Oracle®

Application Adapter for SAP R/3 User's Guide

WebLogic Server 10*g* Release 3 (10.3.1.0)

E15304-01

June 2009

ORACLE®

Oracle Application Adapter for SAP R/3 User's Guide, WebLogic Server 10*g* Release 3 (10.3.1.0)

E15304-01

Primary Author: Stefan Kostial

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

Contents

7 Troubleshooting and Error Messages

8 Advanced User Tools

A Configuring SAP R/3 for Inbound and Outbound Processing

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 SAP R/3 (WebLogic Server 10gr3) User's Guide 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 Adapter Concepts*
- *Oracle Application Adapter (WebLogic Server 10gr3) 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.

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

Convention	Meaning	Example
Italics	Italicized text indicates placeholders or variables for which you must provide 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

Oracle WebLogic Server connects to a SAP R/3 system through Oracle Application Adapter for SAP R/3 (WebLogic Server 10gr3) (Oracle Application Adapter for SAP R/3). Oracle Application Adapter for SAP R/3 provides connectivity and carries out interactions on a SAP R/3 system. This chapter discusses the following topics:

Adapter Features

Oracle Application Adapter for SAP R/3 is a remote function call adapter that provides a means to exchange real-time business data between SAP Enterprise Central Component (ECC) 5.0/6.0 systems and other application, database, or external business partner systems. The **adapter** enables external applications for inbound and outbound processing with SAP R/3. Oracle Application Adapter for SAP R/3 can be deployed as a J2EE Connector Architecture (J2CA) version 1.0 resource adapter. This deployment is referred to as Oracle Adapter J2CA. It can also be deployed as a Web services servlet and is referred to as Oracle Adapter Business Services Engine (BSE).

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

- Services (also known as outbound processing): Enable applications to call an SAP R/3 business object or business operation.
- Events (also known as inbound processing): Enable applications to access SAP R/3 data only when an event occurs.

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

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

Oracle Application Adapter for SAP R/3 provides:

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

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

See Also: *Oracle WebLogic Server Adapter Concepts*

Supported Versions and Platforms

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

- SAP R/3 Enterprise 47x100
- SAP R/3 Enterprise 47x200
- mySAP ERP Central Component (ECC) 5.0, deployed on SAP NetWeaver 2004
- mySAP ERP Central Component (ECC) 6.0, deployed on SAP NetWeaver 2004s
- SAP Java Connector (SAP JCo) Version 2.18.

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

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

Classical SAP Technologies for ABAP

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

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

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

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

> **Note:** Depending on the release or service pack installed, certain RFCs, for example, `RFC_CUSTOMER_GET`, may not exist in your particular SAP R/3 system. Therefore, the examples included in this document may not be relevant to your system. If this is the case, then you should use the examples as a general reference for adapter functionality and choose an RFC that exists within your SAP R/3 application environment.
>
> As described in SAP Release Note 109533, SAP Function Modules (RFCs) can be delivered with different release statuses. SAP supports only RFCs that are awarded with the Released for Customer status. There is no claim to the release independencies of the interfaces and the continued existence/functionality of the modules. For more information on the status of a specific function module, consult the SAP Service Marketplace.

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

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

Integration with SAP R/3

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

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

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

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

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

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

  The developer is freed from the specific details of the SAP R/3 interface (BAPI, RFC, IDoc) and the specific configuration details of the target SAP R/3 system.
- Adherence to SAP R/3 ABAP serialization rules and SAP R/3 Interface Repository standards published by SAP AG.

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

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

Adapter Architecture

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

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

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

Oracle Adapter Business Services Engine (BSE) Architecture

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

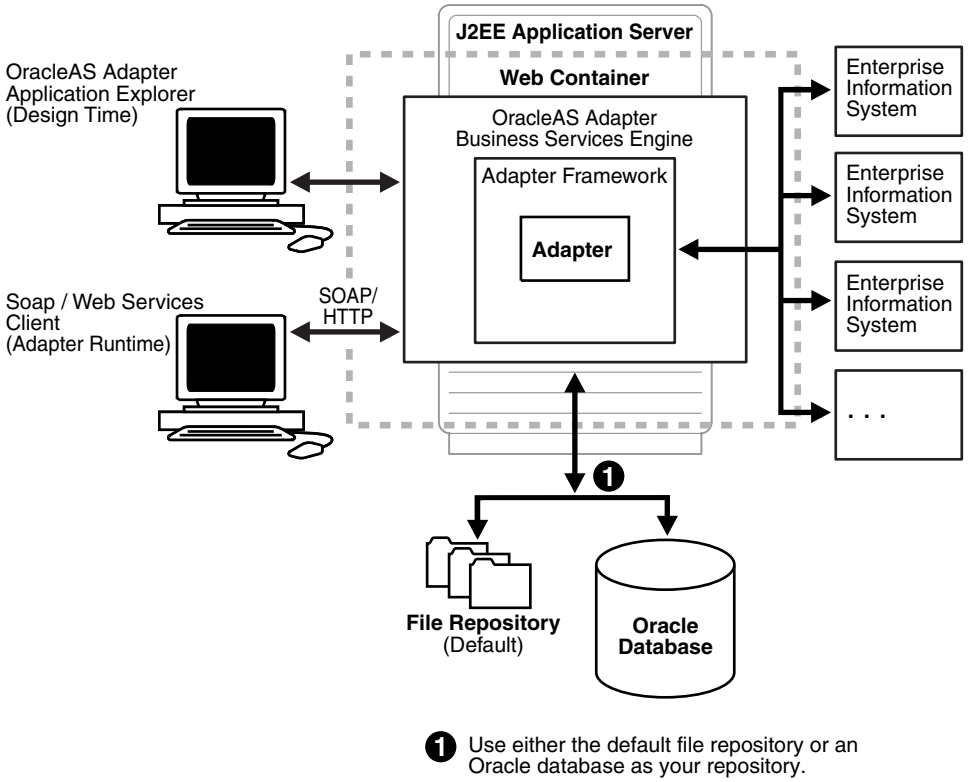
Application Explorer, a design-time tool deployed along with BSE, is used to configure adapter connections, browse EIS objects, configure services, and configure listeners to listen for EIS events. Metadata created while you perform these operations are stored in the repository by BSE.

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

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

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

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

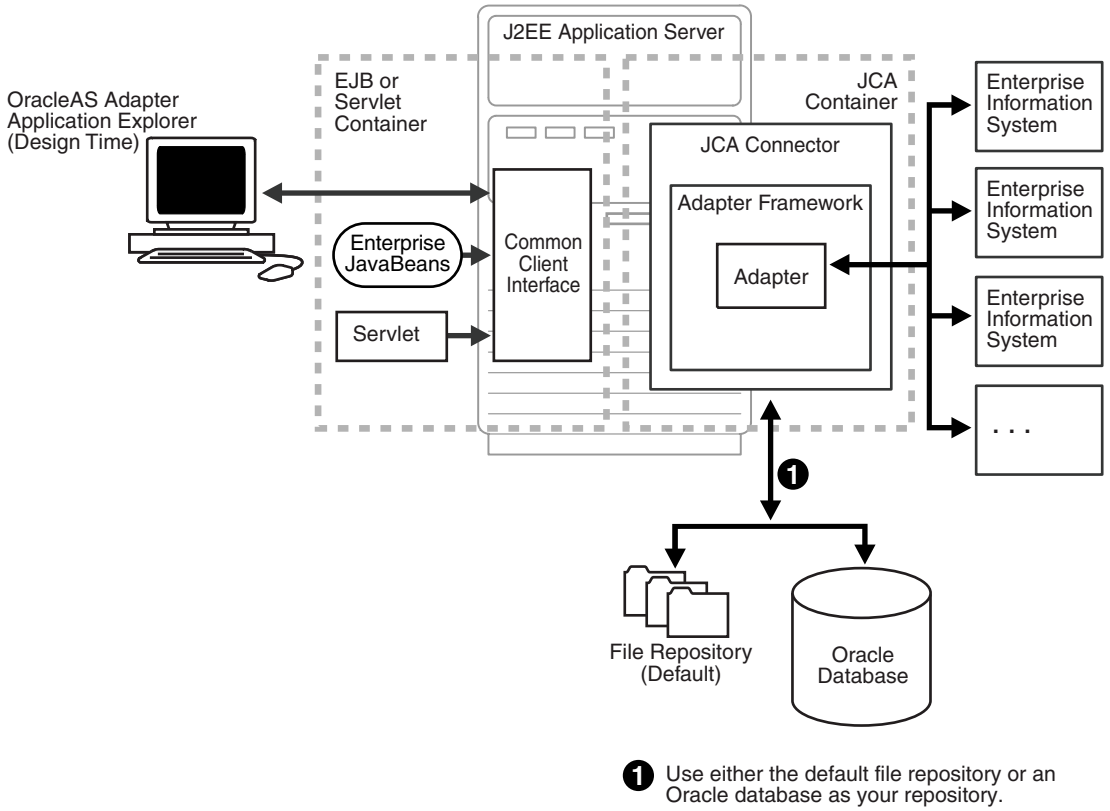


Oracle Adapter J2CA Architecture

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

The Oracle Adapter J2CA repository contains the list of EIS connection names and the associated connection parameters. The repository can be a file system or an Oracle database. It is deployed as a RAR file and has an associated deployment descriptor called `ra.xml`. You can create multiple connector factories by editing the Oracle WebLogic Server deployment descriptor `ra.xml`. See Chapter 4, "Oracle WebLogic Server Deployment and Integration" for more information.

Figure 1–2 Oracle Adapter J2CA Architecture



See Also:

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

BSE Versus Oracle Adapter J2CA Deployment

If you are using Oracle Application Adapter for SAP R/3 with Oracle Service Bus (OSB), please note that:

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

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

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

- Conforms more closely to the Service Oriented Architecture (SOA) model for building applications.

2. Oracle Adapter J2CA provides slightly better performance.

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

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

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

2

Getting Started

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

- Verifying the SAP Java Connector (SAP JCo)
- Identifying SAP R/3 Logon Parameters

Verifying the SAP Java Connector (SAP JCo)

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

Verifying SAP JCo on Windows Platforms

Perform the following steps to verify SAP JCo on Windows:

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

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

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

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

Verifying SAP JCo on UNIX Platforms

Perform the following steps to verify SAP JCo on UNIX:

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

```
$ java -jar sapjco.jar -stdout
```

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

```
---------------------------------------------------------------------
| SAP Java Connector (JCo) |
| Copyright (c) 2000-2005 SAP AG. All rights reserved.|
| Version Information |
---------------------------------------------------------------------
Java Runtime:
Operating System: SunOS 5.7 for sparc
Java VM: 1.4.0-beta3 Sun Microsystems Inc.
Java Codepage: ASCII
Versions:
JCo API: 2.1.8 (2006-12-11)
JCo middleware: 2.1.8 (2006-12-11)
JCo library: 2.1.8 (2006-12-11)
RFC library: 640.0.165
Paths:
JCo classes: /u4/fpgjpr/iWay55sm/lib/sapjco.jarJCo library:
/u4/fpgjpr/iWay55sm/lib/libsapjcorfc.so
RFC library: System-defined path
```

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

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

Identifying SAP R/3 Logon Parameters

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

User Parameters

The following table lists and describes User parameters.

Table 2–1

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

System Settings (Application Server) Parameters

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

Table 2–2

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

System Settings (Message Server) Parameters

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

Table 2–3

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

Connection Pool Parameters

The following table lists and describes Connection Pool parameters.

Table 2–4

Parameter	Description	Example	Comment
Connection pool	A name for a unique pool of client connections.	"foo"	A pool is a set of client-connections to a certain destination with the same logon data. The pool automatically creates new connections to the specified remote system or returns an already existing one. The reusing of existing connections can increase the performance of your application by avoiding logging on to the remote server.
Connection Pool Size	Maximum number of connections for the pool.	10	Sets the maximum number of connections that can be allocated from the pool.
Connection Timeout	Maximum time to keep open a free connection (in minutes).	5	Connections that have not been used for at least the connection timeout interval will be closed.
Connection Wait Time	Maximum wait for a free connection.	30 (seconds)	Sets the maximum time to wait in a connection request for a free connection. If the pool is exhausted, and there is still no connection available after the specified time, a JCO.Exception with the key JCO_ERROR_RESOURCE will be thrown. The default value is 30 seconds.

SAP Gateway Parameters

The following table lists and describes SAP Gateway parameters.

Table 2–5

Parameter	Description	Example	Comment
SAP Gateway Host	Enter the name of a SAP Gateway server.	"isdsrv2"	The SAP Gateway carries out CPI-C services within the SAP world, which are based on TCP/IP. These services enable SAP Systems and external programs to communicate with one another.
SAP Gateway Service	Enter the service name (usually a compound of the service name and system number).	Sapgw00	Service name on the gateway host.
Program ID	A program identifier that has been specified on the SAP Gateway server (case sensitive).	"S1PROG"	Unique identifier for your communication session specified by your system administrator. The value entered in this field must match the one exposed on the gateway.

ALE Parameters

The following table lists and describes ALE parameters.

Table 2–6

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

Global Processing Parameters

The following table lists and describes Global Processing parameters.

Table 2–7

Parameter	Description	Example	Comment
Error Handling	Specifies the error handling method of the adapter.	Selection, see next column.	**Creates error document** writes an exception document with the full error text to the output destination. **Throws exception** creates a java exception, this may or may not display the full error text depending on the underlying component error.
Commit with Wait	Specifies the commit behavior.	Selection, see next column.	Off - default Sends Commit Request to Application Server at the end of the document. If there is a commit error it will not be reflected back (Optimal performance). On (checked) - waits for a full database server commit at the end of the document before returning. Commit errors are reflected back to the adapter level (slowest performance). See your SAP DB admin for your site's recommended setting.

Table 2–7 (Cont.)

Parameter	Description	Example	Comment
TRACE	Turns on the SAP Java connectors trace behavior.	Selection, see next column.	Off default - only hard errors are written to the trace file (dev rfc.trc) in append mode. ON - individual rfc*.trc and JCO*.trc are written for each request. Useful in finding errors, not recommended in a productive system.

SNC Parameters

The following table lists and describes SNC parameters.

Table 2–8

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

3

Configuring Oracle Application Adapter for SAP R/3

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

This chapter discusses the following topics:

Starting Application Explorer

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 are ready to define new targets to your SAP R/3 system.

Configuring Repository Settings

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

Configuring the Database Repository for BSE

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

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

To configure BSE:

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

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

   `wls_home`\erp-adapters\etc

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

   ```
   C:\wls_home\erp-adapters\etc>sqlplus

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

   Enter user-name: scott
   Enter password: scott1

   Connected to:
   Oracle Database 10g Enterprise Edition Release 10.1.0.2.0 - Production
   With the Partitioning, OLAP and Data Mining options

   SQL>@ iwse.ora
   ```

2. Open the following page in your browser:

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

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

   For example,

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

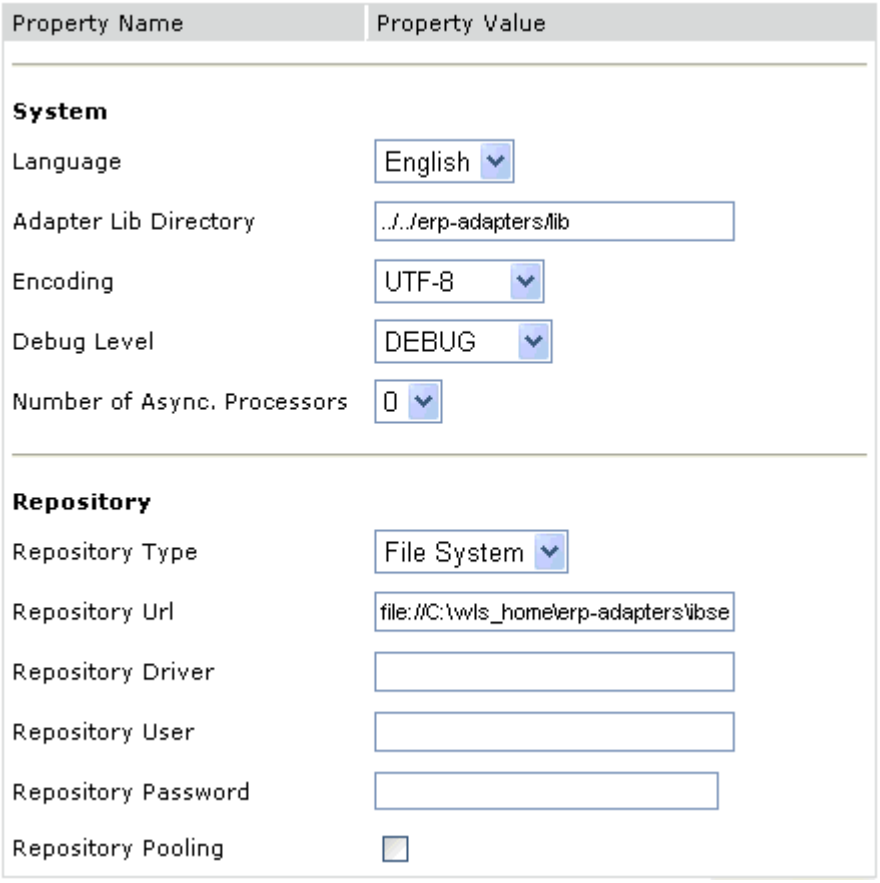
   > **Note:** If you are accessing this page for the first time, it may take longer to load.

3. Log on when prompted.

   Enter the user ID and password, for example:

- User name: weblogic
- Password: weblogic

The BSE configuration page is displayed.



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

   wls_home\erp-adapters\lib

   Where wls_home is the directory where Oracle WebLogic Server is installed.

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

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

5. Click **Save**.

Configuring BSE System Settings

To configure BSE system settings:

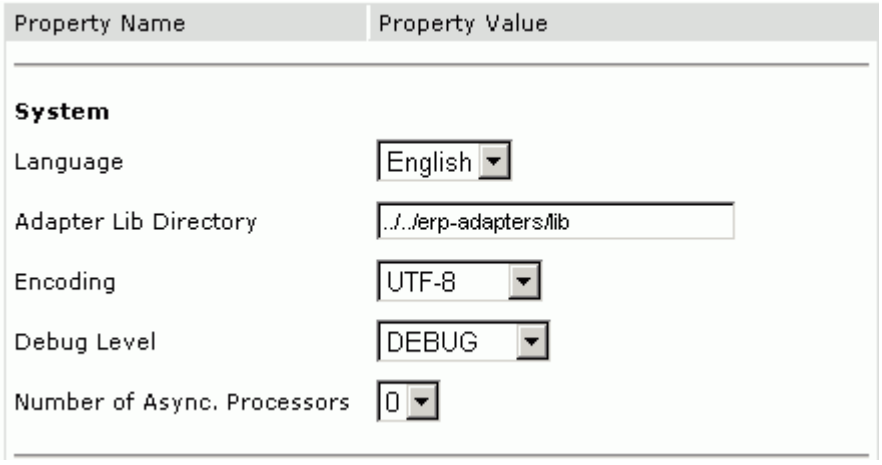
1. Display the **BSE configuration** page in a browser:

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

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

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

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

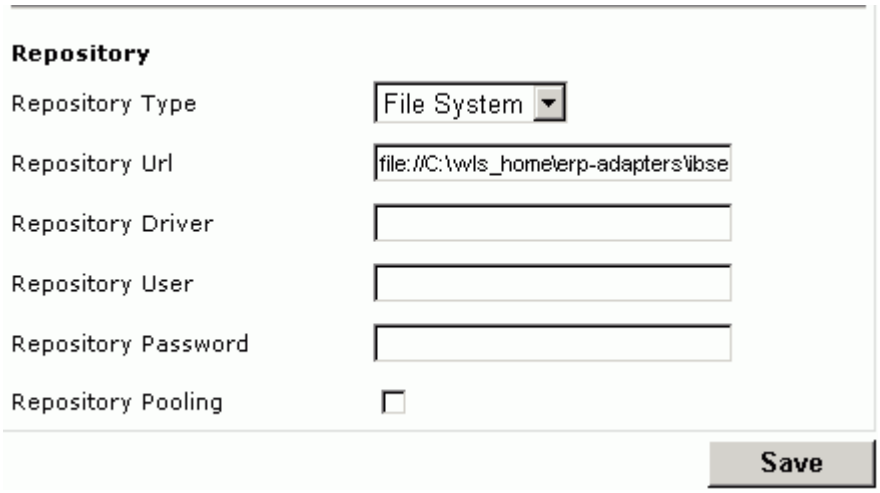


2. Configure the system settings.

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

Parameter	Description
Language	Specify the required language.
Adapter Lib Directory	Enter the full path to the directory where the adapter jar files reside.
Encoding	Only UTF-8 is supported.
Debug Level	Specify the debug level from 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 all fields and check boxes for the Repository pane.



3. Configure the repository settings.

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

   See "Configuring a File System Repository" on page 3-5 for more information.

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

Parameter	Description
Repository Type	Select one of the following repositories from the list: ■ Oracle ■ File (Do not use for BSE in production environments.)
Repository URL	Enter the JDBC URL to use when opening a connection to the database. For example, the following repository URL format is used when connecting to Oracle: `jdbc:oracle:thin:@host name:port;SID`
Repository Driver	Provide the JDBC driver class to use when opening a connection to the database (optional). For example, the following repository driver format is used when connecting to Oracle: `oracle.jdbc.driver.OracleDriver`
Repository User	Enter a valid user ID to use when opening a connection to the database.
Repository Password	Enter a valid password that is associated with the user ID.
Repository Pooling	If selected, repository pooling 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 the Database Repository for J2CA

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

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

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

   wls_home\erp-adapters\etc

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

```
C:\wls_home\erp-adapters\etc>sqlplus

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

Enter user-name: scott
Enter password: scott1

Connected to:
Oracle Database 10g Enterprise Edition Release 10.1.0.2.0 - Production
With the Partitioning, OLAP and Data Mining options

SQL>@ iwse.ora
```

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

   wls_home\erp-adapters\config\J2CA_SampleConfig

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

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

```
iwafjca.repo.url=jdbc:oracle:thin:@90.0.0.51:1521:orcl
```

```
iwafjca.repo.user=scott

iwafjca.repo.password=scott1
```

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

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

4. Navigate to the following directory:

   WLS_HOME\erp-adapters\iwafjca.rar\META-INF
5. Open the `ra.xml` file in a text editor.
6. Provide the JDBC connection information as a value for the `IWAYRepo_URL` property.
7. Provide a valid user name for the `IWAYRepo_User` property.
8. Provide a valid password for the `IWAYRepo_Password` property.
9. 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 is displayed.
3. Type a password in the Password field and click OK.

   An encrypted version of the password displays in the Encryption field.
4. Copy the password.
5. In the jcatransport.properties file, which is used during design time, replace the existing password with the encrypted value only if you are using a database repository.

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

   ```
   iwafjca.log.level=DEBUG
   iwafjca.repo.url=jdbc:oracle:thin:@172.30.166.100:1521:orcl
   iwafjca.repo.user=scott
   ```

```
iwafjca.repo.password=ENCR (3189319731832973218312931643233321232 27)
```

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

Creating a Repository Configuration

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

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

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

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

Creating a Configuration for BSE

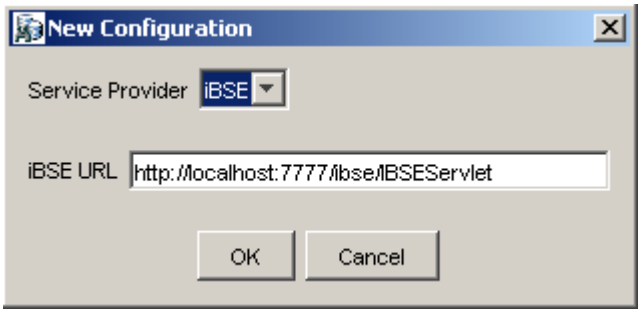
To create a configuration for BSE using Application Explorer, you must first define a new configuration.

Defining a New Configuration for BSE

To define a new configuration for BSE:

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

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



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 with the following format:

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

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

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

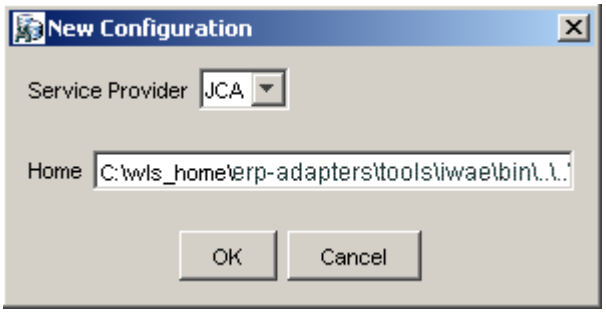
Defining a New Configuration for J2CA

To define a new configuration for J2CA:

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

   The New Configuration dialog box is displayed.

2. Enter a name for the new configuration, for example, SampleConfig, and click **OK**.



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

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

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

   Where wls_home is the directory where Oracle WebLogic Server is installed and configuration_name is the name of the configuration you created; for example, SampleConfig.

HTTP Repository Connection

J2CA users can create an HTTP repository connection, which enables them to generate and store WSDL documents remotely. Perform the following steps to create an HTTP repository connection in Application Explorer. To use the HTTP repository, 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 the Oracle WebLogic Server for run time purposes.

Connecting to a BSE or J2CA Configuration

To connect to a new configuration:

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

   Nodes appear for Adapters, Events, and Business Services (also known as Web services). The Business Services node is only available for BSE configurations. If you are connected to a J2CA configuration, you will not see the Business Services node.

   Events are not applicable when using a BSE configuration. You can configure events using a J2CA configuration only. As a result, you can disregard the Events node that appears for a BSE configuration.

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

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

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

Establishing a Connection (Target) for SAP R/3

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

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

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

Defining a Target to SAP R/3

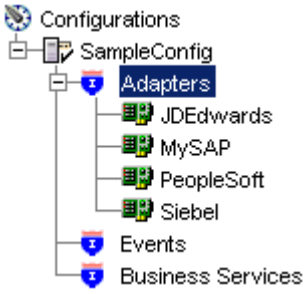
To connect to SAP R/3 for the first time, you must define a new target. Oracle Application Adapter for SAP R/3 supports SAP R/3 standard security and the additional protocol of SNC. Once connected to the SAP R/3 application server, application security is managed by user ID, roles and profiles. For more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

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

To define a target:

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

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



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

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

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

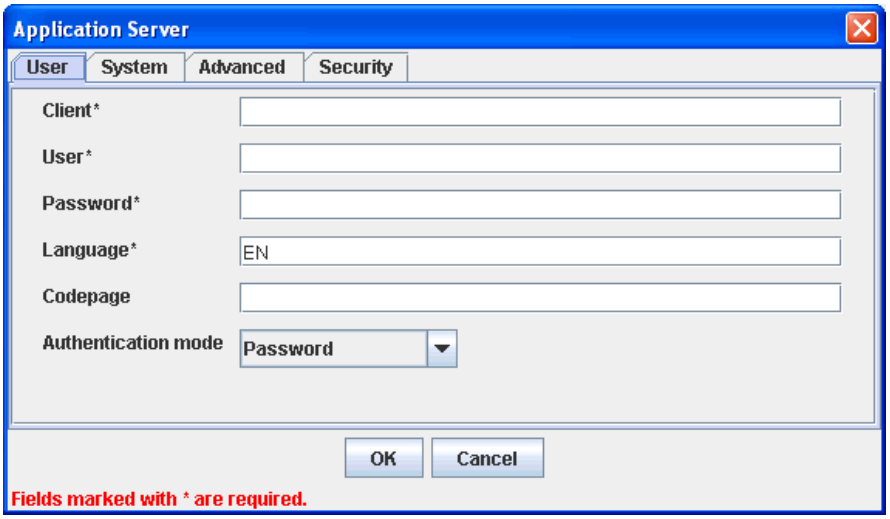
   b. In the **Description** field, enter a description for the target (optional).

   c. From the **Type** list, select the type of target you are connecting to. The supported target types include **Message Server** or **Application Server** (default).

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

3. Click **OK**.

   The Application Server dialog box is displayed.



   The following tabs are available:

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

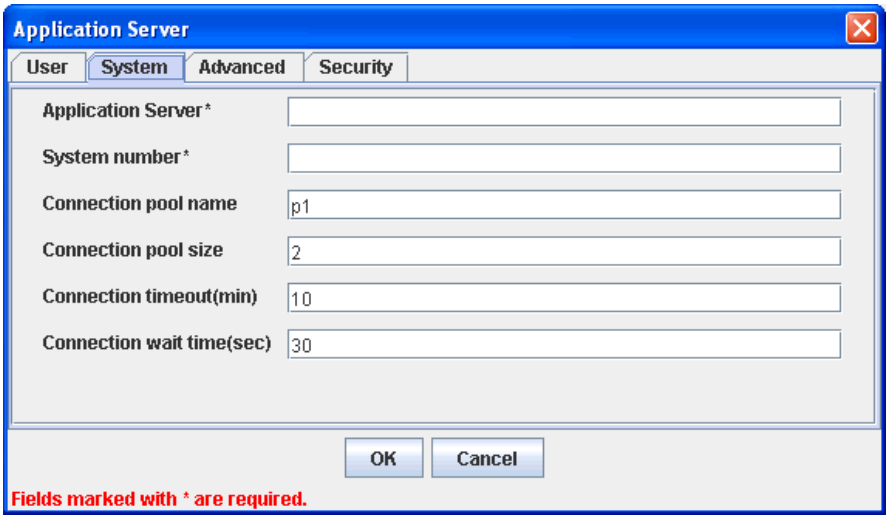
4. For the **User** tab (required), enter the appropriate information for your SAP R/3 target based on the information in the following table.

Table 3–1 User Tab Parameters

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

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

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



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

Table 3–2 System Tab Parameters

Target Parameter	Description
Application Server	The host name or IP address for the computer that is hosting the SAP R/3 application.
System number	The system number defined to SAP R/3 for client communications.
Connection pool name	The name of your SAP R/3 connection pool. A default value, p1, is already provided.
Connection pool size	The number of client connections in a pool you want to make available to SAP R/3 for Web service calls. A default connection pool size of 2 is available by default. **Important:** The default value of 1 does not create a connection pool. Instead, a single SAP R/3connection with sequential processing is shared. A pooled connection invokes multiple connections to SAP R/3 with parallel processing. If you are using Application Explorer to create Web services, the connection pool size value is used by your Web service during run-time. As a result, ensure that the connection pool size is sufficient for your purposes.
Connection timeout(min)	The timeout value for your connection pool in minutes. The default value is 10 minutes.

Table 3–2 (Cont.) System Tab Parameters

Target Parameter	Description
Connection wait time(sec)	The wait time for your connection pool in seconds. The default value is 30 seconds. Connections to an ERP server take up valuable resources on both the client and the remote server. You can create a pool of connections to minimize the resource and time constraints. In estimating the size of the pool, you may calculate pool size by the amount of server resources to be consumed, the number and size of the documents to be received, and the size of your Java Virtual Machine. The section of SAP documentation "Memory Management (BC-CST-MM)" explains in detail the resources required on the SAP R/3 system.

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

   The Advanced tab enables you to specify your EDI and IDoc versions, and configure error handling.

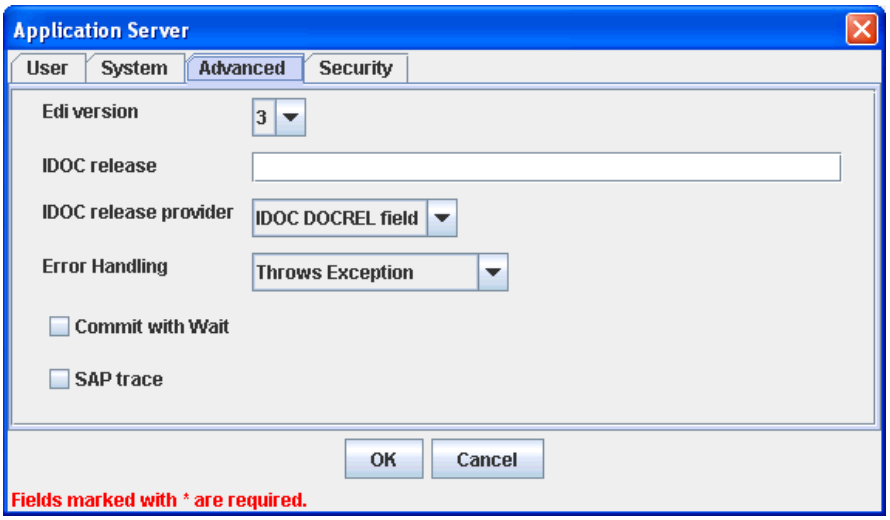


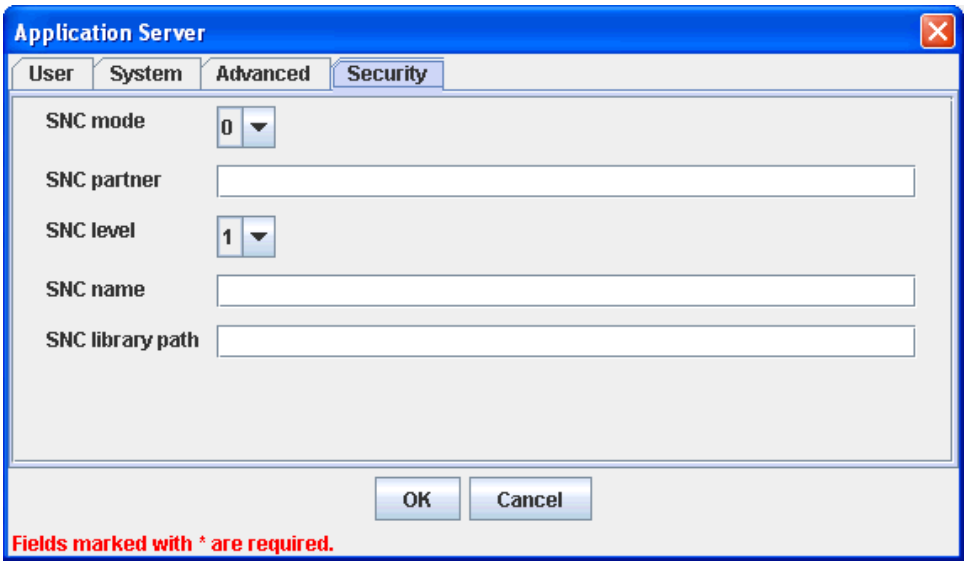
Table 3–3 Advanced Tab Parameters

Target Parameter	Description
Edi version	The Electronic Data Interchange (EDI) document version you are using with the adapter. Version 3 is the default value.
IDOC release	The IDOC versioning you want to use for your connection.
IDOC release provider	The IDOC release provider for your connection. Select **IDOC DOCREL field** (default), **SAP release**, or **user input** from the drop-down list.

Table 3–3 (Cont.) Advanced Tab Parameters

Target Parameter	Description
Error Handling	From the list In the event of an exception, you can select **Creates Error Document** or **Throws Exception**. To receive more detailed error messages, select **Creates Error Document**. As a rule: ■ If your application is **Java centric**, select **Throws Exception** so that code components can catch the exception and react accordingly. ■ If your application is **document based**, select **Creates Document** to create an XML document that contains the Java exception. It is up to your application to read the XML document and obtain the error.
Commit with Wait	If a high degree of accuracy is required in your application, select the **Commit with Wait** check box. The adapter waits until all records are physically written to the database before returning from the function call. The "Commit With Wait" has a performance impact on adapter performance, so consider carefully before selecting it. For more information about the commit behavior of BAPIs, see the SAP documentation under "BAPI Programming Guide and Reference (CA-BFA)." All SAP Business Objects that change data must commit work to the database. Some BAPIs developed in version 3.1 of the R/3 system use an internal commit behavior, and their commit behavior cannot be changed by the adapter. As soon as they are called, they commit the work they did. BAPIs developed since release 3.1 use the external commit method. The adapter issues a commit command, and the commit is put in the database queue. If there is an application error in the first part of the commit, the error message "Posting could not be carried out" is returned, and the adapter rolls back the transaction. If in writing to the database, a database error occurs, a short dump is issued in the database records of SAP, but no message is returned to the adapter about the failure. This option is disabled by default.
SAP trace	Select this option to enable traces.

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



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

Table 3–4 Security Tab Parameters

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

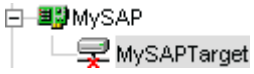
SNC provides protection for the communication links between the distributed components of an SAP R/3 system. Using SNC, SAP R/3 can support products that adhere to the GSS-API Version 2 standard. SNC supports application level (end-to-end security), Smartcard authentication, and single sign-on.

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

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

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

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



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

See "Creating XML Schemas" on page 3-19 for information on how to create schemas for the adapter.

Connecting to a Defined SAP R/3 Target

To connect to an existing target:

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

   The Connection dialog box displays the values you entered for connection parameters.
4. Verify your connection parameters.
5. Provide the correct password.
6. Right-click the target name and select **Connect**.

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

Managing a Connection to SAP R/3

To manage SAP R/3 connections, you can:

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

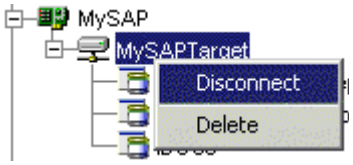
  Although you can maintain multiple open connections to different transaction processing systems, it is recommended to disconnect from connections not in use.
- Edit a target.

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

Disconnecting from a Connection to SAP R/3

To disconnect a target:

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



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

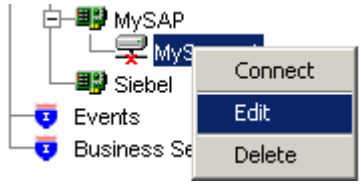
The x icon appears, indicating that the node is disconnected.

Modifying Connection Parameters

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

To edit a target:

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



The Application Server dialog box displays the target connection information.

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

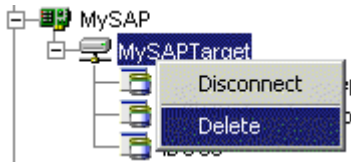
Deleting a Connection to SAP R/3

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

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

To delete a connection to SAP R/3:

1. Locate the target you want to delete.
2. Right-click the target (for example, SAPTarget), and select **Delete**.



The node disappears from the list of available connections.

Viewing Application System Objects

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

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

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

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

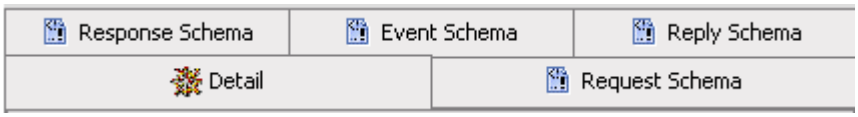
Creating XML Schemas

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

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

1. Connect to a SAP R/3 target as described in "Connecting to a Defined SAP R/3 Target" on page 3-17.
2. Expand the **Business Object Repository** node.
3. Click the icon to the left of the **Financial Accounting** node.
4. Scroll down and click the icon to the left of the **CompanyCode** business object.
5. Scroll down and select the BAPI named `GetDetail`.

   The following screen appears on the right.



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

Generating WSDL (J2CA Configurations Only)

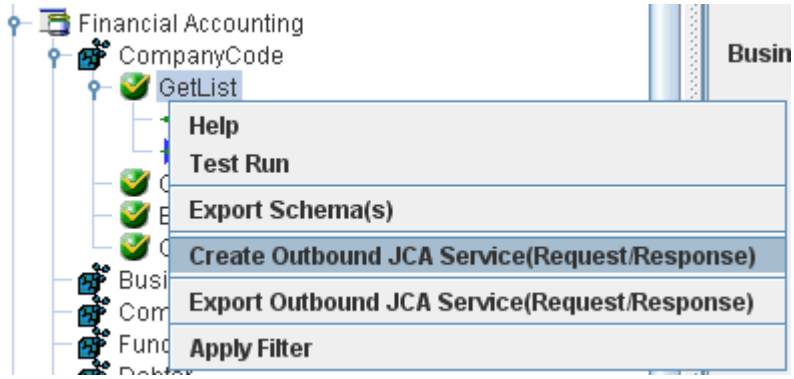
The Web Service Definition Language (WSDL) description of a service enables you to make the service available to other services within a host server. You use Application Explorer to create both request-response (outbound) and event notification (inbound) J2CA services of the adapter.

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

To generate a WSDL file for request-response service:

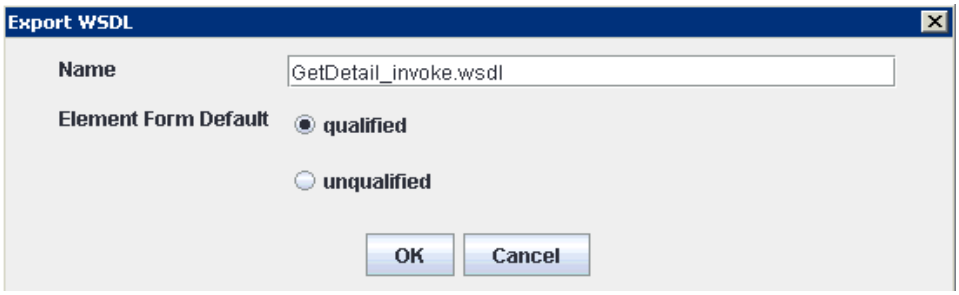
1. After you create a schema, right-click the respective object.

   The following menu is displayed:



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

   The Export WSDL dialog box is displayed.



3. Accept the default name for the file.

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

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

5. Click **OK**.

   The WSDL file is saved in the specified location.

The procedure for generating WSDL for event notification is similar to request-response. To generate WSDL for event notification, you must first create a channel for every event. See Generating WSDL for Event Notification on page 5-24 for a detailed example.

Publishing a WSDL

After you browse the SAP R/3 business object repository, you can publish the specific WSDL document for use with Oracle Service Bus. Make sure that the classpath has been configured with the required .jar files, as described in the *Oracle Application Server Adapters Installation Guide*.

The following section describes how to publish a WSDL using Application Explorer.

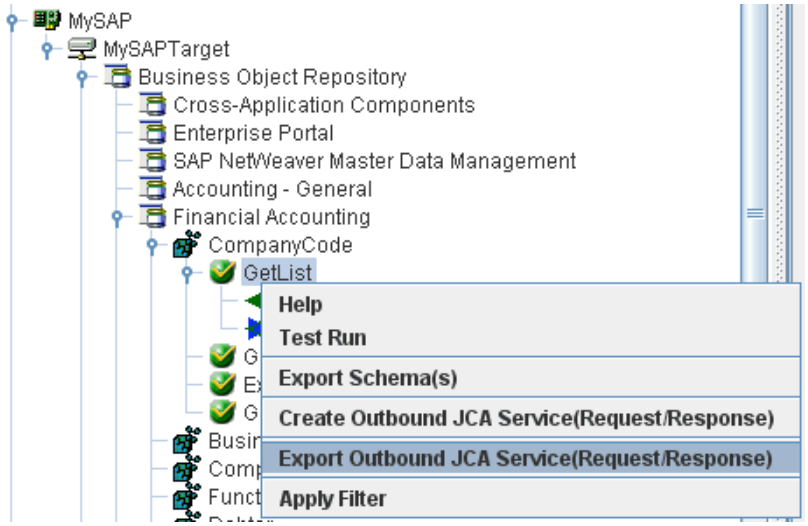
Note: Only users with Group Membership types set to *Administrators* in Oracle Service Bus can publish WSDL files using Application Explorer. For example, a user with the Group Membership type set to *IntegrationDeployers* cannot publish a WSDL file.

