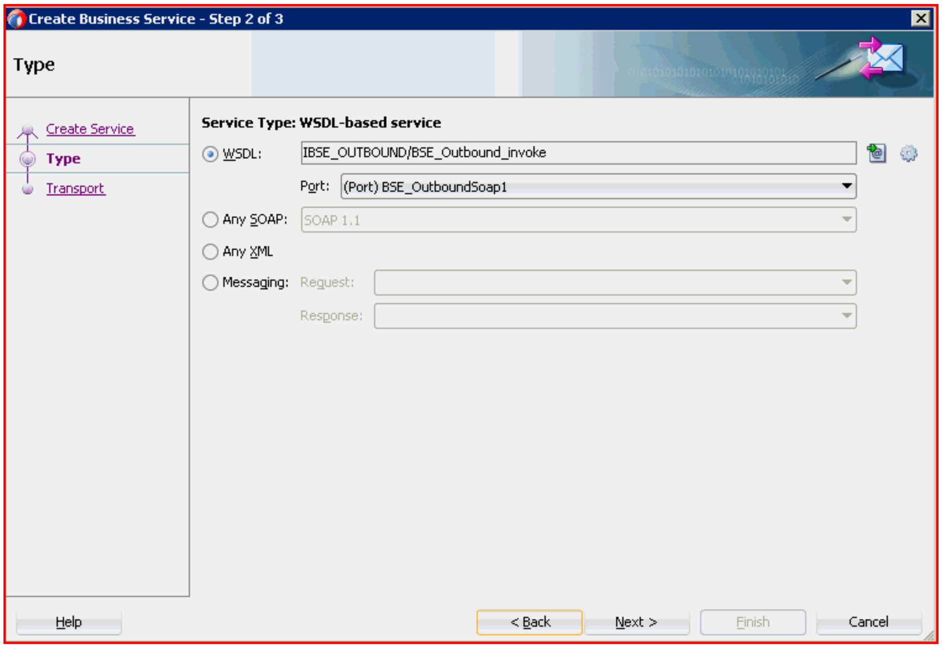
Select the resources to import.

Resource	Operation	URL
☑ ServiceBus_Application		
☑ iBSE_Outbound		
☑ Outbound_ibse.wsdl	Create	file:/C:/ApplicationAdapter...

You are returned to the Create Business Service window.

7. In the displayed Type pane, click **Next**, as shown in Figure 8–52.

Figure 8–52 Type Pane



Create Business Service - Step 2 of 3

Type

Service Type: WSDL-based service

● WSDL: IBSE_OUTBOUND/BSE_Outbound_invoke

Port: (Port) BSE_OutboundSoap1

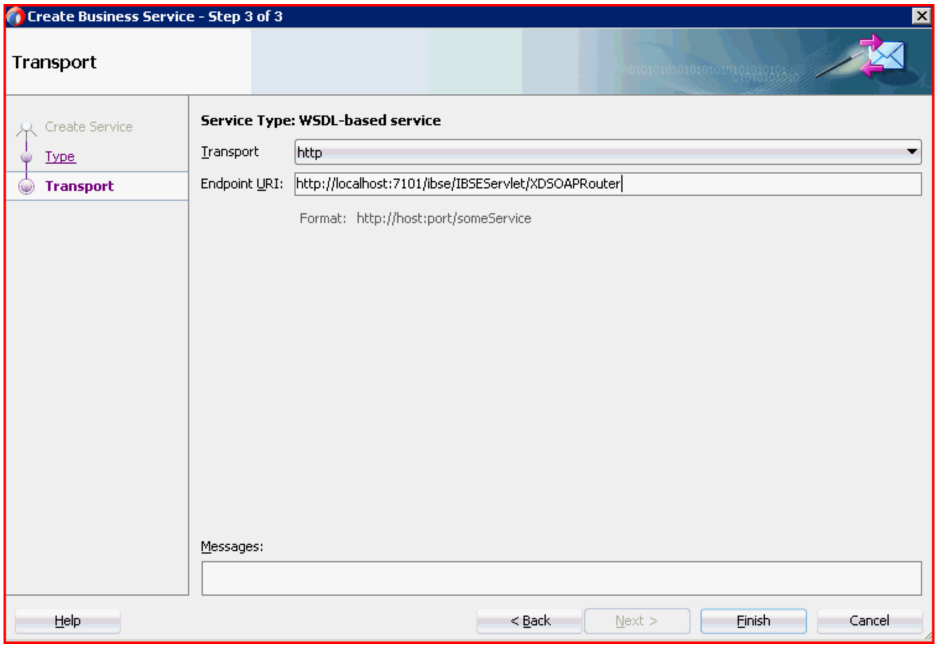
○ Any SOAP: SOAP 1.1

○ Any XML

○ Messaging: Request: Response:

8. In the displayed Transport window, you can modify the Endpoint URI field if the hostname and port number varies, and then click **Finish**, as shown in Figure 8–53.

Figure 8–53 Transport Pane



Create Business Service - Step 3 of 3

Transport

Service Type: WSDL-based service

Transport: http

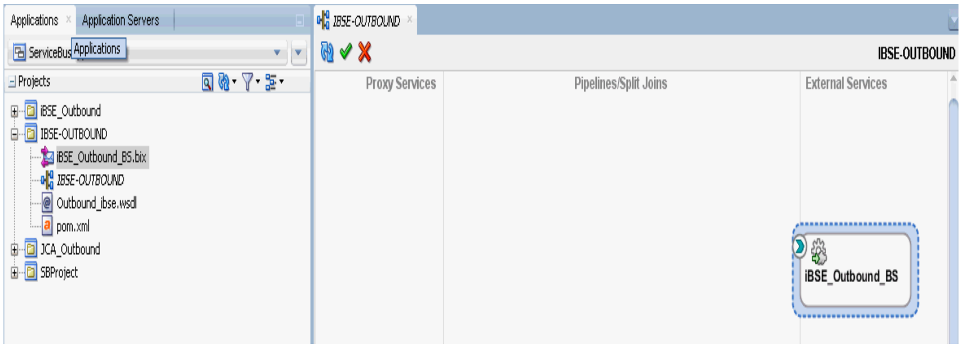
Endpoint URI: http://localhost:7101/ibse/IBSEServlet/XDSOAPRouter

Format: http://host:port/someService

Messages:

The Business Service is created and displayed in the External Services pane, as shown in Figure 8–54.

Figure 8–54 External Services Pane

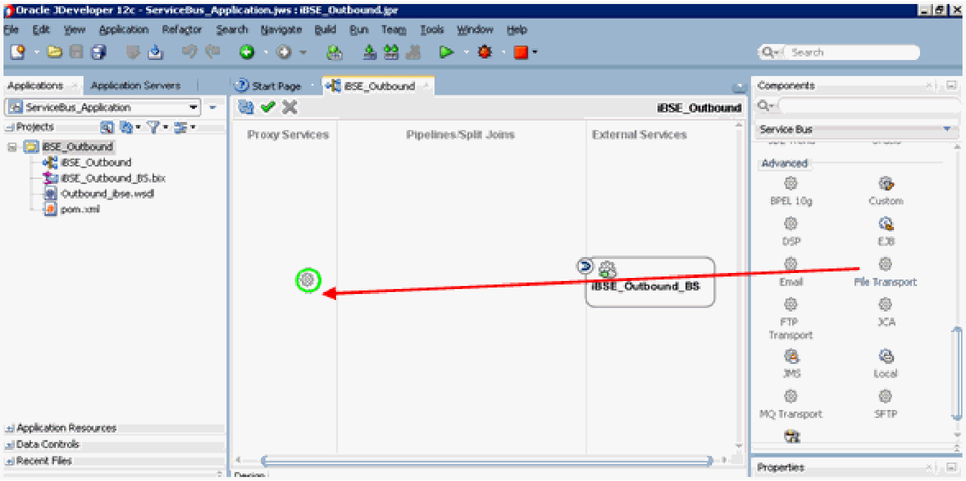


8.3.2.2 Creating a Proxy Service With Pipeline

Perform the following steps to create a Proxy Service with Pipeline:

1. Drag and drop the **File Transport** component from the Advanced Components pane to the Proxy Services pane, as shown in Figure 8–55.

Figure 8–55 File Transport Component

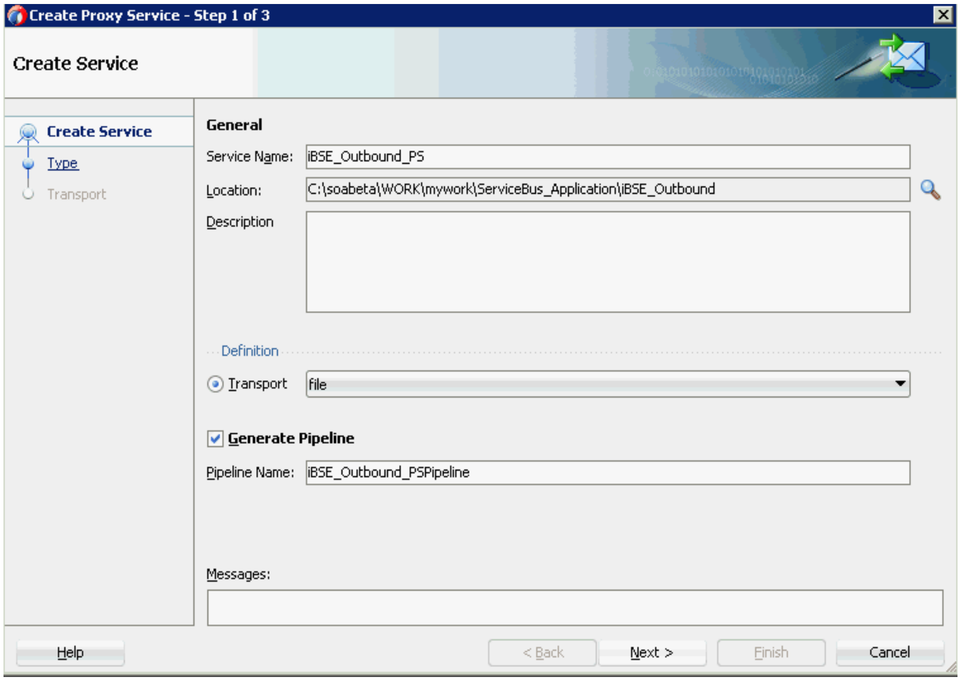


The Create Proxy Service pane is displayed.

2. In the Service Name field, enter any name you wish for the Proxy service (for example, JCA_Outbound_PS). By default, **Generate Pipeline** is selected.

3. Click **Next**, as shown in Figure 8–56.

Figure 8–56 Create Service Pane



General

Service Name: iBSE_Outbound_PS

Location: C:\soabeta\WORK\mywork\ServiceBus_Application\iBSE_Outbound

Description

Definition

● Transport: file

☑ Generate Pipeline

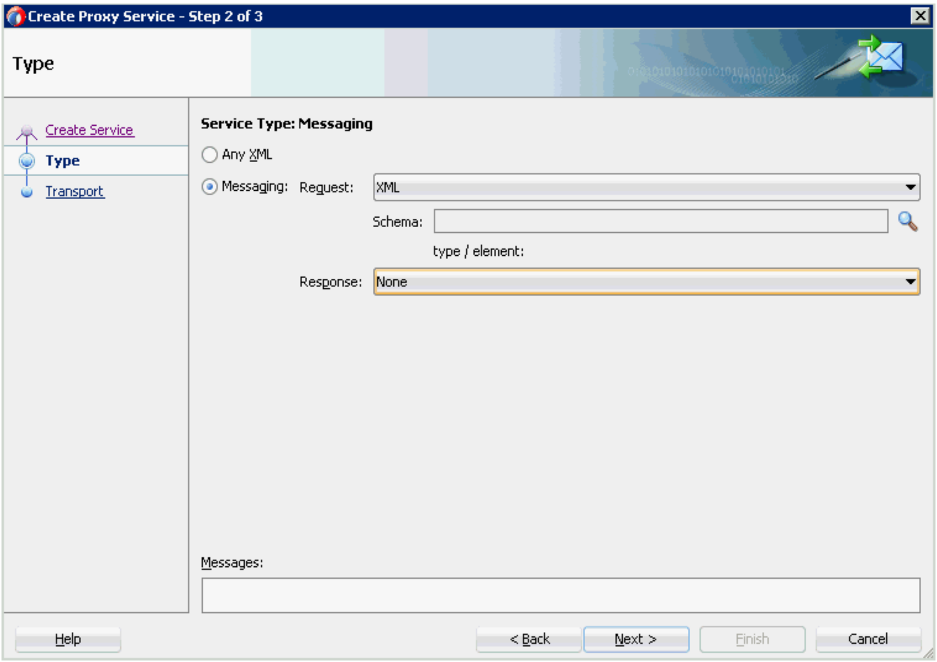
Pipeline Name: iBSE_Outbound_PSPipeline

Messages:

The Type pane is displayed.

4. Select the **Messaging** option, set the Request to **XML** and Response as **None**, and then click **Next**, as shown in Figure 8–57.

Figure 8–57 Type Pane



Service Type: Messaging

○ Any XML

● Messaging: Request: XML

Schema:

type / element:

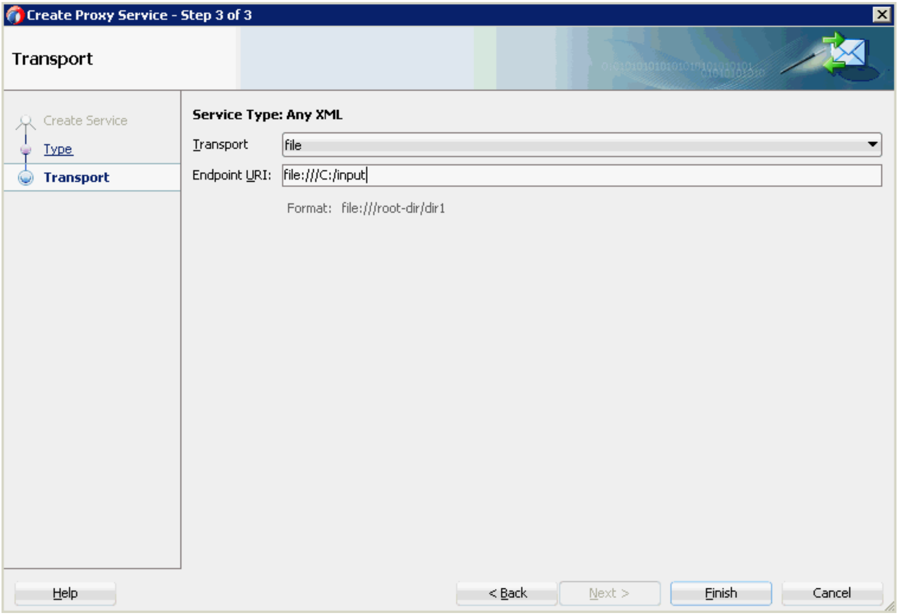
Response: None

Messages:

The Transport window is displayed.

5. Provide the input location in the Endpoint URI field (for example, c:/input) and click **Finish**, as shown in Figure 8–58.

Figure 8–58 Transport Window



Service Type: Any XML

Transport: file

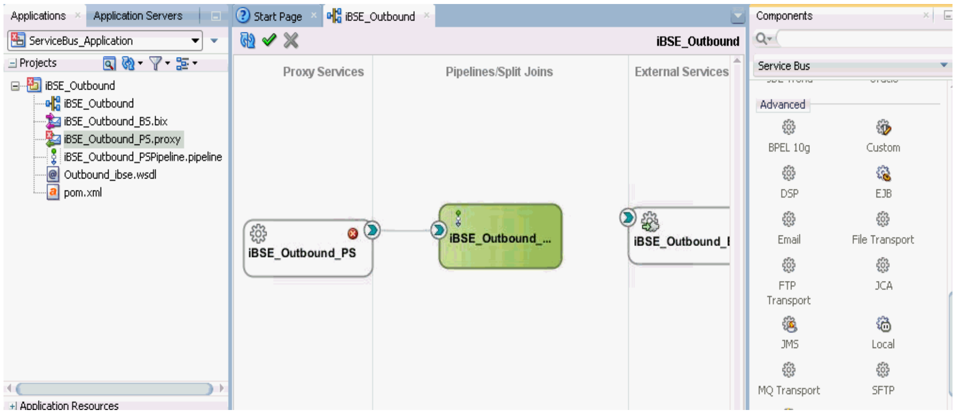
Endpoint URI: file:///C:/input

Format: file:///root-dir/dir1

The Proxy service along with the pipeline is created and displayed.

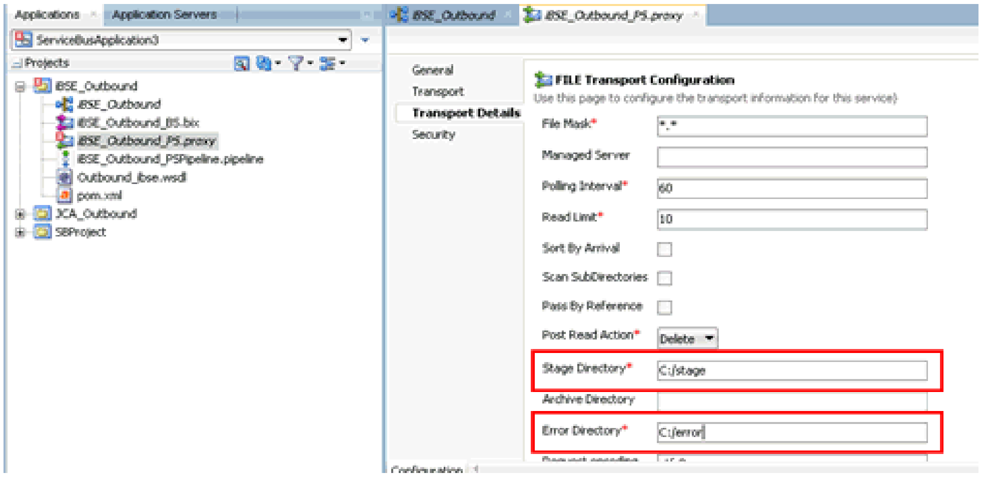
6. Double-click the created Proxy Service (for example: iBSE_Outbound_PS), as shown inFigure 8–59.

Figure 8–59 Proxy Service Edit



7. In the displayed Proxy Service configuration page, select **Transport Details** and provide the values for Stage and Error Directory, as shown in Figure 8–60.

Figure 8–60 File Transport Configuration



FILE Transport Configuration

Use this page to configure the transport information for this service)

File Mask* *.*

Managed Server

Polling Interval* 60

Read Limit* 10

Sort By Arrival □

Scan SubDirectories □

Pass By Reference □

Post Read Action* Delete

Stage Directory* C:/stage

Archive Directory

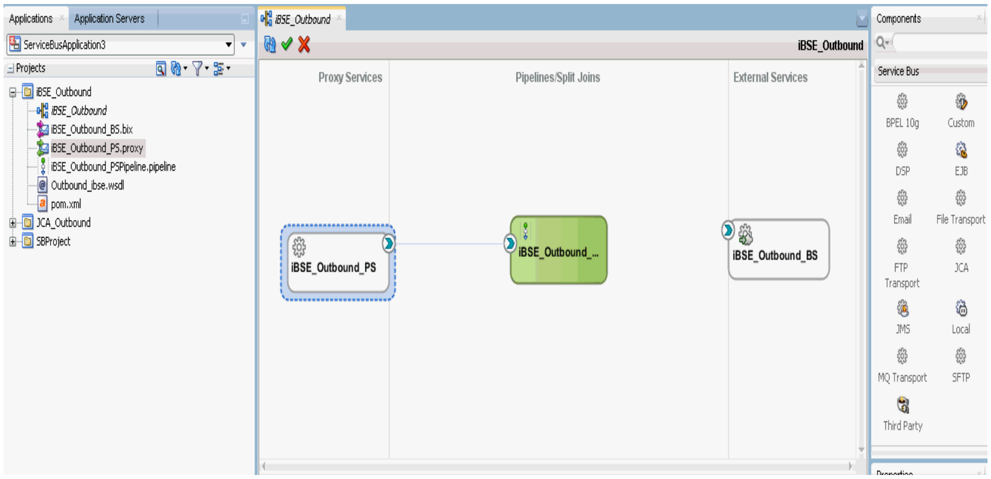
Error Directory* C:/error

8. Save and close the Proxy Service configuration page.

9. Double-click the overview.xml file (for example, iBSE_Outbound).

   The Proxy service is updated and displayed, as shown in Figure 8–61.

Figure 8–61 Proxy Service

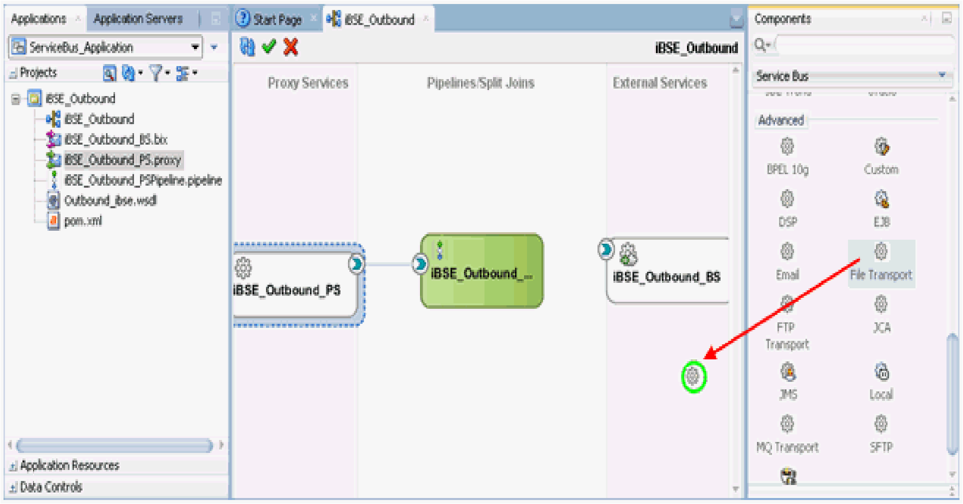


8.3.2.3 Configuring a File Transport Type Business Service

Perform the following steps to create a File Transport Type Business Service:

1. Drag and drop the **File Transport** component from the Advanced pane to the External Services pane, as shown in Figure 8–62.

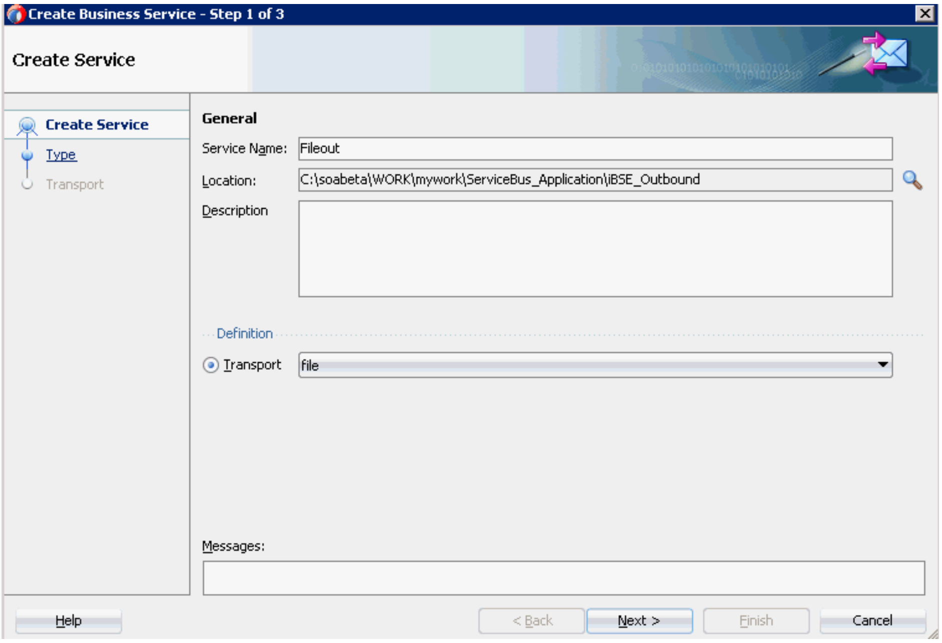
Figure 8–62 File Transport Component



The Create Business Service dialog is displayed.

2. In the Service Name field, enter any name you wish for the Business Service (for example, FileOut), and click **Next**, as shown in Figure 8–63.

Figure 8–63 Create Service Pane



Service Name: Fileout

Location: C:\soabeta\WORK\mywork\ServiceBus_Application\iBSE_Outbound

Description

● Transport file

Messages:

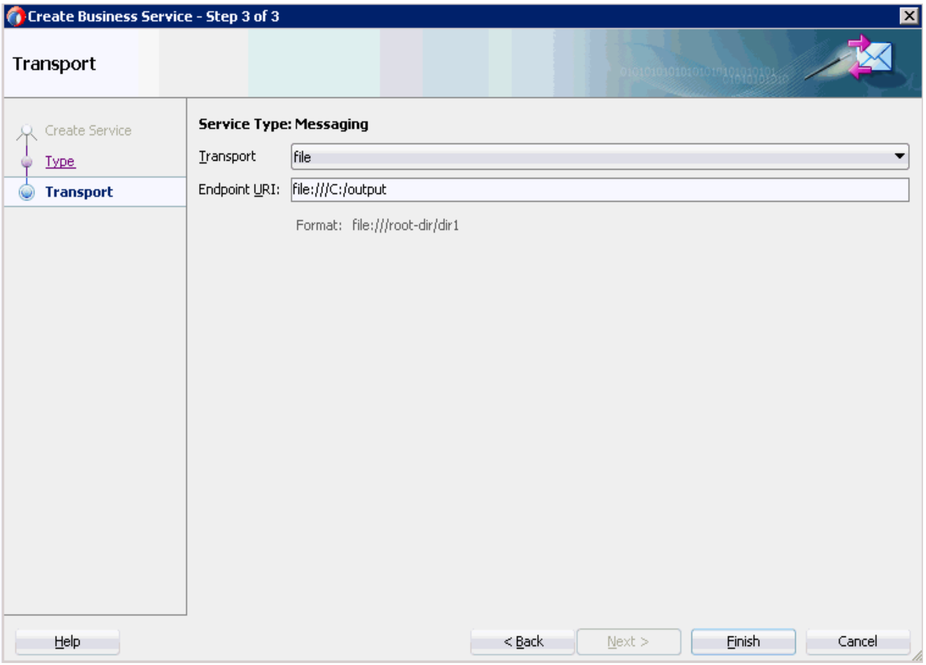
The Type pane is displayed. The **Any XML** option is selected by default.

3. Click **Next**.

The Transport pane appears.

4. Provide the output location in the Endpoint URI field (for example, c:/output) and click **Finish**, as shown in Figure 8–64.

Figure 8–64 Transport Pane



Service Type: Messaging

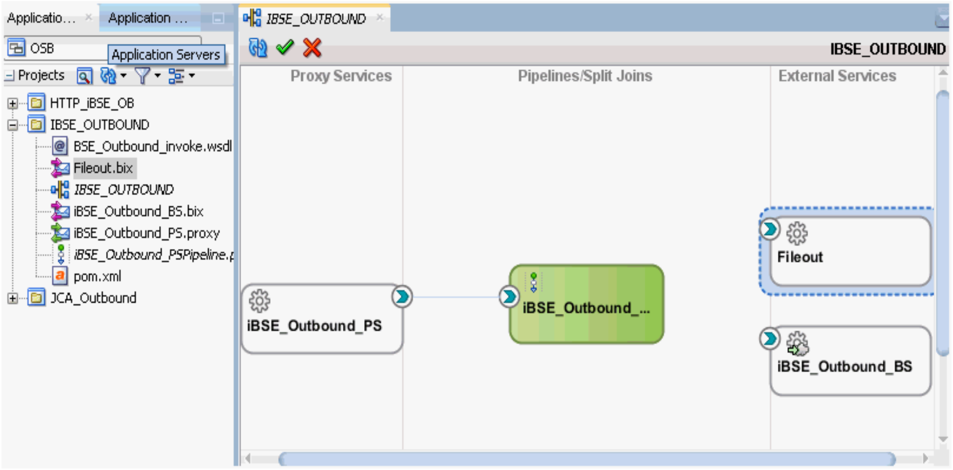
Transport: file

Endpoint URI: file:///C:/output

Format: file:///root-dir/dir1

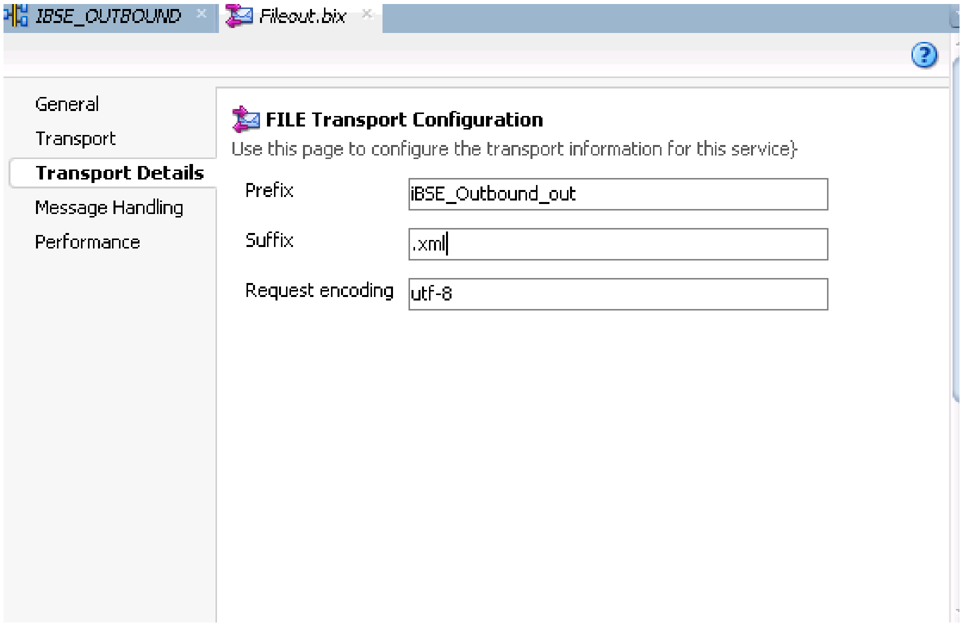
The File Transport Business service Fileout is created and displayed, as shown in Figure 8–65.

Figure 8–65 Fileout Business Service



5. Double-click the created Business service **Fileout** and provide the values for the Prefix and Suffix fields in the Transport Details Tab, as shown in Figure 8–66.

Figure 8–66 Transport Details



FILE Transport Configuration

Use this page to configure the transport information for this service}

Prefix: iBSE_Outbound_out

Suffix: .xml

Request encoding: utf-8

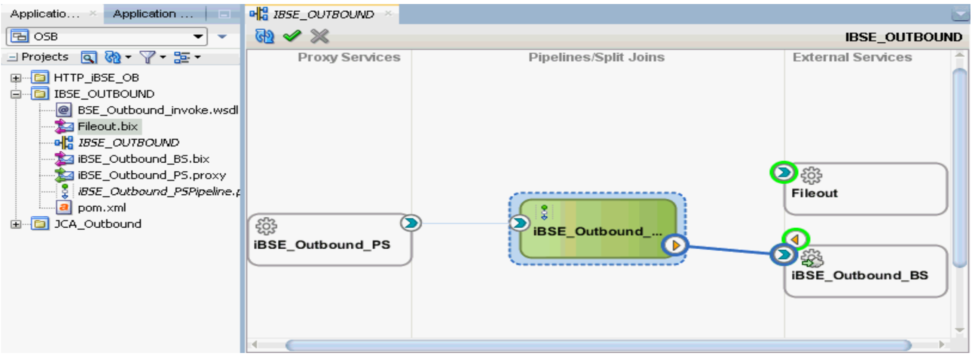
6. Save and close the configuration page, and double-click on overview.xml (for example, iBSE_Outbound).

8.3.2.4 Configuring the Routing Rules

Perform the following steps to configure the routing rules:

1. Create a connection between the Pipeline Component (for example, iBSE_Outbound_PSPipeline) and the WSDL based Business Service (for example, iBSE_Outbound_BS), as shown in Figure 8–67.

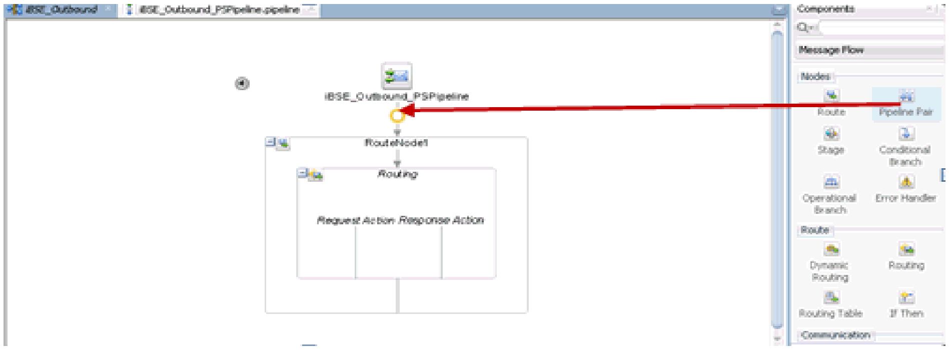
Figure 8–67 Pipeline Component



2. Double-click on the **Pipeline** component (for example, iBSE_Outbound_PSPipeline) in the Pipelines/Split Joins pane.

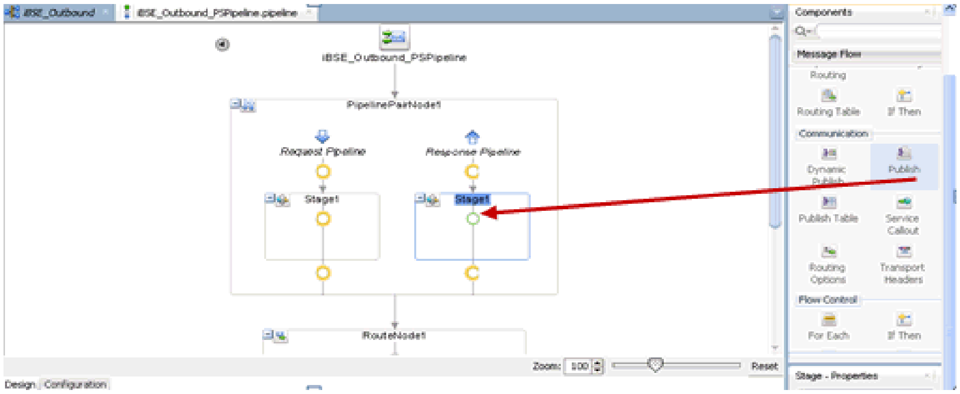
3. Drag and drop the **Pipeline Pair** node from Nodes pane to the area between the Pipeline (for example: iBSE_Outbound_PSPipeline) and RouteNode1, as shown in Figure 8–68.

Figure 8–68 Pipeline Pair Node



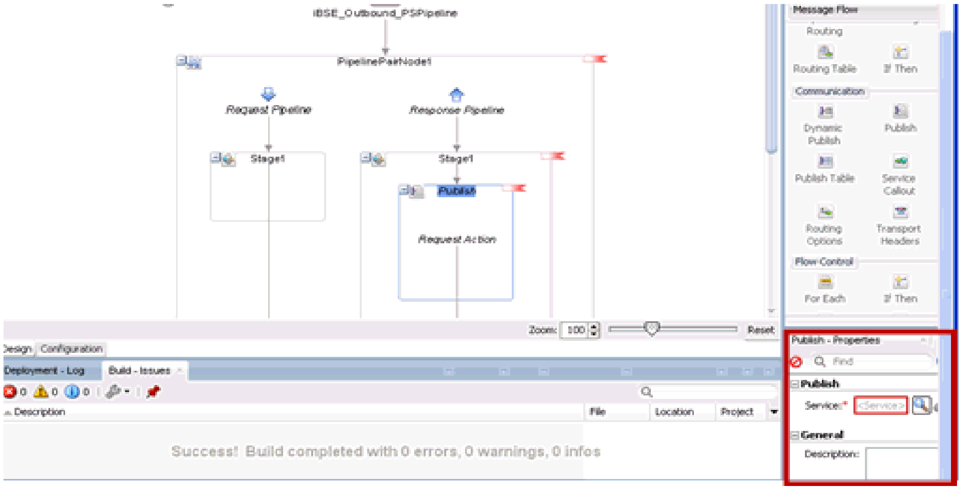
4. Drag and drop the **Publish** node from the Communication pane to the area beneath Stage1 of the Response Pipeline, as shown in Figure 8–69.

Figure 8–69 Publish Node



5. Click on the browse icon to the right of the Service field in the right pane of Publish Properties, as shown in Figure 8–70.

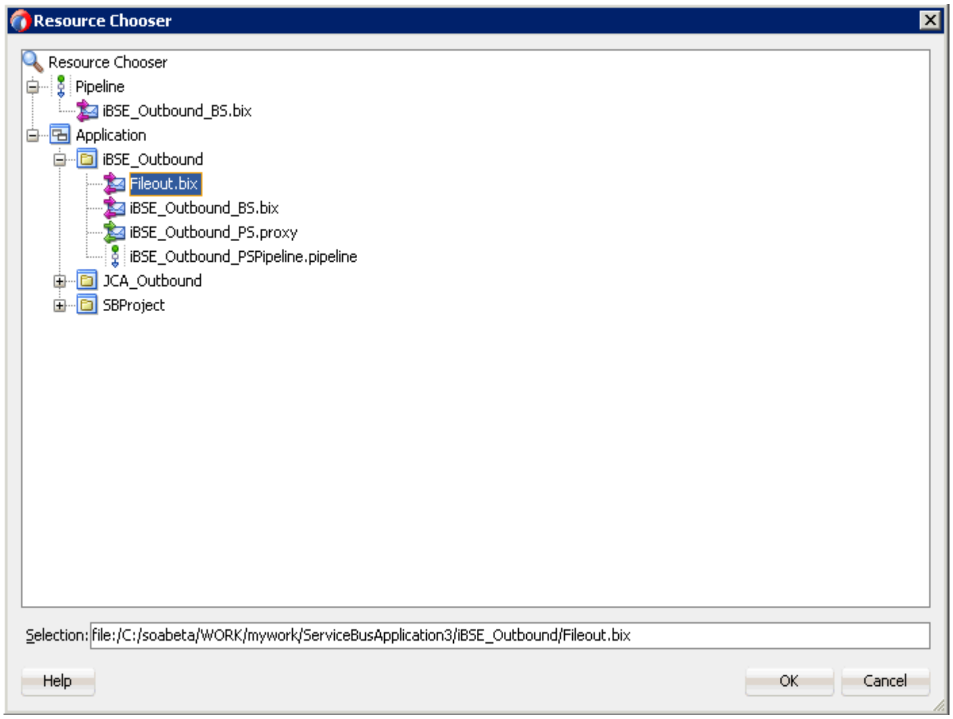
Figure 8–70 Browse Icon



Success! Build completed with 0 errors, 0 warnings, 0 infos

6. In the displayed Resource Chooser window, select the **Fileout.bix** File Transport Business service and click **OK**, as shown in Figure 8–71.

Figure 8–71 Resource Chooser



Resource Chooser

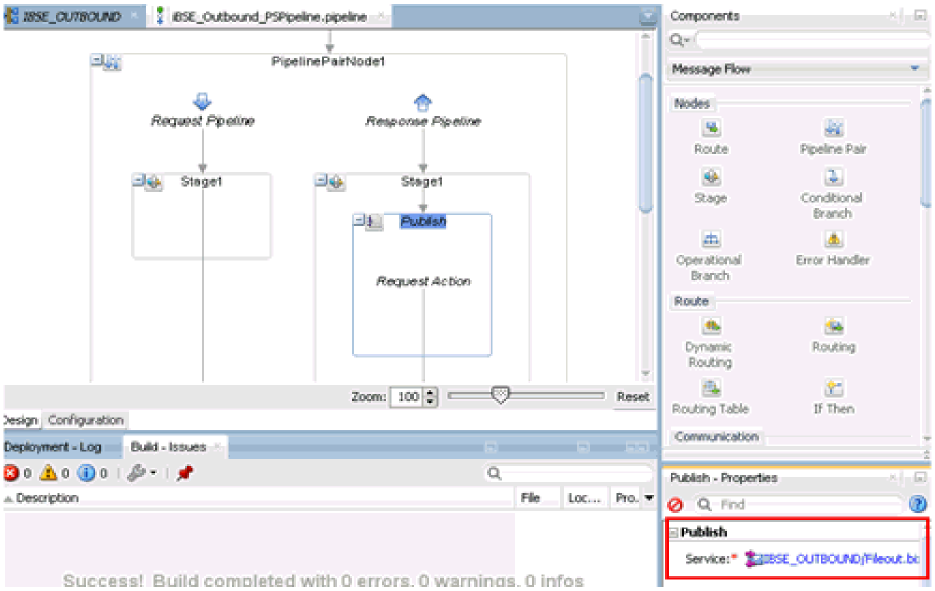
- Resource Chooser
  - Pipeline
    - iBSE_Outbound_BS.bix
  - Application
    - iBSE_Outbound
      - Fileout.bix
      - iBSE_Outbound_BS.bix
      - iBSE_Outbound_PS.proxy
      - iBSE_Outbound_PSPipeline.pipeline
    - JCA_Outbound
    - SBProject

Selection: file:/C:/soabeta/WORK/mywork/ServiceBusApplication3/iBSE_Outbound/Fileout.bix

You are returned to the Pipeline configuration page.

In the right pane, the selected service is configured in the Publish pane, as shown in Figure 8–72.

Figure 8–72 Publish Pane



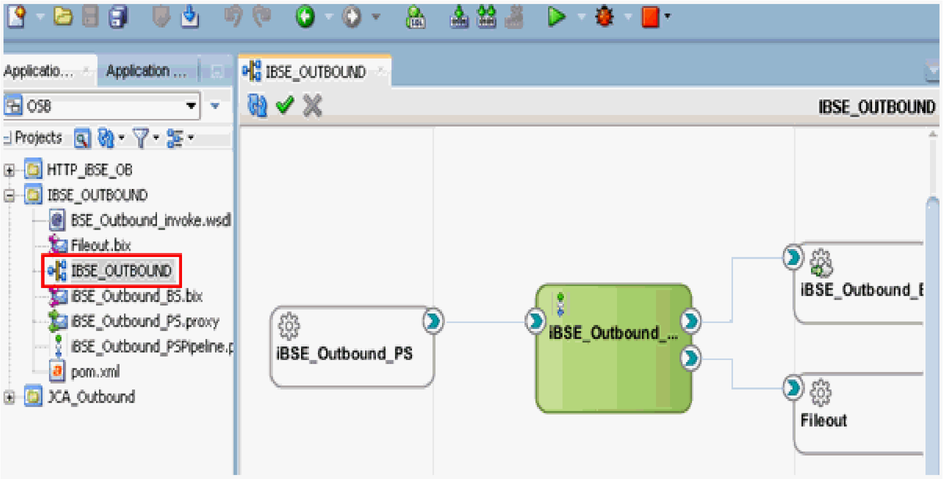
Service:* iBSE_OUTBOUND/Fileout.bi

Success! Build completed with 0 errors, 0 warnings, 0 infos

7. Save and close the Pipeline configuration page.

8. Double-click the overview.xml file (for example: iBSE_Outbound), and click **Save All** in the menu bar to save the OSB process, as shown in Figure 8–73.

Figure 8–73 Save All Icon



8.3.3 Deploying the OSB Outbound Process

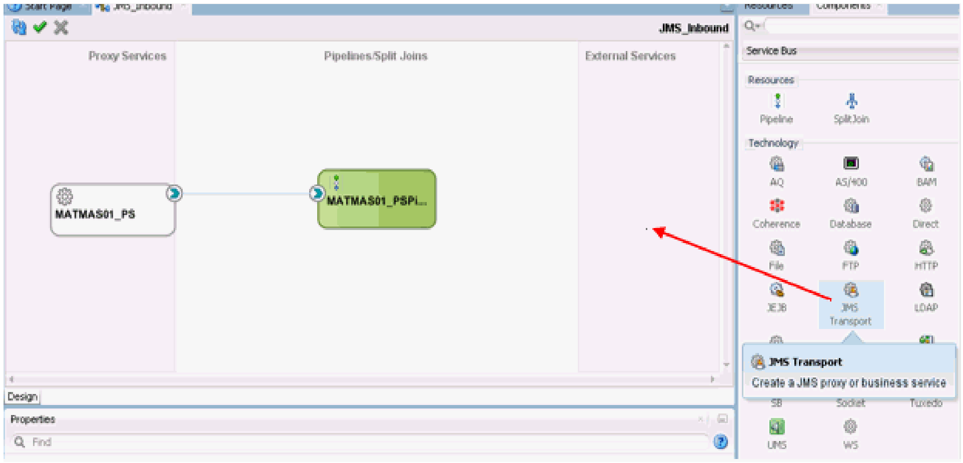
To deploy the created OSB outbound process and invoke the input XML document, see Section 8.1.3, "Deploying the OSB Outbound Process".

8.4 Configuring a JMS Inbound Process Using JDeveloper (J2CA Configuration)

This section describes how to configure a JMS inbound process to your Siebel system, using Oracle JDeveloper for J2CA configurations.

1. Before you design a JMS process, you must generate the respective WSDL file using Application Explorer. For more information, see Section 4.5.1, "Generating WSDL for Event Integration" on page 4-34.
2. Start the Oracle JDeveloper and create a Service Bus Application for OSB. For more information, see Section 8.1.1, "Creating a Service Bus Application for OSB" on page 8-2.
3. Create a Third Party Adapter Service Component. For more information, see Section 8.2.2.1, "Configuring a Third-Party Adapter Service Component" on page 8-20.
4. Create a Proxy Service along with the pipeline from the JCA Binding File. For more information, see Section 8.2.2.2, "Creating a Pipeline" on page 8-22.
5. Create a JMS Transport Business Service and perform the following steps:
   a. Drag and drop the **JMS Transport** component from the Technology Components pane to the External Services pane, as shown in Figure 8–74.

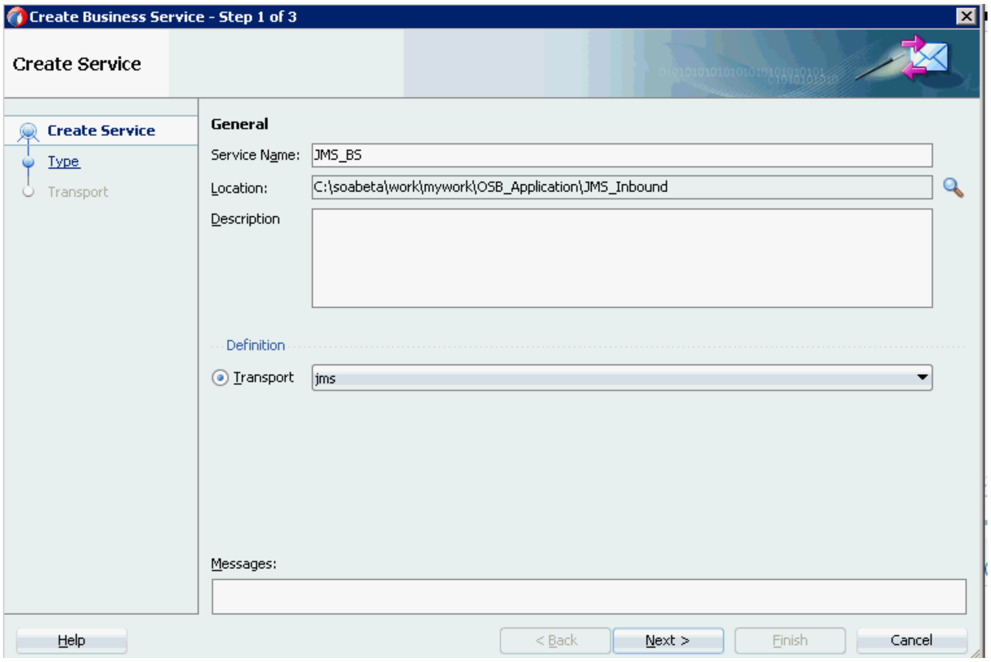
Figure 8–74 JMS Transport Component



The Create Business Service dialog is displayed.

b. In the Service Name field, enter any name you wish for the Business service (for example, JMS_BS) and click **Next**, as shown in Figure 8–75.

Figure 8–75 Create Service Pane



General

Service Name: JMS_BS

Location: C:\soabeta\work\mywork\OSB_Application\JMS_Inbound

Description

Definition

● Transport: jms

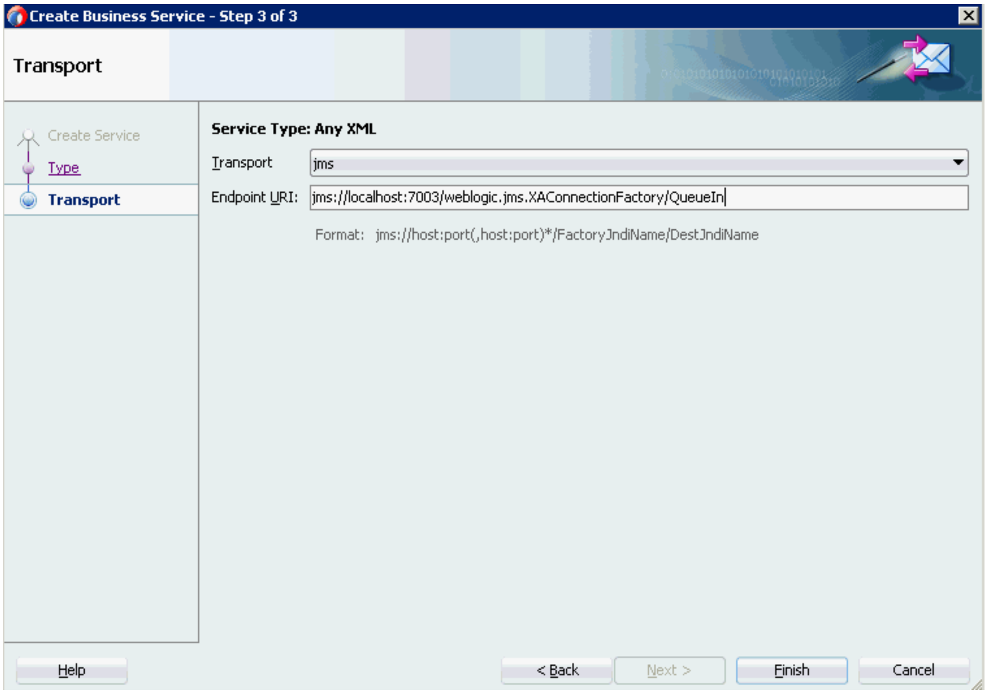
Messages:

c. In the displayed Type window, select **Any XML** and then click **Next**.

The Transport window is displayed.

d. Modify the appropriate hostname and port number by replacing `DestJndiName` with `QueueIn` in the Endpoint URI field (for example, `jms://localhost:7003/weblogic.jms.XAConnectionFactory/QueueIn`), and then click **Finish**, as shown in Figure 8–76.

Figure 8–76 Transport Window



Create Business Service - Step 3 of 3

Transport

Create Service
Type
Transport

Service Type: Any XML

Transport: jms

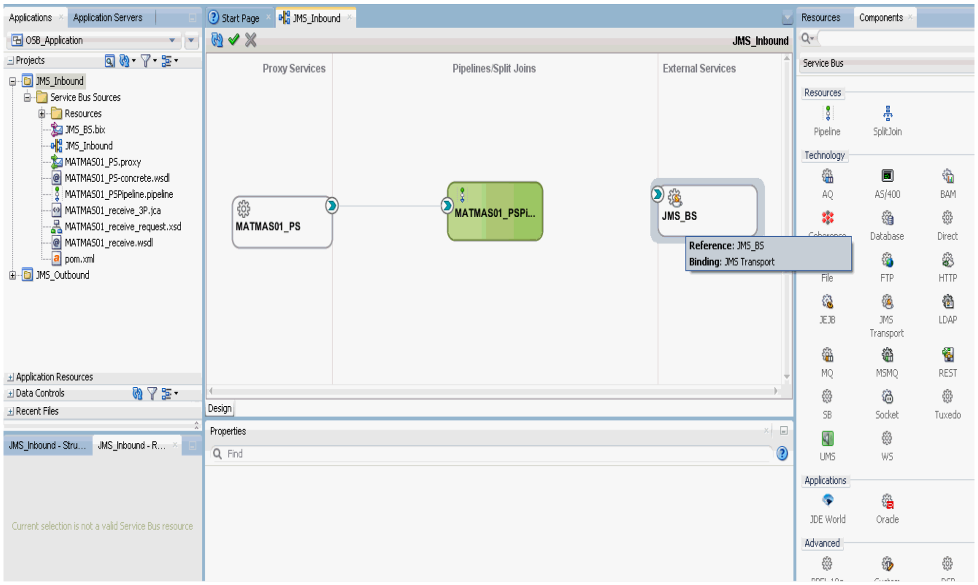
Endpoint URI: jms://localhost:7003/weblogic.jms.XAConnectionFactory/QueueIn

Format: jms://host:port(,host:port)*/FactoryJndiName/DestJndiName

The JMS Business service is created and displayed.

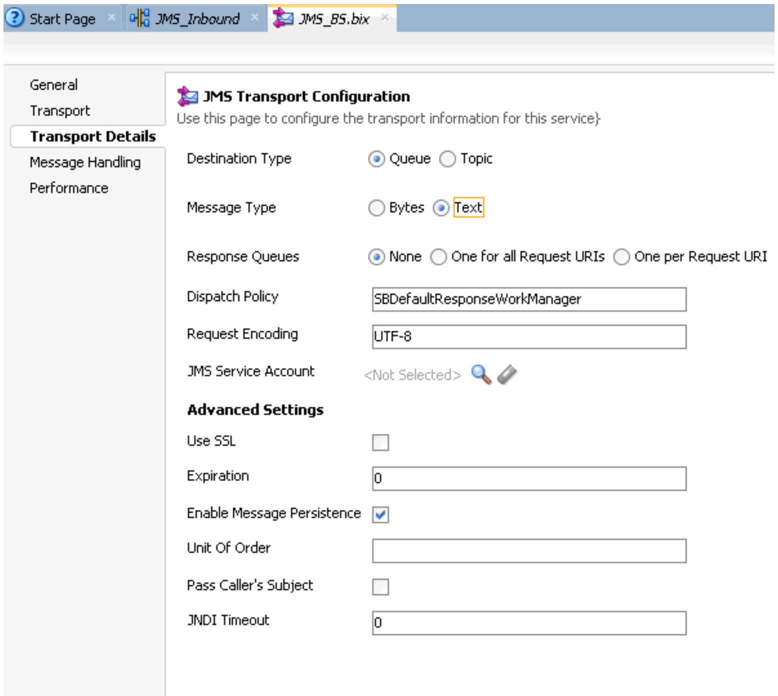
e. Double-click **JMS_BS** as shown in Figure 8–77.

Figure 8–77 JMS Business Service



f. In the displayed Business Service configuration page, provide the following parameters in the Transport Details tab, as shown in Figure 8–78.

Figure 8–78 JMS Transport Configuration



JMS Transport Configuration

Use this page to configure the transport information for this service}

Destination Type ● Queue ○ Topic

Message Type ○ Bytes ● Text

Response Queues ● None ○ One for all Request URIs ○ One per Request URI

Dispatch Policy SBDefaultResponseWorkManager

Request Encoding UTF-8

JMS Service Account <Not Selected>

Advanced Settings

Use SSL □

Expiration 0

Enable Message Persistence ☑

Unit Of Order

Pass Caller's Subject □

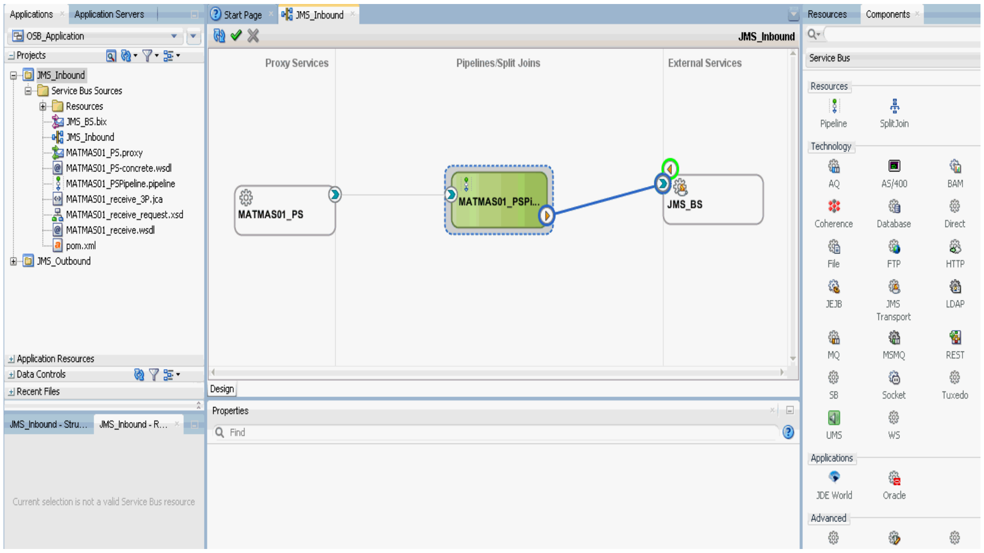
JNDI Timeout 0

- **g.** In the Destination Type section, select **Queue**.
- **h.** In the Message Type section, select **Text**.

6. Save and close the Configuration page of the business service.

7. Create a connection between **Pipeline** (for example, xxxx_PSPipeline) and **JMS Business Service** (for example, JMS_BS) as shown in figure Figure 8–79.

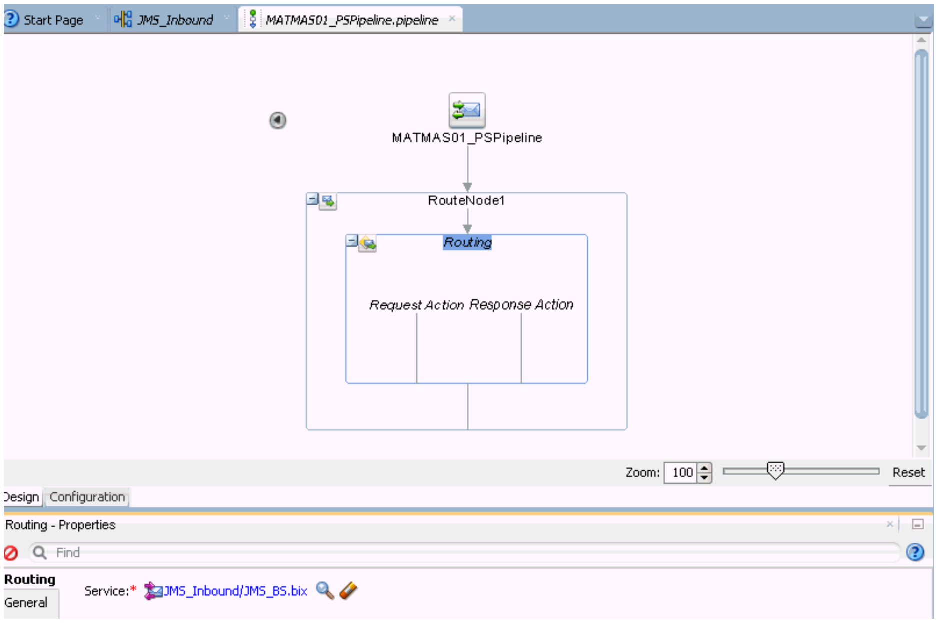
Figure 8–79 Configuration Page



8. Double-click **Pipeline**.

   The Pipeline Configuration page is displayed as shown in Figure 8–80.

Figure 8–80 Pipeline Configuration



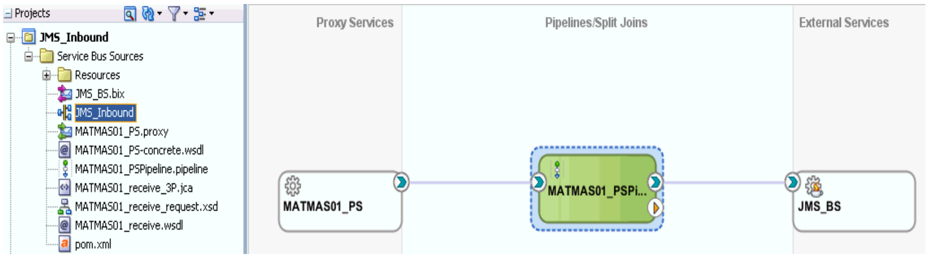
Service:* JMS_Inbound/JMS_BS.bix

9. Check that the details are configured properly, and then save and close the Pipeline configuration page.

   You are returned to the composite editor window.

10. Click **Save All** in the menu bar to save the OSB JMS process, as shown in Figure 8–81.

Figure 8–81 Save All Icon



11. Deploy the OSB JMS inbound process. For more information, see Section 8.2.3, "Deploying the OSB Inbound Process" on page 8-27.

12. Once the process is deployed successfully, trigger the event messages.

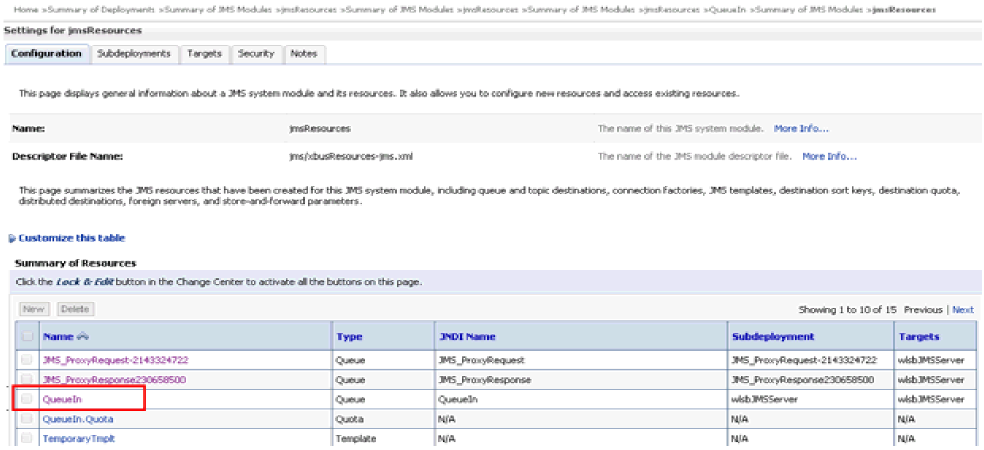
    For more information, see Section 4.5.5, "Triggering an Event in Siebel" on page 4-49.

13. Log on to the Oracle WLS console.

14. In the Oracle WLS console, expand **Services**, click **Messaging**, select **JMS Modules**, and then click **jmsResources**.

15. Click the appropriate response link (for example, QueueIn) as shown in Figure 8–82.

Figure 8–82 QueueIn Response Link



Home >Summary of Deployments >Summary of JMS Modules >jmsResources >Summary of JMS Modules >jmsResources >Summary of JMS Modules >jmsResources >QueueIn >Summary of JMS Modules >**jmsResources**

Settings for jmsResources

This page displays general information about a JMS system module and its resources. It also allows you to configure new resources and access existing resources.

Name:	jmsResources	The name of this JMS system module. More Info...
Descriptor File Name:	jms/xbusResources-jms.xml	The name of the JMS module descriptor file. More Info...

This page summarizes the JMS resources that have been created for this JMS system module, including queue and topic destinations, connection factories, JMS templates, destination sort keys, destination quota, distributed destinations, foreign servers, and store-and-forward parameters.

Summary of Resources

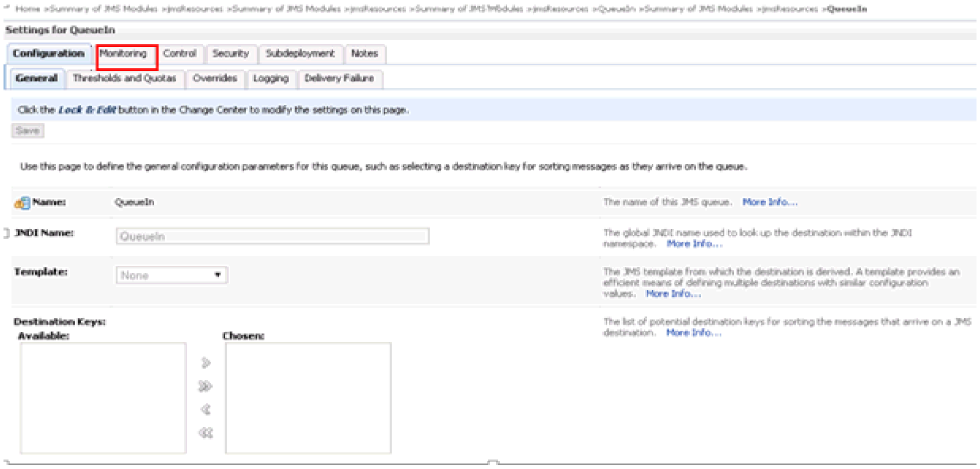
Click the *Lock & Edit* button in the Change Center to activate all the buttons on this page.

