

Oracle® Fusion Middleware Application Adapters

Application Adapter for SAP R/3 User's Guide for

12c (12.2.1.2.0)

E84213-01

December 2016

Provides information on how to integrate with SAP R/3 systems and develop applications.

Oracle Fusion Middleware Application Adapter 12c (12.2.1.2.0) for SAP R/3 User's Guide for Oracle WebLogic Server, 12c (12.2.1.2.0)

E84213-01

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

Primary Author: Stefan Kostial

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

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

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

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Contents

Preface	xi
Audience	xi
Documentation Accessibility	xi
Related Documents	xi
Conventions	xii
1 Introduction	
Adapter Features	1-1
Supported Versions and Platforms	1-3
Classical SAP Technologies for ABAP	1-3
Integration with SAP R/3	1-4
Adapter Architecture	1-5
Oracle Adapter Business Services Engine (BSE) Architecture	1-5
Oracle Adapter J2CA Architecture	1-6
BSE Versus Oracle Adapter J2CA Deployment	1-7
Sample Projects	1-8
Quick Start Guide	1-9
Installation	1-9
Copying Third-Party Library Files	1-10
Configuration	1-10
WebLogic Server Deployment and Integration	1-11
Creating Configurations, Targets, and Channels in Application Explorer	1-11
Working With Service Components in the SOA Suite	1-12
Working With Oracle Service Bus	1-12
Working With Synchronous SAP R/3 Events	1-12
Additional Features	1-13
Other Features	1-13
2 Getting Started	
Verifying the SAP Java Connector (SAP JCo)	2-1
Verifying SAP JCo on Windows Platforms	2-1
Verifying SAP JCo on UNIX Platforms	2-2
Identifying SAP R/3 Logon Parameters	2-3
User Parameters	2-3
System Settings (Application Server) Parameters	2-4

System Settings (Message Server) Parameters.....	2-4
Connection Pool Parameters	2-5
SAP Gateway Parameters	2-5
ALE Parameters.....	2-6
Global Processing Parameters.....	2-6
SNC Parameters	2-7

3 SAP Java Connector 3.0xx Considerations

Supported Platforms	3-1
Connection Management	3-2
Connection Pooling - Events	3-2
Connection Pooling - Services	3-3
Single Server Support	3-6
Server Threads	3-6
Uniqueness.....	3-6
Load Balancing Adapter Configuration	3-6
Load Balancing SAP Configuration.....	3-6
Changing Parameters and Table Parameters	3-7
Trace Level Parameter	3-7

4 Configuring Oracle Application Adapter for SAP R/3

Understanding SAP R/3 Nested Structures and Nested Tables	4-1
Starting Application Explorer	4-4
Configuring Repository Settings	4-5
Creating a Repository Configuration	4-5
Creating a Configuration for BSE	4-5
Creating a Configuration for J2CA.....	4-6
Connecting to a BSE or J2CA Configuration.....	4-7
Establishing a Connection (Target) for SAP R/3	4-8
Defining a Target to SAP R/3	4-8
Connecting to a Defined SAP R/3 Target	4-15
Managing a Connection to SAP R/3.....	4-15
Viewing Application System Objects	4-17
Creating XML Schemas	4-17
Generating WSDL (J2CA Configurations Only)	4-18
Creating and Testing a Web Service (BSE Configurations Only)	4-19
Creating a Web Service	4-20
Testing a Web Service.....	4-21
Identity Propagation.....	4-21
Configuring an Event Adapter	4-22
Creating and Editing a Channel.....	4-22
Schema Validation	4-28
Configuring the SAProuter Service	4-30
Configuring SAProuter	4-30
Testing SAProuter With Oracle Components.....	4-31
Verifying SAProuter Configuration	4-32

5 Oracle WebLogic Server Deployment and Integration

Adapter Integration with Oracle WebLogic Server	5-1
Deployment of Adapter	5-1
Updating Adapter Configuration.....	5-2
Creating a Managed Connector Factory Object.....	5-2
Creating Multiple Managed Connector Factory Objects.....	5-4
Modifying WSDL Files for Additional Connection Factory Values.....	5-6

6 Integration With BPEL Service Components in the Oracle SOA Suite

Overview	6-1
Deployment of Adapter	6-2
Configuring a New Application Server Connection.....	6-2
Designing an Outbound BPEL Process for Service Integration (J2CA Configuration)	6-7
Generating WSDL for Request/Response Service	6-8
Creating an Empty Composite for SOA	6-9
Defining a BPEL Outbound Process.....	6-11
Deploying the BPEL Outbound Process	6-30
Invoking the Input XML Document in the Oracle Enterprise Manager Console.....	6-33
Testing Outbound BPEL and Mediator Processes	6-36
Designing an Inbound BPEL Process for Event Integration (J2CA Configuration)	6-37
Generating WSDL for Event Integration	6-37
Creating an Empty Composite for SOA	6-45
Defining a BPEL Inbound Process.....	6-45
Deploying the BPEL Inbound Process	6-53
Triggering an Event in SAP R/3	6-54
Working With Synchronous SAP R/3 Events Using BPEL (J2CA Configuration).....	6-55
Overview	6-56
Configuring the Adapter Target	6-56
Configuring the Channel	6-57
Generating WSDL for a Message.....	6-58
Creating a BPEL Process in JDeveloper	6-58
Designing an Outbound BPEL Process for Service Integration (BSE Configuration)	6-63
Generating a WSDL File for Request and Response Services Using a Web Service	6-63
Creating an Empty Composite for SOA	6-65
Defining a BPEL Outbound Process.....	6-65

7 Integration With Mediator Service Components in the Oracle SOA Suite

Configuring a New Application Server Connection.....	7-2
Configuring a Mediator Outbound Process (J2CA Configuration)	7-2
Creating an Empty Composite for SOA	7-2
Defining a Mediator Outbound Process	7-3
Deploying the Mediator Outbound Process	7-13
Invoking the Input XML Document in the Oracle Enterprise Manager Console.....	7-13
Configuring a Mediator Inbound Process (J2CA Configuration).....	7-13
Creating an Empty Composite for SOA	7-14
Defining a Mediator Inbound Process	7-14

Configuring a Mediator Outbound Process (BSE Configuration)	7-20
Creating an Empty Composite for SOA	7-21
Defining a Mediator Outbound Process	7-21

8 Integration With BPM Service Components in the Oracle SOA Suite

Overview	8-1
Deployment of Adapter	8-2
Configuring a New Application Server Connection	8-2
Designing an Outbound BPM Process Using Transformations for Service Integration (J2CA Configuration)	8-2
Creating an Empty Composite for BPM	8-2
Defining a BPM Outbound Process	8-3
Adjusting for Known Deployment Issues With 12c	8-25
Deploying the BPM Outbound Process	8-25
Invoking the Input XML Document in the Oracle Enterprise Manager Console	8-25
Designing an Inbound BPM Process Using Transformations for Event Integration (J2CA Configuration)	8-28
Creating an Empty Composite for BPM	8-28
Defining a BPM Inbound Process	8-28
Working With Synchronous SAP R/3 Events Using BPM (J2CA Configuration)	8-44
Configuring the Adapter Target and Channels	8-44
Generating a WSDL File for a Message	8-44
Creating a BPM Process in JDeveloper	8-44
Designing an Outbound BPM Process Using Transformations for Service Integration (BSE Configuration)	8-51
Creating an Empty Composite for BPM	8-51
Defining a BPM Outbound Process	8-51

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

Overview of Application Adapter Integration with Oracle Service Bus	9-1
Configuring an Outbound Process Using sbconsole (J2CA Configuration)	9-2
Starting Oracle Service Bus and Creating Project Folders	9-2
Setting the Class Path for Application Explorer to Integrate With Oracle Service Bus	9-6
Publishing a WSDL From Application Explorer to Oracle Service Bus	9-6
Configuring a WSDL-based Business Service	9-8
Configuring a File Type Business Service	9-10
Configuring a Pipeline With Proxy Service	9-13
Configuring an Inbound Process Using sbconsole (J2CA Configuration)	9-22
Starting Oracle Service Bus and Creating Project Folders	9-23
Setting the Class Path for Application Explorer to Integrate With Oracle Service Bus	9-23
Publishing a WSDL From Application Explorer to Oracle Service Bus	9-23
Configuring a WSDL-based Proxy Service	9-25
Configuring a File Type Business Service	9-26
Configuring a Pipeline	9-30
Configuring an Outbound Process Using sbconsole (BSE Configuration)	9-37
Starting Oracle Service Bus and Creating Project Folders	9-37

Setting the Class Path for Application Explorer to Integrate With Oracle Service Bus.....	9-37
Publishing a WSDL From Application Explorer to Oracle Service Bus.....	9-38
Configuring a File Type Business Service	9-39
Configuring a WSDL-based Business Service.....	9-39
Configuring a Pipeline With Proxy Service	9-41
Working With Synchronous SAP R/3 Events Using Oracle Service Bus (J2CA Configuration)	9-50
Starting Oracle Service Bus and Creating Project Folders	9-51
Configuring Adapter Targets and Channels.....	9-51
Generating a WSDL File for a Message	9-51
Creating a Business Service in the Oracle Service Bus Console	9-52
Creating a Proxy Service in the Oracle Service Bus Console	9-53
Configuring a Pipeline in the Proxy Service	9-57
Configuring JMS Proxy Services Using Oracle Service Bus (J2CA Configuration)	9-58
Configuring HTTP Proxy Services Using Oracle Service Bus (J2CA Configuration)	9-76

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

Configuring an OSB Outbound Process Using JDeveloper (J2CA Configuration)	10-1
Creating a Service Bus Application for OSB	10-2
Defining an OSB Outbound Process	10-3
Deploying the OSB Outbound Process	10-17
Configuring an OSB Inbound Process Using JDeveloper (J2CA Configuration)	10-20
Creating a Service Bus Application for OSB	10-20
Defining an OSB Inbound Process.....	10-20
Deploying the OSB Inbound Process	10-30
Configuring an OSB Outbound Process Using JDeveloper (BSE Configuration).....	10-30
Creating a Service Bus Application for OSB	10-31
Defining an OSB Outbound Process	10-31
Deploying the OSB Outbound Process	10-44
Working With Synchronous SAP R/3 Events of Oracle Service Bus Using JDeveloper (J2CA Configuration).....	10-44
Creating a Service Bus Application for OSB	10-45
Defining an OSB Process.....	10-45
Deploying an OSB Process.....	10-50
Configuring a JMS Inbound Process Using JDeveloper (J2CA Configuration)	10-50
Configuring a JMS Outbound Process Using JDeveloper (J2CA Configuration).....	10-56
Configuring an HTTP Outbound Process Using JDeveloper (J2CA Configuration)	10-65

11 Key Features

Configuring the Logging Feature.....	11-1
Configuring Log File Management for the J2CA Connector Application.....	11-2
Configuring Log File Management for Business Services Engine (BSE)	11-9
Configuring the Diagnosibility Feature	11-12
Supporting Protocols	11-13
Configuring the SOA Debugging Feature	11-13

Guidelines for Using the SOA Debugger	11-14
Prerequisite	11-14
Debugging a BPEL Process in Oracle JDeveloper	11-14
Debugging an OSB Process in Oracle JDeveloper	11-27
Configuring the Exception Filter	11-45
Configuring the Exception Filter	11-45
Credential Mapping for Oracle SOA Suite (BPEL, Mediator, or BPM)	11-57
Configuring Credential Mapping.....	11-58
Credential Mapping for Oracle Service Bus (OSB) Using JDeveloper	11-62
Configuring Credential Mapping.....	11-63
Filtering Unicode Characters From Text	11-70
Overview	11-71
Enabling the Filter	11-71
Specifying the Replacement Character	11-71
Saving the Settings File and Restarting the Server.....	11-72
Validating the Filter	11-72
Quick Start Guide.....	11-72
Known Limitations	11-72

12 Troubleshooting and Error Messages

Troubleshooting	12-1
Application Explorer	12-2
SAP R/3.....	12-4
Oracle Adapter J2CA	12-5
BSE Error Messages	12-5
General Error Handling in BSE.....	12-5
Adapter-Specific Error Handling.....	12-6

A Understanding the SAP System Role in Remote Processing

Functionality of the Adapter	A-1
Roles	A-2
Client.....	A-2
Server	A-2
SAP Inbound Message Processing	A-3
Configuring SAP R/3 Inbound Processing.....	A-4
Configuring a Logical System	A-6
Configuring a Distribution Model for a Logical System.....	A-7
Defining a Partner Profile	A-8
SAP Outbound Message Processing	A-9
SAP Gateway	A-11
Program IDs and Load Balancing.....	A-11
Registering a RFC Destination and Program ID in SAP GUI.....	A-13
Application Link Embedding Configuration for the Event Adapter	A-15
Defining a Port.....	A-15
Configuring a Logical System	A-15
Configuring a Distribution Model for a Logical System.....	A-17
Defining a Partner Profile	A-18

Testing the SAP R/3 ALE Configuration A-19

Glossary

Index

Preface

Welcome to *Oracle Fusion Middleware Application Adapter for SAP R/3 User's Guide for Oracle WebLogic Server*. This document provides information on how to integrate with SAP R/3 systems and develop applications.

Audience

This document is intended for system administrators and developers who integrate with SAP R/3 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 12c (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.
<i>italic</i>	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.

Introduction

Oracle WebLogic Server connects to a SAP R/3 system through Oracle Application Adapter for SAP R/3. Oracle Application Adapter for SAP R/3 provides connectivity and carries out interactions on a SAP R/3 system.

Note: Throughout this document, `<ORACLE_HOME>` refers to the 12c (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`

This chapter contains the following sections:

- [Section 1.1, "Adapter Features"](#)
- [Section 1.2, "Classical SAP Technologies for ABAP"](#)
- [Section 1.3, "Integration with SAP R/3"](#)
- [Section 1.4, "Adapter Architecture"](#)
- [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 SAP R/3 using SAP Java Connector (SAP JCo) 3.0xx is a remote function call adapter that provides a means to exchange real-time business data between SAP Enterprise Central Component (ECC) 5.0/6.0 systems and other application, database, or external business partner systems.

Note: The *xx* in SAP JCo 3.0*xx* indicates the specific version of SAP Java Connector.

SAP policy indicates that only the currently released version of SAP JCo is the supported version.

Users with support issues may be required to test with a newer version of the SAP JCo (if one is present) to verify that a problem may already be resolved.

In addition, the Oracle JRockit JVM is not supported by SAP JCo.

The **adapter** enables external applications for inbound and outbound processing with SAP R/3. Oracle Application Adapter for SAP R/3 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 is referred to as Oracle Adapter Business Services Engine (BSE).

This section contains the following topic:

- [Section 1.1.1, "Supported Versions and Platforms"](#)

Oracle Application Adapter for SAP R/3 uses XML messages to enable non-SAP R/3 applications to communicate and exchange transactions with SAP R/3 using services and events. The role of services and events is outlined. Services and events are described as follows:

- Services (also known as outbound processing): Enable applications to call an SAP R/3 business object or business operation.
- Events (also known as inbound processing): Enable applications to access SAP R/3 data only when an 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, a TCP/IP listener, or an FTP 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. In the case of this adapter, the channel is an RFC server.

Oracle Application Adapter for SAP R/3 provides:

- Support for bidirectional message interactions.
- Oracle Adapter Application Explorer (Application Explorer), a GUI tool which uses SAP R/3 object repository metadata to build XML schemas and Web services to handle adapter requests or event data.
- Support for Remote Function Calls (RFC), Business Application Programming Interfaces (BAPI), and Intermediate Documents (IDoc) interfaces to SAP R/3.
- XML schemas and WSDL files for the J2CA 1.0 and J2CA 1.5 resource adapter.
- Web services for BSE.

Data Type Limitation: Data types *h* and *g* are not supported. Type *h* represents a deep structure. Type *g* represents a variable length string. RFCTYPE_XSTRING and RFCTYPE_XMLDATA, as defined in SAPRFC.H, are not supported due to a limitation in the RFC Protocol.

See Also: *Oracle Application Server Adapter Concepts Guide*

1.1.1 Supported Versions and Platforms

The following SAP R/3 platforms are supported by Oracle Application Adapter for SAP R/3:

- SAP R/3 Enterprise 47x100
- SAP R/3 Enterprise 47x200
- mySAP ERP Central Component (ECC) 5.0, deployed on SAP NetWeaver 2004
- mySAP ERP Central Component (ECC) 6.0, deployed on SAP NetWeaver 2004s
- SAP Java Connector (SAP JCo) Version 3.0xx.

For the current release status of the SAP Java Connector, refer to SAP Note #549268 in the SAP Service Marketplace.

Note: Release versions may vary by product component. In addition, SAP functions may vary by SAP product version and support package.

1.2 Classical SAP Technologies for ABAP

Oracle Application Adapter for SAP R/3 is designed to provide standard access to SAP R/3 interfaces such as Remote Function Call (RFC) modules, BAPIs (Business Application Programming Interfaces), and IDocs (Intermediate Documents), that are used to support existing business processes.

The adapter only supports Enterprise Central Components (ECC) that are accessed by classical SAP technologies. If you require support for additional SAP functionality and components, then contact your Oracle Software Sales Representative.

These business components and methods are available to the adapter as requests of SAP R/3 and to the event adapter when SAP invokes its remote requests and work in the following ways:

- Business Application Programming Interfaces (BAPIs) are interfaces within the business framework that are used to link SAP components to one another or to third-party components. BAPIs are called synchronously and return information.
- Remote Function Call (RFC) Modules are SAP application interfaces that enable clients to invoke SAP technologies and receive responses.

Note: Depending on the release or service pack installed, certain RFCs, for example, RFC_CUSTOMER_GET, may not exist in your particular SAP R/3 system. Therefore, the examples included in this document may not be relevant to your system. If this is the case, then you should use the examples as a general reference for adapter functionality and choose an RFC that exists within your SAP R/3 application environment.

As described in SAP Release Note 109533, SAP Function Modules (RFCs) can be delivered with different release statuses. SAP supports only RFCs that are awarded with the Released for Customer status. There is no claim to the release independencies of the interfaces and the continued existence/functionality of the modules. For more information on the status of a specific function module, consult the SAP Service Marketplace.

- Intermediate Documents (IDocs) are the “logical messages” that correspond to different business processes. They enable different application systems to be linked by a message-based interface. The IDoc type indicates the SAP format to use to transfer the data for a business transaction. An IDoc is a real business process in the form of an IDoc type that can transfer several message types. An IDoc type is described by the following components:
 - Control records. A control record contains data that identifies the sender, the receiver, and the IDoc structure. An IDoc contains one control record.
 - Data records. A data record consists of a fixed administration part and a data part (segment). The number and format of the segments can be different for each IDoc type.
 - Status records. A status record describes the processing stages through which an IDoc passes. The following scenario is an example of IDoc functionality and its components:

Purchase order number 4711 was sent to a vendor as IDoc number 0815. IDoc number 0815 is formatted in IDoc type ORDERS01 and has the status records “created” and “sent.” The purchase order corresponds to the “logical” message ORDERS.

1.3 Integration with SAP R/3

You can use Oracle Application Adapter for SAP R/3 to initiate a SAP R/3 business process, such as add/update account, or you can use the adapter as part of an integration effort to connect SAP R/3 and non-SAP R/3 systems.

All functions are processed synchronously, but all content in ALE IDocs is asynchronous.

In service mode, the Oracle Application Adapter for SAP R/3 can send requests to SAP using the BAPI, RFC, or ALE interfaces.

The adapter quickly and easily integrates your SAP R/3 IDocs, RFCs, and BAPIs with mission critical SAP R/3 system applications and other enterprise applications. The benefits of the adapter include:

- Elimination of the requirement for custom coding.
- Consistent data representation.

Provides a standard XML representation of event data and request/response documents for SAP R/3.

The developer is freed from the specific details of the SAP R/3 interface (BAPI, RFC, IDoc) and the specific configuration details of the target SAP R/3 system.

- Adherence to SAP R/3 ABAP serialization rules and SAP R/3 Interface Repository standards published by SAP AG.

During event processing, the adapter receives RFCs and IDocs directly from SAP R/3. The SAP R/3 system can be configured to send an IDoc or RFC to a logical system when a certain event occurs, in this case to the adapter. The output sent by SAP R/3 can be in any of the following forms:

- An RFC request, for example, RFC_SYSTEM_INFO
- A BAPI request, for example, BAPI_COMPANYCODE_GETLIST
- An IDoc

1.4 Adapter Architecture

This section describes the architecture of the Oracle Application Adapter for SAP R/3. It contains the following topics:

- [Section 1.4.1, "Oracle Adapter Business Services Engine \(BSE\) Architecture"](#)
- [Section 1.4.2, "Oracle Adapter J2CA Architecture"](#)

Oracle Application Adapter for SAP R/3 uses Application Explorer with the following components:

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

Application Explorer (used to configure SAP connections and create Web services and events) can be configured to work in a Web services environment with BSE. When working in a J2CA environment, the connector uses the Common Client Interface (CCI) to provide integration services using adapters instead of Web services.

1.4.1 Oracle Adapter Business Services Engine (BSE) Architecture

[Figure 1–1](#) shows the generic architecture for BSE for packaged applications. The adapter works with BSE, as deployed to a Web container in a J2EE application server.

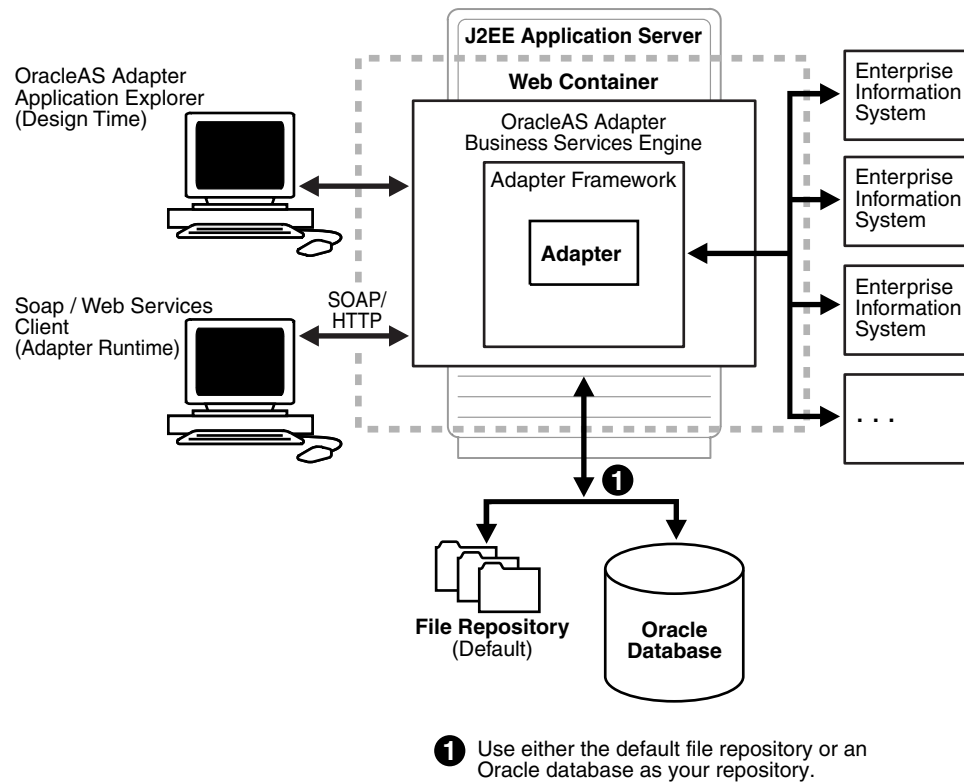
Application Explorer, a design-time tool deployed along with BSE, is used to configure adapter connections, browse EIS objects, configure services, and configure listeners to listen for EIS events. Metadata created while you perform these operations are stored in the repository by BSE.

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

BSE supports both a file-based and an Oracle database repository. The BSE repository stores the EIS connection information and the Web Service Definition Language (WSDL) for adapter services. A single instance of BSE can connect to multiple EIS applications.

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

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

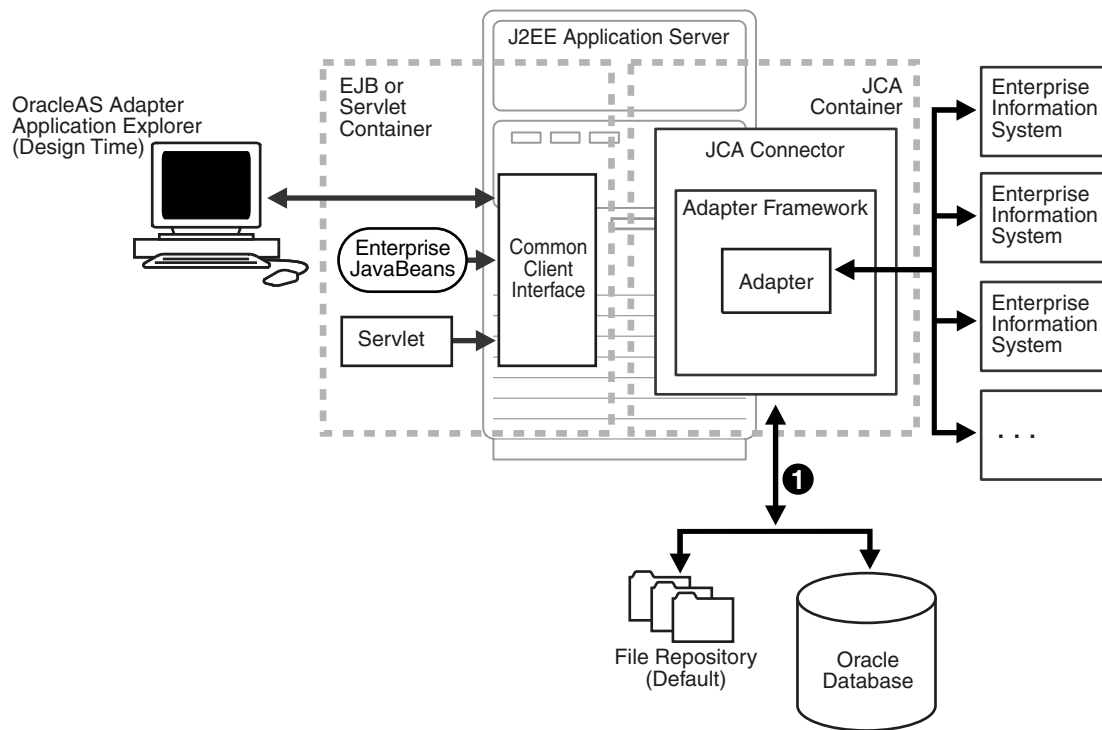


1.4.2 Oracle Adapter J2CA Architecture

Figure 1–2 shows the generic architecture for Oracle Adapter J2CA for packaged applications. This is a pure J2CA 1.0 resource adapter deployed in managed mode to the Oracle WebLogic Server. It is a universal adapter. One adapter can connect to many EIS applications.

The Oracle Adapter J2CA repository contains the list of EIS connection names and the associated connection parameters. 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 5, "Oracle WebLogic Server Deployment and Integration"](#).

Figure 1–2 Oracle Adapter J2CA Architecture



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

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 SAP R/3 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.

- Oracle Adapter J2CA does provide slightly better performance than BSE.

1.6 Sample Projects

Sample projects for the Oracle Application Adapter for SAP R/3 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\SAP_Samples.zip\SAP_Samples\BPEL\J2CA\Outbound_Project
Inbound BPEL Process (J2CA)	<ADAPTER_HOME>\etc\sample\SAP_Samples.zip\SAP_Samples\BPEL\J2CA\Inbound_Project
Synchronous BPEL Process (J2CA)	<ADAPTER_HOME>\etc\sample\SAP_Samples.zip\SAP_Samples\BPEL\J2CA\Synchronous_Events
Outbound BPEL Process (BSE)	<ADAPTER_HOME>\etc\sample\SAP_Samples.zip\SAP_Samples\BPEL\BSE\Outbound_Project
Outbound Mediator Process (J2CA)	<ADAPTER_HOME>\etc\sample\SAP_Samples.zip\SAP_Samples\Mediator\J2CA\Outbound_Project
Inbound Mediator Process (J2CA)	<ADAPTER_HOME>\etc\sample\SAP_Samples.zip\SAP_Samples\Mediator\J2CA\Inbound_Project
Outbound Mediator Process (BSE)	<ADAPTER_HOME>\etc\sample\SAP_Samples.zip\SAP_Samples\Mediator\BSE\Outbound_Project
Outbound BPM Process (J2CA)	<ADAPTER_HOME>\etc\sample\SAP_Samples.zip\SAP_Samples\BPM\J2CA\Outbound_Project
Inbound BPM Process (J2CA)	<ADAPTER_HOME>\etc\sample\SAP_Samples.zip\SAP_Samples\BPM\J2CA\Inbound_Project
Synchronous BPM Process (J2CA)	<ADAPTER_HOME>\etc\sample\SAP_Samples.zip\SAP_Samples\BPM\J2CA\Synchronous_Events
Outbound BPM Process (BSE)	<ADAPTER_HOME>\etc\sample\SAP_Samples.zip\SAP_Samples\BPM\BSE\Outbound_Project
Outbound OSB sbconsole Process (J2CA)	<ADAPTER_HOME>\etc\sample\SAP_Samples.zip\SAP_Samples\OSB\J2CA\SAP_Sample_J2CA_OSB_Outbound_Project
Inbound OSB sbconsole Process (J2CA)	<ADAPTER_HOME>\etc\sample\SAP_Samples.zip\SAP_Samples\OSB\J2CA\SAP_Sample_J2CA_OSB_Inbound_Project

Sample Project	Location
Outbound OSB sbconsole Process (BSE)	<ADAPTER_HOME>\etc\sample\SAP_Samples.zip\SAP_Samples\OSB\BSE\SAP_Sample_BSE_OSB_Outbound_Project
Synchronous OSB sbconsole Process (J2CA)	<ADAPTER_HOME>\etc\sample\SAP_Samples.zip\SAP_Samples\OSB\J2CA\SAP_Sample_J2CA_OSB_Synchronous_Events
Outbound OSB Jdeveloper Process (J2CA)	<ADAPTER_HOME>\etc\sample\SAP_Samples.zip\SAP_Samples\OSB_Jdeveloper\J2CA\SAP_Sample_J2CA_OSB_Outbound_Project
Inbound OSB Jdeveloper Process (J2CA)	<ADAPTER_HOME>\etc\sample\SAP_Samples.zip\SAP_Samples\OSB_Jdeveloper\J2CA\SAP_Sample_J2CA_OSB_Inbound_Project
Outbound OSB Jdeveloper Process (BSE)	<ADAPTER_HOME>\etc\sample\SAP_Samples.zip\SAP_Samples\OSB_Jdeveloper\BSE\SAP_Sample_BSE_OSB_Outbound_Project
Synchronous OSB Jdeveloper Process (J2CA)	<ADAPTER_HOME>\etc\sample\SAP_Samples.zip\SAP_Samples\OSB_Jdeveloper\J2CA\SAP_Sample_J2CA_OSB_Synchronous_Events

1.7 Quick Start Guide

This section enables you to quickly learn the basic steps to install and configure Oracle Application Adapter for SAP R/3 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, "Working With Synchronous SAP R/3 Events"](#)
- [Section 1.7.9, "Additional Features"](#)
- [Section 1.7.10, "Other Features"](#)

1.7.1 Installation

To install Oracle Application Adapter for SAP R/3, 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 (*sapjco3.jar* and *sapjco3.dll*) for SAP Java Connector (SAP JCo) to the following directories:

<ADAPTER_HOME>\lib

<ORACLE_HOME>\user_projects\domains\base_domain\lib

The *sapjco3.dll* file must be copied to the following directory:

C:\Windows\system32

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

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 1-1](#).

Table 1-1

Config-Property-Name	Config-Property-Value
<i>IWayHome</i>	<ADAPTER_HOME> For example: <ul style="list-style-type: none"> ■ For SOA: C:\12c_soa\soa\soa\thirdparty\ApplicationAdapters ■ For OSB: C:\12c_OSB\osb\3rdparty\ApplicationAdapters
<i>IWayConfig</i>	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 1-2](#).

Table 1–2

Param-Name	Param-Value
<i>ibseroot</i>	<p><ADAPTER_HOME>\ibse.war</p> <p>For example:</p> <ul style="list-style-type: none"> ■ For SOA: <p>C:\12C_ soa\soa\soa\thirdparty\ApplicationAdapters\ibse.war</p> ■ For OSB: <p>C:\12c_ OSB\osb\3rdparty\ApplicationAdapters\ibse.war</p>
<i>IWay.home</i>	<p><ADAPTER_HOME></p> <p>For example:</p> <ul style="list-style-type: none"> ■ For SOA: <p>C:\12C_soa\soa\soa\thirdparty\ApplicationAdapters</p> ■ For OSB: <p>C:\12c_OSB\osb\3rdparty\ApplicationAdapters</p>
<i>Iway.config</i>	<p>The name of the configuration. For example:</p> <p>IBSE</p>

Note: These steps are provided only when configuring a File repository. For more information about configuring a database repository and general configuration information, see the *Oracle Fusion Middleware Application Adapters Installation Guide for Oracle WebLogic Server* (Section 2.7.4, "Configuring the Oracle Database Repository").

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 the *Oracle Fusion Middleware Application Adapters Installation Guide for Oracle WebLogic Server* (Section 2.4, "Configuring and Deploying J2CA" and Section 2.5, "Configuring and Deploying Business Services Engine").

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 4.2, "Starting Application Explorer"](#)

- [Creating a BSE Configuration: Section 4.4.1, "Creating a Configuration for BSE"](#)
- [Creating a J2CA Configuration: Section 4.4.2, "Creating a Configuration for J2CA"](#)
- [Connecting the Created Configurations: Section 4.4.3, "Connecting to a BSE or J2CA Configuration"](#)
- [Creating and Connecting to Targets: Section 4.5, "Establishing a Connection \(Target\) for SAP R/3"](#)
- [Creating and Testing Web Services: Section 4.9, "Creating and Testing a Web Service \(BSE Configurations Only\)"](#)
- [Generating WSDL Files: Section 4.8, "Generating WSDL \(J2CA Configurations Only\)"](#)
- [Creating and Working With Channels: Section 4.10, "Configuring an Event Adapter"](#)

1.7.6 Working With Service Components in the SOA Suite

Oracle Application Adapter for SAP R/3 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 6, "Integration With BPEL Service Components in the Oracle SOA Suite"](#)
- [Chapter 7, "Integration With Mediator Service Components in the Oracle SOA Suite"](#)
- [Chapter 8, "Integration With BPM Service Components in the Oracle SOA Suite"](#)

1.7.7 Working With Oracle Service Bus

Oracle Application Adapter for SAP R/3 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, see [Chapter 9, "Configuring an Outbound and Inbound Process for Oracle Service Bus Using sbconsole"](#).

For more information on working with OSB JDeveloper, see [Chapter 10, "Configuring an Outbound and Inbound Process for Oracle Service Bus Using JDeveloper"](#).

1.7.8 Working With Synchronous SAP R/3 Events

Synchronous SAP R/3 events are supported with BPEL, BPM, and OSB. For more information on working with Synchronous events, see the following sections in this user's guide:

- [BPEL: Section 6.6, "Working With Synchronous SAP R/3 Events Using BPEL \(J2CA Configuration\)"](#)
- [BPM: Section 8.6, "Working With Synchronous SAP R/3 Events Using BPM \(J2CA Configuration\)"](#)
- [OSB: Section 9.5, "Working With Synchronous SAP R/3 Events Using Oracle Service Bus \(J2CA Configuration\)"](#)

1.7.9 Additional Features

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

- [Configuring Connection Pooling: Section 3.4, "Connection Pooling - Services"](#)
- [Inbound Schema Validation: See step 7C in Section 6.5.1.2, "Generating WSDL for Event notification"](#).

1.7.10 Other Features

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

- [JMS Proxy Services: Section 9.6, "Configuring JMS Proxy Services Using Oracle Service Bus \(J2CA Configuration\)"](#)
- [HTTP Proxy Services: Section 9.7, "Configuring HTTP Proxy Services Using Oracle Service Bus \(J2CA Configuration\)"](#)
- [Working With the Exception Filter: Section 11.4, "Configuring the Exception Filter"](#)
- [Configuring Credential Mapping:](#)
 - [Section 11.5, "Credential Mapping for Oracle SOA Suite \(BPEL, Mediator, or BPM\)"](#)
 - [Section 11.6, "Credential Mapping for Oracle Service Bus \(OSB\) Using JDeveloper"](#)

Getting Started

This section provides a quick start guide to use the Oracle Application Adapter for SAP R/3. This chapter contains the following sections:

- [Section 2.1, "Verifying the SAP Java Connector \(SAP JCo\)"](#)
- [Section 2.2, "Identifying SAP R/3 Logon Parameters"](#)

2.1 Verifying the SAP Java Connector (SAP JCo)

Once you have installed the SAP Java Connector (SAP JCo), as a best practice, you can verify the connector to make sure it is installed correctly and that all the required SAP JCo library files are available. This section contains the following topics:

- [Section 2.1.1, "Verifying SAP JCo on Windows Platforms"](#)
- [Section 2.1.2, "Verifying SAP JCo on UNIX Platforms"](#)

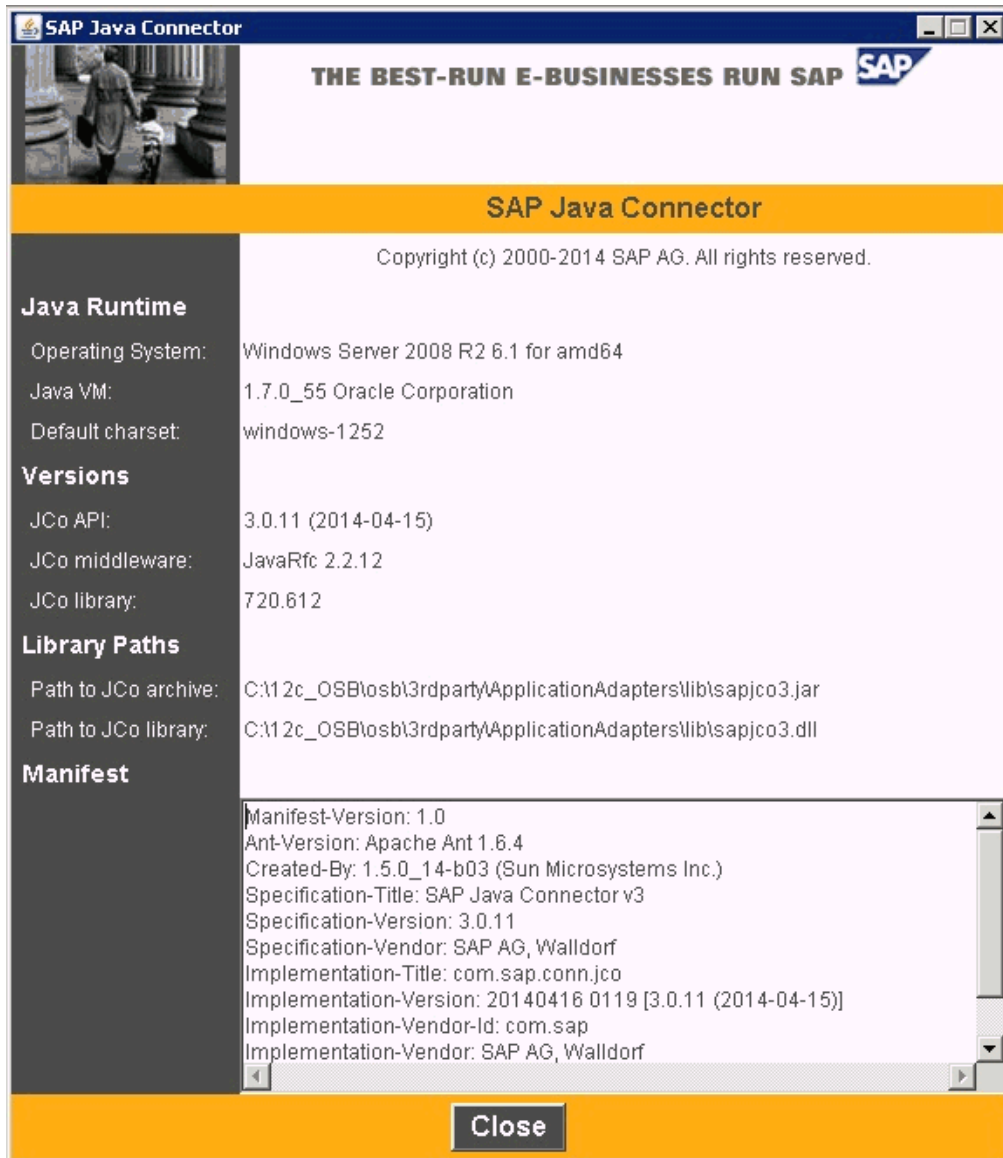
2.1.1 Verifying SAP JCo on Windows Platforms

Perform the following steps to verify SAP JCo on Windows:

1. Navigate to the directory where the sapjco3.jar file is located.
2. Right-click the sapjco3.jar file, select Open With from the context menu, and click Java 2 Platform Standard Edition binary.

The SAP Java Connector (JCo) dialog is displayed, as shown in [Figure 2-1](#).

Figure 2–1 SAP Java Connector (JCo) dialog



All the required information that pertains to the SAP Java Connector on your Windows platform is provided.

3. Once you have reviewed the SAP Java Connector files, click **Close**.

2.1.2 Verifying SAP JCo on UNIX Platforms

Perform the following steps to verify SAP JCo on UNIX:

1. Navigate to a UNIX command prompt.
2. Execute the following command:

```
$ java -jar sapjco3.jar -stdout
```

All the required information that pertains to the SAP Java Connector on your UNIX platform is provided, as shown in the following example.

```
-----  
| SAP Java Connector (JCo) |
```



```

| Copyright (c) 2000-2014 SAP AG. All rights reserved. |
| Version Information |
-----
Java Runtime:
Operating System: Linux 2.6.32-100.28.5.el6.x86_64 for amd64
Java VM: 1.7.0_55 Oracle Corporation
Default Charset: UTF-8
Versions:
JCo API: 3.0.11 (2014-04-05)
JCo middleware: JavaRfc 2.2.12
JCo library: 720.612
Library Paths:
Path to JCo archive: /rdbms/oraoem/Oracle/Middleware/Oracle_
Home/soa/soa/thirdparty/ApplicationAdapters/lib/sapjco3.jar
Path to JCo library: /rdbms/oraoem/Oracle/Middleware/Oracle_
Home/soa/soa/thirdparty/ApplicationAdapters/lib/libsapjco3.so
-----
| Manifest |
-----
Manifest-Version: 1.0
Ant-Version: Apache Ant 1.6.4
Created-By: 5.1.028 (SAP AG)
Specification-Title: SAP Java Connector v3
Specification-Version: 3.0.11
Specification-Vendor: SAP AG, Walldorf
Implementation-Title: com.sap.conn.jco
Implementation-Version: 20140415 1946 [3.0.11 (2014-04-15)]
Implementation-Vendor-Id: com.sap
Implementation-Vendor: SAP AG, Walldorf
Main-Class: com.sap.conn.jco.rt.About
-----
$

```

3. Review the information for the SAP Java Connector on your UNIX platform.

2.2 Identifying SAP R/3 Logon Parameters

This section identifies the SAP R/3 logon parameters, which are used to configure a connection to SAP R/3 using the Oracle Application Adapter for SAP R/3. This information can be used as a reference. This section contains the following topics:

- [Section 2.2.1, "User Parameters"](#)
- [Section 2.2.2, "System Settings \(Application Server\) Parameters"](#)
- [Section 2.2.3, "System Settings \(Message Server\) Parameters"](#)
- [Section 2.2.4, "Connection Pool Parameters"](#)
- [Section 2.2.5, "SAP Gateway Parameters"](#)
- [Section 2.2.6, "ALE Parameters"](#)
- [Section 2.2.7, "Global Processing Parameters"](#)
- [Section 2.2.8, "SNC Parameters"](#)

2.2.1 User Parameters

[Table 2–1](#) lists and describes User parameters.

Table 2–1 User Parameters

Parameter	Description	Example	Comment
Client	Identifies the SAP client.	800	In commercial, organizational, and technical terms, a self-contained unit in an SAP system with separate master records and its own set of tables. A client can, for example, be a corporate group.
User	SAP login ID.	“abc123”	User type for dialog-free communication between systems.
Password	Confidential authentication information.	“xyz999”	A protected word or string of characters that identifies or authenticates a user for access to an SAP system.
Language	A language key. EN (English) is the default.	EN	
Codepage	A character code page value.	Cp1252	
Authentication Mode	How the connection is validated.	Selection, see the next column.	Password - use the value in the supplied field. The password parameter that is mentioned here refers to the Password parameter field in Application Explorer.

2.2.2 System Settings (Application Server) Parameters

Table 2–2 lists and describes System Settings (Application Server) parameters.

Table 2–2 System Settings (Application Server) Parameters

Parameter	Description	Example	Comment
Application Server	Connects to an ABAP application server.	iwjpsap	Application programs in an R/3 System are run on application servers. To obtain metadata information, a connection to a single application server is required.
System Number	Identifies a unique instance on the application server.	00	An application server may have different system numbers. Use the one provided by your administrator.

2.2.3 System Settings (Message Server) Parameters

Table 2–3 lists and describes System Settings (Message Server) parameters.

Table 2–3 System Settings (Message Server) Parameters

Parameter	Description	Example	Comment
Message Server	Connects to an ABAP message server.	iwjpsap	For load balancing purposes, application servers from one SAP system are usually configured in logon groups, where each group serves a particular kind of user. The message server is responsible for communication between the application servers. It passes requests from one application server to another within the system. It also contains information about application server groups and the current load balancing within them. It uses this information to choose an appropriate server when a user logs onto the system.
R/3 Name	Identifies a unique instance on the application server.	P47	Symbolic SAP system name used to identify the system.
Server Group	Identifies the logon group		Logon group that the user ID belongs with.

2.2.4 Connection Pool Parameters

Table 2–4 lists and describes Connection Pool parameters.

Table 2–4 Connection Pool Parameters

Parameter	Description	Example	Comment
Connection Pool Size	Maximum number of connections for the pool.	2	Sets the maximum number of connections that can be allocated from the pool.
Connection Timeout	Maximum time to keep open a free connection (in minutes).	10	Connections that have not been used for at least the connection timeout interval are closed.
Connection Wait Time	Maximum wait for a free connection.	30 (seconds)	Sets the maximum time to wait in a connection request for a free connection. If the pool is exhausted, and there is still no connection available after the specified time, then a JCO exception with the key JCO_ERROR_RESOURCE is generated. The default value is 30 seconds.

2.2.5 SAP Gateway Parameters

Table 2–5 lists and describes SAP Gateway parameters.

Table 2–5 SAP Gateway Parameters

Parameter	Description	Example	Comment
SAP Gateway Host	Enter the name of a SAP Gateway server.	"isdsrv2"	The SAP Gateway carries out CPI-C services within the SAP world, which are based on TCP/IP. These services enable SAP Systems and external programs to communicate with one another.
SAP Gateway Service	Enter the service name (usually a compound of the service name and system number).	sapgw00	Service name on the gateway host.
Program ID	A program identifier that has been specified on the SAP Gateway server (case sensitive).	"S1PROG"	Unique identifier for your communication session specified by your system administrator. The value entered in this field must match the one exposed on the gateway.

2.2.6 ALE Parameters

Table 2–6 lists and describes ALE parameters.

Table 2–6 ALE Parameters

Parameter	Description	Example	Comment
EDI Version	Specifies the ALE version of the target system.	3	Version "3" (Release 4.0 onwards) should be selected in the port description for all R/3 partner systems with Release 4.0 or higher. Version "2" (release 3.0/3.1) must be set in the port description for all R/3 partner systems with releases lower than 4.0.
IDOC Release	Specifies the version in which the IDOC definition was released.	Blank or a specific SAP release version, for example, "46C".	You can assign segment definitions from previous releases to an IDoc type in the current release. This may be necessary if, for example, the partner is using an older release which supports your current IDoc type but not your current segment definitions.
IDOC Release Provider	Specifies where the adapter receives the release information.	Selection, see the next column.	IDOCDOREL uses the information in the IDOC header. SAP release gets the information from the user account logon. USERINPUT uses the IDOC release field above to get the information.

2.2.7 Global Processing Parameters

Table 2–7 lists and describes Global Processing parameters.

Table 2-7 Global Processing Parameters

Parameter	Description	Example	Comment
Error Handling	Specifies the error handling method of the adapter.	Selection, see the next column.	<i>Creates error document</i> writes an exception document with the full error text to the output destination. <i>Throws exception</i> creates a java exception, this may or may not display the full error text depending on the underlying component error.
Commit with Wait	Specifies the commit behavior.	Selection, see the next column.	Off - default Sends Commit Request to Application Server in the document. If there is a commit error, then it is not reflected back (Optimal performance). On (checked) - waits for a full database server commit in the document before returning. Commit errors are reflected back to the adapter level (slowest performance). For the recommended setting, see your SAP administrator.
SAP trace	Enables the SAP Java connectors trace behavior.	Selection, see the next column.	Off default - only hard errors are recorded in a trace file (dev rfc.trc) in append mode. ON - individual rfc*.trc and JCO*.trc are written for each request. Useful in finding errors, not recommended in a productive system.
Trace level	Indicates the level of detail in the SAP traces.	5	Select a value that ranges from 0 through 10 from the list.

2.2.8 SNC Parameters

Table 2-8 lists and describes SNC parameters.

Note: The SAP SNC parameters are documented in this documentation as this functionality is available from the SAP Java Connector (SAP JCo). It is strongly recommended that if SNC is to be used with the adapter, a SAP Router be configured with SNC enabled rather than direct SNC from the adapter. This is the strongest, most secured method of connection. Because SNC depends on the validity of certificates issued by the local SAP host server, it is the responsibility of the local SAP support staff to ensure the validity of SNC certificates and correct configuration of the SAP Router and SAP SNC.

Table 2-8 SNC Parameters

Parameter	Description	Example	Comment
SNC mode	Flag for activating SNC.	1 (on)	Required.

Table 2–8 (Cont.) SNC Parameters

Parameter	Description	Example	Comment
SNC Partner	Specifies the application server's SNC name	p:CN=ABC, O=MyCompany C=US	You can find the application server's SNC name in the profile parameter snc/identity/as.
SNC level	Specifies the level of protection to use for the connection.	Selection, see the next column.	1: Authentication only 2: Integrity protection 3: Privacy protection (default) 8: Use the value from snc/data protection/use on the application server 9: Use the value from snc/data_protection/max on the application server Default value = 3
SNC Name	Specifies SNC name.	p:CN=SAPJ2EE O=MyCompany, C=US	Although this parameter is optional, we do recommend setting it to ensure that the correct SNC name is used for the connection.
SNC library path	Specifies the path and file name of the external library.	C:\SAP J2EE_ Engine\SAPCry ptolib\sap crypto.dll	The default is the system-defined library as defined in the environment variable SNC LIB.

SAP Java Connector 3.0xx Considerations

This chapter describes the new usage considerations of the Oracle Application Adapter for SAP R/3 that are associated with SAP Java Connector (SAP JCo) 3.0xx. It contains the following sections:

- [Section 3.1, "Supported Platforms"](#)
- [Section 3.2, "Connection Management"](#)
- [Section 3.3, "Connection Pooling - Events"](#)
- [Section 3.4, "Connection Pooling - Services"](#)
- [Section 3.5, "Single Server Support"](#)
- [Section 3.6, "Server Threads"](#)
- [Section 3.7, "Changing Parameters and Table Parameters"](#)
- [Section 3.8, "Trace Level Parameter"](#)

Note: The *xx* in SAP JCo 3.0xx indicates the specific version of SAP Java Connector.

SAP policy indicates that only the currently released version of SAP JCo is the supported version.

Users with support issues may be required to test with a newer version of the SAP JCo (if one is present) to verify that a problem may already be resolved.

In addition, the Oracle JRockit JVM is not supported by SAP JCo.

3.1 Supported Platforms

SAP JCo 3.0xx removes 32-bit support on Unix platforms. Only Linux (with Intel processors) and Windows (with Intel processors) now support the 32-bit SAP JCo. All other platforms require true 64-bit processing throughout. Some JVM settings cause the JVM to read as "mixed mode", which is not compatible with SAP JCo. Some platforms require a specific flag of `-d64` to enable complete 64-bit processing in the JVM.

For more information on supported platforms with SAP JCo 3.0xx, see *Oracle Fusion Middleware Application Adapters Installation Guide for Oracle WebLogic Server*.

For supported JVM information that corresponds to each operating system, refer to SAP Note #1077727 in the SAP Service Marketplace. If a specific JVM is not included on the list of supported JVMs, then it is not supported by SAP R/3.

Note: The Oracle JRockit JVM is not supported by SAP JCo 3.0.14.

3.2 Connection Management

With SAP Java Connector (SAP JCo) 3.0xx, the adapter is the facilitator to a JCo instance that manages all connections to the SAP system from that instance. Each channel is configured with SAP-specific parameters, and these become SAP JCo destinations for SAP system connections.

3.3 Connection Pooling - Events

Each of the connection pool parameters may be manually tuned per target when the "Accept pool size" setting is set to **TRUE** in the channel configuration. The default mode is to use the automatic mode as before, so no targets need readjusting unless you are tuning the particular channel.

Each server thread initially has a metadata connection, and the metadata repository automatically uses another connection when used in automatic mode. This pool usually shrinks down to one or two connections very quickly as required objects are added to the metadata repository.

Overview

In "Listener" or "Server" mode, the iWay adapter receives requests from SAP and processes the requests by either delivering to a further endpoint, or executing an endpoint process. Processing a request requires metadata about the function or IDoc received, the parameters, the tables, and fields. Each table or field may have type, length, offset, or other information.

This metadata information is queried from the SAP server and then cached in the local Repository store for the duration of the Listener session. This mechanism of retrieving and caching usually shows a large initial size of connections as the greatest amount of metadata is required first, and then a small size later as the pool of connections is reduced to a size appropriate to the needs of the Repository cache.

SAP JCo controls the connection management. The Listener channel collects and passes the connection parameters to the JCo, and the JCo creates the connections and passes a connection back to iWay to perform work. The JCo performs the connection pooling in a different way than most implementations of connection pooling. A JCo connection pool will always start with a size of one connection. If that one connection is busy when a second request arrives, a second request is opened. After the timeout period has expired, any unused free connections are closed. If the connection pool attempts to go over the size limit, any unused connections are closed immediately. The lower bound for the connection pool size is one, for the keep-alive function of the server to operate.

Each "Listener" or "Server" is composed of three server connections by default:

- The number of connections a parameter is exposed to in the Listener channel configuration.
- Each connection that makes requests for metadata.
- The JCo Repository that automatically makes certain metadata request itself.

The ratio of JCo Server threads to SAP emitter threads in 3:1 size is used for optimal Repository performance.

It is possible, but not recommended, to reduce the number of server connections. This will severely impact the performance of the channel, but the number of metadata connections can be reduced in this indirect way for low throughput servers. If the number of connections is lowered, then the slower the server throughput or number of requests can be handled in a specified time period.

The number of SAP-emitting threads that send out Functions or IDocs can be increased in the SAP QOUT scheduler. For more information, see the SAP help topic "Configuring System Resources for QOUT Scheduler ", and then adjust the listener threads according to the 3:1 ratio. This can allow for parallel processing when using multiple iWay instances. For an advanced topic, see the *Oracle SAP Best Practice Guide*.

Table 3–1 Explanation of Connection Tuning Parameters (Impacts Metadata Only in Server Mode)

Tuning Parameter	Description
Accept pool parameters (True False)	If False , use the default adapter connection tuning. If True , use the values provided by the user.
Connection pool size	Maximum number of connections in a pool. If this is less than the "Maximum Connections", Pool Size will set by the JCo to the "Maximum Connections" size.
Maximum connections	Maximum number of connections created at one time.
Connection timeout (min.)	Time in minutes until the unused connections are released.
Connection wait time (sec.)	Time in seconds to wait for a free connection handle when all connections are busy before throwing a Resource exception.

Cautions

Allocating a connection pool size too small will cause the throughput to suffer, as each server connection will have to wait for a metadata connection until the previous request is released.

A connection timeout that is too small a size requires more SAP, Oracle, and iWay resource allocations to create new connections if the connections have been reduced out of the pool too soon.

Allocating too many connections is to drain SAP, Oracle, and the iWay system of valuable resources that could be used elsewhere. The rule of "fetch and cache" indicates that the connections will be downsized to the lowest available number after the metadata is cached in the repository for the Listener.

It is recommended that all users of the adapter/listener be SAP "Communication 'C'" -type users and not "Dialog 'A'" users because of different resource allocations on the SAP server and for security purposes.

3.4 Connection Pooling - Services

This feature implements shared connection pooling for the mySAP adapter. The term "target" is used in the mySAP30 adapter to identify a saved set of SAP connection parameters used to locate a particular SAP instance (either an application server or a Message Server) and specific logon and password credentials. (A target for an application server will include application server id and system number, a target for a message server will include Message Server, Logon group and other parameters. The complete list of target parameters can be found in the mySAP 30 users guide.)

A target is created in the application explorer when the user enters and saves the SAP user and server parameters. At run time, the target is used to identify and connect to the relevant SAP system via a connection pool.

In a "connection pool", connections use the same user id and password, but a pool of connection handles is kept active so that the overhead of SAP logon and logoff is not incurred for every transaction. The connection handles are reused over and over and dynamically up and down sized with system load. Establishing a connection with an SAP system is a resource intensive and time consuming task, for both the client and the server, reusing an existing connection takes much less time and lessens the resource burden on client and server.

Using the shared JCo feature, a target can be shared among multiple adapter instances in the same Java Virtual Machine. This feature is highly recommended to reduce the consumption of resources on the application server and accelerate application response times.

Description

An SAP connection pool is issued as a request to the SAP Java Connector to shape the connections opened according to the parameters provided. The adapter does not allocate or reserve connections. All of the connection handling is done by the SAP Java Connector.

It is possible to configure multiple targets or a single target for a single SAP system. If multiple SAP systems are installed, create a single or multiple target for each systems, it is possible to be connected to several SAP systems at the same time, each SAP system target will contain the base connection pool for the target system.

Limitations of the Share JCo Pools Feature

The shared JCo feature is an adapter wide feature. Once the feature is enabled, all connections using the same target name will share the same connection pool.

The adapter is managed by the application server within the Managed Connection Interface, and communicates with SAP via SAP JCo connection pools, only the SAP JCo pools are discussed here.

The same user and password are used in connection pools, so it may be desirable to separate connection pools by functional area and user id to obtain the correct SAP application authorizations.

Example, though 'accounting ' and 'finance' share the same SAP system, the user id and password are different in each target, each target allocated according to SAP functional area.

If advanced adapter target options are used ("IDoc version" or "IDoc Release Provider", etc.) and dissimilar settings are required for different application types, two targets must be created as these parameters are not dynamic.

Connection Usage Without the Shared JCo Feature

Without Shared JCo, each adapter instance allocated by the application server (either thread or process) attempts to open a connection pool and repository connection. When the application server resource that requests a connection is removed or finishes, the connection pool and all resources with it are marked for removal by the timeout checker. Not using Shared JCo with multiple application server threads can open many adapter instances and consume application server resources.

Without shared SAP JCo, if a connection pool is marked for maximum 60 connections, and 10 application server threads are instantiated for adapter calls, it is possible for each adapter instance to attempt to obtain 60 connections.

Enabling the Shared JCo Feature

Identify the target or targets to be created or used among processes. Create or modify the targets so sufficient connection handles are available for all resources consuming the pool.

By default, the Shared JCo feature is enabled in the *mysap.settings.xml* file, which is located in the following directory:

```
<ADAPTER_HOME>\lib\settings\mysap.settings.xml
```

If different targets exist for the same SAP system and user logon credentials, consider consolidating the target names for simplicity and maximum performance.

Verification (Optional)

A quick verification process can be done by configuring two outbound processes that share an adapter target, and in that target enable the "Sap trace" parameter. The trace should show the exact same connection pool servicing all requests to the same SAP destination. It is recommended to never run SAP traces in productive systems so perform the verification on a development or test system.

Connection Parameter Tuning (Optional)

A connection pool is determined by the parameters in the adapter target. Connection pools are dynamic entities, the pool grows as requests are added, up to the maximum size specified, and shrinks if no new requests arrive.

Connection pool size - maximum size the pool can grow to (connections start at 1 and will grow up to this limit. If the connections are no longer needed, the pool will be downsized back to the initial size).

Connection Timeout (in minutes): The maximum time a free, unused connection can remain in the pool before being removed.

Connection Wait Time (in seconds): The maximum time a request will wait for a connection when no connections are available. If a connection handle is not returned by the time this wait time is expired, a "Resource Exception" is triggered and processing on the request stops.

Depending on system topology, a new connection may take up to 300 milliseconds or more to connect to SAP, obtain the required metadata, and execute the request, where a connection taken from the pool will take 10 or less ms. This can impact considerably the total application performance.

Connection tuning is done by changing, one at a time, the parameters, and seeing the impact on the application performance. Some consequences of improper connection tuning include:

- Not allocating sufficient connections can lead to waiting requests on the application server.
- Too short a Connection Timeout will result in longer times for processing, as new connections are opened, too long a period means unused connections will sit in the pool.
- Too short a time wait period may result in excessive resource exceptions.

Consider enabling the SAP JCo trace and viewing the connection request results for finer grade tuning.

Disabling the Shared JCo Feature

Perform the following steps to disable the Shared JCo feature:

1. Stop the application server if it is currently running.
2. Edit the `mysap.settings.xml` file, which is located in the following directory:

```
<ADAPTER_HOME>\lib\settings\mysap.settings.xml
```

3. Set the `ShareJCO` option to `false`.
4. Save the `mysap.settings.xml` file.

3.5 Single Server Support

With SAP Java Connector (SAP JCo) 3.0xx, a single instance of a SAP JCo server is permitted for each SAP JCo installation. The Oracle Application Adapter for SAP R/3 maps between ports, channels, and server threads. The changes are transparent to the end-user, however in load balanced and multi-threaded installations, special care must be taken to match the server thread count of the adapter and the active server thread count of the SAP Gateway. This topic can be investigated further in the SAP R/3 help file "Configuring the SAP Gateway", where comprehensive information is exposed on load balancing and server threads.

3.6 Server Threads

This topic describes the behavior of server threads within the context of load balancing. It contains the following sections:

- [Section 3.6.1, "Uniqueness"](#)
- [Section 3.6.2, "Load Balancing Adapter Configuration"](#)
- [Section 3.6.3, "Load Balancing SAP Configuration"](#)

3.6.1 Uniqueness

There can be only one server with a combination of SAP Gateway Server + SAP Gateway Service + Program ID active at one time. All messages sent through that channel will arrive at the same adapter destination, unless routing by message schema is performed.

3.6.2 Load Balancing Adapter Configuration

By default, the SAP R/3 outbound Gateway creates one active connection for each RFC destination for non SAP R/3 systems. For maximum throughput, the `Thread count` parameter is set to three (three server threads for each RFC destination). This parameter is located in the Advanced tab of Application Explorer when configuring channels. Usually, this leaves one thread available to listen for requests, one thread to process a current request, and one thread to perform clean up tasks from processing.

3.6.3 Load Balancing SAP Configuration

In cases where a large amount of traffic on a particular RFC destination is required, increase the thread output size in SAP and increase the thread count and try and keep

the ratio of Gateway threads to server threads at the same 1:3 ratio. For example, if the Gateway has 3 active threads for destination "X", on the outbound server channel side, the server thread count is now 9.

In addition to Gateway load balancing, the amount of SAP Gateway threads can be increased. For more information, consult your SAP R/3 system administrator.

When more than one Program ID with the same name is registered for one SAP Gateway (using a separate JVM), the Gateway can be enabled for a load balancing scenario, depending on the parameters in the Gateway profile on the SAP R/3 system. It is recommended to consult an SAP R/3 system administrator for more information. However, the profile is either set to 0 (no load balancing), 1 (lowest counter), or 2 (lowest load).

The SAP R/3 system administrator can provide the required information on how to configure your destinations to take advantage of each situation. If load balancing is not enabled and you register multiple server channels with the same Program ID without performing Oracle schema validation, then it is possible to receive IDocs in the wrong channel because there is no routing on the channels or destinations.

3.7 Changing Parameters and Table Parameters

SAP Changing parameters can take almost any SAP R/3 type, including nested structures (tables where a column is itself a structure) and data of no fixed length, such as String data types. Changing parameters can be used as input to a function module, exported from a function module, or both. This depends on the requirements of the designer for the specific SAP function. Table parameters are limited to Flat data structures (relational) with fixed columns of defined lengths and a required header line. Changing parameters can take almost any SAP R/3 type, including nested structures (tables where a column is itself a structure) and data of no fixed length, such as String data types. Changing parameters can be used as input to a function module, exported from a function module, or both. This depends on the requirements of the designer for the specific SAP function.

The XML schemas for functions that use Changing parameters do not list them as a separate category. However, just like Table parameters, they appear in the request and response XML schemas. Depending on the function design, if the parameters are marked as *optional*, you do not have to supply a value. The optional indicator displays in Application Explorer and in the XML schema. For more information on using Changing parameters, see your SAP R/3 documentation.

3.8 Trace Level Parameter

Oracle Application Adapter for SAP R/3 (using SAP JCo 3.0xx) includes a parameter called *Trace level*, which can be found in the Advanced tab of Application Explorer, as shown in [Figure 3-1](#).

Figure 3–1 Advanced Tab

This parameter indicates the level of detail in the SAP traces. Select a value that ranges from 0 through 10 from the list. The common values that are used include:

- 0 - nothing
- 1 - errors and warnings
- 2 - execution path, errors, and warnings
- 3 - full execution path, errors, and warnings
- 4 - execution path, info messages, errors, and warnings
- 6 - full execution path, info messages, errors, and warnings
- 7 - debug messages, full execution path, info messages, errors, and warnings
- 8 - verbose debug messages, full execution path, info messages, errors, and warnings

Path values null, stdout, and stderr are allowed. An existing path is also allowed. If the path is equal to null, stdout, or stderr, then SAP JCo generates a trace to the standard output (error) stream. If at least one JCoTraceListener is registered, then the trace output is sent to the listener only. To redirect the SAP JCo trace into a file, the path value has to be set to an existing directory.

The following environment variables have to be set at ORACLE_JAVA INVOKE_HOME:

```
Java -Djco.trace_path=drive:/directory
```

The following is an example of a path on a Solaris platform:

```
/path1/dir1/
```

The following variables can be set at the operating system level:

```
RFC_TRACE=1
CPIC_TRACE=(1,2,3)
RFC_TRACE_DIR=
CPIC_TRACE_DIR=
```

Where 1 is an error trace, 2 is a complete process trace, and 3 is a complete process trace, short data trace.

It is recommended to check your environment settings, especially under a hosted environment.

Configuring Oracle Application Adapter for SAP R/3

This chapter describes how to use Oracle Adapter Application Explorer (Application Explorer) to define a target to connect to a SAP R/3 system, view system objects, and create XML schemas and Web services. This chapter also explains how to configure an event adapter.

This chapter contains the following sections:

- [Section 4.1, "Understanding SAP R/3 Nested Structures and Nested Tables"](#)
- [Section 4.2, "Starting Application Explorer"](#)
- [Section 4.3, "Configuring Repository Settings"](#)
- [Section 4.4, "Creating a Repository Configuration"](#)
- [Section 4.5, "Establishing a Connection \(Target\) for SAP R/3"](#)
- [Section 4.6, "Viewing Application System Objects"](#)
- [Section 4.7, "Creating XML Schemas"](#)
- [Section 4.8, "Generating WSDL \(J2CA Configurations Only\)"](#)
- [Section 4.9, "Creating and Testing a Web Service \(BSE Configurations Only\)"](#)
- [Section 4.10, "Configuring an Event Adapter"](#)
- [Section 4.11, "Configuring the SAProuter Service"](#)

4.1 Understanding SAP R/3 Nested Structures and Nested Tables

SAP R/3 Remote Function Modules can contain several types of parameter lists. The term *parameter* is used to name a variable, and to refer to one of the pieces of data provided as input to the function. All SAP R/3 Remote Function Modules are *strongly typed*, which means that the types of data that can be passed to the parameters is defined beforehand in the definition of the function. A parameter list is grouped by the *type* of parameter or *direction*. A function has the following types of parameter lists:

- **Import.** Values passed to the function.
- **Export.** Values passed from the function.
- **Changing.** Values that can be passed to the function and then modified by the function.
- **Tables.** Relational type data that can be passed to or from the function.

Exceptions: Some functions have strongly typed exceptions. These are not part of the schema, but will be formally generated by the function in the event of an error (in the form of an *ABAP Exception*), with an error key and value.

The Oracle Application Adapter for SAP R/3 provides the following modes of operation:

- **Synchronous processing.** Sending RFCs and BAPIs to SAP R/3.
- **Asynchronous processing.** Synchronous sending (through Transactional RFC) to SAP R/3: IDoc-enabled BAPIs and IDocs.
- **Synchronous event receiving from SAP with reply to SAP.** RFCs and BAPIs in the form of RFCs.
- **Asynchronous processing.** Synchronous receiving (through Transactional RFC) from SAP R/3: IDoc-enabled BAPIs and IDocs.
- **Synchronous receiving only.** RFCs and BAPIs in the form of RFCs.

A Remote Function Call (RFC) can be defined with parameters defined as *optional* or *mandatory*. If a parameter is optional and no value is passed, then the parameter is marked as inactive. If a parameter is mandatory, then the function cannot be called until a value has been assigned to the parameter.

All data is passed to and from Remote Function Modules *by value*, which means that the actual values are sent along with the call.

All parameters must be defined with the data types that are defined in the SAP R/3 Data Dictionary. The caller of the function can call this dictionary to find the types of the parameters, and then pass the appropriate values.

These are the ABAP data types that are defined as *elementary* and *complex*, and their usage helps define the parameter type as well as the data type. The available elementary types are:

Numeric:

- Integer - I
- Packed -P
- Floating - F

Character:

- Character - C
- Date - D
- Time - T
- Numeric - N

Byte:

- Hex - X

Complex data types are built from the elementary types and are defined in the SAP R/3 Data Dictionary as:

- **Structures.** A grouping of elementary types (columns) defined with a name, and treated as a unit.
- **Tables.** A grouping of elementary types as in a Structure, and spanning multiple lines (rows).

The following data types handle variable length records:

- **STRING.** A grouping of characters and length is characters * character width.
- **XSTRING.** A grouping of bytes and length is the number of bytes.

Using STRING or XSTRING in a function parameter causes the function to run more slowly over RFC, as the data must be converted to XML, shipped over the transport, and then converted back to the data type. Using STRING or XSTRING or a Table in a Structure makes the Structure a *Deep Structure*.

In SAP R/3, the use of IMPORT and EXPORT parameters was limited to elementary data types or Structures. With the latest releases of SAP ERP ECC, the ability to allow any data type in any of the parameter lists has been added. Now, an IMPORT parameter can contain a Table just as it can contain an elementary type.

Flat Structures contain only elementary data types with a fixed length (no internal Tables or strings).

Nested Structures

A Structure can contain a column that is itself a Structure or a Table. This makes the Structure a *Nested Structure*. If any of the elements contain a variable length element or Table, then it becomes a *Deep Nested Structure*.

The following is a sample XML structure of a Nested Structure:

```
<REQHDR>
  <MESSGID/>
  <TMSTP/>
  <CORRID/>
  <CONSID/>
  <PROVID/>
  <TRINFO/>
    <TRACK>
      <ZBUSID/>
      <ZCONVID/>
      <ZUSERID/>
    </TRACK>
</REQHDR>
```

The Structure REQHDR has a column called TRACK that is itself a Structure.

The following is a sample XML structure of a Deep Nested Structure:

```
<SYSTEMINFO>
  <FDPOS/>
  <COLNO/>
  <DTSTRING/>
  <SYST2>
    <STINFO/>
    <TABIX/>
    <TFILER/>
  </SYST2>
</SYSTEMINFO>
```

The Structure SYSTEMINFO has a string called DTSTRING and a Structure called SYST2.

A Table can contain an elementary type, consist of elementary types, or contain Nested Structures and Tables.

The following list provides the combinations of types that are possible:

- Structures consisting of a series of elementary data types of fixed length (non-Nested, flat Structures).

- An internal Table whose line type is an elementary type.
- Internal Tables whose line type is a non-Nested structure, which can be considered a real table.
- Structures with Structures as components (Nested Structures, Flat or Deep).
- Structures containing internal tables as components (Deep Structures).
- Internal Tables whose line type contains further internal Tables.

The following is a sample XML structure of a Nested Table:

```
<INBOUNDINFO>
<_SAPCND_SRMALIN>
  <item>
    <CLIENT/>
    <SCALE_ID/>
    <SCALE_LINE_ID/>
    <NUM_FLIGHTS/>
    <DBACTION/>
  </item>
</_SAPCND_SRMALIN>
<_SAPCND_SRMDEF>
  <item>
    <CLIENT/>
    <SCALE_ID/>
    <SCALE_BASE_TYPE/>
    <SCALE_TYPE/>
    <DBACTION/>
  </item>
</_SAPCND_SRMDEF>
```

The Oracle Application Adapter for SAP R/3 supports all of these types, except for a Nested Table whose type (definition) is itself a line type. This is supported on request only because of the complexity of the Structure.

4.2 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:

```
<ORACLE_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 SAP R/3 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
```

4.3 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*.

4.4 Creating a Repository Configuration

Before you use Application Explorer with Oracle Application Adapter for SAP R/3, you must create a repository configuration. You can create two types of repository configurations, Web services and J2CA, depending on the container to which the adapter is deployed.

This section contains the following topics:

- [Section 4.4.1, "Creating a Configuration for BSE"](#)
- [Section 4.4.2, "Creating a Configuration for J2CA"](#)
- [Section 4.4.3, "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.

4.4.1 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:

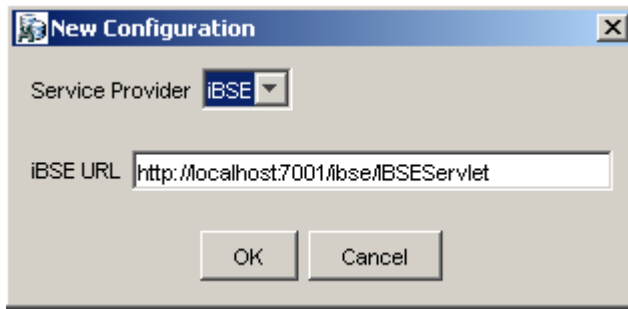
- [Section 4.4.1.1, "Defining a New Configuration for BSE"](#)

4.4.1.1 Defining a New Configuration for BSE

To define 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.

Figure 4–1 New Configuration Dialog



4. From the **Service Provider** list, select **iBSE**, as shown in [Figure 4–1](#).
5. In the **iBSE URL** field, accept the default URL or replace it with a different URL with the following format:

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

Where *host name* is the system on which Oracle WebLogic Server resides and *port* is the HTTP port number of the managed Oracle WebLogic Server (for example, *soa_server1*).

6. Click OK.

A node representing the new configuration appears beneath the root Configurations node, as shown in [Figure 4–2](#).

Figure 4–2 New Configuration Node



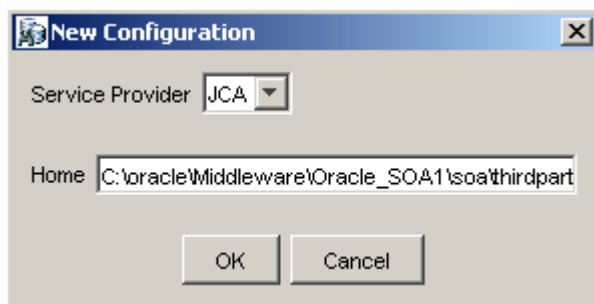
4.4.2 Creating a Configuration for J2CA

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

Figure 4–3 New Configuration Dialog



4. From the **Service Provider** list, select JCA, as shown in [Figure 4-3](#).
5. Click OK.
A node representing the new configuration appears beneath the root Configurations node, as shown in [Figure 4-4](#).

Figure 4-4 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).

4.4.3 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.

Events are not applicable when using a BSE configuration. You can configure events using a J2CA configuration only.

An example of a BSE configuration named SampleConfig, is shown in [Figure 4-5](#).

Figure 4-5 SampleConfig



- Use the **Adapters** node to create inbound interaction with SAP R/3. For example, you use the SAP node in the Adapters node to configure a service that updates SAP R/3.
- Use the **Events** node to configure listeners that listen for events in SAP R/3.
- Use the **Business Services** node (available for BSE configurations only) to test Web services. 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 SAP R/3.

4.5 Establishing a Connection (Target) for SAP R/3

Defining the application includes adding a target for Oracle Application Adapter for SAP R/3. Setting up the target in Application Explorer requires information that is specific to the target.

This section contains the following topics:

- [Section 4.5.1, "Defining a Target to SAP R/3"](#)
- [Section 4.5.2, "Connecting to a Defined SAP R/3 Target"](#)
- [Section 4.5.3, "Managing a Connection to SAP R/3"](#)

To browse the available business functions, you must first define a target to SAP R/3. After you define the target, it is automatically saved. You must connect to the SAP R/3 system every time you start Application Explorer or after you disconnect.

When you launch Application Explorer, the left pane displays (as nodes) the application systems supported by Application Explorer, based on the adapters that are installed.

4.5.1 Defining a Target to SAP R/3

To connect to SAP R/3 for the first time, you must define a new target. Oracle Application Adapter for SAP R/3 supports SAP R/3 standard security and the additional protocol of SNC. Once connected to the SAP R/3 application server, application security is managed by user ID, roles and profiles. For more information on SAP application security, see the appropriate SAP documentation.

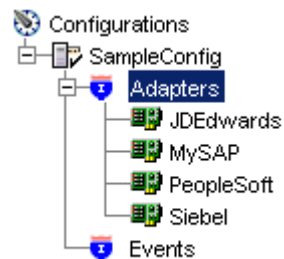
When you are working with a J2CA configuration, creating, updating, and deleting a target requires you to restart the application server. The application server must also be restarted after you create a target, connect to a target, and generate a WSDL for a SAP R/3 business object. In addition, ensure to close Application Explorer before you restart the application server.

To define a target:

1. In the left pane, expand the Adapters node, as shown in [Figure 4-6](#).

The application systems supported by Application Explorer appear as nodes based on the adapters that are installed.

Figure 4-6 Adapters Node



2. Right-click the MySAP node and select **Add Target**, as shown in [Figure 4-7](#).

Figure 4–7 MySAP Node

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

- a. In the **Name** field, enter a descriptive name, for example, **SAPTTarget**.
- b. In the **Description** field, enter a description for the target (optional).
- c. From the **Type** list, select the type of target you are connecting to. The supported target types include **Message Server** or **Application Server** (default).

Note: For load balancing purposes, application servers from one SAP R/3 system are usually configured in logon groups, where each group serves a particular type of user. The application servers in each group are assigned to users by a least-heavily-loaded strategy. This load balancing is done by message servers. Each SAP R/3 system has exactly one message server, which can be reached through TCP on a specific message server port.

3. Click **OK**.

The Application Server dialog is displayed, as shown in [Figure 4–8](#).

Figure 4–8 Application Server Dialog

 A screenshot of a dialog box titled 'Application Server'. It has four tabs: 'User', 'System', 'Advanced', and 'Security'. The 'User' tab is selected. The dialog contains several input fields: 'Client*' (required), 'User*' (required), 'Password*' (required), 'Language*' (with 'EN' entered), 'Codepage', and 'Authentication mode' (a dropdown menu with 'Password' selected). At the bottom, there are 'OK' and 'Cancel' buttons. A red text label at the bottom left reads 'Fields marked with * are required.'

The following tabs are available:

- User (Required)
 - System (Required)
 - Advanced
 - Security
4. For the **User** tab (required), enter the appropriate information for your SAP R/3 target based on the information in the following table.

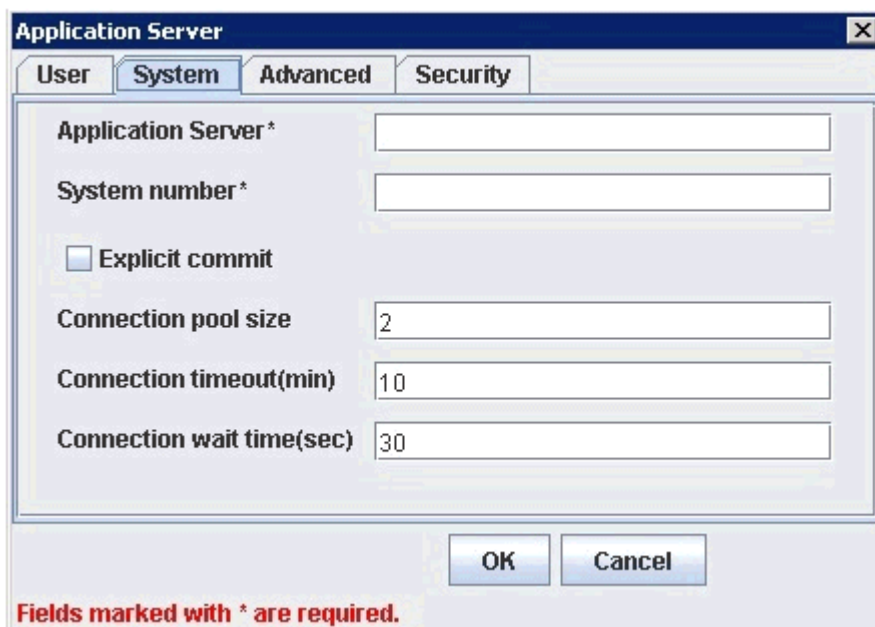
Table 4–1 User Tab Parameters

Target Parameter	Description
Client	The client number defined for the SAP R/3 application for client communications.
User	A valid user ID for the SAP R/3 application.
Password	A valid password for the SAP R/3 application.
Language	A language key. EN (English) is the default.
Codepage	A character code page value.
Authentication Mode	The authentication mode you want to use when connecting to your SAP R/3 system. By default, Password is selected from the list.

For more information, see your SAP R/3 system documentation.

- For the **System** tab (required), enter the appropriate information for your SAP R/3 target based on the information in this section, as shown in [Figure 4–9](#).

Figure 4–9 System Tab



The System tab enables you to provide the application server name, system number, and connection pooling information for the SAP R/3 system to which you are connecting.

Table 4–2 System Tab Parameters

Target Parameter	Description
Application Server	The host name or IP address for the system that is hosting the SAP R/3 application.
System number	The system number defined to SAP R/3 for client communications.

Table 4–2 (Cont.) System Tab Parameters

Target Parameter	Description
Explicit Commit	Select this option to have the BAPI_TRANSACTION_COMMIT function called after every function call that is made by the adapter. Since enabling this option can increase the processing time, it is recommended to use this functionality only if explicit commit functionality is required. This option enables asynchronous commits. The BAPI_TRANSACTION_COMMIT function is called and returns immediately. It is still possible for the database commit to fail, but no notification would be received. This option is the equivalent to the COMMIT WORK ABAP statement.
Connection pool size	<p>The number of client connections in a pool you want to make available to SAP R/3 for Web service calls. A default connection pool size of 2 is available by default.</p> <p>Important: The default value of 1 does not create a connection pool. Instead, a single SAP R/3 connection with sequential processing is shared. A pooled connection invokes multiple connections to SAP R/3 with parallel processing.</p> <p>If you are using Application Explorer to create Web services, then the connection pool size value is used by your Web service during run-time. As a result, ensure that the connection pool size is sufficient for your purposes.</p>
Connection timeout(min)	The timeout value for your connection pool in minutes. The default value is 10 minutes.
Connection wait time(sec)	<p>The wait time for your connection pool in seconds. The default value is 30 seconds.</p> <p>Connections to an ERP server take up valuable resources on both the client and the remote server. You can create a pool of connections to minimize the resource and time constraints. In estimating the size of the pool, you may calculate pool size by the amount of server resources to be consumed, the number and size of the documents to be received, and the size of your Java Virtual Machine. The section of SAP documentation "Memory Management (BC-CST-MM)" explains in detail the resources required on the SAP R/3 system.</p>

- For the **Advanced** tab (optional), enter the appropriate information for your SAP R/3 target based on the information in this section.

The Advanced tab enables you to specify your EDI and IDoc versions, and configure error handling, as shown in [Figure 4–10](#).

Figure 4–10 Advanced Tab

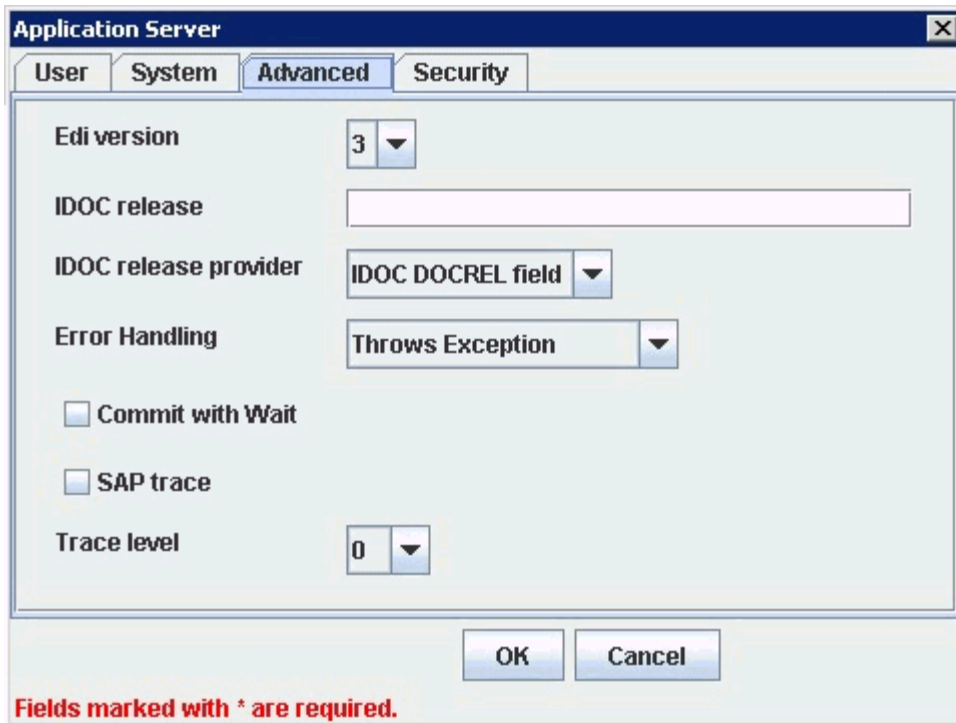


Table 4–3 Advanced Tab Parameters

Target Parameter	Description
Edi version	The Electronic Data Interchange (EDI) document version you are using with the adapter. Version 3 is the default value.
IDOC release	The IDOC versioning you want to use for your connection.
IDOC release provider	The IDOC release provider for your connection. Select IDOC DOCREL field (default), SAP release, or user input from the list.
Error Handling	<p>From the list In the event of an exception, you can select Creates Error Document or Throws Exception. To receive more detailed error messages, select Creates Error Document.</p> <p>As a rule:</p> <ul style="list-style-type: none"> ▪ If your application is Java centric, then select Throws Exception so that code components can catch the exception and react accordingly. ▪ If your application is document based, then select Creates Document to create an XML document that contains the Java exception. <p>It is up to your application to read the XML document and obtain the error.</p>

Table 4–3 (Cont.) Advanced Tab Parameters

Target Parameter	Description
Commit with Wait	<p>If a high degree of accuracy is required in your application, then select the Commit with Wait option.</p> <p>The adapter waits until all records are physically written to the database before returning from the function call. The “Commit With Wait” has a performance impact on adapter performance, so consider carefully before selecting it. For more information about the commit behavior of BAPIs, see the SAP documentation under “BAPI Programming Guide and Reference (CA-BFA).”</p> <p>All SAP Business Objects that change data must commit work to the database. Some BAPIs developed in version 3.1 of the R/3 system use an internal commit behavior, and their commit behavior cannot be changed by the adapter. As soon as they are called, they commit the work they did.</p> <p>BAPIs developed since release 3.1 use the external commit method. The adapter issues a commit command, and the commit is put in the database queue. If there is an application error in the first part of the commit, then the error message “Posting could not be carried out” is returned, and the adapter rolls back the transaction. If writing to the database, a database error occurs, then a short dump is issued in the database records of SAP, but no message is returned to the adapter about the failure.</p> <p>The Commit with Wait option must be used in conjunction with the Explicit Commit option enabled in the System tab. The Commit with Wait option enables synchronous commits. This commit function waits until the actual database commit is made or an error is generated. This process is time intensive, but the most accurate. If an error is generated, then it is returned in an exception message.</p> <p>This option is disabled by default.</p>
SAP trace	<p>Select this option to enable SAP traces.</p> <p>The SAP traces are stored in the following locations during design time and run time:</p> <p>Design Time: <code><ADAPTER_HOME>\tools\iwae\bin</code></p> <p>Run Time: <code><ORACLE_HOME>\user_projects\domains\base_domain</code></p> <p>The file name formats that are used for the SAP traces are as follows (for design time and run time):</p> <ul style="list-style-type: none"> ■ rfc02664_04332.trc ■ JCO100226_043846994.trc
Trace level	<p>Indicates the level of detail in the SAP traces. Select a value that ranges from 0 through 10 from the list.</p>

7. For the **Security** tab (optional), enter the appropriate information for your SAP R/3 target based on the information in this section, as shown in [Figure 4–11](#).

Figure 4–11 Security Tab

The screenshot shows a dialog box titled 'Application Server' with a blue header and a close button in the top right. Below the header are four tabs: 'User', 'System', 'Advanced', and 'Security'. The 'Security' tab is selected. The dialog contains the following fields:

- SNC mode:** A dropdown menu with '0' selected.
- SNC partner:** A text input field.
- SNC level:** A dropdown menu with '1' selected.
- SNC name:** A text input field.
- SNC library path:** A text input field.

At the bottom of the dialog are 'OK' and 'Cancel' buttons. A red text label at the bottom left reads: 'Fields marked with * are required.'

The Security tab enables you to specify Secure Network Communication (SNC) information for the SAP R/3 system to which you are connecting.

Note: The SAP SNC parameters are documented in this documentation as this functionality is available from the SAP Java Connector (SAP JCo). It is strongly recommended that if SNC is to be used with the adapter, a SAP Router be configured with SNC enabled rather than direct SNC from the adapter. This is the strongest, most secured method of connection. Because SNC depends on the validity of certificates issued by the local SAP host server, it is the responsibility of the local SAP support staff to ensure the validity of SNC certificates and correct configuration of the SAP Router and SAP SNC.

Table 4–4 Security Tab Parameters

Target Parameter	Description
SNC mode	By default, SNC is disabled. To enable SNC, select 1 from the list.
SNC partner	Enter the name of the RFC server or message server (load balancing) that provides the SNC services.
SNC level	From the list select the version of the SNC library.
SNC name	Enter the name of the SNC library you are using.
SNC library path	Enter the path to the SNC library.

SNC provides protection for the communication links between the distributed components of an SAP R/3 system. Using SNC, SAP R/3 can support products that adhere to the GSS-API Version 2 standard. SNC supports application level (end-to-end security), Smartcard authentication, and single sign-on.

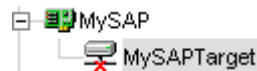
If you are using SAP Enterprise Portal, then the J2EE engine generates the SAP logon ticket automatically. A possible SNC scenario would be from SAP Enterprise Portal to Oracle Application Adapter for SAP R/3.

If you want to use SAP logon tickets to enable single sign-on to non-SAP components, then consult the SAP documentation regarding Pluggable Authentication Services. A possible SNC scenario in this case would be from a non-SAP Enterprise Portal to Oracle Application Adapter for SAP R/3.

8. When you have provided all the required information for your target, click **OK**.

After the extraction finishes, the new target, MySAP Target, appears under the MySAP adapter node, as shown in [Figure 4–12](#).

Figure 4–12 MySAP Target



You can now connect to your SAP R/3 target.

For information on how to create schemas for the adapter, see "[Creating XML Schemas](#)" on page 4-17.

4.5.2 Connecting to a Defined SAP R/3 Target

To connect to an existing target:

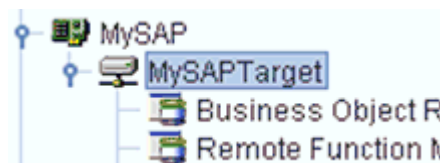
1. In the left pane, expand the Adapters node.
2. Expand the MySAP node.
3. Click the target name under the MySAP node (for example, SAPTarget).

The Connection dialog displays the values you entered for connection parameters.

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

The x icon disappears, indicating that the node is connected, as shown in [Figure 4–13](#).

Figure 4–13 Connected MySAP Target



4.5.3 Managing a Connection to SAP R/3

This section contains the following topics:

- [Section 4.5.3.1, "Disconnecting from a Connection to SAP R/3"](#)
- [Section 4.5.3.2, "Modifying Connection Parameters"](#)
- [Section 4.5.3.3, "Deleting a Connection to SAP R/3"](#)

To manage SAP R/3 connections, you can:

- Disconnect from a connection that is not currently in use.

Although you can maintain multiple open connections to different transaction processing systems, it is recommended to disconnect from connections not in use.

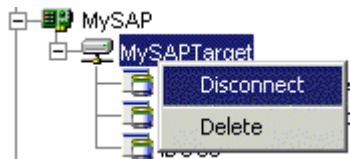
- Edit a target.
You can modify the connection parameters when your system properties change. After you disconnect, you can modify an existing target.
- Delete a connection that is no longer needed.

4.5.3.1 Disconnecting from a Connection to SAP R/3

To disconnect a target:

1. Expand the Adapters node.
2. Expand the MySAP node.
3. Right-click the target to which you are connected, for example, SAPTarget, and select Disconnect, as shown in [Figure 4–14](#).

Figure 4–14 Disconnect Option



Disconnecting from the SAP R/3 target drops the connection with SAP R/3, but the node remains.

The x icon appears, indicating that the node is disconnected, as shown in [Figure 4–15](#).

Figure 4–15 Disconnected Target



4.5.3.2 Modifying Connection Parameters

After you create a target for SAP R/3 using Application Explorer, you can edit any of the information that you provided previously. After you edit the target, you must restart the Oracle WebLogic Server to update the repository for run time purposes.

To edit a target:

1. Verify that the target you want to edit is disconnected.
2. Right-click the target and select **Edit**, as shown in [Figure 4–16](#).

Figure 4–16 Edit Option



The Application Server dialog displays the target connection information.

3. Change the properties in the dialog as required and click **OK**.

4.5.3.3 Deleting a Connection to SAP R/3

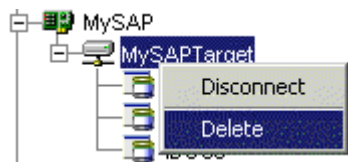
You can delete a connection, rather than just disconnecting and closing it. When you delete the connection, the node disappears from the list of SAP R/3 connections in the left pane of Application Explorer.

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

To delete a connection to SAP R/3:

1. Locate the target you want to delete.
2. Right-click the target (for example, SAPTarget), and select Delete, as shown in [Figure 4-17](#).

Figure 4-17 Delete Option



The node disappears from the list of available connections.

4.6 Viewing Application System Objects

As you connect to SAP R/3, Application Explorer enables you to explore and browse SAP R/3 business objects that are used to support existing business processes.

Note: Depending on the release or service pack installed, certain RFCs, for example, RFC_CUSTOMER_GET, may not exist in your particular SAP R/3 system. Therefore, the examples included in this documentation may not be relevant to your system. If this is the case, then you should use the examples as a general reference for adapter functionality and choose an RFC that exists within your SAP R/3 application environment.

As described in SAP Release Note 109533, SAP Function Modules (RFCs) can be delivered with different release statuses. SAP supports only RFCs that are awarded the Released for Customer status. There is no claim to the release independencies of the interfaces and the continued existence/functionality of the modules. For more information on the status of a specific function module, consult your SAP Service Marketplace.

4.7 Creating XML Schemas

After you explore the SAP R/3 business function library and select an object, you can use Application Explorer to create the XML request schema and the XML response schema for that function.

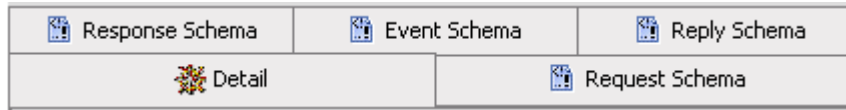
To create request and response schemas for a SAP R/3 business function.

1. Connect to a SAP R/3 target as described in "[Connecting to a Defined SAP R/3 Target](#)" on page 4-15.
2. Expand the **Business Object Repository** node.

3. Click the icon to the left of the **Financial Accounting** node.
4. Scroll down and click the icon to the left of the **CompanyCode** business object.
5. Scroll down and select the BAPI named `GetDetail`.

The XML schema tabs appear on the right, as shown in [Figure 4–18](#).

Figure 4–18 XML Schema Tabs



6. To view the XML for each schema type, click the appropriate tab.

4.8 Generating WSDL (J2CA Configurations Only)

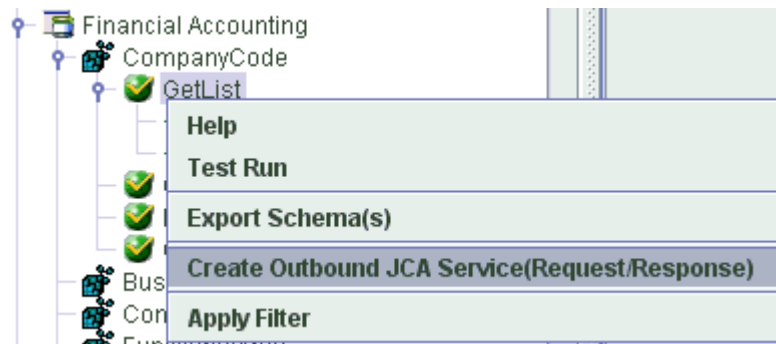
The Web Service Definition Language (WSDL) description of a service enables you to make the service available to other services within a host server. You use Application Explorer to create both request-response (outbound) and event notification (inbound) J2CA services of the adapter.

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. After you create a schema, right-click the respective object.
The `GetList` menu is displayed, as shown in [Figure 4–19](#).

Figure 4–19 Create Outbound JCA Service (Request/Response)



2. Select **Create Outbound JCA Service (Request/Response)**.
The Export WSDL dialog is displayed, as shown in [Figure 4–20](#).

Figure 4–20 Export WSDL Dialog

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. For more information, see ["Generating WSDL for Event Integration"](#) on page 6-37.

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

Using Application Explorer, you can explore the business function repository and generate Web services (also known as a **business service**) for the SAP R/3 functions you want to use with the adapter. The following procedure uses the SAP R/3 BAPI method called `BAPI_MATERIAL_GETLIST` as an example and returns a list of materials from SAP R/3.

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

This section contains the following topics:

- [Section 4.9.1, "Creating a Web Service"](#)
- [Section 4.9.2, "Testing a Web Service"](#)

- [Section 4.9.3, "Identity Propagation"](#)

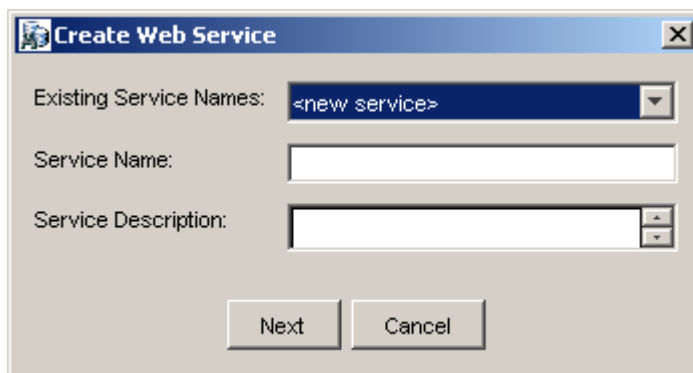
4.9.1 Creating a Web Service

To create a Web service for a SAP R/3 business function:

1. Connect to your **SAP R/3** target and expand the **Business Object Repository** node.
2. Select the `BAPI_MATERIAL_GETLIST` method from the **Business Object Repository**.
3. Right-click the node from which you want to create a business service and select **Create Web Service**.

The Create Web Service dialog is displayed. You can add the business function as a method for a new Web service or as a method for an existing one, as shown in [Figure 4–21](#).

Figure 4–21 Create Web Service Dialog



Perform the following steps:

- a. From the **Existing Service Names** list, select either `<new service>` or an existing service.
 - b. If you are creating a new service, then specify a service name. This name identifies the Web service in the list of services under the **Business Services** node.
 - c. Enter a brief description for the service (optional).
4. Click **Next**.

The License and Method dialog is displayed.

Provide the following information:

- a. In the **License Name** field, select one or more license codes to assign to the Web service. To make multiple selections, press the **Ctrl** key and click the licenses.
 - b. In the **Method Name** field, leave the default method name.
 - c. In the **Method Description** field, enter a brief description of the method (optional).
 - d. In the **DTD Directory** field, specify the location of the DTD you want to use.
5. Click **OK**.

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 **Save WSDL** from the menu.
The Save dialog is displayed.
7. Provide a name for the WSDL file and a location to save the WSDL file on your file system.
8. Click **Save**.

4.9.2 Testing a Web Service

After a Web service is created, you can test it to ensure it functions properly. A test tool is provided for testing the Web service.

To test a Web service:

1. Click the **Business Services** node to access your Web services.
2. Expand the **Services** node.
3. Select the name of the business service you want to test.

The business service name appears as a link in the right pane.

4. In the right pane, click the named business services link.

The test option appears in a separate BSE Servlet page. If you are testing a Web service that requires XML input, then an input field is displayed.

5. Enter the appropriate input.
6. Click **Invoke**.

The BSE Servlet page displays the results.

4.9.3 Identity Propagation

If you test or run a Web service using a third party XML editor, then the user name and password values that you specify in the SOAP header must be valid and are used to connect to SAP R/3. The user name and password values that you provided for SAP R/3 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.

4.10 Configuring an Event Adapter

Events are generated by activity in a database or in an application 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.

Note: Before Oracle Application Adapter for SAP R/3 can be configured to process events (outbound activity) on a SAP system, there are important configuration steps that must take place on the SAP server before the adapter is connected to the SAP system through Application Explorer and channels can be created. For more information, see [Section A.3, "SAP Outbound Message Processing"](#) on page A-9.

This section contains the following topics:

- [Section 4.10.1, "Creating and Editing a Channel"](#)
- [Section 4.10.2, "Schema Validation"](#)

After you create a connection to your application system, you can add events using Application Explorer. To create 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 creating an inbound service. Creating a channel is required for J2CA configurations only. In addition, each channel must be associated with a unique SAP R/3 Program ID. For example, if you are working with MATMAS and DEBMAS, then two separate channels are required for each object and two unique SAP R/3 Program IDs.

A **channel** represents configured connections to particular instances of back-end systems. For more information, see ["Creating and Editing a Channel"](#) on page 4-22.

4.10.1 Creating and Editing a Channel

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

This section contains the following topics:

- [Section 4.10.1.1, "Creating a Channel"](#)
- [Section 4.10.1.2, "Editing a Channel"](#)
- [Section 4.10.1.3, "Deleting 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.

Note: Channels can be configured and started only on the system where the Oracle Application Adapter for SAP R/3 is installed. Configuring and starting a channel for a remote host is not supported.

Events are not applicable when using a BSE configuration. You can configure events using a J2CA configuration only.

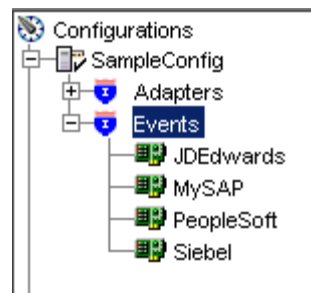
Creating and updating a channel requires you to restart the application server. The application server must also be restarted after you create a channel and generate an inbound WSDL. In addition, ensure to close Application Explorer before you restart the application server.

4.10.1.1 Creating a Channel

To create a channel:

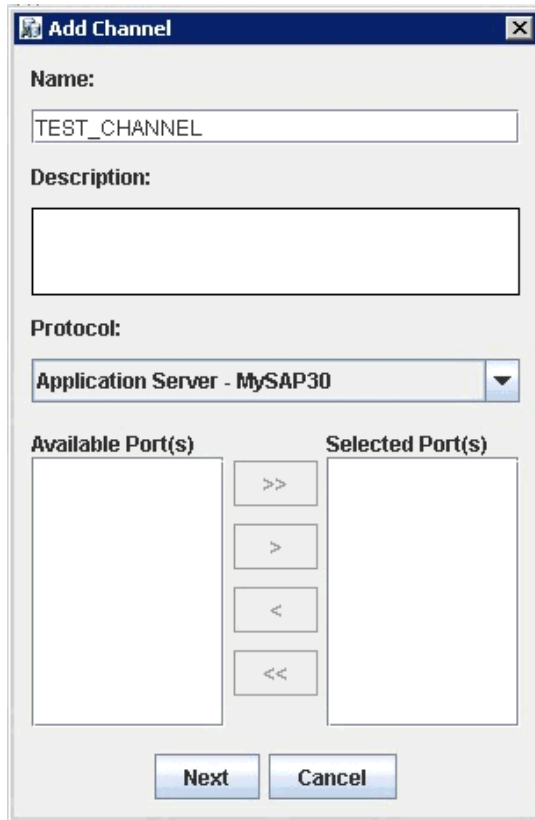
1. Click the **Events** node, as shown in [Figure 4–22](#).

Figure 4–22 Events Node



2. Expand the **MySAP** node.
The ports and channels nodes appear in the left pane.
3. Right-click **Channels** and select **Add Channel**.
The Add Channel dialog is displayed, as shown in [Figure 4–23](#).

Figure 4–23 Add Channel Dialog



Perform the following steps:

- a. Enter a name for the channel, for example, TEST_CHANNEL.
 - b. Enter a brief description.
 - c. From the **Protocol** list, select **Application Server - MySAP30** or **Message Server - MySAP30**.
4. Click Next.

The Message Server dialog is displayed. The following tabs are available:

- User (Required)
 - System (Required)
 - Security
 - Advanced
5. For the **User** tab, enter the appropriate information for your SAP R/3 channel based on the information in the following table.

Table 4–5 User Tab Parameters

Target Parameter	Description
Client	The client number defined for the SAP application for client communications.
User	A valid user ID for the SAP application.
Password	A valid password for the SAP application.

Table 4–5 (Cont.) User Tab Parameters

Target Parameter	Description
Authentication Mode	The authentication mode you want to use when connecting to your SAP R/3 system. By default, Password is selected from the list.

6. For the **System** tab, enter the appropriate information for your SAP R/3 channel based on the information in the following table.

Table 4–6 System Tab Parameters

Target Parameter	Description
Gateway host	A host name for the SAP R/3 Gateway.
Gateway service	A service for the SAP R/3 Gateway.
Program ID of the server	A SAP R/3 program ID you want to use for this channel.
Application Server	A host name for the application server.
System Number	The system number defined to SAP R/3 for client communications.
Accept pool parameters (True False)	If False , use the default adapter connection tuning. If True , use the values provided by the user
Connection pool size	Maximum number of connections in a pool. If this is less than "Maximum Connections," the Pool Size will be set by the JCo to the "Maximum Connections" size.
Maximum connections	Maximum number of connections created at one time.
Connection timeout (min.)	Time in minutes until the unused connections are released.
Connection wait time (sec.)	Time in seconds to wait for a free connection handle when all connections are busy, before throwing a Resource exception.

7. For the **Security** tab (optional), enter the appropriate information for your SAP R/3 channel based on the information in the following table.

Table 4–7 Security Tab Parameters

Target Parameter	Description
SNC mode	By default, SNC is disabled. To enable SNC, select 1 from the list.
SNC partner	Enter the name of the RFC server or message server (load balancing) that provides the SNC services.
SNC level	From the list select the version of the SNC library.
SNC name	Enter the name of the SNC library you are using.
SNC library path	Enter the path to the SNC library.

8. For the **Advanced** tab (optional), enter the appropriate information for your SAP R/3 channel based on the information in the following table.

Table 4–8 Advanced Tab Parameters

Target Parameter	Description
IDOC Confirm	<p>If this option is selected, a STATUS IDOC is returned to SAP confirming receipt for every IDOC that is received.</p> <p>When a standard IDOC is sent, the usual last status available through the SAP IDOC monitor is "Data Sent to Port OK" - "03", in the WE02 transaction IDOC list. If you want to use the WE02 transaction to ensure that the IDOC was received at the target destination, then enable this option. When an IDOC is received, the adapter returns a STATUS IDOC to SAP and in the WE02 transaction IDOC list, the status of the IDOC is changed to "12", "Dispatch OK".</p> <p>Note that if a large number of IDOCs are sent in bursts, enabling this option may impact performance. If performance is an issue, then it is recommended to disable this option.</p>
IDOC Format	<p>Select an IDOC type from the list:</p> <ul style="list-style-type: none"> ■ XML (default) ■ XML-CDATA-ENVELOPED
IDOC release	The IDOC versioning you want to use for your connection.
IDOC release provider	<p>The IDOC release provider for your connection. Select IDOC DOCREL field (default), SAP release, user input, or Automatic check from the list.</p> <p>For more information about the Automatic check option, see the extended description that follows this procedure.</p>
SAP trace	<p>Select this option to enable SAP traces.</p> <p>The SAP traces are stored in the following locations during design time and run time:</p> <p>Design Time:</p> <pre><ADAPTER_HOME>\tools\iwaeb\bin</pre> <p>Run Time:</p> <pre><ORACLE_HOME>\user_projects\domains\base_domain</pre> <p>The file name formats that are used for the SAP traces are as follows (for design time and run time):</p> <ul style="list-style-type: none"> ■ rfc02664_04332.trc ■ JCO100226_043846994.trc
Trace level	Indicates the level of detail in the SAP traces. Select a value that ranges from 0 through 10 from the list.
Thread count	The number of open threads. By default, three threads are specified.
Processing Mode	<p>Select the type of synchronous processing from the list. Possible values include REQUEST and REQUEST_RESPONSE.</p> <p>If a channel is created with a processing mode set to REQUEST_RESPONSE, then this channel is listed during the WSDL creation process only if the object has a reply schema. Otherwise, the channel will not be listed.</p>

9. Click **OK**.

The channel appears under the channels node in the left pane, as shown in [Figure 4–24](#).

Figure 4–24 MySAP_Channel

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

Note: If you are planning to integrate Oracle Application Adapter for SAP R/3 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 all the channels and close the Application Explorer before run-time deployment for events (when working with BPM, BPEL, Mediator, or OSB process components).

10. Right-click the **channels** node and select **Start**.

The channel you created becomes active, as shown in [Figure 4–25](#).

Figure 4–25 Active MySAP_Channel

The X over the icon disappears.

11. To stop the channel, right-click the connected channel node and select **Stop**.

The channel becomes inactive and an X appears over the icon.

IDOC Release Provider Parameter: Automatic Check Option

If you have non-standard IDOC releases, earlier versions of the adapter required you to select the **user input** option, and manually specify the IDOC releases. However, this option applies to all IDOCs that are handled by the target, which can be a limitation. The **Automatic check** option allows the adapter to read the SAP Partner Profile from the Logical System that is available through the WE02 transaction.

The API matches the Logical System for the Program ID and the incoming IDOC type for the message, which retrieves the IDoc version from the Partner Profile through the EDI partner API.

The **Automatic check** option retains all IDOC metadata configuration on the SAP system where the IDOC is being sent, without requiring you to guess what version of SAP an IDOC type is configured.

The following options are also available for the IDOC Release Provider parameter:

- **IDOC DOCREL field** - Uses the DOCREL field of the IDOC to obtain the metadata release of the IDOC.

- **SAP release** - Uses the logon SAP release to obtain the metadata release of the IDoc.
- **user input** - Requires you to enter the SAP release of the IDOC that is being sent (for example, 40A, 40B, 45B, 700, and so on).

Consider the following about IDOC terms:

- **Message Types** - Describe business messages. The message type is not direction specific, but describes in general terms a business process, such as Order for Order Processing. A message type is usually independent of the release and version.
- **Segments** - IDOC data is composed of groups or segments of fields. Each segment is released or bound to a particular SAP release. There may be only one active segment version on a particular SAP release at a time.

A basic type or IDOC type describes a physical message composed of a hierarchy of segments composed in a particular order. Each segment belongs to a particular SAP version. Each IDOC type is an order of a type of segments. Each IDOC type is released or bound to a particular version of SAP. If you are required to send a message to a partner with an earlier version of SAP, then you must send it using an earlier IDOC version and an earlier SAP IDOC release. In an IDOC, the order of fields vary. However, the number of fields at the end of each segment continues to be added with each segment version increase and the number of segments increase with each IDOC version increase.

4.10.1.2 Editing a Channel

To edit a channel:

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

The Edit Channel pane is displayed.

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

When you edit a channel, you must restart the Oracle WebLogic Server to recognize the change and update the repository for run time purposes.

4.10.1.3 Deleting a Channel

To delete a channel:

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

The channel disappears from the list in the left pane.

When you delete a channel, you must restart the Oracle WebLogic Server to recognize the change and update the repository for run time purposes.

4.10.2 Schema Validation

Root validation, namespace validation, and schema validation for inbound processing (events) are supported for the Oracle Application Adapter for SAP R/3 with BPEL.

To validate inbound processing using the Oracle Application Adapter for SAP R/3, perform the following steps. This procedure uses MATMAS (Material Master) as an example for inbound processing.

1. Start Application Explorer.

2. Connect to the MySAP target.
3. Expand the IDOCs node.
4. Verify that you created a channel for the MySAP adapter.
5. Select and expand the MATMAS – Material Master node.
6. Right-click MATMAS01 select **Create Inbound JCA Service (event)**.
The Export WSDL dialog opens and includes three check boxes for Root, Namespace, and Schema validation, as shown in [Figure 4–26](#).

Figure 4–26 Export WSDL Dialog

- Selection of multiple validation options is allowed.
- Root validation is used to validate the root element in the inbound XML document.
- Namespace validation is used to validate the namespace in the inbound XML document.
- Schema validation is used to validate the inbound XML document with the schema in the WSDL document.
- During run time, validation is processed based on the validation options that are selected.
- If multiple validation options are selected, then during run time if the first validation option fails, the remaining validation options are not processed.

- Root and namespace validations are considered modest levels of validation. Schema validation is a stricter validation level.
 - It is recommended to use root and namespace validation options, unless the root element and namespace are different between the IDOCs in the SAP environment.
7. Generate the WSDL document and create the BPEL process.
 8. Trigger the transactions (IDOCs) from the SAP GUI.
Inbound transactions that fail for the validation are shown in the SAP Transaction Monitor (SM58).
The status text field shows "java.lang.exception" for the documents that have failed the validation process.

4.11 Configuring the SAProuter Service

This section describes how to configure the SAProuter service. It includes the following topics:

- [Section 4.11.1, "Configuring SAProuter"](#)
- [Section 4.11.2, "Testing SAProuter With Oracle Components"](#)
- [Section 4.11.3, "Verifying SAProuter Configuration"](#)

SAProuter can be considered as a proxy server or intermediate endpoint between different types of SAP connections (internal and external). It can act as a firewall or port filter to keep unwanted traffic from entering the internal SAP network or to confine network traffic between specific ports on specific systems.

For more information on SAProuter that is related to its functionality and configuration, refer to the SAProuter documentation at the following Web site:

<http://help.sap.com>

Once you have accessed this Web site, you can search for SAProuter.

The configuration and setup instructions for SAProuter may differ by platform and the version of SAP that is installed.

4.11.1 Configuring SAProuter

To configure SAProuter:

1. Edit or create a route table for SAProuter.

The following is a sample of the route string syntax:

```
/H/ hostname  
/S/ service  
/W/ password
```

Note: The characters between the forward slashes are case-sensitive.

2. Configure a route entry in the permissions table, using the following format:

```
{P/S/D} <source host> [<dest host>] <dest serv> <password>
```

where:

- Specifying *P* in front of the command instructs SAProuter to configure (permit) the connection. P(ermit) entries can contain a password in the `<password>` element.
- Specifying *S* in front of the command only allows connections to be made with the SAP protocol. Connections with other protocols (for example, TCP) are not allowed.
- Specifying *D* in front of the command prevents (denies) the connection from being configured.
- Source Host `<source host>` specifies the host from where the connection originates (from the viewpoint of SAProuter). This can be a host name, an IP address, or an IP subnetwork.
- Destination Host `<dest host>` specifies the host to which the connection is accessing (from the viewpoint of SAProuter). This can be a host name, an IP address, or an IP subnetwork.
- Destination Port `<dest serve>` specifies the port (service) of the destination host to which the connection is accessing. If required, you can also specify port ranges by separating the two ports that demarcate the range by using a period (.) character. If `<dest serv>` has the value 3200.3298, this means connections to the destination server on all ports between 3200 and 3298. For example:

```
p 192.168.128.166 SAP-Server1 3300
```

3. Start SAProuter.

You can now test SAProuter with Oracle components.

4.11.2 Testing SAProuter With Oracle Components

After SAProuter has started without any issues, you can create:

- A target in Application Explorer with the route string for the Application Server.
- A channel with the route string for the Gateway Host and Application Server.

Ensure that connections can be made to the Gateway Host and Application Server without any issues.

For more information on creating a target, see [Section 4.5, "Establishing a Connection \(Target\) for SAP R/3"](#). For more information on creating a channel, see [Section 4.10.1, "Creating and Editing a Channel"](#).

The following is a sample of the route string syntax:

```
/H/host/H/host/S/service
```

where:

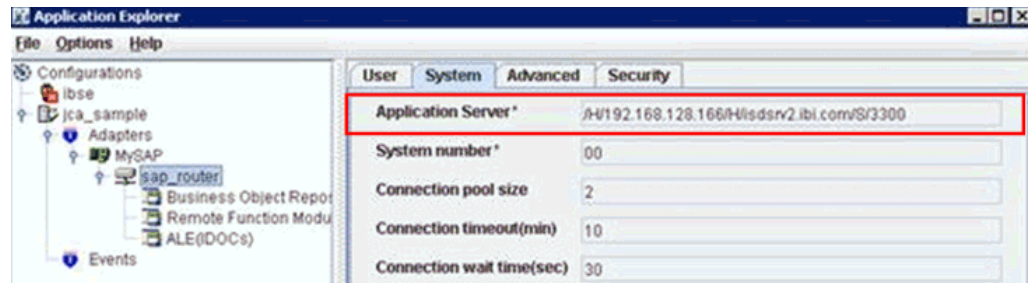
- *host* indicates the host name.
- *service* specifies the service (port).

For example:

```
/H/192.168.128.166/H/isdsrv2.ibi.com/S/3300
```

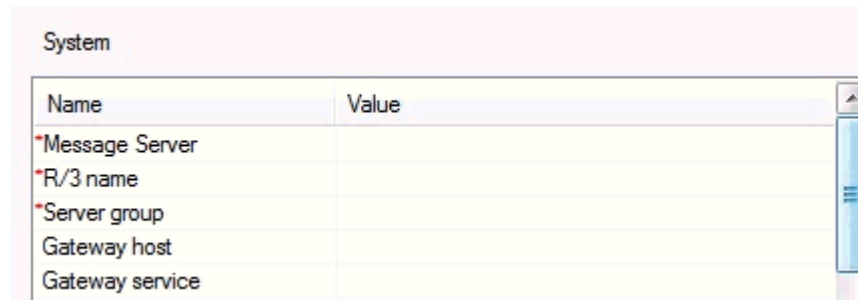
[Figure 4-27](#) shows a route string configured for a target in Application Explorer.

Figure 4–27 Configured Target Route String



To access a message server, you must specify the service name and port for the message server (not the application host), and provide it to the router, as shown in [Figure 4–28](#).

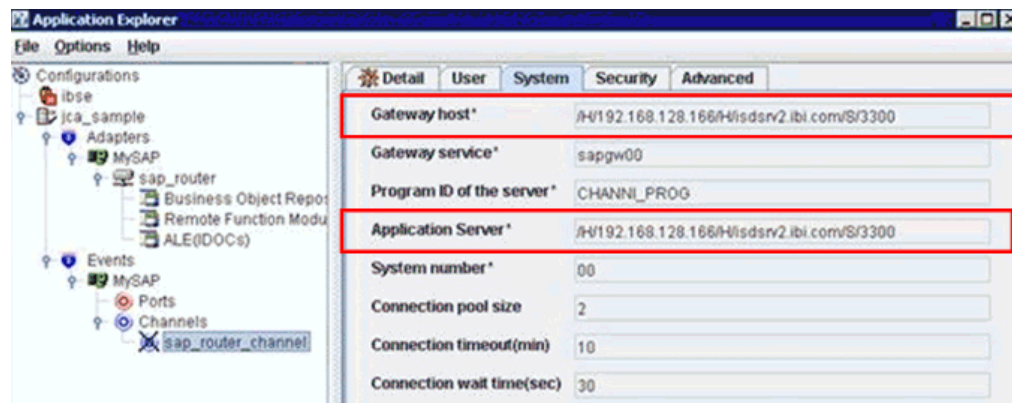
Figure 4–28 Message Server Connection Parameters



You must also specify the logon group to provide connection balancing.

[Figure 4–29](#) shows a route string configured for a channel in Application Explorer.

Figure 4–29 Configured Channel Route String



Once connected, create outbound and inbound WSDL files, and check if the BPEL, Mediator, BPM, and OSB processes are working correctly. You can refer to the corresponding sections in this documentation for more information.

4.11.3 Verifying SAProuter Configuration

As a best practice, start SAProuter initially with traces enabled or the output displayed in the command prompt window to check for errors, and submit a sample command. Use the following command to specify the logging and logging level if required:


```
saprouter -r -T logfile.name -V3
```

where:

- `-T` activates the tracing option.
 - `-V(n)` specifies the logging level.
 - `logfile.name` is a name for the log file (for example, logfile.trc).
1. Once SAProuter is started in the command prompt window, check for any errors in the command prompt or in the trace file, which is generated in the folder where SAProuter is started.

Figure 4–30 shows SAProuter started in the command prompt window with the command to activate tracing.

Figure 4–30 SAProuter Command Prompt



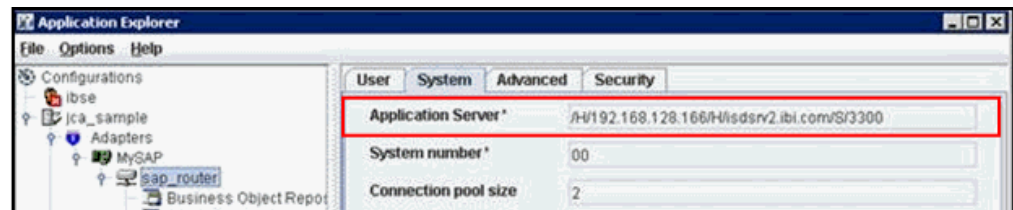
```
C:\sap_router>saprouter -r -T saproutertrace.trc -U3 -G saprouterlog.trc
trcfile saproutertrace.trc
logfile saprouterlog.trc
```

2. After SAProuter is started, create a target in Application Explorer with the route string for the Application Server, and a channel with the route string for the Gateway Host and Application Server.

Ensure that connections can be made to the Gateway Host and Application Server without any issues.

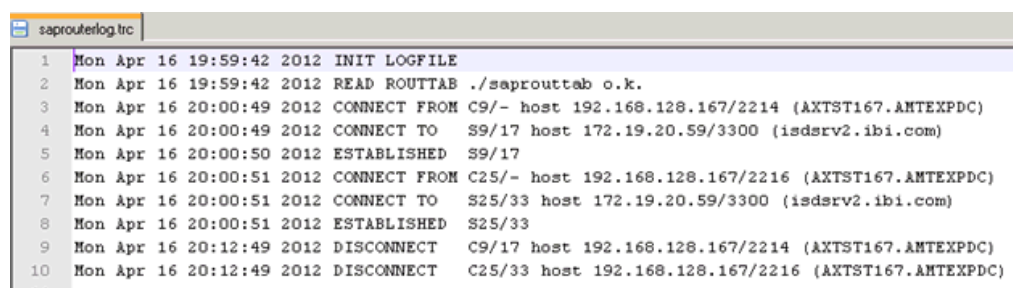
Figure 4–31 shows a route string configured for a target in Application Explorer.

Figure 4–31 Configured Target Route String



3. To verify that the connection has been properly established, see the defined log file (for example, saprouterlog.trc), as shown in Figure 4–32.

Figure 4–32 SAProuter Log File



```
saprouterlog.trc
1 Mon Apr 16 19:59:42 2012 INIT LOGFILE
2 Mon Apr 16 19:59:42 2012 READ ROUPTAB ./saprountab o.k.
3 Mon Apr 16 20:00:49 2012 CONNECT FROM C9/- host 192.168.128.167/2214 (AXTST167.AMTEXPDC)
4 Mon Apr 16 20:00:49 2012 CONNECT TO S9/17 host 172.19.20.59/3300 (isdsv2.ibi.com)
5 Mon Apr 16 20:00:50 2012 ESTABLISHED S9/17
6 Mon Apr 16 20:00:51 2012 CONNECT FROM C25/- host 192.168.128.167/2216 (AXTST167.AMTEXPDC)
7 Mon Apr 16 20:00:51 2012 CONNECT TO S25/33 host 172.19.20.59/3300 (isdsv2.ibi.com)
8 Mon Apr 16 20:00:51 2012 ESTABLISHED S25/33
9 Mon Apr 16 20:12:49 2012 DISCONNECT C9/17 host 192.168.128.167/2214 (AXTST167.AMTEXPDC)
10 Mon Apr 16 20:12:49 2012 DISCONNECT C25/33 host 192.168.128.167/2216 (AXTST167.AMTEXPDC)
```

4. After you have successfully verified the SAProuter configuration, you can stop SAProuter, disable the tracing options, and then restart SAProuter.

Note: After the SAProuter verification has been successfully completed, the Oracle Application Adapter for SAP R/3 target can work with the SAProuter. SAProuter configuration information that is defined in the adapter target is stored in the adapter repository. As a result, the Oracle Application Adapter for SAP R/3 makes use of the SAProuter settings that are defined in the adapter target during runtime when invoked from SOA components, such as BPEL, Mediator, BPM, and OSB.

Oracle WebLogic Server Deployment and Integration

This chapter describes Oracle WebLogic Server (OracleWLS) deployment and integration with Oracle Application Adapter for SAP R/3. It contains the following sections:

- [Section 5.1, "Adapter Integration with Oracle WebLogic Server"](#)
- [Section 5.2, "Deployment of Adapter"](#)
- [Section 5.3, "Updating Adapter Configuration"](#)

See Also:

- *Oracle WebLogic Server Adapter Concepts*

5.1 Adapter Integration with Oracle WebLogic Server

Oracle Application Adapter for SAP R/3 is deployed within an OracleWLS container during installation. All client applications run within the OracleWLS environment. In J2CA deployment, the Common Client Interface (CCI) integrates an OracleWLS client application with a resource adapter.

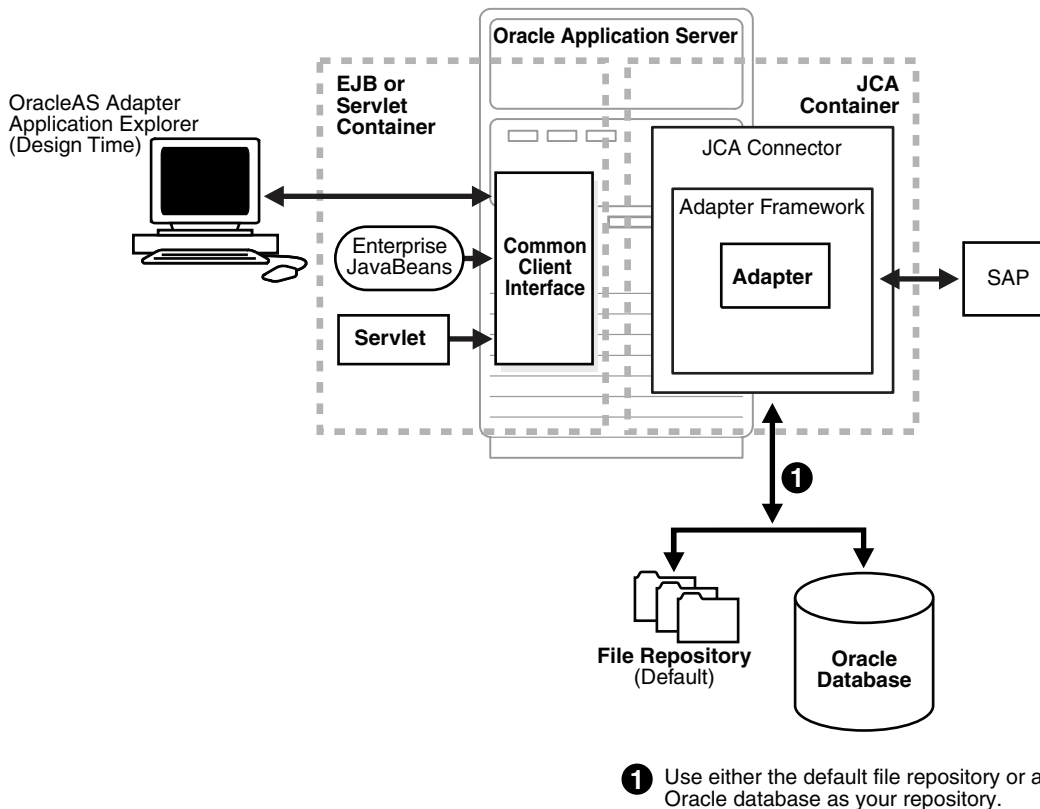
See Also:

- *Oracle WebLogic Server Adapter Concepts*

5.2 Deployment of Adapter

[Figure 5-1](#) shows deployment of the J2CA Connector to the Oracle Application 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 5–1 Oracle Application Server J2CA Architecture

**See Also:**

- *Oracle WebLogic Server Adapter Concepts*

5.3 Updating Adapter Configuration

This section contains the following topics:

- [Section 5.3.1, "Creating a Managed Connector Factory Object"](#)
- [Section 5.3.2, "Creating Multiple Managed Connector Factory Objects"](#)
- [Section 5.3.3, "Modifying WSDL Files for Additional Connection Factory Values"](#)

During the J2CA deployment of Oracle Application Adapter for SAP R/3, OracleWLS generates a deployment descriptor called `ra.xml`, located in:

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

The OracleWLS deployment descriptor is located in the directory specified above.

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

5.3.1 Creating a Managed Connector Factory Object

The `ra.xml` descriptor provides OracleWLS-specific deployment information for resource adapters. For example, the `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>IWAYFJCA10</display-name>
  <vendor-name>IWAY Software</vendor-name>
  <spec-version>1.0</spec-version>
  <eis-type>IWAY</eis-type>
  <version>1.0</version>
  <license>
    <license-required>>false</license-required>
  </license>
  <resourceadapter>

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

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

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

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

```

```

        <config-property-value></config-property-value>
    </config-property>
    <config-property>
        <config-property-name>IWayRepoPassword</config-property-name>
        <config-property-type>java.lang.String</config-property-type>
        <config-property-value></config-property-value>
    </config-property>
    <config-property>
        <config-property-name>LogLevel</config-property-name>
        <config-property-type>java.lang.String</config-property-type>
        <config-property-value>DEBUG</config-property-value>
    </config-property>
    <authentication-mechanism>
        <authentication-mechanism-type>BasicPassword</authentication-mechanism-type>
    </authentication-mechanism>
    <credential-interface>javax.resource.spi.security.PasswordCredential</credential-interface>
    </credential-interface>
    </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 SAP R/3 has a preconfigured <code>jca_sample</code> 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 the configuration. This is necessary only when using an Oracle database as the repository.
loglevel	It overwrites the level set by the <code>ManagedConnectorFactory</code> property.

5.3.2 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-factory-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-factory-interface>
  <connection-instance>
    <jndi-name>eis/OracleJCAAdapter/DefaultConnection</jndi-name>
  </connection-instance>
  <connection-instance>
    <jndi-name>eis/OracleJCAAdapter/DefaultConnection1</jndi-name>
    <connection-properties>
      <properties>
        <property>
<name>IWayHome</name>
<value>C:\oracle\Middleware\Oracle_SOA1\soa\thirdparty\ApplicationAdapters</value>
        </property>
        <property>
<name>IWayConfig</name>
<value>jca_sample2</value>
        </property>
        <property>
<name>IWayRepoURL</name>
<value></value>
        </property>
        <property>
<name>IWayRepoUser</name>

```

