

**Oracle® Fusion Middleware**

Application Adapter for SAP R/3 User's Guide for Oracle  
WebLogic Server

11g Release 1 (11.1.1)

**E17057-01**

April 2010

Oracle Fusion Middleware Application Adapter for SAP R/3 User's Guide for Oracle WebLogic Server, 11g Release 1 (11.1.1)

E17057-01

Copyright © 2009, 2010, Oracle and/or its affiliates. All rights reserved.

Primary Author: Stefan Kostial

Contributing Authors: Sunil Gopal, Marian Jones, Vikas Anand, Sunil Wadhwa, Vishal Saxena, Vimmika Dinesh

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 software or related documentation 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 USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software 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 which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

This software 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

<b>Preface</b> .....	vii
Audience .....	vii
Documentation Accessibility .....	vii
Related Documents .....	viii
Conventions .....	viii
<b>1 Introduction</b>	
<b>Adapter Features</b> .....	1-1
Supported Versions and Platforms .....	1-2
<b>Classical SAP Technologies for ABAP</b> .....	1-2
<b>Integration with SAP R/3</b> .....	1-3
<b>Adapter Architecture</b> .....	1-4
<b>BSE Versus Oracle Adapter J2CA Deployment</b> .....	1-6
<b>2 Getting Started</b>	
<b>Verifying the SAP Java Connector (SAP JCo)</b> .....	2-1
Verifying SAP JCo on Windows Platforms .....	2-1
Verifying SAP JCo on UNIX Platforms .....	2-2
<b>Identifying SAP R/3 Logon Parameters</b> .....	2-3
User Parameters .....	2-3
System Settings (Application Server) Parameters .....	2-3
System Settings (Message Server) Parameters .....	2-4
Connection Pool Parameters .....	2-4
SAP Gateway Parameters .....	2-5
ALE Parameters .....	2-5
Global Processing Parameters .....	2-6
SNC Parameters .....	2-7
<b>3 Configuring Oracle Application Adapter for SAP R/3</b>	
<b>Starting Application Explorer</b> .....	3-1
<b>Configuring Repository Settings</b> .....	3-2
Configuring the Database Repository for BSE .....	3-2
Configuring the Database Repository for J2CA .....	3-6
<b>Creating a Repository Configuration</b> .....	3-8
Creating a Configuration for BSE .....	3-8

Creating a Configuration for J2CA .....	3-9
Connecting to a BSE or J2CA Configuration.....	3-11
<b>Establishing a Connection (Target) for SAP R/3 .....</b>	<b>3-11</b>
Defining a Target to SAP R/3 .....	3-11
Connecting to a Defined SAP R/3 Target .....	3-18
Managing a Connection to SAP R/3.....	3-18
<b>Viewing Application System Objects .....</b>	<b>3-19</b>
<b>Creating XML Schemas .....</b>	<b>3-20</b>
<b>Generating WSDL (J2CA Configurations Only).....</b>	<b>3-20</b>
<b>Creating and Testing a Web Service (BSE Configurations Only) .....</b>	<b>3-21</b>
<b>Configuring an Event Adapter .....</b>	<b>3-24</b>
Creating and Editing a Channel.....	3-24
Schema Validation .....	3-28

## 4 Oracle WebLogic Server Deployment and Integration

Adapter Integration with Oracle WebLogic Server .....	4-1
Deployment of Adapter .....	4-1
Updating Adapter Configuration.....	4-2

## 5 Integration With BPEL Service Components in the Oracle SOA Suite

Overview .....	5-1
Deployment of Adapter .....	5-1
Configuring a New Application Server Connection .....	5-2
<b>Designing an Outbound BPEL Process for Service Integration.....</b>	<b>5-6</b>
Generating WSDL for Request/Response Service .....	5-7
Creating an Empty Composite for SOA .....	5-8
Defining a BPEL Outbound Process.....	5-10
Deploying the BPEL Outbound Process .....	5-28
Invoking the Input XML Document in the Oracle Enterprise Manager Console.....	5-32
Testing Outbound BPEL and Mediator Processes .....	5-34
<b>Designing an Inbound BPEL Process for Event Integration.....</b>	<b>5-35</b>
Generating WSDL for Event Integration .....	5-35
Creating an Empty Composite for SOA .....	5-41
Defining a BPEL Inbound Process.....	5-43
Deploying the BPEL Inbound Process .....	5-51
Triggering an Event in SAP R/3.....	5-55

## 6 Integration With Mediator Service Components in the Oracle SOA Suite

Configuring a New Application Server Connection .....	6-2
<b>Configuring a Mediator Outbound Process.....</b>	<b>6-6</b>
Creating an Empty Composite for SOA .....	6-6
Defining a Mediator Outbound Process .....	6-8
Deploying the Mediator Outbound Process .....	6-22
Invoking the Input XML Document in the Oracle Enterprise Manager Console.....	6-26
<b>Configuring a Mediator Inbound Process.....</b>	<b>6-27</b>
Creating an Empty Composite for SOA .....	6-28

Defining a Mediator Inbound Process .....	6-30
<b>7 Troubleshooting and Error Messages</b>	
<b>Troubleshooting</b> .....	7-1
<b>BSE Error Messages</b> .....	7-5
General Error Handling in BSE.....	7-5
Adapter-Specific Error Handling.....	7-6
<b>8 Advanced User Tools</b>	
<b>Web Services Policy-Based Security</b> .....	8-1
Configuring Web Services Policy-Based Security .....	8-2
<b>Migrating Repositories</b> .....	8-8
<b>A Configuring SAP R/3 for Inbound and Outbound Processing</b>	
<b>Configuring SAP R/3 Inbound Processing</b> .....	A-1
Configuring a Logical System .....	A-1
Configuring a Distribution Model.....	A-4
Defining a Partner Profile .....	A-7
<b>Configuring SAP R/3 Outbound Processing</b> .....	A-9
Related Concepts and Terminology .....	A-9
Registering Your Program ID in SAPGUI.....	A-11
Testing the SAP R/3 Event Adapter .....	A-12
Application Link Embedding Configuration for the Event Adapter .....	A-14
Defining a Port.....	A-14
Creating a Logical System.....	A-14
Creating a Partner Profile .....	A-16
Collected IDocs.....	A-17
Creating a Distribution Model for the Partner and Message Type .....	A-17
Testing the SAP R/3 ALE Configuration .....	A-19

**Glossary**

**Index**



---

---

# Preface

*Oracle Fusion Middleware Application Adapter for SAP R/3 User's Guide for Oracle WebLogic Server* describes how to provide connectivity and integrate with SAP R/3.

## Audience

*Oracle Fusion Middleware Application Adapter for SAP R/3 User's Guide for Oracle WebLogic Server* is intended for those who integrate with SAP R/3 systems and develop applications.

## Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible to all users, including users that are disabled. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at <http://www.oracle.com/accessibility/>.

### Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

### Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

### Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/support/contact.html> or visit <http://www.oracle.com/accessibility/support.html> if you are hearing impaired.

## Related Documents

For more information, see the following documents in the Oracle Enterprise Repository 11g Release 1 (11.1.1) 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](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:

<b>Convention</b>	<b>Meaning</b>
<b>boldface</b>	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. This chapter discusses the following topics:

- [Adapter Features](#)
- [Classical SAP Technologies for ABAP](#)
- [Integration with SAP R/3](#)
- [Adapter Architecture](#)
- [BSE Versus Oracle Adapter J2CA Deployment](#)

## 1.1 Adapter Features

Oracle Application Adapter for SAP R/3 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. 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).

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 2.18.

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, contact your iWay 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

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.

### 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.

---

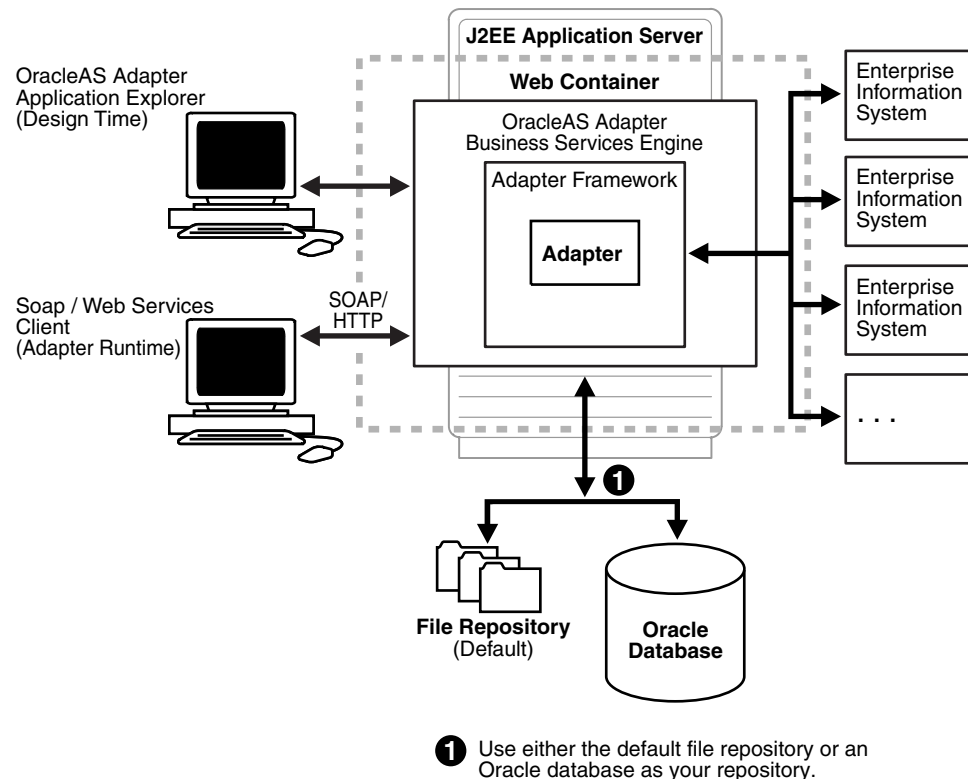
---

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

---

---

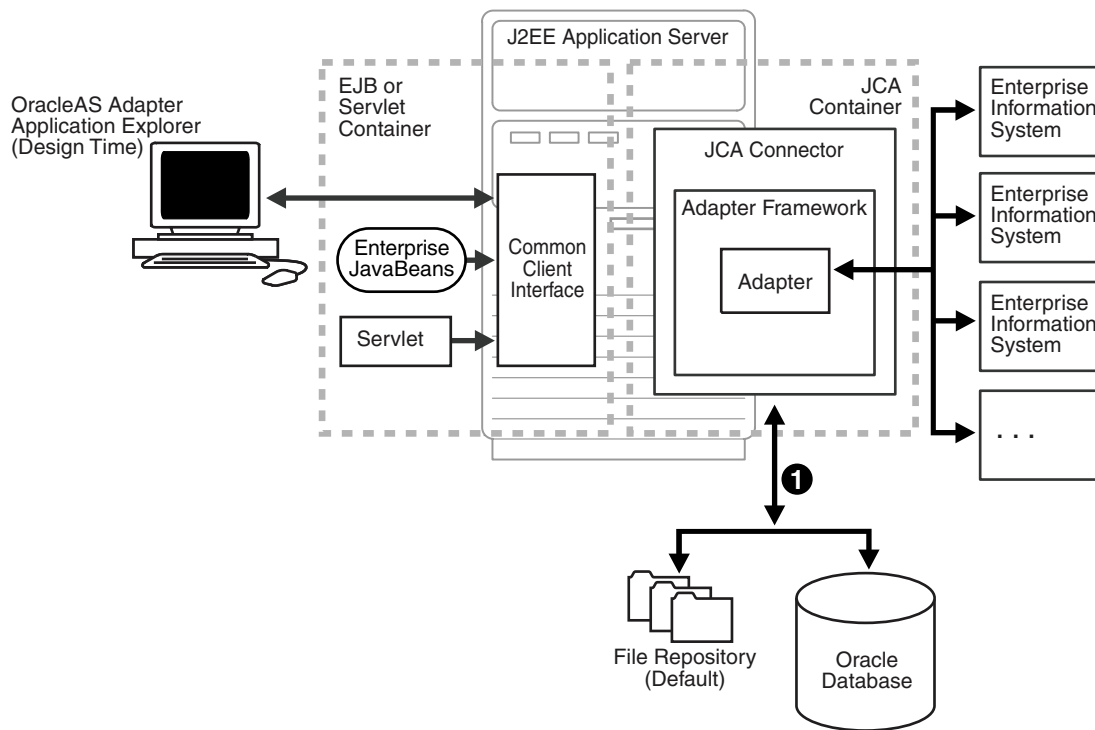
**Figure 1–1 Oracle Adapter Business Services (BSE) Architecture**



### 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`. See [Chapter 4, "Oracle WebLogic Server Deployment and Integration"](#) for more information.

**Figure 1–2 Oracle Adapter J2CA Architecture**

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

**See Also:**

- *Oracle Application Server Adapter Concepts Guide*
- *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 Business Process Execution Language (BPEL) Process Manager, note that:

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

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

1. BSE is the preferred deployment option because it:
  - Can be deployed in a separate instance of Oracle WebLogic Server.
  - Provides better distribution of load.
  - Provides better isolation from any errors from third party libraries.
  - Provides better capability to isolate issues for debugging purposes.

- Conforms more closely to the Service Oriented Architecture (SOA) model for building applications.
2. Oracle Adapter J2CA provides slightly better performance.

Oracle Adapter J2CA does provide slightly better performance than BSE. However, the difference decreases as the transaction rate increases.

3. Oracle Adapter J2CA and the BSE option both provide identity propagation at run-time.

The BSE option provides the capability to pass identity using the SOAP header. For Oracle Adapter J2CA, user name and password can be passed using the connection specification of the CCI.





---

---

## Getting Started

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

- [Verifying the SAP Java Connector \(SAP JCo\)](#)
- [Identifying SAP R/3 Logon Parameters](#)

### 2.1 Verifying the SAP Java Connector (SAP JCo)

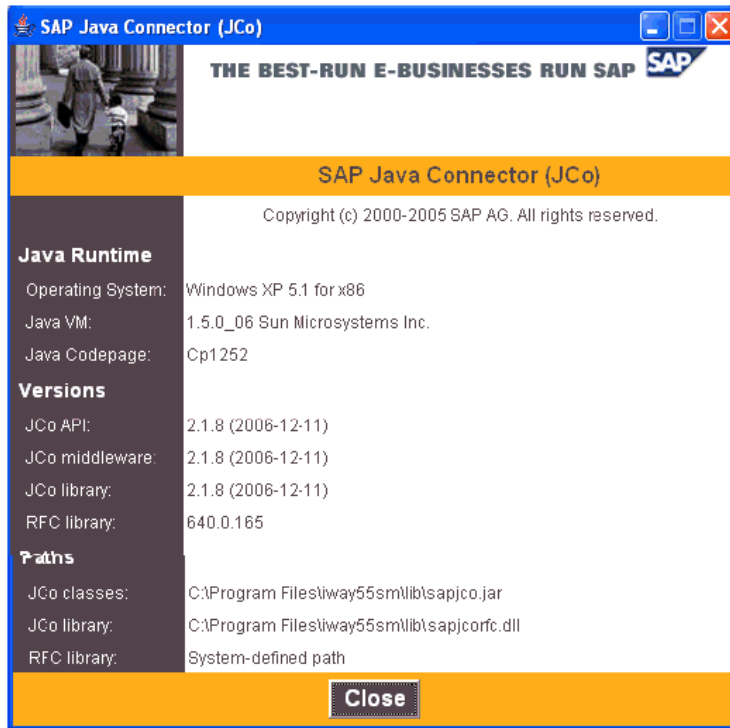
Once you have installed the SAP Java Connector (SAP JCo), as a best practice, you can verify the connector to make sure it is installed correctly and that all the required SAP JCo library files are available.

#### 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 sapjco.jar file is located.
2. Right-click the sapjco.jar file, select Open With from the context menu, and click Java 2 Platform Standard Edition binary.

The SAP Java Connector (JCo) dialog box is displayed.



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 sapjco.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) 2000-2005 SAP AG. All rights reserved. |
| Version Information |
-----

Java Runtime:
Operating System: SunOS 5.7 for sparc
Java VM: 1.4.0-beta3 Sun Microsystems Inc.
Java Codepage: ASCII
Versions:
JCo API: 2.1.8 (2006-12-11)
JCo middleware: 2.1.8 (2006-12-11)
JCo library: 2.1.8 (2006-12-11)
RFC library: 640.0.165
Paths:
JCo classes: /u4/fpgjpr/iWay55sm/lib/sapjco.jarJCo library:
/u4/fpgjpr/iWay55sm/lib/libsapjcorfc.so
RFC library: System-defined path
```

```

-----
| Manifest |
-----
Manifest-Version: 1.0
Ant-Version: Apache Ant 1.6.4
Created-By: 1.3.1_18-b01 (Sun Microsystems Inc.)
Specification-Title: SAP Java Connector
Specification-Version: 2.1.8
Specification-Vendor: SAP AG, Walldorf
Implementation-Title: com.sap.mw.jco
Implementation-Version: 20070108 2139 [2.1.8 (2006-12-11)]
Implementation-Vendor-Id: com.sap
Implementation-Vendor: SAP AG, Walldorf
Main-Class: com.sap.mw.jco.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.

### 2.2.1 User Parameters

The following table lists and describes User parameters.

**Table 2-1**

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.
Authentication Mode	How the connection is validated.	Selection, see 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

The following table lists and describes System Settings (Application Server) parameters.

**Table 2–2**

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 meta data 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

The following table lists and describes System Settings (Message Server) parameters.

**Table 2–3**

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

The following table lists and describes Connection Pool parameters.

**Table 2–4**

Parameter	Description	Example	Comment
Connection pool	A name for a unique pool of client connections.	"foo"	A pool is a set of client-connections to a certain destination with the same logon data. The pool automatically creates new connections to the specified remote system or returns an existing connection. The reusing of existing connections can increase the performance of your application by avoiding logging on to the remote server.

**Table 2–4 (Cont.)**

Parameter	Description	Example	Comment
Connection Pool Size	Maximum number of connections for the pool.	10	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).	5	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, a JCO.Exception with the key JCO_ERROR_RESOURCE are thrown. The default value is 30 seconds.

## 2.2.5 SAP Gateway Parameters

The following table lists and describes SAP Gateway parameters.

**Table 2–5**

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

The following table lists and describes ALE parameters.

**Table 2-6**

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 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

The following table lists and describes Global Processing parameters.

**Table 2-7**

Parameter	Description	Example	Comment
Error Handling	Specifies the error handling method of the adapter.	Selection, see next column.	Creates error document writes an exception document with the full error text to the output destination.  Throws exception 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 next column.	Off - default  Sends Commit Request to Application Server in the document. If there is a commit error, 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).  See your SAP DB admin for your site's recommended setting.

**Table 2-7 (Cont.)**

Parameter	Description	Example	Comment
TRACE	Turns on the SAP Java connectors trace behavior.	Selection, see 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.

## 2.2.8 SNC Parameters

The following table lists and describes SNC parameters.

**Table 2-8**

Parameter	Description	Example	Comment
SNC mode	Flag for activating SNC.	1 (on)	Required.
SNC library path	Specifies the path and file name of the external library.	C:\SAP J2EE_Engine\SAPCryptolib\sapcrypto.dll	The default is the system-defined library as defined in the environment variable SNC LIB.
SNC level	Specifies the level of protection to use for the connection.	Selection, see 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 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.





---

## 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 discusses the following topics:

- Starting Application Explorer
- Configuring Repository Settings
- Creating a Repository Configuration
- Establishing a Connection (Target) for SAP R/3
- Viewing Application System Objects
- Creating XML Schemas
- Generating WSDL (J2CA Configurations Only)
- Creating and Testing a Web Service (BSE Configurations Only)
- Configuring an Event Adapter

### 3.1 Starting Application Explorer

To start Application Explorer:

1. Ensure that Oracle WebLogic Server is started, which is where Application Explorer is deployed.
2. Start Application Explorer by clicking the Windows **Start** menu, selecting **All Programs**, **Oracle Application Adapters**, and clicking **Application Explorer**.



You can also start Application Explorer by executing the **ae.bat** file, which is located in the following directory:

```
C:\oracle\Middleware\home_0309\oracle_
SOA1\soa\thirdparty\ApplicationAdapters\tools\iwaeb\bin\ae.bat
```

It is a good practice to create a shortcut for the **ae.bat** file on your desktop.

If you are using a UNIX or Linux platform you can start Application Explorer by executing the **iwae.sh** file.

Application Explorer starts. You are ready to define new targets to your SAP R/3 system.

## 3.2 Configuring Repository Settings

You need not configure BSE for a file-based repository because it is configured during the Oracle installation.

### 3.2.1 Configuring the Database Repository for BSE

After BSE is deployed to Oracle WebLogic Server, you can configure it through the BSE configuration page. This configuration is required only when using a database repository with BSE.

---

---

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

---

---

To configure BSE:

1. Execute the `iwse.ora` SQL script on the system where the database is installed.

The `iwse.ora` SQL script is located in the following directory:

```
C:\oracle\Middleware\Oracle_SOA1\soa\thirdparty\ApplicationAdapters\etc
```

This script creates the required tables that are used to store the adapter configuration information in the database. These tables are used by Application Explorer and by adapters during design time and run time. It is recommended that you use the same credentials to create the database repository and also in the `web.xml` file for database user credentials.

```
C:\oracle\Middleware\Oracle_SOA1\soa\thirdparty\ApplicationAdapters\etc>sqlplus
```

```
SQL*Plus: Release 10.1.0.2.0 - Production on Tue Dec 27 18:10:44 2005  
Copyright (c) 1982, 2004, Oracle. All rights reserved.
```

```
Enter user-name: scott
```

```
Enter password: scott1
```

```
Connected to:
```

```
Oracle Database 11g Enterprise Edition Release 11.1.1.2.0 - Production  
With the Partitioning, OLAP and Data Mining options
```

```
SQL>@ iwse.ora
```

2. Open the following page in your browser:

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

Where *host name* is the system where BSE is installed and *port* is the HTTP port for Oracle WebLogic Server.

For example,

```
http://localhost:8001/ibse
```

---

**Note:** If you are accessing this page for the first time, it may take longer to load.

---

3. Log on when prompted.

Enter the user ID and password, for example:

- User name: webllogic
- Password: welcome1

The BSE configuration page is displayed.

Property Name	Property Value
<b>System</b>	
Language	English ▼
Adapter Lib Directory	../ApplicationAdapters/lib
Encoding	UTF-8 ▼
Debug Level	DEBUG ▼
Number of Async. Processors	0 ▼
<b>Repository</b>	
Repository Type	File System ▼
Repository Url	file://C:\oracle\Middleware\Oracle_SO
Repository Driver	
Repository User	
Repository Password	
Repository Pooling	<input type="checkbox"/>

4. The **ojdbc14.jar** file must be copied to the following directory:

C:\oracle\Middleware\Oracle\_SOA1\soa\thirdparty\ApplicationAdapters\lib

5. Ensure that the Adapter Lib Directory parameter specifies the path to the lib directory, for example:

C:\oracle\Middleware\Oracle\_SOA1\soa\thirdparty\ApplicationAdapters\lib

After you specify the path, adapters in the lib directory are available to BSE.

**Note:** The Repository URL field specifies where the file system repository is located. To use a database repository, you must enter the repository connection information. For the initial verification, use a file system repository.

6. Click **Save**.

### 3.2.1.1 Configuring BSE System Settings

To configure BSE system settings:

1. Display the **BSE configuration** page in a browser:

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

Where *host name* is the system where BSE is installed and *port* is the port number on which BSE is listening.

**Note:** The server to which BSE is deployed must be running.

The BSE settings pane is displayed, as shown in the following image.

Property Name	Property Value
<b>System</b>	
Language	English
Adapter Lib Directory	../ApplicationAdapters/lib
Encoding	UTF-8
Debug Level	DEBUG
Number of Async. Processors	0

2. Configure the system settings.

The following table lists the parameters with descriptions of the information to provide.

Parameter	Description
Language	Specify the required language.
Adapter Lib Directory	Enter the full path to the directory where the adapter jar files reside.
Encoding	Only UTF-8 is supported.
Debug Level	Specify the debug level from the following options: <ul style="list-style-type: none"> <li>■ None</li> <li>■ Fatal</li> <li>■ Error</li> <li>■ Warning</li> <li>■ Info</li> <li>■ Debug</li> </ul>
Number of Async Processors	Select the number of asynchronous processors.

The following image shows all fields and check boxes for the Repository pane.

The screenshot shows a configuration window titled "Repository". It contains the following fields and controls:

- Repository Type:** A dropdown menu currently set to "File System".
- Repository Url:** A text input field containing the value "file://C:\oracle\Middleware\Oracle\_SC".
- Repository Driver:** An empty text input field.
- Repository User:** An empty text input field.
- Repository Password:** An empty text input field.
- Repository Pooling:** A checkbox that is currently unchecked.
- Save:** A grey button located at the bottom right of the pane.

**3. Configure the repository settings.**

BSE requires a repository to store transactions and metadata required for the delivery of Web services.

See "[Configuring a File System Repository](#)" on page 3-6 for more information.

The following table lists the parameters with descriptions of the information to provide.

Parameter	Description
Repository Type	Select a repository from the list: <ul style="list-style-type: none"> <li>■ Oracle</li> <li>■ File (Do not use for BSE in production environments.)</li> </ul>
Repository URL	Enter the JDBC URL to use when opening a connection to the database. For example, the following repository URL format is used when connecting to Oracle: <pre>jdbc:oracle:thin:@host name:port;SID</pre>

Parameter	Description
Repository Driver	Provide the JDBC driver class to use when opening a connection to the database (optional). For example, the following repository driver format is used when connecting to Oracle:  oracle.jdbc.driver.OracleDriver
Repository User	Enter a valid user ID to use when opening a connection to the database.
Repository Password	Enter a valid password that is associated with the user ID.
Repository Pooling	If selected, repository pooling is used. This option is disabled by default.

4. Click **Save**.

### Configuring a File System Repository

If you do not have access to a database for the repository, you can store repository information in an XML file on your local system. However, a file system repository is less secure and efficient than a database repository. When BSE is first installed, it is automatically configured to use a file system repository.

---

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

---

The default location for the repository on Windows is:

```
C:\oracle\Middleware\user_projects\domains\base_domain\servers\soa_server1\stage\ibse\ibse.war
```

On other platforms, use the corresponding location.

If you are using a file system repository, you are not required to configure any additional BSE components.

## 3.2.2 Configuring the Database Repository for J2CA

This section describes how to configure the database repository for J2CA.

1. Execute the `iwse.ora` SQL script on the system where the database is installed.

The `iwse.ora` SQL script is located in the following directory:

```
C:\oracle\Middleware\Oracle_SOA1\soa\thirdparty\ApplicationAdapters\etc
```

This script creates the required tables that are used to store the adapter configuration information in the database. These tables are used by Application Explorer and by adapters during design time and run time. It is recommended that you use the same credentials to create the database repository and also in the `ra.xml` file for database user credentials.

```
C:\oracle\Middleware\Oracle_SOA1\soa\thirdparty\ApplicationAdapters\etc>sqlplus
```

```
SQL*Plus: Release 10.1.0.2.0 - Production on Tue Dec 27 18:10:44 2005  
Copyright (c) 1982, 2004, Oracle. All rights reserved.
```

```
Enter user-name: scott  
Enter password: scott1
```

```
Connected to:
Oracle Database 11g Enterprise Edition Release 11.1.1.2.0 - Production
With the Partitioning, OLAP and Data Mining options
```

```
SQL>@ iwse.ora
```

2. Create the `jcatransport.properties` file and save it in the following directory:

```
C:\oracle\Middleware\Oracle_
SOA1\soa\thirdparty\ApplicationAdapters\config\J2CA_SampleConfig
```

---

**Note:** The `jcatransport.properties` file is required for each J2CA configuration that is created using Application Explorer. The J2CA configuration folder, for example, `J2CA_SampleConfig`, is named according to the configuration name that is specified in Application Explorer.

---

3. Enter values for `iwafjca.repo.url`, `iwafjca.repo.user` and `iwafjca.repo.password` fields in the newly created `jcatransport.properties` file, as shown in the following example:

```
iwafjca.repo.url=jdbc:oracle:thin:@90.0.0.51:1521:orcl
iwafjca.repo.user=scott
iwafjca.repo.password=scott1
```

The following table lists the parameters with descriptions of the information to provide.

Parameter	Description
<code>iwafjca.repo.url</code>	Specify the JDBC URL to use when opening a connection to the database. For example, the following repository URL format is used when connecting to Oracle: <code>jdbc:oracle:thin:@host name:port;SID</code>
<code>iwafjca.repo.user</code>	Specify a valid user ID to use when opening a connection to the database.
<code>iwafjca.repo.password</code>	Specify a valid password that is associated with the user ID.

4. The `ojdbc14.jar` file must be copied to the following directory:

```
C:\oracle\Middleware\Oracle_SOA1\soa\thirdparty\ApplicationAdapters\lib
```

5. Navigate to the following directory:

```
C:\oracle\Middleware\Oracle_
SOA1\soa\thirdparty\ApplicationAdapters\iwafjca.rar\META-INF
```

6. Open the `ra.xml` file in a text editor.
7. Provide the JDBC connection information as a value for the `IWAYRepo_URL` property.
8. Provide a valid user name for the `IWAYRepo_User` property.

9. Provide a valid password for the `IWAYRepo_Password` property.
10. Save your changes to the `ra.xml` file.

### 3.2.2.1 Password Encryption

When creating J2CA configurations, you can also encrypt a password using Application Explorer and use this value in the `jcatransport.properties` and `ra.xml` files for added security.

#### Configuring Password Encryption

To encrypt a password:

1. Open Application Explorer.
2. Click Help and select Encryption.  
The Encryption dialog is displayed.
3. Type a password in the Password field and click OK.  
An encrypted version of the password displays in the Encryption field.
4. Copy the password.
5. In the `jcatransport.properties` file, which is used during design time, replace the existing password with the encrypted value only if you are using a database repository.

The following is a sample of the `jcatransport.properties` file where the password is replaced:

```
iwafjca.log.level=DEBUG
iwafjca.repo.url=jdbc:oracle:thin:@172.30.166.100:1521:orcl
iwafjca.repo.user=scott
iwafjca.repo.password=ENCR (318931973183297321831293164323332123227)
```

6. In the `ra.xml` file, which is used during run time, replace the existing password with the encrypted value for the `IWayRepoPassword` element. This is applicable for file system and database repositories.
7. Restart the Oracle WebLogic Server.

## 3.3 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 kinds of repository configurations, Web services and J2CA, depending on the container to which the adapter is deployed.

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. See "[Adapter Features](#)" on page 1-1 for more information.

### 3.3.1 Creating a Configuration for BSE

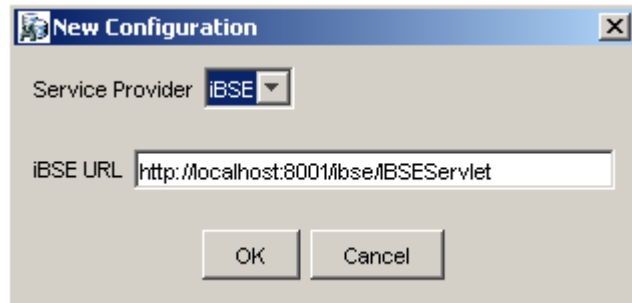
To create a configuration for BSE using Application Explorer, you must first define a new configuration.



### 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.



4. From the **Service Provider** list, select **iBSE**.
5. In the **iBSE URL** field, accept the default URL or replace it with a different URL 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.



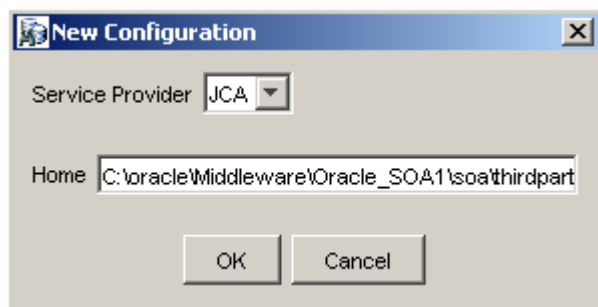
### 3.3.2 Creating a Configuration for J2CA

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

#### Defining a New Configuration for J2CA

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.



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

A node representing the new configuration appears beneath the root Configurations node.



The Oracle Adapter J2CA configuration file is stored in:

C:\oracle\Middleware\Oracle\_SOA1\soa\thirdparty\ApplicationAdapters\config\configuration\_name

Where *configuration\_name* is the name of the configuration you created (for example, SampleConfig).

### HTTP Repository Connection

J2CA users can create an HTTP repository connection, which enables them to generate and store WSDL documents remotely. Perform the following steps to create an HTTP repository connection in Application Explorer. To use the HTTP repository, ensure that the iwjaivp test tool(jca-app-adapter-test) is successfully deployed and running.

1. Start the Application Explorer.
2. Right-click the **Configurations** node in the left pane and select **New**.  
The New Configuration dialog opens.
3. Type a name for the configuration and click **OK**.
4. Select **JCA** from the Service Provider list box and enter an HTTP target value in the Home field.

Use the following format for the HTTP target value:

```
http://host name:port/iwafjca/JCAServlet
```

For example:

```
http://iwserv14:7777/iwafjca/JCAServlet
```

5. Click **OK**.

The new HTTP repository connection is added to the Configurations node.

Once you connect to the remote server, you can create new Adapter targets, generate WSDL documents, and store them in the remote server.

**Note:** When you configure an Adapter target with the J2CA HTTP repository, you are not required to restart the Oracle WebLogic Server for run time purposes.

### 3.3.3 Connecting to a BSE or J2CA Configuration

To connect to a new configuration:

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

Nodes appear for Adapters, Events, and Business Services (also known as Web services). The Business Services node is only available for BSE configurations. If you are connected to a J2CA configuration, 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. As a result, you can disregard the Events node that appears for a BSE configuration.

The following is an example of a BSE configuration named SampleConfig:



- Use the **Adapters** folder to create inbound interaction with SAP R/3. For example, you use the SAP node in the Adapters folder to configure a service that updates SAP R/3.
- Use the **Events** folder to configure listeners that listen for events in SAP R/3.
- Use the **Business Services** folder (available for BSE configurations only) to test Web services created in the Adapters folder. You can also control security settings for the Web services by using the security features of the Business Services folder.

You can now define new targets to SAP R/3.

## 3.4 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.

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.

### 3.4.1 Defining a Target to SAP R/3

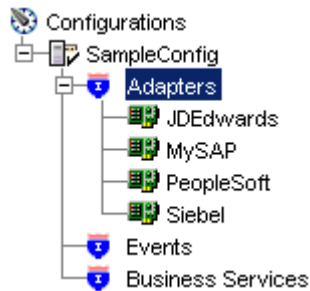
To connect to SAP R/3 for the first time, you must define a new target. Oracle Application Adapter for SAP R/3 supports SAP R/3 standard security and the additional protocol of SNC. Once connected to the SAP R/3 application server, application security is managed by user ID, roles and profiles. For more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

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



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



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

- a. In the **Name** field, enter a descriptive name, for example, **SAPTarget**.
- 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).

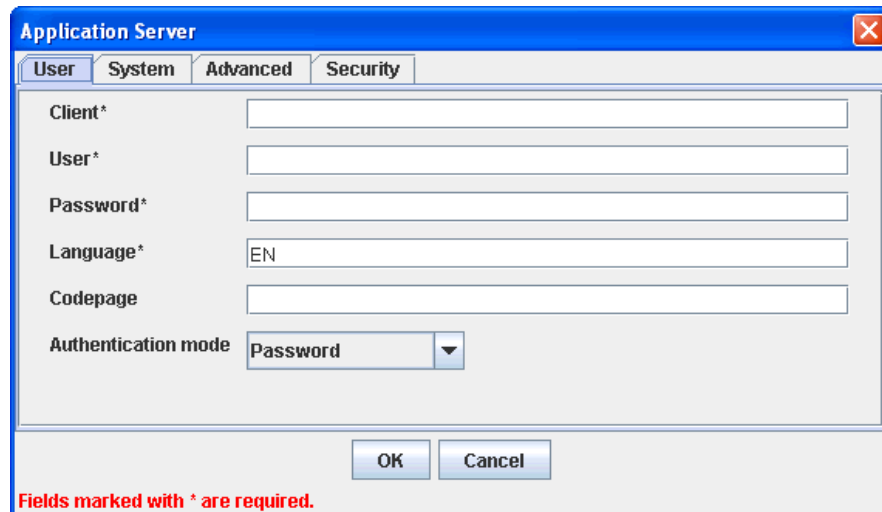
---

**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 kind 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.



The screenshot shows the 'Application Server' dialog box with the 'User' tab selected. The fields are: Client\* (empty), User\* (empty), Password\* (empty), Language\* (EN), Codepage (empty), and Authentication mode (Password). The 'OK' and 'Cancel' buttons are at the bottom. A red note at the bottom left says 'Fields marked with \* are required.'

The following tabs are available:

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

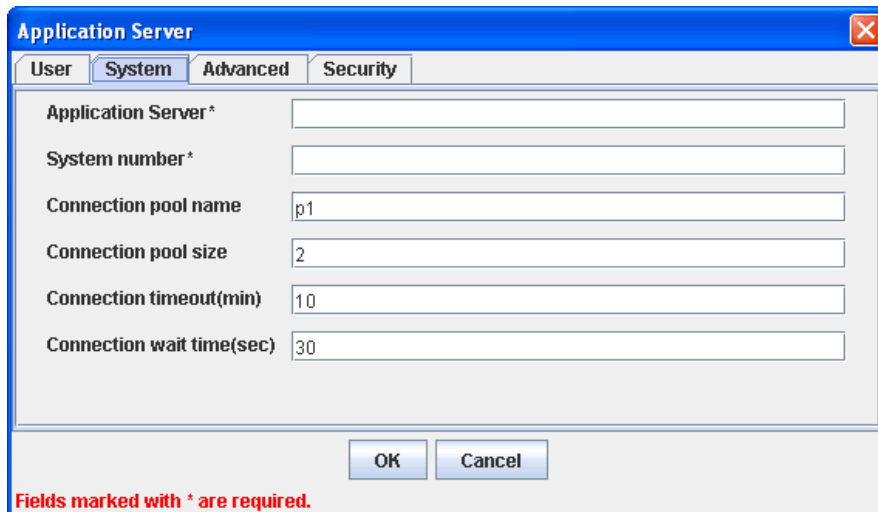
4. For the **User** tab (required), enter the appropriate information for your SAP R/3 target based on the information in the following table.

**Table 3–1 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 drop-down list.

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

5. For the **System** tab (required), enter the appropriate information for your SAP R/3 target based on the information in this section.



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 3–2 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.
Connection pool name	The name of your SAP R/3 connection pool. A default value, p1, is provided.
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, 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>

6. 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.

**Table 3–3 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 drop-down list.
Error Handling	<p>From the list In the event of an exception, you can select <b>Creates Error Document</b> or <b>Throws Exception</b>. To receive more detailed error messages, select <b>Creates Error Document</b>.</p> <p>As a rule:</p> <ul style="list-style-type: none"> <li>▪ If your application is <b>Java centric</b>, select <b>Throws Exception</b> so that code components can catch the exception and react accordingly.</li> <li>▪ If your application is <b>document based</b>, select <b>Creates Document</b> to create an XML document that contains the Java exception.</li> </ul> <p>It is up to your application to read the XML document and obtain the error.</p>

**Table 3–3 (Cont.) Advanced Tab Parameters**

Target Parameter	Description
Commit with Wait	<p>If a high degree of accuracy is required in your application, select the <b>Commit with Wait</b> check box.</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, the error message “Posting could not be carried out” is returned, and the adapter rolls back the transaction. If in writing to the database, a database error occurs, a short dump is issued in the database records of SAP, but no message is returned to the adapter about the failure.</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:</p> <p>C:\oracle\Middleware\home_GA\Oracle_SOA1\soa\thirdparty\ApplicationAdapters\tools\iwae\bin</p> <p>Run Time:</p> <p>C:\oracle\Middleware\home_GA\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"> <li>■ rfc02664_04332.trc</li> <li>■ JCO100226_043846994.trc</li> </ul>

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



The screenshot shows the 'Application Server' dialog box with the 'Security' tab selected. The fields are as follows:

- SNC mode: 0
- SNC partner: (empty text box)
- SNC level: 1
- SNC name: (empty text box)
- SNC library path: (empty text box)

Buttons: OK, Cancel

Note: 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.

**Table 3–4 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, 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, 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.

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

After the extraction finishes, the new target, SAPTarget, appears under the MySAP adapter node.



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

See ["Creating XML Schemas"](#) on page 3-20 for information on how to create schemas for the adapter.

### 3.4.2 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. Provide the correct password.
6. Right-click the target name and select Connect.

The x icon disappears, indicating that the node is connected.



### 3.4.3 Managing 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.

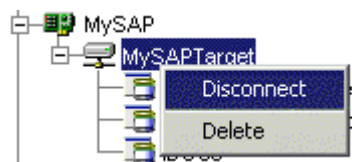
You can modify the connection parameters when your system properties change. After you disconnect, you can modify an existing target.

- Delete a connection that is no longer needed.

#### 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.



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.

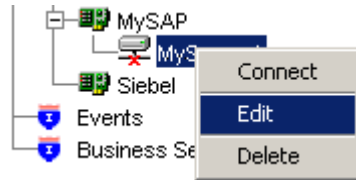


### Modifying Connection Parameters

After you create a target for SAP R/3 using Application Explorer, you can edit any of the information that you provided previously.

To edit a target:

1. Verify that the target you want to edit is disconnected.
2. Right-click the target and select **Edit**.



The Application Server dialog displays the target connection information.

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

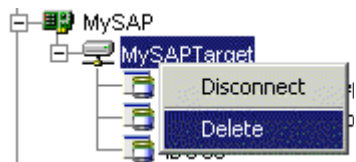
### Deleting a Connection to SAP R/3

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**.



The node disappears from the list of available connections.

## 3.5 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, 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.

See the *SAP R/3 User's Guide* for more information.

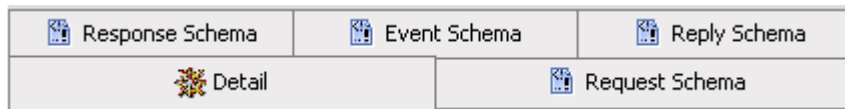
### 3.6 Creating XML Schemas

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 3-18.
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 following screen appears on the right.



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

### 3.7 Generating WSDL (J2CA Configurations Only)

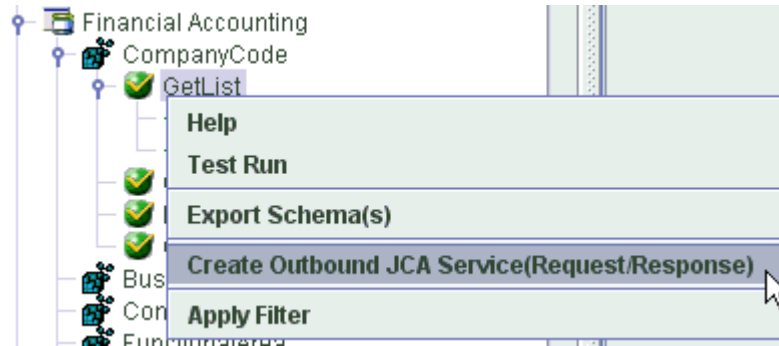
The Web Service Definition Language (WSDL) description of a service enables you to make the service available to other services within a host server. You use Application Explorer to create both request-response (outbound) and event notification (inbound) J2CA services of the adapter.

**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:

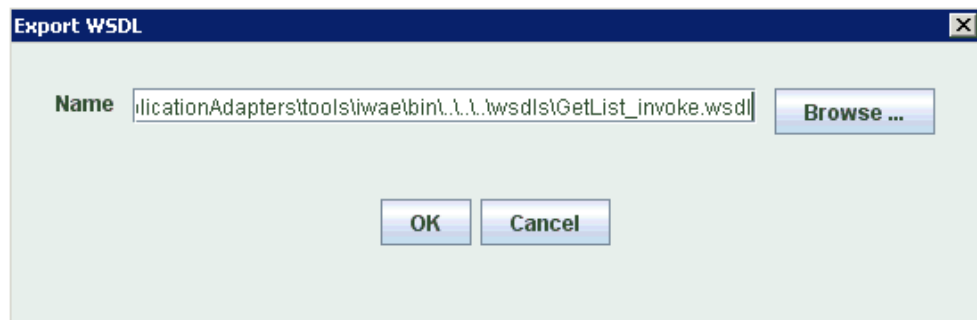
1. After you create a schema, right-click the respective object.

The following menu is displayed:



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

The Export WSDL dialog is displayed.



3. Accept the default name 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. See "[Generating WSDL for Event Integration](#)" on page 5-35 for a detailed example.

### 3.8 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.

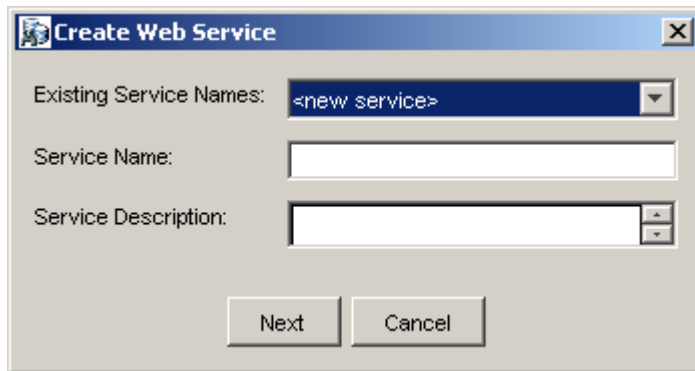
---

## 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.



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, 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, enter a descriptive name for the method.
  - c. In the **Method Description** field, enter a brief description of the method.
  - 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.

## 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, an input field appears.

5. Enter the appropriate input.

6. Click **Invoke**.

The BSE Servlet page displays the results.

### Identity Propagation

If you test or run a Web service using a third party XML editor, the Username 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.

### Oracle ERP Adapter WSDL Definition Limitations

This section describes certain limitations with WSDL definitions that are generated by Oracle ERP Adapters, and provides workarounds where applicable.

#### Qualified/Unqualified Options

The *qualified* and *unqualified* options are not enforced by the Oracle ERP Adapter when WSDL definition are generated. For example:

1. Export a WSDL with the *qualified* option specified.
2. Create a Web service based on this WSDL.
3. Send XML data that contains unqualified elements to this Web service.

A correct response document is received.

#### Namespaces

Namespaces that are defined in the generated WSDL definition are not enforced by the Oracle ERP Adapter. For example:

1. Create a Web service based on a generated WSDL.
2. Send XML data where all namespaces have been removed to this Web service.  
A correct response document is received.

#### Unchecked Data

The Oracle ERP Adapter does not verify if XML data conforms to the generated WSDL definition. For demonstration purposes only, the following example uses SAP R/3:

1. Create a Web service for the *RFC\_SYSTEM\_INFO* function module.
2. Invoke this Web service (*RFC\_SYSTEM\_INFO*) with XML data pertaining to *BAPI\_COMPANYCODE\_GETLIST*.

A response document is received with data for *BAPI\_COMPANYCODE\_GETLIST*.

This behavior should be taken into consideration when designing an application using Oracle ERP Adapters. As a best practice, always validate the XML data that is received according to the WSDL definition before a Web service is invoked. Otherwise, all functions can be exposed on a system (for example, SAP R/3) if only one function is called. In this example, only the *RFC\_SYSTEM\_INFO* function module was called, but an end-user could have invoked any function on SAP R/3 using this function. Only a valid XML document that is recognized by the Oracle ERP Adapter for a given function is required.

## 3.9 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, your application is a consumer of this event.

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, 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. See "[Creating and Editing a Channel](#)" on page 3-24 for more information.

### 3.9.1 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.

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.

---

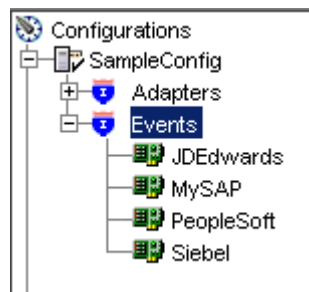
Events are not applicable when using a BSE configuration. You can configure events using a J2CA configuration only.

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.

## Creating a Channel

To create a channel:

1. Click the **Events** node.

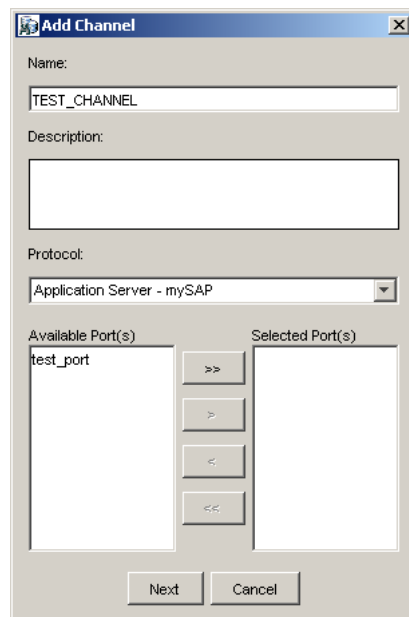


2. Expand the **MySAP** node.

The ports and channels nodes appear in the left pane.

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

The Add Channel dialog is displayed.



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 - mySAP** or **Message Server - mySAP**.
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 3–5 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.
Authentication Mode	The authentication mode you want to use when connecting to your SAP R/3 system. By default, Password is selected from the drop-down 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 3–6 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.

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 3–7 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.

**Table 3–7 (Cont.) Security Tab Parameters**

Target Parameter	Description
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 3–8 Advanced Tab Parameters**

Target Parameter	Description
IDOC Format	Select an IDOC type from the list: <ul style="list-style-type: none"> <li>■ XML (default)</li> <li>■ XML-CDATA-ENVELOPED</li> <li>■ NATIVE IDOC</li> </ul>
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 drop-down list.
SAP trace	Select this option to enable SAP traces. The SAP traces are stored in the following locations during design time and run time: Design Time: C:\oracle\Middleware\home_GA\Oracle_SOA1\soa\thirdparty\ApplicationAdapters\tools\iwae\bin Run Time: C:\oracle\Middleware\home_GA\user_projects\domains\base_domain The file name formats that are used for the SAP traces are as follows (for design time and run time): <ul style="list-style-type: none"> <li>■ rfc02664_04332.trc</li> <li>■ JCO100226_043846994.trc</li> </ul>
Processing Mode	Select the type of synchronous processing from the list. Possible values include REQUEST and REQUEST_RESPONSE.

9. Click **OK**.

The channel appears under the channels node in the left pane.



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

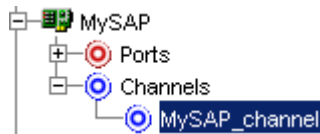
---

**Note:** If you are planning to integrate Oracle Application Adapter for SAP R/3 with BPEL inbound or Mediator inbound process components, do not start the channel. The channel is managed by the run-time server after the BPEL or Mediator process component is deployed. If you start the channel from Application Explorer for testing and debugging purposes, stop it before run-time (when working with BPEL or Mediator process components).

---

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

The channel you created becomes active.



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.

### 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.

### 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.

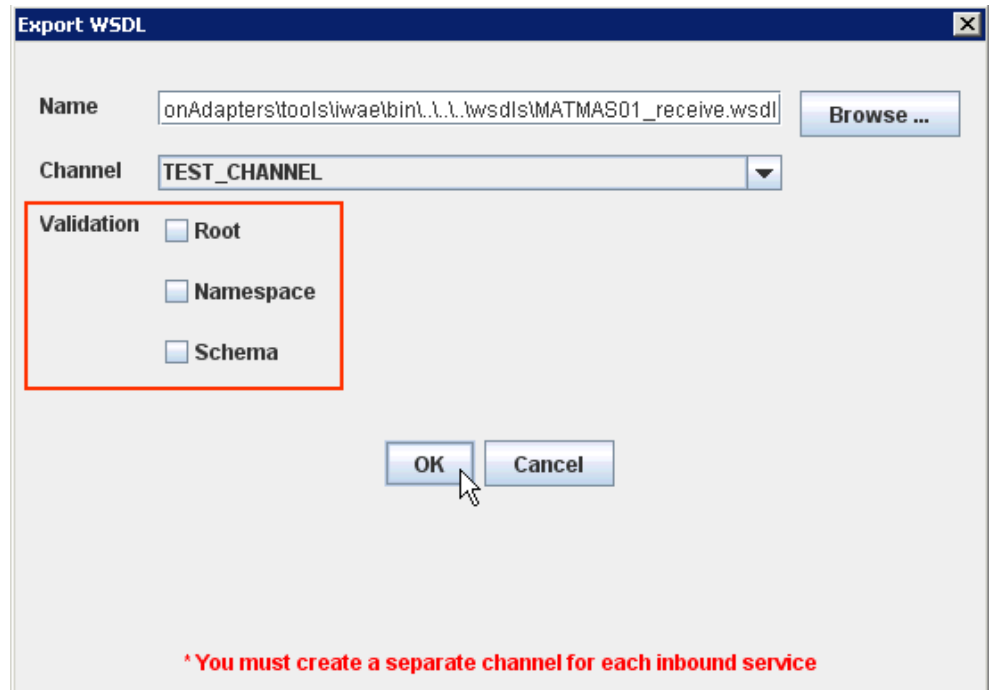
## 3.9.2 Schema Validation

Root validation, namespace validation, and schema validation for inbound processing (events) are supported for the Oracle Application Adapter for SAP R/3 with BPEL 11.1.1.

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.



- 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 option is selected, 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.

---

---

# Oracle WebLogic Server Deployment and Integration

This chapter describes Oracle WebLogic Server (OracleWLS) deployment and integration with Oracle Application Adapter for SAP R/3.

This chapter discusses the following topics:

- [Adapter Integration with Oracle WebLogic Server](#)
- [Deployment of Adapter](#)
- [Updating Adapter Configuration](#)

**See Also:**

- *Oracle WebLogic Server Adapter Concepts*

## 4.1 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.

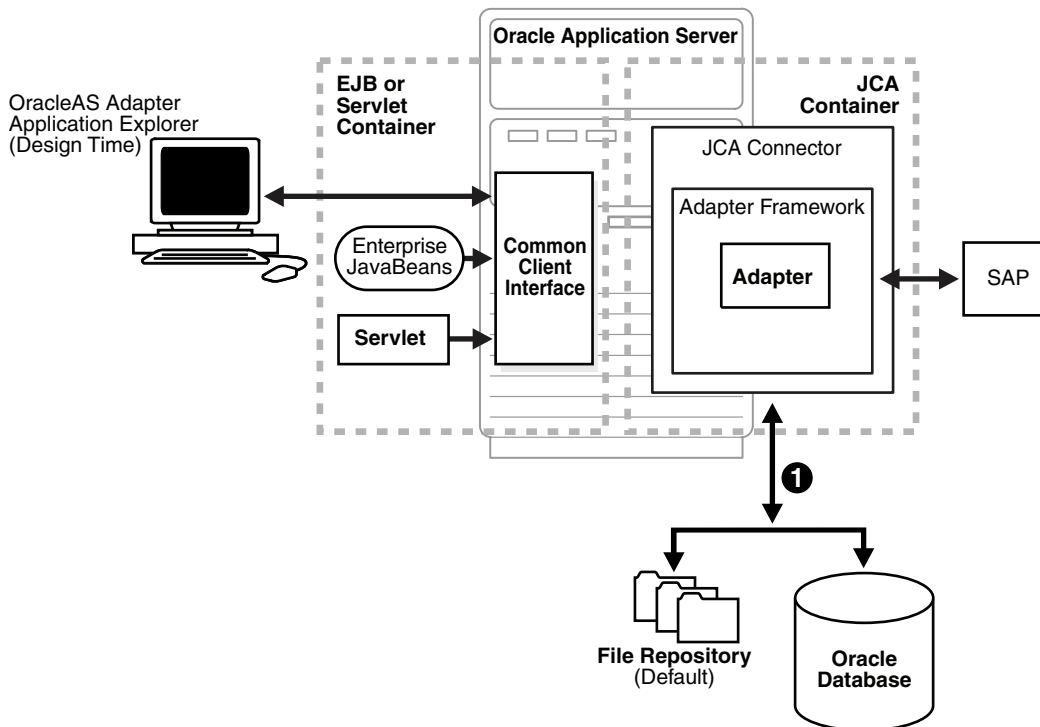
**See Also:**

- *Oracle WebLogic Server Adapter Concepts*

## 4.2 Deployment of Adapter

[Figure 4-1](#) 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 4–1 Oracle Application Server J2CA Architecture



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

**See Also:**

- *Oracle WebLogic Server Adapter Concepts*

## 4.3 Updating Adapter Configuration

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

```
C:\oracle\Middleware\Oracle_
SOA1\soa\thirdparty\ApplicationAdapters\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).

---

### 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>IWAFJCA10</display-name>
```



```

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

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

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

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

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

```

```

        <config-property-value></config-property-value>
    </config-property>
    <config-property>
        <config-property-name>LogLevel</config-property-name>
        <config-property-type>java.lang.String</config-property-type>
        <config-property-value>DEBUG</config-property-value>
    </config-property>
    <authentication-mechanism>
        <authentication-mechanism-type>BasicPassword</authentication-mechanism-type>
    </authentication-mechanism>
    <credential-interface>javax.resource.spi.security.PasswordCredential</credential-interface>
    </credential-interface>
    </authentication-mechanism>
    <reauthentication-support>true</reauthentication-support>
</resourceadapter>
</connector>

```

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

Parameter Name	Description
IWayHome	The base installation directory for the OracleWLS packaged application adapter.
IWayConfig	The adapter configuration name as defined in Application Explorer. For example, Oracle Application Adapter for 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, it overwrites 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.

