Oracle®

Application Adapter for J.D. Edwards OneWorld User's Guide

WebLogic Server 10*g* Release 3 (10.3.1.0)

E15305-01

June 2009

Oracle Application Adapter for J.D. Edwards OneWorld User's Guide, WebLogic Server 10*g* Release 3 (10.3.1.0)

E15305-01

Primary Author: Stefan Kostial

Contributing Authors: Sheela Vasudevan, Sunil Gopal, Marian Jones, Vikas Anand, Sunil Wadhwa, Vishal Saxena

Contents

7 Advanced User Tools

A Configuring J.D. Edwards OneWorld for Outbound and Inbound Processing

B Sample Files

Glossary

Index

Preface

This Preface contains these topics:

- Audience
- Documentation Accessibility
- Related Documents
- Conventions
- Help Us to Serve You Better

Audience

Oracle Application Adapter for J.D. Edwards OneWorld User's Guide (WebLogic Server 10gr3) is intended for those who perform the following tasks:

- Install applications
- Maintain applications

To use this document, you need to know how to install and configure Oracle Service Bus (Business Service and Proxy Service).

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.

TTY Access to Oracle Support Services

To reach AT&T Customer Assistants, dial 711 or 1.800.855.2880. An AT&T Customer Assistant will relay information between the customer and Oracle Support Services at 1.800.223.1711. Complete instructions for using the AT&T relay services are available at http://www.consumer.att.com/relay/tty/standard2.html. After the AT&T Customer Assistant contacts Oracle Support Services, an Oracle Support Services engineer will handle technical issues and provide customer support according to the Oracle service request process.

Related Documents

For more information, refer to these Oracle resources:

- *Oracle Application Server Adapter Concepts*
- *Oracle Application Server Adapters Installation Guide*

Printed documentation is available for sale in the Oracle Store at

`http://oraclestore.oracle.com/`

To download free release notes, installation documentation, white papers, or other collateral, please visit the Oracle Technology Network (OTN). You must register online before using OTN; registration is free and can be done at

`http://www.oracle.com/technology/membership/`

If you already have a user name and password for OTN, then you can go directly to the documentation section of the OTN Web site at

`http://www.oracle.com/technology/documentation/`

Conventions

This section describes the conventions used in the text and code examples of this documentation set. It describes:

- Conventions in Text
- Conventions in Code Examples
- Conventions for Windows Operating Systems

Conventions in Text

We use the following conventions in text to help you more quickly identify special terms. The table also provides examples of their use.

Convention	Meaning	Example
Bold	Bold typeface indicates terms that are defined in the text or terms that appear in a glossary, or both.	When you specify this clause, you create an **index-organized table**.

Convention	Meaning	Example
Italics	Italic typeface indicates book titles or emphasis.	*Oracle Database Concepts* Ensure that the recovery catalog and target database do *not* reside on the same disk.
`UPPERCASE monospace (fixed-width) font`	Uppercase monospace typeface indicates elements supplied by the system. Such elements include parameters, privileges, datatypes, Recovery Manager keywords, SQL keywords, SQL*Plus or utility commands, packages and methods, and system-supplied column names, database objects and structures, user names, and roles.	You can specify this clause only for a `NUMBER` column. You can back up the database by using the `BACKUP` command. Query the `TABLE_NAME` column in the `USER_TABLES` data dictionary view. Use the `DBMS_STATS.GENERATE_STATS` procedure.
`lowercase monospace (fixed-width) font`	Lowercase monospace typeface indicates executable programs, filenames, directory names, and sample user-supplied elements. *Note:* Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown.	Enter `sqlplus` to start SQL*Plus. The password is specified in the `orapwd` file. Back up the datafiles and control files in the `/disk1/oracle/dbs` directory. The `department_id`, `department_name`, and `location_id` columns are in the `hr.departments` table. Connect as `oe` user. The `JRepUtil` class implements these methods.
`lowercase italic monospace (fixed-width) font`	Lowercase italic monospace font represents placeholders or variables.	You can specify the *`parallel_clause`*. Run *`old_release`*`.SQL` where *`old_release`* refers to the release you installed before upgrading.

Conventions in Code Examples

Code examples illustrate SQL, PL/SQL, SQL*Plus, or other command-line statements. They are displayed in a monospace (fixed-width) font and separated from normal text as shown in this example:

```
SELECT username FROM dba_users WHERE username = 'MIGRATE';
```

The following table describes typographic conventions used in code examples and provides examples of their use.

<table>
<tr><th>Convention</th><th>Meaning</th><th>Example</th></tr>
<tr><td><code>[ ]</code></td><td>Anything enclosed in brackets is optional.</td><td><code>DECIMAL (<i>digits</i> [, <i>precision</i>])</code></td></tr>
<tr><td><code>{ }</code></td><td>Braces are used for grouping items.</td><td><code>{ENABLE | DISABLE}</code></td></tr>
<tr><td>|</td><td>A vertical bar represents a choice of two options.</td><td><code>{ENABLE | DISABLE}</code>
<code>[COMPRESS | NOCOMPRESS]</code></td></tr>
<tr><td><code>. . .</code></td><td>Ellipsis points mean repetition in syntax descriptions.
In addition, ellipsis points can mean an omission in code examples or text.</td><td><code>CREATE TABLE ... AS <i>subquery</i>;</code>
<code>SELECT <i>col1</i>, <i>col2</i>, ... , <i>coln</i> FROM employees;</code></td></tr>
<tr><td>Other symbols</td><td>You must use symbols other than brackets ([ ]), braces ({ }), vertical bars (|), and ellipsis points (...) exactly as shown.</td><td><code>acctbal NUMBER(11,2);</code>
<code>acct   CONSTANT NUMBER(4) := 3;</code></td></tr>
</table>

Convention	Meaning	Example

Italics	Italicized text indicates placeholders or variables for which you must supply particular values.	`CONNECT SYSTEM/`*`system_password`*
		`DB_NAME = `*`database_name`*

`UPPERCASE`	Uppercase typeface indicates elements supplied by the system. We show these terms in uppercase to distinguish them from terms you define. Unless terms appear in brackets, enter them in the order and with the spelling shown. Because these terms are not case sensitive, you can use them in either UPPERCASE or lowercase.	`SELECT last_name, employee_id FROM employees;`
		`SELECT * FROM USER_TABLES;`
		`DROP TABLE hr.employees;`

`lowercase`	Lowercase typeface indicates user-defined programmatic elements, such as names of tables, columns, or files.
	Note: Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown.
		`SELECT last_name, employee_id FROM employees;`
		`sqlplus hr`
		`CREATE USER mjones IDENTIFIED BY ty3MU9;`

Conventions for Windows Operating Systems

The following table describes conventions for Windows operating systems and provides examples of their use.

Convention	Meaning	Example
Click **Start**, and then choose the ***menu item***	How to start a program.	To start the Database Configuration Assistant, click **Start**, and choose **Programs**. In the Programs menu, choose **Oracle** - ***HOME_NAME*** and then click **Configuration and Migration Tools**. Choose **Database Configuration Assistant**.
File and directory names	File and directory names are not case sensitive. The following special characters are not allowed: left angle bracket (<), right angle bracket (>), colon (:), double quotation marks ("), slash (/), pipe (|), and dash (-). The special character backslash (\) is treated as an element separator, even when it appears in quotes. If the filename begins with \\, then Windows assumes it uses the Universal Naming Convention.	c:\winnt"\"system32 is the same as C:\WINNT\SYSTEM32
`C:\>`	Represents the Windows command prompt of the current hard disk drive. The escape character in a command prompt is the caret (^). Your prompt reflects the subdirectory in which you are working. Referred to as the *command prompt* in this manual.	`C:\oracle\oradata>`
Special characters	The backslash (\) special character is sometimes required as an escape character for the double quotation mark (") special character at the Windows command prompt. Parentheses and the single quotation mark (') do not require an escape character. Refer to your Windows operating system documentation for more information on escape and special characters.	`C:\>exp HR/HR TABLES=employees QUERY=\"WHERE job_id='SA_REP' and salary<8000\"`

Convention	Meaning	Example
HOME_NAME	Represents the Oracle home name. The home name can be up to 16 alphanumeric characters. The only special character allowed in the home name is the underscore.	`C:\> net start Oracle`*`HOME_NAME`*`TNSListener`
ORACLE_HOME and *ORACLE_BASE*	In Oracle8i release 8.1.3 and lower, when you installed Oracle components, all subdirectories were located under a top level *ORACLE_HOME* directory. This release complies with Optimal Flexible Architecture (OFA) guidelines. All subdirectories are not under a top level *ORACLE_HOME* directory. There is a top level directory called *ORACLE_BASE* that by default is `C:\oracle\product\10.1.0`. If you install the latest Oracle release on a computer with no other Oracle software installed, then the default setting for the first Oracle home directory is `C:\oracle\product\10.1.0\db_`*n*, where *n* is the latest Oracle home number. The Oracle home directory is located directly under *ORACLE_BASE*. All directory path examples in this guide follow OFA conventions. Refer to *Oracle Database Installation Guide for Windows* for additional information about OFA compliances and for information about installing Oracle products in non-OFA compliant directories.	Change to the *ORACLE_BASE*`\`*ORACLE_HOME*`\rdbms\admin` directory.

Help Us to Serve You Better

To help our consultants answer your questions effectively, please be prepared to provide specifications and sample files and to answer questions about errors and problems.

The following list includes the specifications our consultants require.

- **Platform**:
- **Operating System**:
- **Operating System Version**:
- **Product List**:
- **Adapters**:
- **Adapter Deployment**:

  For example, *J2CA* or *Business Services Engine (BSE)*
- **Container Version**:

The following table lists components. Specify the version in the column provided.

Component	Version
Adapter	
EIS (DBMS/APP)	
HOTFIX/Service Pack	

In the following table, specify the JVM version and vendor.

JVM Version	Vendor

The following table lists additional questions to help us serve you better.

Request/Question	Error/Problem Details or Information
Provide usage scenarios or summarize the application that produces the problem.	
Has this happened previously?	
Can you reproduce this problem consistently?	
Any **change in the application environment**: software configuration, EIS/database configuration, application, and so on?	
Under what circumstance does the problem *not* occur?	
Describe the **steps** to reproduce the problem.	
Describe the **problem**.	
Specify the **error** message(s).	

The following is a list of error or problem files that might be applicable.

- XML schema
- XML instances
- Other input documents (transformation)
- Error screen shots
- Error output files
- Trace and log files
- Log transaction

1 Introduction

OracleWLS Application Adapter for J.D. Edwards OneWorld provides connectivity and carries out interactions on a J.D. Edwards OneWorld system. This chapter provides information about OracleWLS Application Adapter for J.D. Edwards OneWorld to help you accomplish your integration projects.

This chapter discusses the following topics:

- Adapter Features
- J.D. Edwards OneWorld Platforms, Products, and Releases Supported
- J.D. Edwards OneWorld Concepts
- Integration with J.D. Edwards OneWorld
- Adapter Architecture
- BSE Versus OracleWLS Adapter J2CA Deployment

Adapter Features

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

OracleWLS Application Adapter for J.D. Edwards OneWorld can be deployed as a J2EE Connector Architecture (J2CA) 1.0 resource adapter. This deployment is referred to as OracleWLS J2CA adapter. It can also be deployed as a Web services servlet and is referred to as OracleWLS Adapter Business Services Engine (BSE).

OracleWLS Application Adapter for J.D. Edwards OneWorld uses XML messages to enable non-J.D. Edwards applications to communicate and exchange transactions with J.D. Edwards using services and events. Services and events are described as follows:

- Services: Enables applications to initiate a J.D. Edwards business event.
- Events: Enables applications to access J.D. Edwards data only when a J.D. Edwards business event occurs.

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

The channel is the adapter component that receives events in real time from the Enterprise Information System (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.

OracleWLS Application Adapter for J.D. Edwards OneWorld provides:

- XML schemas and WSDLs for the J2CA 1.0 and 1.5 resource adapter.
- Web services for BSE.

See Also: *Oracle WebLogic Server Adapter Concepts*

J.D. Edwards OneWorld Platforms, Products, and Releases Supported

The following table indicates which combinations of adapter platforms and J.D. Edwards OneWorld platforms are supported, and for each combination, which J.D. Edwards OneWorld products and releases are supported.

Adapter Platform	J.D. Edwards OneWorld Platform	J.D. Edwards OneWorld Product and Release
Windows, AS400, HP 9000/B, Sun or IBM RS/6000	Windows, AS400, HP 9000/B, Sun or IBM RS/6000	■ XE (B7333) from SP19 to SP23 ■ ERP 8.0 (B7334) ■ EnterpriseOne B9 (8.9) ■ EnterpriseOne 8.10 (with Tools release 8.93 and 8.94) ■ EnterpriseOne 8.11 (SP1 and Tools Release 8.95) ■ EnterpriseOne 8.12 (Tools Release 8.96 2.0)

J.D. Edwards OneWorld Inbound WSDL Generation on UNIX Platforms

On UNIX platforms, manual inbound WSDL documents can be generated for J.D. Edwards OneWorld.

J.D. Edwards OneWorld Unit Of Work (UOW)

The following section provides information on the J.D. Edwards OneWorld Unit of Work (UOW).

1. iWay recommends the customer to generate individual business functions of J.D. Edwards OneWorld and then group them together.
2. Generating the individual business functions and grouping them together is completely based on the experience and knowledge related to the business functions of J.D. Edwards OneWorld. There is no documentation on generating the UOW.
3. To create the XML Schema Document (XSD) files that can be used for UOW, perform the following steps:
   a. Trigger the appropriate event from the J.D. Edwards OneWorld GUI and generate an XML output file based on the event.
   b. Using the XML file, create an XSD file using an XML editor, such as XMLSPY.
   c. When creating the XSD, make sure that the XSD satisfies the SOA 10.1.3.1 namespace requirements. Manually add the namespace, target namespace, and other items that are required for SOA 10.1.3.1.

4. Once generated, copy the XSD files for the UOW into the repository folder. This repository folder is automatically configured on your file system when a J.D. Edwards OneWorld target is created using Application Explorer.

J.D. Edwards OneWorld Versions and Library Files

The following table lists the library files by version that are required for the OracleWLS Adapter for J.D. Edwards OneWorld.

J.D. Edwards OneWorld Version	Required Library Files
XE (B7333)	Connector.jar and Kernel.jar
ERP 8.0 (B7334)	Connector.jar and Kernel.jar
EnterpriseOne 8.9 (B9)	Connector.jar, Kernel.jar, jdeutil.jar, and log4j.jar
EnterpriseOne 8.10	Connector.jar, Kernel.jar, jdeutil.jar, and log4j.jar
EnterpriseOne 8.11 (SP1 and Tools Release 8.95)	Base_JAR.jar, Connector.jar, JdeNet_JAR.jar, log4j.jar, and System_JAR.jar
EnterpriseOne 8.12 (Tools Release 8.96 2.0)	Connector.jar, log4j.jar, Base_JAR.jar, EventProcesser_EJB.jar, EventProcesser_JAR.jar. JdeNet_JAR.jar, and System_JAR.jar

On the J.D. Edwards OneWorld system, these library files are located in the following folder:

```
\\system\classes
```

J.D. Edwards OneWorld Concepts

You can use OracleWLS Application Adapter for J.D. Edwards OneWorld to call a J.D. Edwards OneWorld Master Business Function, such as Address Book, Purchase Order, and Sales Order. You can also use the adapter as a part of an integration effort to connect OneWorld with non-OneWorld systems.

OracleWLS Application Adapter for J.D. Edwards OneWorld can receive an XML document, or it can run one or more J.D. Edwards Master Business Functions (MBF) by passing an XML document into OneWorld through the J.D. Edwards OneWorld ThinNet API.

Integration with J.D. Edwards OneWorld

J.D. Edwards OneWorld supports multiple methods and technologies to provide interoperability. The three supported entry points are:

- Flat files
- Database tables
- Master Business Function (MBF) interactive calls

You configure Oracle AS Adapter to send requests to J.D. Edwards OneWorld. The adapter processes requests for J.D. Edwards OneWorld Master Business Functions (MBF), embedded in XML documents, and forwards them to a back-end J.D. Edwards

OneWorld system. The resulting response information is then returned and processed for further routing.

OracleWLS Application Adapter for J.D. Edwards OneWorld can receive an XML request document from a client and call a specific function in the target Enterprise Information System (EIS). OracleWLS Application Adapter for J.D. Edwards OneWorld acts as a consumer of request messages and provides a response. An adapter performs the following functions:

- Receives requests from a legacy system, another EIS, or a non-EIS client.
- Transforms the XML request document into the EIS-specific format.

  The request document conforms to a request XML schema.
  The schema is based on metadata in the EIS.
- Calls the underlying function in the EIS and waits for its response.
- Transforms the response from the EIS-specific data format to an XML document.

  The response document conforms to a response XML schema that is generated by the adapter.
  The schema is based on metadata in the EIS.

You can configure a channel for the adapter to receive messages from J.D. Edwards OneWorld. The information the channel receives is used to build an XML record and is forwarded to any specified disposition for further processing.

Channels are consumers of EIS-specific messages and may or may not provide a response. A channel performs the following functions:

- Receives messages from an EIS client
- Transforms the EIS-specific message format into an XML format.

Propagating Internal Listeners out of J.D. Edwards OneWorld

Integrating a J.D. Edwards OneWorld listener with external systems is similar to the outbound process, except in reverse. The Data Export Control table maintains the determination of whether a transaction must be integrated with an external system. When a transaction must be integrated, the MBF handles logging of all additions, changes, and deletions to the unedited transaction table. After the transaction information writes to the table, a key for that record is sent from the MBF to the subsystem data queue.

The subsystem data queue triggers the processing of the new record by launching an outbound subsystem batch process that is generic and handles all inbound transactions. The J. D. Edwards outbound subsystem then accesses the Data Export Control table to determine the configured external subscriber to run.

J.D. Edwards OneWorld Interoperability Framework

J.D. Edwards OneWorld enables integration with systems through its interoperability framework. The adapter uses the OneWorld framework and leverages various integration access methods to provide the greatest amount of flexibility and functionality.

OracleWLS Application Adapter for J.D. Edwards OneWorld supports the following integration access methods:

- J.D. Edwards OneWorld ThinNet API
- J.D. Edwards OneWorld XML

- J.D. Edwards unedited transaction tables (Z tables)

Figure 1–1 illustrates the outbound processing framework.

The adapter uses the J.D. Edwards OneWorld ThinNet API to communicate with the OneWorld application. Using the ThinNet API, the adapter can run one or more MBF in a single Unit Of Work (UOW). When any of the MBF fail, the entire UOW fails, preventing partial updates. Validation of data, business rules, and communications to the underlying database are handled by the OneWorld application because the adapter runs the MBF.

Figure 1–1 J.D. Edwards OneWorld Outbound Processing

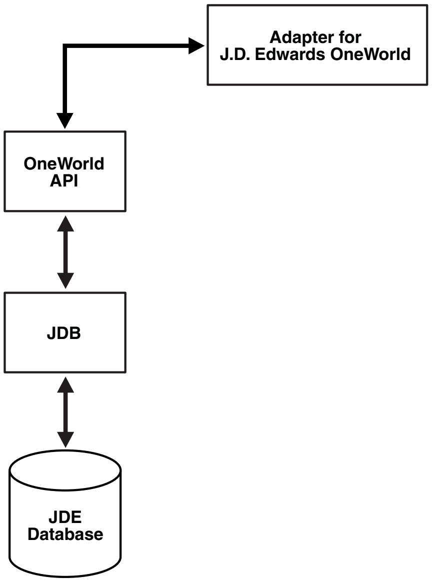
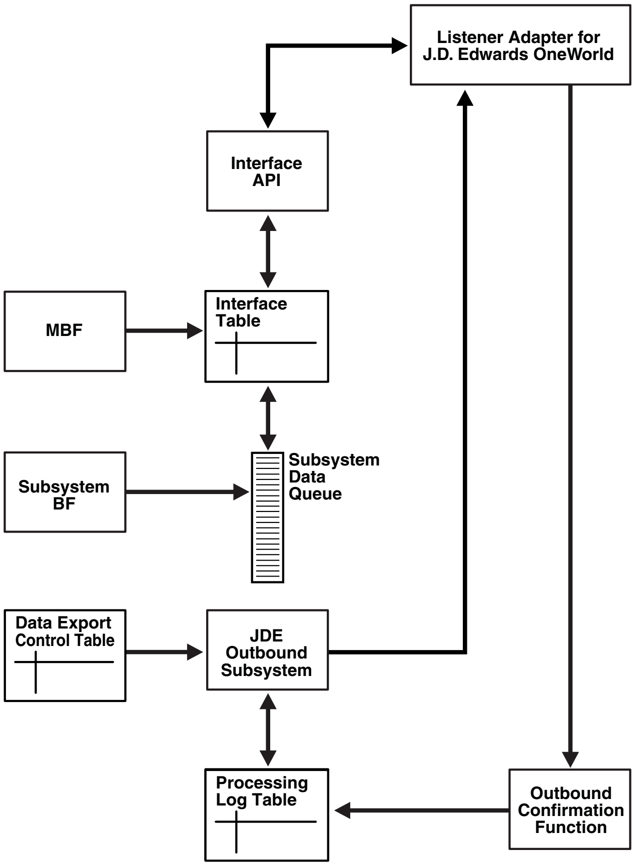


Figure 1–2 illustrates the inbound processing framework.

Figure 1–2 J.D. Edwards OneWorld Inbound Processing



In the outbound process, the event starts when a specific MBF is executed in the J.D. Edwards OneWorld environment. The MBF writes the required information for the event into the appropriate interface table and then notifies the subsystem Batch Function (BF) that an event occurred. The subsystem BF then places an entry about the event on the Subsystem Data Queue.

The J.D. Edwards outbound subsystem retrieves the data queue entry and looks in the Data Export Control table for the external processes to notify. The J.D. Edwards outbound subsystem then calls the OracleWLS Application Adapter for J.D. Edwards OneWorld listener with notification. The listener passes the notification to the generator. The generator then uses the J.D. Edwards OneWorld ThinNet API to retrieve the appropriate information from the interface table.

Adapter Architecture

OracleWLS Application Adapter for J.D. Edwards OneWorld uses Application Explorer with one of the following components:

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

OracleWLS Adapter Application Explorer (Application Explorer)

Application Explorer is used to configure database connections and create Web services and events. It can be configured to work in a Web services environment with BSE or with the Enterprise Connector for J2EE Connector Architecture (J2CA). When working in a J2CA environment, the connector uses the Common Client Interface (CCI) to provide fast integration services using Adapters instead of using Web services.

Both BSE and the connector for J2CA are deployed to an application server with Application Explorer and the adapters.

Application Explorer uses an explorer metaphor for browsing the J.D. Edwards OneWorld system for business functions. Application Explorer enables you to create XML schemas and Web services for the associated business function.

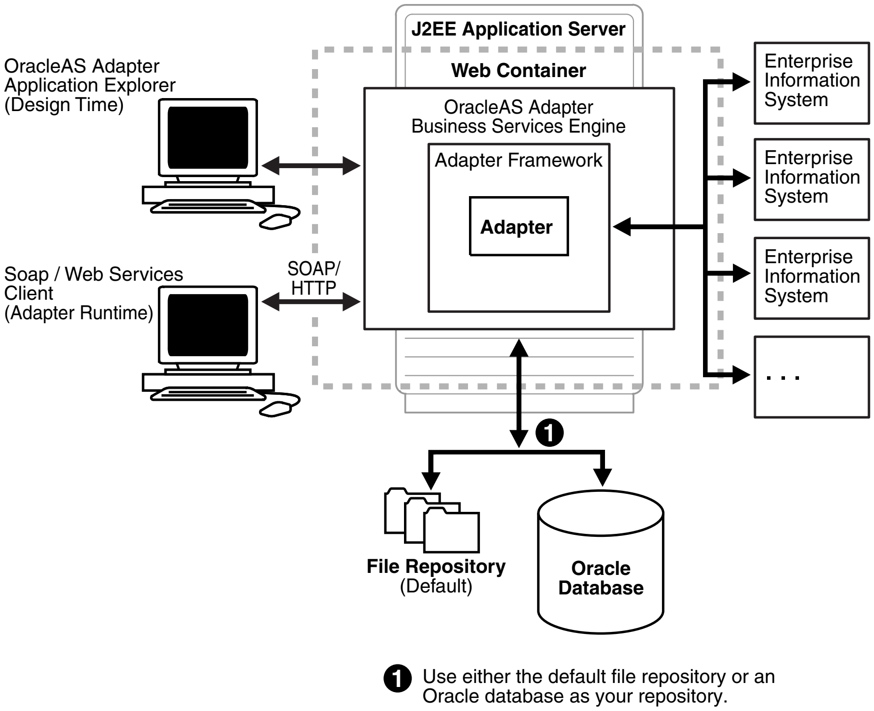
Resource Adapters

OracleWLS Application Adapter for J.D. Edwards OneWorld is a J2CA-based component also known as resource adapter. Resource adapters connect applications that were not originally designed to communicate with each other. Adapters are bidirectional, that is, they can send requests to an Enterprise Information System (EIS), and receive notification of events occurring in an EIS.

Oracle WebLogic Server Adapter Business Services Engine (BSE) Architecture

Figure 1–3 shows the generic architecture for the Oracle Web service adapter for packaged applications. The adapter works with BSE, as deployed to a Web container in a J2EE application server.

Figure 1–3 OracleWLS Adapter Business Services Engine (BSE) Architecture


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

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

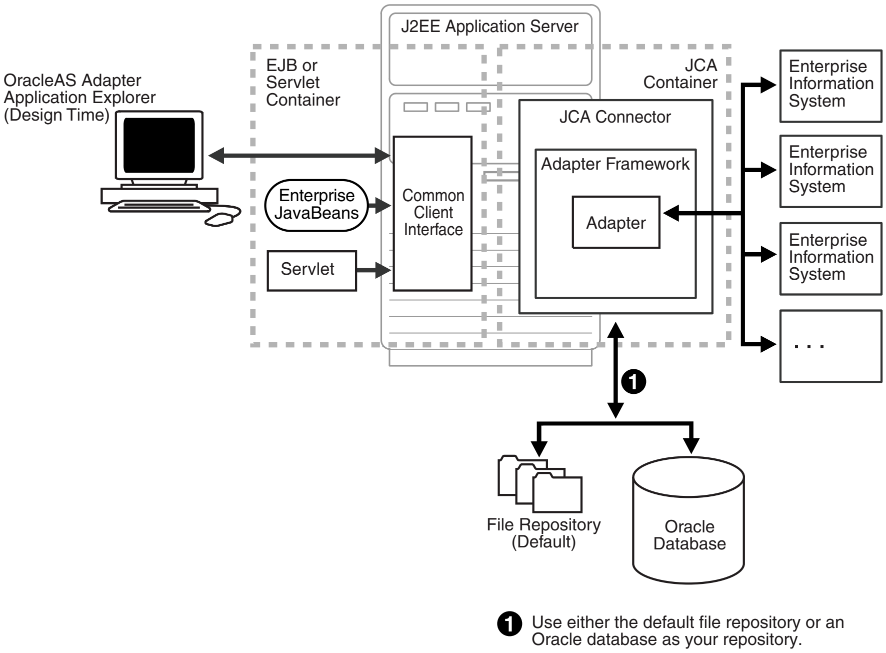
Application Explorer, a design-time tool deployed along with BSE, is used to configure adapter connections, browse EIS objects, configure services, and configure listeners to listen for EIS events. Metadata created while you perform these operations are stored in the repository by BSE.

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

Oracle WebLogic Server Adapter Generic J2CA Architecture

Figure 1–4 shows the generic architecture for Oracle J2CA adapter for packaged applications. The J2CA connector is deployed to a standard J2CA Container and serves as host container to the adapters. The connector is configured with a repository.

Figure 1–4 OracleWLS Adapter Generic J2CA Architecture


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

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

Processing Business Functions

OracleWLS Application Adapter for J.D. Edwards OneWorld enables the processing of OneWorld business functions through the J.D. Edwards OneWorld ThinNet API. Using the API eliminates the requirement of creating complex and impractical batch processes. In addition, a transport layer, such as IBM MQSeries, File, or HTTP is not required because a listener is defined through a TCP connection.

External applications that access OneWorld through OracleWLS Application Adapter for J.D. Edwards OneWorld use either XML schemas or Web services to pass data between the external application and the adapter. Chapter 2, "Configuring OracleWLS Application Adapter for J.D. Edwards One World" describes how to use Application Explorer to create XML schemas and Web services for the J.D. Edwards Master Business Functions (MBF) used with the adapter.

See Also:

- *Oracle WebLogic Server Adapter Concepts*
- *Oracle Application Adapter Installation Guide (WebLogic Server 10gr3)*

BSE Versus OracleWLS Adapter J2CA Deployment

If you are using OracleWLS Application Adapter for J.D. Edwards OneWorld with Oracle Service Bus (OSB), please note that:

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

The following three factors explain the differences between deploying BSE and OracleWLS 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. OracleWLS Adapter J2CA provides slightly better performance.

   OracleWLS Adapter J2CA does provide slightly better performance than BSE. However, the difference decreases as the transaction rate increases.
3. OracleWLS 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 OracleWLS Adapter J2CA, user name and password can be passed using the connection specification of the CCI.

2

Configuring OracleWLS Application Adapter for J.D. Edwards One World

This chapter describes how to use OracleWLS Adapter Application Explorer (Application Explorer) to define a target to connect to a J.D. Edwards OneWorld 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

To start Application Explorer:

1. Ensure the server is started where Application Explorer is deployed.
2. On Windows, execute the `ae.bat` file, which is found under *`wls_home`*`\erp-adapters\tools\iwae\bin`, where *`wls_home`* is the directory where Oracle WebLogic Server is installed.

   On UNIX, load the `iwae.sh` script file, which is found under *`wls_home`*`/erp-adapters/tools/iwae/bin`, where *`wls_home`* is the directory where Oracle WebLogic Server is installed.

Application Explorer starts. You can now define new targets to your Enterprise Information System (EIS).

Configuring Settings for BSE or J2CA

Before a repository configuration can be created, you must configure OracleWLS Adapter Business Services Engine (BSE). You need not configure the Connector for J2CA because the `ra.xml` file is configured automatically during installation.

Configuring BSE

After BSE is deployed to Oracle WebLogic Server, you can configure it through the BSE configuration page.

To configure BSE:

1. Display the following page in your browser:

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

   Where `host name` is the host name of Oracle WebLogic Server and`port` is the HTTP port for Oracle WebLogic Server.

   For example,

   ```
   http://localhost:7777/ibse
   ```

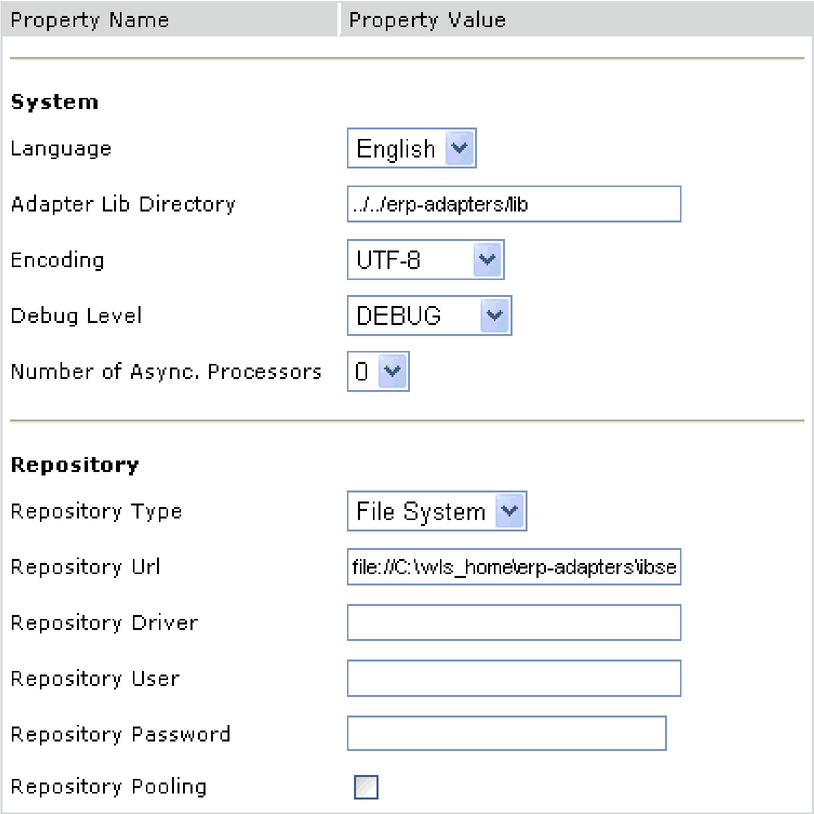
   > **Note:** The first time you access this page, it may take time to load.

2. Log on when prompted.

   When first installed, the user ID and the password are:

   - User name: weblogic
   - Password: weblogic

   The BSE configuration page is displayed.



Property Name	Property Value
System	
Language	English
Adapter Lib Directory	../../erp-adapters/lib
Encoding	UTF-8
Debug Level	DEBUG
Number of Async. Processors	0
Repository	
Repository Type	File System
Repository Url	file://C:\wls_home\erp-adapters\ibse
Repository Driver	
Repository User	
Repository Password	
Repository Pooling	☐

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

   ```
   wl_home\erp-adapters\lib
   ```

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

4. For security purposes, enter a new password in the **Admin Password** field.

   > **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. See "Configuring an Oracle Repository" on page 2-5 for information on switching to a database repository.

5. Click **Save**.

Configuring BSE System Settings

To configure BSE system settings:

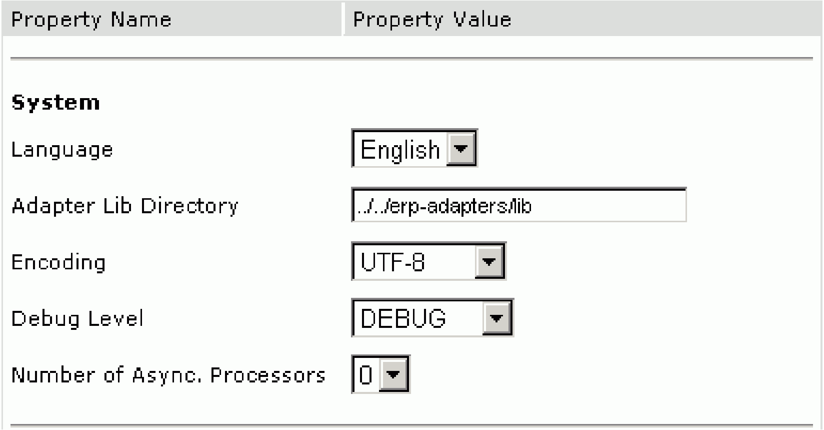
1. Display the BSE configuration page by entering the following URL in a browser:

   ```
   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 configuration page is displayed, as shown in the following figure.

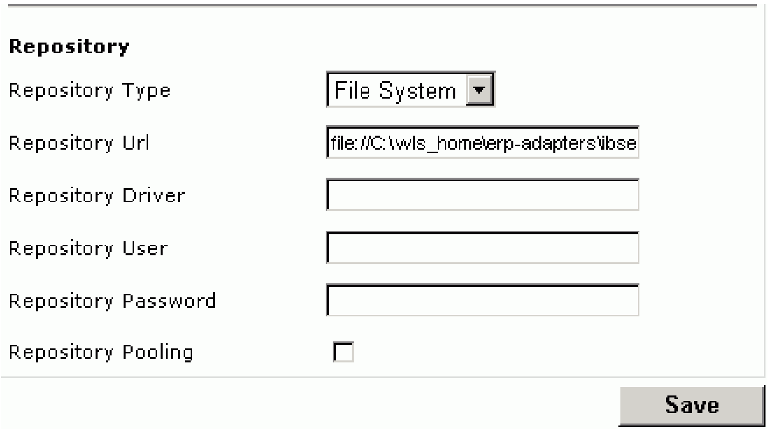


Property Name	Property Value
System	
Language	English
Adapter Lib Directory	../../erp-adapters/lib
Encoding	UTF-8
Debug Level	DEBUG
Number of Async. Processors	0

2. Configure the system settings according to the information in the following table.

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 one of the following options: ▪ None ▪ Fatal ▪ Error ▪ Warning ▪ Info ▪ Debug
Number of Async. Processors	Select the number of asynchronous processors.

The following image shows the Repository pane of the BSE configuration page.



Repository	
Repository Type	File System
Repository Url	file://C:\wls_home\erp-adapters\ibse
Repository Driver	
Repository User	
Repository Password	
Repository Pooling	□

BSE requires a repository to store transactions and metadata required for the delivery of Web services. For more information, see "Configuring a File System Repository" on page 2-5 and "Configuring an Oracle Repository" on page 2-5.

3. Configure the repository settings according to the information in the following table.

Parameter	Description
Repository Type	Select one of the following repositories from the list: ■ Oracle. ■ File. (Do not use a file repository for BSE in production environments.)
Repository URL	Enter the URL to use when opening a connection to the database. For example, the following repository URL format is used when connecting to Oracle: `jdbc:oracle:thin:@host name:port;SID`
Repository Driver	Provide the 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 will be 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:

```
wls_home\erp-adapters\ibse.war\ibserepo.xml
```

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.

Configuring an Oracle Repository

To configure an Oracle repository:

1. Contact your database administrator to obtain an Oracle user ID and password to create the BSE repository.

This user ID should have rights to create and modify tables, and the ability to create and execute stored procedures.

2. Open a command prompt and navigate to the setup directory. Its default location on Windows is:

   ```
   wls_home\erp-adapters\etc
   ```

   For other platforms, see the corresponding location.

   This directory contains SQL to create the repository tables in the following file:

   ```
   iwse.ora
   ```

   > **Note:** If the Oracle database is not on the same system as the Oracle WebLogic Server, copy the iwse.ora file to the system that has the Oracle database installed. Then, from a command prompt on the system, navigate to the directory containing the iwse.ora file.

3. Enter the following command:

   ```
   sqlplus userid/password @database @ iwse.ora
   ```

Configuring J2CA

During the J2CA deployment of OracleWLS Application Adapter for J.D. Edwards OneWorld, Oracle WebLogic Server generates a deployment descriptor called `ra.xml`. This descriptor provides OracleWLS-specific deployment information for resource adapters. See Chapter 3, "Oracle WebLogic Server Deployment and Integration" for more information.

No configuration changes are necessary if you are using the default file based repository with J2CA deployment.

Configuring an Oracle Repository

To configure an Oracle repository:

1. Contact your database administrator to obtain an Oracle user ID and password to create the J2CA repository.

   This user ID should have rights to create and modify tables, and the ability to create and execute stored procedures.

2. Open a command prompt and navigate to the setup directory. Its default location on Windows is:

   ```
   wls_home\erp-adapters\etc
   ```

   For other platforms, see the corresponding location.

   This directory contains SQL to create the repository tables in the following file:

   ```
   iwse.ora
   ```

   > **Note:** If the Oracle database is not on the same system as the Oracle WebLogic Server, copy the iwse.ora file to the system that has the Oracle database installed. Then, from a command prompt on the system, navigate to the directory containing the iwse.ora file.

3. Enter the following command:

```
sqlplus userid/password @database @ iwse.ora
```

Configuring a Database Repository for J2CA

To configure a database repository for J2CA:

1. Execute the `iwse.ora` SQL statement on the system where the database is installed.
2. Create the `jcatransport.properties` file and save it in the following directory:

   ```
   wls_home\erp-adapters\config\J2CA_SampleConfig
   ```

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

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

4. Open the `ra.xml` file in a text editor.
5. Provide the JDBC connection information as a value for the IWAYRepo_URL property.
6. Provide a valid user name for the IWAYRepo_User property.
7. Provide a valid password for the IWAYRepo_Password property.
8. Save your changes to the `ra.xml` file.

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 box opens.

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.

   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 (3189319731832973218312931643233332123227)
   ```

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

Creating a Repository Configuration

Before you use Application Explorer with OracleWLS Application Adapter for J.D. Edwards OneWorld, 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.

A default J2CA repository is created for the default ManagedConnectinFactory. The name of this configuration is `jca_sample`.

See "Adapter Features" on page 1-1 for more information.

Creating a Configuration for BSE

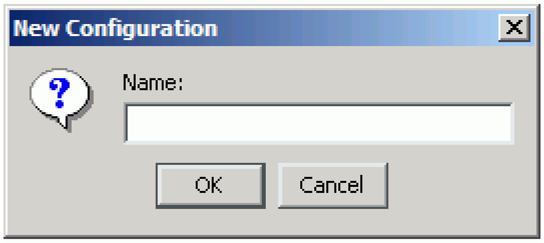
To create a repository configuration for BSE using Application Explorer, you must first define a new configuration.

Defining a New Configuration for BSE

To define a new configuration for BSE:

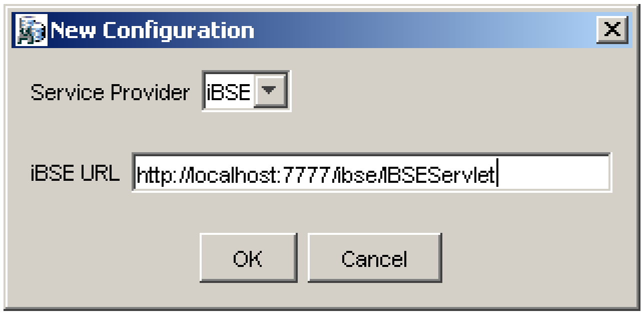
1. Right-click **Configurations** and select **New**.

   The New Configuration dialog box is displayed.



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

   The New Configuration dialog box is displayed.



Service Provider iBSE

iBSE URL http://localhost:7777/ibse/IBSEServlet

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

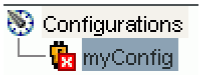
4. In the **iBSE URL** field, accept the default URL or replace it with a different URL using the following format:

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

   Where `host name` is the system where your application server resides and `port` is the HTTP port number where the application server is running.

5. Click **OK**.

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



Creating a Configuration for J2CA

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

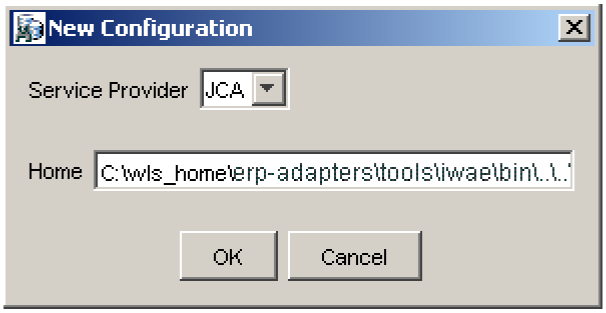
Defining a New Configuration for J2CA

To define a new configuration for J2CA:

1. Right-click **Configurations** and select **New**.

   The New Configuration dialog box is displayed.

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



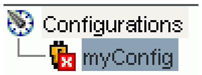
Service Provider: JCA

Home: C:\wls_home\erp-adapters\tools\iwae\bin\..\..\

3. From the **Service Provider** list, select **JCA**.

4. Click **OK**.

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



The OracleWLS Adapter J2CA configuration file is stored in *`wls_home`*`\erp-adapters\config\`*`configuration_name`*

Where *`configuration_name`* is the name of the configuration you created; for example, myConfig.

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, make sure that the iwjcaivp 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 box 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.

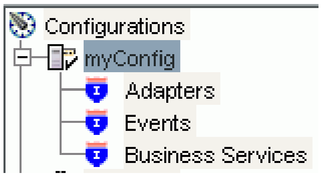
   Note: When you configure an Adapter target with the J2CA HTTP repository, you are not required to restart Oracle WebLogic Server for run time purposes.

Connecting to a BSE or J2CA Configuration

To connect to a BSE or J2CA configuration:

1. Right-click the configuration to which you want to connect, for example, myConfig.
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, you will not see the Business Services node. The following is an example of a BSE configuration named myConfig:



- Use the **Adapters** folder to create inbound interaction with J.D. Edwards OneWorld. For example, you use the J.D. Edwards One World node in the Adapters folder to configure a service that updates J.D. Edwards OneWorld.
- Use the **Events** folder to configure listeners that listen for events in J.D. Edwards OneWorld.
- 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 J.D. Edwards OneWorld.

Establishing a Connection (Target) for J.D. Edwards OneWorld

Part of the application definition includes adding a target for the adapter. Setting up the target in Application Explorer requires information which is specific to the target.

To browse the available Master Business Functions (MBF), you must first define a target to the system you use. After you define the target, it is automatically saved. You must connect to the 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.

Defining a Target to J.D. Edwards OneWorld

To connect to an application system for the first time, you must define a new target.

> **Note:** Before you create a new target, you must obtain the required library files for your J.D. Edwards OneWorld system and copy them to the appropriate location where the OracleWLS Application Adapter for J.D. Edwards OneWorld is deployed. For more information, see the *Oracle Application Adapter Installation Guide (WebLogic Server 10gr3)*.

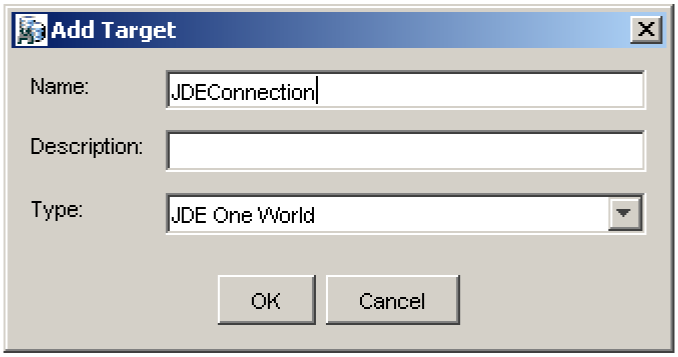
To define a target:

1. In the left pane, expand the **Adapters** node.

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

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

   The Add Target dialog box is displayed.



Name: JDEConnection

Description:

Type: JDE One World

   Perform the following steps:

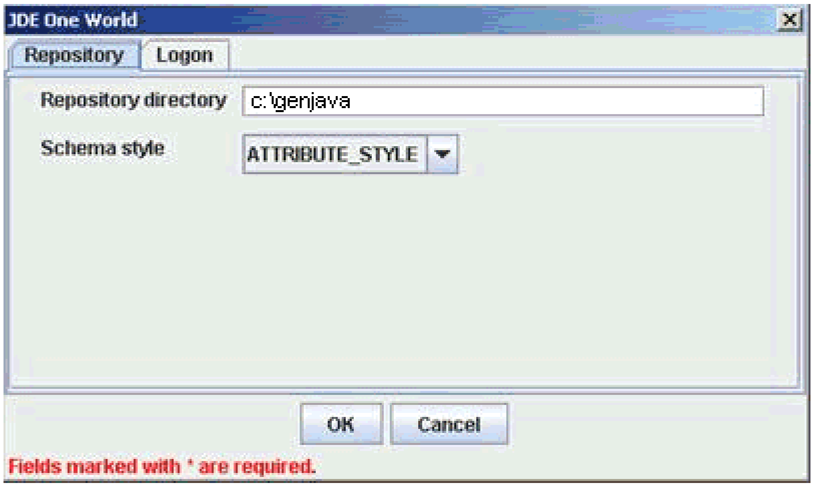
   a. In the **Name** field, enter a descriptive name, for example, JDEConnection.

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

   c. From the **Type** list, select **JDE One World**.

3. Click **OK**.

   The JDE One World dialog box appears.



Repository directory: c:\genjava

Schema style: ATTRIBUTE_STYLE

Fields marked with * are required.

a. In the **Repository** tab, enter the path to the GenJava repository in the Repository directory field.

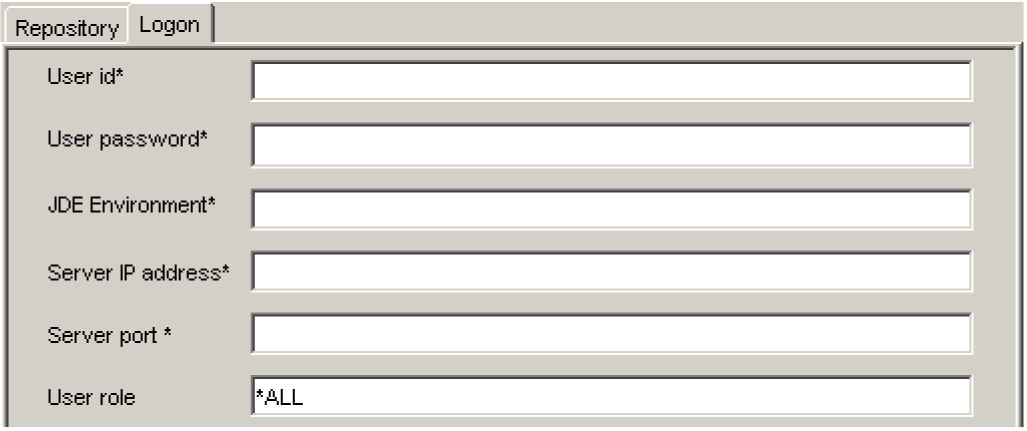
   This is the location of the Java wrappers for accessing the J.D. Edwards OneWorld business functions, which are created by the GenJava development tool. Please note that this is a prerequisite step, which must be performed before a new target is created using Application Explorer.

   Note: Generating schemas requires the GenJava repository. For more comprehensive information on building the J.D. Edwards OneWorld Master Business Function repository, see the *J.D. Edwards Interoperability Guide for OneWorld Xe.* For information on how to use the GenJava program, see Using the GenJava Development Tool (Outbound Processing) in Appendix A, "Configuring J.D. Edwards OneWorld for Outbound and Inbound Processing".

   ---

b. From the **Schema style** list, select **ELEMENT_STYLE** or **ATTRIBUTE_STYLE**.

c. Click the **Logon** tab and enter the appropriate information for your target type based on the information in the following table. Fields marked with an asterisk are required.



User id*

User password*

JDE Environment*

Server IP address*

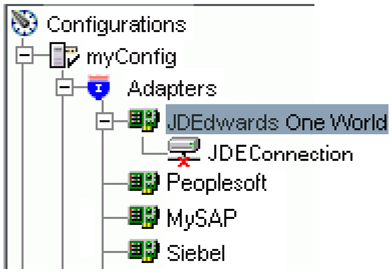
Server port *

User role: *ALL

Parameter	Description
User id*	A valid user ID for J.D. Edwards OneWorld.
User password*	The password associated with the user ID.
JDE environment*	The J.D. Edwards OneWorld environment, for example, DU7333. For more information about this parameter, see your J.D. Edwards OneWorld documentation or ask your OneWorld system administrator.
Server IP address*	The name of the server on which J.D. Edwards OneWorld is running. This can be the name of the server, for example, JDEOW, or its IP address.
Server Port*	The port number on which the server is listening, for example, 6009.
User role	This property setting depends on your J.D. Edwards system version. ▪ For **B7333**: You can leave this field blank. ▪ For **EnterpriseOne 8.10**: You must specify ***ALL**.

4. Click **OK**.

   The new target, JDEConnection, appears under the JDEdwards node.



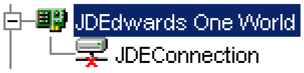
   For information on how to create schemas for the adapter, see "Creating an XML Schema" on page 2-14.

Connecting to a Defined J.D. Edwards OneWorld Target

To connect to a target:

1. Expand the **Service Adapters** node.
2. Expand the **JDEdwards One World** node.

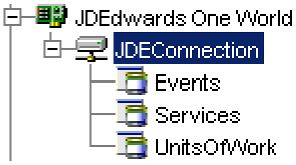
   The disconnected target is displayed.



3. Click the target name (for example, JDEConnection) under the JDEdwards node.
4. Click the **Logon** tab on the right.

   The Logon tab displays the values you entered for connection parameters.
5. Verify your connection parameters.
6. Provide the correct password.
7. Right-click the target name and select **Connect**.

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

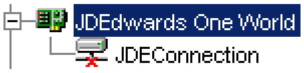


Disconnecting from J.D. Edwards OneWorld

To disconnect from a target:

1. Expand the **Adapters** node.
2. Expand the **JDEdwards** node.
3. Right-click the target to which you are connected (for example, JDEConnection), and select **Disconnect**.

   The x icon appears, indicating that the node is disconnected.



Editing a Target

To edit a target:

1. In the left pane, ensure that the target you want to edit is disconnected.
2. Right-click the target and select **Edit**.

   A window is displayed that enables you to edit the existing connection parameters.
3. Modify the target information.
4. Click **OK**.

Deleting a Target

You can delete a target, rather than just disconnecting and closing it. When you delete the target, the node disappears from the list of J.D. Edwards OneWorld targets in the left pane of the explorer.

To delete a target:

1. Expand the **Adapters** node.
2. Expand the **JDEdwards One World** node.
3. Right-click the target to which you are connected (for example, JDEConnection), and select **Delete**.

   The node disappears from the list of available connections.

For information on how to view application system objects, see *J.D .Edwards Interoperability Guide Release OneWorld XE*.

Creating an XML Schema

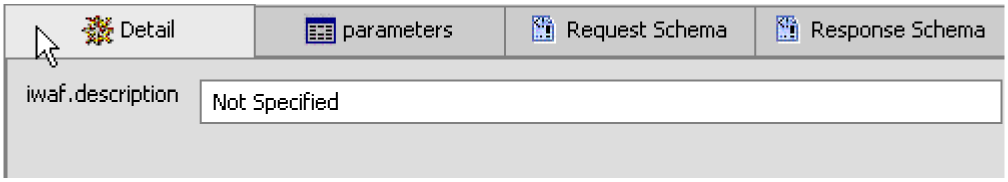
To execute an MBF, the adapter must receive a request document through the J.D. Edwards OneWorld ThinNet API. The agent processes the request and sends an XML response document indicating the result. Application Explorer creates both the XML request schema and the XML response schema.

Creating a Request and a Response Schema

The following procedure explains how to create request and response schemas for a J.D. Edwards OneWorld business function. Application Explorer enables you to create XML schemas for this function.

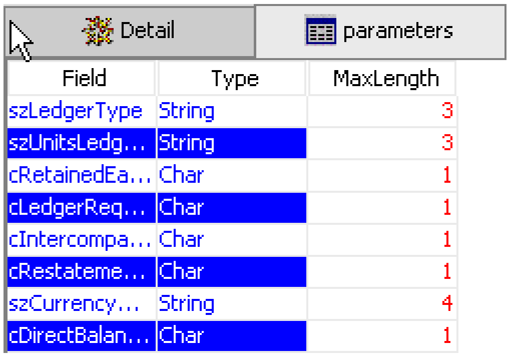
1. Connect to a J.D. Edwards OneWorld target as described in "Connecting to a Defined J.D. Edwards OneWorld Target" on page 2-13.
2. Expand the **Services** node.
3. Expand the node of the MBF for which you want to create the schema.
4. Expand and then select the node beneath the MBF.

   The following image shows the tabs that appear on the right.



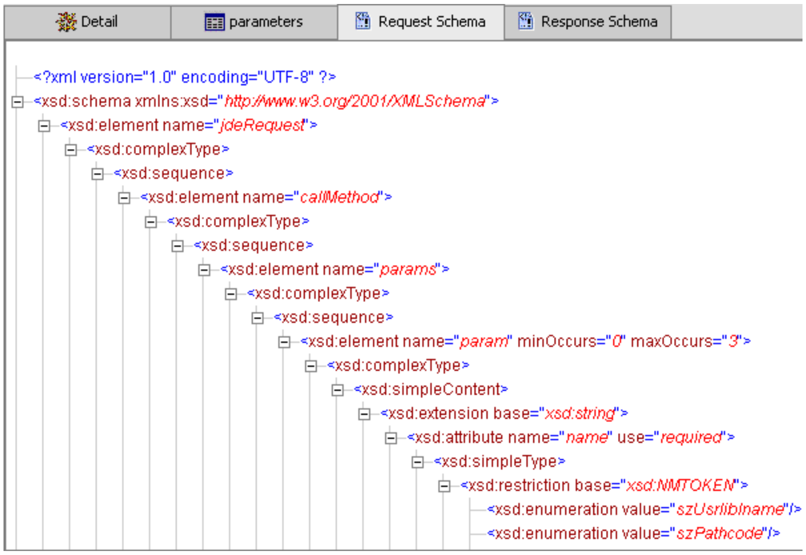
iwaf.description: Not Specified

5. Click the **parameters** tab to view the parameter information.



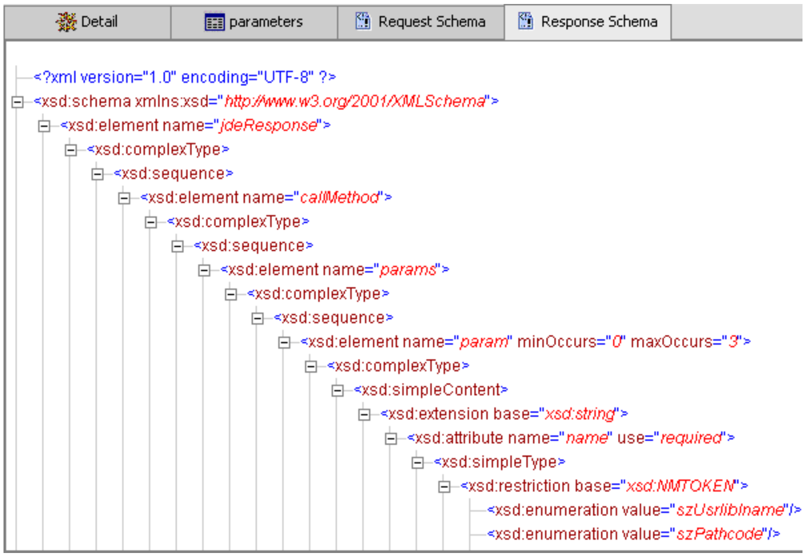
Field	Type	MaxLength
szLedgerType	String	3
szUnitsLedg...	String	3
cRetainedEa...	Char	1
cLedgerReq...	Char	1
cIntercompa...	Char	1
cRestateme...	Char	1
szCurrency...	String	4
cDirectBalan...	Char	1

6. Click **Request Schema** to view the request schema information.



```
<?xml version="1.0" encoding="UTF-8" ?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="jdeRequest">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="callMethod">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="params">
                <xsd:complexType>
                  <xsd:sequence>
                    <xsd:element name="param" minOccurs="0" maxOccurs="3">
                      <xsd:complexType>
                        <xsd:simpleContent>
                          <xsd:extension base="xsd:string">
                            <xsd:attribute name="name" use="required">
                              <xsd:simpleType>
                                <xsd:restriction base="xsd:NMTOKEN">
                                  <xsd:enumeration value="szUsrlibIname"/>
                                  <xsd:enumeration value="szPathcode"/>
```

7. Click **Response Schema** to view the response schema information.



```
<?xml version="1.0" encoding="UTF-8" ?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="jdeResponse">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="callMethod">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="params">
                <xsd:complexType>
                  <xsd:sequence>
                    <xsd:element name="param" minOccurs="0" maxOccurs="3">
                      <xsd:complexType>
                        <xsd:simpleContent>
                          <xsd:extension base="xsd:string">
                            <xsd:attribute name="name" use="required">
                              <xsd:simpleType>
                                <xsd:restriction base="xsd:NMTOKEN">
                                  <xsd:enumeration value="szUsrlibIname"/>
                                  <xsd:enumeration value="szPathcode"/>
```

Generating WSDL (J2CA Configurations Only)

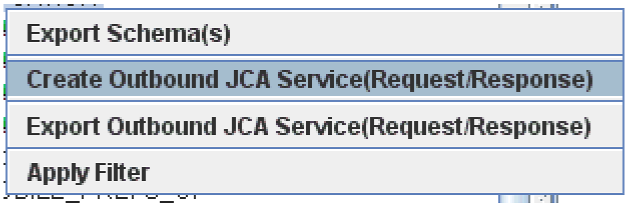
The procedure for generating WSDL (Web Service Definition Language) for request-response (outbound) services differs from that of generating WSDL for event notification (inbound) J2CA services of the adapter. The following sections describe each procedure.

Generating WSDL for Outbound Interaction

To generate a WSDL file for request-response service:

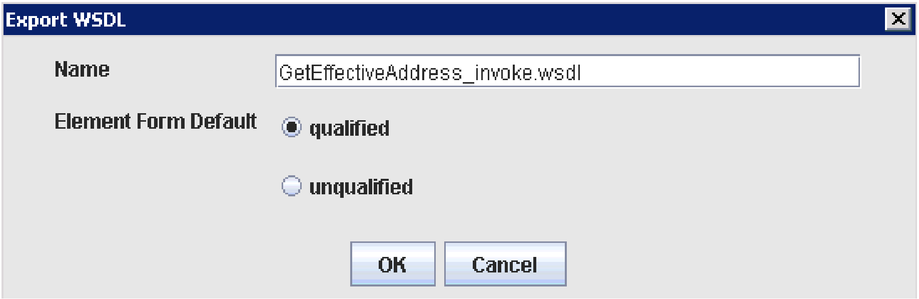
1. Start Application Explorer and connect to a defined J.D. Edwards target.
2. Expand **Services**, **JDEJAVA_CFIN**, and then **B0100033**. Select **GetEffectiveAddress**.
3. Right-click GetEffectiveAddress.

   The following menu is displayed.


Export Schema(s)
Create Outbound JCA Service(Request/Response)
Export Outbound JCA Service(Request/Response)
Apply Filter

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

   The Export WSDL dialog box is displayed.



Export WSDL

Name: GetEffectiveAddress_invoke.wsdl

Element Form Default: ● qualified ○ unqualified

5. Accept the default name for the file and make sure **qualified** is selected.

   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`.

6. Click **OK**.

   The WSDL file is saved in the specified location.

Generating WSDL for Inbound Interaction

You cannot generate a WSDL for J.D. Edwards OneWorld event notification using Application Explorer. To generate a WSDL from the command prompt, you must perform the steps that are described in the following section.

You can create an inbound J2CA service only if the node that you selected supports events.

1. Create a channel using Application Explorer under the J.D. Edwards Events node.

2. Start the channel.

   Do not restart Oracle WebLogic Server after the channel is started.

3. Send an inbound message from J.D. Edwards OneWorld.

4. Capture the inbound message payload in the log file, which is located in the following directory:

   ```
   wls_home\erp-adapters\application\config\jca_sample\log\iwaf_jca15.log
   ```

   Alternatively, you can create a port using the File protocol under the Events node in Application Explorer, which will dispose the event message to the file system.

5. Use a third party tool (for example, XMLSpy) to create the XML schema (.xsd file) using the XML payload that was captured in the previous step.

6. Verify that **orabpel-adapters.jar** is in your class path

7. Open a command prompt and navigate to the following directory:

   ```
   wls_home\adapters\application\tools
   ```

8. Execute the **obadapter.bat** file to set the environment.

9. Enter the following command to generate a WSDL:

   ```
   java com.iwaysoftware.af.container.tools.wsdl.IWayWSILBrowser jcaHome adapter
   target channel schemaPrefix wsdlFileName
   ```

   where:

   `jcaHome` is the path to your J2CA configuration. For example:

```
wls_home\erp-adapters\application\
```

adapter is the name of the adapter. For example, JDEdwards.

target is the name of the adapter target you created in Application Explorer.

channel is the name of the channel you created in Application Explorer.

schemaPrefix is the prefix for the XSD schema. The schema file must be in the same directory where the Java command is executed, for example:

```
java com.iwaysoftware.af.container.tools.wsdl.IWayWSILBrowser
c:\bea\erp-adapters\ JDEdwards jde812_tgt jde_ch Jde812_Schema Jde812_
salesorder_receive_osb.wsdl
```

Once the command is executed, the following is displayed in the command window:

```
Running Inbound WSDL generation tool...
-> user.dir = java com.iwaysoftware.af.container.tools.wsdl.IWayWSILBrowser
c:\bea\erp-adapters\ JDEdwards jde812_tgt jde_ch Jde812_Schema Jde812_
salesorder_receive_osb.wsdl

-> Generating WSDL...
-> Done.
-> Writing WSDL 'c:\bea\erp-adapters\application\\wsdls\ Jde812_salesorder_
receive_osb.wsdl ' to disk...
-> Done.
```

> **Note:** It is good practice to append **_receive** to the names of WSDL files that are generated for event notification services. This will allow you to easily distinguish between them and those generated for request-response services.

10. Stop the channel in Application Explorer.

Publishing a WSDL

After you browse the J.D. Edwards OneWorld business object repository, you can publish the specific WSDL document for use with Oracle Service Bus. The following section describes how to publish a WSDL using Application Explorer.