```

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

```

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 be created in Application Explorer.

Note: When you modify the `ra.xml` and `weblogic-ra.xml` files, the application server must be restarted. If the application server is running, then stop the application server and then restart it.

In addition, the `iwafjca.rar` file must be redeployed in the Oracle WebLogic Administration Console to activate these changes.

5.3.3 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 should 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 SAP R/3 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="MySAP" 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="MySAP" 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.

Integration With BPEL Service Components in the Oracle SOA Suite

Oracle Application Adapter for SAP R/3 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 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 BPEL Process for Service Integration \(J2CA Configuration\)"](#)
- [Section 6.5, "Designing an Inbound BPEL Process for Event Integration \(J2CA Configuration\)"](#)
- [Section 6.6, "Working With Synchronous SAP R/3 Events Using BPEL \(J2CA Configuration\)"](#)
- [Section 6.7, "Designing an Outbound BPEL Process for Service Integration \(BSE Configuration\)"](#)

6.1 Overview

To integrate with Oracle BPEL Process Manager, Oracle Application Adapter for SAP R/3 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 4-18.

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 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, and administer BPEL processes, and listen to adapter events.

6.2 Deployment of Adapter

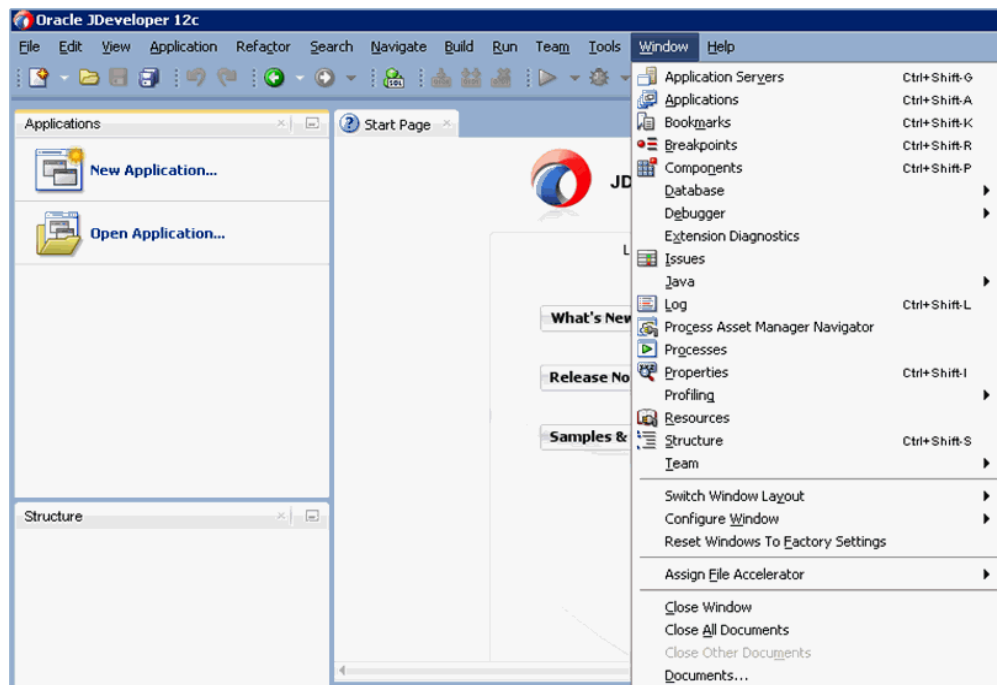
During installation, Oracle Application Adapter for SAP R/3 is deployed as a J2CA 1.0 resource adapter within the WLS container. The adapter must be deployed in the same WLS container as Oracle BPEL Process Manager.

6.3 Configuring a New Application Server Connection

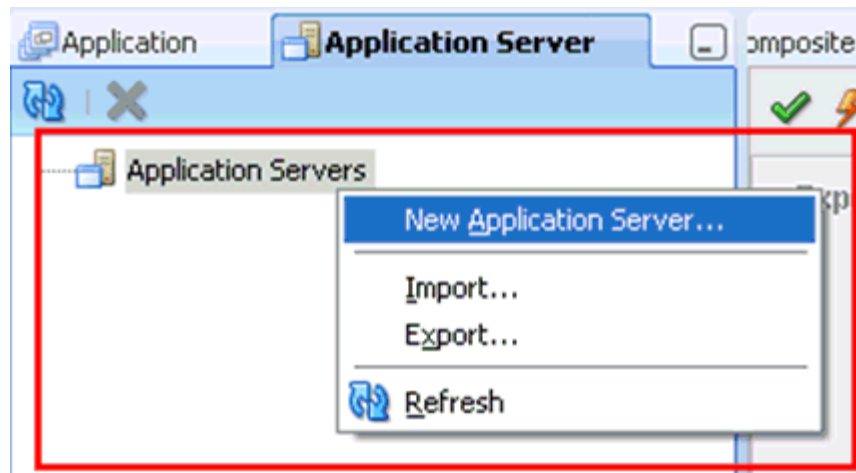
To configure a new Application Server connection in Oracle JDeveloper:

1. Open **Oracle JDeveloper** on your system.
2. From the menu bar, click **Window** and select **Application Server Navigator**, as shown in [Figure 6-1](#).

Figure 6-1 Application Server Navigator

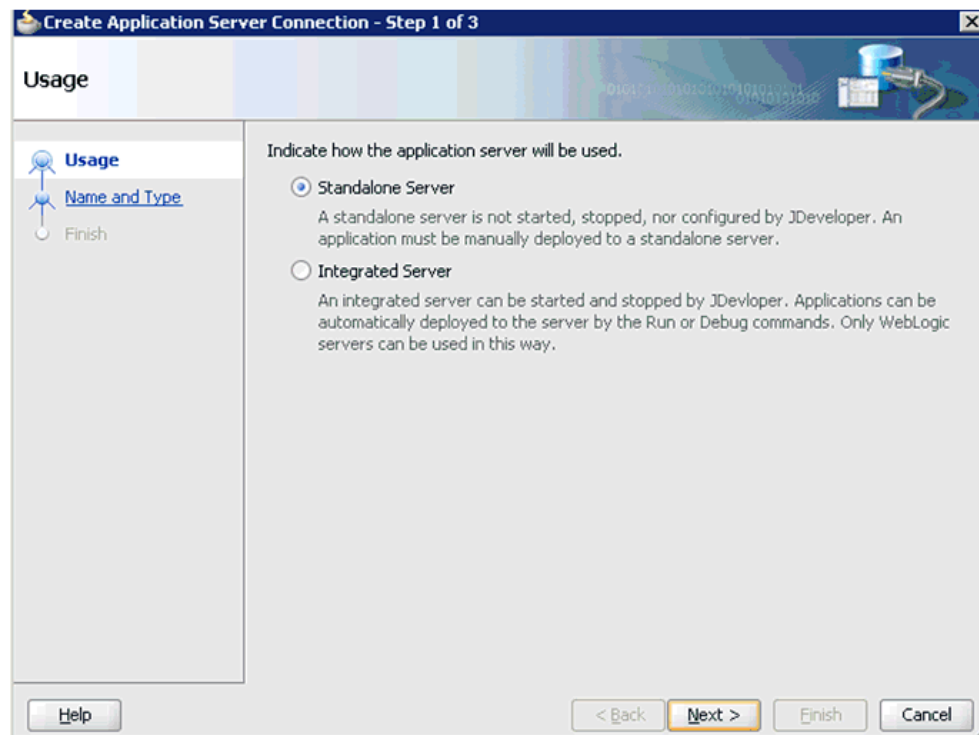


The Application Server tab is displayed, as shown in [Figure 6-2](#).

Figure 6–2 Application Server Tab

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

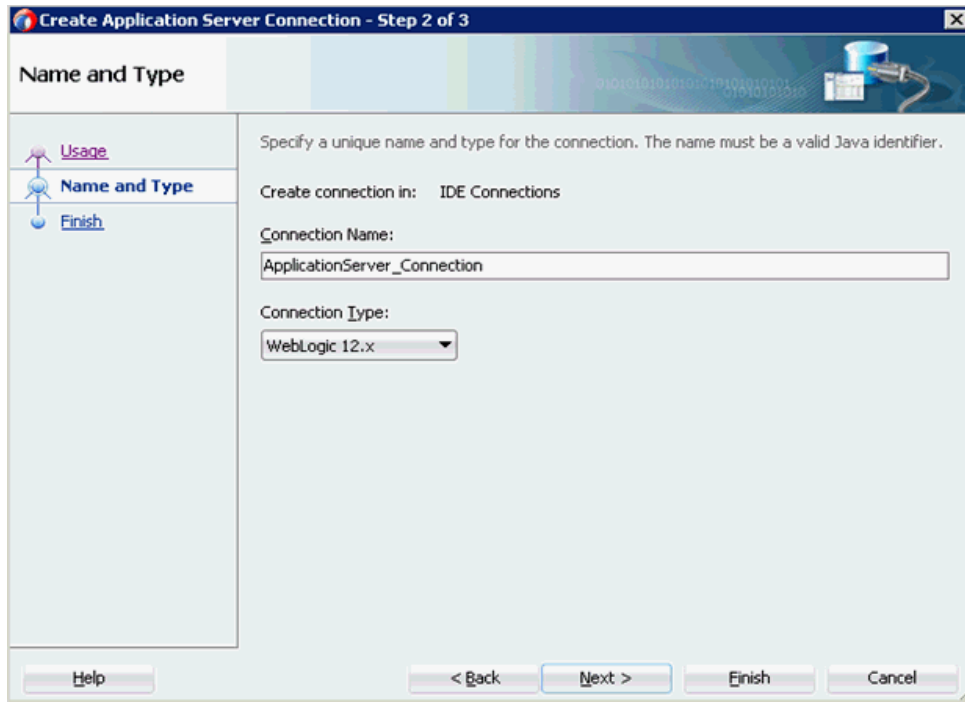
The Create Application Server Connection Wizard is displayed, as shown in [Figure 6–3](#).

Figure 6–3 Create Application Server Connection Wizard

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

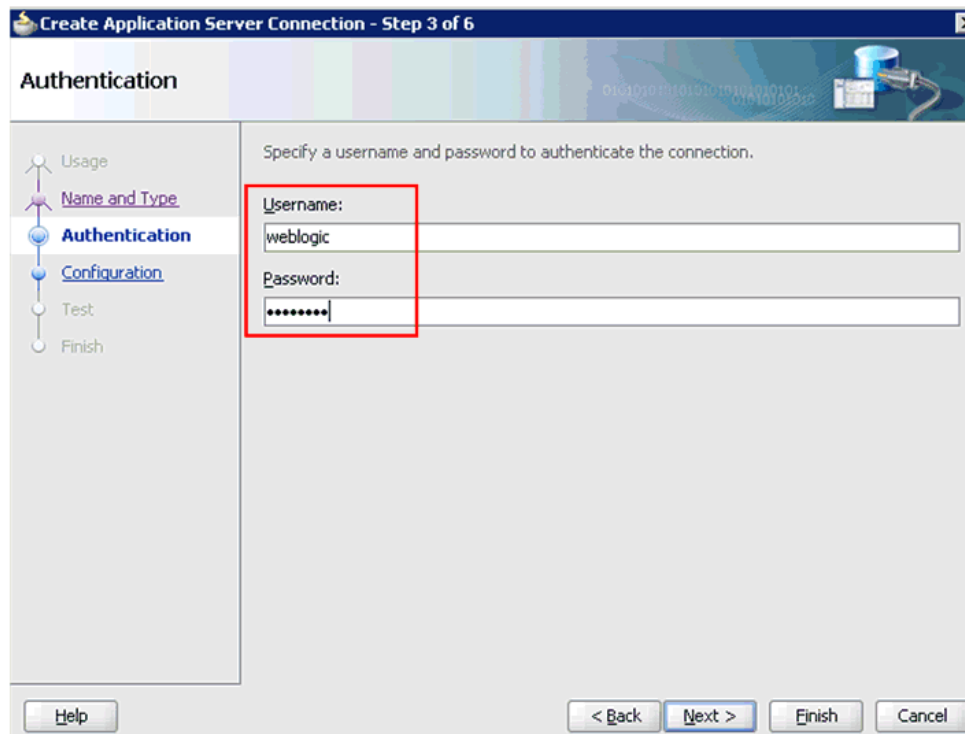
The Name and Type page is displayed, as shown in [Figure 6–4](#).

Figure 6–4 Name and Type Page



5. Specify a new name for the Application Server connection and click **Next**.
The Authentication page is displayed, as shown in [Figure 6–5](#).

Figure 6–5 Authentication Page



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 6-6](#).

Figure 6-6 Configuration Page

Create Application Server Connection - Step 4 of 6

Configuration

Usage
Name and Type
Authentication
Configuration
Test
Finish

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

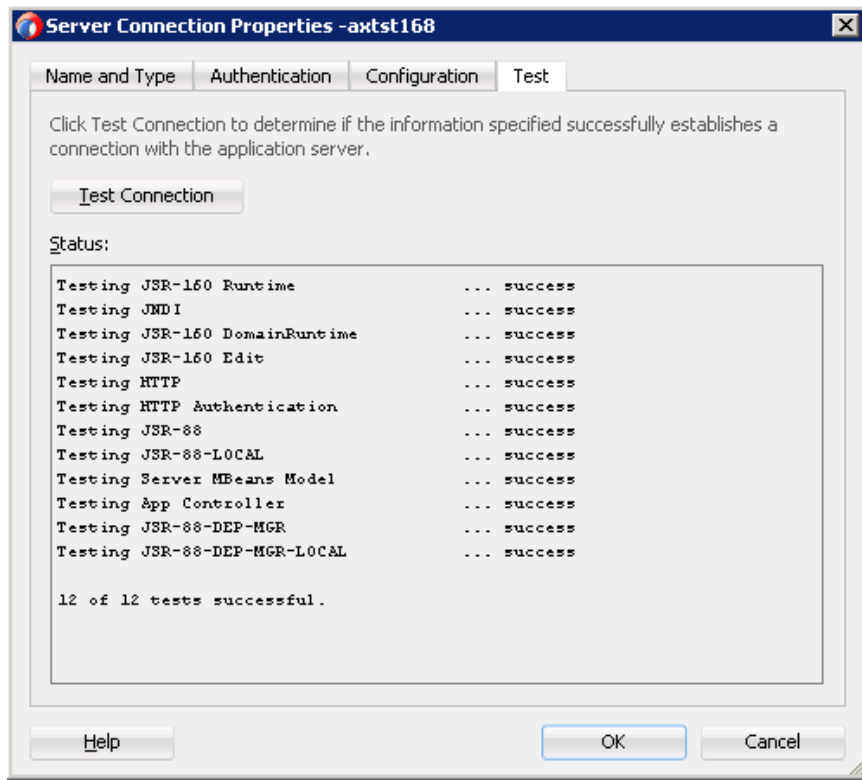
WebLogic Domain:
base_domain

Help < Back Next > Finish Cancel

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 6-7](#).

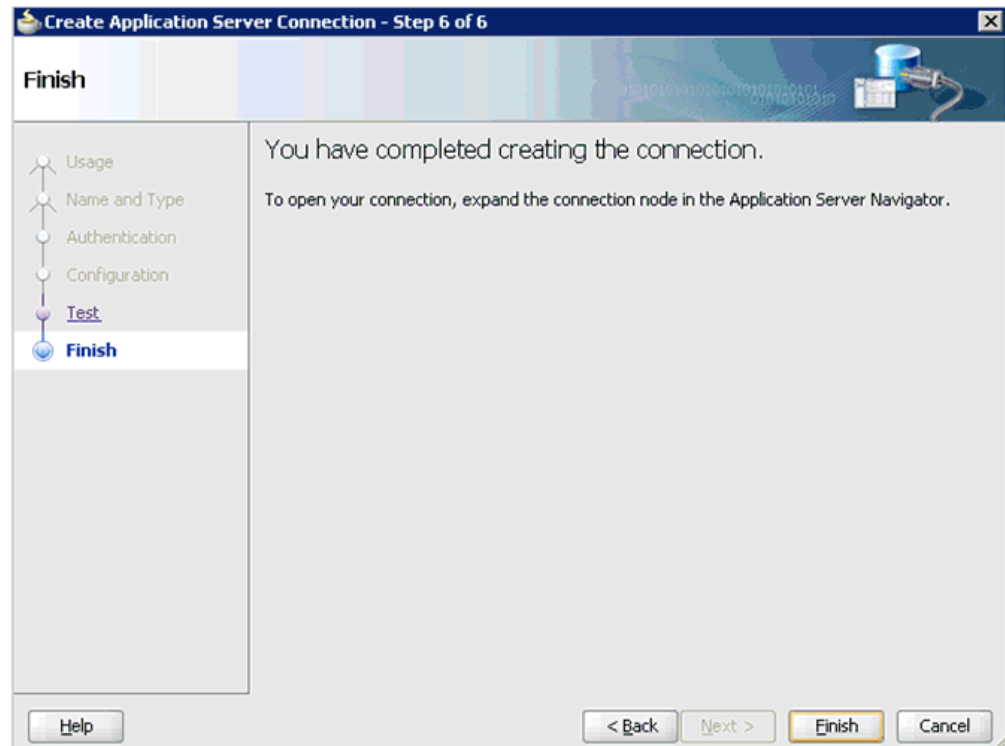
Figure 6–7 Test Page



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 6–8](#).

Figure 6–8 Finish Page



13. Click **Finish**.

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

6.4 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\SAP_Samples.zip\SAP_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 6.4.1, "Generating WSDL for Request/Response Service"](#)
- [Section 6.4.2, "Creating an Empty Composite for SOA"](#)

- [Section 6.4.3, "Defining a BPEL Outbound Process"](#)
- [Section 6.4.4, "Deploying the BPEL Outbound Process"](#)
- [Section 6.4.5, "Invoking the Input XML Document in the Oracle Enterprise Manager Console"](#)
- [Section 6.4.6, "Testing Outbound BPEL and Mediator Processes"](#)

6.4.1 Generating WSDL for Request/Response Service

Before you design a BPEL process, you must generate the respective WSDL file using Application Explorer.

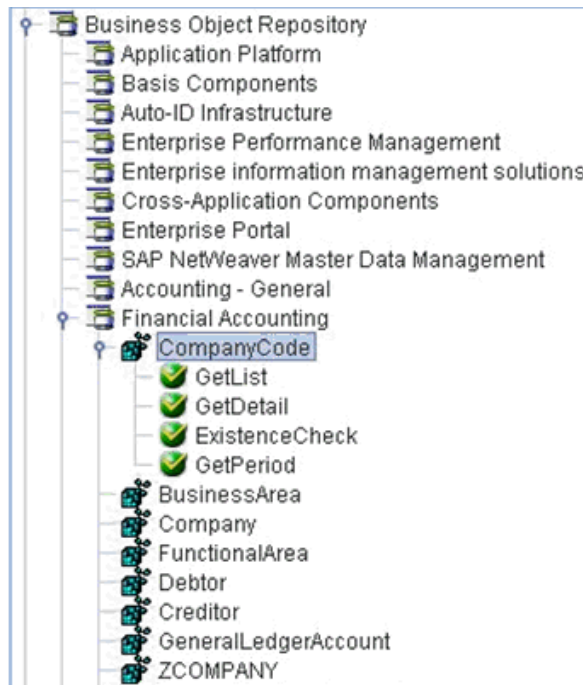
Perform the following steps to generate a WSDL for request/response service:

1. Start **Application Explorer** and connect to a defined MySAP target (a J2CA configuration).

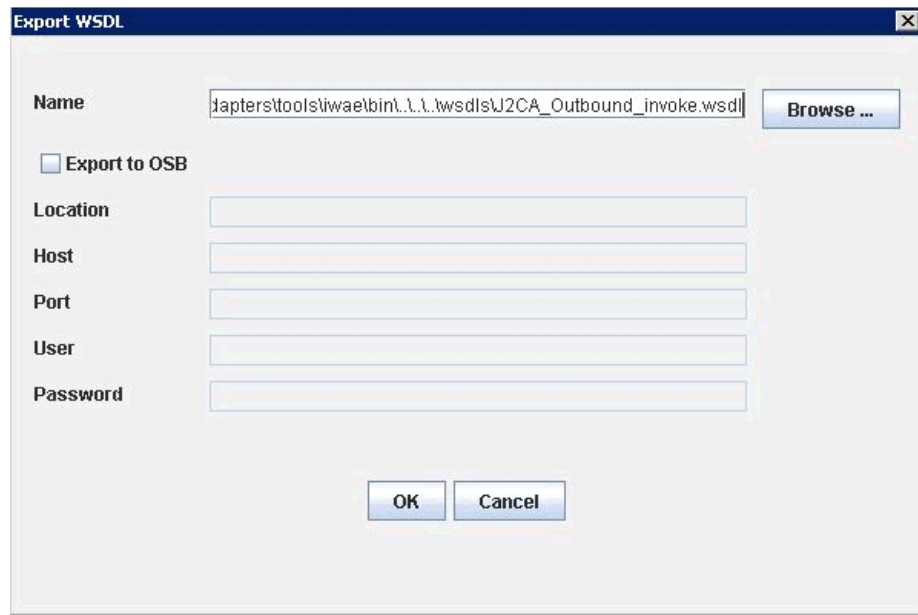
For more information on starting the Application Explorer and defining a target, and connecting to SAP R/3, see [Section 4.2, "Starting Application Explorer"](#) on page 4-4 and [Section 4.5.1, "Defining a Target to SAP R/3"](#) on page 4-8.

2. Expand the MySAP target to which you are connected.
3. Expand **Business Object Repository, Financial Accounting, Company Code Business Object**, and then select **GetDetail**, as shown in [Figure 6-9](#).

Figure 6-9 Connected MySAP_Target



4. Right-click the **GetDetail** node.
A menu is displayed.
5. Click **Create Outbound JCA Service (Request/Response)**.
The Export WSDL dialog is displayed, as shown in [Figure 6-10](#).

Figure 6–10 Export WSDL Dialog

6. Click **OK**.

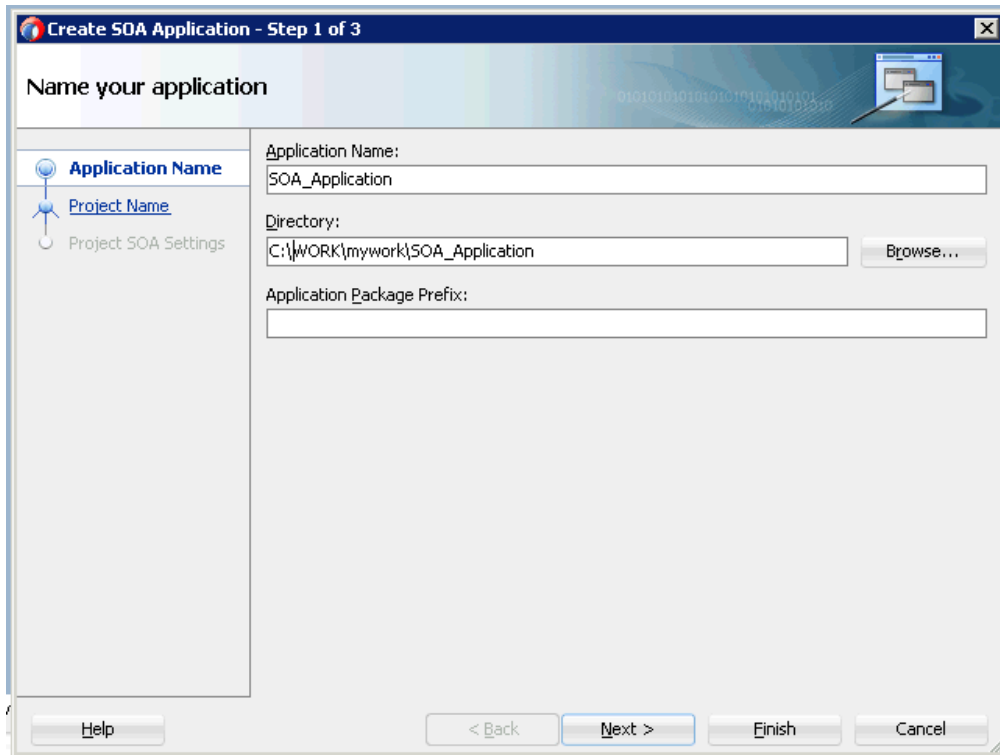
You can now create an empty composite for SOA, which is the first step that is required to define a BPEL outbound process in JDeveloper.

6.4.2 Creating an Empty Composite for SOA

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

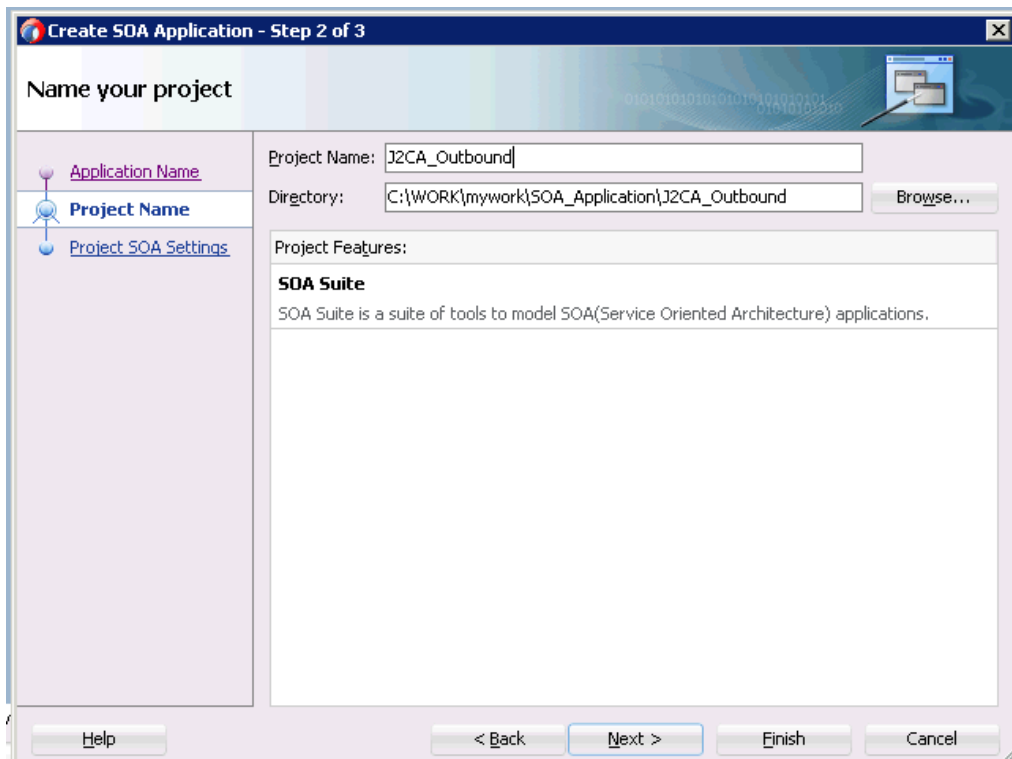
1. Create a new SOA application.
2. Enter a name for the new SOA Application and click **Next**, as shown in [Figure 6–11](#).

Figure 6–11 Name Your Application Page



The Name your project page is displayed, as shown in [Figure 6–12](#).

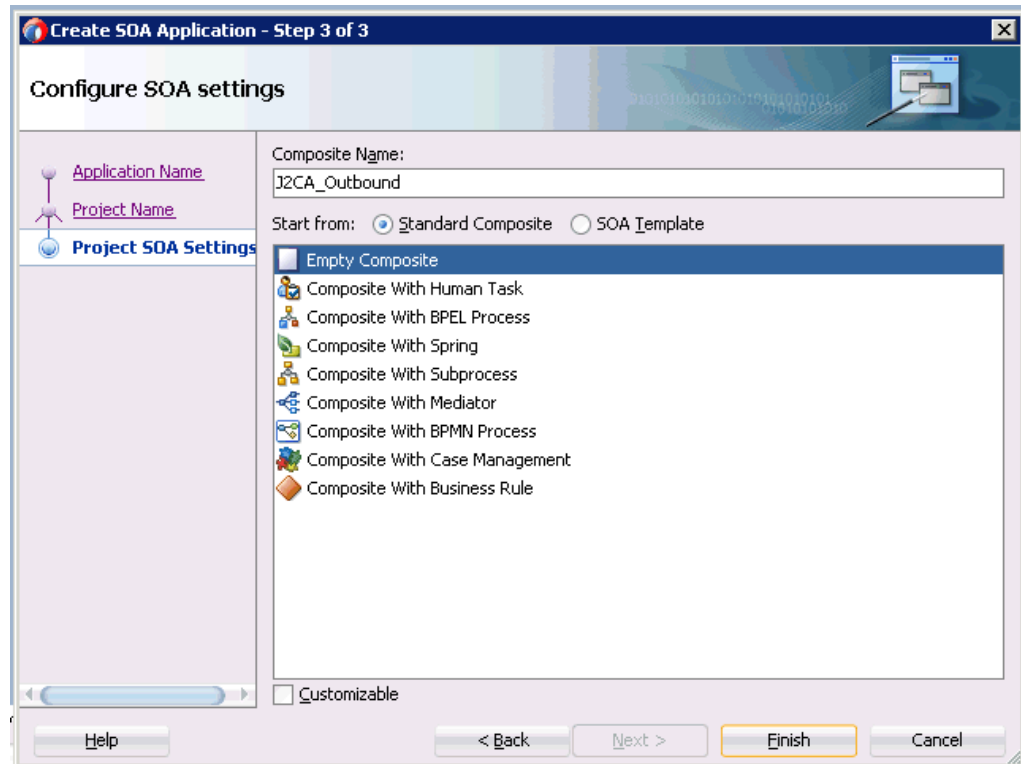
Figure 6–12 Name Your Project Page



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

The Configure SOA settings page is displayed, as shown in [Figure 6–13](#).

Figure 6–13 *Configure SOA Settings Page*



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

6.4.3 Defining a BPEL Outbound Process

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

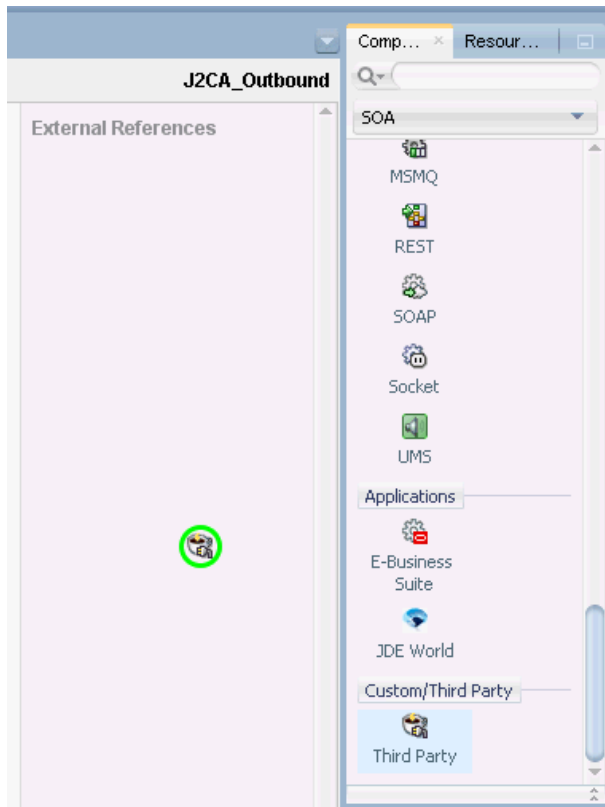
- [Section 6.4.3.1, "Configuring a Third Party Adapter Service Component"](#)
- [Section 6.4.3.2, "Configuring an Outbound BPEL Process Component"](#)
- [Section 6.4.3.3, "Adjusting for Known Deployment Issues With 12c"](#)

6.4.3.1 Configuring a Third Party Adapter Service Component

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

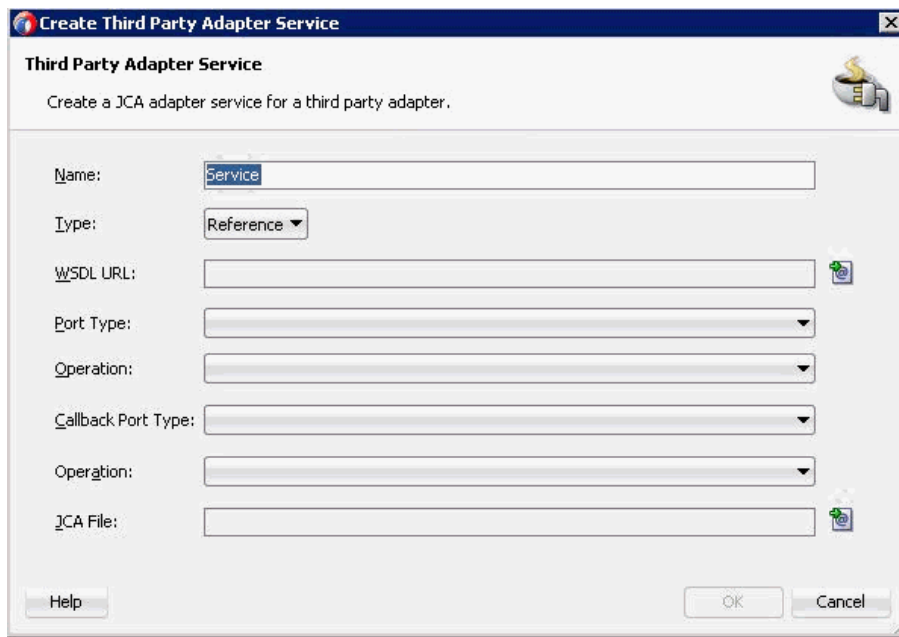
1. Drag and drop the **Third Party Adapter** component from the Service Adapters pane to the External References pane, as shown in [Figure 6–14](#).

Figure 6–14 Third Party Adapter Component



The Create Third Party Adapter Service dialog is displayed, as shown in [Figure 6–15](#).

Figure 6–15 Create Third Party Adapter Service Dialog

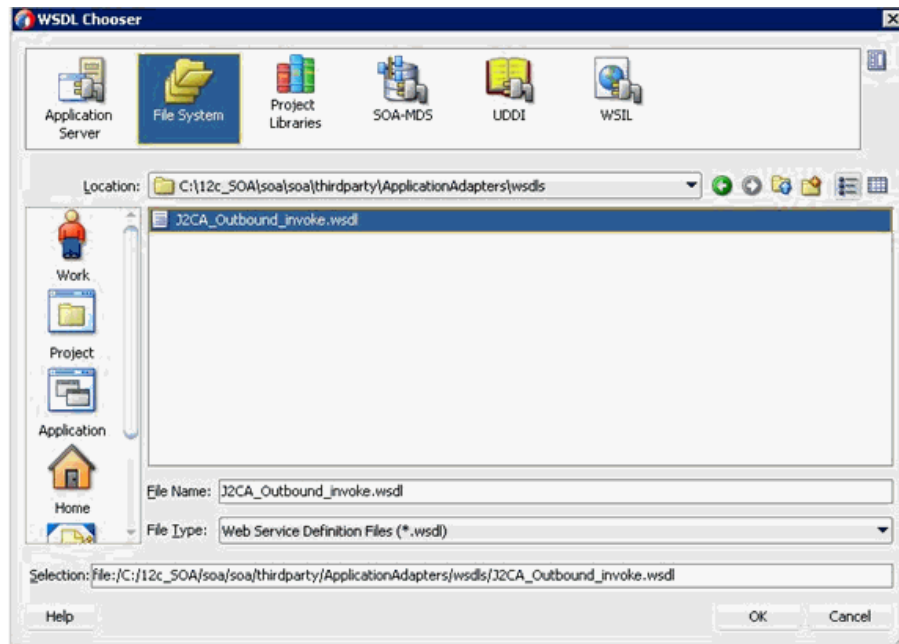


2. Enter a name for the third party adapter service, or leave the default name.

3. Ensure that **Reference** is selected from the Type 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, as shown in [Figure 6–16](#).

Figure 6–16 WSDL Chooser Dialog



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

`<ADAPTER_HOME>\wsdl`

6. Click **OK**.

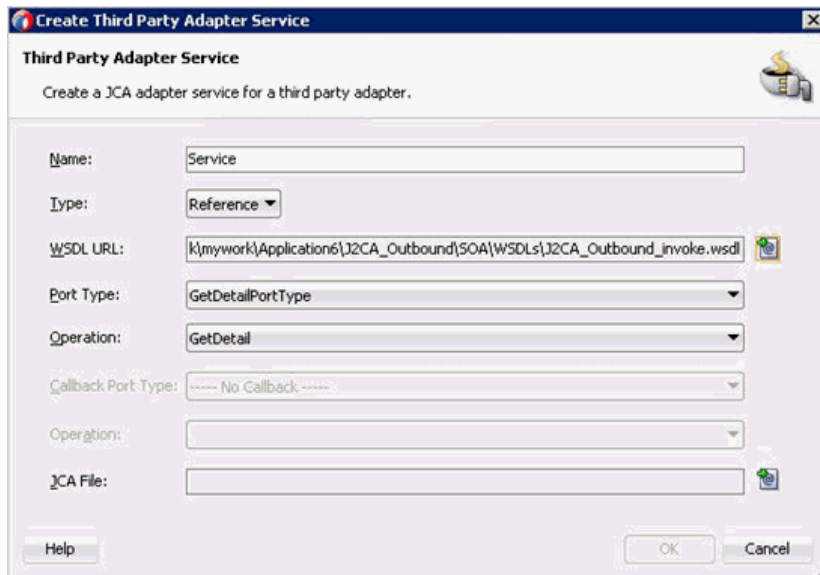
The Localize Files dialog is displayed.

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.

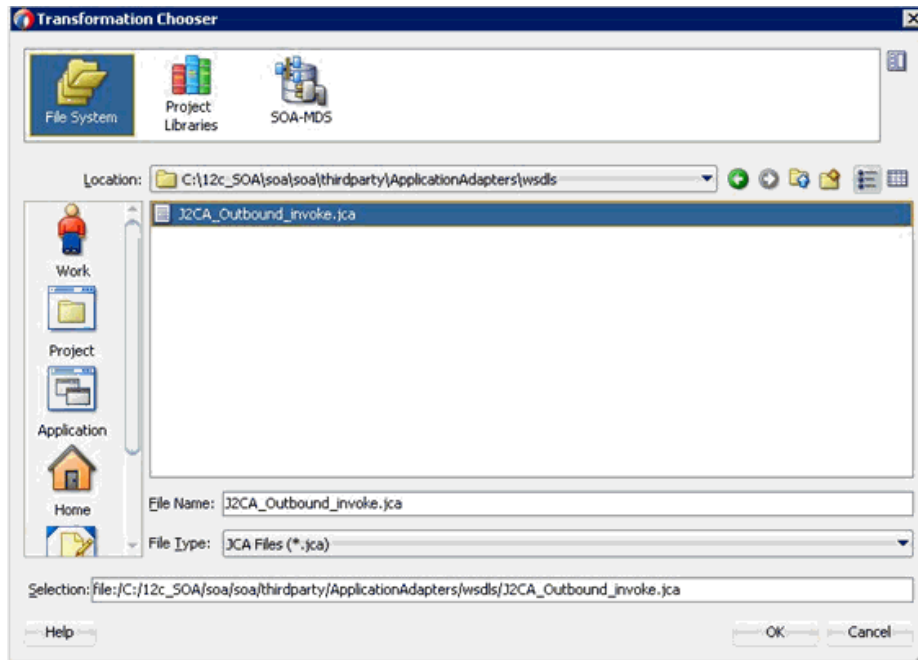
You are returned to the Create Third Party Adapter Service dialog, as shown in [Figure 6–17](#).

Figure 6–17 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, as shown in [Figure 6–18](#).

Figure 6–18 Transformation Chooser Dialog

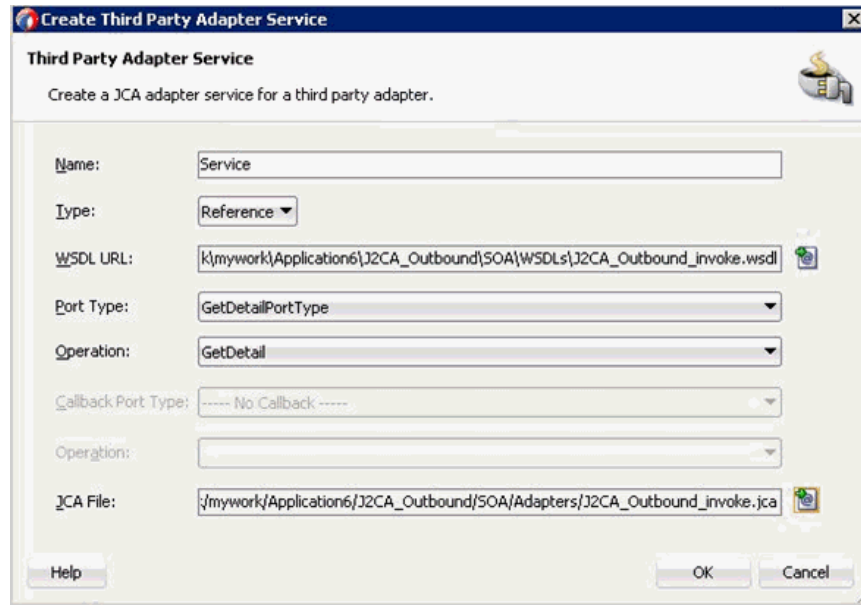


9. Browse and select the JCA properties file from the following directory:
`<ADAPTER_HOME>\wsdl\`
10. Click **OK**.
 The Copy File message is displayed.
11. Click **Yes**.

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

You are returned to the Create Third Party Adapter Service dialog, as shown in [Figure 6–19](#).

Figure 6–19 Create Third Party Adapter Service Dialog



12. 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.

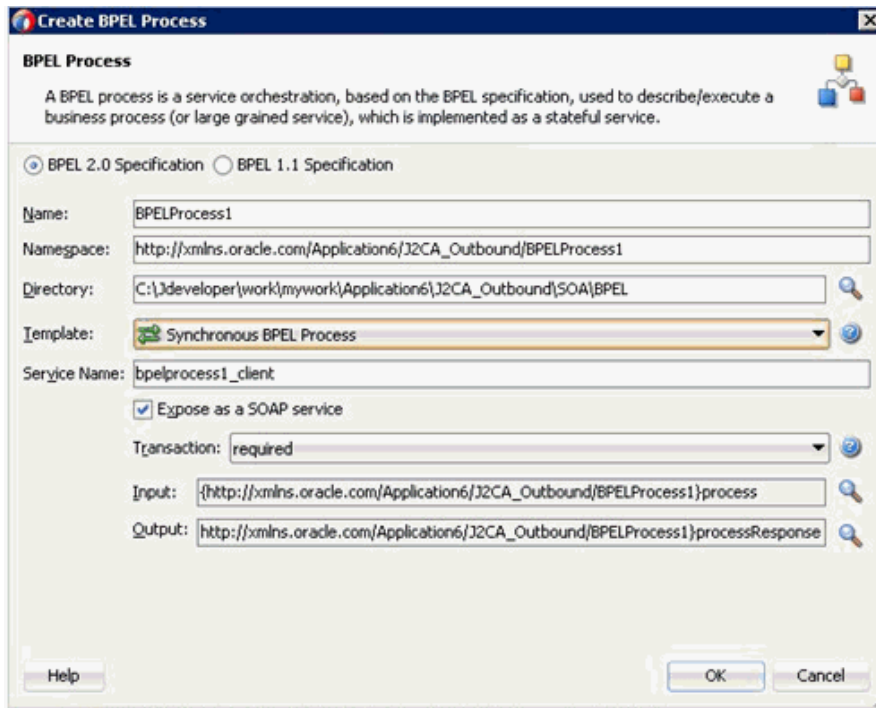
6.4.3.2 Configuring an Outbound BPEL Process Component

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.

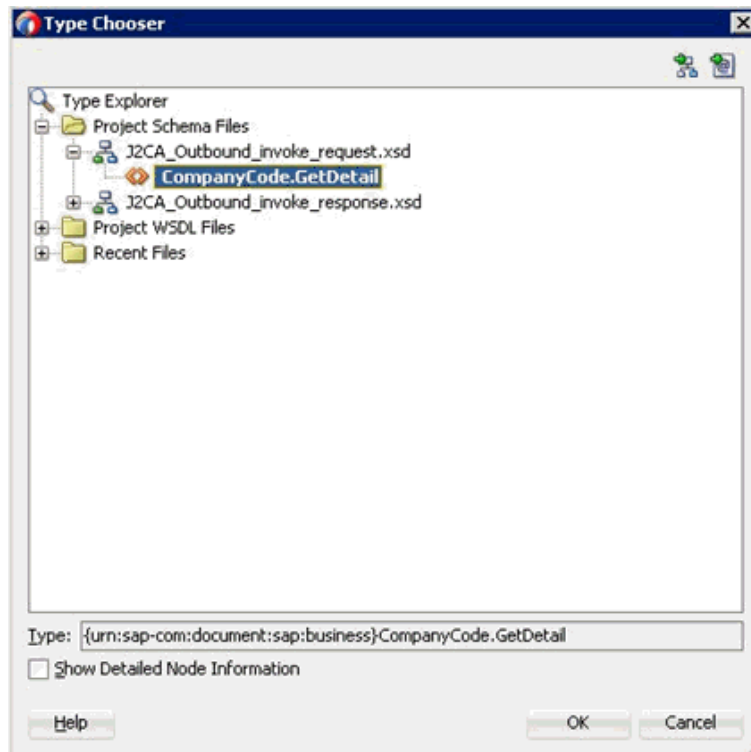
The Create BPEL Process dialog is displayed, as shown in [Figure 6–20](#).

Figure 6–20 Create BPEL Process Dialog



2. In the Name field, enter a name to identify the new outbound BPEL process component, or leave the default name.
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 6–21](#).

Figure 6–21 Type Chooser Dialog

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

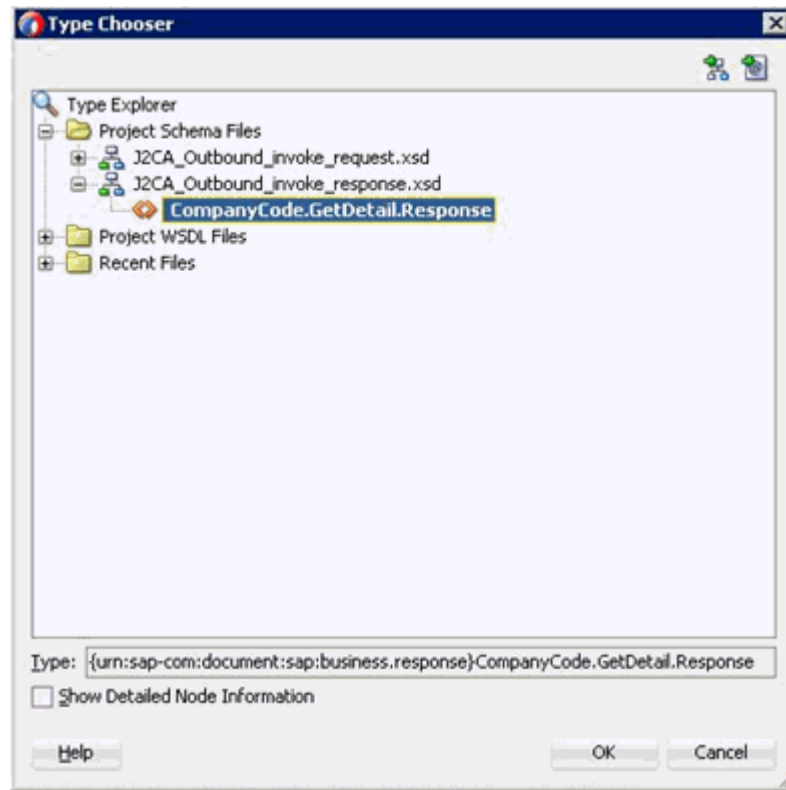
You are returned to the Create BPEL Process dialog, as shown in [Figure 6–22](#).

Figure 6–22 Create BPEL Process Dialog



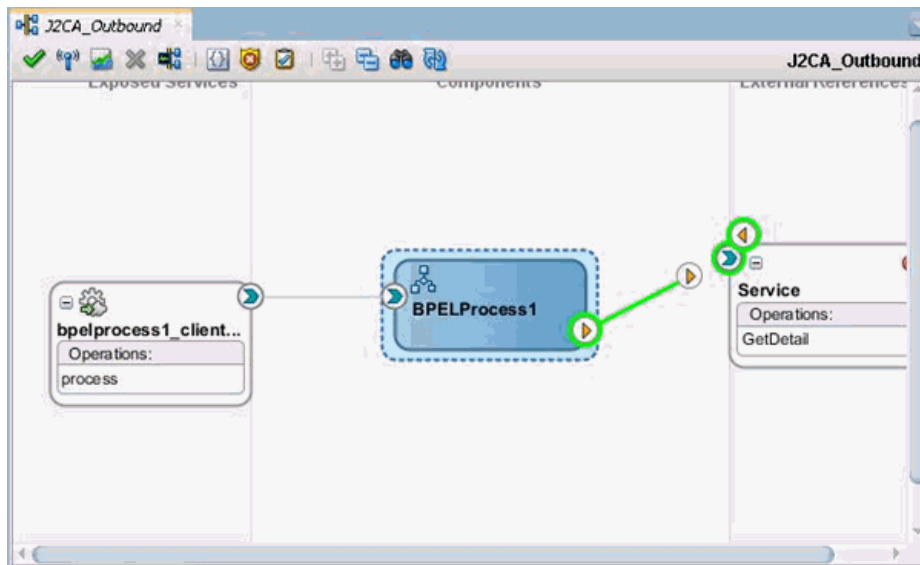
7. Click the **Browse** icon, which is located to the right of the Output field to select the associated XML response schema file.

The Type Chooser dialog is displayed, as shown in [Figure 6–23](#).

Figure 6–23 Type Chooser Dialog

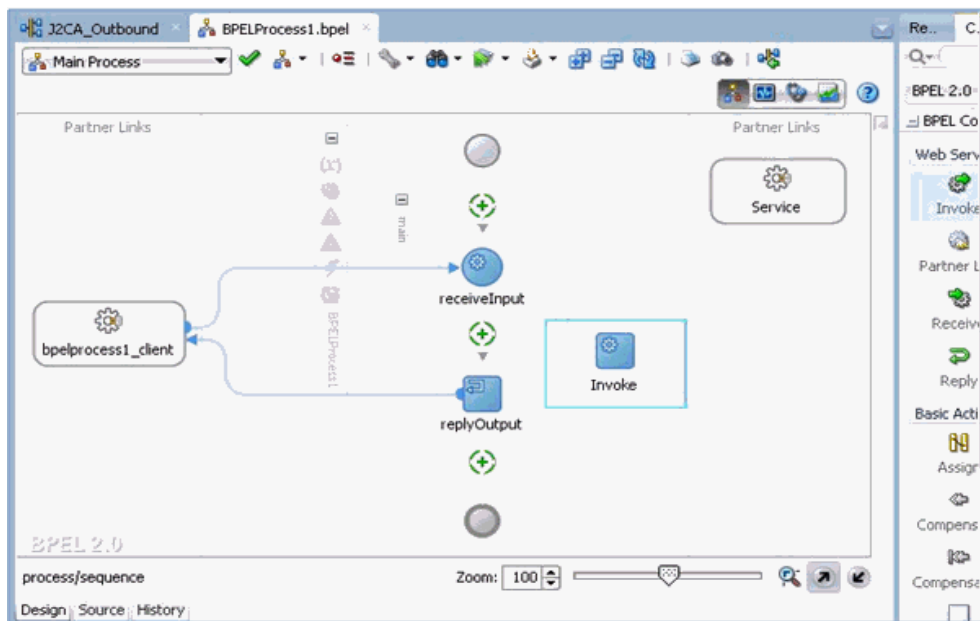
8. Expand **Project Schema Files**, **J2CA_Outbound_invoke_response.xsd**, and select **CompanyCode.GetDetailResponse**.
9. Click **OK**.
You are returned to the Create BPEL Process dialog.
10. Click **OK**.
11. Create a connection between the outbound BPEL process component and the third party adapter service component, as shown in [Figure 6–24](#).

Figure 6–24 Created Connection

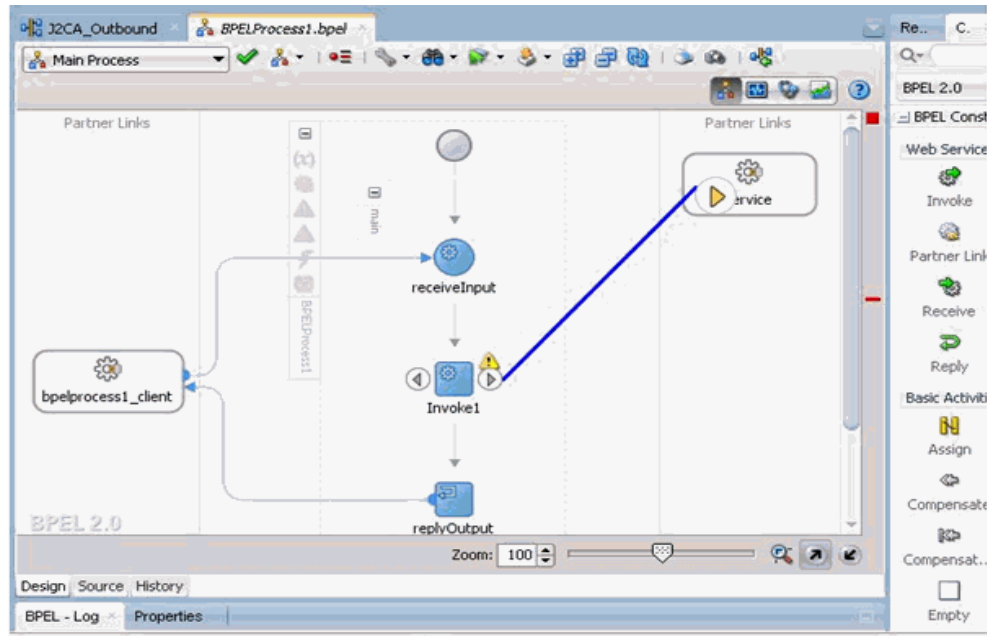


12. Double-click the outbound BPEL process component in the Components pane.
13. Drag and drop the **Invoke** activity component under BPEL Constructs - Web Service, to the Components pane and place it between the **receiveInput** activity component and the **replyOutput** activity component, as shown in Figure 6–25.

Figure 6–25 Invoke Activity Component



14. Create a connection between the new Invoke activity component (Invoke1) and the third party adapter service component, as shown in Figure 6–26.

Figure 6–26 Created Connection

The Edit Invoke dialog is displayed.

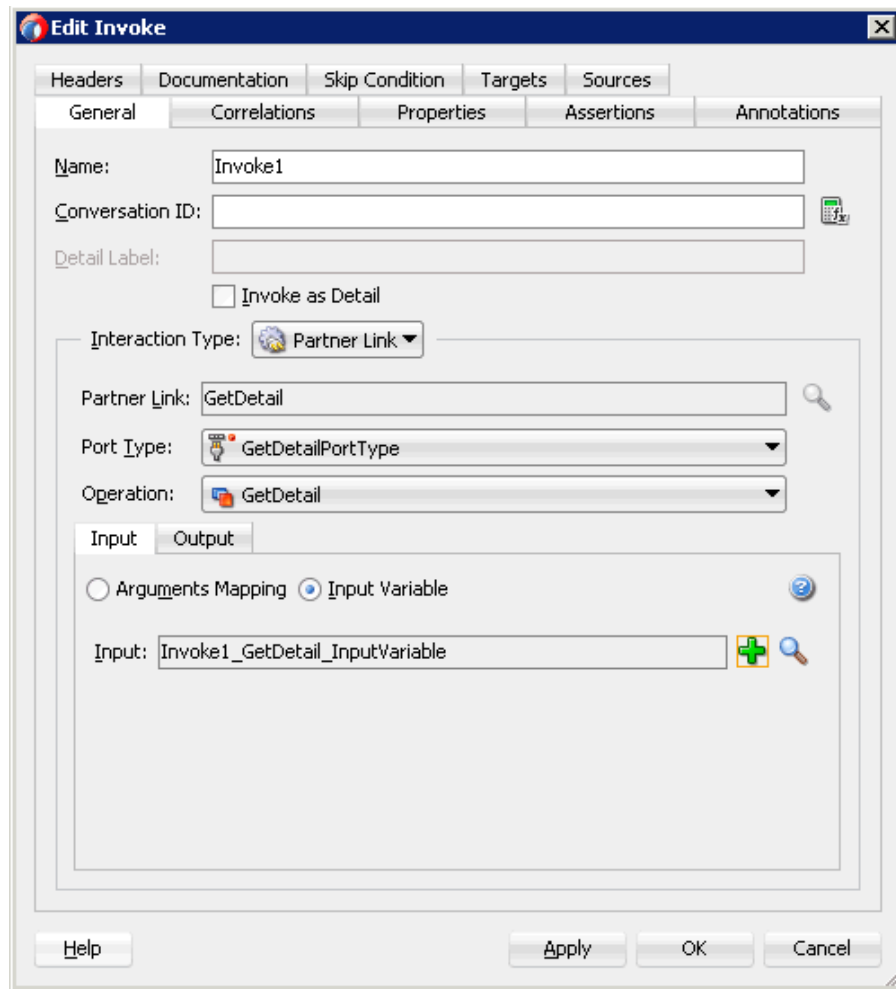
15. Click the **Plus sign** icon, which is located to the right of the Input field to configure a new input variable.

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 6–27](#).

Figure 6–27 Edit Invoke Dialog



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 6–28](#).

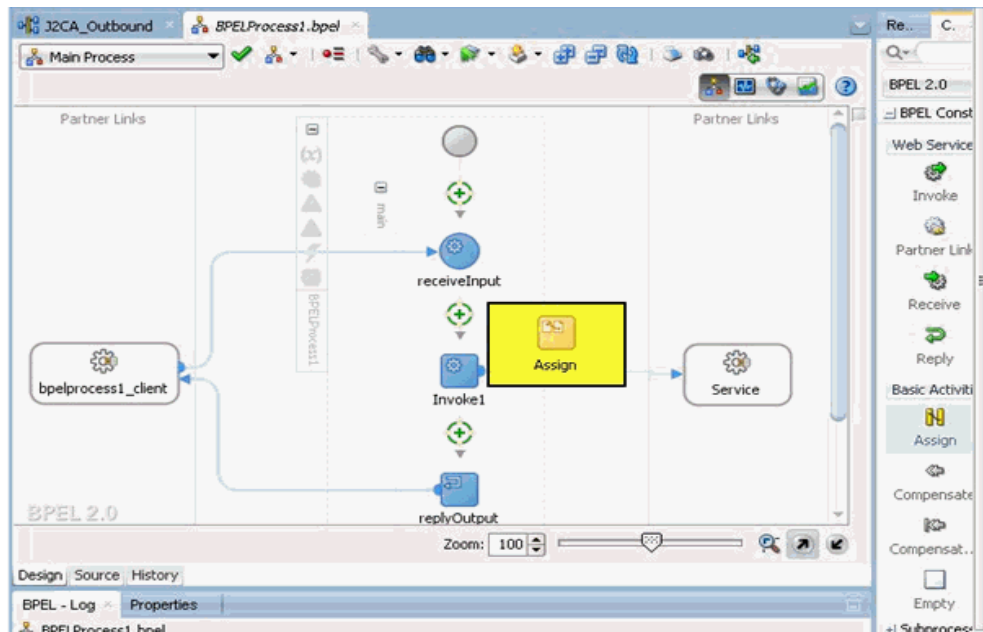
Figure 6–28 Edit Invoke Dialog

The screenshot shows the 'Edit Invoke' dialog box with the following configuration:

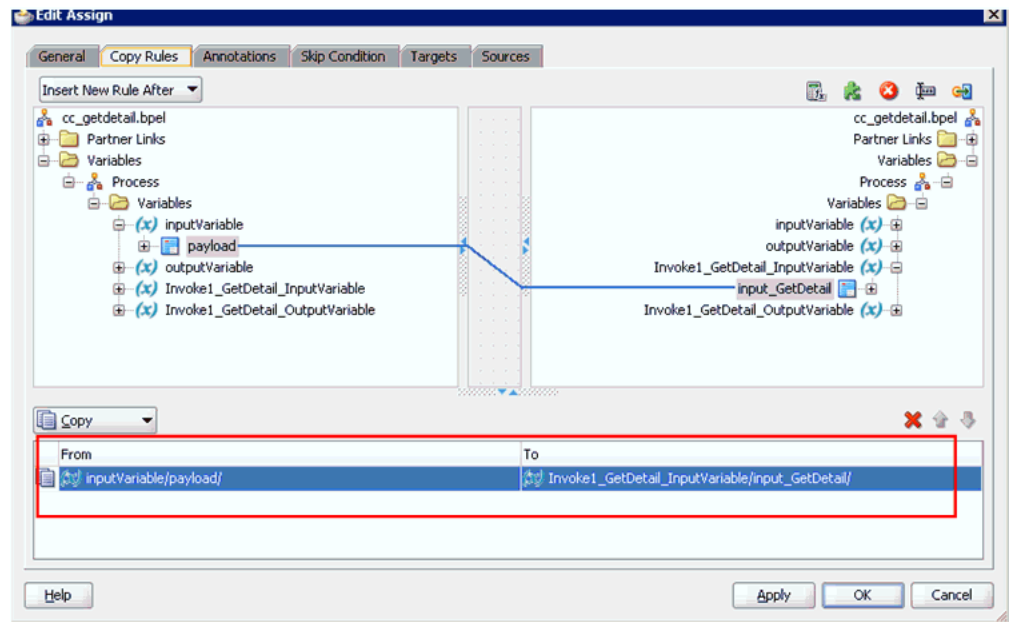
- Name:** Invoke1
- Conversation ID:** (empty)
- Detail Label:** (empty)
- Invoke as Detail:**
- Interaction Type:** Partner Link
- Partner Link:** GetDetail
- Port Type:** GetDetailPortType
- Operation:** GetDetail
- Output:** Invoke1_GetDetail_OutputVariable

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 6–29](#).

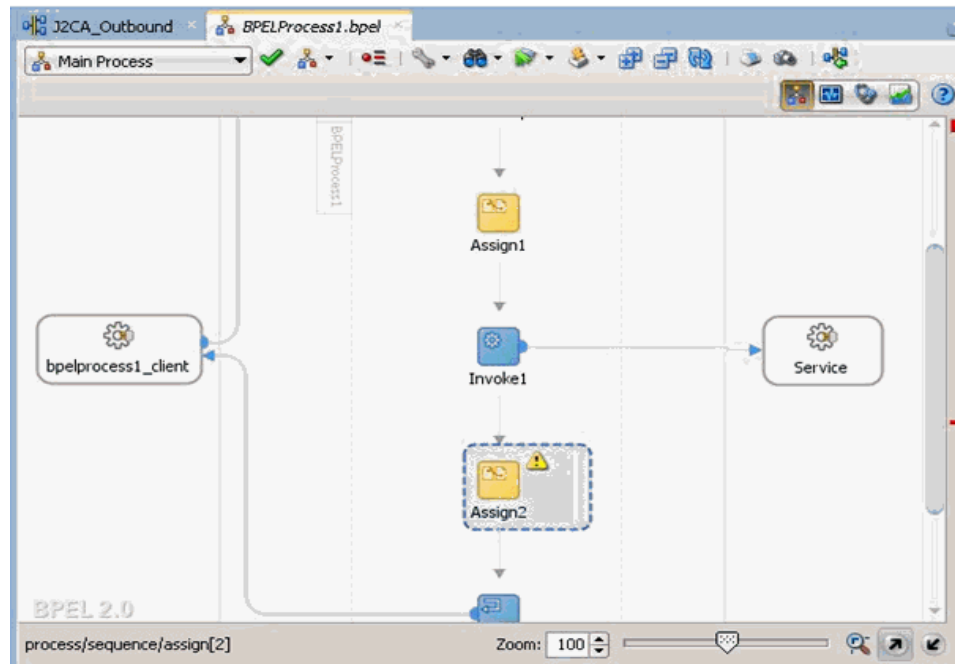
Figure 6–29 Assign Activity Component



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_GetDetail_InputVariable**, and then select **input_GetDetail**.
24. Drag and map the **payload** variable to the **input_GetDetail** variable.
The mapped variables are populated in the highlighted area as shown in [Figure 6–30](#).

Figure 6–30 Edit Assign Dialog

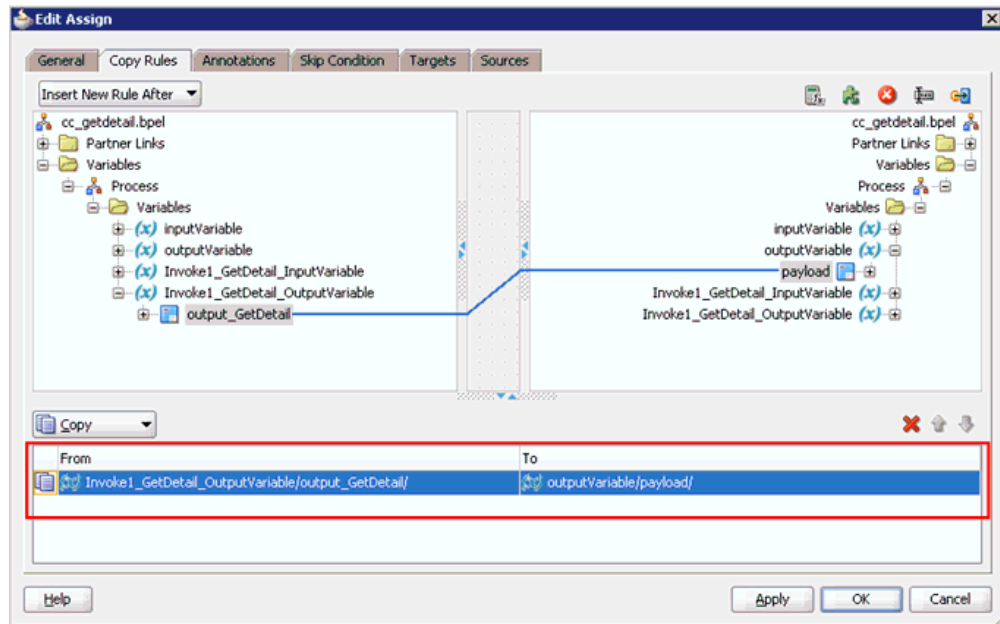
25. Click **Apply** and then **OK**.
26. Drag and drop the **Assign** activity component to the Components pane and place it between the Invoke activity (Invoke1) and the Reply activity (replyOutput).
27. Double-click the new Assign activity component (**Assign2**), as shown in [Figure 6–31](#).

Figure 6–31 New Assign Activity Component

The Edit Assign dialog is displayed.

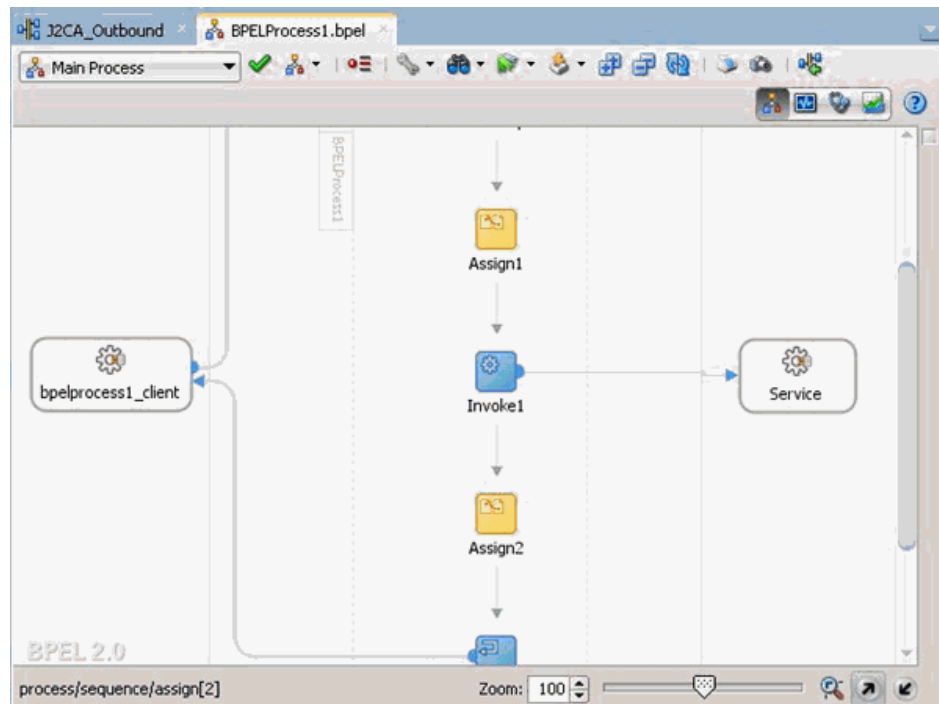
28. In the left pane, under Variables, expand **Invoke1_GetDetail_OutputVariable**, and then select **output_GetDetail**.
29. In the right pane, under Variables, expand **outputVariable** and select **payload**.
30. Drag and map the **output_GetDetail** variable to the **payload** variable.
The mapped variables are populated in the highlighted area as shown in [Figure 6–32](#).

Figure 6–32 Edit Assign Dialog



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

You are returned to the Activity component pane, as shown in [Figure 6–33](#).

Figure 6–33 Activity Component Pane

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

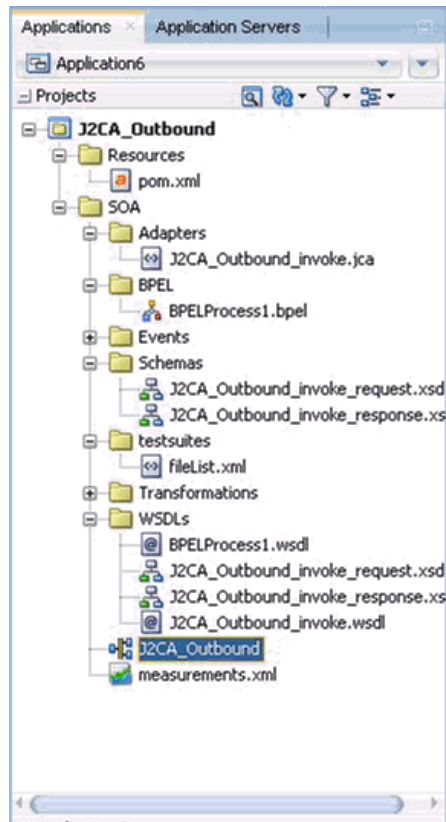
You are now ready to deploy the BPEL outbound process.

6.4.3.3 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 6–34](#).

Figure 6–34 J2CA_Outbound Node



2. Click the **Source** tab below the opened process, as shown in [Figure 6–35](#).

Figure 6–35 Source Tab

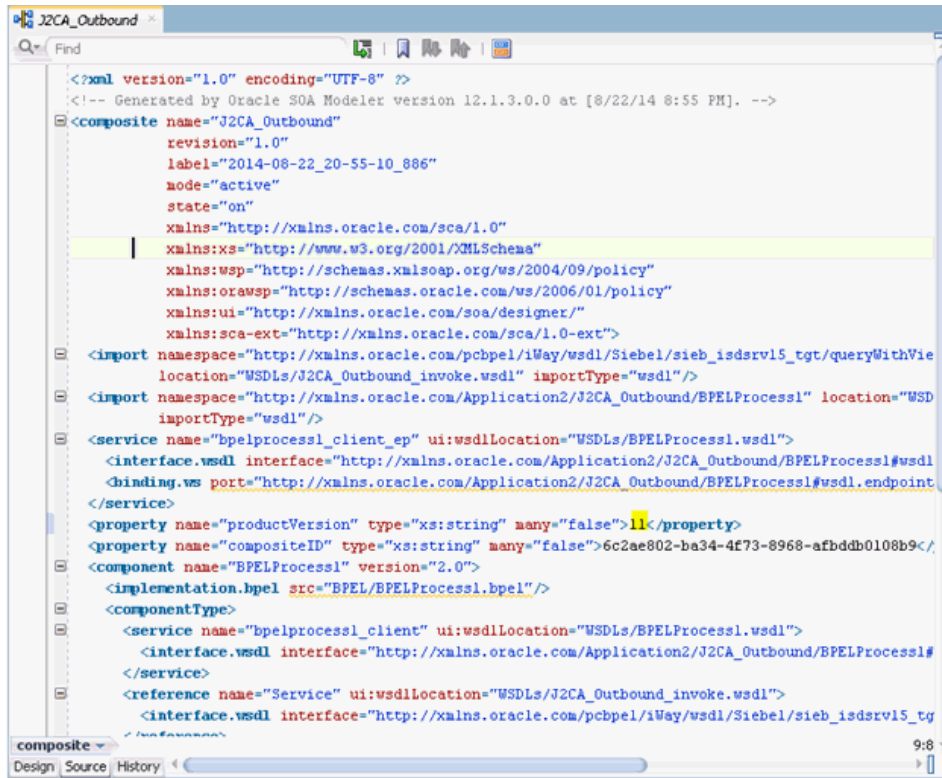
```

<?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/wsd1/Siebel/sieb_isdsrv15_tgt/queryWithVie
    location="WSDLs/J2CA_Outbound_invoke.wsd1" importType="wsdl"/>
  <import namespace="http://xmlns.oracle.com/Application2/J2CA_Outbound/BPELProcess1" location="WSD
    importType="wsdl"/>
  <service name="bpelprocess1_client_ep" ui:wsdLocation="WSDLs/BPELProcess1.wsd1">
    <interface.wsd1 interface="http://xmlns.oracle.com/Application2/J2CA_Outbound/BPELProcess1#wsd1
    <binding.ws port="http://xmlns.oracle.com/Application2/J2CA_Outbound/BPELProcess1#wsd1.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:wsdLocation="WSDLs/BPELProcess1.wsd1">
        <interface.wsd1 interface="http://xmlns.oracle.com/Application2/J2CA_Outbound/BPELProcess1#
        </service>
        <reference name="Service" ui:wsdLocation="WSDLs/J2CA_Outbound_invoke.wsd1">
          <interface.wsd1 interface="http://xmlns.oracle.com/pcbpel/iWay/wsd1/Siebel/sieb_isdsrv15_tg
    </componentType>
  </component>
</composite>

```

3. Change the productVersion property value from 12.1.3.0.0 to 11, as shown in Figure 6–36.

Figure 6–36 Property Value



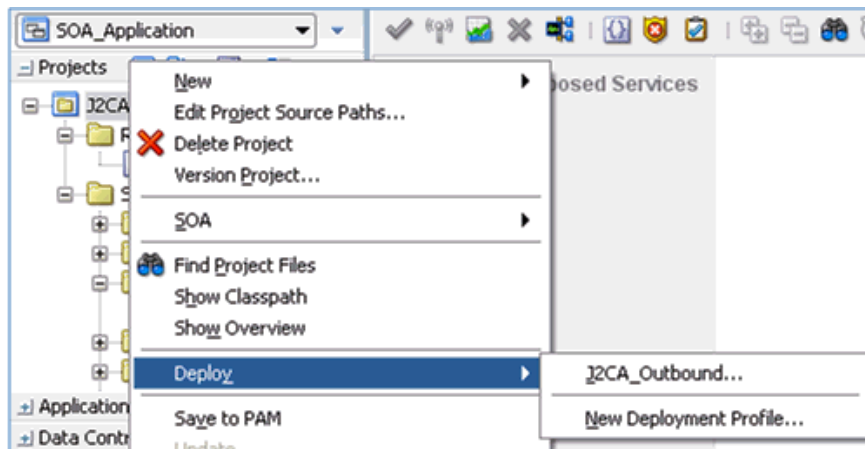
4. Save the changes and proceed to deploy the project.

6.4.4 Deploying the BPEL Outbound Process

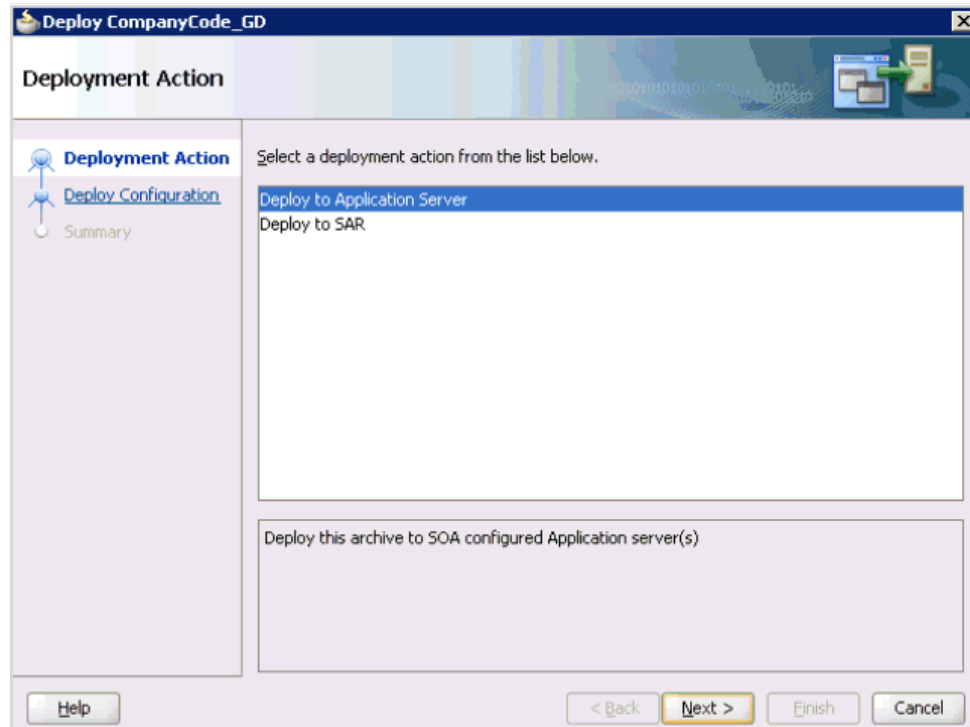
Perform the following steps to deploy the BPEL outbound process.

1. Right-click the project name in the left pane, select **Deploy**, and then click **J2CA_Outbound**, as shown in Figure 6–37.

Figure 6–37 J2CA_Outbound Option



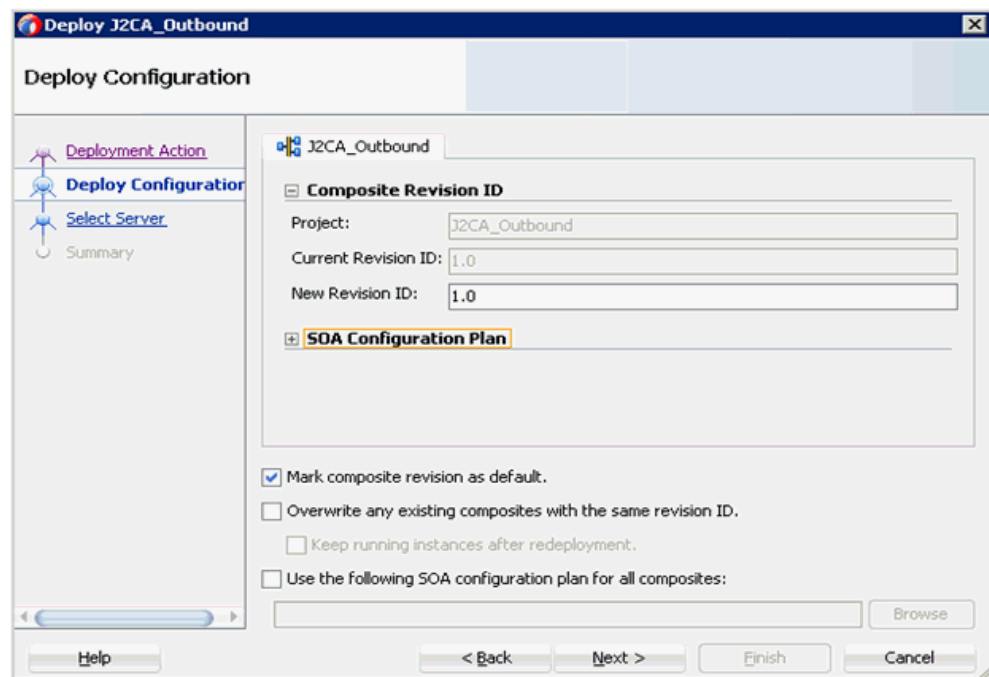
The Deployment Action page is displayed, as shown in Figure 6–38.

Figure 6–38 Deployment Action Page

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

3. Click **Next**.

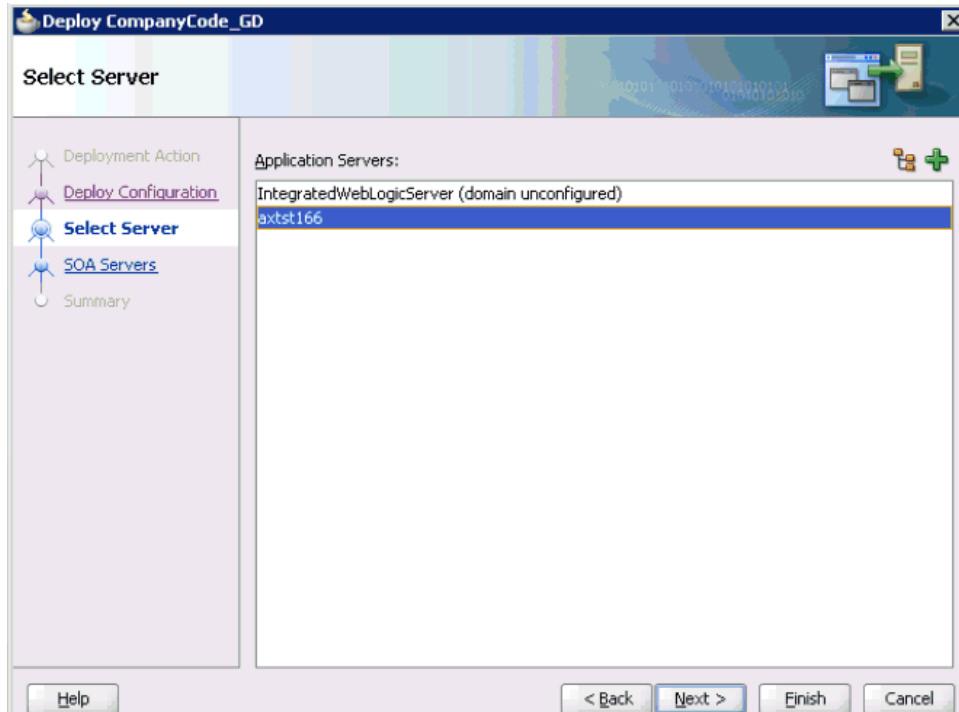
The **Deploy Configuration** page is displayed, as shown in [Figure 6–39](#).

Figure 6–39 Deploy Configurations Page

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

The Select Server page is displayed, as shown in Figure 6-40.

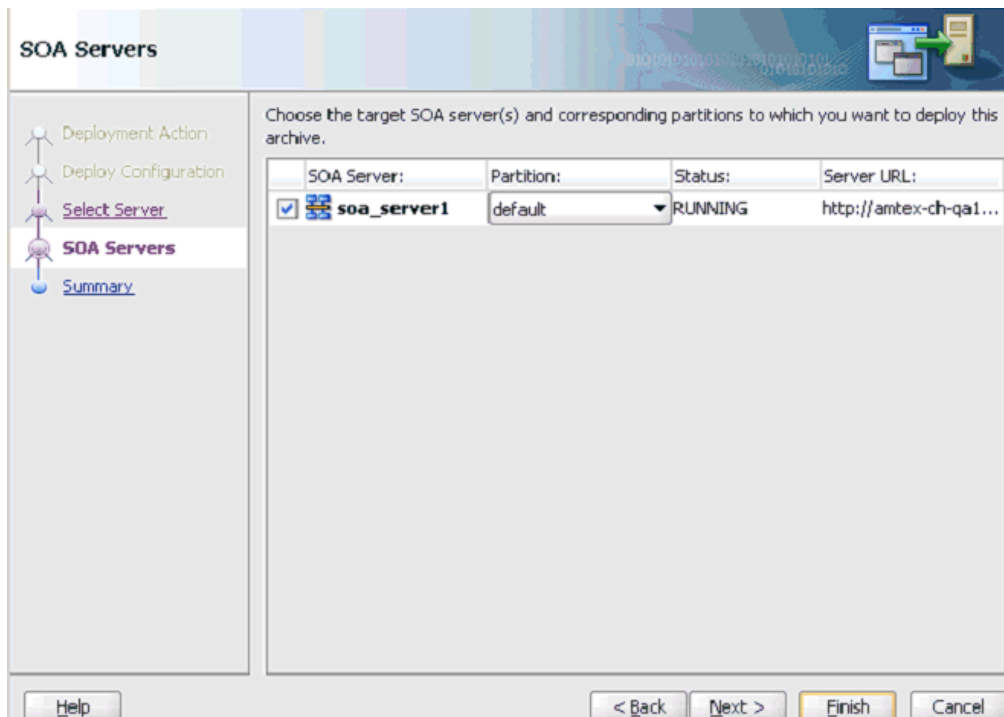
Figure 6-40 Select Server Page



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

The SOA Servers page is displayed, as shown in Figure 6-41.

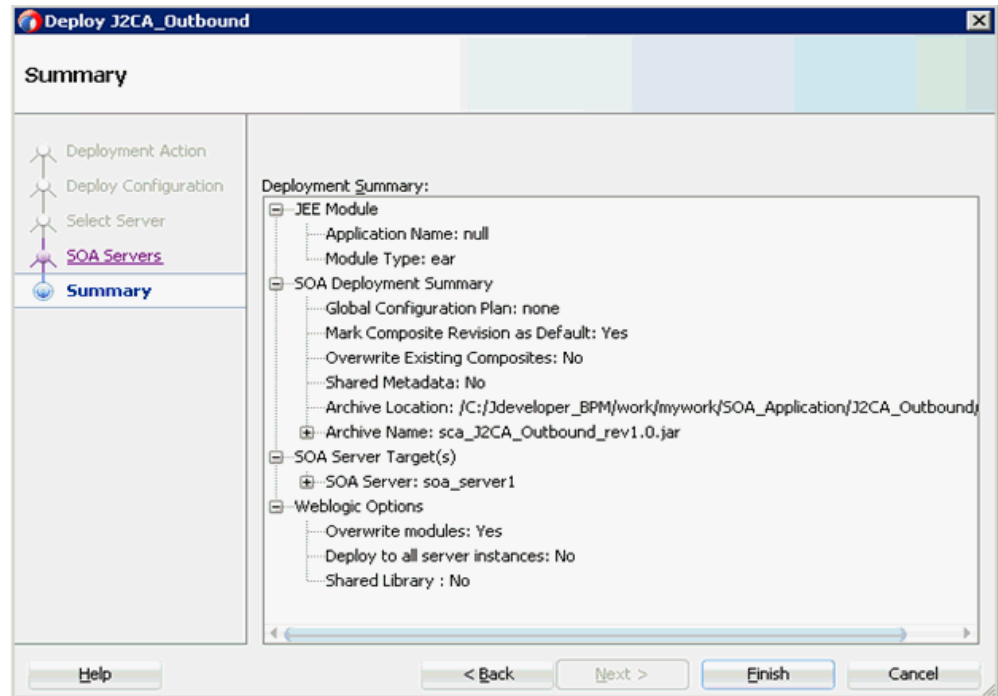
Figure 6-41 SOA Servers Page



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

The Summary page is displayed, as shown in [Figure 6–42](#).

Figure 6–42 Summary Page



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

The process is deployed successfully, as shown in [Figure 6–43](#).

Figure 6–43 Successful Deployment Message



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.

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

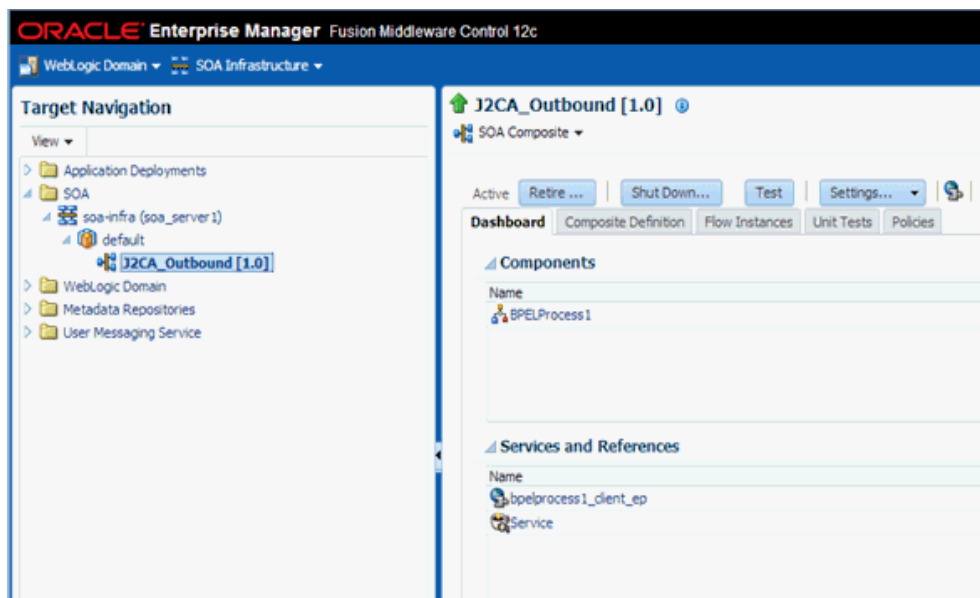
2. Click **Target Navigation** in the left pane, expand **SOA**, and then select **soa-infra (soa_server1)**.

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.

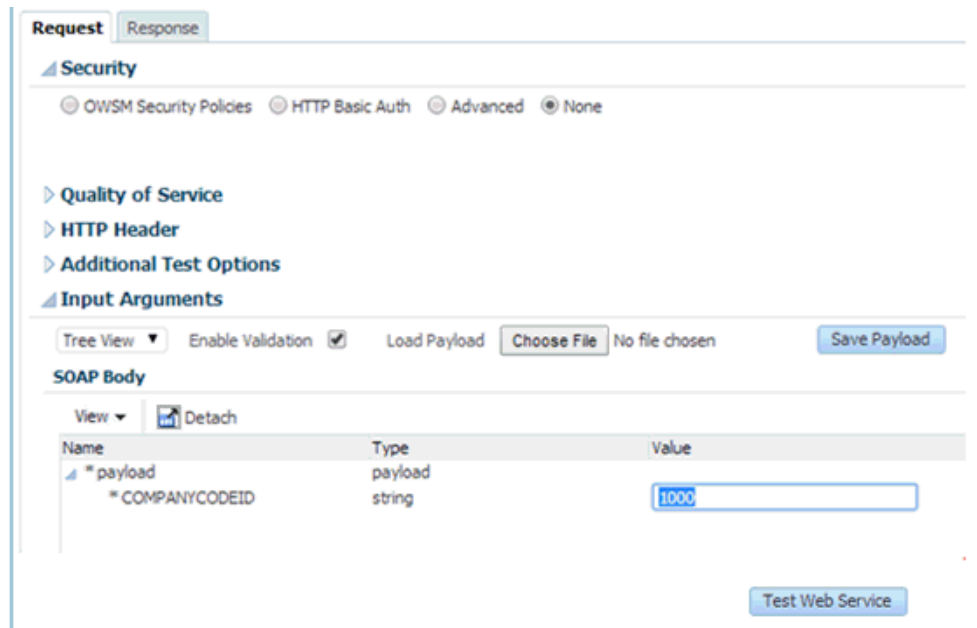
2. Expand **SOA**, select **soa-infra (soa_server1)**, and then click **Default**.
3. Select an available project (for example, J2CA_Outbound).

Figure 6–44 Test Button



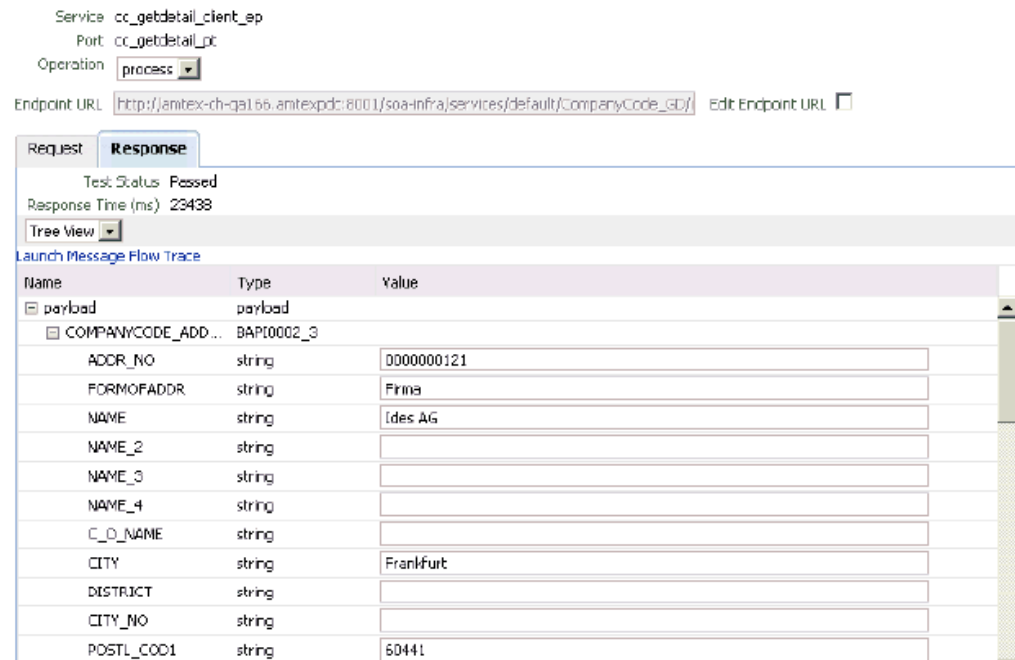
4. Click **Test**, as shown in [Figure 6–44](#).

Figure 6–45 Request Tab



5. Click the **Request** tab, as shown in [Figure 6–45](#).
6. Provide an appropriate input value in the Value field and click **Test Web Service**.
The output response is received in the Oracle Enterprise Manager console, as shown in [Figure 6–46](#).

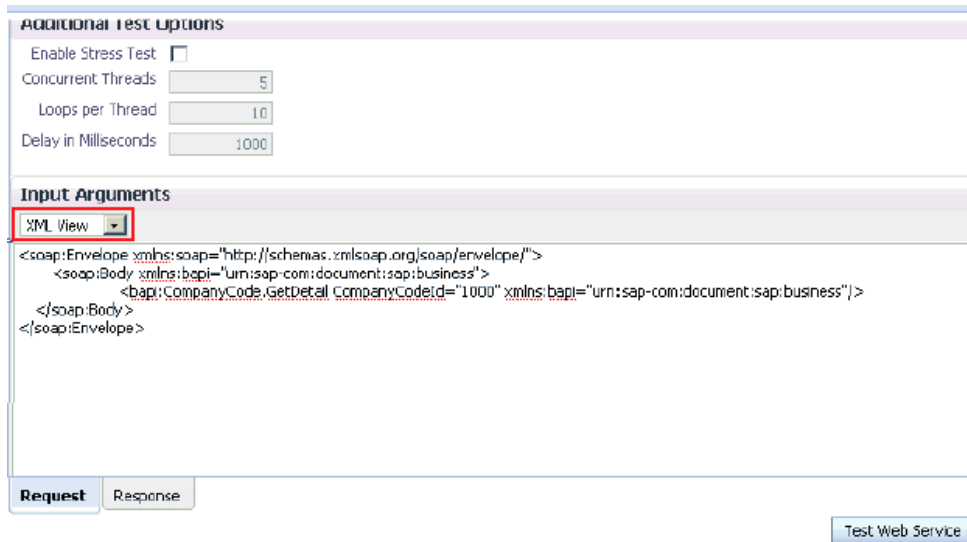
Figure 6–46 Output Response



Perform the following steps to invoke the input XML document using XML View:

- a. Select **XML View** from the list, as shown in [Figure 6–47](#).

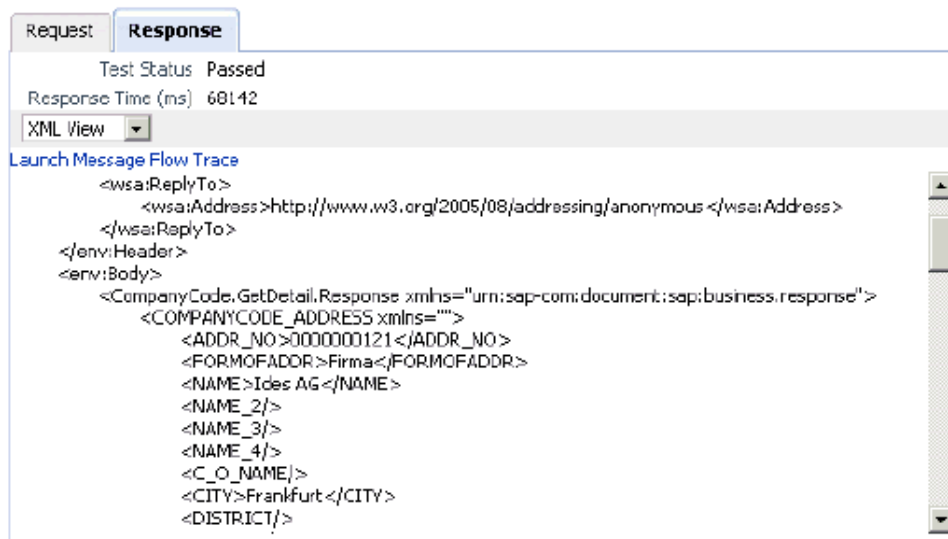
Figure 6–47 Input Arguments List



- b. 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 6–48](#).

Figure 6–48 Received Output Response



6.4.6 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 7, "Integration With Mediator Service Components in the Oracle SOA Suite"](#).

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

This section demonstrates how Oracle Application Adapter for SAP R/3 integrates with SAP R/3 to receive event data. In this example, an SAP R/3 event occurs when a customer record is added to a SAP R/3 system.

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\SAP_Samples.zip\SAP_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 JDeveloper.

This section contains the following topics:

- [Section 6.5.1, "Generating WSDL for Event Integration"](#)
- [Section 6.5.2, "Creating an Empty Composite for SOA"](#)
- [Section 6.5.3, "Defining a BPEL Inbound Process"](#)
- [Section 6.5.4, "Deploying the BPEL Inbound Process"](#)
- [Section 6.5.5, "Triggering an Event in SAP R/3"](#)

6.5.1 Generating WSDL for Event Integration

Before you design a BPEL process, you must generate the respective WSDL file using Application Explorer.

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 6.5.1.1, "Creating a Channel"](#)
- [Section 6.5.1.2, "Generating WSDL for Event notification"](#)

6.5.1.1 Creating a Channel

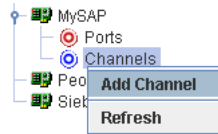
To create a channel:

1. Start **Application Explorer** and connect to a J2CA configuration.
2. In the left pane, expand the **Events** node.

3. Expand the **MySAP** node.

The Ports and Channels nodes are displayed in the left pane, as shown in [Figure 6–49](#).

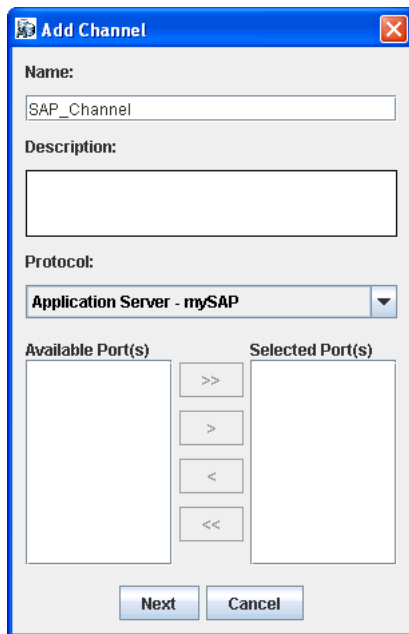
Figure 6–49 MySAP Channel Node



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

The Add Channel dialog is displayed, as shown in [Figure 6–50](#).

Figure 6–50 Add Channel Dialog



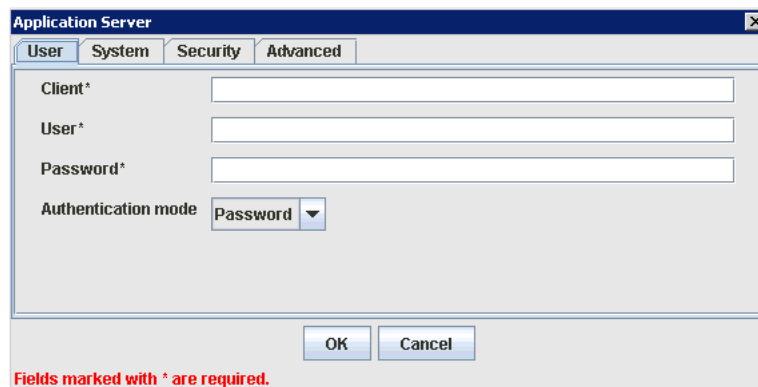
Perform the following steps:

- a. Enter a name for the channel, for example, `SAP_Channel1`.
 - b. Enter a brief description (optional).
 - c. From the **Protocol** list, select **Application Server - mySAP**.
5. Click **Next**.

The Application Server dialog is displayed, as shown in [Figure 6–51](#). The following tabs are available:

- User (Required)
- System (Required)
- Security
- Advanced

Figure 6–51 Application Server Dialog



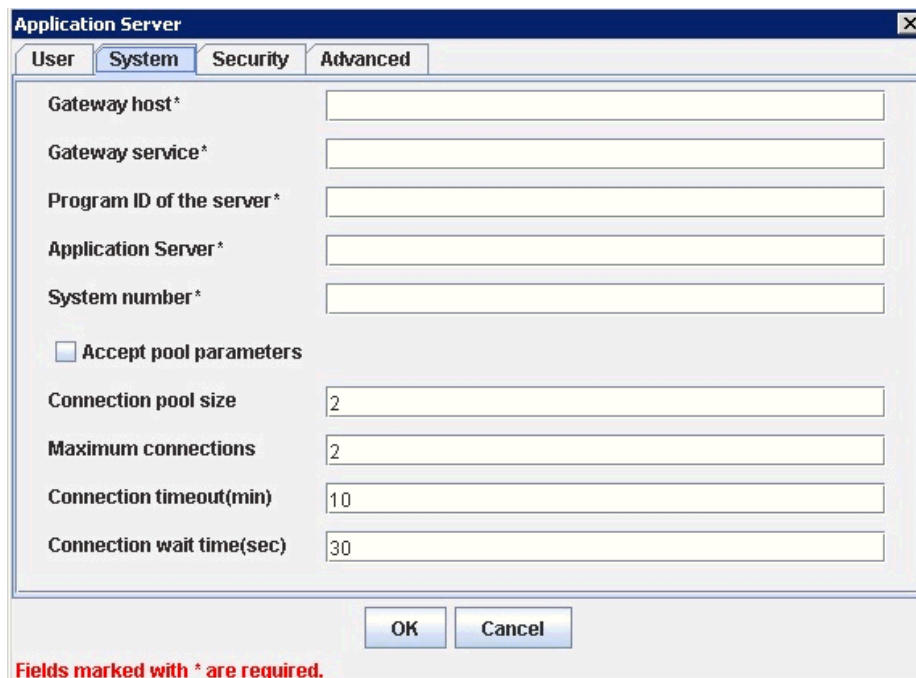
- For the **User** tab, enter the appropriate information for your SAP R/3 channel based on the information in the following table.

Table 6–1 User Tab Parameters

Target Parameter	Description
Client	The client number defined for the SAP R/3 application for client communications.
User	A valid user ID for the SAP R/3 application.
Password	A valid password for the SAP R/3 application.
Authentication Mode	The authentication mode you want to use when connecting to your SAP R/3 system. By default, Password is selected from the list.

- Click the **System** tab, as shown in [Figure 6–52](#).

Figure 6–52 System Tab



8. For the **System** tab, enter the appropriate information for your SAP R/3 channel based on the information in the following table.

Table 6–2 System Tab Parameters

Target Parameter	Description
Gateway host	A host name for the SAP R/3 Gateway.
Gateway service	A service for the SAP R/3 Gateway.
Program ID of the server	An SAP R/3 program ID you want to use for this channel.
Application Server	The name of the SAP R/3 application server you are using.
System Number	An SAP R/3 system number.
Accept pool parameters (True False)	If False , use the default adapter connection tuning. If True , use the values provided by the user
Connection pool size	Maximum number of connections in a pool. If this is less than "Maximum Connections," the Pool Size will be set by the JCo to the "Maximum Connections" size.
Maximum connections	Maximum number of connections created at one time.
Connection timeout (min.)	Time in minutes until the unused connections are released.
Connection wait time (sec.)	Time in seconds to wait for a free connection handle when all connections are busy, before throwing a Resource exception.

9. For the **Security** tab (optional), enter the appropriate information for your SAP R/3 channel based on the information in the following table.

Table 6–3 Security Tab Parameters

Target Parameter	Description
SNC mode	By default, SNC is disabled. To enable SNC, select 1 from the list.
SNC partner	Enter the name of the RFC server or message server (load balancing) that provides the SNC services.
SNC level	From the list select the version of the SNC library.
SNC name	Enter the name of the SNC library you are using.
SNC library path	Enter the path to the SNC library.

10. For the **Advanced** tab (optional), enter the appropriate information for your SAP R/3 channel based on the information in the following table.

Table 6–4 Advanced Tab Parameters

Target Parameter	Description
IDOC Confirm	<p>If this option is selected, a STATUS IDOC is returned to SAP confirming receipt for every IDOC that is received.</p> <p>When a standard IDOC is sent, the usual last status available through the SAP IDOC monitor is "Data Sent to Port OK" - "03", in the WE02 transaction IDOC list. If you want to use the WE02 transaction to ensure that the IDOC was received at the target destination, then enable this option. When an IDOC is received, the adapter returns a STATUS IDOC to SAP and in the WE02 transaction IDOC list, the status of the IDOC is changed to "12", "Dispatch OK".</p> <p>Note that if a large number of IDOCs are sent in bursts, enabling this option may impact performance. If performance is an issue, then it is recommended to disable this option.</p>
IDOC Format	<p>Select an IDOC type from the list:</p> <ul style="list-style-type: none"> ■ XML (default) ■ XML-CDATA-ENVELOPED
IDOC release	The IDOC version you want to use for your connection.
IDOC release provider	The IDOC release provider for your connection. Select IDOC DOCREL field (default), SAP release , or user input from the list.
SAP trace	<p>Select this option to enable SAP traces.</p> <p>The SAP traces are stored in the following locations during design time and run time:</p> <p>Design Time:</p> <p><ADAPTER_HOME>\tools\iwae\bin</p> <p>Run Time:</p> <p><ORACLE_HOME>\user_projects\domains\base_domain</p> <p>The file name formats that are used for the SAP traces are as follows (for design time and run time):</p> <ul style="list-style-type: none"> ■ rfc02664_04332.trc ■ JCO100226_043846994.trc
Processing Mode	Select the type of synchronous processing from the list. Possible values include REQUEST and REQUEST_RESPONSE .

11. Click OK.

The channel is displayed under the channels node in the left pane. An X over the icon indicates that the channel is currently disconnected.

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 all the channels and close the Application Explorer before run-time deployment for events.

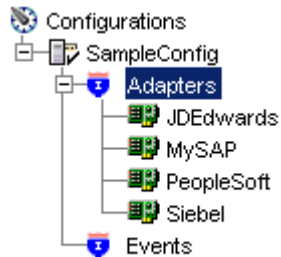
6.5.1.2 Generating WSDL for Event notification

After you create a channel and verify that it is not started, you must generate WSDL for the event using Application Explorer.

1. Start Application Explorer.
2. Expand the **Adapters** node.

A list of all adapters is displayed, as shown in [Figure 6–53](#).

Figure 6–53 Adapters Node

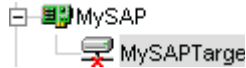


Perform the following steps:

- a. Expand the **MySAP** node.

A list of your available targets is displayed, as shown in [Figure 6–54](#).

Figure 6–54 Disconnected MySAP Target Node



- b. Click a target name under the **MySAP** node, for example, **SAPTTarget**.

The Connection dialog displays the saved parameters.

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

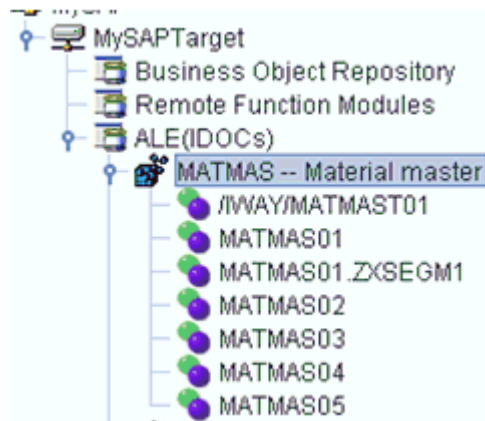
The x icon disappears, indicating that the node is connected, as shown in [Figure 6–55](#).

Figure 6–55 Connected MySAP Target

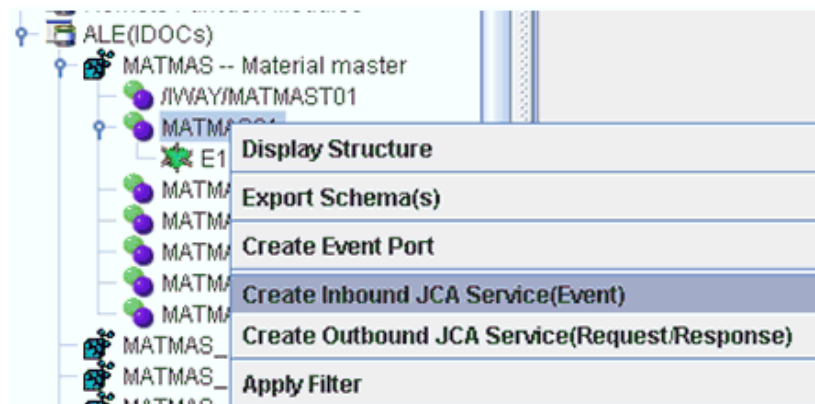


5. Expand the **ALE(IDOCs)** node and select **MATMAS - Material Master**.

The MATMAS list is displayed, as shown in [Figure 6–56](#).

Figure 6–56 MATMAS Material Master List

6. Right-click MATMAS01 from the MATMAS list.

Figure 6–57 Create Inbound JCA Service (Event) Option

7. Select **Create Inbound JCA Service (Event)**, as shown in [Figure 6–57](#).
The Export WSDL dialog is displayed, as shown in [Figure 6–58](#).

Figure 6–58 Export WSDL Dialog

Perform the following steps:

- a. In the **Name** field, specify a name for the WSDL file.
- b. From the **Channel** list, select the channel you created for this inbound service.

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

- c. Three check boxes for Root, Namespace, and Schema validation are also available. Selection of multiple validation options is allowed.
 - Root validation is used to validate the root element in the inbound XML document.
 - Namespace validation is used to validate the namespace in the inbound XML document.
 - Schema validation is used to validate the inbound XML document with the schema in the WSDL document.

During run time, validation is processed based on the validation options that are selected. If multiple validation options are selected, then during run time if the first validation option fails, the remaining validation options are not processed. Root and namespace validations are considered modest levels of validation. Schema validation is a stricter validation level. It is recommended to use root and namespace validation options, unless the root element and namespace are different between the IDOCs in the SAP environment.

8. Click **OK**.

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

6.5.2 Creating an Empty Composite for SOA

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

1. Create a new SOA application.
The Name your project page is displayed.
2. Enter a name for the new SOA Application and click **Next**.
The Configure SOA settings page is displayed.
3. Enter a project name and click **Next**.
The Composite Template list, select **Empty Composite** and click **Finish**.
For more information, see [Section 6.4.2, "Creating an Empty Composite for SOA"](#) on page 6-9.

6.5.3 Defining a BPEL Inbound Process

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

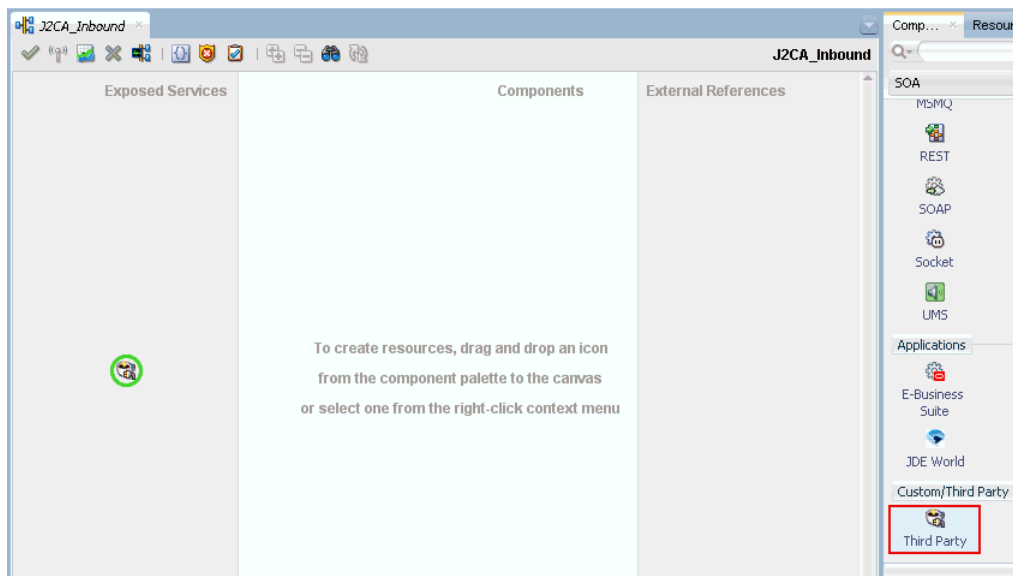
- [Section 6.5.3.1, "Creating a Third Party Adapter Service Component"](#)
- [Section 6.5.3.2, "Creating an Inbound BPEL Process Component"](#)

6.5.3.1 Creating a Third Party Adapter Service Component

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

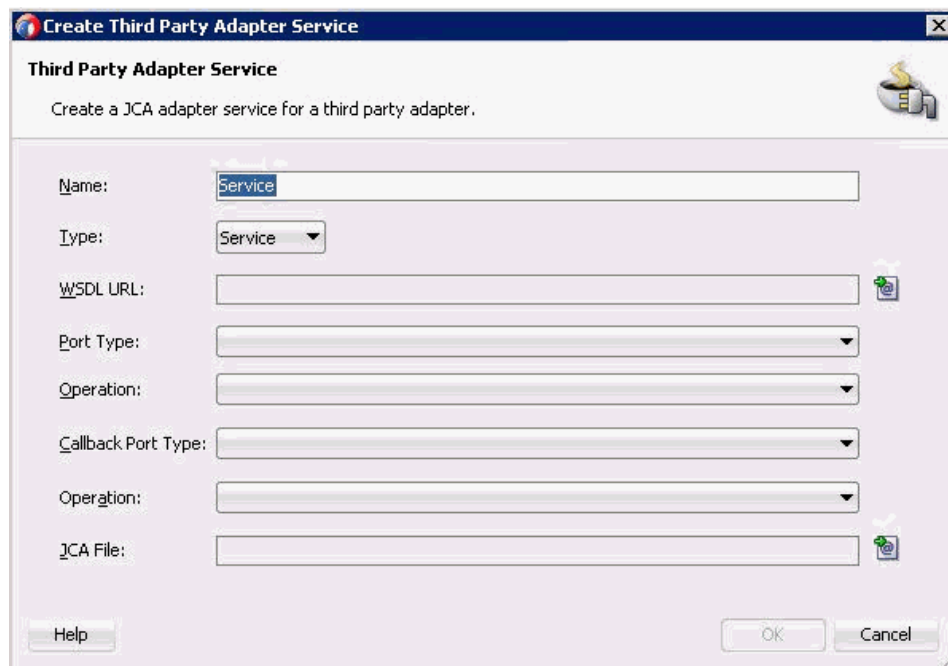
1. Drag and drop the **Third Party Adapter** component from the Service Adapters pane to the Exposed Services pane, as shown in [Figure 6–59](#).

Figure 6–59 *Third Party Adapter Component*



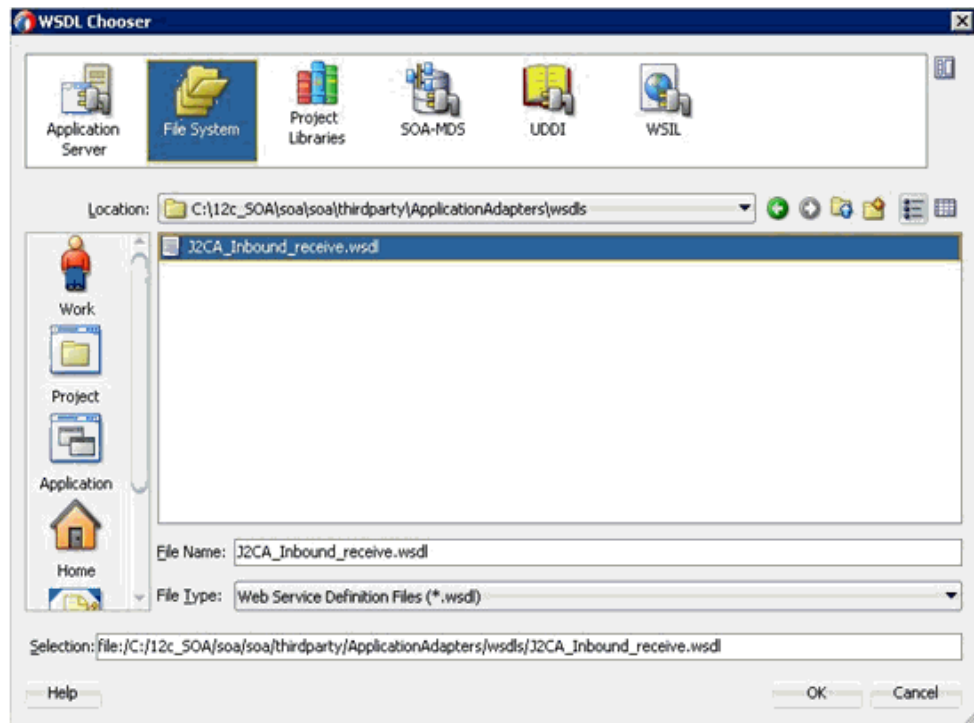
The Create Third Party Adapter Service dialog is displayed, as shown in [Figure 6–60](#).

Figure 6–60 Create Third Party Adapter Service Dialog



2. Enter a name for the third party adapter service, or leave the default name.
3. Ensure that **Service** is selected from the Type 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, as shown in [Figure 6–61](#).

Figure 6–61 WSDL Chooser Dialog

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

`<ADAPTER_HOME>\wsdl`

6. Click **OK**.

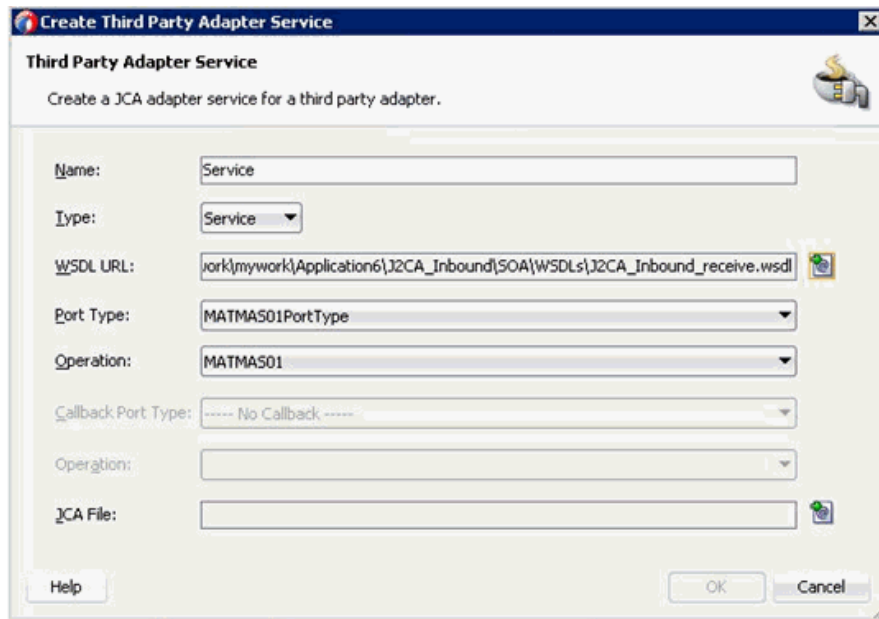
The Localize Files dialog is displayed.

7. Click **OK**.

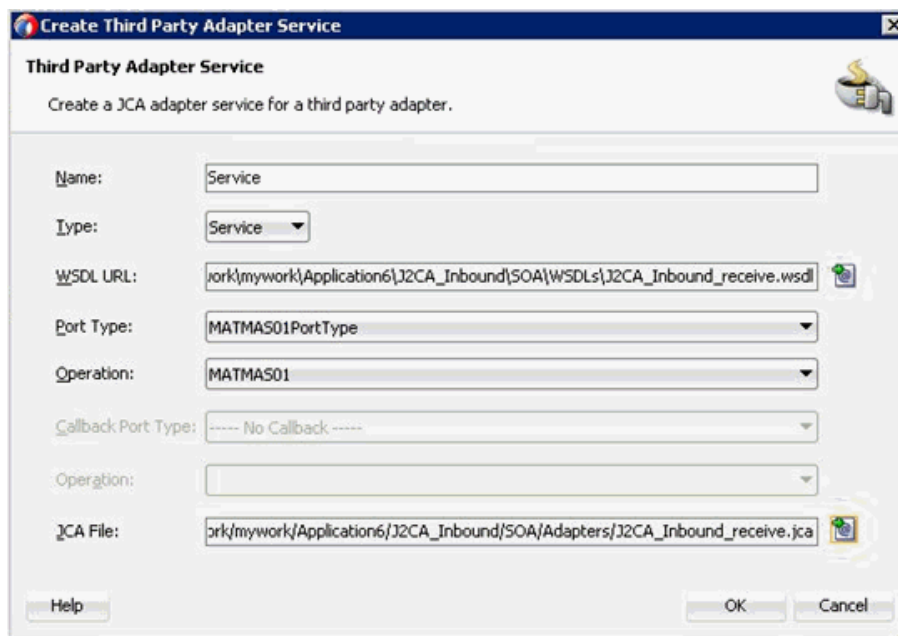
The inbound WSDL file and associated receive/request XML schema file (.xsd) are imported to the project folder that has been created.

You are returned to the Create Third Party Adapter Service dialog, as shown in [Figure 6–62](#).

Figure 6–62 Create Third Party Adapter Service Dialog

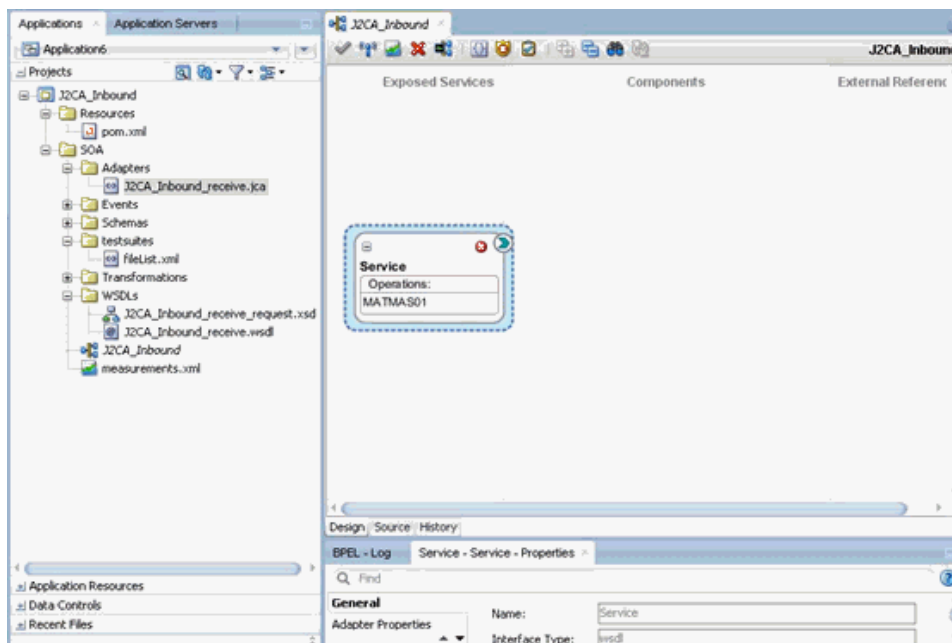


8. Click the **Find JCA file** icon, which is located to the right of the JCA File field. The Transformation Chooser dialog is displayed.
9. Browse and select the JCA properties file from the following directory:
`<ADAPTER_HOME>\wsdl.s`
10. Click **OK**.
 A Copy File message is displayed.
11. Click **Yes**.
 A copy of the JCA properties file is made in the project folder.
 You are returned to the Create Third Party Adapter Service dialog, as shown in [Figure 6–63](#).

Figure 6–63 Create Third Party Adapter Service Dialog

12. Click OK.

The third party adapter service component is created and displayed in the Exposed Services pane, as shown in [Figure 6–64](#).

Figure 6–64 Third Party Adapter Service Component

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

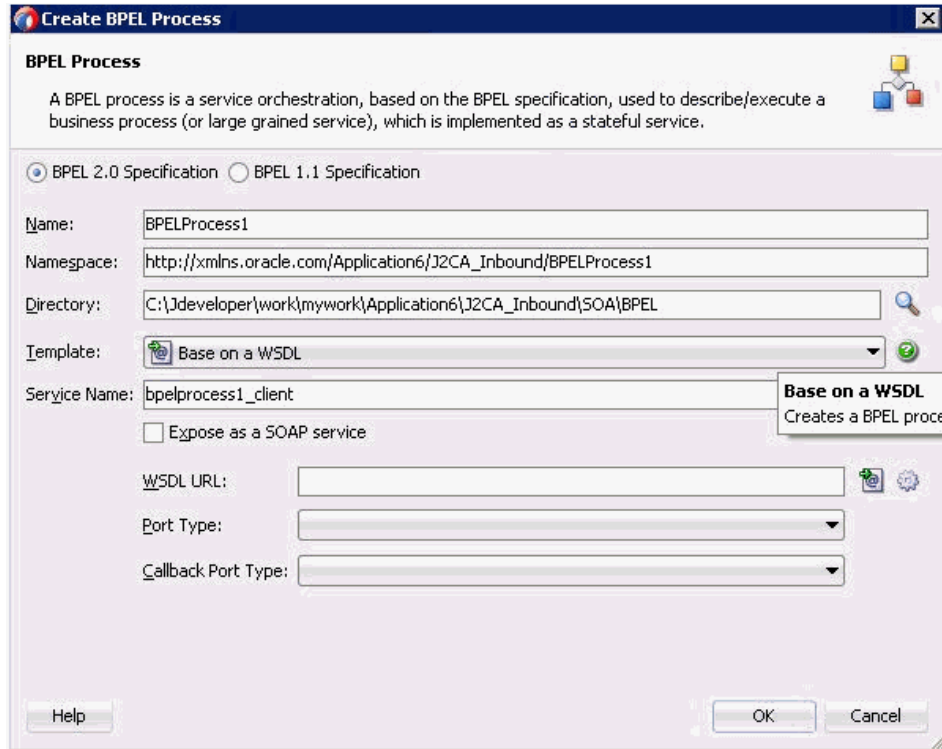
6.5.3.2 Creating an Inbound BPEL Process Component

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.

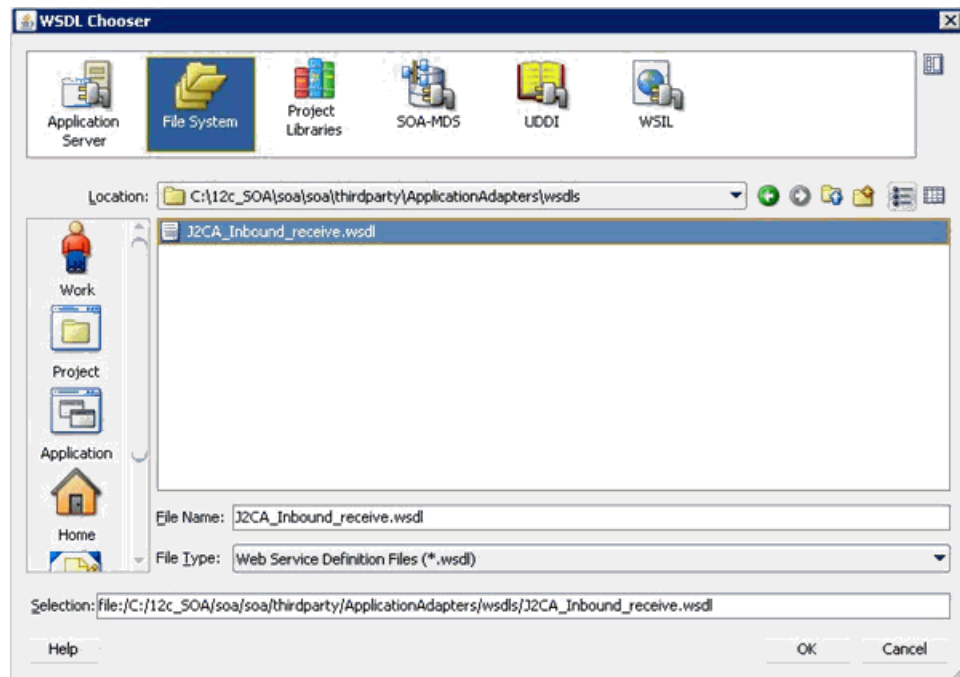
The Create BPEL Process dialog is displayed, as shown in [Figure 6-65](#).

Figure 6-65 Create BPEL Process Dialog



2. In the Name field, enter a name to identify the new inbound BPEL process component, or leave the default name.
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, as shown in [Figure 6-66](#).

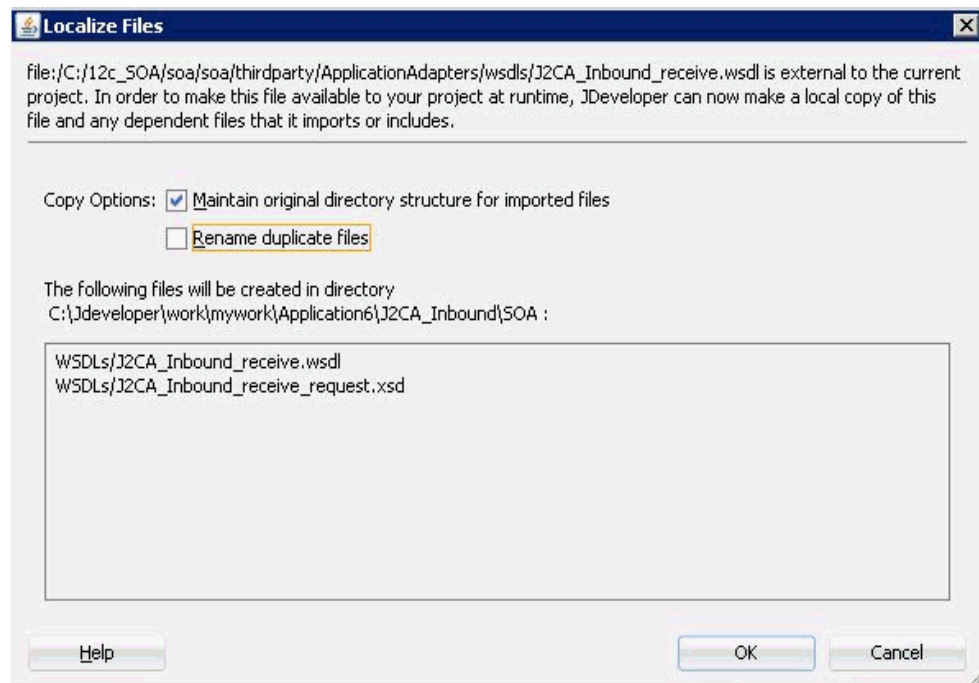
Figure 6–66 WSDL Chooser Dialog

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

```
<ADAPTER_HOME>\wsdls
```

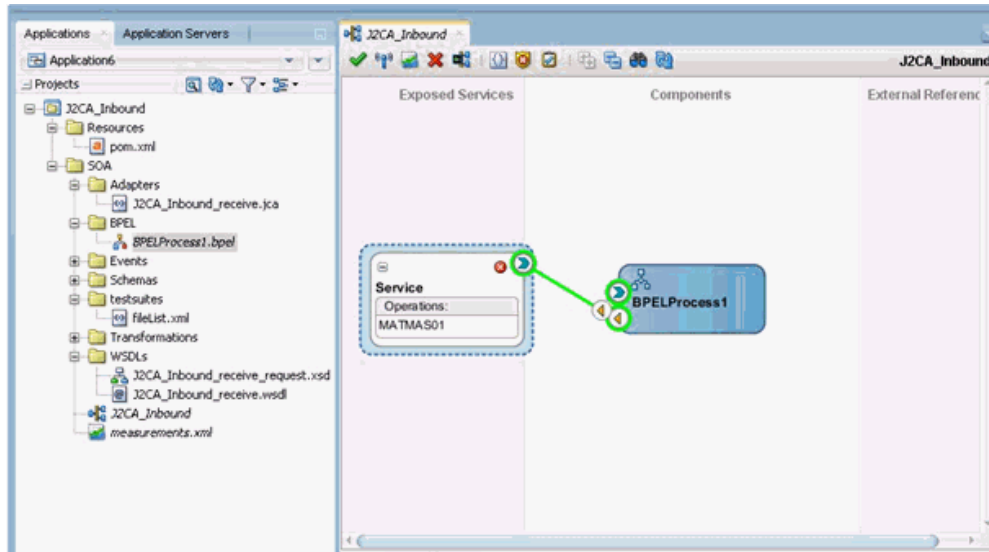
7. Click OK.

The Localize Files dialog is displayed, as shown in [Figure 6–67](#).

Figure 6–67 Localize Files Dialog

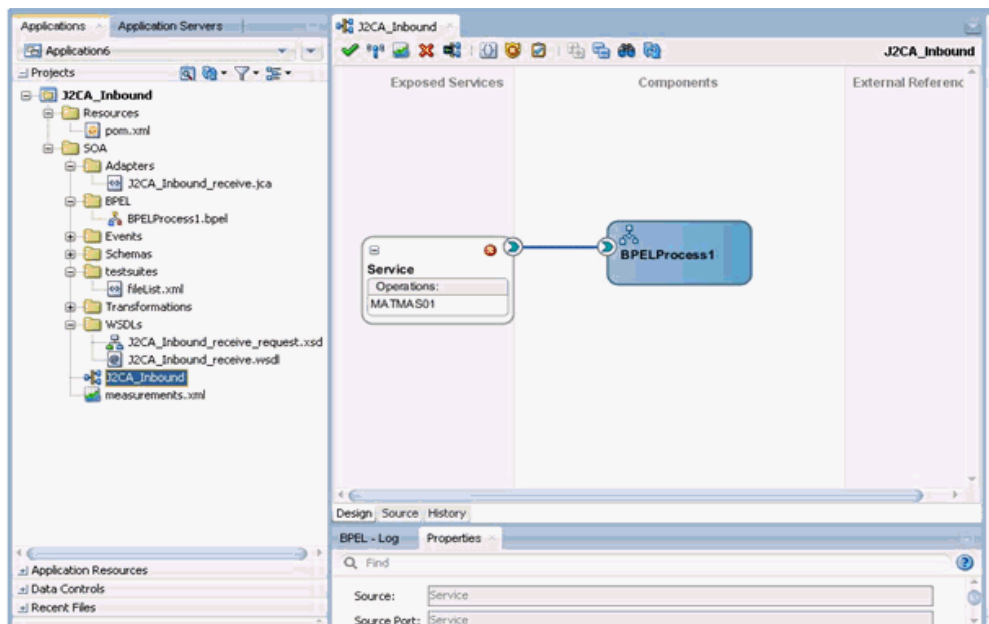
8. Uncheck the **Rename duplicate files** option.
9. Click **OK**.
You are returned to the Create BPEL Process dialog.
10. Click **OK**.

Figure 6–68 Created Connection



11. Create a connection between the third party adapter service component and the inbound BPEL process component, as shown in [Figure 6–68](#).
12. Double-click **J2CA_Inbound** in the left pane.

Figure 6–69 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 6–69](#).

You are now ready to deploy the BPEL inbound process.

6.5.3.3 Adjusting for Known Deployment Issues With 12c

For more information on how to adjust for known deployment issues with 12c, see [Section 6.4.3.3, "Adjusting for Known Deployment Issues With 12c"](#) on page 6-27

6.5.4 Deploying the BPEL Inbound Process

Perform the following steps to deploy the BPEL inbound process.

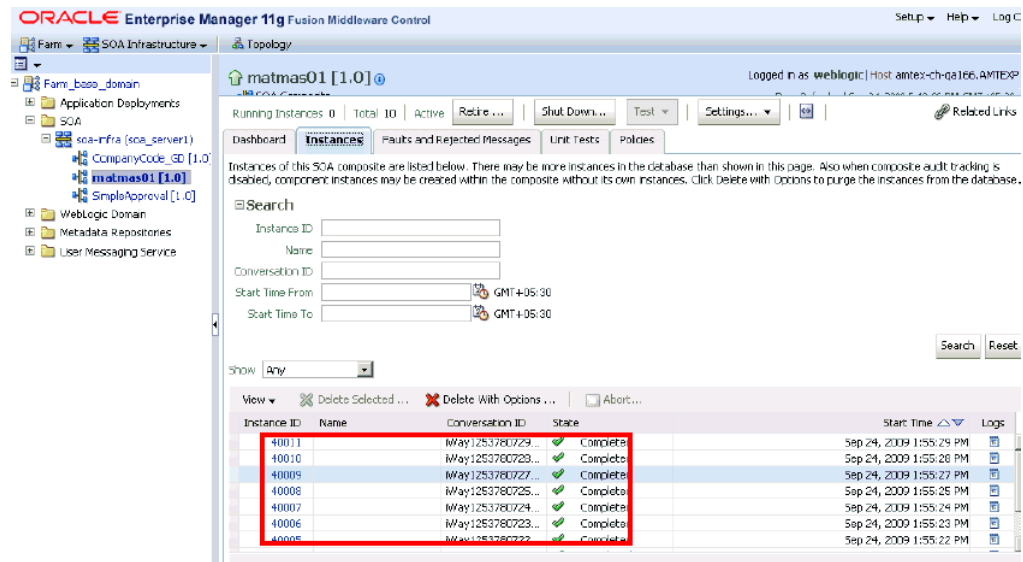
1. Right-click the project name in the left pane, select **Deploy**, and click **matmas01**.
The Deployment Action page is displayed.
2. Ensure that **Deploy to Application Server** is selected.
3. Click **Next**.
The Deploy Configuration page is displayed.
4. Leave the default values selected and click **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 6.4.4, "Deploying the BPEL Outbound Process"](#) on page 6-30.

Once event messages are triggered through SAP GUI, successful instances are received in the Oracle Enterprise Manager console, as shown in [Figure 6–70](#).

Figure 6–70 Received Instances



6.5.5 Triggering an Event in SAP R/3

Events are generated by activity in an application system. For example, SAP R/3 may generate an event as customer information is updated in the system. For more information on events, see ["Configuring an Event Adapter"](#) on page 4-22.

This section contains the following topic:

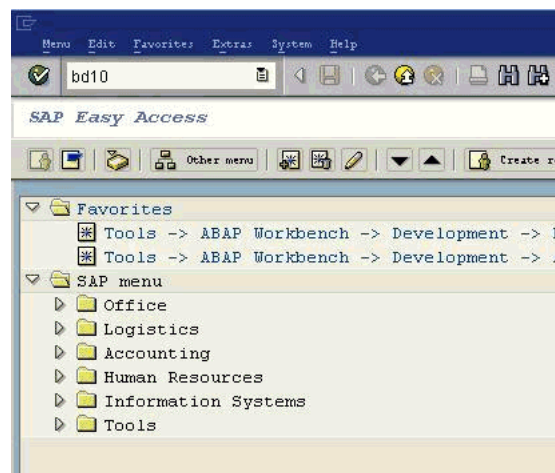
- [Section 6.5.5.1, "Verifying the Results"](#)

The following topics describe how to trigger an event in SAP R/3 and verify event integration using Oracle Application Adapter for SAP R/3.

To trigger an event in SAP R/3:

1. Start the SAP Workbench and log in to the SAP R/3 system, as shown in [Figure 6–71](#).

Figure 6–71 SAP Workbench



2. Run the **bd10** transaction.

Figure 6–72 Send Material Window

Enter the following information in the Send Material window:

- a. In the Material field, enter a material number with a range from 1 to 100, as shown in Figure 6–72.
 - b. In the Logical system field, specify the logical system you are using with SAP R/3.
3. Click the **Execute** button.

Material master data is sent to the logical system specified. If a channel in Application Explorer defined the Program ID with the same value, then the channel receives this material master data from SAP R/3.

6.5.5.1 Verifying the Results

To verify your results:

1. Logon to the Oracle Enterprise Manager console by using the following URL:
`http://localhost:7001/em`
2. Expand your domain in the left pane followed by the **SOA** folder.
3. Select an available inbound BPEL process.
4. Click the **Instances** tab.
Recently received run-time event messages are displayed in the Instances tab.
5. Click the Instance ID to view the received event message.

6.6 Working With Synchronous SAP R/3 Events Using BPEL (J2CA Configuration)

This section describes how to work with synchronous SAP R/3 events using BPEL.

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

```
<ADAPTER_HOME>\etc\sample\SAP_Samples.zip\SAP_Samples\BPEL\J2CA\Synchronous_Events
```

This section includes the following topics:

- [Section 6.6.1, "Overview"](#)
- [Section 6.6.2, "Configuring the Adapter Target"](#)
- [Section 6.6.3, "Configuring the Channel"](#)
- [Section 6.6.4, "Generating WSDL for a Message"](#)
- [Section 6.6.5, "Creating a BPEL Process in JDeveloper"](#)

6.6.1 Overview

Oracle Application Adapter for SAP R/3 supports two ways of receiving SAP R/3 events: Remote Function Call (RFC) or Intermediate Document (IDoc). During event processing, the adapter receives RFCs and IDocs directly from SAP R/3. SAP Message Objects define SAP R/3 events and are published from SAP R/3. The SAP R/3 Integration Broker is used to configure a process for publishing SAP R/3 Message Objects to other third-party applications. Oracle Application Adapter for SAP R/3 can subscribe to SAP R/3 Message Objects.

Oracle Application Adapter for SAP R/3 supports events either asynchronously or synchronously. In the synchronous event model, the enterprise application publishes an output for an event and waits until it receives a response for the publication of the message. For example, when a user adds a customer in a banking application, it initiates an event by sending a request from the banking application to check the credit score of that customer with a credit check application. The credit check application responds with the credit score for the customer. The banking application receives the credit score details and updates the customer records in the banking application. All these constitute one single transaction.

Oracle Application Adapter for SAP R/3 supports inbound transactions through RFCs and IDocs and synchronous events with SAP R/3.

In the case of synchronous events with SAP R/3, when SAP R/3 publishes an event, it becomes the client. Oracle Application Adapter for SAP R/3 acts as a server by receiving the request and does the processing by invoking other applications and sends the response back to SAP. Until that point, the SAP R/3 application waits to receive the response back from the adapter. RFC objects of SAP R/3 support synchronous events and the IDocs do not support synchronous events.

This use case scenario describes how Oracle Application Adapter for SAP R/3 works synchronously. In this use case, two SAP R/3 systems are configured to show how they communicate with each other using synchronous events. One SAP R/3 system acts as a source system, which initiates an event. The other SAP R/3 system acts as a target system from where details are extracted. This use case scenario can be used with any other EIS. However, the source system must always be an SAP R/3 system. For example, the synchronous event origination can be done only from an SAP R/3 system. The target system can be a non-SAP system (for example, Siebel, PeopleSoft, or J.D. Edwards).

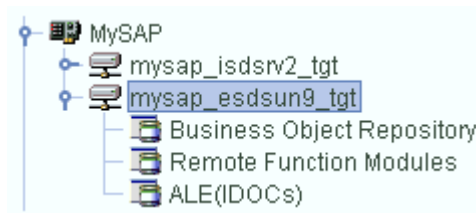
Note: Synchronous events are supported only for objects that have reply schema and are not supported for IDocs.

6.6.2 Configuring the Adapter Target

The following section describes how to configure the adapter target.

1. Open Application Explorer and create two targets for SAP R/3 using a J2CA configuration, as shown in [Figure 6-73](#).

Figure 6-73 SAP Targets



For more information on starting Application Explorer, creating a J2CA configuration, and defining a target to SAP R/3, see [Chapter 4, "Configuring Oracle Application Adapter for SAP R/3"](#).

6.6.3 Configuring the Channel

This section describes how to configure the channel.

1. Create a channel and ensure that REQUEST_RESPONSE is selected from the Processing Mode list in the **Advanced** tab, as shown in [Figure 6-74](#).

Figure 6-74 Application Server Dialog



For more information on creating channels, see [Section 4.10, "Configuring an Event Adapter"](#).

The created channel is listed in Application Explorer. The channel is displayed under the channels node in the left pane. An X over the icon indicates that the channel is currently stopped, as shown in [Figure 6-75](#).

Figure 6–75 New Channel

Note: Do not start the channel, as it is managed by the BPEL PM Server. If you start the channel for testing and debugging purposes, then stop all the channels and close the Application Explorer before run-time deployment for events.

6.6.4 Generating WSDL for a Message

Perform the following steps to generate WSDL for a message:

1. Ensure that the channel is not started before generating the WSDL.
2. Connect to the first SAP adapter target (target1).
3. Generate an inbound WSDL for BAPI_COMPANYCODE_GETLIST (navigate using Remote Function Modules, Financial Accounting, 0002 - Company Code Business Object).
4. Connect to the second SAP adapter target (target2).
5. Generate an outbound WSDL for BAPI_COMPANYCODE_GETLIST (navigate using Remote Function Modules, Financial Accounting, 0002 - Company Code Business Object).

For more information, see [Section 6.4.1, "Generating WSDL for Request/Response Service"](#) and [Section 6.5.1, "Generating WSDL for Event Integration"](#).

6.6.5 Creating a BPEL Process in JDeveloper

This section describes how to create a BPEL process in JDeveloper.

6.6.5.1 Creating an Empty Composite for SOA

To create an empty composite for SOA:

1. In JDeveloper, select the SOA application and create a new project.
2. In the New Gallery window, select the SOA project and click **OK**.
3. Enter the project name and click **Next**.
4. Select **Empty Composite** as the template and click **Finish**.

6.6.5.2 Creating a Third Party Adapter Service for an Inbound WSDL File

To create a third party adapter service for an inbound WSDL file:

1. Drag and drop the **Third Party Adapter** component from the Custom/Thirdparty area to the Exposed Services pane.

The Create Third Party Adapter Service dialog is displayed.

2. Enter a name in the Name field and click the icon next to the WSDL URL field.

The WSDL Chooser dialog is displayed.

3. Select the **File System** tab, browse and select the inbound WSDL created from the default WSDL location and click **Yes**.

The inbound WSDL and request and response XML schema files (.xsd) are imported to the created project folder.

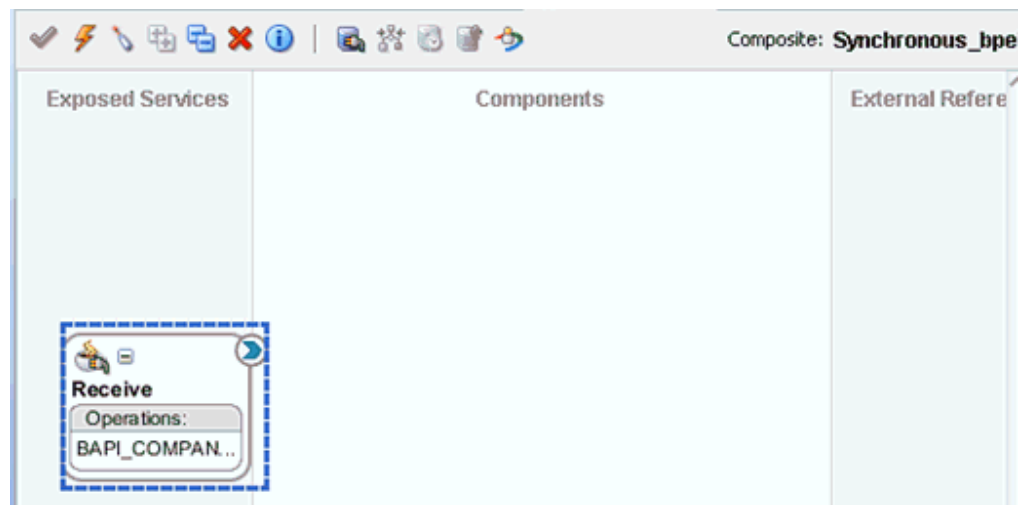
4. Click the **Find jca file** icon to select the particular inbound JCA properties file.
5. Select the JCA properties file from the WSDL location and click **OK**.
6. Click **Yes**.

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

7. Click **OK**.

The third party adapter service for the inbound WSDL is created in the Exposed Services pane, as shown in [Figure 6-76](#).

Figure 6-76 Exposed Services Pane



6.6.5.3 Creating a Third Party Adapter Service for an Outbound WSDL File

To create a third party adapter service for an outbound WSDL file:

1. Drag and drop the **Third Party Adapter** component from the Custom/Thirdparty area to the External References pane.

The Create Third Party Adapter Service dialog is displayed.

2. Enter a name in the Name field and click the icon next to the WSDL URL field.

The WSDL Chooser dialog is displayed.

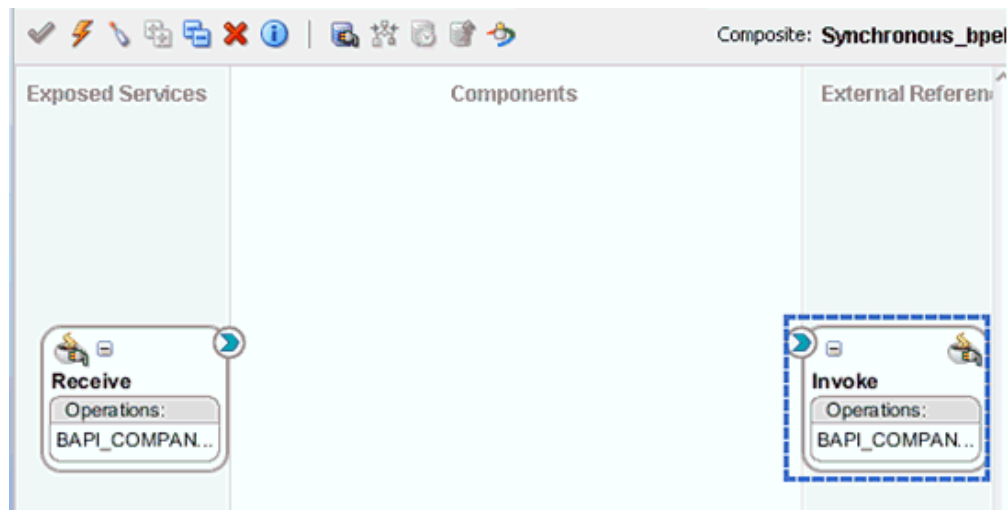
3. Select the **File System** tab, browse and select the outbound WSDL created from the default WSDL location and click **Yes**.

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

4. Click the **Find jca file** icon to select the particular outbound JCA properties file.

5. Select the JCA properties file from the WSDL location and click **OK**.
6. Click **Yes**.
A copy of the JCA properties file is made in the project folder.
7. Click **OK**.
The third party adapter service for the outbound WSDL is created in the External References pane, as shown in [Figure 6–77](#).

Figure 6–77 External References Pane



6.6.5.4 Creating a Synchronous BPEL Process

To create a synchronous BPEL process:

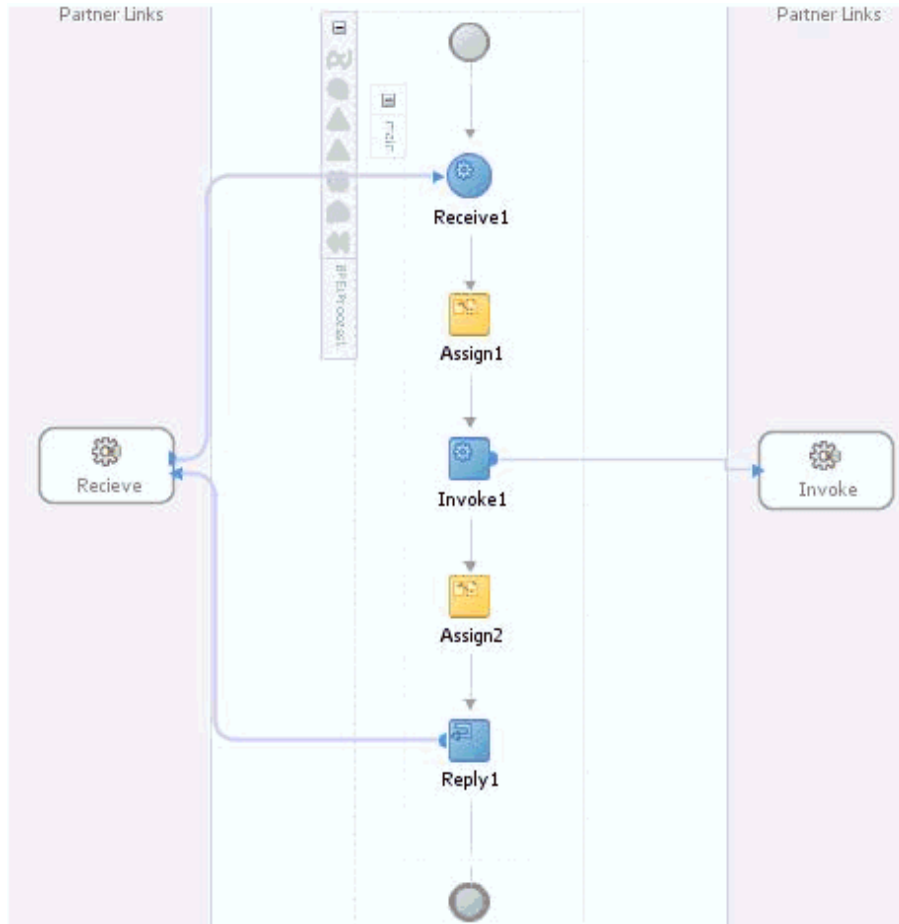
1. Drag and drop the **BPEL Process** component from the Components area to the Components pane.
The Create BPEL Process window is displayed.
2. Enter a name for the BPEL process in the Name field and select **Define Service Later** from the Template list.
3. Click **OK**.
The BPEL Process component is created in the Components pane.
4. Connect the third party adapter service **Receive** to the BPEL process component **Synchronous_Events**.
5. Click **OK**.
6. Connect the BPEL process component **Synchronous_Events** to the third party adapter service component **Invoke**.
7. Double-click the BPEL Process component in the Components pane.
8. Drag and drop the **Receive** activity to the Visual Editor and place it in the designated placeholder labeled **Drop Activity Here**.
9. Connect the **Receive** activity to the **Receive** third party adapter service.
The Edit Receive dialog is displayed.
10. Click the **plus** icon to the right of the Variable field.

- The Create Variable dialog is displayed.
11. Click **OK**.
You are returned to Edit Receive dialog.
 12. Select the **Create Instance** option.
 13. Click **Apply** and then **OK**.
 14. Drag and drop the **Assign** activity to the Visual Editor and place it below the **Receive1** icon.
 15. Drag and drop the **Invoke** activity to the Visual Editor and place it below the **Assign1** icon.
 16. Connect the **Invoke** activity to the **Invoke** third party adapter service.
The Edit Invoke dialog is displayed.
 17. Under the Input tab, click the **plus** icon to the right of the **Input** field.
The Create Variable dialog is displayed.
 18. Click **OK**.
You are returned to the Edit Invoke dialog.
 19. Select the **Output** tab, click the **plus** icon to the right of the Output field.
The Create Variable dialog is displayed.
 20. Click **OK**.
You are returned to the Edit Invoke dialog.
 21. Click **Apply** then **OK**.
 22. Drag and drop the **Assign** activity to the Visual Editor and place it below the **Invoke1** icon.
 23. Drag and drop **Reply** activity to the Visual Editor and place it below the **Assign2** icon.
 24. Connect the **Reply** activity to the **Receive** third party adapter service.
The Edit Reply dialog is displayed.
 25. Click the **plus** icon in the Variable field.
The Create Variable dialog is displayed.
 26. Click **OK**.
You are returned to the Edit Reply dialog.
 27. Click **Apply** and then **OK**.
 28. Double-click the **Assign1** icon to open it in the edit mode.
 29. In this assign activity, the input variable of the Receive activity must be assigned to the Input variable of the Invoke activity.
 30. Click **Apply** and then **OK**.
You are returned to the Visual Editor.
 31. Double click the **Assign2** activity to open it in the edit mode.
 32. In this assign activity, the output variable of the Invoke activity must be assigned to the output variable of the Reply activity.

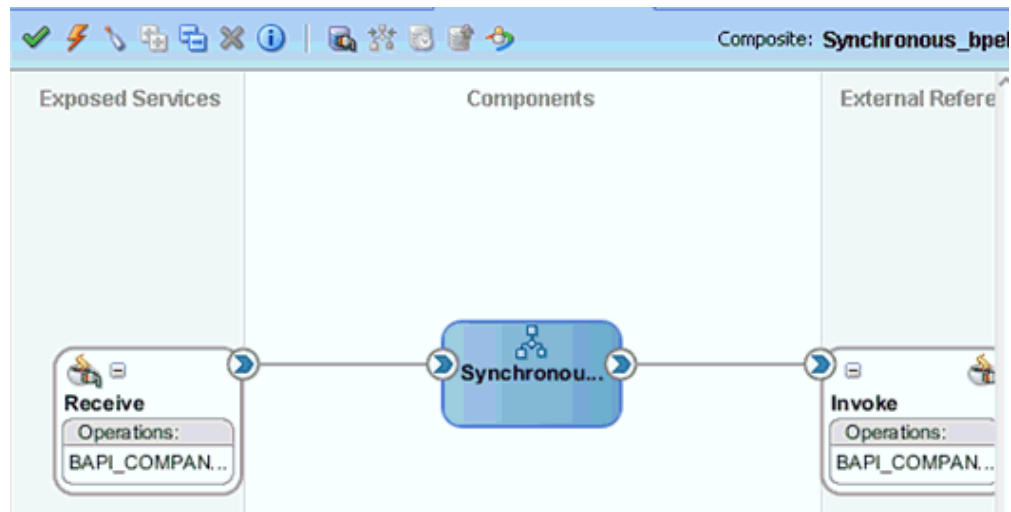
33. Click **Apply** and then **OK**.

You are returned to the Visual Editor. The BPEL process for synchronous events is completed, as shown in [Figure 6-78](#).

Figure 6-78 Visual Editor



34. Save all the process from the File menu, as shown in [Figure 6-79](#).

Figure 6–79 Completed BPEL Process

6.6.5.5 Adjusting for Known Deployment Issues With 12c

For more information on how to adjust for known deployment issues with 12c, see [Section 6.4.3.3, "Adjusting for Known Deployment Issues With 12c"](#) on page 6-27.

6.6.5.6 Deploying the BPEL Process

For more information on how to deploy the BPEL process, see [Section 6.4.4, "Deploying the BPEL Outbound Process"](#) on page 6-30.

6.7 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\SAP_Samples.zip\SAP_Samples\BPEL\BSE\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)

This section includes the following topics:

- [Section 6.7.1, "Generating a WSDL File for Request and Response Services Using a Web Service"](#)
- [Section 6.7.2, "Creating an Empty Composite for SOA"](#)
- [Section 6.7.3, "Defining a BPEL Outbound Process"](#)

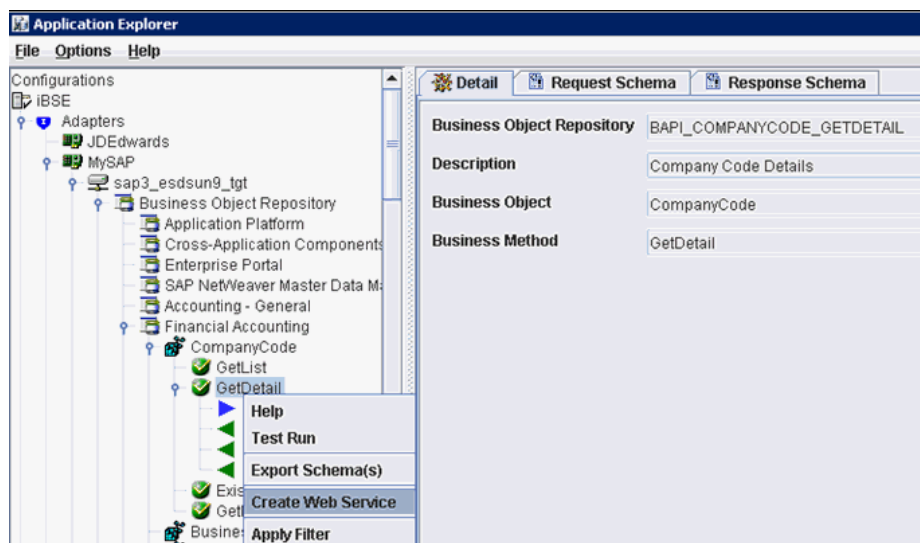
6.7.1 Generating a WSDL File for Request and Response Services Using a Web Service

Before you design a BPEL process, you must generate the respective WSDL file using Application Explorer.

To generate a WSDL file for request and response services using a Web service:

1. Start Application Explorer and connect to a defined MySAP target (BSE configuration).
For more information on defining a target and connecting to SAP R/3, see [Section 4.5.1, "Defining a Target to SAP R/3"](#).
2. Expand the MySAP target to which you are connected.
3. Expand **Business Object Repository, Financial Accounting, Company Code Business Object**, and then select **GetDetail**.
4. Right-click the **GetDetail** node and select **Create Web Service** from the menu, as shown in [Figure 6–80](#).

Figure 6–80 GetDetail Node



The Create Web Service dialog is displayed, as shown in [Figure 6–81](#).

Figure 6–81 Create Web Service Dialog



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 wsdl 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.

6.7.2 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 6.4.2, "Creating an Empty Composite for SOA,"](#) on page 6-9.

6.7.3 Defining a BPEL Outbound Process

This section describes how to configure a BPEL outbound process component.

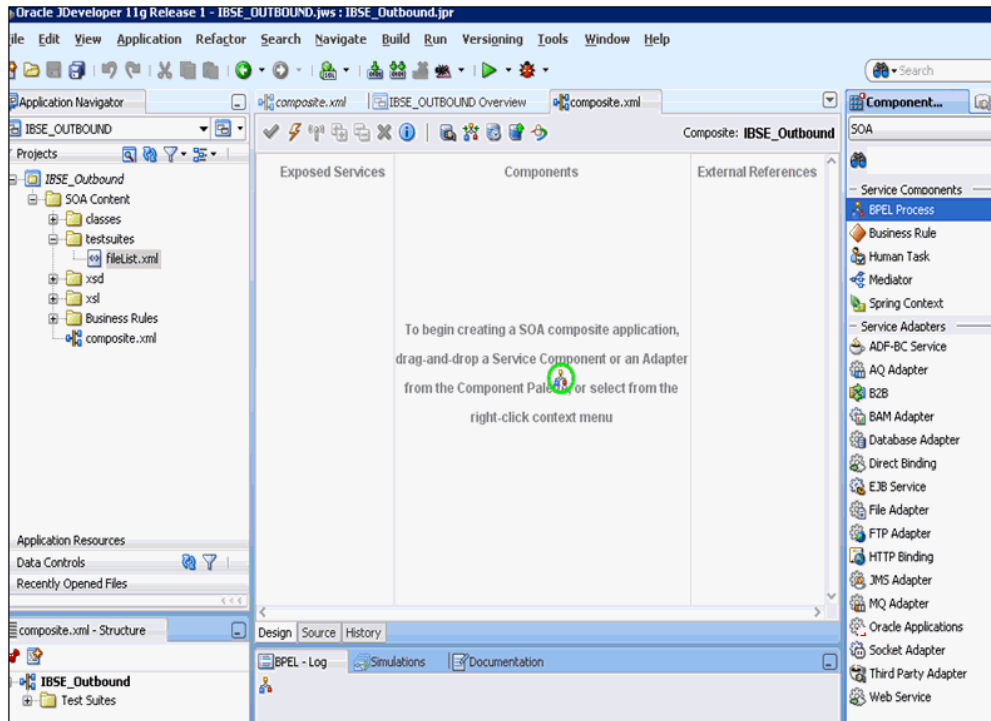
This section includes the following topics:

- [Section 6.7.3.1, "Creating a Partner Link"](#)
- [Section 6.7.3.2, "Creating BPEL Activities and Mappings With the Created Partner Link"](#)

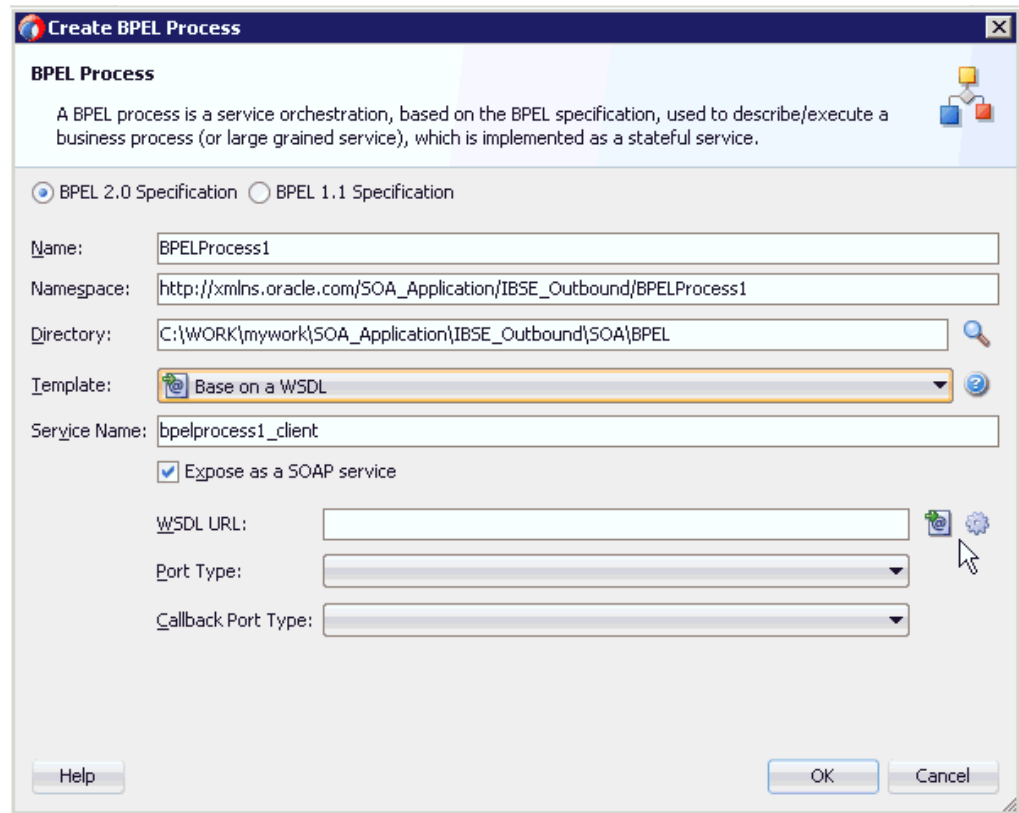
To define a BPEL outbound process:

1. Drag and drop the **BPEL Process** component from the Service Components pane to the Components pane, as shown in [Figure 6-82](#).

Figure 6–82 BPEL Process Component



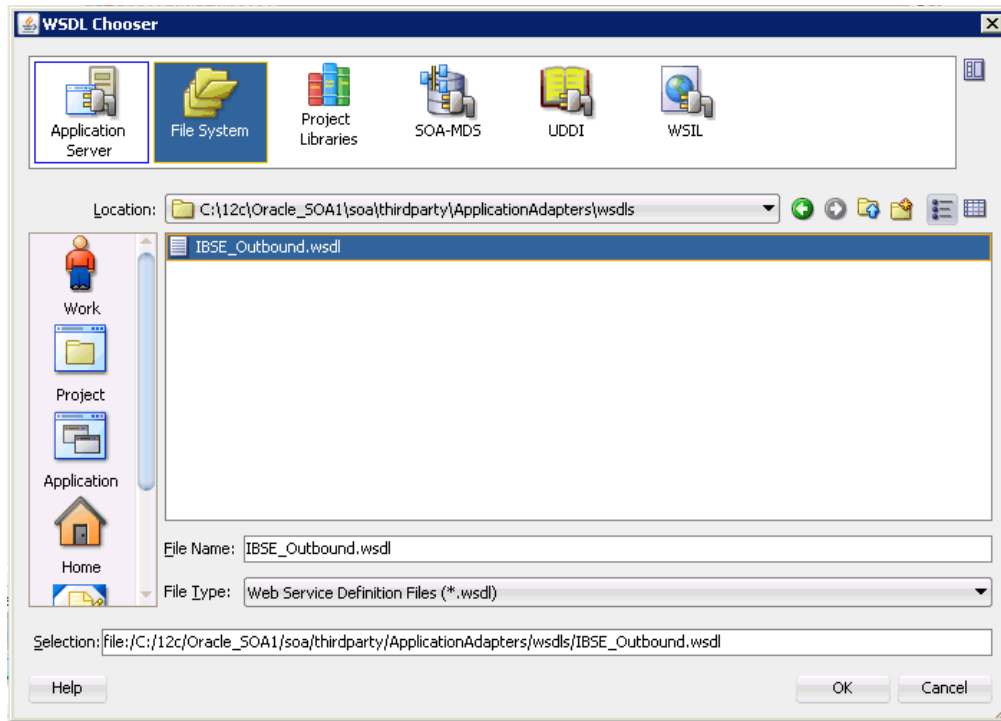
2. In the Name field, enter a name to identify the new outbound BPEL process component.
By default, the BPEL 2.0 Specification option is selected.
3. From the Template drop-down list, select **Base on a WSDL**.
4. Click the **Find existing WSDLs** icon, which is located to the right of the WSDL URL field, as shown in [Figure 6–83](#).

Figure 6–83 Find Existing WSDLs Icon

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 6–84](#).

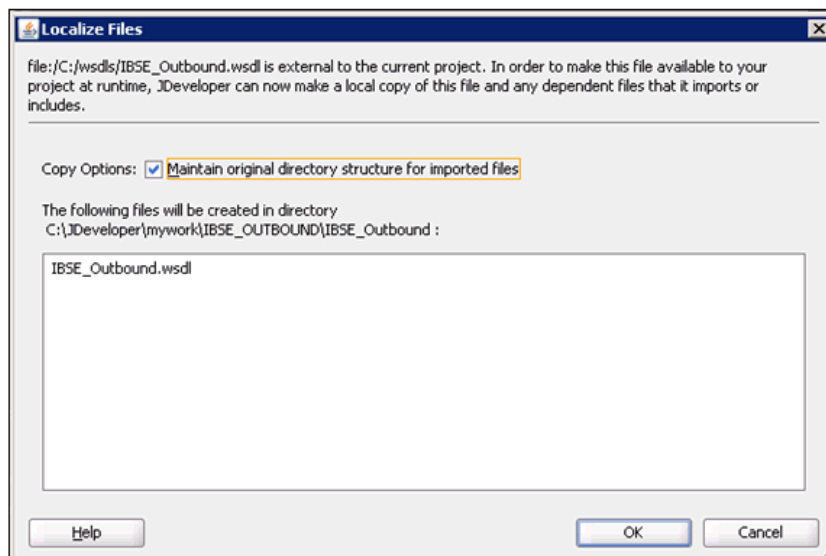
Figure 6–84 WSDL Chooser Dialog



The Localize Files window is displayed.

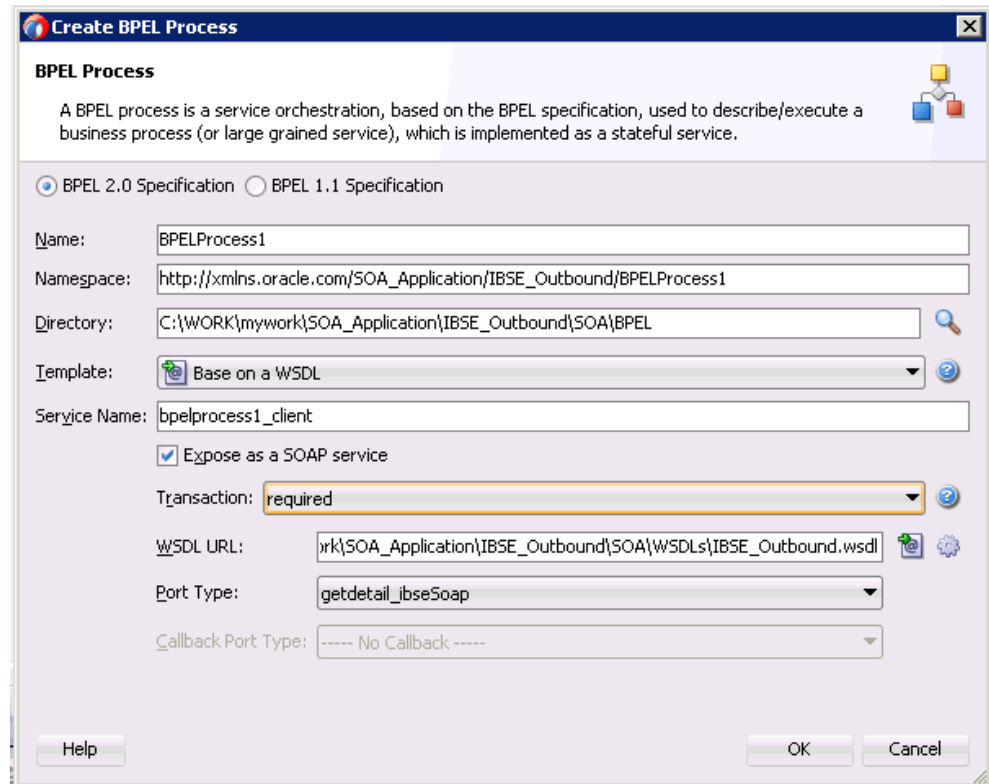
6. In the displayed Localize Files window, click **OK**. This imports the WSDL file to the project folder, as shown in [Figure 6–85](#).

Figure 6–85 Localize Files Window

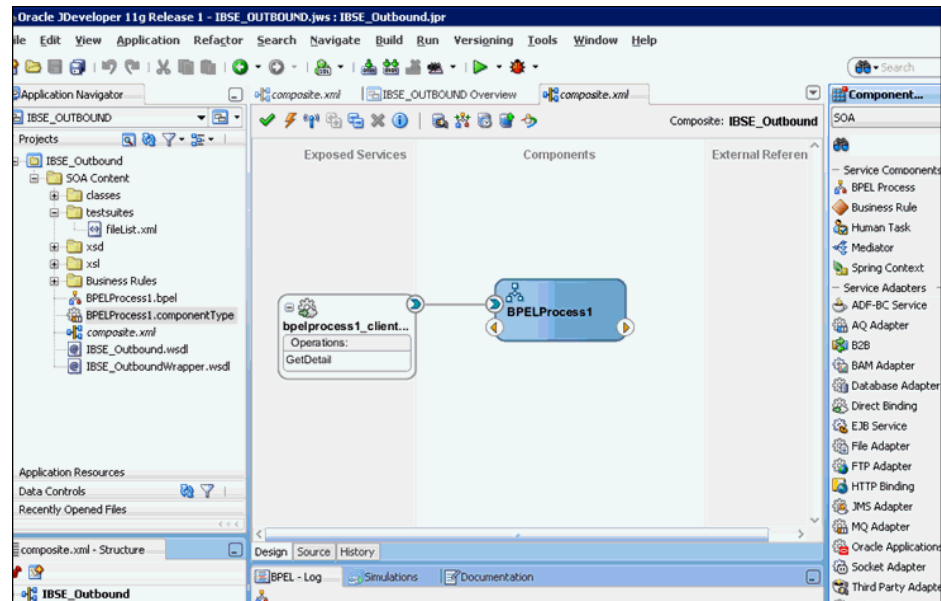


The Create BPEL Process window is displayed.

7. In the BPEL Process pane, click **OK**, as shown in [Figure 6–86](#).

Figure 6–86 BPEL Process Pane

The BPEL Process component is created and displayed, as shown in [Figure 6–87](#).

Figure 6–87 BPEL Process Component

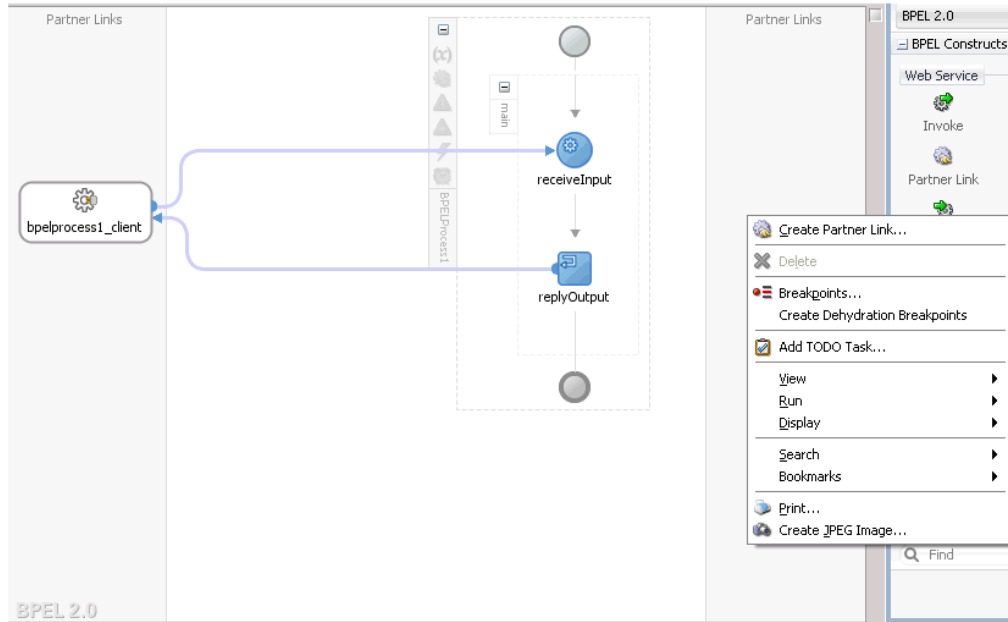
6.7.3.1 Creating a Partner Link

This section describes how to create a partner link.

To create a partner link:

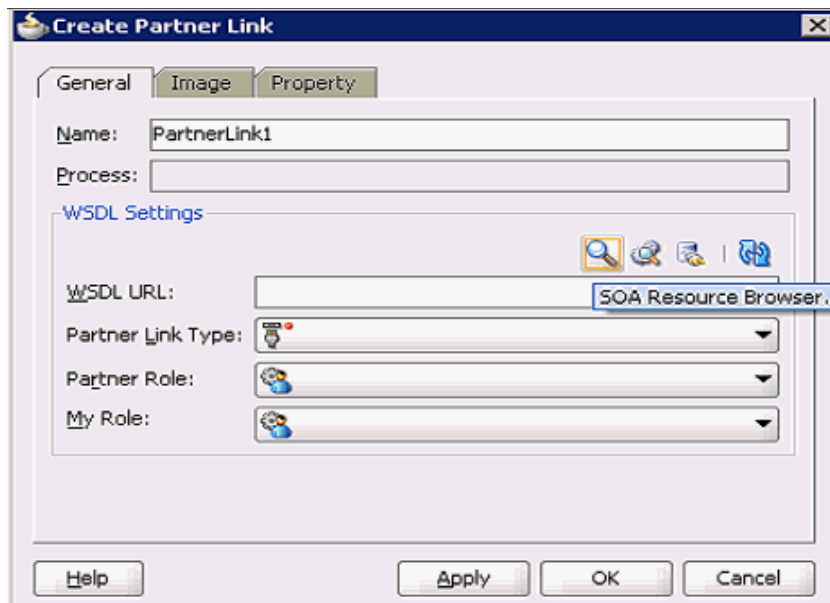
1. Double-click the outbound BPEL process component in the Components pane.
2. Right-click on the **Partner Links** pane and select **Create Partner Link**, as shown in [Figure 6–88](#).

Figure 6–88 Create Partner Link

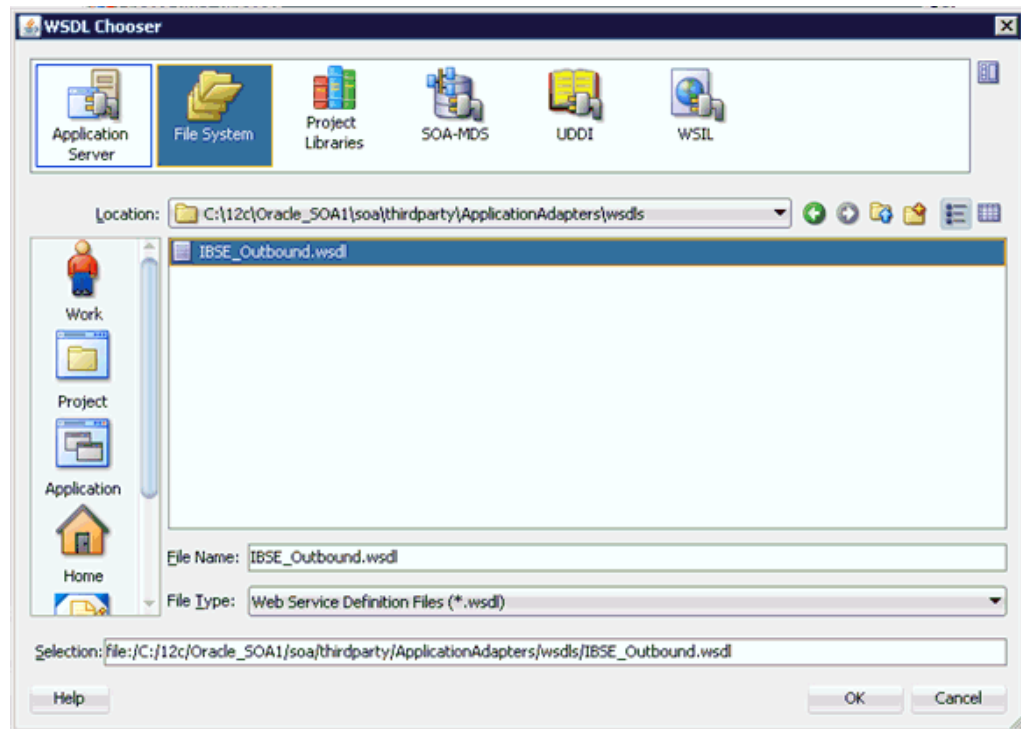


3. In the displayed Create Partner Link window, provide an appropriate name and click on the SOA Resource Browser tool, as shown in [Figure 6–89](#).

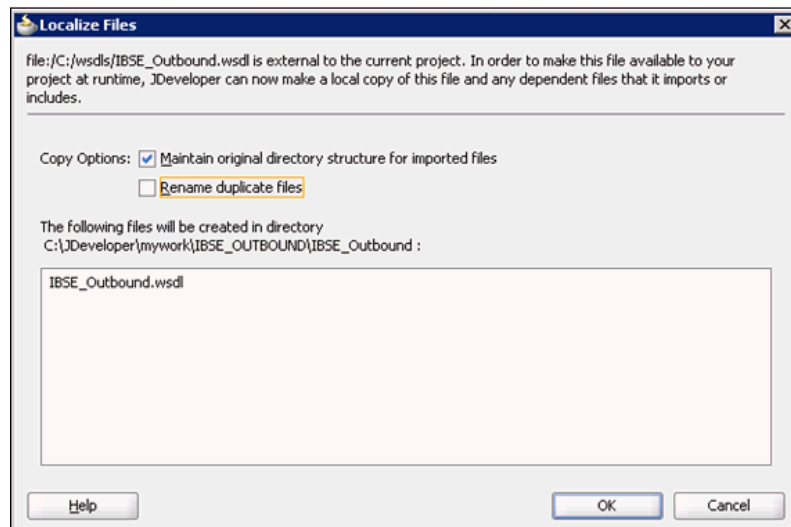
Figure 6–89 SOA Resource Browser Tool



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 6–90](#).

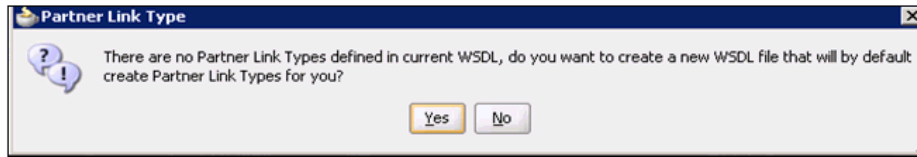
Figure 6–90 WSDL Chooser Dialog

5. In the displayed Localize Files window, uncheck the **Rename duplicate files** check box and click **OK**, as shown in [Figure 6–91](#).

Figure 6–91 Localize Files Window

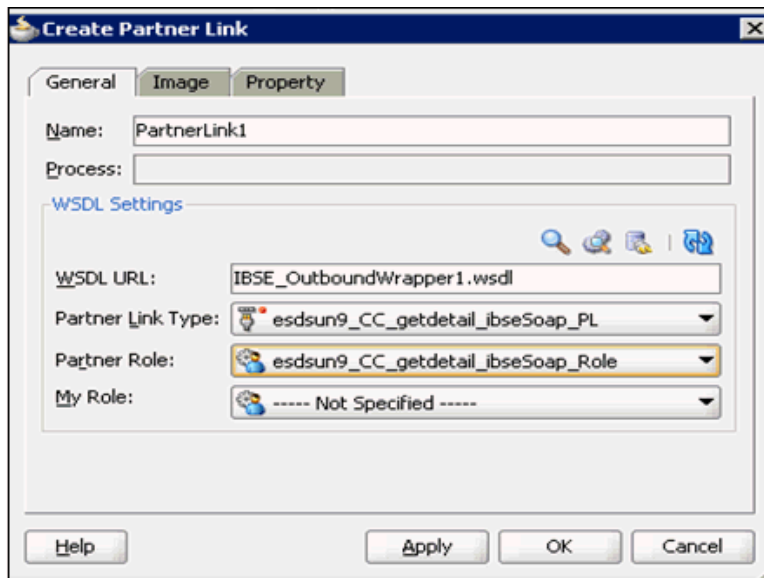
6. Click **Yes** in the displayed Partner Link Type window, as shown in [Figure 6–92](#).

Figure 6–92 Partner Link Type



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 6–93](#).

Figure 6–93 Create Partner Link



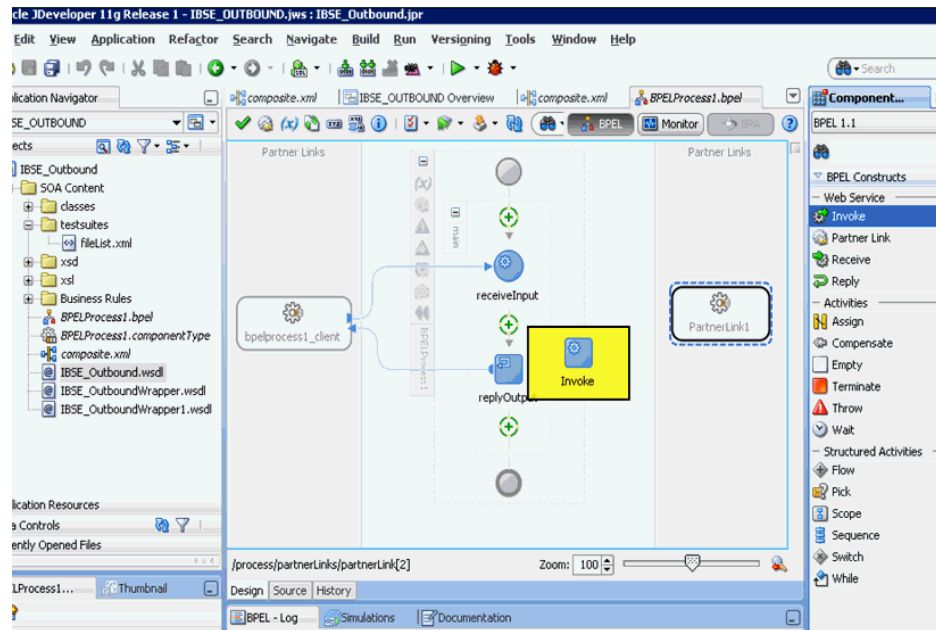
6.7.3.2 Creating BPEL Activities and Mappings With the Created Partner Link

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:

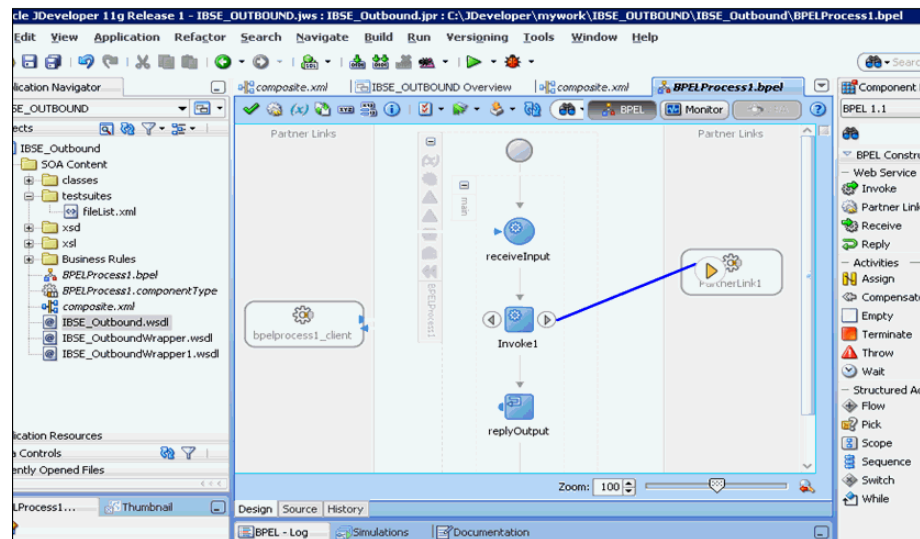
1. Drag and drop the **Invoke** activity component from BPEL Constructs to the Components pane. Place it between the **receiveInput** activity component and the **replyOutput** activity component, as shown in [Figure 6–94](#).

Figure 6–94 Invoke Activity Component



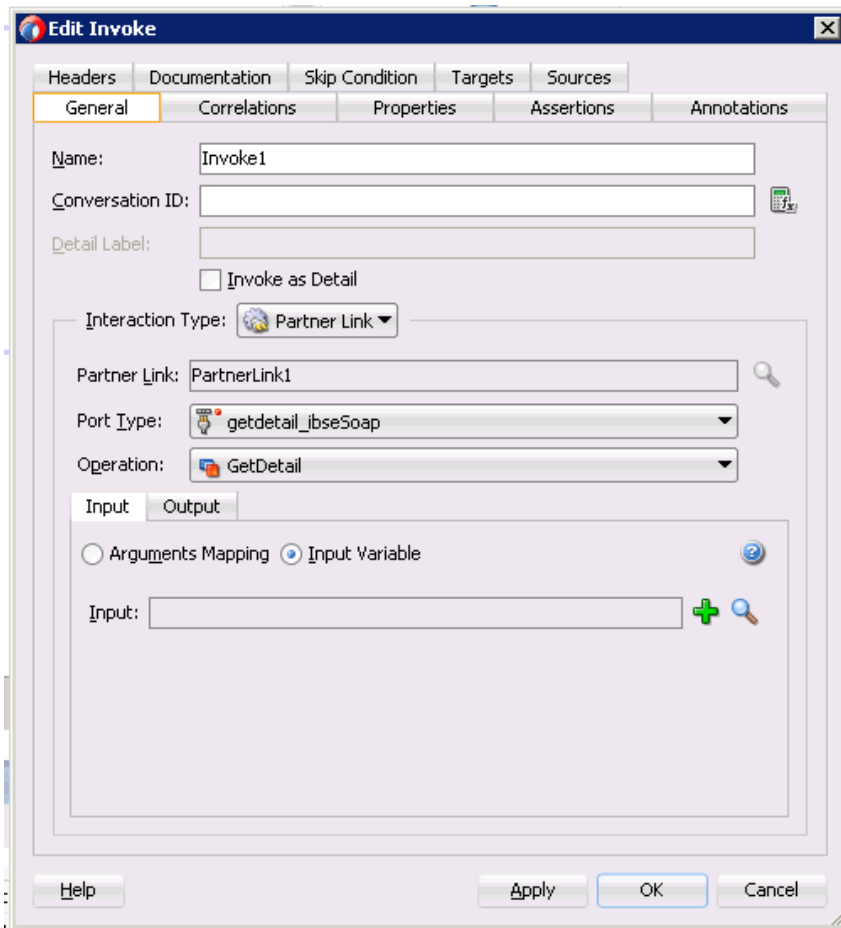
2. Create a connection between the new **Invoke** activity component (Invoke1) and the **Partner Link** component (Partner link1), as shown in [Figure 6–95](#).

Figure 6–95 Partner Link Component



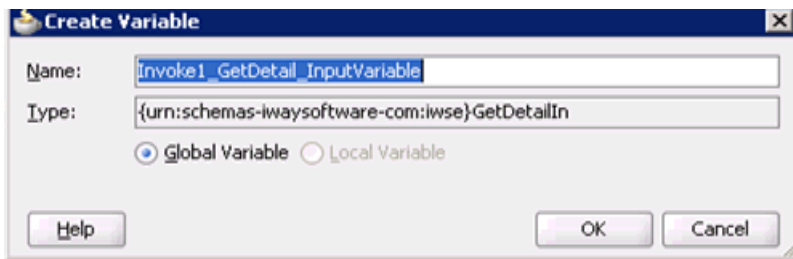
3. In the displayed Edit Invoke window, click the Plus (+) icon, located to the right of the Input field, to configure a new input variable, as shown in [Figure 6–96](#).

Figure 6–96 Edit Invoke Window

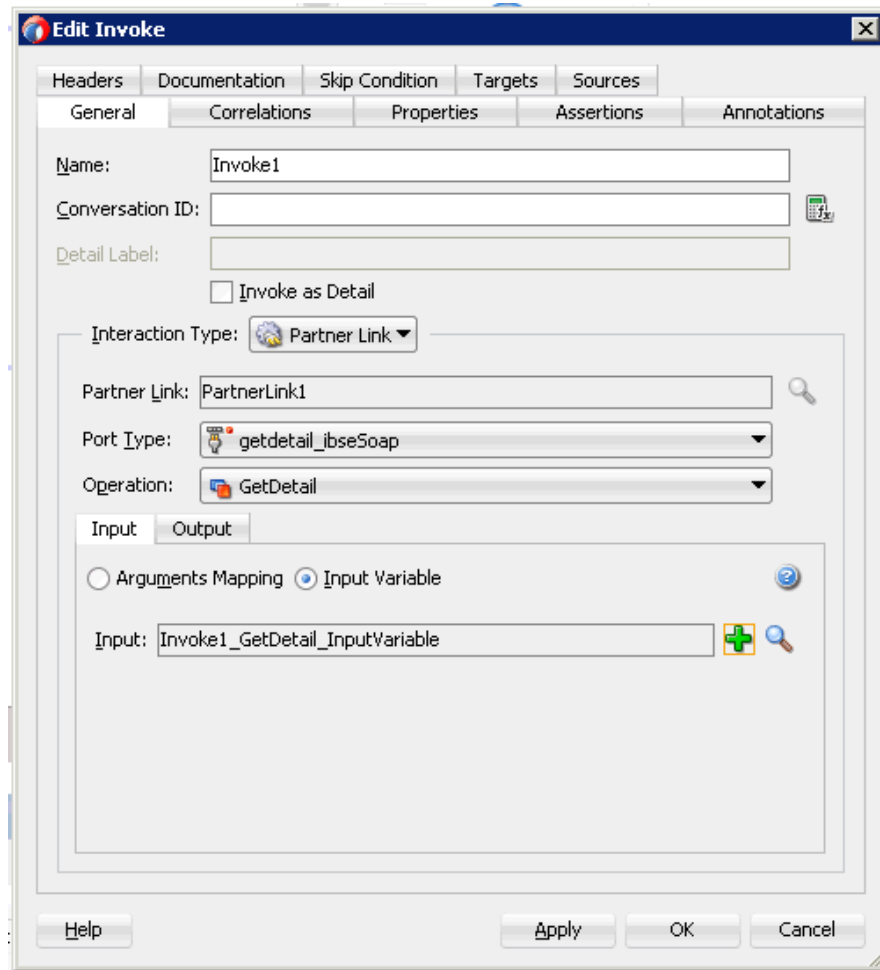


4. Accept the default values that are provided for the new input variable and click **OK**, as shown in [Figure 6–97](#).

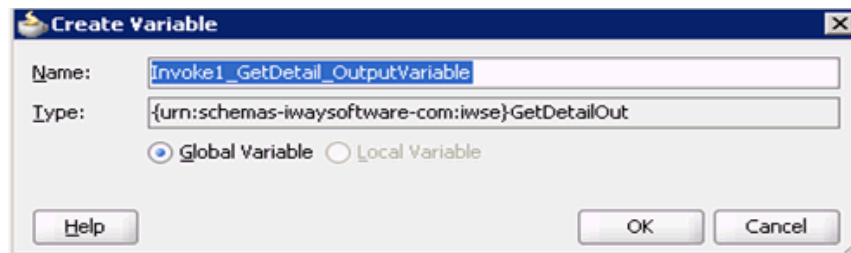
Figure 6–97 New Input Variable



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 6–98](#).

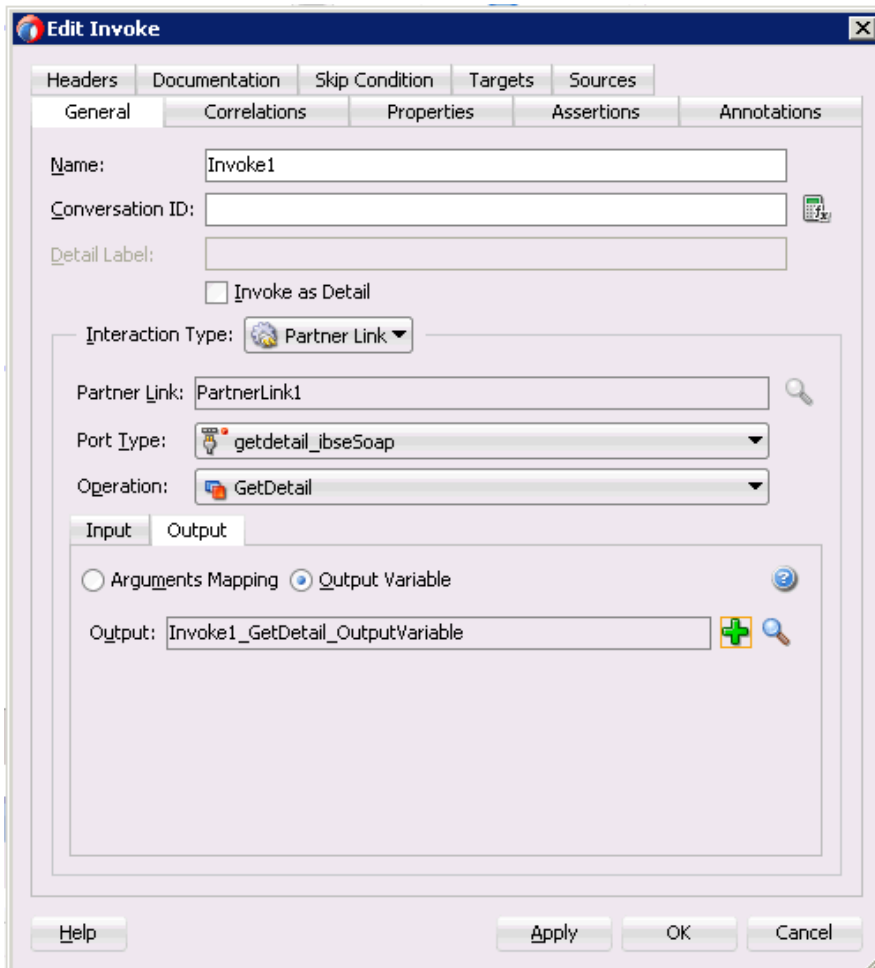
Figure 6–98 Edit Invoke Window

6. Accept the default values that are provided for the new output variable and click **OK**, as shown in [Figure 6–99](#).

Figure 6–99 New Output Variable

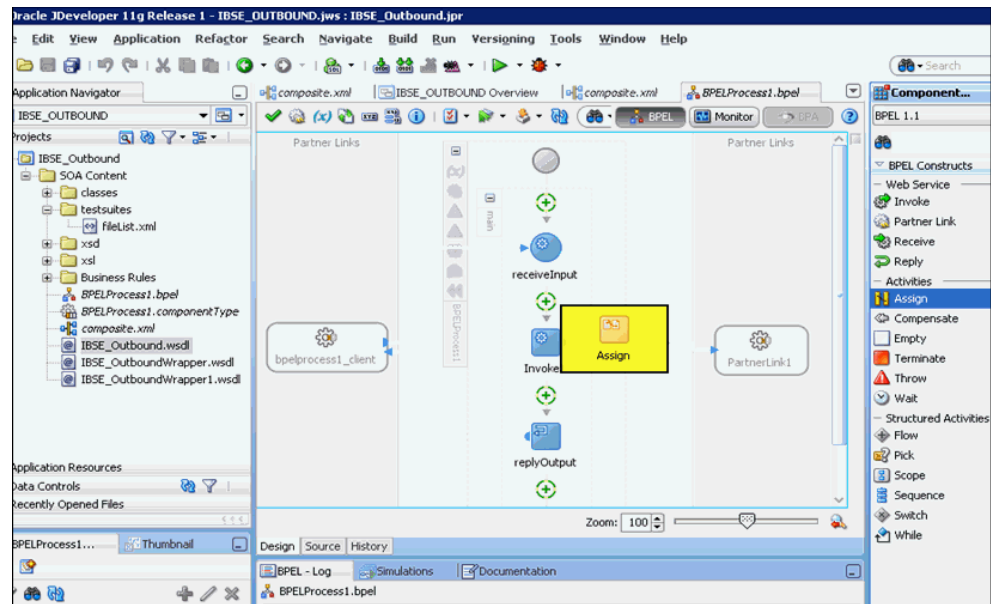
7. Click **Apply** and then **OK**, as shown in [Figure 6–100](#).

Figure 6–100 Edit Invoke Window



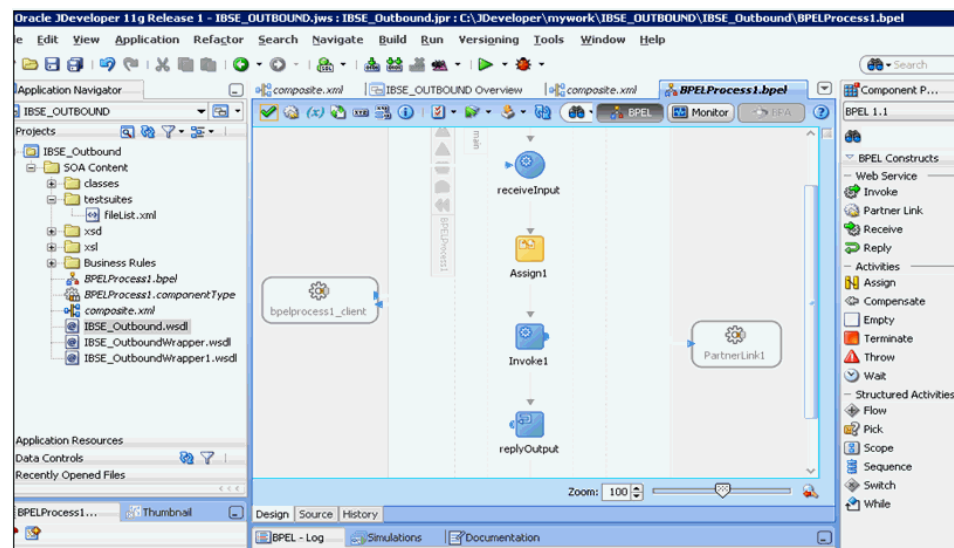
8. Drag and drop the **Assign** activity component from BPEL Constructs to the Components pane. Place it between the **Receive** activity component (receiveInput) and the **Invoke** activity component (Invoke1), as shown in [Figure 6–101](#).

Figure 6–101 Assign Activity Component



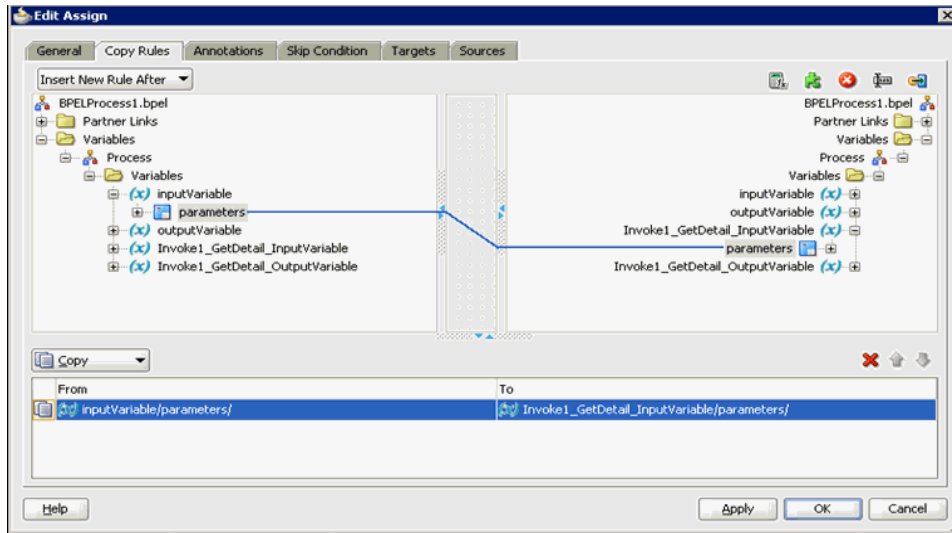
9. Double-click the new **Assign** activity component (Assign1), as shown in [Figure 6–102](#).

Figure 6–102 Assign Activity Component



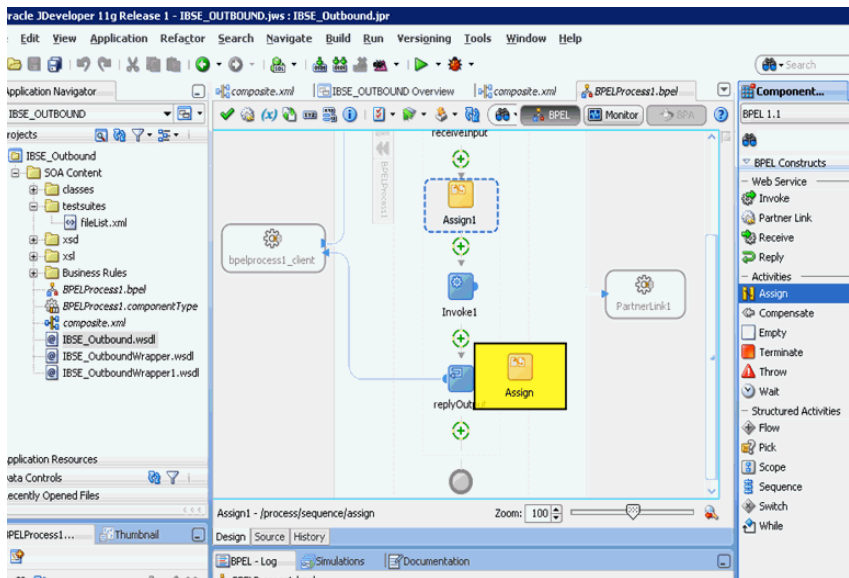
10. In the left pane, under **Variables**, expand **InputVariable**, and then select **parameters**.
11. In the right pane, under **Variables**, expand **Invoke1_GetDetail_InputVariable**, and then select **parameters**.
12. Drag and map the **InputVariable** parameters to the **Invoke1_GetDetail_InputVariable** parameters, as shown in [Figure 6–103](#).

Figure 6–103 *InputVariable Parameters*



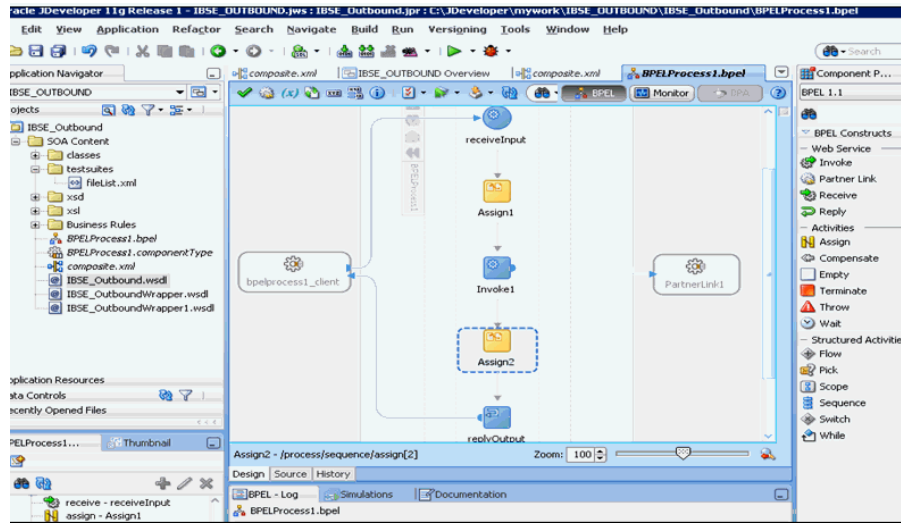
13. Click **Apply** and then **OK**.
14. Drag and drop the **Assign** activity component to the Components pane and place it between the **Invoke** activity (Invoke1) and the **Reply** activity (replyOutput), as shown in [Figure 6–104](#).

Figure 6–104 *Assign Activity Component*



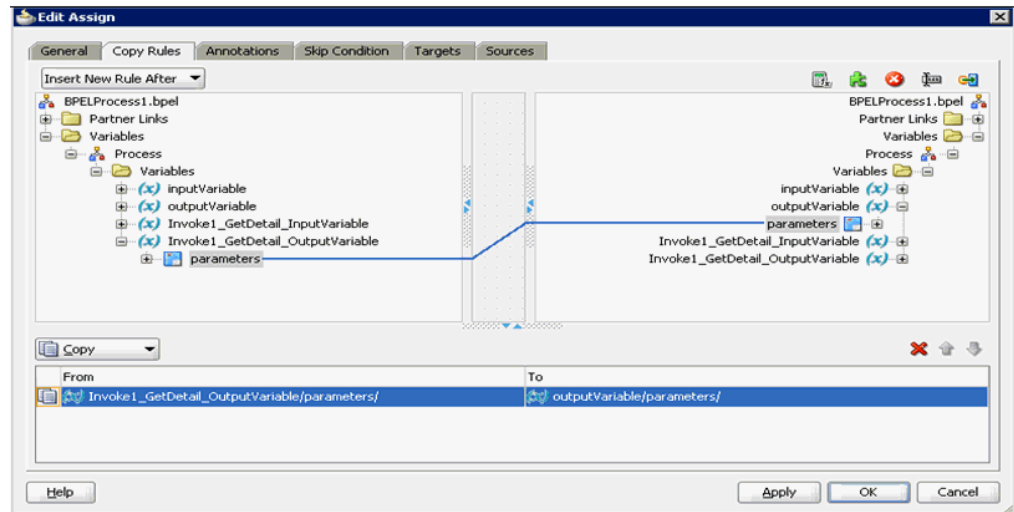
15. Double-click the new **Assign** activity component (Assign2), as shown in [Figure 6–105](#).

Figure 6-105 New Assign Activity Component



16. In the left pane, under Variables, expand **Invoke1_GetDetail_OutputVariable**, and then select **parameters**.
17. In the right pane, under Variables, expand **outputVariable**, and then select **parameters**.
18. Drag and map the **Invoke1_GetDetail_OutputVariable** parameters to the **outputVariable** parameters, as shown in Figure 6-106.

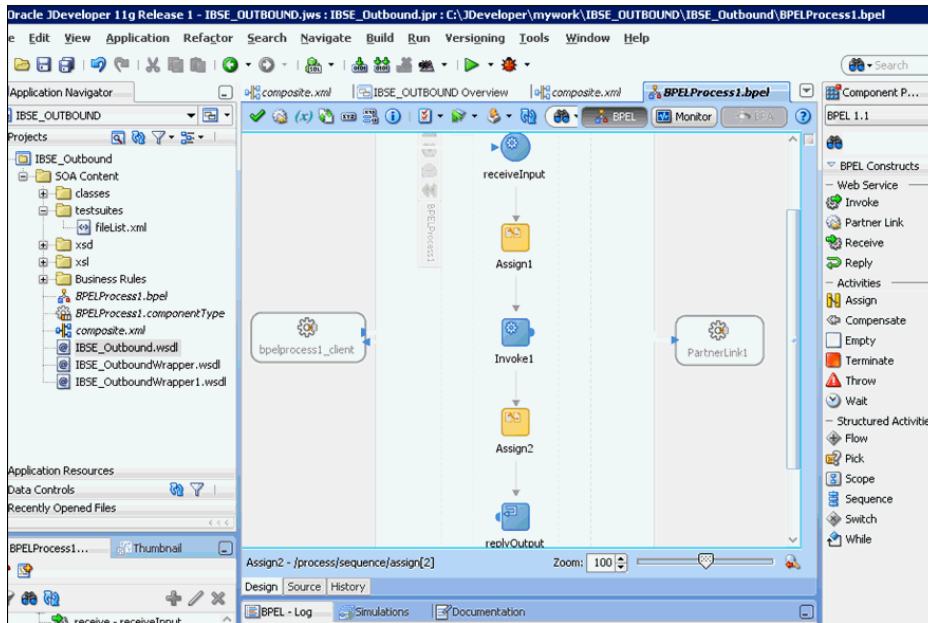
Figure 6-106 outputVariable Parameters



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

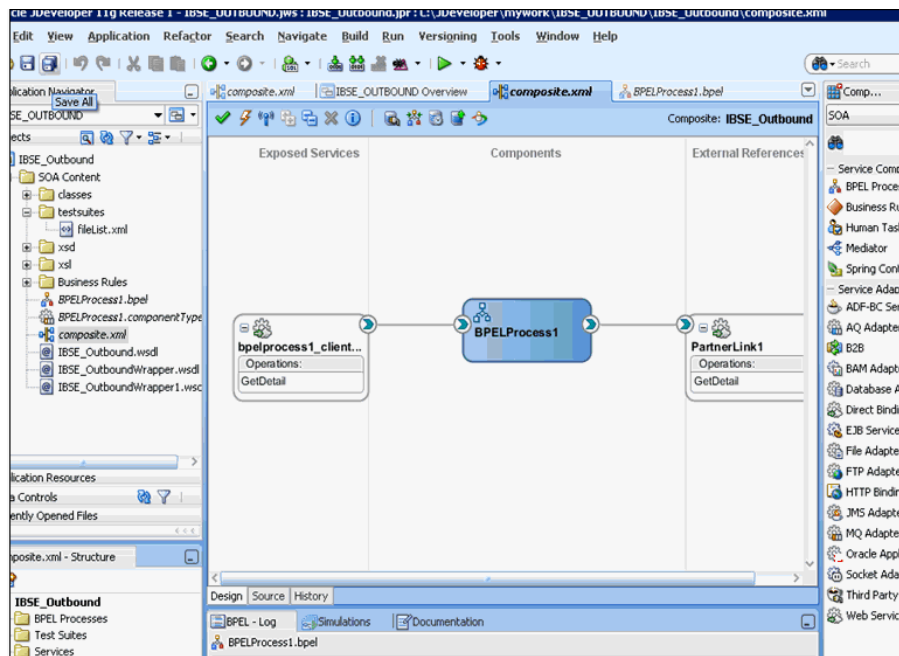
You are returned to the component pane, as shown in Figure 6-107.

Figure 6–107 Component Pane



20. Click the **Save All** icon in the menu bar to save the new outbound BPEL process component that was configured, as shown in [Figure 6–108](#).

Figure 6–108 Save All Icon



You are now ready to deploy the BPEL Outbound process. You can follow the same procedure as in [Section 6.4.4, "Deploying the BPEL Outbound Process"](#).

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"](#).

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 7.1, "Configuring a New Application Server Connection"](#)
- [Section 7.2, "Configuring a Mediator Outbound Process \(J2CA Configuration\)"](#)
- [Section 7.3, "Configuring a Mediator Inbound Process \(J2CA Configuration\)"](#)
- [Section 7.4, "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 SAP R/3 must be installed on Oracle WebLogic Server.
- SAP R/3 must be configured for inbound and outbound processing. For more information, see [Appendix A, "Understanding the SAP System Role in Remote Processing"](#).

The examples in this chapter present the configuration steps necessary for demonstrating service and event integration with SAP R/3. Prior to using this material, you must be familiar with the following:

- How to configure Oracle Application Adapter for SAP R/3 for services and events. For more information, see [Chapter 4, "Configuring Oracle Application Adapter for SAP R/3"](#).
- How to configure Oracle JDeveloper. For more information, see [Chapter 6, "Integration With BPEL Service Components in the Oracle SOA Suite"](#).

Overview of Mediator Integration

Mediator provides a comprehensive application integration framework. Oracle Application Adapter for SAP R/3 used with Mediator enables you to seamlessly integrate enterprise software, eliminating 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 Mediator is a two-step process:

1. **Design Time:** Oracle Application Adapter for SAP R/3 is configured in Application Explorer for services and events, as described in [Chapter 4, "Configuring Oracle Application Adapter for SAP R/3"](#). Integration logic is modeled in Mediator.
2. **Runtime:** The underlying platform treats this metadata as run-time instructions to enable the communication between participating applications.

7.1 Configuring a New Application Server Connection

For more information on how to configure a new Application Server connection in Oracle JDeveloper, see [Section 6.3, "Configuring a New Application Server Connection"](#).

7.2 Configuring a Mediator Outbound Process (J2CA Configuration)

This section describes how to configure a Mediator outbound process to your SAP R/3 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\SAP_Samples.zip\SAP_Samples\Mediator\J2CA\Outbound_  
Project
```

This section contains the following topics:

- [Section 7.2.1, "Creating an Empty Composite for SOA"](#)
- [Section 7.2.2, "Defining a Mediator Outbound Process"](#)
- [Section 7.2.3, "Deploying the Mediator Outbound Process"](#)
- [Section 7.2.4, "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 ["Generating WSDL for Request/Response Service"](#) on page 6-8.