Showing 1 to 10 of 15 Previous | Next

□	Name	Type	JNDI Name	Subdeployment	Targets
□	JMS_ProxyRequest-2143324722	Queue	JMS_ProxyRequest	JMS_ProxyRequest-2143324722	wlsbJMSServer
□	JMS_ProxyResponse230658500	Queue	JMS_ProxyResponse	JMS_ProxyResponse230658500	wlsbJMSServer
□	QueueIn	Queue	QueueIn	wlsbJMSServer	wlsbJMSServer
□	QueueIn.Quota	Quota	N/A	N/A	N/A
□	TemporaryTmplt	Template	N/A	N/A	N/A

16. Click the Monitoring tab, as shown in Figure 8–83.

Figure 8–83 Monitoring Tab



Home >Summary of JMS Modules >jmsResources >Summary of JMS Modules >jmsResources >Summary of JMS Modules >jmsResources >QueueIn >Summary of JMS Modules >jmsResources >**QueueIn**

Settings for QueueIn

Click the *Lock & Edit* button in the Change Center to modify the settings on this page.

Use this page to define the general configuration parameters for this queue, such as selecting a destination key for sorting messages as they arrive on the queue.

Name:	QueueIn	The name of this JMS queue. More Info...
JNDI Name:	QueueIn	The global JNDI name used to look up the destination within the JNDI namespace. More Info...
Template:	None	The JMS template from which the destination is derived. A template provides an efficient means of defining multiple destinations with similar configuration values. More Info...

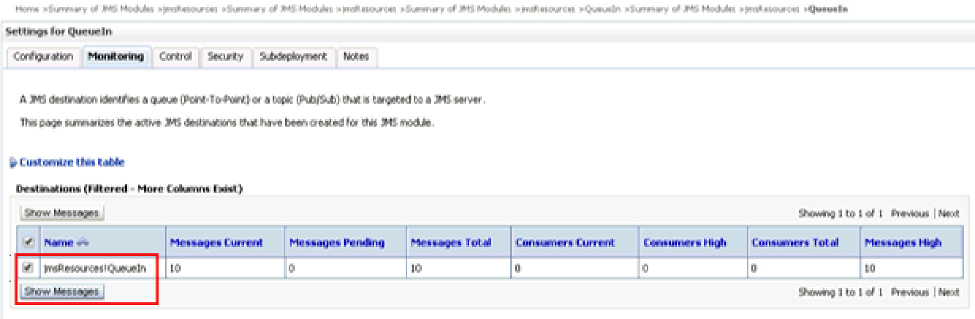
Destination Keys:

Available: **Chosen:**

The list of potential destination keys for sorting the messages that arrive on a JMS destination. More Info...

17. Select the check box and click the **Show Messages** button, as shown in Figure 8–84.

Figure 8–84 Show Messages Button



Home >Summary of JMS Modules >jmsResources >Summary of JMS Modules >jmsResources >Summary of JMS Modules >jmsResources >QueueIn >Summary of JMS Modules >jmsResources >**QueueIn**

Settings for QueueIn

A JMS destination identifies a queue (Point-To-Point) or a topic (Pub/Sub) that is targeted to a JMS server.

This page summarizes the active JMS destinations that have been created for this JMS module.

Destinations (Filtered - More Columns Exist)

Showing 1 to 1 of 1 Previous | Next

☑	Name	Messages Current	Messages Pending	Messages Total	Consumers Current	Consumers High	Consumers Total	Messages High
☑	jmsResources!QueueIn	10	0	10	0	0	0	10

Showing 1 to 1 of 1 Previous | Next

18. Click the ID link with the appropriate time and date.

    The response document is shown under the Text field.

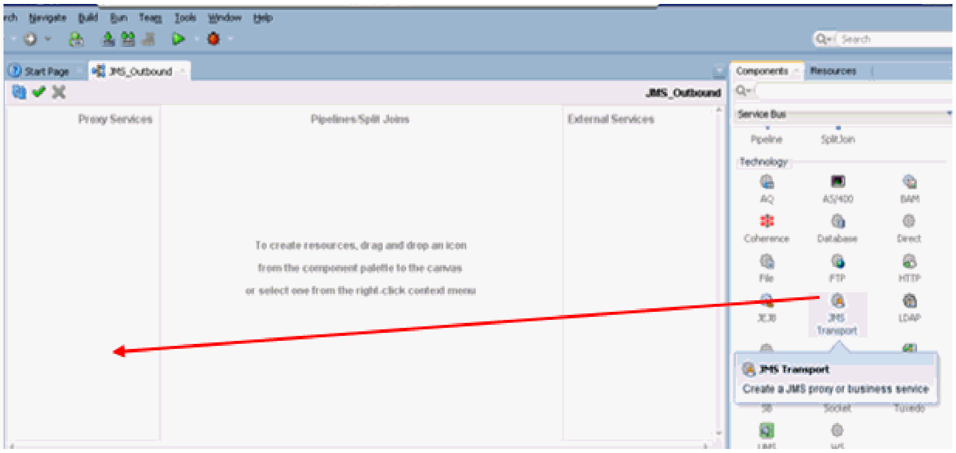
8.5 Configuring a JMS Outbound Process Using JDeveloper (J2CA Configuration)

This section describes how to configure a JMS outbound process to your Siebel system, using Oracle JDeveloper for J2CA configurations.

1. Before you design a JMS process, you must generate the respective WSDL file using Application Explorer. For more information, see Section 4.4.1, "Generating WSDL for Request/Response Service" on page 4-8.
2. Start the Oracle JDeveloper and create a Service Bus Application for OSB. For more information, see Section 8.1.1, "Creating a Service Bus Application for OSB" on page 8-2.
3. Create a Third Party Adapter Service Component. For more information, see Section 8.1.2.1, "Configuring a Third-Party Adapter Service Component" on page 8-3.
4. Create a WSDL-based Business Service from the JCA Binding File. For more information, see Section 8.1.2.2, "Configuring a File Transport Type Business Service" on page 8-7.
5. Create a JMS Proxy Service with a Pipeline and perform the following steps:

   a. Drag and drop the **JMS Transport** component from the Technology Components pane to the Proxy Services pane, as shown in Figure 8–85.

Figure 8–85 JMS Transport Component



To create resources, drag and drop an icon from the component palette to the canvas or select one from the right-click context menu

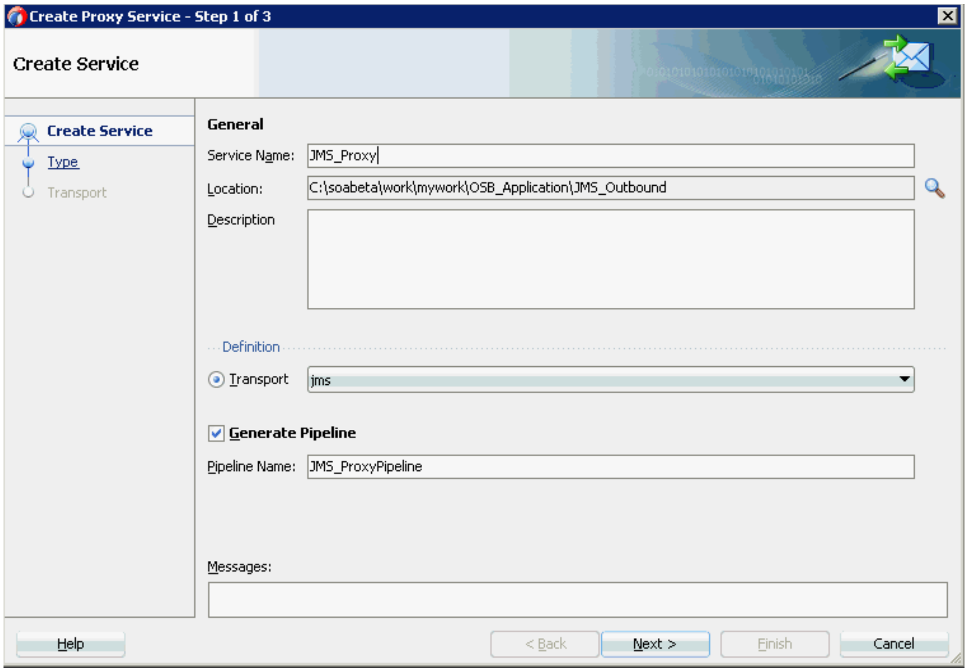
Create a JMS proxy or business service

   The Create Business Service dialog is displayed.

   b. In the Service Name field, enter any name you wish for the Proxy service (for example, JMS_Proxy). By default, Generate Pipeline is selected.

   c. Click **Next**, as shown in Figure 8–86.

Figure 8–86 Create Proxy Service Pane



Create Proxy Service - Step 1 of 3

Create Service

General

Service Name: JMS_Proxy

Location: C:\soabeta\work\mywork\OSB_Application\JMS_Outbound

Description

Definition

● Transport: jms

☑ Generate Pipeline

Pipeline Name: JMS_ProxyPipeline

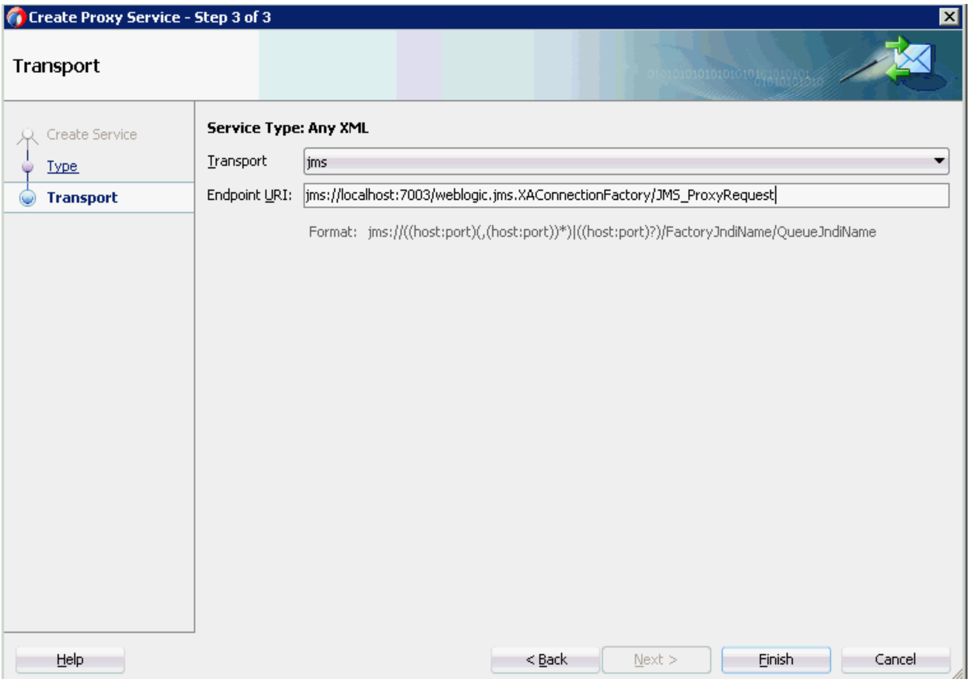
Messages:

d. In the displayed Type window, select **Any XML** and then click **Next**.

The Transport window is displayed.

e. Modify the appropriate hostname and port number by replacing the Endpoint URI field (for example, `jms://localhost:7003/weblogic.jms.XAConnectionFactory/JMS_ProxyRequest`), and then click **Finish**, as shown in Figure 8–87.

Figure 8–87 Transport Window



Create Proxy Service - Step 3 of 3

Transport

Service Type: Any XML

Transport: jms

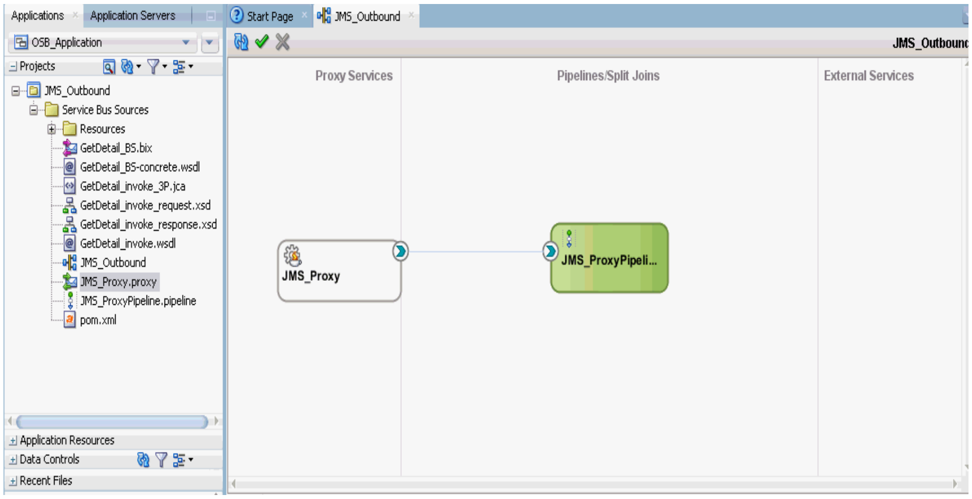
Endpoint URI: jms://localhost:7003/weblogic.jms.XAConnectionFactory/JMS_ProxyRequest

Format: jms://((host:port)(,(host:port))*)|((host:port)?)/FactoryJndiName/QueueJndiName

The JMS Proxy service along with the pipeline is created and displayed.

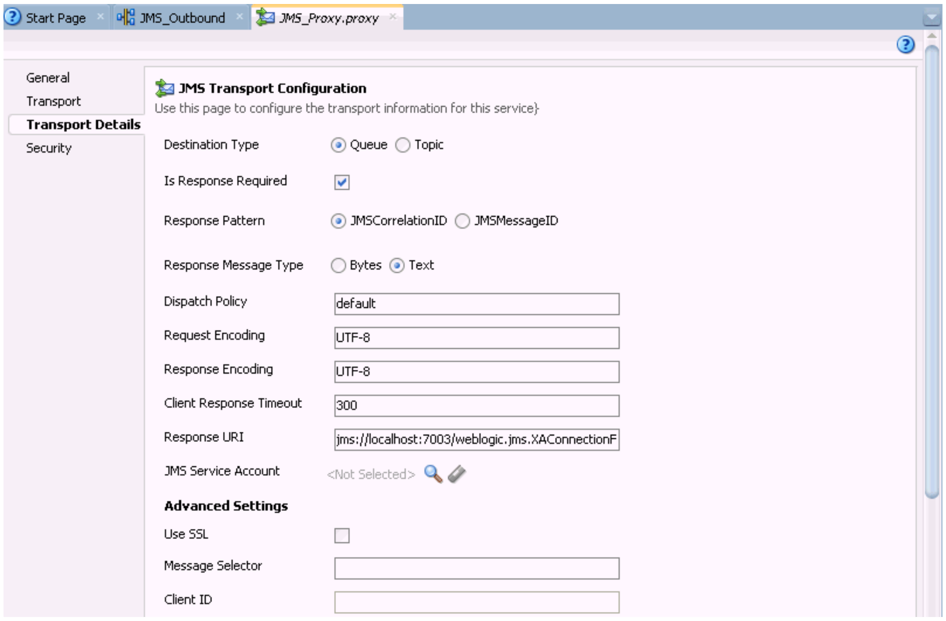
f. Double-click the created Proxy Service (for example, JMS_Proxy), as shown in Figure 8–88.

Figure 8–88 JMS Proxy Service



g. In the displayed configuration page of the Proxy Service, provide the following parameters in the Transport Details tab, as shown in Figure 8–89.

Figure 8–89 JMS Transport Configuration



JMS Transport Configuration

Use this page to configure the transport information for this service}

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	jms://localhost:7003/weblogic.jms.XAConnectionF
JMS Service Account	<Not Selected>

Advanced Settings

Use SSL	□
Message Selector	
Client ID	

h. In the Destination Type section, select **Queue**.

i. Select the **Is Response Required** check box.

j. In the Response Message Type section, select **Text**.

k. In the Response URI field, provide the Endpoint URI used in the JMS Transport Configuration and change `Request` to `Response`. For example,

```
jms://localhost:7003/weblogic.jms.XAConnectionFactory/JMS_
ProxyResponse
```

6. Save and close the Configuration page of the Proxy service.

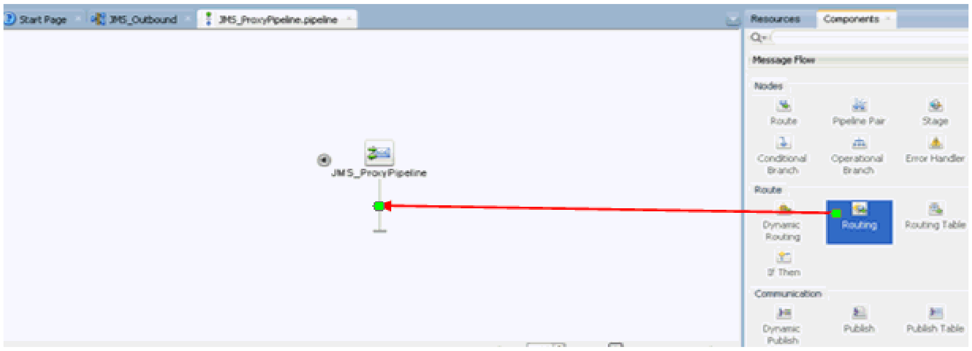
7. Configure the Routing Rules and proceed with the following steps:

   a. Double-click on the pipeline (for example, JMS_ProxyPipeline) in the Pipelines/Split Joins pane.

      The Pipeline configuration page is displayed.

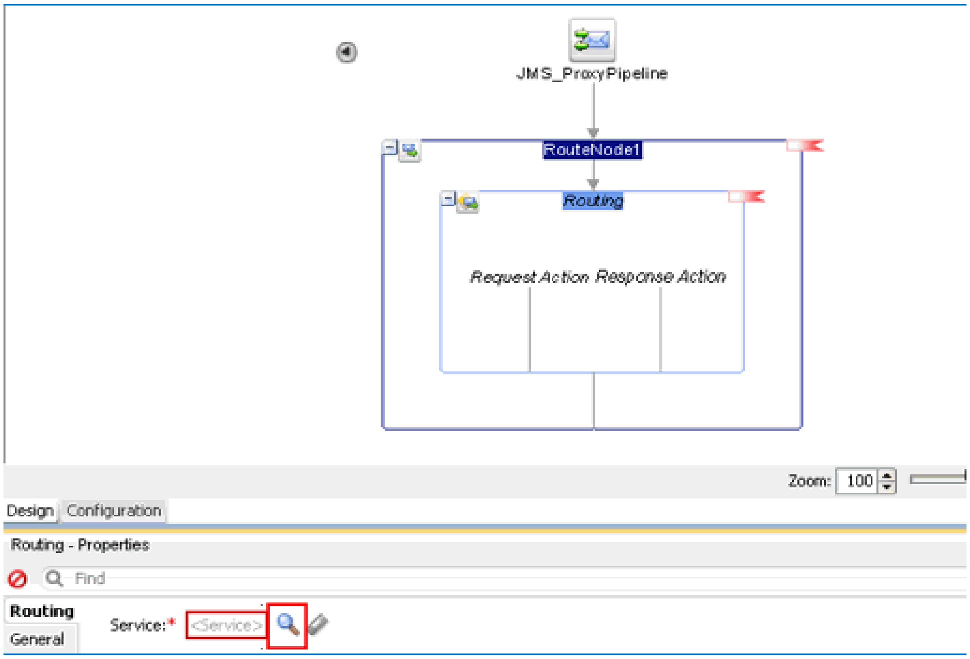
   b. Drag and drop the **Routing** component from the Route section to the area below the Pipeline (for example, JMS_ProxyPipeline), as shown in Figure 8–90.

Figure 8–90 Routing Component



   c. In the Pipeline Configuration page, select **Routing** and click the browse icon to the right of the Service field in the Routing Properties pane, as shown in Figure 8–91.

Figure 8–91 Browse Service



d. In the displayed Resource Chooser window, select the WSDL-based Business service (for example, xxxxx_BS.bix) and click **OK**.

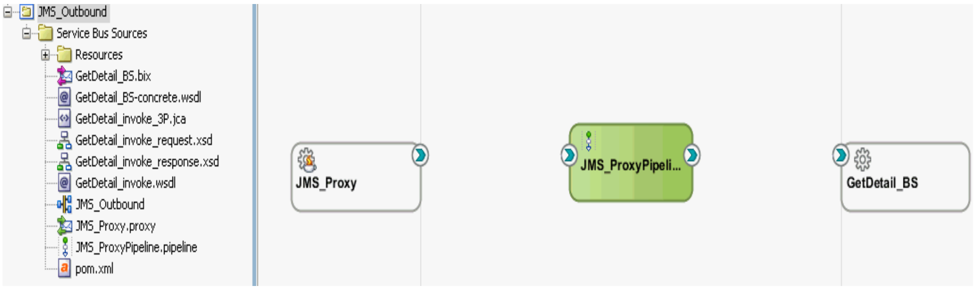
You are returned to the Pipeline configuration page.

e. Save and Close the Pipeline configuration page.

You are returned to the composite editor window.

f. Click **Save All** in the menu bar to save the OSB JMS process, as shown in Figure 8–92.

Figure 8–92 Transport Window

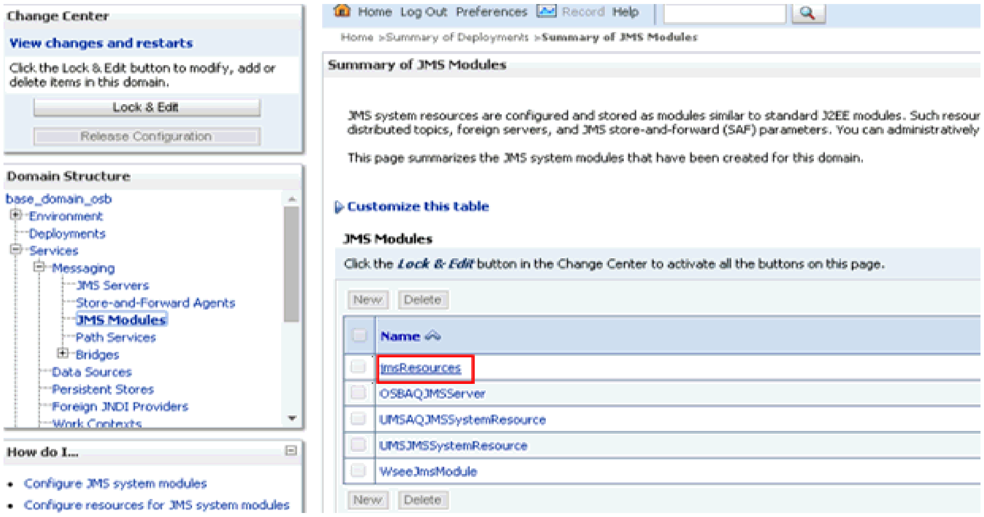


8. Deploy the OSB JMS outbound process. For more information, see Section 8.1.3, "Deploying the OSB Outbound Process" on page 8-16.

9. Once the process is deployed successfully, log on to the Oracle WLS Console.

10. In the Oracle WLS console, expand **Services**, click **Messaging**, select **JMS Modules**, and then click **jmsResources**, as shown in Figure 8–93.

Figure 8–93 JMS Resources



Click the Lock & Edit button to modify, add or delete items in this domain.

JMS system resources are configured and stored as modules similar to standard J2EE modules. Such resour distributed topics, foreign servers, and JMS store-and-forward (SAF) parameters. You can administratively

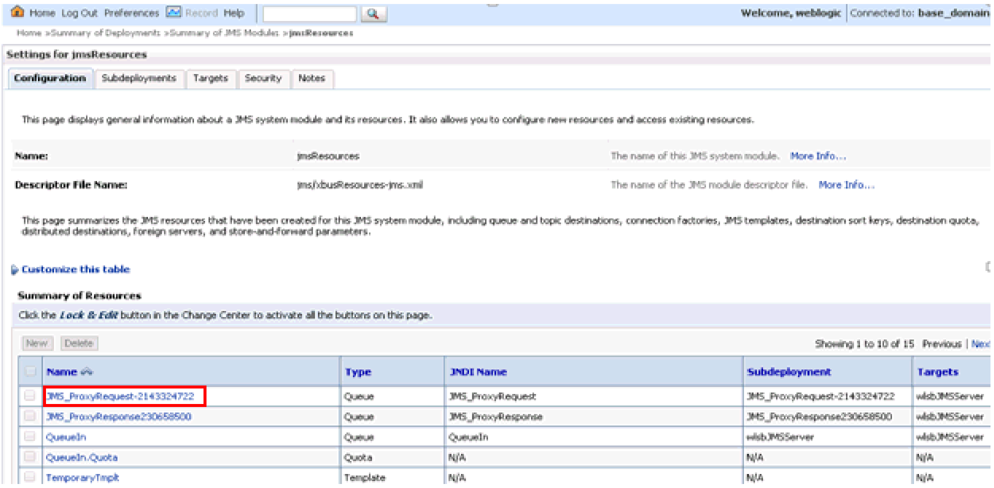
This page summarizes the JMS system modules that have been created for this domain.

Click the *Lock & Edit* button in the Change Center to activate all the buttons on this page.

□	Name
□	jmsResources
□	OSBAQJMSServer
□	UMSAQJMSSystemResource
□	UMSJMSSystemResource
□	WseeJmsModule

11. Click the appropriate request link (for example, JMS_ProxyRequest) as shown in Figure 8–94.

Figure 8–94 JMS_ProxyRequest Link


Home Log Out Preferences Record Help
Welcome, weblogic Connected to: base_domain
Home >Summary of Deployments >Summary of JMS Modules >jmsResources
Configuration Subdeployments Targets Security Notes
Customize this table
New Delete
Showing 1 to 10 of 15 Previous | Next

Settings for jmsResources

This page displays general information about a JMS system module and its resources. It also allows you to configure new resources and access existing resources.

Name: jmsResources — The name of this JMS system module. More Info...

Descriptor File Name: jms/xbusResources-jms.xml — The name of the JMS module descriptor file. More Info...

This page summarizes the JMS resources that have been created for this JMS system module, including queue and topic destinations, connection factories, JMS templates, destination sort keys, destination quota, distributed destinations, foreign servers, and store-and-forward parameters.

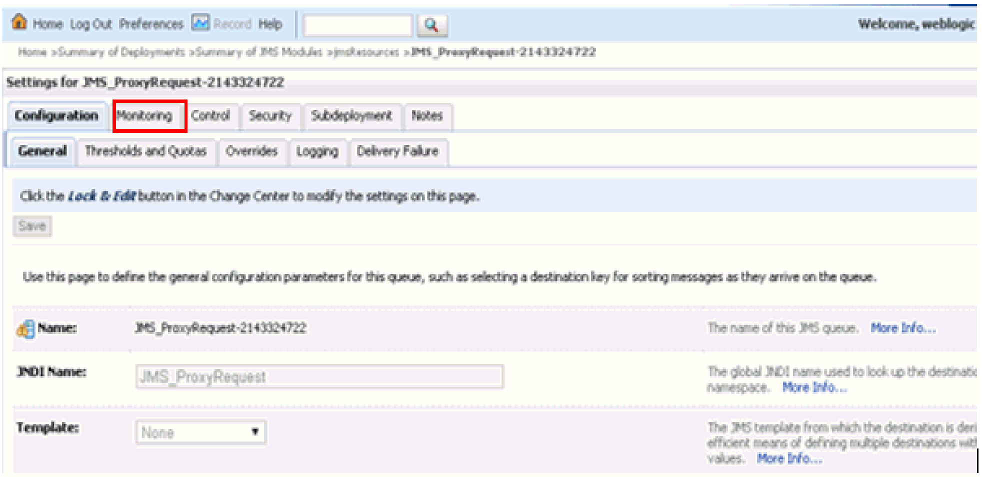
Summary of Resources

Click the *Lock & Edit* button in the Change Center to activate all the buttons on this page.

□	Name	Type	JNDI Name	Subdeployment	Targets
□	JMS_ProxyRequest-2143324722	Queue	JMS_ProxyRequest	JMS_ProxyRequest-2143324722	wlsbJMSServer
□	JMS_ProxyResponse230658500	Queue	JMS_ProxyResponse	JMS_ProxyResponse230658500	wlsbJMSServer
□	QueueIn	Queue	QueueIn	wlsbJMSServer	wlsbJMSServer
□	QueueIn.Quota	Quota	N/A	N/A	N/A
□	TemporaryTmplt	Template	N/A	N/A	N/A

12. Click the Monitoring tab, as shown in Figure 8–95.

Figure 8–95 Monitoring Tab


Home Log Out Preferences Record Help
Welcome, weblogic
Home >Summary of Deployments >Summary of JMS Modules >jmsResources >JMS_ProxyRequest-2143324722
Configuration Monitoring Control Security Subdeployment Notes
General Thresholds and Quotas Overrides Logging Delivery Failure
Save

Settings for JMS_ProxyRequest-2143324722

Click the *Lock & Edit* button in the Change Center to modify the settings on this page.

Use this page to define the general configuration parameters for this queue, such as selecting a destination key for sorting messages as they arrive on the queue.

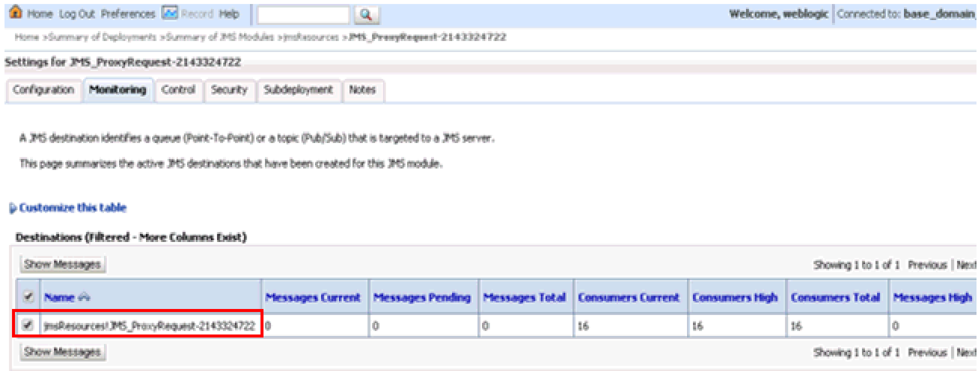
Name: JMS_ProxyRequest-2143324722 — The name of this JMS queue. More Info...

JNDI Name: JMS_ProxyRequest — The global JNDI name used to look up the destinati namespace. More Info...

Template: None — The JMS template from which the destination is deri efficient means of defining multiple destinations wit values. More Info...

13. Select the check box and click the **Show Messages** button, as shown in Figure 8–96.

Figure 8–96 Show Messages Button


Home Log Out Preferences Record Help
Welcome, weblogic Connected to: base_domain
Home >Summary of Deployments >Summary of JMS Modules >jmsResources >JMS_ProxyRequest-2143324722
Configuration Monitoring Control Security Subdeployment Notes
Customize this table
Show Messages
Showing 1 to 1 of 1 Previous | Next

Settings for JMS_ProxyRequest-2143324722

A JMS destination identifies a queue (Point-To-Point) or a topic (Pub/Sub) that is targeted to a JMS server.

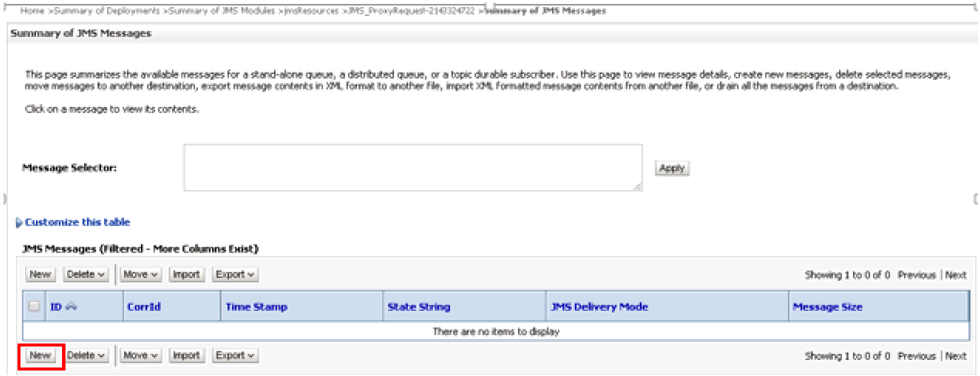
This page summarizes the active JMS destinations that have been created for this JMS module.

Destinations (Filtered - More Columns Exist)

☑	Name	Messages Current	Messages Pending	Messages Total	Consumers Current	Consumers High	Consumers Total	Messages High
☑	jmsResources!JMS_ProxyRequest-2143324722	0	0	0	16	16	16	0

14. Click **New**, as shown in Figure 8–97.

Figure 8–97 JMS Messages


Home >Summary of Deployments >Summary of JMS Modules >jmsResources >JMS_ProxyRequest-2143324722 >Summary of JMS Messages
Apply
Customize this table
New
Delete
Move
Import
Export
Showing 1 to 0 of 0 Previous | Next

Summary of JMS Messages

This page summarizes the available messages for a stand-alone queue, a distributed queue, or a topic durable subscriber. Use this page to view message details, create new messages, delete selected messages, move messages to another destination, export message contents in XML format to another file, import XML formatted message contents from another file, or drain all the messages from a destination.

Click on a message to view its contents.

Message Selector:

JMS Messages (Filtered - More Columns Exist)

□	ID	CorrId	Time Stamp	State String	JMS Delivery Mode	Message Size
There are no items to display						

15. Provide the input payload in the Body field and click **OK**.

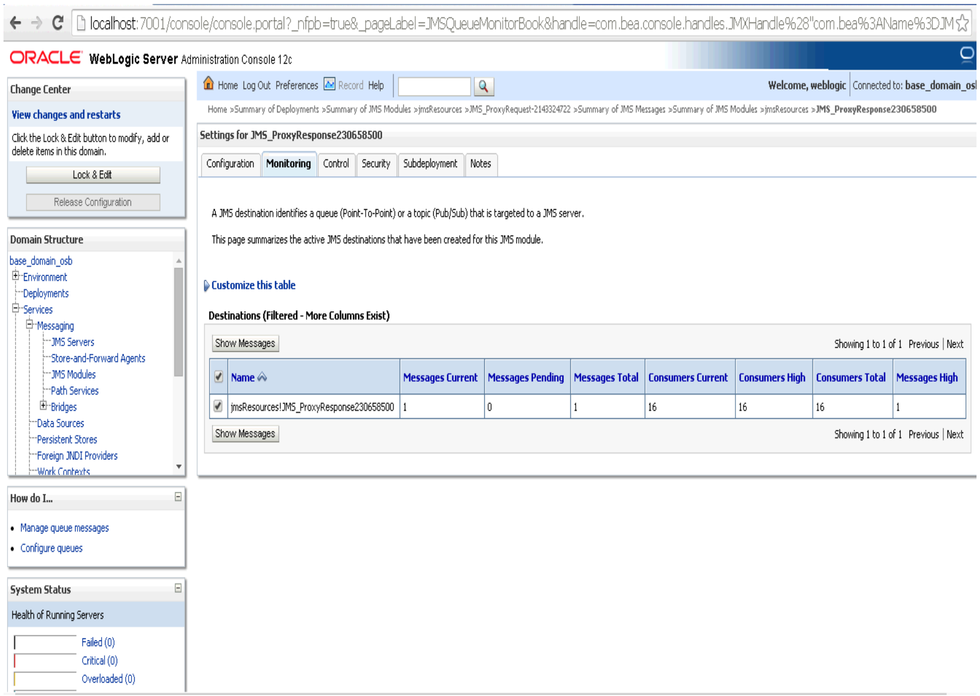
16. In the Oracle WLS console, expand **Services**, click **Messaging**, select **JMS Modules**, and then click **jmsResources**.

17. Click the appropriate response link (for example, JMS_ProxyResponse).

18. Click the Monitoring tab.

19. Select the check box and click **Show Messages**, as shown in Figure 8–98.

Figure 8–98 Destination Messages


localhost:7001/console/console.portal?_nfpb=true&_pageLabel=JMSQueueMonitorBook&handle=com.bea.console.handles.JMXHandle%28"com.bea%3AName%3DJM
ORACLE WebLogic Server Administration Console 12c
Home Log Out Preferences Record Help
Welcome, weblogic
Connected to: base_domain_osl
Change Center
View changes and restarts
Lock & Edit
Release Configuration
Domain Structure
base_domain_osb
Environment
Deployments
Services
Messaging
JMS Servers
Store-and-Forward Agents
JMS Modules
Path Services
Bridges
Data Sources
Persistent Stores
Foreign JNDI Providers
Work Contexts
How do I...
Manage queue messages
Configure queues
System Status
Health of Running Servers
Failed (0)
Critical (0)
Overloaded (0)
Home >Summary of Deployments >Summary of JMS Modules >jmsResources >JMS_ProxyRequest-2143324722 >Summary of JMS Messages >Summary of JMS Modules >jmsResources >JMS_ProxyResponse230658500
Configuration
Monitoring
Control
Security
Subdeployment
Notes
Customize this table
Show Messages
Showing 1 to 1 of 1 Previous | Next

Click the Lock & Edit button to modify, add or delete items in this domain.

Settings for JMS_ProxyResponse230658500

A JMS destination identifies a queue (Point-To-Point) or a topic (Pub/Sub) that is targeted to a JMS server.

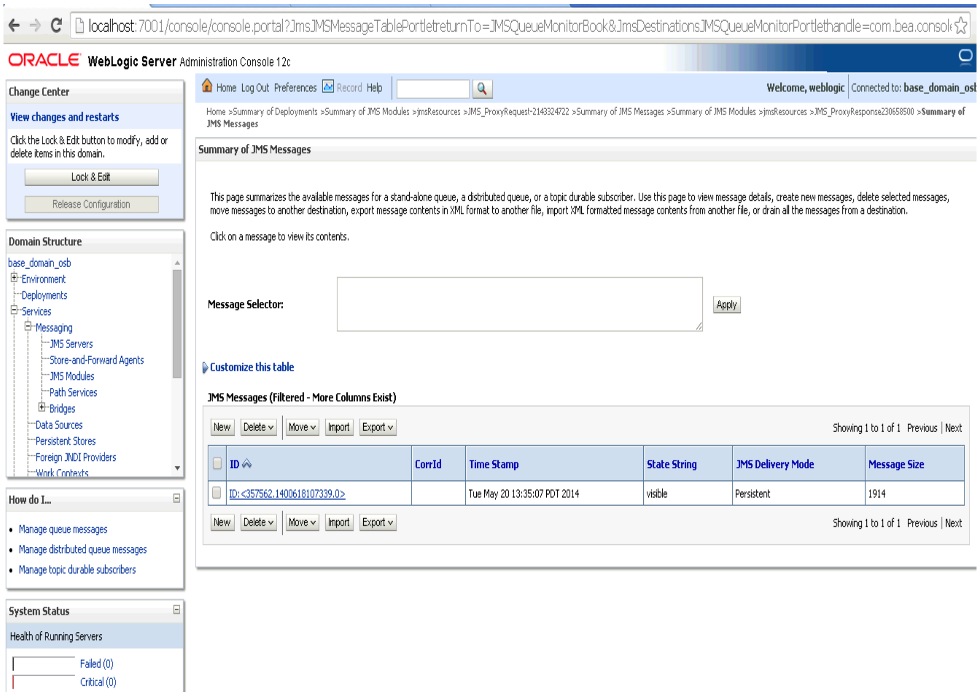
This page summarizes the active JMS destinations that have been created for this JMS module.

Destinations (Filtered - More Columns Exist)

☑	Name	Messages Current	Messages Pending	Messages Total	Consumers Current	Consumers High	Consumers Total	Messages High
☑	jmsResources!JMS_ProxyResponse230658500	1	0	1	16	16	16	1

20. Click the ID link with the appropriate time and date, as shown in Figure 8–99.

Figure 8–99 Summary of JMS Messages Window



localhost:7001/console/console.portal?JmsJMSMessageTablePortletreturnTo=JMSQueueMonitorBook&JmsDestinationsJMSQueueMonitorPortlethandle=com.bea.consol
ORACLE WebLogic Server Administration Console 12c
Home Log Out Preferences Record Help
Welcome, weblogic
Connected to: base_domain_osb
Change Center
View changes and restarts
Lock & Edit
Release Configuration
Domain Structure
base_domain_osb
Environment
Deployments
Services
Messaging
JMS Servers
Store-and-Forward Agents
JMS Modules
Path Services
Bridges
Data Sources
Persistent Stores
Foreign JNDI Providers
Work Contexts
How do I...
Manage queue messages
Manage distributed queue messages
Manage topic durable subscribers
System Status
Health of Running Servers
Failed (0)
Critical (0)
Apply
Customize this table
New
Delete
Move
Import
Export
Showing 1 to 1 of 1 Previous | Next

Click the Lock & Edit button to modify, add or delete items in this domain.

Home >Summary of Deployments >Summary of JMS Modules >jmsResources >JMS_ProxyRequest-2143324722 >Summary of JMS Messages >Summary of JMS Modules >jmsResources >JMS_ProxyResponse230658500 >**Summary of JMS Messages**

Summary of JMS Messages

This page summarizes the available messages for a stand-alone queue, a distributed queue, or a topic durable subscriber. Use this page to view message details, create new messages, delete selected messages, move messages to another destination, export message contents in XML format to another file, import XML formatted message contents from another file, or drain all the messages from a destination.

Click on a message to view its contents.

Message Selector:

JMS Messages (Filtered - More Columns Exist)

□	ID	CorrId	Time Stamp	State String	JMS Delivery Mode	Message Size
□	ID:<357562.1400618107339.0>		Tue May 20 13:35:07 PDT 2014	visible	Persistent	1914

The response document is shown under the Text field.

8.6 Configuring an HTTP Outbound Process Using JDeveloper (J2CA Configuration)

This section describes how to configure HTTP Outbound process to your Siebel system, using Oracle JDeveloper for J2CA configurations.

1. Before you design an HTTP Outbound process, you must generate the respective WSDL file using Application Explorer. For more information, see Section 4.4.1, "Generating WSDL for Request/Response Service" on page 4-8.

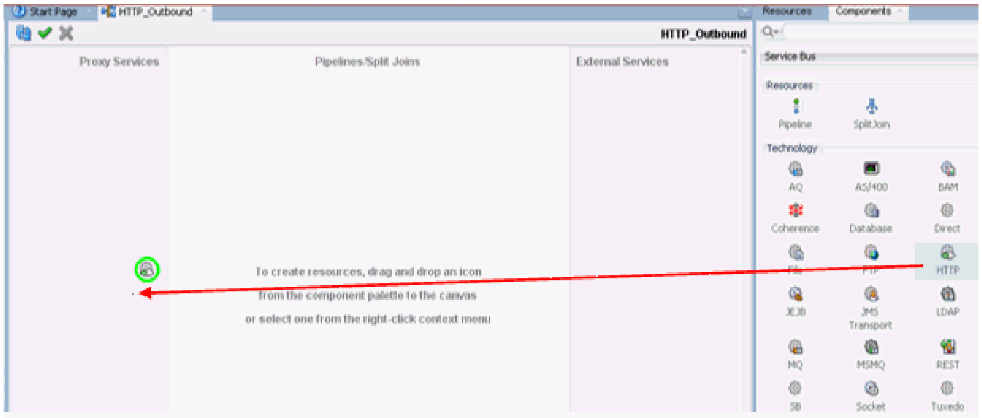
2. Start the Oracle JDeveloper and create a Service Bus Application for OSB. For more information, see Section 8.1.1, "Creating a Service Bus Application for OSB" on page 8-2.

3. Create a Third Party Adapter Service Component. For more information, see Section 8.1.2.1, "Configuring a Third-Party Adapter Service Component" on page 8-3.

4. Create an HTTP Proxy Service with a Pipeline and perform the following steps:

   a. Drag and drop the **HTTP** component from the Technology Components pane to the Proxy Services pane, as shown in Figure 8–100.

Figure 8–100 HTTP Component



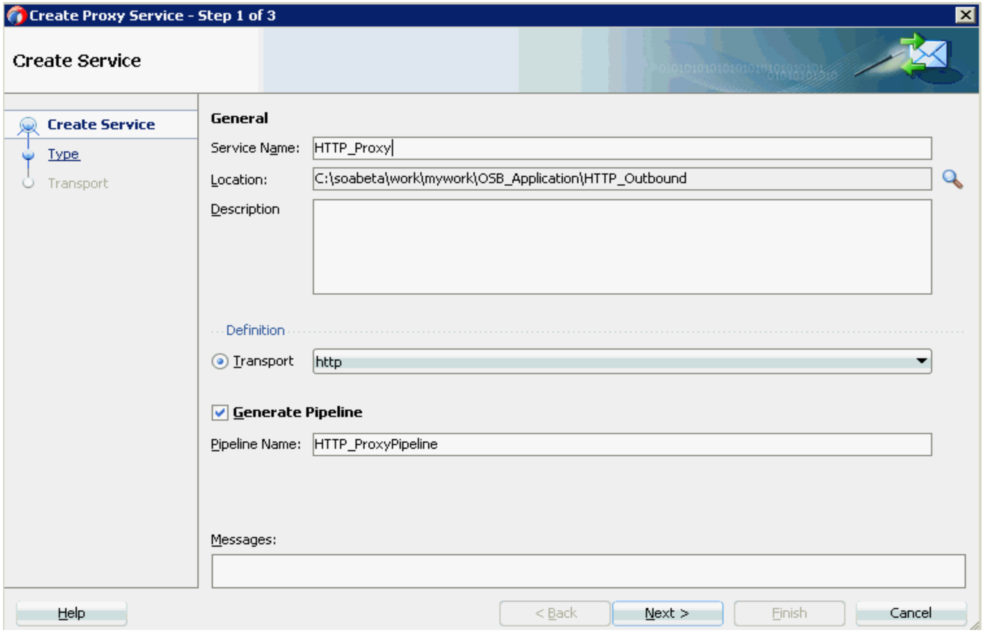
To create resources, drag and drop an icon from the component palette to the canvas or select one from the right-click context menu

The Create Proxy Service dialog is displayed.

b. In the Service Name field, enter any name you wish for the Proxy service (for example, HTTP_Proxy). By default, Generate Pipeline is selected.

c. Click **Next**, as shown in Figure 8–101.

Figure 8–101 Create Proxy Service Pane



General

Service Name: HTTP_Proxy

Location: C:\soabeta\work\mywork\OSB_Application\HTTP_Outbound

Description

Definition

● Transport http

☑ Generate Pipeline

Pipeline Name: HTTP_ProxyPipeline

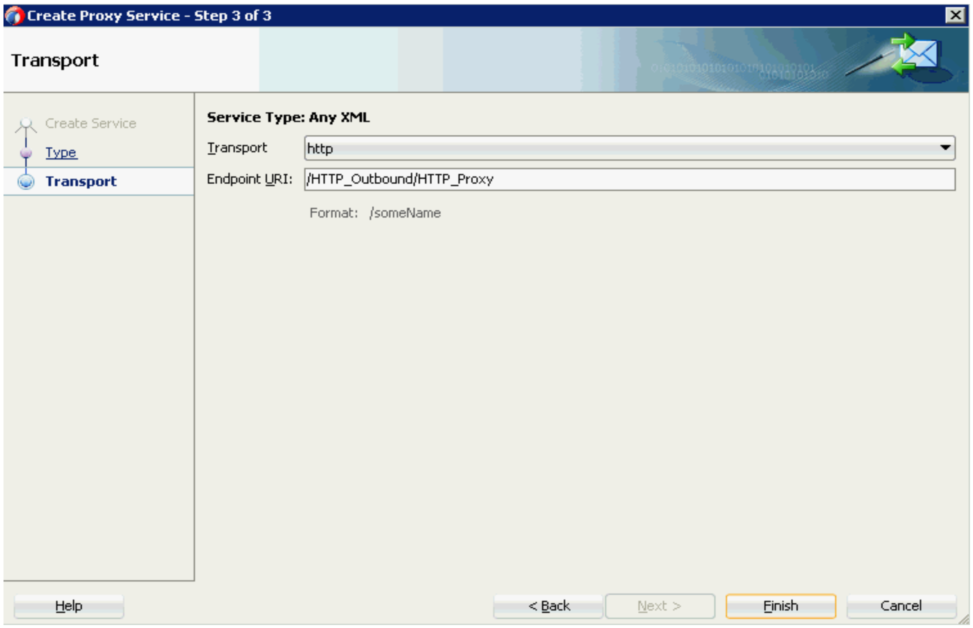
Messages:

d. In the displayed Type window, select **Any XML** and then click **Next**.

The Transport window is displayed.

e. Leave the default values and then click **Finish**, as shown in Figure 8–102.

Figure 8–102 Transport Window



Service Type: Any XML

Transport: http

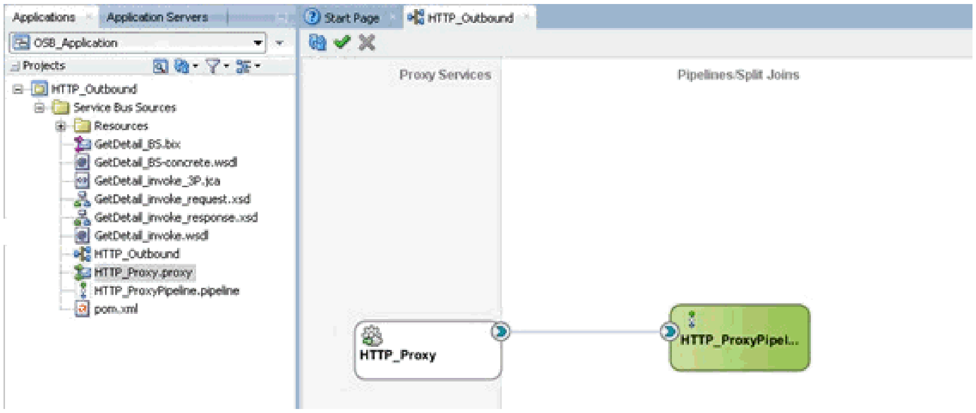
Endpoint URI: /HTTP_Outbound/HTTP_Proxy

Format: /someName

The HTTP Proxy service along with the pipeline is created and displayed.

f. Double-click the created pipeline (for example, HTTP_ProxyPipeline) in the Pipelines/Split Joins pane, as shown in Figure 8–103.

Figure 8–103 Proxy Service

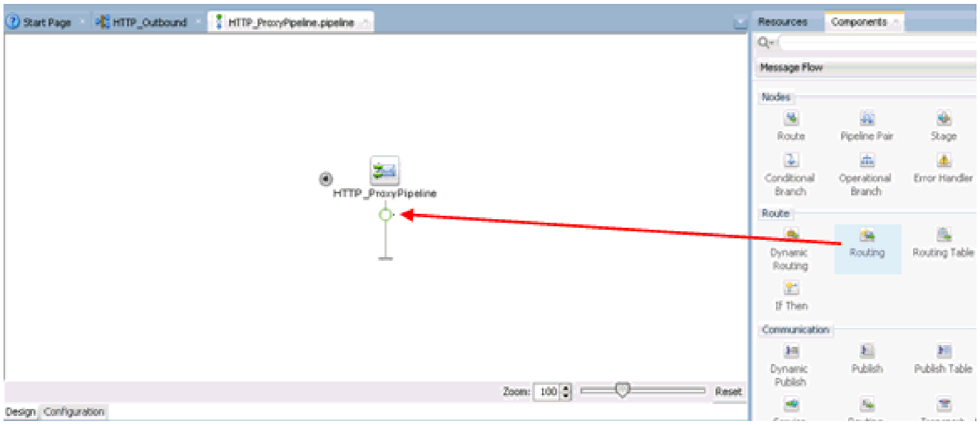


The Pipeline Configuration page is displayed.

5. Configure the Routing Rules and proceed with the following steps:

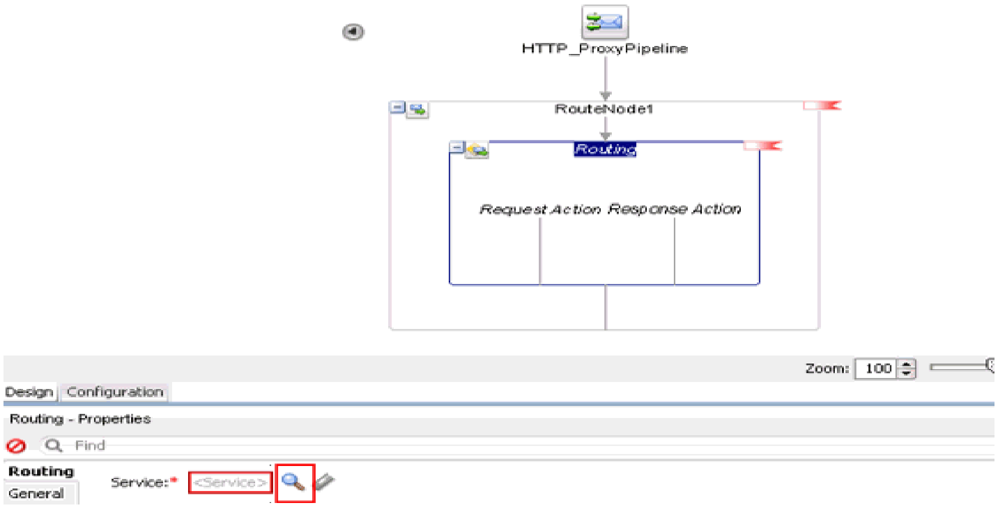
a. Drag and drop the **Routing** component from the Route section to the area below the Pipeline (for example, HTTP_ProxyPipeline), as shown in Figure 8–104.

Figure 8–104 Routing Component



b. In the Pipeline Configuration page, select **Routing** and click the browse icon to the right of the Service field in the Routing Properties pane, as shown in Figure 8–105.

Figure 8–105 Browse Service



c. In the displayed Resource Chooser window, select the WSDL-based Business service (for example, xxxxx_BS.bix) and click **OK**.

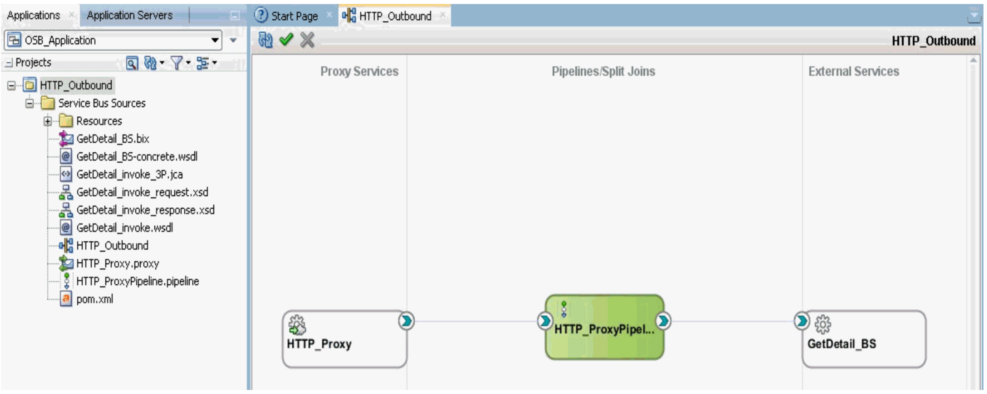
You are returned to the Pipeline configuration page.

d. Save and Close the Pipeline configuration page.

You are returned to the composite editor window.

e. Click **Save All** in the menu bar to save the OSB HTTP process, as shown in Figure 8–106.

Figure 8–106 Transport Window

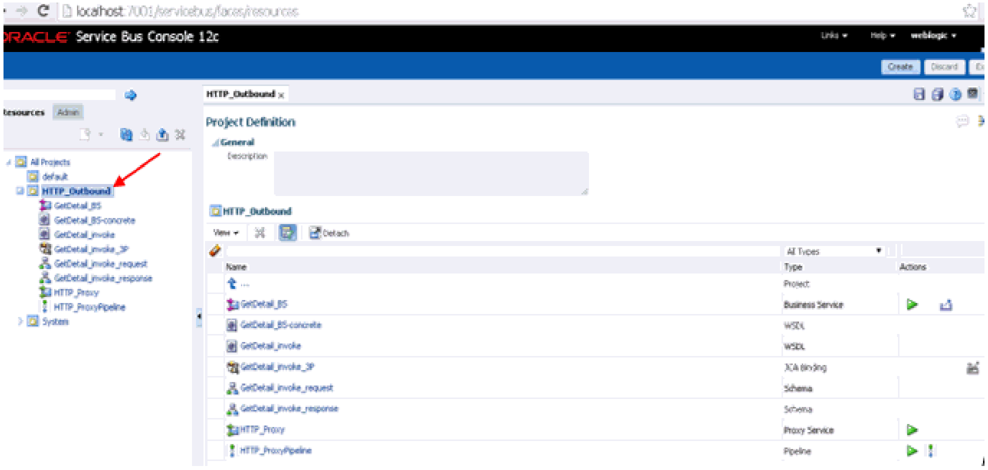


6. Deploy the OSB HTTP outbound process. For more information, see Section 8.1.3, "Deploying the OSB Outbound Process" on page 8-16.

7. Once the process is deployed successfully, log on to the Service Bus Console.

8. In the Service Bus console, click on the deployed HTTP Outbound project (for example, HTTP_Outbound), as shown in Figure 8–107.

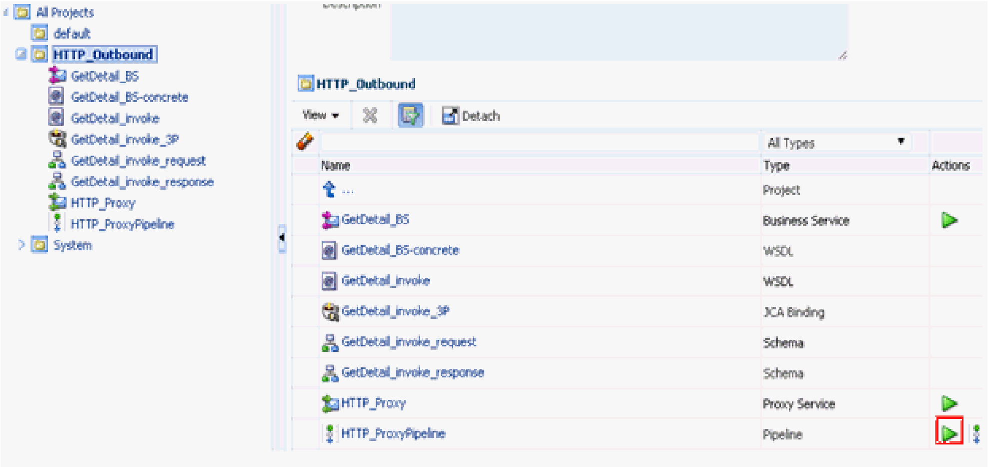
Figure 8–107 Service Bus Console



Name	Type	Actions
...	Project	
GetDetail_BS	Business Service	
GetDetail_BS-concrete	WSDL	
GetDetail_invoke	WSDL	
GetDetail_invoke_3P	JCA Binding	
GetDetail_invoke_request	Schema	
GetDetail_invoke_response	Schema	
HTTP_Proxy	Proxy Service	
HTTP_ProxyPipeline	Pipeline	

9. Click on the Test OSB Console icon for the created pipeline, as shown in Figure 8–108.

Figure 8–108 Test OSB Console Icon



Name	Type	Actions
...	Project	
GetDetail_BS	Business Service	▶
GetDetail_BS-concrete	WSDL	
GetDetail_invoke	WSDL	
GetDetail_invoke_3P	JCA Binding	
GetDetail_invoke_request	Schema	
GetDetail_invoke_response	Schema	
HTTP_Proxy	Proxy Service	▶
HTTP_ProxyPipeline	Pipeline	▶

10. In the displayed Test OSB Console page, provide the input XML and click the **Execute** button.

    In the displayed Test OSB Console page, the response is received.

9 Key Features

This chapter describes new features for the Oracle Application Adapter for Siebel. This chapter contains the following sections:

- Section 9.1, "Configuring the Logging Feature"
- Section 9.2, "Configuring the Diagnosibility Feature"
- Section 9.3, "Configuring the SOA Debugging Feature"
- Section 9.4, "Exception Filter"
- Section 9.5, "Credential Mapping for Oracle SOA Suite (BPEL, Mediator, or BPM)"
- Section 9.6, "Credential Mapping for Oracle Service Bus (OSB) Using JDeveloper"

9.1 Configuring the Logging Feature

In Oracle 12*c* (12.2.1.0.0), J2CA and BSE adapter logs will be updated in Oracle logs in the *{server-name}*-diagnostic.log file available in the following location:

```
<ORACLE_HOME>\user_projects\domains\base_domain\servers\<server_Name>\logs.
```

> **Note:** The Application Explorer log files for J2CA would be created under the <ADAPTER_HOME>\config\xxxxxxx\log folder where *xxxxxxx* is the name of the J2CA configuration that was created in Application Explorer. Each J2CA configuration in Application Explorer has a corresponding log folder under the named J2CA configuration folder.

This section describes how to configure the Logging feature. It contains the following topics:

- Section 9.1.1, "Configuring Log File Management for the J2CA Connector Application"
- Section 9.1.2, "Configuring Log File Management for Business Services Engine (BSE)"

9.1.1 Configuring Log File Management for the J2CA Connector Application

Log file management for the J2CA Connector Application is governed by the Loggers defined in:

```
<ORACLE_HOME>\user_projects\domains\base_
```

```
domain\config\fmwconfig\servers\${server-name}\logging.xml
```

Any new loggers will have to be added to this file if they are to be managed from the em console.

For example:

```
<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.Siebel'
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.Siebel' 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.Siebel' 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.Siebel'
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.Siebel' useParentHandlers='true'/>
```

This sets the logging level of all the loggers under oracle.soa.adapter.iwaf to NOTIFICATION:1 (INFO), which is the default setting level by Oracle.

The logging level of all the loggers can also be configured from the em console with the following steps:

1. Start the Oracle WebLogic Server for the Oracle WebLogic Server domain that you configured.
2. Open the Oracle WebLogic Server Enterprise Manager Console in a web browser by entering the following URL:

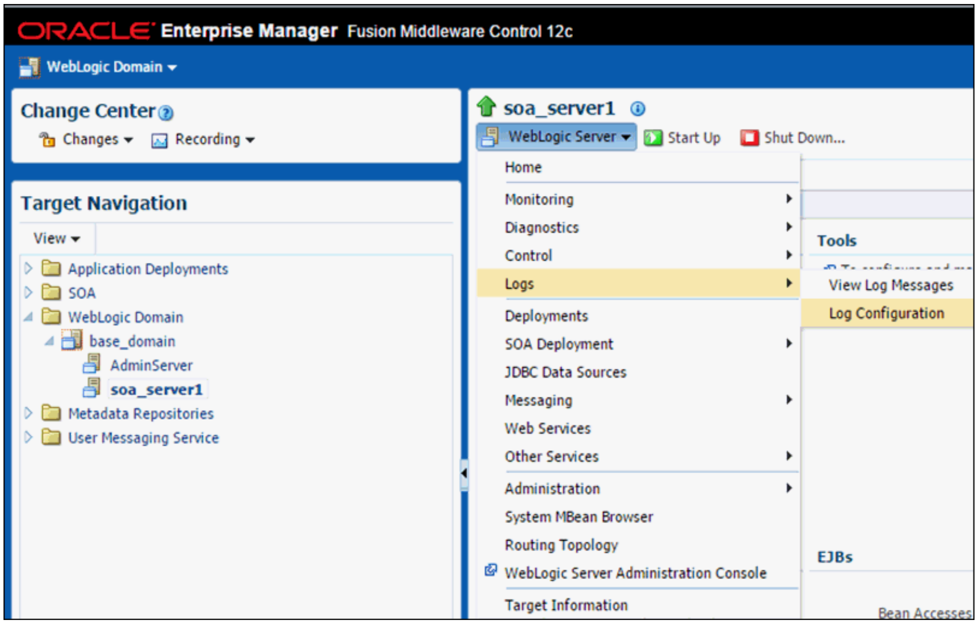
   `http://host name:port/em`

   where `host name` is the name of the system where Oracle WebLogic Server is running and `port` is the port for the Oracle WebLogic Server that is running. The default port for the Oracle WebLogic Server is 7001. However, this value can vary between installations.
3. Log in to the Oracle WebLogic Server Administrative Console using an account that has administrator privileges.

4. Under the Target Navigation pane, click **Weblogic Domain**, select **Domain Created**, and click the appropriate server (Managed Server or Integrated Server).