### 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:

```

C:\oracle\Middleware\Oracle_
SOA1\soa\thirdparty\ApplicationAdapters\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, 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, 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.

---

### 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, 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 includes the following topics:

- [Overview](#)
- [Deployment of Adapter](#)
- [Configuring a New Application Server Connection](#)
- [Designing an Outbound BPEL Process for Service Integration](#)
- [Designing an Inbound BPEL Process for Event Integration](#)

## 5.1 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. See "[Generating WSDL \(J2CA Configurations Only\)](#)" on page 3-20 for more information.

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.

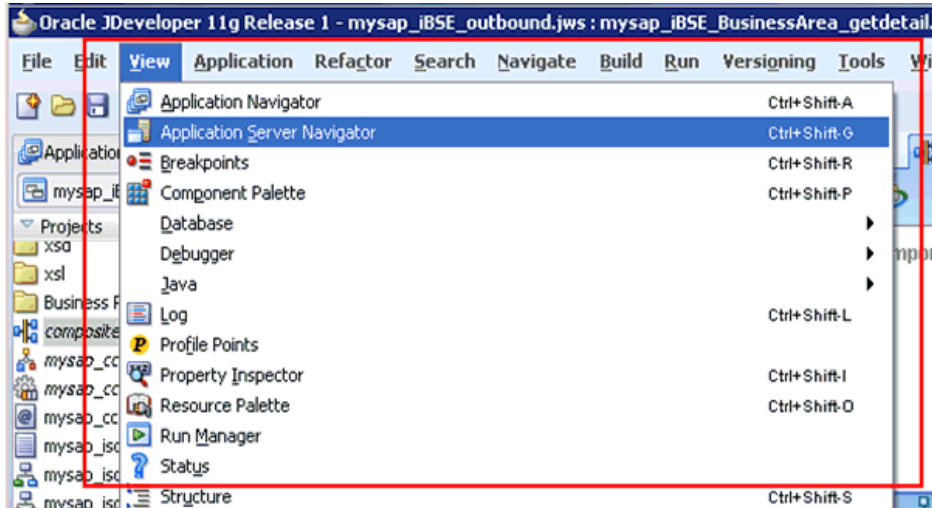
## 5.2 Deployment of Adapter

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.

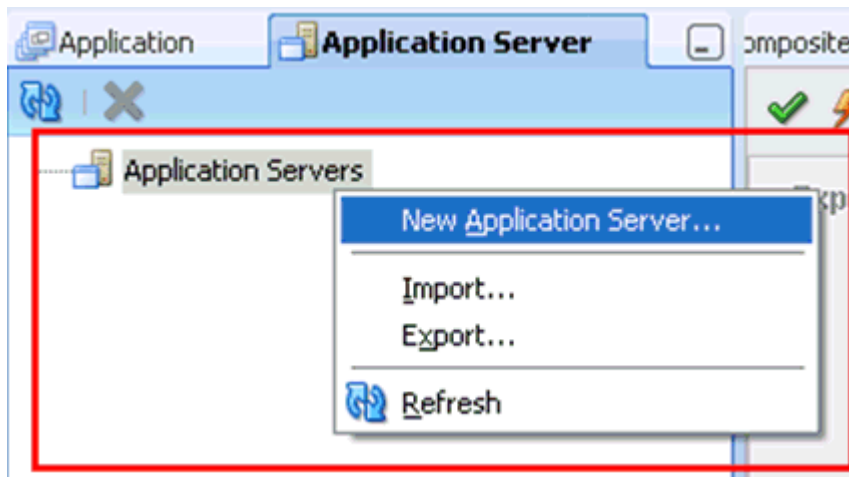
### 5.3 Configuring a New Application Server Connection

To configure a new Application Server connection in Oracle JDeveloper:

1. Open **Oracle JDeveloper** on your system.
2. From the menu bar, click **View** and select **Application Server Navigator**.

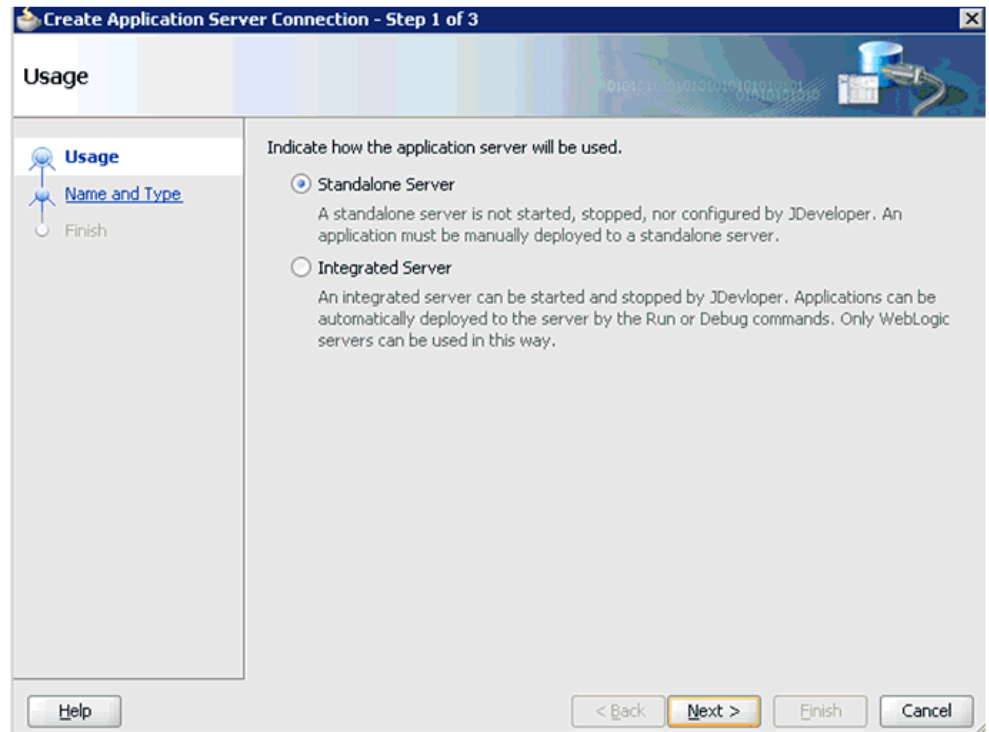


The Application Server tab is displayed.

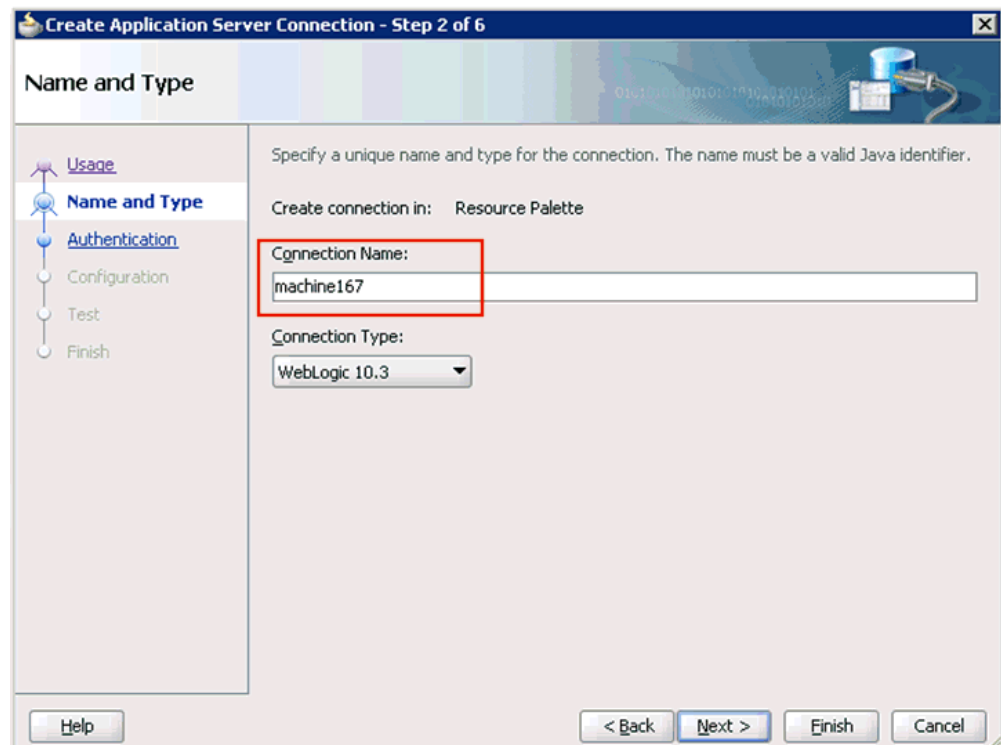


3. Right-click **Application Servers** and select **New Application Server**.

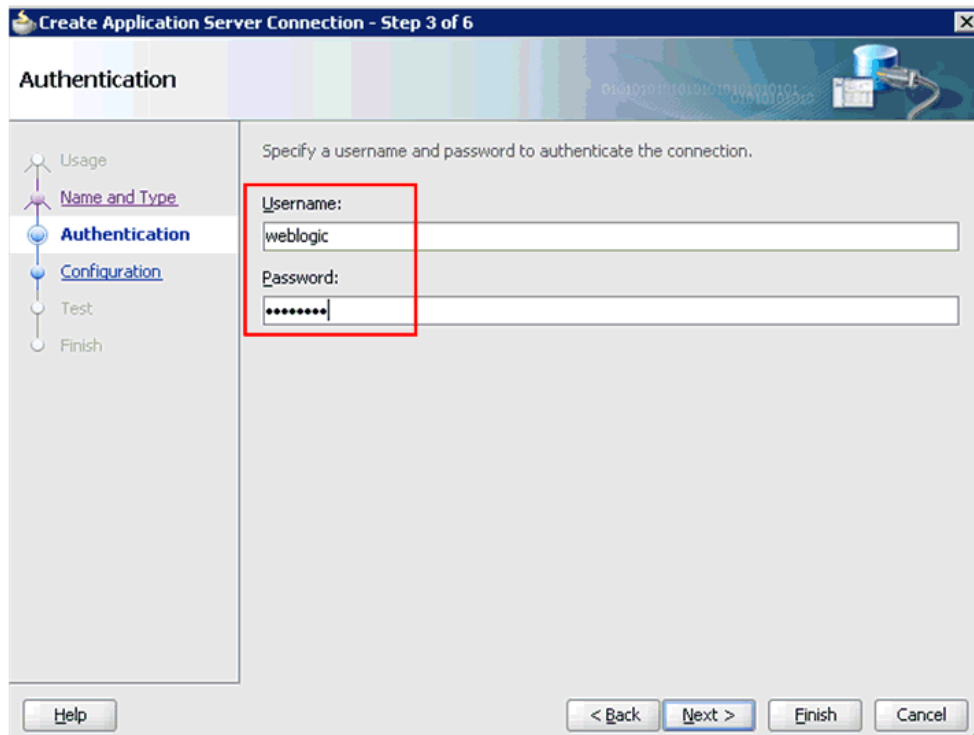
The Create Application Server Connection Wizard is displayed.



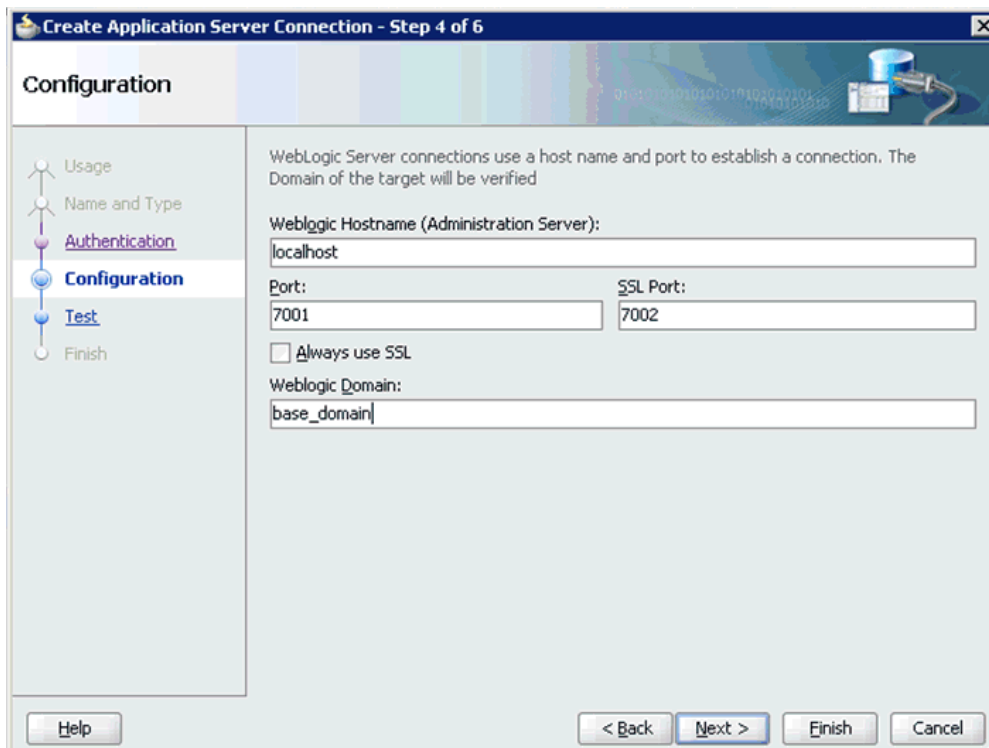
4. Accept the default selection (Standalone Server) and click **Next**.  
The Name and Type page is displayed.



5. Specify a new name for the Application Server connection and click **Next**.  
The Authentication page is displayed.



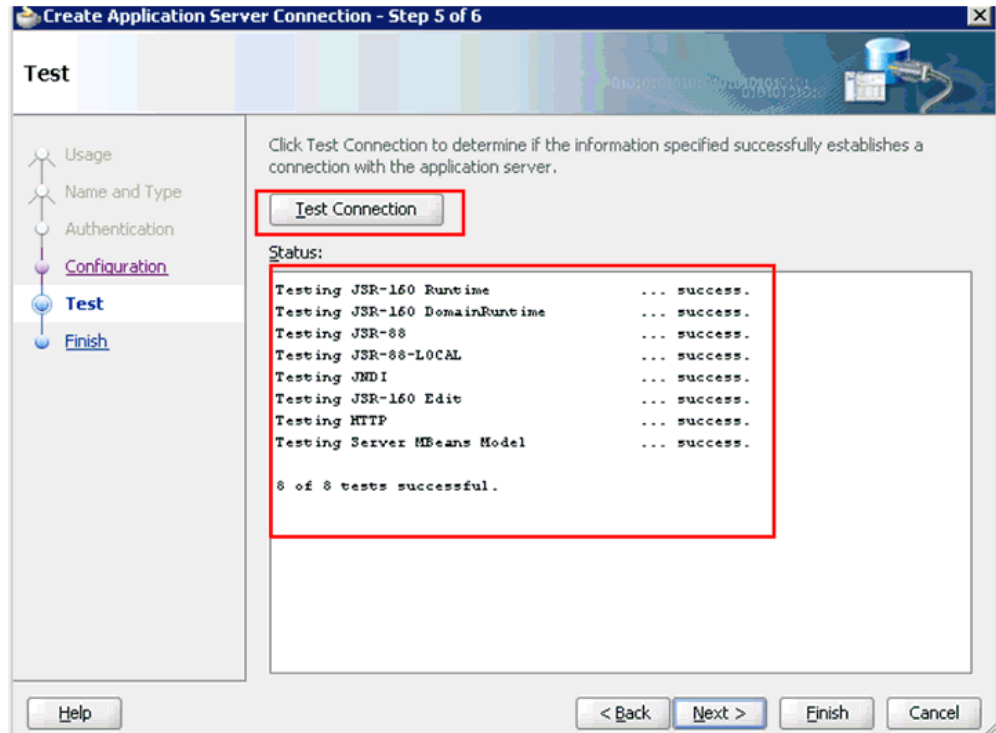
6. Specify a valid user name (for example, weblogic) and a password (for example, welcome1) for your new connection.
7. Click Next.  
The Configuration page is displayed.





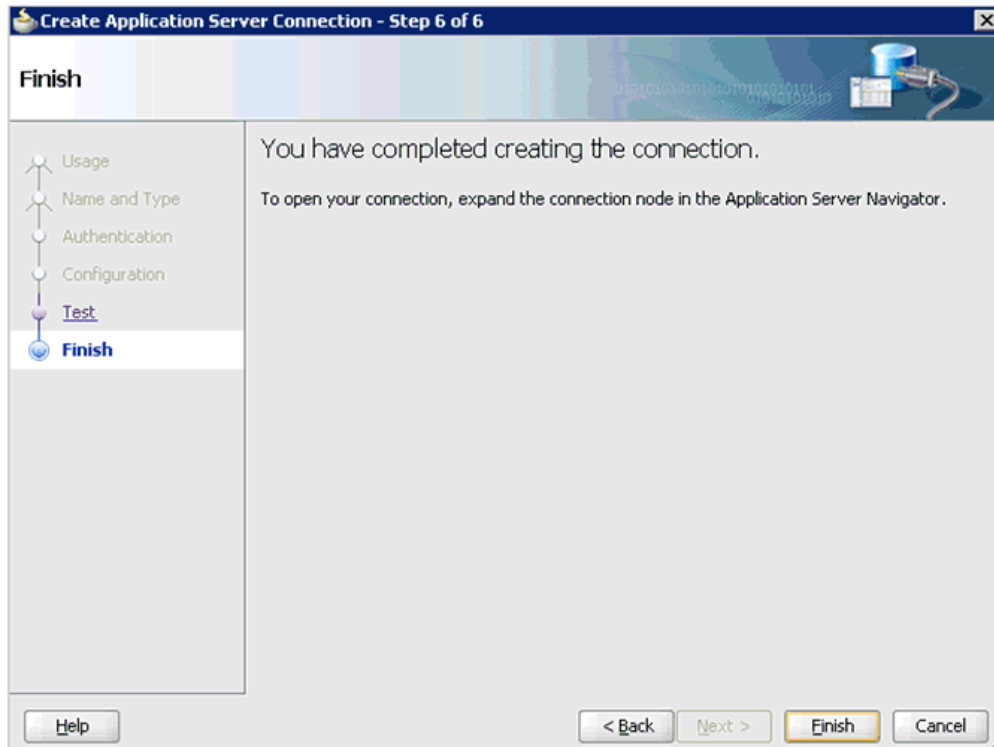
8. Specify the Oracle WebLogic host name (for example, localhost), which is the system IP where the process must deploy and Oracle WebLogic domain (for example, base\_domain).
9. Click **Next**.

The Test page is displayed.



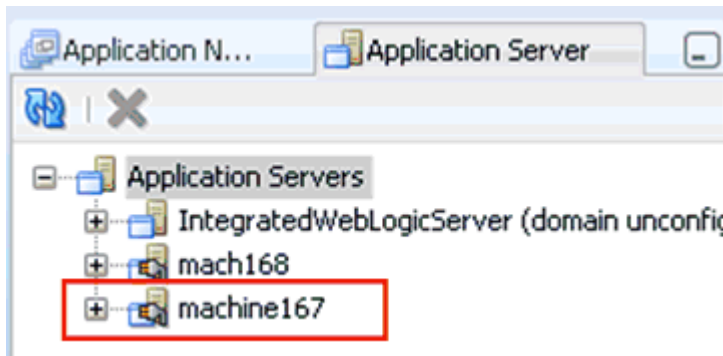
10. Click **Test Connection**.
11. Make sure that the test status is successful.
12. Click **Next**.

The Finish page is displayed.



**13. Click Finish.**

The new Application Server connection is listed in the left pane (Application Server tab), as shown in the following image.



## 5.4 Designing an Outbound BPEL Process for Service Integration

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

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

---



---

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

---



---

Before you design a BPEL process, you must generate the respective WSDL file using Application Explorer. See "[Generating WSDL for Request/Response Service](#)" on page 5-7 for more information.

### 5.4.1 Generating WSDL for Request/Response Service

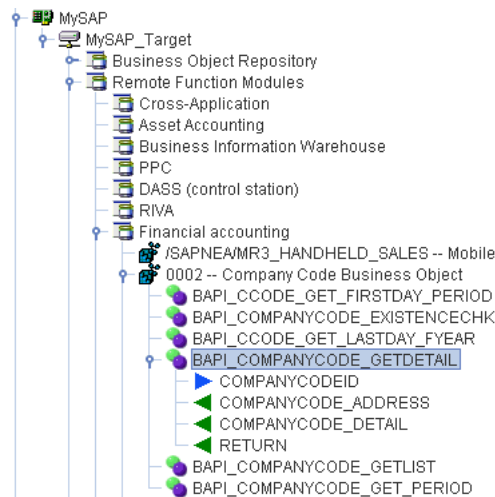
Perform the following steps to generate a WSDL for request/response service:

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

See "[Defining a Target to SAP R/3](#)" on page 3-11 for more information on defining a target and connecting to SAP R/3.

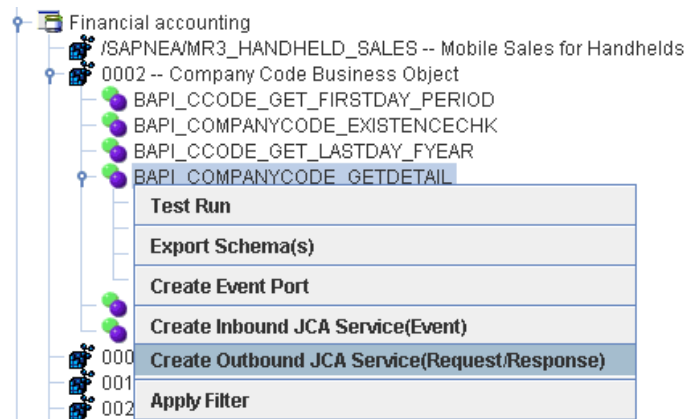
2. Expand the MySAP target to which you are connected.
3. Expand **Remote Function Modules, Financial Accounting, 0002 -- Company Code Business Object**, and then select **BAPI\_COMPANYCODE\_GETDETAIL**.

The following image shows a connected and expanded target.



4. Right-click the **BAPI\_COMPANYCODE\_GETDETAIL** node.

The following menu is displayed:



5. Click **Create Outbound JCA Service (Request/Response)**.

The Export WSDL dialog is displayed.



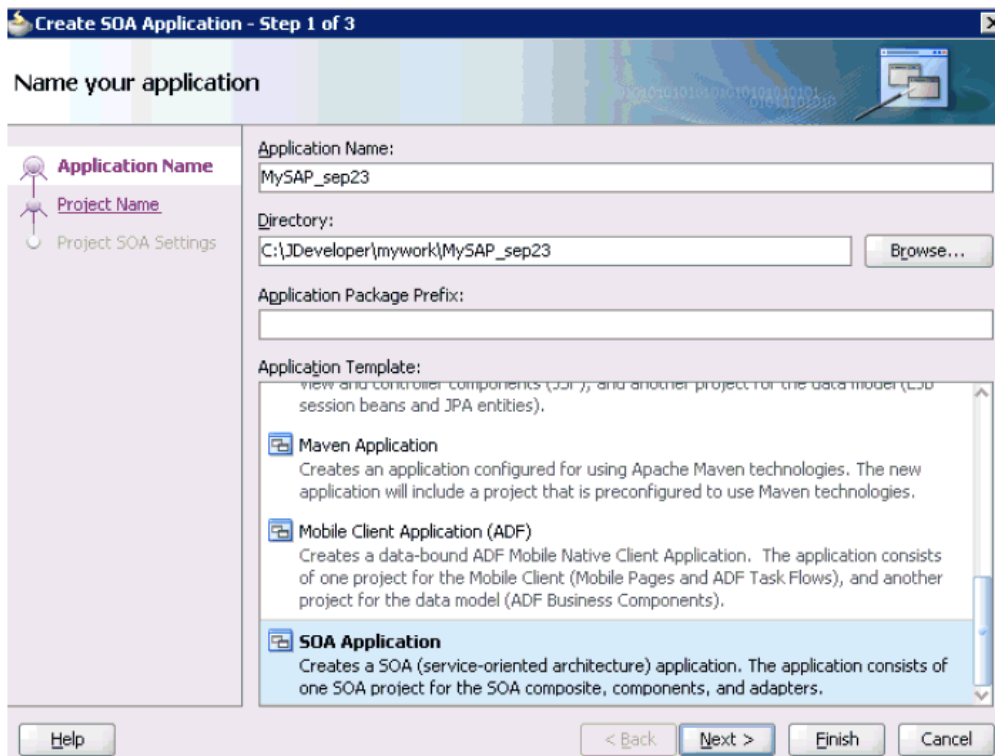
6. Click **OK**.

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.

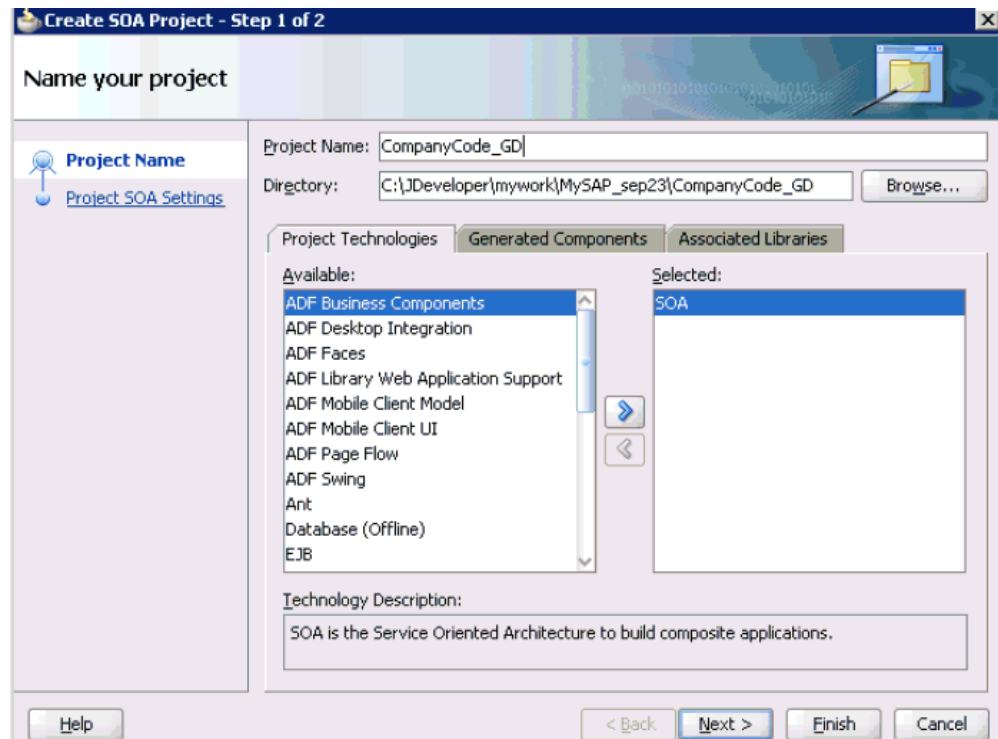
## 5.4.2 Creating an Empty Composite for SOA

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

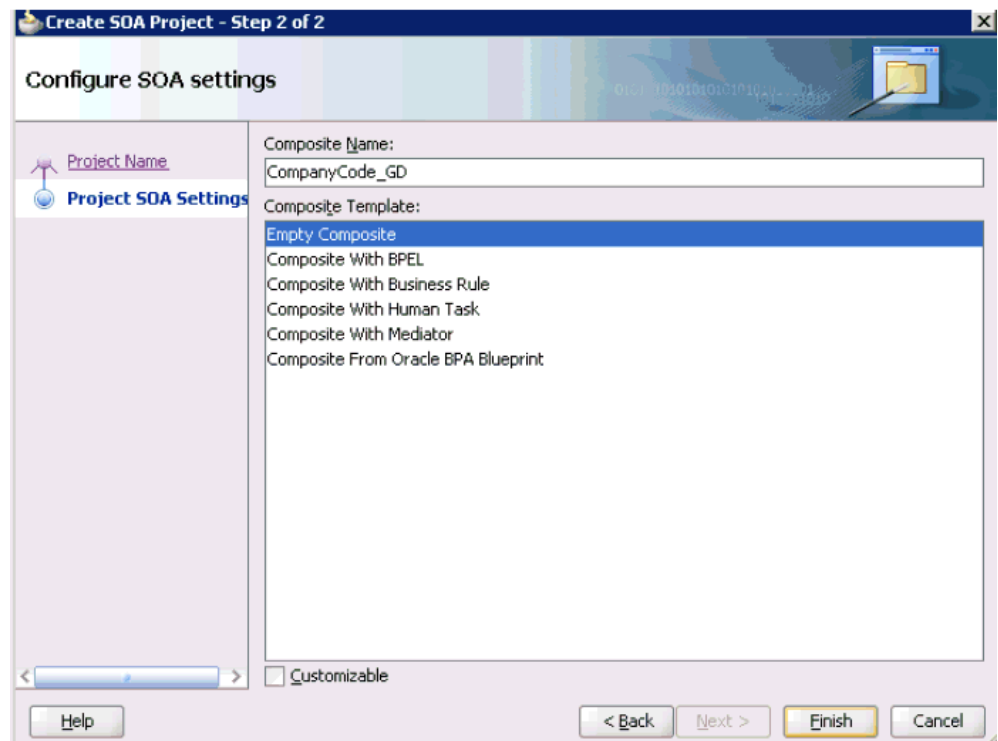
1. Create a new SOA application.
2. Enter name for the new SOA Application and click **Next**.



The Name your project page is displayed.



3. Enter a project name (for example, CompanyCode\_GD) and click **Next**.  
The Configure SOA settings page is displayed.



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

### 5.4.3 Defining a BPEL Outbound Process

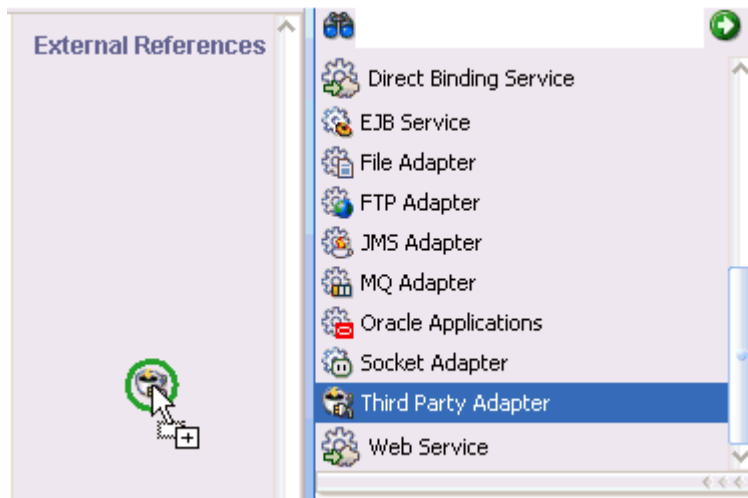
This section describes how to define a BPEL outbound process, which consists of the following stages:

1. Configuring a Third Party Adapter Service Component
2. Configuring an Outbound BPEL Process Component

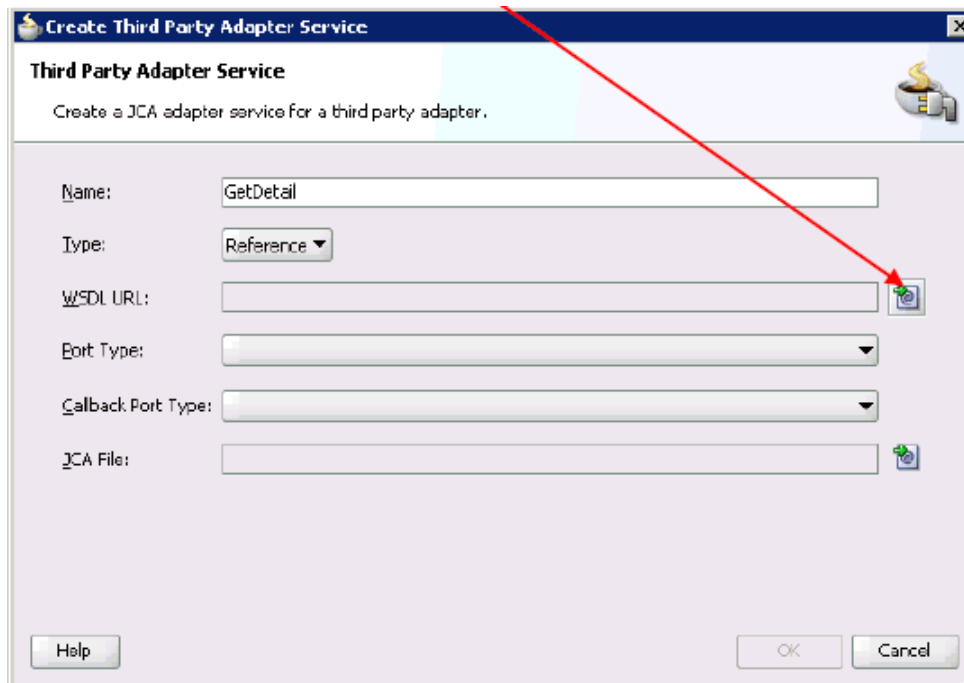
#### Configuring a Third Party Adapter Service Component

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.



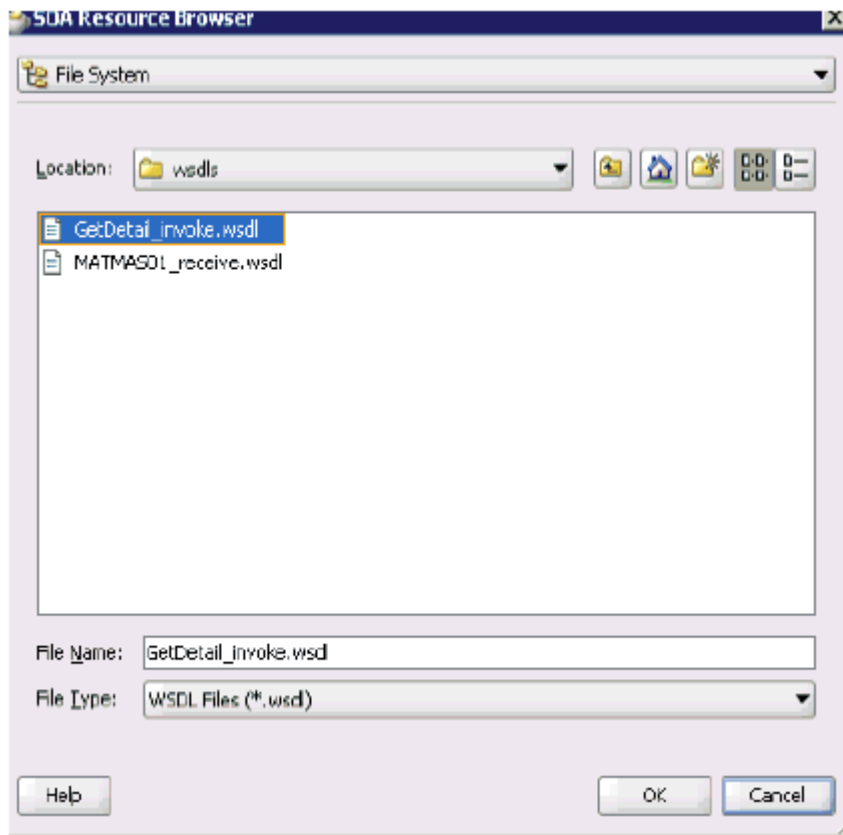
The Create Third Party Adapter Service dialog is displayed.



2. Enter a name for the third party adapter service.

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 SOA Resource Browser dialog is displayed.

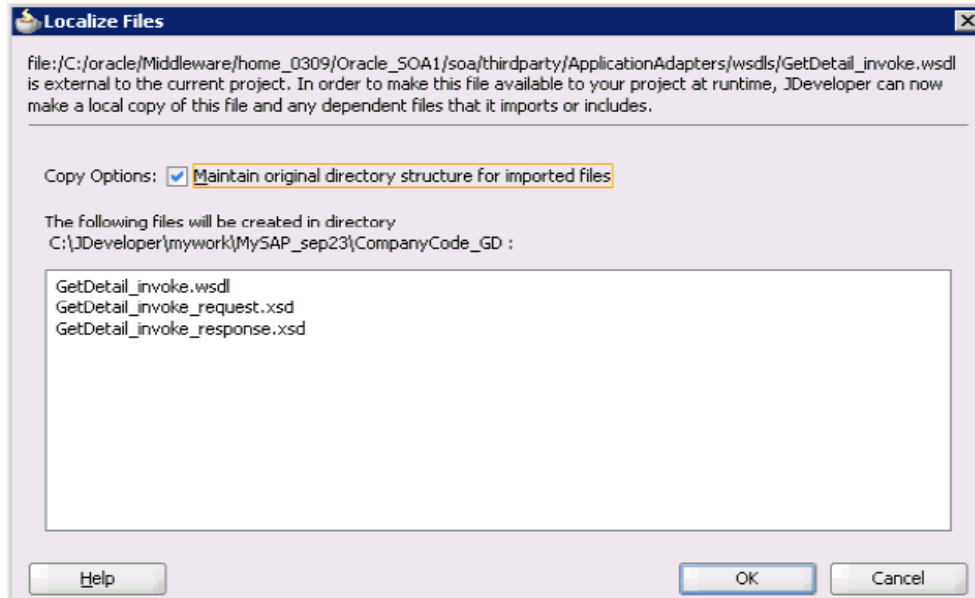


5. Browse and select an outbound WSDL file from the following directory:

```
C:\oracle\Middleware\home_GA\Oracle_
SOA1\soa\thirdparty\ApplicationAdapters\wsdlis
```

6. Click **OK**.

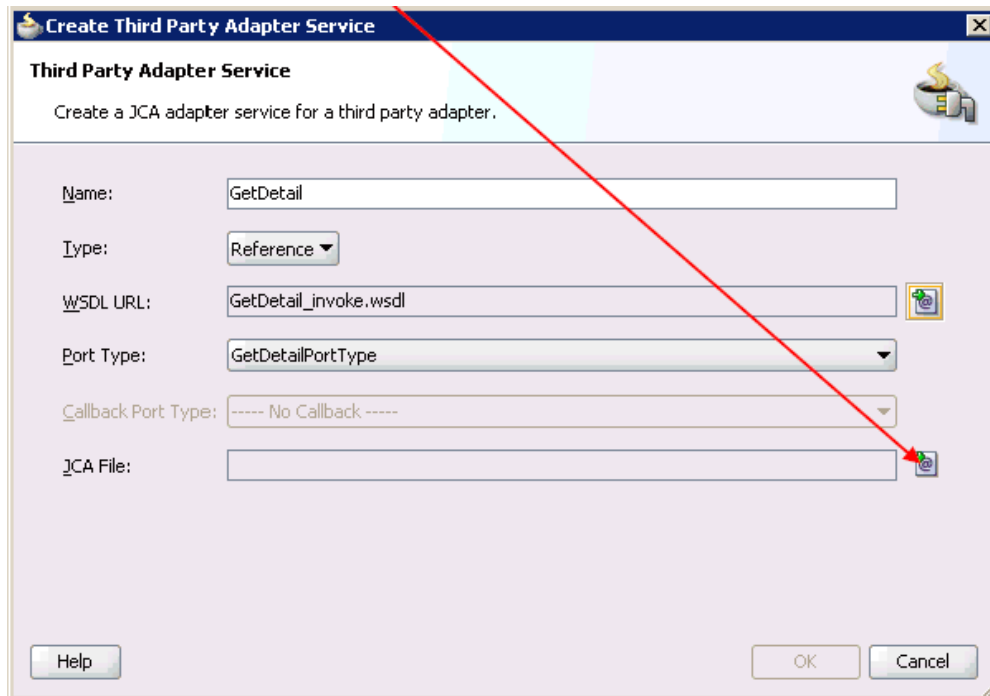
The Localize Files dialog is displayed.



7. Click **OK**.

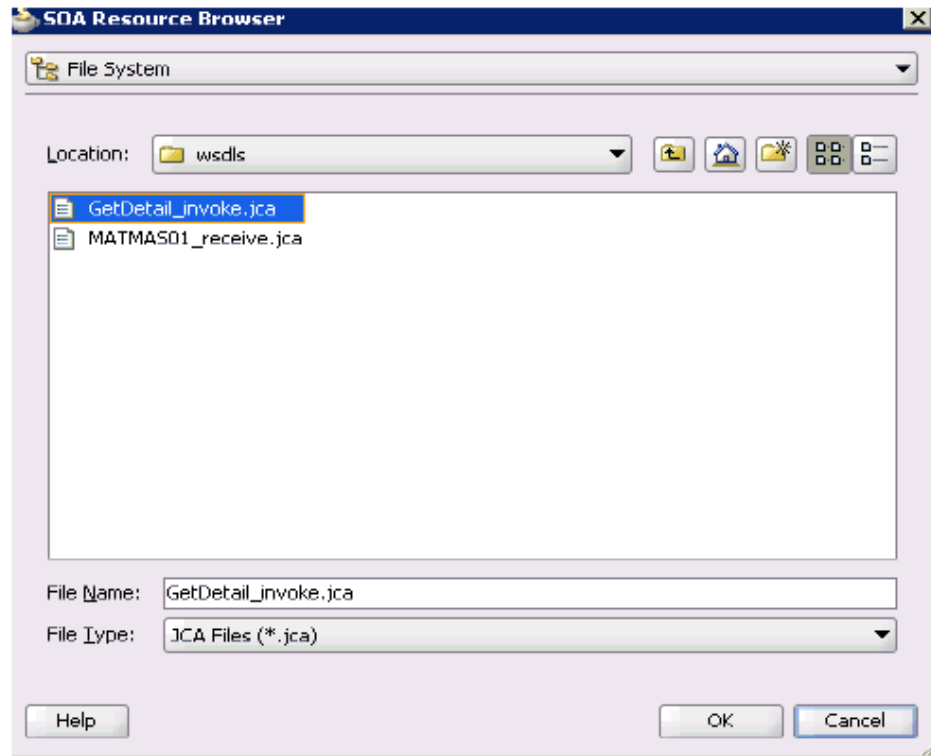
The outbound WSDL file and associated request and response XML schema files (.xsd) are imported to the project folder that has been created.

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



8. Click the **Find JCA file** icon, which is located to the right of the JCA File field. The SOA Resource Browser dialog is displayed.



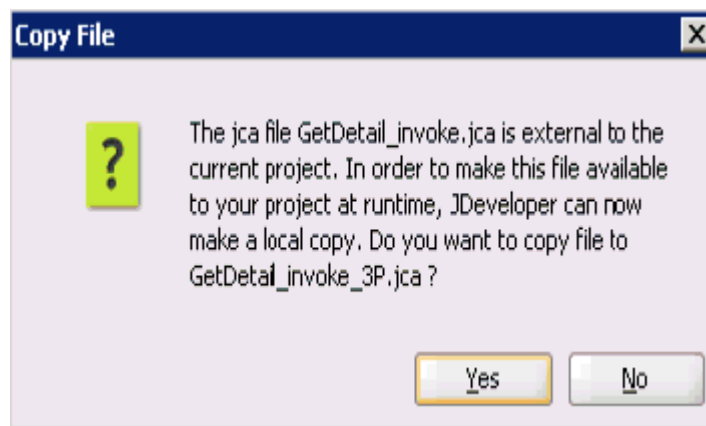


9. Browse and select the JCA properties file from the following directory:

C:\oracle\Middleware\home\_GA\Oracle\_  
SOA1\soa\thirdparty\ApplicationAdapters\wsdls

10. Click **OK**.

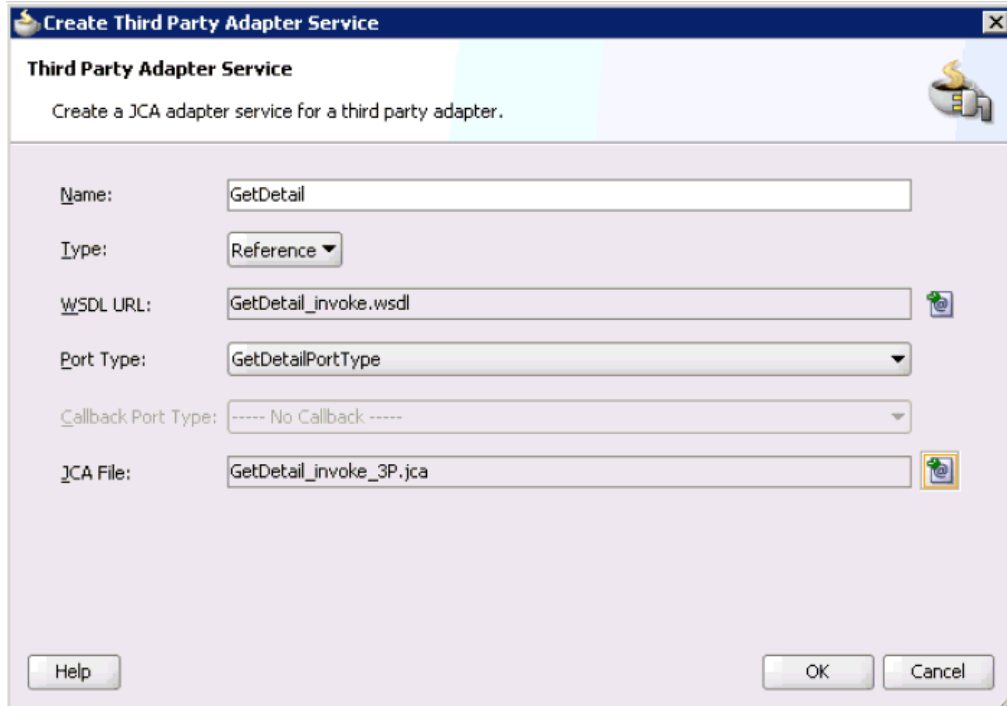
The following message is displayed.



11. Click **Yes**.

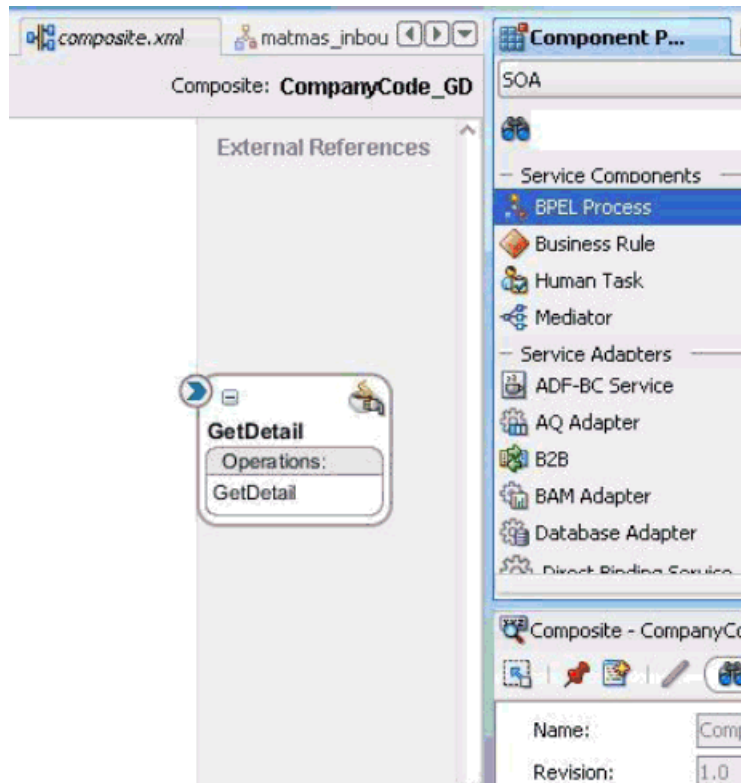
A copy of the JCA properties file is made in the project folder.

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



12. Click **OK**.

The third party adapter service component (GetDetail) is created in the External References pane, as shown in the following image.

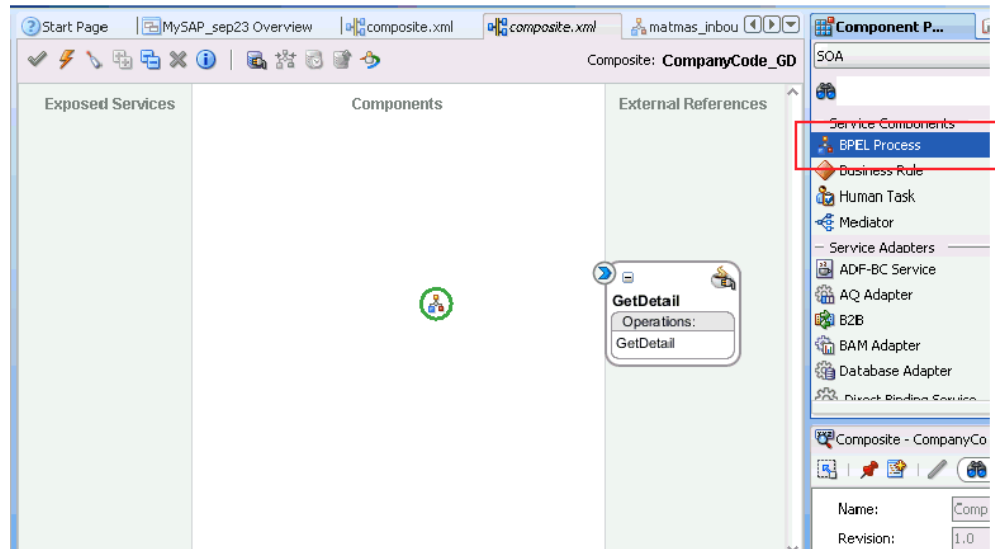


You are now ready to configure an outbound BPEL process component.

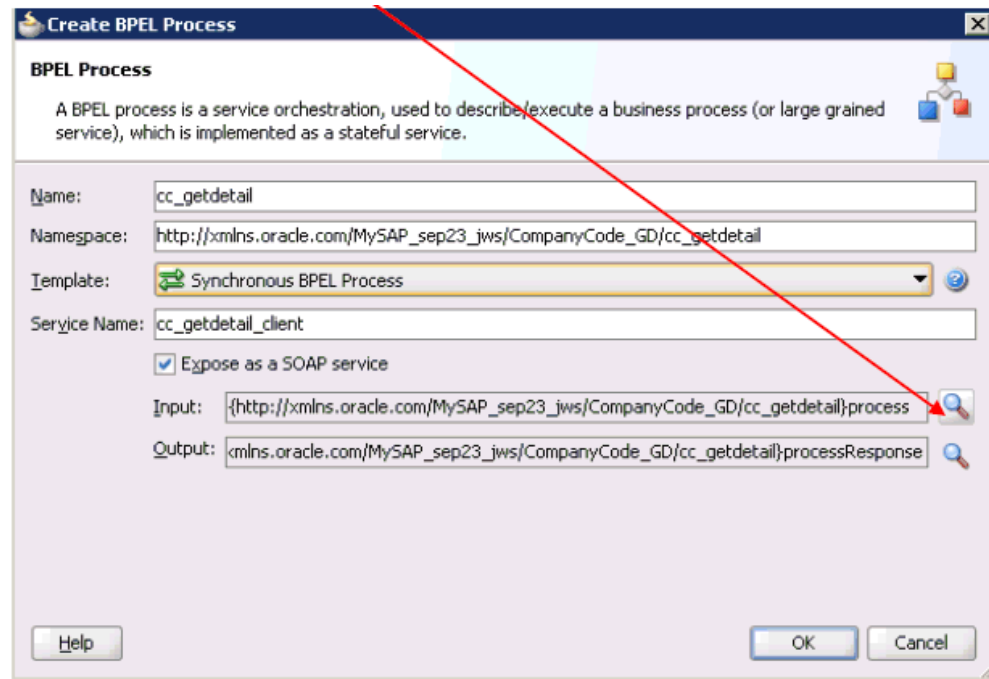
## Configuring an Outbound BPEL Process Component

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

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

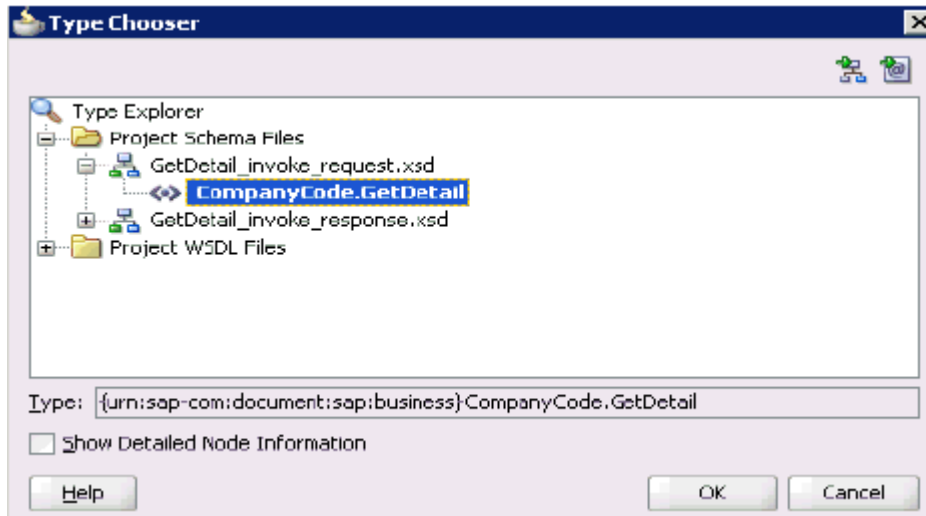


The Create BPEL Process dialog is displayed.



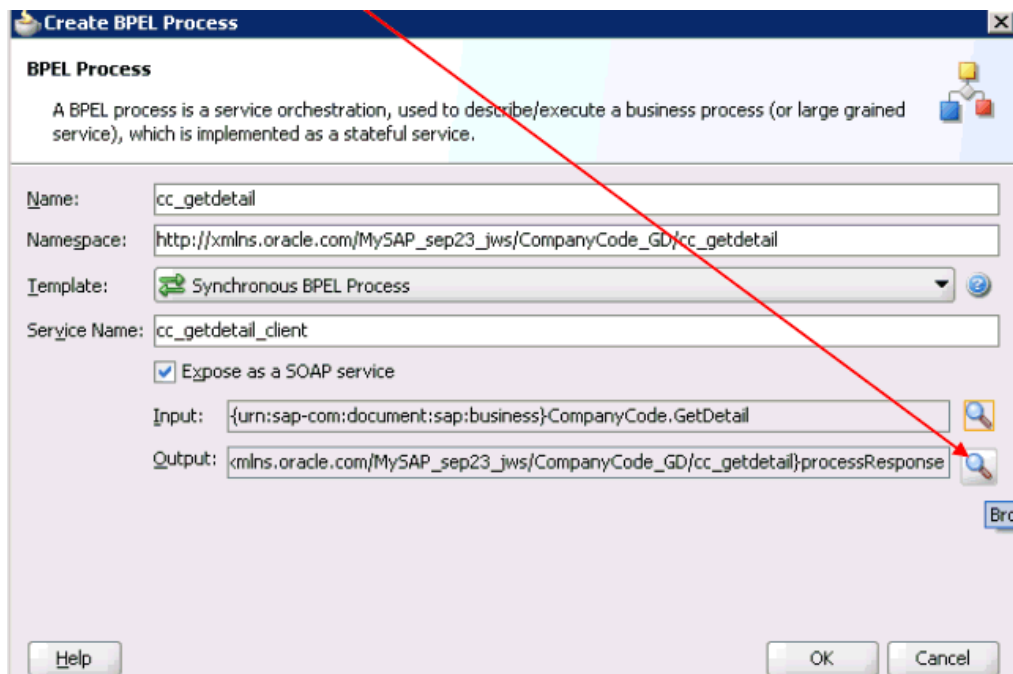
2. In the Name field, enter a name to identify the new outbound BPEL process component (for example, cc\_getdetail).
3. From the Template list, select **Synchronous BPEL Process**.
4. Click the **Browse** icon, which is located to the right of the Input field to select the associated XML request schema file.

The Type Chooser dialog is displayed.

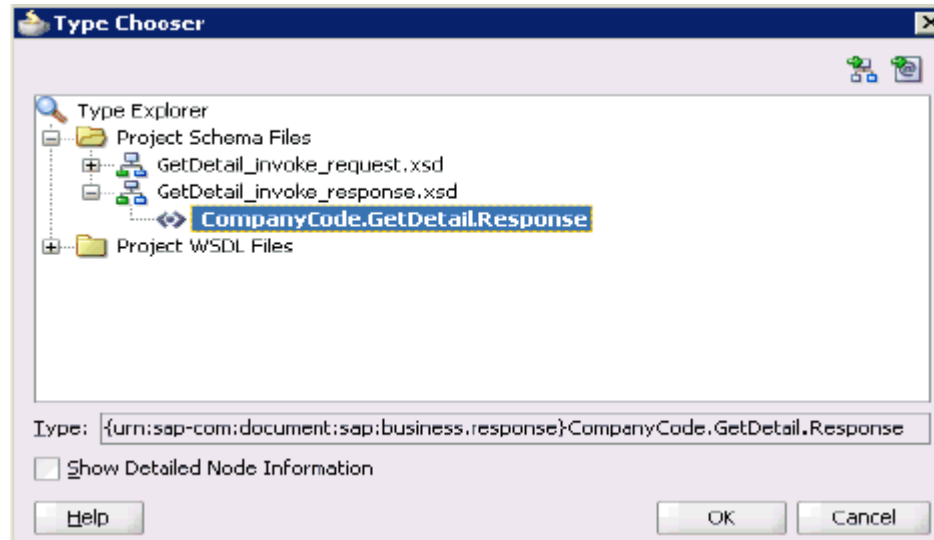


5. Expand **Project Schema Files**, **GetDetail\_invoke\_request.xsd**, and select **CompanyCode.GetDetail**.
6. Click **OK**.

You are returned to the Create BPEL Process dialog.

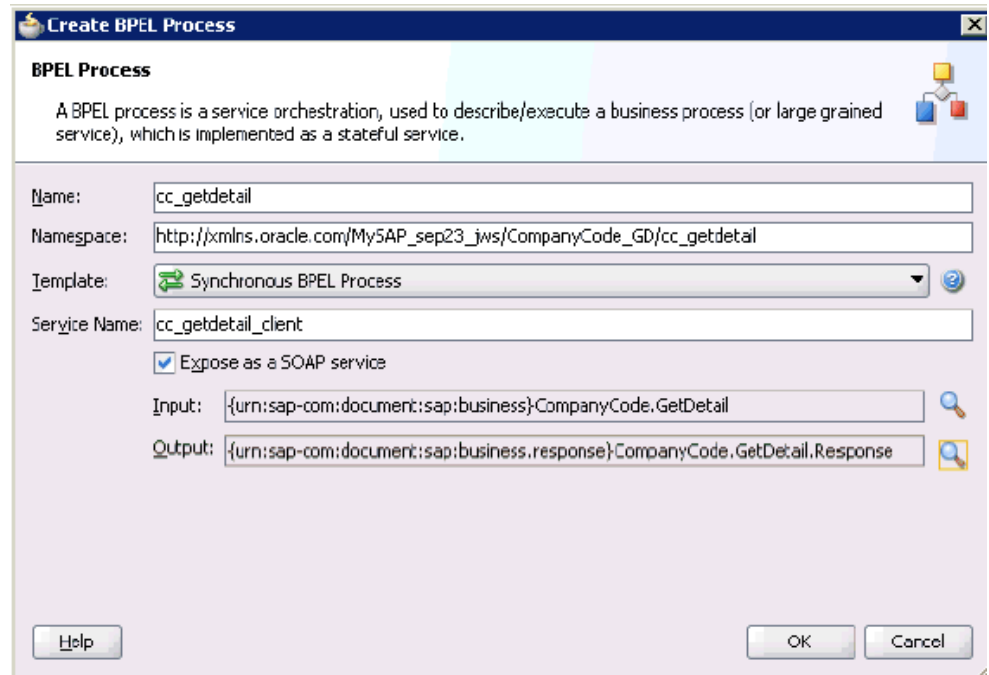


7. Click the **Browse** icon, which is located to the right of the Output field to select the associated XML response schema file.  
The Type Chooser dialog is displayed.

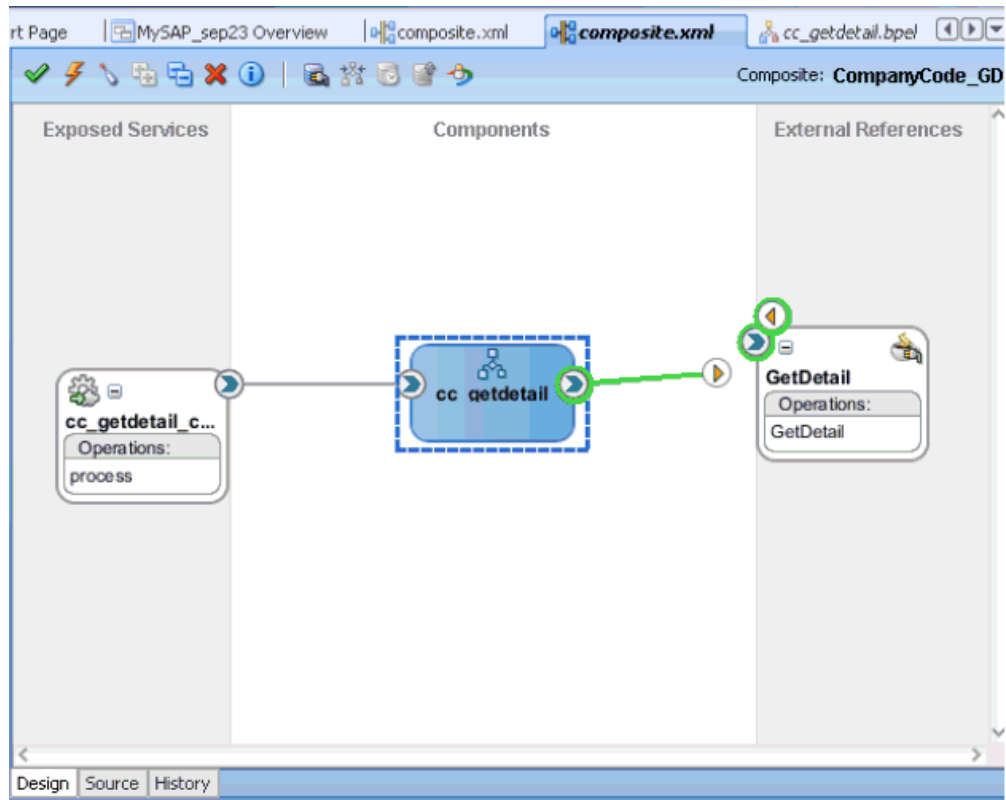


8. Expand **Project Schema Files**, **GetDetail\_invoke\_response.xsd**, and select **CompanyCode.GetDetail.Response**.
9. Click **OK**.

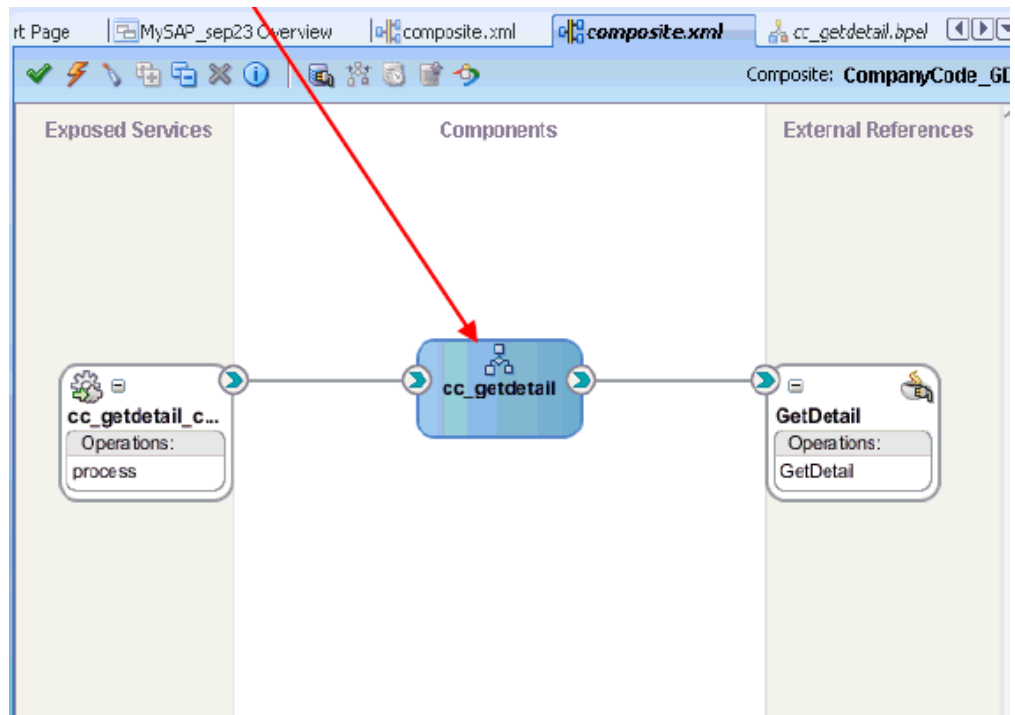
You are returned to the Create BPEL Process dialog.



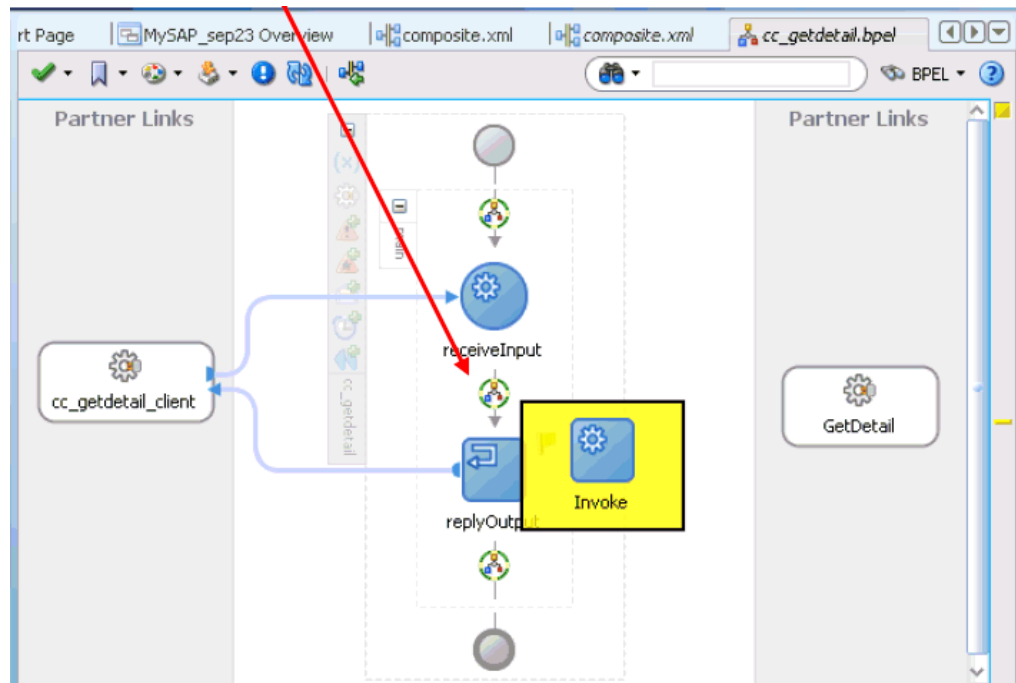
10. Click **OK**.
11. Create a connection between the outbound BPEL process component (cc\_getdetail) and the third party adapter service component (GetDetail).



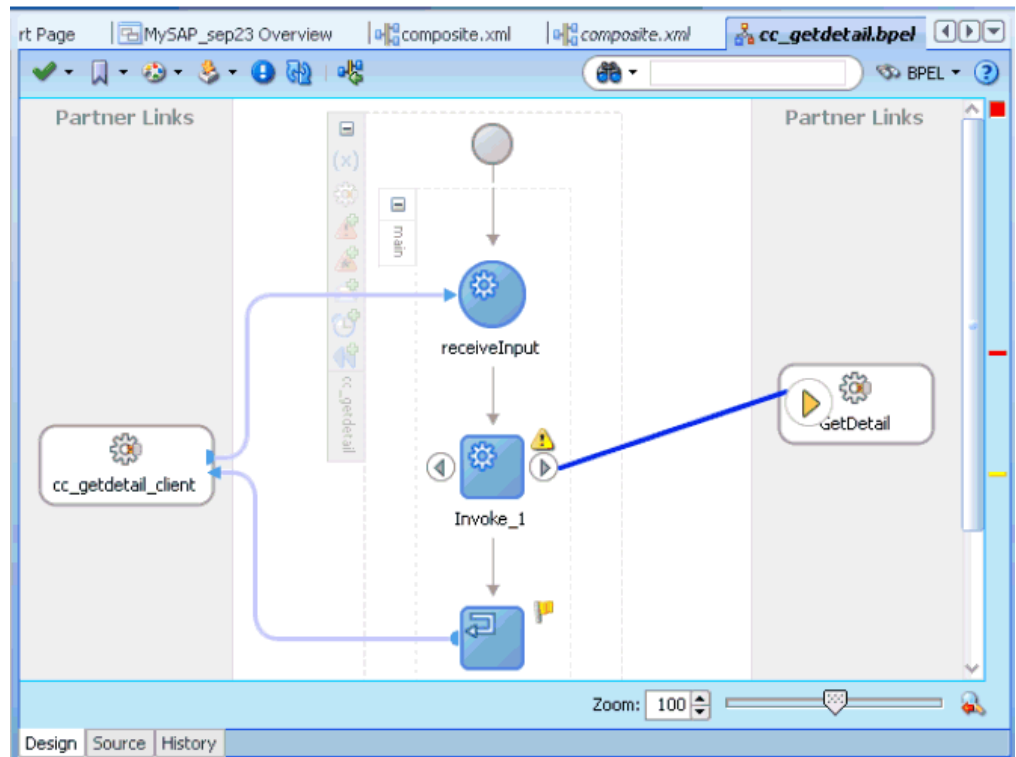
12. Double-click the outbound BPEL process component (cc\_getdetail) in the Components pane.



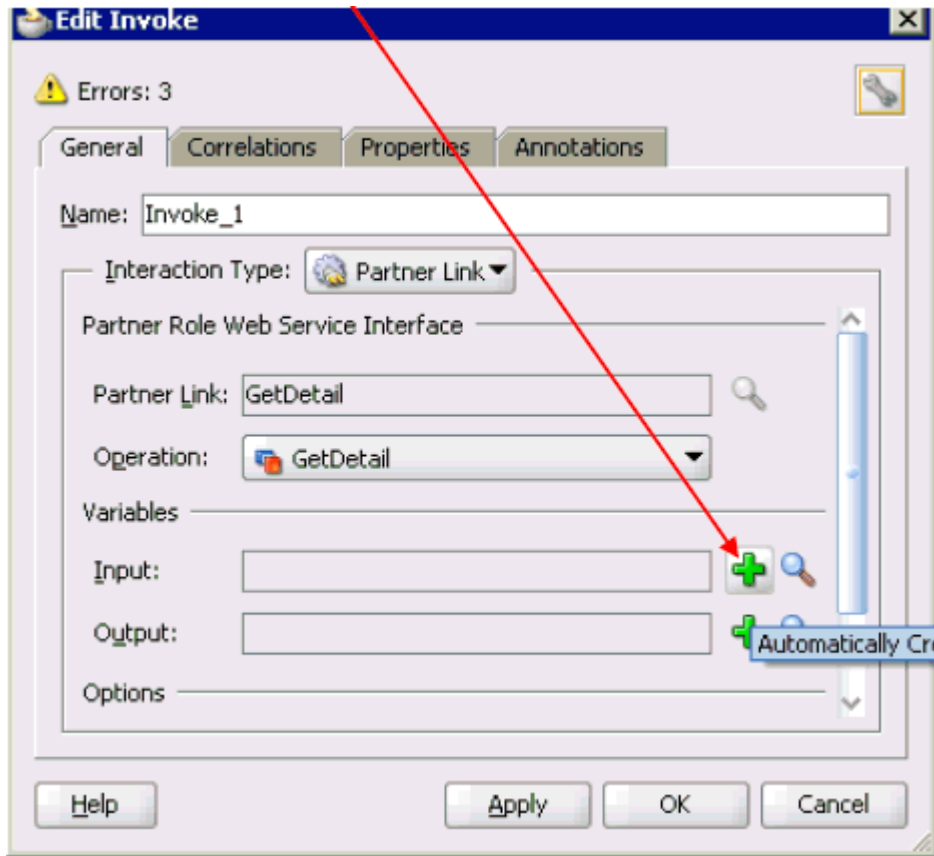
13. Drag and drop the **Invoke** activity component to the Components pane and place it between the **receiveInput** activity component and the **replyOutput** activity component.



14. Create a connection between the new Invoke activity component (Invoke\_1) and the third party adapter service component (GetDetail).

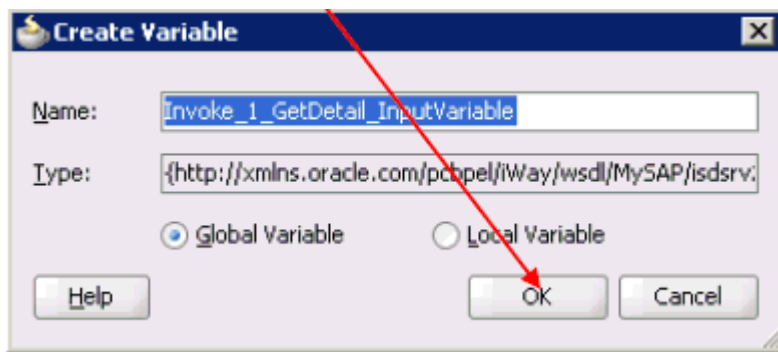


15. Double-click the new Invoke activity component (**Invoke\_1**).  
The Edit Invoke dialog is displayed.



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

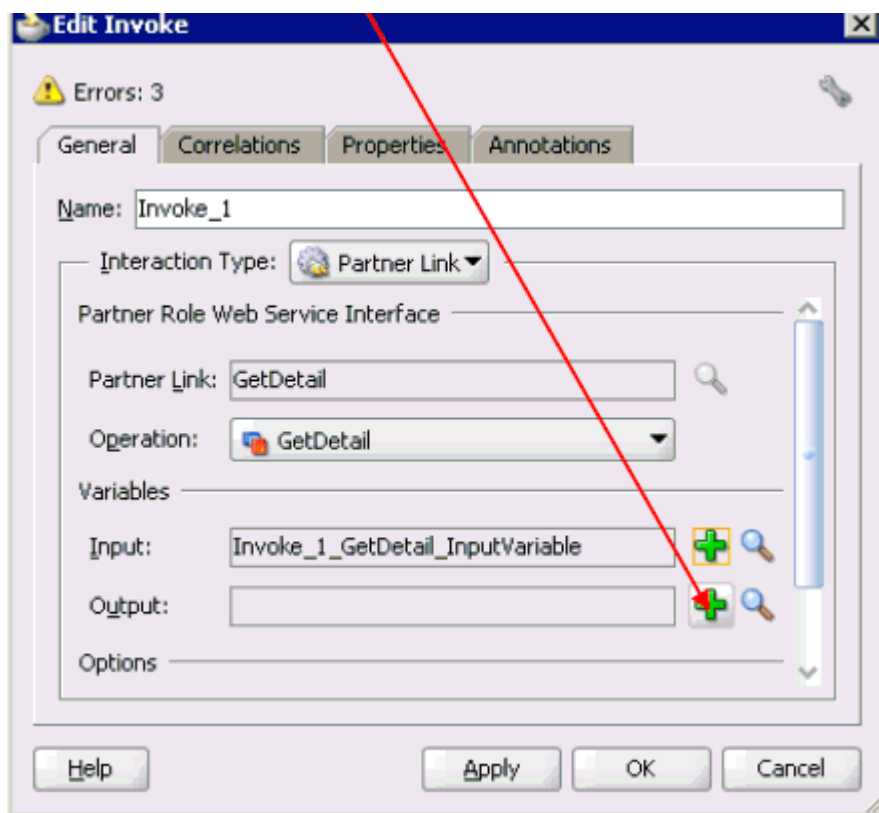
The Create Variable dialog is displayed.



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

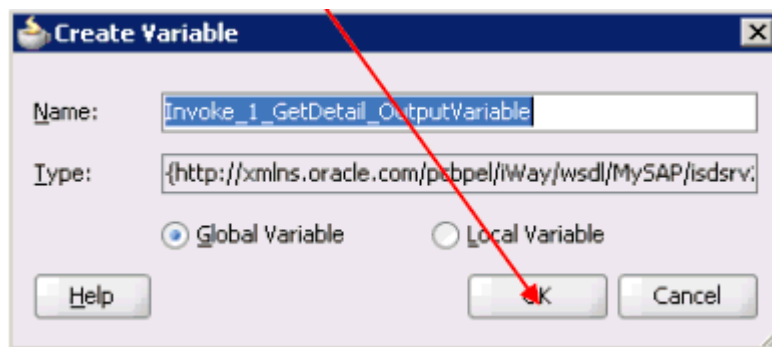
You are returned to the Edit Invoke dialog.





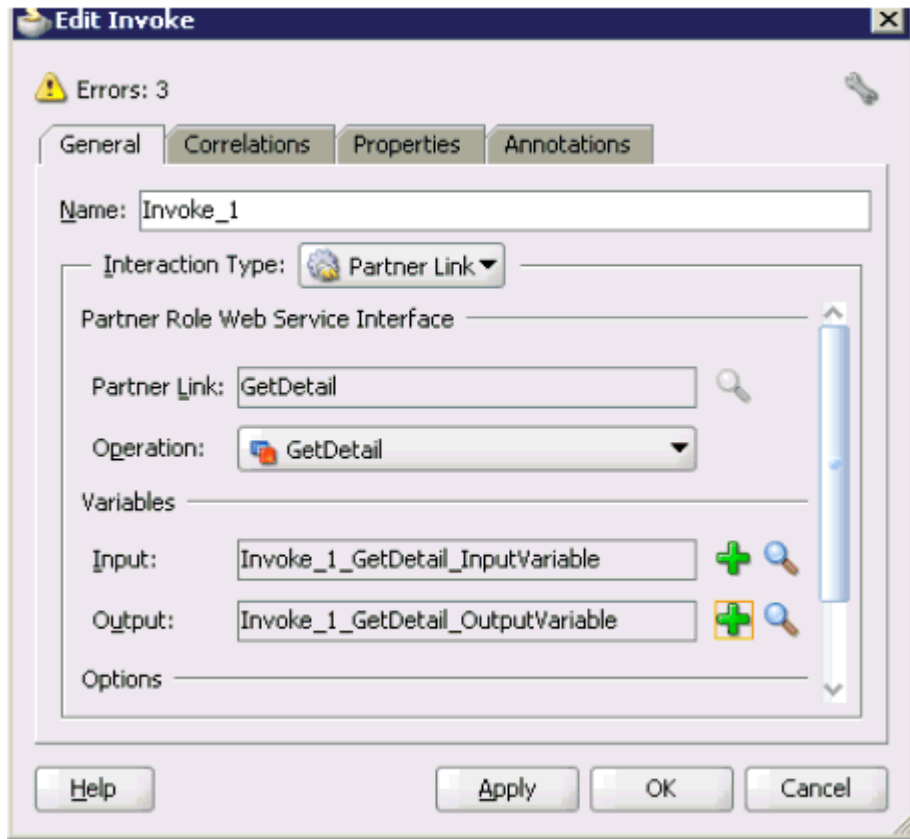
18. Click the **Plus sign** icon, which is located to the right of the Output field to configure a new output variable.

The Create Variable dialog is displayed.

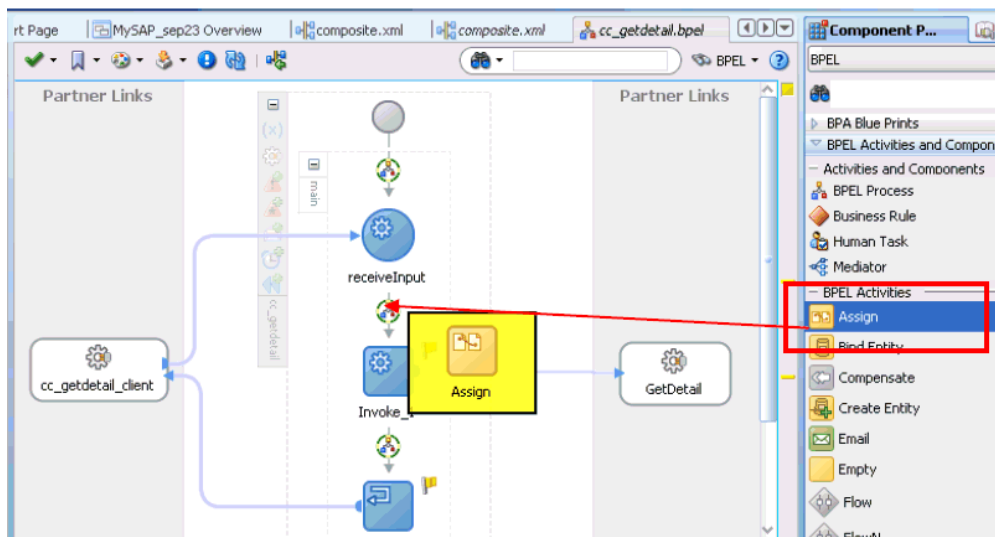


19. Accept the default values that are provided for the new output variable and click **OK**.

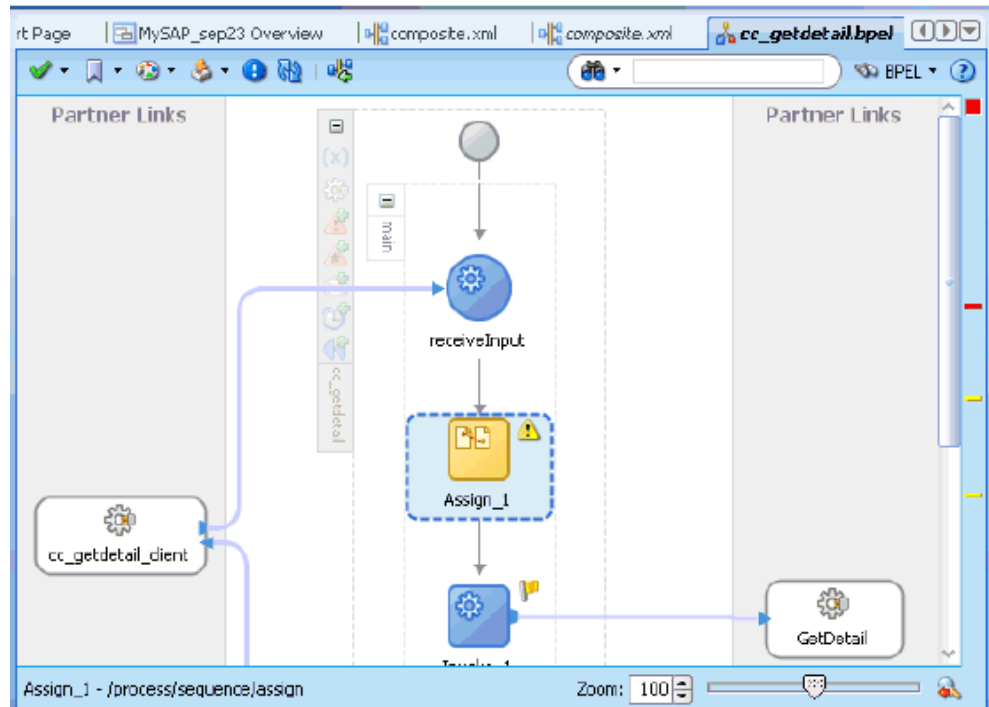
You are returned to the Edit Invoke dialog.



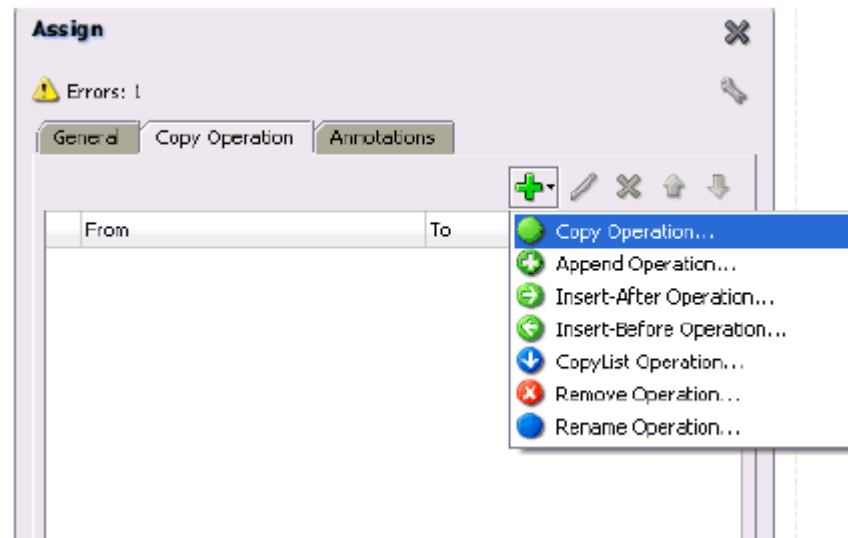
20. Click **Apply** and then **OK**.
21. Drag and drop the **Assign** activity component to the Components pane and place it between the Receive activity component (receiveInput) and the Invoke activity component (Invoke\_1).



22. Double-click the new Assign activity component (**Assign\_1**).

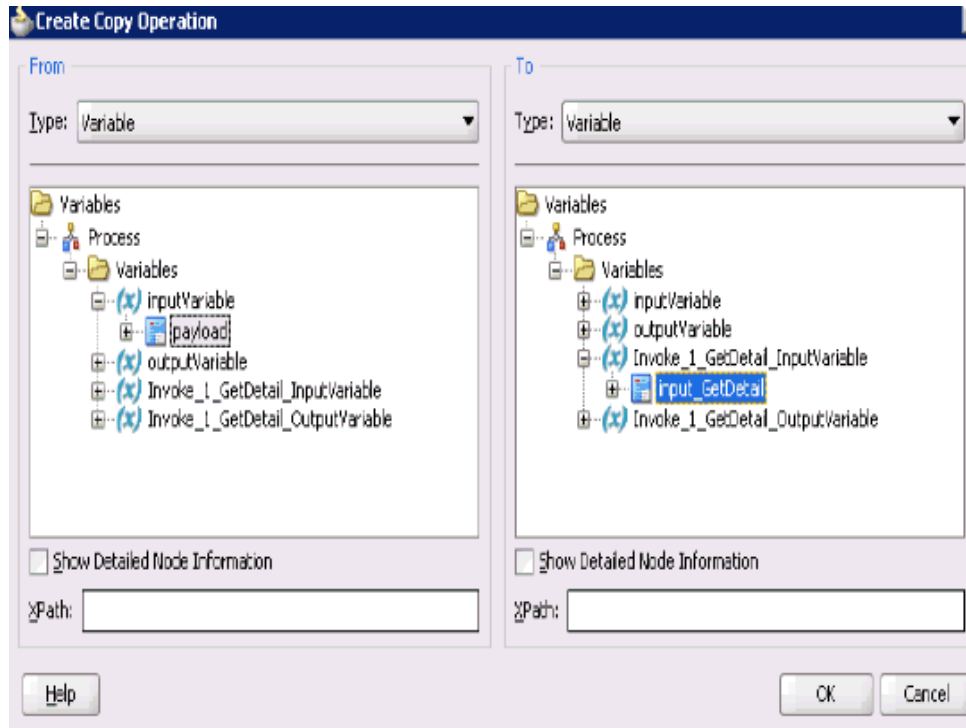


The Assign dialog is displayed.



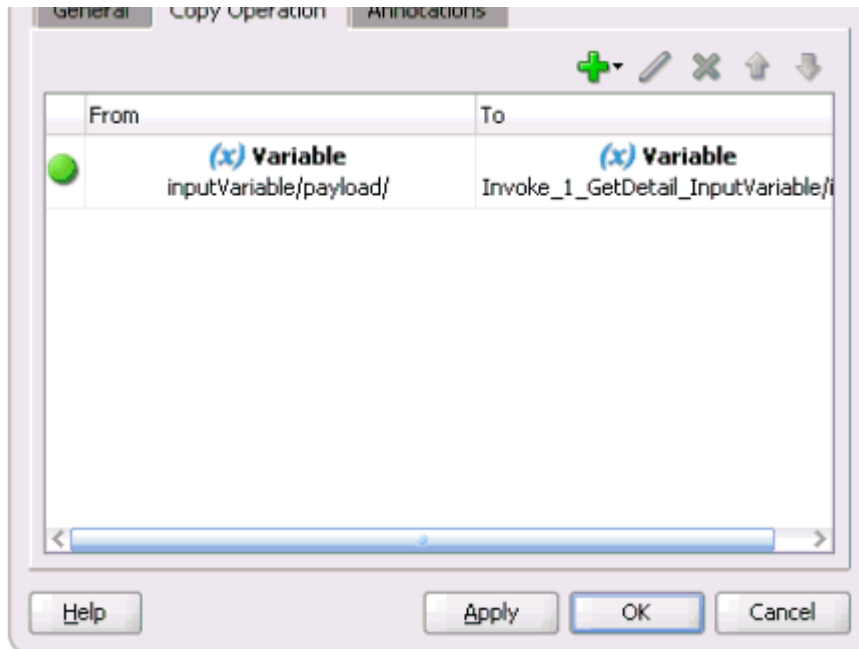
23. Click the **Copy Operation** tab.
24. Click the **Plus sign** icon and select **Copy Operation** from the list of available operations.

The Create Copy Operation dialog is displayed.



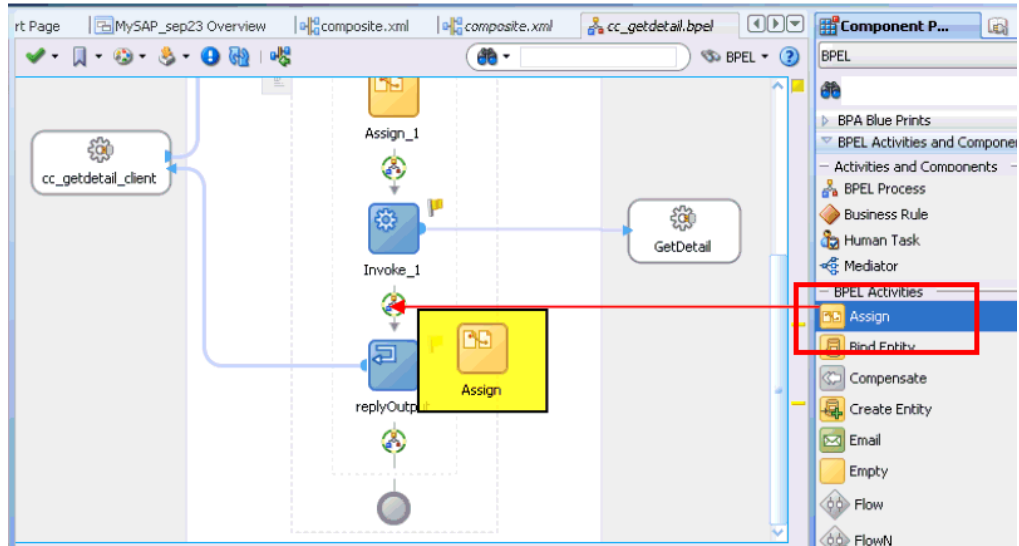
25. In the From pane, expand **Variables**, **InputVariable**, and then select **payload**.
26. In the To pane, expand **Variables**, **Invoke\_1\_GetDetail\_InputVariable**, and then select **input\_GetDetail**.
27. Click **OK**.

You are returned to the Assign dialog.

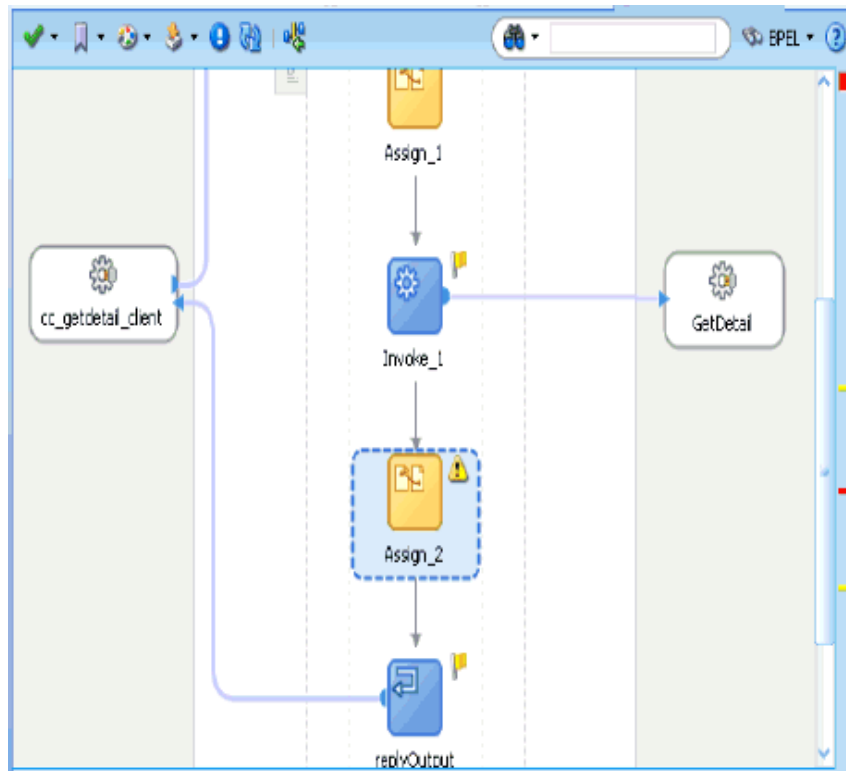


28. Click **Apply** and then **OK**.

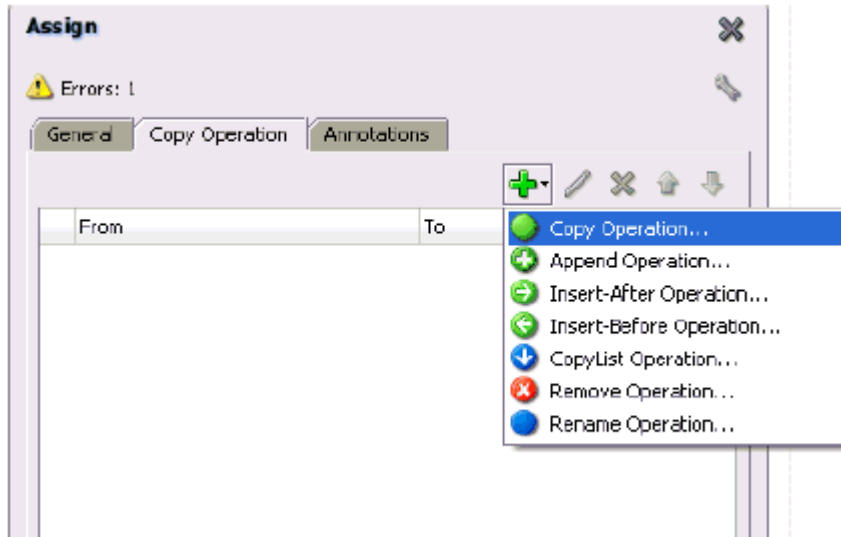
29. Drag and drop the **Assign** activity component to the Components pane and place it between the Invoke activity (Invoke\_1) and the Reply activity (replyOutput).



30. Double-click the new Assign activity component (**Assign\_2**).

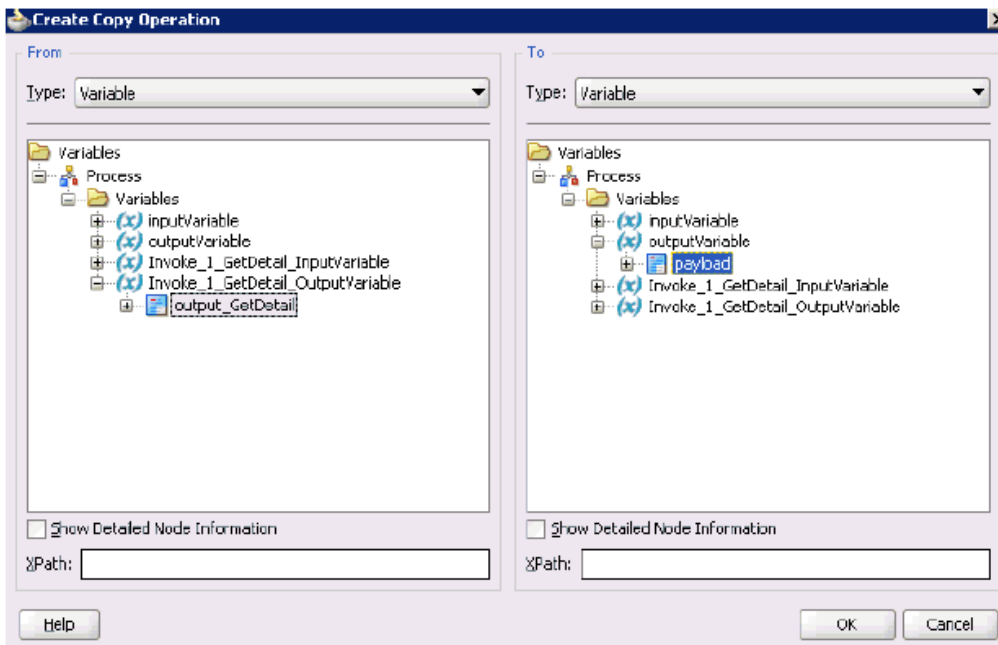


The Assign dialog is displayed.



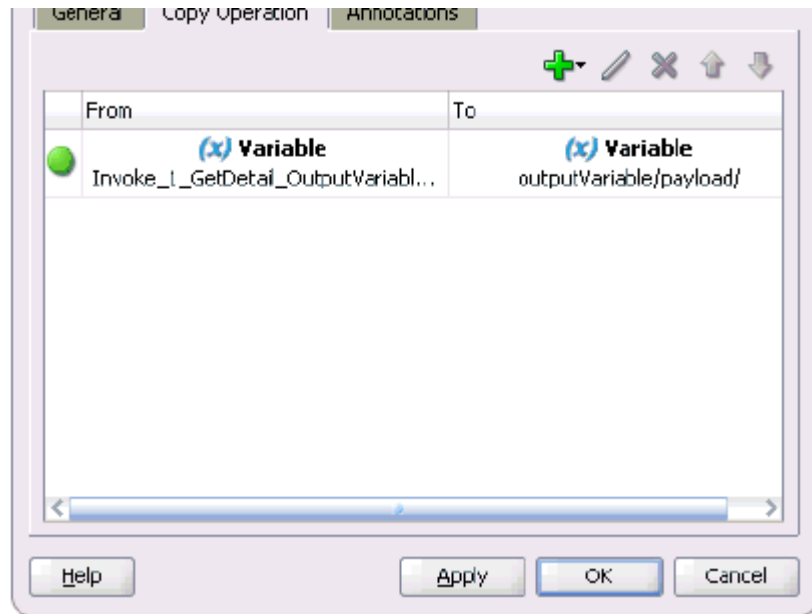
31. Click the **Copy Operation** tab.
32. Click the **Plus sign** icon and select **Copy Operation** from the list of available operations.

The Create Copy Operation dialog is displayed.

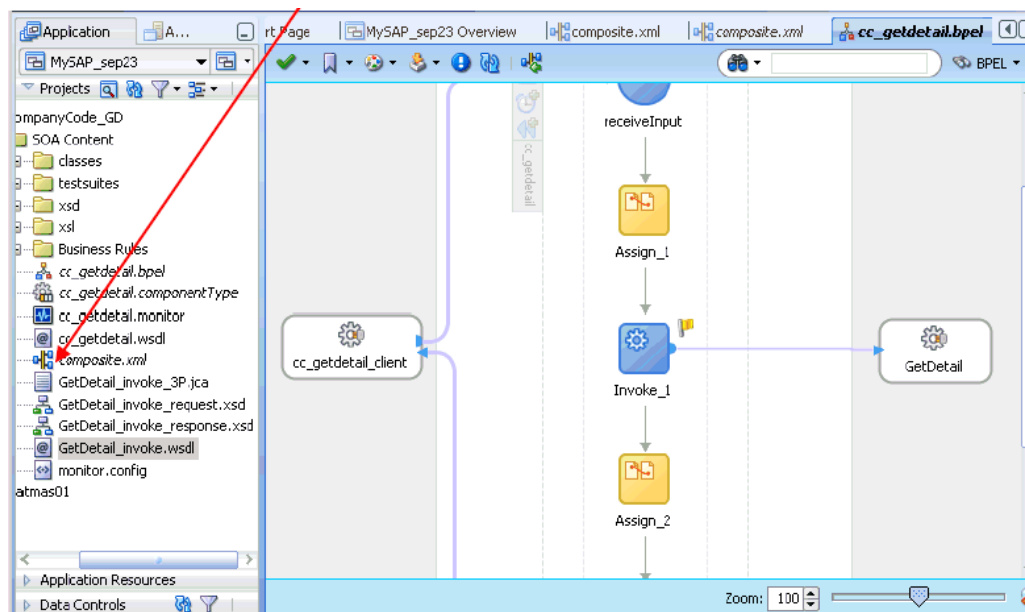


33. In the From pane, expand **Variables**, **Invoke\_1\_GetDetail\_OutputVariable**, and then select **output\_GetDetail**.
34. In the To pane, expand **Variables**, **outputVariable**, and then select **payload**.
35. Click **OK**.

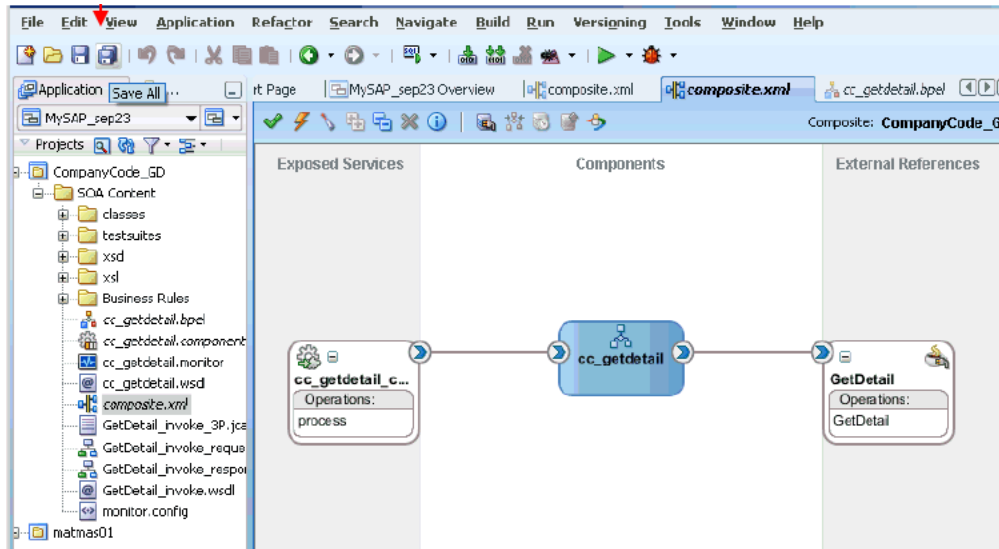
You are returned to the Assign dialog.



36. Click Apply and then OK.



37. Double-click **composite.xml** in the left pane.



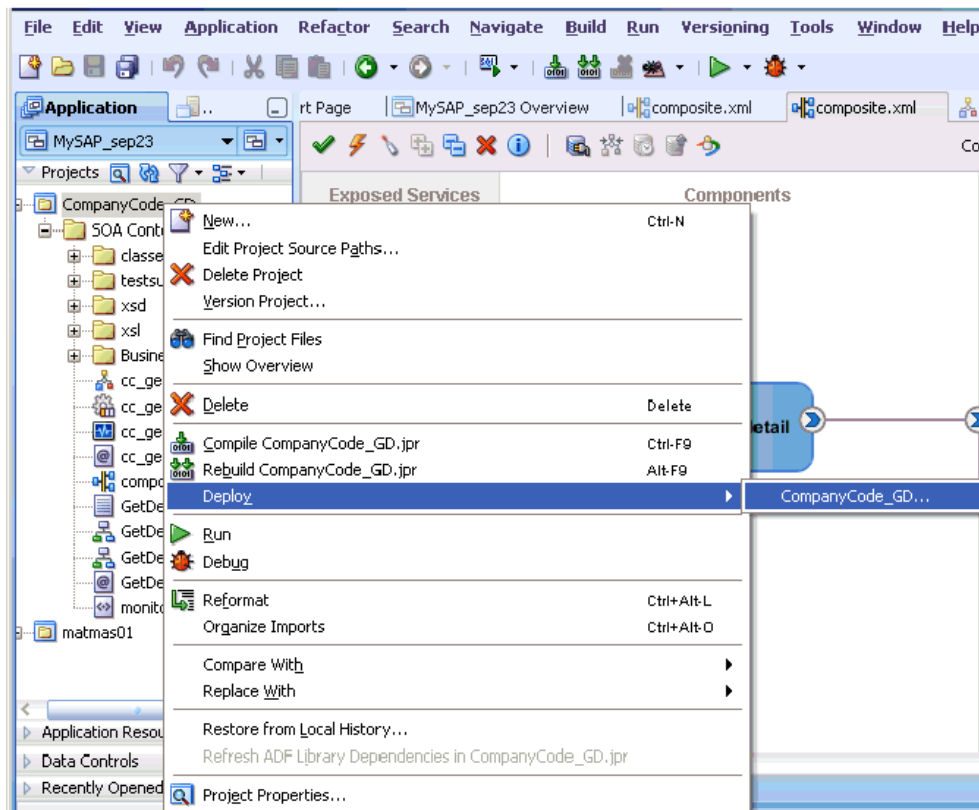
38. Click the **Save All** icon in the menu bar to save the new outbound BPEL process component that was configured.

You are now ready to deploy the BPEL outbound process.

### 5.4.4 Deploying the BPEL Outbound Process

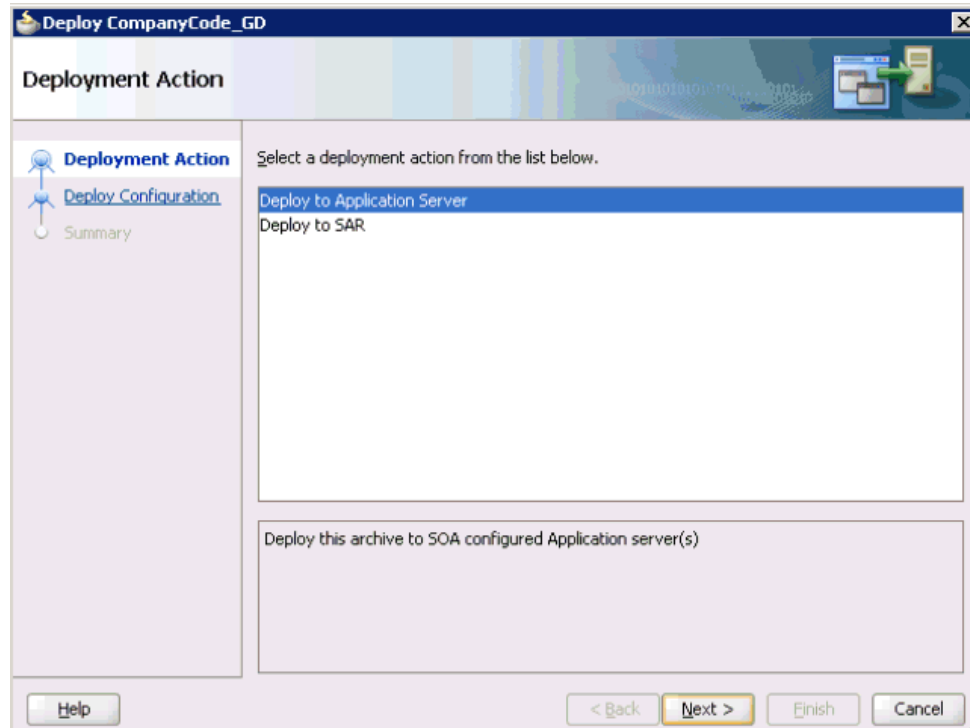
Perform the following steps to deploy the BPEL outbound process.

1. Right-click the project name in the left pane (for example, **CompanyCode\_GD**), select **Deploy**, and then click **CompanyCode\_GD**.



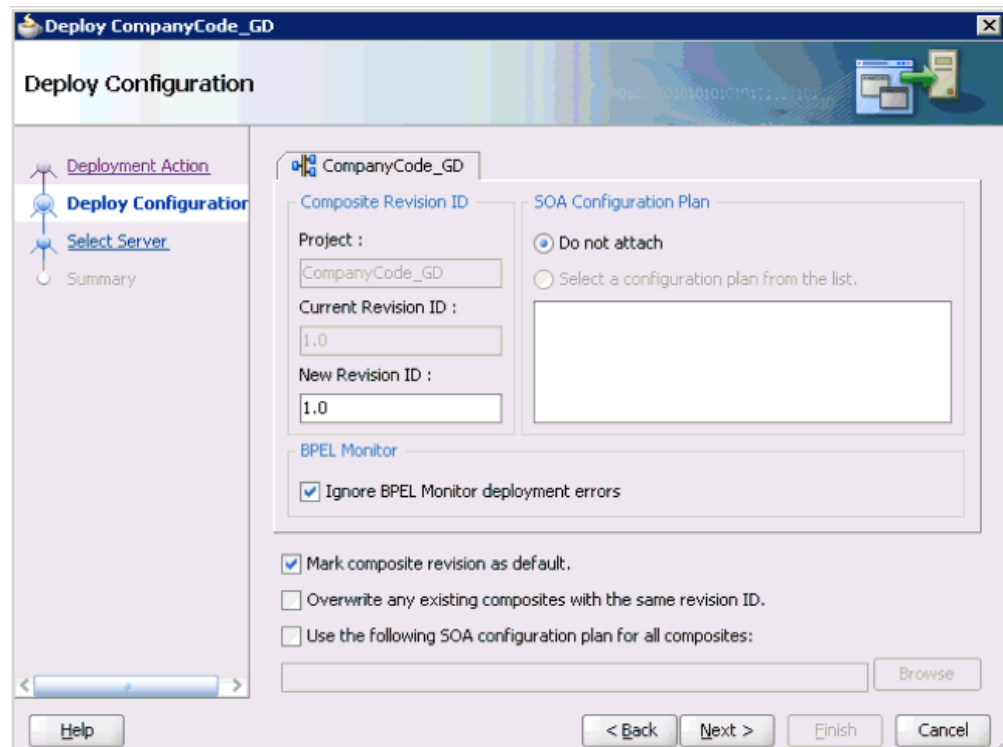


The Deployment Action page is displayed.



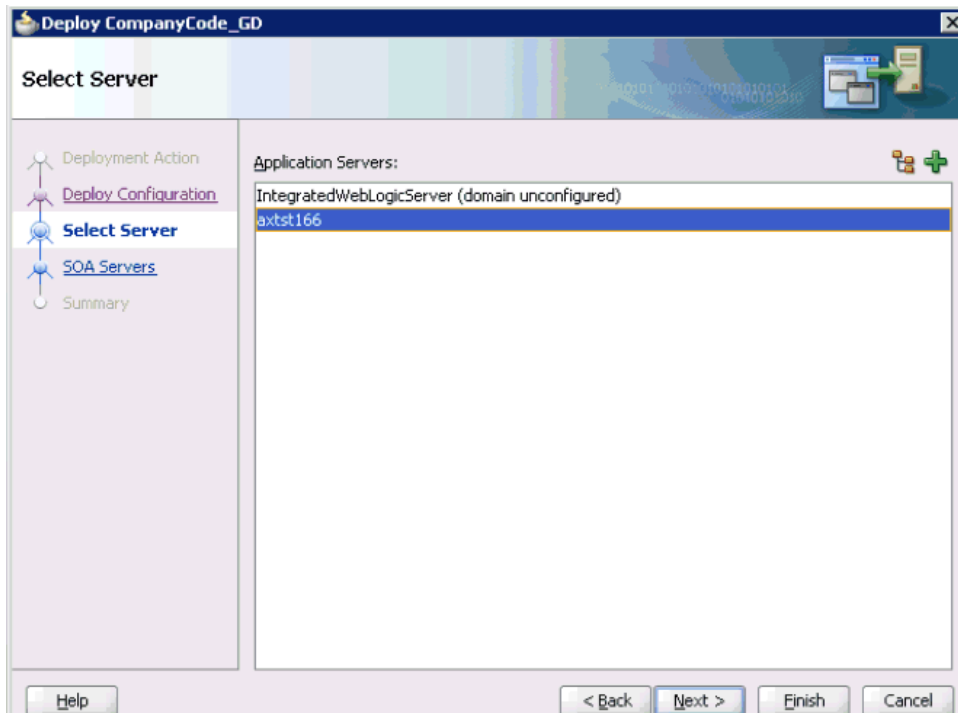
2. Ensure that **Deploy to Application Server** is selected.
3. Click **Next**.

The Deploy Configuration page is displayed.



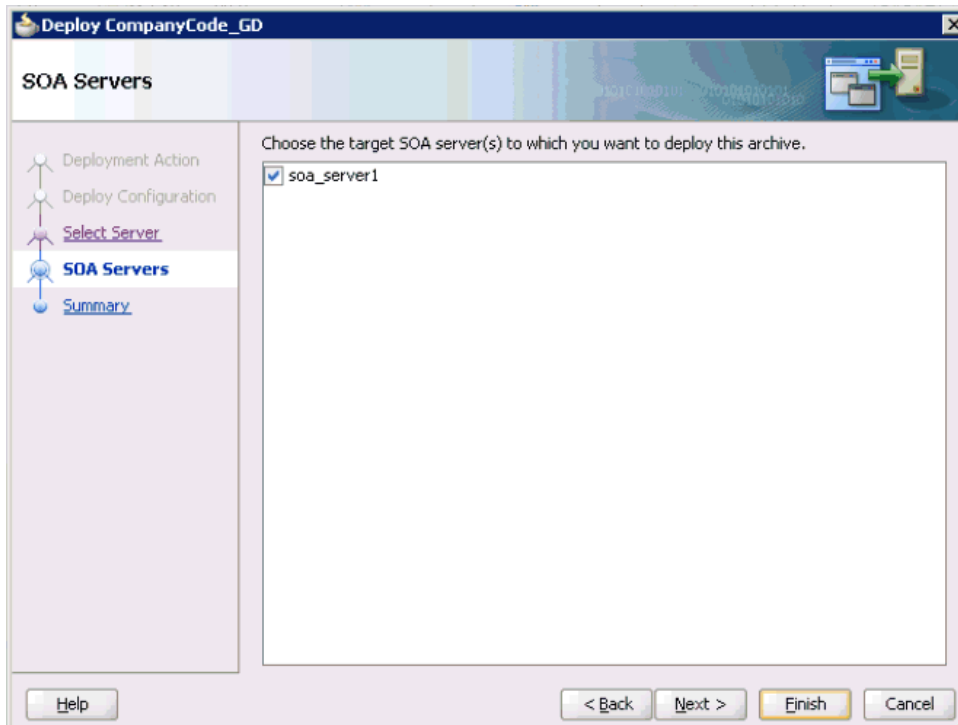
4. Leave the default values selected and click **Next**.

The Select Server page is displayed.



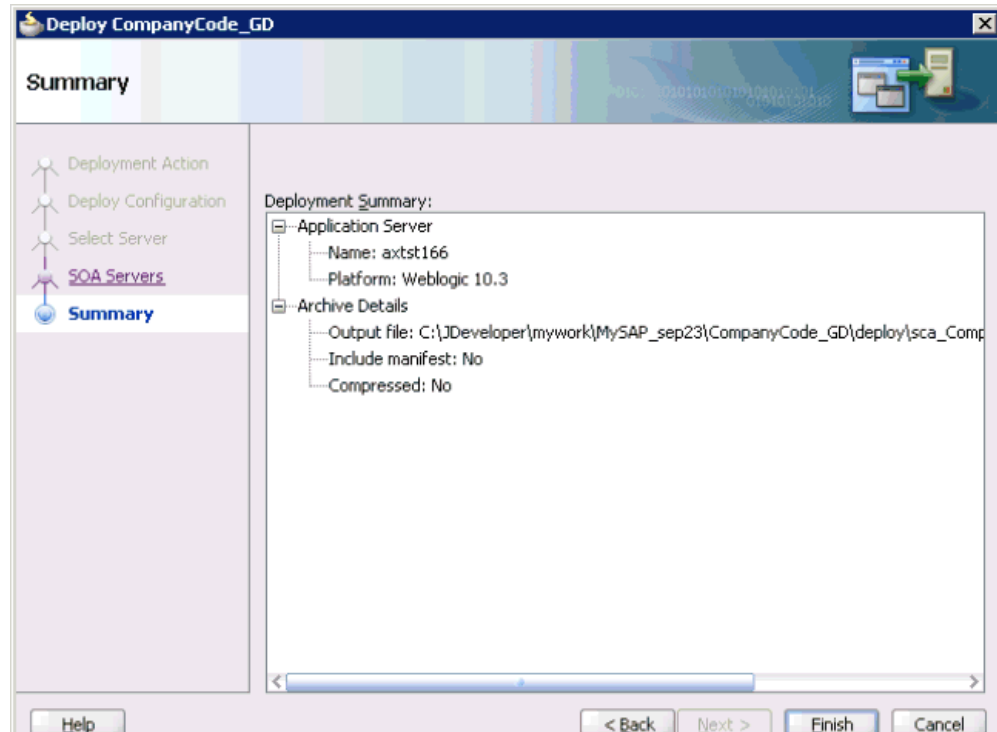
5. Select an available application server that was configured and click **Next**.

The SOA Servers page is displayed.



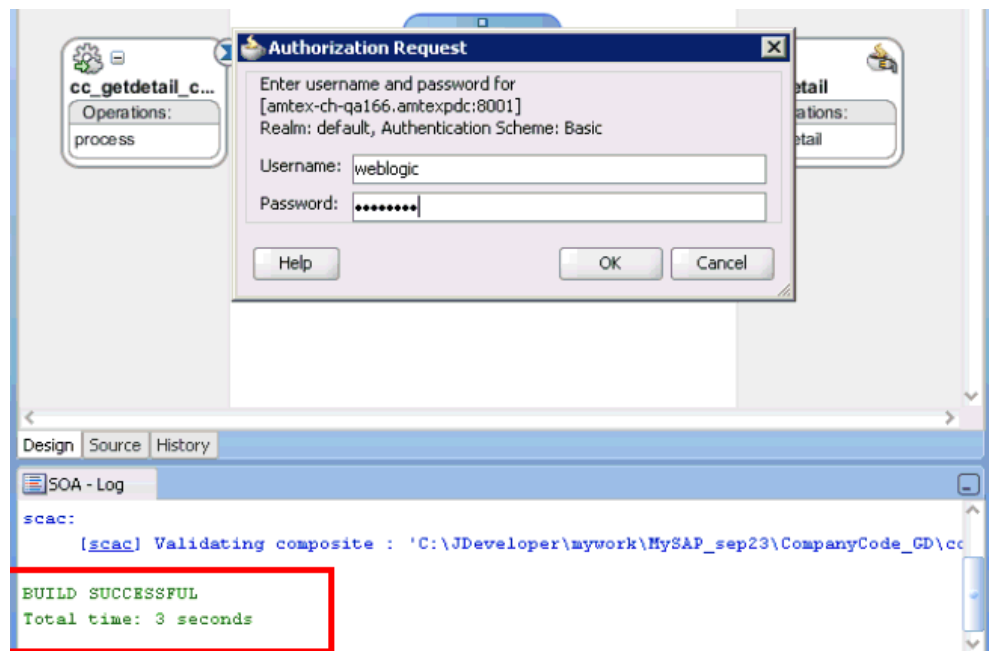
6. Select a target SOA server and click **Next**.

The Summary page is displayed.



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

The process is deployed successfully.

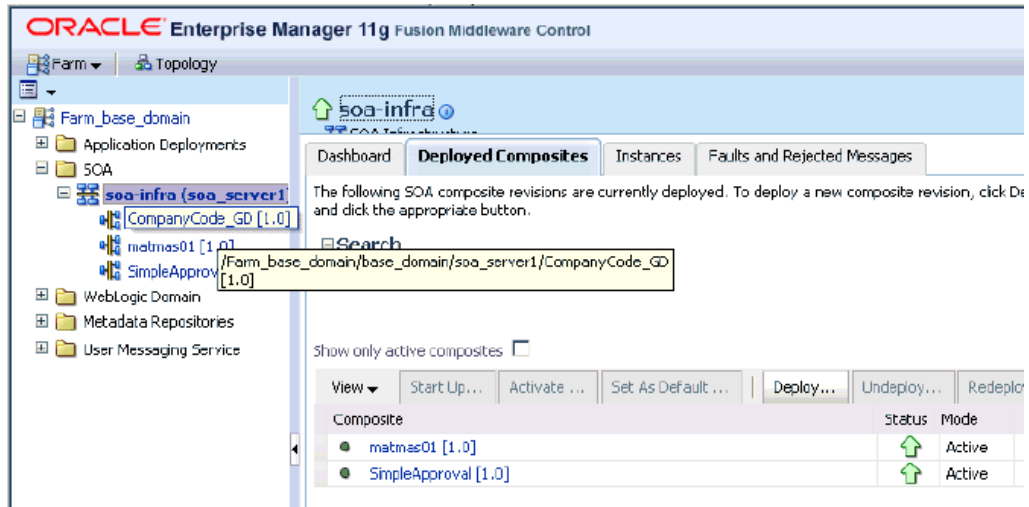


If an Authorization Request dialog is displayed during the deployment process, provide the required user name and password and click **OK**.

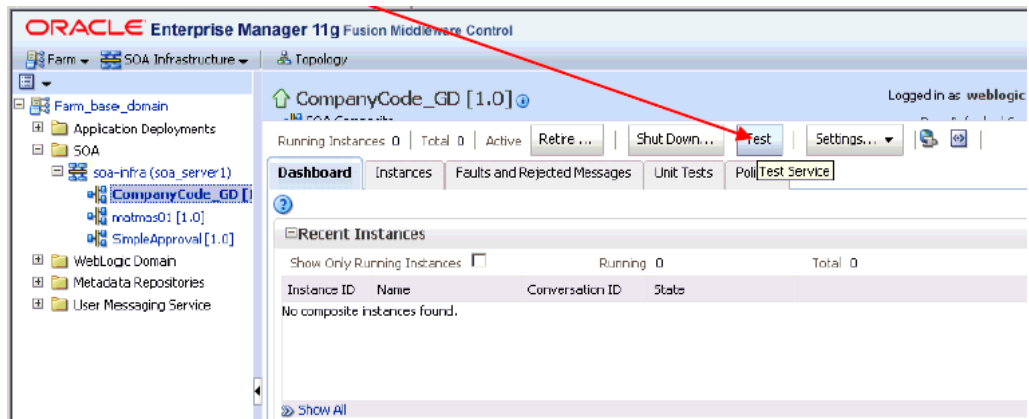
### 5.4.5 Invoking the Input XML Document in the Oracle Enterprise Manager Console

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

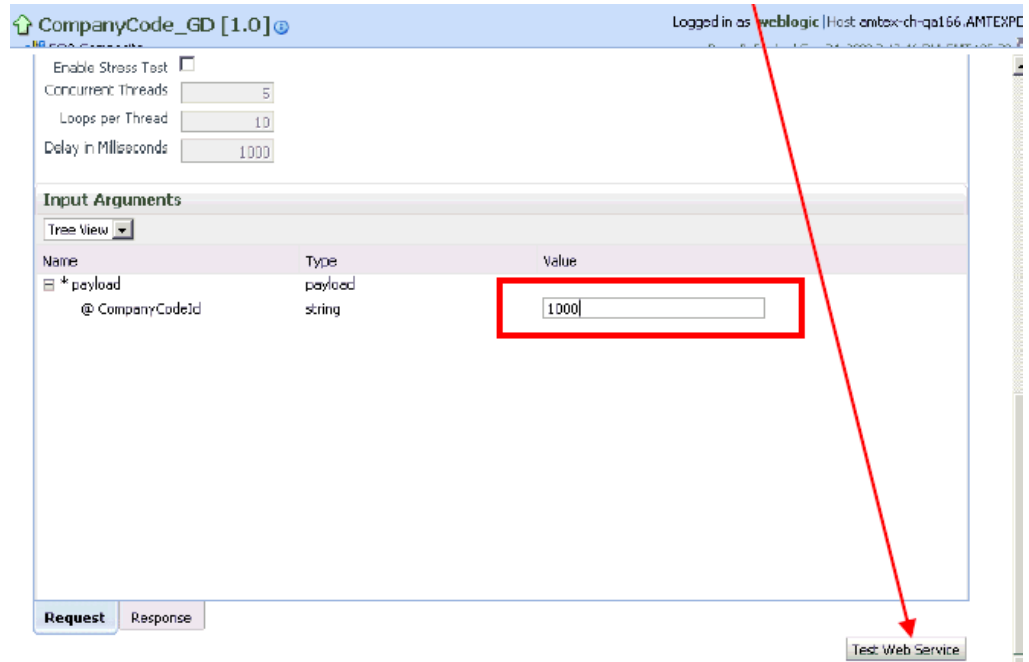
1. Logon to the Oracle Enterprise Manager console.



2. Expand your domain in the left pane followed by the **SOA** folder.
3. Select an available project (for example, CompanyCode\_GD).

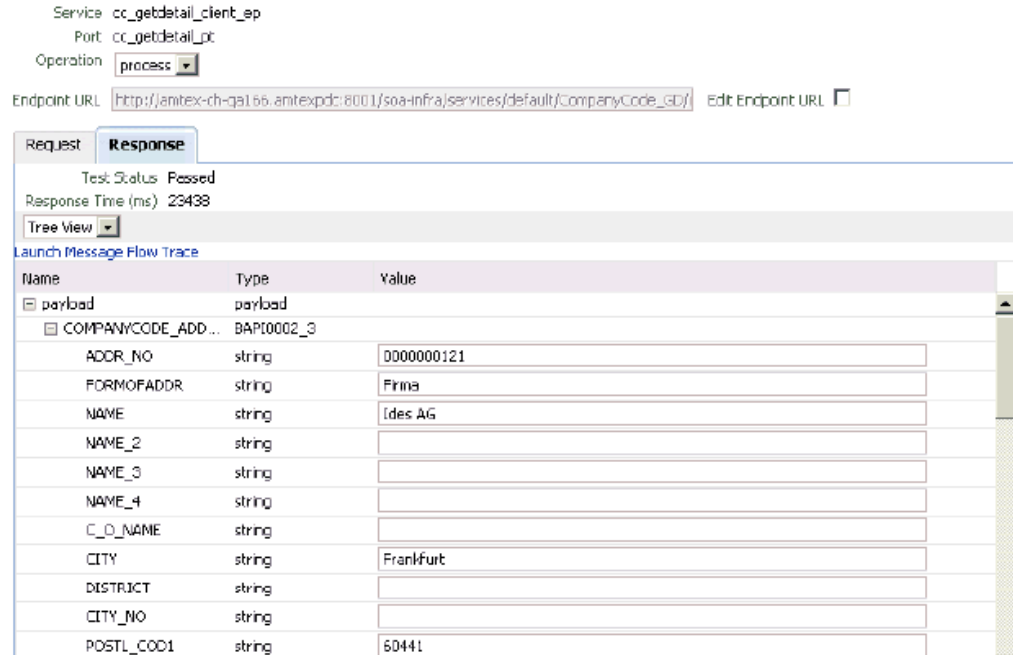


4. Click **Test**.



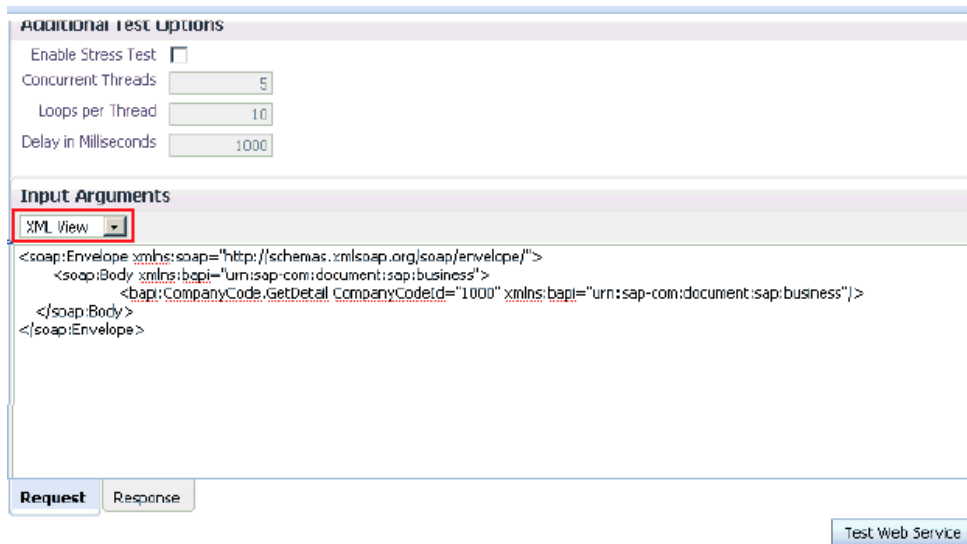
5. Click the **Request** tab.
6. Provide an appropriate input value in the Value field and click **Test Web Service**.

The output response is received in the Oracle Enterprise Manager console, as shown in the following image.



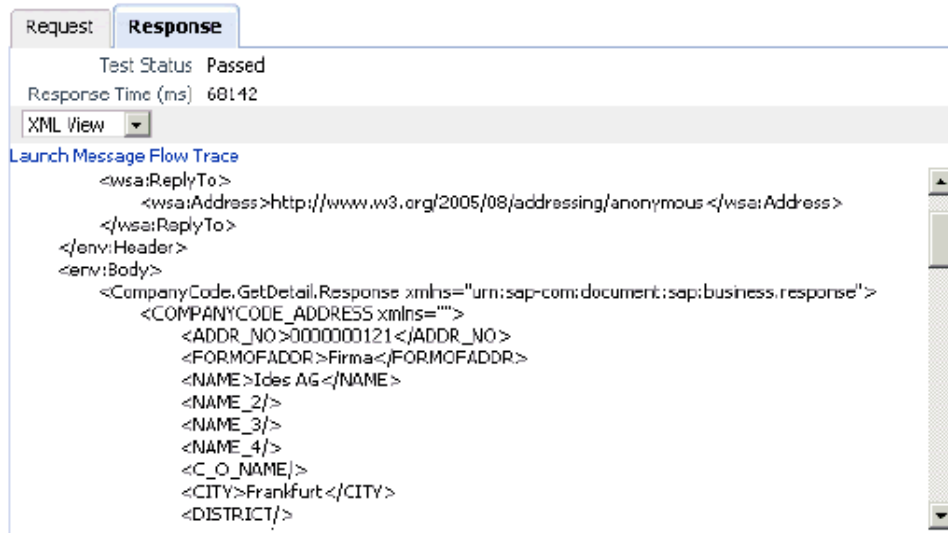
Perform the following steps to invoke the input XML document using XML View:

- a. Select **XML View** from the 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 the following image.



### 5.4.6 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 6, "Integration With Mediator Service Components in the Oracle SOA Suite"](#).

## 5.5 Designing an Inbound BPEL Process for Event Integration

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.

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

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

---



---

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

---



---

Before you design a BPEL process, you must generate the respective WSDL file using Application Explorer. See "[Generating WSDL for Event Integration](#)" on page 5-35 for more information.

### 5.5.1 Generating WSDL for Event Integration

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

---



---

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

---



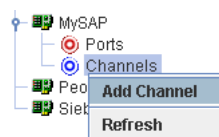
---

#### Creating a Channel

To create a channel:

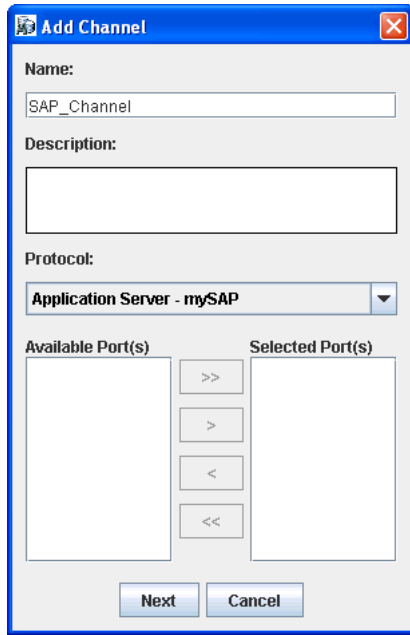
1. Start **Application Explorer** and connect to a J2CA configuration.
2. In the left pane, expand the **Events** node.
3. Expand the **MySAP** node.

The Ports and Channels nodes appear in the left pane.



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

The Add Channel dialog is displayed.