7.2.1 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 6.4.2, "Creating an Empty Composite for SOA"](#) on page 6-9.

7.2.2 Defining a Mediator Outbound Process

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

- [Section 7.2.2.1, "Configuring a Third Party Adapter Service Component"](#)
- [Section 7.2.2.2, "Configuring an Outbound Mediator Process Component"](#)
- [Section 7.2.2.3, "Configuring the Routing Rules"](#)
- [Section 7.2.2.4, "Adjusting for Known Deployment Issues With 12c"](#)

7.2.2.1 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, or leave the default name.
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>\wsdl.s`

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>\wsdl.s`

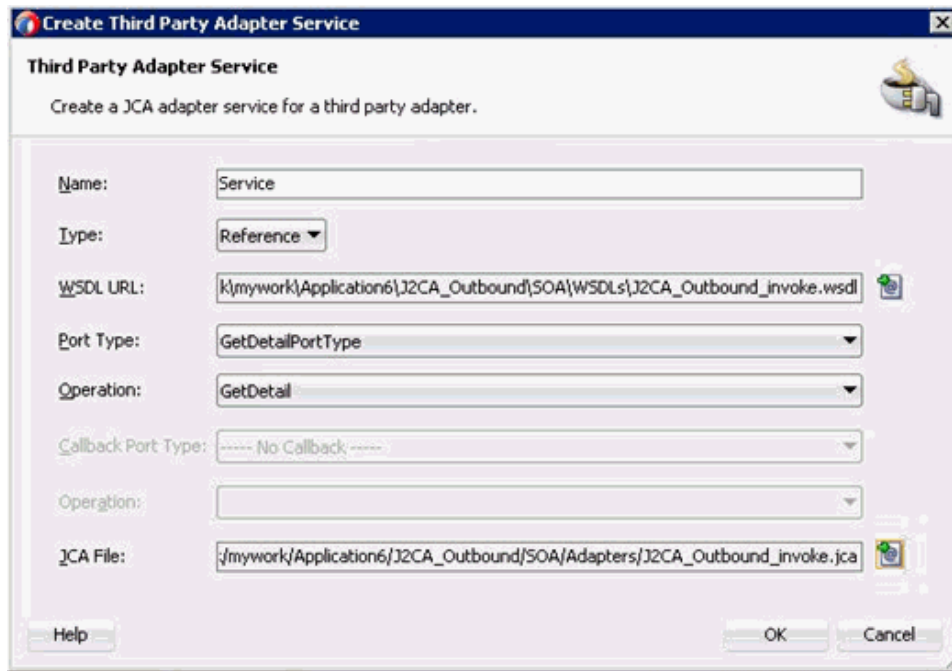
10. Click **OK**.

A Copy File confirmation message is displayed.

11. Click **Yes**.

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

Figure 7-1 Create Third Party Adapter Service Dialog



12. Click **OK**.

The third party adapter service component 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.

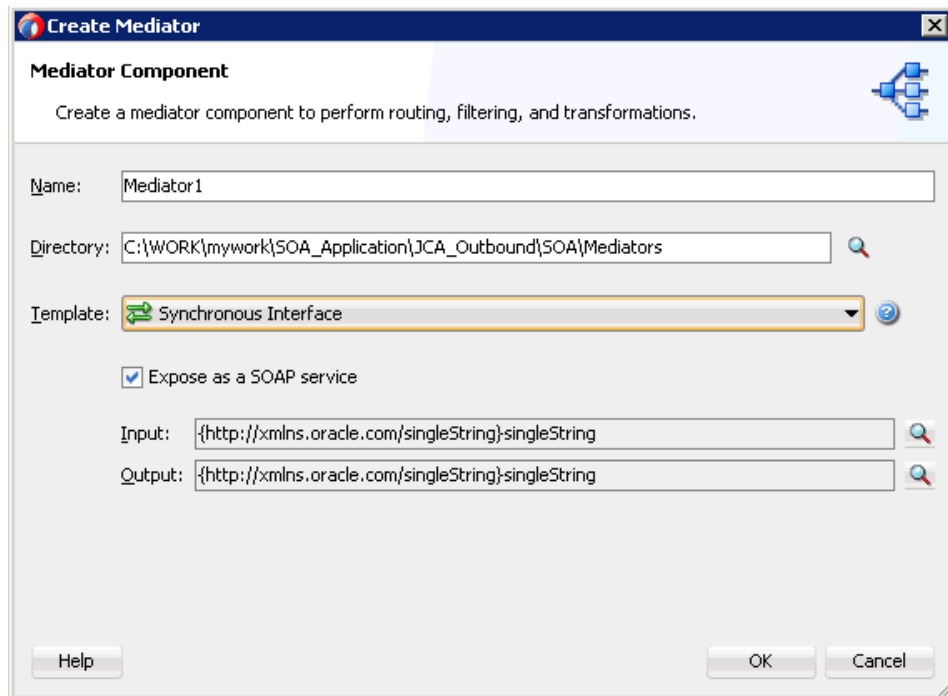
7.2.2.2 Configuring an Outbound Mediator Process Component

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 7-2](#).

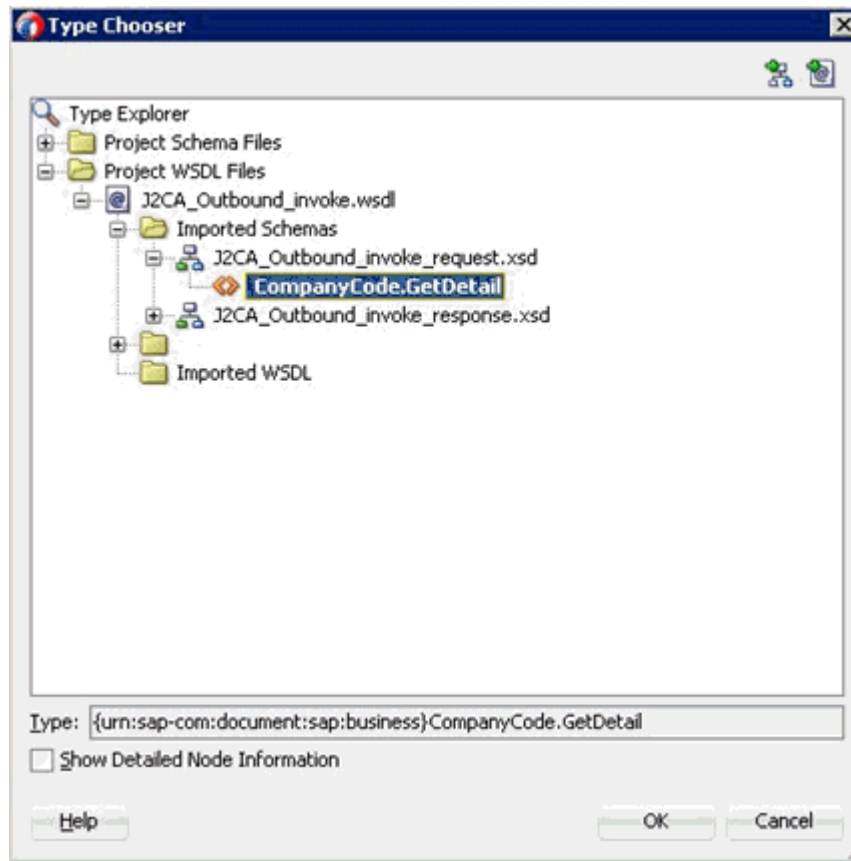
Figure 7-2 Create Mediator Dialog



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.

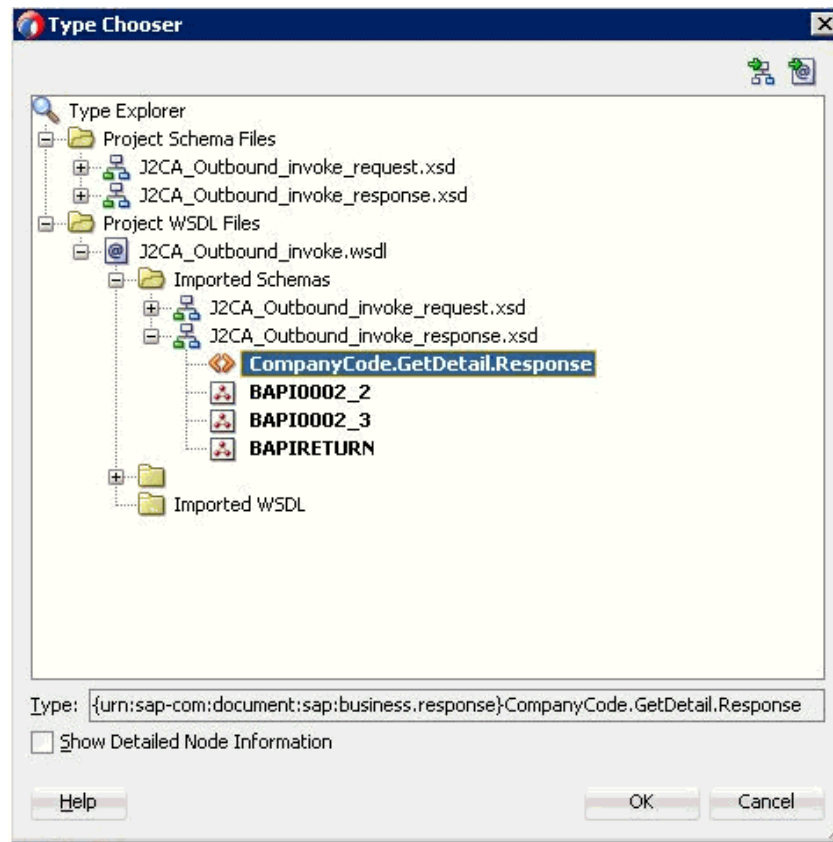
The Type Chooser dialog is displayed, as shown in [Figure 7-3](#).

Figure 7-3 Type Chooser Dialog



5. Expand **Project WSDL Files**, **J2CA_Outbound_invoke.wSDL**, **Imported Schemas**, **J2CA_Outbound_invoke_request.xsd**, and select **CompanyCode.GetDetail**.
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 7-4](#).

Figure 7-4 Type Chooser Dialog

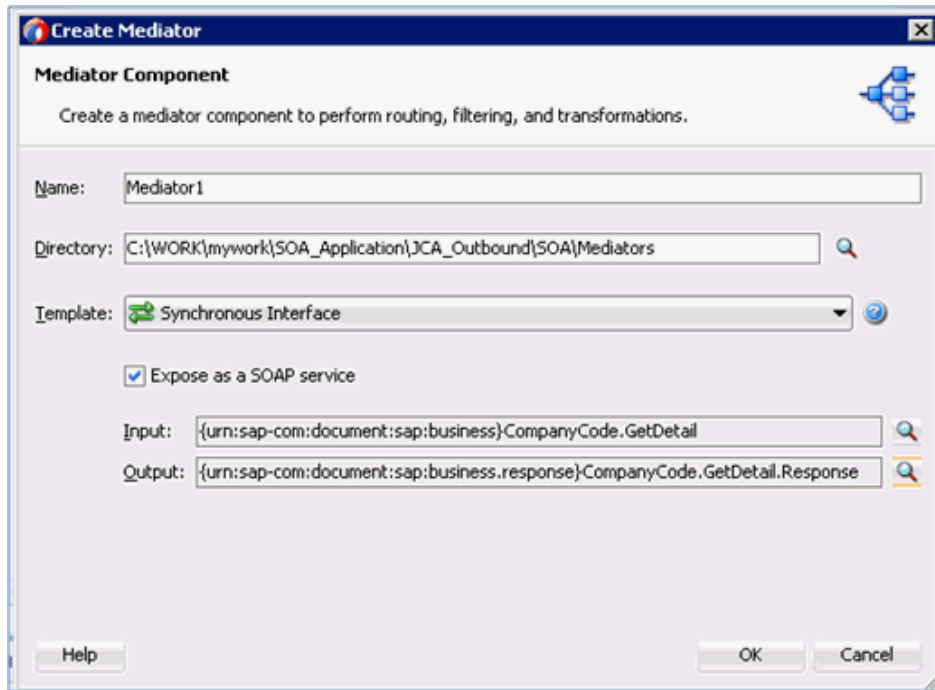


- Expand **Project WSDL Files**, **J2CA_Outbound_invoke.wsdl**, **Imported Schemas**, **J2CA_Outbound_invoke_response.xsd**, and select **CompanyCode.GetDetailResponse**.

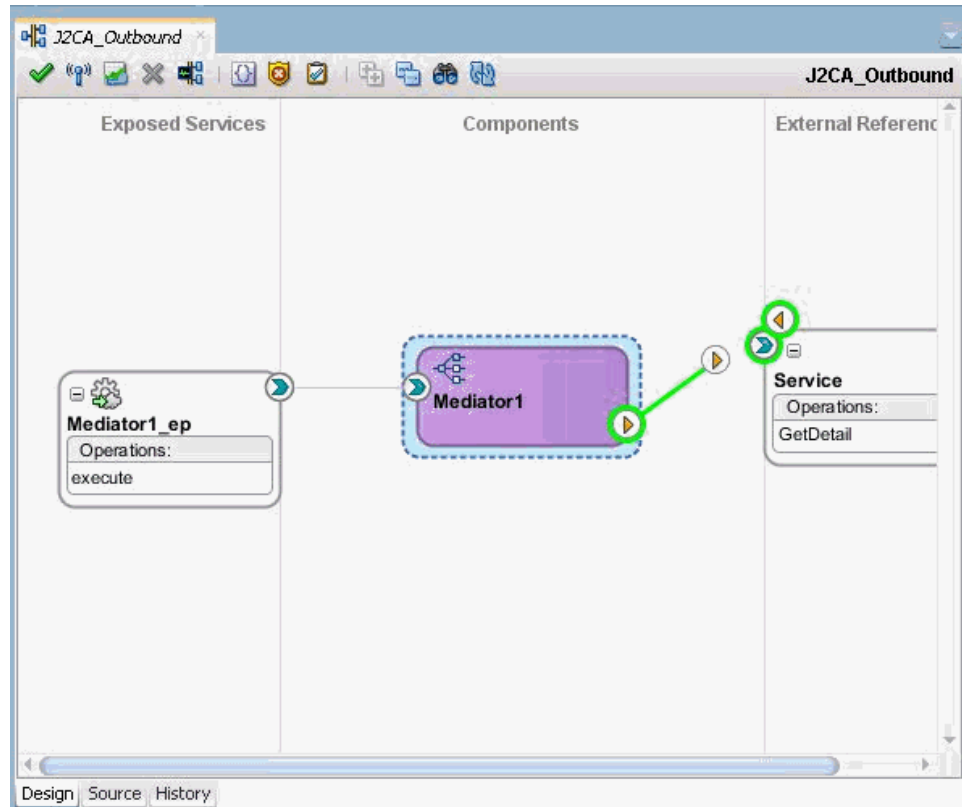
- Click **OK**.

You are returned to the Create Mediator dialog, as shown in [Figure 7-5](#).

Figure 7-5 Create Mediator Dialog



10. Click **OK**.
11. Create a connection between the outbound Mediator process component and the third party adapter service component, as shown in [Figure 7-6](#).

Figure 7-6 Created Connection

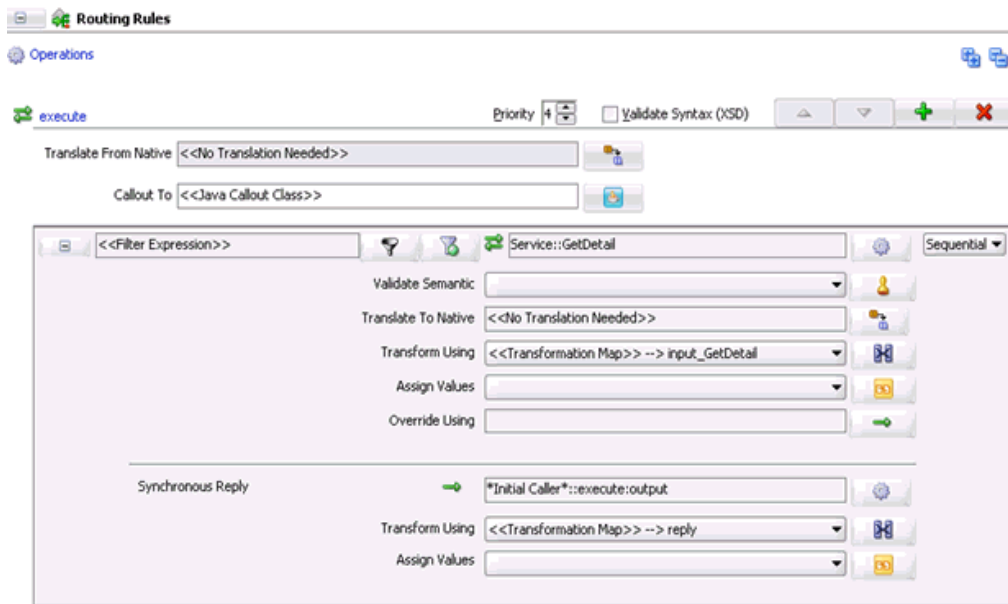
You are now ready to configure the routing rules.

7.2.2.3 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 7-7](#).

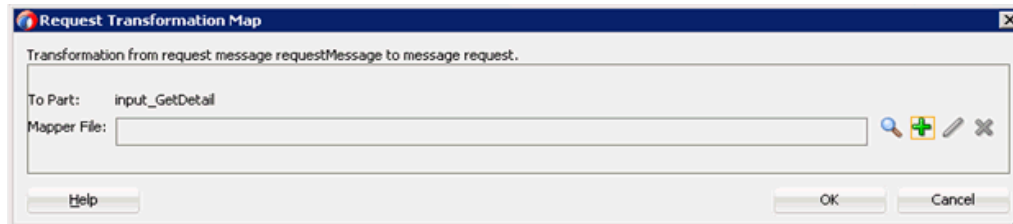
Figure 7-7 Routing Rules Dialog



- In the <<Filter Expression>> area, click the icon to the right of the Transform Using field.

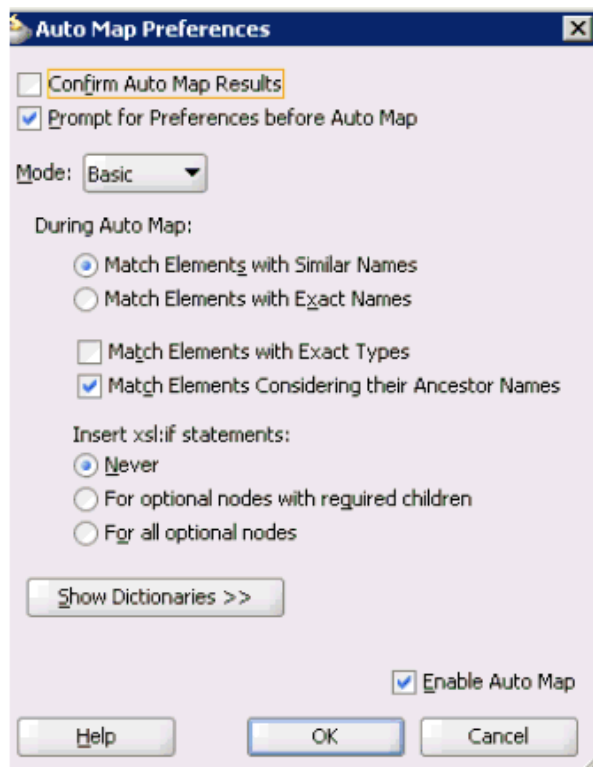
The Request Transformation Map dialog is displayed, as shown in [Figure 7-8](#).

Figure 7-8 Request Transformation Map Dialog

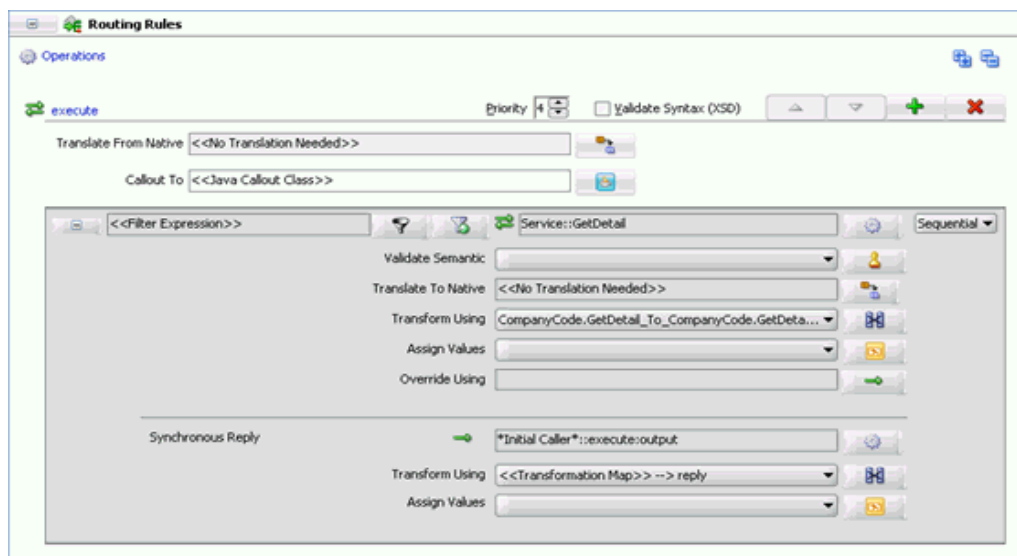


- Click the Add (+) icon.
- The Create Transformation Map page is displayed.
- Make sure the Type is selected as XSLT and click OK.
 - Click OK.
 - Map the `ns0:CompanyCode.GetDetail.Response` source element to the `ns0:CompanyCode.GetDetail.Response` target element.

The Auto Map Preferences dialog is displayed, as shown in [Figure 7-9](#).

Figure 7–9 Auto Map Preferences Dialog

7. Retain the default values and click **OK**.
8. Return to the Routing Rules dialog, as shown in [Figure 7–10](#).

Figure 7–10 Routing Rules Dialog

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**.

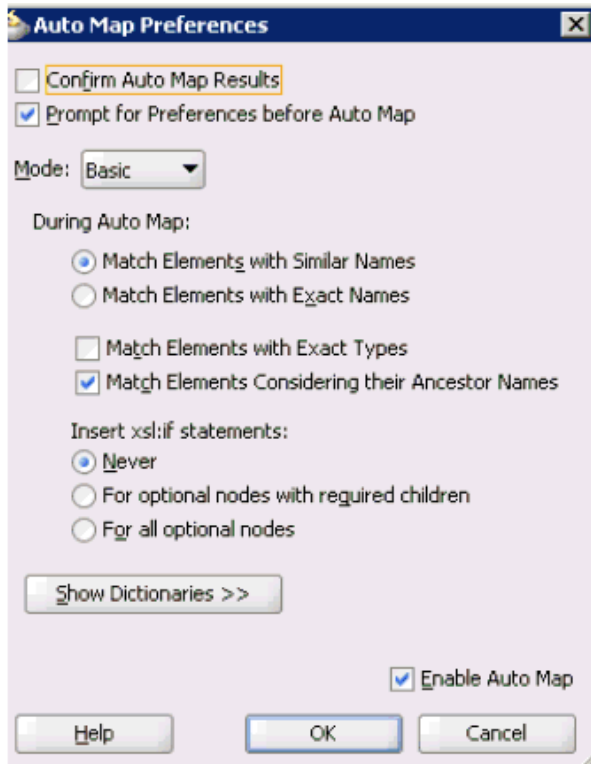
A mapping page is displayed.

12. Click **OK**.

13. Map the `ns0:CompanyCode.GetDetail.Response` source element to the `ns0:CompanyCode.GetDetail.Response` target element.

The Auto Map Preferences dialog is displayed, as shown in [Figure 7–11](#).

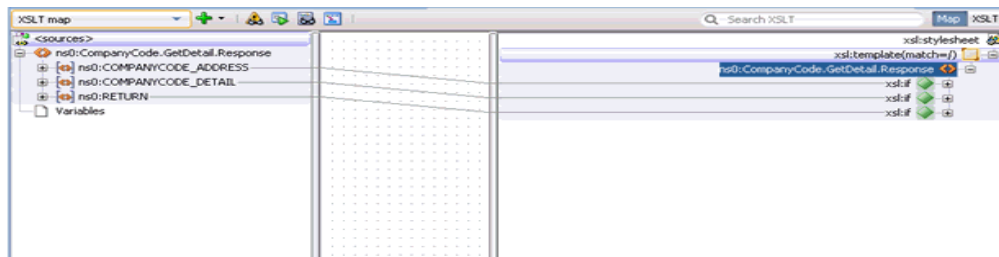
Figure 7–11 Auto Map Preferences Dialog



14. Retain the default values and click **OK**.

The mapping is completed, as shown in [Figure 7–12](#).

Figure 7–12 Completed Mapping



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

You are now ready to deploy the Mediator outbound process.

7.2.2.4 Adjusting for Known Deployment Issues With 12c

For more information on how to adjust for known deployment issues with 12c, see [Section 6.4.3.3, "Adjusting for Known Deployment Issues With 12c"](#) on page 6-27.

7.2.3 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 6.4.4, "Deploying the BPEL Outbound Process"](#) on page 6-30.

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

For more information, see [Section 6.4.5, "Invoking the Input XML Document in the Oracle Enterprise Manager Console"](#) on page 6-33.

7.3 Configuring a Mediator Inbound Process (J2CA Configuration)

This section describes how to configure a Mediator inbound process to your SAP R/3 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\SAP_Samples.zip\SAP_Samples\Mediator\J2CA\Inbound_
Project
```

This section contains the following topics:

- [Section 7.3.1, "Creating an Empty Composite for SOA"](#)
- [Section 7.3.2, "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 ["Generating WSDL for Event Integration"](#) on page 6-37.

7.3.1 Creating an Empty Composite for SOA

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

1. Create a new SOA application.
The Name your project page is displayed.
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 6.4.2, "Creating an Empty Composite for SOA"](#) on page 6-9.

7.3.2 Defining a Mediator Inbound Process

This section describes how to define a Mediator inbound process, which contains the following topics:

- [Section 7.3.2.1, "Configuring a Third Party Adapter Service Component"](#)
- [Section 7.3.2.2, "Configuring an Inbound Mediator Process Component With a File Adapter"](#)
- [Section 7.3.2.3, "Configuring the Routing Rules"](#)
- [Section 7.3.2.4, "Adjusting for Known Deployment Issues With 12c"](#)

7.3.2.1 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>\wsdl.s`
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>\wsdl.s

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 6.5.3.1, "Creating a Third Party Adapter Service Component"](#) on page 6-45.

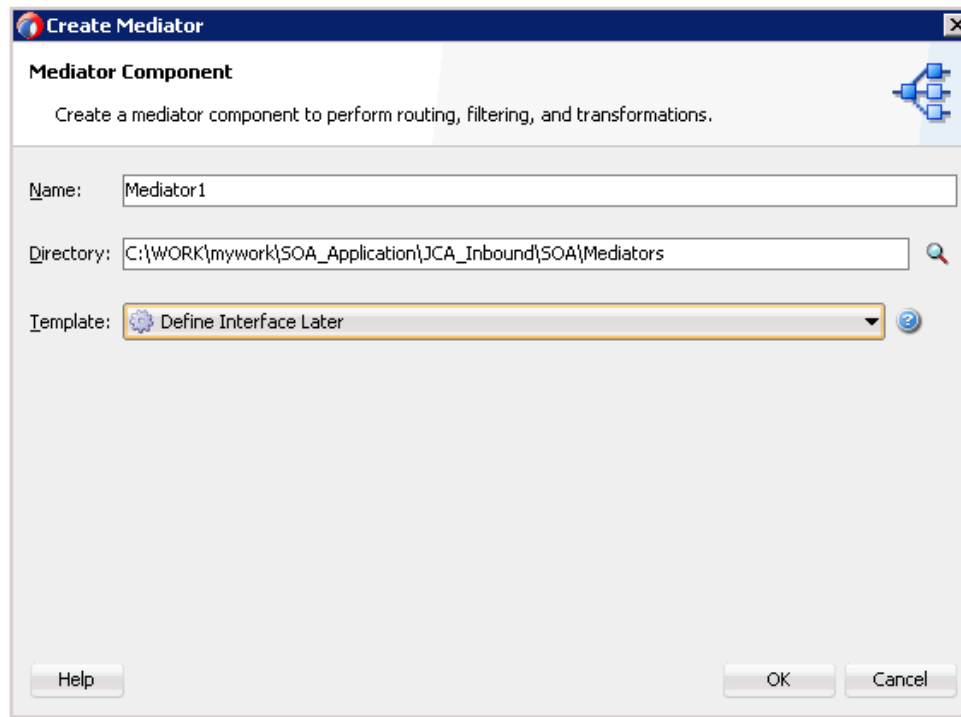
7.3.2.2 Configuring an Inbound Mediator Process Component With a File Adapter

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 7-13](#).

Figure 7-13 Create Mediator Dialog



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.

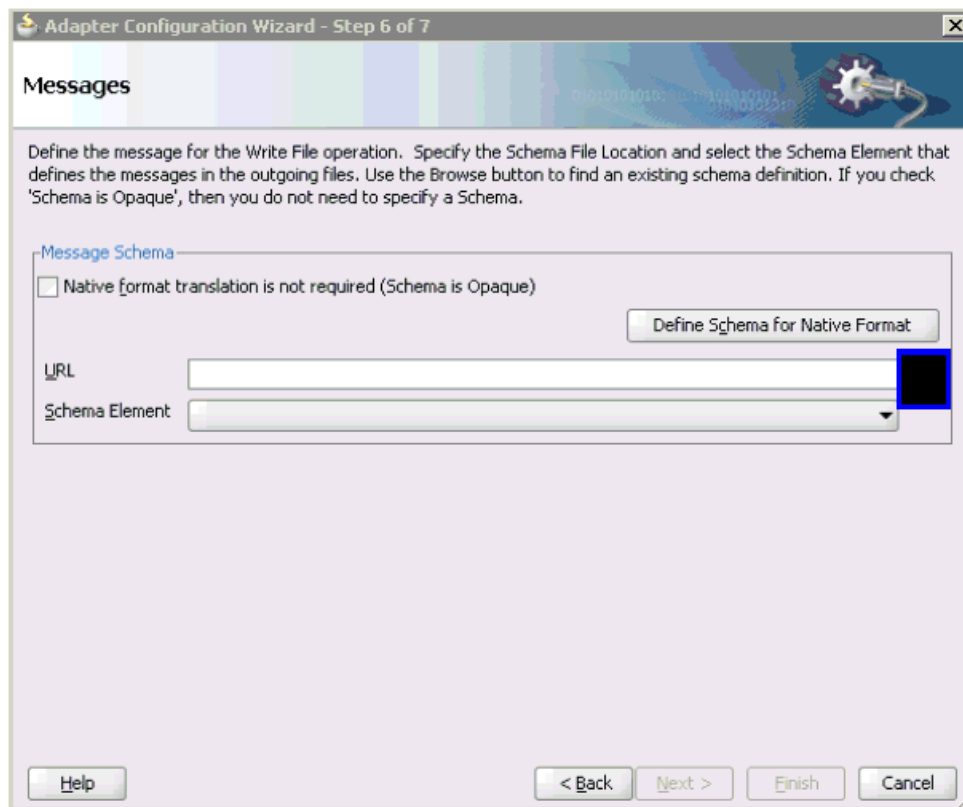
The File Adapter Configuration Wizard is displayed.
6. Type a name for the new File adapter and click **Next**.

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**.

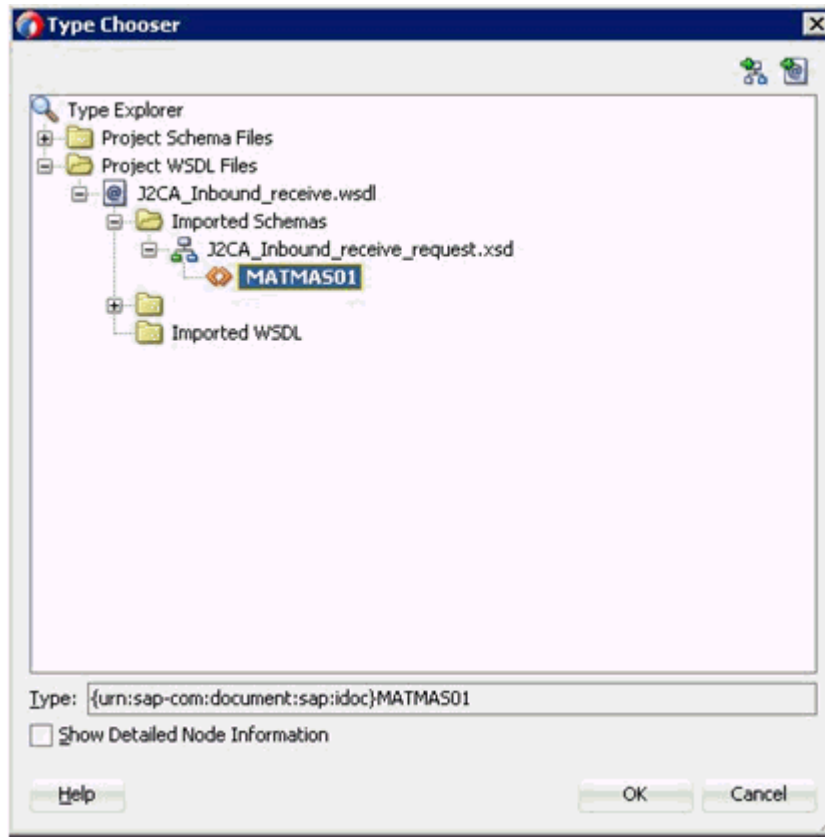
The Messages page is displayed, as shown in [Figure 7-14](#).

Figure 7–14 Messages Page

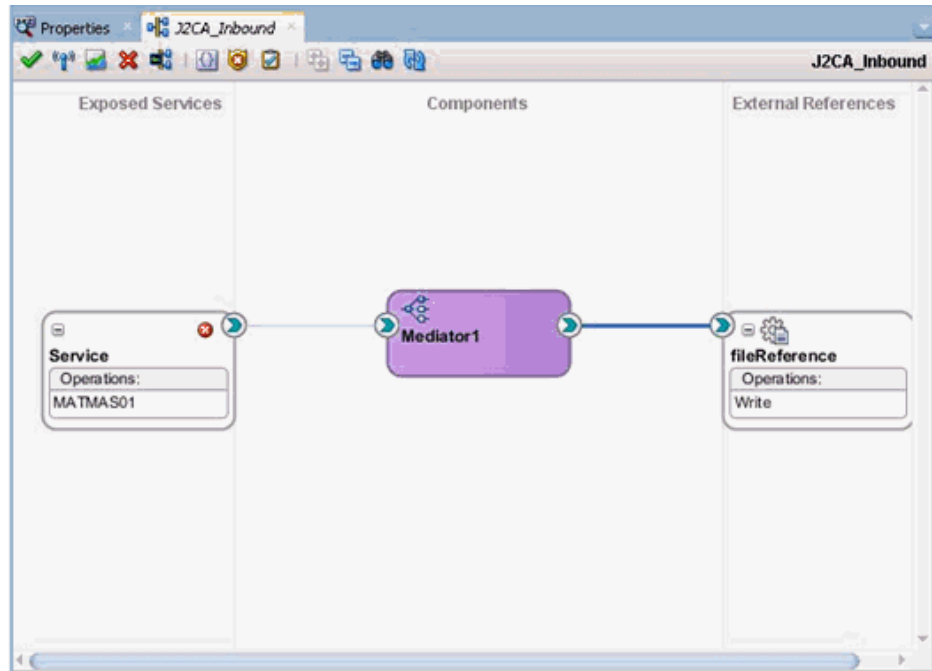
15. Click **Browse**, which is located to the right of the URL field.

The Type Chooser dialog is displayed, as shown in [Figure 7–15](#).

Figure 7–15 Type Chooser Dialog



16. Expand **Project WSDL Files**, **J2CA_Inbound_receive.wSDL**, **Imported Schemas**, **J2CA_Inbound_receive_request.xsd**, and select **MATMAS01**.
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 7–16](#).

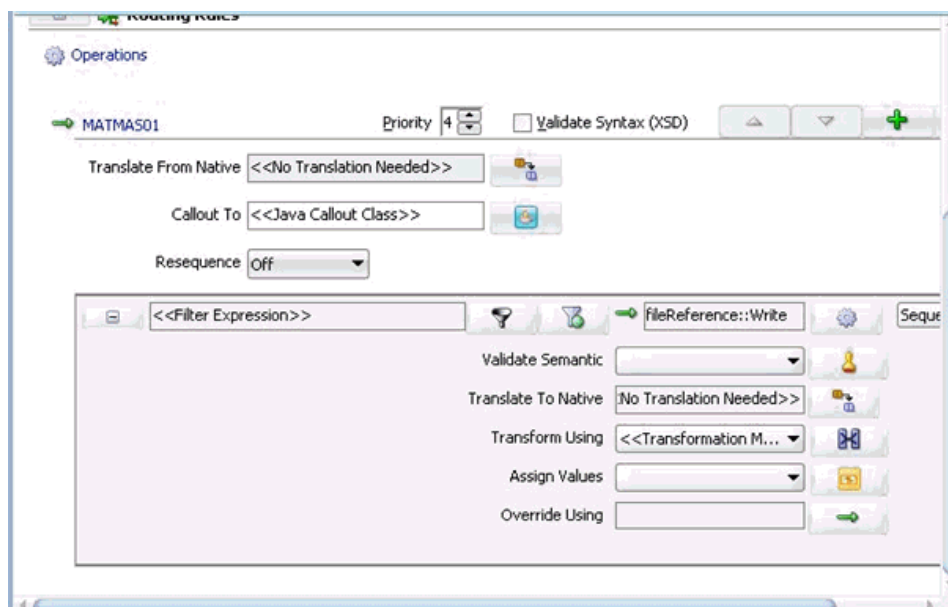
Figure 7–16 Created Connection

You are now ready to configure the routing rules.

7.3.2.3 Configuring the Routing Rules

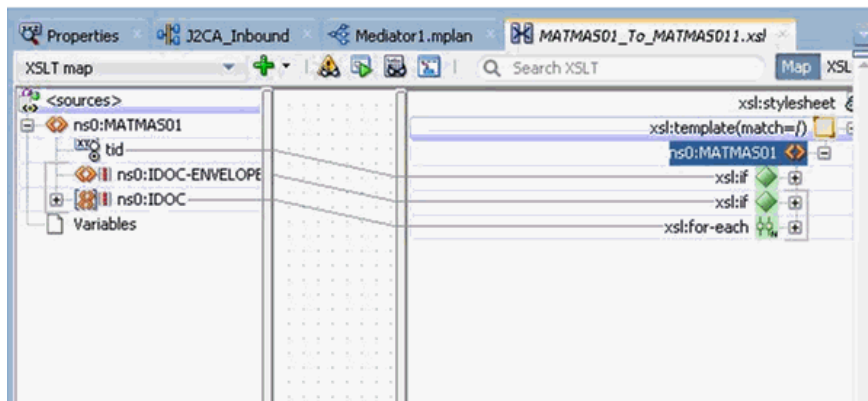
Perform the following steps to configure routing rules for the Mediator inbound process component:

1. Double-click the inbound Mediator process component in the Components page. The Routing Rules dialog is displayed, as shown in [Figure 7–17](#).

Figure 7–17 Routing Rules Dialog

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 7-18](#).

Figure 7-18 Mapping Page



5. Click OK.
6. Map the `ns0:MATMAS01` source element to the `ns0:MATMAS01` 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.

7.3.2.4 Adjusting for Known Deployment Issues With 12c

For more information on how to adjust for known deployment issues with 12c, see [Section 6.4.3.3, "Adjusting for Known Deployment Issues With 12c"](#) on page 6-27.

You are now ready to deploy the Mediator inbound process. You can follow the same procedure in ["Deploying the BPEL Inbound Process"](#) on page 6-53.

Once event messages are triggered through SAP GUI, output XML is received in the location that was specified for the File adapter component. For more information on triggering events in SAP R/3, see ["Triggering an Event in SAP R/3"](#) on page 6-54.

7.4 Configuring a Mediator Outbound Process (BSE Configuration)

This section describes how to configure a Mediator outbound process to your SAP R/3 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\SAP_Samples.zip\SAP_Samples\Mediator\BSE\Outbound_
Project
```

This section contains the following topics:

- [Section 7.4.1, "Creating an Empty Composite for SOA"](#)
- [Section 7.4.2, "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 6.7.1, "Generating a WSDL File for Request and Response Services Using a Web Service"](#).

7.4.1 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 6.7.2, "Creating an Empty Composite for SOA"](#) on page 6-65.

7.4.2 Defining a Mediator Outbound Process

This section describes how to define a Mediator outbound process. The following topics are included:

- [Section 7.4.2.1, "Configuring a SOAP Service"](#)
- [Section 7.4.2.2, "Creating a Mediator Component"](#)
- [Section 7.4.2.3, "Configuring the Routing Rules"](#)

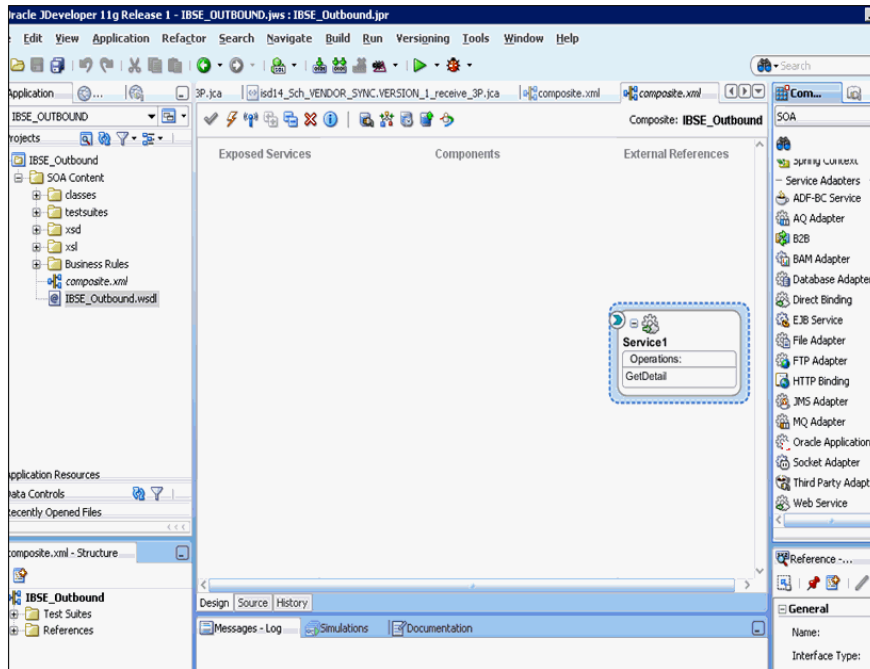
7.4.2.1 Configuring a SOAP Service

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, as shown in [Figure 7-19](#).

Figure 7–19 Created Web Service

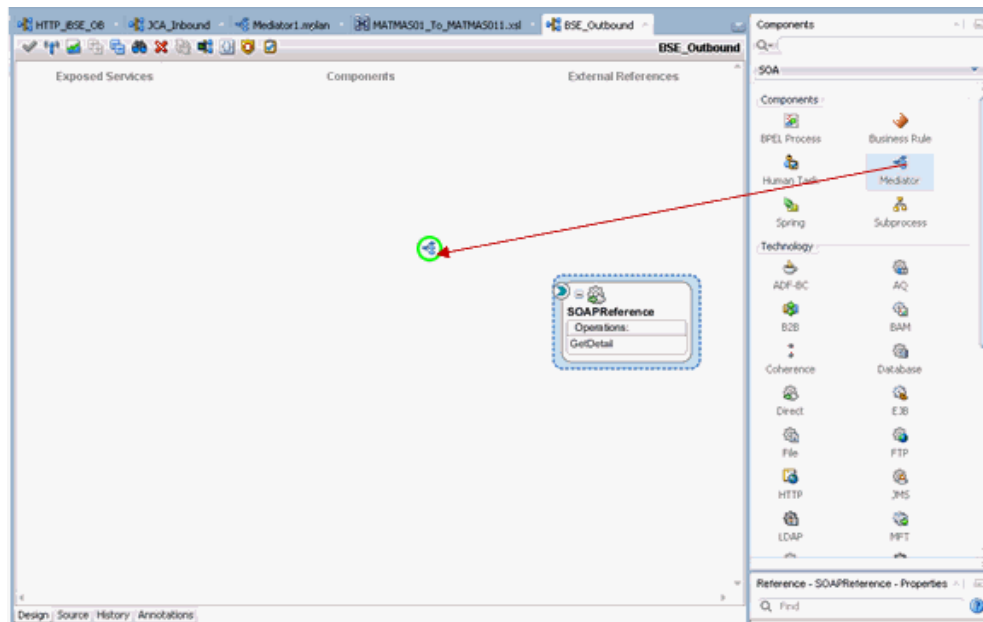


7.4.2.2 Creating a Mediator Component

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, as shown in [Figure 7–20](#).

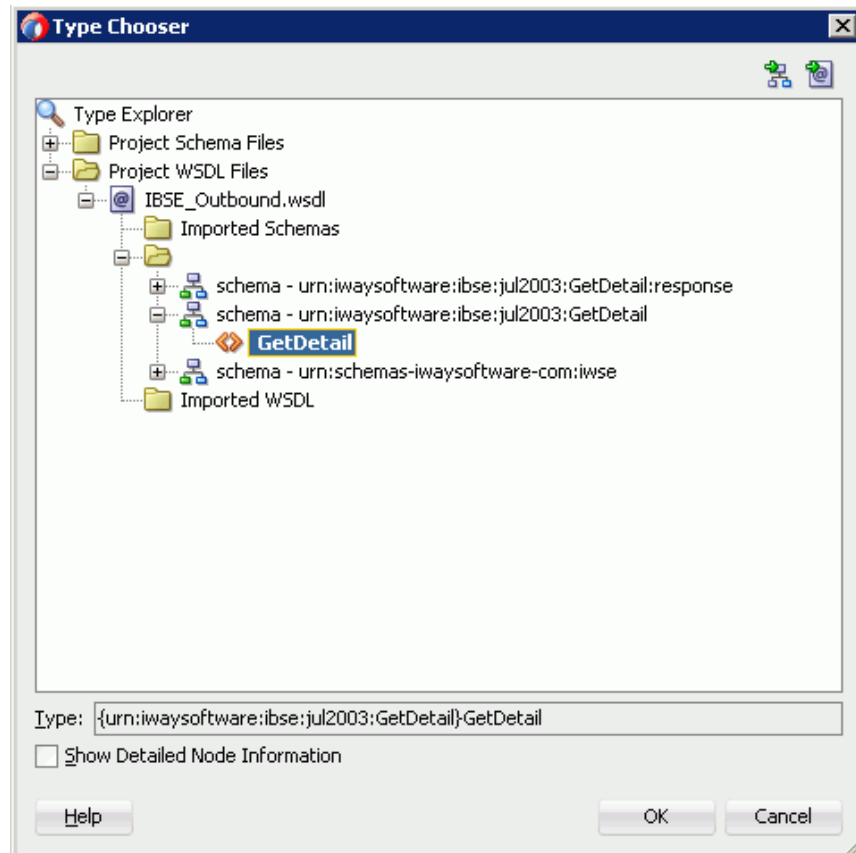
Figure 7–20 Mediator Component Dialog



2. In the Name field, enter a name to identify the new outbound Mediator process component.

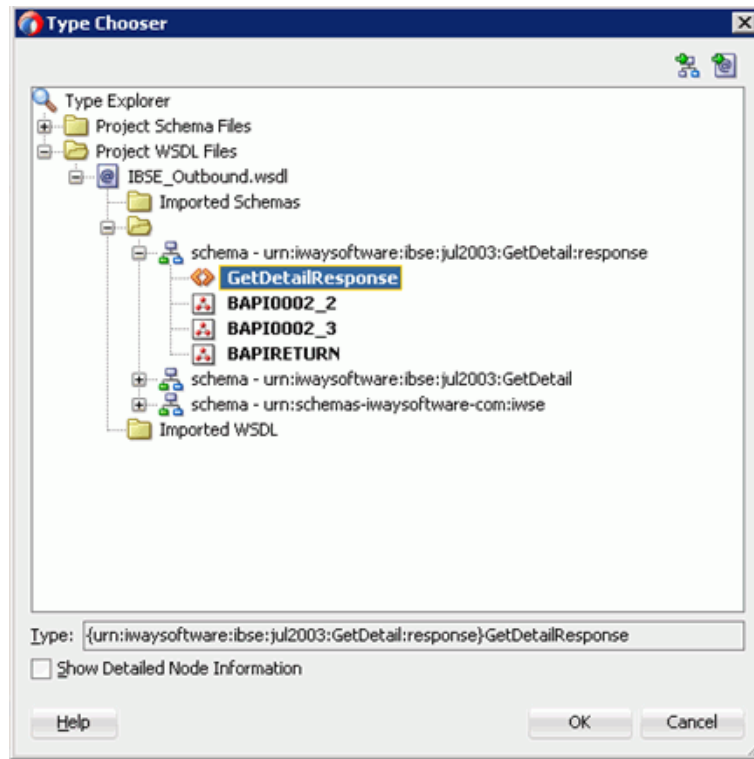
3. From the Template drop-down list, select **Synchronous Interface**.
4. Click the **Browse** icon, which is located to the right of the Input field, to select the associated XML request schema file.
5. In the Type Chooser dialog, expand **Project WSDL Files**, select **IBSE_Outbound.wsdl**, and click **GetDetail**, as shown in [Figure 7-21](#).

Figure 7-21 Type Chooser Dialog



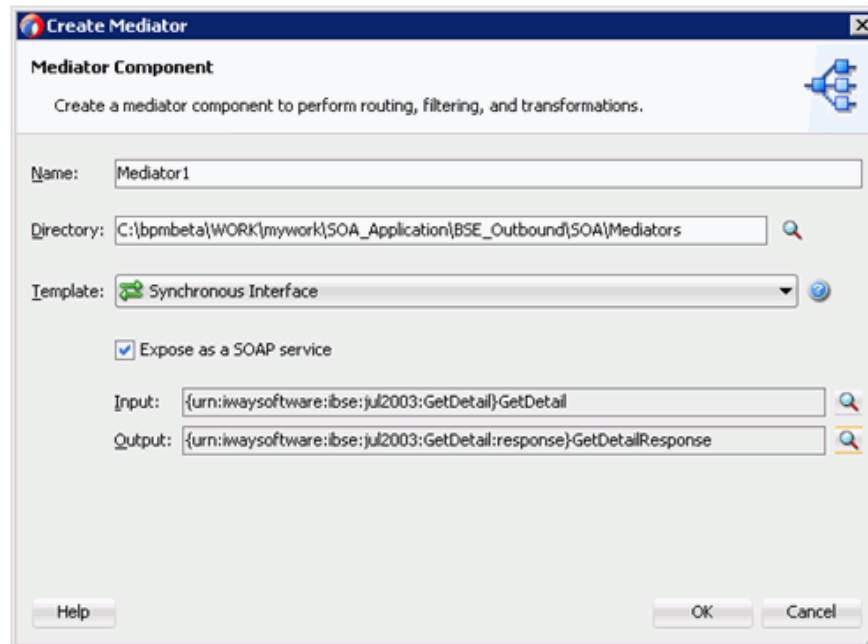
6. Click **OK**.
7. Click the **Browse** icon, which is located to the right of the Output field, to select the associated XML response schema file.
8. In The Type Chooser dialog, expand **Project WSDL Files**, select **IBSE_Outbound.wsdl**, and click **GetDetailResponse**, as shown in [Figure 7-22](#).

Figure 7–22 Type Chooser Dialog



9. Click OK, as shown in [Figure 7–23](#).

Figure 7–23 Create Mediator Dialog

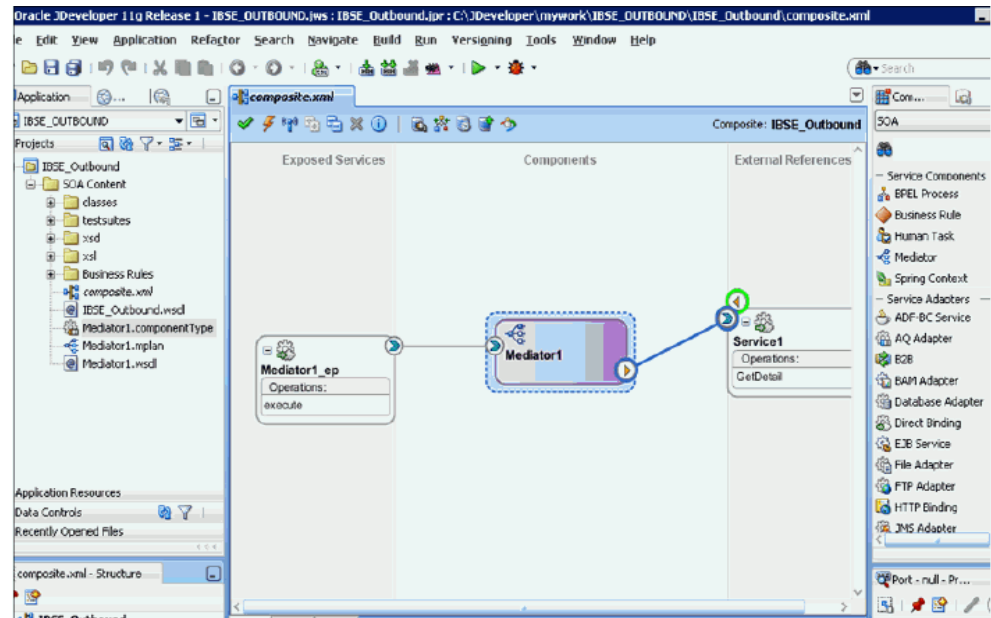


10. Click OK.

The Mediator component is created and displayed.

11. Create a connection between the **Mediator** component and the **SOAP service** component, as shown in [Figure 7–24](#).

Figure 7–24 Created Connection

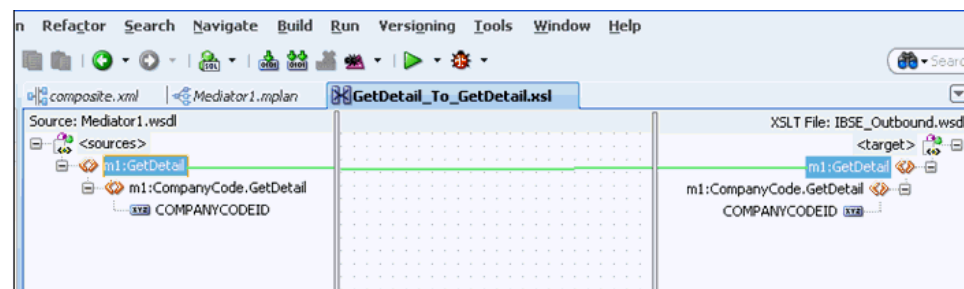


7.4.2.3 Configuring the Routing Rules

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:GetDetail** source element to the **ns0:GetDetail** target element, as shown in [Figure 7–25](#).

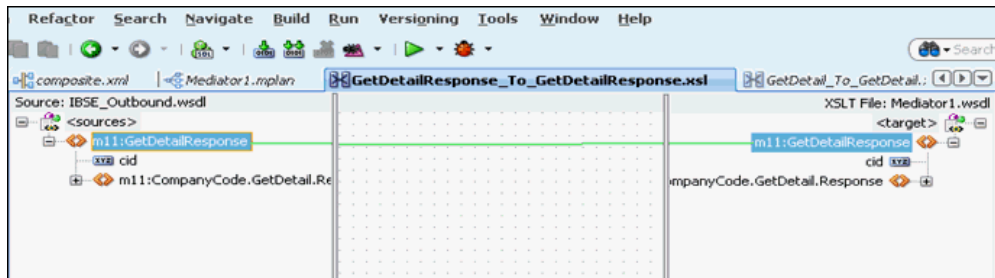
Figure 7–25 GetDetail_To_GetDetail.xsl Tab



6. In the displayed Auto Map Preferences window, retain the default values and click **OK**.

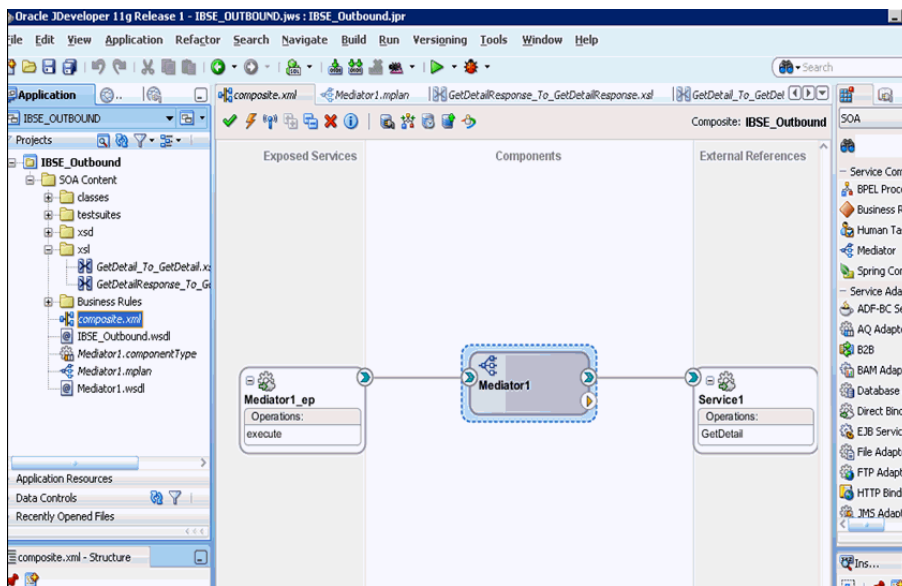
7. In the Synchronous Reply area, click the icon to the right of the Transform Using field.
8. In the displayed Reply Transformation Map window, click the Add (+) icon and make sure the Type is selected as XSLT in the Create Transformation Map dialog box, and then click OK.
9. Map the `ns0:GetDetailResponse` source element to the `ns0:GetDetailResponse` target element, as shown in [Figure 7–26](#).

Figure 7–26 Source and Target Elements



10. In the displayed Auto Map Preferences window, retain the default values and click OK.
11. Click the **Save All** icon in the menu bar to save the new outbound Mediator component that was configured, as shown in [Figure 7–27](#).

Figure 7–27 Save All Icon



You are now ready to deploy the Mediator IBSE outbound process. You can follow the same procedure found in [Section 7.2.3, "Deploying the Mediator Outbound Process"](#).

Once deployed, you can invoke the input XML, as defined in [Section 7.2.4, "Invoking the Input XML Document in the Oracle Enterprise Manager Console"](#).

Integration With BPM Service Components in the Oracle SOA Suite

Oracle Application Adapter for SAP R/3 integrates seamlessly with 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 8.1, "Overview"](#)
- [Section 8.2, "Deployment of Adapter"](#)
- [Section 8.3, "Configuring a New Application Server Connection"](#)
- [Section 8.4, "Designing an Outbound BPM Process Using Transformations for Service Integration \(J2CA Configuration\)"](#)
- [Section 8.5, "Designing an Inbound BPM Process Using Transformations for Event Integration \(J2CA Configuration\)"](#)
- [Section 8.6, "Working With Synchronous SAP R/3 Events Using BPM \(J2CA Configuration\)"](#)
- [Section 8.7, "Designing an Outbound BPM Process Using Transformations for Service Integration \(BSE Configuration\)"](#)

8.1 Overview

To integrate with Oracle BPM, Oracle Application Adapter for SAP R/3 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 4-18.

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.

8.2 Deployment of Adapter

During installation, Oracle Application Adapter for SAP R/3 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.

8.3 Configuring a New Application Server Connection

For more information on how to configure a new Application Server connection in Oracle JDeveloper, see [Section 6.3, "Configuring a New Application Server Connection"](#).

8.4 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\SAP_Samples.zip\SAP_Samples\BPM\J2CA\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 8.4.1, "Creating an Empty Composite for BPM"](#)
- [Section 8.4.2, "Defining a BPM Outbound Process"](#)
- [Section 8.4.3, "Adjusting for Known Deployment Issues With 12c"](#)
- [Section 8.4.4, "Deploying the BPM Outbound Process"](#)
- [Section 8.4.5, "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 ["Generating WSDL for Request/Response Service"](#) on page 6-8.

8.4.1 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**.

8.4.2 Defining a BPM Outbound Process

This section describes how to define a BPM outbound process, which contains the following topics:

- [Section 8.4.2.1, "Configuring a Third Party Adapter Service Component"](#)
- [Section 8.4.2.2, "Configuring an Outbound BPM Process Component"](#)
- [Section 8.4.2.3, "Creating a File Adapter for the Write Operation"](#)

8.4.2.1 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>\wsdl.s
```

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>\wsdl.s
```

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 6.4.3.1, "Configuring a Third Party Adapter Service Component"](#) on page 6-11.

8.4.2.2 Configuring an Outbound BPM Process Component

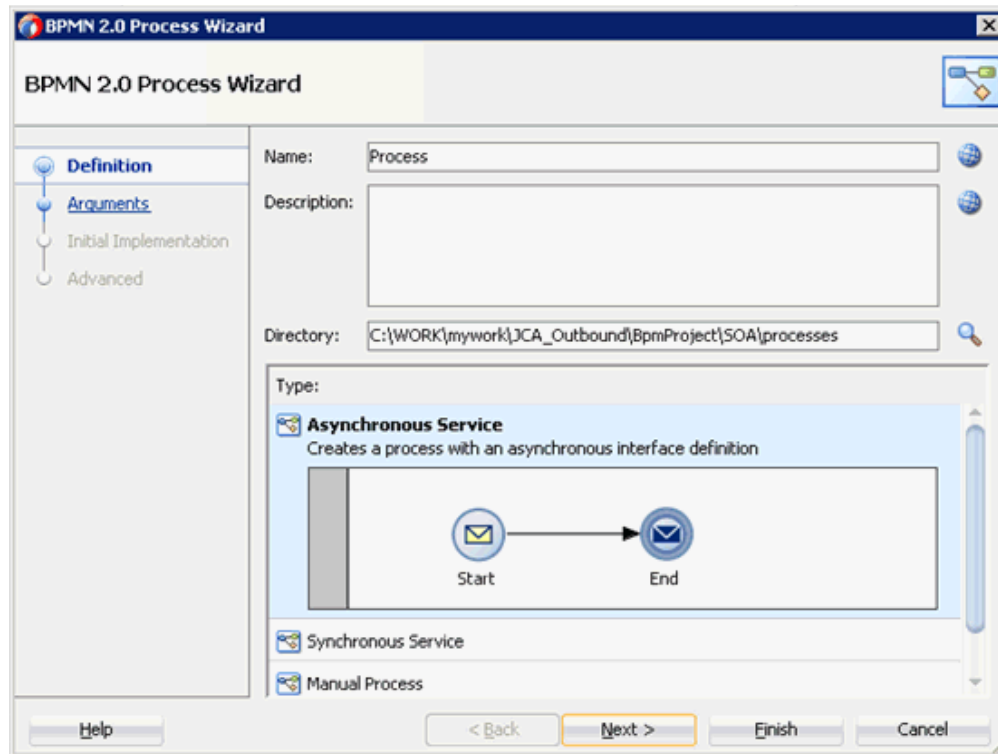
This section describes how to configure an outbound BPM process component.

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 8-1](#).

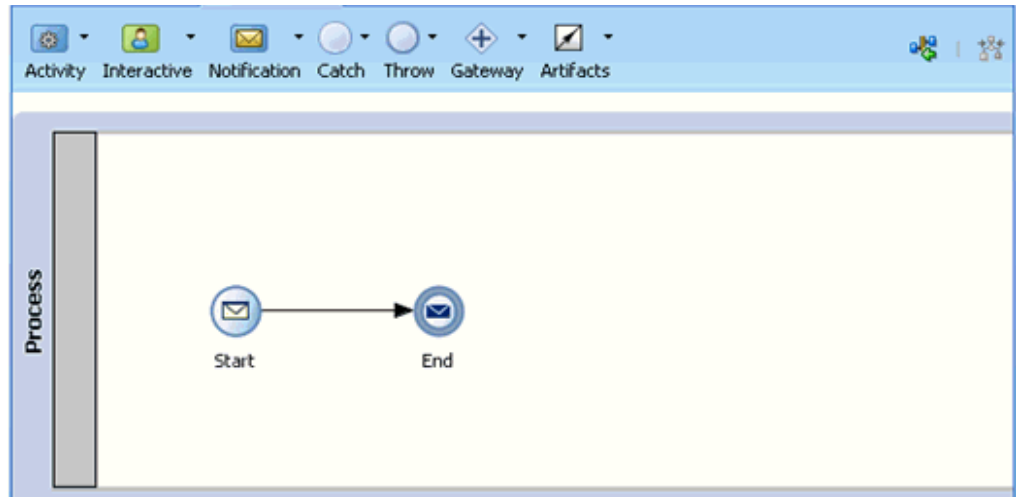
Figure 8-1 Create BPMN Process Dialog



2. Accept the default option that is selected under the Type area (Asynchronous Service) and click **Finish**.

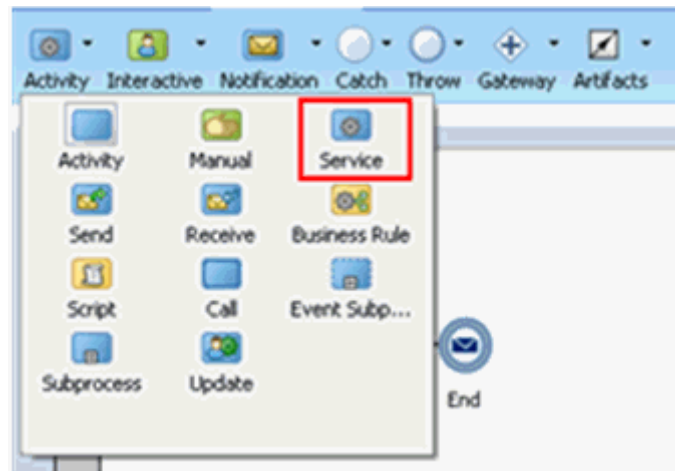
The BPMN process is displayed, as shown in [Figure 8-2](#).

Figure 8–2 BPMN Process



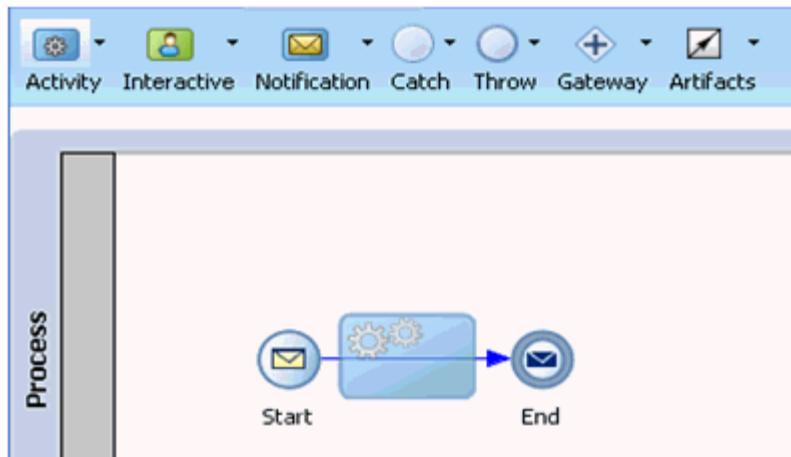
3. Click the **Activity** drop-down menu and select **Service**, as shown in [Figure 8–3](#).

Figure 8–3 Activity Drop-down Menu



4. Drop the **Service** icon on the wire between the Start and End event components, as shown in [Figure 8–4](#).

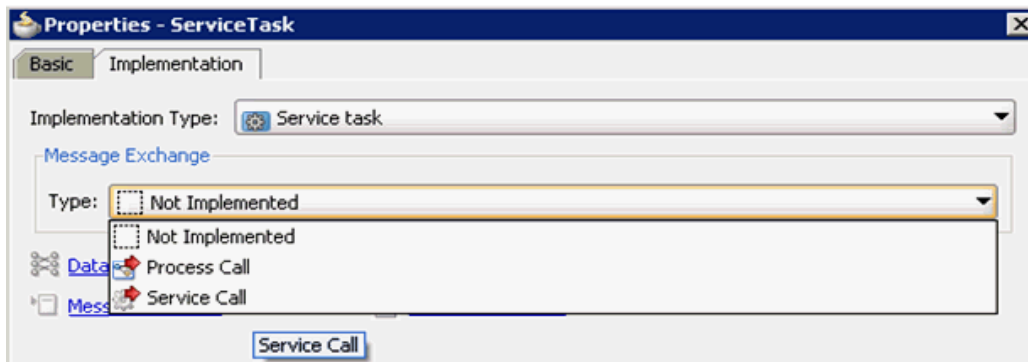
Figure 8–4 Activity Icon



The Properties - ServiceTask window is displayed.

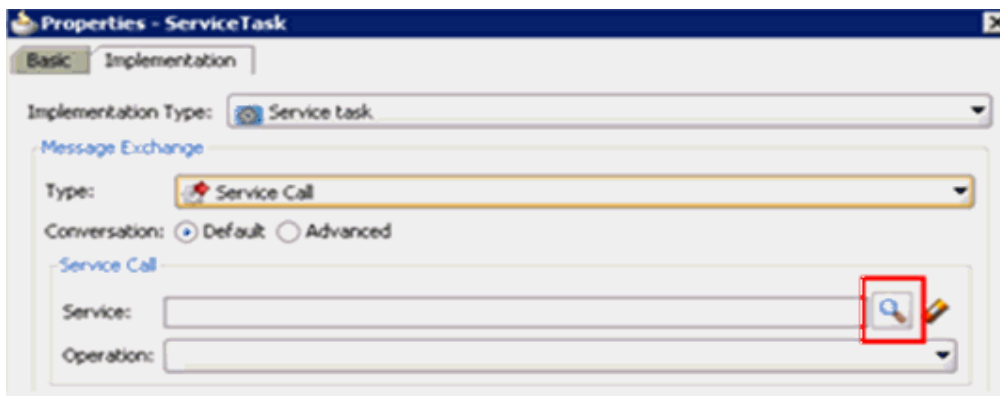
5. Click the **Implementation** tab.
6. Select **Service Call** from the Message Exchange Type list, as shown in [Figure 8–5](#).

Figure 8–5 Service Call

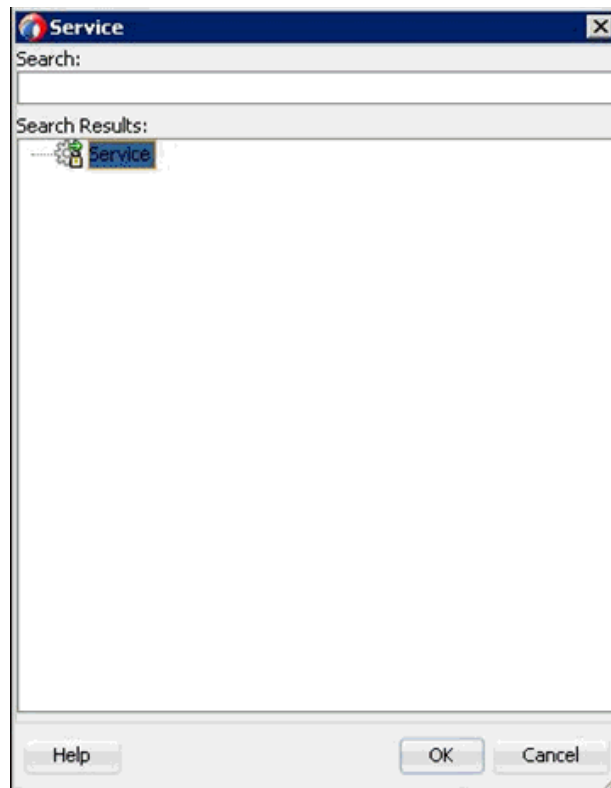


7. Click the Browse icon to the right of the Service field, as shown in [Figure 8–6](#).

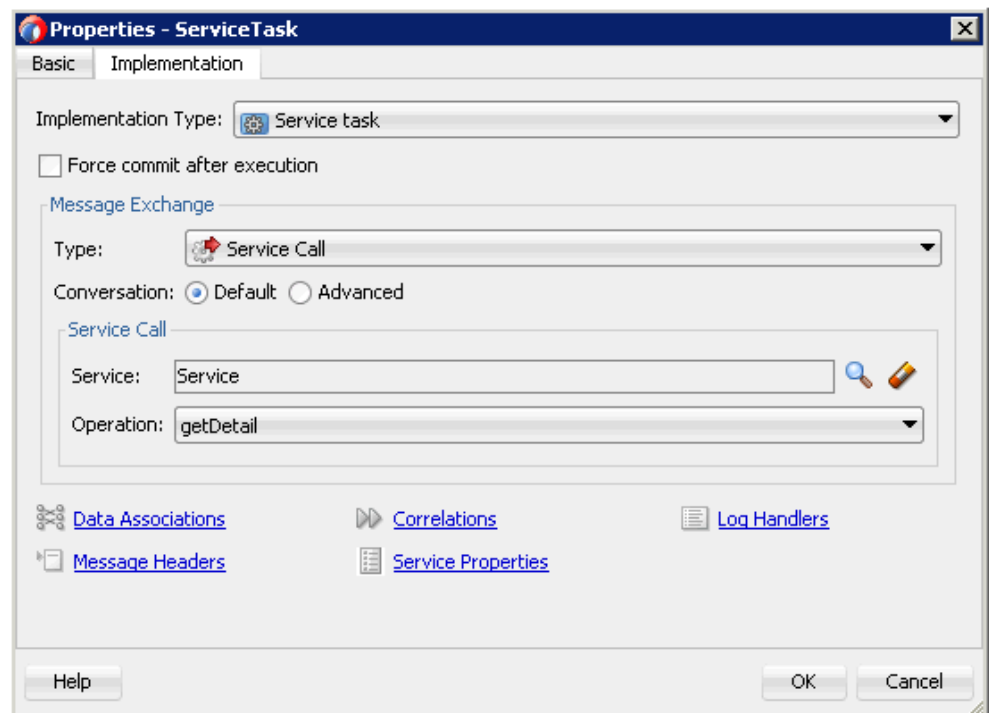
Figure 8–6 Browse Icon



The Service dialog is displayed, as shown in [Figure 8–7](#).

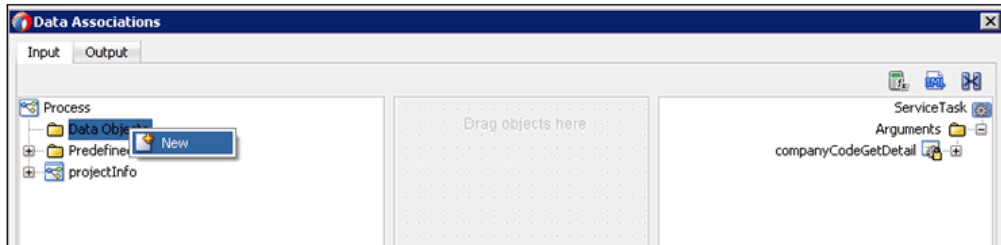
Figure 8-7 Service Dialog

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 8-8](#).

Figure 8-8 Properties - ServiceTask Dialog

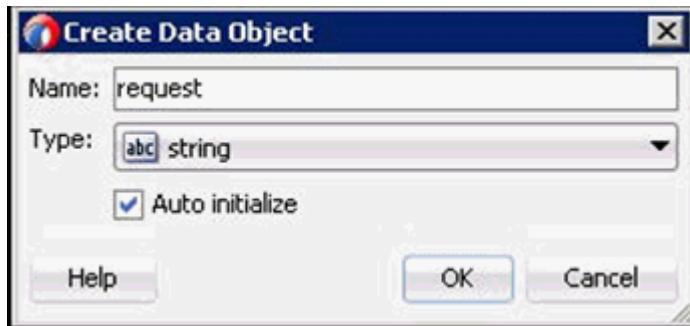
9. Click the **Data Associations** hyperlink.
The Data Associations dialog is displayed.
10. Right-click the **Data Objects** node in the left pane under Process, and select **New** as shown in [Figure 8-9](#).

Figure 8-9 New Option



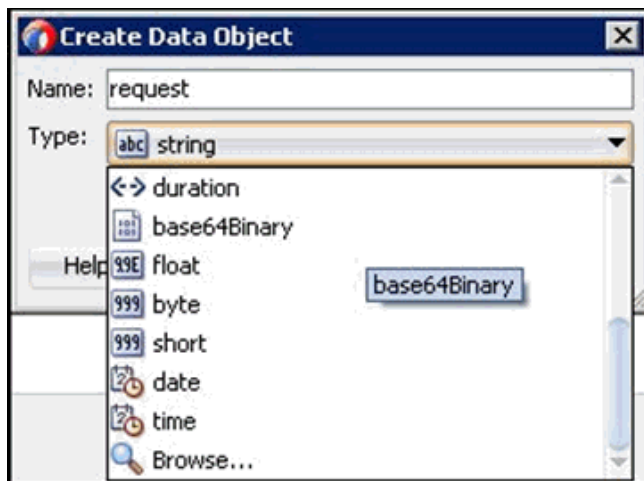
The Create Data Object dialog is displayed, as shown in [Figure 8-10](#).

Figure 8-10 Create Data Object Dialog

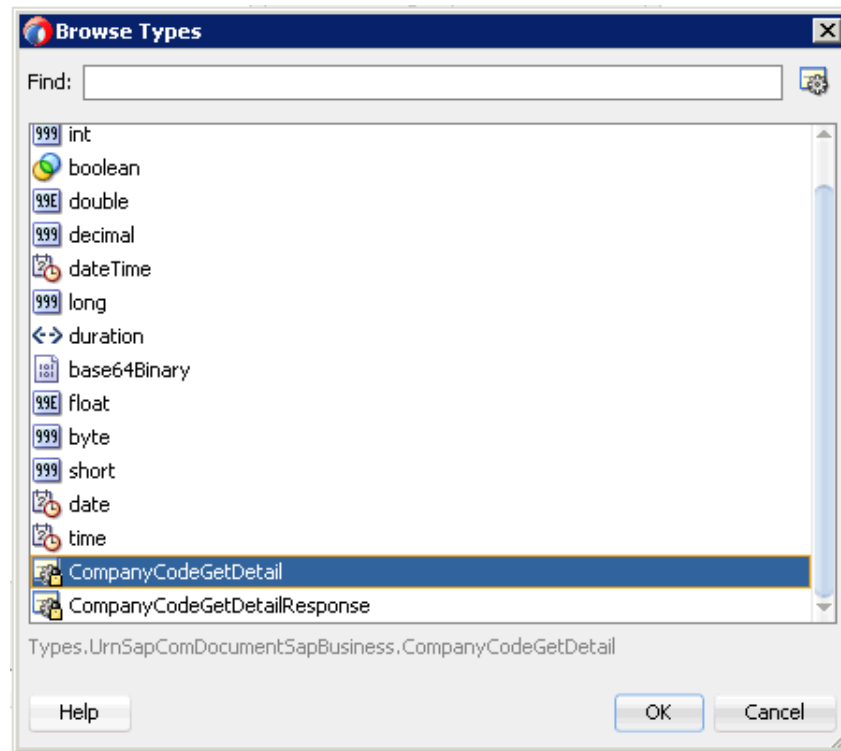


11. Enter a name in the Name field (for example, Request), click the drop-down button in the Type field, and select **Browse** from the list, as shown in [Figure 8-11](#).

Figure 8-11 Create Data Object Dialog

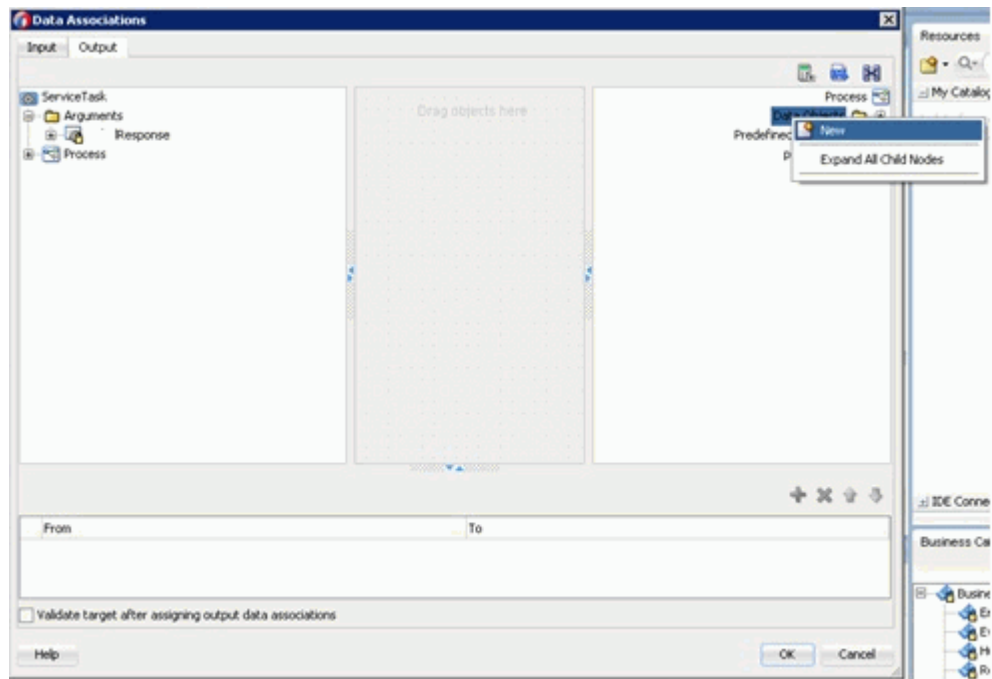


The Browse Types dialog is displayed, as shown in [Figure 8-12](#).

Figure 8–12 Browse Types Dialog

12. Select the first component (for example, `CompanyCodeGetDetail`) 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 8–13](#).

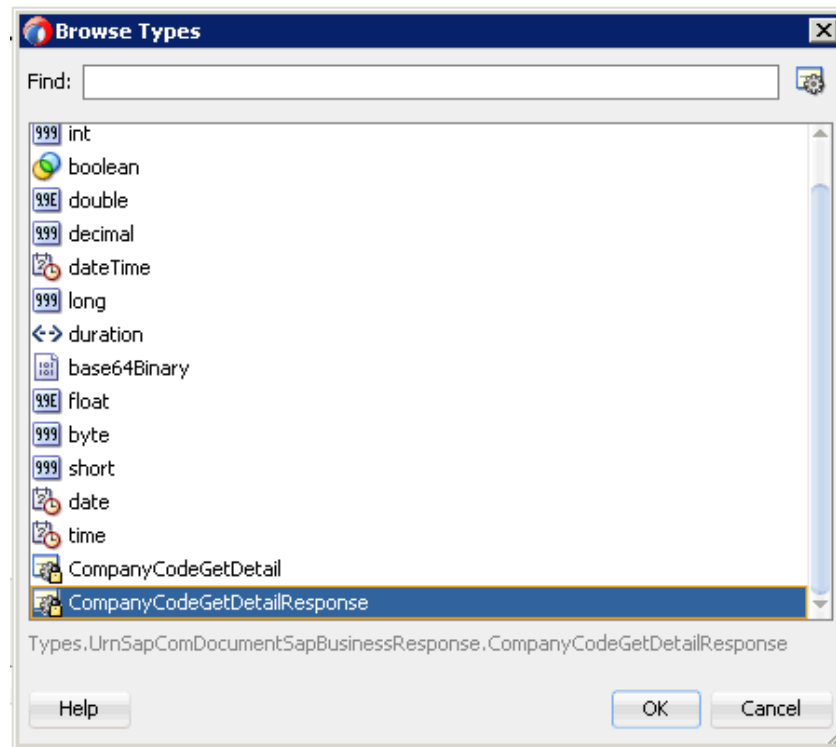
Figure 8–13 Data Associations Dialog



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 8–14](#).

Figure 8–14 Browse Types Dialog

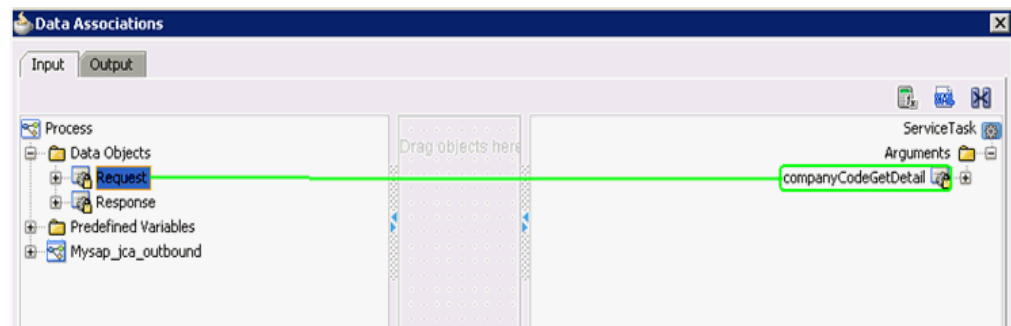
16. Select the second component (for example, `CompanyCodeGetDetailResponse`) and click **OK**.

You are returned to the Create Data Object dialog.

17. Click **OK**.

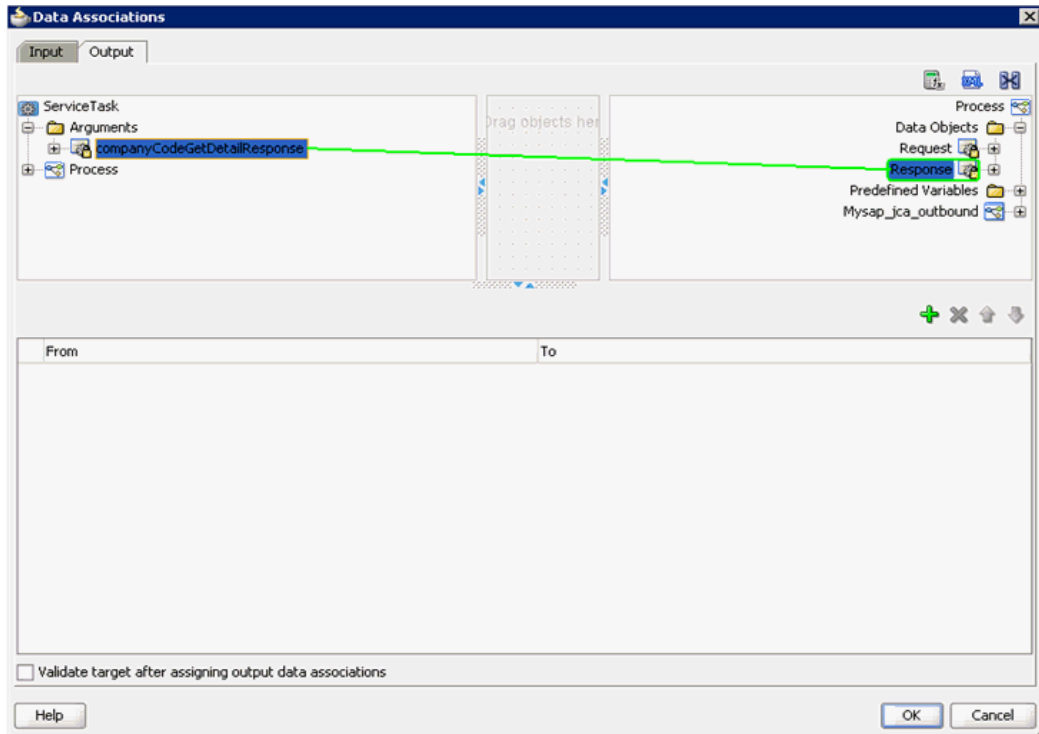
The Data Object (for example, `Response`) that has been created is displayed under the Process node in the Data Associations dialog.

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 `CompanyCodeGetDetail` under the Arguments node in the right pane, as shown in [Figure 8–15](#).

Figure 8–15 Request Data Object

19. Click on the **Output** tab and select `CompanyCodeGetDetailResponse` 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 8–16](#).

Figure 8–16 Response Data Object



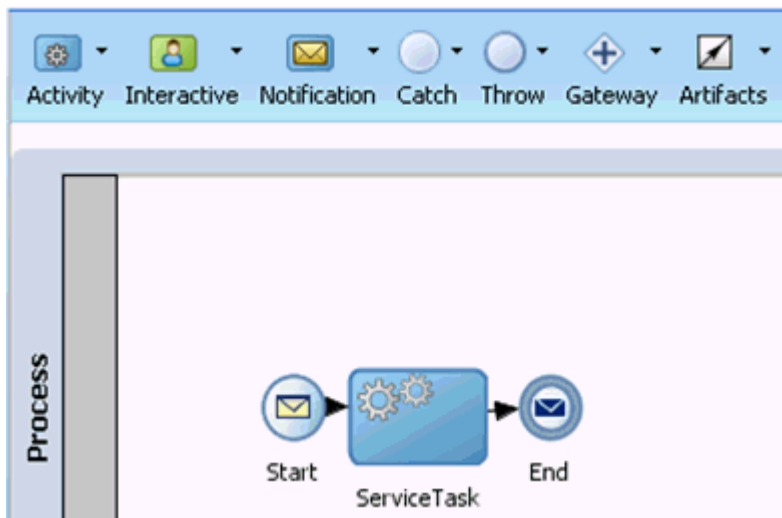
20. Click **OK**.

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 8–17](#).

Figure 8–17 Service Task

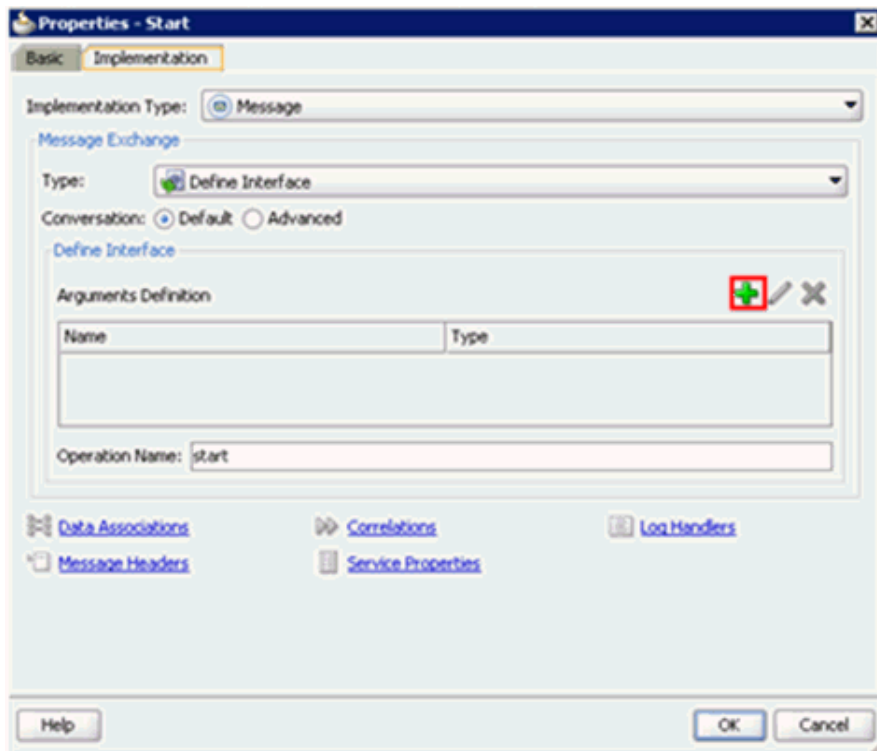


22. Save the process and double-click the Start event component.

The Properties - Start dialog is displayed, as shown in [Figure 8–18](#).

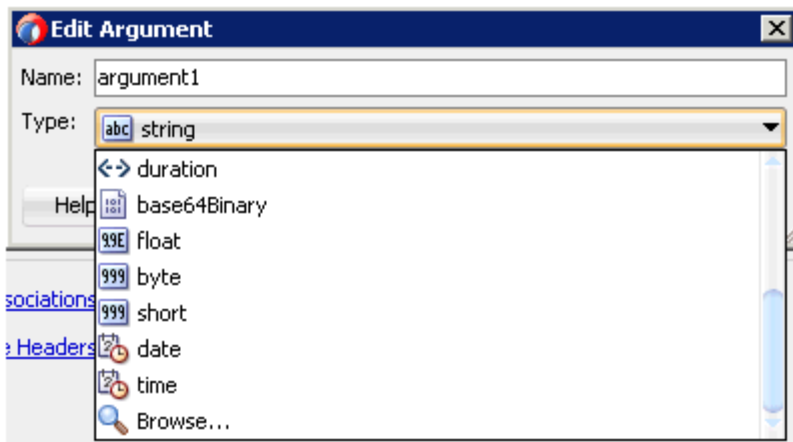
Figure 8–18 Properties - Start Dialog

23. Click the **Implementation** tab, as shown in [Figure 8–19](#).

Figure 8–19 Implementation Tab

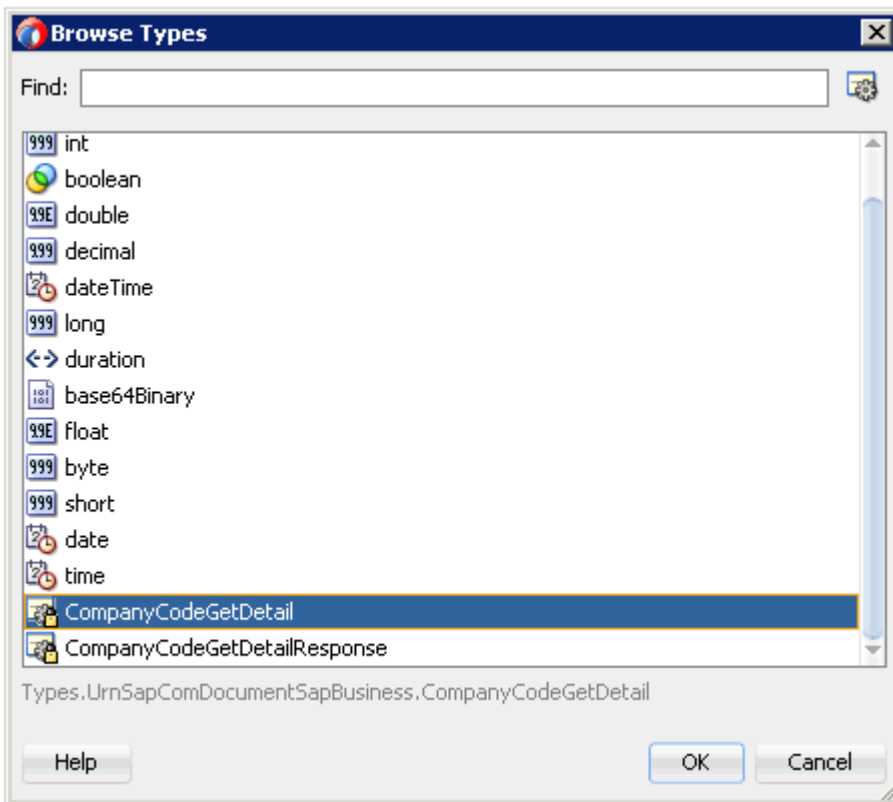
24. Click the **Plus** icon to the right of the Arguments Definition field.
The Create Argument dialog is displayed.
25. Enter a name in the Name field (by default, argument1), and then click the drop-down button in the Type field and select **Browse** from the list, as shown in [Figure 8–20](#).

Figure 8–20 Create Argument Dialog



The Browse Types dialog is displayed, as shown in [Figure 8–21](#).

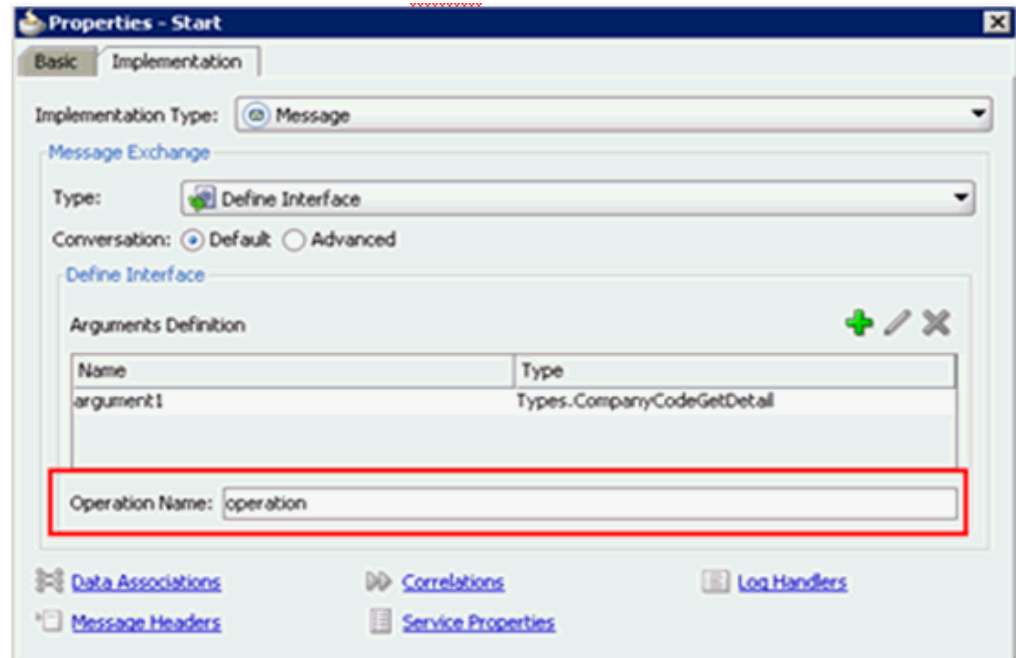
Figure 8–21 Browse Types Dialog



26. Select the first component (for example, CompanyCodeGetDetail) 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 8–22](#).

Note: This change is necessary to work with old BPM payloads.

Figure 8–22 Operation Name Field



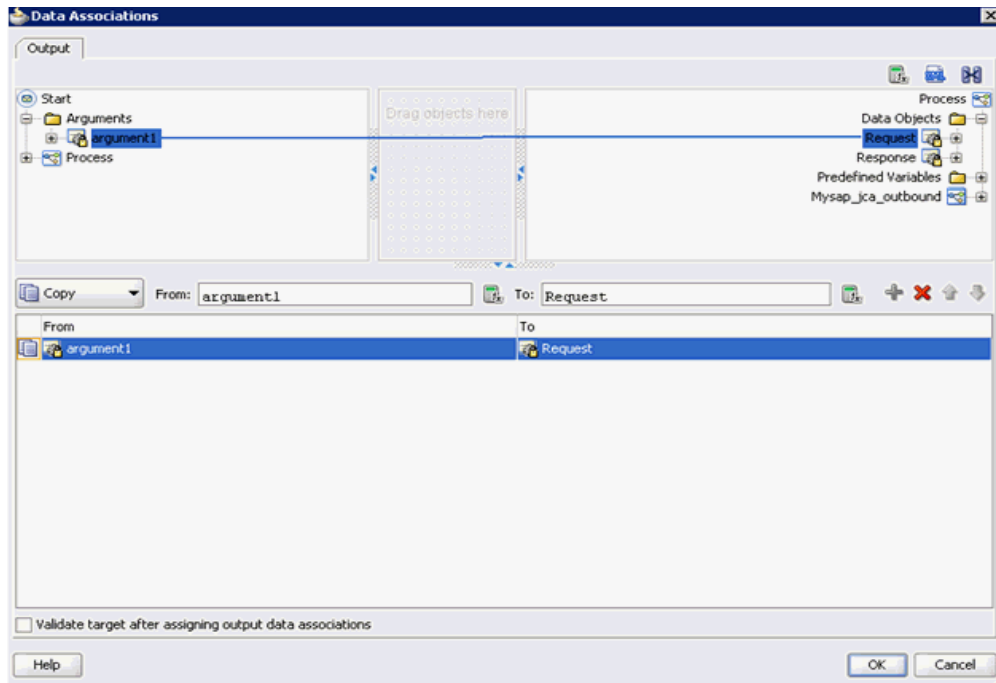
29. Click the **Data Associations** hyperlink.

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 8–23](#).

Figure 8–23 OK Button

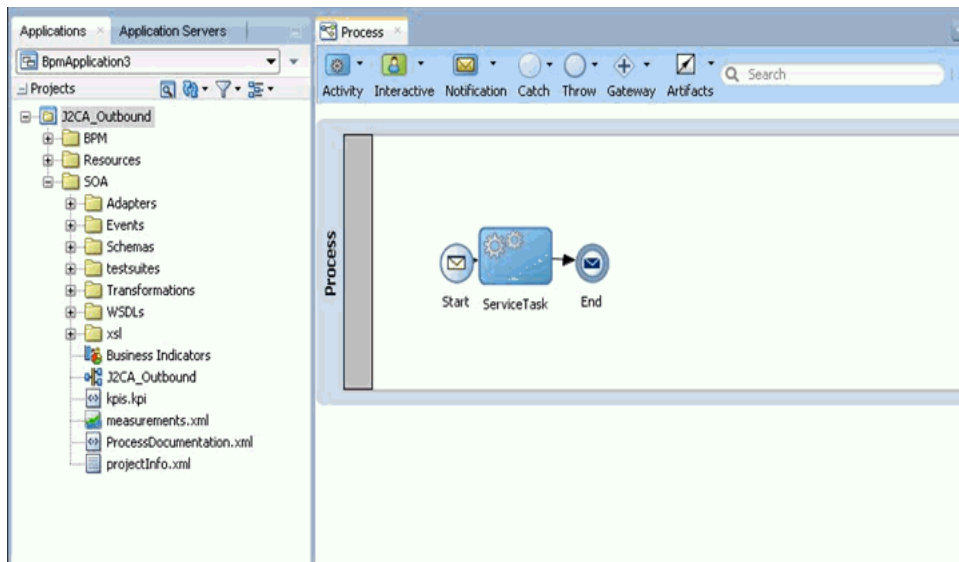


You are returned to the Properties - Start dialog.

32. Click **OK**.

You are returned to the Process workspace area, as shown in [Figure 8–24](#).

Figure 8–24 Process Workspace Area



33. 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.

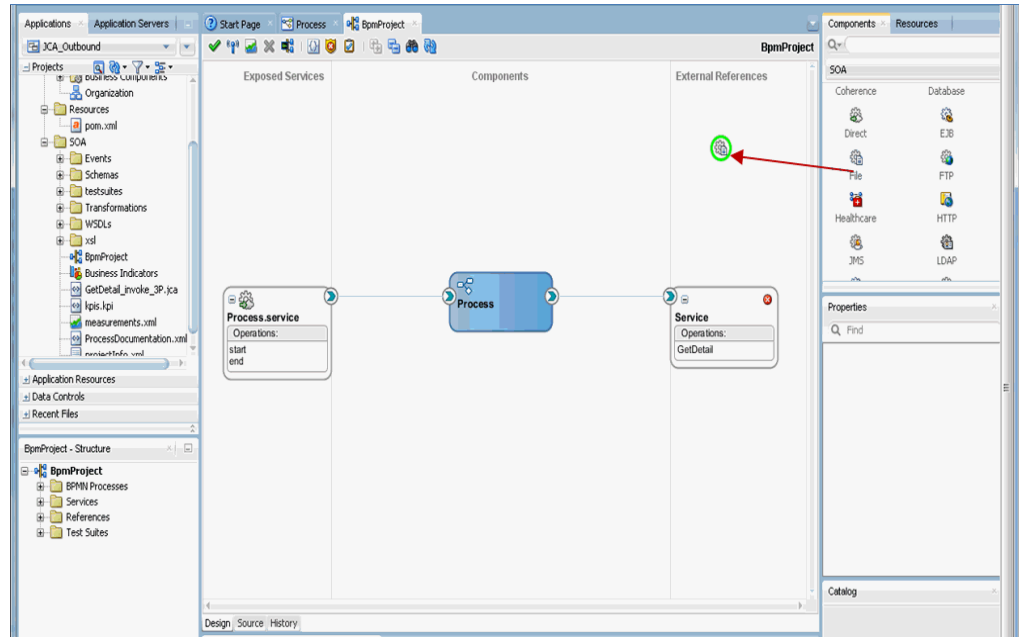
8.4.2.3 Creating a File Adapter for the Write Operation

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 8–25](#).

Figure 8–25 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.

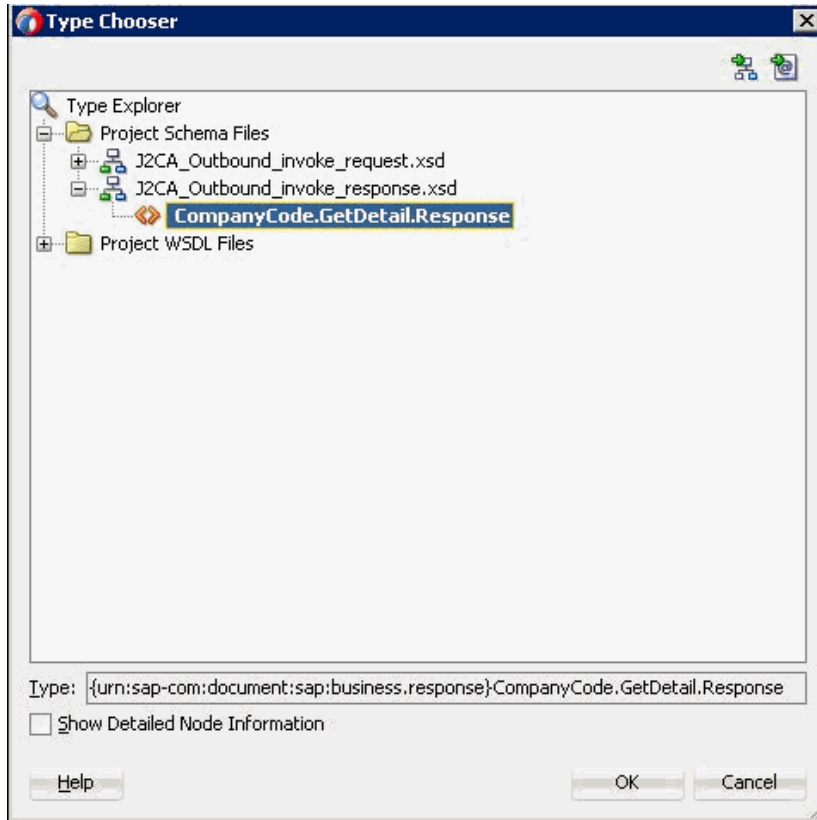
9. Specify a location on your file system where the output file is written.
10. In the File Naming Convention field, specify a name for the output file.
11. Click **Next**.

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 8–26](#).

Figure 8–26 Type Chooser Dialog



13. Expand **Project Schema Files** and **J2CA_Outbound_invoke_response.xsd**.
14. Select the available schema (for example, **CompanyCode.GetDetail.Response**).
15. Click **OK**.

You are returned to the Messages page.

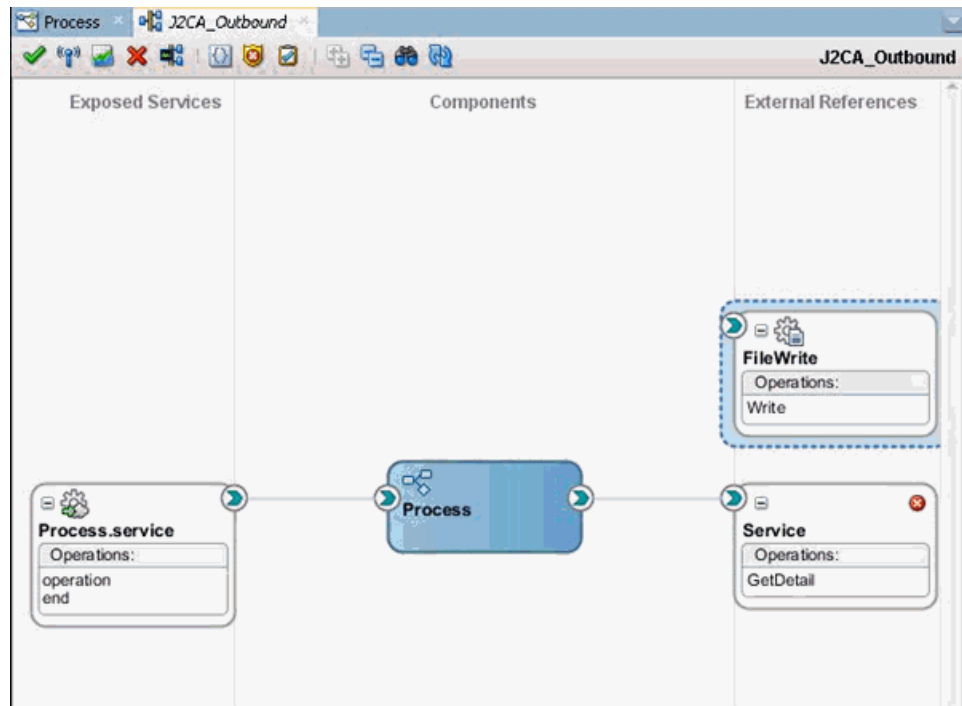
16. Click **Next**.

The Finish page is displayed.

17. Click **Finish**.

The File Adapter service is created in the External References pane, as shown in [Figure 8–27](#).

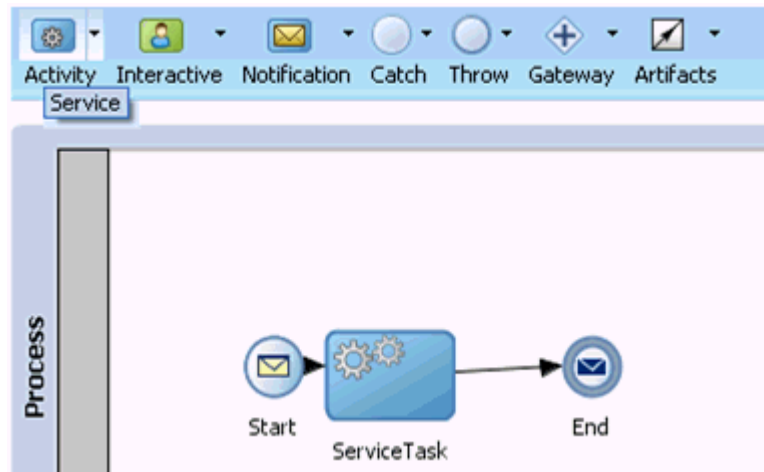
Figure 8–27 File Adapter Service



18. Double-click the BPMN Process component.

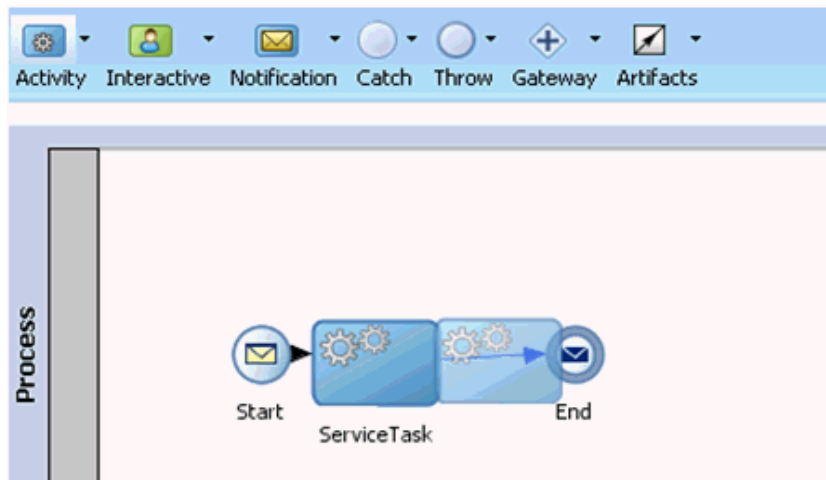
The BPMN process is displayed, as shown in [Figure 8–28](#).

Figure 8–28 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 8–29](#).

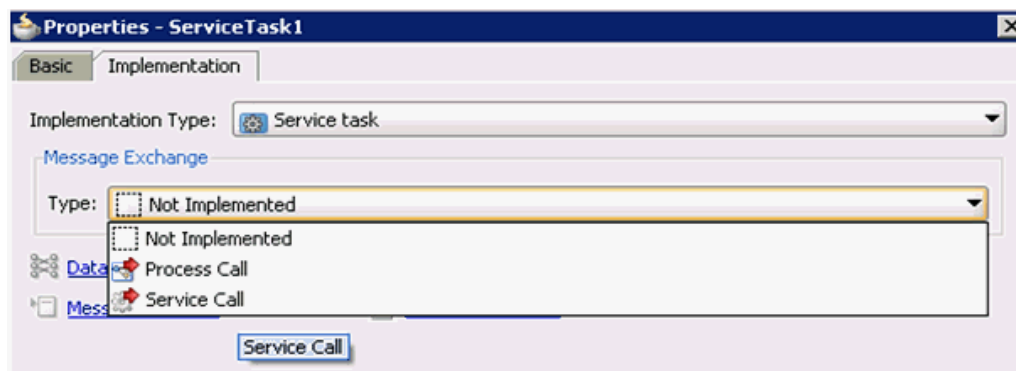
Figure 8–29 Activity Icon



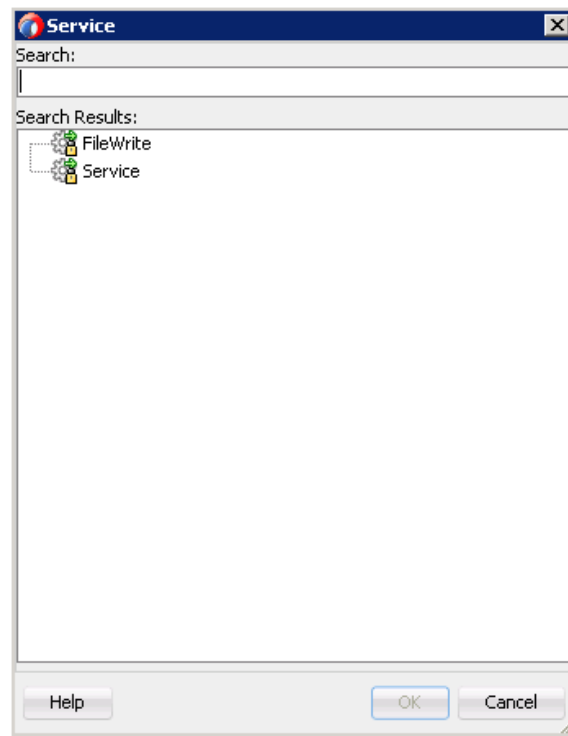
The Properties - ServiceTask1 dialog is displayed.

21. Click the **Implementation** tab.
22. Select **Service Call** from the Type drop-down list in the Message Exchange section, as shown in [Figure 8–30](#).

Figure 8–30 Service Call



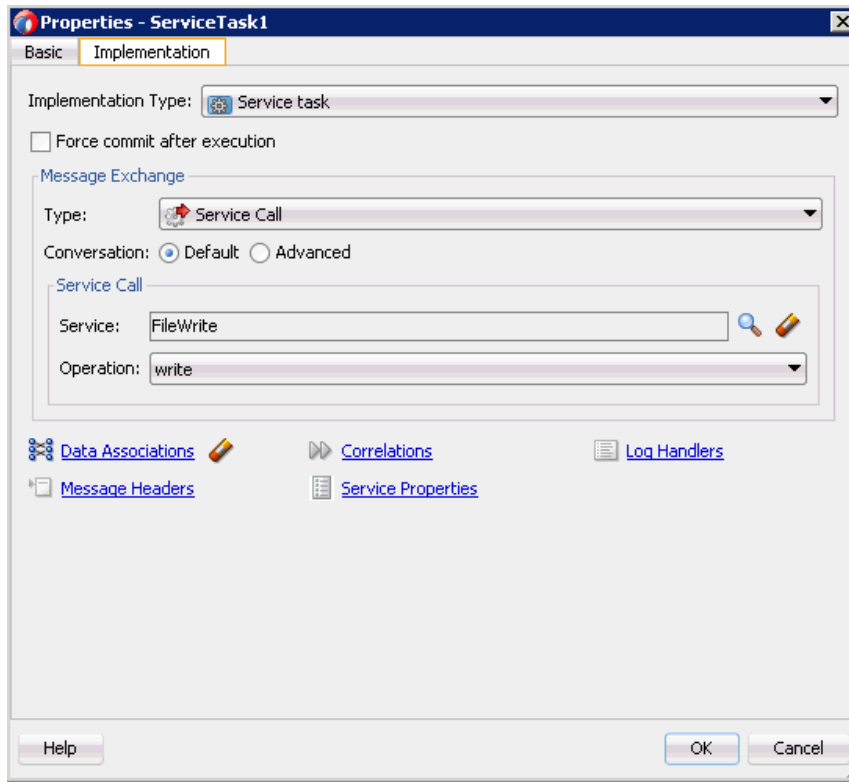
23. Click the **Browse** icon to the right of the Service field.
The Service dialog is displayed, as shown in [Figure 8–31](#).

Figure 8–31 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 8–32](#).

Figure 8–32 Properties - ServiceTask1 Dialog



25. Click the **Data Associations** hyperlink.

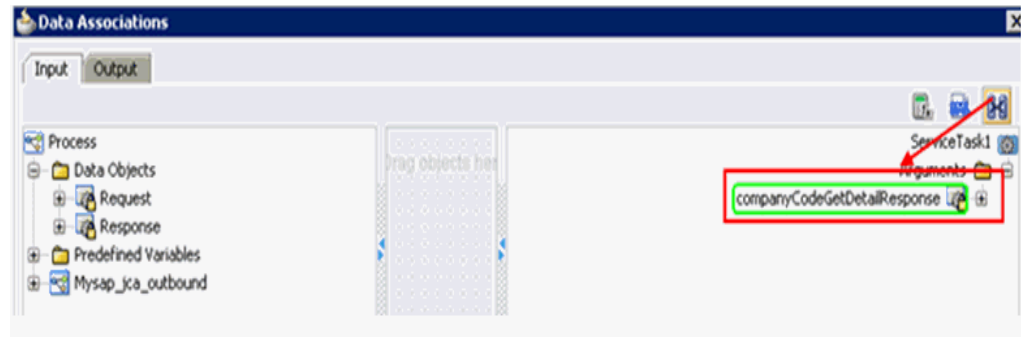
The Data Associations dialog is displayed, as shown in [Figure 8–33](#).

Figure 8–33 Data Associations Dialog



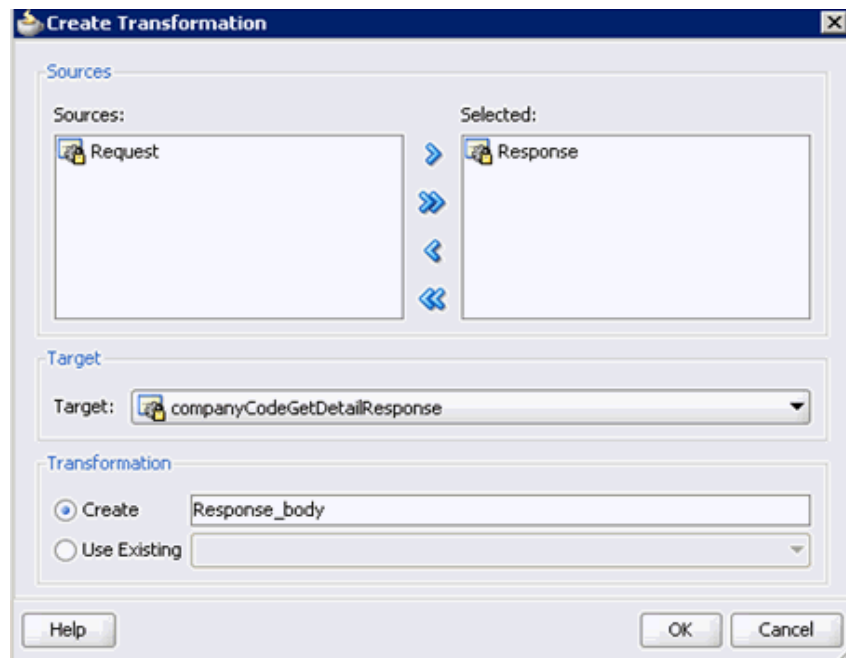
26. In the Input tab, click the XSL Transformation icon in the top right pane.

27. Drag and drop the XSL Transformation icon to the **companyCodeGetDetailResponse** node, as shown in [Figure 8–34](#).

Figure 8–34 *CompanyCodeGetDetailResponse Node*

The Create Transformation dialog is displayed.

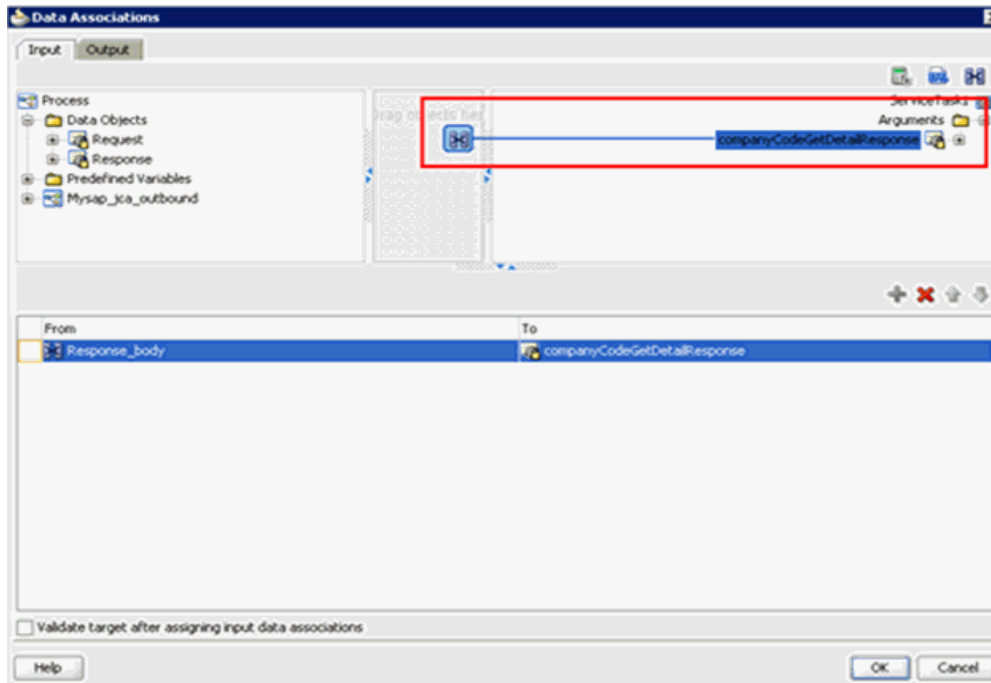
28. Select **Response** in the Sources section and click the right arrow symbol.
The Response object is added to the Selected elements area as shown in [Figure 8–35](#).

Figure 8–35 *Response Object*

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 8–36](#).

Figure 8–36 Data Associations Dialog



30. Click **OK**.

You are returned to the Properties - ServiceTask1 dialog.

31. Click **OK**.

The Response_body.xsl tab is displayed.

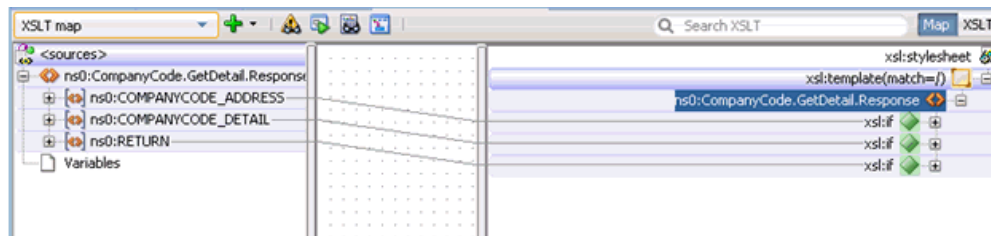
32. Automap the Source and Target elements.

The Auto Map Preferences dialog is displayed.

33. Accept the default values and click **OK**.

The transformation is completed, as shown in [Figure 8–37](#).

Figure 8–37 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.

8.4.3 Adjusting for Known Deployment Issues With 12c

For more information on how to adjust for known deployment issues with 12c, see [Section 6.4.3.3, "Adjusting for Known Deployment Issues With 12c"](#) on page 6-27.

8.4.4 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**.

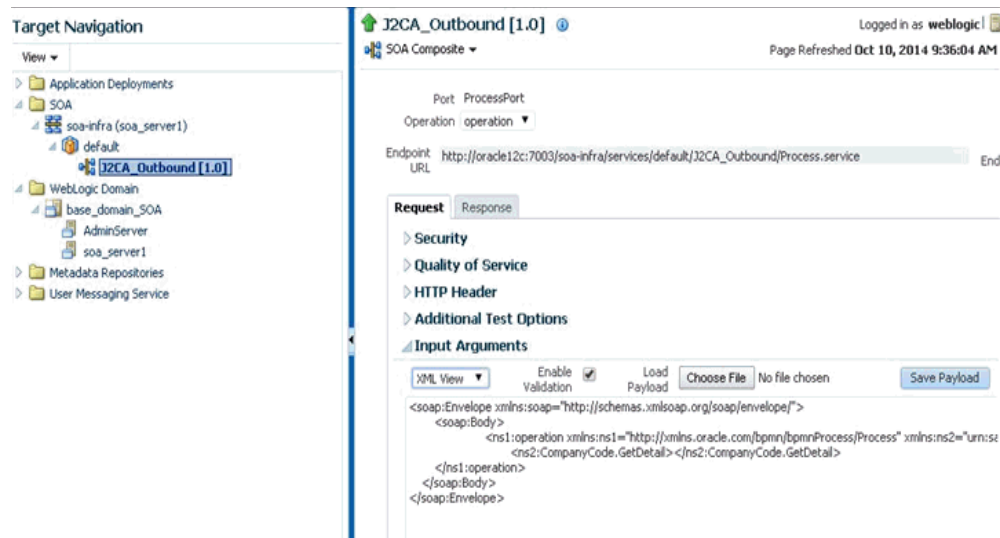
For more information, see [Section 6.4.4, "Deploying the BPEL Outbound Process"](#) on page 6-30

8.4.5 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, Mysap_jca_outbound).
4. Click **Test**.
5. Click the **Request** tab.

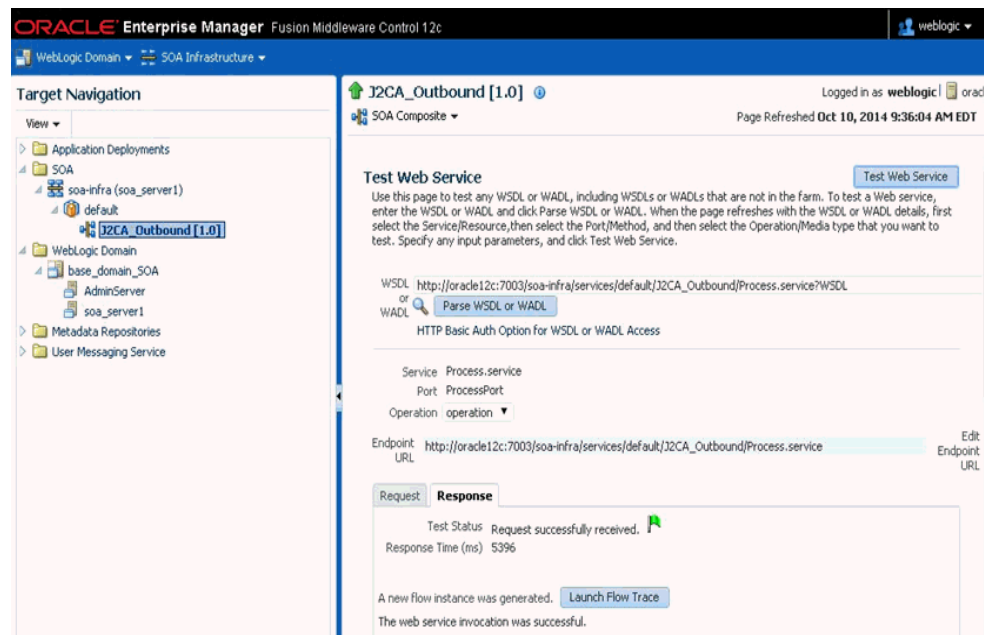
Figure 8–38 Request Tab



6. Provide an appropriate input value in the Value field and click **Test Web Service**, as shown in Figure 8–38.

A response is received in the Response tab to indicate that invocation was successful in the Oracle Enterprise Manager console, as shown in Figure 8–39.

Figure 8–39 Received Response



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. For example:

```
<?xml version="1.0" encoding="UTF-8" ?><bapi:CompanyCode.GetDetail.Response
xmlns:bapi="urn:sap-com:document:sap:business.response"
```

```

xmlns="urn:sap-com:document:sap:business.response">
  <bapi:COMPANYCODE_ADDRESS>
    <bapi:ADDR_NO>000000121</bapi:ADDR_NO>
    <bapi:FORMOFADDR>Firma</bapi:FORMOFADDR>
    <bapi:NAME>Ides AG</bapi:NAME>
    <bapi:NAME_2/>
    <bapi:NAME_3/>
    <bapi:NAME_4/>
    <bapi:C_O_NAME/>
    <bapi:CITY>Frankfurt</bapi:CITY>
    <bapi:DISTRICT/>
    <bapi:CITY_NO/>
    <bapi:POSTL_COD1>60441</bapi:POSTL_COD1>
    <bapi:POSTL_COD2>60070</bapi:POSTL_COD2>
    <bapi:POSTL_COD3/>
    <bapi:PO_BOX>160529</bapi:PO_BOX>
    <bapi:PO_BOX_CIT/>
    <bapi:DELIV_DIS/>
    <bapi:STREET>Lyoner Stern 231</bapi:STREET>
    <bapi:STREET_NO/>
    <bapi:STR_ABBR/>
    <bapi:HOUSE_NO/>
    <bapi:STR_SUPPL1/>
    <bapi:STR_SUPPL2/>
    <bapi:LOCATION/>
    <bapi:BUILDING/>
    <bapi:FLOOR/>
    <bapi:ROOM_NO/>
    <bapi:COUNTRY>DE</bapi:COUNTRY>
    <bapi:LANGU>D</bapi:LANGU>
    <bapi:REGION>06</bapi:REGION>
    <bapi:SORT1>IDES</bapi:SORT1>
    <bapi:SORT2/>
    <bapi:TIME_ZONE>CET</bapi:TIME_ZONE>
    <bapi:TAXJURCODE/>
    <bapi:ADR_NOTES/>
    <bapi:COMM_TYPE/>
    <bapi:TEL1_NUMBR/>
    <bapi:TEL1_EXT/>
    <bapi:FAX_NUMBER/>
    <bapi:FAX_EXTENS/>
  </bapi:COMPANYCODE_ADDRESS>
  <bapi:COMPANYCODE_DETAIL>
    <bapi:COMP_CODE>1000</bapi:COMP_CODE>
    <bapi:COMP_NAME>IDES AG</bapi:COMP_NAME>
    <bapi:CITY>Frankfurt</bapi:CITY>
    <bapi:COUNTRY>DE</bapi:COUNTRY>
    <bapi:CURRENCY>EUR</bapi:CURRENCY>
    <bapi:LANGU>D</bapi:LANGU>
    <bapi:CHRT_ACCTS>INT</bapi:CHRT_ACCTS>
    <bapi:FY_VARIANT>K4</bapi:FY_VARIANT>
    <bapi:VAT_REG_NO>DE123456789</bapi:VAT_REG_NO>
    <bapi:COMPANY>001000</bapi:COMPANY>
    <bapi:ADDR_NO>000000121</bapi:ADDR_NO>
    <bapi:COUNTRY_ISO>DE</bapi:COUNTRY_ISO>
    <bapi:CURRENCY_ISO>EUR</bapi:CURRENCY_ISO>
    <bapi:LANGU_ISO>DE</bapi:LANGU_ISO>
  </bapi:COMPANYCODE_DETAIL>
  <bapi:RETURN>
    <bapi:TYPE/>

```

```
<bapi:CODE/>
<bapi:MESSAGE/>
<bapi:LOG_NO/>
<bapi:LOG_MSG_NO>000000</bapi:LOG_MSG_NO>
<bapi:MESSAGE_V1/>
<bapi:MESSAGE_V2/>
<bapi:MESSAGE_V3/>
<bapi:MESSAGE_V4/>
</bapi:RETURN>
</bapi:CompanyCode.GetDetail.Response>
```

8.5 Designing an Inbound BPM Process Using Transformations for Event Integration (J2CA Configuration)

This section demonstrates how Oracle Application Adapter for SAP R/3 integrates with SAP R/3 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\SAP_Samples.zip\SAP_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 8.5.1, "Creating an Empty Composite for BPM"](#)
- [Section 8.5.2, "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 ["Generating WSDL for Event Integration"](#) on page 6-37.

8.5.1 Creating an Empty Composite for BPM

For more information on how to configure a new Application Server connection in Oracle JDeveloper, see [Section 6.3, "Configuring a New Application Server Connection"](#) on page 6-2.

8.5.2 Defining a BPM Inbound Process

This section describes how to define a BPM inbound process, which contains the following topics:

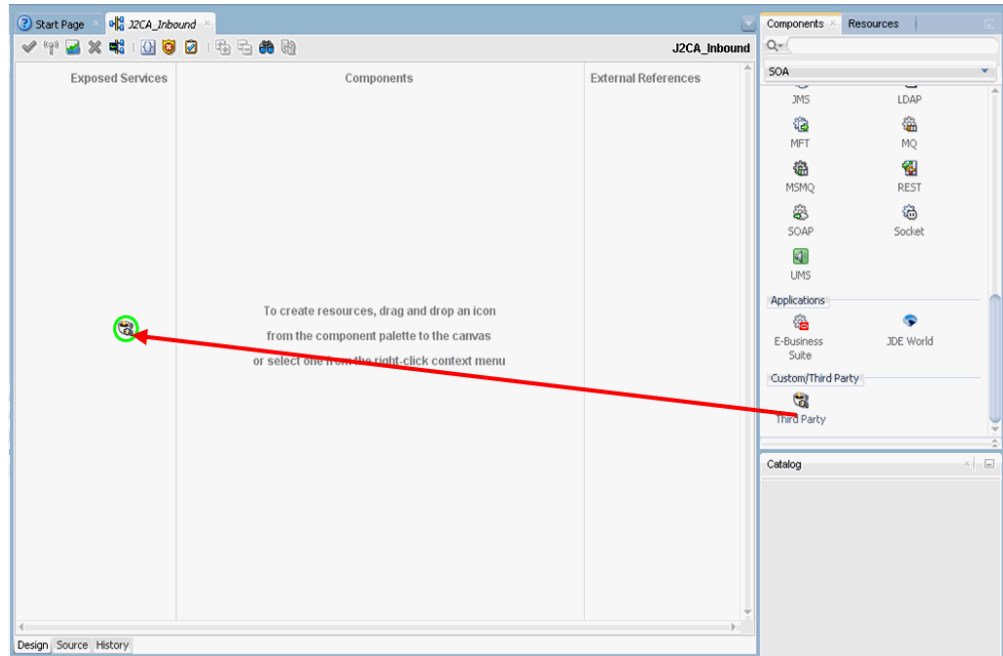
- [Section 8.5.2.1, "Configuring a Third Party Adapter Service Component"](#)
- [Section 8.5.2.2, "Configuring an Inbound BPM Process Component"](#)
- [Section 8.5.2.3, "Creating a File Adapter for the Write Operation"](#)
- [Section 8.5.2.4, "Adjusting for Known Deployment Issues With 12c"](#)

8.5.2.1 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 8–40](#).