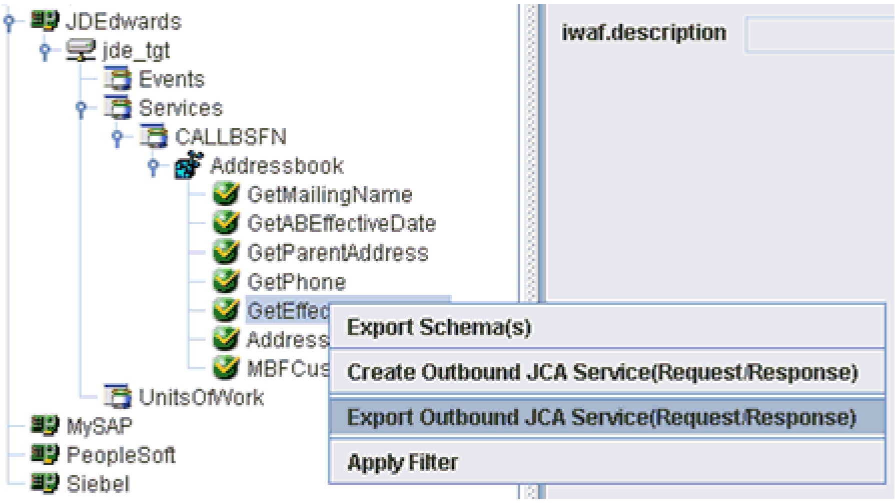
> **Note:** Only users with Group Membership types set to *Administrators* in Oracle Service Bus can publish WSDL files using Application Explorer. For example, a user with the Group Membership type set to *IntegrationDeployers* cannot publish a WSDL file.

Publishing a WSDL

To publish a WSDL:

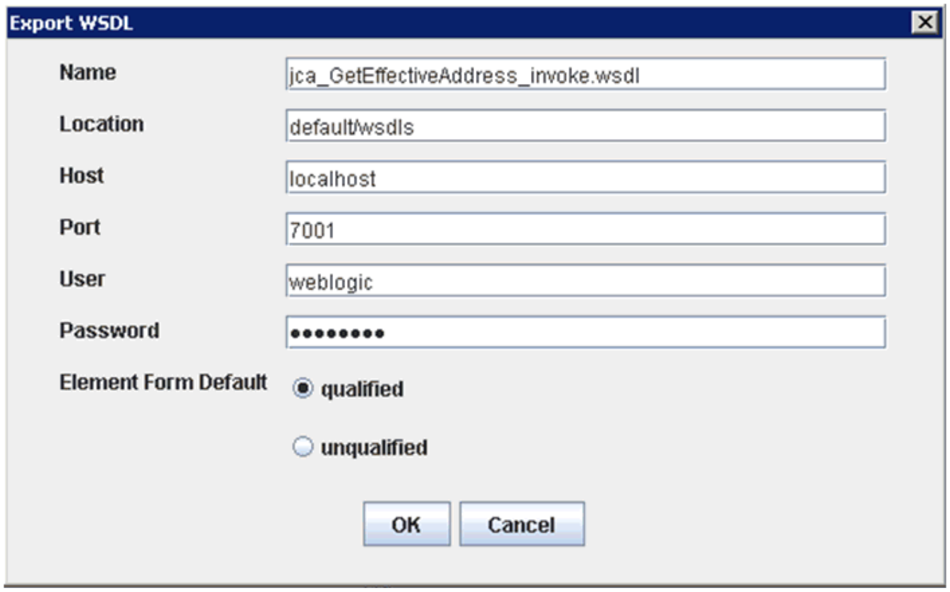
1. After you create a schema, right-click the respective object.

   The following menu is displayed.



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

   The Export WSDL dialog box is displayed.


Export WSDL

Name	jca_GetEffectiveAddress_invoke.wsdl
Location	default/wsdls
Host	localhost
Port	7001
User	weblogic
Password	••••••••
Element Form Default	● qualified ○ unqualified

3. In the Name field, a default file name for the WSDL file is provided. You can accept the default or provide your own.

4. In the Location field, enter the location where you want to publish the WSDL document.

   The location is composed of an Oracle Service Bus project name and optionally, one or more folder names. The project name and any folder names must be separated by a back slash character "\".

5. In the Host field, enter the name of the machine where Oracle WebLogic Server is running.

6. In the Port field, enter the port for the domain you are using. The port for the default domain is 7001.

7. In the User field, enter your username to access Oracle Service Bus.

8. In the Password field, enter your password to access Oracle Service Bus.

9. Ensure that **qualified** is selected as the element form, which is the default.

10. Click **OK**.

    The WSDL is published to the location specified in the Export WSDL dialog box and is now available for use with a Business Service or Proxy Service in Oracle Service Bus.

Creating and Testing a Web Service (BSE Configurations Only)

You can generate a Web service (also known as a **business service**) using Application Explorer. You can explore the business function repository and generate Web services for the functions you want to use with the adapter. The following procedure uses an example called BusinessUnitExistenceCheck.

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

Creating a Web Service

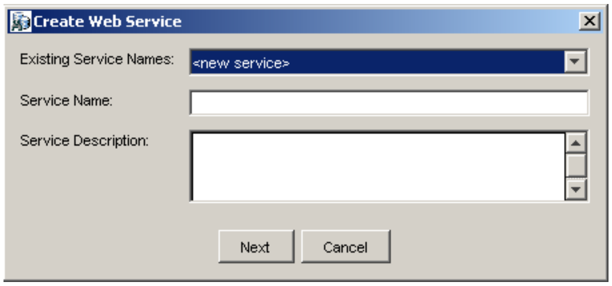
To create a Web service for a business function:

1. Expand the **JDEdwards** node and then the **Services** node.

2. Expand the MBF, **B1000012**, also called BusinessUnitExistenceCheck.

3. Right-click the node from which you want to create a business service and select **Create Web Service**.

   The Create Web Service dialog box is displayed.



Create Web Service

Existing Service Names: <new service>

Service Name:

Service Description:

   You can add the business function as a method for a new Web service or as a method for an existing one.

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

   b. In the **Service Name** field, specify a service name if you are creating a new service. This name identifies the Web service in the list of services under the Business Services node.

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

4. Click **Next**.

   Perform the following steps:

   a. In the **License Name** field, select one or more license codes to assign to the Web service.

   b. In the **Method Name** field, enter a descriptive name for the method or accept the default name.

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

   d. In the **DTD Directory** field, specify a location where the Web service will be saved. If you want to select a location different than the default, click **Browse** and navigate to the desired location.

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.

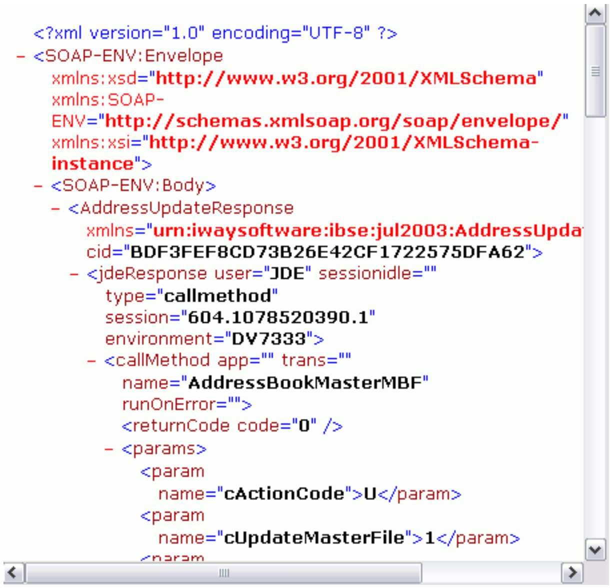
   The business service name appears as a link in the right pane.

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

   The test option appears in the right pane. If you are testing a Web service that requires XML input, an input field appears.

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

   Application Explorer displays the results.



```
  <?xml version="1.0" encoding="UTF-8" ?>
- <SOAP-ENV:Envelope
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:SOAP-
    ENV="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-
    instance">
  - <SOAP-ENV:Body>
    - <AddressUpdateResponse
        xmlns="urn:iwaysoftware:ibse:jul2003:AddressUpda
        cid="BDF3FEF8CD73B26E42CF1722575DFA62">
      - <jdeResponse user="JDE" sessionidle=""
          type="callmethod"
          session="604.1078520390.1"
          environment="DV7333">
        - <callMethod app="" trans=""
            name="AddressBookMasterMBF"
            runOnError="">
            <returnCode code="0" />
          - <params>
              <param
                name="cActionCode">U</param>
              <param
                name="cUpdateMasterFile">1</param>
              <param
```

Identity Propagation

If you test or execute a Web service using a third party XML editor, for example XMLSPY, the Username and Password values that you specify in the SOAP header must be valid and are used to connect to J.D. Edwards OneWorld. The user name and password values that you provided for J.D. Edwards OneWorld 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.

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 event object and select this channel when you generate WSDL. Creating a channel is required for J2CA configurations only.

A **channel** represents configured connections to particular instances of back-end systems. See "Creating and Editing a Channel" on page 2-22 for more information.

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.

> **Note:** If you are using a J2CA configuration, you must create a new channel for every event and select this channel when you generate WSDL. Creating a channel is required for both BSE and J2CA configurations.

Creating a Channel

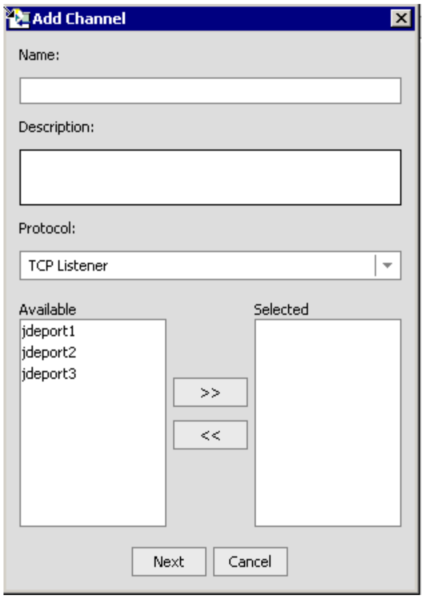
To create a channel:

1. Click **Event Adapters**.
2. Expand the **JDEdwards One World** node.

   The ports and channels nodes appear in the left pane.

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

   The Add Channel dialog box is displayed.



   Perform the following steps:

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

   b. Enter a brief description.

   c. From the disposition **Protocol** list, select **TCP Listener**.

4. Click **Next**.

   The TCP Listener dialog box is displayed with the Basic tab active. Perform the following steps:

   a. Enter the parameters that are specific to your J.D. Edwards environment.

   b. Click the **preparser** tab.

   c. Enter the required parameters.

   The following table lists the parameters with their descriptions. Parameters with an asterisk are required.

Parameter	Description
Port Number*	Port on which the Host database is listening.
Host/IP Binding	Name or URL of the system where the database resides.

Synchronization Type	Possible values are:
	`REQUEST_RESPONSE`
	`REQUEST_ACK`
	`REQUEST`

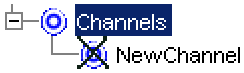
Is Length Prefix	For J.D. Edwards OneWorld events that send data back that is not in XML format. The TCP/IP event application must prefix the data with a 4-byte binary length field when writing the data to the TCP/IP port.
Is XML	For J.D. Edwards OneWorld events that send data back in XML format. No preparser is required.

Parameter	Description
Is Keep Alive	Maintains continuous communication between the event transaction and the channel.
User id*	A valid user ID for J.D. Edwards OneWorld.
User password*	The password associated with the user ID.
JDE Environment*	The J.D. Edwards OneWorld environment, for example, DU7333. For more information about this parameter, see your J.D. Edwards OneWorld documentation or ask your OneWorld system administrator.
Application	XMLInterop or the application name in J.D. Edwards OneWorld. Optional.
Server IP address*	Name of the server on which J.D. Edwards OneWorld is running. This can be the name of the server, for example, JDEOW, or its IP address, for example, 123.45.67.89.
Server port*	Port number on which the server is listening, for example, 6009.

For additional parameters, see your J.D. Edwards OneWorld Administrator.

5. Click **OK**.

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

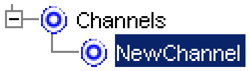


An X over the icon indicates that the channel is currently disconnected. You must start the channel to activate your event configuration.

> **Note:** If you are using OracleWLS Application Adapter for J.D. Edwards OneWorld with Oracle Service Bus (OSB) Proxy Services, do not start the channel as it is managed by OSB. If you start the channel from Application Explorer for testing and debugging purposes, stop it before run-time (when working with OSB components).

6. Right-click the channel node and select **Start**.

   The channel becomes active.



The X over the icon disappears.

7. 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 channels pane is displayed.

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

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**.

   A confirmation dialog box is displayed.
3. To delete the channel you selected, click **OK**.

   The channel disappears from the list in the left pane.

The OneWorld Event Listener

OracleWLS Application Adapter for J.D. Edwards OneWorld Event Listener is designed specifically to provide J.D. Edwards approved access to your OneWorld business events. The OneWorld Event Listener refers to a specialized application that runs with OneWorld business functions and is called by the OneWorld application system.

The OneWorld application system provides the Event Listener with the information required to retrieve the event information for only the desired events. For information about configuring the OneWorld environment, see the *J.D. Edwards Interoperability Guide for OneWorld.*

The OneWorld Event Listener is called directly from the OneWorld application and is passed a Z-file record identifier. This identifier then generates a request document that is passed to the server for processing. The server retrieves the event information from the J.D. Edwards OneWorld system and propagates the information for integration with other application systems.

Configuring the OneWorld Event Listener

The OneWorld Event Listener is installed as part of the basic installation. The OneWorld Adapter is automatically installed in the appropriate directory. If the integration server is not installed on the same computer as the J.D. Edwards application server, you must configure the OneWorld Event Listener.

The OneWorld Event Listener is invoked by J.D. Edwards for specific transactions as configured in the OneWorld environment.

The OneWorld Event Listener includes the following components:

- The listener exit `(IWOEvent),` located under *`adapters_home`*`\etc\jde`, where `adapters_home` is *`wls_home`*`\erp-adapters\etc\application\`. For example:

  ```
  C:\wls_home\erp-adapters\etc\application\jde\Iwoevent.dll
  ```

  The file extension varies depending on your operating system:

  - For **Windows**, the exit is `Iwoevent.dll`.
  - For **Sun Solaris**, the exit is `libiwoevent.so`.
  - For **HP-UX**, the exit is `libiwoevent.sl`.
  - For **AS/400**, the exit is `iwaysav.sav`.
  - For **IBM AIX**, the exit is `libiwoevent.so`.

- The listener configuration file (iwoevent.cfg), which must be created by the user.

The OneWorld Event listener exit is the function that passes the key fields for a record in the OneWorld outbound transaction tables to the integration server for processing by the inbound OracleWLS Application Adapter for J.D. Edwards OneWorld. The OneWorld Event listener is deployed under the J.D. Edwards OneWorld Enterprise Server. The Java class for the OneWorld Event listener is called IWOEvent (the file extension depends on the operating system) and is case-sensitive.

1. Create a folder called Outbound under the JDE structure on the JDE Enterprise Sever, for example:

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

2. Copy the iwoevent.dll file in the new Outbound folder.
3. Create an environment variable, *IWOEVENT_HOME*, to point to the directory containing the iwoevent.dll file.
   - On Windows: Add *IWOEVENT_HOME* to the system environment variables.
   - On UNIX: Add the following command to your start-up script:

     ```
     export IWOEVENT_HOME =/directory_name
     ```

4. On the J.D. Edwards OneWorld Server, create an iwoevent.cfg file in the defined directory, *IWOEVENT_HOME*.

   The OneWorld Event listener requires connection information for the associated adapter to initiate events properly. This information is contained in the iwoevent.cfg file. You must create this file and add the connection information to it. The OneWorld Event Listener requires connection information for the associated integration server to function properly. This information is contained in the iwoevent.cfg file. The iwoevent.cfg file has three distinct sections:

   - **Common**

     The common section of the configuration file contains basic configuration options. Currently, only the trace option is supported.

     To set the trace option, select **on** or **off**.

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

     Where on sets the tracing to on and off sets the tracing to off. Off is the default value.

   - **Alias**

     The alias section of the configuration file contains the connection information required to send transactions to specific servers. Currently, the OracleWLS Application Adapter for J.D. Edwards OneWorld supports 100 entries (alias names) in the configuration file.

     The alias values to these entries are as follows:

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

     Where:

     aliasname is the symbolic name given to the connection.

     ipaddress|dsn is the IP address or DSN name for the server containing OracleWLS Application Adapter for J.D. Edwards OneWorld (required).

`port` is the port defined for OracleWLS Application Adapter for J.D. Edwards OneWorld in the TCP channel configuration (required).

`trace={on|off}` sets the tracing to on for the particular alias.

- **Trans**

  The trans section of the configuration file contains transaction information required to route J.D. Edwards OneWorld transactions to specified servers.

  If a particular J.D. Edwards OneWorld transaction is not defined to an alias, it is sent to all aliases. The trans values to these entries are as follows:

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

  Where `jdeTransactionName` is the JDE-defined name for the outbound transaction and `alias1,alias2,aliasn` is the list of aliases to which the transactions are sent.

The following is a sample entry for `iwoevent.cfg` that supplies connection information:

```
common.trace=on

alias.edamcs1=172.1.1.1:3694
alias.edamcs1t=172.1.1.1:3694, trace=on
alias.edamcs2=222.2.2.2:1234

trans.JDESOOUT=edamcs1t,edamcs2
trans.JDEPOOUT=edamcs1
```

5. Create a folder using the alias names that are specified in the iwoevent.cfg file under the defined directory, *IWOEVENT_HOME*. For example:

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

Runtime Overview

After OneWorld starts the OneWorld Event listener, the listener accesses the configuration file, called `iwoevent.cfg` (case-sensitive). Based on the information in the configuration file, the listener sends the event notification to the integration server. All log information is saved in a file called `iwoevent.log`. The `iwoevent.log` file is created in the outbound folder where the `iwoevent.dll` and `iwoevent.cfg` files are located.

Modifying the JDE.INI File for Outbound and Inbound Processing

This section describes the settings that are required in the JDE.INI file for the XML call object kernel (outbound and inbound processing).

Open the JDE.INI file and modify the **[JDENET_KERNEL_DEF6]** and **[JDENET_KERNEL_DEF15]** sections as follows:

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

[JDENET_KERNEL_DEF15]
```

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

The parameters containing an underscore (_) and @28 are for Windows NT operating systems only. For other operating systems, replace the parameters with the values in the following table:

Operating System	Call Object dispatch DLLName	XML Trans dispatch DLLName
AS400	XMLCALLOBJ	XMLTRANS
HP9000B	libxmlcallojb.sl	libxmltransactions.lo
Sun or RS6000	libxmlcallojb.so	Libxmltransactions.so

Note: The J.D. Edwards OneWorld installation for version B7333(XE) does not include **[JDENET_KERNEL_DEF15]**. As a result, if you are using version B7333(XE), you must manually add it to the jde.ini file. For all other J.D. Edwards OneWorld versions, **[JDENET_KERNEL_DEF15]** is included with the installation.

3

Oracle WebLogic Server Deployment and Integration

This chapter describes Oracle WebLogic Server (OracleWLS) deployment and integration with OracleWLS Application Adapter for J.D. Edwards OneWorld.

This chapter discusses the following topics:

- Adapter Integration with OracleWLS
- Deployment of Adapter
- Updating Adapter Configuration
- How to Write a Java Application Client Using the CCI API

See Also:

- *Oracle WebLogic Server Adapter Concepts*

Adapter Integration with OracleWLS

OracleWLS Application Adapter for J.D. Edwards OneWorld 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.

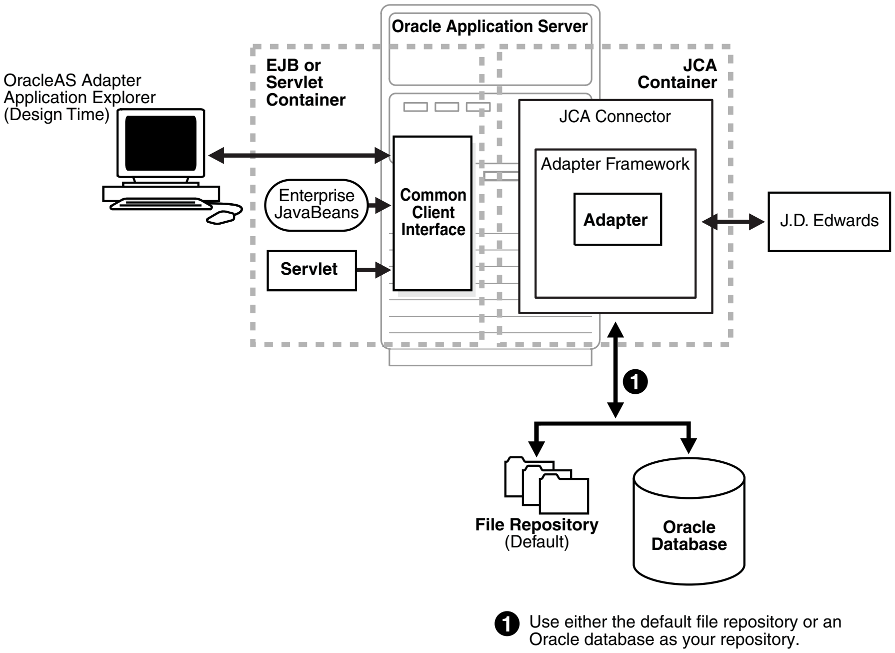
See Also:

- Oracle WebLogic Server Adapters Integration with OracleWLS in *Oracle WebLogic Server Adapter Concepts*

Deployment of Adapter

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

Figure 3–1 Oracle WebLogic Server Adapter J2CA Architecture


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

See Also:

- *Oracle WebLogic Server Adapter Concepts*

Updating Adapter Configuration

During the J2CA deployment of OracleWLS Application Adapter for J.D. Edwards OneWorld, OracleWLS generates a deployment descriptor called `ra.xml`, located in `wls_home\erp-adapters\iwafjca.rar\META-INF`.

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

Creating a Managed Connector Factory Object

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

```
<?xml version="1.0"?>
<!DOCTYPE oc4j-connector-factories PUBLIC "-//Oracle//DTD Oracle Connector
9.04//EN"
"http://xmlns.oracle.com/ias/dtds/oc4j-connector-factories-9_04.dtd">
<oc4j-connector-factories>
   <connector-factory location="eis/OracleJCAAdapter/DefaultConnection"
connector-name="IWAFJCA10">
   <config-property name="IWayHome" value="../../adapters/application"/>
   <config-property name="IWayConfig" value="jca_sample"/>
   <config-property name="IWayRepoURL" value=""/>
   <config-property name="IWayRepoUser" value=""/>
   <config-property name="IWayRepoPassword" value=""/>
   <config-property name="logLevel" value="debug"/>
   </connector-factory>
</oc4j-connector-factories>
```

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

Parameter Name	Description
IWayHome	The base installation directory for the OracleAS packaged application adapter.
IWayConfig	The adapter configuration name as defined in Application Explorer. For example, OracleWLS Application Adapter for J.D. Edwards OneWorld has a preconfigured jca_sample configuration in Application Explorer.
IWayRepoURL	The URL to use when opening a connection to the database. This is necessary only when using an Oracle database as the BSE repository. See "Configuring BSE System Settings" on page 2-3 for more information.
IWayRepoUser	User name to use when connecting to the database. This is necessary only when using an Oracle database as the BSE repository. See "Configuring BSE System Settings" on page 2-3 for more information.
IWayRepoPassword	Password. If provided, it overwrites configuration. This is necessary only when using an Oracle database as the BSE repository. See "Configuring BSE System Settings" on page 2-3 for more information.
loglevel	It overwrites the level set by the ManagedConnectionFactory property.

Creating Multiple Managed Connector Factory Objects

To establish multiple managed connector factory objects, you must edit the ra.xml file and add more <connector-factory> nodes. For example, the default jca_sample configuration in Application Explorer is represented in the ra.xml file as follows:

```
<?xml version="1.0"?>
<!DOCTYPE oc4j-connector-factories PUBLIC "-//Oracle//DTD Oracle Connector
9.04//EN"
"http://xmlns.oracle.com/ias/dtds/oc4j-connector-factories-9_04.dtd">
<oc4j-connector-factories>
   <connector-factory location="eis/OracleJCAAdapter/DefaultConnection"
connector-name="IWAFJCA10">
   <config-property name="IWayHome" value="../../adapters/application"/>
   <config-property name="IWayConfig" value="jca_sample"/>
   <config-property name="IWayRepoURL" value=""/>
   <config-property name="IWayRepoUser" value=""/>
   <config-property name="IWayRepoPassword" value=""/>
   <config-property name="logLevel" value="debug"/>
   </connector-factory>
</oc4j-connector-factories>
```

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

```
<?xml version="1.0"?>
```

```
<!DOCTYPE oc4j-connector-factories PUBLIC "-//Oracle//DTD Oracle Connector
9.04//EN"
"http://xmlns.oracle.com/ias/dtds/oc4j-connector-factories-9_04.dtd">
<oc4j-connector-factories>
   <connector-factory location="eis/OracleJCAAdapter/DefaultConnection1"
connector-name="IWAFJCA10">
   <config-property name="IWayHome" value="../../adapters/application"/>
   <config-property name="IWayConfig" value="jca_sample"/>
   <config-property name="IWayRepoURL" value=""/>
   <config-property name="IWayRepoUser" value=""/>
   <config-property name="IWayRepoPassword" value=""/>
   <config-property name="logLevel" value="debug"/>
   </connector-factory>
<connector-factory location="eis/OracleJCAAdapter/DefaultConnection2"
connector-name="IWAFJCA10">
   <config-property name="IWayHome" value="../../adapters/application"/>
   <config-property name="IWayConfig" value="jca_sample2"/>
   <config-property name="IWayRepoURL" value=""/>
   <config-property name="IWayRepoUser" value=""/>
   <config-property name="IWayRepoPassword" value=""/>
   <config-property name="logLevel" value="debug"/>
   </connector-factory>
</oc4j-connector-factories>
```

How to Write a Java Application Client Using the CCI API

The following example shows the code structure for using CCI with packaged application adapters. The code sample is shown in four steps.

> **Note:** The OracleWLS Application Adapter for J.D. Edwards OneWorld does not support invocation through the CCI API. Only invocation using Oracle Service Bus (OSB) is supported. The following section is only provided for reference purposes.

Step 1. Obtain the Connection Factory

The connection factory is obtained by JNDI lookup.

```
InitialContext context = new InitialContext();
ConnectionFactory cf = (ConnectionFactory)context.lookup(iwayJndi)
```

Step 2. Obtaining a Connection for the Adapter

IWAFConnectionSpec is an implementation of ConnectionSpec used for creating a design time or run-time service adapter connection. The ConnectionSpec has seven parameters. Connection Pooling is fully supported and established based on these parameters, except log level.

Parameter Name	Description
adapterName	Name of the packaged application adapter.
config -	Adapter configuration name. NOT REQUIRED FOR IWAEAdapter.
language	Default is `en`.
country	Default is `us`.

Parameter Name	Description
userName	User name. If provided, it overwrites configuration.
password	Password. If provided, it overwrites configuration.
logLevel	It overwrites the level set by the ManagedConnectionFactory property.

> **Note:** Currently the OracleAS Adapter J2CA supports only basic security mapping. The DEBUG log level provides detailed information on the mapping behavior. It functions as follows:
>
> - If the user name and password are not set, and no security is provided by the application server, the OracleAS Adapter J2CA will still let it pass and rely on the adapter configuration security information.
> - If the user name and password are set, these values will overwrite the adapter configuration. The OracleAS Adapter J2CA compares this information with the security information provided by the application server and log in case the values do not match. However, it still allows the information through.

The iWAFConnectionSpec can initiate an interaction with J.D. Edwards OneWorld if the adapter name and configuration parameters are specified in the ConnectionSpec. For example,

```
iWAFConnectionSpec cs = new IWAFConnectionSpec();
  cs.setAdapterName(ADAPTER);
  cs.setConfig(TARGET);
  cs.setLogLevel(LOG_LEVEL);  // Adapter layer log level
  Connection c = cf.getConnection(cs);// where cf is the connection factory
```

In this snippet, `ADAPTER` and `TARGET` refer to the adapter being deployed, in this case J.D. Edwards OneWorld, and the name of a target defined in Application Explorer. See "Complete Code Sample" on page 3-7 for more information.

Step 3. Create interaction with interactionSpec for runtime

```
Interaction i = c.createInteraction();
  IWAFInteractionSpec is = new IWAFInteractionSpec();
  is.setFunctionName(IWAFInteractionSpec.PROCESS);
```

Two functions can be set: PROCESS and IWAE. PROCESS is used at run-time. IWAE is used when you are using the IAEAdapter at design time.

Step 4. Create Input Record and Execute Interaction

In this case, to complete the EIS invocation, a schema is provided by Application Explorer.

A standard J2CA Indexed Record is used in this example:

```
// Use JCA IndexRecord, named "input" for run-time processing.
IndexedRecord rIn = cf.getRecordFactory().createIndexedRecord("input");
rIn.add(msg_run);
  IndexedRecord rOut = (IndexedRecord)i.execute(is, rIn);
```

```
System.out.println((String)rOut.get(0));
```

A special record is supported in this example:

```
//IWAFRecord rIn = new IWAFRecord("input");
//rIn.setRootXML(msg_run);
//IWAFRecord response = executeRunInteraction(c, rIn);
  //IWAFRecord rOut = (IWAFRecord)i.execute(is, rIn);
//System.out.println(rOut.getRootXML());
```

Where *msg_run* is an instance XML document generated from the schema created by Application Explorer. For example, the following is a sample J.D. Edwards OneWorld request XML document.

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml version="1.0" encoding="UTF-8"?>
<jdeRequest type="callmethod" user="JDE" pwd="JDE" environment="DV7333" session=""
sessionidle="">
<callMethod name="AddressBookMasterMBF" app="" runOnError="" trans="">
<params>
  <param name="cActionCode">A</param>
  <param name="cUpdateMasterFile">1</param>
  <param name="cProcessEdits">1</param>
  <param name="cSuppressErrorMessages"/>
  <param name="szErrorMessageID" />
  <param name="szVersion">ZJDE0001</param>
  <param name="mnSameAsExcept" />
  <param name="mnAddressBookNumber" id="1" />
  <param name="szLongAddressNumber" />
  <param name="szTaxId" />
  <param name="szSearchType">C</param>
  <param name="szAlphaName">John J. Smith</param>
  <param name="szSecondaryAlphaName">John J. Smith</param>
  <param name="szMailingName">John J. Smith</param>
  <param name="szSecondaryMailingName">John J. Smith</param>
  <param name="szDescriptionCompressed" />
  <param name="szBusinessUnit"/>
  <param name="szAddressLine1">1 Main St.</param>
  <param name="szAddressLine2">Apt 101</param>
  <param name="szAddressLine3"/>
  <param name="szAddressLine4"/>
  <param name="szPostalCode">75000</param>
  <param name="szCity">AnyTown</param>
  <param name="szCounty"/>
  <param name="szState">TX</param>
  <param name="szCountry">US</param>
  <param name="szCarrierRoute" />
  <param name="szBulkMailingCenter" />
  <param name="szPrefix1" />
  <param name="szPhoneNumber1">917-123-4567</param>
  <param name="szPhoneNumberType1" />
  <param name="szPhoneAreaCode2" />
  <param name="szPhoneNumber2" />
  <param name="szPhoneNumberType2" />
  <param name="cPayablesYNM">Y</param>
  <param name="cReceivablesYN">N</param>
  <param name="cEmployeeYN">N</param>
  <param name="cUserCode" />
  <param name="cARAPNettingY">N</param>
  <param name="cPersonCorporationCode" />
  <param name="szCertificate" />
  <param name="szAddlIndTaxID" />
  <param name="szCreditMessage" />
  <param name="szLanguage" />
  <param name="szIndustryClassification" />
  <param name="cEMail" />
  <param name="szCategoryCode01" />
  <param name="szRemark"/>
  <param name="szUserReservedCode"/>
  <param name="jdUserReservedDate"/>
  <param name="mnUserReservedAmount"/>
  <param name="mnUserReservedNumber"/>
  <param name="szUserReservedReference"/>
  <param name="jdDateEffective" />
  <param name="szRemark1" /> </params>
<onError abort=""/>
</callMethod>
</jdeRequest>
```

Complete Code Sample

The following is a sample of the complete code:

```
import javax.resource.cci.*;
import com.ibi.afjca.cci.*;
import com.ibi.afjca.spi.*;

/**
 * The purpose of this sample is to illustrate how to use the IWAF Universal
 * JCA connector.
 */
public class IWAFJCASimple {

private static String  HOME    = "c:/iway/xfoc/components/iwafcont/dist";
private static String  CONFIG    = "base";
private static String  LOG_LEVEL = "FATAL";

private static String ADAPTER = "JDE";
private static String TARGET  = "JDE_connection";

// Input Message
private static String msg_run = "<JDE/>";

  public static void main(String[] args) throws Exception {

// 1. Getting the Connection factory through JNDI lookup
// --------------------------------------------------------
  InitialContext context = new InitialContext();
  ConnectionFactory cf = (ConnectionFactory)context.lookup(iwayJndi)
  // 2. Getting a connection for a particular adapter target, in this case JDE
// --------------------------------------------------------
  IWAFConnectionSpec cs = new IWAFConnectionSpec();
  cs.setAdapterName(ADAPTER);
  cs.setConfig(TARGET);
  cs.setLogLevel(LOG_LEVEL);  // Adapter layer log level
  Connection c = cf.getConnection(cs);// where cf is the connection factory

  // 3. Create interaction with interactionSpec for RUNTIME
// --------------------------------------------------------
  Interaction i = c.createInteraction();
  IWAFInteractionSpec is = new IWAFInteractionSpec();
  is.setFunctionName("PROCESS");
  // 4. Create input Record and execute interaction
// --------------------------------------------------------

  // 4.1 Using JCA standard Indexed Record
// Use JCA IndexRecord, named "input" for run-time processing.
IndexedRecord rIn = cf.getRecordFactory().createIndexedRecord("input");
rIn.add(msg_run);
  IndexedRecord rOut = (IndexedRecord)i.execute(is, rIn);
System.out.println((String)rOut.get(0));

  // 4.2 Our own Record is supported here
//IWAFRecord rIn = new IWAFRecord("input");
//rIn.setRootXML(msg_run);
//IWAFRecord response = executeRunInteraction(c, rIn);
  //IWAFRecord rOut = (IWAFRecord)i.execute(is, rIn);
//System.out.println(rOut.getRootXML());

  } // main()

}
```

4

Configuring Outbound Processing Using Oracle Service Bus (BSE Configuration)

OracleWLS Application Adapter for J.D. Edwards OneWorld integrates seamlessly with Oracle Service Bus (OSB) to facilitate Web service integration. OSB is based on the Service-Oriented Architecture (SOA). It consumes adapter services exposed as Web Service Definition Language (WSDL) documents.

This chapter includes the following topics:

Overview of Application Adapter Integration with Oracle Service Bus

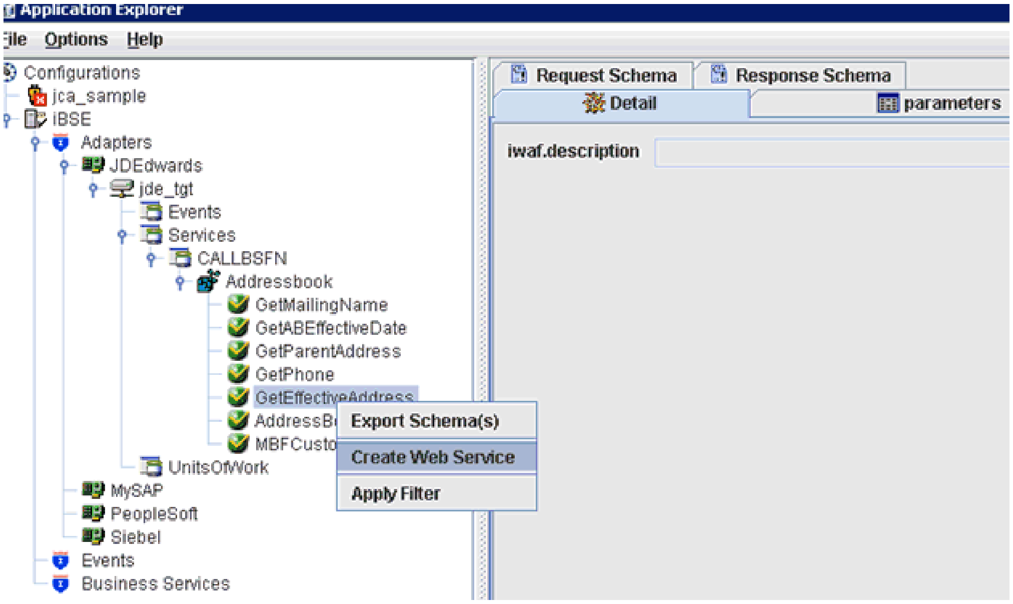
To integrate with Oracle Service Bus (OSB), OracleWLS Application Adapter for J.D. Edwards OneWorld must be deployed on the same Oracle WebLogic Server as OSB. The underlying adapter services must be exposed as WSDL files, which are generated during design time in Oracle WebLogic Server Adapter Application Explorer (Application Explorer) for both request-response (outbound) and event notification (inbound) services of the adapter.

Publishing a WSDL From Application Explorer to Oracle Service Bus

This section describes how to publish a WSDL from Application Explorer (BSE configuration) to Oracle Service Bus.

1. Start Application Explorer, connect to a BSE configuration, and connect to a J.D. Edwards OneWorld target.

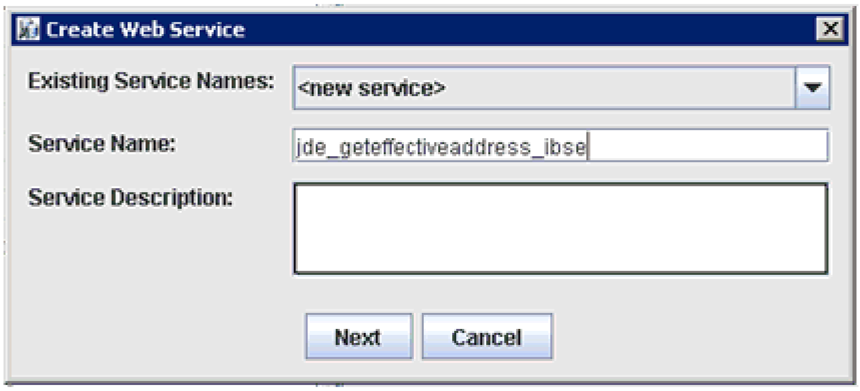
   For more information, see Chapter 2, "Configuring OracleWLS Application Adapter for J.D. Edwards One World".



2. Expand the **Services** node, followed by **CALLBSFN** and **Addressbook**.

3. Right-click the **GetEffectiveAddress** business object and select **Create Web Service** from the context menu.

   The Create Web Service dialog box is displayed.



Existing Service Names: <new service>

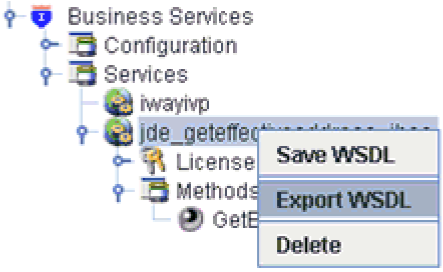
Service Name: jde_geteffectiveaddress_ibse

Service Description:

4. Enter a service name and click **Next**.

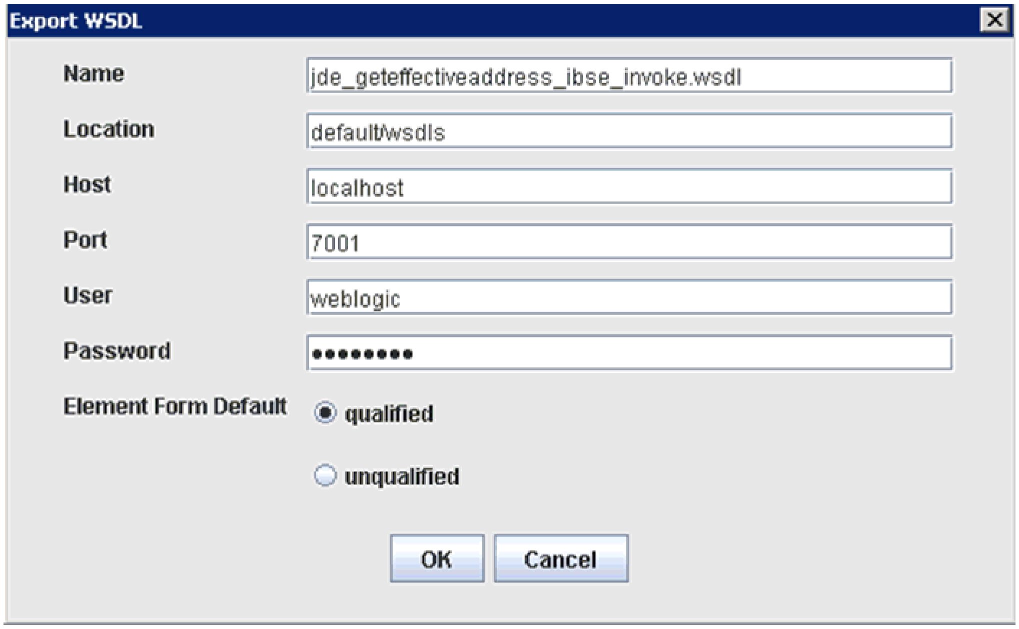
5. Click **OK** on the next dialog box that is displayed.

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



6. Right-click the new Web service and select **Export WSDL** from the context menu.

The Export WSDL dialog box is displayed.



Export WSDL

Name: jde_geteffectiveaddress_ibse_invoke.wsdl

Location: default/wsdls

Host: localhost

Port: 7001

User: weblogic

Password: ●●●●●●●●

Element Form Default: ● qualified ○ unqualified

7. In the Name field, a default file name for the WSDL file is provided. You can accept the default or provide your own.
8. In the Location field, enter the location where you want to publish the WSDL document.

   The location is composed of an Oracle Service Bus project name and optionally, one or more folder names. The project name and any folder names must be separated by a back slash character “\”.
9. In the Host field, enter the name of the machine where Oracle WebLogic Server is running.
10. In the Port field, enter the port for the domain you are using.
11. In the User field, enter your username to access Oracle Service Bus.
12. In the Password field, enter your password to access Oracle Service Bus.
13. Ensure that **qualified** is selected as the element form, which is the default.
14. Click **OK**.

    The WSDL is published to the location specified in the Export WSDL dialog box and is now available for use with a Business Service or Proxy Service in Oracle Service Bus.

Starting Oracle Service Bus and Creating Project Folders

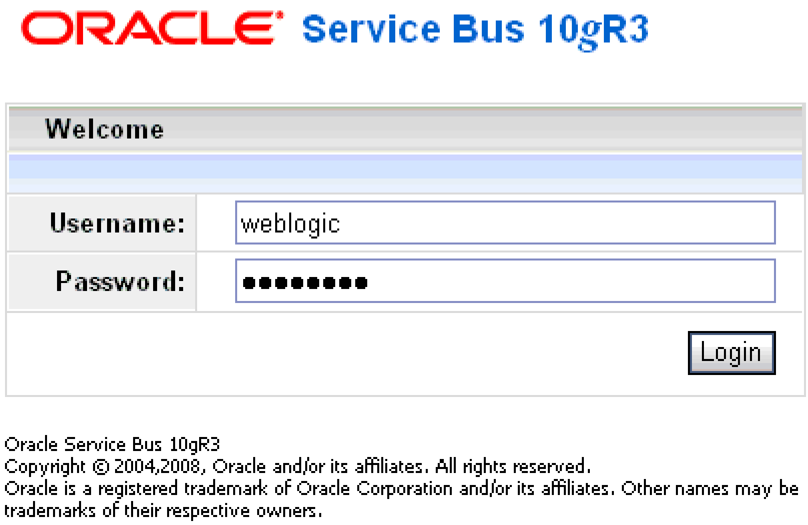
This section describes how to access the Oracle Service Bus Console and create project folders.

1. Start the Oracle WebLogic Server for the Oracle WebLogic Server domain that you have configured.
2. Open the Oracle Service Bus Console in a Web browser by entering the following URL:

   ```
   http://hostname:port/sbconsole
   ```

Where *hostname* is the name of the machine where Oracle WebLogic Server is running and *port* is the port for the domain you are using. The port for the default domain is 7001.

The Oracle Service Bus Console logon page is displayed.



ORACLE Service Bus 10gR3

Welcome

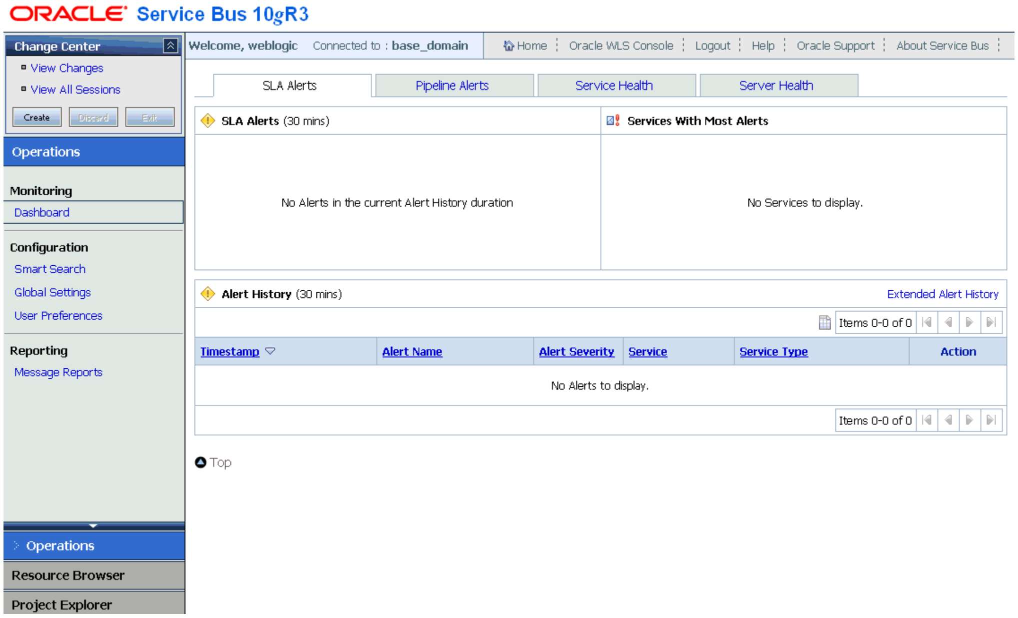
Username: weblogic

Password: ●●●●●●●●●

Oracle Service Bus 10gR3
Copyright © 2004,2008, Oracle and/or its affiliates. All rights reserved.
Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

3. Log on to the Oracle Service Bus Console using a valid user name and password.

   The Oracle Service Bus Console home page is displayed.



ORACLE Service Bus 10gR3

Welcome, weblogic Connected to : base_domain

SLA Alerts (30 mins)

No Alerts in the current Alert History duration

Services With Most Alerts

No Services to display.

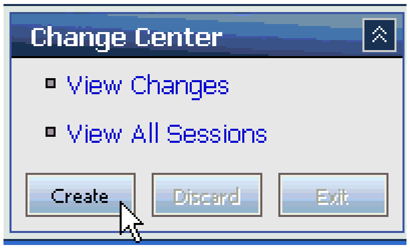
Alert History (30 mins)

Items 0-0 of 0

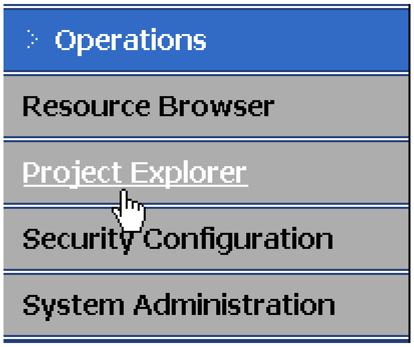
Timestamp	Alert Name	Alert Severity	Service	Service Type	Action
No Alerts to display.					

Items 0-0 of 0

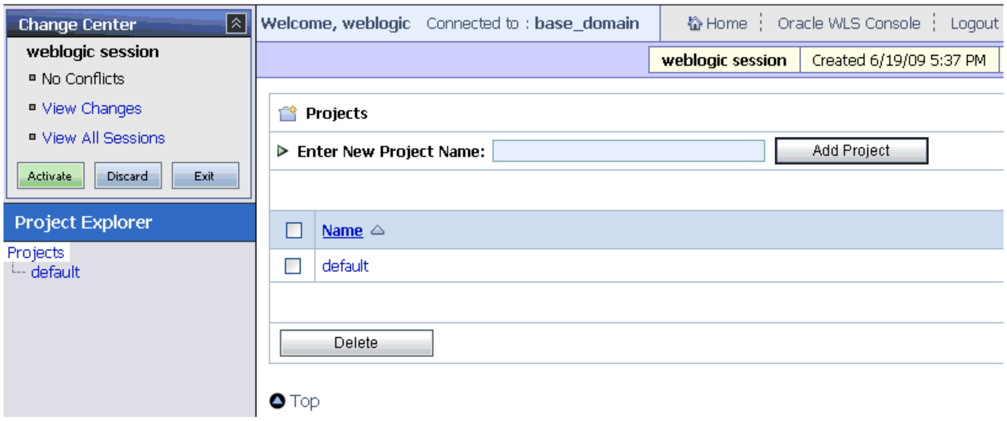
4. Click **Create** in the Change Center area to start a new Oracle Service Bus session.



5. Click **Project Explorer** in the left pane.

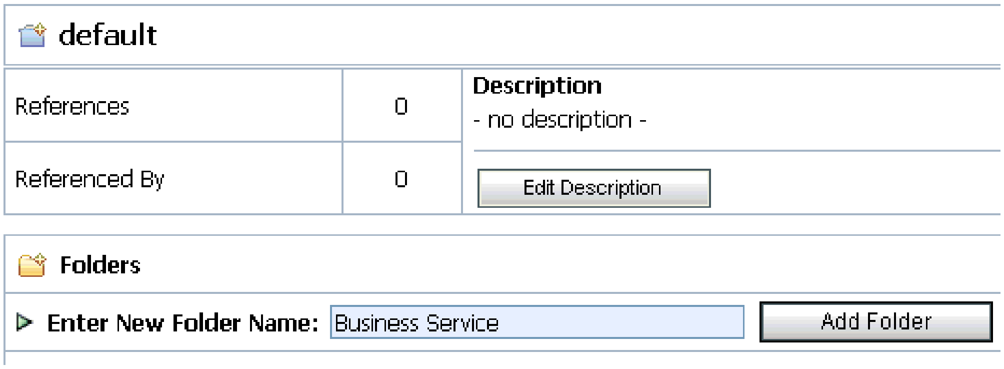


The Project Explorer page is displayed.



6. Click the **default** project node in the left pane.

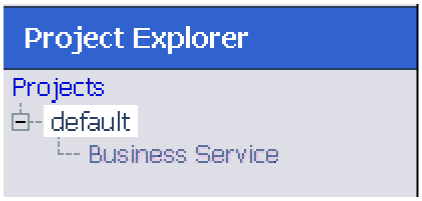
The default project page is displayed.



Enter New Folder Name: Business Service

7. In the Enter New Folder Name field, type **Business Service** and click **Add Folder**.

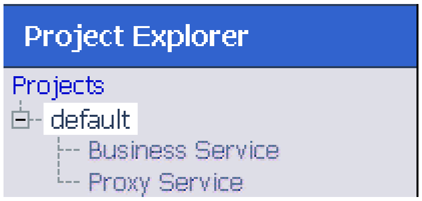
The Business Service folder is listed in the left pane below the default project node.


Project Explorer

Projects
- default
  - Business Service

8. In the Enter New Folder Name field, type **Proxy Service** and click **Add Folder**.

   The Business Service and Proxy Service folders are listed in the left pane below the default project node.

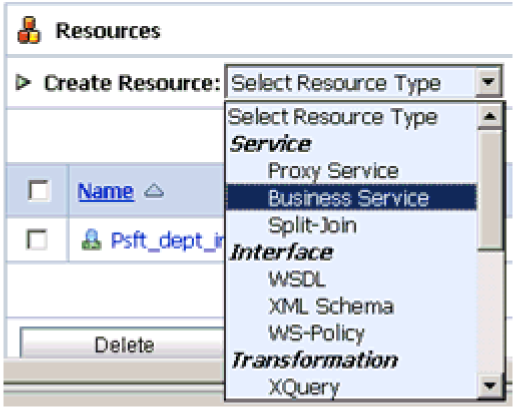

Project Explorer

Projects
- default
  - Business Service
  - Proxy Service

Configuring a File Type Business Service

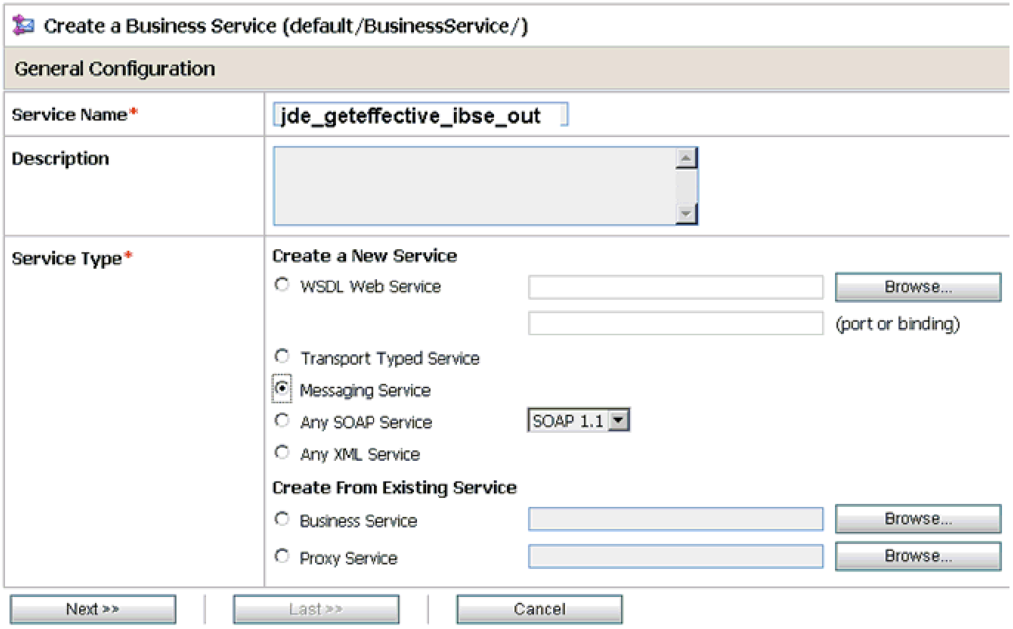
This section describes how to configure a File type Business Service using the Oracle Service Bus Console.

1. Select the **Business Service** folder you created in the left pane.



2. In the right pane, select **Business Service** from the Create Resource menu.

   The Create a Business Service - General Configuration page is displayed.



Create a Business Service (default/BusinessService/)

General Configuration

Service Name* jde_geteffective_ibse_out

Description

Service Type*

Create a New Service

○ WSDL Web Service (port or binding)

○ Transport Typed Service

● Messaging Service

○ Any SOAP Service SOAP 1.1

○ Any XML Service

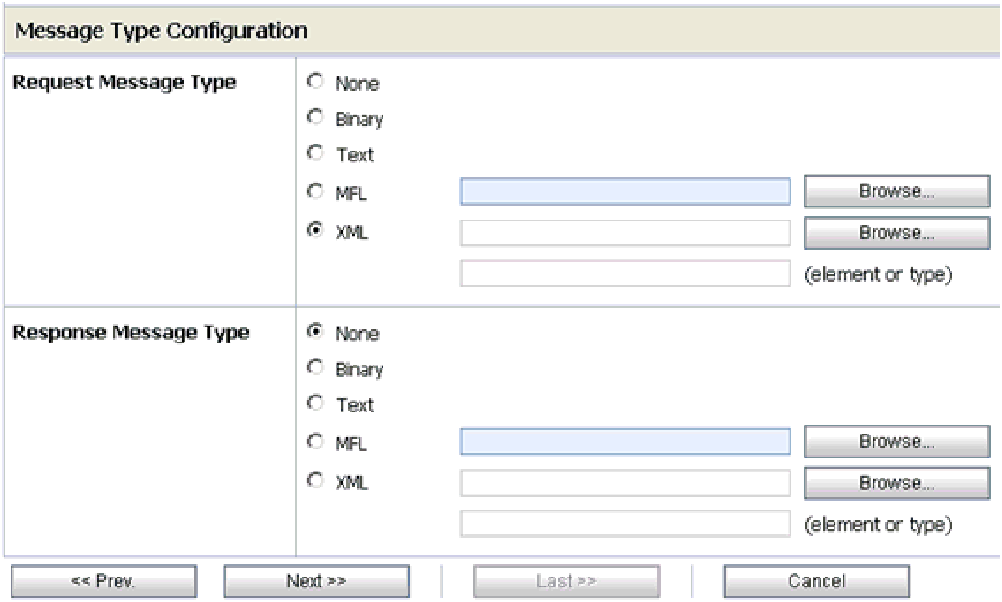
Create From Existing Service

○ Business Service

○ Proxy Service

3. Provide a name for the Business Service and from the Service Type area select **Messaging Service**.

4. Click **Next**.



Message Type Configuration

Request Message Type

○ None

○ Binary

○ Text

○ MFL

● XML (element or type)

Response Message Type

● None

○ Binary

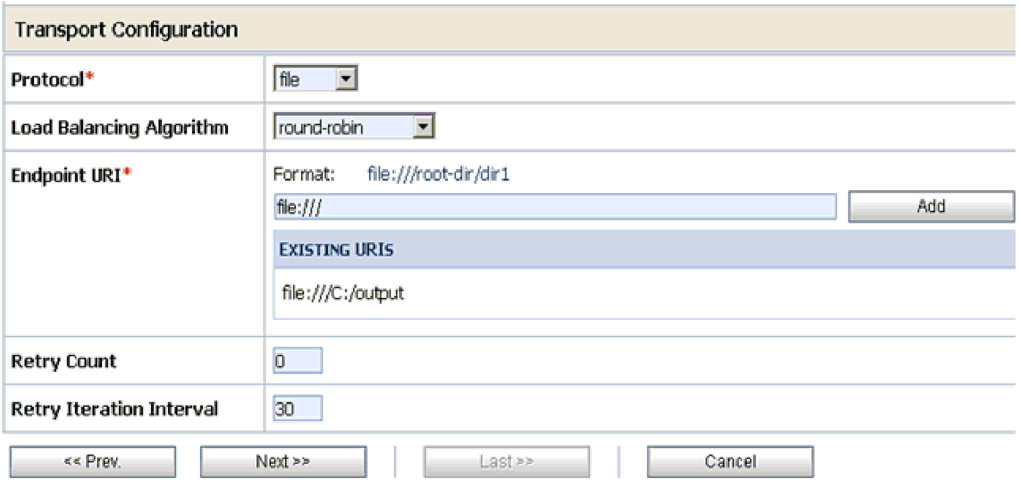
○ Text

○ MFL

○ XML (element or type)

5. Select **XML** as the Request Message Type and **None** as the Response Message Type.

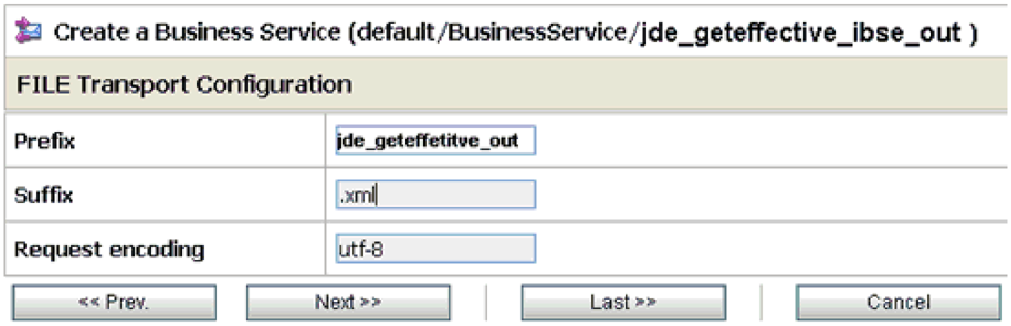
6. Click **Next**.



Transport Configuration

Protocol*	file
Load Balancing Algorithm	round-robin
Endpoint URI*	Format: file:///root-dir/dir1 file:/// EXISTING URIS file:///C:/output
Retry Count	0
Retry Iteration Interval	30

7. Select **file** from the Protocol drop-down list.
8. Enter the path to a destination folder on your file system in the Endpoint URI field and click **Add**.
9. Click **Next**.

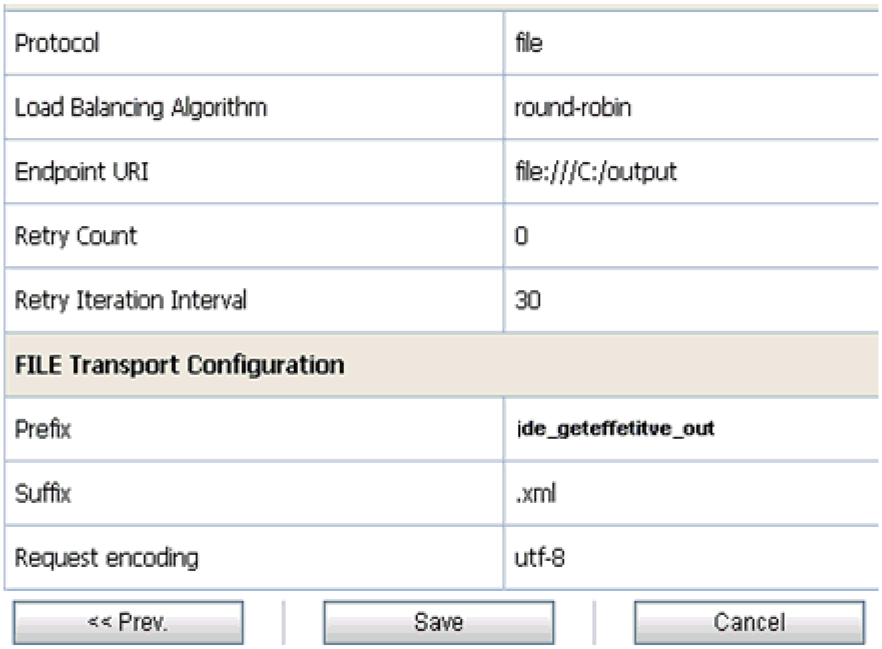


Create a Business Service (default/BusinessService/jde_geteffective_ibse_out)

FILE Transport Configuration

Prefix	jde_geteffetitve_out
Suffix	.xml
Request encoding	utf-8

10. Enter the prefix and suffix for the output file to be received and click **Next**.



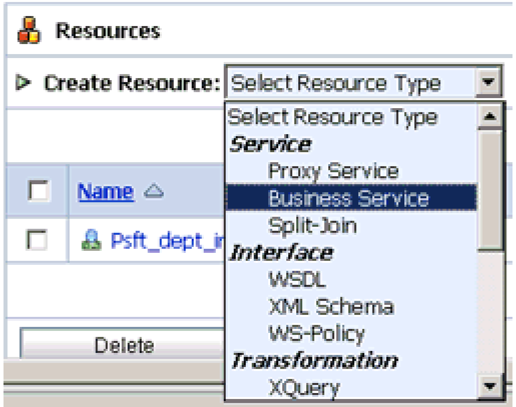
Protocol	file
Load Balancing Algorithm	round-robin
Endpoint URI	file:///C:/output
Retry Count	0
Retry Iteration Interval	30
FILE Transport Configuration	
Prefix	**jde_geteffetitve_out**
Suffix	.xml
Request encoding	utf-8

11. Review all the information for your Business Service and click **Save**.

Configuring a WSDL Type Business Service

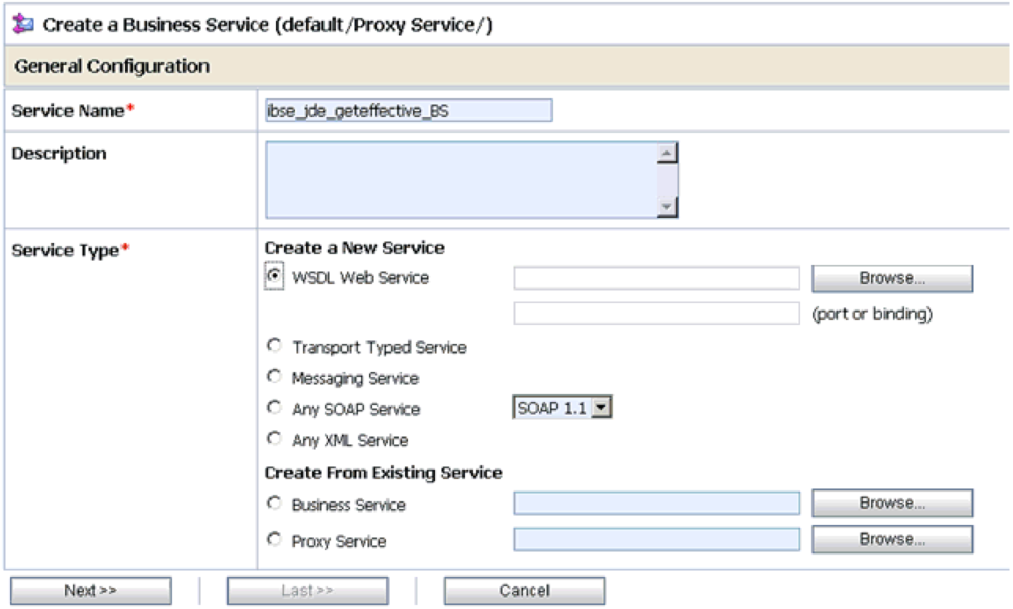
This section describes how to configure a WSDL type Business Service using the Oracle Service Bus Console.