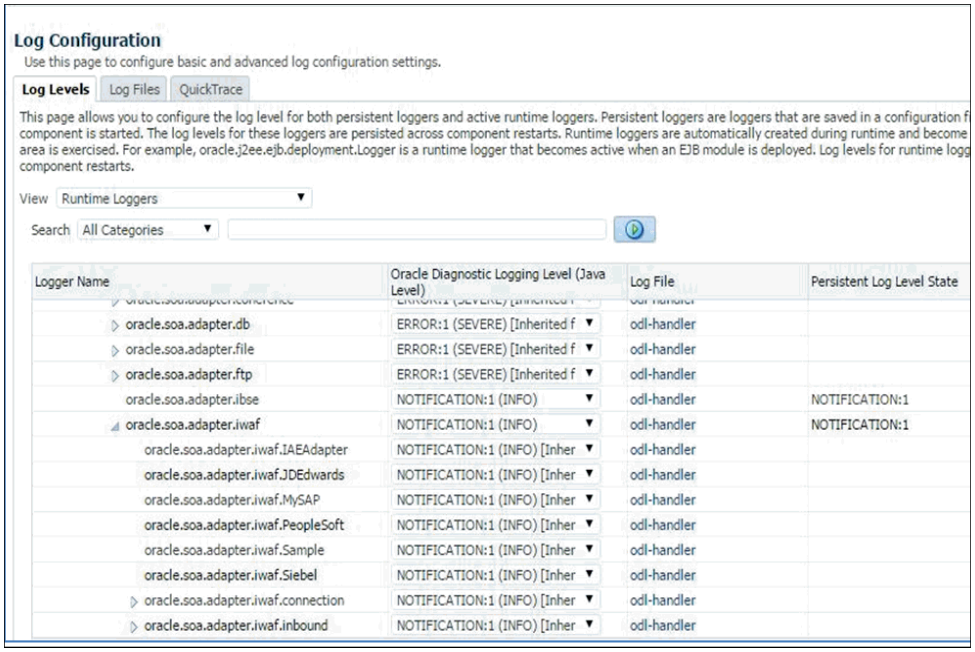
5. In the Server pane, expand WebLogic Server, select **Logs**, and then click **Log Configuration**, as shown in Figure 9–1.

Figure 9–1 Log Configuration Option



6. Under the **Log Levels** tab, expand the Oracle root logger until oracle.soa.adapter.iwaf is visible, as shown in Figure 9–2.

Figure 9–2 Log Levels Tab



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 fi component is started. The log levels for these loggers are persisted across component restarts. Runtime loggers are automatically created during runtime and become area is exercised. For example, oracle.j2ee.ejb.deployment.Logger is a runtime logger that becomes active when an EJB module is deployed. Log levels for runtime logg component restarts.

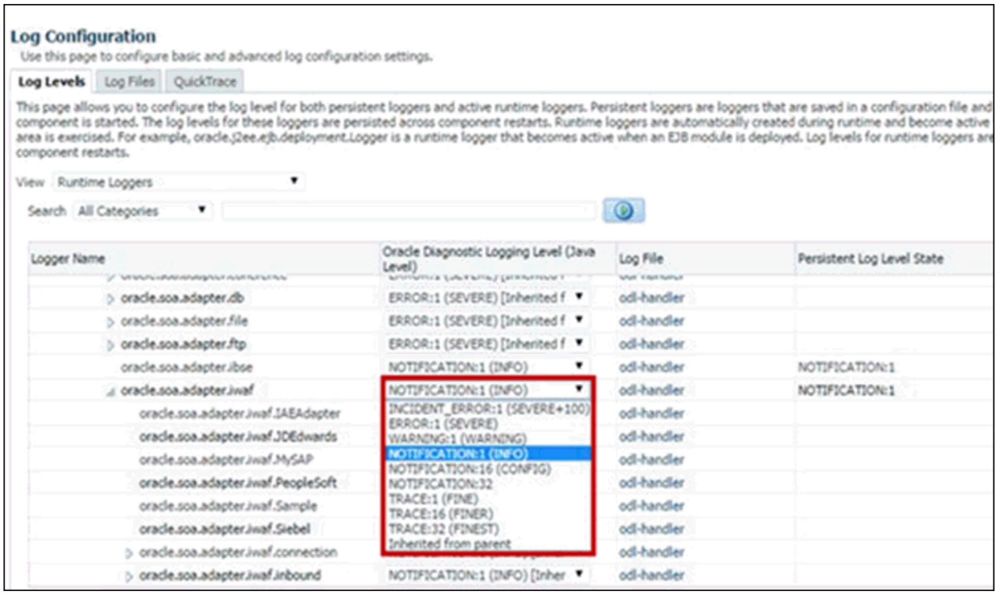
View: Runtime Loggers

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 f	odl-handler	
oracle.soa.adapter.file	ERROR:1 (SEVERE) [Inherited f	odl-handler	
oracle.soa.adapter.ftp	ERROR:1 (SEVERE) [Inherited f	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.IAEAdapter	NOTIFICATION:1 (INFO) [Inher	odl-handler	
oracle.soa.adapter.iwaf.JDEdwards	NOTIFICATION:1 (INFO) [Inher	odl-handler	
oracle.soa.adapter.iwaf.MySAP	NOTIFICATION:1 (INFO) [Inher	odl-handler	
oracle.soa.adapter.iwaf.PeopleSoft	NOTIFICATION:1 (INFO) [Inher	odl-handler	
oracle.soa.adapter.iwaf.Sample	NOTIFICATION:1 (INFO) [Inher	odl-handler	
oracle.soa.adapter.iwaf.Siebel	NOTIFICATION:1 (INFO) [Inher	odl-handler	
oracle.soa.adapter.iwaf.connection	NOTIFICATION:1 (INFO) [Inher	odl-handler	
oracle.soa.adapter.iwaf.inbound	NOTIFICATION:1 (INFO) [Inher	odl-handler	

7. In the Oracle Diagnostic Logging Level (Java Level) column, select the required log level from the oracle.soa.adapter.iwaf drop-down list, as shown in Figure 9–3.

Figure 9–3 Oracle Diagnostic Logging Level (Java Level) Column



Log Configuration

Use this page to configure basic and advanced log configuration settings.

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 and component is started. The log levels for these loggers are persisted across component restarts. Runtime loggers are automatically created during runtime and become active area is exercised. For example, oracle.j2ee.ejb.deployment.Logger is a runtime logger that becomes active when an EJB module is deployed. Log levels for runtime loggers are component restarts.

View: Runtime Loggers

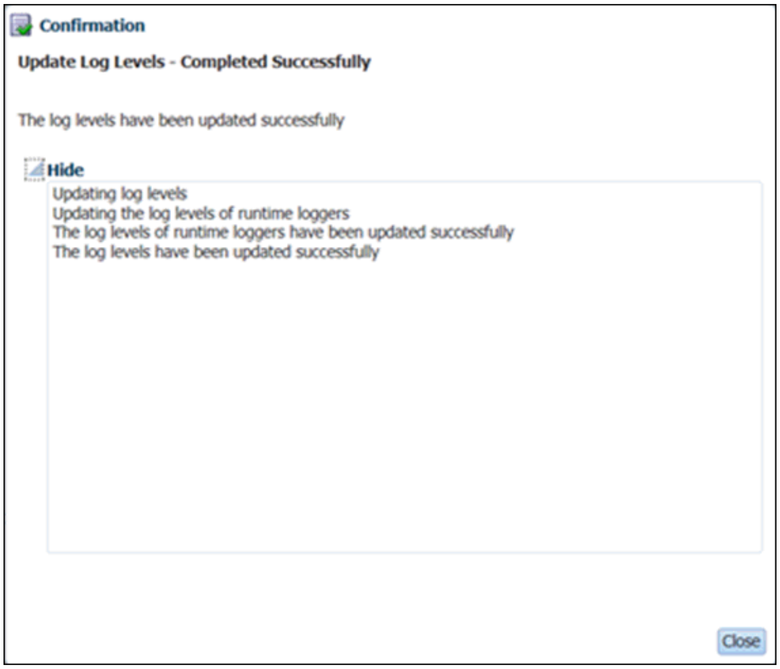
Search: All Categories

Logger Name	Oracle Diagnostic Logging Level (Java Level)	Log File	Persistent Log Level State
oracle.soa.adapter.db	ERROR:1 (SEVERE) [Inherited f	odl-handler	
oracle.soa.adapter.file	ERROR:1 (SEVERE) [Inherited f	odl-handler	
oracle.soa.adapter.ftp	ERROR:1 (SEVERE) [Inherited f	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.IAEAdapter	INCIDENT_ERROR:1 (SEVERE+100)	odl-handler	
oracle.soa.adapter.iwaf.JDEdwards	ERROR:1 (SEVERE)	odl-handler	
oracle.soa.adapter.iwaf.MySAP	WARNING:1 (WARNING)	odl-handler	
oracle.soa.adapter.iwaf.PeopleSoft	NOTIFICATION:1 (INFO)	odl-handler	
oracle.soa.adapter.iwaf.Sample	NOTIFICATION:16 (CONFIG) NOTIFICATION:32	odl-handler	
oracle.soa.adapter.iwaf.Siebel	TRACE:1 (FINE) TRACE:16 (FINER) TRACE:32 (FINEST)	odl-handler	
oracle.soa.adapter.iwaf.connection	Inherited from parent	odl-handler	
oracle.soa.adapter.iwaf.inbound	NOTIFICATION:1 (INFO) [Inher	odl-handler	

8. Click **Apply**.

   A confirmation message appears, indicating that the update was completed successfully, as shown in Figure 9–4.

Figure 9–4 Confirmation Message



Confirmation

Update Log Levels - Completed Successfully

The log levels have been updated successfully

Hide

Updating log levels
Updating the log levels of runtime loggers
The log levels of runtime loggers have been updated successfully
The log levels have been updated successfully

9. Click **Close**.

The following table shows how the Log Level property is updated in the *{server-name}*-diagnostic.log based on the corresponding Log Level property settings in the em console.

Table 9–1 Log Level Property Settings

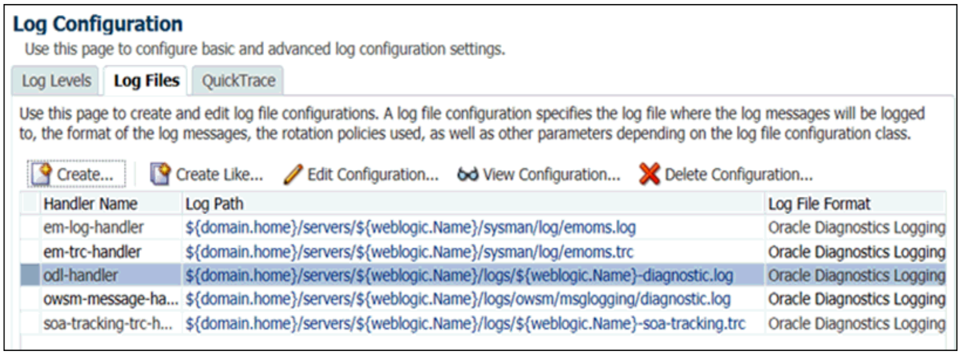
Log Level Set in Em Console	Log Level Updated in {server-name}-diagnostic.log
ERROR:1	ERROR
WARNING:1	WARNING
NOTIFICATION:1	NOTIFICATION
NOTIFICATION:16	NOTIFICATION
NOTIFICATION:32	NOTIFICATION
TRACE:1	NOTIFICATION
TRACE:16	NOTIFICATION
TRACE:32	TRACE:32

Note: Setting the LogLevel as TRACE:32 in the em console, displays the FINEST details in the log (displaying the input passed to the adapter, response received from the adapter and other additional details) with the log level displayed as TRACE:32 in the *{server-name}*-diagnostic.log.

For development and test environments, TRACE:32 is the preferred log level, which displays all of the log details. For production environments, ERROR is the preferred log level.

The log messages are written to a disk file, and the file path can be found in the **Log Files** tab. The Handler Name in the Log Files table corresponds to the Log File name in the Log Levels table. All loggers in the hierarchy below oracle.soa.adapter are currently handled by the odl-handler, as shown in Figure 9–5.

Figure 9–5 Log Configuration Pane



Log Configuration

Use this page to configure basic and advanced log configuration settings.

Use this page to create and edit log file configurations. A log file configuration specifies the log file where the log messages will be logged to, the format of the log messages, the rotation policies used, as well as other parameters depending on the log file configuration class.

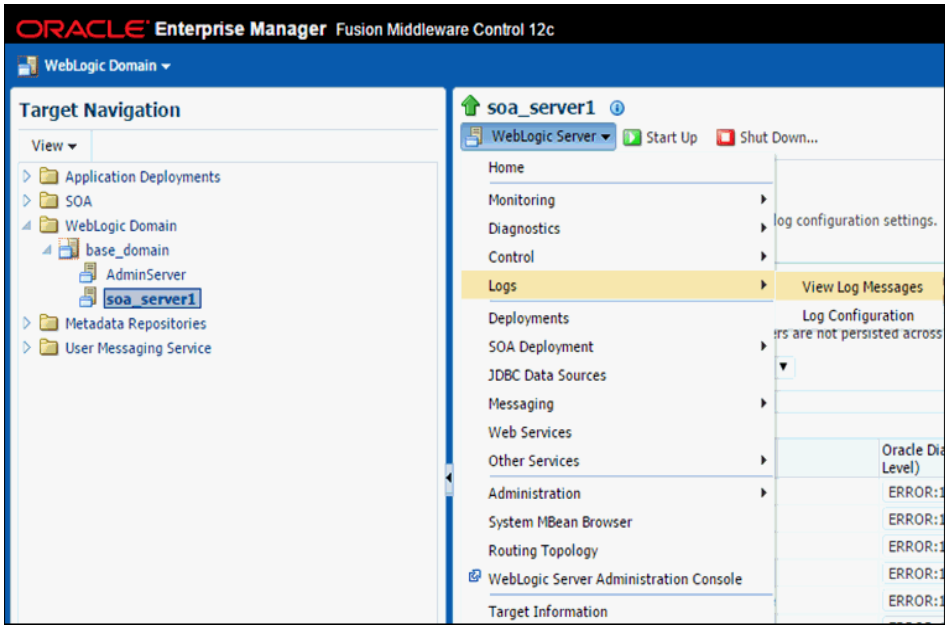
Handler Name	Log Path	Log File Format
em-log-handler	${domain.home}/servers/${weblogic.Name}/sysman/log/emoms.log	Oracle Diagnostics Logging
em-trc-handler	${domain.home}/servers/${weblogic.Name}/sysman/log/emoms.trc	Oracle Diagnostics Logging
odl-handler	${domain.home}/servers/${weblogic.Name}/logs/${weblogic.Name}-diagnostic.log	Oracle Diagnostics Logging
owsm-message-ha...	${domain.home}/servers/${weblogic.Name}/logs/owsm/msglogging/diagnostic.log	Oracle Diagnostics Logging
soa-tracking-trc-h...	${domain.home}/servers/${weblogic.Name}/logs/${weblogic.Name}-soa-tracking.trc	Oracle Diagnostics Logging

The logs are updated in *{server-name}*-diagnostic.log available in the following location:

```
<ORACLE_HOME>\user_projects\domains\base_domain\servers\<server_Name>\logs
```

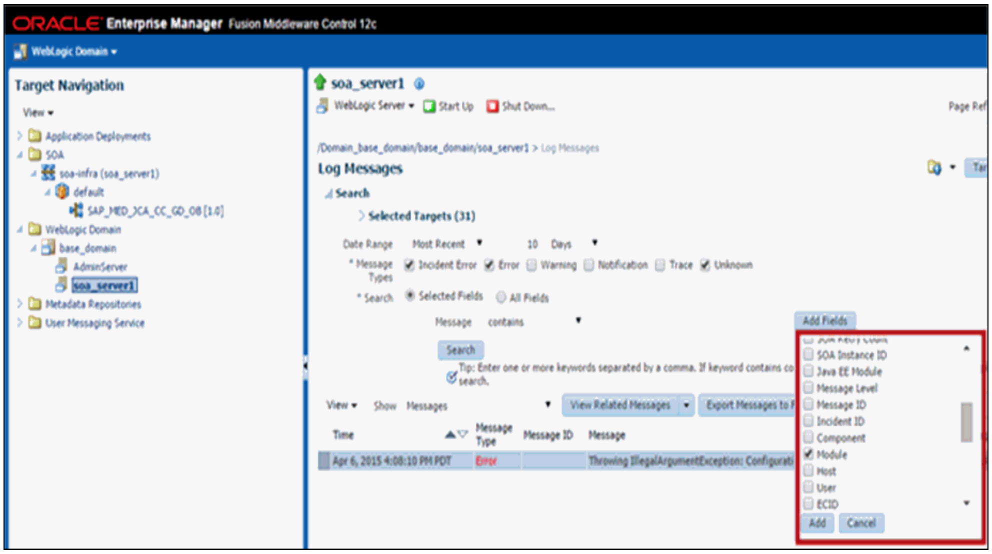
10. In the Server pane, display the log messages in the em console by clicking the **Weblogic Server** drop-down list, selecting **Logs**, and then clicking **View Log Messages**, as shown in Figure 9–6.

Figure 9–6 View Log Messages Option



11. On the Log Messages pane, complete the required search criteria. You can also add the **Module** field to the search criteria, which contains the name of the logger of interest, as shown in Figure 9–7.

Figure 9–7 Module Field



Date Range: Most Recent 10 Days

* Message Types: ☑ Incident Error ☑ Error □ Warning □ Notification □ Trace ☑ Unknown

* Search: ● Selected Fields ○ All Fields

Message contains

Tip: Enter one or more keywords separated by a comma. If keyword contains co search.

Show: Messages

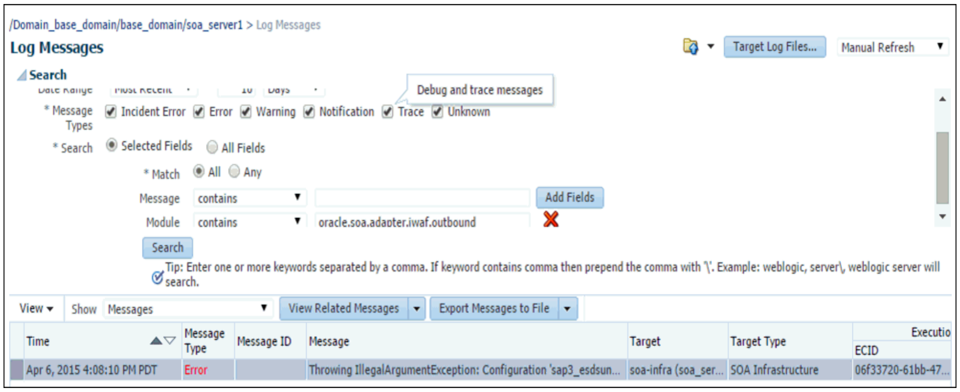
□ SOA Instance ID
□ Java EE Module
□ Message Level
□ Message ID
□ Incident ID
□ Component
☑ Module
□ Host
□ User
□ ECID

Time	Message Type	Message ID	Message
Apr 6, 2015 4:08:10 PM PDT	Error		Throwing IllegalArgumentException: Configurati

12. Click **Add**.

13. In the Module field, enter the name of the logger of interest, and if required, select the additional Message Types (Warning, Notification, Trace, and so on) and then click **Search**, as shown in Figure 9–8.

Figure 9–8 Log Messages Pane



/Domain_base_domain/base_domain/soa_server1 > Log Messages

Log Messages

Date Range Most Recent 10 Days

Debug and trace messages

* Message Types ☑ Incident Error ☑ Error ☑ Warning ☑ Notification ☑ Trace ☑ Unknown

* Search ● Selected Fields ○ All Fields

* Match ● All ○ Any

Message contains ▼

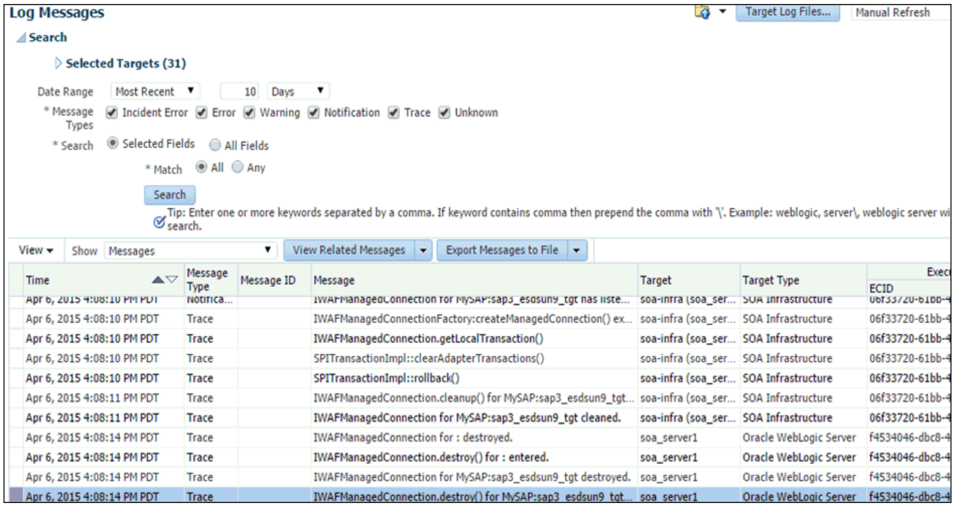
Module contains ▼ oracle.soa.adapter.iwaf.outbound

Tip: Enter one or more keywords separated by a comma. If keyword contains comma then prepend the comma with '\'. Example: weblogic, server\, weblogic server will search.

Time	Message Type	Message ID	Message	Target	Target Type	Executio ECID
Apr 6, 2015 4:08:10 PM PDT	Error		Throwing IllegalArgumentException: Configuration 'sap3_esdsun...	soa-infra (soa_ser...	SOA Infrastructure	06f33720-61bb-47...

The messages from the specified logger are displayed in the table below the search criteria, as shown in Figure 9–9.

Figure 9–9 Logger Messages



Log Messages

Selected Targets (31)

Date Range Most Recent ▼ 10 Days ▼

* Message Types ☑ Incident Error ☑ Error ☑ Warning ☑ Notification ☑ Trace ☑ Unknown

* Search ● Selected Fields ○ All Fields

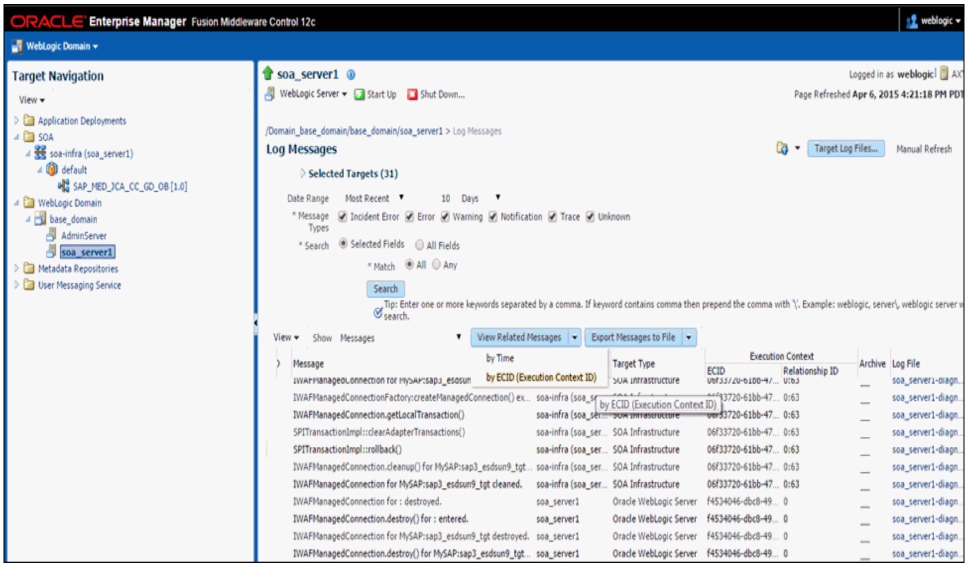
* Match ● All ○ Any

Tip: Enter one or more keywords separated by a comma. If keyword contains comma then prepend the comma with '\'. Example: weblogic, server\, weblogic server wi search.

Time	Message Type	Message ID	Message	Target	Target Type	Exec ECID
Apr 6, 2015 4:08:10 PM PDT	Notifica...		IWAFManagedConnection for MySAP:sap3_esdsun9_tgt has liste...	soa-infra (soa_ser...	SOA Infrastructure	06f33720-61bb-4
Apr 6, 2015 4:08:10 PM PDT	Trace		IWAFManagedConnectionFactory:createManagedConnection() ex...	soa-infra (soa_ser...	SOA Infrastructure	06f33720-61bb-4
Apr 6, 2015 4:08:10 PM PDT	Trace		IWAFManagedConnection.getLocalTransaction()	soa-infra (soa_ser...	SOA Infrastructure	06f33720-61bb-4
Apr 6, 2015 4:08:10 PM PDT	Trace		SPITransactionImpl::clearAdapterTransactions()	soa-infra (soa_ser...	SOA Infrastructure	06f33720-61bb-4
Apr 6, 2015 4:08:10 PM PDT	Trace		SPITransactionImpl::rollback()	soa-infra (soa_ser...	SOA Infrastructure	06f33720-61bb-4
Apr 6, 2015 4:08:11 PM PDT	Trace		IWAFManagedConnection.cleanup() for MySAP:sap3_esdsun9_tgt...	soa-infra (soa_ser...	SOA Infrastructure	06f33720-61bb-4
Apr 6, 2015 4:08:11 PM PDT	Trace		IWAFManagedConnection for MySAP:sap3_esdsun9_tgt cleaned.	soa-infra (soa_ser...	SOA Infrastructure	06f33720-61bb-4
Apr 6, 2015 4:08:14 PM PDT	Trace		IWAFManagedConnection for : destroyed.	soa_server1	Oracle WebLogic Server	f4534046-dbc8-4
Apr 6, 2015 4:08:14 PM PDT	Trace		IWAFManagedConnection.destroy() for : entered.	soa_server1	Oracle WebLogic Server	f4534046-dbc8-4
Apr 6, 2015 4:08:14 PM PDT	Trace		IWAFManagedConnection for MySAP:sap3_esdsun9_tgt destroyed.	soa_server1	Oracle WebLogic Server	f4534046-dbc8-4
Apr 6, 2015 4:08:14 PM PDT	Trace		IWAFManagedConnection.destroy() for MySAP:sap3_esdsun9_tgt...	soa_server1	Oracle WebLogic Server	f4534046-dbc8-4

14. Select any row in the table. To get identical details, click the **View Related Messages** drop-down list, and select *ECID* (execution Context ID) as shown in Figure 9–10.

Figure 9–10 ECID Option



Page Refreshed Apr 6, 2015 4:21:18 PM PDT

/Domain_base_domain/base_domain/soa_server1 > Log Messages

Log Messages

Selected Targets (31)

Date Range: Most Recent ▼ 10 Days ▼

* Message Types: ☑ Incident Error ☑ Error ☑ Warning ☑ Notification ☑ Trace ☑ Unknown

* Search: ● Selected Fields ○ All Fields

* Match: ● All ○ Any

Search

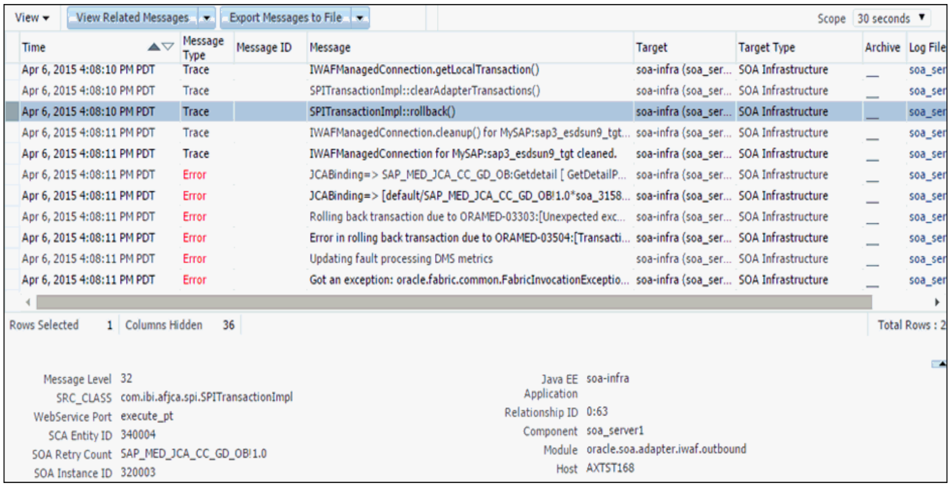
Tip: Enter one or more keywords separated by a comma. If keyword contains comma then prepend the comma with '\'. Example: weblogic, server\, weblogic server w search.

View ▾ Show Messages ▼

Message		Target Type	Execution Context		Archive	Log File
			ECID	Relationship ID		
IWAFManagedConnection for MySAP:sap3_esdsun		SOA Infrastructure	06f33720-61bb-47...	0:63	—	soa_server1-diagn...
IWAFManagedConnectionFactory:createManagedConnection() ex...	soa-infra (soa_se		...f33720-61bb-47...	0:63	—	soa_server1-diagn...
IWAFManagedConnection.getLocalTransaction()	soa-infra (soa_ser...	SOA Infrastructure	06f33720-61bb-47...	0:63	—	soa_server1-diagn...
SPITransactionImpl::clearAdapterTransactions()	soa-infra (soa_ser...	SOA Infrastructure	06f33720-61bb-47...	0:63	—	soa_server1-diagn...
SPITransactionImpl::rollback()	soa-infra (soa_ser...	SOA Infrastructure	06f33720-61bb-47...	0:63	—	soa_server1-diagn...
IWAFManagedConnection.cleanup() for MySAP:sap3_esdsun9_tgt...	soa-infra (soa_ser...	SOA Infrastructure	06f33720-61bb-47...	0:63	—	soa_server1-diagn...
IWAFManagedConnection for MySAP:sap3_esdsun9_tgt cleaned.	soa-infra (soa_ser...	SOA Infrastructure	06f33720-61bb-47...	0:63	—	soa_server1-diagn...
IWAFManagedConnection for : destroyed.	soa_server1	Oracle WebLogic Server	f4534046-dbc8-49...	0	—	soa_server1-diagn...
IWAFManagedConnection.destroy() for : entered.	soa_server1	Oracle WebLogic Server	f4534046-dbc8-49...	0	—	soa_server1-diagn...
IWAFManagedConnection for MySAP:sap3_esdsun9_tgt destroyed.	soa_server1	Oracle WebLogic Server	f4534046-dbc8-49...	0	—	soa_server1-diagn...
IWAFManagedConnection.destroy() for MySAP:sap3_esdsun9_tgt...	soa_server1	Oracle WebLogic Server	f4534046-dbc8-49...	0	—	soa_server1-diagn...

Details are displayed, as shown in Figure 9–11.

Figure 9–11 Message Details



Scope 30 seconds ▼

Time	Message Type	Message ID	Message	Target	Target Type	Archive	Log File
Apr 6, 2015 4:08:10 PM PDT	Trace		IWAFManagedConnection.getLocalTransaction()	soa-infra (soa_ser...	SOA Infrastructure	—	soa_ser
Apr 6, 2015 4:08:10 PM PDT	Trace		SPITransactionImpl::clearAdapterTransactions()	soa-infra (soa_ser...	SOA Infrastructure	—	soa_ser
Apr 6, 2015 4:08:10 PM PDT	Trace		SPITransactionImpl::rollback()	soa-infra (soa_ser...	SOA Infrastructure	—	soa_ser
Apr 6, 2015 4:08:11 PM PDT	Trace		IWAFManagedConnection.cleanup() for MySAP:sap3_esdsun9_tgt...	soa-infra (soa_ser...	SOA Infrastructure	—	soa_ser
Apr 6, 2015 4:08:11 PM PDT	Trace		IWAFManagedConnection for MySAP:sap3_esdsun9_tgt cleaned.	soa-infra (soa_ser...	SOA Infrastructure	—	soa_ser
Apr 6, 2015 4:08:11 PM PDT	Error		JCABinding=> SAP_MED_JCA_CC_GD_OB:Getdetail [GetDetailP...	soa-infra (soa_ser...	SOA Infrastructure	—	soa_ser
Apr 6, 2015 4:08:11 PM PDT	Error		JCABinding=> [default/SAP_MED_JCA_CC_GD_OB!1.0*soa_3158...	soa-infra (soa_ser...	SOA Infrastructure	—	soa_ser
Apr 6, 2015 4:08:11 PM PDT	Error		Rolling back transaction due to ORAMED-03303:[Unexpected exc...	soa-infra (soa_ser...	SOA Infrastructure	—	soa_ser
Apr 6, 2015 4:08:11 PM PDT	Error		Error in rolling back transaction due to ORAMED-03504:[Transacti...	soa-infra (soa_ser...	SOA Infrastructure	—	soa_ser
Apr 6, 2015 4:08:11 PM PDT	Error		Updating fault processing DMS metrics	soa-infra (soa_ser...	SOA Infrastructure	—	soa_ser
Apr 6, 2015 4:08:11 PM PDT	Error		Got an exception: oracle.fabric.common.FabricInvocationExceptio...	soa-infra (soa_ser...	SOA Infrastructure	—	soa_ser

Rows Selected 1 Columns Hidden 36 Total Rows : 2

Message Level	32	Java EE Application	soa-infra
SRC_CLASS	com.ibi.afjca.spi.SPITransactionImpl	Relationship ID	0:63
WebService Port	execute_pt	Component	soa_server1
SCA Entity ID	340004	Module	oracle.soa.adapter.iwaf.outbound
SOA Retry Count	SAP_MED_JCA_CC_GD_OB!1.0	Host	AXTST168
SOA Instance ID	320003		

9.1.2 Configuring Log File Management for Business Services Engine (BSE)

Similar to J2CA for BSE, the Log file management is governed by the Loggers defined in:

```
<ORACLE_HOME>\user_projects\domains\base_
domain\config\fmwconfig\servers\${server-name}\logging.xml
```

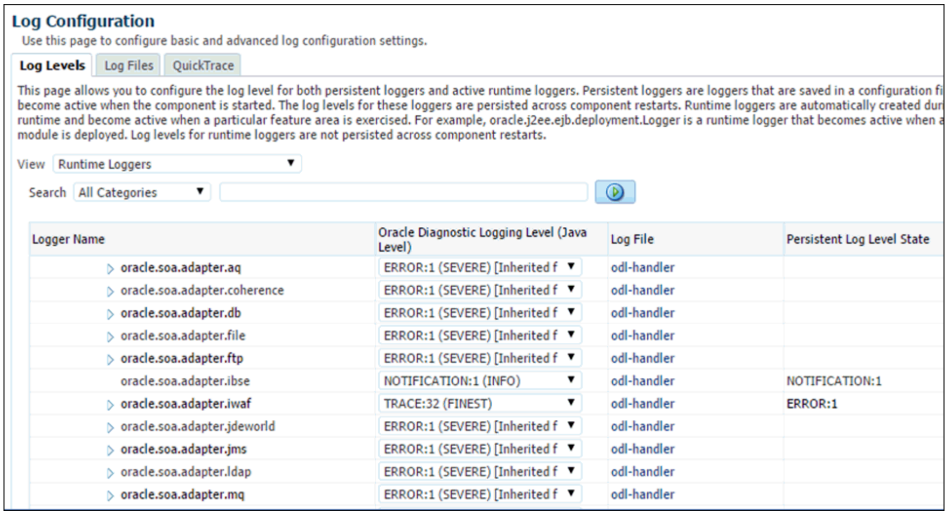
The following syntax sets the logging level of all the loggers under oracle.soa.adapter.ibse to NOTIFICATION:1 (INFO) which is the default setting level by oracle.

```
<logger name='oracle.soa.adapter.ibse' level='NOTIFICATION:1'
useParentHandlers='true'/>
```

The logging level of all the loggers can also be configured from the em console with the following steps:

1. Repeat steps 1 through 5, as described in Section 9.1.1, "Configuring Log File Management for the J2CA Connector Application".
2. Under the **Log Levels** tab, expand the Oracle root logger until the oracle.soa.adapter.ibse Logger name is visible, as shown in Figure 9–12.

Figure 9–12 Log Levels Tab



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 fi become active when the component is started. The log levels for these loggers are persisted across component restarts. Runtime loggers are automatically created dur runtime and become active when a particular feature area is exercised. For example, oracle.j2ee.ejb.deployment.Logger is a runtime logger that becomes active when a module is deployed. Log levels for runtime loggers are not persisted across component restarts.

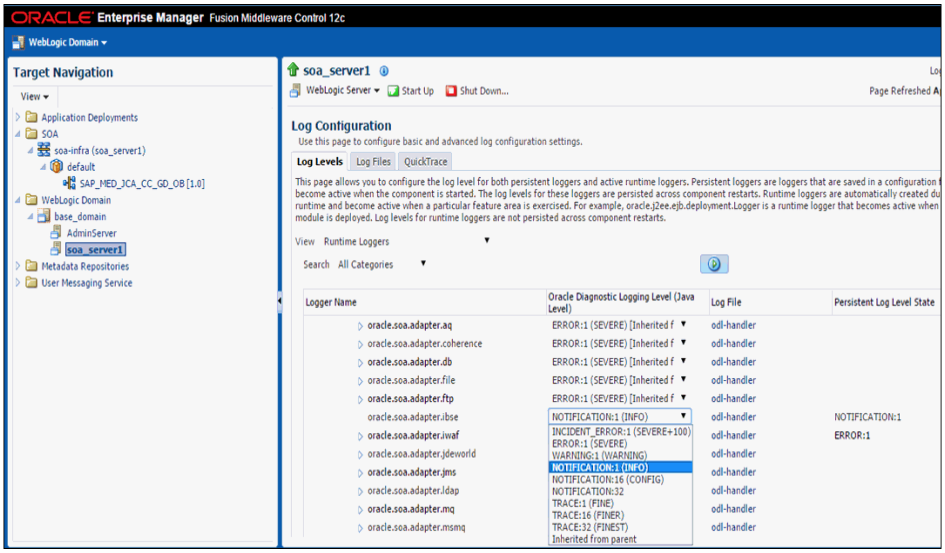
View Runtime Loggers

Search All Categories

Logger Name	Oracle Diagnostic Logging Level (Java Level)	Log File	Persistent Log Level State
oracle.soa.adapter.aq	ERROR:1 (SEVERE) [Inherited f	odl-handler	
oracle.soa.adapter.coherence	ERROR:1 (SEVERE) [Inherited f	odl-handler	
oracle.soa.adapter.db	ERROR:1 (SEVERE) [Inherited f	odl-handler	
oracle.soa.adapter.file	ERROR:1 (SEVERE) [Inherited f	odl-handler	
oracle.soa.adapter.ftp	ERROR:1 (SEVERE) [Inherited f	odl-handler	
oracle.soa.adapter.ibse	NOTIFICATION:1 (INFO)	odl-handler	NOTIFICATION:1
oracle.soa.adapter.iwaf	TRACE:32 (FINEST)	odl-handler	ERROR:1
oracle.soa.adapter.jdeworld	ERROR:1 (SEVERE) [Inherited f	odl-handler	
oracle.soa.adapter.jms	ERROR:1 (SEVERE) [Inherited f	odl-handler	
oracle.soa.adapter.ldap	ERROR:1 (SEVERE) [Inherited f	odl-handler	
oracle.soa.adapter.mq	ERROR:1 (SEVERE) [Inherited f	odl-handler	

3. In the Oracle Diagnostic Logging Level (Java Level) column, select the required log level from the oracle.soa.adapter.ibse drop-down list, as shown in Figure 9–13.

Figure 9–13 Oracle Diagnostic Logging Level Column



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 f become active when the component is started. The log levels for these loggers are persisted across component restarts. Runtime loggers are automatically created du runtime and become active when a particular feature area is exercised. For example, oracle.j2ee.ejb.deployment.Logger is a runtime logger that becomes active when module is deployed. Log levels for runtime loggers are not persisted across component restarts.

View Runtime Loggers

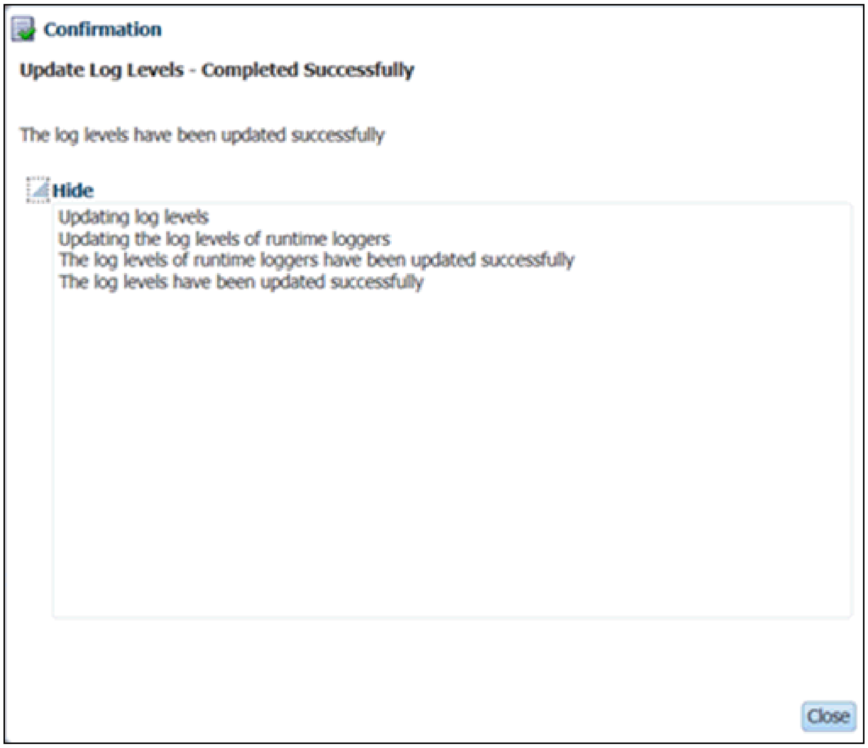
Search All Categories

Logger Name	Oracle Diagnostic Logging Level (Java Level)	Log File	Persistent Log Level State
oracle.soa.adapter.aq	ERROR:1 (SEVERE) [Inherited f	odl-handler	
oracle.soa.adapter.coherence	ERROR:1 (SEVERE) [Inherited f	odl-handler	
oracle.soa.adapter.db	ERROR:1 (SEVERE) [Inherited f	odl-handler	
oracle.soa.adapter.file	ERROR:1 (SEVERE) [Inherited f	odl-handler	
oracle.soa.adapter.ftp	ERROR:1 (SEVERE) [Inherited f	odl-handler	
oracle.soa.adapter.ibse	NOTIFICATION:1 (INFO)	odl-handler	NOTIFICATION:1
oracle.soa.adapter.iwaf		odl-handler	ERROR:1
oracle.soa.adapter.jdeworld		odl-handler	
oracle.soa.adapter.jms		odl-handler	
oracle.soa.adapter.ldap		odl-handler	
oracle.soa.adapter.mq		odl-handler	
oracle.soa.adapter.msmq		odl-handler	

4. Click **Apply**.

   A confirmation message appears, indicating that the update was completed successfully, as shown in Figure 9–14.

Figure 9–14 Confirmation Message



Confirmation

Update Log Levels - Completed Successfully

The log levels have been updated successfully

Updating log levels
Updating the log levels of runtime loggers
The log levels of runtime loggers have been updated successfully
The log levels have been updated successfully

5. Click **Close**.

   The following table shows how the Log Level property is updated in the *{server-name}*-diagnostic.log based on the corresponding Log Level property settings in the em console.

Table 9–2 Log Level Property Settings

Log Level Set in Em Console	Log Level Updated in {server-name}-diagnostic.log
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

For verification of logs, see steps 11 to 18 of the J2CA Logging feature, found in Section 9.1.1, "Configuring Log File Management for the J2CA Connector Application".

9.2 Configuring the Diagnosibility Feature

This section describes how to configure and use the Diagnosibility feature for the Oracle Fusion Middleware Application Adapters for Oracle WebLogic Server. It contains the following topic:

- Section 9.2.1, "Supporting Protocols"

The Diagnosibility feature captures the endpoint health status (where available) of the adapters, and provides a corresponding alert to the Oracle Adapter Framework, so it may be displayed in the EM console.

> **Note:** The Diagnosibility feature supports only inbound adapter processes
>
> Make sure that there is an inbound process deployed before moving to the next section

This information can be viewed for the adapters as mentioned in the following steps:

1. Start the Oracle WebLogic Servers and open the Oracle WebLogic Server Enterprise Manager Console in a web browser by entering the following URL:

   ```
   http://host name:port/em
   ```

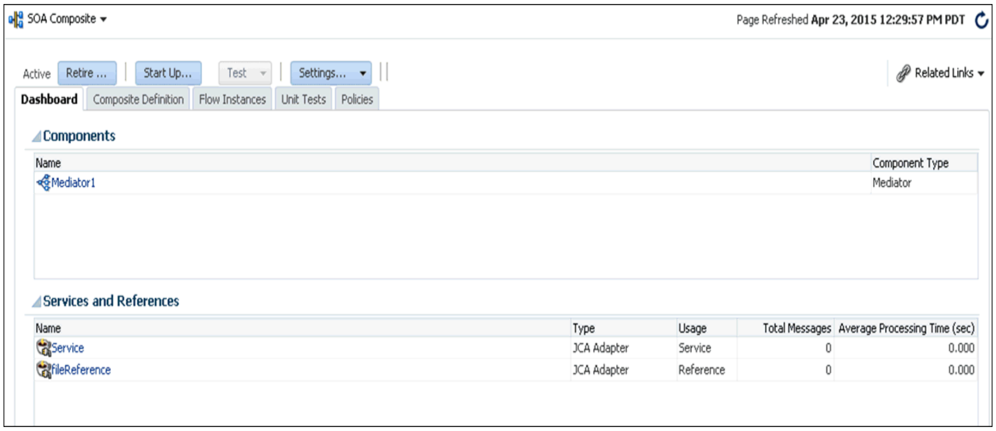
   Where *host name* is the name of the system where Oracle WebLogic Server is running, and *port* is the port for the Oracle WebLogic Server that is running.

2. Log in to the Oracle WebLogic Server Administrative Console using an account that has administrator privileges.

3. On the right pane, expand **SOA**, **soa-Infra (server_name)**, **Default**, and then **Deployed inbound process**.

4. Click the process and then select the service listed in the Services and References Section in the right pane, as shown in Figure 9–15.

Figure 9–15 Services and References Section



Page Refreshed Apr 23, 2015 12:29:57 PM PDT

Components

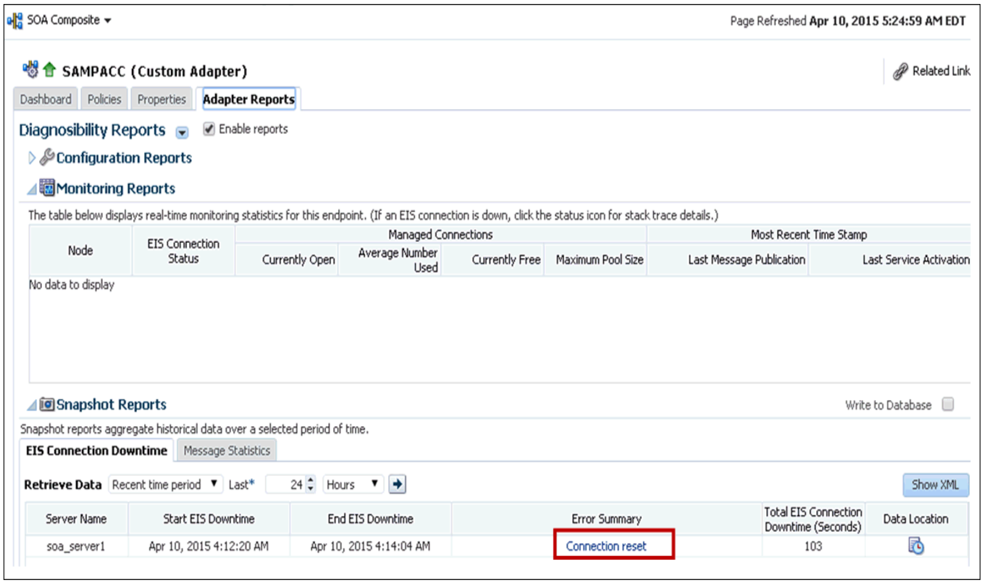
Name	Component Type
Mediator1	Mediator

Services and References

Name	Type	Usage	Total Messages	Average Processing Time (sec)
Service	JCA Adapter	Service	0	0.000
fileReference	JCA Adapter	Reference	0	0.000

5. In the Adapter Reports tab, select the **Enable reports** check box, and then expand **Snapshot Reports** to view the details of the EIS downtime. Clicking on the Error Summary will show the stack trace, as shown in Figure 9–16.

Figure 9–16 Stack Trace



The table below displays real-time monitoring statistics for this endpoint. (If an EIS connection is down, click the status icon for stack trace details.)

Node	EIS Connection Status	Managed Connections				Most Recent Time Stamp	
		Currently Open	Average Number Used	Currently Free	Maximum Pool Size	Last Message Publication	Last Service Activation

No data to display

Snapshot reports aggregate historical data over a selected period of time.

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 Supporting Protocols

This section describes the supporting protocols for the Oracle Fusion Middleware Application Adapter for Siebel, and consists of the following topic:

- Section 9.2.1.1, "Oracle Fusion Middleware Application Adapter for Siebel Endpoint Status"

9.2.1.1 Oracle Fusion Middleware Application Adapter for Siebel Endpoint Status

The Oracle Fusion Middleware Application Adapter for Siebel (inbound) can use the following protocols:

- HTTP
- MQ
- File (Not Supported)

The HTTP listening protocol adapters listen on a socket. As a result, by their nature, they cannot determine whether there is anything live on the other side until they receive something. Even when receiving a request, it is impossible to determine with certainty where the request originated.

For these protocols, the EIS is regarded and used when a request is being received. EIS determines if the communication error happens while the request is being received. However, this is a very unrefined and rudimentary determination.

However, the MQ inbound listener adapter polls for MQ events, so a communication failure (or successful communication) will be detected while polling occurs using the MQ API.

9.3 Configuring the SOA Debugging Feature

This section describes how to configure and use the SOA Debugging feature for the Oracle Application Adapter for Siebel. It contains the following topics:

- Section 9.3.1, "Guidelines for Using the SOA Debugger"

- Section 9.3.2, "Prerequisite"
- Section 9.3.3, "Debugging a BPEL Process in Oracle JDeveloper"
- Section 9.3.4, "Debugging an OSB Process in Oracle JDeveloper"

> **Note:** The SOA Debugging feature is currently supported only for J2CA configurations and it is not applicable for BSE configurations.
>
> For SOA, this feature is explained using a BPEL process. The same is applicable for Mediator and BPM processes.

9.3.1 Guidelines for Using the SOA Debugger

This section describes guidelines for using the SOA Debugger.

1. Only one client at a time can connect to the SOA Debugger.
2. Adapter endpoint errors are not displayed in the SOA Debugger in Oracle JDeveloper. These errors are logged in the log file.

> **Note:** The SOA Debugger is currently available for BPEL, Mediator, BPM, and OSB processes with Development mode only.

9.3.2 Prerequisite

Ensure that the *IntegratedWebLogicServer* domain and a BPEL process are already created in Oracle JDeveloper.

9.3.3 Debugging a BPEL Process in Oracle JDeveloper

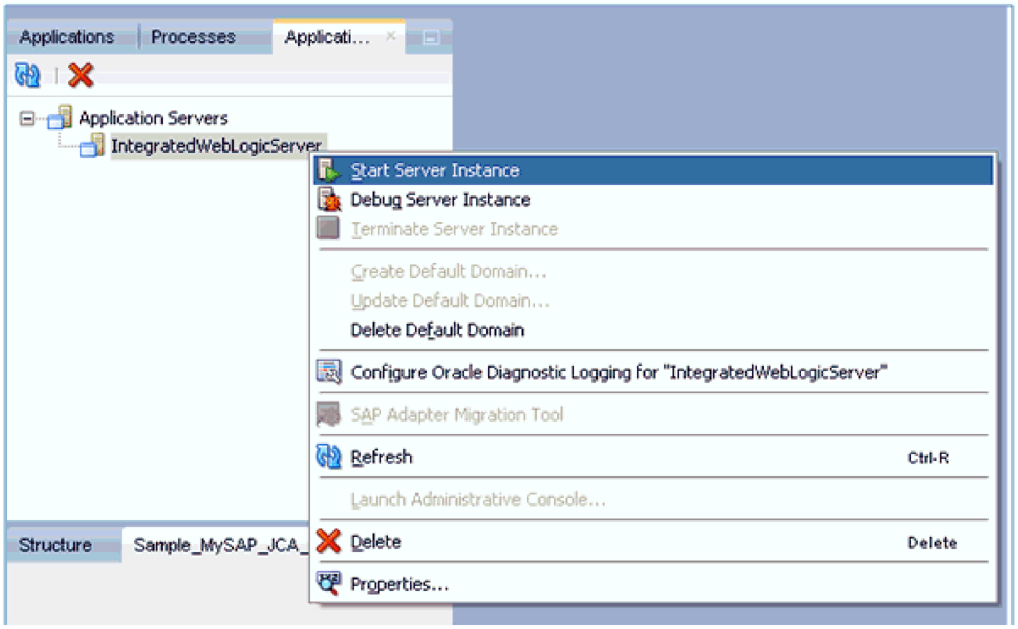
This section describes how to debug a BPEL process in Oracle JDeveloper. It contains the following topics:

- Section 9.3.3.1, "Debugging an Outbound BPEL Process in Oracle JDeveloper"
- Section 9.3.3.2, "Debugging an Inbound BPEL Process in Oracle JDeveloper"

9.3.3.1 Debugging an Outbound BPEL Process in Oracle JDeveloper

1. Open Oracle JDeveloper.
2. Start the *IntegratedWeblogicServer* domain.
   a. Click the **Application Servers** tab in the left pane.
   b. Under the Application Servers node, right-click **IntegratedWeblogicServer** and select **Start Server Instance** from the context menu, as shown in Figure 9–17.

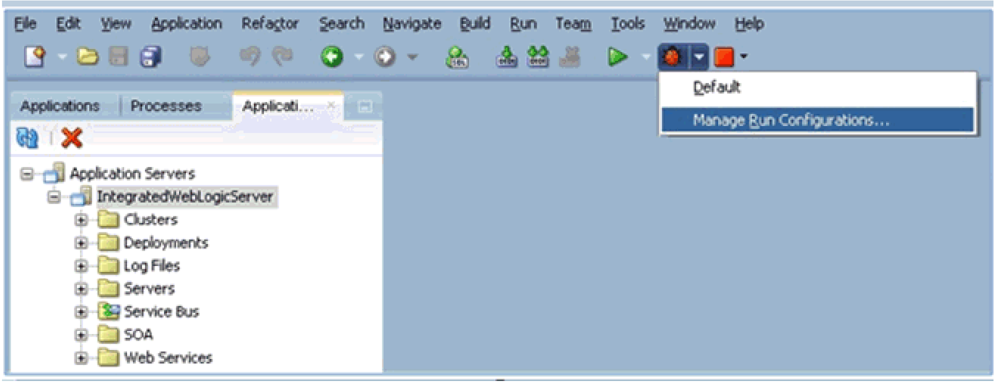
Figure 9–17 Start Server Instance



3. Set the Debugging environment.

   a. Click the down arrow next to the Debug icon and select **Manage Run Configurations** from the context menu, as shown in Figure 9–18.

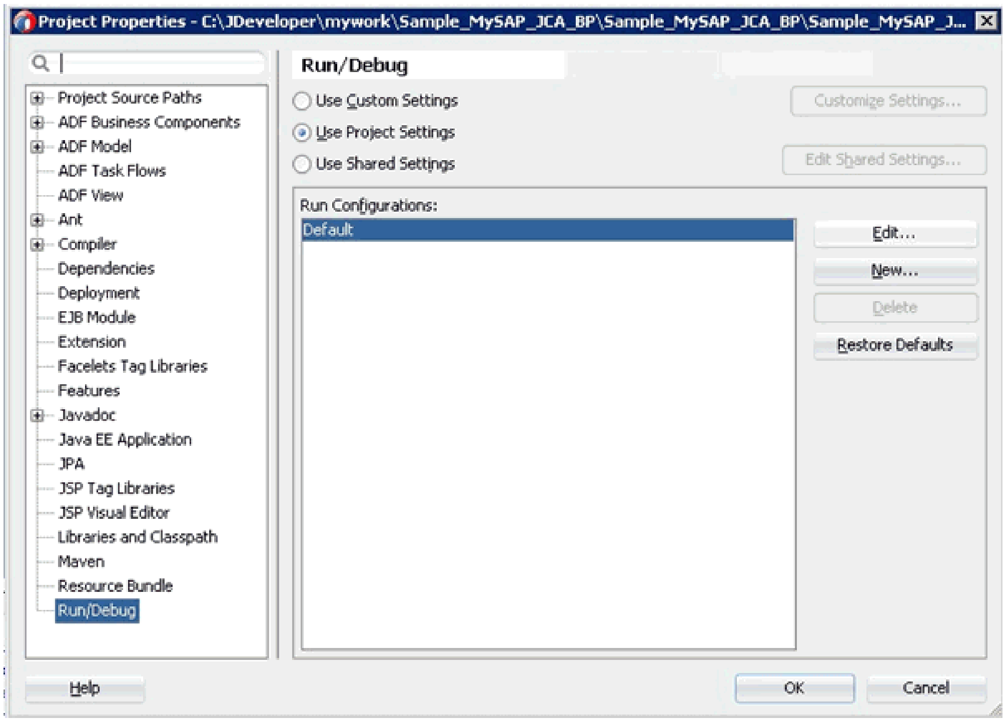
Figure 9–18 Manage Run Configurations



   b. Or, right-click the project and select **Project Properties**.

      The Project Properties dialog is displayed, as shown in Figure 9–19.

Figure 9–19 Project Properties Dialog



Project Properties - C:\JDeveloper\mywork\Sample_MySAP_JCA_BP\Sample_MySAP_JCA_BP\Sample_MySAP_J...

Run/Debug

○ Use Custom Settings

● Use Project Settings

○ Use Shared Settings

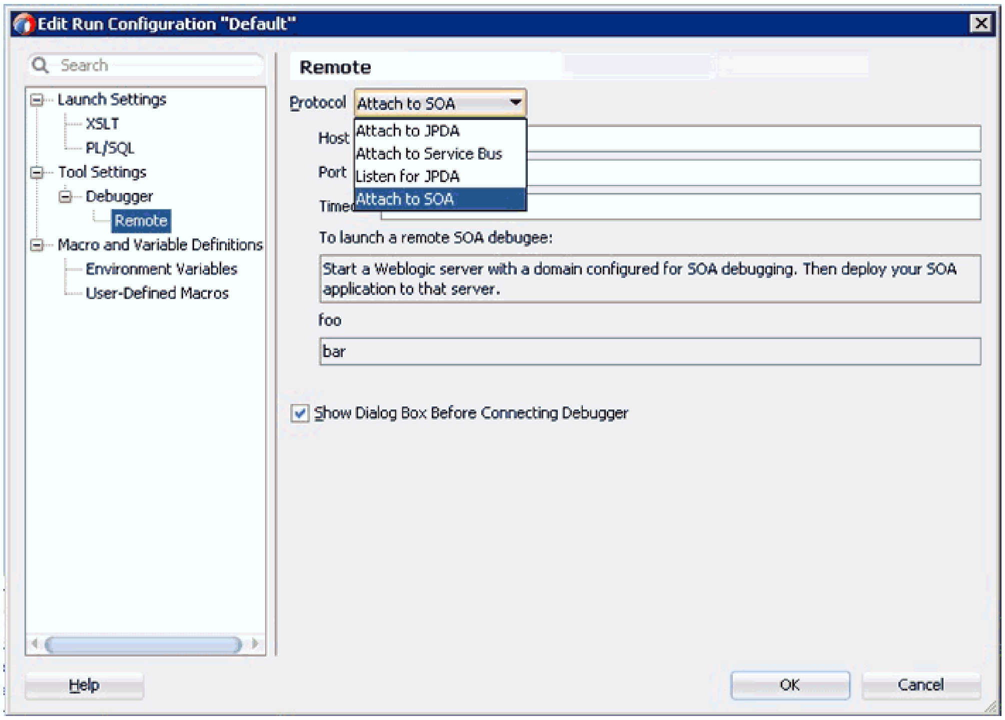
Run Configurations:

Default

c. Select **Run/Debug** in the left pane and then click **Edit** in the Run/Debug pane (Run Configurations area).

The Edit Run Configuration dialog is displayed, as shown in Figure 9–20.

Figure 9–20 Edit Run Configuration Dialog



Edit Run Configuration "Default"

Remote

Protocol: Attach to SOA

Host

Port

Timeo

To launch a remote SOA debugee:

Start a Weblogic server with a domain configured for SOA debugging. Then deploy your SOA application to that server.

foo

bar

☑ Show Dialog Box Before Connecting Debugger

   d. Expand **Tool Settings**, **Debugger** in the left pane, and then click **Remote**.

   e. From the Protocol list, select **Attach to SOA**.

   f. Leave the default values for the Host and Port.

   g. Click **OK**.

4. Deploy the project.

   a. Right-click the project and select **Deploy**.

      The Deployment Action dialog is displayed.

      Select the application to deploy and click **Next**.

      The Deploy Configuration dialog is displayed.

   b. Click **Next**.

      The Application Servers dialog is displayed.

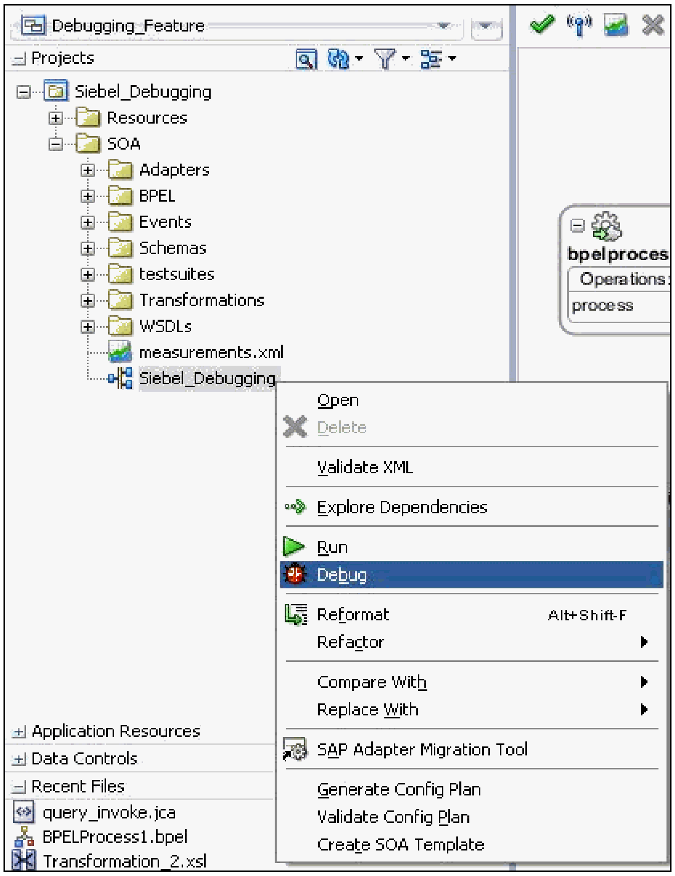
   c. Select **IntegratedWebLogicServer** and then click **Finish** to complete the deployment.

   d. Ensure that the project deployment has completed without any errors or issues before proceeding to the next step.

5. Connect a BPEL process to the SOA Debugger.

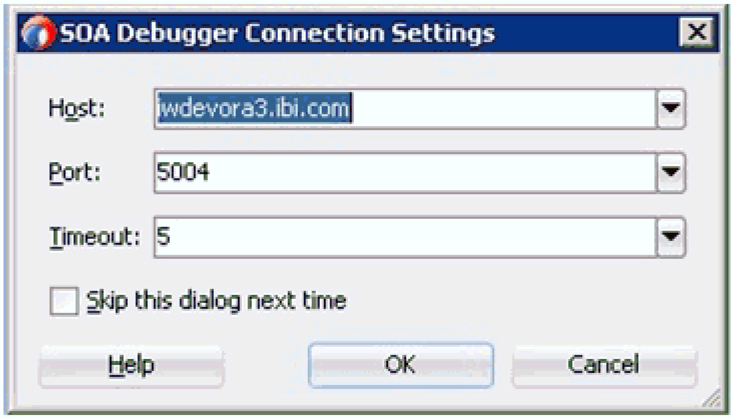
   a. In the Applications tab on the left pane, right-click a composite XML or project for an existing BPEL process and then select **Debug** from the context menu, as shown in Figure 9–21.