Figure 8–40 Third Party Adapter Component



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>\wds1s
```

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>\wsdl.s

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 6.5.3.1, "Creating a Third Party Adapter Service Component"](#) on page 6-45

8.5.2.2 Configuring an Inbound BPM Process Component

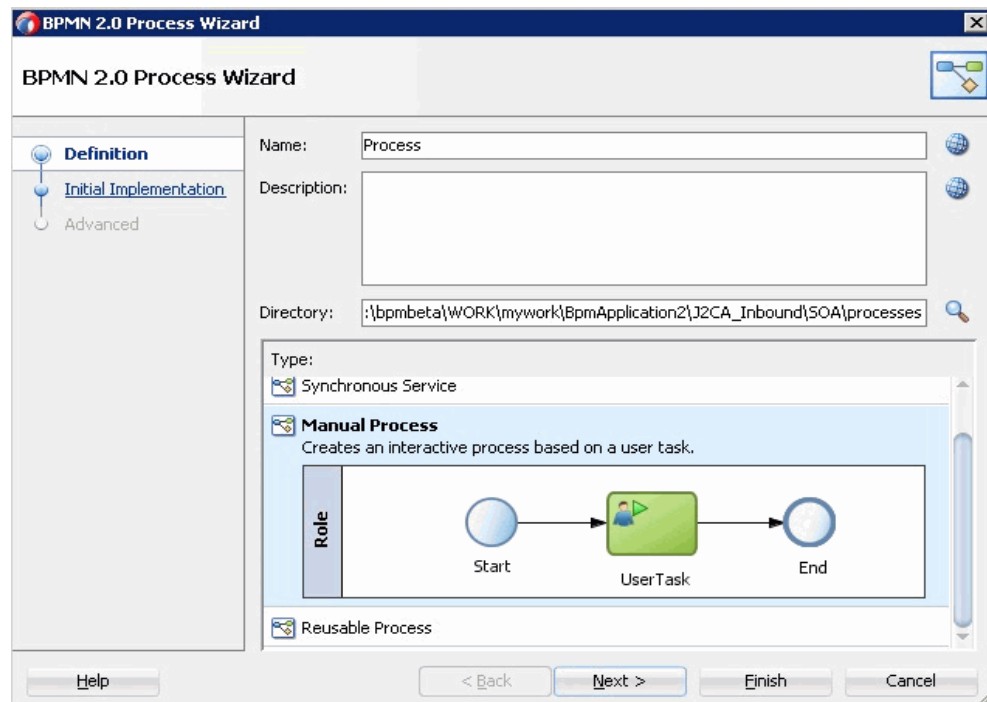
This section describes how to configure an inbound BPM process component.

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 8-41](#).

Figure 8-41 Create BPMN Process Dialog

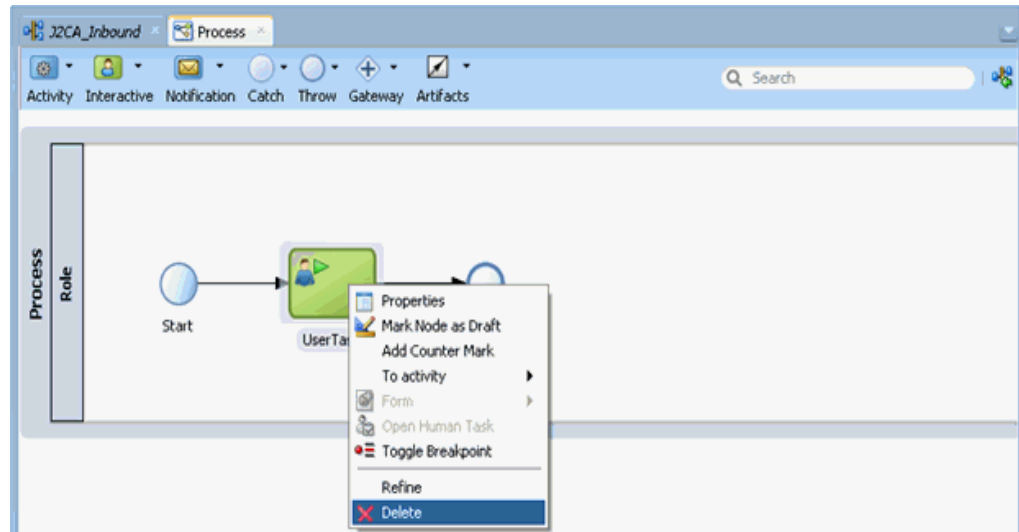


2. Select **Manual Process** in the Type section.

3. Click **Finish**.

The BPMN process is displayed, as shown in [Figure 8–42](#).

Figure 8–42 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.

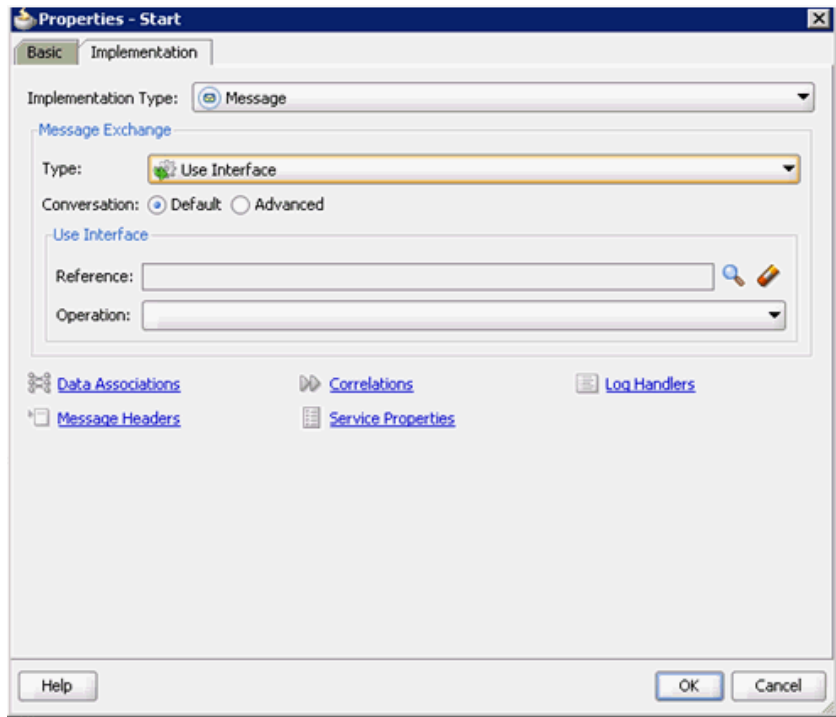
6. Click the **Implementation** tab.

7. Select **Message** from the Implementation Type list.

8. Select **Use Interface** from the Message Exchange Type drop-down list.

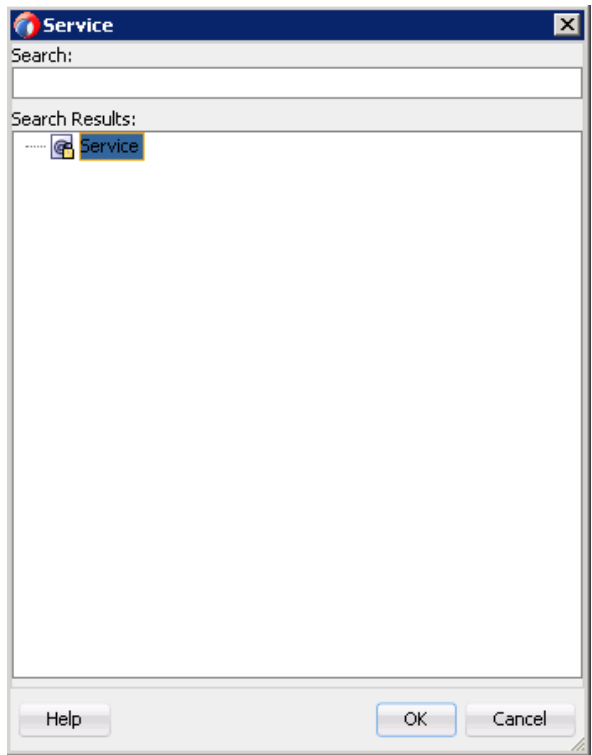
9. Click the **Browse** icon to the right of the Reference field, as shown in [Figure 8–43](#).

Figure 8–43 Browse Icon



The Service dialog is displayed, as shown in [Figure 8–44](#).

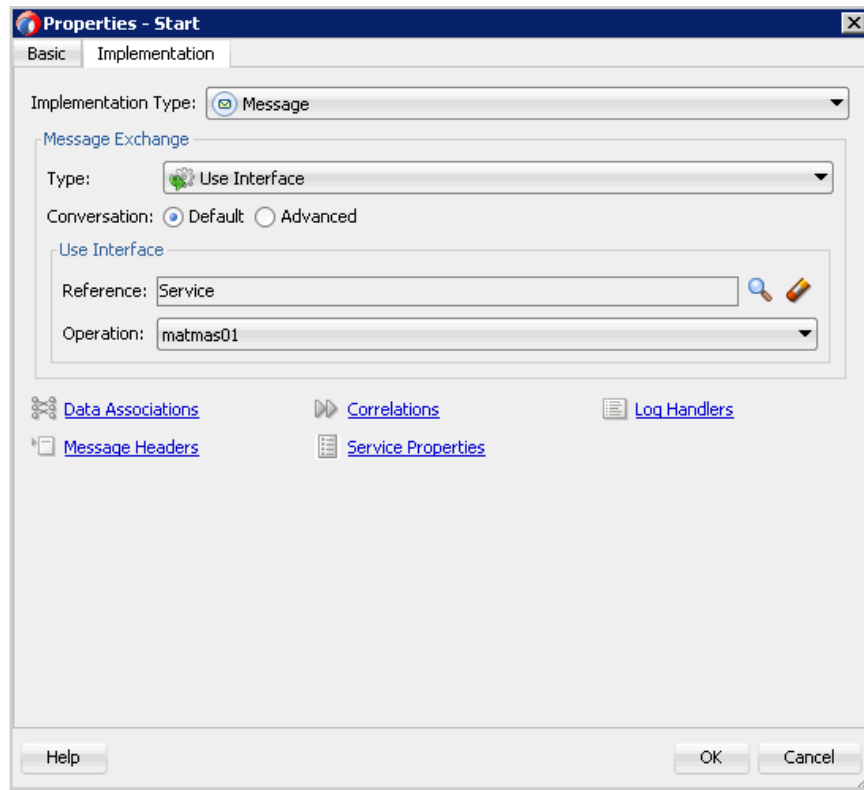
Figure 8–44 Service Dialog



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 8-45](#).

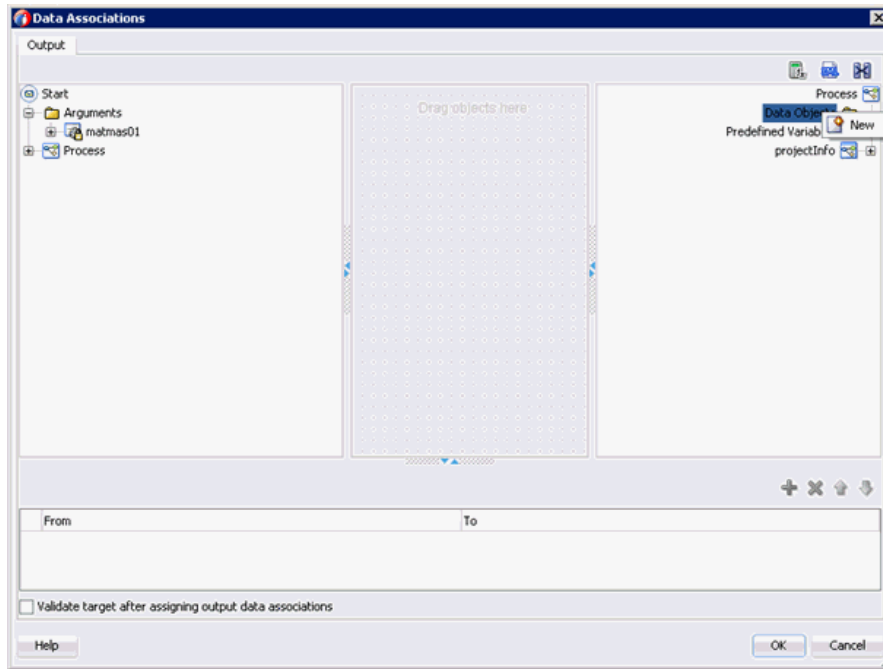
Figure 8-45 *Properties - Start Dialog*



11. Click the **Data Associations** icon.

The Data Associations dialog is displayed, as shown in [Figure 8-46](#).

Figure 8–46 Data Associations Dialog

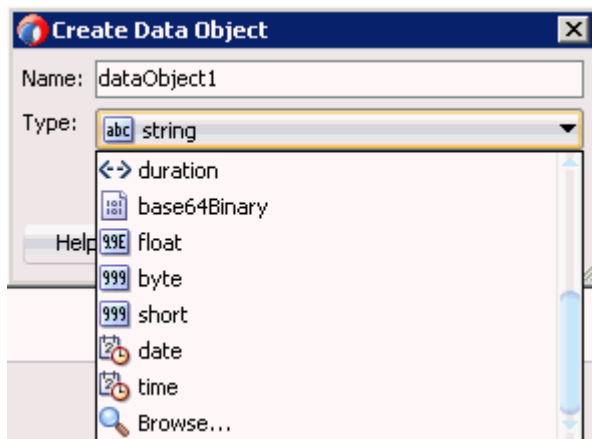


12. Right-click the **Data Object** node in the right pane and select **New**.

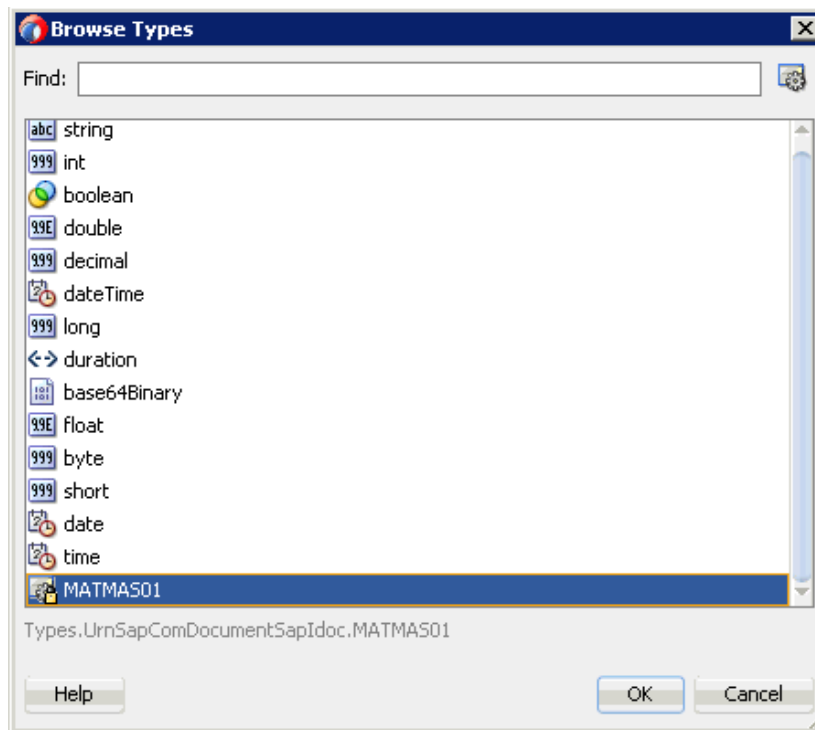
The Create Data Object dialog is displayed.

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 8–47](#).

Figure 8–47 Create Data Object Dialog



The Browse Types dialog is displayed, as shown in [Figure 8–48](#).

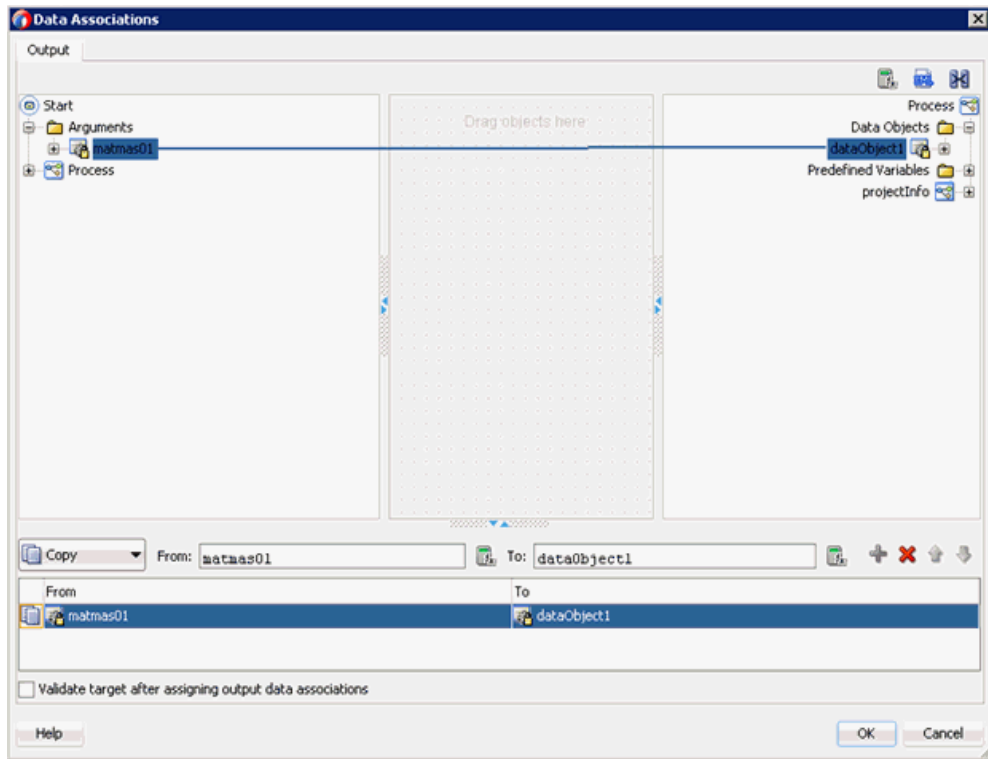
Figure 8–48 Browse Types Dialog

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 8–49](#).

Figure 8–49 Data Associations Dialog

16. Select and drag the **matmas01** 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. 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.

8.5.2.3 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:

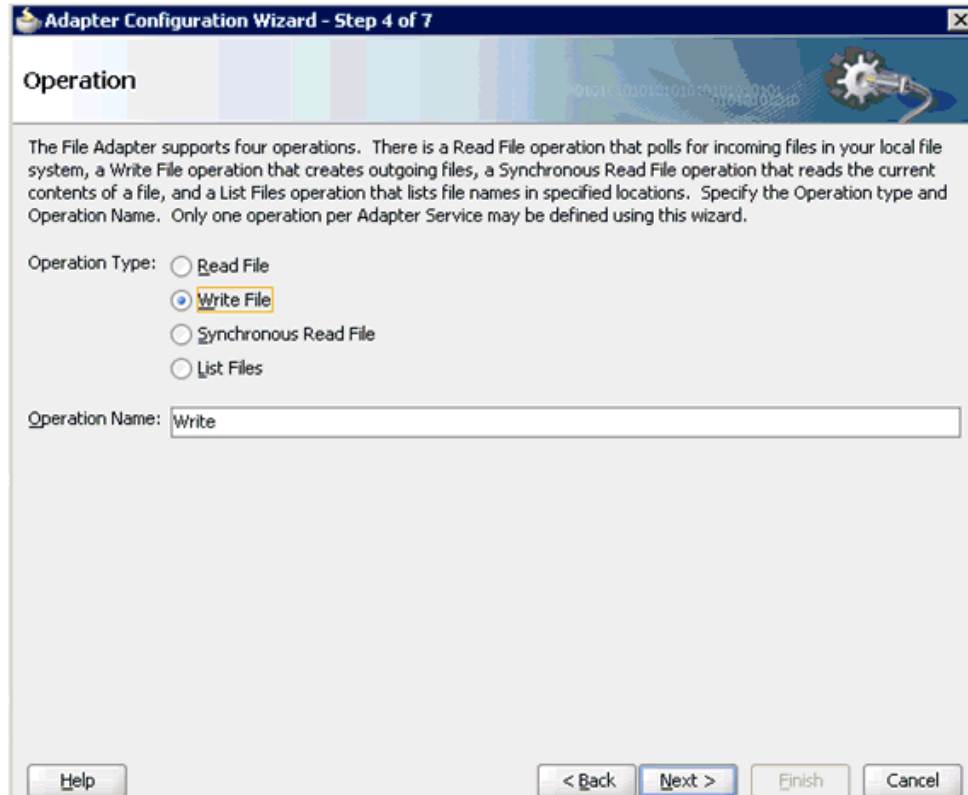
1. Drag and drop the **File Adapter** component from the Technology Adapters pane to the External References pane.
The Adapter Configuration Wizard is displayed.
2. Type a name for the new File adapter in the Name field and click **Next**.
The Adapter Interface page is displayed.
3. Ensure that the **Define from operation and schema (specified later)** option is selected.
4. Click **Next**.

The File Server Connection page is displayed.

5. Click **Next**.

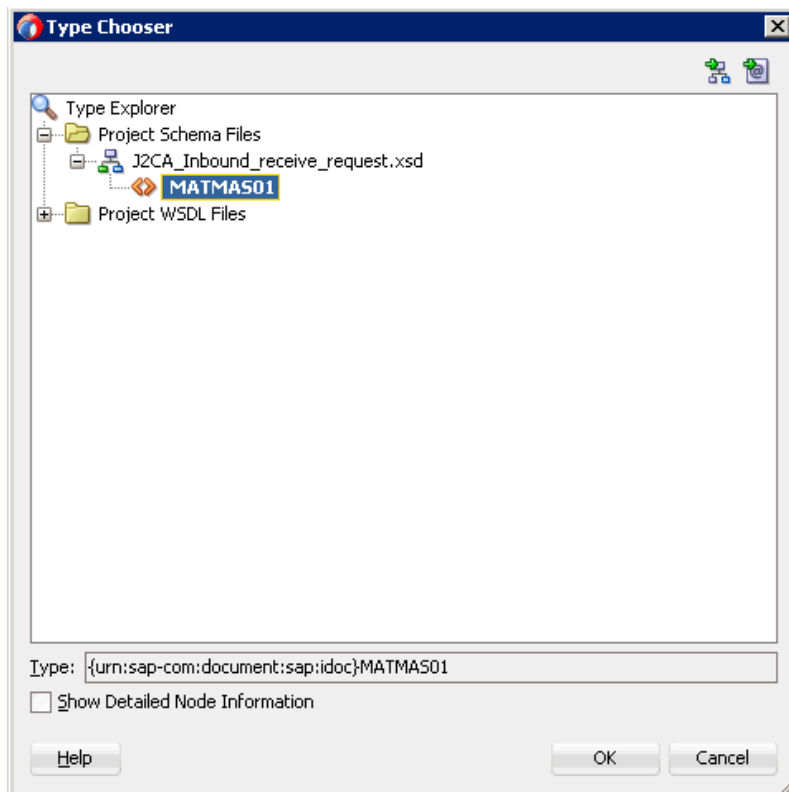
The Operation page is displayed, as shown in [Figure 8–50](#).

Figure 8–50 Operation Page



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 8–51](#).

Figure 8-51 Type Chooser Dialog



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.

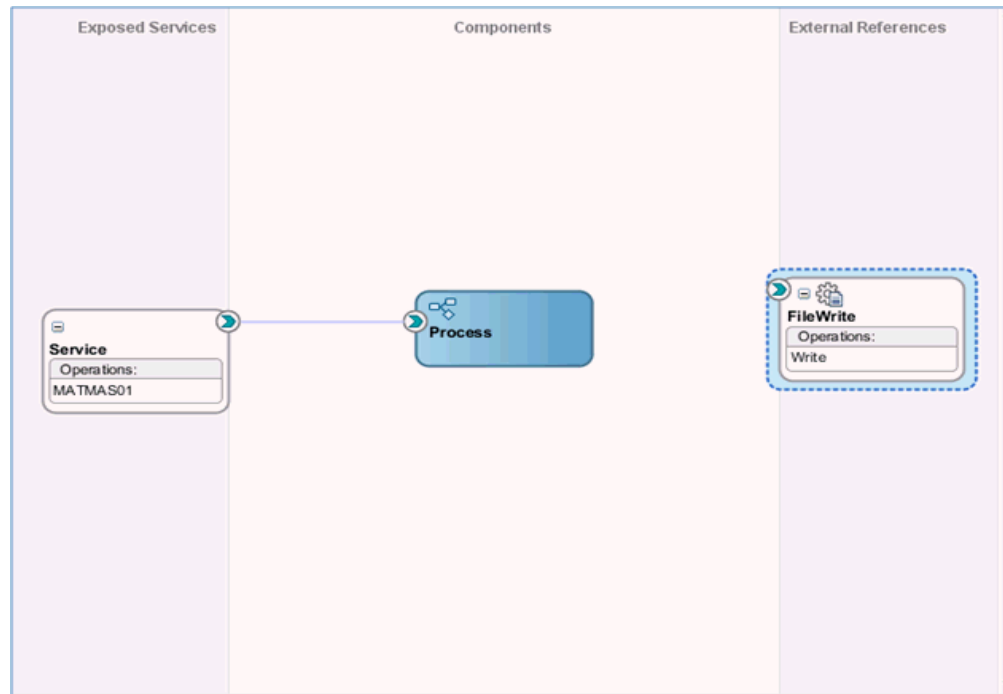
15. Click **Next**.

The Finish page is displayed.

16. Click **Finish**.

The File Adapter service is created in the External References pane, as shown in [Figure 8-52](#).

Figure 8–52 File Adapter Service



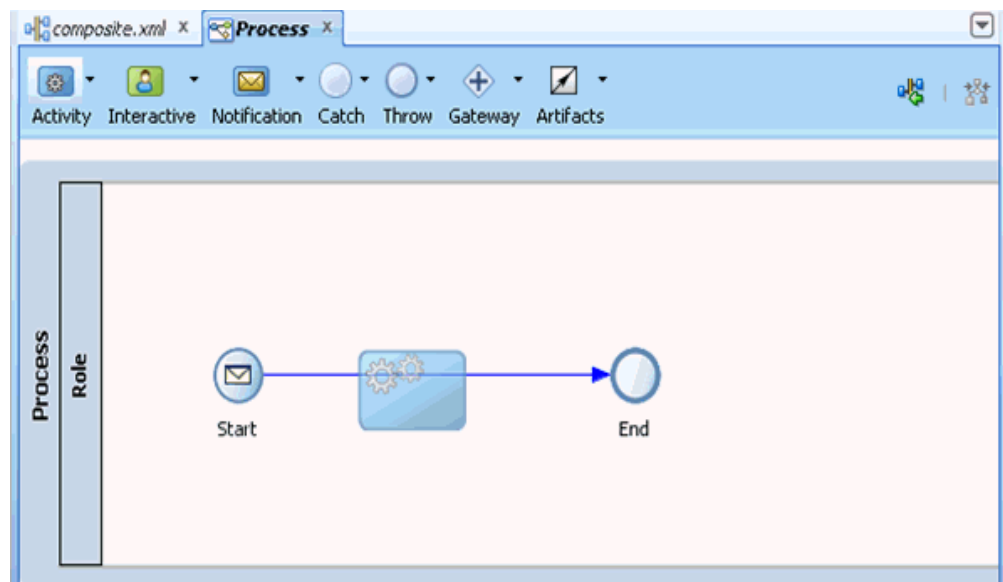
17. Double-click the BPMN Process component.

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 8–53](#).

Figure 8–53 Activity Icon



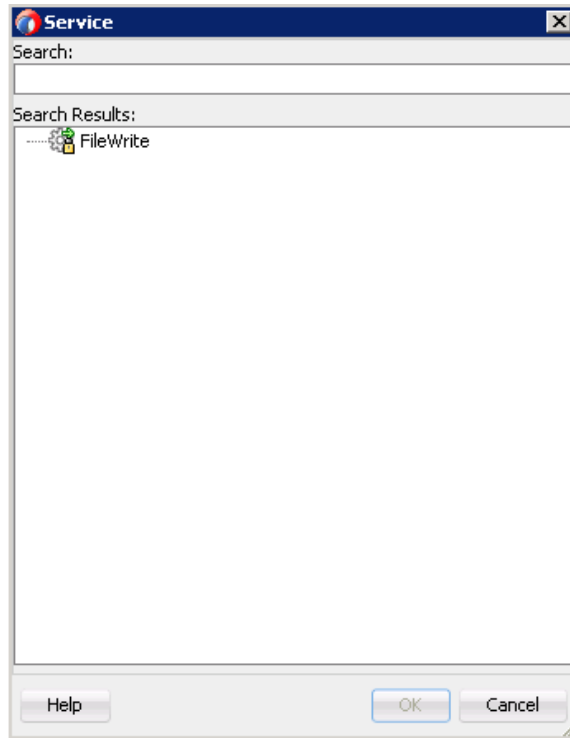
The Properties - ServiceTask dialog is displayed.

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.

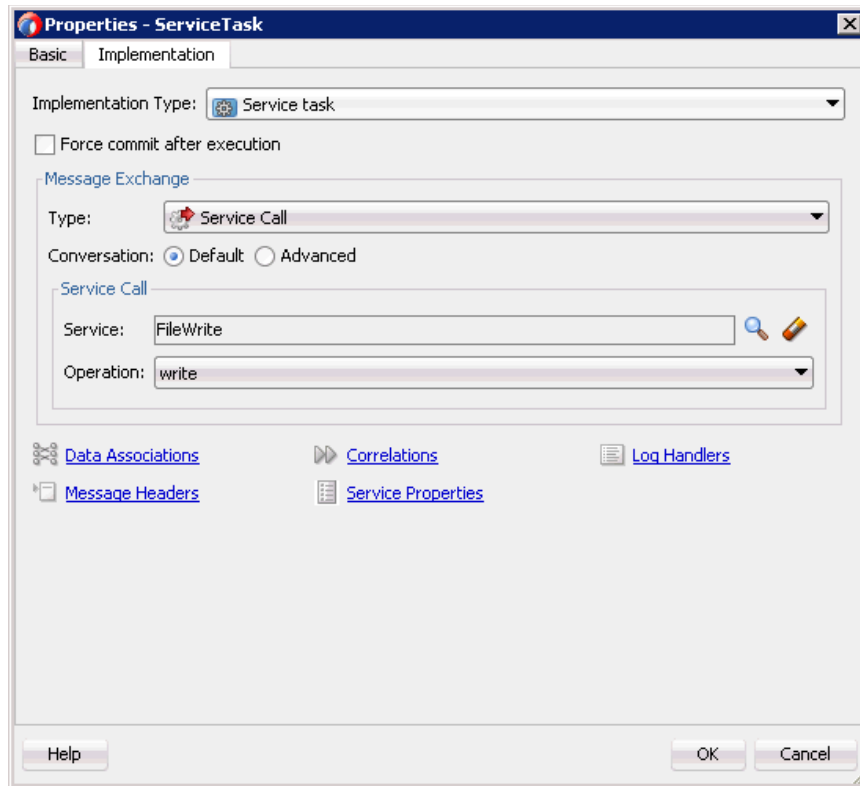
The Type dialog is displayed, as shown in [Figure 8–54](#).

Figure 8–54 *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 8–55](#).

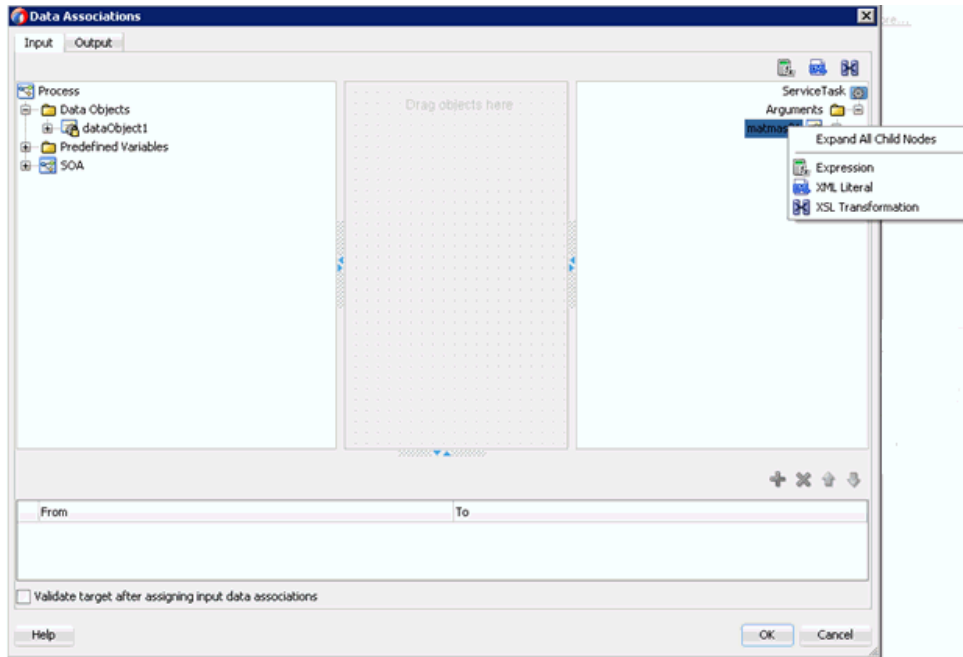
Figure 8–55 Properties - ServiceTask Dialog

25. Click the **Data Associations** hyperlink.

The Data Associations dialog is displayed.

26. Right-click the **matmas01** argument on the right pane and select **XSL Transformation**, as shown in [Figure 8–56](#).

Figure 8–56 XSL Transformation

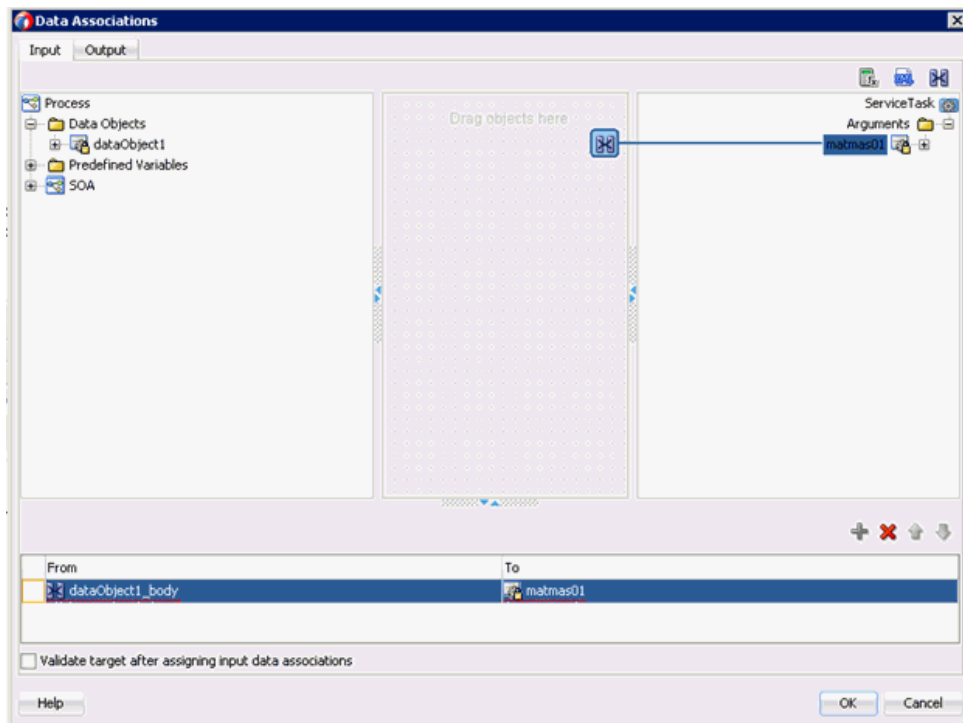


The Create Transformation dialog is displayed.

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 8–57](#).

Figure 8–57 Data Associations Dialog



29. Click **OK**.

You are returned to the Properties - ServiceTask dialog.

30. Click **OK**.

The dataobject1_body.xml tab is displayed.

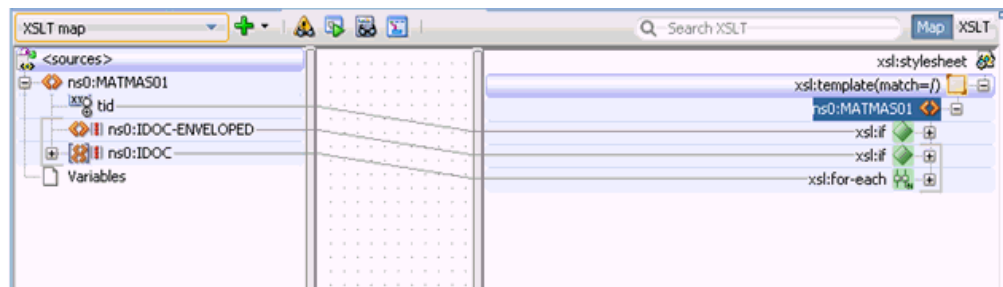
31. Automap the Source and Target elements.

The Auto Map Preferences dialog is displayed.

32. Accept the default values and click **OK**.

The transformation is completed, as shown in [Figure 8–58](#).

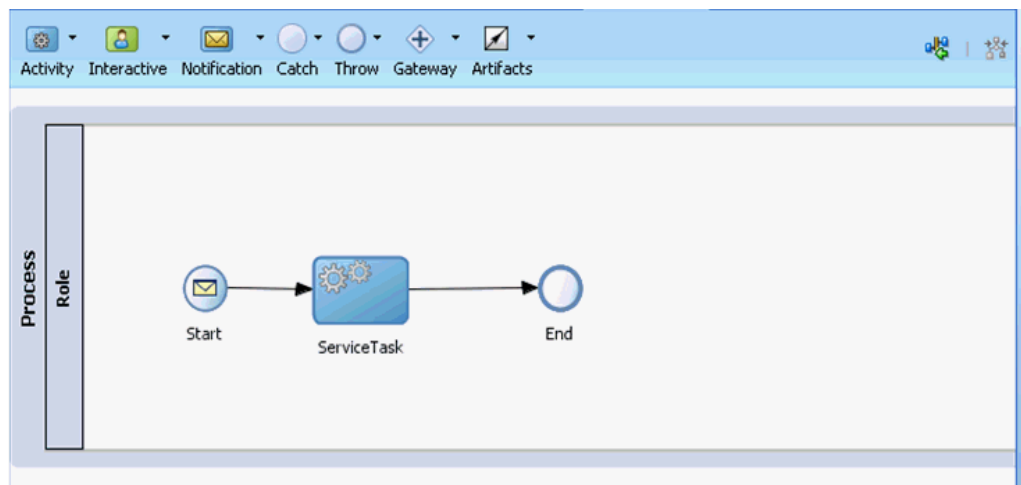
Figure 8–58 Completed Transformation



33. Save the transformation.

34. Return to the Process workspace area, as shown in [Figure 8–59](#).

Figure 8–59 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.

8.5.2.4 Adjusting for Known Deployment Issues With 12c

For more information on how to adjust for known deployment issues with 12c, see [Section 6.4.3.3, "Adjusting for Known Deployment Issues With 12c"](#) on page 6-27.

You are now ready to deploy the inbound BPM process. You can follow the same procedure that is described in ["Deploying the BPEL Inbound Process"](#) on page 6-53. For more information on how to trigger events in SAP R/3, see ["Triggering an Event in SAP R/3"](#) on page 6-54.

8.6 Working With Synchronous SAP R/3 Events Using BPM (J2CA Configuration)

This section describes how to work with synchronous SAP R/3 events using BPM. For more information on synchronous SAP R/3 events, see [Section 6.6.1, "Overview"](#).

Note: Synchronous events are supported only for objects that have reply schema and are not supported for IDocs.

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

```
<ADAPTER_HOME>\etc\sample\SAP_Samples.zip\SAP_Samples\BPM\J2CA\Synchronous_Events
```

This section includes the following topics:

- [Section 8.6.1, "Configuring the Adapter Target and Channels"](#)
- [Section 8.6.2, "Generating a WSDL File for a Message"](#)
- [Section 8.6.3, "Creating a BPM Process in JDeveloper"](#)

8.6.1 Configuring the Adapter Target and Channels

Open Application Explorer and create two targets and one channel for SAP using a J2CA configuration.

For more information, see [Chapter 4, "Configuring Oracle Application Adapter for SAP R/3"](#).

8.6.2 Generating a WSDL File for a Message

Perform the following steps to generate a WSDL file for a message:

1. Ensure that the channel is not started before generating the WSDL file.
2. Connect to the first SAP adapter target (target1).
3. Generate an inbound WSDL file for BAPI_COMPANYCODE_GETLIST (navigate using Remote Function Modules, Financial Accounting, 0002 - Company Code Business Object).
4. Connect to the second SAP adapter target (target2).
5. Generate an outbound WSDL file for BAPI_COMPANYCODE_GETLIST (navigate using Remote Function Modules, Financial Accounting, 0002 - Company Code Business Object).

For more information, see [Section 6.4.1, "Generating WSDL for Request/Response Service"](#) and [Section 6.5.1, "Generating WSDL for Event Integration"](#).

8.6.3 Creating a BPM Process in JDeveloper

This section describes how to create a BPM process in JDeveloper.

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

For more information, see [Section 6.4.2, "Creating an Empty Composite for SOA"](#) on page 6-9.

8.6.3.2 Creating a Third Party Adapter Service for an Inbound WSDL File

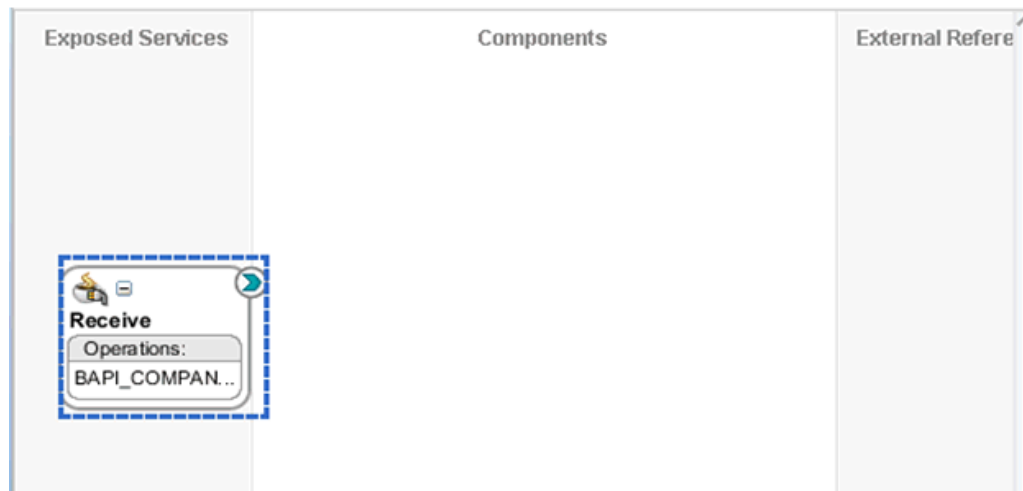
To create a third party adapter service for an inbound WSDL file:

1. Drag and drop the **Third Party Adapter** component from the Custom/Third Party area to the Exposed Services pane.
The Create Third Party Adapter Service dialog is displayed.
2. Enter a name in the Name field and click on the icon next to the WSDL URL field.
The WSDL Chooser dialog is displayed.
3. Select the **File System** tab and then browse and select the inbound WSDL file from the WSDL location, and click **OK**.
4. In the Localize Files dialog, click **OK**.

The inbound WSDL, request, and response XML schema files (.xsd), are imported to the created project folder.

5. Click the **Find jca file** icon to select the inbound JCA properties file.
The Transformation Chooser dialog is displayed.
6. Select the JCA properties file from the WSDL location and click **OK**.
The Copy File message window is displayed.
7. Click **Yes**.
A copy of the JCA properties file is made in the project folder.
8. Click **OK**.

The third party adapter service for the inbound WSDL file is created in the Exposed Services pane, as shown in [Figure 8-60](#).

Figure 8–60 Exposed Services Pane

8.6.3.3 Creating a Third Party Adapter Service for an Outbound WSDL File

To create a third party adapter service for an outbound WSDL file:

1. Drag and drop the **Third Party Adapter** component from the Custom/Third Party area to the External References pane.

The Create Third Party Adapter Service dialog is displayed.

2. Enter a name in the Name field and click on the icon next to the WSDL URL field.

The WSDL Chooser dialog is displayed.

3. Browse and select the outbound WSDL created from the WSDL location and click **OK**.

The Localize Files dialog is displayed.

4. Click **OK**.

The outbound WSDL, request, and response XML schema files (.xsd) are imported to the created project folder.

5. Click the **Find jca file** icon to select the outbound JCA properties file.

6. Select the JCA properties file from the WSDL location and click **OK**.

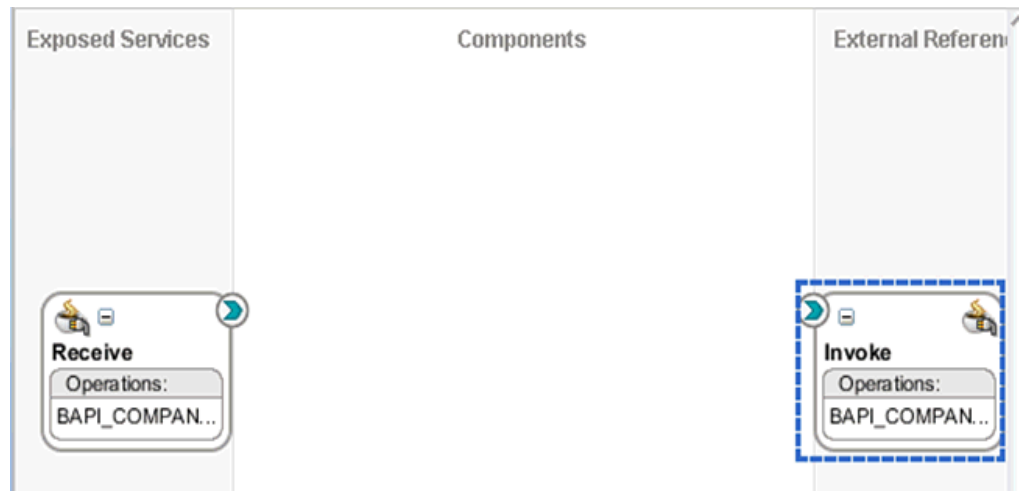
7. The Copy File message window is displayed.

8. Click **Yes**.

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

9. Click **OK**.

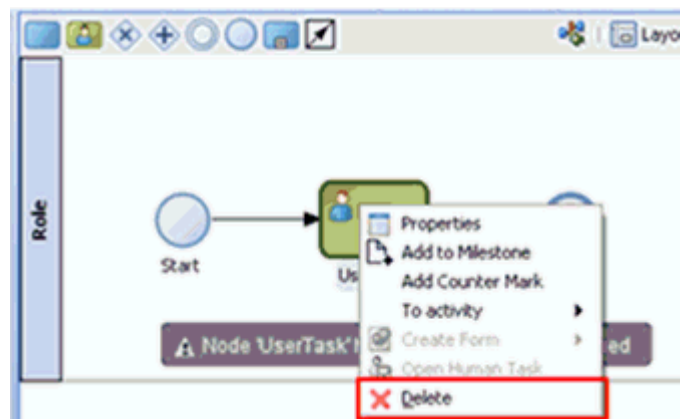
The third party adapter service for the outbound WSDL is created in the External References pane, as shown in [Figure 8–61](#).

Figure 8–61 External References Pane

8.6.3.4 Creating a Synchronous BPM Process

To create a synchronous BPMN process:

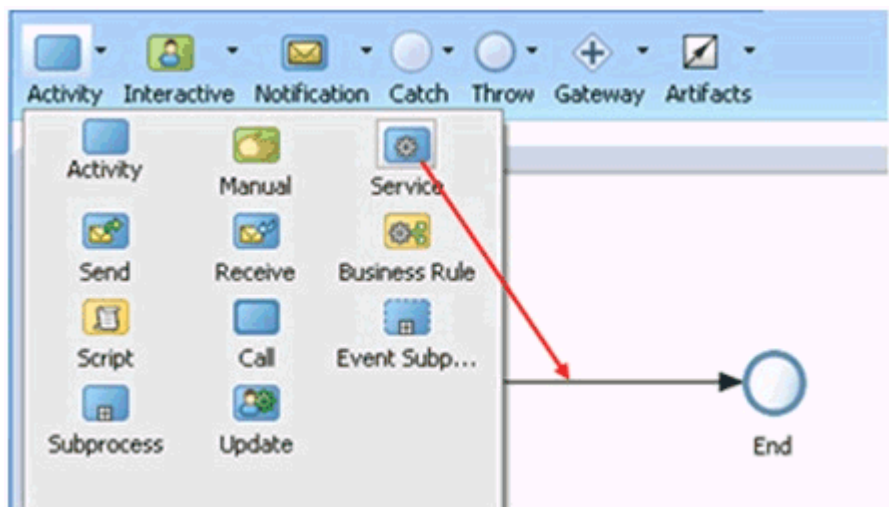
1. Drag and drop the **BPMN Process** component from the Components area to the Components pane.
The BPMN2.0 Process Wizard window is displayed.
2. Select **Manual Process** in the Type section and then click **Finish**.
The BPM process workspace area is displayed.
3. Right-click the **UserTask** icon and select **Delete** from the menu, as shown in [Figure 8–62](#).

Figure 8–62 Delete Option

4. Double-click the **Start** event component.
The Properties - Start dialog is displayed.
5. Click the **Implementation** tab.
6. Click **Message** from the Implementation Type list and then select **Use Interface** from the Type list under the Message Exchange tab.

7. Click the **Browse** icon to the right of the Reference field under the Use Interface tab.
The Service dialog is displayed.
8. Select the Third Party Service that has been created for the inbound WSDL and click **OK**.
You are returned to the Properties - Start dialog.
9. Click **Data Associations**.
The Data Associations dialog is displayed.
10. Right-click **Data Objects** under the Process node in the right pane and select **New**.
The Create Data Object Dialog is displayed.
11. Enter a name (for example, receive_in) in the Name field.
12. Click the drop-down button in the Type field and select **Browse** from the list.
The Browse Types dialog is displayed.
13. Select the corresponding Request component and click **OK**.
You are returned to the Create Data Object dialog.
14. Click **OK**.
The Data Object (for example, receive_in) that has been created is displayed under the Data Objects in the Data Associations dialog.
15. Select the component listed under the Arguments node in the left pane and drag and connect it to the created data object under the Data Objects in the right pane.
16. Click **OK**.
You are returned to the Properties - Start dialog.
17. Click **OK**.
You are returned to the Process workspace area.
18. Select and drop the **Service Activity** icon on the wire between the Start and End event components, as shown in [Figure 8-63](#).

Figure 8-63 Service Activity Icon



The Properties - ServiceTask dialog is displayed.

19. Click the **Implementation** tab, and select **Service Call** from the Type list under the Message Exchange tab.

20. Click the **Browse** icon to the right of the Service field.

The Service dialog is displayed.

21. Select the Third Party Service that has been created for the outbound WSDL and click **OK**.

You are returned to the Properties - Start dialog window.

22. Click **Data Associations**.

The Data Associations dialog is displayed.

23. Below the Input Area, right-click **Data Objects** under the Process node in the left pane and select **New**.

The Create Data Object dialog is displayed.

24. Enter a name (for example, invoke_in) in the Name field, click the drop-down button in the Type field and select **Browse...** from the list.

The Browse Types dialog is displayed.

25. Select the corresponding Request component and click **OK**.

You are returned to the Create Data Object dialog.

26. Click **OK**.

The Data Object (for example, invoke_in) that has been created is displayed under the Data Objects in the Data Associations dialog.

27. Select the created data object under Data Objects in the left pane, and drag and connect it to the component listed under the Arguments node in the right pane.

28. Select the Output Area, right-click **Data Objects** under the Process node in the right pane and select **New**.

The Create Data Object dialog is displayed.

29. Enter a name (for example, invoke_out) in the Name field, click the drop-down button in the Type field and select **Browse...** from the list.

The Browse Types dialog is displayed.

30. Select the corresponding Response component and click **OK**.

You are returned to the Create Data Object dialog.

31. Click **OK**.

The Data Object (for example, invoke_out) that has been created is displayed under Data Objects in the Data Associations dialog.

32. Select the component listed under the Arguments node in the left pane, and drag and connect it to the created data object (Response) under Data Objects in the right pane.

33. Click **OK**.

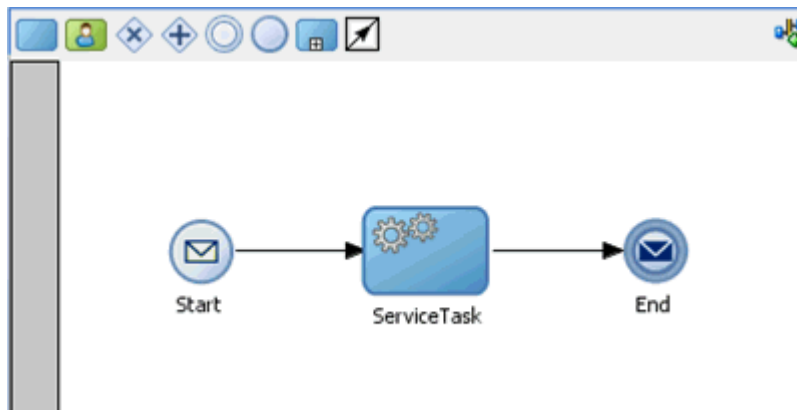
You are returned to the Properties - Start dialog.

34. Click **OK**.

You are returned to the Process workspace area.

35. Double-click the **End** event component.
The Properties - End dialog is displayed.
36. Click the **Implementation** tab.
37. Select **Message** from the Implementation Type list and select **Use Interface** from the Type list under the Message Exchange tab.
38. Select the **Synchronous** button in the Type field below User Interface.
39. Click the **Browse** icon to the right of the Reference field under the Use Interface tab.
The Service dialog is displayed.
40. Select the **Third Party Service** that has been created for the inbound WSDL and click **OK**.
41. Click **Data Associations**.
The Data Associations dialog is displayed.
42. Select the created data object for response (for example, invoke_out) displayed under Data Objects in the left pane, and drag and connect it to the component listed under Arguments in the right pane.
43. Click **OK**.
You are returned to the Properties - End dialog.
44. Click **OK** and then **Save All**.
You are returned to the Process workspace area, as shown in [Figure 8-64](#).

Figure 8-64 Process Workspace Area



The BPM process creation for synchronous events is completed.

8.6.3.5 Adjusting for Known Deployment Issues With 12c

For more information on how to adjust for known deployment issues with 12c, see [Section 6.4.3.3, "Adjusting for Known Deployment Issues With 12c"](#) on page 6-27.

8.6.3.6 Deploying the BPM Process

For more information on how to deploy the BPM process, see [Section 8.4.4, "Deploying the BPM Outbound Process"](#) on page 8-25.

8.7 Designing an Outbound BPM Process Using Transformations for Service Integration (BSE Configuration)

This section describes how to configure a BPM outbound process to your SAP R/3 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\SAP_Samples.zip\SAP_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 8.7.1, "Creating an Empty Composite for BPM"](#)
- [Section 8.7.2, "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 6.7.1, "Generating a WSDL File for Request and Response Services Using a Web Service"](#).

8.7.1 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 6.4.2, "Creating an Empty Composite for SOA"](#) on page 6-9.

8.7.2 Defining a BPM Outbound Process

This section describes how to define a BPM outbound process. It contains the following topics:

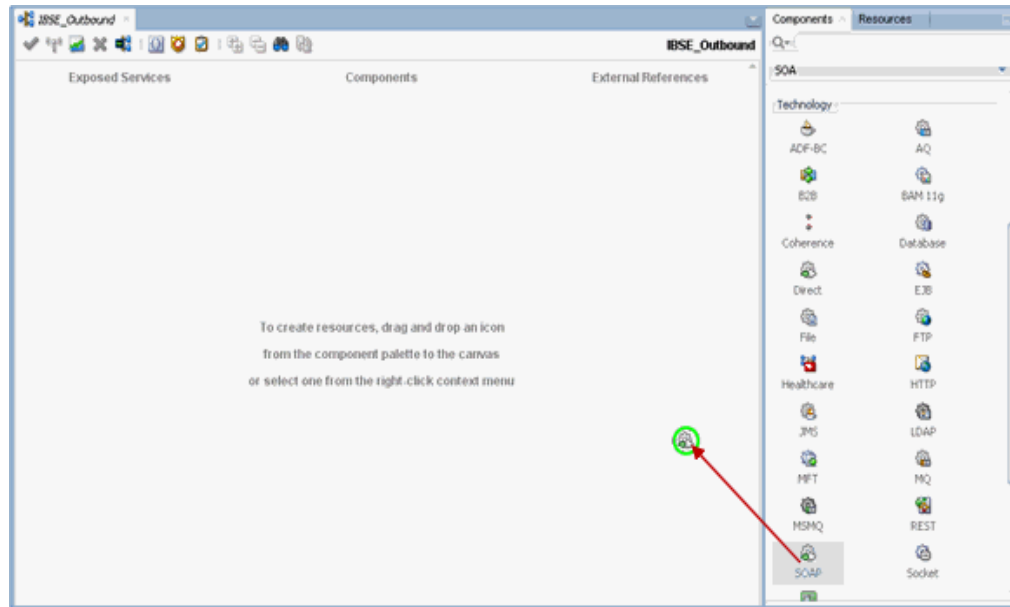
- [Section 8.7.2.1, "Configuring a Web Service Component"](#)
- [Section 8.7.2.2, "Configuring a BPM Process Component"](#)
- [Section 8.7.2.3, "Creating a File Adapter for the Write Operation"](#)

8.7.2.1 Configuring a Web Service Component

Perform the following steps to configure a Web Service component:

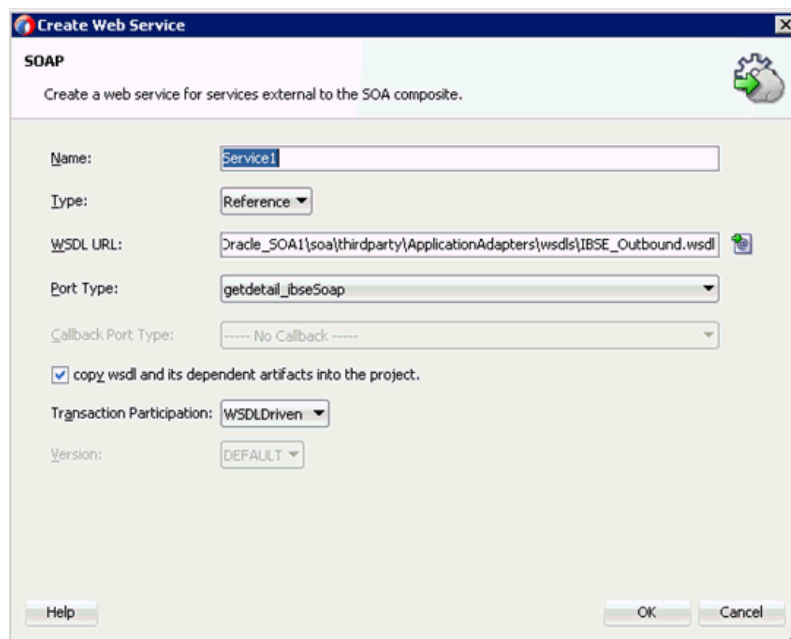
1. Double-click the created project to load the components.
2. Drag and drop the **Web Service** node from the Technology Adapters pane to the External References pane, as shown in [Figure 8-65](#).

Figure 8–65 Web Service Node



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 8–66](#).

Figure 8–66 Web Service Pane



7. In the displayed Localize Files window, click **OK**.
This will import the WSDL file to the project folder

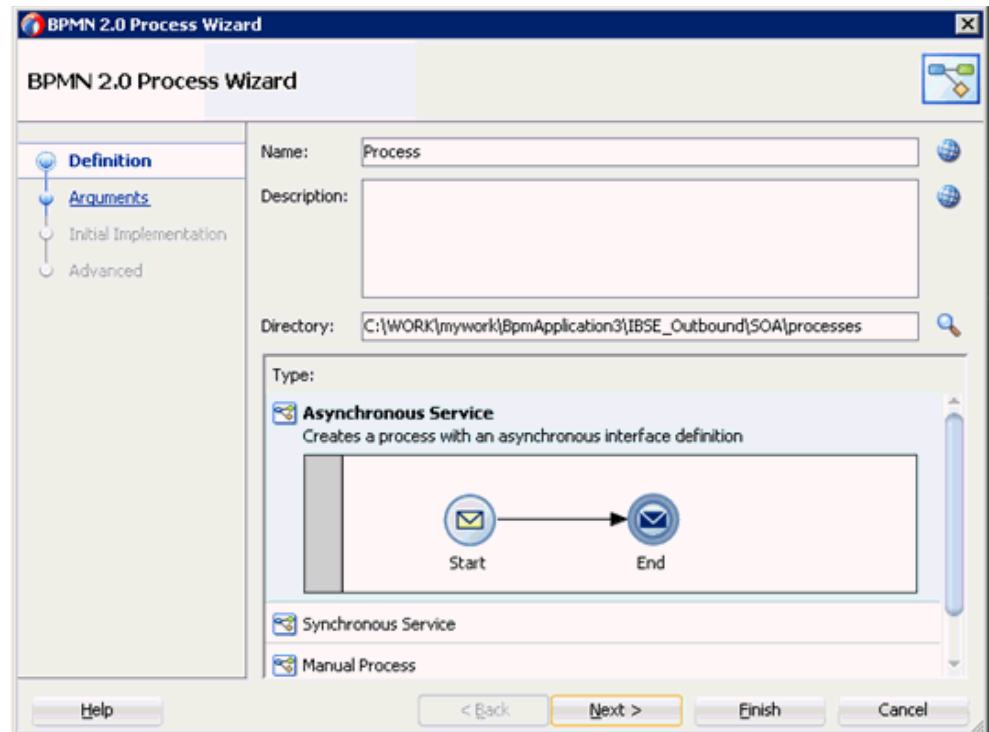
8.7.2.2 Configuring a BPM Process Component

This section describes how to configure an outbound BPM process component.

Perform the following steps to configure a BPM Component:

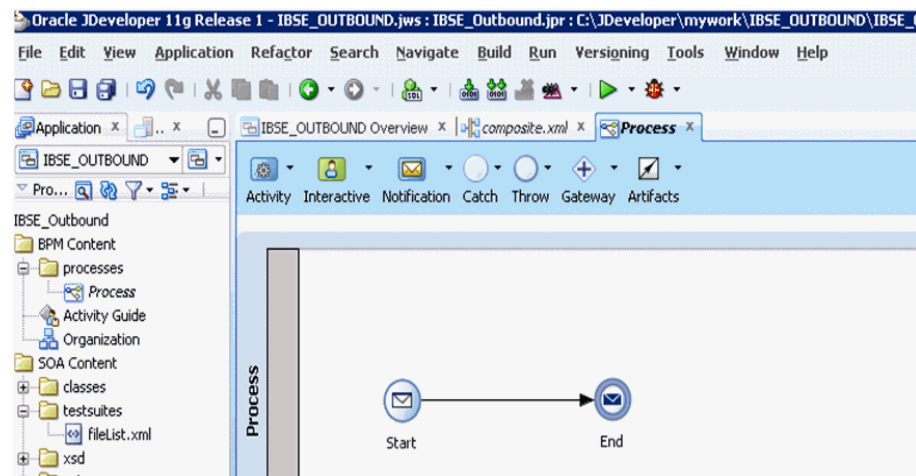
1. Drag and drop the **BPMN Process** component from the Components pane in to the Components pane.
2. Accept the default option that is selected under the Type area (Asynchronous Service) and click **Finish**, as shown in [Figure 8–67](#).

Figure 8–67 Type Area



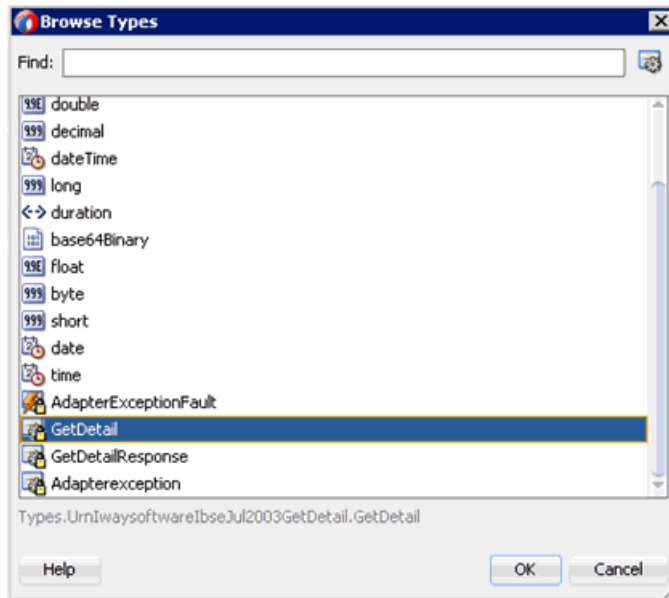
3. Double click on the Start Event component, as shown in [Figure 8–68](#).

Figure 8–68 Start Event Component



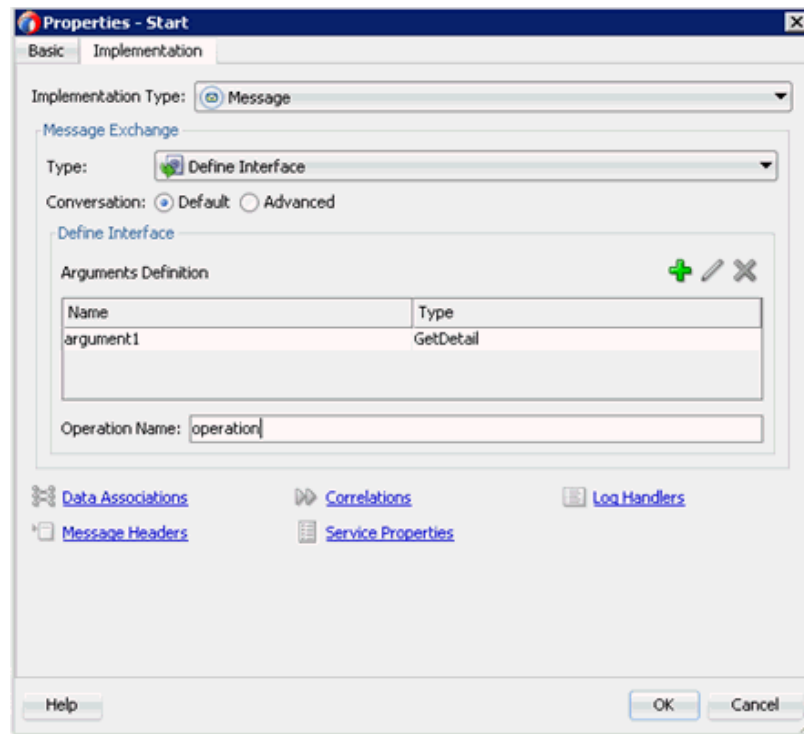
4. In the displayed Properties-start window, click the **Implementation** tab.
5. Click the Plus (+) icon to the right of the Arguments Definition field.
The Edit Argument window is displayed.
6. Enter a name in the Name field, and then click the Type drop-down list and select **Browse**.
7. Select the **Request** component (for example, GetDetail), and click **OK**, as shown in [Figure 8–69](#).

Figure 8–69 Request Component



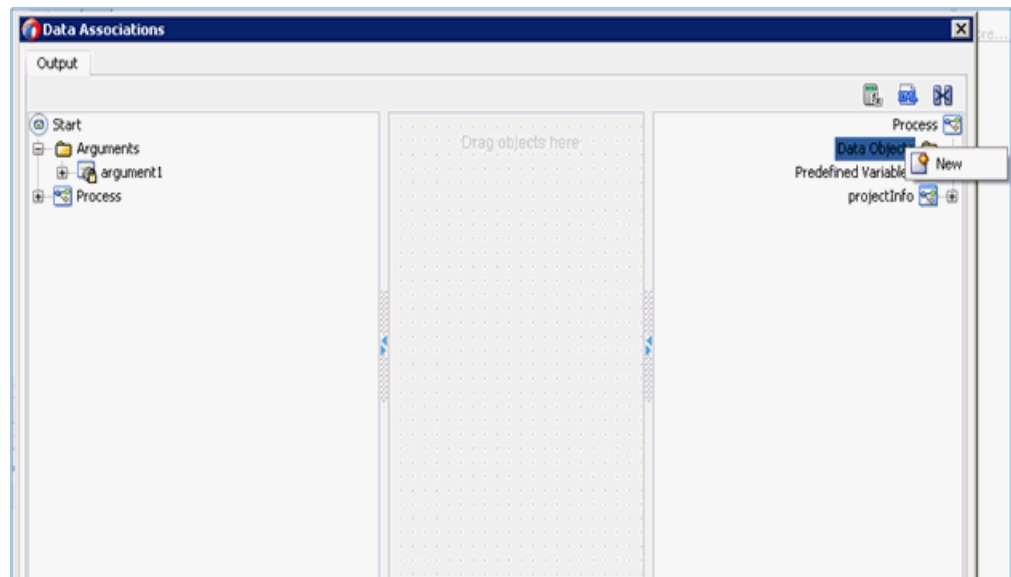
8. In the Edit Argument window that is displayed, click **OK**.
The Properties - Start window is displayed.
9. In the Operation Name field, change the default entry from **start** to **operation**.
10. Click the **Data Associations** hyperlink, as shown in [Figure 8–70](#).

Figure 8–70 Properties - Start Window



- Right-click the **Data Objects** node in the right pane, under Process, and select **New**, as shown in Figure 8–71.

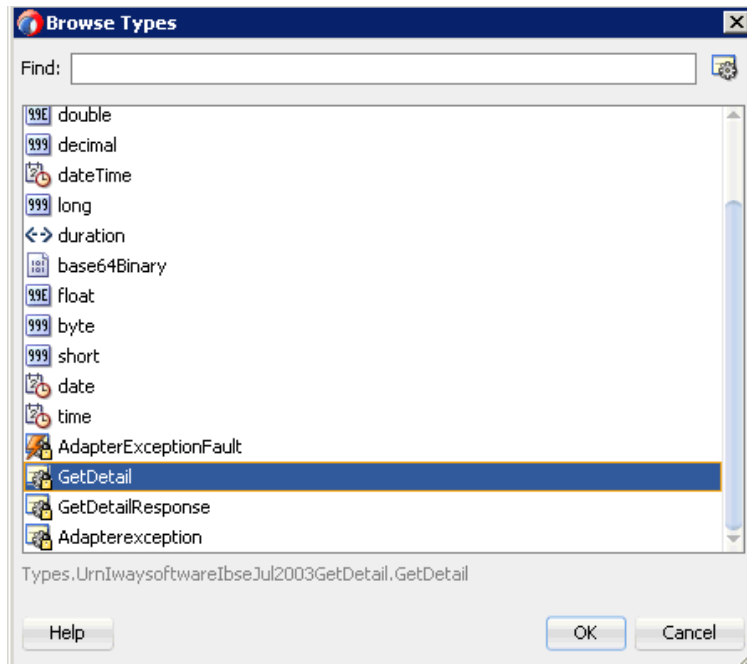
Figure 8–71 Data Objects Node



The Create Data Object window is displayed.

- Enter a name in the Name field, click the Type drop-down list, and select **Browse**.
- Select the **Request** component (for example, GetDetail) and click **OK**, as shown in Figure 8–72.

Figure 8–72 Request Component (GetDetail)

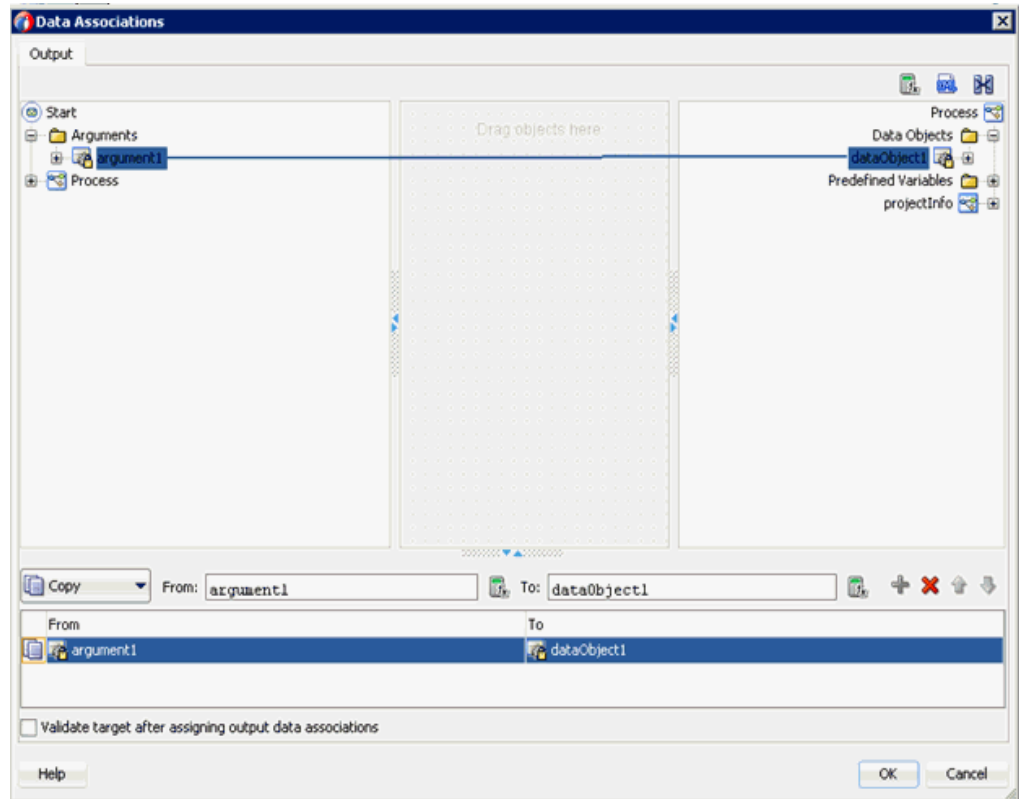


14. In the Create Data Object window, click **OK**.

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 8–73](#).

Figure 8–73 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**.

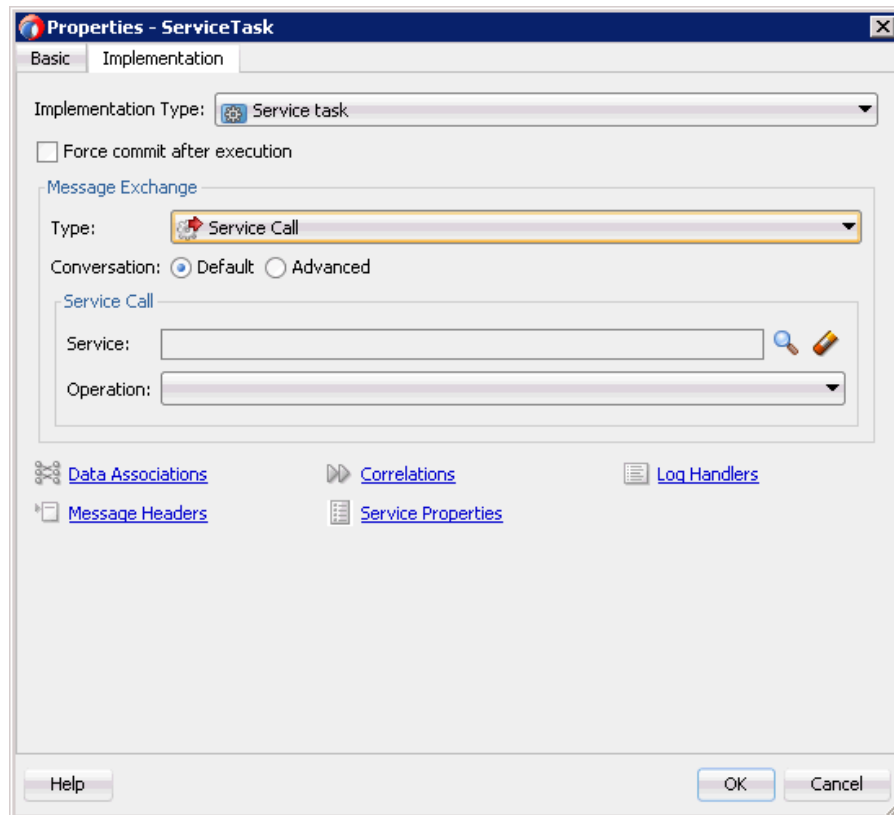
19. Drop the **Service** icon on the wire between the **Start** and **End** event components.

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 8–74](#).

Figure 8–74 Browse Icon



The Service window is displayed.

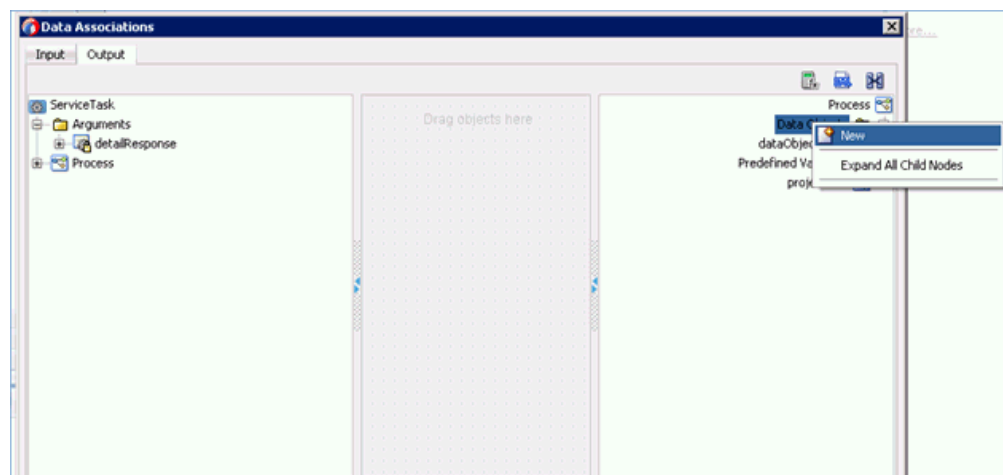
23. Select the Web Service that has been created and click **OK**, as shown in [Figure 8–75](#).

Figure 8–75 Created Web Service

24. In the Properties - ServiceTask window that is displayed, click the **Data Associations** hyperlink.

The Data Associations window is displayed.

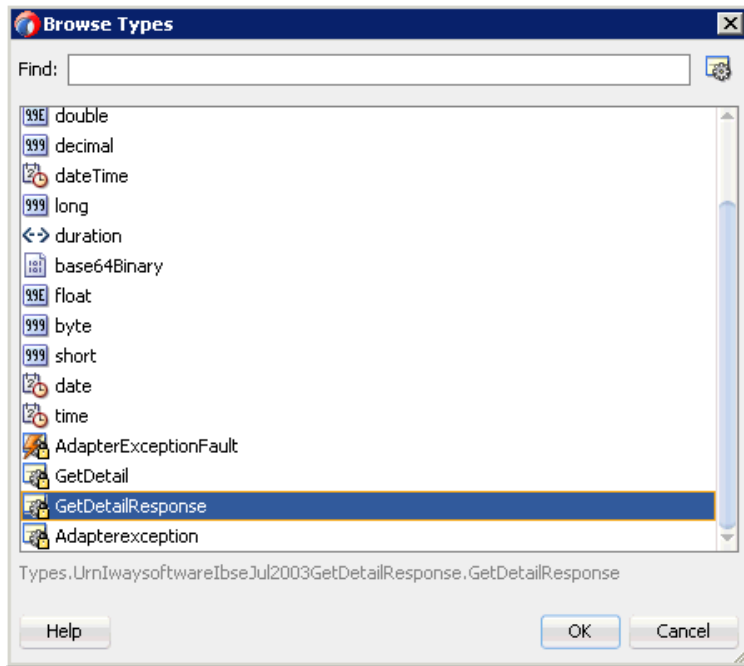
25. Create response Data Object by right-clicking the **Data Objects** node in the right pane of the Output tab and selecting **New**, as shown in Figure 8–76.

Figure 8–76 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, GetDetailResponse) and click **OK**, as shown in Figure 8–77.

Figure 8–77 Response Component (GetDetailResponse)

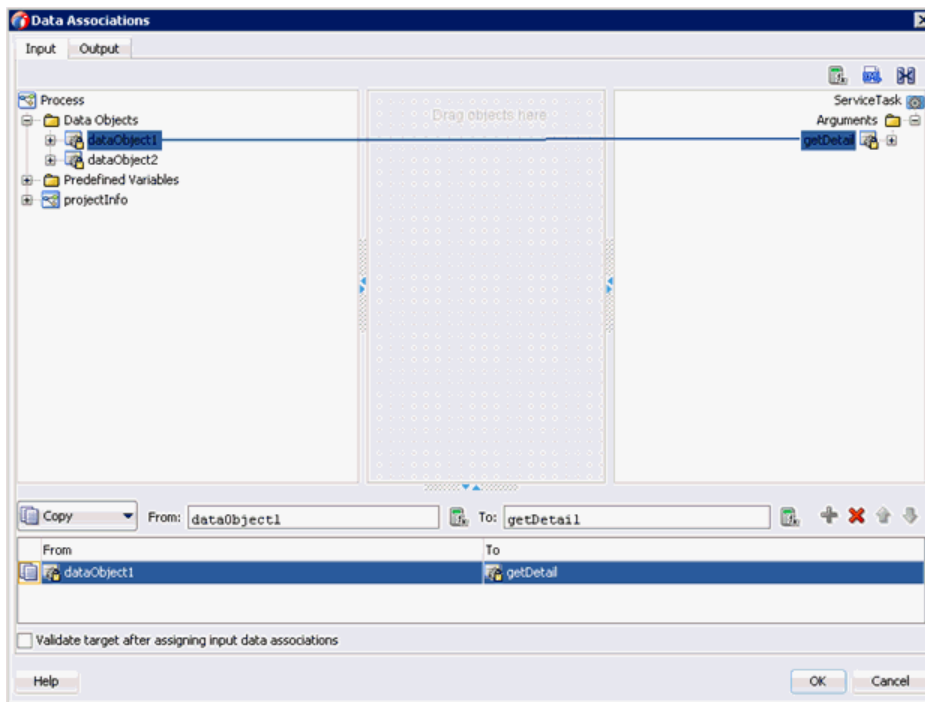


28. In the Create Data Object window, click **OK**.

The Data Associations window is displayed.

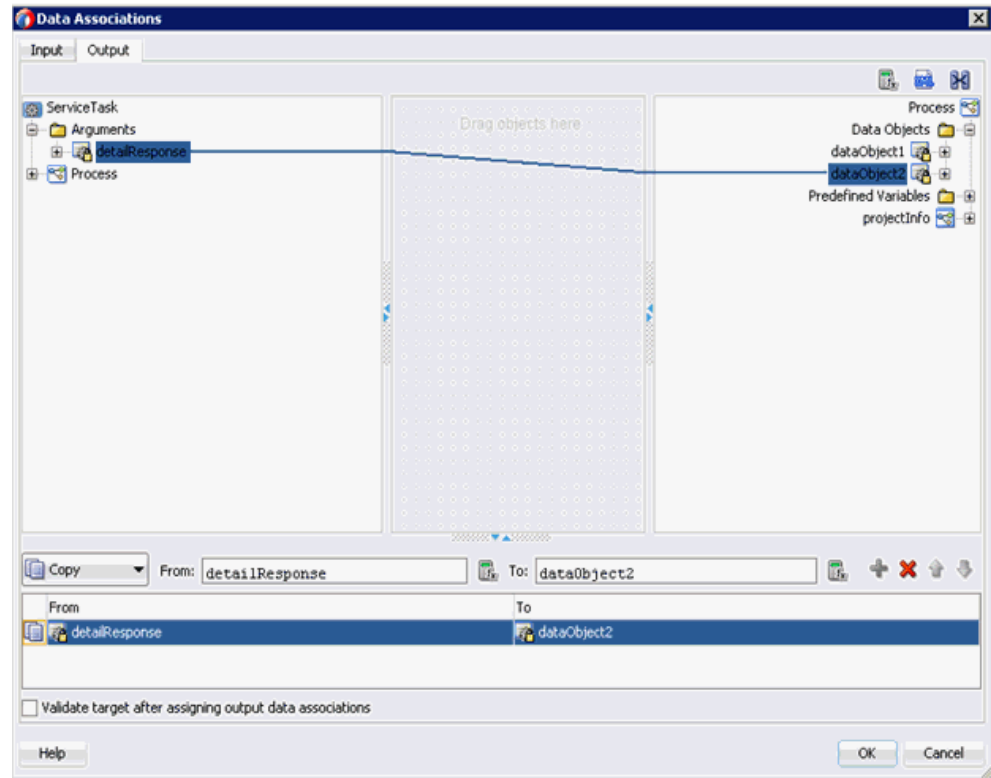
29. Select **dataObject1**, under the Data Objects node in the left pane of the Input tab, and drag and connect it to the **getDetail** node, under the Arguments node in the right pane, as shown in [Figure 8–78](#).

Figure 8–78 Data Associations



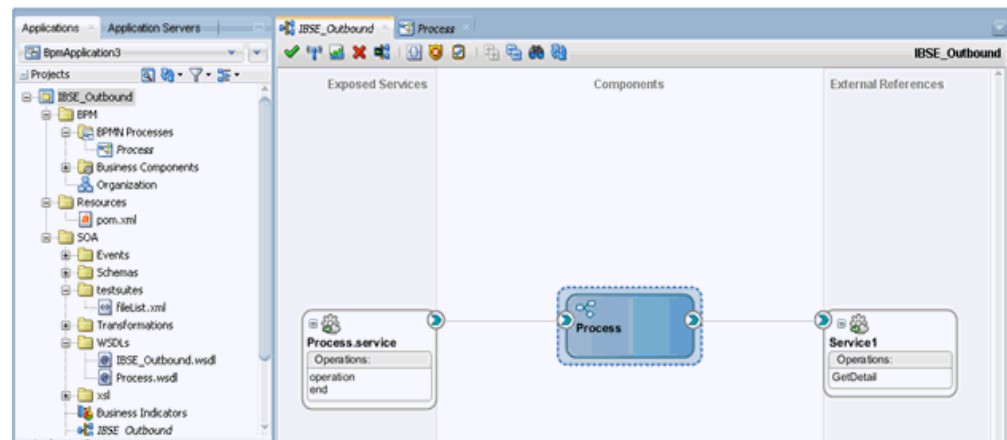
30. Click on the **Output** tab and select **detailResponse** under the Arguments node in the left pane and drag and connect it to **dataObject2** under the Data Objects node.
31. Click **OK**, as shown in [Figure 8–79](#).

Figure 8–79 Output Tab



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 **IBSE_Outbound** node in the left pane, as shown in [Figure 8–80](#).

Figure 8–80 Composite.xml Node



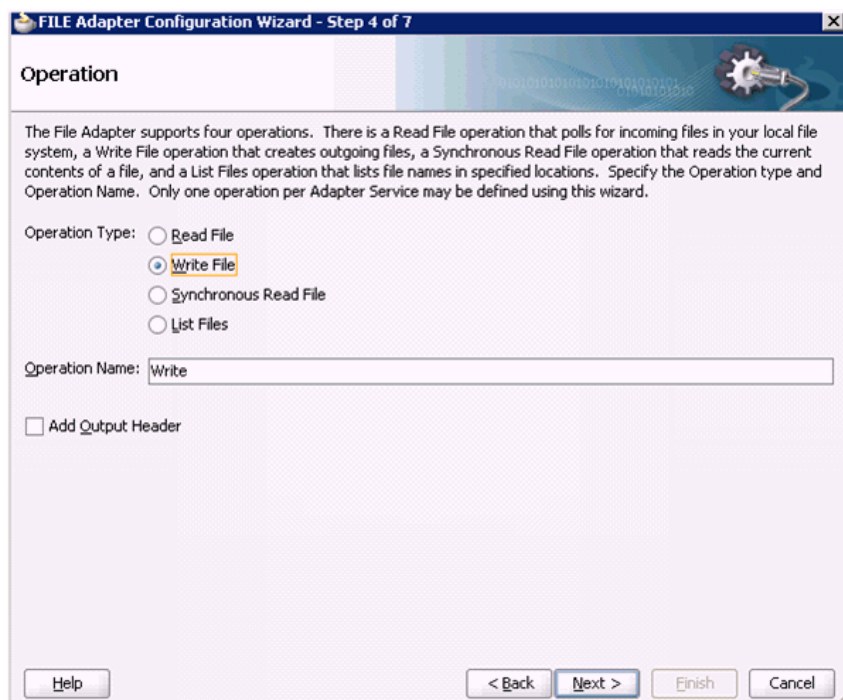
8.7.2.3 Creating a File Adapter for the Write Operation

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 8–81](#).

Figure 8–81 Operation Pane



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 8–82](#).

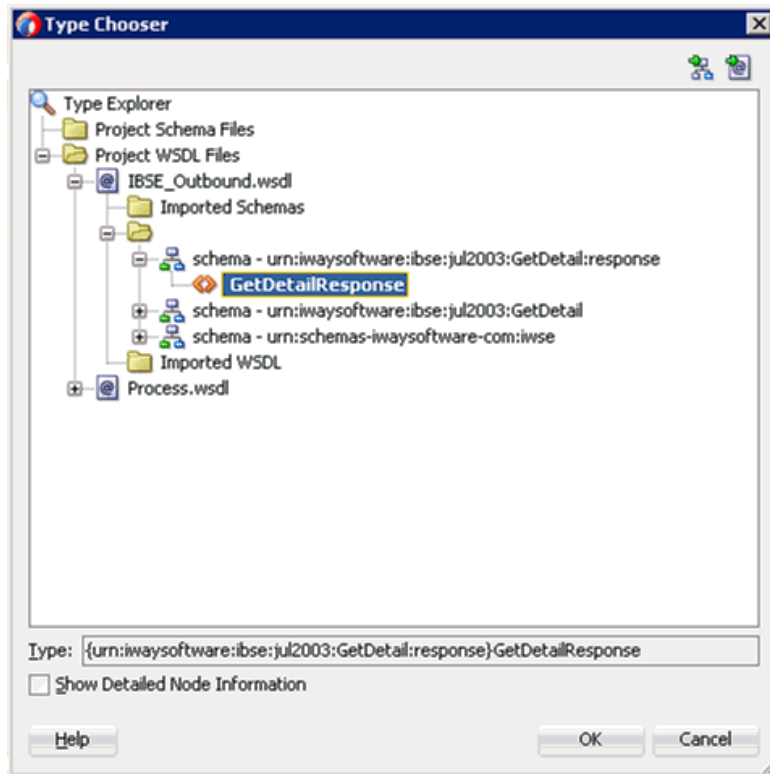
Figure 8–82 File Configuration Pane

The screenshot shows the 'FILE Adapter Configuration Wizard - Step 5 of 7' window. The title bar includes a gear icon and a close button. The main window has a header 'File Configuration' and a sub-header 'Specify the parameters for the Write File operation.' Below this, there are two radio buttons: 'Physical Path' (selected) and 'Logical Name'. A text field labeled 'Directory for Outgoing Files (physical path):' contains 'C:\output' and a 'Browse' button is to its right. Below that, a text field for 'File Naming Convention (po_%SEQ%.txt):' contains 'IBSE_OUTBOUND.log'. There is an unchecked checkbox for 'Append to existing file'. A section titled 'Write to output file when any of these conditions are met:' contains three rows: 'Number of Messages Equals:' with a checked checkbox, a text field containing '1', and a spinner; 'Elapsed Time Exceeds:' with an unchecked checkbox, a text field containing '1', a spinner, and a dropdown menu set to 'minutes'; and 'File Size Exceeds:' with an unchecked checkbox, a text field containing '1000', a spinner, and a dropdown menu set to 'kilobytes'. At the bottom, there are buttons for 'Help', '< Back', 'Next >', 'Finish', and 'Cancel'.

The Messages pane is displayed.

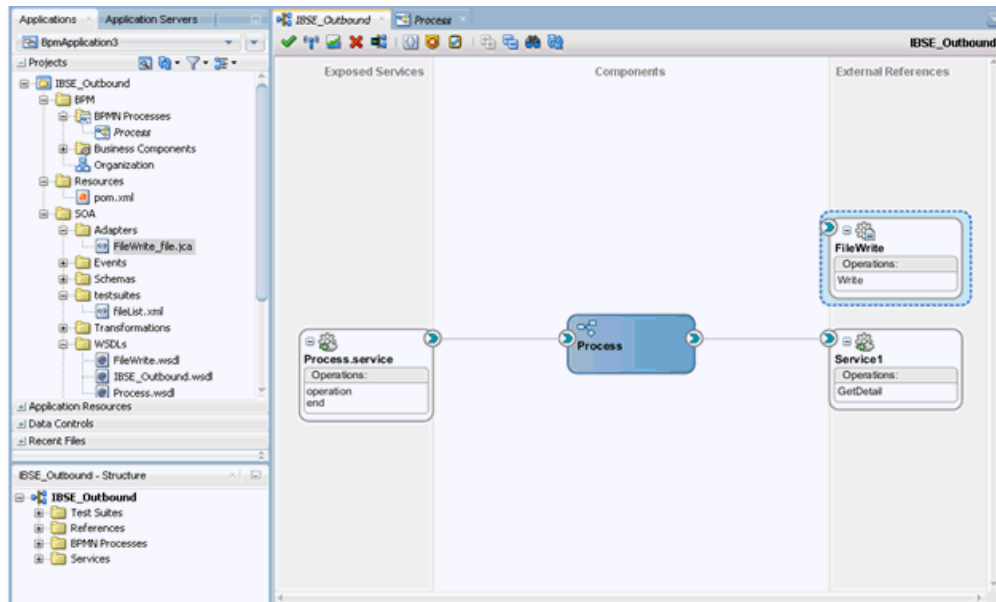
8. Click the **Browse**, which is located to the right of the URL field.
9. In the displayed Type Chooser window, expand **Project WSDL Files, IBSE_Outbound.wsdl, Inline Schemas** and then select **GetDetailresponse**.
10. Click **OK**, as shown in [Figure 8–83](#).

Figure 8–83 Type Chooser Window



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 8–84](#).

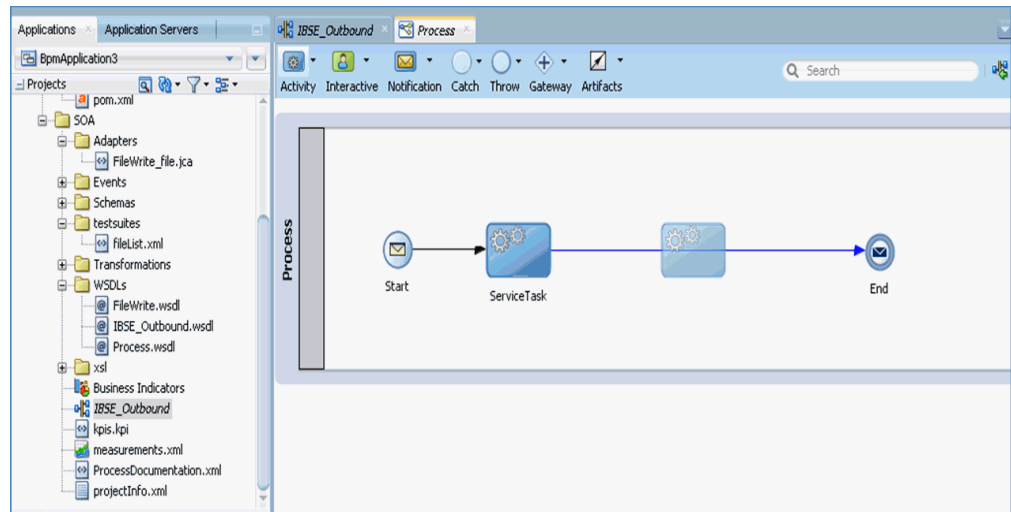
Figure 8–84 Composite.xml Tab



14. Click the **Activity** icon.

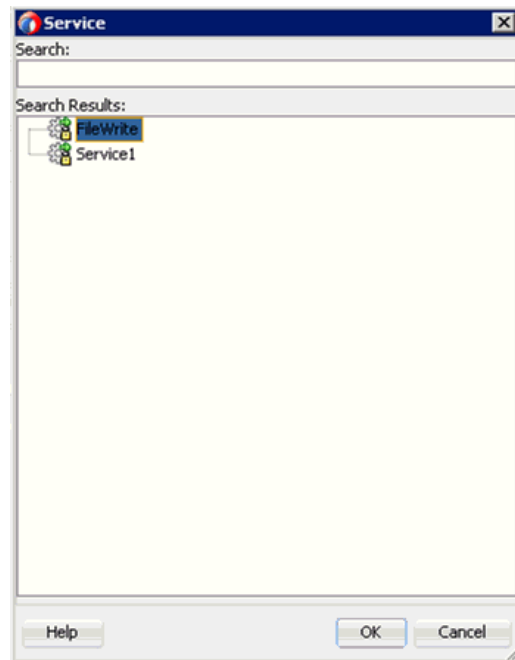
15. Drop the **Activity** icon on the wire between the **Service Task** and **End** event components, as shown in [Figure 8–85](#).

Figure 8–85 Activity Icon



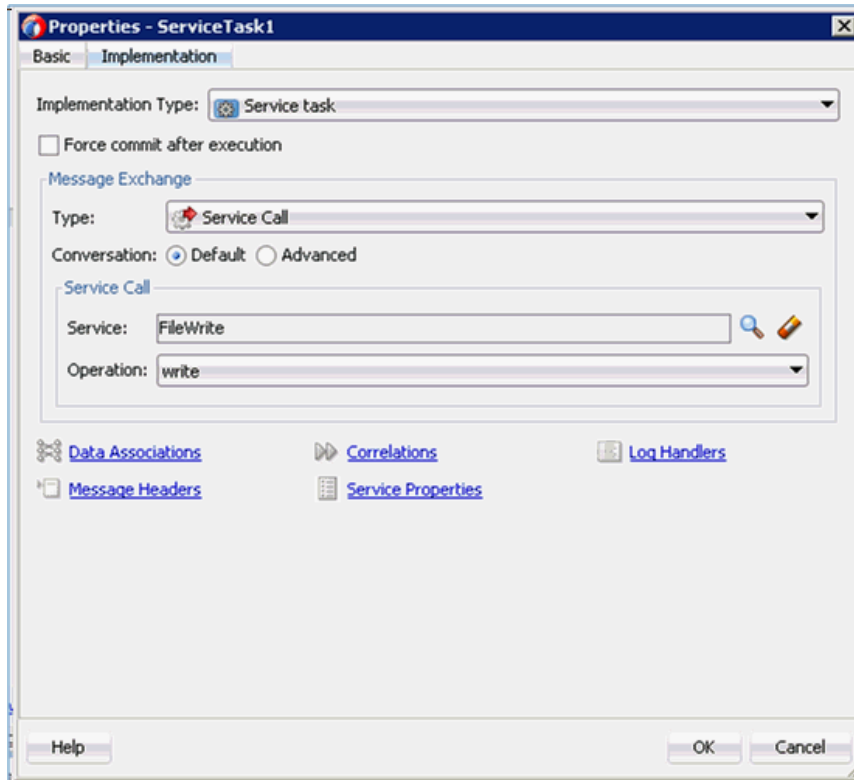
16. In the displayed Properties-ServiceTask1 window, click the **Implementation** tab
17. Select **Service Call** from the Type drop-down list in the Message Exchange section.
18. Click the **Browse** icon to the right of the Service field.
19. Select the service for write operation that has been created and click **OK**, as shown in [Figure 8–86](#).

Figure 8–86 Service Window



20. In the Properties - ServiceTask1 window, click the **Data Associations** hyperlink, as shown in [Figure 8–87](#).

Figure 8–87 Data Associations



21. In the Input tab, click the **XSL Transformation** icon in the top right pane, as shown in [Figure 8–88](#).

Figure 8–88 XSL Transformation Icon



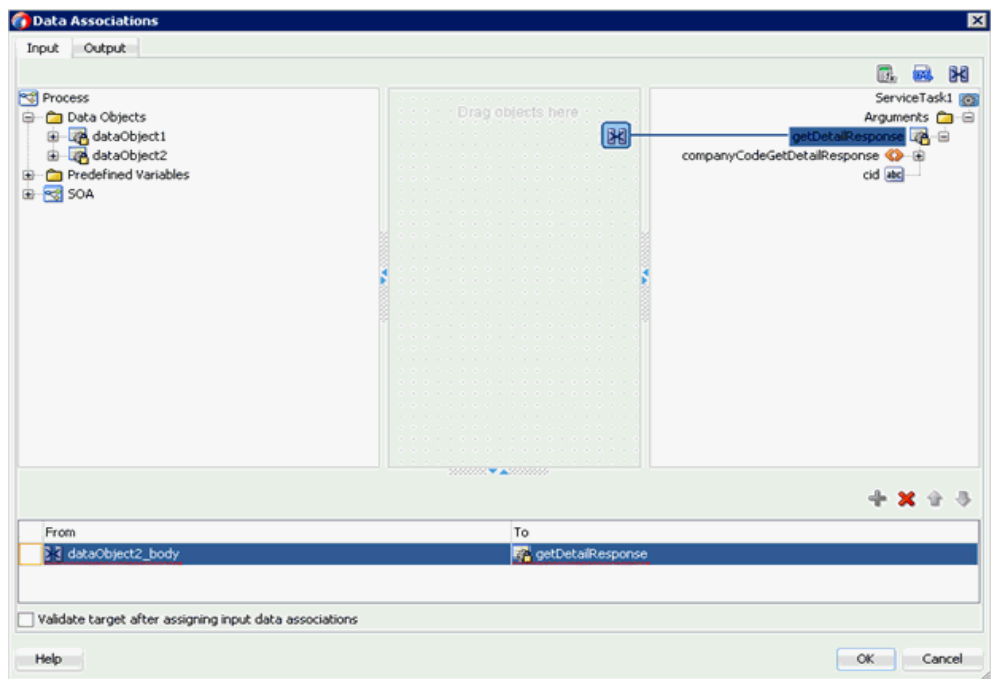
22. Drag and drop the **XSL Transformation** icon to the `getDetailResponse` node, as shown in [Figure 8–89](#).

Figure 8–89 GetDetailResponse Node



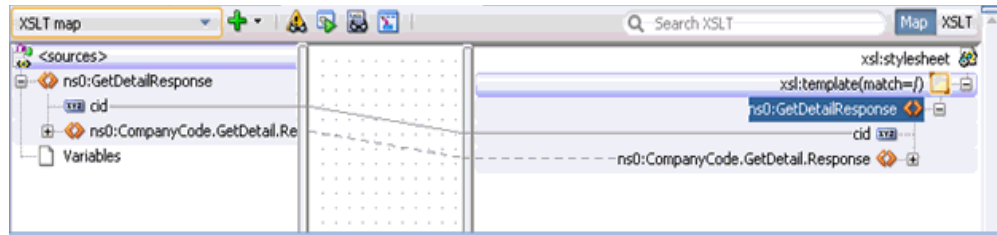
23. In the displayed Create Transformation window, select **dataObject2** in the Sources section and click the right arrow symbol.
24. Accept the default value selected in the Target drop-down list and the default name in the Create field by clicking **OK**.
25. In the Data Associations window, click **OK**, as shown in [Figure 8–90](#).

Figure 8–90 Data Associations Window



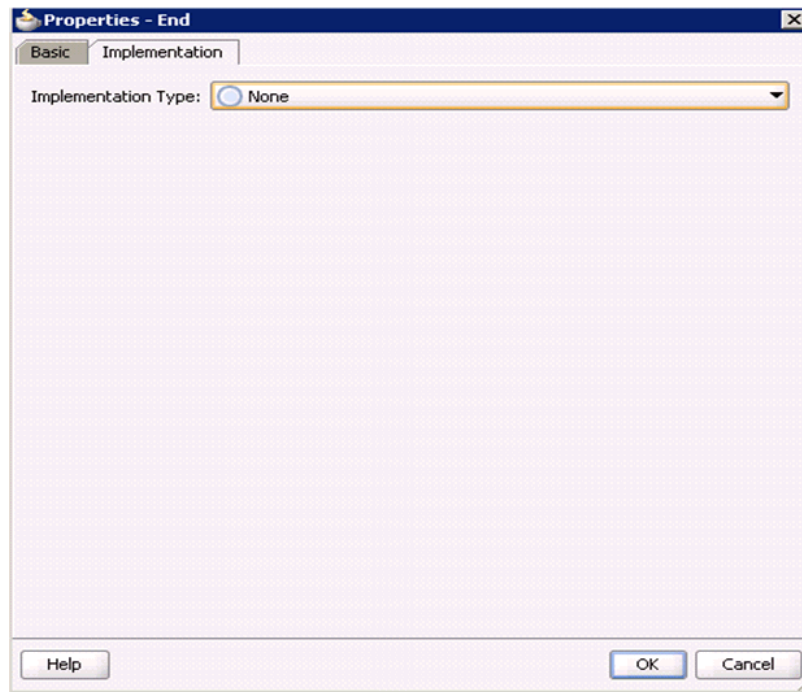
26. In the Properties - ServiceTask1 window, click **OK**.
27. In the response_body.xml tab, map the **ns0:GetDetailResponse** source element to the **ns0:GetDetailResponse** target element.
28. In the displayed Auto Map Preferences window, retain the default values and click **OK**, as shown in [Figure 8–91](#).

Figure 8–91 Auto Map Preferences

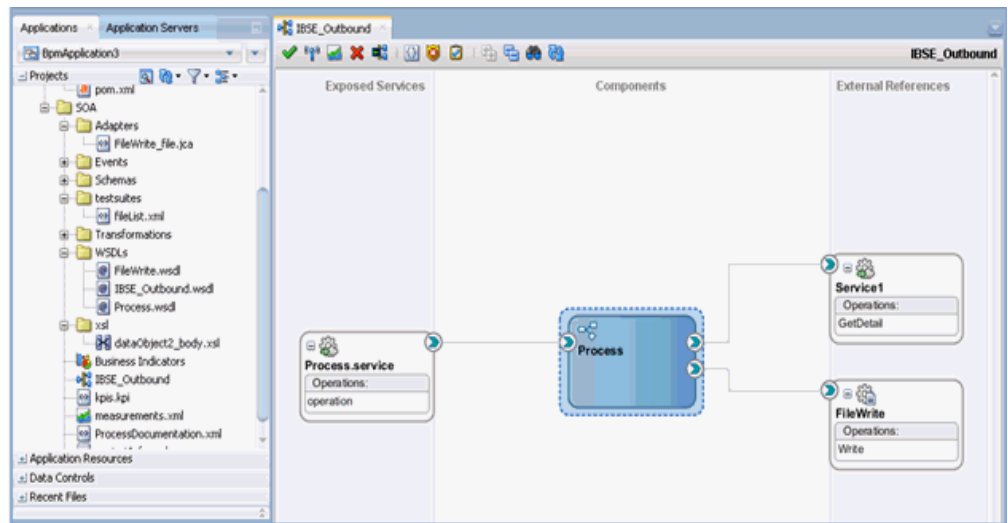


29. Return to the Process workspace area and double-click the **End** event component.
30. In the displayed Properties - End window, click the **Implementation** tab.
31. Select **None** from the Implementation Type drop-down list.
32. Click **OK**, as shown in [Figure 8–92](#).

Figure 8–92 Implementation Tab



33. Click the **Save All** icon in the menu bar to save the new outbound BPM component that was configured, as shown in [Figure 8–93](#).

Figure 8–93 Save All Icon

You are now ready to deploy the BPM BSE Outbound process. You can follow the same procedure as [Section 8.4.4, "Deploying the BPM Outbound Process"](#).

Once deployed, you can invoke the input XML as defined in [Section 8.4.5, "Invoking the Input XML Document in the Oracle Enterprise Manager Console"](#).

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

Note: With Release 12c (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 SAP R/3 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 9.1, "Overview of Application Adapter Integration with Oracle Service Bus"](#)
- [Section 9.2, "Configuring an Outbound Process Using sbconsole \(J2CA Configuration\)"](#)
- [Section 9.3, "Configuring an Inbound Process Using sbconsole \(J2CA Configuration\)"](#)
- [Section 9.4, "Configuring an Outbound Process Using sbconsole \(BSE Configuration\)"](#)
- [Section 9.5, "Working With Synchronous SAP R/3 Events Using Oracle Service Bus \(J2CA Configuration\)"](#)
- [Section 9.6, "Configuring JMS Proxy Services Using Oracle Service Bus \(J2CA Configuration\)"](#)
- [Section 9.7, "Configuring HTTP Proxy Services Using Oracle Service Bus \(J2CA Configuration\)"](#)

9.1 Overview of Application Adapter Integration with Oracle Service Bus

To integrate with Oracle Service Bus (OSB), Oracle Application Adapter for SAP R/3 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.

9.2 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\SAP_Samples.zip\SAP_Samples\OSB\J2CA\SAP_Sample_J2CA_
OSB_Outbound_Project
```

This section includes the following topics:

- [Section 9.2.1, "Starting Oracle Service Bus and Creating Project Folders"](#)
- [Section 9.2.2, "Setting the Class Path for Application Explorer to Integrate With Oracle Service Bus"](#)
- [Section 9.2.3, "Publishing a WSDL From Application Explorer to Oracle Service Bus"](#)
- [Section 9.2.4, "Configuring a WSDL-based Business Service"](#)
- [Section 9.2.5, "Configuring a File Type Business Service"](#)
- [Section 9.2.6, "Configuring a Pipeline With Proxy Service"](#)

9.2.1 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:

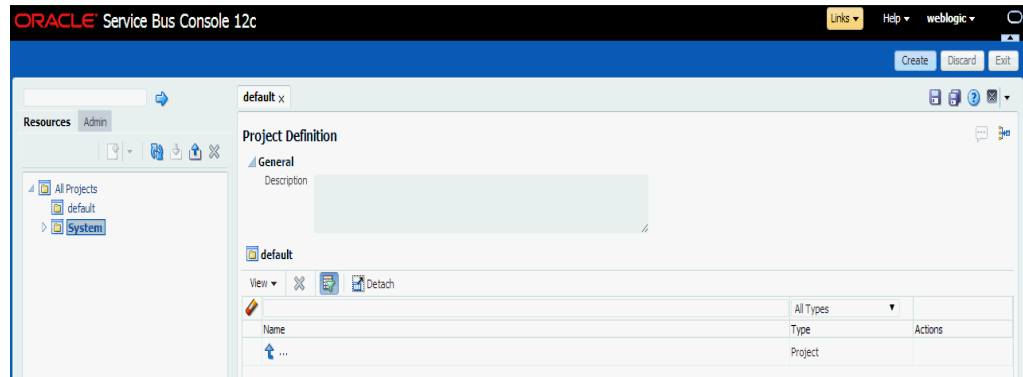
```
http://hostname:port/sbconsole
```

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.

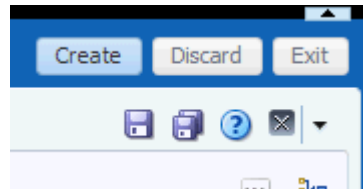
3. Log on to the Oracle Service Bus Console using a valid user name and password. The Oracle Service Bus Console home page is displayed, as shown in [Figure 9–1](#).

Figure 9-1 Oracle Service Bus Console Home Page



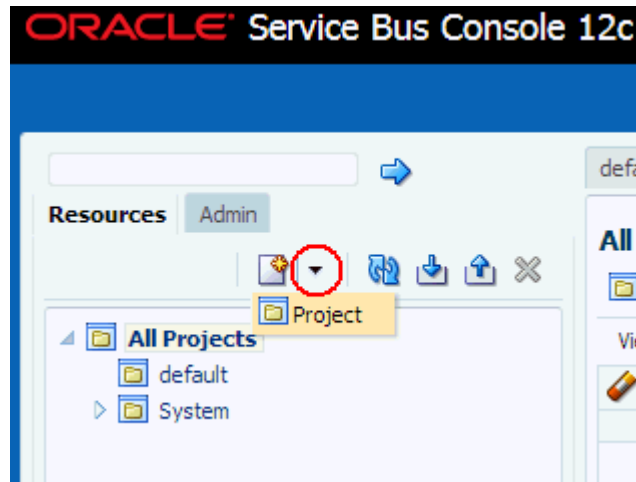
4. Click **Create** in the right pane of the Oracle Service Bus session, as shown in [Figure 9-2](#).

Figure 9-2 Oracle Service Bus Session



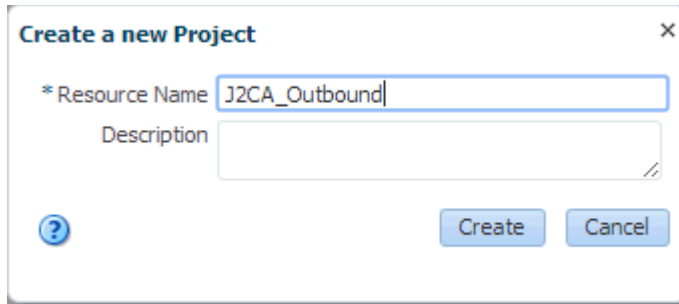
5. Select **All Projects**, click the down arrow in the left pane, and select **Project**, as shown in [Figure 9-3](#).

Figure 9-3 All Projects Folder



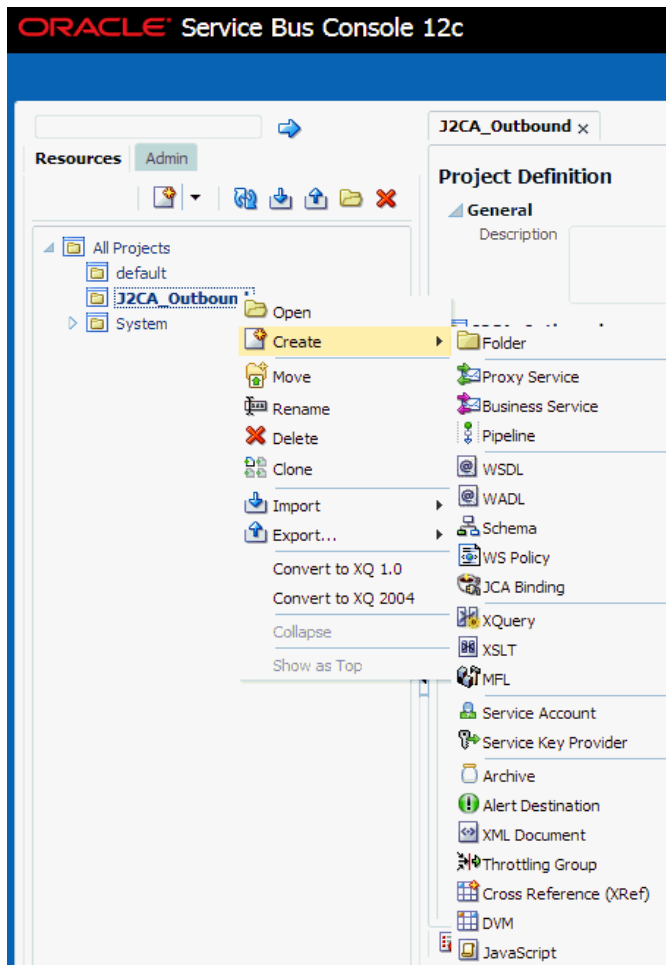
The Create a new Project window is displayed, as shown in [Figure 9-4](#).

Figure 9–4 Create New Project Window



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 9–5.

Figure 9–5 Create Option



The Create a new Folder window is displayed.

8. In the Resource Name field, type **Business Service** and click **Create**.

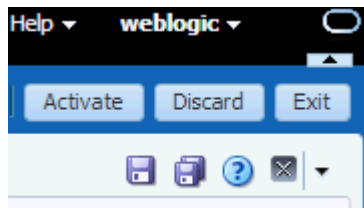
9. Repeat steps 7 and 8 to create folders with the names **Proxy Service** and **Wsdls**.
The Business Service, Proxy Service, and Wsdls folders are listed in the left pane below the project node, as shown in [Figure 9-6](#).

Figure 9-6 Project Node



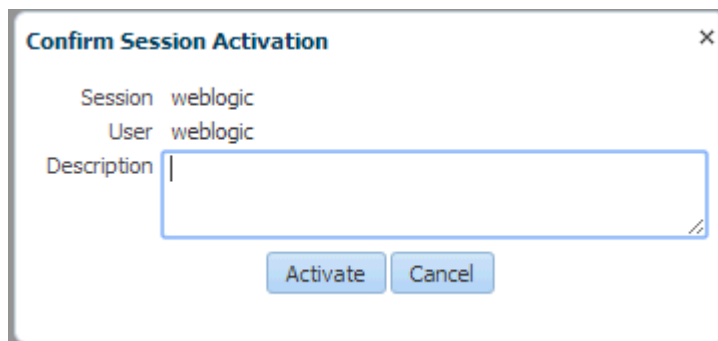
10. Click **Activate** in the right pane of the Oracle Service Bus session, as shown in [Figure 9-7](#).

Figure 9-7 Activate Button



11. In the Confirm Session Activation page, click **Activate** to save the changes, as shown in [Figure 9-8](#).

Figure 9-8 Confirm Session Activation Window



9.2.2 Setting the Class Path for Application Explorer to Integrate With Oracle Service Bus

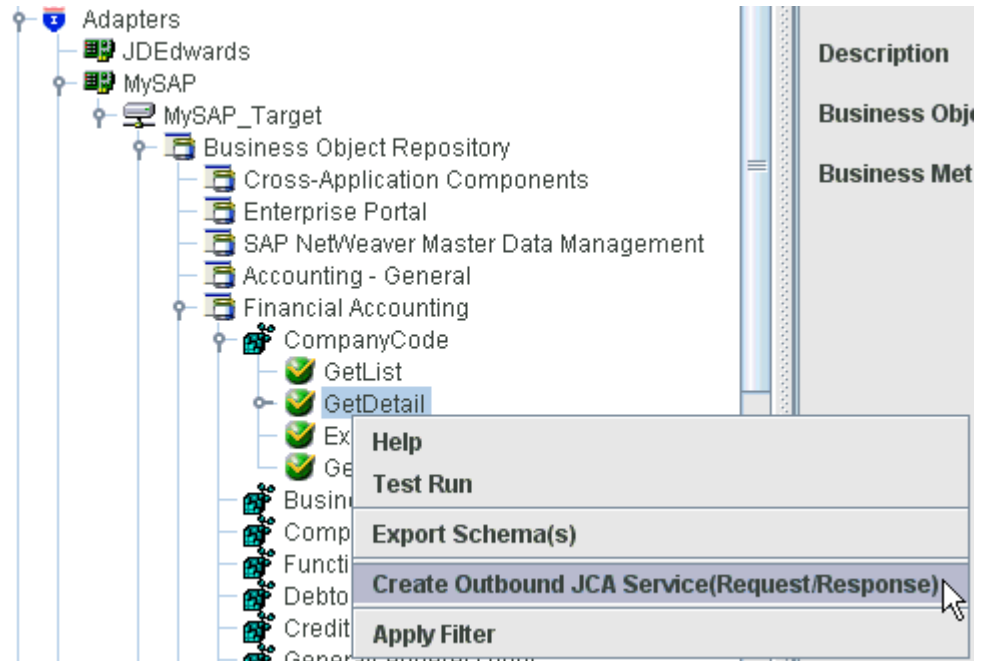
Before starting and using Application Explorer to publish a WSDL directly to the Oracle Service Bus (OSB) Console (project/folder), OSB users must perform the following steps:

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.

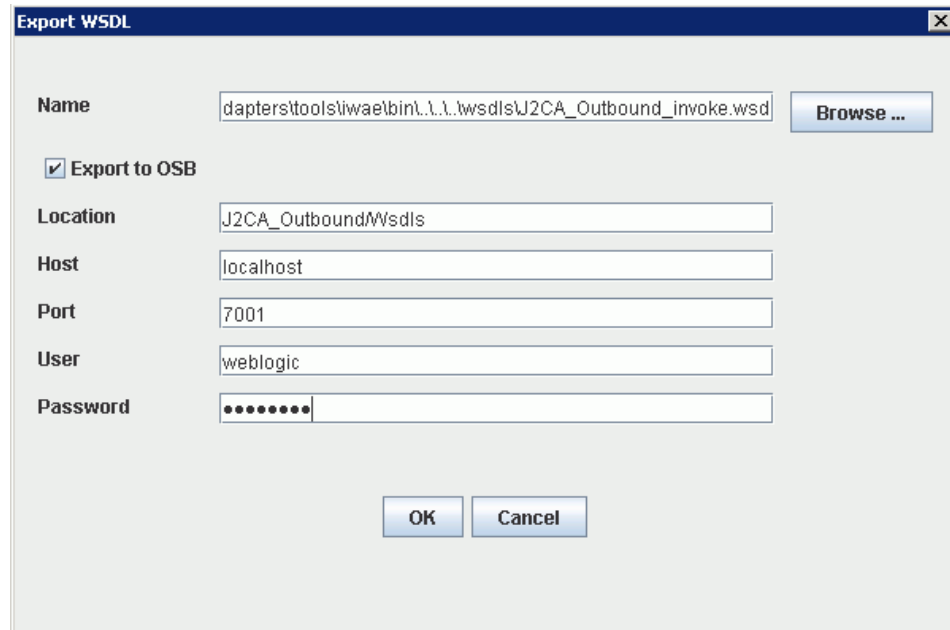
9.2.3 Publishing a WSDL From Application Explorer to Oracle Service Bus

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 SAP R/3 target.
For more information, see [Chapter 4, "Configuring Oracle Application Adapter for SAP R/3"](#).
2. Expand **Business Object Repository, Financial Accounting**, and then **CompanyCode**.
3. Right-click the **GetDetail** method and select **Create Outbound JCA Service(Request/Response)** from the menu, as shown in [Figure 9–9](#).

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

The Export WSDL dialog is displayed, as shown in [Figure 9–10](#).

Figure 9–10 Export WSDL Dialog

4. In the Name field, a default file name for the WSDL file is provided. You can accept the default or provide your own.
5. Select the **Export to OSB** option.
6. 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 “/”.

7. In the Host field, enter the name of the machine where Oracle Service Bus is installed.
8. In the Port field, enter the port that is being used by Oracle Service Bus.
9. In the User field, enter your username to access Oracle Service Bus.
10. In the Password field, enter your password to access Oracle Service Bus.
11. Click **OK**.

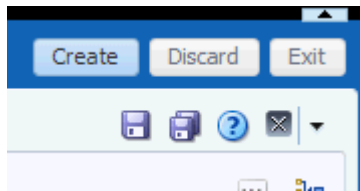
The WSDL is published to the location specified in the Export WSDL dialog and is now available for use with a Business Service or Proxy Service in Oracle Service Bus.

9.2.4 Configuring a WSDL-based Business Service

Perform the following steps to configure a WSDL-based Proxy Service:

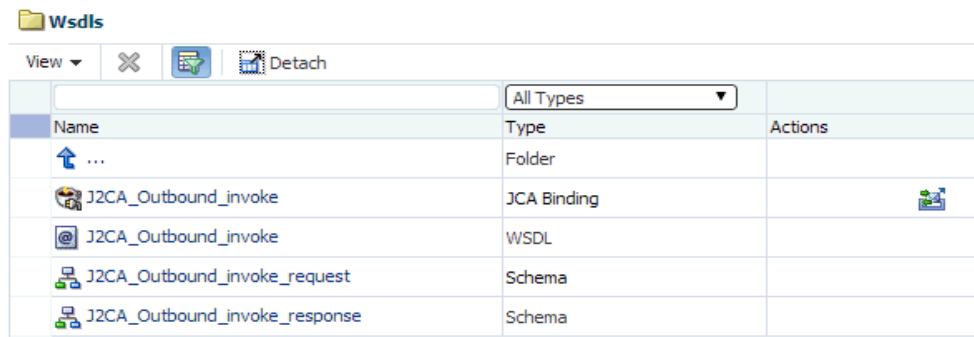
1. Open the Oracle Service Bus Console and click **Create** in the right pane of the Oracle Service Bus session, as shown in [Figure 9–11](#).

Figure 9–11 Create Button

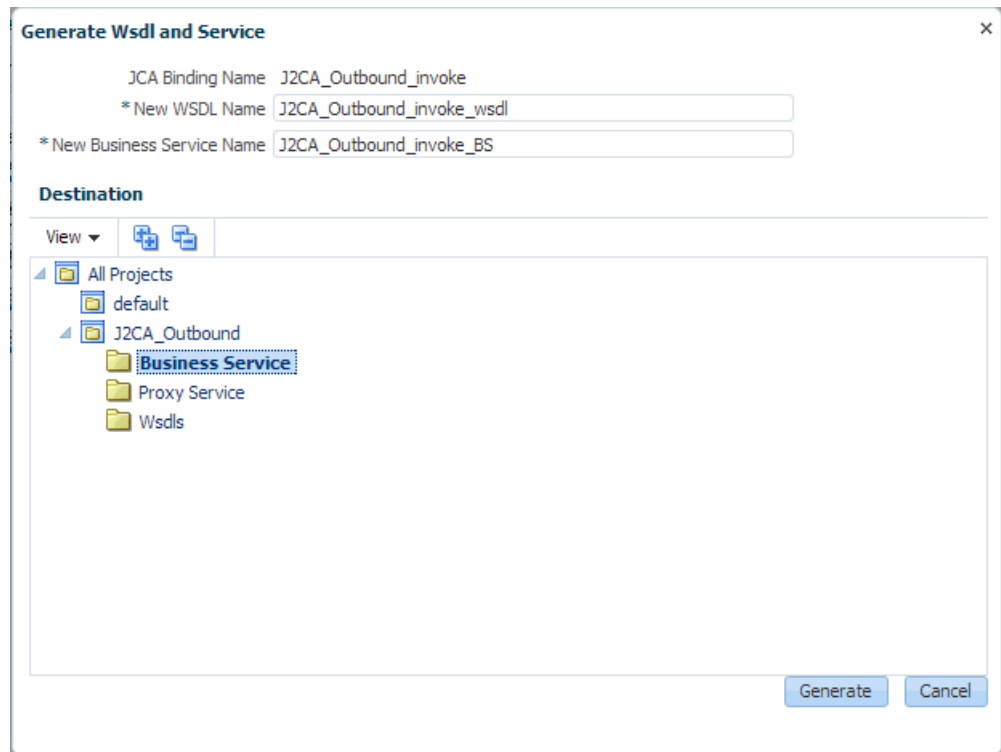


2. Double-click the created WSDL folder in the left pane (for example, Wsdls) and ensure that the exported WSDL is listed in the right pane, as shown in [Figure 9–12](#).

Figure 9–12 Wsdls Folder

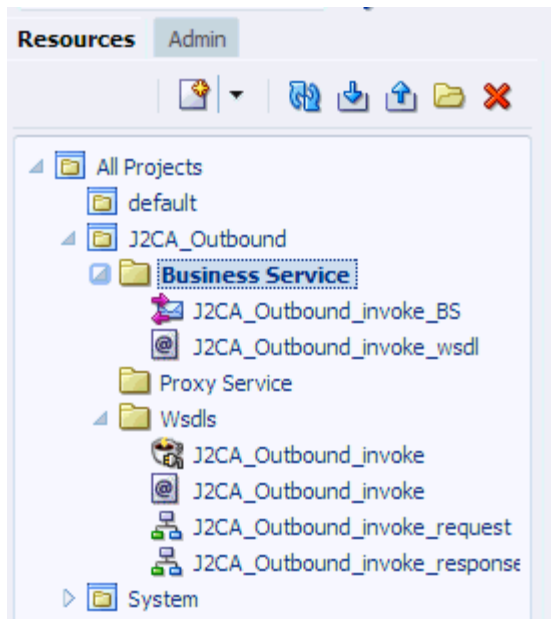


3. Click the icon that corresponds to the JCA Binding in the Actions column.
The Generate WSDL and Service window is displayed, as shown in [Figure 9–13](#).

Figure 9–13 Generate WSDL and Service Window

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 9–14](#).

Figure 9–14 Business Service Folder

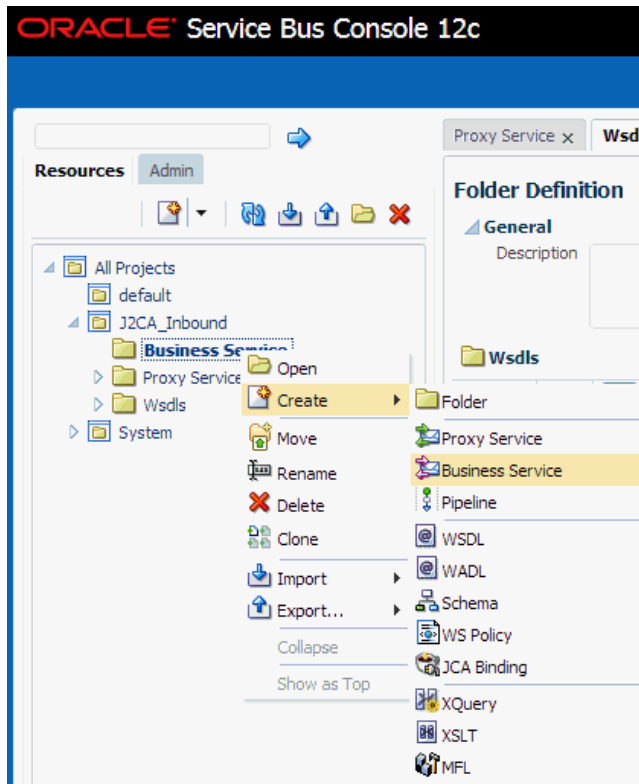


9.2.5 Configuring a File Type Business Service

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 9–15](#).

Figure 9–15 Business Service Folder



The Create Business Service window is displayed.

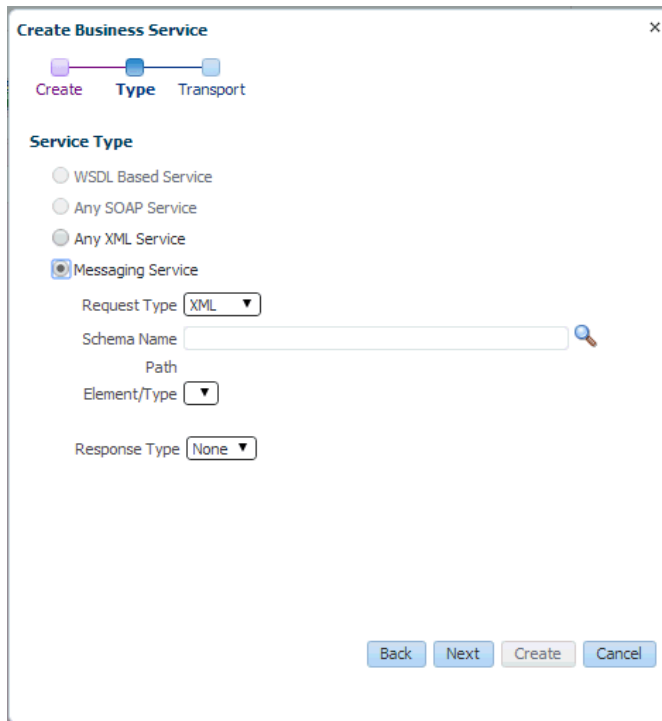
2. In the Resource Name field, provide a name for the Business Service, select the **File** option in the Transport section under Service Definition, and click **Next**, as shown in [Figure 9–16](#).

Figure 9–16 Service Definition

The screenshot shows the 'Create Business Service' dialog box. At the top, there is a progress bar with three steps: 'Create' (active), 'Type', and 'Transport'. Below the progress bar, the 'Create Service' section has a text field for '*Resource Name' containing 'File_Out' and a text area for 'Description'. The 'Service Definition' section has two radio buttons: 'WSDL Based Service' (unselected) and 'Transport' (selected). Under 'WSDL Based Service', there are fields for 'Name', 'Path', and 'Port/Binding'. Under 'Transport', there is a dropdown menu currently showing 'file'. At the bottom of the dialog are four buttons: 'Back', 'Next', 'Create', and 'Cancel'.

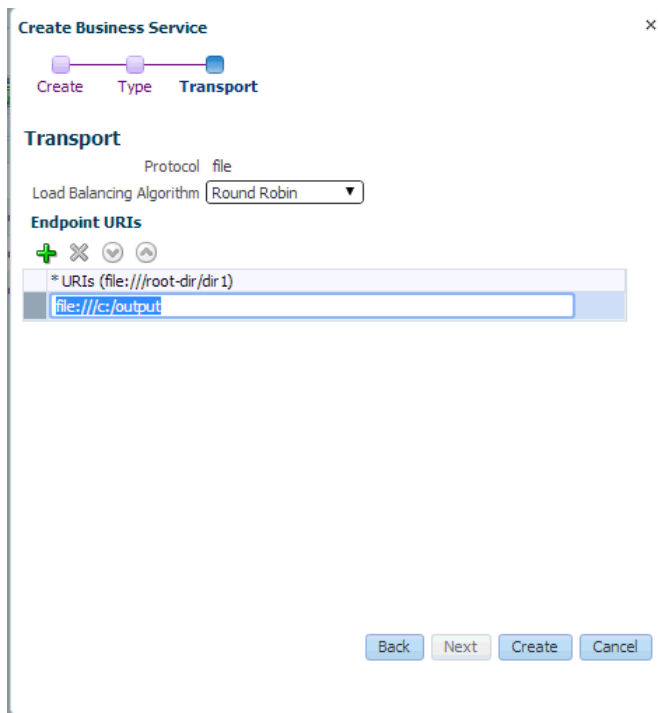
3. In the Service Type section, select **Messaging Service**. By default, the Request Type is set to XML, and the Response Type is set to None. Then click **Next**, as shown in [Figure 9–17](#).

Figure 9–17 Service Type Configuration Page



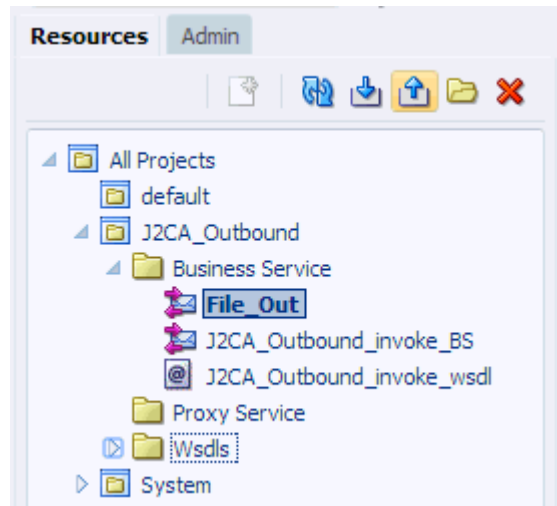
4. Enter the path to a destination folder on your file system in the Endpoint URI field.
5. Click **Create**, as shown in [Figure 9–18](#).

Figure 9–18 Transport Page



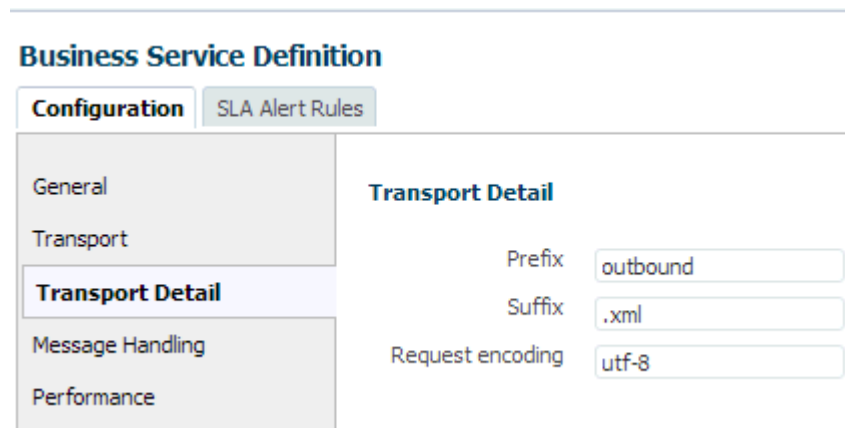
The Business Service **File_Out** is created and listed under Business Service, as shown in [Figure 9–19](#).

Figure 9–19 File_Out Business Service



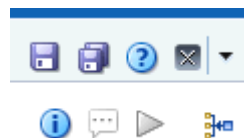
- Double-click **File_Out**, click **Transport Detail** in the left pane, and enter the prefix and suffix for the output file to be received, as shown in [Figure 9–20](#).

Figure 9–20 Transport Detail



- Click the **Save** or **Save All** icon in the right corner, as shown in [Figure 9–21](#).

Figure 9–21 Save/Save All Icons

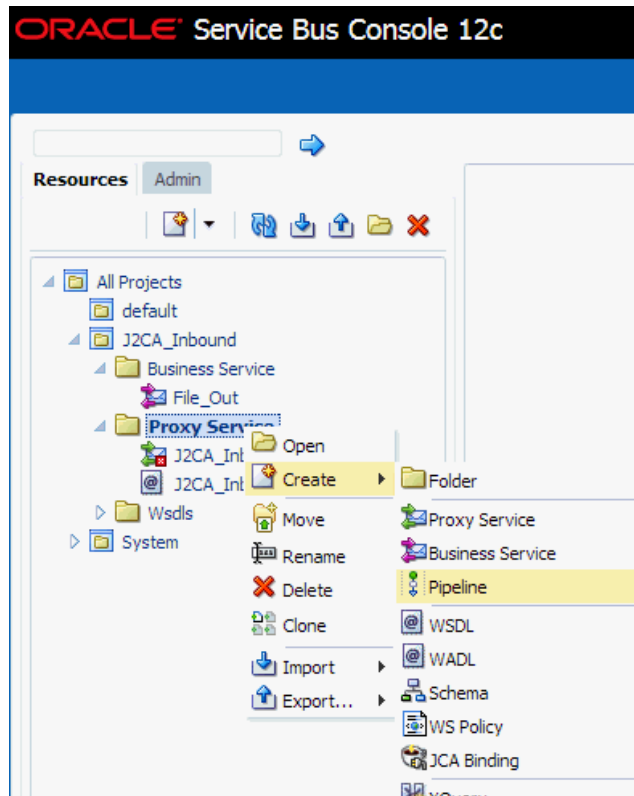


9.2.6 Configuring a Pipeline With Proxy Service

Perform the following steps to configure a Pipeline:

1. Right-click the Proxy Service folder, select **Create** and click **Pipeline**, as shown in Figure 9–22.

Figure 9–22 Pipeline Option

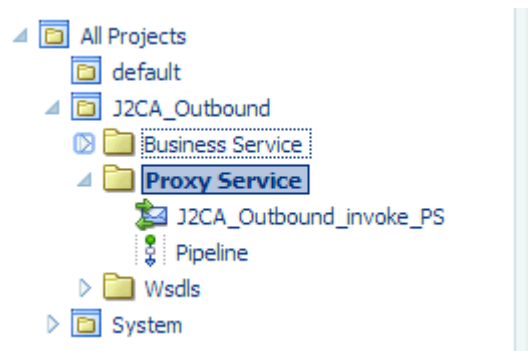


The Create Pipeline window is displayed.

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 9–23.

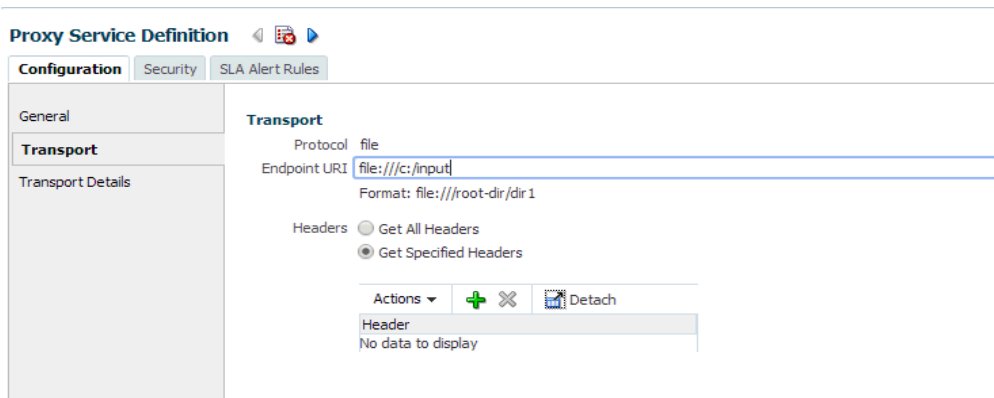
Figure 9–23 Create Pipeline Window

The created Pipeline and the Proxy Service is listed under Proxy Service, as shown in [Figure 9–24](#).

Figure 9–24 Pipeline Node

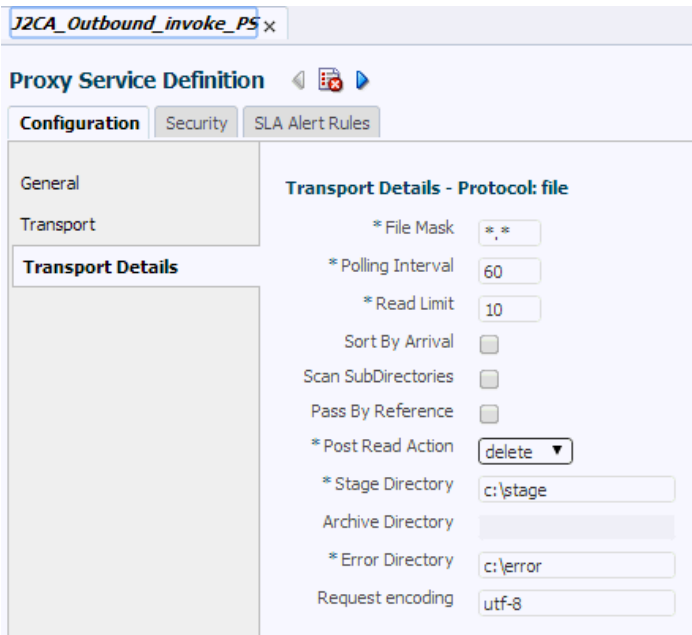
3. Double-click the created proxy service and click **Transport** in the left pane. Provide the input location in the Endpoint URI field, as shown in [Figure 9–25](#).

Figure 9–25 Transport



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 9–26](#).

Figure 9–26 Transport Details



5. Click the **Save All** icon in the right corner, as shown in [Figure 9–27](#).

Figure 9–27 Save All Icon



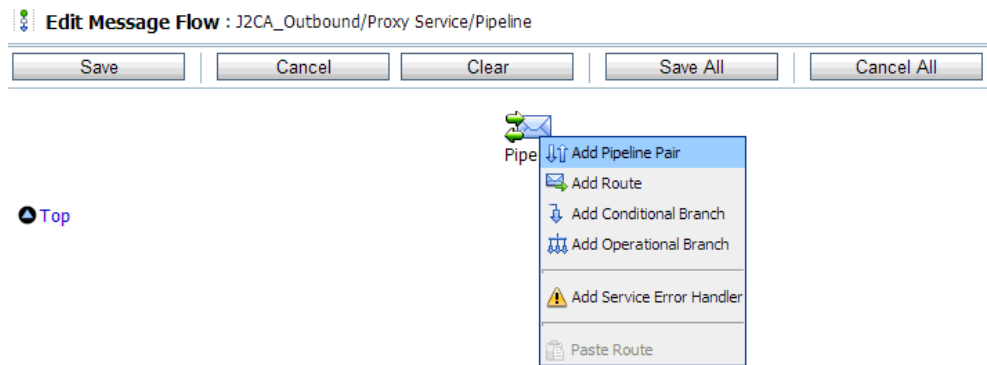
6. Double-click the **Pipeline** node and click the **Open Message Flow** icon on the right pane to open the message flow, as shown in [Figure 9–28](#).

Figure 9–28 Open Message Flow Icon



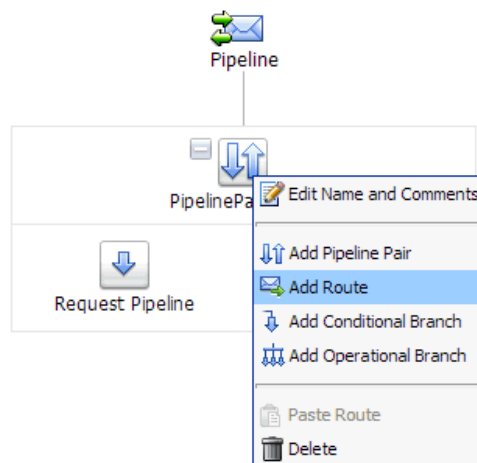
7. Click the Proxy Service icon and select **Add Pipeline Pair** from the menu, as shown in [Figure 9–29](#).

Figure 9–29 Add Pipeline Pair Option



8. Click the **PipelinePairNode1** icon and select **Add Route** from the menu, as shown in [Figure 9–30](#).

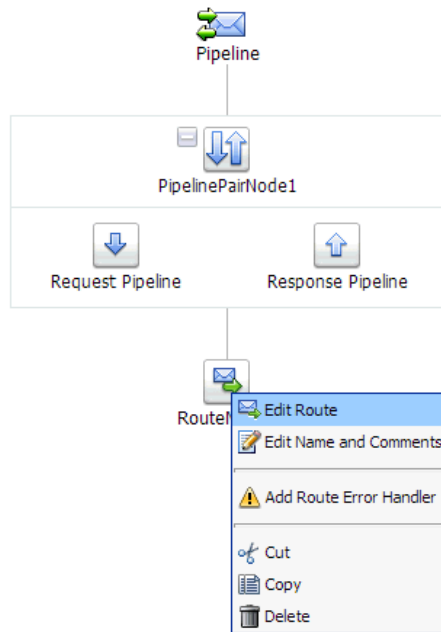
Figure 9–30 Add Route Option



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 9–31](#).

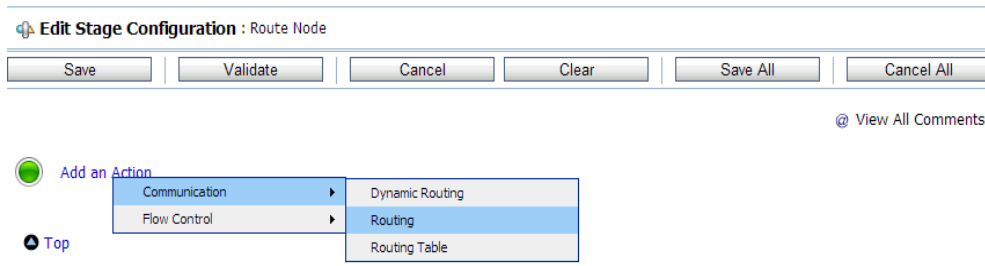
Figure 9–31 Edit Route Option



The Edit Stage Configuration workspace area is displayed.

10. Click **Add an Action**, select **Communication** and click **Routing**, as shown in Figure 9–32.

Figure 9–32 Edit Stage Configuration Workspace Area



11. Click **<Service>**, as shown in Figure 9–33.

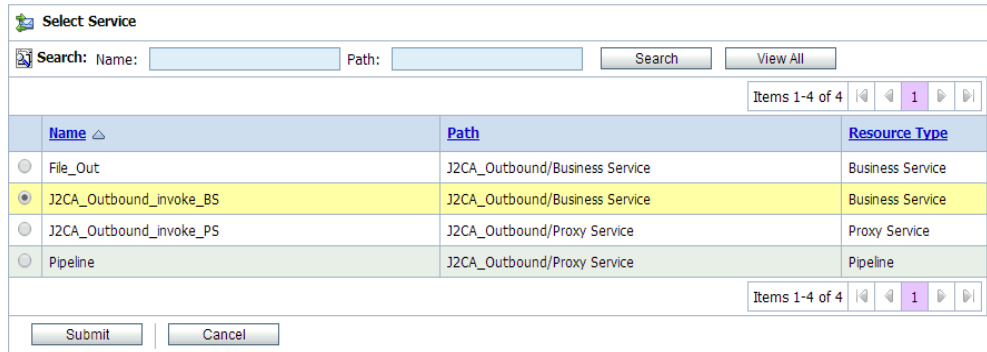
Figure 9–33 Actions



The Select Service dialog is displayed.

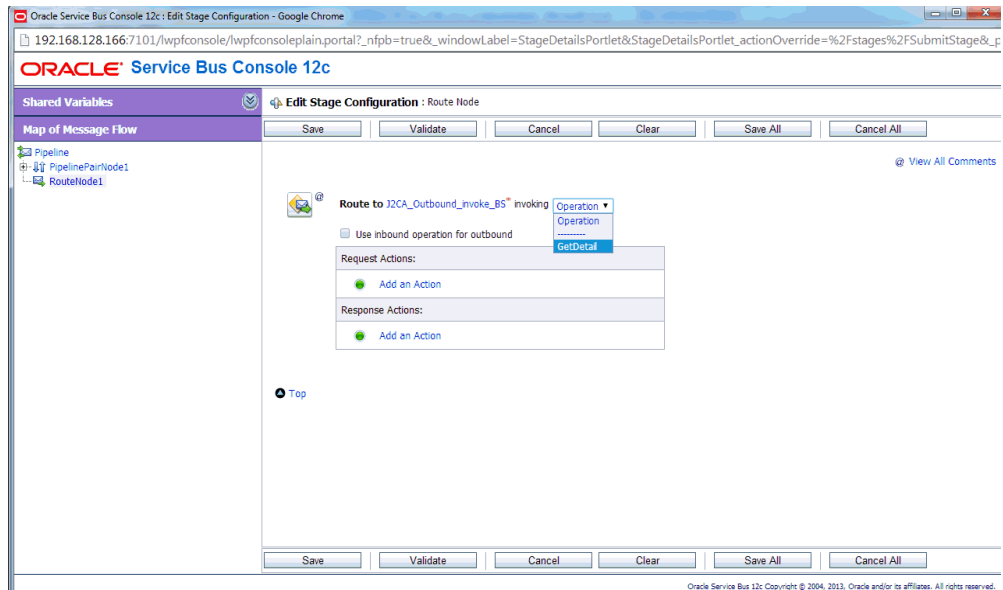
12. Select the WSDL type Business Service configured for SAP R/3 and click on **Submit**, as shown in Figure 9–34.

Figure 9–34 Select Service Dialog



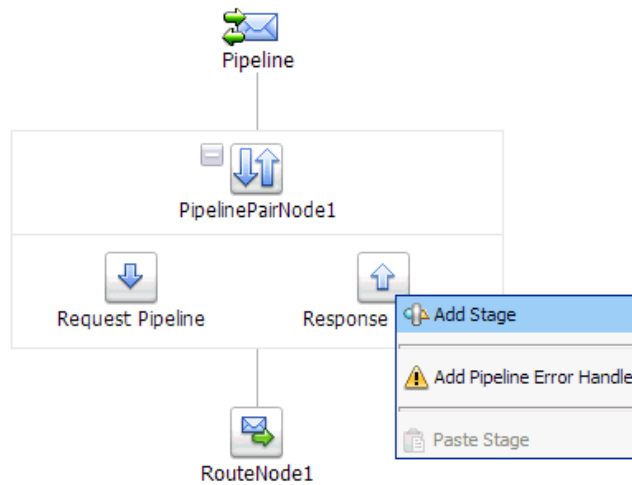
13. Select the name of the SAP R/3 business object (for example, GetDetail) as the operational attribute from the list, and click **Save**, as shown in Figure 9–35.

Figure 9–35 Edit Stage Configuration



14. Click the Response Pipeline icon and select **Add Stage** from the menu, as shown in Figure 9–36.

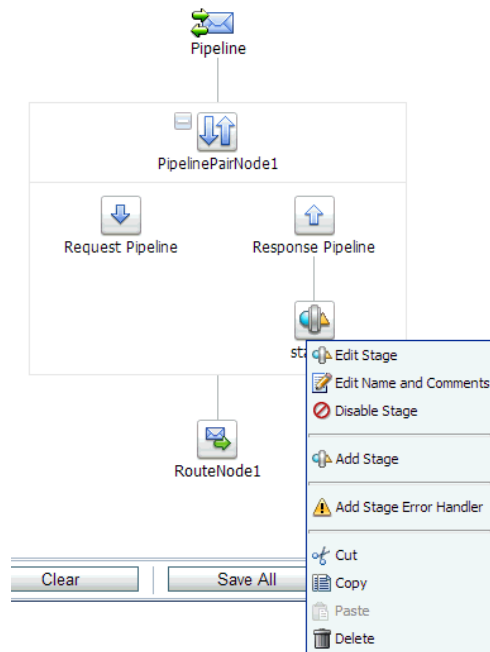
Figure 9–36 Response Pipeline Icon



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 9–37](#).

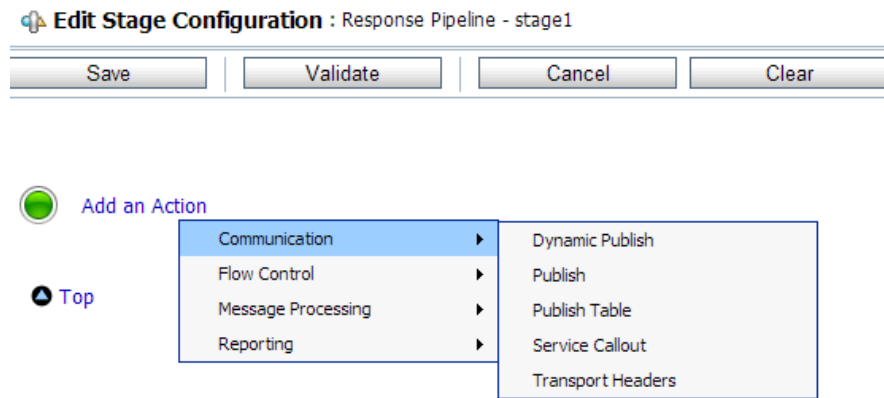
Figure 9–37 Edit Stage Option



The Edit Stage Configuration workspace area is displayed.

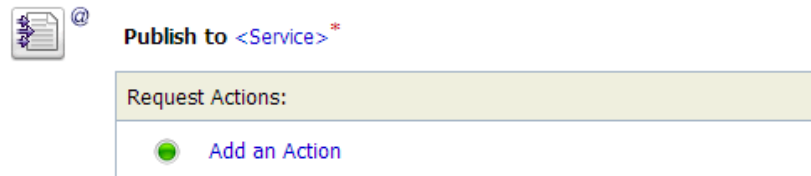
16. Click **Add an Action**, select **Communication**, and then click **Publish**, as shown in [Figure 9–38](#).

Figure 9–38 Edit Stage Configuration Workspace Area



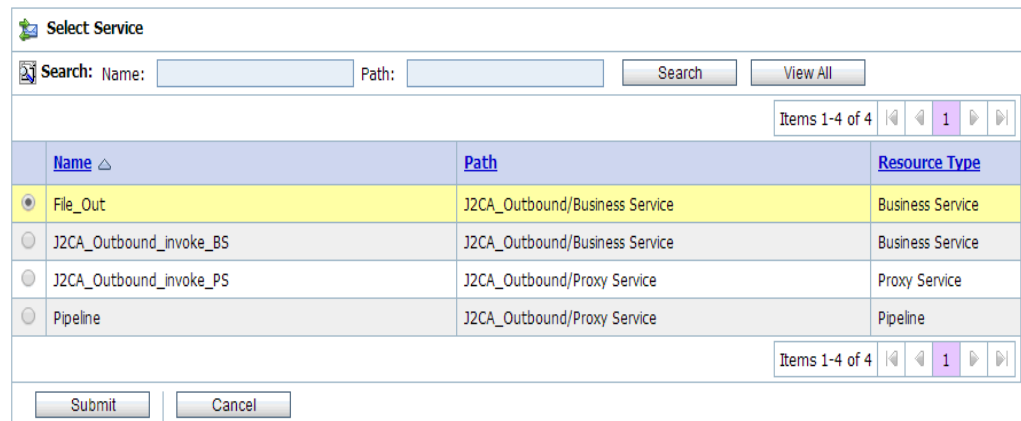
17. Click <Service>, as shown in Figure 9–39.

Figure 9–39 <Service> Action



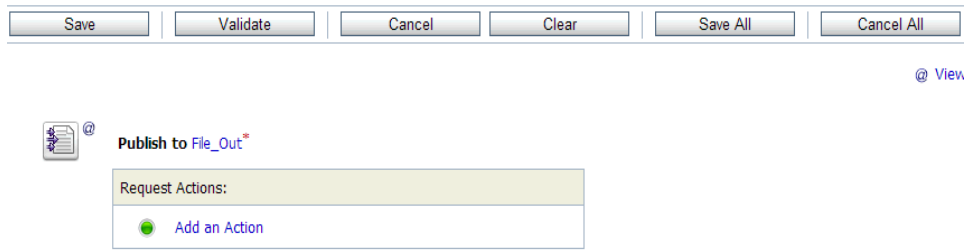
18. In the Select Service dialog, select a File type Business Service and click **Submit**, as shown in Figure 9–40.

Figure 9–40 Select Service Dialog



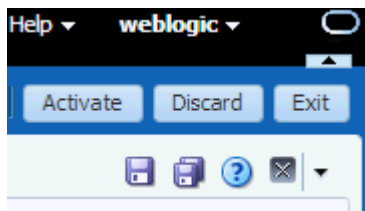
19. Click **Save All**, as shown in Figure 9–41.

Figure 9–41 Save All Button



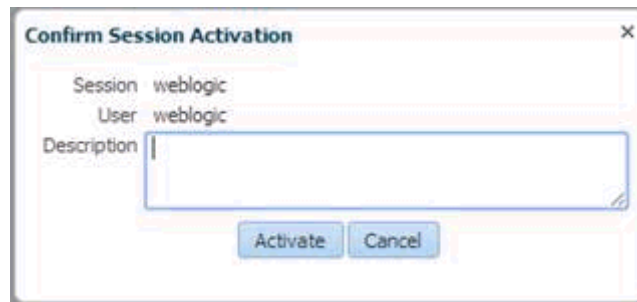
20. Click **Activate** in the right pane of the Oracle Service Bus session, as shown in [Figure 9–42](#).

Figure 9–42 Activate Button



21. Click **Activate** to save the changes, as shown in [Figure 9–43](#).

Figure 9–43 Confirm Session Activation



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).