1. Select the **Business Service** folder you created in the left pane.



2. In the right pane, select **Business Service** from the Create Resource menu.

   The Create a Business Service - General Configuration page is displayed.



Create a Business Service (default/Proxy Service/)

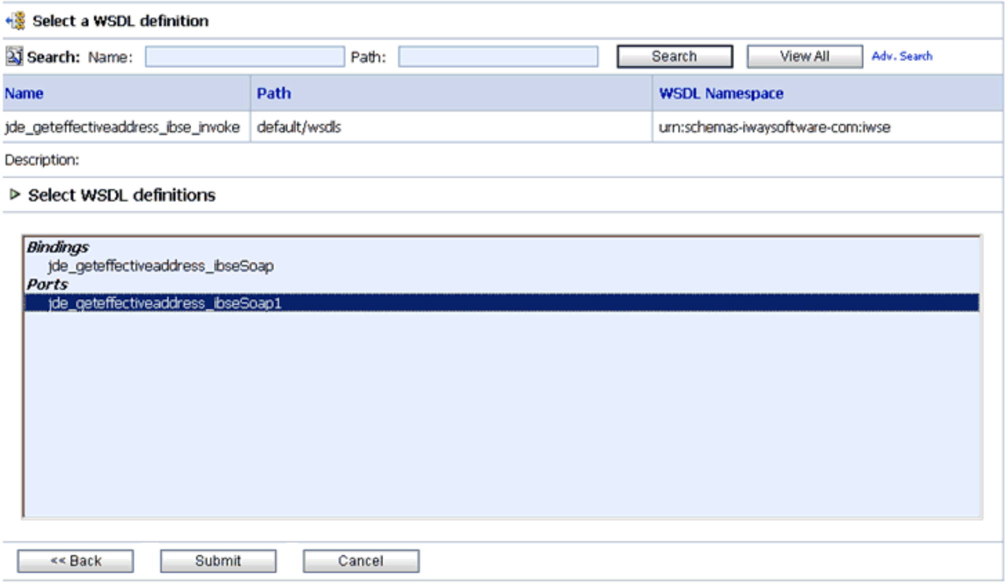
General Configuration

Service Name*	ibse_jde_geteffective_BS
Description	
Service Type*	**Create a New Service** ● WSDL Web Service (port or binding) ○ Transport Typed Service ○ Messaging Service ○ Any SOAP Service — SOAP 1.1 ○ Any XML Service **Create From Existing Service** ○ Business Service ○ Proxy Service

3. Provide a name for the Business Service and from the Service Type area select **WSDL Web Service**.

4. Click **Browse**.

   The Select a WSDL Definition dialog box is displayed.



Select a WSDL definition

Search: Name: Path:

Name	Path	WSDL Namespace
jde_geteffectiveaddress_ibse_invoke	default/wsdls	urn:schemas-iwaysoftware-com:iwse

Description:

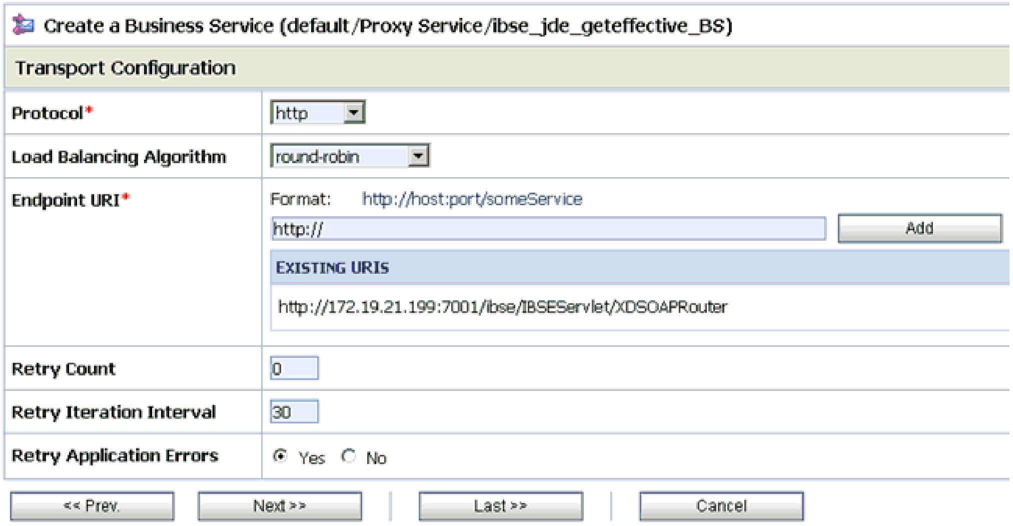
Select WSDL definitions

Bindings
jde_geteffectiveaddress_ibseSoap
Ports
jde_geteffectiveaddress_ibseSoap1

5. Select an available WSDL definition and click **Submit**.

   You are returned to the General Configuration page where the WSDL you selected is now available.

6. Click **Next**.



Create a Business Service (default/Proxy Service/ibse_jde_geteffective_BS)

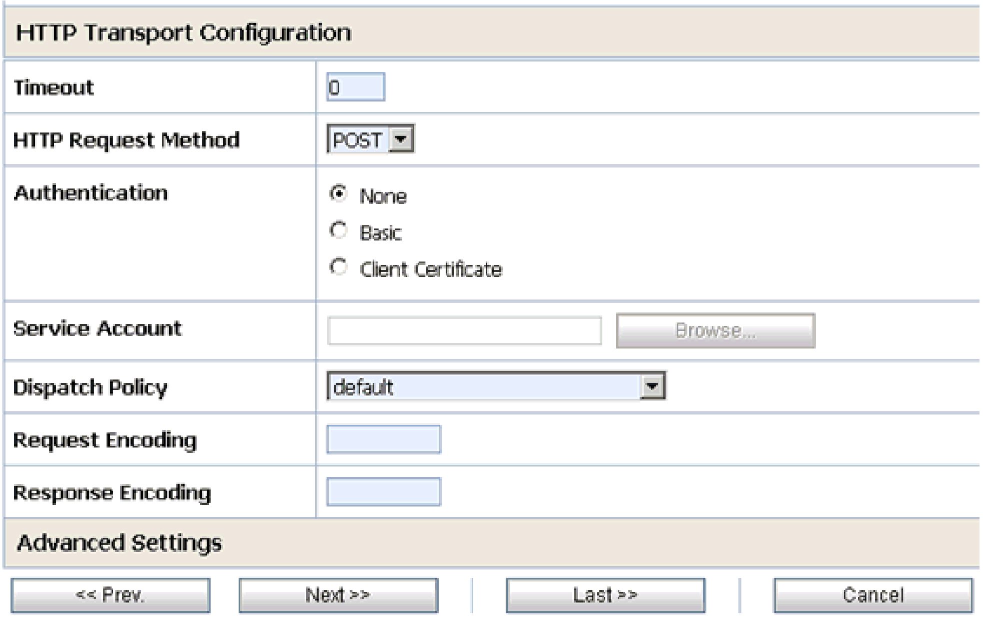
Transport Configuration

Protocol*	http
Load Balancing Algorithm	round-robin
Endpoint URI*	Format: http://host:port/someService http:// EXISTING URIS http://172.19.21.199:7001/ibse/IBSEServlet/XDSOAPRouter
Retry Count	0
Retry Iteration Interval	30
Retry Application Errors	● Yes ○ No

7. Select **http** from the Protocol drop-down list.

8. Enter the Endpoint URI in HTTP format and click **Add**.

9. Click **Next**.

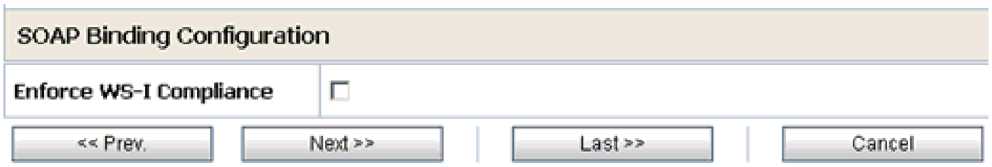


HTTP Transport Configuration

Timeout	0
HTTP Request Method	POST
Authentication	● None ○ Basic ○ Client Certificate
Service Account	
Dispatch Policy	default
Request Encoding	
Response Encoding	

Advanced Settings

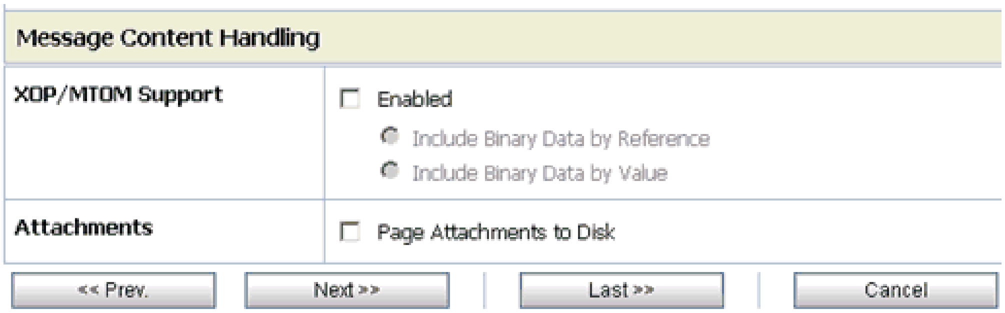
10. Click **Next**.



SOAP Binding Configuration

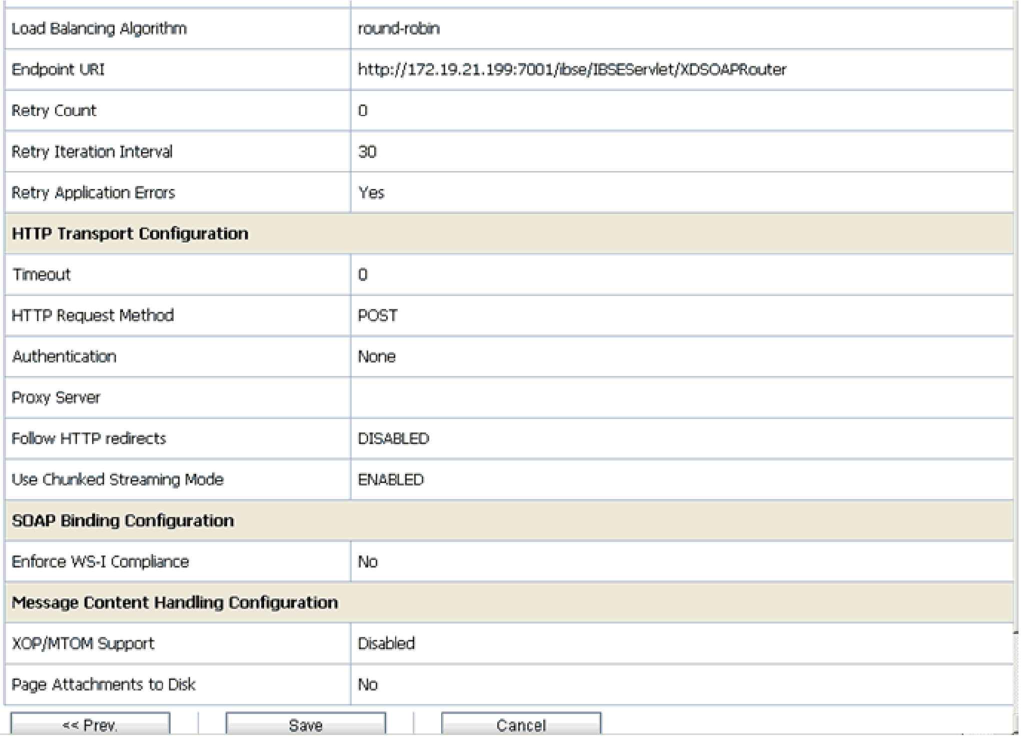
Enforce WS-I Compliance	□

11. Click **Next**.



Message Content Handling	
XOP/MTOM Support	□ Enabled ○ Include Binary Data by Reference ○ Include Binary Data by Value
Attachments	□ Page Attachments to Disk

12. Click **Next**.



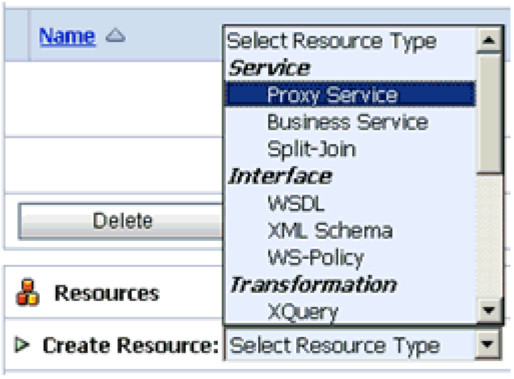
Load Balancing Algorithm	round-robin
Endpoint URI	http://172.19.21.199:7001/ibse/IBSEServlet/XDSOAPRouter
Retry Count	0
Retry Iteration Interval	30
Retry Application Errors	Yes
HTTP Transport Configuration	
Timeout	0
HTTP Request Method	POST
Authentication	None
Proxy Server	
Follow HTTP redirects	DISABLED
Use Chunked Streaming Mode	ENABLED
SOAP Binding Configuration	
Enforce WS-I Compliance	No
Message Content Handling Configuration	
XOP/MTOM Support	Disabled
Page Attachments to Disk	No

13. Review all the information for your Business Service and click **Save**.

Configuring a Proxy Service

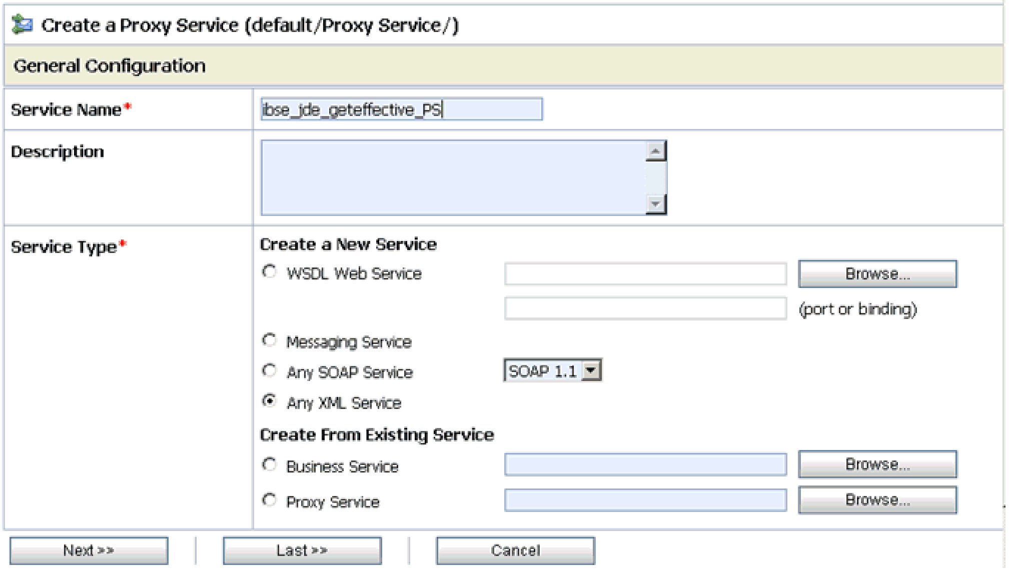
This section describes how to configure a Proxy Service using the Oracle Service Bus Console.

1. Select the **Proxy Service** folder you created in the left pane.



2. In the right pane, select **Proxy Service** from the Create Resource menu.

   The Create a Proxy Service - General Configuration page is displayed.



Create a Proxy Service (default/Proxy Service/)

General Configuration

Service Name*: ibse_jde_geteffective_PS

Description:

Service Type*:

Create a New Service

○ WSDL Web Service [Browse...]

(port or binding)

○ Messaging Service

○ Any SOAP Service SOAP 1.1

● Any XML Service

Create From Existing Service

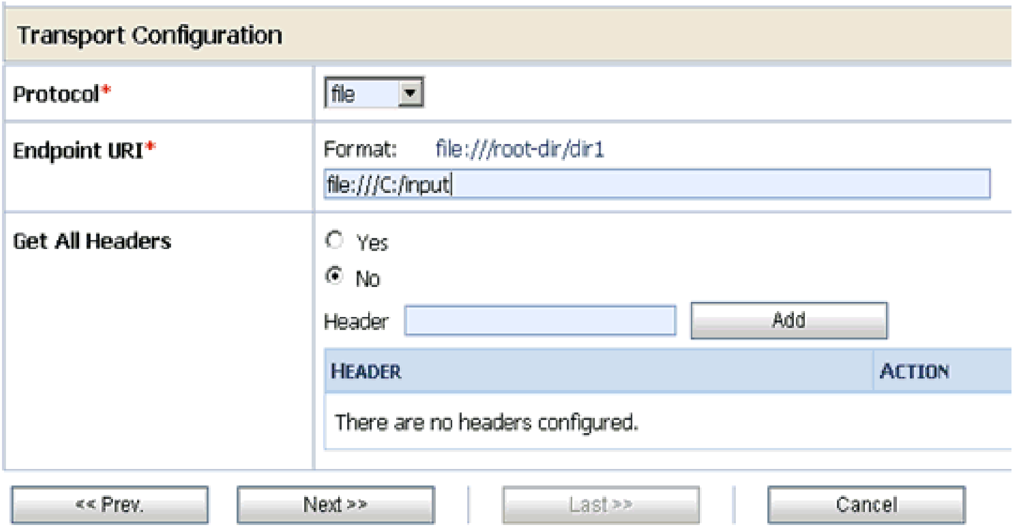
○ Business Service [Browse...]

○ Proxy Service [Browse...]

[Next >>] [Last >>] [Cancel]

3. Provide a name for the Proxy Service and from the Service Type area select **Any XML Service**.

4. Click **Next**.



Transport Configuration

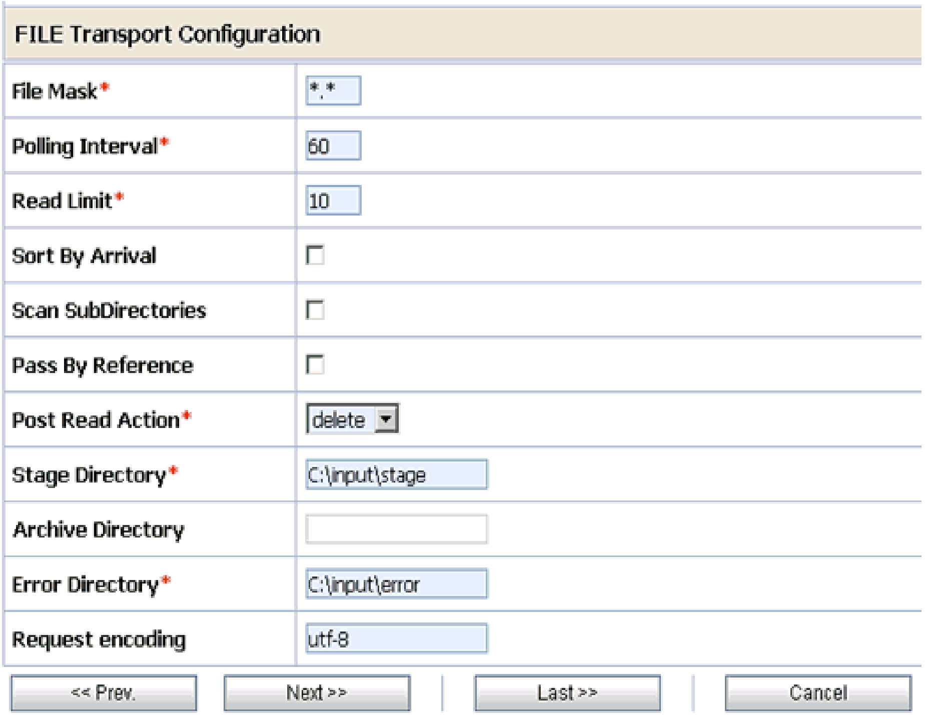
Protocol*	file
Endpoint URI*	Format: file:///root-dir/dir1 file:///C:/input
Get All Headers	○ Yes ● No Header

HEADER	ACTION
There are no headers configured.	

5. Select **file** from the Protocol drop-down list.

6. Enter the path to an input folder on your file system in the Endpoint URI field.

7. Click **Next**.

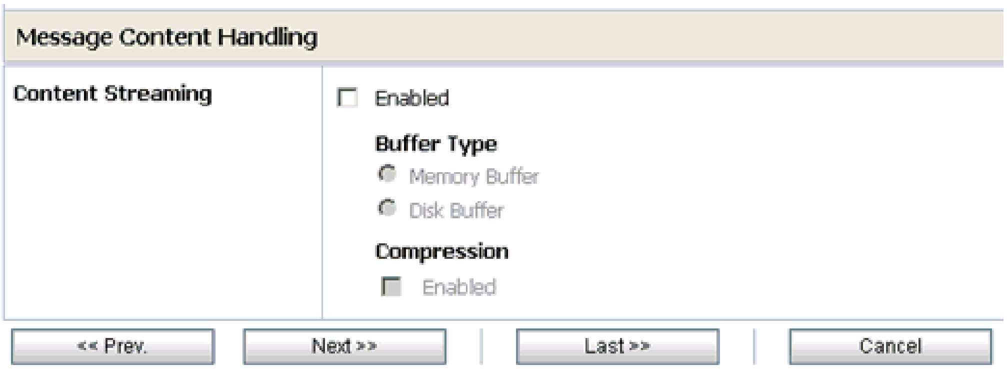


FILE Transport Configuration

File Mask*	*.*
Polling Interval*	60
Read Limit*	10
Sort By Arrival	□
Scan SubDirectories	□
Pass By Reference	□
Post Read Action*	delete
Stage Directory*	C:\input\stage
Archive Directory	
Error Directory*	C:\input\error
Request encoding	utf-8

8. Provide any folder locations on your file system for the Stage Directory and Error Directory fields.

9. Click **Next**.



Message Content Handling

Content Streaming

☐ Enabled

Buffer Type

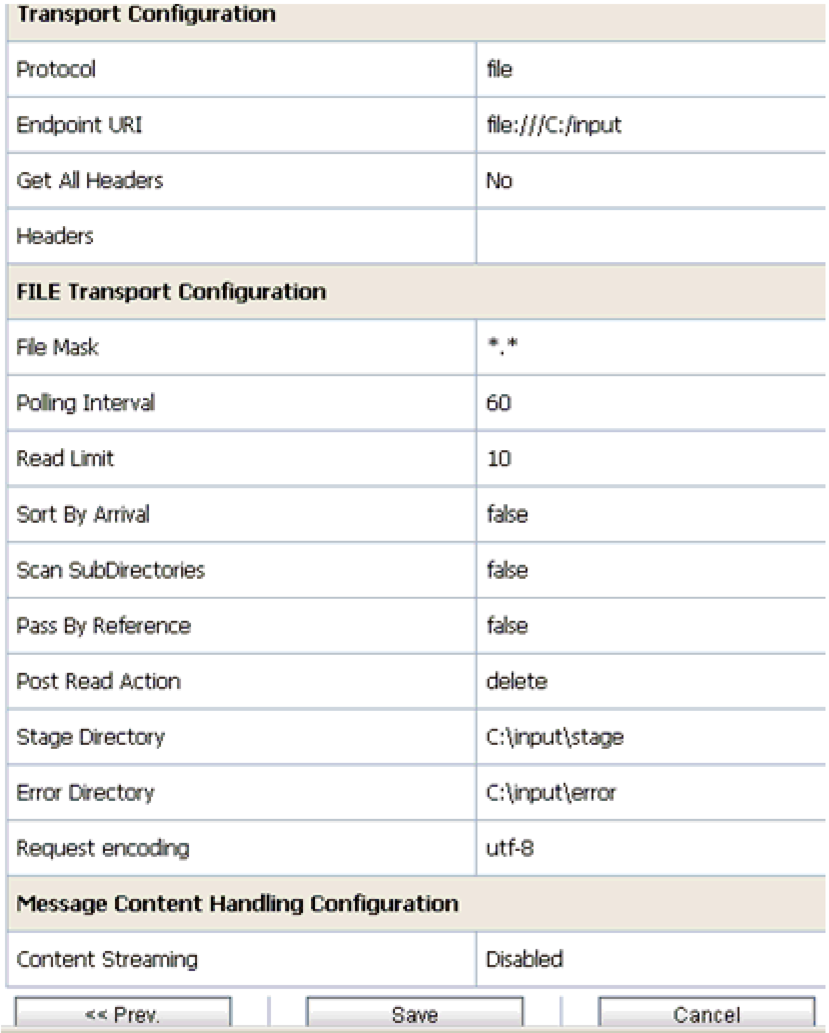
○ Memory Buffer

○ Disk Buffer

Compression

☐ Enabled

10. Click **Next**.



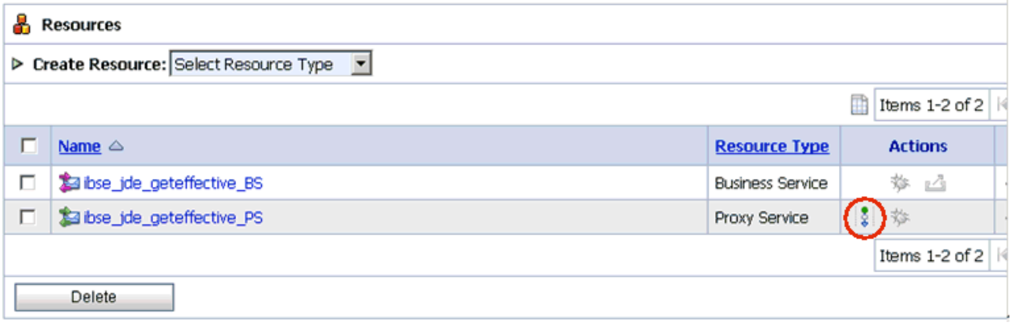
Transport Configuration	
Protocol	file
Endpoint URI	file:///C:/input
Get All Headers	No
Headers	
FILE Transport Configuration	
File Mask	*.*
Polling Interval	60
Read Limit	10
Sort By Arrival	false
Scan SubDirectories	false
Pass By Reference	false
Post Read Action	delete
Stage Directory	C:\input\stage
Error Directory	C:\input\error
Request encoding	utf-8
Message Content Handling Configuration	
Content Streaming	Disabled

11. Review all the information for your Proxy Service and click **Save**.

Configuring a Pipeline

This section describes how to configure a Pipeline using the Oracle Service Bus Console.

1. Click the **Edit Message Flow** icon in the row of the Proxy Service you created.



Resources

Create Resource: Select Resource Type

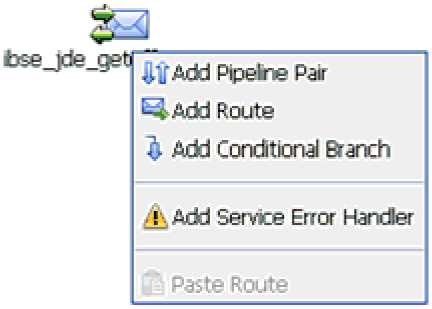
Items 1-2 of 2

□	Name	Resource Type	Actions
□	ibse_jde_geteffective_BS	Business Service	
□	ibse_jde_geteffective_PS	Proxy Service	

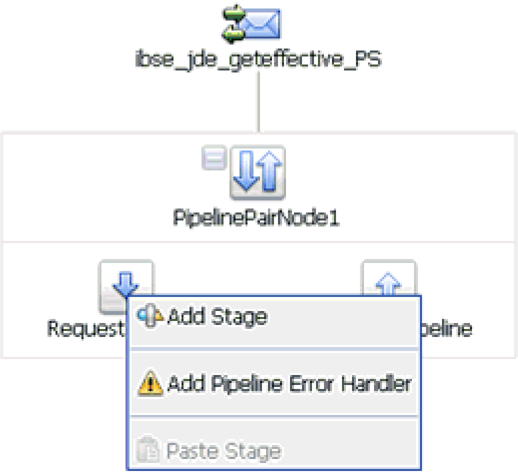
Items 1-2 of 2

The Edit Message Flow workspace area is displayed.

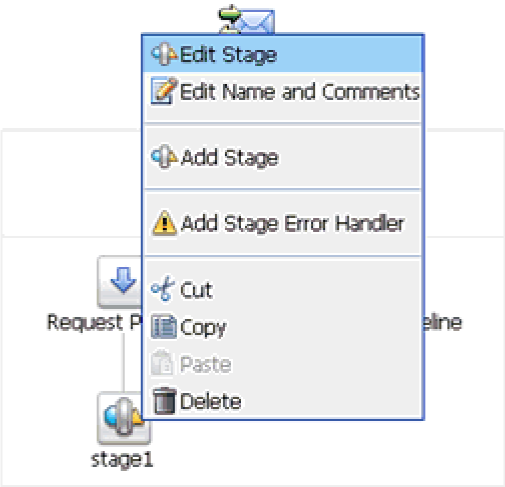
2. Click the **Proxy Service** icon and select **Add Pipeline Pair** from the context menu, as shown in the following image.



3. Click the **Request Pipeline** icon and select **Add Stage** from the context menu.

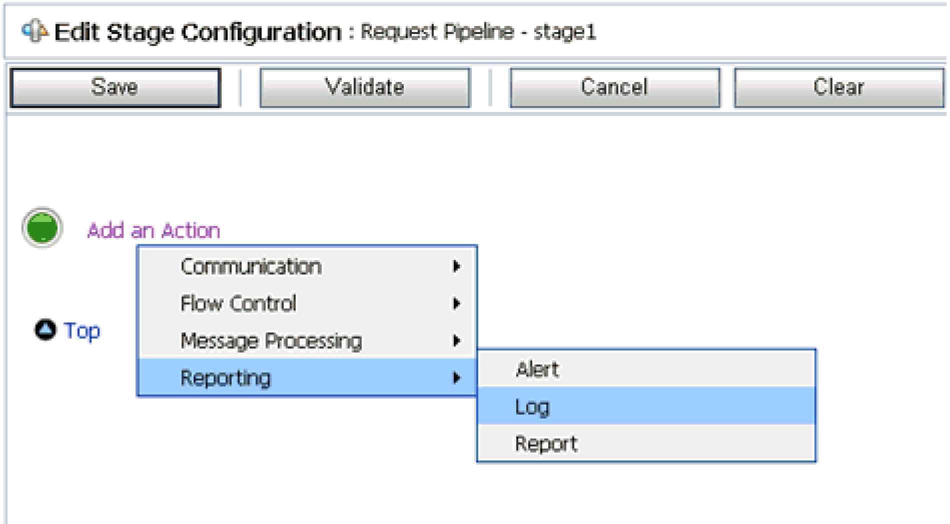


The Stage1 icon is added below the Request Pipeline icon.

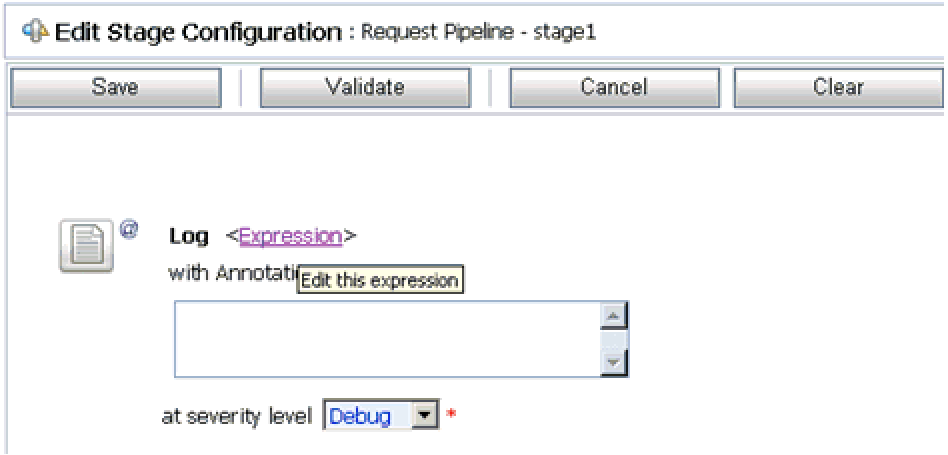


4. Click the **Stage1** icon and select **Edit Stage** from the context menu.

   The Edit Stage Configuration workspace area is displayed.



5. Click **Add an Action**, select **Reporting** from the context menu, and click **Log**.



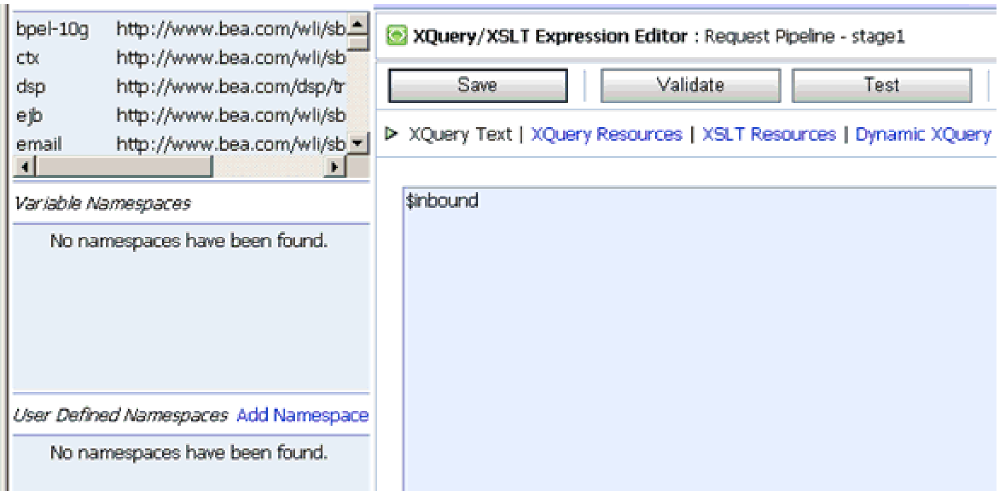
Log <Expression>

with Annotati

at severity level Debug *

6. Click **<Expression>** to edit the expression.

   The XQuery/XSLT Expression Editor is displayed.


XQuery/XSLT Expression Editor : Request Pipeline - stage1
Save
Validate
Test
XQuery Text | XQuery Resources | XSLT Resources | Dynamic XQuery
Variable Namespaces
User Defined Namespaces
Add Namespace

bpel-10g	http://www.bea.com/wli/sb
ctx	http://www.bea.com/wli/sb
dsp	http://www.bea.com/dsp/tr
ejb	http://www.bea.com/wli/sb
email	http://www.bea.com/wli/sb

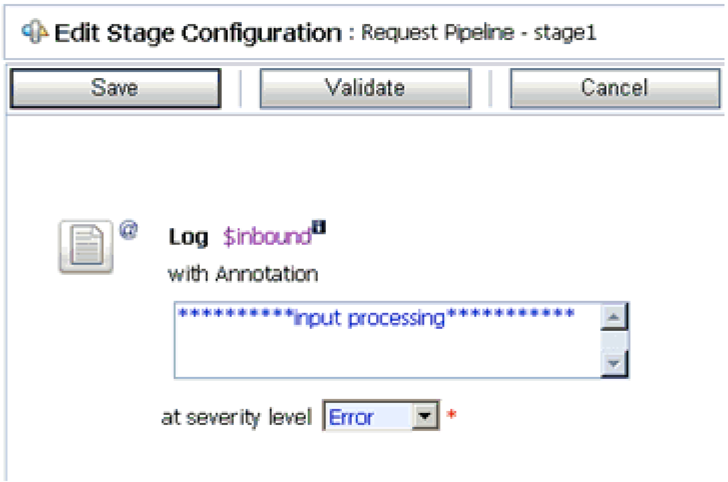
No namespaces have been found.

No namespaces have been found.

$inbound

7. In the XQuery Text area, type **$inbound**.
8. Click **Validate** and then **Save**.

   You are returned to the Edit Stage Configuration workspace area.

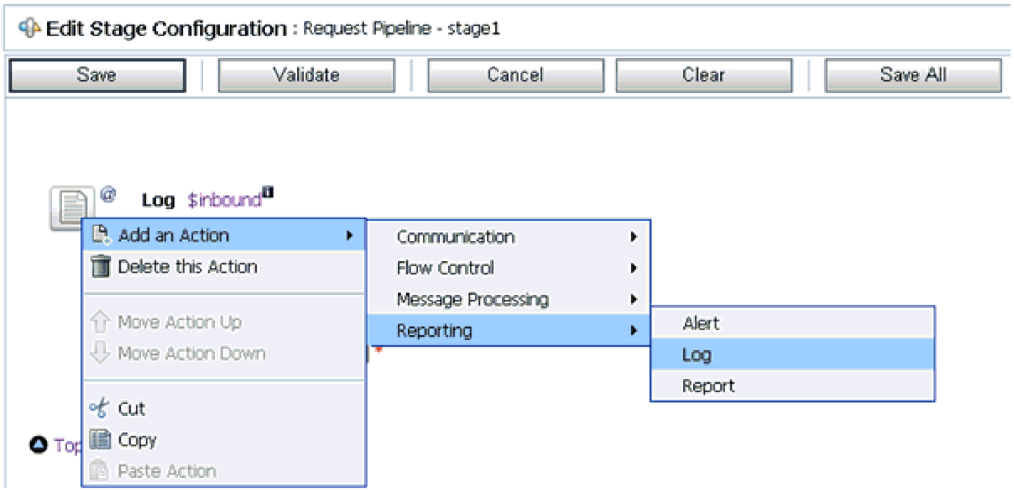


Log $inbound

**********input processing***********

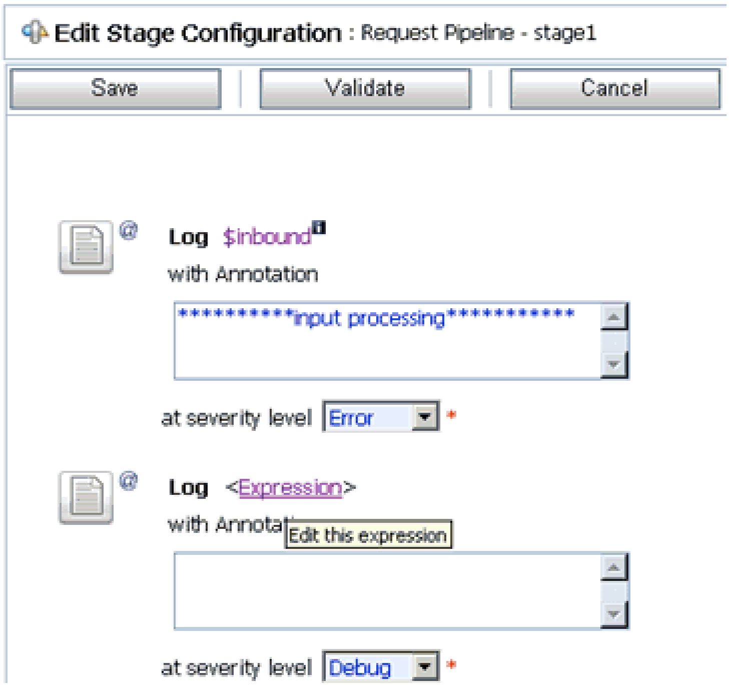
at severity level Error

9. Type any annotation/comments in the text box (for example, **********input processing***********).
10. Select **Error** from the severity level drop-down list.
11. Add one more Log action as shown in the following image.



Log $inbound

A new Log configuration is added, as shown in the following image.

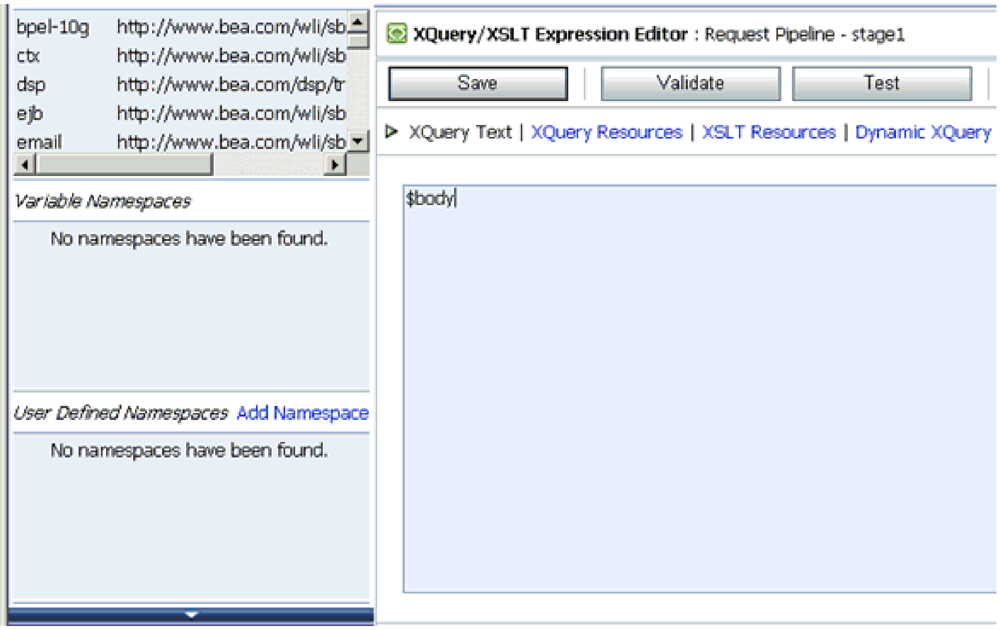


Log $inbound
with Annotation
**********input processing***********
at severity level Error *

Log <Expression>
with Annotation
at severity level Debug *

12. Click **<Expression>** to edit the expression.

The XQuery/XSLT Expression Editor is displayed.



XQuery/XSLT Expression Editor : Request Pipeline - stage1
Save
Validate
Test
XQuery Text | XQuery Resources | XSLT Resources | Dynamic XQuery

bpel-10g	http://www.bea.com/wli/sb
ctx	http://www.bea.com/wli/sb
dsp	http://www.bea.com/dsp/tr
ejb	http://www.bea.com/wli/sb
email	http://www.bea.com/wli/sb

Variable Namespaces

No namespaces have been found.

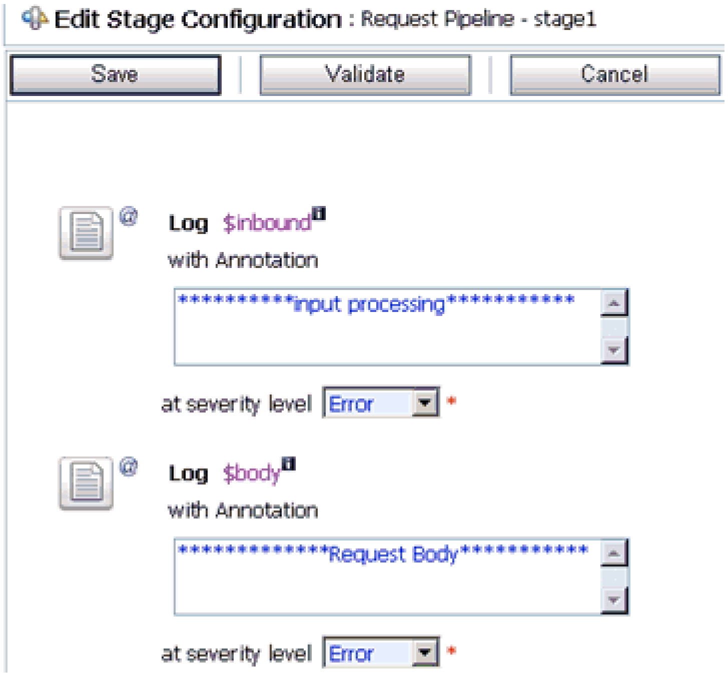
User Defined Namespaces Add Namespace

No namespaces have been found.

$body

13. In the XQuery Text area, type **$body**.
14. Click **Validate** and then **Save**.

    You are returned to the Edit Stage Configuration workspace area.



Log $inbound

with Annotation

**********input processing***********

at severity level Error *

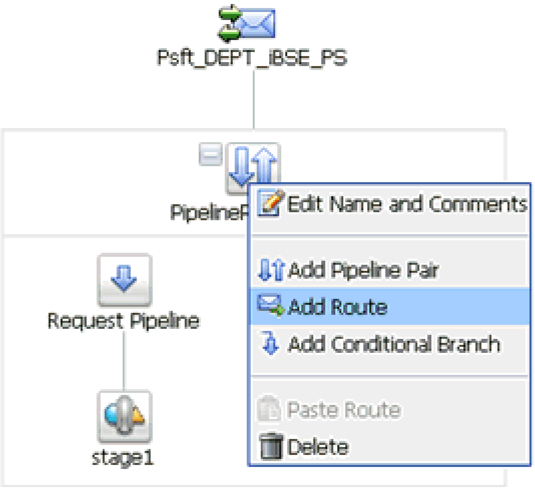
Log $body

with Annotation

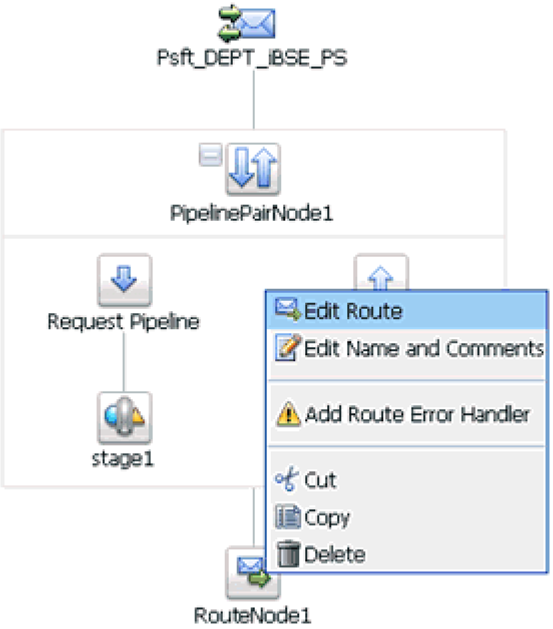
*************Request Body***********

at severity level Error *

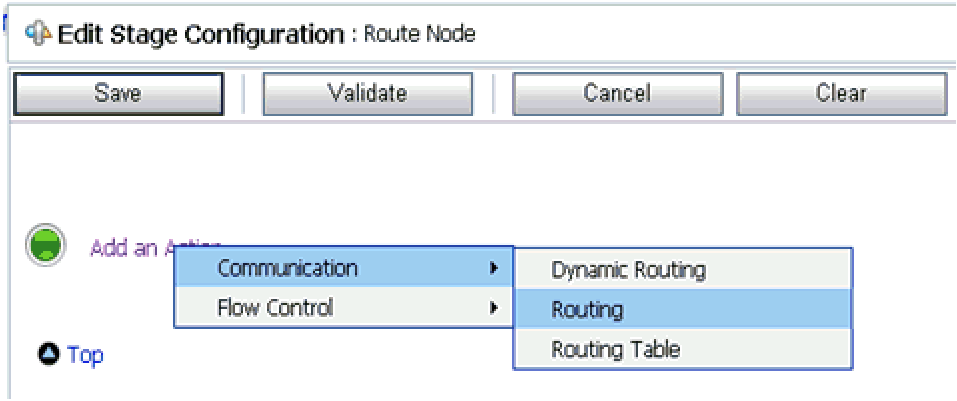
15. Type any annotation/comments in the text box (for example, **********Request Body***********).
16. Select **Error** from the severity level drop-down list.
17. Click **Validate** and then **Save**.



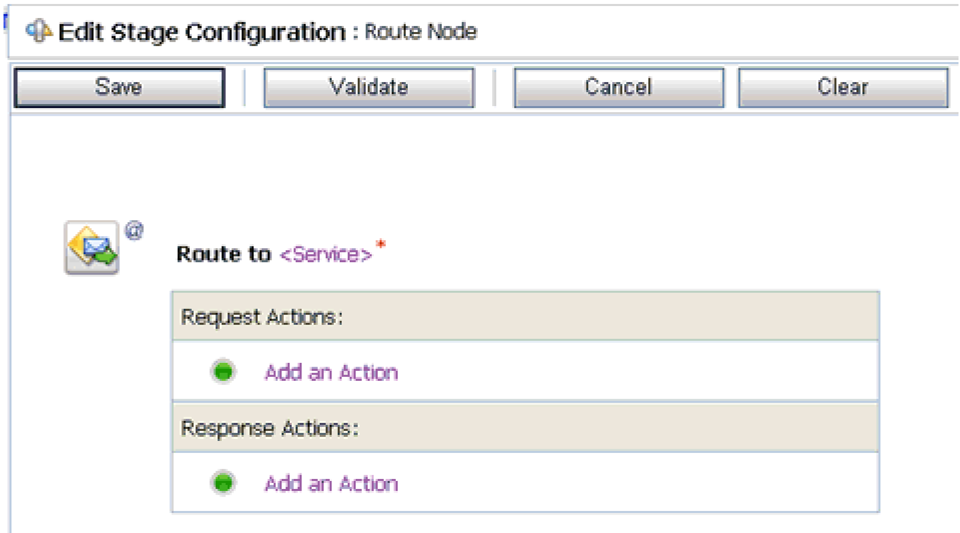
18. Click the **PipelinePairNode1** icon and select **Add Route** from the context menu.



19. Click the **RouteNode1** icon and select **Edit Route** from the context menu.

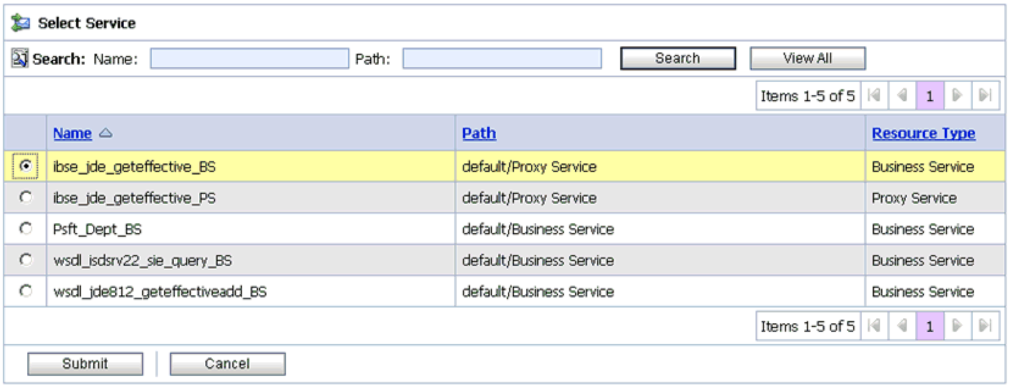


20. Click **Add an Action**, select **Communication** from the context menu, and click **Routing**.



21. Click **<Service>**.

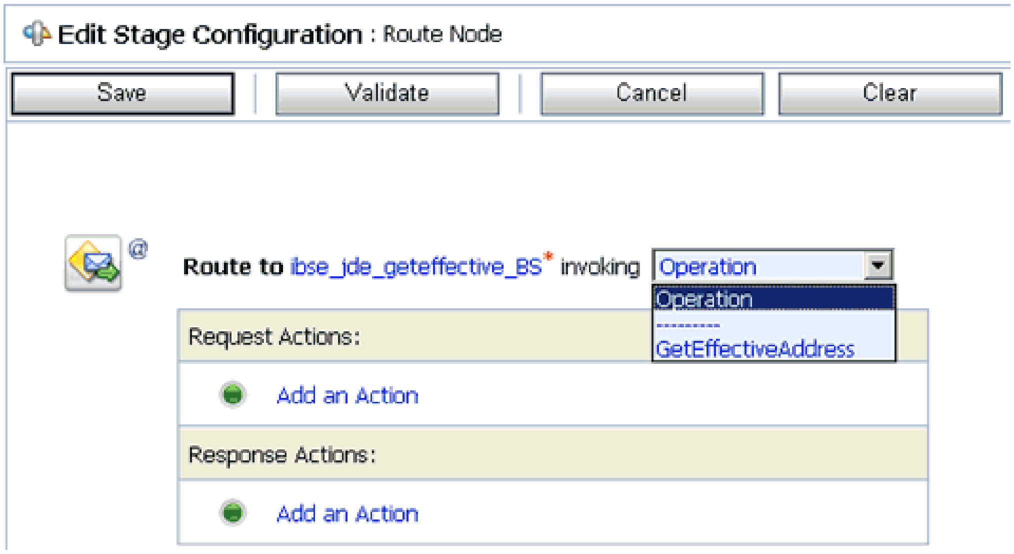
    The Select Service dialog box is displayed.



	Name	Path	Resource Type
●	ibse_jde_geteffective_BS	default/Proxy Service	Business Service
○	ibse_jde_geteffective_PS	default/Proxy Service	Proxy Service
○	Psft_Dept_BS	default/Business Service	Business Service
○	wsdl_isdsrv22_sie_query_BS	default/Business Service	Business Service
○	wsdl_jde812_geteffectiveadd_BS	default/Business Service	Business Service

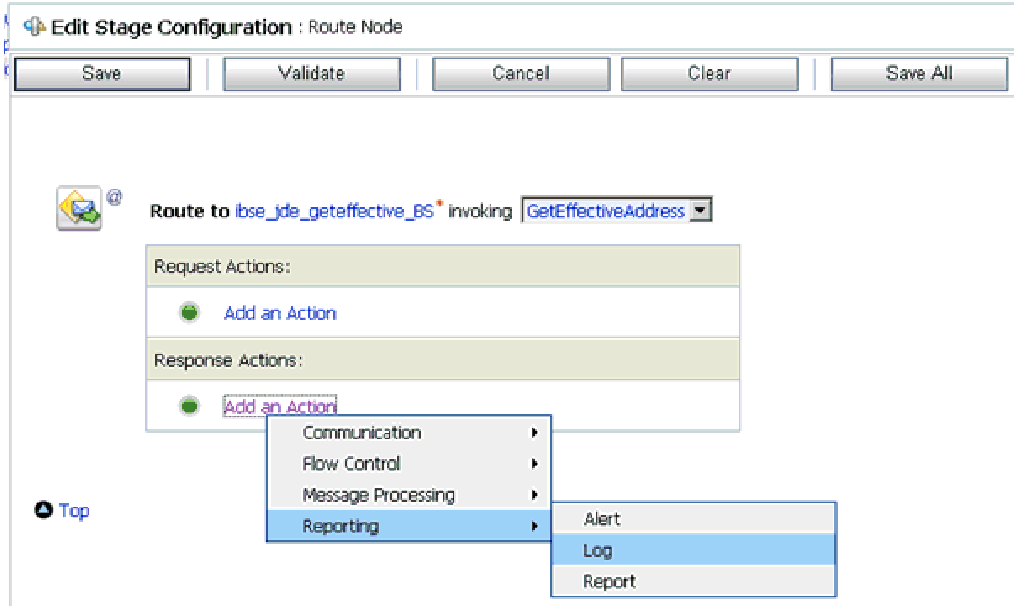
22. Select a WSDL type Business Service and click **Submit**.

    You are returned to the Edit Stage Configuration workspace area.



23. Select **GetEffectiveAddress** as the operational attribute from the drop-down list.

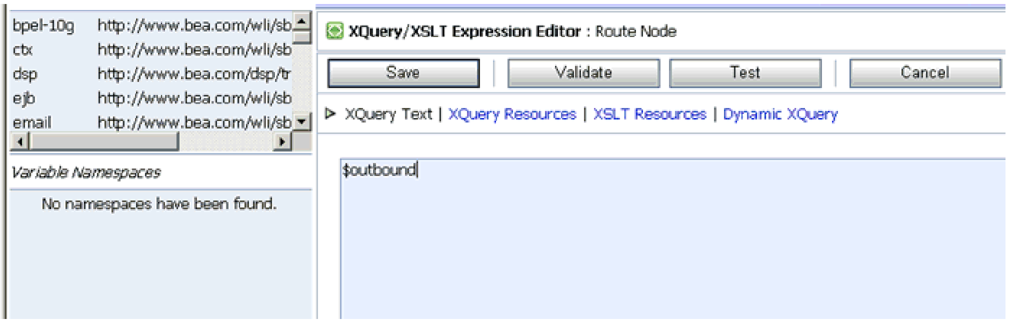
24. Click **Validate** and then **Save**.



25. In the Response Actions area, click **Add an Action**, select **Reporting** from the context menu, and click **Log**.

26. Click **<Expression>** to edit the expression.

    The XQuery/XSLT Expression Editor is displayed.



bpel-10g	http://www.bea.com/wli/sb
ctx	http://www.bea.com/wli/sb
dsp	http://www.bea.com/dsp/tr
ejb	http://www.bea.com/wli/sb
email	http://www.bea.com/wli/sb

Variable Namespaces

No namespaces have been found.

XQuery/XSLT Expression Editor : Route Node

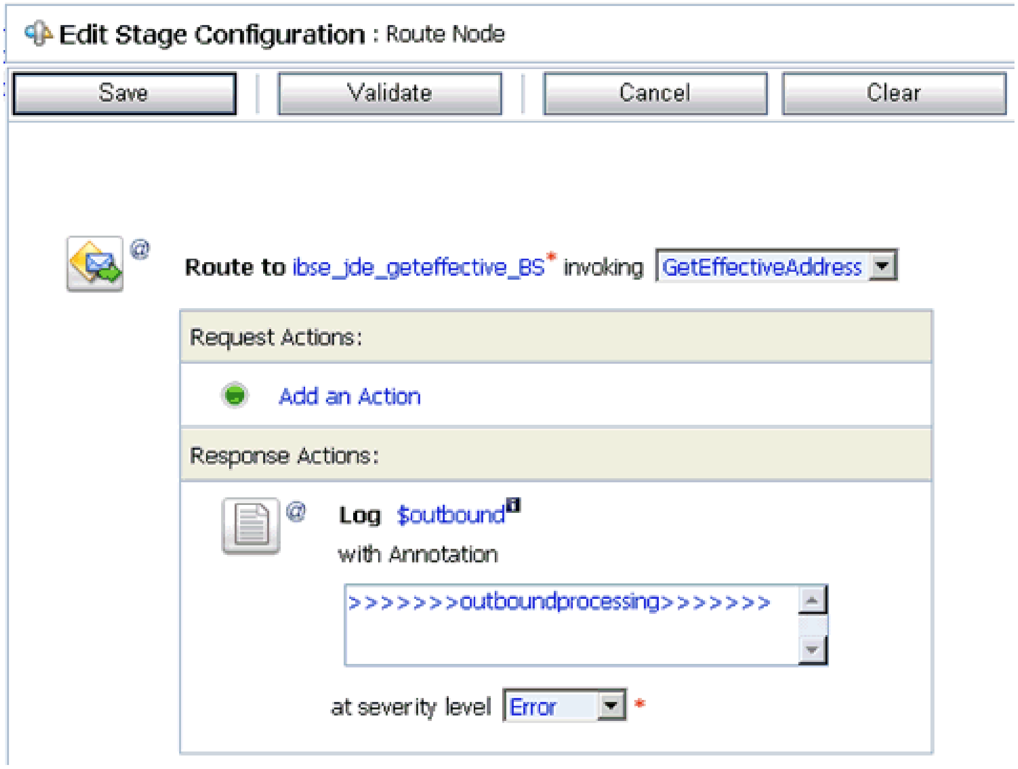
Save
Validate
Test
Cancel
XQuery Text | XQuery Resources | XSLT Resources | Dynamic XQuery

```
$outbound
```

27. In the XQuery Text area, type **$outbound**.

28. Click **Validate** and then **Save**.

    You are returned to the Edit Stage Configuration workspace area.



Edit Stage Configuration : Route Node

Route to ibse_jde_geteffective_BS* invoking GetEffectiveAddress

Request Actions:

Add an Action

Response Actions:

Log $outbound

with Annotation

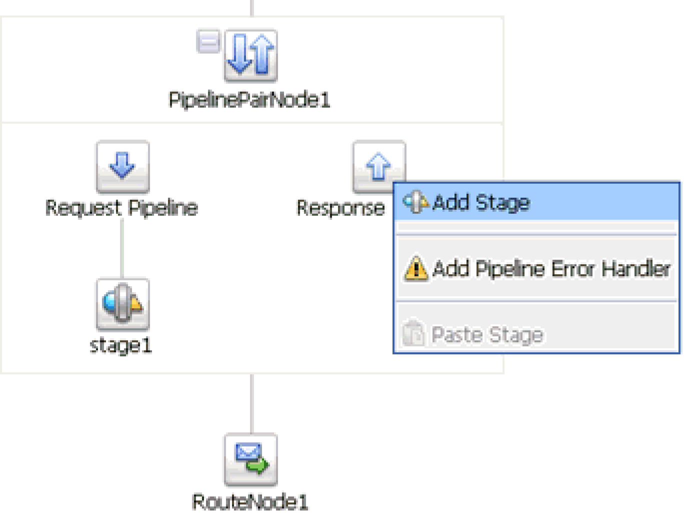
>>>>>>>outboundprocessing>>>>>>

at severity level Error *

29. Type any annotation/comments in the text box (for example, **********output processing***********).

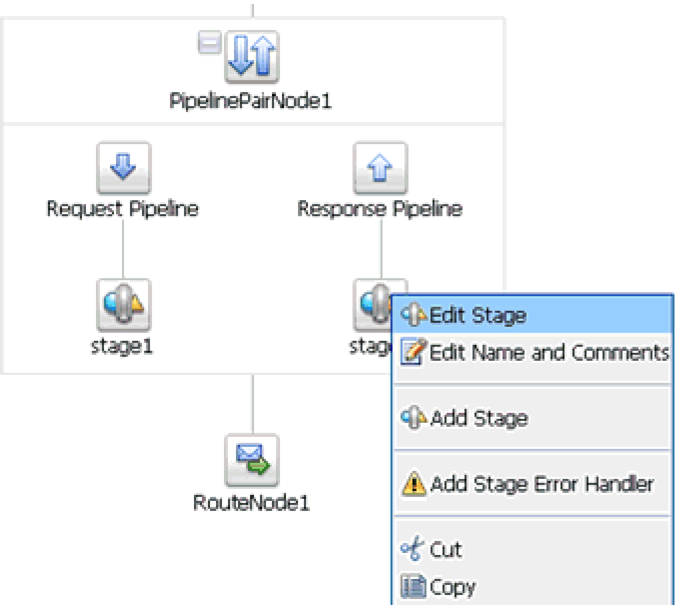
30. Select **Error** from the severity level drop-down list.

31. Click **Validate** and then **Save**.



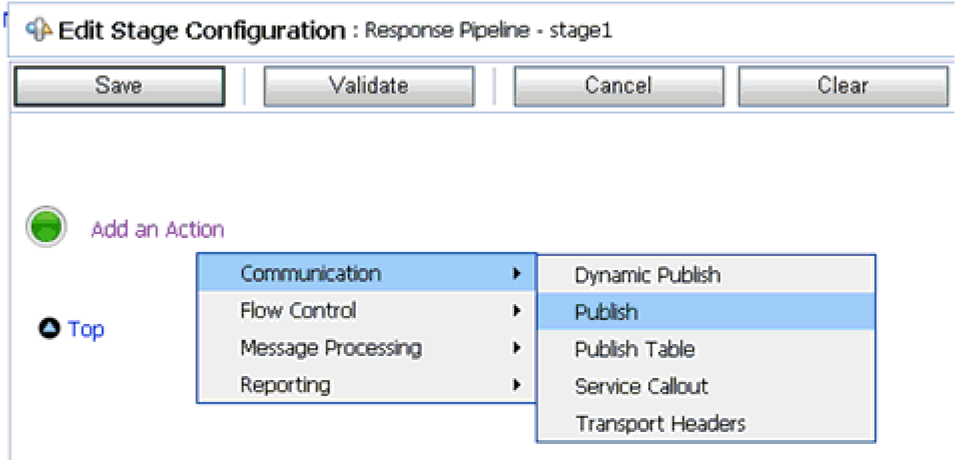
32. Click the **Response Pipeline** icon and select **Add Stage** from the context menu.

    The Stage1 icon is added below the Response Pipeline icon.