Publishing a WSDL

To publish a WSDL:

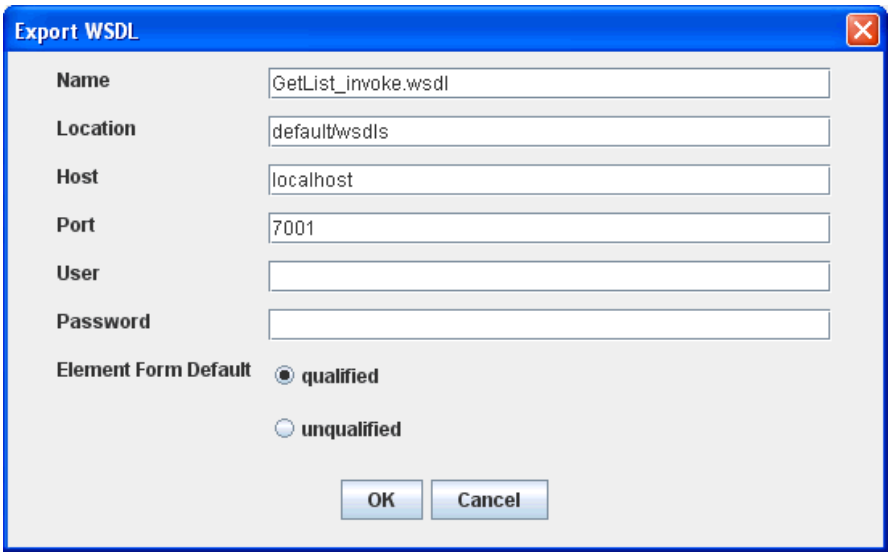
1. After you create a schema, right-click the respective object.

   The following menu is displayed.



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

   The Export WSDL dialog box is displayed.



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 forward 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)

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

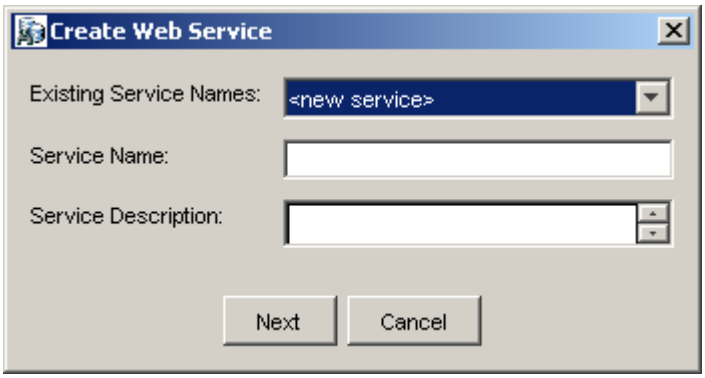
Note: In a J2EE Connector Architecture (J2CA) implementation of the adapter, Web services are not available. When the adapter is deployed to use the Oracle Adapter J2CA, the Common Client Interface provides integration services using the adapter.

Creating a Web Service

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

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

   The Create Web Service dialog box is displayed. You can add the business function as a method for a new Web service or as a method for an existing one.



Perform the following steps:

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

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

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

4. Click **Next**.

   The License and Method dialog box is displayed.

   Provide the following information:

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

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

   c. In the **Method Description** field, enter a brief description of the method.

   d. In the **DTD Directory** field, specify the location of the DTD you want to use.

5. Click **OK**.

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

Testing a Web Service

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

To test a Web service:

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

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

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

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

5. Enter the appropriate input.

6. Click **Invoke**.

   The BSE Servlet page displays the results.

Identity Propagation

If you test or run a Web service using a third party XML editor, the Username and Password values that you specify in the SOAP header must be valid and are used to connect to SAP R/3. The user name and password values that you provided for SAP R/3 during target creation using Application Explorer are overwritten for this Web service request. The following is a sample SOAP header that is included in the WSDL file for a Web service:

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

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

Oracle ERP Adapter WSDL Definition Limitations

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

Qualified/Unqualified Options

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

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

   A correct response document is received.

Namespaces

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

1. Create a Web service based on a generated WSDL.
2. Send XML data where all namespaces have been removed to this Web service.

   A correct response document is received.

Unchecked Data

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

1. Create a Web service for the *RFC_SYSTEM_INFO* function module.

2. Invoke this Web service (RFC_SYSTEM_INFO) with XML data pertaining to BAPI_COMPANYCODE_GETLIST.

   A response document is received with data for BAPI_COMPANYCODE_GETLIST.

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

Configuring an Event Adapter

Events are generated by activity in a database or in an application system. You can use events to trigger an action in your application. For example, an update to a database can reflect an update to customer information. If your application must perform when this happens, your application is a consumer of this event.

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

Note: If you are using a J2CA configuration, you must create a new channel for every different event object and select this channel when creating an inbound service. Creating a channel is required for J2CA configurations only. In addition, each channel must be associated with a unique SAP R/3 Program ID. For example, if you are working with MATMAS and DEBMAS, then two separate channels are required for each object and two unique SAP R/3 Program IDs.

A **channel** represents configured connections to particular instances of back-end systems. See "Creating and Editing a Channel" on page 3-25 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.

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

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

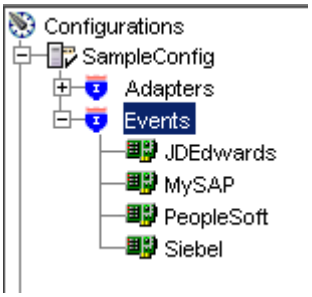
Events are not applicable when using a BSE configuration. You can configure events using a J2CA configuration only.

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

Creating a Channel

To create a channel:

1. Click the **Events** node.

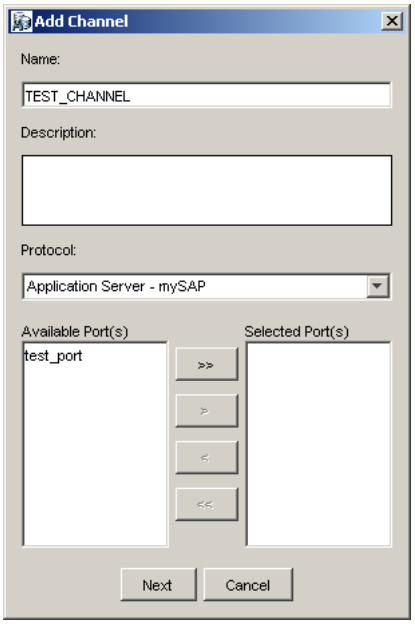


2. Expand the **MySAP** node.

   The ports and channels nodes appear in the left pane.

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

   The Add Channel dialog box is displayed.



   Provide the following information:

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

   b. Enter a brief description.

   c. From the **Protocol** list, select **Application Server - mySAP** or **Message Server - mySAP**.

4. Click **Next**.

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

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

- Advanced

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

Table 3–5 User Tab Parameters

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

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

Table 3–6 System Tab Parameters

Target Parameter	Description
Gateway host	A host name for the SAP R/3 Gateway.
Gateway service	A service for the SAP R/3 Gateway.
Program ID of the server	A SAP R/3 program ID you want to use for this channel.
Message Server	A host name for the message server.
R/3 name	A SAP R/3 name.
Server group	A SAP R/3 server group.

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

Table 3–7 Security Tab Parameters

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

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

Table 3–8 Advanced Tab Parameters

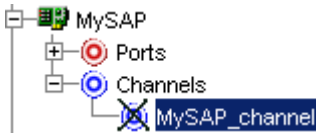
Target Parameter	Description
IDOC Format	Select an IDOC type from the list.

Table 3–8 (Cont.) Advanced Tab Parameters

Target Parameter	Description
IDOC release	The IDOC versioning you want to use for your connection.
IDOC release provider	The IDOC release provider for your connection. Select **IDOC DOCREL field** (default), **SAP release**, or **user input** from the drop-down list.
SAP trace	Select this option to enable traces.
Processing Mode	Select the type of synchronous processing from the list. Possible values include **REQUEST** and **REQUEST_ RESPONSE**.

9. Click **OK**.

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

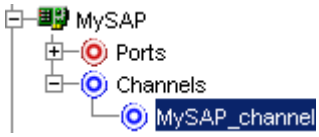


   An X over the icon indicates that the channel is currently disconnected. You must start the channel to activate your event configuration.

> **Note:** If you are using Oracle Application Adapter for SAP R/3 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).

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

    The channel you created becomes active.



    The X over the icon disappears.

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

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

Editing a Channel

To edit a channel:

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

   The Edit Channel pane is displayed.

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

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

Deleting a Channel

To delete a channel:

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

   The channel disappears from the list in the left pane.

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

4

Oracle WebLogic Server Deployment and Integration

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

This chapter discusses the following topics:

- Adapter Integration with OracleWLS
- Deployment of Adapter
- Updating Adapter Configuration

See Also:

- *Oracle WebLogic Server Adapter Concepts*

Adapter Integration with OracleWLS

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

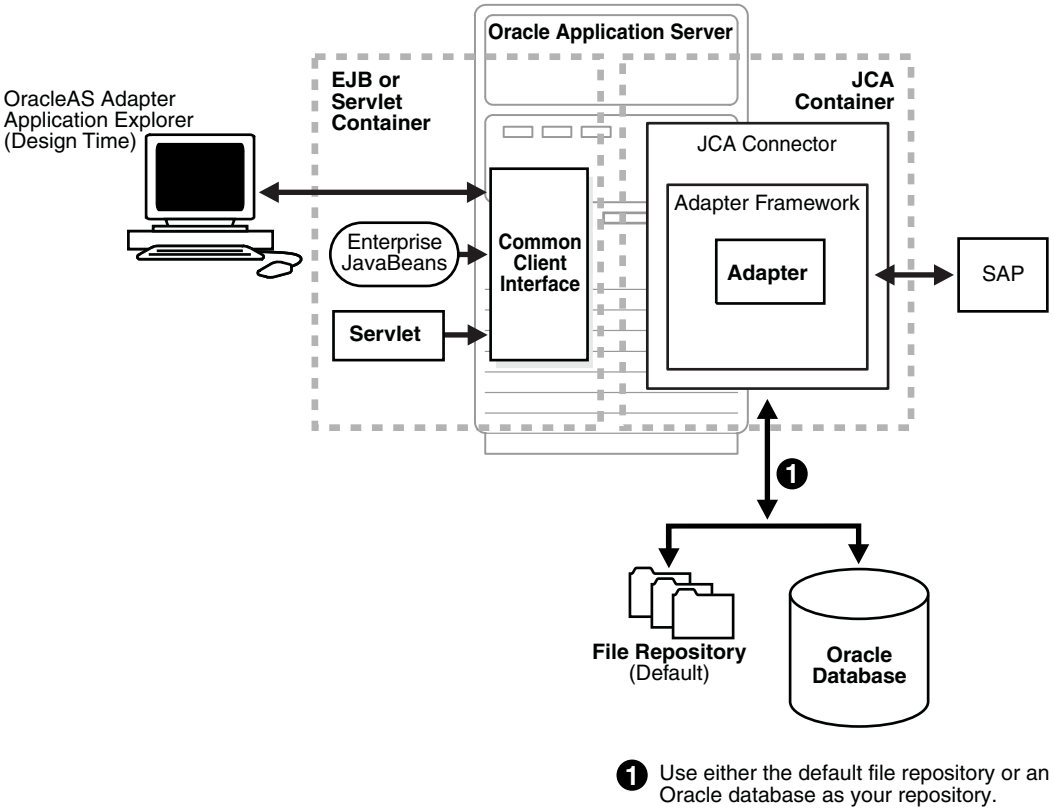
See Also:

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

Deployment of Adapter

Figure 4–1 shows deployment of the J2CA Connector to the Oracle Application Server. In a run-time service scenario, an Enterprise Java Bean, servlet, or Java program client makes CCI calls to J2CA resource adapters. The adapters process the calls as requests and send them to the EIS. The EIS response is then sent back to the client.

Figure 4–1 Oracle Application Server J2CA Architecture



See Also:

- *Oracle WebLogic Server Adapter Concepts*

Updating Adapter Configuration

During the J2CA deployment of Oracle Application Adapter for SAP R/3, OracleWLS generates a deployment descriptor called `ra.xml`, located in *`wls_home`*`\erp-adapters\iwafjca.rar\META-INF`.

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

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

Creating a Managed Connector Factory Object

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

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE connector PUBLIC '-//Sun Microsystems, Inc.//DTD Connector 1.0//EN'
'http://java.sun.com/dtd/connector_1_0.dtd'>
<connector>
```

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

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

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

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

<connection-impl-class>com.ibi.afjca.cci.IWAFConnection</connection-impl-class>
    <transaction-support>NoTransaction</transaction-support>
    <config-property>
      <config-property-name>AdapterName</config-property-name>
      <config-property-type>java.lang.String</config-property-type>
      <config-property-value></config-property-value>
    </config-property>
    <config-property>
      <config-property-name>Config</config-property-name>
      <config-property-type>java.lang.String</config-property-type>
      <config-property-value></config-property-value>
    </config-property>
    <config-property>
      <config-property-name>IWayHome</config-property-name>
      <config-property-type>java.lang.String</config-property-type>
      <config-property-value>C:\bea\erp-adapters\</config-property-value>
    </config-property>
    <config-property>
      <config-property-name>IWayConfig</config-property-name>
      <config-property-type>java.lang.String</config-property-type>
      <config-property-value>jca_sample</config-property-value>
    </config-property>
    <config-property>
      <config-property-name>IWayRepoDriver</config-property-name>
      <config-property-type>java.lang.String</config-property-type>
      <config-property-value></config-property-value>
    </config-property>
    <config-property>
      <config-property-name>IWayRepoURL</config-property-name>
      <config-property-type>java.lang.String</config-property-type>
      <config-property-value></config-property-value>
    </config-property>
    <config-property>
      <config-property-name>IWayRepoUser</config-property-name>
      <config-property-type>java.lang.String</config-property-type>
      <config-property-value></config-property-value>
    </config-property>
    <config-property>
      <config-property-name>IWayRepoPassword</config-property-name>
      <config-property-type>java.lang.String</config-property-type>
```

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

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

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

Parameter Name	Description
IWayHome	The base installation directory for the OracleAS packaged application adapter.
IWayConfig	The adapter configuration name as defined in Application Explorer. For example, Oracle Application Adapter for SAP R/3 has a preconfigured `jca_sample` configuration in Application Explorer.
IWayRepoURL	The URL to use when opening a connection to the database. This is necessary only when using an Oracle database as the repository.
IWayRepoUser	User name to use when connecting to the database. This is necessary only when using an Oracle database as the repository.
IWayRepoPassword	Password. If provided, it overwrites configuration. This is necessary only when using an Oracle database as the repository.
loglevel	It overwrites the level set by the ManagedConnectorFactory property.

Creating Multiple Managed Connector Factory Objects

To establish multiple managed connector factory objects, you must edit the `weblogic-ra.xml` file and add more `<connection-instance>` nodes. This file is located in *`wls_home`*`\erp-adapters\iwafjca.rar\META-INF`. For example, the first jca_configuration in Application Explorer is represented in the `weblogic-ra.xml` file as follows:

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

```
        </default-connection-properties>
        <connection-definition-group>

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

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

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

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

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

<connection-factory-interface>javax.resource.cci.ConnectionFactory</connection-fac
tory-interface>
            <connection-instance>
                <jndi-name>eis/OracleJCAAdapter/DefaultConnection</jndi-name>
            </connection-instance>
            <connection-instance>
                <jndi-name>eis/OracleJCAAdapter/DefaultConnection1</jndi-name>
                <connection-properties>
                <properties>
                <property>
<name>IWayHome</name>
<value>C:\bea\erp-adapters\</value>
                </property>
                <property>
                <name>IWayConfig</name>
                <value>jca_sample2</value>
                </property>
                <property>
    <name>IWayRepoURL</name>
    <value></value>
                </property>
                <property>
      <name>IWayRepoUser</name>
      <value></value>
                </property>
                <property>
      <name>IWayRepoPassword</name>
      <value></value>
                </property>
                <property>
      <name>LogLevel</name>
```

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

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

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

Modifying WSDL Files for Additional Connection Factory Values

Application Explorer generates outbound WSDL files using the default connection factory name `eis/OracleJCAAdapter/DefaultConnection`. If you created additional connection factories, the WSDLs generated for the additional configuration and connection factory should be changed to reflect the location field of the `jca:address` section in the outbound WSDLs. The default outbound WSDL for the Oracle Application Adapter for SAP R/3 with a configuration of `isdsrv2_conn2` is shown in the following example.

Notice that the outbound WSDL has the following default connection factory: `eis/OracleJCAAdapter/DefaultConnection`

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

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

For example:

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

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

5

Configuring Outbound Processing Using Oracle Service Bus (BSE Configuration)

To integrate with Oracle Service Bus (OSB), Oracle Application Adapter for SAP R/3 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.

Oracle Application Adapter for SAP R/3 integrates seamlessly with Oracle Service Bus (OSB) to facilitate Web service integration. OSB is based on the Service-Oriented Architecture (SOA). It consumes adapter services exposed as Web Service Definition Language (WSDL) documents.

For demonstration purposes, this chapter uses an SAP R/3 BAPI for outbound processing.

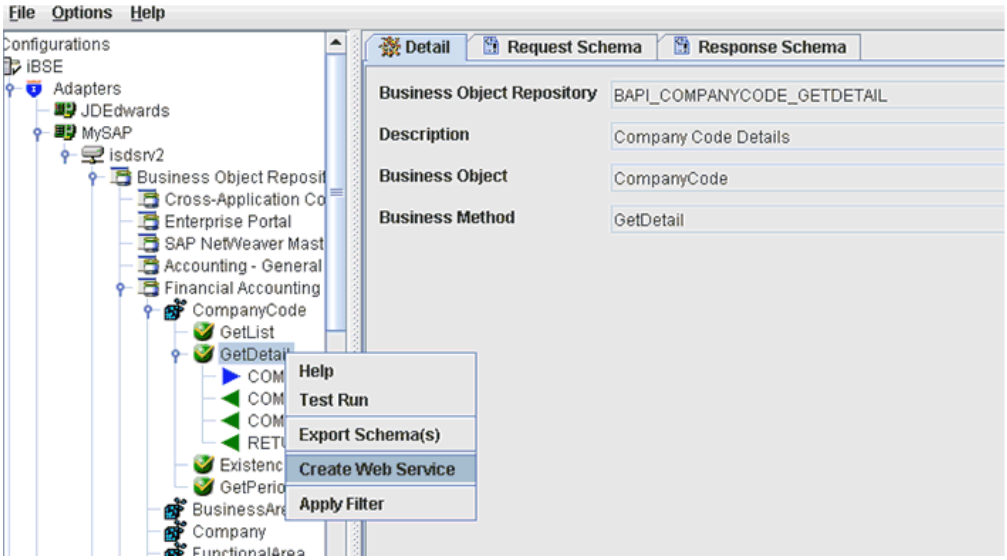
This chapter includes the following topics:

Publishing a WSDL From Application Explorer to Oracle Service Bus

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

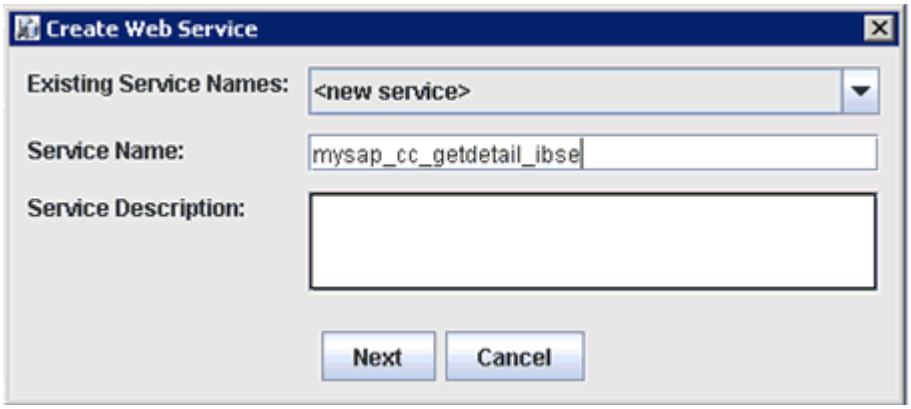
1. Start Application Explorer, connect to a BSE configuration, and connect to a SAP R/3 target.

   For more information, see Chapter 3, "Configuring Oracle Application Adapter for SAP R/3".



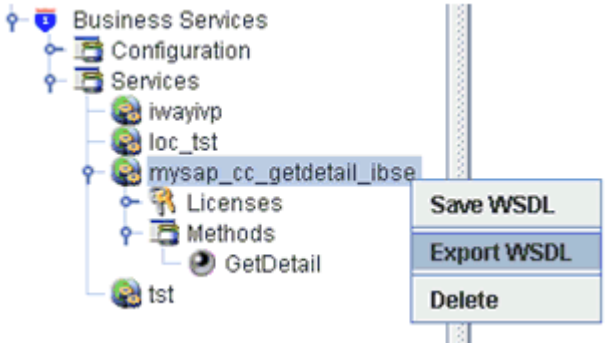
2. Expand the **Business Object Repository** node, **Financial Accounting** node, and the **CompanyCode** business object.
3. Right-click the **GetDetail** method and select **Create Web Service** from the context menu.

   The Create Web Service dialog box is displayed.



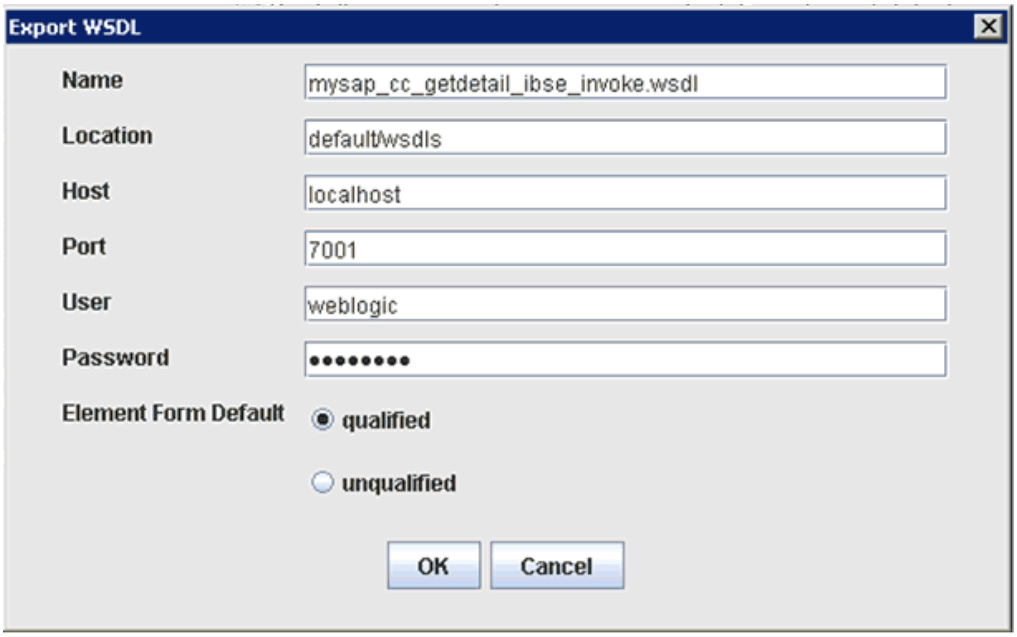
4. Enter a service name and click **Next**.
5. Click **OK** on the next dialog box that is displayed.

   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.



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

   The location is composed of an Oracle Service Bus project name and optionally, one or more folder names. The project name and any folder names must be separated by a forward slash character "/".
9. In the Host field, enter the name of the machine where Oracle WebLogic Server is running.
10. In the Port field, enter the port for the domain you are using.
11. In the User field, enter your username to access Oracle Service Bus.
12. In the Password field, enter your password to access Oracle Service Bus.
13. 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.

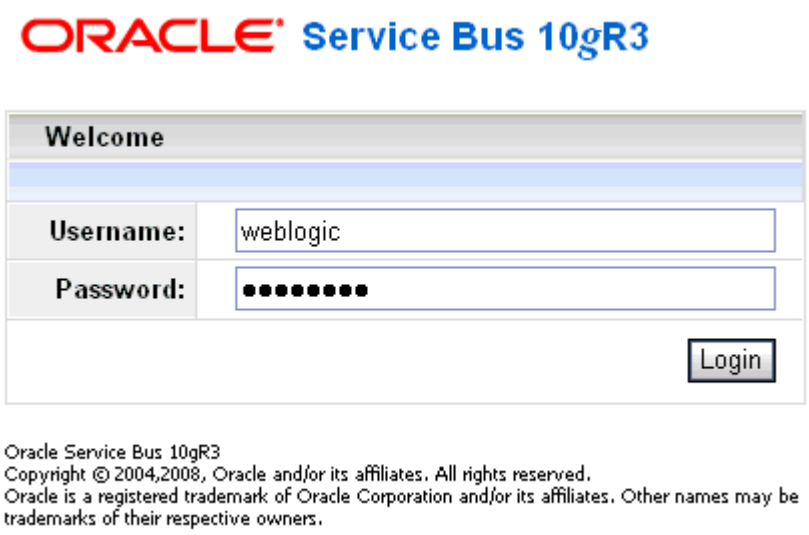
1. Start the Oracle WebLogic Server for the Oracle WebLogic Server domain that you have configured.

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

   `http://hostname:port/sbconsole`

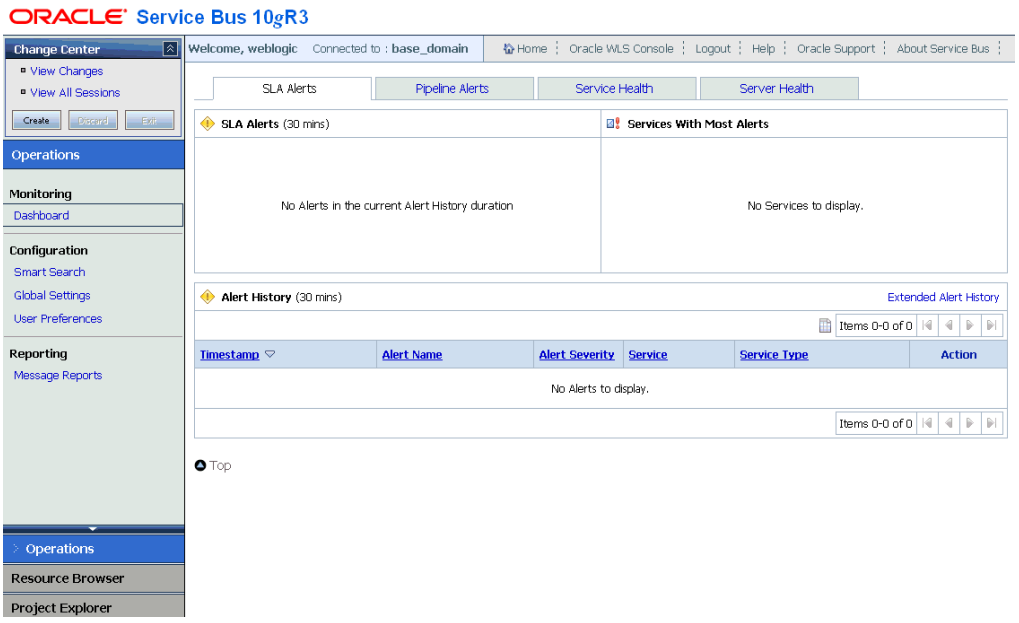
   Where *hostname* is the name of the machine where Oracle WebLogic Server is running and *port* is the port for the domain you are using. The port for the default domain is 7001.

   The Oracle Service Bus Console logon page is displayed.



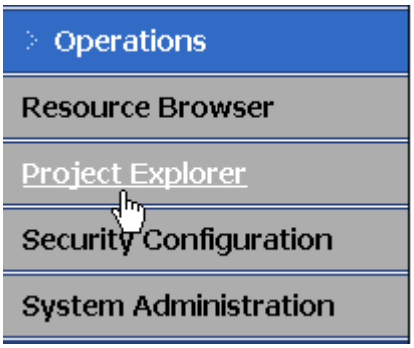
3. Log on to the Oracle Service Bus Console using a valid user name and password.

   The Oracle Service Bus Console home page is displayed.

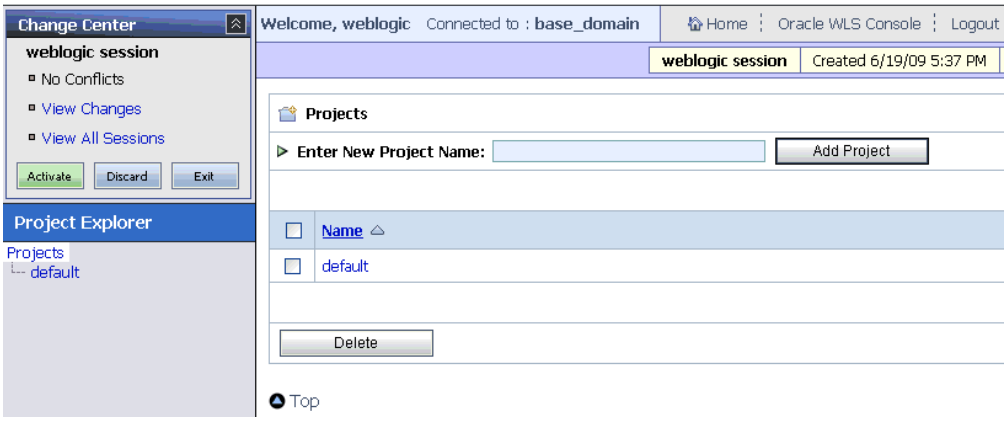


4. Click **Create** in the Change Center area to start a new Oracle Service Bus session.

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

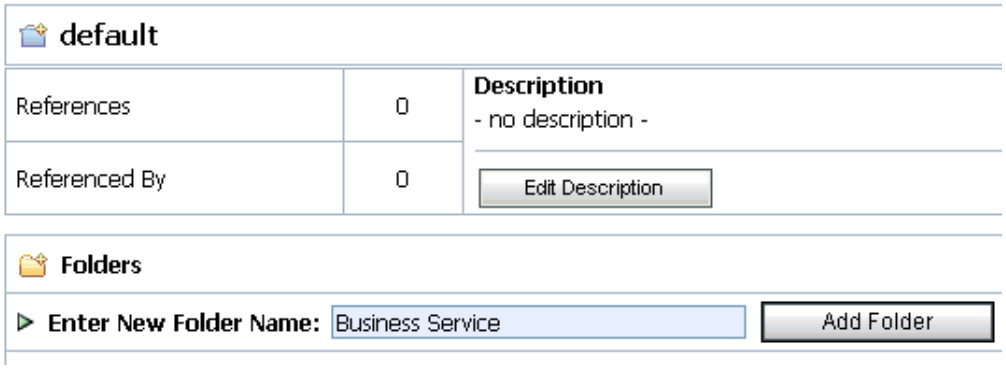


The Project Explorer page is displayed.



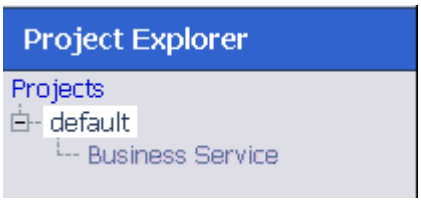
6. Click the **default** project node in the left pane.

The default project page is displayed.



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.

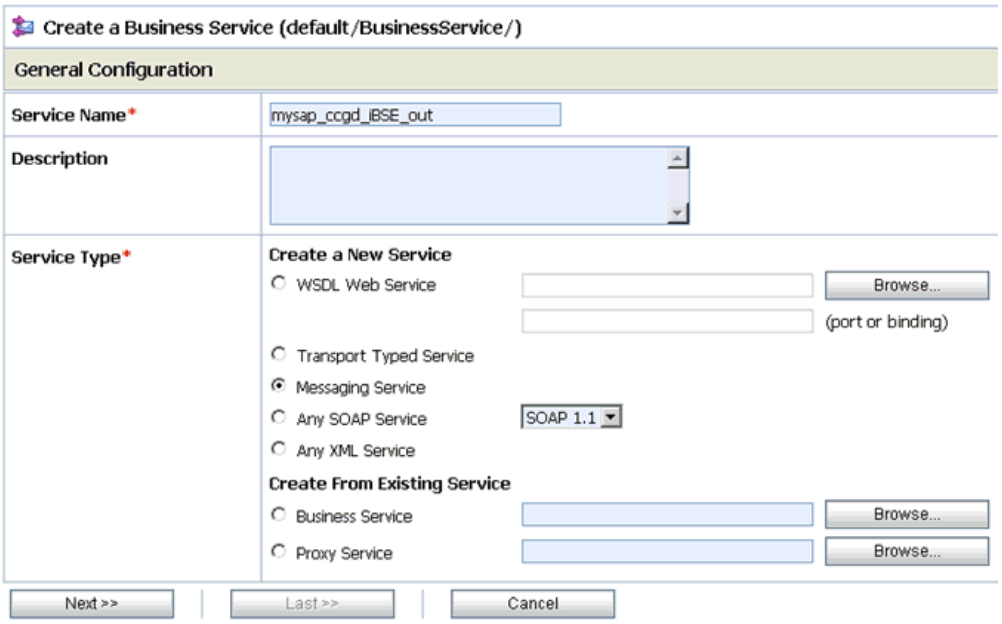
Configuring a File Type Business Service

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

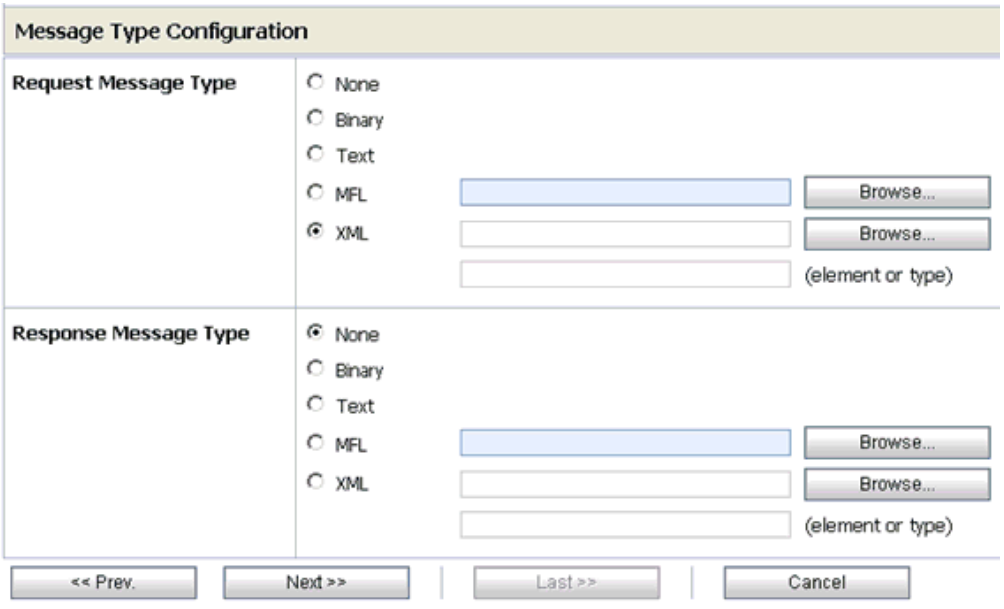
1. Select the **Business Service** folder you created in the left pane.

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

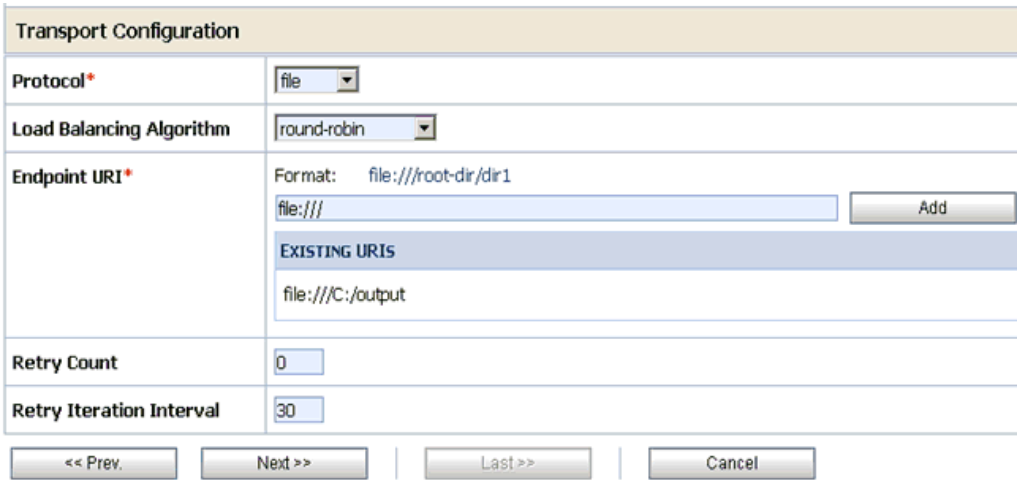
   The Create a Business Service - General Configuration page is displayed.



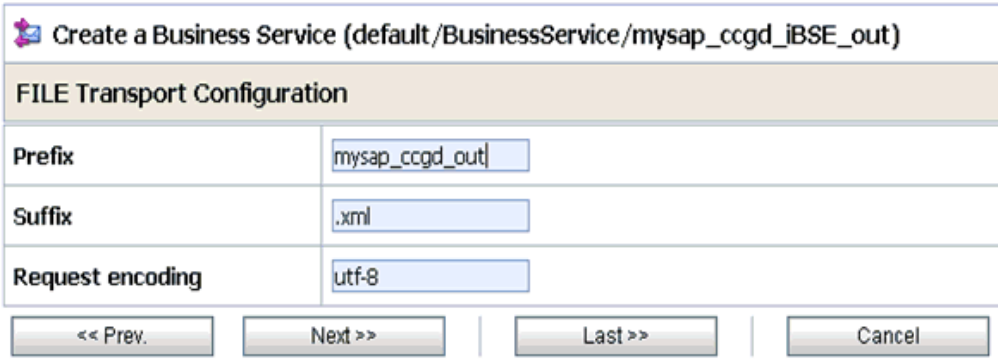
3. Provide a name for the Business Service and from the Service Type area select **Messaging Service**.
4. Click **Next**.



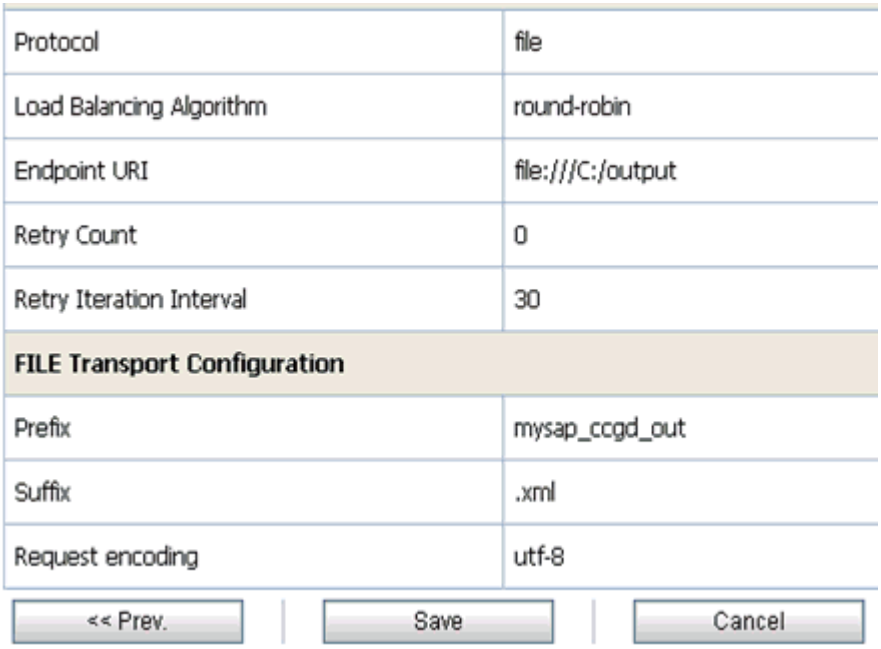
5. Select **XML** as the Request Message Type and **None** as the Response Message Type.
6. Click **Next**.



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**.



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



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

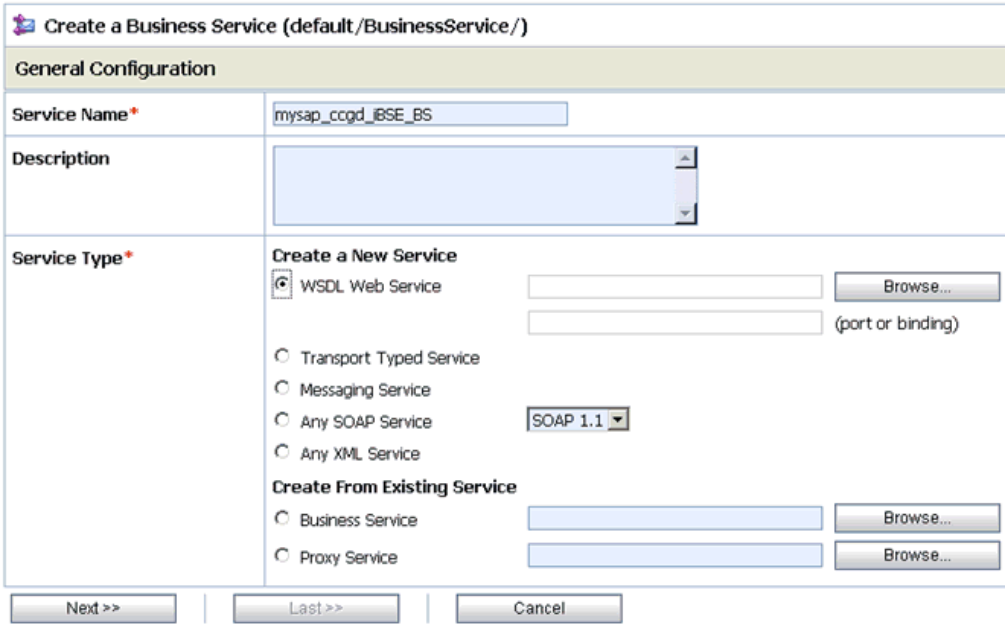
Configuring a WSDL Type Business Service

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.

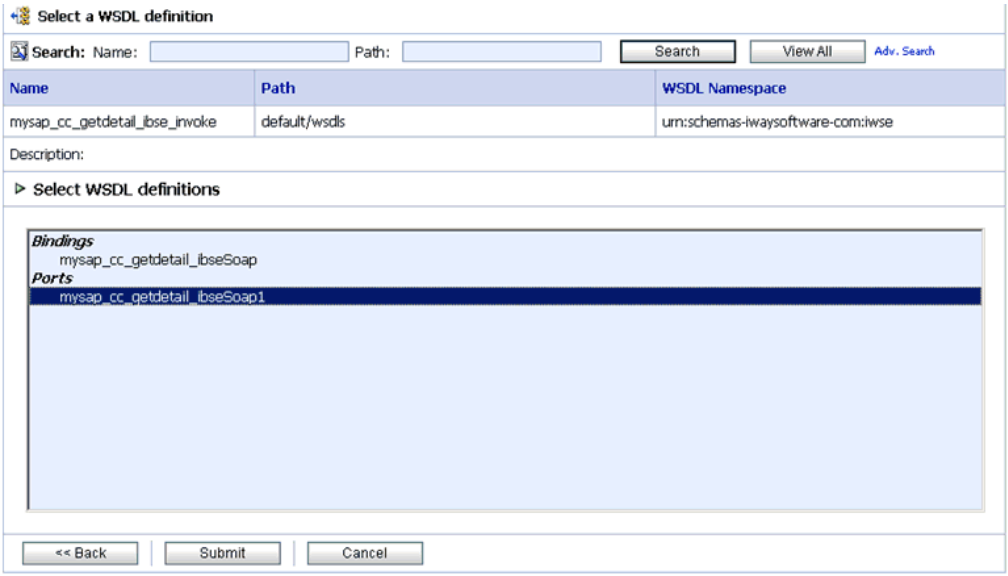
2. In the right pane, select **Business Service** from the Create Resource menu.