9.3 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\SAP_Samples.zip\SAP_Samples\OSB\J2CA\SAP_Sample_J2CA_
OSB_Inbound_Project
```

This section includes the following topics:

- [Section 9.3.1, "Starting Oracle Service Bus and Creating Project Folders"](#)
- [Section 9.3.2, "Setting the Class Path for Application Explorer to Integrate With Oracle Service Bus"](#)
- [Section 9.3.3, "Publishing a WSDL From Application Explorer to Oracle Service Bus"](#)
- [Section 9.3.4, "Configuring a WSDL-based Proxy Service"](#)
- [Section 9.3.5, "Configuring a File Type Business Service"](#)
- [Section 9.3.6, "Configuring a Pipeline"](#)

9.3.1 Starting Oracle Service Bus and Creating Project Folders

For more information on starting Oracle Service Bus and creating project folders, see [Section 9.2.1, "Starting Oracle Service Bus and Creating Project Folders"](#) on page 9-2.

9.3.2 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 9.2.2, "Setting the Class Path for Application Explorer to Integrate With Oracle Service Bus"](#) on page 9-6.

9.3.3 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 SAP R/3 target.

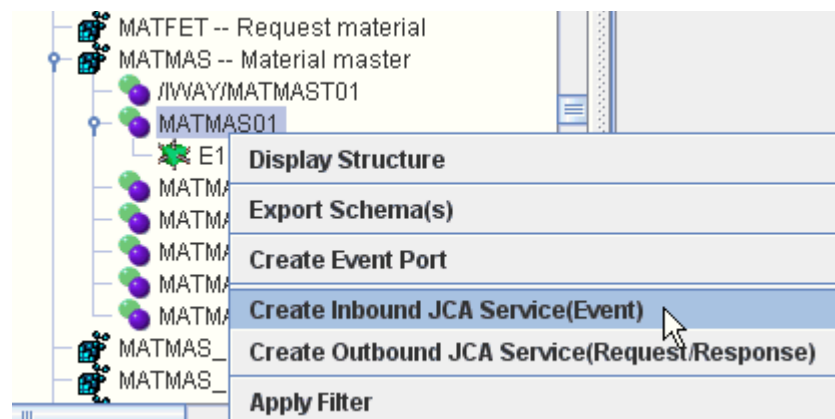
For more information, see [Chapter 4, "Configuring Oracle Application Adapter for SAP R/3"](#).

2. Create a SAP R/3 channel.

For more information, see [Section 4.10, "Configuring an Event Adapter"](#) on page 4-22.

3. Expand the **ALE (IDOCs)** node and then the **MATMAS -- Material master** node in the target.

Figure 9-44 Create Inbound JCA Service(Event) Option



- Right-click **MATMAS01** and select **Create Inbound JCA Service(Event)** from the menu, as shown in [Figure 9–44](#).

The Export WSDL dialog is displayed, as shown in [Figure 9–45](#).

Figure 9–45 Export WSDL Dialog

- In the Name field, a default file name for the WSDL file is provided. You can accept the default or provide your own.
- From the Channel list, select the channel you created for this inbound service.
- Three check boxes for Root, Namespace, and Schema validation are also available. Selection of multiple validation options is allowed.
- Select the **Export to OSB** option.
- 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 “/”.
- In the Host field, enter the name of the machine where Oracle Service Bus is installed.
- In the Port field, enter the port that is being used by Oracle Service Bus.
- In the User field, enter your username to access Oracle Service Bus.
- In the Password field, enter your password to access Oracle Service Bus.
- Click **OK**.

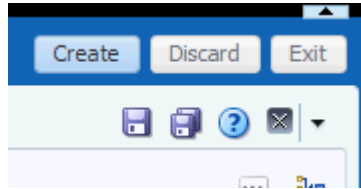
The inbound WSDL is published to the location specified in the Export WSDL dialog and is now available for use with a Proxy Service in Oracle Service Bus.

9.3.4 Configuring a WSDL-based Proxy Service

Perform the following steps to configure a WSDL-based Proxy Service:

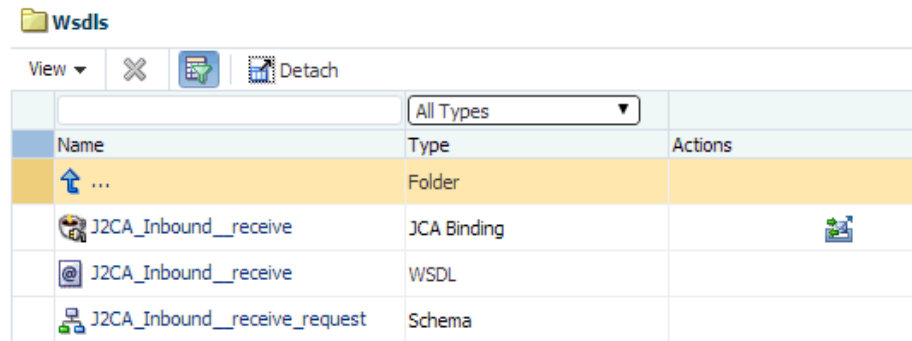
1. Open the Oracle Service Bus Console and click **Create** in the right pane of the Oracle Service Bus session, as shown in [Figure 9-46](#).

Figure 9-46 Create Button



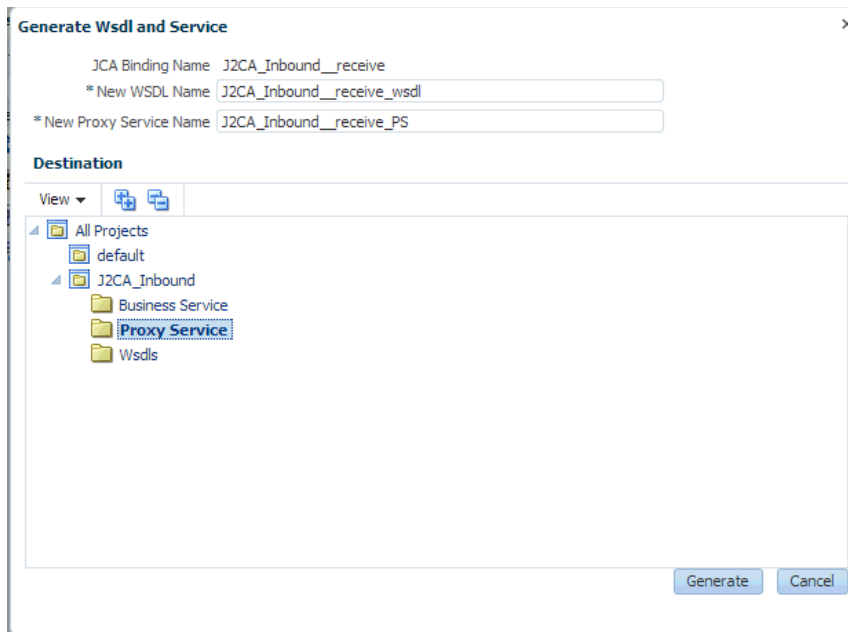
2. Double-click the created WSDL folder in the left pane (for example, Wsdls), and ensure that the exported WSDL is listed in the right pane, as shown in [Figure 9-47](#).

Figure 9-47 Exported WSDL



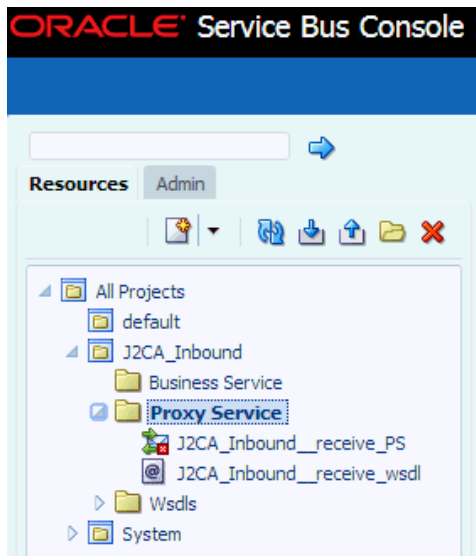
3. Click the icon that corresponds to the JCA Binding in the Actions column. The Generate WSDL and Service page is displayed, as shown in [Figure 9-48](#).

Figure 9–48 Generate WSDL and Service Page



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 9–49](#).

Figure 9–49 Generated WSDL

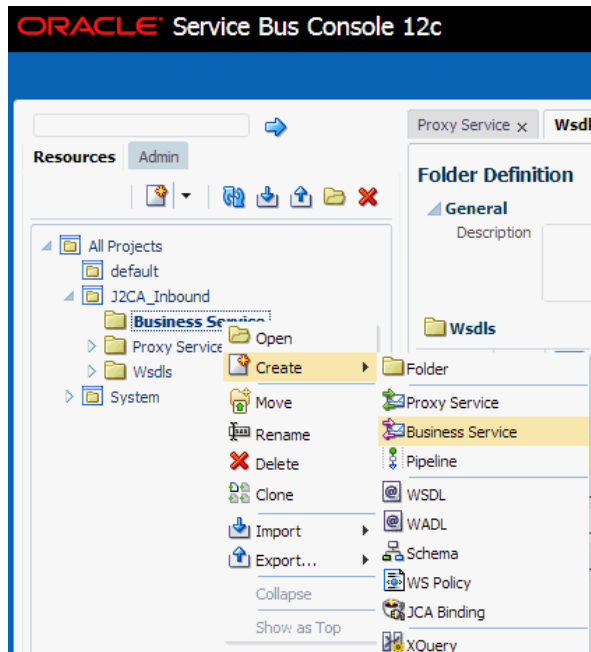


9.3.5 Configuring a File Type Business Service

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 9–50](#).

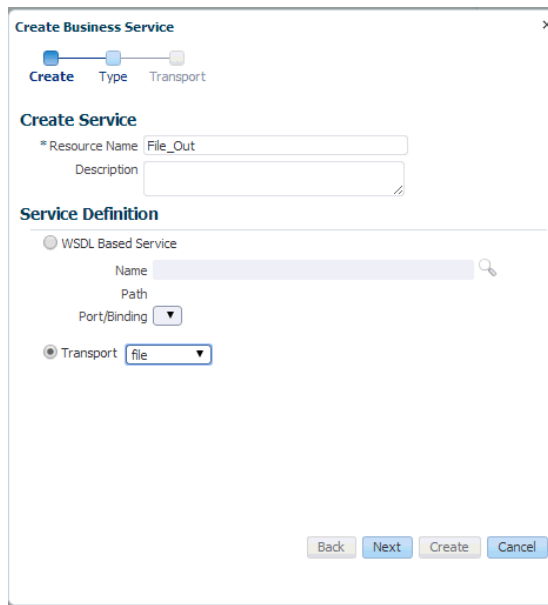
Figure 9–50 Business Service Folder



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 9–51](#).

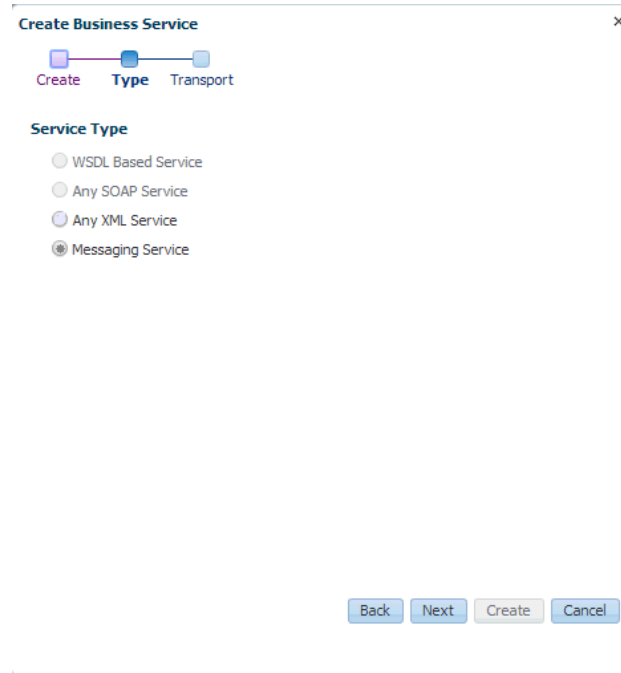
Figure 9–51 Create Business Service Window



3. Click **Next**.

4. In the Service Type area, select **Messaging Service** as the service type, as shown in [Figure 9–52](#).

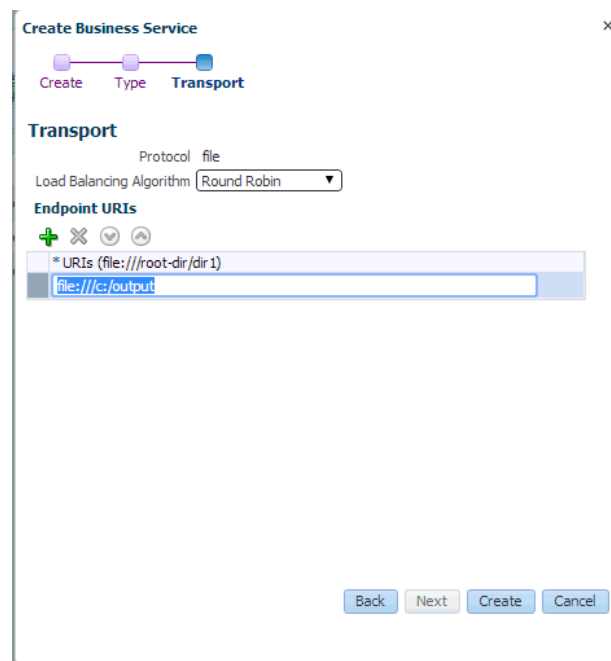
Figure 9–52 Service Type Area



5. Click **Next**.

The Transport page is displayed, as shown in [Figure 9–53](#).

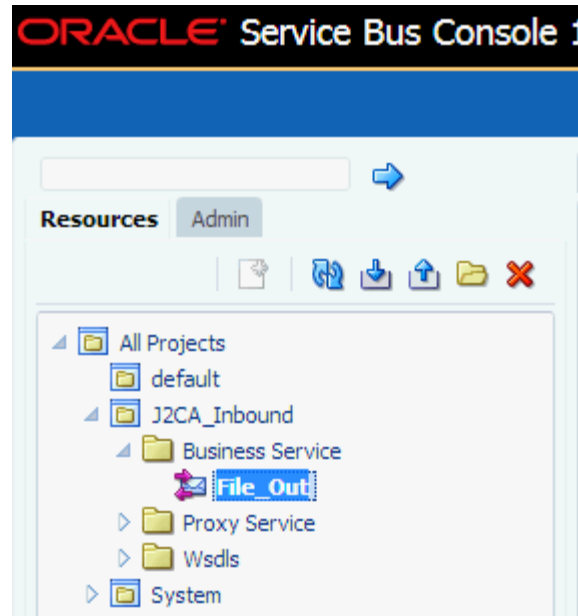
Figure 9–53 Transport Page



6. Enter the path to a destination folder on your file system in the Endpoint URI field and click **Create**.

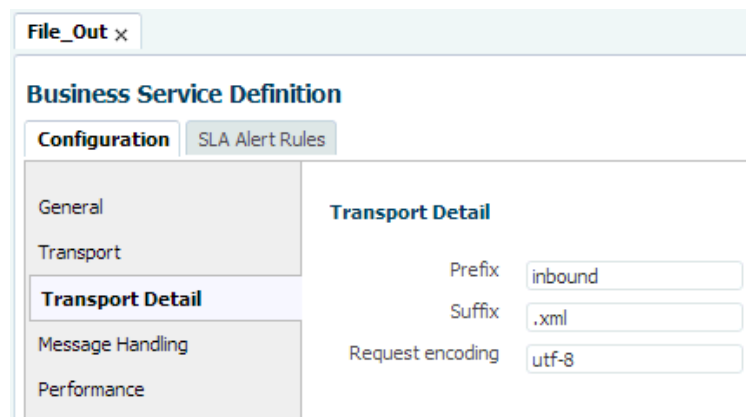
The Business Service File_Out is created and listed under Business Service, as shown in [Figure 9-54](#).

Figure 9-54 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 9-55](#).

Figure 9-55 Transport Detail Page



8. Click the Save or Save All icon in the right corner, as shown in [Figure 9-56](#).

Figure 9–56 Save Icons

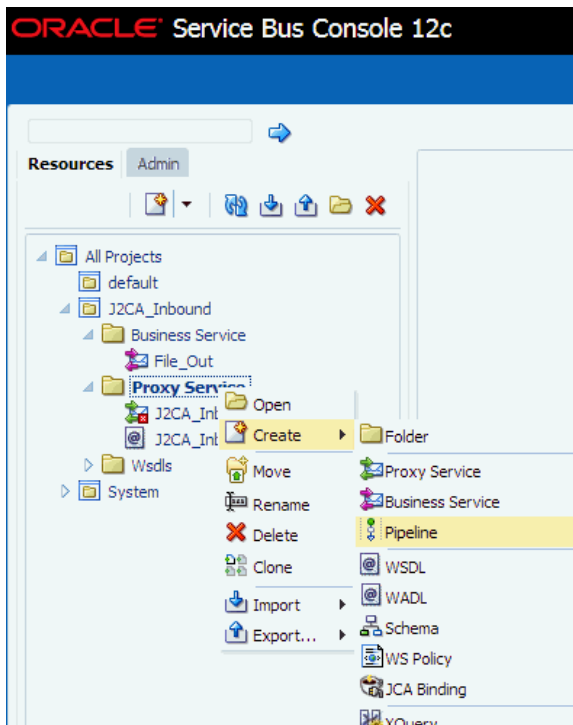


9.3.6 Configuring a Pipeline

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 9–57](#).

Figure 9–57 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 9–58](#).

Figure 9–58 Create Pipeline Window

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 9–59](#).

Figure 9–59 Search and Select: WSDL Resource Window

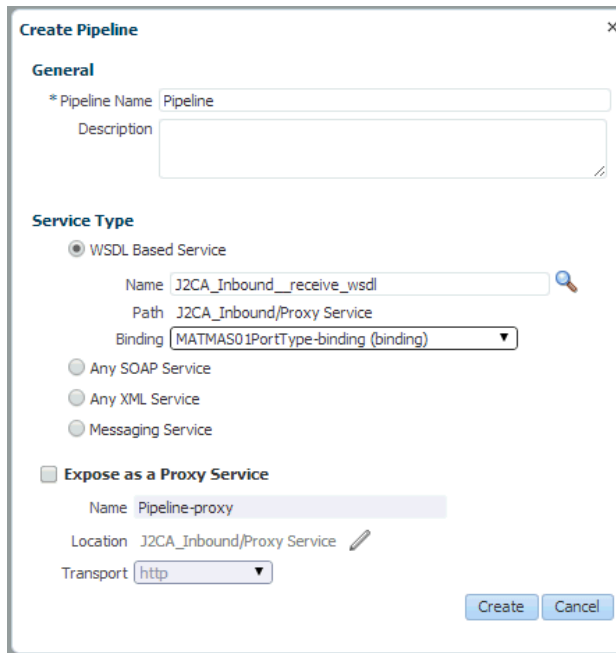
Name	Path	Namespace
J2CA_Inbound__receive_wsdl	J2CA_Inbound/Proxy Service	http://xmlns.oracle...
J2CA_Inbound__receive	J2CA_Inbound/Wsdls	http://xmlns.oracle...

OK Cancel

The Create Pipeline window opens.

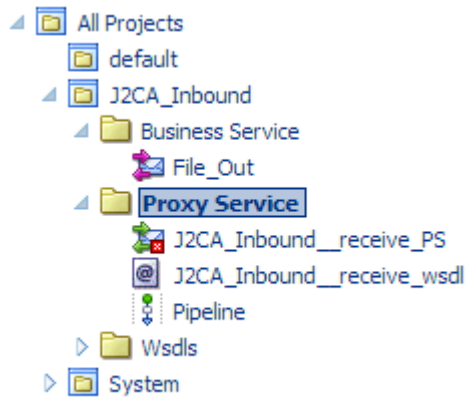
4. Clear the check box for **Expose as a Proxy Service**, and click **Create**, as shown in [Figure 9–60](#).

Figure 9–60 Create Pipeline Window



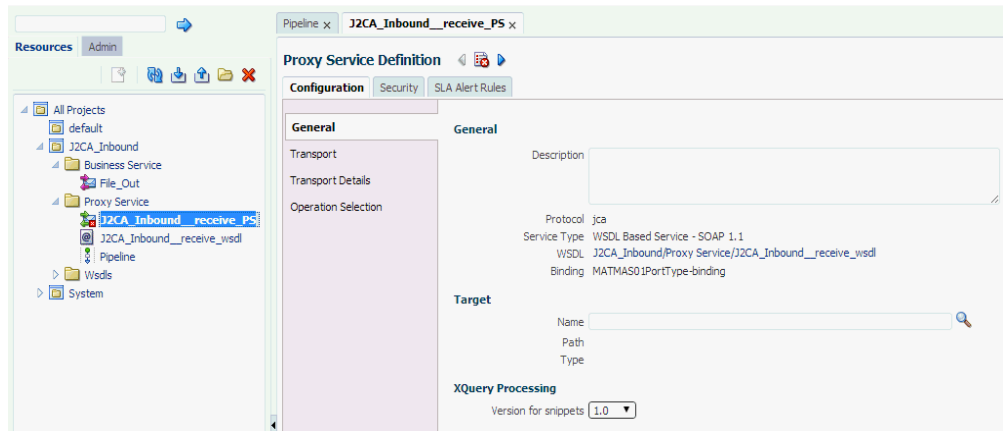
The pipeline is created and listed under Proxy Service, as shown in [Figure 9–61](#).

Figure 9–61 Proxy Service Pipeline



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 9–62](#).

Figure 9–62 Proxy Service Definition Window

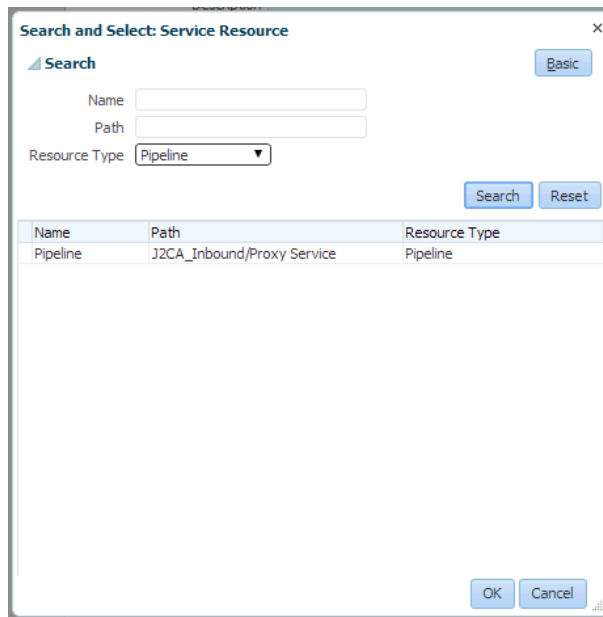


The Search and Select: Service Resource window appears.

- From the Resource Type drop-down list, select **Pipeline** and then click the **Search** button.

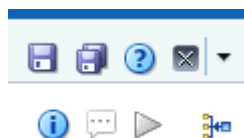
The Pipeline is listed, as shown in [Figure 9–63](#).

Figure 9–63 Search and Select: Service Resource Window



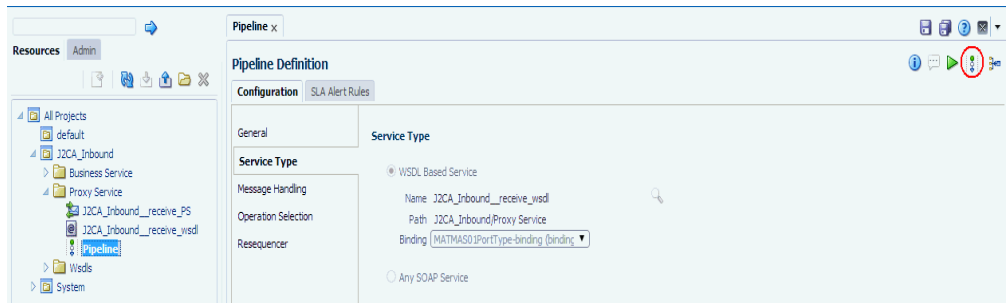
- Select the Pipeline and click **OK**.
- Click the Save or Save All icon in the right corner, as shown in [Figure 9–64](#).

Figure 9–64 Save and Save All Icons



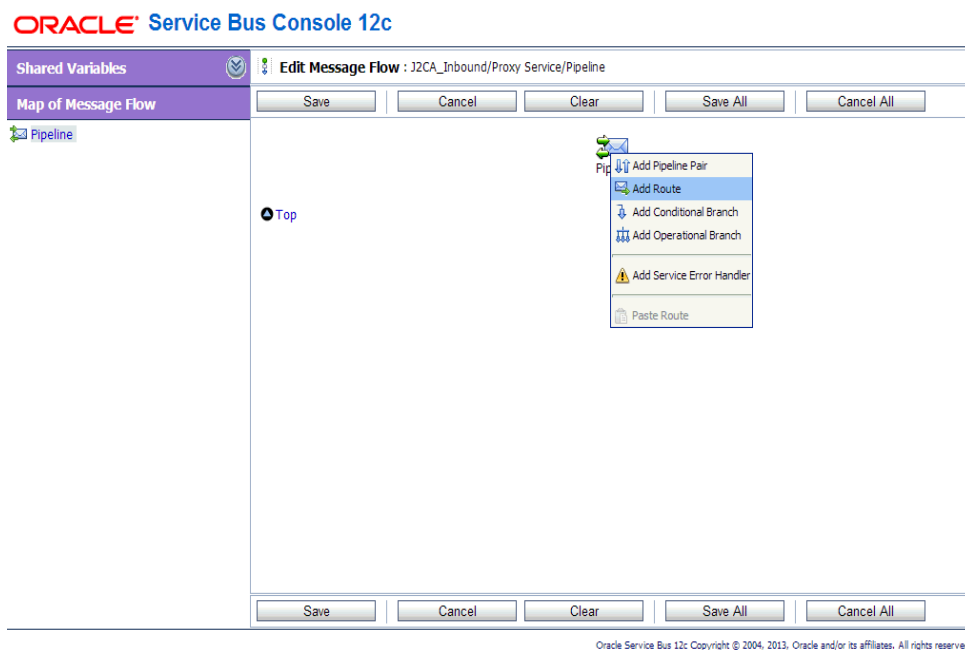
- In the left pane, double-click **Pipeline** under the Proxy Service folder and click the down-pointing icon on the right pane to open the message flow, as shown in Figure 9–65.

Figure 9–65 Message Flow



- Click the displayed Proxy service icon and select **Add Route** from the menu, as shown in Figure 9–66.

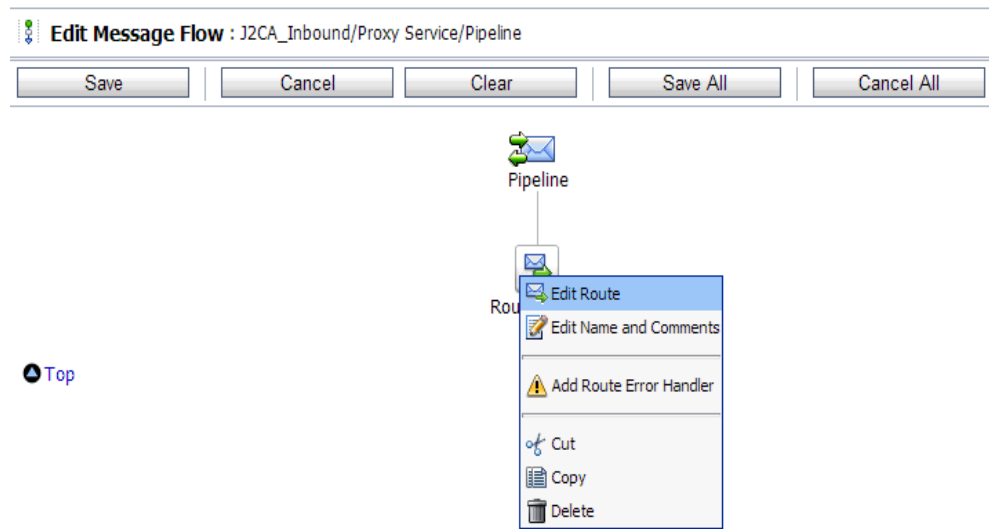
Figure 9–66 Add Route Option



The RouteNode1 icon is added.

- Click the RouteNode1 icon and select **Edit Route** from the menu, as shown in Figure 9–67.

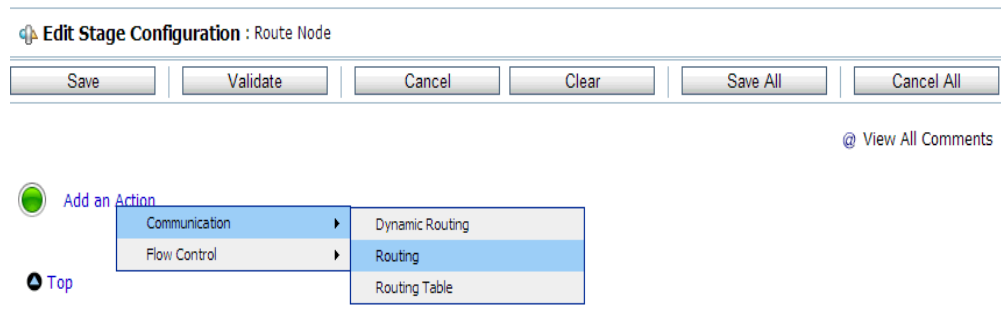
Figure 9–67 Edit Route Option



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 9–68.

Figure 9–68 Edit Stage Configuration Workspace



13. Click **<Service>**, as shown in Figure 9–69.

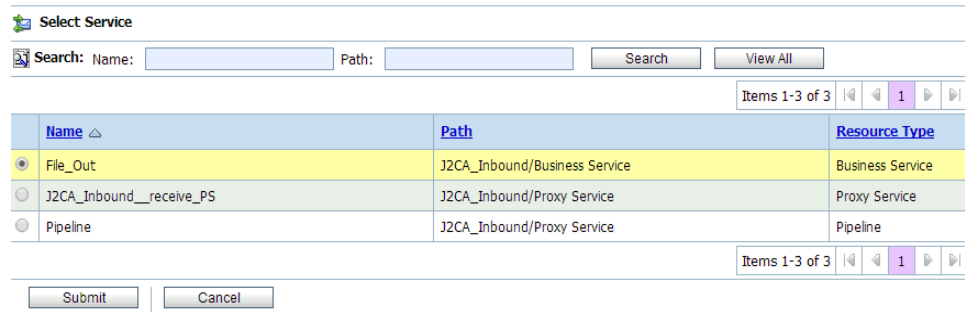
Figure 9–69 Service Route Actions



The Select Service dialog is displayed.

14. Select the **File_Out** Business service and click **Submit** as shown in [Figure 9–70](#).

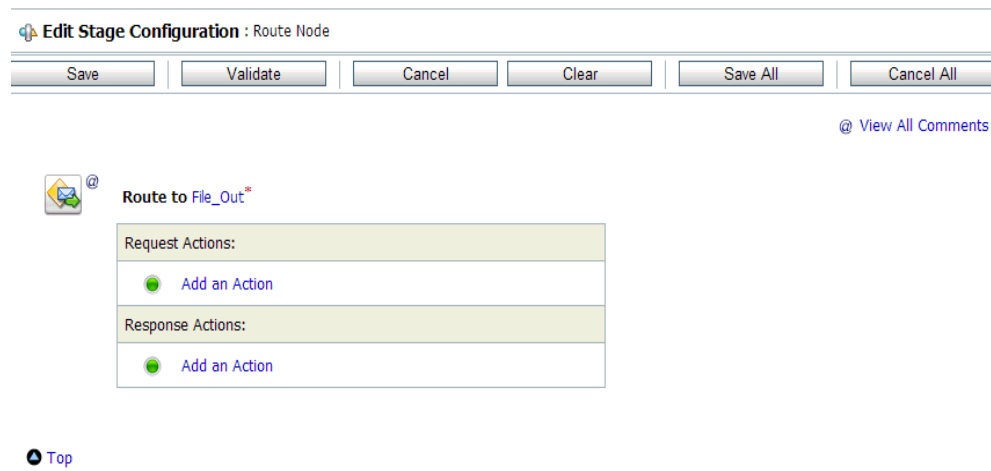
Figure 9–70 Select Service Dialog



You are returned to the Edit Stage Configuration workspace area.

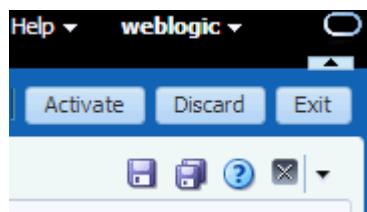
15. Click **Save All**, as shown in [Figure 9–71](#).

Figure 9–71 Edit Stage Configuration Workspace Area



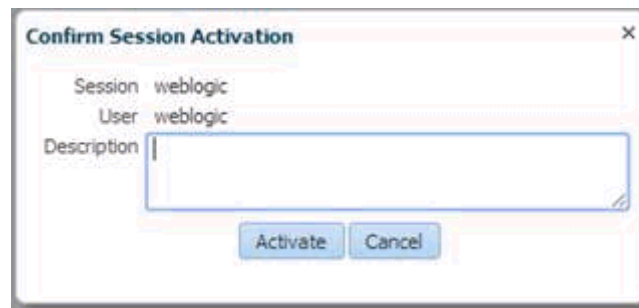
16. Click **Activate** in the right pane of the Oracle Service Bus session, as shown in [Figure 9–72](#).

Figure 9–72 Activate Button



The Confirm Session Activation window appears.

17. Click **Activate** to save the changes, as shown in [Figure 9–73](#).

Figure 9–73 Confirm Session Activation Window

18. Trigger an event from the SAP R/3 system and check if the output is received in the configured output location.

9.4 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\SAP_Samples.zip\SAP_Samples\OSB\BSE\SAP_Sample_BSE_OSB_Outbound_Project
```

This section includes the following topics:

- [Section 9.4.1, "Starting Oracle Service Bus and Creating Project Folders"](#)
- [Section 9.4.2, "Setting the Class Path for Application Explorer to Integrate With Oracle Service Bus"](#)
- [Section 9.4.3, "Publishing a WSDL From Application Explorer to Oracle Service Bus"](#)
- [Section 9.4.4, "Configuring a File Type Business Service"](#)
- [Section 9.4.5, "Configuring a WSDL-based Business Service"](#)
- [Section 9.4.6, "Configuring a Pipeline With Proxy Service"](#)

9.4.1 Starting Oracle Service Bus and Creating Project Folders

For more information on starting Oracle Service Bus and creating project folders, see [Section 9.2.1, "Starting Oracle Service Bus and Creating Project Folders"](#) on page 9-2.

9.4.2 Setting the Class Path for Application Explorer to Integrate With Oracle Service Bus

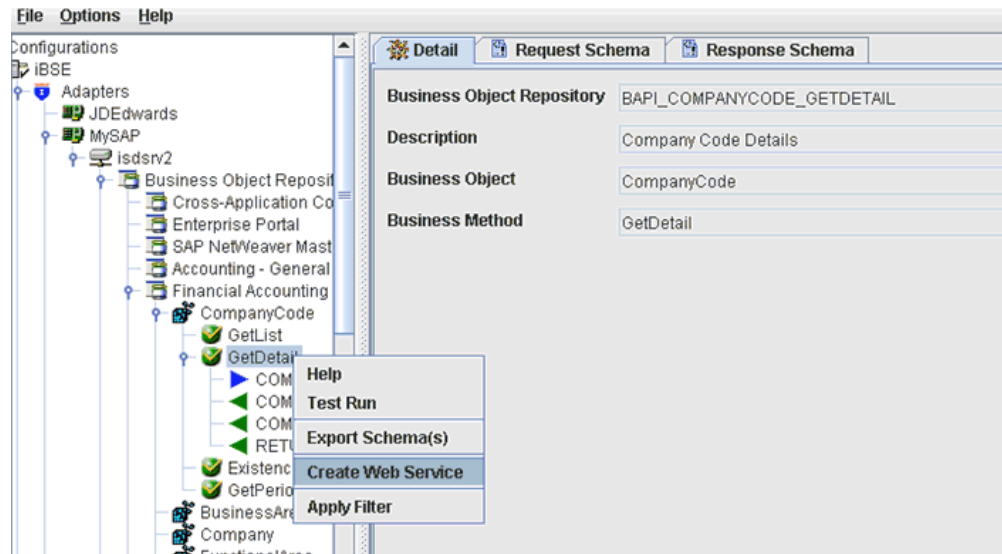
For more information on setting the class path for Application Explorer to integrate with Oracle Service Bus, see [Section 9.2.2, "Setting the Class Path for Application Explorer to Integrate With Oracle Service Bus"](#) on page 9-6.

9.4.3 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 SAP R/3 target.

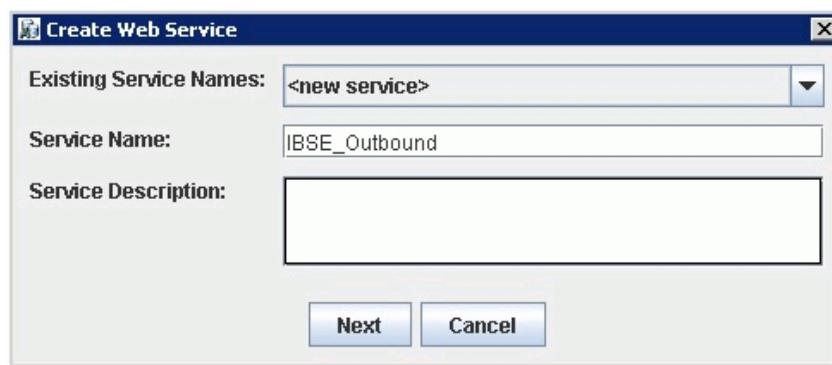
Figure 9–74 Create Web Service Option



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 9–74](#).

The Create Web Service dialog is displayed, as shown in [Figure 9–75](#).

Figure 9–75 Create Web Service Dialog



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 9–76](#).

Figure 9–76 Export WSDL Dialog

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.

9.4.4 Configuring a File Type Business Service

For more information on configuring a file type business service, see [Section 9.2.5, "Configuring a File Type Business Service"](#) on page 9-10.

9.4.5 Configuring a WSDL-based Business Service

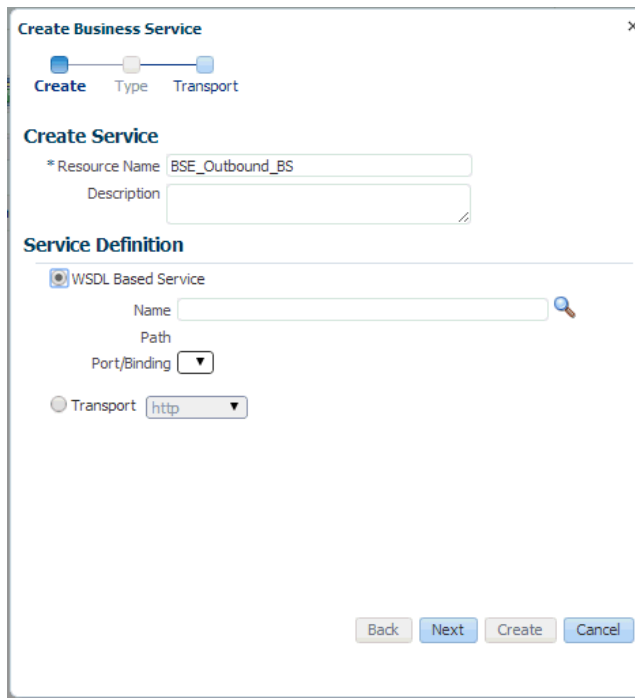
This section describes how to configure a WSDL type Business Service using the Oracle Service Bus Console.

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 9–77](#).

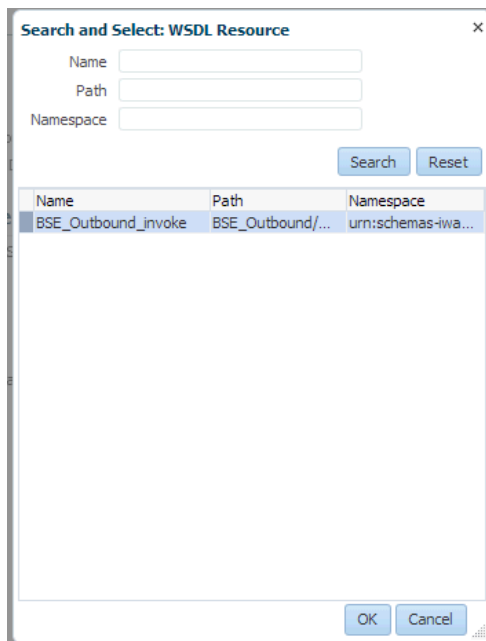
Figure 9–77 Create Business Service Window



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 9–78](#).

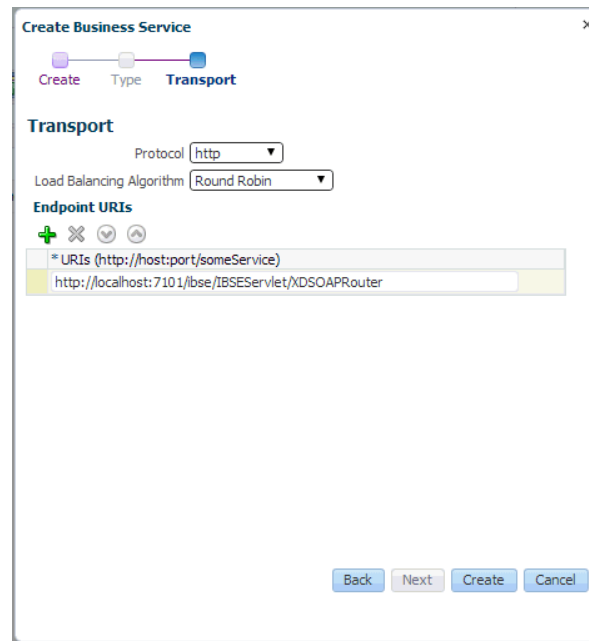
Figure 9–78 Search and Select: WSDL Resource Window



3. Click the **Search** button, select the BSE Outbound WSDL, and click **OK**.
You are returned to the Create Business Service window.

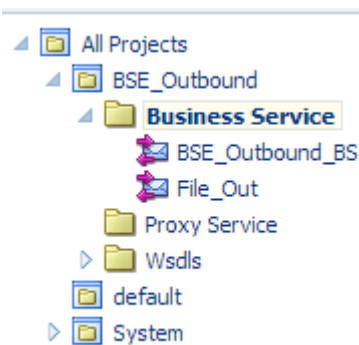
4. Click **Next**.
5. Accept the default values and click the **Create** button, as shown in [Figure 9–79](#).

Figure 9–79 Create Business Service Window



The created WSDL-based Business Service is listed under the Business Service folder, as shown in [Figure 9–80](#).

Figure 9–80 WSDL-based Business Service

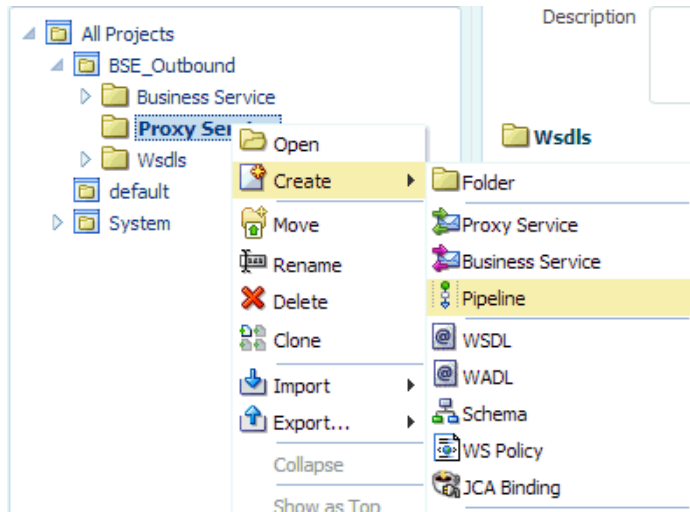


9.4.6 Configuring a Pipeline With Proxy Service

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 9–81](#).

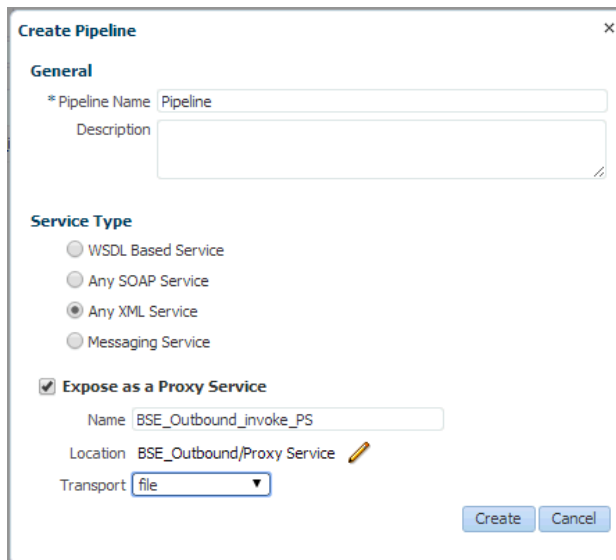
Figure 9–81 Pipeline Option



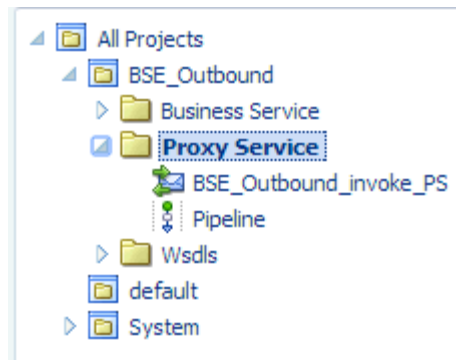
The Create Pipeline window is displayed.

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 9–82](#).

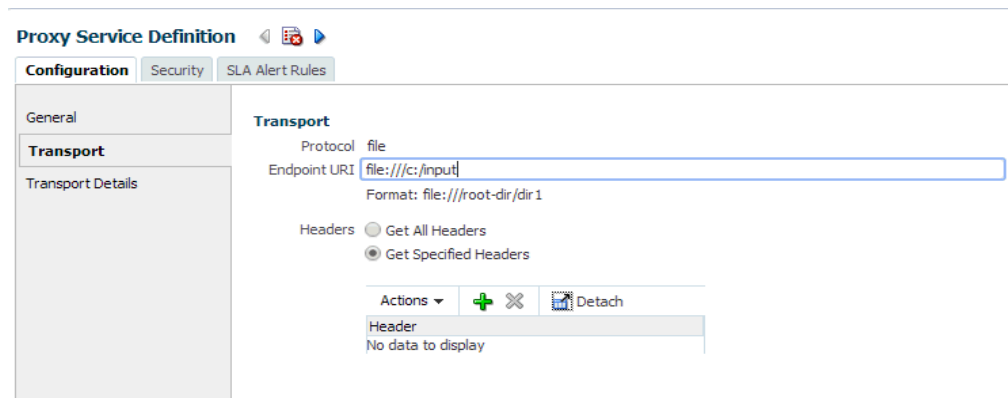
Figure 9–82 Create Pipeline Window



The created Pipeline and the Proxy Service is listed under Proxy Service, as shown in [Figure 9–83](#).

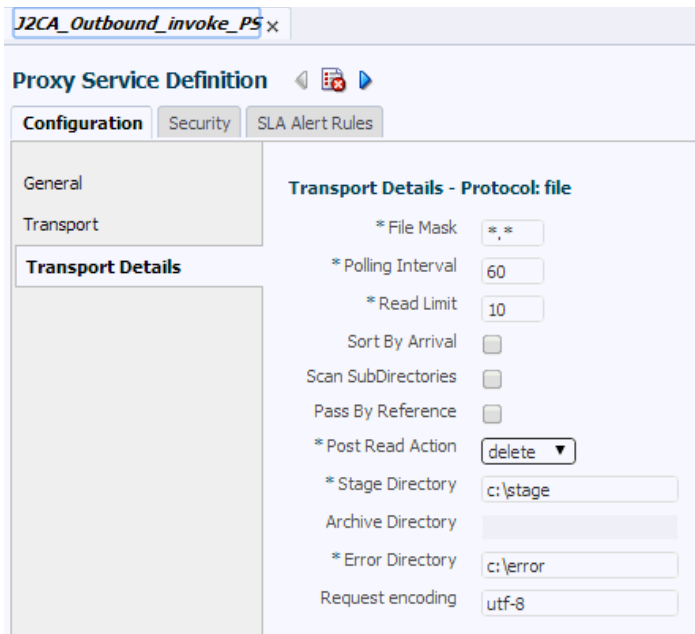
Figure 9–83 Pipeline Node

- 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 9–84](#).

Figure 9–84 Transport

- Click **Transport Details** in the left pane and provide the location for the Stage Directory and the Error Directory fields, as shown in [Figure 9–85](#).

Figure 9–85 Transport Details



5. Click the **Save All** icon in the right corner, as shown in [Figure 9–86](#).

Figure 9–86 Save All Icon



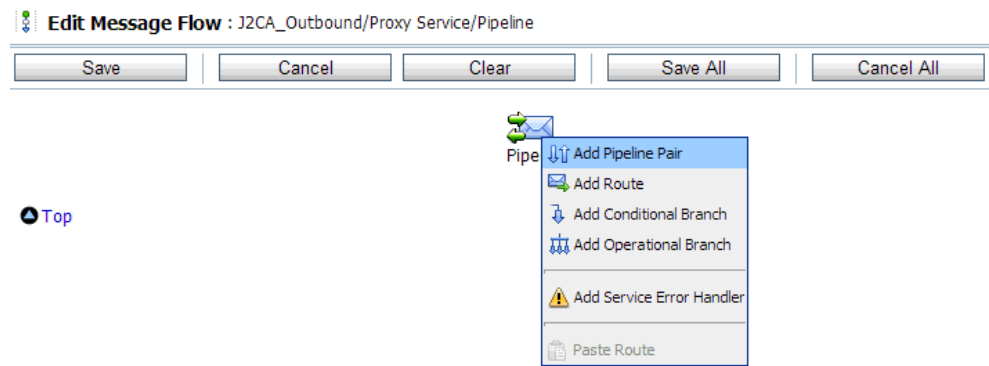
6. Double-click the **Pipeline** node and click the **Open Message Flow** icon on the right pane to open the message flow, as shown in [Figure 9–87](#).

Figure 9–87 Open Message Flow Icon



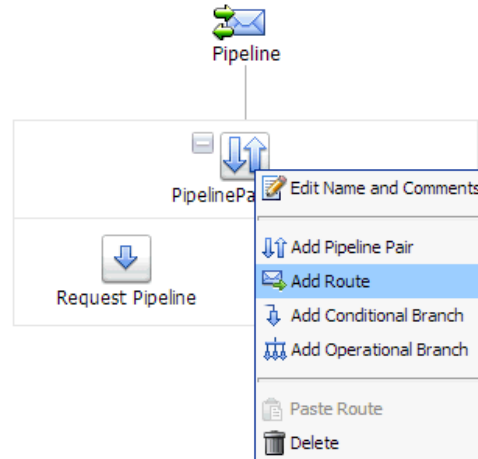
7. Click the Proxy Service icon and select **Add Pipeline Pair** from the menu, as shown in [Figure 9–88](#).

Figure 9–88 Add Pipeline Pair Option



8. Click the **PipelinePairNode1** icon and select **Add Route** from the menu, as shown in [Figure 9–89](#).

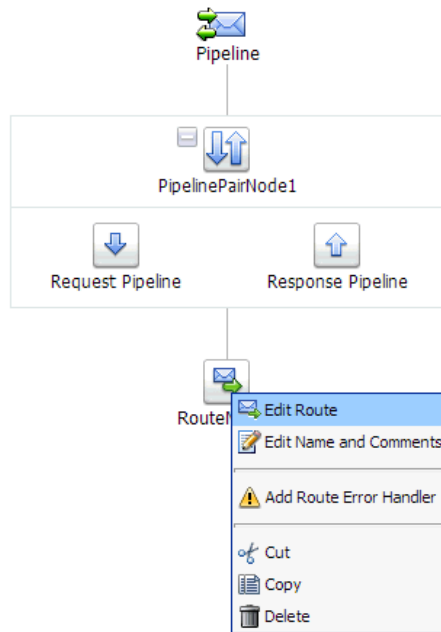
Figure 9–89 Add Route Option



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 9–90](#).

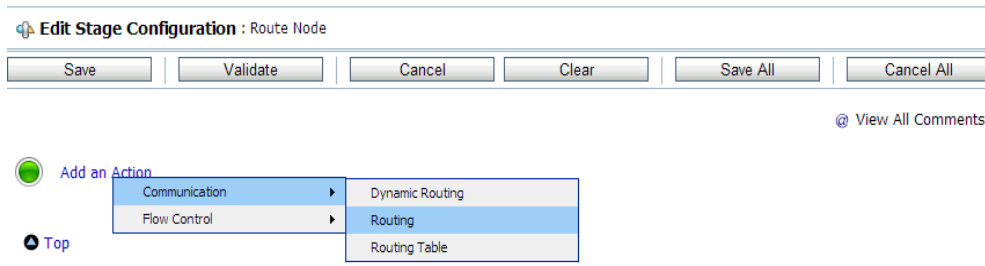
Figure 9–90 Edit Route Option



The Edit Stage Configuration workspace area is displayed.

10. Click **Add an Action**, select **Communication** and click **Routing**, as shown in Figure 9–91.

Figure 9–91 Edit Stage Configuration Workspace Area



11. Click **<Service>**, as shown in Figure 9–92.

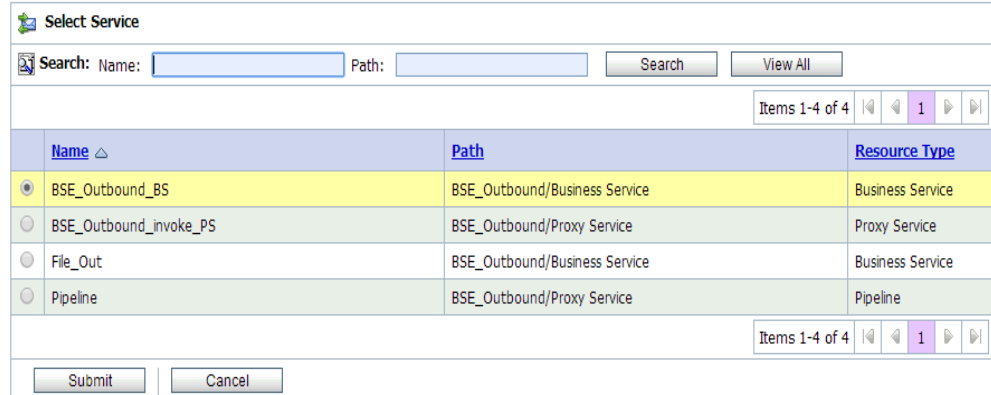
Figure 9–92 Actions



The Select Service dialog is displayed.

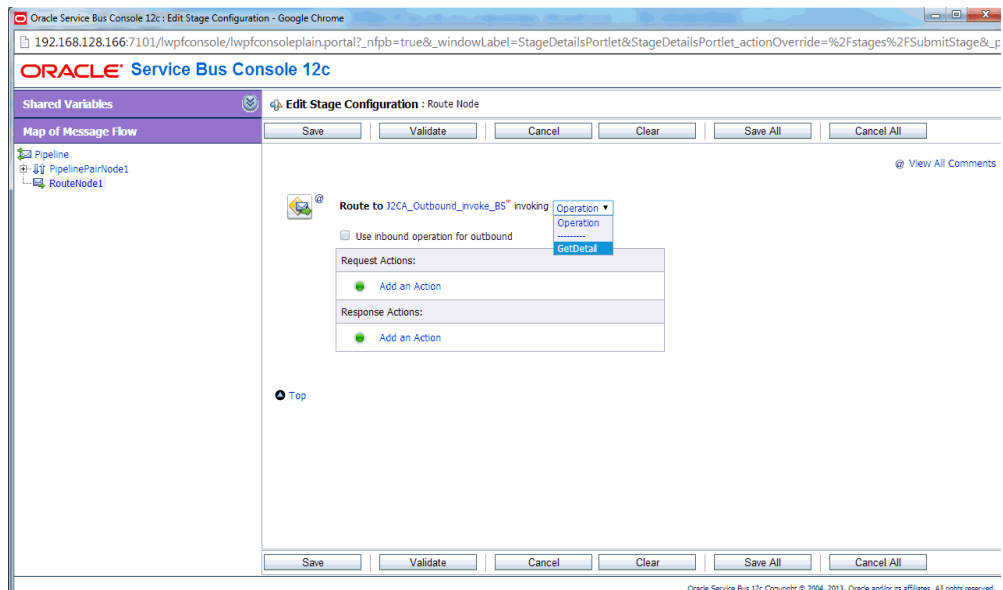
12. Select the WSDL type Business Service configured for SAP R/3 and click on **Submit**, as shown in Figure 9–93.

Figure 9–93 Select Service Dialog



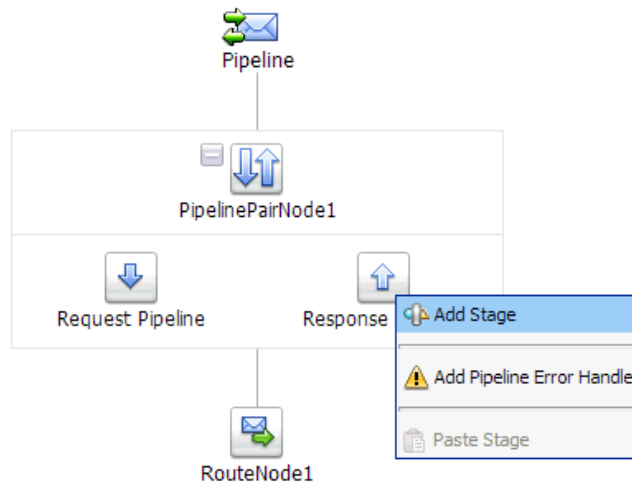
13. Select the name of the SAP R/3 business object (for example, GetDetail) as the operational attribute from the list, and click **Save**, as shown in Figure 9–94.

Figure 9–94 Edit Stage Configuration



14. Click the Response Pipeline icon and select **Add Stage** from the menu, as shown in Figure 9–95.

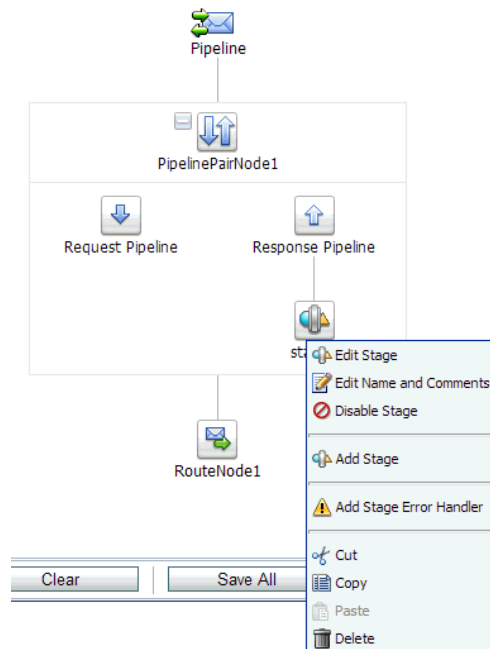
Figure 9–95 Response Pipeline Icon



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 9–96.

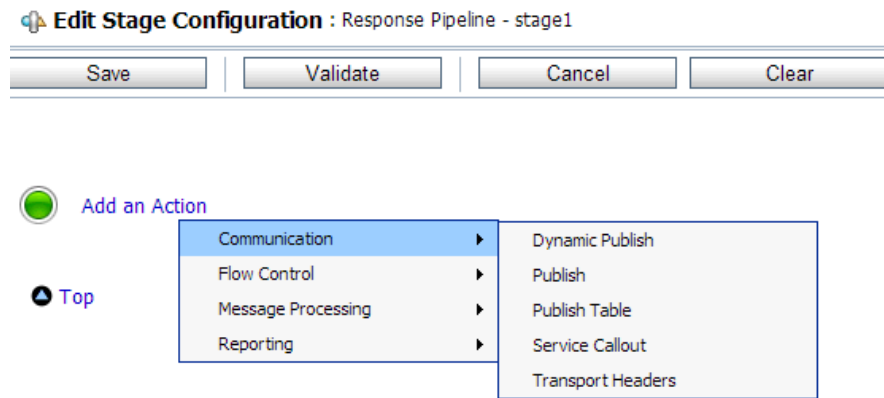
Figure 9–96 Edit Stage Option



The Edit Stage Configuration workspace area is displayed.

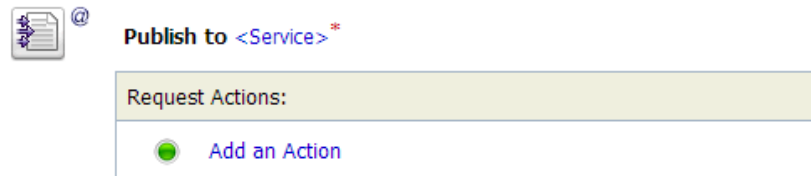
16. Click **Add an Action**, select **Communication**, and then click **Publish**, as shown in Figure 9–97.

Figure 9–97 Edit Stage Configuration Workspace Area



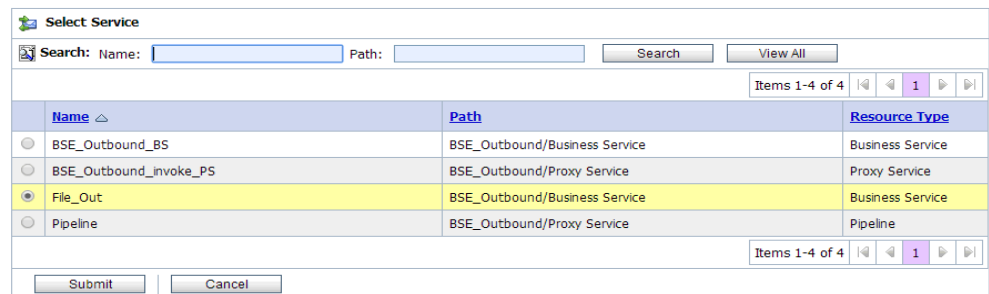
17. Click <Service>, as shown in Figure 9–98.

Figure 9–98 <Service> Action

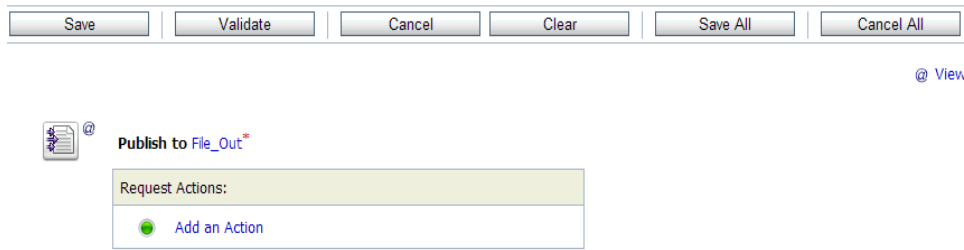


18. In the Select Service dialog, select a File type Business Service and click **Submit**, as shown in Figure 9–99.

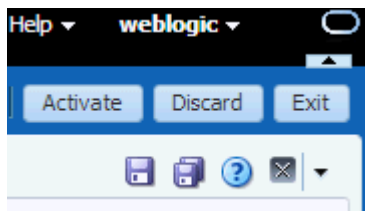
Figure 9–99 Select Service Dialog



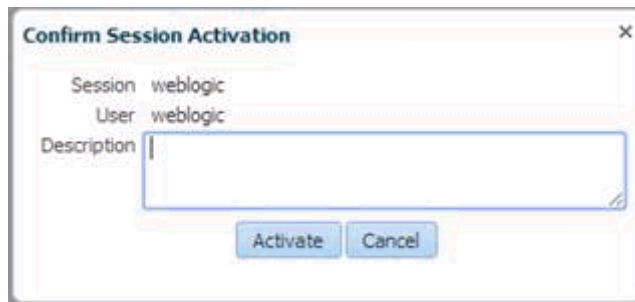
19. Click **Save All**, as shown in Figure 9–100.

Figure 9–100 Save All Button

20. Click **Activate** in the right pane of the Oracle Service Bus session, as shown in [Figure 9–101](#).

Figure 9–101 Activate Button

21. Click **Activate** to save the changes, as shown in [Figure 9–102](#).

Figure 9–102 Confirm Session Activation

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).

9.5 Working With Synchronous SAP R/3 Events Using Oracle Service Bus (J2CA Configuration)

This section describes how to work with synchronous SAP R/3 events using Oracle Service Bus. For more information on synchronous SAP R/3 events, see [Section 6.6.1, "Overview"](#).

Note: Synchronous events are supported only for objects that have reply schema and are not supported for IDocs.

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

```
<ADAPTER_HOME>\etc\sample\SAP_Samples.zip\SAP_Samples\OSB\J2CA\SAP_Sample_J2CA_
OSB_Synchronous_Events
```

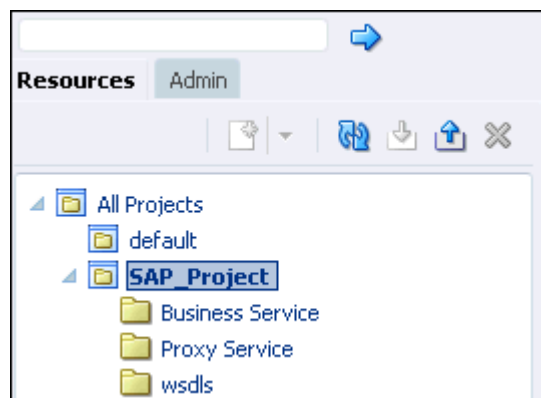
This section includes the following topics:

- [Section 9.5.1, "Starting Oracle Service Bus and Creating Project Folders"](#)
- [Section 9.5.2, "Configuring Adapter Targets and Channels"](#)
- [Section 9.5.3, "Generating a WSDL File for a Message"](#)
- [Section 9.5.4, "Creating a Business Service in the Oracle Service Bus Console"](#)
- [Section 9.5.5, "Creating a Proxy Service in the Oracle Service Bus Console"](#)
- [Section 9.5.6, "Configuring a Pipeline in the Proxy Service"](#)

9.5.1 Starting Oracle Service Bus and Creating Project Folders

Start Oracle Service Bus and create the required project folders, as shown in [Figure 9–103](#).

Figure 9–103 Project Explorer



For more information, see [Section 9.2.1, "Starting Oracle Service Bus and Creating Project Folders"](#).

9.5.2 Configuring Adapter Targets and Channels

Start Application Explorer and create two adapter targets and one channel for SAP R/3 using a J2CA configuration.

For more information, see [Chapter 4, "Configuring Oracle Application Adapter for SAP R/3"](#).

9.5.3 Generating a WSDL File for a Message

Perform the following steps to generate a WSDL file for a message:

1. Ensure that the channel is not started before generating the WSDL file.
2. Connect to the first SAP adapter target (target1).

3. Generate an inbound WSDL file for BAPI_COMPANYCODE_GETLIST (select **Remote Function Modules, Financial Accounting**, and then **0002 - Company Code Business Object**).
4. Connect to the second SAP adapter target (target2).
5. Generate an outbound WSDL file for BAPI_COMPANYCODE_GETLIST (select **Remote Function Modules, Financial Accounting**, and then **0002 - Company Code Business Object**).

For more information, see [Section 9.2.3, "Publishing a WSDL From Application Explorer to Oracle Service Bus"](#) and [Section 9.3.3, "Publishing a WSDL From Application Explorer to Oracle Service Bus"](#).

9.5.4 Creating a Business Service in the Oracle Service Bus Console

This section describes how to create a Business Service in the Oracle Service Bus console.

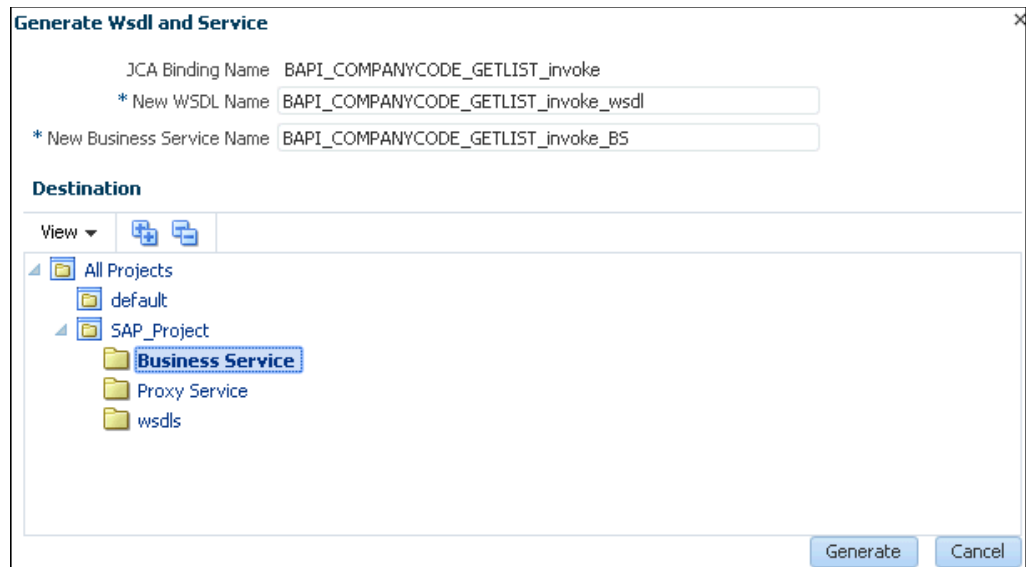
1. In the Oracle Service Bus console, click **Create**, which is located on the top and select the created WSDL folder in the left pane.

WSDL, JCA binding, and XSD files are listed in the right pane, as shown in [Figure 9–104](#).

Figure 9–104 Contents of the WSDL Folder

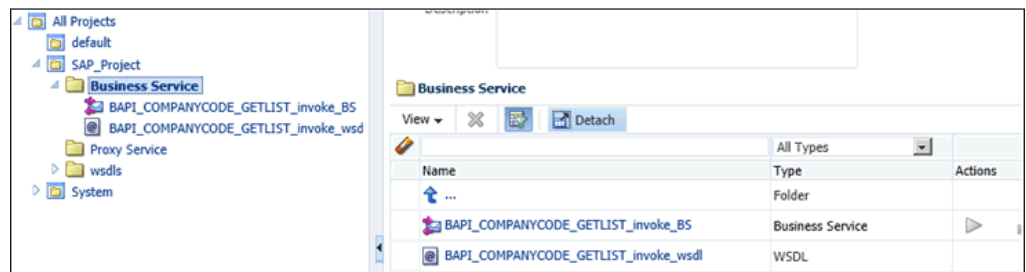
Name	Type	Actions
...	Folder	
BAPI_COMPANYCODE_GETLIST_invoke	JCA Binding	
BAPI_COMPANYCODE_GETLIST_invoke	WSDL	
BAPI_COMPANYCODE_GETLIST_invoke_request	Schema	
BAPI_COMPANYCODE_GETLIST_invoke_response	Schema	
BAPI_COMPANYCODE_GETLIST_receive	JCA Binding	
BAPI_COMPANYCODE_GETLIST_receive	WSDL	
BAPI_COMPANYCODE_GETLIST_receive_request	Schema	
BAPI_COMPANYCODE_GETLIST_receive_response	Schema	

2. Click the icon corresponding to the exported outbound JCA binding.
The Generate WSDL and Service page opens, as shown in [Figure 9–105](#).

Figure 9–105 Generate WSDL and Service Page

3. Provide a new WSDL name and a new Business Service name in the corresponding fields.
4. In the Destination area, select the **Business Service** subfolder from the corresponding project.
5. Click **Generate**.

Select the Business Service folder in the left pane under the project folder (for example, SAP_Project) and check that the WSDL and service resources are successfully created, as shown in [Figure 9–106](#).

Figure 9–106 Checking the WSDL and Service Resources

9.5.5 Creating a Proxy Service in the Oracle Service Bus Console

This section describes how to create a Proxy Service in the Oracle Service Bus console.

1. In the Oracle Service Bus console, select the created WSDL folder in the left pane. WSDL, JCA binding, and XSD files are listed in the right pane, as shown in [Figure 9–107](#).

Figure 9–107 Contents of the WSDL Folder

Name	Type	Actions
...	Folder	
BAPI_COMPANYCODE_GETLIST_invoke	JCA Binding	
BAPI_COMPANYCODE_GETLIST_invoke	WSDL	
BAPI_COMPANYCODE_GETLIST_invoke_request	Schema	
BAPI_COMPANYCODE_GETLIST_invoke_response	Schema	
BAPI_COMPANYCODE_GETLIST_receive	JCA Binding	
BAPI_COMPANYCODE_GETLIST_receive	WSDL	
BAPI_COMPANYCODE_GETLIST_receive_request	Schema	
BAPI_COMPANYCODE_GETLIST_receive_response	Schema	

- Click the icon corresponding to the exported inbound JCA binding.
The Generate WSDL and Service page is displayed, as shown in [Figure 9–108](#).

Figure 9–108 Generate WSDL and Service Page

Generate WSDL and Service

JCA Binding Name: BAPI_COMPANYCODE_GETLIST_receive

* New WSDL Name: BAPI_COMPANYCODE_GETLIST_receive_wsd

* New Proxy Service Name: BAPI_COMPANYCODE_GETLIST_receive_PS

Destination

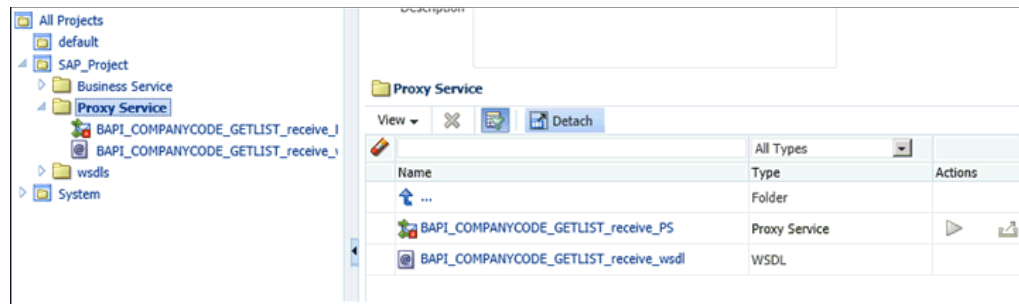
View: [Icons]

- All Projects
 - default
 - SAP_Project
 - Business Service
 - Proxy Service**
 - wsdls

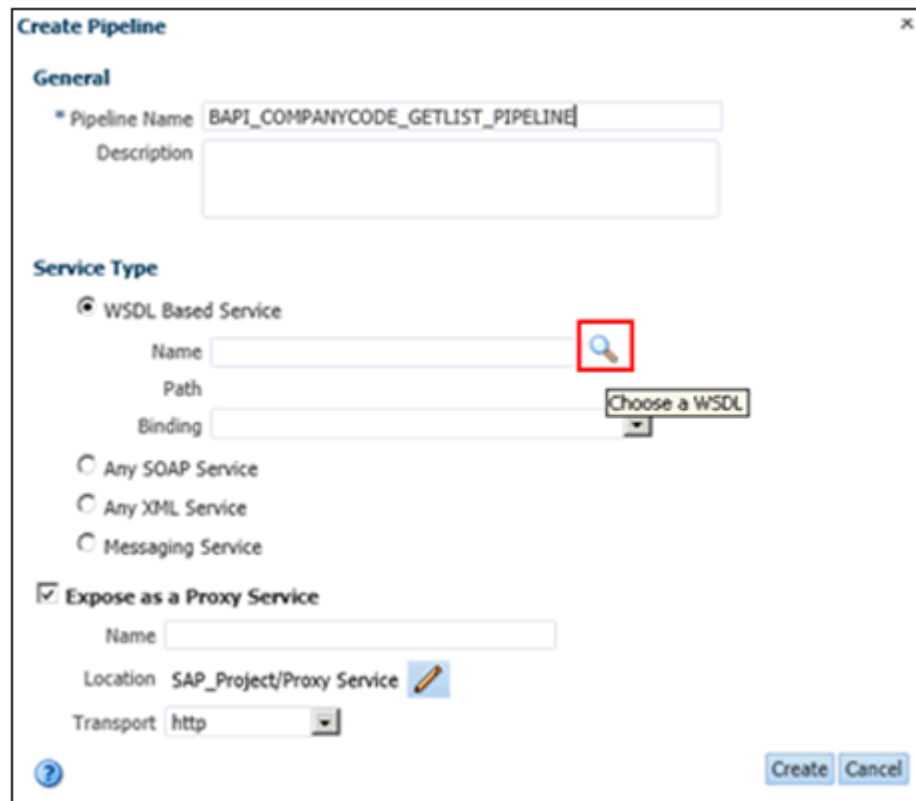
Generate Cancel

- Provide a new WSDL name and a new Proxy Service name in the corresponding fields.
- In the Destination area, select the **Proxy Service** subfolder from the corresponding project.
- Click **Generate**.

Select the Proxy Service folder in the left pane under the project folder (for example, SAP_Project) and check that the WSDL and service resources are successfully created, as shown in [Figure 9–109](#).

Figure 9–109 Checking the WSDL and Service Resources

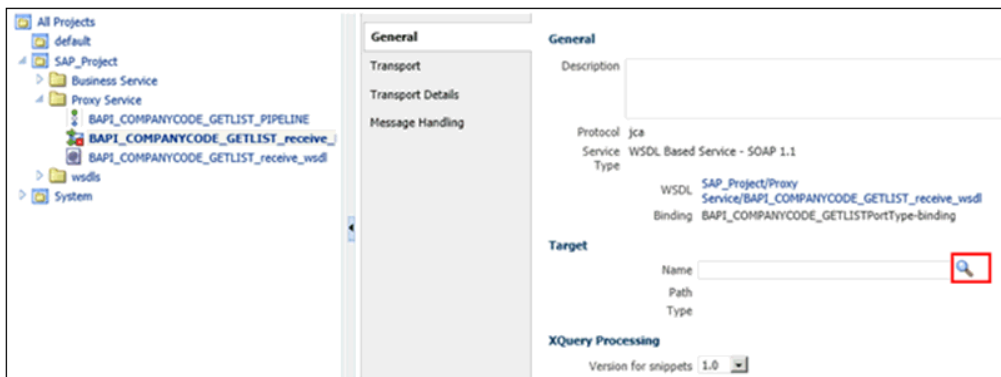
- Right-click the Proxy Service, select **Create** and then click **Pipeline**.
The Create Pipeline window is displayed, as shown in [Figure 9–110](#).

Figure 9–110 Create Pipeline Window

- Enter a Pipeline name, select **WSDL Based Service** as the Service Type, and then click the **Search** icon next to the Name field.
The Search and Select: WSDL Resource window is displayed.
- Click **Search** and in the displayed list, select **BAPI_COMPANYCODE GETLIST_receive_wsdl**, which is available under the Proxy service, and then click **Ok**.
You are returned to the Create Pipeline window, as shown [Figure 9–111](#).

Figure 9–111 Create Pipeline Window

9. Uncheck the **Expose as a Proxy Service** option and then click **Create**.
The Pipeline will be created and listed under the Proxy Service folder.
10. Double-click **BAPI_COMPANYCODE_GETLIST_receive_PS** under the Proxy Service folder and then click the **Search** icon in the Target area in left pane, as shown in [Figure 9–112](#).

Figure 9–112 Search Icon

The Search and Select: Service Resource window is displayed.

11. Click **Search** and in the displayed list select **BAPI_COMPANYCODE_GETLIST_PIPELINE**, which is available under the Proxy Service, and then click **Ok**.

12. Click **Save/Save All** in the left corner.

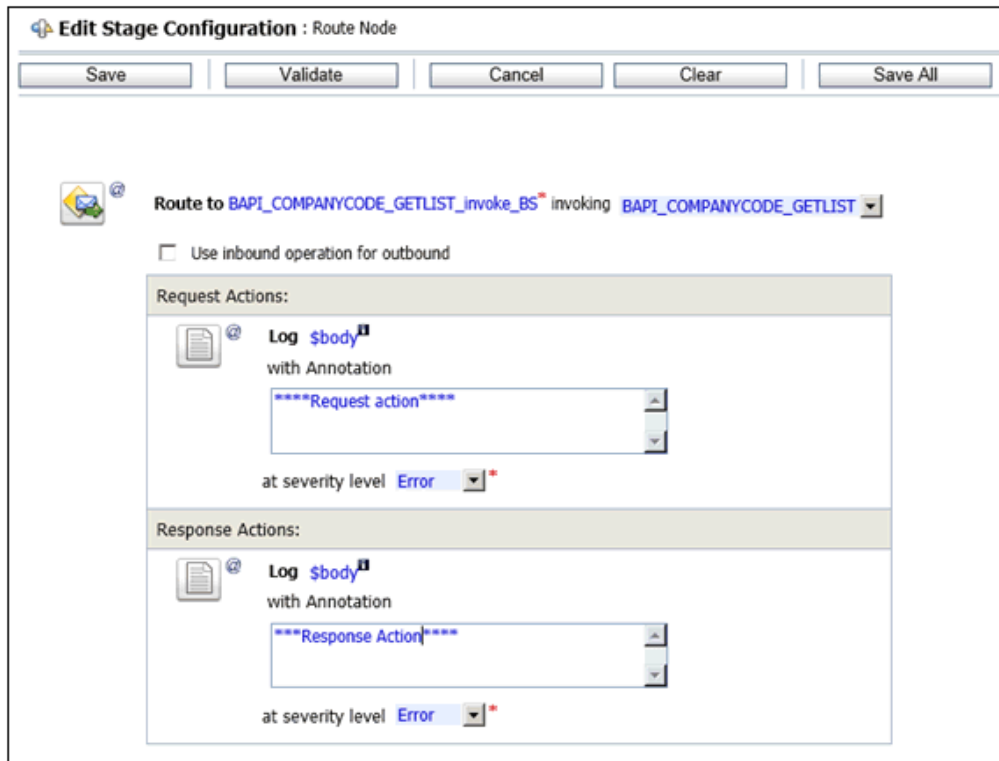
9.5.6 Configuring a Pipeline in the Proxy Service

This section describes how to configure a Pipeline in the Proxy Service.

1. Click the created Pipeline under the Proxy Service folder and click the **Open Message Flow** icon at the top of right pane.
2. In the Edit Message flow workspace area, click the displayed Proxy Service icon and select **Add Route** from the menu.
3. Click the **RouteNode1** icon and select **Edit Route** from the menu.
4. Click **Add an Action**, select **Communication** from the menu, and then click **Routing**.
5. Click **<Service>**.
6. Select the created Business Service and click **Submit**.
7. Select the **BAPI_COMPANYCODE_GETLIST** operation from the list.
8. Under **Request Actions**, click **Add an Action - Reporting**, and select **Log**.
9. Click the Log **<Expression>**.
10. In the XQuery/XSLT Expression Editor page, type **\$body**, click **Validate**, and then click **Save**.
11. Select **Error** from the Severity Level list under the Log area in the Request Actions section.
12. Under **Response Actions**, click **Add an Action - Reporting**, and select **Log**.
13. Click on the Log **<Expression>**.
14. In the XQuery/XSLT Expression Editor page, type **\$body**, click **Validate**, and then click **Save**.
15. Select **Error** from the Severity Level list under the Log area in the Response Actions section.

The final pipeline structure is shown in [Figure 9-113](#).

Figure 9–113 Final Pipeline Structure



16. Click **Save**.

You are returned to the Edit Message Flow workspace area.

17. Click **Save**.

18. Click **Activate** located at the top of the screen.

In the Confirm Session Activation window that is displayed, click **Activate**.

9.6 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.

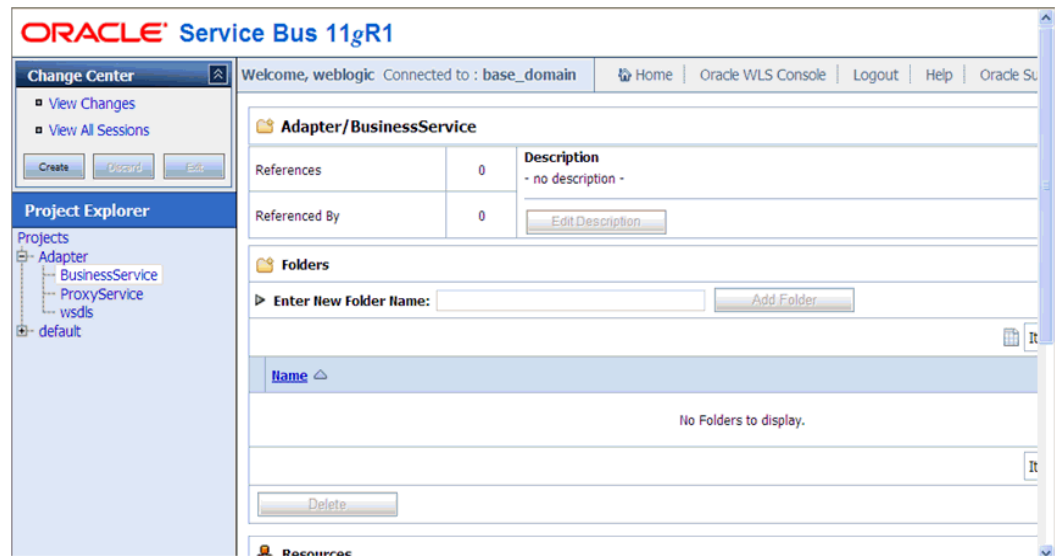
1. Start Oracle Service Bus and create the required project folder.

For more information, see [Section 9.2.1, "Starting Oracle Service Bus and Creating Project Folders"](#).

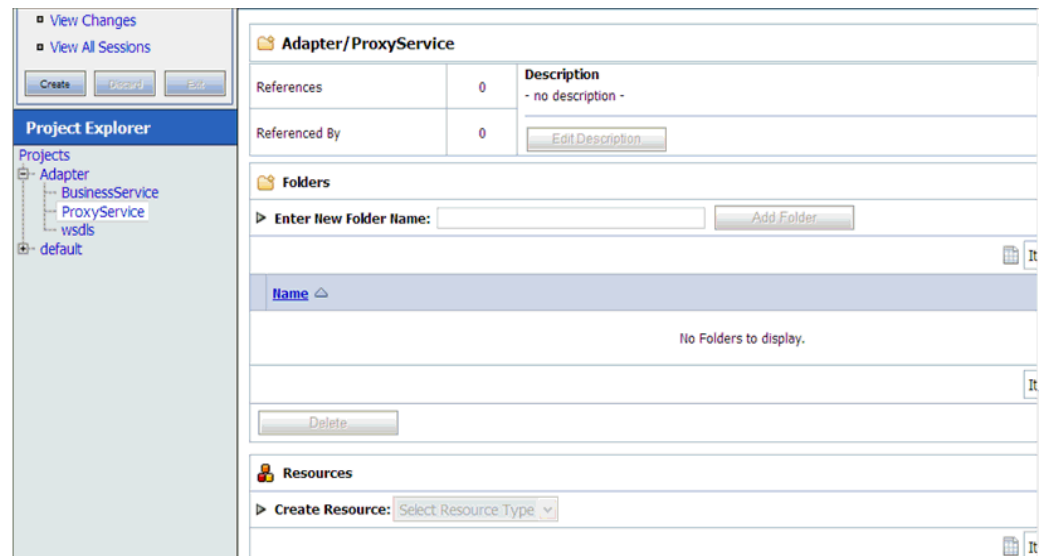
2. Generate and publish the WSDL from Application Explorer to the created project folder. Using the published WSDL, create a Business Service.

For more information, see [Section 9.2.3, "Publishing a WSDL From Application Explorer to Oracle Service Bus"](#).

3. Open the Service Bus Console page, as shown in [Figure 9–114](#).

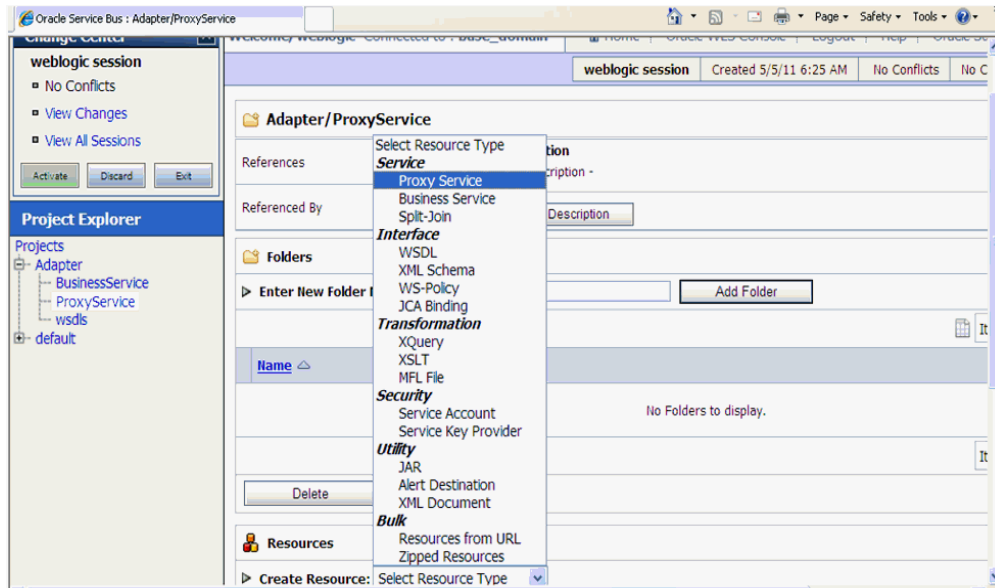
Figure 9–114 Service Bus Console

4. Select the ProxyService project folder in the left pane, and click **Create**, as shown in Figure 9–115.

Figure 9–115 Proxy Service

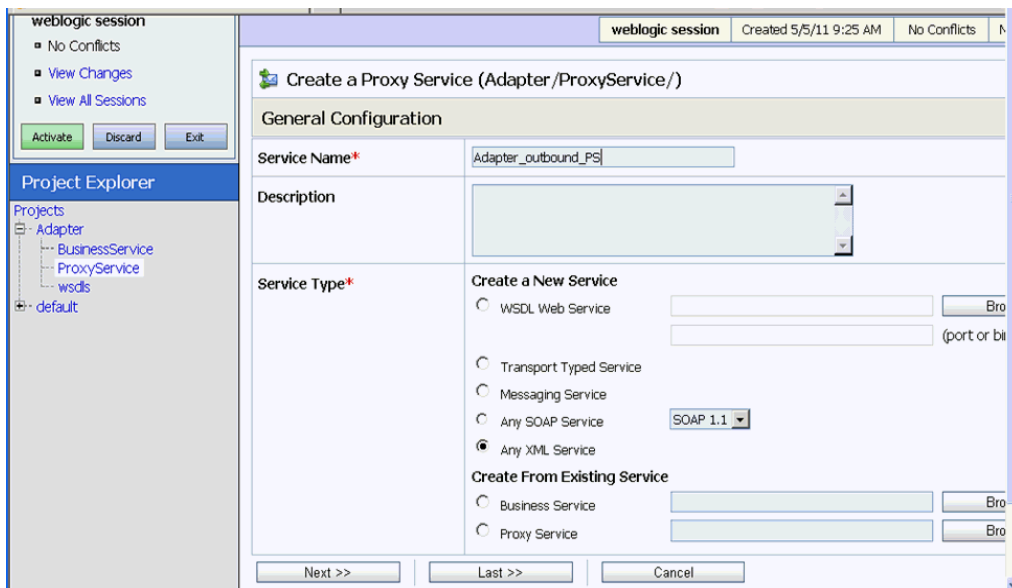
5. In the right pane, select **Proxy Service** from the Create Resource list, as shown in Figure 9–116.

Figure 9–116 Create Resource Menu



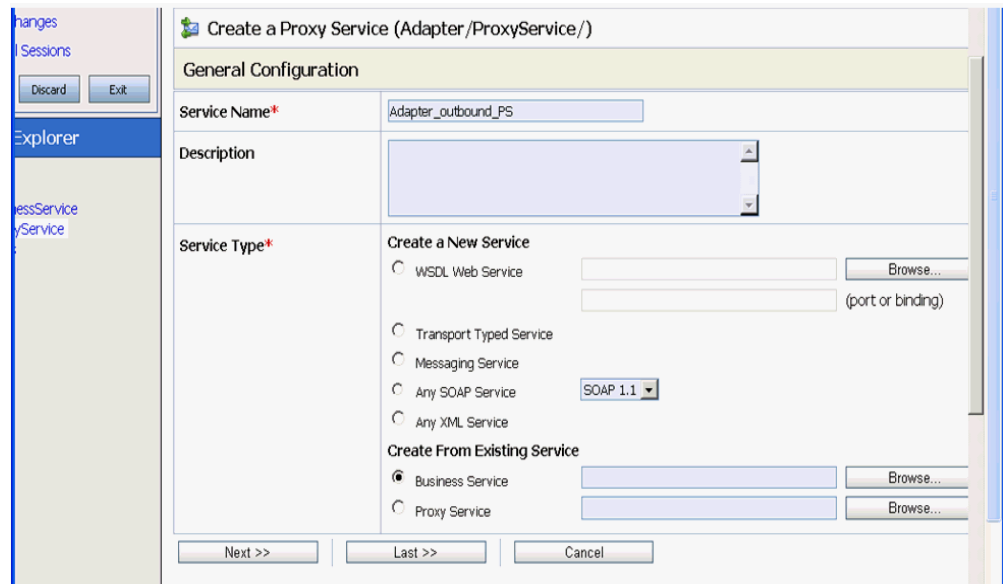
6. Enter an appropriate name in the **Service Name** field, as shown in Figure 9–117.

Figure 9–117 Service Name



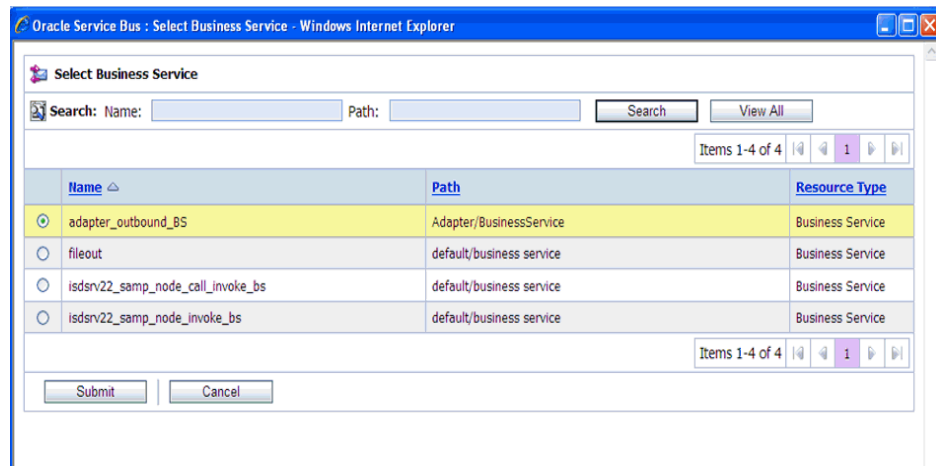
7. In the Service Type section, under Create From Existing Service, select the **Business Service** radio button and click **Browse**, as shown in Figure 9–118.

Figure 9–118 Business Service



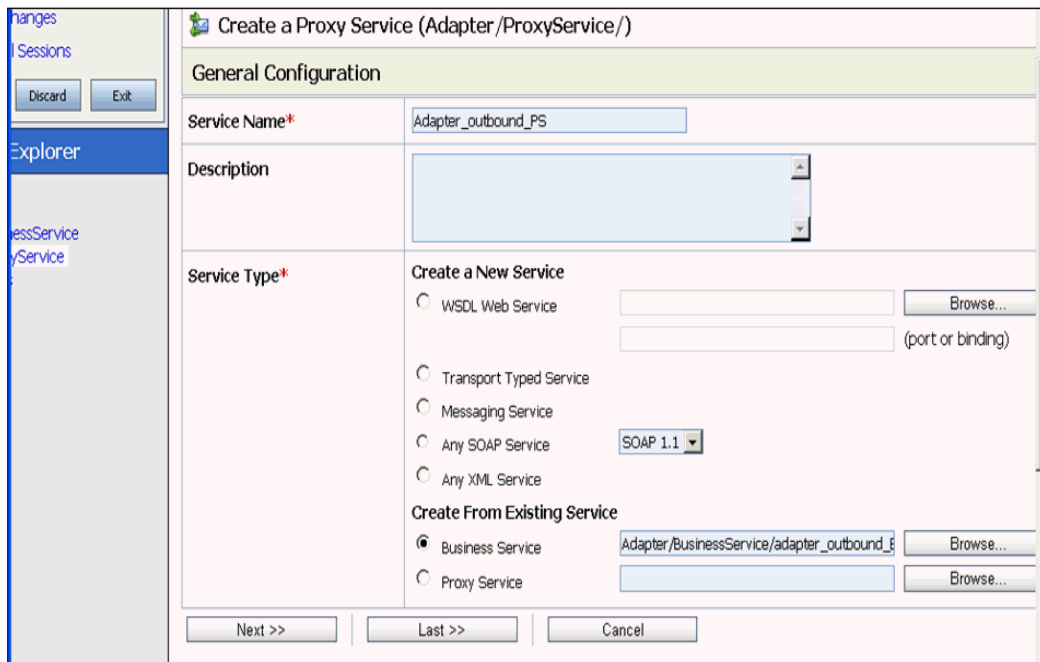
8. Select the existing business service and click **Submit**, as shown in Figure 9–119.

Figure 9–119 Existing Business Service



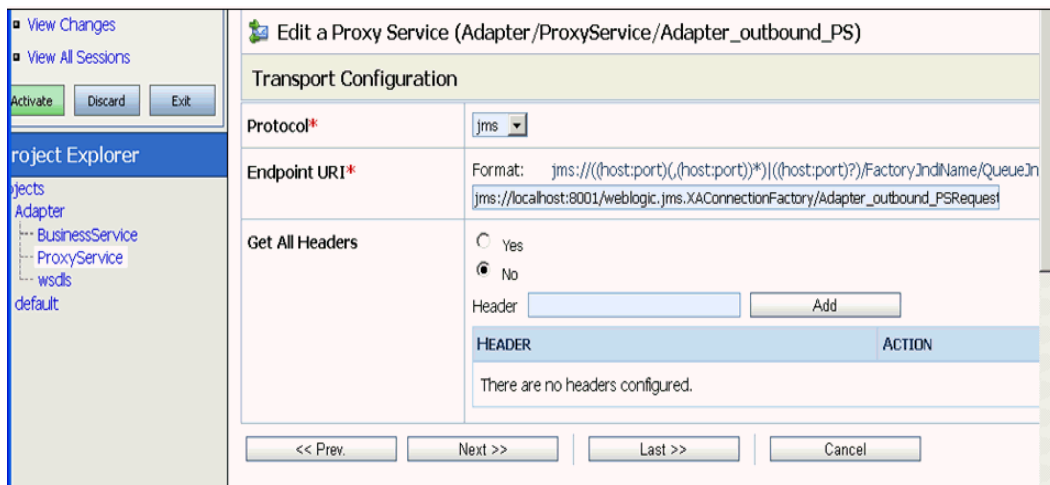
9. Click **Next**, as shown in Figure 9–120.

Figure 9–120 Next



10. Select **jms** from the Protocol list and click **Next**, as shown in Figure 9–121.

Figure 9–121 Protocol List



11. Provide the following parameters, as shown in Figure 9–122.

- 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/Adapter_outbound_PSResponse
```


Figure 9–122 Edit a Proxy Service

JMS Transport Configuration	
Destination Type	<input checked="" type="radio"/> Queue <input type="radio"/> Topic
Is Response Required	<input checked="" type="checkbox"/>
Response Pattern	<input checked="" type="radio"/> JMSCorrelationID <input type="radio"/> JMSMessageID
Response Message Type	<input type="radio"/> Bytes <input checked="" type="radio"/> 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	<input type="text"/> <input type="button" value="Browse..."/>
Advanced Settings	
<input type="button" value=" << Prev. "/> <input type="button" value=" Next >> "/> <input type="button" value=" Last >> "/> <input type="button" value=" Cancel "/> 	

12. Click Next.

The Operation Selection Configuration pane appears, as shown in [Figure 9–123](#).

Figure 9–123 Operation Selection Configuration Pane

Operation Selection Configuration	
Selection Algorithm	<input type="radio"/> Transport Header <input type="radio"/> SOAPAction Header <input type="radio"/> WS-Addressing <input type="radio"/> SOAP Header <input checked="" type="radio"/> SOAP Body Type
<input type="button" value=" << Prev. "/> <input type="button" value=" Next >> "/> <input type="button" value=" Last >> "/> <input type="button" value=" Cancel "/> 	
Top	

13. Ensure the **SOAP Body Type** is selected and click Next.

14. Enable the **Transaction Required** box and click Next, as shown in [Figure 9–124](#).

Figure 9–124 Message Handling

Message Handling	
Transaction Required	<input checked="" type="checkbox"/> Enabled
Same Transaction For Response	<input type="checkbox"/> Enabled
Content Streaming	<input type="checkbox"/> Enabled
	Buffer Type <input type="radio"/> Memory Buffer <input type="radio"/> Disk Buffer
	Compression <input type="checkbox"/> Enabled

<< Prev. Next >> Last >> Cancel

15. Click **Save**, as shown in [Figure 9–125](#).

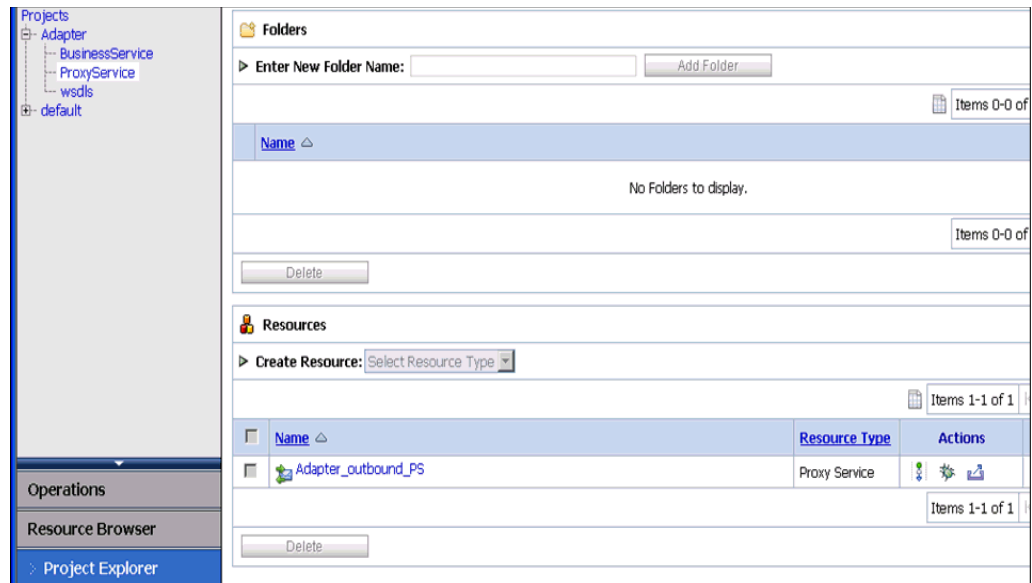
Figure 9–125 Save

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

<< Prev. Save Cancel

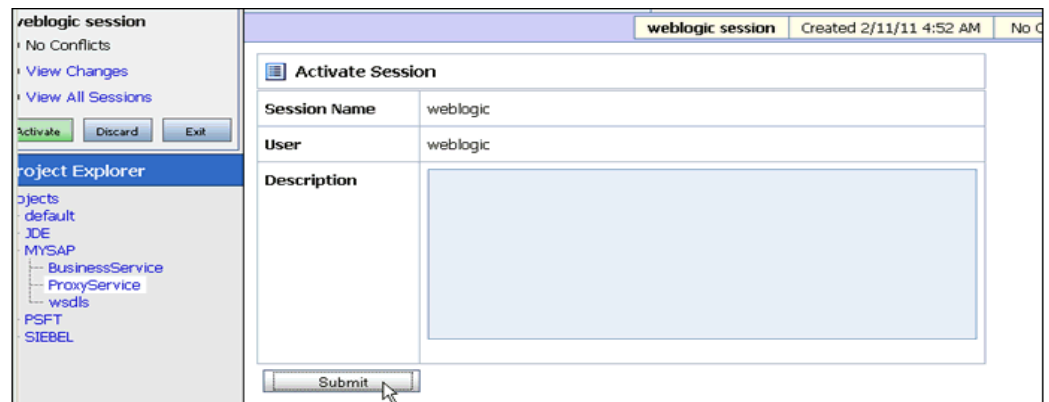
The created Proxy Service is saved, as shown in [Figure 9–126](#).

Figure 9–126 Proxy Service



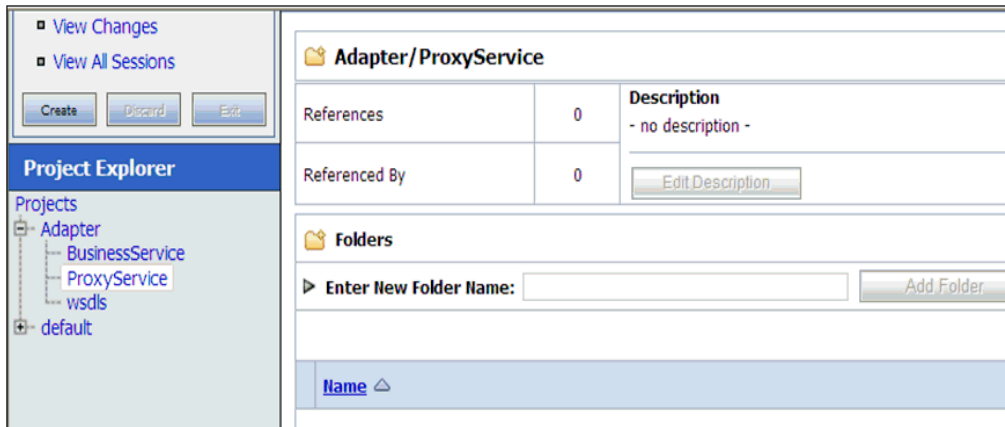
16. In the left pane, click **Activate**, and then **Submit**, as shown in [Figure 9–127](#).

Figure 9–127 Activate Session



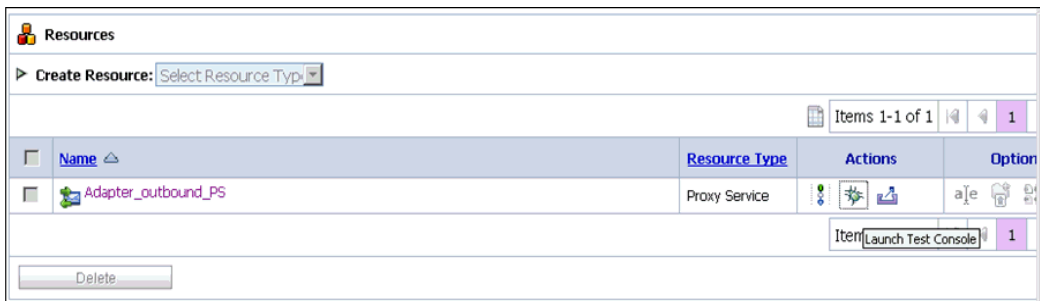
17. In the left pane, click **ProxyService** under the Projects folder, as shown in [Figure 9–128](#).

Figure 9–128 Adapter/ProxyService



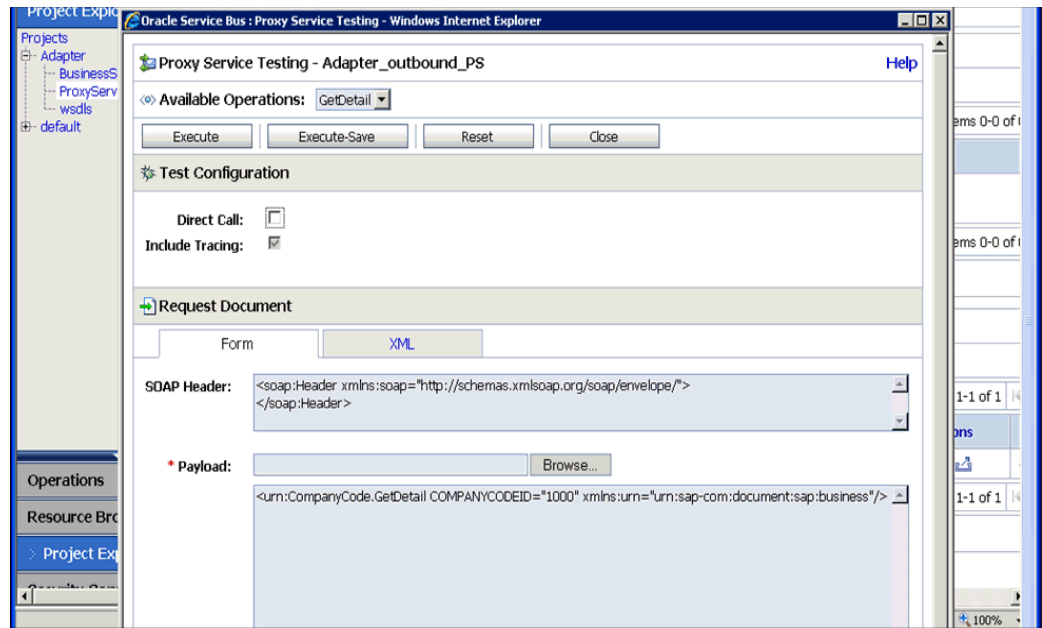
18. Click the **Launch Test Console** icon for the created Proxy Service, as shown in Figure 9–129.

Figure 9–129 Launch Test Console Icon



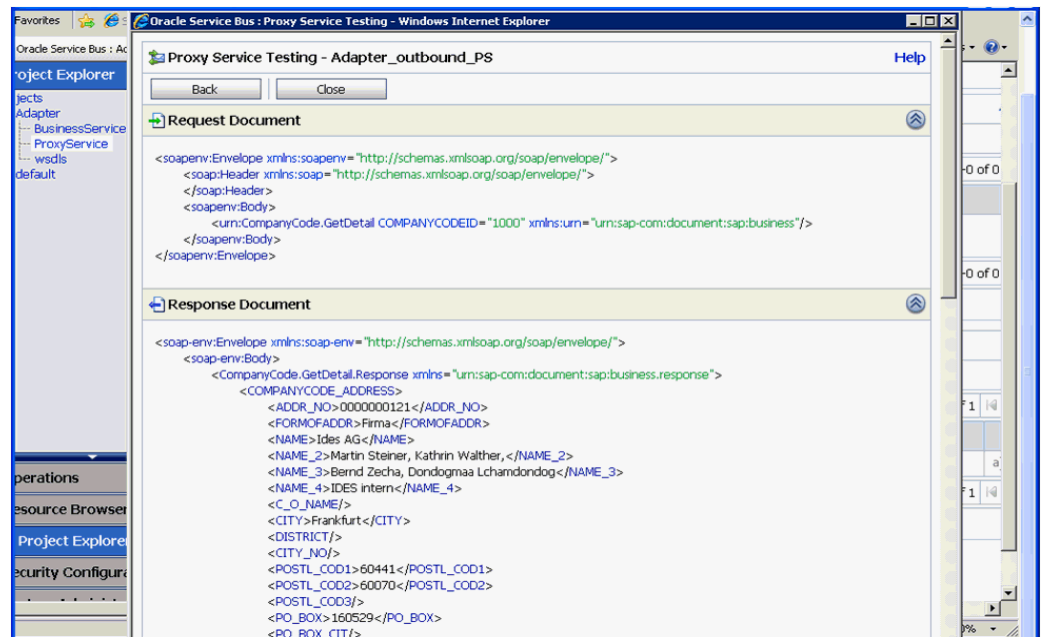
19. Provide the input values for **Payload**, uncheck the **Direct Call** box, and click **Execute**, as shown in Figure 9–130.

Figure 9–130 Proxy Service Testing



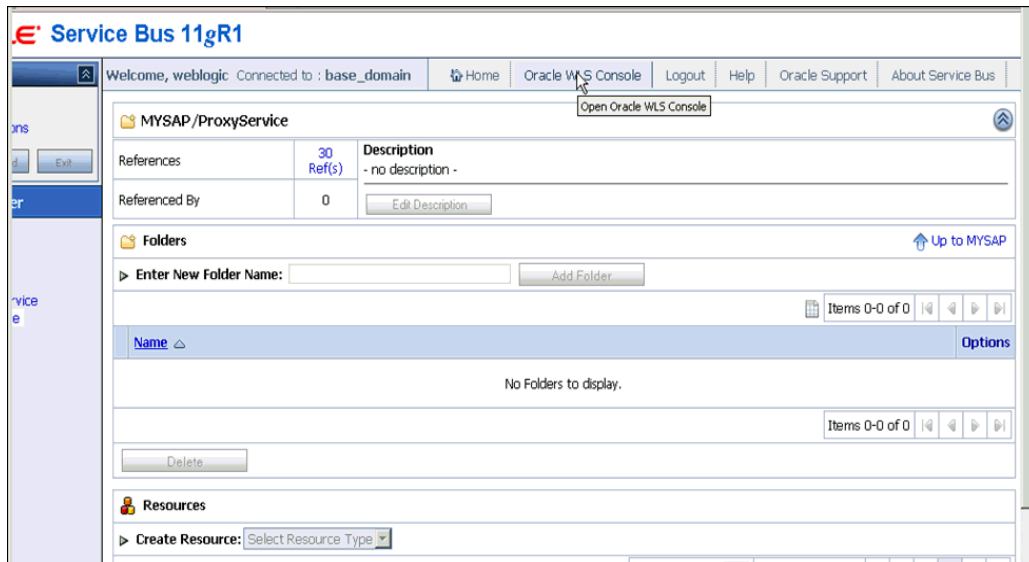
20. Review the Response document, and then click **Close**, as shown in [Figure 9–131](#).

Figure 9–131 Response Document



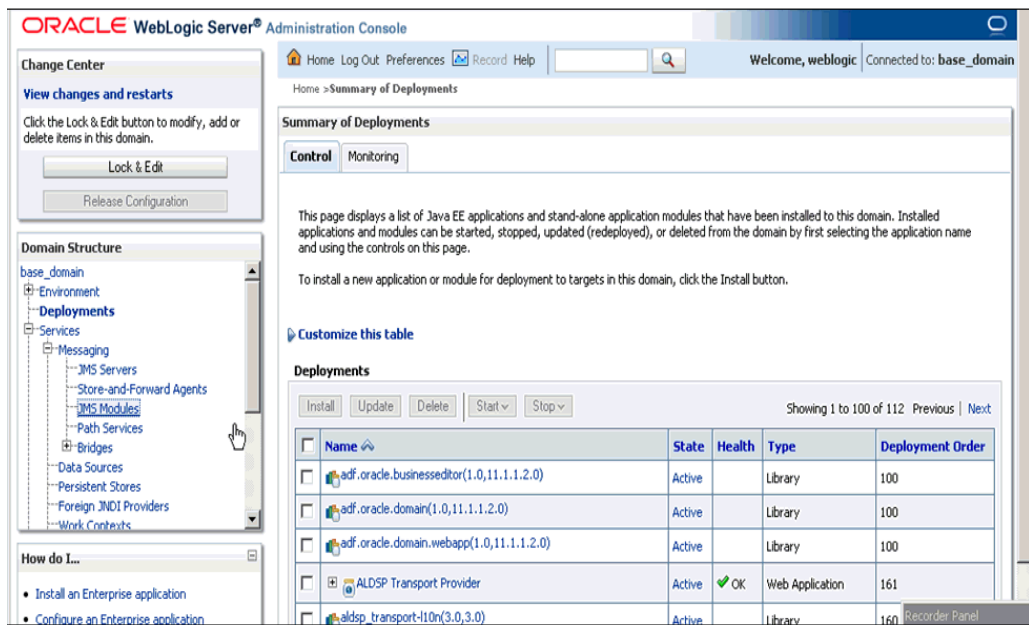
21. Click the **Oracle WLS Console** tab, as shown in [Figure 9–132](#).

Figure 9–132 ProxyService



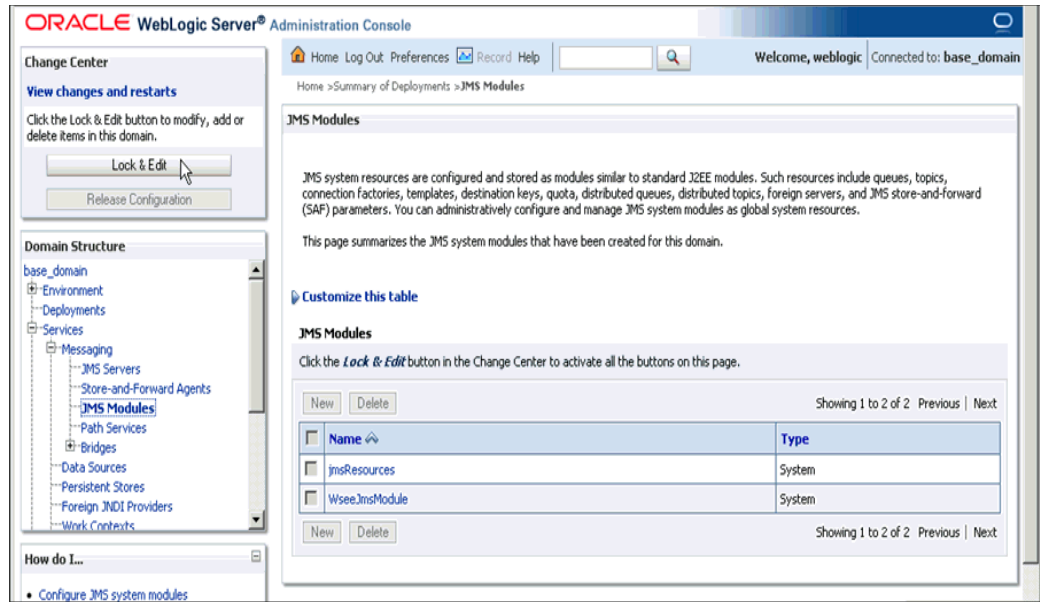
22. In the Oracle WLS Console, expand **Services**, expand **Messaging**, and click **JMS Modules**, as shown in Figure 9–133.

Figure 9–133 Oracle WLS Console



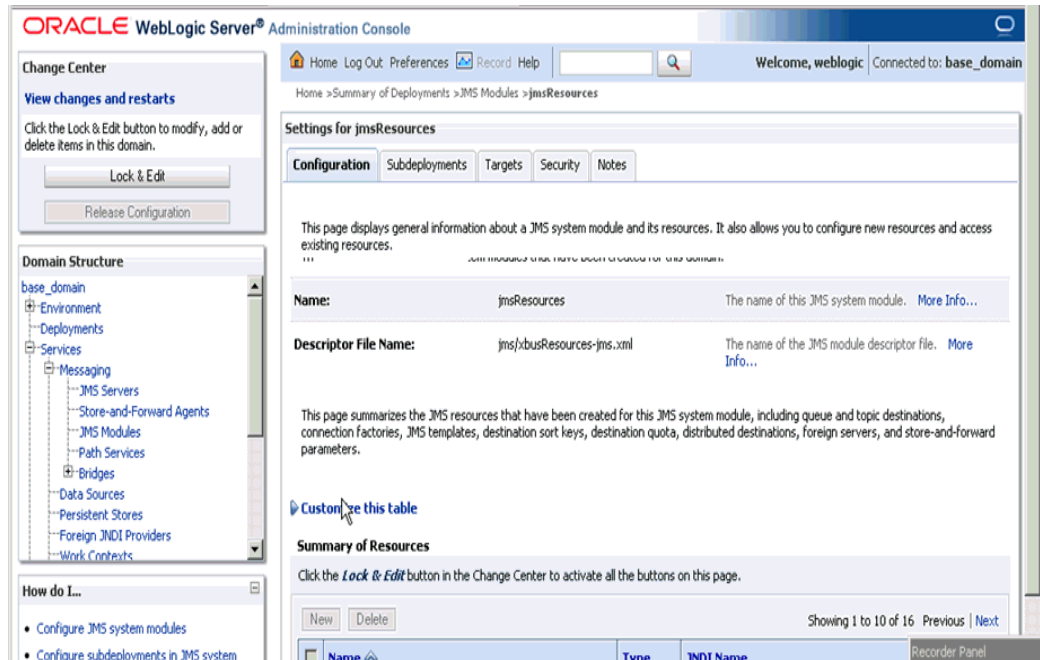
23. Click **jmsResources**, as shown in Figure 9–134.

Figure 9–134 JMS Modules



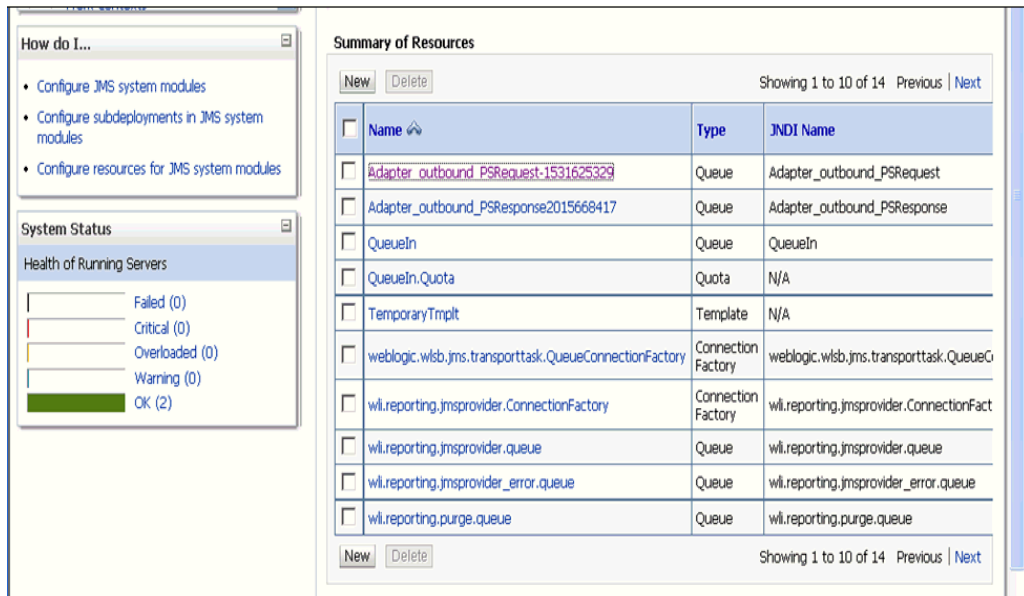
24. Click **Lock & Edit**, as shown in Figure 9–135.

Figure 9–135 Configuration Settings



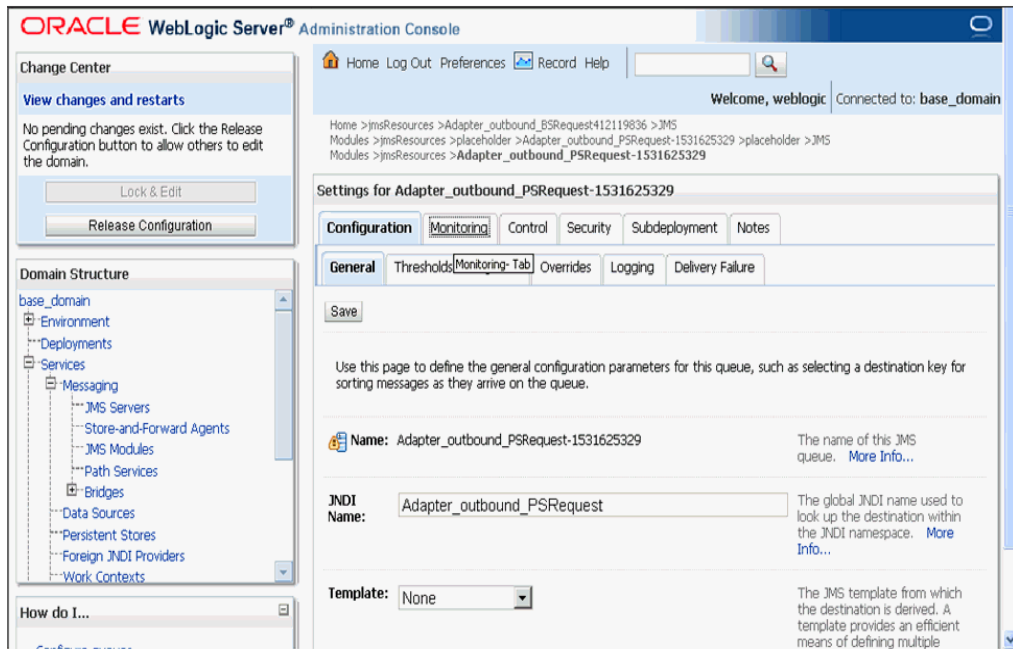
25. Click the appropriate request link, for example, **Adapter_outbound_PSRequest**, as shown in Figure 9–136.

Figure 9–136 Adapter_outbound_PSRequest



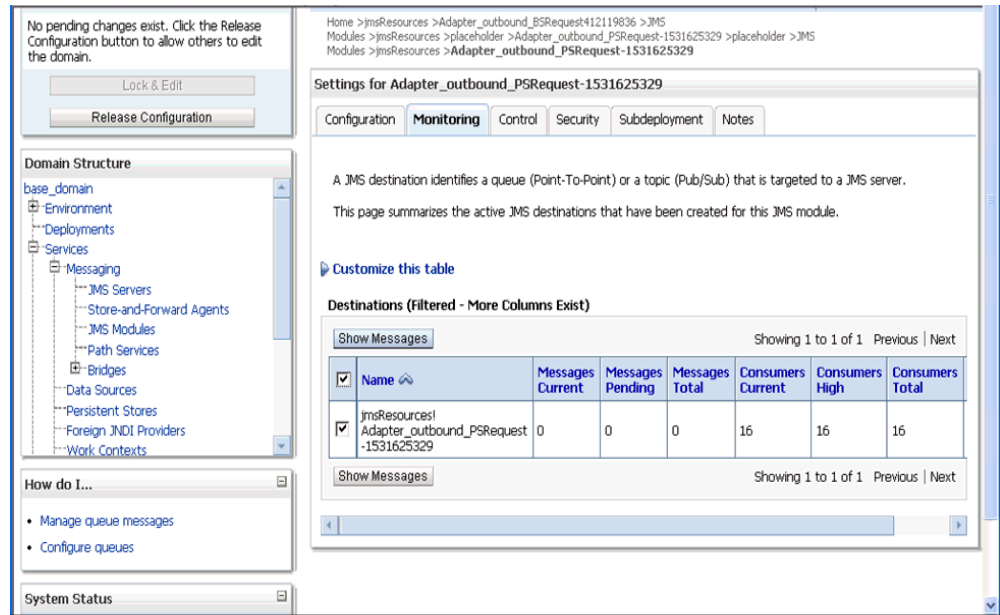
26. Click the **Monitoring** tab, as shown in Figure 9–137.

Figure 9–137 Monitoring Tab



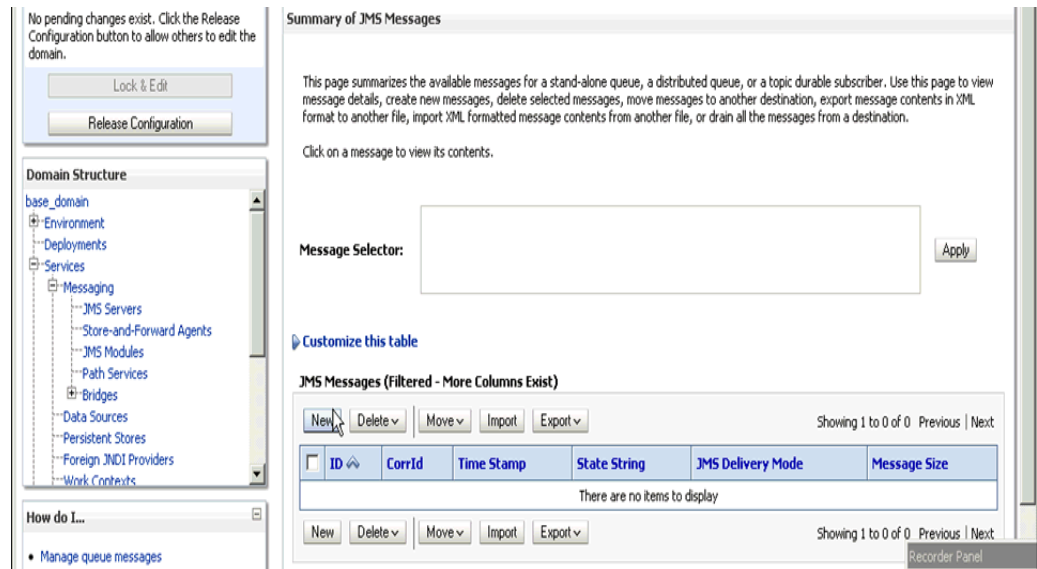
27. Enable the check box and click **Show Messages**, as shown in Figure 9–138.

Figure 9–138 Adapter Settings



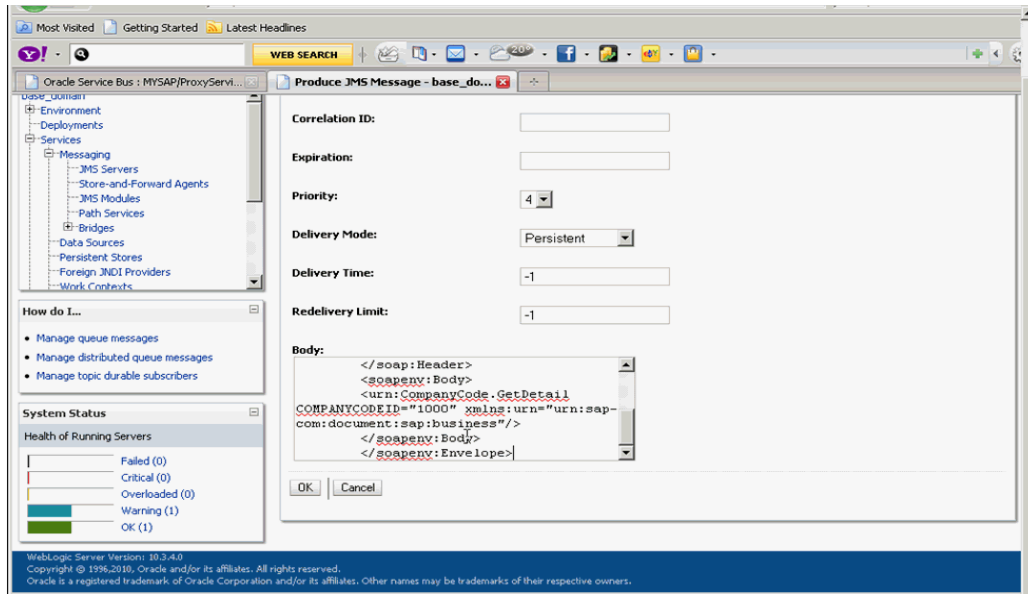
28. Click **New**, as shown in Figure 9–139.

Figure 9–139 JMS Messages



29. Provide the input payload in the **Body** field and click **OK**, as shown in Figure 9–140.

Figure 9–140 Input Payload



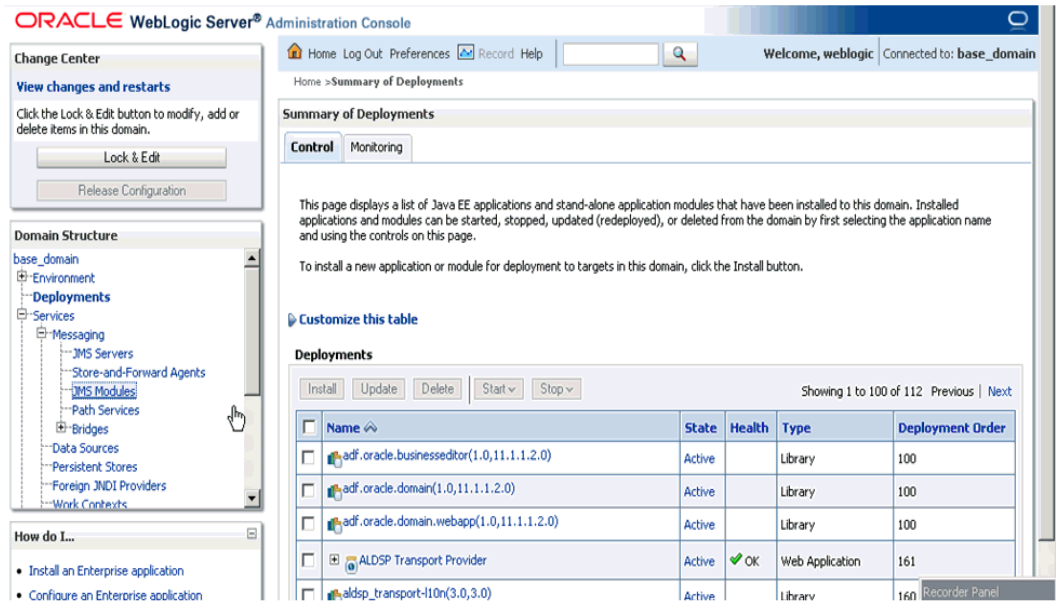
A Success message appears, as shown in Figure 9–141.

Figure 9–141 JMS Success Message



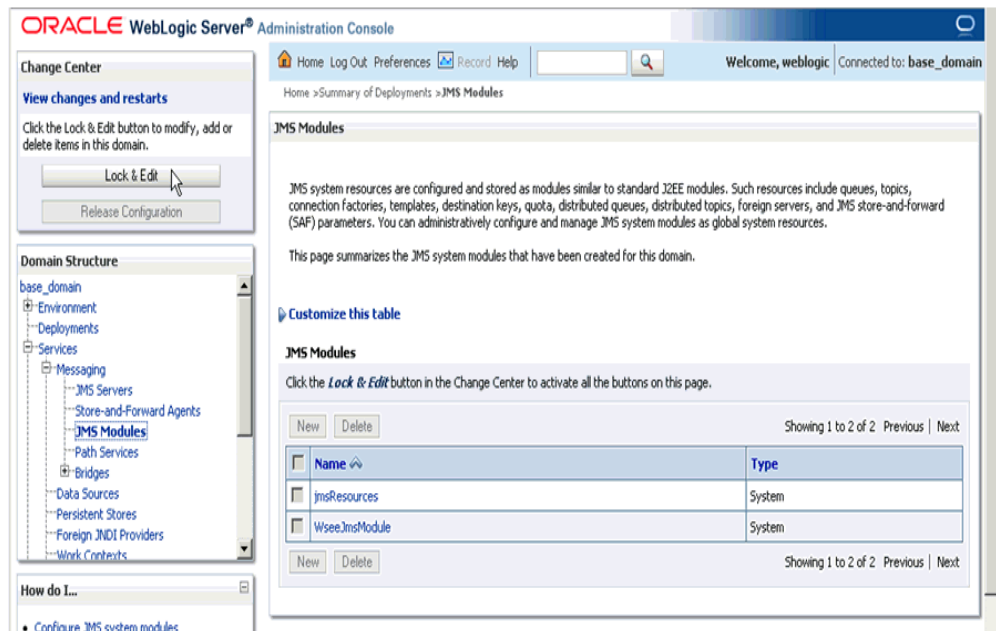
30. In the Oracle WLS Console, expand **Services**, expand **Messaging**, and click **JMS Modules**, as shown in Figure 9–142.

Figure 9–142 JMS Modules



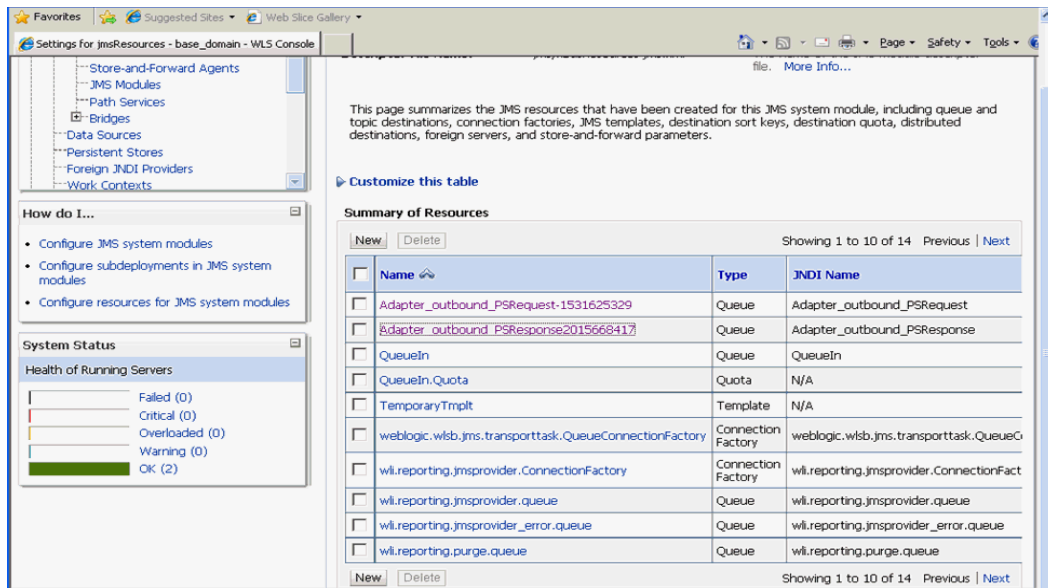
31. Click `jmsResources`, as shown in Figure 9–143.

Figure 9–143 `jmsResources`



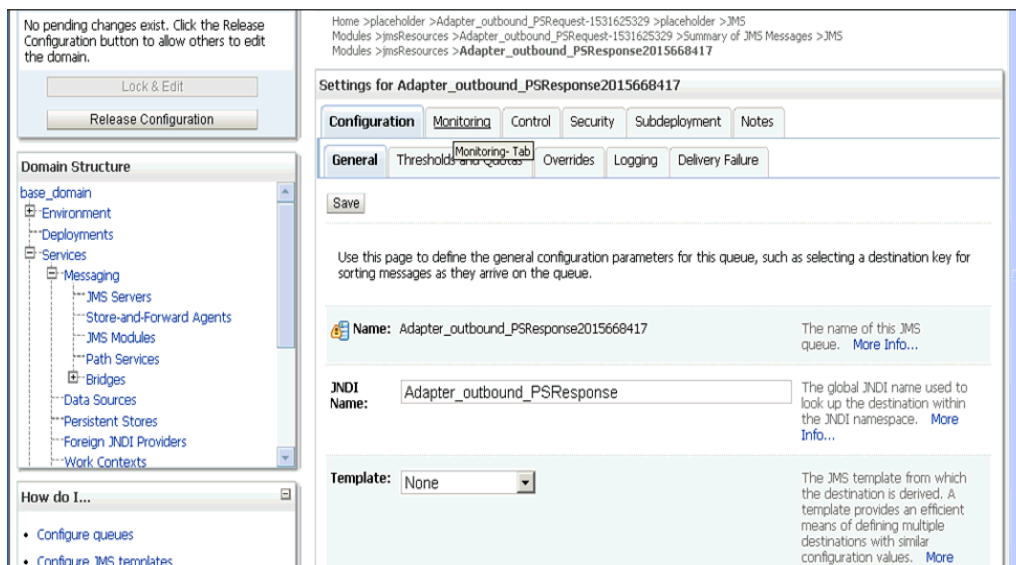
32. Click the appropriate response link, for example, `Adapter_outbound_PSResponse`, as shown in Figure 9–144.

Figure 9–144 Summary of Resources



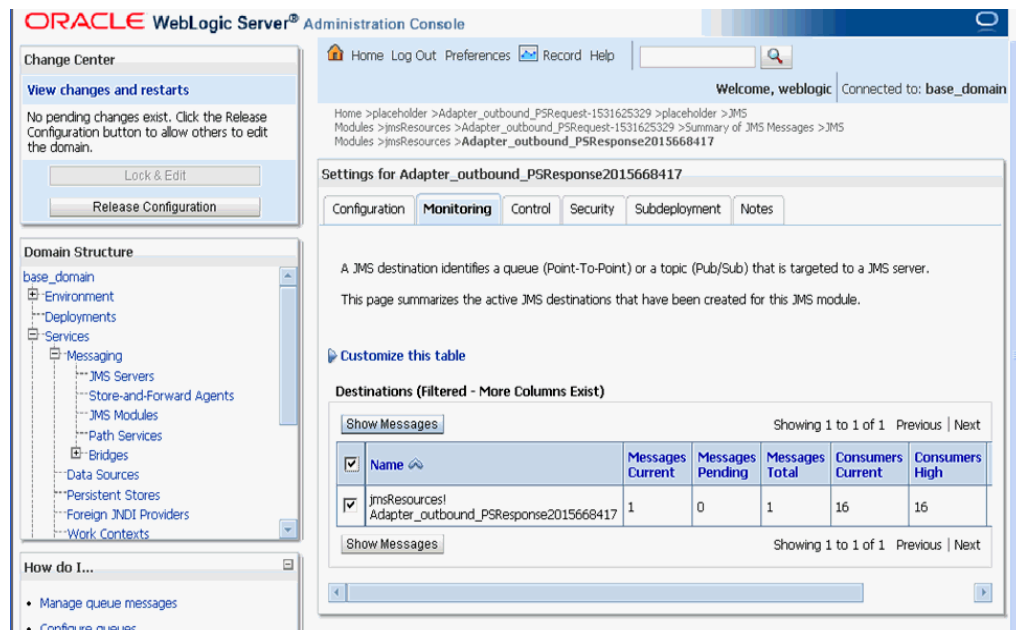
33. Click the **Monitoring** tab, as shown in Figure 9–145.

Figure 9–145 Monitoring Tab



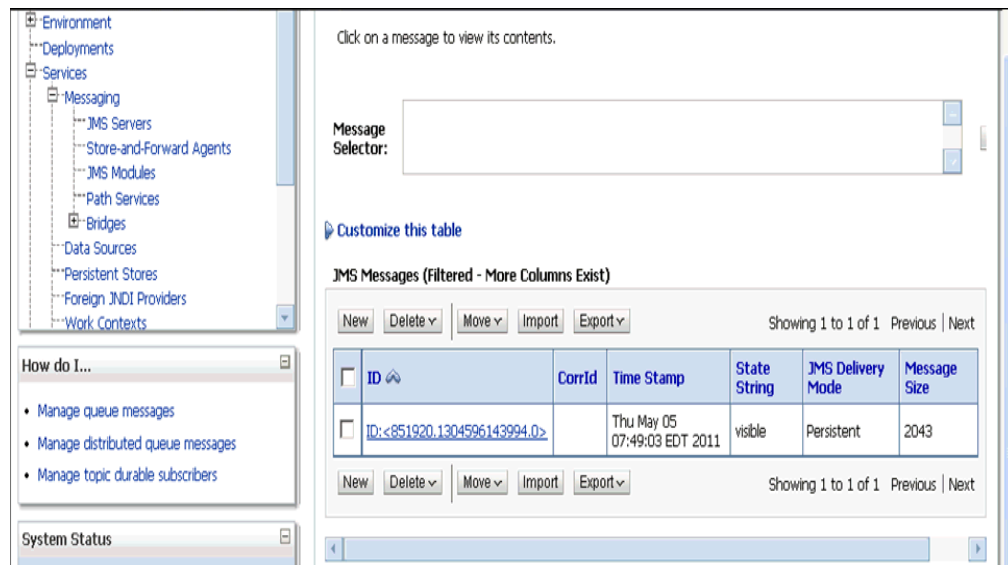
34. Enable the check box and click **Show Messages**, as shown in Figure 9–146.

Figure 9–146 Destination Messages



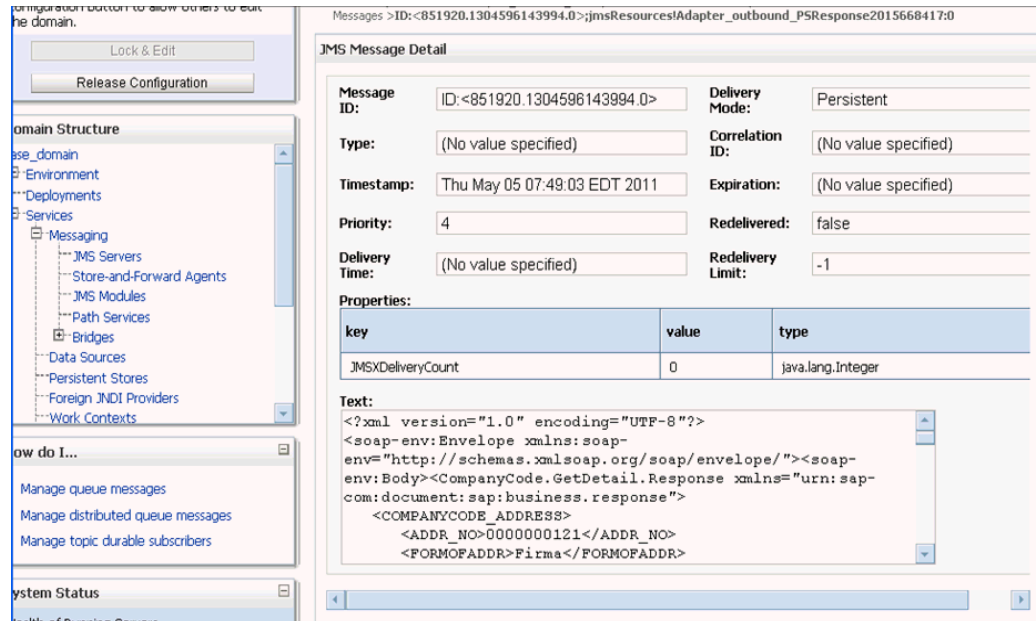
35. Click the ID link, as shown in Figure 9–147.

Figure 9–147 JMS Messages



The response document is shown under the Text field, as shown in Figure 9–148.

Figure 9–148 Response Document

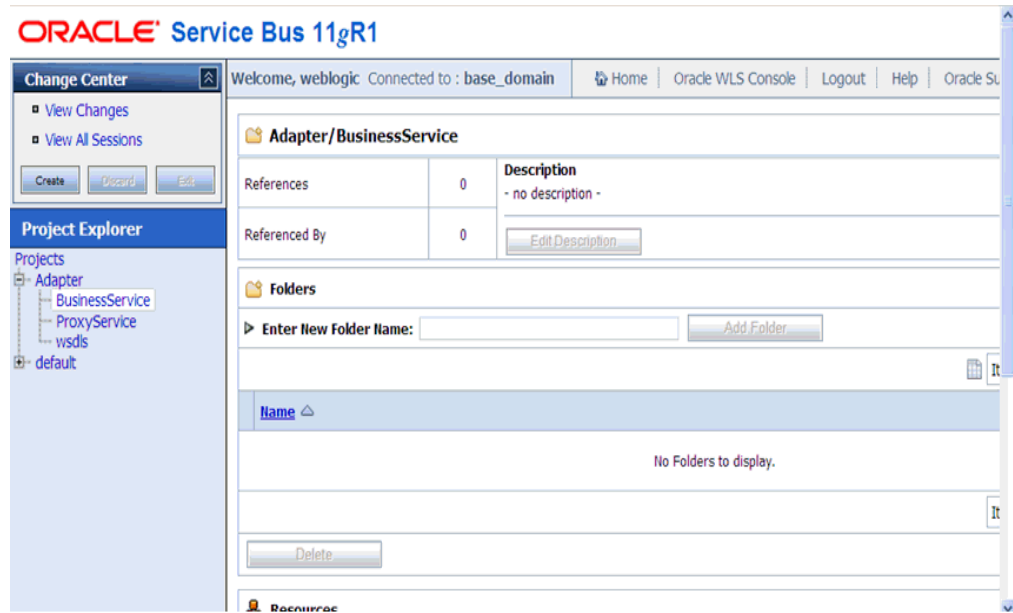


9.7 Configuring HTTP Proxy Services Using Oracle Service Bus (J2CA Configuration)

This section describes how to configure HTTP Proxy Services using Oracle Service Bus for a J2CA configuration.

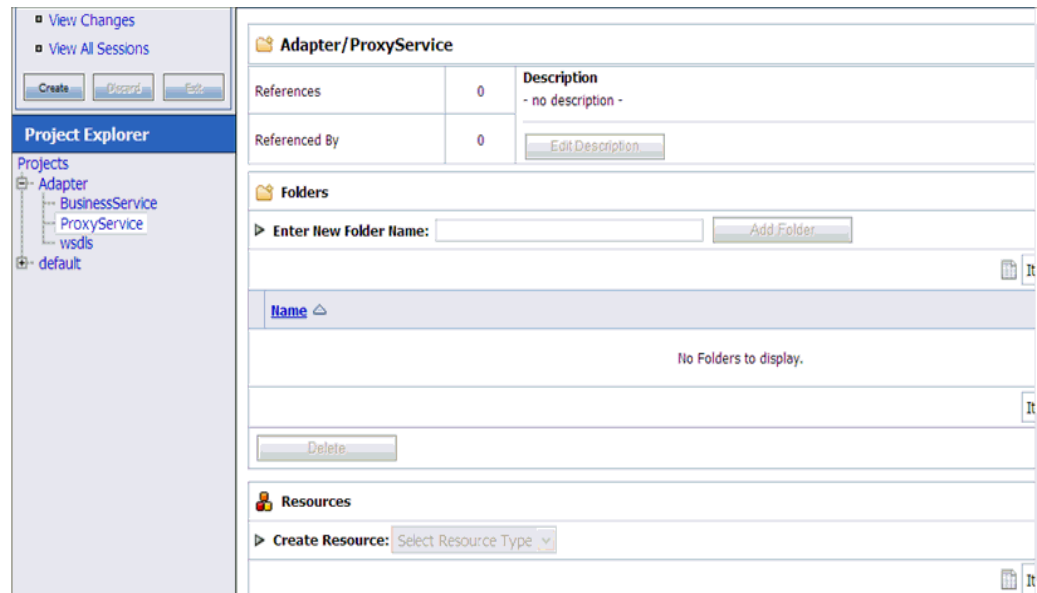
1. Start the Oracle Service Bus and create the required project folders.
For more information, see [Section 9.2.1, "Starting Oracle Service Bus and Creating Project Folders"](#).
2. Generate and publish the WSDL from Application Explorer to the created project folder, and create a Business Service using the published WSDL.
For more information, see [Section 9.2.3, "Publishing a WSDL From Application Explorer to Oracle Service Bus"](#).
3. Open the Service Bus console page, as shown in [Figure 9–149](#).

Figure 9–149 Service Bus Console Page



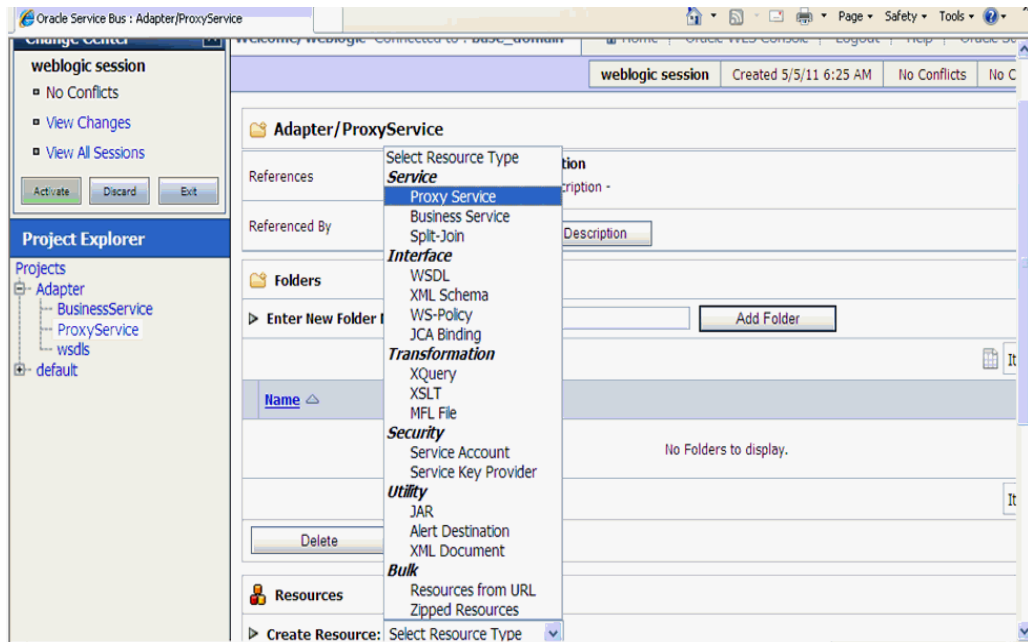
4. In the Project Explorer, select the **ProxyService** project folder, and click **Create**, as shown in Figure 9–150.

Figure 9–150 Project Explorer



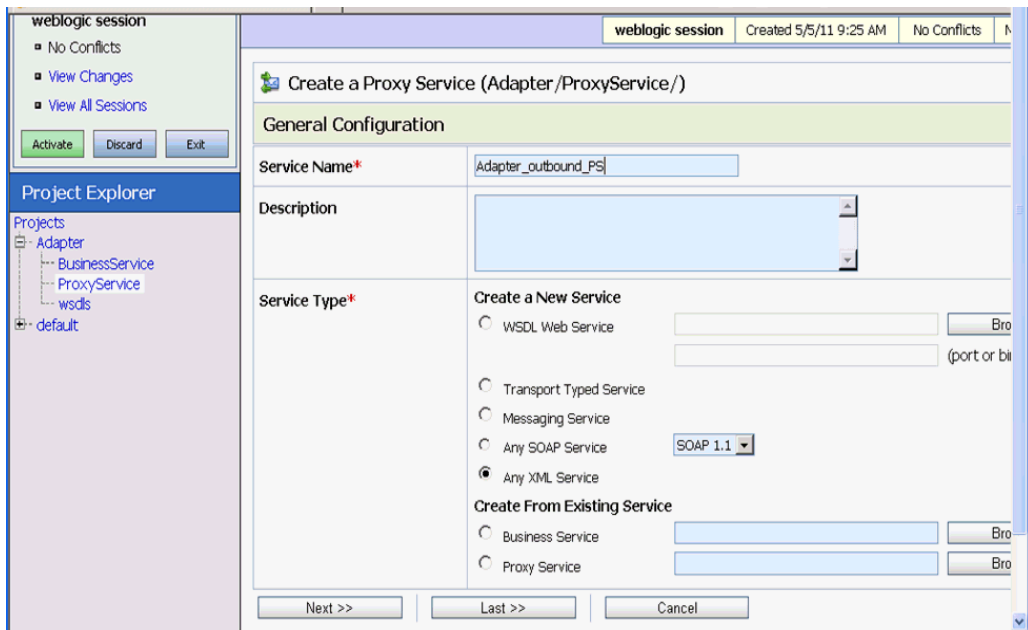
5. In the Create Resource list on the right pane, select **Proxy Service**, as shown in Figure 9–151.

Figure 9–151 Proxy Service



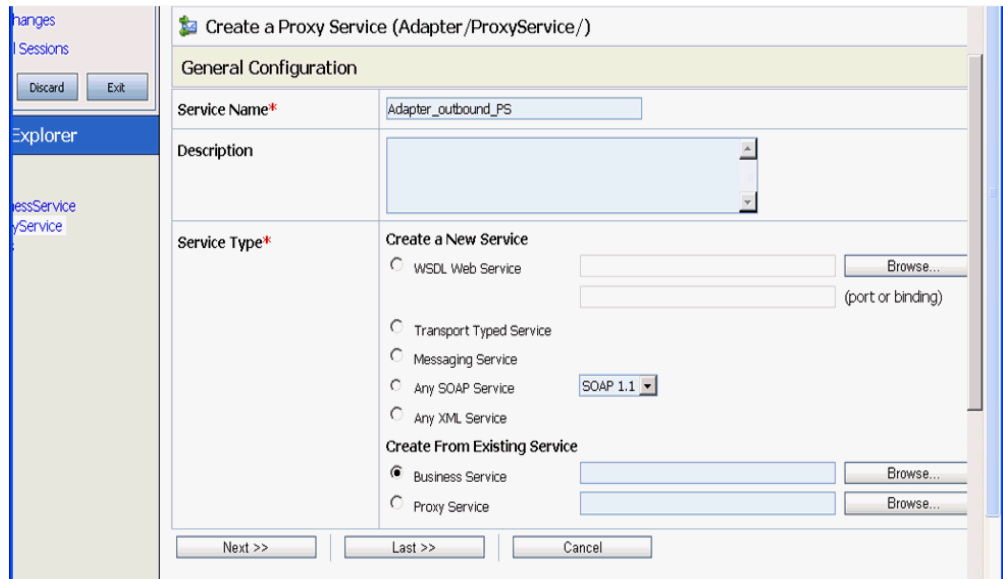
6. In the Service Name field, enter an appropriate name, as shown in [Figure 9–152](#).

Figure 9–152 Service Name



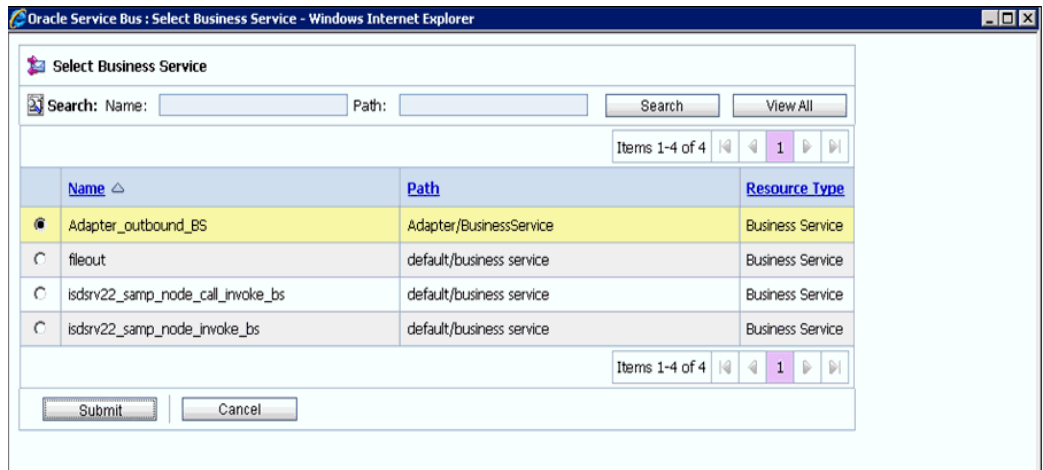
7. In the Service Type section, under Create From Existing Service, select the **Business Service** radio button and click **Browse**, as shown in [Figure 9–153](#).

Figure 9–153 General Configuration



8. Select the existing Business Service and click **Submit**, as shown in Figure 9–154.

Figure 9–154 Business Service



9. Click **Next**, as shown in Figure 9–155.

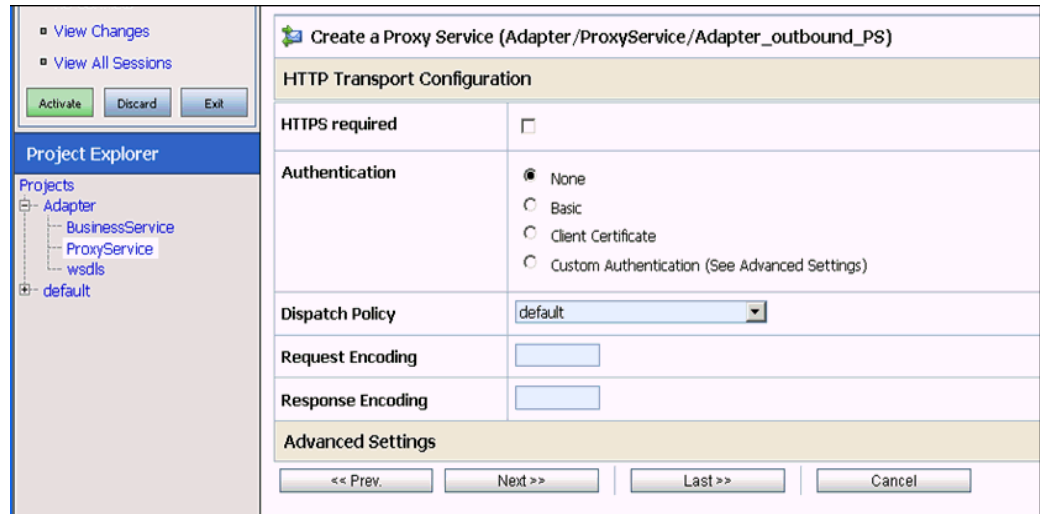
Figure 9–155 General Configuration

10. Select **http** in the Protocol list and click **Next**, as shown in [Figure 9–156](#).

Figure 9–156 Transport Configuration

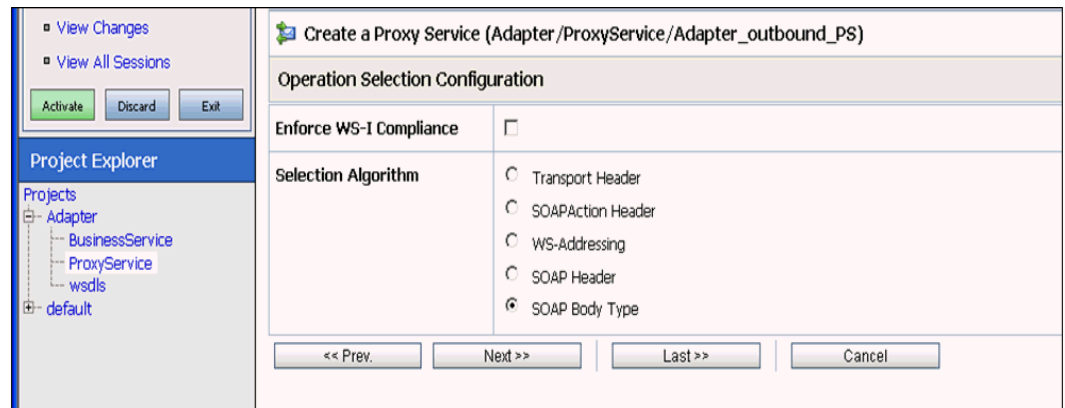
11. Click **Next**, as shown in [Figure 9–157](#).

Figure 9–157 HTTP Transport Configuration



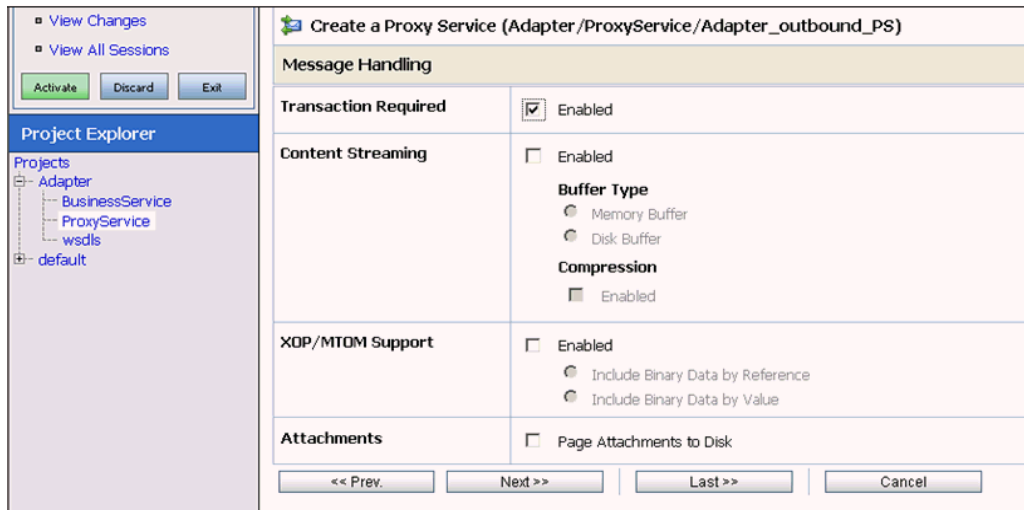
12. Click **Next**, as shown in [Figure 9–158](#).

Figure 9–158 Operation Selection Configuration



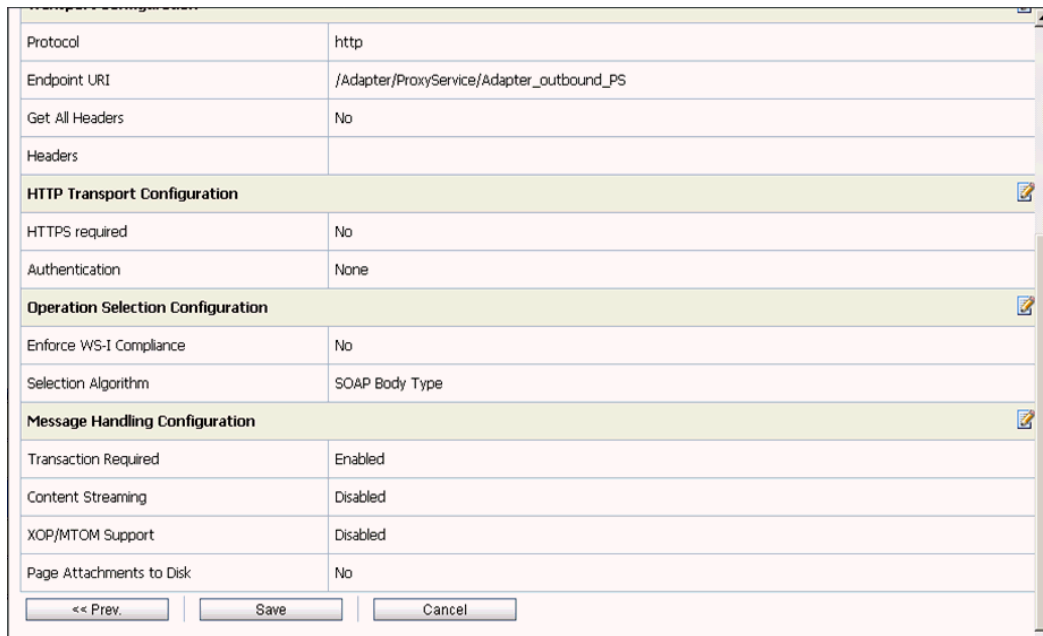
13. Enable the **Transaction Required** check box and click **Next**, as shown in [Figure 9–159](#).

Figure 9–159 Message Handling



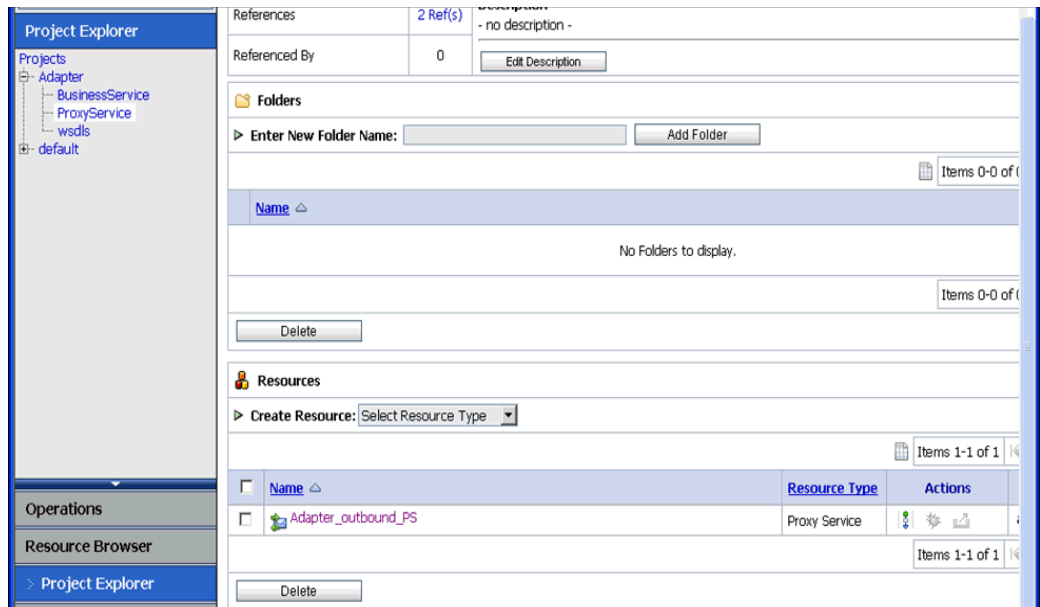
14. Click **Save**, as shown in [Figure 9–160](#).

Figure 9–160 Save



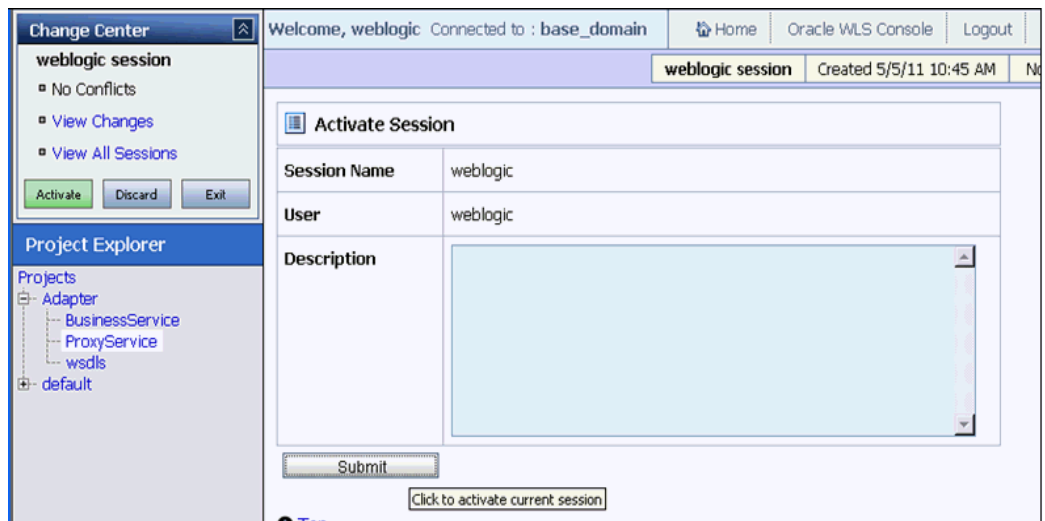
The created Proxy Service is saved, as shown in [Figure 9–161](#).

Figure 9–161 Proxy Service



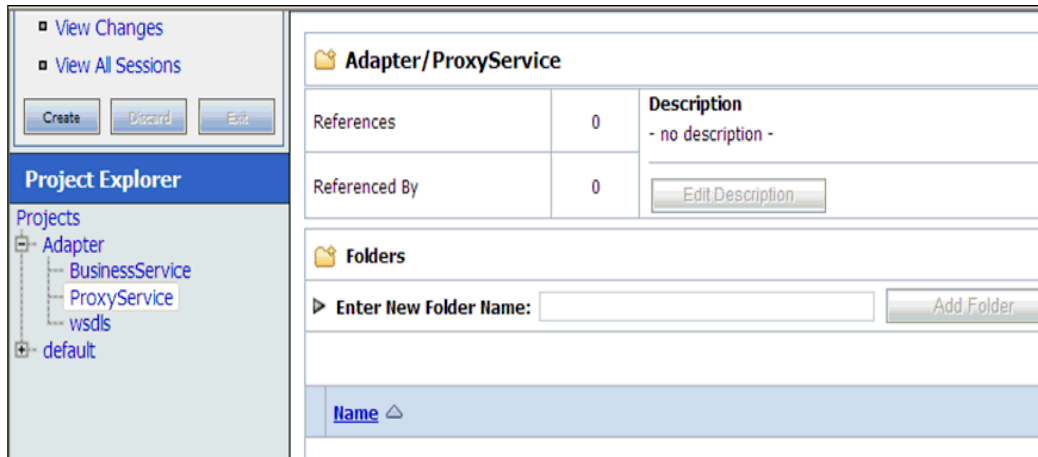
15. Click **Activate** in the left pane, and then **Submit** on the right pane, as shown in Figure 9–162.

Figure 9–162 Activate Session



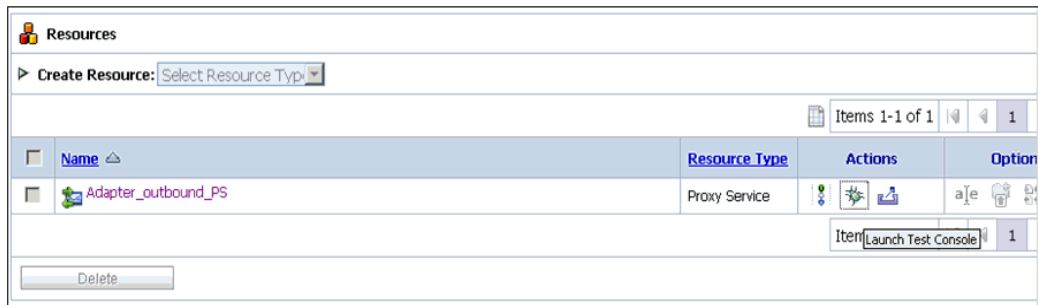
16. Click **ProxyService** in the Projects folder on the left pane, as shown in Figure 9–163.

Figure 9–163 ProxyService



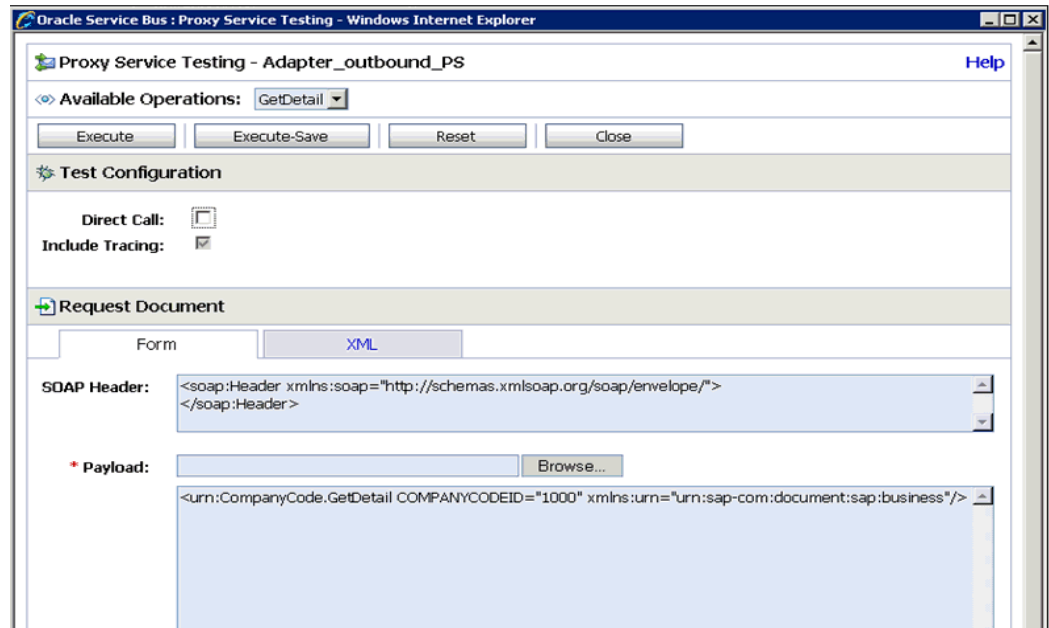
17. Click the **Launch Test Console** icon for the created Proxy Service, as shown in Figure 9–164.

Figure 9–164 Launch Test Console



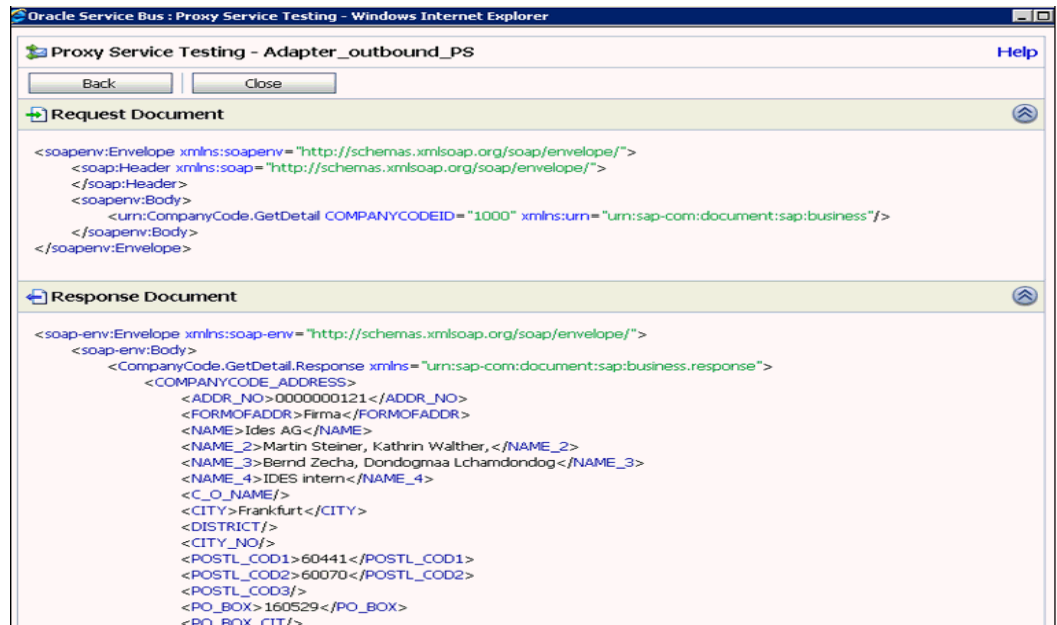
18. Uncheck the **Direct Call** check box, provide the input values for **Payload**, and click **Execute**, as shown in Figure 9–165.

Figure 9–165 Proxy Service Testing



19. Review the Response Document, as shown in Figure 9–166.

Figure 9–166 Response Document



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

Oracle Application Adapter for SAP R/3 integrates seamlessly with Oracle JDeveloper to facilitate Web service integration.

This chapter contains the following sections:

- [Section 10.1, "Configuring an OSB Outbound Process Using JDeveloper \(J2CA Configuration\)"](#)
- [Section 10.2, "Configuring an OSB Inbound Process Using JDeveloper \(J2CA Configuration\)"](#)
- [Section 10.3, "Configuring an OSB Outbound Process Using JDeveloper \(BSE Configuration\)"](#)
- [Section 10.4, "Working With Synchronous SAP R/3 Events of Oracle Service Bus Using JDeveloper \(J2CA Configuration\)"](#)
- [Section 10.5, "Configuring a JMS Inbound Process Using JDeveloper \(J2CA Configuration\)"](#)
- [Section 10.6, "Configuring a JMS Outbound Process Using JDeveloper \(J2CA Configuration\)"](#)
- [Section 10.7, "Configuring an HTTP Outbound Process Using JDeveloper \(J2CA Configuration\)"](#)

10.1 Configuring an OSB Outbound Process Using JDeveloper (J2CA Configuration)

This section describes how to configure an OSB outbound process to your SAP R/3 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\SAP_Samples.zip\SAP_Samples\OSB_JDeveloper\J2CA\SAP_Sample_J2CA_OSB_Outbound_Project
```

This section includes the following topics:

- [Section 10.1.1, "Creating a Service Bus Application for OSB"](#)
- [Section 10.1.2, "Defining an OSB Outbound Process"](#)

- [Section 10.1.3, "Deploying the OSB Outbound Process"](#)

Prerequisites

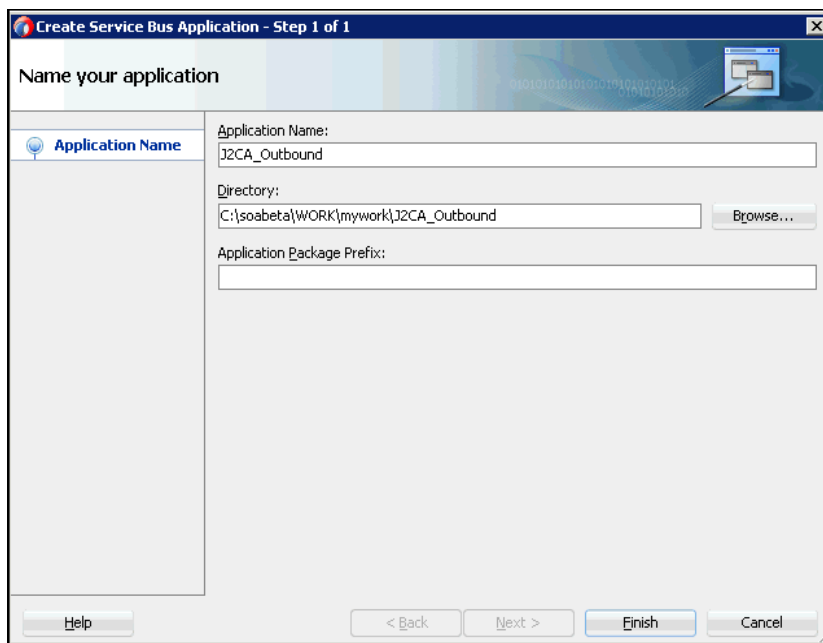
Before you design an OSB outbound process, you must generate the respective WSDL file using Application Explorer. For more information, see [4.8 , "Generating WSDL \(J2CA Configurations Only\)"](#) on page 4-18.