Figure 9–21 Select BPEL Process to Debug



The SOA Debugger Connection Settings dialog is displayed, as shown in Figure 9–22.

Figure 9–22 SOA Debugger Connection Settings Dialog



Host: iwdevora3.ibi.com

Port: 5004

Timeout: 5

☐ Skip this dialog next time

b. Click **OK**.

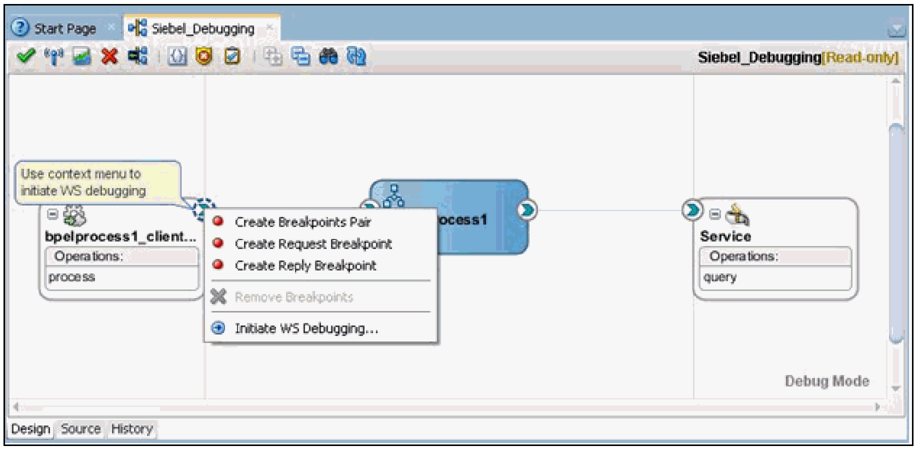
Once the BPEL process is connected to the SOA Debugger, the following messages will be displayed in the Debugging log:

```
Debugger attempting to connect to remote process at iwdevora3.ibi.com 5004.
Debugger connected to remote process at iwdevora3.ibi.com 5004.
Debugger process virtual machine is SOA Debugger
```

6. Set the Breakpoints and initiate debugging.

   a. Right-click on the components and select the Breakpoint type to set, as shown in Figure 9–23.

Figure 9–23 Selecting Breakpoints



Create Breakpoints Pair - Set this Breakpoint type for a request-reply (outbound-inbound) interaction. This is useful for scenarios in which both the request and reply are important.

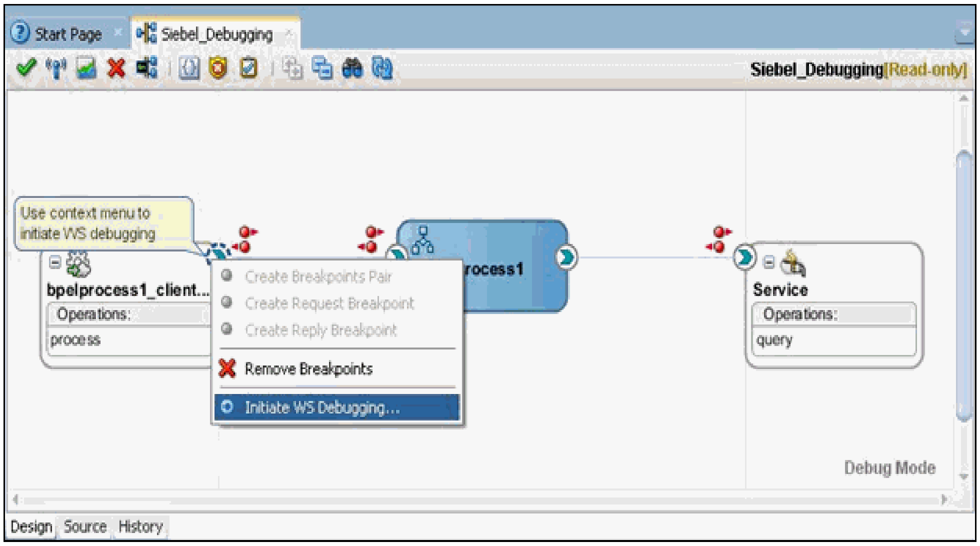
Create Request Breakpoint - Set this Breakpoint type for a request (outbound) interaction. This is useful for scenarios in which only the request is important.

Create Reply Breakpoint - Set this Breakpoint type for a reply (inbound) interaction. This is useful for scenarios in which only the reply is important.

Initiate WS Debugging - Set this Breakpoint type to initiate a debugging session. For example, the debugging session encompasses an initiating SOAP request from a web service to a BPEL process to an adapter reference binding component.

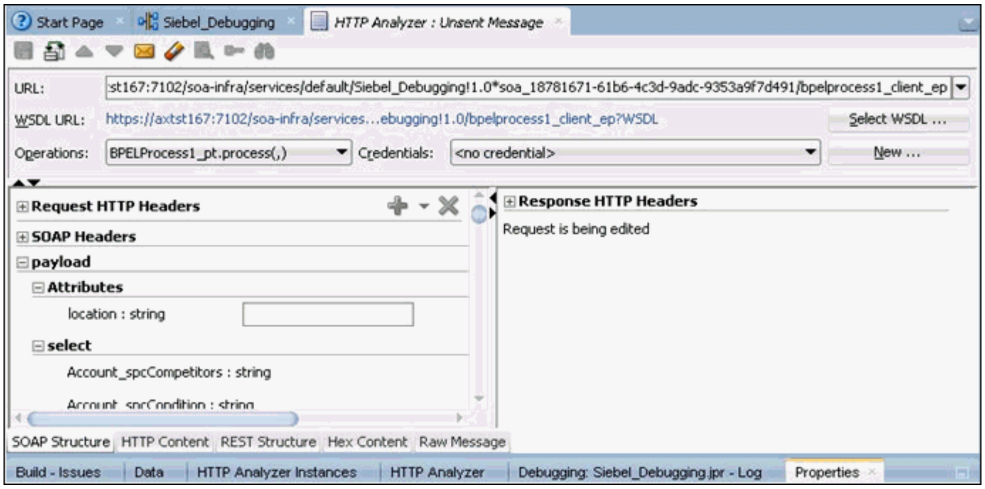
   b. Once the Breakpoints are set, right-click the right handle and select **Initiate WS Debugging**, as shown in Figure 9–24.

Figure 9–24 Initiate WS Debugging



The HTTP Analyzer dialog is displayed, as shown in Figure 9–25.

Figure 9–25 HTTP Analyzer Dialog



URL: st167:7102/soa-infra/services/default/Siebel_Debugging!1.0*soa_18781671-61b6-4c3d-9adc-9353a9f7d491/bpelprocess1_client_ep

WSDL URL: https://axtst167:7102/soa-infra/services...ebugging!1.0/bpelprocess1_client_ep?WSDL

Operations: BPELProcess1_pt.process(,) Credentials: <no credential>

location : string

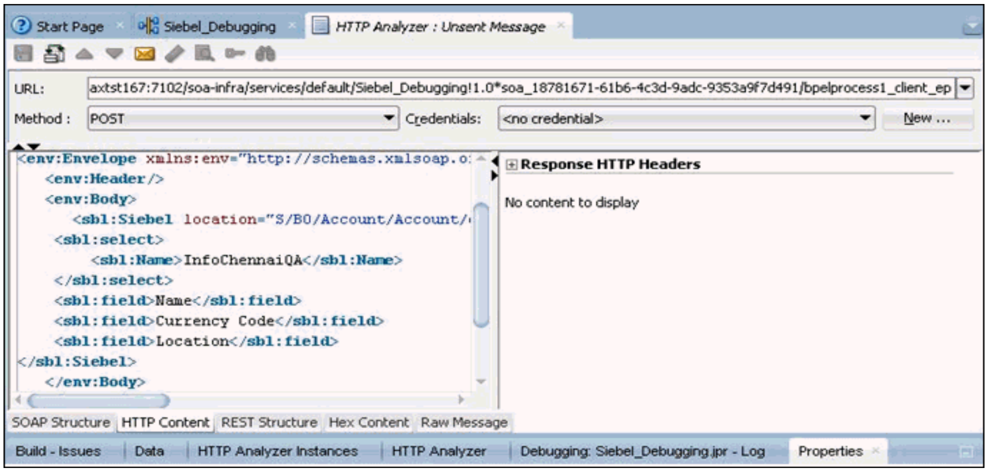
Account_spcCompetitors : string

Account_spcCondition : string

Request is being edited

c. Select HTTP content from the below tab. Now, copy and paste the payload into the body, as shown in Figure 9–26.

Figure 9–26 Select HTTP Content



URL: axtst167:7102/soa-infra/services/default/Siebel_Debugging!1.0*soa_18781671-61b6-4c3d-9adc-9353a9f7d491/bpelprocess1_client_ep

Method : POST

Credentials: <no credential>

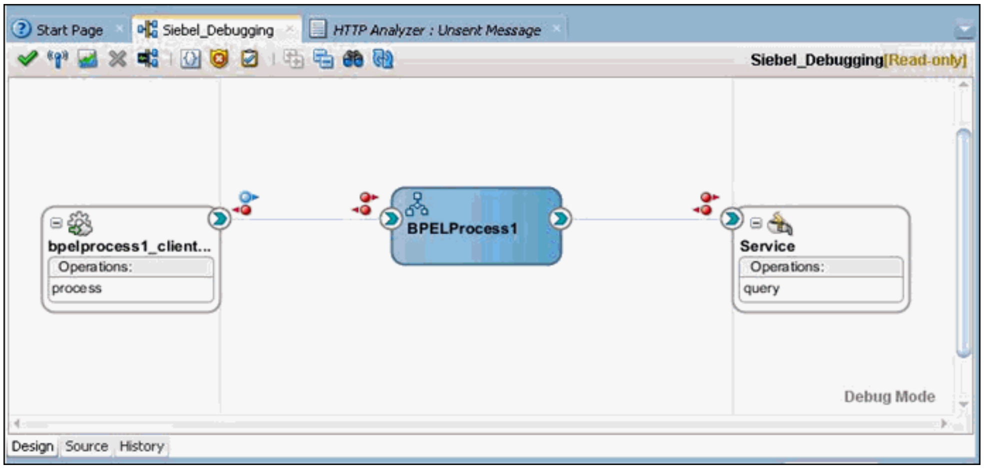
```
<env:Envelope xmlns:env="http://schemas.xmlsoap.o:
   <env:Header/>
   <env:Body>
      <sbl:Siebel location="S/BO/Account/Account/
    <sbl:select>
        <sbl:Name>InfoChennaiQA</sbl:Name>
    </sbl:select>
    <sbl:field>Name</sbl:field>
    <sbl:field>Currency Code</sbl:field>
    <sbl:field>Location</sbl:field>
</sbl:Siebel>
   </env:Body>
```

No content to display

d. Click **Send Request**.

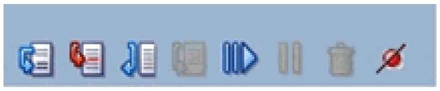
The BPEL Process stops at the designated Breakpoint and blinks in a blue color, as shown in Figure 9–27.

Figure 9–27 BPEL Process Stopped at Breakpoint



e. Use the available Step options to step through the Debugging process, as shown in Figure 9–28.

Figure 9–28 Step Options



Icon	Description
🟥	Ends or detaches from a debugging session.

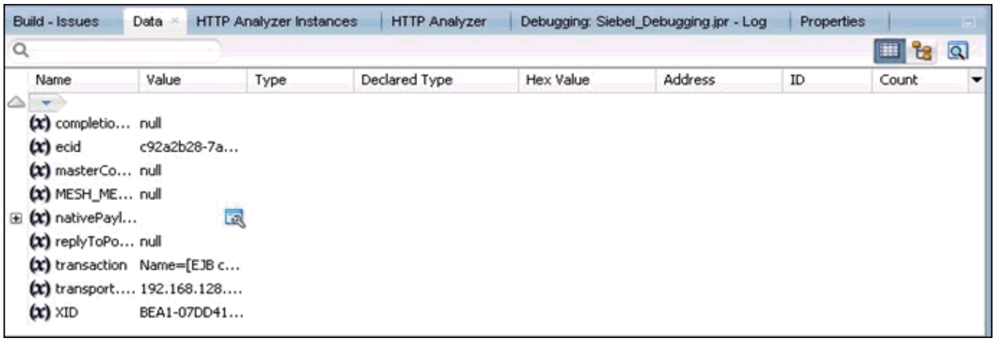
Icon	Description
	Steps over a frame. This places you at the next Breakpoint (for example, the receive activity in the BPEL process on which a Breakpoint was set. If there are no Breakpoints, it steps over all the frames and returns to the first frame. You can also press **F8** to step over a frame.
	Steps into the next valid location. This can be a new frame or the same frame, but in a different location. You can also press **F7** to step into a frame.
	Steps out of a frame. This option is only used to process a BPEL scope or sequence activity. After completion of scope processing, it pauses at the next scope or activity in the process. You can also press **Shift-F7**.
	Resumes a step operation. You can also press **F9** to resume.

7. View the Request payload and header information.

   a. Click **Windows**, select **Debugger**, and then **Data**.

      To view sample header information when a Breakpoint stops at the Request Breakpoint for Oracle Application Adapter for Siebel, see Figure 9–29.

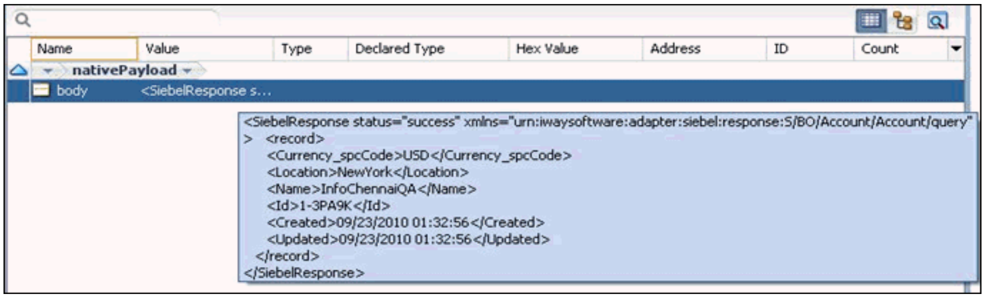
Figure 9–29 Header Information



Name	Value	Type	Declared Type	Hex Value	Address	ID	Count
completio...	null						
ecid	c92a2b28-7a...						
masterCo...	null						
MESH_ME...	null						
nativePayl...							
replyToPo...	null						
transaction	Name=[EJB c...						
transport....	192.168.128....						
XID	BEA1-07DD41...						

A sample response payload for Oracle Application Adapter for Siebel is shown in Figure 9–30.

Figure 9–30 Sample Response Payload



Name	Value	Type	Declared Type	Hex Value	Address	ID	Count
nativePayload							
body	<SiebelResponse s...						

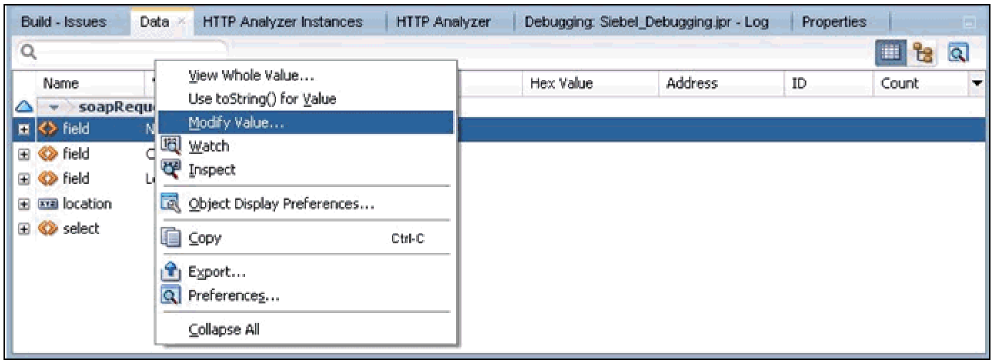
```
<SiebelResponse status="success" xmlns="urn:iwaysoftware:adapter:siebel:response:S/BO/Account/Account/query"
>   <record>
      <Currency_spcCode>USD</Currency_spcCode>
      <Location>NewYork</Location>
      <Name>InfoChennaiQA</Name>
      <Id>1-3PA9K</Id>
      <Created>09/23/2010 01:32:56</Created>
      <Updated>09/23/2010 01:32:56</Updated>
   </record>
</SiebelResponse>
```

> **Note:** The payload display is limited to the screen size as shown in Figure 9–30. However, the Breakpoint at the BPEL process displays the complete payload, allows scrolling and viewing all elements of the payload.

8. Modify the Request payload content.

   a. Expand the SOAP request, select the field to modify, right-click and select **Modify Value** from the context menu, as shown in Figure 9–31.

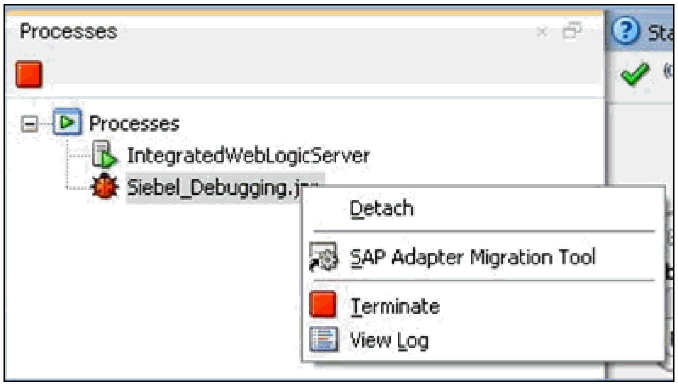
Figure 9–31 Modify Value



9. End or detach the Debugging session.

   a. Click **Window** and then **Processes**. Right click on the process in the Processes tab and select **Detach** or **Terminate**, as shown in Figure 9–32.

Figure 9–32 Detach Debugging Session



   b. Select one of the following options:

   Detach - Removes the SOA Debugger without ending the debugging process.

   Terminate - Ends the debugging process.

9.3.3.2 Debugging an Inbound BPEL Process in Oracle JDeveloper

1. Deploy the Inbound BPEL process.

   a. Right-click the project and select **Deploy**.

   The Deployment Action dialog is displayed.

   b. Select **Deploy to Application Server** and click **Next**.

      The Deploy Configuration dialog is displayed.

   c. Click **Next**.

      The Select Server dialog is displayed.

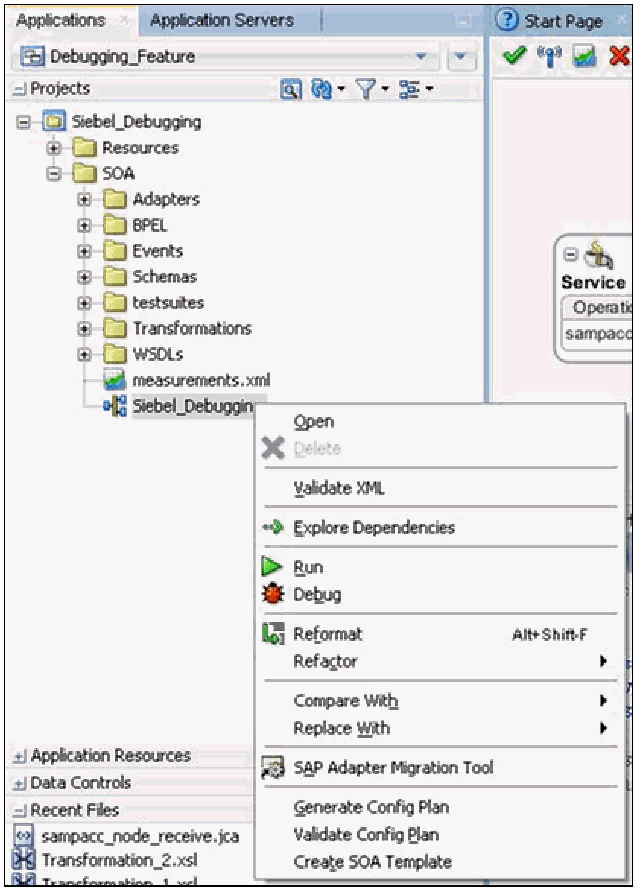
   d. Select the server to deploy and click **Finish**.

   e. Ensure that the project deployment has completed without any errors or issues before proceeding to the next step.

2. Connect a BPEL process to the SOA Debugger.

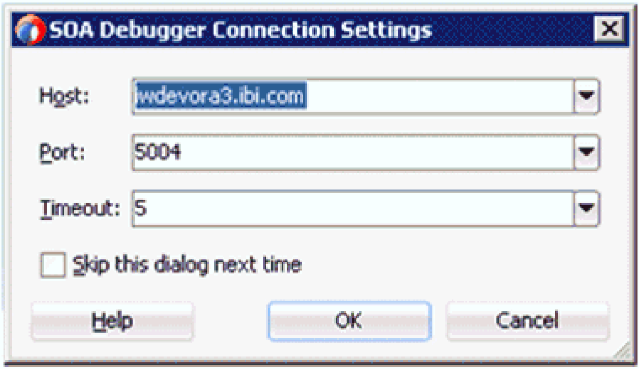
   a. Right-click an inbound composite and select **Debug**, as shown in Figure 9–33.

Figure 9–33 Select BPEL Process to Debug



      The SOA Debugger Connection Settings dialog is displayed, as shown in Figure 9–34.

Figure 9–34 SOA Debugger Connection Settings Dialog



SOA Debugger Connection Settings

Host: iwdevora3.ibi.com

Port: 5004

Timeout: 5

☐ Skip this dialog next time

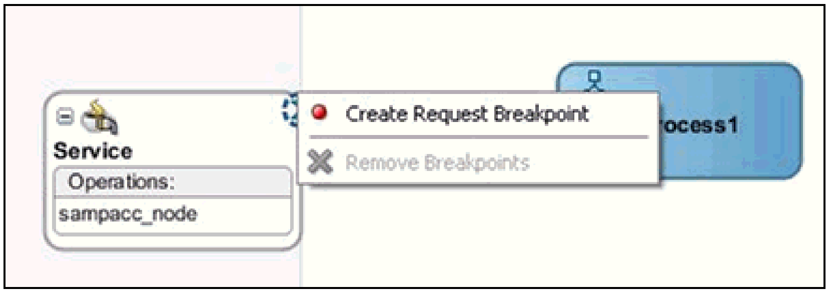
b. Click **OK**.

Once the BPEL process is connected to the SOA Debugger, the following messages will be displayed in the Debugging log:

```
Debugger attempting to connect to remote process at iwdevora3.ibi.com 5004.
Debugger connected to remote process at iwdevora3.ibi.com 5004.
Debugger process virtual machine is SOA Debugger
```

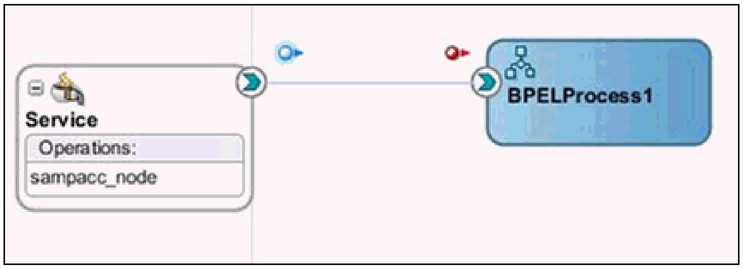
3. Set the Breakpoints, as shown in Figure 9–35.

Figure 9–35 Setting Breakpoints



4. Once the Breakpoints are set and a message is received through the inbound process (for example, by triggering from Siebel), the process stops at the designated Breakpoints, as shown in Figure 9–36.

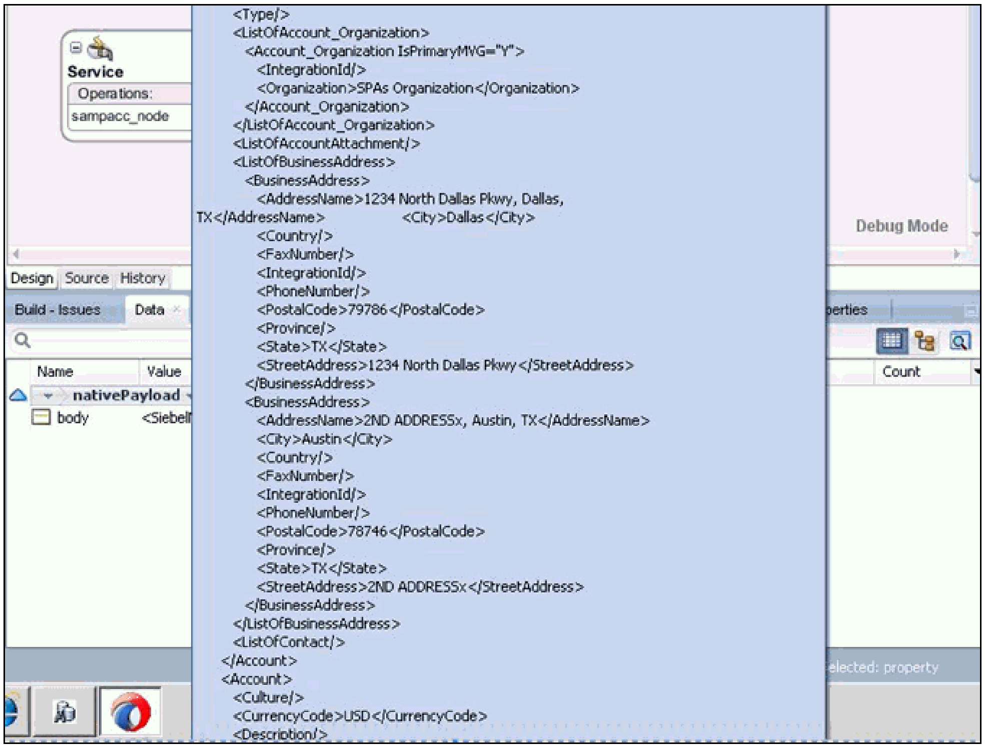
Figure 9–36 Stopping the Process at the Breakpoint



5. View the Request payload and header information.

a. Click **Windows**, select **Debugger**, and then **Data**, as shown in Figure 9–37.

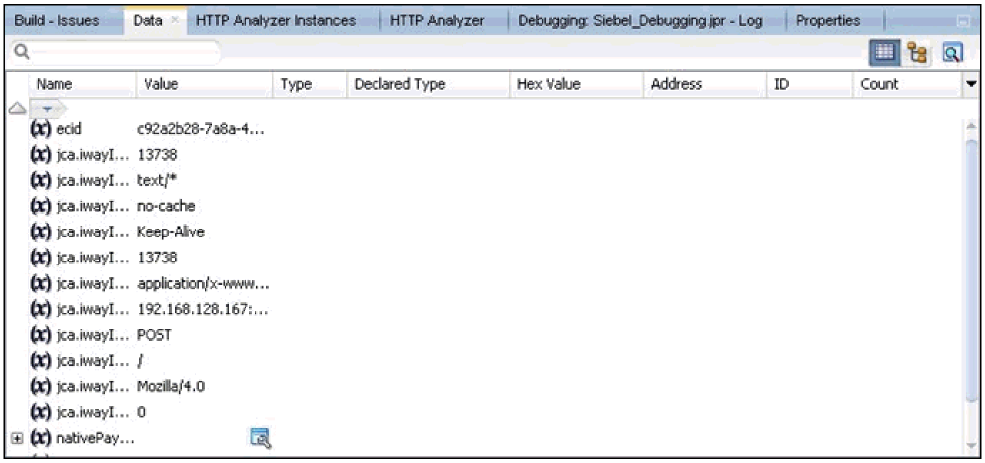
Figure 9–37 Debugger Data



```
    <Type/>
    <ListOfAccount_Organization>
      <Account_Organization IsPrimaryMVG="Y">
        <IntegrationId/>
        <Organization>SPAs Organization</Organization>
      </Account_Organization>
    </ListOfAccount_Organization>
    <ListOfAccountAttachment/>
    <ListOfBusinessAddress>
      <BusinessAddress>
        <AddressName>1234 North Dallas Pkwy, Dallas,
TX</AddressName>              <City>Dallas</City>
        <Country/>
        <FaxNumber/>
        <IntegrationId/>
        <PhoneNumber/>
        <PostalCode>79786</PostalCode>
        <Province/>
        <State>TX</State>
        <StreetAddress>1234 North Dallas Pkwy</StreetAddress>
      </BusinessAddress>
      <BusinessAddress>
        <AddressName>2ND ADDRESSx, Austin, TX</AddressName>
        <City>Austin</City>
        <Country/>
        <FaxNumber/>
        <IntegrationId/>
        <PhoneNumber/>
        <PostalCode>78746</PostalCode>
        <Province/>
        <State>TX</State>
        <StreetAddress>2ND ADDRESSx</StreetAddress>
      </BusinessAddress>
    </ListOfBusinessAddress>
    <ListOfContact/>
  </Account>
  <Account>
    <Culture/>
    <CurrencyCode>USD</CurrencyCode>
    <Description/>
```

b. Header information displayed for Inbound Request Breakpoint (SampleAccount) is shown in Figure 9–38.

Figure 9–38 Inbound Request Breakpoint (SampleAccount)

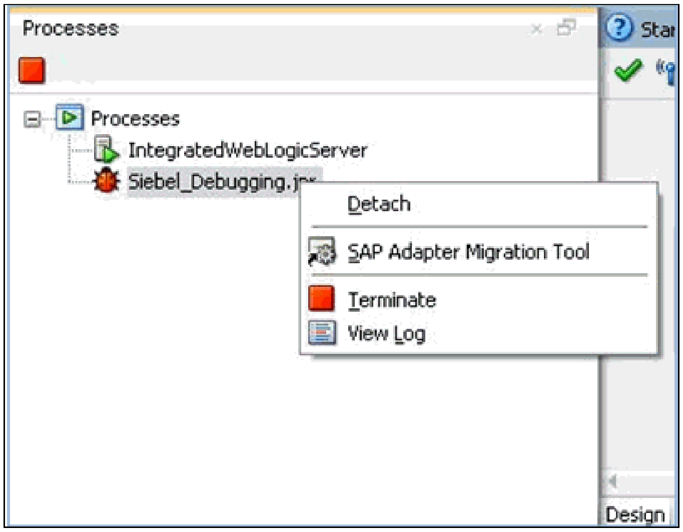


Name	Value	Type	Declared Type	Hex Value	Address	ID	Count
ecid	c92a2b28-7a8a-4...						
jca.iwayI...	13738						
jca.iwayI...	text/*						
jca.iwayI...	no-cache						
jca.iwayI...	Keep-Alive						
jca.iwayI...	13738						
jca.iwayI...	application/x-www...						
jca.iwayI...	192.168.128.167:...						
jca.iwayI...	POST						
jca.iwayI...	/						
jca.iwayI...	Mozilla/4.0						
jca.iwayI...	0						
nativePay...							

6. End or detach the Debugging session.

   a. Click **Window** and then **Processes**. Right click on the process in the Processes tab and select **Detach** or **Terminate**, as shown in Figure 9–39.

Figure 9–39 Detach Debugging Session



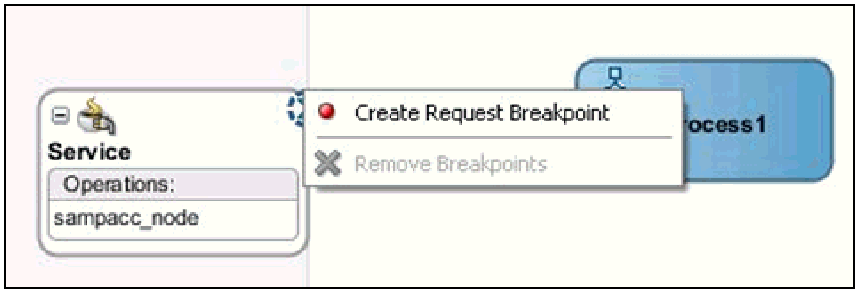
b. Select one of the following options:

Detach - Removes the SOA Debugger without ending the debugging process.

Terminate - Ends the debugging process.

The process will be detached and is displayed, as shown in Figure 9–40.

Figure 9–40 BPEL Process Detached



9.3.4 Debugging an OSB Process in Oracle JDeveloper

This section describes how to debug an OSB process in Oracle JDeveloper. It contains the following topics:

- Section 9.3.4.1, "Prerequisites"
- Section 9.3.4.2, "Debugging an Outbound OSB Process in Oracle JDeveloper"
- Section 9.3.4.3, "Debugging an Inbound OSB Process in Oracle JDeveloper"

9.3.4.1 Prerequisites

Ensure that the *IntegratedWebLogicServer* domain and an OSB process with file output are already created in Oracle JDeveloper.

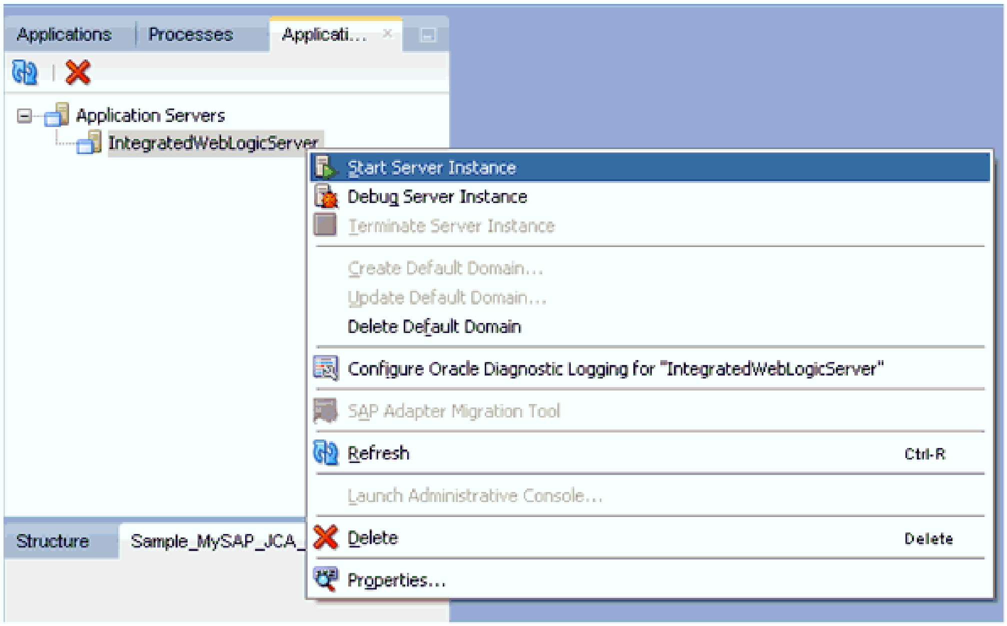
Note: Ensure that the application name and the project name are the same.

1. Open Oracle JDeveloper and start *IntegratedWeblogicServer* or *`startWebLogic.cmd`*.

   a. Click the **Application Servers** tab in the left pane.

   b. Under the Application Servers node, right-click **IntegratedWeblogicServer** and select **Start Server Instance** from the context menu, as shown in Figure 9–41.

Figure 9–41 Start Server Instance

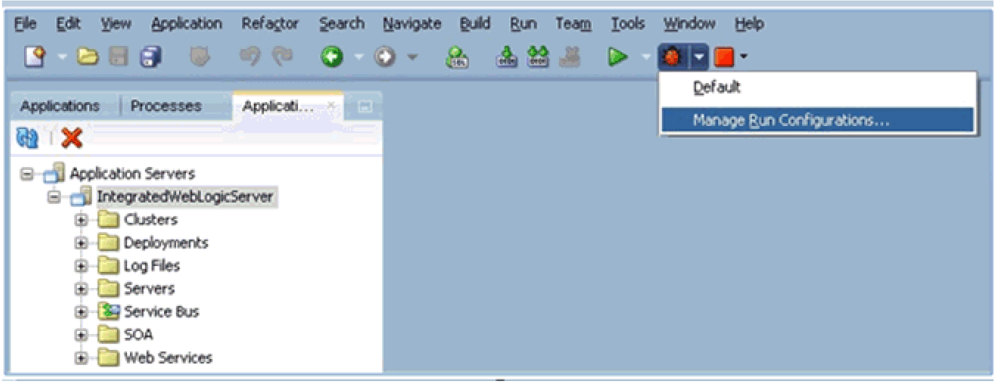


   c. Or, start Oracle WebLogic server from the command prompt using *`startWebLogic.cmd`*.

2. Set the Debugging environment.

   a. Click the down arrow next to the Debug icon and select **Manage Run Configurations** from the context menu, as shown in Figure 9–42.

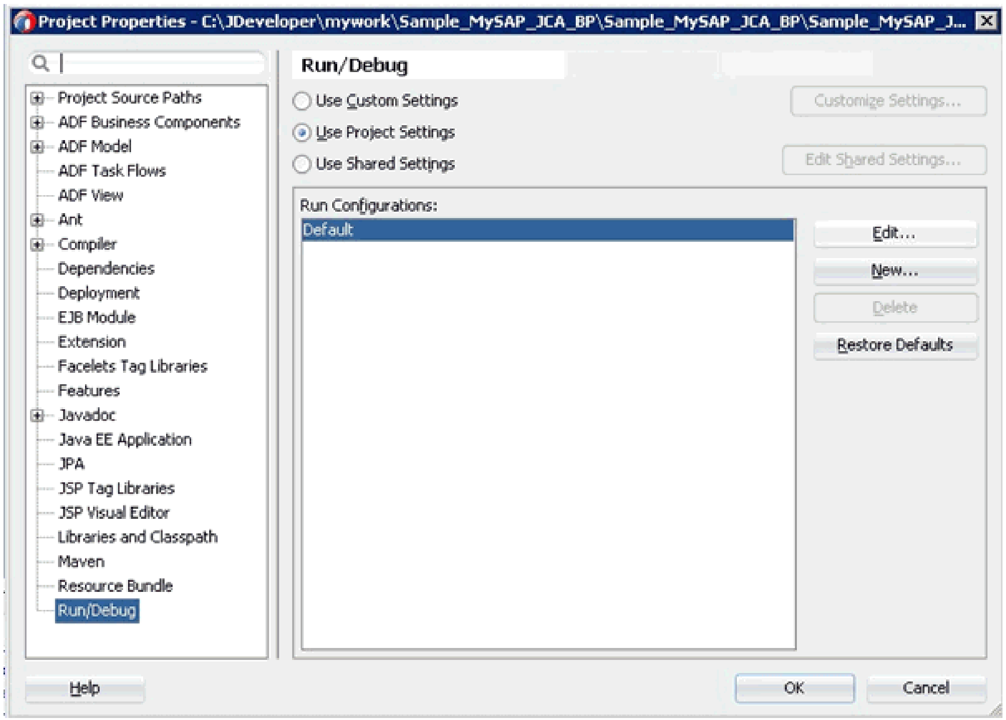
Figure 9–42 Manage Run Configurations



   b. Or, right-click the project and select **Project Properties**.

      The Project Properties dialog is displayed, as shown in Figure 9–43.

Figure 9–43 Project Properties Dialog



Run/Debug

○ Use Custom Settings

● Use Project Settings

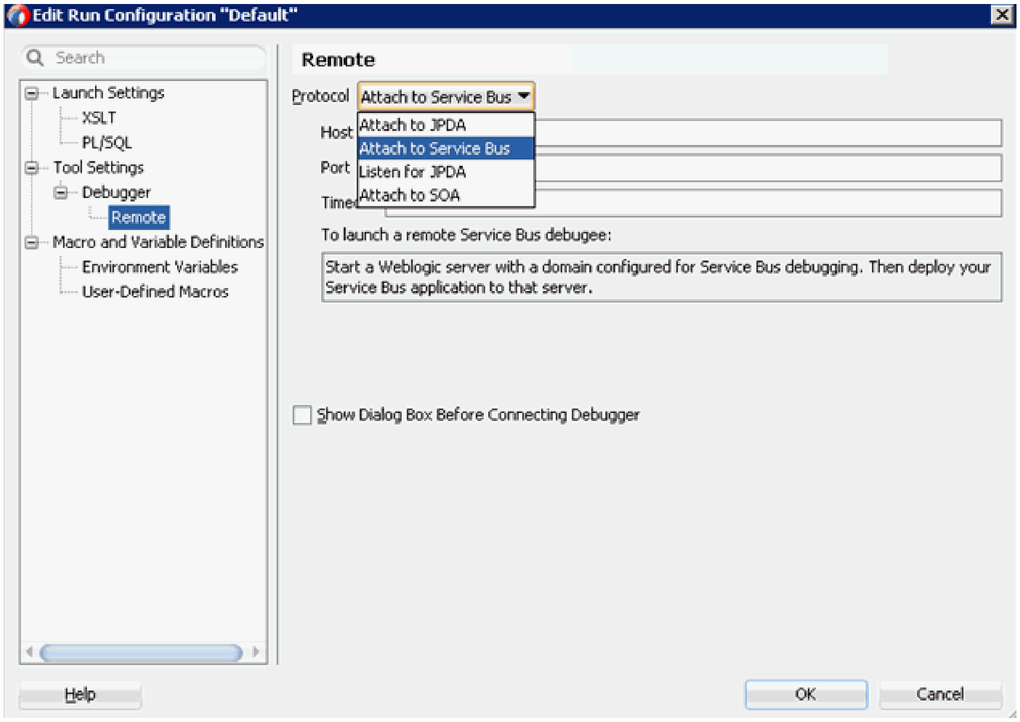
○ Use Shared Settings

Run Configurations: Default

c. Select **Run/Debug** in the left pane and then click **Edit** in the Run/Debug pane (Run Configurations area).

The Edit Run Configuration dialog is displayed, as shown in Figure 9–44.

Figure 9–44 Edit Run Configuration Dialog



Remote

Protocol: Attach to Service Bus

To launch a remote Service Bus debugee:

Start a Weblogic server with a domain configured for Service Bus debugging. Then deploy your Service Bus application to that server.

□ Show Dialog Box Before Connecting Debugger

d. Expand **Tool Settings**, **Debugger** in the left pane, and then click **Remote**.

e. From the Protocol list, select **Attach to Service Bus**.

f. Leave the default values for the Host and Port.

g. Click **OK**.

9.3.4.2 Debugging an Outbound OSB Process in Oracle JDeveloper

1. Deploy the project.

   a. Right-click the project and select **Deploy**.

      The Deployment Action dialog is displayed.

   b. Select the application to deploy and click **Next**.

      The Deploy Configuration dialog is displayed.

   c. Click **Next**.

      The Application Servers dialog is displayed.

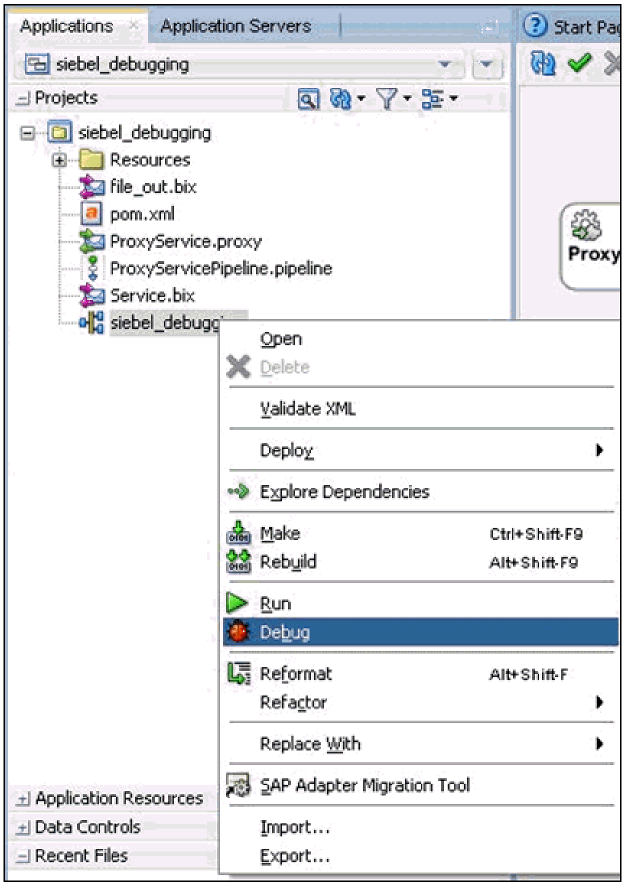
   d. Select **IntegratedWebLogicServer** and then click **Finish** to complete the deployment.

   e. Ensure that the project deployment has completed without any errors or issues before proceeding to the next step.

2. Connect an OSB process to the SOA Debugger.

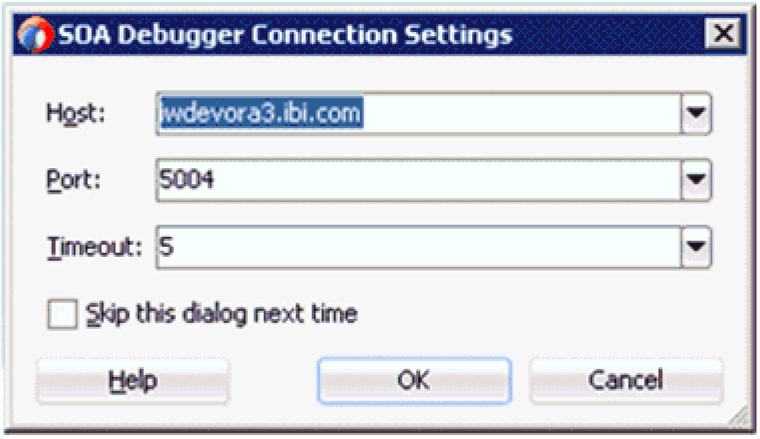
   a. In the Applications tab on the left pane, right-click a composite XML or project for an existing OSB process and then select **Debug** from the context menu, as shown in Figure 9–45.

Figure 9–45 Select OSB Process to Debug



The SOA Debugger Connection Settings dialog is displayed, as shown in Figure 9–46.

Figure 9–46 SOA Debugger Connection Settings Dialog


SOA Debugger Connection Settings

Host: iwdevora3.ibi.com

Port: 5004

Timeout: 5

☐ Skip this dialog next time

b. Click **OK**.

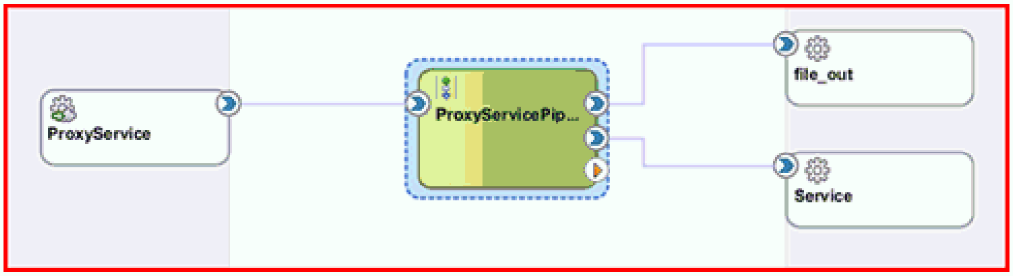
Once the OSB process is connected to the SOA Debugger, the following messages will be displayed in the Debugging log:

```
Debugger attempting to connect to remote process at iwdevora3.ibi.com 5004.
Debugger connected to remote process at iwdevora3.ibi.com 5004.
Debugger process virtual machine is SOA Debugger
```

3. Set the Breakpoints.

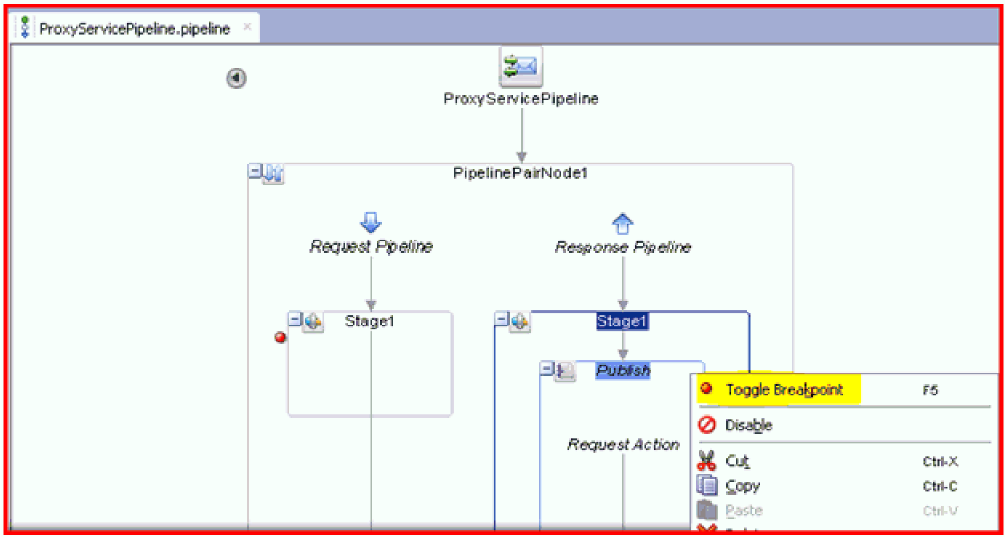
   a. Open the pipeline in its editor by double-clicking on the pipeline, as shown in Figure 9–47.

Figure 9–47 ProxyServicePipeline



   b. Expand the actions until you reach the node where the Breakpoint must be added. In this case, expand **PipelinePairNode1**, right-click the **Stage1** node (under **Request Pipeline**), and select **Toggle Breakpoint** from the context menu, as shown in Figure 9–48.

Figure 9–48 Toggle Breakpoint



Repeat this step for the Publish node (under **Response Pipeline**).

A red icon appears next to the node to indicate that a Breakpoint has been set.

Note: To disable a Breakpoint, right-click the node and select **Disable Breakpoint**.

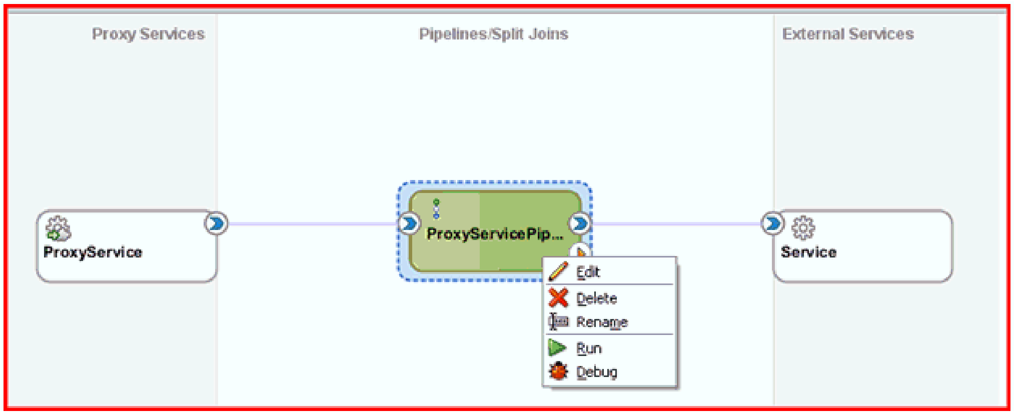
To remove a Breakpoint, right-click the node and select **Toggle Breakpoint** again.

---

4. Initiate Debugging.

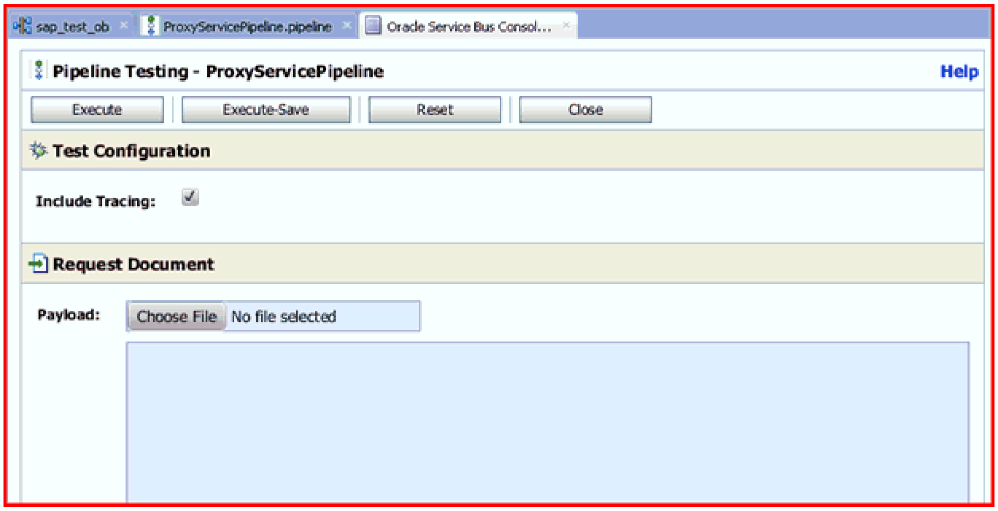
a. Right-click the pipeline in the Application Navigator, and select **Debug**, as shown in Figure 9–49.

Figure 9–49 Select Debug



The process is deployed to the integrated server and the Test Configuration pane will be displayed, as shown in Figure 9–50.

Figure 9–50 Test Configuration Pane



Pipeline Testing - ProxyServicePipeline

Test Configuration

Include Tracing: ☑

Request Document

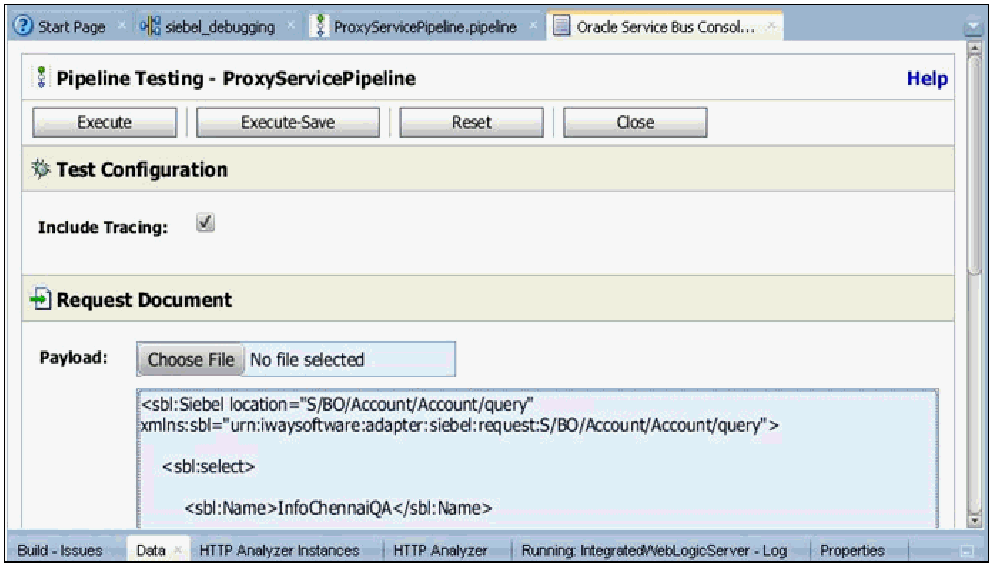
Payload: No file selected

Note: If there is no domain currently running, then the Create Default Domain dialog will be displayed. Enter the connection information for the integrated server and then click **OK**. This process may take several minutes.

---

b. In the Test Configuration pane, enter the test data in the Request Document area, and configure any additional input as required, as shown in Figure 9–51.

Figure 9–51 Test Request Document



Pipeline Testing - ProxyServicePipeline

Test Configuration

Include Tracing: ☑

Request Document

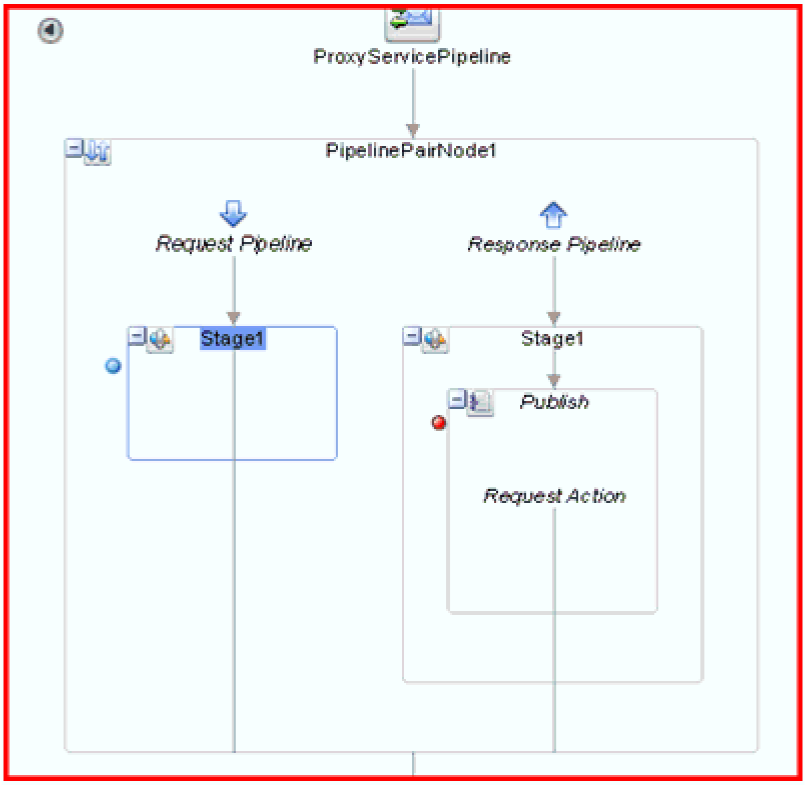
Payload: Choose File No file selected

```
<sbl:Siebel location="S/BO/Account/Account/query"
xmlns:sbl="urn:iwaysoftware:adapter:siebel:request:S/BO/Account/Account/query">

   <sbl:select>

      <sbl:Name>InfoChennaiQA</sbl:Name>
```

c. Click **Execute**.

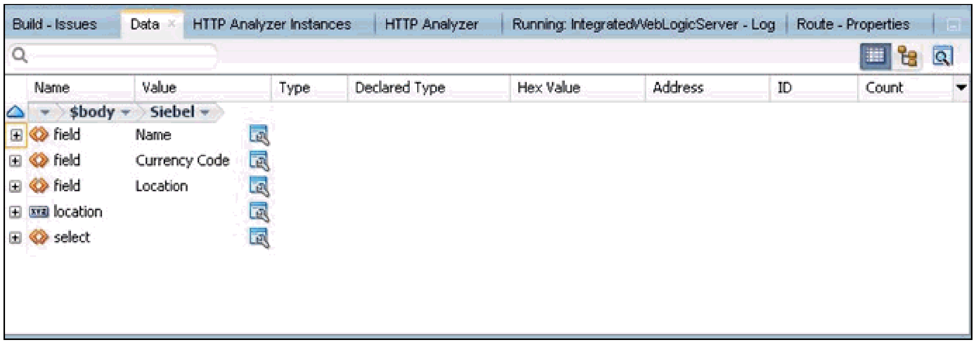
The Test Configuration pane executes the command, but the OSB process stops at the designated Breakpoint and blinks in a blue color, as shown in Figure 9–52.

Figure 9–52 OSB Process Stopped at Breakpoint



d. Check the Data tab at the bottom to verify that the input has passed, as shown in Figure 9–53.

Figure 9–53 Data Tab

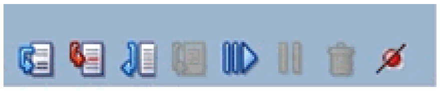


Build - Issues | Data | HTTP Analyzer Instances | HTTP Analyzer | Running: IntegratedWebLogicServer - Log | Route - Properties

Name	Value	Type	Declared Type	Hex Value	Address	ID	Count
$body	Siebel						
field	Name						
field	Currency Code						
field	Location						
location							
select							

e. Use the available Step options to step through the Debugging process, as shown in Figure 9–54.

Figure 9–54 Step Options

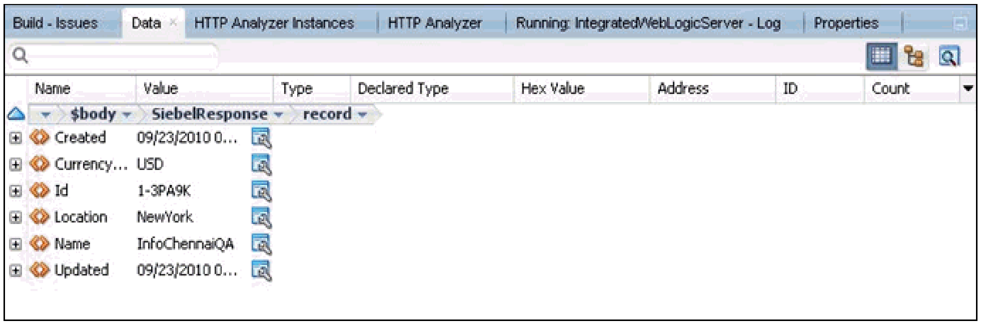


Icon	Description
🟥	Ends or detaches from a debugging session.
	Steps over a frame. This places you at the next Breakpoint (for example, the receive activity in the OSB process on which a Breakpoint was set. If there are no Breakpoints, it steps over all the frames and returns to the first frame. You can also press **F8** to step over a frame.
	Steps into the next valid location. This can be a new frame or the same frame, but in a different location. You can also press **F7** to step into a frame.
	Steps out of a frame. This option is only used to process an OSB scope or sequence activity. After completion of scope processing, it pauses at the next scope or activity in the process. You can also press **Shift-F7**.
▶	Resumes a step operation. You can also press **F9** to resume.

f. Use **Step Over** to go to the next Breakpoint (Response Pipeline in this example).

You will be able to see the output in the Data tab, as shown in Figure 9–55.

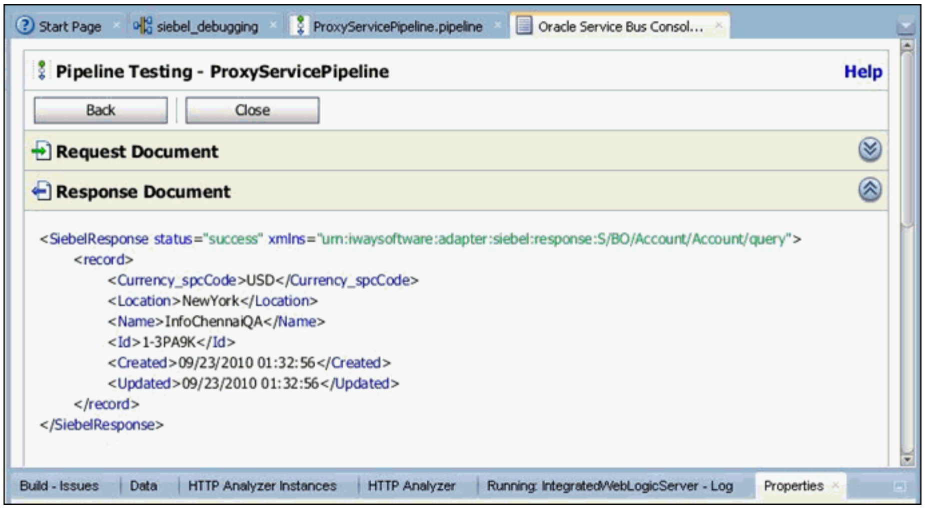
Figure 9–55 Data Tab Output



Name	Value	Type	Declared Type	Hex Value	Address	ID	Count
Created	09/23/2010 0...						
Currency...	USD						
Id	1-3PA9K						
Location	NewYork						
Name	InfoChennaiQA						
Updated	09/23/2010 0...						

You will also be able to see the response in the Oracle Service Bus Console, as shown in Figure 9–56.

Figure 9–56 Oracle Service Bus Console

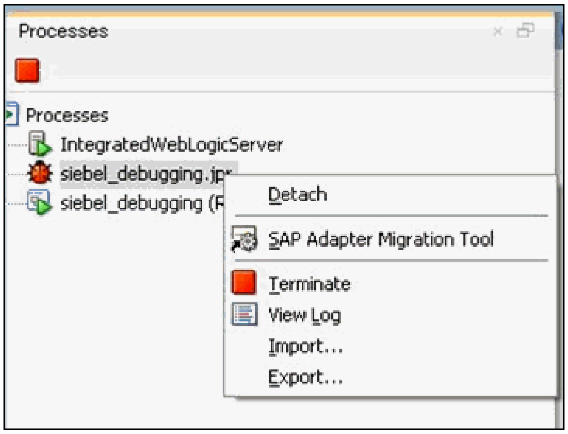


```
<SiebelResponse status="success" xmlns="urn:iwaysoftware:adapter:siebel:response:S/BO/Account/Account/query">
    <record>
        <Currency_spcCode>USD</Currency_spcCode>
        <Location>NewYork</Location>
        <Name>InfoChennaiQA</Name>
        <Id>1-3PA9K</Id>
        <Created>09/23/2010 01:32:56</Created>
        <Updated>09/23/2010 01:32:56</Updated>
    </record>
</SiebelResponse>
```

5. End or detach the Debugging session.

   a. Click **Window** and then **Processes**. Right click on the process in the Processes tab and select **Detach** or **Terminate**, as shown in Figure 9–57.