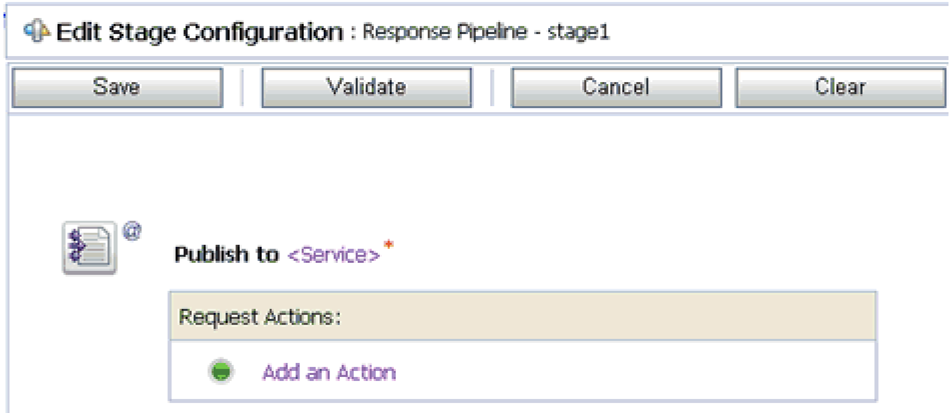


33. Click the **Stage1** icon and select **Edit Stage** from the context menu.

    The Edit Stage Configuration workspace area is displayed.

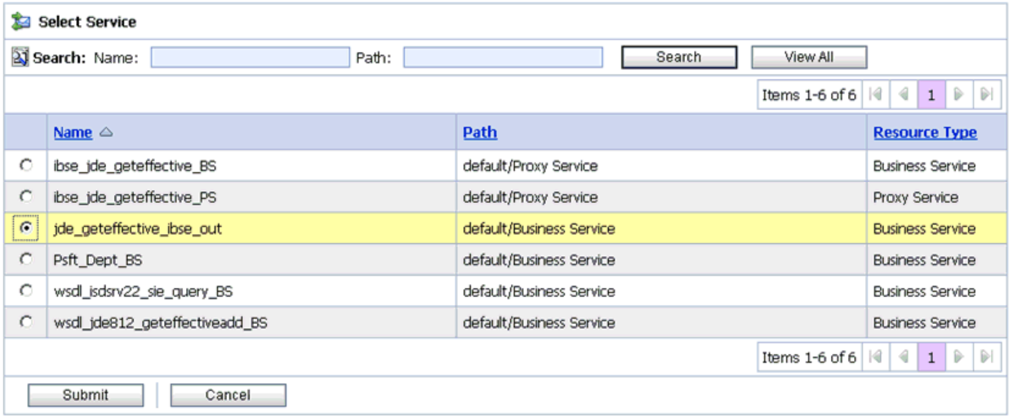


34. Click **Add an Action**, select **Communication** from the context menu, and click **Publish**.



35. Click **<Service>**.

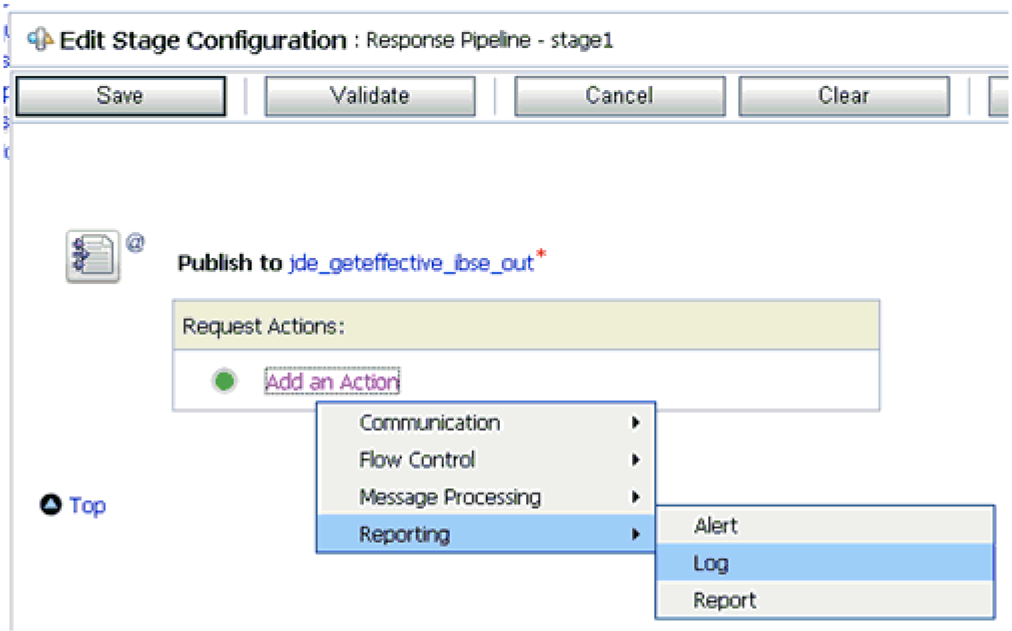
The Select Service dialog box is displayed.



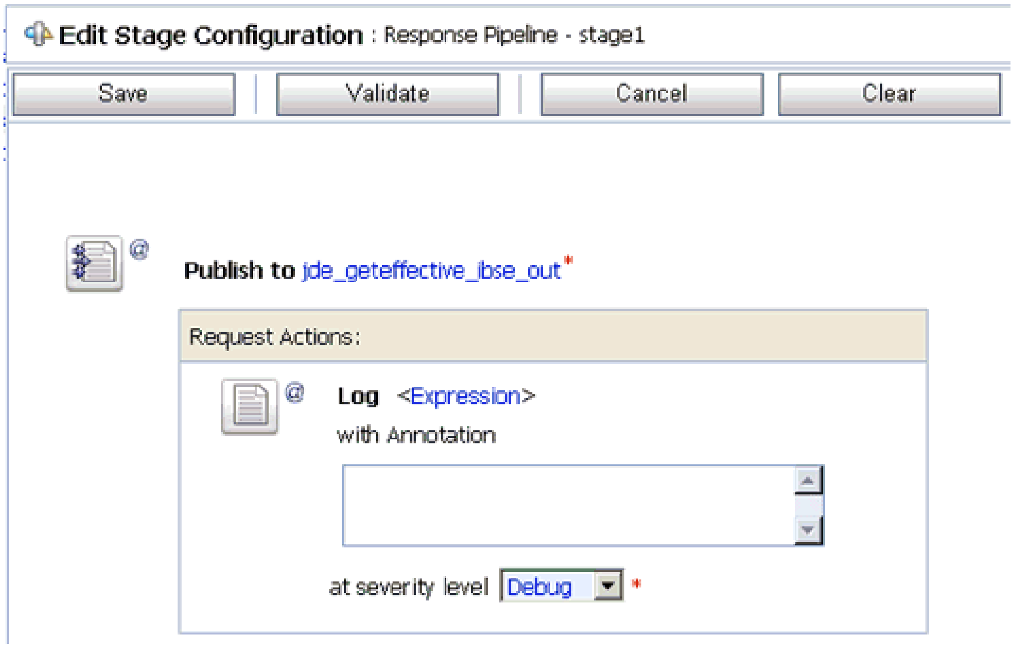
	Name	Path	Resource Type
○	ibse_jde_geteffective_BS	default/Proxy Service	Business Service
○	ibse_jde_geteffective_PS	default/Proxy Service	Proxy Service
●	jde_geteffective_ibse_out	default/Business Service	Business Service
○	Psft_Dept_BS	default/Business Service	Business Service
○	wsdl_jsdsrv22_sie_query_BS	default/Business Service	Business Service
○	wsdl_jde812_geteffectiveadd_BS	default/Business Service	Business Service

36. Select a File type Business Service and click **Submit**.

You are returned to the Edit Stage Configuration workspace area.

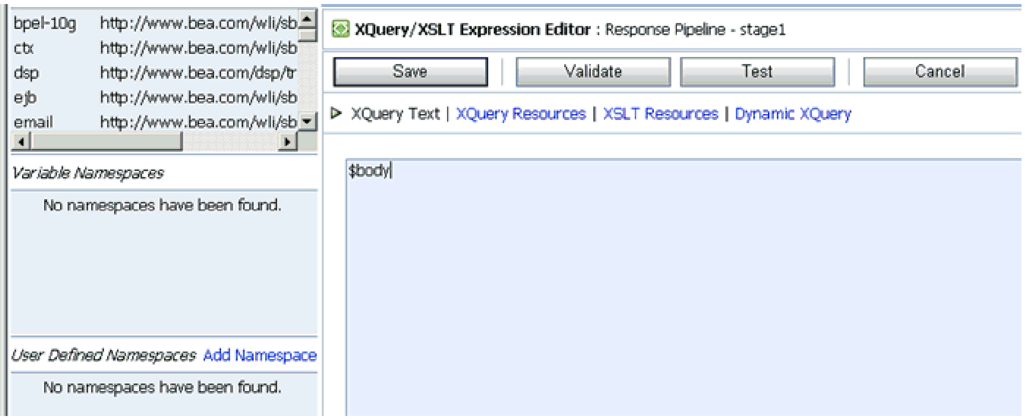


37. In the Request Actions area, click **Add an Action**, select **Reporting** from the context menu, and click **Log**.


Edit Stage Configuration : Response Pipeline - stage1
Save
Validate
Cancel
Clear
Publish to jde_geteffective_ibse_out*
Request Actions:
Log <Expression>
with Annotation
at severity level Debug *

38. Click **<Expression>** to edit the expression.

    The XQuery/XSLT Expression Editor is displayed.



bpel-10g	http://www.bea.com/wli/sb
ctx	http://www.bea.com/wli/sb
dsp	http://www.bea.com/dsp/tr
ejb	http://www.bea.com/wli/sb
email	http://www.bea.com/wli/sb

No namespaces have been found.

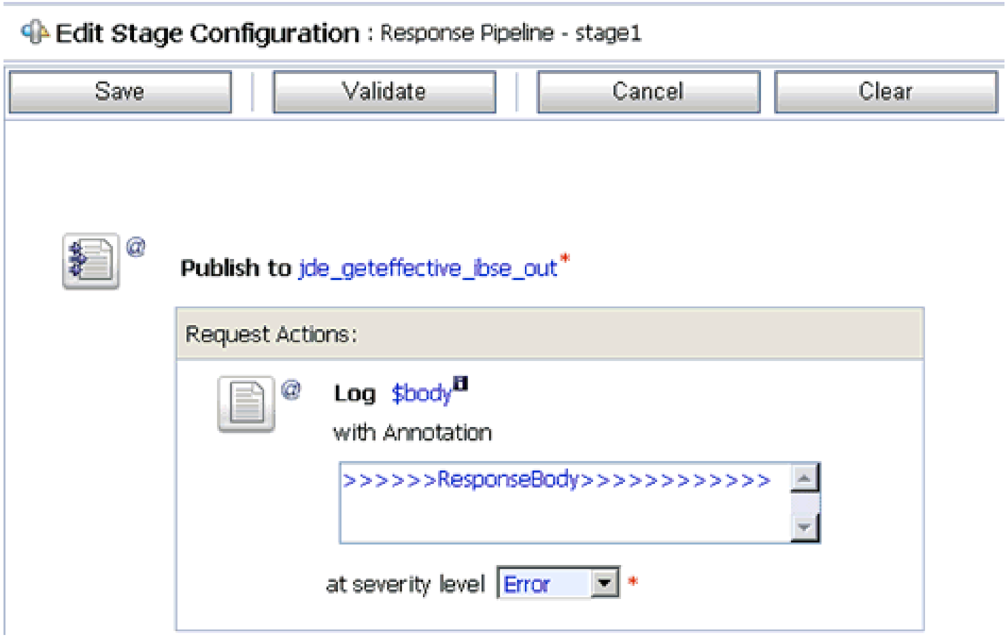
No namespaces have been found.

$body

39. In the XQuery Text area, type **$body**.

40. Click **Validate** and then **Save**.

    You are returned to the Edit Stage Configuration workspace area.



Publish to jde_geteffective_ibse_out *

Log $body

with Annotation

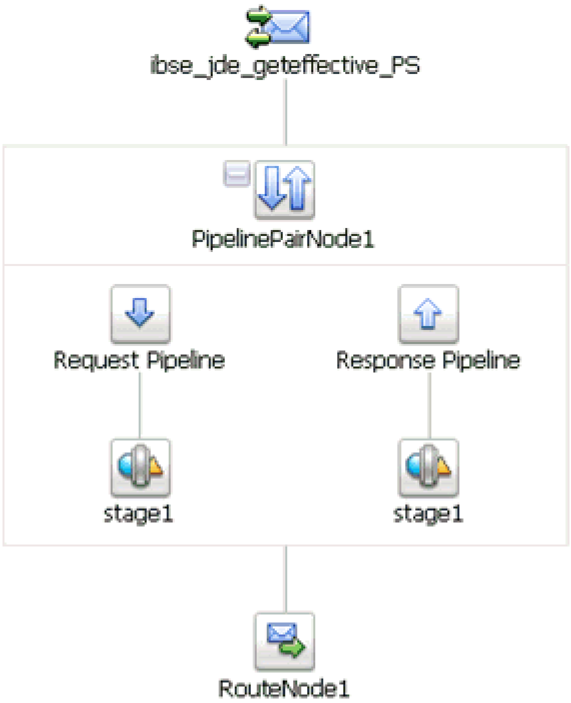
>>>>>>ResponseBody>>>>>>>>>>>>

at severity level Error *

41. Type any annotation/comments in the text box (for example, **********Response Body***********).

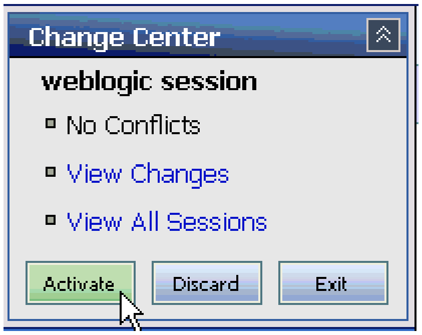
42. Select **Error** from the severity level drop-down list.

43. Click **Validate** and then **Save**.

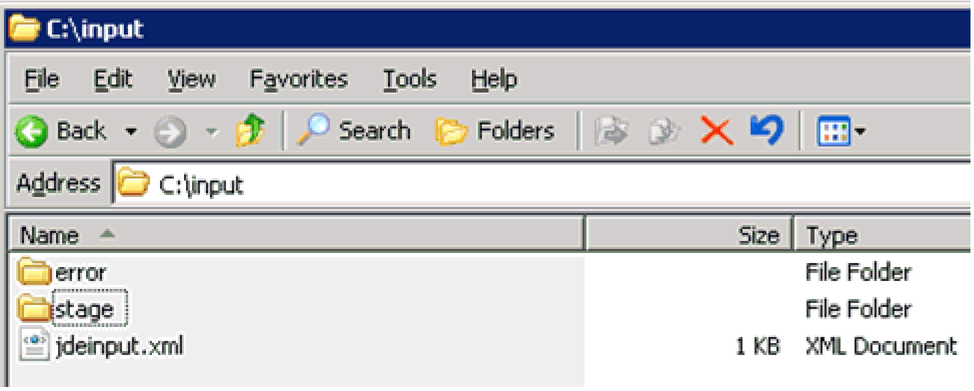


44. Click **Save**.

45. Click **Activate** in the Change Center area to activate your changes in the Oracle Service Bus session.

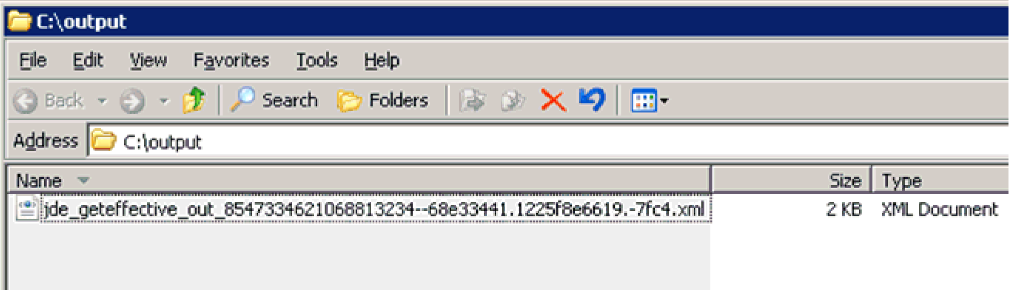


46. Copy and paste an input XML file in the input folder you have configured.



Name	Size	Type
error		File Folder
stage		File Folder
jdeinput.xml	1 KB	XML Document

An output XML file is received in the destination folder.



jde_geteffective_out_8547334621068813234--68e33441.1225f8e6619.-7fc4.xml 2 KB XML Document

5

Configuring Inbound and Outbound Processing Using Oracle Service Bus (J2CA Configuration)

OracleWLS Application Adapter for J.D. Edwards OneWorld integrates seamlessly with Oracle Service Bus (OSB) to facilitate Web service integration. OSB is based on the Service-Oriented Architecture (SOA). It consumes adapter services exposed as Web Service Definition Language (WSDL) documents.

This chapter includes the following topics:

- Overview of Application Adapter Integration with Oracle Service Bus
- Configuring Inbound Processing Using Oracle Service Bus (J2CA Configuration)
- Configuring Outbound Processing Using Oracle Service Bus (J2CA Configuration)

Overview of Application Adapter Integration with Oracle Service Bus

To integrate with Oracle Service Bus (OSB), OracleWLS Application Adapter for J.D. Edwards OneWorld must be deployed on the same Oracle WebLogic Server as OSB. The underlying adapter services must be exposed as WSDL files, which are generated during design time in Oracle WebLogic Server Adapter Application Explorer (Application Explorer) for both request-response (outbound) and event notification (inbound) services of the adapter.

Configuring Inbound Processing Using Oracle Service Bus (J2CA Configuration)

This section describes how to configure inbound processing using Oracle Service Bus (J2CA Configuration).

This section includes the following topics:

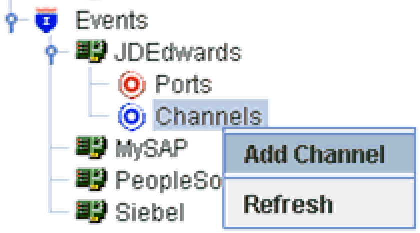
- Creating a Channel
- Generating a WSDL File for Event Notification
- Starting Oracle Service Bus and Creating Project Folders
- Importing a WSDL File From the File System to Oracle Service Bus
- Configuring a File Type Business Service
- Configuring a Proxy Service
- Configuring a Pipeline

Creating a Channel

This section describes how to create a channel using Application Explorer.

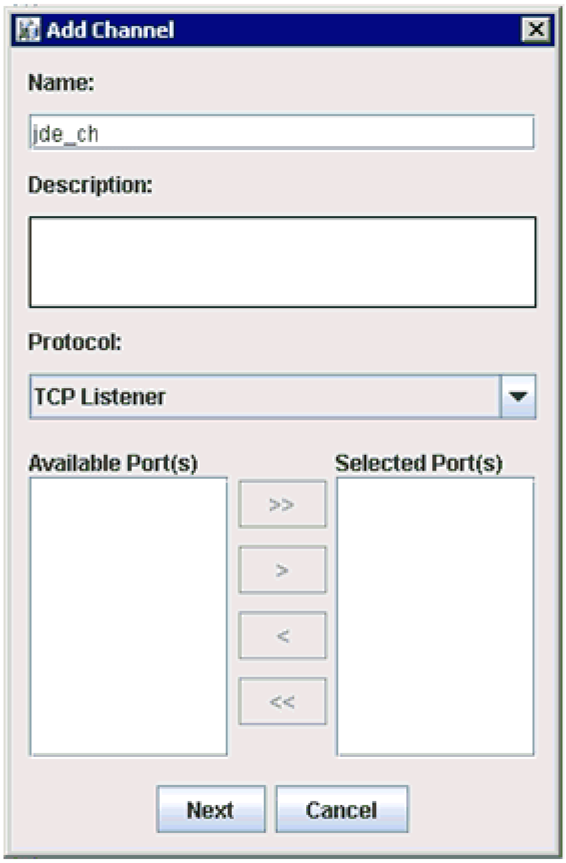
1. Start Application Explorer, connect to a J2CA configuration, and connect to a J.D. Edwards OneWorld target.

   For more information, see Chapter 2, "Configuring OracleWLS Application Adapter for J.D. Edwards One World".



2. Right-click the **Channels** node for J.D. Edwards OneWorld and select **Add Channel** from the context menu.

   The Add Channel dialog box is displayed.

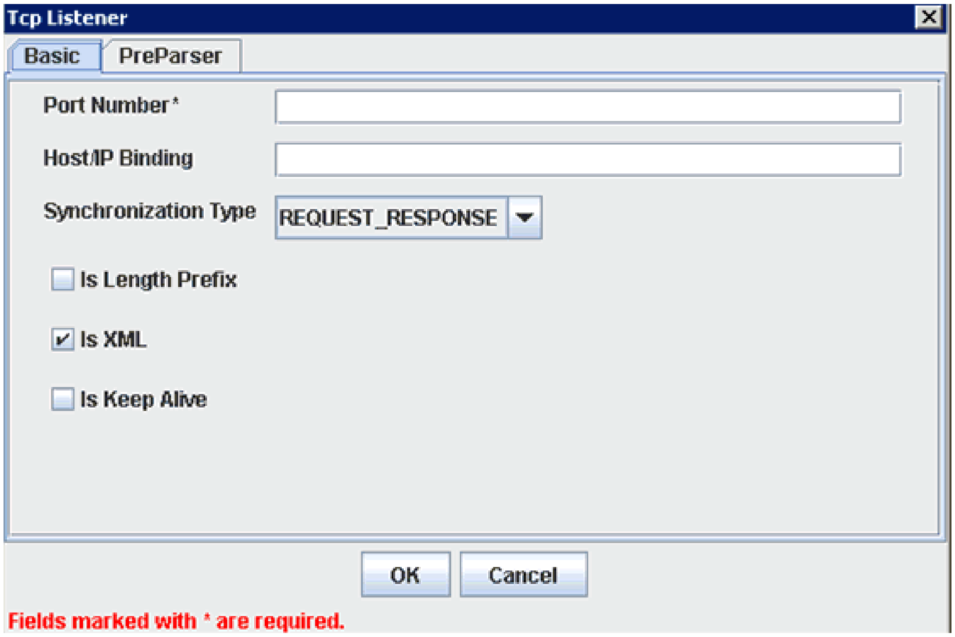


Name: jde_ch

Protocol: TCP Listener

3. Enter a name for the new J.D. Edwards OneWorld channel, select **TCP Listener** from the Protocol drop-down list, and click **Next**.

   The Tcp Listener dialog box is displayed.



Port Number*

Host/IP Binding

Synchronization Type: REQUEST_RESPONSE

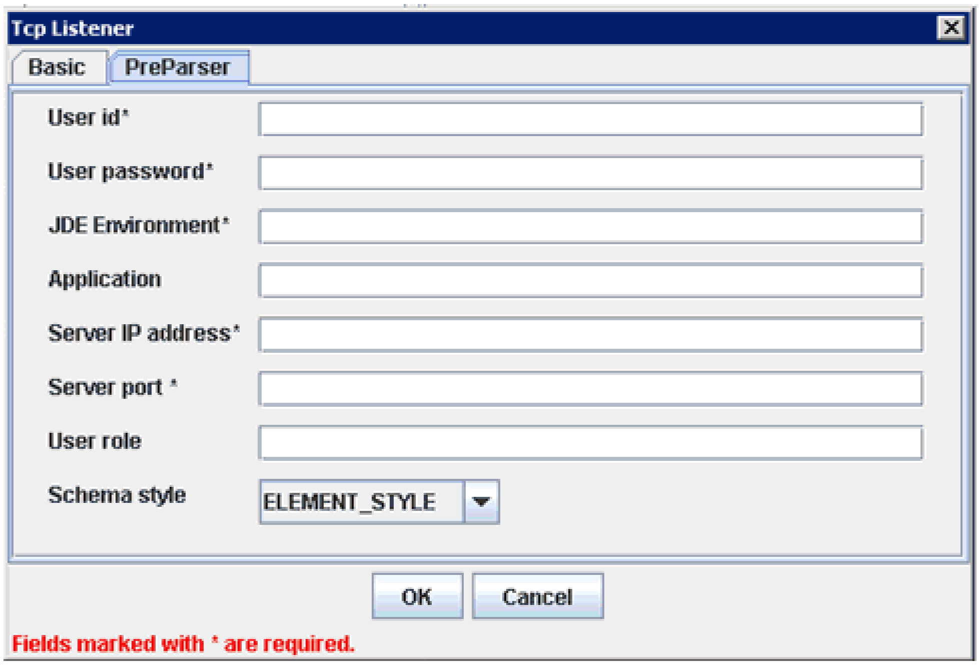
□ Is Length Prefix

☑ Is XML

□ Is Keep Alive

Fields marked with * are required.

4. In the Port Number field, enter the port number of the channel.
5. In the Host/IP Binding field, enter the location of the server or leave this field empty.
6. Select **REQUSEST_RESPONSE** from the Synchronization Type drop-down list.
7. Click the PreParser tab.



User id*

User password*

JDE Environment*

Application

Server IP address*

Server port *

User role

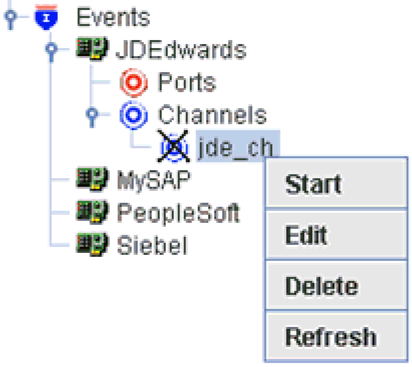
Schema style: ELEMENT_STYLE

Fields marked with * are required.

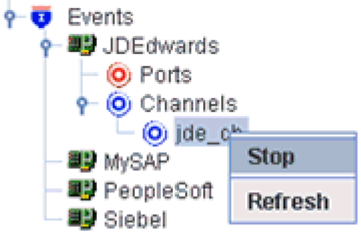
8. Provide the required connection information for your J.D. Edwards OneWorld system.

   For more information, see Chapter 2, "Configuring OracleWLS Application Adapter for J.D. Edwards One World".

9. Click **OK**.



10. Right-click the new channel that you just created and select **Start** from the context menu.



11. Right-click the started channel and select **Stop** from the context menu.

Generating a WSDL File for Event Notification

You cannot generate a WSDL for J.D. Edwards OneWorld event notification using Application Explorer. To generate a WSDL from the command prompt, you must perform the steps that are described in the following section.

You can create an inbound J2CA service only if the node that you selected supports events.

1. Create a channel using Application Explorer under the J.D. Edwards Events node.
2. Start the channel.

   Do not restart Oracle WebLogic Server after the channel is started.
3. Send an inbound message from J.D. Edwards OneWorld.
4. Capture the inbound message payload in the log file, which is located in the following directory:

   ```
   wls_home\erp-adapters\application\config\jca_sample\log\iwaf_jca15.log
   ```

   Alternatively, you can create a port using the File protocol under the Events node in Application Explorer, which will dispose the event message to the file system.
5. Use a third party tool (for example, XMLSpy) to create the XML schema (.xsd file) using the XML payload that was captured in the previous step.
6. Verify that **orabpel-adapters.jar** is in your class path
7. Open a command prompt and navigate to the following directory:

   ```
   wls_home\adapters\application\tools
   ```

8. Execute the **obadapter.bat** file to set the environment.

9. Enter the following command to generate a WSDL:

   ```
   java com.iwaysoftware.af.container.tools.wsdl.IWayWSILBrowser jcaHome adapter
   target channel schemaPrefix wsdlFileName
   ```

   where:

   jcaHome is the path to your J2CA configuration. For example:

   ```
   wls_home\erp-adapters\application\
   ```

   adapter is the name of the adapter. For example, JDEdwards.

   target is the name of the adapter target you created in Application Explorer.

   channel is the name of the channel you created in Application Explorer.

   schemaPrefix is the prefix for the XSD schema. The schema file must be in the same directory where the Java command is executed, for example:

   ```
   java com.iwaysoftware.af.container.tools.wsdl.IWayWSILBrowser
   c:\bea\erp-adapters\ JDEdwards jde812_tgt jde_ch Jde812_Schema Jde812_
   salesorder_receive_osb.wsdl
   ```

   Once the command is executed, the following is displayed in the command window:

   ```
   Running Inbound WSDL generation tool...
   -> user.dir = java com.iwaysoftware.af.container.tools.wsdl.IWayWSILBrowser
   c:\bea\erp-adapters\ JDEdwards jde812_tgt jde_ch Jde812_Schema Jde812_
   salesorder_receive_osb.wsdl

   -> Generating WSDL...
   -> Done.
   -> Writing WSDL 'c:\bea\erp-adapters\application\\wsdls\ Jde812_salesorder_
   receive_osb.wsdl ' to disk...
   -> Done.
   ```

   Note: It is good practice to append **_receive** to the names of WSDL files that are generated for event notification services. This will allow you to easily distinguish between them and those generated for request-response services.

10. Stop the channel in Application Explorer.

Starting Oracle Service Bus and Creating Project Folders

This section describes how to access the Oracle Service Bus Console and create project folders.

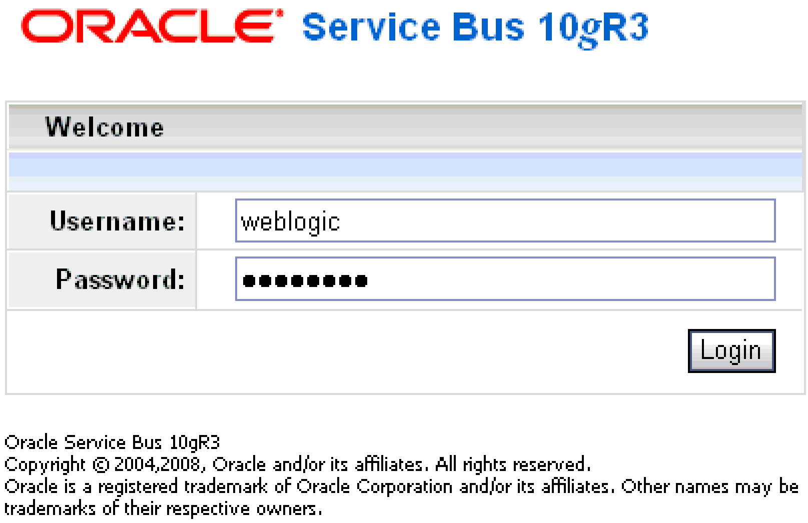
1. Start the Oracle WebLogic Server for the Oracle WebLogic Server domain that you have configured.

2. Open the Oracle Service Bus Console in a Web browser by entering the following URL:

   ```
   http://hostname:port/sbconsole
   ```

Where *hostname* is the name of the machine where Oracle WebLogic Server is running and *port* is the port for the domain you are using. The port for the default domain is 7001.

The Oracle Service Bus Console logon page is displayed.



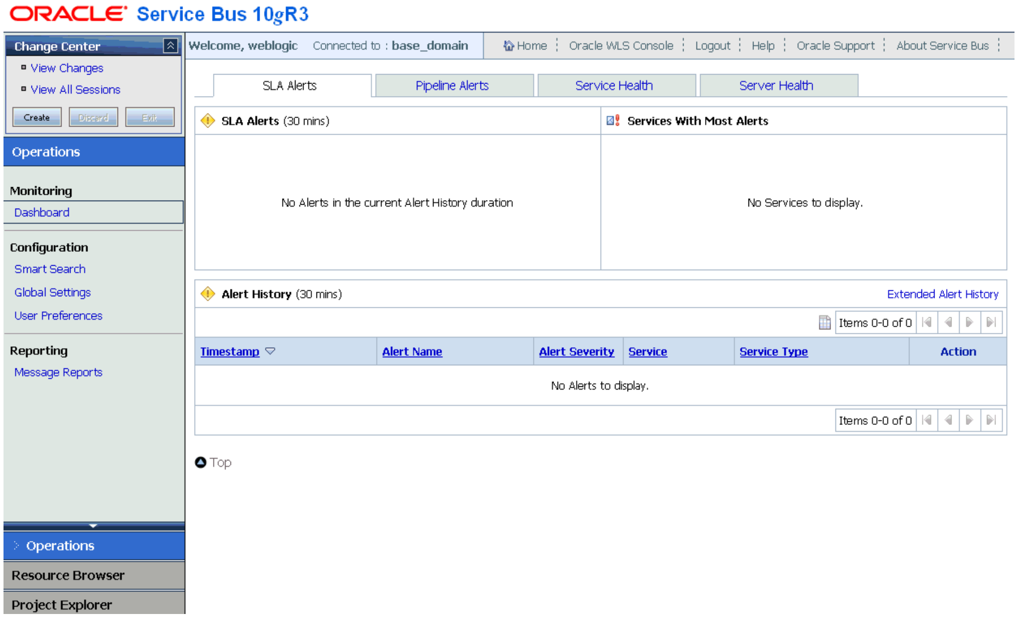
Username: weblogic

Password: ●●●●●●●●●

Oracle Service Bus 10gR3
Copyright © 2004,2008, Oracle and/or its affiliates. All rights reserved.
Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

3. Log on to the Oracle Service Bus Console using a valid user name and password.

   The Oracle Service Bus Console home page is displayed.



Welcome, weblogic Connected to : base_domain

SLA Alerts (30 mins)

No Alerts in the current Alert History duration

Services With Most Alerts

No Services to display.

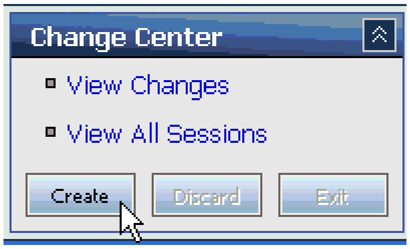
Alert History (30 mins)

Items 0-0 of 0

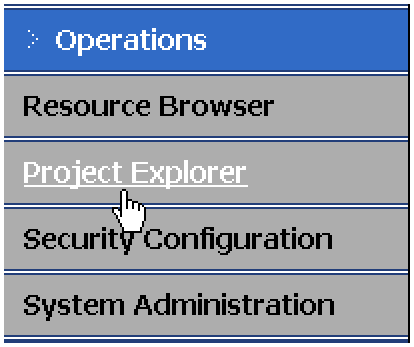
Timestamp	Alert Name	Alert Severity	Service	Service Type	Action
No Alerts to display.					

Items 0-0 of 0

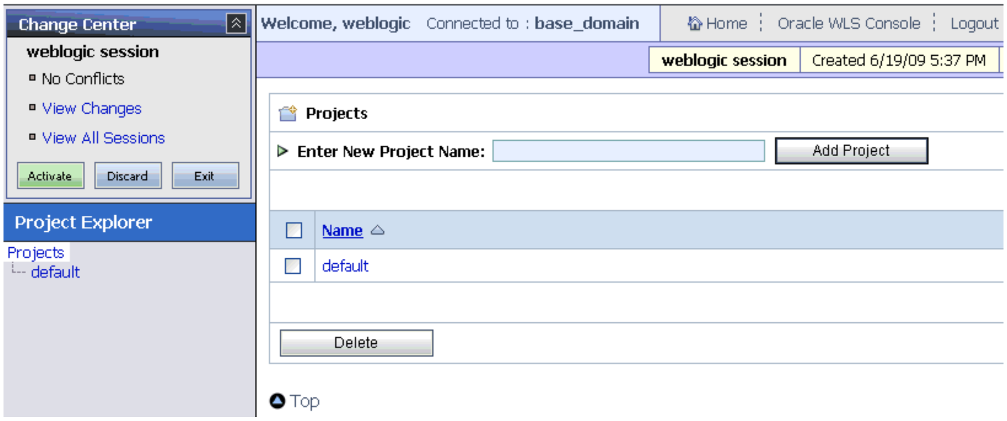
4. Click **Create** in the Change Center area to start a new Oracle Service Bus session.



5. Click **Project Explorer** in the left pane.

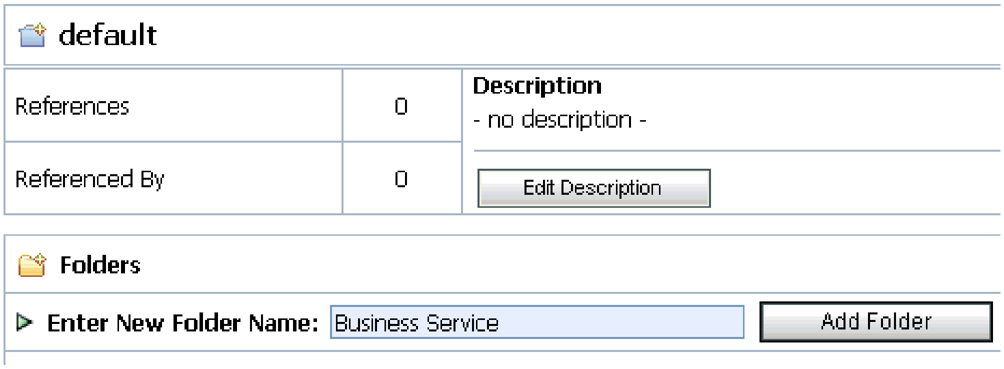


The Project Explorer page is displayed.



6. Click the **default** project node in the left pane.

The default project page is displayed.

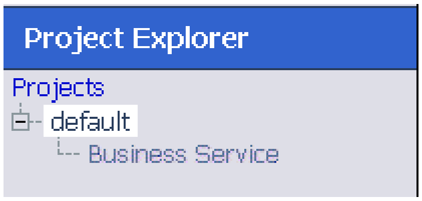


References	0	
Referenced By	0	

Enter New Folder Name: Business Service

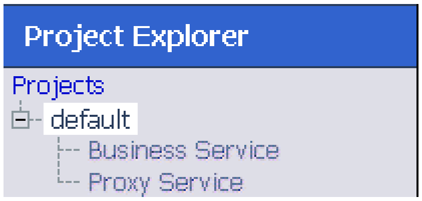
7. In the Enter New Folder Name field, type **Business Service** and click **Add Folder**.

The Business Service folder is listed in the left pane below the default project node.



8. In the Enter New Folder Name field, type **Proxy Service** and click **Add Folder**.

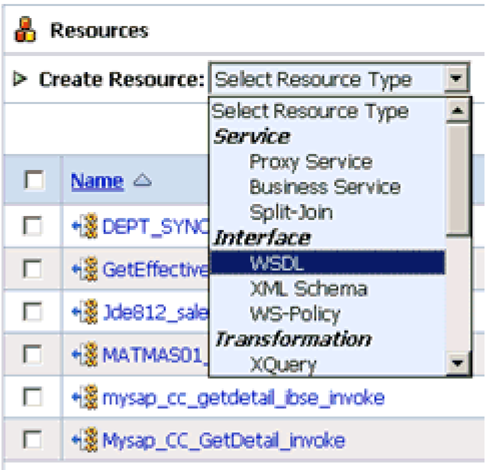
   The Business Service and Proxy Service folders are listed in the left pane below the default project node.



Importing a WSDL File From the File System to Oracle Service Bus

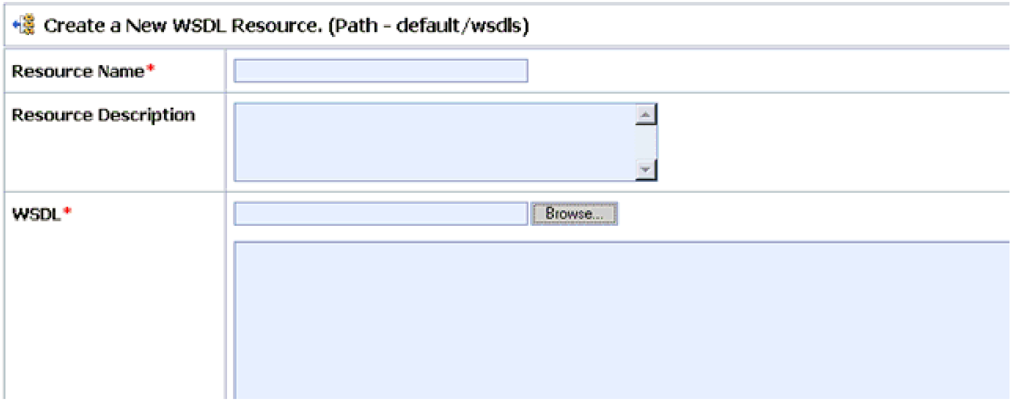
This section describes how to import a WSDL file from the file system to Oracle Service Bus.

1. Create a new project folder called **wsdls** and select this folder.



2. From the Create Resource drop-down list, select **WSDL**.

   The Create a New WSDL Resource page is displayed.



Create a New WSDL Resource. (Path - default/wsdls)

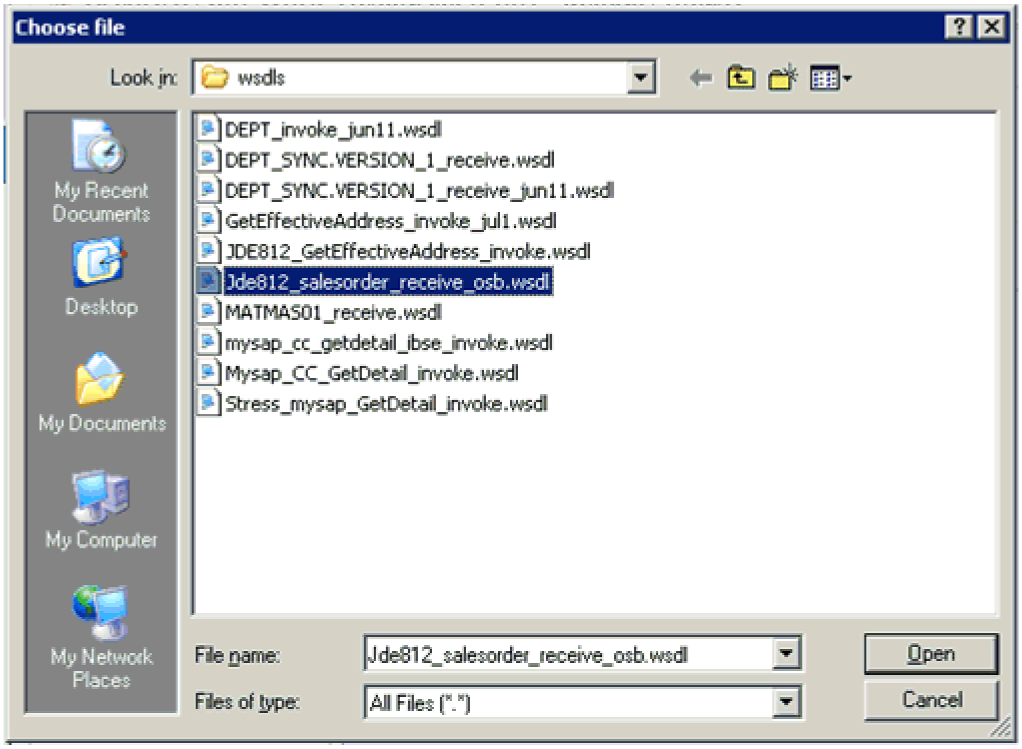
Resource Name*

Resource Description

WSDL*

3. Click **Browse**.

   The Choose File dialog box is displayed.



Choose file

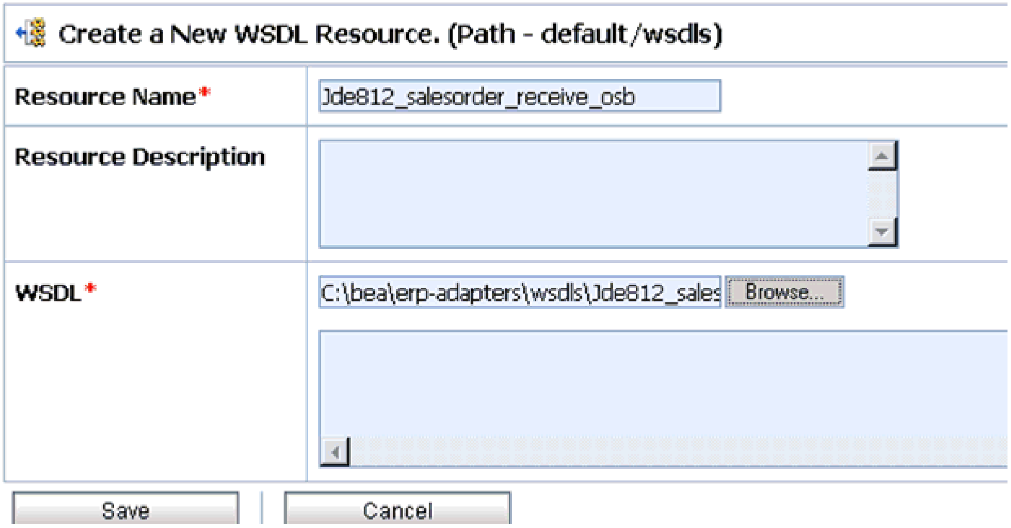
Look in: wsdls

DEPT_invoke_jun11.wsdl
DEPT_SYNC.VERSION_1_receive.wsdl
DEPT_SYNC.VERSION_1_receive_jun11.wsdl
GetEffectiveAddress_invoke_jul1.wsdl
JDE812_GetEffectiveAddress_invoke.wsdl
Jde812_salesorder_receive_osb.wsdl
MATMAS01_receive.wsdl
mysap_cc_getdetail_ibse_invoke.wsdl
Mysap_CC_GetDetail_invoke.wsdl
Stress_mysap_GetDetail_invoke.wsdl

File name: Jde812_salesorder_receive_osb.wsdl

Files of type: All Files (*.*)

4. Select a generated WSDL file and click **Open**.



Create a New WSDL Resource. (Path - default/wsdls)

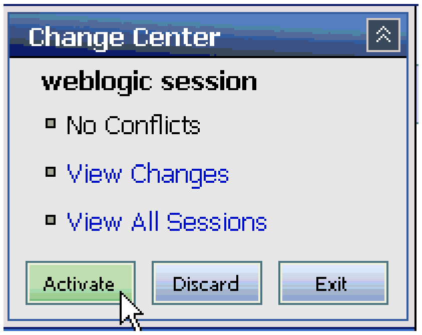
Resource Name* Jde812_salesorder_receive_osb

Resource Description

WSDL* C:\bea\erp-adapters\wsdls\Jde812_sales

5. Click **Save**.

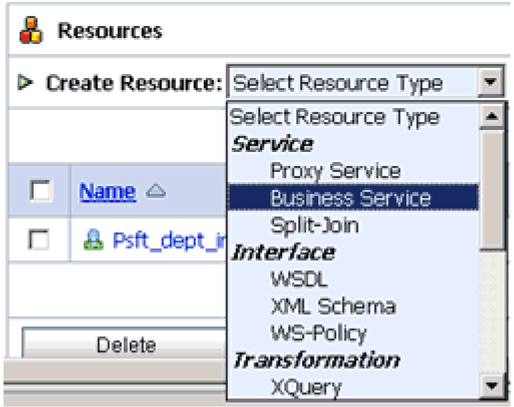
6. Click **Activate** in the Change Center area to activate your changes in the Oracle Service Bus session.



Configuring a File Type Business Service

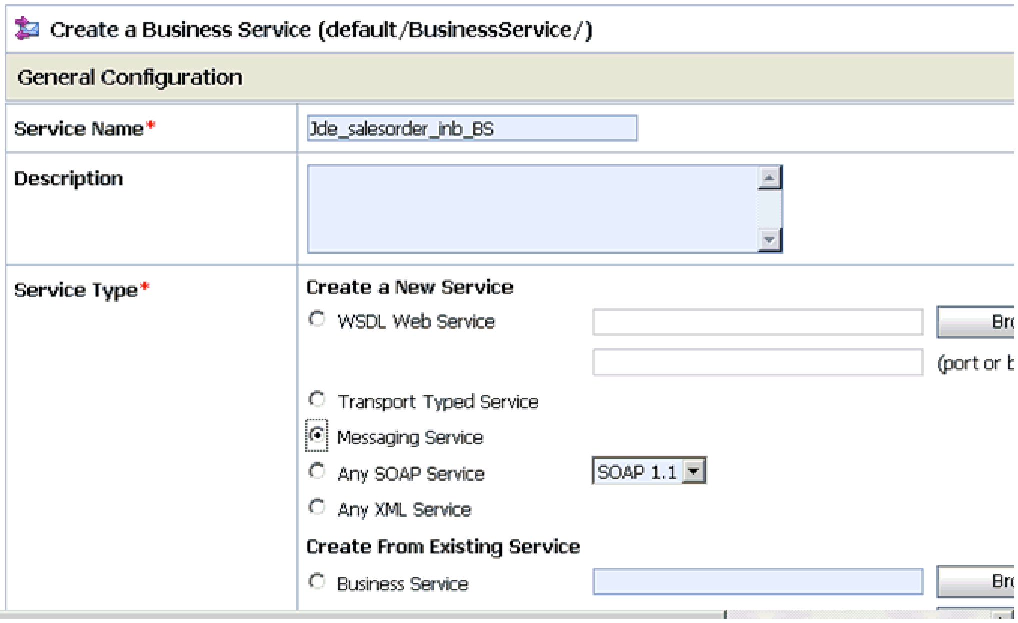
This section describes how to configure a File type Business Service using the Oracle Service Bus Console.

1. Select the **Business Service** folder you created in the left pane.



2. In the right pane, select **Business Service** from the Create Resource menu.

The Create a Business Service - General Configuration page is displayed.



Create a Business Service (default/BusinessService/)

General Configuration

Service Name*: Jde_salesorder_inb_BS

Description:

Service Type*:

Create a New Service

○ WSDL Web Service

(port or b

○ Transport Typed Service

● Messaging Service

○ Any SOAP Service — SOAP 1.1

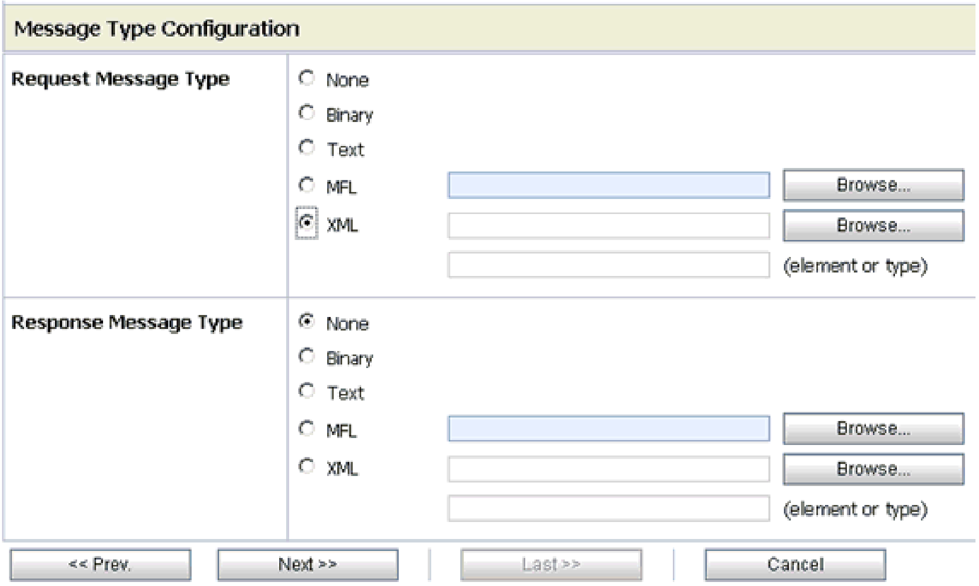
○ Any XML Service

Create From Existing Service

○ Business Service

3. Provide a name for the Business Service and from the Service Type area select **Messaging Service**.

4. Click **Next**.



Message Type Configuration

Request Message Type:

○ None

○ Binary

○ Text

○ MFL

● XML

(element or type)

Response Message Type:

● None

○ Binary

○ Text

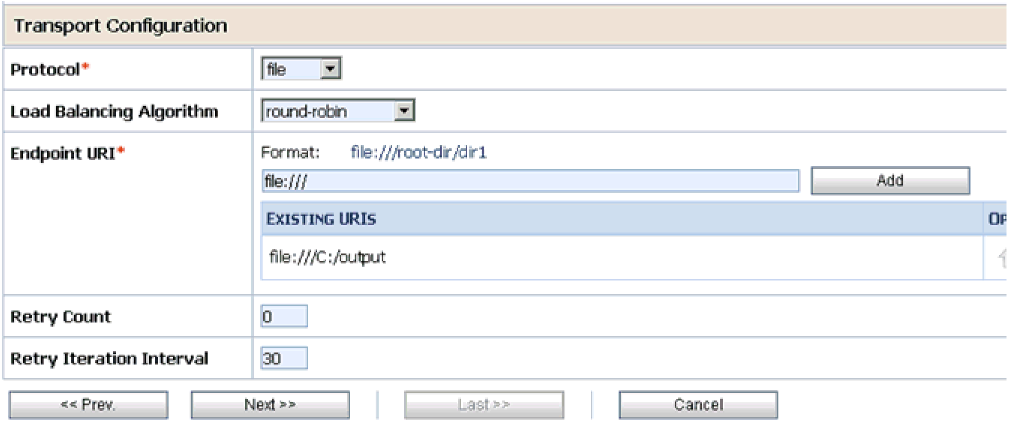
○ MFL

○ XML

(element or type)

5. Select **XML** as the Request Message Type and **None** as the Response Message Type.

6. Click **Next**.



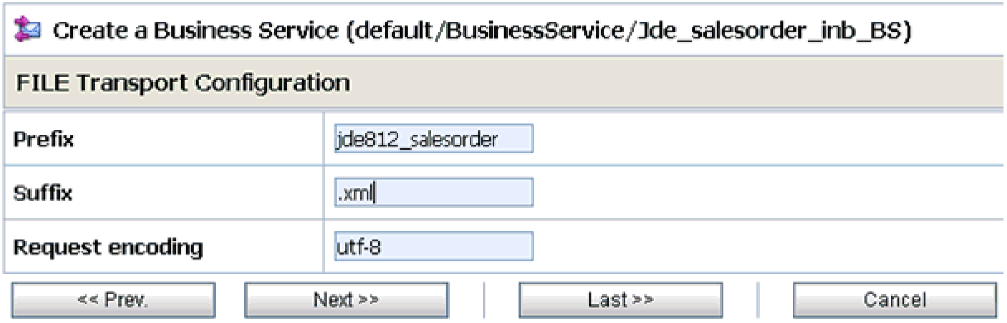
Transport Configuration

Protocol*	file
Load Balancing Algorithm	round-robin
Endpoint URI*	Format: file:///root-dir/dir1 file:/// EXISTING URIS file:///C:/output
Retry Count	0
Retry Iteration Interval	30

7. Select **file** from the Protocol drop-down list.

8. Enter the path to a destination folder on your file system in the Endpoint URI field and click **Add**.

9. Click **Next**.

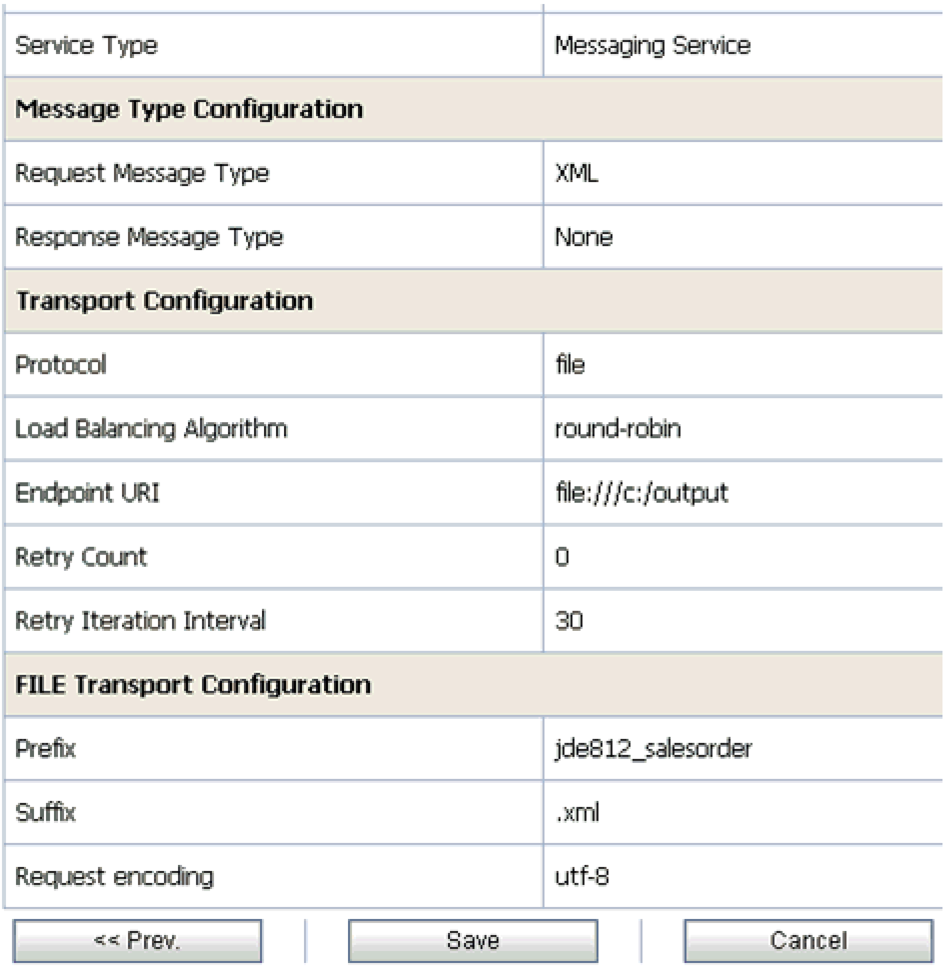


Create a Business Service (default/BusinessService/Jde_salesorder_inb_BS)

FILE Transport Configuration

Prefix	jde812_salesorder
Suffix	.xml
Request encoding	utf-8

10. Enter the prefix and suffix for the output file to be received and click **Next**.



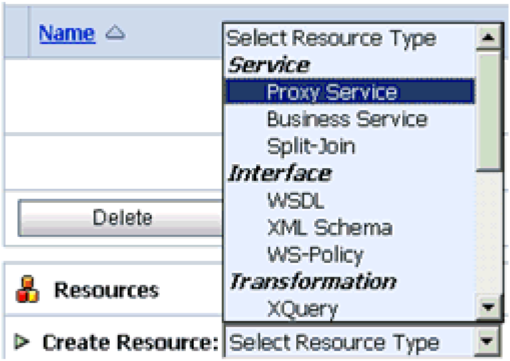
Service Type	Messaging Service
Message Type Configuration	
Request Message Type	XML
Response Message Type	None
Transport Configuration	
Protocol	file
Load Balancing Algorithm	round-robin
Endpoint URI	file:///c:/output
Retry Count	0
Retry Iteration Interval	30
FILE Transport Configuration	
Prefix	jde812_salesorder
Suffix	.xml
Request encoding	utf-8

11. Review all the information for your Business Service and click **Save**.

Configuring a Proxy Service

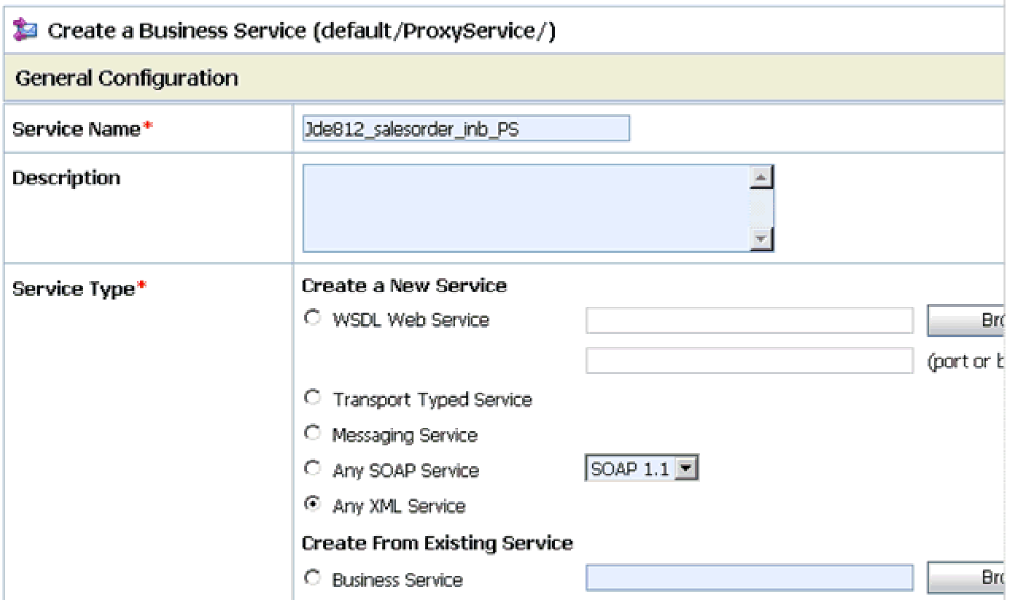
This section describes how to configure a Proxy Service using the Oracle Service Bus Console.

1. Select the **Proxy Service** folder you created in the left pane.



2. In the right pane, select **Proxy Service** from the Create Resource menu.

   The Create a Proxy Service - General Configuration page is displayed.



Create a Business Service (default/ProxyService/)

General Configuration

Service Name* Jde812_salesorder_inb_PS

Description

Service Type*

Create a New Service

○ WSDL Web Service

○ Transport Typed Service

○ Messaging Service

○ Any SOAP Service SOAP 1.1

● Any XML Service

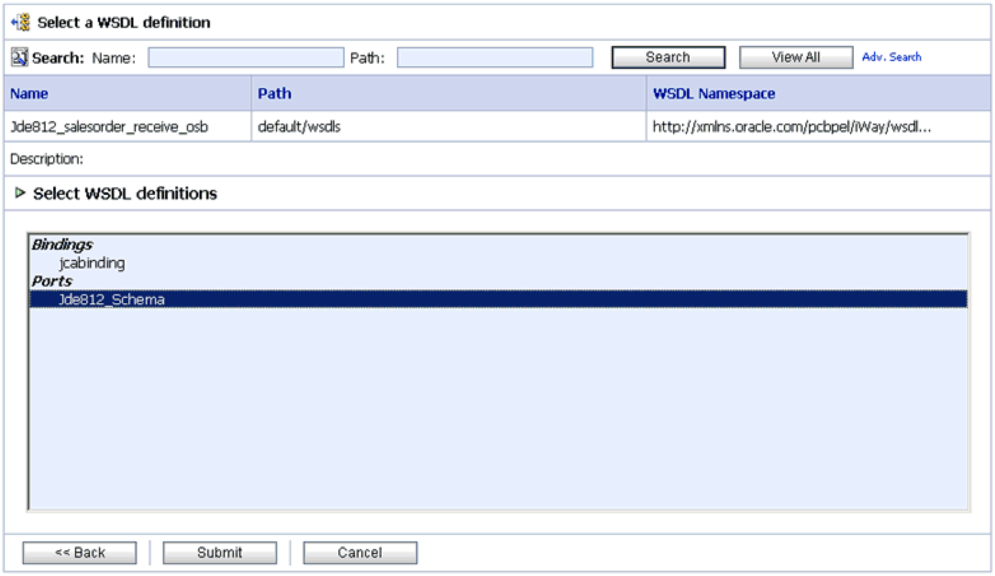
Create From Existing Service

○ Business Service

3. Provide a name for the Proxy Service and from the Service Type area select **WSDL Web Service**.

4. Click **Browse**.

   The Select a WSDL definition dialog box is displayed.



Select a WSDL definition

Search: Name: Path:

Name	Path	WSDL Namespace
Jde812_salesorder_receive_osb	default/wsdls	http://xmlns.oracle.com/pcbpel/iWay/wsdl...

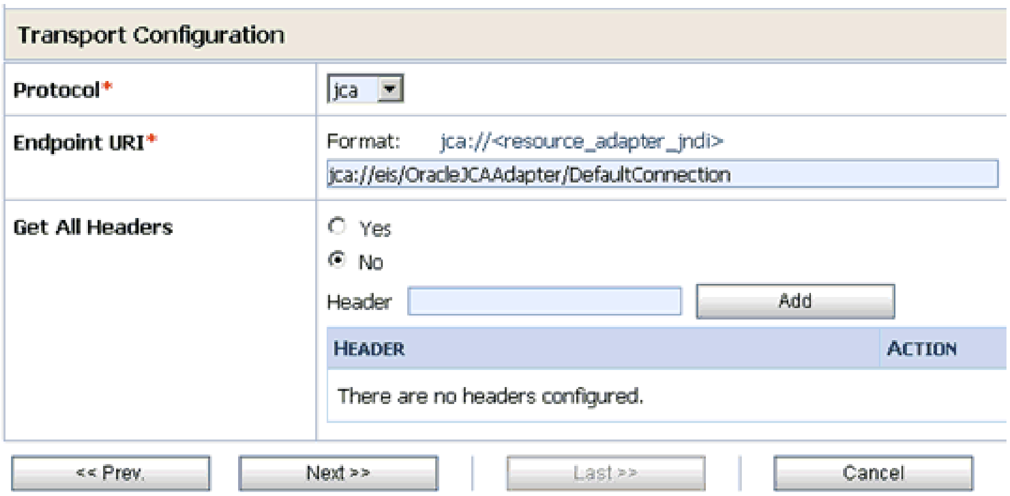
Description:

▷ Select WSDL definitions

Bindings
jcabinding
Ports
Jde812_Schema

5. Select an inbound WSDL for the Proxy Service and click **Submit**, then select any WSDL definition and click **Submit**.