10.1.1 Creating a Service Bus Application for OSB

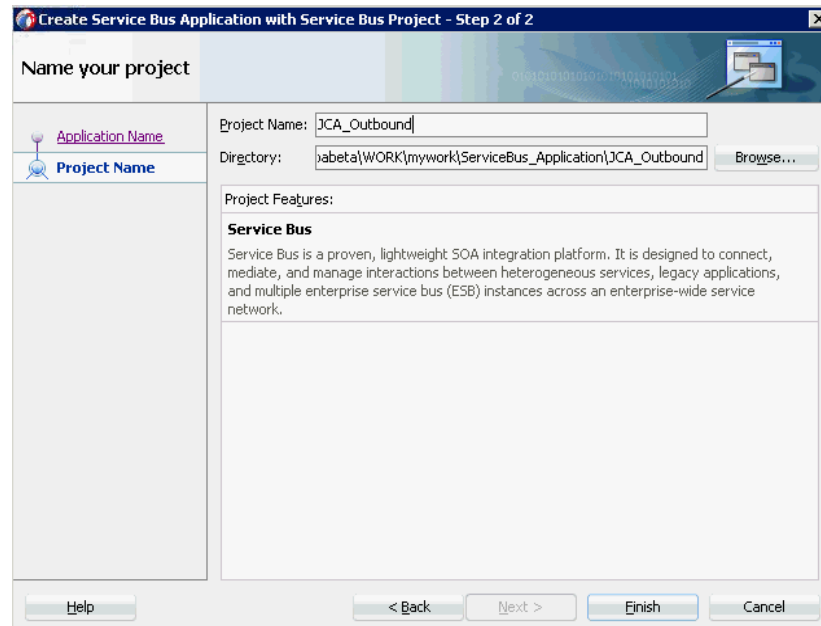
Perform the following steps in JDeveloper to create a service bus application for OSB.

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 10-1](#).

Figure 10-1 Name Your Application Pane



3. Enter a project name (for example, JCA_Outbound), and click **Finish**, as shown in [Figure 10-2](#).

Figure 10–2 Name Your Project Pane

10.1.2 Defining an OSB Outbound Process

This section describes how to define an OSB outbound process. The following topics are included:

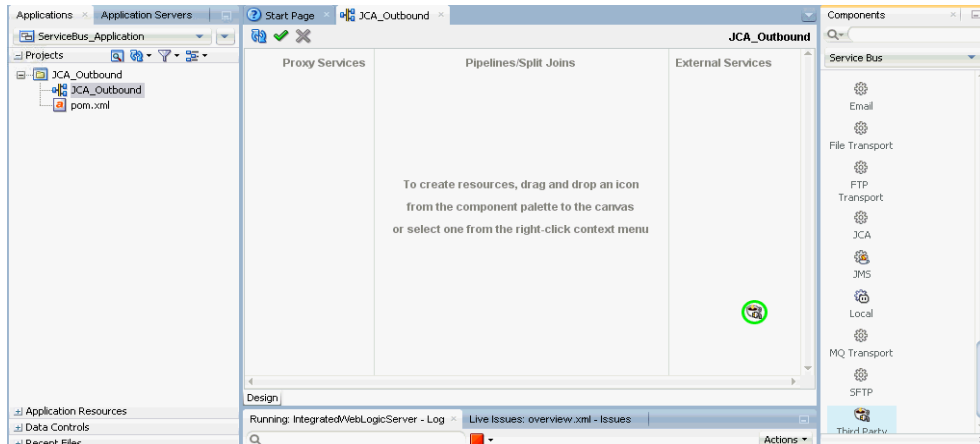
- [Section 10.1.2.1, "Configuring a Third-Party Adapter Service Component"](#)
- [Section 10.1.2.2, "Configuring a File Transport Type Business Service"](#)
- [Section 10.1.2.3, "Creating a Proxy Service With Pipeline"](#)
- [Section 10.1.2.4, "Configuring the Routing Rules"](#)

10.1.2.1 Configuring a Third-Party Adapter Service Component

Perform the following steps to create a third party adapter service component along with the Business Service:

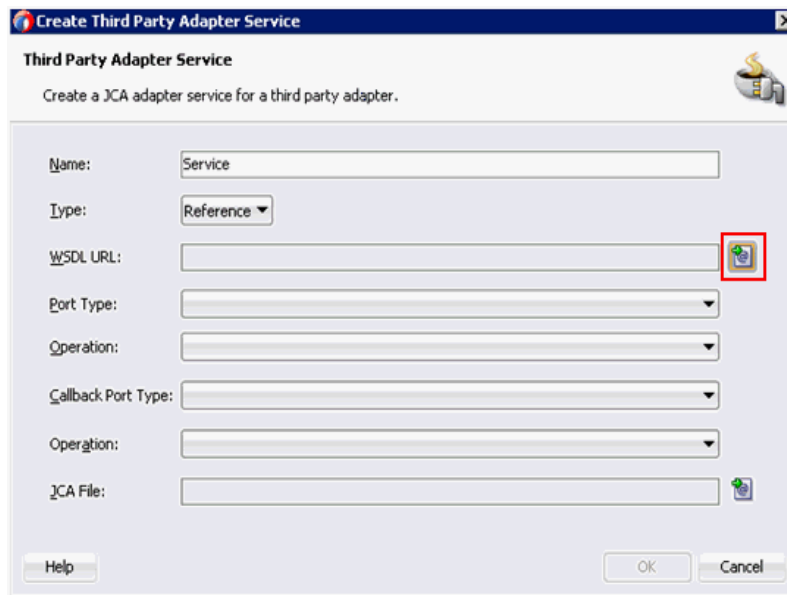
1. Drag and drop the **Third Party Adapter** component from the Service Bus Components pane to the External Services pane, as shown in [Figure 10–3](#).

Figure 10–3 Third Party Adapter Component



The Create Third Party Adapter Service dialog is displayed, as shown in [Figure 10–4](#).

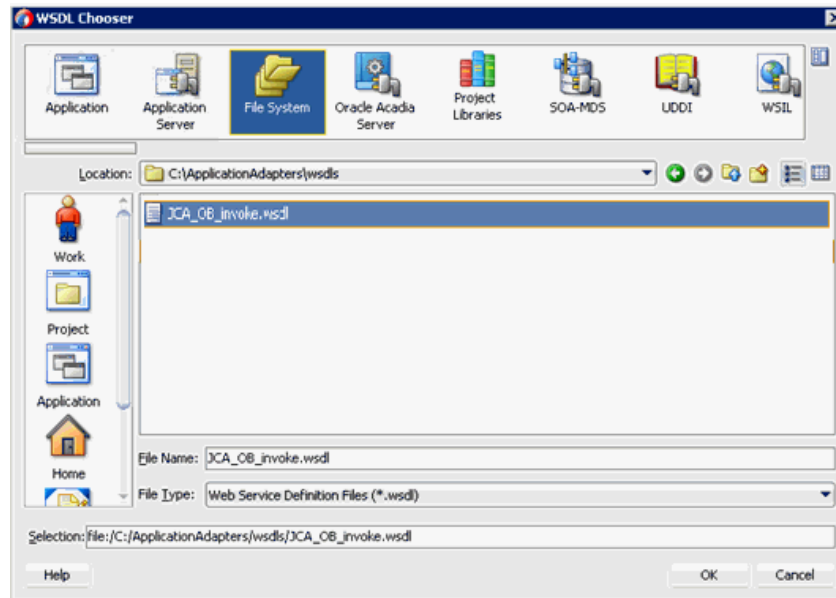
Figure 10–4 Create Third Party Adapter Service Pane



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.

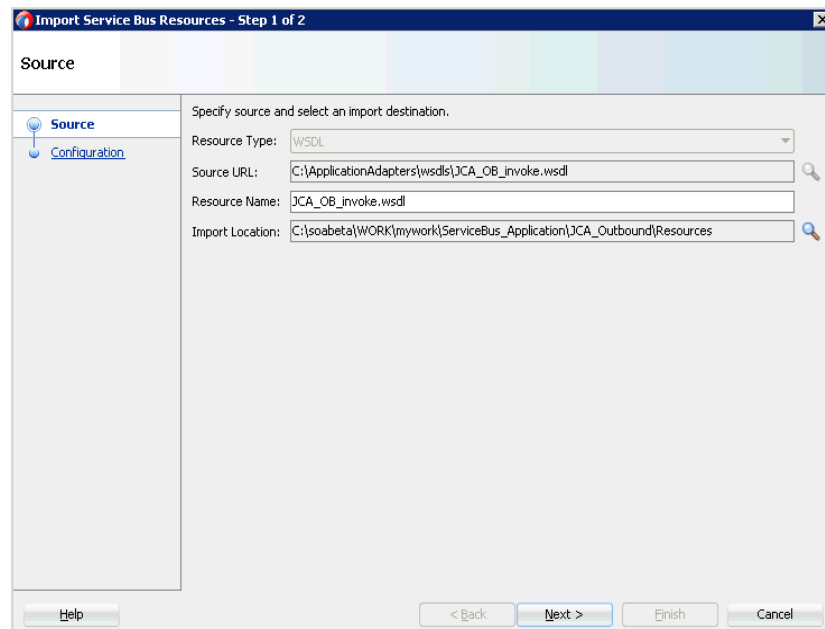
The WSDL Chooser dialog is displayed, as shown in [Figure 10–5](#).

Figure 10–5 WSDL Chooser Dialog



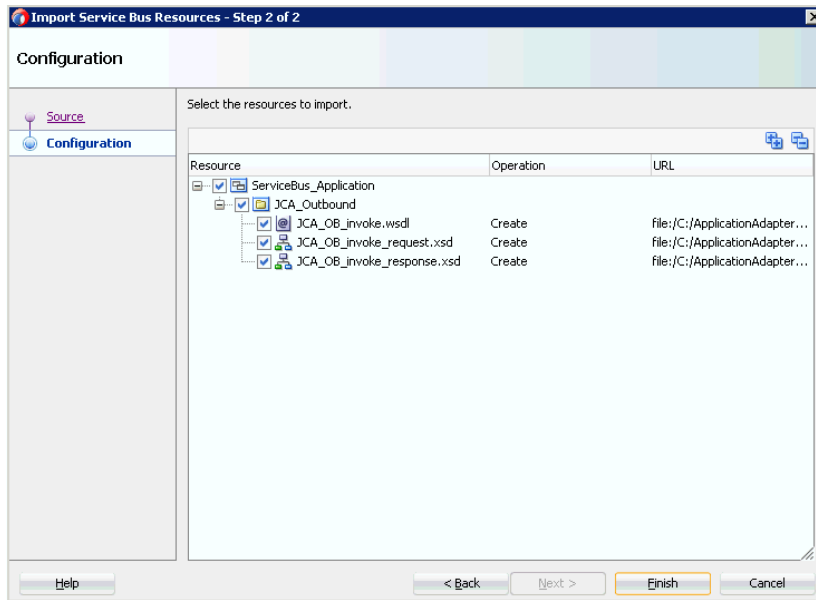
5. Select the **File System** tab, then browse, and select an outbound WSDL file from the WSDL directory.
6. Click **OK**.
The Import Service Bus Resources dialog is displayed.
7. Click **Next**, as shown in [Figure 10–6](#).

Figure 10–6 Source Pane



8. In the Configuration pane, click **Finish**, as shown in [Figure 10–7](#).

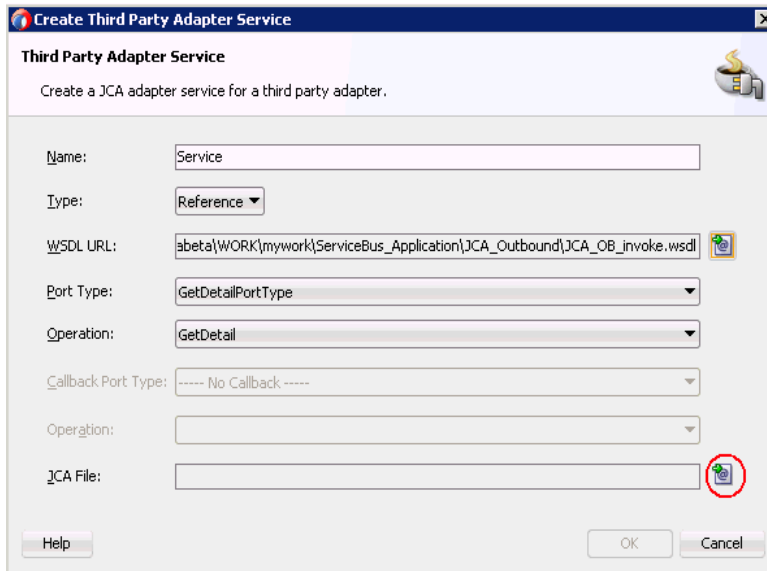
Figure 10–7 Configuration Pane



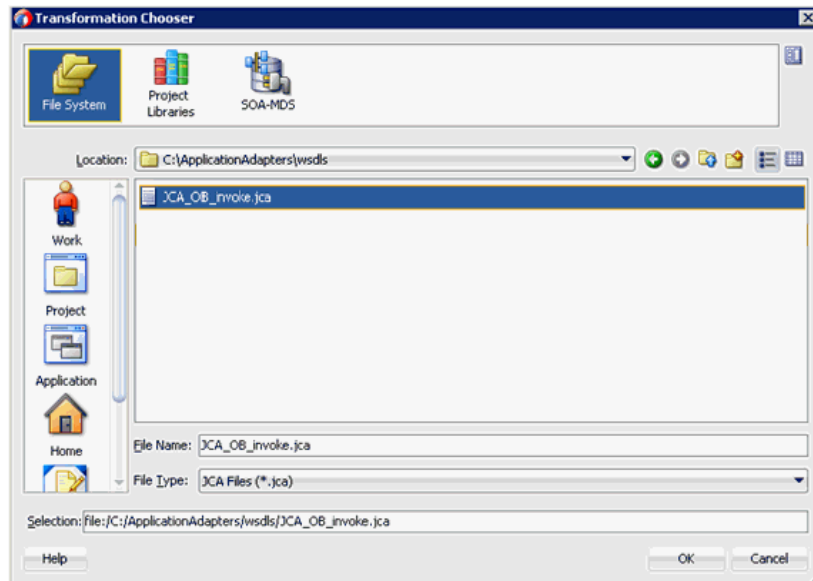
You are returned to the Create Third Party Adapter Service Dialog.

9. Click the Find JCA file icon which is located to the right of the JCA File field, as shown in [Figure 10–8](#).

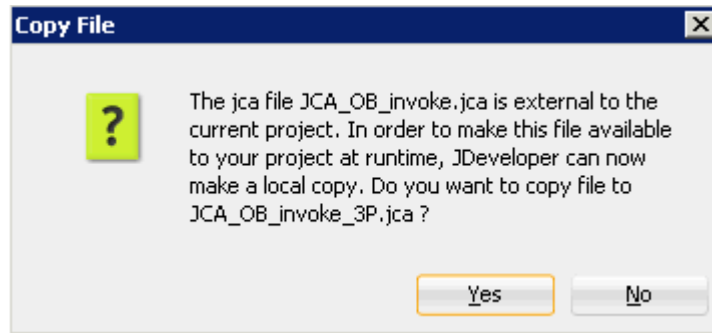
Figure 10–8 Find JCA File Icon



The Transformation Chooser dialog is displayed, as shown in [Figure 10–9](#).

Figure 10–9 Transformation Chooser Dialog

10. Select the JCA properties file from the WSDL directory.
11. Click **OK**. The Copy File message is displayed, as shown in [Figure 10–10](#).

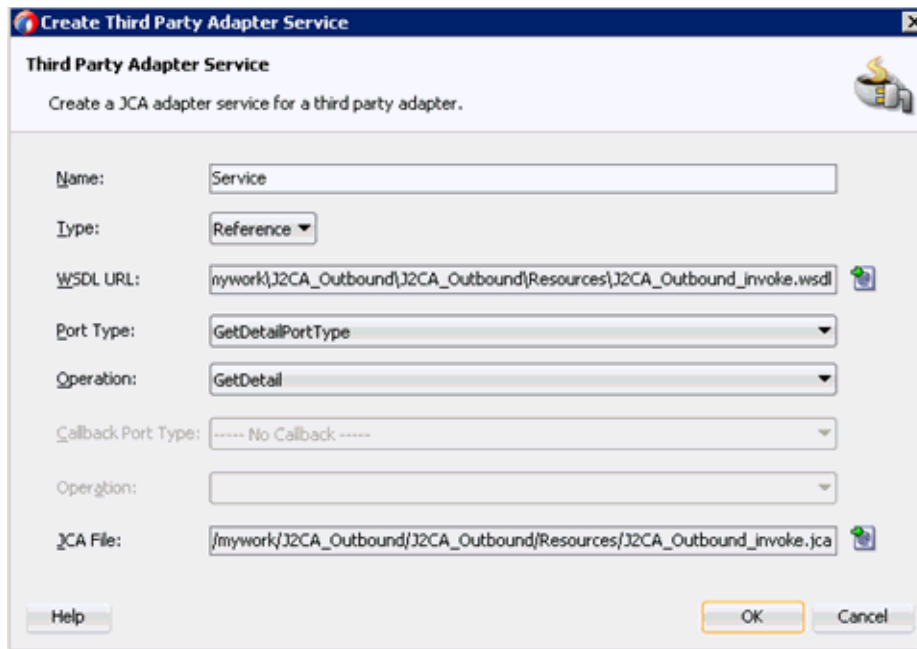
Figure 10–10 Copy File Message

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 10–11](#).

Figure 10–11 Create Third Party Adapter Service Dialog



13. Click **OK**.

The Business service component is created in the External Services pane.

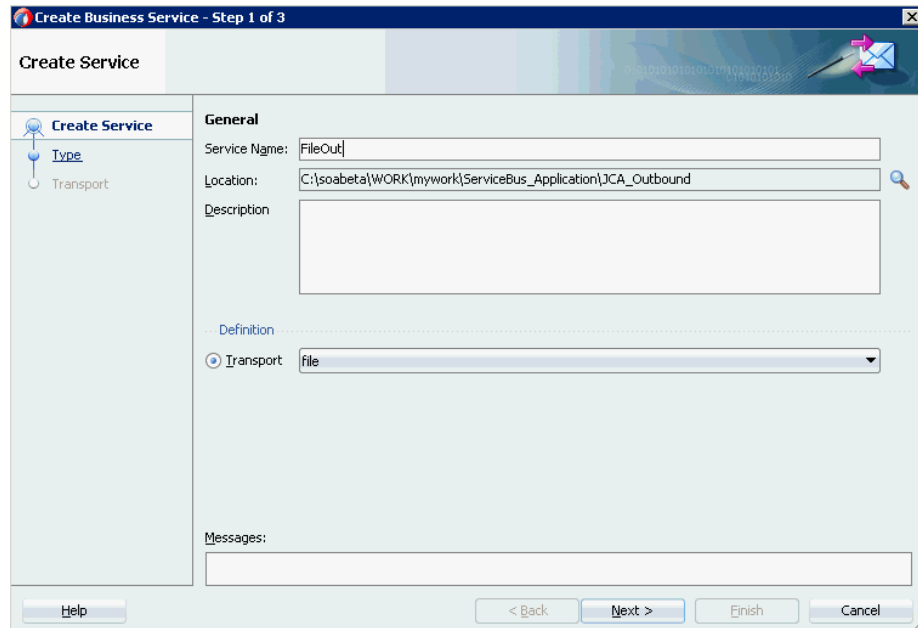
10.1.2.2 Configuring a File Transport Type Business Service

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.

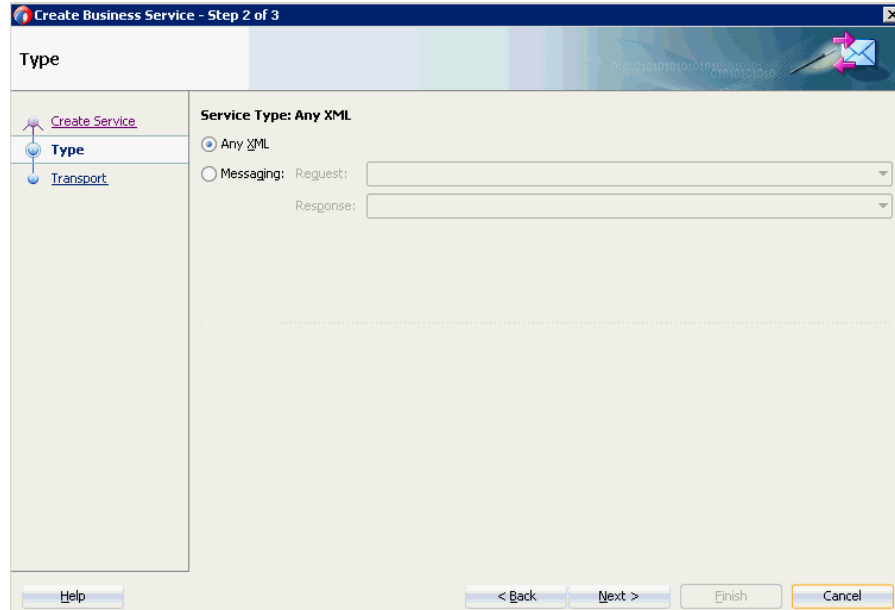
The Create Business Service dialog is displayed.

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 10–12](#).

Figure 10–12 Create Service Pane

The Type pane is displayed. The **Any XML** option is selected by default.

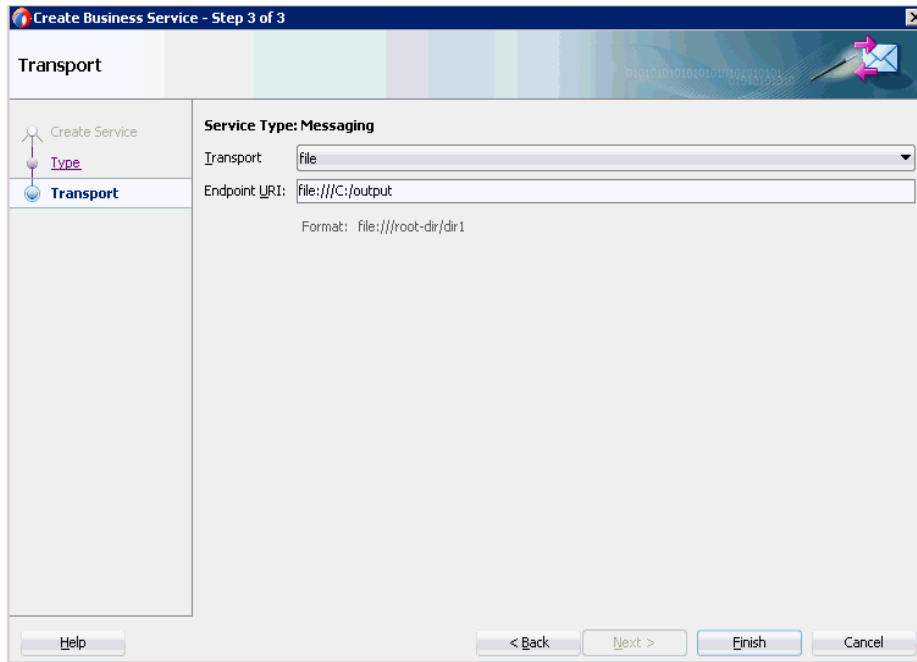
3. Click **Next**, as shown in [Figure 10–13](#).

Figure 10–13 Type Pane

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 10–14](#).

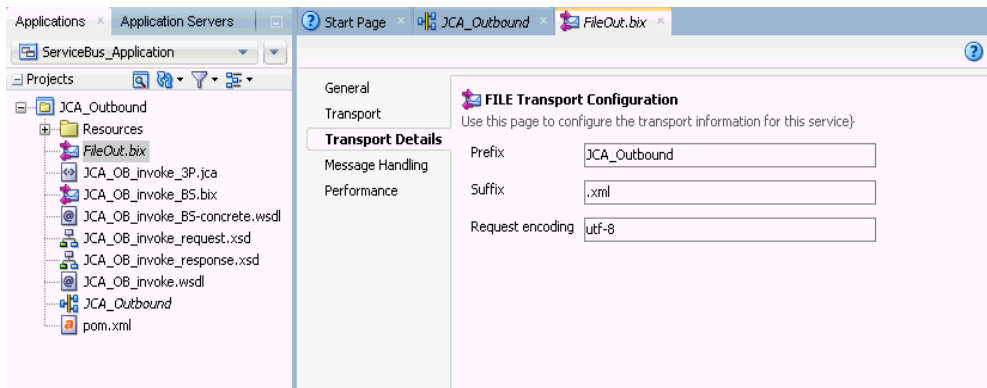
Figure 10–14 Transport Pane



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 10–15](#).

Figure 10–15 Transport Details

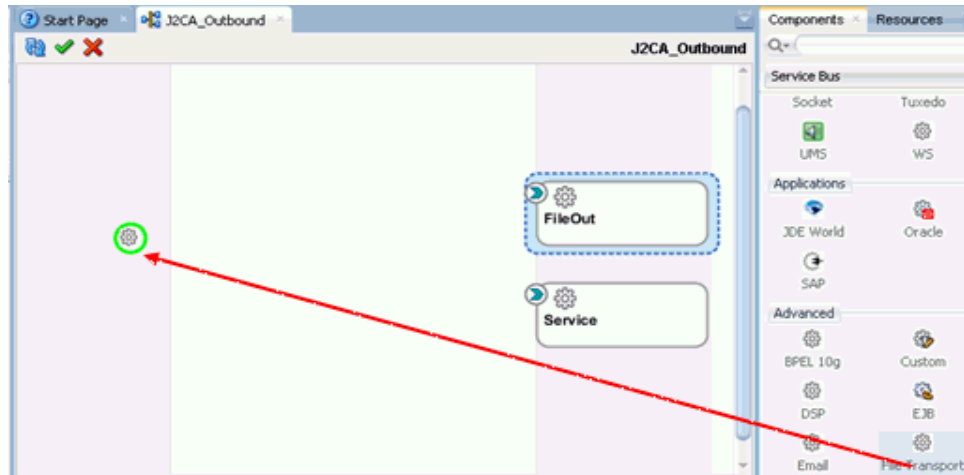


10.1.2.3 Creating a Proxy Service With Pipeline

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 10–16](#).

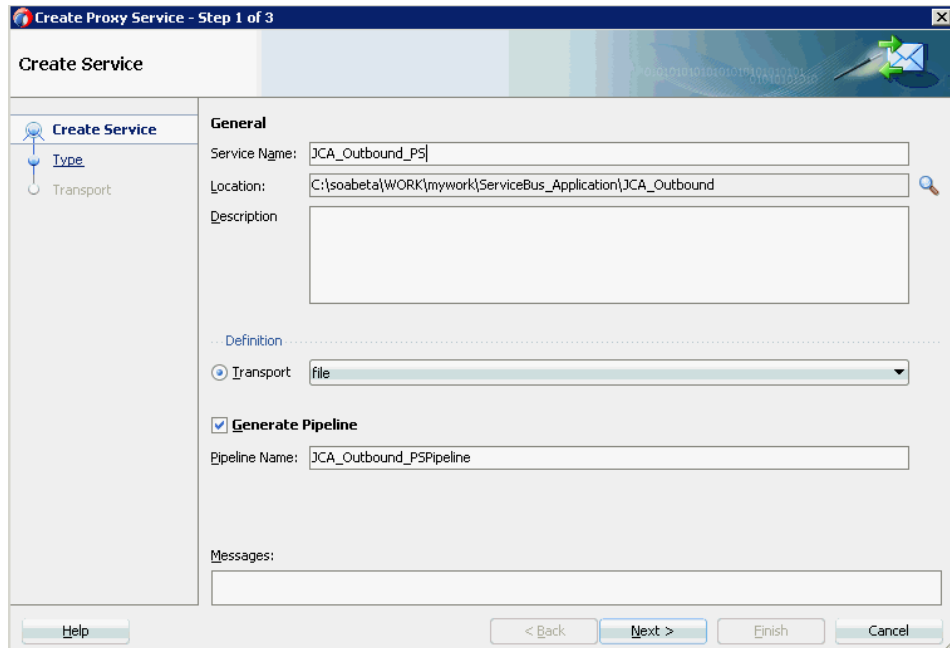
Figure 10–16 File Transport Component



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 10–17](#).

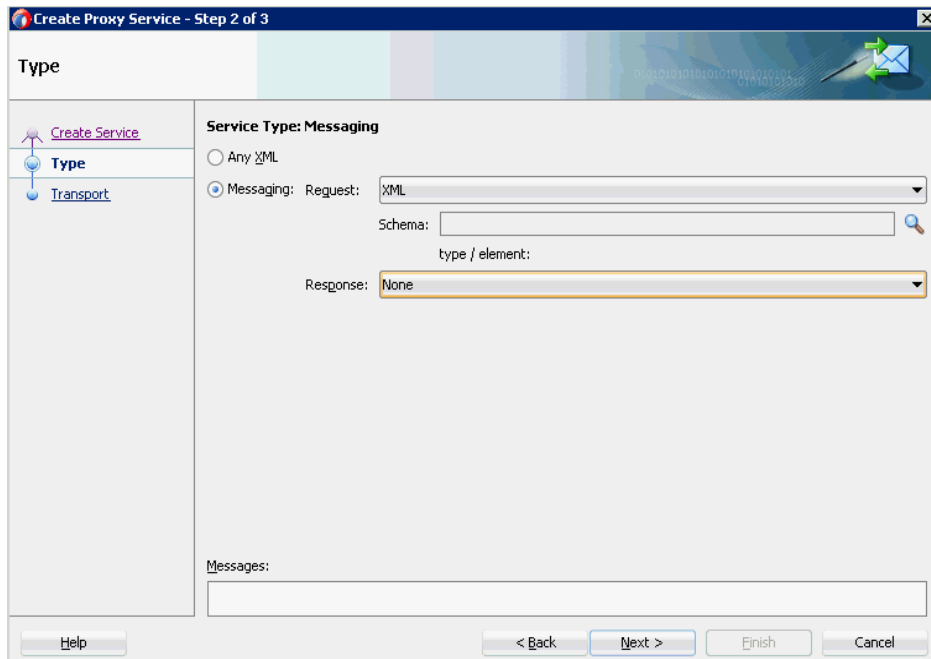
Figure 10–17 Create Service Pane



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 10–18](#).

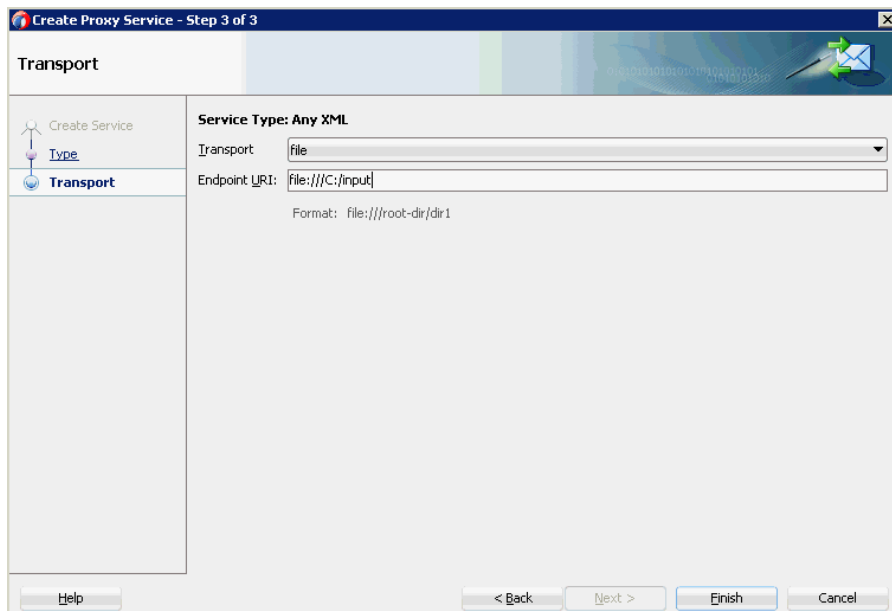
Figure 10–18 Type Pane



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 10–19](#).

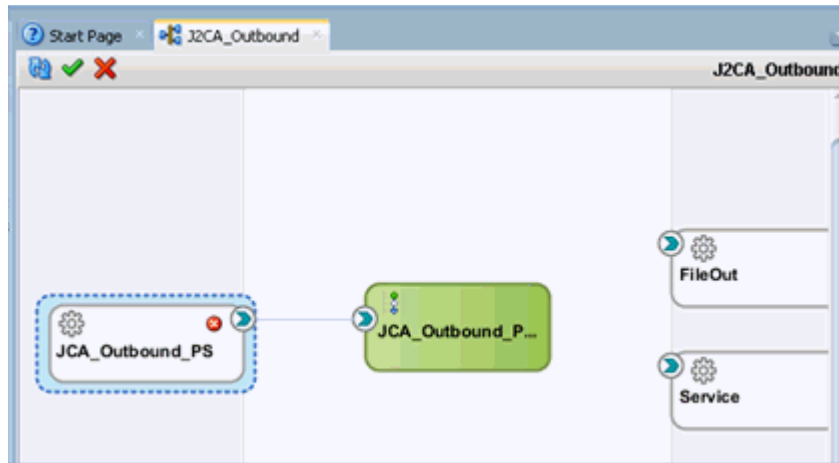
Figure 10–19 Transport Window



The Proxy service along with the pipeline is created and displayed.

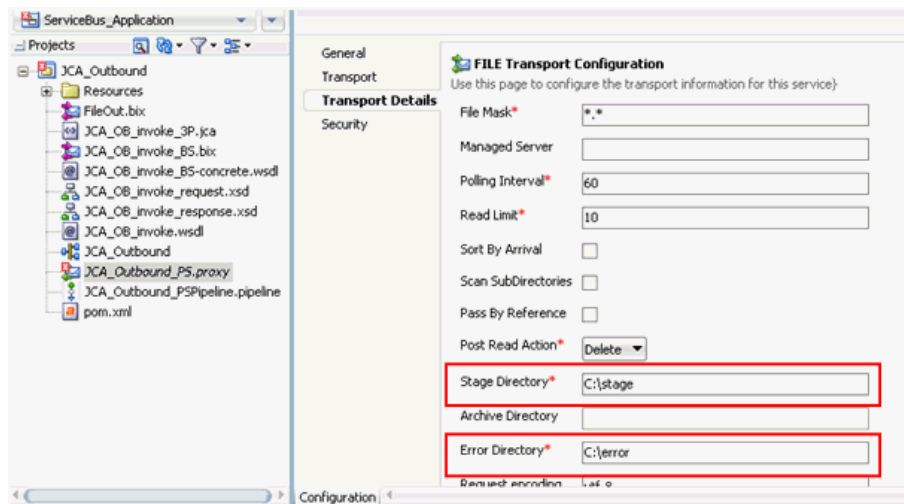
6. Double-click the created Proxy Service (for example: JCA_Outbound_PS), as shown in [Figure 10–20](#).

Figure 10–20 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 10–21](#).

Figure 10–21 File Transport Configuration



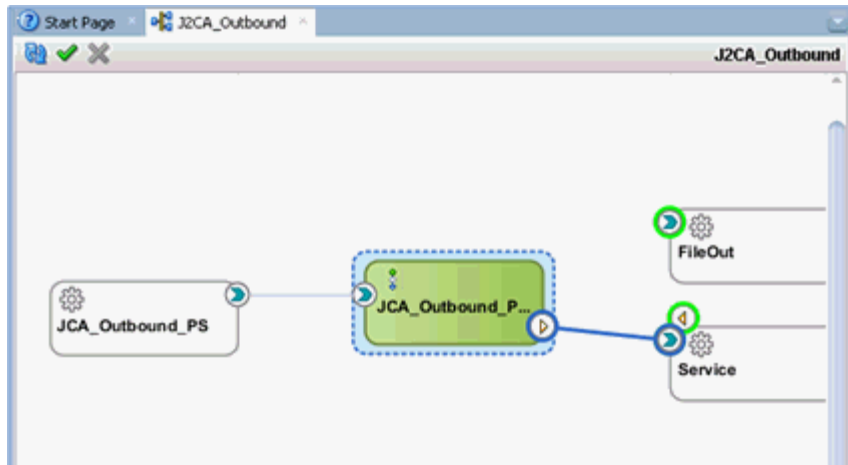
8. Save and close the Proxy Service configuration page.

10.1.2.4 Configuring the Routing Rules

Perform the following steps to configure the routing rules:

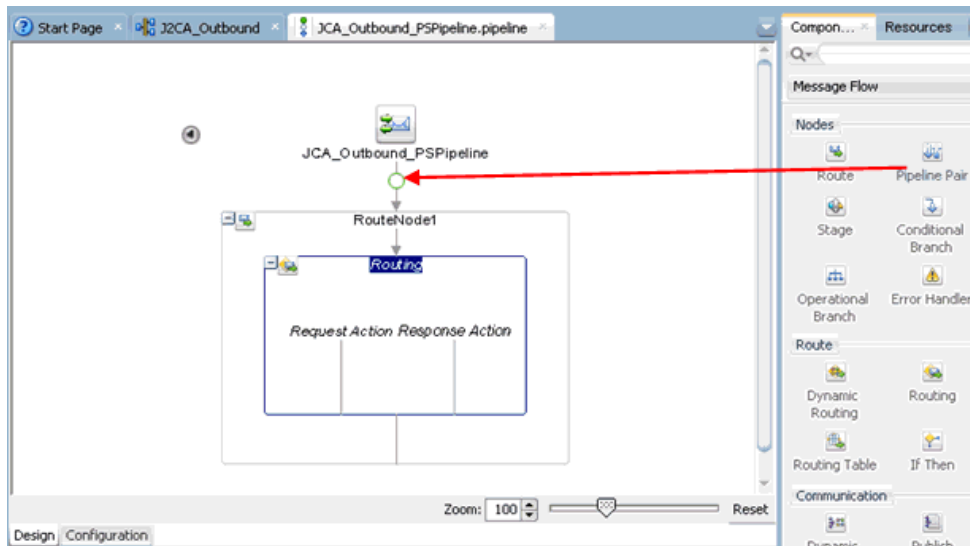
1. Connect the Pipeline to the Business Service (for example, Service) as shown in [Figure 10–22](#).

Figure 10–22 Business Service Pipeline



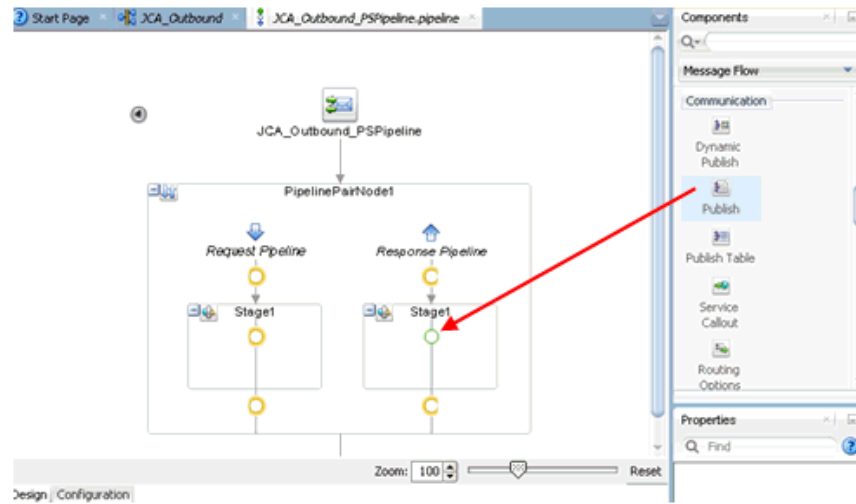
2. Double-click on the pipeline (for example, JCA_Outbound_PSPipeline) in the Pipelines/Split Joins pane.
The Pipeline configuration page is displayed.
3. Drag and drop the **Pipeline Pair** node from Nodes pane to the area below the Pipeline (for example: JCA_Outbound_PSPipeline), as shown in [Figure 10–23](#).

Figure 10–23 Pipeline Pair Node



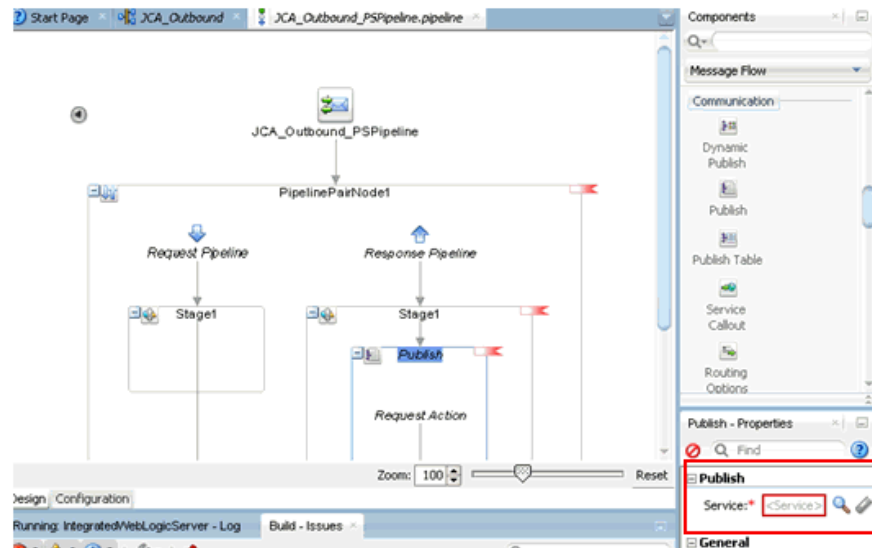
4. Drag and drop the **Publish** node from the Communication pane to the area beneath Stage1 of the Response Pipeline, as shown in [Figure 10–24](#).

Figure 10–24 Publish Node



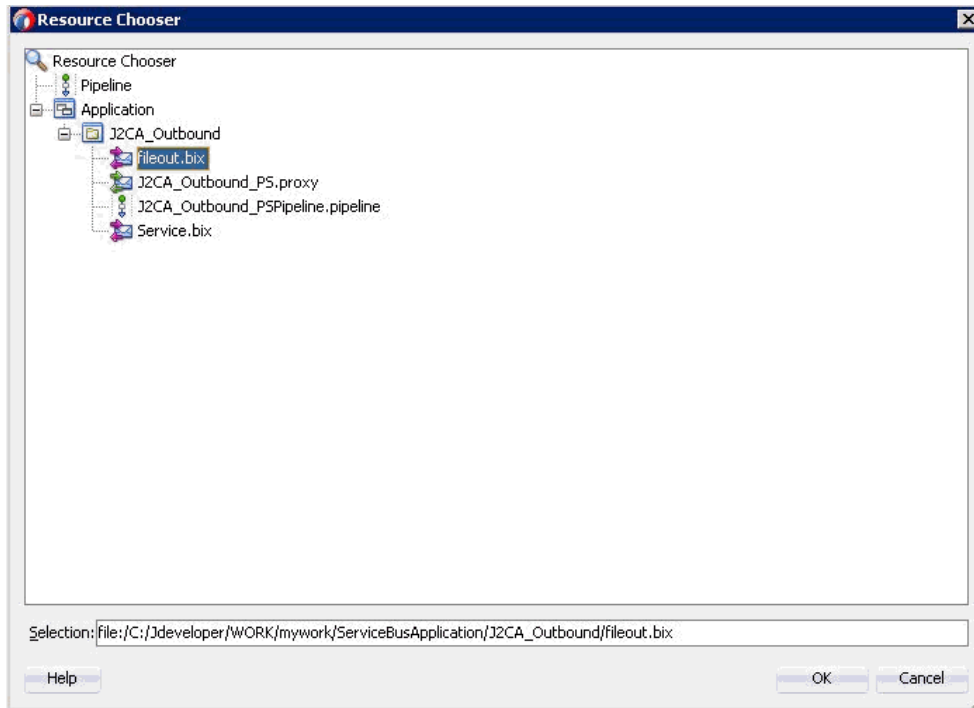
5. Click on the browse icon to the right of the Service field in the right pane of Publish Properties, as shown in [Figure 10–25](#).

Figure 10–25 Browse Icon



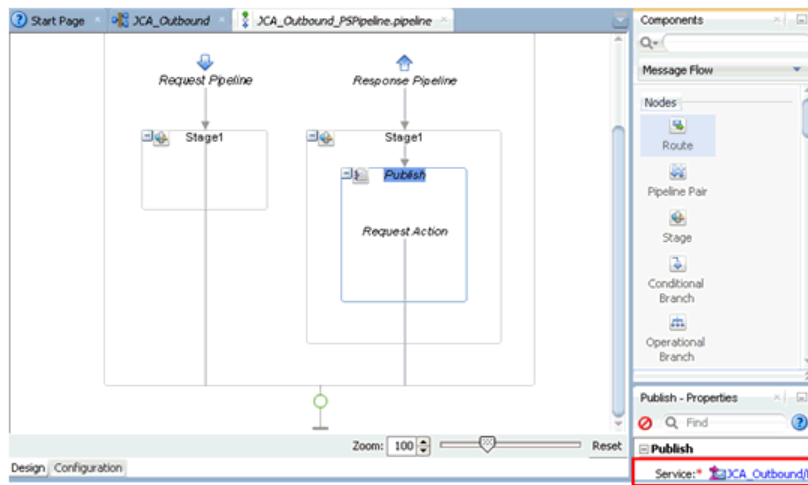
6. In the displayed Resource Chooser window, select the **Fileout.bix** File Transport Business service and click **OK**, as shown in [Figure 10–26](#).

Figure 10–26 Resource Chooser



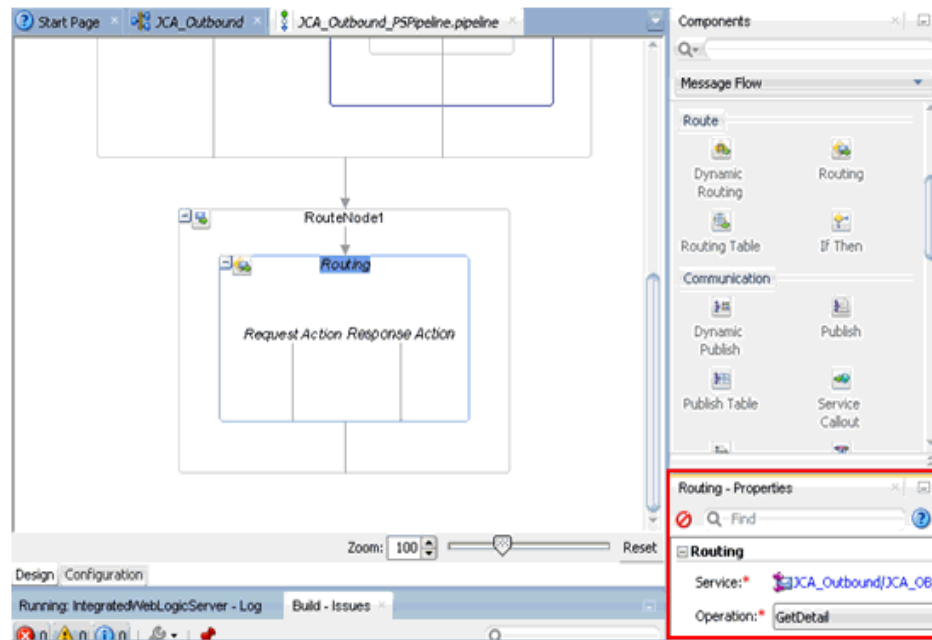
In the right pane, the selected service is configured in the Publish pane, as shown in Figure 10–27.

Figure 10–27 Publish Pane



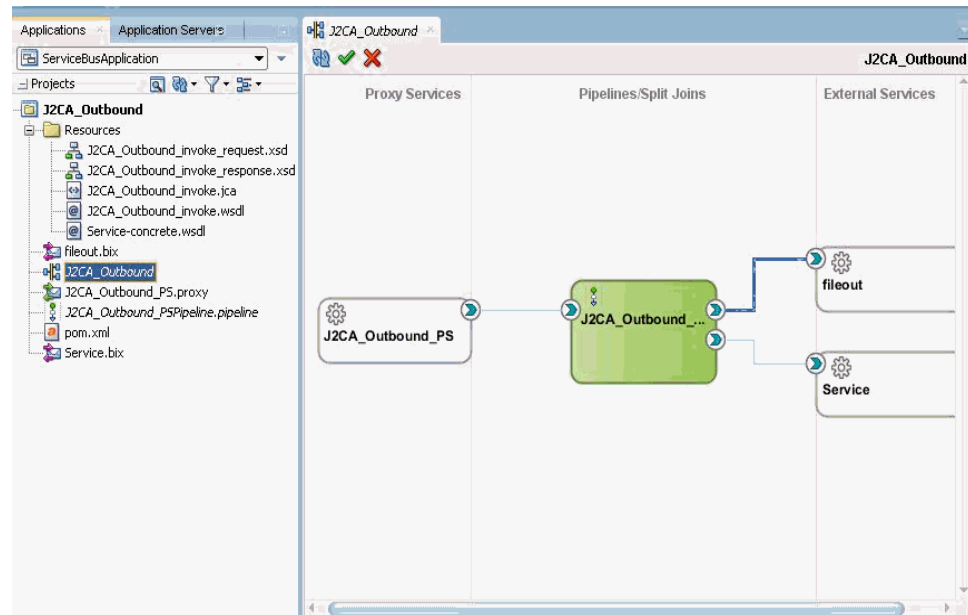
7. Click on the Routing to verify the Service is selected properly, as shown in Figure 10–28.

Figure 10–28 Pipeline Configuration



8. Save and Close the Pipeline configuration page.
9. Double-click the overview.xml file (for example: J2CA_Outbound), and click **Save All** in the menu bar to save the OSB process, as shown in [Figure 10–29](#).

Figure 10–29 Save All Icon

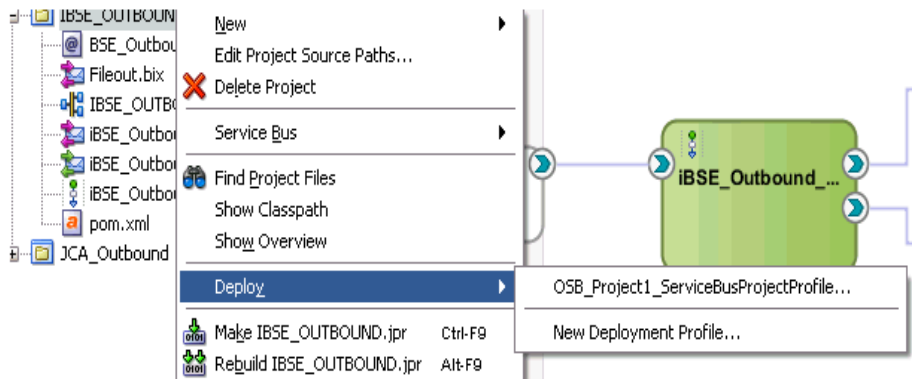


10.1.3 Deploying the OSB Outbound Process

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 10–30](#).

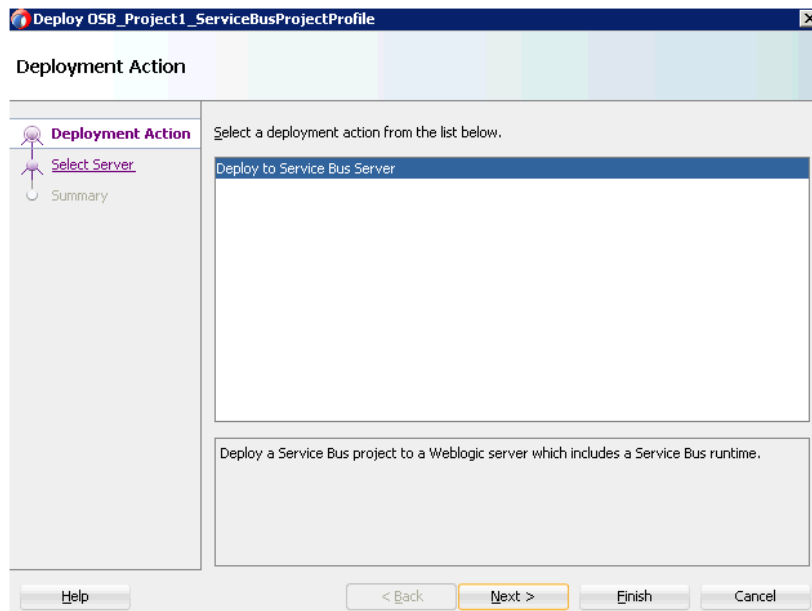
Figure 10–30 Deploy Option



The Deployment Action page is displayed.

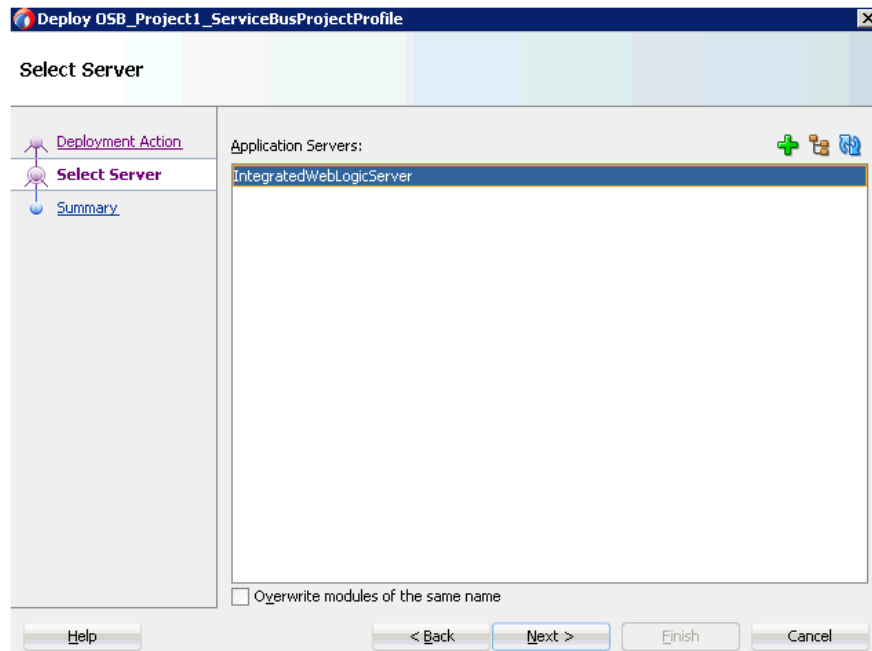
2. Click **Next**, as shown in [Figure 10–31](#).

Figure 10–31 Deployment Action Page

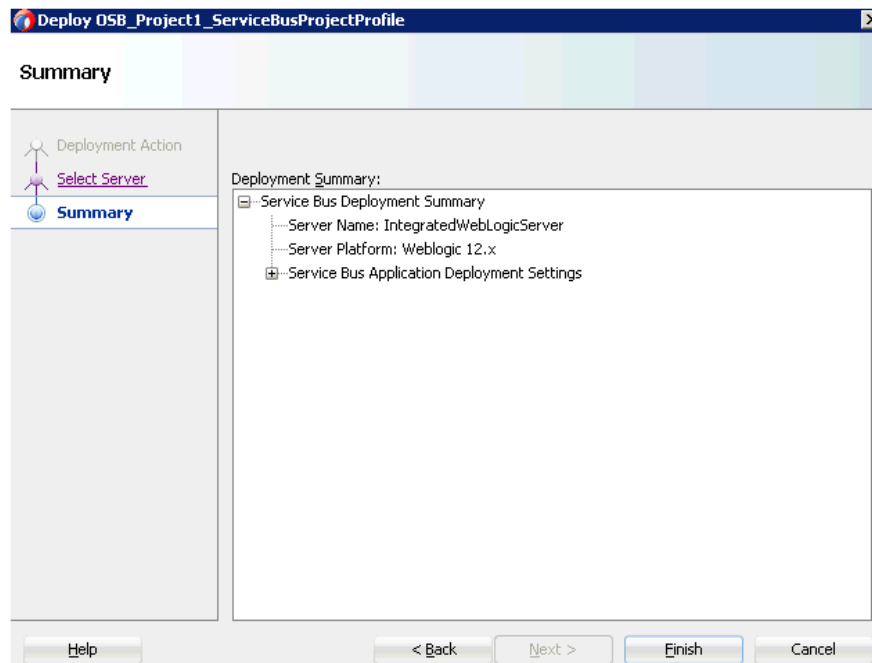


The Select Server page is displayed.

3. Select an available application server that was configured and click **Next**, as shown in [Figure 10–32](#).

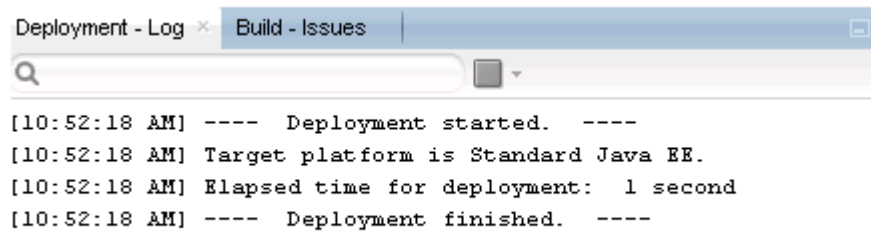
Figure 10–32 *Select Server Page*

The Summary page is displayed, as shown in [Figure 10–33](#).

Figure 10–33 *Summary Page*

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

The process is deployed successfully, as shown in [Figure 10–34](#).

Figure 10–34 Successful Deployment Message

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).

10.2 Configuring an OSB Inbound Process Using JDeveloper (J2CA Configuration)

This section describes how to configure an OSB inbound process to your SAP R/3 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\SAP_Samples.zip\SAP_Samples\OSB_Jdeveloper\J2CA\SAP_Sample_J2CA_OSB_Inbound_Project
```

This section includes the following topics:

- [Section 10.2.1, "Creating a Service Bus Application for OSB"](#)
- [Section 10.2.2, "Defining an OSB Inbound Process"](#)
- [Section 10.2.3, "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 6.5.1, "Generating WSDL for Event Integration"](#) on page 6-37.

10.2.1 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 10.1.1, "Creating a Service Bus Application for OSB"](#) on page 10-2.

10.2.2 Defining an OSB Inbound Process

This section describes how to define an OSB inbound process. The following topics are included:

- [Section 10.2.2.1, "Configuring a Third-Party Adapter Service Component"](#)
- [Section 10.2.2.2, "Creating a Pipeline"](#)
- [Section 10.2.2.3, "Configuring a File Transport Type Business Service"](#)

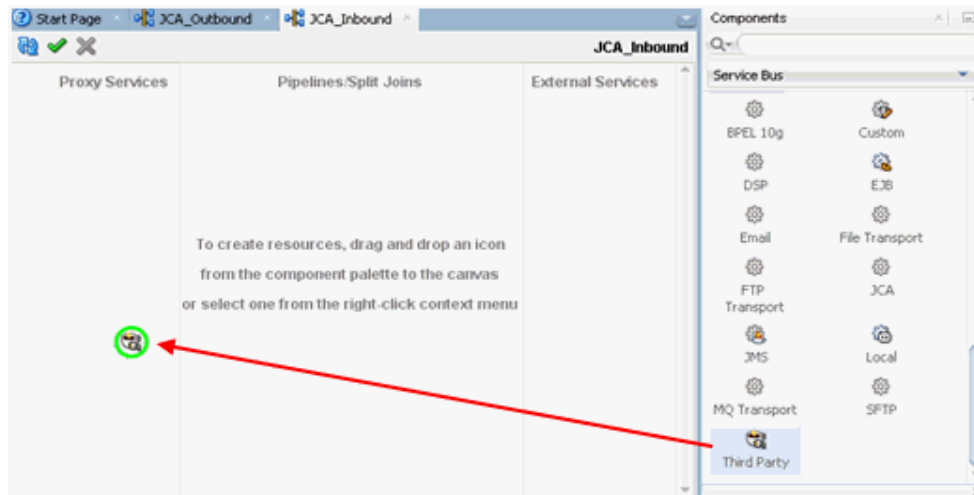
- [Section 10.2.2.4, "Configuring the Routing Rules"](#)

10.2.2.1 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 10–35](#).

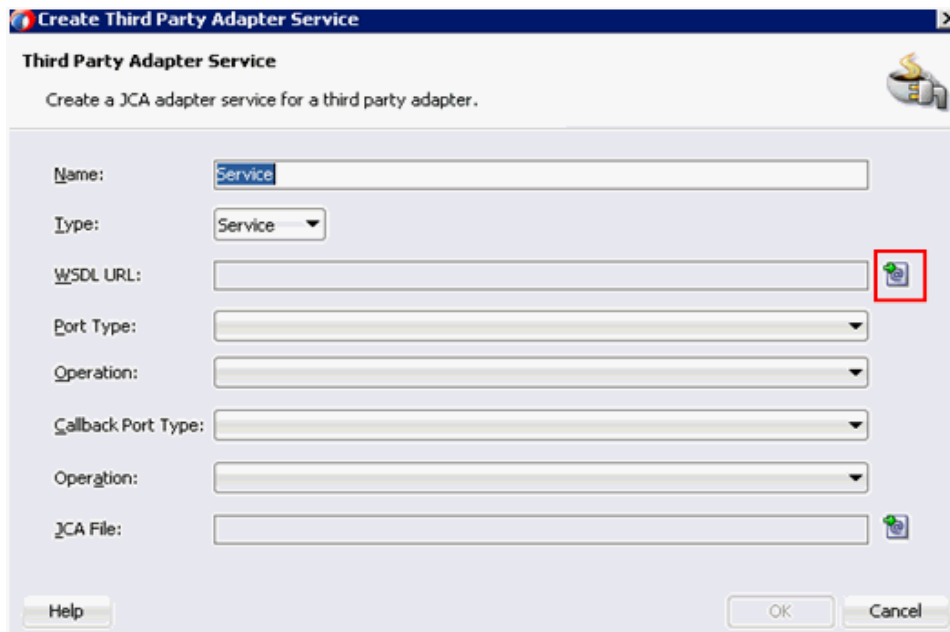
Figure 10–35 Third Party Adapter Service Component



The Create Third Party Adapter Service dialog is displayed.

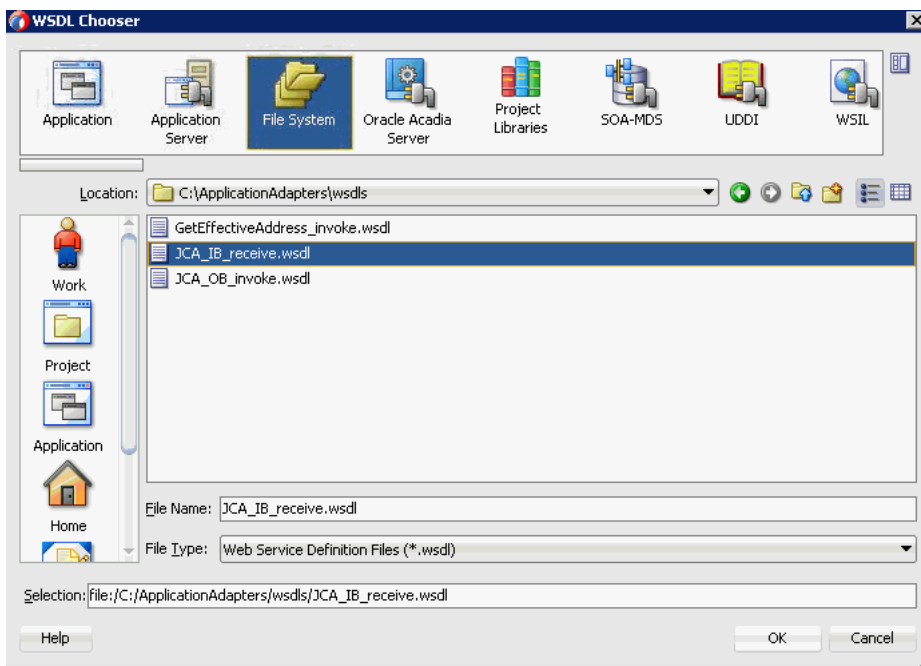
2. Enter any name you wish for the Third Party Adapter Service (for example, CC_GetDetail).
3. Ensure that **Service** is selected from the Type drop-down list (by default).
4. Click the Find existing WSDLs icon, which is located to the right of the WSDL URL field, as shown in [Figure 10–36](#).

Figure 10–36 Third Party Adapter Service Dialog



The WSDL Chooser dialog is displayed, as shown in [Figure 10–37](#).

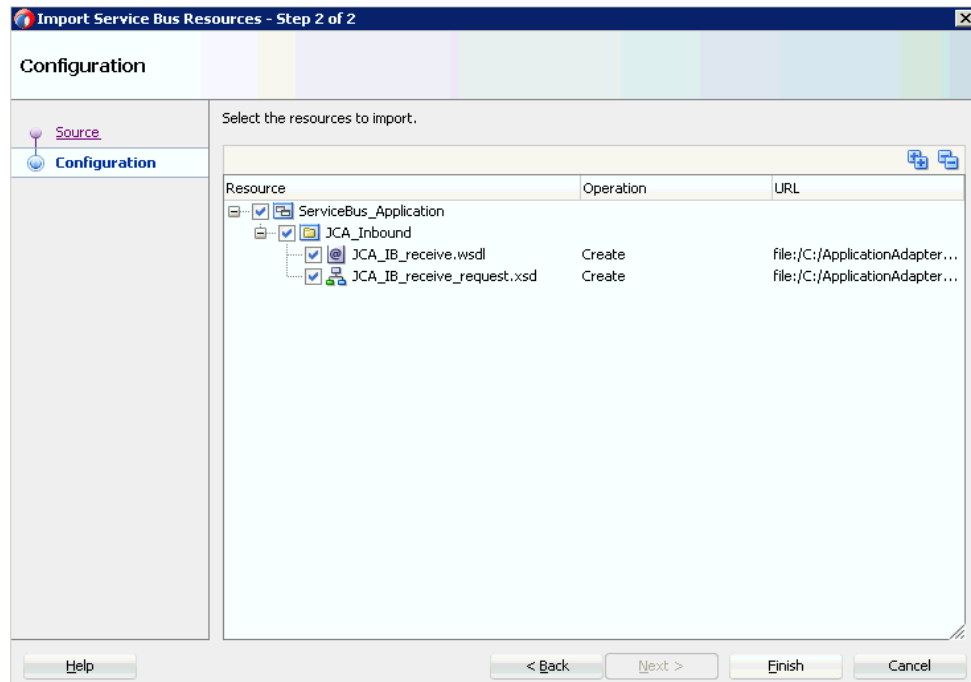
Figure 10–37 WSDL Chooser Dialog



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**.

- In the Configuration window, click **Finish**, as shown in [Figure 10-38](#).

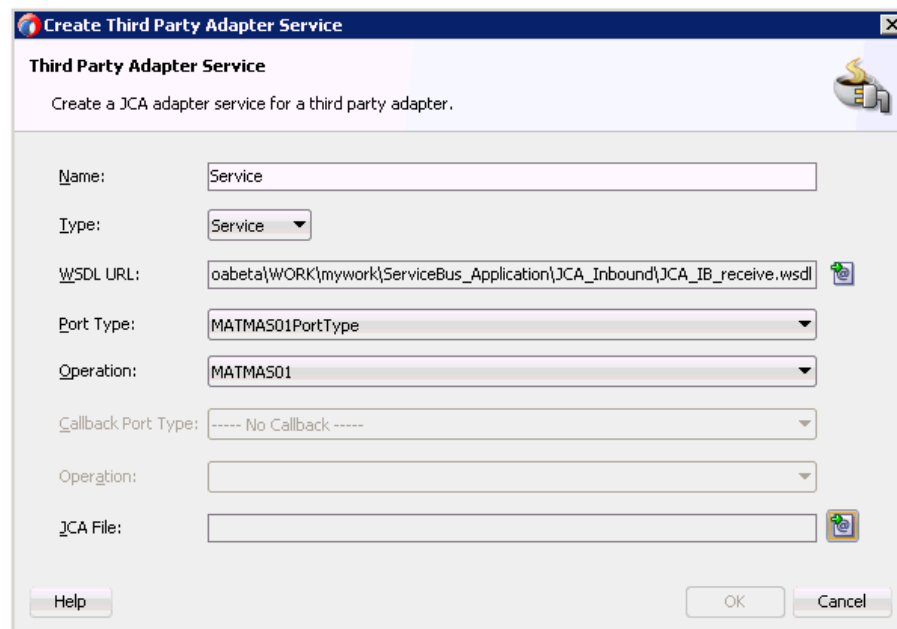
Figure 10-38 Configuration Window



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

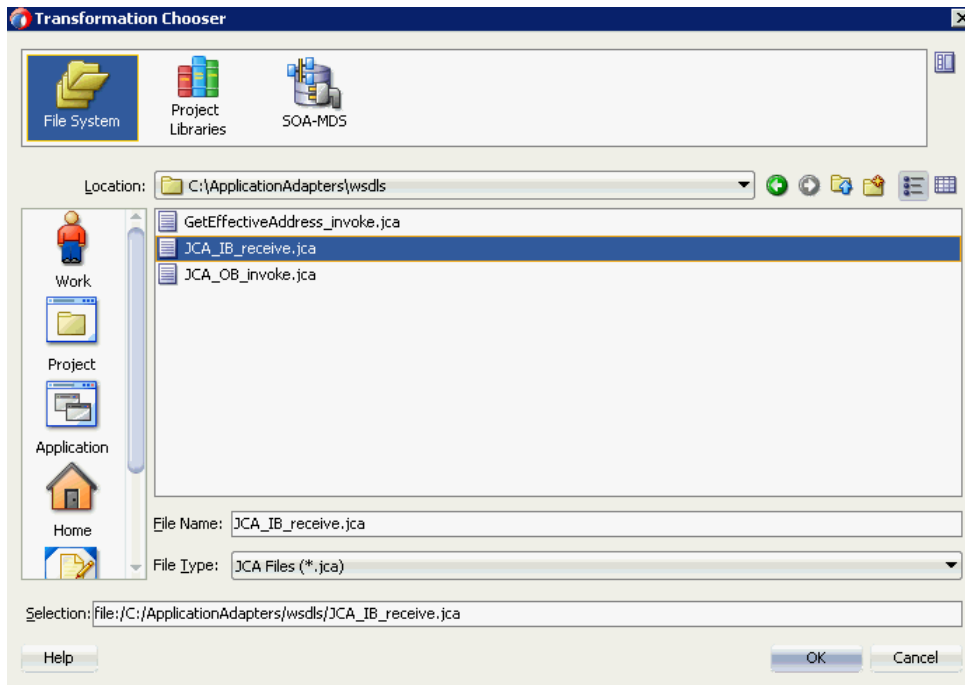
- Click the Find JCA file icon, which is located to the right of the JCA File field, as shown in [Figure 10-39](#).

Figure 10-39 Create Third Party Adapter Service Dialog



The Transformation Chooser dialog is displayed, as shown in [Figure 10-40](#).

Figure 10–40 Transformation Chooser Dialog

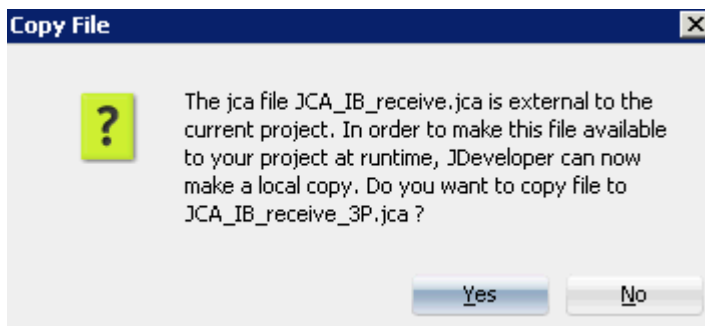


10. Select the JCA properties file from the WSDL directory.

11. Click OK.

The Copy File message is displayed, as shown in [Figure 10–41](#).

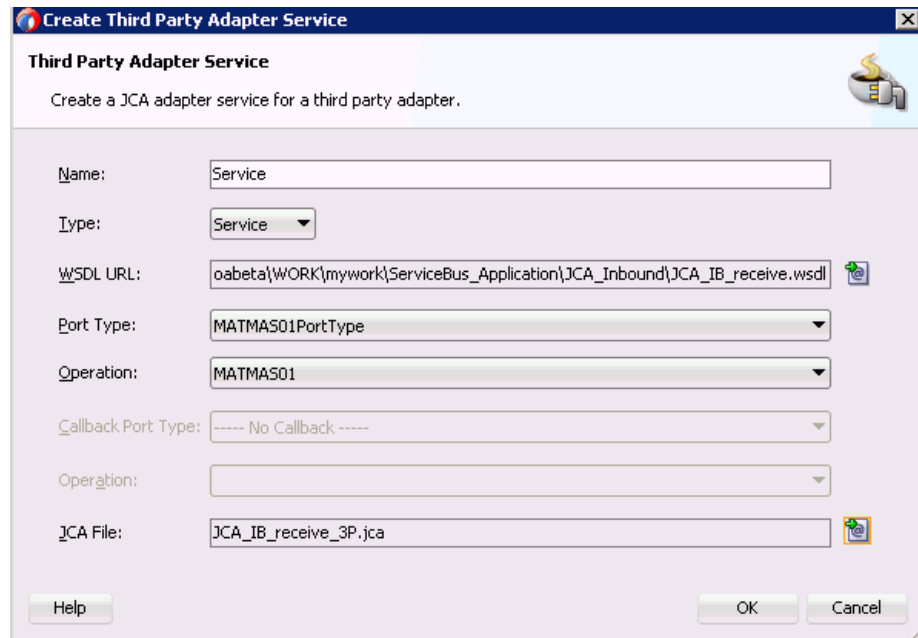
Figure 10–41 Copy File Message



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 10–42](#).

Figure 10–42 Create Third Party Adapter Service Dialog**13. Click OK.**

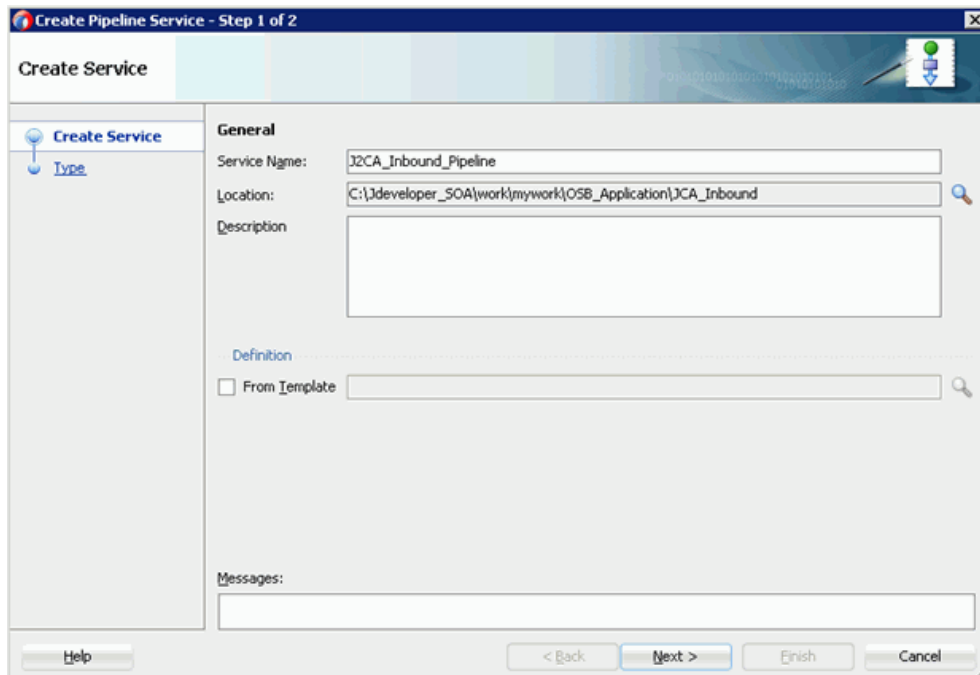
The third party adapter service component is created in the Proxy Services pane.

10.2.2.2 Creating a Pipeline

Perform the following steps to generate inbound proxy service with Pipeline:

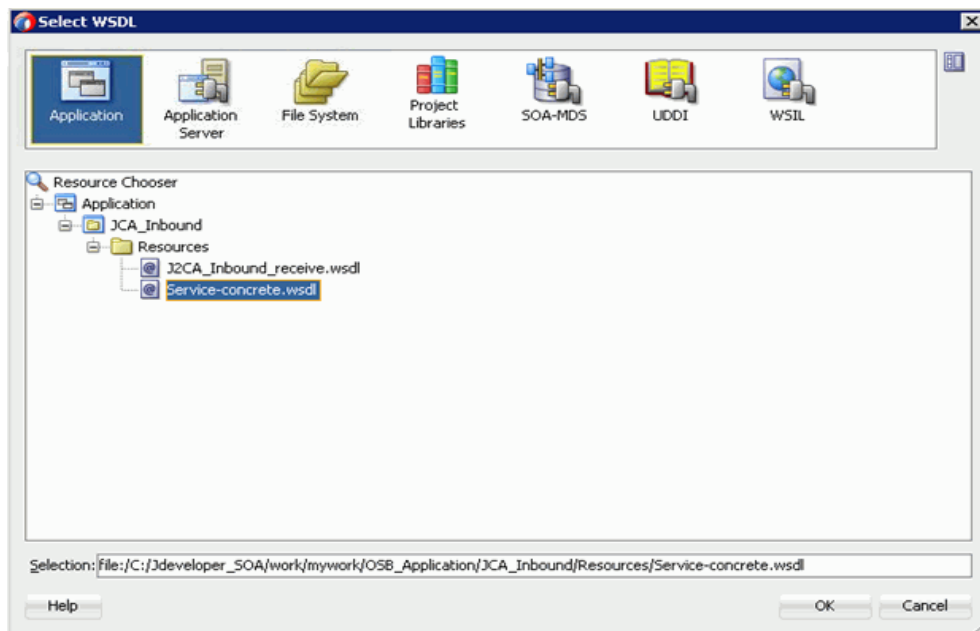
1. Under Service Bus, click **Resources**.
2. Drag and drop the Pipeline to the Pipelines/Split Joins pane.
3. Provide a name for the Pipeline and click next, as shown in [Figure 10–43](#).

Figure 10-43 Create Service Page



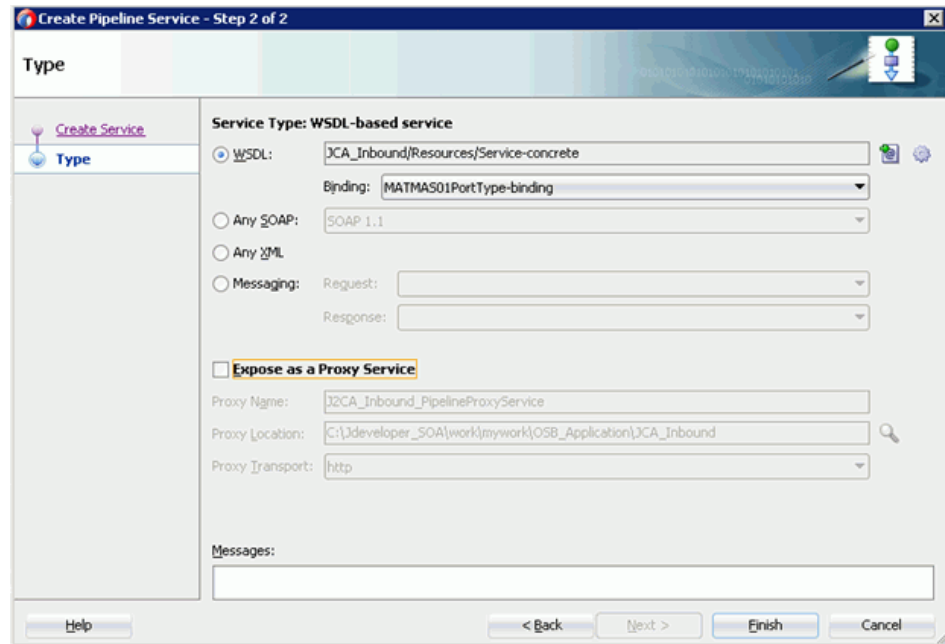
4. In the Create Pipeline Service window, select the **WSDL** option and click on the WSDL URL.
5. Select **Application** in the WSDL chooser window, then select **service-concrete.wsdl** in the appropriate OSB project, and then click **OK**, as shown in [Figure 10-44](#).

Figure 10-44 Select WSDL Page



6. Clear the **Expose as a Proxy Service** check box and click **Finish**, as shown in [Figure 10-45](#).

Figure 10–45 Type Page



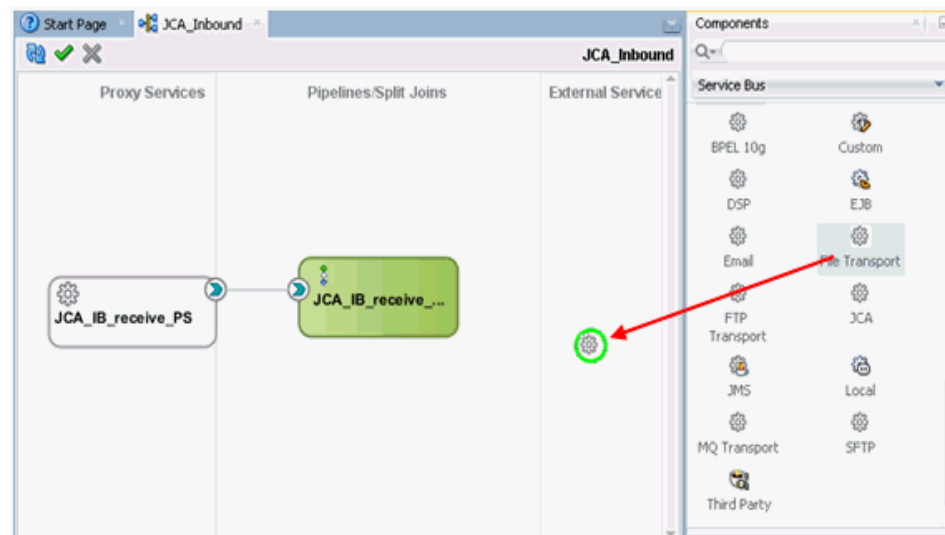
7. Drag and drop the Proxy Service to the Pipelines/Split Joins pane.

10.2.2.3 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 10–46.

Figure 10–46 File Transport Node



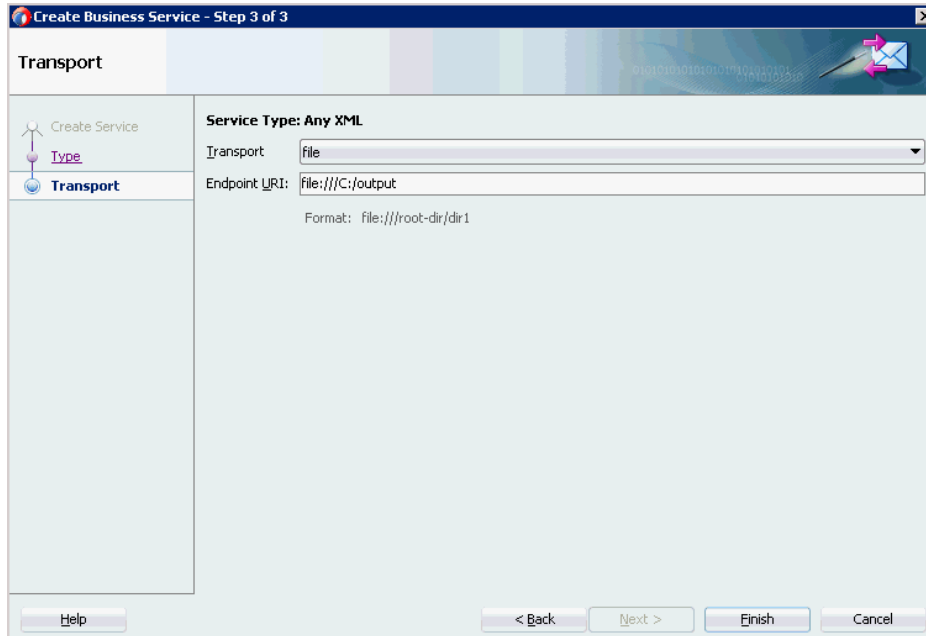
The Create Business Service dialog is displayed.

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 10-47](#).

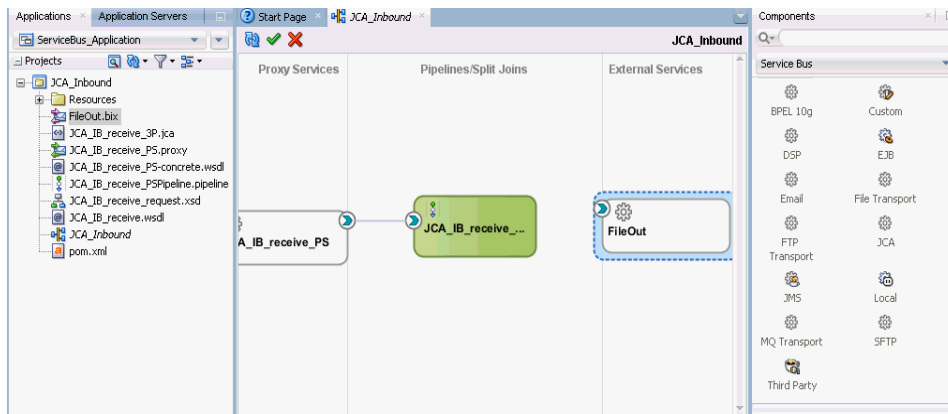
Figure 10-47 Transport Pane



The FileOut Business service is created.

5. Double-click the FileOut Business service, as shown in [Figure 10-48](#).

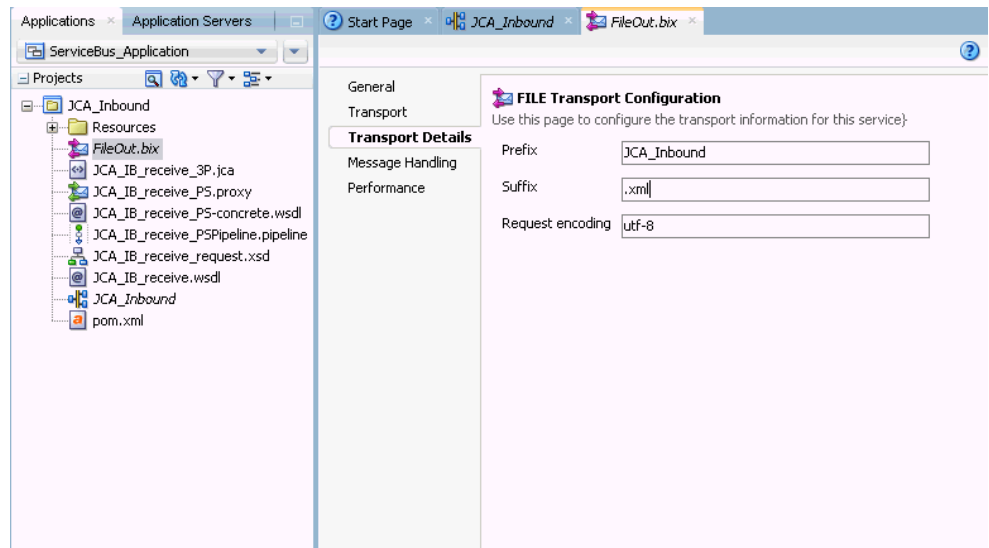
Figure 10-48 FileOut Business Service



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 10-49](#).

Figure 10–49 File Transport Configuration



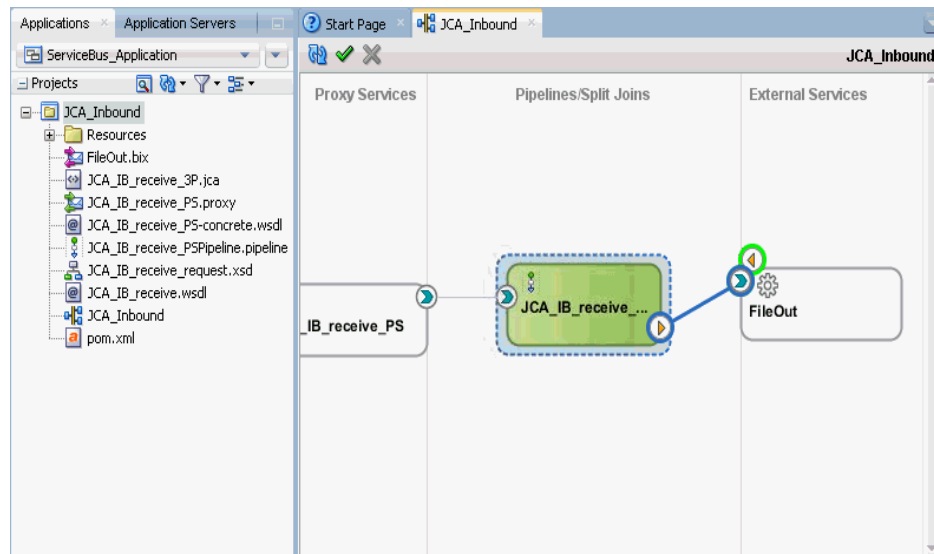
7. Save and close the Configuration page.

10.2.2.4 Configuring the Routing Rules

Perform the following steps to configure the routing rules.

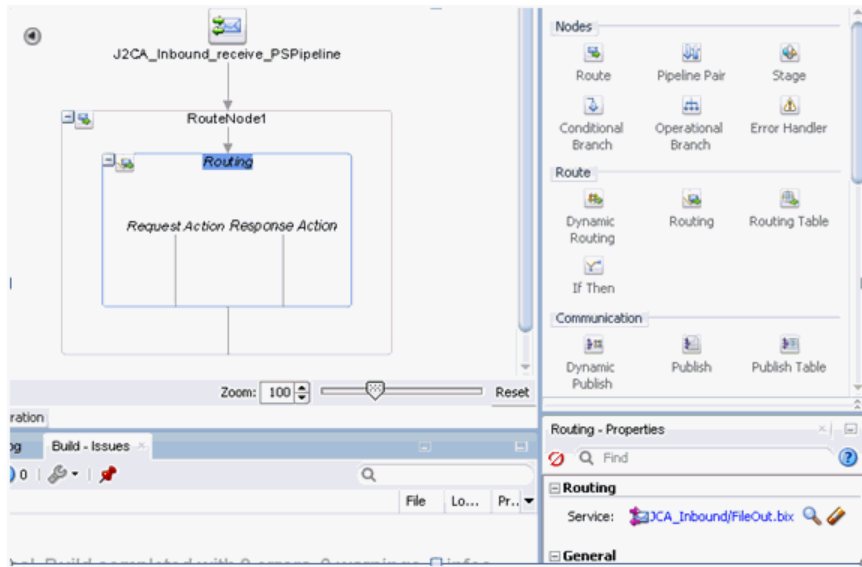
1. Create a connection between the Pipeline (for example, JCA_IB_receive_PSPipeline) and the File Type Business Service (for example, FileOut), as shown in Figure 10–50.

Figure 10–50 Mapping Proxy and FileOut



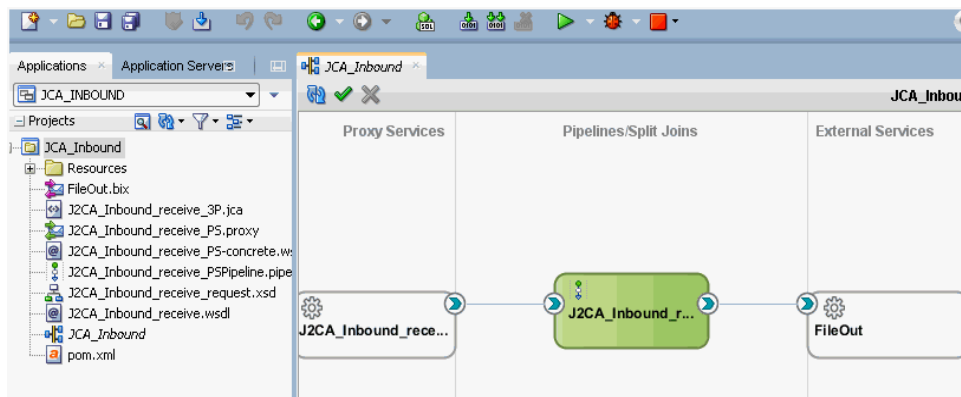
2. Double-click the Pipeline (for example, JCA_IB_receive_PSPipeline).
3. Click the Routing pane and ensure that the File Type Business Service (for example, FileOut) is properly configured in the Service field, as shown in Figure 10–51.

Figure 10–51 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 10–52](#).

Figure 10–52 Save All



10.2.3 Deploying the OSB Inbound Process

To deploy the created OSB inbound process, see steps 1 - 4 in [Section 10.1.3, "Deploying the OSB Outbound Process"](#) on page 10-17.

Once the OSB inbound process is deployed successfully, trigger an event from the SAP R/3 system and check if the output is received in the configured output location (for example, C:\output).

10.3 Configuring an OSB Outbound Process Using JDeveloper (BSE Configuration)

This section describes how to configure an OSB outbound process to your SAP R/3 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\SAP_Samples.zip\SAP_Samples\OSB_Jdeveloper\BSE\SAP_Sample_BSE_OSB_Outbound_Project
```

This section includes the following topics:

- [Section 10.3.1, "Creating a Service Bus Application for OSB"](#)
- [Section 10.3.2, "Defining an OSB Outbound Process"](#)
- [Section 10.3.3, "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 6.7.1, "Generating a WSDL File for Request and Response Services Using a Web Service"](#) on page 6-63.

10.3.1 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 10.1.1, "Creating a Service Bus Application for OSB"](#) on page 10-2.

10.3.2 Defining an OSB Outbound Process

This section describes how to define an OSB outbound process. The following topics are included:

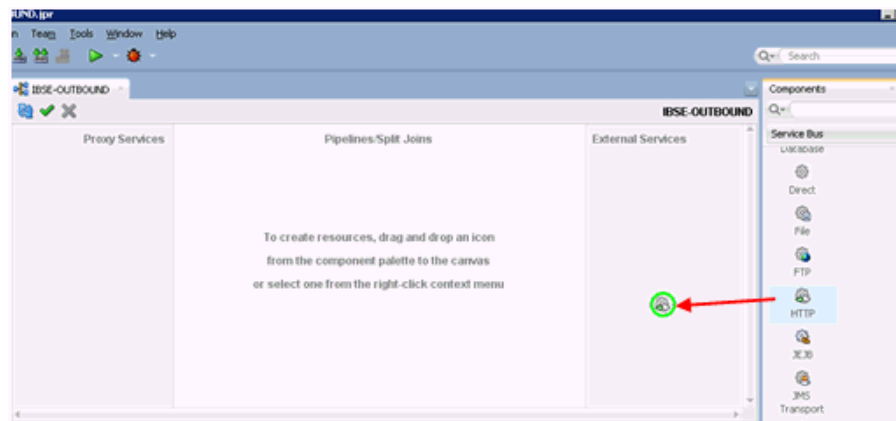
- [Section 10.3.2.1, "Configuring a WSDL-based Business Service"](#)
- [Section 10.3.2.2, "Creating a Proxy Service With Pipeline"](#)
- [Section 10.3.2.3, "Configuring a File Transport Type Business Service"](#)
- [Section 10.3.2.4, "Configuring the Routing Rules"](#)

10.3.2.1 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 10-53](#).

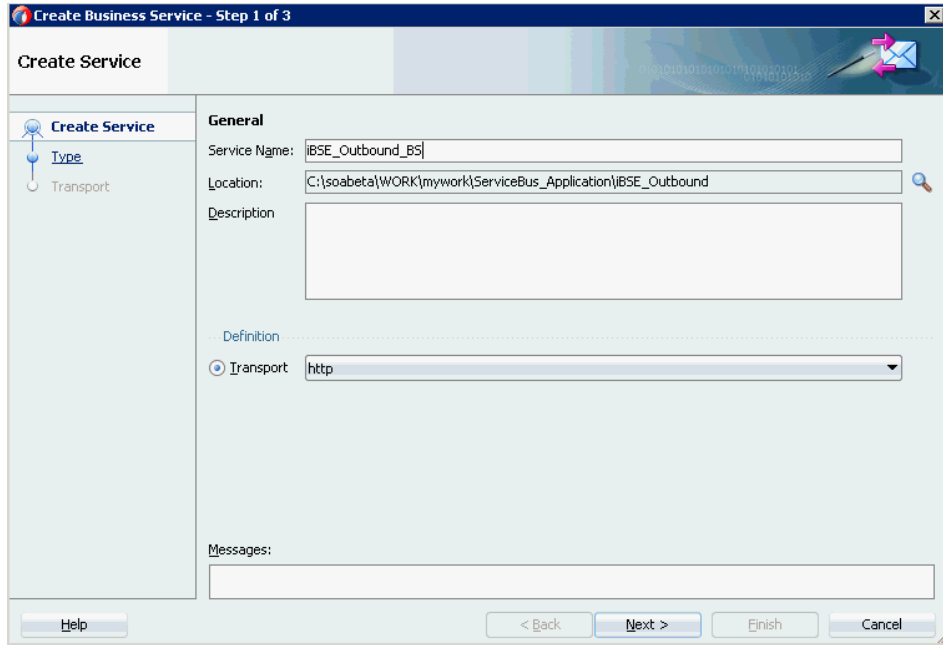
Figure 10-53 HTTP Component



The Create Business Service window is displayed.

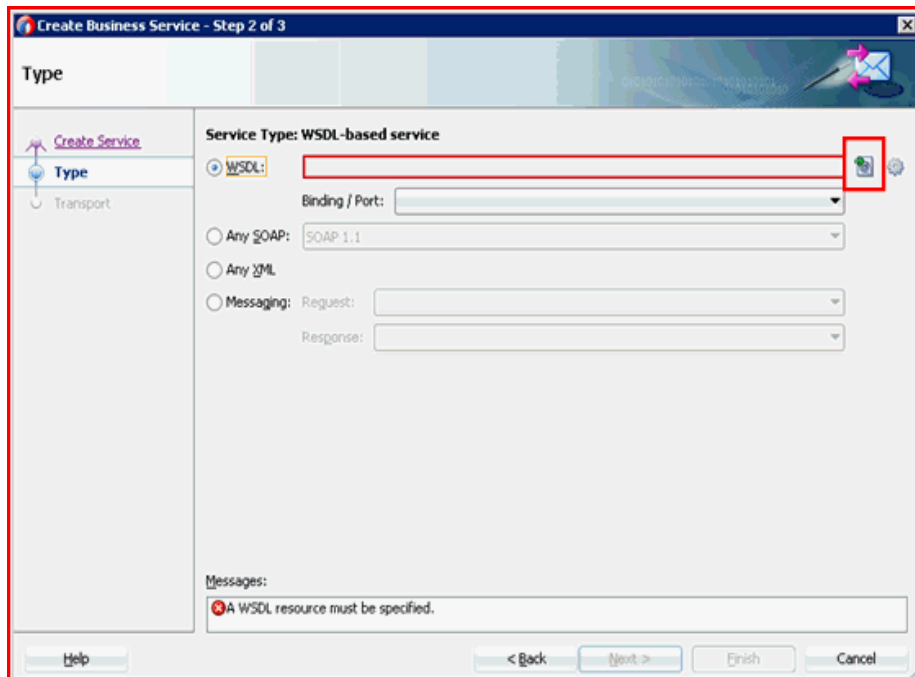
2. In the Service Name field, enter any name you wish for the Business Service and click **Next**, as shown in [Figure 10-54](#).

Figure 10-54 Create Business Service



3. In the displayed Service Type window, select the WSDL option and click the **Select WSDL** icon, as shown in [Figure 10-55](#).

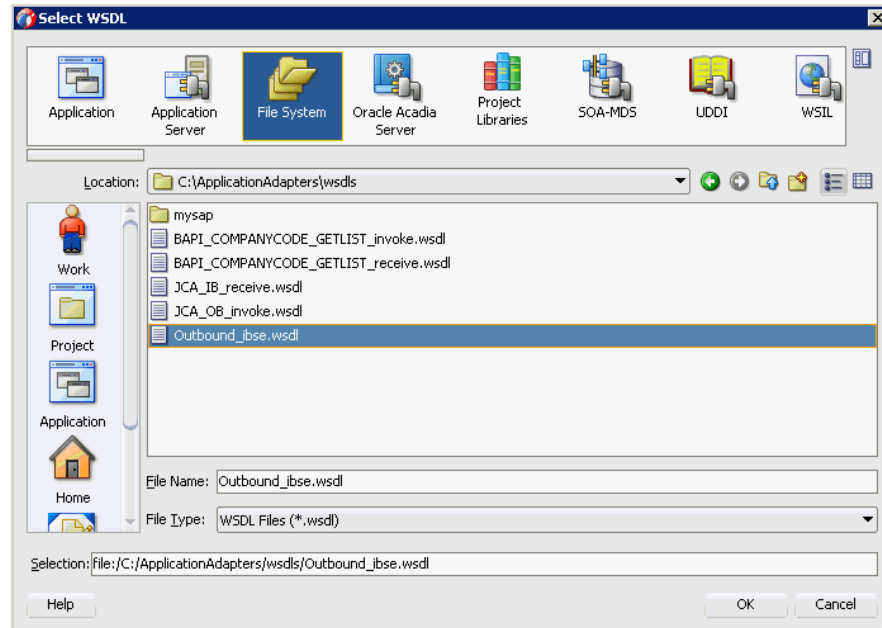
Figure 10-55 Type Pane



The Select WSDL window is displayed.

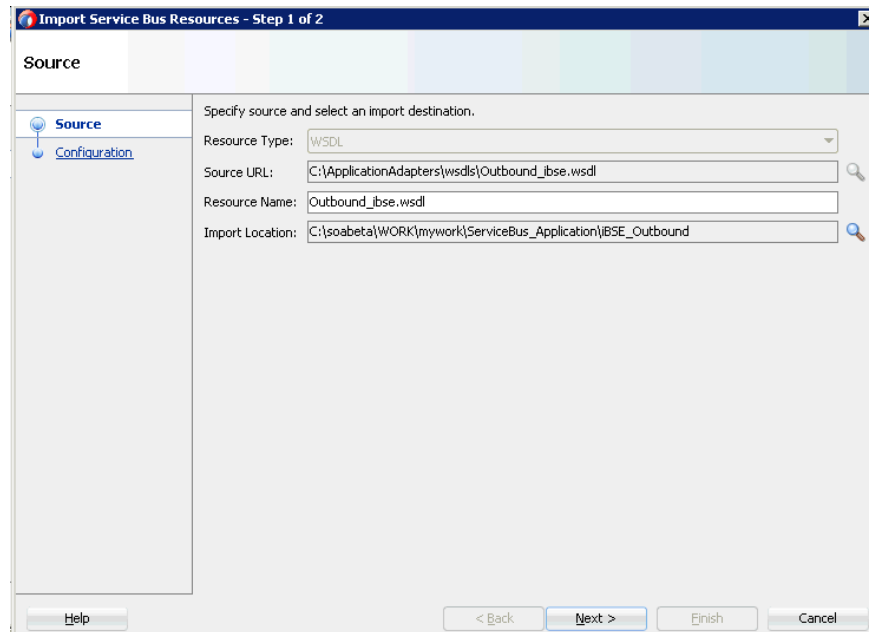
4. Select the File System folder icon, browse to the iBSE WSDL file and select it from the WSDL location, and then click **OK**, as shown in [Figure 10-56](#).

Figure 10-56 Select WSDL Window



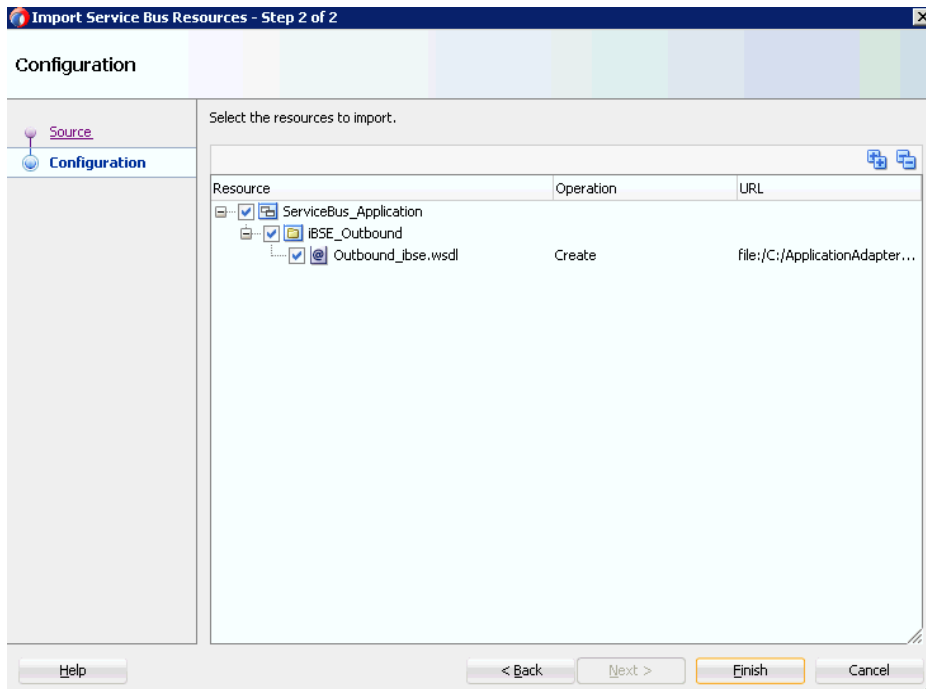
5. In the displayed Source pane, click **Next**, as shown in [Figure 10-57](#).

Figure 10-57 Source Pane



6. In the displayed Configuration pane, click **Finish**, as shown in [Figure 10-58](#).

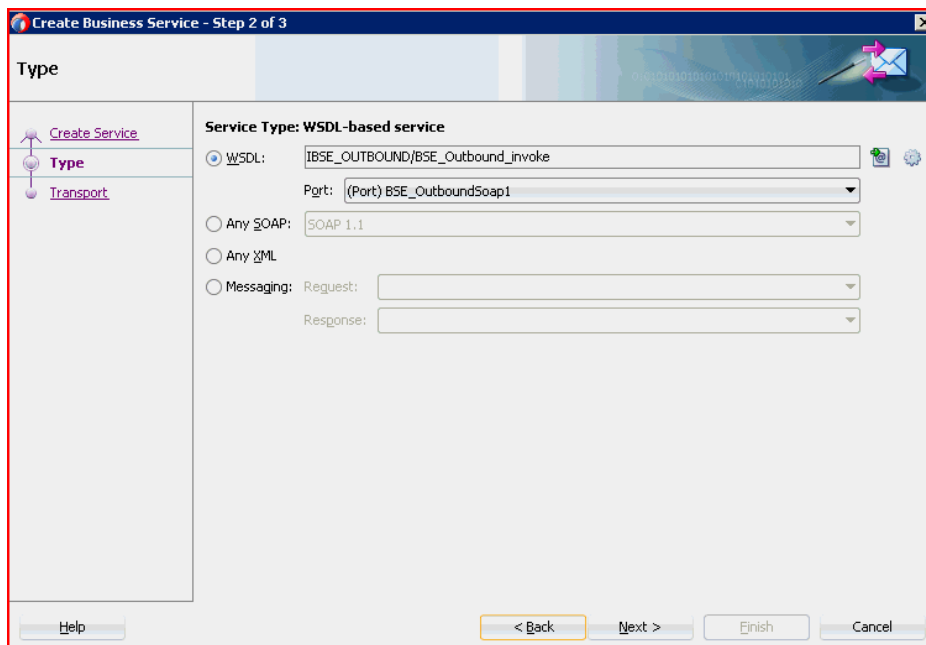
Figure 10–58 Configuration Pane



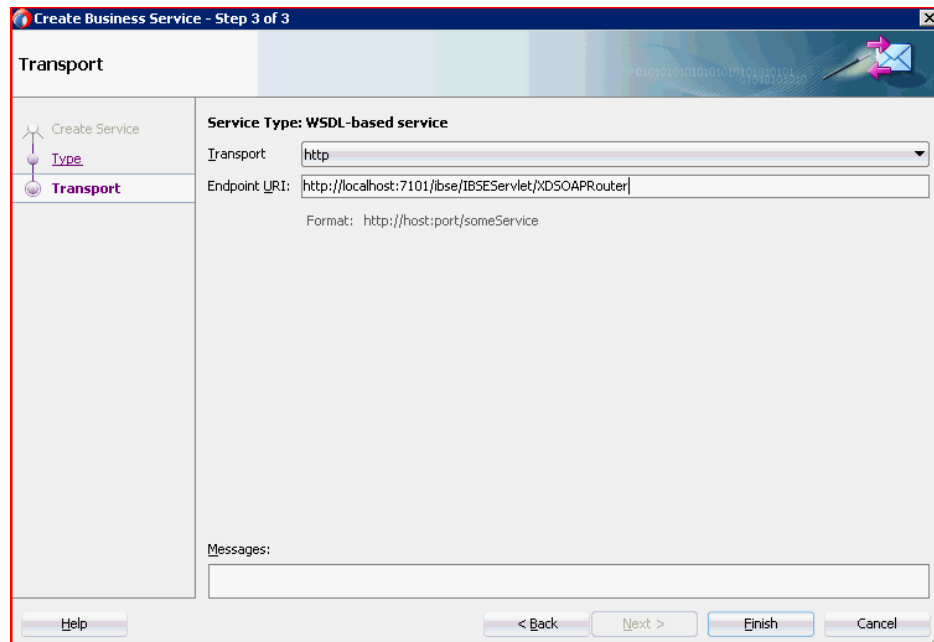
You are returned to the Create Business Service window.

- In the displayed Type pane, click **Next**, as shown in [Figure 10–59](#).

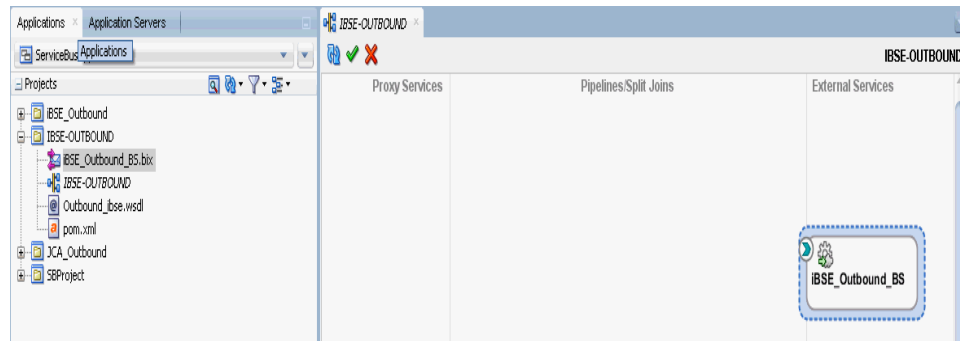
Figure 10–59 Type Pane



- 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 10–60](#).

Figure 10–60 Transport Pane

The Business Service is created and displayed in the External Services pane, as shown in [Figure 10–61](#).

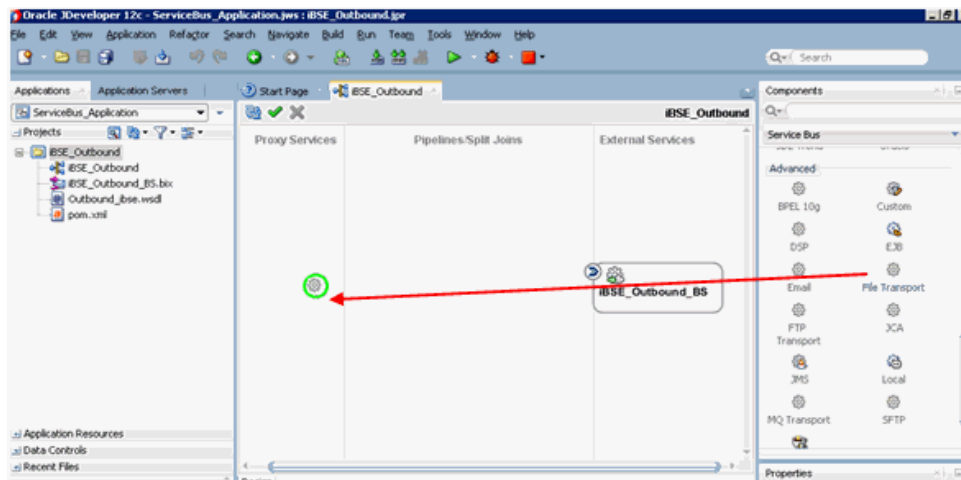
Figure 10–61 External Services Pane

10.3.2.2 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 10–62](#).

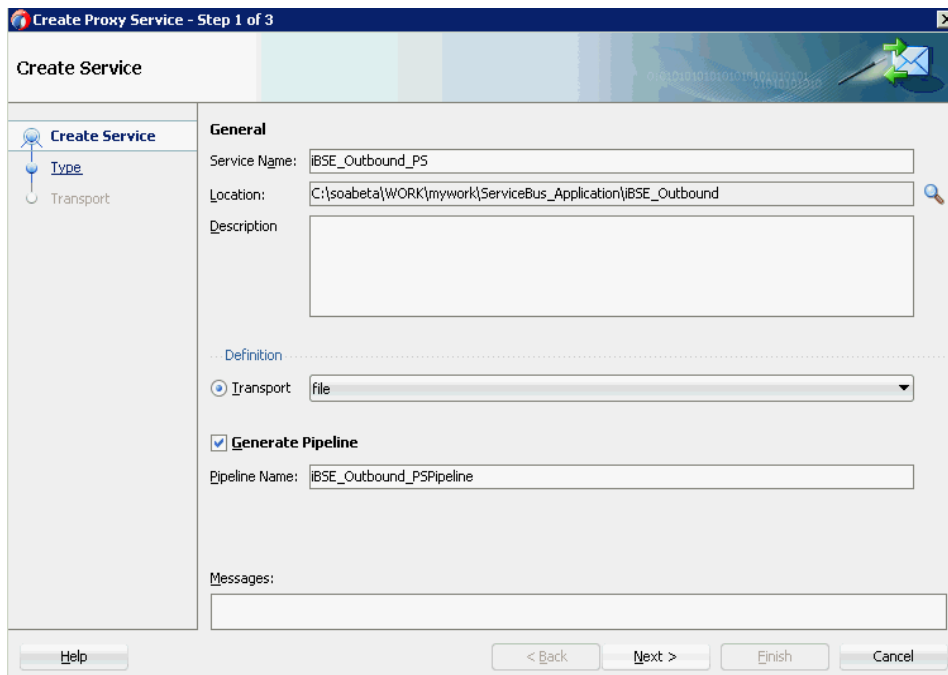
Figure 10–62 File Transport Component



The Create Proxy Service pane is displayed.

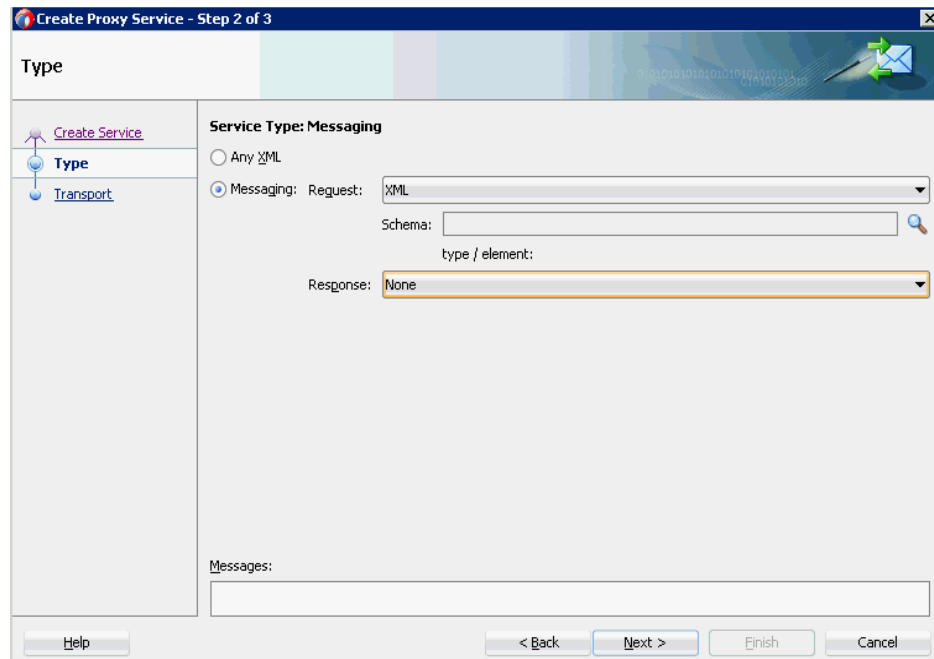
2. In the Service Name field, enter any name you wish for the Proxy service (for example, JCA_Outbound_PS). By default, **Generate Pipeline** is selected.
3. Click **Next**, as shown in [Figure 10–63](#).

Figure 10–63 Create Service Pane



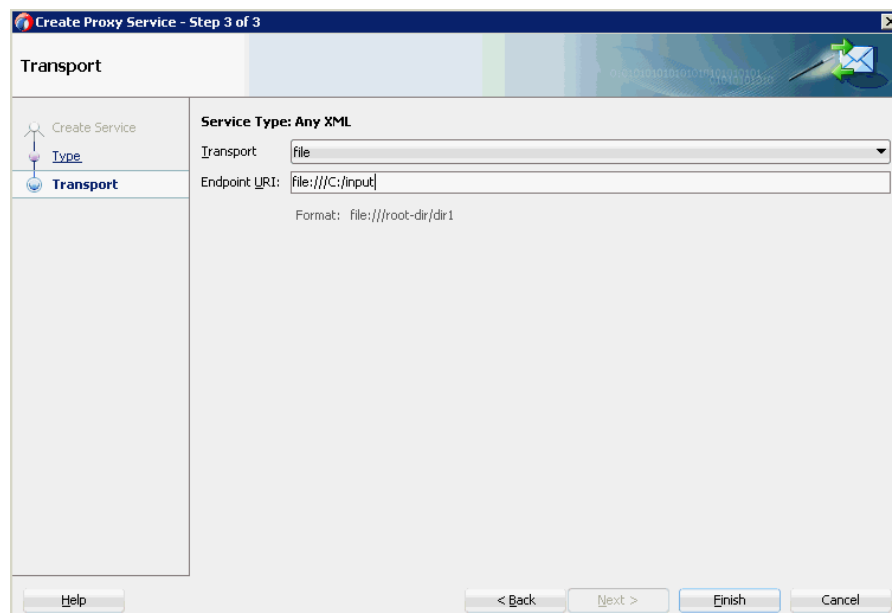
The Type pane is displayed.

4. Select the **Messaging** option, set the Request to **XML** and Response as **None**, and then click **Next**, as shown in [Figure 10–64](#).

Figure 10–64 Type Pane

The Transport window is displayed.

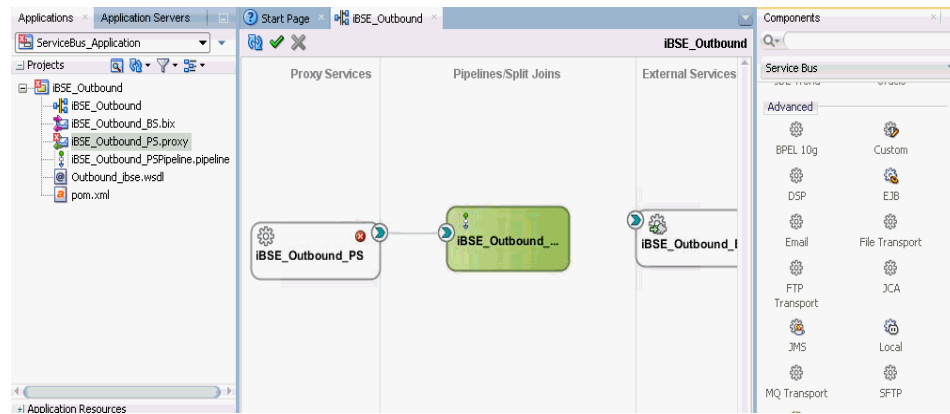
5. Provide the input location in the Endpoint URI field (for example, c:/input) and click **Finish**, as shown in [Figure 10–65](#).

Figure 10–65 Transport Window

The Proxy service along with the pipeline is created and displayed.

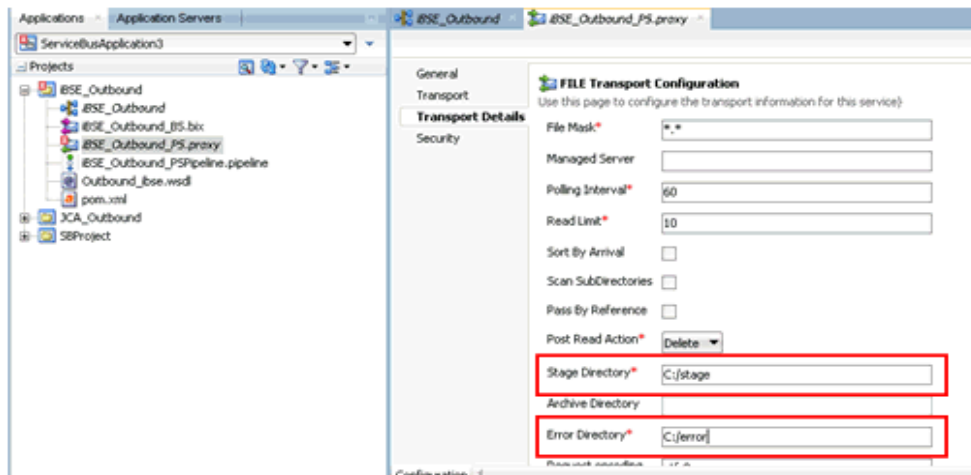
6. Double-click the created Proxy Service (for example: iBSE_Outbound_PS), as shown in [Figure 10–66](#).

Figure 10–66 Proxy Service Edit



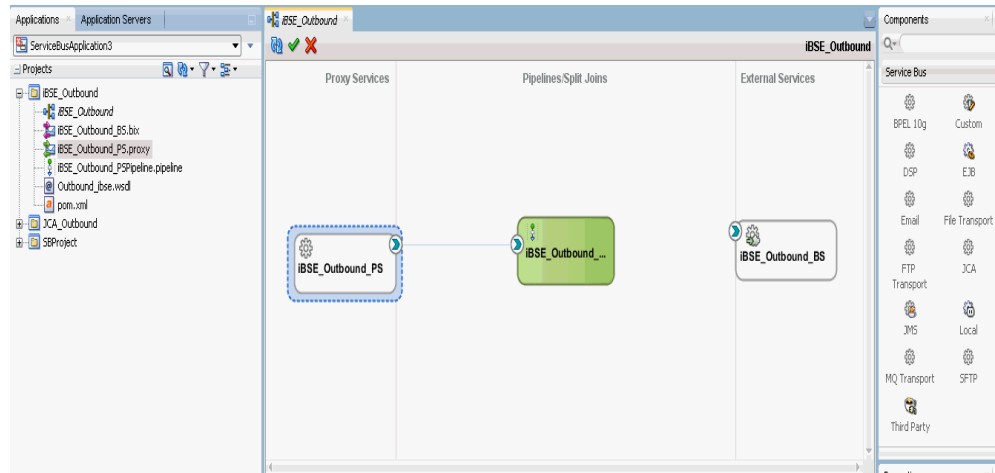
7. In the displayed Proxy Service configuration page, select **Transport Details** and provide the values for Stage and Error Directory, as shown in [Figure 10–67](#).

Figure 10–67 File Transport Configuration



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 10–68](#).

Figure 10–68 Proxy Service

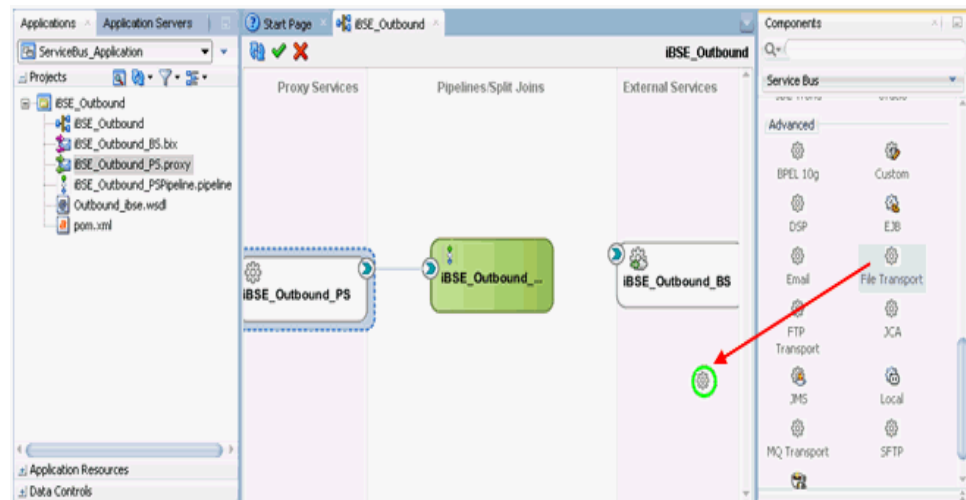


10.3.2.3 Configuring a File Transport Type Business Service

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 10–69](#).

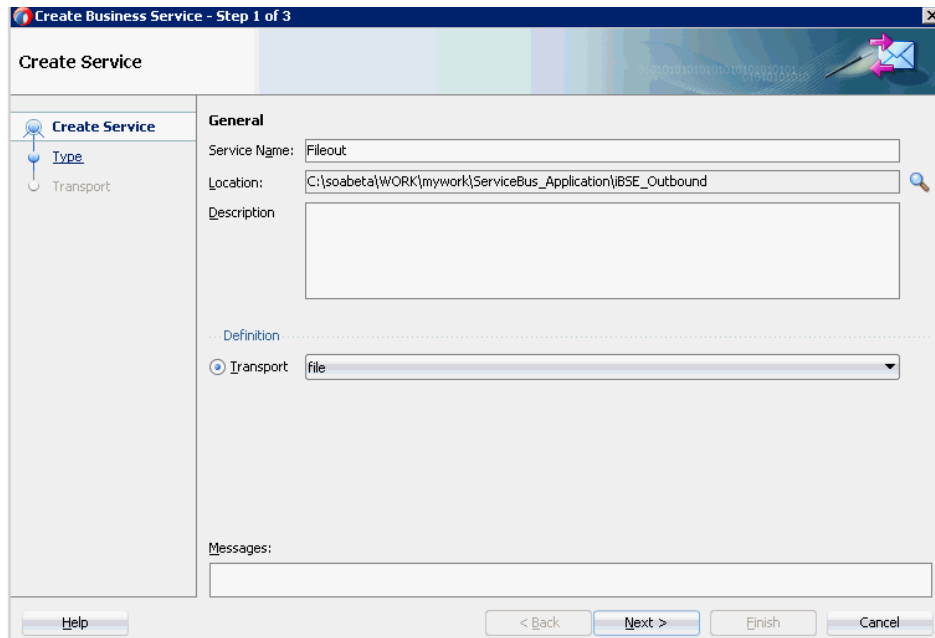
Figure 10–69 File Transport Component



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 10–70](#).

Figure 10–70 Create Service Pane



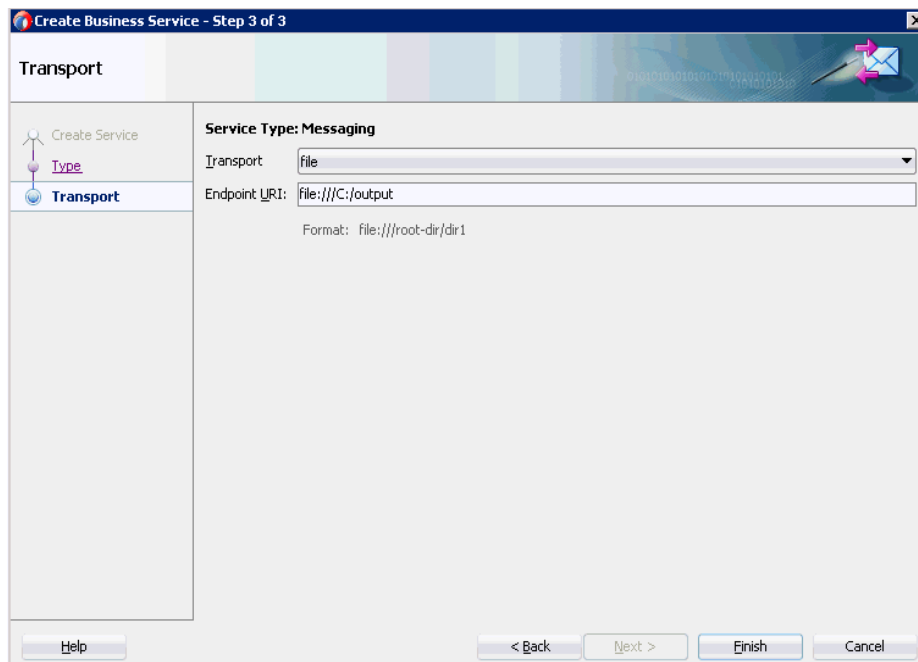
The Type pane is displayed. The **Any XML** option is selected by default.

3. Click Next.

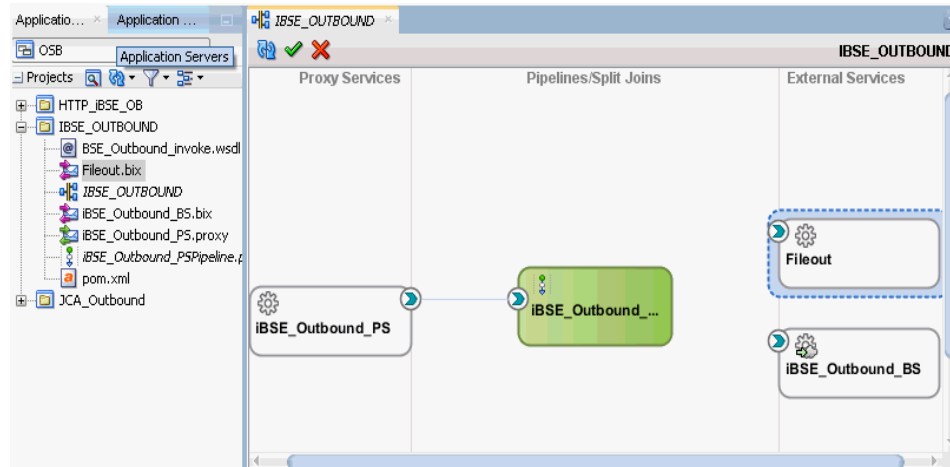
The Transport pane appears.

4. Provide the output location in the Endpoint URI field (for example, c:/output) and click Finish, as shown in Figure 10–71.

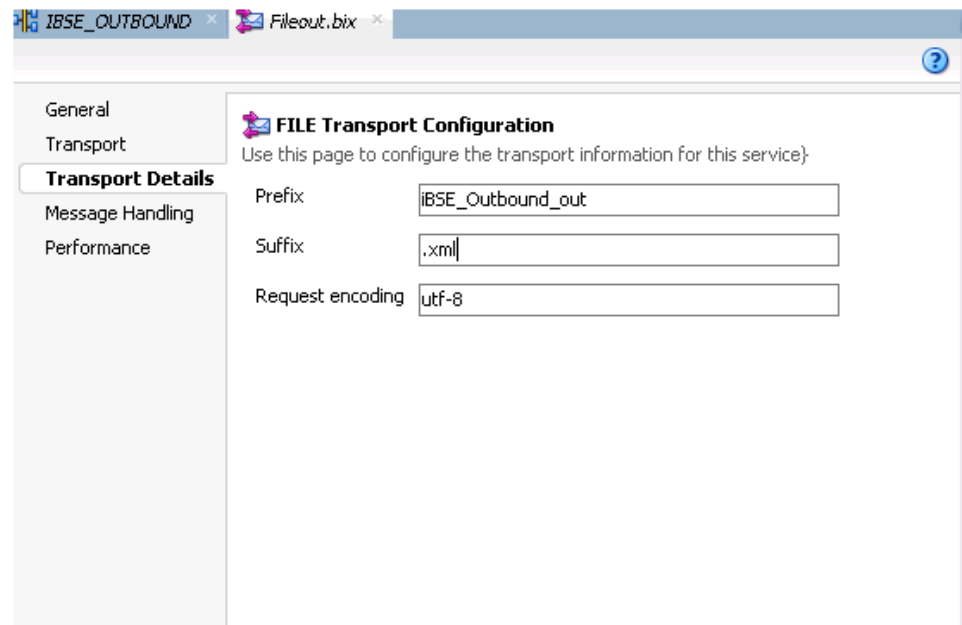
Figure 10–71 Transport Pane



The File Transport Business service Fileout is created and displayed, as shown in Figure 10–72.

Figure 10–72 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 10–73](#).

Figure 10–73 Transport Details

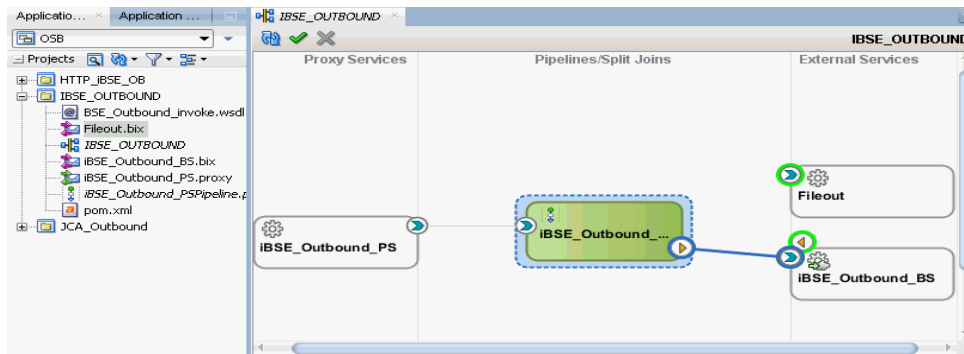
6. Save and close the configuration page, and double-click on overview.xml (for example, iBSE_Outbound).

10.3.2.4 Configuring the Routing Rules

Perform the following steps to configure the routing rules:

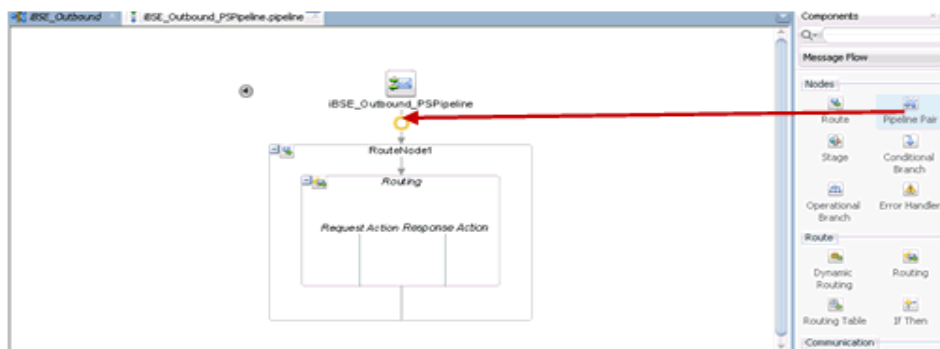
1. Create a connection between the Pipeline Component (for example, iBSE_Outbound_PSPipeline) and the WSDL based Business Service (for example, iBSE_Outbound_BS), as shown in [Figure 10–74](#).

Figure 10–74 Pipeline Component



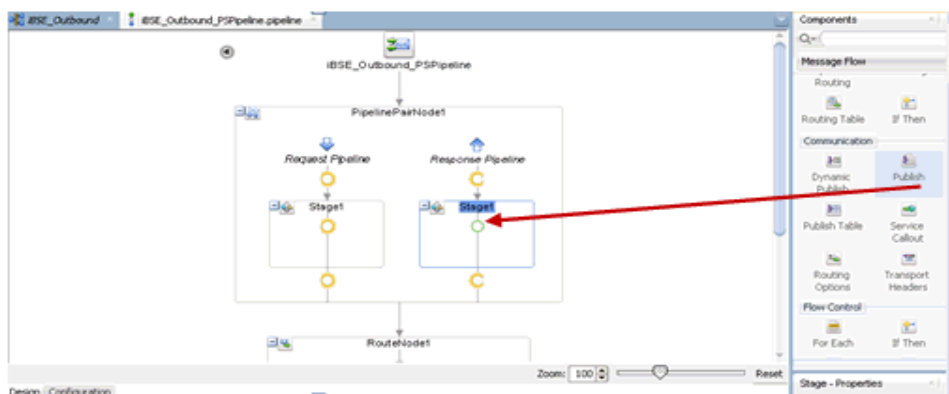
2. Double-click on the **Pipeline** component (for example, iBSE_Outbound_PSPipeline) in the Pipelines/Split Joins pane.
3. Drag and drop the **Pipeline Pair** node from Nodes pane to the area between the Pipeline (for example: iBSE_Outbound_PSPipeline) and RouteNode1, as shown in Figure 10–75.

Figure 10–75 Pipeline Pair Node



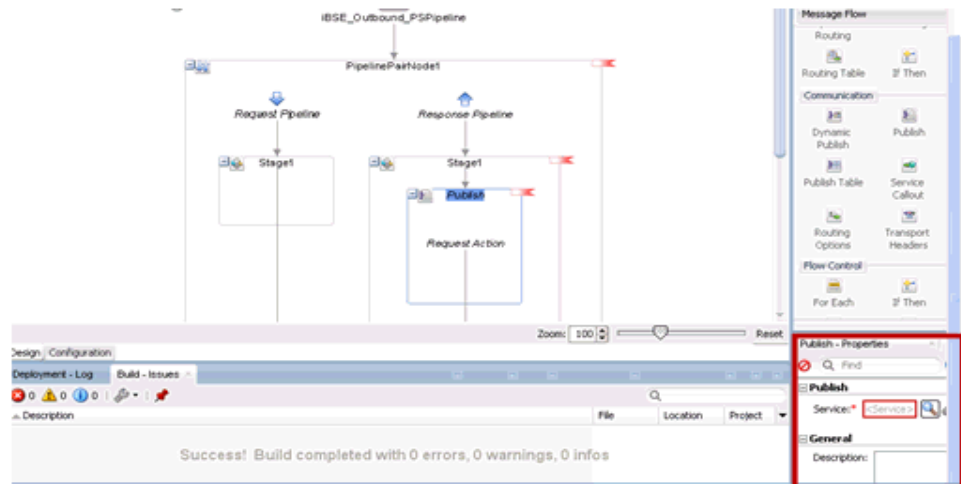
4. Drag and drop the **Publish** node from the Communication pane to the area beneath Stage1 of the Response Pipeline, as shown in Figure 10–76.

Figure 10–76 Publish Node



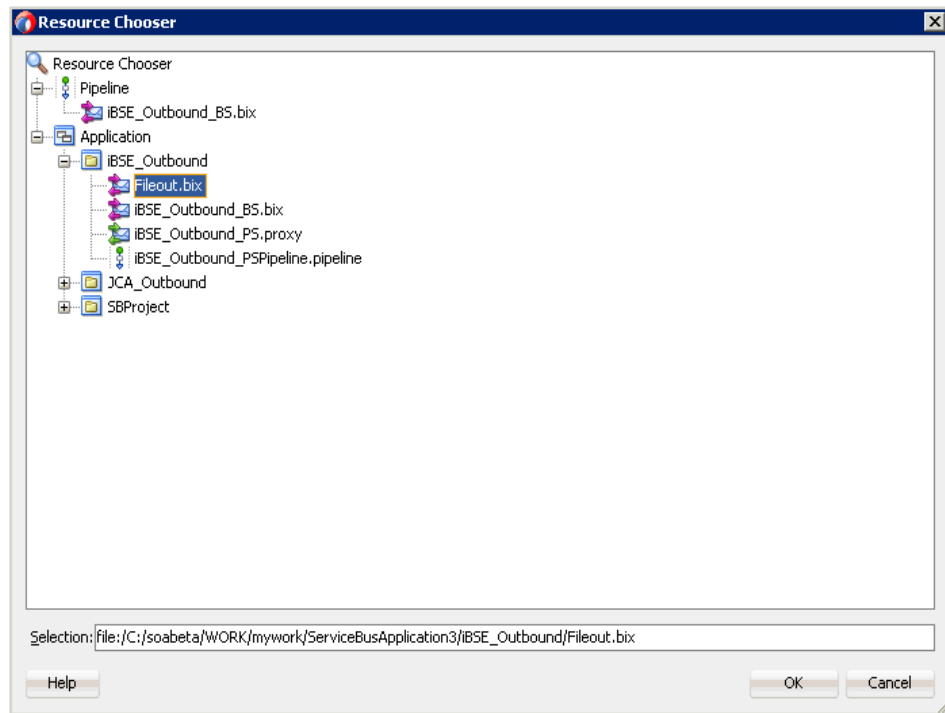
5. Click on the browse icon to the right of the Service field in the right pane of Publish Properties, as shown in Figure 10–77.

Figure 10-77 Browse Icon



6. In the displayed Resource Chooser window, select the **Fileout.bix** File Transport Business service and click **OK**, as shown in [Figure 10-78](#).

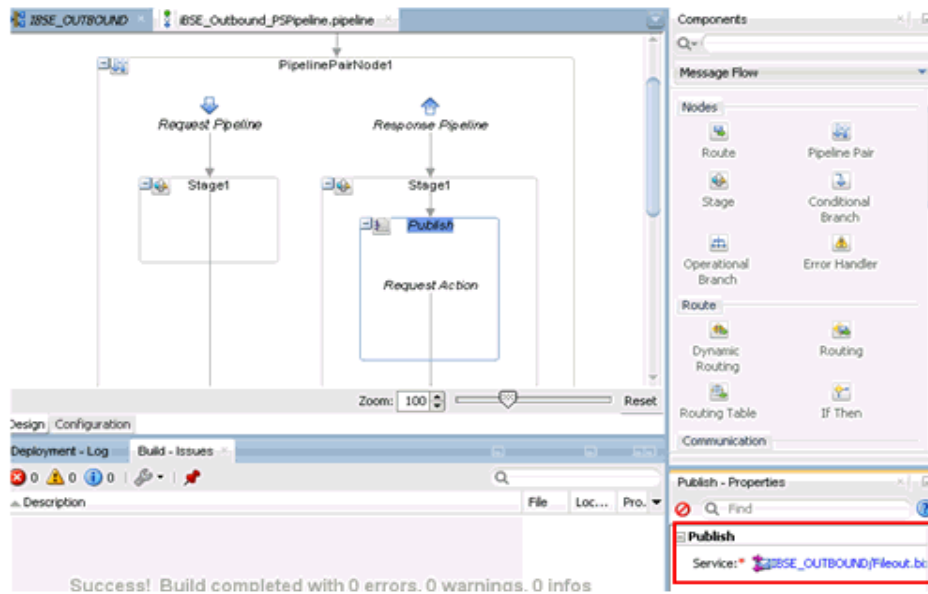
Figure 10-78 Resource Chooser



You are returned to the Pipeline configuration page.

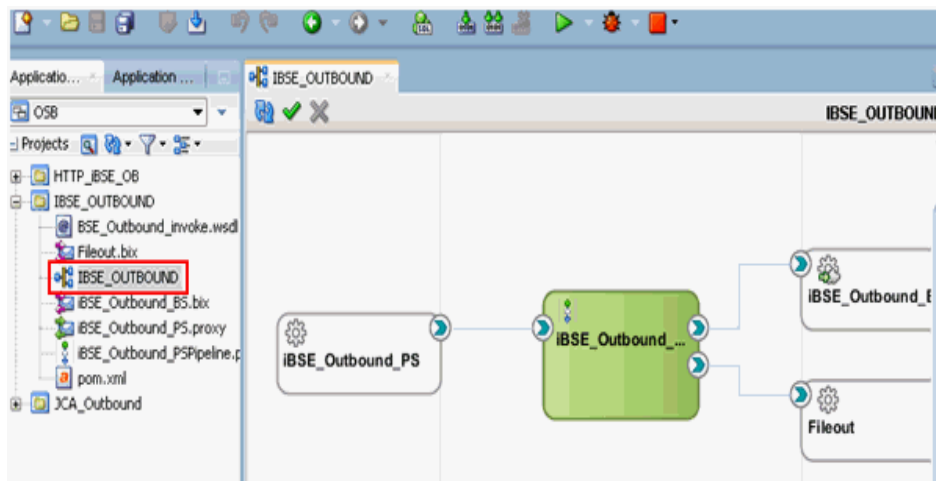
In the right pane, the selected service is configured in the Publish pane, as shown in [Figure 10-79](#).

Figure 10–79 Publish Pane



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 10–80](#).

Figure 10–80 Save All Icon



10.3.3 Deploying the OSB Outbound Process

To deploy the created OSB outbound process and invoke the input XML document, see [Section 10.1.3, "Deploying the OSB Outbound Process"](#).

10.4 Working With Synchronous SAP R/3 Events of Oracle Service Bus Using JDeveloper (J2CA Configuration)

This section describes how to work with synchronous SAP R/3 events of Oracle service Bus using Oracle JDeveloper for J2CA configurations.

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

```
<ADAPTER_HOME>\etc\sample\SAP_Samples.zip\SAP_Samples\OSB_Jdeveloper\J2CA\SAP_Sample_J2CA_OSB_Synchronous_Events
```

This section includes the following topics:

- [Section 10.4.1, "Creating a Service Bus Application for OSB"](#)
- [Section 10.4.2, "Defining an OSB Process"](#)
- [Section 10.4.3, "Deploying an OSB Process"](#)

Prerequisites

Before you design an OSB process, you must generate the respective WSDL files using Application Explorer.

Perform the following steps to generate a WSDL file for a message:

1. Create the first SAP adapter target (for example, target1) and channel. Ensure that the channel is not started before generating the WSDL file.
2. Connect to the first SAP adapter target (for example, target1).
3. Generate an inbound WSDL file for BAPI_COMPANYCODE_GETLIST by clicking **Remote Function Modules**, selecting **Financial Accounting**, and then clicking the **0002 - Company Code** business object.
4. Connect to the second SAP adapter target (for example, target2).
5. Generate an outbound WSDL file for BAPI_COMPANYCODE_GETLIST by clicking **Remote Function Modules**, selecting **Financial Accounting**, and then clicking on the **0002 - Company Code** business object.

For more information on creating a target, channel, and WSDL files, see [Chapter 4, "Configuring Oracle Application Adapter for SAP R/3"](#).

10.4.1 Creating a Service Bus Application for OSB

For more information on creating a Service bus application for OSB, see [Section 10.1.1, "Creating a Service Bus Application for OSB"](#) on page 10-2.

10.4.2 Defining an OSB Process

This section describes how to define an OSB process. The following topics are included:

- [Section 10.4.2.1, "Configuring a Third Party Adapter Service Component for Outbound and Inbound WSDLs"](#)
- [Section 10.4.2.2, "Creating a Pipeline"](#)
- [Section 10.4.2.3, "Configuring the Routing Rules"](#)

10.4.2.1 Configuring a Third Party Adapter Service Component for Outbound and Inbound WSDLs

Perform the following steps to create a third party adapter service component for Outbound and Inbound WSDLs:

1. Drag and drop the **Third Party Adapter** component from the Service Bus Components pane to the External Services pane.

The Create Third Party Adapter Service dialog is displayed.

2. Enter an appropriate name for the Business Service.
3. Click the Find existing WSDLs icon, which is located to the right of the WSDL URL field.

The WSDL Chooser dialog is displayed.

4. Select the **File System** tab, then browse, and select the **outbound WSDL** file from the WSDL directory.
5. Click **OK**.

The Import Service Bus Resources dialog is displayed.

6. Click **Next**.
7. In the Configuration pane, click **Finish**.

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. Select the outbound JCA properties file from the WSDL directory.
10. Click **OK**.

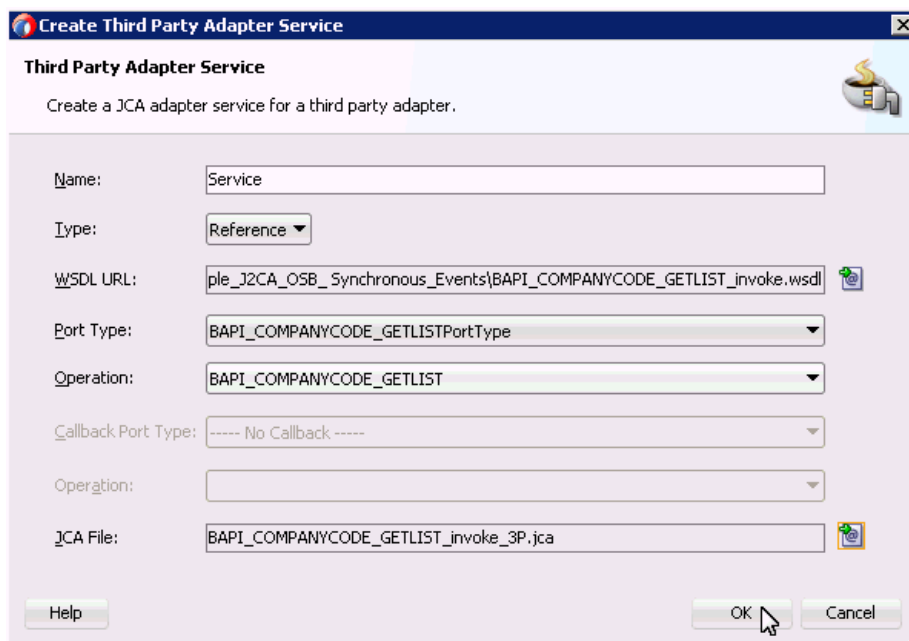
The Copy File dialog window is displayed.

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, as shown in [Figure 10–81](#).

Figure 10–81 Create Third Party Adapter Service Dialog



12. Click **OK**.

The Business Service will be created in the External Services.

13. Drag and drop the **Third Party Adapter** component from the Service Bus Components pane to the Proxy Services pane.

The Create Third Party Adapter Service dialog is displayed.

14. Enter an appropriate name for the Proxy Service.
15. Click the Find existing WSDLs icon, which is located to the right of the WSDL URL field.

The WSDL Chooser dialog is displayed.

16. Under the **File System** tab, select the **inbound WSDL** file from the WSDL directory.
17. Click **OK**.

The Import Service Bus Resources dialog is displayed.

18. Click **Next**.
19. In the Configuration pane, click **Finish**.

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

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

The Transformation Chooser dialog is displayed.

21. Select the inbound JCA properties file from the WSDL directory.

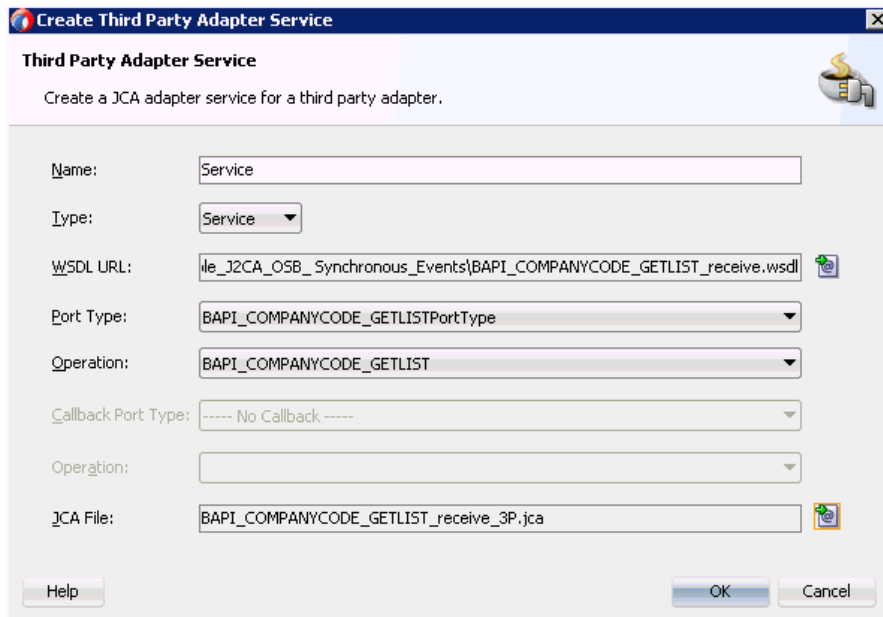
22. Click **OK**.

The Copy File dialog window is displayed.

23. 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 10-82](#).

Figure 10–82 Create Third Party Adapter Service Dialog

24. Click **OK**.

Proxy Service will be created in the Proxy Services pane.

Note: A red mark will be shown in the created Proxy service which will be removed after mapping with the configured Pipeline.

10.4.2.2 Creating a Pipeline

Perform the following steps to create a Pipeline:

1. Drag and drop the Pipeline component from the Resources pane to the Pipelines/Split Joins pane.
The Create Pipeline Service window is displayed.
2. Enter any name for the Pipeline and click **Next**.
3. Select the **WSDL** button and click on the Browse WSDLs icon near the field.
The Select WSDL window is displayed.
4. Select the Application tab.
5. Click the appropriate project folder and select the inbound concrete WSDL (for example, ProxyService_Name-concrete.wsdl) and then click **OK**.
You are returned to the Create Pipeline Service window.
6. Clear the check box for **Expose as a Proxy Service**.
7. Click **Finish**.
The Proxy service is created.
8. Connect the Proxy Service and Pipeline, and then save the changes.

10.4.2.3 Configuring the Routing Rules

Perform the following steps to configure the routing rules:

1. Connect the Pipeline and the Business Service.
2. Double-click on the pipeline in the Pipelines/Split Joins pane.
The Pipeline configuration page is displayed.
3. Drag and drop the **Log** node from the Reporting components pane to the area beneath Request Action.
4. In the Log - Properties pane, click the icon near <expression> in the Content field.
5. In the XQuery Expression Builder page, type `$body` in the text area below the Expression field and click **Ok**.
6. Enter any text message you wish in the Summary field (optional) and select **Error** from the Severity level drop-down list under the Log area, as shown in [Figure 10–83](#).

Figure 10–83 Severity Level Drop-down List

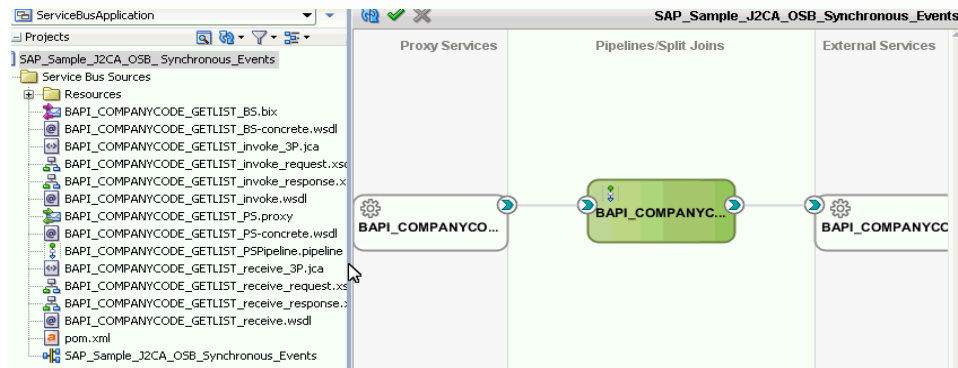
The screenshot shows the 'Log' configuration pane with the 'General' tab selected. The 'Content' field is set to '\$body' with a small icon to its right. The 'Summary' field contains the text '*****Inbound*****'. The 'Severity' field is a dropdown menu with 'Error' selected. A vertical scrollbar is visible on the right side of the pane.

7. Drag and drop the **Log** node from the Reporting components pane to the area beneath Response Action.
8. In the Log - Properties pane, click the icon near <expression> in the Content field.
9. In the XQuery Expression Builder page, type `$body` in the text area below the Expression field and click **Ok**.
10. Enter any text message you wish in the Summary field (optional) and select **Error** from the Severity level drop-down list under the Log area, as shown in [Figure 10–84](#).

Figure 10–84 Severity Level Drop-down List

The screenshot shows the 'Log' configuration pane with the 'General' tab selected. The 'Content' field is set to '\$body' with a small icon to its right. The 'Summary' field contains the text '*****Outbound Output*****'. The 'Severity' field is a dropdown menu with 'Error' selected.

11. Save and Close the Pipeline configuration page, as shown in [Figure 10–85](#).

Figure 10–85 Pipeline Configuration Page

You are now ready to deploy the OSB process.

10.4.3 Deploying an OSB Process

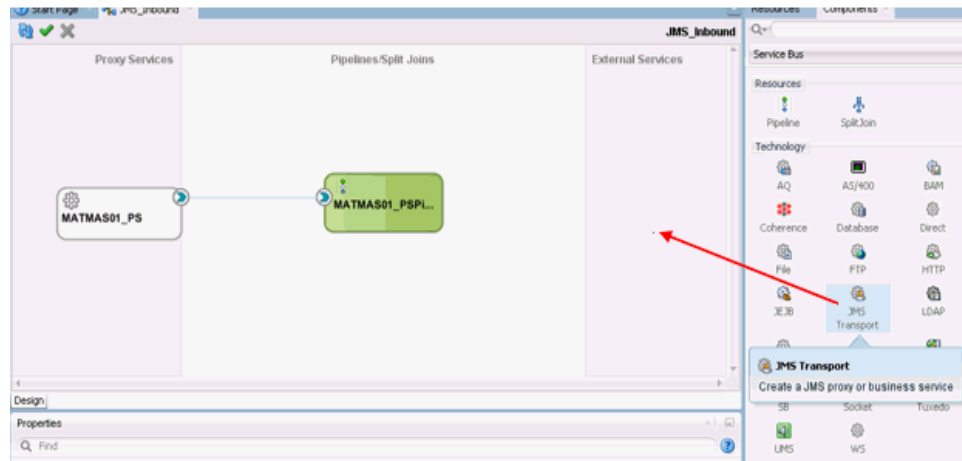
For more information on deploying an OSB process, see [Section 10.1.3, "Deploying the OSB Outbound Process"](#) on page 10-17.