Figure 9–57 Detach Debugging Session



b. Select one of the following options:

   Detach - Removes the SOA debugger without ending the debugging process.

   Terminate - Ends the debugging process.

9.3.4.3 Debugging an Inbound OSB Process in Oracle JDeveloper

1. Create an inbound OSB process.

   Ensure that the application name and the project name are the same.

2. Deploy the project.

   a. Right-click the project and select **Deploy**.

      The Deployment Action dialog is displayed.

   b. Select the application to deploy and click **Next**.

      The Deploy Configuration dialog is displayed.

   c. Click **Next**.

      The Application Servers dialog is displayed.

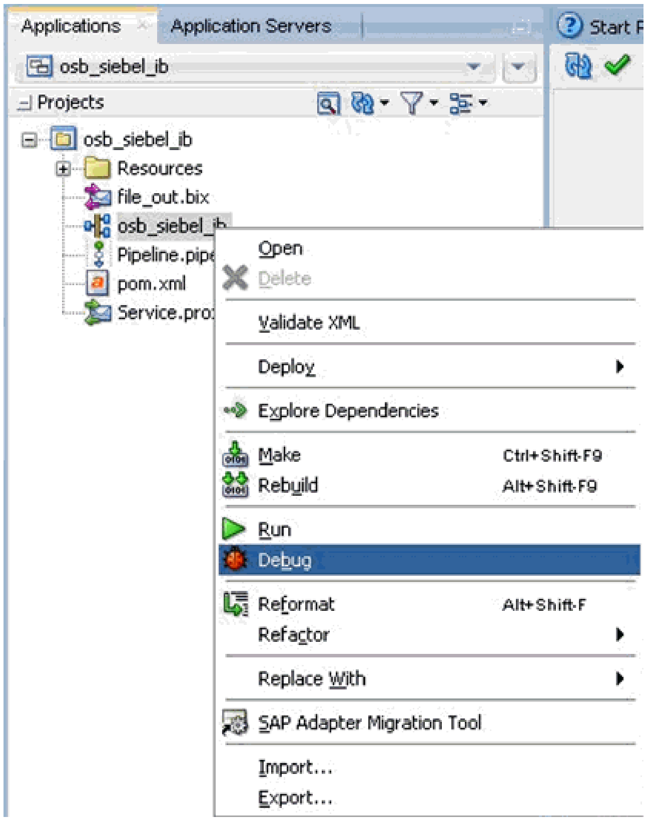
   d. Select **IntegratedWebLogicServer** and then click **Finish** to complete the deployment.

   e. Ensure that the project deployment has completed without any errors or issues before proceeding to the next step.

3. Connect an OSB process to the SOA Debugger.

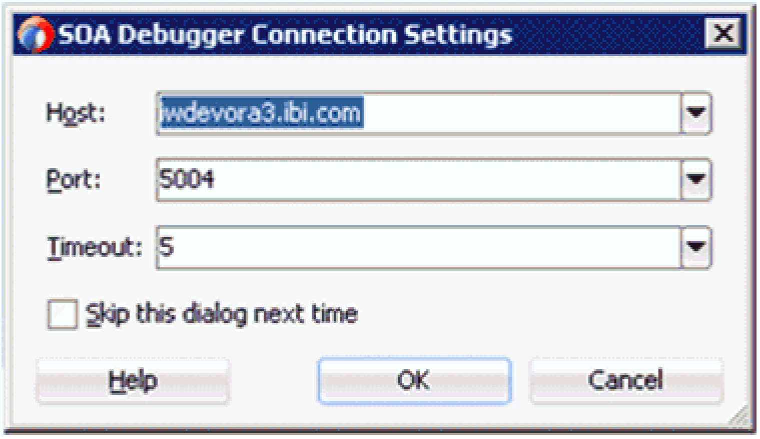
   a. In the Applications tab on the left pane, right-click a composite XML or project for an existing OSB process and then select **Debug** from the context menu, as shown in Figure 9–58.

Figure 9–58 Select OSB Process to Debug



The SOA Debugger Connection Settings dialog is displayed, as shown in Figure 9–59.

Figure 9–59 SOA Debugger Connection Settings Dialog



Host: iwdevora3.ibi.com

Port: 5004

Timeout: 5

□ Skip this dialog next time

b. Click **OK**.

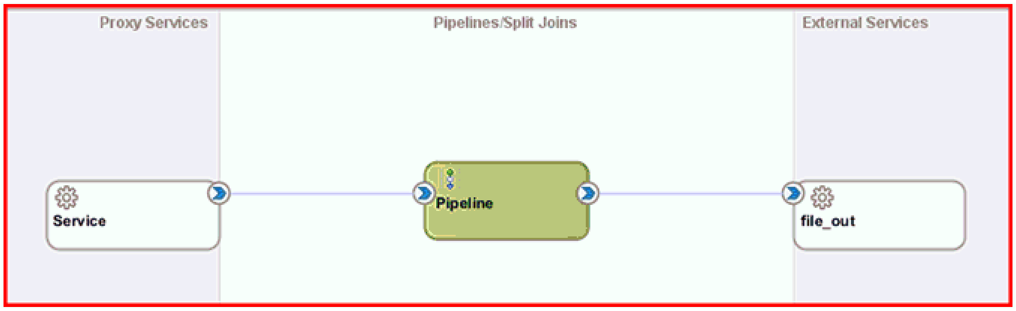
Once the OSB process is connected to the SOA Debugger, the following messages will be displayed in the Debugging log:

```
Debugger attempting to connect to remote process at iwdevora3.ibi.com 5004.
Debugger connected to remote process at iwdevora3.ibi.com 5004.
Debugger process virtual machine is SOA Debugger
```

4. Set the Breakpoints.

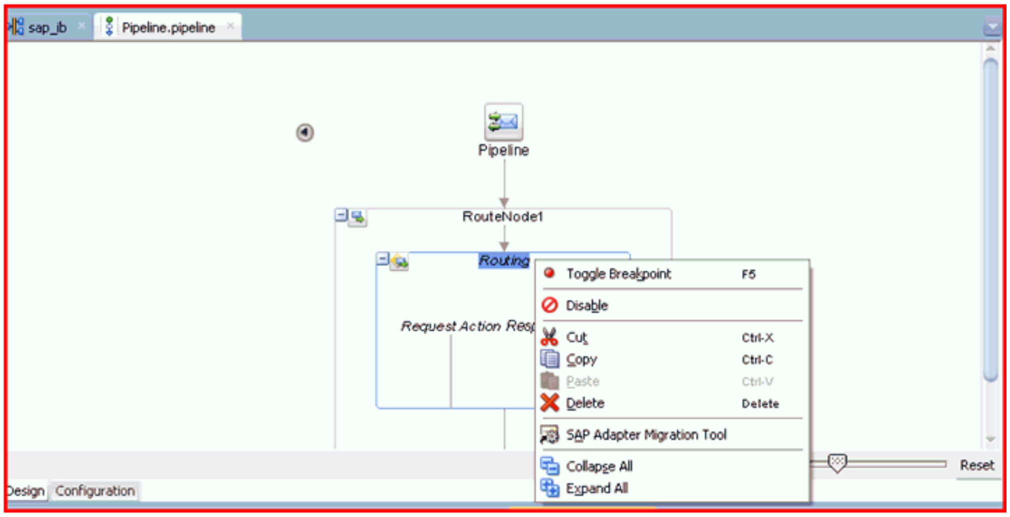
   a. Open the pipeline in its editor by double-clicking on the pipeline, as shown in Figure 9–60.

Figure 9–60 ProxyServicePipeline



   b. Expand the actions until you reach the node where the Breakpoint must be added. In this case, expand until you reach **Routing**. Right-click the **Routing** node and select **Toggle Breakpoint** from the context menu, as shown in Figure 9–61.

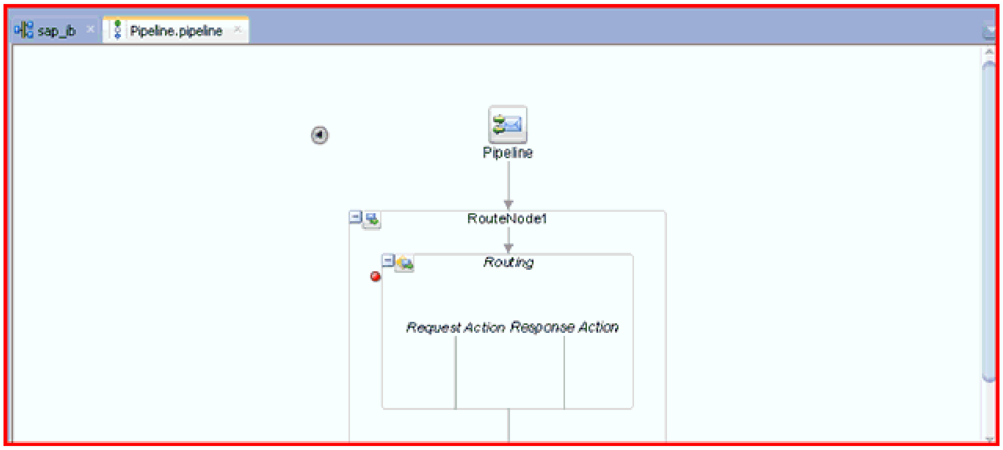
Figure 9–61 Toggle Breakpoint



Repeat the above step for each node to which you want to add a Breakpoint. In this example, a Breakpoint is set only for the **Routing** node.

A red icon appears next to the node to indicate that a Breakpoint has been set, as shown in Figure 9–62.

Figure 9–62 Red Icon for Set Breakpoint



Note: To disable a Breakpoint, right-click the node and select **Disable Breakpoint**.

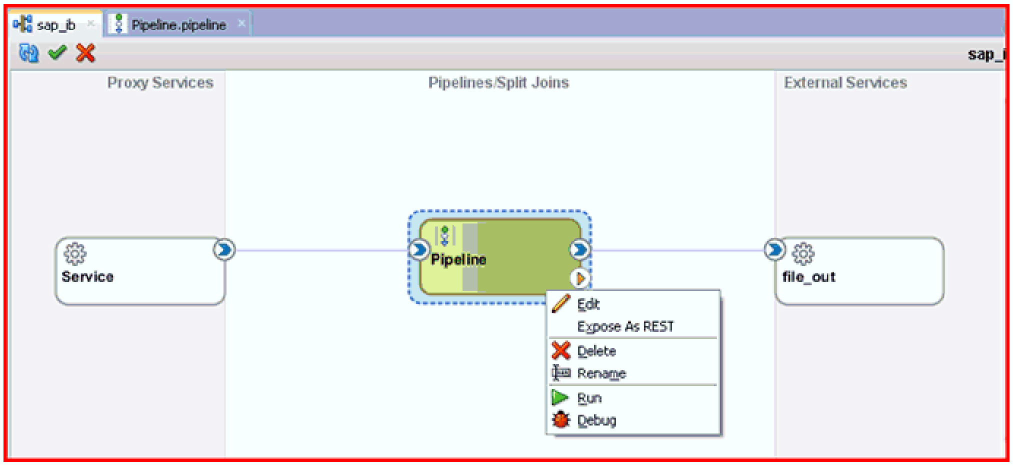
To remove a Breakpoint, right-click the node and select **Toggle Breakpoint** again.

---

5. Initiate Debugging.

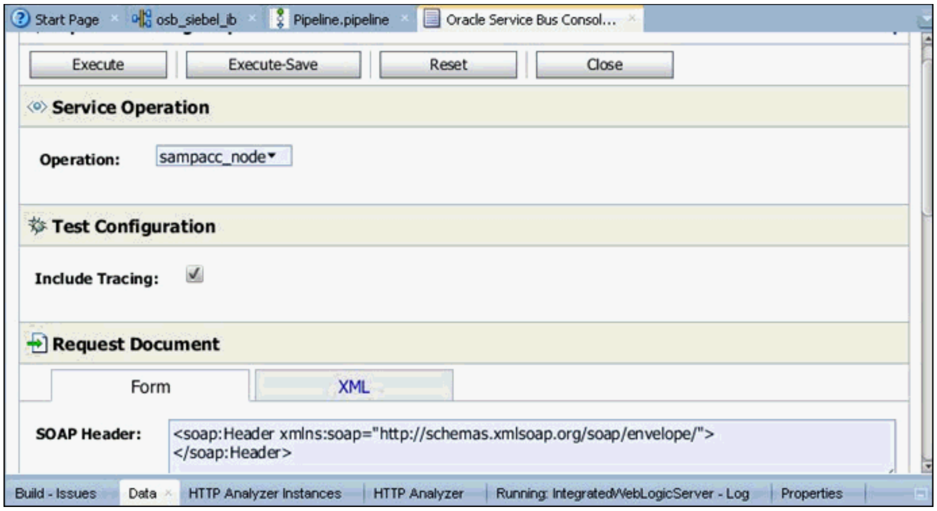
   a. Right-click the pipeline in the Application Navigator, and select **Debug**, as shown in Figure 9–63.

Figure 9–63 Select Debug



The process is deployed to the integrated server and the Test Configuration pane will be displayed, as shown in Figure 9–64.

Figure 9–64 Test Configuration Pane



Operation: sampacc_node

Include Tracing: ☑

SOAP Header:

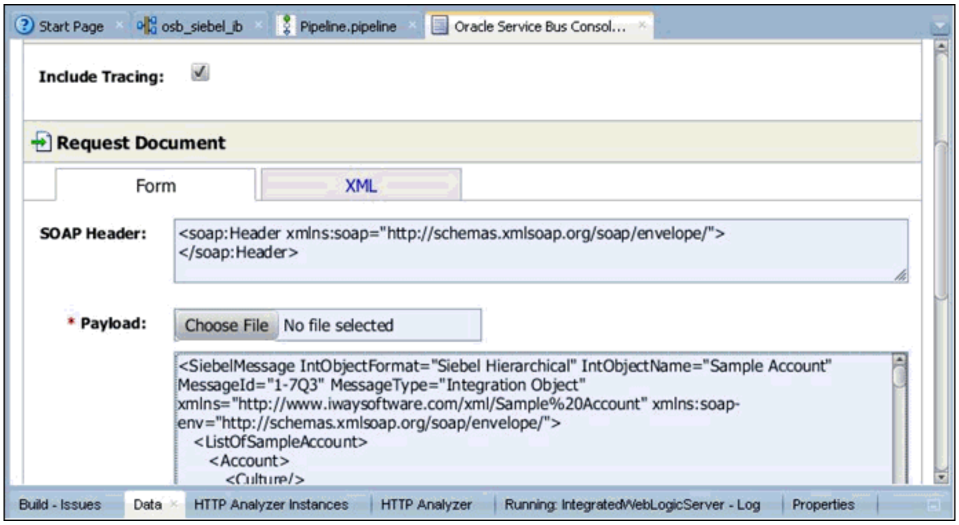
```
<soap:Header xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
</soap:Header>
```

Note: If there is no domain currently running, then the Create Default Domain dialog will be displayed. Enter the connection information for the integrated server and then click **OK**. This process may take several minutes.

---

b. In the Test Configuration pane, enter the test data in the Request Document area, and configure any additional input as required, as shown in Figure 9–65.

Figure 9–65 Test Request Document



Include Tracing: ☑

SOAP Header:

```
<soap:Header xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
</soap:Header>
```

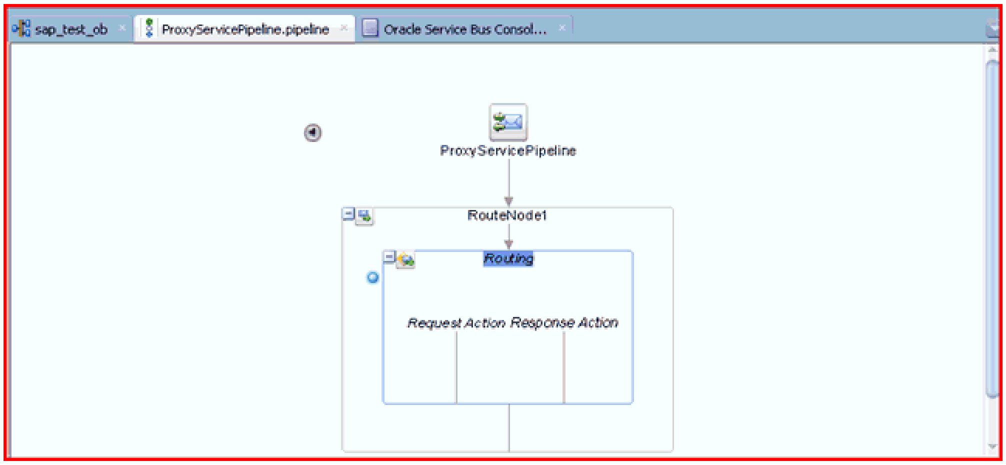
* Payload: No file selected

```
<SiebelMessage IntObjectFormat="Siebel Hierarchical" IntObjectName="Sample Account"
MessageId="1-7Q3" MessageType="Integration Object"
xmlns="http://www.iwaysoftware.com/xml/Sample%20Account" xmlns:soap-
env="http://schemas.xmlsoap.org/soap/envelope/">
  <ListOfSampleAccount>
    <Account>
      <Culture/>
```

c. Click **Execute**.

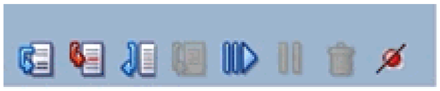
The Test Configuration pane executes the command, but the OSB process stops at the designated Breakpoint and blinks in a blue color, as shown in Figure 9–66.

Figure 9–66 OSB Process Stopped at Breakpoint


sap_test_ob
ProxyServicePipeline.pipeline
Oracle Service Bus Consol...
ProxyServicePipeline
RouteNode1
Routing
Request Action Response Action

d. Use the available Step options to step through the Debugging process, as shown in Figure 9–67.

Figure 9–67 Step Options



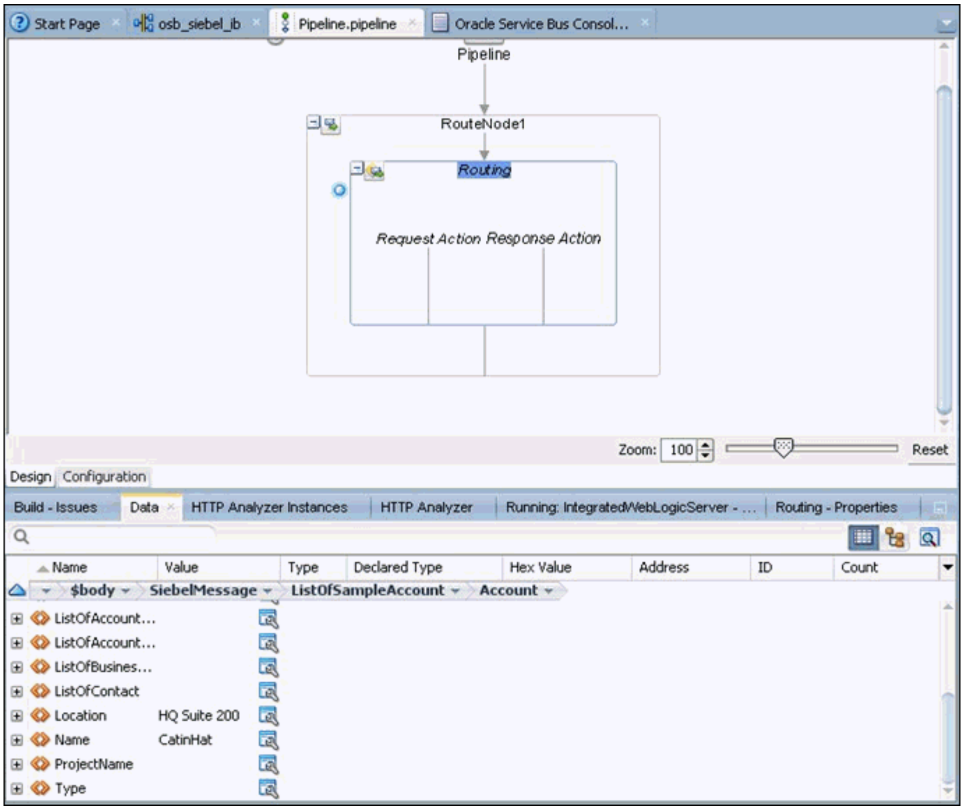
Icon	Description
🟥	Ends or detaches from a debugging session.
	Steps over a frame. This places you at the next Breakpoint (for example, the receive activity in the OSB process on which a Breakpoint was set. If there are no Breakpoints, it steps over all the frames and returns to the first frame. You can also press **F8** to step over a frame.
	Steps into the next valid location. This can be a new frame or the same frame, but in a different location. You can also press **F7** to step into a frame.
	Steps out of a frame. This option is only used to process an OSB scope or sequence activity. After completion of scope processing, it pauses at the next scope or activity in the process. You can also press **Shift-F7**.
	Resumes a step operation. You can also press **F9** to resume.

e. Use **Step Over** to complete the process execution

Note: Since there is only one Breakpoint in this example, using **Step Over** completes the process execution.

You will be able to see the response document displayed in the Data tab, as shown in Figure 9–68.

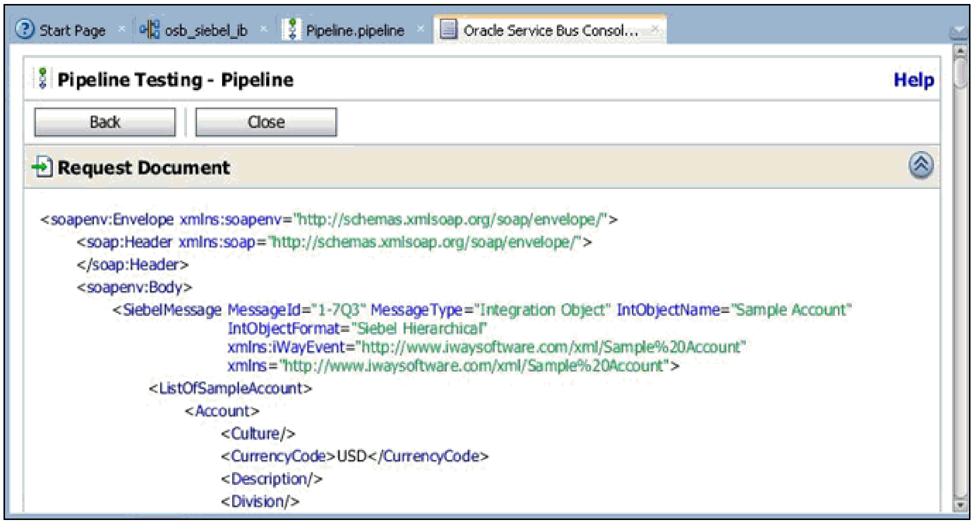
Figure 9–68 Data Tab Output



Name	Value	Type	Declared Type	Hex Value	Address	ID	Count
ListOfAccount...							
ListOfAccount...							
ListOfBusines...							
ListOfContact							
Location	HQ Suite 200						
Name	CatinHat						
ProjectName							
Type							

You will also be able to see the response in the Oracle Service Bus Console, as shown in Figure 9–69.

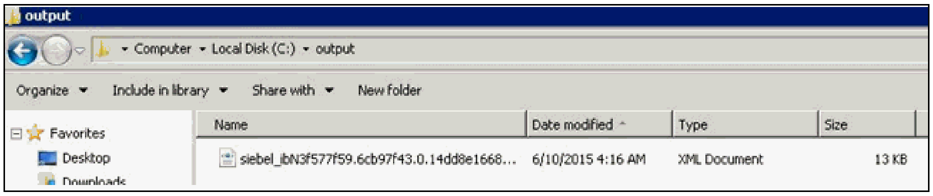
Figure 9–69 Oracle Service Bus Console



```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Header xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    </soap:Header>
    <soapenv:Body>
        <SiebelMessage MessageId="1-7Q3" MessageType="Integration Object" IntObjectName="Sample Account"
                    IntObjectFormat="Siebel Hierarchical"
                    xmlns:iWayEvent="http://www.iwaysoftware.com/xml/Sample%20Account"
                    xmlns="http://www.iwaysoftware.com/xml/Sample%20Account">
            <ListOfSampleAccount>
                <Account>
                    <Culture/>
                    <CurrencyCode>USD</CurrencyCode>
                    <Description/>
                    <Division/>
```

The output will also be available in the configured output location, as show in Figure 9–70.

Figure 9–70 Configured Output Location

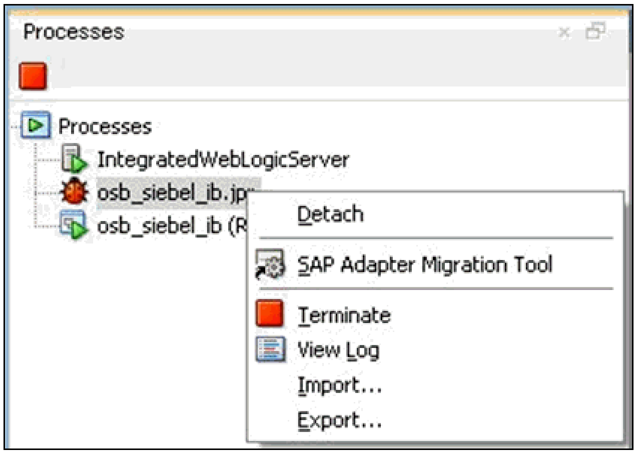


Name	Date modified	Type	Size
siebel_ibN3f577f59.6cb97f43.0.14dd8e1668...	6/10/2015 4:16 AM	XML Document	13 KB

6. End or detach the Debugging session.

   a. Click **Window** and then **Processes**. Right click on the process in the Processes tab and select **Detach** or **Terminate**, as shown in Figure 9–71.

Figure 9–71 Detach Debugging Session



   b. Select one of the following options:

   Detach - Removes the SOA debugger without ending the debugging process.

Terminate - Ends the debugging process.

9.4 Exception Filter

This section describes how to configure exception filter functionality for the Oracle Application Adapter for Siebel and includes a sample testing scenario.

This section contains the following topic:

- Section 9.4.1, "Configuring the Exception Filter"

The exception filter is supported only for outbound processes that use J2CA configurations. This feature is not supported for BSE configurations and inbound processes that use J2CA configurations.

The exception filter uses the `com.ibi.afjca.oracle.AdapterExceptionFilter` class to filter the generated exceptions. This class filters the exceptions and categorizes them into the following categories:

- PCRetriableResourceException
- PCResourceException

The following exceptions are represented in the fault policies file:

- PCRetriableResourceException - A remote fault.
- PCResourceException - A binding fault.

9.4.1 Configuring the Exception Filter

Exception filter configuration consists of the following steps and topics:

1. Section 9.4.1.1, "Generating a WSDL File"
2. Section 9.4.1.2, "Creating a BPEL process With Exception Filter Functionality"
3. Section 9.4.1.3, "Creating Fault Policies and Fault Binding Files"
4. Section 9.4.1.4, "Adjusting for Known Deployment Issues With 12c"
5. Section 9.4.1.5, "Deploying and Testing the BPEL Process With Exception Filter Functionality"

9.4.1.1 Generating a WSDL File

To generate a WSDL file:

1. Open Application Explorer and create a J2CA configuration.

   For more information, see "Creating a Configuration for J2CA" on page 2-4.

2. Create a target for the Siebel adapter and then connect to the target.

   For more information, see "Establishing a Connection (Target) for Siebel" on page 2-5.

3. Generate a WSDL for the appropriate object.

   For more information, see "Generating WSDL (J2CA Configurations Only)" on page 2-27.

9.4.1.2 Creating a BPEL process With Exception Filter Functionality

To create a BPEL process with exception filter functionality:

1. Open JDeveloper and create a new SOA application.

   For more information, see "Creating an Empty Composite for SOA" on page 4-9.

2. Create a new SOA project (for example, Exception_Filter).

3. Create a third party adapter service component.

   For more information, see "Configuring a Third Party Adapter Service Component" on page 4-11.

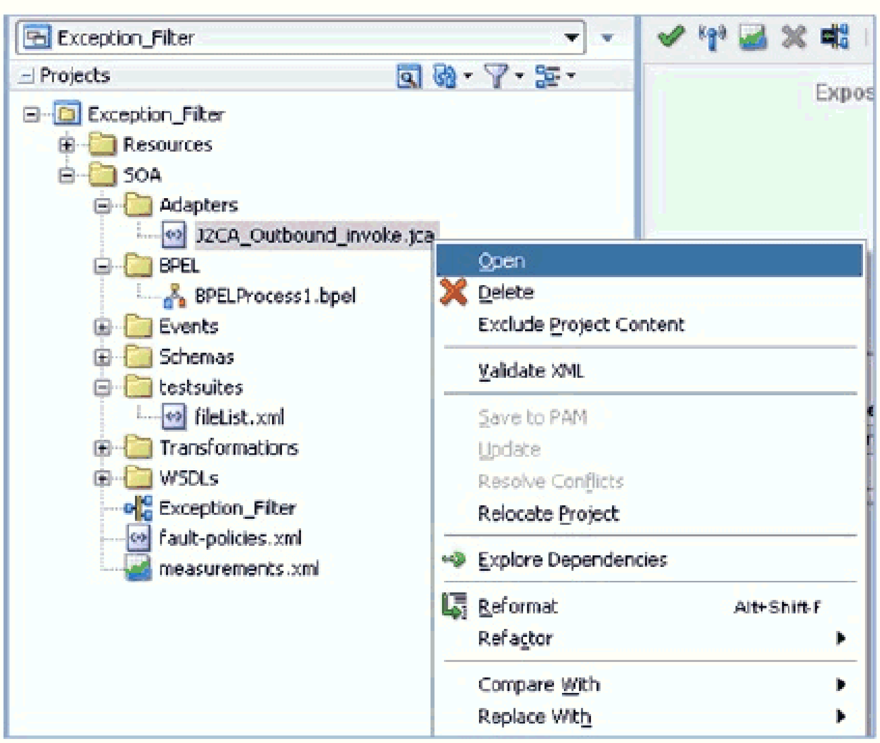
   Once the third party adapter service component is created, the WSDL file (with corresponding schemas and JCA file) is imported to the JDeveloper project.

   For more information, see "Defining a BPEL Outbound Process" on page 4-11.

4. Modify the imported JCA file.

   a. Right-click the imported JCA file and select **Open** from the menu, as shown in Figure 9–72.

Figure 9–72 Application Navigator Tab



   b. In the `<interaction-spec>` element, add the `ExceptionFilter` property. For example:

```
<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. Save the modified JCA file.

5. Once the third party adapter service component is created and the JCA file is modified, continue with the remainder of the BPEL process creation.

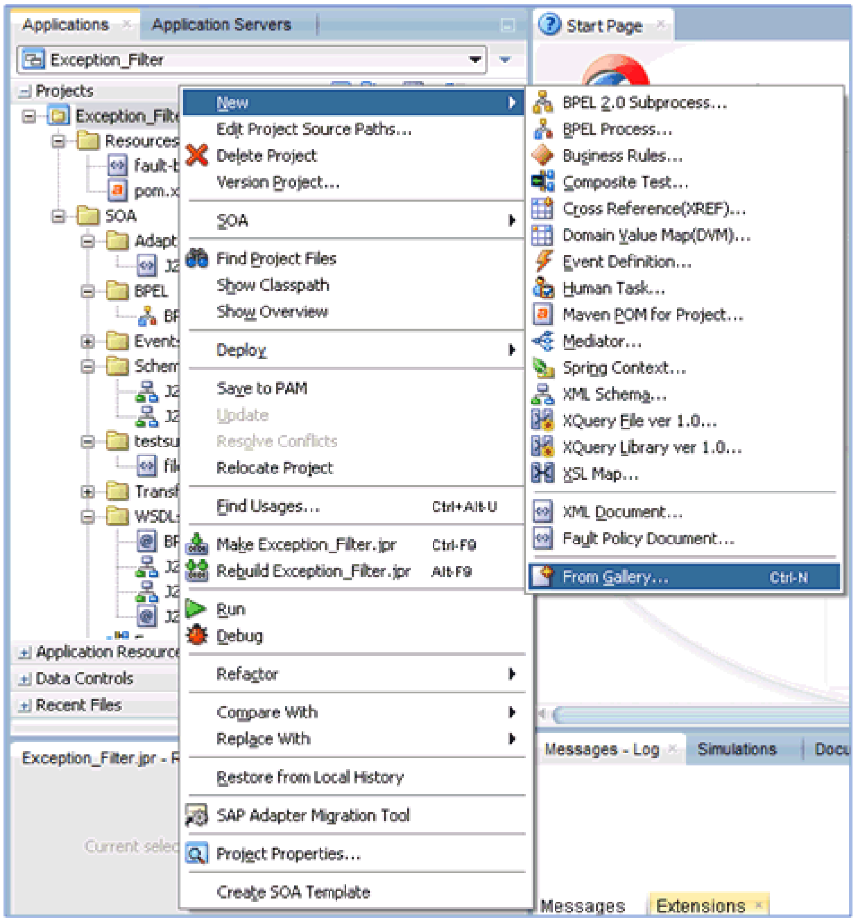
   For more information, see "Defining a BPEL Outbound Process" on page 4-11.

9.4.1.3 Creating Fault Policies and Fault Binding Files

To create fault binding files:

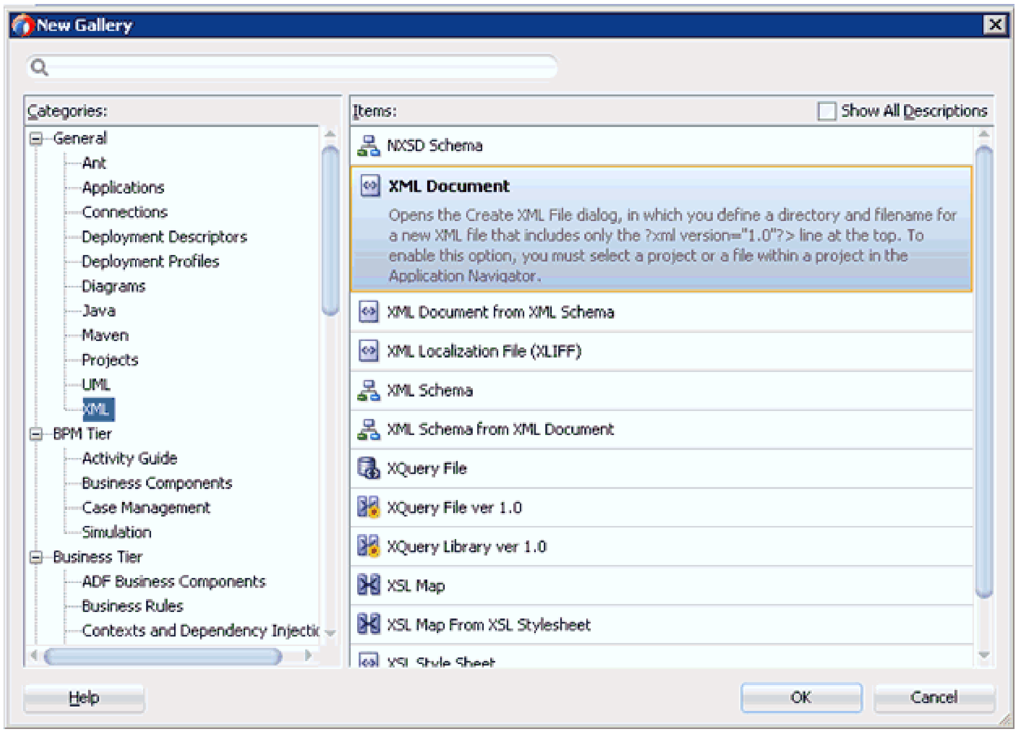
1. Right-click the created SOA project (for example, Exception_Filter), select **New**, and then click **From Gallery**, as shown in Figure 9–73.

Figure 9–73 Applications Tab



The New Gallery dialog is displayed. Under the General category, click **XML**, as shown in Figure 9–74.

Figure 9–74 New Gallery Dialog

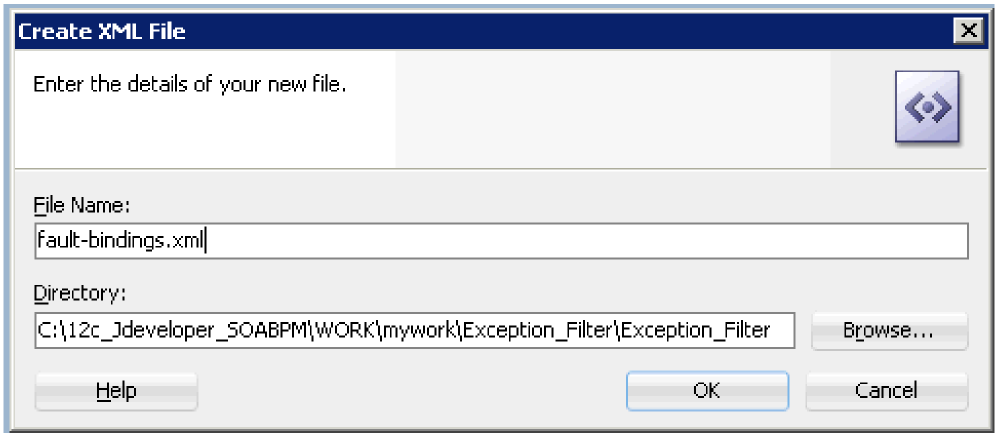


Opens the Create XML File dialog, in which you define a directory and filename for a new XML file that includes only the ?xml version="1.0"?> line at the top. To enable this option, you must select a project or a file within a project in the Application Navigator.

2. Select **XML Document** under Items and then click **OK**.

   The Create XML File dialog is displayed, as shown in Figure 9–75.

Figure 9–75 Create XML File Dialog



Enter the details of your new file.

File Name: fault-bindings.xml

Directory: C:\12c_Jdeveloper_SOABPM\WORK\mywork\Exception_Filter\Exception_Filter

3. In the File Name field, type **fault-bindings.xml** and click **OK**.

4. Add the appropriate fault binding functions in the **fault-bindings.xml** file.

   To view a sample **fault-bindings.xml** file, see "Sample Fault-Bindings.xml File" on page 9-48.

> **Note:** The parameter in the `<name>` element is the name of the created BPEL process.

5. Save the **fault-bindings.xml** file.

Sample Fault-Bindings.xml File

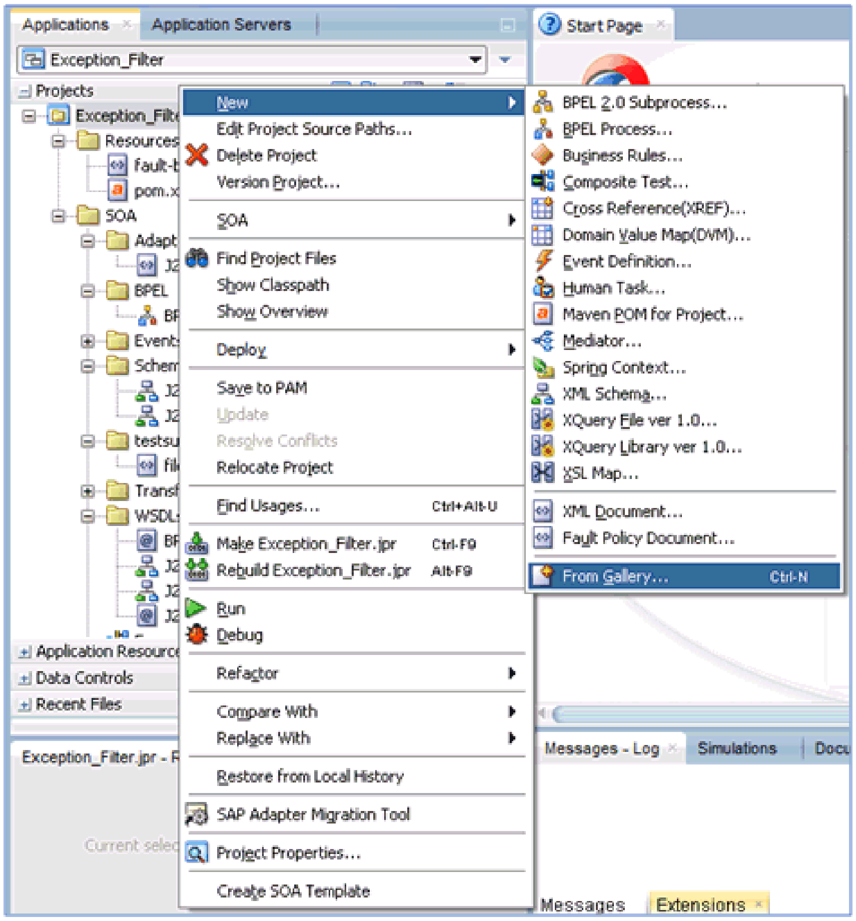
```
<?xml version="1.0" encoding="UTF-8" ?>
<faultPolicyBindings version="2.0.1"
xmlns="http://schemas.oracle.com/bpel/faultpolicy"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<component faultPolicy="bpelFaultHandling">
<name>BPELProcess1</name>
</component>

</faultPolicyBindings>
```

Creating Fault Policies Files

To create fault policies files:

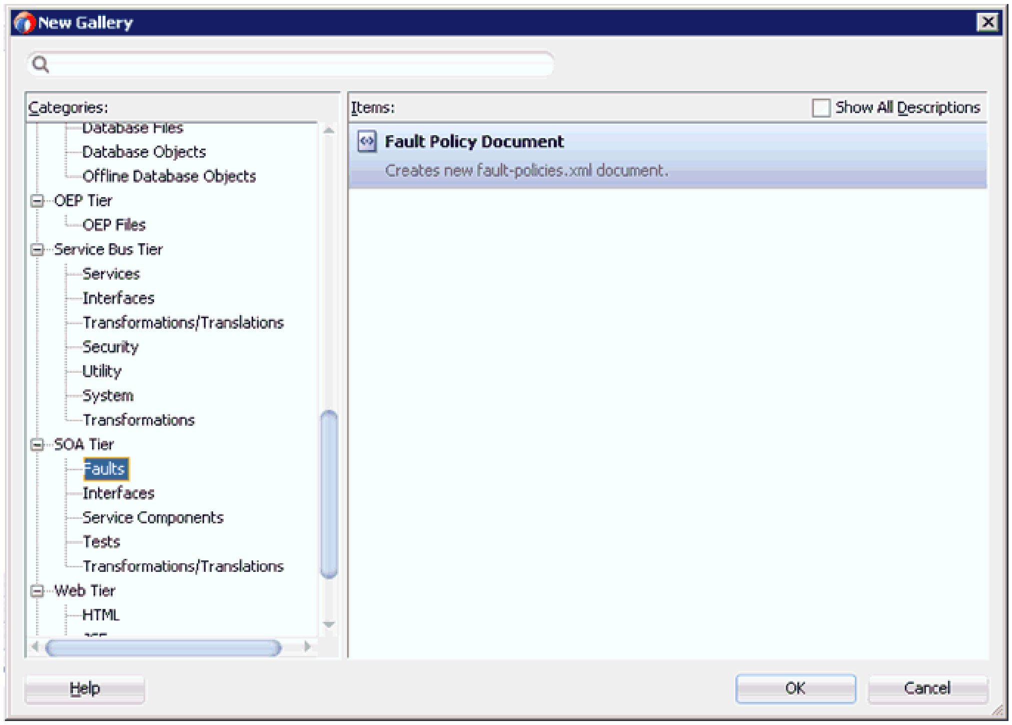
1. Right-click the created SOA project (for example, Exception_Filter), select **New**, and then click **From Gallery**, as shown in Figure 9–76.

Figure 9–76 Applications Tab



The New Gallery dialog is displayed. Under the SOA Tier category, select **Faults**, as shown in Figure 9–77.

Figure 9–77 New Gallery Dialog



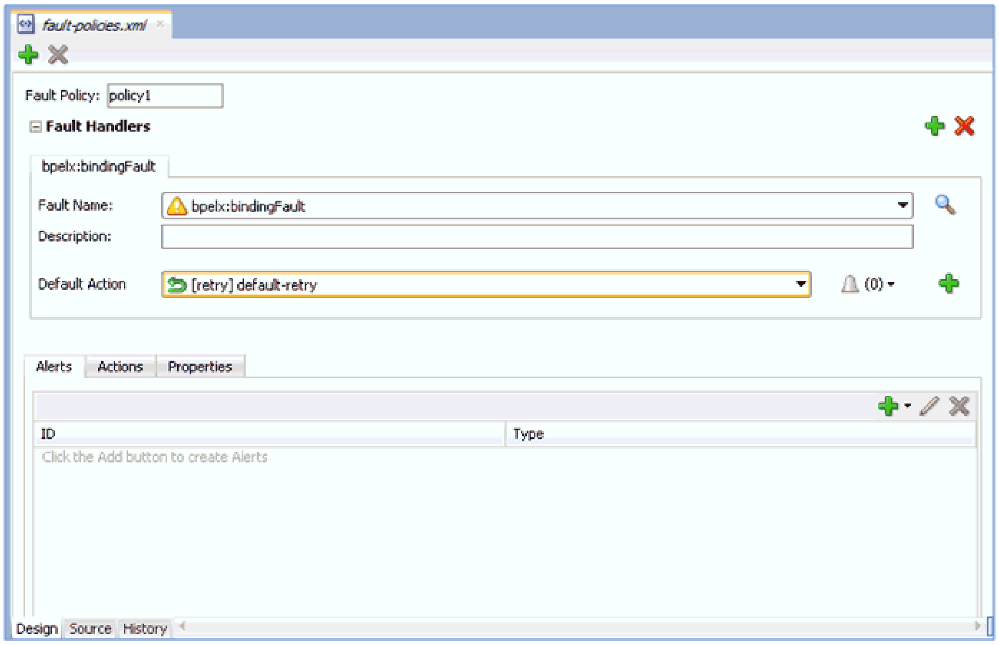
Fault Policy Document

Creates new fault-policies.xml document.

2. Select **Fault Policy Document** under Items and then click **OK**.

3. In the fault-policies.xml tab, select **bpelx:bindingFault** from the Fault Name drop-down list and **[retry] default-retry** from the Default Action drop-down list, as shown in Figure 9–78.

Figure 9–78 Fault-policies.xml Tab



Fault Policy: policy1

Fault Name: bpelx:bindingFault

Description:

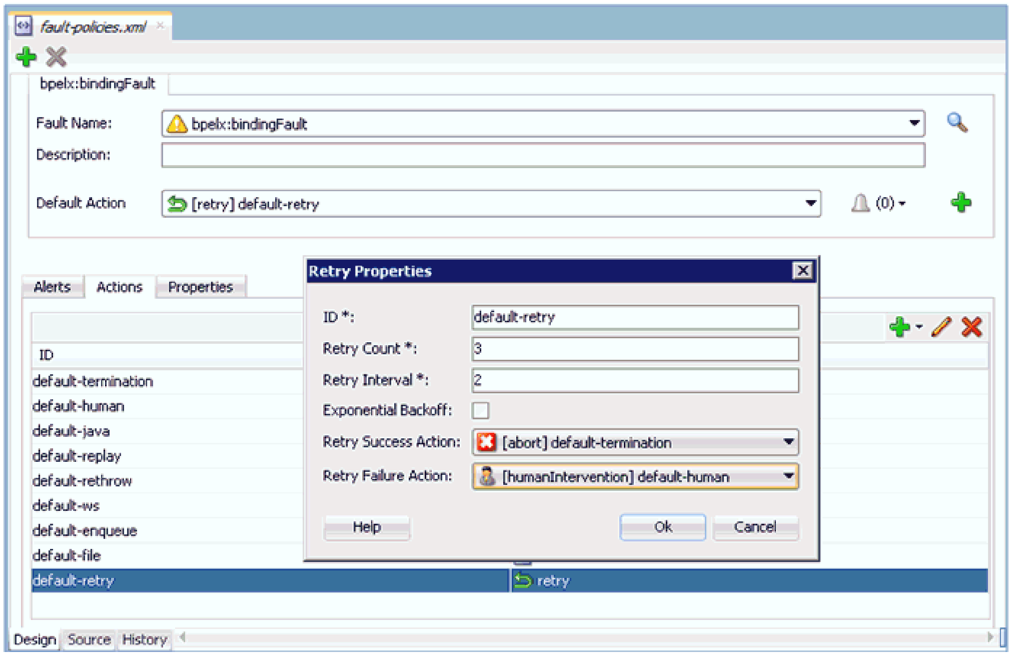
Default Action [retry] default-retry

Click the Add button to create Alerts

4. Click the **Actions** tab and then double-click **default-retry**.

The Retry Properties dialog box is displayed, as shown in Figure 9–79.

Figure 9–79 Retry Properties Dialog Box



Fault Name: bpelx:bindingFault

Description:

Default Action: [retry] default-retry

Retry Properties

ID *: default-retry

Retry Count *: 3

Retry Interval *: 2

Exponential Backoff: □

Retry Success Action: [abort] default-termination

Retry Failure Action: [humanIntervention] default-human

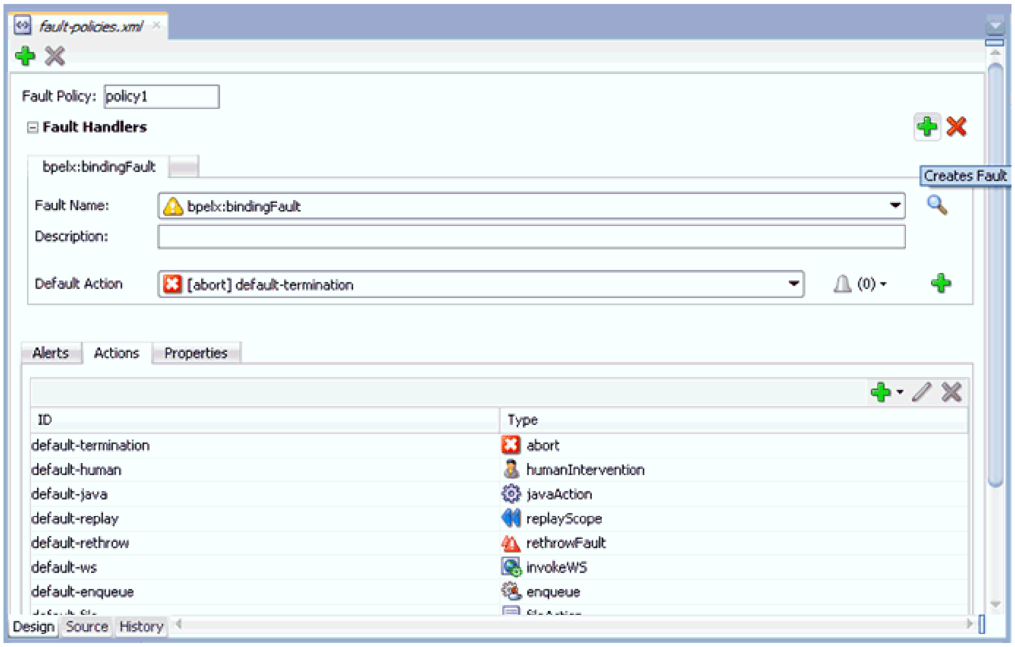
ID	
default-termination	
default-human	
default-java	
default-replay	
default-rethrow	
default-ws	
default-enqueue	
default-file	
default-retry	retry

5. Select **[abort] default-termination** from the Retry Success Action drop-down list and **[humanIntervention] default-human** from the Retry Failure Action drop-down list.

6. Click **OK**.

7. Click **Add** to create another fault handler, as shown in Figure 9–80.

Figure 9–80 Fault-policies.xml Tab



Fault Policy: policy1

Fault Name: bpelx:bindingFault

Description:

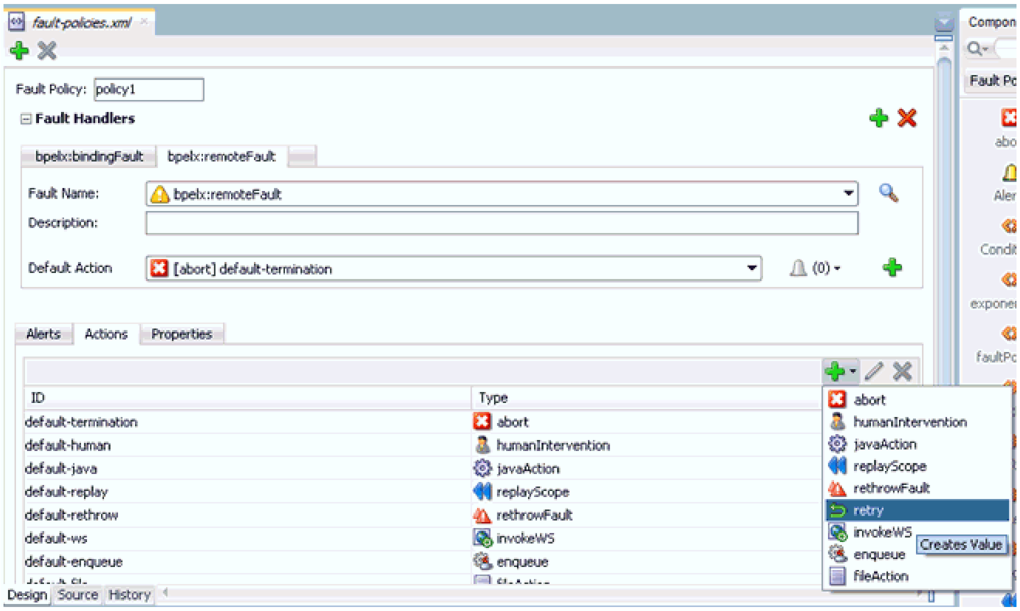
Default Action: [abort] default-termination

ID	Type
default-termination	abort
default-human	humanIntervention
default-java	javaAction
default-replay	replayScope
default-rethrow	rethrowFault
default-ws	invokeWS
default-enqueue	enqueue

8. In the fault-policies.xml tab, select **bpelx:remoteFault** from the Fault Name drop-down list and **[abort] default-termination** from the Default Action drop-down list.

9. In the Actions tab, click **Add** and then select **retry**, as shown in Figure 9–81.

Figure 9–81 Actions Tab



Fault Policy: policy1

Fault Handlers

bpelx:bindingFault | bpelx:remoteFault

Fault Name: bpelx:remoteFault

Description:

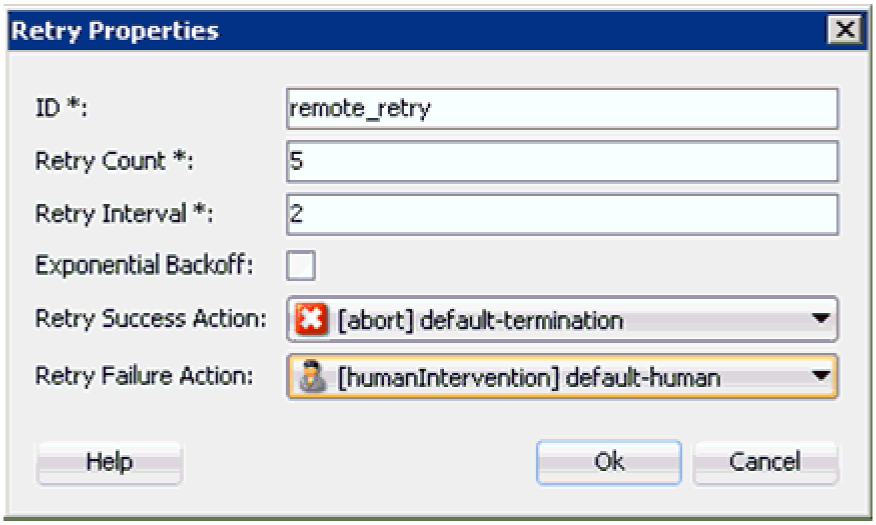
Default Action: [abort] default-termination (0)

Alerts | Actions | Properties

ID	Type
default-termination	abort
default-human	humanIntervention
default-java	javaAction
default-replay	replayScope
default-rethrow	rethrowFault
default-ws	invokeWS
default-enqueue	enqueue

The Retry Properties dialog is displayed, as shown in Figure 9–82.

Figure 9–82 Retry Properties Dialog Box



Retry Properties

ID *: remote_retry

Retry Count *: 5

Retry Interval *: 2

Exponential Backoff: □

Retry Success Action: [abort] default-termination

Retry Failure Action: [humanIntervention] default-human

10. Provide values for the ID, Retry Count, and Retry Interval fields.

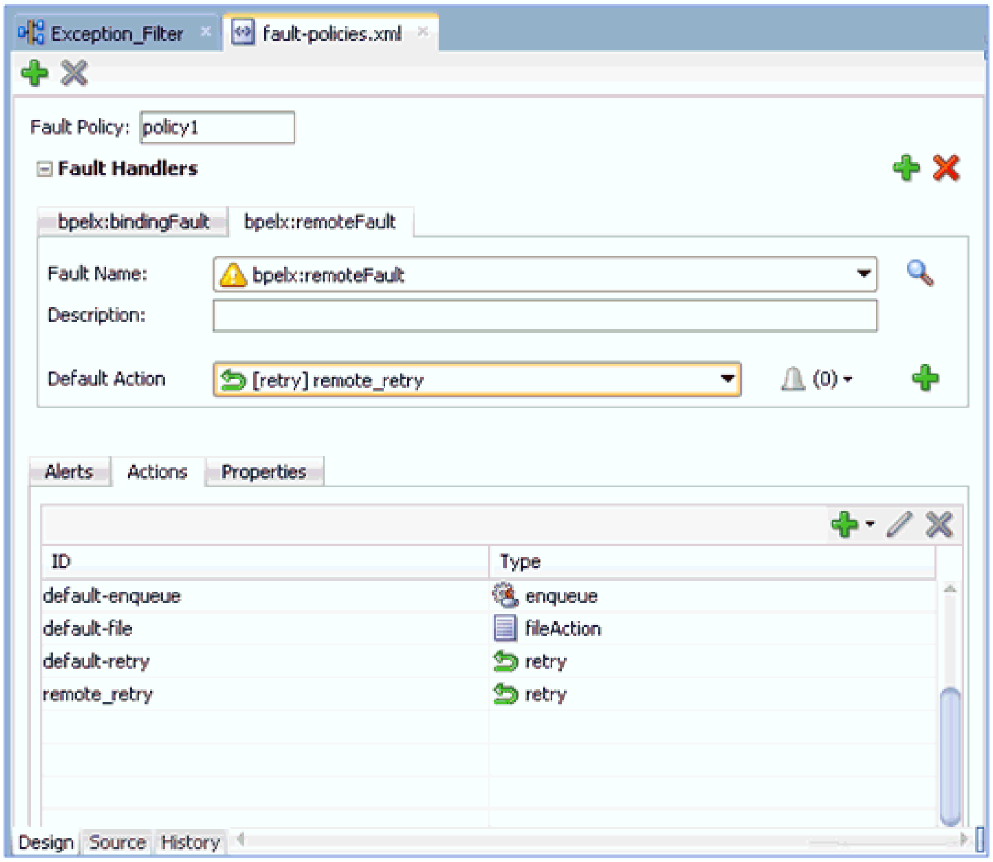
11. Select **[abort] default-termination** from the Retry Success Action drop-down list and **[humanIntervention] default-human** from the Retry Failure Action drop-down list.

12. Click **OK**.

The created Retry ID will be listed under the Actions tab.

From the Default Action drop-down list, select the newly created Retry ID (for example, remote_retry) as shown in Figure 9–83.

Figure 9–83 Fault-policies.xml Tab



Fault Policy: policy1

Fault Name: bpelx:remoteFault

Description:

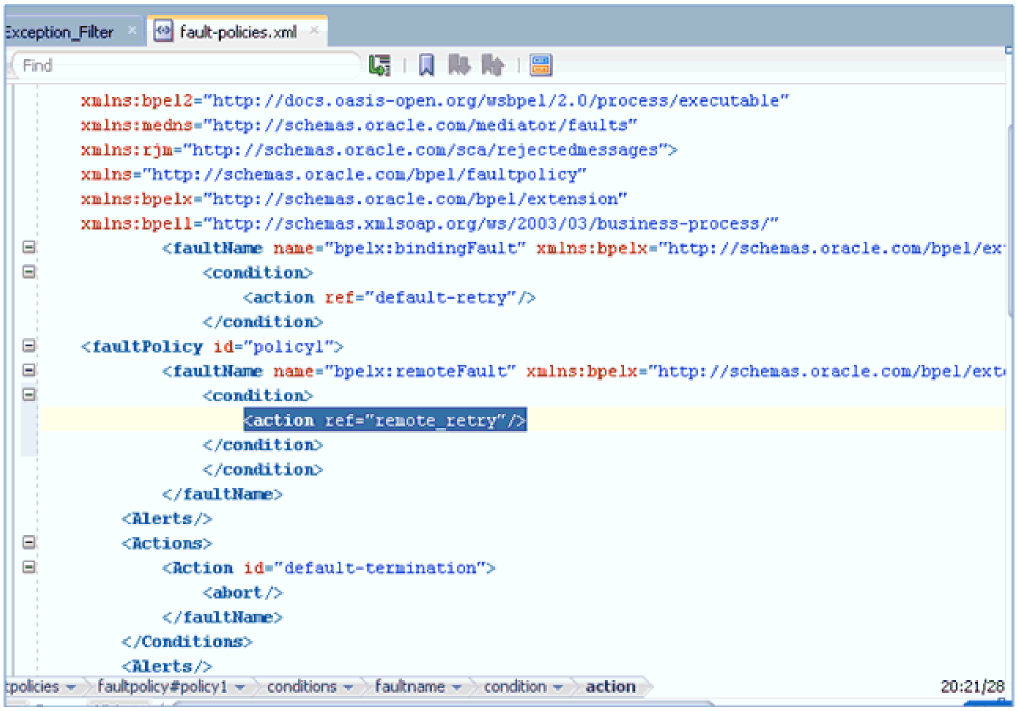
Default Action [retry] remote_retry

ID	Type
default-enqueue	enqueue
default-file	fileAction
default-retry	retry
remote_retry	retry

13. Click **Save All**.

14. Click the **Source** tab to verify that the fault polices are added properly, as shown in Figure 9–84.

Figure 9–84 Source Tab

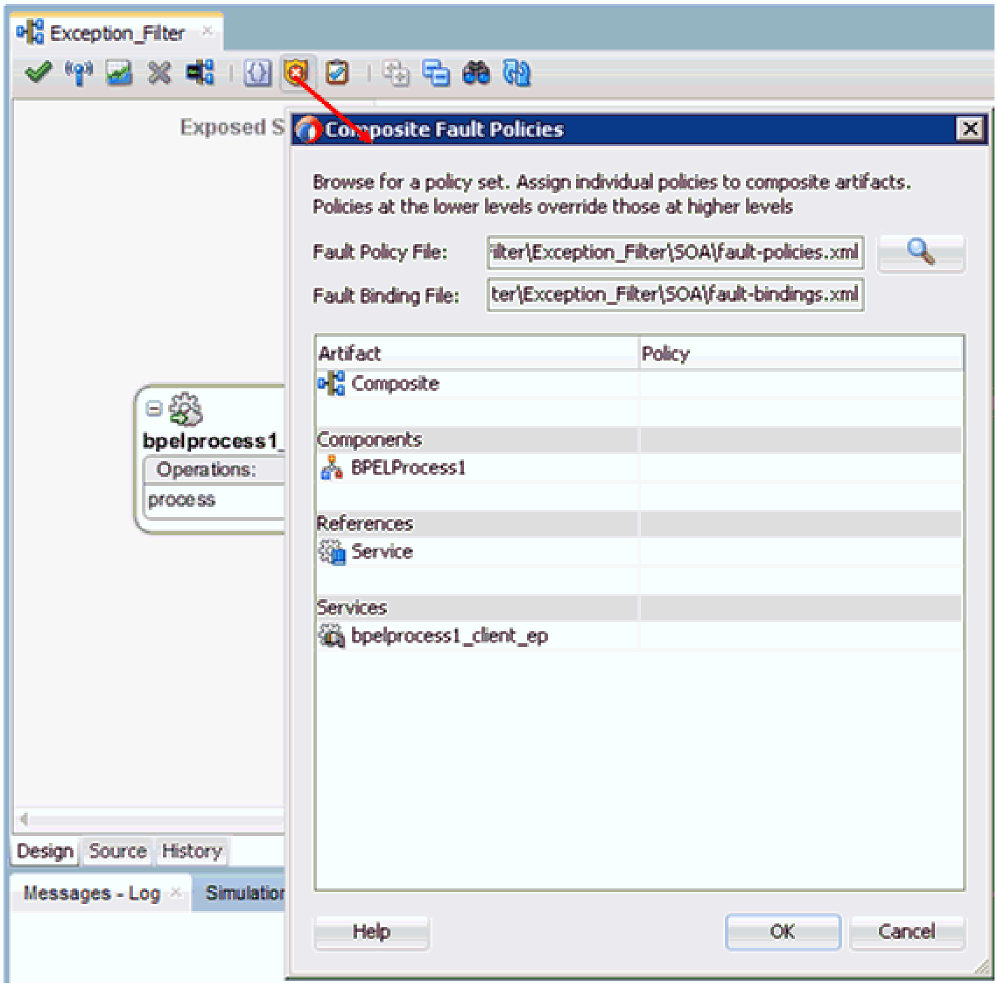