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



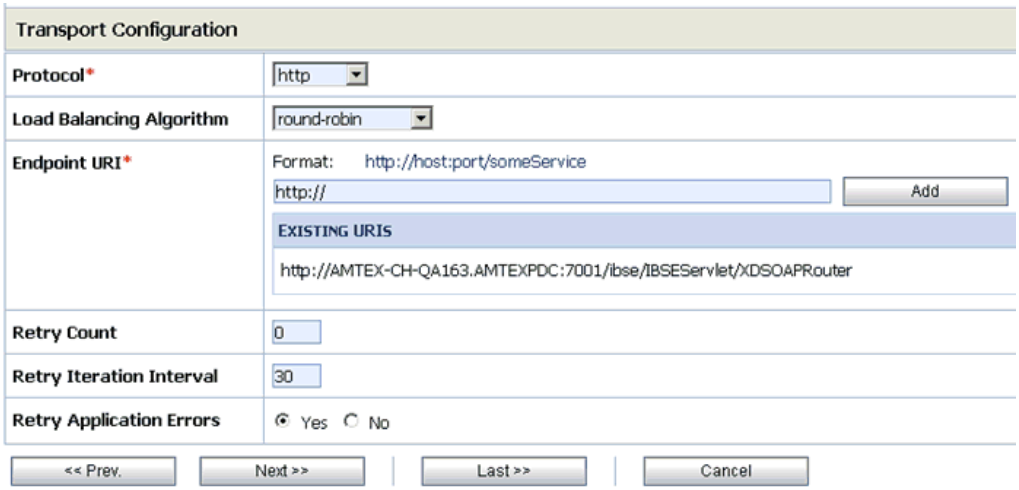
3. Provide a name for the Business Service and from the Service Type area select **WSDL Web Service**.
4. Click **Browse**.

   The Select a WSDL Definition dialog box is displayed.

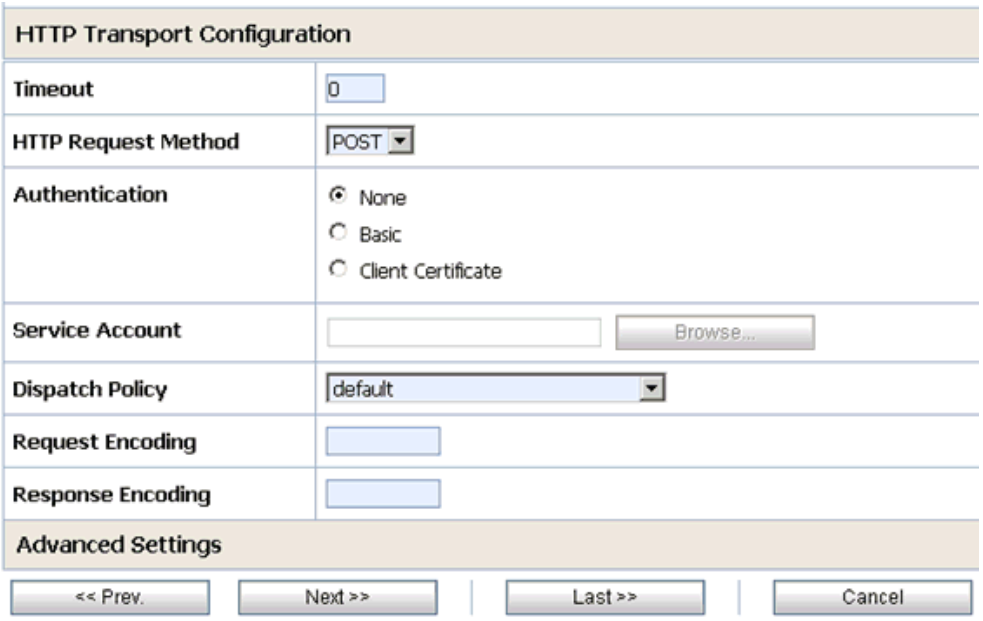


5. Select the required WSDL and then select the WSDL definition under the Ports section.
6. Click **Submit**.

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

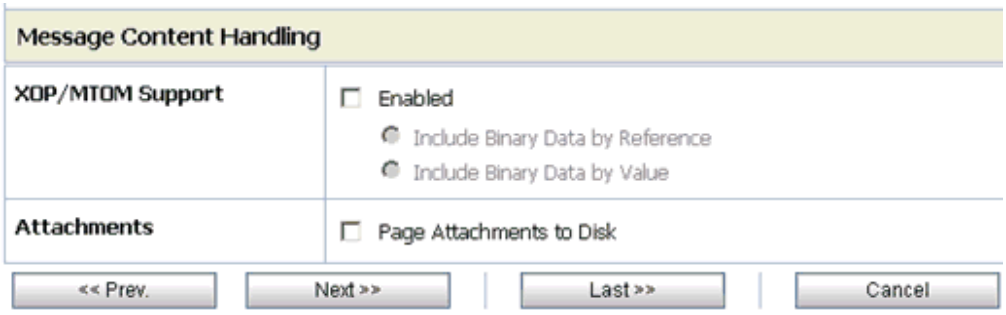


8. Select **http** from the Protocol drop-down list.
9. Enter the Endpoint URI in HTTP format and click **Add**.
10. Click **Next**.

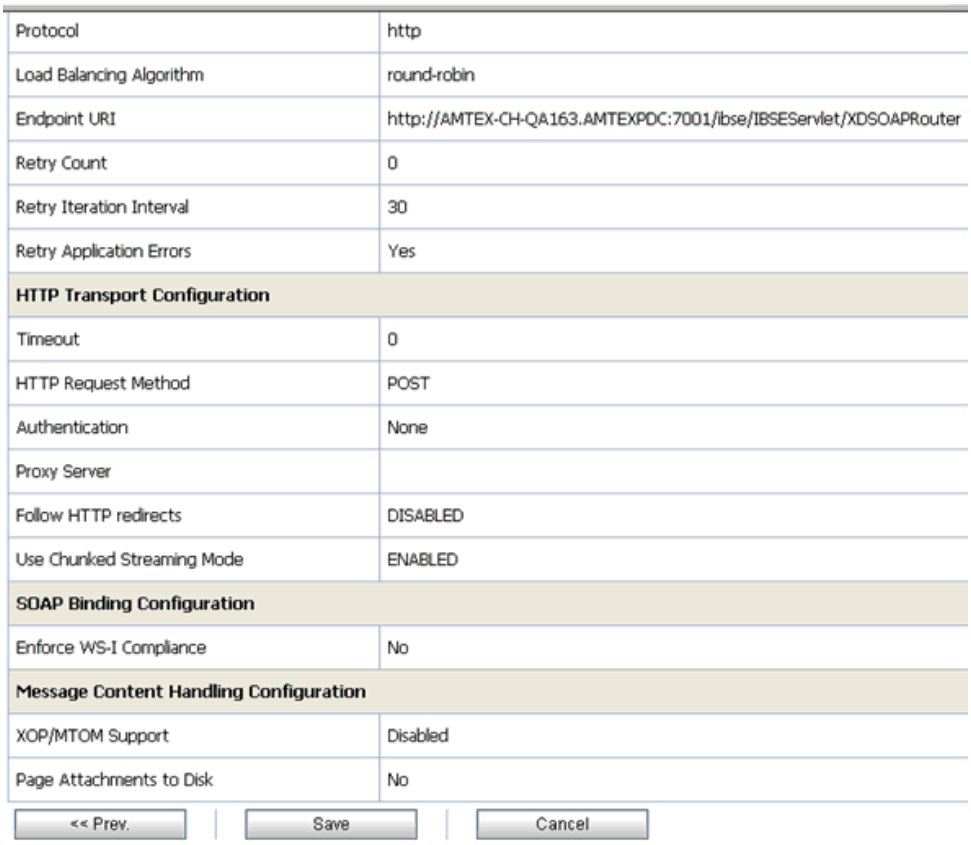


11. Click **Next**.

12. Click **Next**.



13. Click **Next**.

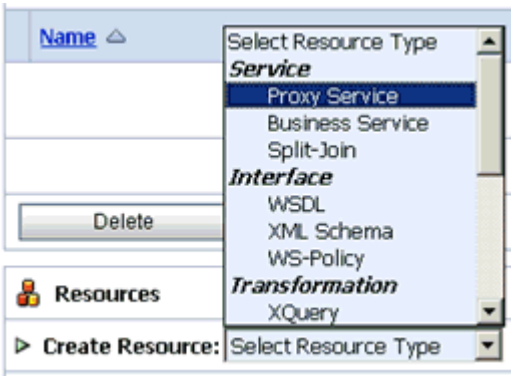


14. Review all the information for your Business Service and click **Save**.

Configuring a Proxy Service

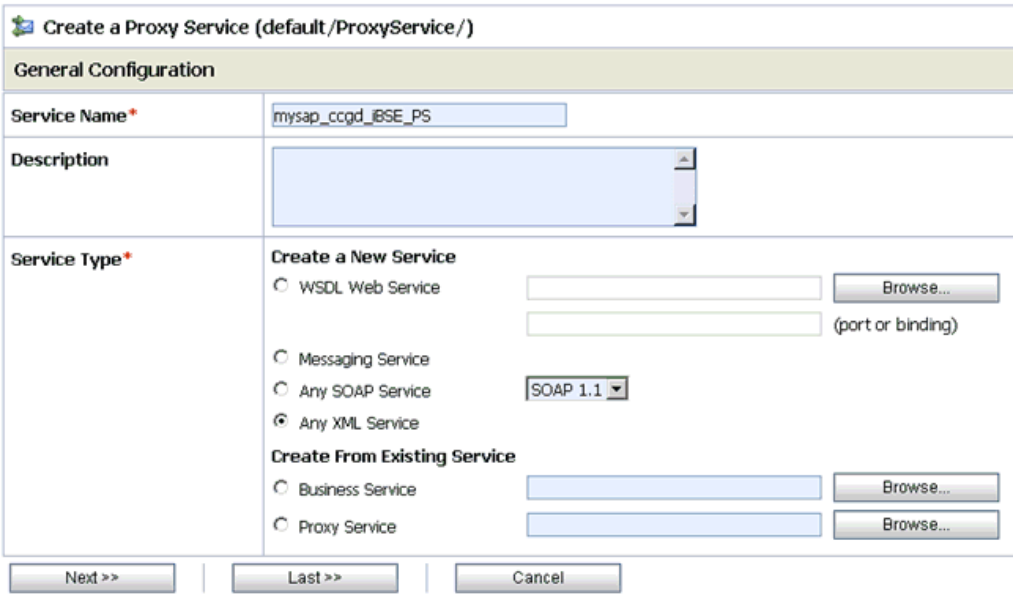
This section describes how to configure a Proxy Service using the Oracle Service Bus Console.

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

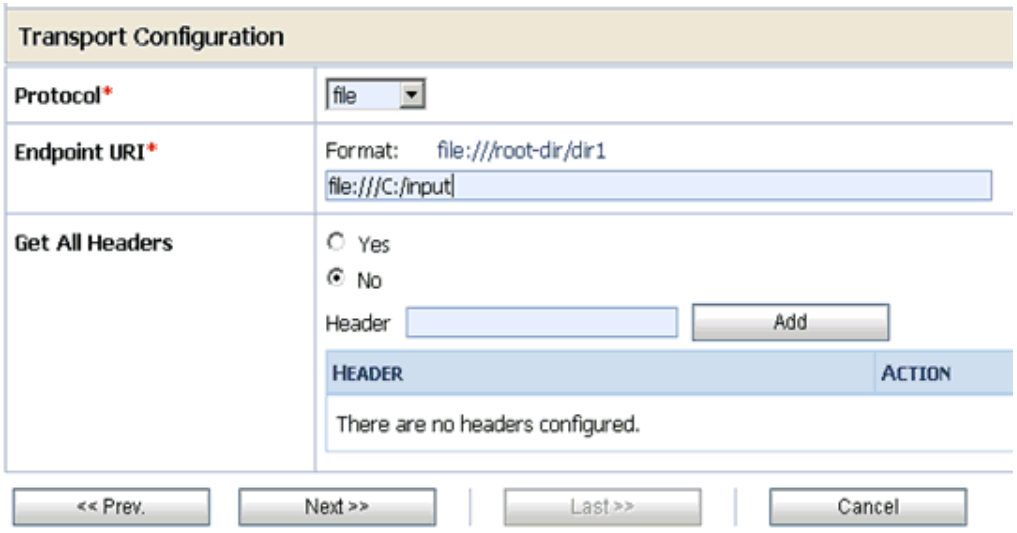


2. In the right pane, select **Proxy Service** from the Create Resource menu.

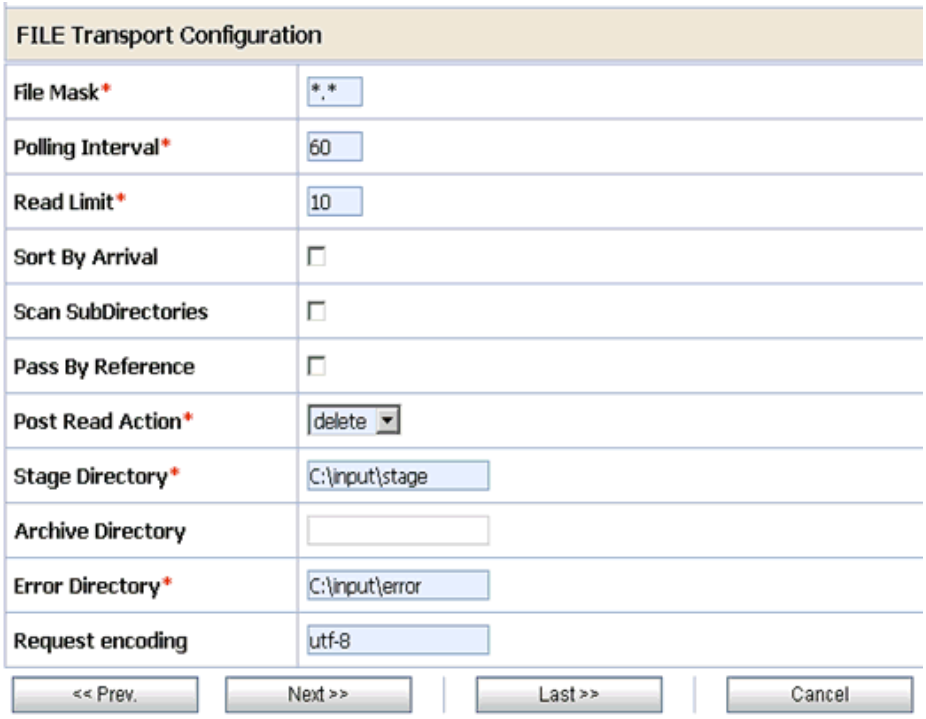
   The Create a Proxy Service - General Configuration page is displayed.



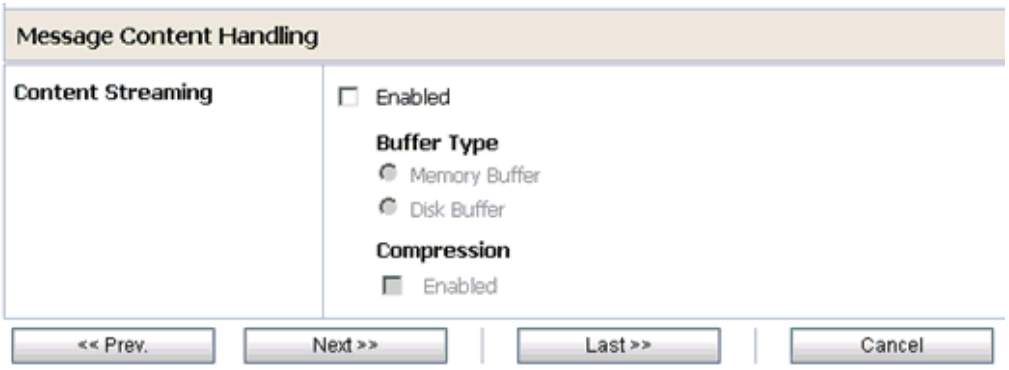
3. Provide a name for the Proxy Service and from the Service Type area select **Any XML Service**.
4. Click **Next**.



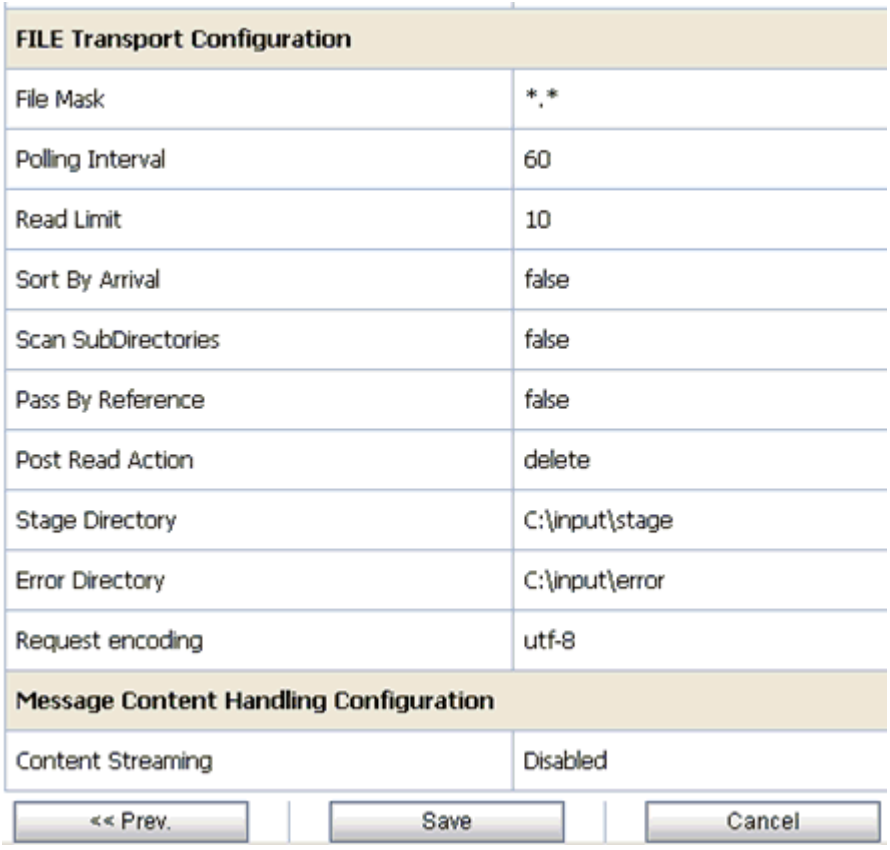
5. Select **file** from the Protocol drop-down list.
6. Enter the path to an input folder on your file system in the Endpoint URI field.
7. Click **Next**.



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



10. Click **Next**.

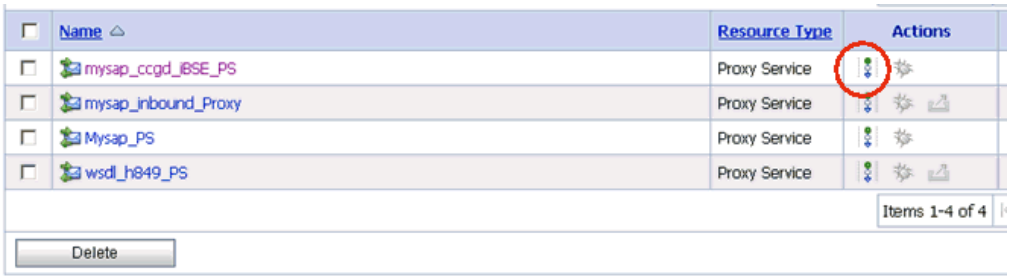


11. Review all the information for your Proxy Service and click **Save**.

Configuring a Pipeline

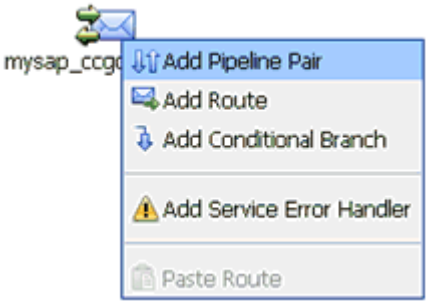
This section describes how to configure a Pipeline using the Oracle Service Bus Console.

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

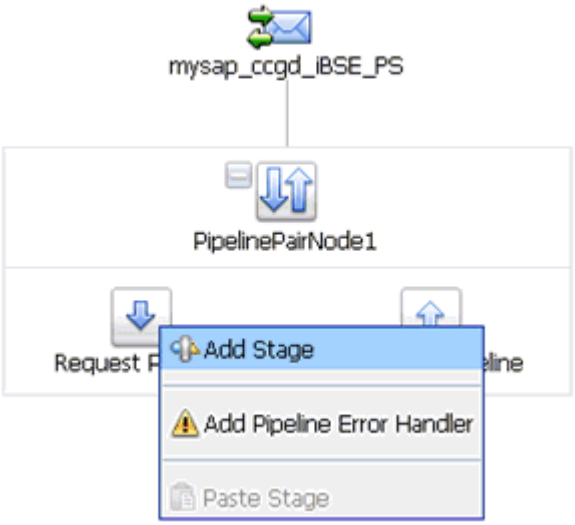


The Edit Message Flow workspace area is displayed.

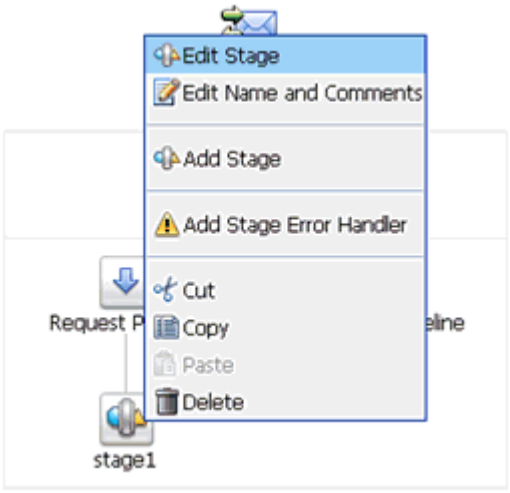
2. Click the **Proxy Service** icon and select **Add Pipeline Pair** from the context menu, as shown in the following image.



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

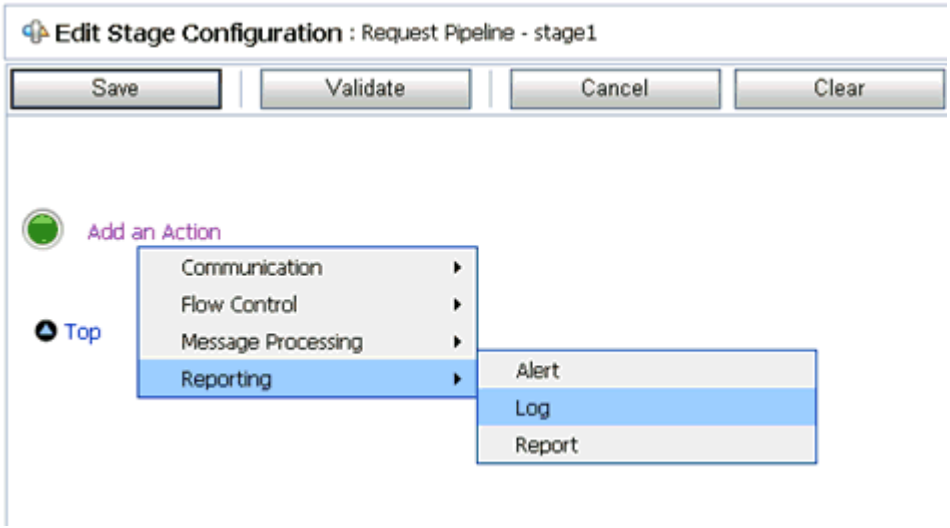


The Stage1 icon is added below the Request Pipeline icon.

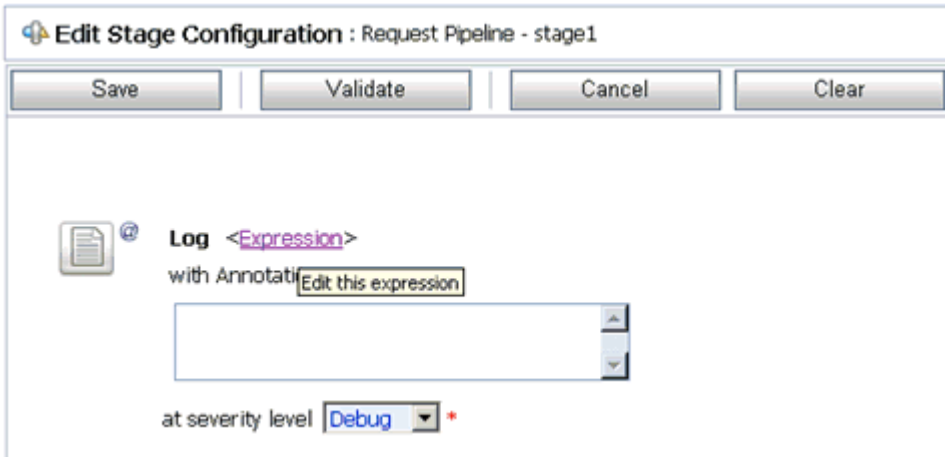


4. Click the **Stage1** icon and select **Edit Stage** from the context menu.

   The Edit Stage Configuration workspace area is displayed.

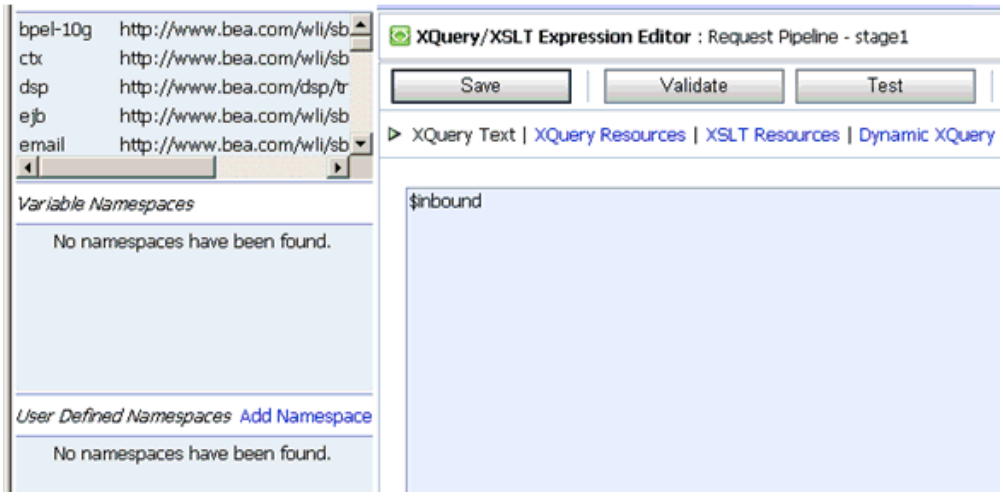


5. Click **Add an Action**, select **Reporting** from the context menu, and click **Log**.



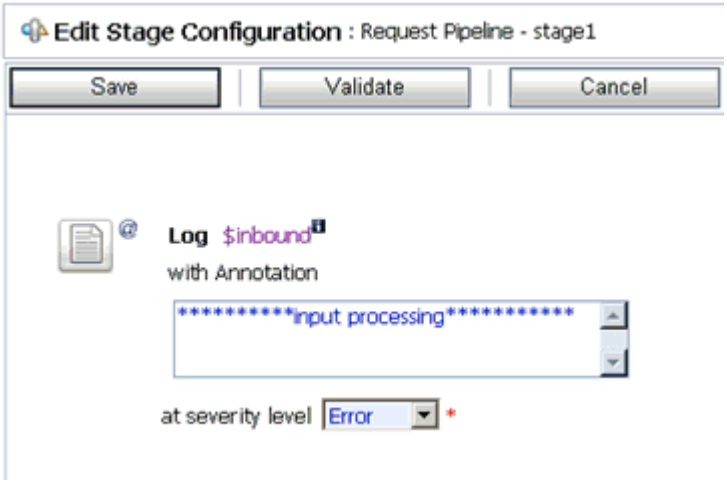
6. Click **<Expression>** to edit the expression.

   The XQuery/XSLT Expression Editor is displayed.

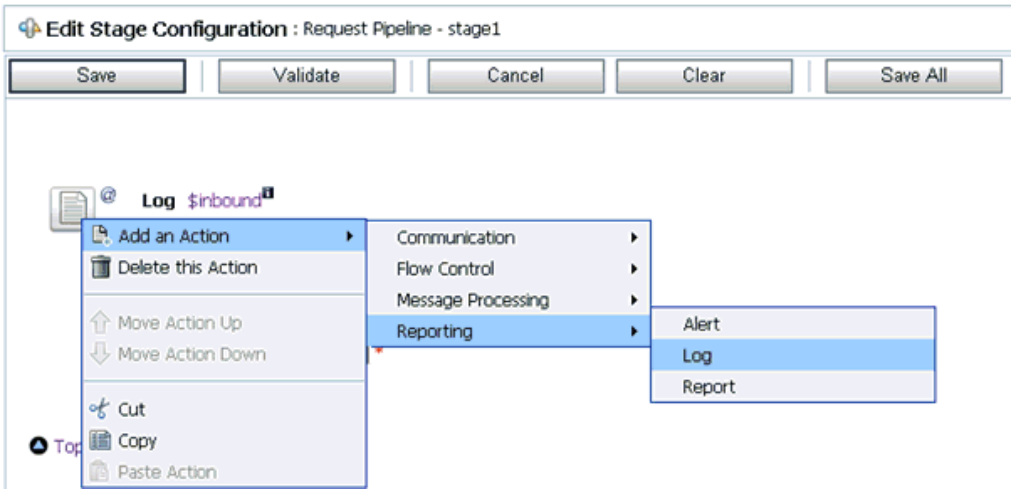


7. In the XQuery Text area, type **$inbound**.
8. Click **Validate** and then **Save**.

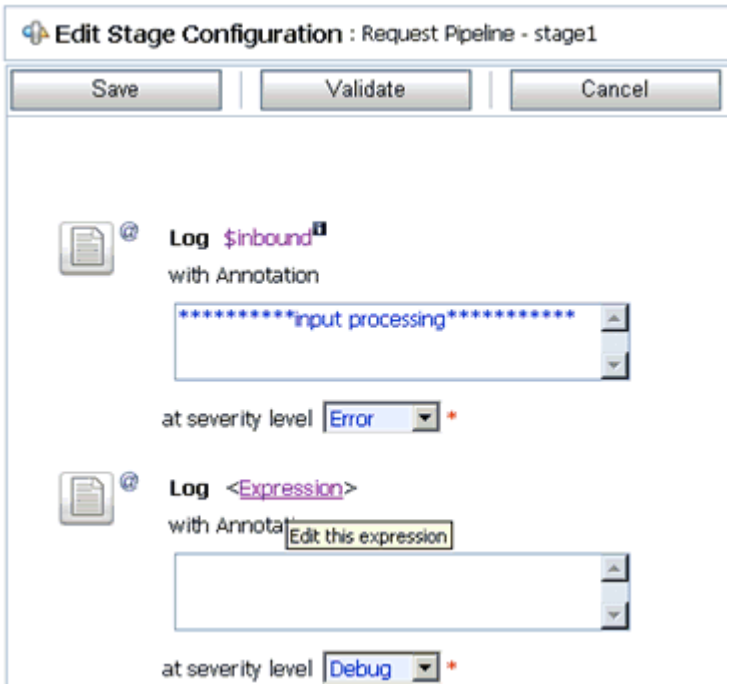
   You are returned to the Edit Stage Configuration workspace area.



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.

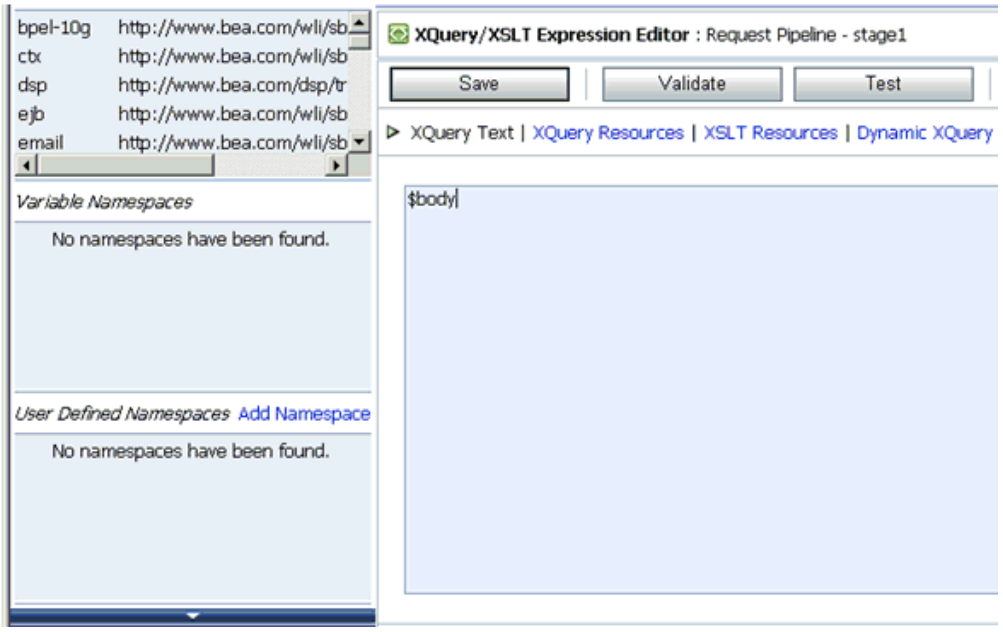


A new Log configuration is added, as shown in the following image.



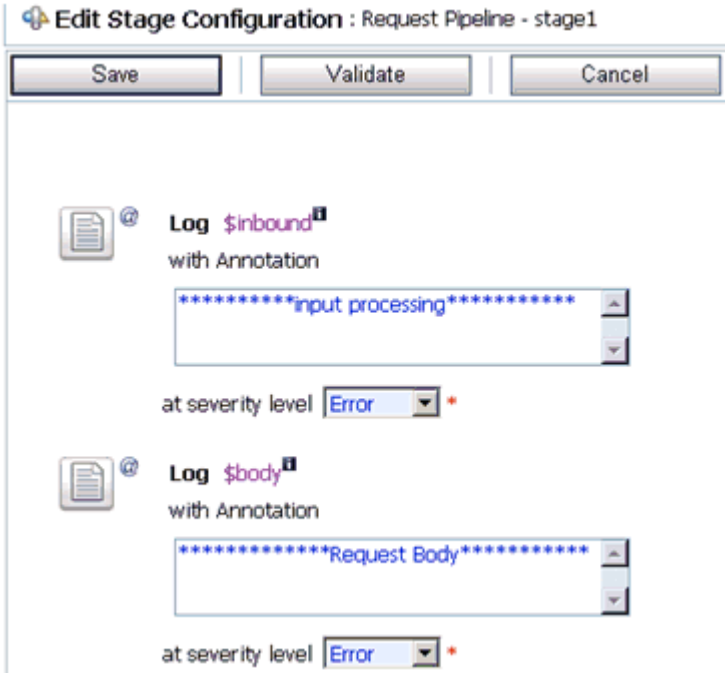
12. Click **<Expression>** to edit the expression.

    The XQuery/XSLT Expression Editor is displayed.

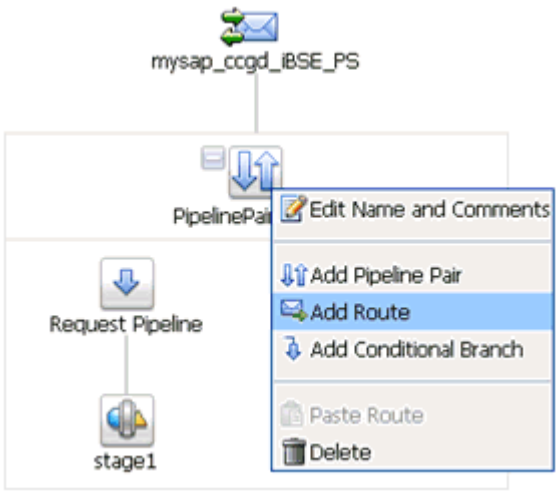


13. In the XQuery Text area, type **$body**.
14. Click **Validate** and then **Save**.

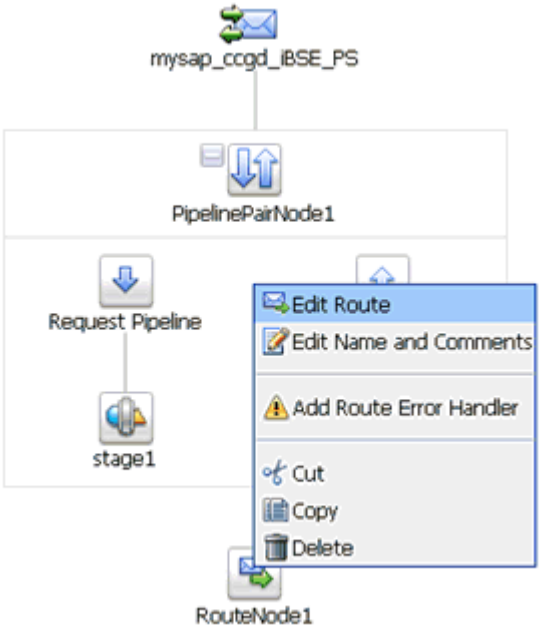
    You are returned to the Edit Stage Configuration workspace area.



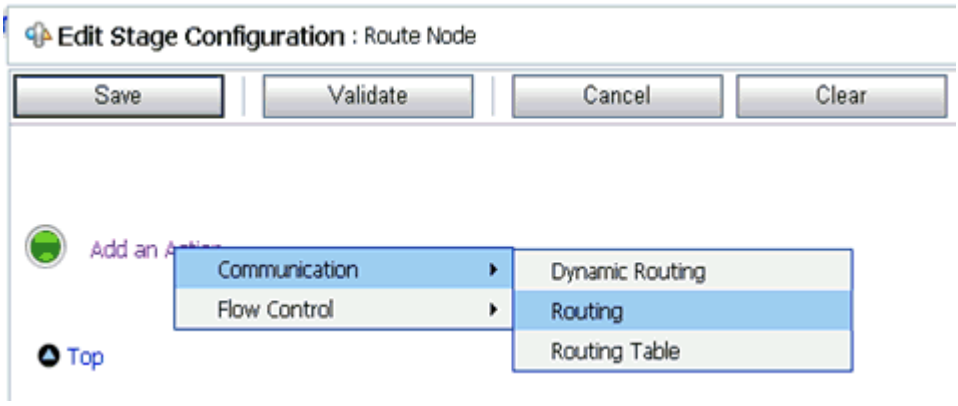
15. Type any annotation/comments in the text box (for example, **********Request Body**********).
16. Select **Error** from the severity level drop-down list.
17. Click **Validate** and then **Save**.



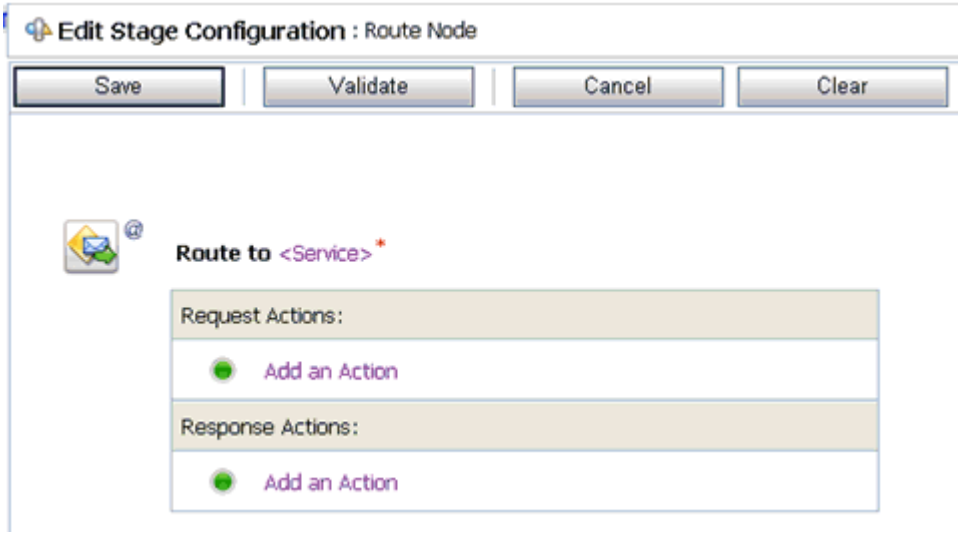
18. Click the **PipelinePairNode1** icon and select **Add Route** from the context menu.



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

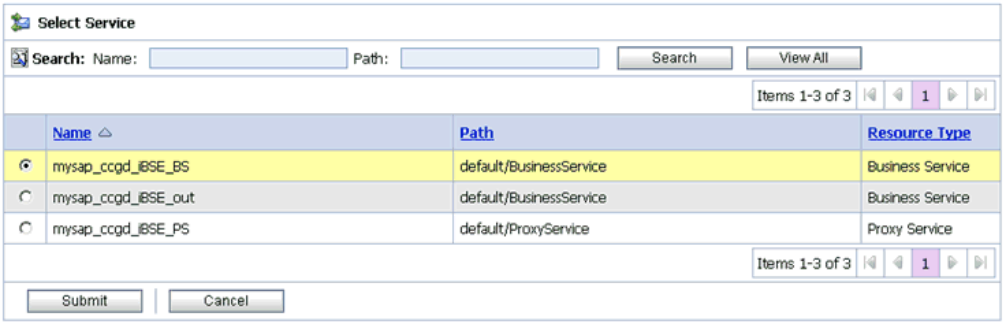


20. Click **Add an Action**, select **Communication** from the context menu, and click **Routing**.



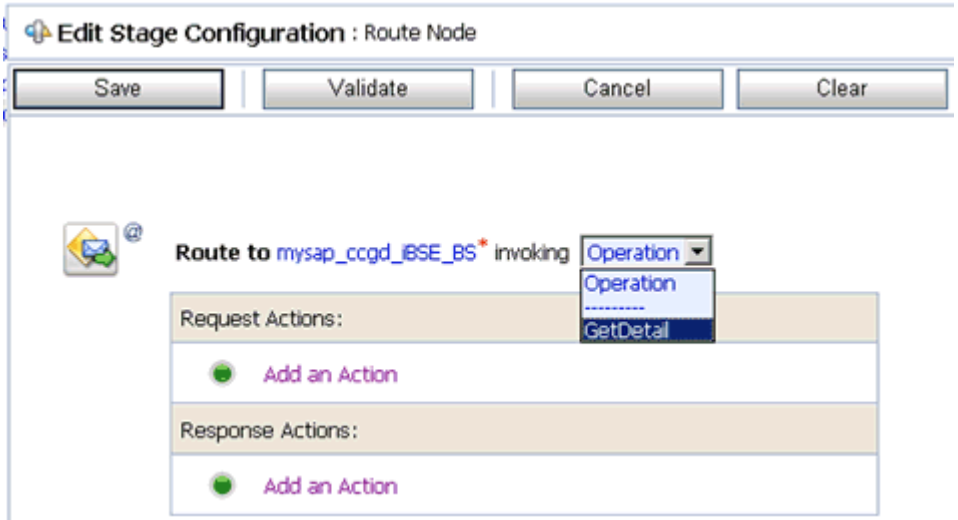
21. Click **<Service>**.

    The Select Service dialog box is displayed.