10.5 Configuring a JMS Inbound Process Using JDeveloper (J2CA Configuration)

This section describes how to configure a JMS inbound process to your SAP R/3 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 6.5.1, "Generating WSDL for Event Integration"](#) on page 6-37.
2. Start the Oracle JDeveloper and create a Service Bus Application for OSB. For more information, see [Section 10.1.1, "Creating a Service Bus Application for OSB"](#) on page 10-2.
3. Create a Third Party Adapter Service Component. For more information, see [Section 7.3.2.1, "Configuring a Third Party Adapter Service Component"](#) on page 7-14.
4. Create a Proxy Service along with the pipeline from the JCA Binding File. For more information, see [Section 10.2.2.2, "Creating a Pipeline"](#) on page 10-25.
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 10–86](#).

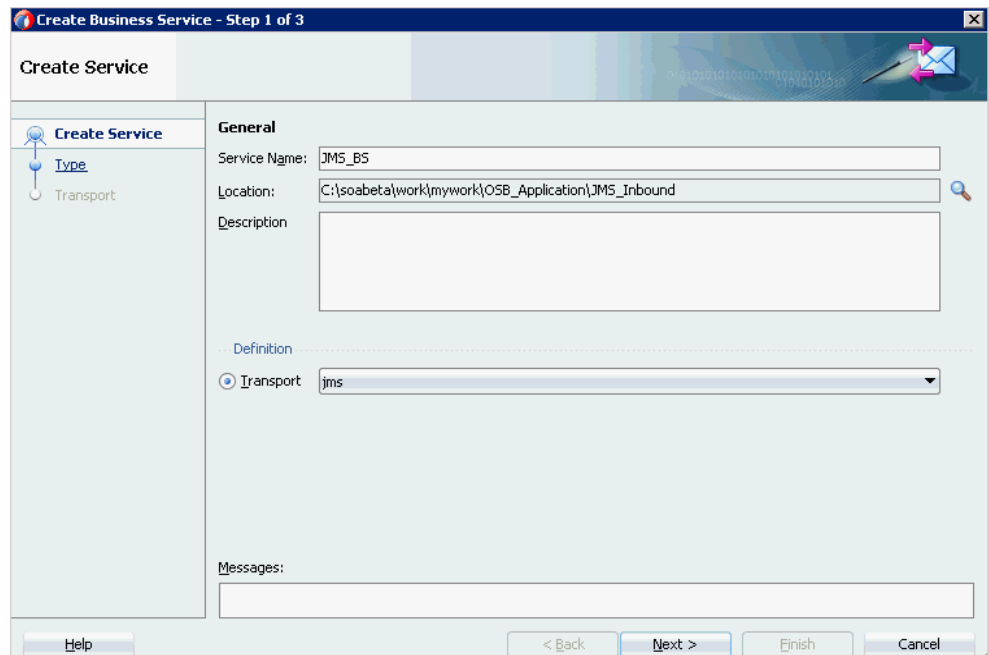
Figure 10–86 JMS Transport Component



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 10–87](#).

Figure 10–87 Create Service Pane

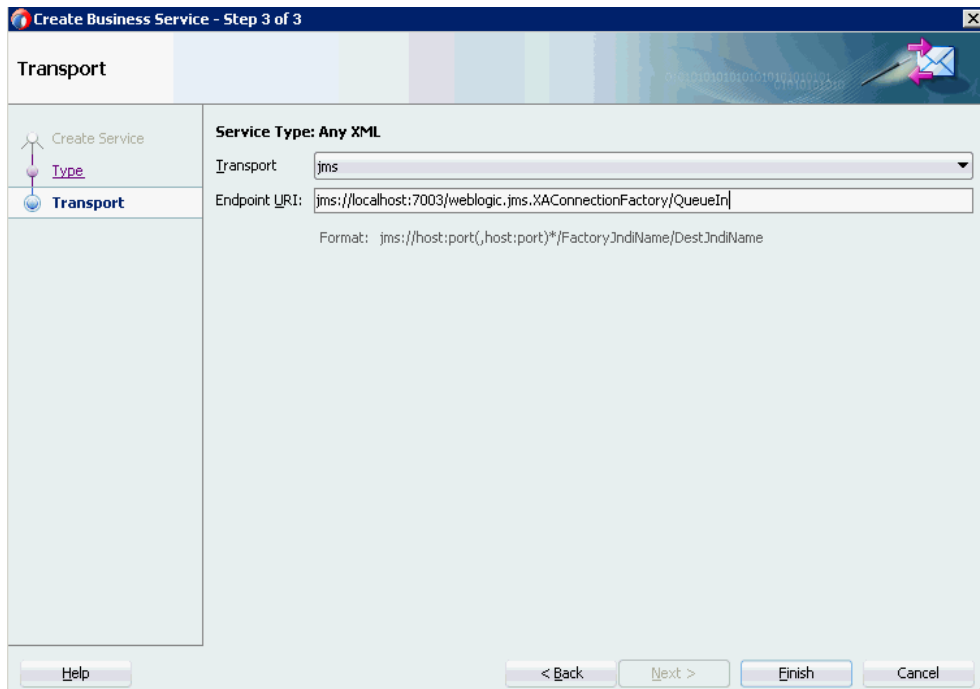


- c. In the displayed Type window, select **Any XML** and then click **Next**.

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 10–88](#).

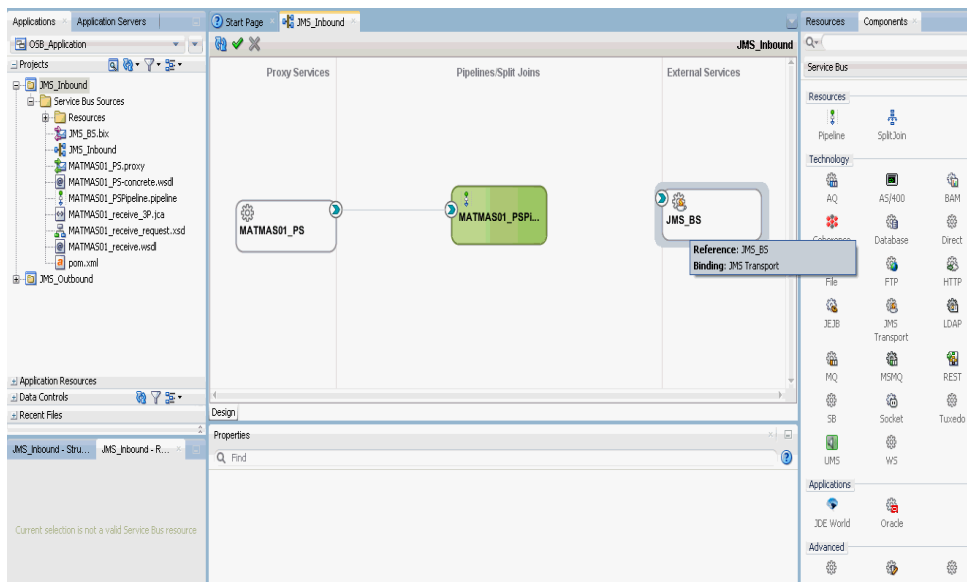
Figure 10–88 Transport Window



The JMS Business service is created and displayed.

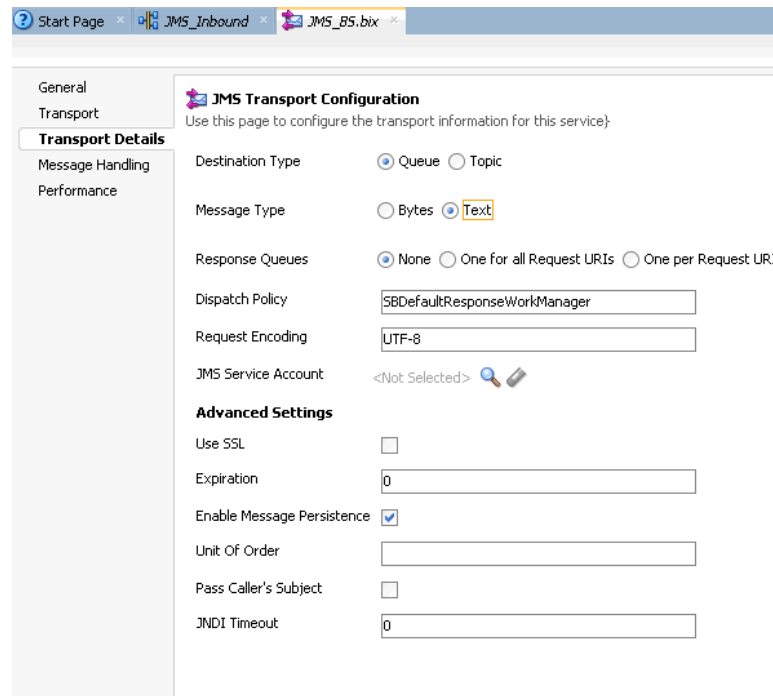
- e. Double-click **JMS_BS** as shown in [Figure 10–89](#).

Figure 10–89 JMS Business Service



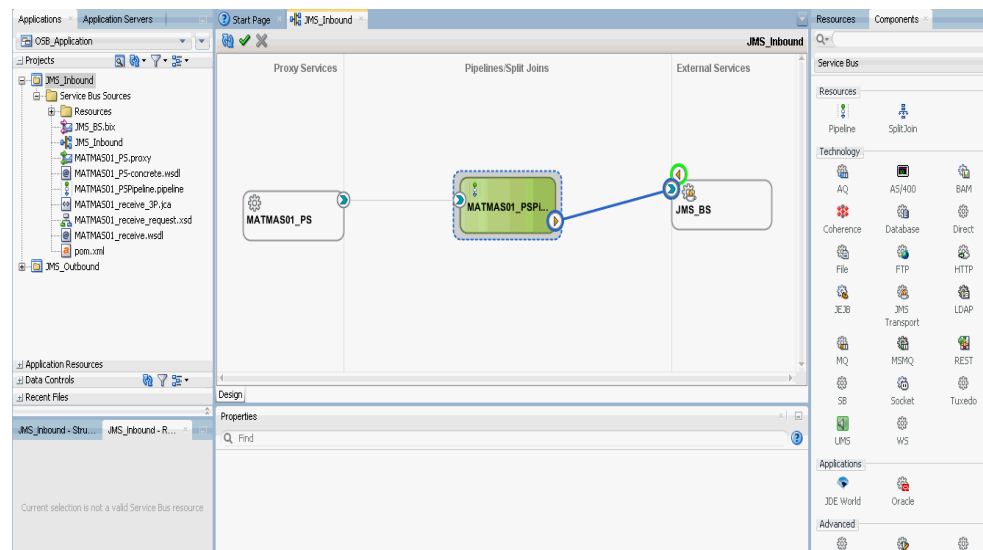
- f. In the displayed Business Service configuration page, provide the following parameters in the Transport Details tab, as shown in [Figure 10–90](#).

Figure 10–90 JMS Transport Configuration



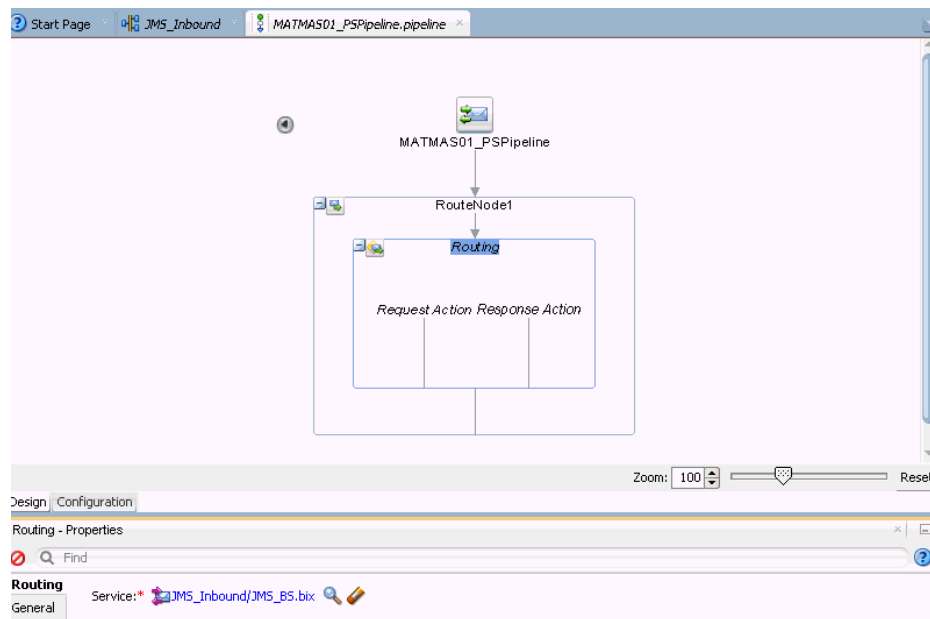
- g. In the Destination Type section, select **Queue**.
 - h. In the Message Type section, select **Text**.
6. Save and close the Configuration page of the business service.
 7. Create a connection between **Pipeline** (for example, xxxx_PSPipeline) and **JMS Business Service** (for example, JMS_BS) as shown in figure [Figure 10–91](#).

Figure 10–91 Configuration Page



8. Double-click **Pipeline**.
The Pipeline Configuration page is displayed as shown in [Figure 10–92](#).

Figure 10–92 Pipeline Configuration

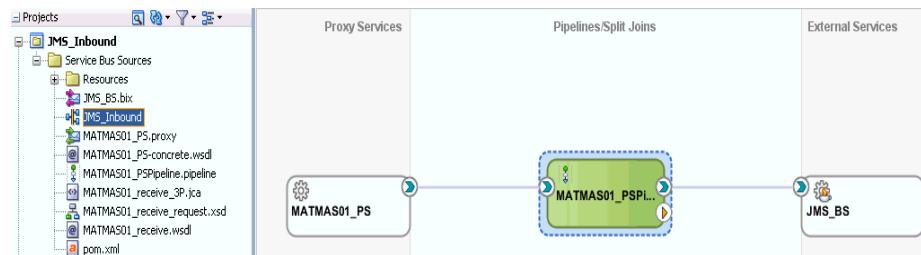


9. Check that the details are configured properly, and then save and close the Pipeline configuration page.

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 10–93](#).

Figure 10–93 Save All Icon



11. Deploy the OSB JMS inbound process. For more information, see [Section 10.2.3, "Deploying the OSB Inbound Process"](#) on page 10-30.
12. Once the process is deployed successfully, trigger the event messages. For more information, see [Section 6.5.5, "Triggering an Event in SAP R/3"](#) on page 6-54.
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 10–94](#).

Figure 10–94 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

Configuration | Subdeployments | Targets | Security | Notes

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/busResources-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.

[Customize this table](#)

Summary of Resources

Click the **Lock & Edit** button in the Change Center to activate all the buttons on this page.

New | Delete 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_ProxyResponse230659500	Queue	JMS_ProxyResponse	JMS_ProxyResponse230659500	wlsbJMSServer
QueueIn	Queue	QueueIn	wlsbJMSServer	wlsbJMSServer
QueueIn.Quota	Quota	N/A	N/A	N/A
TemporaryTmp8	Template	N/A	N/A	N/A

16. Click the Monitoring tab, as shown in Figure 10–95.

Figure 10–95 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

Configuration | **Monitoring** | Control | Security | Subdeployment | Notes

General | Thresholds and Quotas | Overrides | Logging | Delivery Failure

Click the **Lock & Edit** button in the Change Center to modify the settings on this page.

Save

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:

17. Select the check box and click the **Show Messages** button, as shown in Figure 10–96.

Figure 10–96 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

Configuration | **Monitoring** | Control | Security | Subdeployment | Notes

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	Messages High
jmsResourcesQueueIn	10	0	10	0	0	0	10

Show Messages Showing 1 to 1 of 1 Previous | Next

18. Click the ID link with the appropriate time and date, as shown in Figure 10–97.

Figure 10–97 Summary of JMS Messages Window

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)

New Delete Move Import Export Showing 1 to 10 of 10 Previous Next

ID	CorrId	Time Stamp	State String	JMS Delivery Mode	Message Size
ID:<785570.1400633930707.0>		Tue May 20 17:58:50 PDT 2014	visible	Persistent	13320
ID:<785570.1400633930774.0>		Tue May 20 17:58:50 PDT 2014	visible	Persistent	4473
ID:<785570.1400633930794.0>		Tue May 20 17:58:50 PDT 2014	visible	Persistent	12961
ID:<785570.1400633932186.0>		Tue May 20 17:58:52 PDT 2014	visible	Persistent	33583
ID:<785570.1400633932958.0>		Tue May 20 17:58:52 PDT 2014	visible	Persistent	33548
ID:<785570.1400633933037.0>		Tue May 20 17:58:53 PDT 2014	visible	Persistent	25156
ID:<785570.1400633933871.0>		Tue May 20 17:58:53 PDT 2014	visible	Persistent	18775
ID:<785570.1400633934379.0>		Tue May 20 17:58:54 PDT 2014	visible	Persistent	16671
ID:<785570.1400633934420.0>		Tue May 20 17:58:54 PDT 2014	visible	Persistent	14408
ID:<785570.1400633935250.0>		Tue May 20 17:58:55 PDT 2014	visible	Persistent	5684

New Delete Move Import Export Showing 1 to 10 of 10 Previous Next

The response document is shown under the Text field, as shown in [Figure 10–98](#).

Figure 10–98 Response Document

Home > Summary of JMS Modules > jmsResources > Summary of JMS Modules > QueueIn > Summary of JMS Modules > jmsResources > QueueIn > Summary of JMS Messages > ID:<785570.1400633930707.0>; jmsResources!QueueIn!0

JMS Message Detail

Message ID: ID:<785570.1400633930707.0> Delivery Mode: Persistent

Type: (No value specified) Correlation ID: (No value specified)

Timestamp: Tue May 20 17:58:50 PDT 2014 Expiration: (No value specified)

Priority: 4 Redelivered: false

Delivery Time: (No value specified) Redelivery Limit: -1

Properties:

key	value	type
JMSXDeliveryCount	0	java.lang.Integer

Text:

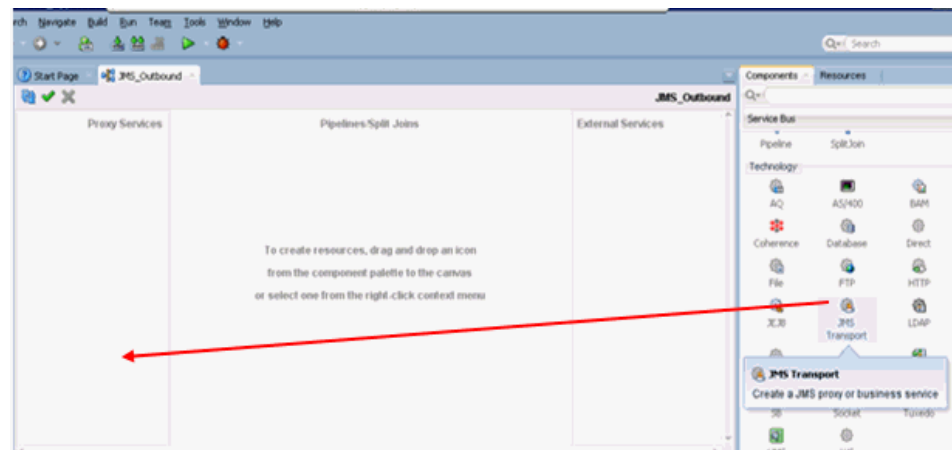
```
<?xml version="1.0" encoding="UTF-8"?>
<MATHAS01 tid="AC1301D72A27537B3D4600B9" xmlns="urn:sap-com:document:sap:idoc">
  <IDOC BEGIN="1">
    <EDI_DC40 SEGMENT="1">
      <TABNAM>EDI_DC40</TABNAM>
      <MANDT>800</MANDT>
      <DOCNUM>0000000005013447</DOCNUM>
    </EDI_DC40 SEGMENT="1">
  </IDOC BEGIN="1">
</MATHAS01 tid="AC1301D72A27537B3D4600B9" xmlns="urn:sap-com:document:sap:idoc">
```

10.6 Configuring a JMS Outbound Process Using JDeveloper (J2CA Configuration)

This section describes how to configure a JMS outbound process to your SAP R/3 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 6.4.1, "Generating WSDL for Request/Response Service"](#) on page 6-8.
2. Start the Oracle JDeveloper and create a Service Bus Application for OSB. For more information, see [Section 10.1.1, "Creating a Service Bus Application for OSB"](#) on page 10-2.
3. Create a Third Party Adapter Service Component. For more information, see [Section 7.3.2.1, "Configuring a Third Party Adapter Service Component"](#) on page 7-14.
4. Create a WSDL-based Business Service from the JCA Binding File. For more information, see [Section 10.1.2.2, "Configuring a File Transport Type Business Service"](#) on page 10-8.
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 10-99](#).

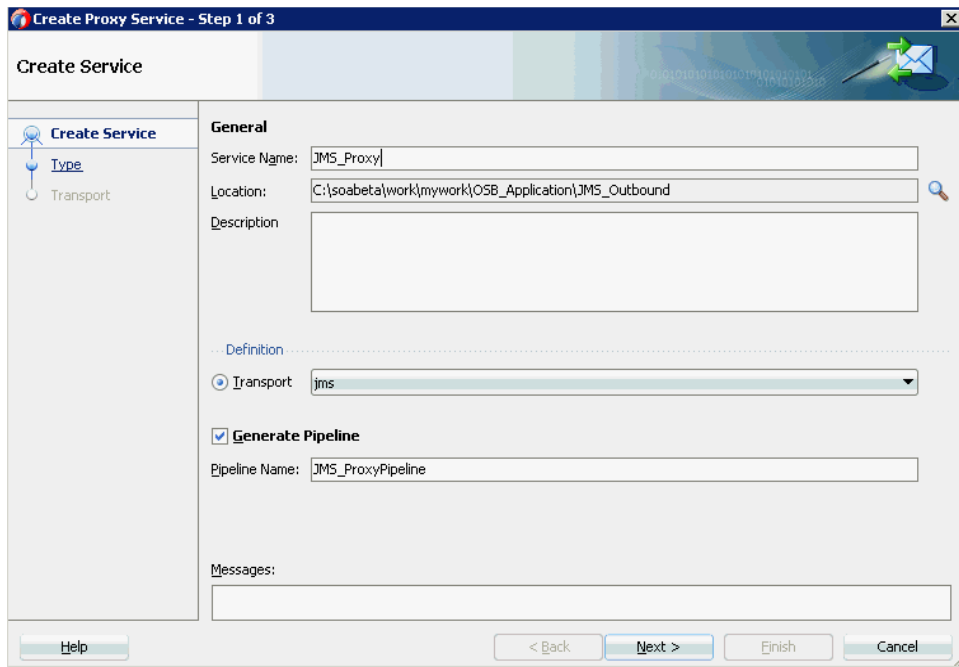
Figure 10-99 JMS Transport Component



The Create Business Service dialog is displayed.

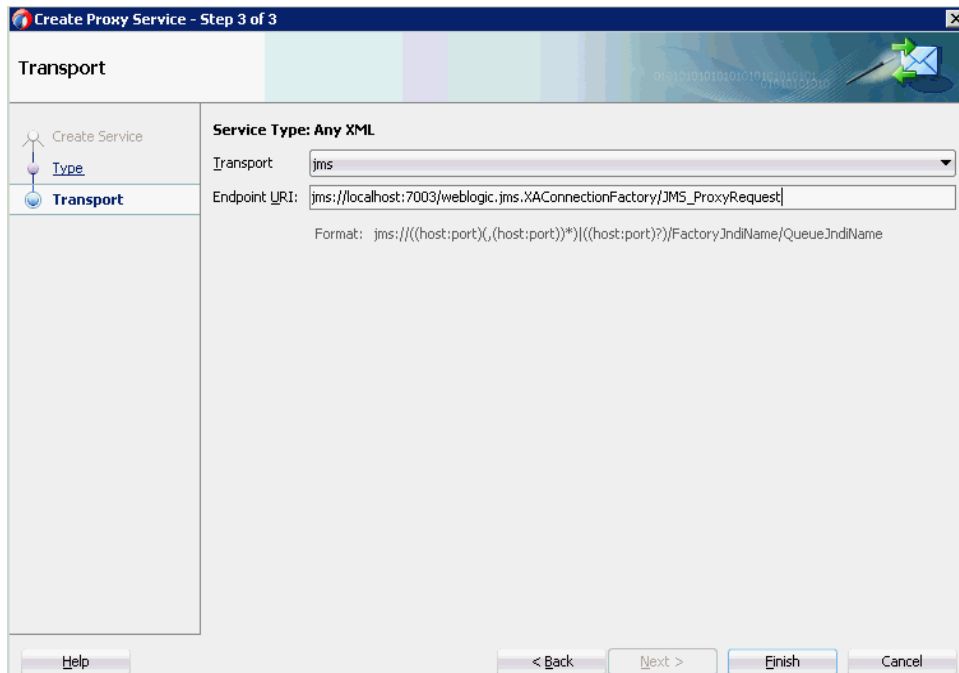
- b. In the Service Name field, enter any name you wish for the Proxy service (for example, JMS_Proxy). By default, Generate Pipeline is selected.
- c. Click **Next**, as shown in [Figure 10-100](#).

Figure 10–100 Create Proxy Service Pane



- 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 10–101](#).

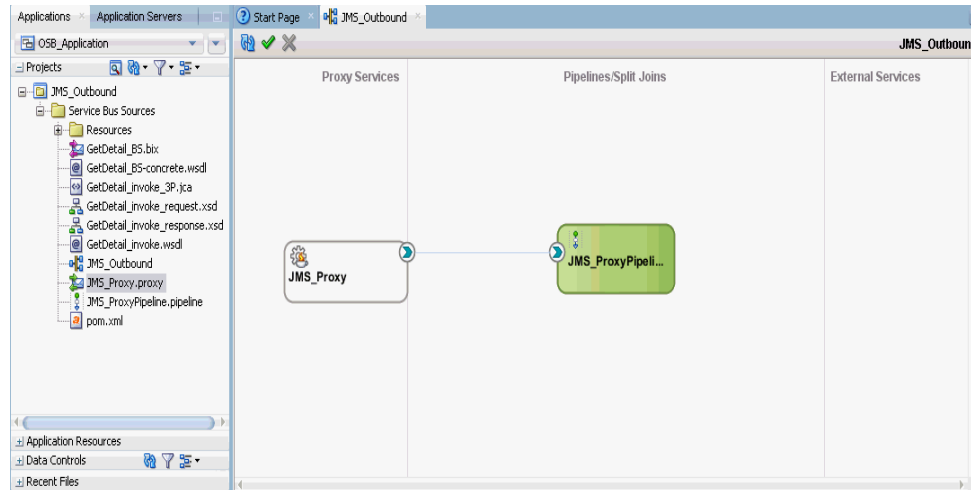
Figure 10–101 Transport Window



The JMS Proxy service along with the pipeline is created and displayed.

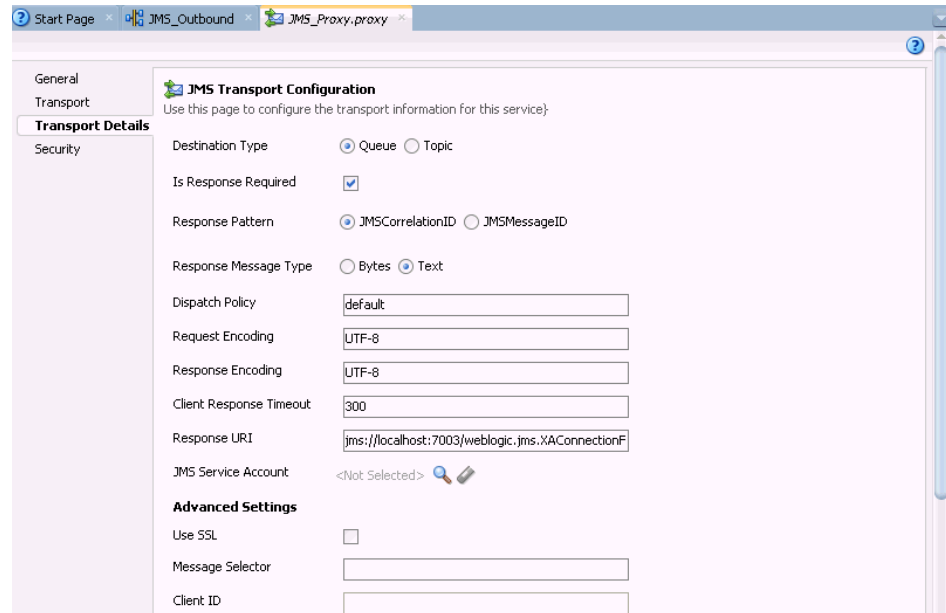
- f. Double-click the created Proxy Service (for example, JMS_Proxy), as shown in [Figure 10-102](#).

Figure 10-102 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 10-103](#).

Figure 10-103 JMS Transport Configuration

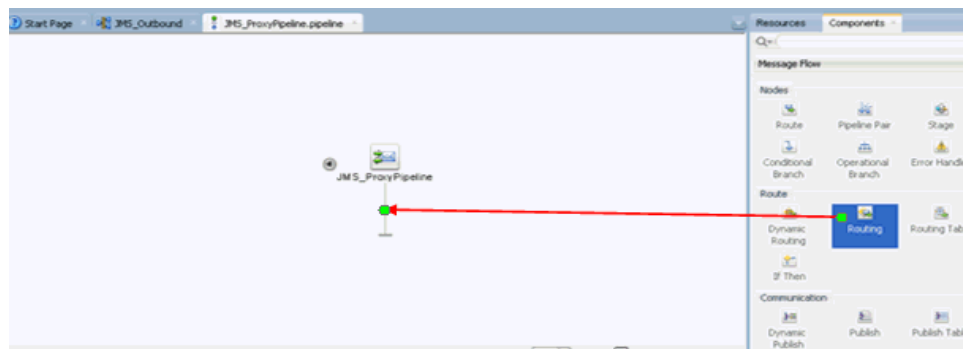


- h. In the Destination Type section, select **Queue**.
- i. Select the **Is Response Required** check box.
- j. In the Response Message Type section, select **Text**.
- k. In the Response URI field, provide the Endpoint URI used in the JMS Transport Configuration and change Request to Response. For example,

```
jms://localhost:7003/weblogic.jms.XAConnectionFactory/JMS_ProxyResponse
```

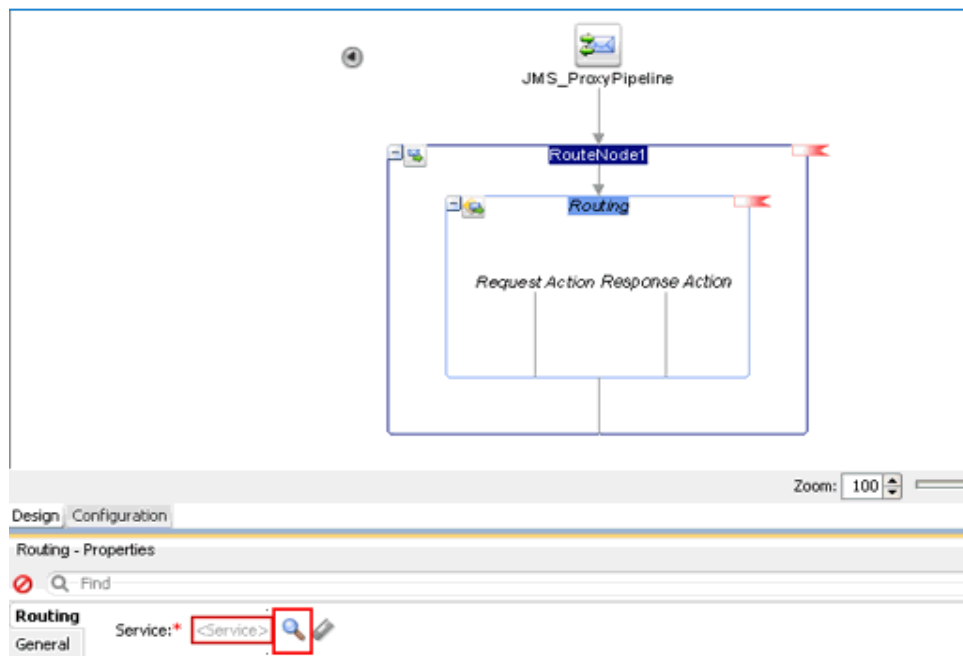
6. Save and close the Configuration page of the Proxy service.
7. Configure the Routing Rules and proceed with the following steps:
 - a. Double-click on the pipeline (for example, JMS_ProxyPipeline) in the Pipelines/Split Joins pane.
The Pipeline configuration page is displayed.
 - b. Drag and drop the **Routing** component from the Route section to the area below the Pipeline (for example, JMS_ProxyPipeline), as shown in [Figure 10-104](#).

Figure 10-104 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 10-105](#).

Figure 10-105 Browse Service



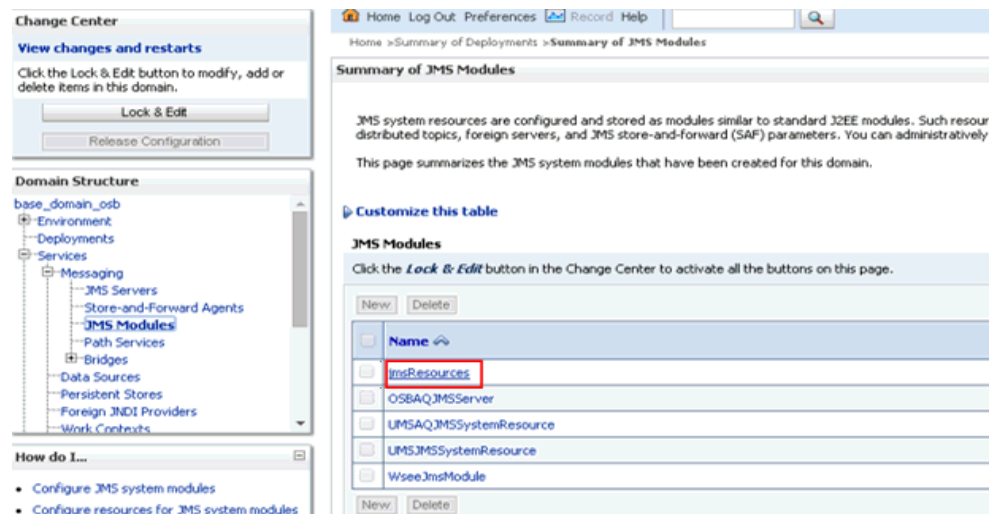
- d. In the displayed Resource Chooser window, select the WSDL-based Business service (for example, xxxxx_BS.bix) and click **OK**.
You are returned to the Pipeline configuration page.
- e. Save and Close the Pipeline configuration page.
You are returned to the composite editor window.
- f. Click **Save All** in the menu bar to save the OSB JMS process, as shown in [Figure 10–106](#).

Figure 10–106 Transport Window



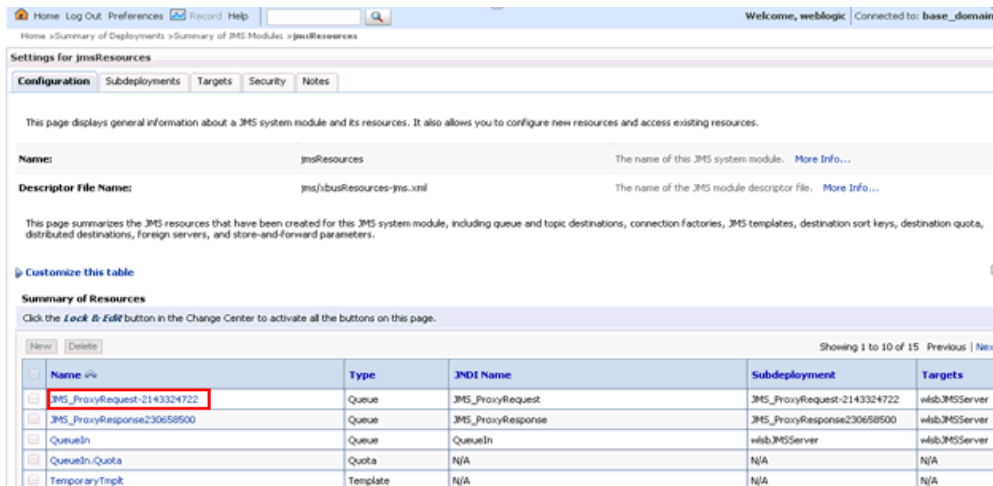
8. Deploy the OSB JMS outbound process. For more information, see [Section 10.1.3, "Deploying the OSB Outbound Process"](#) on page 10-17.
9. Once the process is deployed successfully, log on to the Oracle WLS Console.
10. In the Oracle WLS console, expand **Services**, click **Messaging**, select **JMS Modules**, and then click **jmsResources**, as shown in [Figure 10–107](#).

Figure 10–107 JMS Resources



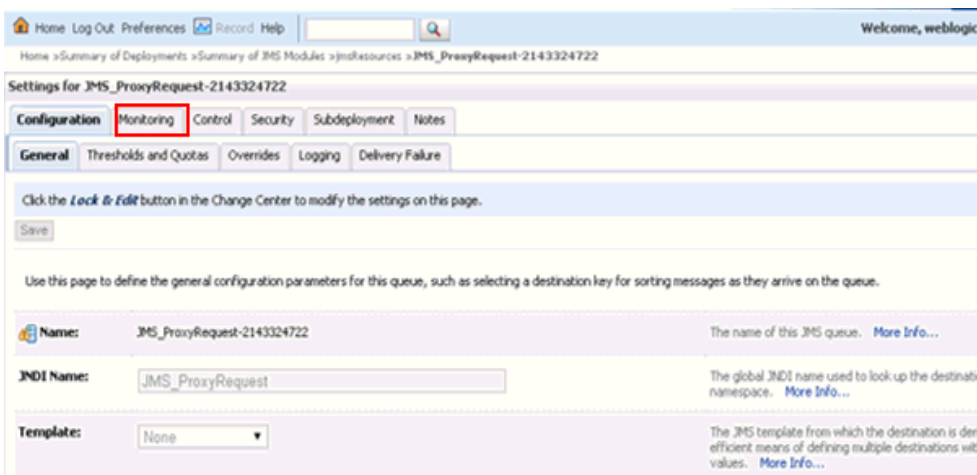
11. Click the appropriate request link (for example, JMS_ProxyRequest) as shown in [Figure 10–108](#).

Figure 10–108 JMS_ProxyRequest Link



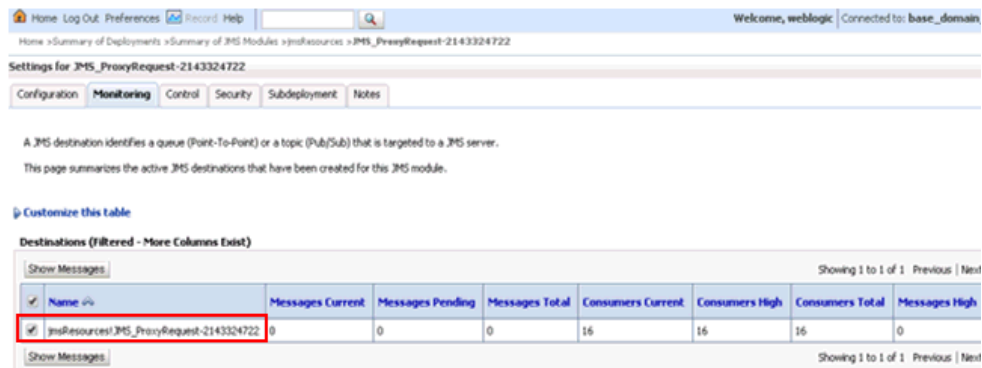
12. Click the Monitoring tab, as shown in Figure 10–109.

Figure 10–109 Monitoring Tab



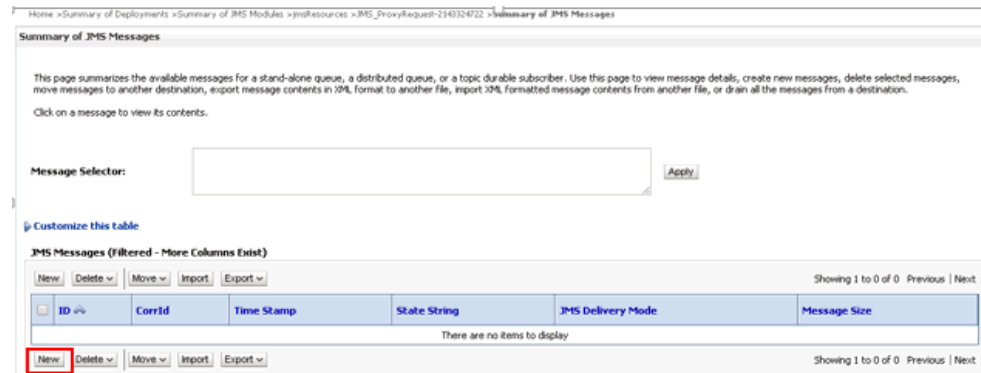
13. Select the check box and click the **Show Messages** button, as shown in Figure 10–110.

Figure 10–110 Show Messages Button



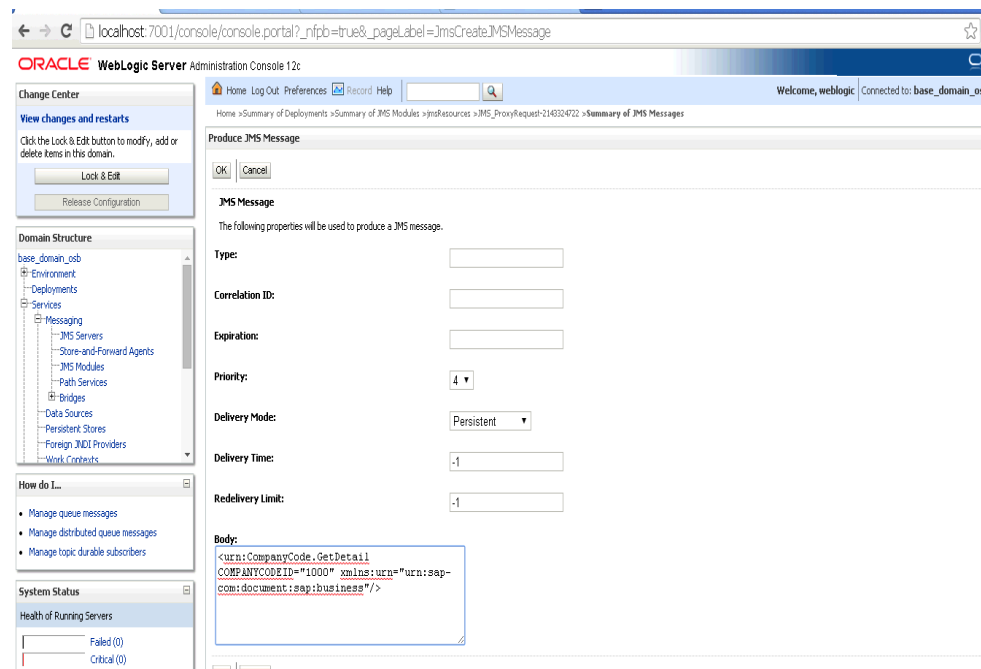
14. Click **New**, as shown in [Figure 10–111](#).

Figure 10–111 JMS Messages



15. Provide the input payload in the Body field and click **OK**, as shown in [Figure 10–112](#).

Figure 10–112 Input Payload



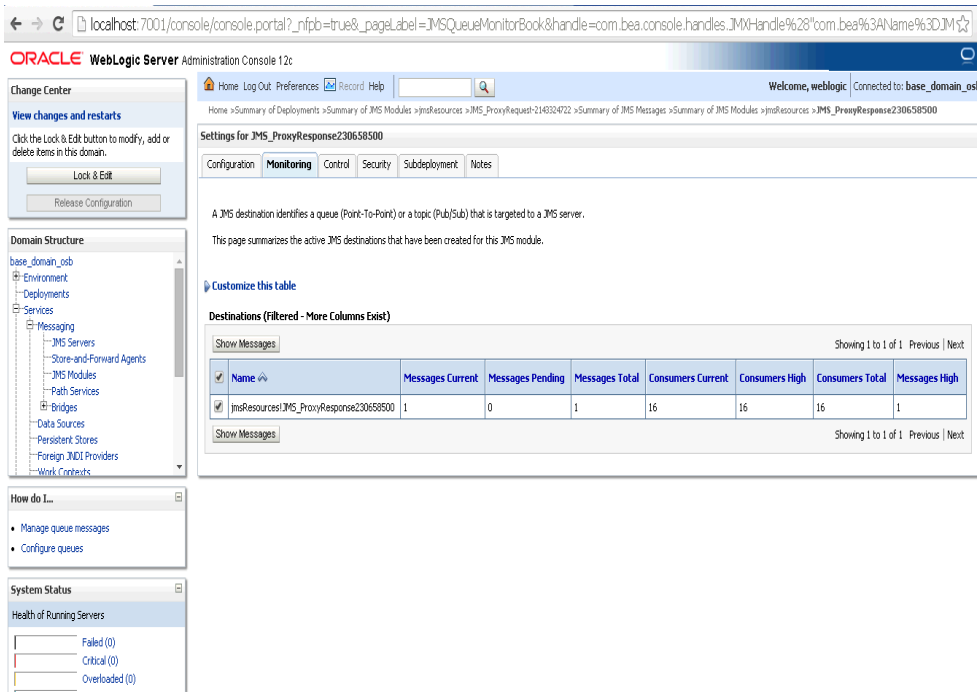
16. In the Oracle WLS console, expand **Services**, click **Messaging**, select **JMS Modules**, and then click **jmsResources**.

17. Click the appropriate response link (for example, **JMS_ProxyResponse**).

18. Click the **Monitoring** tab.

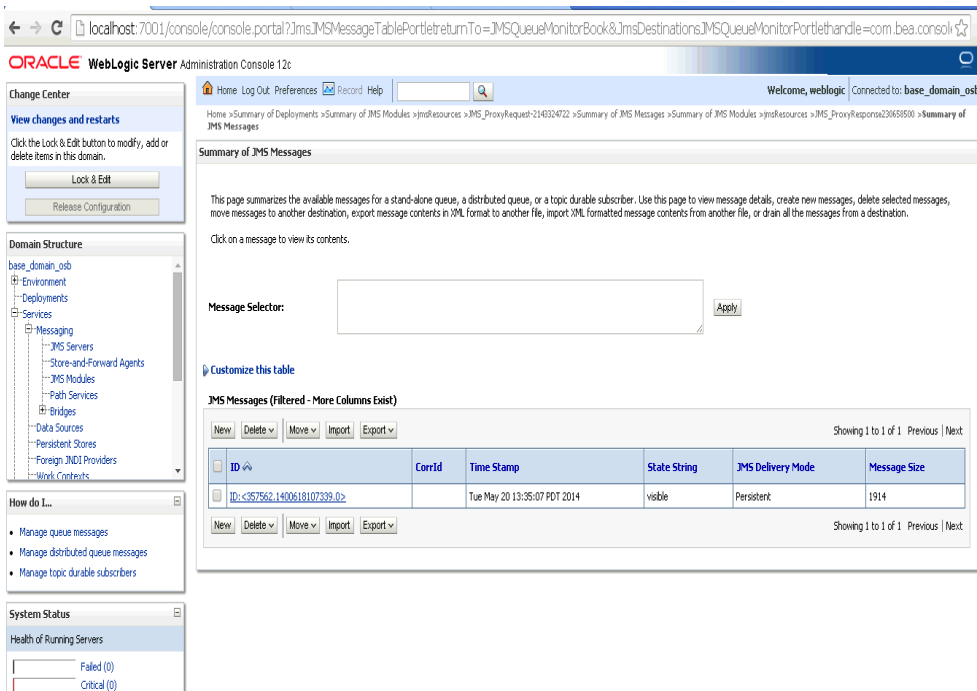
19. Select the check box and click **Show Messages**, as shown in [Figure 10–113](#).

Figure 10–113 Destination Messages



20. Click the ID link with the appropriate time and date, as shown in Figure 10–114.

Figure 10–114 Summary of JMS Messages Window



The response document is shown under the Text field, as shown in Figure 10–115.

Figure 10-115 Response Document

The screenshot shows the Oracle WebLogic Server Administration Console. The main content area displays the 'JMS Message Detail' for a specific message. The message ID is <367562.1400618107339.0>. The delivery mode is Persistent, and the timestamp is Tue May 20 13:35:07 PDT 2014. The priority is 4. The text area shows the following XML response:

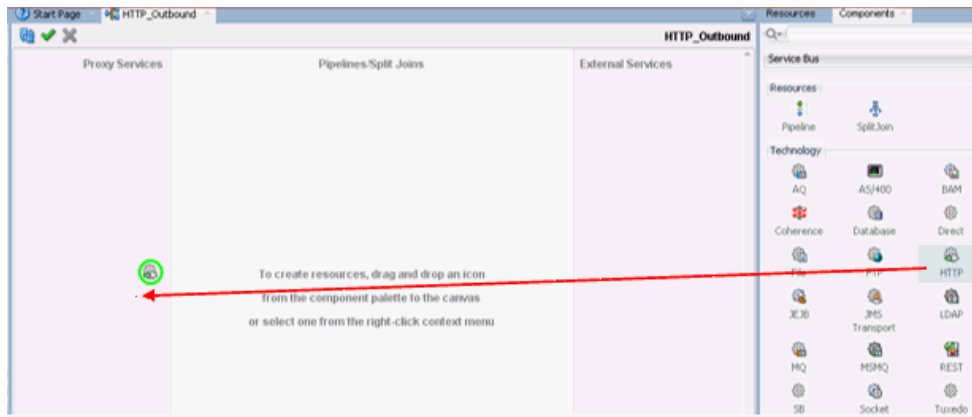
```
<?xml version="1.0" encoding="UTF-8"?>
<CompanyCode.GetDetail.Response xmlns="urn:sap-com:document:sap:business:response">
  <COMPANTCODE_ADDRESS>
    <ADDR_NO>0000000121</ADDR_NO>
    <FORMOFADDR>Firma</FORMOFADDR>
    <NAME>Ideas AG</NAME>
    <NAME_2>Martin Steiner, Kathrin Waither,</NAME_2>
```

10.7 Configuring an HTTP Outbound Process Using JDeveloper (J2CA Configuration)

This section describes how to configure HTTP Outbound process to your SAP R/3 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 6.4.1, "Generating WSDL for Request/Response Service"](#) on page 6-8.
2. Start the Oracle JDeveloper and create a Service Bus Application for OSB. For more information, see [Section 10.1.1, "Creating a Service Bus Application for OSB"](#) on page 10-2.
3. Create a Third Party Adapter Service Component. For more information, see [Section 7.3.2.1, "Configuring a Third Party Adapter Service Component"](#) on page 7-14.
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 10-116](#).

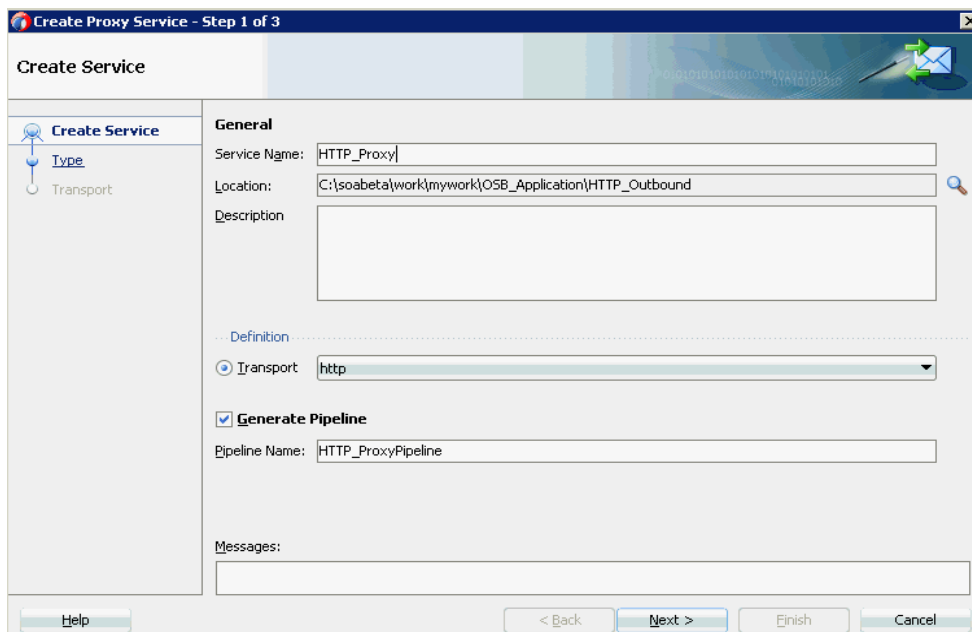
Figure 10–116 HTTP Component



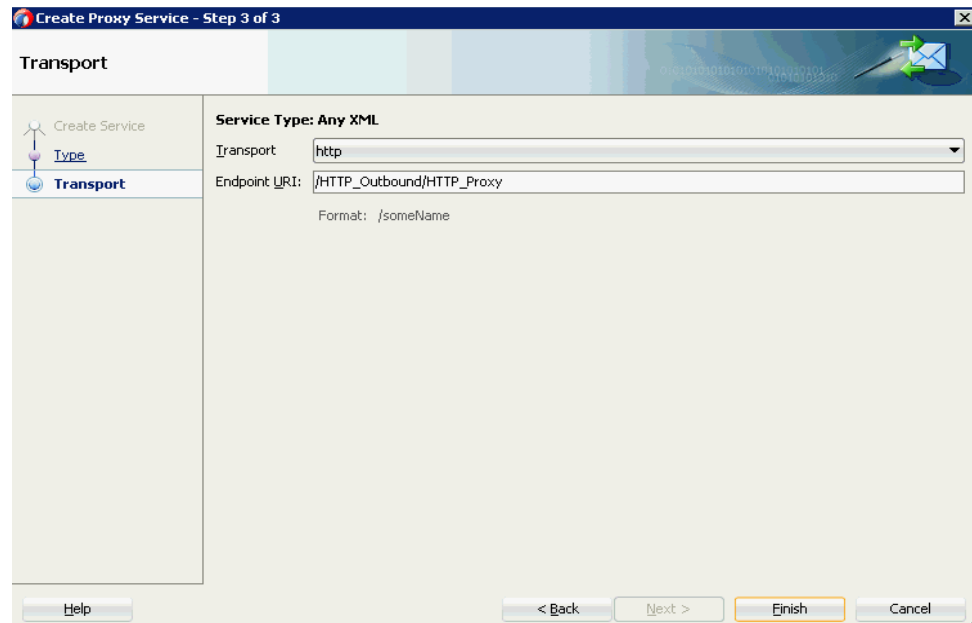
The Create Proxy Service dialog is displayed.

- 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 10–117](#).

Figure 10–117 Create Proxy Service Pane

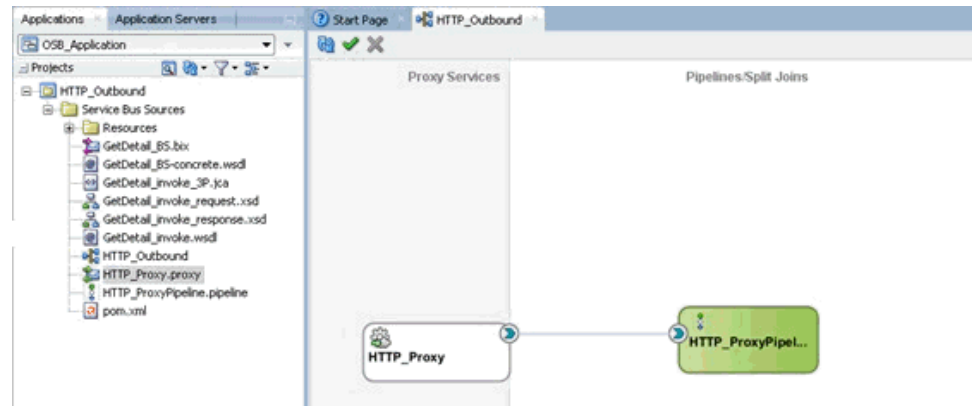


- d. In the displayed Type window, select **Any XML** and then click **Next**. The Transport window is displayed.
- e. Leave the default values and then click **Finish**, as shown in [Figure 10–118](#).

Figure 10–118 Transport Window

The HTTP Proxy service along with the pipeline is created and displayed.

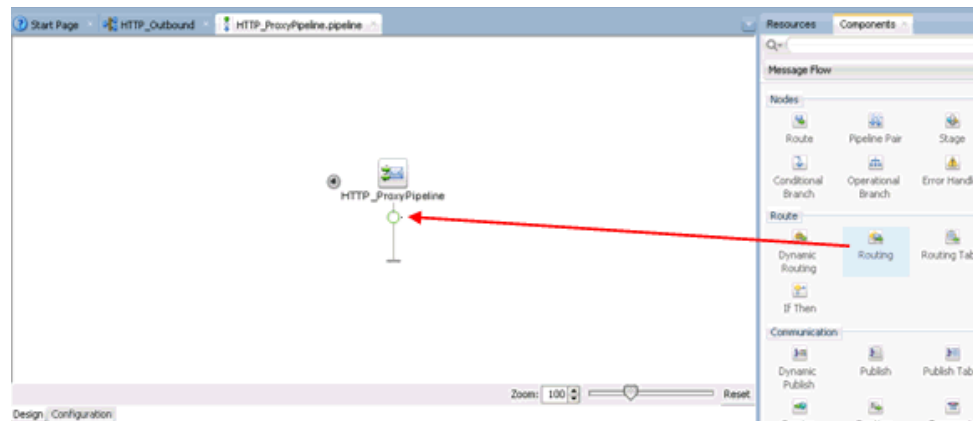
- f. Double-click the created pipeline (for example, HTTP_ProxyPipeline) in the Pipelines/Split Joins pane, as shown in [Figure 10–119](#).

Figure 10–119 Proxy Service

The Pipeline Configuration page is displayed.

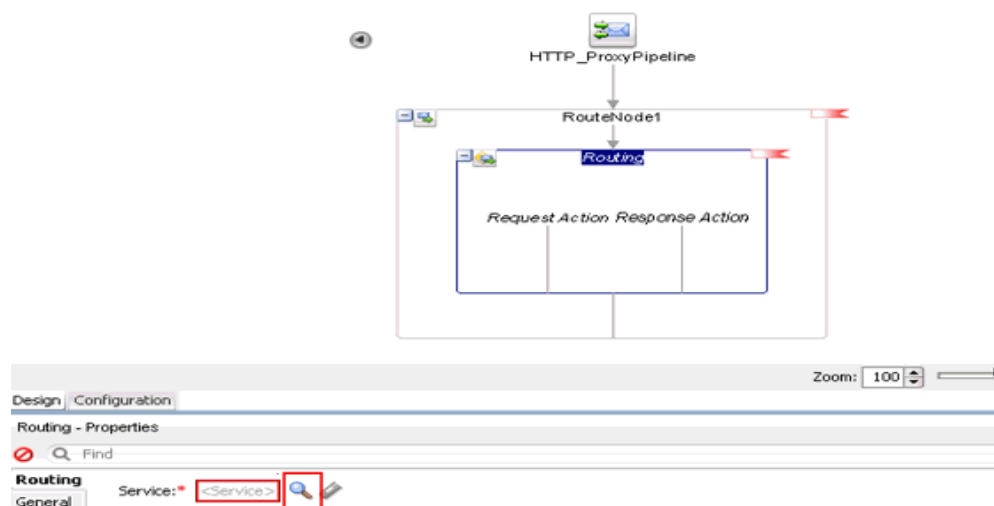
5. Configure the Routing Rules and proceed with the following steps:
 - a. Drag and drop the **Routing** component from the Route section to the area below the Pipeline (for example, HTTP_ProxyPipeline), as shown in [Figure 10–120](#).

Figure 10–120 Routing Component



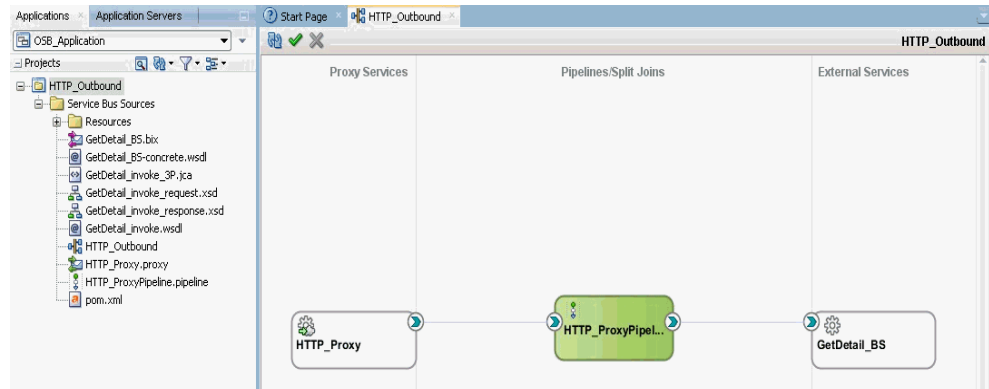
- b. In the Pipeline Configuration page, select **Routing** and click the browse icon to the right of the Service field in the Routing Properties pane, as shown in Figure 10–121.

Figure 10–121 Browse Service



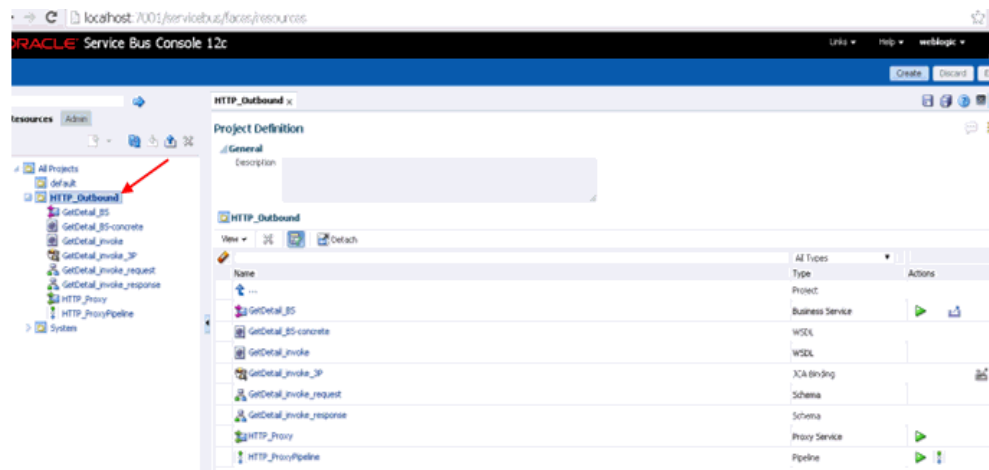
- c. In the displayed Resource Chooser window, select the WSDL-based Business service (for example, xxxxx_BS.bix) and click **OK**.
You are returned to the Pipeline configuration page.
- d. Save and Close the Pipeline configuration page.
You are returned to the composite editor window.
- e. Click **Save All** in the menu bar to save the OSB HTTP process, as shown in Figure 10–122.

Figure 10–122 Transport Window



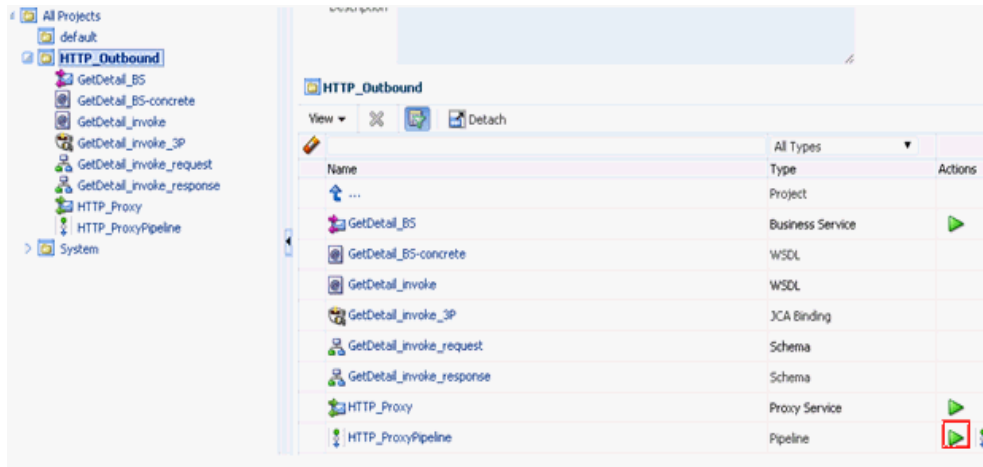
6. Deploy the OSB HTTP outbound process. For more information, see [Section 10.1.3, "Deploying the OSB Outbound Process"](#) on page 10-17.
7. Once the process is deployed successfully, log on to the Service Bus Console.
8. In the Service Bus console, click on the deployed HTTP Outbound project (for example, HTTP_Outbound), as shown in [Figure 10–123](#).

Figure 10–123 Service Bus Console



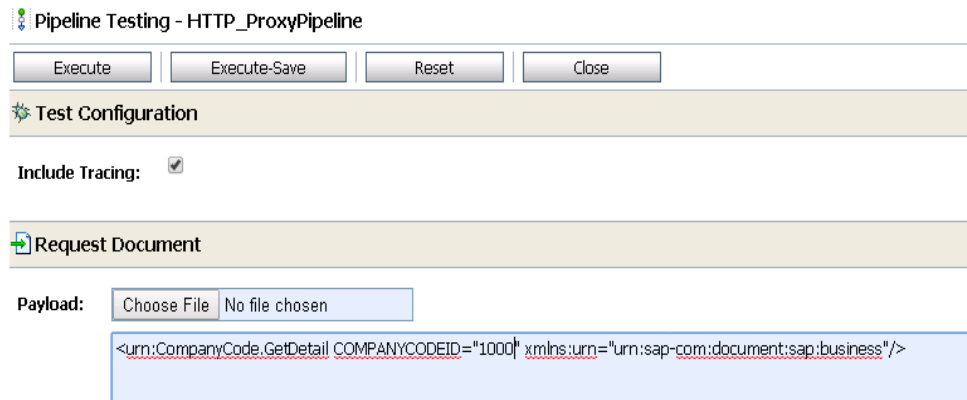
9. Click on the Test OSB Console icon for the created pipeline, as shown in [Figure 10–124](#).

Figure 10–124 Test OSB Console Icon

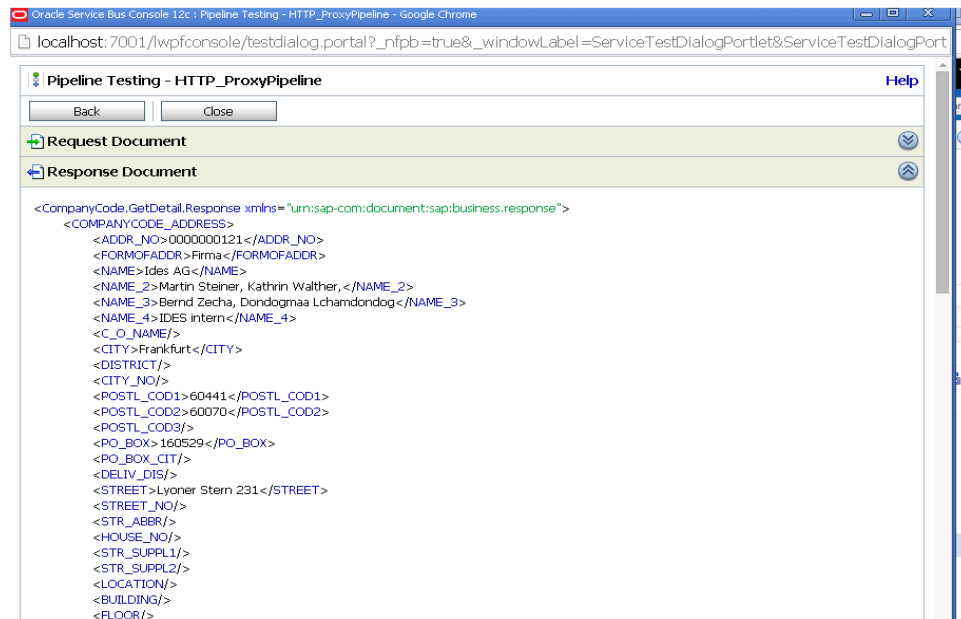


10. In the displayed Test OSB Console page, provide the input XML and click the **Execute** button, as shown as shown in [Figure 10–125](#).

Figure 10–125 Pipeline Testing



In the displayed Test OSB Console page, the response is received, as shown in [Figure 10–126](#).

Figure 10–126 Response Document

The screenshot shows the Oracle Service Bus Console interface. The title bar reads "Oracle Service Bus Console 12c: Pipeline Testing - HTTP_ProxyPipeline - Google Chrome". The address bar shows "localhost: 7001/lwpcfconsole/testdialog.portal?_nfpb=true&_windowLabel=ServiceTestDialogPortlet&ServiceTestDialogPort". The main content area is titled "Pipeline Testing - HTTP_ProxyPipeline" and contains a "Request Document" and a "Response Document" section. The "Response Document" section is expanded, showing the following XML content:

```
<CompanyCode.GetDetail.Response xmlns="urn:sap-com:document:sap:business:response">
  <COMPANYCODE_ADDRESS>
    <ADDR_NO>000000121</ADDR_NO>
    <FORMOFADDR>Firma</FORMOFADDR>
    <NAME>Ides AG</NAME>
    <NAME_2>Martin Steiner, Kathrin Walther,</NAME_2>
    <NAME_3>Bernd Zecha, Dondogmaa Lchamdondog</NAME_3>
    <NAME_4>IDES intern</NAME_4>
    <C_O_NAME/>
    <CITY>Frankfurt</CITY>
    <DISTRICT/>
    <CITY_NO/>
    <POSTL_COD1>60441</POSTL_COD1>
    <POSTL_COD2>60070</POSTL_COD2>
    <POSTL_COD3/>
    <PO_BOX>160529</PO_BOX>
    <PO_BOX_CTT/>
    <DELIV_DIS/>
    <STREET>Lyoner Stern 231</STREET>
    <STREET_NO/>
    <STR_ABBR/>
    <HOUSE_NO/>
    <STR_SUPPL1/>
    <STR_SUPPL2/>
    <LOCATION/>
    <BUILDING/>
    <FLOOR/>
  </COMPANYCODE_ADDRESS>
</CompanyCode.GetDetail.Response>
```


This chapter describes key features for the Oracle Application Adapter for SAP R/3. This chapter contains the following sections:

- [Section 11.1, "Configuring the Logging Feature"](#)
- [Section 11.2, "Configuring the Diagnosibility Feature"](#)
- [Section 11.3, "Configuring the SOA Debugging Feature"](#)
- [Section 11.4, "Configuring the Exception Filter"](#)
- [Section 11.5, "Credential Mapping for Oracle SOA Suite \(BPEL, Mediator, or BPM\)"](#)
- [Section 11.6, "Credential Mapping for Oracle Service Bus \(OSB\) Using JDeveloper"](#)
- [Section 11.7, "Filtering Unicode Characters From Text"](#)

11.1 Configuring the Logging Feature

In Oracle 12c (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 11.1.1, "Configuring Log File Management for the J2CA Connector Application"](#)
- [Section 11.1.2, "Configuring Log File Management for Business Services Engine \(BSE\)"](#)

11.1.1 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.MySAP' 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.MySAP' 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.MySAP' 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.MySAP'
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.MySAP' 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:

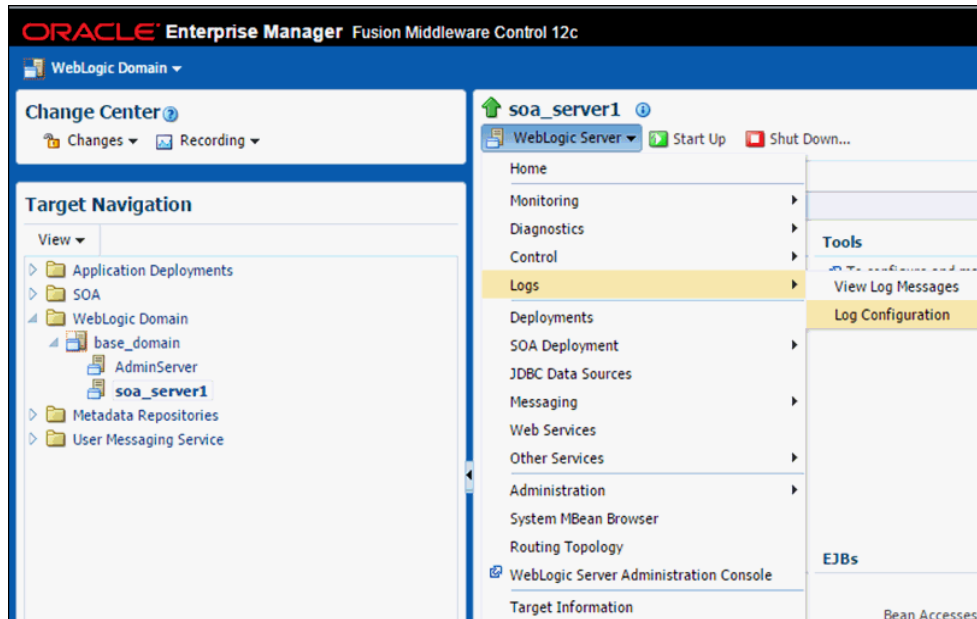
```
http://host name:port/em
```

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.

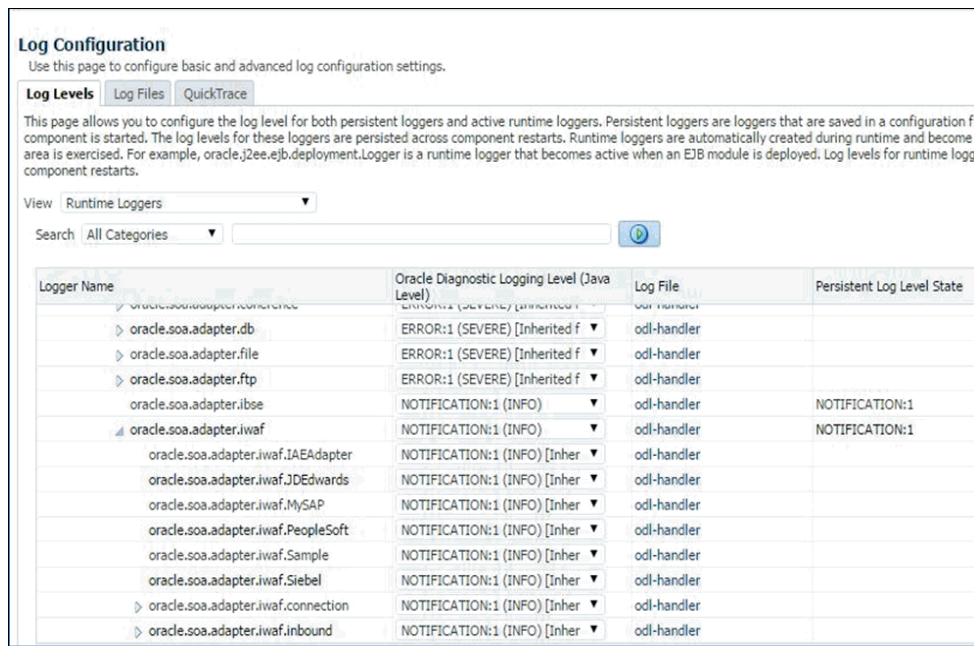
4. Under the Target Navigation pane, click **Weblogic Domain**, select **Domain Created**, and click the appropriate server (Managed Server or Integrated Server).
5. In the Server pane, expand WebLogic Server, select **Logs**, and then click **Log Configuration**, as shown in Figure 11–1.

Figure 11–1 Log Configuration Option



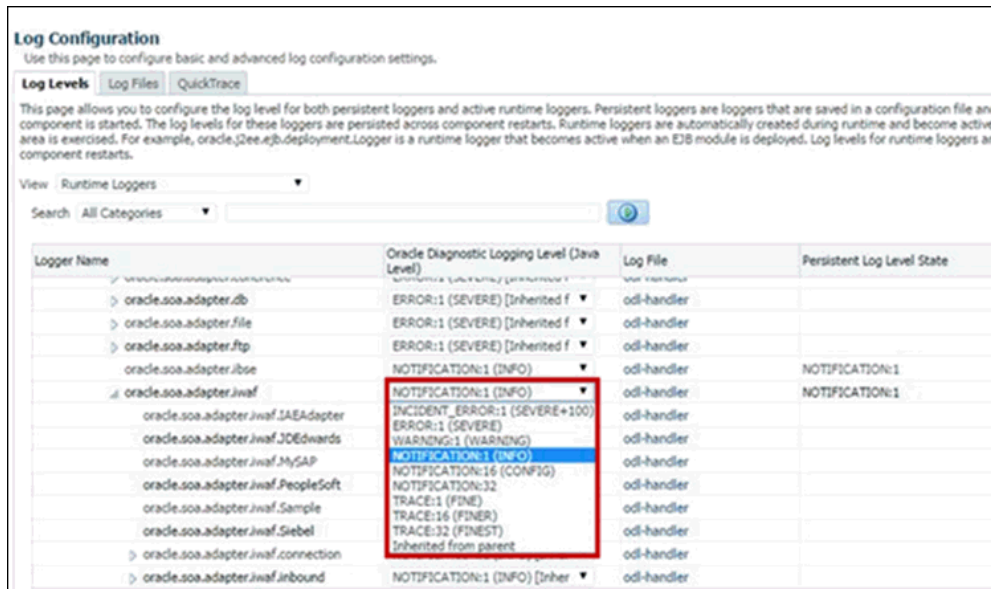
6. Under the **Log Levels** tab, expand the Oracle root logger until `oracle.soa.adapter.iwaf` is visible, as shown in Figure 11–2.

Figure 11–2 Log Levels Tab



- 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 11-3.

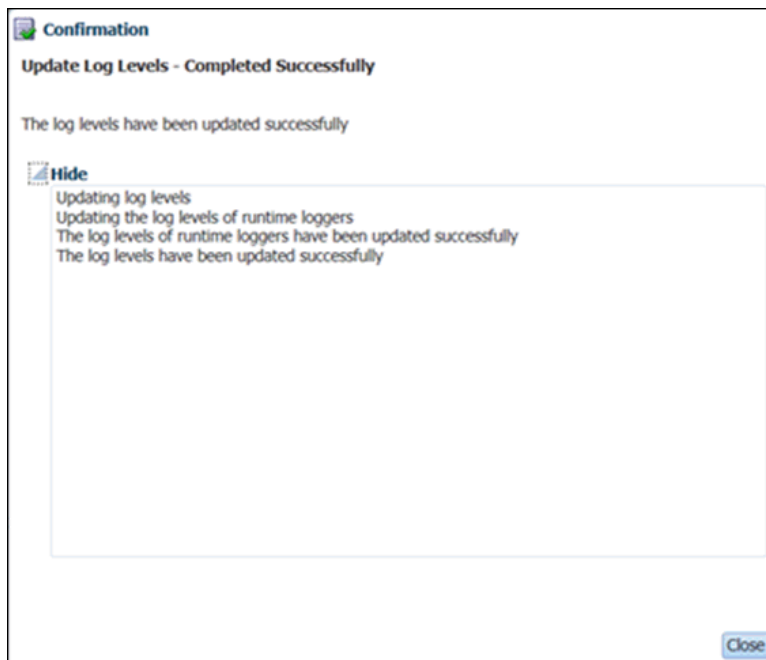
Figure 11-3 Oracle Diagnostic Logging Level (Java Level) Column



- Click **Apply**.

A confirmation message appears, indicating that the update was completed successfully, as shown in Figure 11-4.

Figure 11-4 Confirmation Message



- 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 11–1 Log Level Property Settings

Log Level Set in Em Console	Log Level Updated in <code>{server-name}-diagnostic.log</code>
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 11–5](#).

Figure 11–5 Log Configuration Pane

Log Configuration
Use this page to configure basic and advanced log configuration settings.

Log Levels | **Log Files** | QuickTrace

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.

Create... Create Like... Edit Configuration... View Configuration... Delete Configuration...

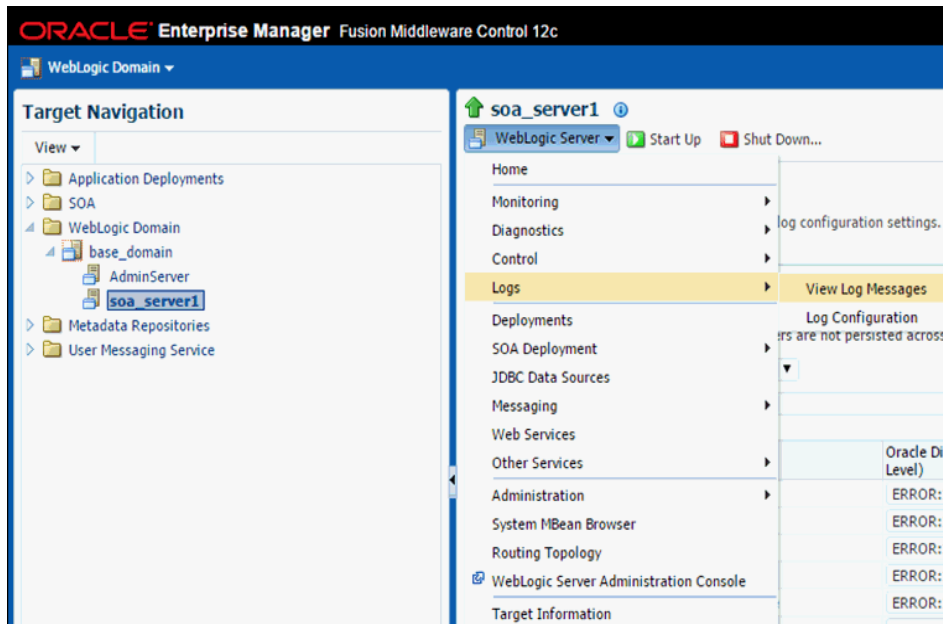
Handler Name	Log Path	Log File Format
em-log-handler	<code>\${domain.home}/servers/\${weblogic.Name}/sysman/log/emoms.log</code>	Oracle Diagnostics Logging
em-trc-handler	<code>\${domain.home}/servers/\${weblogic.Name}/sysman/log/emoms.trc</code>	Oracle Diagnostics Logging
odl-handler	<code>\${domain.home}/servers/\${weblogic.Name}/logs/\${weblogic.Name}-diagnostic.log</code>	Oracle Diagnostics Logging
owsm-message-ha...	<code>\${domain.home}/servers/\${weblogic.Name}/logs/owsm/msglogging/diagnostic.log</code>	Oracle Diagnostics Logging
soa-tracking-trc-h...	<code>\${domain.home}/servers/\${weblogic.Name}/logs/\${weblogic.Name}-soa-tracking.trc</code>	Oracle Diagnostics Logging

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
```

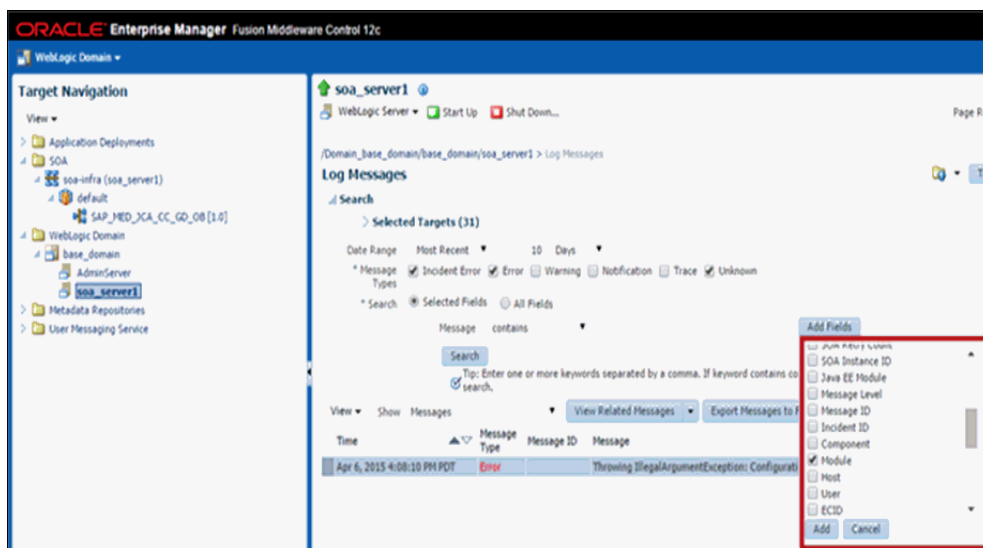
- In the Server pane, display the log messages in the em console by clicking the **WebLogic Server** drop-down list, selecting **Logs**, and then clicking **View Log Messages**, as shown in [Figure 11-6](#).

Figure 11-6 View Log Messages Option



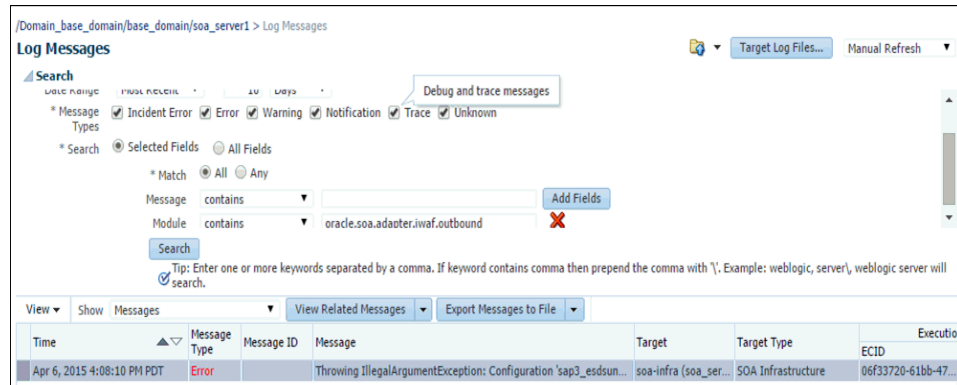
- On the Log Messages pane, complete the required search criteria. You can also add the **Module** field to the search criteria, which contains the name of the logger of interest, as shown in [Figure 11-7](#).

Figure 11-7 Module Field



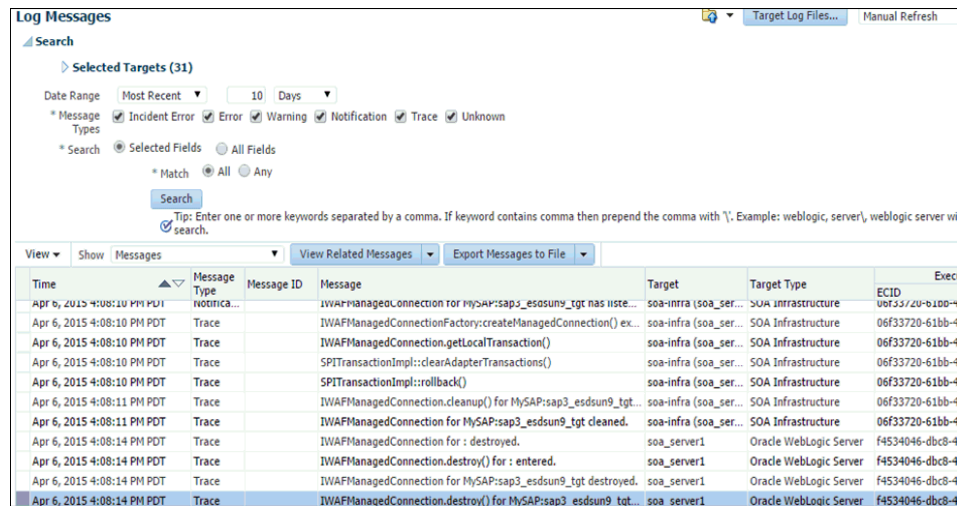
- Click **Add**.
- In the Module field, enter the name of the logger of interest, and if required, select the additional Message Types (Warning, Notification, Trace, and so on) and then click **Search**, as shown in [Figure 11-8](#).

Figure 11–8 Log Messages Pane



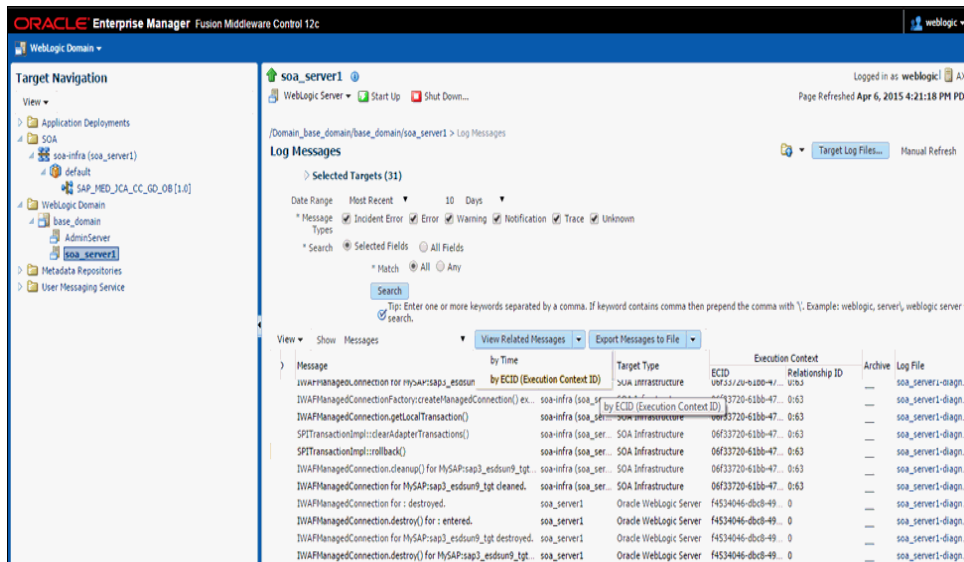
The messages from the specified logger are displayed in the table below the search criteria, as shown in Figure 11–9.

Figure 11–9 Logger Messages



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 11–10.

Figure 11–10 ECID Option



Details are displayed, as shown in Figure 11–11.

Figure 11–11 Message Details

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...

Message Level	32	Java EE	soa-infra
SRC_CLASS	com.ibm.afjca.spi.SPITransactionImpl	Application	
WebService Port	execute_pt	Relationship ID	0:63
SCA Entity ID	340004	Component	soa_server1
SOA Retry Count	SAP_MED_JCA_CC_GD_OB:1.0	Module	oracle.soa.adapter.iwaf.outbound
SOA Instance ID	320003	Host	AXTST168

11.1.2 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
```

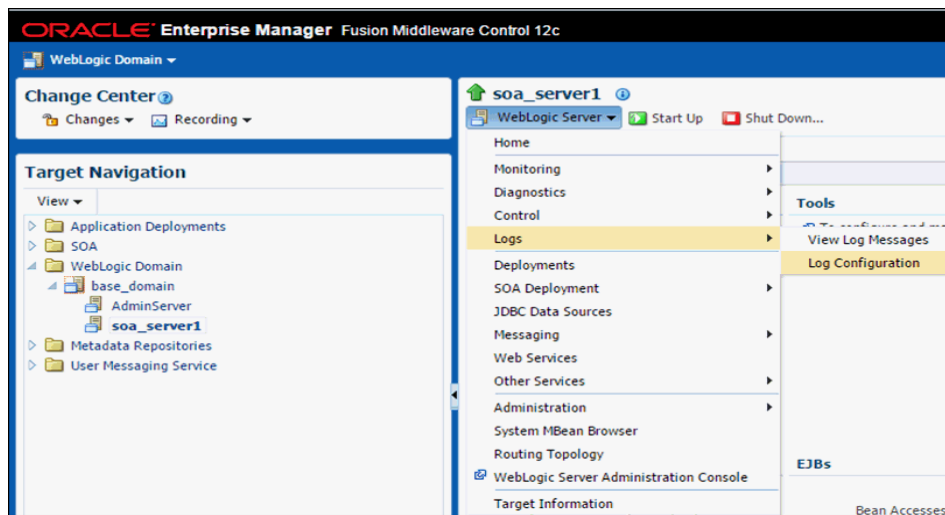
The following syntax sets the logging level of all the loggers under oracle.soa.adapter.ibse to NOTIFICATION:1 (INFO) which is the default setting level by oracle.

```
<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:

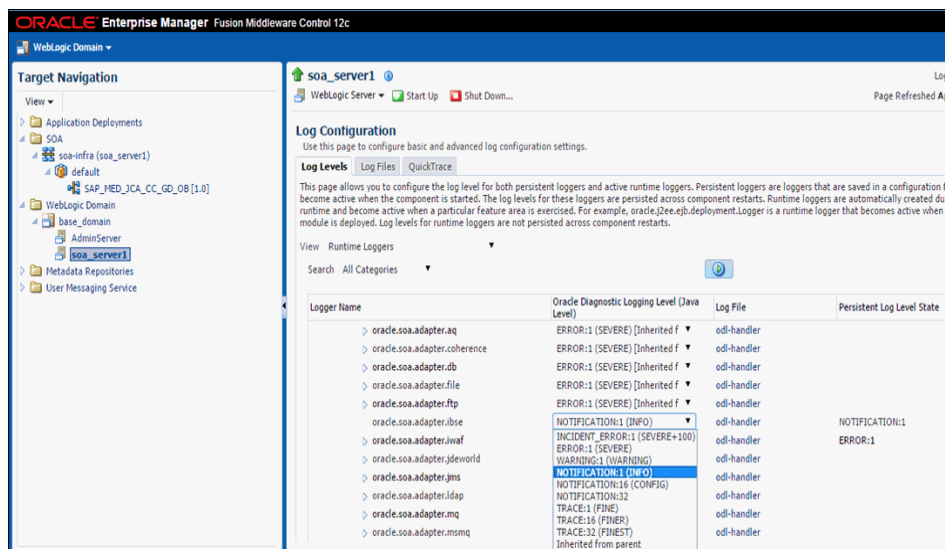
1. Repeat steps 1 through 5, as described in Section 11.1.1, "Configuring Log File Management for the J2CA Connector Application".
2. Under the **Log Levels** tab, expand the Oracle root logger until the oracle.soa.adapter.ibse Logger name is visible, as shown in Figure 11–12.

Figure 11–12 Log Levels Tab



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 11–13.

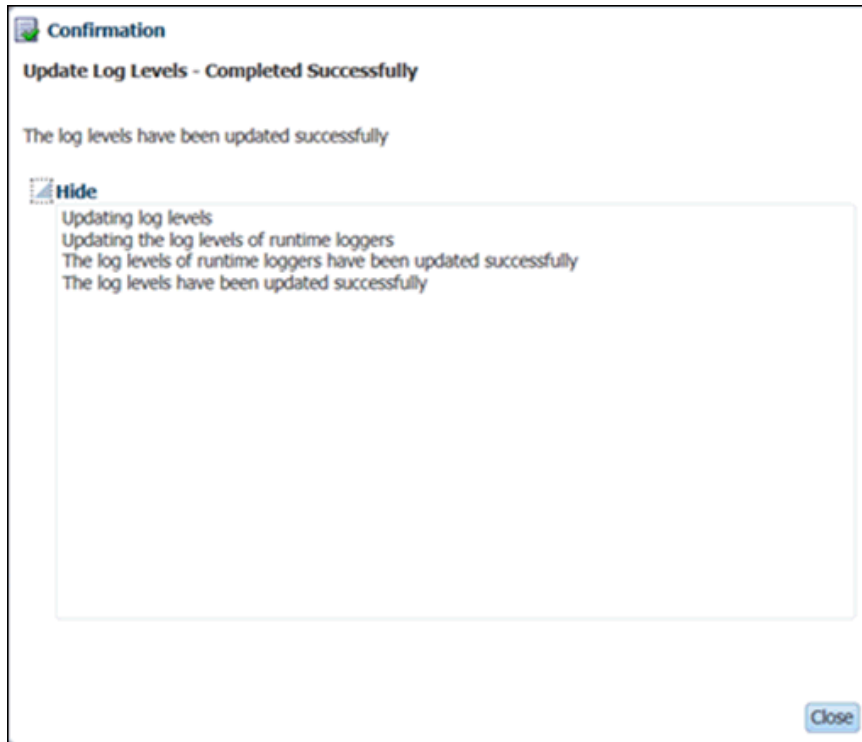
Figure 11–13 Oracle Diagnostic Logging Level Column



4. Click **Apply**.

A confirmation message appears, indicating that the update was completed successfully, as shown in Figure 11–14.

Figure 11–14 Confirmation Message



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 11–2 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 11.1.1, "Configuring Log File Management for the J2CA Connector Application"](#).

11.2 Configuring the Diagnosability Feature

This section describes how to configure and use the Diagnosability feature for the Oracle Fusion Middleware Application Adapters for Oracle WebLogic Server. It contains the following topic:

- [Section 11.2.1, "Supporting Protocols"](#)

The Diagnosability 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 Diagnosability 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 11–15](#).

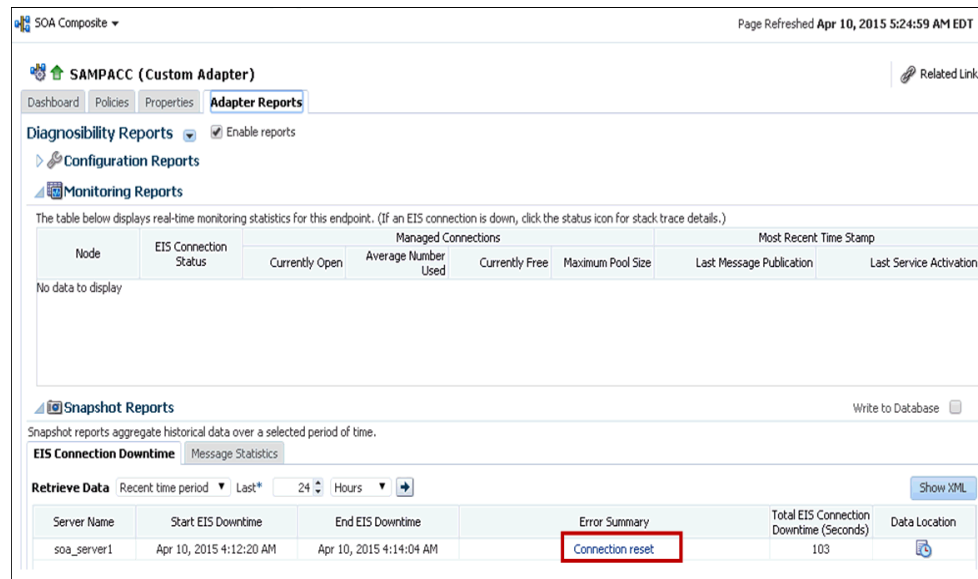
Figure 11–15 Services and References Section

The screenshot shows the Oracle WebLogic Server Administrative Console interface. The top navigation bar includes 'SOA Composite' and 'Page Refreshed Apr 23, 2015 12:29:57 PM PDT'. Below the navigation bar, there are tabs for 'Dashboard', 'Composite Definition', 'Flow Instances', 'Unit Tests', and 'Policies'. The 'Services and References' section is expanded, showing a table with the following data:

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 11–16](#).

Figure 11–16 Stack Trace



11.2.1 Supporting Protocols

This section describes the supporting protocols for the Oracle Fusion Middleware Application Adapter for SAP R/3 (SAP JCo 3.0), and consists of the following topic:

- [Section 11.2.1.1, "Oracle Fusion Middleware Application Adapter for SAP R/3 Endpoint Status"](#)

11.2.1.1 Oracle Fusion Middleware Application Adapter for SAP R/3 Endpoint Status

The Oracle Fusion Middleware Application Adapter for SAP R/3 (inbound) uses the SAP API protocols.

The Oracle Fusion Middleware Application Adapter for SAP R/3 polls the server for an event. The adapter then alerts the Oracle Adapter Framework that the server is either:

- Down when it determines that it is not available.
- Running when communication is available.

11.3 Configuring the SOA Debugging Feature

This section describes how to configure and use the SOA Debugging feature for the Oracle Application Adapter for SAP R/3. It contains the following topics:

- [Section 11.3.1, "Guidelines for Using the SOA Debugger"](#)
- [Section 11.3.2, "Prerequisite"](#)
- [Section 11.3.3, "Debugging a BPEL Process in Oracle JDeveloper"](#)
- [Section 11.3.4, "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.

11.3.1 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.

11.3.2 Prerequisite

Ensure that the *IntegratedWebLogicServer* domain and a BPEL process are already created in Oracle JDeveloper.

11.3.3 Debugging a BPEL Process in Oracle JDeveloper

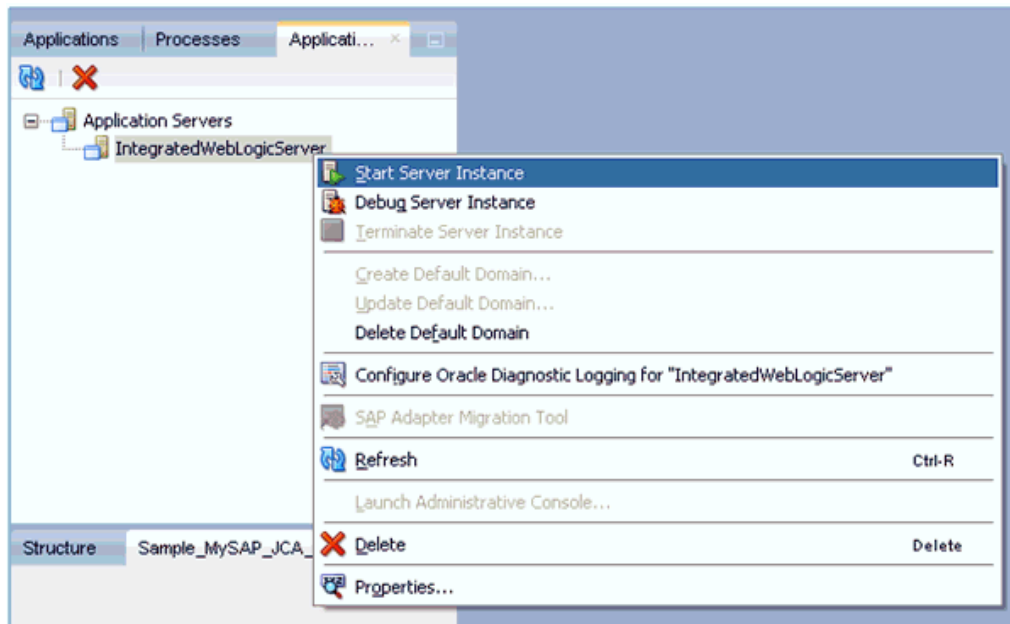
This section describes how to debug a BPEL process in Oracle JDeveloper. It contains the following topics:

- [Section 11.3.3.1, "Debugging an Outbound BPEL Process in Oracle JDeveloper"](#)
- [Section 11.3.3.2, "Debugging an Inbound BPEL Process in Oracle JDeveloper"](#)

11.3.3.1 Debugging an Outbound BPEL Process in Oracle JDeveloper

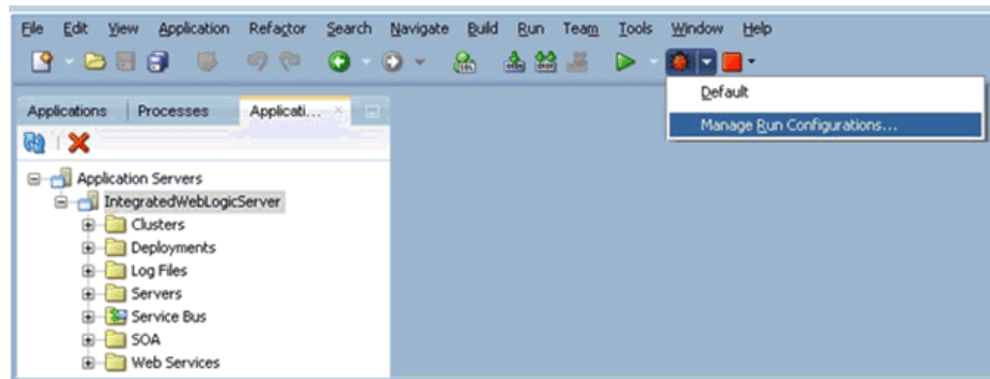
1. Open Oracle JDeveloper.
2. Start the *IntegratedWeblogicServer* domain.
 - a. Click the **Application Servers** tab in the left pane.
 - b. Under the Application Servers node, right-click **IntegratedWeblogicServer** and select **Start Server Instance** from the context menu, as shown in [Figure 11-17](#).

Figure 11–17 Start Server Instance

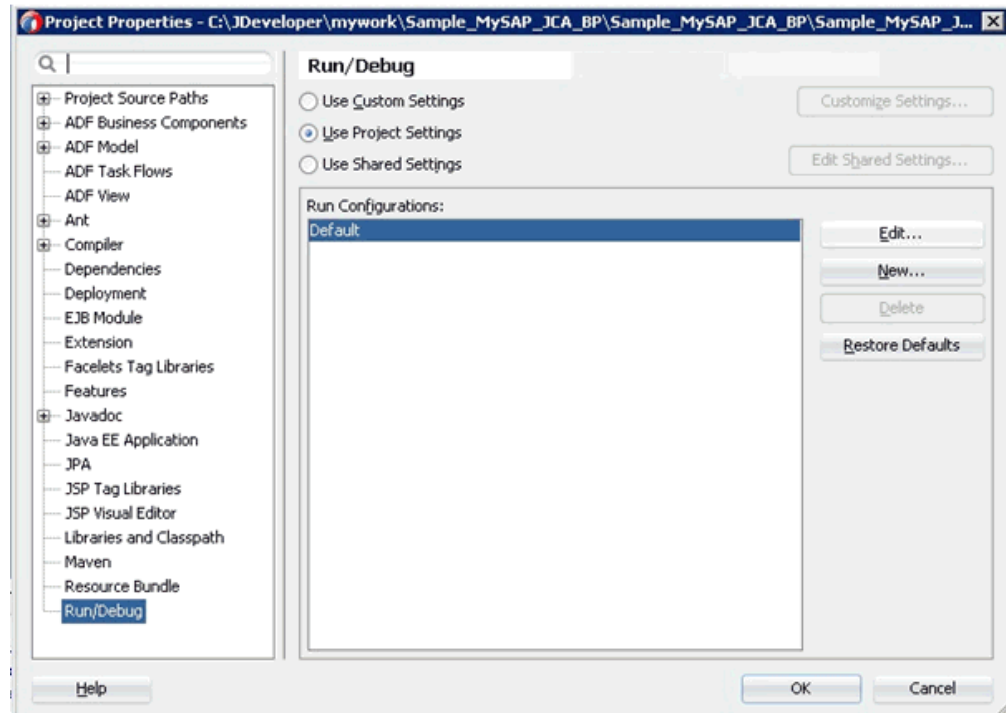


3. Set the Debugging environment.
 - a. Click the down arrow next to the Debug icon and select **Manage Run Configurations** from the context menu, as shown in [Figure 11–18](#).

Figure 11–18 Manage Run Configurations



- b. Or, right-click the project and select **Project Properties**.
The Project Properties dialog is displayed, as shown in [Figure 11–19](#).

Figure 11–19 Project Properties Dialog

- 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 11–20](#).

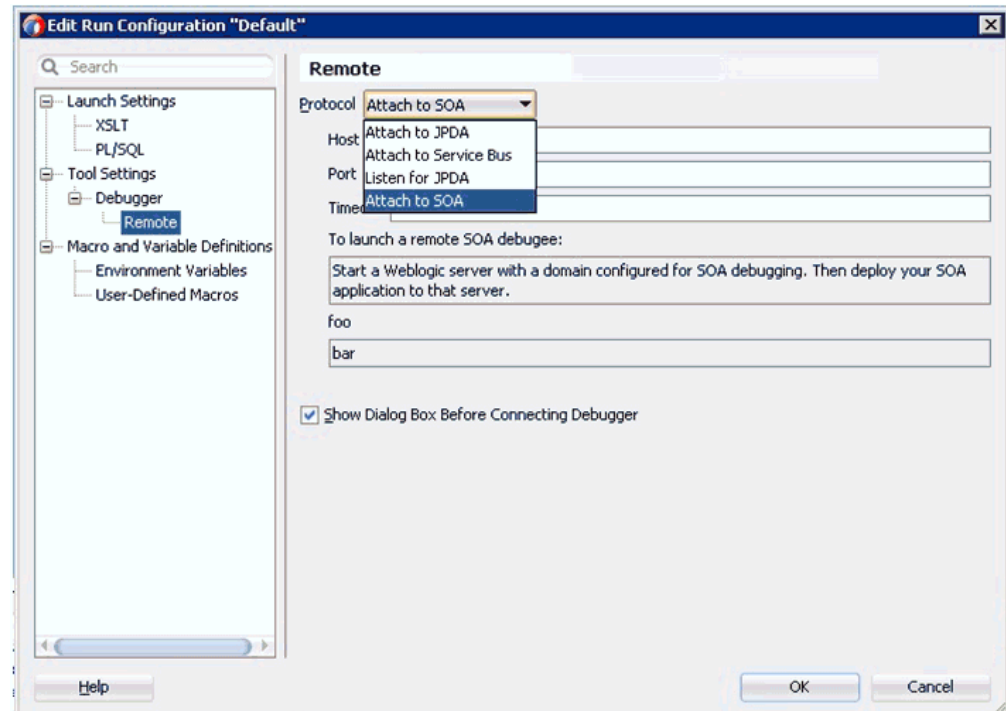
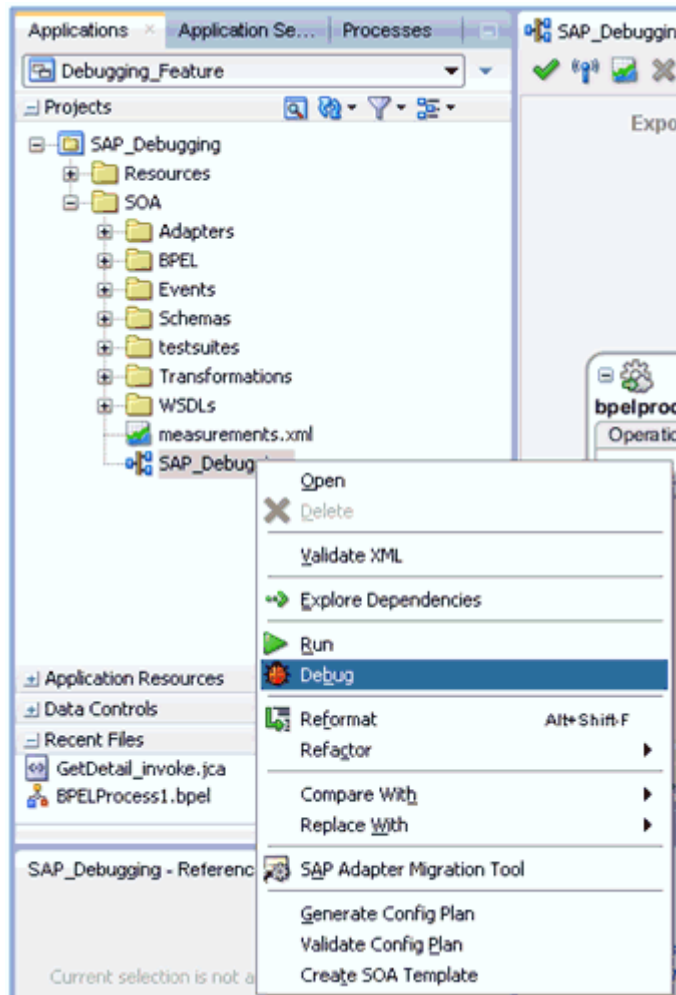
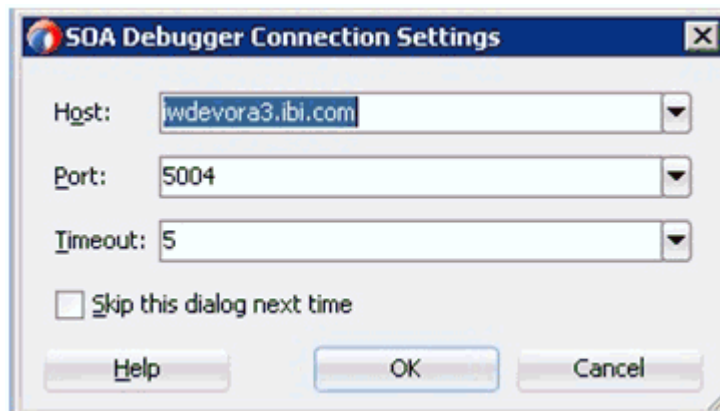
Figure 11–20 Edit Run Configuration Dialog

Figure 11–21 Select BPEL Process to Debug

The SOA Debugger Connection Settings dialog is displayed, as shown in [Figure 11–22](#).

Figure 11–22 SOA Debugger Connection Settings Dialog

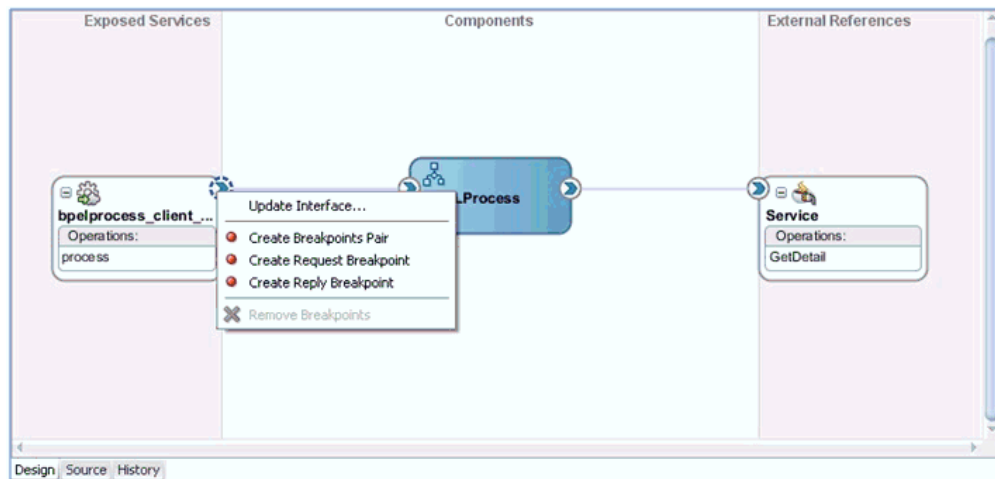
- b. Click OK.

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 11-23](#).

Figure 11-23 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.

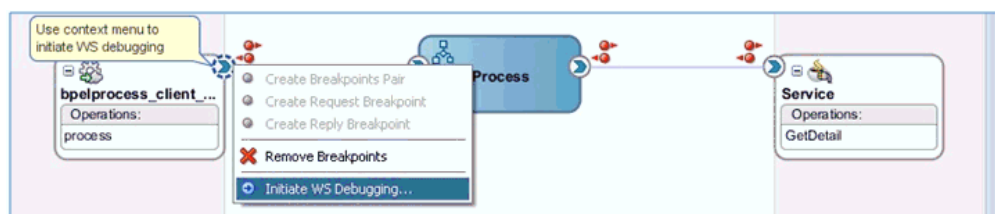
Create Request Breakpoint - Set this Breakpoint type for a request (outbound) interaction. This is useful for scenarios in which only the request is important.

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.

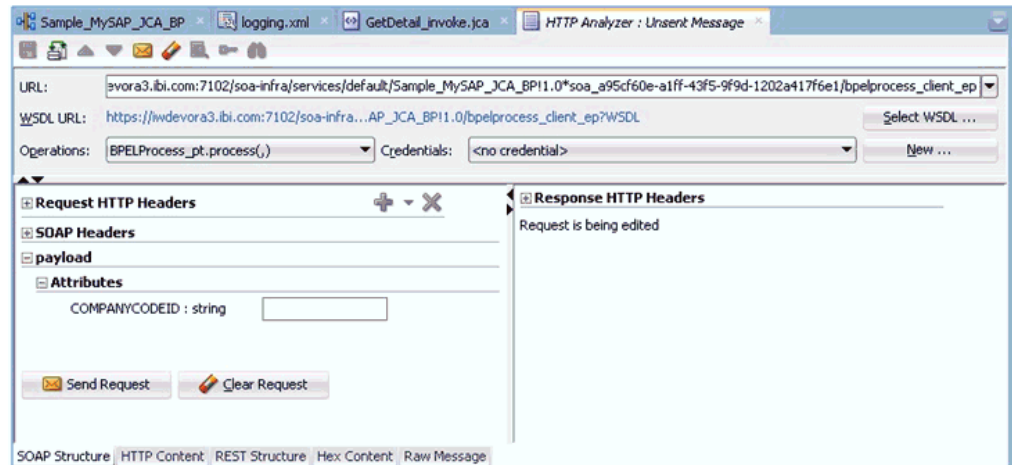
- b. Once the Breakpoints are set, right-click the right handle and select **Initiate WS Debugging**, as shown in [Figure 11-24](#).

Figure 11-24 Initiate WS Debugging



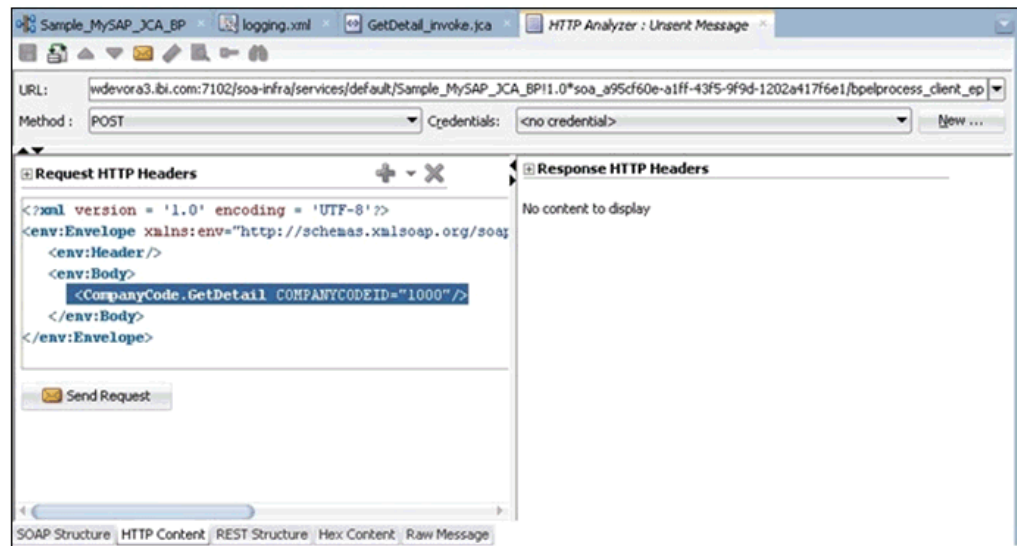
The HTTP Analyzer dialog is displayed, as shown in [Figure 11-25](#).

Figure 11–25 HTTP Analyzer Dialog



- c. Select HTTP content from the below tab. Now, copy and paste the payload into the body, as shown in [Figure 11–26](#).

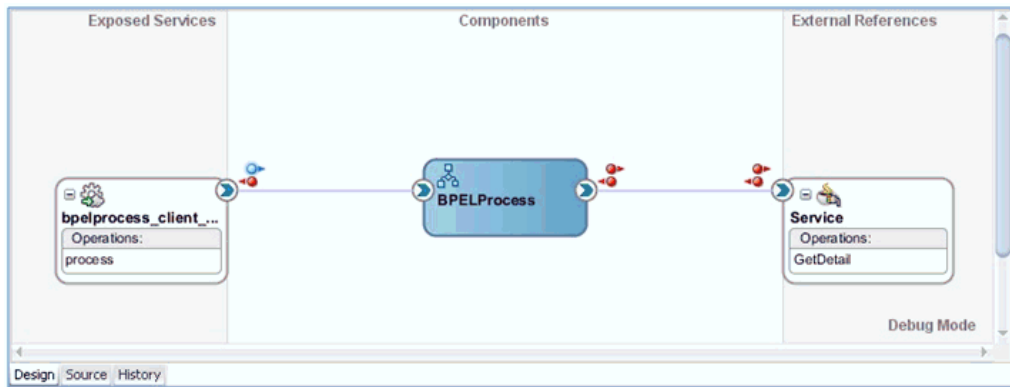
Figure 11–26 Select HTTP Content



- d. Click **Send Request**.

The BPEL Process stops at the designated Breakpoint and blinks in a blue color, as shown in [Figure 11–27](#).






Figure 11–27 BPEL Process Stopped at Breakpoint



- e. Use the available Step options to step through the Debugging process, as shown in [Figure 11–28](#).

Figure 11–28 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 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 SAP R/3, see [Figure 11–29](#).

Figure 11–29 Header Information

Name	Value	Decla...	Type	Hex V...	Address	ID
(x) completionPersistPolicy	null		osb.Variable			
(x) euid	148f5c57-bd1a-4571-81b0-2f9e2c29ccc2...		osb.Variable			
(x) masterConversationId	null		osb.Variable			
(x) MESH_METRICS	null		osb.Variable			
(x) nativePayload			osb.Variable			
(x) replyToPortType	null		osb.Variable			
(x) transaction	Name=[EJB com.collaxa.cube.engine.ejb.l...		osb.Variable			
(x) transport.http.remoteAddress	172.19.25.18		osb.Variable			
(x) XID	BEA1-0AD7677B5C6B9E472B1E(1799957...		osb.Variable			

A sample response payload for Oracle Application Adapter for SAP R/3 is shown in Figure 11–30.

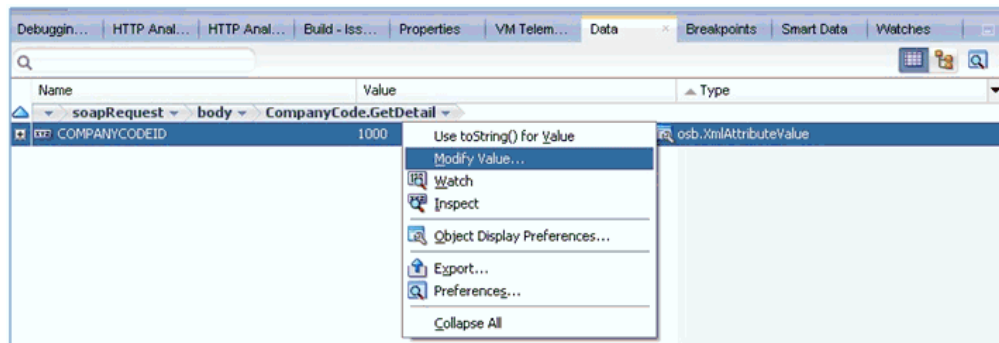
Figure 11–30 Sample Response Payload

```
<CompanyCode.GetDetail.Response xmlns="urn:sap-com:document:sap:business:response"
>
  <COMPANYCODE_ADDRESS>
    <ADDR_NO>0000000121</ADDR_NO>
    <FORMOFADDR>Firma</FORMOFADDR>
    <NAME>Ides AG</NAME>
    <NAME_2>Martin Steiner, Kathrin Walther,</NAME_2>
    <NAME_3>Bernd Zecha, Dondogmaa Lchamdondog</NAME_3>
    <NAME_4>IDES intern</NAME_4>
    <C_O_NAME/>
    <CITY>Frankfurt</CITY>
    <DISTRICT/>
    <CITY_NO/>
    <POSTL_COD1>60441</POSTL_COD1>
    <POSTL_COD2>60070</POSTL_COD2>
    <POSTL_COD3/>
    <PO_BOX>160529</PO_BOX>
    <PO_BOX_CITY/>
    <DELIV_DIS/>
    <STREET>Lyoner Stern 231</STREET>
    <STREET_NO/>
    <STR_ABBR/>
    <HOUSE_NO/>
    <STR_SUPPL1/>
    <STR_SUPPL2/>
    <LOCATION/>
    <BUILDING/>
    <FLOOR/>
    <ROOM_NO/>
    <COUNTRY>DE</COUNTRY>
    <LANGU>D</LANGU>
    <REGION>06</REGION>
    <SORT1>IDES</SORT1>
    <SORT2/>
    <TIME_ZONE>CET</TIME_ZONE>
    <TAXCURCODE/>
    <ADR_NOTES/>
    <COMM_TYPE/>
    <TEL1_NUMBER/>
    <TEL1_EXT/>
    <FAX_NUMBER/>
    <FAX_EXTENS/>
  </COMPANYCODE_ADDRESS>
  <COMPANYCODE_DETAIL>
    <COMP_CODE>1000</COMP_CODE>
    <COMP_NAME>IDES AG</COMP_NAME>
    <CITY>Frankfurt</CITY>
    <COUNTRY>DE</COUNTRY>
    <CURRENCY>EUR</CURRENCY>
    <LANGU>D</LANGU>
    <CHRT_ACCTS>INT</CHRT_ACCTS>
    <FY_VARIANT>K4</FY_VARIANT>
    <VAT_REG_NO>DE129456789</VAT_REG_NO>
    <COMPANY>001000</COMPANY>
    <ADDR_NO>0000000121</ADDR_NO>
    <COUNTRY_ISO>DE</COUNTRY_ISO>
    <CURRENCY_ISO>EUR</CURRENCY_ISO>
    <LANGU_ISO>DE</LANGU_ISO>
  </COMPANYCODE_DETAIL>
  <RETURN>
    <TYPE/>
    <CODE/>
    <MESSAGE/>
    <LOG_NO/>
    <LOG_MSG_NO>000000</LOG_MSG_NO>
  </RETURN>
</CompanyCode.GetDetail.Response>
```

Note: The payload display is limited to the screen size as shown in [Figure 11–30](#). However, the Breakpoint at the BPEL process displays the complete payload, allows scrolling and viewing all elements of the payload.

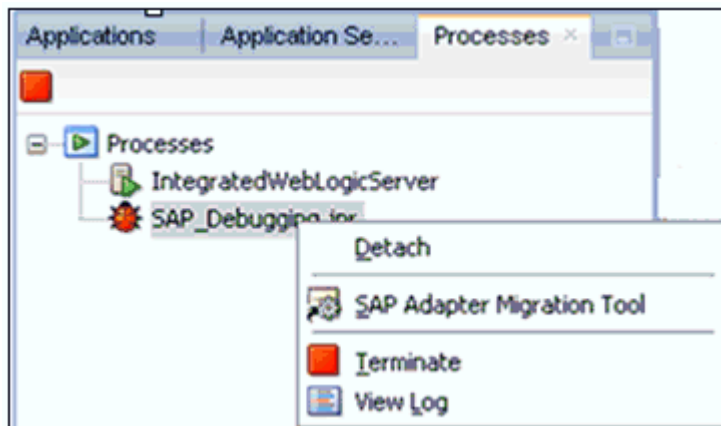
8. Modify the Request payload content.
 - a. Expand the SOAP request, select the field to modify, right-click and select **Modify Value** from the context menu, as shown in [Figure 11–31](#).

Figure 11–31 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 11–32](#).

Figure 11–32 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.

11.3.3.2 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**.

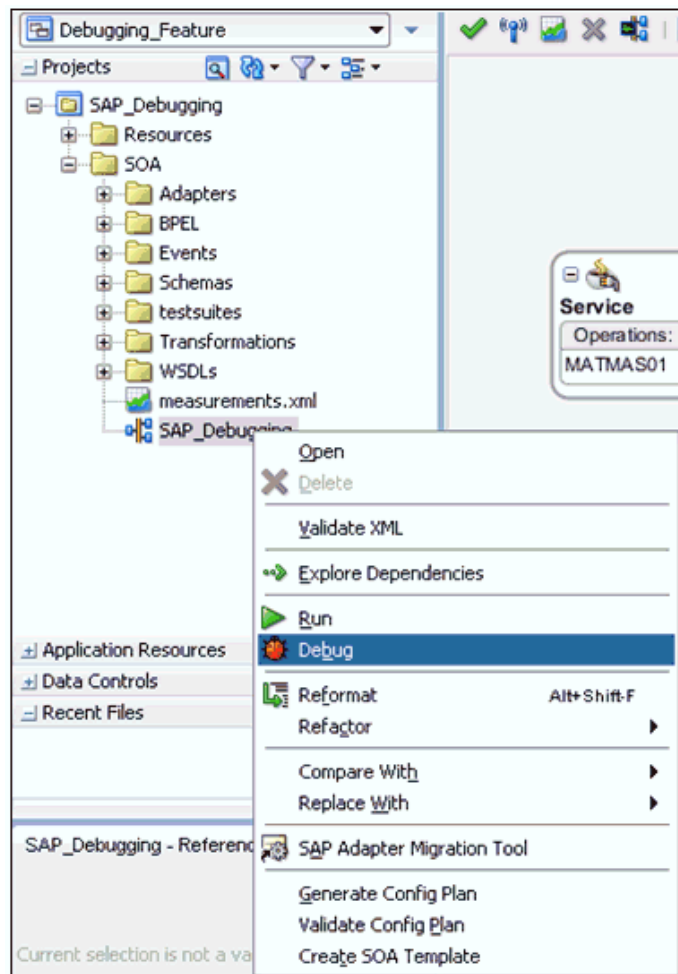
The Select Server dialog is displayed.

- 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.

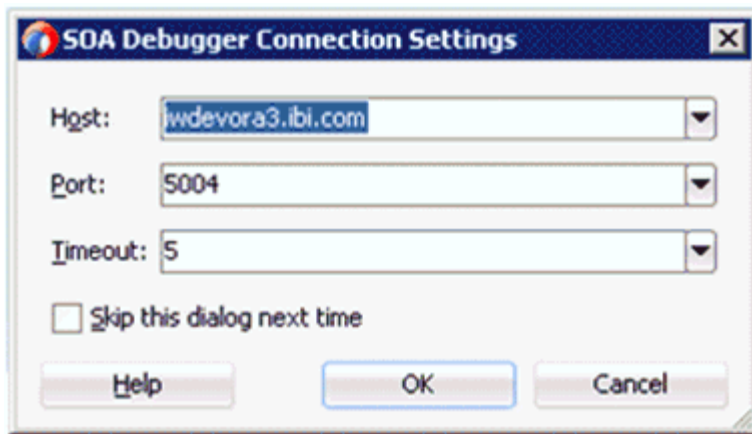
2. Connect a BPEL process to the SOA Debugger.
 - a. Right-click an inbound composite and select **Debug**, as shown in [Figure 11-33](#).

Figure 11-33 Select BPEL Process to Debug



The SOA Debugger Connection Settings dialog is displayed, as shown in [Figure 11-34](#).

Figure 11–34 SOA Debugger Connection Settings Dialog



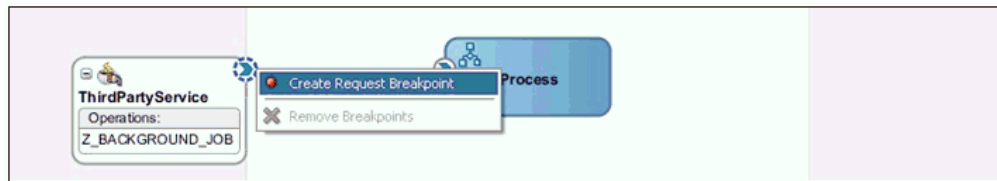
- b. Click **OK**.

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
```

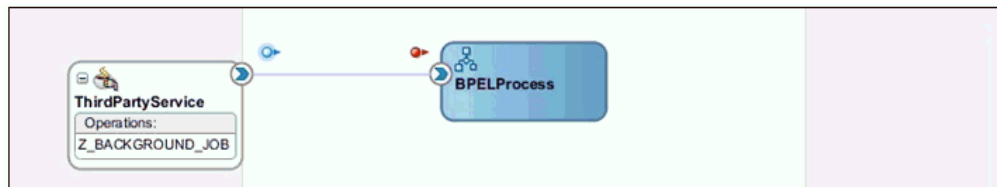
- 3. Set the Breakpoints, as shown in [Figure 11–35](#).

Figure 11–35 Setting Breakpoints



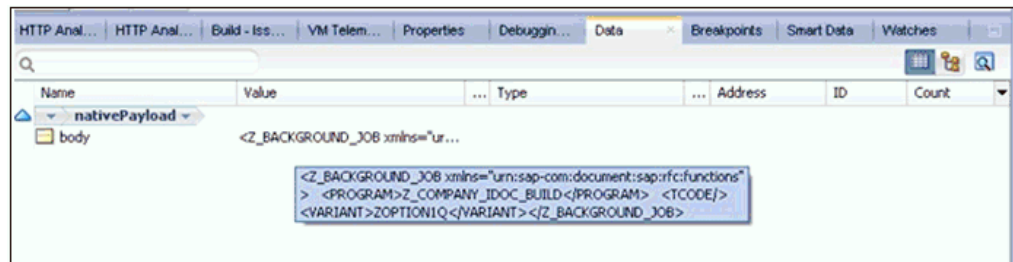
- 4. Once the Breakpoints are set and a message is received through the inbound process (for example, by triggering from SAP), the process stops at the designated Breakpoints, as shown in [Figure 11–36](#).

Figure 11–36 Stopping the Process at the Breakpoint



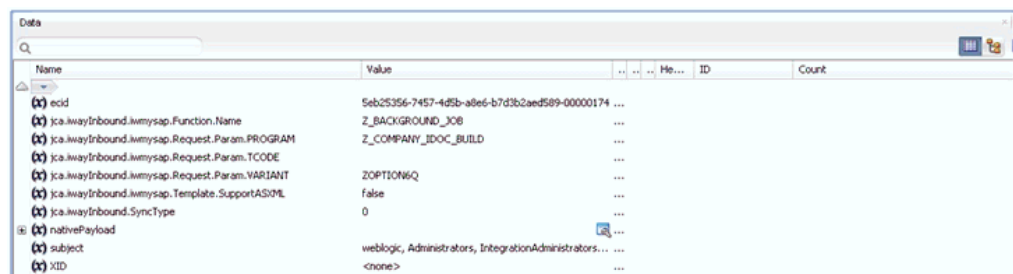
- 5. View the Request payload and header information.
 - a. Click **Windows**, select **Debugger**, and then **Data**, as shown in [Figure 11–37](#).

Figure 11–37 Debugger Data



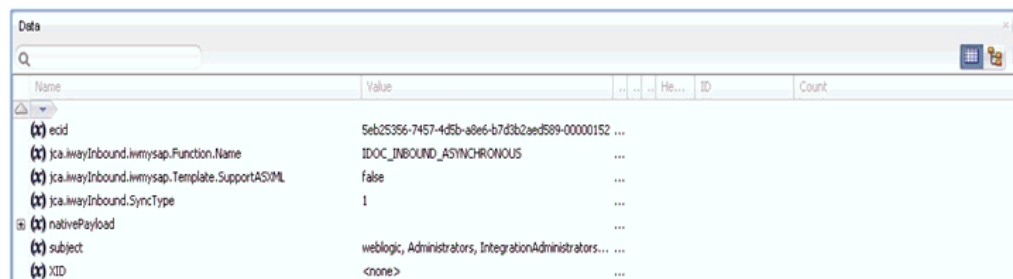
- b. Header information displayed for Inbound Request Breakpoint (Z_BACKGROUND_JOB) is shown in Figure 11–38.

Figure 11–38 Inbound Request Breakpoint (Z_BACKGROUND_JOB)



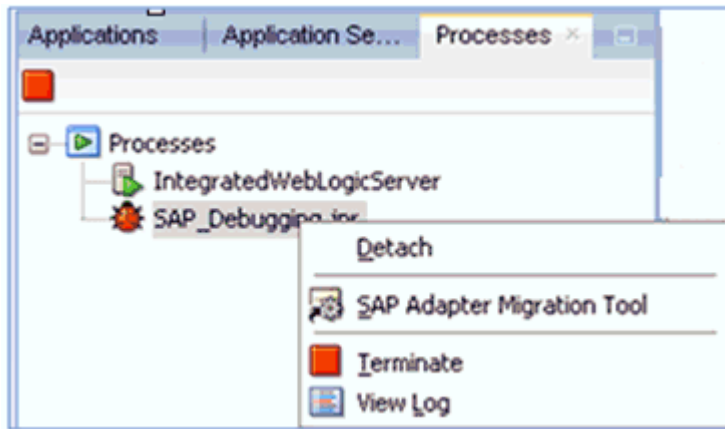
- c. Header information displayed for Inbound Request Breakpoint (IDOC MATMAS01) is shown in Figure 11–39.

Figure 11–39 Inbound Request Breakpoint (IDOC MATMAS01)



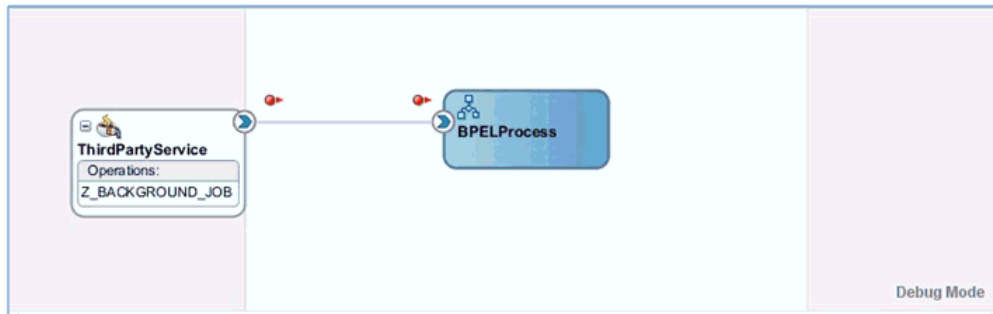
6. End or detach the Debugging session.
 - a. Click **Window** and then **Processes**. Right click on the process in the Processes tab and select **Detach** or **Terminate**, as shown in Figure 11–40.

Figure 11–40 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 11–41](#).

Figure 11–41 BPEL Process Detached



11.3.4 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 11.3.4.1, "Prerequisites"](#)
- [Section 11.3.4.2, "Debugging an Outbound OSB Process in Oracle JDeveloper"](#)
- [Section 11.3.4.3, "Debugging an Inbound OSB Process in Oracle JDeveloper"](#)

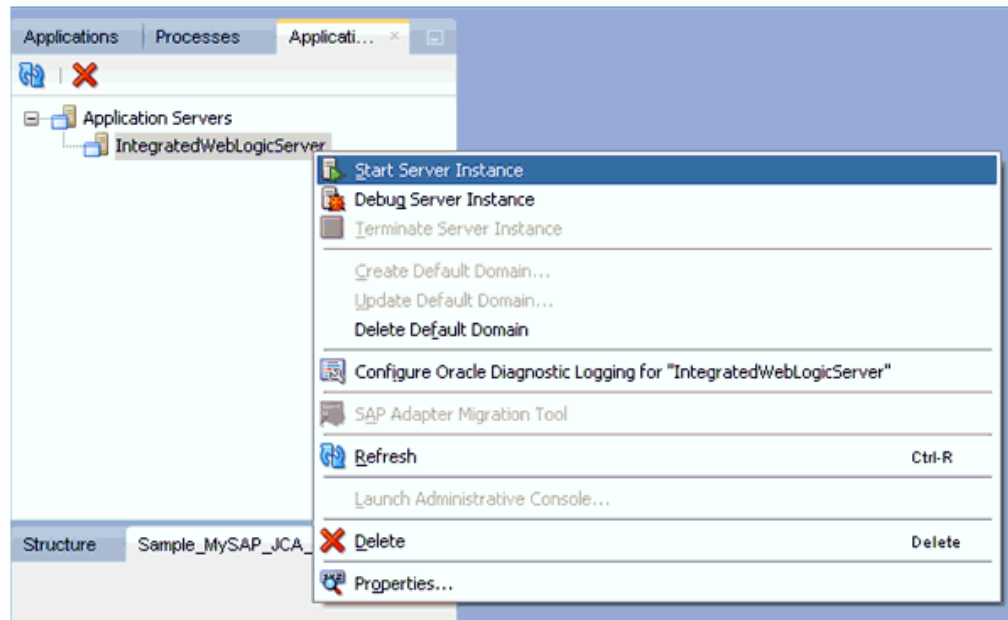
11.3.4.1 Prerequisites

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.

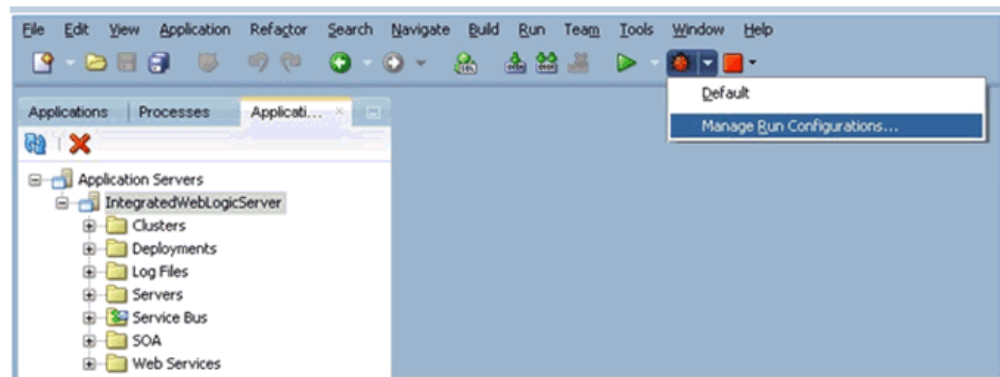
1. Open Oracle JDeveloper and start *IntegratedWeblogicServer* or *startWebLogic.cmd*.
 - a. Click the **Application Servers** tab in the left pane.
 - b. Under the Application Servers node, right-click **IntegratedWeblogicServer** and select **Start Server Instance** from the context menu, as shown in [Figure 11-42](#).

Figure 11-42 Start Server Instance



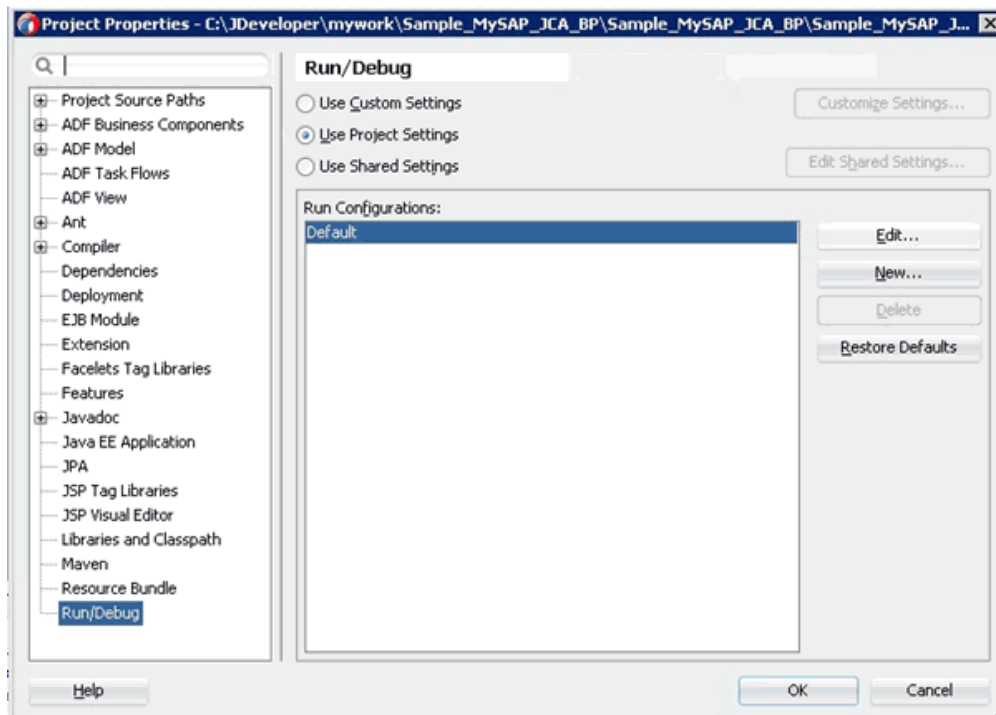
- c. Or, start Oracle WebLogic server from the command prompt using *startWebLogic.cmd*.
2. Set the Debugging environment.
 - a. Click the down arrow next to the Debug icon and select **Manage Run Configurations** from the context menu, as shown in [Figure 11-43](#).

Figure 11-43 Manage Run Configurations



- b. Or, right-click the project and select **Project Properties**.
The Project Properties dialog is displayed, as shown in [Figure 11-44](#).

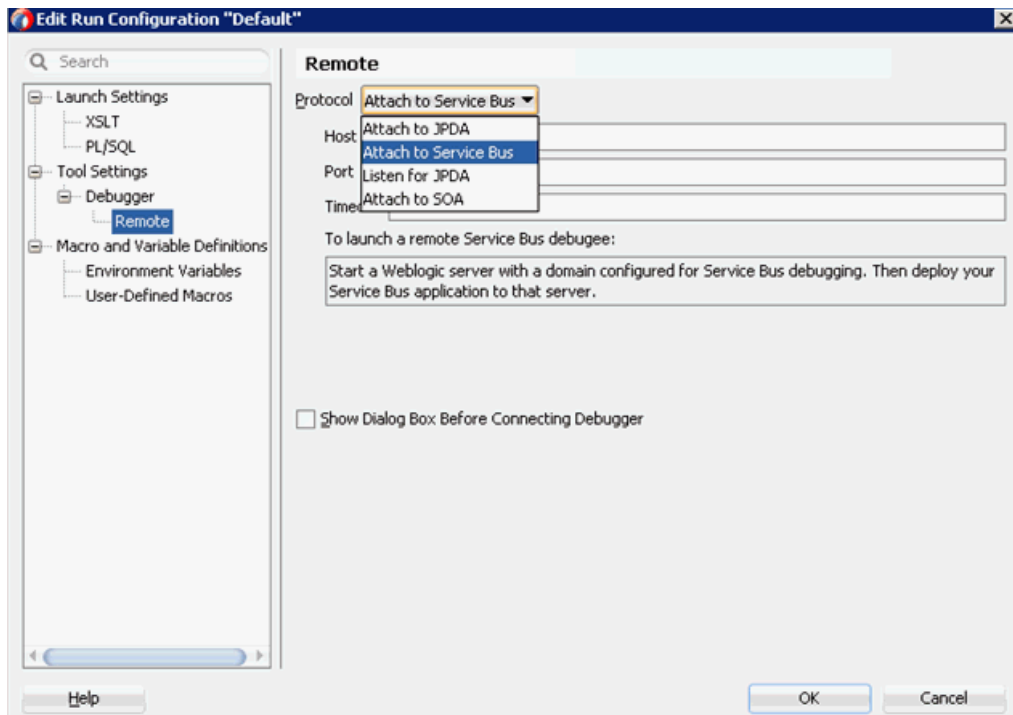
Figure 11–44 Project Properties Dialog



- 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 11–45](#).

Figure 11–45 Edit Run Configuration Dialog

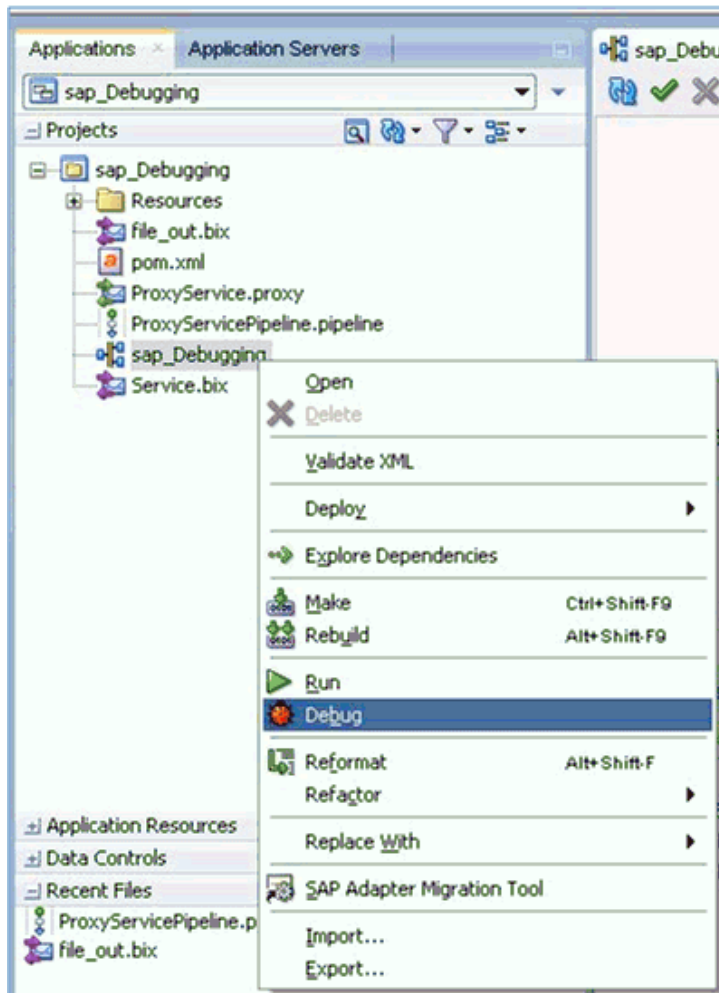


- d. Expand **Tool Settings, Debugger** in the left pane, and then click **Remote**.
- e. From the Protocol list, select **Attach to Service Bus**.
- f. Leave the default values for the Host and Port.
- g. Click **OK**.

11.3.4.2 Debugging an Outbound OSB Process in Oracle JDeveloper

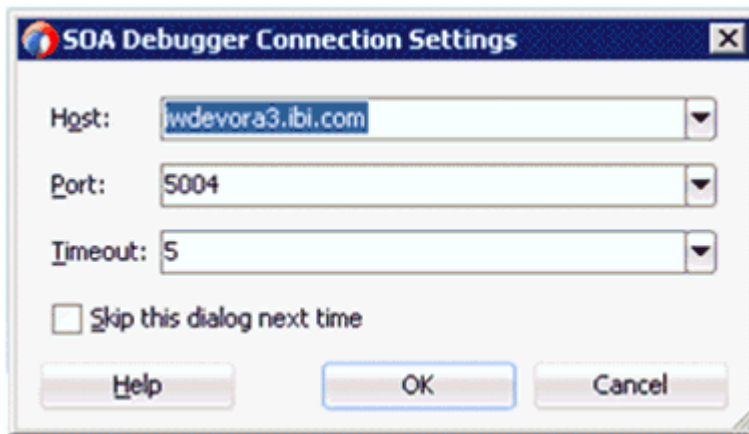
1. Deploy the project.
 - a. Right-click the project and select **Deploy**.
The Deployment Action dialog is displayed.
 - b. Select the application to deploy and click **Next**.
The Deploy Configuration dialog is displayed.
 - c. Click **Next**.
The Application Servers dialog is displayed.
 - d. Select **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 11-46](#).

Figure 11-46 Select OSB Process to Debug



The SOA Debugger Connection Settings dialog is displayed, as shown in Figure 11-47.

Figure 11-47 SOA Debugger Connection Settings Dialog



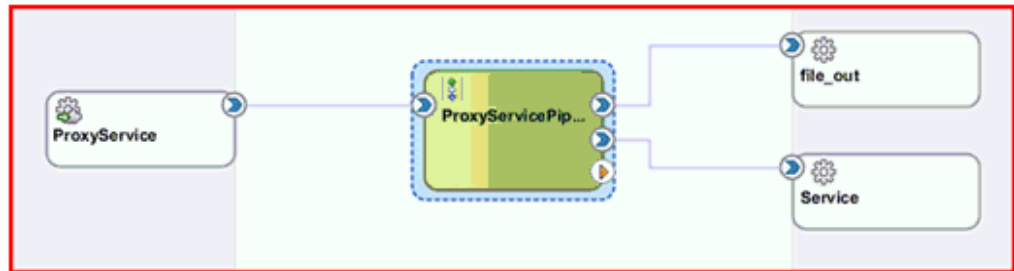
b. Click OK.