```
xmlns:bpel2="http://docs.oasis-open.org/wsbpel/2.0/process/executable"
xmlns:medns="http://schemas.oracle.com/mediator/faults"
xmlns:rjm="http://schemas.oracle.com/sca/rejectedmessages">
xmlns="http://schemas.oracle.com/bpel/faultpolicy"
xmlns:bpelx="http://schemas.oracle.com/bpel/extension"
xmlns:bpel1="http://schemas.xmlsoap.org/ws/2003/03/business-process/"
        <faultName name="bpelx:bindingFault" xmlns:bpelx="http://schemas.oracle.com/bpel/ex
            <condition>
                <action ref="default-retry"/>
            </condition>
<faultPolicy id="policy1">
        <faultName name="bpelx:remoteFault" xmlns:bpelx="http://schemas.oracle.com/bpel/ext
            <condition>
                <action ref="remote_retry"/>
            </condition>
            </condition>
        </faultName>
    <Alerts/>
    <Actions>
        <Action id="default-termination">
            <abort/>
        </faultName>
    </Conditions>
    <Alerts/>
```

15. Double-click the **Exception_Filter** project and then click **Edit Composite Fault Policies**.

    The Composite Fault Policies window is displayed. Ensure that the Fault Policy and the fault-bindings are selected properly, as shown in Figure 9–85.

Figure 9–85 Composite Fault Policies Window



Browse for a policy set. Assign individual policies to composite artifacts. Policies at the lower levels override those at higher levels

Fault Policy File: ilter\Exception_Filter\SOA\fault-policies.xml

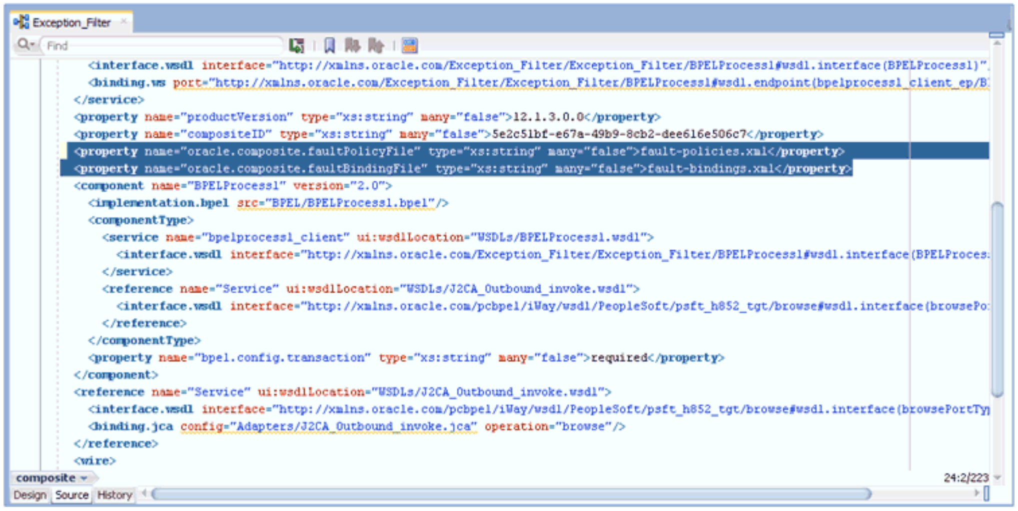
Fault Binding File: ter\Exception_Filter\SOA\fault-bindings.xml

Artifact	Policy
Composite	
Components	
BPELProcess1	
References	
Service	
Services	
bpelprocess1_client_ep	

16. Click **Save All**.

17. Click the **Source** tab to verify that the `fault-bindings.xml` and `fault-policies.xml` files are added properly, as shown in Figure 9–86.

Figure 9–86 Source Tab



```
    <interface.wsdl interface="http://xmlns.oracle.com/Exception_Filter/Exception_Filter/BPELProcess1#wsdl.interface(BPELProcess1)"
    <binding.ws port="http://xmlns.oracle.com/Exception_Filter/Exception_Filter/BPELProcess1#wsdl.endpoint(bpelprocess1_client_ep/B
  </service>
  <property name="productVersion" type="xs:string" many="false">12.1.3.0.0</property>
  <property name="compositeID" type="xs:string" many="false">5e2c51bf-e67a-49b9-8cb2-dee616e506c7</property>
  <property name="oracle.composite.faultPolicyFile" type="xs:string" many="false">fault-policies.xml</property>
  <property name="oracle.composite.faultBindingFile" type="xs:string" many="false">fault-bindings.xml</property>
  <component name="BPELProcess1" version="2.0">
    <implementation.bpel src="BPEL/BPELProcess1.bpel"/>
    <componentType>
      <service name="bpelprocess1_client" ui:wsdlLocation="WSDLs/BPELProcess1.wsdl">
        <interface.wsdl interface="http://xmlns.oracle.com/Exception_Filter/Exception_Filter/BPELProcess1#wsdl.interface(BPELProces
      </service>
      <reference name="Service" ui:wsdlLocation="WSDLs/J2CA_Outbound_invoke.wsdl">
        <interface.wsdl interface="http://xmlns.oracle.com/pcbpel/iWay/wsdl/PeopleSoft/psft_h852_tgt/browse#wsdl.interface(browsePo
      </reference>
    </componentType>
    <property name="bpel.config.transaction" type="xs:string" many="false">required</property>
  </component>
  <reference name="Service" ui:wsdlLocation="WSDLs/J2CA_Outbound_invoke.wsdl">
    <interface.wsdl interface="http://xmlns.oracle.com/pcbpel/iWay/wsdl/PeopleSoft/psft_h852_tgt/browse#wsdl.interface(browsePortTy
    <binding.jca config="Adapters/J2CA_Outbound_invoke.jca" operation="browse"/>
  </reference>
  <wire>
```

9.4.1.4 Adjusting for Known Deployment Issues With 12*c*

For more information on how to adjust for known deployment issues with 12c, see Section 4.4.3.3, "Adjusting for Known Deployment Issues With 12c" on page 4-26.

9.4.1.5 Deploying and Testing the BPEL Process With Exception Filter Functionality

To deploy and test the BPEL process with exception filter functionality:

1. Deploy the created BPEL process.

   For more information, see "Deploying the BPEL Outbound Process" on page 4-28.

2. Simulate a communication error by disconnecting the system (where the servers are running) from the network.

3. Invoke the deployed BPEL process with a valid input.

   For more information, see "Invoking the Input XML Document in the Oracle Enterprise Manager Console" on page 4-31.

4. Select the process ID.

   You can observe the BPEL process being retried or aborted based on the configuration of the **fault-policies.xml** file.

9.5 Credential Mapping for Oracle SOA Suite (BPEL, Mediator, or BPM)

This section describes how to configure credential mapping functionality for the Oracle Application Adapter for Siebel in a configuration that uses Oracle SOA Suite (BPEL, Mediator, or BPM). A sample testing scenario is also included. This section contains the following topic:

- Section 9.5.1, "Configuring Credential Mapping"

Credential mapping is supported only for outbound processes that use J2CA configurations. This feature is not supported for BSE configurations and inbound processes that use J2CA configurations.

Note: The J2CA connector is common to all four application adapters (SAP R/3, PeopleSoft, Siebel, and J.D. Edwards OneWorld). If credential mapping is required, then ensure that only one application adapter is used in a particular instance. For example, in one adapter instance only the Siebel application adapter can be used. Credential mapping cannot be configured at the individual adapter level. If you require the use of credential mapping for two adapters, then both adapters must be running in two independent adapter instances.

To pass user credentials to the J2CA resource adapter, create a credential map from the Oracle WebLogic Server user credentials to the EIS user credentials (Siebel adapter). Then associate a credential policy with a BPEL, Mediator, or BPM Web service and invoke the Web service using Oracle WebLogic Server user credentials. These credentials are mapped to the EIS user credentials and then passed to the J2CA container, which uses them to connect with the EIS adapter (Siebel).

9.5.1 Configuring Credential Mapping

This section discusses configuring credential mapping, and consists of the following steps and topics:

1. Deploy the adapter.

   For more information, see Chapter 3, "Oracle WebLogic Server Deployment and Integration".

2. Associate Oracle WebLogic Server credentials with EIS credentials.

   For more information, see Section 9.5.1.1, "Associating Oracle WebLogic Server Credentials With EIS Credentials" on page 9-56.

3. Generate a WSDL file.

   For more information, see Section 9.5.1.2, "Generating a WSDL File" on page 9-59.

4. Create and deploy an outbound process.

   For more information, see Section 9.5.1.3, "Creating and Deploying an Outbound Process" on page 9-59.

5. Invoke and verify that the EIS credentials have passed.

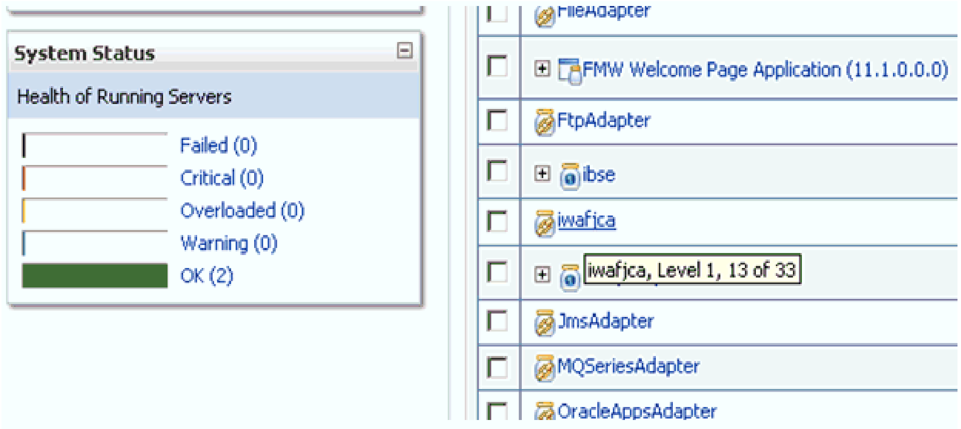
   For more information, see Section 9.5.1.4, "Verifying the EIS Credentials" on page 9-60.

9.5.1.1 Associating Oracle WebLogic Server Credentials With EIS Credentials

To associate Oracle WebLogic Server credentials with EIS credentials:

1. Log in to the Oracle WebLogic Server Administration Console.
2. In the Domain Structure section in the left pane, click **Deployments**, as shown in Figure 9–87.

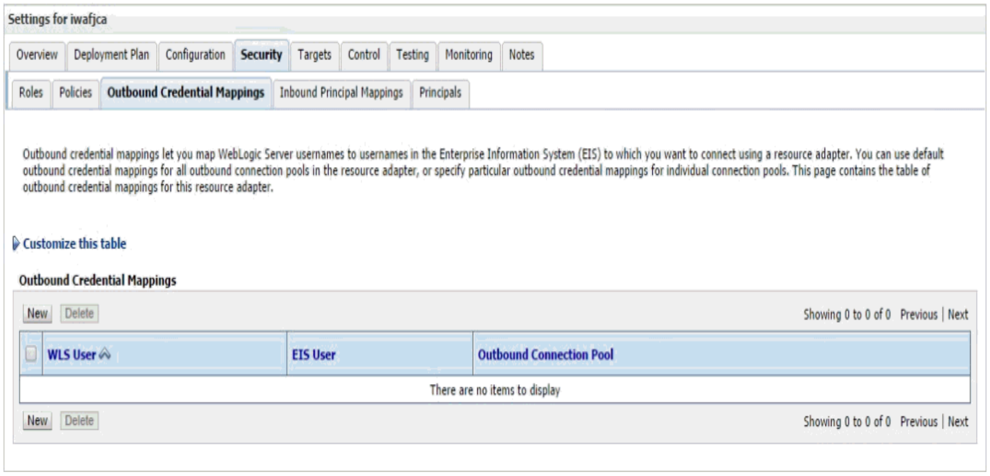
Figure 9–87 Deployments Page



3. Click the **iwafjca** resource adapter.

   The Settings for iwafjca page is displayed, as shown in Figure 9–88.

Figure 9–88 Settings for iwafjca Page



Outbound credential mappings let you map WebLogic Server usernames to usernames in the Enterprise Information System (EIS) to which you want to connect using a resource adapter. You can use default outbound credential mappings for all outbound connection pools in the resource adapter, or specify particular outbound credential mappings for individual connection pools. This page contains the table of outbound credential mappings for this resource adapter.

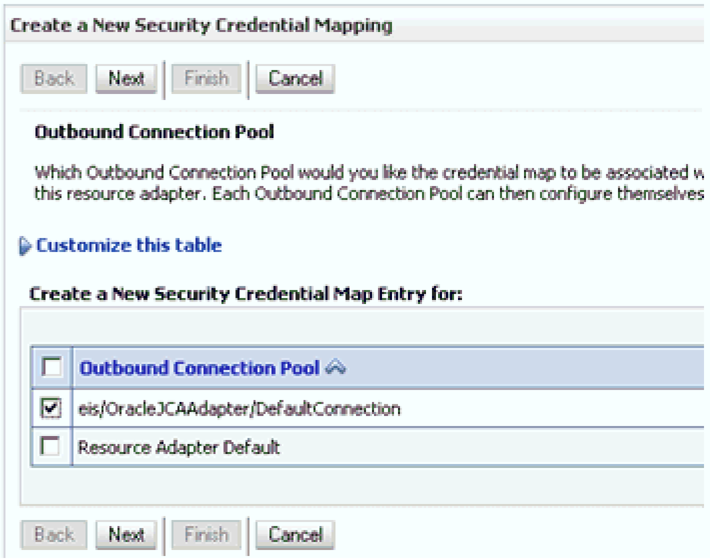
Outbound Credential Mappings

□	WLS User	EIS User	Outbound Connection Pool
There are no items to display			

4. Click the **Outbound Credential Mappings** tab under the Security tab, and then click **New**.

   The Create a New Security Credential Mapping page is displayed, as shown in Figure 9–89.

Figure 9–89 Create a New Security Credential Mapping Page



Create a New Security Credential Mapping

Outbound Connection Pool

Which Outbound Connection Pool would you like the credential map to be associated w
this resource adapter. Each Outbound Connection Pool can then configure themselves

Create a New Security Credential Map Entry for:

□	Outbound Connection Pool
☑	eis/OracleJCAAdapter/DefaultConnection
□	Resource Adapter Default

5. Select the outbound connection pool.

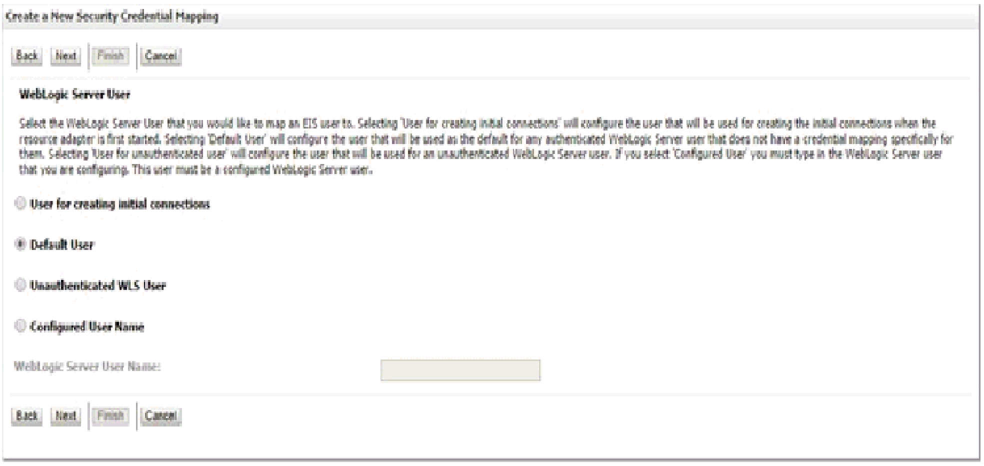
   For example:

   ```
   eis/OracleJCAAdapter/DefaultConnection
   ```

6. Click **Next**.

   The WebLogic Server User page is displayed, as shown in Figure 9–90.

Figure 9–90 WebLogic Server User Page



Create a New Security Credential Mapping

WebLogic Server User

Select the WebLogic Server User that you would like to map an EIS user to. Selecting 'User for creating initial connections' will configure the user that will be used for creating the initial connections when the resource adapter is first started. Selecting 'Default User' will configure the user that will be used as the default for any authenticated WebLogic Server user that does not have a credential mapping specifically for them. Selecting 'User for unauthenticated user' will configure the user that will be used for an unauthenticated WebLogic Server user. If you select 'Configured User' you must type in the WebLogic Server user that you are configuring. This user must be a configured WebLogic Server user.

○ User for creating initial connections

● Default User

○ Unauthenticated WLS User

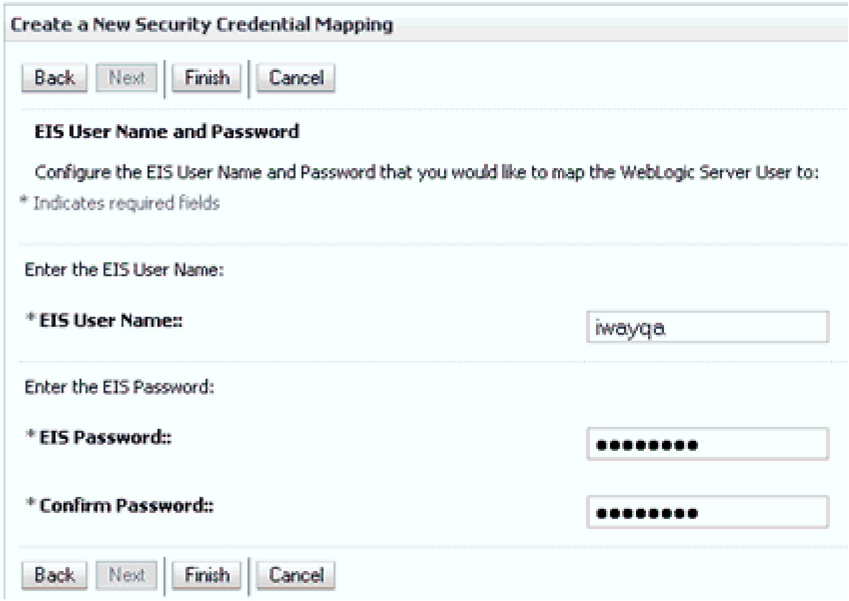
○ Configured User Name

WebLogic Server User Name:

7. Select **Default User**, enter a valid Oracle WebLogic Server user name, and then click **Next**.

   The EIS User Name and Password page is displayed, as shown in Figure 9–91.

Figure 9–91 EIS User Name and Password Page



EIS User Name and Password

Configure the EIS User Name and Password that you would like to map the WebLogic Server User to:

* Indicates required fields

Enter the EIS User Name:

* **EIS User Name::** iwayqa

Enter the EIS Password:

* **EIS Password::** ●●●●●●●●

* **Confirm Password::** ●●●●●●●●

8. Enter the user name and password for the EIS and click **Finish**.

   The credentials for an Oracle WebLogic Server user are now mapped with an EIS user (Siebel). The mapping is invoked automatically before invoking the J2CA service.

9.5.1.2 Generating a WSDL File

To generate a WSDL file:

1. Open Application Explorer and create a J2CA configuration.

   For more information, see Section 2.3.2, "Creating a Configuration for J2CA" on page 2-4.

2. Create a target for the Siebel adapter and then connect to the target.

   For more information, see Section 2.4, "Establishing a Connection (Target) for Siebel" on page 2-5.

3. Generate a WSDL for the appropriate object.

   For more information, see Section 2.12, "Generating WSDL (J2CA Configurations Only)" on page 2-27.

9.5.1.3 Creating and Deploying an Outbound Process

This section describes how to configure an outbound process. For demonstration purposes, specific references to the BPEL outbound process are made. However, the same steps apply to Mediator and BPM outbound processes.

For more information about creating a Mediator outbound process, see Chapter 5, "Integration With Mediator Service Components in the Oracle SOA Suite".

For more information about creating a BPM outbound process, see Chapter 6, "Integration With BPM Service Components in the Oracle SOA Suite".

To create a BPEL outbound process, see the following sections:

- Section 4.4.2, "Creating an Empty Composite for SOA"

- Section 4.4.3, "Defining a BPEL Outbound Process"
- Section 4.4.4, "Deploying the BPEL Outbound Process"

9.5.1.4 Verifying the EIS Credentials

Invoke the input XML and ensure that the EIS target credentials are overridden with the credentials configured in the WebLogic Administration Console for the Default User as described in this section.

1. Invoke the deployed BPEL outbound process with a valid input.

   For more information, see Section 4.4.5, "Invoking the Input XML Document in the Oracle Enterprise Manager Console" on page 4-31.

2. Check the J2CA log files and locate the encrypted password, which shows that the user credentials have been passed to the EIS through Oracle WebLogic Server.

   For example:

   ```
   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 Credential Mapping for Oracle Service Bus (OSB) Using JDeveloper

This section describes how to configure credential mapping functionality for the Oracle Application Adapter for Siebel in a configuration that uses Oracle Service Bus (OSB). A sample testing scenario is also included. This section contains the following topic:

- Section 9.6.1, "Configuring Credential Mapping"

Credential mapping is supported only for outbound processes that use J2CA configurations. This feature is not supported for BSE configurations and inbound processes that use J2CA configurations.

> **Note:** The J2CA connector is common to all four application adapters (SAP R/3, PeopleSoft, Siebel, and J.D. Edwards OneWorld). If credential mapping is required, then ensure that only one application adapter is used in a particular instance. For example, in one adapter instance only the Siebel application adapter can be used. Credential mapping cannot be configured at the individual adapter level. If you require the use of credential mapping for two adapters, then both adapters must be running in two independent adapter instances.

To pass user credentials to the J2CA resource adapter, create a credential map from the Oracle WebLogic Server user credentials to the EIS user credentials (Siebel adapter). Then associate a credential policy with a Web service and invoke the Web service using Oracle WebLogic Server user credentials. These credentials are mapped to the EIS user credentials and then passed to the J2CA container, which uses them to connect with the EIS adapter (Siebel).

9.6.1 Configuring Credential Mapping

Configuring credential mapping consists of the following steps and topics:

1. Deploy the adapter.

   For more information, see Chapter 3, "Oracle WebLogic Server Deployment and Integration".

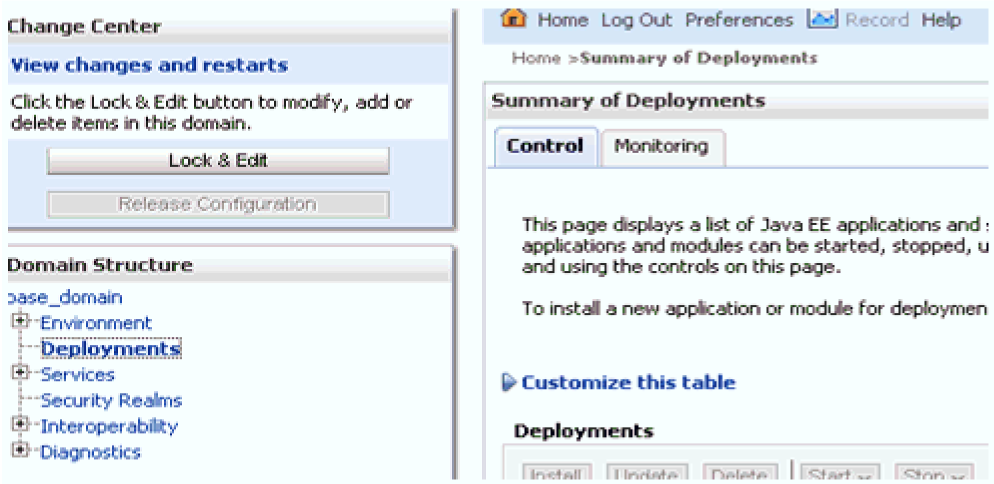
2. Section 9.6.1.1, "Associating Oracle WebLogic Server Credentials With EIS Credentials"
3. Section 9.6.1.2, "Generating a WSDL File"
4. Section 9.6.1.3, "Creating an Oracle Service Bus (OSB) Outbound Process"

9.6.1.1 Associating Oracle WebLogic Server Credentials With EIS Credentials

To associate Oracle WebLogic Server credentials with EIS credentials:

1. Log in to the Oracle WebLogic Server Administration Console.
2. In the Domain Structure section in the left pane, click **Deployments**, as shown in Figure 9–92.

Figure 9–92 Domain Structure Section



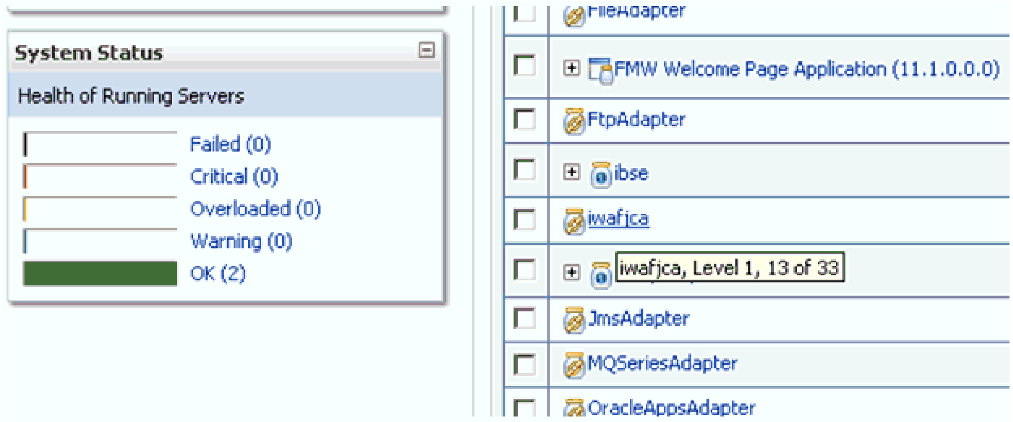
Click the Lock & Edit button to modify, add or delete items in this domain.

This page displays a list of Java EE applications and : applications and modules can be started, stopped, u and using the controls on this page.

To install a new application or module for deploymen

The Deployments page is displayed, as shown in Figure 9–93.

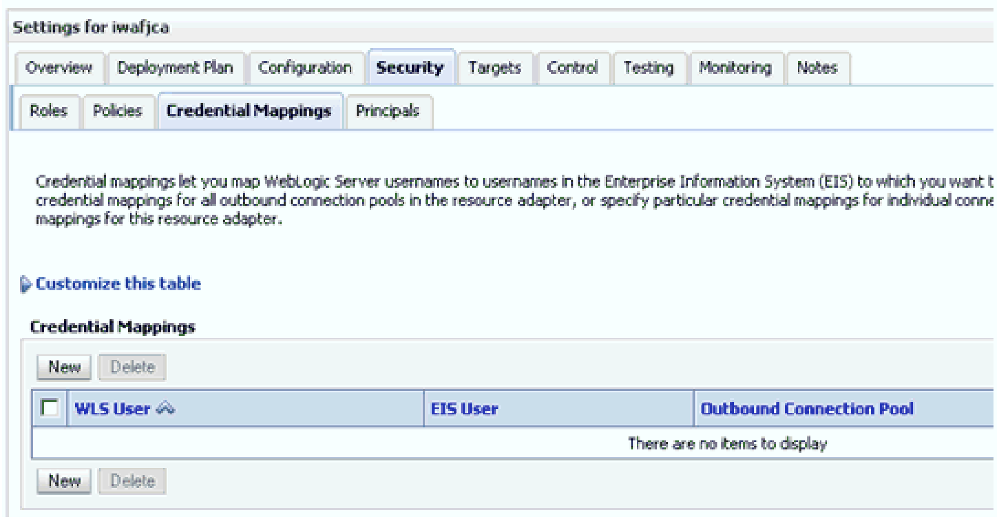
Figure 9–93 Deployments Page



3. Click the **iwafjca** resource adapter.

   The Settings for iwafjca page is displayed, as shown in Figure 9–94.

Figure 9–94 Settings for iwafjca Page



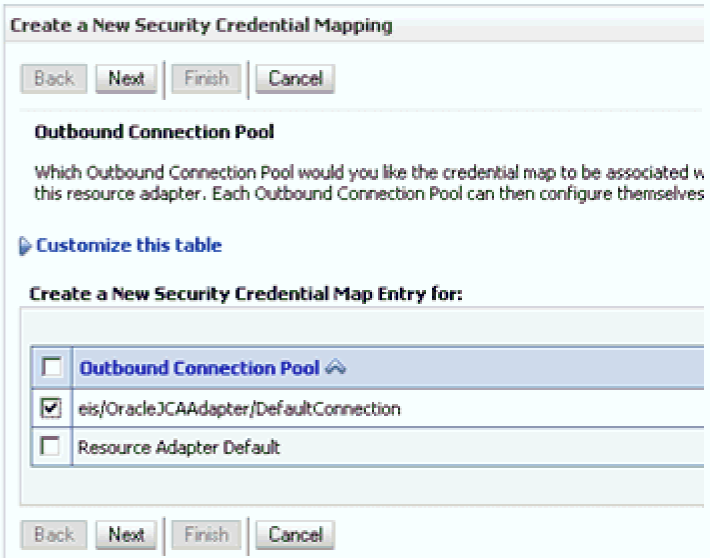
Credential mappings let you map WebLogic Server usernames to usernames in the Enterprise Information System (EIS) to which you want t credential mappings for all outbound connection pools in the resource adapter, or specify particular credential mappings for individual conne mappings for this resource adapter.

□	WLS User	EIS User	Outbound Connection Pool
		There are no items to display	

4. Click the **Credential Mappings** tab under the Security tab, and then click **New**.

   The Create a New Security Credential Mapping page is displayed, as shown in Figure 9–95.

Figure 9–95 Create a New Security Credential Mapping Page



Create a New Security Credential Mapping

Outbound Connection Pool

Which Outbound Connection Pool would you like the credential map to be associated w
this resource adapter. Each Outbound Connection Pool can then configure themselves

Create a New Security Credential Map Entry for:

□	**Outbound Connection Pool**
☑	eis/OracleJCAAdapter/DefaultConnection
□	Resource Adapter Default

5. Select the outbound connection pool.

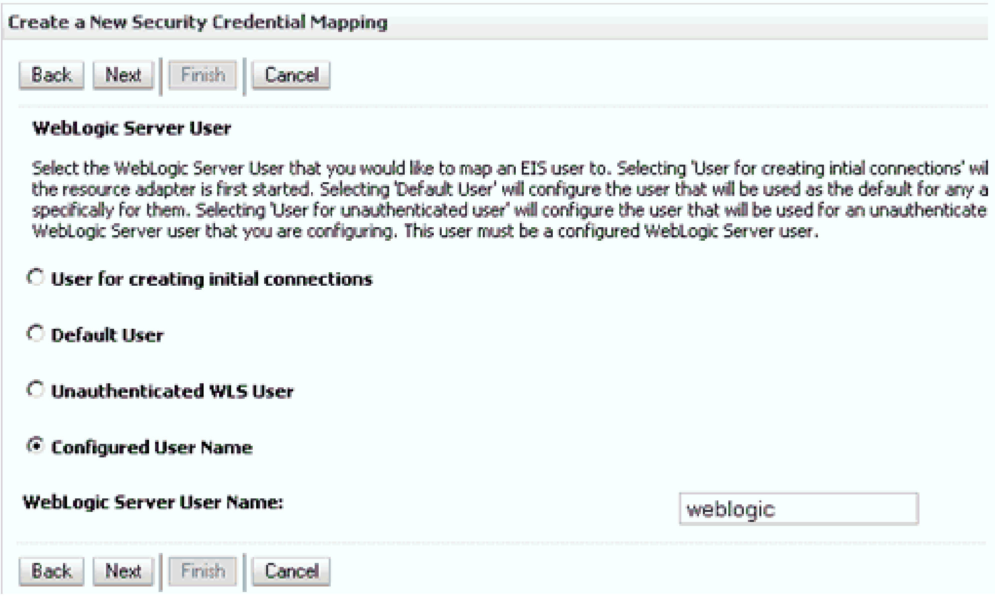
   For example:

   `eis/OracleJCAAdapter/DefaultConnection`

6. Click **Next**.

   The WebLogic Server User page is displayed, as shown in Figure 9–96.

Figure 9–96 WebLogic Server User Page



Create a New Security Credential Mapping

WebLogic Server User

Select the WebLogic Server User that you would like to map an EIS user to. Selecting 'User for creating intial connections' wil
the resource adapter is first started. Selecting 'Default User' will configure the user that will be used as the default for any a
specifically for them. Selecting 'User for unauthenticated user' will configure the user that will be used for an unauthenticate
WebLogic Server user that you are configuring. This user must be a configured WebLogic Server user.

○ **User for creating initial connections**

○ **Default User**

○ **Unauthenticated WLS User**

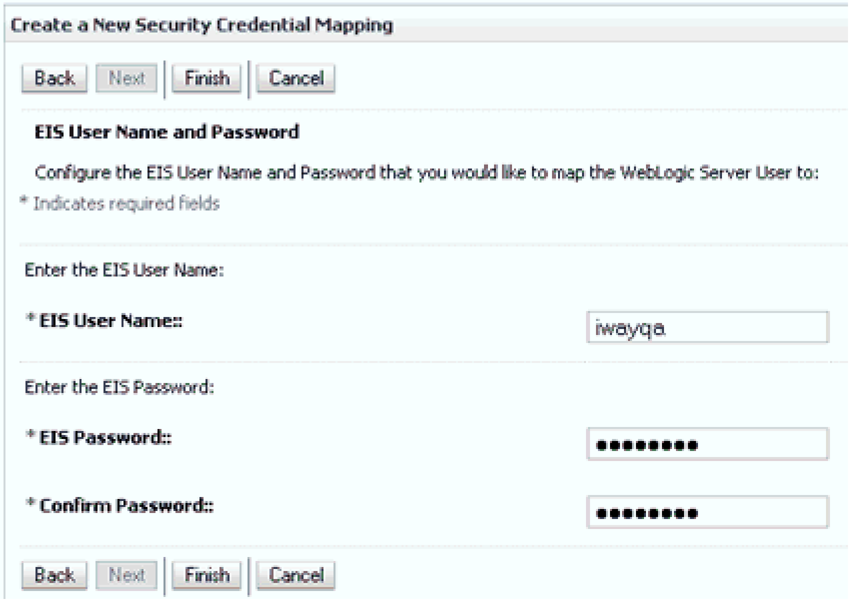
● **Configured User Name**

WebLogic Server User Name: weblogic

7. Select Configured User Name, enter a valid Oracle WebLogic Server user name, and then click **Next**.

   The EIS User Name and Password page is displayed, as shown in Figure 9–97.

Figure 9–97 EIS User Name and Password Page



Create a New Security Credential Mapping

EIS User Name and Password

Configure the EIS User Name and Password that you would like to map the WebLogic Server User to:

* Indicates required fields

Enter the EIS User Name:

* EIS User Name:: iwayqa

Enter the EIS Password:

* EIS Password:: ••••••••

* Confirm Password:: ••••••••

8. Enter the user name and password for the EIS and click **Finish**.

   The credentials for an Oracle WebLogic Server user are now mapped with an EIS user (Siebel). The mapping is invoked automatically before invoking the J2CA service.

9.6.1.2 Generating a WSDL File

To generate a WSDL file:

1. Set the class path for Application Explorer to integrate with Oracle Service Bus (OSB).

   For more information, see "Setting the Class Path for Application Explorer to Integrate With Oracle Service Bus" on page 7-6.

2. Open Application Explorer and create a J2CA configuration.

   For more information, see "Creating a Configuration for J2CA" on page 2-4.

3. Create a target for the Siebel adapter and then connect to the target.

   For more information, see "Establishing a Connection (Target) for Siebel" on page 2-5.

4. Generate a WSDL for the appropriate object.

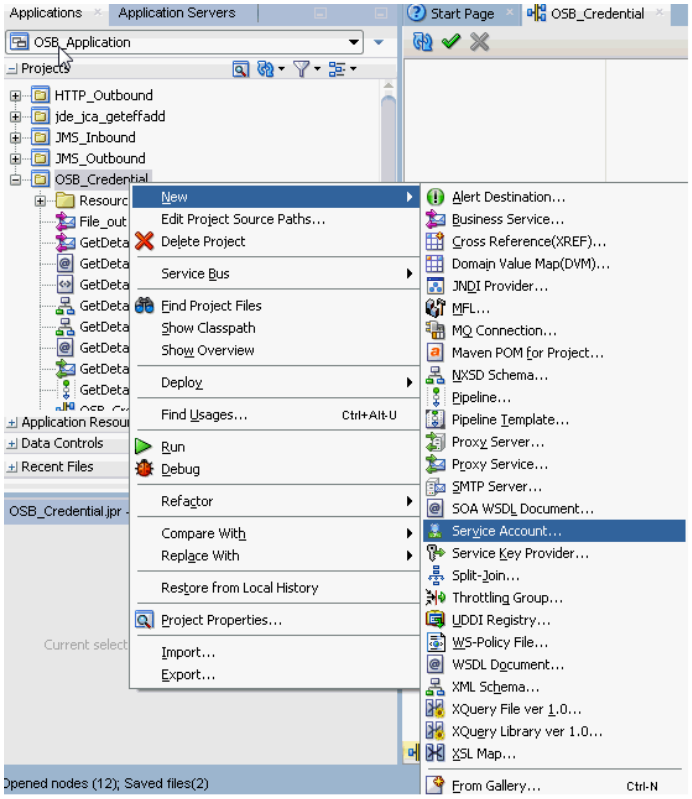
   For more information, see Section 4.4.1, "Generating WSDL for Request/Response Service" on page 4-8.

9.6.1.3 Creating an Oracle Service Bus (OSB) Outbound Process

For more information on creating an Oracle Service Bus (OSB) outbound process, see Section 8.1.2, "Defining an OSB Outbound Process" on page 8-3.

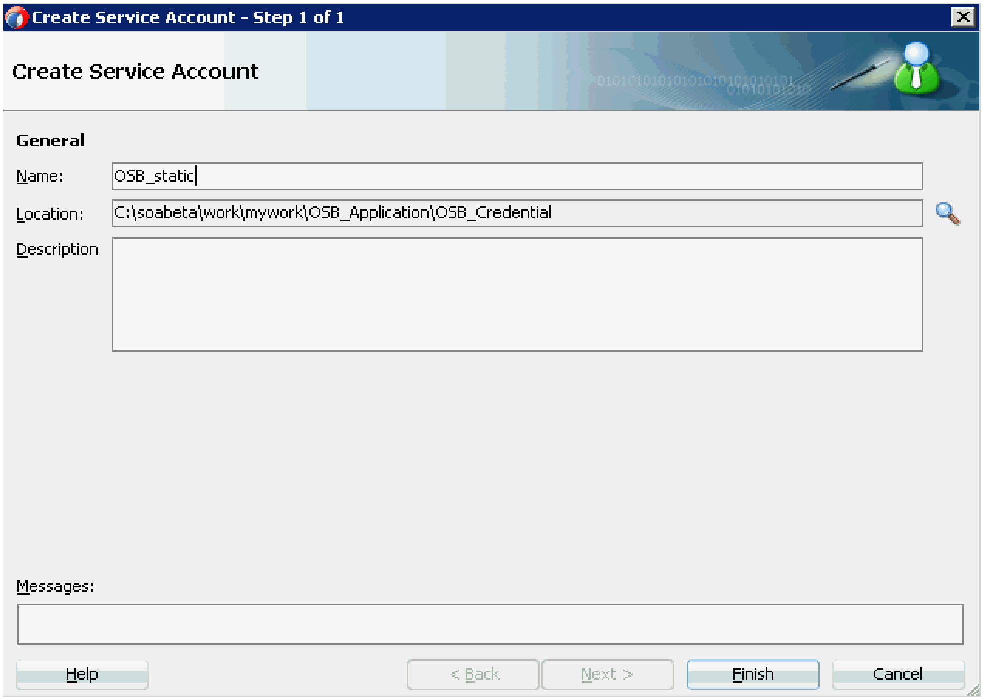
1. Configure a Service account by right-clicking the OSB Project, selecting **New**, and then clicking **Service Account**, as shown in Figure 9–98.

Figure 9–98 Select Service Account Option



The Create Service Account pane is displayed, as shown in Figure 9–99.

Figure 9–99 Create Service Account Pane



General

Name: OSB_static

Location: C:\soabeta\work\mywork\OSB_Application\OSB_Credential

Description

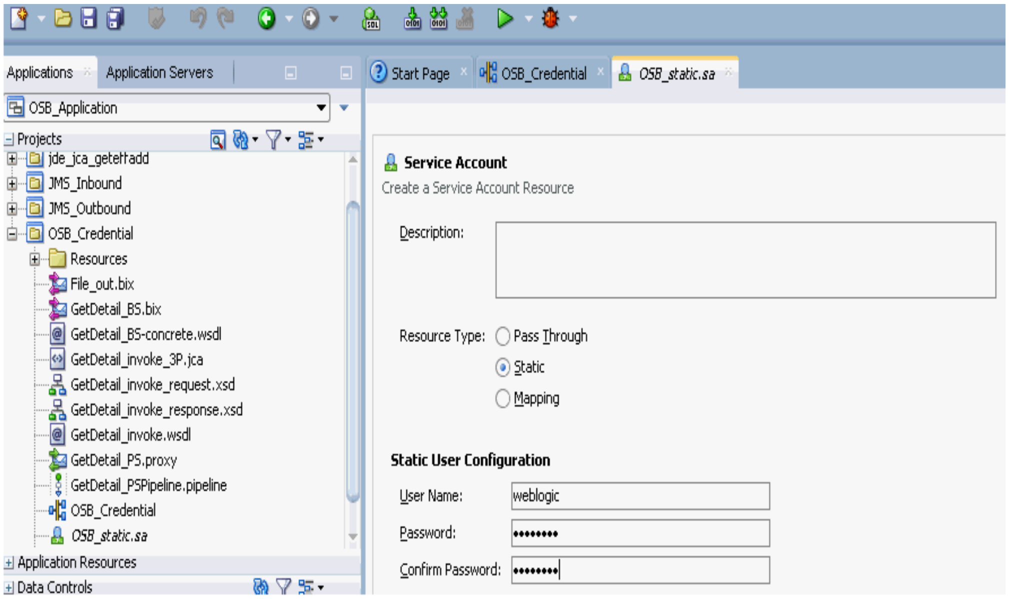
Messages:

2. Provide a name for the Service Account and click **Finish**.

The configuration page of Service Account is displayed.

3. In the Resource Type section, select **Static**.
4. Provide a valid user name and password for the Oracle WebLogic Server, as shown in Figure 9–100.

Figure 9–100 Service Account Configuration Page



Service Account

Create a Service Account Resource

Description:

Resource Type: ○ Pass Through
● Static
○ Mapping

Static User Configuration

User Name: weblogic

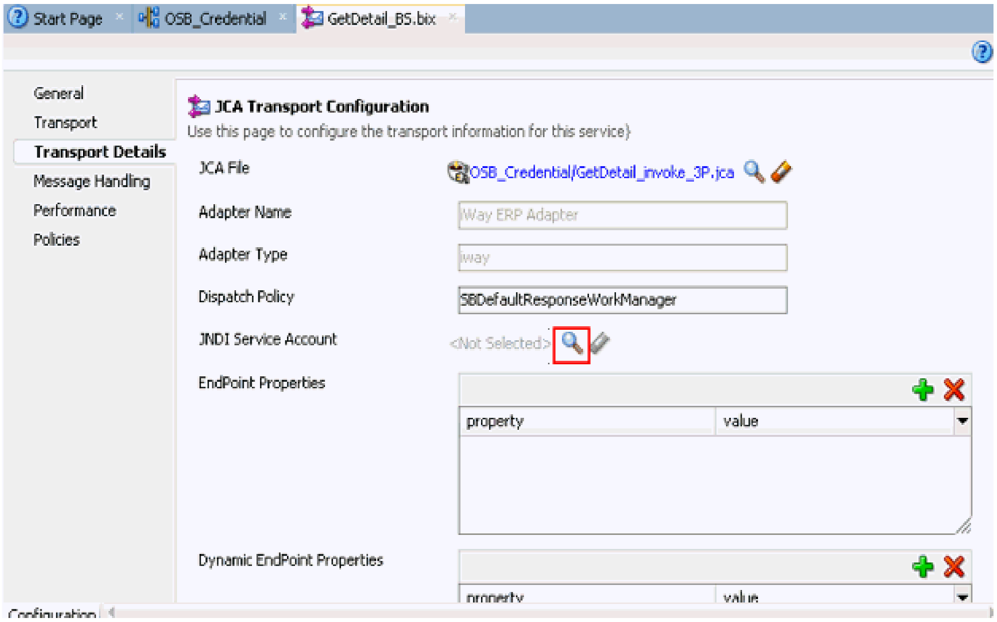
Password: ••••••••

Confirm Password: ••••••••

5. Save and close the configuration page.
6. In the composite Editor window, double-click the created WSDL-based Business Service from step 3.

   The configuration page of the WSDL-based Business Service is displayed.
7. Select the Transport Details tab, as shown in Figure 9–101.

Figure 9–101 Transport Details Tab



JCA Transport Configuration

Use this page to configure the transport information for this service)

JCA File: OSB_Credential/GetDetail_invoke_3P.jca

Adapter Name: iWay ERP Adapter

Adapter Type: iway

Dispatch Policy: SBDefaultResponseWorkManager

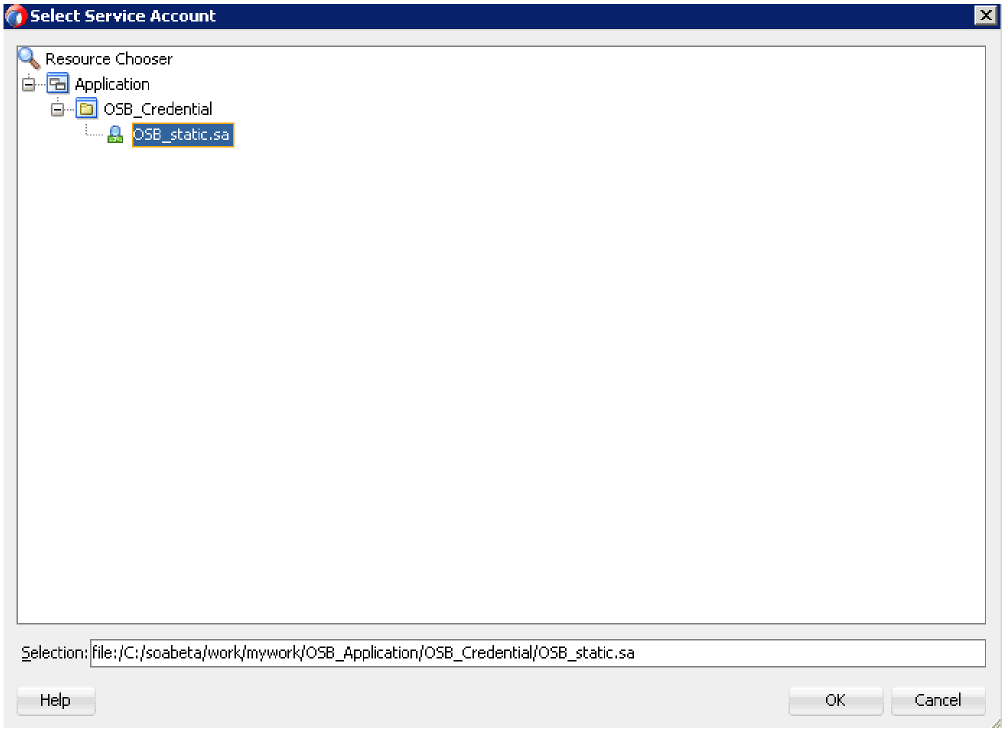
JNDI Service Account: <Not Selected>

8. In the JNDI Service Account section, click the Browse icon.

   The Select Service Account window is displayed.

9. Select the created service account and click **OK**, as shown in Figure 9–102.

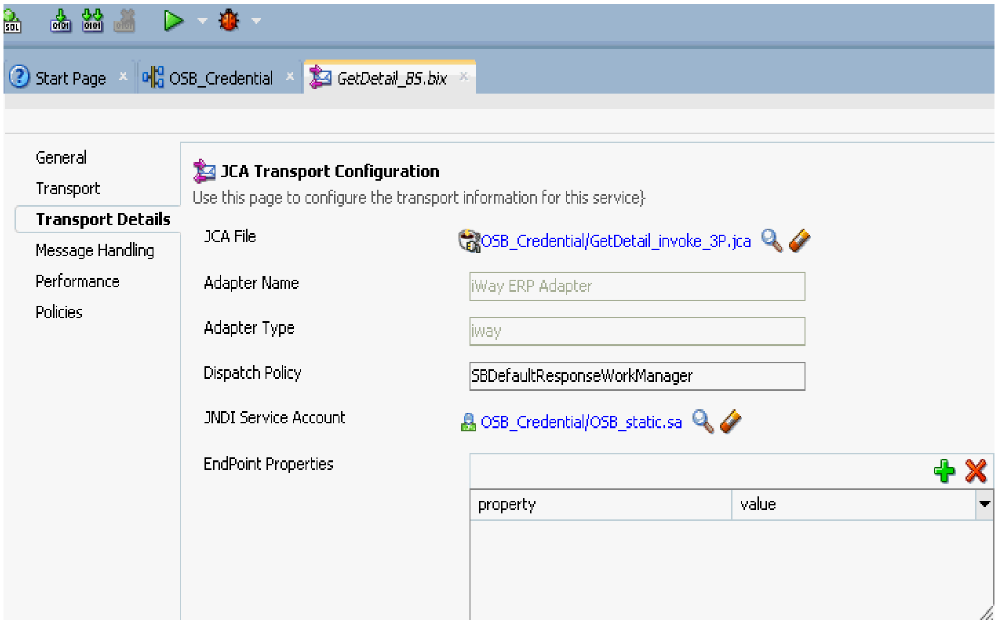
Figure 9–102 Select Service Account



Selection: file:/C:/soabeta/work/mywork/OSB_Application/OSB_Credential/OSB_static.sa

10. Save and close the configuration page, as shown in Figure 9–103

Figure 9–103 Business Service Configuration Page



JCA Transport Configuration

Use this page to configure the transport information for this service}

JCA File: OSB_Credential/GetDetail_invoke_3P.jca

Adapter Name: iWay ERP Adapter

Adapter Type: iway

Dispatch Policy: SBDefaultResponseWorkManager

JNDI Service Account: OSB_Credential/OSB_static.sa

EndPoint Properties

property	value

11. Deploy the OSB process.

    For more information, see Section 8.1.3, "Deploying the OSB Outbound Process" on page 8-16.

12. Once the process is deployed successfully, copy and paste a valid input XML file in the input folder you configured, and check to see that the output is received in the configured output location.

13. Check the J2CA log files and locate the encrypted password, which shows that the user credentials have been passed to the EIS through Oracle WebLogic Server.

    For example:

```
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)
```

10 Troubleshooting and Error Messages

This chapter explains the limitations and workarounds when connecting to Siebel. It contains the following topics:

- Section 10.1, "Troubleshooting"
- Section 10.2, "BSE Error Messages"

10.1 Troubleshooting

This topic provides troubleshooting information for Siebel, separated into four categories:

- Section 10.1.1, "General Usage Notes for the Oracle Application Adapter for Siebel"
- Section 10.1.2, "Application Explorer"
- Section 10.1.3, "Siebel"
- Section 10.1.4, "Oracle Adapter J2CA"

Log file information that can be relevant in troubleshooting can be found in the following locations based on your adapter installation:

- The Oracle Adapter J2CA trace information can be found under the following directory:

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

- BSE trace information can be found under the following directory:

  ```
  <ORACLE_HOME>\user_projects\domains\base_domain\servers\soa_
  server1\stage\ibse\ibse.war\ibselogs
  ```

- The log file for Application Explorer can be found under the following directory:

  ```
  <ADAPTER_HOME>\tools\iwae\bin
  ```

10.1.1 General Usage Notes for the Oracle Application Adapter for Siebel

The Oracle Application Adapter for Siebel is subject to the following limitations:

- The HTTPS protocol is not supported for services and events.
- Updates for multi-value (MVG) fields with join specifications are not supported.
- When a connection is lost, the adapter does not automatically reconnect to Siebel.

10.1.2 Application Explorer

This topic discusses the different types of errors that can occur when using Application Explorer.

Error	Solution
Siebel does not appear in the Application Explorer Adapter node list.	Ensure that the Siebel jar files supplied with your Siebel distribution media have been placed in the `<ADAPTER_HOME>\Oracle_SOA1\soa\thirdparty\ApplicationAdapters\lib` directory. For example, for Siebel 7.03 environments, the `SiebelJI_Common.jar` and `SiebelJI_enu.jar` should be placed in this directory.
Target Type drop down contains only Java Data Bean Connection and COM connection type is desired.	Ensure that the Siebel thin client is installed correctly on the system hosting Application Explorer so that appropriate COM environment is available.
An error message that includes the name of the Siebel Gateway server appears when you try to connect to a Siebel target. For example, `Problem activating adapter (<server_name>). Check logs for more information.`	Ensure that the name of the Siebel Gateway server is correctly defined for the target you are using.
You receive the following error when trying to connect to a Siebel target: `Problem activating adapter. (You have entered an invalid set of logon parameters. Please type in your logon parameters again.). Check logs for more information.`	Ensure that the User ID and password parameter values to connect to your Siebel system are correct.
You receive the following error when trying to connect to a Siebel target: `Problem activating adapter. (Couldn't get nameserver connection). Check logs for more information.`	Check on network connectivity to Siebel environment. Correct networking problem and retry connection.
You receive the following error when attempting to connect to a Siebel target: `Problem activating adapter. (NSReadKey request failed (no error information)...). Check logs for more information.`	Ensure that the values defined for Siebel Server, Enterprise Name, and Object Manager for the target you are using are correct, and retry the connection
You receive the following error when attempting to connect to a Siebel target: `Problem activating adapter. (Error loading translatable messages: com.siebel.locale.enux.messages.SSAMessages_enux). Check logs for more information`	Ensure that the value of the Language parameter on the Advanced tab is defined correctly for the target you are using to connect to your Siebel system (for example, `enu` for English).

Error	Solution
A successful connection is made to Siebel environment but no values are available in Business Object, Business Service, and Integration Object nodes in Application Explorer tree.	The Repository Name specified on the Advanced tab in the Siebel target configuration is either void or empty of any components in the targeted Siebel environment or that Repository Name is not valid for the targeted Siebel environment. Verify that the Repository Name is valid and contains components for interrogation then re-connect.
Logon failure error at run-time.	If the password for connecting to your Siebel system is not specified when creating a target or with the Edit option in Application Explorer, then you are unable to connect to Siebel. The connection password is not saved in `repository.xml`. Update the password using the Edit option in Application Explorer, then restart the application server.
The following exception occurs when you start Application Explorer by activating `ae.bat` (not `iaexplorer.exe`): `java.lang.ClassNotFoundException: org.bouncycastle.jce.provider.BouncyCastleProvider`	This is a benign exception. It does not affect adapter functionality. Download BouncyCastle files from: `ftp://ftp.bouncycastle.org/pub`

Unable to start Application Explorer in a Solaris environment. The following exception is thrown in the console:
`javax.resource.ResourceException: IWAFManagedConnectionFactory: License violation.at com.ibi.afjca.spi.IWAFManagedConnectionFactory.createConnectionFactory(IWAFManagedConnectionFactory.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)`
`Could not create the connection factory.`
	`JAVACMD` is not set on the user system. Before starting Application Explorer, export `JAVACMD` as follows:
	`JAVACMD=/<jdk_home>/bin/java`, where `<jdk_home>` is the directory where JDK is installed on your system.

10.1.3 Siebel

The error messages listed can occur when using the adapter with either a BSE or Oracle Adapter J2CA repository project.

Error	Solution
A successful connection is made to Siebel environment but no values are available in Business Object, Business Service, and Integration Object nodes in Application Explorer tree.	The Repository Name specified on the Advanced tab in the Siebel Target configuration is either void or empty of any components in the targeted Siebel environment or that Repository Name is not valid for the targeted Siebel environment. Verify that the Repository Name is valid and contains components for interrogation then re-connect.
When executing a request, the following error message appears: `AdapterException: Unsupported Action: {0} Tquery`	Verify that method is available for specific request by verifying schema.
When executing a request, the following error message appears: `AdapterException: Field 'NFame' does not exist in definition for business component 'Account'. Please ask your systems administrator to check your application configuration.`	Ensure that field names are valid within request document by referring to schema for that specific object, and then re-submit the request.
When connecting to releases before Siebel 7.7 using the Java Data Bean Interface, you cannot reconnect after initial connection loss. This might occur when Application Explorer experiences a brief loss of network connection or if the Siebel Server or Gateway Service is restarted while Application Explorer is logged into the Siebel application.	Restart Oracle WebLogic Server and Application Explorer to log in successfully to the Siebel application. This is a known Siebel API issue. For more information, see Siebel Alert 984.
The following error may occur when adding a service node for a Business Service that includes methods containing method arguments having hierarchy data types. If you enter a valid `XMLCharEncoding` value such as UTF-8 or UTF-16, then the following error is received: `Invocation of Service failed.`	The method argument `XMLCharEncoding` is not supported. Leave this element blank in the XML payload.

10.1.4 Oracle Adapter J2CA

Error	Solution
In Application Explorer, the following error message appears when you attempt to connect to an Oracle Adapter J2CA configuration: `Could not initialize JCA`	In the Details tab in the right pane, ensure that the directory specified in the Home field points to the correct directory, for example: *<ADAPTER_HOME>*`\tools\iwae\bin\..\..\..\`

10.2 BSE Error Messages

This topic discusses the different types of errors that can occur when processing Web services through BSE.

This section contains the following topics:

- Section 10.2.1, "General Error Handling in BSE"
- Section 10.2.2, "Adapter-Specific Error Handling"

10.2.1 General Error Handling in BSE

BSE serves as both a SOAP gateway into the adapter framework and as the engine for some of the adapters. In both design time and run-time, various conditions can cause errors in BSE when Web services that use adapters run. Some of these conditions and resulting errors are exposed the same way, regardless of the specific adapter; others are exposed differently, based on the adapter being used. This topic explains what you can expect when you encounter some of the more common error conditions on an adapter-specific basis.

Usually the SOAP gateway (agent) inside BSE passes a SOAP request message to the adapter required for the Web service. If an error occurs, then how it is exposed depends on the adapter and the API or interfaces that the adapter uses. A few scenarios cause the SOAP gateway to generate a SOAP fault. In general, anytime the SOAP agent inside BSE receives an invalid SOAP request, a SOAP fault element is generated in the SOAP response. The SOAP fault element contains fault string and fault code elements. The fault code contains a description of the SOAP agent error.

The following SOAP response document results when BSE receives an invalid SOAP request:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">

 <SOAP-ENV:Body>
      <SOAP-ENV:Fault>
         <faultcode>SOAP-ENV:Client</faultcode>
         <faultstring>Parameter node is missing</faultstring>
      </SOAP-ENV:Fault>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

In this example, BSE did not receive an element in the SOAP request message that is mandatory for the WSDL for this Web service.

10.2.2 Adapter-Specific Error Handling

This section contains the following topics:

- Section 10.2.2.1, "Oracle Application Adapter for Siebel Invalid SOAP Request"
- Section 10.2.2.2, "Empty Result From Siebel Request"
- Section 10.2.2.3, "Oracle WebLogic Server Integration Adapters"
- Section 10.2.2.4, "Invalid SOAP Request"
- Section 10.2.2.5, "Empty Result From Oracle WebLogic Server Adapter Request"

When an adapter raises an exception during run-time, the SOAP agent in BSE produces a SOAP fault element in the generated SOAP response. The SOAP fault element contains fault code and fault string elements. The fault string contains the native error description from the adapter target system. Since adapters use the target system interfaces and APIs, whether an exception is raised depends on how the target systems interface or API treats the error condition. If a SOAP request message is passed to an adapter by the SOAP agent in BSE, and that request is invalid based on the WSDL for that service, then the adapter may raise an exception yielding a SOAP fault.

While it is almost impossible to anticipate every error condition that an adapter may encounter, the following is a description of how adapters handle common error conditions and how they are then exposed to the Web services consumer application.

10.2.2.1 Oracle Application Adapter for Siebel Invalid SOAP Request

If Oracle Application Adapter for Siebel receives a SOAP request message that does not conform to the WSDL for the Web services being executed, then the following SOAP response is generated

```
<?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>XD[FAIL] Parse failure (IS) 3: org.xml.sax.SAXParseException:
Premature end of file.</faultstring>
 </SOAP-ENV:Fault>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

10.2.2.2 Empty Result From Siebel Request

If Oracle Application Adapter for Siebel cannot connect to Siebel when executing a Web service, then the following SOAP response is generated.

```
<?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><Exception> - major:4096 minor: -1 message:NSReadKey request 11 was
abandoned
after 37846ms connection:12a due to Connection shutdown request
Connection reset by peer:JVM_recv in socket input stream
stream read DetailedMessage:Unknown</Exception></faultstring>
  </SOAP-ENV:Fault>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

10.2.2.3 Oracle WebLogic Server Integration Adapters

Oracle Adapters connect BSE to adapters whose engines are other Oracle servers. Therefore, since this type of adapter is used to connect BSE to many different target systems, the error handling behavior is consistent. Check the user guide for your adapter to see if you require the Oracle WebLogic Server Integration Adapter when running Web services.

10.2.2.4 Invalid SOAP Request

If Oracle Application Adapter for Siebel receives a SOAP request message that does not conform to the WSDL for the Web services being executed, then the following SOAP response is generated.

```
<?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 server connection failed: Connection refused:
connect</faultstring>
 </SOAP-ENV:Fault>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

10.2.2.5 Empty Result From Oracle WebLogic Server Adapter Request

If Oracle Application Adapter for Siebel executes a SOAP request using input parameters passed that do not match records in the target system, then the following SOAP response is generated.

Note: The condition for this adapter does not yield a SOAP fault.