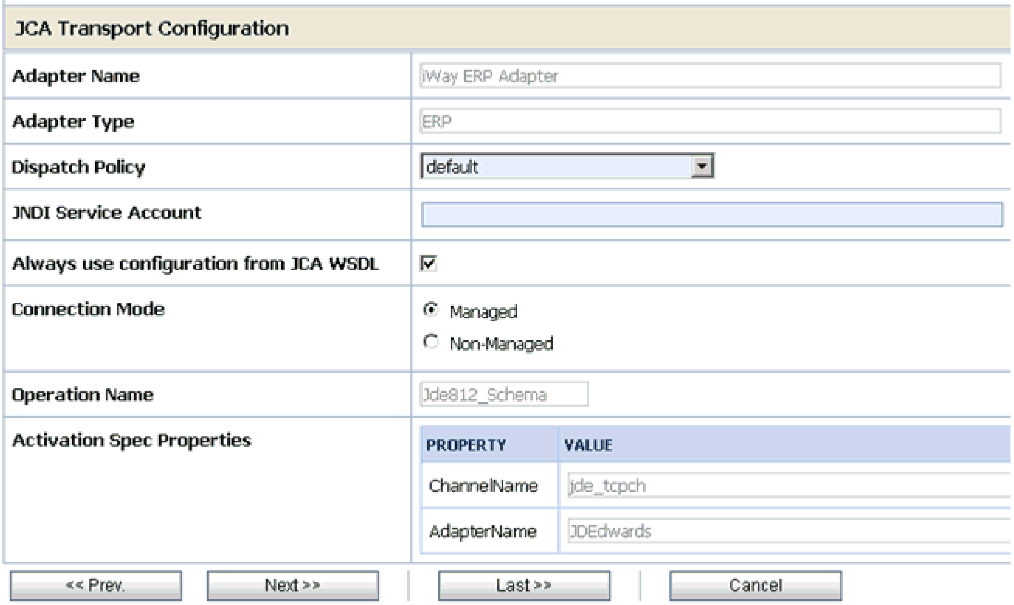
6. Click **Next**.



Transport Configuration

Protocol*	jca
Endpoint URI*	Format: jca://<resource_adapter_jndi> jca://eis/OracleJCAAdapter/DefaultConnection
Get All Headers	○ Yes ● No Header HEADER / ACTION There are no headers configured.

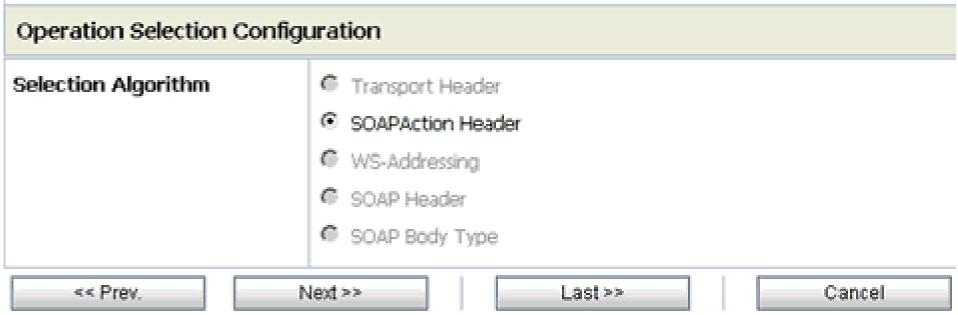
7. Click **Next**.



JCA Transport Configuration

Adapter Name	iWay ERP Adapter
Adapter Type	ERP
Dispatch Policy	default
JNDI Service Account	
Always use configuration from JCA WSDL	☑
Connection Mode	● Managed ○ Non-Managed
Operation Name	Jde812_Schema
Activation Spec Properties	PROPERTY: ChannelName, VALUE: jde_tcpch PROPERTY: AdapterName, VALUE: JDEdwards

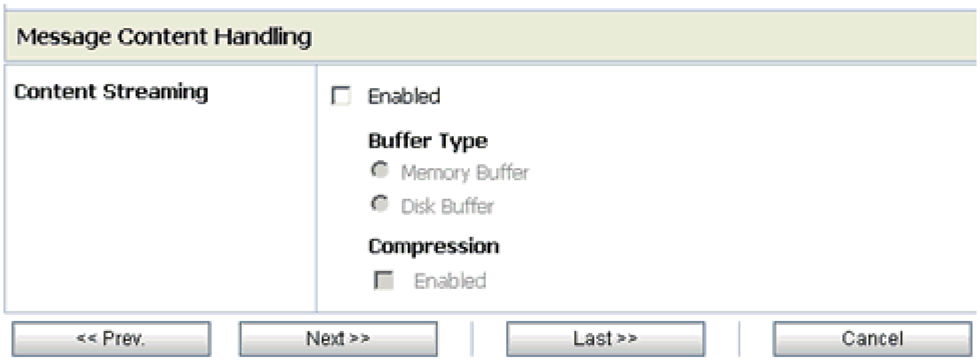
8. Click **Next**.



Operation Selection Configuration

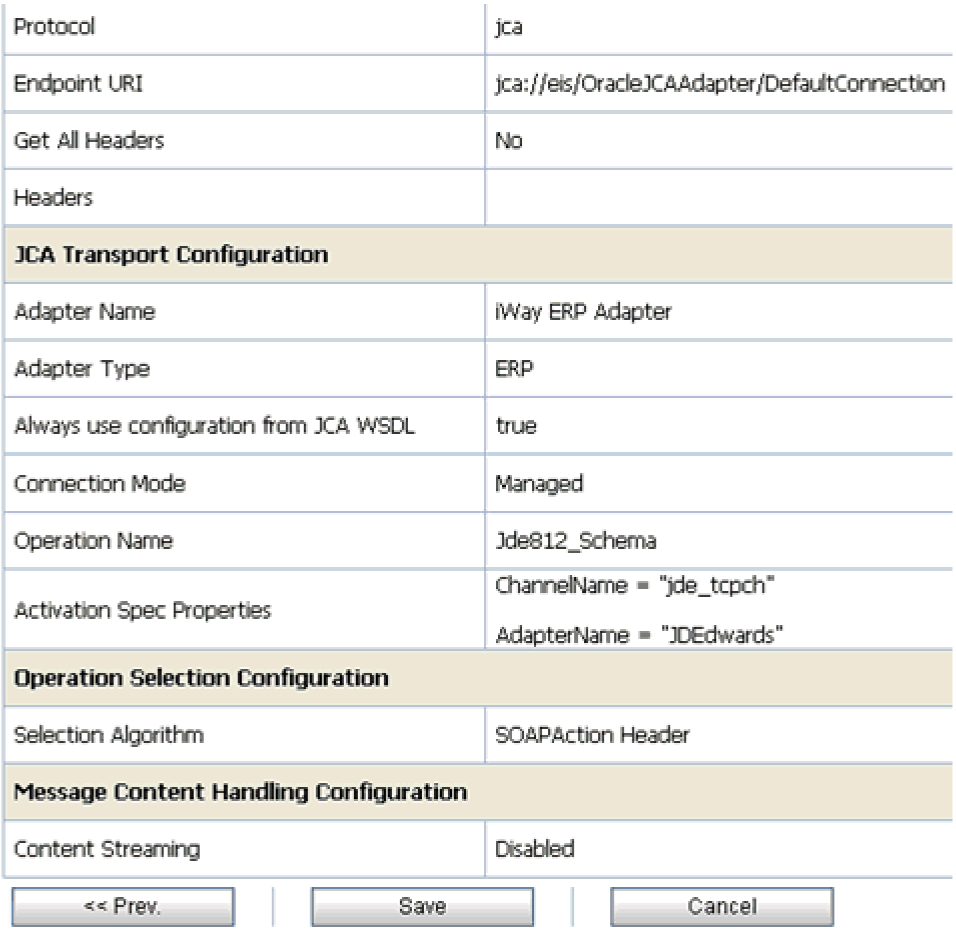
Selection Algorithm	○ Transport Header ● SOAPAction Header ○ WS-Addressing ○ SOAP Header ○ SOAP Body Type

9. Click **Next**.



Message Content Handling	
Content Streaming	□ Enabled **Buffer Type** ○ Memory Buffer ○ Disk Buffer **Compression** □ Enabled

10. Click **Next**.



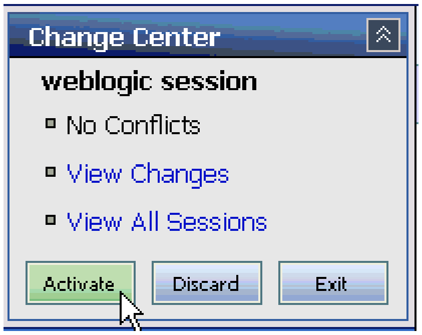
Protocol	jca
Endpoint URI	jca://eis/OracleJCAAdapter/DefaultConnection
Get All Headers	No
Headers	
JCA Transport Configuration	
Adapter Name	iWay ERP Adapter
Adapter Type	ERP
Always use configuration from JCA WSDL	true
Connection Mode	Managed
Operation Name	Jde812_Schema

Activation Spec Properties	ChannelName = "jde_tcpch"
	AdapterName = "JDEdwards"

Operation Selection Configuration	
Selection Algorithm	SOAPAction Header
Message Content Handling Configuration	
Content Streaming	Disabled

11. Review all the information for your Proxy Service and click **Save**.

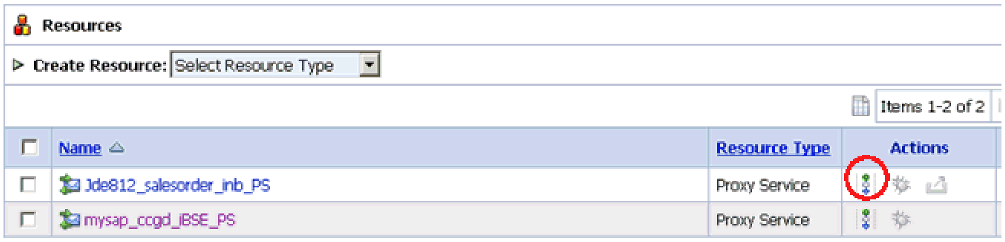
12. Click **Activate** in the Change Center area to activate your changes in the Oracle Service Bus session.



Configuring a Pipeline

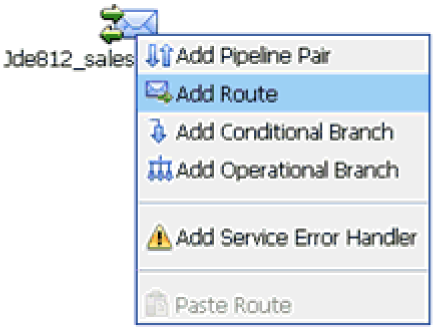
This section describes how to configure a Pipeline using the Oracle Service Bus Console.

1. Click the **Edit Message Flow** icon in the row of the Proxy Service you created.

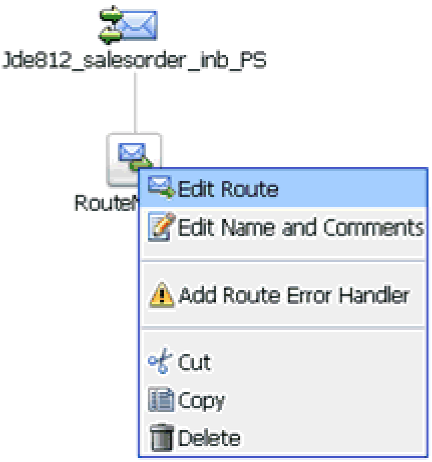


□	Name	Resource Type	Actions
□	Jde812_salesorder_inb_PS	Proxy Service	
□	mysap_ccgd_iBSE_PS	Proxy Service	

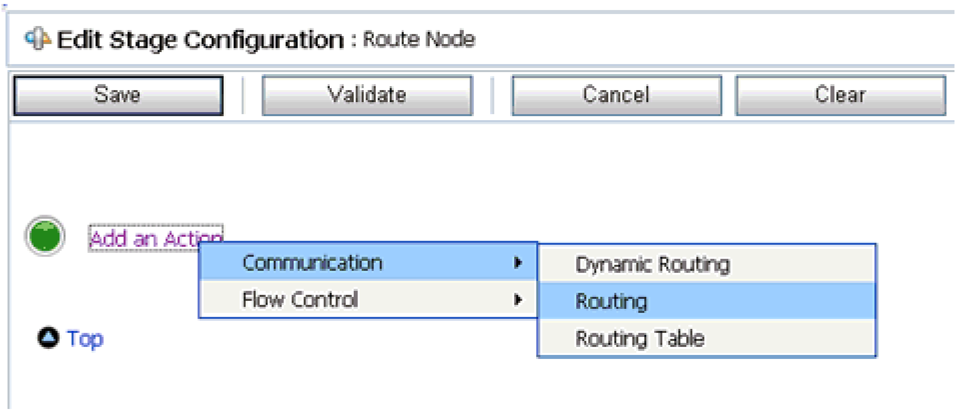
The Edit Message Flow workspace area is displayed.



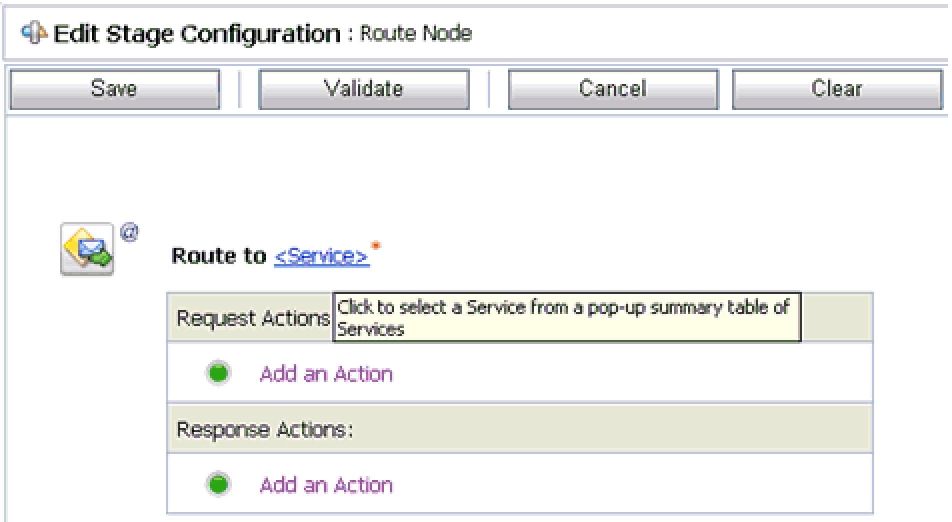
2. Click the **Proxy Service** icon and select **Add Route** from the context menu, as shown in the following image.



3. Click the **RouteNode1** icon and select **Edit Route** from the context menu.



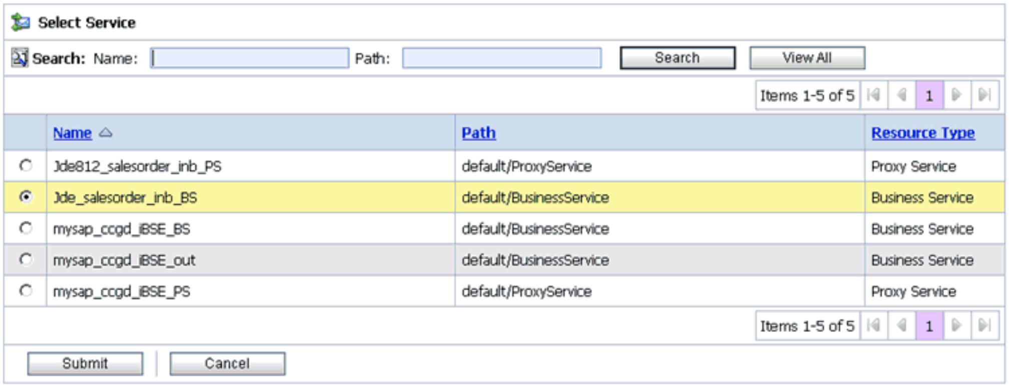
4. Click **Add an Action**, select **Communication** from the context menu, and click **Routing**.



Click to select a Service from a pop-up summary table of Services

5. Click **<Service>**.

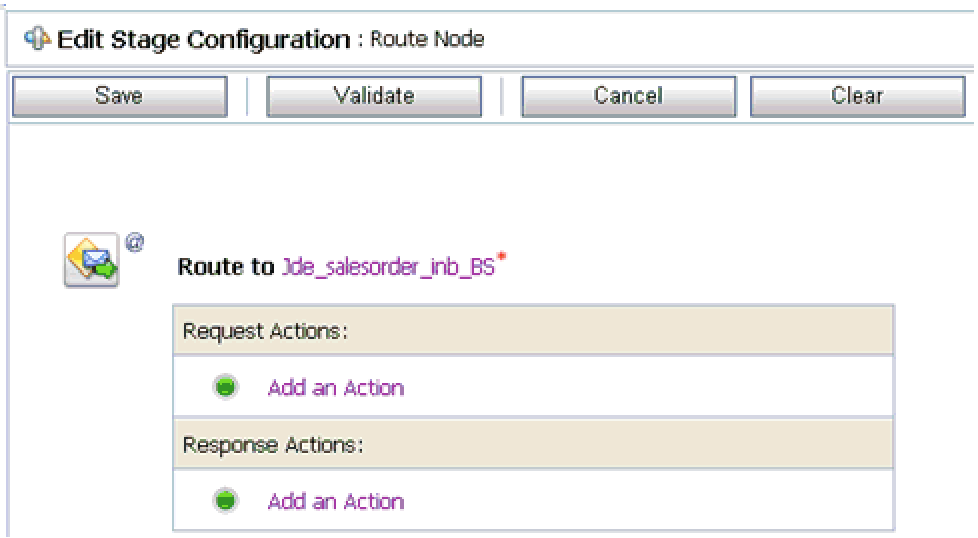
   The Select Service dialog box is displayed.



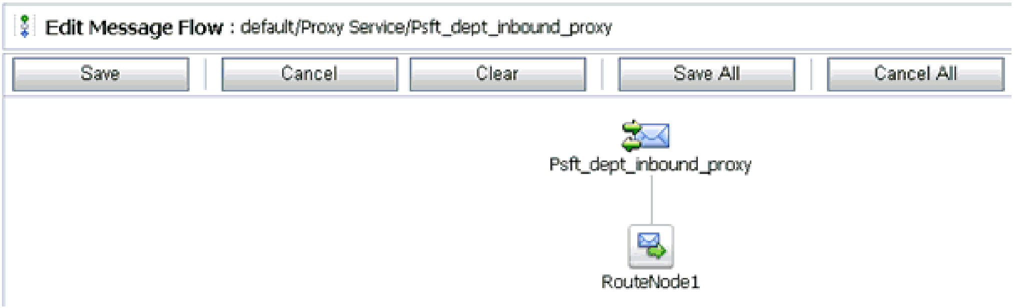
	Name	Path	Resource Type
○	Jde812_salesorder_inb_PS	default/ProxyService	Proxy Service
●	Jde_salesorder_inb_BS	default/BusinessService	Business Service
○	mysap_ccgd_iBSE_BS	default/BusinessService	Business Service
○	mysap_ccgd_iBSE_out	default/BusinessService	Business Service
○	mysap_ccgd_iBSE_PS	default/ProxyService	Proxy Service

6. Select a File type Business Service and click **Submit**.

   You are returned to the Edit Stage Configuration workspace area.

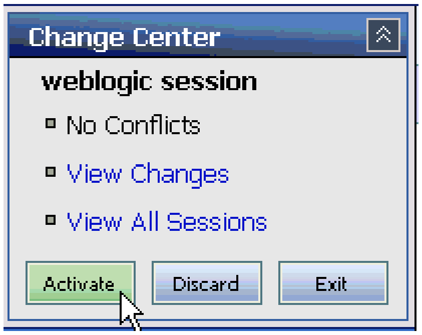


7. Click **Validate** and then **Save**.

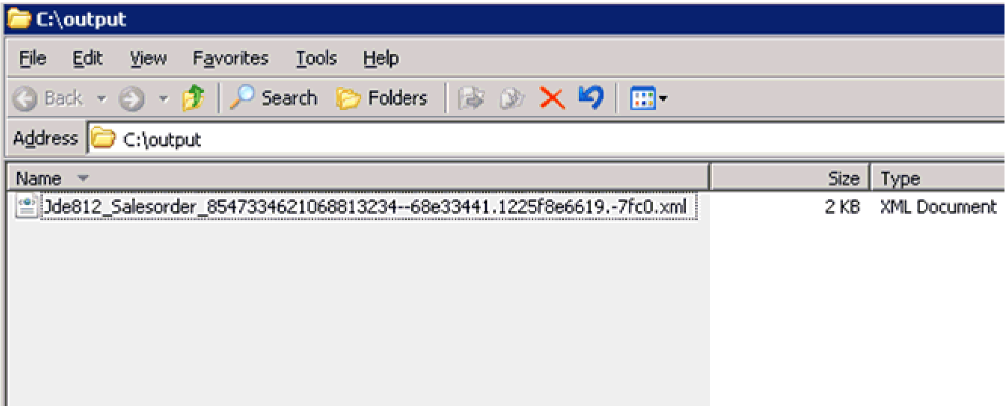


8. Click **Save**.

9. Click **Activate** in the Change Center area to activate your changes in the Oracle Service Bus session.



10. Trigger an event message from the J.D. Edwards OneWorld system and you will receive the output XML in the destination folder.



Name	Size	Type
Jde812_Salesorder_8547334621068813234--68e33441.1225f8e6619.-7fc0.xml	2 KB	XML Document

Configuring Outbound Processing Using Oracle Service Bus (J2CA Configuration)

This section describes how to configure outbound processing using Oracle Service Bus (J2CA Configuration).

This section includes the following topics:

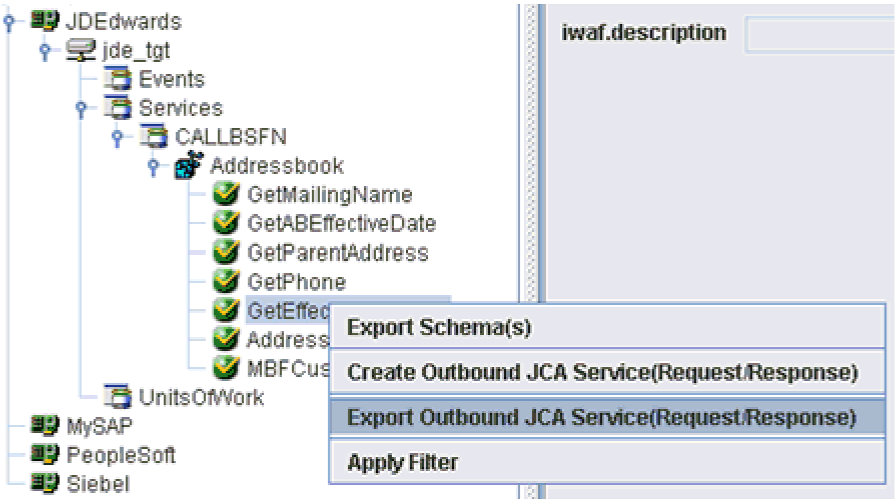
- Publishing a WSDL From Application Explorer to Oracle Service Bus
- Starting Oracle Service Bus and Creating Project Folders
- Configuring a File Type Business Service
- Configuring a WSDL Type Business Service
- Configuring a Proxy Service
- Configuring a Pipeline

Publishing a WSDL From Application Explorer to Oracle Service Bus

This section describes how to publish a WSDL from Application Explorer (J2CA configuration) to Oracle Service Bus.

1. Start Application Explorer, connect to a J2CA configuration, and connect to a J.D. Edwards OneWorld target.

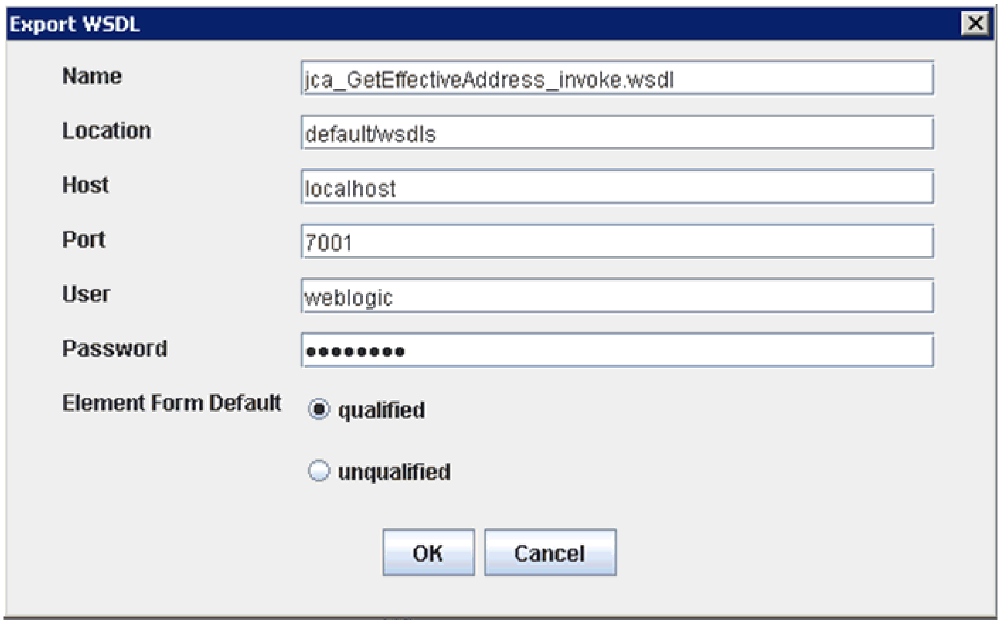
   For more information, see Chapter 2, "Configuring OracleWLS Application Adapter for J.D. Edwards One World".



2. Expand the **Services** node, followed by **CALLBSFN** and **Addressbook**.

3. Right-click the **GetEffectiveAddress** business object and select **Export Outbound JCA Service(Request/Response)** from the context menu.

   The Export WSDL dialog box is displayed.



Name: jca_GetEffectiveAddress_invoke.wsdl
Location: default/wsdls
Host: localhost
Port: 7001
User: weblogic
Password: ••••••••
Element Form Default: ● qualified ○ unqualified

4. In the Name field, a default file name for the WSDL file is provided. You can accept the default or provide your own.

5. In the Location field, enter the location where you want to publish the WSDL document.

   The location is composed of an Oracle Service Bus project name and optionally, one or more folder names. The project name and any folder names must be separated by a back slash character “\”.

6. In the Host field, enter the name of the machine where Oracle WebLogic Server is running.

7. In the Port field, enter the port for the domain you are using.

8. In the User field, enter your username to access Oracle Service Bus.
9. In the Password field, enter your password to access Oracle Service Bus.
10. Ensure that **qualified** is selected as the element form, which is the default.
11. Click **OK**.

    The WSDL is published to the location specified in the Export WSDL dialog box and is now available for use with a Business Service or Proxy Service in Oracle Service Bus.

Starting Oracle Service Bus and Creating Project Folders

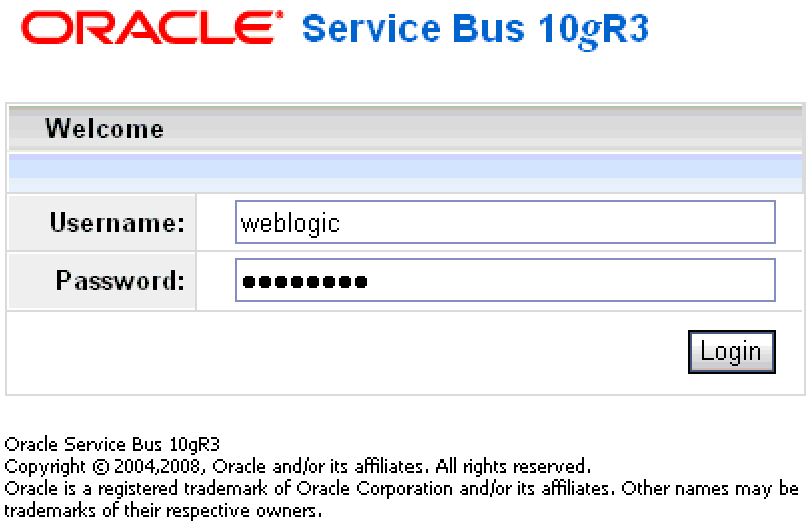
This section describes how to access the Oracle Service Bus Console and create project folders.

1. Start the Oracle WebLogic Server for the Oracle WebLogic Server domain that you have configured.
2. Open the Oracle Service Bus Console in a Web browser by entering the following URL:

    ```
    http://hostname:port/sbconsole
    ```

    Where *hostname* is the name of the machine where Oracle WebLogic Server is running and *port* is the port for the domain you are using. The port for the default domain is 7001.

    The Oracle Service Bus Console logon page is displayed.



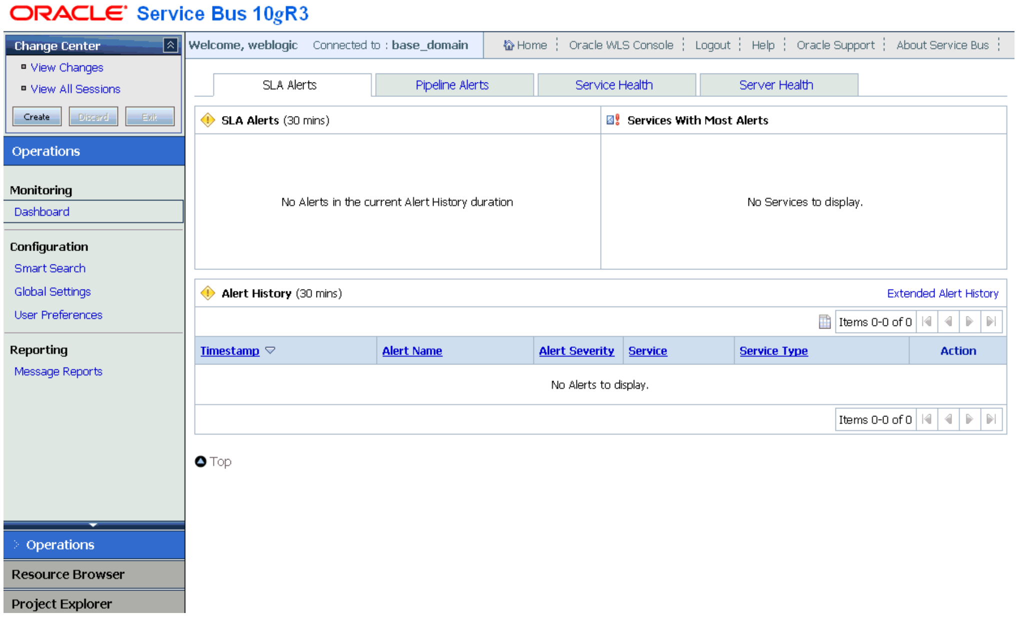
Username: weblogic

Password: ●●●●●●●●●

Oracle Service Bus 10gR3
Copyright © 2004,2008, Oracle and/or its affiliates. All rights reserved.
Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

3. Log on to the Oracle Service Bus Console using a valid user name and password.

    The Oracle Service Bus Console home page is displayed.

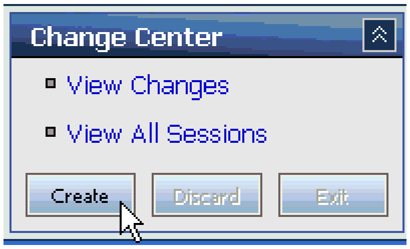


No Alerts in the current Alert History duration

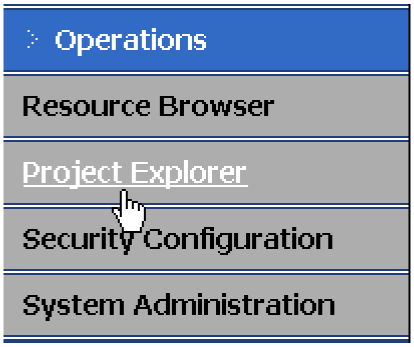
No Services to display.

Timestamp	Alert Name	Alert Severity	Service	Service Type	Action
No Alerts to display.					

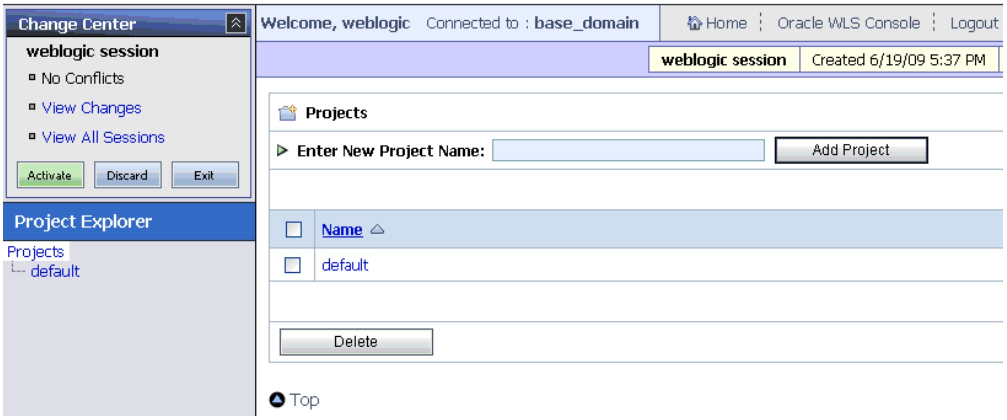
4. Click **Create** in the Change Center area to start a new Oracle Service Bus session.



5. Click **Project Explorer** in the left pane.

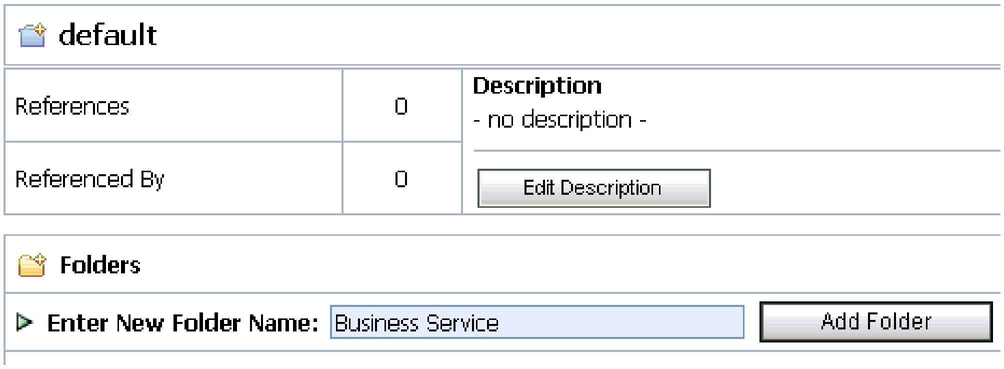


The Project Explorer page is displayed.



6. Click the **default** project node in the left pane.

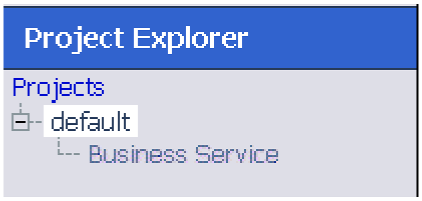
   The default project page is displayed.



Enter New Folder Name: Business Service

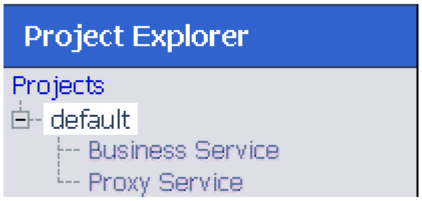
7. In the Enter New Folder Name field, type **Business Service** and click **Add Folder**.

   The Business Service folder is listed in the left pane below the default project node.



8. In the Enter New Folder Name field, type **Proxy Service** and click **Add Folder**.

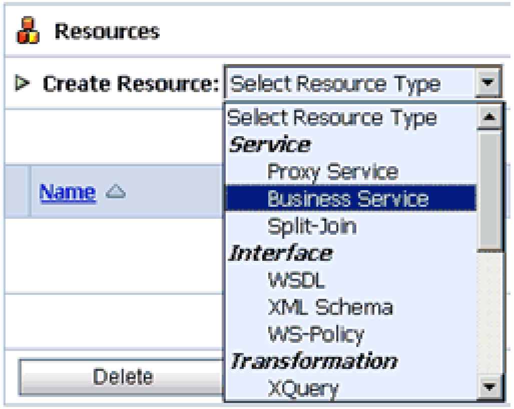
   The Business Service and Proxy Service folders are listed in the left pane below the default project node.



Configuring a File Type Business Service

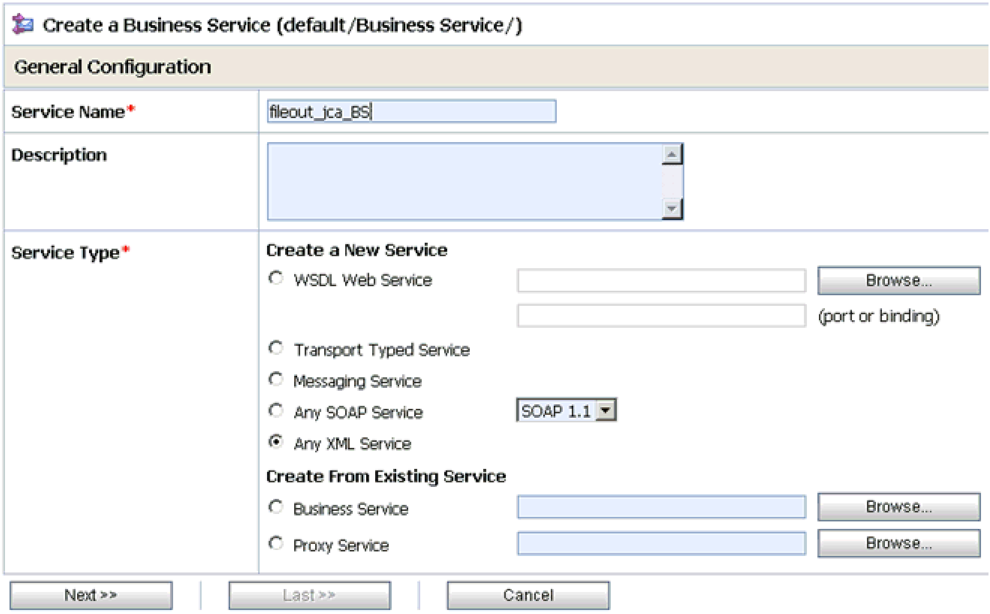
This section describes how to configure a File type Business Service using the Oracle Service Bus Console.

1. Select the **Business Service** folder you created in the left pane.



2. In the right pane, select **Business Service** from the Create Resource menu.

   The Create a Business Service - General Configuration page is displayed.


Create a Business Service (default/Business Service/)

General Configuration

Service Name*: fileout_jca_BS

Description:

Service Type*:

Create a New Service

○ WSDL Web Service (port or binding)

○ Transport Typed Service

○ Messaging Service

○ Any SOAP Service SOAP 1.1

● Any XML Service

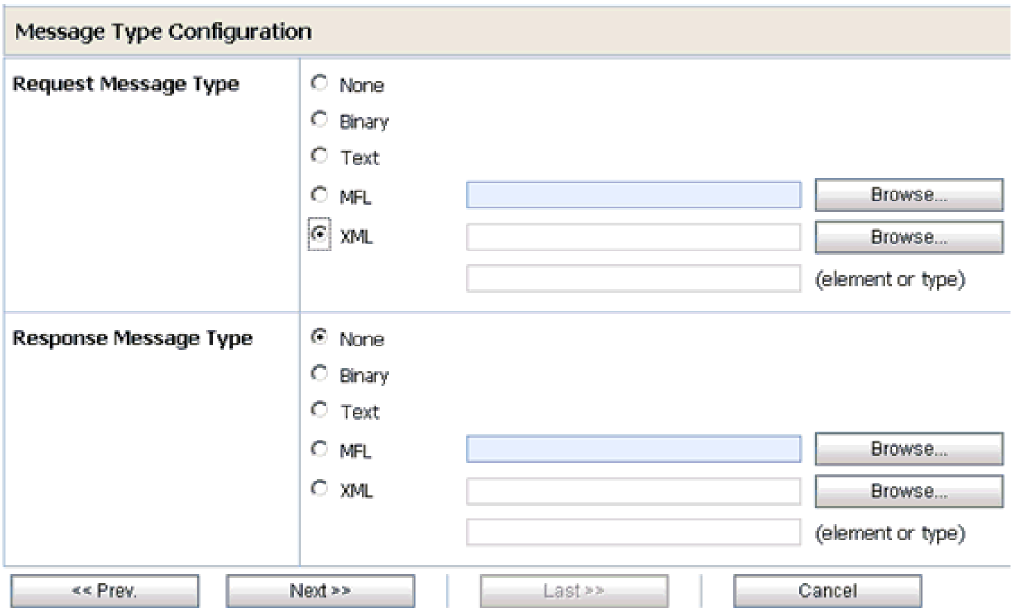
Create From Existing Service

○ Business Service

○ Proxy Service

3. Provide a name for the Business Service and from the Service Type area select **Messaging Service**.

4. Click **Next**.

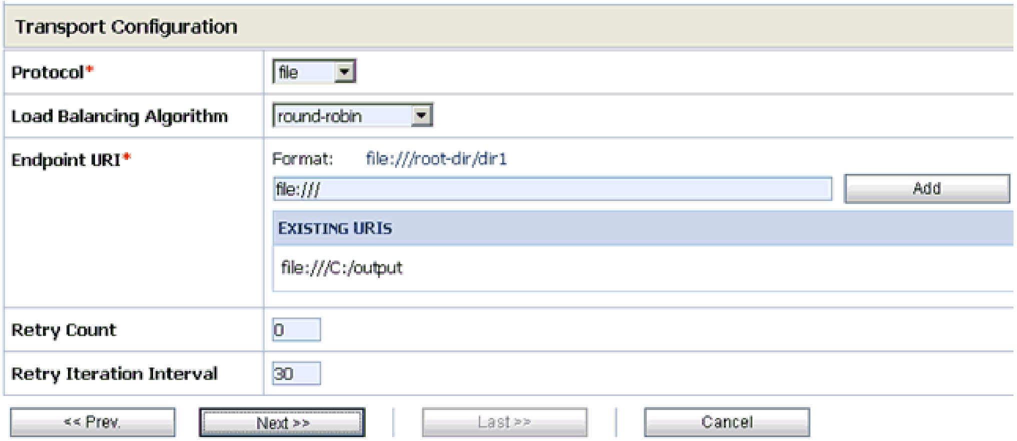


Message Type Configuration

Request Message Type	○ None ○ Binary ○ Text ○ MFL ● XML
Response Message Type	● None ○ Binary ○ Text ○ MFL ○ XML

5. Select **XML** as the Request Message Type and **None** as the Response Message Type.

6. Click **Next**.



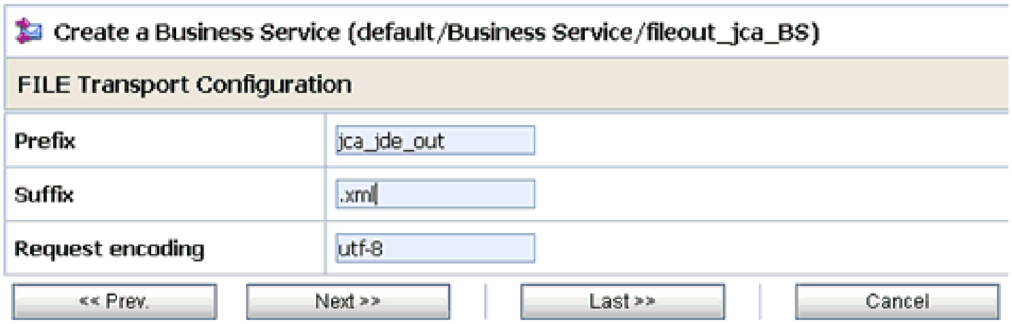
Transport Configuration

Protocol*	file
Load Balancing Algorithm	round-robin
Endpoint URI*	Format: file:///root-dir/dir1 file:/// EXISTING URIS file:///C:/output
Retry Count	0
Retry Iteration Interval	30

7. Select **file** from the Protocol drop-down list.

8. Enter the path to a destination folder on your file system in the Endpoint URI field and click **Add**.

9. Click **Next**.

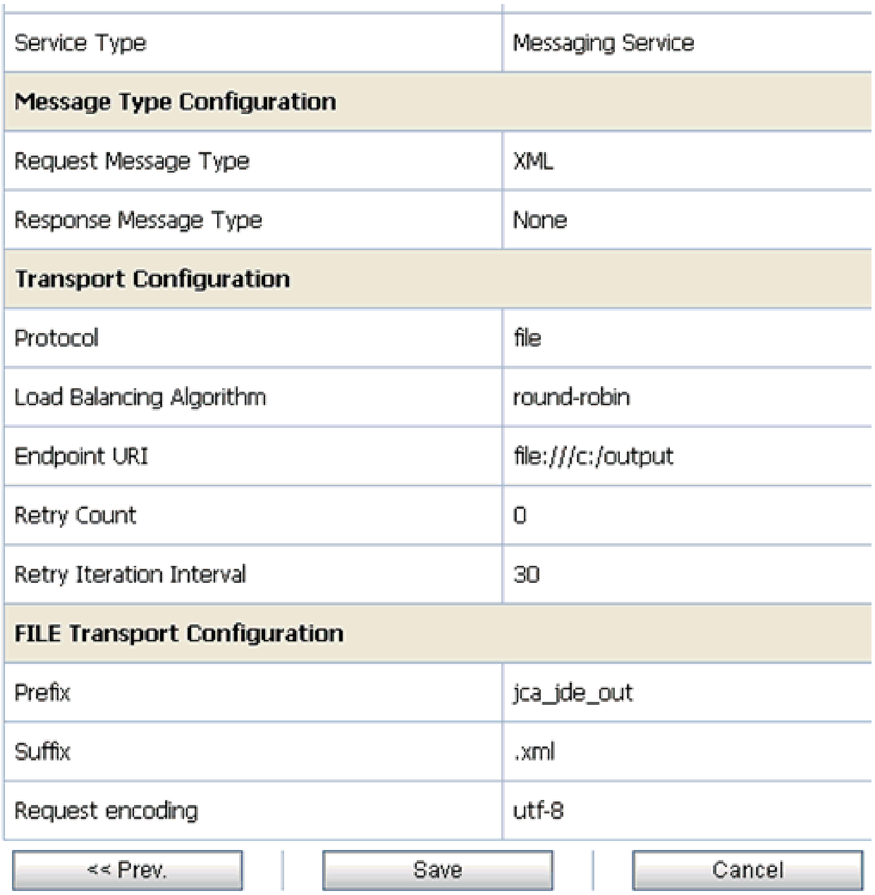


Create a Business Service (default/Business Service/fileout_jca_BS)

FILE Transport Configuration

Prefix	jca_jde_out
Suffix	.xml
Request encoding	utf-8

10. Enter the prefix and suffix for the output file to be received and click **Next**.



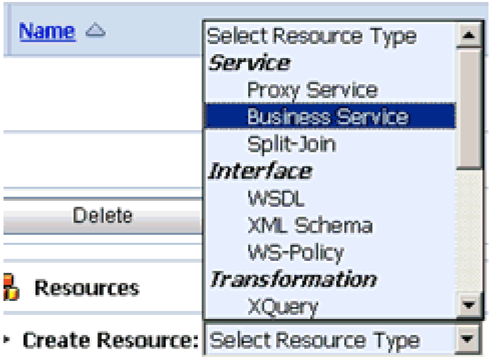
Service Type	Messaging Service
Message Type Configuration	
Request Message Type	XML
Response Message Type	None
Transport Configuration	
Protocol	file
Load Balancing Algorithm	round-robin
Endpoint URI	file:///c:/output
Retry Count	0
Retry Iteration Interval	30
FILE Transport Configuration	
Prefix	jca_jde_out
Suffix	.xml
Request encoding	utf-8

11. Review all the information for your Business Service and click **Save**.

Configuring a WSDL Type Business Service

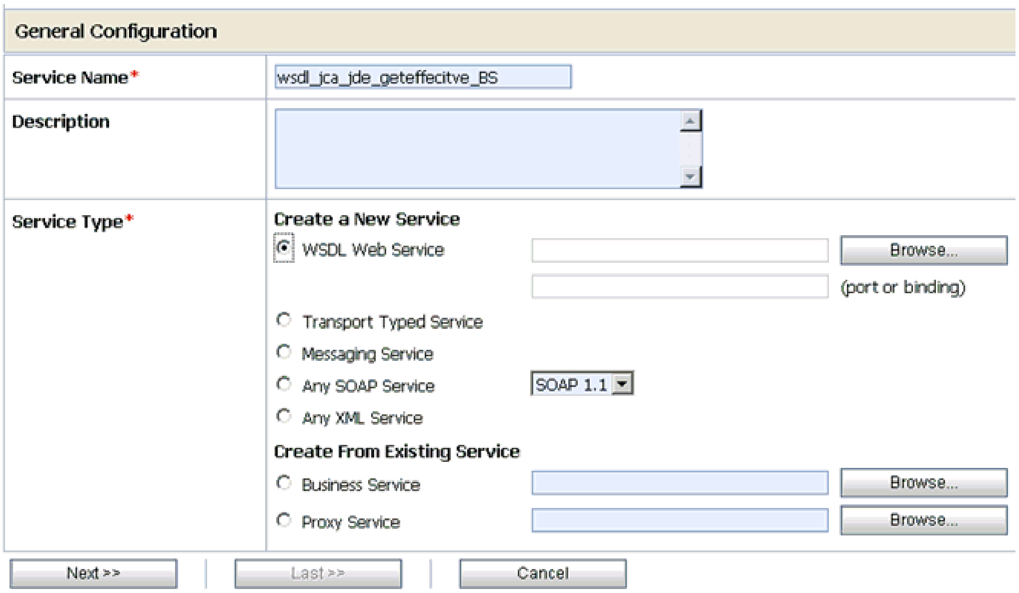
This section describes how to configure a WSDL type Business Service using the Oracle Service Bus Console.

1. Select the **Business Service** folder you created in the left pane.



2. In the right pane, select **Business Service** from the Create Resource menu.

   The Create a Business Service - General Configuration page is displayed.



General Configuration

Service Name*: wsdl_jca_jde_geteffecitve_BS

Description:

Service Type*:

Create a New Service

● WSDL Web Service (port or binding)

○ Transport Typed Service

○ Messaging Service

○ Any SOAP Service SOAP 1.1

○ Any XML Service

Create From Existing Service

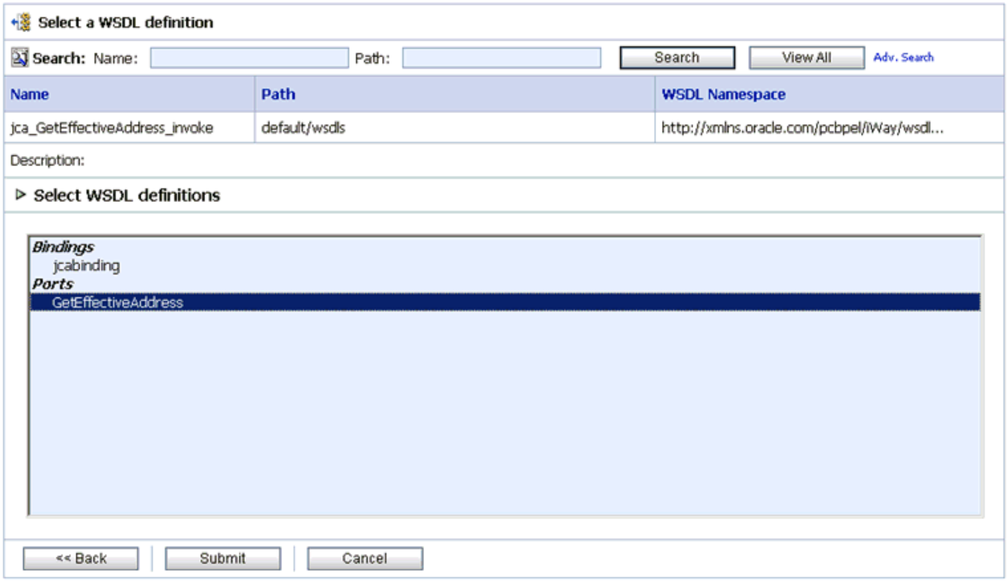
○ Business Service

○ Proxy Service

3. Provide a name for the Business Service and from the Service Type area select **WSDL Web Service**.

4. Click **Browse**.

   The Select a WSDL Definition dialog box is displayed.



Select a WSDL definition

Search: Name: Path:

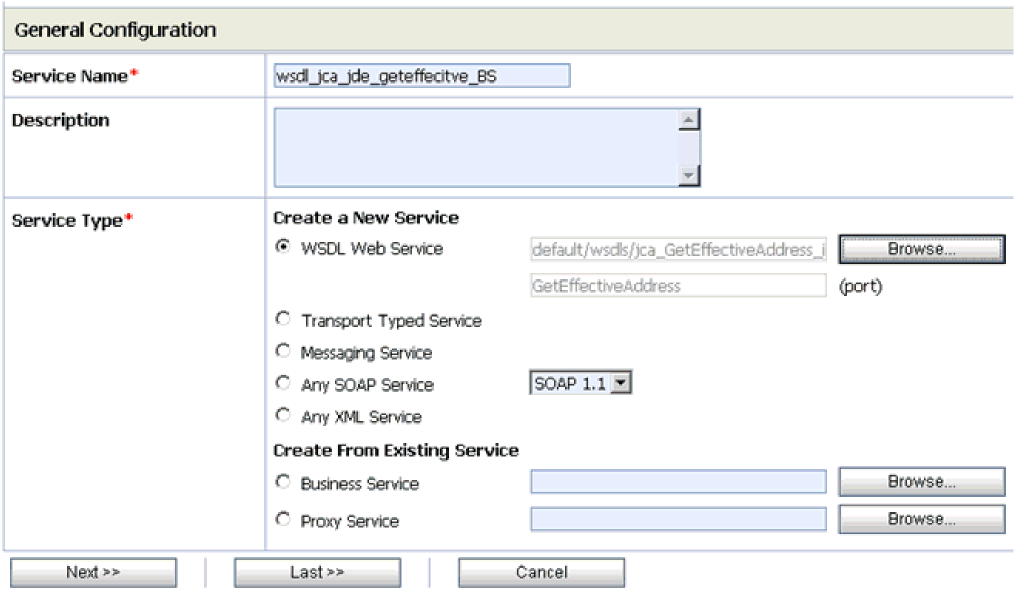
Name	Path	WSDL Namespace
jca_GetEffectiveAddress_invoke	default/wsdls	http://xmlns.oracle.com/pcbpel/iWay/wsdl...

Description:

Select WSDL definitions

Bindings
jcabinding
Ports
GetEffectiveAddress

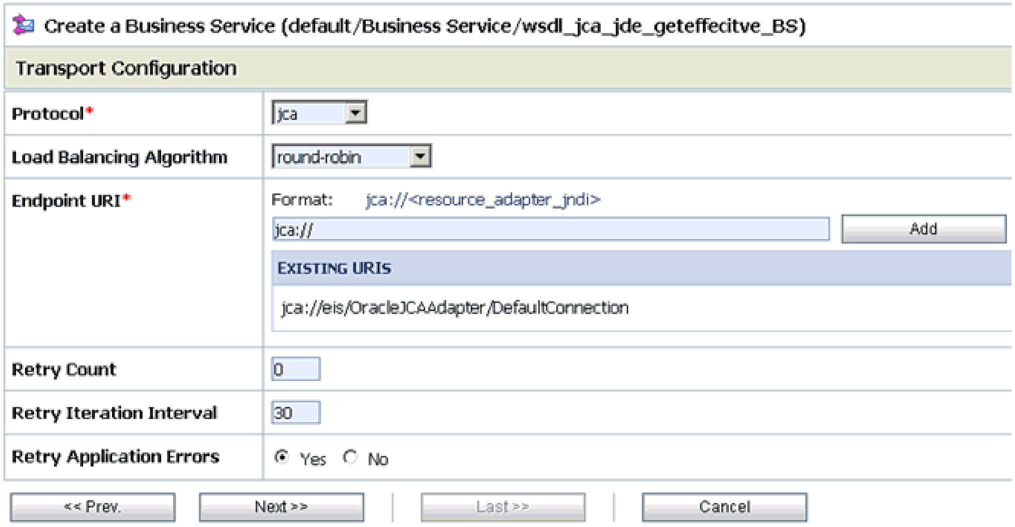
5. Select an outbound WSDL and click **Submit**, then select any WSDL definition and click **Submit**.



General Configuration

Service Name*	wsdl_jca_jde_geteffecitve_BS
Description	
Service Type*	**Create a New Service** ● WSDL Web Service default/wsdls/jca_GetEffectiveAddress_i GetEffectiveAddress (port) ○ Transport Typed Service ○ Messaging Service ○ Any SOAP Service SOAP 1.1 ○ Any XML Service **Create From Existing Service** ○ Business Service ○ Proxy Service

6. Click **Next**.



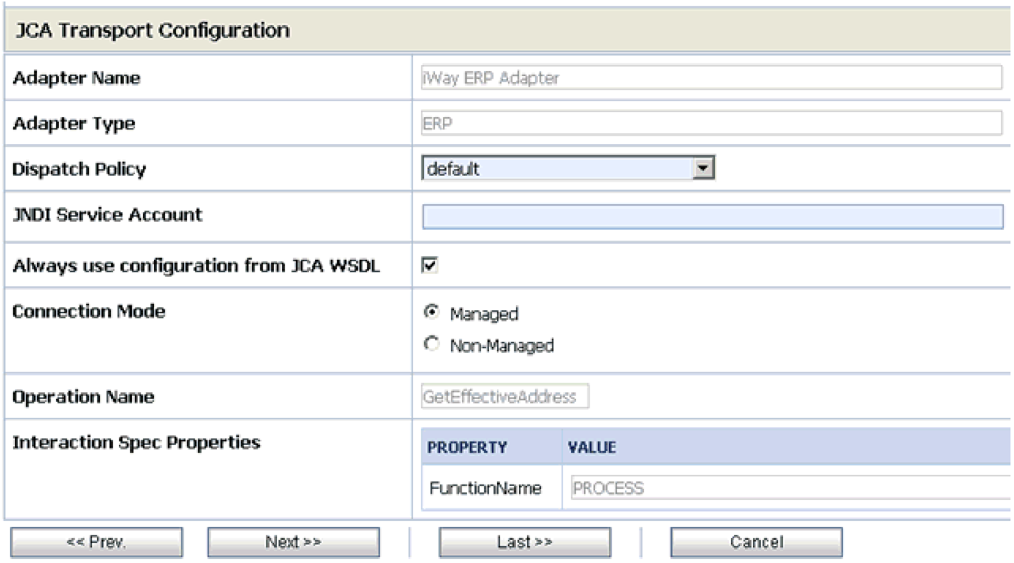
Create a Business Service (default/Business Service/wsdl_jca_jde_geteffecitve_BS)

Transport Configuration

Protocol*	jca
Load Balancing Algorithm	round-robin
Endpoint URI*	Format: jca://<resource_adapter_jndi> jca:// EXISTING URIS jca://eis/OracleJCAAdapter/DefaultConnection
Retry Count	0
Retry Iteration Interval	30
Retry Application Errors	● Yes ○ No

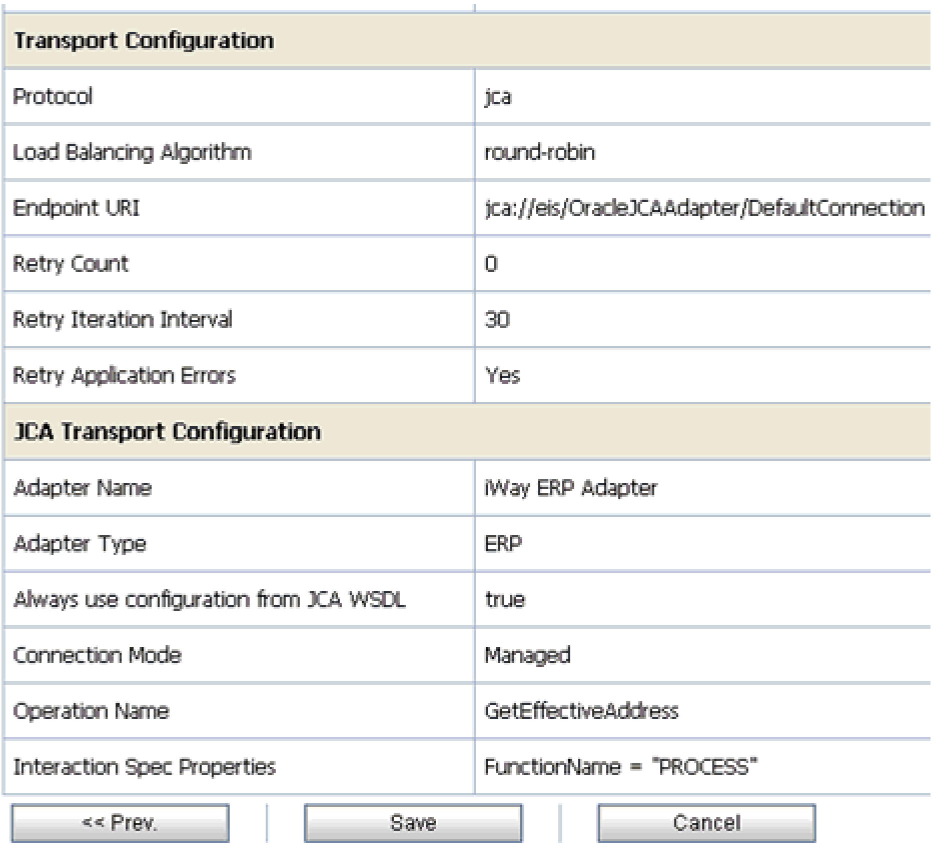
7. Select **jca** from the Protocol drop-down list.

8. Click **Next**.



JCA Transport Configuration	
Adapter Name	iWay ERP Adapter
Adapter Type	ERP
Dispatch Policy	default
JNDI Service Account	
Always use configuration from JCA WSDL	☑
Connection Mode	● Managed ○ Non-Managed
Operation Name	GetEffectiveAddress
Interaction Spec Properties	PROPERTY: FunctionName VALUE: PROCESS

9. Click **Next**.



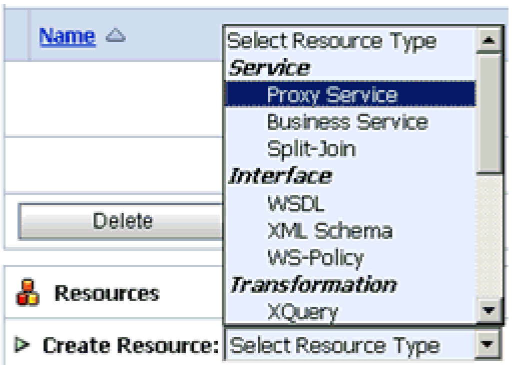
Transport Configuration	
Protocol	jca
Load Balancing Algorithm	round-robin
Endpoint URI	jca://eis/OracleJCAAdapter/DefaultConnection
Retry Count	0
Retry Iteration Interval	30
Retry Application Errors	Yes
JCA Transport Configuration	
Adapter Name	iWay ERP Adapter
Adapter Type	ERP
Always use configuration from JCA WSDL	true
Connection Mode	Managed
Operation Name	GetEffectiveAddress
Interaction Spec Properties	FunctionName = "PROCESS"

10. Review all the information for your Business Service and click **Save**.

Configuring a Proxy Service

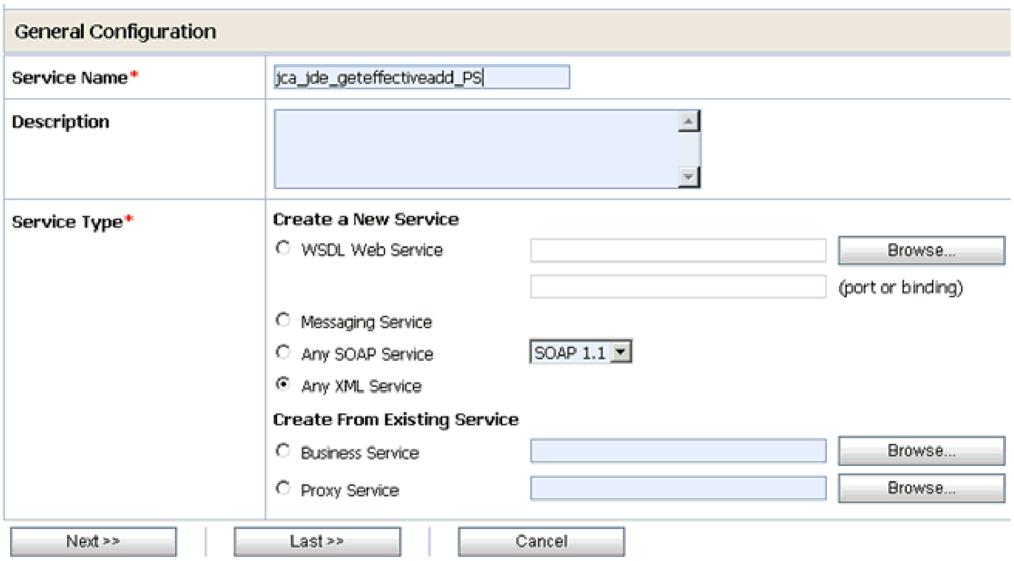
This section describes how to configure a Proxy Service using the Oracle Service Bus Console.