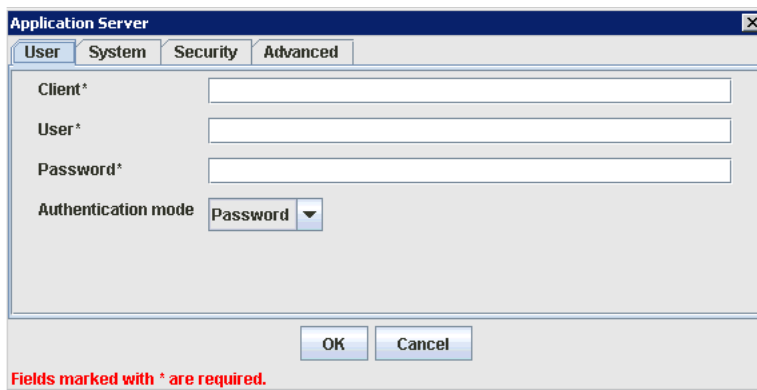


Perform the following steps:

- 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. The following tabs are available:

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



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



**Table 5–1 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, <b>Password</b> is selected from the list.

7. Click the **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 5–2 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.

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 5–3 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.

**Table 5–3 (Cont.) Security Tab Parameters**

Target Parameter	Description
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 5–4 Advanced Tab Parameters**

Target Parameter	Description
IDOC Format	Select an IDOC type from the list: <ul style="list-style-type: none"> <li>■ XML (default)</li> <li>■ XML-CDATA-ENVELOPED</li> <li>■ NATIVE IDOC</li> </ul>
IDOC release	The IDOC versioning you want to use for your connection.
IDOC release provider	The IDOC release provider for your connection. Select <b>IDOC DOCREL field</b> (default), <b>SAP release</b> , or <b>user input</b> from the list.
SAP trace	Select this option to enable SAP traces.  The SAP traces are stored in the following locations during design time and run time:  Design Time: C:\oracle\Middleware\home_GA\Oracle_SOA1\soa\thirdparty\ApplicationAdapters\tools\iwae\bin  Run Time: C:\oracle\Middleware\home_GA\user_projects\domains\base_domain  The file name formats that are used for the SAP traces are as follows (for design time and run time): <ul style="list-style-type: none"> <li>■ rfc02664_04332.trc</li> <li>■ JCO100226_043846994.trc</li> </ul>
Processing Mode	Select the type of synchronous processing from the list. Possible values include <b>REQUEST</b> and <b>REQUEST_RESPONSE</b> .

11. Click **OK**.

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

---

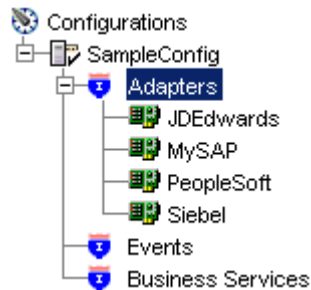
**Note:** Do not start the channel, as it is managed by BPEL PM Server. If you start the channel for testing and debugging purposes, stop it before run-time.

---

### Generating WSDL for Event Notification

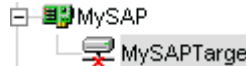
After you create a channel and verify that it is not started, you must generate WSDL for the event using Application Explorer.

1. Start Application Explorer.
2. Expand the **Adapters** node.  
A list of all adapters is displayed.



Perform the following steps:

- a. Expand the **MySAP** node.  
A list of your available targets is displayed.

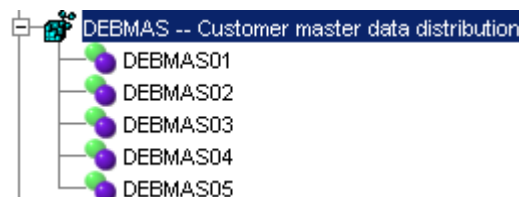


- b. Click a target name under the **MySAP** node, for example, **SAPTarget..**  
The Connection dialog displays the saved parameters.

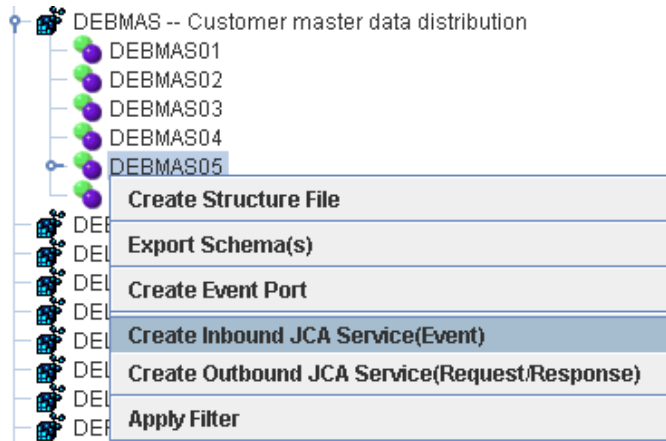
3. Verify your connection parameters.
4. Provide the required password.
5. Right-click the target name and select **Connect**.  
The x icon disappears, indicating that the node is connected.



6. Expand the **IDOCs** node and select **DEBMAS**.  
The DEBMAS list is displayed.

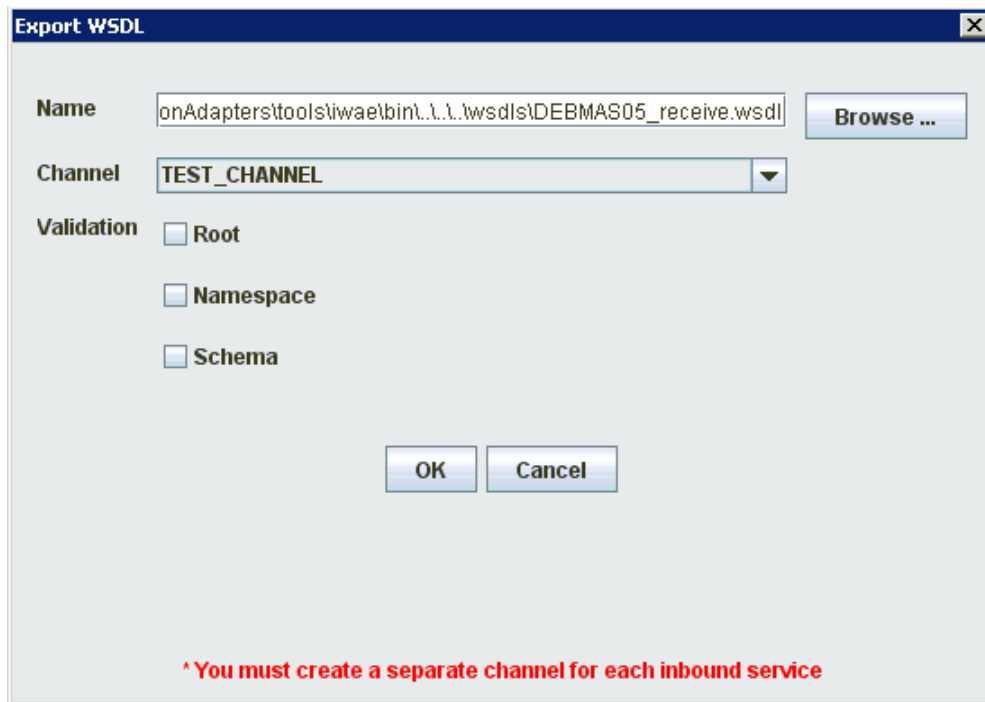


7. Right-click **DEBMAS05** from the **DEBMAS** list.



8. Select **Create Inbound JCA Service (Event)**.

The Export WSDL dialog is displayed.



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, 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.

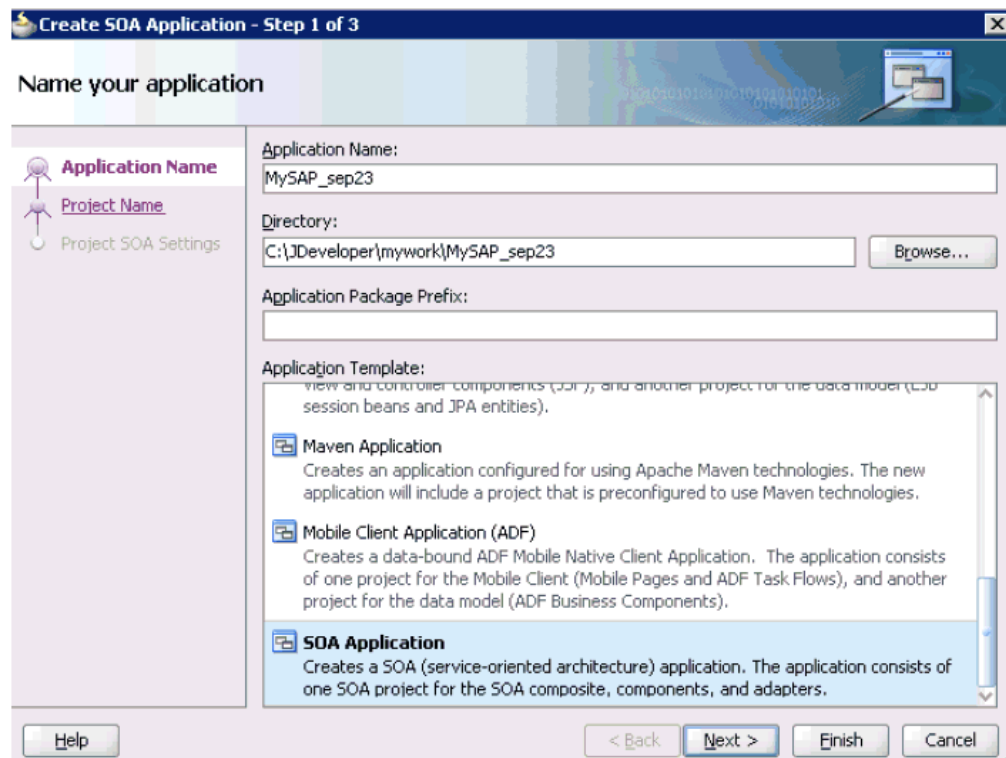
9. Click OK.

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.

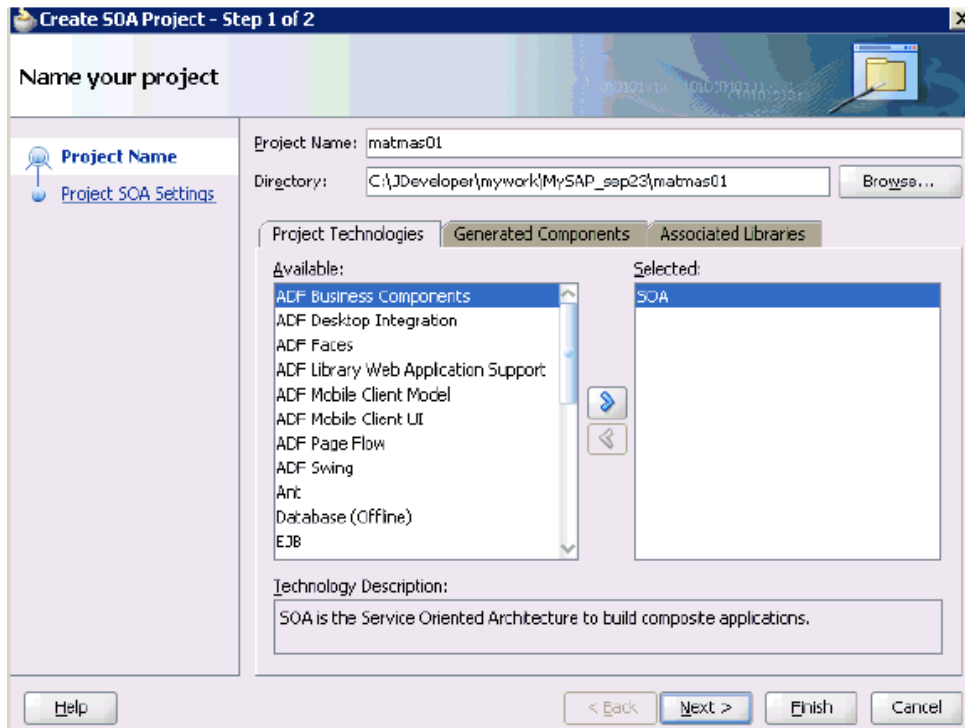
## 5.5.2 Creating an Empty Composite for SOA

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

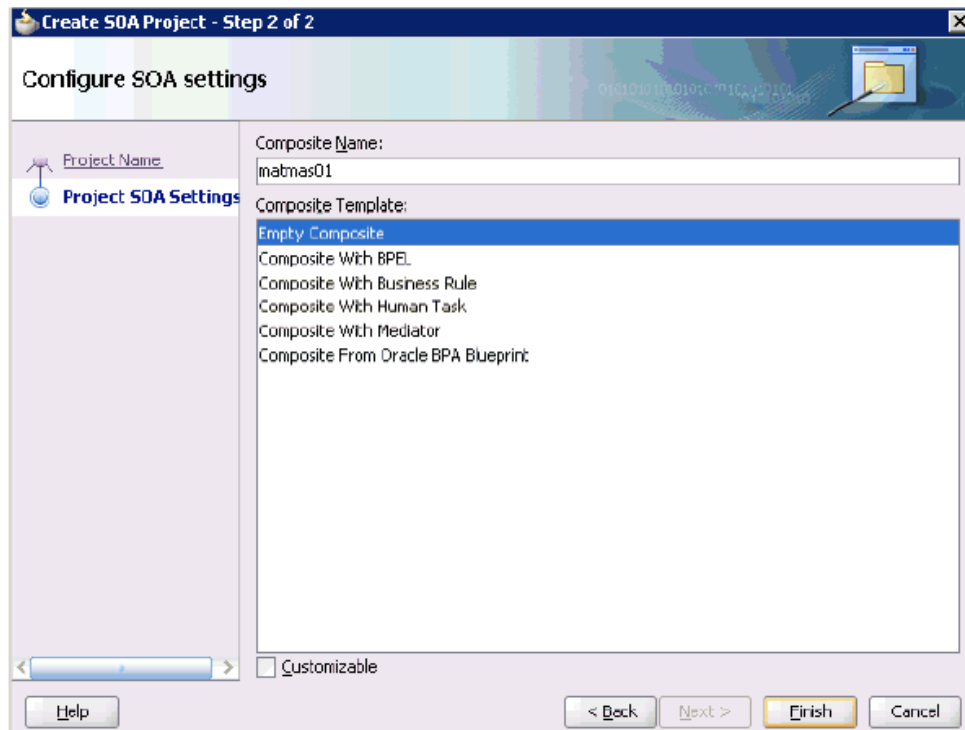
1. Create a new SOA application.
2. Enter name for the new SOA Application and click Next.



The Name your project page is displayed.



3. Enter a project name (for example, matmas01) and click **Next**.  
The Configure SOA settings page is displayed.



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

### 5.5.3 Defining a BPEL Inbound Process

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

1. Configuring a Third Party Adapter Service Component
2. Configuring an Inbound BPEL Process Component

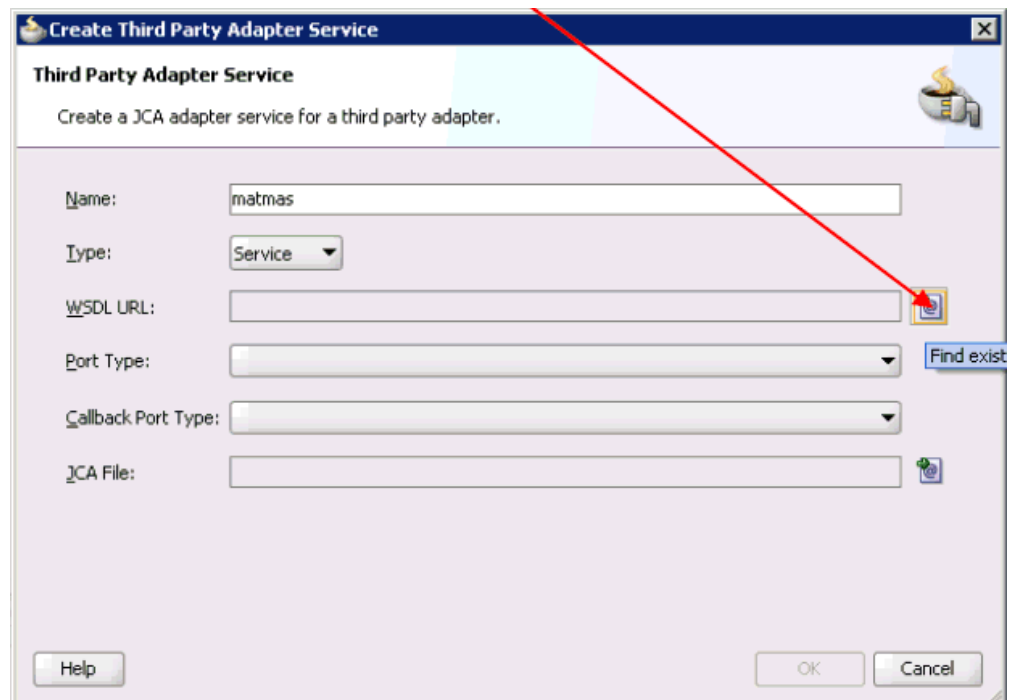
#### Creating a Third Party Adapter Service Component

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.

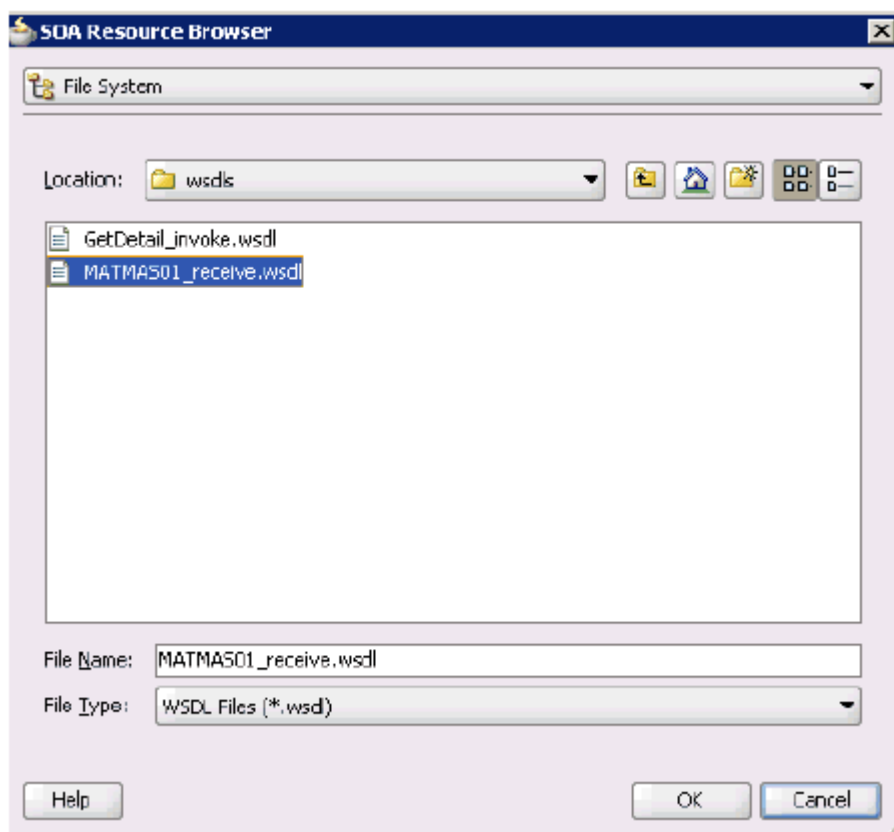


The Create Third Party Adapter Service dialog is displayed.



2. Enter a name for the third party adapter service.
3. Ensure that **Service** is selected from the Type list (default).
4. Click the **Find existing WSDLs** icon, which is located to the right of the WSDL URL field.

The SOA Resource Browser dialog is displayed.



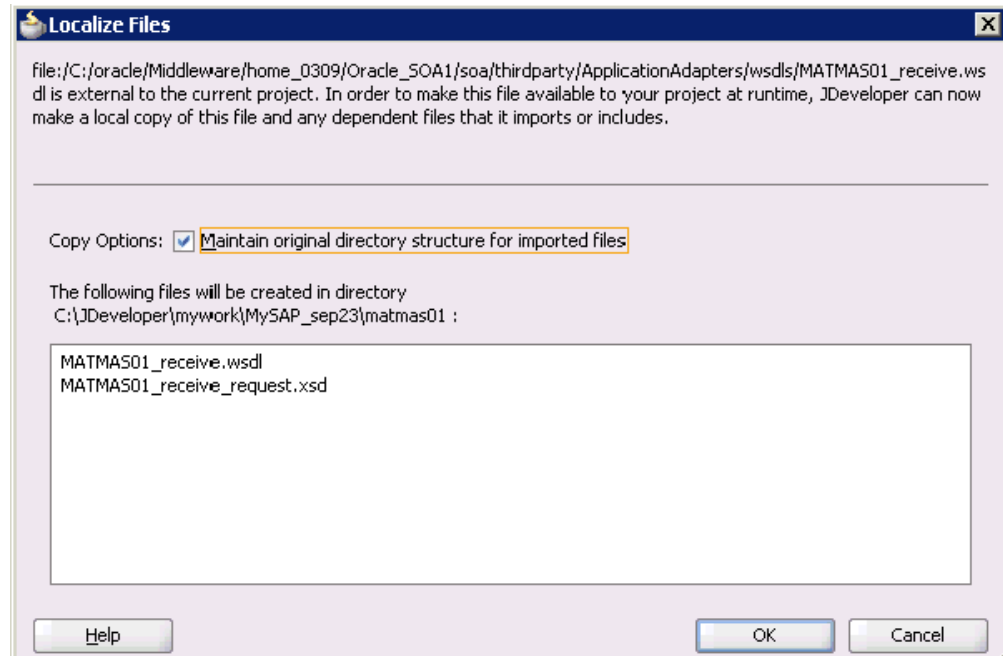
5. Browse and select an inbound WSDL file from the following directory:

C:\oracle\Middleware\home\_GA\Oracle\_  
SOA1\soa\thirdparty\ApplicationAdapters\wsdl

6. Click **OK**.

The Localize Files dialog is displayed.

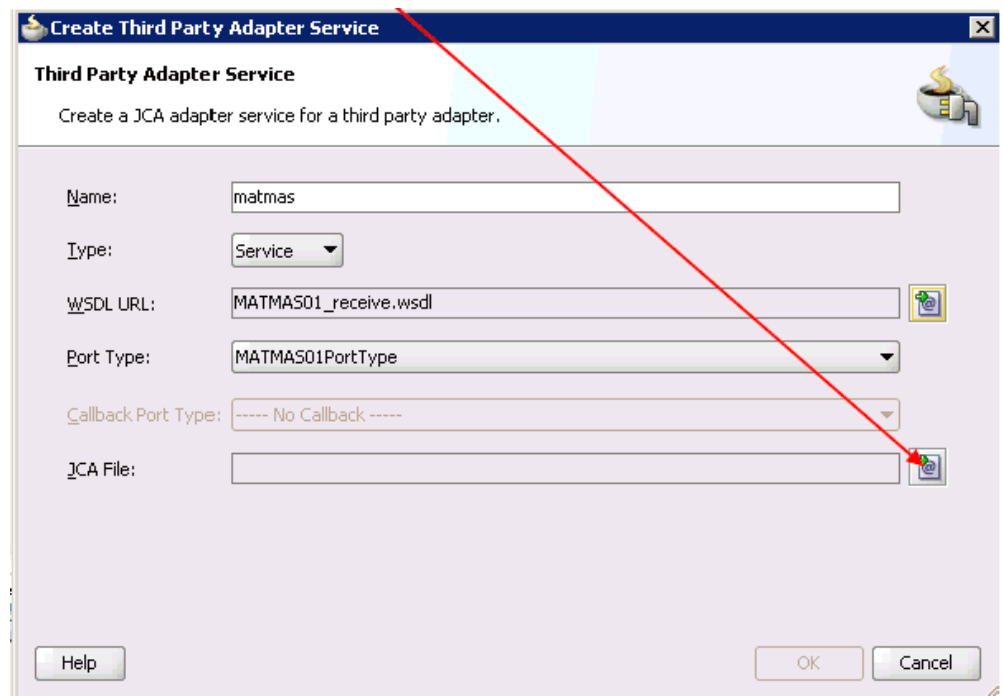




7. Click **OK**.

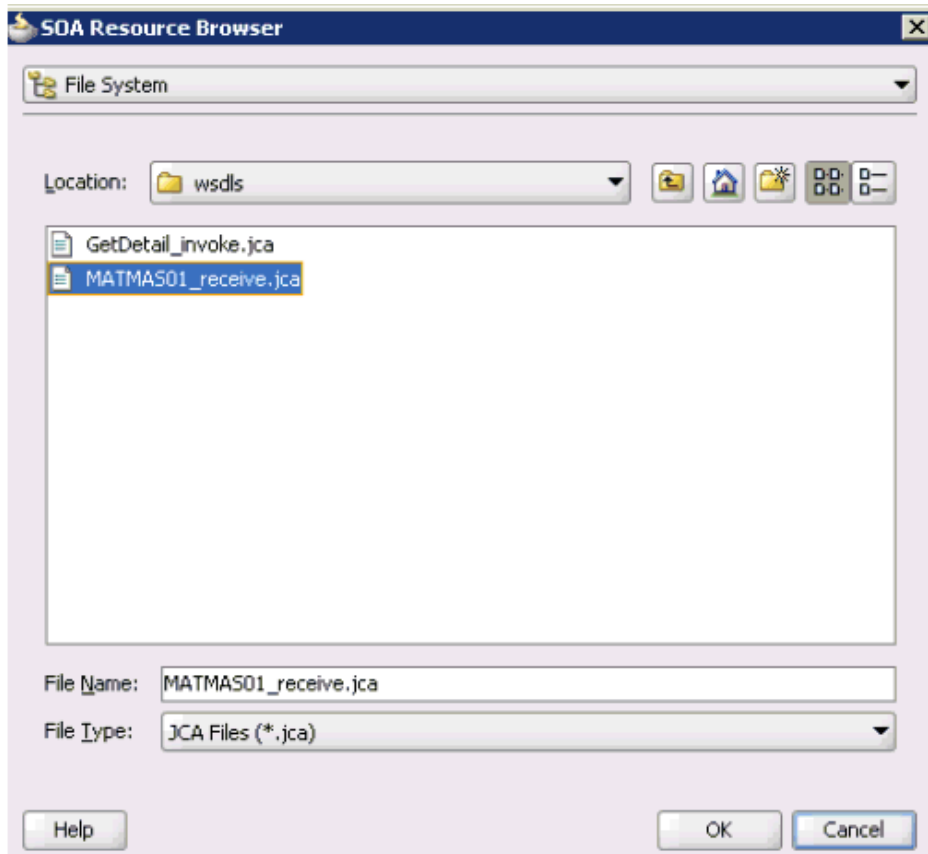
The inbound WSDL file and associated receive/request XML schema file (.xsd) are imported to the project folder that has been created.

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



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

The SOA Resource Browser dialog is displayed.

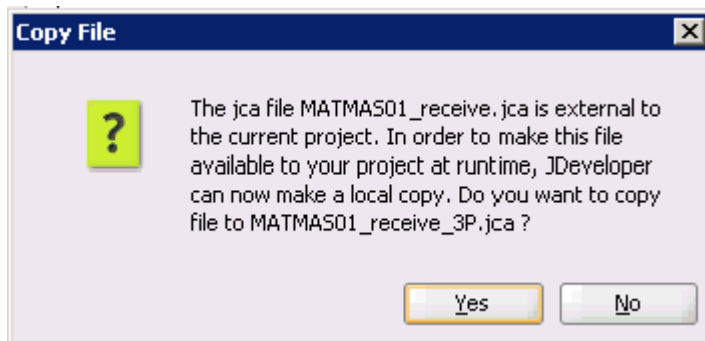


9. Browse and select the JCA properties file from the following directory:

C:\oracle\Middleware\home\_GA\Oracle\_  
SOA1\soa\thirdparty\ApplicationAdapters\wsdl:s

10. Click **OK**.

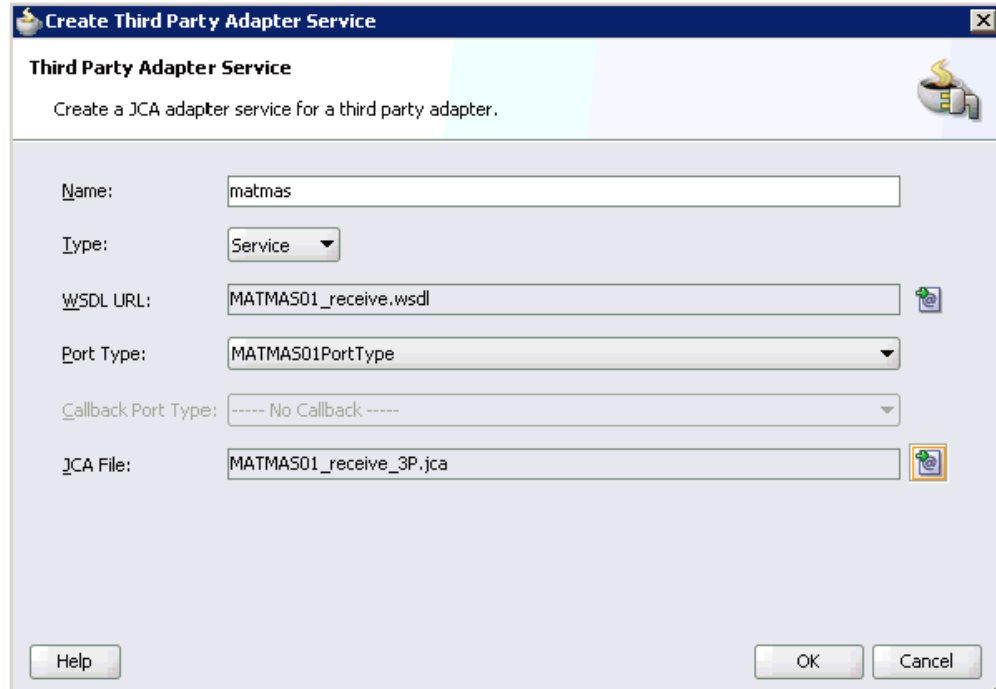
The following message is displayed.



11. Click **Yes**.

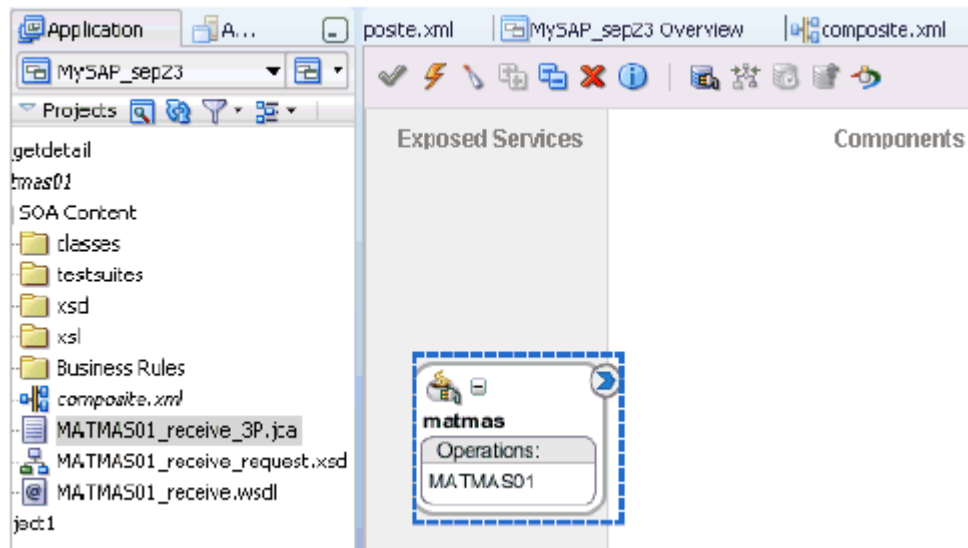
A copy of the JCA properties file is made in the project folder.

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



12. Click OK.

The third party adapter service component (matmas) is created in the Exposed Services pane, as shown in the following image.

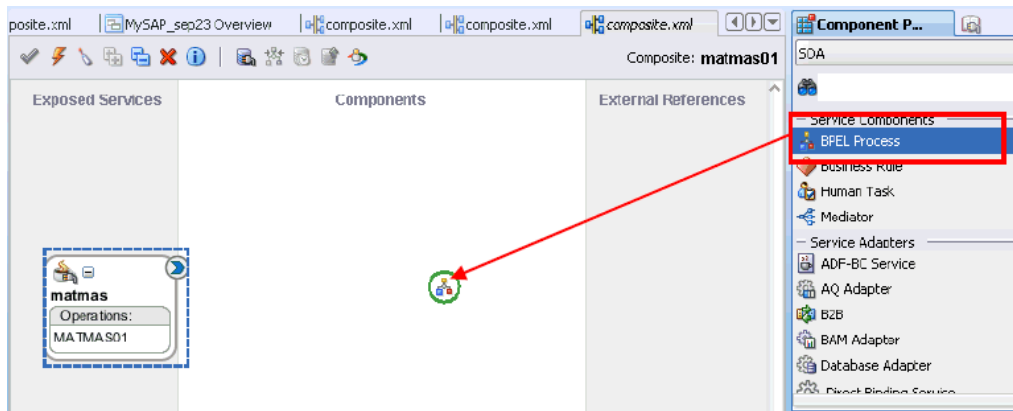


You are now ready to configure an inbound BPEL process component.

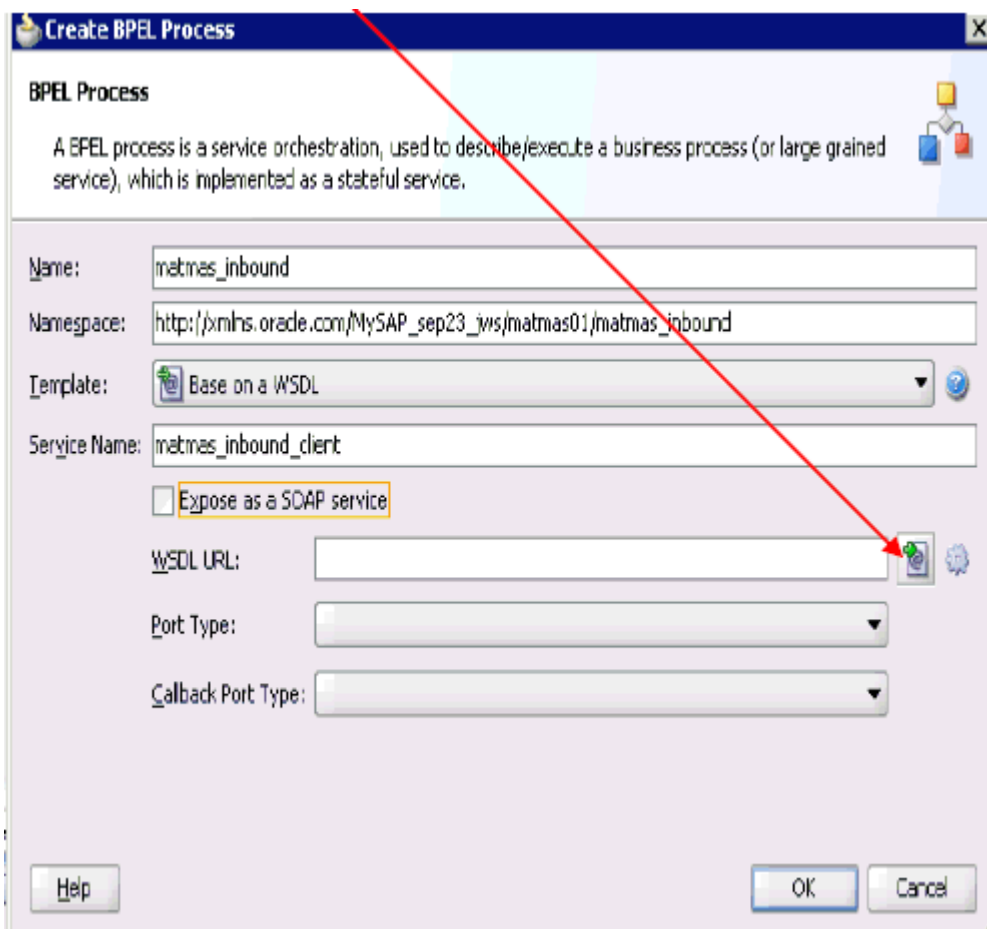
### Creating an Inbound BPEL Process Component

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.

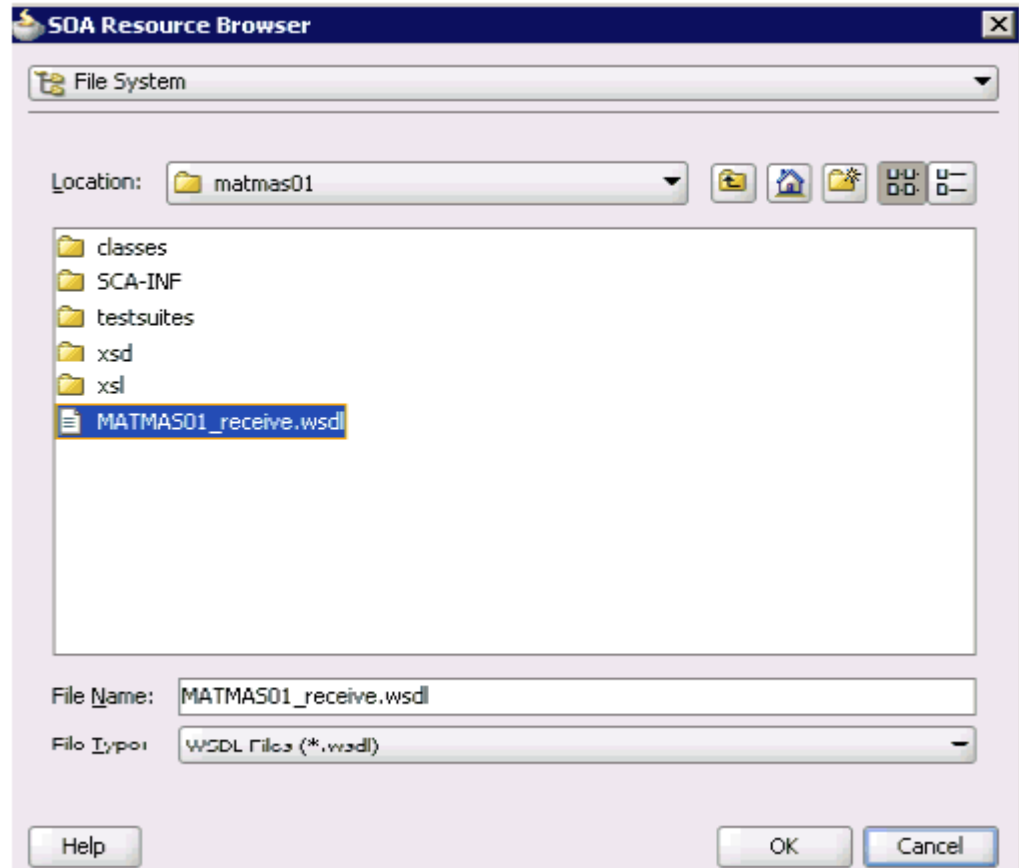


The Create BPEL Process dialog is displayed.



2. In the Name field, enter a name to identify the new inbound BPEL process component (for example, matmas\_inbound).
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 SOA Resource Browser dialog is displayed.



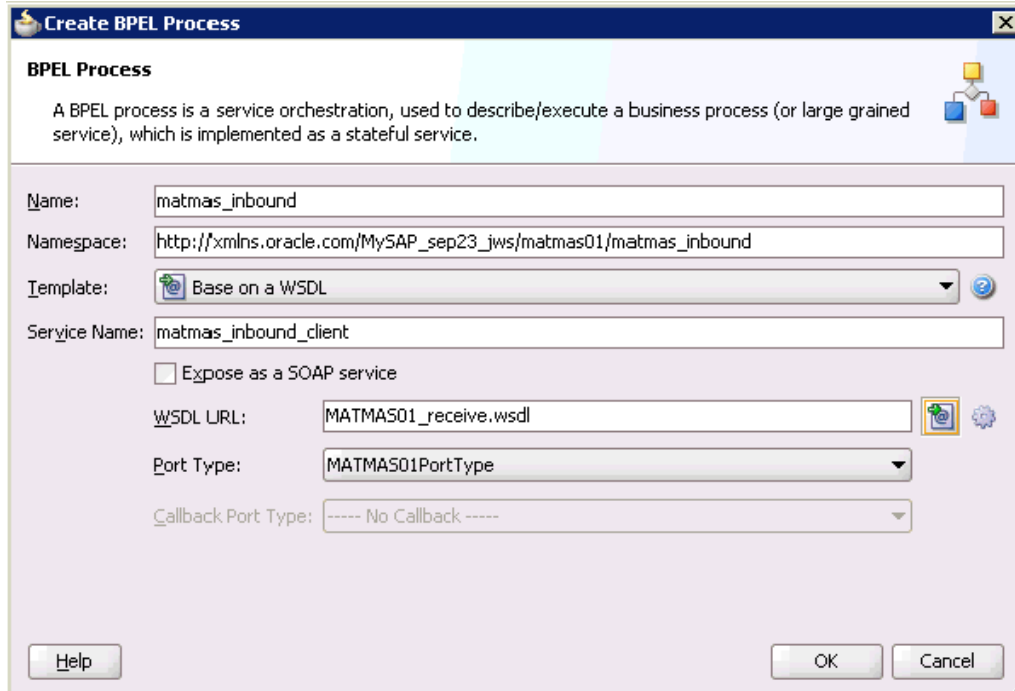
6. Browse and select an inbound WSDL file from the project folder.

For example:

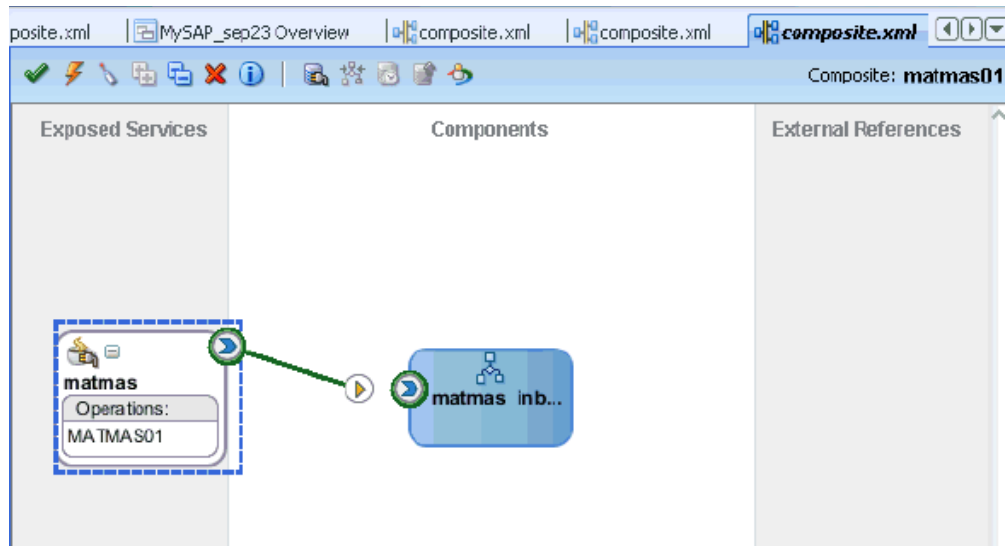
C:\JDeveloper\mywork\MySAP\_sep23\matmas01\MATMAS01\_receive.wsdl

7. Click **OK**.

You are returned to the Create BPEL Process dialog.

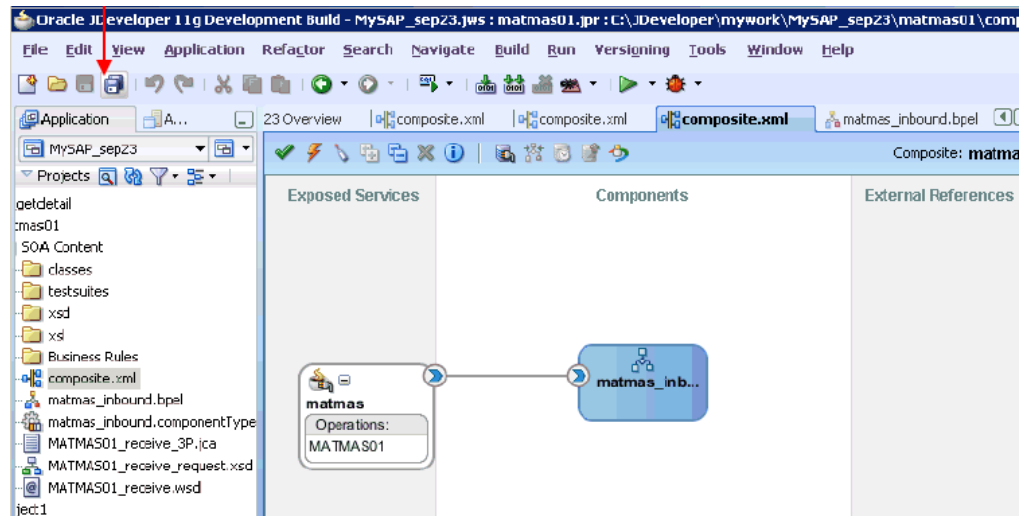


8. Click **OK**.



9. Create a connection between the third party adapter service component (matmas) and the inbound BPEL process component (matmas\_inbound).

10. Double-click **composite.xml** in the left pane.



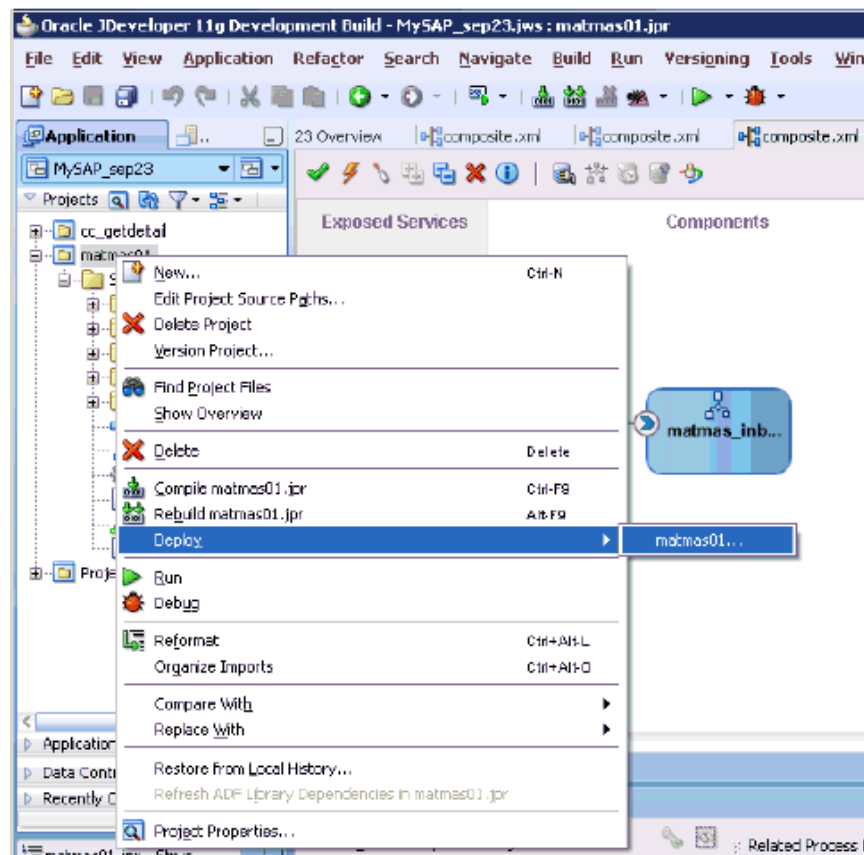
11. Click the **Save All** icon in the menu bar to save the new inbound BPEL process component that was configured.

You are now ready to deploy the BPEL inbound process.

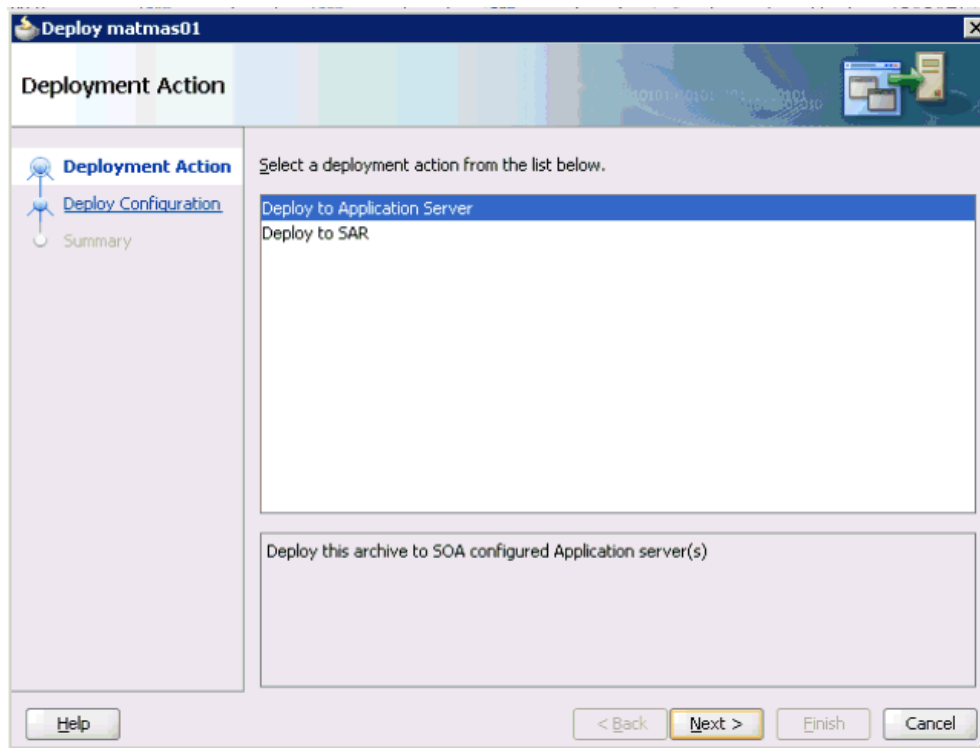
#### 5.5.4 Deploying the BPEL Inbound Process

Perform the following steps to deploy the BPEL inbound process.

1. Right-click the project name in the left pane (for example, **matmas01**), select **Deploy**, and click **matmas01**.

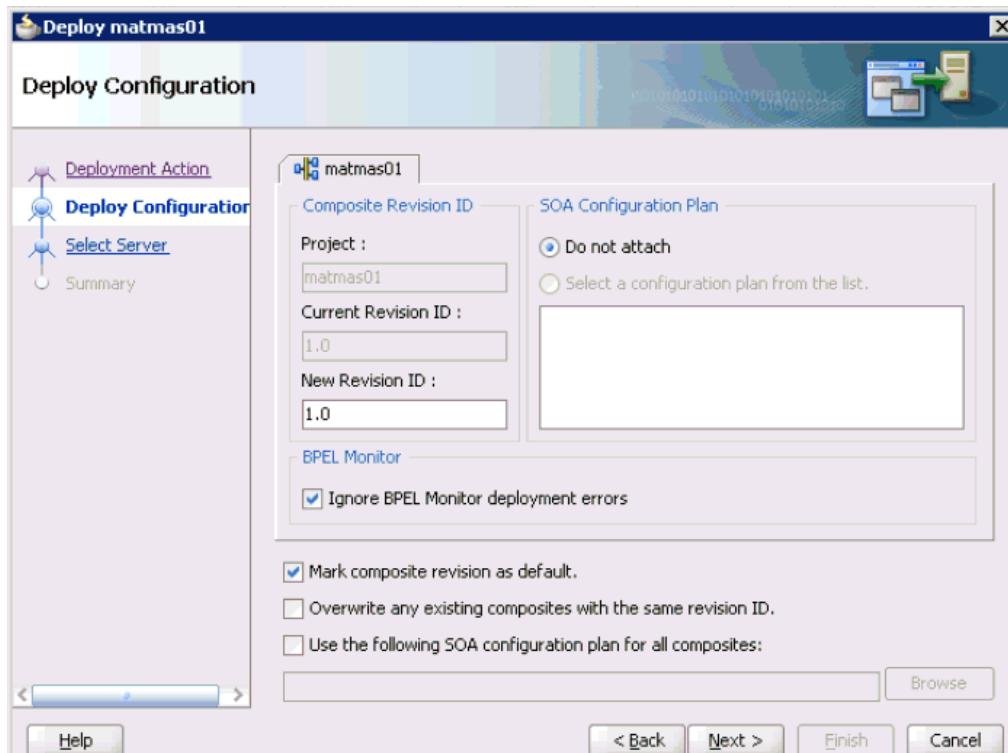


The Deployment Action page is displayed.



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

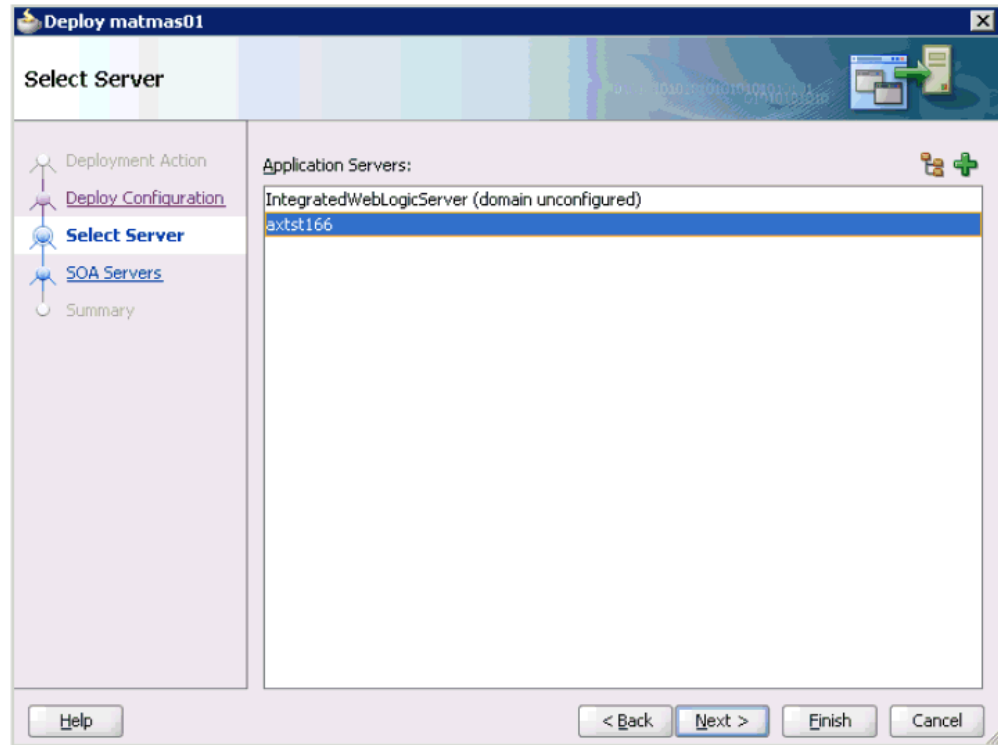
The Deploy Configuration page is displayed.





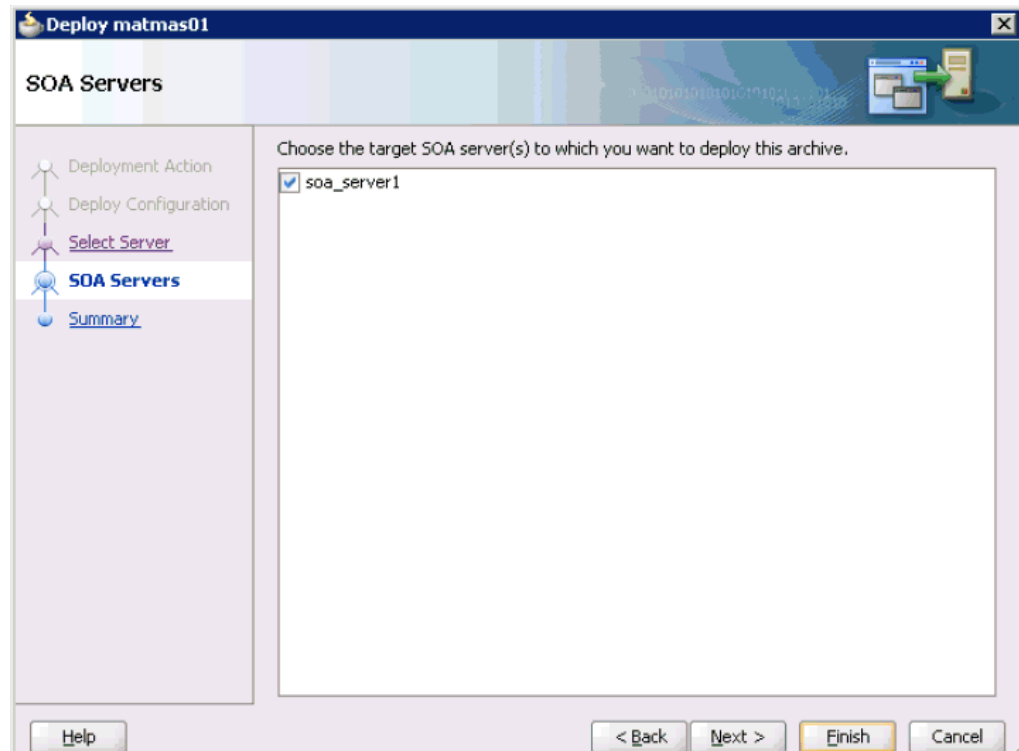
4. Leave the default values selected and click **Next**.

The Select Server page is displayed.



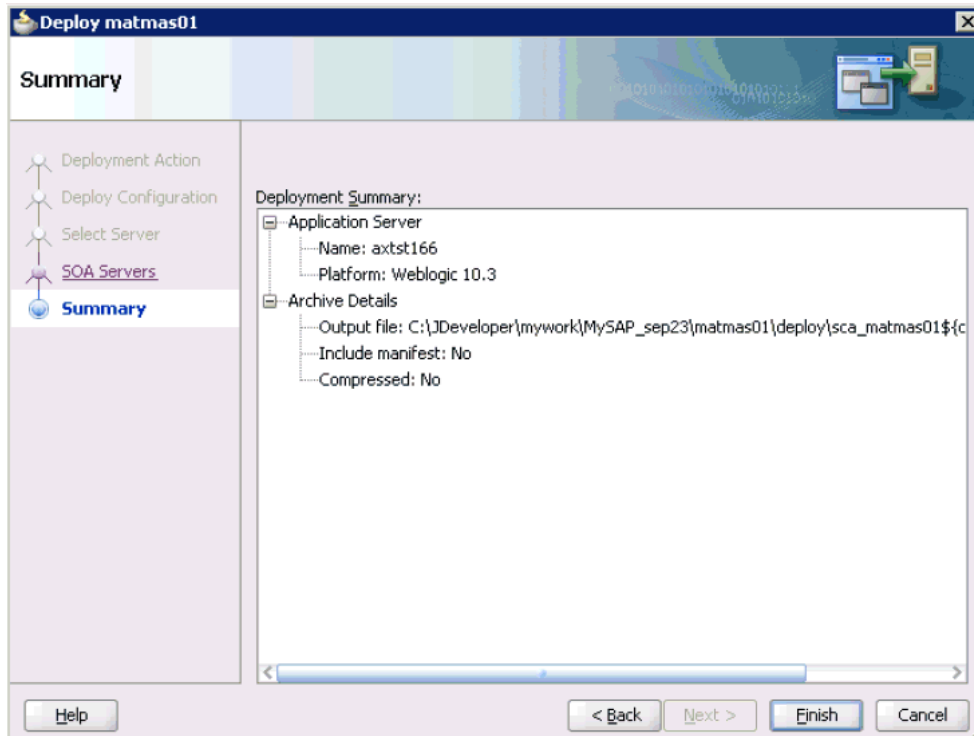
5. Select an available application server that was configured and click **Next**.

The SOA Servers page is displayed.



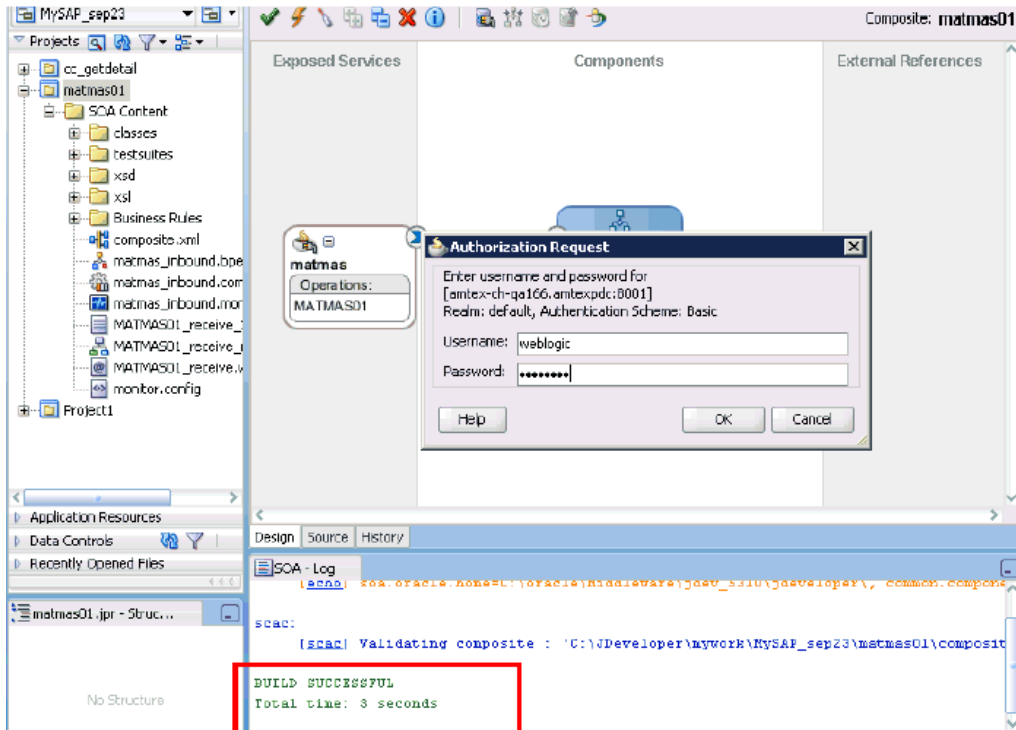
6. Select a target SOA server and click **Next**.

The Summary page is displayed.



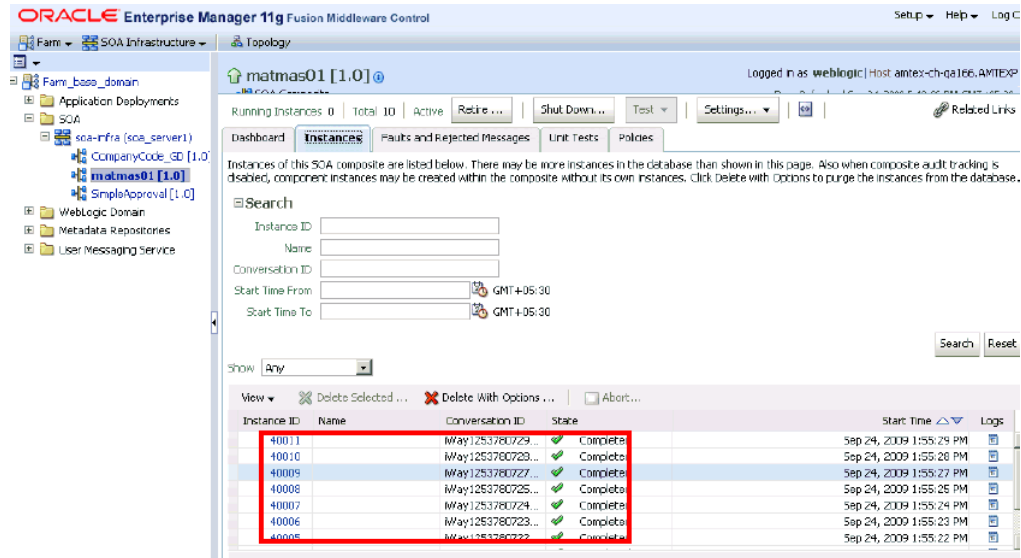
7. Review and verify all the available deployment information for your project and click **Finish**.

The process is deployed successfully.



If an Authorization Request dialog is displayed during the deployment process, provide the required user name and password and click **OK**.

Once event messages are triggered through SAP GUI, successful instances are received in the Oracle Enterprise Manager console.



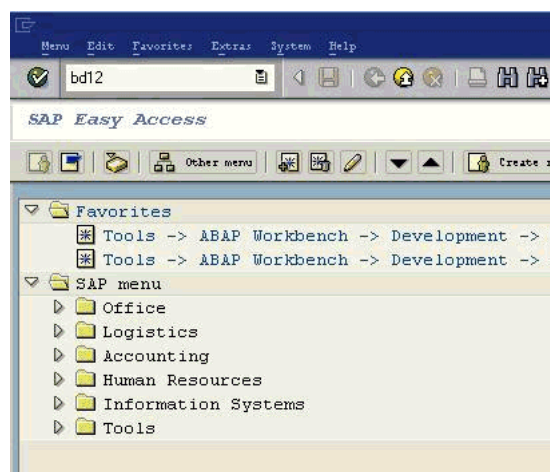
## 5.5.5 Triggering an Event in SAP R/3

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 3-24

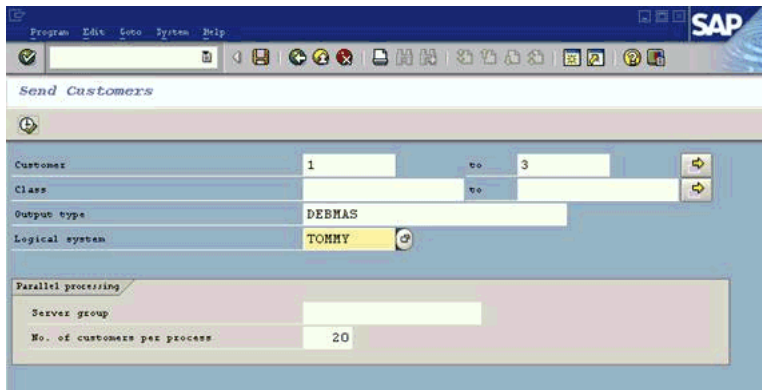
The following topics describe how to trigger an event in SAP R/3 and verify event integration using Oracle Application Adapter for SAP R/3.

To trigger an event in SAP R/3:

1. Start the SAP Workbench and log in to the SAP R/3 system.

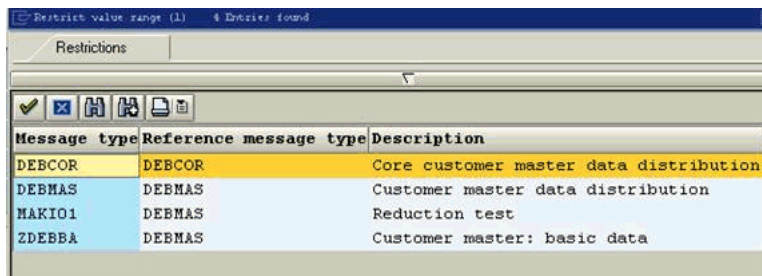


2. Run the **bd12** transaction.



Enter the following information in the Send Customers window:

- a. In the Customer field, enter a customer number with a range from 1 to 3.
  - b. In the Output type field, enter **DEBMAS**.
  - c. In the Logical system field, specify the logical system you are using with SAP R/3.
3. Click the **check mark** icon in the upper left-hand corner.



4. Ensure **DEBMAS** appears in the Message type column.
5. Click the **Execute** button.

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

### Verifying the Results

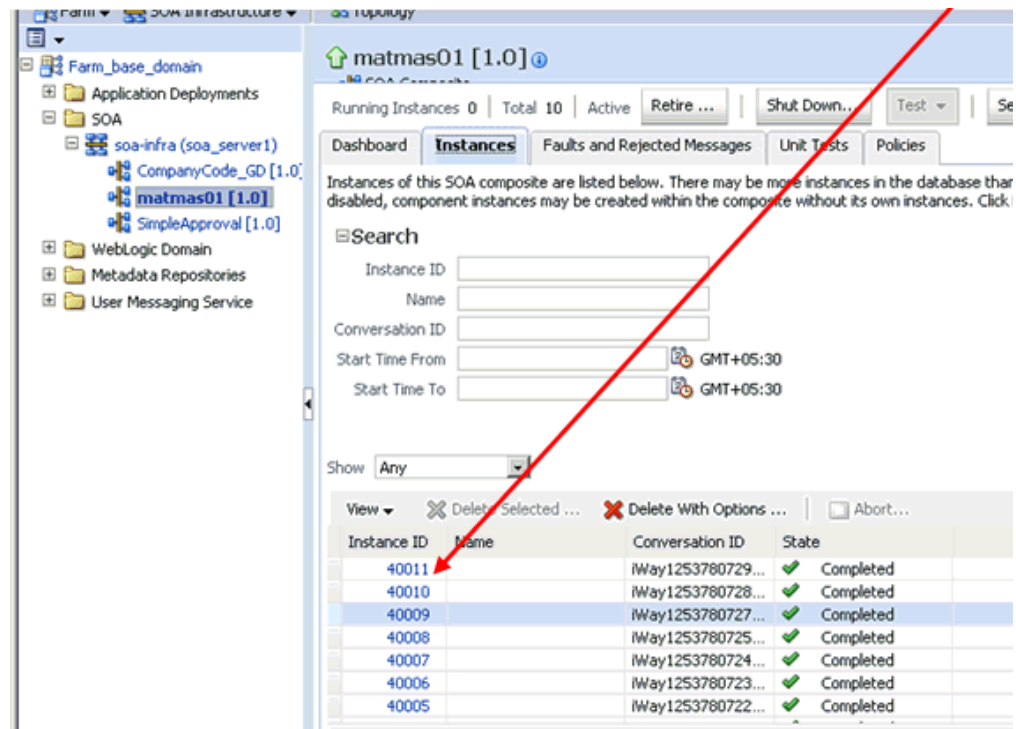
To verify your results:

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 (for example, matmas01).



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.



---

---

## Integration With Mediator Service Components in the Oracle SOA Suite

This chapter contains the following examples:

- [Configuring a New Application Server Connection](#)
- [Configuring a Mediator Outbound Process](#)
- [Configuring a Mediator Inbound Process](#)

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. See [Appendix A, "Configuring SAP R/3 for Inbound and Outbound Processing"](#) for more information.

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 3, "Configuring Oracle Application Adapter for SAP R/3"](#).
- How to configure Oracle JDeveloper. For more information, see [Chapter 5, "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 3,](#)

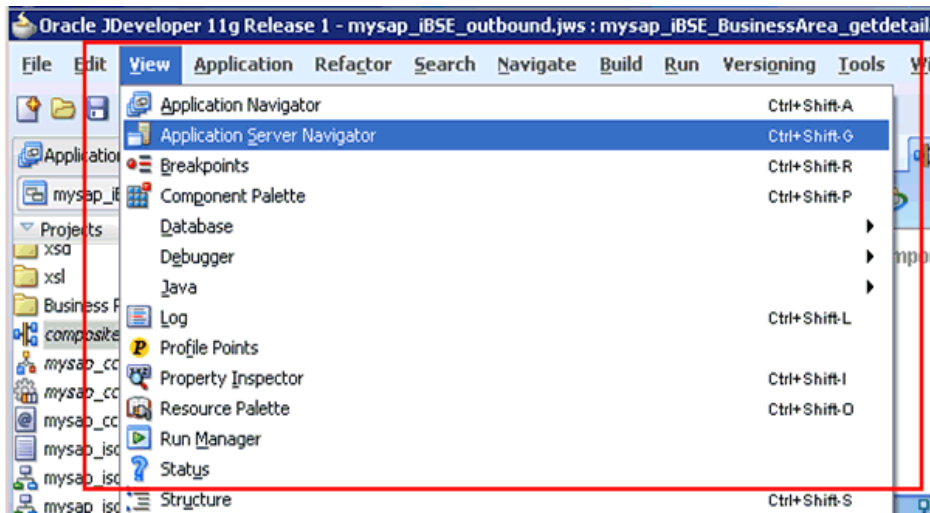
"Configuring Oracle Application Adapter for SAP R/3". Integration logic is modeled in Mediator.

2. **Runtime:** The underlying platform treats this metadata as run-time instructions to enable the communication between participating applications.

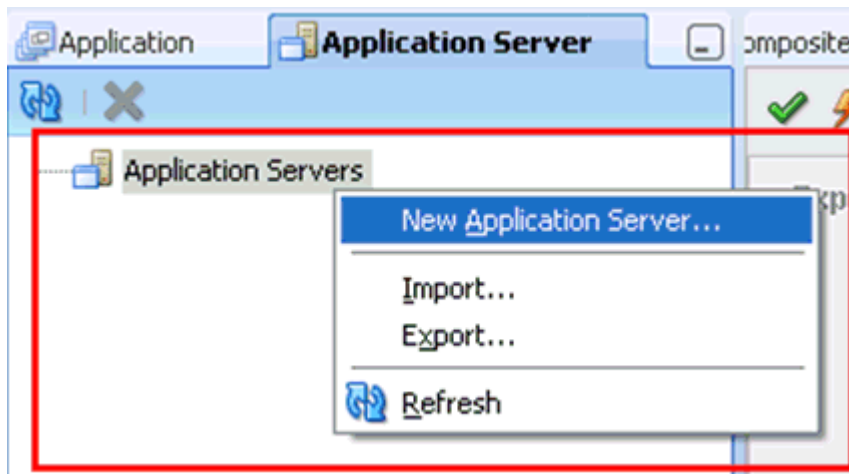
## 6.1 Configuring a New Application Server Connection

To configure a new Application Server connection in Oracle JDeveloper:

1. Open Oracle JDeveloper on your system.
2. From the menu bar, click **View** and select **Application Server Navigator**.



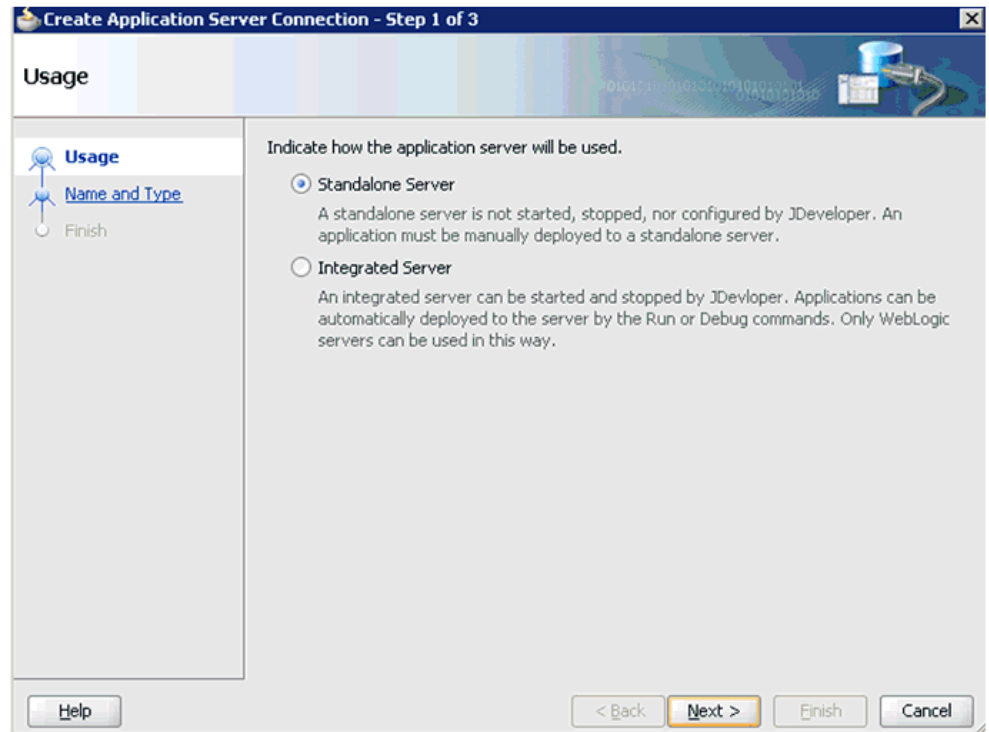
The Application Server tab is displayed.



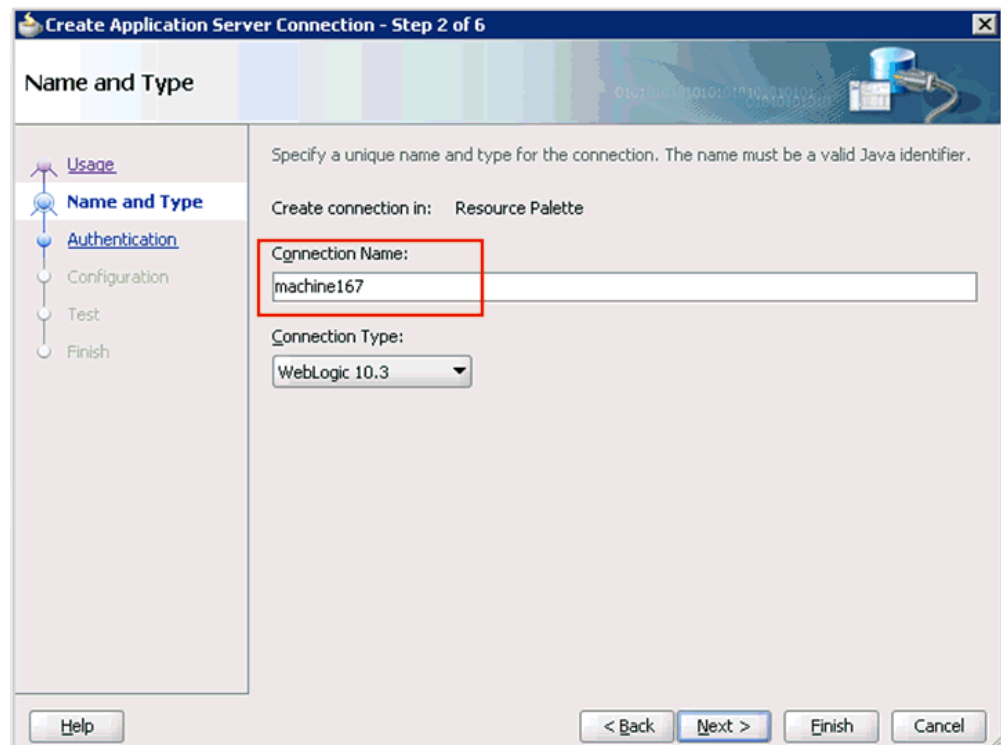
3. Right-click **Application Servers** and select **New Application Server** from the context menu.

The Create Application Server Connection Wizard is displayed.

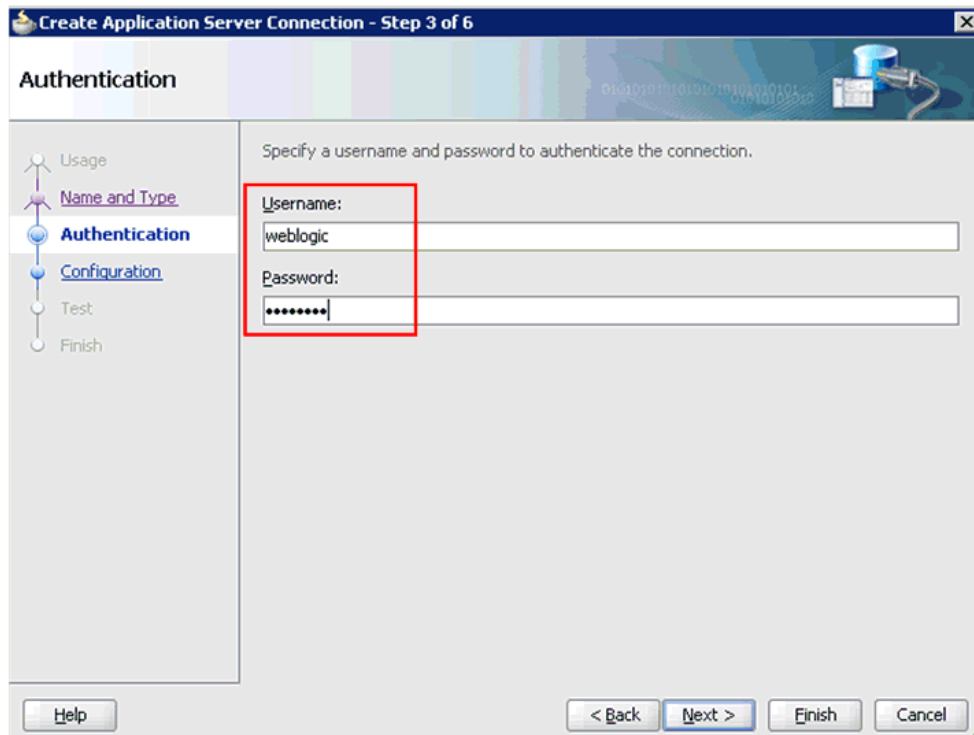




- Accept the default selection (Standalone Server) and click **Next**.  
The Name and Type page is displayed.



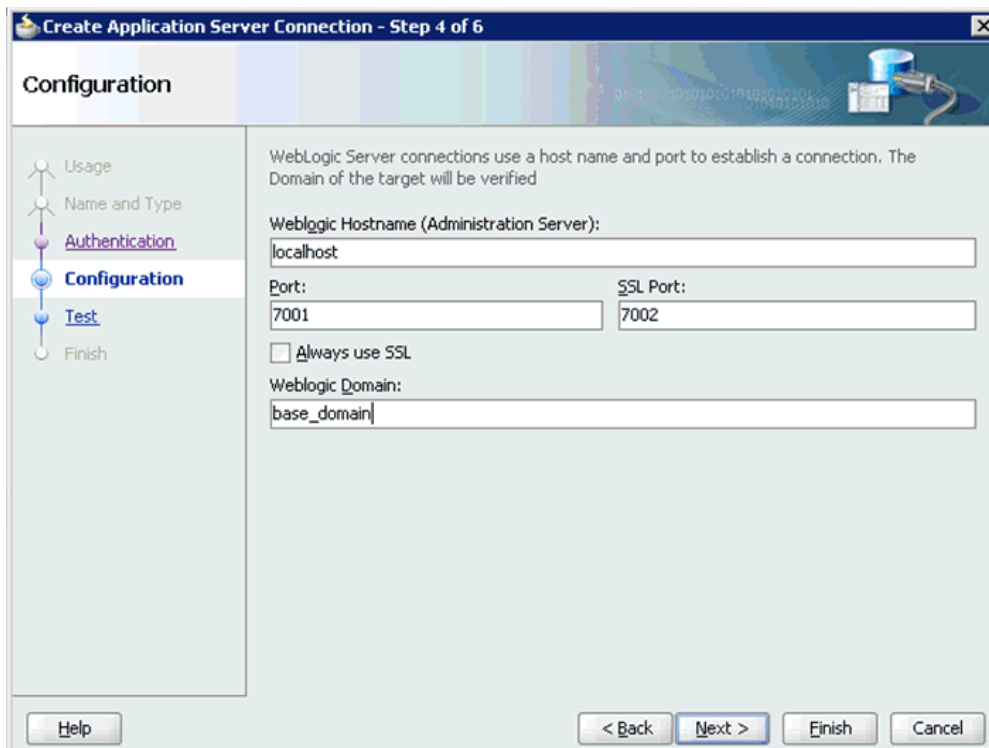
- Specify a new name for the Application Server connection and click **Next**.  
The Authentication page is displayed.



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

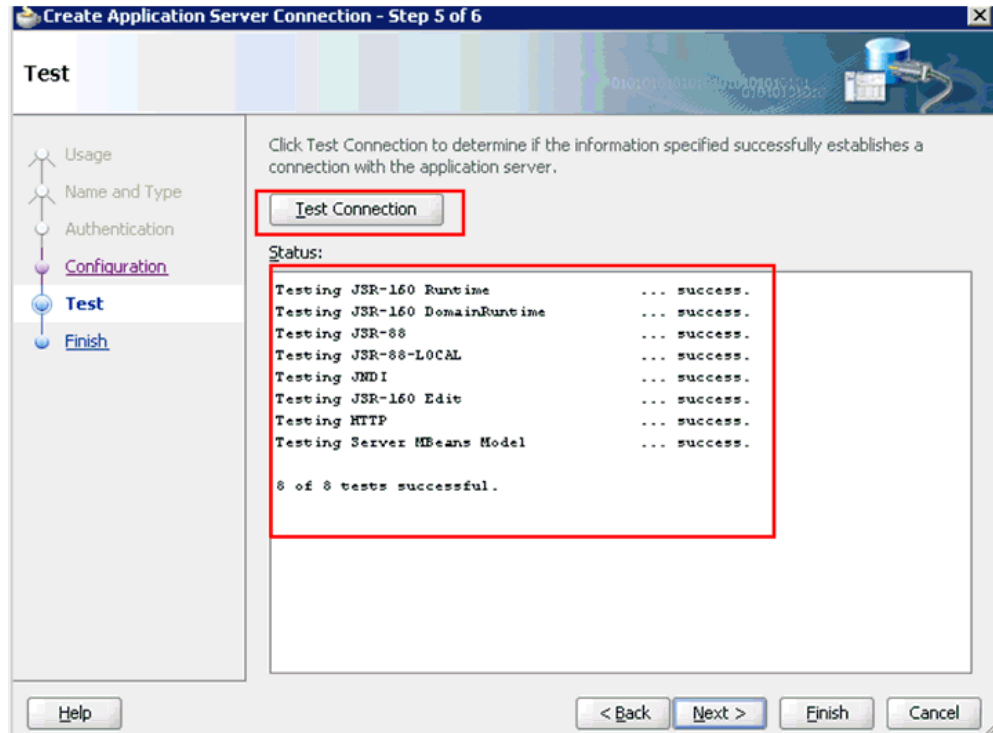
7. Click Next.

The Configuration page is displayed.



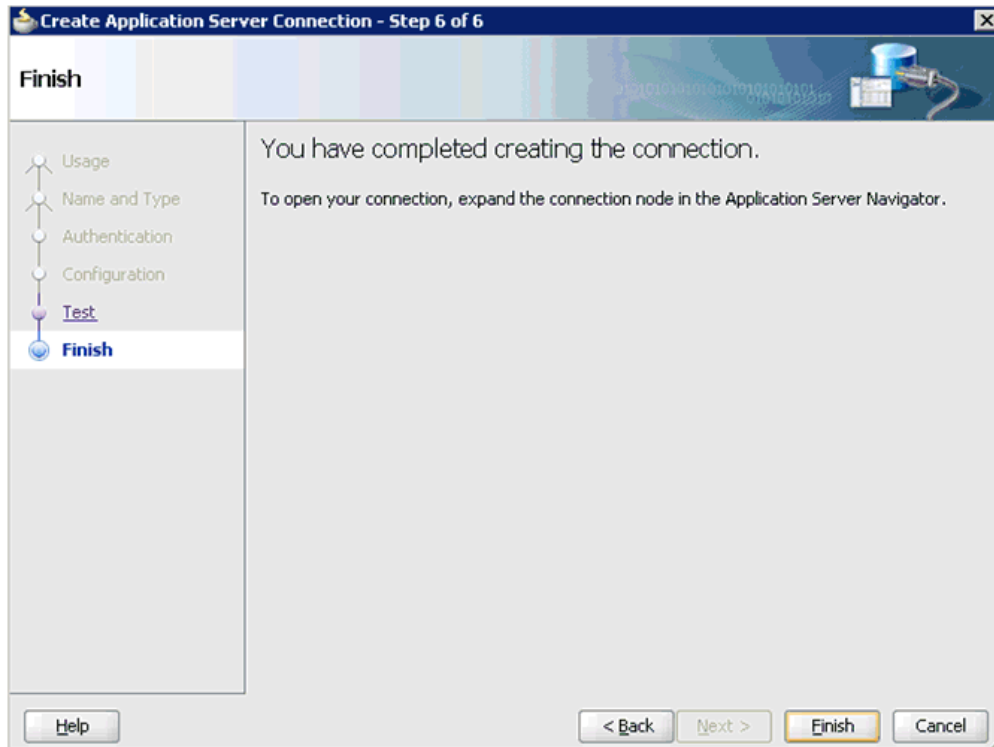
8. Specify the Oracle WebLogic host name (for example, localhost), which is the system IP where the process must deploy and Oracle WebLogic domain (for example, base\_domain).
9. Click **Next**.

The Test page is displayed.



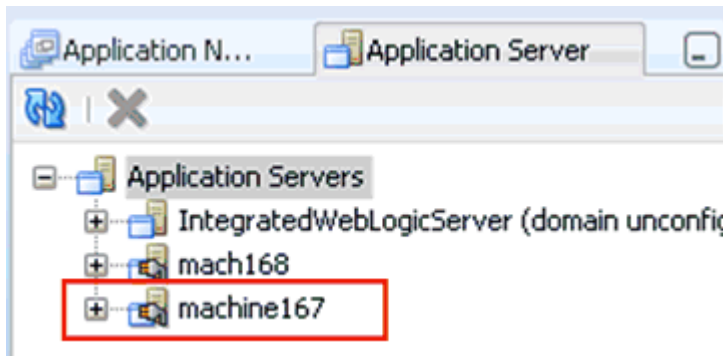
10. Click **Test Connection**.
11. Make sure that the test status is successful.
12. Click **Next**.

The Finish page is displayed.



**13. Click Finish.**

The new Application Server connection is listed in the left pane (Application Server tab), as shown in the following image.



## 6.2 Configuring a Mediator Outbound Process

The following example describes how to configure a Mediator outbound process to your SAP R/3 system, using a Mediator project in Oracle JDeveloper.

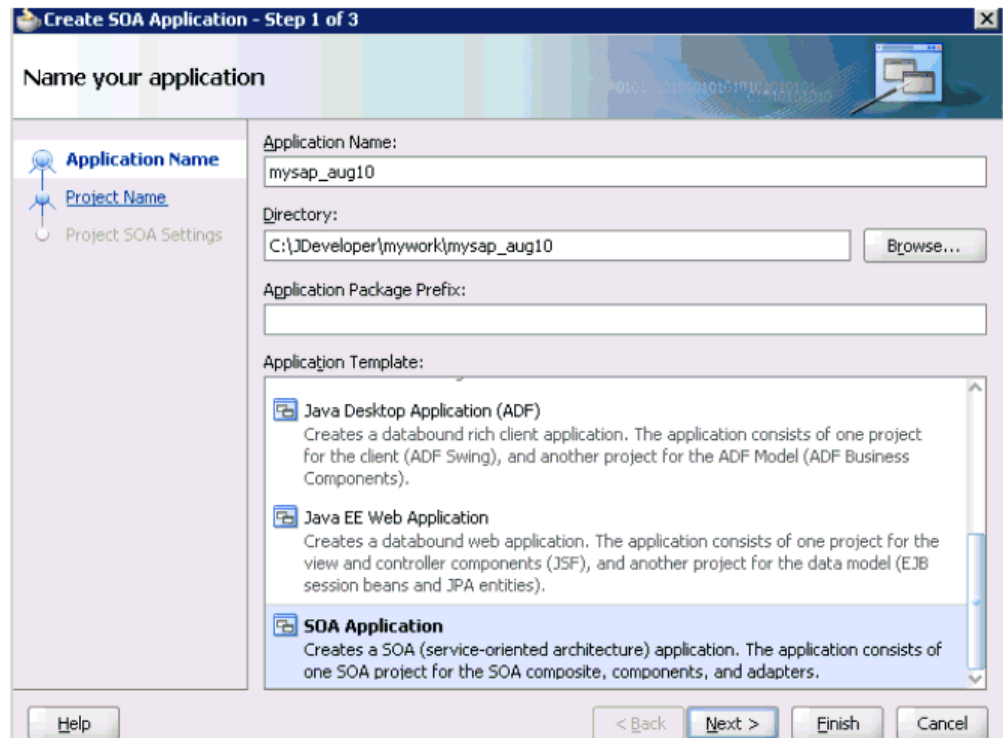
### Prerequisites

Before you design a Mediator outbound process, you must generate the respective WSDL file using Application Explorer. See ["Generating WSDL for Request/Response Service"](#) on page 5-7 for more information.

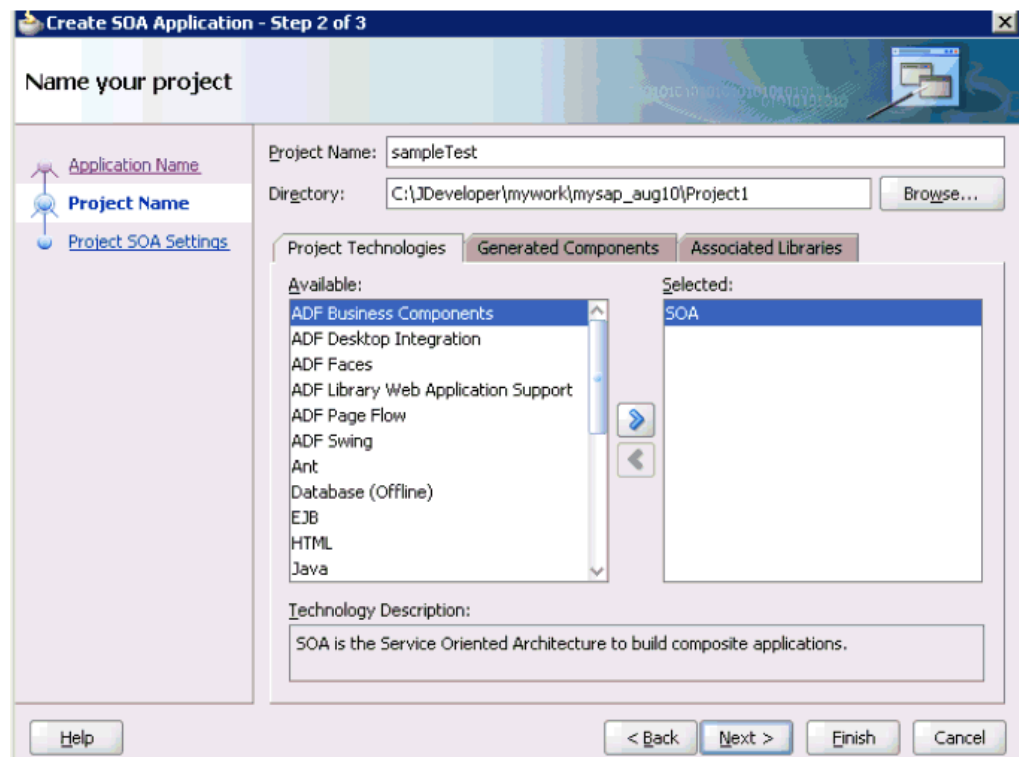
### 6.2.1 Creating an Empty Composite for SOA

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

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

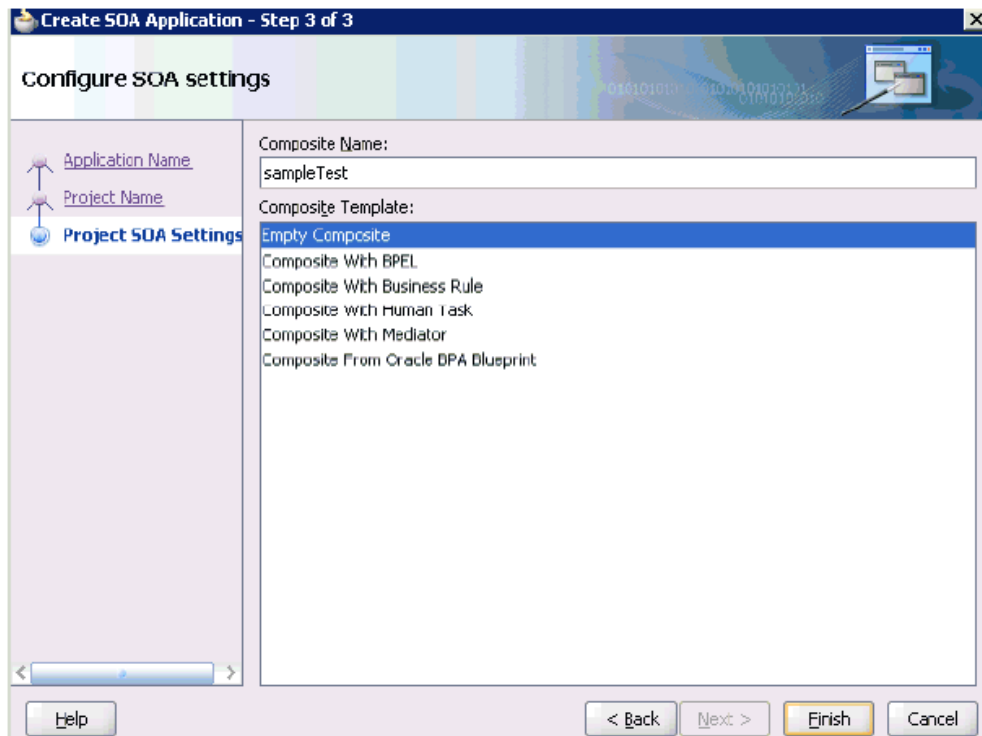


The Name your project page is displayed.



3. Enter a project name (for example, sampleTest) and click **Next**.

The Configure SOA settings page is displayed.



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

## 6.2.2 Defining a Mediator Outbound Process

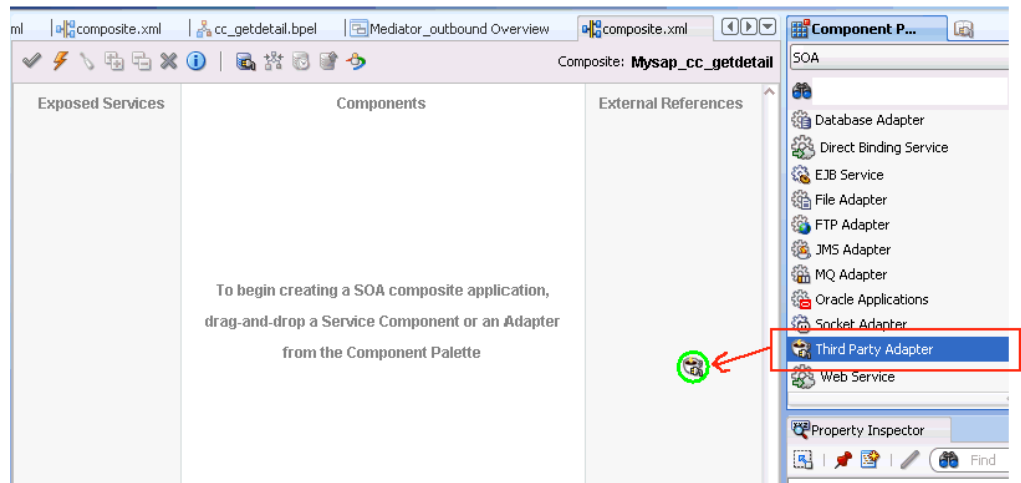
This section describes how to define a Mediator outbound process, which consists of the following stages:

1. Configuring a Third Party Adapter Service Component
2. Configuring an Outbound Mediator Process Component
3. Configuring the Routing Rules

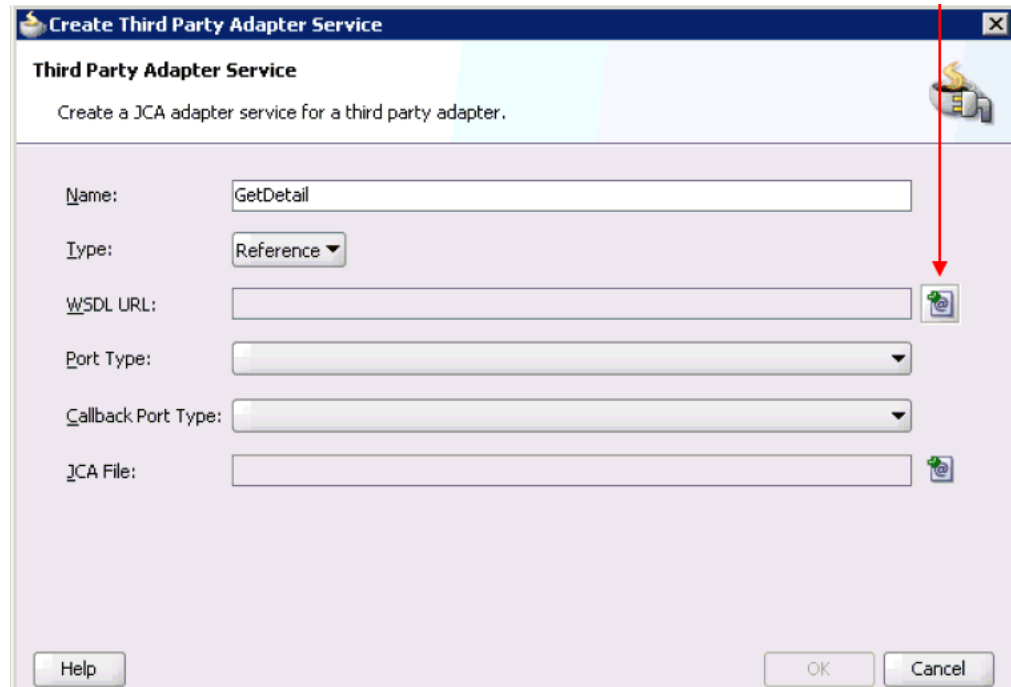
### Configuring a Third Party Adapter Service Component

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.

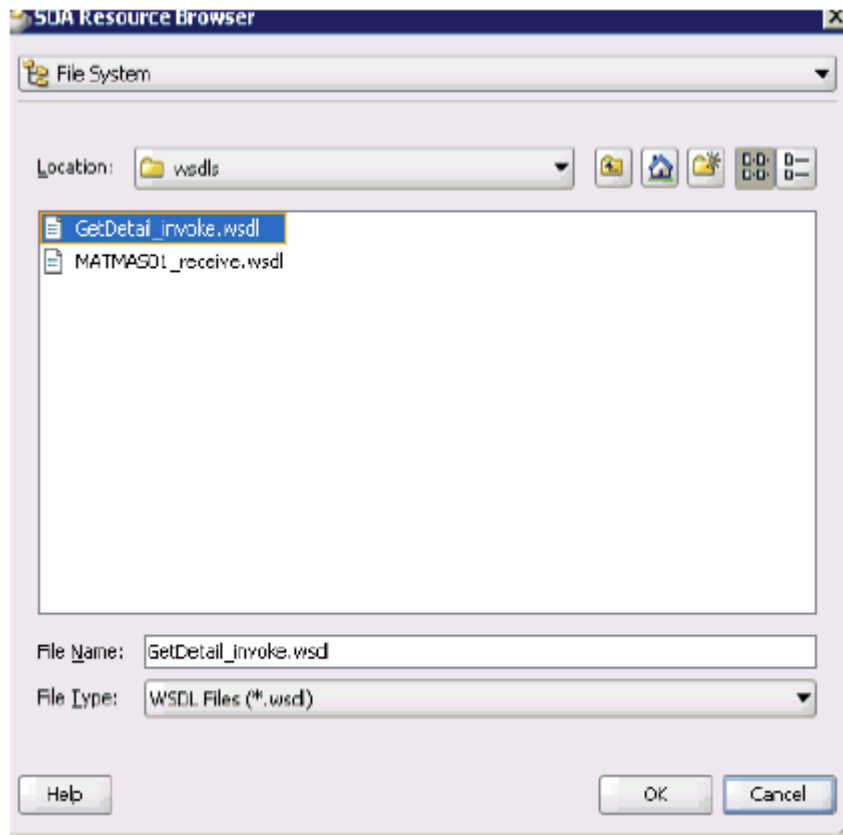


The Create Third Party Adapter Service dialog is displayed.



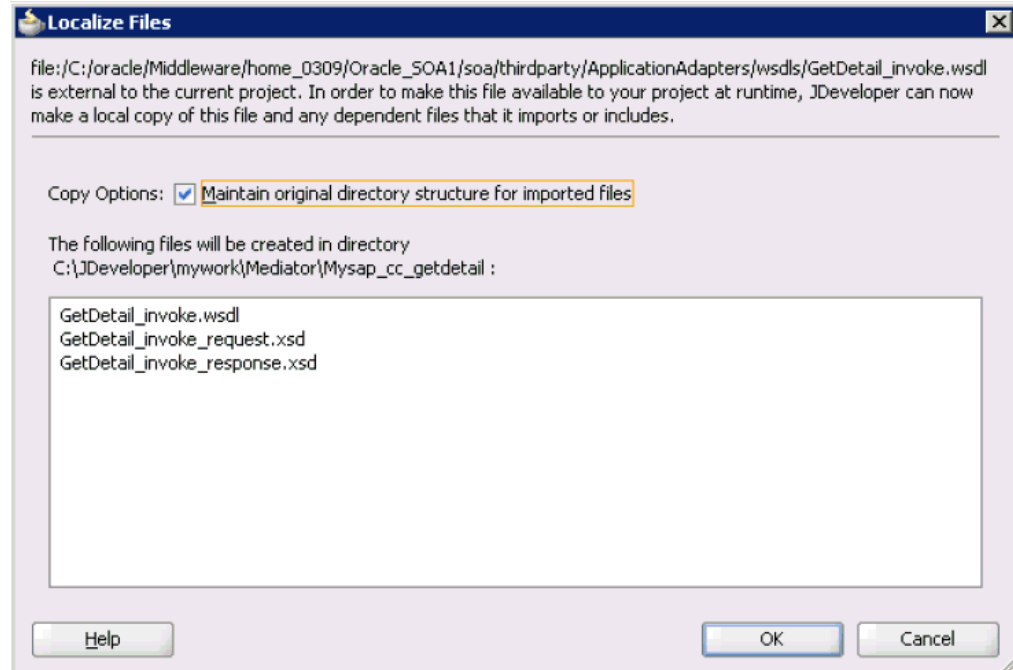
2. Enter a name for the third party adapter service.
3. Ensure that **Reference** is selected from the Type drop-down list (default).
4. Click the **Find existing WSDLs** icon, which is located to the right of the WSDL URL field.

The SOA Resource Browser dialog is displayed.



5. Browse and select an outbound WSDL file from the following directory:  
C:\oracle\Middleware\home\_GA\Oracle\_  
SOA1\soa\thirdparty\ApplicationAdapters\wsdl
6. Click **OK**.  
The Localize Files dialog is displayed.

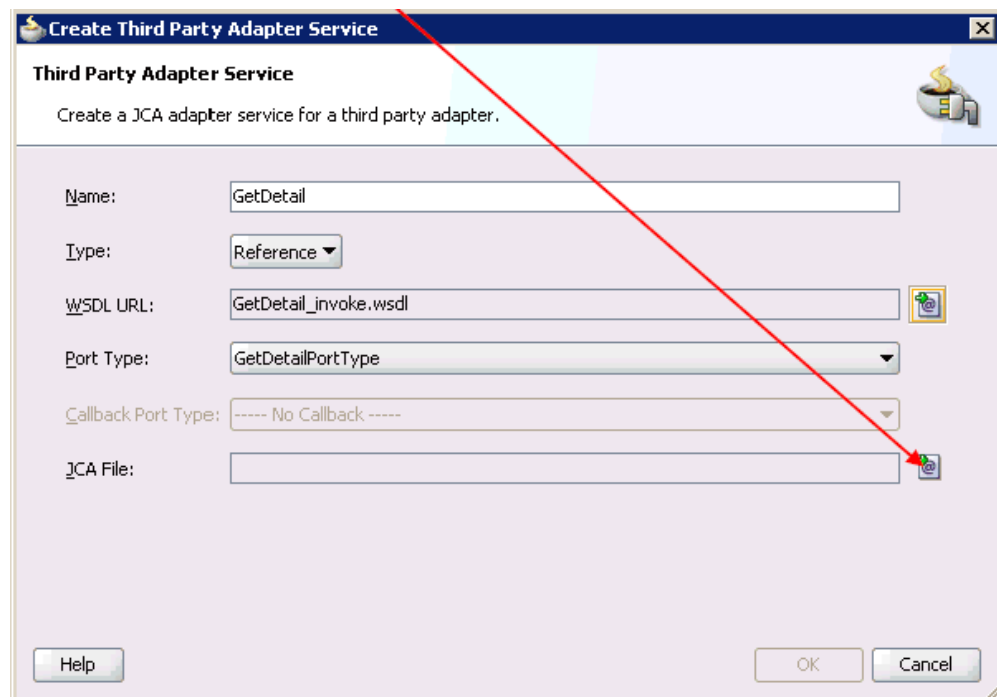




7. Click **OK**.

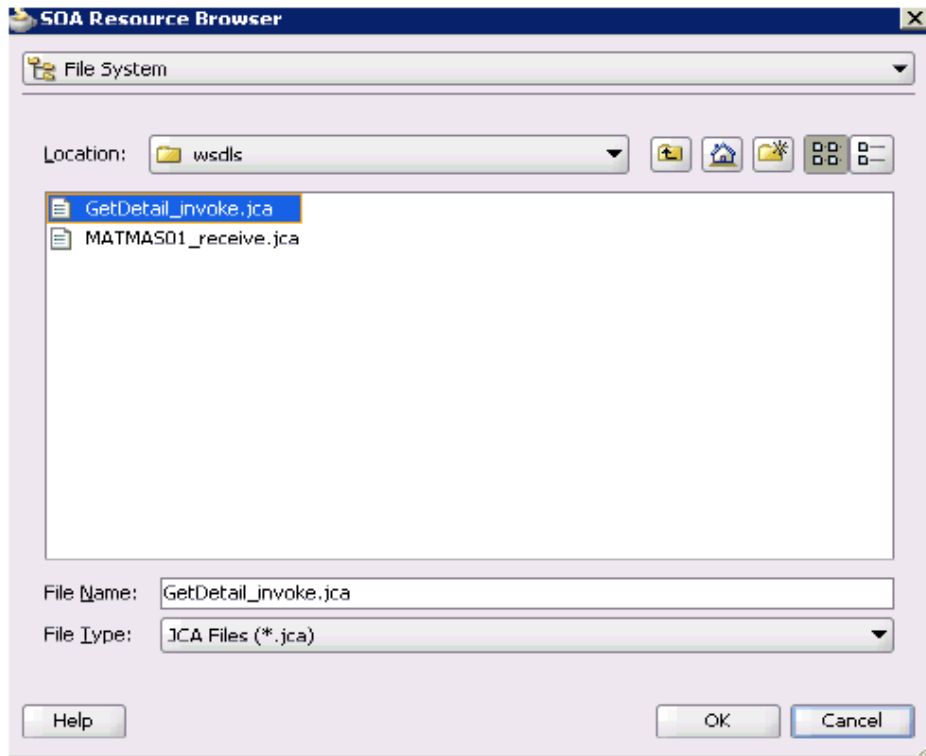
The outbound WSDL file and associated request and response XML schema files (.xsd) are imported to the project folder that has been created.

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



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

The SOA Resource Browser dialog is displayed.

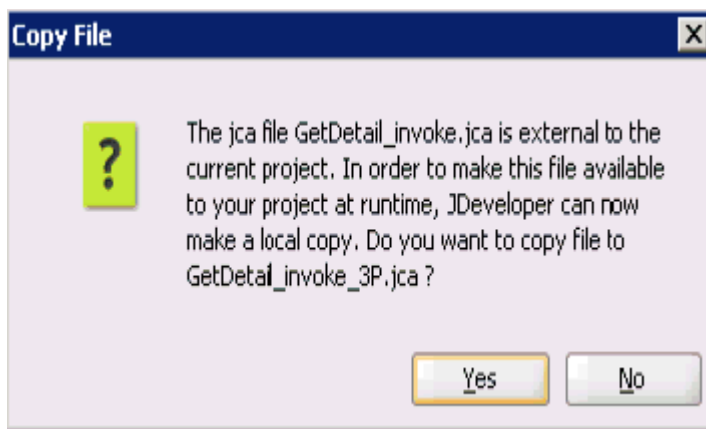


9. Browse and select the JCA properties file from the following directory:

C:\oracle\Middleware\home\_GA\Oracle\_  
SOA1\soa\thirdparty\ApplicationAdapters\wsdls

10. Click **OK**.

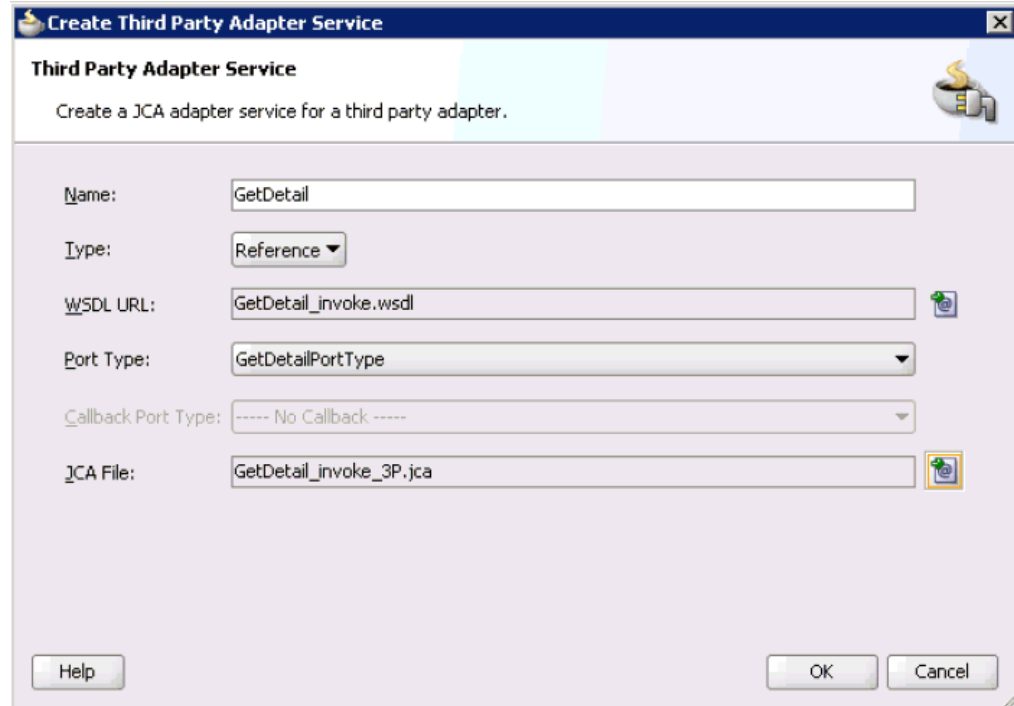
The following message is displayed.



11. Click **Yes**.

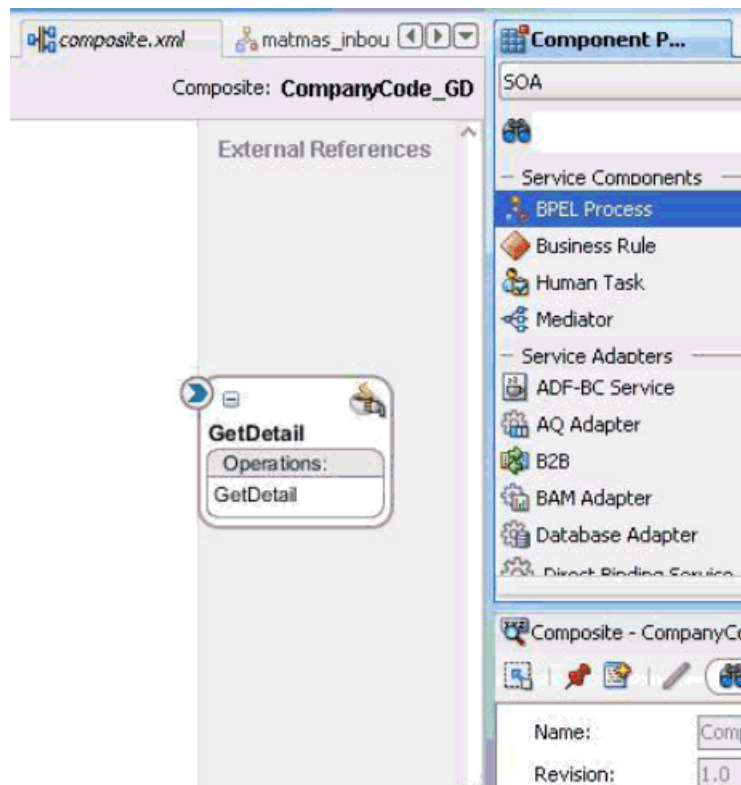
A copy of the JCA properties file is made in the project folder.

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



12. Click OK.

The third party adapter service component (GetDetail) is created in the External References pane, as shown in the following image.

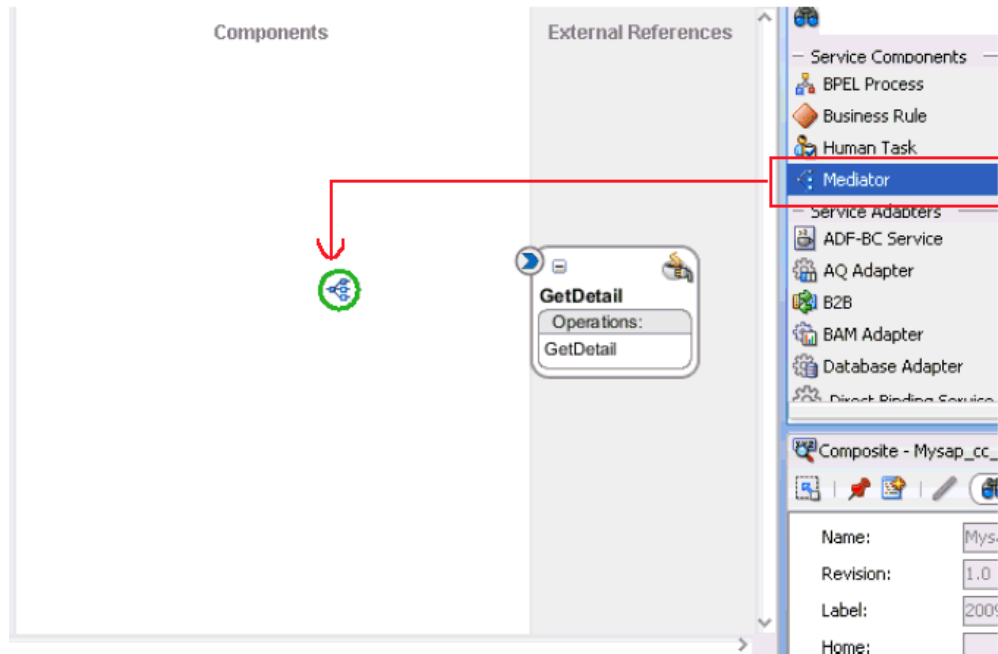


You are now ready to configure an outbound Mediator process component.

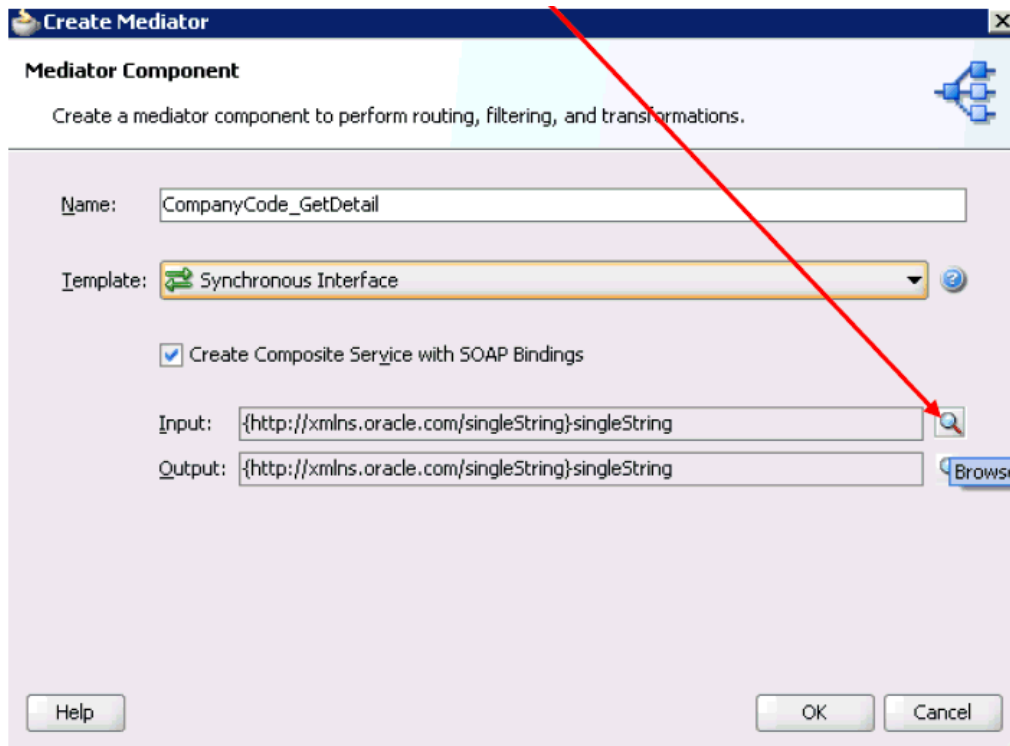
### Configuring an Outbound Mediator Process Component

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

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



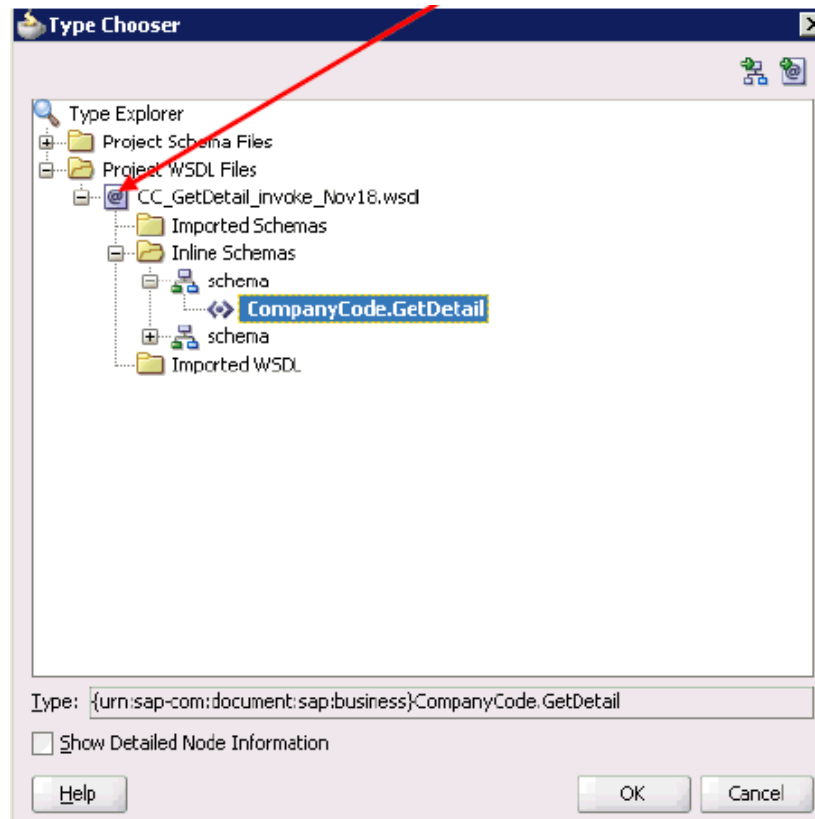
The Create Mediator dialog is displayed.



2. In the Name field, enter a name to identify the new outbound Mediator process component (for example, CompanyCode\_GetDetail).

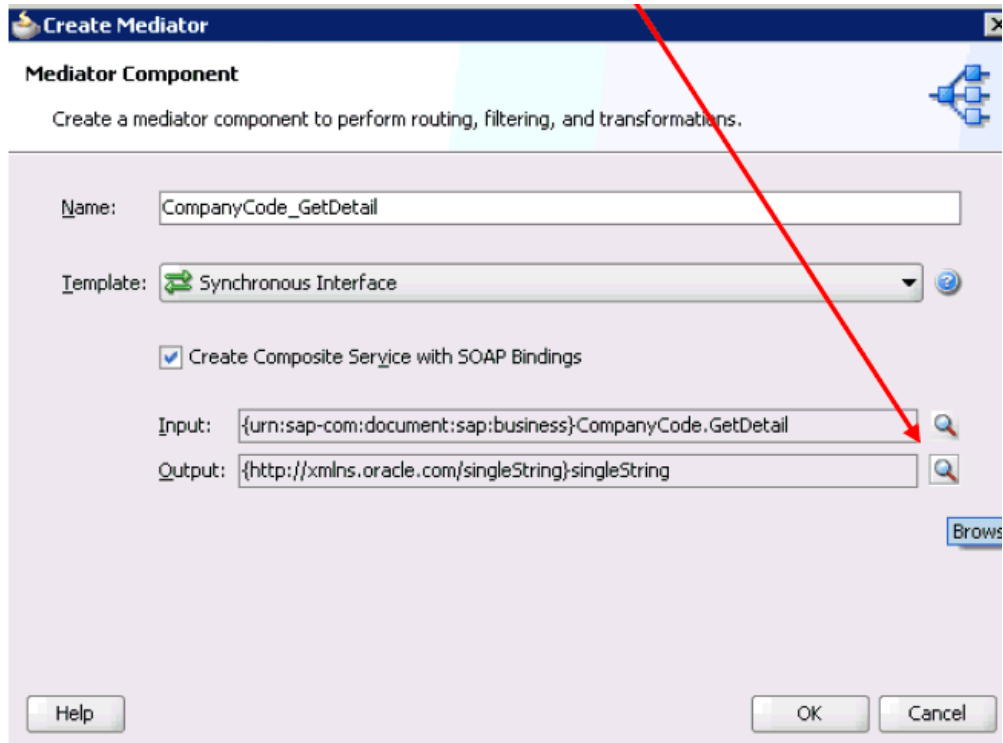
3. From the Template drop-down list, select **Synchronous Interface**.
4. Click the **Browse** icon, which is located to the right of the Input field to select the associated XML request schema file.

The Type Chooser dialog is displayed.



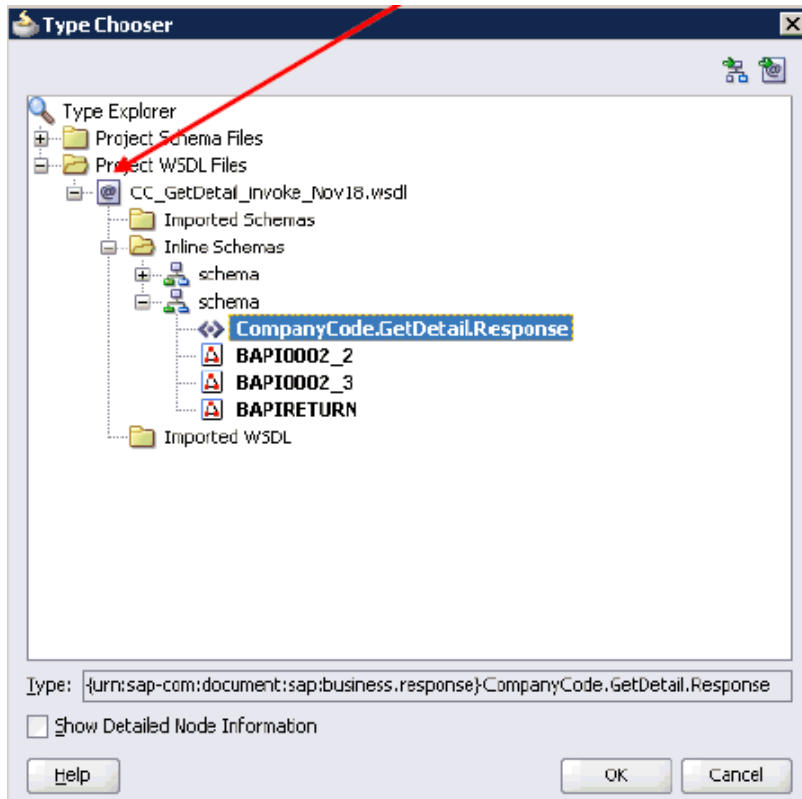
5. Expand **Project WSDL Files**, **CC\_GetDetail\_invoke\_Nov18.wsdl**, **Inline Schemas**, and select **CompanyCode.GetDetail**.
6. Click **OK**.

You are returned to the Create Mediator dialog.



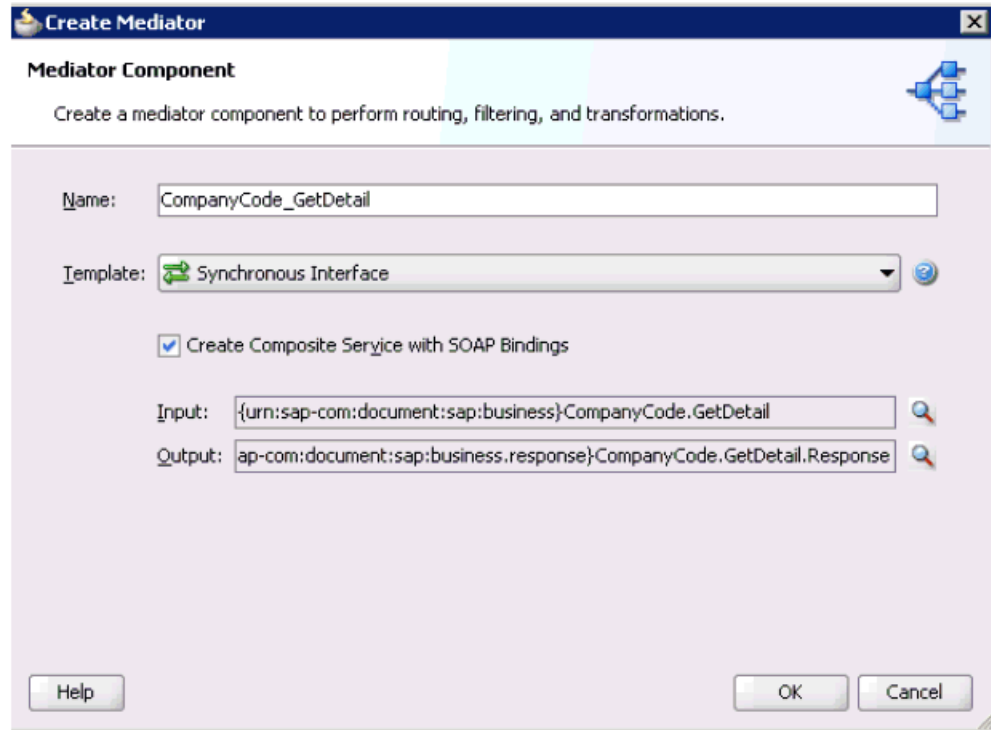
7. Click the **Browse** icon, which is located to the right of the Output field to select the associated XML response schema file.

The Type Chooser dialog is displayed.

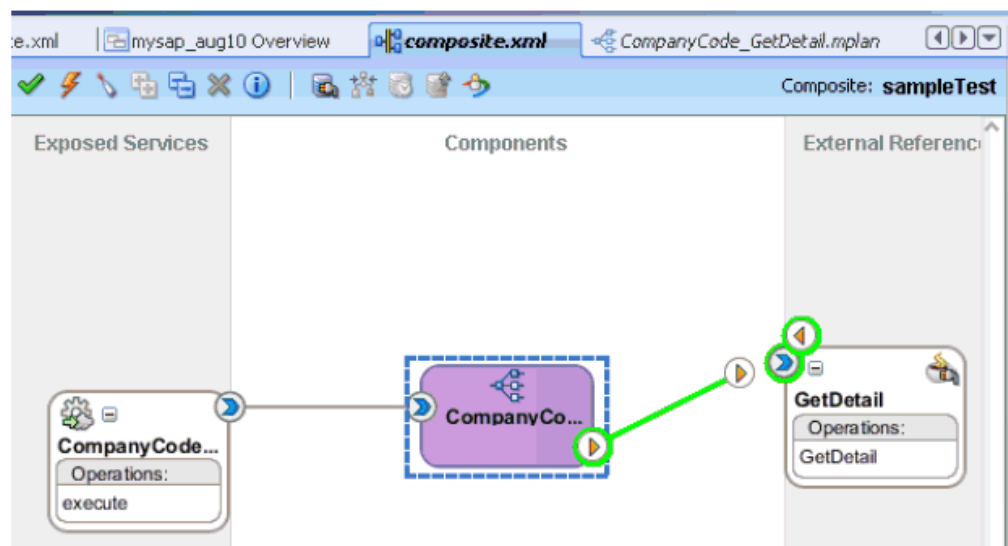


8. Expand **Project WSDL Files, CC\_GetDetail\_invoke\_Nov18.wsdl, Inline Schemas, schema**, and select **CompanyCode.GetDetailResponse**.
9. Click **OK**.

You are returned to the Create Mediator dialog.



10. Click **OK**.
11. Create a connection between the outbound Mediator process component (CompanyCode\_GetDetail) and the third party adapter service component (GetDetail).

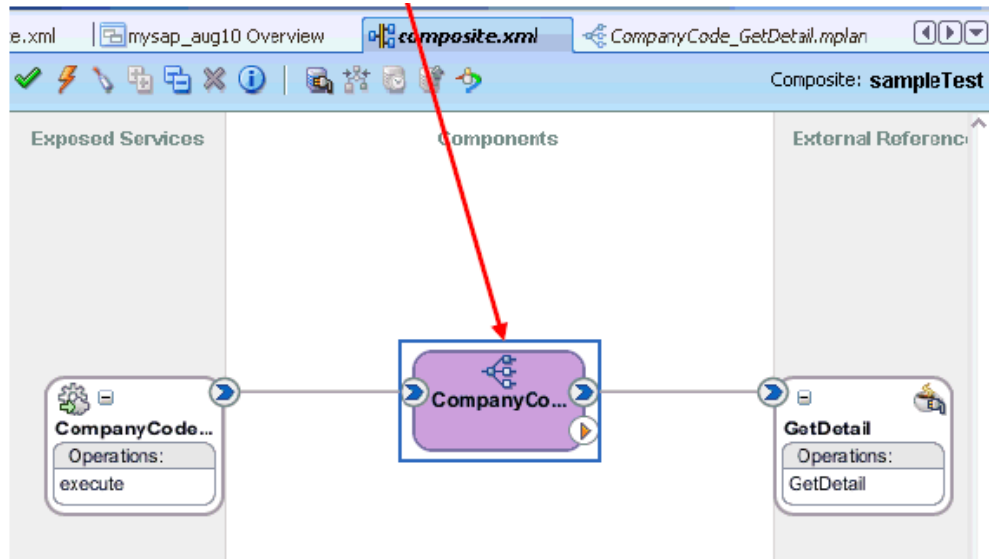


You are now ready to configure the routing rules.

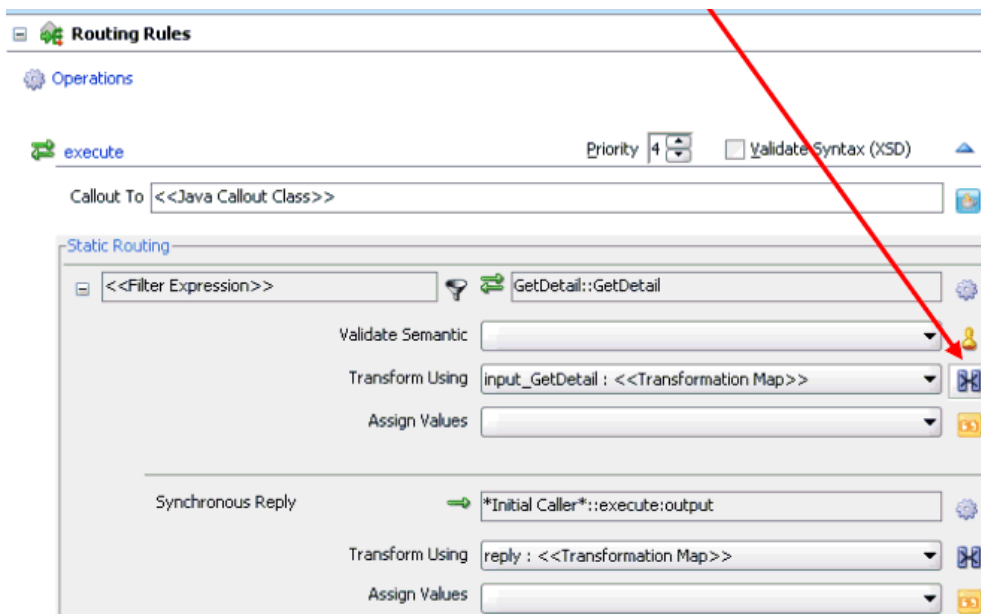
### Configuring the Routing Rules

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

1. Double-click the outbound Mediator process component (CompanyCode\_GetDetail) in the Components pane.



The Routing Rules dialog is displayed.



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

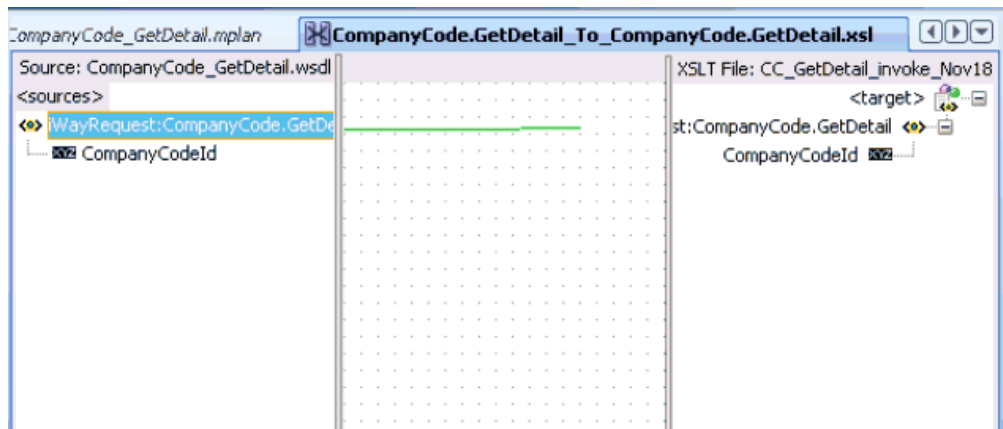
The Request Transformation Map dialog is displayed.





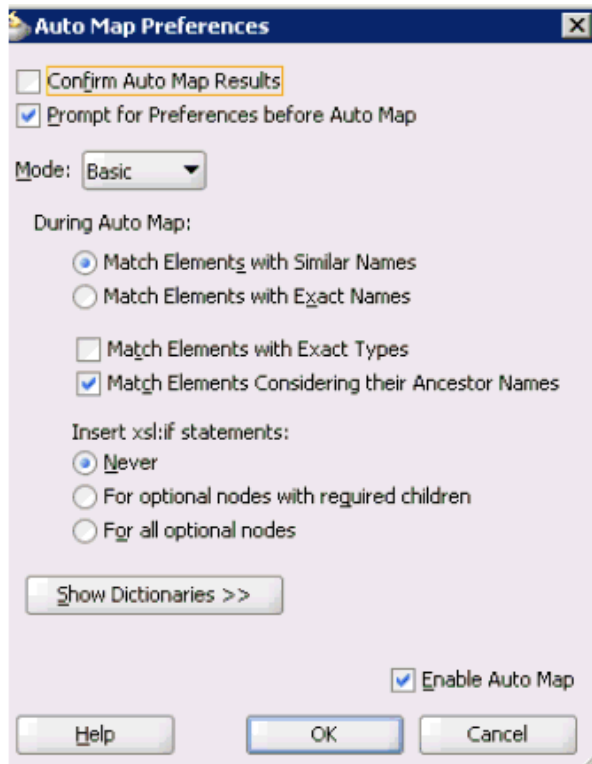
3. Select the **Create New Mapper File** option and click **OK**.

The following mapping page is displayed.



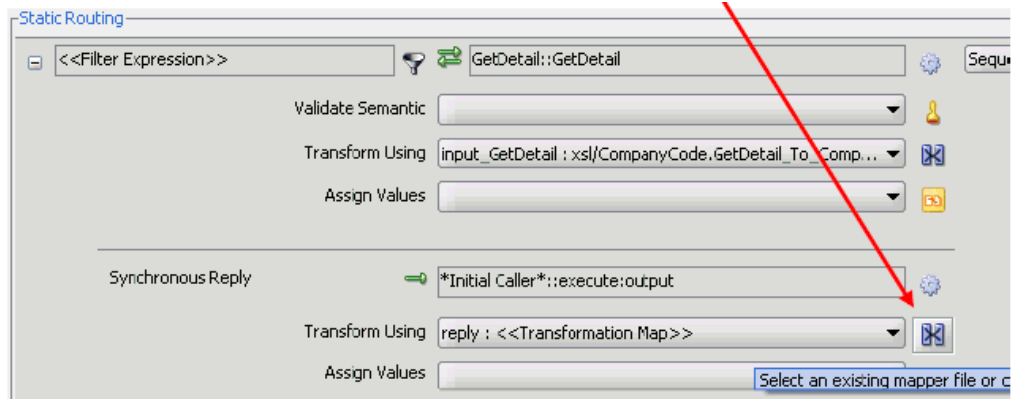
4. Map the **iWayRequest:CompanyCode.GetDetail** source element to the **iWayRequest:CompanyCode.GetDetail** target element.

The Auto Map Preferences dialog is displayed.



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

You are returned to the Routing Rules dialog.



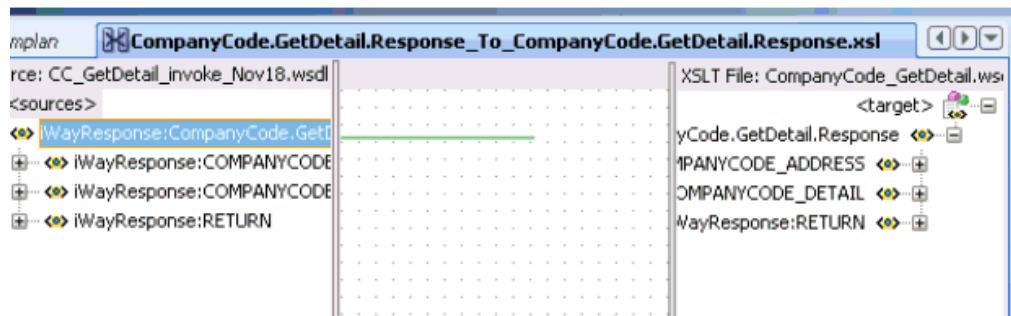
6. In the Synchronous Reply area, click the icon to the right of the Transform Using field.

The Reply Transformation Map dialog is displayed.



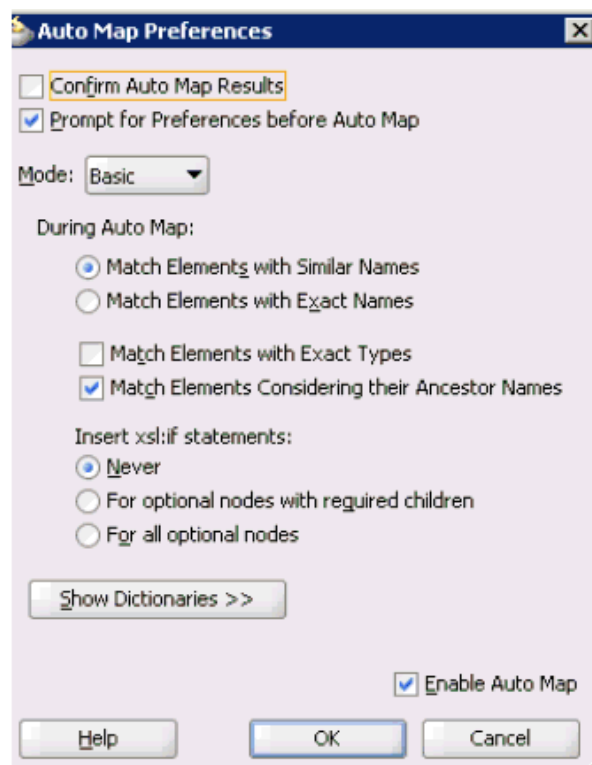
7. Select the **Create New Mapper File** option and click **OK**.

The following mapping page is displayed.



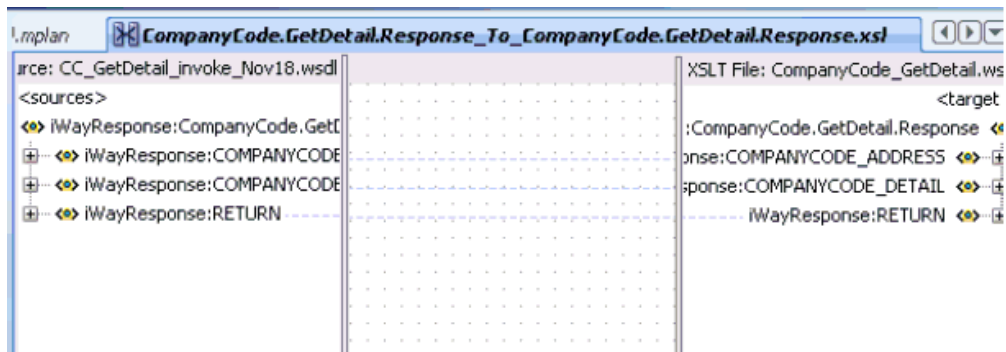
8. Map the **iWayResponse:CompanyCode.GetDetail.Response** source element to the **iWayResponse:CompanyCode.GetDetail.Response** target element.

The Auto Map Preferences dialog is displayed.



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

The mapping is complete, as shown in the following image.



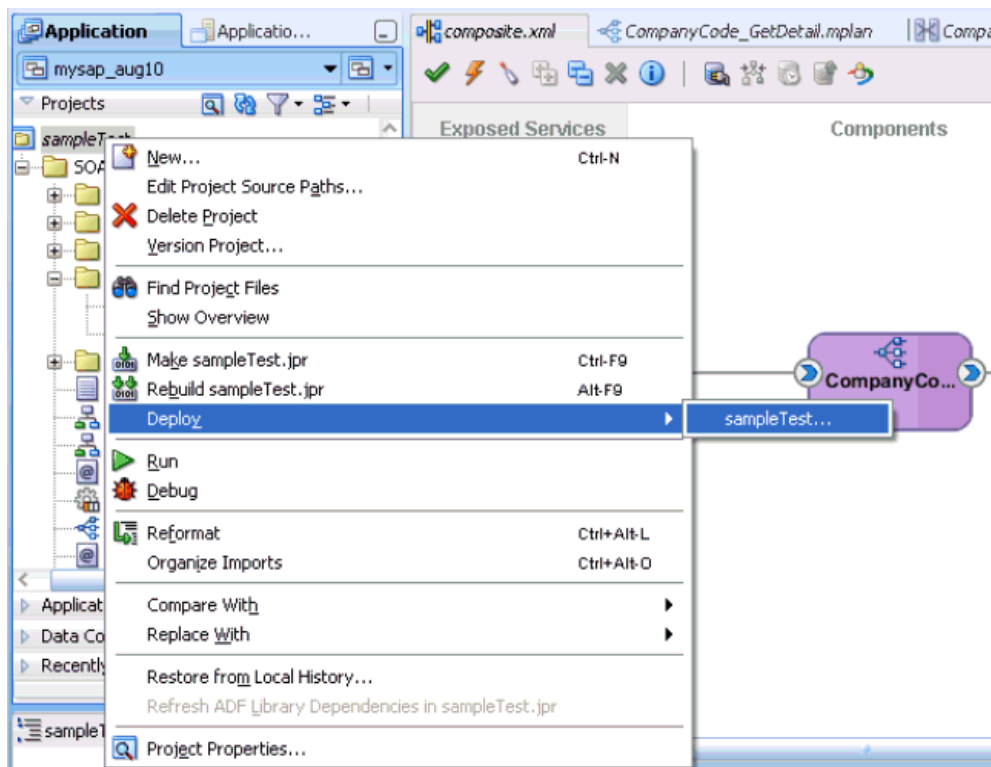
10. Click the **Save All** icon in the menu bar to save the new outbound Mediator process component that was configured.

You are now ready to deploy the Mediator outbound process.

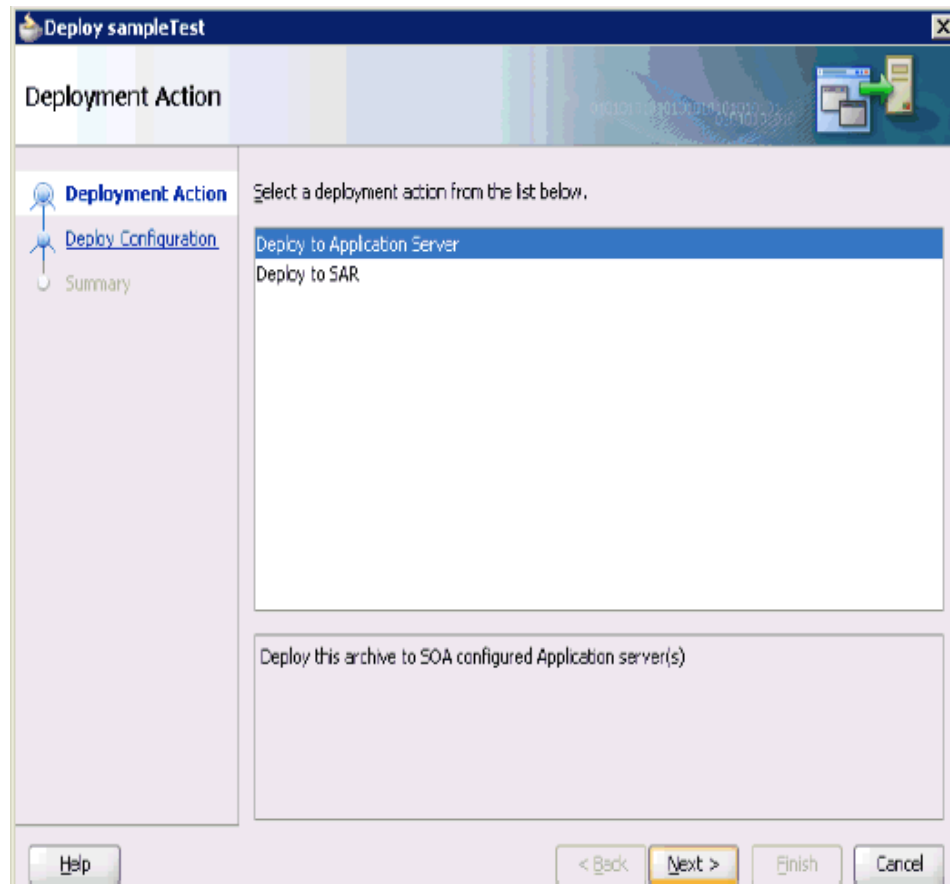
### 6.2.3 Deploying the Mediator Outbound Process

Perform the following steps to deploy the Mediator outbound process.

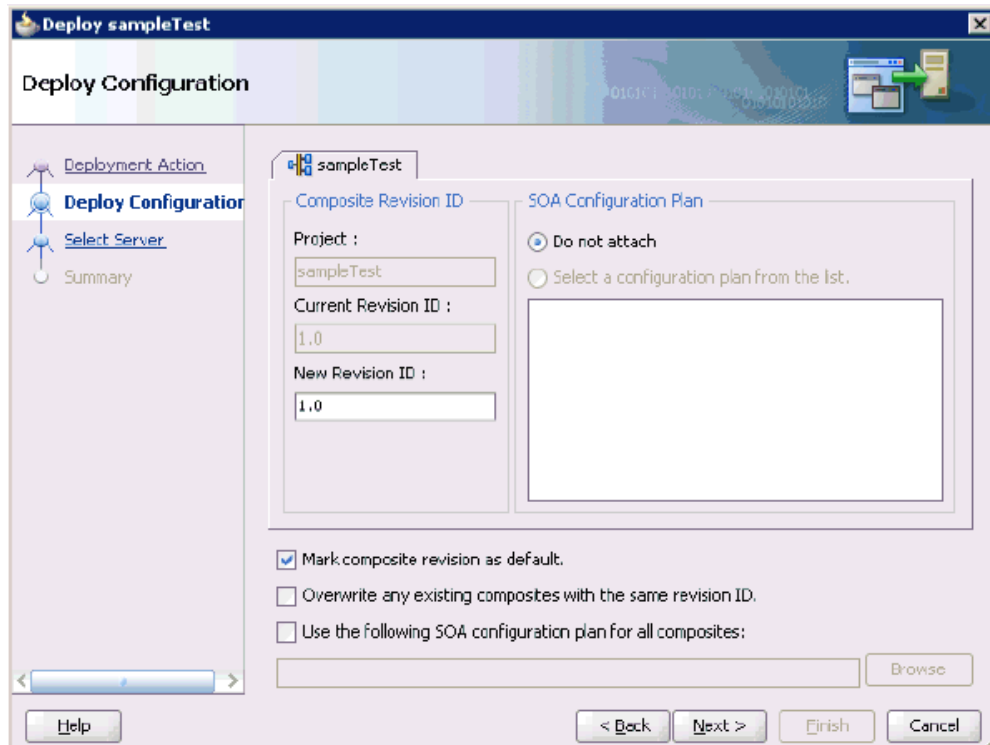
1. Right-click the project name in the left pane (for example, **sampleTest**), select **Deploy** from the context menu, and click **sampleTest**.



The Deployment Action page is displayed.

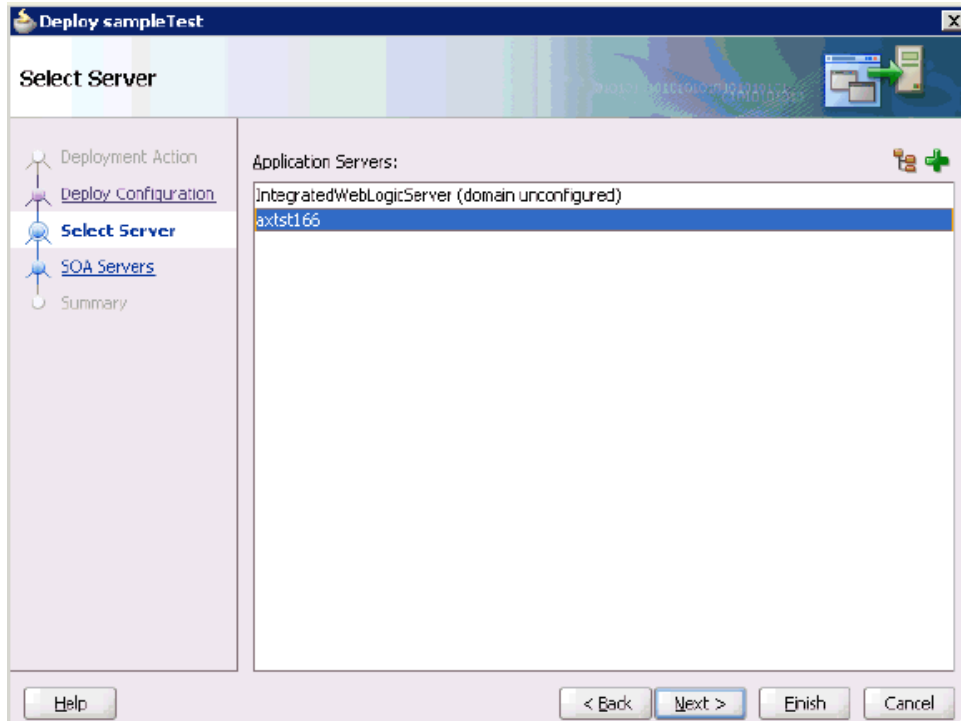


2. Ensure that **Deploy to Application Server** is selected.
3. Click **Next**.  
The Deploy Configuration page is displayed.



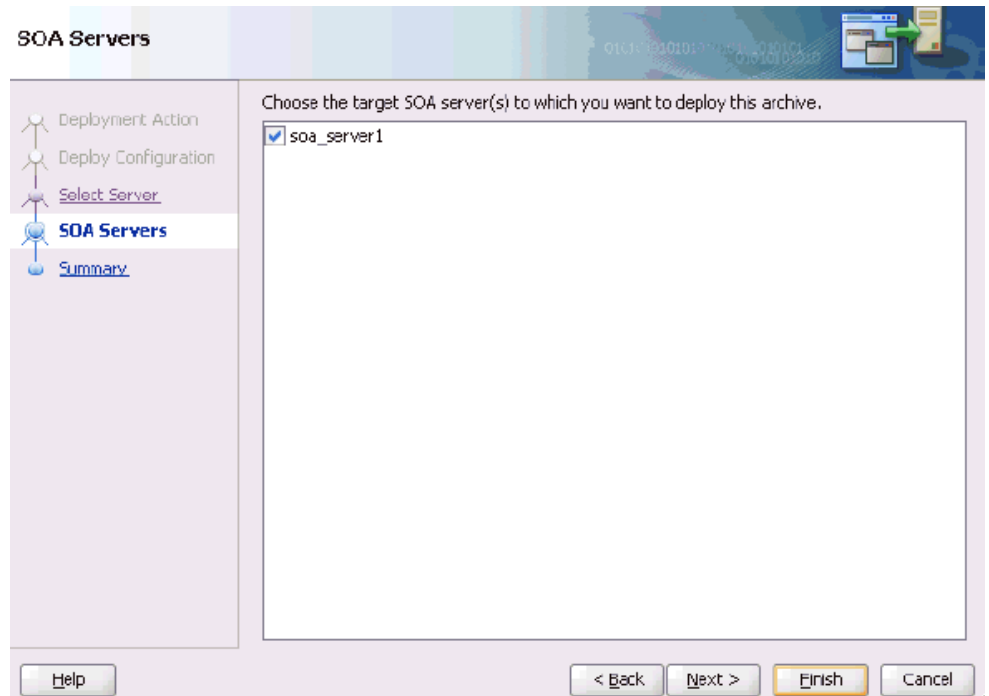
4. Leave the default values selected and click **Next**.

The Select Server page is displayed.



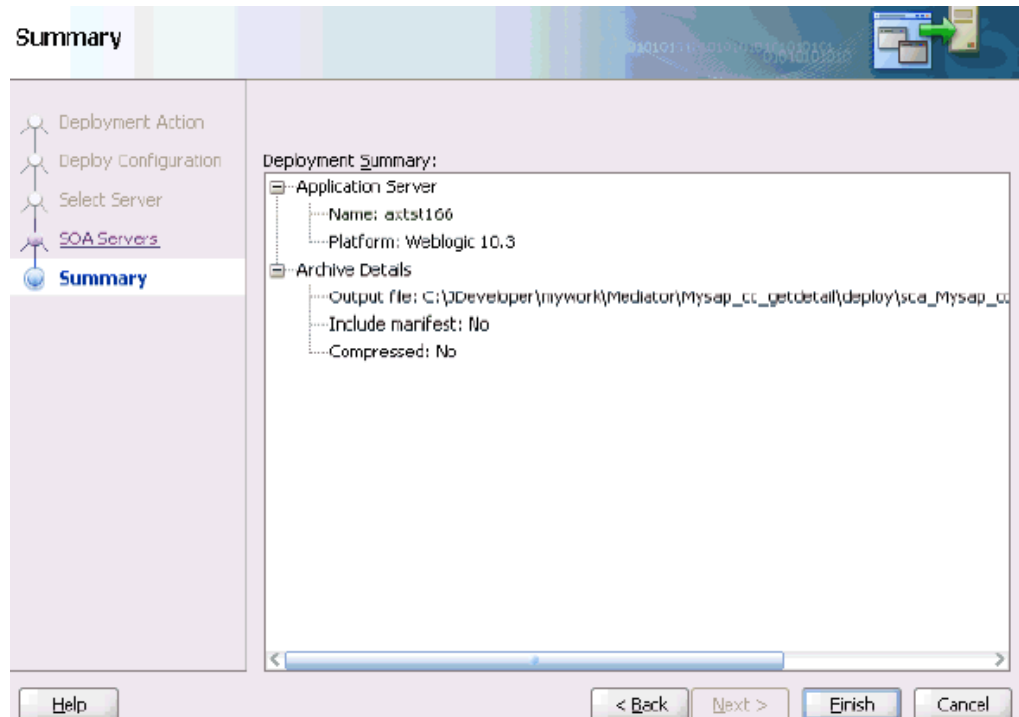
5. Select an available application server that was configured and click **Next**.