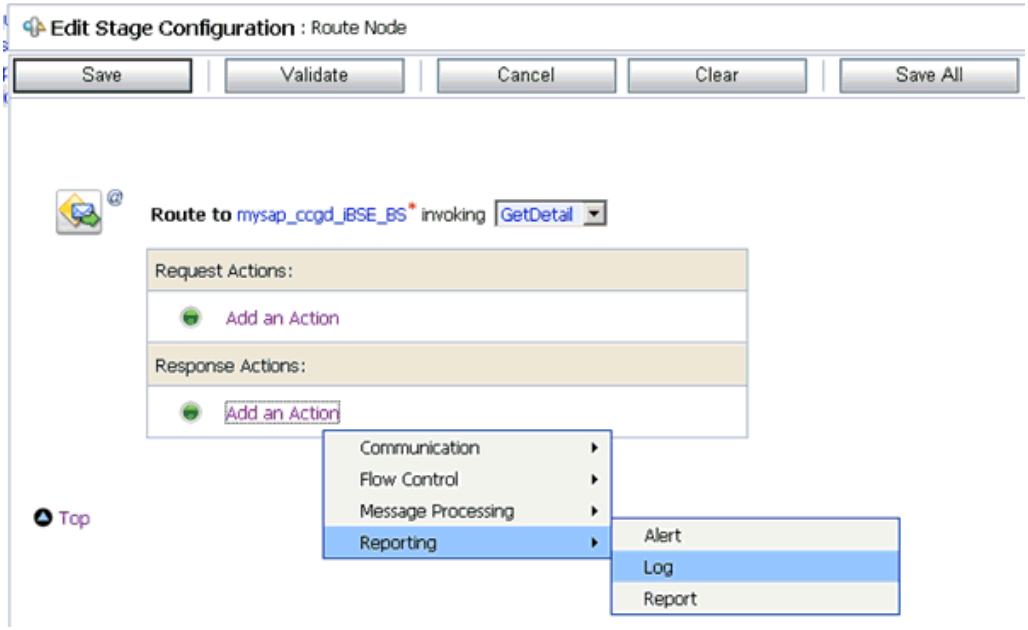


22. Select a WSDL type Business Service and click **Submit**.

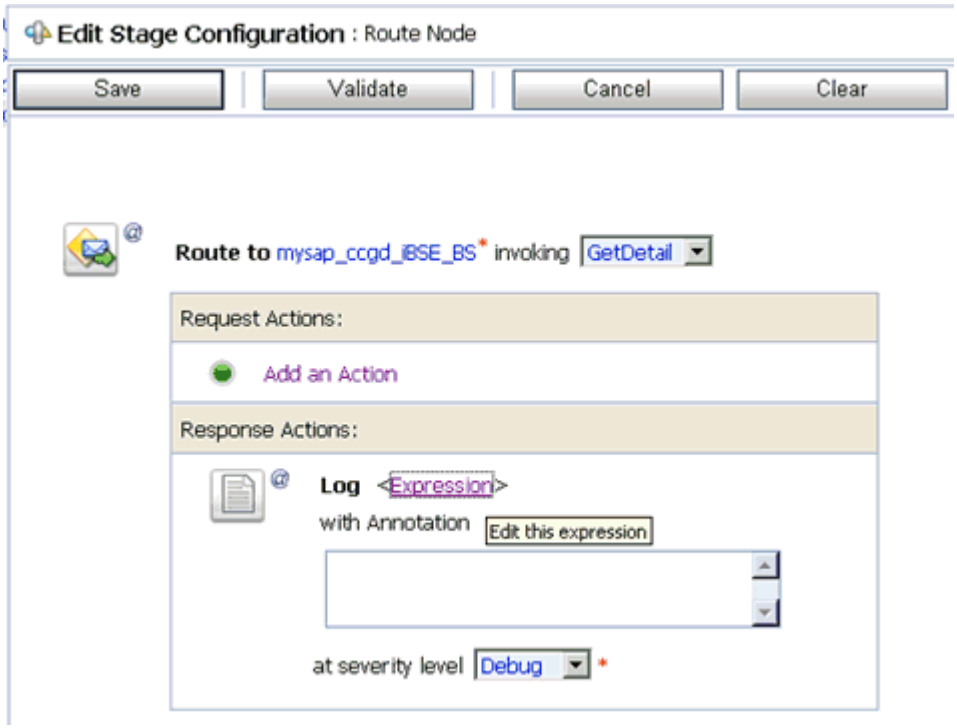
    You are returned to the Edit Stage Configuration workspace area.



23. Select **GetDetail** 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**.

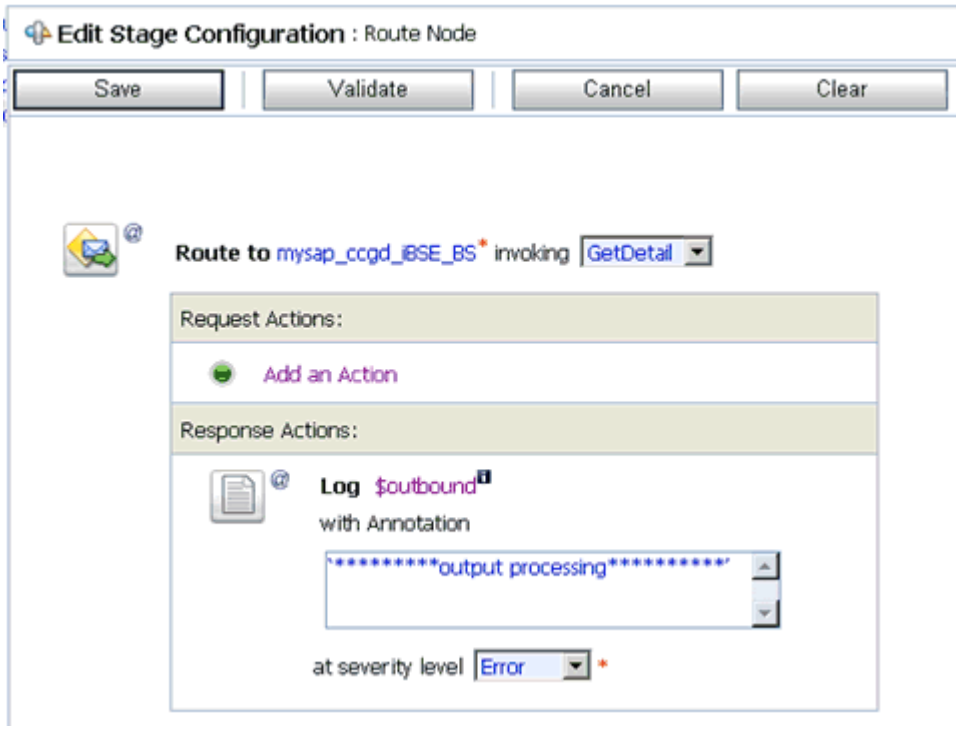


26. Click **<Expression>** to edit the expression.

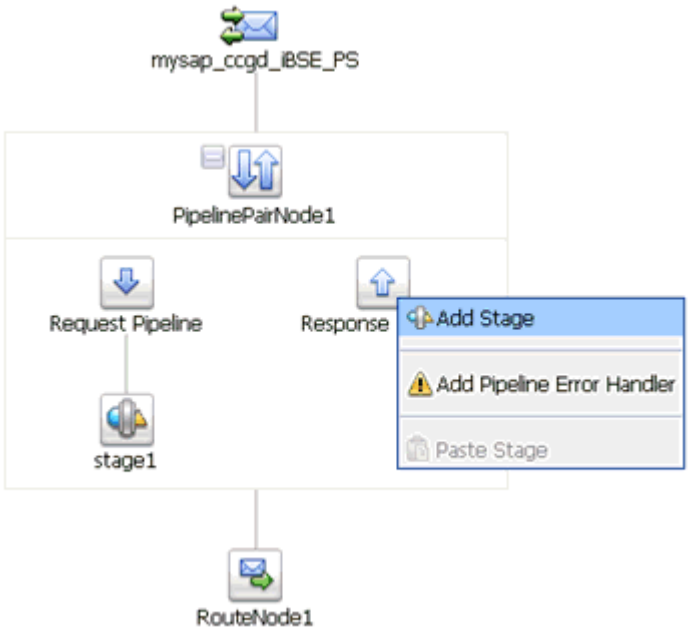
    The XQuery/XSLT Expression Editor is displayed.

27. In the XQuery Text area, type **$outbound**.
28. Click **Validate** and then **Save**.

    You are returned to the Edit Stage Configuration workspace area.

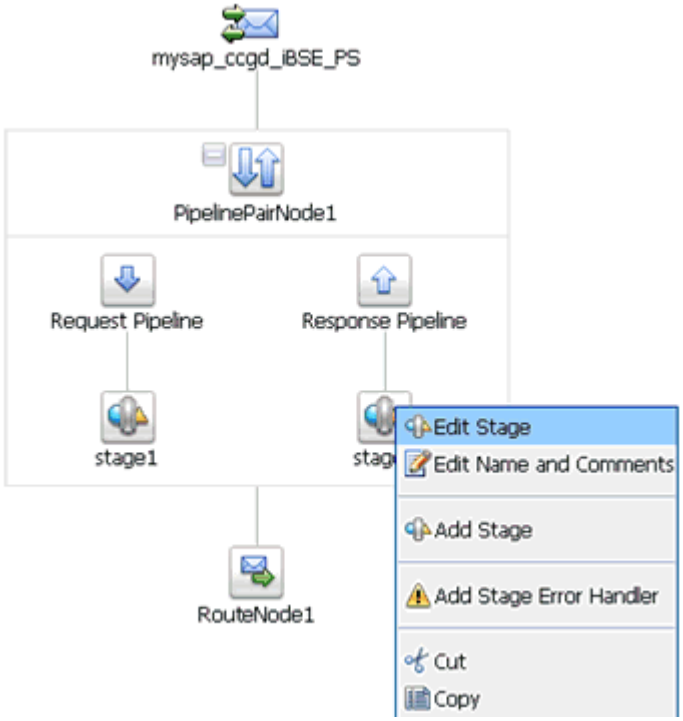


29. Type any annotation/comments in the text box (for example, **********output processing**********).
30. Select **Error** from the severity level drop-down list.
31. Click **Validate** and then **Save**.



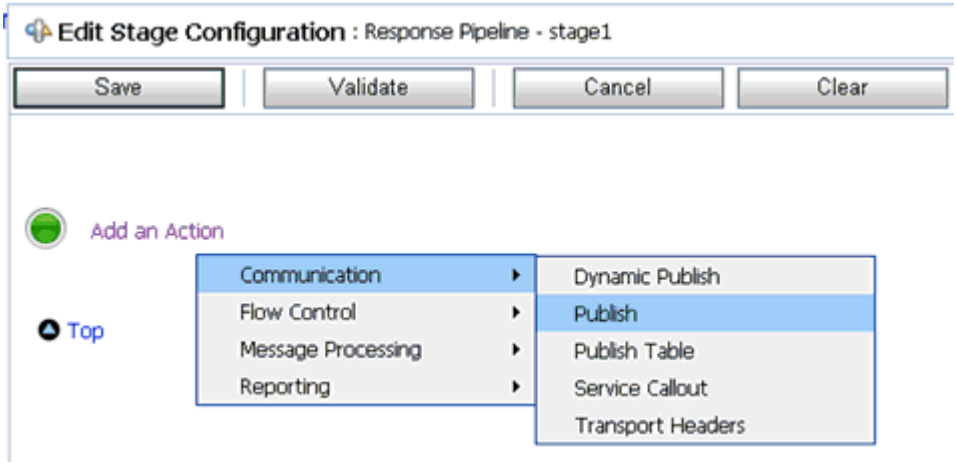
32. Click the **Response Pipeline** icon and select **Add Stage** from the context menu.

    The Stage1 icon is added below the Response Pipeline icon.

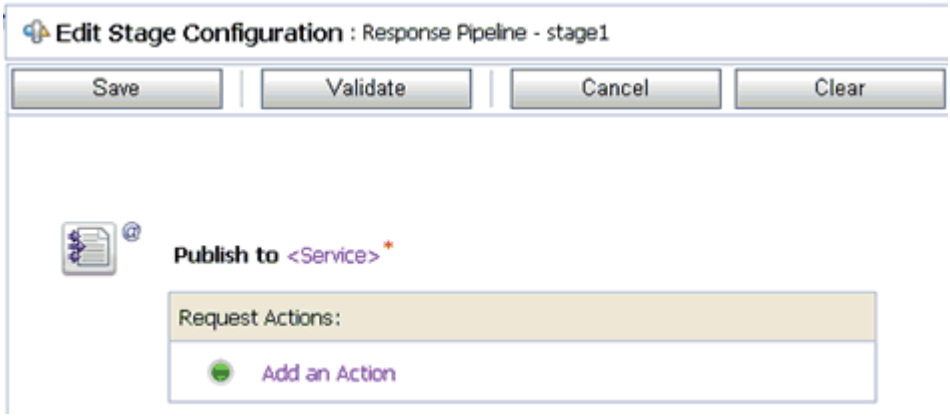


33. Click the **Stage1** icon and select **Edit Stage** from the context menu.

    The Edit Stage Configuration workspace area is displayed.

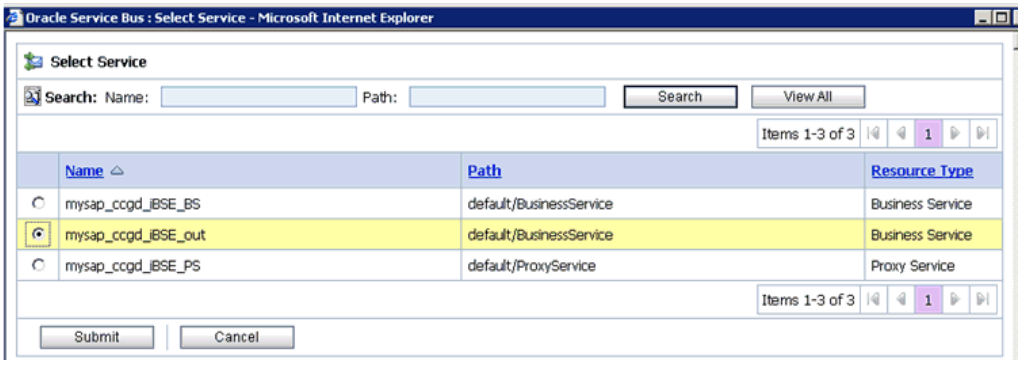


34. Click **Add an Action**, select **Communication** from the context menu, and click **Publish**.



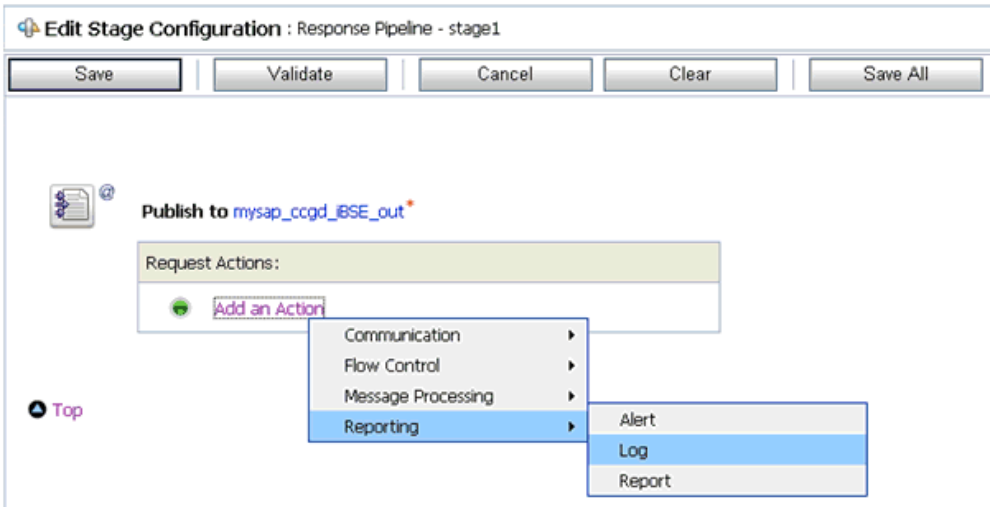
35. Click **<Service>**.

    The Select Service dialog box is displayed.



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

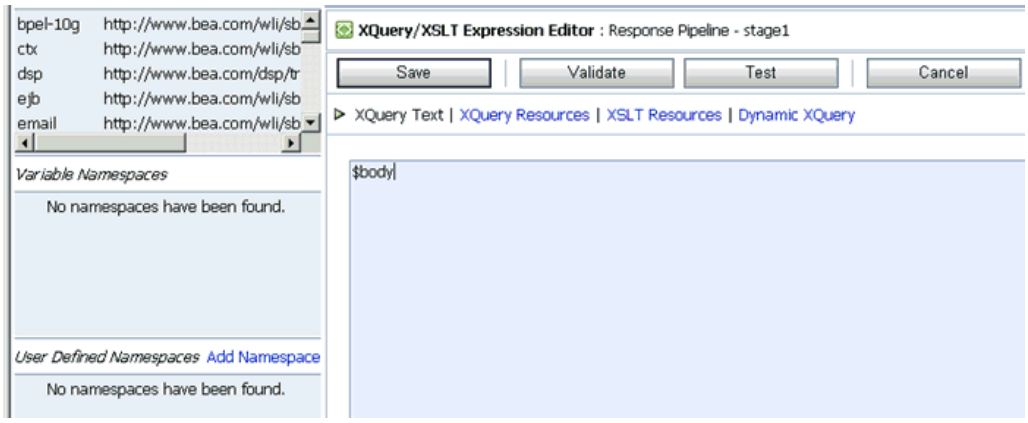
    You are returned to the Edit Stage Configuration workspace area.



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

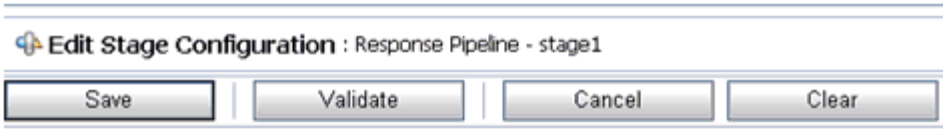
38. Click **<Expression>** to edit the expression.

    The XQuery/XSLT Expression Editor is displayed.

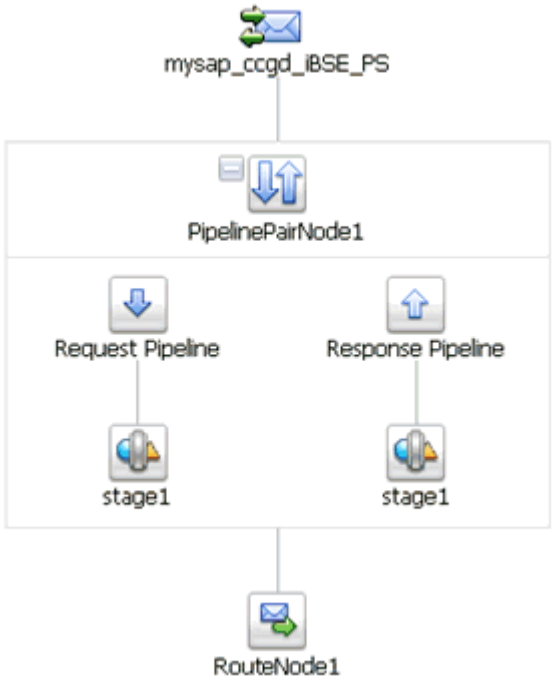


39. In the XQuery Text area, type **$body**.
40. Click **Validate** and then **Save**.

    You are returned to the Edit Stage Configuration workspace area.



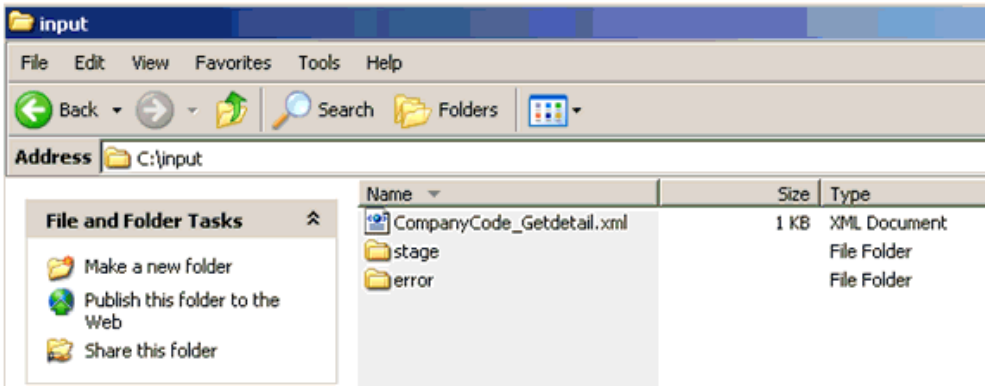
41. Type any annotation/comments in the text box (for example, **********Response Body**********).
42. Select **Error** from the severity level drop-down list.
43. Click **Validate** and then **Save**.



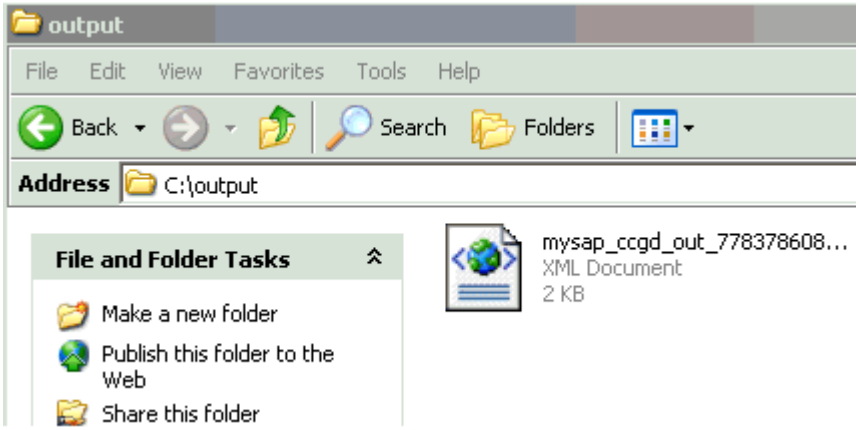
44. Click **Save**.

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.



An output XML file is received in the destination folder.


output
File Edit View Favorites Tools Help
Back
Search
Folders
Address
C:\output
File and Folder Tasks
Make a new folder
Publish this folder to the Web
Share this folder
mysap_ccgd_out_778378608...
XML Document
2 KB

6

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

Oracle Application Adapter for SAP R/3 integrates seamlessly with Oracle Service Bus (OSB) to facilitate Web service integration. OSB is based on the Service-Oriented Architecture (SOA). It consumes adapter services exposed as Web Service Definition Language (WSDL) documents.

This chapter includes the following topics:

Overview of Application Adapter Integration with Oracle Service Bus

To integrate with Oracle Service Bus (OSB), Oracle Application Adapter for SAP R/3 must be deployed 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).

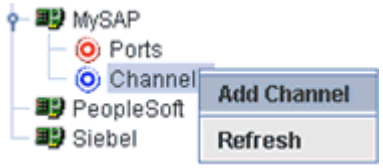
This section includes the following topics:

Creating a Channel and Publishing a WSDL From Application Explorer to Oracle Service Bus

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

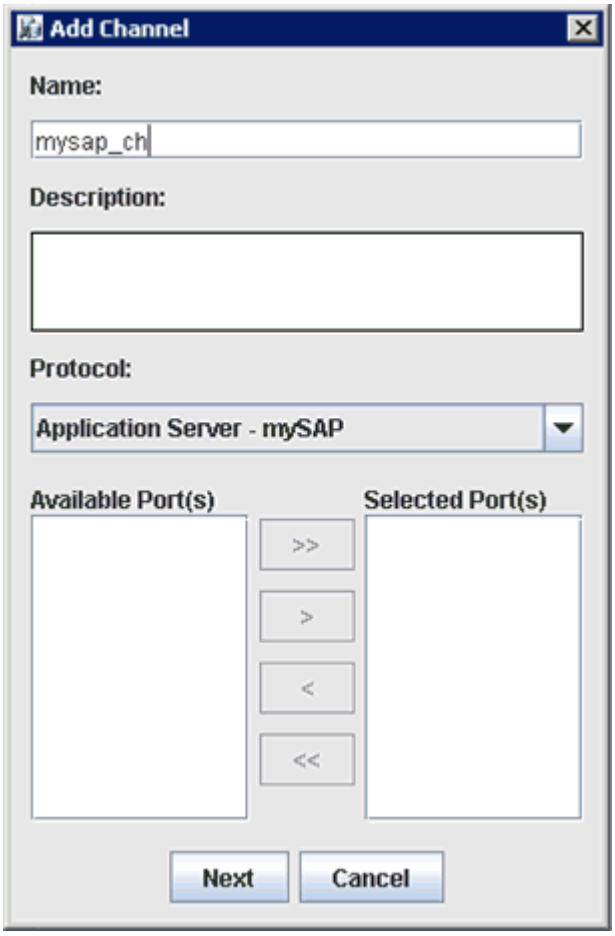
1. Start Application Explorer, connect to a J2CA configuration, and connect to a SAP R/3 target.

   For more information, see Chapter 3, "Configuring Oracle Application Adapter for SAP R/3".



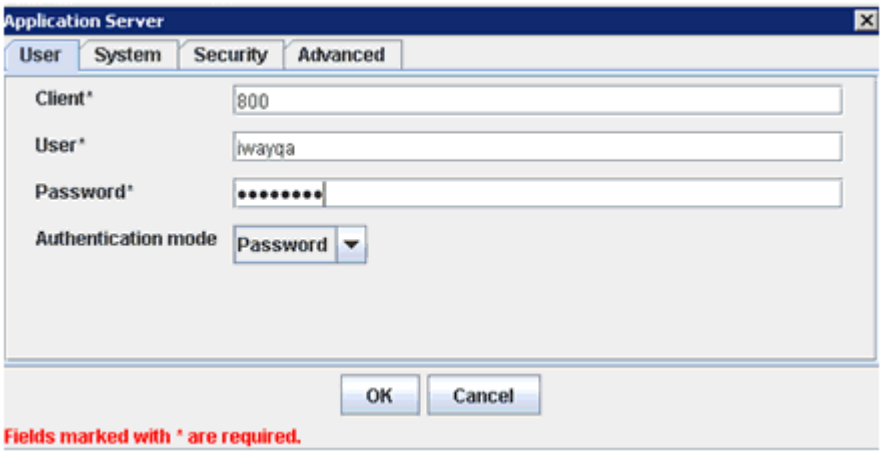
2. Right-click the **Channels** node for SAP R/3 and select **Add Channel** from the context menu.

   The Add Channel dialog box is displayed.

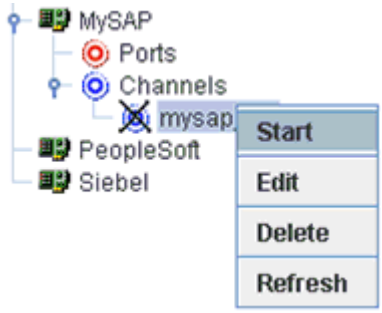


3. Enter a name for the new SAP R/3 channel, select **Application Server - mySAP** from the Protocol drop-down list, and click **Next**.

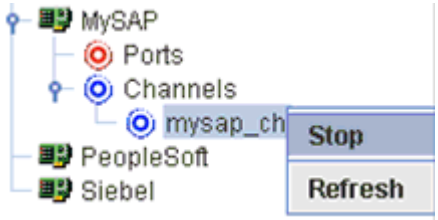
   The Application Server dialog box is displayed.



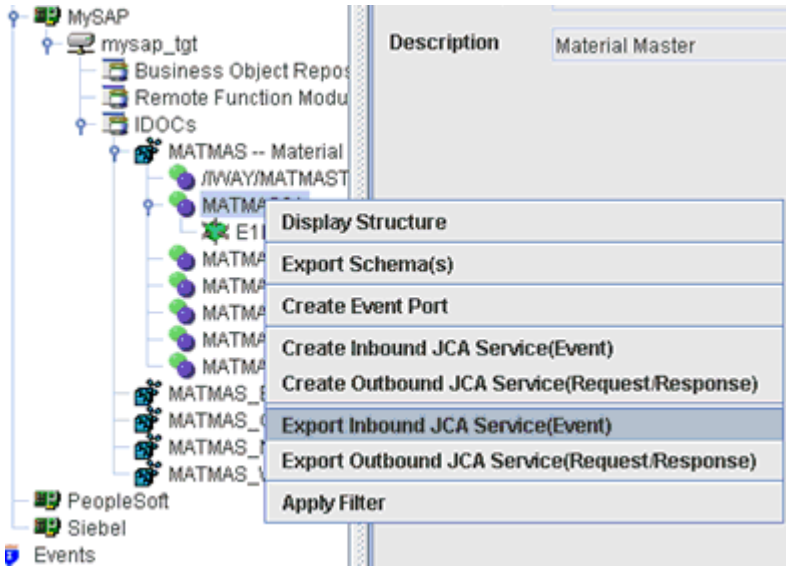
4. Provide the required connection information for your SAP R/3 system.
5. Click **OK**.



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

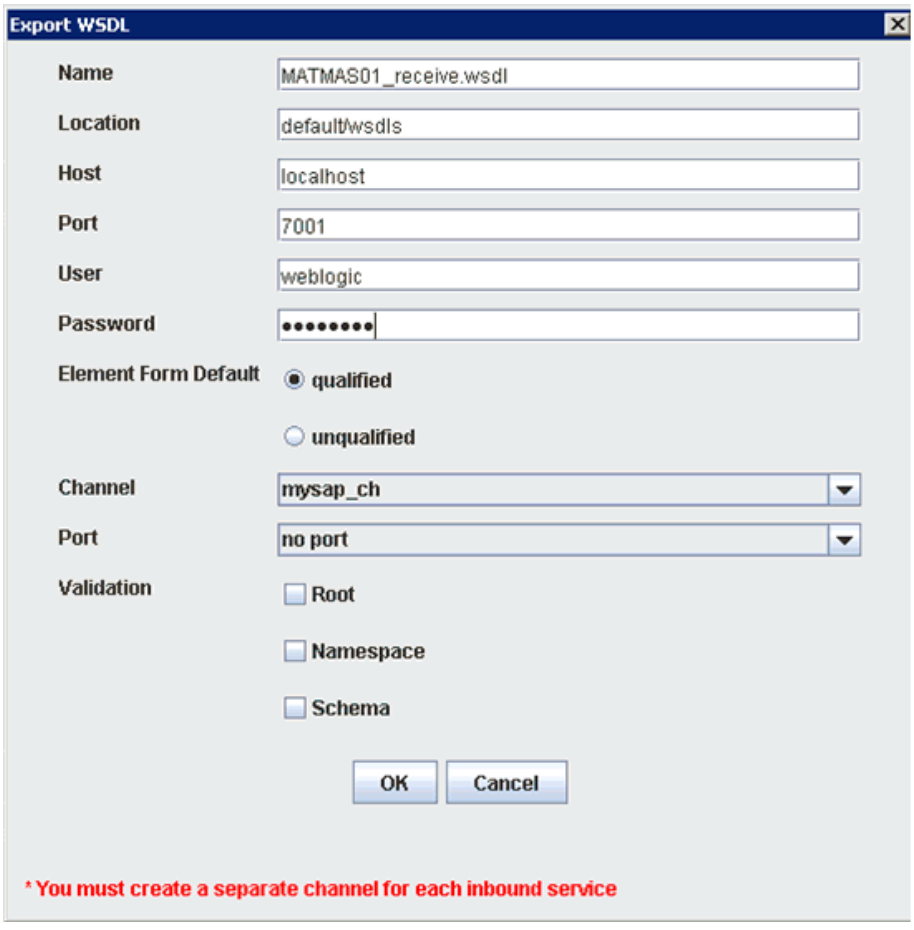


7. Right-click the started channel and select **Stop** from the context menu.



8. Expand the **IDOCs** node, **MATMAS -- Material**, right-click the **MATMAS01** IDoc, and select **Export Inbound JCA Service(Event)** from the context menu.

   The Export WSDL dialog box is displayed.



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

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

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

11. In the Host field, enter the name of the machine where Oracle WebLogic Server is running.
12. In the Port field, enter the port for the domain you are using.
13. In the User field, enter your username to access Oracle Service Bus.
14. In the Password field, enter your password to access Oracle Service Bus.
15. Ensure that **qualified** is selected as the element form, which is the default.
16. Select an available channel from the Channel drop-down list.
17. Select **no port** from the Port drop-down list.
18. Select the validation type (if required) by selecting either Root, Namespace, or Schema.
19. 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. If it is already running, restart the server.
2. Open the Oracle Service Bus Console in a Web browser by entering the following URL:

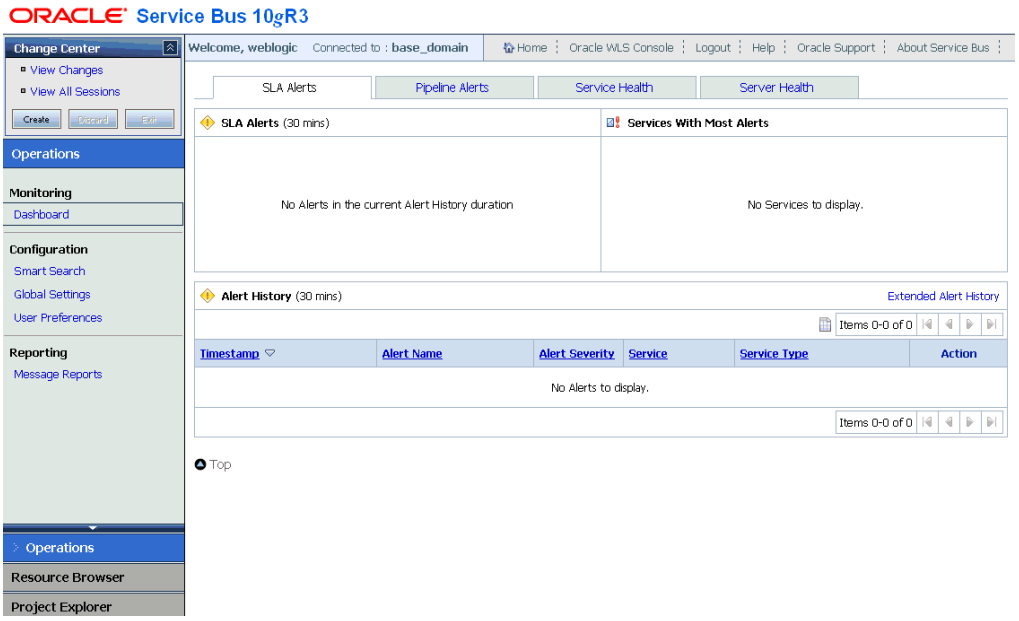
   `http://hostname:port/sbconsole`

   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.

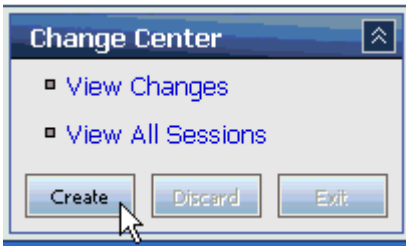
   The Oracle Service Bus Console logon page is displayed.

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

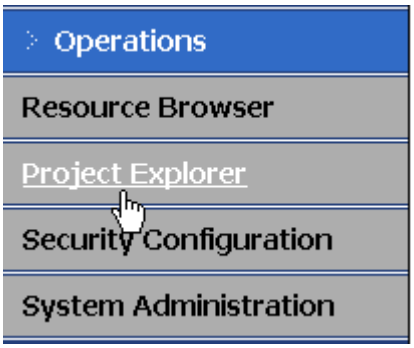
   The Oracle Service Bus Console home page is displayed.



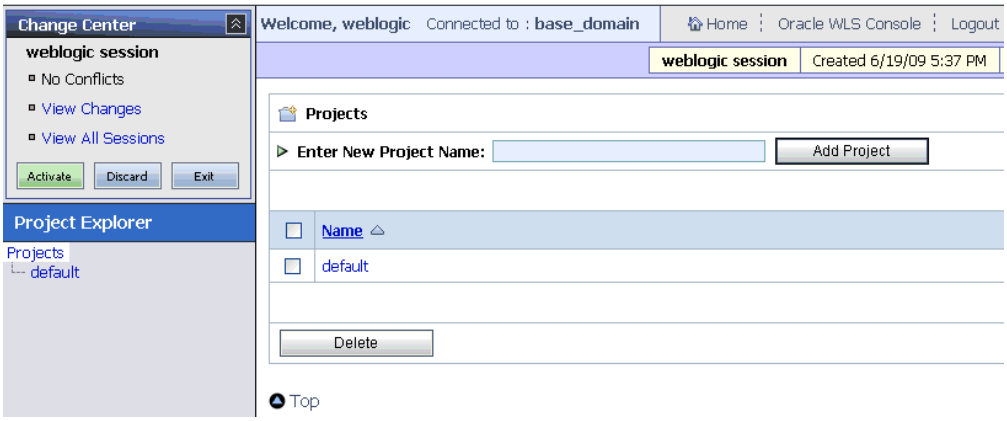
4. Click **Create** in the Change Center area to start a new Oracle Service Bus session.



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

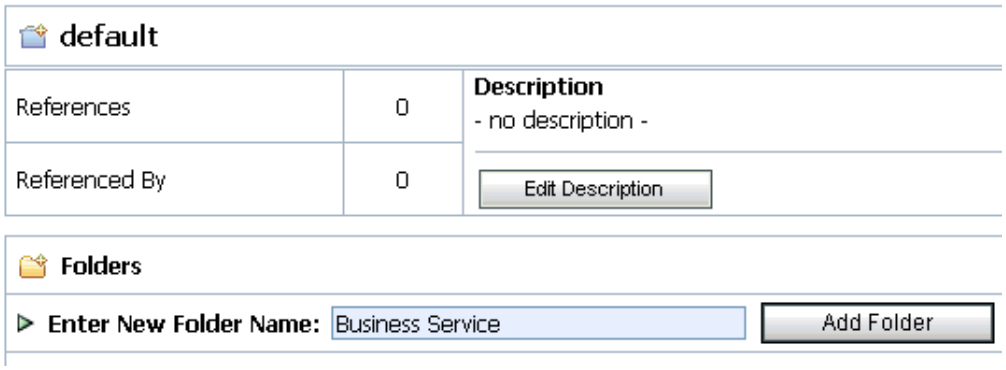


The Project Explorer page is displayed.



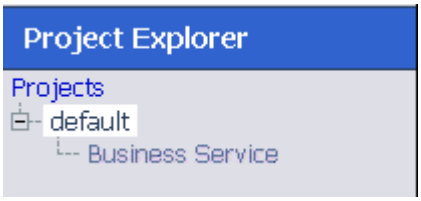
6. Click the **default** project node in the left pane.

   The default project page is displayed.



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.



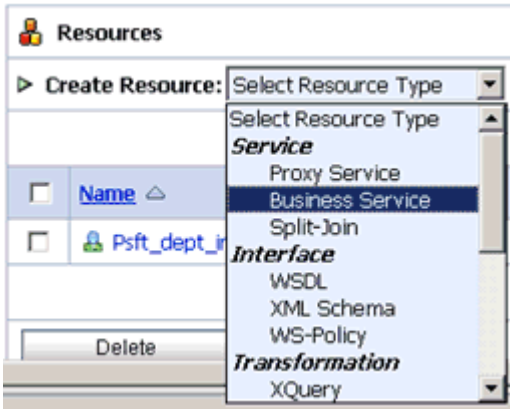
8. In the Enter New Folder Name field, type **Proxy Service** and click **Add Folder**.

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

Configuring a File Type Business Service

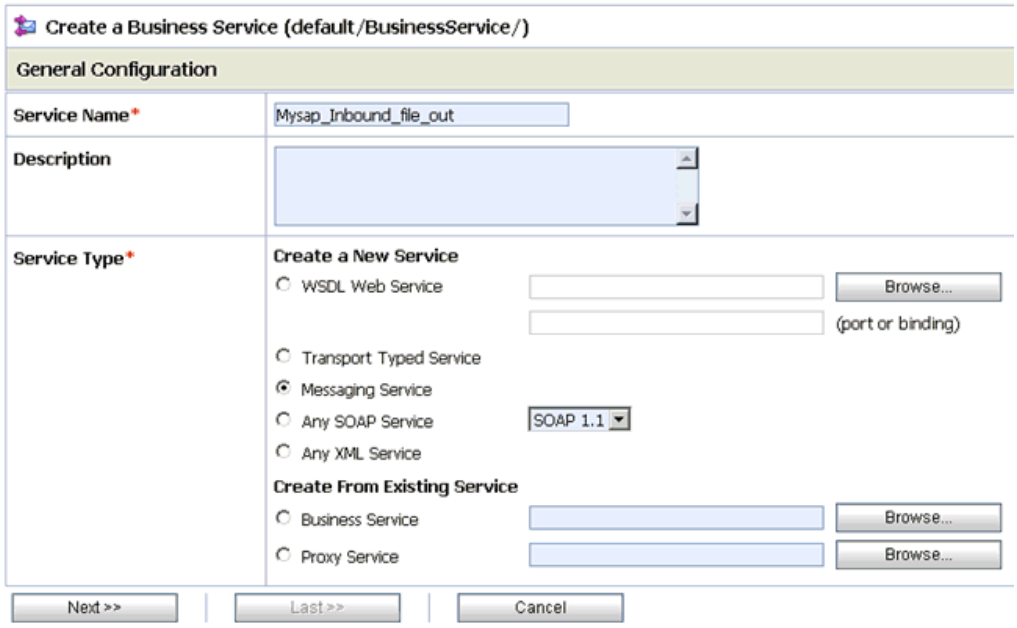
This section describes how to configure a File type Business Service using the Oracle Service Bus Console.

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

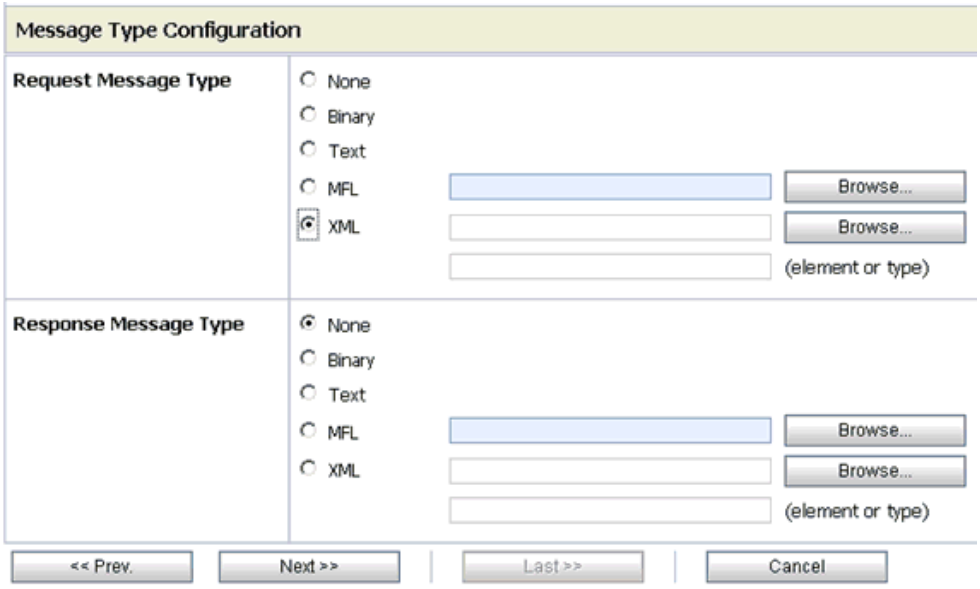


2. In the right pane, select **Business Service** from the Create Resource menu.

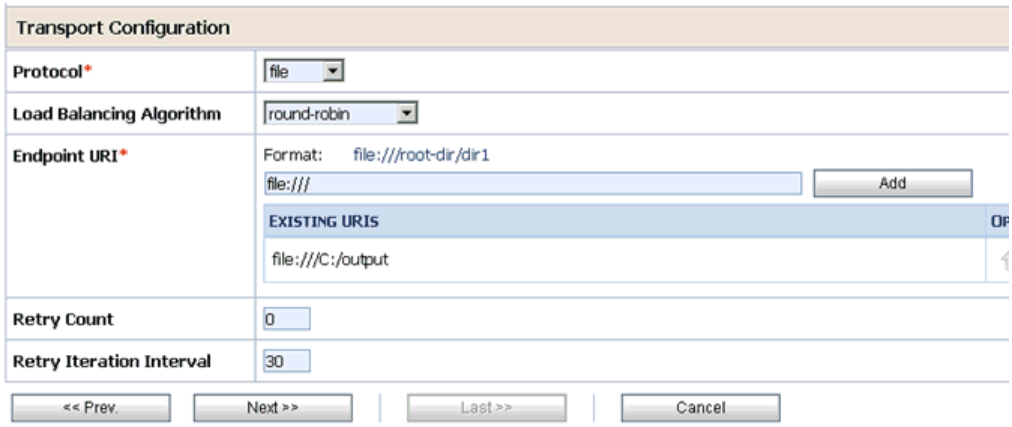
   The Create a Business Service - General Configuration page is displayed.