1. Select the **Proxy Service** folder you created in the left pane.



2. In the right pane, select **Proxy Service** from the Create Resource menu.

   The Create a Proxy Service - General Configuration page is displayed.

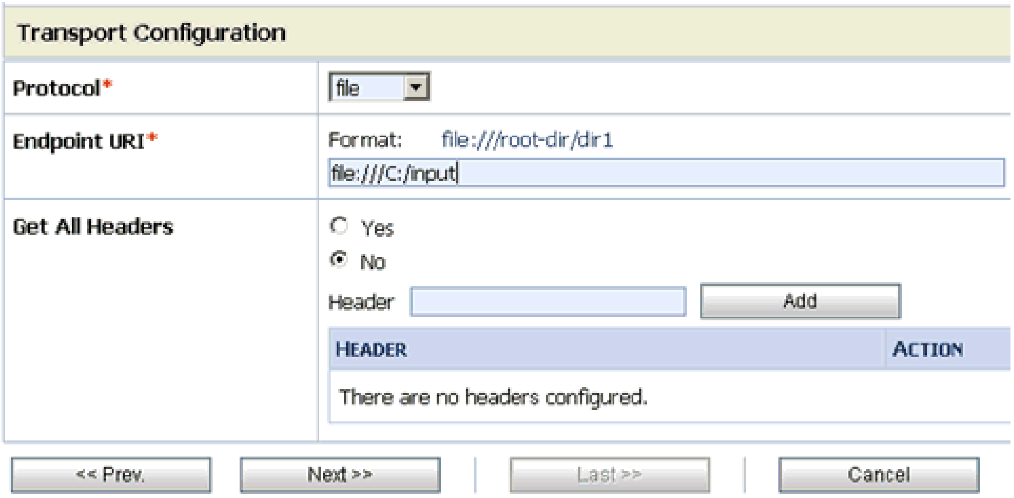


General Configuration

Service Name*	jca_jde_geteffectiveadd_PS
Description	
Service Type*	**Create a New Service** ○ WSDL Web Service (port or binding) ○ Messaging Service ○ Any SOAP Service SOAP 1.1 ● Any XML Service **Create From Existing Service** ○ Business Service ○ Proxy Service

3. Provide a name for the Proxy Service and from the Service Type area select **Any XML Service**.

4. Click **Next**.

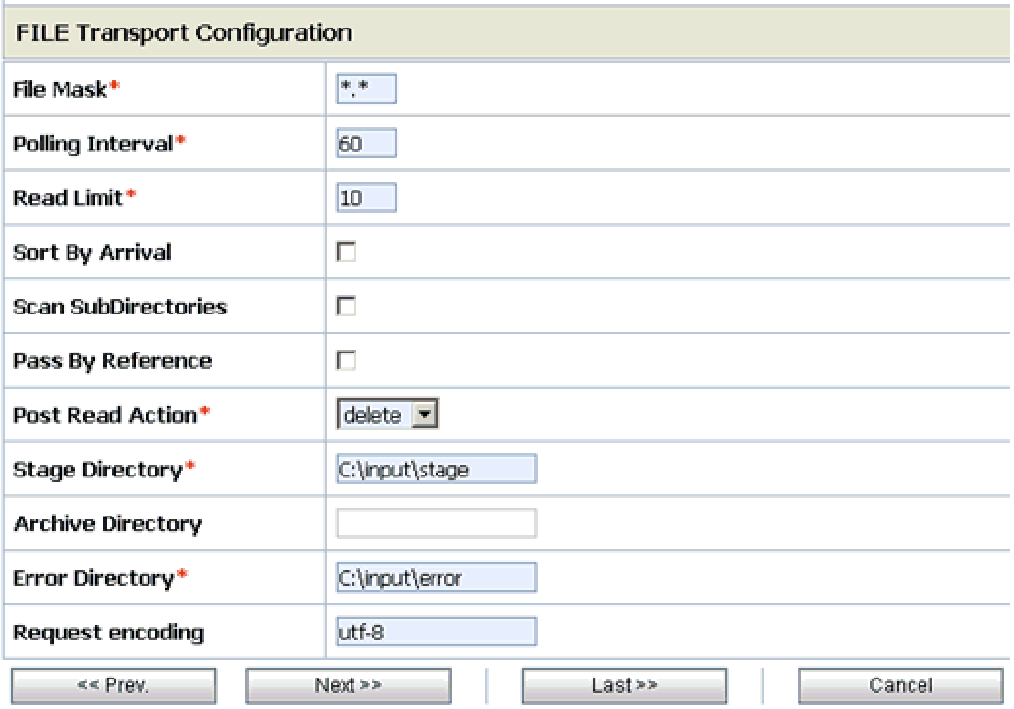


Transport Configuration

Protocol*	file
Endpoint URI*	Format: file:///root-dir/dir1 file:///C:/input
Get All Headers	○ Yes ● No Header

HEADER	ACTION
There are no headers configured.	

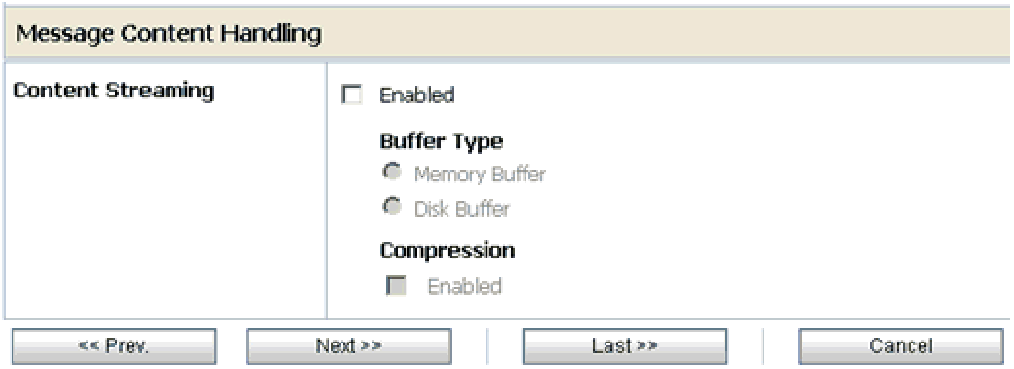
5. Select **file** from the Protocol drop-down list.
6. Enter the path to an input folder on your file system in the Endpoint URI field.
7. Click **Next**.



FILE Transport Configuration

File Mask*	*.*
Polling Interval*	60
Read Limit*	10
Sort By Arrival	□
Scan SubDirectories	□
Pass By Reference	□
Post Read Action*	delete
Stage Directory*	C:\input\stage
Archive Directory	
Error Directory*	C:\input\error
Request encoding	utf-8

8. Provide any folder locations on your file system for the Stage Directory and Error Directory fields.
9. Click **Next**.



Message Content Handling

Content Streaming

□ Enabled

Buffer Type

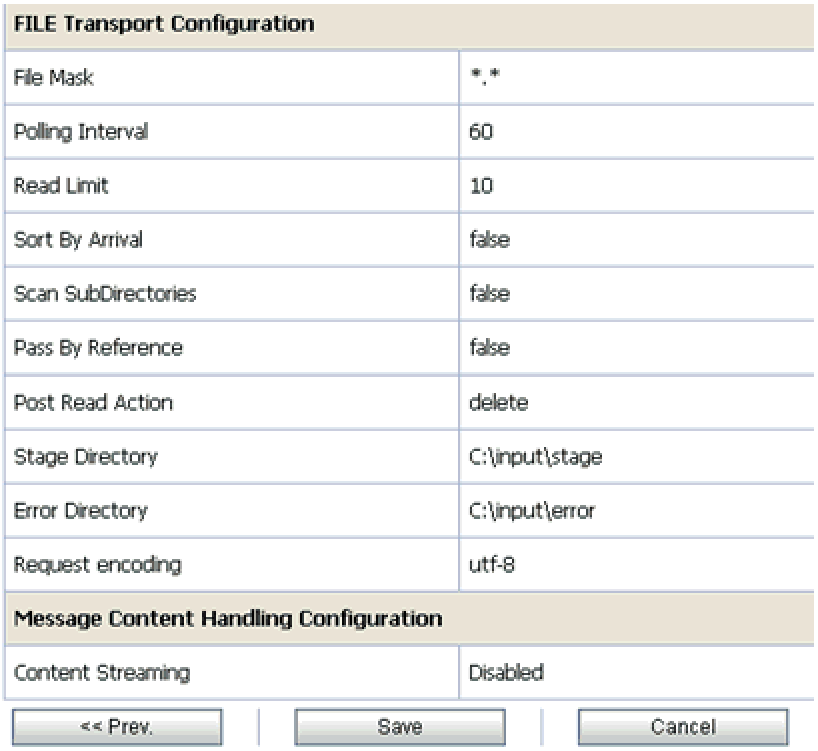
○ Memory Buffer

○ Disk Buffer

Compression

□ Enabled

10. Click **Next**.



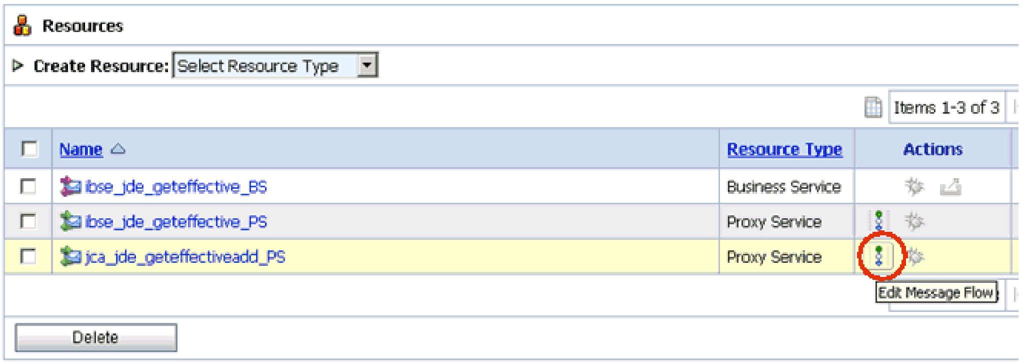
FILE Transport Configuration	
File Mask	*.*
Polling Interval	60
Read Limit	10
Sort By Arrival	false
Scan SubDirectories	false
Pass By Reference	false
Post Read Action	delete
Stage Directory	C:\input\stage
Error Directory	C:\input\error
Request encoding	utf-8
Message Content Handling Configuration	
Content Streaming	Disabled

11. Review all the information for your Proxy Service and click **Save**.

Configuring a Pipeline

This section describes how to configure a Pipeline using the Oracle Service Bus Console.

1. Click the **Edit Message Flow** icon in the row of the Proxy Service you created.



Resources

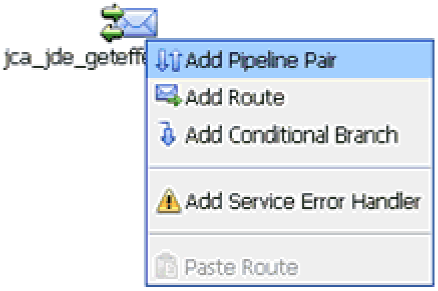
Create Resource: Select Resource Type

Items 1-3 of 3

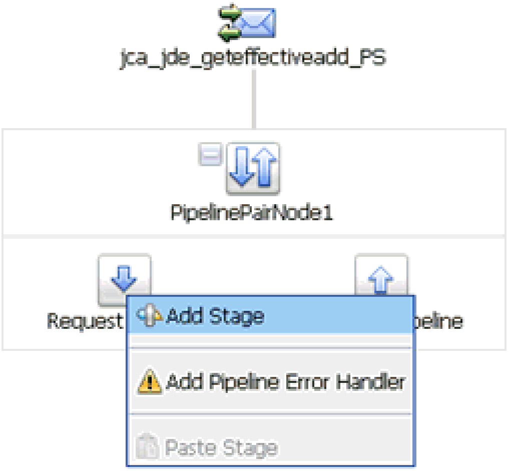
□	Name	Resource Type	Actions
□	ibse_jde_geteffective_BS	Business Service	
□	ibse_jde_geteffective_PS	Proxy Service	
□	jca_jde_geteffectiveadd_PS	Proxy Service	

The Edit Message Flow workspace area is displayed.

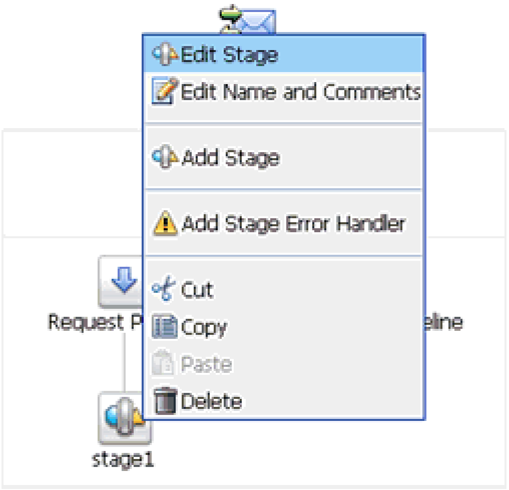
2. Click the **Proxy Service** icon and select **Add Pipeline Pair** from the context menu, as shown in the following image.



3. Click the **Request Pipeline** icon and select **Add Stage** from the context menu.

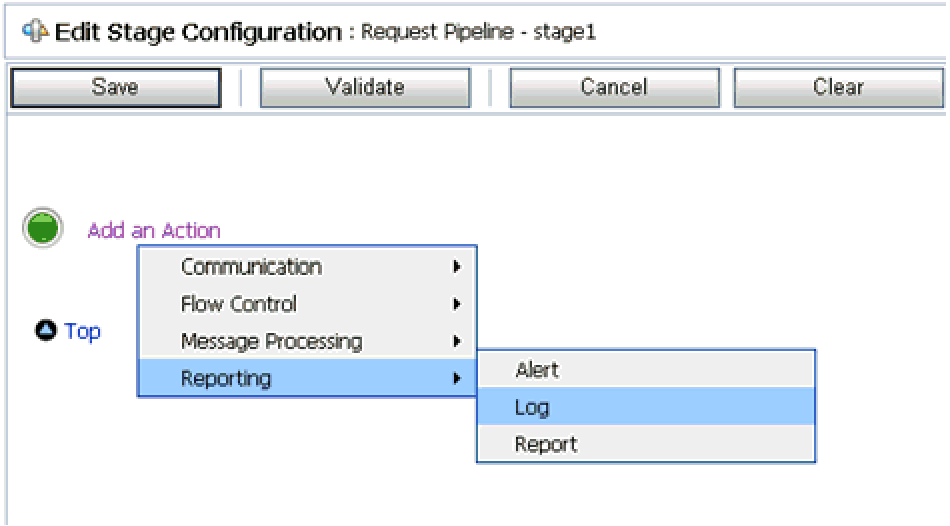


The Stage1 icon is added below the Request Pipeline icon.

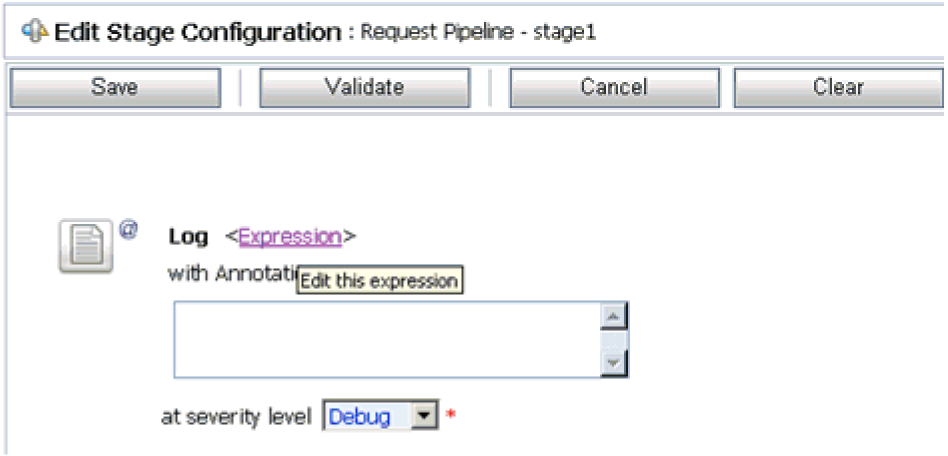


4. Click the **Stage1** icon and select **Edit Stage** from the context menu.

The Edit Stage Configuration workspace area is displayed.



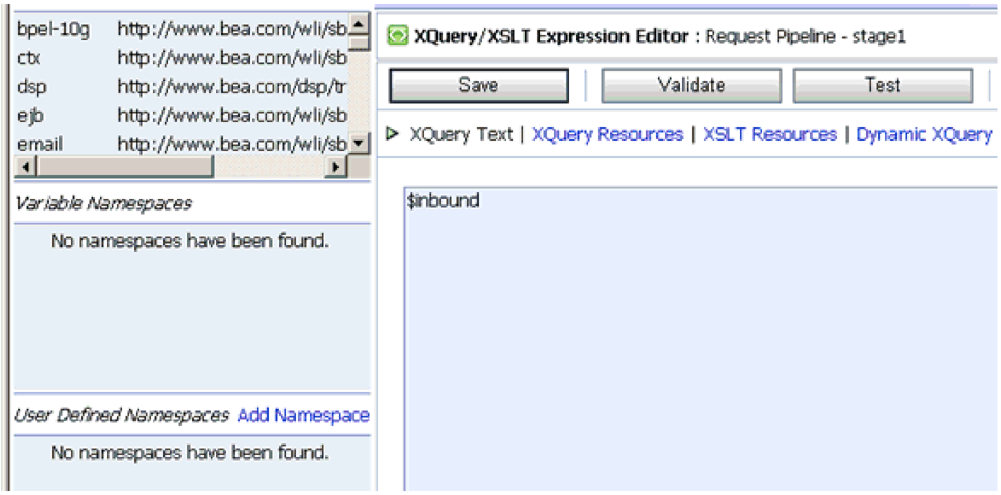
5. Click **Add an Action**, select **Reporting** from the context menu, and click **Log**.



at severity level Debug

6. Click **<Expression>** to edit the expression.

The XQuery/XSLT Expression Editor is displayed.


XQuery/XSLT Expression Editor : Request Pipeline - stage1
Save
Validate
Test
XQuery Text | XQuery Resources | XSLT Resources | Dynamic XQuery
Variable Namespaces
User Defined Namespaces Add Namespace

bpel-10g	http://www.bea.com/wli/sb
ctx	http://www.bea.com/wli/sb
dsp	http://www.bea.com/dsp/tr
ejb	http://www.bea.com/wli/sb
email	http://www.bea.com/wli/sb

No namespaces have been found.

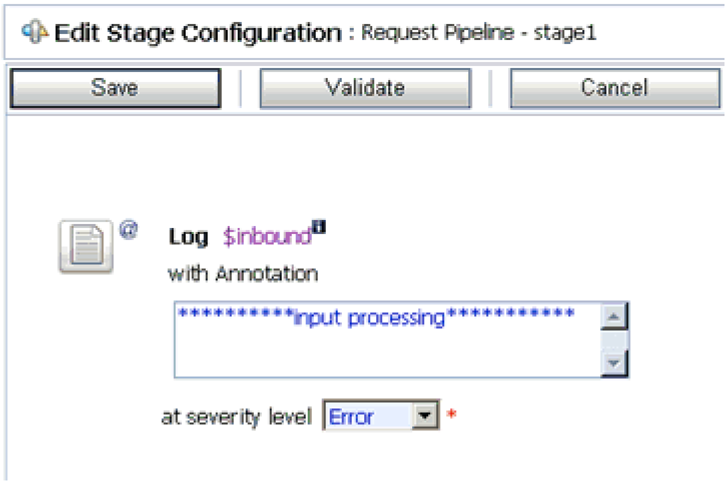
No namespaces have been found.

$inbound

7. In the XQuery Text area, type **$inbound**.

8. Click **Validate** and then **Save**.

   You are returned to the Edit Stage Configuration workspace area.



Edit Stage Configuration : Request Pipeline - stage1

Log $inbound

with Annotation

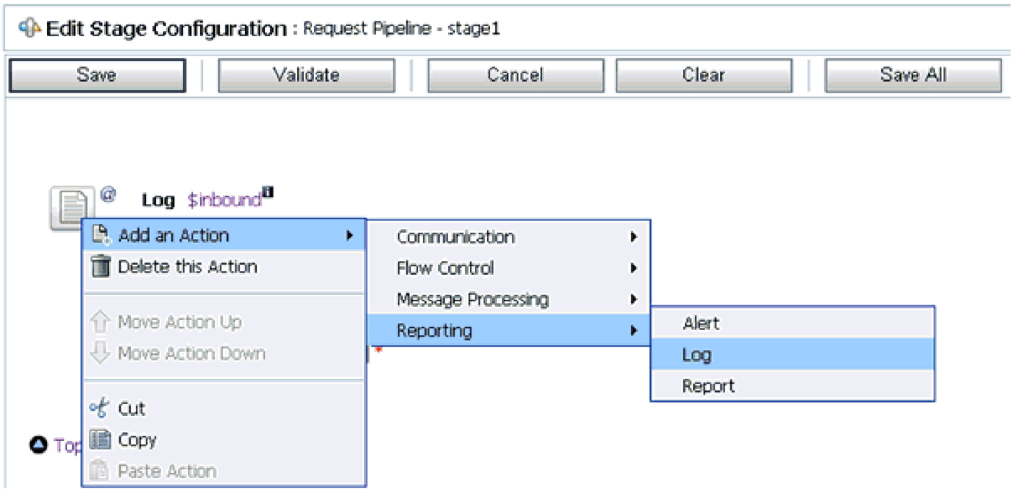
**********input processing***********

at severity level Error *

9. Type any annotation/comments in the text box (for example, **********input processing***********).

10. Select **Error** from the severity level drop-down list.

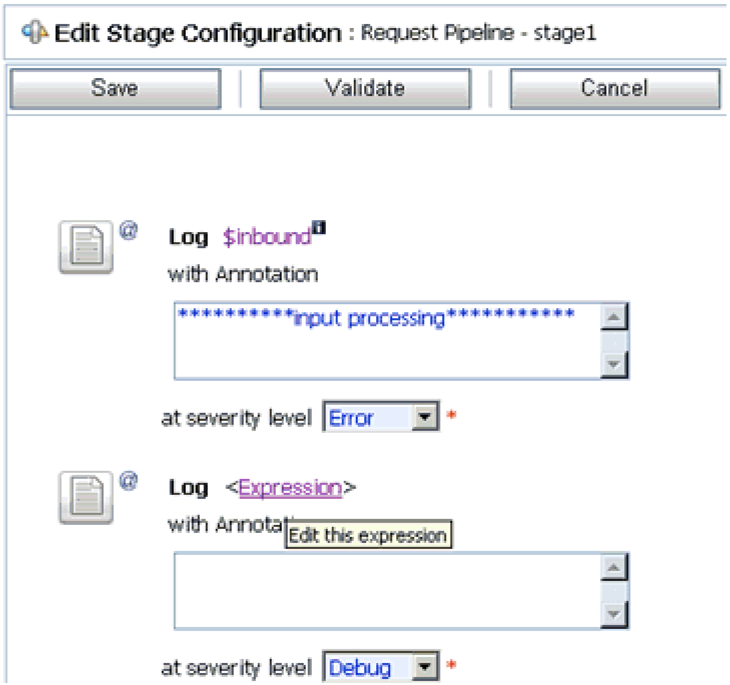
11. Add one more Log action as shown in the following image.



Edit Stage Configuration : Request Pipeline - stage1

Log $inbound

A new Log configuration is added, as shown in the following image.



Edit Stage Configuration : Request Pipeline - stage1

Log $inbound

with Annotation

**********input processing***********

at severity level Error *

Log <Expression>

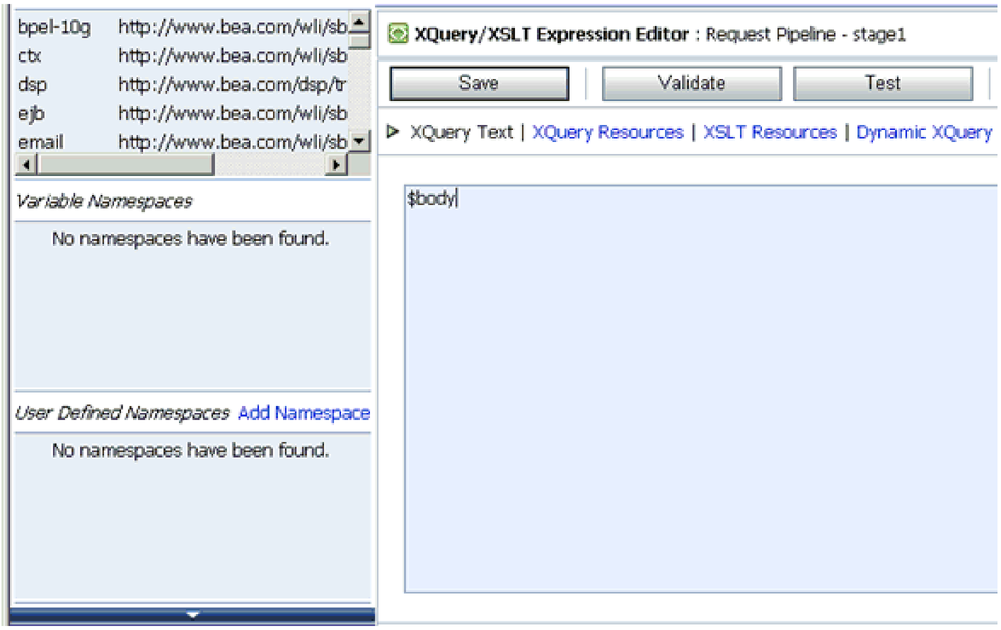
with Annotat

Edit this expression

at severity level Debug *

12. Click **<Expression>** to edit the expression.

    The XQuery/XSLT Expression Editor is displayed.



bpel-10g	http://www.bea.com/wli/sb
ctx	http://www.bea.com/wli/sb
dsp	http://www.bea.com/dsp/tr
ejb	http://www.bea.com/wli/sb
email	http://www.bea.com/wli/sb

Variable Namespaces

No namespaces have been found.

User Defined Namespaces Add Namespace

No namespaces have been found.

XQuery/XSLT Expression Editor : Request Pipeline - stage1

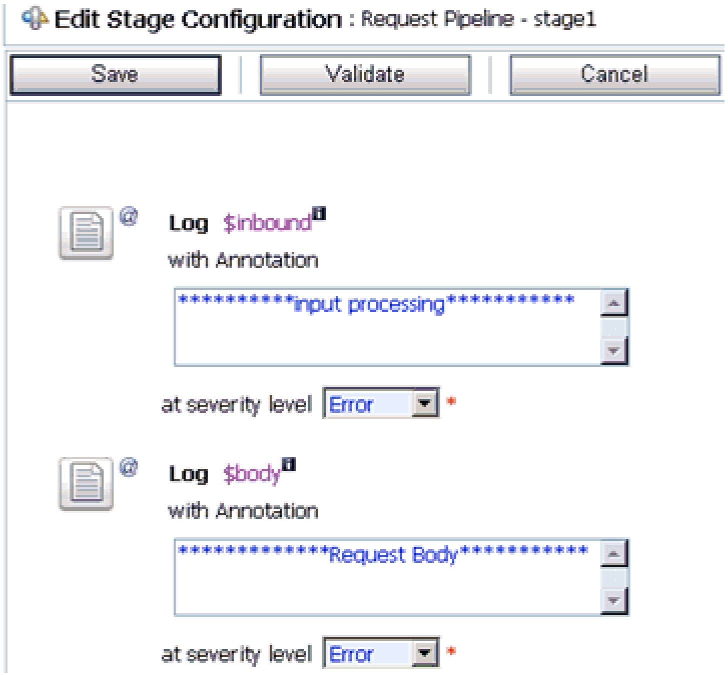
Save
Validate
Test
XQuery Text | XQuery Resources | XSLT Resources | Dynamic XQuery

$body

13. In the XQuery Text area, type **$body**.

14. Click **Validate** and then **Save**.

    You are returned to the Edit Stage Configuration workspace area.



Edit Stage Configuration : Request Pipeline - stage1

Save | Validate | Cancel

Log $inbound

with Annotation

**********input processing***********

at severity level Error *

Log $body

with Annotation

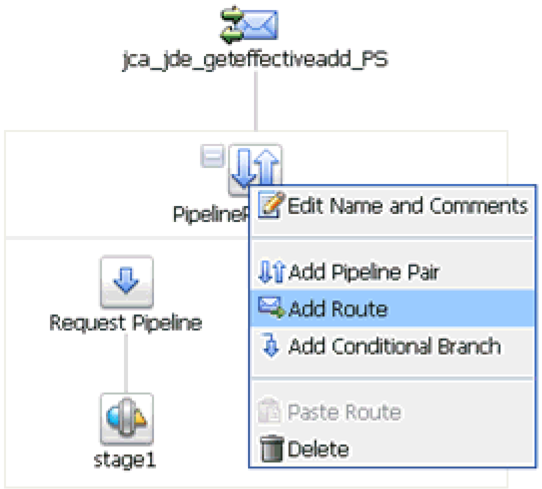
*************Request Body***********

at severity level Error *

15. Type any annotation/comments in the text box (for example, **********Request Body**********).

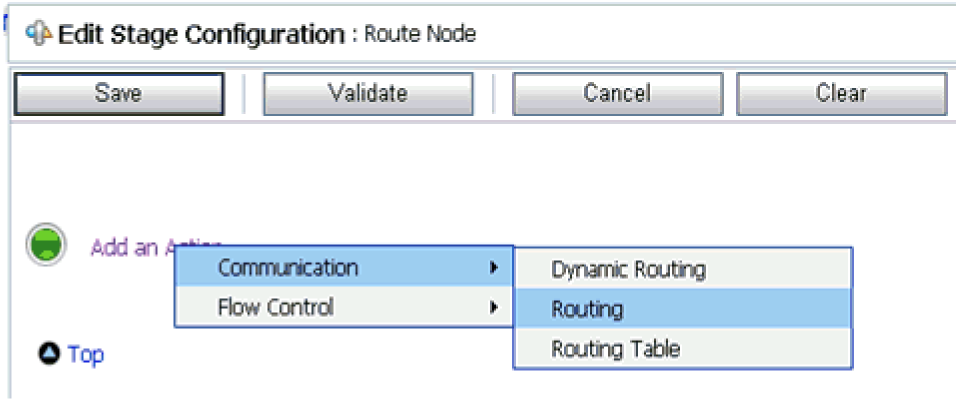
16. Select **Error** from the severity level drop-down list.

17. Click **Validate** and then **Save**.

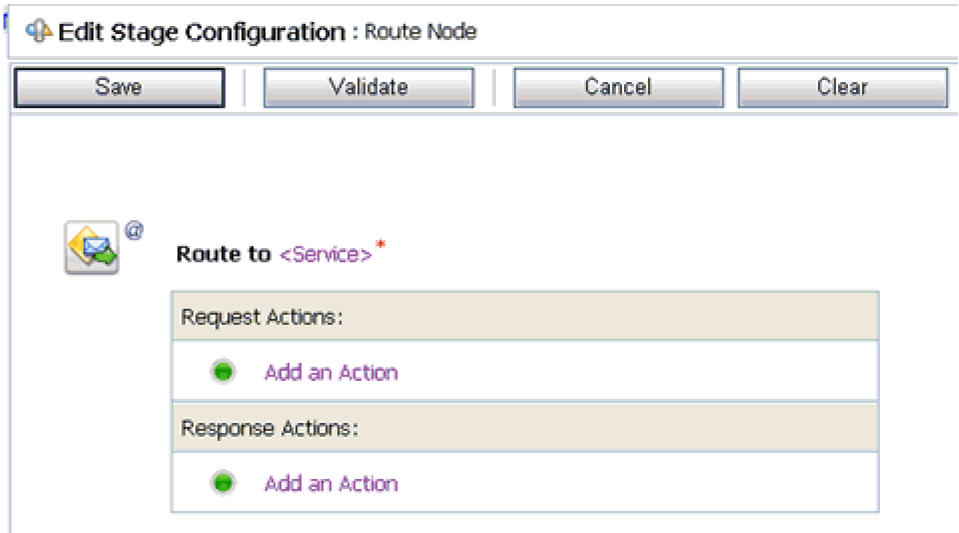


18. Click the **PipelinePairNode1** icon and select **Add Route** from the context menu.

19. Click the **RouteNode1** icon and select **Edit Route** from the context menu.

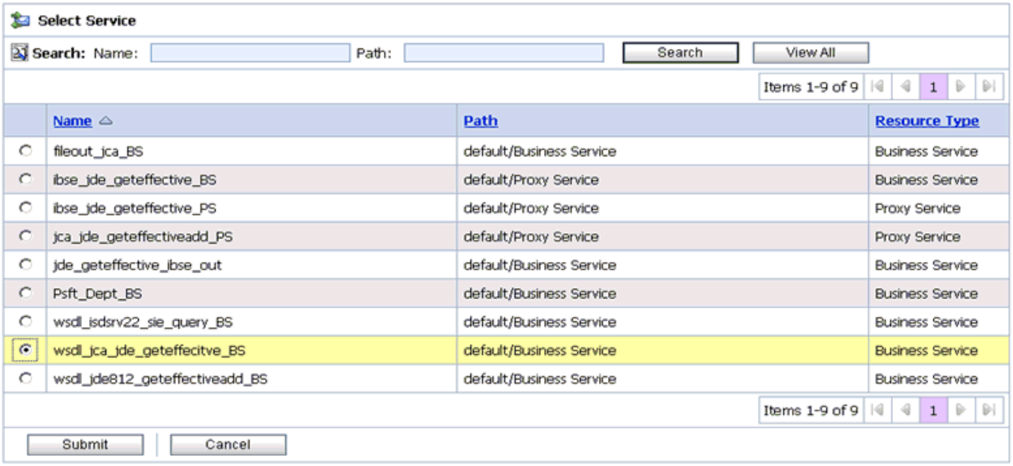


20. Click **Add an Action**, select **Communication** from the context menu, and click **Routing**.



21. Click **<Service>**.

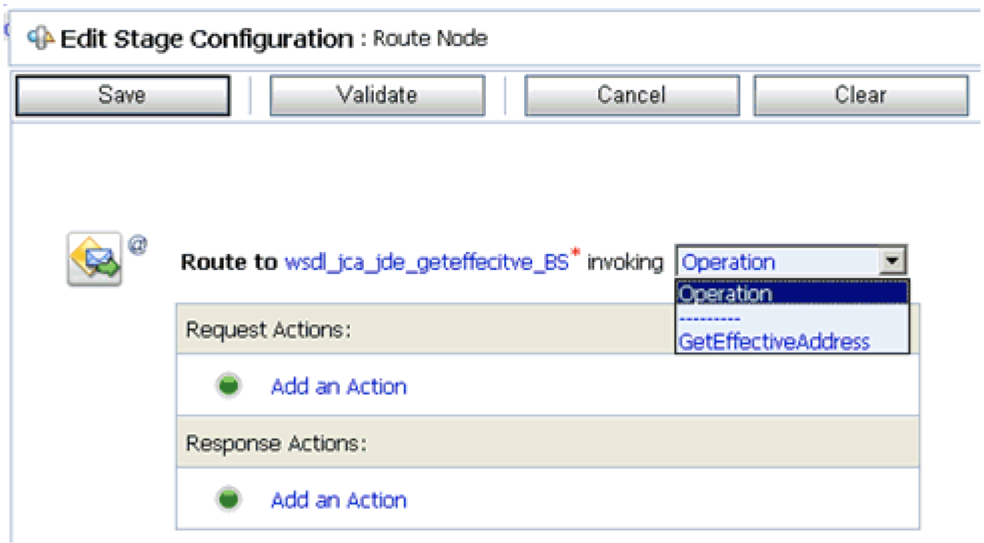
The Select Service dialog box is displayed.



	Name	Path	Resource Type
○	fileout_jca_BS	default/Business Service	Business Service
○	ibse_jde_geteffective_BS	default/Proxy Service	Business Service
○	ibse_jde_geteffective_PS	default/Proxy Service	Proxy Service
○	jca_jde_geteffectiveadd_PS	default/Proxy Service	Proxy Service
○	jde_geteffective_ibse_out	default/Business Service	Business Service
○	Psft_Dept_BS	default/Business Service	Business Service
○	wsdl_isdsrv22_sie_query_BS	default/Business Service	Business Service
●	wsdl_jca_jde_geteffecitve_BS	default/Business Service	Business Service
○	wsdl_jde812_geteffectiveadd_BS	default/Business Service	Business Service

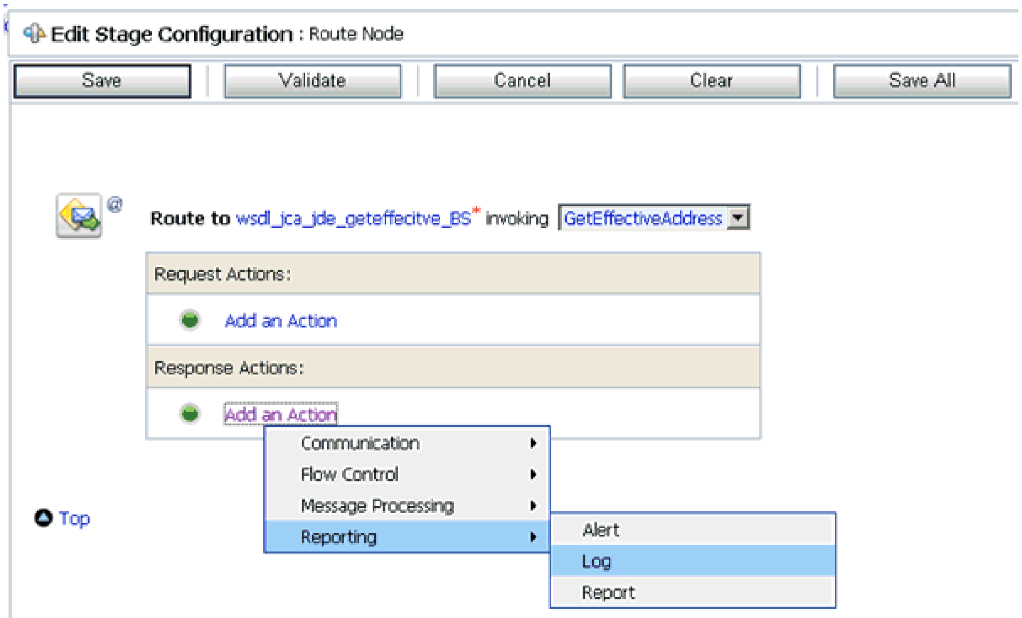
22. Select a WSDL type Business Service and click **Submit**.

You are returned to the Edit Stage Configuration workspace area.

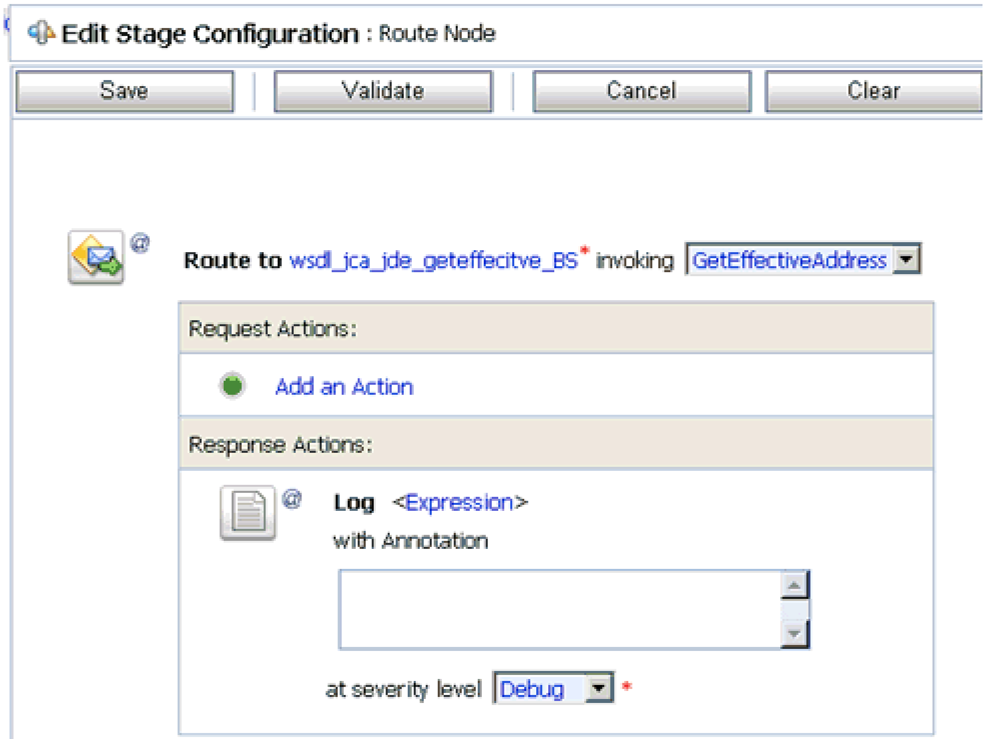


23. Select **GetEffectiveAddress** as the operational attribute from the drop-down list.

24. Click **Validate** and then **Save**.



25. In the Response Actions area, click **Add an Action**, select **Reporting** from the context menu, and click **Log**.



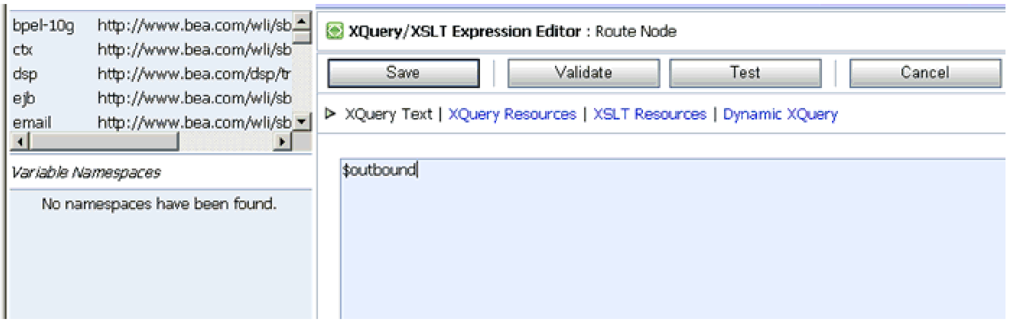
Route to wsdl_jca_jde_geteffecitve_BS* invoking GetEffectiveAddress

Log <Expression>
with Annotation

at severity level Debug *

26. Click **<Expression>** to edit the expression.

    The XQuery/XSLT Expression Editor is displayed.


XQuery/XSLT Expression Editor : Route Node
Save
Validate
Test
Cancel
XQuery Text | XQuery Resources | XSLT Resources | Dynamic XQuery
Variable Namespaces

bpel-10g	http://www.bea.com/wli/sb
ctx	http://www.bea.com/wli/sb
dsp	http://www.bea.com/dsp/tr
ejb	http://www.bea.com/wli/sb
email	http://www.bea.com/wli/sb

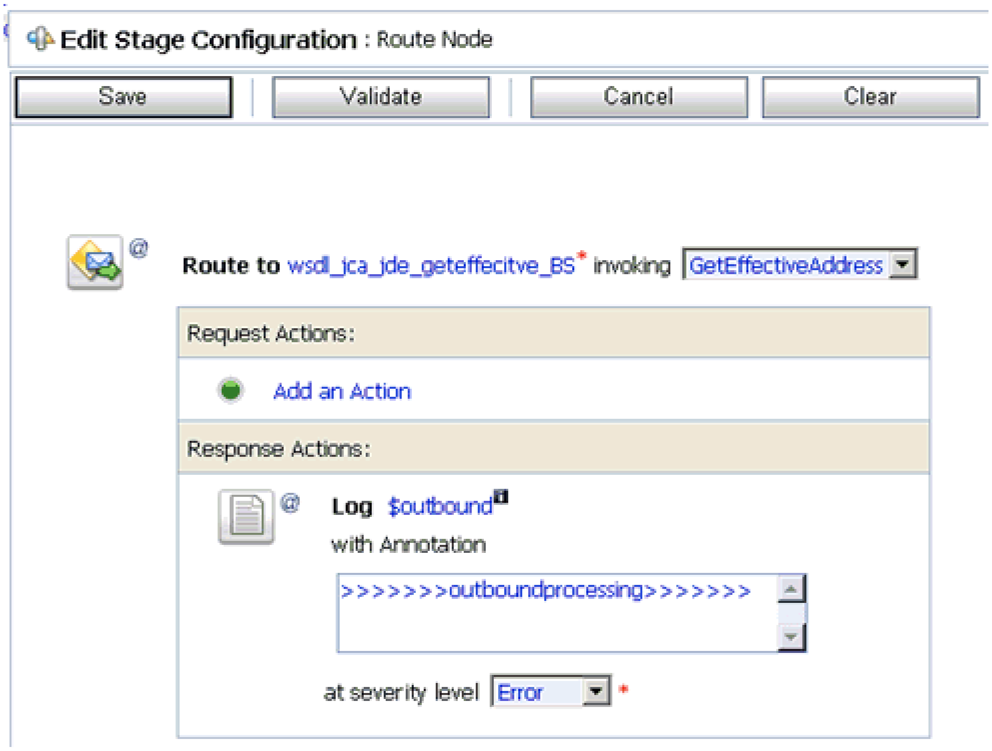
No namespaces have been found.

```
$outbound
```

27. In the XQuery Text area, type **$outbound**.

28. Click **Validate** and then **Save**.

    You are returned to the Edit Stage Configuration workspace area.

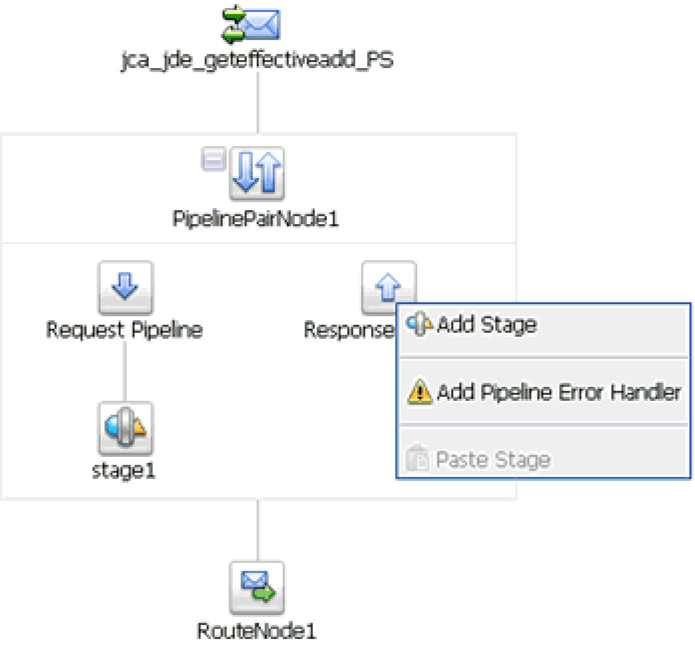


Edit Stage Configuration : Route Node
Save
Validate
Cancel
Clear
Route to wsdl_jca_jde_getefffecitve_BS* invoking GetEffectiveAddress
Request Actions:
Add an Action
Response Actions:
Log $outbound
with Annotation
>>>>>>>outboundprocessing>>>>>>>
at severity level Error *

29. Type any annotation/comments in the text box (for example, **********output processing***********).

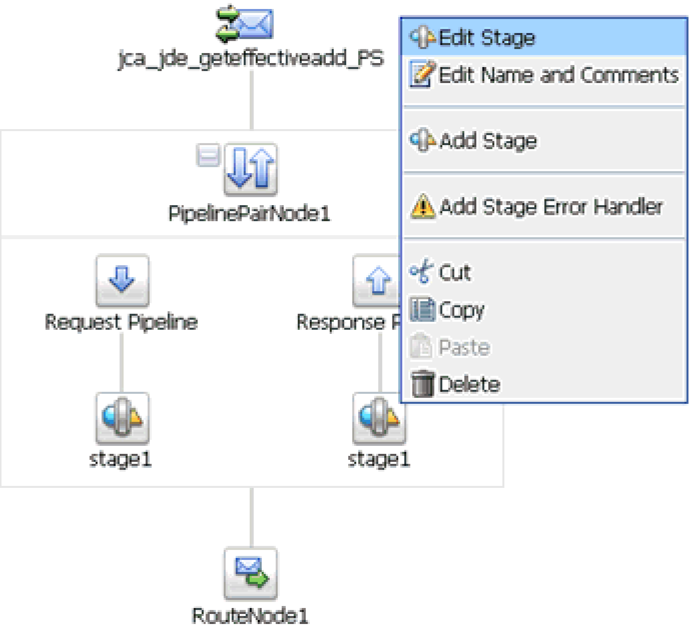
30. Select **Error** from the severity level drop-down list.

31. Click **Validate** and then **Save**.



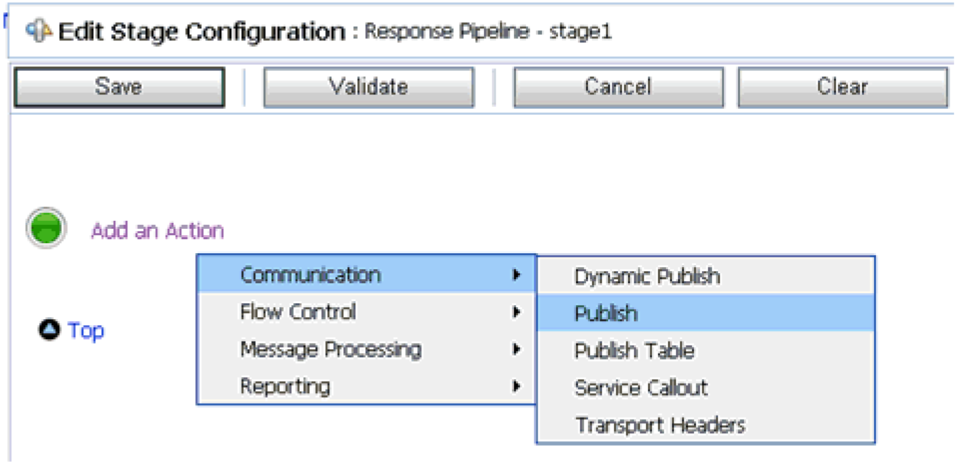
32. Click the **Response Pipeline** icon and select **Add Stage** from the context menu.

The Stage1 icon is added below the Response Pipeline icon.

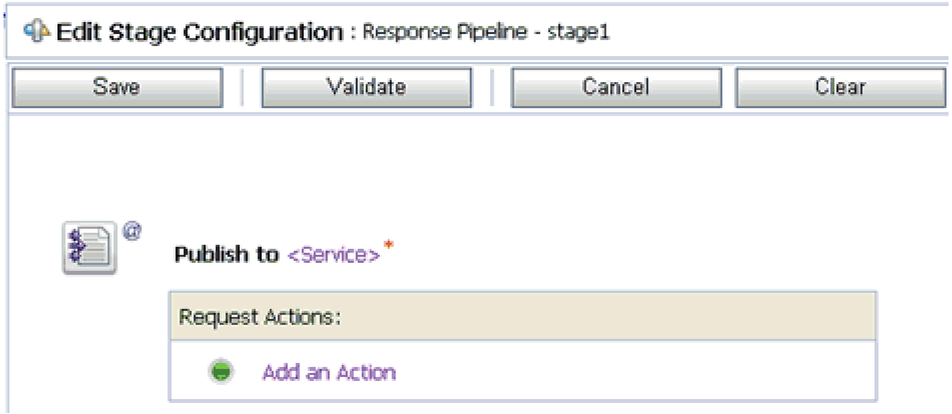


33. Click the **Stage1** icon and select **Edit Stage** from the context menu.

    The Edit Stage Configuration workspace area is displayed.

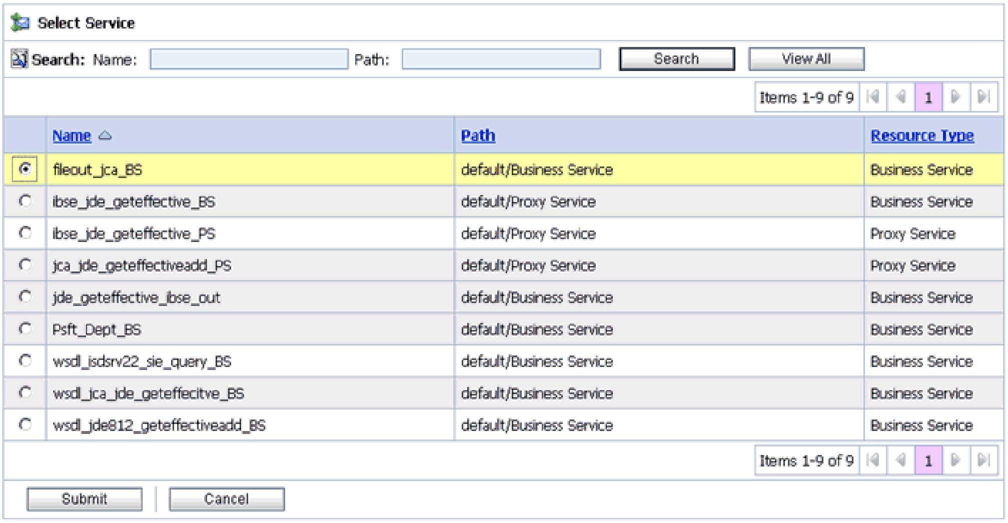


34. Click **Add an Action**, select **Communication** from the context menu, and click **Publish**.



35. Click **<Service>**.

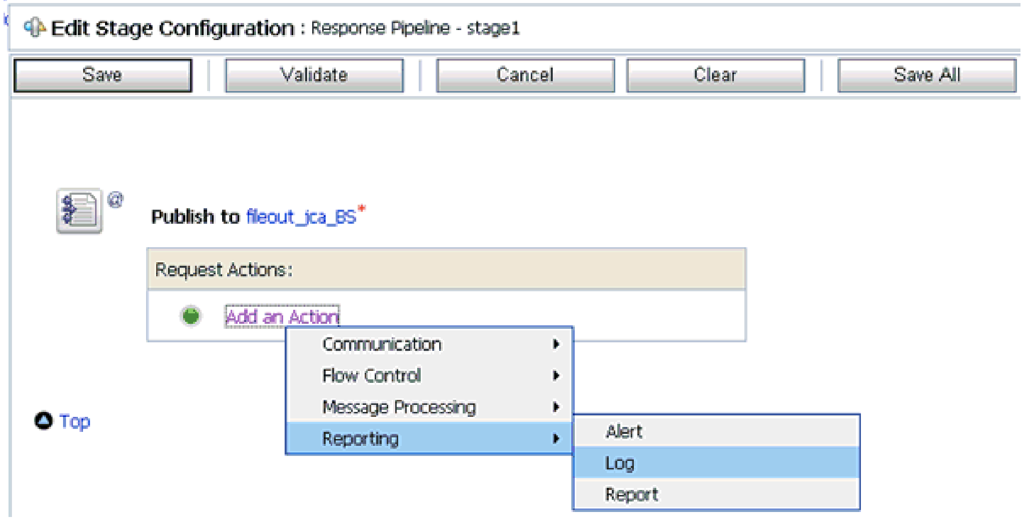
The Select Service dialog box is displayed.



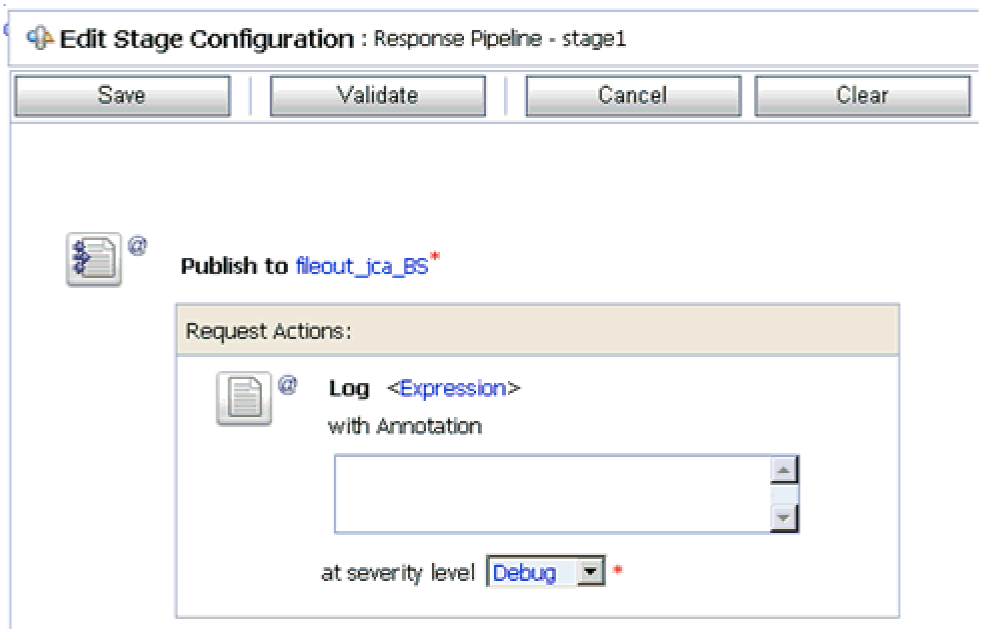
	Name	Path	Resource Type
●	fileout_jca_BS	default/Business Service	Business Service
○	ibse_jde_geteffective_BS	default/Proxy Service	Business Service
○	ibse_jde_geteffective_PS	default/Proxy Service	Proxy Service
○	jca_jde_geteffectiveadd_PS	default/Proxy Service	Proxy Service
○	jde_geteffective_ibse_out	default/Business Service	Business Service
○	Psft_Dept_BS	default/Business Service	Business Service
○	wsdl_isdsrv22_sie_query_BS	default/Business Service	Business Service
○	wsdl_jca_jde_geteffecitve_BS	default/Business Service	Business Service
○	wsdl_jde812_geteffectiveadd_BS	default/Business Service	Business Service

36. Select a File type Business Service and click **Submit**.

You are returned to the Edit Stage Configuration workspace area.

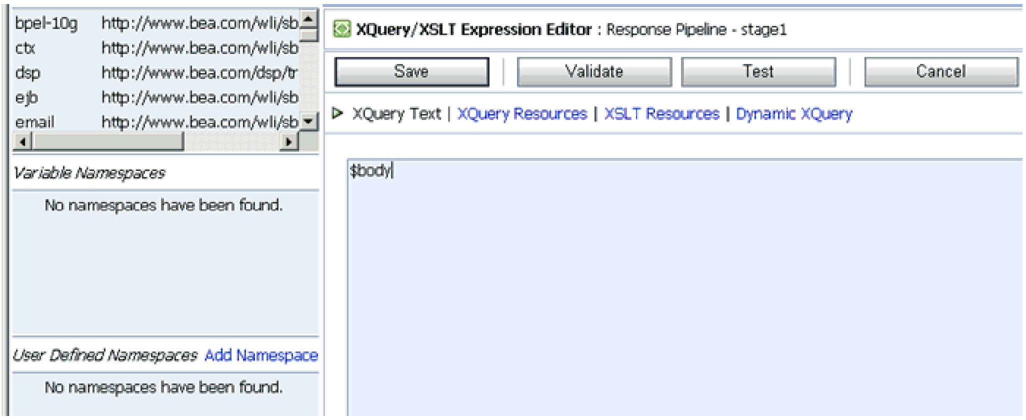


37. In the Request Actions area, click **Add an Action**, select **Reporting** from the context menu, and click **Log**.


Edit Stage Configuration : Response Pipeline - stage1
Save
Validate
Cancel
Clear
Publish to fileout_jca_BS*
Request Actions:
Log <Expression>
with Annotation
at severity level Debug *

38. Click **<Expression>** to edit the expression.

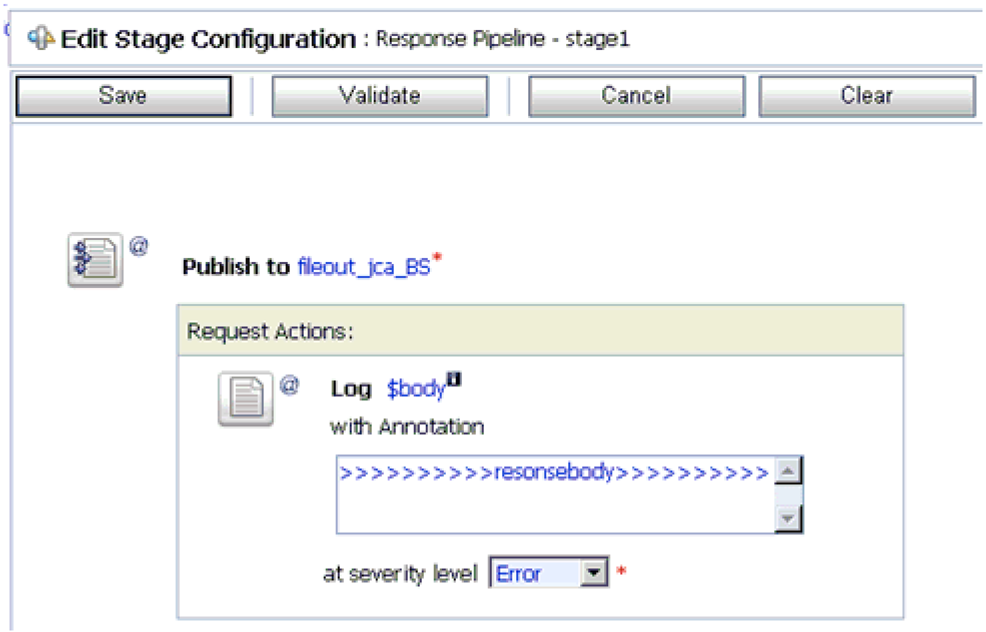
The XQuery/XSLT Expression Editor is displayed.


bpel-10g http://www.bea.com/wli/sb
ctx http://www.bea.com/wli/sb
dsp http://www.bea.com/dsp/tr
ejb http://www.bea.com/wli/sb
email http://www.bea.com/wli/sb
Variable Namespaces
No namespaces have been found.
User Defined Namespaces Add Namespace
No namespaces have been found.
XQuery/XSLT Expression Editor : Response Pipeline - stage1
Save
Validate
Test
Cancel
XQuery Text | XQuery Resources | XSLT Resources | Dynamic XQuery
$body

39. In the XQuery Text area, type **$body**.

40. Click **Validate** and then **Save**.

You are returned to the Edit Stage Configuration workspace area.



Publish to fileout_jca_BS*

Request Actions:

Log $body

with Annotation

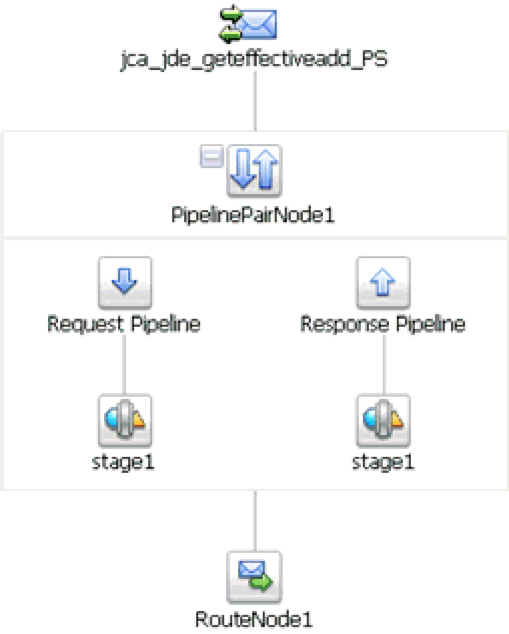
>>>>>>>>>>resonsebody>>>>>>>>>>

at severity level Error *

41. Type any annotation/comments in the text box (for example, **********Response Body***********).

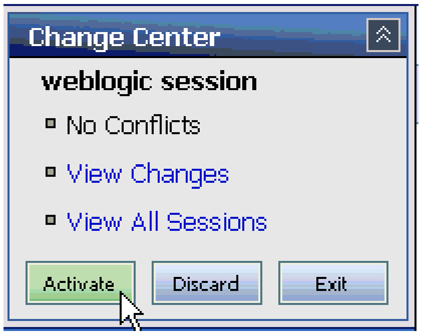
42. Select **Error** from the severity level drop-down list.