The SOA Servers page is displayed.



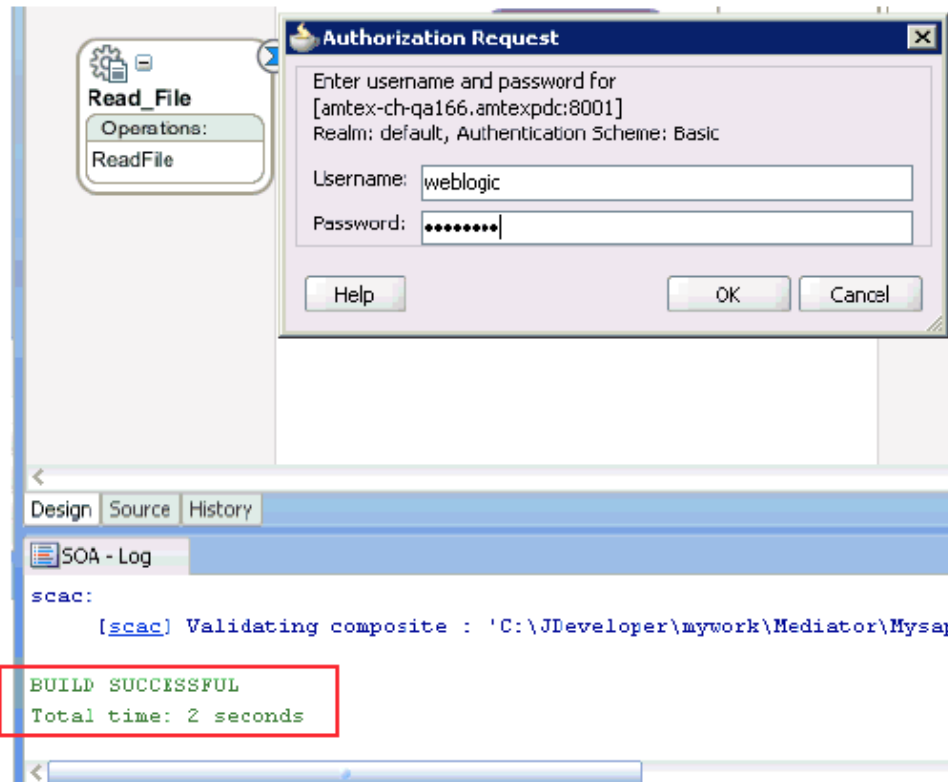
6. Select a target SOA server and click **Next**.

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.

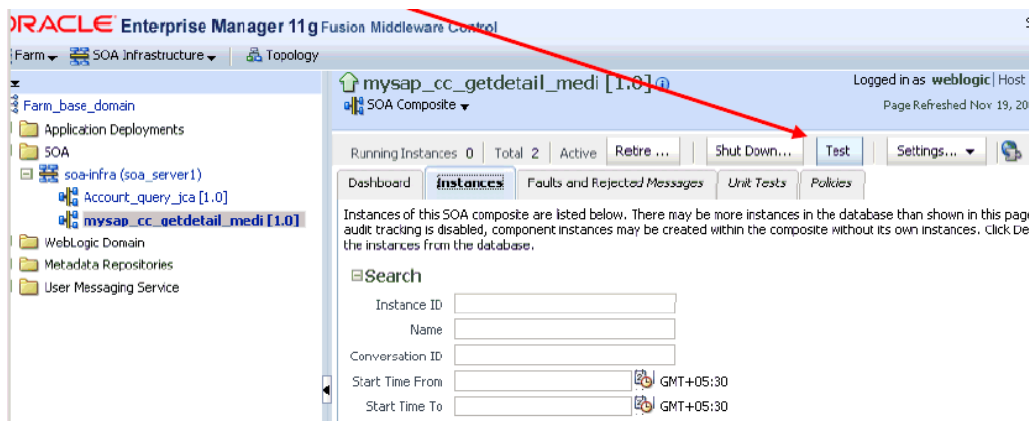


If an Authorization Request dialog is displayed during the deployment process, provide the required user name and password and click **OK**.

## 6.2.4 Invoking the Input XML Document in the Oracle Enterprise Manager Console

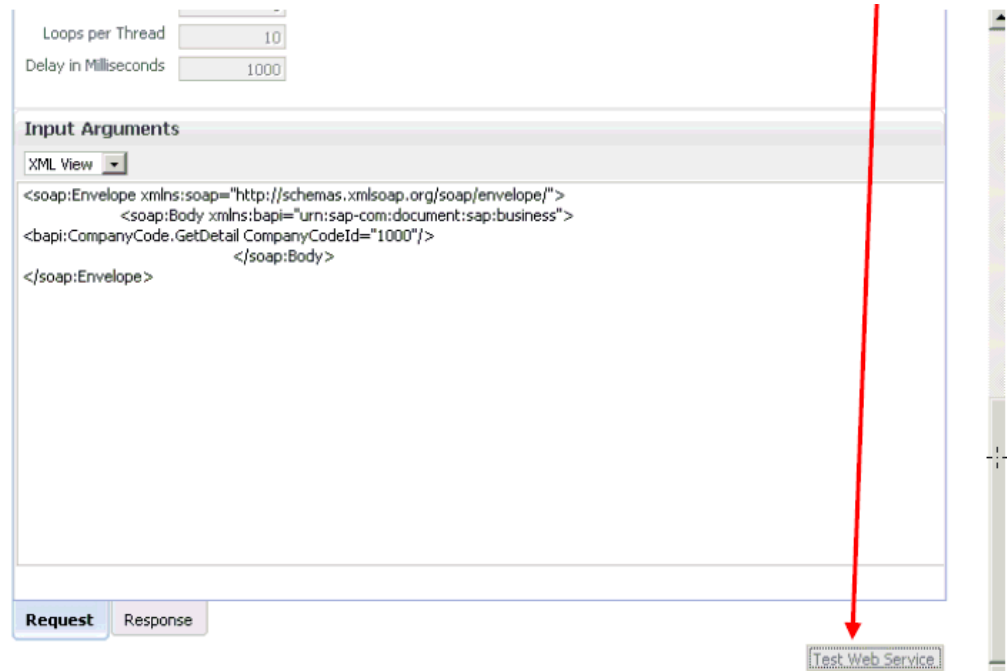
Perform the following steps to invoke the input XML document in the Oracle Enterprise Manager console.

1. Logon to the Oracle Enterprise Manager console.



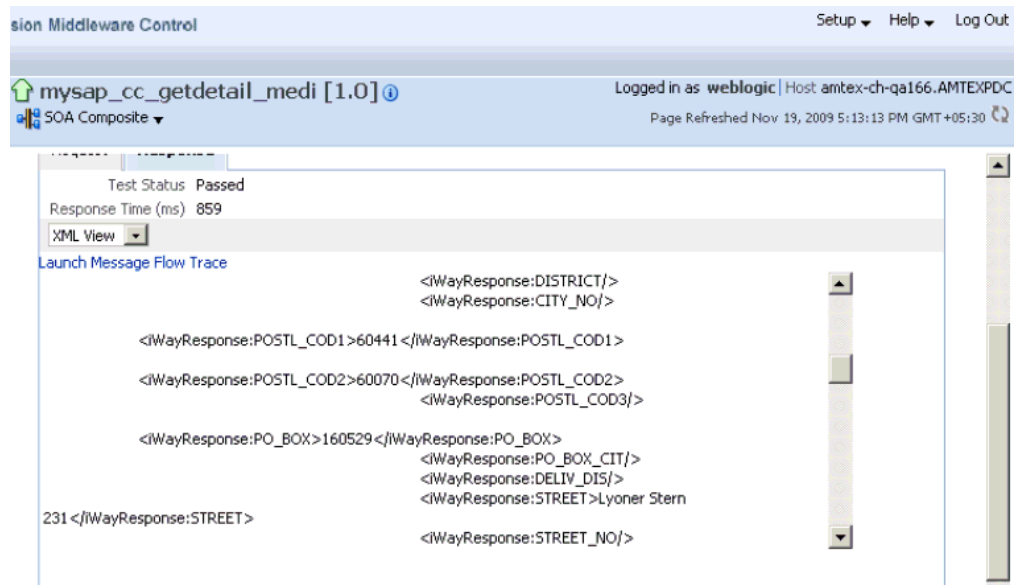
2. Expand your domain in the left pane followed by the **SOA** folder.
3. Select an available project in the left pane.
4. Click **Test**.
5. Click the **Request** tab.





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

The output response is received in the Oracle Enterprise Manager console, as shown in the following image.



### 6.3 Configuring a Mediator Inbound Process

The following example describes how to configure a Mediator inbound process to your SAP R/3 system, using a Mediator project in Oracle JDeveloper.

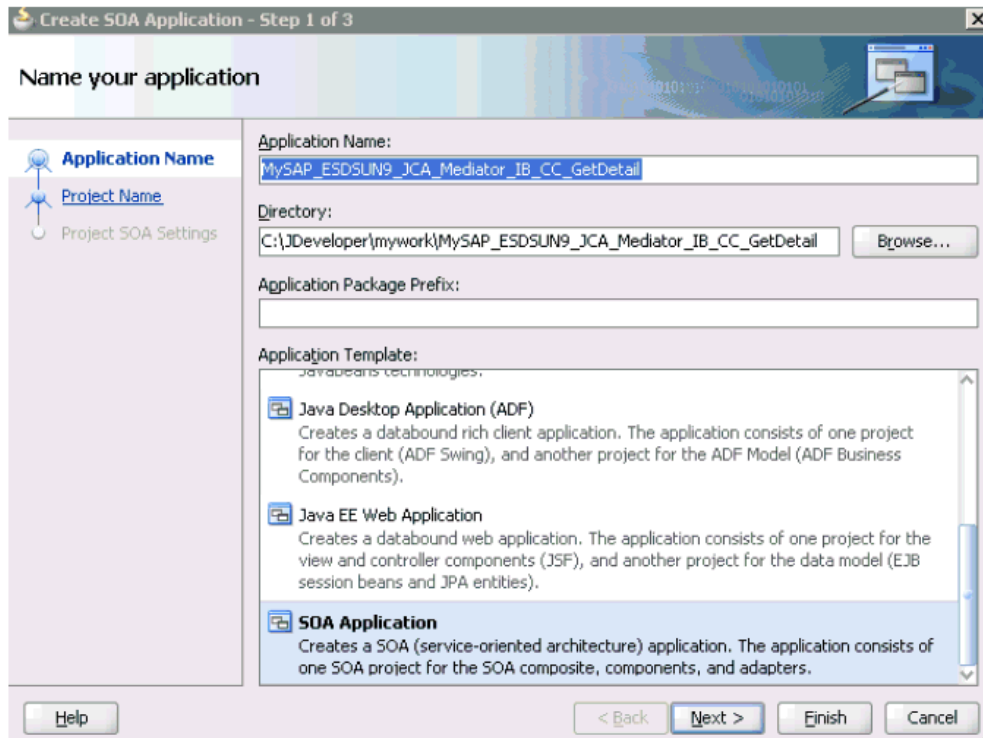
### Prerequisites

Before you design a Mediator inbound process, you must generate the respective WSDL file using Application Explorer. See ["Generating WSDL for Event Integration"](#) on page 5-35 for more information.

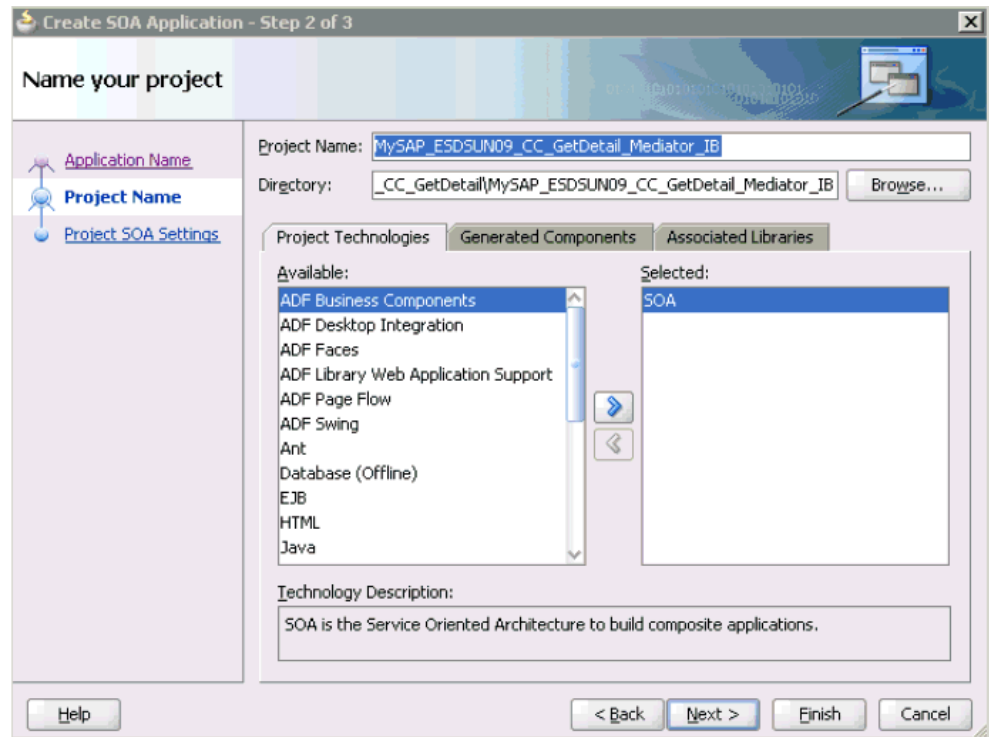
## 6.3.1 Creating an Empty Composite for SOA

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

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

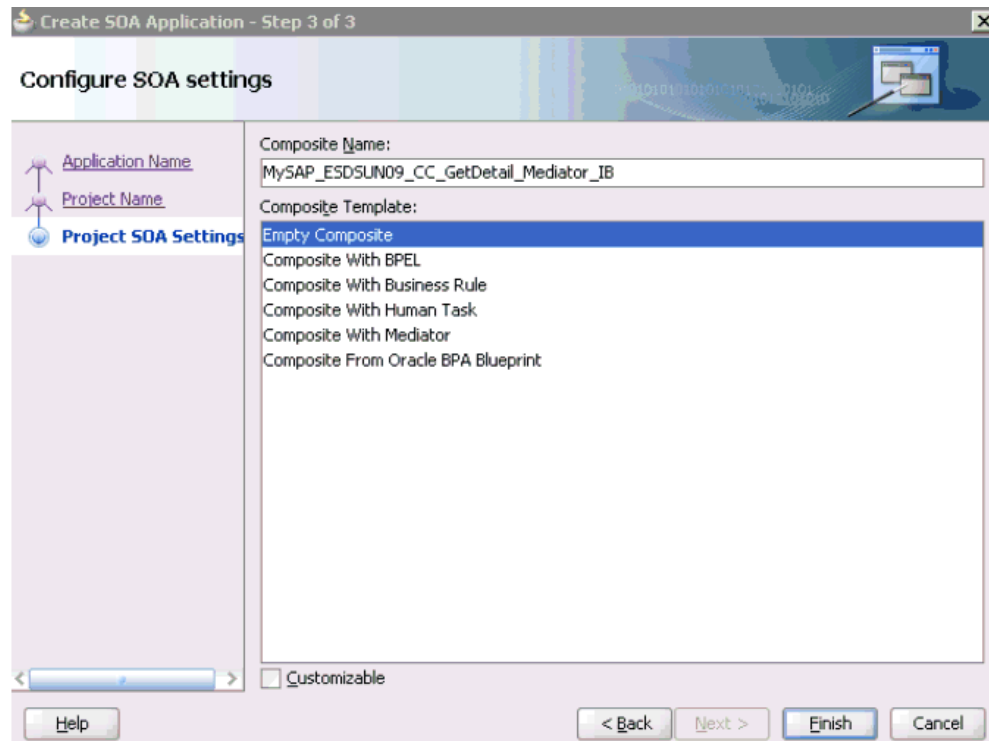


The Name your project page is displayed.



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**.

### 6.3.2 Defining a Mediator Inbound Process

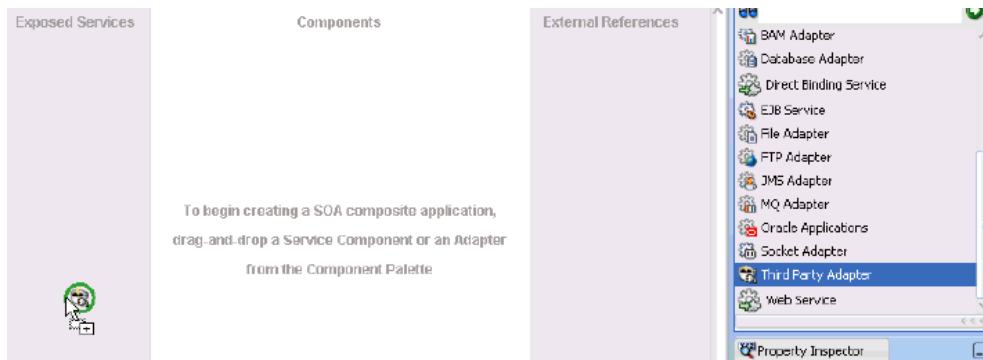
This section describes how to define a Mediator inbound process, which consists of the following stages:

1. Configuring a Third Party Adapter Service Component
2. Configuring an Inbound Mediator Process Component With a File Adapter
3. Configuring the Routing Rules

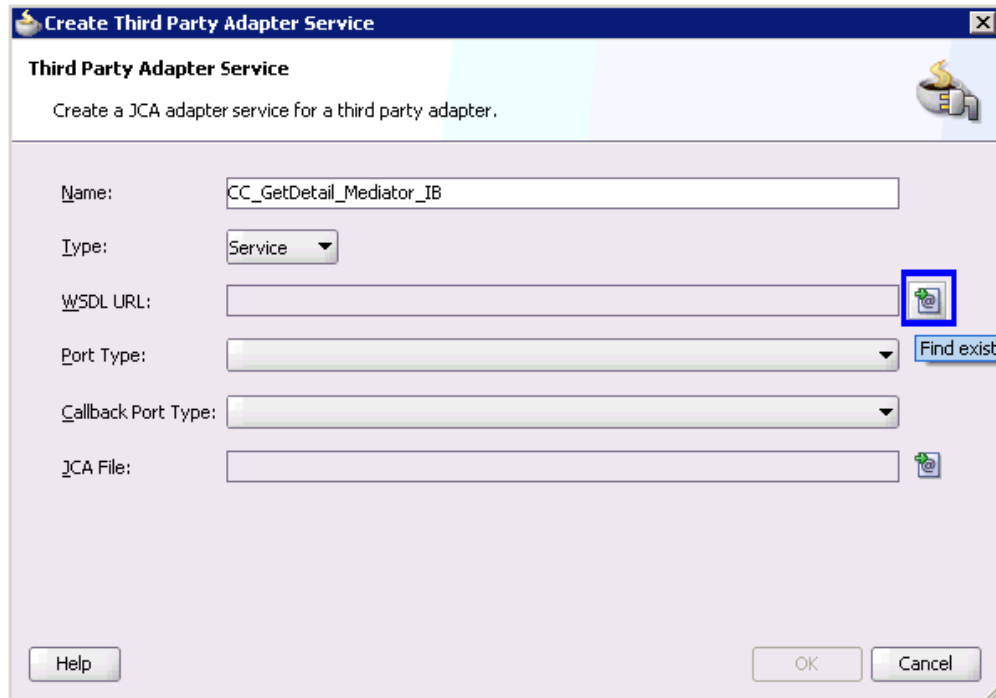
#### Configuring a Third Party Adapter Service Component

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.



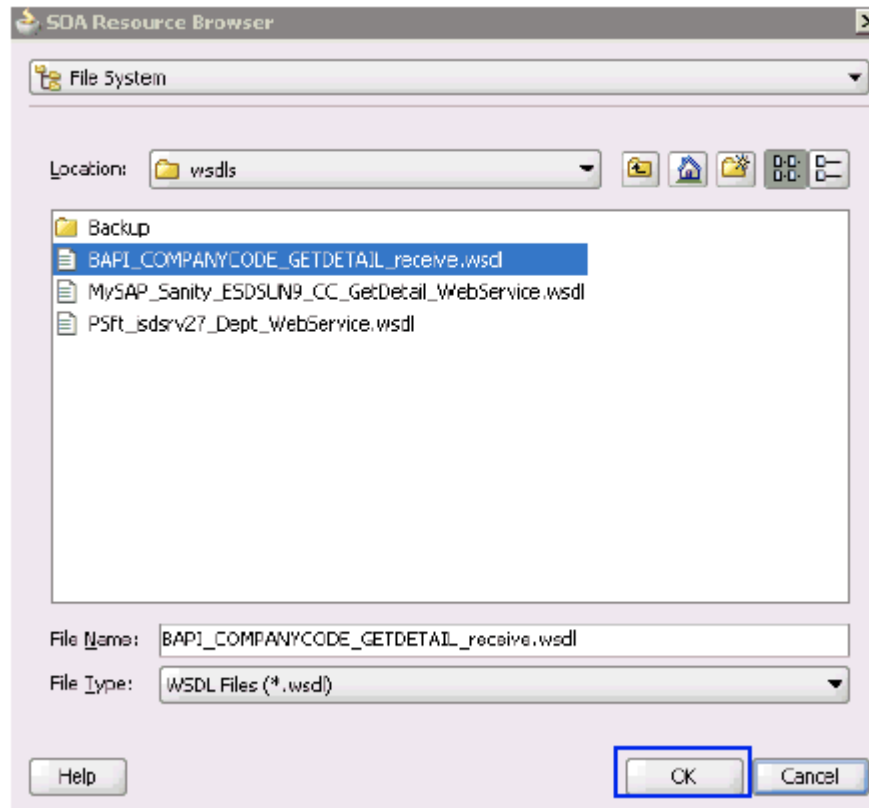
The Create Third Party Adapter Service dialog is displayed.



2. Enter a name for the third party adapter service (for example, CC\_GetDetail\_Mediator\_IB).
3. Ensure that **Service** is selected from the Type drop-down list (default).

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

The SOA Resource Browser dialog is displayed.

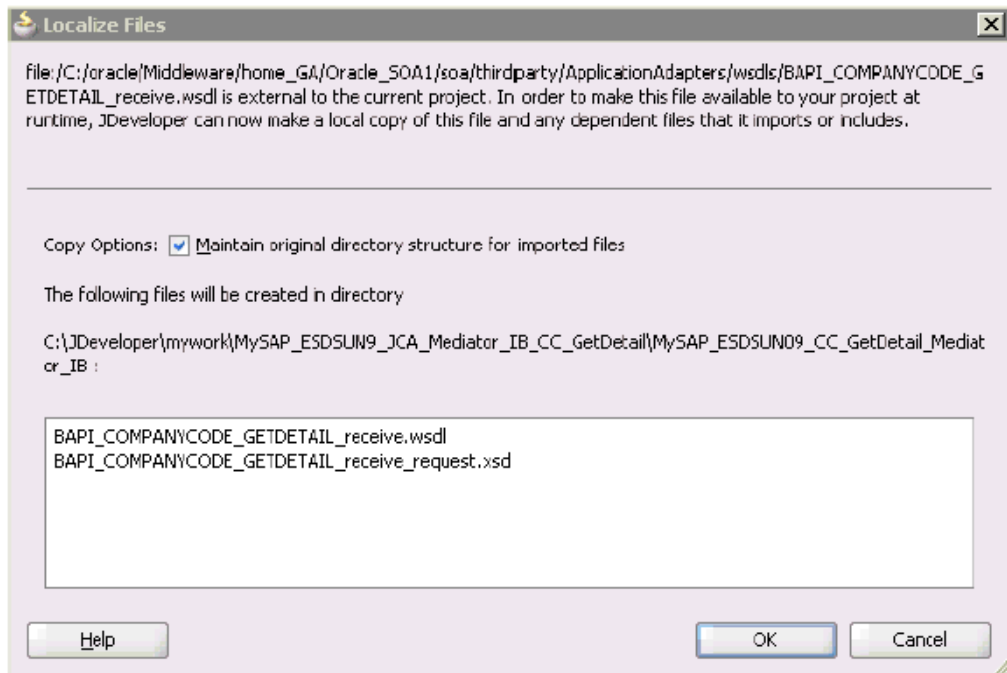


- Browse and select an inbound WSDL file from the following directory:

C:\oracle\Middleware\home\_GA\Oracle\_SOA1\soa\thirdparty\ApplicationAdapters\wsdls

- Click **OK**.

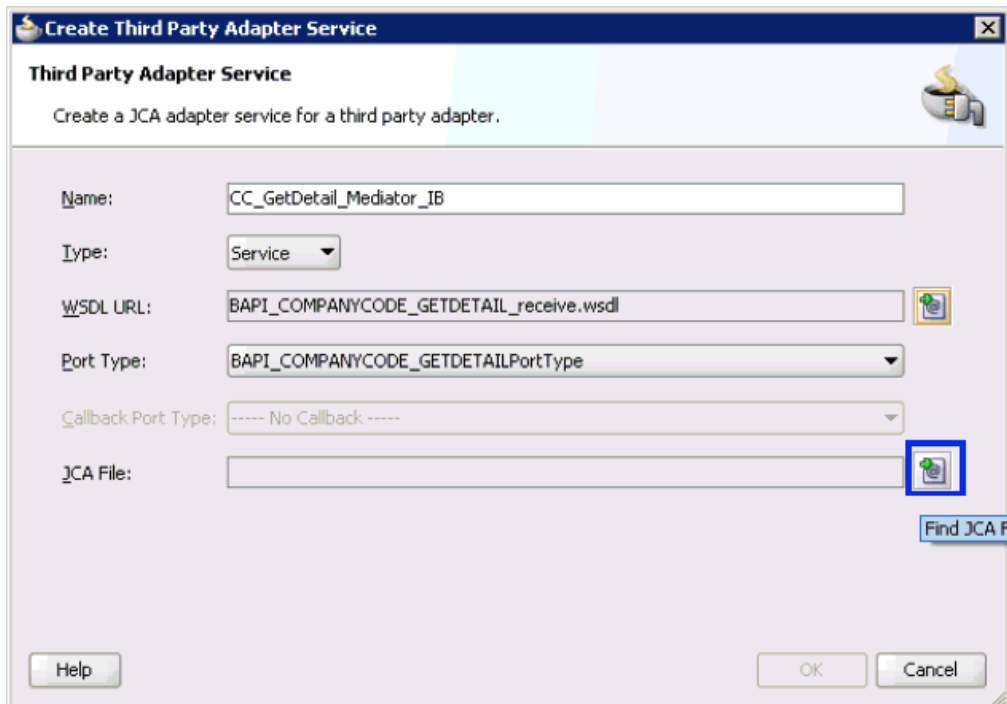
The Localize Files dialog is displayed.



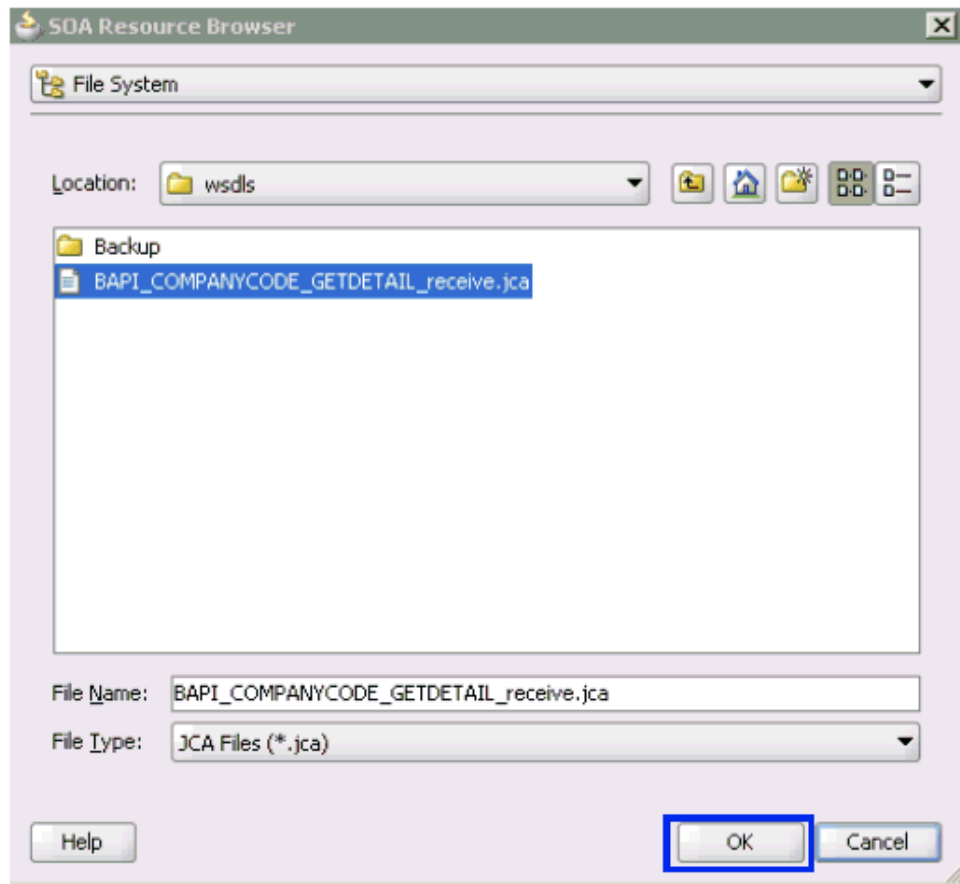
7. Click **OK**.

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 SOA Resource Browser dialog is displayed.

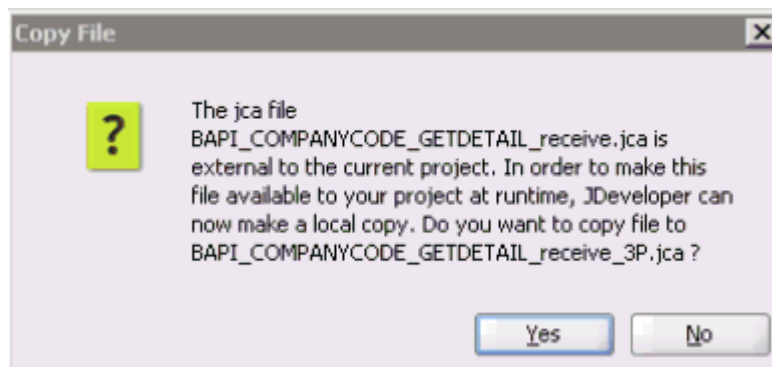


9. Browse and select the JCA properties file from the following directory:

C:\oracle\Middleware\home\_GA\Oracle\_  
SOA1\soa\thirdparty\ApplicationAdapters\wsdls

10. Click **OK**.

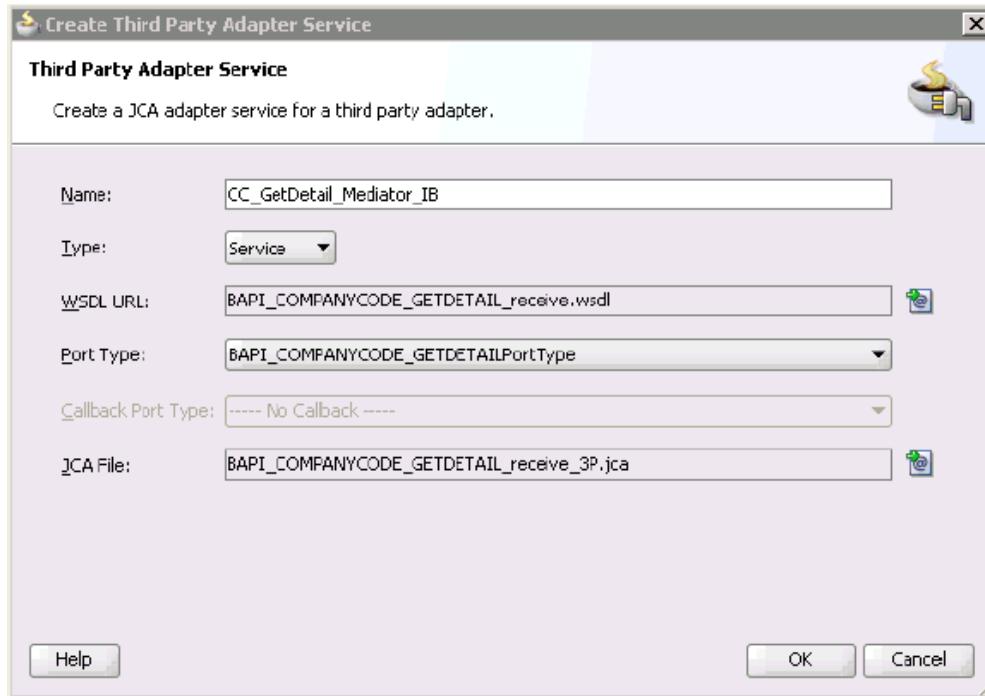
The following message is displayed.



11. Click **Yes**.

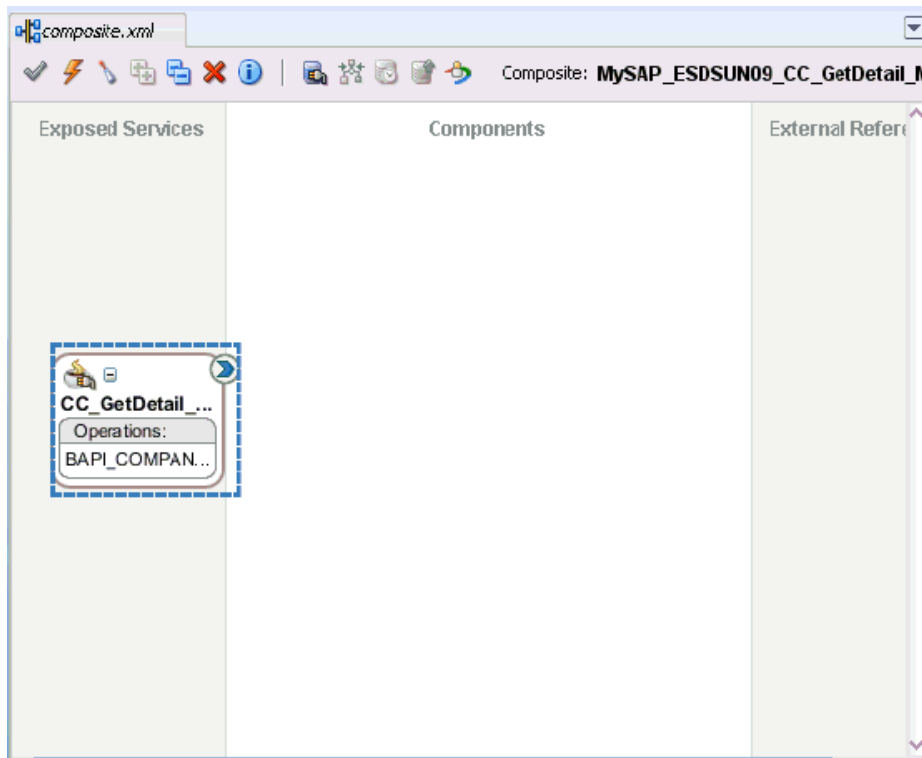
A copy of the JCA properties file is made in the project folder.

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



12. Click **OK**.

The third party adapter service component (CC\_GetDetail\_Mediator\_IB) is created in the Exposed Services pane, as shown in the following image.



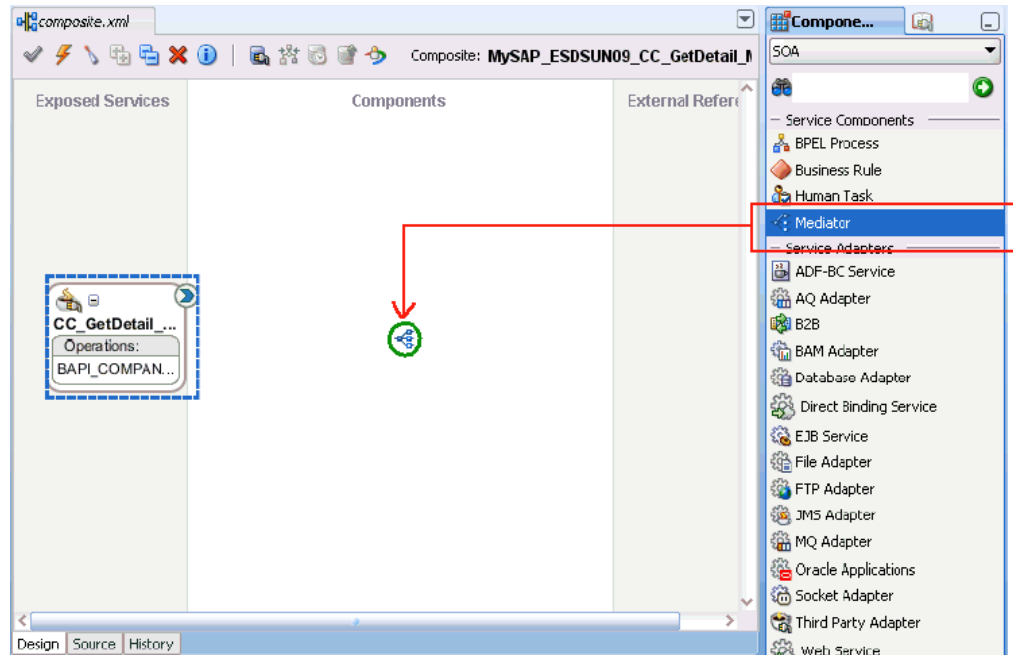
You are now ready to configure an inbound Mediator process component.



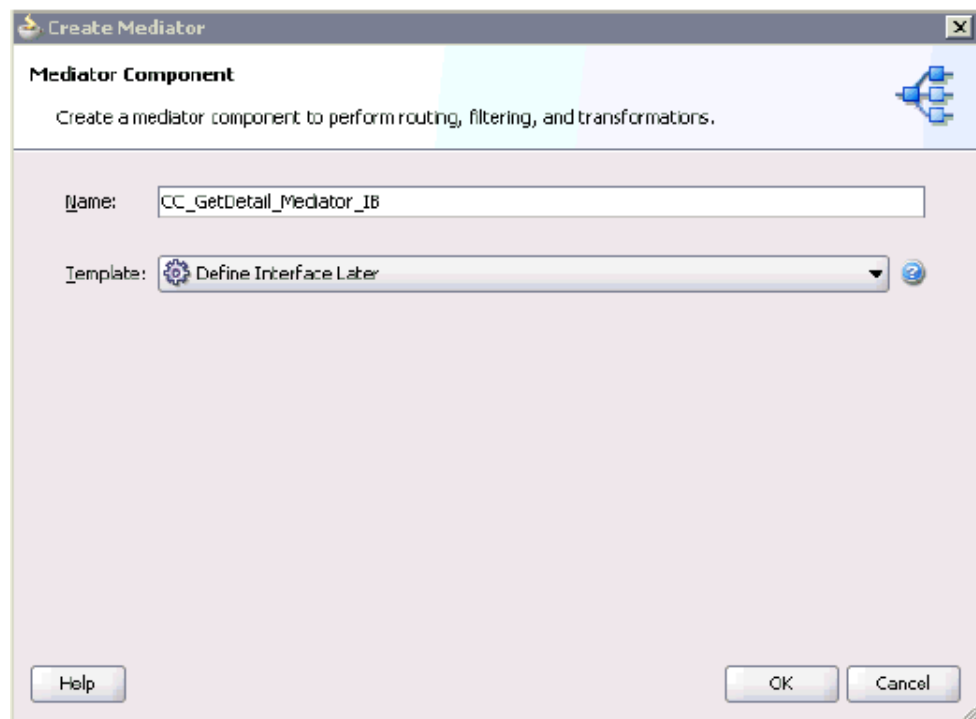
### Configuring an Inbound Mediator Process Component With a File Adapter

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.



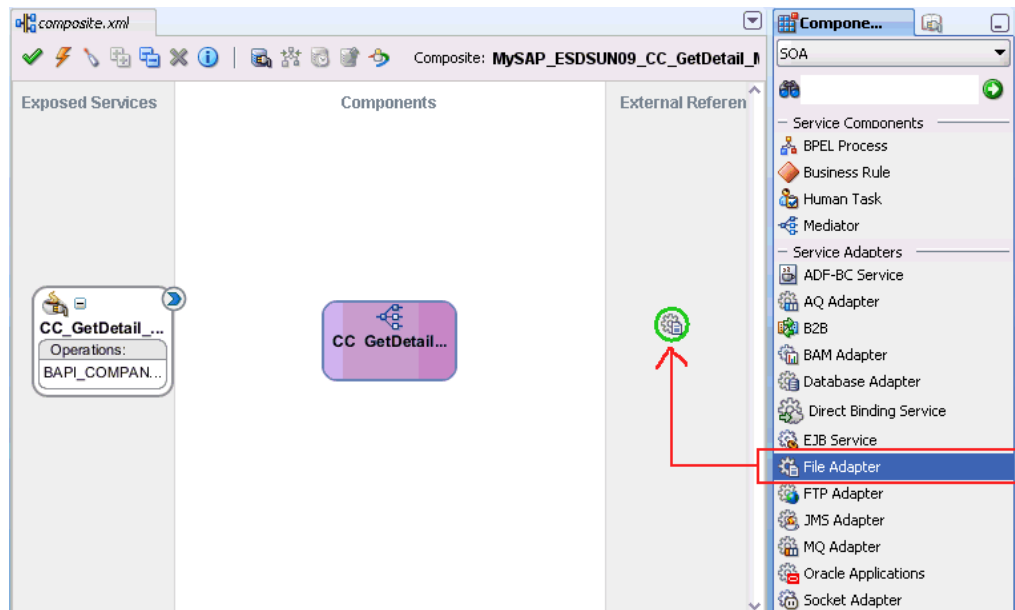
The Create Mediator dialog is displayed.



2. In the Name field, enter a name to identify the new inbound Mediator process component (for example, CC\_GetDetail\_Mediator\_IB).

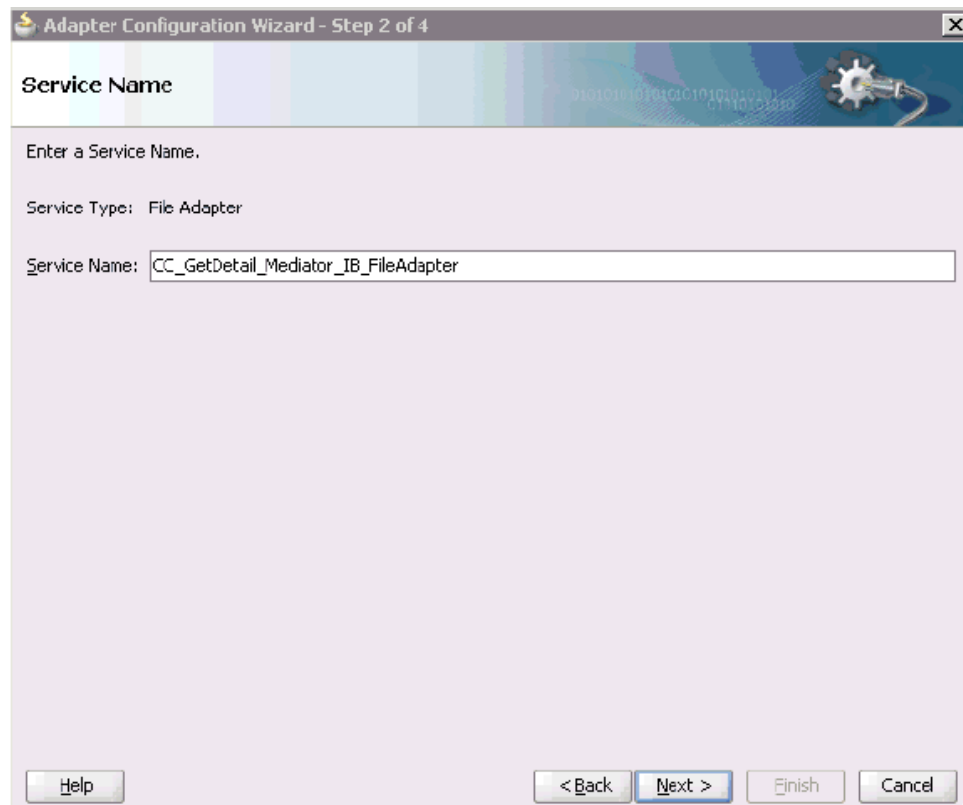
3. From the Template drop-down list, select **Define Interface Later**.
4. Click the **OK**.

The new Mediator process component is added to the Components pane, as shown in the following image.

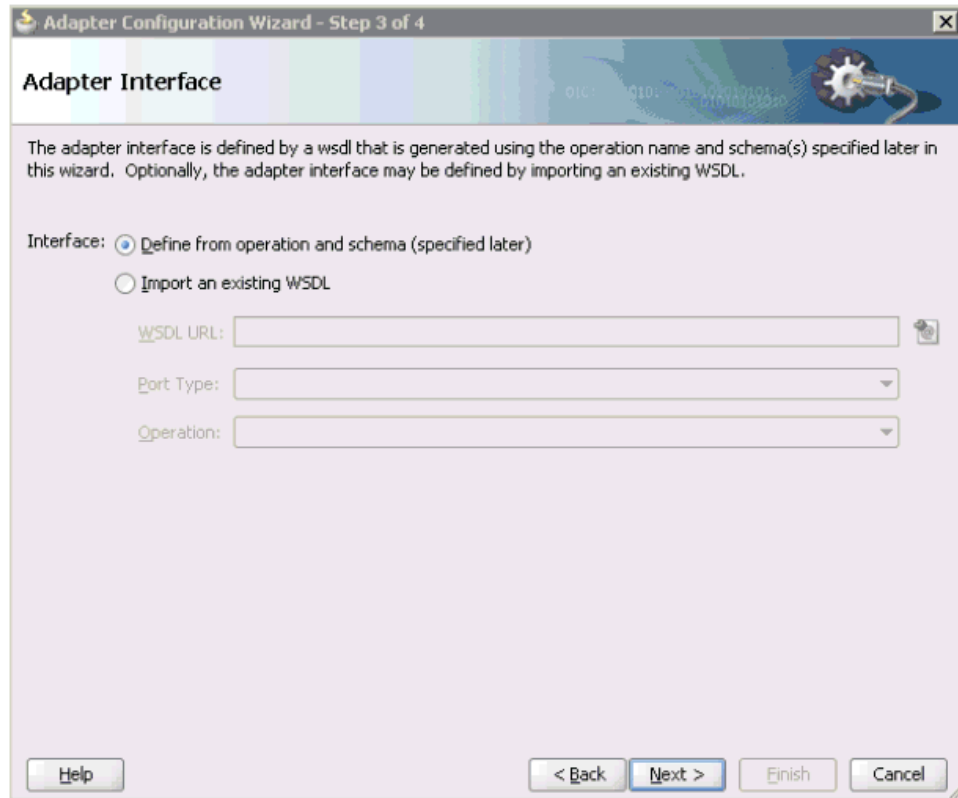


5. Drag and drop the **File Adapter** component from the Service Adapters pane to the External References pane.

The Adapter Configuration Wizard is displayed, showing the Service Name page.

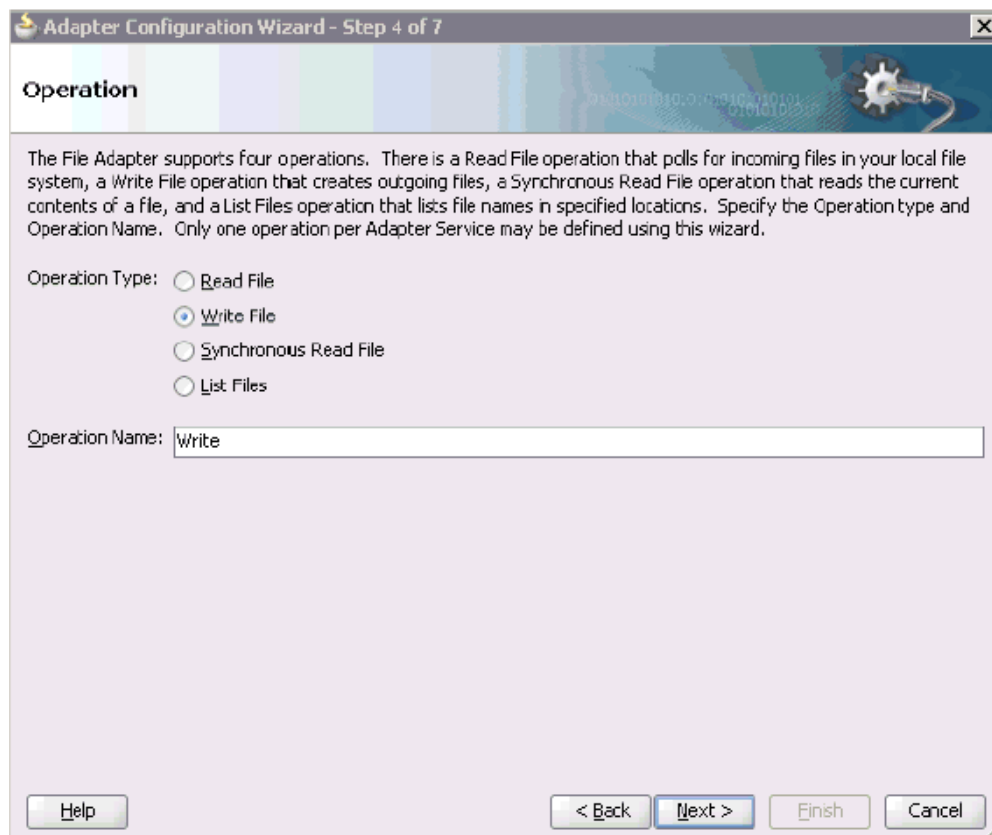


6. Type a name for the new File adapter in the Service Name field and click **Next**.  
The Adapter Interface page is displayed.



The screenshot shows a window titled "Adapter Configuration Wizard - Step 3 of 4". The main heading is "Adapter Interface". Below the heading, there is a descriptive text: "The adapter interface is defined by a wsdl that is generated using the operation name and schema(s) specified later in this wizard. Optionally, the adapter interface may be defined by importing an existing WSDL." Under the heading "Interface:", there are two radio button options: "Define from operation and schema (specified later)" (which is selected) and "Import an existing WSDL". Below these options are three input fields: "WSDL URL:" with a text box and a file icon, "Port Type:" with a dropdown menu, and "Operation:" with a dropdown menu. At the bottom of the window, there are four buttons: "Help", "< Back", "Next >", and "Cancel".

7. Ensure that the **Define from operation and schema (specified later)** option is selected.
8. Click **Next**.  
The Operation page is displayed.



9. Select **Write File** from the list of Operation Type options and specify an Operation Name (for example, Write).

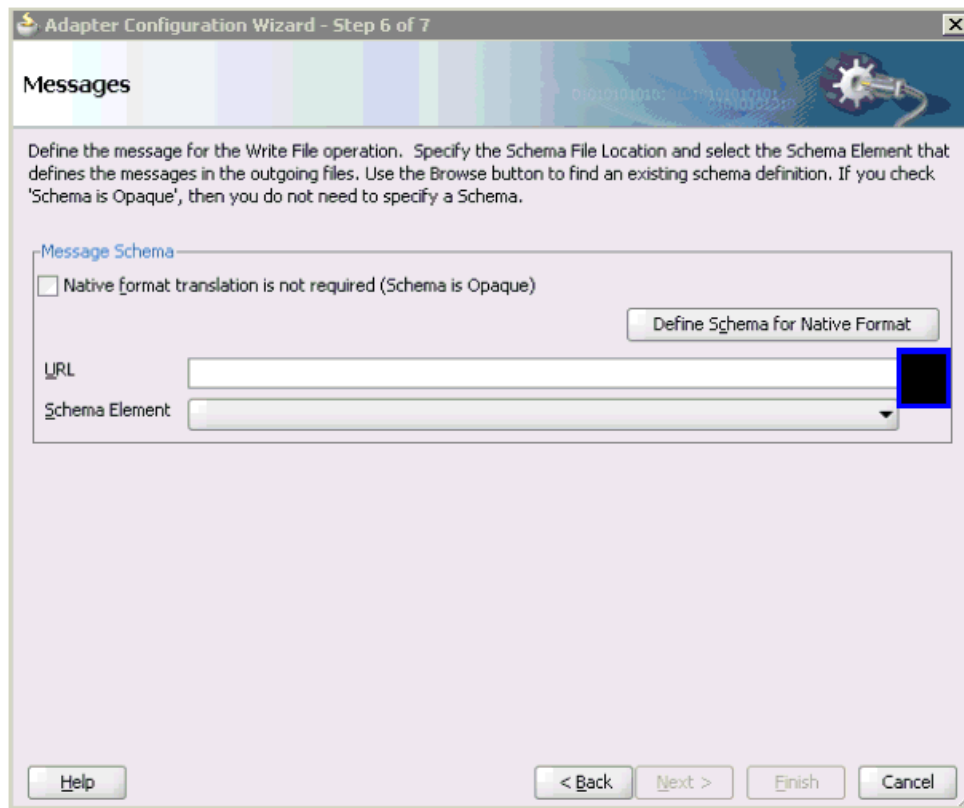
10. Click **Next**.

The File Configuration page is displayed.

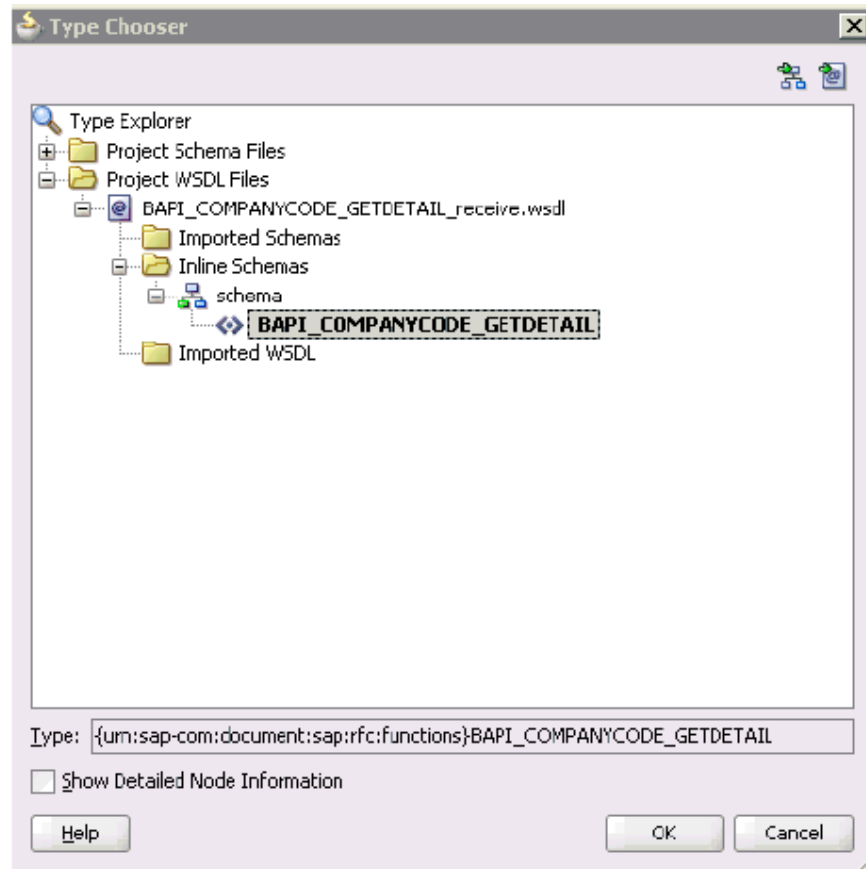
The screenshot shows the 'Adapter Configuration Wizard - Step 5 of 7' window. The title bar includes a close button (X). The main heading is 'File Configuration'. Below the heading, the instruction reads: 'Specify the parameters for the Write File operation.' There are two radio buttons: 'Physical Path' (selected) and 'Logical Name'. The text 'Directory for Outgoing Files (physical path):' is followed by a text box containing 'C:\MySAP\_Output' and a 'Browse' button. Below this is the 'File Naming Convention (po\_%SEQ%.txt):' field with the value 'MySAP\_ESDSUN9\_OUTPUT.xml'. A checkbox labeled 'Append to existing file' is unchecked. A section titled 'Write to output file when any of these conditions are met' contains three rows: 'Number of Messages Equals: 1' (checked), 'Elapsed Time Exceeds: 1 minutes' (unchecked), and 'File Size Exceeds: 1000 kilobytes' (unchecked). At the bottom, there are buttons for 'Help', '< Back', 'Next >', 'Finish', and 'Cancel'.

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

The Messages page is displayed.

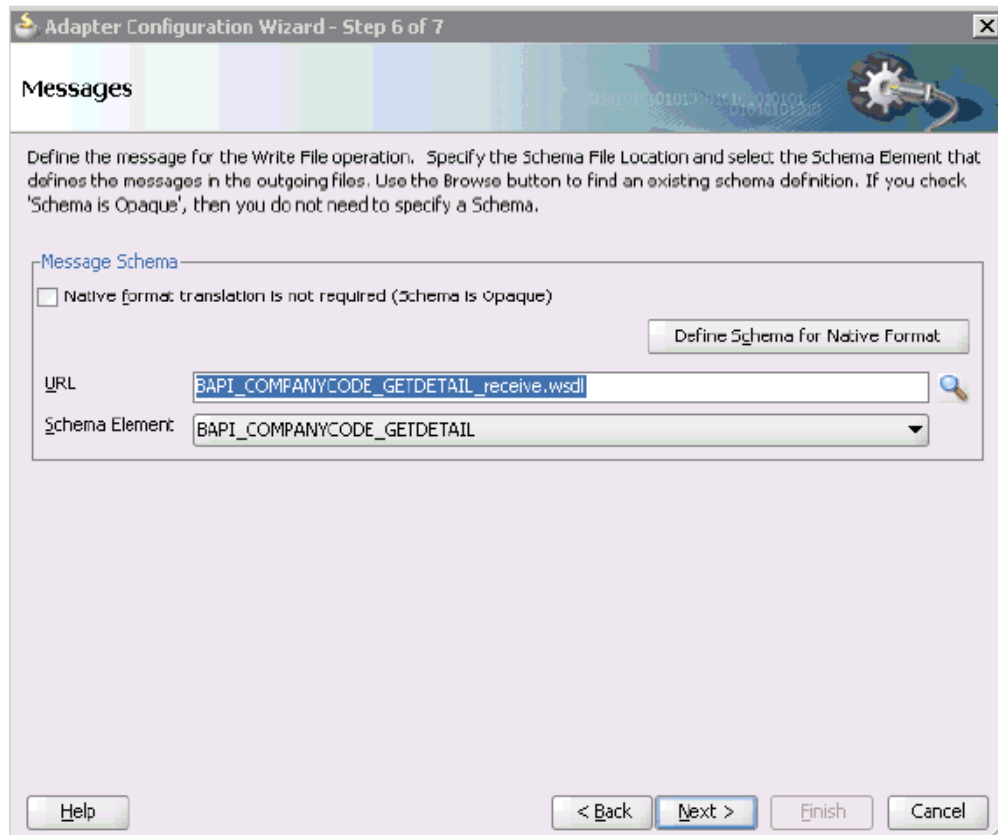


14. Click **Browse**, which is located to the right of the URL field.  
The Type Chooser dialog is displayed.



15. Expand **Project WSDL Files**, **BAPI\_COMPANYCODE\_GETDETAIL\_receive.wSDL**, **Inline Schemas**, and **schema**.
16. Select the available schema (for example, **BAPI.CompanyCode.GetDetail**)
17. Click **OK**.

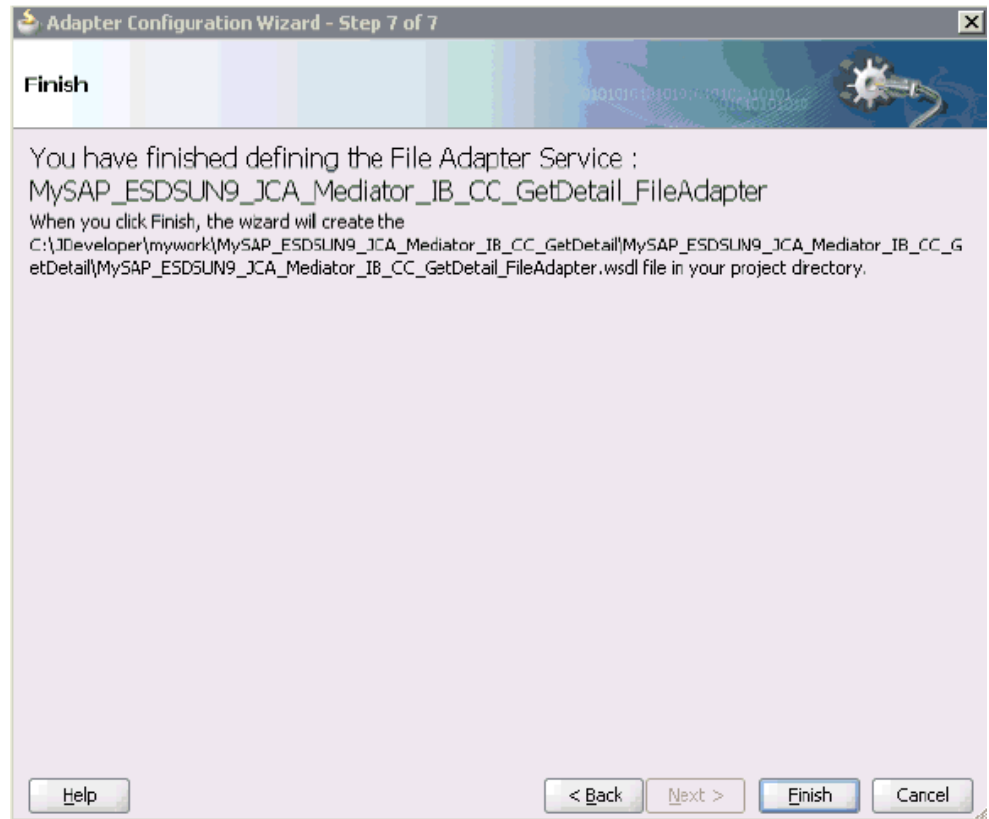
You are returned to the Messages page.



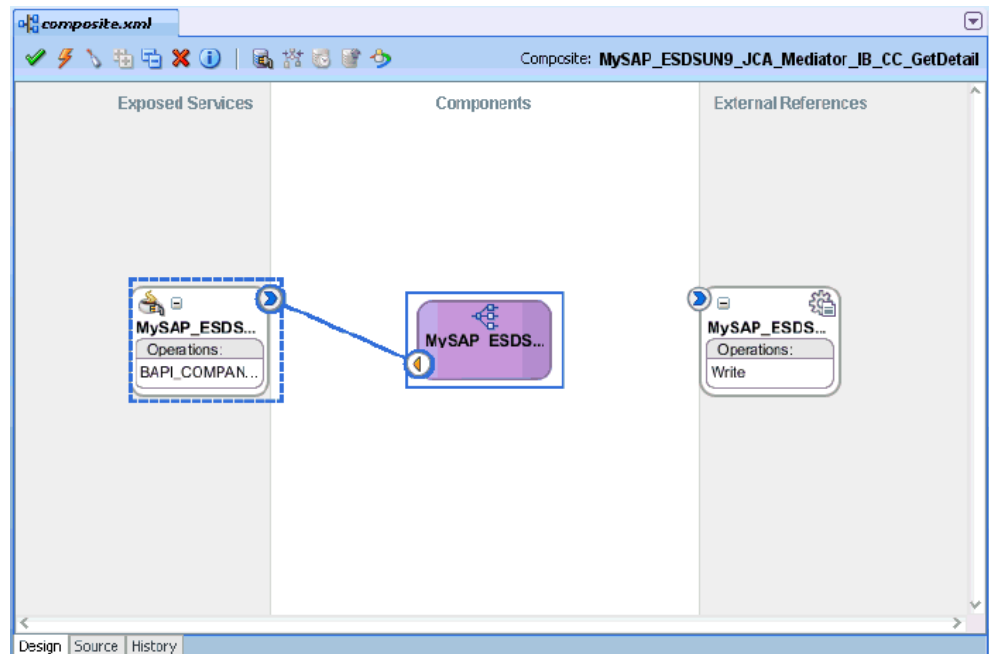
18. Click Next.

The Finish page is displayed.

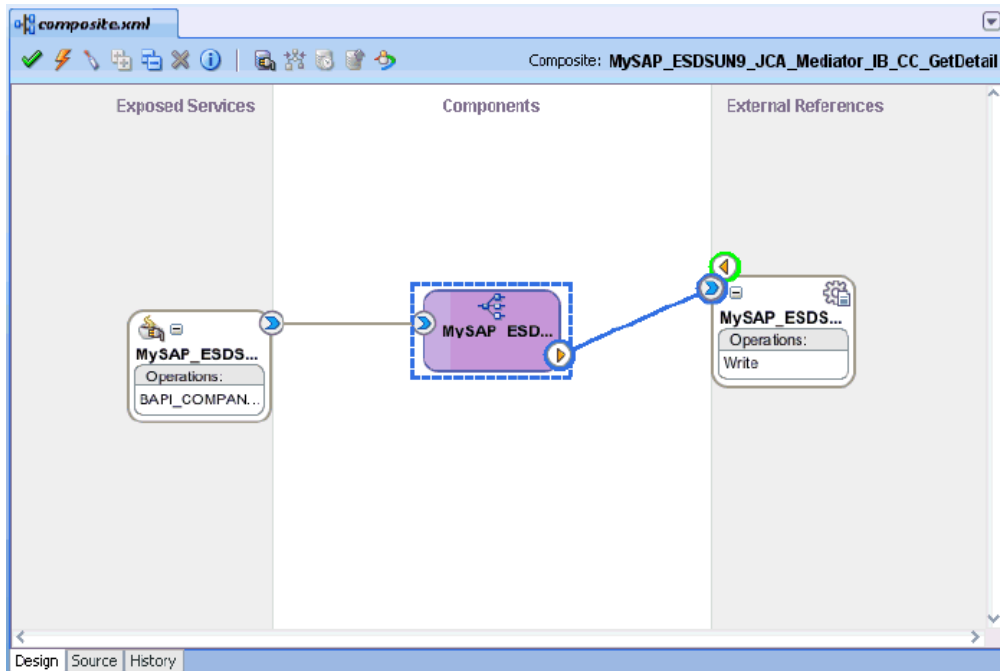




19. Click **Finish**.
20. Create a connection between the inbound Mediator process component and the third party adapter service component.



21. Create a connection between the inbound Mediator process component and the File adapter component.

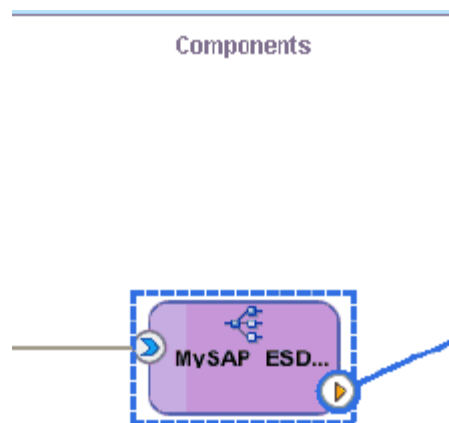


You are now ready to configure the routing rules.

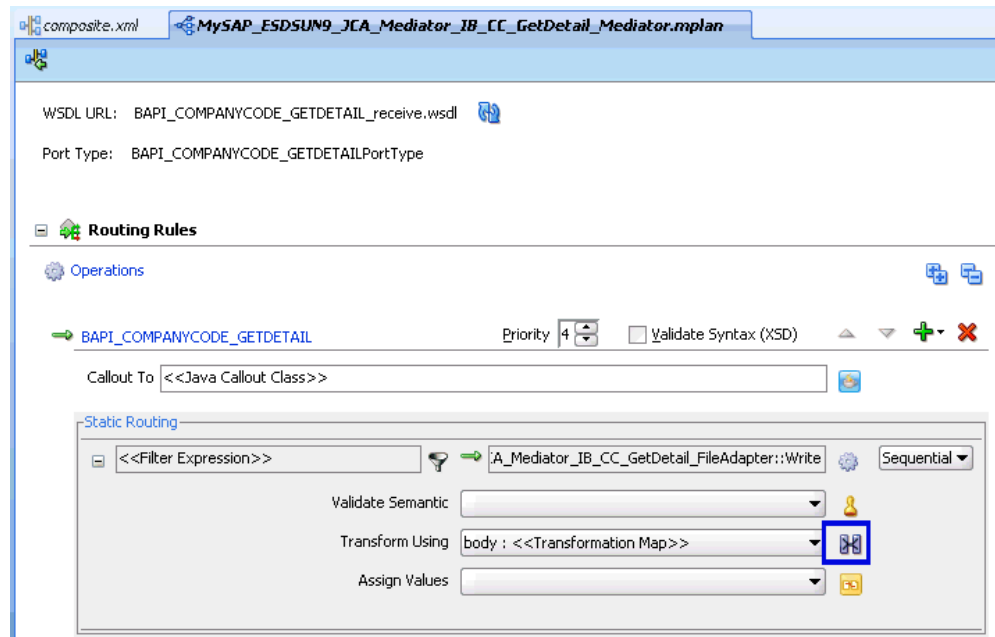
### Configuring the Routing Rules

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.

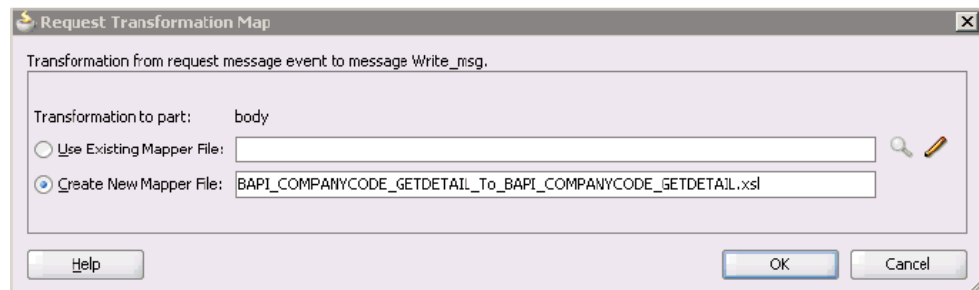


The Routing Rules dialog is displayed.



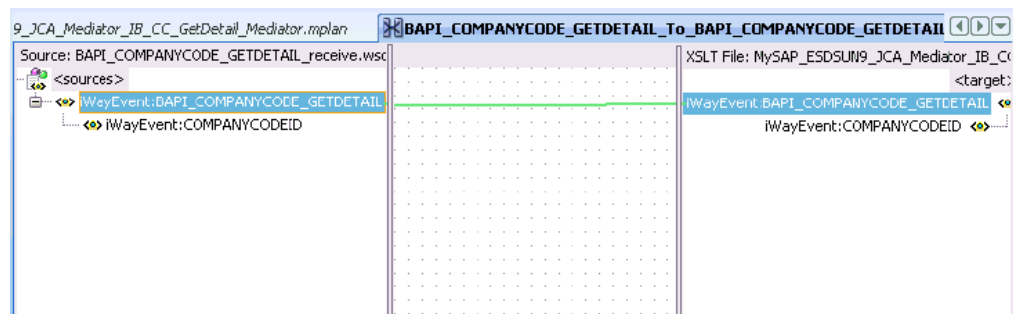
2. In the <<Filter Expression>> area, click the icon to the right of the Transform Using field.

The Request Transformation Map dialog is displayed.



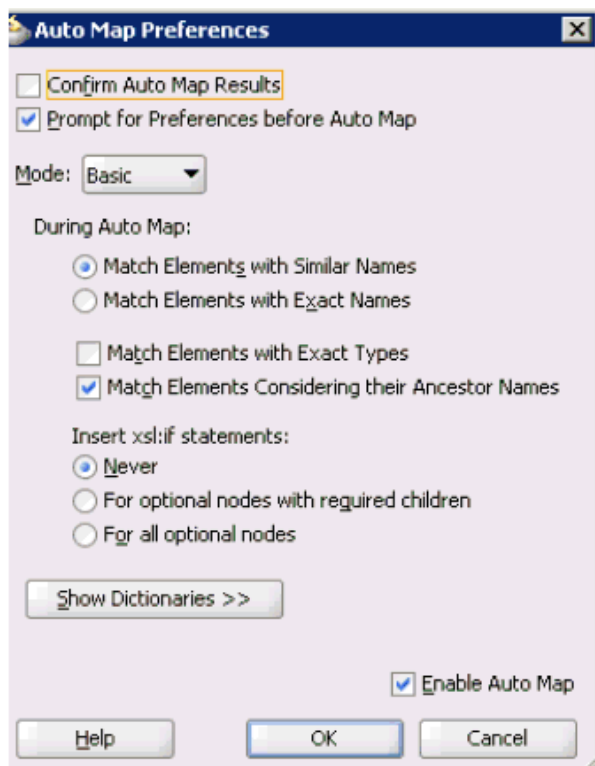
3. Select the **Create New Mapper File** option and click **OK**.

The following mapping page is displayed.



4. Map the **iWayEvent:BAPI\_COMPANYCODE\_GETDETAIL** source element to the **iWayEvent:BAPI\_COMPANYCODE\_GETDETAIL** target element.

The Auto Map Preferences dialog is displayed.



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

You are returned to the Routing Rules dialog. The mapping is now complete.

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

You are now ready to deploy the Mediator inbound process. You can follow the same procedure in "[Deploying the BPEL Inbound Process](#)" on page 5-51.

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 5-55.

---

---

## Troubleshooting and Error Messages

This chapter explains the limitations and workarounds when connecting to SAP R/3. The following topics are discussed:

- [Troubleshooting](#)
- [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.

### 7.1 Troubleshooting

This topic provides troubleshooting information for SAP R/3, separated into four categories:

- Application Explorer
- SAP R/3
- Oracle Adapter J2CA
- Oracle Adapter Business Services Engine (BSE)

---

---

**Note:** Log file information that can be relevant in troubleshooting can be found in the following locations:

- The Oracle Adapter J2CA trace information can be found under the following directory:

```
C:\oracle\Middleware\Oracle_  
SOA1\soa\thirdparty\ApplicationAdapters\config\config_name\log
```

- BSE trace information can be found under the following directory:

```
C:\oracle\Middleware\ 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:

```
C:\oracle\Middleware\Oracle_  
SOA1\soa\thirdparty\ApplicationAdapters\tools\iwae\bin
```

---

---

### Application Explorer

To use Application Explorer on **Windows** for debugging or testing purposes:

1. Ensure that Oracle WebLogic Server is started, which is where Application Explorer is deployed.
2. Start Application Explorer by clicking the Windows **Start** menu, selecting **All Programs, Oracle Application Adapters**, and clicking **Application Explorer**.



You can also start Application Explorer by executing the **ae.bat** file, which is located in the following directory:

```
C:\oracle\Middleware\home_0309\Oracle_
SOA1\soa\thirdparty\ApplicationAdapters\tools\iwae\bin\ae.bat
```

It is a good practice to create a shortcut for the **ae.bat** file on your desktop.

If you are using a UNIX or Linux platform you can start Application Explorer by executing the **iwae.sh** file.

Error	Solution
SAP R/3 does not appear in the Application Explorer Adapter node list.	Ensure that the <code>sapjco.jar</code> and <code>sapjcorfc.dll</code> files are added to the <code>lib</code> directory. Ensure the <code>librfc32.dll</code> file is added to the Windows system32 folder.
Cannot connect to Oracle Application Adapter for SAP R/3 from Application Explorer. Problem activating adapter	Ensure that: <ul style="list-style-type: none"> <li>■ SAP R/3 is running.</li> <li>■ The Application Server name, System Number, and Client Number are correct.</li> <li>■ The SAP R/3 user ID and password are correct.</li> </ul>
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.sapr3.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.sapr3.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 sapjcorfc [no sapjcorfc in java.library.path]. java.library.path </pre>	<p>Ensure that the <code>sapjcorfc.dll</code> file is added to the <code>lib</code> directory and the <code>librfc32.dll</code> file is added to the Windows <code>system32</code> folder.</p>
<p>The <code>dll</code> is loaded in another classloader (BSE and J2CA are installed on the same server). The following error message appears:</p> <pre> com.ibi.sapr3.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 sapjcorfc [Native Library F:\iWay55.008.0628\lib\sapjcorfc.d ll loaded in another classloader]. java.library.path </pre>	<p>Add <code>sapjco.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.&lt;init&gt;(JCATransport.java:6 2)at com.iwaysoftware.iwae.common.Adapt erClient.&lt;init&gt;(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>&lt;jdk_home&gt;</code>/bin/java, where <code>&lt;jdk_home&gt;</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, 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><a href="ftp://ftp.bouncycastle.org/pub">ftp://ftp.bouncycastle.org/pub</a></p>

### 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>
<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>



Error	Solution
<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.

---

### 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>C:\oracle\Middleware\Oracle_ SOA1\soa\thirdparty\ApplicationAdapters\ tools\iwae\bin\...\...</pre>

## 7.2 BSE Error Messages

This topic discusses the different types of errors that can occur when processing Web services through BSE.

### 7.2.1 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, 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.

## 7.2.2 Adapter-Specific Error Handling

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, 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.

### 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.sapjco.SapCallableStatement: execute() j
java.util.NoSuchElementException</faultstring>
  </SOAP-ENV:Fault>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

### 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.sapjco.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>

```

### 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>

```

### 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>

```

### 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>
```

---

---

## Advanced User Tools

This chapter includes the following topics:

- [Web Services Policy-Based Security](#)
- [Migrating Repositories](#)

### 8.1 Web Services Policy-Based Security

Application Explorer provides a security model called Web services policy-based security. The following topics describe how the feature works and how to configure it.

Web services provide a layer of abstraction between the back-end business logic and the user or application running the Web service, which enables easy application integration. However, the issue of controlling the use and implementation of critical and sensitive business logic that is run as a Web service is raised.

Application Explorer controls the use of Web services that use adapters, using a feature called policy-based security. This feature enables an administrator to apply "policies" to Business Services (Web services) to deny or permit their execution.

A policy is a set of privileges dealing with the execution of a Business Service (BS) that can be applied to an existing or new BS. When you set specific rights or privileges inside a policy, you do not have to re-create privileges for every BS that has security concerns in common with other Business Services. Instead, you reuse a policy on multiple Business Services.

The goal of the feature is to secure requests at both the transport and the SOAP request level transmitted on the wire. Some of the policies do not deal with security issues directly, but do affect the run-time behavior of the Web services to which they have been applied.

The Business Services administrator creates an "instance" of a policy type, names it, associates individual users or groups (a collection of users), and then applies that policy to one or more Business Services.

You can assign a policy to a Business Service, or to a method within a Business Service. If a policy is only applied to a method, other methods in that Business Service are not governed by it. However, if a policy is applied to the Business Service, all methods are governed by it. At run-time, the user ID and password that are sent to BSE in the SOAP request message are verified against the list of users for all policies applied to that specific Business Service. The policy type that is supported is Resource Execution, which dictates who can or cannot perform the Business Service.

When a policy is not applied, the default value for a Business Service is to "grant all". For example, anybody can run the Business Service, until the Resource Execution policy is associated to the Business Service. At that time, only those granted execution

permissions, or users not part of the group that has been denied execution permissions, have access to the Business Service.

### 8.1.1 Configuring Web Services Policy-Based Security

The following procedures describe how to configure Web services policy-based security.

#### Creating and Associating a User with a Policy

Before you create instances of policies, you must have a minimum of one user or one group to associate to an instance. You can create users and groups using Application Explorer.

1. Open Application Explorer.
2. Right-click the configuration to which you want to connect, for example, SampleConfig. See [Chapter 3, "Configuring Oracle Application Adapter for SAP R/3"](#) for information on creating a new configuration.
3. Select Connect.

Nodes appear for Adapters and Business Services (also known as Web services).



Perform the following steps:

- a. Expand the Business Services node.
- b. Expand the Configuration node.
- c. Expand the Security node.
- d. Expand the Users and Groups node.



4. Right-click Users and click New User.

The New User dialog is displayed.

Provide the following information:

- a. In the Name field, enter a user ID.
  - b. In the Password field, enter the password associated with the user ID.
  - c. In the Description field, enter a description of the user (optional).
5. Click OK.



The new user is added under the Users node.

### Creating a Group to Use With a Policy

To create a group to use with a policy:

1. Open Application Explorer.
2. Right-click the configuration to which you want to connect, for example, SampleConfig. See [Chapter 3, "Configuring Oracle Application Adapter for SAP R/3"](#) for information on creating a new configuration.
3. Select Connect.

Nodes appear for Adapters and Business Services (also known as Web services).

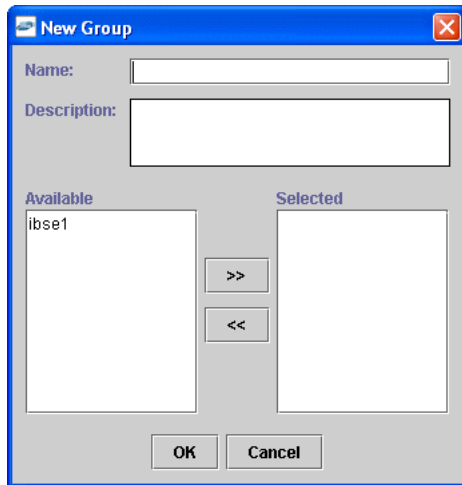


Perform the following steps:

- a. Expand the Business Services node.
  - b. Expand the Configuration node.
  - c. Expand the Security node.
  - d. Expand the Users and Groups node.
4. Right-click Groups and select New Group.



The New Group dialog is displayed.



Provide the following information:

- a. In the Name field, enter a name for the group.
  - b. In the Description field, enter a description for the group (optional).
  - c. From the available list of users in the left pane, select one or more users and add them to the Selected list by clicking the double right-facing arrow.
5. When you have selected at least one user, click OK.

The new group is added under the Group node.

### Creating an Execution Policy

An execution policy governs who can run the Business Services to which the policy is applied.

To create an execution policy:

1. Open Application Explorer.
2. Right-click the configuration to which you want to connect, for example, SampleConfig. See [Chapter 3, "Configuring Oracle Application Adapter for SAP R/3"](#) for information on creating a new configuration.
3. Select Connect.

Nodes appear for Adapters and Business Services (also known as Web services).



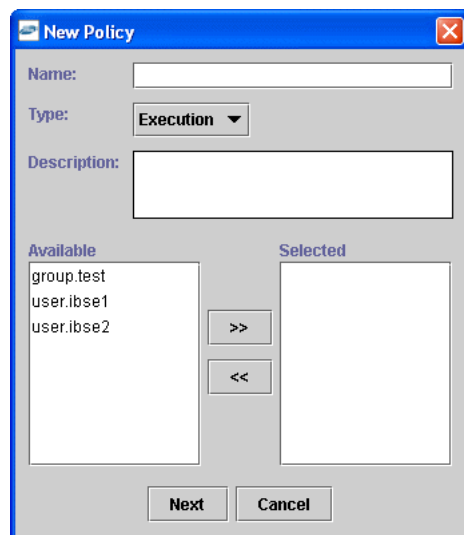
Perform the following steps:



- a. Expand the Business Services node.
  - b. Expand the Configuration node.
  - c. Expand the Security node.
  - d. Expand the Policies node.
4. Right-click Policies and select New Policy.



The New policy dialog is displayed.



Provide the following information:

- a. In the Name field, enter a name for the policy.
- b. From the Type list, select Execution.
- c. In the Description field, enter a description for the policy (optional).
- d. From the available list of users in the left pane, select one or more users and add them to the Selected list by clicking the double right-facing arrow.

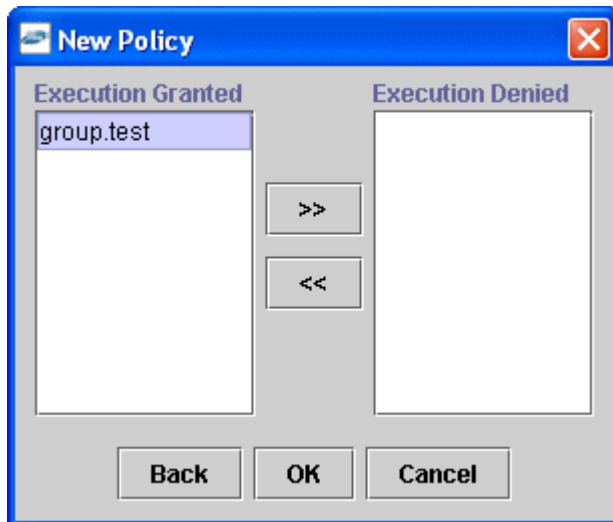
---

**Note:** This user ID is verified against the value in the user ID element of the SOAP header sent to BSE in a SOAP request.

---

5. When you have selected at least one user selected, click OK.
6. Click Next.

The New Policy permissions dialog is displayed.



- To grant permission to a user or group to run a Business Service, select the user or group and move them into the Execution Granted list by selecting the double left-facing arrow.
- To deny permission to a user or group to run a Business Service, select the user or group and move them into the Execution Denied list by selecting the double right-facing arrow.

7. Click OK.

The following pane summarizes your configuration.

- **Name**            test
- **Type**            Execution
- **Description**
- **User and Group Restrictions**
  - group.test Execution Granted

### Using the IP and Domain Restrictions Policy Type

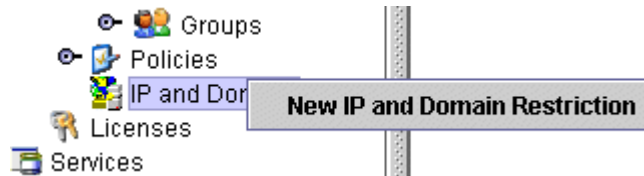
You configure the IP and Domain Restriction policy type slightly differently from other policy types. The IP and Domain Restriction policy type controls connection access to BSE and therefore need not be applied to individual Web services. You need not create a policy; however, you must enable the Security Policy option in Application Explorer.

1. Open Application Explorer.
2. Right-click the configuration to which you want to connect, for example, SampleConfig. See [Chapter 3, "Configuring Oracle Application Adapter for SAP R/3"](#) for information on creating a new configuration.
3. Select Connect.

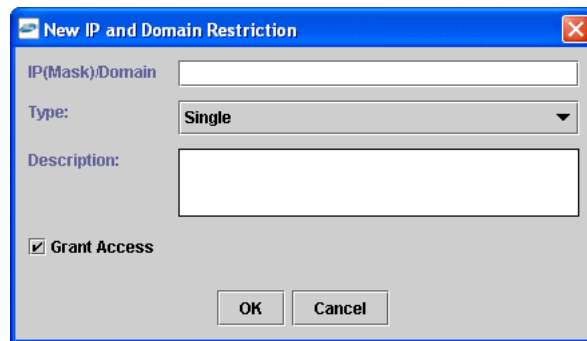
Nodes appear for Adapters and Business Services (also known as Web services).

- a. Expand the Business Services node.

- b. Expand the Configuration node.
  - c. Expand the Security node.
4. Right-click IP and Domain and select New IP and Domain Restriction.



The New IP and Domain Restriction dialog is displayed.



Provide the following information:

- a. In the IP(Mask)/Domain field, enter the IP or domain name using the following guidelines.
    - If you select Single (Computer) from the Type list, you must provide the IP address for that computer. If you only know the DNS name for the computer, click DNS Lookup to obtain the IP Address based on the DNS name.
    - If you select Group (of Computers), you must provide the IP address and subnet mask for the computer group.
    - If you select Domain, you must provide the domain name.
  - b. From the Type list, select the type of restriction.
  - c. In the Description field, enter a description (optional).
  - d. To grant access, select the Grant Access check box.
5. Click OK.

The new domain is added under the IP and Domain node.

The following pane summarizes your configuration.

- **IP Address (Mask) /Domain** www.yahoo.com
- **Type** Domain
- **Access** Denied
- **Description**

## 8.2 Migrating Repositories

During design time, the Oracle 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. For management purposes, you can migrate BSE and J2CA repositories that are configured for Oracle to new destinations without affecting your existing configuration. For example, you may want to migrate a repository from a test environment to a production environment.

### Migrating a BSE Repository

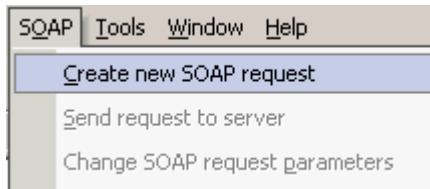
To migrate a BSE repository:

1. Copy the BSE control service URL, for example:

```
http://localhost:8001/ibse/IBSEServlet/admin/iwcontrol.ibs
```

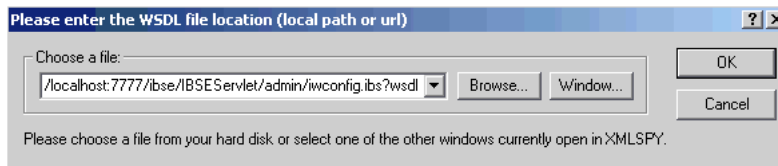
2. Open a third-party XML editor, for example, XMLSPY.
3. From the menu bar, click SOAP.

A list of options appears.



4. Select Create new SOAP request.

The WSDL file location dialog is displayed.



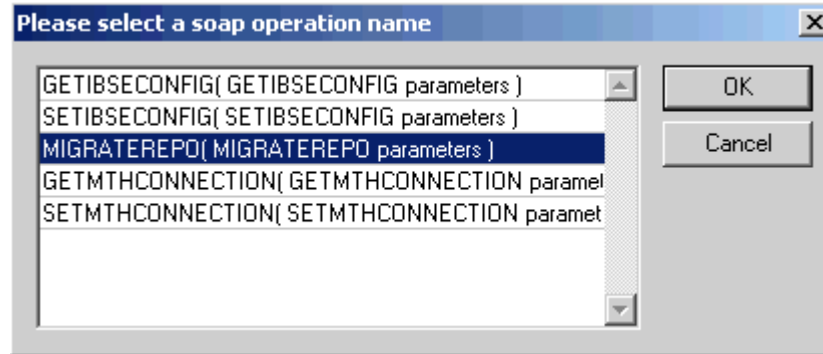
Perform the following steps:

- a. In the Choose a file field, paste the BSE control service URL.
- b. Append ?wsdl to the URL, for example:

```
http://localhost:8001/ibse/IBSEServlet/admin/iwcontrol.ibs?wsdl
```

5. Click OK.

The soap operation name dialog is displayed and the available control methods are listed.



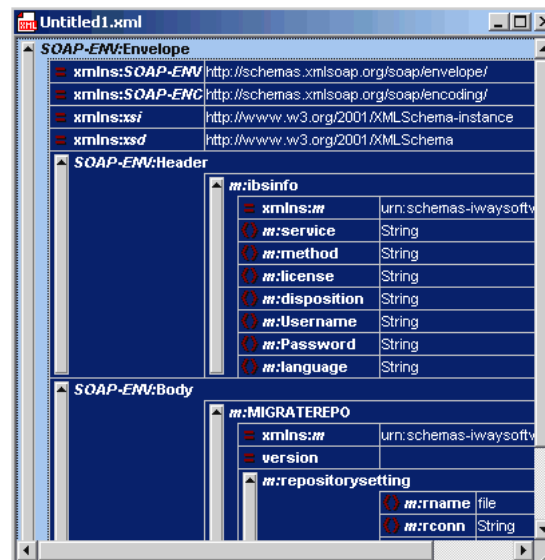
6. Select the MIGRATEREPO(MIGRATEREPO parameters) control method and click OK.

---

**Note:** The MIGRATEREPO(MIGRATEREPO parameters) control method is available from the BSE administration console. This control method migrates all Web services to the new (empty) repository. You can choose to migrate select Web services only.

---

The following window is displayed, showing the structure of the SOAP envelope.



7. Locate the Text view icon in the toolbar.



8. To display the structure of the SOAP envelope as text, click Text view.

The `<SOAP-ENV:Header>` tag is not required and can be deleted from the SOAP envelope.

9. Locate the following section:

```
<m:MIGRATEREPO xmlns:m="urn:schemas-iwaysoftware-com:jul2003:ibse:config"
version="">
```

```

<m:repositorysetting>
  <m:rname>oracle</m:rname>
  <m:rconn>String</m:rconn>
  <m:rdriver>String</m:rdriver>
  <m:ruser>String</m:ruser>
  <m:rpwd>String</m:rpwd>
</m:repositorysetting>
<m:servicename>String</m:servicename>
</m:MIGRATEREPO>

```

Perform the following steps:

- a. For the `<m:rconn>` tag, replace the String placeholder with a repository URL where you want to migrate your existing BSE repository.

The Oracle repository URL has the following format:

```
jdbc:oracle:thin:@[host]:[port]:[sid]
```

- b. For the `<m:rdriver>` tag, replace the String placeholder with the location of your Oracle driver.
- c. For the `<m:ruser>` tag, replace the String placeholder with a valid user name to access the Oracle repository.
- d. For the `<m:rpwd>` tag, replace the String placeholder with a valid password to access the Oracle repository.

10. Select and perform a migration option:

- If you want to migrate a single Web service from the current BSE repository, enter the Web service name in the `<m:servicename>` tag, for example:

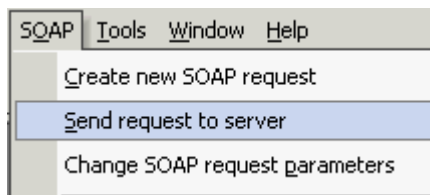
```
<m:servicename>SAPService1</m:servicename>
```

- If you want to migrate multiple Web services from the current BSE repository, duplicate the `<m:servicename>` tag for each Web service, for example:

```
<m:servicename>SAPService1</m:servicename>
<m:servicename>SAPService2</m:servicename>
```

- If you want to migrate all Web services from the current BSE repository, remove the `<m:servicename>` tag.

11. From the menu bar, click SOAP and select Send request to server.



Your BSE repository and any Web services you specified are now migrated to the new Oracle repository URL you specified.

### Migrating a J2CA Repository

To migrate a J2CA repository:

1. Navigate to the location of your J2CA configuration directory where the repository schemas and other information is stored, for example:

```
C:\oracle\Middleware\Oracle_SOA1\soa\thirdparty\ApplicationAdapters\config\JCA_
```

*CONFIG*

Where *JCA\_CONFIG* is the name of your J2CA configuration.

2. Locate and copy the `repository.xml` file.
3. Place this file in a new J2CA configuration directory to migrate the existing repository.

Your J2CA repository is migrated to the new J2CA configuration directory.





---

---

# Configuring SAP R/3 for Inbound and Outbound Processing

During inbound (client) processing, IDocs are transferred to the interface and stored in the SAP R/3 system. The document data is generated in a second step, also during a workflow.

Outbound processing in SAP R/3 involves event handling. An event in SAP R/3 is defined as an occurrence of a status change in an object. Events are created when the relevant status change occurs.

The following topics describe how to enable inbound and outbound SAP R/3 processing.

- [Configuring SAP R/3 Inbound Processing](#)
- [Configuring SAP R/3 Outbound Processing](#)

## A.1 Configuring SAP R/3 Inbound Processing

SAP R/3 inbound processing requires the upstream system to transfer an IDoc to the IDoc interface through the ERP System port. For this reason, you do not have to specify a port in the inbound partner profiles; the IDoc interface only must recognize the upstream system as a port. A port definition, which provides a unique ID for the upstream system, must be available for the port. The technical parameters of this port definition can (and usually are) overwritten by the upstream system.

If the upstream system is recognized, then the IDoc is saved in the database. If a partner is defined with the corresponding message in partner profiles, the IDoc is then processed further. This is done independently in the second step. This ensures that the external system can receive the data quickly and reliably (automatically).

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.1.1 Configuring a Logical System

In any distributed environment, each participating system must have a unique ID to avoid confusion. In SAP R/3, the name of the logical system is used as the unique ID. This name is assigned explicitly to one client in a SAP R/3 system.

## Defining a Logical System

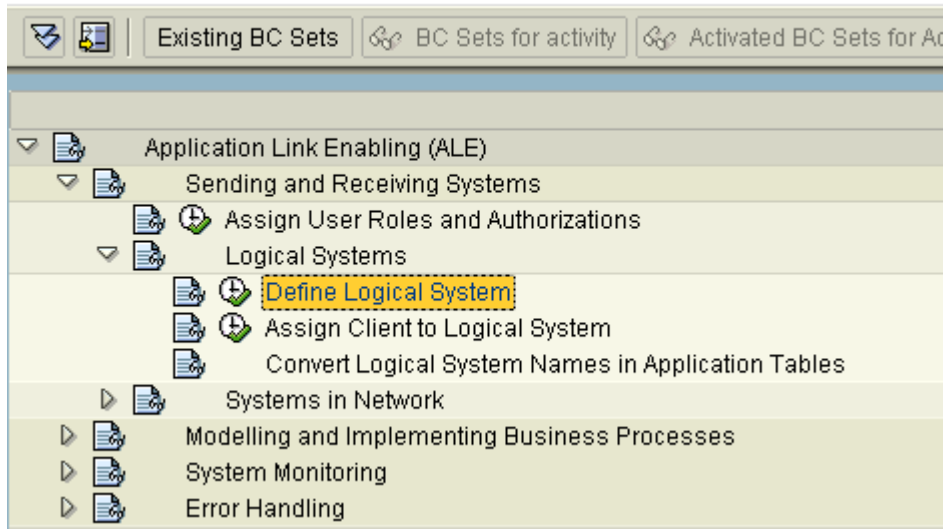
To define a logical system:

1. Run the **sale** transaction.



The Display IMG window is displayed.

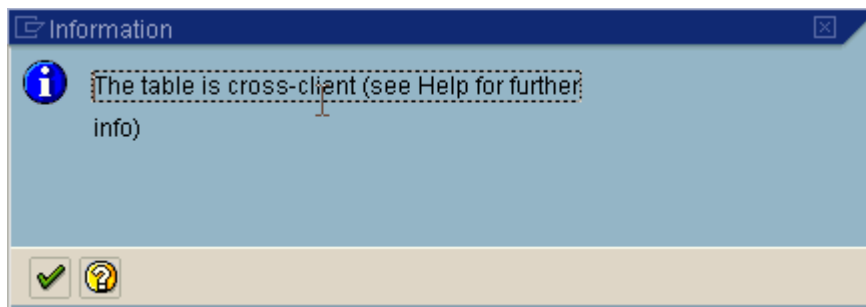
### Display IMG



Perform the following steps:

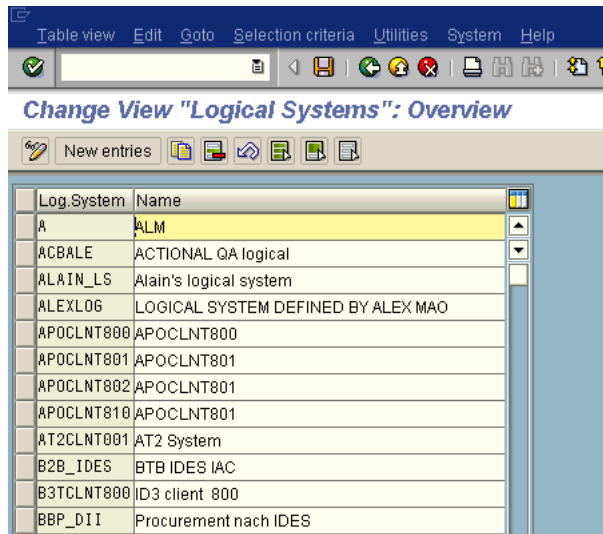
- a. Expand Sending and Receiving Systems.
  - b. Expand **Logical Systems**.
  - c. Select Define Logical System.
2. Click the IMG - Activity icon.

A message window is displayed. It indicates that the table is cross-client.