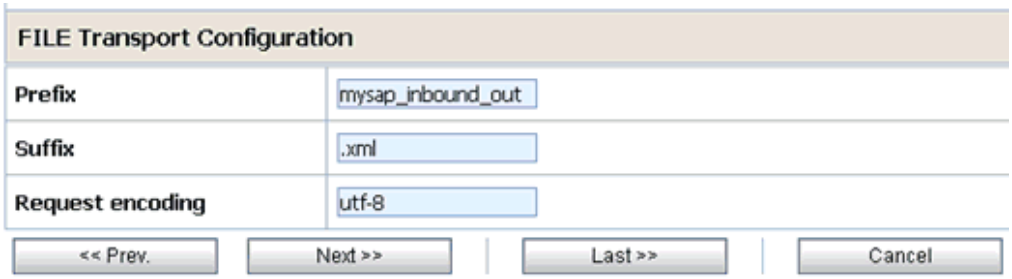
3. Provide a name for the Business Service and from the Service Type area select **Messaging Service**.
4. Click **Next**.



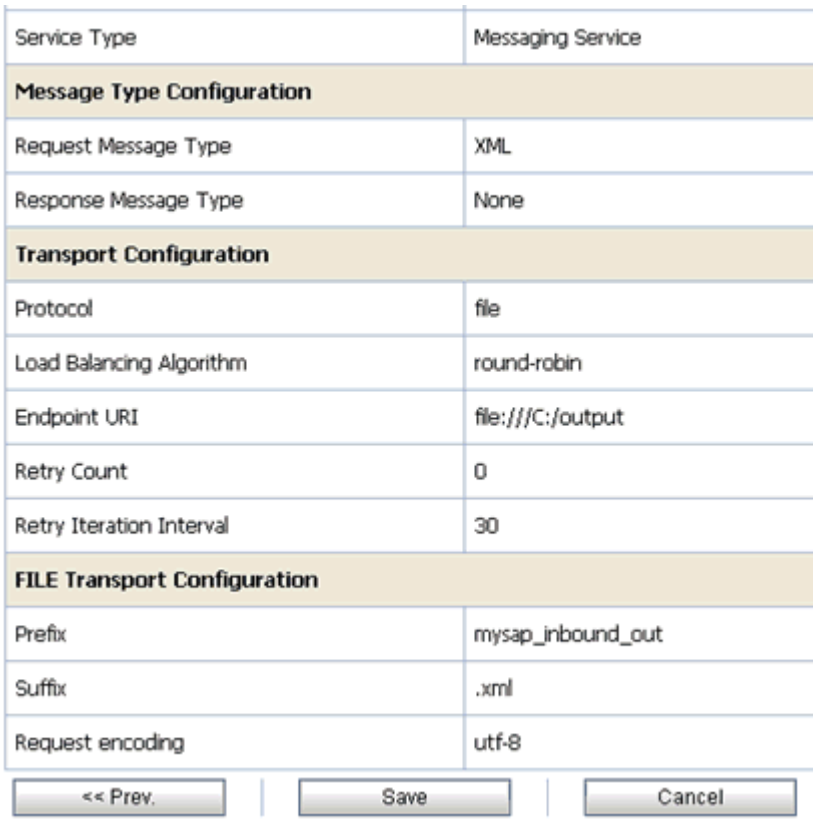
5. Select **XML** as the Request Message Type and **None** as the Response Message Type.
6. Click **Next**.



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**.



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

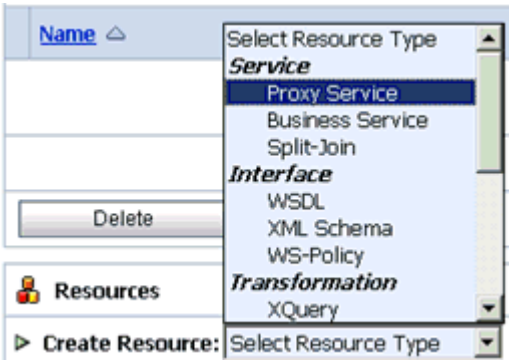


11. Review all the information for your Business Service and click **Save**.

Configuring a Proxy Service

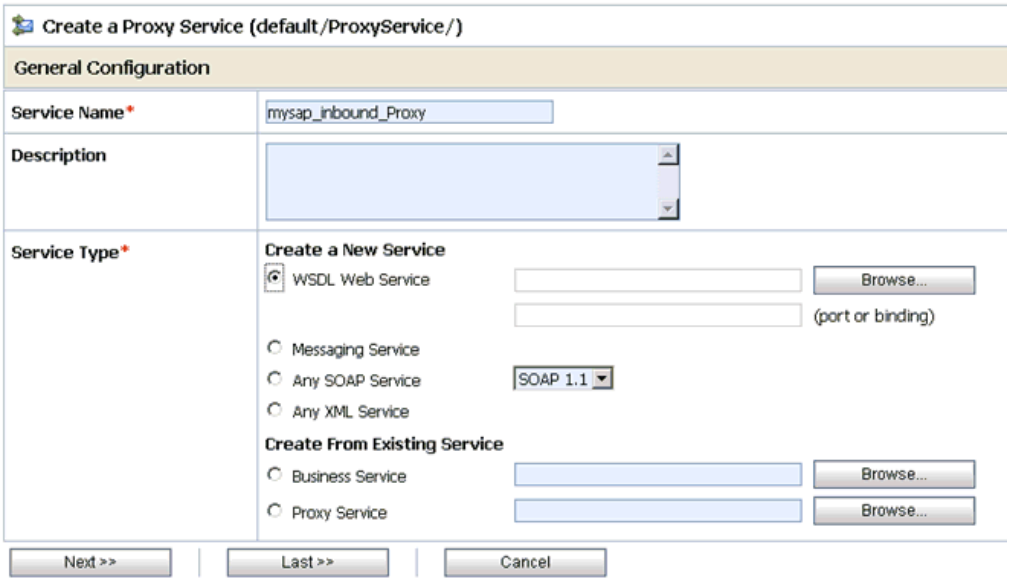
This section describes how to configure a Proxy Service using the Oracle Service Bus Console.

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.

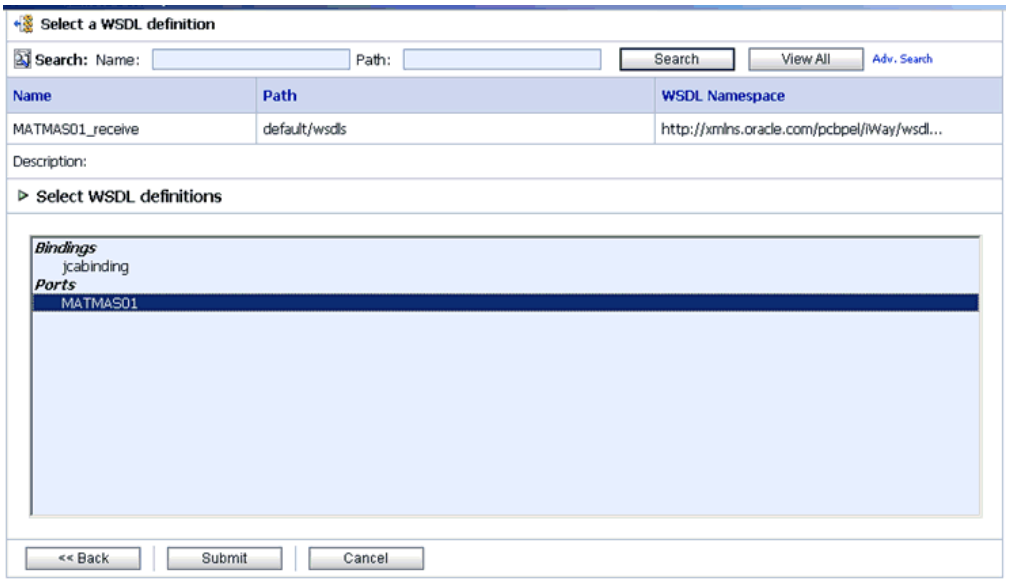
   The Create a Proxy Service - General Configuration page is displayed.



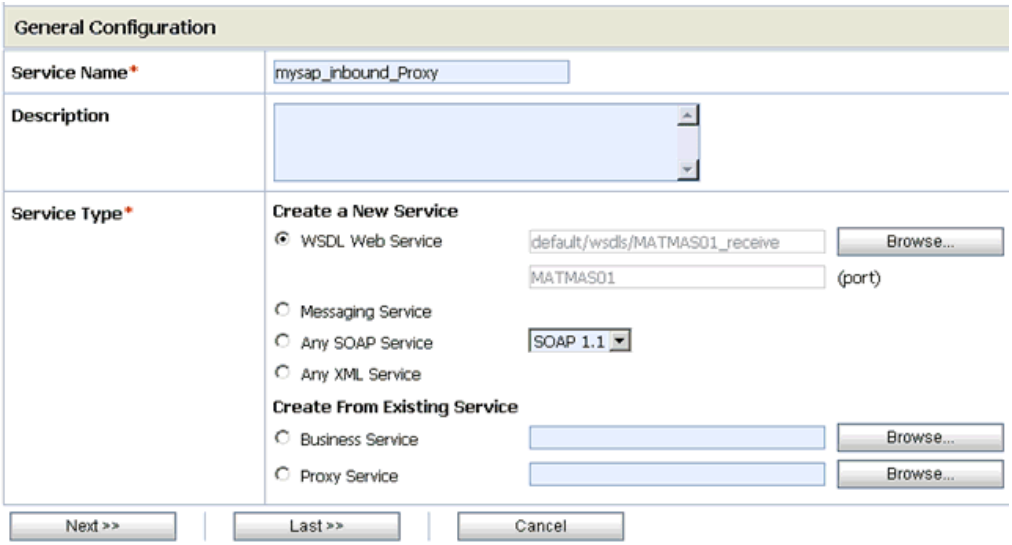
3. Provide a name for the Proxy Service and from the Service Type area select **WSDL Web Service**.

4. Click **Browse**.

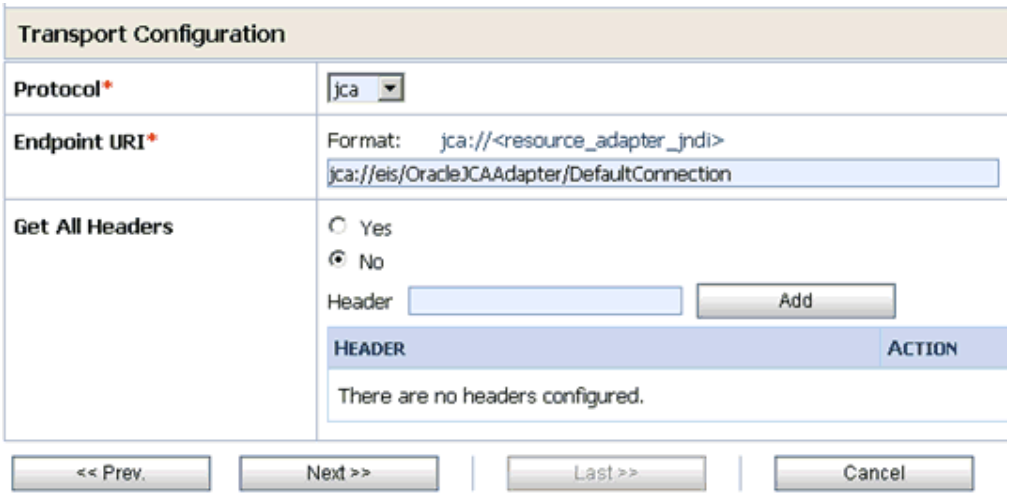
   The Select a WSDL definition dialog box is displayed.



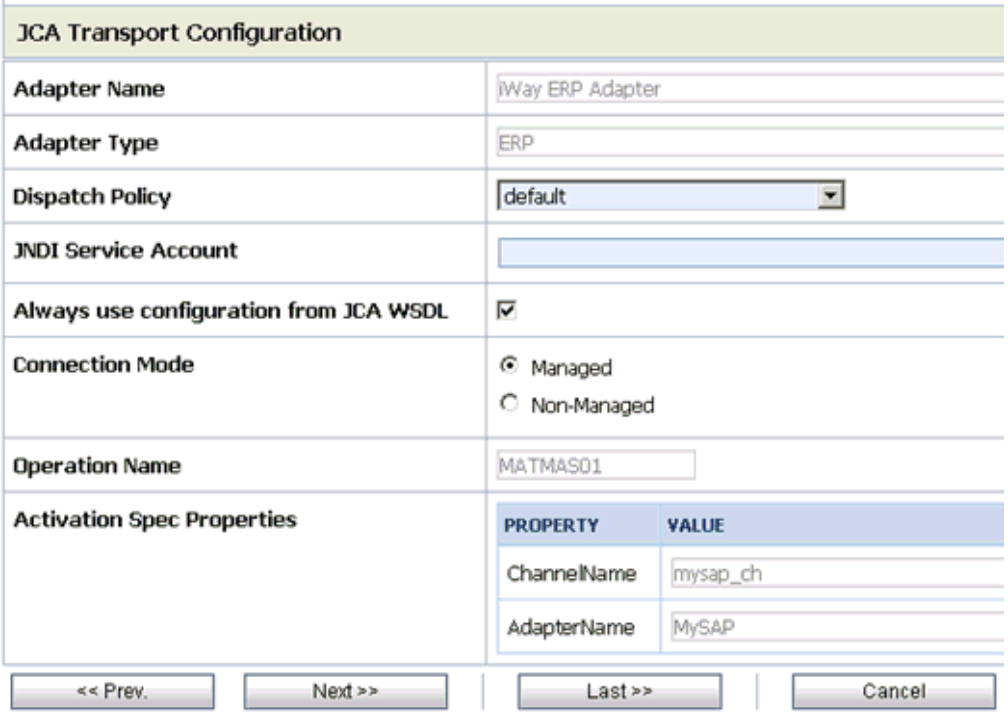
5. Select an inbound WSDL for the Proxy Service and click **Submit**, then select any WSDL definition and click **Submit**.



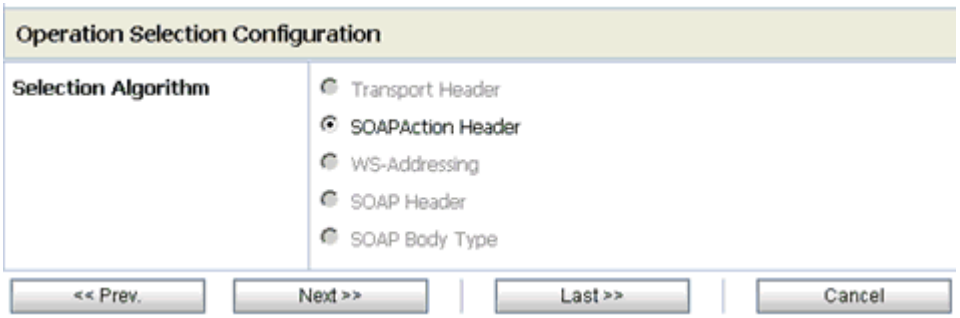
6. Click **Next**.



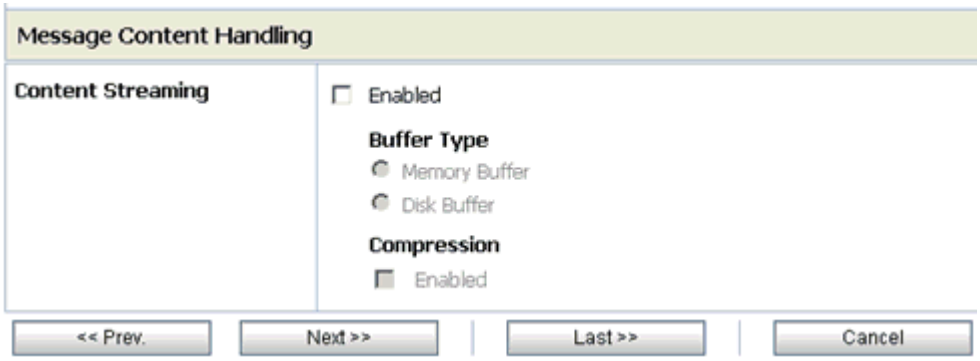
7. Click **Next**.



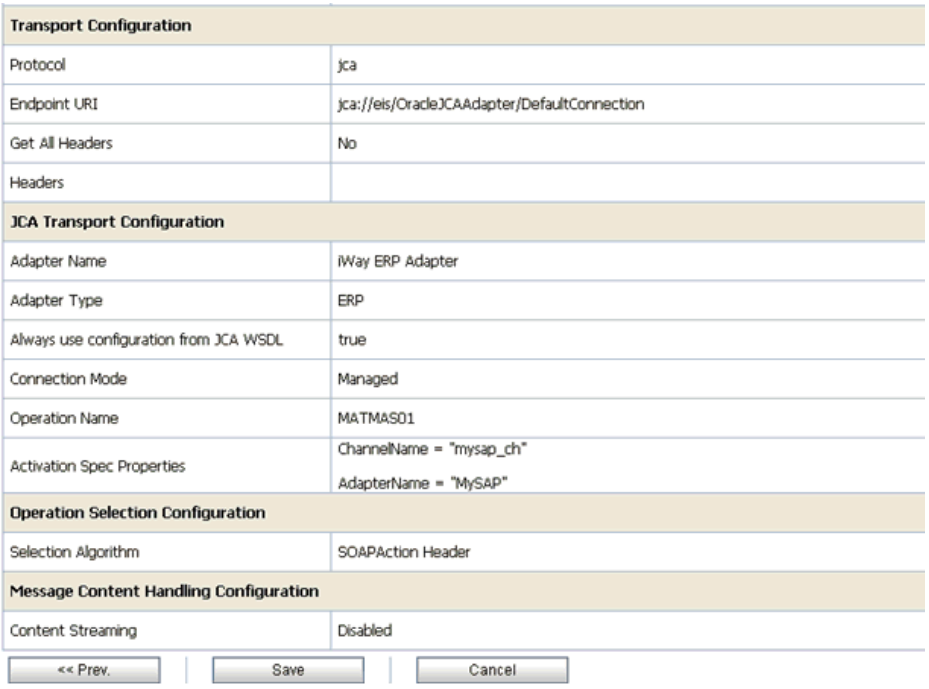
8. Click **Next**.



9. Click **Next**.



10. Click **Next**.



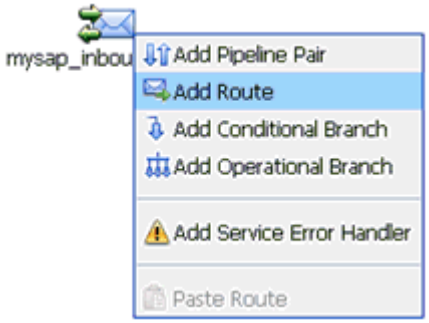
11. Review all the information for your Proxy Service and click **Save**.
12. Click **Activate** in the Change Center area to activate your changes in the Oracle Service Bus session.

Configuring a Pipeline

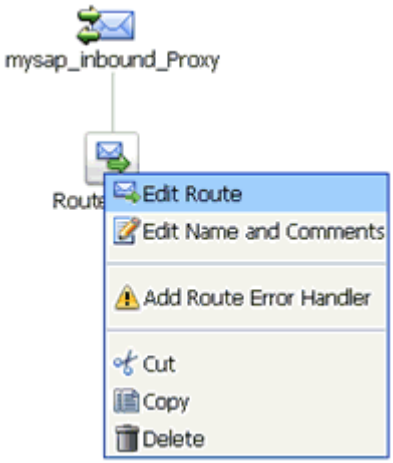
This section describes how to configure a Pipeline using the Oracle Service Bus Console.

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

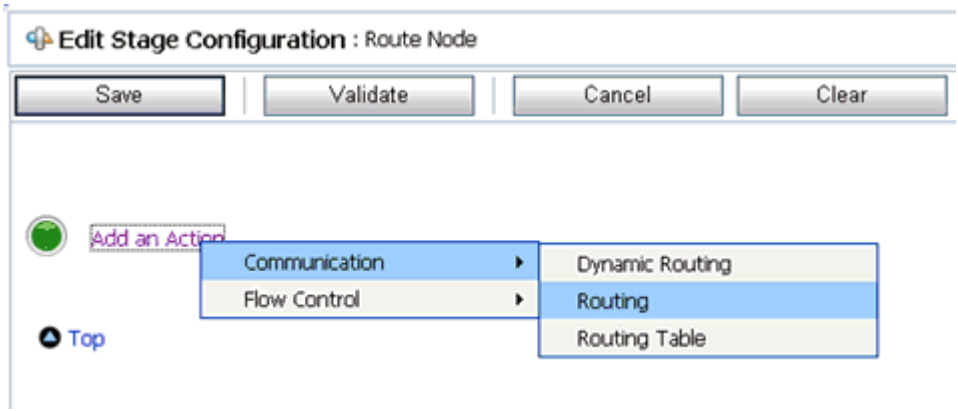
The Edit Message Flow workspace area is displayed.



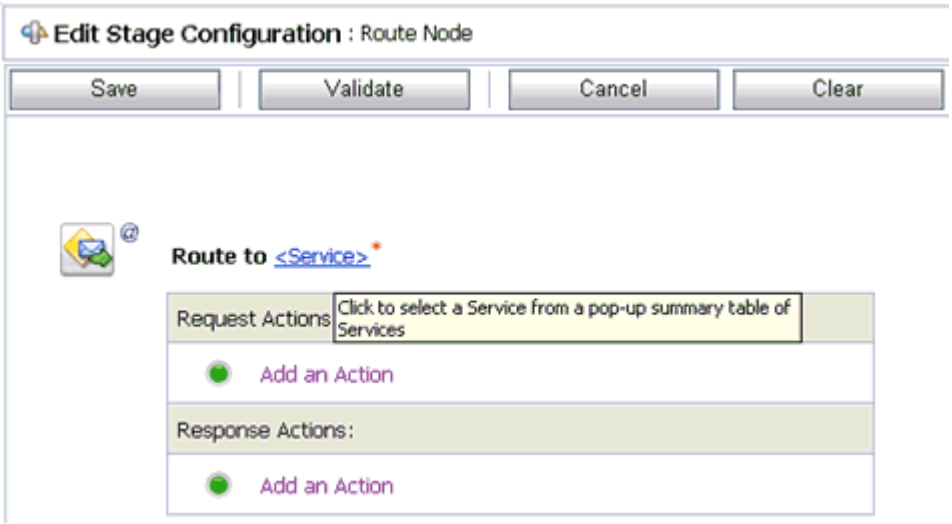
2. Click the **Proxy Service** icon and select **Add Route** from the context menu, as shown in the following image.



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

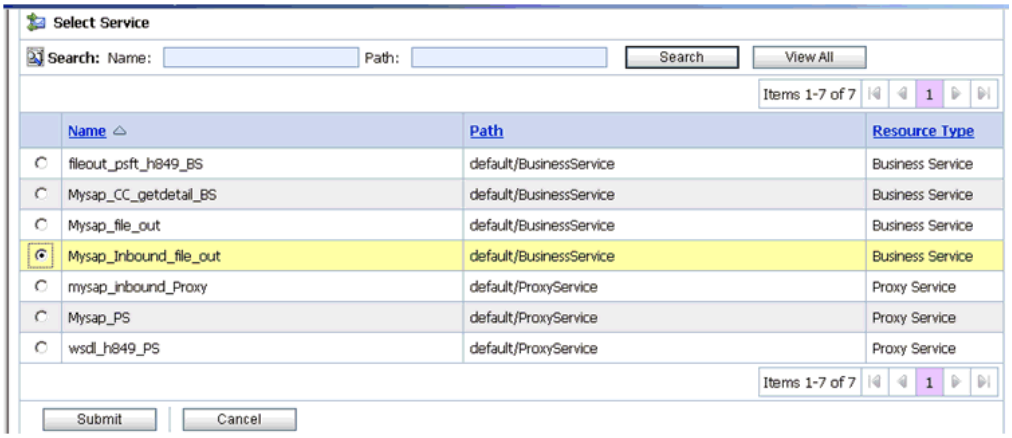


4. Click **Add an Action**, select **Communication** from the context menu, and click **Routing**.



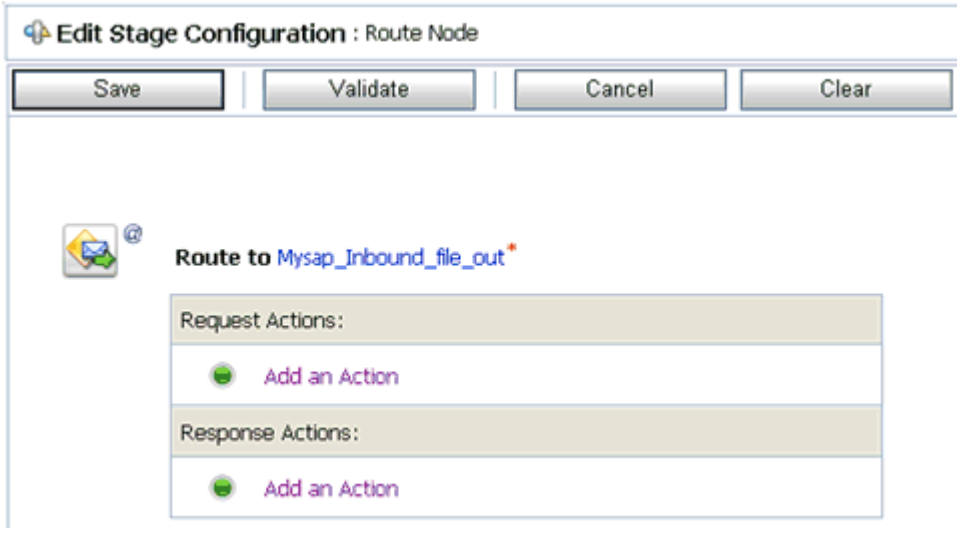
5. Click **<Service>**.

   The Select Service dialog box is displayed.

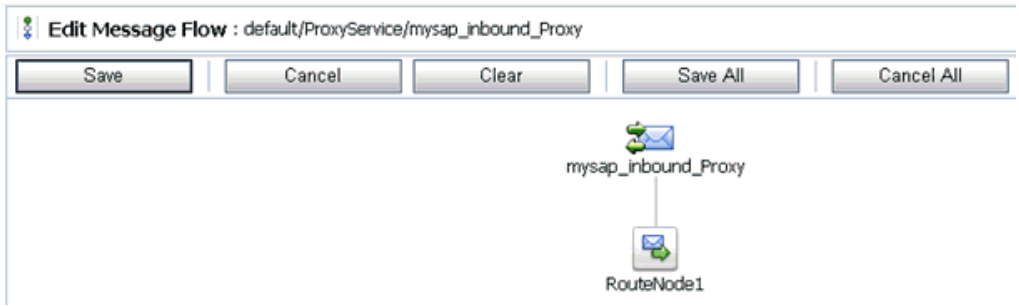


6. Select a File type Business Service and click **Submit**.

   You are returned to the Edit Stage Configuration workspace area.

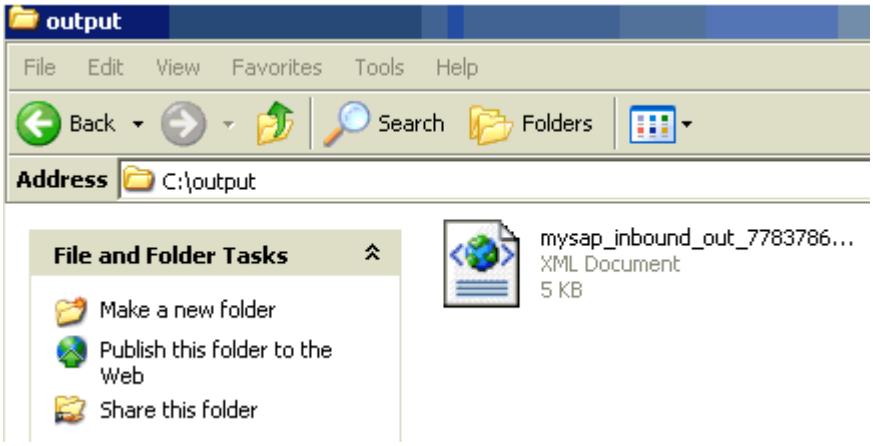


7. Click **Validate** and then **Save**.



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 SAP R/3 system and you will receive the output XML in the destination folder.



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:

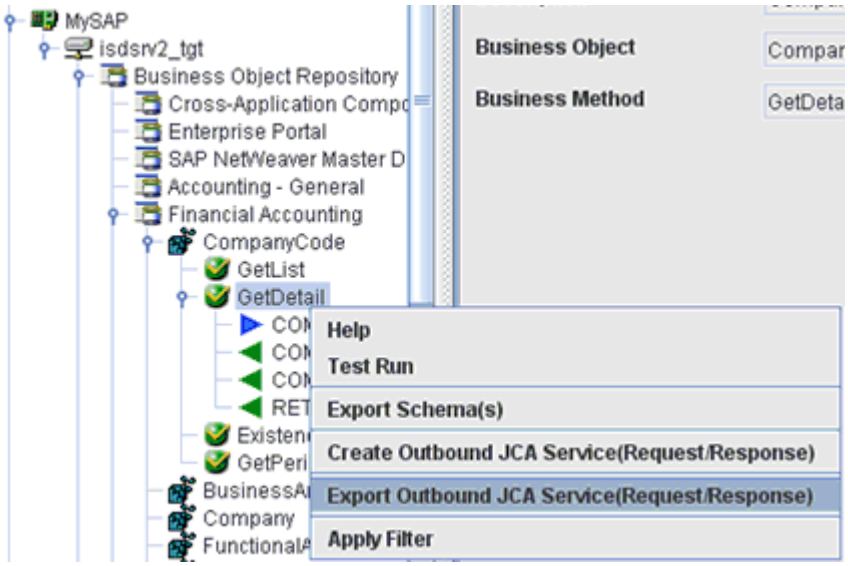
- Publishing a WSDL From Application Explorer to Oracle Service Bus
- Starting Oracle Service Bus and Creating Project Folders
- Configuring a File Type Business Service
- Configuring a WSDL Type Business Service
- Configuring a Proxy Service
- Configuring a Pipeline

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.

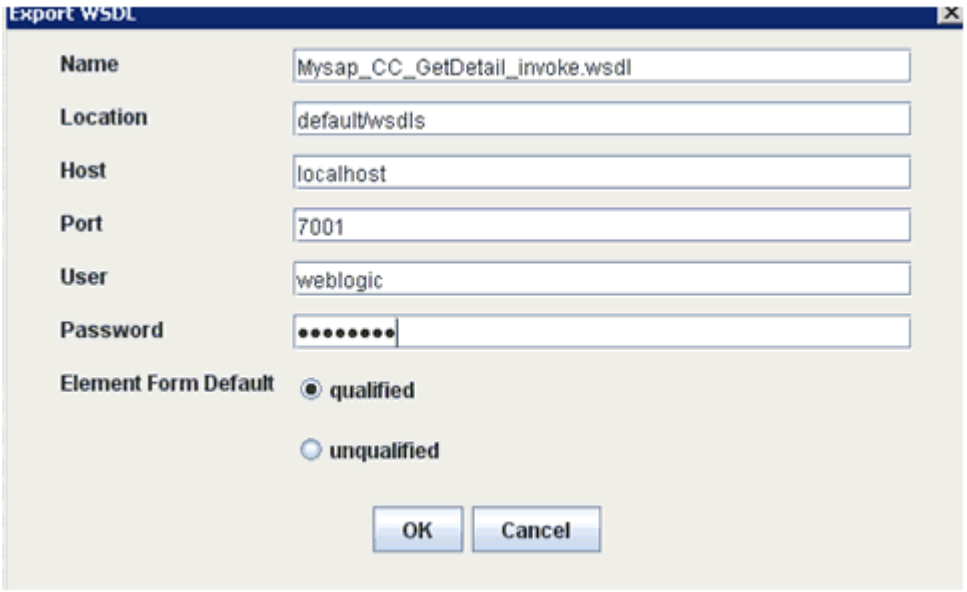
1. Start Application Explorer, connect to a J2CA configuration, and connect to a SAP R/3 target.

   For more information, see Chapter 3, "Configuring Oracle Application Adapter for SAP R/3".



2. Expand the **CompanyCode** business object.
3. Right-click the **GetDetail** method and select **Export Outbound JCA Service(Request/Response)** from the context menu.

   The Export WSDL dialog box is displayed.



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 forward 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:

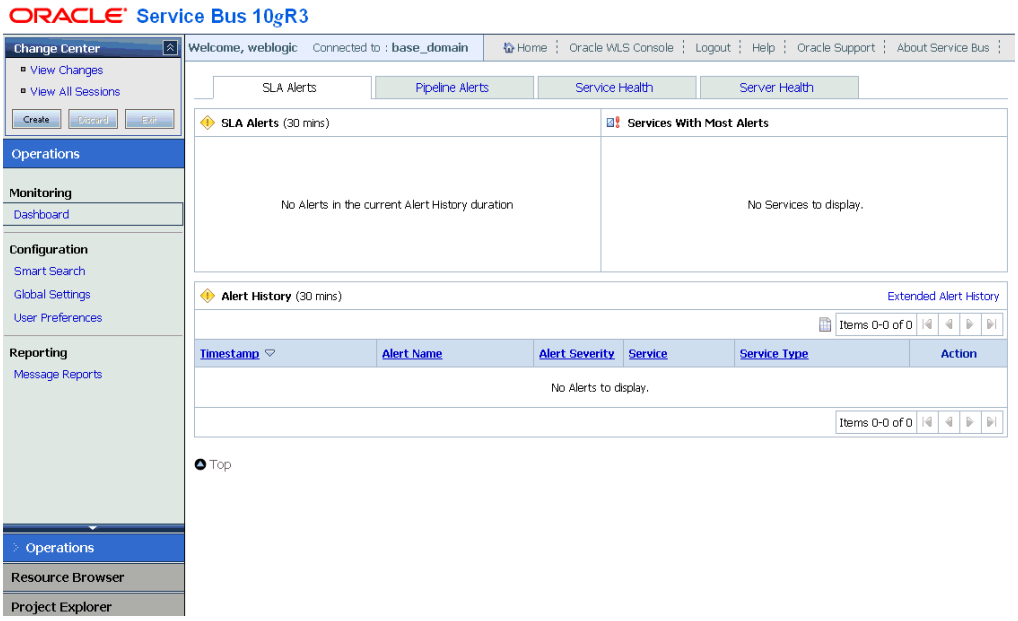
   `http://hostname:port/sbconsole`

   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.

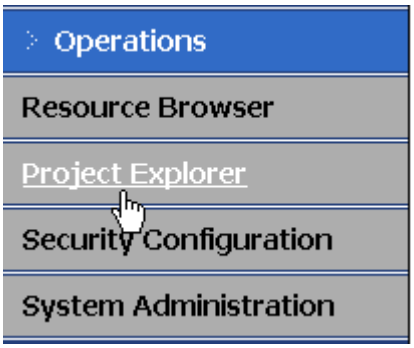
3. Log on to the Oracle Service Bus Console using a valid user name and password.

   The Oracle Service Bus Console home page is displayed.

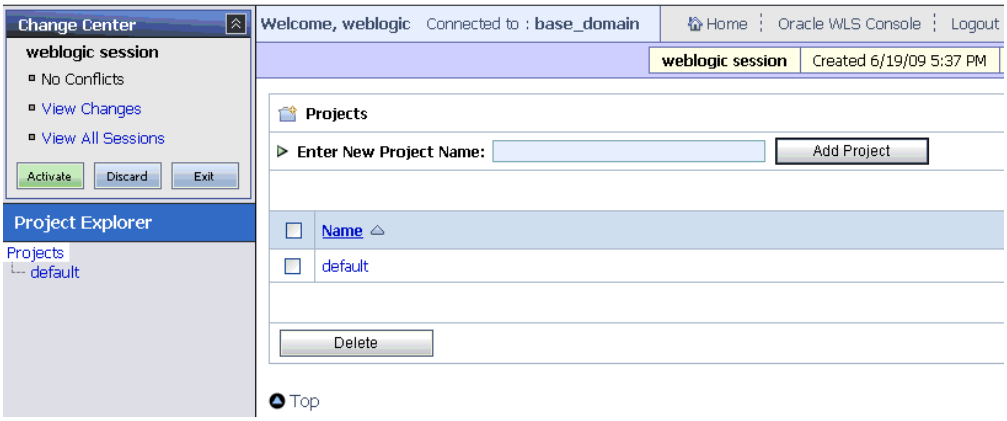


4. Click **Create** in the Change Center area to start a new Oracle Service Bus session.

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

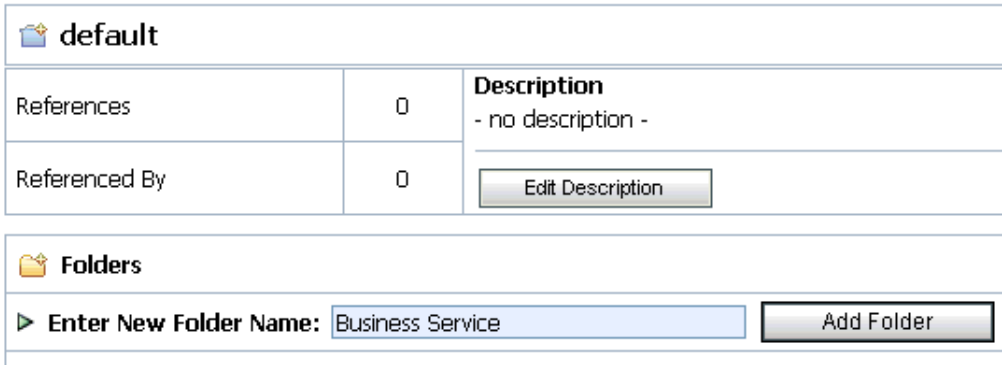


The Project Explorer page is displayed.



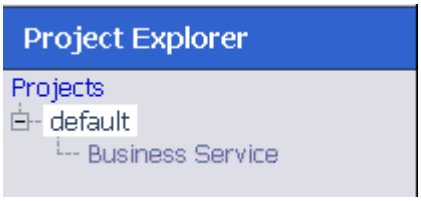
6. Click the **default** project node in the left pane.

   The default project page is displayed.



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.



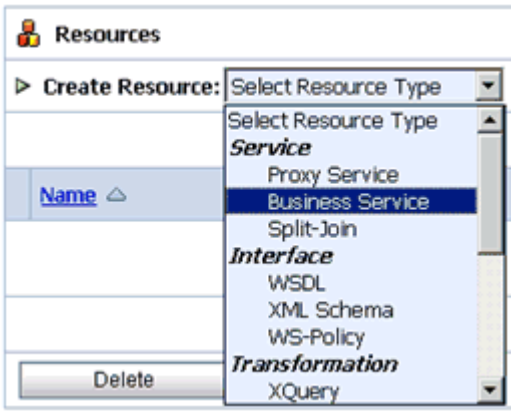
8. In the Enter New Folder Name field, type **Proxy Service** and click **Add Folder**.

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

Configuring a File Type Business Service

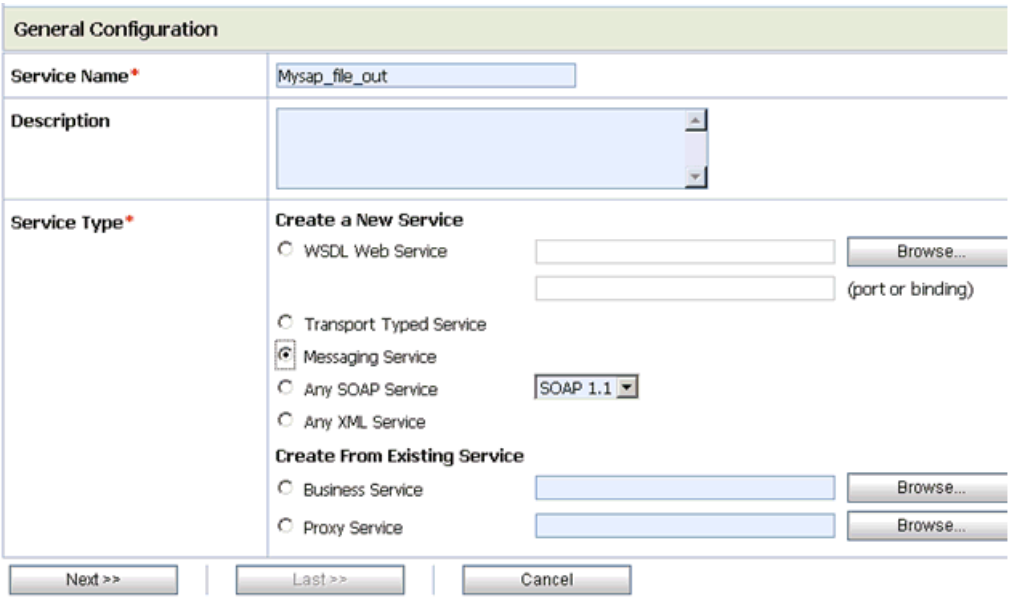
This section describes how to configure a File type Business Service using the Oracle Service Bus Console.