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
```

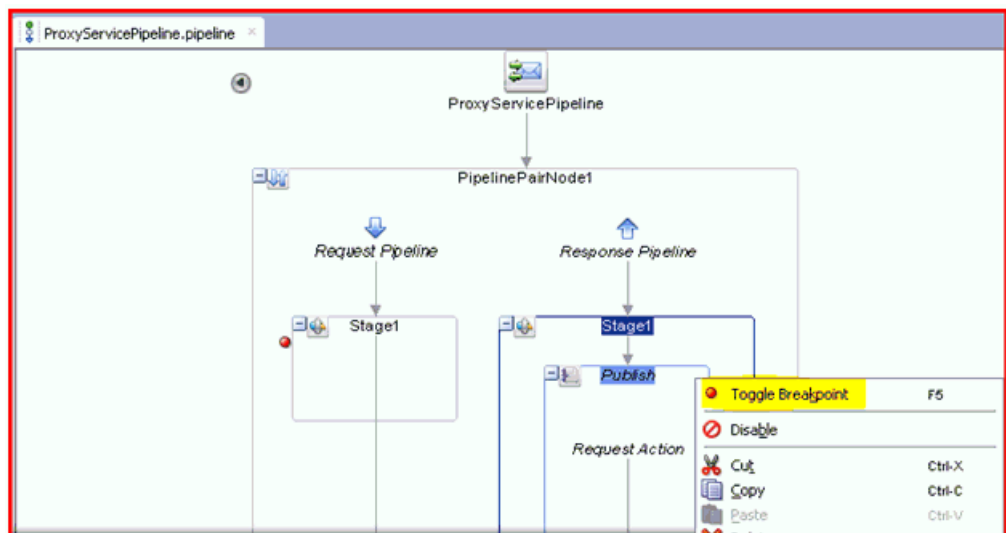
3. Set the Breakpoints.
 - a. Open the pipeline in its editor by double-clicking on the pipeline, as shown in [Figure 11-48](#).

Figure 11-48 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 11-49](#).

Figure 11-49 Toggle Breakpoint



Repeat this step for the Publish node (under **Response Pipeline**).

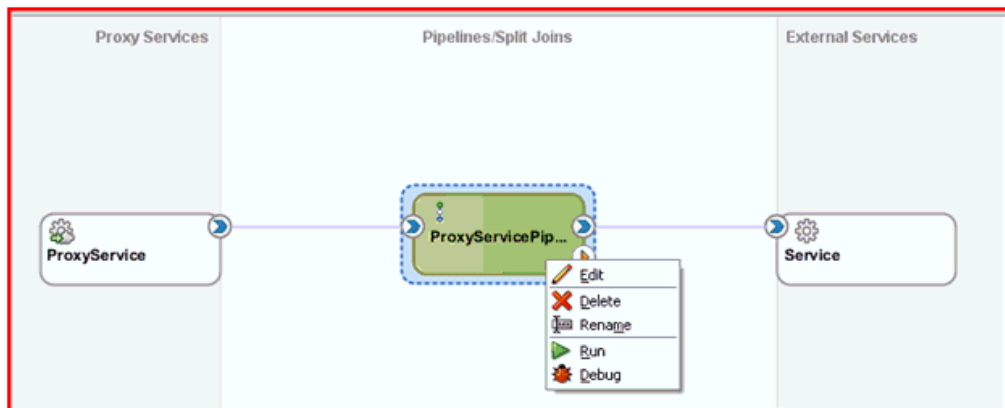
A red icon appears next to the node to indicate that a Breakpoint has been set.

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.

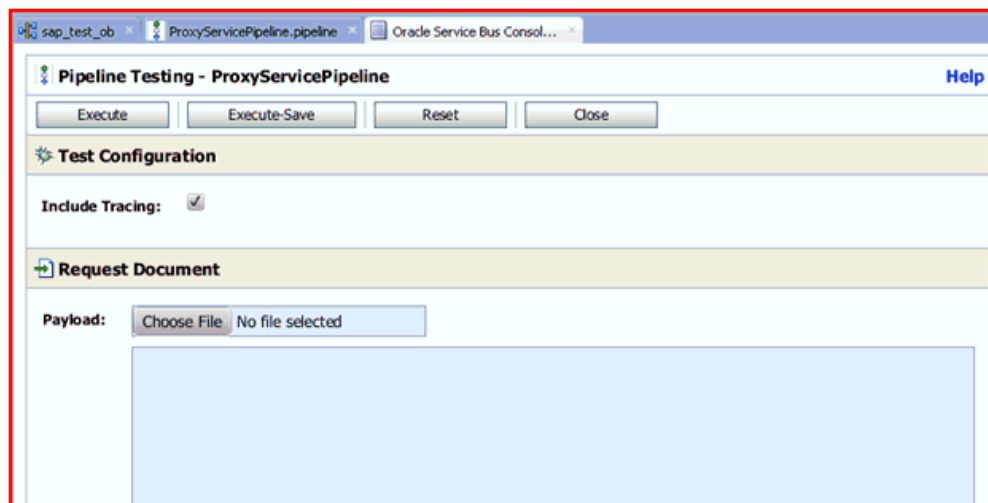
4. Initiate Debugging.
 - a. Right-click the pipeline in the Application Navigator, and select **Debug**, as shown in [Figure 11-50](#).

Figure 11-50 *Select Debug*



The process is deployed to the integrated server and the Test Configuration pane will be displayed, as shown in [Figure 11-51](#).

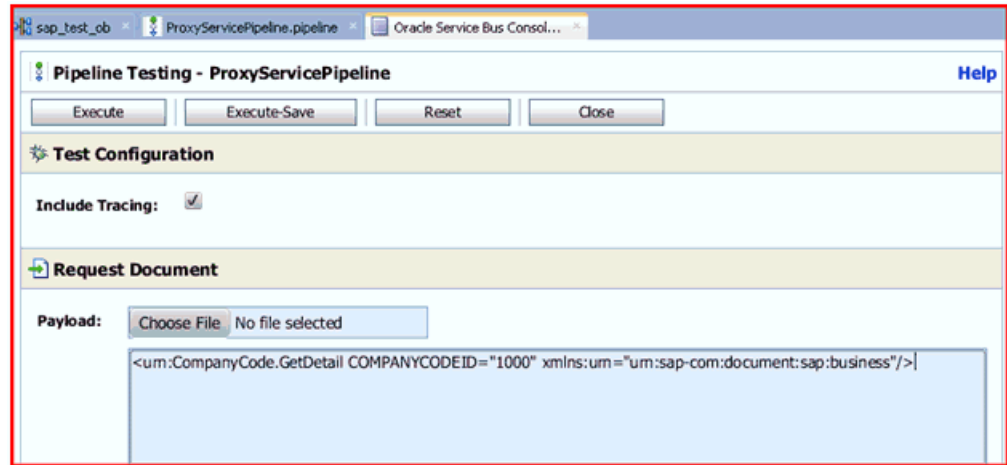
Figure 11-51 *Test Configuration Pane*



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 11-52.

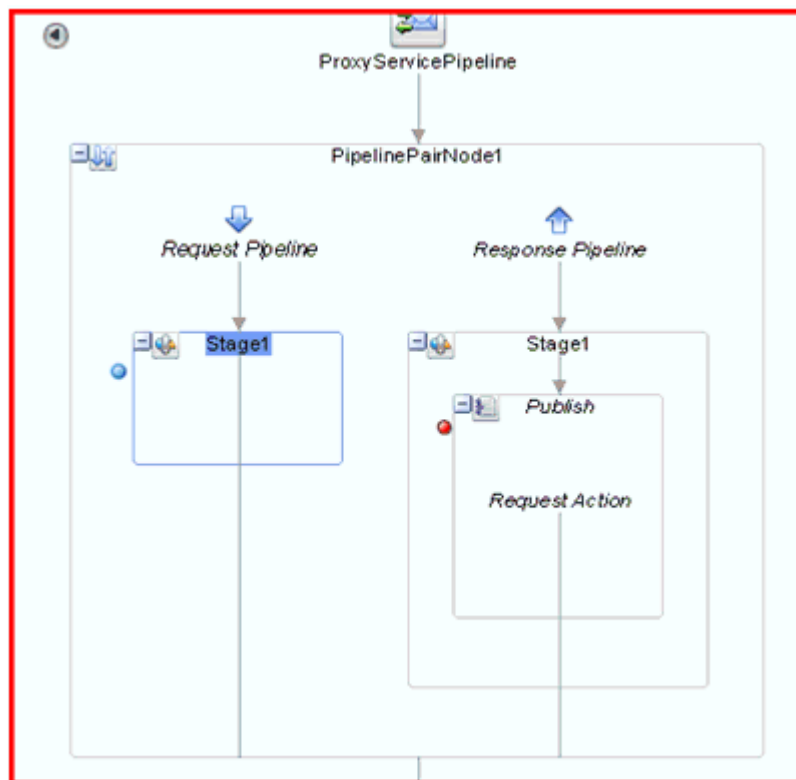
Figure 11-52 Test Request Document



- c. Click **Execute**.

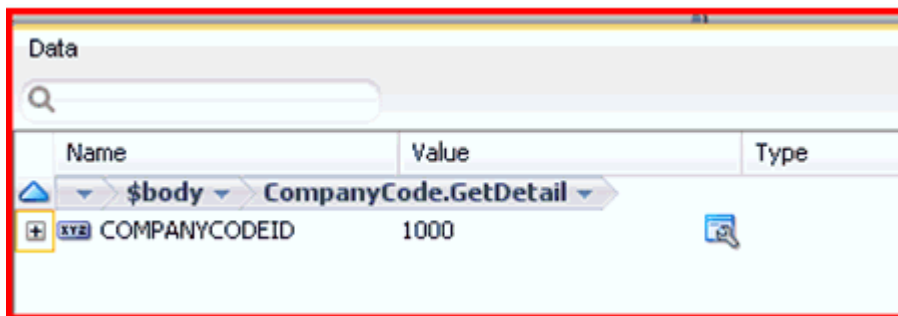
The Test Configuration pane executes the command, but the OSB process stops at the designated Breakpoint and blinks in a blue color, as shown in Figure 11-53.

Figure 11-53 OSB Process Stopped at Breakpoint



- d. Check the Data tab at the bottom to verify that the input has passed, as shown in [Figure 11-54](#).






Figure 11-54 Data Tab



- e. Use the available Step options to step through the Debugging process, as shown in [Figure 11-55](#).

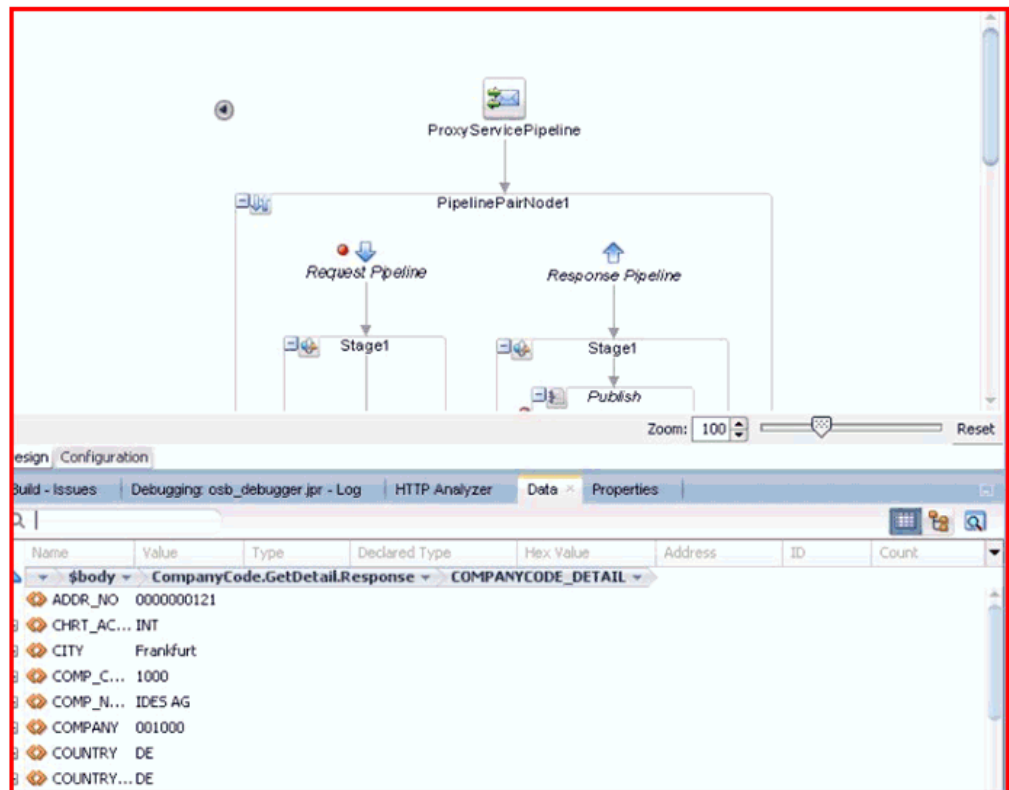
Figure 11-55 Step Options



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.

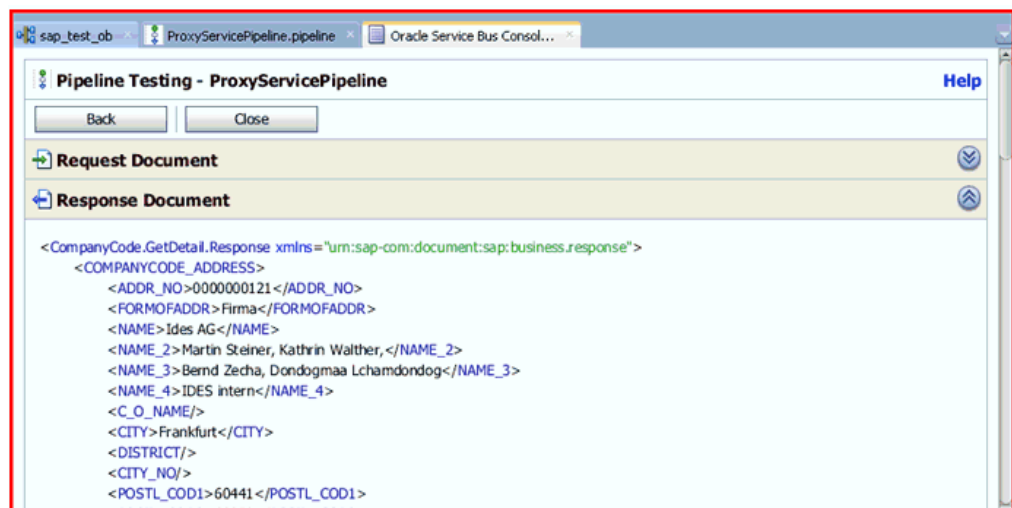
- f. Use **Step Over** to go to the next Breakpoint (Response Pipeline in this example).
You will be able to see the output in the Data tab, as shown in [Figure 11-56](#).

Figure 11–56 Data Tab Output

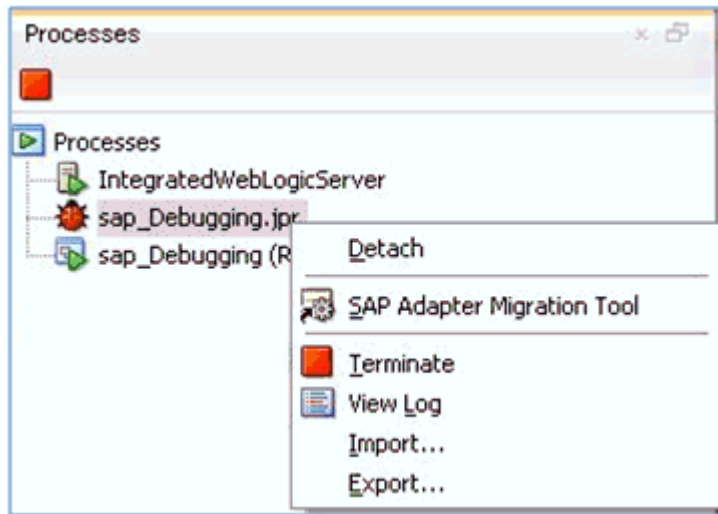


You will also be able to see the response in the Oracle Service Bus Console, as shown in Figure 11–57.

Figure 11–57 Oracle Service Bus Console



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 11–58.

Figure 11–58 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.

11.3.4.3 Debugging an Inbound OSB Process in Oracle JDeveloper

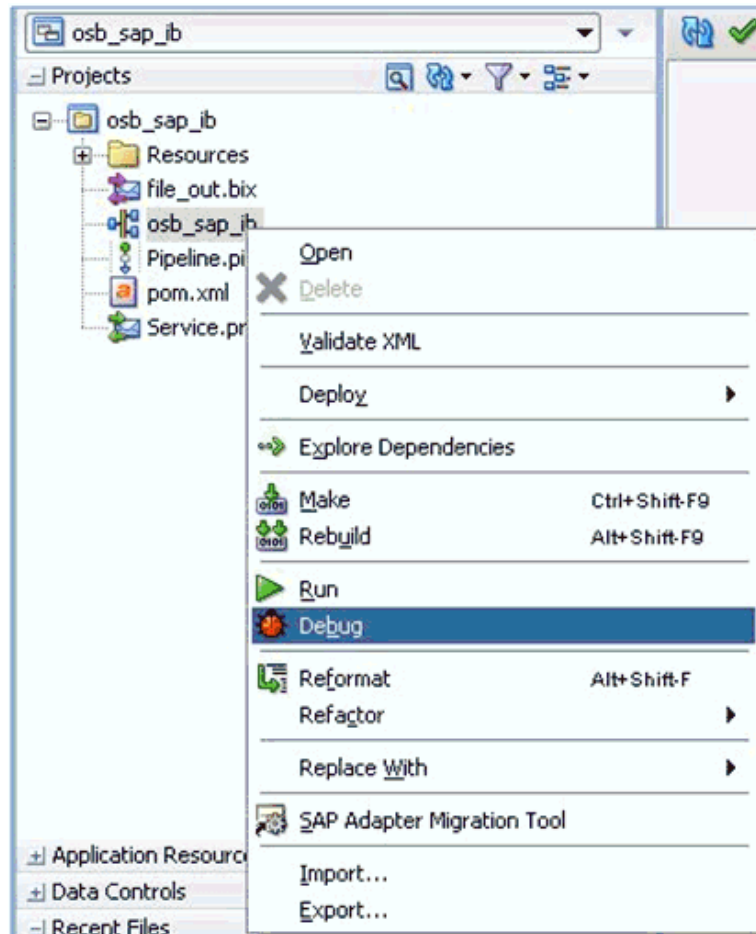
1. Create an inbound OSB process.

Ensure that the application name and the project name are the same.
2. Deploy the project.
 - a. Right-click the project and select **Deploy**.

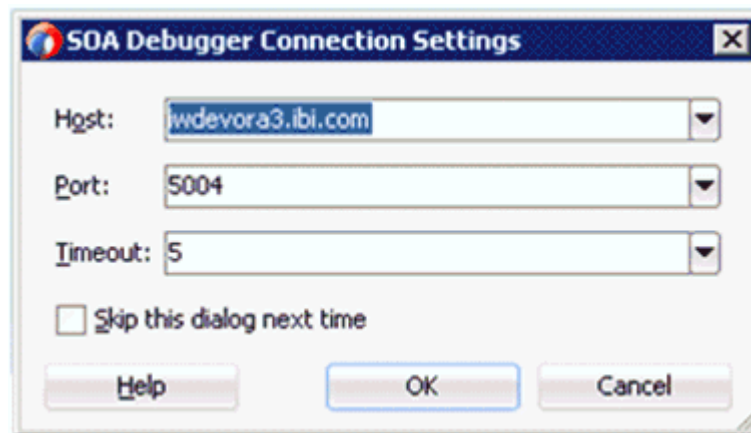
The Deployment Action dialog is displayed.
 - b. Select the application to deploy and click **Next**.

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 11–46](#).

Figure 11–59 Select OSB Process to Debug

The SOA Debugger Connection Settings dialog is displayed, as shown in [Figure 11–60](#).

Figure 11–60 SOA Debugger Connection Settings Dialog

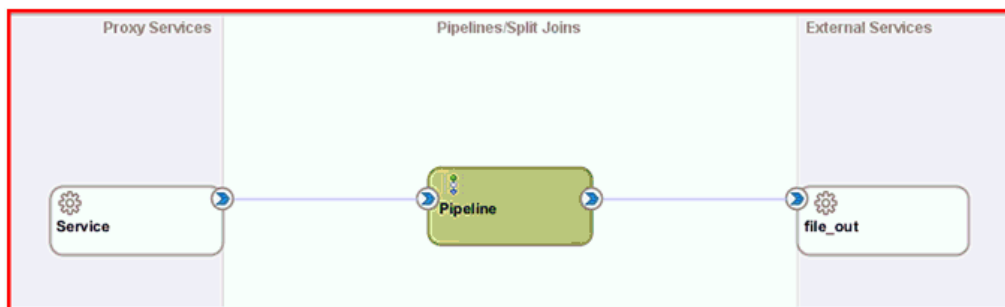
- b. Click OK.

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

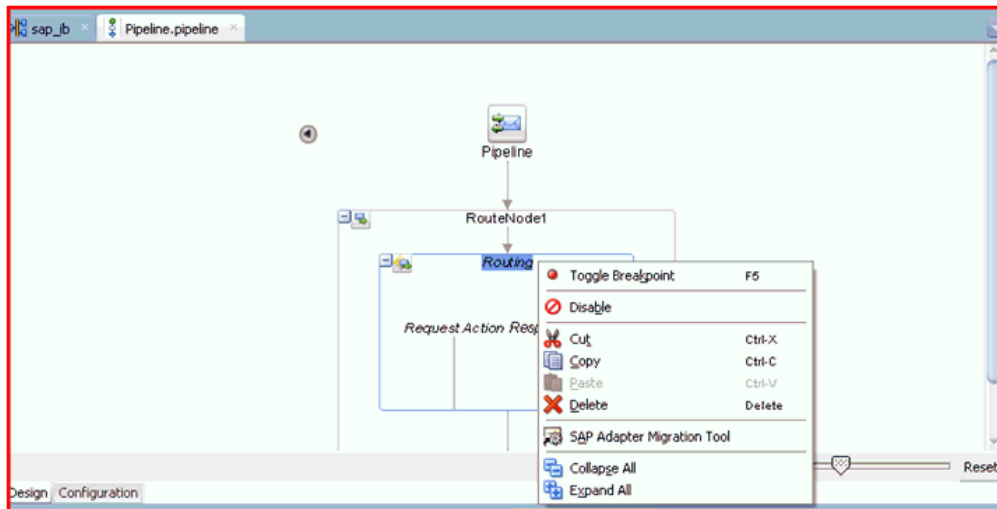
4. Set the Breakpoints.
 - a. Open the pipeline in its editor by double-clicking on the pipeline, as shown in [Figure 11-48](#).

Figure 11-61 ProxyServicePipeline



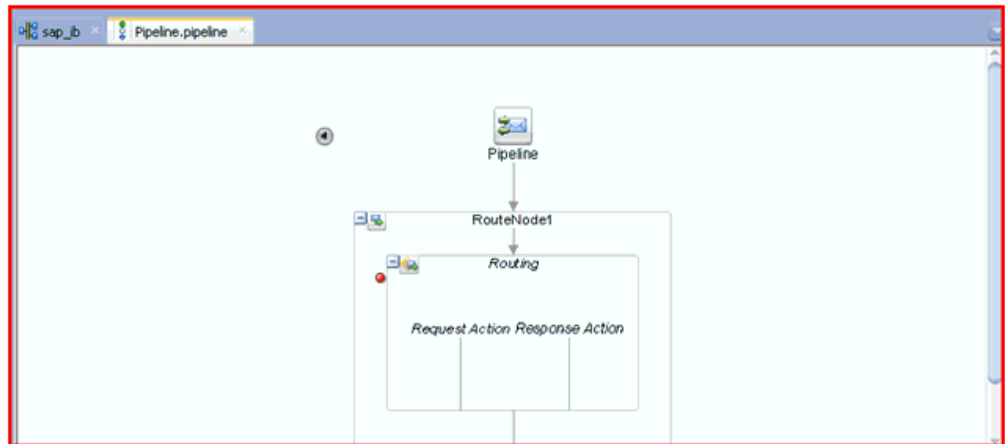
- b. Expand the actions until you reach the node where the Breakpoint must be added. In this case, expand until you reach **Routing**. Right-click the **Routing** node and select **Toggle Breakpoint** from the context menu, as shown in [Figure 11-62](#).

Figure 11-62 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.

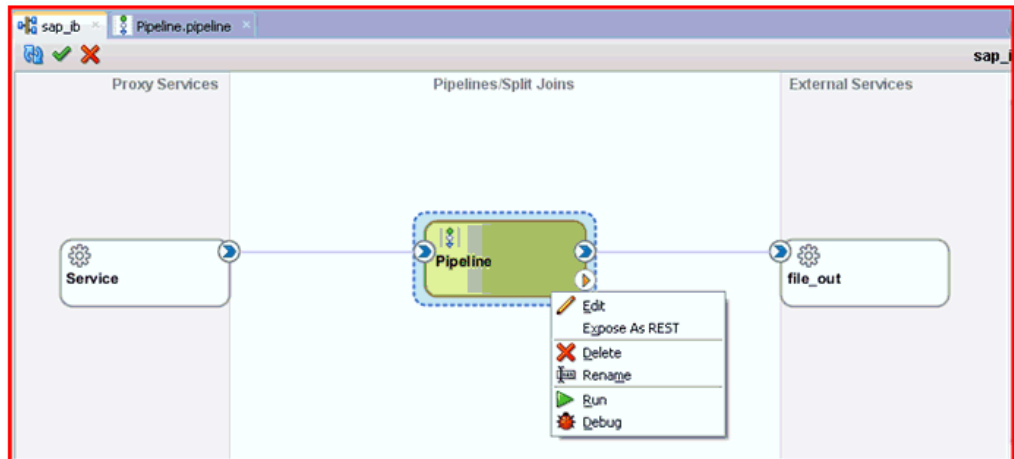
A red icon appears next to the node to indicate that a Breakpoint has been set, as shown in [Figure 11-63](#).

Figure 11–63 Red Icon for Set Breakpoint

Note: To disable a Breakpoint, right-click the node and select **Disable Breakpoint**.

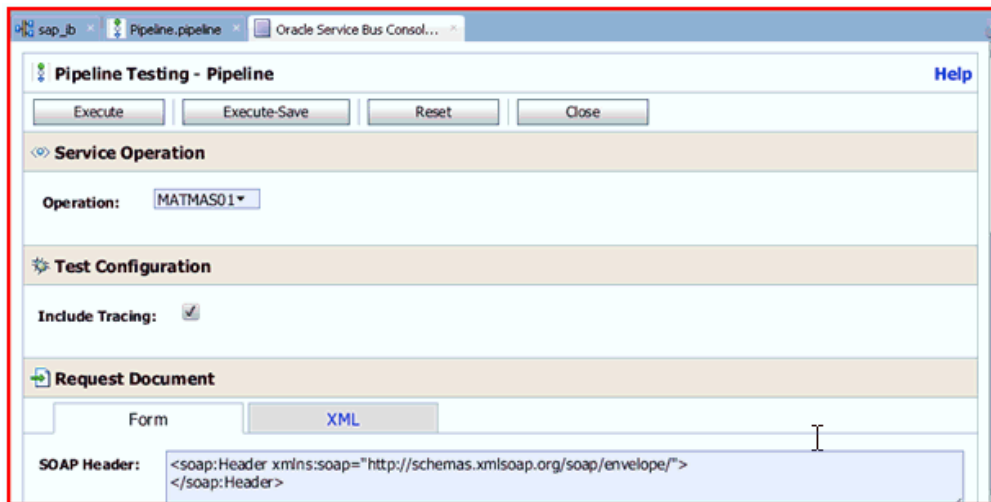
To remove a Breakpoint, right-click the node and select **Toggle Breakpoint** again.

5. Initiate Debugging.
 - a. Right-click the pipeline in the Application Navigator, and select **Debug**, as shown in [Figure 11–64](#).

Figure 11–64 Select Debug

The process is deployed to the integrated server and the Test Configuration pane will be displayed, as shown in [Figure 11–65](#).

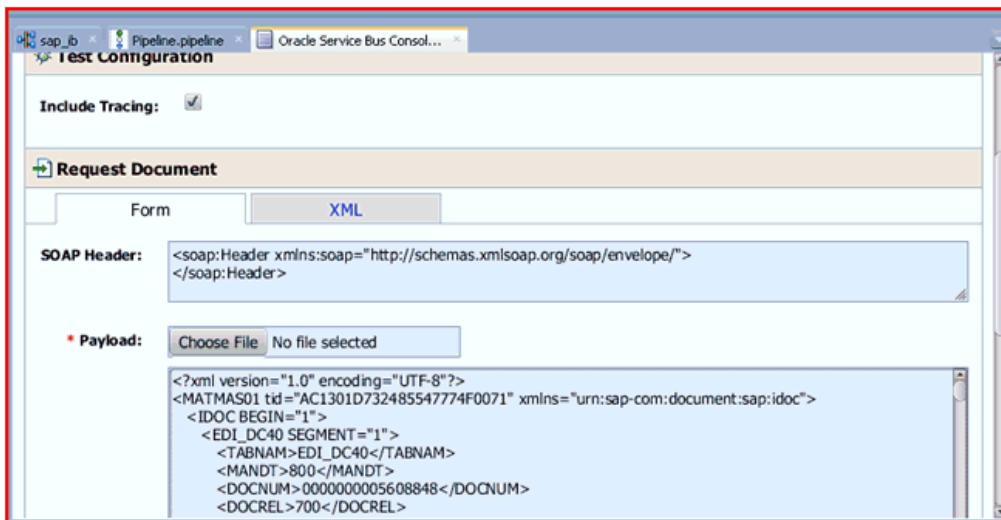
Figure 11–65 Test Configuration Pane



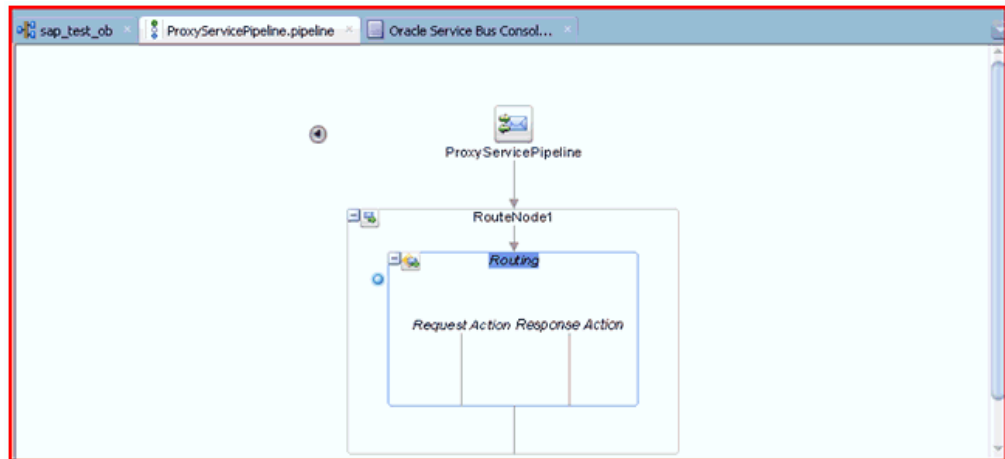
Note: If there is no domain currently running, then the Create Default Domain dialog will be displayed. Enter the connection information for the integrated server and then click **OK**. This process may take several minutes.

- 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 11–66.

Figure 11–66 Test Request Document








- 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 11–67.

Figure 11–67 OSB Process Stopped at Breakpoint

- d. Use the available Step options to step through the Debugging process, as shown in [Figure 11–68](#).

Figure 11–68 Step Options

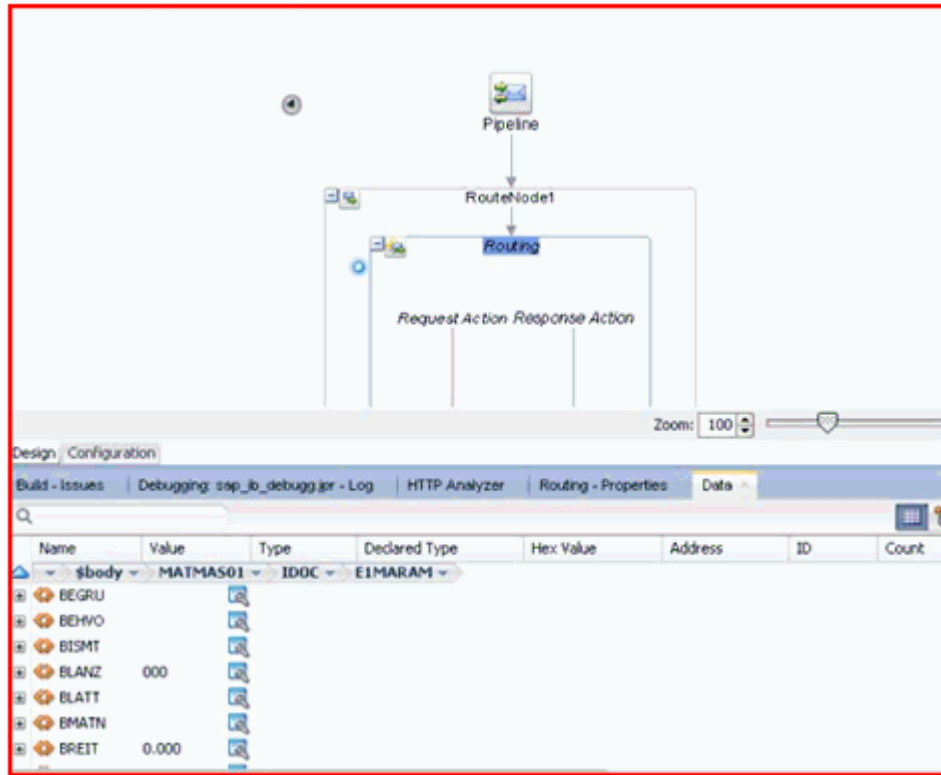
Icon	Description
	Ends or detaches from a debugging session.
	Steps over a frame. This places you at the next Breakpoint (for example, the receive activity in the OSB process on which a Breakpoint was set. If there are no Breakpoints, it steps over all the frames and returns to the first frame. You can 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.

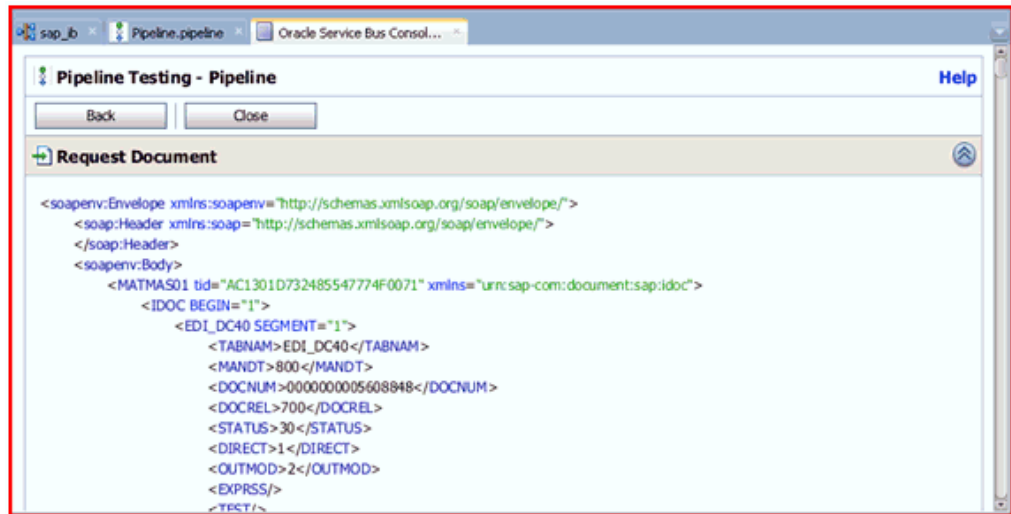
You will be able to see the response document displayed in the Data tab, as shown in [Figure 11-56](#).

Figure 11-69 Data Tab Output



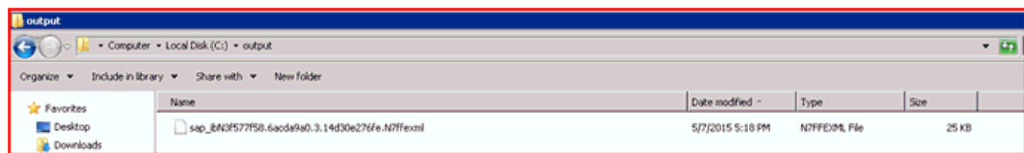
You will also be able to see the response in the Oracle Service Bus Console, as shown in [Figure 11-70](#).

Figure 11–70 Oracle Service Bus Console



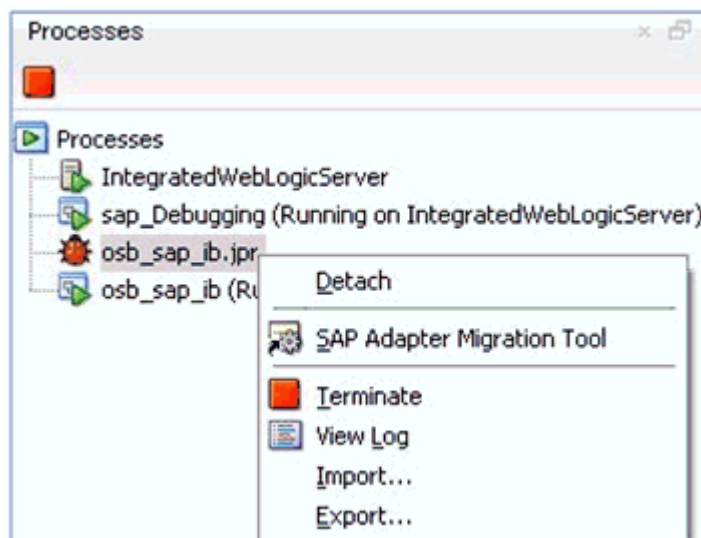
The output will also be available in the configured output location, as show in Figure 11–71.

Figure 11–71 Configured Output Location



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 11–72.

Figure 11–72 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.

11.4 Configuring the Exception Filter

This section describes how to configure exception filter functionality for the Oracle Application Adapter for SAP R/3 and includes a sample testing scenario. This section contains the following topic:

- [Section 11.4.1, "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.SAPExceptionFilter` 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.

Note: The `com.ibi.afjca.oracle.AdapterExceptionFilter` class can also be used to filter the generated exceptions.

11.4.1 Configuring the Exception Filter

This section contains the following topics:

- [Section 11.4.1.1, "Generating a WSDL File"](#)
- [Section 11.4.1.2, "Creating a BPEL Outbound Process With Exception Filter Functionality"](#)
- [Section 11.4.1.3, "Creating Fault Policies and Fault Binding Files"](#)
- [Section 11.4.1.4, "Adjusting for Known Deployment Issues With 12c"](#)
- [Section 11.4.1.5, "Deploying and Testing the BPEL Outbound Process With Exception Filter Functionality"](#)

Exception filter configuration consists of the following steps:

1. Generate a WSDL file.
For more information, see ["Generating a WSDL File"](#) on page 11-46.
2. Create a BPEL outbound process with exception filter functionality.
For more information, see ["Creating a BPEL Outbound Process With Exception Filter Functionality"](#) on page 11-46.
3. Create fault policies and fault binding files.
For more information, see ["Creating Fault Policies and Fault Binding Files"](#) on page 11-47.

4. Deploy and test the BPEL outbound process with exception filter functionality.
For more information, see ["Deploying and Testing the BPEL Outbound Process With Exception Filter Functionality"](#) on page 11-57.

11.4.1.1 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 4-6.
2. Create a target for the SAP R/3 adapter and then connect to the target.
For more information, see ["Establishing a Connection \(Target\) for SAP R/3"](#) on page 4-8.
3. Generate a WSDL for the appropriate object (for example, the GetDetail BAPI for the CompanyCode business object).
For more information, see ["Generating WSDL \(J2CA Configurations Only\)"](#) on page 4-18.

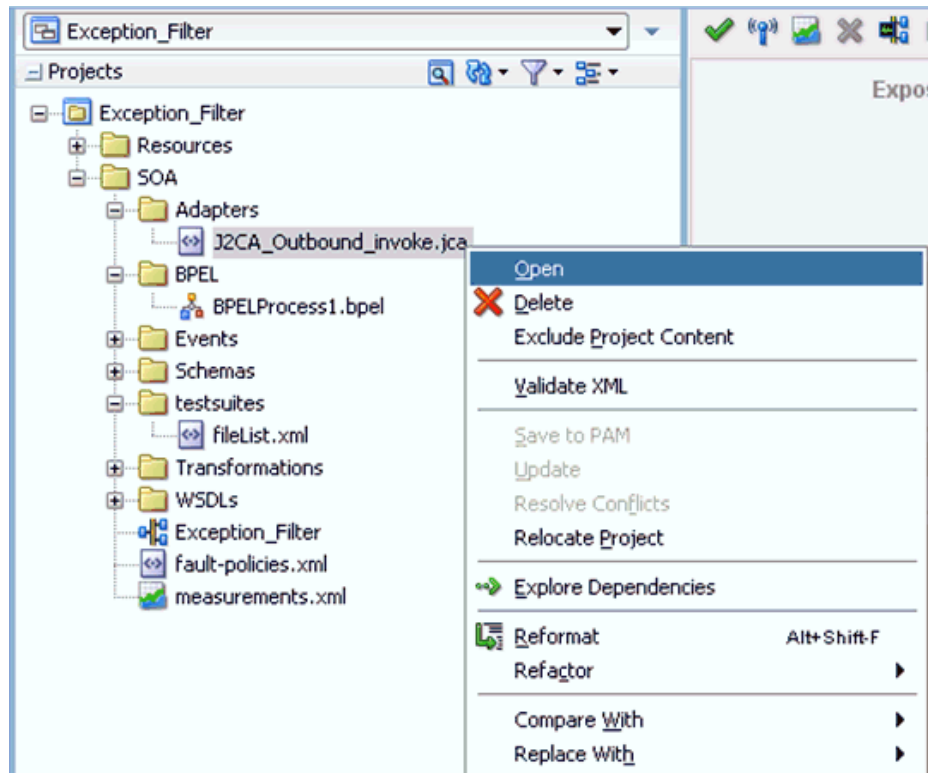
11.4.1.2 Creating a BPEL Outbound Process With Exception Filter Functionality

To create a BPEL outbound 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 6-9.
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 6-11.

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 6-11.
4. Modify the imported JCA file.
 - a. Right-click the imported JCA file and select **Open** from the menu, as shown in [Figure 11-73](#).

Figure 11–73 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.SAPEExceptionFilter"/></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.

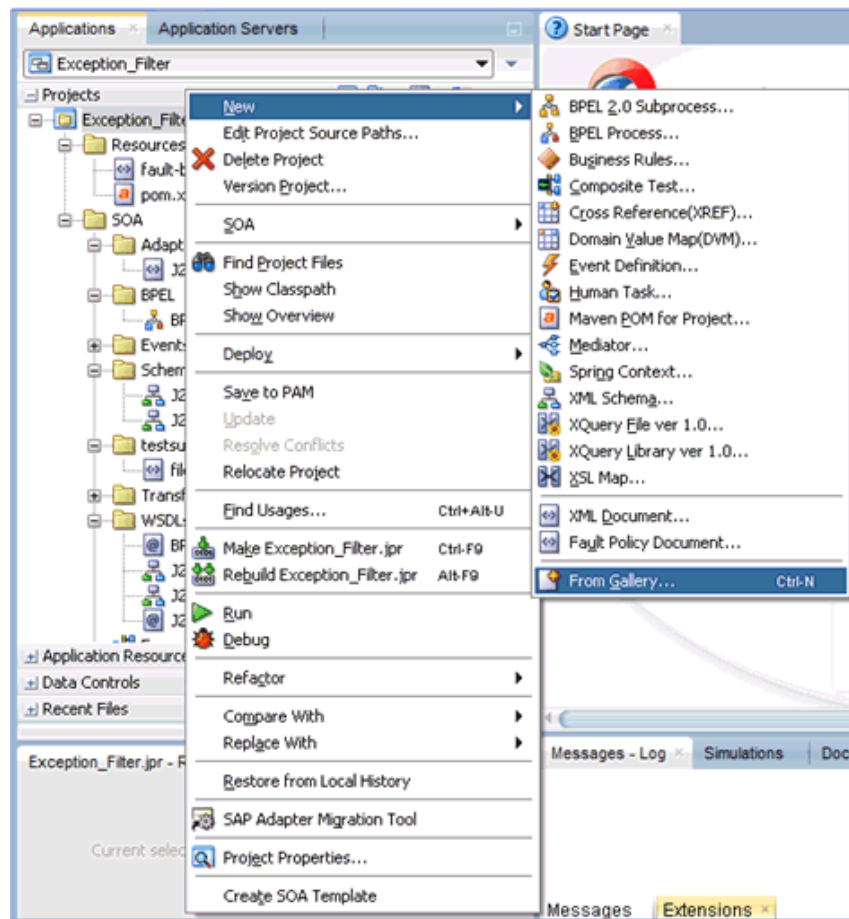
For more information, see ["Defining a BPEL Outbound Process"](#) on page 6-11.

11.4.1.3 Creating Fault Policies and Fault Binding Files

To create fault binding files:

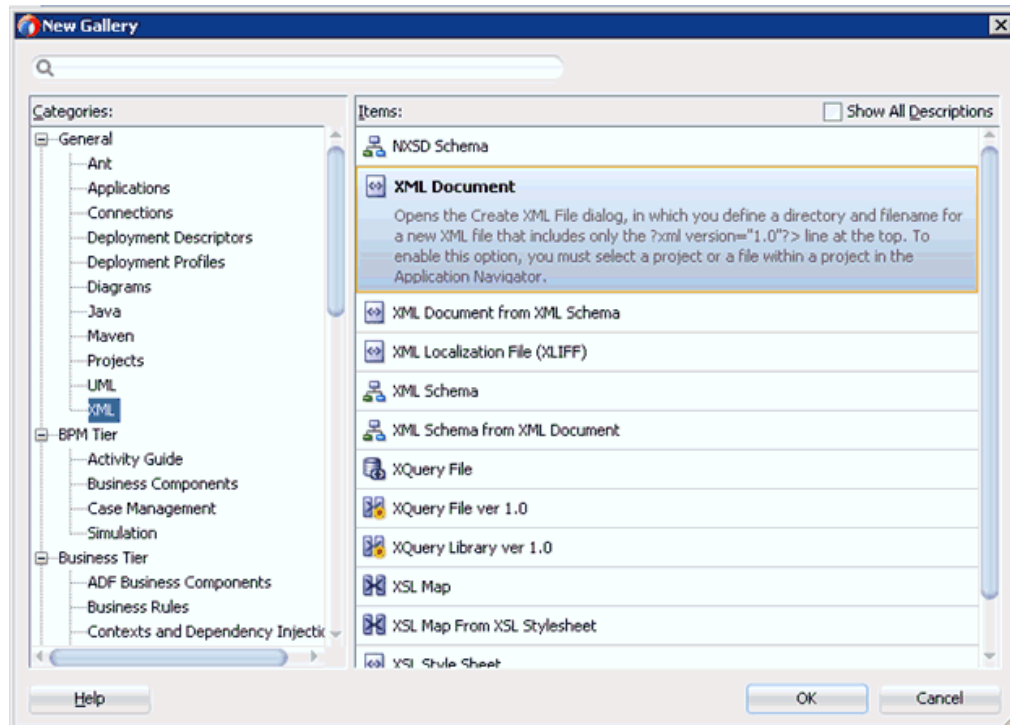
1. Right-click the created SOA project (for example, Exception_Filter), select **New**, and then click **From Gallery**, as shown in [Figure 11–74](#).

Figure 11-74 Applications Tab



The New Gallery dialog is displayed. Under the General category, click XML, as shown in Figure 11-75.

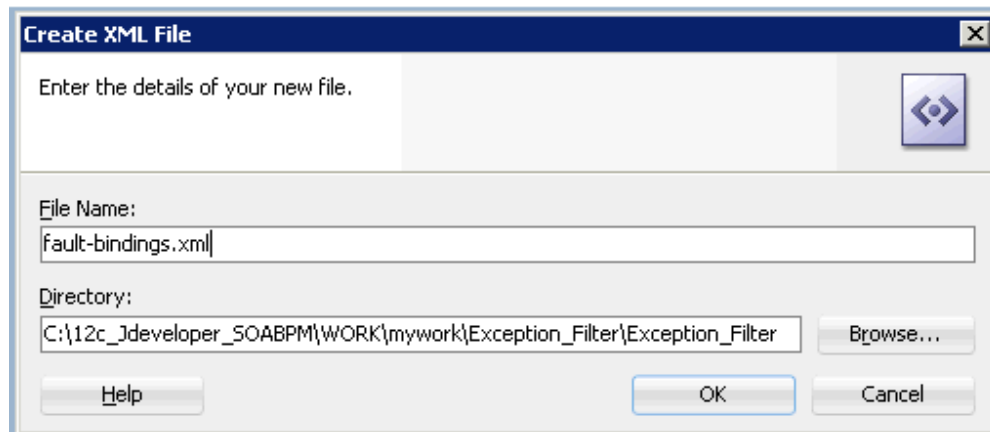
Figure 11–75 New Gallery Dialog



2. Select **XML Document** under Items and then click **OK**.

The Create XML File dialog is displayed, as shown in [Figure 11–76](#).

Figure 11–76 Create XML File Dialog



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 11-50.

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>

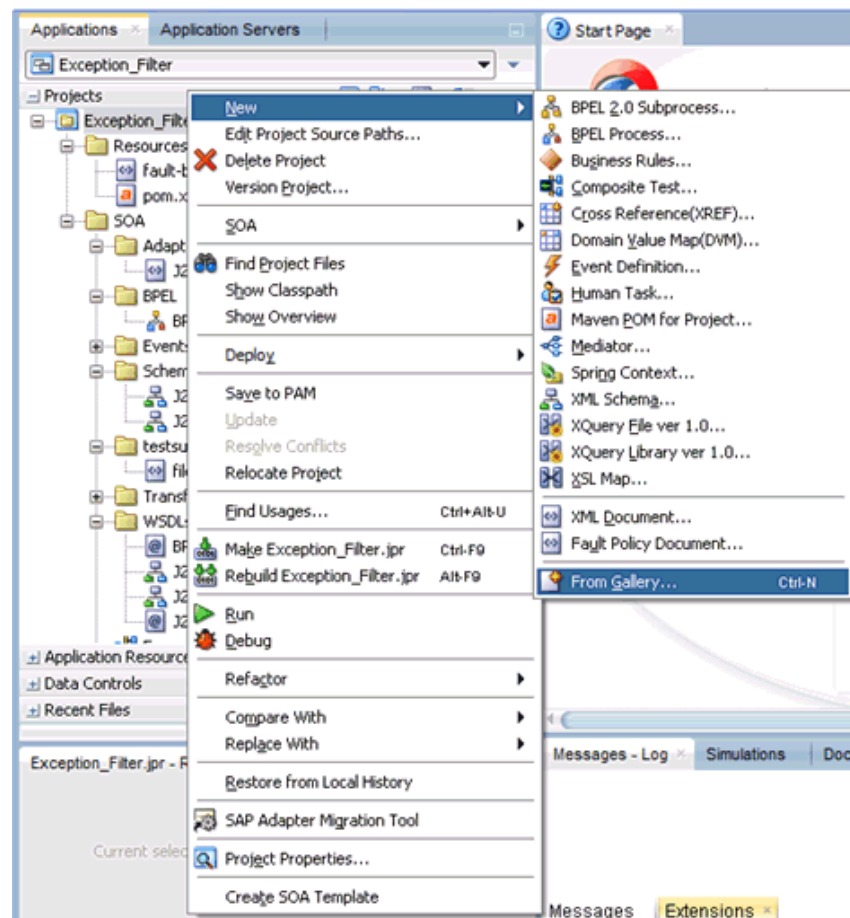
</faultPolicyBindings>
```

Creating Fault Policies Files

To create fault policies files:

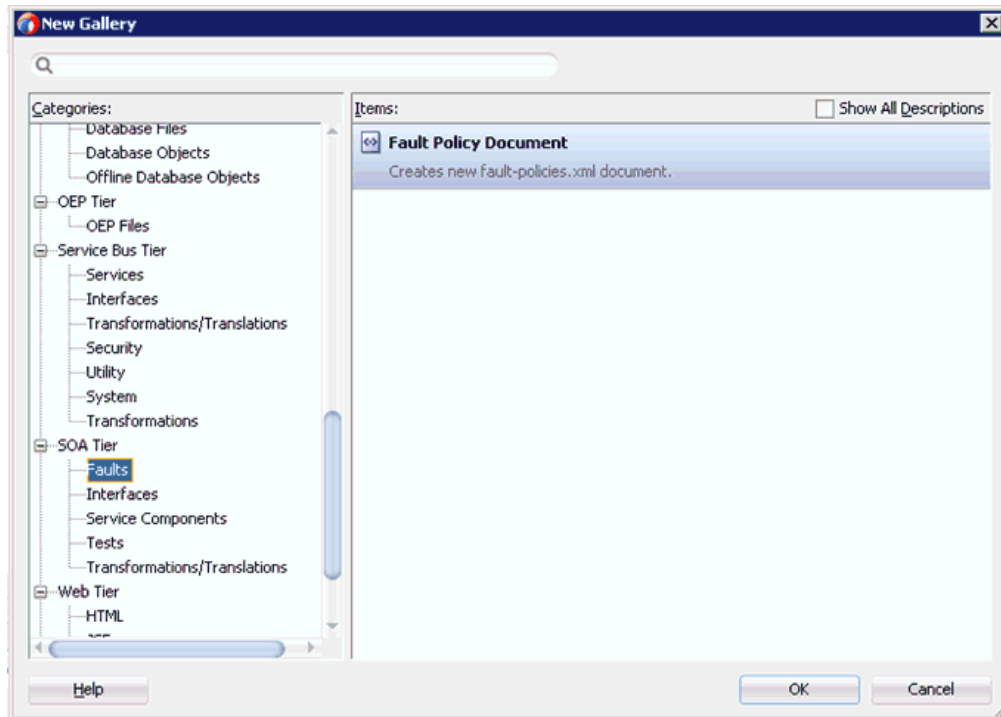
1. Right-click the created SOA project (for example, `Exception_Filter`), select **New**, and then click **From Gallery**, as shown in [Figure 11-77](#).

Figure 11-77 Applications Tab



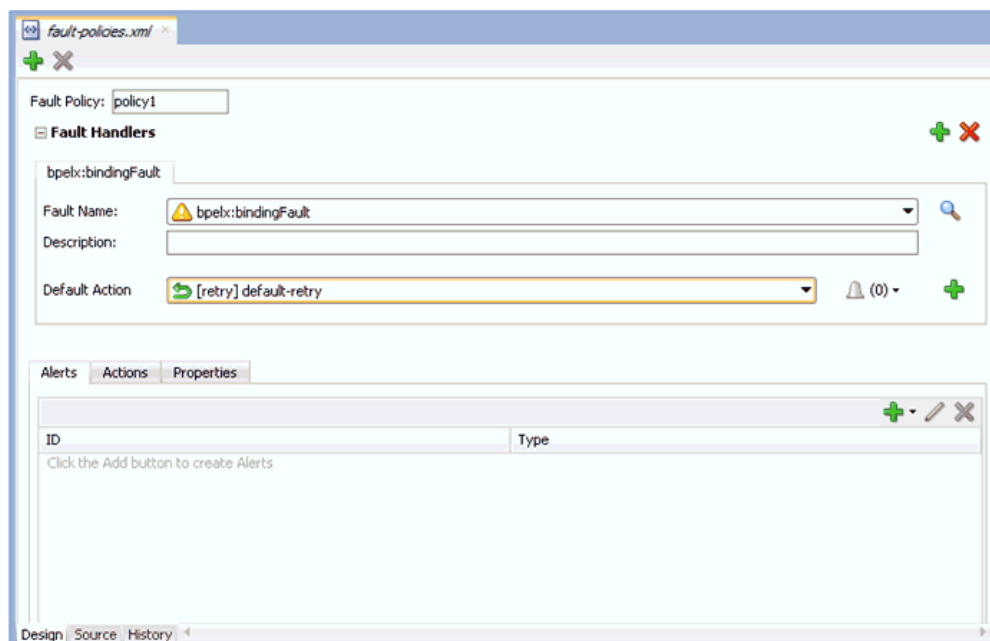
The New Gallery dialog is displayed. Under the SOA Tier category, select **Faults**, as shown in [Figure 11-78](#).

Figure 11–78 New Gallery Dialog



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 11–79](#).

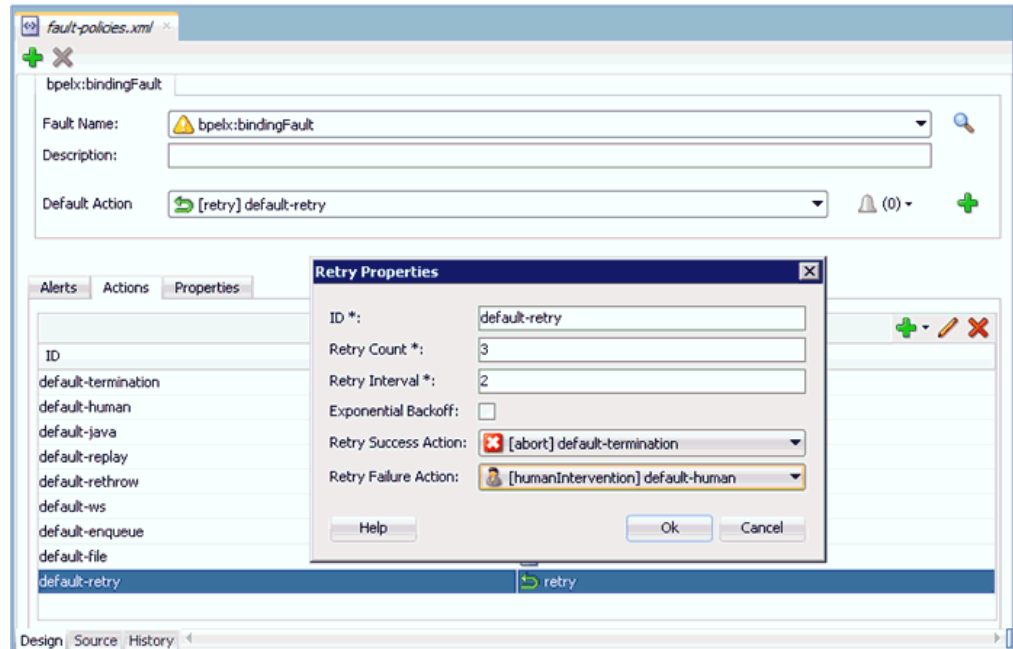
Figure 11–79 Fault-policies.xml Tab



4. Click the **Actions** tab and then double-click **default-retry**.

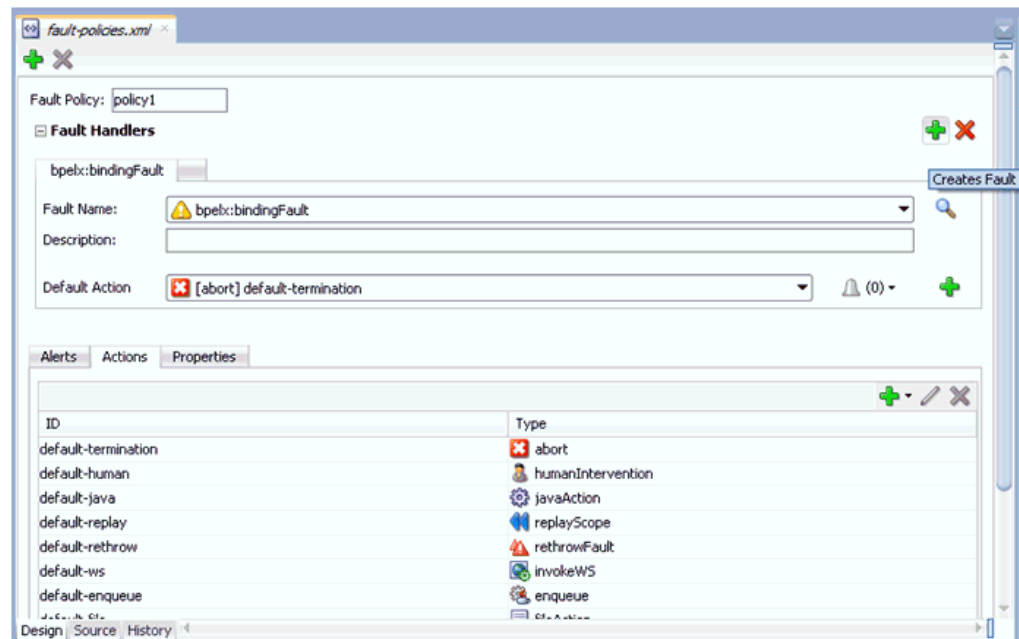
The Retry Properties dialog box is displayed, as shown in [Figure 11–80](#).

Figure 11–80 *Retry Properties Dialog Box*



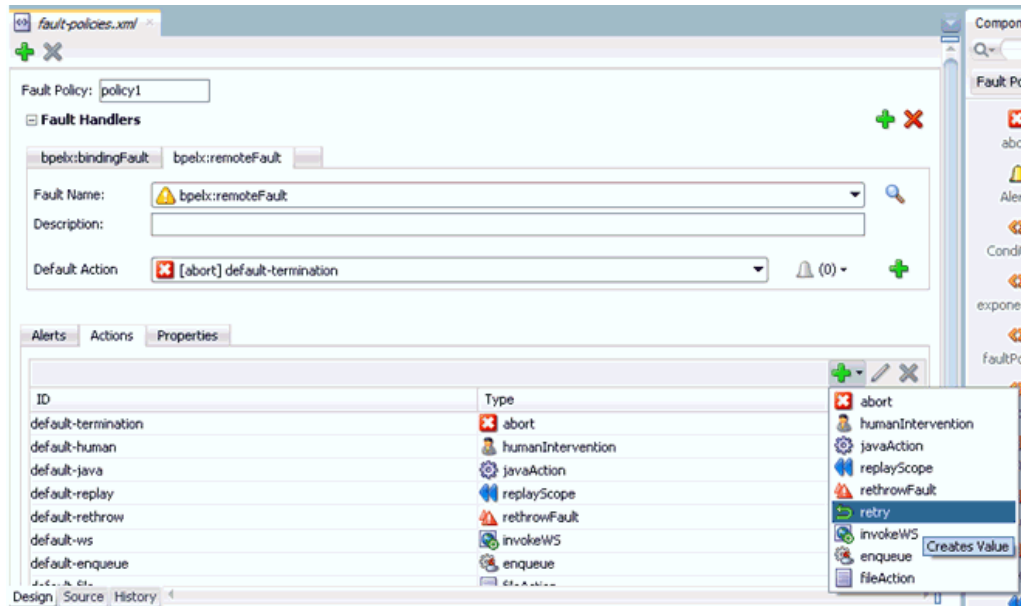
5. Select **[abort] default-termination** from the Retry Success Action drop-down list and **[humanIntervention] default-human** from the Retry Failure Action drop-down list.
6. Click **OK**.
7. Click **Add** to create another fault handler, as shown in [Figure 11–81](#).

Figure 11–81 *Fault-policies.xml Tab*



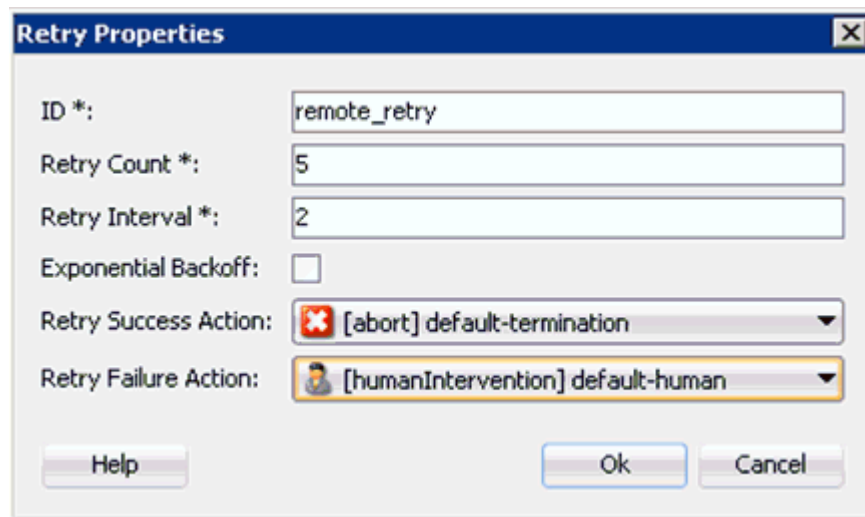
8. In the fault-policies.xml tab, select **bpelx:remoteFault** from the Fault Name drop-down list and **[abort] default-termination** from the Default Action drop-down list.
9. In the Actions tab, click **Add** and then select **retry**, as shown in [Figure 11–82](#).

Figure 11–82 *Actions Tab*



The Retry Properties dialog is displayed, as shown in [Figure 11–83](#).

Figure 11–83 *Retry Properties Dialog Box*

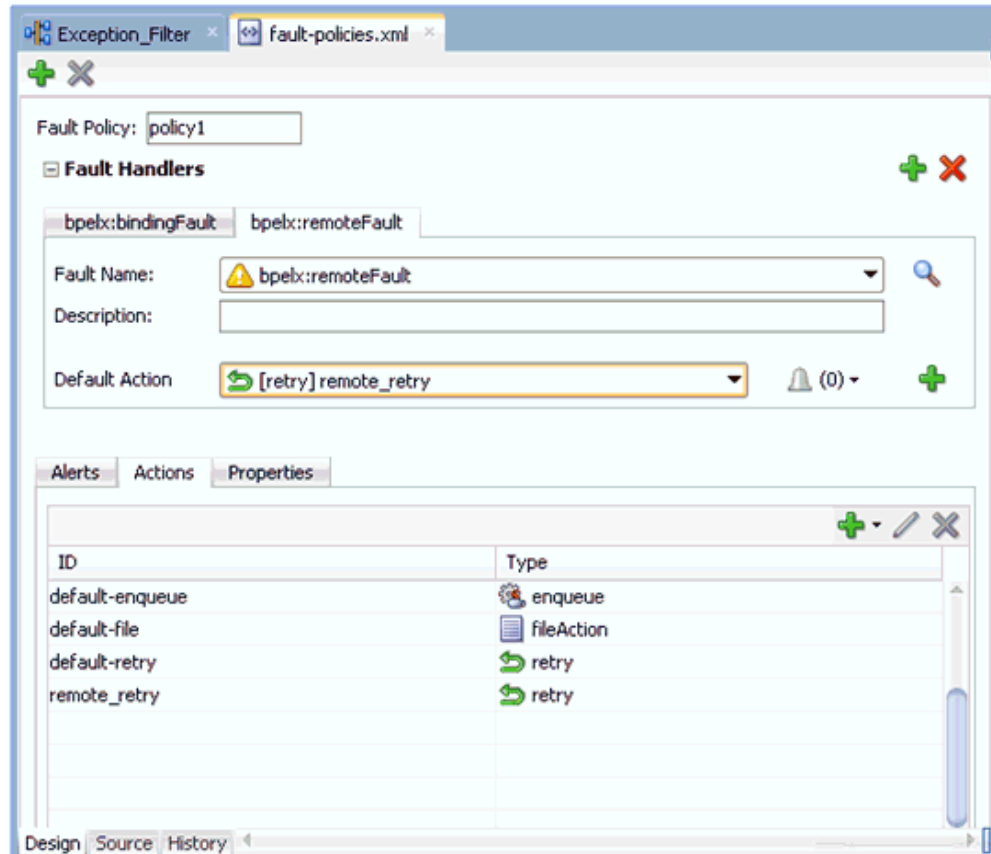


10. Provide values for the ID, Retry Count, and Retry Interval fields.
11. Select **[abort] default-termination** from the Retry Success Action drop-down list and **[humanIntervention] default-human** from the Retry Failure Action drop-down list.
12. Click **OK**.

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 11-84](#).

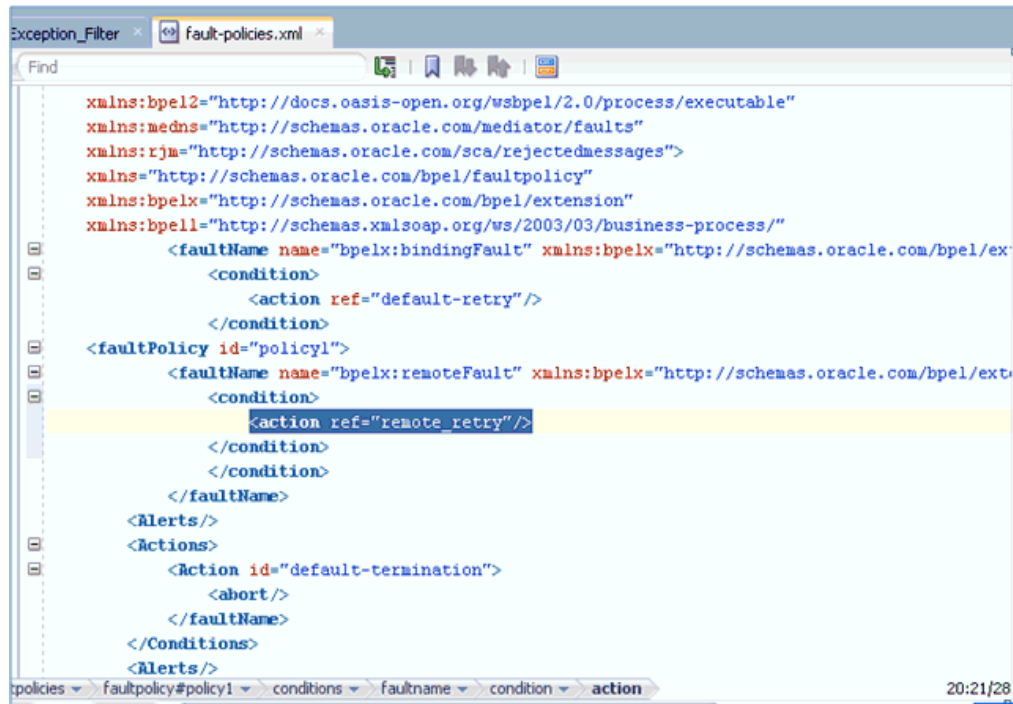
Figure 11-84 *Fault-policies.xml* Tab



13. Click **Save All**.

14. Click the **Source** tab to verify that the fault polices are added properly, as shown in [Figure 11-85](#).

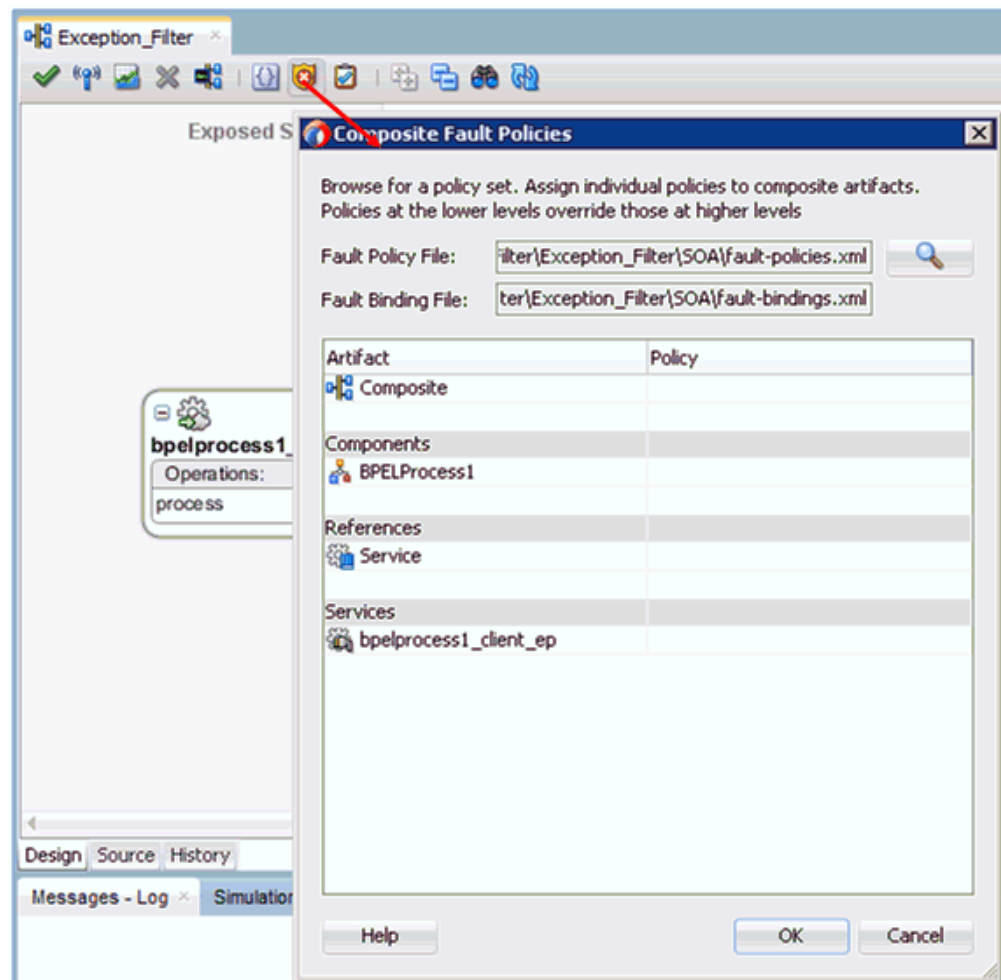
Figure 11–85 Source Tab



15. Double-click the **Exception_Filter** project and then click **Edit Composite Fault Policies**.

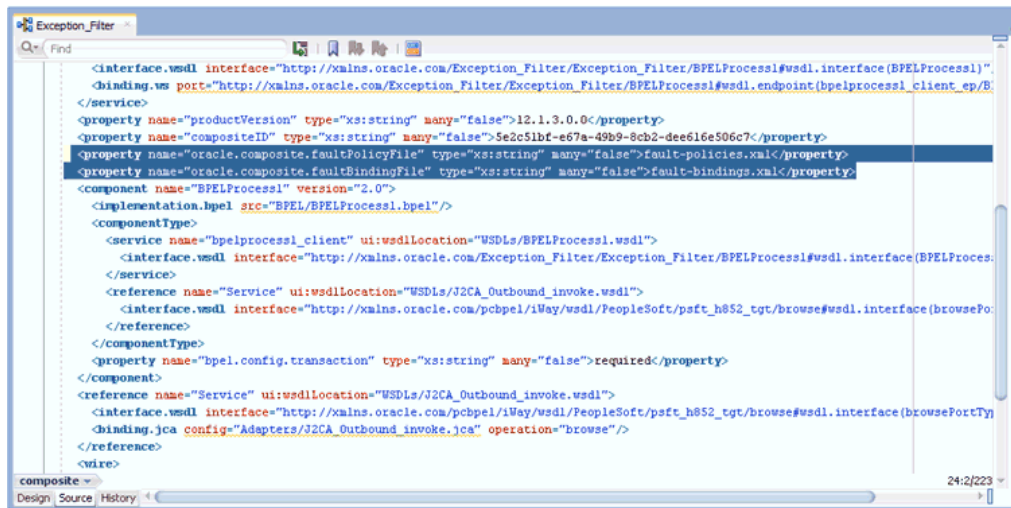
The Composite Fault Policies window is displayed. Ensure that the Fault Policy and the fault-bindings are selected properly, as shown in [Figure 11–86](#).

Figure 11–86 Composite Fault Policies Window



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 11–87.

Figure 11–87 Source Tab



11.4.1.4 Adjusting for Known Deployment Issues With 12c

For more information on how to adjust for known deployment issues with 12c, see [Section 6.4.3.3, "Adjusting for Known Deployment Issues With 12c"](#) on page 6-27.

11.4.1.5 Deploying and Testing the BPEL Outbound Process With Exception Filter Functionality

To deploy and test the BPEL outbound process with exception filter functionality:

1. Deploy the created BPEL outbound process.

For more information, see ["Deploying the BPEL Outbound Process"](#) on page 6-30.
2. Simulate a communication error by disconnecting the system (where the servers are running) from the network.
3. Invoke the deployed BPEL outbound process with a valid input.

For more information, see ["Invoking the Input XML Document in the Oracle Enterprise Manager Console"](#) on page 6-33.
4. Select the process ID.

You can observe the BPEL outbound process being retried or aborted based on the configuration of the `fault-policies.xml` file.

11.5 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 SAP R/3 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 11.5.1, "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 SAP R/3 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 (SAP R/3 adapter). 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 (SAP R/3).

11.5.1 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 5, "Oracle WebLogic Server Deployment and Integration"](#).

2. Associate Oracle WebLogic Server credentials with EIS credentials.

For more information, see [Section 11.5.1.1, "Associating Oracle WebLogic Server Credentials With EIS Credentials"](#) on page 11-58.

3. Generate a WSDL file.

For more information, see [Section 11.5.1.2, "Generating a WSDL File"](#) on page 11-61.

4. Create and deploy an outbound process.

For more information, see [Section 11.5.1.3, "Creating and Deploying an Outbound Process"](#) on page 11-61.

5. Invoke and verify that the EIS credentials have passed.

For more information, see [Section 11.5.1.4, "Verifying the EIS Credentials"](#) on page 11-62.

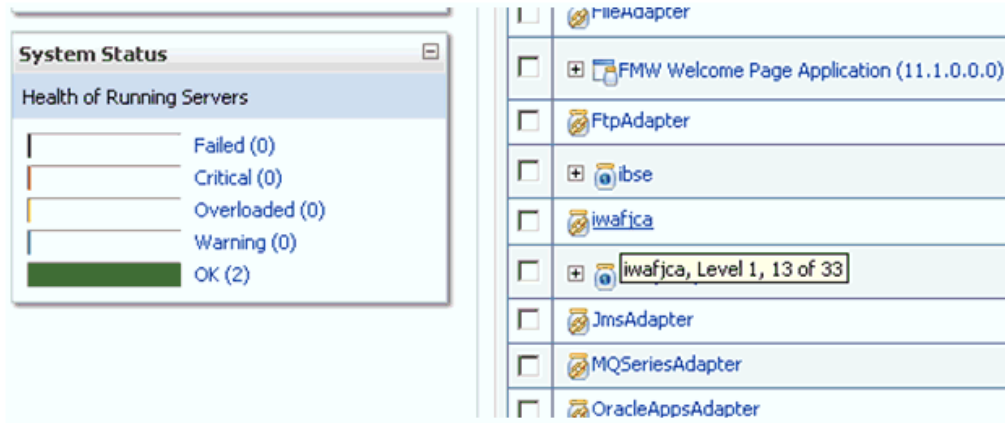
11.5.1.1 Associating Oracle WebLogic Server Credentials With EIS Credentials

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**.

The Deployments page is displayed, as shown in [Figure 11-88](#).

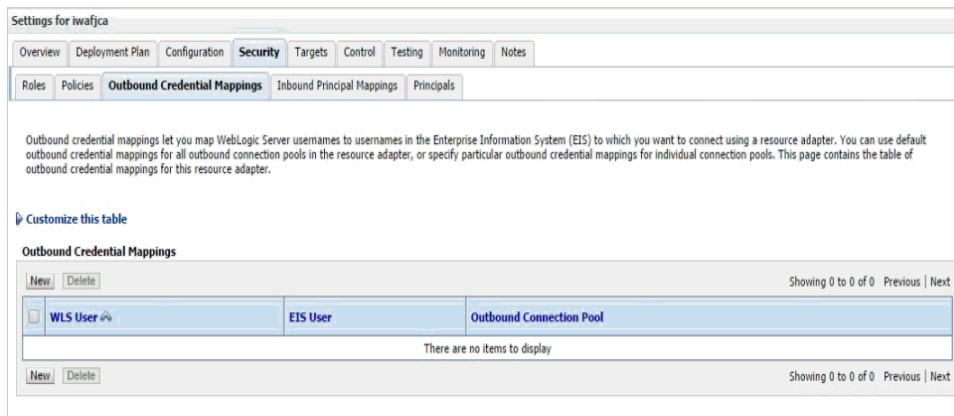
Figure 11–88 Deployments Page



3. Click the **iwafjca** resource adapter.

The Settings for iwafjca page is displayed, as shown in [Figure 11–89](#).

Figure 11–89 Settings for iwafjca Page



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 11–90](#).

Figure 11–90 Create a New Security Credential Mapping Page

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 11–91](#).

Figure 11–91 WebLogic Server User Page

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 11–92](#).

Figure 11–92 EIS User Name and Password Page

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 (SAP R/3). The mapping is invoked automatically before invoking the J2CA service.

11.5.1.2 Generating a WSDL File

To generate a WSDL file:

1. Open Application Explorer and create a J2CA configuration.
For more information, see [Section 4.4.2, "Creating a Configuration for J2CA"](#) on page 4-6.
2. Create a target for the SAP R/3 adapter and then connect to the target.
For more information, see [Section 4.5, "Establishing a Connection \(Target\) for SAP R/3"](#) on page 4-8.
3. Generate an outbound WSDL for the appropriate object.
For more information, see [Section 4.8, "Generating WSDL \(J2CA Configurations Only\)"](#) on page 4-18.

11.5.1.3 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 7, "Integration With Mediator Service Components in the Oracle SOA Suite"](#).

For more information about creating a BPM outbound process, see [Chapter 8, "Integration With BPM Service Components in the Oracle SOA Suite"](#).

To create and deploy a BPEL outbound process, see the following sections:

- [Section 6.4.2, "Creating an Empty Composite for SOA"](#)

- [Section 6.4.3, "Defining a BPEL Outbound Process"](#)
- [Section 6.4.4, "Deploying the BPEL Outbound Process"](#)

11.5.1.4 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 ["Invoking the Input XML Document in the Oracle Enterprise Manager Console"](#) on page 6-33.

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 IWAManagedConnectionFactory com.ibi.afjca.Util
getPasswordCredential(78) InLoop:
User-iwayqa:Password-ENCR(3109311731831131382333215315332323192322731773172)
FINEST IWAManagedConnectionFactory com.ibi.afjca.Util
getPasswordCredential(90) Use the system PasswordCredential:
User-iwayqa:Password-ENCR(3109311731831131382333215315332323192322731773172)
```

11.6 Credential Mapping for Oracle Service Bus (OSB) Using JDeveloper

This section describes how to configure credential mapping functionality for the Oracle Application Adapter for SAP R/3 in a configuration that uses Oracle Service Bus (OSB). A sample testing scenario is also included. This section contains the following topic:

- [Section 11.6.1, "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 SAP R/3 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 (SAP R/3 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 (SAP R/3).

11.6.1 Configuring Credential Mapping

This section discusses configuring credential mapping, and consists of the following steps and topics:

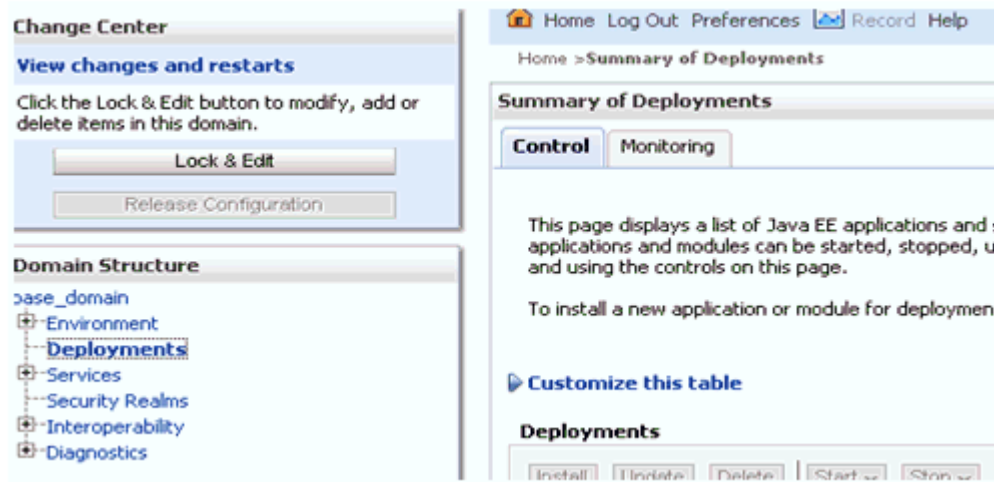
1. Deploy the adapter.
For more information, see [Chapter 5, "Oracle WebLogic Server Deployment and Integration"](#).
2. Associate Oracle WebLogic Server credentials with EIS credentials.
For more information, see [Section 11.6.1.1, "Associating Oracle WebLogic Server Credentials With EIS Credentials"](#).
3. Generate a WSDL file.
For more information, see [Section 11.6.1.2, "Generating a WSDL File"](#).
4. Create an Oracle Service Bus (OSB) outbound process.
For more information, see [Section 11.6.1.3, "Creating an Oracle Service Bus \(OSB\) Outbound Process"](#).

11.6.1.1 Associating Oracle WebLogic Server Credentials With EIS Credentials

To associate Oracle WebLogic Server credentials with EIS credentials:

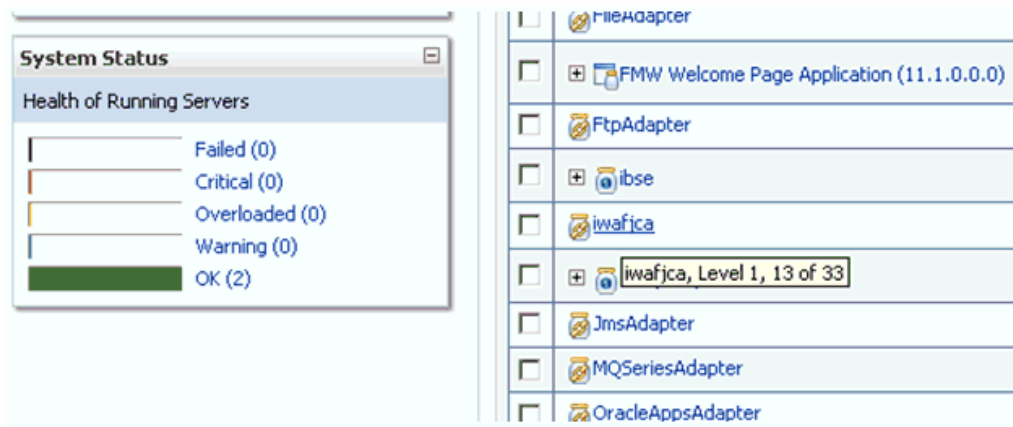
1. Log in to the Oracle WebLogic Server Administration Console.
2. In the Domain Structure section in the left pane, click **Deployments**, as shown in [Figure 11–93](#).

Figure 11–93 Domain Structure Section



The Deployments page is displayed, as shown in [Figure 11–94](#).

Figure 11–94 Deployments Page



3. Click the `iwafjca` resource adapter.

The Settings for `iwafjca` page is displayed, as shown in [Figure 11–95](#).

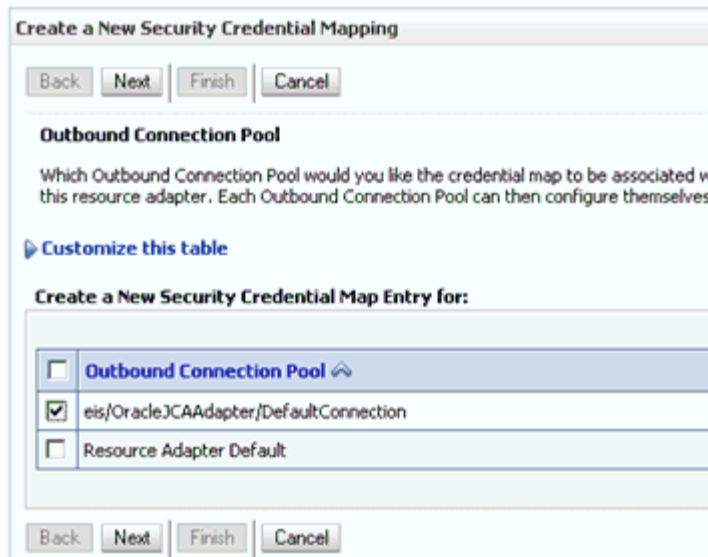
Figure 11–95 Settings for iwafjca Page



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 11–96](#).

Figure 11–96 Create a New Security Credential Mapping Page



5. Select the outbound connection pool.

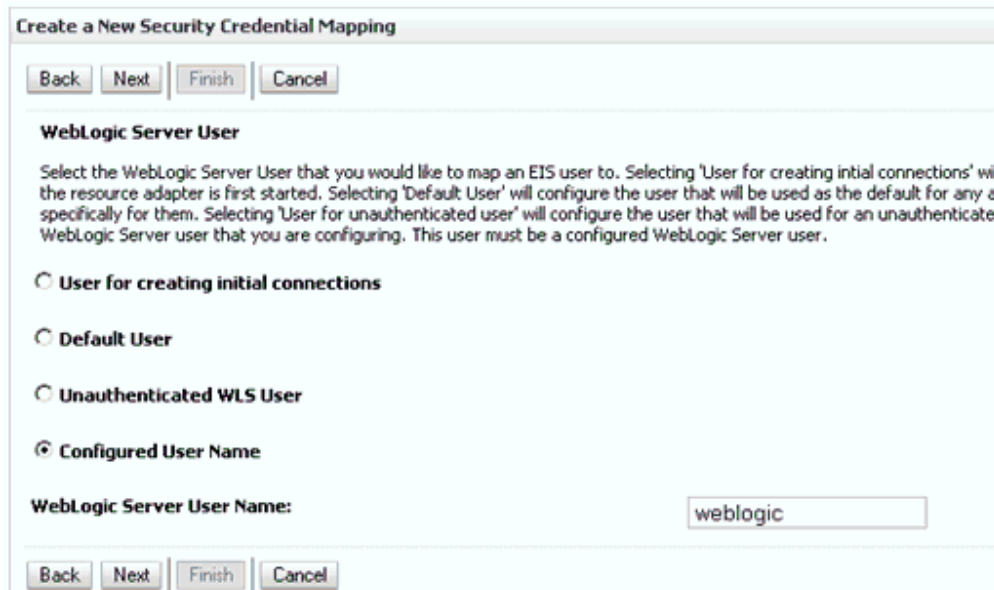
For example:

`eis/OracleJCAAdapter/DefaultConnection`

6. Click **Next**.

The WebLogic Server User page is displayed, as shown in [Figure 11–97](#).

Figure 11–97 WebLogic Server User Page



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 11–98](#).

Figure 11–98 EIS User Name and Password Page

Create a New Security Credential Mapping

Back Next Finish Cancel

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::

Enter the EIS Password:

* EIS Password::

* Confirm Password::

Back Next Finish Cancel

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 (SAP R/3). The mapping is invoked automatically before invoking the J2CA service.

11.6.1.2 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 9-6.
2. Open Application Explorer and create a J2CA configuration.
For more information, see ["Creating a Configuration for J2CA"](#) on page 4-6.
3. Create a target for the SAP R/3 adapter and then connect to the target.
For more information, see ["Establishing a Connection \(Target\) for SAP R/3"](#) on page 4-8.
4. Publish a WSDL to OSB for the appropriate object (for example, the GetDetail BAPI for the CompanyCode business object).
For more information, see ["Publishing a WSDL From Application Explorer to Oracle Service Bus"](#) on page 9-6.

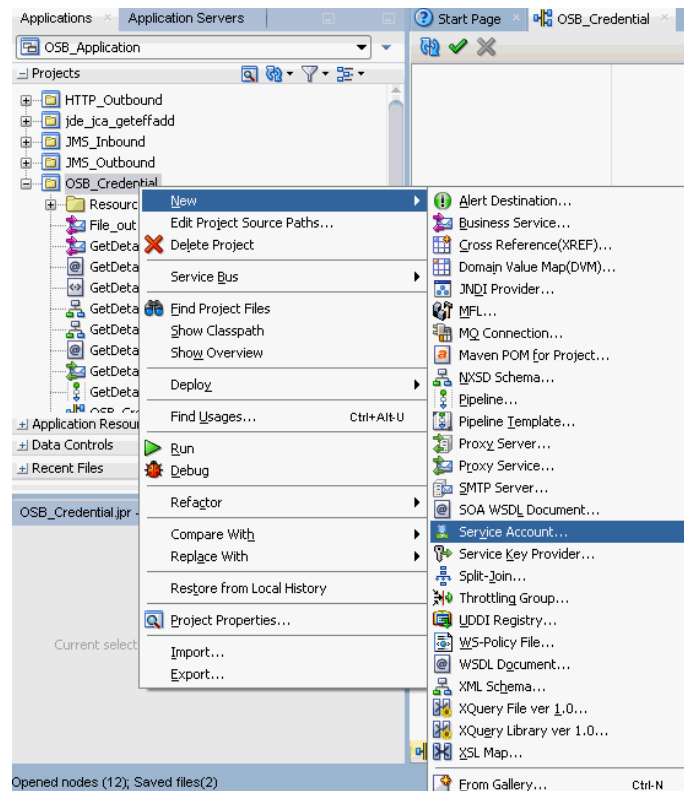
11.6.1.3 Creating an Oracle Service Bus (OSB) Outbound Process

This section describes how to create an Oracle Service Bus (OSB) outbound process with credential mapping functionality, using Oracle JDeveloper.

1. Start the Oracle JDeveloper and create a Service Bus Application for OSB.
For more information, see [Section 10.2.1, "Creating a Service Bus Application for OSB"](#) on page 10-20.

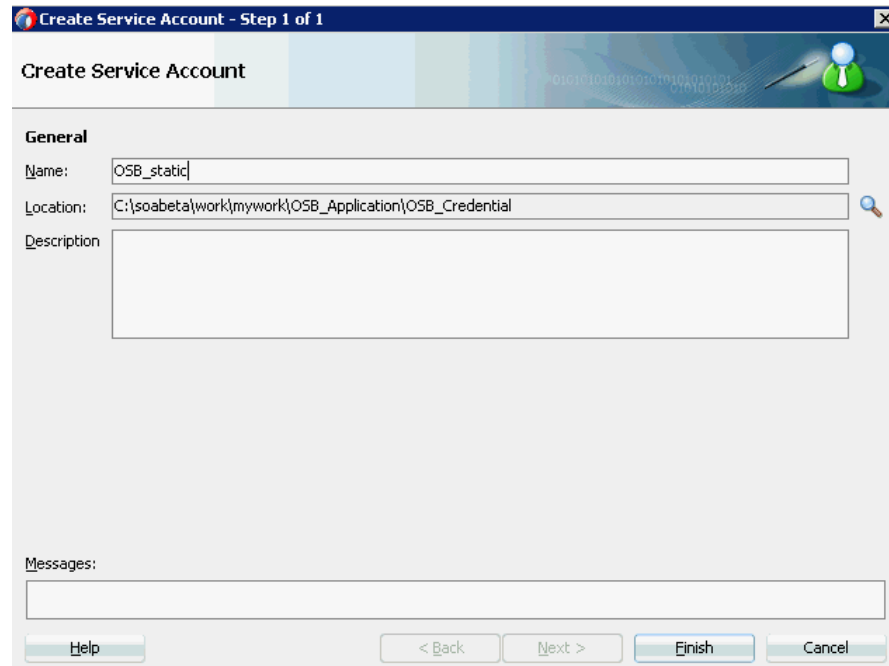
2. Create a Third Party Adapter Service Component.
For more information, see [Section 10.2.2.1, "Configuring a Third-Party Adapter Service Component"](#) on page 10-21.
3. Create a WSDL-based Business Service from the JCA Binding File.
For more information, see [Section 10.3.2.1, "Configuring a WSDL-based Business Service"](#) on page 10-31.
4. Create a File Transport Type Business Service for an output destination.
For more information, see [Section 10.2.2.3, "Configuring a File Transport Type Business Service"](#) on page 10-27.
5. Create a Proxy Service with Pipeline.
For more information, see [Section 10.3.2.2, "Creating a Proxy Service With Pipeline"](#) on page 10-35.
6. Configure the routing rules in the created Pipeline.
For more information, see [Section 10.2.2.4, "Configuring the Routing Rules"](#) on page 10-29
7. Configure a Service account by right-clicking the OSB Project, selecting **New**, and then clicking **Service Account**, as shown in [Figure 11–99](#).

Figure 11–99 Select Service Account Option



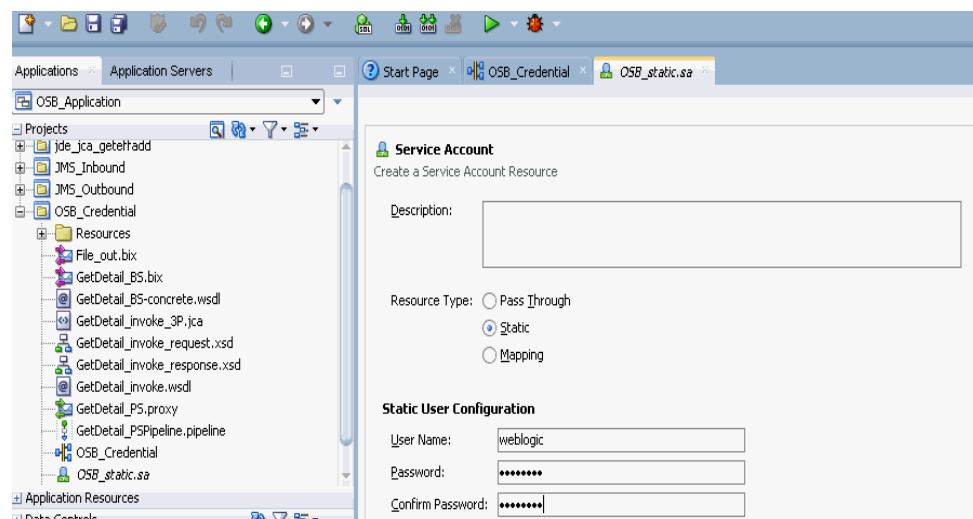
The Create Service Account pane is displayed, as shown in [Figure 11–100](#).

Figure 11–100 Create Service Account Pane



8. Provide a name for the Service Account and click **Finish**.
The configuration page of Service Account is displayed.
9. In the Resource Type section, select **Static**.
10. Provide a valid user name and password for the Oracle WebLogic Server, as shown in [Figure 11–101](#).

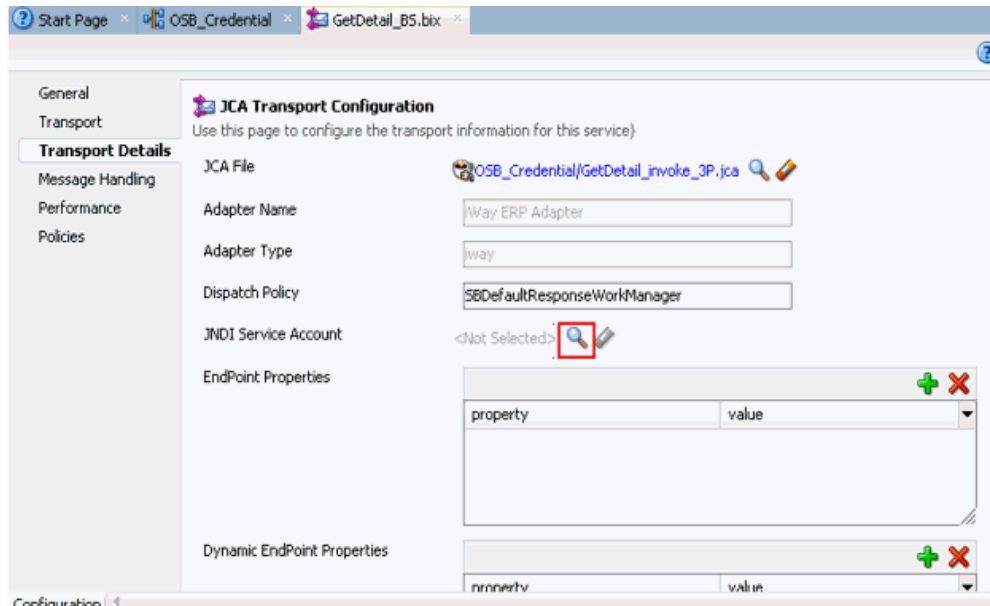
Figure 11–101 Service Account Configuration Page



11. Save and close the configuration page.
12. 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.

13. Select the Transport Details tab, as shown in [Figure 11–102](#).

Figure 11–102 Transport Details Tab

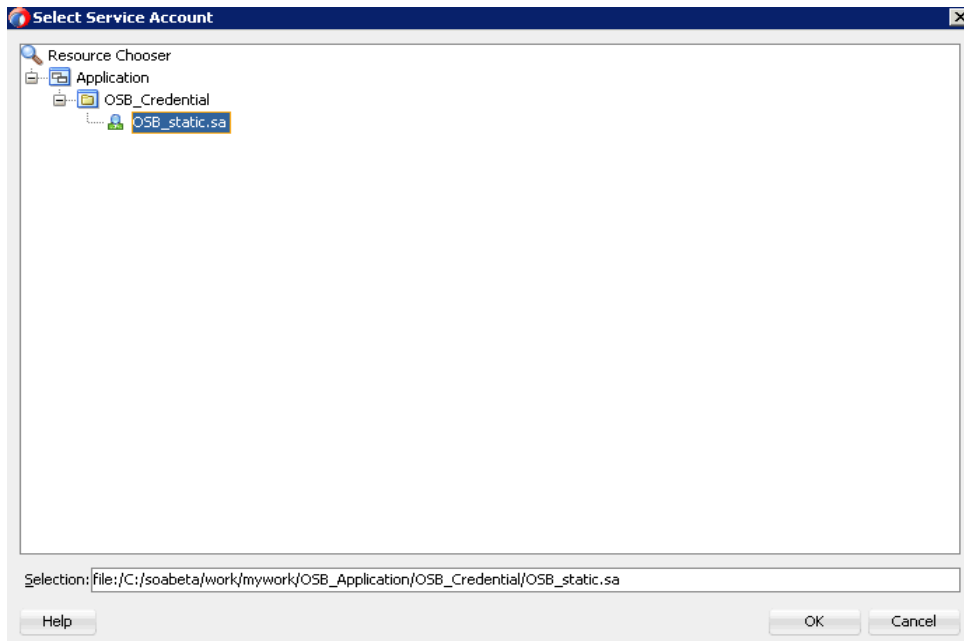


14. In the JNDI Service Account section, click the Browse icon.

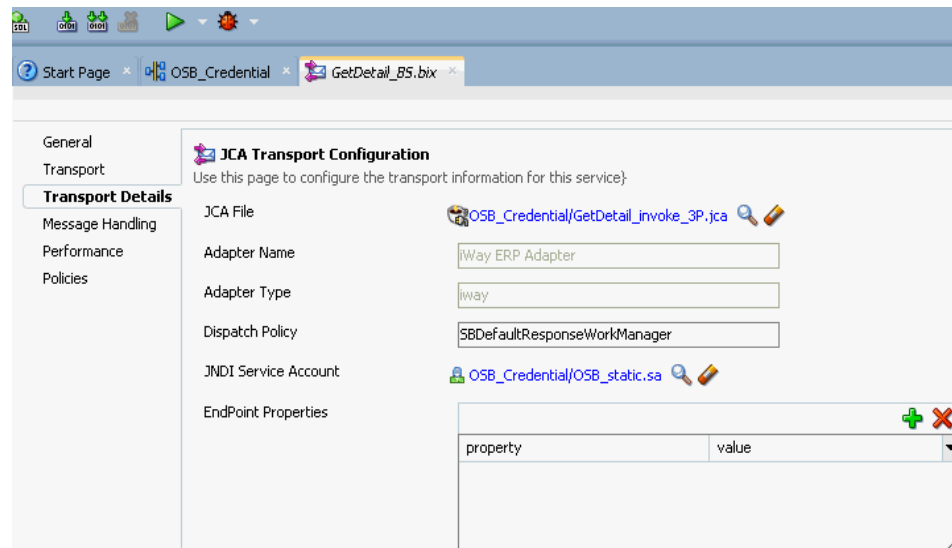
The Select Service Account window is displayed.

15. Select the created service account and click **OK**, as shown in [Figure 11–103](#).

Figure 11–103 Select Service Account



16. Save and close the configuration page, as shown in [Figure 11–104](#)

Figure 11–104 Business Service Configuration Page**17. Deploy the OSB process.**

For more information, see [Section 10.1.3, "Deploying the OSB Outbound Process"](#) on page 10-17.

18. 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.**19. 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)
```

11.7 Filtering Unicode Characters From Text

This section describes how to filter Unicode characters from text that are not valid for XML 1.0. It includes the following topics:

- [Section 11.7.1, "Overview"](#)
- [Section 11.7.2, "Enabling the Filter"](#)
- [Section 11.7.3, "Specifying the Replacement Character"](#)
- [Section 11.7.4, "Saving the Settings File and Restarting the Server"](#)
- [Section 11.7.5, "Validating the Filter"](#)
- [Section 11.7.6, "Quick Start Guide"](#)
- [Section 11.7.7, "Known Limitations"](#)

11.7.1 Overview

A Unicode special character filter for characters outside of the acceptable XML 1.0 parser specification that occur in SAP IDoc data is available. The filter screens certain Unicode characters from SAP to the adapter where exceptions can occur within the XML parser. The filter then removes the characters from the data stream before serialization, so no errors occur during parsing.

This feature implements a filter for the following significant invalid XML characters:

- VT: Vertical Tab, U+000B
- FF: Form Feed, U+000C
- CR: Carriage Return, U+000D
- NEL: Next Line, U+0085
- LS: Line Separator, U+2028
- PS: Paragraph Separator, U+2029

The following characters are explicitly not to be filtered:

- LF: Line Feed, U+000A
- CR+LF: CR (U+000D) followed by LF (U+000A)

This feature will replace all instances of designated Unicode characters in SAP outbound / Oracle inbound IDocs with a specific replacement character. There is one system-wide replacement character per adapter installation. This feature is controlled by a settings file and a designated replacement character.

11.7.2 Enabling the Filter

Open the `mysap.settings.xml` file using a XML or text editor. The `mysap.settings.xml` file is located in the following directory:

```
<ADAPTER_HOME>\lib\settings
```

The `filtering` tag controls the filter behavior. Setting `enabled` to `on` will filter incoming SAP data. Setting `enabled` to `off` disables the filter behavior. Do not enter any other values in the `enabled` field.

11.7.3 Specifying the Replacement Character

The `substitutionChar` field controls the character that will appear in place of the invalid character or characters in an IDoc.

The substitution character should be unique enough so that it does not occur naturally in the data, keeping in mind that the # (pound or hash) character is the SAP substitution character. You can use the # (pound or hash) character as a replacement character, keeping in mind that SAP itself may use the same character on occasion. The SAP replacement character is set in the RFC Destination maintenance transaction (SM59) on the SAP server, under the `MDMP & Unicode` tab.

The following is a sample file:

```
<mysap>
  <filtering>
    <enabled>on</enabled>
    <substitutionChar>'#</substitutionChar>
  </filtering>
</mysap>
```

Note: If no substitution character is specified in the *mysap.settings.xml* file within the `<substitutionChar></substitutionChar>` element tag, then a space will be used as the default substitution character.

11.7.4 Saving the Settings File and Restarting the Server

Save the changes made to the *mysap.settings.xml* file using the chosen application editor. The Oracle Application Adapter for SAP R/3 adapter locates the settings file automatically upon activation.

Restart the servers to activate the changes made to the *mysap.settings.xml* file.

11.7.5 Validating the Filter

When the adapter is activated, if *DEBUG* level is enabled in the log, a message indicating whether the settings file is located and whether the filter option is enabled is displayed in the log. The message is not displayed during normal trace levels.

11.7.6 Quick Start Guide

This section describes the key steps that are required to use the filter feature.

1. Browse to the *mysap.settings.xml* file, which is located in the following folder:

```
<ADAPTER_HOME>\lib\settings
```
2. Edit the *mysap.settings.xml* file and set the *enabled* option for the *filtering* tag to *on*.
3. Specify a replacement character as described in [Section 11.7.3, "Specifying the Replacement Character"](#) on page 11-71.
4. Save the *mysap.settings.xml* file.
5. Restart the server.
6. Verify whether the *DEBUG* level is enabled in the log. If enabled, then a message indicating whether the settings file is located and whether the filter option is enabled is displayed in the log.

11.7.7 Known Limitations

This section describes known limitations for the Unicode special character filter. It includes the following topics:

- [Section 11.7.7.1, "Adapter-Wide Installation"](#)
- [Section 11.7.7.2, "Replace All Restriction"](#)
- [Section 11.7.7.3, "SAP System Type"](#)
- [Section 11.7.7.4, "Character Restrictions"](#)
- [Section 11.7.7.5, "Document Type and Direction"](#)
- [Section 11.7.7.6, "Whitespace Rules Display"](#)
- [Section 11.7.7.7, "Sax Parser"](#)

- [Section 11.7.7.8, "Replacement Character Type"](#)
- [Section 11.7.7.9, "Replacement Character Number"](#)
- [Section 11.7.7.10, "Font Usage"](#)
- [Section 11.7.7.11, "Viewing Whitespace Characters"](#)

11.7.7.1 Adapter-Wide Installation

This feature operates per adapter installation and not by adapter channel target. Ensure that all configured channels for the adapter meet the requirements before deploying the feature.

11.7.7.2 Replace All Restriction

The replacement character will replace all instances of the designated replacement characters in an IDoc document. For more information, see [Section 11.7.3, "Specifying the Replacement Character"](#) on page 11-71.

11.7.7.3 SAP System Type

This feature is designed for Unicode SAP systems only. SAP MDMP systems use Code pages for languages and Unicode characters may not pass correctly through the RFC data stream without Unicode on both ends of the communication.

11.7.7.4 Character Restrictions

This feature is only enabled for characters explicitly listed in this document and no other characters. Different control characters may occur in IDoc documents and will pass through the filter because no rule exists for their handling.

11.7.7.5 Document Type and Direction

This feature is limited to SAP outbound into Oracle inbound IDocs. These documents are designed for intersystem communication and may contain characters from non-SAP or other systems that are not valid for XML communication. SAP outbound event data delivered through SAP Remote Function Calls are not filtered. Data sent from Oracle into SAP (request/response or client communication) is not impacted by this feature. The owner of the data is responsible for correctness of the data sent to SAP.

11.7.7.6 Whitespace Rules Display

Characters that have the *is Whitespace* attribute set to *true* in the Java Unicode definition can only be seen in raw text (not XML) format, using the UTF-8 hexadecimal code for the character. See the Unicode character guide using the following URL:

<http://www.fileformat.info/unicode/char/search.htm>

For more information, see [Section 11.7.7.11, "Viewing Whitespace Characters"](#) on page 11-74.

11.7.7.7 Sax Parser

The character sequence *CR+LF* is never filtered, even if it occurs in the middle of the IDoc text. When the XML IDoc is viewed in a text editor, only the *LF* character is found, is usually done by the Sax parser. For more information, see the following rule:

<http://www.w3.org/TR/REC-xml/#sec-line-ends>

This rule translates $U+000D+U+000A$ to a single $U+000A$ character. The character sequence $U+000D$ is filtered by the adapter.

11.7.7.8 Replacement Character Type

The replacement character must come from the standard set of characters as defined in ISO/IEC 646, and not be a C0 or C1 control character itself. The control character is usually chosen from the ASCII *punctuation and symbols* set and must be a single byte character on representation. The substitution character must be enclosed in single quote marks.

11.7.7.9 Replacement Character Number

There can be only one substitution character in a filter file for an adapter installation.

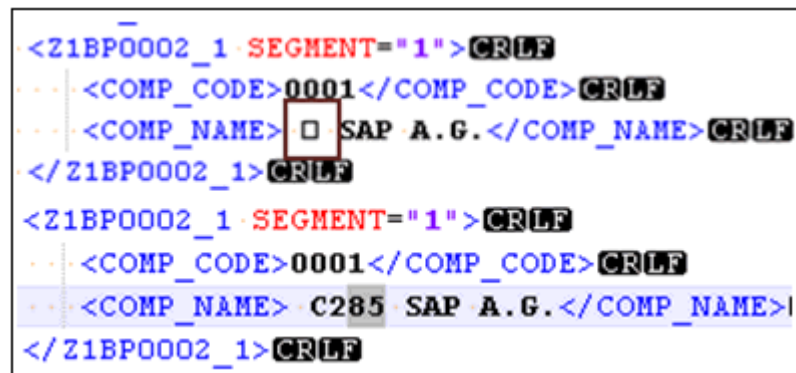
11.7.7.10 Font Usage

Ensure that the system receiving the data has the correct language packs and character sets for the data to be received. If an attempt to display character data not available in a font on the receiving system is sent, then a box character may appear indicating the raster or printer font has no character point to render the character to be displayed.

11.7.7.11 Viewing Whitespace Characters

Character $U+0085$ (NextLine) can only be seen by examining the XML in a special text editor, highlighting the blank area where the character is supposed to occur, and receiving the hex value, which displays as $c285$ in UTF-8 hexadecimal, as shown in Figure 11-105.

Figure 11-105 Viewing Whitespace Characters



```

<Z1BP0002_1 SEGMENT="1">CRLF
  <COMP_CODE>0001</COMP_CODE>CRLF
  <COMP_NAME> □ SAP A.G.</COMP_NAME>CRLF
</Z1BP0002_1>CRLF

<Z1BP0002_1 SEGMENT="1">CRLF
  <COMP_CODE>0001</COMP_CODE>CRLF
  <COMP_NAME> C285 SAP A.G.</COMP_NAME>
</Z1BP0002_1>CRLF

```

Otherwise, the XML parser directive is to ignore the character as whitespace, and so will not display in the file.

Troubleshooting and Error Messages

This chapter explains the limitations and workarounds when connecting to SAP R/3. It contains the following sections:

- [Section 12.1, "Troubleshooting"](#)
- [Section 12.2, "BSE Error Messages"](#)

The adapter-specific errors listed in this chapter can arise whether using the adapter with an Oracle Adapter J2CA or with an Oracle Adapter Business Services Engine (BSE) configuration.

12.1 Troubleshooting

This section provides troubleshooting information for SAP R/3, separated into four topics:

- [Section 12.1.1, "Application Explorer"](#)
- [Section 12.1.2, "SAP R/3"](#)
- [Section 12.1.3, "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
```

```

Note: SEVERE
IWAManagedConnectionFactory
com.iwaysoftware.af.container.channel.Channel
onMessage(277) No subscribers for this channel.
Event thrown away. message:
java.lang.IllegalStateException: Message failed to
deliver to any listener
    
```

The above error means a channel in the Application Explorer is in an illegal state to receive messages, hence the exception.

Perform the following to resolve the issue:

- Stop all the channels and close the Application Explorer before run-time deployment for events.
- If an instance of AE is running and no BPM, BPEL, or OSB Process is deployed to handle a message, IDocs sent from SAP will be returned to SAP with an Illegal State Exception and can be re-sent from transaction SM58 once a proper process handler is deployed.

12.1.1 Application Explorer

This topic discusses the different types of errors that can occur when using Application Explorer.

Error	Solution
SAP R/3 does not appear in the Application Explorer Adapter node list.	Ensure that the <code>sapjco3.jar</code> and <code>sapjco3.dll</code> files are added to the <code>lib</code> directory.
Cannot connect to Oracle Application Adapter for SAP R/3 from Application Explorer. Problem activating adapter	Ensure that: <ul style="list-style-type: none"> ■ SAP R/3 is running. ■ The Application Server name, System Number, and Client Number are correct. ■ The SAP R/3 user ID and password are correct.
Cannot connect to the SAP R/3 target through Application Explorer. The following error message appears: Error getting target [SAP] - java.lang.Exception: Error Logon to SAP System	Ensure that you enter the correct connection parameters when connecting to the SAP R/3 target.

Error	Solution
<p>Cannot connect to your SAP R/3 system through Application Explorer. The following error message appears:</p> <pre> Problem activating adapter. (com.ibi.sap3.SapAdapterException : com.sap.mw.jco.JCO\$Exception: (102) RFC_ERROR_COMMUNICATION: Connect to SAP gateway failed Connect_PM GWHOST=isdsvr8, GWSERV=sapgw00, ASHOST=isdsvr8, SYSNR=00 LOCATION CPIC (TCP/IP) on local host ERROR partner not reached (host isdsvr8, service 3300) TIME Fri Aug 27 11:49:14 2004 RELEASE 620 COMPONENT NI (network interface) VERSION 36 RC -10 MODULE ninti.c LINE 979 DETAIL NiPConnect2 SYSTEM CALL SO_ERROR ERRNO 10061 ERRNO TEXT WSAECONNREFUSED: Connection refused COUNTER 1). Check logs for more information </pre>	<p>Ensure that SAP R/3 is running and that you are using the correct parameter values to connect to your application server.</p>
<p>Cannot connect to your SAP R/3 system through Application Explorer even though SAP R/3 is running. The following error message appears:</p> <pre> Problem activating adapter. (com.ibi.sap3.SapAdapterException : java.lang.ExceptionInInitializerEr ror: JCO.classInitialize(): Could not load middleware layer 'com.sap.mw.jco.rfc.MiddlewareRFC' JCO.nativeInit(): Could not initialize dynamic link library sapjco3 [no sapjco3 in java.library.path]. java.library.path </pre>	<p>Ensure that the <code>sapjco3.dll</code> file is added to the lib directory and to the Windows system32 folder.</p>
<p>The dll is loaded in another classloader (BSE and J2CA are installed on the same server). The following error message appears:</p> <pre> com.ibi.sap3.SapAdapterException: java.lang.ExceptionInInitializerEr ror: JCO.classInitialize(): Could not load middleware layer 'com.sap.mw.jco.rfc.MiddlewareRFC' JCO.nativeInit(): Could not initialize dynamic link library sapjco3 [Native Library F:\iWay55.008.0628\lib\sapjco3.dll loaded in another classloader]. java.library.path </pre>	<p>Add <code>sapjco3.jar</code> to the server classpath.</p>

Error	Solution
<p>Unable to start Application Explorer in a Solaris environment. The following exception is thrown in the console:</p> <pre> javax.resource.ResourceException: IWAManagedConnectionFactory: License violation.at com.ibi.afjca.spi.IWAManagedConne ctionFactory.createConnectionFacto ry(IWAManagedConnectionFactory.ja va:98)at com.iwaysoftware.iwae.common.JCATr ansport.getConnectionFactory(JCATr ansport.java:133) at com.iwaysoftware.iwae.common.JCATr ansport.initJCA(JCATransport.java: 69)at com.iwaysoftware.iwae.common.JCATr ansport.<init>(JCATransport.java:6 2)at com.iwaysoftware.iwae.common.Adapt erClient.<init>(AdapterClient.java :85)at com.ibi.bse.ConfigWorker.run(Confi gWorker.java:41)at java.lang.Thread.run(Thread.java:5 34) Could not create the connection factory.</pre>	<p>JAVACMD is not set on the user system. Before starting Application Explorer, export JAVACMD as follows:</p> <p>JAVACMD=/<code><jdk_home></code>/bin/java, where <code><jdk_home></code> is the directory where JDK is installed on your system.</p>
<p>Logon failure error at run-time</p>	<p>If the password for connecting to your SAP R/3 system is not specified when creating a target or with the Edit option in Application Explorer, then you are unable to connect to SAP R/3. The connection password is not saved in <code>repository.xml</code>. Update the password using the Edit option in Application Explorer, then restart the application server.</p>
<p>The following exception occurs when you start Application Explorer by activating <code>ae.bat</code> (not <code>iaexplorer.bat</code>):</p> <pre> java.lang.ClassNotFoundException: org.bouncycastle.jce.provider.Boun cyCastleProvider</pre>	<p>This is a benign exception. It does not affect adapter functionality. Download BouncyCastle files from:</p> <p>ftp://ftp.bouncycastle.org/pub</p>

12.1.2 SAP R/3

Error	Solution
<p>When executing a request, the following error message appears:</p> <pre> AdapterException: java.lang.Exception: Function module CUSTOMER_GETDETAIL2 does NOT exist.</pre>	<p>Check the syntax of your input XML document and make sure the name of the Remote Function module is correct, available in SAP R/3, and activated.</p>

Error	Solution
<p>When executing a request, the following error message appears:</p> <pre>AdapterException: java.lang.Exception: Object type unknown for business object: CUST</pre>	<p>Check the syntax of your input XML document and verify that the Business Object type exists in SAP R/3 and is correct and activated. Also verify that the name of your document is correct.</p>
<p>When executing a request, the following error message appears:</p> <pre>AdapterException: java.lang.Exception: Unable to retrieve BAPI name for: CUSTOMER.DETAIL2</pre>	<p>Check the syntax of your input XML document and verify that the name of the BAPI is correct, available in SAP R/3, and activated.</p>
<p>When executing a request, the following error message appears:</p> <pre>java.lang.RuntimeException: com.sap.mw.jco.JCO\$AbapException: (126) OBJECT_UNKNOWN: Basic type or extension does not exist.</pre>	<p>Check the syntax of your input XML document and verify that the IDoc type or extension type is correct, available in SAP R/3, and activated.</p>
<p>When executing a request, the following error message appears:</p> <pre>AdapterException: java.lang.Exception: BapiError/BapiAbort: You are not authorized to display customers.</pre>	<p>Verify that your user ID has the correct permissions configured in SAP R/3. Consult your SAP R/3 administrator for more information.</p>

Note: Activation is an important SAP concept. If an object does not exist in an activated state, it may appear as present on the system, but is not available for use. This is especially important for IDOCs and SAP Business Objects. Consult your SAP documentation for further information.

12.1.3 Oracle Adapter J2CA

Error	Solution
<p>In Application Explorer, the following error message appears when you attempt to connect to an Oracle Adapter J2CA configuration:</p> <pre>Could not initialize JCA</pre>	<p>In the Details tab in the right pane, ensure that the directory specified in the Home field points to the correct directory, for example:</p> <pre><ADAPTER_HOME>\tools\iwae\bin\..\..\..\</pre>

12.2 BSE Error Messages

This topic discusses the different types of errors that can occur when processing Web services through BSE.

12.2.1 General Error Handling in BSE

BSE serves as both a SOAP gateway into the adapter framework and as the engine for the adapters. In both design-time and run-time, various conditions can cause errors in BSE when Web services that use adapters run. 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 if you encounter 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.

12.2.2 Adapter-Specific Error Handling

This section contains the following topics:

- [Section 12.2.2.1, "Oracle Application Adapter for SAP R/3 Invalid SOAP Request"](#)
- [Section 12.2.2.2, "Empty Result From SOAP Request"](#)
- [Section 12.2.2.3, "Failure to Connect to SAP R/3"](#)
- [Section 12.2.2.4, "Invalid SOAP Request"](#)
- [Section 12.2.2.5, "Empty Result From Oracle Application Adapter for SAP R/3 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.

12.2.2.1 Oracle Application Adapter for SAP R/3 Invalid SOAP Request

If Oracle Application Adapter for SAP R/3 receives a SOAP request message that does not conform to the WSDL for the Web services being carried out, 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>Error processing agent [XDSapIfrAgent] - XD[FAIL] SapIFRException:
java.sql.SQLException:
com.ibi.sapjco3.SapCallableStatement: execute() j
java.util.NoSuchElementException</faultstring>
  </SOAP-ENV:Fault>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

12.2.2.2 Empty Result From SOAP Request

If Oracle Application Adapter for SAP R/3 carries out an SAP R/3 object using input parameters passed in the SOAP request message that do not match records in SAP R/3, 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>Error processing agent [XDSapIfrAgent] - XD[FAIL] SapIFRException:
java.sql.SQLException: com.ibi.sapjco3.SapCallableStatement: execute()
java.sql.SQLException: JCO Error Key: NO_RECORD_FOUND Short Description:
com.sap.mw.jco.JCO$AbapException: (126) NO_RECORD_FOUND: NO_RECORD_
FOUND</faultstring>
  </SOAP-ENV:Fault>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

12.2.2.3 Failure to Connect to SAP R/3

If Oracle Application Adapter for SAP R/3 cannot connect to SAP R/3 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>Error processing agent [XDSapIfrAgent] - XD[RETRY]
Connect to SAP gateway failed Connect_PM GWHOST=ESDSUN, GWSERV=sapgw00,
ASHOST=ESDSUN,
SYSNR=00 LOCATION CPIC (TCP/IP) on local host ERROR partner not reached (host
ESDSUN, service 3300)
TIME Mon Jun 30 16:01:02 2003 RELEASE 620 COMPONENT NI (network interface) VERSION
36 RC -10 MODULE ninti.c LINE 976 DETAIL NipConnect2
SYSTEM CALL SO_ERROR ERRNO 10061 ERRNO TEXT WSAECONNREFUSED: Connection refused
COUNTER 1</faultstring>
  </SOAP-ENV:Fault>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

12.2.2.4 Invalid SOAP Request

If Oracle Application Adapter for SAP R/3 receives a SOAP request message that does not conform to the WSDL for the Web services being carried out, 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>
```

12.2.2.5 Empty Result From Oracle Application Adapter for SAP R/3 Request

If Oracle Application Adapter for SAP R/3 carries out 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>
```

Understanding the SAP System Role in Remote Processing

The Oracle Application Adapter for SAP R/3 exchanges messages with SAP using three different message type scenarios and two communication roles. Each role has configuration options that must be enabled on the SAP server and on the adapter to enable communications and send messages. This appendix explains the role types, message types, and their related terms. In an SAP Productive system environment, only the proper administrator for a particular configuration role has the authority to change or create the communications entries. This appendix provides a checklist for administrators for the required tasks and describes the entire process of configuring inbound and outbound message processing. This information can be used to configure non-productive systems or for reference purposes.

This appendix contains the following sections:

- [Section A.1, "Functionality of the Adapter"](#)
- [Section A.2, "SAP Inbound Message Processing"](#)
- [Section A.3, "SAP Outbound Message Processing"](#)

A.1 Functionality of the Adapter

This section contains the following topics:

- [Section A.1.1, "Roles"](#)
- [Section A.1.2, "Client"](#)
- [Section A.1.3, "Server"](#)

Implementation of the Oracle Application Adapter for SAP R/3 uses the SAP Java Connector (JCo), a Java interface that communicates via Java Native Interface (JNI) to a native C communications layer. The RFC (Remote Function Call) API, implements the CPIC (Common Programming Interface - Communications) and TCP/IP protocols to communicate with a corresponding component on the SAP system. Each message is serialized to or from XML to RFC format, then marshaled into the current SAP message type for transmission to SAP. The SAP server unmarshals the message into the current runtime object and executes the object. The response message is marshaled into RFC and returned to the adapter for unmarshalling and serialization to XML. This process is seamless and very rapid. Errors can arise from improper credentials, document encoding or actual application errors (for example, a wrong value for a given field).

A.1.1 Roles

The RFC protocol is a dynamic protocol depending on the mode of the request. When a request is inbound to a RFC connected node, the function of that node is a server to fill the request. When a request is outbound from a node, the function of that node is a client, waiting for a server to respond to the request. Outbound events, which need no response, are also processed by servers.

A.1.2 Client

The system instantiating a connection to another system is a client, and the client role processes request/response messages in a synchronous manner. A client system does not execute, but enables message processing. Messages begin and end on the client. The adapter functions as a client when it sends messages from Oracle to SAP.

Types of client connections include:

- Application Server, which connects to a single SAP application server.
- Message Server, which connects to an SAP logon load balancer message server for connection to a group of SAP application servers for distributed load processing.

Both connection types use a special type of connection called a Connection Pool, which is discussed later in this appendix.

Types of client message types include:

- BAPI messages, which are object-style messages with parameters as XML attributes.
- RFC messages, which are RFC-style messages with parameters as XML elements.
- Intermediate Documents (IDocs), which use the ALE message system to send positional delimited flat text or XML element parameters.

A.1.3 Server

The system receiving a connection request from another system is a server. A server executes message processing and returns results to a client. A server can reject a message request if it does not implement the processing for a given message with an exception for NOT_IMPLEMENTED. A server can process two types of messages:

- Inbound messages, which contain the message request and the parameters for the request.
- Synchronous messages, which take the inbound message request and either perform additional work, pass it to another system, or process it for additional processing before returning it to SAP.

The SAP application server waits for a synchronous message process to finish before continuing.

Types of server connections include:

- Processing Mode (Request), a simple receipt of message type and parameters.
- Response, which executes a synchronous process.

Types of server message types include:

- RFC messages, which are RPC-style messages with parameters as elements.
- Intermediate Documents (IDocs), which use the ALE message system to send positional delimited flat text or XML element parameters.

A.2 SAP Inbound Message Processing

This section contains the following topics:

- [Section A.2.1, "Configuring SAP R/3 Inbound Processing"](#)
- [Section A.2.2, "Configuring a Logical System"](#)
- [Section A.2.3, "Configuring a Distribution Model for a Logical System"](#)
- [Section A.2.4, "Defining a Partner Profile"](#)

Sending request / response (client) messages to an SAP system for Remote Function Call (RFC) or Business Application Interface (BAPI) objects requires a valid set of logon credentials. For more information on how to configure adapter targets, including the list of required parameters, see ["Identifying SAP R/3 Logon Parameters"](#) on page 2-3. For the valid values for the adapter target, see your SAP system administrator.

Obtain the following values before initial logon:

- Valid SAP connection parameters.
- SAP Communications user (a SAP GUI DIALOG user is not recommended for security reasons).
- The correct SAP authorizations for the tasks the adapter is expected to perform.

SAP Authorizations

The proper SAP authorizations must be obtained before connecting to the server. It is a recommended SAP security practice to not allow RFC and DIAG access to the same user ID.

The basic authorization object used to secure RFC access is:

S_RFC: Secures the Function Group (located in authorization object class AAAB).

- *Authorization Object+S_RFC:*
 - *Authorization Field:*RFC_TYPE
 - *Authorization Value:* 'FUGR' is the only valid value.
 - *Authorization Field:* RFC_NAME
 - *Authorization Value:* The name of an activated function group. For example: ("ABCD") or '*'.
 - *Authorization Field:* ACTVT
 - *Authorization Value:* 16 (Execute) is the only valid value.

Individual function modules may have additional security inside their executable code. It is the requirement of the function developer or application creator to inform the users of the function or application of the authorization requirements to use the function or application. Consult the administrator to determine if there are additional authorizations required.

Individual tables may be secured from access by using the following authorization object:

S_TABU_DIS: Secures tables (located in authorization object class BC_A)

- *Authorization Object+S_TABU_DIS:*
 - *Authorization Field:* DICBERCLS: Or ("Authorization Group").

- *Authorization Value*: For example, Table: MARA is in group "MA".
- *Authorization Field*: ACTVT
- *Authorization Value*: 03 (Display)

For a complete explanation on using this authorization object, see the SAP system documentation.

Once the correct credentials are specified for the adapter target and the correct authorizations are assembled on the SAP application system, the adapter is ready to be initialized.

Request / Response (Client) Messages

Business Object Repository

Use Application Explorer to expand the appropriate application tree branch, such as *Financial Accounting*, and then navigate to and expand a Business Object, such as *Company Code*. Click the object to view the methods of that object. BAPIs have release and versioning. All BAPIs used at runtime must be in "release" status or a runtime exception is generated. Select a method to generate the XML schema. Create an XML instance document from the XML schema and populate it with data. Create an Oracle process with an appropriate input and output (file/html/etc) transmission method. Submit the document to the adapter for processing to receive a result.

Remote Function Call

Use Application Explorer to expand the appropriate application tree branch such as *Financial Accounting* and then navigate to and expand a Function Group, such as *0002 - Company Code*. Each function exposes a different method of processing. All BAPIs are present as RFC objects because their implementation is via RFC. They can be called in this manner allowing for event processing, for example, when BAPI processing does not. Not all RFCs are present as BAPIs. Select a function and generate the XML schema. Create an XML instance document from the XML schema and populate it with data. Create an Oracle process with an appropriate input and output (file/html/etc) transmission method. Submit the document to the adapter for processing to receive a result.

ALE IDOC Message

SAP ERP ALE* Inbound Message processing, where IDocs are sent and stored in the SAP R/3 system, requires some configuration on the SAP server to be used. In this role, messages are sent to SAP as transactional data and stored on the SAP server. The document data is processed in a second step inside SAP, during a triggered workflow.

A.2.1 Configuring SAP R/3 Inbound Processing

This section contains the following topic:

- [Section A.2.1.1, "Adapter Checklist"](#)

Sending an IDoc to SAP ERP requires the following information about the message parties:

- Sender of the message
- Receiver of the message
- Type of message that is being sent

The Sender and Receiver must be defined in the ALE system, and for each party, what kind of message each party is accepting and how it is being processed.

The Sending and Receiving system information are defined in units called Logical Systems, because the connection information does not involve physical connection information. A Logical System acts as a container for all the information about a particular Sender or Receiver.

Once a Logical System is defined, information about the different messages it sends and receives is stored in a Distribution Model. This model must be defined before any processing can occur, and it must be defined precisely to avoid Sender or Receiver errors. In the Distribution Model, the IDoc Message Application may also enable filters, which allow only specific parts of the IDoc to be sent, or distributed. There is a one-to-one correspondence between Logical System, Distribution Model, and Message Type. Complex filter rules can be configured by filter groups, filter objects, rules, and dependencies. For a complete explanation on configuring distribution models with filters, see the SAP system documentation.

Processing an IDoc by the SAP system is defined in Partner Profiles. For each partner and message, it is possible to define process codes to (possibly) process messages differently.

A.2.1.1 Adapter Checklist

For sending messages to SAP, an ALE administrator must assign (after logon credentials):

1. Logical System name.
2. Message Types and IDoc Basic Types to be transmitted to the SAP application server.

Inside the SAP ERP system, the ALE Administrator must:

1. Create the Logical System.
2. Create the Distribution Model.
 - a. Assign the Sender for the model.
 - b. Assign the Receiver for the model.
 - c. Assign the Message Types.
3. Create a Partner Profile for the Logical System.
 - a. For each message type, assign a process code for SAP to process the message.

Example

- Logical System "ORACLIENT" is created (via SAP transaction SALE).
- Distribution Model "ZORACLNT" is created (via SAP transaction BD64).
- Message Type "MATMAS" is added for sending partner "ORACLIENT" and receiving partner "T90ALECNT" (ALE component of central host).
- Partner Profile "ORACLIENT" is created (via SAP transaction WE20) and Inbound Message "MATMAS" is assigned with Process code "MATM".

Note: The steps for configuring SAP ERP for inbound messages are described here as a reference. For more comprehensive information, see the SAP ERP documentation.

You must perform the following steps to configure SAP R/3 for inbound IDoc processing:

1. Configure a Logical System.
2. Configure a Distribution Model.
3. Define an inbound Partner Profile.

A.2.2 Configuring a Logical System

This section contains the following topics:

- [Section A.2.2.1, "SAP Transaction Shortcuts"](#)
- [Section A.2.2.2, "Defining a Logical System"](#)

A Logical System is the container object for configuration information about the Sender or Receiver party of a message. It is a Logical System because no connection information is stored, only information about the message that the end point is able to process and the method used to process the message(s).

A.2.2.1 SAP Transaction Shortcuts

- `/n` - Ends the current transaction and opens a new one immediately.
- `/o` - Keeps the current transaction context and opens a new transaction in a new window.

A.2.2.2 Defining a Logical System

A Logical System definition is created by creating an entry in the SAP Implementation Guide for Customizing (IMG).

ALE is the SAP ERP Application Link Enablement system, so use the `/nSALE` transaction to navigate to the IMG for ALE.

1. Navigate to the SALE transaction.

The Display IMG window displays the ALE configuration tree.

2. Perform the following steps:
 - a. Expand the **Basic Settings** node.
 - b. Expand **Logical Systems**.
 - c. Click **Define Logical System**.
 - d. Since SAP R/3 users may see a slightly different screen, expand the extra node, **Sending and Receiving Systems**.

The "Caution: The table is cross-client" message is displayed.

This message appears to inform you that changes to customizing data in Logical Systems applies to all users of the SAP ERP server. This is in contrast to most application data, which is specifically bound to a logon client. Do not make any changes to Logical System entries. Only add the new information. Use an SAP logon ID with the correct privileges to do so.

3. Click the green check mark to continue.

The Change View "Logical Systems" window is displayed.

4. Click the **New Entries** button.

The New Entries: Overview of Added Entries window is displayed.

5. Enter the Logical System (for example, ORACLETDS) in the Log.System column and provide a description in the Name column.

Note: The fields are highlighted in yellow to remind you that they are required fields.

6. Click the **Save** icon on the top menu bar.

The Prompt for Workbench request dialog is displayed.

Note: Depending on system usage, the last workbench request number appears in the dialog. Do not use it and proceed to the next step.

7. Click the **New Request** icon to create a new request.
8. Enter the description for your entry in the request system. The remaining fields are filled in automatically and do not need to be changed.

The Prompt for Workbench Request dialog is displayed again, with the new request number filled in with the last request number created in the Workbench; do not use it.

9. Click the check mark to create a new request number.

The Logical System Entries window refreshes with the new Logical System information. The new Logical System is highlighted in blue to indicate the successful completion of this step.

A.2.3 Configuring a Distribution Model for a Logical System

This section contains the following topic:

- [Section A.2.3.1, "Defining a Distribution Model"](#)

A Distribution Model contains a Sender Logical System and a Receiver Logical System the type of Messages sent and received. If a given Sender or Receiver does not have a valid Distribution Model, then the message is not processed and routed to an error location. For advanced topics in Distribution Model, including SAP BAPI Objects and filtering, see the SAP documentation.

A.2.3.1 Defining a Distribution Model

To define a distribution model:

1. Run the **bd64** transaction.

The Display Distribution Model window is displayed in the default Display mode, which shows the current model view entries.

2. Expand the tree to view the entries for each model view.

To add items, the transaction must be switched into "change mode".

3. Click **Distribution model** and select **Switch processing mode**.

The screen refreshes and the model view window is switched to Change Distribution Model.

4. From the available menu buttons, select **Create Model View**.

The Create Model View dialog is displayed.

5. Enter a Model View name in the Short text field and a name in the Technical name field, which also serves as a description.

A good convention is to use names starting with the letter Z, since SAP does not overwrite these names in case of system updates. This is only the name of the Model View.

6. Click the green check mark icon to enter the information.

The screen refreshes and returns to the main Change Distribution Model window. The Distribution Model configured is now added to the list.

The Model View is in temporary transaction storage (and the entry appears in lighter color) until you click the Save icon. If the system connection is lost or times out, then the entry is lost. As a best practice, save your changes after each dialog during the Distribution Model configuration process to avoid losing changes.

7. Place the cursor on the entry previously defined.

The entry is now highlighted.

8. In the middle menu bar, click the **Add Message Type** button.

The Add Message Type dialog is displayed.

9. Perform the following steps:

- a. In the Sender field, enter the name of the sending Logical System for the Oracle system (for example, ORACLETDS). For more information, see ["Configuring a Logical System"](#) on page A-6.

- b. In the Receiver field, enter the name of the Client ALE Logical System for the SAP system (ALE receiver), or select from the drop-down (for example, T90CLNT090).

- c. In the Message type field, enter the message type you want to use (for example, MATMAS).

You can click the drop-down icon to the right of each field to browse available values from a list.

- d. Click the green check mark icon to enter the information.

The main Change Distribution Model window refreshes.

10. Click **Save**.

11. Navigate the ALE Hierarchy tree to the Distribution Model recently created. The Sender and Receiver are depicted in a graphical relationship with Sender, Receiver, and Message.

For subsequent message types for the same systems, expand the Model View to the receiver name and place the cursor on the name. Then click the **Add Message Type** button. All the fields are populated automatically. You can then add the message type.

A.2.4 Defining a Partner Profile

A Partner Profile defines for each logical system, the messages processed, and how the message is processed. This allows for flexible configuration of processing details for partners. Each message type may have only one entry in a Partner Profile. To define a Partner Profile:

1. Run the **we20** transaction by typing **/nwe20** at the command bar, which is located at the upper-left of the screen.

The Partner profiles window is displayed.

2. Select **Partner Type LS** and click the white icon on the middle menu bar. You can also select **Create** from the top menu.

The right side of the screen changes to a blank Partner Profile form.

3. Enter the name of the Logical System (for example, ORACLETDS) in the partner number field and click the **Save** icon.

The screen changes with the new profile.

4. Scroll down until the embedded text "Inbound parmtrs" entry is displayed.
5. Click the green plus sign icon.

The Partner profiles: Inbound parameters dialog is displayed.

6. Enter **MATMAS** as the Message Type.
7. Use the Process code drop down to find and select **MATM** for MATMAS.
8. Select **Immediate** or **Background** as the Process Mode.

Note: Consult your SAP administrator for the proper settings.

9. Click the **Save** icon on the menu bar to save the Inbound Partner Profile entry for MATMAS.
10. Click the **Back** (left facing arrow) button on the menu bar to return to the previous window.

MATMAS is displayed in the entry list.

This completes the inbound configuration for Logical System/Partner Profile: OracleTDS/MATMAS. To add additional inbound IDoc Message Type entries, repeat the steps beginning with "[Configuring a Distribution Model for a Logical System](#)" on page A-7. To add another Logical System and Message Type, repeat the steps from "[Configuring a Logical System](#)" on page A-6.

You can now begin sending IDocs after configuring the appropriate Oracle process.

From the SAP IDoc monitor console, you can monitor the IDoc transactions in SAP transaction **WE02**. The processing state and status of IDocs are available, as well as the Sender, Receiver, and the data. For more information, see the *IDoc Display* topic on the online SAP Help page.

A.3 SAP Outbound Message Processing

The Oracle Application Adapter for SAP R/3 can process outbound from SAP messages with interface style Remote Function Module, BAPI ALE, or Transactional RFC messages with transaction IDs. The message types can be Remote Function Calls or IDocs. All processing through the SAP JCo is synchronous (request / response), but message processing can either be asynchronous (Delivery only) or synchronous (Event/Reply), where an orchestrated process performs work and returns information to the waiting SAP application server. RFC messages can be sent from SAP ABAP programs or the SAP workflow system. IDocs are sent through system transactions or standard programs. BAPI events in SAP are for internal use unless specifically programmed and enabled for the ALE interface.

Events in SAP are enabled by event linkages to event handlers that do the work of processing the event. The mechanisms of sending the events are all enabled on the SAP system, the adapter is the recipient of the end process of the SAP event handler.

Outbound messages are defined in the adapter as events. Events are either asynchronous, delivering a message only, or synchronous, returning a reply. For an adapter inbound event to be two way (event-reply), the receiving process must implement a request handler in the process flow. The request handler takes the incoming SAP data, manipulates it, then returns the data in valid event reply format to the adapter to return to SAP.

For RFC outbound messages, only a SAP RFC Destination and Program ID are required to receive the Remote Function Call. For IDoc processing, a Logical System, Distribution Model, Partner Profile, and Port must be created to receive message events through RFC.

SAP Outbound Terminology

- Event Creator - JCO/RFC Client, which is the SAP system creating the event (request), and optionally waiting for a event reply (response).
- Event Listener - JCO/ RFC Server, which is the adapter receiving the event, handling the request, and optionally returning a response.

This section contains the following topics:

- [Section A.3.1, "SAP Gateway"](#)
- [Section A.3.2, "Program IDs and Load Balancing"](#)
- [Section A.3.3, "Registering a RFC Destination and Program ID in SAP GUI"](#)
- [Section A.3.4, "Application Link Embedding Configuration for the Event Adapter"](#)
- [Section A.3.5, "Defining a Port"](#)
- [Section A.3.6, "Configuring a Logical System"](#)
- [Section A.3.7, "Configuring a Distribution Model for a Logical System"](#)
- [Section A.3.8, "Defining a Partner Profile"](#)
- [Section A.3.9, "Testing the SAP R/3 ALE Configuration"](#)

Outbound processing often is implemented as Events. On the SAP application server, there are many stages of defined events. Event processing is held together by linkages. Not all linkages are enabled by default on a Productive SAP server. For an Event to be active, and capable of sending messages, there must be an object capable of sending changes in the status of the object, a link to list of valid events, and event handlers. An event handler implements the processing of the message.

Active SAP Events are status changes on defined components with implemented event handlers. There are many types of events in the SAP application server, the most well known event handler is ALE Message Control, implemented by SAP around objects like the Material Master and Customer Master. Changes in these objects can generate outbound event IDocs.

Event creation is usually implemented by SAP, but many objects can be enabled for event handling, but this often requires advanced configuration or programming. Consult the SAP documentation for information on the type of SAP events and how they are configured.

For RFC outbound messages, only a SAP RFC Destination and Program ID are needed to receive the Remote Function Call.

For IDoc processing, a Logical System, Distribution Model, Partner Profile and Port must be created to receive message control events via tRFC.

SAP Outbound Terminology

- JCO/RFC Client, which is the system making the RFC request, and optionally waiting for a response.
- JCO/ RFC Server, which is the system receiving the RFC request, handling the request, and optionally receiving a response.

A.3.1 SAP Gateway

The SAP Gateway is a process that runs on an SAP application server or a dedicated gateway server. The gateway process manages all connections to SAP. No server connections to SAP are accepted unless they are configured previously as registered programs in the SAP application and present the required program identifier.

A server connection presents itself to the Gateway and exposes a Program Identifier. If the Program Identifier is found in the list of registered Program IDs, then the Gateway server offers a connection to the server, which "accepts" a connection. This ProgramID is then linked with an RFC Destination within SAP, which enables SAP Function Modules and ALE documents (IDocs or BAPI IDocs) to be routed to the destination. The RFC Destination functions as a key to identify the SAP Program ID. The Program ID is also exposed by the application server making the connection to SAP and is the direct link to the IP address of the connecting server. Messages are routed to the application server via the RFC Destination. A good security practice is to separate Oracle Application Adapter for SAP R/3 users with access to the adapter instance implementing the Program id and the users with SAP transaction authority to change or view the Program ID inside SAP, all requests are routed through the RFC Destination.

Note: An RFC server program is registered with the SAP gateway and waits for incoming RFC call requests. An RFC server program registers itself under a Program ID at a SAP Gateway and not for a specific SAP system. In SAPGUI, the destination must be defined with transaction SM59, using connection type T and Register Mode. Moreover, this entry must contain information on the SAP gateway at which the RFC server program is registered.

A.3.2 Program IDs and Load Balancing

This section contains the following topics:

- [Section A.3.2.1, "SAP RFC Outbound Outline"](#)
- [Section A.3.2.2, "SAP IDoc Events"](#)

If the Gateway Server has a connection to a particular server instance and another server instance presents itself to the gateway, then the gateway offers the connection and then begins functioning in Load Balancing mode. Using a proprietary algorithm, the Gateway sends different messages to each server depending on demand and total processing time. This may cause unpredictable results when messages are validated by schema and application.

When configuring multiple events in the Oracle using a single SAP program ID, SAP load balances the event data. For example, if multiple remote function calls or BAPIs use the same program ID (for example, ORACLETDS) and multiple SAP listeners are configured with this programID, then SAP sends one request to one listener and the

next to another listener, and so on. There is a load-balancing algorithm present in the SAP Gateway Server. This mechanism is proprietary to SAP application development and might work by comparing total throughput of the connection, the number of times in wait state, and so on. One connection might receive nine messages and a second connection might receive one message. If five of the nine messages are rejected for schema validation and the one message on the other connection is rejected for schema validation, then you might suspect that you are missing SAP event handling messages.

Load balancing in server (inbound to adapter from SAP) situations is handled by connecting multiple instances of the adapter to the SAP system. The SAP system then load balances the connections. You cannot tune this performance.

A.3.2.1 SAP RFC Outbound Outline

1. Create a RFC Destination (via SM59).
 - a. Enter a Destination Name.
 - b. Enter a Registered Program ID.
 - c. In the MDMP & Unicode tab, ensure that the Unicode option is selected.
 - d. Start Oracle BPEL/Mediator.
 - e. Enter the same registered Program ID value from step b for the BPEL or Mediator channel.
 - f. Start the channel.
 - g. Test the connection using the Test button in SAP.
2. Simulate a SAP RFC message (via SE37).

ABAP Workbench authorization is required.

 - a. Enter the RFC Destination name from step 1a in the Destination field.
 - b. Enter a valid SAP RFC Function name in the Function field.
 - c. Enter data in the parameters for the function.
 - d. Click the Execute (clock) button.

Data arrives at the BPEL channel.

A.3.2.2 SAP IDoc Events

Before IDocs can be received from SAP, a channel for the Oracle Application Adapter for SAP R/3 must be configured using Application Explorer. In addition, configuration on the SAP application server is also required.

1. Create a RFC Destination (via SM59) or use the same destination defined in the previous section (SAP RFC Outbound Outline).
 - a. Enter a Destination Name.
 - b. Enter a Registered Program ID.
 - c. In the MDMP & Unicode tab, ensure that the Unicode option is selected.
 - d. Start Oracle BPEL/Mediator.
 - e. Enter the same Registered Program ID value from step b for the BPEL or Mediator channel.
 - f. Start the channel.
 - g. Test the connection using the Test button in SAP.

2. Create or use a Logical System as described in "[Configuring a Logical System](#)" on page A-6.
3. Define or use a Distribution Model as described in "[Configuring a Distribution Model for a Logical System](#)" on page A-7.
4. Create an SAP Transactional RFC Port via transaction WE21.
5. Create an SAP Outbound Partner Profile via transaction WE20.

A.3.3 Registering a RFC Destination and Program ID in SAP GUI

This section contains the following topics:

- [Section A.3.3.1, "Registering a Program ID"](#)
- [Section A.3.3.2, "Testing the Configuration"](#)
- [Section A.3.3.3, "Synchronous RFC Calls"](#)

To enable your SAP R/3 system to issue the following calls or interfaces to the SAP R/3 event adapter, you must register your program ID under an RFC destination.

- Remote Function Calls (RFC)
- Business Application Programming Interfaces (BAPI) - Bapis have no external events, use the RFC form
- Intermediate Documents (IDocs)

The RFC Destination is a symbolic name (for example, ORACLETDS) that is used to direct events to a target system, masking the Program ID. The Program ID is configured in SAP GUI and the event adapter.

A.3.3.1 Registering a Program ID

To register a Program ID:

1. Launch the SAP GUI and log in to the SAP R/3 system.
2. Select **Tools, Administration, Network**, and then **RFC destination**.
3. Run the **SM59** transaction.

The Configuration of RFC Connections window is displayed.

4. Select **TCP/IP connections** and click **Create** (white paper icon).

The RFC Destination window is displayed.

5. Provide the following information:

- a. In the RFC destination field, enter a name, for example, ORACLETDS.
This field value is case-sensitive. Match the same case in the sending programs.

- b. In the Connection type field, enter T for destination type TCP/IP.

- c. In the Description field, enter a brief description.

6. Click the Save icon (Disk) from the tool bar or select **Connection, Save** from the top menu bar.

The window refreshes and the Technical Settings tab is highlighted in the second panel.

7. Immediately click the **Registered Server Program** button.

The screen refreshes again.

8. Enter the case sensitive Program Id, this must match the same name in the adapter channel
9. Click the Save icon (Disk) from the tool bar or select Connection, Save from the top menu bar.
10. Select the **MDMP & Unicode** tab and click the **Unicode** button.
11. Click the Save icon (Disk) from the tool bar or select Connection, Save from the top menu bar.

The RFC Destination and Program ID are now defined. Any Remote Function Call (RFC) can now be sent to the server via the Destination parameter of the Call Function API. The data for the function must be marshaled and sent to the adapter as this is a reverse invocation.

A.3.3.2 Testing the Configuration

To test the configuration:

1. Enter the Program ID in the BPEL process / channel for Oracle Application Adapter for SAP R/3 and start the channel.
2. Enter SAP GUI and navigate to SE37.
ABAP Workbench authorization is required.
3. Enter a valid function name and click the Test Run (tool) icon.
The Test Function Module: Initial Screen window is displayed.
4. Enter the name of the RFC Destination, any data, and then click the Execute (clock) icon.

After a successful test, the execution time is displayed. The first run is always the longest, subsequent runs are faster, because the adapter must first obtain the metadata for the function and cache the metadata.

A.3.3.3 Synchronous RFC Calls

Synchronous RFC calls are defined in the same way as asynchronous on the SAP side. On the adapter side, ensure "request_response" is selected as the mode for the call. Orchestrate a process that takes the SAP delivered data, processes it, and returns it to SAP in the function format. Any format errors immediately terminate the request. Example (in pseudo code):

Function: MY_COMPANY_GETLIST DESTINATION 'ORACLETDS'

```
tables mycompanies
    field1: name(20)
    field2: company(20)
```

The synchronous Get List call passes the empty table *mycompanies* to BPEL server and SAP waits. BPEL orchestration includes an object that has the following SQL statement:

```
select name(20), company(20) into mycompanies where country eq us
```

The SQL statement is executed on an Oracle database and the result set is returned and passed into the *mycompanies* table by an assignment statement in the orchestration. Remember that the SAP application is waiting during the entire process, and that the

SAP server may have a maximum online process time at which the process is automatically terminated by SAP.

The function returns to SAP via synchronous call and the table is returned with data to the calling process.

Time outs can occur if the execution time is longer than the connection pool execution time. Format exceptions can occur for type, length, or decimal places.

A.3.4 Application Link Embedding Configuration for the Event Adapter

The SAP R/3 event adapter receives IDocs (Intermediate Documents) from SAP R/3. To configure an SAP R/3 system to send IDocs to the SAP R/3 event adapter, use the ALE (Application Link Embedding) configuration to:

1. Register your program ID in SAP GUI.
2. Define a Port.
3. Create a Logical System.
4. Create a Partner Profile.
5. Create a Distribution Model for the Partner and Message Type.
6. Test the SAP R/3 event adapter.

A.3.5 Defining a Port

A Port identifies where to send messages. The Port type used by Oracle Application Adapter for SAP R/3 is the *Transactional RFC port*. By defining an ALE port and linking it to the RFC Destination, this creates the message execution process pathway.

To define a port:

1. In the ALE configuration, select **Tools, Business Communications, IDocs Basis, IDoc**, and then **Port Definition**.

You can also run the **WE21** transaction.

The Ports in IDoc processing window is displayed.

2. In the left pane under Ports, select **Transactional RFC** and click **Create** (white paper icon).
3. Click **Generate port name**.
The system generates the port name.
4. Enter the version of the IDocs to send through this port (usually 4).
5. Type or select the name of the RFC Destination created (for example, ORACLETDS) or enter the first letters and an asterisk for the select list.
6. Save the session, making note of the generated RFC port.

A.3.6 Configuring a Logical System

Note: You can skip this section if you are reusing a Logical System that has been previously defined.

A Logical System is the container object for configuration information about the Sender or Receiver party of a message. It is a Logical System because no connection information is stored, only information about the message that the end point is able to process and the method used to process the message(s).

This section contains the following topics:

- [Section A.3.6.1, "SAP Transaction Shortcuts"](#)
- [Section A.3.6.2, "Defining a Logical System"](#)

A.3.6.1 SAP Transaction Shortcuts

- **/n** - Ends the current transaction and opens a new one immediately.
- **/o** - Keeps the current transaction context and opens a new transaction in a new window.

A.3.6.2 Defining a Logical System

A Logical System definition is created by creating an entry in the SAP Implementation Guide for Customizing (IMG).

ALE is the SAP ERP Application Link Enablement system, so use the **/nSALE** transaction to navigate to the IMG for ALE.

1. Navigate to the SALE transaction.

The Display IMG window displays the ALE configuration tree.

2. Perform the following steps:

- a. Expand the **Basic Settings** node.
- b. Expand **Logical Systems**.
- c. Click **Define Logical System**.
- d. Since SAP R/3 users may see a slightly different screen, expand the extra node, **Sending and Receiving Systems**.

The "Caution: The table is cross-client" message is displayed.

This message appears to inform you that changes to customizing data in Logical Systems applies to all users of the SAP ERP server. This is in contrast to most application data, which is specifically bound to a logon client. Do not make any changes to Logical System entries. Only add the new information. Use an SAP logon ID with the correct privileges to do so.

3. Click the green check mark to continue.

The Change View "Logical Systems" window is displayed without highlighting.

4. Click the **New Entries** button.

The New Entries: Overview of Added Entries window is displayed.

5. Enter the Logical System (for example, ORACLETD5) in the Log.System column and provide a description in the Name column.

Note: The fields are highlighted in yellow to remind you that they are required fields.

6. Click the **Save** icon (floppy disk icon) on the top menu bar.

The Prompt for Workbench request dialog is displayed.

Note: Depending on system usage, the last workbench request number appears in the dialog. Do not use it and proceed to the next step.

7. Click the **New Request** icon to create a new request.
8. Enter the description for your entry in the request system. The remaining fields are filled in automatically and do not need to be changed.

The Prompt for Workbench Request dialog is displayed again, with the new request number filled in with the last request number created in the Workbench; do not use it.

9. Click the check mark to create a new request number.

The new Logical System is highlighted in blue to indicate the successful completion of this step.

A.3.7 Configuring a Distribution Model for a Logical System

A Distribution Model contains a Sender Logical System and a Receiver Logical System the type of Messages sent and received. If a given Sender or Receiver does not have a valid Distribution Model, then the message is not processed and routed to an error location. For advanced topics in Distribution Model, including SAP BAPI Objects and filtering, see the SAP documentation.

This section contains the following topic:

- [Section A.3.7.1, "Defining a Distribution Model"](#)

A.3.7.1 Defining a Distribution Model

To define a distribution model:

1. Run the **bd64** transaction by typing **/nbd64** at the command bar, which is located at the upper-left of the screen.

The Display Distribution Model window is displayed in the default Display mode, which shows the current model view entries.

2. Expand the tree to view the entries for each model view.

To add items, the transaction must be switched into "change mode".

3. From the top menu, click **Distribution model** and select **Switch processing mode** from the list.

The screen refreshes and returns to the main Change Distribution Model window.

4. From the available menu buttons, select **Create Model View**.

The Create Model View dialog is displayed.

5. Enter a Model View name in the Short text field and a name in the Technical name field, which also serves as a description.

A good convention is to use names starting with the letter Z, since SAP does not overwrite these names in case of system updates. This is only the name of the Model View.

6. Click the green check mark icon to enter the information.

The screen refreshes and returns to the main Change Distribution Model window. The Distribution Model configured is now added to the list.

The Model View is in temporary transaction storage (and the entry appears in lighter color) until you click the Save icon. If the system connection is lost or times out, then the entry is lost. As a best practice, save your changes after each dialog during the Distribution Model configuration process to avoid losing changes.

7. Place the cursor on the entry previously defined.
The entry is now highlighted.
8. In the middle menu bar, click the **Add Message Type** button.
The Add Message Type dialog is displayed.
9. Enter the ALE Central Instance Sender.
Consult your SAP administrator for this information.
10. Enter the Receiver (for example, ORACLETDS) in the Receiver field.
11. Enter a valid Message Type.
12. Repeat this procedure for each outbound Message Type in this partner scenario.

A.3.8 Defining a Partner Profile

Note: You can skip this section if you are reusing a Partner Profile that has been previously defined.

A Partner Profile defines for each logical system, the messages processed, and how the message is processed. This allows for flexible configuration of processing details for partners. Each message type may have only one entry in a Partner Profile. To define a Partner Profile:

1. Run the **we20** transaction by typing **/nwe20** at the command bar, which is located at the upper-left of the screen.
The Partner profiles window is displayed.
2. Select **Partner Type LS** and click the white icon on the middle menu bar. You can also select **Create** from the top menu.
The right side of the screen changes to a blank Partner Profile form.
3. Enter the name of the Logical System (for example, ORACLETDS) in the partner number field and click the **Save** icon.
The screen changes with the new profile.
4. Enter the Message Type in the Message Type field.
5. Enter the Port Number in the Receiver Port field
6. Select the **Transfer IDoc immed.** button or select **Collect IDoc**, specify a batch size, and use bd87 to release IDocs for processing.
7. Enter the Basic Type (MATMAS05 for MATMAS, and so on).
8. Click the **Save** icon.

A.3.9 Testing the SAP R/3 ALE Configuration

In the SAP Server, the BD12 transaction enables the sending of Master Data IDocs to any logical system, for example, to an event adapter.

This section contains the following topics:

- [Section A.3.9.1, "Testing the SAP R/3 ALE Configuration"](#)
- [Section A.3.9.2, "SAP IDoc Events"](#)
- [Section A.3.9.3, "Working With IDoc Versions"](#)

A.3.9.1 Testing the SAP R/3 ALE Configuration

To test the SAP R/3 Application Link Embedding (ALE) configuration:

1. In the Send Customers window, enter the IDoc message type (for example, DEBMAS) in the Output type field.
2. In the Logical system field, enter the logical system, for example, ORACLETDS
3. Click **Run**.

The SAP R/3 event adapter receives the IDoc in XML format. No response is expected from the event adapter.

A confirmation message is displayed.

The same number of communication IDocs should also be generated. If a different number appears for Communication IDoc count versus the Master IDoc count, some IDocs may have failed the syntax check. Check with the IDoc application developer. Usually, the IDocs should appear at the adapter server almost immediately. If no IDocs appear, then check the following:

1. The Collect IDocs option is selected in the Partner Profile (go to bd87 to release them for processing).
2. No other server is exposing the same Program ID.

A.3.9.2 SAP IDoc Events

Before IDocs can be received from SAP, a channel for the Oracle Application Adapter for SAP R/3 must be configured using Application Explorer. In addition, configuration on the SAP application server is also required.

1. Create a RFC Destination (via SM59) or use the same destination defined in the previous section (SAP RFC Outbound Outline).
 - a. Enter a Destination Name.
 - b. Enter a Registered Program ID.
 - c. In the MDMP & Unicode tab, ensure that the Unicode option is selected.
 - d. Start Oracle BPEL/Mediator.
 - e. Enter the same Registered Program ID value from step b for the BPEL or Mediator channel.
 - f. Start the channel.
 - g. Test the connection using the Test button in SAP.
2. Create or use a Logical System as described in "[Configuring a Logical System](#)" on page A-6.

3. Define or use a Distribution Model as described in "[Configuring a Distribution Model for a Logical System](#)" on page A-7.
4. Create an SAP Transactional RFC Port via transaction WE21.
5. Create an SAP Outbound Partner Profile via transaction WE20.

A.3.9.3 Working With IDoc Versions

IDoc Outbound Segments in an SAP outbound IDoc must match the metadata appropriate for the release.

The adapter has three settings for IDoc release version:

- SAP logon release - The current version.
- IDoc release - The sending application would have to modify the IDoc control record to output the correct version, if not, it defaults to the SAP logon release.
- User input

The user input field can have two types of input. If the input is an SAP release, such as 700, then the metadata from that release will be used.

A blank value in this field causes SAP to return the highest version metadata it has, released or not. This is often the best option when working with unknown or recent IDocs. If the Segment name cannot be found in the released Segments list, SAP will return (through the adapter) the message "Segment unknown". The only option to process the IDoc in this case is to either release the segment or change the adapter setting to "user input" with a blank value.

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, 5-2
adapter exceptions, 12-4 to 12-6
adapter types
 resource, 1-2, 5-1 to 5-2
adapters
 configuring, 4-1 to 4-28, 5-2
 deploying, 1-5, 5-1 to 5-2, 6-2, 8-2
 integrating with BPEL Process Manager, 6-1, 8-1
 troubleshooting, 12-1 to 12-8
Adapters node, 6-42
Add Channel dialog box, 4-22
Add Target dialog box, 4-9
Advanced tab, 4-11, 4-25, 6-40
application adapters
 configuring, 4-1 to 4-28, 5-2
 deploying, 5-1 to 5-2, 6-2, 8-2
 integrating with BPEL Process Manager, 6-1, 8-1
 troubleshooting, 12-1 to 12-8
Application Explorer, 1-2, 6-7, 6-37
 application systems and, 4-8
 channels and, 4-22
 debugging and, 12-3
 J2CA configuration and, 5-2
 OracleAS Adapter J2CA and, 12-5
 schemas and, 4-17
 testing and, 12-3
 troubleshooting, 12-2 to 12-3
 WSDL files and, 6-37
Application Server dialog box, 4-9
Application Server parameter, 4-10
application systems
 Application Explorer and, 4-8
 supported, 4-8

B

BAPI (Business Application Programming Interfaces), 12-4
BAPI Exception parameter, 4-11, 4-25, 6-40
BAPI requests, 1-5
BPEL Designer, 6-1, 8-1
BPEL Process Manager
 integrating with adapters, 6-1, 8-1

 OracleAS Adapter for MySAP ERP and, 6-1, 8-1
BSE (OracleAS Adapter Business Services Engine), 1-2, 1-5
 troubleshooting, 12-5
BSE URL field, 4-6
Business Application Programming Interfaces (BAPI), 12-4
business events, 1-2
business functions, 4-8, 4-17 to 4-21
 creating schemas for, 4-17
business processes, 1-4
business services
 creating, 4-19 to 4-21
 testing, 4-21

C

CCI calls, 5-1
channels, 4-22
 Application Explorer and, 4-22
 creating, 4-22, 6-37 to 6-41
 deleting, 4-28
 editing, 4-28
 starting, 4-27, 6-41
 stopping, 4-27, 6-41
 testing and debugging and, 6-41
Channels node, 4-23, 6-38
channels. *See also* listeners
Client parameter, 4-9, 4-25, 6-40
Code page parameter, 4-9, 4-25, 6-40
Commit with wait parameter, 4-11, 4-25, 6-40
configuration parameters, 5-4
 IWayConfig, 5-4
 IWayHome, 5-4
 IWayRepoPassword, 5-4
 IWayRepoURL, 5-4
 IWayRepoUser, 5-4
 Loglevel, 5-4
configurations
 connecting to, 4-7
 defining, 4-5 to 4-7
 overwriting, 5-4
Configurations node, 4-5 to 4-6
configuring adapters, 4-1 to 4-28, 5-2
connecting to MySAP ERP, 4-8 to 4-15, 6-8, 6-42, 12-2, 12-7

- connecting to OracleAS Adapter J2CA, 12-5
- Connection dialog box, 4-15, 6-42
- connection parameters, 4-9 to 4-15, 4-24 to 4-25, 12-2
 - Advanced, 4-11, 4-25, 6-40
 - Application Server, 4-10
 - BAPI Exception Handling, 4-11
 - Client, 4-9, 4-25, 6-40
 - Code page, 4-9
 - Commit with wait, 4-11
 - Connection pool name, 4-11
 - Connection pool size, 4-11
 - EDI Version, 4-10
 - Language, 4-9
 - Logon ticket (SSO2), 4-13
 - Logon ticket (X509), 4-13
 - MySAP ERP Trace, 4-9
 - Password, 4-9, 4-25, 6-40
 - Security, 4-13
 - SNC level, 4-13
 - SNC library path, 4-14, 4-25, 6-40
 - SNC mode, 4-13
 - SNC name, 4-13
 - SNC partner, 4-13
 - System, 4-10, 4-24, 6-39
 - System Number, 4-10
 - User, 4-9, 4-25, 6-40
- Connection pool name parameter, 4-11, 4-25, 6-40
- Connection pool size parameter, 4-11, 4-25, 6-40
- connections
 - closing, 4-15
 - deleting, 4-15
 - establishing, 4-8 to 4-15, 6-42
- Connector
 - deploying to Oracle Application Server, 5-1
- connector factories, 5-2
- connector factory objects, 5-2
 - multiple, 5-4
- Create Outbound JCA Service (Request/Response), 6-8
- Create Web Service dialog box, 4-20
- creating channels, 6-37 to 6-41
- creating events, 6-42 to 6-44
- creating repository projects, 4-5 to 4-7
- customer master data, 6-54

D

- database connections
 - opening, 5-4
- databases
 - connecting to, 5-4
 - Oracle, 5-4
- DEBMAS05
 - locating, 6-42
- deploying adapters, 5-1 to 5-2, 6-2, 8-2
- Description field, 4-9, 4-20
- design time, 6-37
 - configuring, 6-7, 6-37
- disconnecting from MySAP ERP, 4-15

E

- Eclipse. *See* JDeveloper
- EDI Version parameter, 4-10
- editing channels, 4-28
- editing targets, 4-16
- EJB (Enterprise Java Beans), 5-1
- Enterprise Connector for J2EE Connector Architecture (J2CA), 1-5
- Enterprise Java Beans (EJB), 5-1
- error messages, 12-2 to 12-8
 - target systems and, 12-6
- event adapters
 - configuring, 4-22
- event handling, A-1
- event integration
 - verifying, 6-54
- event messages, 6-37
- event ports
 - creating, 6-42 to 6-44
- events, 1-2, 4-22, A-1
 - configuring, 4-22
 - creating, 6-42 to 6-44
 - triggering, 6-54
- Existing Service Names list, 4-20
- Export WSDL dialog box, 6-43

F

- fault code elements, 12-6
- fault string elements, 12-6

H

- Hostname parameter, 4-6

I

- IDoc (Intermediate Documents), 12-4
- IDoc requests, 1-5
- in, 4-25
- inbound J2CA services, 6-37
- input XML documents, 12-4
- installation directories, 5-4
- Intermediate Documents (IDoc), 12-4
- IWayConfig parameter, 5-4
- IWayHome parameter, 5-4
- IWayRepoPassword parameter, 5-4
- IWayRepoURL parameter, 5-4
- IWayRepoUser parameter, 5-4

J

- J2CA (Enterprise Connector for J2EE Connector Architecture), 1-5
 - OracleAS Adapter and, 5-1
- J2CA configuration
 - Application Explorer and, 5-2
- J2CA resource adapters, 1-2, 5-1
- J2CA services, 6-37
- Java program clients, 5-1

JDeveloper, 6-1, 8-1

L

Language parameter, 4-9, 4-25, 6-40
License and Method dialog box, 4-20
License field, 4-20
licenses, 4-20
list of nodes, 12-2
listeners, 1-5, 4-7
listeners. *See also* channels
locating DEBMS05, 6-42
log files, 12-1
log levels
 overwriting, 5-4
logical systems, 6-54
LogLevel parameter, 5-4
Logon ticket (SS02) parameter, 4-13
Logon ticket (X509) parameter, 4-13

M

managed connector factories, 5-2
managed connector factory objects, 5-2
 multiple, 5-4
ManagedConnectionFactory parameter, 5-4
Mediator Inbound Process, 7-13
Mediator Outbound Process, 7-2
Message Server dialog box, 4-22
message types
 event, 6-37
messages, 1-2
Method Name field, 4-20
MySAP Easy Access window, 6-54
MySAP ERP
 connecting to, 1-4, 4-8 to 4-15, 6-8, 6-42, 12-2, 12-7
 disconnecting from, 4-15
 integrating, 1-4
MySAP ERP business functions, 4-8, 4-17 to 4-21
 creating schemas for, 4-17
MySAP ERP business processes, 1-4
MySAP ERP platforms, 1-4
MySAP ERP Trace parameter, 4-9, 4-25, 6-40
MySAP node, 6-42
MySAP program ID (identifiers), 6-54
MySAP Workbench, 6-54

N

Name field, 4-9
New Configuration dialog box, 4-5 to 4-7
Node list, 12-2
nodes
 Adapters, 6-42
 Channels, 4-23, 6-38
 Configurations, 4-5 to 4-6
 connected, 4-15, 6-42
 disconnected, 4-16
 MySAP, 6-42
 Ports, 6-38

O

Object type, 12-4
OC4J-ra.xml file, 5-2 to 5-4
Oracle Application Server
 deployment of Connector to, 5-1
Oracle databases, 5-4
Oracle JDeveloper, 6-1, 8-1
Oracle JDeveloper BPEL Designer. *See* BPEL Designer,
 JDeveloper, or Oracle JDeveloper
Oracle's Unified Method (OUM), xi
OracleAS Adapter
 installation directory and, 5-4
 J2CA and, 5-1
OracleAS Adapter Application Explorer. *See*
 Application Explorer
OracleAS Adapter Business Services Engine
 (BSE), 1-2, 1-5
 troubleshooting, 12-5
OracleAS Adapter for MySAP ERP
 BPEL Process Manager and, 6-1, 8-1
 configuring, 4-1 to 4-28
 deploying, 1-2, 6-2, 8-2
 troubleshooting, 12-1 to 12-8
OracleAS Adapter J2CA
 Application Explorer and, 12-5
 connecting to, 12-5
outbound MySAP ERP processing, A-1 to ??

P

parameter types
 channel configuration, 4-28
 configuration, 5-4
 connection, 4-24 to 4-25, 12-2
Password parameter, 4-9, 4-25, 6-40
passwords, 4-9, 4-25, 5-4, 6-40
permissions
 configuring, 12-5
Port Number parameter, 4-6
ports
 creating, 6-42 to 6-44
Ports node, 6-38
Process Manager. *See* BPEL Process Manager
program ID (identifiers), 6-54
properties, 5-4

R

Remote Function module, 12-4
repository projects
 creating, 4-5 to 4-7
 Web services and, 4-5
request processing
 BAPI, 1-5
 IDoc, 1-5
 RFC, 1-5
request schemas, 4-17
requests
 executing, 12-4
resource adapters, 5-1 to 5-2

- deploying, 1-2
- response schemas, 4-17
- Restrictions page, 6-54
- RFC requests, 1-5

S

- sapjco3.dll file, 12-2
- sapjco3.jar file, 12-2 to 12-3
- schemas
 - Application Explorer and, 4-17
 - creating, 4-17
- Security tab, 4-13
- Send Customers window, 6-54
- service names, 4-20
- Service Provider list, 4-6 to 4-7
- Service-Oriented Architecture (SOA), 6-1, 8-1
- services, 1-2
 - creating, 4-19 to 4-21
 - testing, 4-21
- servlets, 5-1
- SNC level parameter, 4-13
- SNC library path parameter, 4-13
- SNC mode parameter, 4-13
- SNC name parameter, 4-13
- SNC partner parameter, 4-13
- SOA (Service-Oriented Architecture), 6-1, 8-1
- SOAP agents, 12-6
- SOAP faults, 12-6
- SOAP requests, 12-6 to 12-8
 - errors and, 12-6 to 12-7
- System Number parameter, 4-10
- System tab, 4-10, 4-24, 6-39

T

- target systems
 - errors and, 12-6
- Target Type list, 4-9
- targets
 - connecting to, 4-15, 6-42, 12-7
 - defining, 4-8
 - deleting, 4-15
 - disconnecting from, 4-15
 - editing, 4-16
- testing Web services, 4-21
- trace information, 12-1
- transaction processing, 1-2
- triggering events in MySAP ERP, 6-54
- troubleshooting, 12-1 to 12-8
 - Application Explorer, 12-2 to 12-3
 - BSE, 12-5
 - Web services, 12-5 to 12-7

U

- updating adapter configuration, 5-2
- User parameter, 4-9, 4-25, 6-40, 12-5
- User tab, 4-9, 4-25, 6-40

W

- Web Service Definition Language (WSDL), 4-18 to 4-19
- Web service names, 4-20
- Web service projects
 - creating, 4-5
- Web services, 1-2
 - creating, 4-19 to 4-21
 - deploying, 12-6 to 12-8
 - integrating, 6-1, 8-1
 - repository projects and, 4-5
 - testing, 4-21
 - troubleshooting, 12-5 to 12-7
- WSDL (Web Service Definition Language), 4-18 to 4-19
- WSDL documents, 6-1, 8-1
- WSDL files, 6-1, 8-1
 - Application Explorer and, 6-37

X

- XML documents
 - input, 12-4
- XML messages, 1-2
- XML schemas
 - creating, 4-17