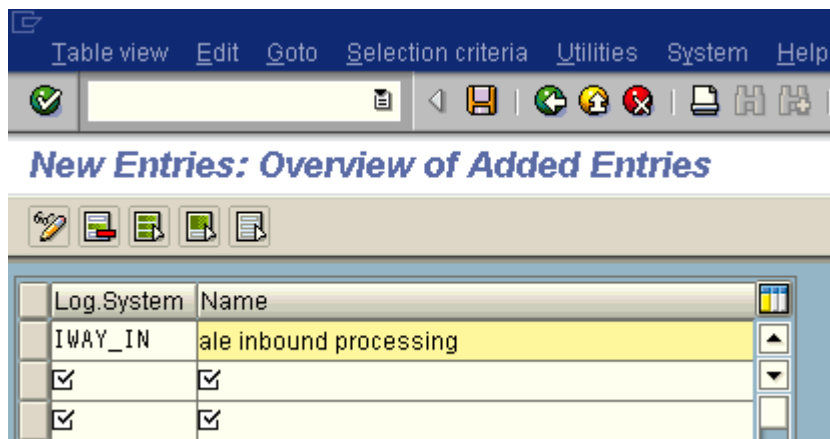
3. Click the **check mark** icon to continue.

The Change View "Logical Systems": Overview window is displayed.



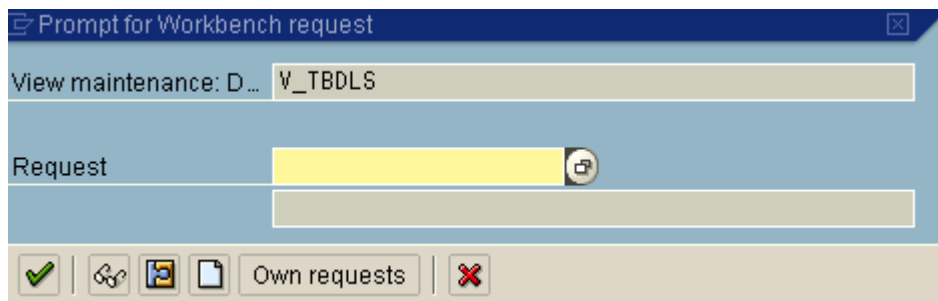
4. Click New Entries.

The New Entries: Overview of Added Entries window is displayed.



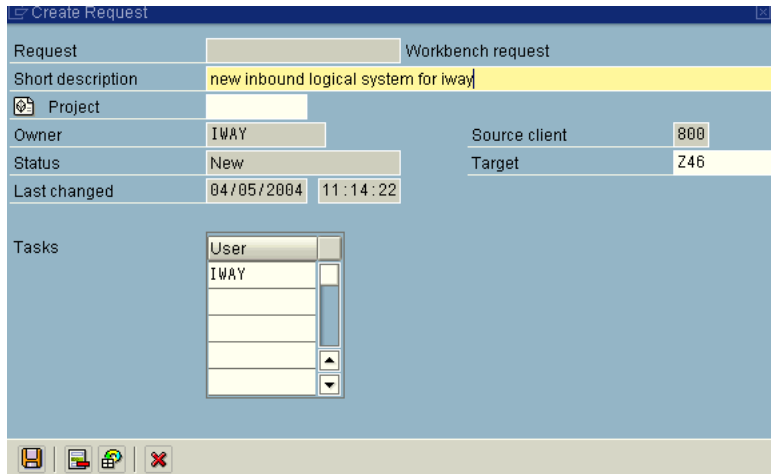
5. Enter the Logical System, for example, ORACLETDS, in the **Log.System** column and provide a description in the **Name** column.
6. Click Save.

The Prompt for Workbench request dialog box is displayed.



7. Click the Create Request icon.

The Create Request dialog box is displayed.



- Enter a name and description for your request and click Save.

The logical system you configured, for example, ORACLETDS, is now added to the list.

IWAYMKT	IWAY marketing logical system
IWAY_IN	ale inbound processing
JRB46LS	jr logical

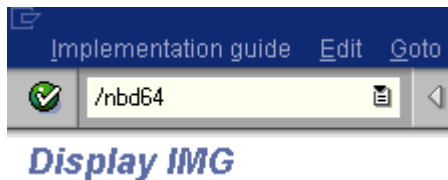
### A.1.2 Configuring a Distribution Model

A distribution model is used to describe the ALE message flow between logical systems. Business objects are distributed to connected recipients according to a unique distribution model that can contain rules of varying complexity depending on the type of business objects involved.

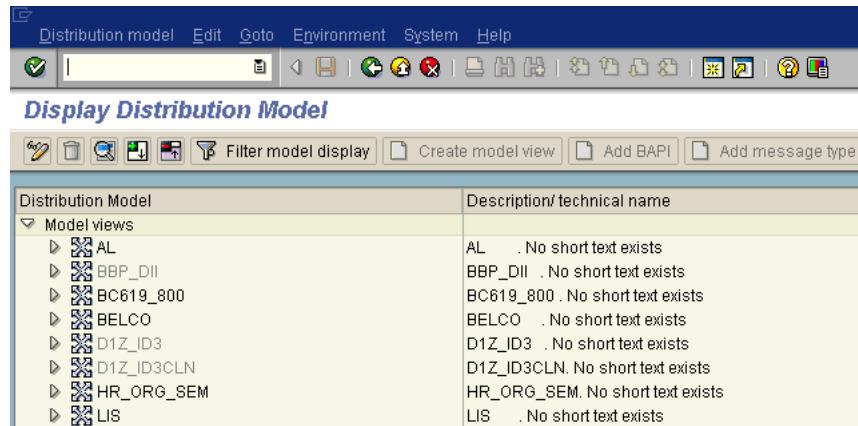
#### Defining a Distribution Model

To define a distribution model:

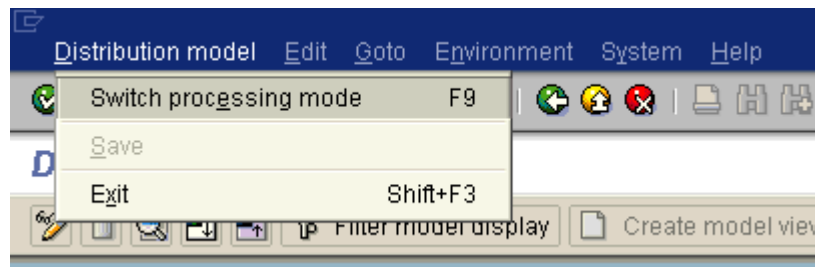
- Run the **bd64** transaction.



The Display Distribution Model window is displayed.

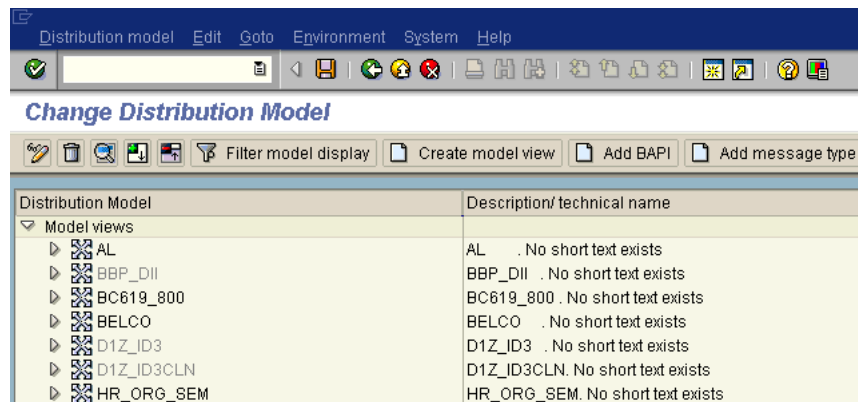


2. Click Distribution Model from the menu bar.



3. Select **Switch processing mode**.

The Display Distribution Model window is switched to Change Distribution Model.



4. Click Create model view.

The Create Model View dialog box is displayed.

Short text	iway ale inbound
Technical name	ziwayale
Start date	04/05/2004
End date	12/31/9999

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.
6. Click the **check mark** icon to enter the information.

You are returned to the main Change Distribution Model window. The distribution model you configured is now added to the list.

▶	✖	detlef	DETLEF
▶	✖	iway Distribution Model for alpha class	IWAYMOD09
▶	✖	iway ale inbound	ZIWAYALE
▶	✖	iway marketing distribution model	IWAYMKT

7. Click Add message type.

The Add Message Type dialog box is displayed.

Model view	ZIWAYALE
Sender	IWAY_IN
Receiver	IWAY_IN
Message type	MATMAS

Perform the following steps:

- a. In the **Sender** and **Receiver** fields, enter the logical system you configured, for example, ORACLETDS.  
 You can click the icon to the right of each field to browse from a list of logical systems.
- b. In the **Message type** field, enter the message type you want to use, for example, MATMAS.  
 You can click the icon to the right of each field to browse from a list of available message types.
8. Click the **check mark** icon to enter the information.  
 You are returned to the main Change Distribution Model window.
9. Click Save.

### A.1.3 Defining a Partner Profile

Partner profiles are a prerequisite for data exchange. This involves defining who can exchange messages with the SAP R/3 system and using which port.

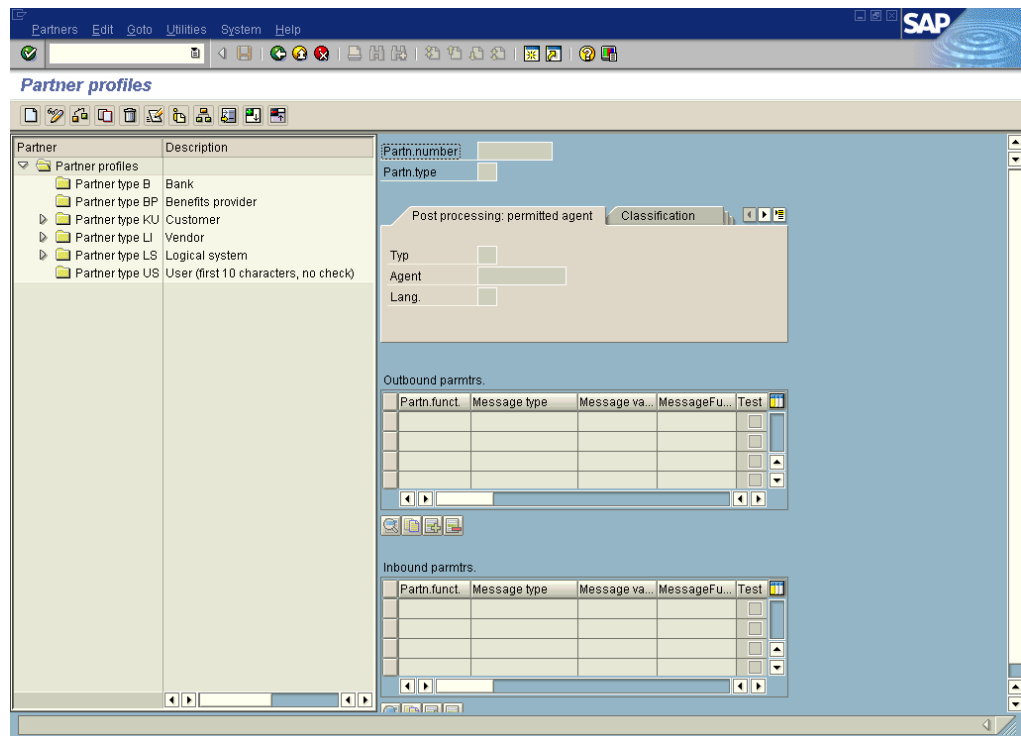
#### Defining a Partner Profile

To define a partner profile:

1. Run the **we20** transaction.

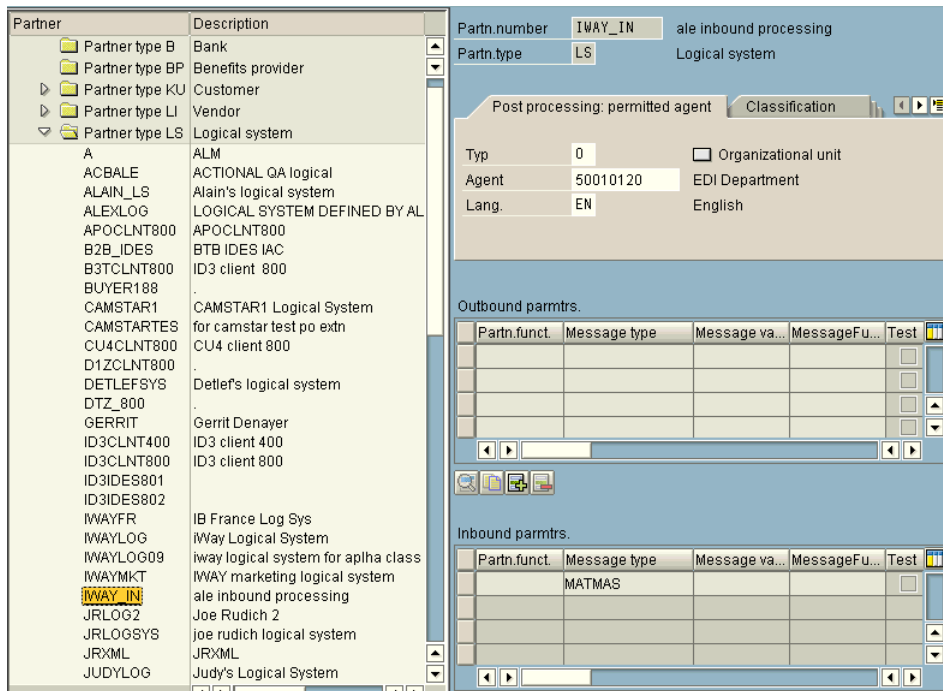


The Partner profiles window is displayed.



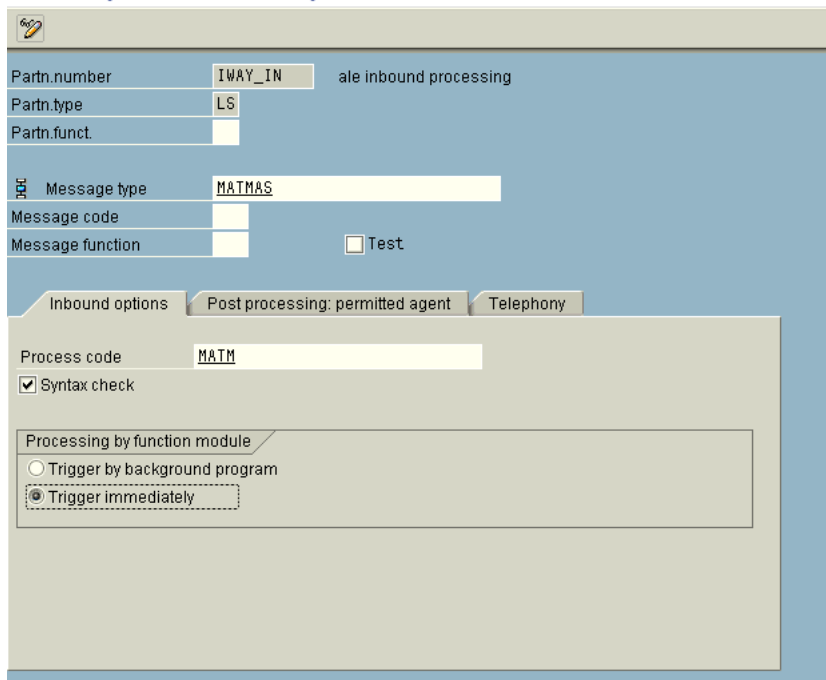
2. In the left pane, expand Partner type LS and select the logical system you configured from the list, for example, ORACLETDS.

In the right pane, the Partn.number field refers to the name of the logical system.



3. Click Save.
4. From the **Inbound** parameters table, click the Create inbound parameter icon.  
The Partner profiles: Inbound parameters window is displayed.

**Partner profiles: Inbound parameters**



5. In the **Message type** field, enter the message type you want to use, for example, MATMAS.  
You can click the icon to the right of each field to browse from a list of available message types.



The Inbound options tab is selected by default.

6. In the **Process code** field, enter the process code you want to use, for example, MATM.

You can click the icon to the right of each field to browse from a list of available process codes.

7. In the **Processing by function module** area, select an option from the list:
  - Trigger by background program.  
In this case the adapter writes IDocs to the SAP R/3 database, which is processed immediately.
  - Trigger immediately.  
In this case, the adapter waits for the SAP R/3 system to process IDocs. This can take anywhere from 1 to 15 minutes.
8. Click Save.

## A.2 Configuring SAP R/3 Outbound Processing

Event creation must be implemented by you or by SAP R/3. An event is created from specific application programs (the event creator) and then published systemwide. Any number of receivers can respond to the event with their own response mechanisms. An event is usually defined as a component of an object type.

SAP R/3 pseudo events are not processed by the SAP R/3 Event manager, but are called from an ABAP program or Remote Function Call (using the Destination parameter).

### A.2.1 Related Concepts and Terminology

The following topic lists and defines specific terminology related to SAP R/3 and SAP R/3 event handling.

#### Client and Server Programs

RFC programs for non-SAP R/3 systems can function as either the caller or the called program in an RFC communication. There are two types of RFC programs:

- RFC Client
- RFC Server

The RFC client is the instance that calls the RFC to run the function that is provided by an RFC server. The functions that can be called remotely are called RFC functions, and the functions provided by the RFC API are called RFC calls.

#### SAP R/3 Gateway

The SAP R/3 Gateway is a secure application server. No connections are accepted unless they have been preregistered previously from the SAP R/3 presentation Client. 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, the Gateway server then offers a connection to the server, which "Accepts" a connection. This ProgramID is then linked with an RFC Destination within SAP R/3, which enables SAP R/3 Function Modules and ALE documents (IDocs or BAPI IDocs) to be routed to the destination. The RFC Destination functions as a tag to mask the Program ID to SAP R/3 users.

An RFC server program can be registered with the SAP R/3 gateway and wait for incoming RFC call requests. An RFC server program registers itself under a Program ID at a SAP R/3 gateway and not for a specific SAP R/3 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 R/3 gateway at which the RFC server program is registered.

### **Program IDs and Load Balancing**

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 WebLogic Server using a single SAP R/3 program ID, SAP R/3 load balances the event data. For example, if multiple remote function calls or BAPIs use the same program ID (for example, ORACLETD5) and multiple SAP R/3 listeners are configured with this programID, then SAP R/3 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 R/3 Gateway Server. This mechanism is proprietary to SAP R/3 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, you might suspect that you are missing SAP R/3 event handling messages.

Load balancing in server (inbound to adapter from SAP R/3) situations is handled by connecting multiple instances of the adapter to the SAP R/3 system. The SAP R/3 system then load balances the connections. You cannot tune this performance.

Load balancing in client (outbound from adapter to SAP R/3) situations is handled only by the SAP R/3 application design. If your system supports a Message Server, then you can load balance in client situations. If you have only one application server, you cannot load balance except by application server tuning, such as maximum number of connections permitted or time of day limits on connections.

The SAP R/3 system default limit is 100 RFC (communication) or adapter users. Each user takes up more than 2 MB of memory on the application server of the SAP R/3 system, and more or less on the adapter depending on the workload.

### **Connection Pooling**

A connection pool is a set of client connections to a specific destination. The pool may automatically create new connections to the specified remote system or return an existing connection. It also provides methods to return a connection back to the pool when it is no longer needed.

A connection pool can check which connections are no longer in use and can be closed to save system resources. The time period after which the pool checks the connections and the time after which a connection times out can be configured by the calling application.

A pool is always bound to one user ID and password, meaning that all connections taken from this pool use these credentials. A SAP R/3 connection is always bound to an SAP R/3 user ID and a SAP R/3 Client number.

If you log on with a pool size that is set to 1, no connection pool is created (1 userid – 1 process thread). If you log on with a pool size that is greater than 1, a pool is created with a size of *n*, which is the number you specified.

For more information about connection pooling, see the SAP JCO API documentation.

## A.2.2 Registering Your Program ID in SAPGUI

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)
- Intermediate Documents (IDoc)

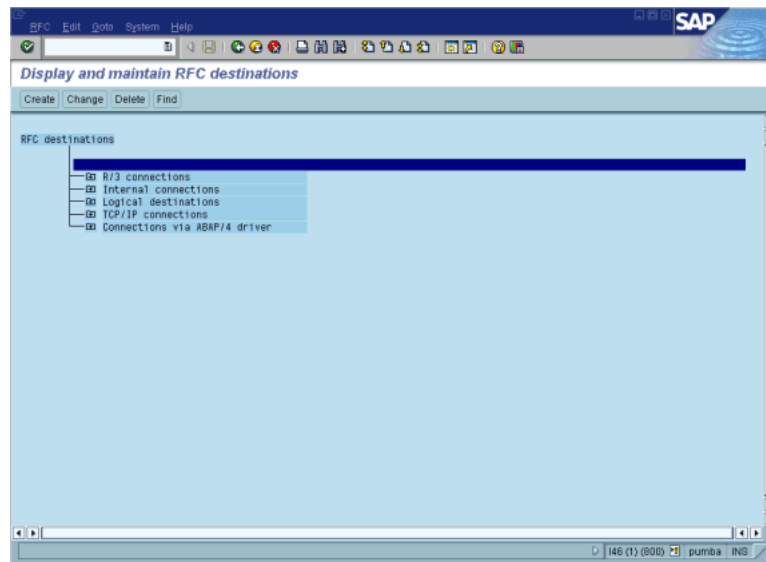
The RFC destination is a symbolic name (for example, ORACLETDS) that is used to direct events to a target system, masking the program ID. The Program ID is configured in both SAPGUI and the event adapter.

### Registering Your Program ID

To register your program ID:

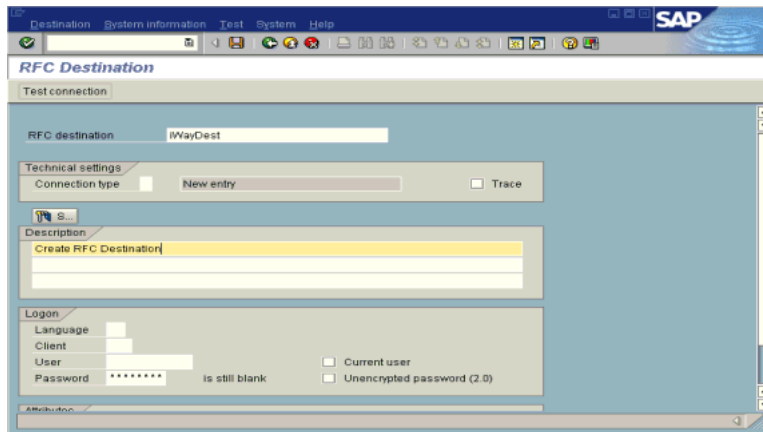
1. Launch the SAP GUI and log in to the SAP R/3 system.
2. Select **Tools, Administration, Network**, and then RFC destination.
3. Run the **SM59** transaction.

The Display and maintain RFC destinations window is displayed.



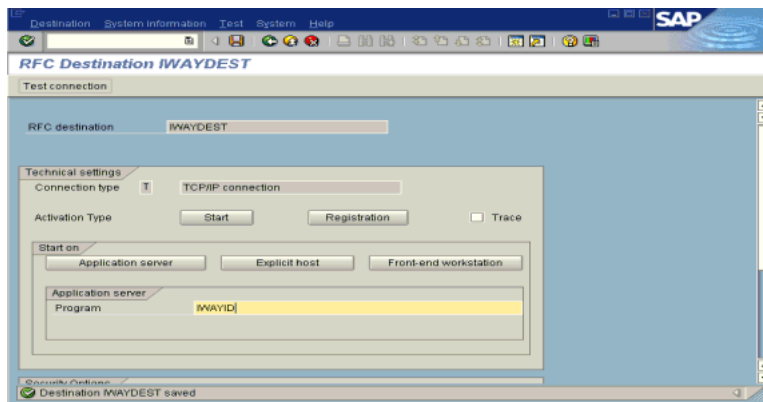
4. Select TCP/IP connections and click Create.

The RFC Destination window is displayed.



Provide the following information:

- a. In the **RFC destination** field, enter a name, for example, **ORACLETDS**.  
The value you enter in this field is case sensitive.
  - b. In the **Connection type** field, enter T for destination type TCP/IP.
  - c. In the **Description** field, enter a brief description.
5. Click Save from the tool bar or select Save from the Destination menu.  
The RFC Destination ORACLETDS window is displayed.



Perform the following steps:

- a. For the **Activation Type**, click Registration.
  - b. In the **Program** field, enter ORACLETDS.
6. Click Save from the tool bar or select Save from the Destination menu.
  7. Ensure your event adapter is running.
  8. Verify that the SAP R/3 system and Oracle Application Adapter for SAP R/3 are communicating.
  9. Click TestConnection.

### A.2.3 Testing the SAP R/3 Event Adapter

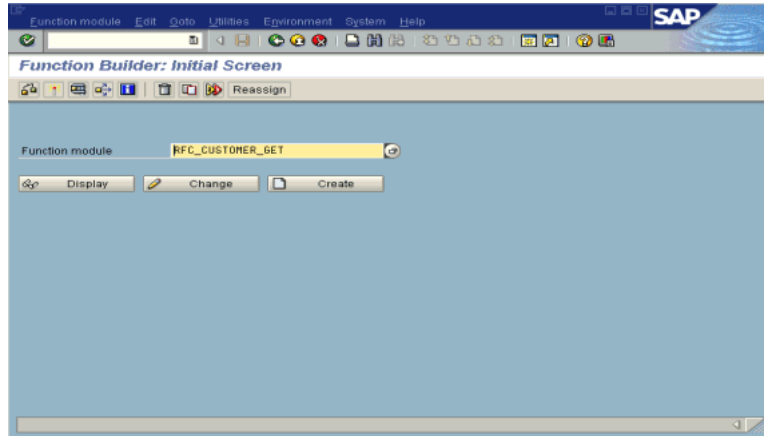
In the SAP Server, the SE37 transaction enables you to send an RFC (Remote Function Call) or a BAPI (Business Application Programming Interface) to any RFC destination.

For more information on RFC destination, see [Registering Your Program ID in SAPGUI](#) on page A-11.

### Testing the SAP R/3 Event Adapter by Sending an RFC or a BAPI Manually

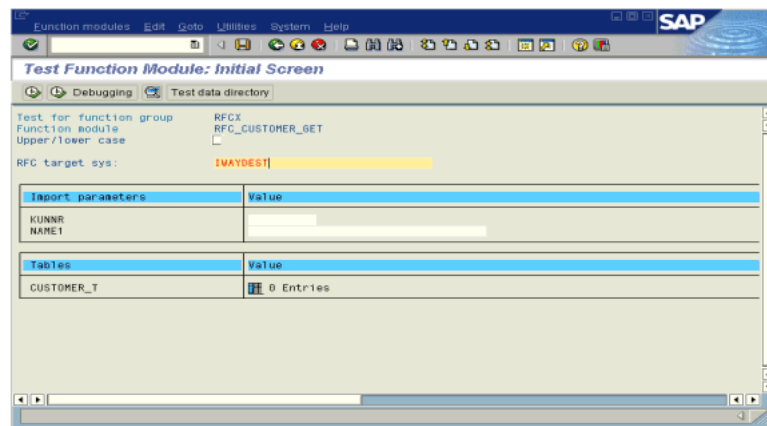
To test the SAP R/3 event adapter:

1. In the Function Builder, select a function module, for example, RFC\_CUSTOMER\_GET.



2. To choose single test, press F8 and click the Single Test icon or choose **Function module**, select Test and then Single Test.
3. Enter an RFC target system, for example, ORACLETD5.
4. Enter input data for the particular RFC modules, for example, AB\*.
5. To execute, press F8.

The Test Function Module: Initial Screen window is displayed.



6. Enter data into the SAP GUI and click Execute.

The function name and input data are transferred through RFC to create an XML document on the Oracle WebLogic Server with the parameters input in SAPGUI.

## A.2.4 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.2.5 Defining a Port

A port identifies where to send messages. This port can be used only if an RFC destination was created previously.

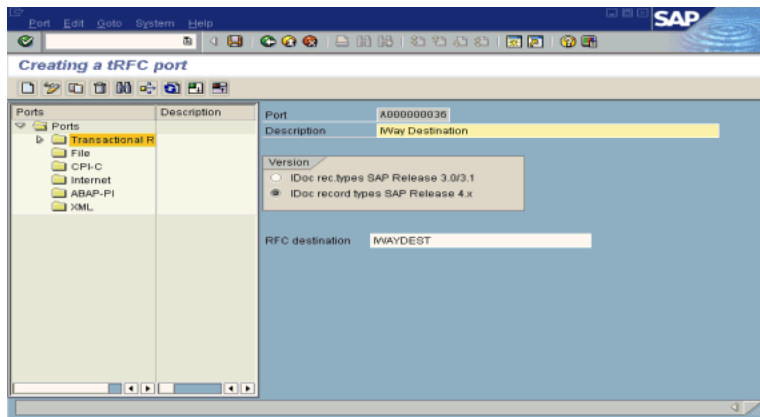
### Defining a Port

To define a port:

1. In the ALE configuration, choose **Tools, Business Communications, IDocs Basis, IDoc**, and then **Port Definition**.

You can also run the WE21 transaction.

The Creating a tRFC port window is displayed.



2. In the left pane under **Ports**, select **Transactional RFC** and click **Create**.
3. Select **Generate port name**.  
The system generates the port name.
4. Enter the IDoc version you want to send through this port.
5. Click the destination you created, for example, **ORACLETDS**.
6. Save the session, making note of the system-generated RFC port.

## A.2.6 Creating a Logical System

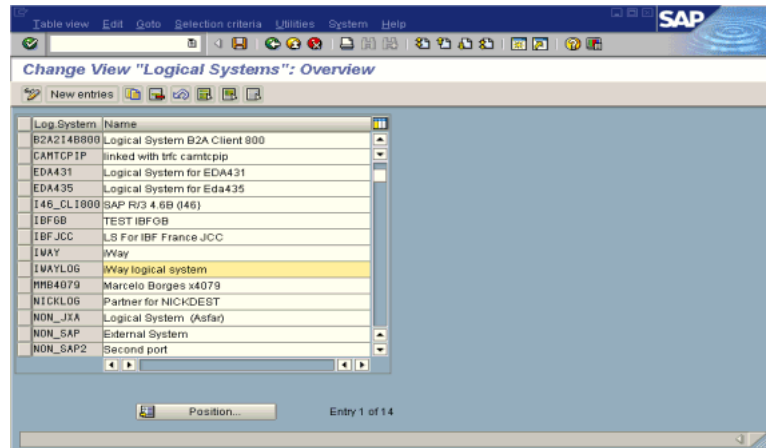
One type of partner is a logical system. A logical system manages one or more RFC destinations.

## Creating a Logical System

To create a logical system called ORACLETDS:

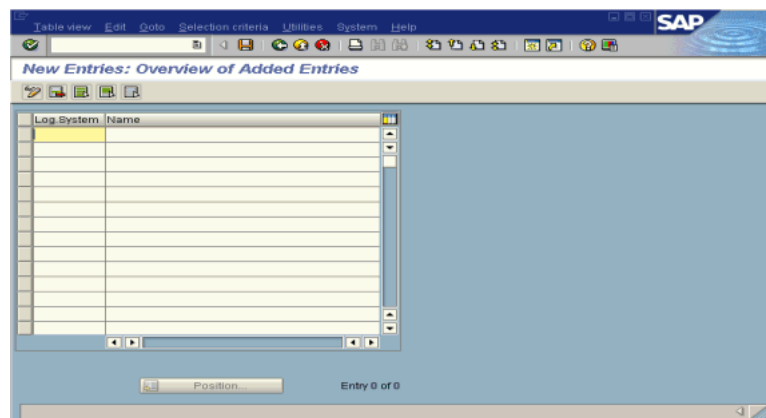
1. In the ALE configuration, enter the area menu selection SALE transaction.
2. Select SAP Reference IMG.
3. Expand the following nodes: **Basis Components, Application Link Enabling (ALE), Sending and Receiving Systems, Logical Systems,** and Define Logical System.
4. Click the check mark beside Define Logical System.

The Change View "Logical Systems": Overview window displays a list of logical systems and their names.



5. Click New entries.

The New Entries: Overview of Added Entries window is displayed with Log.System and Name columns for new log system.



6. Type an entry for Log System, for example, ORACLETDS.
7. In the **Name** column, enter a name (description) for the partner profile.
8. Click Save to save the session.

## A.2.7 Creating a Partner Profile

A partner profile is a definition of parameters for the electronic interchange of data with a trading partner using the IDoc interface.

To communicate with a partner using the IDoc interface, you must create a partner profile.

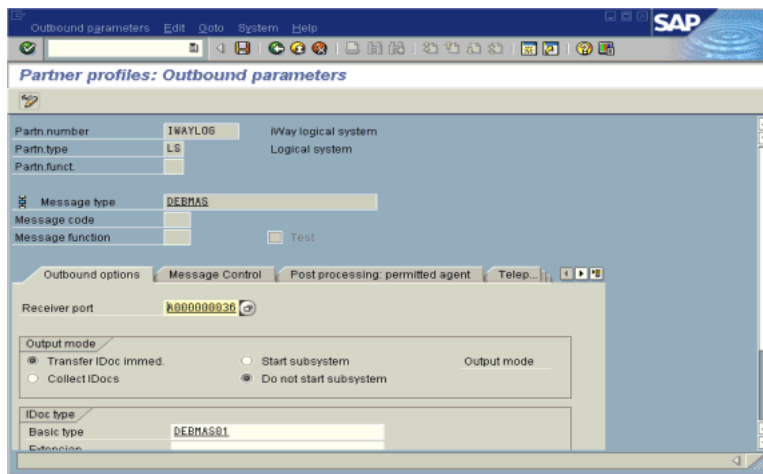
### Creating a Partner Profile

To create a partner profile:

1. In SAP GUI, choose **Tools, Business Communication, IDoc Basis, and Partner profile**.

You can also run the WE21 transaction.

The Partner profiles: Outbound parameters window is displayed and shows fields for specifying details for the partner profile.



Perform the following steps:

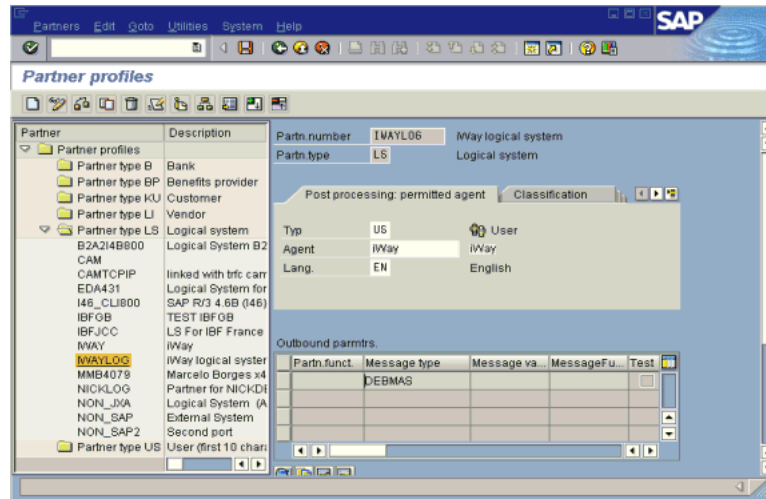
- a. Select Partner type LS (Logical system).
  - b. Press F5 (Create).
2. For Type, enter USER.
  3. For Agent, enter the current user ID, or you may select another agent type.
  4. Under the outbound parameter table control, select Create outbound parameter.  
Partner type is LS, and the Message type is DEBMA5, which is the IDoc document type.
  5. Leave Partn.funct blank.
  6. Click the Outbound options tab.

Provide the following information:

- a. Depending on your performance requirements, click Transfer IDoc Immed or Collect IDocs.
  - b. For the IDoc, enter a message type, for example, DEBMA5.
  - c. Enter a receiver port, for example, A000000036.
7. Click Save to save the session.



The Partner profiles summary window is displayed. It contains information for the logical system that you created.



## A.2.8 Collected IDocs

When using collected IDocs on any platform during inbound processing (service mode), if the DOCNUM field does not have a unique document number for each IDoc, the system creates an IDoc for each header record in the collected IDoc file and duplicates the data for each IDoc.

Make sure the DOCNUM field is included in the EDI\_DC40 structure and that each IDoc has a unique sequence number within the collected IDoc file.

## A.2.9 Creating a Distribution Model for the Partner and Message Type

You must create a distribution model for the partner and message type you designated.

### Creating a Distribution Model

To create a distribution model called ORAMOD:

1. In SAP GUI, choose **Tools, AcceleratedSAP, Customizing**, and then Project Management.

You can also run the BD64 transaction.

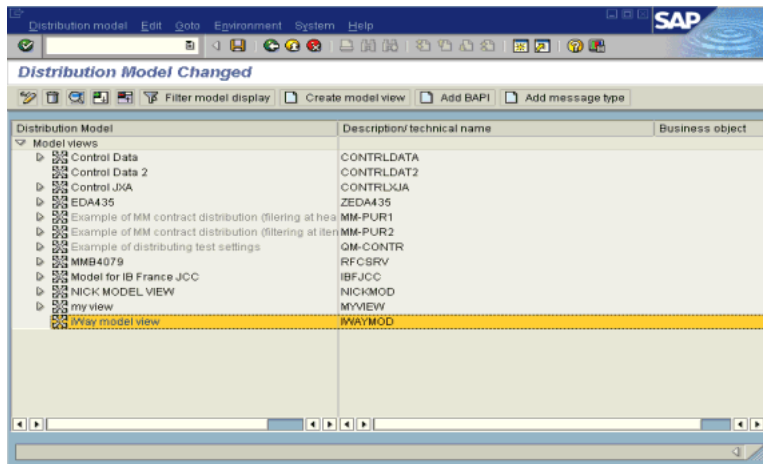
The Display Distribution Model window is displayed.

2. Select Create model view.

If required, switch the processing mode to edit within Distribution Model/Switch Processing Mode.

3. Enter a short text string and a technical name for your new model view.
4. Click Save.

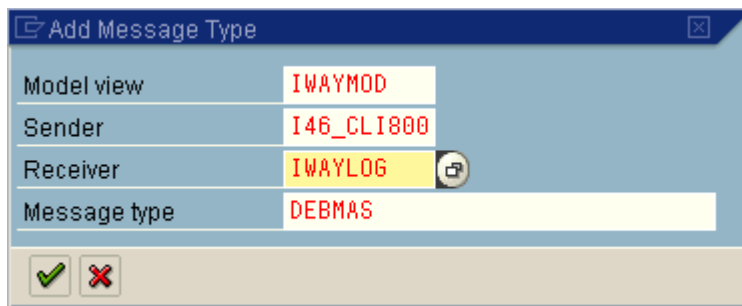
The Distribution Model Changed window is displayed, showing a tree structure of the distribution model.



Perform the following steps:

- a. In the Distribution Model tree, select a new model view.
- b. On the right, select Add message type.

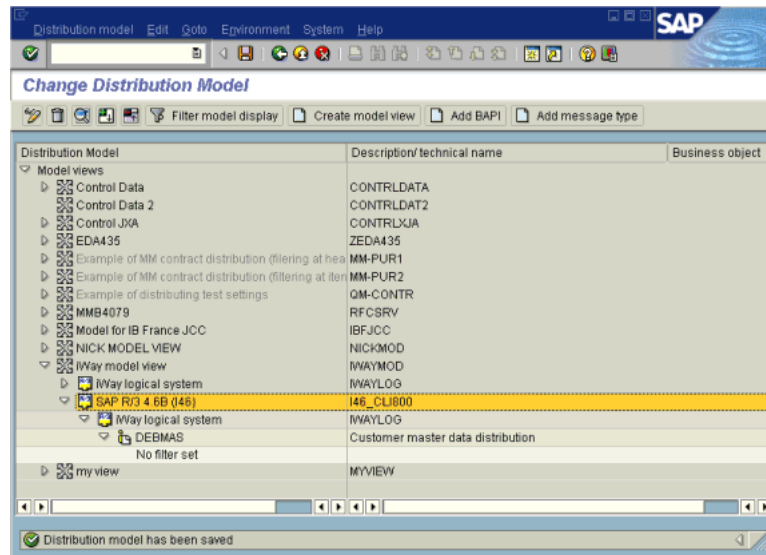
The Add Message Type box is displayed. It contains fields for specifying the sender and receiver of the message and the message type.



Provide the following information:

- a. In the **Sender** field, provide the sender that points to the SAP R/3 system, which sends the IDoc, for example, I46\_CLI800.  
In this case, the sender is an SAP 4.6B system.
- b. In the **Receiver** field, provide the logical system, for example, ORACLETDS.
- c. In the **Message type** field, provide the type of IDoc, for example, DEBMAS.
5. Click the **check mark** icon.
6. Click Save.

The Change Distribution Model window displays the new model view to use to send message type, DEBMAS, from the I46\_CLI800 SAP system to the ORACLETDS logical system.



You are now ready to test the connection to the logical system.

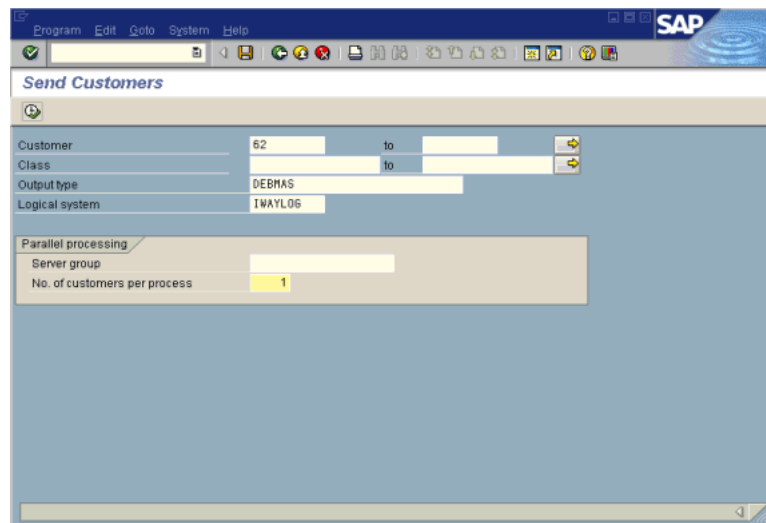
## A.2.10 Testing the SAP R/3 ALE Configuration

In the SAP Server, the BD12 transaction enables you to send IDocs to any logical system, for example, to an event adapter.

### Testing the SAP R/3 ALE Configuration

To test the SAP R/3 Application Link Embedding (ALE) configuration:

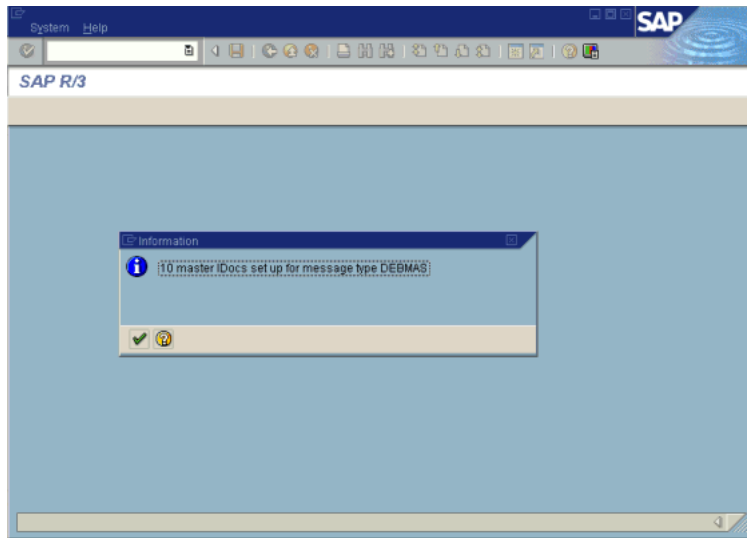
1. In the Send Customers window, enter the IDoc message type, for example, DEBMAS in the **Output type** field.



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 window is displayed.



---

---

# 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

---

ABAP programs, A-9  
access rights, 8-1  
Activation Type, A-12  
adapter configuration  
  updating, 4-2  
adapter exceptions, 7-4 to 7-6  
Adapter Lib Directory parameter, 3-3 to 3-4  
adapter types  
  resource, 1-1, 4-1 to 4-2  
adapters  
  configuring, 3-1 to 3-28, 4-2  
  deploying, 1-4, 4-1 to 4-2, 5-1  
  integrating with BPEL Process Manager, 5-1  
  troubleshooting, 7-1 to 7-7  
Adapters node, 5-39  
Add Channel dialog box, 3-24  
Add Message Type dialog box, A-6  
Add Target dialog box, 3-12  
Advanced tab, 3-15, 3-26, 5-37  
ALE (Application Link Embedding)  
  documents, A-9, A-14  
application adapters  
  configuring, 3-1 to 3-28, 4-2  
  deploying, 4-1 to 4-2, 5-1  
  integrating with BPEL Process Manager, 5-1  
  troubleshooting, 7-1 to 7-7  
Application Explorer, 1-2, 5-6, 5-35  
  application systems and, 3-11  
  channels and, 3-24  
  debugging and, 7-3  
  J2CA configuration and, 4-2  
  OracleAS Adapter J2CA and, 7-5  
  schemas and, 3-20  
  security and, 8-1, 8-6  
  testing and, 7-3  
  troubleshooting, 7-2 to 7-3  
  WSDL files and, 5-35  
Application Link Embedding (ALE)  
  documents, A-9, A-14  
Application Server dialog box, 3-12  
Application Server parameter, 3-14  
application systems  
  Application Explorer and, 3-11  
  supported, 3-11

Available list, 8-4 to 8-5

## B

---

BAPI (Business Application Programming Interfaces), 7-4  
  registering, A-11  
BAPI Exception parameter, 3-15, 3-26, 5-37  
BAPI requests, 1-4  
BPEL Designer, 5-1  
BPEL Process Manager  
  integrating with adapters, 5-1  
  OracleAS Adapter for MySAP ERP and, 5-1  
BSE (OracleAS Adapter Business Services Engine), 1-1, 1-4  
  configuring, 3-2  
  connection access to, 8-6  
  troubleshooting, 7-5  
BSE configuration page, 3-3 to 3-4  
BSE control service URL, 8-8  
BSE repositories  
  migrating, 8-8  
BSE settings window, 3-4  
BSE system settings, 3-4 to 3-6  
BSE URL field, 3-9  
Business Application Programming Interfaces (BAPI), 7-4  
  registering, A-11  
business events, 1-1  
business functions, 3-11, 3-20 to 3-22  
  creating schemas for, 3-20  
business processes, 1-3  
business services  
  creating, 3-21 to 3-22  
  deploying, 8-1  
  testing, 3-22  
Business Services node, 8-2 to 8-5, 8-6

## C

---

CCI calls, 4-1  
Change Distribution Model window, A-6  
channels, 3-24  
  Application Explorer and, 3-24  
  creating, 3-24, 5-35 to 5-38  
  deleting, 3-28

- editing, 3-28
- starting, 3-28, 5-38
- stopping, 3-28, 5-38
- testing and debugging and, 5-38
- Channels node, 3-25, 5-35
- channels. *See also* listeners
- Client parameter, 3-13, 3-26, 5-37
- client programs
  - RFC Client, A-9
- Code page parameter, 3-13, 3-26, 5-37
- Commit with wait parameter, 3-15, 3-26, 5-37
- Configuration node under Business
  - Services, 8-2 to 8-5, 8-6
- configuration parameters, 4-4
  - IWayConfig, 4-4
  - IWayHome, 4-4
  - IWayRepoPassword, 4-4
  - IWayRepoURL, 4-4
  - IWayRepoUser, 4-4
  - Loglevel, 4-4
- configurations
  - connecting to, 3-11
  - defining, 3-9 to 3-10
  - overwriting, 4-4
- Configurations node, 3-9
- configuring adapters, 3-1 to 3-28, 4-2
- configuring BSE system settings, 3-4 to 3-6
- configuring repositories, 3-6
- connecting to MySAP ERP, 3-11 to 3-18, 5-7, 5-38 to 5-39, 7-2, 7-7
- connecting to OracleAS Adapter J2CA, 7-5
- Connection dialog box, 3-18, 5-39
- connection parameters, 3-13 to 3-18, 3-26 to 3-27, 7-2
  - Advanced, 3-15, 3-26, 5-37
  - Application Server, 3-14
  - BAPI Exception Handling, 3-15
  - Client, 3-13, 3-26, 5-37
  - Code page, 3-13
  - Commit with wait, 3-15
  - Connection pool name, 3-15
  - Connection pool size, 3-15
  - EDI Version, 3-14
  - Language, 3-13
  - Logon ticket (SSO2), 3-16
  - Logon ticket (X509), 3-16
  - MySAP ERP Trace, 3-13
  - Password, 3-13, 3-26, 5-37
  - Port, 3-2 to 3-4
  - Security, 3-16
  - SNC level, 3-16
  - SNC library path, 3-17, 3-27, 5-38
  - SNC mode, 3-16
  - SNC name, 3-16
  - SNC partner, 3-16
  - System, 3-13, 3-26, 5-36
  - System Number, 3-14
  - User, 3-13, 3-26, 5-37
- Connection pool name parameter, 3-15, 3-26, 5-37
- Connection pool size parameter, 3-15, 3-26, 5-37
- Connection type field, A-11

- connections
  - closing, 3-18
  - deleting, 3-18
  - establishing, 3-11 to 3-18, 5-39
- Connector
  - deploying to Oracle Application Server, 4-1
- connector factories, 4-2
- connector factory objects, 4-2
  - multiple, 4-4
- control methods, 8-9
- Create Model View dialog box, A-5
- Create Outbound JCA Service
  - (Request/Response), 5-7
- Create Request dialog box, A-3
- Create Web Service dialog box, 3-22
- creating channels, 5-35 to 5-38
- creating events, 5-38 to 5-41
- creating repository projects, 3-8 to 3-10
- customer master data, 5-55

## D

---

- data exchange
  - partner profiles and, A-7
- database connections
  - opening, 4-4
- databases
  - connecting to, 4-4
  - Oracle, 4-4
- DEBMS05
  - locating, 5-38
- Debug Level parameter, 3-5
- deploying adapters, 4-1 to 4-2, 5-1
- Description field, 3-12, 3-22, 8-3 to 8-5, 8-7, A-11
- design time, 5-35, 8-8
  - configuring, 5-6, 5-35
- Destination parameter, A-9
- disconnecting from MySAP ERP, 3-18
- Display and maintain RFC Destination
  - window, A-11
- Display Distribution Model window, A-4
- Display IMG window, A-2
- DNS Lookup option, 8-7
- DNS name, 8-7
- Domain Name System (DNS), 8-7
- domain names, 8-7
- Domain option, 8-7

## E

---

- Eclipse. *See* JDeveloper
- EDI Version parameter, 3-14
- editing channels, 3-28
- editing targets, 3-19
- EJB (Enterprise Java Beans), 4-1
- Encoding parameter, 3-5
- Enterprise Connector for J2EE Connector Architecture
  - (J2CA), 1-4
- Enterprise Java Beans (EJB), 4-1
- error messages, 7-2 to 7-7



- target systems and, 7-6
- event adapters
  - configuring, 3-24
  - testing, A-12
- event handling, A-1, A-9
- event integration
  - verifying, 5-55
- Event Manager, A-9
- event messages, 5-35
- event ports
  - creating, 5-38 to 5-41
- events, 1-1, 3-24, A-1
  - configuring, 3-24
  - creating, 5-38 to 5-41, A-9
  - triggering, 5-55
- Execution Denied list, 8-6
- Execution Granted list, 8-6
- Existing Service Names list, 3-22
- Export WSDL dialog box, 5-40

## F

---

- fault code elements, 7-6
- fault string elements, 7-6
- file system repositories
  - configuring, 3-6
- Function Builder, A-13
- functions, A-9

## G

---

- Gateway Server, A-10
- Grant Access check box, 8-7
- Group (of Computers) option, 8-7
- Group node, 8-4
- groups
  - creating, 8-3

## H

---

- Hostname parameter, 3-4, 3-9

## I

---

- IDoc (Intermediate Documents), 7-4, A-1
  - defining, A-1
  - registering, A-11
  - saving, A-1
  - transferring, A-1
- IDoc interface, A-1
- IDoc requests, 1-4
- in, 3-26
- inbound IDoc
  - configuring, A-1
- inbound J2CA services, 5-35
- inbound MySAP ERP processing, A-1
  - configuring, A-1
- Inbound options tab, A-8
- input XML documents, 7-4
- installation directories, 4-4
- instances of policy types, 8-1

- Intermediate Documents (IDoc), 7-4, A-1
  - defining, A-1
  - registering, A-11
  - saving, A-1
  - transferring, A-1
- IP (Mask)/Domain field, 8-7
- IP addresses, 8-7
- IP and Domain Restriction policy type, 8-6
- IWayConfig parameter, 4-4
- IWayHome parameter, 4-4
- IWayRepoPassword parameter, 4-4
- IWayRepoURL parameter, 4-4
- IWayRepoUser parameter, 4-4

## J

---

- J2CA (Enterprise Connector for J2EE Connector Architecture), 1-4
  - OracleAS Adapter and, 4-1
- J2CA configuration
  - Application Explorer and, 4-2
- J2CA repositories
  - migrating, 8-10
- J2CA resource adapters, 1-1, 4-1
- J2CA services, 5-35
- Java program clients, 4-1
- JDeveloper, 5-1

## L

---

- Language parameter, 3-5, 3-13, 3-26, 5-37
- librfc32.dll file, 7-2
- License and Method dialog box, 3-22
- License field, 3-22
- licenses, 3-22
- list of nodes, 7-2
- listeners, 1-4, 3-11, A-10
- listeners. *See also* channels
- load balancing, A-10
- load balancing algorithms, A-10
- locating DEBMA505, 5-38
- log files, 7-1
- log levels
  - overwriting, 4-4
- logical systems, 5-55, A-1 to A-6
  - configuring, A-1
  - defining, A-2
  - unique IDs, A-1 to A-2
- Loglevel parameter, 4-4
- Logon ticket (SS02) parameter, 3-16
- Logon ticket (X509) parameter, 3-16
- Log.System column, A-3

## M

---

- managed connector factories, 4-2
- managed connector factory objects, 4-2
  - multiple, 4-4
- ManagedConnectionFactory parameter, 4-4
- Mediator Inbound Process, 6-27
- Mediator Outbound Process, 6-6

- Message Server dialog box, 3-24
- Message Type field, A-8
- message types
  - event, 5-35
- messages, 1-1
- metadata
  - storing, 3-5, 8-8
- Method Name field, 3-22
- methods, 8-1
- migrating repositories, 8-8
- migrating Web services, 8-10
- My SAP ERP System port, A-1
- MySAP Easy Access window, 5-55
- MySAP ERP
  - connecting to, 1-3, 3-11 to 3-18, 5-7, 5-38 to 5-39, 7-2, 7-7
  - disconnecting from, 3-18
  - integrating, 1-3
- MySAP ERP business functions, 3-11, 3-20 to 3-22
  - creating schemas for, 3-20
- MySAP ERP business processes, 1-3
- MySAP ERP event adapters
  - configuring, A-14
  - testing, A-13
- MySAP ERP event handling, A-9 to A-10
- MySAP ERP Event Manager, A-9
- MySAP ERP function modules
  - routing, A-9
- MySAP ERP Gateway server, A-9
- MySAP ERP inbound processing, A-1
- MySAP ERP outbound processing, A-9
- MySAP ERP platforms, 1-3
- MySAP ERP program ID (identifiers)
  - BAPI and, A-10
  - multiple remote function calls and, A-10
  - registering, A-11 to A-12
  - RFC Destination and, A-9
- MySAP ERP Trace parameter, 3-13, 3-26, 5-37
- MySAP node, 5-39
- MySAP program ID (identifiers), 5-55
- MySAP Workbench, 5-55

## N

---

- Name field, 3-12, 8-3 to 8-5
- New Configuration dialog box, 3-9 to 3-10
- New Group dialog box, 8-4
- New Policy permissions dialog box, 8-5
- New User dialog box, 8-2
- Node list, 7-2
- nodes
  - Adapters, 5-39
  - Business Services, 8-2 to 8-5
  - Channels, 3-25, 5-35
  - Configuration under Business Services, 8-2 to 8-5, 8-6
  - Configurations, 3-9
    - connected, 3-18, 5-39
    - disconnected, 3-18
  - Group, 8-4

- MySAP, 5-39
- Policies, 8-5
- Ports, 5-35
- Security, 8-5, 8-6
- Users, 8-3
  - Users and Groups, 8-2 to 8-3
- Number of Async. Processors parameter, 3-5

## O

---

- Object type, 7-4
- OC4J-ra.xml file, 4-2 to 4-4
- Oracle Application Server
  - deployment of Connector to, 4-1
- Oracle databases, 4-4
- Oracle JDeveloper, 5-1
- Oracle JDeveloper BPEL Designer. *See* BPEL Designer, JDeveloper, or Oracle JDeveloper
- Oracle repositories
  - migrating, 8-8
- Oracle's Unified Method (OUM), viii
- OracleAS Adapter
  - installation directory and, 4-4
  - J2CA and, 4-1
- OracleAS Adapter Application Explorer. *See* Application Explorer
- OracleAS Adapter Business Services Engine (BSE), 1-1, 1-4
  - configuring, 3-2
  - troubleshooting, 7-5
- OracleAS Adapter for MySAP ERP
  - BPEL Process Manager and, 5-1
  - configuring, 3-1 to 3-28
  - deploying, 1-1, 5-1
  - troubleshooting, 7-1 to 7-7
- OracleAS Adapter J2CA
  - Application Explorer and, 7-5
  - connecting to, 7-5
- outbound MySAP ERP processing, A-1 to A-9
  - configuring, A-9

## P

---

- parameter types
  - channel configuration, 3-28
  - configuration, 4-4
  - connection, 3-26 to 3-27, 7-2
  - repository, 3-5
  - repository migration, 8-9
  - system, 3-5
- partner profiles
  - data exchange and, A-7
  - defining, A-7
- Partner profiles window, A-7
- Partner type LS, A-7
- Partn.number field, A-7
- Password parameter, 3-3, 3-13, 3-26, 5-37, 8-3
- passwords, 3-3, 3-13, 3-26, 4-4, 5-37, 8-3
- permissions, 8-1
  - configuring, 7-5

- denying, 8-6
- granting, 8-6
- policies, 8-1
  - applying, 8-1
  - creating, 8-4
- Policies node, 8-5
- policy types
  - instances of, 8-1
    - IP and Domain Restriction, 8-6
- policy-based security, 8-1 to 8-7
- port definitions, A-1
- Port Number parameter, 3-9
- Port parameter, 3-2 to 3-4
- ports
  - creating, 5-38 to 5-41
- Ports node, 5-35
- privileges, 8-1
  - setting, 8-1
- Process code field, A-8
- Process Manager. *See* BPEL Process Manager
- Processing by function module area, A-9
- Program field, A-12
- program ID (identifiers), 5-55
  - BAPI and, A-10
  - multiple remote function calls and, A-10
  - registering, A-11 to A-12
  - RFC Destination and, A-9
- Prompt for Workbench request dialog box, A-3
- properties, 4-4
- pseudo events
  - Remote Function Calls (RFC) and, A-9

## R

---

- registering program ID (identifiers), A-11 to A-12
- Remote Function Calls (RFC)
  - pseudo events and, A-9
  - registering, A-11
- Remote Function module, 7-4
- repositories
  - configuring, 3-6
  - migrating, 8-8
- Repository Driver parameter, 3-5
- repository information
  - storing, 3-6
- repository migration parameters, 8-9
- repository parameters
  - Driver, 3-5
  - Password, 3-5
  - Pooling, 3-5
  - Type, 3-5
  - URL, 3-5
  - User, 3-5
- Repository Password parameter, 3-5
- Repository Pooling parameter, 3-5
- repository projects
  - creating, 3-8 to 3-10
  - Web services and, 3-8
- Repository Type parameter, 3-5
- Repository URL parameter, 3-4, 3-5

- repository URLs, 8-10
- Repository User parameter, 3-5
- repository.xml file, 8-11
- request processing
  - BAPI, 1-4
  - IDoc, 1-4
  - RFC, 1-4
- request schemas, 3-20
- requests
  - executing, 7-4
- resource adapters, 4-1 to 4-2
  - deploying, 1-1
- Resource Execution policy type, 8-1
- response schemas, 3-20
- Restrictions page, 5-55
- RFC (Remote Function Calls)
  - pseudo events and, A-9
  - registering, A-11
- RFC API
  - RFC calls and, A-9
- RFC Client program, A-9
- RFC Destination, A-9 to A-11
- RFC Destination field, A-11
- RFC Destination window, A-11
- RFC functions, A-9
- RFC programs, A-9
  - Client, A-9
  - Server, A-9
- RFC requests, 1-4
- RFC target systems, A-13
- runtime, 8-8

## S

---

- SAP GUI, A-11, A-14
- sapjco.jar file, 7-2 to 7-3
- sapjcorfc.dll file, 7-2
- schemas, A-10
  - Application Explorer and, 3-20
  - creating, 3-20
- SE37 transaction, A-12
- security, 8-1 to 8-7
  - Application Explorer and, 8-1, 8-6
  - configuring, 8-2
- Security node, 8-2 to 8-5, 8-6
- security policies
  - applying, 8-1
  - creating, 8-4
- Security Policy option, 8-6
- Security tab, 3-16
- Selected list, 8-4 to 8-5
- Send Customers window, 5-55
- server programs
  - RFC Client, A-9
  - RFC Server, A-9
- service names, 3-22
- Service Provider list, 3-9 to 3-10
- Service-Oriented Architecture (SOA), 5-1
- services, 1-1
  - creating, 3-21 to 3-22

- testing, 3-22
- servlets, 4-1
- Single (Computer) option, 8-7
- SM59 transaction, A-9 to A-11
- SNC level parameter, 3-16
- SNC library path parameter, 3-16
- SNC mode parameter, 3-16
- SNC name parameter, 3-16
- SNC partner parameter, 3-16
- SOA (Service-Oriented Architecture), 5-1
- SOAP agents, 7-6
- SOAP envelopes, 8-9
- SOAP faults, 7-6
- soap operation name dialog box, 8-8
- SOAP requests, 7-6 to 7-7, 8-1, 8-5
  - creating, 8-8
  - errors and, 7-6 to 7-7
- storing metadata, 8-8
- System Number parameter, 3-14
- system parameters
  - Adapter Lib Directory, 3-5
  - Debug Level, 3-5
  - Encoding, 3-5
  - Language, 3-5
  - Number of Async. Processors, 3-5
- system settings
  - configuring, 3-4 to 3-6
- System tab, 3-13, 3-26, 5-36

## T

---

- target systems
  - errors and, 7-6
- Target Type list, 3-12
- targets
  - connecting to, 3-18, 5-39, 7-7
  - defining, 3-12
  - deleting, 3-18
  - disconnecting from, 3-18
  - editing, 3-19
- Test Function Module Initial Screen window, A-13
- testing event adapters, A-13
- testing Web services, 3-22
- trace information, 7-1
- transaction processing, 1-1
- transactions
  - storing, 3-5
- trigger by background program option, A-9
- trigger immediately option, A-9
- triggering events in MySAP ERP, 5-55
- troubleshooting, 7-1 to 7-7
  - Application Explorer, 7-2 to 7-3
  - BSE, 7-5
  - Web services, 7-5 to 7-7
- Type list, 8-7

## U

---

- unique IDs, A-1
- updating adapter configuration, 4-2

- User parameter, 3-3, 3-13, 3-26, 5-37, 7-5
- User tab, 3-13, 3-26, 5-37
- users
  - associating, 8-2
- Users and Groups node, 8-2 to 8-3
- Users node, 8-3

## W

---

- Web Service Definition Language (WSDL), 3-20 to 3-21
- Web service names, 3-22
- Web service projects
  - creating, 3-8
- Web services, 1-1
  - creating, 3-21 to 3-22
  - delivering, 3-5
  - deploying, 7-5 to 7-7, 8-1
  - integrating, 5-1
  - migrating, 8-10
  - repository projects and, 3-8
  - testing, 3-22
  - troubleshooting, 7-5 to 7-7
- Web services policy-based security, 8-1 to 8-7
- WSDL (Web Service Definition Language), 3-20 to 3-21
- WSDL documents, 5-1
- WSDL file location dialog box, 8-8
- WSDL files, 5-1
  - Application Explorer and, 5-35

## X

---

- XML documents
  - input, 7-4
- XML messages, 1-1
- XML schemas
  - creating, 3-20