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

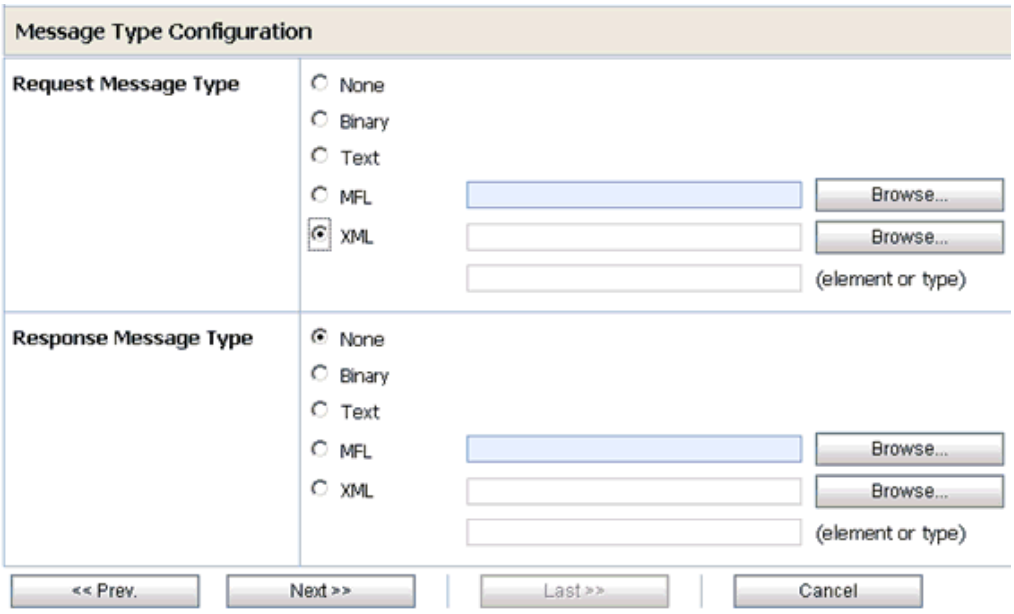


2. In the right pane, select **Business Service** from the Create Resource menu.

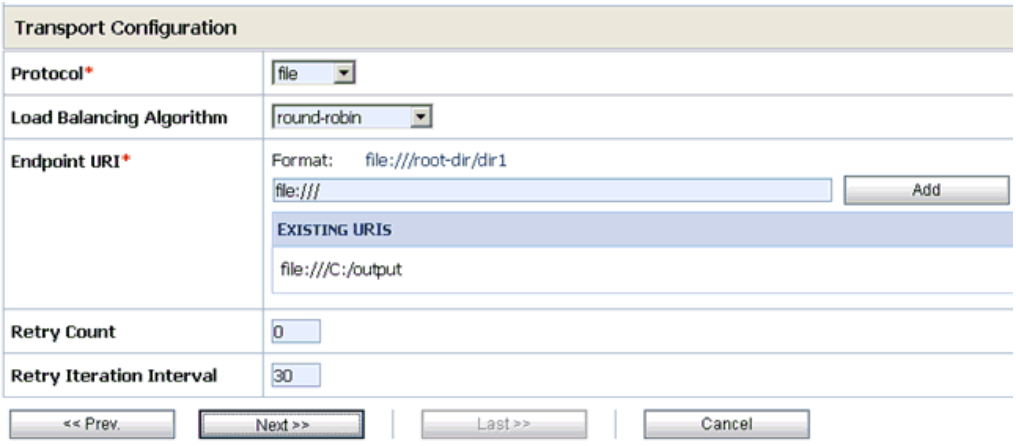
   The Create a Business Service - General Configuration page is displayed.



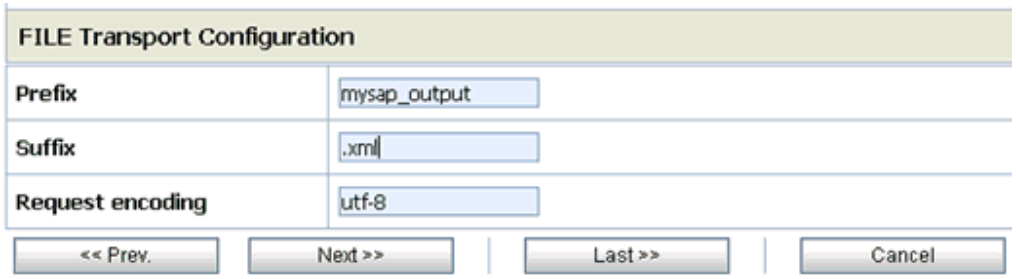
3. Provide a name for the Business Service and from the Service Type area select **Messaging Service**.
4. Click **Next**.



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



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**.



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	mysap_output
Suffix	.xml
Request encoding	utf-8

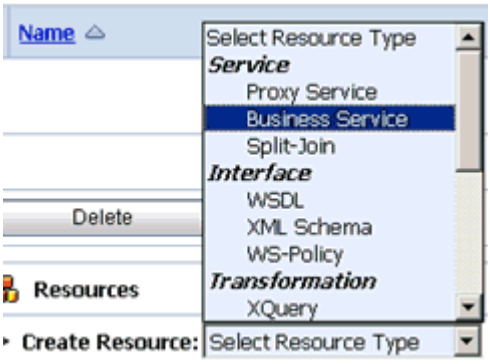
<< Prev. | Save | Cancel

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

Configuring a WSDL Type Business Service

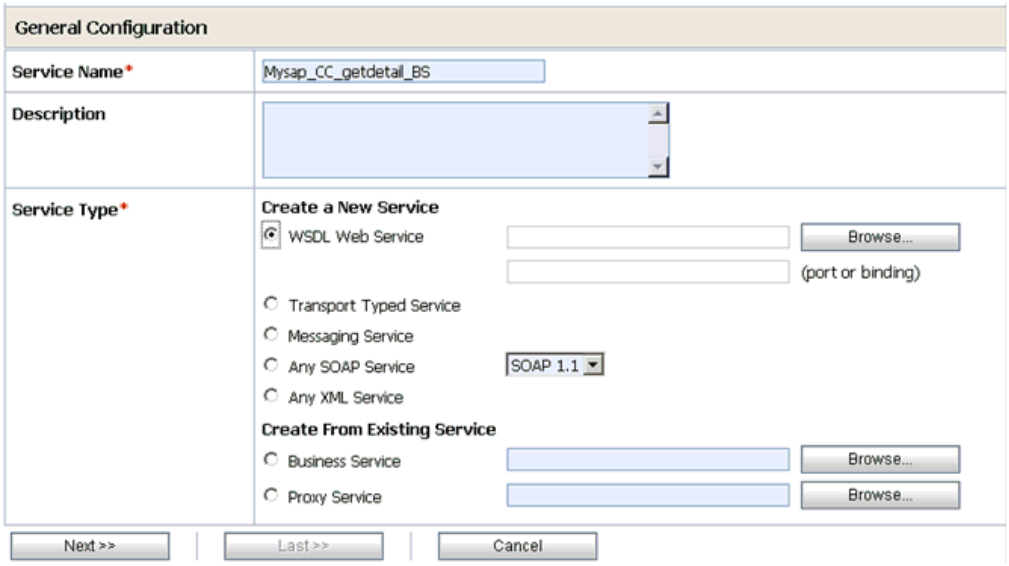
This section describes how to configure a WSDL type Business Service using the Oracle Service Bus Console.

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



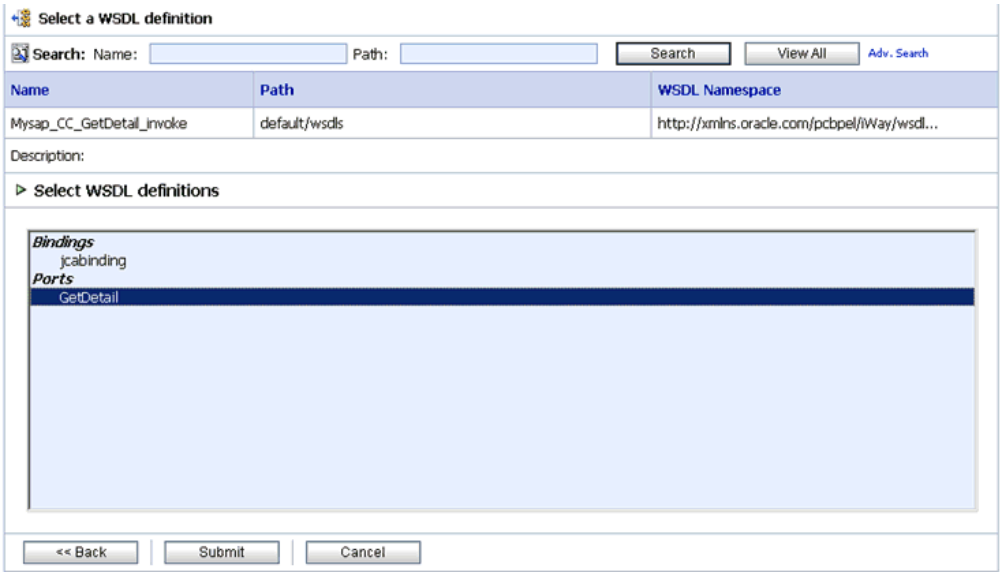
2. In the right pane, select **Business Service** from the Create Resource menu.

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

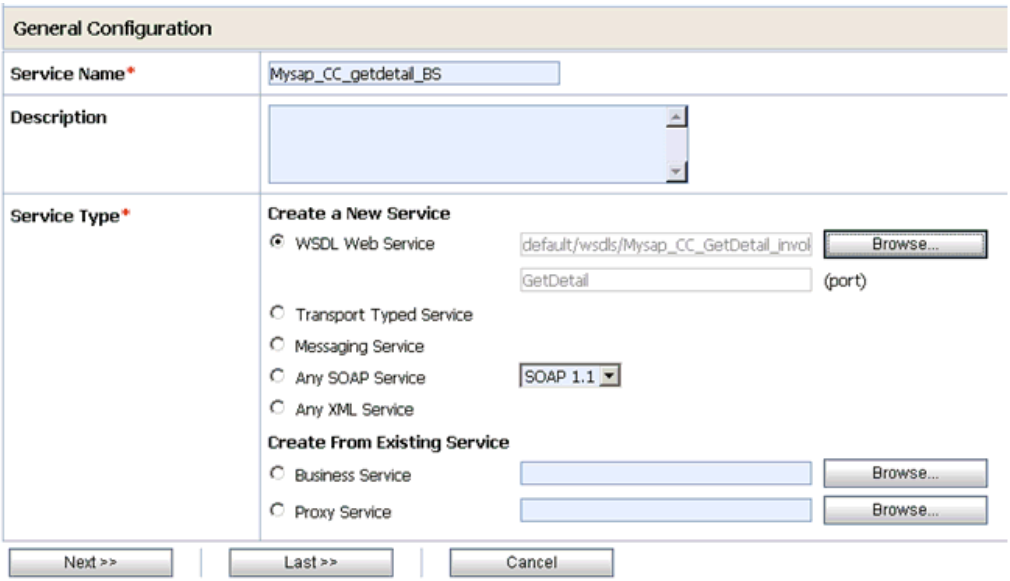


3. Provide a name for the Business Service and from the Service Type area select **WSDL Web Service**.
4. Click **Browse**.

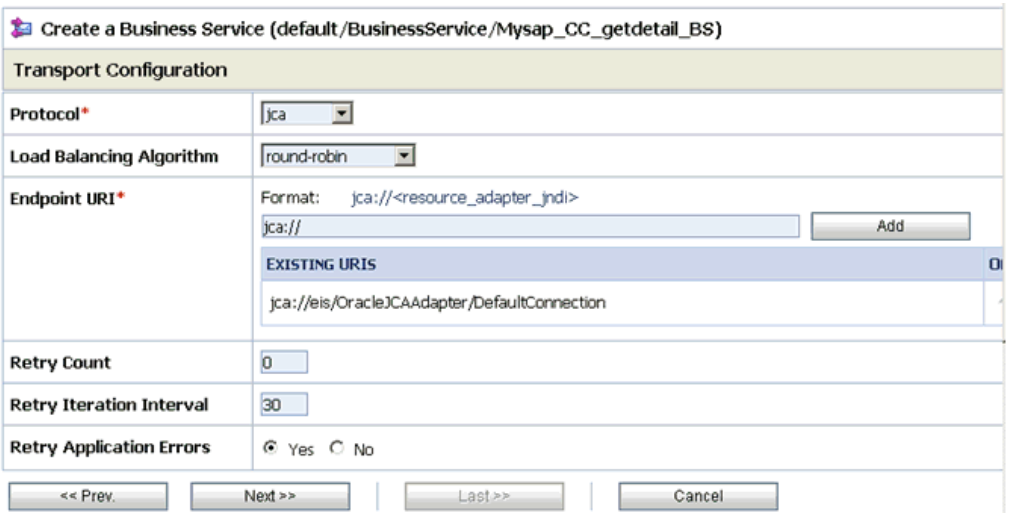
   The Select a WSDL Definition dialog box is displayed.



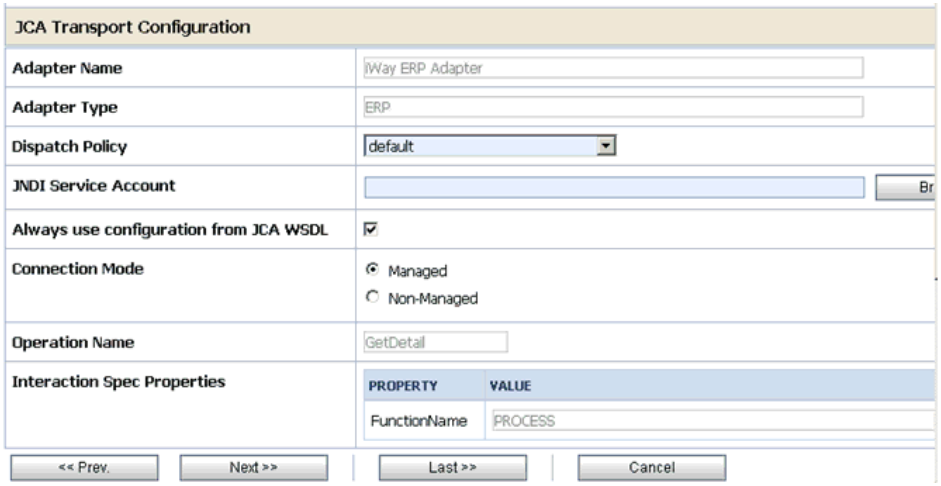
5. Select the required WSDL and then select the WSDL definition under the Ports section.
6. Click **Submit**.



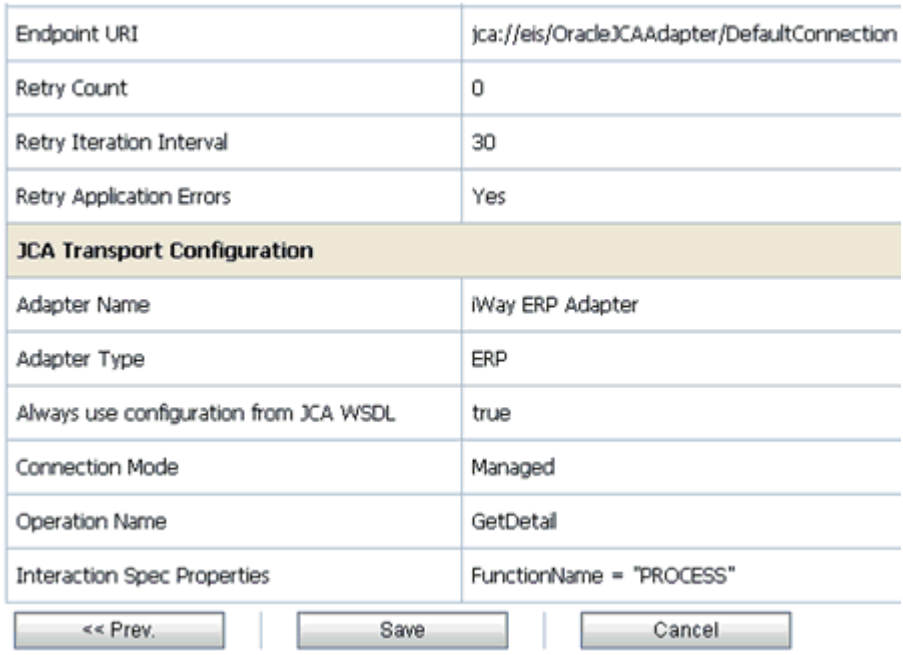
7. Click **Next**.



8. Select **jca** from the Protocol drop-down list.
9. Click **Next**.



10. Click **Next**.

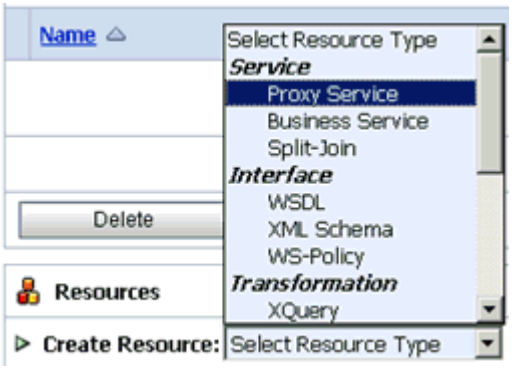


11. Review all the information for your Business Service and click **Save**.

Configuring a Proxy Service

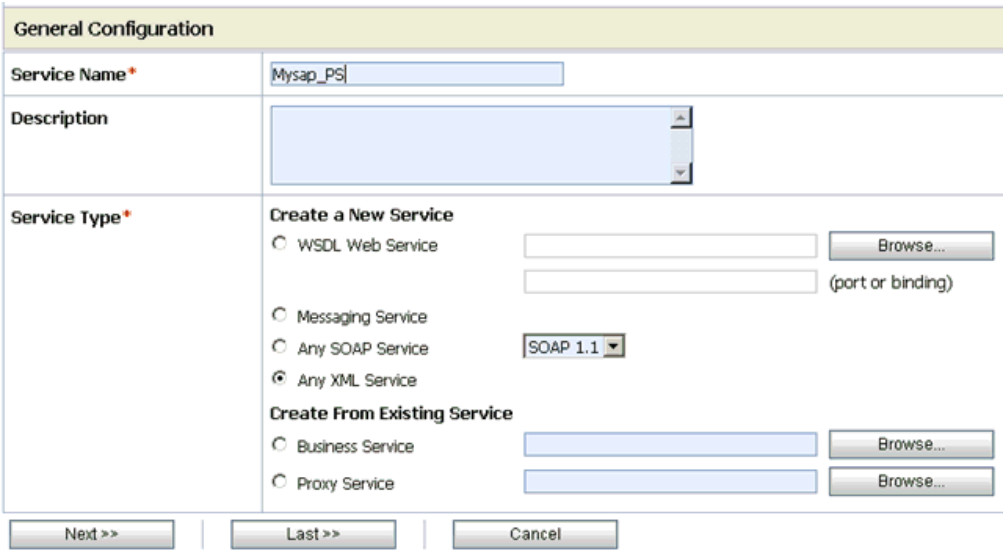
This section describes how to configure a Proxy Service using the Oracle Service Bus Console.

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

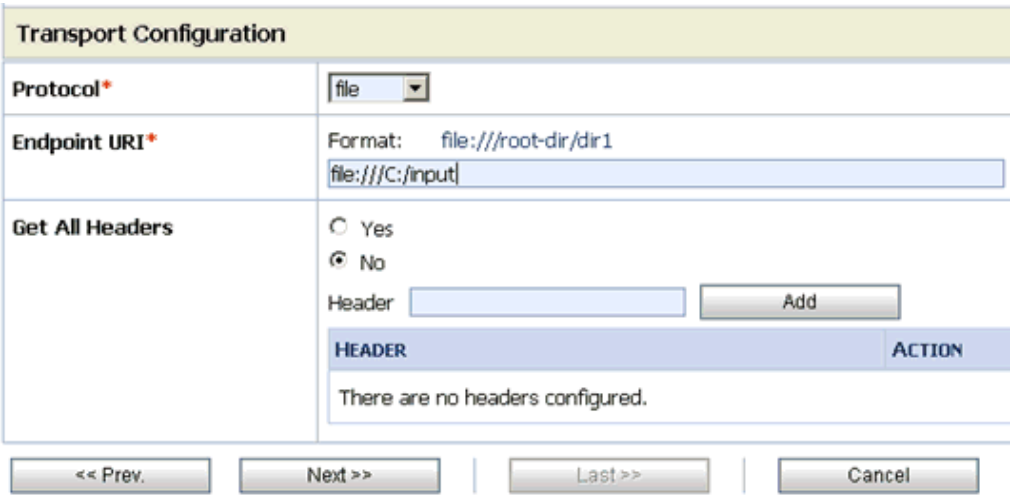


2. In the right pane, select **Proxy Service** from the Create Resource menu.

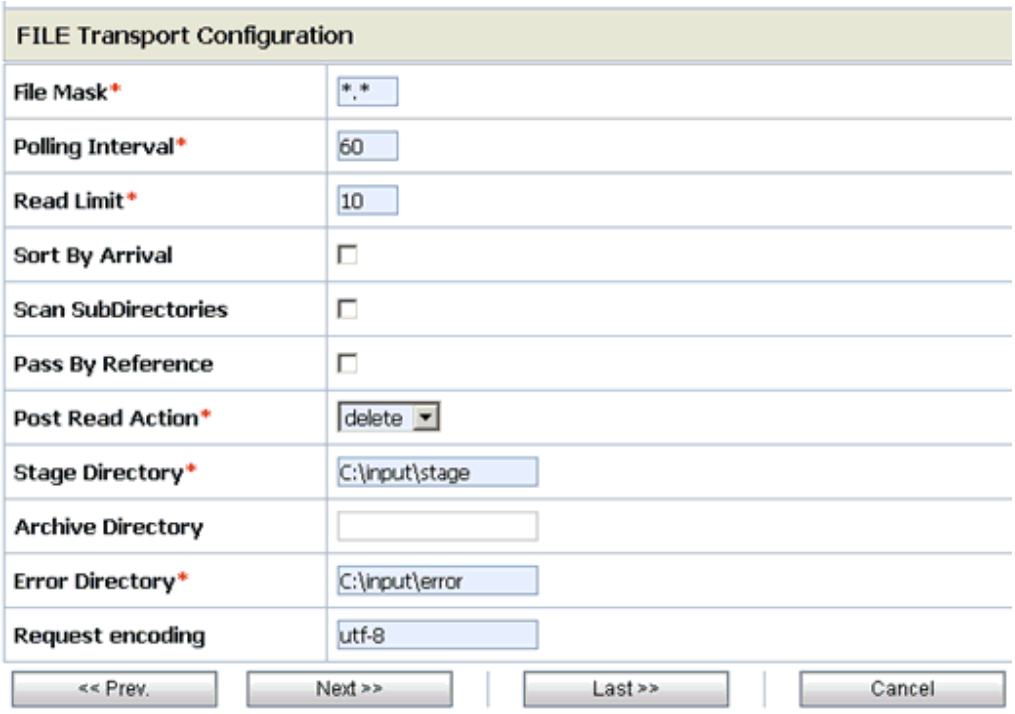
   The Create a Proxy Service - General Configuration page is displayed.



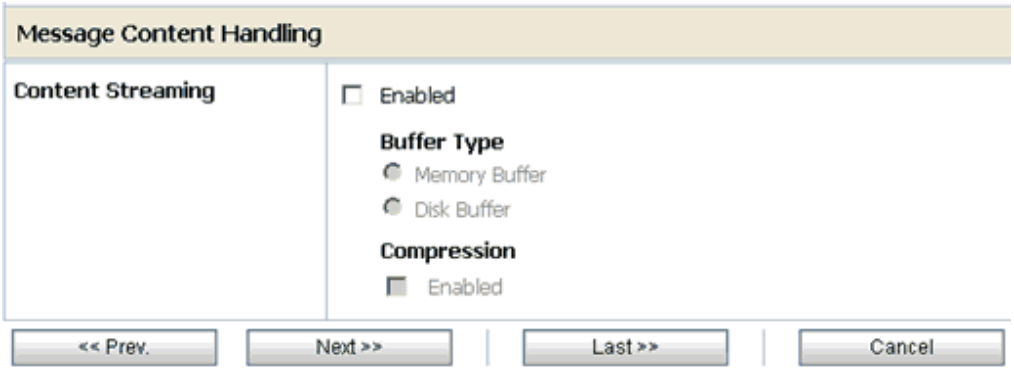
3. Provide a name for the Proxy Service and from the Service Type area select **Any XML Service**.
4. Click **Next**.



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**.



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



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

<< Prev. | Save | Cancel

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

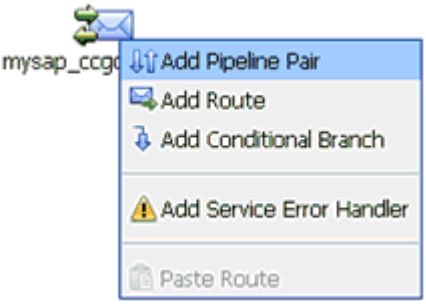
Configuring a Pipeline

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.

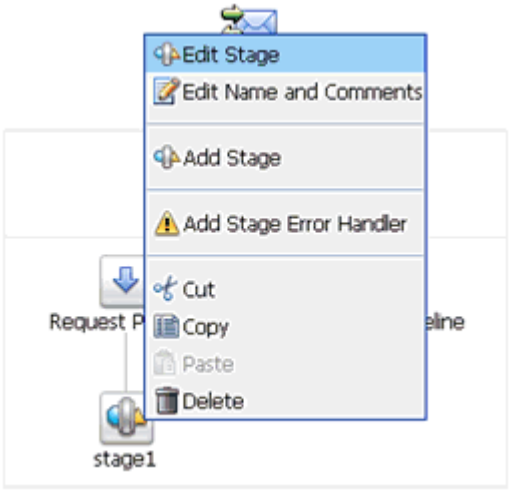
The Edit Message Flow workspace area is displayed.

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



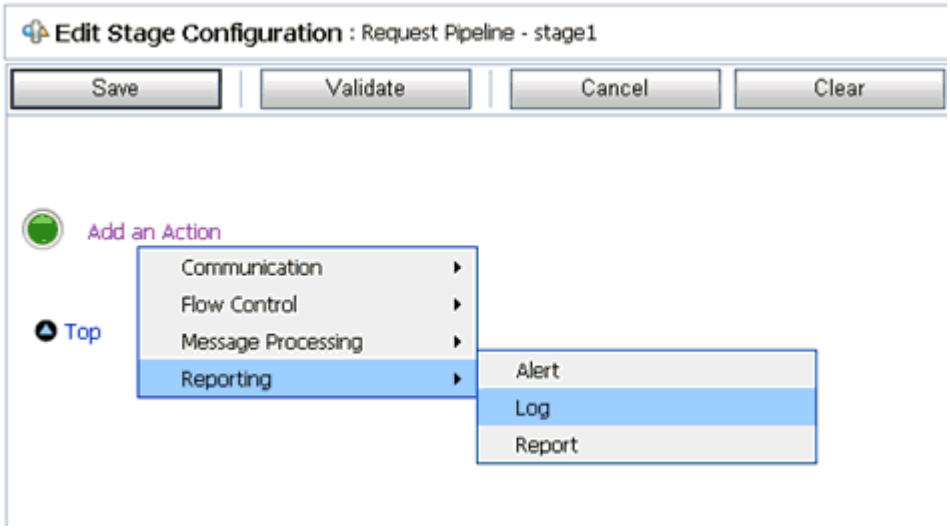
3. Click the **Request Pipeline** icon and select **Add Stage** from the context menu.

The Stage1 icon is added below the Request Pipeline icon.

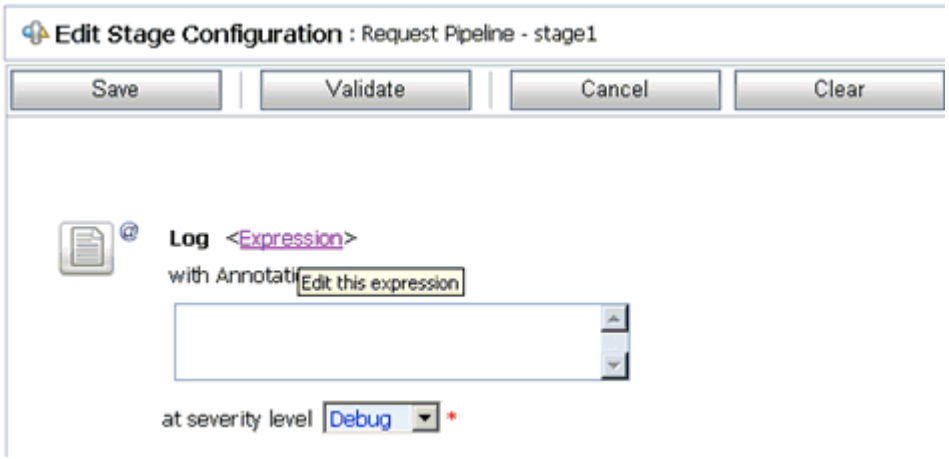


4. Click the **Stage1** icon and select **Edit Stage** from the context menu.

The Edit Stage Configuration workspace area is displayed.

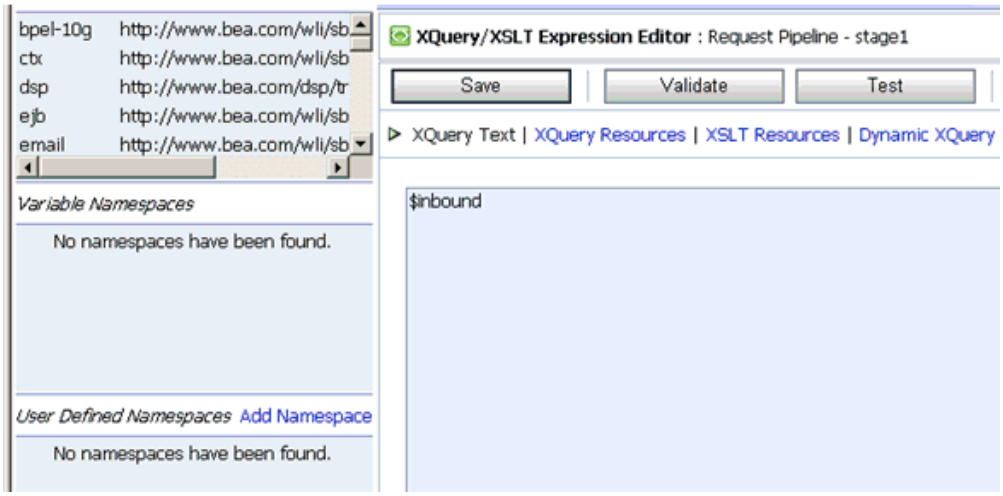


5. Click **Add an Action**, select **Reporting** from the context menu, and click **Log**.



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

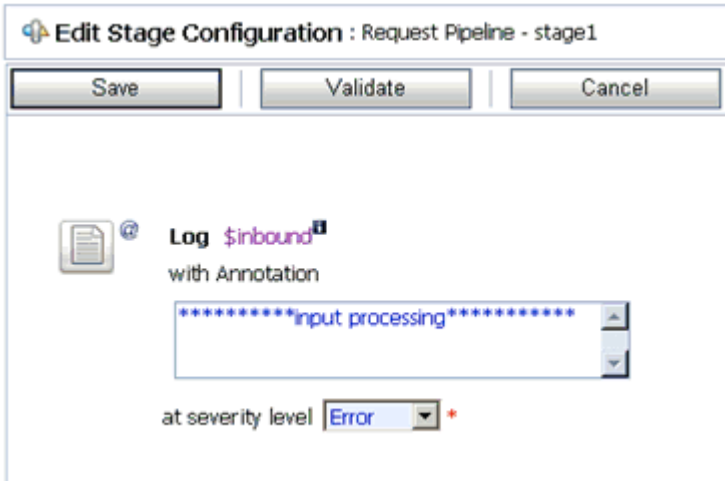
   The XQuery/XSLT Expression Editor is displayed.



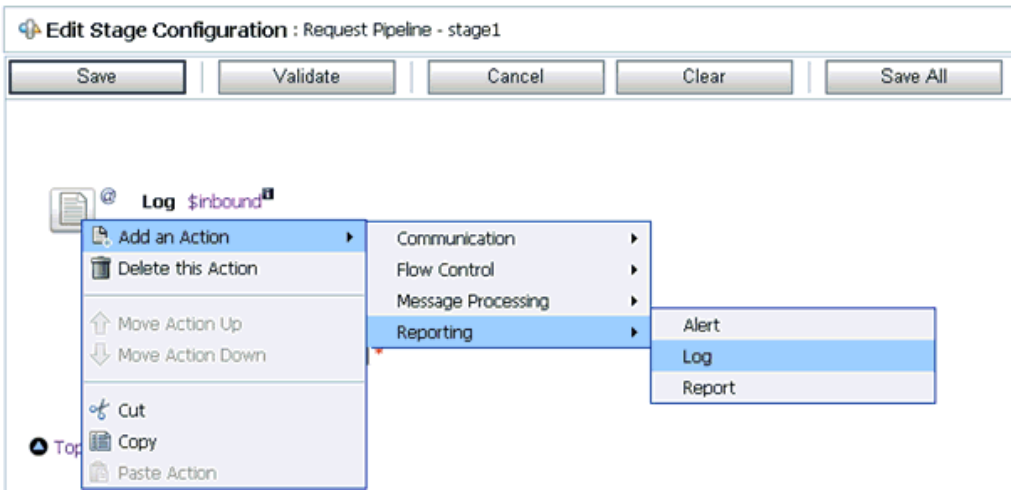
7. In the XQuery Text area, type **$inbound**.

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

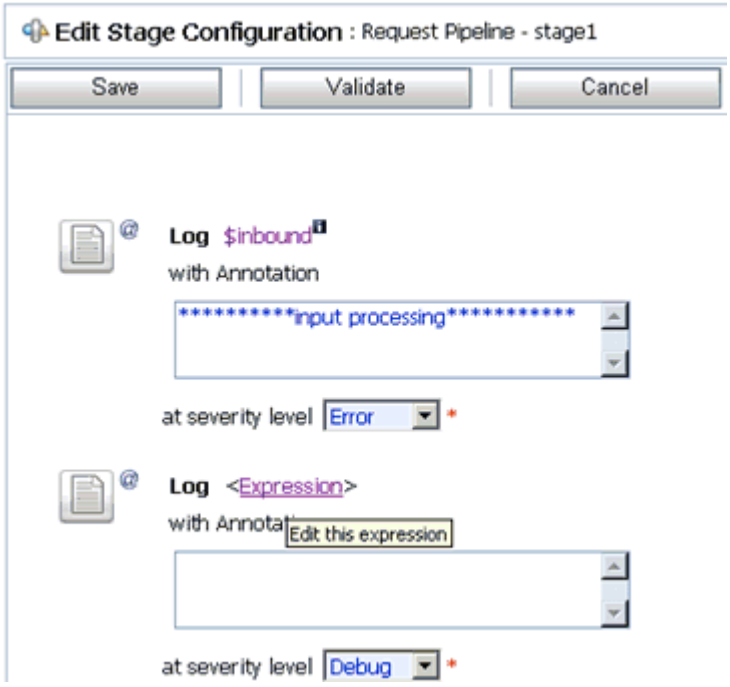
   You are returned to the Edit Stage Configuration workspace area.



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.

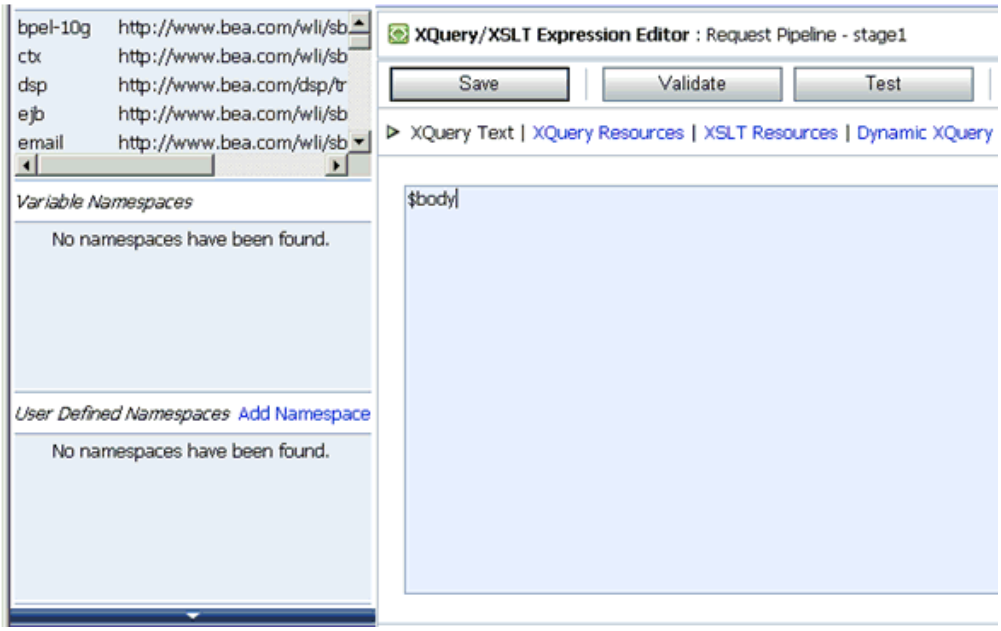


   A new Log configuration is added, as shown in the following image.



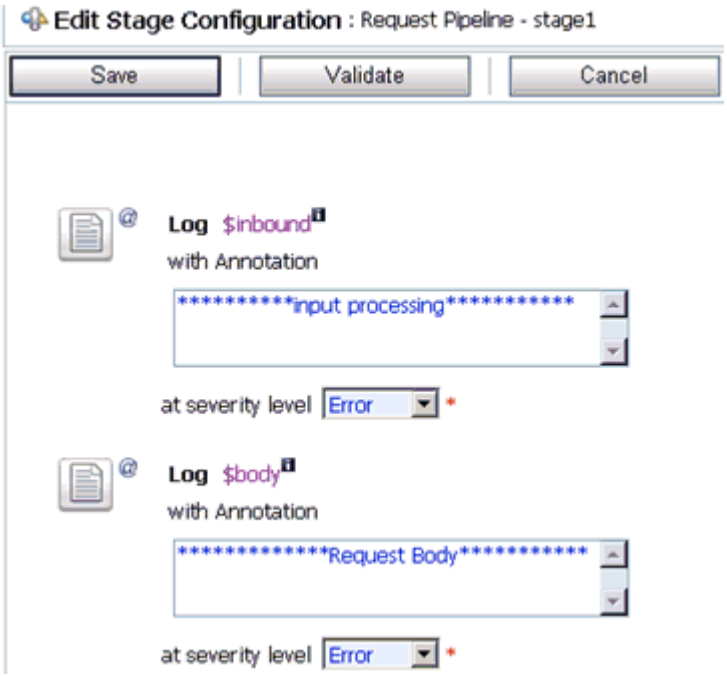
12. Click **<Expression>** to edit the expression.

    The XQuery/XSLT Expression Editor is displayed.

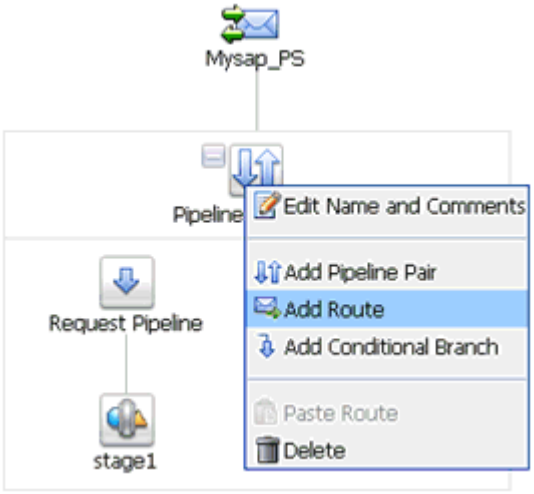


13. In the XQuery Text area, type **$body**.
14. Click **Validate** and then **Save**.

    You are returned to the Edit Stage Configuration workspace area.

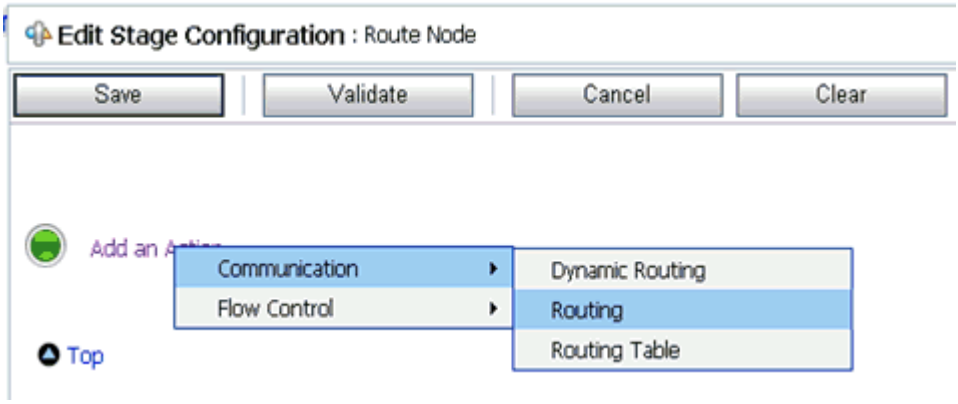


15. Type any annotation/comments in the text box (for example, **********Request Body**********).
16. Select **Error** from the severity level drop-down list.
17. Click **Validate** and then **Save**.

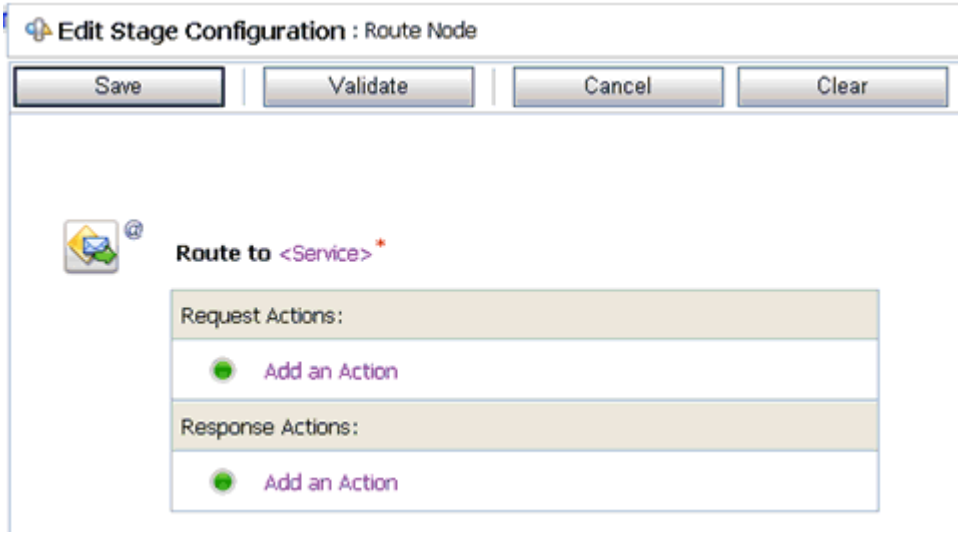


18. Click the **PipelinePairNode1** icon and select **Add Route** from the context menu.

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

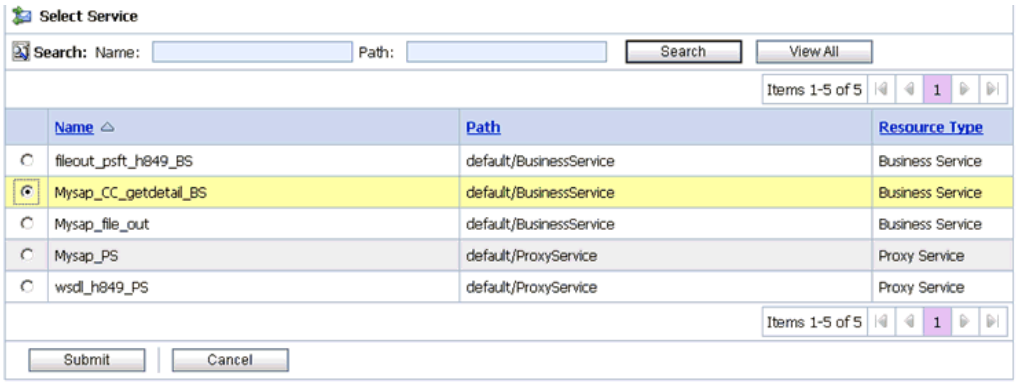


20. Click **Add an Action**, select **Communication** from the context menu, and click **Routing**.



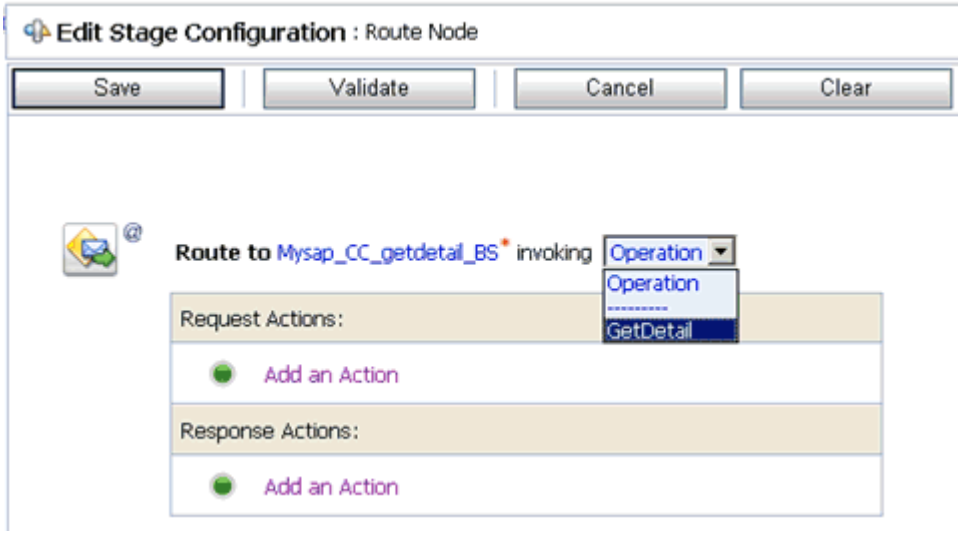
21. Click **<Service>**.