```
<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>
```

A

Using Siebel Workflows

When using Siebel XML to integrate with Siebel Integration Objects, the interface uses a Siebel Workflow.

> **Note:** This appendix is intended as a supplement to the documentation designed for Oracle Application Adapter for Siebel user and is not intended as a substitute for Siebel documentation. For complete and up-to-date information on Siebel Workflow and policy topics, see the Siebel Bookshelf for your Siebel system.

This appendix contains the following sections:

- Section A.1, "Overview"
- Section A.2, "Creating a Siebel Workflow"

A.1 Overview

A Siebel Workflow is defined within Siebel to emit or to receive Siebel XML. In either case, emitting or receiving is handled by Siebel transport services for MQSeries, File, or HTTP. This section contains the following topics that describe the use and creation of workflows that employ the supported transport services:

- Section A.1.1, "Siebel Workflows"
- Section A.1.2, "Using a Policy to Invoke a Siebel EAI Workflow"
- Section A.1.3, "Siebel Workflow - Outbound"
- Section A.1.4, "Siebel Workflow - Inbound"

A.1.1 Siebel Workflows

A Siebel Workflow is a series of Siebel Business Services linked together to accomplish a business task. You create workflows using the Siebel Client Workflow Administration screens. Workflows are invoked through one of the following methods:

- Using a workflow policy
- Using a run-time event (Siebel Event)
- Using a script (eScript or Siebel VB)

The following topic briefly describes how to invoke the workflow through a policy condition.

See Also:

Siebel Bookshelf documentation for more information on policy and other methods.

A.1.2 Using a Policy to Invoke a Siebel EAI Workflow

A workflow policy is defined by a set of conditions that performs a set of defined actions. A Siebel workflow policy consists of:

- Conditions that define circumstances, based on changes in the state of a Siebel database.
- Actions that define steps taken when conditions are fulfilled.

Creating a policy to invoke a workflow as an action involves the following steps:

1. Define an action to be executed after a policy is triggered. Use the Run Integration Process program.
2. Create a policy by setting conditions and selecting appropriate policy groups and actions.
3. Activate the policy by choosing an activation date.
4. Run the Generate Triggers server task from Server Administration windows to set the conditions to be monitored.
5. Start the Workflow Monitor agent after editing with the appropriate policy group (to which your policy belongs) to evaluate whether to perform an action.
6. Start the Workflow Action Agent server task from Server Administration windows to perform the action.

A.1.3 Siebel Workflow - Outbound

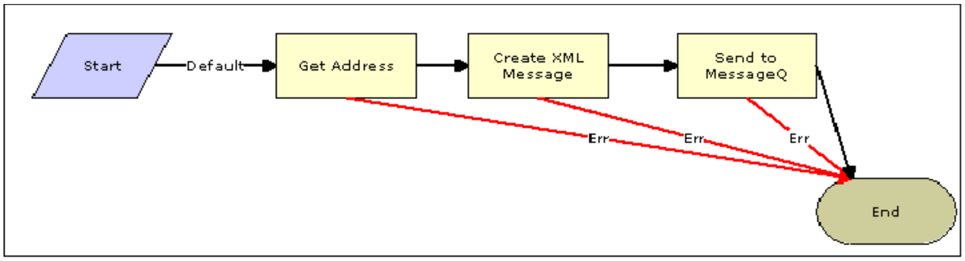
When a Siebel Workflow is triggered based on a Siebel policy, run-time, or script (eScript or Siebel VB) event, the result is the generation of a Siebel XML document that is placed on one of the Siebel transports. For example, when you add a new account in the Siebel Call Center application, you can design and configure a workflow to be triggered on the account transaction. You can design the workflow to extract the data for the new record, convert it to Siebel XML, and then, place it on an MQSeries message queue.

In this example, the Siebel Workflow process executes the following series of Siebel Business Services:

1. Calls the Siebel EAI Siebel Adapter that queries for the newly updated account record and places the data in its original internal structure into memory.
2. Calls the Siebel EAI XML Converter that converts the data into an XML message.
3. Calls the Siebel EAI MQSeries Transport that places the newly created XML message into the appropriate MQSeries message queue

   After the message is placed in the message queue, it is retrieved by Oracle Application Adapter for Siebel 6.3 and higher. The following Workflow sequence illustrates the previous steps, as shown in Figure A–1.

Figure A–1 How to Create a Siebel Workflow That Exports Siebel Update Data



A.1.4 Siebel Workflow - Inbound

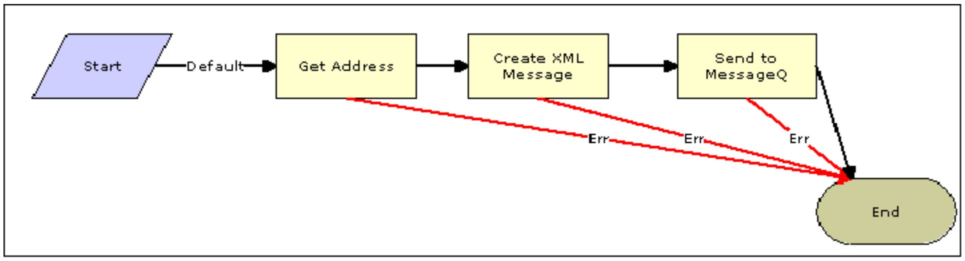
A Siebel Workflow that is triggered by an external event begins by receiving a Siebel XML document placed on one of its transports. The result might be the update of a Siebel record using the XML as input, for example, when a new account is added in another CRM system but also must be updated in the Siebel Call Center application. You can design and configure a Workflow to receive or listen on an MQSeries message queue. Upon receipt of the XML message, the Workflow processes the transaction into the Siebel system to update the record.

In this example, upon receipt of the Siebel XML message in the message queue, the Siebel MQSeries Receiver server task initiates a Siebel Workflow process, which in turn executes a series of Siebel Business Services as follows:

1. Calls the Siebel EAI XML Converter, which converts the XML message into Siebel internal format.
2. Calls the Siebel EAI Siebel Adapter, which applies the newly updated account record based on the methods defined in its service.

The following is a sample of the Workflow process, as shown in Figure A–2.

Figure A–2 Sample Workflow Process



A.2 Creating a Siebel Workflow

This section contains the following topics that include procedures for creating Siebel Workflows in the Siebel Workflow Administration window:

- Section A.2.1, "Creating a Siebel Workflow for an Event Using MQSeries Transport"
- Section A.2.2, "Creating a Siebel Workflow for an Event Using File Transport"
- Section A.2.3, "Creating a Siebel Workflow for an Event Using HTTP Transport"
- Section A.2.4, "Creating a Siebel Workflow for a Service Using MQSeries Transport"

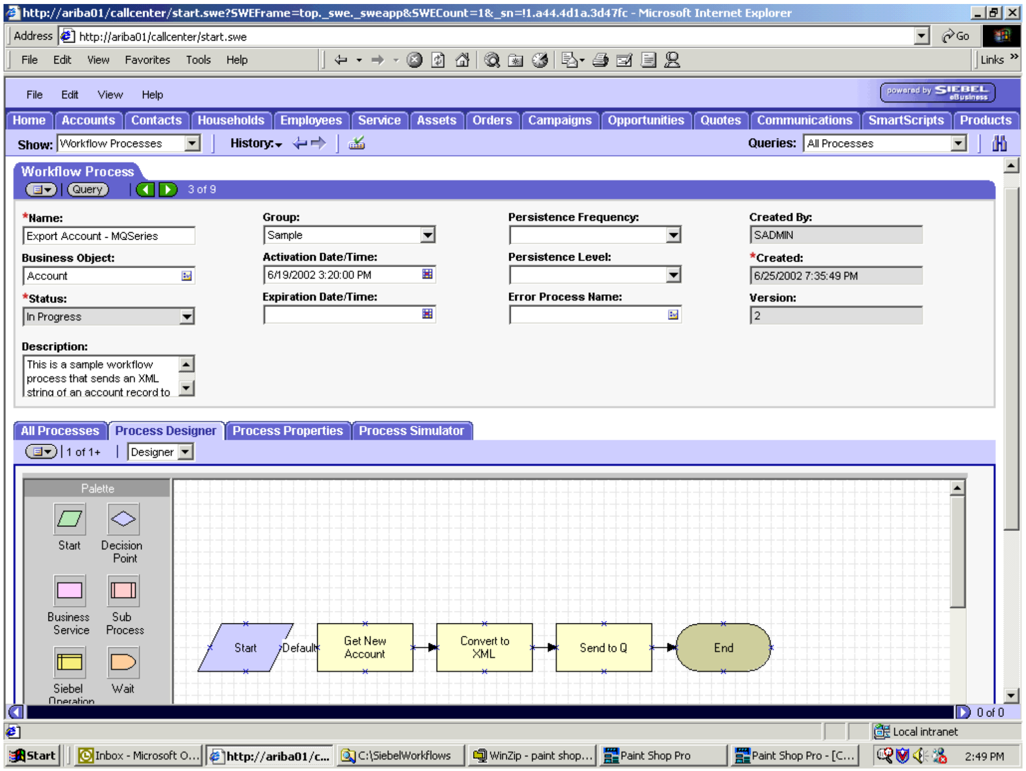
- Section A.2.5, "Creating a Siebel Workflow for a Service Using File Transport"
- Section A.2.6, "Creating a Siebel Workflow for a Service Using HTTP Transport"

A.2.1 Creating a Siebel Workflow for an Event Using MQSeries Transport

The following procedure is an example of a Siebel Workflow illustrated in the Siebel Workflow Administration window. The Workflow was designed for exporting Siebel Account record information using the MQSeries transport.

The following is a Siebel Workflow Administration window, as shown in Figure A–3.

Figure A–3 Siebel Workflow Administration Window



http://ariba01/callcenter/start.swe?SWEFrame=top._swe._sweapp&SWECount=1&_sn=!1.a44.4d1a.3d47fc - Microsoft Internet Explorer

Address http://ariba01/callcenter/start.swe

Show: Workflow Processes

Queries: All Processes

Workflow Process 3 of 9

*Name: Export Account - MQSeries

Group: Sample

Persistence Frequency:

Created By: SADMIN

Business Object: Account

Activation Date/Time: 6/19/2002 3:20:00 PM

Persistence Level:

*Created: 6/25/2002 7:35:49 PM

*Status: In Progress

Expiration Date/Time:

Error Process Name:

Version: 2

Description: This is a sample workflow process that sends an XML string of an account record to

The following procedure describes how to create a Siebel Workflow that generates Siebel XML when an Account record is updated in the Siebel Call Center application. The Workflow is then placed on an MQSeries message queue.

To create a Siebel Workflow:

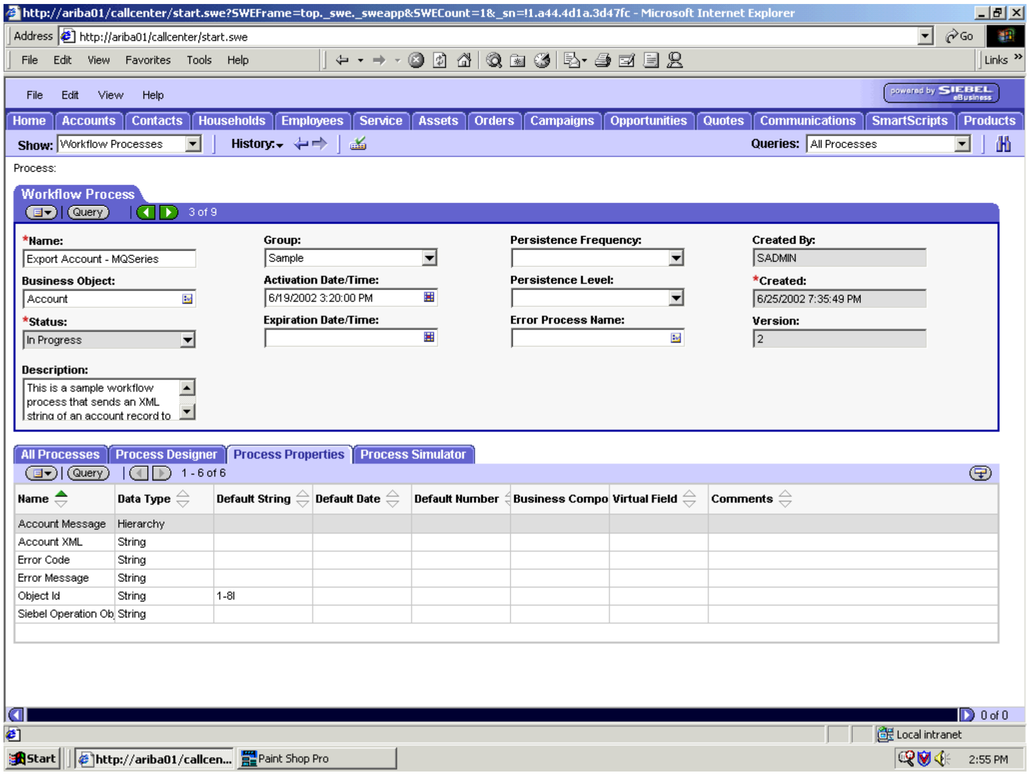
1. In the Process Properties tab of the Workflow Process window, define the Account message and Account XML process properties.

   The Account message contains Siebel Account data in hierarchical format.

   Account XML specifies the Siebel Account data that the workflow has converted to XML.

   The following window is displayed, showing the Process Properties tab active, as shown in Figure A–4.

Figure A–4 Process Properties Tab of the Workflow Process Window


http://ariba01/callcenter/start.swe?SWEFrame=top._swe._sweapp&SWECount=1&_sn=!1.a44.4d1a.3d47fc - Microsoft Internet Explorer
Address http://ariba01/callcenter/start.swe
Go
File Edit View Favorites Tools Help
Links
File Edit View Help
powered by SIEBEL eBusiness
Home Accounts Contacts Households Employees Service Assets Orders Campaigns Opportunities Quotes Communications SmartScripts Products
History:
Query
Process Properties
All Processes Process Designer Process Simulator
0 of 0
Local intranet
Start
http://ariba01/callcen...
Paint Shop Pro
2:55 PM

Show: Workflow Processes

Queries: All Processes

Process:

Workflow Process

3 of 9

*Name: Export Account - MQSeries	Group: Sample	Persistence Frequency:	Created By: SADMIN
Business Object: Account	Activation Date/Time: 6/19/2002 3:20:00 PM	Persistence Level:	*Created: 6/25/2002 7:35:49 PM
*Status: In Progress	Expiration Date/Time:	Error Process Name:	Version: 2

Description: This is a sample workflow process that sends an XML string of an account record to

1 - 6 of 6

Name	Data Type	Default String	Default Date	Default Number	Business Compo	Virtual Field	Comments
Account Message	Hierarchy						
Account XML	String						
Error Code	String						
Error Message	String						
Object Id	String	1-8I					
Siebel Operation Ob	String						

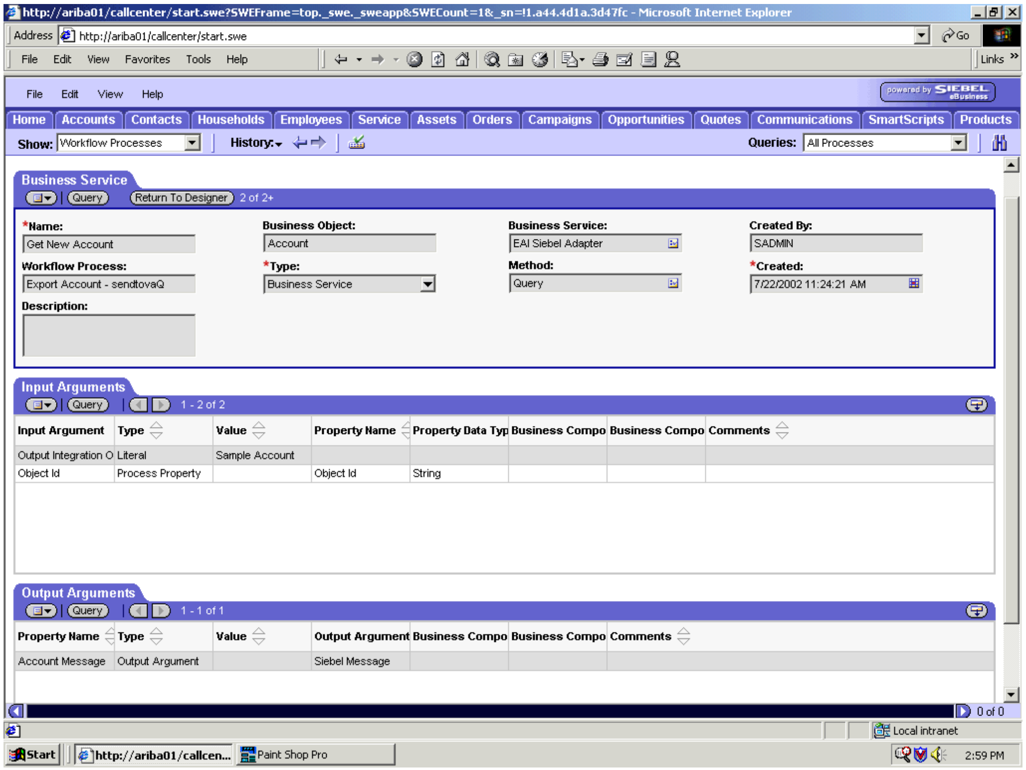
2. Use the Siebel Workflow Administration windows to create a Workflow.

3. Define an EAI Siebel Adapter Business Service step to receive an instance of Account data and call it `Get New Account`.

   The Business Service obtains the Account information from Siebel using the Query method.

   Output from this Business Service is generated in hierarchical format, as shown in Figure A–5.

Figure A–5 Output From Business Service Generated From a Hierachical Format



http://ariba01/callcenter/start.swe?SWEFrame=top._swe._sweapp&SWECount=1&_sn=!1.a44.4d1a.3d47fc - Microsoft Internet Explorer
Address http://ariba01/callcenter/start.swe
File Edit View Favorites Tools Help
Links
File Edit View Help
powered by SIEBEL eBusiness
Home Accounts Contacts Households Employees Service Assets Orders Campaigns Opportunities Quotes Communications SmartScripts Products
Show: Workflow Processes
History:
Queries: All Processes
Business Service
Query
Return To Designer
2 of 2+

*Name: Get New Account	Business Object: Account	Business Service: EAI Siebel Adapter	Created By: SADMIN
Workflow Process: Export Account - sendtovaQ	*Type: Business Service	Method: Query	*Created: 7/22/2002 11:24:21 AM
Description:			

Input Arguments
Query
1 - 2 of 2

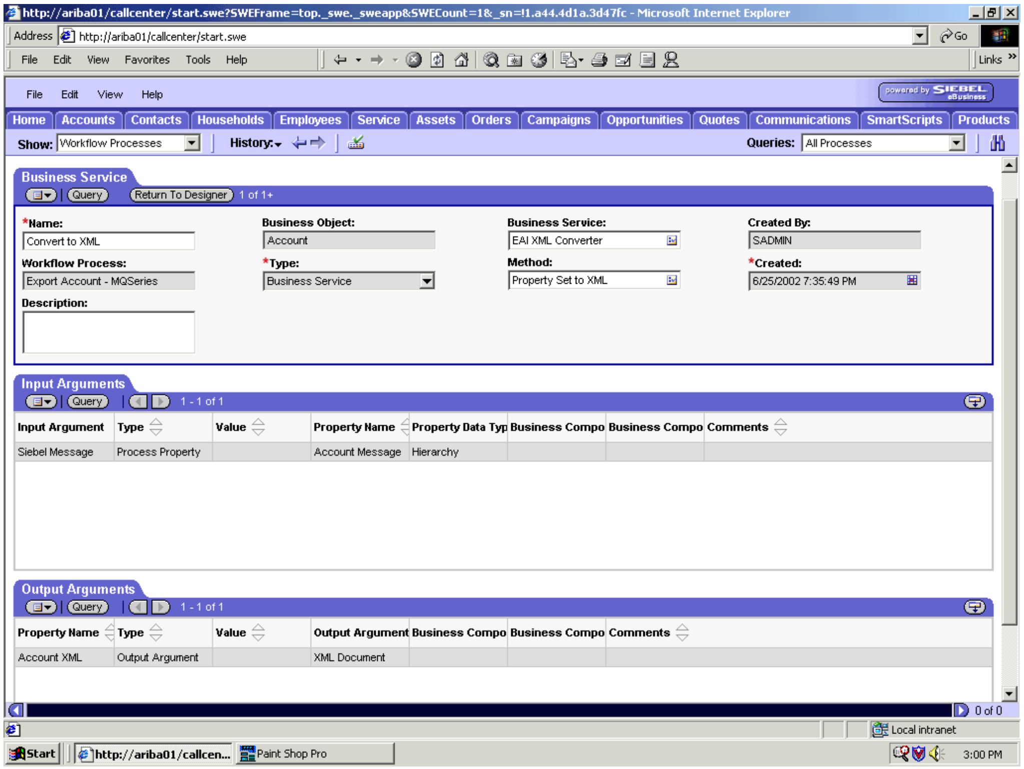
Input Argument	Type	Value	Property Name	Property Data Typ	Business Compo	Business Compo	Comments
Output Integration O	Literal	Sample Account					
Object Id	Process Property		Object Id	String			

Property Name	Type	Value	Output Argument	Business Compo	Business Compo	Comments
Account Message	Output Argument		Siebel Message			

4. Define an EAI XML Converter Business Service step and call it `Convert to XML`.

   It is defined to receive the Account data from the EAI Siebel Adapter Business Service in hierarchical format and convert it to XML format, as shown in Figure A–6.

Figure A–6 EAI XML Converter Business Service


http://ariba01/callcenter/start.swe?SWEFrame=top._swe._sweapp&SWECount=1&_sn=!1.a44.4d1a.3d47fc - Microsoft Internet Explorer
Address http://ariba01/callcenter/start.swe
Go
Links
File Edit View Favorites Tools Help
File Edit View Help
powered by SIEBEL eBusiness
Home Accounts Contacts Households Employees Service Assets Orders Campaigns Opportunities Quotes Communications SmartScripts Products
Show: Workflow Processes
History:
Queries: All Processes
Query
Return To Designer
1 of 1+
1 - 1 of 1
0 of 0
Local intranet
Start
http://ariba01/callcen...
Paint Shop Pro
3:00 PM

Business Service

*Name: Convert to XML	Business Object: Account	Business Service: EAI XML Converter	Created By: SADMIN
Workflow Process: Export Account - MQSeries	*Type: Business Service	Method: Property Set to XML	*Created: 6/25/2002 7:35:49 PM
Description:			

Input Arguments

Input Argument	Type	Value	Property Name	Property Data Typ	Business Compo	Business Compo	Comments
Siebel Message	Process Property		Account Message	Hierarchy			

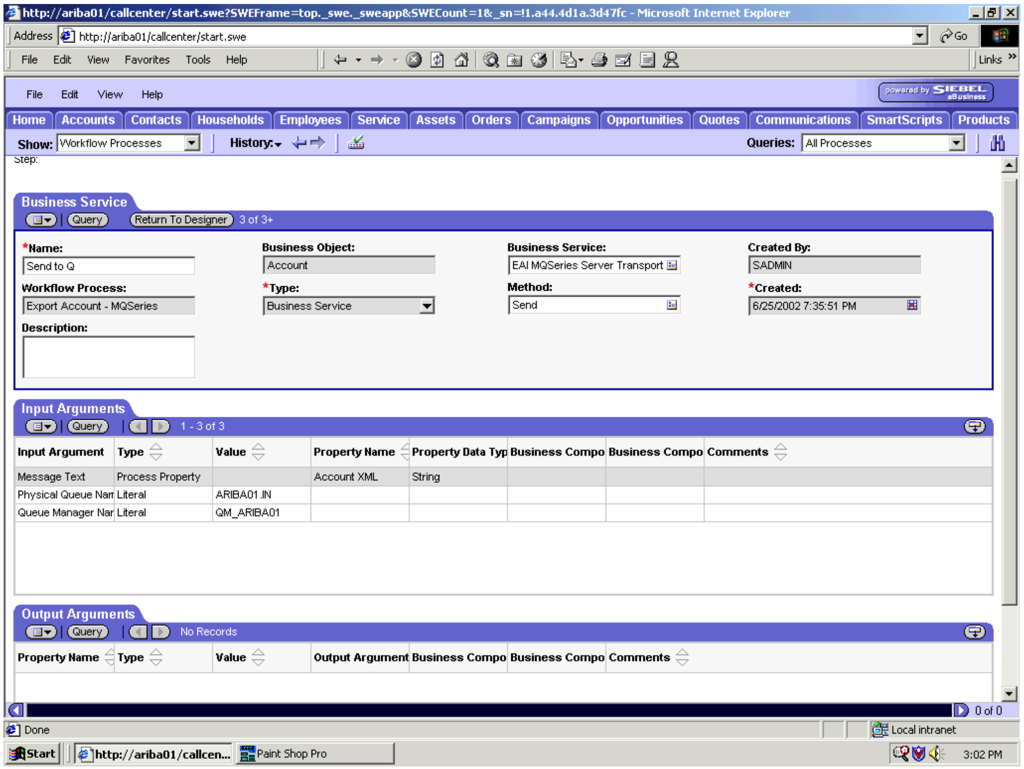
Output Arguments

Property Name	Type	Value	Output Argument	Business Compo	Business Compo	Comments
Account XML	Output Argument		XML Document			

5. Define an EAI MQSeries server transport Business Service step and call it `Send to Q`, as shown in Figure A–7.

   It is defined to receive the Account data from the EAI XML Converter Business Service in Siebel XML format and send the Account XML to MQSeries using the Send method.

Figure A–7 EAI MQ Series Server Transport Business Server Send to Q


http://ariba01/callcenter/start.swe?SWEFrame=top._swe._sweapp&SWECount=1&_sn=!1.a44.4d1a.3d47fc - Microsoft Internet Explorer
Address http://ariba01/callcenter/start.swe
File Edit View Favorites Tools Help
Links
File Edit View Help
Home Accounts Contacts Households Employees Service Assets Orders Campaigns Opportunities Quotes Communications SmartScripts Products
Show: Workflow Processes
History:
Queries: All Processes
Step:
Business Service
Query
Return To Designer
3 of 3+
Input Arguments
Query
1 - 3 of 3
Output Arguments
Query
No Records
0 of 0
Done
Local intranet
Start
http://ariba01/callcen...
Paint Shop Pro
3:02 PM

Business Service

*Name: Send to Q	Business Object: Account	Business Service: EAI MQSeries Server Transport	Created By: SADMIN
Workflow Process: Export Account - MQSeries	*Type: Business Service	Method: Send	*Created: 6/25/2002 7:35:51 PM
Description:			

Input Arguments

Input Argument	Type	Value	Property Name	Property Data Typ	Business Compo	Business Compo	Comments
Message Text	Process Property		Account XML	String			
Physical Queue Nam	Literal	ARIBA01.IN					
Queue Manager Nar	Literal	QM_ARIBA01					

Output Arguments

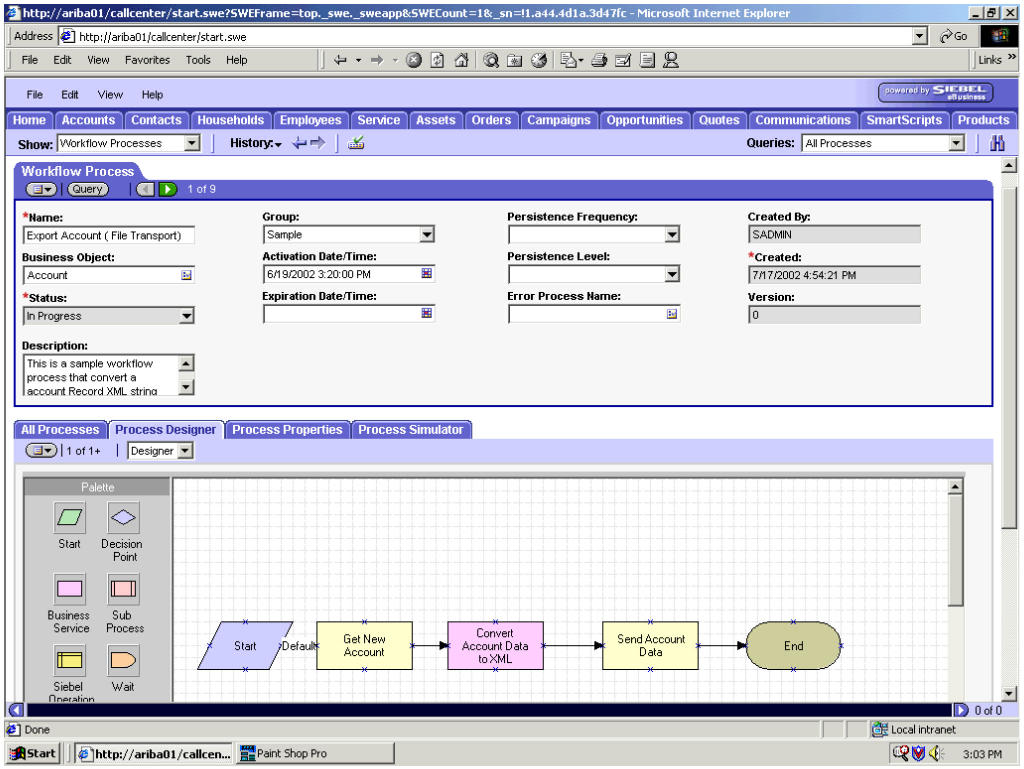
Property Name	Type	Value	Output Argument	Business Compo	Business Compo	Comments

A.2.2 Creating a Siebel Workflow for an Event Using File Transport

The following procedure is an example of a Siebel Workflow illustrated in the Siebel Workflow Administration window. The Workflow was designed for exporting Siebel Account record information using the File transport.

The following window is displayed with the Process Designer tab active, as shown in Figure A–8.

Figure A–8 Siebel Workflow Administration Window


http://ariba01/callcenter/start.swe?SWEFrame=top._swe._sweapp&SWECount=1&_sn=!1.a44.4d1a.3d47fc - Microsoft Internet Explorer
Address http://ariba01/callcenter/start.swe
File Edit View Favorites Tools Help
Links
File Edit View Help
Home Accounts Contacts Households Employees Service Assets Orders Campaigns Opportunities Quotes Communications SmartScripts Products
History:
Workflow Process
Query
1 of 9
All Processes Process Designer Process Properties Process Simulator
1 of 1+
Designer
Palette
Start
Decision Point
Business Service
Sub Process
Siebel Operation
Wait
Start
Default
Get New Account
Convert Account Data to XML
Send Account Data
End
0 of 0
Done
Local intranet
Start
http://ariba01/callcen...
Paint Shop Pro
3:03 PM

Show: Workflow Processes

Queries: All Processes

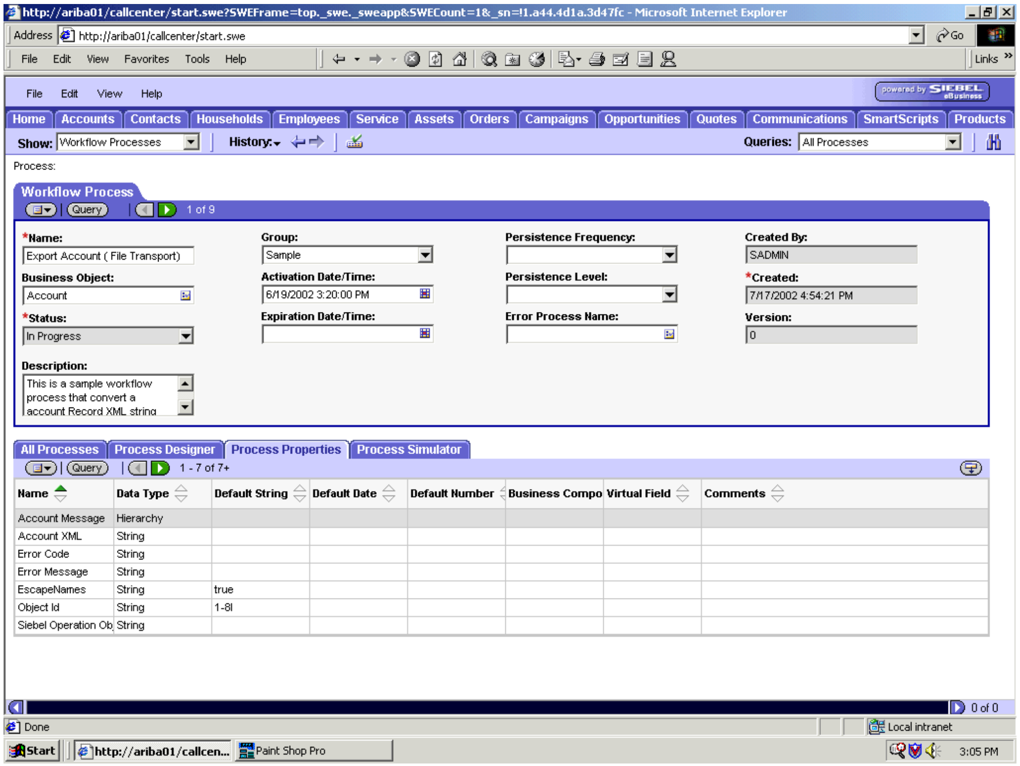
*Name: Export Account (File Transport)	Group: Sample	Persistence Frequency:	Created By: SADMIN
Business Object: Account	Activation Date/Time: 6/19/2002 3:20:00 PM	Persistence Level:	*Created: 7/17/2002 4:54:21 PM
*Status: In Progress	Expiration Date/Time:	Error Process Name:	Version: 0

Description: This is a sample workflow process that convert a account Record XML string

This procedure describes how to create a Siebel Workflow that generates Siebel XML when an Account record is updated in the Siebel Call Center application and then places Siebel XML on the file system.

To create a Siebel Workflow:

Figure A–9 Process Properties Tab of the Workflow Process Window



http://ariba01/callcenter/start.swe?SWEFrame=top._swe._sweapp&SWECount=1&_sn=!1.a44.4d1a.3d47fc - Microsoft Internet Explorer
Address http://ariba01/callcenter/start.swe
File Edit View Favorites Tools Help
Links
File Edit View Help
Home Accounts Contacts Households Employees Service Assets Orders Campaigns Opportunities Quotes Communications SmartScripts Products
History:
Start
http://ariba01/callcen...
Paint Shop Pro
3:05 PM

Show: Workflow Processes

Queries: All Processes

Process:

Workflow Process

Query 1 of 9

*Name:	Group:	Persistence Frequency:	Created By:
Export Account (File Transport)	Sample		SADMIN
Business Object:	Activation Date/Time:	Persistence Level:	*Created:
Account	6/19/2002 3:20:00 PM		7/17/2002 4:54:21 PM
*Status:	Expiration Date/Time:	Error Process Name:	Version:
In Progress			0

Description:
This is a sample workflow process that convert a account Record XML string

All Processes | Process Designer | Process Properties | Process Simulator

Query 1 - 7 of 7+

Name	Data Type	Default String	Default Date	Default Number	Business Compo	Virtual Field	Comments
Account Message	Hierarchy						
Account XML	String						
Error Code	String						
Error Message	String						
EscapeNames	String	true					
Object Id	String	1-8I					
Siebel Operation Ob	String						

0 of 0

Done

Local intranet

1. On the Process Properties tab of the Workflow Process window, define the Account message and Account XML process properties, as shown in Figure A–9.

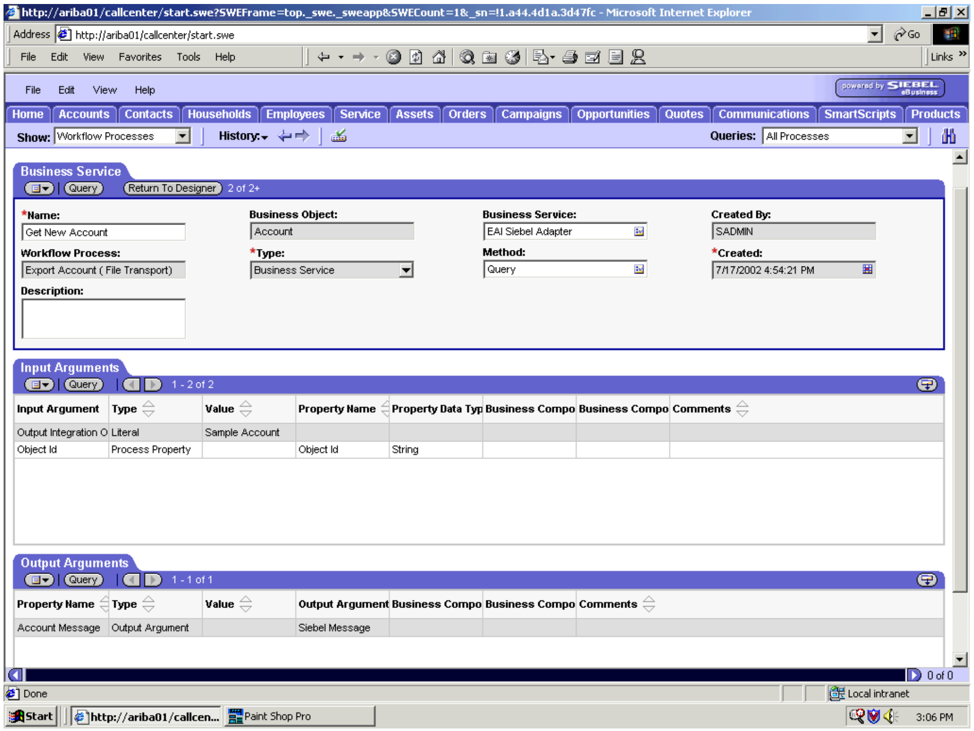
   Account message contains the Siebel Account data in hierarchical format.

   Account XML specifies which Siebel Account data the Workflow converted to XML.

2. Use the Siebel Workflow Administration windows to create a Workflow.

   As shown in Figure A–10, the following is an example of a Siebel Workflow Administration window.

Figure A–10 Siebel Workflow Administration Window


http://ariba01/callcenter/start.swe?SWEFrame=top._swe._sweapp&SWECount=1&_sn=!1.a44.4d1a.3d47fc - Microsoft Internet Explorer
Address http://ariba01/callcenter/start.swe
File Edit View Favorites Tools Help
Links
File Edit View Help
powered by SIEBEL eBusiness
Home Accounts Contacts Households Employees Service Assets Orders Campaigns Opportunities Quotes Communications SmartScripts Products
History:
Done
Local intranet
Start
http://ariba01/callcen...
Paint Shop Pro
3:06 PM

Show: Workflow Processes

Queries: All Processes

Business Service

Query | Return To Designer | 2 of 2+

*Name: Get New Account	Business Object: Account	Business Service: EAI Siebel Adapter	Created By: SADMIN
Workflow Process: Export Account (File Transport)	*Type: Business Service	Method: Query	*Created: 7/17/2002 4:54:21 PM
Description:			

Input Arguments

Query | 1 - 2 of 2

Input Argument	Type	Value	Property Name	Property Data Typ	Business Compo	Business Compo	Comments
Output Integration O	Literal	Sample Account					
Object Id	Process Property		Object Id	String			

Output Arguments

Query | 1 - 1 of 1

Property Name	Type	Value	Output Argument	Business Compo	Business Compo	Comments
Account Message	Output Argument		Siebel Message			

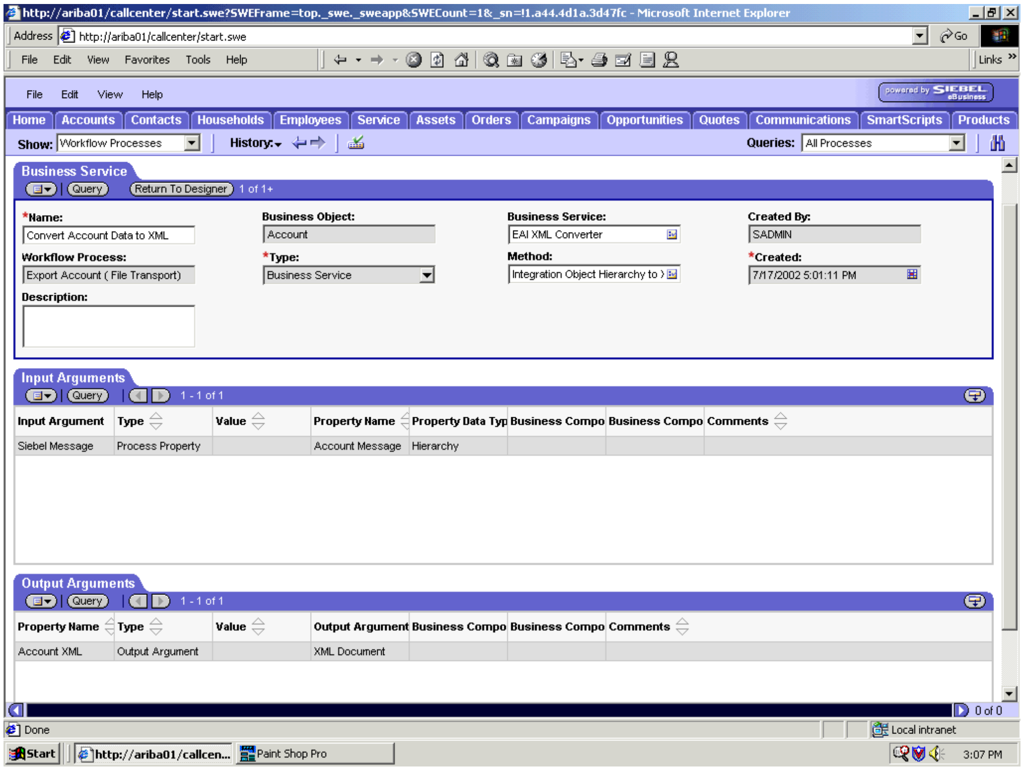
0 of 0

3. Define an EAI Siebel Adapter Business Service step to receive an instance of Account data and call it `Get New Account`.

   The Business Service obtains the Account information from Siebel using the Query method.

   Output from this Business Service is generated in hierarchical format.

Figure A–11 Creation of an EAI XML Converter


http://ariba01/callcenter/start.swe?SWEFrame=top._swe._sweapp&SWECount=1&_sn=!1.a44.4d1a.3d47fc - Microsoft Internet Explorer
Address http://ariba01/callcenter/start.swe
File Edit View Favorites Tools Help
File Edit View Help
Home Accounts Contacts Households Employees Service Assets Orders Campaigns Opportunities Quotes Communications SmartScripts Products
Show: Workflow Processes
History:
Queries: All Processes
Query
Return To Designer
1 of 1+
Done
Local intranet
Start
Paint Shop Pro
3:07 PM

Business Service

*Name: Convert Account Data to XML	Business Object: Account	Business Service: EAI XML Converter	Created By: SADMIN
Workflow Process: Export Account (File Transport)	*Type: Business Service	Method: Integration Object Hierarchy to X	*Created: 7/17/2002 5:01:11 PM
Description:			

Input Arguments 1 - 1 of 1

Input Argument	Type	Value	Property Name	Property Data Typ	Business Compo	Business Compo	Comments
Siebel Message	Process Property		Account Message	Hierarchy			

Output Arguments 1 - 1 of 1

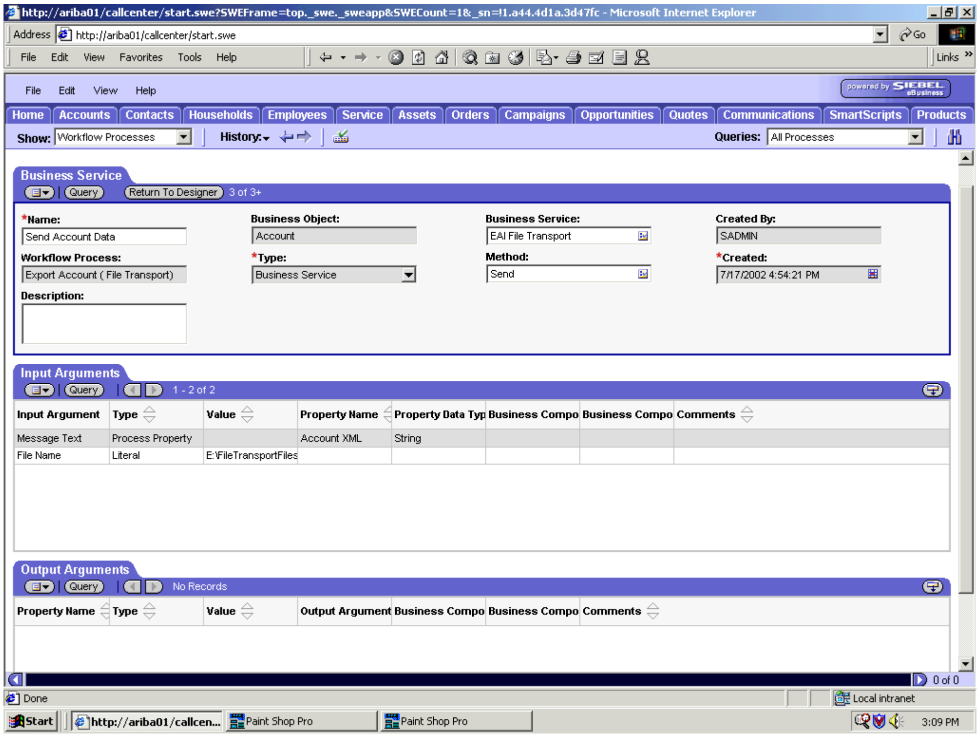
Property Name	Type	Value	Output Argument	Business Compo	Business Compo	Comments
Account XML	Output Argument		XML Document			

0 of 0

4. As shown in Figure A–11, define an EAI XML Converter Business Service step and call it `Convert Account Data to XML`.

   This Business Service is defined to receive the Account data from the EAI Siebel Adapter Business Service in hierarchical format and convert it to XML format.

Figure A–12 File Transport for the EAI XML Converter Business Service



http://ariba01/callcenter/start.swe?SWEFrame=top._swe._sweapp&SWECount=1&_sn=!1.a44.4d1a.3d47fc - Microsoft Internet Explorer
Address http://ariba01/callcenter/start.swe
Go
File Edit View Favorites Tools Help
Links
File Edit View Help
powered by SIEBEL eBusiness
Home Accounts Contacts Households Employees Service Assets Orders Campaigns Opportunities Quotes Communications SmartScripts Products
Show: Workflow Processes
History:
Queries: All Processes
Business Service
Query
Return To Designer
3 of 3+

*Name: Send Account Data

Workflow Process: Export Account (File Transport)

Description:

Business Object: Account

*Type: Business Service

Business Service: EAI File Transport

Method: Send

Created By: SADMIN

*Created: 7/17/2002 4:54:21 PM

Input Arguments
Query
1 - 2 of 2

Input Argument	Type	Value	Property Name	Property Data Typ	Business Compo	Business Compo	Comments
Message Text	Process Property		Account XML	String			
File Name	Literal	E:\FileTransportFiles					

Property Name	Type	Value	Output Argument	Business Compo	Business Compo	Comments

5. As shown in Figure A–12, define an EAI File Transport Business Service step and call it `Send Account Data`.

   This Business Service is defined to receive the Account data from the EAI XML Converter Business Service in Siebel XML format and send the Account XML to the file system in a specified directory using the Send method.

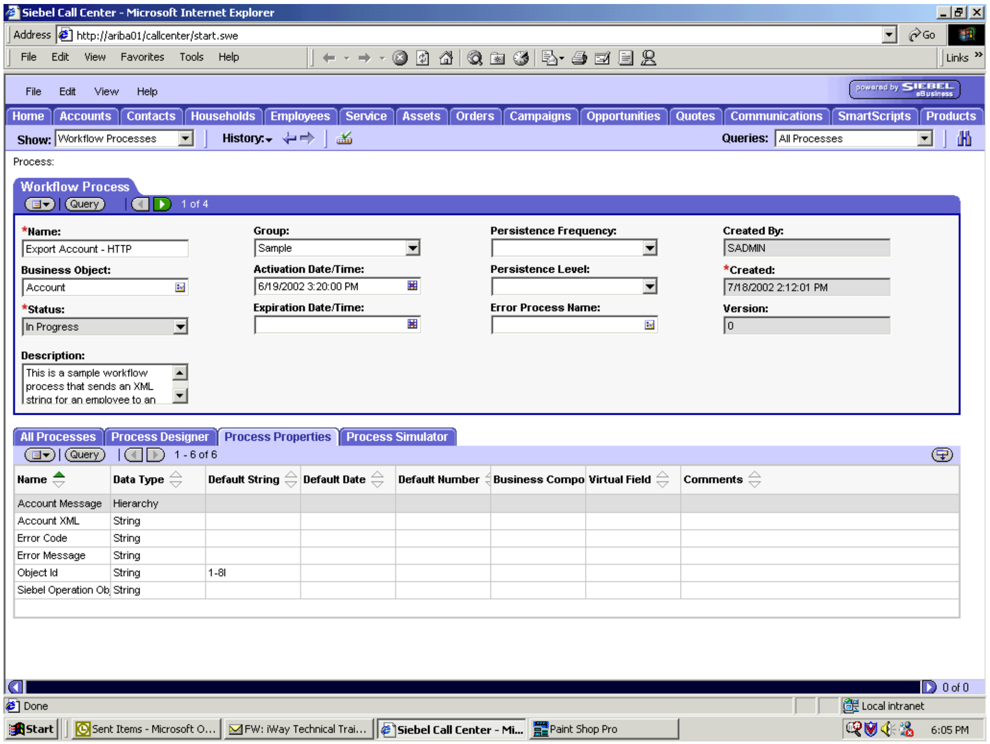
A.2.3 Creating a Siebel Workflow for an Event Using HTTP Transport

The following procedure is an example of a Siebel Workflow illustrated in the Siebel Workflow Administration window. The Workflow was designed for exporting Siebel Account record information using the HTTP transport.

This procedure describes how to create a Siebel Workflow that generates Siebel XML when an Account record is updated in the Siebel Call Center application.

To create a Siebel Workflow:

Figure A–13 Process Properties Tab of the Siebel Workflow Process Window


Siebel Call Center - Microsoft Internet Explorer
Address http://ariba01/callcenter/start.swe
Go
Links
File Edit View Favorites Tools Help
File Edit View Help
powered by SIEBEL eBusiness
Home Accounts Contacts Households Employees Service Assets Orders Campaigns Opportunities Quotes Communications SmartScripts Products
History:
Workflow Process
Query
1 of 4
All Processes
Process Designer
Process Properties
Process Simulator
Query
1 - 6 of 6
0 of 0
Done
Local intranet
Start
Sent Items - Microsoft O...
FW: iWay Technical Trai...
Siebel Call Center - Mi...
Paint Shop Pro
6:05 PM

Show: Workflow Processes

Queries: All Processes

Process:

*Name: Export Account - HTTP

Business Object: Account

*Status: In Progress

Description: This is a sample workflow process that sends an XML string for an employee to an

Group: Sample

Activation Date/Time: 6/19/2002 3:20:00 PM

Expiration Date/Time:

Persistence Frequency:

Persistence Level:

Error Process Name:

Created By: SADMIN

*Created: 7/18/2002 2:12:01 PM

Version: 0

Name	Data Type	Default String	Default Date	Default Number	Business Compo	Virtual Field	Comments
Account Message	Hierarchy						
Account XML	String						
Error Code	String						
Error Message	String						
Object Id	String	1-8I					
Siebel Operation Ob	String						

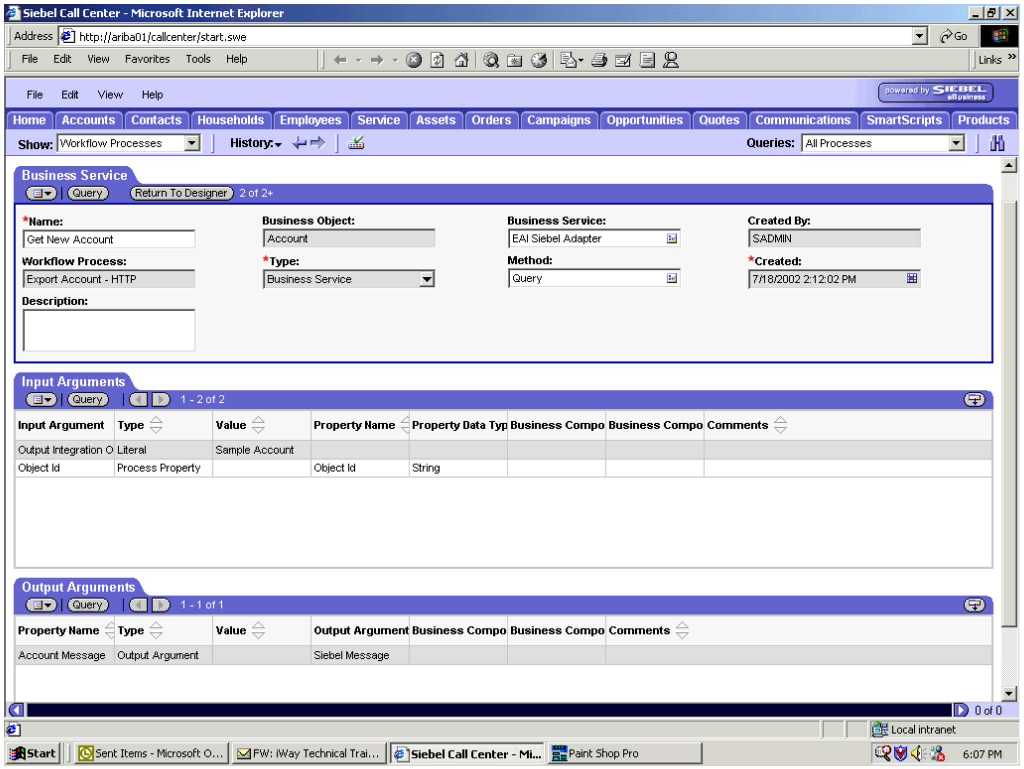
1. As shown in Figure A–13, in the Process Properties tab of the Workflow Process window, define the Account message and Account XML process properties.

   Account message contains the Siebel Account data in hierarchical format.

   Account XML specifies the Siebel Account data that the Workflow has converted to XML.

2. Use the Siebel Workflow Administration windows to create a Workflow.

Figure A–14 Siebel Workflow Administration Window


Siebel Call Center - Microsoft Internet Explorer
Address http://ariba01/callcenter/start.swe
Go
File Edit View Favorites Tools Help
Links
File Edit View Help
powered by SIEBEL eBusiness
Home Accounts Contacts Households Employees Service Assets Orders Campaigns Opportunities Quotes Communications SmartScripts Products
Show: Workflow Processes
History:
Queries: All Processes
Query
Return To Designer
Local intranet
Start
Sent Items - Microsoft O...
FW: iWay Technical Trai...
Siebel Call Center - Mi...
Paint Shop Pro
6:07 PM

Business Service 2 of 2+

*Name: Get New Account	Business Object: Account	Business Service: EAI Siebel Adapter	Created By: SADMIN
Workflow Process: Export Account - HTTP	*Type: Business Service	Method: Query	*Created: 7/18/2002 2:12:02 PM
Description:			

Input Arguments 1 - 2 of 2

Input Argument	Type	Value	Property Name	Property Data Typ	Business Compo	Business Compo	Comments
Output Integration O	Literal	Sample Account					
Object Id	Process Property		Object Id	String			

Output Arguments 1 - 1 of 1

Property Name	Type	Value	Output Argument	Business Compo	Business Compo	Comments
Account Message	Output Argument		Siebel Message			

0 of 0

3. As shown in Figure A–14, define an EAI Siebel Adapter Business Service step to receive an instance of Account data and call it `Get New Account`.

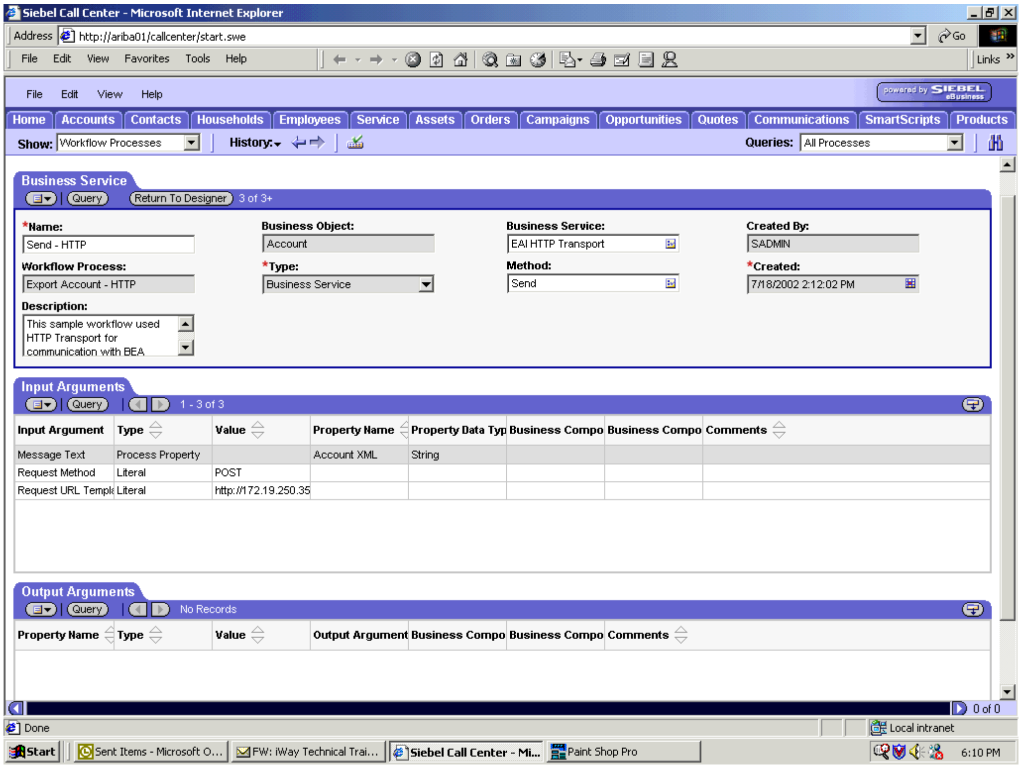
   The Business Service obtains the Account information from Siebel using the Query method.

   Output from this Business Service is generated in hierarchical format.

4. Define an EAI XML Converter Business Service step and call it `Convert to XML`.

   This Business Service is defined to receive the Account data from the EAI Siebel Adapter Business Service in hierarchical format and convert it to XML format.

Figure A–15 EAI XML Converter Business Converter Business Service


Siebel Call Center - Microsoft Internet Explorer
Address http://ariba01/callcenter/start.swe
File Edit View Favorites Tools Help
Links
File Edit View Help
powered by SIEBEL eBusiness
Home Accounts Contacts Households Employees Service Assets Orders Campaigns Opportunities Quotes Communications SmartScripts Products
Show: Workflow Processes
History:
Queries: All Processes
Business Service
Query
Return To Designer
3 of 3+
Input Arguments
Query
1 - 3 of 3
Output Arguments
Query
No Records
0 of 0
Done
Local intranet
Start
Sent Items - Microsoft O...
FW: iWay Technical Trai...
Siebel Call Center - Mi...
Paint Shop Pro
6:10 PM

Business Service

*Name: Send - HTTP	Business Object: Account	Business Service: EAI HTTP Transport	Created By: SADMIN
Workflow Process: Export Account - HTTP	*Type: Business Service	Method: Send	*Created: 7/18/2002 2:12:02 PM
Description: This sample workflow used HTTP Transport for communication with BEA			

Input Arguments

Input Argument	Type	Value	Property Name	Property Data Typ	Business Compo	Business Compo	Comments
Message Text	Process Property		Account XML	String			
Request Method	Literal	POST					
Request URL Templ	Literal	http://172.19.250.35					

Output Arguments

Property Name	Type	Value	Output Argument	Business Compo	Business Compo	Comments

5. Define an EAI HTTP Transport Business Service step and call it `Send - HTTP`, as shown in Figure A–15.

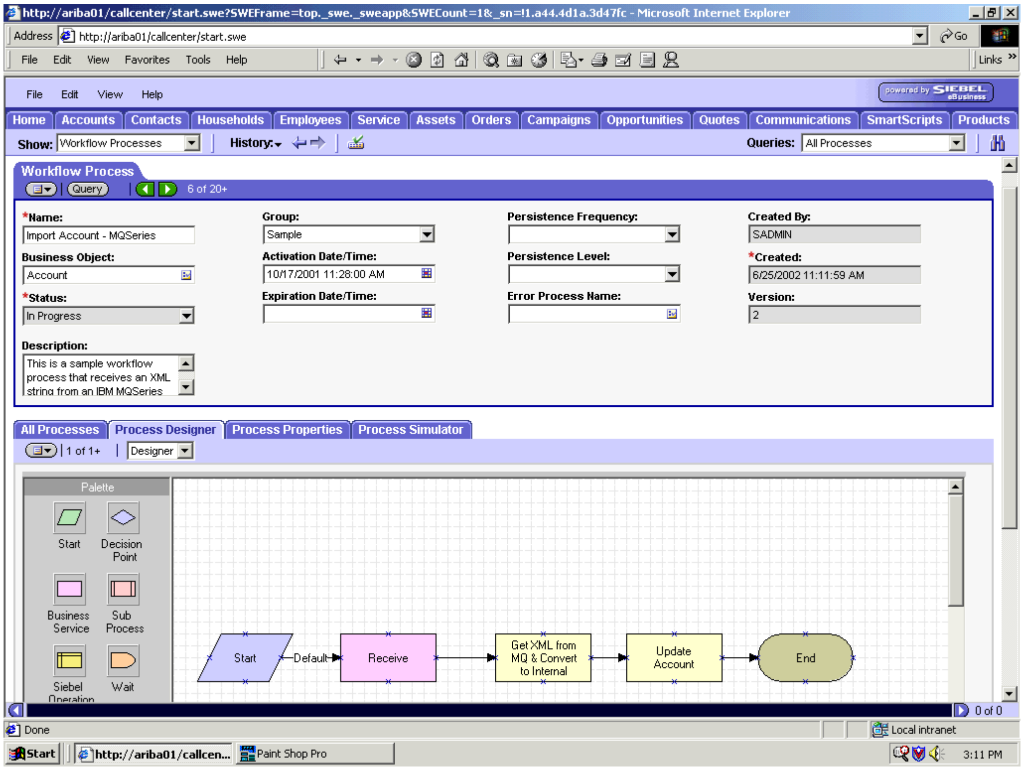
   This Business Service is defined to receive the Account data from the EAI XML Converter Business Service in Siebel XML format and send the Account XML to HTTP using the Send method.

A.2.4 Creating a Siebel Workflow for a Service Using MQSeries Transport

The following procedure is an example of a Siebel Workflow illustrated in the Siebel Workflow Administration window. The Workflow was designed for importing Siebel Account record information through the MQSeries Transport.

Figure A–16 shows a sample Siebel Workflow Administration window.

Figure A–16 Siebel Workflow Administration Window


http://ariba01/callcenter/start.swe?SWEFrame=top._swe._sweapp&SWECount=1&_sn=!1.a44.4d1a.3d47fc - Microsoft Internet Explorer
Address http://ariba01/callcenter/start.swe
Go
File Edit View Favorites Tools Help
Links
File Edit View Help
powered by SIEBEL eBusiness
Home Accounts Contacts Households Employees Service Assets Orders Campaigns Opportunities Quotes Communications SmartScripts Products
Show: Workflow Processes
History:
Queries: All Processes
Workflow Process
Query
6 of 20+
All Processes Process Designer Process Properties Process Simulator
1 of 1+
Designer
Palette
Start
Decision Point
Business Service
Sub Process
Siebel Operation
Wait
Start
Default
Receive
Get XML from MQ & Convert to Internal
Update Account
End
0 of 0
Done
Local intranet
Start
http://ariba01/callcen...
Paint Shop Pro
3:11 PM

*Name: Import Account - MQSeries

Business Object: Account

*Status: In Progress

Description: This is a sample workflow process that receives an XML string from an IBM MQSeries

Group: Sample

Activation Date/Time: 10/17/2001 11:28:00 AM

Expiration Date/Time:

Persistence Frequency:

Persistence Level:

Error Process Name:

Created By: SADMIN

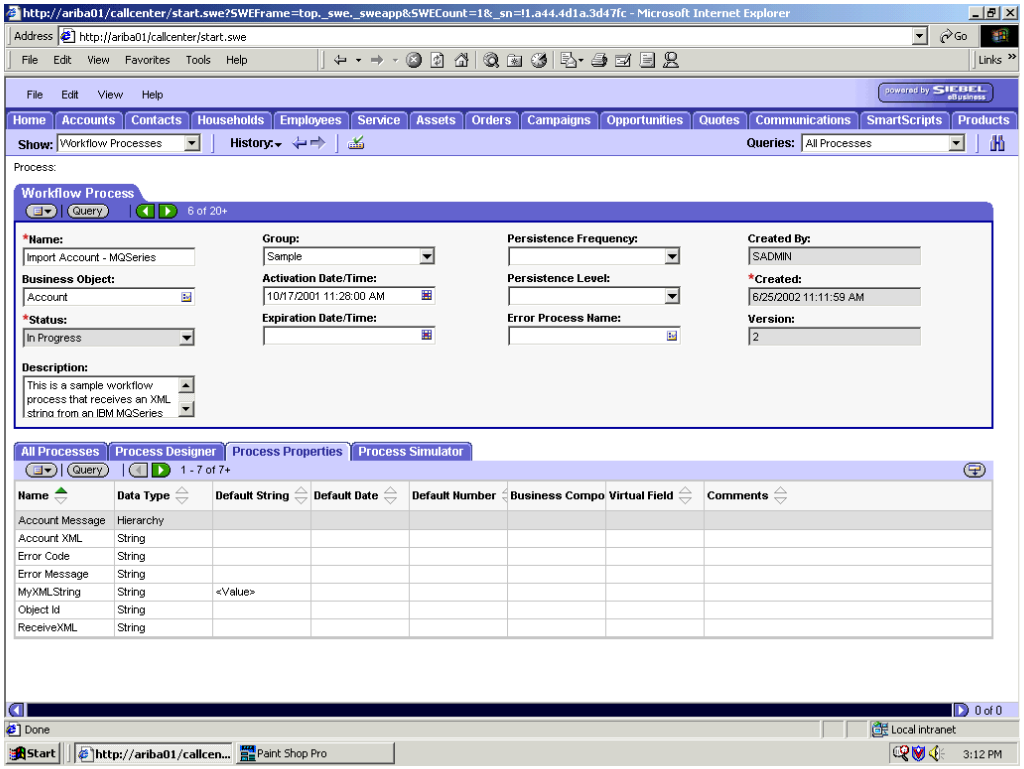
*Created: 6/25/2002 11:11:59 AM

Version: 2

This procedure describes how to create a Siebel Workflow that generates Siebel XML when an Account record is updated in the Siebel Call Center application.

To create a Siebel Workflow:

Figure A–17 Process Properties Tab of the Siebel Workflow Process Window


http://ariba01/callcenter/start.swe?SWEFrame=top._swe._sweapp&SWECount=1&_sn=!1.a44.4d1a.3d47fc - Microsoft Internet Explorer
Address http://ariba01/callcenter/start.swe
Go
File Edit View Favorites Tools Help
Links
File Edit View Help
powered by SIEBEL eBusiness
Home Accounts Contacts Households Employees Service Assets Orders Campaigns Opportunities Quotes Communications SmartScripts Products
History:
Workflow Process
Query
6 of 20+
All Processes Process Designer Process Properties Process Simulator
Query
1 - 7 of 7+
0 of 0
Done
Local intranet
Start
http://ariba01/callcen...
Paint Shop Pro
3:12 PM

Show: Workflow Processes

Queries: All Processes

Process:

*Name: Import Account - MQSeries	Group: Sample	Persistence Frequency:	Created By: SADMIN
Business Object: Account	Activation Date/Time: 10/17/2001 11:28:00 AM	Persistence Level:	*Created: 6/25/2002 11:11:59 AM
*Status: In Progress	Expiration Date/Time:	Error Process Name:	Version: 2
Description: This is a sample workflow process that receives an XML string from an IBM MQSeries			

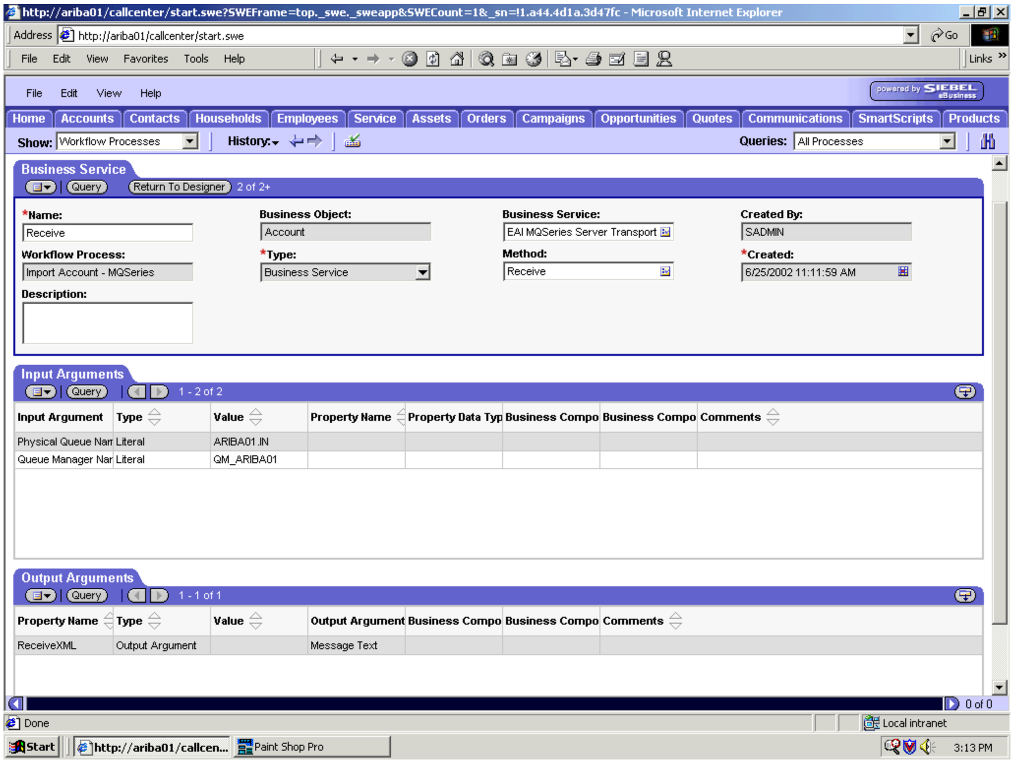
Name	Data Type	Default String	Default Date	Default Number	Business Compo	Virtual Field	Comments
Account Message	Hierarchy						
Account XML	String						
Error Code	String						
Error Message	String						
MyXMLString	String	<Value>					
Object Id	String						
ReceiveXML	String						

1. In the Process Properties tab of the Workflow Process window, define the Account message and Account XML process properties, as shown in Figure A–17.