43. Click **Validate** and then **Save**.

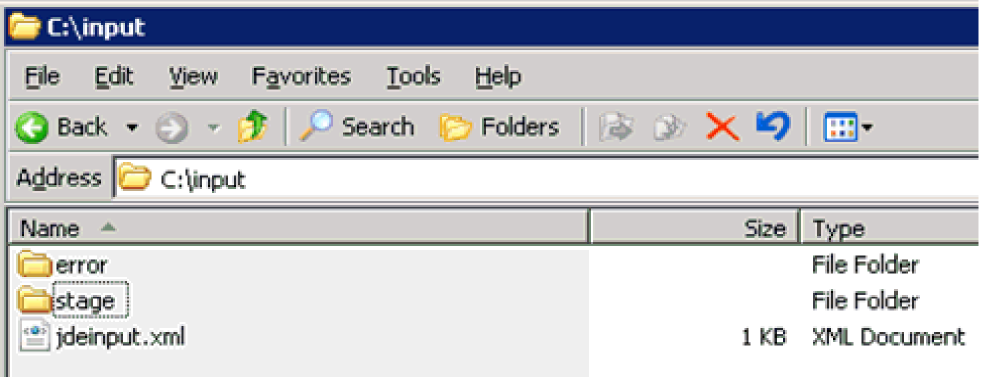


44. Click **Save**.

45. Click **Activate** in the Change Center area to activate your changes in the Oracle Service Bus session.

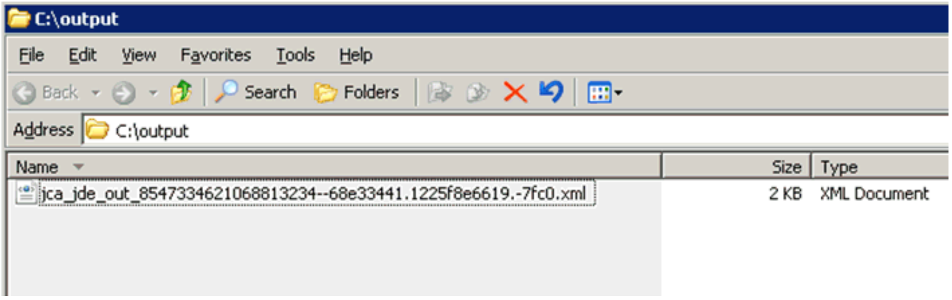


46. Copy and paste an input XML file in the input folder you have configured.



Name	Size	Type
error		File Folder
stage		File Folder
jdeinput.xml	1 KB	XML Document

An output XML file is received in the destination folder.



Name	Size	Type
jca_jde_out_8547334621068813234--68e33441.1225f8e6619.-7fc0.xml	2 KB	XML Document

6

Troubleshooting and Error Messages

This chapter explains the limitations and workarounds when connecting to J.D. Edwards OneWorld. 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 OracleWLS Adapter J2CA or with a OracleWLS Adapter Business Services Engine (BSE) configuration.

Troubleshooting

This topic provides troubleshooting information for J.D. Edwards OneWorld, separated into four categories:

- OracleWLS Adapter Application Explorer (Application Explorer)
- J.D. Edwards OneWorld
- OracleWLS Adapter J2CA
- BSE

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

- The OracleWLS Adapter J2CA trace information can be found under the `wls_home\erp-adapters\config\config_name\log` directory.
- BSE trace information can be found under the `wls_home\erp-adapters\ibse.war\ibselogs` directory.
- The log file for Application Explorer can be found under the `wls_home\erp-adapters\tools\iwae\bin` directory.

Application Explorer

To use Application Explorer on **Windows** for debugging or testing purposes, load the batch script `ae.bat`, found under:

```
wls_home\erp-adapters\tools\iwae\bin
```

On **UNIX**, load the shell script `iwae.sh`, found under:

```
wls_home/erp-adapters/tools/iwae/bin
```

Error	Solution
Cannot connect to OracleWLS Application Adapter for J.D. Edwards OneWorld from Application Explorer: `Problem activating adapter. (Failed to connect to J.D.Edwards, check system availability and configuration parameters:...) Check logs for more information.`	Ensure that: ■ J.D. Edwards OneWorld is running. ■ The J.D. Edwards OneWorld user ID and password is correct. ■ The port number is correct.
The following error message appears: java.lang.IllegalStateException: java.lang.Exception: Error Logon to J.D. Edwards OneWorld System	You have provided invalid connection information for J.D. Edwards OneWorld or the wrong JAR file is in the lib directory.
J.D. Edwards OneWorld does not appear in the Application Explorer Adapter node list.	Ensure that the J.D. Edwards OneWorld JAR files, are added to the lib directory.
Logon failure error at run-time.	If the password for connecting to your J.D. Edwards OneWorld system is not specified when creating a target or with the Edit option in Application Explorer, you will be unable to connect to J.D. Edwards OneWorld. The connection password is not saved in `repository.xml`. Update the password using the Edit option in Application Explorer, then restart the application server.
The following exception occurs when you start Application Explorer by activating `ae.bat` (not `iaexplorer.exe`): `java.lang.ClassNotFoundException: org.bouncycastle.jce.provider.BouncyCastleProvider`	This is a benign exception. It does not affect adapter functionality. Download BouncyCastle files from: `ftp://ftp.bouncycastle.org/pub`

Unable to start Application Explorer in a Solaris environment. The following exception is thrown in the console:
`javax.resource.ResourceException: IWAFManagedConnectionFactory: License violation.at com.ibi.afjca.spi.IWAFManagedConnectionFactory.createConnectionFactory(IWAFManagedConnectionFactory.java:98)at com.iwaysoftware.iwae.common.JCATransport.getConnectionFactory(JCATransport.java:133) at com.iwaysoftware.iwae.common.JCATransport.initJCA(JCATransport.java:69)at com.iwaysoftware.iwae.common.JCATransport.<init>(JCATransport.java:62)at com.iwaysoftware.iwae.common.AdapterClient.<init>(AdapterClient.java:85)at com.ibi.bse.ConfigWorker.run(ConfigWorker.java:41)at java.lang.Thread.run(Thread.java:534)`
`Could not create the connection factory.`
	`JAVACMD` is not set on the user system. Before starting Application Explorer, export `JAVACMD` as follows:
	`JAVACMD=/<jdk_home>/bin/java`, where `<jdk_home>` is the directory where JDK is installed on your system.

J.D. Edwards OneWorld

Error	Cause	Solution
Action code invalid	In the Sales Order request, the Action code appears as "H," an invalid action code.	Use: ■ "I" for inquiry. ■ "C" for change. ■ "D" for delete. ■ "A" to add a new record.
Invalid address number.	The address number does not exist in the Address Book Master file (F0101).	Enter an address number using the Address Book Revisions program (PO1051). Ensure that the number entered is correct.
Record invalid	The record being processed either already exists for an ADD function or does not exist for an INQUIRY, CHANGE, or DELETE function.	If you are attempting to inquire, change, or delete a record you added previously, there could be database problems in your production library. Contact your data processing department.
Item Branch record does not exist.	An Item Branch record (F4102) does not exist for this item in the Branch/Plant specified.	Correct the Branch or enter an Item Branch record for this item in Branch Plant Item Information (P41026).
&1 does not match any of the valid values.	The &1 does not match any of the valid values specified in the Data Dictionary for this field.	Enter a valid value.
Date out of range.	The Last Service Date and the Inspection Date must be within the range of the effective dates of the Service Contract.	Change the date to be greater than or equal to the beginning effective date and less than or equal to the ending effective date of the Service Contract.
Jde.net timeout exception	Net timeout is set to a wrong value	Verify that net timeout is set to 180 at `jde.ini` of [NETWORK QUEUE SETTINGS], for example `JDENETTimeout=180`
Cannot connect to EnterpriseOne Version 8.10	Missing required library files	`Kernel.jar` and `Connector.jar` are required for version B7333. `jdeutil.jar` and `log4j.jar` are required for EnterpriseOne Version 8.10, in addition to `Kernel.jar` and `Connector.jar`.

OracleWLS Adapter J2CA

Error	Solution
In Application Explorer, the following error message appears when you attempt to connect to an OracleWLS Adapter J2CA configuration: `Could not initialize JCA`	In the Details tab in the right pane, ensure that the directory specified in the Home field points to the correct directory, for example, *wls_home*`\erp-adapters\tools\iwae\bin\..\..\..\`

BSE Error Messages

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

General Error Handling in BSE

BSE serves as both a SOAP gateway into the adapter framework and as the engine for some of the adapters. In both design time and run-time, various conditions can cause errors in BSE when Web services that use adapters run. Some of these conditions and resulting errors are exposed the same way, regardless of the specific adapter; others are exposed differently, based on the adapter being used. This topic explains what you can expect when you encounter some of the more common error conditions on an adapter-specific basis.Usually the SOAP gateway (agent) inside BSE passes a SOAP request message to the adapter required for the Web service. If an error occurs, 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.

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.

Invalid SOAP Request

If Oracle WebLogic Server Adapter receives a SOAP request message that does not conform to the WSDL for the Web services being executed, then the following SOAP response is generated.

```
<?xml version="1.0" encoding="ISO-8859-1"
 ?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Body>
 <SOAP-ENV:Fault>
  <faultcode>SOAP-ENV:Server</faultcode>
  <faultstring>RPC server connection failed: Connection refused:
```

```
connect</faultstring>
 </SOAP-ENV:Fault>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Empty Result From Oracle WebLogic Server Adapter Request

If Oracle WebLogic Server Adapter executes a SOAP request using input parameters passed that do not match records in the target system, then the following SOAP response is generated.

Note: The condition for this adapter does not yield a SOAP fault.

```
<SOAP-ENV:Envelope xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/1999/XMLSchema">
   <SOAP-ENV:Body>
      <m:RunDBQueryResponse xmlns:m="urn:schemas-iwaysoftware-com:iwse"
        xmlns="urn:schemas-iwaysoftware-com:iwse"
        cid="2A3CB42703EB20203F91951B89F3C5AF">
        <RunDBQueryResult run="1" />
     </m:RunDBQueryResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Error Logging In

If Oracle WebLogic Server Adapter executes an invalid SOAP log in request, then the following SOAP response is generated.

```
[2004-07-19T16:28:56:718Z] DEBUG (SOAP1) W.SOAP1.2: POST received
[2004-07-19T16:28:56:718Z] DEBUG (SOAP1) W.SOAP1.2: in XDSOAPHTTPWorker agentName
is [XDSOAPRouter]
[2004-07-19T16:28:56:718Z] DEBUG (SOAP1) W.SOAP1.2: before parse:
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<SOAP-ENV:Header>
<m:ibsinfo xmlns:m="urn:schemas-iwaysoftware-...[861]
[2004-07-19T16:28:56:718Z] ERROR (SOAP1) W.SOAP1.2: Attempting string, no encoding
recognized in document
[2004-07-19T16:28:56:734Z] DEEP (SOAP1) W.SOAP1.2: parse complete in 16 msecs
[2004-07-19T16:28:56:859Z] DEEP (SOAP1) W.SOAP1.2: ST_NODICT
[2004-07-19T16:28:56:859Z] DEEP (SOAP1) W.SOAP1.2: ST_FINISH
[2004-07-19T16:28:56:859Z] DEBUG (SOAP1) extractControl - edaDoc: false
[2004-07-19T16:28:56:859Z] DEBUG (SOAP1) now: 2004-07-19T16:28:56Z expires:
2004-07-20T16:28:56Z
[2004-07-19T16:28:56:859Z] DEBUG (SOAP1) W.SOAP1.2: checking for cached agent
[2004-07-19T16:28:56:859Z] DEBUG (SOAP1) W.SOAP1.2: pushagent: adding agent
com.ibi.iwse.XDSOAPRouter
[2004-07-19T16:28:56:875Z] DEBUG (SOAP1) W.SOAP1.2: inside worker the soap Action
is [B0100033.GetEffectiveAddressRequest#test##]
[2004-07-19T16:28:56:875Z] DEBUG (SOAP1) W.SOAP1.2: precedence: 1
[2004-07-19T16:28:56:875Z] DEBUG (SOAP1) W.SOAP1.2: precedence: 1
[2004-07-19T16:28:56:875Z] DEBUG (SOAP1) W.SOAP1.2: numagents: 1
[2004-07-19T16:28:56:890Z] DEBUG (SOAP1) W.SOAP1.2: running agent 1 name
com.ibi.iwse.XDSOAPRouter document 1
[2004-07-19T16:28:56:890Z] INFO  (manager) MGR00X01: Adding active worker:
```

```
W.SOAP1.2
[2004-07-19T16:28:56:890Z] DEBUG (SOAP1) W.SOAP1.2: <?xml version="1.0"
encoding="UTF-8" ?>
<SOAP-ENV:Envelope xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <SOAP-ENV:Header>
      <m:ibsinfo xmlns:m="urn:schemas-iwaysoftware-com:iwse">
         <m:service>B0100033</m:service>
         <m:method>GetEffectiveAddress</m:method>
         <m:license>test</m:license>
         <m:Username>user</m:Username>
         <m:Password>password</m:Password>
      </m:ibsinfo>
   </SOAP-ENV:Header>
   <SOAP-ENV:Body>
      <m:GetEffectiveAddress
xmlns:m="urn:iwaysoftware:ibse:jul2003:GetEffectiveAddress">
         <m:jdeRequest type="callmethod">
            <m:callMethod name="GetEffectiveAddress">
               <m:params>
                  <m:param name="mnAddressNumber">12345</m:param>
               </m:params>
               <m:onError/>
            </m:callMethod>
         </m:jdeRequest>
      </m:GetEffectiveAddress>
   </SOAP-ENV:Body>
   <SOAPAction agentName="XDSOAPRouter"
cid="1FF3D44E0B0AFB2A4E9538ED42B71437">B0100033.GetEffectiveAddressRequest#test##<
/SOAPAction>
</SOAP-ENV:Envelope>
[2004-07-19T16:28:56:890Z] DEBUG (SOAP1) W.SOAP1.2: business method:
m:GetEffectiveAddress
[2004-07-19T16:28:56:906Z] DEBUG (SOAP1) W.SOAP1.2: input:
[2004-07-19T16:28:56:906Z] DEBUG (SOAP1) W.SOAP1.2: <?xml version="1.0"
encoding="UTF-8" ?><jdeRequest xmlns:xsd="http://www.w3.org/2001/XMLSchema"
type="callmethod" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><callMethod
name="GetEffectiveAddress"><params><param name="mnAddressNumber">12345</param>
      </params><onError/></callMethod></jdeRequest>
[2004-07-19T16:28:58:234Z] DEBUG (SOAP1) W.SOAP1.2: Agent returned success
[2004-07-19T16:28:58:234Z] INFO  (manager) MGR00X02: Removing active worker:
W.SOAP1.2
[2004-07-19T16:28:58:234Z] DEBUG (SOAP1) W.SOAP1.2: doing docTran, docVal,
listTran for agent(1)
[2004-07-19T16:28:58:250Z] DEBUG (SOAP1) W.SOAP1.2: sendToAll reply to XDReply:
[protocol=http */null]
[2004-07-19T16:28:58:250Z] DEBUG (SOAP1) W.SOAP1.2: preemitters from doc: null
[2004-07-19T16:28:58:250Z] DEBUG (SOAP1) W.SOAP1.2: no preemitters, emitting
contents of doc, usestream=false encoding=UTF-8
[2004-07-19T16:28:58:250Z] DEBUG (SOAP1) W.SOAP1.2: writeEntity, len: 670 data:
<?xml version="1.0" encoding="UTF-8" ?><SOAP-ENV:Envelope
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><SOAP-ENV:Body><GetEffective
AddressResponse xmlns="urn:iwaysoftware:ibse:jul2003:GetEffectiveAddress:response"
cid="1FF3D44E0B0AFB2A4E9538ED42B71437"><jdeResponse user="USER" type="callmethod"
```

```
session="" environment="DV7333"><returnCode code="12">Environment
&apos;DV7333&apos; could not be initialized for user, check user, pwd and
environment attribute
values</returnCode></jdeResponse></GetEffectiveAddressResponse></SOAP-ENV:Body></S
OAP-ENV:Envelope>
[2004-07-19T16:28:58:250Z] DEBUG (SOAP1) W.SOAP1.2: writeString: HTTP/1.0
[2004-07-19T16:28:58:250Z] DEBUG (SOAP1) W.SOAP1.2: writeString: 200
[2004-07-19T16:28:58:250Z] DEBUG (SOAP1) W.SOAP1.2: writeString: OK
[2004-07-19T16:28:58:250Z] DEBUG (SOAP1) W.SOAP1.2: writeString: Content-Type:
[2004-07-19T16:28:58:250Z] DEBUG (SOAP1) W.SOAP1.2: writeString: text/xml
[2004-07-19T16:28:58:250Z] DEBUG (SOAP1) W.SOAP1.2: writeString: Content-Length:
[2004-07-19T16:28:58:265Z] DEBUG (SOAP1) W.SOAP1.2: writeString: 670
[2004-07-19T16:28:58:265Z] INFO  (SOAP1) W.SOAP1.2: W0000X13: Ended message
processing, rc=0
[2004-07-19T16:28:58:265Z] DEEP (SOAP1) W.SOAP1.2: storing used socket
[2004-07-19T16:28:58:265Z] DEBUG (SOAP1) W.SOAP1.2: entering waitforDocument
[2004-07-19T16:29:03:875Z] DEEP (SOAP1) W.SOAP1.2: cleanup: closing sockets(0)
```

Empty Result From Oracle WebLogic Server Adapter Request

If Oracle WebLogic Server Adapter executes a SOAP request using input parameters passed that do not match records in the target system, then the following SOAP response is generated.

Note: The condition for this adapter does not yield a SOAP fault.

```
[2004-07-19T16:27:05:640Z] DEBUG (SOAP1) W.SOAP1.2: POST received
[2004-07-19T16:27:05:640Z] DEBUG (SOAP1) W.SOAP1.2: in XDSOAPHTTPWorker agentName
is [XDSOAPRouter]
[2004-07-19T16:27:05:640Z] DEBUG (SOAP1) W.SOAP1.2: before parse:
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<SOAP-ENV:Body>
<m:GetEffectiveAddress xmlns:m="urn:iwaysoftwar...[590]
[2004-07-19T16:27:05:640Z] ERROR (SOAP1) W.SOAP1.2: Attempting string, no encoding
recognized in document
[2004-07-19T16:27:05:640Z] DEEP (SOAP1) W.SOAP1.2: parse complete in 0 msecs
[2004-07-19T16:27:05:781Z] DEEP (SOAP1) W.SOAP1.2: ST_NODICT
[2004-07-19T16:27:05:781Z] DEEP (SOAP1) W.SOAP1.2: ST_FINISH
[2004-07-19T16:27:05:781Z] DEBUG (SOAP1) extractControl - edaDoc: false
[2004-07-19T16:27:05:781Z] DEBUG (SOAP1) now: 2004-07-19T16:27:05Z expires:
2004-07-20T16:27:05Z
[2004-07-19T16:27:05:781Z] DEBUG (SOAP1) W.SOAP1.2: inside isAsync() the soap
Action is ["B0100033.GetEffectiveAddressRequest#test##"]
[2004-07-19T16:27:05:781Z] DEBUG (SOAP1) W.SOAP1.2: inside isAsync() the soap
Action is [B0100033.GetEffectiveAddressRequest#test##]
[2004-07-19T16:27:05:781Z] DEBUG (SOAP1) W.SOAP1.2: checking for cached agent
[2004-07-19T16:27:05:796Z] DEBUG (SOAP1) W.SOAP1.2: pushagent: adding agent
com.ibi.iwse.XDSOAPRouter
[2004-07-19T16:27:05:796Z] DEBUG (SOAP1) W.SOAP1.2: inside worker the soap Action
is [B0100033.GetEffectiveAddressRequest#test##]
[2004-07-19T16:27:05:796Z] DEBUG (SOAP1) W.SOAP1.2: precedence: 1
[2004-07-19T16:27:05:796Z] DEBUG (SOAP1) W.SOAP1.2: precedence: 1
[2004-07-19T16:27:05:796Z] DEBUG (SOAP1) W.SOAP1.2: numagents: 1
[2004-07-19T16:27:05:812Z] DEBUG (SOAP1) W.SOAP1.2: running agent 1 name
com.ibi.iwse.XDSOAPRouter document 1
[2004-07-19T16:27:05:812Z] INFO  (manager) MGR00X01: Adding active worker:
```

```
W.SOAP1.2
[2004-07-19T16:27:05:812Z] DEBUG (SOAP1) W.SOAP1.2: <?xml version="1.0"
encoding="UTF-8" ?>
<SOAP-ENV:Envelope xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <SOAP-ENV:Body>
      <m:GetEffectiveAddress
xmlns:m="urn:iwaysoftware:ibse:jul2003:GetEffectiveAddress">
         <m:jdeRequest type="callmethod">
            <m:callMethod name="GetEffectiveAddress">
               <m:params>
                  <m:param name="mnAddressNumber">12345</m:param>
               </m:params>
               <m:onError/>
            </m:callMethod>
         </m:jdeRequest>
      </m:GetEffectiveAddress>
   </SOAP-ENV:Body>
   <SOAPAction agentName="XDSOAPRouter"
cid="9F71FEA4C932CD8786F7388D7EF293A1">B0100033.GetEffectiveAddressRequest#test##<
/SOAPAction>
</SOAP-ENV:Envelope>
[2004-07-19T16:27:05:812Z] DEBUG (SOAP1) W.SOAP1.2: business method:
m:GetEffectiveAddress
[2004-07-19T16:27:05:828Z] DEBUG (SOAP1) W.SOAP1.2: input:
[2004-07-19T16:27:05:828Z] DEBUG (SOAP1) W.SOAP1.2: <?xml version="1.0"
encoding="UTF-8" ?><jdeRequest xmlns:xsd="http://www.w3.org/2001/XMLSchema"
type="callmethod" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><callMethod
name="GetEffectiveAddress"><params><param name="mnAddressNumber">12345</param>
</params><onError/></callMethod></jdeRequest>
[2004-07-19T16:27:07:843Z] DEBUG (SOAP1) W.SOAP1.2: Agent returned success
[2004-07-19T16:27:07:843Z] INFO  (manager) MGR00X02: Removing active worker:
W.SOAP1.2
[2004-07-19T16:27:07:843Z] DEBUG (SOAP1) W.SOAP1.2: doing docTran, docVal,
listTran for agent(1)
[2004-07-19T16:27:07:859Z] DEBUG (SOAP1) W.SOAP1.2: sendToAll reply to XDReply:
[protocol=http */null]
[2004-07-19T16:27:07:859Z] DEBUG (SOAP1) W.SOAP1.2: preemitters from doc: null
[2004-07-19T16:27:07:859Z] DEBUG (SOAP1) W.SOAP1.2: no preemitters, emitting
contents of doc, usestream=false encoding=UTF-8
[2004-07-19T16:27:07:859Z] DEBUG (SOAP1) W.SOAP1.2: writeEntity, len: 643 data:
<?xml version="1.0" encoding="UTF-8" ?><SOAP-ENV:Envelope
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><SOAP-ENV:Body><GetEffective
AddressResponse xmlns="urn:iwaysoftware:ibse:jul2003:GetEffectiveAddress:response"
cid="9F71FEA4C932CD8786F7388D7EF293A1"><jdeResponse user="JDE" type="callmethod"
environment="DV7333"><callMethod name="GetEffectiveAddress"><returnCode code="2"/>
<params><param
name="mnAddressNumber">12345</param></params></callMethod></jdeResponse></GetEffec
tiveAddressResponse></SOAP-ENV:Body></SOAP-ENV:Envelope>
[2004-07-19T16:27:07:859Z] DEBUG (SOAP1) W.SOAP1.2: writeString: HTTP/1.0
[2004-07-19T16:27:07:859Z] DEBUG (SOAP1) W.SOAP1.2: writeString: 200
[2004-07-19T16:27:07:859Z] DEBUG (SOAP1) W.SOAP1.2: writeString: OK
[2004-07-19T16:27:07:859Z] DEBUG (SOAP1) W.SOAP1.2: writeString: Content-Type:
[2004-07-19T16:27:07:859Z] DEBUG (SOAP1) W.SOAP1.2: writeString: text/xml
```

```
[2004-07-19T16:27:07:875Z] DEBUG (SOAP1) W.SOAP1.2: writeString: Content-Length:
[2004-07-19T16:27:07:875Z] DEBUG (SOAP1) W.SOAP1.2: writeString: 643
[2004-07-19T16:27:07:875Z] INFO  (SOAP1) W.SOAP1.2: W0000X13: Ended message
processing, rc=0
[2004-07-19T16:27:07:875Z] DEEP (SOAP1) W.SOAP1.2: storing used socket
[2004-07-19T16:27:07:875Z] DEBUG (SOAP1) W.SOAP1.2: entering waitforDocument
[2004-07-19T16:27:12:781Z] DEEP (SOAP1) W.SOAP1.2: cleanup: closing sockets(0)
```

Invalid Call Method

If an invalid call is made to Oracle WebLogic Server Adapter, then the following SOAP response is generated.

```
[2004-07-19T16:24:34:859Z] DEBUG (SOAP1) W.SOAP1.2: POST received
[2004-07-19T16:24:34:859Z] DEBUG (SOAP1) W.SOAP1.2: in XDSOAPHTTPWorker agentName
is [XDSOAPRouter]
[2004-07-19T16:24:34:859Z] DEBUG (SOAP1) W.SOAP1.2: before parse:
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<SOAP-ENV:Body>
<m:GetEffectiveAddress xmlns:m="urn:iwaysoftwar...[581]
[2004-07-19T16:24:34:859Z] ERROR (SOAP1) W.SOAP1.2: Attempting string, no encoding
recognized in document
[2004-07-19T16:24:34:859Z] DEEP (SOAP1) W.SOAP1.2: parse complete in 0 msecs
[2004-07-19T16:24:34:875Z] DEEP (SOAP1) W.SOAP1.2: ST_NODICT
[2004-07-19T16:24:34:875Z] DEEP (SOAP1) W.SOAP1.2: ST_FINISH
[2004-07-19T16:24:34:875Z] DEBUG (SOAP1) extractControl - edaDoc: false
[2004-07-19T16:24:34:875Z] DEBUG (SOAP1) now: 2004-07-19T16:24:34Z expires:
2004-07-20T16:24:34Z
[2004-07-19T16:24:34:875Z] DEBUG (SOAP1) W.SOAP1.2: inside isAsync() the soap
Action is ["B0100033.GetEffectiveAddressRequest#test##"]
[2004-07-19T16:24:34:875Z] DEBUG (SOAP1) W.SOAP1.2: inside isAsync() the soap
Action is [B0100033.GetEffectiveAddressRequest#test##]
[2004-07-19T16:24:34:875Z] DEBUG (SOAP1) W.SOAP1.2: checking for cached agent
[2004-07-19T16:24:34:875Z] DEBUG (SOAP1) W.SOAP1.2: pushagent: adding agent
com.ibi.iwse.XDSOAPRouter
[2004-07-19T16:24:34:875Z] DEBUG (SOAP1) W.SOAP1.2: inside worker the soap Action
is [B0100033.GetEffectiveAddressRequest#test##]
[2004-07-19T16:24:34:890Z] DEBUG (SOAP1) W.SOAP1.2: precedence: 1
[2004-07-19T16:24:34:890Z] DEBUG (SOAP1) W.SOAP1.2: precedence: 1
[2004-07-19T16:24:34:890Z] DEBUG (SOAP1) W.SOAP1.2: numagents: 1
[2004-07-19T16:24:34:890Z] DEBUG (SOAP1) W.SOAP1.2: running agent 1 name
com.ibi.iwse.XDSOAPRouter document 1
[2004-07-19T16:24:35:031Z] INFO  (manager) MGR00X01: Adding active worker:
W.SOAP1.2
[2004-07-19T16:24:35:031Z] DEBUG (SOAP1) W.SOAP1.2: <?xml version="1.0"
encoding="UTF-8" ?>
<SOAP-ENV:Envelope xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <SOAP-ENV:Body>
      <m:GetEffectiveAddress
xmlns:m="urn:iwaysoftware:ibse:jul2003:GetEffectiveAddress">
         <m:jdeRequest type="callmethod">
            <m:callMethod name="GetAddress">
               <m:params>
                  <m:param name="mnAddressNumber">34518</m:param>
               </m:params>
               <m:onError/>
            </m:callMethod>
         </m:jdeRequest>
      </m:GetEffectiveAddress>
   </SOAP-ENV:Body>
   <SOAPAction agentName="XDSOAPRouter"
cid="4C0AD8398CB7A5B4DED18057D963AA44">B0100033.GetEffectiveAddressRequest#test##<
/SOAPAction>
</SOAP-ENV:Envelope>
[2004-07-19T16:24:35:031Z] DEBUG (SOAP1) W.SOAP1.2: business method:
m:GetEffectiveAddress
[2004-07-19T16:24:35:031Z] DEBUG (SOAP1) W.SOAP1.2: input:
[2004-07-19T16:24:35:031Z] DEBUG (SOAP1) W.SOAP1.2: <?xml version="1.0"
encoding="UTF-8" ?><jdeRequest xmlns:xsd="http://www.w3.org/2001/XMLSchema"
type="callmethod" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><callMethod
name="GetAddress"><params><param name="mnAddressNumber">34518</param>
      </params><onError/></callMethod></jdeRequest>
[2004-07-19T16:24:36:781Z] DEBUG (SOAP1) W.SOAP1.2: Agent returned success
[2004-07-19T16:24:36:781Z] INFO  (manager) MGR00X02: Removing active worker:
W.SOAP1.2
[2004-07-19T16:24:36:781Z] DEBUG (SOAP1) W.SOAP1.2: doing docTran, docVal,
listTran for agent(1)
[2004-07-19T16:24:36:781Z] DEBUG (SOAP1) W.SOAP1.2: sendToAll reply to XDReply:
[protocol=http */null]
[2004-07-19T16:24:36:781Z] DEBUG (SOAP1) W.SOAP1.2: preemitters from doc: null
[2004-07-19T16:24:36:781Z] DEBUG (SOAP1) W.SOAP1.2: no preemitters, emitting
contents of doc, usestream=false encoding=UTF-8
[2004-07-19T16:24:36:781Z] DEBUG (SOAP1) W.SOAP1.2: writeEntity, len: 595 data:
<?xml version="1.0" encoding="UTF-8" ?><SOAP-ENV:Envelope
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><SOAP-ENV:Body><GetEffective
AddressResponse xmlns="urn:iwaysoftware:ibse:jul2003:GetEffectiveAddress:response"
cid="4C0AD8398CB7A5B4DED18057D963AA44"><jdeResponse user="JDE" type="callmethod"
environment="DV7333"><callMethod name="GetAddress"><returnCode code="99"/><params>
</params></callMethod></jdeResponse></GetEffectiveAddressResponse></SOAP-ENV:Body>
</SOAP-ENV:Envelope>
[2004-07-19T16:24:36:796Z] DEBUG (SOAP1) W.SOAP1.2: writeString: HTTP/1.0
[2004-07-19T16:24:36:796Z] DEBUG (SOAP1) W.SOAP1.2: writeString: 200
[2004-07-19T16:24:36:796Z] DEBUG (SOAP1) W.SOAP1.2: writeString: OK
[2004-07-19T16:24:36:796Z] DEBUG (SOAP1) W.SOAP1.2: writeString: Content-Type:
[2004-07-19T16:24:36:796Z] DEBUG (SOAP1) W.SOAP1.2: writeString: text/xml
[2004-07-19T16:24:36:796Z] DEBUG (SOAP1) W.SOAP1.2: writeString: Content-Length:
[2004-07-19T16:24:36:796Z] DEBUG (SOAP1) W.SOAP1.2: writeString: 595
[2004-07-19T16:24:36:796Z] INFO  (SOAP1) W.SOAP1.2: W0000X13: Ended message
processing, rc=0
[2004-07-19T16:24:36:796Z] DEEP (SOAP1) W.SOAP1.2: storing used socket
[2004-07-19T16:24:36:812Z] DEBUG (SOAP1) W.SOAP1.2: entering waitforDocument
[2004-07-19T16:24:42:671Z] DEEP (SOAP1) W.SOAP1.2: cleanup: closing sockets(0)
```

7

Advanced User Tools

This chapter includes the following topics:

- Web Services Policy-Based Security
- Migrating Repositories

Web Services Policy-Based Security

OracleWLS Adapter Application Explorer (Application Explorer) provides a security model called Web services policy-based security. This section describes 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. Easy application integration is enabled, but 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 will not be 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 OracleWLS Adapter Business Services Engine (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 execute the Business Service.

When a policy is not applied, the default value for a Business Service is to "grant all". For example, anybody can execute 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.

Configuring Web Services Policy-Based Security

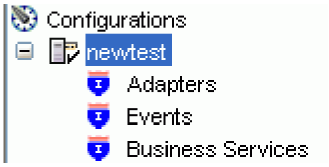
The following procedures describe how to configure Web services policy-based security.

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. Start Application Explorer.
2. Right-click the configuration to which you want to connect, for example, **newtest**. See Chapter 2, "Configuring OracleWLS Application Adapter for J.D. Edwards One World" for information on creating a new configuration.
3. Select **Connect**.

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



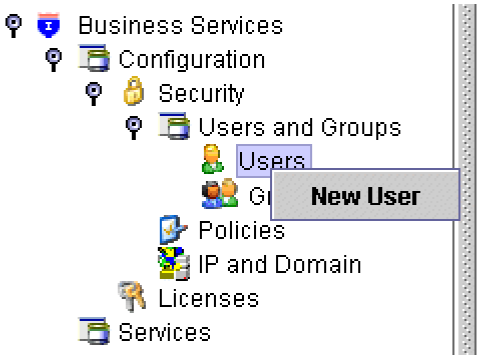
   Perform the following steps:

   a. Expand the **Business Services** node.

   b. Expand the **Configuration** node.

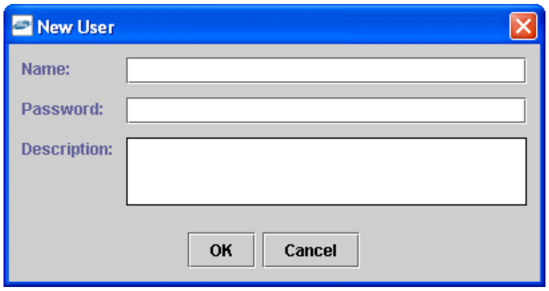
   c. Expand the **Security** node.

   d. Expand the **Users and Groups** node.



4. Right-click **Users** and click **New User**.

The New User dialog box is displayed.



Perform the following steps:

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

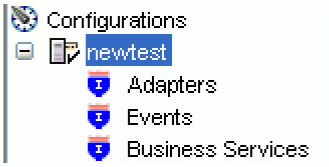
To create a group to use with a policy:

1. Start Application Explorer.

2. Right-click the configuration to which you want to connect, for example, **newtest**. See Chapter 2, "Configuring OracleWLS Application Adapter for J.D. Edwards One World" for information on creating a new configuration.

3. Select **Connect**.

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



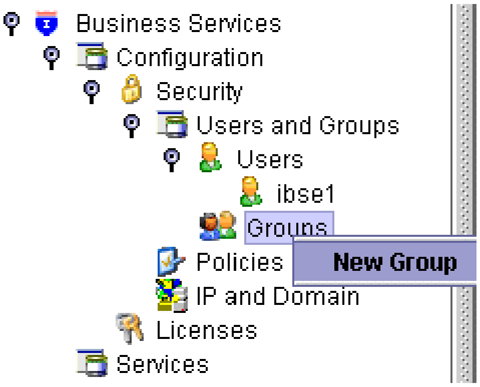
Perform the following steps:

a. Expand the **Business Services** node.

b. Expand the **Configuration** node.

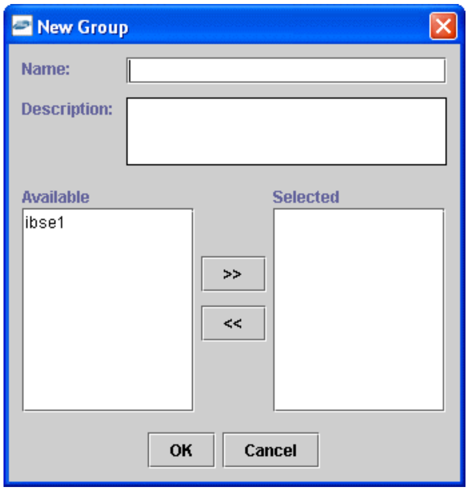
c. Expand the **Security** node.

d. Expand the **Users and Groups** node.



4. Right-click **Groups** and select **New Group**.

   The New Group dialog box is displayed.



   Perform the following steps:

   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 Groups node.

Creating an Execution Policy

An execution policy determines who can execute the Business Services to which the policy is applied.

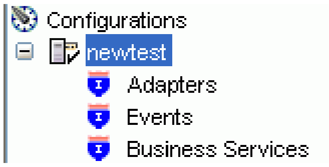
To create an execution policy:

1. Start Application Explorer.

2. Right-click the configuration to which you want to connect, for example, SampleConfig. See Chapter 2, "Configuring OracleWLS Application Adapter for J.D. Edwards One World" for information on creating a new configuration.

3. Select **Connect**.

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



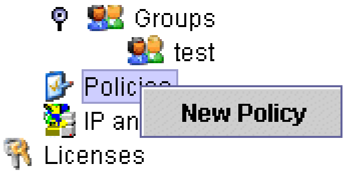
   Perform the following steps:

   a. Expand the **Business Services**node.

   b. Expand the **Configuration** node.

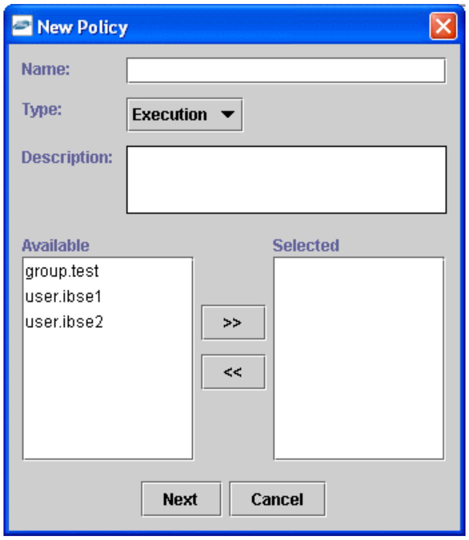
   c. Expand the **Security** node.

   d. Expand the **Policies** node.



4. Right-click **Policies** and select **New Policy**.

   The New Policy dialog box is displayed.



Type: Execution

group.test
user.ibse1
user.ibse2

   Perform the following steps:

   a. In the **Name** field, enter a name for the policy.

   b. From the **Type** list, select **Execution**.

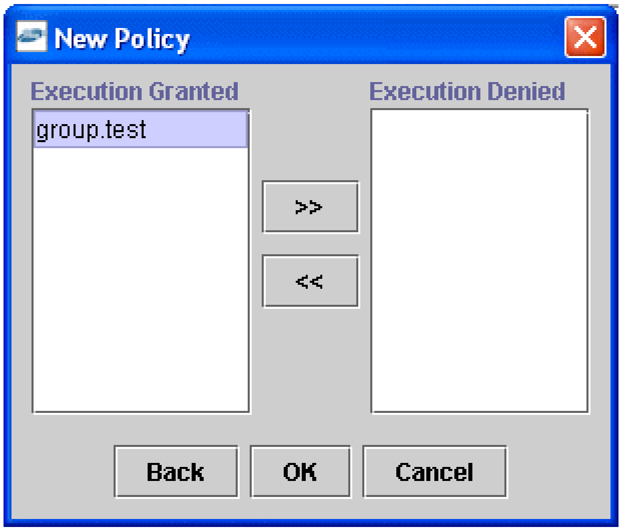
   c. In the **Description** field, enter a description for the policy (optional).

   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.

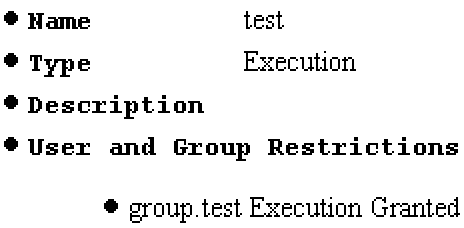
5. When you have selected at least one user selected, click **OK**.
6. Click **Next**.

The New Policy permissions dialog box is displayed.



7. To grant permission to a user or group to execute a Business Service, select the user or group and move them into the **Execution Granted** list by selecting the double left facing arrow.
8. To deny permission to a user or group to execute a Business Service, select the user or group and move them into the **Execution Denied** list by selecting the double right facing arrow.
9. 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. Start Application Explorer.

2. Right-click the configuration to which you want to connect, for example, SampleConfig. See Chapter 2, "Configuring OracleWLS Application Adapter for J.D. Edwards One World" for information on creating a new configuration.

3. Select **Connect**.

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

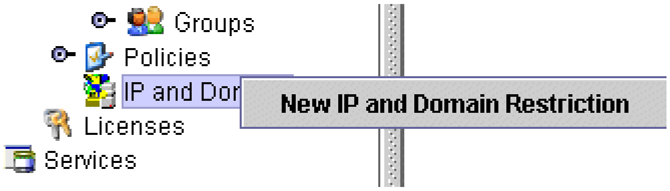
   Perform the following steps:

   a. Expand the **Business Services** node.

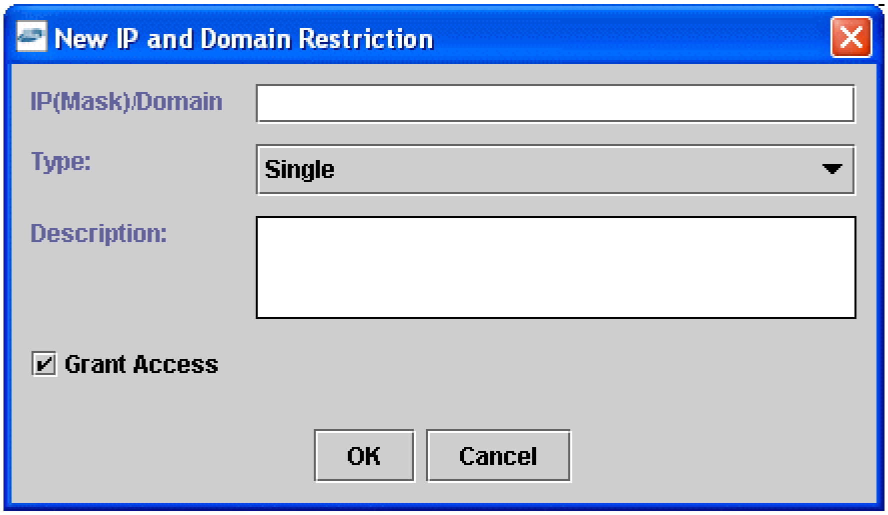
   b. Expand the **Configuration** node.

   c. Expand the **Security** node.

4. Right-click **IP and Domain** and select **New IP and Domain Restriction**.



   The New IP and Domain Restriction dialog box is displayed.



Type: Single

☑ Grant Access

   Perform the following steps:

   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.

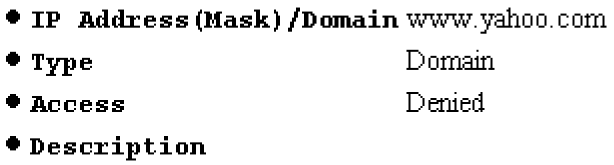
   c. In the **Description** field, enter a description (optional).

   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**

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

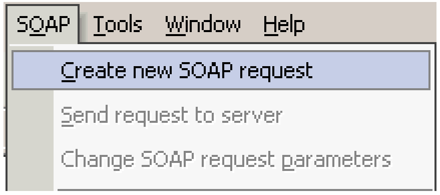
To migrate a BSE repository:

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

   ```
   http://localhost:7777/ibse/IBSEServlet/admin/iwcontrol.ibs
   ```

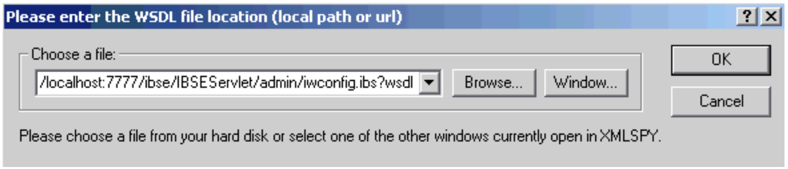
2. Open a third party XML editor, for example, XMLSPY.
3. From the menu bar, click **SOAP**.

   A list of options appears.



4. Select **Create new SOAP request**.

   The WSDL file location dialog box is displayed.



/localhost:7777/ibse/IBSEServlet/admin/iwconfig.ibs?wsdl

Please choose a file from your hard disk or select one of the other windows currently open in XMLSPY.

Perform the following steps:

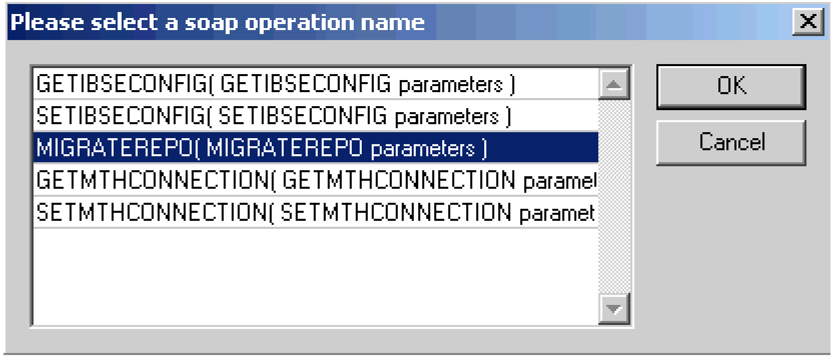
a. In the **Choose a file** field, paste the BSE control service URL.

b. Append **?wsdl** to the URL, for example:

```
http://localhost:7777/ibse/IBSEServlet/admin/iwcontrol.ibs?wsdl
```

5. Click **OK**.

The soap operation name dialog box is displayed and lists the available control methods.



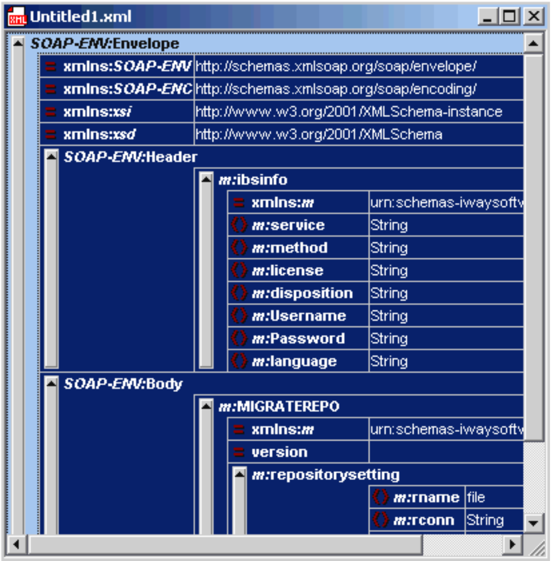
Please select a soap operation name

GETIBSECONFIG(GETIBSECONFIG parameters)
SETIBSECONFIG(SETIBSECONFIG parameters)
MIGRATEREPO(MIGRATEREPO parameters)
GETMTHCONNECTION(GETMTHCONNECTION paramet
SETMTHCONNECTION(SETMTHCONNECTION paramet

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.

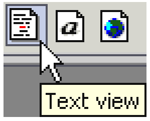
The following window shows the structure of the SOAP envelope.



Untitled1.xml

```
SOAP-ENV:Envelope
  xmlns:SOAP-ENV   http://schemas.xmlsoap.org/soap/envelope/
  xmlns:SOAP-ENC   http://schemas.xmlsoap.org/soap/encoding/
  xmlns:xsi        http://www.w3.org/2001/XMLSchema-instance
  xmlns:xsd        http://www.w3.org/2001/XMLSchema
  SOAP-ENV:Header
                   m:ibsinfo
                     xmlns:m          urn:schemas-iwaysoftv
                     m:service        String
                     m:method         String
                     m:license        String
                     m:disposition    String
                     m:Username       String
                     m:Password       String
                     m:language       String
  SOAP-ENV:Body
                   m:MIGRATEREPO
                     xmlns:m          urn:schemas-iwaysoftv
                     version
                     m:repositorysetting
                                      m:rname    file
                                      m:rconn    String
```

7. Locate the **Text view** icon in the toolbar.



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

   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. Perform one of the following migration options.

   - 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:

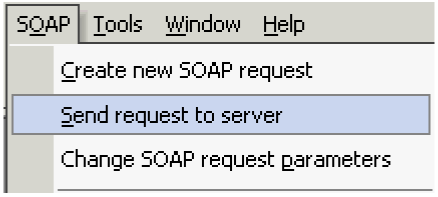
```
<m:servicename>JDEService1</m:servicename>
```

   - 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>JDEService1</m:servicename>
<m:servicename>JDEService2</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:

   ```
   wls_home\erp-adapters\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.

A

Configuring J.D. Edwards OneWorld for Outbound and Inbound Processing

J.D. Edwards OneWorld enables you to specify inbound functionality for Master Business Functions (MBF).

The following topics describe how to enable outbound and inbound transaction processing in J.D. Edwards OneWorld and how to modify the `jde.ini` file for XML support.

Modifying the JDE.INI File for Outbound and Inbound Processing

This section describes the settings that are required in the JDE.INI file for the XML call object kernel (outbound and inbound processing).

Open the JDE.INI file and modify the **[JDENET_KERNEL_DEF6]** and **[JDENET_KERNEL_DEF15]** sections as follows:

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

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

The parameters containing an underscore (_) and @28 are for Windows NT operating systems only. For other operating systems, replace the parameters with the values in the following table:

Operating System	Call Object dispatch DLLName	XML Trans dispatch DLLName
AS400	XMLCALLOBJ	XMLTRANS

Operating System	Call Object dispatch DLLName	XML Trans dispatch DLLName
HP9000B	libxmlcallojb.sl	libxmltransactions.lo
Sun or RS6000	libxmlcallojb.so	Libxmltransactions.so

Note: The J.D. Edwards OneWorld installation for version B7333(XE) does not include **[JDENET_KERNEL_DEF15]**. As a result, if you are using version B7333(XE), you must manually add it to the jde.ini file. For all other J.D. Edwards OneWorld versions, **[JDENET_KERNEL_DEF15]** is included with the installation.

Using the GenJava Development Tool (Outbound Processing)

This section describes how to use the GenJava development tool, which is used to create Java wrappers for accessing the J.D. Edwards OneWorld business functions. The OracleWLS Application Adapter for J.D. Edwards OneWorld uses these wrappers to call the J.D. Edwards OneWorld business functions.

J.D. Edwards OneWorld provides a Java Generation tool called GenJava that you can use to expose OneWorld business functions externally as Java class files. A J.D. Edwards OneWorld system administrator usually runs the GenJava tool.

During GenJava operation, you must specify a library of business functions, for example CALLBSFN. GenJava creates the associated Java class files for the business functions and related data structures. GenJava also compiles the business functions, generates Java documents, and packages them into two .JAR files. One .JAR file contains Java classes and the second .JAR file contains Java documents.

For example, if the business function library you specified in GenJava is CALLBSFN, you will find the following files in the <install>\system\classes directory or any user-specified directory redirected by GenJava:

- JDEJAVA_CALLBSFN.xml
- JDEJAVA_CALLBSFNInterop.jar
- JDEJAVA_CALLBSFNInteropDoc.jar

Once they are generated, these library files must be added to the CLASSPATH.

GenJava also provides access to J.D. Edwards OneWorld business functions by generating pure Java interfaces for these business functions. GenJava can be generated from a thick client or a deployment server.

Running GenJava

GenJava is located in the <install>\system\bin32 directory. You run GenJava from the command line. There are two GenJava command options that can be used to generate the wrappers.

GenJava Command Option 1

The following command generates Java wrappers for Category 1 (Master Business Functions), Category 2 (Major Business Functions), Category 3 (Minor Business Functions), and Category - (Uncategorized Business Functions) in the CALLBSFN library:

```
GenJava /Cat 1 /Cat 2 /Cat 3 /Cat - CALLBSFN
```

GenJava Command Option 2

The GenJava command can also be run with a JDEScript file and prompts a J.D. Edwards OneWorld log on window, where you must enter a valid user ID, password, and environment.

1. Using an editor, create a new file called `AddressBook.cmd` and enter the following commands:

   ```
   define library CALLBSFN
   login
   library CALLBSFN
   interface AddressBook
   import B0100031
   import B0100019
   import B0100032
   import B0100002
   import B0100033
   build
   logout
   ```

2. Run the following GenJava command:

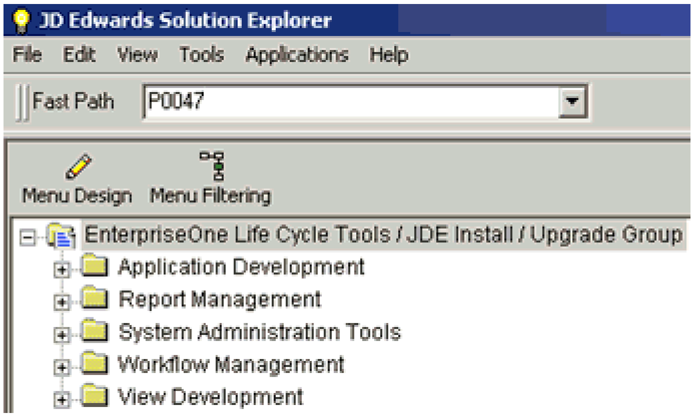
   ```
   GenJava /cmd .\AddressBook.cmd
   ```

3. GenJava generates the wrappers (CALLBSFNInterop.jar, CALLBSFNInteropDoc.jar, and CALLBSFN.xml) in Java for all business functions that are mentioned in the script file.

Triggering J.D. Edwards OneWorld Events

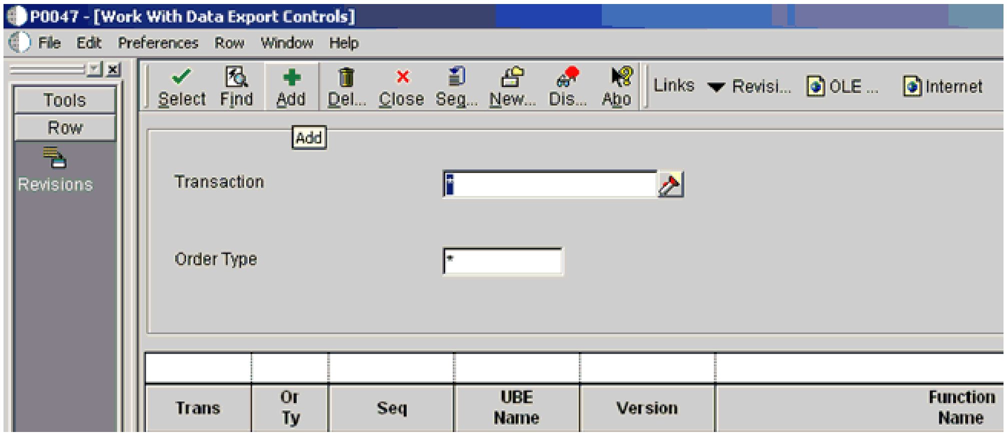
The flow of inbound data to third parties is controlled through the Data Export Controls application. For each transaction type and order type, one or more records can be defined with different function names and libraries.

1. Type **P0047** in the Fast Path field and press **Enter**.


JD Edwards Solution Explorer
File Edit View Tools Applications Help
Menu Design
Menu Filtering
EnterpriseOne Life Cycle Tools / JDE Install / Upgrade Group
Application Development
Report Management
System Administration Tools
Workflow Management
View Development

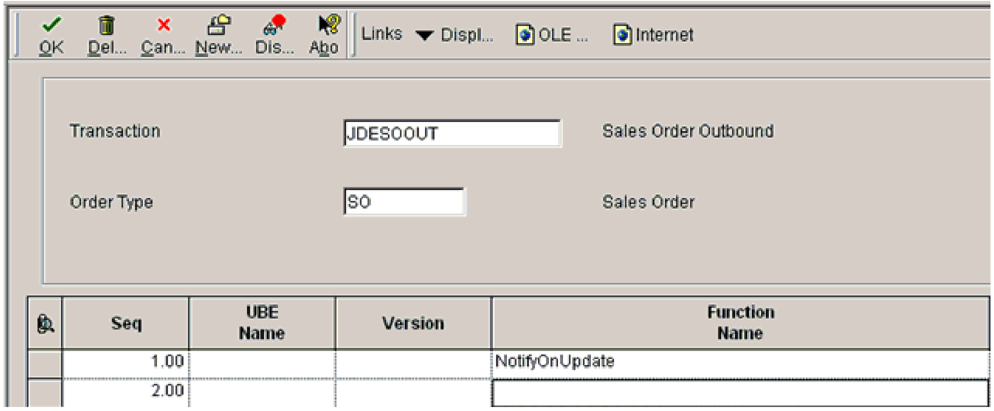
Fast Path: P0047

The Work With Data Export Controls window opens.



2. Click **Add**.

   The Data Export Control Revisions window opens. Notice that the sequence (Seq) number automatically increments for each new line.



Transaction	JDESOOUT	Sales Order Outbound
Order Type	SO	Sales Order

Seq	UBE Name	Version	Function Name
1.00			NotifyOnUpdate
2.00			

3. Perform the following steps:

   a. Type **JDESOOUT** in the Transaction field.

   b. Type **SO** in the Order Type field.

   c. Type **NotifyOnUpdate** in the first row of the Function Name column.

   d. Type the absolute path to the location of the iwoevent.dll file in the first row of the Function Library column, for example:

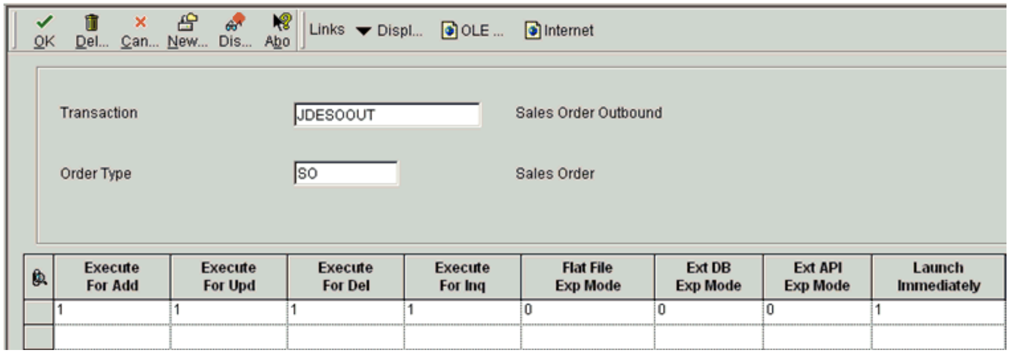
   ```
   D:\JDEdwards\E812\DDP\Outbound\iwoevent.dll
   ```

   e. Type **1** in the first row of the Execute for Add column if you want the notifications for add/insert.

   f. Make the same decision for update, delete, and inquiry and type **1** in the appropriate column.

   g. Type **1** in the Launch Immediately column to launch the Outbound Subsystem batch process (R00460).

4. Click **OK**.



Transaction JDESOOUT Sales Order Outbound

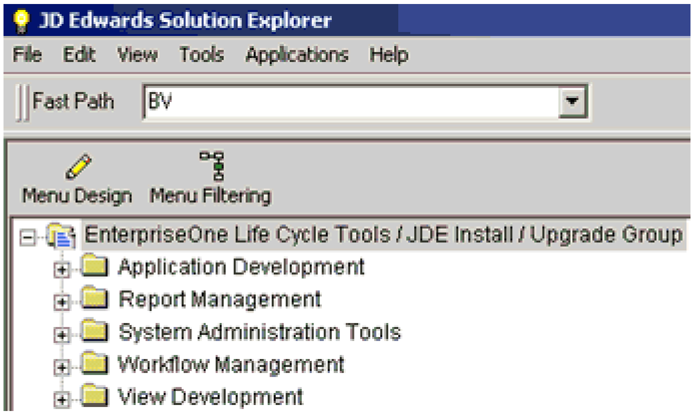
Order Type SO Sales Order

Execute For Add	Execute For Upd	Execute For Del	Execute For Inq	Flat File Exp Mode	Ext DB Exp Mode	Ext API Exp Mode	Launch Immediately
1	1	1	1	0	0	0	1

Starting the Outbound Scheduler Subsystem Process (R00460)

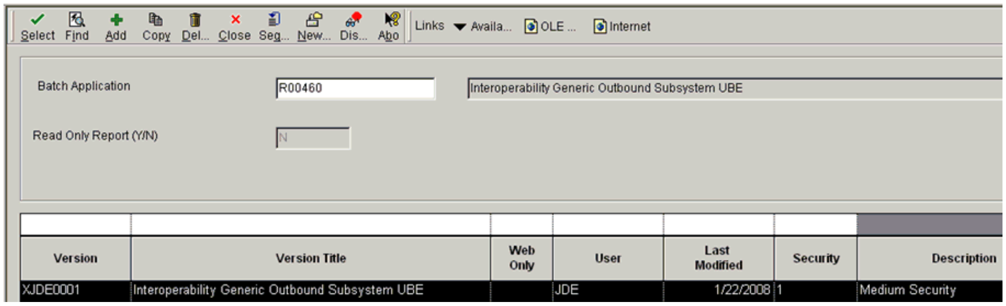
Once you have finished defining one or more records for each transaction type and order type, you must manually start the outbound scheduler subsystem process.

1. Type **BV** in the Fast Path field and press **Enter**.



Fast Path BV

The Work With Batch Versions - Available Versions window opens.

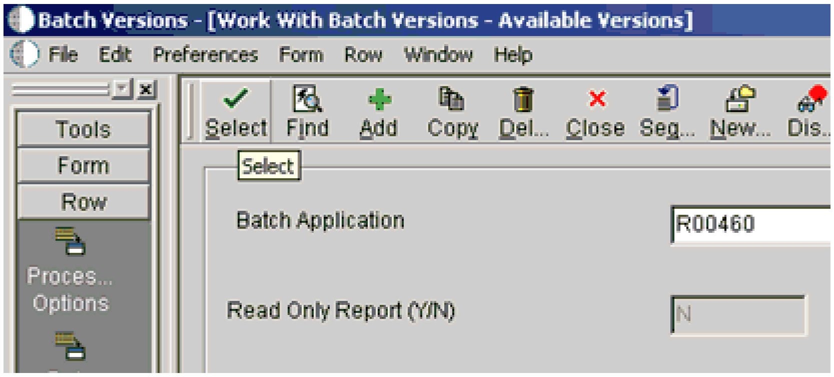


Batch Application R00460 Interoperability Generic Outbound Subsystem UBE

Read Only Report (Y/N) N

Version	Version Title	Web Only	User	Last Modified	Security	Description
XJDE0001	Interoperability Generic Outbound Subsystem UBE		JDE	1/22/2008	1	Medium Security

2. Type **R00460** in the Batch Application field and click **Find**.

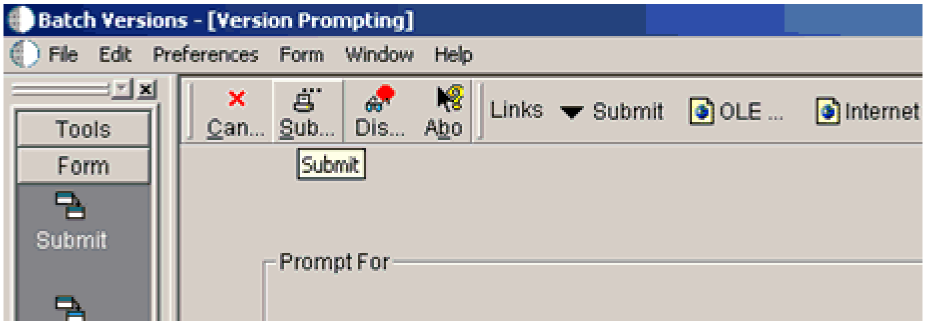

Batch Versions - [Work With Batch Versions - Available Versions]
File Edit Preferences Form Row Window Help
Tools
Form
Row
Proces... Options
Select
Find
Add
Copy
Del...
Close
Seq...
New...
Dis...
Select

Batch Application R00460

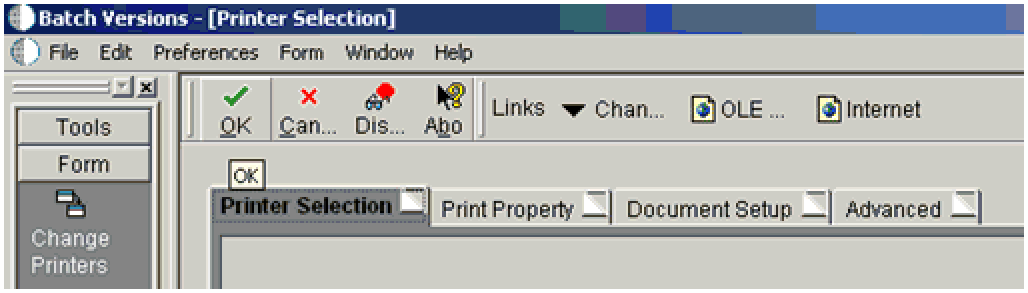
Read Only Report (Y/N) N

3. Select **Interoperability Generic Outbound Subsystem UBE (XJDE0001)** and click **Select**.

   The Version Prompting window opens.



4. Click **Submit**.

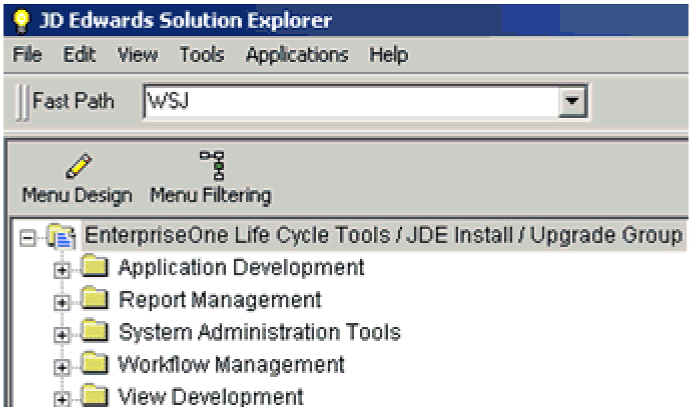


5. Navigate to the last screen and click **OK**.

Verifying the Subsystem Process

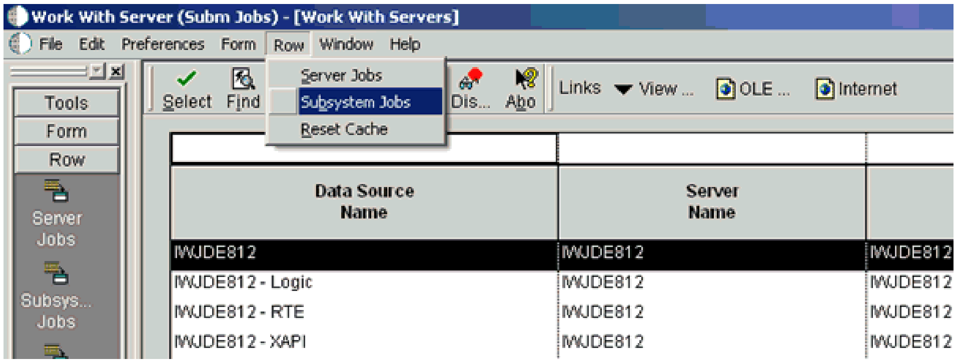
This section describes how to verify the outbound scheduler subsystem processs.

1. Type **WSJ** in the Fast Path field and press **Enter**.


JD Edwards Solution Explorer
File Edit View Tools Applications Help
Menu Design
Menu Filtering
EnterpriseOne Life Cycle Tools / JDE Install / Upgrade Group
Application Development
Report Management
System Administration Tools
Workflow Management
View Development

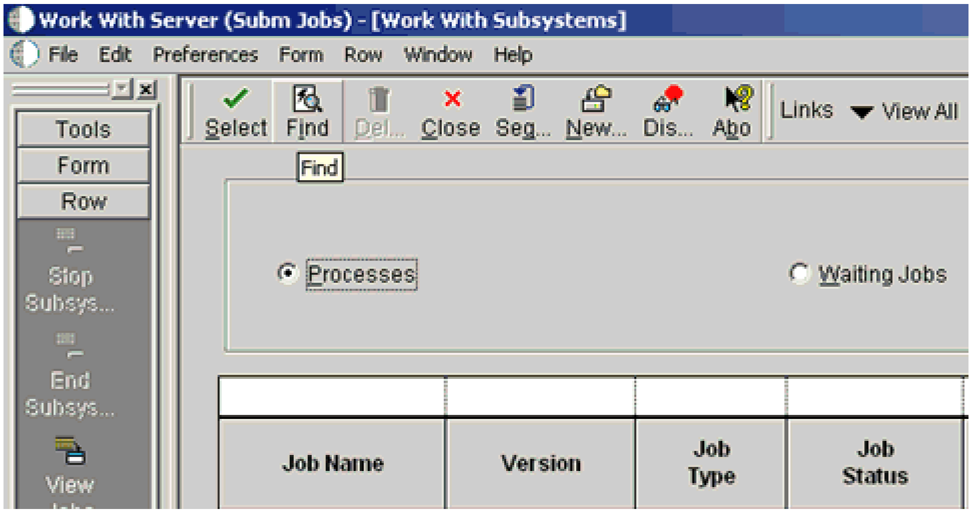
Fast Path: WSJ

The Work With Server (Subm Jobs) window opens.


Work With Server (Subm Jobs) - [Work With Servers]
File Edit Preferences Form Row Window Help
Select Find
Server Jobs
Subsystem Jobs
Reset Cache
Dis... Abo
Links View ... OLE ... Internet
Tools
Form
Row
Server Jobs
Subsys... Jobs

Data Source Name	Server Name	
IWJDE812	IWJDE812	IWJDE812
IWJDE812 - Logic	IWJDE812	IWJDE812
IWJDE812 - RTE	IWJDE812	IWJDE812
IWJDE812 - XAPI	IWJDE812	IWJDE812

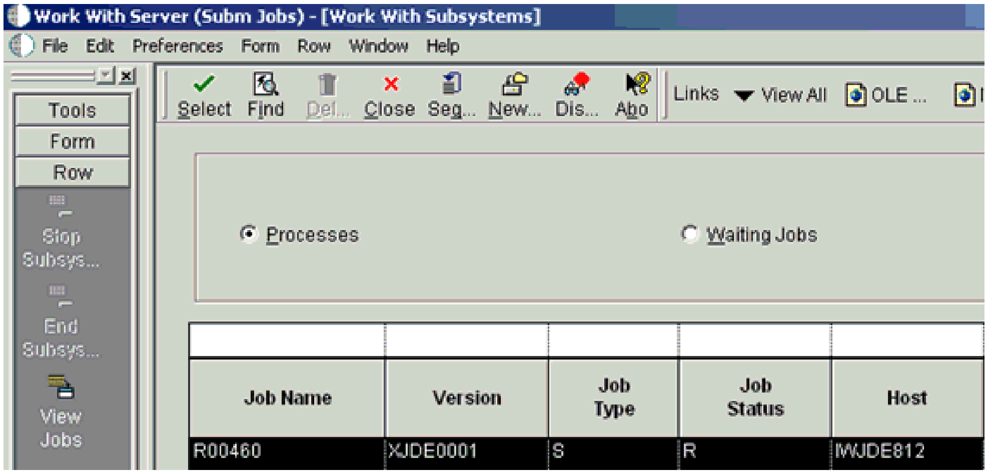
2. Select a corresponding server from the table.
3. Click **Row** from the menu bar and select **Subsystem Jobs**.


Work With Server (Subm Jobs) - [Work With Subsystems]
File Edit Preferences Form Row Window Help
Select Find Del... Close Seq... New... Dis... Abo
Links View All
Tools
Form
Row
Stop Subsys...
End Subsys...
View Jobs
Find

● Processes ○ Waiting Jobs

Job Name	Version	Job Type	Job Status

4. Click **Find**.


Work With Server (Subm Jobs) - [Work With Subsystems]
File Edit Preferences Form Row Window Help
Tools
Form
Row
Stop Subsys...
End Subsys...
View Jobs
Select Find Del... Close Seq... New... Dis... Abo
Links View All OLE ...

● Processes ○ Waiting Jobs

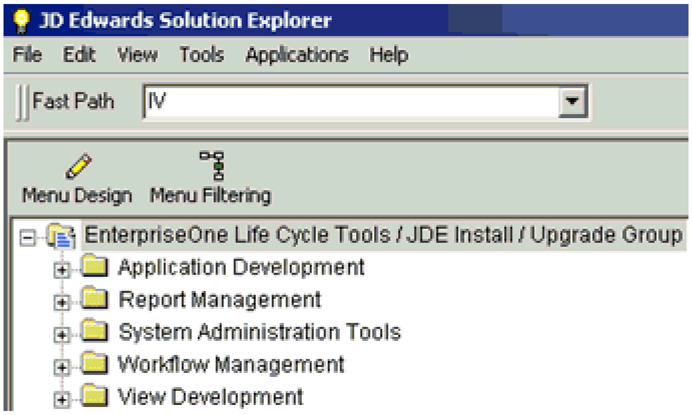
Job Name	Version	Job Type	Job Status	Host
R00460	XJDE0001	S	R	IWJDE812

5. Verify that **R** is listed in the Job Status column.

Configuring P4210 (Sales Order) to Trigger an Event

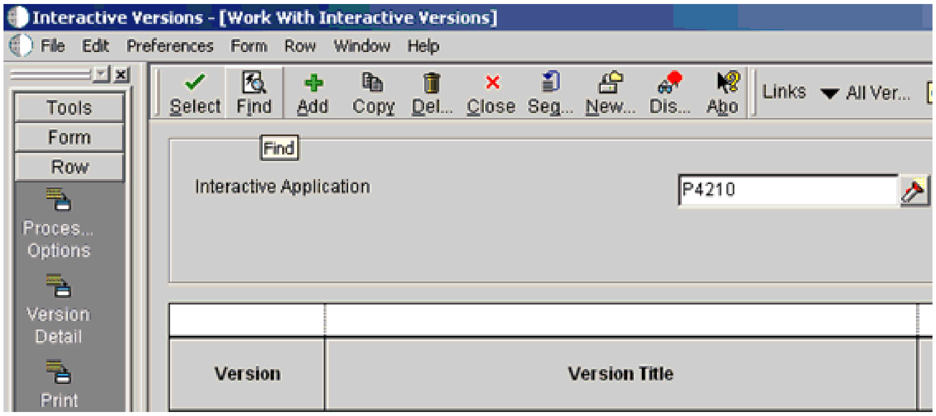
This section describes how to configure a P4210 (Sales Order) to trigger an event.

1. Type **IV** in the Fast Path field and press **Enter**.


JD Edwards Solution Explorer
File Edit View Tools Applications Help
Menu Design
Menu Filtering
EnterpriseOne Life Cycle Tools / JDE Install / Upgrade Group
Application Development
Report Management
System Administration Tools
Workflow Management
View Development

Fast Path: IV

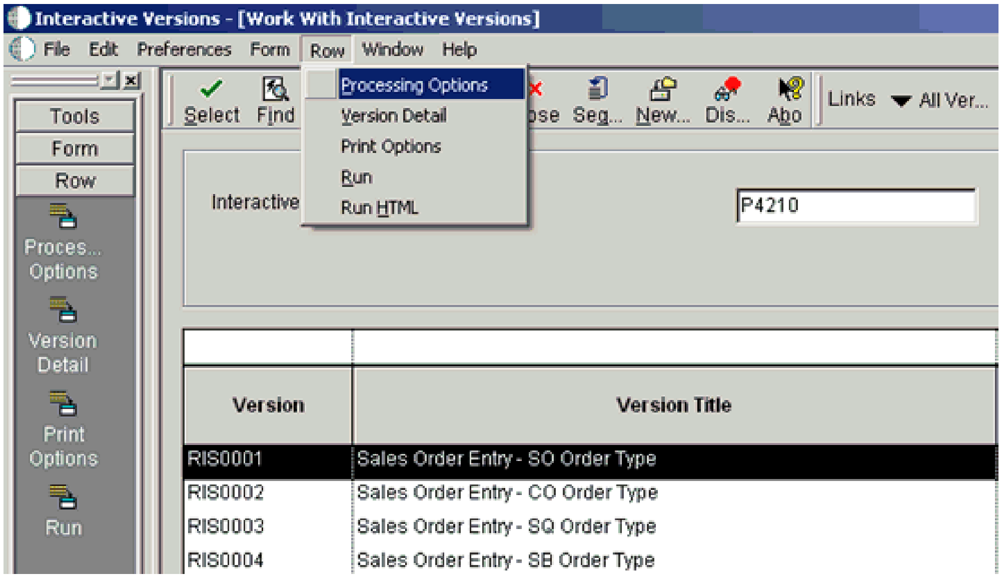
The Interactive Versions window opens.


Interactive Versions - [Work With Interactive Versions]
File Edit Preferences Form Row Window Help
Tools
Form
Row
Proces... Options
Version Detail
Print
Select Find Add Copy Del... Close Seq... New... Dis... Abo
Links All Ver...
Find

Interactive Application: P4210

Version	Version Title

2. Type **P4210** in the Interactive Application field and click **Find**.



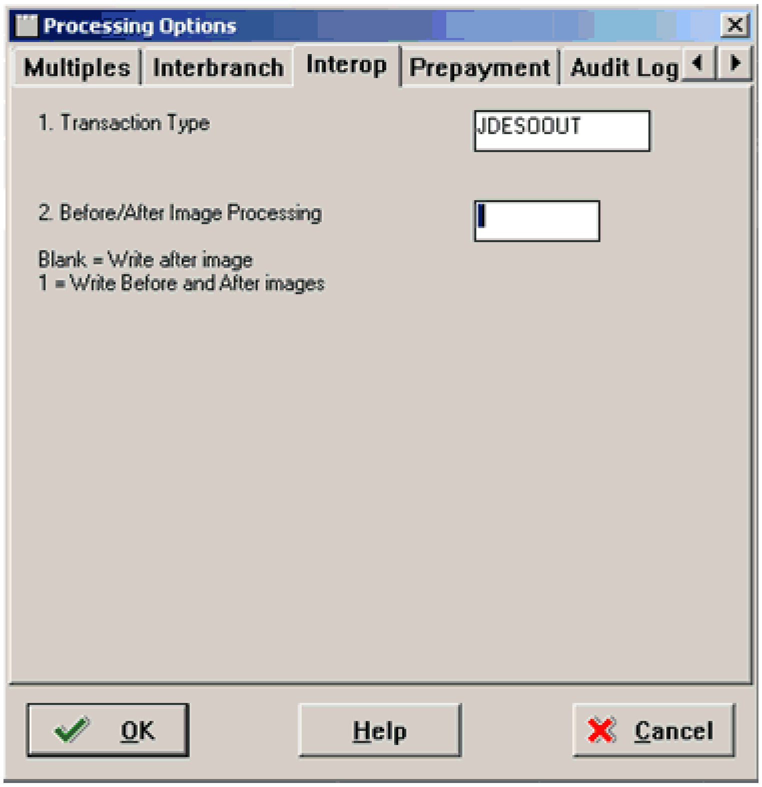
Interactive P4210

Version	Version Title
RIS0001	Sales Order Entry - SO Order Type
RIS0002	Sales Order Entry - CO Order Type
RIS0003	Sales Order Entry - SQ Order Type
RIS0004	Sales Order Entry - SB Order Type

3. Select a document version from the table, for example, **RIS0001 - Sales Order Entry - SO Order Type**.

4. Click **Row** from the menu bar and select **Processing Options**.

   The Processing Options dialog box opens.



1. Transaction Type JDESOOUT

2. Before/After Image Processing

Blank = Write after image
1 = Write Before and After images

5. Click the **Interop** tab.

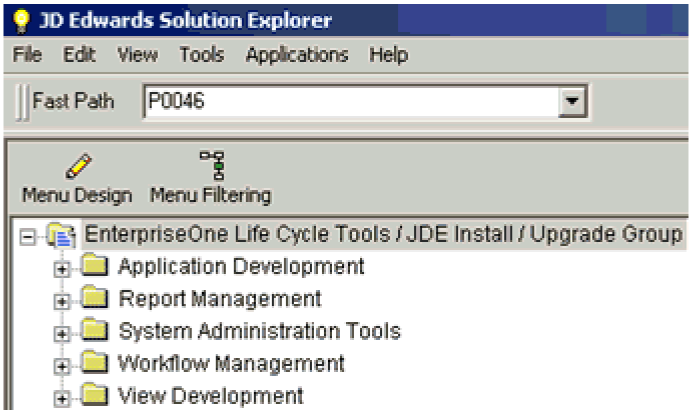
6. Type **JDESOOUT** in the Transaction Type field.

7. Add **Sales Order**.

Verifying the Configuration Steps

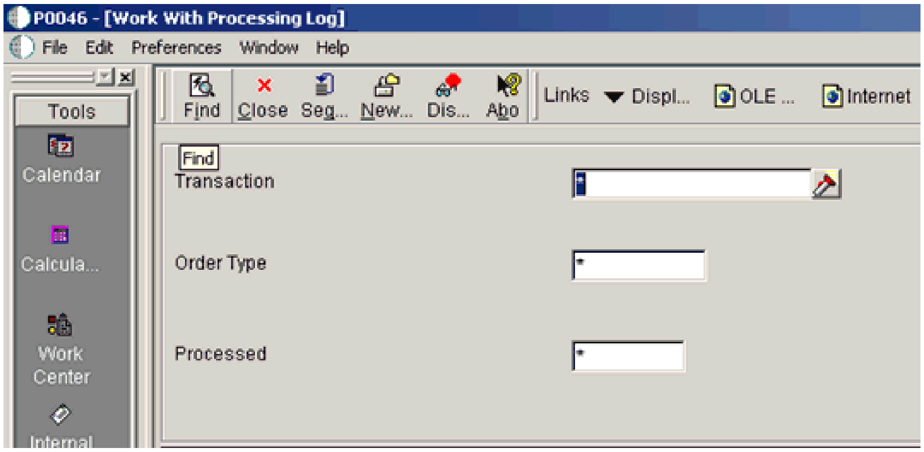
This section describes how to verify the configuration steps by updating F0046.

1. Type **P0046** in the Fast Path field and press **Enter**.


JD Edwards Solution Explorer
File Edit View Tools Applications Help
Menu Design
Menu Filtering
EnterpriseOne Life Cycle Tools / JDE Install / Upgrade Group
Application Development
Report Management
System Administration Tools
Workflow Management
View Development

Fast Path: P0046

The P0046 - Work With Processing Log window opens.



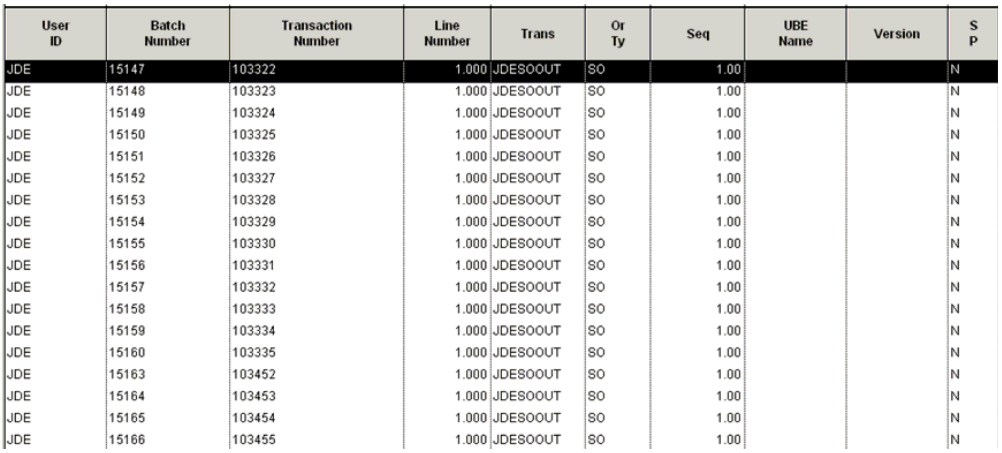
Transaction:

Order Type: *

Processed: *

2. Click **Find**.

   The following data is displayed.



User ID	Batch Number	Transaction Number	Line Number	Trans	Or Ty	Seq	UBE Name	Version	S P
JDE	15147	103322	1.000	JDESOOUT	SO	1.00			N
JDE	15148	103323	1.000	JDESOOUT	SO	1.00			N
JDE	15149	103324	1.000	JDESOOUT	SO	1.00			N
JDE	15150	103325	1.000	JDESOOUT	SO	1.00			N
JDE	15151	103326	1.000	JDESOOUT	SO	1.00			N
JDE	15152	103327	1.000	JDESOOUT	SO	1.00			N
JDE	15153	103328	1.000	JDESOOUT	SO	1.00			N
JDE	15154	103329	1.000	JDESOOUT	SO	1.00			N
JDE	15155	103330	1.000	JDESOOUT	SO	1.00			N
JDE	15156	103331	1.000	JDESOOUT	SO	1.00			N
JDE	15157	103332	1.000	JDESOOUT	SO	1.00			N
JDE	15158	103333	1.000	JDESOOUT	SO	1.00			N
JDE	15159	103334	1.000	JDESOOUT	SO	1.00			N
JDE	15160	103335	1.000	JDESOOUT	SO	1.00			N
JDE	15163	103452	1.000	JDESOOUT	SO	1.00			N
JDE	15164	103453	1.000	JDESOOUT	SO	1.00			N
JDE	15165	103454	1.000	JDESOOUT	SO	1.00			N
JDE	15166	103455	1.000	JDESOOUT	SO	1.00			N

3. Search for the corresponding transaction.

   The iwoevnt.log file is created in the following directory:

```
\\iwJDE812\JDEdwards\E812\DDP\system\bin32
```

The iwoevent.log file is created in the outbound folder where the iwoevent.dll and iwoevent.cfg files are located. The following is an example of the event log file:

```
Event call begin...
Server time      : Tue May 27 07:23:55 2008
userId           : JDE
batchNumber      : 15205
transactionNumber: 103494
lineNumber       : 1.000000
transactionType  : JDESOOUT
sequenceNumber   : 1.000000
Request xml:
========================
<? xml version="1.0" encoding="UTF-8"?><jde><request><connection><dsn /><user
/><password /><sp><proc>JDESOOUT</proc><data><ediUserId>JDE
</ediUserId><ediBatchNumber>15205
</ediBatchNumber><ediTransactionNumber>103494
</ediTransactionNumber></data></sp></connection></request></jde>
========================
```

B

Sample Files

The Adapter for J.D. Edwards OneWorld supports the jdeRequest and jdeResponse XML structures for executing business functions within OneWorld. Using J.D. Edwards OneWorld XML, you can:

- Aggregate business function calls into a single object.
- Use the J.D. Edwards OneWorld ThinNet API.
- Access both Z files and business functions.

The following topics provide examples of the jdeRequest and jdeResponse XML structures for executing business functions within OneWorld:

- Issuing a Single-Function Request
- Issuing a Multiple-Function Request
- Sample Sales Order Request
- Sample Sales Order Response

Issuing a Single-Function Request

The following example, GetEffectiveAddress, is a single-function call to J.D. Edwards OneWorld, and the result of this request is a standard jdeResponse document. In a single-function request, only one callMethod within the XML object is specified.

Executing a Business Function with a Single-Function Call

The following is a sample GetEffectiveAddress jdeRequest.

```
<jdeRequest type="callmethod" user="JDE" pwd="JDE" environment="DV7333"
session="">
<callMethod name="GetEffectiveAddress" app="BSE" runOnError="no">
<params>
   <param name="mnAddressNumber">1001</param>
   <param name="jdDateBeginningEffective"></param>
   <param name="cEffectiveDateExistence10"></param>
   <param name="szAddressLine1"></param>
   <param name="szAddressLine2"></param>
   <param name="szAddressLine3"></param>
   <param name="szAddressLine4"></param>
   <param name="szZipCodePostal"></param>
   <param name="szCity"></param>
   <param name="szCountyAddress"></param>
   <param name="szState"></param>
   <param name="szCountry"></param>
   <param name="szUserid"></param>
   <param name="szProgramid"></param>
   <param name="jdDateupdated"></param>
   <param name="szWorkstationid"></param>
   <param name="mnTimelastupdated"></param>
   <param name="szNamealpha"></param>
</params>
<onError abort="yes"></onError>
</callMethod>
</jdeRequest>
```

The following is a sample GetEffectiveAddress jdeResponse.

```
<?xml version="1.0"?>

<!DOCTYPE jdeResponse>
<jdeResponse environment="DV7333"
             pwd="JDE"
             session="516.1029417972.68"
             type="callmethod"
             user="JDE">
  <callMethod app="BSE"
              name="GetEffectiveAddress"
              runOnError="no">
    <returnCode code="0"/>
    <params>
      <param name="mnAddressNumber">1001</param>
      <param name="jdDateBeginningEffective"/>
      <param name="cEffectiveDateExistence10"/>
       <param name="szAddressLine1">8055 Tufts Avenue, Suite 1331
</param>
      <param name="szAddressLine2">
</param>
      <param name="szAddressLine3">
</param>
      <param name="szAddressLine4">
</param>
      <param name="szZipCodePostal">80237       </param>
      <param name="szCity">Denver                    </param>
      <param name="szCountyAddress">                          </param>
      <param name="szState">CO</param>
      <param name="szCountry"/>
      <param name="szUserid"/>
      <param name="szProgramid"/>
      <param name="jdDateupdated"/>
      <param name="szWorkstationid"/>
      <param name="mnTimelastupdated">0</param>
      <param name="szNamealpha">J.D. Edwards & Company                  </param>
    </params>
  </callMethod>
</jdeResponse>
```

Issuing a Multiple-Function Request

The following example, GetEffectiveAddress, is a multiple-function call to J.D. Edwards OneWorld, and the result of this request is a standard jdeResponse document with multiple sections. In a multiple-function request, more than one callMethod within the XML object is specified.

Executing a Business Function with a Multiple-Function Call

The following is a sample Purchase Order in the jdeRequest format. The XML contains return parameter specifications, and file cleanup logic.

```
<?xml version='1.0' encoding='utf-8' ?>
<jdeRequest pwd='password' type='callmethod' user='user' session=''
environment='DV7333' sessionidle=''>
   <callMethod app='XMLTest' name='GetLocalComputerId'
      runOnError='no'>
   <params>
      <param name='szMachineKey' id='machineKey'></param>
   </params>
   <onError abort='yes'>
   </onError>
   </callMethod>
   <callMethod app='XMLTest' name='F4311InitializeCaching'
      runOnError='no'>
   <params>
      <param name='cUseWorkFiles'>2</param>
   </params>
   </callMethod>
   <callMethod app='XMLTest' name='F4311FSBeginDoc' runOnError='no'
      returnNullData='yes'>
   <params>
      <param name='mnJobNumber' id='jobNumber'></param>
      <param name='szComputerID' idref='machineKey'></param>
      <param name='cHeaderActionCode'>A</param>
      <param name='cProcessEdits'>1</param>
      <param name='cUpdateOrWriteToWorkFile'>2</param>
      <param name='cRecordWrittenToWorkFile'>0</param>
      <param name='szOrderCOmpany' id='orderCompany'>00200</param>
      <param name='szOrderType'>OP</param>
      <param name='szOrderSuffix'>000</param>
      <param name='szBranchPlant'>         M30</param>
      <param name='mnSupplierNumber'
         id='supplierNumber'>4343</param>
      <param name='mnShipToNumber'>0.0</param>
      <param name='jdOrderDate'>2000/03/02</param>
      <param name='cEvaluatedReceiptsFlag'>N</param>
      <param name='cCurrencyMode'>D</param>
      <param name='szTransactionCurrencyCode'>USD</param>
      <param name='mnCurrencyExchangeRate'>0.0</param>
      <param name='szOrderedPlacedBy'>SUBSTITUTE</param>
      <param name='szProgramID'>EP4310</param>
      <param name='szPurchaseOrderPrOptVersion'
         id='Version'>ZJDE0001</param>
      <param name='szUserID'>SUBSTITUTE</param>
      <param name='mnProcessID' id='processID'></param>
      <param name='mnTransactionID' id='transactionID'></param>
   </params>
   <onError abort='yes'>
   <callMethod app='XMLTest' name='F4311ClearWorkFiles'
      runOnError='yes' returnNullData='yes'>
   <params>
      <param name='szComputerID' idref='jobNumber'></param>
      <param name='mnJobNumber' idref='machineKey'></param>
      <param name='cClearHeaderFile'>1</param>
      <param name='cClearDetailFile'>1</param>
      <param name='mnLineNumber'>0</param>
      <param name='cUseWorkFiles'>2</param>
      <param name='mnProcessID' idref='processID'></param>
      <param name='mnTransactionID' idref='transactionID'></param>
   </params>
   </callMethod>
   </onError>
   </callMethod>
   <!-- This is the first EditLine entry -->
   <callMethod app='XMLTest' name='F4311EditLine' runOnError='yes'
      returnNullData='no'>
   <params>
      <param name='mnJobNumber' idref='jobNumber'></param>
      <param name='szComputerID' idref='machineKey'></param>
      <param name='cDetailActionCode'>A</param>
      <param name='cProcessEdits'>1</param>
      <param name='cUpdateOrWriteWorkFile'>2</param>
      <param name='cCurrencyProcessingFlag'>Y</param>
      <param name='szPurchaseOrderPrOptVersion'
      idref='version'></param>
      <param name='szOrderCompany' idref='orderCompany'></param>
      <param name='szOrderType'>OP</param>
      <param name='szOrderSuffix'>000</param>
      <param name='szBranchPlant'>         M30</param>
      <param name='mnSupplierNumber'
         idref='supplierNumber'></param>
      <param name='mnShipToNumber'>0.0</param>
      <param name='jdRequestedDate'>2000/03/02</param>
      <param name='jdTransactionDate'>2000/03/02</param>
      <param name='jdPromisedDate'>2000/03/02</param>
      <param name='jdGLDate'>2000/03/02</param>
      <param name='szUnformattedItemNumber'>1001</param>
      <param name='mnQuantityOrdered'>1</param>
      <param name='szDetailLineBranchPlant'>         M30</param>
      <param name='szLastStatus'>220</param>
      <param name='szNextStatus'>230</param>
      <param name='cEvaluatedReceipts'>N</param>
      <param name='szTransactionCurrencyCode'>USD</param>
      <param name='cSourceRequestingPOGeneration'>0</param>
      <param name='szProgramID'>XMLTest</param>
      <param name='szUserID'>SUBSTITUTE</param>
      <param name='szAgreementNumber'></param>
      <param name='mnAgreementSupplement'>0</param>
      <param name='jdEffectiveDate'></param>
      <param name='szPurchasingCostCenter'></param>
      <param name='szObjectAccount'></param>
      <param name='szSubsidiary'></param>
      <param name='mnProcessID' idref='processID'></param>
      <param name='mnTransactionID' idref='transactionID'></param>
   </params>
   </callMethod>
   <!-- This is the second EditLine entry -->
   <callMethod app='XMLTest' name='F4311EditLine' runOnError='yes'
      returnNullData='no'>
   <params>
      <param name='mnJobNumber' idref='jobNumber'></param>
      <param name='szComputerID' idref='machineKey'></param>
      <param name='cDetailActionCode'>A</param>
      <param name='cProcessEdits'>1</param>
      <param name='cUpdateOrWriteWorkFile'>2</param>
      <param name='cCurrencyProcessingFlag'>Y</param>
      <param name='szPurchaseOrderPrOptVersion'
```

```
      idref='version'></param>
   <param name='szOrderCompany' idref='orderCompany'></param>
   <param name='szOrderType'>OP</param>
   <param name='szOrderSuffix'>000</param>
   <param name='szBranchPlant'>         M30</param>
   <param name='mnSupplierNumber'
      idref='supplierNumber'></param>
   <param name='mnShipToNumber'>0.0</param>
   <param name='jdRequestedDate'>2000/03/02</param>
   <param name='jdTransactionDate'>2000/03/02</param>
   <param name='jdPromisedDate'>2000/03/02</param>
   <param name='jdGLDate'>2000/03/02</param>
   <param name='szUnformattedItemNumber'>2001</param>
   <param name='mnQuantityOrdered'>3</param>
   <param name='szDetailLineBranchPlant'>         M30</param>
   <param name='szLastStatus'>220</param>
   <param name='szNextStatus'>230</param>
   <param name='cEvaluatedReceipts'>N</param>
   <param name='szTransactionCurrencyCode'>USD</param>
   <param name='cSourceRequestingPOGeneration'>0</param>
   <param name='szProgramID'>XMLTest</param>
   <param name='szUserID'>SUBSTITUTE</param>
   <param name='szAgreementNumber'></param>
   <param name='mnAgreementSupplement'>0</param>
   <param name='jdEffectiveDate'></param>
   <param name='szPurchasingCostCenter'></param>
   <param name='szObjectAccount'></param>
   <param name='szSubsidiary'></param>
   <param name='mnProcessID' idref='processID'></param>
   <param name='mnTransactionID' idref='transactionID'></param>
</params>
</callMethod>
<callMethod app='XMLTest' name='F4311EditDoc' runOnError='no'
   returnNullData='no'>
<params>
   <param name='szOrderSuffix'>000</param>
   <param name='szComputerID' idref='machineKey'></param>
   <param name='mnJobnumber' idref='jobNumber'></param>
   <param name='mnAddressNumber' idref='supplierNumber'></param>
   <param name='szOrderType'>OP</param>
   <param name='szOrderCompany' idref='orderCompany'></param>
   <param name='szVersionProcOption' idref='version'></param>
   <param name='cActionCode'>A</param>
   <param name='mnProcessID' idref='processID'></param>
   <param name='mnTransactionID' idref='transactionID'></param>
</params>
</callMethod>
<callMethod app='XMLTest' name='F4311EndDoc' runOnError='no'
   returnNullData='no'>
<params>
   <param name='szComputerID' idref='machineKey'></param>
   <param name='mnJobNumber' idref='jobNumber'></param>
   <param name='szCallingApplicationName'>XMLTest</param>
   <param name='szVersion' idref='version'></param>
   <param name='szUserID'>SUBSTITUTE</param>
   <param name='mnOrderNumberAssigned'
      id='orderNumberAssigned'></param>
   <param name='cUseWorkFiles'>2</param>
   <param name='cConsolidateLines'>0</param>
   <param name='mnProcessID' idref='processID'></param>
```

```
      <param name='mnTransactionID' idref='transactionID'></param>
   </params>
   </callMethod>
   <returnParams runOnError='yes' returnNullData='no'>
      <param name='JobNumber' idref='machineKey'></param>
      <param name='ComputerID' idref='jobNumber'></param>
      <param name='OrderNumberAssigned'
         idref='orderNumberAssigned'></param>
   </returnParams>
   <!-- This is a default error catch for the entire document-->
   <onError abort='yes'>
   <callMethod app='XMLTest' name='F4311ClearWorkFiles'
      runOnError='yes' returnNullData='no'>
   <params>
      <param name='szComputerID' idref='jobNumber'></param>
      <param name='mnJobNumber' idref='machineKey'></param>
      <param name='cClearHeaderFile'>1</param>
      <param name='cClearDetailFile'>1</param>
      <param name='mnLineNumber'>0</param>
      <param name='cUseWorkFiles'>2</param>
      <param name='mnProcessID' idref='processID'></param>
      <param name='mnTransactionID' idref='transactionID'></param>
   </params>
   </callMethod>
   </onError>
</jdeRequest>
```

The Purchase Order response document contains individual return codes for each callMethod executed. In addition, this method returns the order number assigned for the Purchase Order.

```
<?xml version="1.0" encoding="utf-8" ?>

<jdeResponse environment="DV7333" user="JDE" type="callmethod" sessionidle=""
session="2612.1026498135.5" pwd="JDE">
   <callMethod name="GetLocalComputerId" runOnError="no" app="XMLTest">
   <returnCode code="0"/>
   <params>
      <param name="szMachineKey" id="machineKey">XEENT</param>
   </params>
   </callMethod>
   <callMethod name="F4311InitializeCaching" runOnError="no" app="XMLTest">
   <returnCode code="0"/>
   <params>
      <param name="cUseWorkFiles">2</param>
   </params>
   </callMethod>
   <callMethod name="F4311FSBeginDoc" returnNullData="yes" runOnError="no"
app="XMLTest">
   <returnCode code="0"/>
   <params>
      <param name="mnJobNumber" id="jobNumber">3</param>
      <param name="szComputerID" idref="machineKey">XEENT</param>
      <param name="cHeaderActionCode">1</param>
      <param name="cProcessEdits">1</param>
      <param name="cUpdateOrWriteToWorkFile">2</param>
      <param name="cRecordWrittenToWorkFile">1</param>
      <param name="cCurrencyProcessingFlag">Z</param>
      <param name="szOrderCOmpany" id="orderCompany">00200</param>
      <param name="mnOrderNumber">0</param>
      <param name="szOrderType">OP</param>
      <param name="szOrderSuffix">000</param>
      <param name="szBranchPlant">         M30</param>
      <param name="szOriginalOrderCompany"/>
      <param name="szOriginalOrderNumber"/>
      <param name="szOriginalOrderType"/>
      <param name="szRelatedOrderCompany"/>
      <param name="szRelatedOrderNumber"/>
      <param name="szRelatedOrderType"/>
      <param name="mnSupplierNumber" id="supplierNumber">17000</param>
       <param name="mnShipToNumber">6074</param>
       <param name="jdRequestedDate">2002/07/12</param>
       <param name="jdOrderDate">2000/03/02</param>
       <param name="jdPromisedDate">2002/07/12</param>
       <param name="jdCancelDate"/>
       <param name="szReference01"/>
       <param name="szReference02"/>
        <param name="szDeliveryInstructions01">
   </param>
         <param name="szDeliveryInstructions02">
   </param>
       <param name="szPrintMessage"/>
       <param name="szSupplierPriceGroup"/>
       <param name="szPaymentTerms"/>
       <param name="szTaxExplanationCode"/>
       <param name="szTaxRateArea"/>
       <param name="szTaxCertificate">                    </param>
       <param name="cAssociatedText"/>
       <param name="szHoldCode"/>
       <param name="szFreightHandlingCode"/>
       <param name="mnBuyerNumber">0</param>
       <param name="mnCarrierNumber">0</param>
       <param name="cEvaluatedReceiptsFlag">N</param>
       <param name="cSendMethod"/>
       <param name="szLandedCostRule">   </param>
       <param name="szApprovalRouteCode"/>
       <param name="mnChangeOrderNumber">0</param>
       <param name="cCurrencyMode">D</param>
       <param name="szTransactionCurrencyCode">USD</param>
       <param name="mnCurrencyExchangeRate">0</param>
       <param name="szOrderedPlacedBy">SUBSTITUTE</param>
       <param name="szOrderTakenBy"/>
       <param name="szProgramID">EP4310</param>
       <param name="szApprovalRoutePO"/>
       <param name="szPurchaseOrderPrOptVersion" id="Version">ZJDE0001</param>
       <param name="szBaseCurrencyCode">USD</param>
       <param name="szUserID">SUBSTITUTE</param>
       <param name="cAddNewLineToExistingOrder"/>
       <param name="idInternalVariables">0</param>
       <param name="cSourceOfData"/>
       <param name="mnSODOrderNumber">0</param>
       <param name="szSODOrderType"/>
       <param name="szSODOrderCompany"/>
       <param name="szSODOrderSuffix"/>
       <param name="mnRetainage">0</param>
       <param name="szDescription"/>
      <param name="szRemark"/>
      <param name="jdEffectiveDate"/>
      <param name="jdPhysicalCompletionDate"/>
      <param name="mnTriangulationRateFromCurrenc">0</param>
```

```
      <param name="mnTriangulationRateToCurrency">0</param>
      <param name="cCurrencyConversionMethod"/>
      <param name="szPriceAdjustmentScheduleN"/>
      <param name="cAIADocument"/>
      <param name="mnProcessID" id="processID">2612</param>
      <param name="mnTransactionID" id="transactionID">4</param>
   </params>
   </callMethod>
   <callMethod name="F4311EditLine" returnNullData="no" runOnError="yes"
app="XMLTest">
   <returnCode code="0"/>
   <params>
      <param name="mnJobNumber" idref="jobNumber">3</param>
      <param name="szComputerID" idref="machineKey">XEENT</param>
      <param name="mnOrderLineNumber">1</param>
      <param name="cDetailActionCode">1</param>
      <param name="cProcessEdits">1</param>
      <param name="cUpdateOrWriteWorkFile">2</param>
      <param name="cRecordWrittenToWorkFile">1</param>
      <param name="cCurrencyProcessingFlag">Y</param>
      <param name="szPurchaseOrderPrOptVersion"
         idref="version">ZJDE0001</param>
      <param name="szOrderCompany"
         idref="orderCompany">00200</param>
      <param name="szOrderType">OP</param>
      <param name="szOrderSuffix">000</param>
      <param name="szBranchPlant">         M30</param>
      <param name="mnSupplierNumber" idref="supplierNumber">17000</param>
      <param name="mnShipToNumber">6074</param>
      <param name="jdRequestedDate">2000/03/02</param>
      <param name="jdTransactionDate">2000/03/02</param>
      <param name="jdPromisedDate">2000/03/02</param>
      <param name="jdGLDate">2000/03/02</param>
        <param name="szUnformattedItemNumber">1001
   </param>
      <param name="mnQuantityOrdered">1</param>
      <param name="mnUnitPrice">32,1000</param>
      <param name="mnExtendedPrice">32,1</param>
      <param name="szLineType">S</param>
      <param name="szDescription1">Bike Rack - Trunk Mount</param>
     <param name="szDescription2">                              </param>
     <param name="szDetailLineBranchPlant">         M30</param>
     <param name="szLocation">  .   .             </param>
     <param name="szLotNumber">                              </param>
     <param name="szTransactionUoM">EA</param>
     <param name="szPurchasingUoM">EA</param>
     <param name="szLastStatus">220</param>
     <param name="szNextStatus">230</param>
     <param name="mnDiscountFactor">1</param>
     <param name="szInventoryPriceRule">        </param>
     <param name="szPrintMessage"> </param>
     <param name="cTaxable">Y</param>
     <param name="szGLClassCode">IN30</param>
     <param name="mnBuyerNumber">8444</param>
     <param name="szPurchasingCategoryCode1"> </param>
     <param name="szPurchasingCategoryCode2"> </param>
     <param name="szPurchasingCategoryCode3"> </param>
     <param name="szPurchasingCategoryCode4">240</param>
     <param name="szLandedCostRule"> </param>
     <param name="mnWeight">80</param>
```

```
  <param name="szWeightUoM">OZ</param>
  <param name="mnVolume">2,25</param>
  <param name="szVolumeUoM">FC</param>
  <param name="cEvaluatedReceipts">N</param>
  <param name="cInventoryInterface">Y</param>
  <param name="szTransactionCurrencyCode">USD</param>
  <param name="szBaseCurrencyCode">USD</param>
  <param name="cSourceRequestingPOGeneration">0</param>
  <param name="szProgramID">XMLTest</param>
  <param name="szUserID">SUBSTITUTE</param>
  <param name="szAgreementNumber"/>
  <param name="mnAgreementSupplement">0</param>
  <param name="jdEffectiveDate"/>
  <param name="szPurchasingCostCenter"/>
  <param name="szObjectAccount"/>
  <param name="szSubsidiary"/>
  <param name="cStockingType">P</param>
  <param name="mnProcessID" idref="processID">2612</param>
  <param name="mnTransactionID" idref="transactionID">4</param>
  <param name="mnIdentifierShortItem">60003</param>
</params>
</callMethod>
<callMethod name="F4311EditLine" returnNullData="no"
      runOnError="yes" app="XMLTest">
<returnCode code="0"/>
<params>
  <param name="mnJobNumber" idref="jobNumber">3</param>
  <param name="szComputerID" idref="machineKey">XEENT</param>
  <param name="mnOrderLineNumber">2</param>
  <param name="cDetailActionCode">1</param>
  <param name="cProcessEdits">1</param>
  <param name="cUpdateOrWriteWorkFile">2</param>
  <param name="cRecordWrittenToWorkFile">1</param>
  <param name="cCurrencyProcessingFlag">Y</param>
  <param name="szPurchaseOrderPrOptVersion"
      idref="version">ZJDE0001</param>
  <param name="szOrderCompany"
      idref="orderCompany">00200</param>
  <param name="szOrderType">OP</param>
  <param name="szOrderSuffix">000</param>
  <param name="szBranchPlant">         M30</param>
  <param name="mnSupplierNumber"
   idref="supplierNumber">17000</param>
  <param name="mnShipToNumber">6074</param>
  <param name="jdRequestedDate">2000/03/02</param>
  <param name="jdTransactionDate">2000/03/02</param>
  <param name="jdPromisedDate">2000/03/02</param>
  <param name="jdGLDate">2000/03/02</param>
  <param name="szUnformattedItemNumber">2001
  </param>
  <param name="mnQuantityOrdered">3</param>
  <param name="mnUnitPrice">164,0817</param>
  <param name="mnExtendedPrice">492,2451</param>
  <param name="szLineType">S</param>
  <param name="szDescription1">Cro-Moly Frame, Red           </param>
  <param name="szDescription2">                              </param>
  <param name="szDetailLineBranchPlant">         M30</param>
  <param name="szLocation">  .   .             </param>
  <param name="szLotNumber">                              </param>
  <param name="szTransactionUoM">EA</param>
```

```
     <param name="szPurchasingUoM">EA</param>
     <param name="szLastStatus">220</param>
     <param name="szNextStatus">230</param>
     <param name="mnDiscountFactor">1</param>
     <param name="szInventoryPriceRule">        </param>
     <param name="szPrintMessage"> </param>
     <param name="cTaxable">Y</param>
     <param name="szGLClassCode">IN30</param>
     <param name="szPurchasingCategoryCode1"> </param>
     <param name="szPurchasingCategoryCode2"> </param>
     <param name="szPurchasingCategoryCode3"> </param>
     <param name="szPurchasingCategoryCode4">200</param>
     <param name="szLandedCostRule"> </param>
     <param name="mnWeight">3</param>
     <param name="szWeightUoM">OZ</param>
     <param name="szVolumeUoM">FC</param>
     <param name="cEvaluatedReceipts">N</param>
     <param name="cInventoryInterface">Y</param>
     <param name="szTransactionCurrencyCode">USD</param>
     <param name="szBaseCurrencyCode">USD</param>
     <param name="cSourceRequestingPOGeneration">0</param>
     <param name="szProgramID">XMLTest</param>
     <param name="szUserID">SUBSTITUTE</param>
     <param name="szAgreementNumber"/>
     <param name="mnAgreementSupplement">0</param>
     <param name="jdEffectiveDate"/>
     <param name="szPurchasingCostCenter"/>
     <param name="szObjectAccount"/>
     <param name="szSubsidiary"/>
     <param name="cStockingType">M</param>
     <param name="mnProcessID" idref="processID">2612</param>
     <param name="mnTransactionID" idref="transactionID">4</param>
     <param name="mnIdentifierShortItem">60062</param>
   </params>
   </callMethod>
   <callMethod name="F4311EditDoc" returnNullData="no"
         runOnError="no" app="XMLTest">
   <returnCode code="0"/>
   <params>
     <param name="szOrderSuffix">000</param>
     <param name="szComputerID" idref="machineKey">XEENT</param>
     <param name="mnJobnumber" idref="jobNumber">3</param>
     <param name="mnAddressNumber"
         idref="supplierNumber">17000</param>
     <param name="szOrderType">OP</param>
     <param name="szOrderCompany"
         idref="orderCompany">00200</param>
     <param name="szVersionProcOption"
         idref="version">ZJDE0001</param>
     <param name="cActionCode">A</param>
     <param name="mnProcessID" idref="processID">2612</param>
     <param name="mnTransactionID" idref="transactionID">4</param>
</params>
   </callMethod>
   <callMethod name="F4311EndDoc" returnNullData="no"
         runOnError="no" app="XMLTest">
   <returnCode code="0"/>
   <params>
     <param name="szComputerID" idref="machineKey">XEENT</param>
     <param name="mnJobNumber" idref="jobNumber">3</param>
     <param name="szCallingApplicationName">XMLTest</param>
     <param name="szVersion" idref="version">ZJDE0001</param>
     <param name="szUserID">SUBSTITUTE</param>
     <param name="mnOrderNumberAssigned"
         id="orderNumberAssigned">4884</param>
     <param name="cUseWorkFiles">2</param>
     <param name="cConsolidateLines">0</param>
     <param name="mnProcessID" idref="processID">2612</param>
     <param name="mnTransactionID" idref="transactionID">4</param>
   </params>
   </callMethod>
   <returnParams>
     <param name="JobNumber" idref="machineKey">XEENT</param>
     <param name="ComputerID" idref="jobNumber">3</param>
     <param name="OrderNumberAssigned" idref="orderNumberAssigned">4884</param>
</returnParams>
</jdeResponse>
```

Sample Sales Order Request

The following is a sample Sales Order request.

Executing a Sales Order Request

The following is an example of a Sales Order request.

```
<?xml version='1.0' encoding='utf-8' ?>
<jdeRequest type='callmethod' user='JDE' pwd='JDE' environment='DV7333'>
   <callMethod name='GetLocalComputerId' app='XMLInterop'
         runOnError='no'>
   <params>
     <param name='szMachineKey' id='2'></param>
   </params>
   <onError abort='yes'>
   </onError>
   </callMethod>
   <callMethod name='F4211FSBeginDoc' app='XMLInterop'
         runOnError='no'>
   <params>
     <param name='mnCMJobNumber' id='1'></param>
     <param name='cCMDocAction'>A</param>
     <param name='cCMProcessEdits'>1</param>
     <param name='szCMComputerID' idref='2'></param>
     <param name='cCMUpdateWriteToWF'>2</param>
     <param name='szCMProgramID'>XMLInterop</param>
     <param name='szCMVersion'>ZJDE0001</param>
     <param name='szOrderType'>SO</param>
     <param name='szBusinessUnit'>         M30</param>
     <param name='mnAddressNumber'>4242</param>
     <param name='jdOrderDate'>2000/03/29</param>
     <param name='szReference'>10261</param>
     <param name='cApplyFreightYN'>Y</param>
     <param name='szCurrencyCode'></param>
     <param name='cWKSourceOfData'></param>
     <param name='cWKProcMode'></param>
     <param name='mnWKSuppressProcess'>0</param>
   </params>
   <onError abort='yes'>
   <callMethod name='F4211ClearWorkFile' app='XMLInterop'
      runOnError='yes'>
   <params>
<param name='mnJobNo' idref='1'></param>
<param name='szComputerID' idref='2'></param>
<param name='mnFromLineNo'>0</param>
<param name='mnThruLineNo'>0</param>
<param name='cClearHeaderWF'>2</param>
<param name='cClearDetailWF'>2</param>
<param name='szProgramID'>XMLInterop</param>
<param name='szCMVersion'>ZJDE0001</param>
   </params>
   </callMethod>
   </onError>
   </callMethod>
   <callMethod name='F4211FSEditLine' app='XMLInterop'
      runOnError='yes'>
   <params>
     <param name='mnCMJobNo' idref='1'></param>
     <param name='cCMLineAction'>A</param>
     <param name='cCMProcessEdits'>1</param>
     <param name='cCMWriteToWFFlag'>2</param>
     <param name='szCMComputerID' idref='2'></param>
<!-- param name='mnLineNo'>10261</param -->
     <param name='szItemNo'>1001</param>
     <param name='mnQtyOrdered'>1</param>
     <param name='cSalesTaxableYN'>N</param>
     <param name='szTransactionUOM'>EA</param>
     <param name='szCMProgramID'>XMLInterop</param>
     <param name='szCMVersion'>ZJDE0001</param>
     <param name='cWKSourceOfData'></param>
   </params>
   <onError abort='no'>
   </onError>
   </callMethod>
   <callMethod name='F4211FSEditLine' app='XMLInterop'
      runOnError='yes'>
   <params>
     <param name='mnCMJobNo' idref='1'></param>
     <param name='cCMLineAction'>A</param>
     <param name='cCMProcessEdits'>1</param>
     <param name='cCMWriteToWFFlag'>2</param>
     <param name='szCMComputerID' idref='2'></param>
<!-- param name='mnLineNo'>10262</param -->
     <param name='szItemNo'>1001</param>
     <param name='mnQtyOrdered'>10</param>
     <param name='cSalesTaxableYN'>N</param>
     <param name='szTransactionUOM'>EA</param>
     <param name='szCMProgramID'>XMLInterop</param>
     <param name='szCMVersion'>ZJDE0001</param>
     <param name='cWKSourceOfData'></param>
   </params>
   <onError abort='no'>
   </onError>
   </callMethod>
   <callMethod name='F4211FSEndDoc' app='XMLInterop'
      runOnError='no'>
   <params>
     <param name='mnCMJobNo' idref='1'></param>
     <param name='szCMComputerID' idref='2'></param>
     <param name='szCMProgramID'>XMLInterop</param>
     <param name='szCMVersion'>ZJDE0001</param>
     <param name='cCMUseWorkFiles'>2</param>
   </params>
   <onError abort='no'>
   <callMethod name='F4211ClearWorkFile' app='XMLInterop'
      runOnError='yes'>
<params>
     <param name='mnJobNo' idref='1'></param>
     <param name='szComputerID' idref='2'></param>
     <param name='mnFromLineNo'>0</param>
     <param name='mnThruLineNo'>0</param>
     <param name='cClearHeaderWF'>2</param>
     <param name='cClearDetailWF'>2</param>
     <param name='szProgramID'>XMLInterop</param>
     <param name='szCMVersion'>ZJDE0001</param>
   </params>
   </callMethod>
   </onError>
   </callMethod>
   <returnParams failureDestination='ERROR.Q'
      successDestination='SUCCESS.Q' runOnError='yes'>
   </returnParams>
   <onError abort='yes'>
   <callMethod name='F4211ClearWorkFile' app='XMLInterop'
      runOnError='yes'>
   <params>
     <param name='mnJobNo' idref='1'></param>
     <param name='szComputerID' idref='2'></param>
     <param name='mnFromLineNo'>0</param>
     <param name='mnThruLineNo'>0</param>
     <param name='cClearHeaderWF'>2</param>
     <param name='cClearDetailWF'>2</param>
     <param name='szProgramID'>XMLInterop</param>
     <param name='szCMVersion'>ZJDE0001</param>
   </params>
   </callMethod>
   </onError>
   </jdeRequest>
```

Sample Sales Order Response

This is the corresponding response document for the Sales Order request. There are error messages returned in the document. The error messages can be used within a workflow. For example:

```
<error code="2597">Warning: WARNING: Duplicate Customer Order Number
</error>
<error code="4136">Warning: Pick date is less than todays date</error>
```

Using the Sales Order Response

The following is the jdeResponse document.

```
<?xml version="1.0" encoding="utf-8" ?>
<jdeResponse environment="DV7333" user="JDE" type="callmethod" pwd="JDE">

  <callMethod name="GetLocalComputerId" runOnError="no"
      app="XMLInterop">
   <returnCode code="0"/>
   <params>
      <param name="szMachineKey" id="2">XEENT</param>
   </params>
```

```
</callMethod><callMethod name="F4211FSBeginDoc" runOnError="no"
   app="XMLInterop">
<returnCode code="1"/>
<params>
  <param name="mnCMJobNumber" id="1">3</param>
  <param name="cCMDocAction">A</param>
  <param name="cCMProcessEdits">1</param>
  <param name="szCMComputerID" idref="2">XEENT</param>
  <param name="cCMErrorConditions">1</param>
  <param name="cCMUpdateWriteToWF">2</param>
  <param name="szCMProgramID">XMLInterop</param>
  <param name="szCMVersion">ZJDE0001</param>
  <param name="szOrderCo">00200</param>
  <param name="szOrderType">SO</param>
  <param name="szBusinessUnit">              M30</param>
  <param name="mnAddressNumber">4242</param>
  <param name="mnShipToNo">4242</param>
  <param name="jdRequestedDate">2000/03/29</param>
  <param name="jdOrderDate">2000/03/29</param>
  <param name="jdPromisedDate">2000/03/29</param>
  <param name="szReference">10261</param>
  <param name="szDeliveryInstructions1">              </param>
  <param name="szDeliveryInstructions2">               </param>
  <param name="szPrintMesg">          </param>
  <param name="szPaymentTerm">   </param>
  <param name="cPaymentInstrument"> </param>
  <param name="mnTradeDiscount">,000</param>
  <param name="szTaxExplanationCode">S </param>
  <param name="szTaxArea">DEN       </param>
  <param name="szCertificate">                    </param>
  <param name="szHoldOrdersCode">  </param>
  <param name="cPricePickListYN">Y</param>
  <param name="szRouteCode">   </param>
  <param name="szStopCode">   </param>
  <param name="szZoneNumber">   </param>
  <param name="szFreightHandlingCode">   </param>
  <param name="cApplyFreightYN">Y</param>
  <param name="mnCommissionCode1">6001</param>
  <param name="mnCommissionRate1">5,000</param>
  <param name="mnCommissionRate2">,000</param>
  <param name="szWeightDisplayUOM">  </param>
  <param name="szVolumeDisplayUOM">  </param>
  <param name="cMode">D</param>
  <param name="szCurrencyCode">USD</param>
  <param name="jdDateUpdated">2002/07/12</param>
  <param name="szWKBaseCurrency">USD</param>
  <param name="cWKAdvancedPricingYN">N</param>
  <param name="szWKCreditMesg">  </param>
  <param name="szWKTempCreditMesg">  </param>
  <param name="cWKSourceOfData"/>
  <param name="cWKProcMode"/>
  <param name="mnWKSuppressProcess">0</param>
  <param name="szPricingGroup">PREFER  </param>
  <param name="mnProcessID">2252</param>
  <param name="mnTransactionID">4</param>
</params><errors><error code="2597">Warning: WARNING: Duplicate
   Customer Order Number</error><error code="4136">Warning: Pick
  date is less than todays date</error></errors>
</callMethod><callMethod name="F4211FSEditLine" runOnError="yes"
   app="XMLInterop">
```

```
<returnCode code="1"/><params>
     <param name="mnCMJobNo" idref="1">3</param>
     <param name="cCMLineAction">A</param>
     <param name="cCMProcessEdits">1</param>
     <param name="cCMWriteToWFFlag">2</param>
     <param name="cCMRecdWrittenToWF">1</param>
     <param name="szCMComputerID" idref="2">XEENT</param>
     <param name="cCMErrorConditions">1</param>
     <param name="szOrderCo">00200</param>
      <param name="szOrderType">SO</param>     <param name="szBusinessUnit">
      M30</param>
     <param name="mnShipToNo">4242</param>
     <param name="jdRequestedDate">2000/03/29</param>
     <param name="jdPromisedDate">2000/03/29</param>
     <param name="jdPromisedDlvryDate">2000/03/29</param>
     <param name="szItemNo">1001                     </param>
     <param name="szLocation">  .   .   </param>
     <param name="szDescription1">Bike Rack Trunk Mount </param>
     <param name="szDescription2">                       </param>
     <param name="szLineType">S</param>
     <param name="szLastStatus">900</param>
     <param name="szNextStatus">540</param>
     <param name="mnQtyOrdered">1</param>
     <param name="mnQtyBackordered">1</param>
     <param name="mnUnitPrice">44,99</param>
     <param name="mnUnitCost">32,1000</param>
     <param name="szPrintMesg">          </param>
     <param name="cPaymentInstrument"> </param>
     <param name="cSalesTaxableYN">N</param>
     <param name="cAssociatedText"> </param>
     <param name="szTransactionUOM">EA</param>
     <param name="szPricingUOM">EA</param>
     <param name="mnItemWeight">80</param>
     <param name="szWeightUOM">OZ</param>
     <param name="mnForeignUnitPrice">44,99</param>
     <param name="mnForeignUnitCost">32,1000</param>
     <param name="mnDiscountFactor">1</param>
     <param name="mnCMLineNo">1</param>
     <param name="szCMProgramID">XMLInterop</param>
     <param name="szCMVersion">ZJDE0001</param>
     <param name="mnSupplierNo">4343</param>
     <param name="mnWKOrderTotal">44,99</param>
     <param name="mnWKForeignOrderTotal">44,99</param>
     <param name="mnWKTotalCost">32,1</param>
     <param name="mnWKForeignTotalCost">32,1</param>
     <param name="cWKSourceOfData"/>
     <param name="cWKCheckAvailability">1</param>
     <param name="mnLastLineNoAssigned">1</param>
     <param name="cStockingType">P</param>
     <param name="cParentItmMethdOfPriceCalcn">1</param>
     <param name="mnShortItemNo">60003</param>
     <param name="szSalesOrderFlags">0</param>
     <param name="jdPriceEffectiveDate">2000/03/29</param>
     <param name="jdPromisedShip">2000/03/29</param>
     <param name="mnQuantityAvailable">-34</param>
     <param name="mnItemVolume_ITVL">2,25</param>
     <param name="szVolumeUOM_VLUM">FC</param>
     <param name="szRevenueBusinessUnit">  M30</param>
     <param name="mnProcessID">2252</param>
     <param name="mnTransactionID">4</param>
```

```
</params><errors><error code="030B">Warning: Order Quantity
   Exceeds what&apos;s Available</error></errors>
</callMethod><callMethod name="F4211FSEditLine" runOnError="yes"
   app="XMLInterop"><returnCode code="1"/><params>
  <param name="mnCMJobNo" idref="1">3</param>
  <param name="cCMLineAction">A</param>
  <param name="cCMProcessEdits">1</param>
  <param name="cCMWriteToWFFlag">2</param>
  <param name="cCMRecdWrittenToWF">1</param>
  <param name="szCMComputerID" idref="2">XEENT</param>
  <param name="cCMErrorConditions">1</param>
  <param name="szOrderCo">00200</param>
  <param name="szOrderType">SO</param>
  <param name="szBusinessUnit">         M30</param>
  <param name="mnShipToNo">4242</param>
  <param name="jdRequestedDate">2000/03/29</param>
  <param name="jdPromisedDate">2000/03/29</param>
  <param name="jdPromisedDlvryDate">2000/03/29</param>
  <param name="szItemNo">1001                     </param>
  <param name="szLocation">  .   .   </param>
  <param name="szDescription1">Bike Rack-Trunk Mount   </param>
  <param name="szDescription2">                        </param>
  <param name="szLineType">S</param>
  <param name="szLastStatus">900</param>
  <param name="szNextStatus">540</param>
  <param name="mnQtyOrdered">10</param>
  <param name="mnQtyBackordered">10</param>
  <param name="mnUnitPrice">44,99</param>
  <param name="mnUnitCost">32,1000</param>
  <param name="szPrintMesg">          </param>
  <param name="cPaymentInstrument"> </param>
  <param name="cSalesTaxableYN">N</param>
  <param name="cAssociatedText"> </param>
  <param name="szTransactionUOM">EA</param>
  <param name="szPricingUOM">EA</param>
  <param name="mnItemWeight">800</param>
  <param name="szWeightUOM">OZ</param>
  <param name="mnForeignUnitPrice">44,99</param>
  <param name="mnForeignUnitCost">32,1000</param>
  <param name="mnDiscountFactor">1</param>
  <param name="mnCMLineNo">2</param>
  <param name="szCMProgramID">XMLInterop</param>
  <param name="szCMVersion">ZJDE0001</param>
  <param name="mnSupplierNo">4343</param>
  <param name="mnWKOrderTotal">494,89</param>
  <param name="mnWKForeignOrderTotal">494,89</param>
  <param name="mnWKTotalCost">321</param>
  <param name="mnWKForeignTotalCost">321</param>
  <param name="cWKSourceOfData"/>
  <param name="cWKCheckAvailability">1</param>
  <param name="mnLastLineNoAssigned">2</param>
  <param name="cStockingType">P</param>
  <param name="cParentItmMethdOfPriceCalcn">1</param>
  <param name="mnShortItemNo">60003</param>
  <param name="szSalesOrderFlags">                 0   </param>
  <param name="jdPriceEffectiveDate">2000/03/29</param>
  <param name="jdPromisedShip">2000/03/29</param>
  <param name="mnQuantityAvailable">-44</param>
  <param name="mnItemVolume_ITVL">22,5</param>
  <param name="szVolumeUOM_VLUM">FC</param>
  <param name="szRevenueBusinessUnit">         M30</param>
  <param name="mnProcessID">2252</param>
  <param name="mnTransactionID">4</param>
</params><errors><error code="030B">Warning: Order Quantity
   Exceeds what&apos;s Available</error></errors>
</callMethod><callMethod name="F4211FSEndDoc" runOnError="no"
   app="XMLInterop"><returnCode code="0"/>
<params>
  <param name="mnCMJobNo" idref="1">3</param>
  <param name="mnSalesOrderNo">2623</param>
  <param name="szCMComputerID" idref="2">XEENT</param>
  <param name="cCMErrorCondition">0</param>
  <param name="szOrderType">SO</param>
  <param name="szKeyCompany">00200</param>
  <param name="mnOrderTotal">494,89</param>
  <param name="szWorkstationID">XEENT</param>
  <param name="szCMProgramID">XMLInterop</param>
  <param name="szCMVersion">ZJDE0001</param>
  <param name="mnTimeOfDay">174220</param>
  <param name="cCMUseWorkFiles">2</param>
  <param name="cCMProcessEdits">1</param>
  <param name="mnProcessID">2252</param>
  <param name="mnTransactionID">4</param>
</params>   </callMethod><returnParams failureDestination="ERROR.Q"
   successDestination="SUCCESS.Q">
</returnParams></jdeResponse>
```

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