    The Select Service dialog box is displayed.

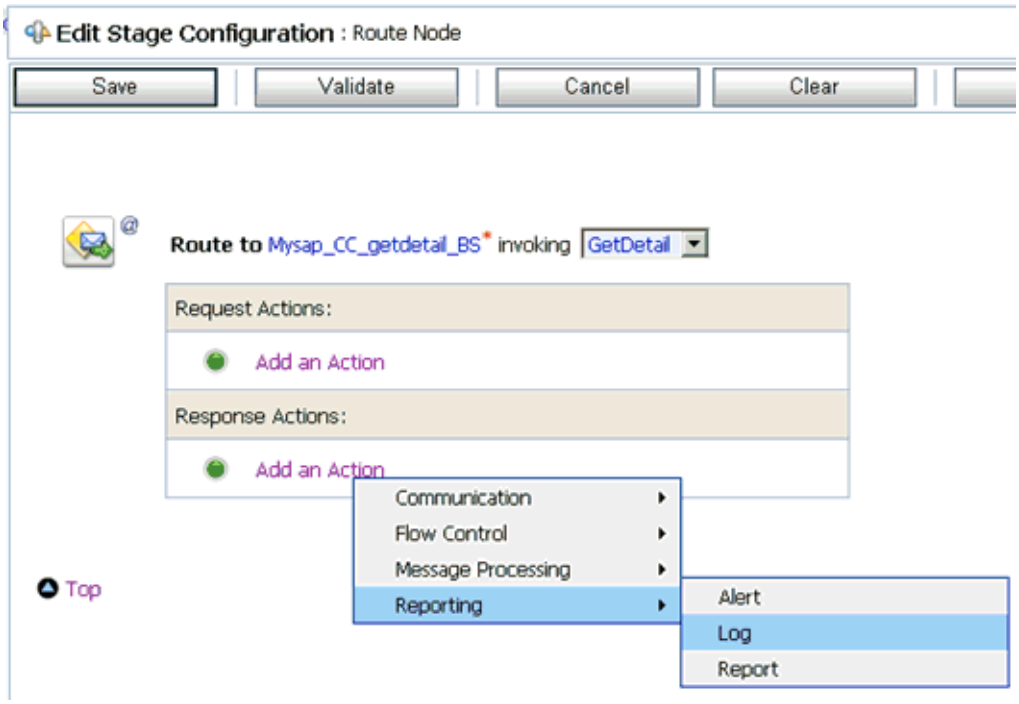


22. Select a WSDL type Business Service and click **Submit**.

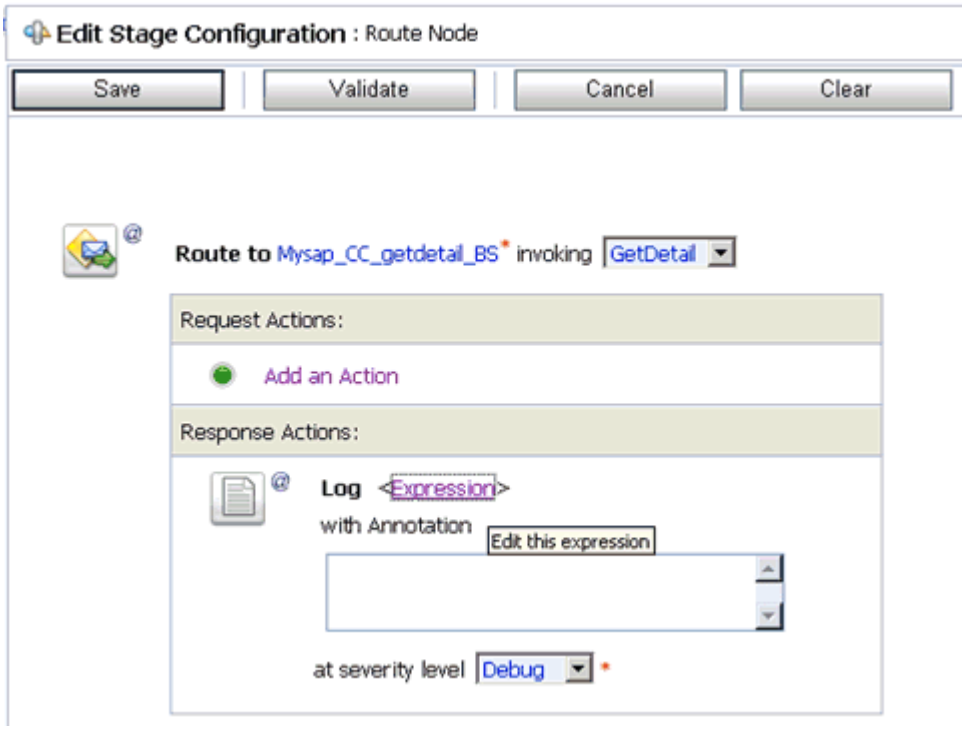
    You are returned to the Edit Stage Configuration workspace area.



23. Select **GetDetail** 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**.

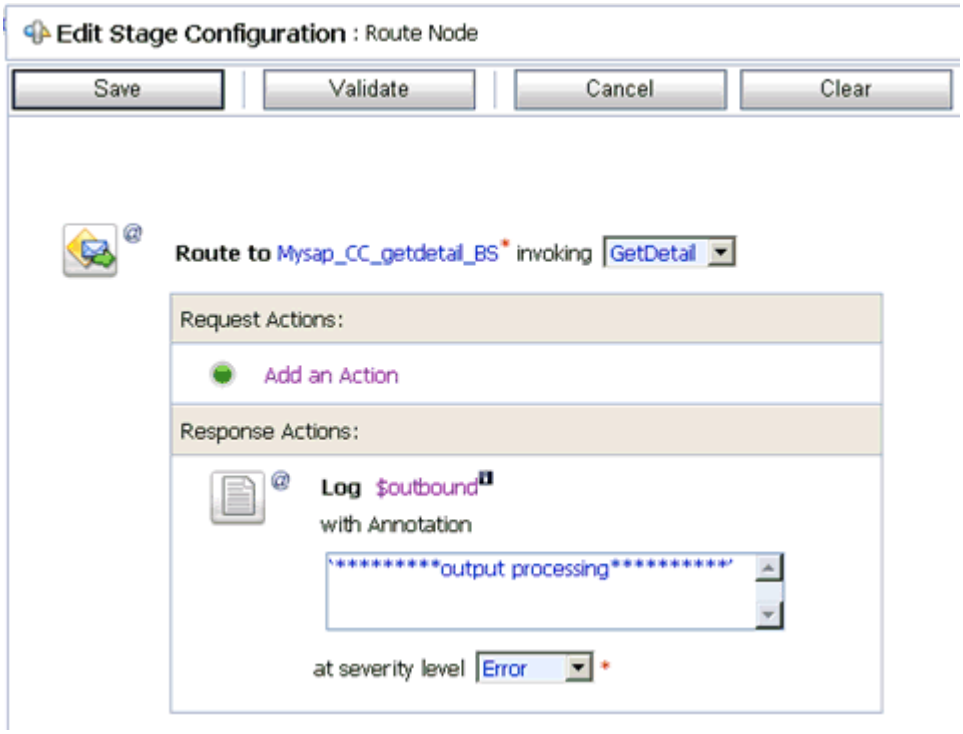


26. Click **<Expression>** to edit the expression.

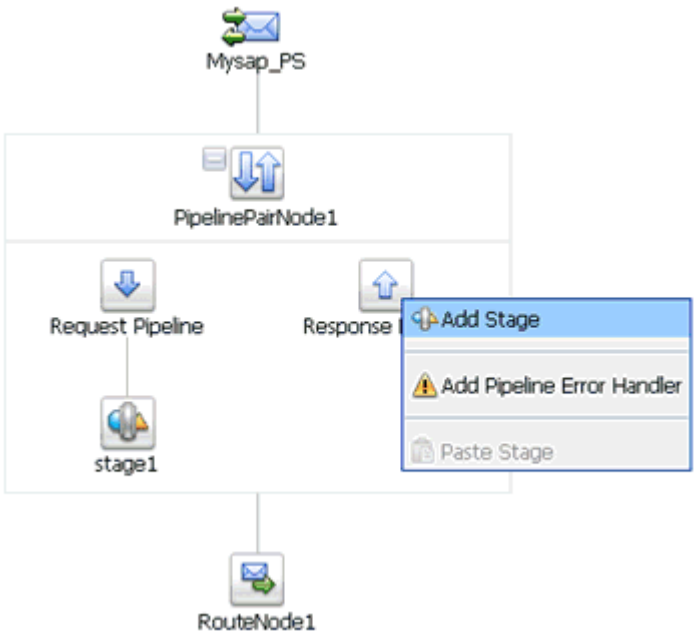
    The XQuery/XSLT Expression Editor is displayed.

27. In the XQuery Text area, type **$outbound**.
28. Click **Validate** and then **Save**.

    You are returned to the Edit Stage Configuration workspace area.

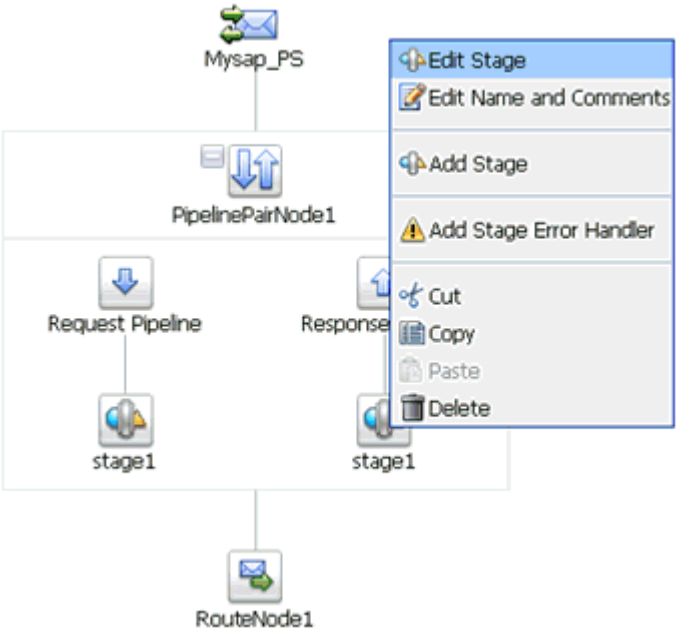


29. Type any annotation/comments in the text box (for example, **********output processing**********).
30. Select **Error** from the severity level drop-down list.
31. Click **Validate** and then **Save**.



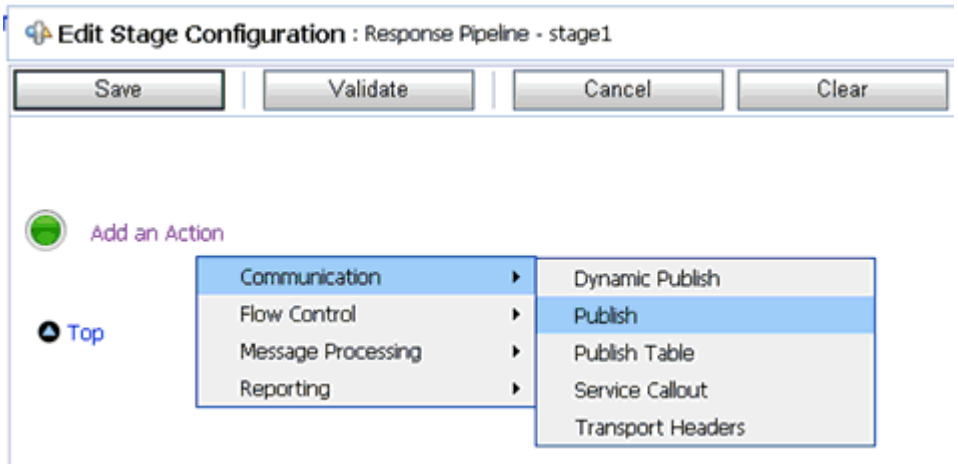
32. Click the **Response Pipeline** icon and select **Add Stage** from the context menu.

    The Stage1 icon is added below the Response Pipeline icon.

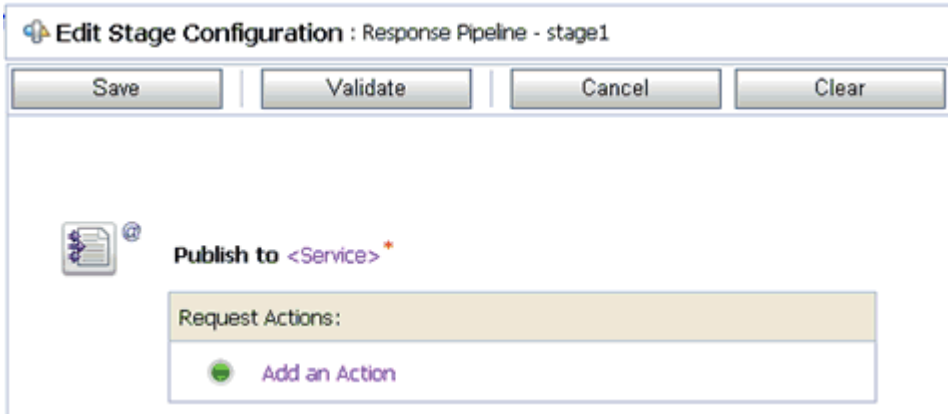


33. Click the **Stage1** icon and select **Edit Stage** from the context menu.

    The Edit Stage Configuration workspace area is displayed.

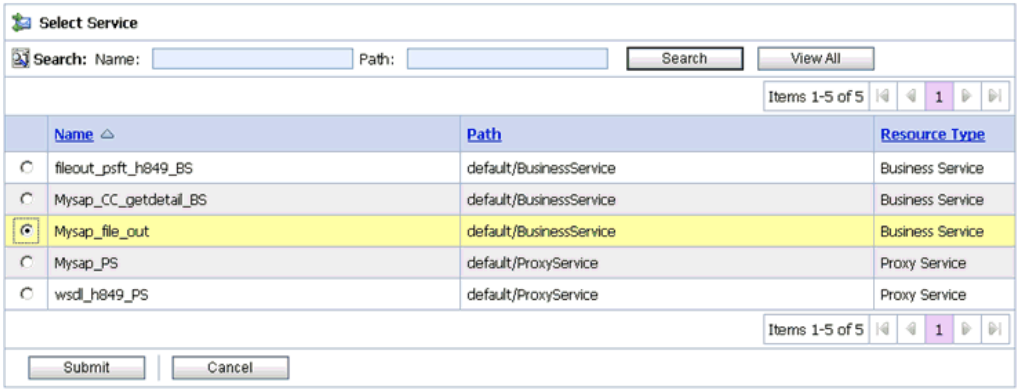


34. Click **Add an Action**, select **Communication** from the context menu, and click **Publish**.



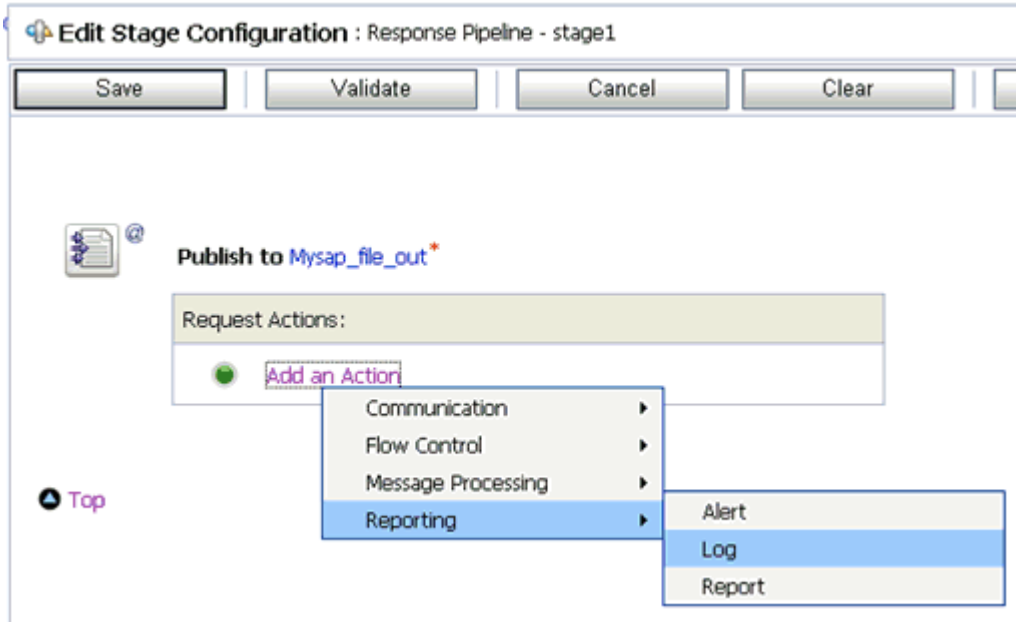
35. Click **<Service>**.

    The Select Service dialog box is displayed.

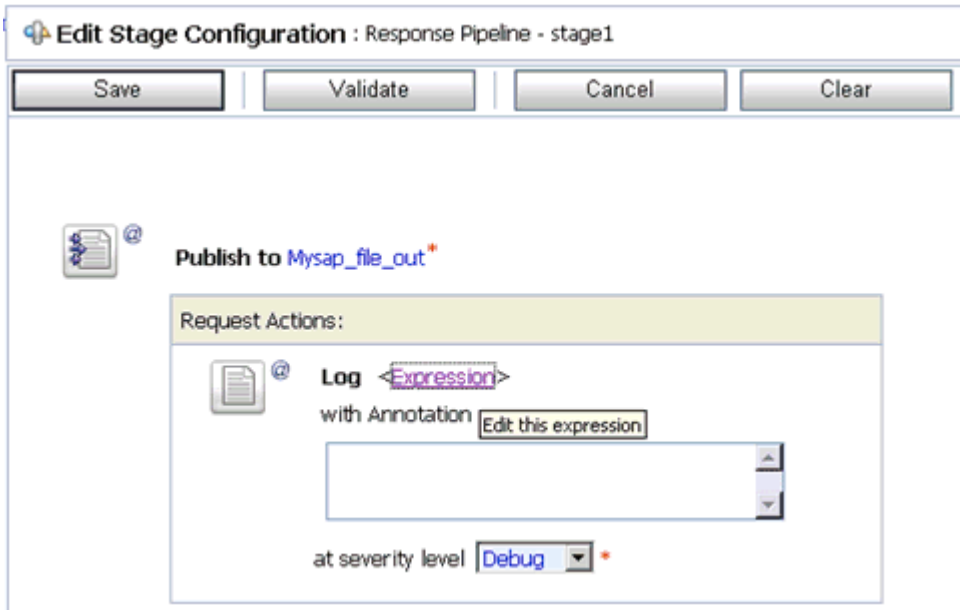


36. Select a File type Business Service and click **Submit**.

    You are returned to the Edit Stage Configuration workspace area.

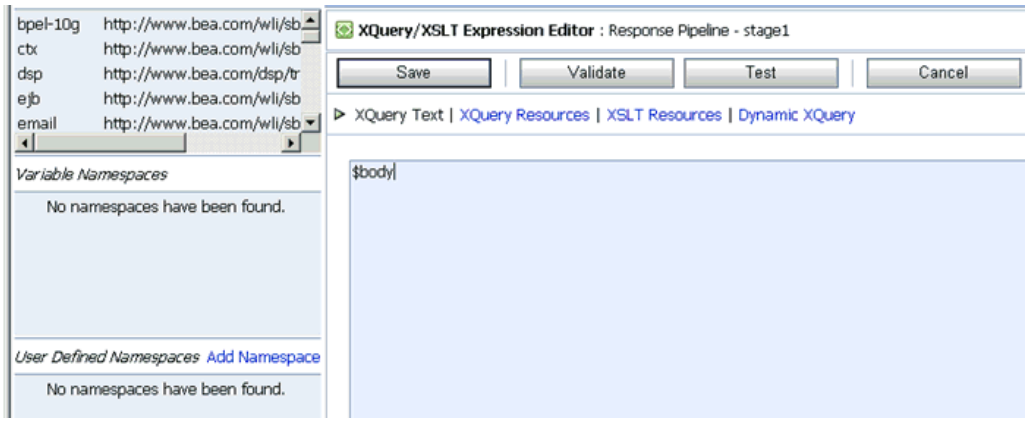


37. In the Request Actions area, click **Add an Action**, select **Reporting** from the context menu, and click **Log**.



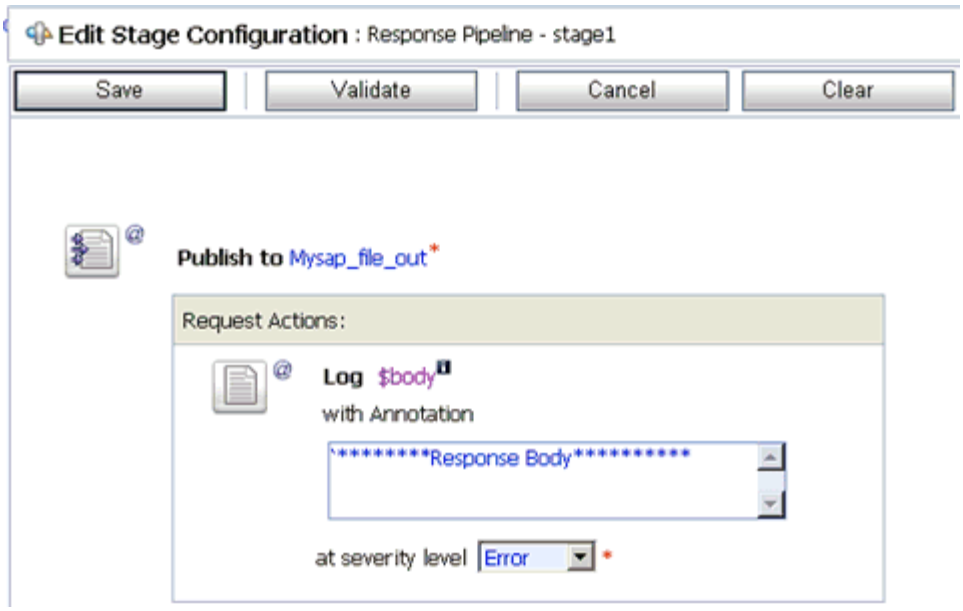
38. Click **<Expression>** to edit the expression.

    The XQuery/XSLT Expression Editor is displayed.



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

    You are returned to the Edit Stage Configuration workspace area.

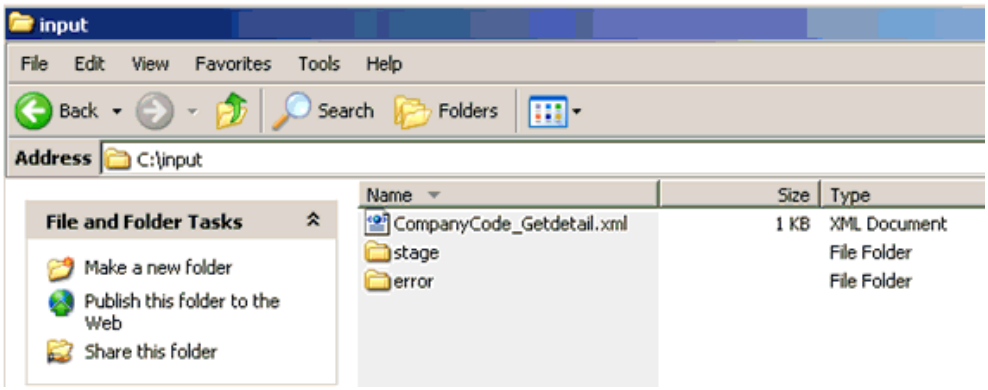


41. Type any annotation/comments in the text box (for example, **********Response Body**********).
42. Select **Error** from the severity level drop-down list.
43. Click **Validate** and then **Save**.

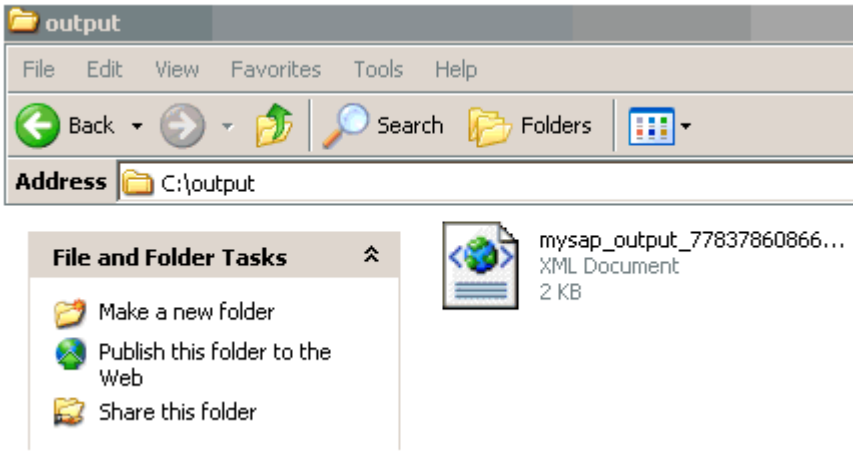
44. Click **Save**.

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.



An output XML file is received in the destination folder.


output
File Edit View Favorites Tools Help
Back
Search
Folders
Address
C:\output
File and Folder Tasks
Make a new folder
Publish this folder to the Web
Share this folder
mysap_output_77837860866...
XML Document
2 KB

7

Troubleshooting and Error Messages

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

The adapter-specific errors listed in this chapter can arise whether using the adapter with an Oracle Adapter J2CA or with an Oracle Adapter Business Services Engine (BSE) configuration.

Troubleshooting

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

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

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

- The Oracle Adapter J2CA trace information can be found under the `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
SAP R/3 does not appear in the Application Explorer Adapter node list.	Ensure that the `sapjco.jar` and `sapjcorfc.dll` files are added to the `lib` directory. Ensure the `librfc32.dll` file is added to the Windows system32 folder.
Cannot connect to Oracle Application Adapter for SAP R/3 from Application Explorer. `Problem activating adapter`	Ensure that: ■ SAP R/3 is running. ■ The Application Server name, System Number, and Client Number are correct. ■ The SAP R/3 user ID and password are correct.
Cannot connect to the SAP R/3 target through Application Explorer. The following error message appears: `Error getting target [SAP] - java.lang.Exception: Error Logon to SAP System`	Ensure that you enter the correct connection parameters when connecting to the SAP R/3 target.
Cannot connect to your SAP R/3 system through Application Explorer. The following error message appears: `Problem activating adapter. (com.ibi.sapr3.SapAdapterException : com.sap.mw.jco.JCO$Exception: (102) RFC_ERROR_COMMUNICATION: Connect to SAP gateway failed Connect_PM GWHOST=isdsrv8, GWSERV=sapgw00, ASHOST=isdsrv8, SYSNR=00 LOCATION CPIC (TCP/IP) on local host ERROR partner not reached (host isdsrv8, service 3300) TIME Fri Aug 27 11:49:14 2004 RELEASE 620 COMPONENT NI (network interface) VERSION 36 RC -10 MODULE ninti.c LINE 979 DETAIL NiPConnect2 SYSTEM CALL SO_ERROR ERRNO 10061 ERRNO TEXT WSAECONNREFUSED: Connection refused COUNTER 1). Check logs for more information`	Ensure that SAP R/3 is running and that you are using the correct parameter values to connect to your application server.
Cannot connect to your SAP R/3 system through Application Explorer even though SAP R/3 is running. The following error message appears: `Problem activating adapter. (com.ibi.sapr3.SapAdapterException : java.lang.ExceptionInInitializerError: JCO.classInitialize(): Could not load middleware layer 'com.sap.mw.jco.rfc.MiddlewareRFC' JCO.nativeInit(): Could not initialize dynamic link library sapjcorfc [no sapjcorfc in java.library.path]. java.library.path`	Ensure that the `sapjcorfc.dll` file is added to the lib directory and the `librfc32.dll` file is added to the Windows system32 folder.

Error	Solution

The dll is loaded in another classloader (BSE and J2CA are installed on the same server). The following error message appears:
`com.ibi.sapr3.SapAdapterException: java.lang.ExceptionInInitializerError: JCO.classInitialize(): Could not load middleware layer 'com.sap.mw.jco.rfc.MiddlewareRFC'`
`JCO.nativeInit(): Could not initialize dynamic link library sapjcorfc [Native Library F:\iWay55.008.0628\lib\sapjcorfc.dll already loaded in another classloader]. java.library.path`
	Add `sapjco.jar` to the server classpath.

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.
Logon failure error at run-time	If the password for connecting to your SAP R/3 system is not specified when creating a target or with the Edit option in Application Explorer, you will be unable to connect to SAP R/3. 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.bat`): `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`

SAP R/3

Error	Solution
When executing a request, the following error message appears: `AdapterException: java.lang.Exception: Function module CUSTOMER_GETDETAIL2 does NOT exist.`	Check the syntax of your input XML document and make sure the name of the Remote Function module is correct, available in SAP R/3, and activated.
When executing a request, the following error message appears: `AdapterException: java.lang.Exception: Object type unknown for business object: CUST`	Check the syntax of your input XML document and verify that the Business Object type exists in SAP R/3 and is correct and activated. Also verify that the name of your document is correct.
When executing a request, the following error message appears: `AdapterException: java.lang.Exception: Unable to retrieve BAPI name for: CUSTOMER.DETAIL2`	Check the syntax of your input XML document and verify that the name of the BAPI is correct, available in SAP R/3, and activated.
When executing a request, the following error message appears: `java.lang.RuntimeException: com.sap.mw.jco.JCO$AbapException: (126) OBJECT_UNKNOWN: Basic type or extension does not exist.`	Check the syntax of your input XML document and verify that the IDoc type or extension type is correct, available in SAP R/3, and activated.
When executing a request, the following error message appears: `AdapterException: java.lang.Exception: BapiError/BapiAbort: You are not authorized to display customers.`	Verify that your user ID has the correct permissions configured in SAP R/3. Consult your SAP R/3 administrator for more information.

Note: Activation is an important SAP concept. If an object does not exist in an activated state, it may appear as present on the system, but is not available for use. This is especially important for IDOCs and SAP Business Objects. Consult your SAP documentation for further information.

Oracle Adapter J2CA

Error	Solution
In Application Explorer, the following error message appears when you attempt to connect to an Oracle Adapter J2CA configuration: `Could not initialize JCA`	In the Details tab in the right pane, ensure that the directory specified in the Home field points to the correct directory, for example, *`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 if 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.

Oracle Application Adapter for SAP R/3 Invalid SOAP Request

If Oracle Application Adapter for SAP R/3 receives a SOAP request message that does not conform to the WSDL for the Web services being carried out, then the following SOAP response is generated.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Body>
<SOAP-ENV:Fault>
  <faultcode>SOAP-ENV:Server</faultcode>
  <faultstring>Error processing agent [XDSapIfrAgent] - XD[FAIL] SapIFRException:
java.sql.SQLException:
```

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

Empty Result From SOAP Request

If Oracle Application Adapter for SAP R/3 carries out an SAP R/3 object using input parameters passed in the SOAP request message that do not match records in SAP R/3, then the following SOAP response is generated.

```
<?xml version="
1.0" encoding="ISO-8859-1" ?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Body>
 <SOAP-ENV:Fault>
  <faultcode>SOAP-ENV:Server</faultcode>
  <faultstring>Error processing agent [XDSapIfrAgent] - XD[FAIL] SapIFRException:
java.sql.SQLException: com.ibi.sapjco.SapCallableStatement: execute()
java.sql.SQLException: JCO Error Key: NO_RECORD_FOUND Short Description:
com.sap.mw.jco.JCO$AbapException: (126) NO_RECORD_FOUND: NO_RECORD_
FOUND</faultstring>
   </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
  </SOAP-ENV:Envelope>
```

Failure to Connect to SAP R/3

If Oracle Application Adapter for SAP R/3 cannot connect to SAP R/3 when executing a Web service, then the following SOAP response is generated:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Body>
 <SOAP-ENV:Fault>
  <faultcode>SOAP-ENV:Server</faultcode>
  <faultstring>Error processing agent [XDSapIfrAgent] - XD[RETRY]
Connect to SAP gateway failed Connect_PM GWHOST=ESDSUN, GWSERV=sapgw00,
ASHOST=ESDSUN,
SYSNR=00 LOCATION CPIC (TCP/IP) on local host ERROR partner not reached (host
ESDSUN, service 3300)
TIME Mon Jun 30 16:01:02 2003 RELEASE 620 COMPONENT NI (network interface) VERSION
36 RC -10 MODULE ninti.c LINE 976 DETAIL NiPConnect2
SYSTEM CALL SO_ERROR ERRNO 10061 ERRNO TEXT WSAECONNREFUSED: Connection refused
COUNTER 1</faultstring>
  </SOAP-ENV:Fault>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Invalid SOAP Request

If Oracle Application Adapter for SAP R/3 receives a SOAP request message that does not conform to the WSDL for the Web services being carried out, then the following SOAP response is generated.

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

```
  <faultstring>RPC server connection failed: Connection refused:
connect</faultstring>
 </SOAP-ENV:Fault>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Empty Result From Oracle Application Adapter for SAP R/3 Request

If Oracle Application Adapter for SAP R/3 carries out a SOAP request using input parameters passed that do not match records in the target system, then the following SOAP response is generated.

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

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

8

Advanced User Tools

This chapter includes the following topics:

Web Services Policy-Based Security

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

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

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

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

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

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

You can assign a policy to a Business Service, or to a method within a Business Service. If a policy is only applied to a method, other methods in that Business Service 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 BSE in the SOAP request message are verified against the list of users for all policies applied to that specific Business Service. The policy type that is supported is Resource Execution, which dictates who can or cannot perform the Business Service.

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

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

Configuring Web Services Policy-Based Security

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

Creating and Associating a User with a Policy

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

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

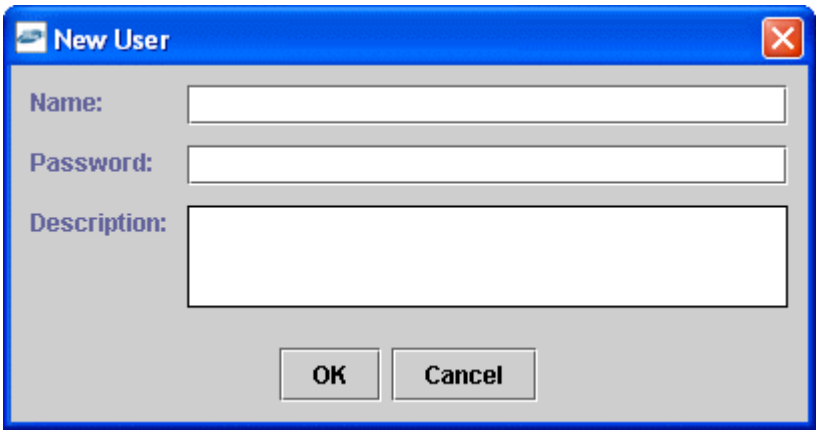
   Nodes appear for Adapters, Events, and Business Services (also known as Web services).

   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.



Provide the following information:

a. In the **Name** field, enter a user ID.

b. In the **Password** field, enter the password associated with the user ID.

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

5. Click **OK**.

The new user is added under the Users node.

Creating a Group to Use With a Policy

To create a group to use with a policy:

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

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

Perform the following steps:

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

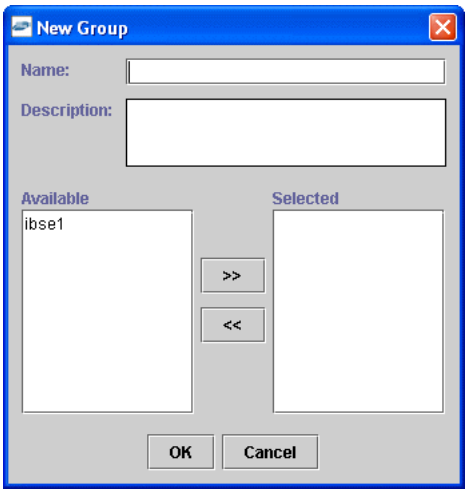
b. Expand the **Configuration** node.

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.



Provide the following information:

a. In the **Name** field, enter a name for the group.

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

c. From the available list of users in the left pane, select one or more users and add them to the **Selected** list by clicking the double right-facing arrow.

5. When you have selected at least one user, click **OK**.

The new group is added under the Group node.

Creating an Execution Policy

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

To create an execution policy:

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

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

Perform the following steps:

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

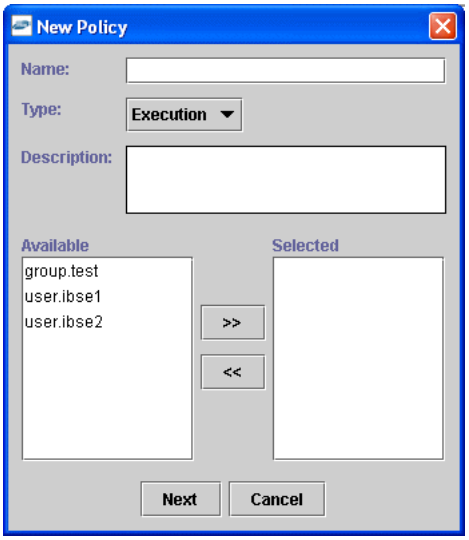
b. Expand the **Configuration** node.

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.



Provide the following information:

a. In the **Name** field, enter a name for the policy.

b. From the **Type** list, select **Execution**.

c. In the **Description** field, enter a description for the policy (optional).

d. From the available list of users in the left pane, select one or more users and add them to the **Selected** list by clicking the double right-facing arrow.

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

---

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

   The New Policy permissions dialog box is displayed.

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

7. Click **OK**.

   The following pane summarizes your configuration.

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

Using the IP and Domain Restrictions Policy Type

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

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

3. Select **Connect**.

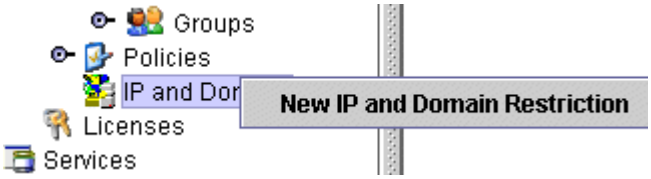
   Nodes appear for Adapters, Events, and Business Services (also known as Web services).

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

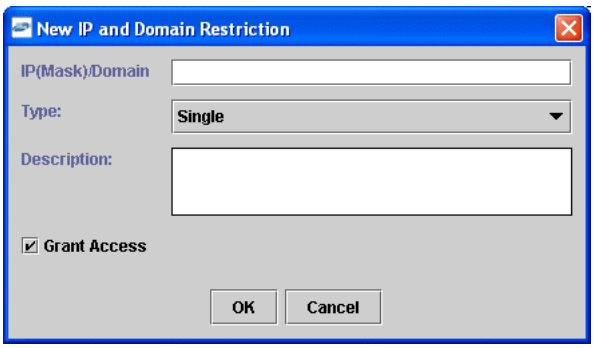
   b. Expand the **Configuration** node.

   c. Expand the **Security** node.

4. Right-click **IP and Domain** and select **New IP and Domain Restriction**.



   The New IP and Domain Restriction dialog box is displayed.



   Provide the following information:

   a. In the **IP(Mask)/Domain** field, enter the IP or domain name using the following guidelines.

      If you select **Single** (Computer) from the **Type** list, you must provide the IP address for that computer. If you only know the DNS name for the computer, click **DNS Lookup** to obtain the IP Address based on the DNS name.

      If you select **Group** (of Computers), you must provide the IP address and subnet mask for the computer group.

      If you select **Domain**, you must provide the domain name.

   b. From the **Type** list, select the type of restriction.

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

   d. To grant access, select the **Grant Access** check box.

5. Click **OK**.

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

The following pane summarizes your configuration.

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

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

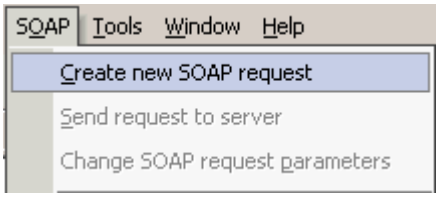
To migrate a BSE repository:

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

   ```
   http://localhost:7777/ibse/IBSEServlet/admin/iwcontrol.ibs
   ```

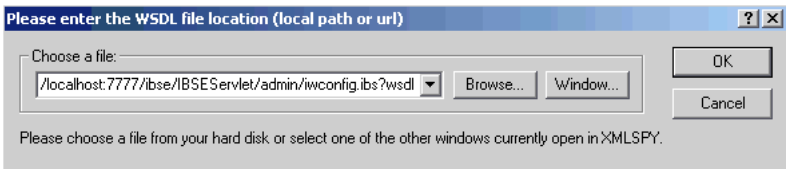
2. Open a third-party XML editor, for example, XMLSPY.
3. From the menu bar, click **SOAP**.

   A list of options appears.



4. Select **Create new SOAP request**.

   The WSDL file location dialog box is displayed.



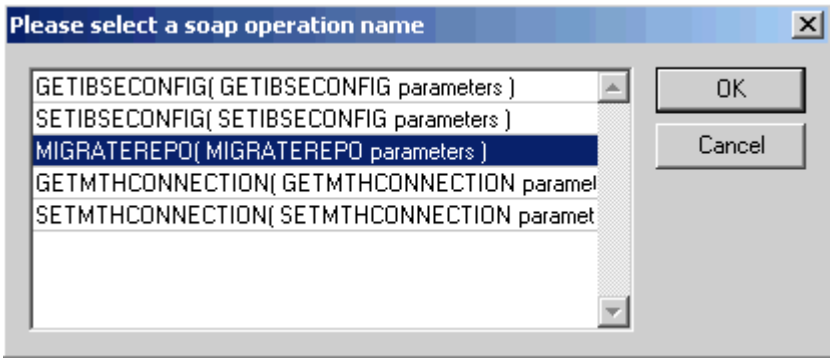
   Perform the following steps:

   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**.