   Account message contains the Siebel Account data in hierarchical format.

   Account XML specifies the Siebel Account data that the Workflow converted to XML.

Figure A–18 Creation of an EAI MQ Series Server Transport Business Service



http://ariba01/callcenter/start.swe?SWEFrame=top._swe._sweapp&SWECount=1&_sn=!1.a44.4d1a.3d47fc - Microsoft Internet Explorer
Address http://ariba01/callcenter/start.swe
Go
File Edit View Favorites Tools Help
Links
File Edit View Help
powered by SIEBEL eBusiness
Home Accounts Contacts Households Employees Service Assets Orders Campaigns Opportunities Quotes Communications SmartScripts Products
Show: Workflow Processes
History:
Queries: All Processes
Business Service
Query
Return To Designer
2 of 2+
Input Arguments
1 - 2 of 2
Output Arguments
1 - 1 of 1
0 of 0
Done
Local intranet
Start
http://ariba01/callcen...
Paint Shop Pro
3:13 PM

*Name: Receive	Business Object: Account	Business Service: EAI MQSeries Server Transport	Created By: SADMIN
Workflow Process: Import Account - MQSeries	*Type: Business Service	Method: Receive	*Created: 6/25/2002 11:11:59 AM
Description:			

Input Argument	Type	Value	Property Name	Property Data Typ	Business Compo	Business Compo	Comments
Physical Queue Nar	Literal	ARIBA01.IN					
Queue Manager Nar	Literal	QM_ARIBA01					

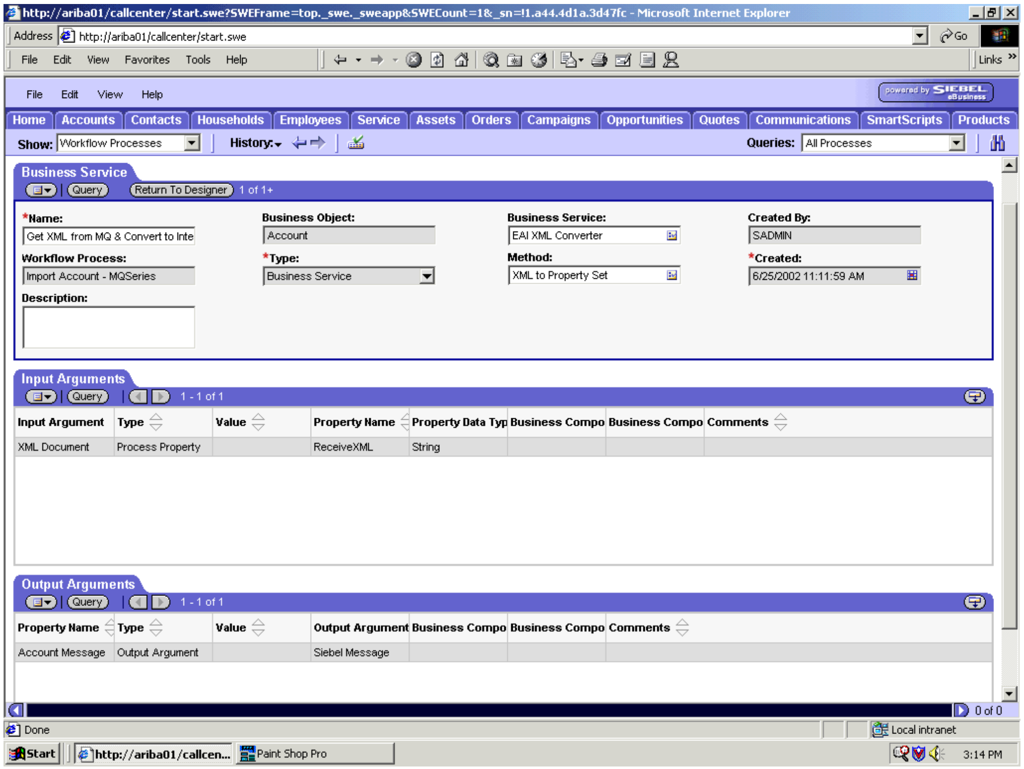
Property Name	Type	Value	Output Argument	Business Compo	Business Compo	Comments
ReceiveXML	Output Argument		Message Text			

2. Define an EAI MQSeries Server Transport Business Service step and call it `Receive`, as shown in Figure A–18.

   The Business Service is defined to receive the Account data from the MQSeries message queue.

   The EAI MQSeries Server Transport Business Service receives the Account data in Siebel XML format and sends it to the EAI XML Converter Business Service.

Figure A–19 Configuration of an EAI XML Converter Business Service


http://ariba01/callcenter/start.swe?SWEFrame=top._swe._sweapp&SWECount=1&_sn=!1.a44.4d1a.3d47fc - Microsoft Internet Explorer
Address http://ariba01/callcenter/start.swe
File Edit View Favorites Tools Help
Links
File Edit View Help
powered by SIEBEL eBusiness
Home Accounts Contacts Households Employees Service Assets Orders Campaigns Opportunities Quotes Communications SmartScripts Products
Show: Workflow Processes
History:
Queries: All Processes
Business Service
Query
Return To Designer
1 of 1+
Input Arguments
Output Arguments
1 - 1 of 1
0 of 0
Done
Local intranet
Start
http://ariba01/callcen...
Paint Shop Pro
3:14 PM

*Name: Get XML from MQ & Convert to Inte	Business Object: Account	Business Service: EAI XML Converter	Created By: SADMIN
Workflow Process: Import Account - MQSeries	*Type: Business Service	Method: XML to Property Set	*Created: 6/25/2002 11:11:59 AM
Description:			

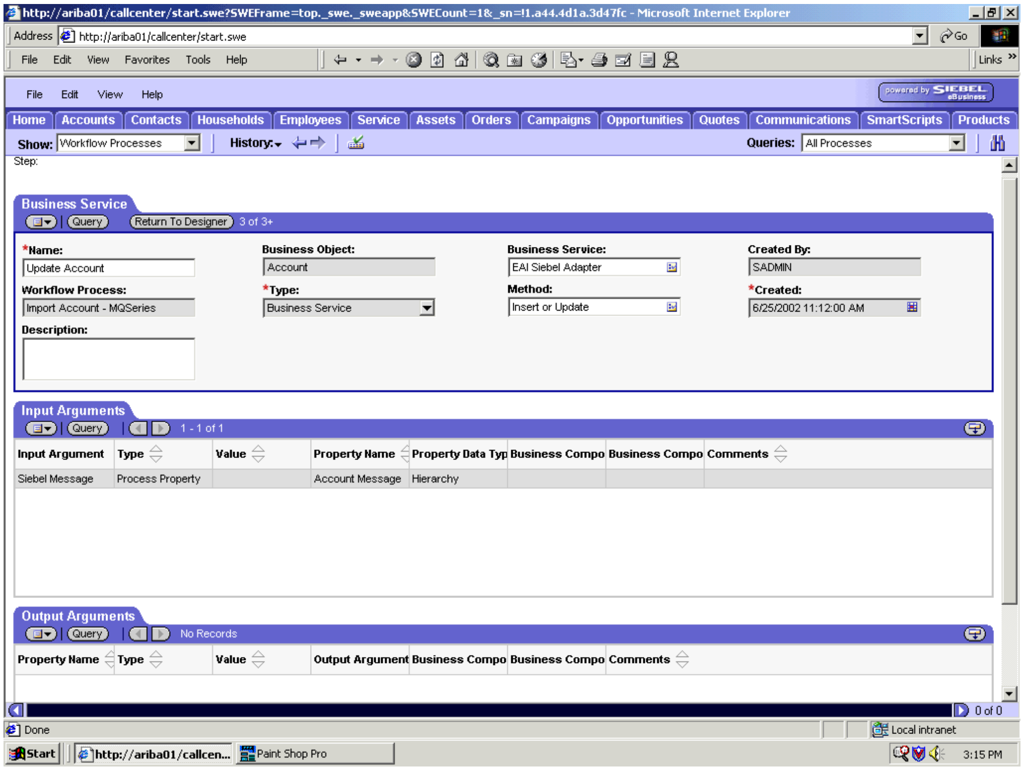
Input Argument	Type	Value	Property Name	Property Data Typ	Business Compo	Business Compo	Comments
XML Document	Process Property		ReceiveXML	String			

Property Name	Type	Value	Output Argument	Business Compo	Business Compo	Comments
Account Message	Output Argument		Siebel Message			

3. Define an EAI XML Converter Business Service step and call it `Get XML from MQ & Convert to XML`, as shown in Figure A–19.

   This Business Service is defined to receive the Account data from the EAI MQSeries Server Transport Business Service in XML format and convert it to hierarchical format.

Figure A–20 Update Account Configuration for the EAI XML Converter that Uses MQ Series Transport



http://ariba01/callcenter/start.swe?SWEFrame=top._swe._sweapp&SWECount=1&_sn=!1.a44.4d1a.3d47fc - Microsoft Internet Explorer
Address http://ariba01/callcenter/start.swe
File Edit View Favorites Tools Help
Links
File Edit View Help
powered by SIEBEL eBusiness
Home Accounts Contacts Households Employees Service Assets Orders Campaigns Opportunities Quotes Communications SmartScripts Products
Show: Workflow Processes
History:
Queries: All Processes
Step:

Business Service

Query
Return To Designer
3 of 3+

*Name: Update Account	Business Object: Account	Business Service: EAI Siebel Adapter	Created By: SADMIN
Workflow Process: Import Account - MQSeries	*Type: Business Service	Method: Insert or Update	*Created: 6/25/2002 11:12:00 AM
Description:			

Input Arguments

Input Argument	Type	Value	Property Name	Property Data Typ	Business Compo	Business Compo	Comments
Siebel Message	Process Property		Account Message	Hierarchy			

Output Arguments

Property Name	Type	Value	Output Argument	Business Compo	Business Compo	Comments

4. Define an EAI Siebel Adapter Business Service step and call it `Update Account`, as shown in Figure A–20.

   This Business Service is defined to receive from the EAI XML Converter Business Service the instance of Account data in hierarchical format.

   The Business Service applies the Account information into Siebel using the Insert or Update method.

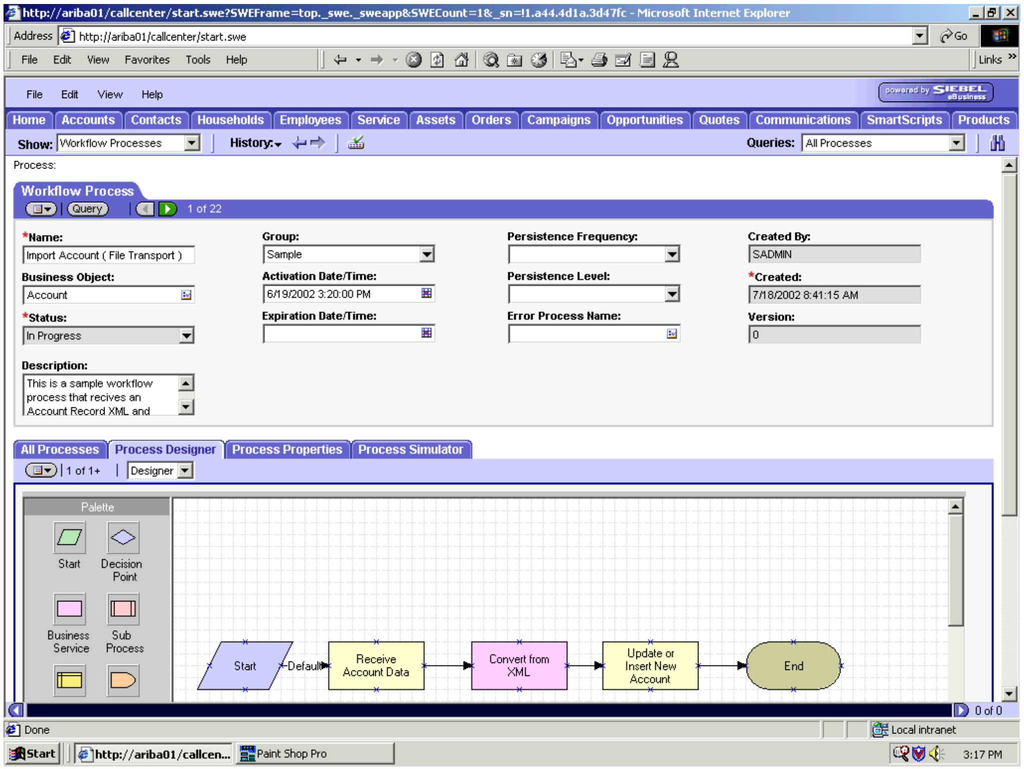
A.2.5 Creating a Siebel Workflow for a Service Using File Transport

The following procedure is an example of a Siebel Workflow illustrated in the Siebel Workflow Administration window. The workflow was designed for importing Siebel Account record information through the File transport

This procedure describes how to create a Siebel Workflow that generates Siebel XML when an Account record is updated in the Siebel Call Center application and then places Siebel XML on the file system.

The following is a Siebel Workflow Administration window with the Process Designer tab active, as shown in Figure A–21.

Figure A–21 Siebel Workflow Admininstration Window


http://ariba01/callcenter/start.swe?SWEFrame=top._swe._sweapp&SWECount=1&_sn=!1.a44.4d1a.3d47fc - Microsoft Internet Explorer
Address http://ariba01/callcenter/start.swe
File Edit View Favorites Tools Help
Links
File Edit View Help
Home Accounts Contacts Households Employees Service Assets Orders Campaigns Opportunities Quotes Communications SmartScripts Products
Show: Workflow Processes
History:
Queries: All Processes
Process:
Workflow Process
Query
1 of 22
All Processes
Process Designer
Process Properties
Process Simulator
1 of 1+
Designer
Palette
Start
Decision Point
Business Service
Sub Process
Start
Default
Receive Account Data
Convert from XML
Update or Insert New Account
End
0 of 0
Done
Local intranet
Start
http://ariba01/callcen...
Paint Shop Pro
3:17 PM

*Name: Import Account (File Transport)

Business Object: Account

*Status: In Progress

Description: This is a sample workflow process that recives an Account Record XML and

Group: Sample

Activation Date/Time: 6/19/2002 3:20:00 PM

Expiration Date/Time:

Persistence Frequency:

Persistence Level:

Error Process Name:

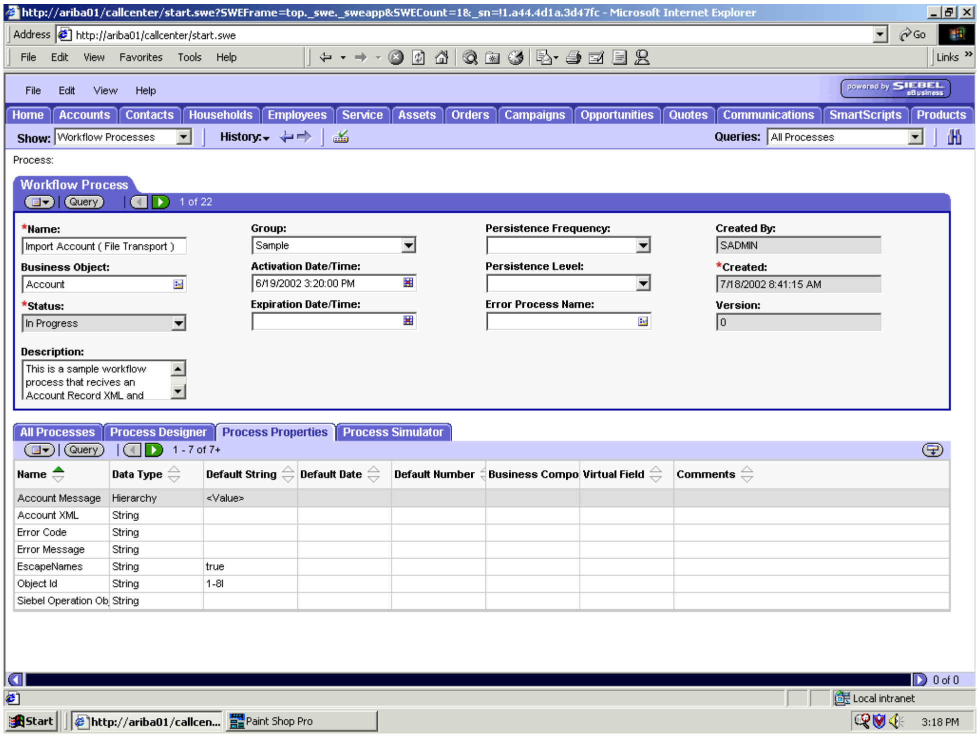
Created By: SADMIN

*Created: 7/18/2002 8:41:15 AM

Version: 0

To create a Siebel Workflow:

Figure A–22 Process Properties Tab of the Workflow Process Window


http://ariba01/callcenter/start.swe?SWEFrame=top._swe._sweapp&SWECount=1&_sn=!1.a44.4d1a.3d47fc - Microsoft Internet Explorer
Address http://ariba01/callcenter/start.swe
File Edit View Favorites Tools Help
Links
File Edit View Help
Home Accounts Contacts Households Employees Service Assets Orders Campaigns Opportunities Quotes Communications SmartScripts Products
Show: Workflow Processes
History:
Queries: All Processes
Process:
Workflow Process
Query
1 of 22
All Processes
Process Designer
Process Properties
Process Simulator
Query
1 - 7 of 7+
0 of 0
Local intranet
Start
http://ariba01/callcen...
Paint Shop Pro
3:18 PM

*Name: Import Account (File Transport)

Business Object: Account

*Status: In Progress

Description: This is a sample workflow process that recives an Account Record XML and

Group: Sample

Activation Date/Time: 6/19/2002 3:20:00 PM

Expiration Date/Time:

Persistence Frequency:

Persistence Level:

Error Process Name:

Created By: SADMIN

*Created: 7/18/2002 8:41:15 AM

Version: 0

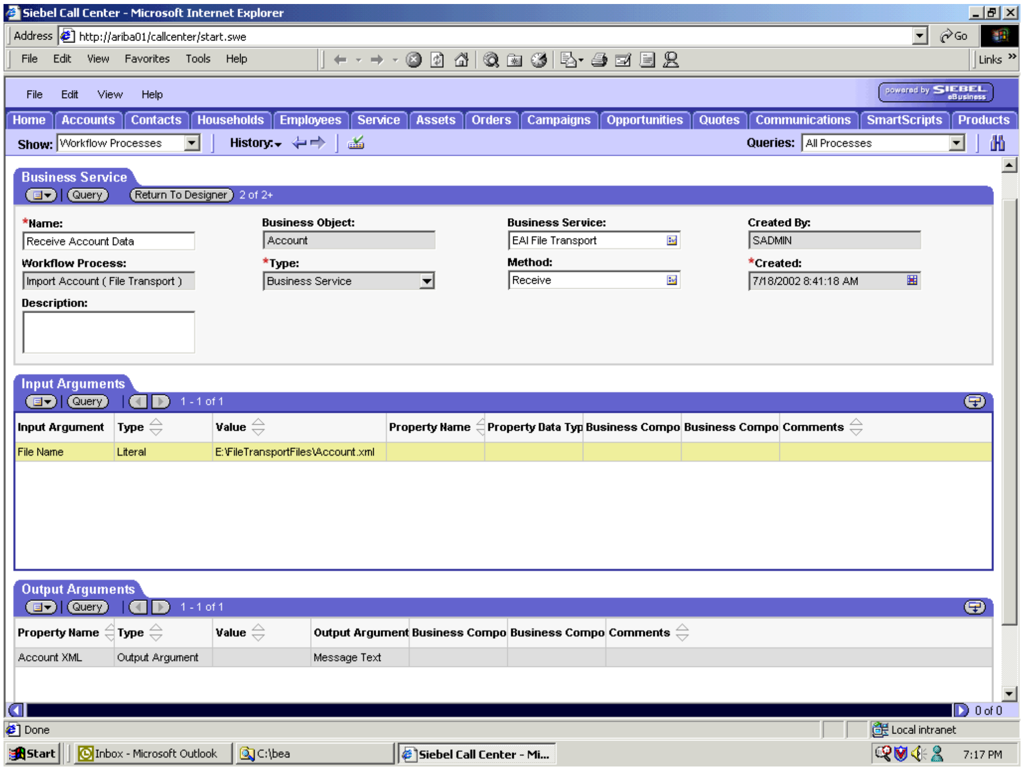
Name	Data Type	Default String	Default Date	Default Number	Business Compo	Virtual Field	Comments
Account Message	Hierarchy	<Value>					
Account XML	String						
Error Code	String						
Error Message	String						
EscapeNames	String	true					
Object Id	String	1-8I					
Siebel Operation Ob	String						

1. In the Process Properties tab of the Workflow Process window, define the Account message and Account XML process properties, as shown in Figure A–22.

   Account message contains the Siebel Account data in hierarchical format.

   Account XML specifies the Siebel Account data that the workflow converted to XML.

Figure A–23 EAI File Transport Business Service Step


Siebel Call Center - Microsoft Internet Explorer
Address http://ariba01/callcenter/start.swe
Go
Links
File Edit View Favorites Tools Help
File Edit View Help
powered by SIEBEL eBusiness
Home Accounts Contacts Households Employees Service Assets Orders Campaigns Opportunities Quotes Communications SmartScripts Products
History:
Business Service
Query
Return To Designer
2 of 2+
Input Arguments
Output Arguments
1 - 1 of 1
0 of 0
Done
Local intranet
Start
Inbox - Microsoft Outlook
C:\bea
Siebel Call Center - Mi...
7:17 PM

Show: Workflow Processes

Queries: All Processes

*Name: Receive Account Data

Workflow Process: Import Account (File Transport)

Description:

Business Object: Account

*Type: Business Service

Business Service: EAI File Transport

Method: Receive

Created By: SADMIN

*Created: 7/18/2002 8:41:18 AM

Input Argument	Type	Value	Property Name	Property Data Typ	Business Compo	Business Compo	Comments
File Name	Literal	E:\FileTransportFiles\Account.xml					

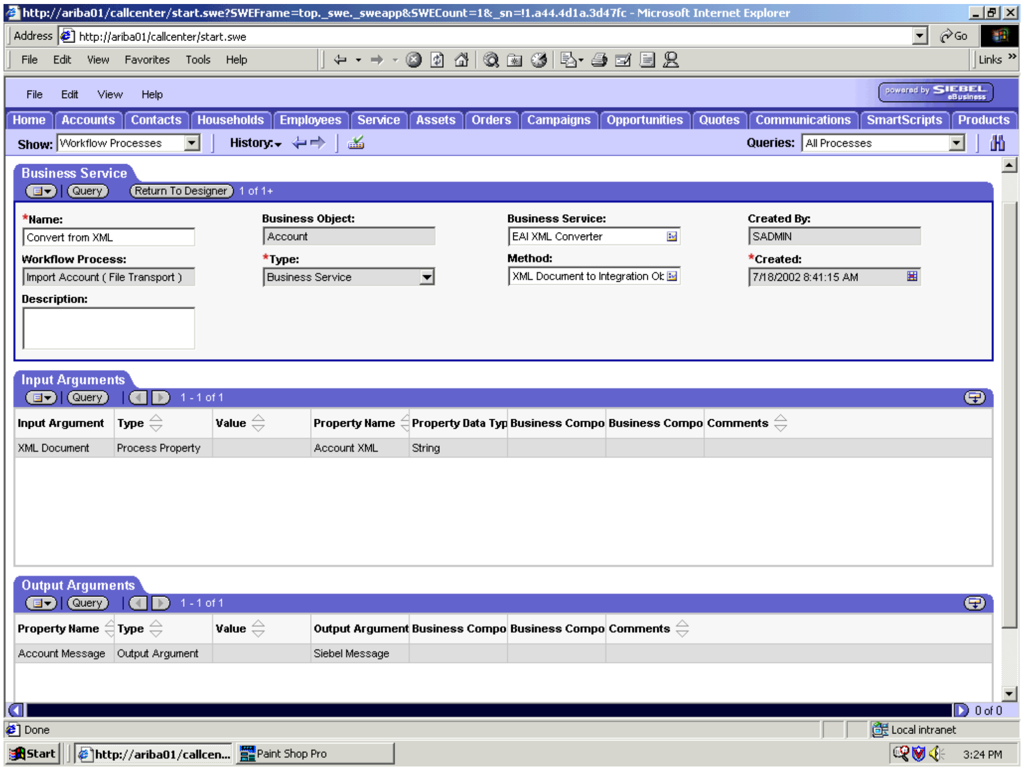
Property Name	Type	Value	Output Argument	Business Compo	Business Compo	Comments
Account XML	Output Argument		Message Text			

2. Define an EAI FileTransport Business Service step and call it `Receive Account Data`, as shown in Figure A–23.

   The Business Service is defined to receive the Account data from the file system.

   The EAI File Transport Business Service receives the Account data in Siebel XML format and sends it to the EAI XML Converter Business Service.

Figure A–24 EAI XML Converter Business Service Step


http://ariba01/callcenter/start.swe?SWEFrame=top._swe._sweapp&SWECount=1&_sn=!1.a44.4d1a.3d47fc - Microsoft Internet Explorer
Address http://ariba01/callcenter/start.swe
Go
File Edit View Favorites Tools Help
Links
File Edit View Help
powered by SIEBEL eBusiness
Home Accounts Contacts Households Employees Service Assets Orders Campaigns Opportunities Quotes Communications SmartScripts Products
History:
Query
Return To Designer
1 of 1+
1 - 1 of 1
0 of 0
Done
Local intranet
Start
http://ariba01/callcen...
Paint Shop Pro
3:24 PM

Show: Workflow Processes

Queries: All Processes

Business Service

*Name: Convert from XML	Business Object: Account	Business Service: EAI XML Converter	Created By: SADMIN
Workflow Process: Import Account (File Transport)	*Type: Business Service	Method: XML Document to Integration Ok	*Created: 7/18/2002 8:41:15 AM
Description:			

Input Arguments

Input Argument	Type	Value	Property Name	Property Data Typ	Business Compo	Business Compo	Comments
XML Document	Process Property		Account XML	String			

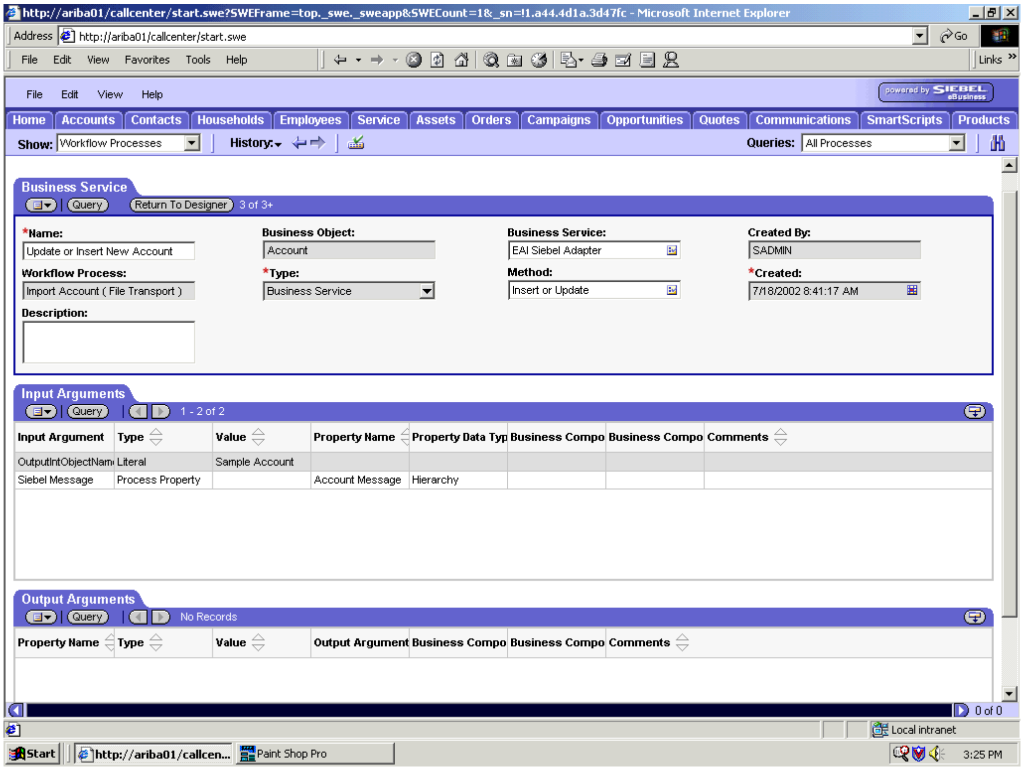
Output Arguments

Property Name	Type	Value	Output Argument	Business Compo	Business Compo	Comments
Account Message	Output Argument		Siebel Message			

3. Define an EAI XML Converter Business Service step and call it `Convert from XML`, as shown in Figure A–24.

   This Business Service is defined to receive the Account data from the EAI File Transport Business Service in XML format and convert it to hierarchical format.

Figure A–25 Creation of the EAI Adapter Business Service Step Called Update



http://ariba01/callcenter/start.swe?SWEFrame=top._swe._sweapp&SWECount=1&_sn=!1.a44.4d1a.3d47fc - Microsoft Internet Explorer
Address http://ariba01/callcenter/start.swe
Go
File Edit View Favorites Tools Help
Links
File Edit View Help
powered by SIEBEL eBusiness
Home Accounts Contacts Households Employees Service Assets Orders Campaigns Opportunities Quotes Communications SmartScripts Products
History:
Query
Return To Designer
Local intranet
Start
http://ariba01/callcen...
Paint Shop Pro
3:25 PM

Show: Workflow Processes

Queries: All Processes

Business Service

3 of 3+

*Name: Update or Insert New Account	Business Object: Account	Business Service: EAI Siebel Adapter	Created By: SADMIN
Workflow Process: Import Account (File Transport)	*Type: Business Service	Method: Insert or Update	*Created: 7/18/2002 8:41:17 AM
Description:			

Input Arguments

1 - 2 of 2

Input Argument	Type	Value	Property Name	Property Data Typ	Business Compo	Business Compo	Comments
OutputIntObjectNam	Literal	Sample Account					
Siebel Message	Process Property		Account Message	Hierarchy			

Output Arguments

No Records

Property Name	Type	Value	Output Argument	Business Compo	Business Compo	Comments

0 of 0

4. Define an EAI Siebel Adapter Business Service step and call it `Update` or `Insert New Account`, as shown in Figure A–25.

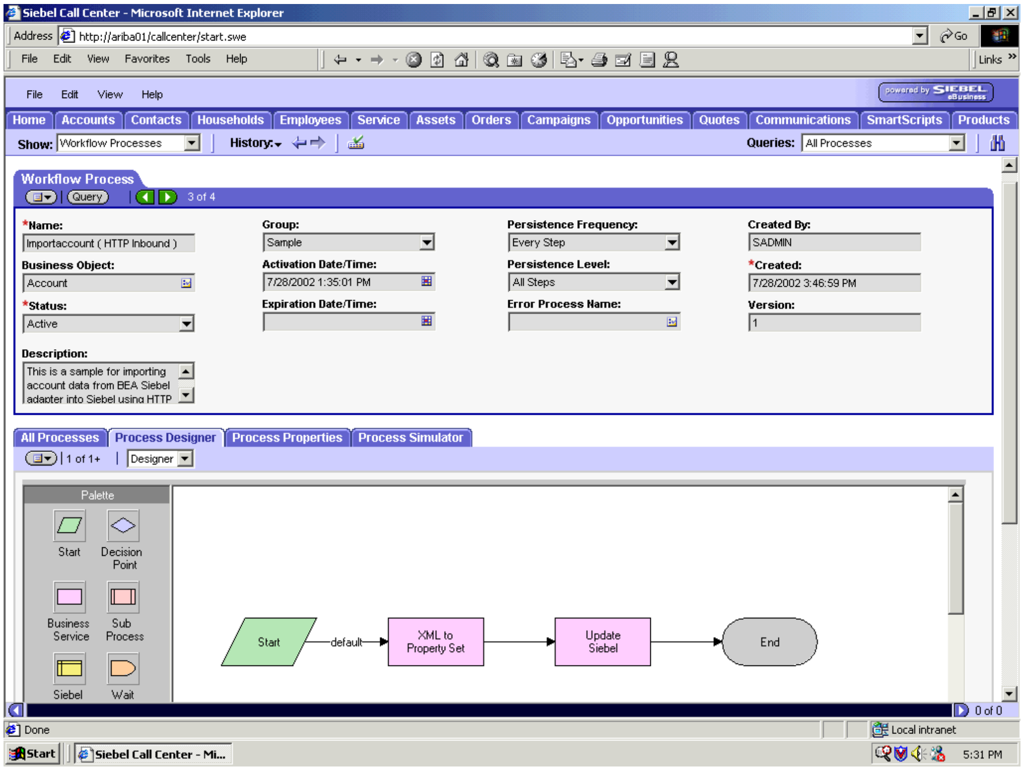
This Business Service is defined to receive from the EAI XML Converter Business Service the instance of Account data in hierarchical format.

The Business Service applies the Account information into Siebel using the Insert or Update method.

A.2.6 Creating a Siebel Workflow for a Service Using HTTP Transport

The following procedure is an example of a Siebel workflow illustrated in the Siebel Workflow Administration window, as shown in Figure A–26. The Workflow was designed for importing Siebel Account record information through the HTTP transport, as shown in .

Figure A–26 Siebel Workflow Administration Window



*Name: Importaccount (HTTP Inbound)

Business Object: Account

*Status: Active

Description: This is a sample for importing account data from BEA Siebel adapter into Siebel using HTTP

Group: Sample

Activation Date/Time: 7/28/2002 1:35:01 PM

Expiration Date/Time:

Persistence Frequency: Every Step

Persistence Level: All Steps

Error Process Name:

Created By: SADMIN

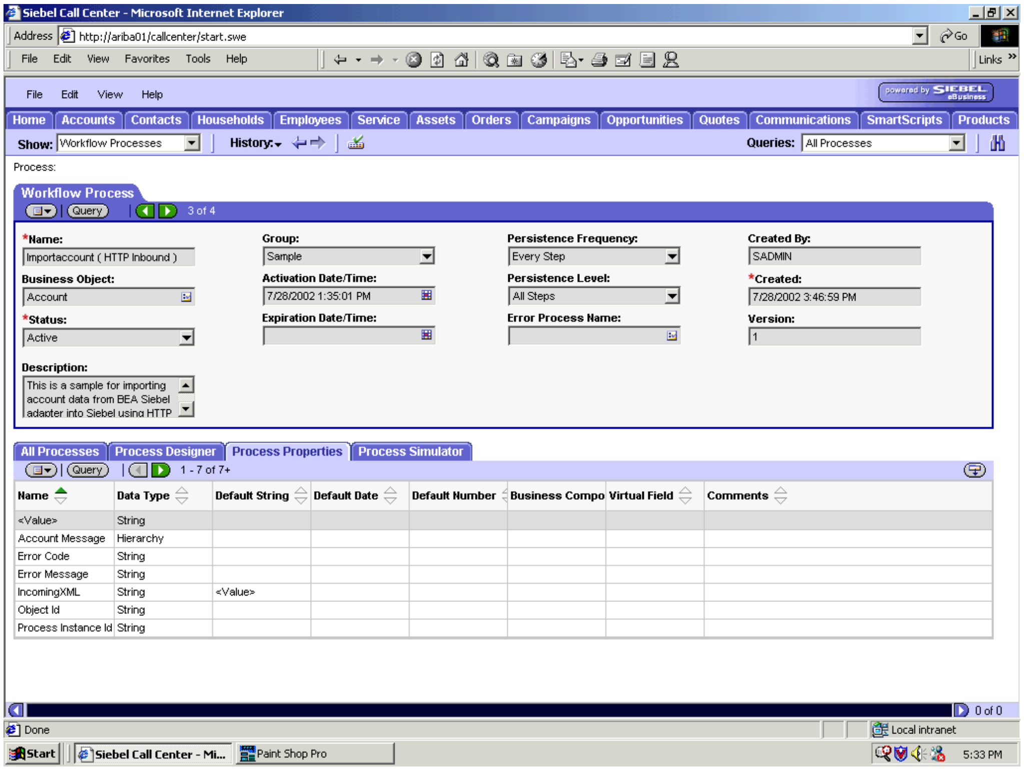
*Created: 7/28/2002 3:46:59 PM

Version: 1

The following procedure describes how to create a Siebel Workflow that generates Siebel XML when an Account record is updated in the Siebel Call Center application and then places Siebel XML on the file system.

To create a Siebel Workflow:

Figure A–27 Process Properties Tab of the Workflow Process Window


Siebel Call Center - Microsoft Internet Explorer
Address http://ariba01/callcenter/start.swe
Go
Links
File Edit View Favorites Tools Help
File Edit View Help
powered by SIEBEL eBusiness
Home Accounts Contacts Households Employees Service Assets Orders Campaigns Opportunities Quotes Communications SmartScripts Products
History:
Queries:
Workflow Process
Query
3 of 4
All Processes
Process Designer
Process Properties
Process Simulator
Query
1 - 7 of 7+
0 of 0
Done
Local intranet
Start
Siebel Call Center - Mi...
Paint Shop Pro
5:33 PM

Show: Workflow Processes

Queries: All Processes

Process:

*Name: Importaccount (HTTP Inbound)	Group: Sample	Persistence Frequency: Every Step	Created By: SADMIN
Business Object: Account	Activation Date/Time: 7/28/2002 1:35:01 PM	Persistence Level: All Steps	*Created: 7/28/2002 3:46:59 PM
*Status: Active	Expiration Date/Time:	Error Process Name:	Version: 1

Description: This is a sample for importing account data from BEA Siebel adapter into Siebel using HTTP

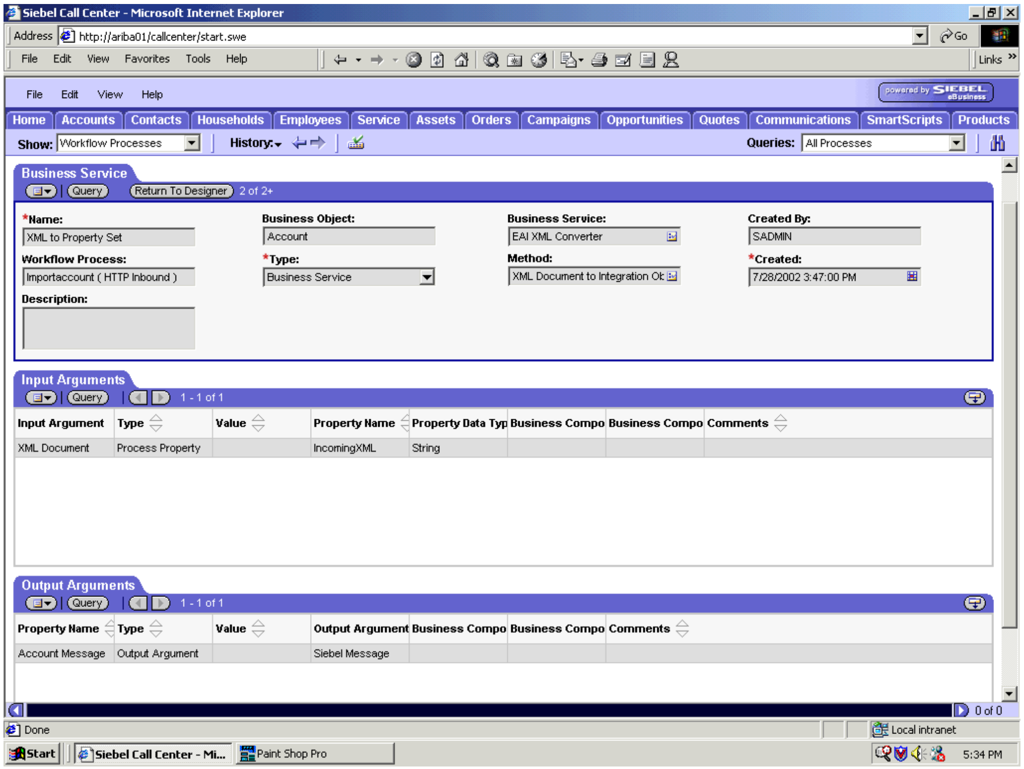
Name	Data Type	Default String	Default Date	Default Number	Business Compo	Virtual Field	Comments
<Value>	String						
Account Message	Hierarchy						
Error Code	String						
Error Message	String						
IncomingXML	String	<Value>					
Object Id	String						
Process Instance Id	String						

1. In the Process Properties tab of the Workflow Process window, define the Account message and Account XML process properties, as shown in Figure A–27.

   Account message contains the Siebel Account data in hierarchical format.

   Account XML specifies the Siebel Account data that the workflow converted to XML.

Figure A–28 EAI XML Converter Business Service Step



Siebel Call Center - Microsoft Internet Explorer
Address http://ariba01/callcenter/start.swe
File Edit View Favorites Tools Help
Links
File Edit View Help
Home Accounts Contacts Households Employees Service Assets Orders Campaigns Opportunities Quotes Communications SmartScripts Products
History:
Query
Return To Designer
2 of 2+
1 - 1 of 1
0 of 0
Done
Local intranet
Start
Siebel Call Center - Mi...
Paint Shop Pro
5:34 PM

Show: Workflow Processes

Queries: All Processes

Business Service

*Name: XML to Property Set	Business Object: Account	Business Service: EAI XML Converter	Created By: SADMIN
Workflow Process: Importaccount (HTTP Inbound)	*Type: Business Service	Method: XML Document to Integration Ok	*Created: 7/28/2002 3:47:00 PM
Description:			

Input Arguments

Input Argument	Type	Value	Property Name	Property Data Typ	Business Compo	Business Compo	Comments
XML Document	Process Property		IncomingXML	String			

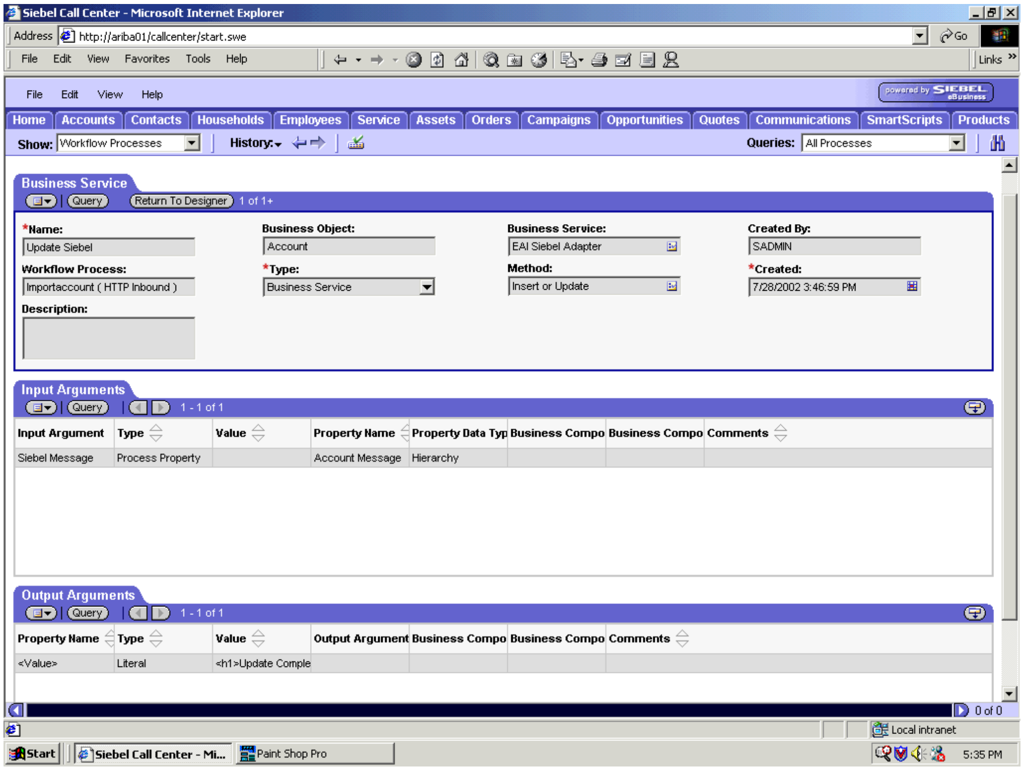
Output Arguments

Property Name	Type	Value	Output Argument	Business Compo	Business Compo	Comments
Account Message	Output Argument		Siebel Message			

2. Define an EAI XML Converter Business Service step and call it `XML to Property Set`, as shown in Figure A–28.

   The Business Service is defined to receive the Account data from the EAI HTTP Transport Business Service in XML format and convert it to hierarchical format.

Figure A–29 EAI Adapter Business Service Update Step


Siebel Call Center - Microsoft Internet Explorer
Address http://ariba01/callcenter/start.swe
Go
File Edit View Favorites Tools Help
Links
File Edit View Help
powered by SIEBEL eBusiness
Home Accounts Contacts Households Employees Service Assets Orders Campaigns Opportunities Quotes Communications SmartScripts Products
History:
Query
Return To Designer
1 of 1+
1 - 1 of 1
0 of 0
Local intranet
Start
Siebel Call Center - Mi...
Paint Shop Pro
5:35 PM

Show: Workflow Processes

Queries: All Processes

Business Service

*Name: Update Siebel	Business Object: Account	Business Service: EAI Siebel Adapter	Created By: SADMIN
Workflow Process: Importaccount (HTTP Inbound)	*Type: Business Service	Method: Insert or Update	*Created: 7/28/2002 3:46:59 PM
Description:			

Input Arguments

Input Argument	Type	Value	Property Name	Property Data Typ	Business Compo	Business Compo	Comments
Siebel Message	Process Property		Account Message	Hierarchy			

Output Arguments

Property Name	Type	Value	Output Argument	Business Compo	Business Compo	Comments
<Value>	Literal	<h1>Update Comple				

3. Define an EAI Siebel Adapter Business Service step and call it `Update Siebel`, as shown in Figure A–29.

   The Business Service is defined to receive from the EAI XML Converter Business Service the instance of Account data in hierarchical format.

   The Business Service applies the Account information into Siebel using the Insert or Update method.

Glossary

adapter

Provides universal connectivity by enabling an electronic interface to be accommodated (without loss of function) to another electronic interface.

agent

Supports service protocols in listeners and documents.

business service

Also known as a Web service. A Web service is a self-contained, modularized function that can be published and accessed across a network using open standards. It is the implementation of an interface by a component and is an executable entity.

channel

Represents configured connections to particular instances of back-end systems. A channel binds one or more event ports to a particular listener managed by an adapter.

listener

A component that accepts requests from client applications.

port

Associates a particular business object exposed by the adapter with a particular disposition. A disposition is a URL that defines the protocol and location of the event data. The port defines the end point of the event consumption.

Index

A

adapter configuration
  updating, 3-3
adapter exceptions, 10-4
adapter types
  resource, 3-3
adapters, 1-1
  configuring, 3-3
  deploying, 3-1 to 3-3, 4-2
  integrating with BPEL Process Manager, 4-1, 6-1
  troubleshooting, 10-1 to 10-7
Adapters node, 4-38, 4-40
Add Channel dialog box, 2-29, 4-35
Add Target dialog box, 2-6
Advanced tab, 2-7, 2-33, 2-35
ae batch script, 10-2
application adapters
  configuring, 3-3
  deploying, 3-1 to 3-3, 4-2
  integrating with BPEL Process Manager, 4-1, 6-1
  troubleshooting, 10-1 to 10-7
Application Explorer, 1-2, 1-5, 2-10 to 2-12, 4-7
  debugging and, 10-2 to 10-3
  J2CA configuration and, 3-3
  OracleAS Adapter J2CA and, 10-4
  targets and, 4-40
  testing and, 10-2 to 10-3
  troubleshooting, 10-2 to 10-3
  WSDL files and, 4-34, 4-35
application system objects
  viewing, 2-10

B

BPEL Designer, 4-1, 6-1
BPEL Process Manager
  integrating with adapters, 4-1, 6-1
  OracleAS Adapter for Siebel and, 4-1
browsing metadata, 2-9
BSE (OracleAS Adapter Business Services Engine)
  configuring, 2-3
  troubleshooting, 10-4
BSE URL field, 2-4
Business Components, 1-2 to 1-6, 2-13, 2-24
business events, 1-2
Business Integration Manager, 1-5
Business Objects, 1-4, 2-24
  browsing, 2-9
Business Service list, 2-18
business services, 1-2 to 1-6, 2-13, A-1 to A-3
  browsing, 2-9
  creating, 2-24 to 2-25
  testing, 2-25 to 2-26

C

CCI (Common Client Interface), 3-1
CCI calls, 3-1
channel configuration parameters
  Document type XML, 2-32
  Error Directory, 2-33, 2-35
  File Mask, 2-34
  Message wait interval, 2-33
  Mode of operation, 2-33
  MQ server channel, 2-32
  MQ server host, 2-32
  MQ server port, 2-32
  Poll interval, 2-35
  Polling Location, 2-34
  Port, 2-30, 4-37
  Processing Mode, 2-35
  Queue manager name, 2-32
  Request queue name, 2-33
  Response/Ack Directory, 2-35
  Server port, 2-30, 4-37
  Synchronization Type, 2-30, 2-33, 2-35, 4-37
  Thread limit, 2-33, 2-35
Channel field, 4-41
channel types
  File, 2-34
  HTTP, 2-29, 4-35 to 4-38
  MQSeries, 2-32
channels
  creating, 1-2, 2-29 to 2-35, 4-35 to 4-38
  deleting, 2-36
  editing, 2-35
  starting, 2-32 to 2-35, 4-38
  stopping, 2-32 to 2-35, 4-38
  testing and debugging, 4-38
Channels node, 2-29, 4-35
channels. *See also* listeners

closing connections, 2-9
COM environment, 10-2
Common Client Interface (CCI), 3-1
configuration parameters, 3-4
  IWayConfig, 3-4
  IWayHome, 3-4
  IWayRepoPassword, 3-4
  IWayRepoURL, 3-4
  IWayRepoUser, 3-4
  Loglevel, 3-4
configurations
  connecting to, 2-5
  defining, 2-3 to 2-4
  overwriting, 3-4
Configurations node, 2-4
configuring adapters, 3-3
configuring events, 2-28 to 2-35
connecting to OracleAS Adapter J2CA, 10-4
connecting to Siebel, 2-5 to 2-9, 4-38, 4-40, 10-2 to 10-4, 10-6
Connection dialog box, 4-38
connection parameters, 10-2
  Language, 2-7
  Object Manager, 2-7
  Repository Manager, 2-7
connection types, 10-2
connections
  closing, 2-9
  deleting, 2-10
  establishing, 2-5 to 2-9, 2-19, 4-38
Connector
  deploying to Oracle Application Server, 3-1
connector factories, 3-3
connector factory objects, 3-3
  multiple, 3-5
Create Web Service dialog box, 2-24
creating channels, 4-35 to 4-38
creating events, 4-40 to 4-41
creating repository projects, 2-3 to 2-4
creating schemas, 2-12 to 2-14
creating Web services, 2-24 to 2-25

D

database connections
  opening, 3-4
databases
  connecting to, 3-4
  Oracle, 3-4
defining targets to Siebel, 2-6
deleting channels, 2-36
deleting targets, 2-10
deploying adapters, 3-1 to 3-3, 4-2
Description field, 2-24
design time, 2-3, 4-1, 4-34
  configuring, 4-7
disconnecting from Siebel, 2-9
Document type XML parameter, 2-32

E

EAI (Enterprise Application Integration), 1-5
Eclipse. *See* JDeveloper
editing channels, 2-35
editing targets, 2-10
EJB (Enterprise Java Beans), 3-1
Enterprise Application Integration (EAI), 1-5
Enterprise Connector for J2EE Connector Architecture (J2CA), 1-1
Enterprise Java Beans (EJB), 3-1
Envelope type list, 2-18
Error Directory parameter, 2-33, 2-35
error messages, 10-2 to 10-7
  target systems and, 10-5
eScript script, A-1
event adapters, 2-28 to 2-35
event integration, 4-34
event messages, 4-34, 4-35
event ports
  creating, 4-40 to 4-41
event schemas, 2-16
events, 1-2
  configuring, 2-28 to 2-35
  creating, 4-40 to 4-41
  triggering, 4-49 to 4-53
  workflows and, A-4 to A-16
Events node, 2-29
Existing Service Names list, 2-24
Export WSDL dialog box, 4-8

F

fault code elements, 10-5
fault string elements, 10-5
File channel, 2-34
File Mask parameter, 2-34
File transport, A-8 to A-13, A-21

G

Generate XML Schema wizard, 2-18

H

Hostname parameter, 2-4
HTTP channel, 2-29, 4-35 to 4-38
HTTP transport, A-13, A-25

I

inbound J2CA services, 4-34, 4-35
inbound processing, 1-1 to 1-5, A-3
input arguments, 4-52
Input Arguments tab, 4-52
installation directories, 3-4
integration access methods
  OracleAS Adapter for Siebel and, 1-5
Integration Object (IO) node, 2-20
  creating, 4-39
Integration Objects, 1-2, 2-17

browsing, 2-9
IO (Integration Object) node, 2-20
creating, 4-39
IWayConfig parameter, 3-4
IWayHome parameter, 3-4
IWayRepoPassword parameter, 3-4
IWayRepoURL parameter, 3-4
IWayRepoUser parameter, 3-4

J

J2CA (Enterprise Connector for J2EE Connector Architecture), 1-1
J2CA architecture
Oracle Application Server Adapter and, 3-1
J2CA configuration
Application Explorer and, 3-3
J2CA resource adapters, 1-1, 3-1
J2CA services, 4-34, 4-35
JAR files, 10-2
Java Data Bean, 1-5
Java program clients, 3-1
JDeveloper, 6-1

L

Language parameter, 2-7, 10-2
License and Method dialog box, 2-24
License field, 2-24
licenses, 2-24
list of nodes, 10-2
listeners. *See also* channels
log files, 10-1
log levels
overwriting, 3-4
Loglevel parameter, 3-4
logon parameters, 10-2

M

managed connector factories, 3-3
managed connector factory objects, 3-3
multiple, 3-5
ManagedConnectionFactory parameter, 3-4
Mediator Inbound Process, 5-11
Mediator Outbound Process, 5-2
message interactions
asynchronous, 1-2
synchronous, 1-2
message types
event, 4-34, 4-35
Message wait interval parameter, 2-33
messages, 1-2
metadata
browsing, 2-9
viewing, 2-10
metadata tables, 2-11
Method Name field, 2-24
Methods node, 2-25
Mode of operation parameter, 2-33
MQ server channel parameter, 2-32
MQ server host parameter, 2-32
MQ server port parameter, 2-32
MQSeries channel, 2-32
MQSeries transport, A-4, A-16

N

navigation paths, 1-4
New Configuration dialog box, 2-3 to 2-4
Node list, 10-2
Node name field, 4-39
nodes
Adapters, 4-38, 4-40
Channels, 2-29, 4-35
Configurations, 2-4
connected, 2-9, 4-38
disconnected, 2-9
Events, 2-29
Integration Object, 2-20, 4-39
Methods, 2-25
Ports, 4-35
Sample Account, 2-20, 4-39
Services, 2-25
Siebel, 4-38 to 4-40

O

Object Manager, 1-2
Object Manager parameter, 2-7
Oracle Application Server
deployment of Connector to, 3-1
Oracle Application Server Adapter
J2CA architecture and, 3-1
troubleshooting, 10-6
Oracle databases, 3-4
Oracle JDeveloper, 6-1
Oracle JDeveloper BPEL Designer. *See* BPEL Designer, JDeveloper, or Oracle JDeveloper
Oracle's Unified Method (OUM), ix
OracleAS Adapter
installation directory and, 3-4
OracleAS Adapter Application Explorer. *See* Application Explorer
OracleAS Adapter Business Services Engine (BSE), 1-1, 2-3
troubleshooting, 10-4
OracleAS Adapter for Siebel
BPEL Process Manager and, 4-1
deploying, 1-1, 4-2
integration access methods and, 1-5
troubleshooting, 10-1 to 10-7
OracleAS Adapter J2CA, 2-3 to 2-4
Application Explorer and, 10-4
connecting to, 10-4
outbound processing, 1-1 to 1-5, A-2

P

parameter types
channel configuration, 2-30 to 2-35, 4-37
configuration, 3-4

connection, 2-7, 10-2
Password parameter, 10-2
passwords, 3-4
Poll interval parameter, 2-35
Polling Location parameter, 2-34
Port Number parameter, 2-4
Port parameter, 2-30, 4-37
ports
creating, 2-28, 4-40 to 4-41
Ports node, 4-35
Process Designer tab, 4-52
Process Manager. *See* BPEL Process Manager
Process Properties tab, A-4, A-23, A-27
Process Simulator tab, 4-53
Processing Mode parameter, 2-35
projects
repository, 2-3 to 2-4
properties, 3-4
Protocol list, 2-20, 4-36, 4-39

Q

Query method, A-11
Queue manager name parameter, 2-32

R

Repository Manager parameter, 2-7
Repository Name parameter, 10-3 to 10-4
repository projects
creating, 2-3 to 2-4
Web services and, 2-3
REQUEST option, 2-33
Request queue name parameter, 2-33
request schemas, 2-13
Request tab, 2-32, 2-34
REQUEST_ACK option, 2-33
REQUEST_RESPONSE option, 2-33
requests
executing, 10-4
response schemas, 2-13
Response tab, 2-33 to 2-35
Response/Ack Directory parameter, 2-35
runtime, 2-3, 4-38 to 4-41
runtime events (Siebel Events), A-1
runtime integration, 4-49

S

Sample Account node, 2-20, 4-39
schema considerations, 2-13
Schema location field, 4-39
schemas
creating, 2-12 to 2-14, 2-16
scripts, A-1
Server port parameter, 2-30, 4-37
service names, 2-24
Service Provider list, 2-3 to 2-4
service schemas
creating, 2-12 to 2-14
Service-Oriented Architecture (SOA), 4-1
services, 1-2
creating, 2-24 to 2-25
testing, 2-25 to 2-26
workflows and, A-16 to A-29
Services node, 2-25
servlets, 3-1
Siebel
connecting to, 1-4, 2-5 to 2-9, 4-38, 10-2 to 10-4, 10-6
disconnecting from, 2-9
Siebel Business Components, 1-2 to 1-6, 2-13, 2-24
Siebel Business Objects, 1-4, 2-24
browsing, 2-9
Siebel business processes
invoking, 1-4
Siebel Business Services, 1-2 to 1-6, 2-13, A-1 to A-3
browsing, 2-9
Siebel Call Center
starting, 4-49
Siebel Client Workflow Administration screens, A-1
Siebel COM Data Interface, 1-5
Siebel Events (run-time events), A-1
Siebel Gateway server, 10-2
Siebel Integration Objects, 1-2, 2-17
browsing, 2-9
Siebel Java Data Bean, 1-5
Siebel node, 4-38 to 4-40
Siebel Object Manager, 1-2
Siebel Tools Schema Wizard, 2-16, 2-19
Siebel Tools window, 2-17
Siebel transports, 1-5
Siebel VB script, A-1
Siebel Workflow Administration window, 4-50
Siebel workflow policies, A-2
Siebel Workflows, 1-5, A-1 to A-29
creating, A-3
SiebelJI_Common.jar file, 10-2
SiebelJI_enu.jar file, 10-2
SOA (Service-Oriented Architecture), 4-1
SOAP agents, 10-5
SOAP faults, 10-5
SOAP requests, 10-5 to 10-7
errors and, 10-5 to 10-7
SOAP responses, 10-5 to 10-7
starting the Siebel Call Center, 4-49
Synchronization Type parameter, 2-30, 2-33 to 2-35, 4-37
system objects
viewing, 2-10

T

target systems
errors and, 10-5
Target Type list, 2-7
targets
connecting to, 2-5 to 2-9, 4-38, 4-40, 10-2 to 10-4
defining, 2-6
deleting, 2-10
disconnecting from, 2-9

editing, 2-10
testing Web services, 2-25 to 2-26
thin clients, 10-2
Thread limit parameter, 2-33, 2-35
trace information, 10-1
transaction processing, 1-2
transports, 1-5
File, A-8 to A-13, A-21
HTTP, A-13, A-25
MQSeries, A-4, A-16
triggering an event, 4-49 to 4-53
troubleshooting, 10-1 to 10-7
Application Explorer, 10-2 to 10-3
BSE, 10-4
Web services, 10-4 to 10-7

U

updating adapter configuration, 3-3
UserName parameter, 10-2

V

viewing system objects, 2-10

W

Web Service Definition Language (WSDL), 2-27 to 2-28
Web service names, 2-24
Web services, 1-1 to 1-2, 1-6, 2-3
creating, 2-24 to 2-25
integrating, 4-1
repository projects and, 2-3
testing, 2-25 to 2-26
troubleshooting, 10-4 to 10-7
workflow elements, 4-52
workflow policies, A-2
Workflow Process window, A-4 to A-23, A-27
Workflow Processes window, 4-50
Workflows, A-1 to A-29
creating, A-3
WSDL (Web Service Definition Language), 2-27 to 2-28
WSDL documents, 4-1
WSDL File Name field, 4-41
WSDL files, 4-1, 6-1
Application Explorer and, 4-34, 4-35
creating, 4-35

X

XDR schemas
creating, 2-16
XML documents, 1-4, A-2
XML messages, 1-2, A-2 to A-3
XML schemas, 1-2, 1-6
creating, 2-12 to 2-14, 2-16