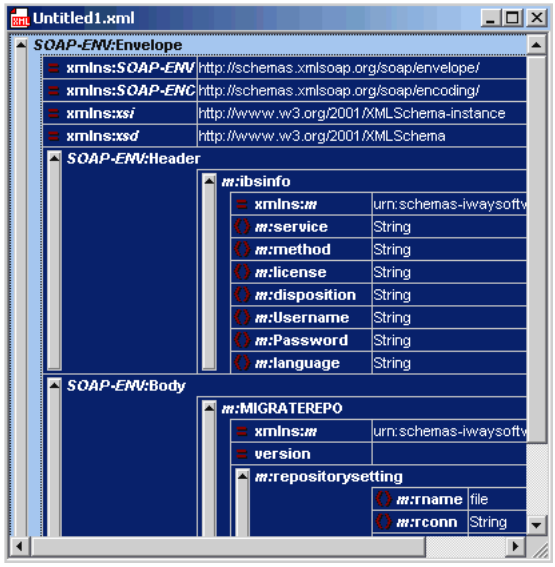
The soap operation name dialog box is displayed and the available control methods are listed.



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

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

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



7. Locate the **Text view** icon in the toolbar.

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

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

9. Locate the following section:

```
<m:MIGRATEREPO xmlns:m="urn:schemas-iwaysoftware-com:jul2003:ibse:config"
version="">
  <m:repositorysetting>
    <m:rname>oracle</m:rname>
    <m:rconn>String</m:rconn>
    <m:rdriver>String</m:rdriver>
    <m:ruser>String</m:ruser>
    <m:rpwd>String</m:rpwd>
  </m:repositorysetting>
  <m:servicename>String</m:servicename>
</m:MIGRATEREPO>
```

   Perform the following steps:

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

      The Oracle repository URL has the following format:

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

   b. For the `<m:rdriver>` tag, replace the String placeholder with the location of your Oracle driver.

   c. For the `<m:ruser>` tag, replace the String placeholder with a valid user name to access the Oracle repository.

   d. For the `<m:rpwd>` tag, replace the String placeholder with a valid password to access the Oracle repository.

10. 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:

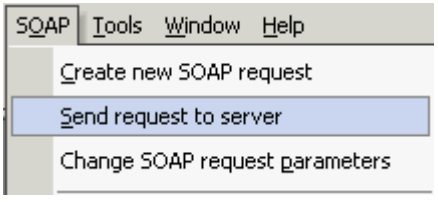
     ```
     <m:servicename>SAPService1</m:servicename>
     ```

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

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

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

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



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

Migrating a J2CA Repository

To migrate a J2CA repository:

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

   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 SAP R/3 for Inbound and Outbound Processing

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

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

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

Configuring SAP R/3 Inbound Processing

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

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

You must perform the following steps to configure SAP R/3 for inbound IDoc processing:

1. Configure a logical system.
2. Configure a distribution model.
3. Define an inbound partner profile.

Configuring a Logical System

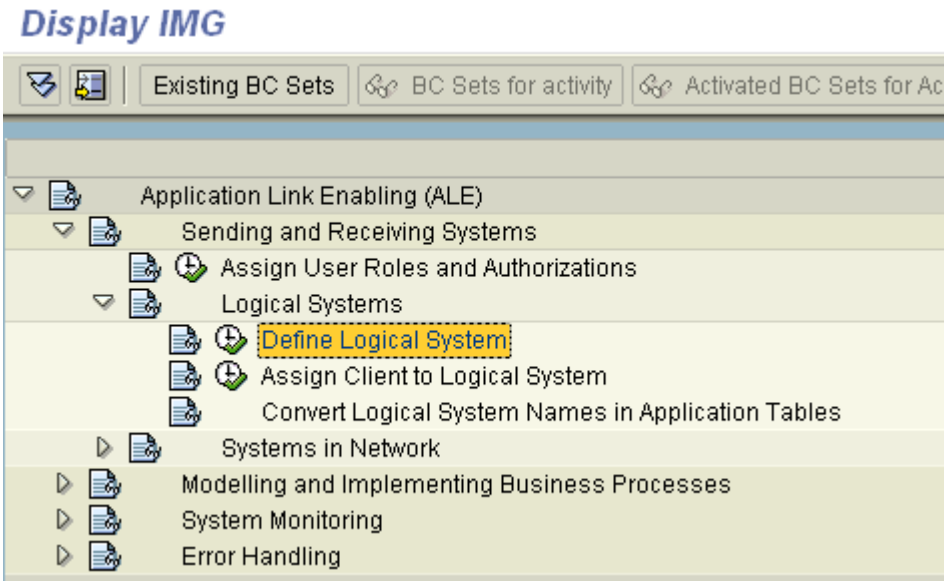
In any distributed environment, each participating system must have a unique ID to avoid confusion. In SAP R/3, the name of the logical system is used as the unique ID. This name is assigned explicitly to one client in a SAP R/3 system.

Defining a Logical System

To define a logical system:

1. Run the **sale** transaction.

The Display IMG window is displayed.



Perform the following steps:

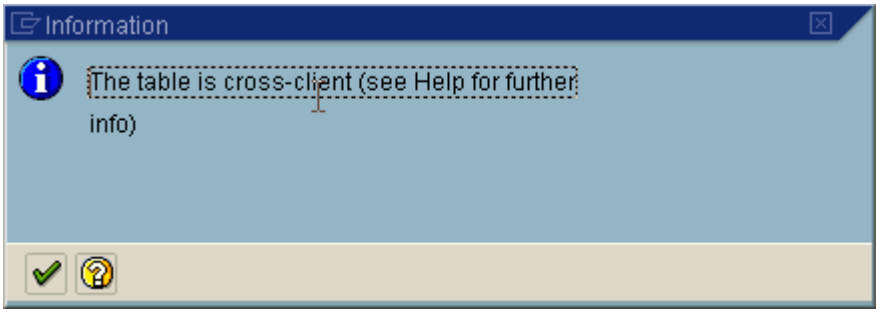
a. Expand **Sending and Receiving Systems**.

b. Expand **Logical Systems**.

c. Select **Define Logical System**.

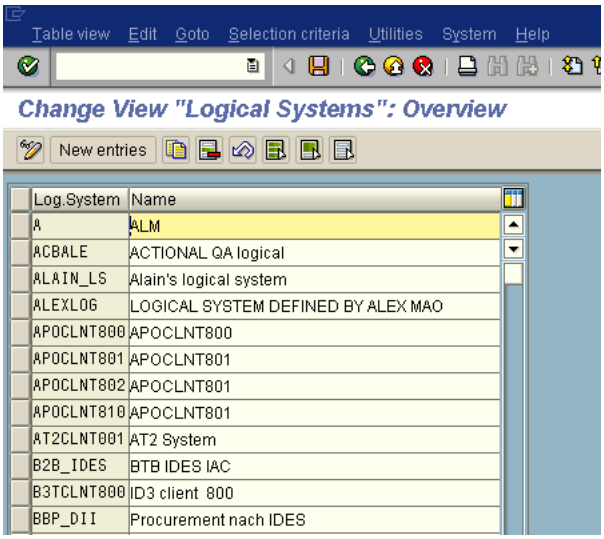
2. Click the **IMG - Activity** icon.

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



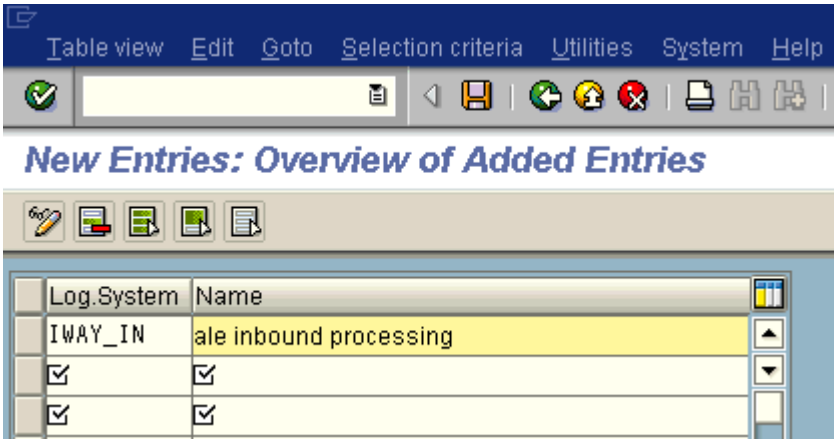
3. Click the **check mark** icon to continue.

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



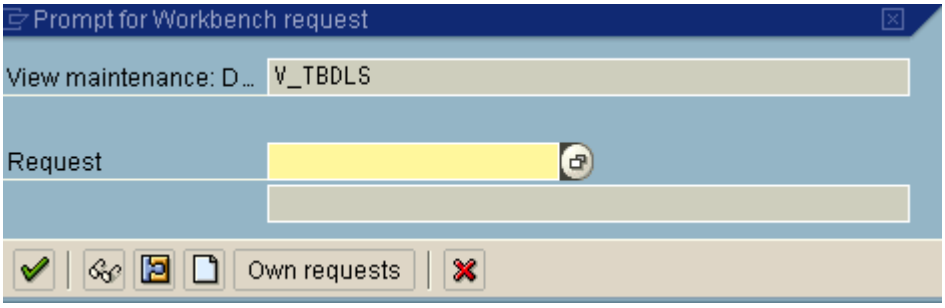
4. Click **New Entries**.

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



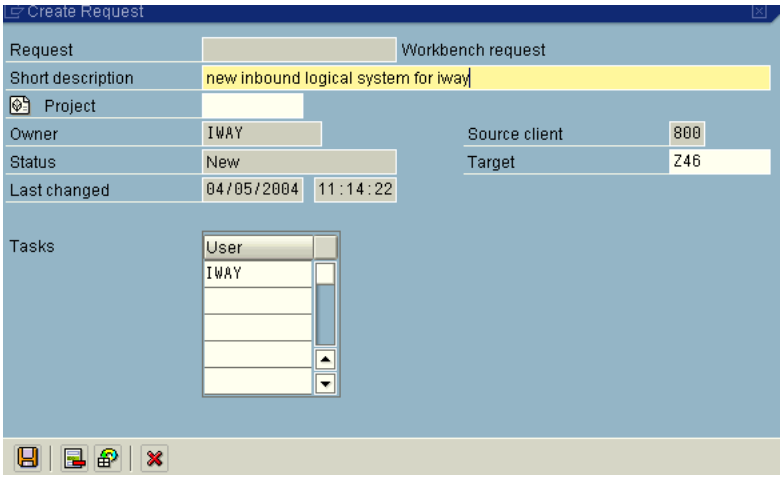
5. Enter the Logical System, for example, ORACLETDS, in the **Log.System** column and provide a description in the **Name** column.
6. Click **Save**.

   The Prompt for Workbench request dialog box is displayed.



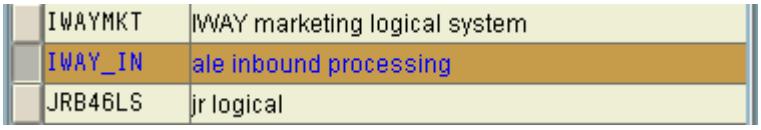
7. Click the **Create Request** icon.

   The Create Request dialog box is displayed.



8. Enter a name and description for your request and click **Save**.

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



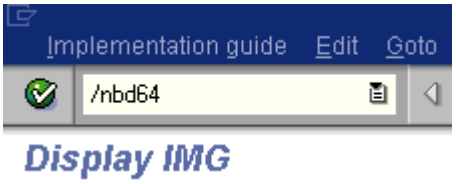
Configuring a Distribution Model

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

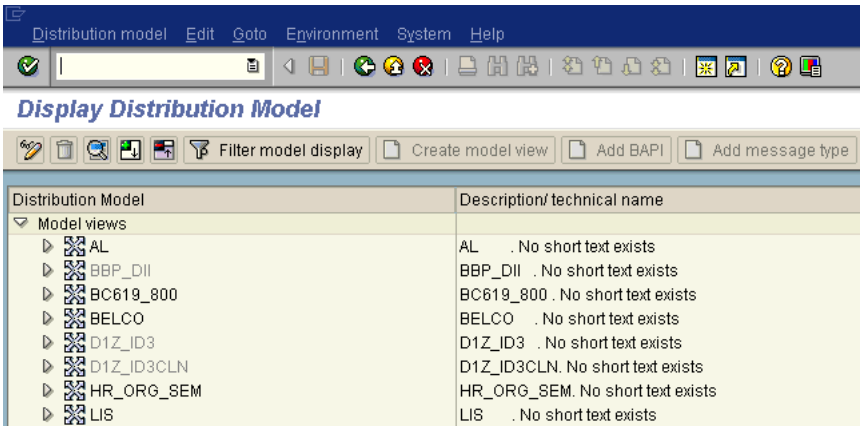
Defining a Distribution Model

To define a distribution model:

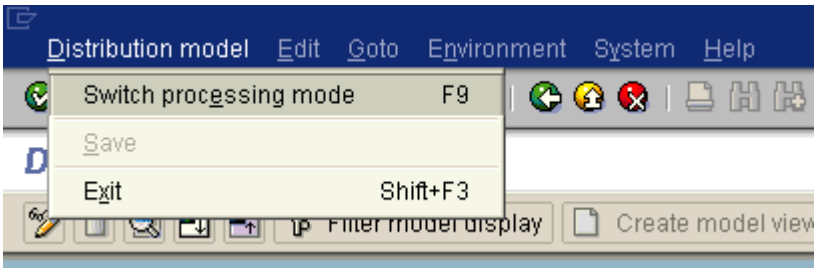
1. Run the **bd64** transaction.



   The Display Distribution Model window is displayed.

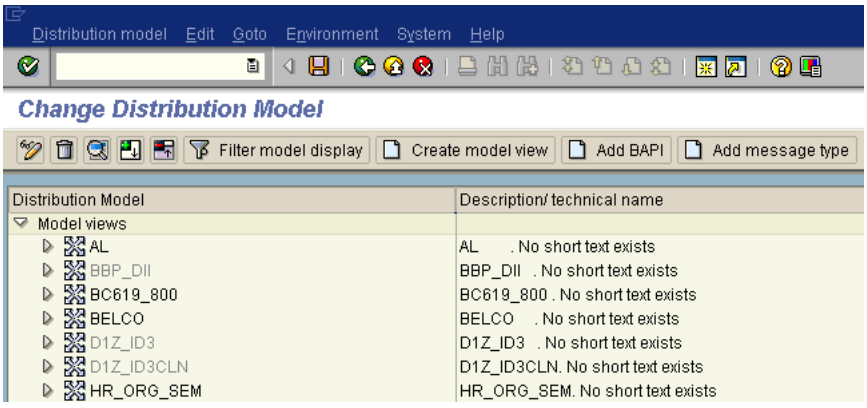


2. Click **Distribution Model** from the menu bar.



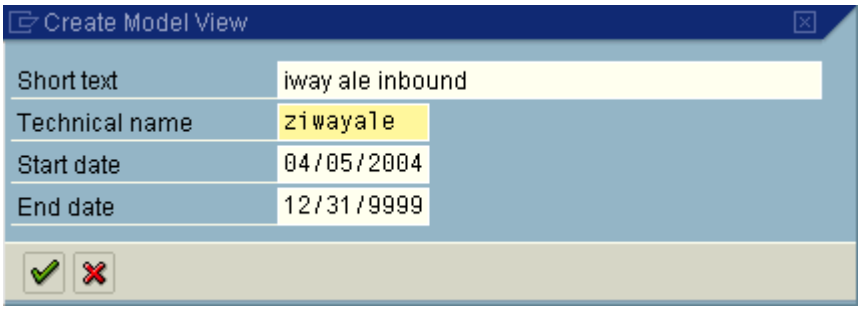
3. Select **Switch processing mode**.

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



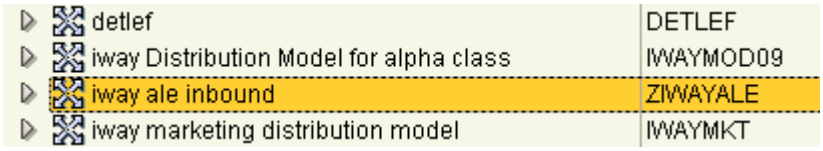
4. Click **Create model view**.

   The Create Model View dialog box is displayed.



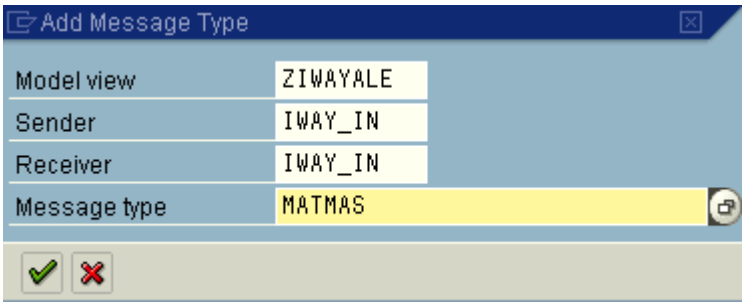
5. Enter a model view name in the **Short text** field and a name in the **Technical name** field, which also serves as a description.
6. Click the **check mark** icon to enter the information.

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



7. Click **Add message type**.

   The Add Message Type dialog box is displayed.



   Perform the following steps:

   a. In the **Sender** and **Receiver** fields, enter the logical system you configured, for example, ORACLETDS.

      You can click the icon to the right of each field to browse from a list of logical systems.

   b. In the **Message type** field, enter the message type you want to use, for example, MATMAS.

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

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

   You are returned to the main Change Distribution Model window.

9. Click **Save**.

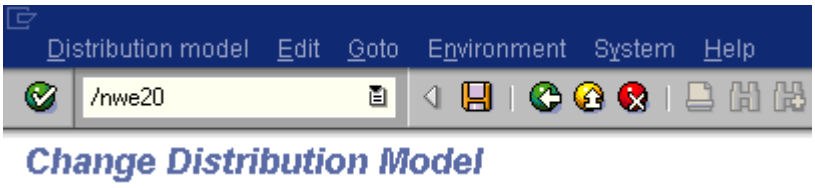
Defining a Partner Profile

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

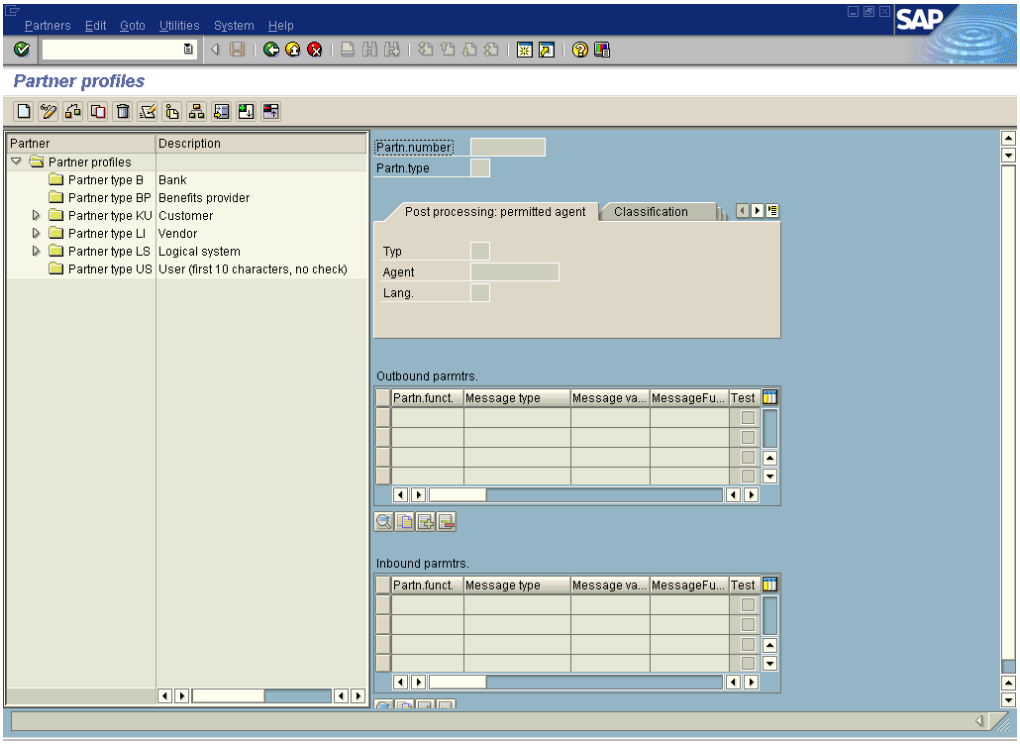
Defining a Partner Profile

To define a partner profile:

1. Run the **we20** transaction.

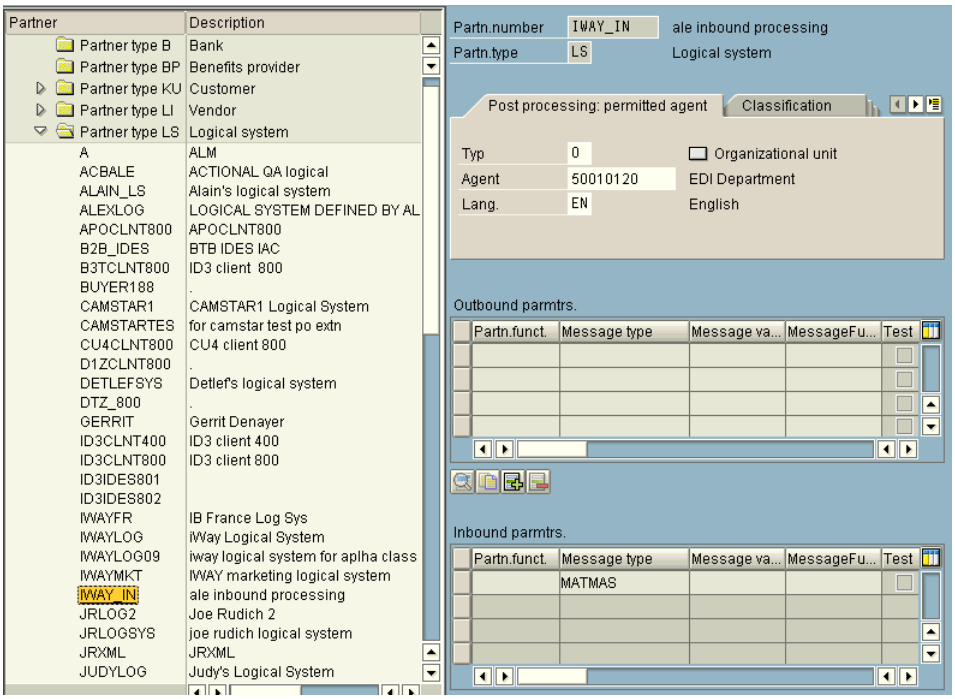


   The Partner profiles window is displayed.



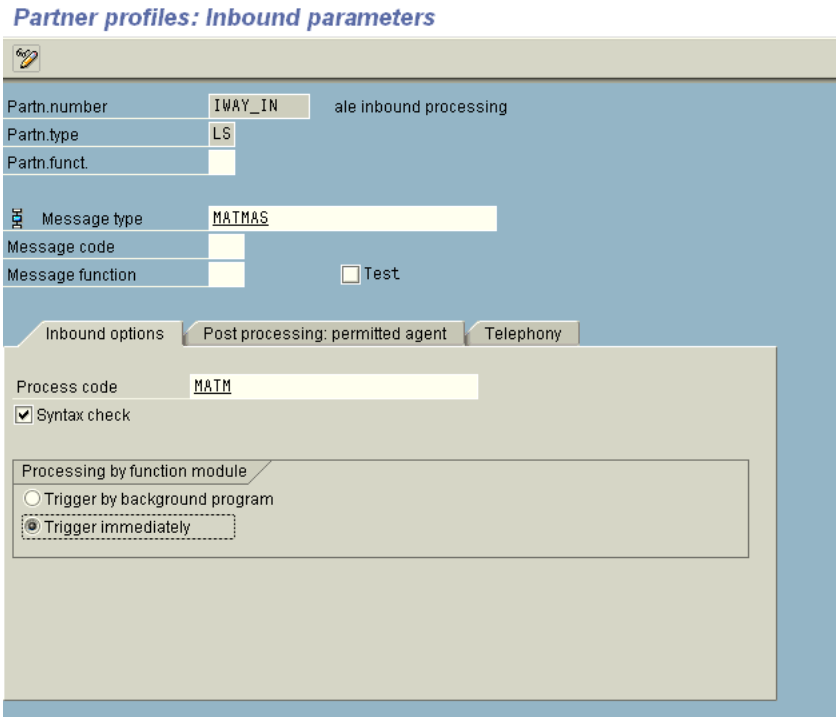
2. In the left pane, expand **Partner type LS** and select the logical system you configured from the list, for example, ORACLETDS.

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



3. Click **Save**.
4. From the **Inbound** parameters table, click the **Create inbound parameter** icon.

   The Partner profiles: Inbound parameters window is displayed.



5. In the **Message type** field, enter the message type you want to use, for example, MATMAS.

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

The Inbound options tab is selected by default.

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

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

7. In the **Processing by function module** area, select one of the following options:

   - Trigger by background program.

     In this case the adapter writes IDocs to the SAP R/3 database, which is processed immediately.

   - Trigger immediately.

     In this case, the adapter waits for the SAP R/3 system to process IDocs. This can take anywhere from 1 to 15 minutes.

8. Click **Save**.

Configuring SAP R/3 Outbound Processing

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

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

Related Concepts and Terminology

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

Client and Server Programs

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

- RFC Client
- RFC Server

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

SAP R/3 Gateway

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

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

In SAPGUI, the destination must be defined with transaction SM59, using connection type T and Register Mode. Moreover, this entry must contain information on the SAP R/3 gateway at which the RFC server program is registered.

Program IDs and Load Balancing

If the Gateway Server has a connection to a particular server instance and another server instance presents itself to the gateway, then the gateway offers the connection and then begins functioning in Load Balancing mode. Using a proprietary algorithm, the Gateway sends different messages to each server depending on demand and total processing time. This may cause unpredictable results when messages are validated by schema and application.

When configuring multiple events in the Oracle WebLogic Server using a single SAP R/3 program ID, SAP R/3 load balances the event data. For example, if multiple remote function calls or BAPIs use the same program ID (for example, ORACLETDS) and multiple SAP R/3 listeners are configured with this progamID, then SAP R/3 sends one request to one listener and the next to another listener, and so on.

There is a load-balancing algorithm present in the SAP R/3 Gateway Server. This mechanism is proprietary to SAP R/3 application development and might work by comparing total throughput of the connection, the number of times in wait state, and so on. One connection might receive nine messages and a second connection might receive one message. If five of the nine messages are rejected for schema validation and the one message on the other connection is rejected for schema validation, you might suspect that you are missing SAP R/3 event handling messages.

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

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

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

Connection Pooling

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

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

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

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

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

Registering Your Program ID in SAPGUI

To enable your SAP R/3 system to issue the following calls or interfaces to the SAP R/3 event adapter, you must register your program ID under an RFC destination.

- Remote Function Calls (RFC)
- Business Application Programming Interfaces (BAPI)
- Intermediate Documents (IDoc)

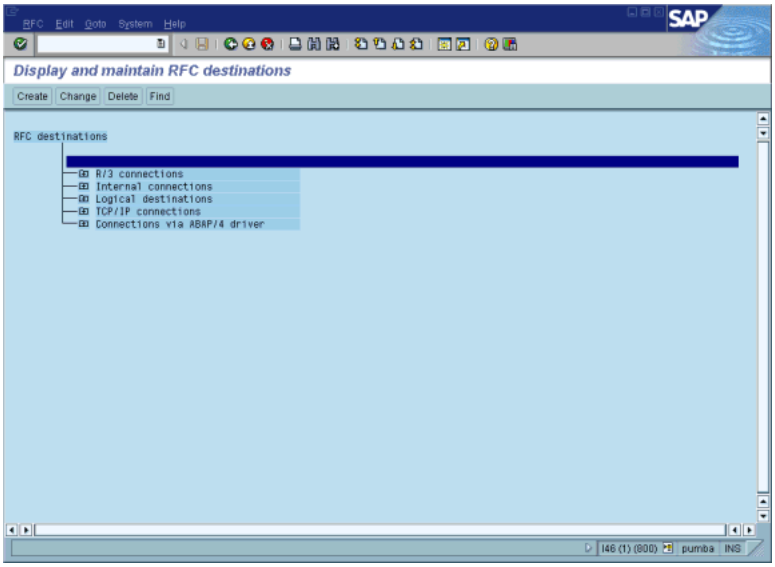
The RFC destination is a symbolic name (for example, ORACLETDS) that is used to direct events to a target system, masking the program ID. The Program ID is configured in both SAPGUI and the event adapter.

Registering Your Program ID

To register your program ID:

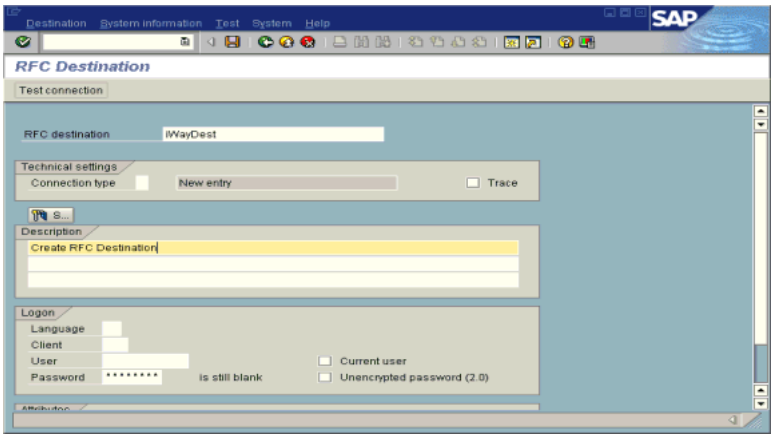
1. Launch the SAP GUI and log in to the SAP R/3 system.
2. Select **Tools**, **Administration**, **Network**, and then **RFC destination**.
3. Run the **SM59** transaction.

   The Display and maintain RFC destinations window is displayed.



4. Select **TCP/IP connections** and click **Create**.

   The RFC Destination window is displayed.



Provide the following information:

a. In the **RFC destination** field, enter a name, for example, **ORACLETDS**.

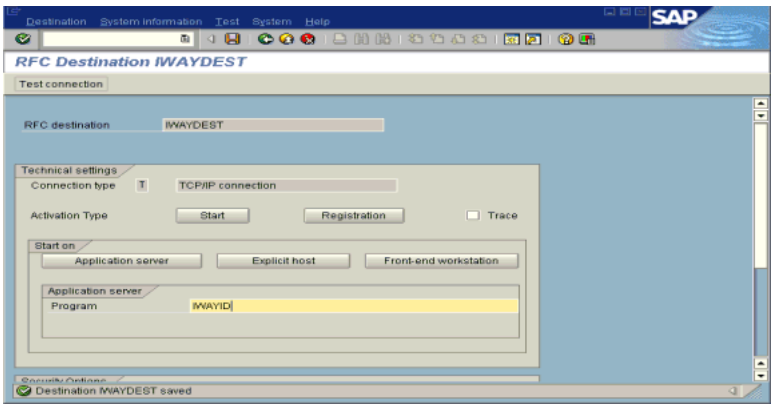
   The value you enter in this field is case sensitive.

b. In the **Connection type** field, enter **T** for destination type TCP/IP.

c. In the **Description** field, enter a brief description.

5. Click **Save** from the tool bar or select **Save** from the Destination menu.

   The RFC Destination ORACLETDS window is displayed.



Perform the following steps:

a. For the **Activation Type**, click **Registration**.

b. In the **Program** field, enter **ORACLETDS**.

6. Click **Save** from the tool bar or select **Save** from the Destination menu.
7. Ensure your event adapter is running.
8. Verify that the SAP R/3 system and Oracle Application Adapter for SAP R/3 are communicating.
9. Click **TestConnection**.

Testing the SAP R/3 Event Adapter

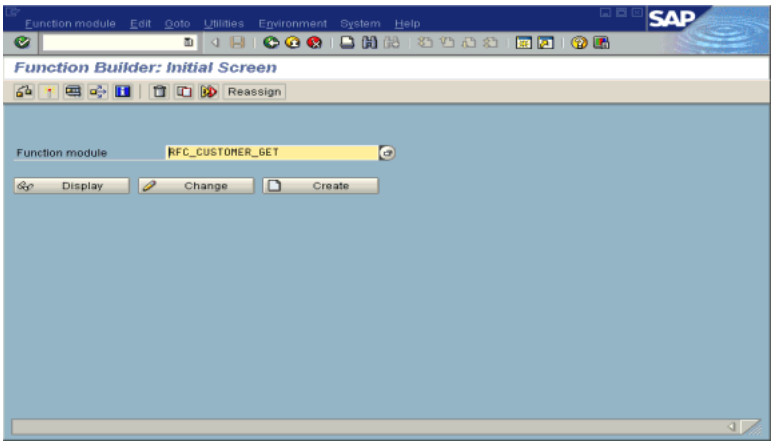
In the SAP Server, the SE37 transaction enables you to send an RFC (Remote Function Call) or a BAPI (Business Application Programming Interface) to any RFC destination.

For more information on RFC destination, see Registering Your Program ID in SAPGUI on page A-11.

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

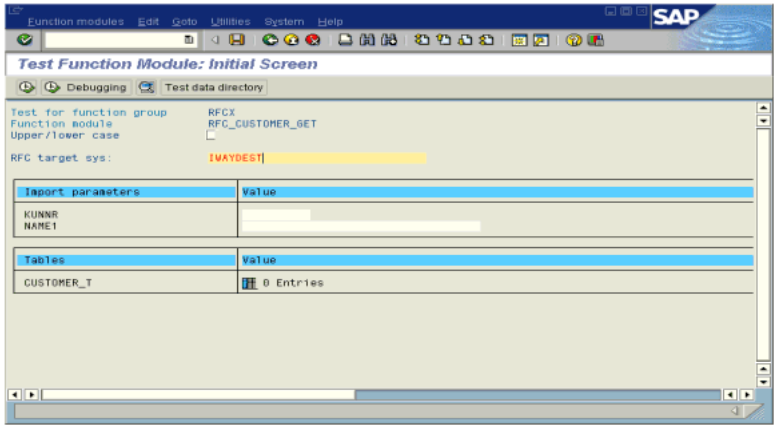
To test the SAP R/3 event adapter:

1. In the Function Builder, select a function module, for example, `RFC_CUSTOMER_GET`.



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

   The Test Function Module: Initial Screen window is displayed.



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

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

Application Link Embedding Configuration for the Event Adapter

The SAP R/3 event adapter receives IDocs (Intermediate Documents) from SAP R/3. To configure an SAP R/3 system to send IDocs to the SAP R/3 event adapter, use the ALE (Application Link Embedding) configuration to:

1. Register your program ID in SAP GUI.
2. Define a port.
3. Create a logical system.
4. Create a partner profile.
5. Create a distribution model for the partner and message type.
6. Test the SAP R/3 event adapter.

Defining a Port

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

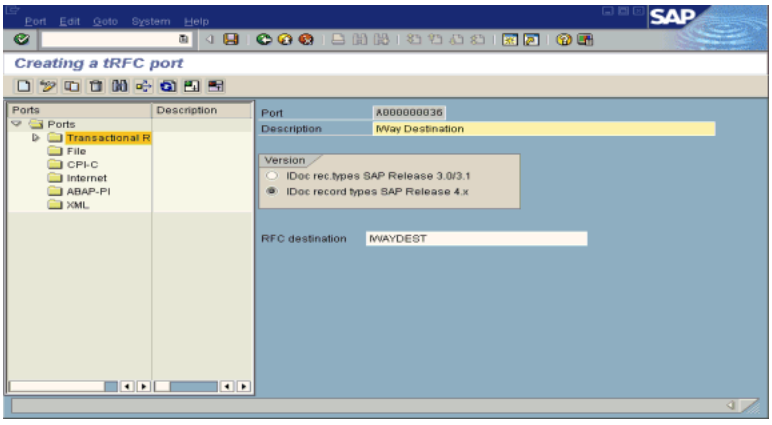
Defining a Port

To define a port:

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

   You can also run the WE21 transaction.

   The Creating a tRFC port window is displayed.



2. In the left pane under **Ports**, select **Transactional RFC** and click **Create**.
3. Select **Generate port name**.

   The system generates the port name.
4. Enter the IDoc version you want to send through this port.
5. Click the destination you created, for example, ORACLETDS.
6. Save the session, making note of the system-generated RFC port.

Creating a Logical System

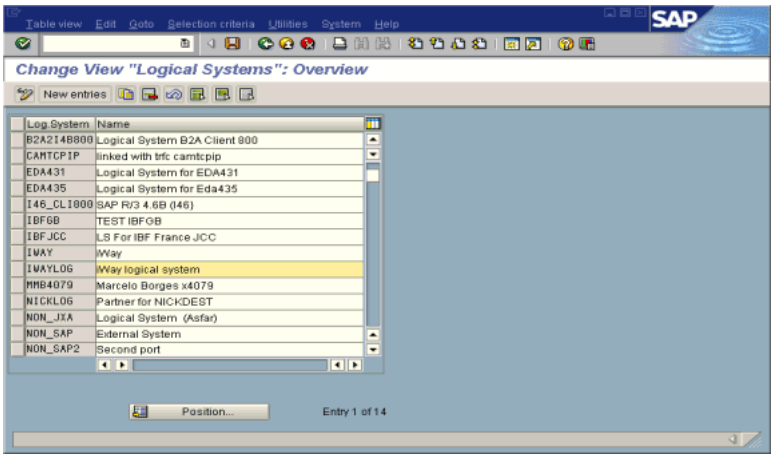
One type of partner is a logical system. A logical system manages one or more RFC destinations.

Creating a Logical System

To create a logical system called ORACLETDS:

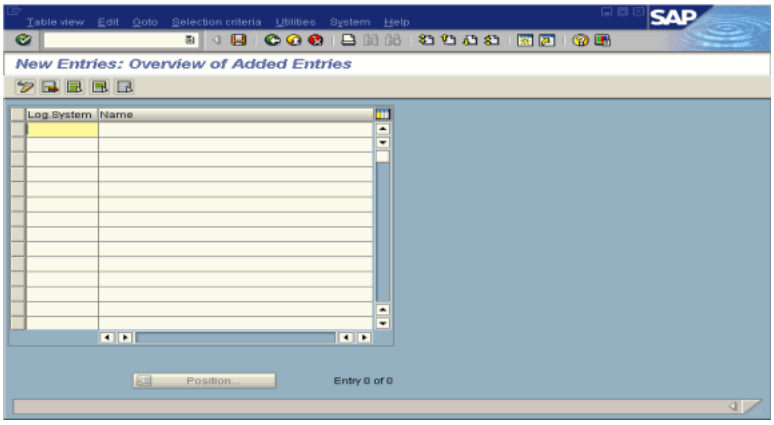
1. In the ALE configuration, enter the area menu selection **SALE** transaction.
2. Select **SAP Reference IMG**.
3. Expand the following nodes: **Basis Components**, **Application Link Enabling (ALE)**, **Sending and Receiving Systems**, **Logical Systems**, and **Define Logical System**.
4. Click the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

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



5. Click **New entries**.

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



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

Creating a Partner Profile

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

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

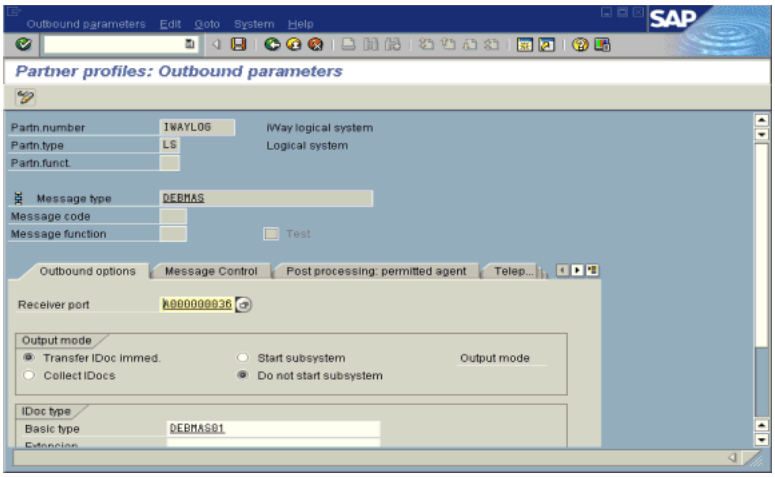
Creating a Partner Profile

To create a partner profile:

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

   You can also run the WE21 transaction.

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



   Perform the following steps:

   a. Select Partner type **LS** (Logical system).

   b. Press **F5** (Create).

2. For Type, enter **USER**.

3. For Agent, enter the current user ID, or you may select another agent type.

4. Under the outbound parameter table control, select **Create outbound parameter**.

   Partner type is **LS,** and the Message type is **DEBMAS**, which is the IDoc document type.

5. Leave **Partn.funct** blank.

6. Click the **Outbound options** tab.

   Provide the following information:

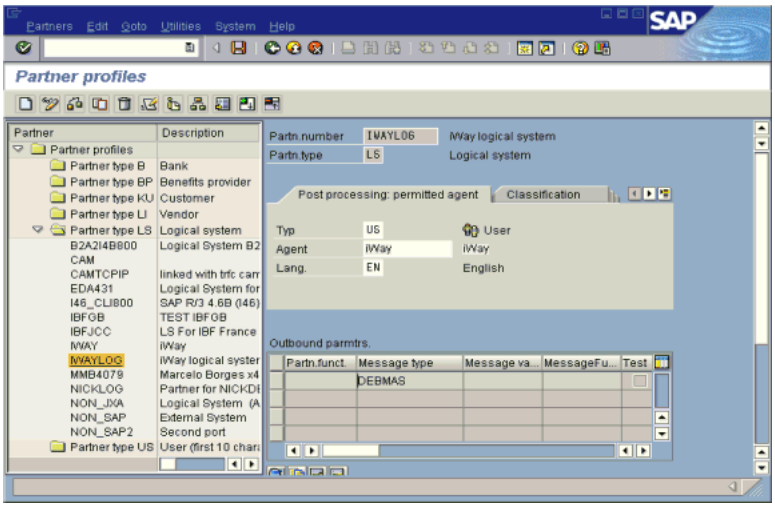
   a. Depending on your performance requirements, click **Transfer IDoc Immed** or **Collect IDocs**.

   b. For the IDoc, enter a message type, for example, DEBMAS.

   c. Enter a receiver port, for example, A000000036.

7. Click **Save** to save the session.

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



Collected IDocs

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

Make sure the DOCNUM field is included in the EDI_DC40 structure and that each IDoc has a unique sequence number within the collected IDoc file.

Creating a Distribution Model for the Partner and Message Type

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

Creating a Distribution Model

To create a distribution model called ORAMOD:

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

   You can also run the BD64 transaction.

   The Display Distribution Model window is displayed.

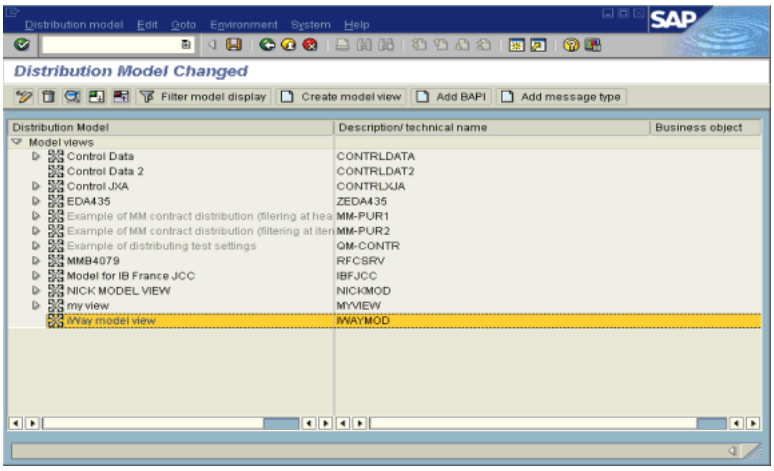
2. Select **Create model view**.

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

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

4. Click **Save**.

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

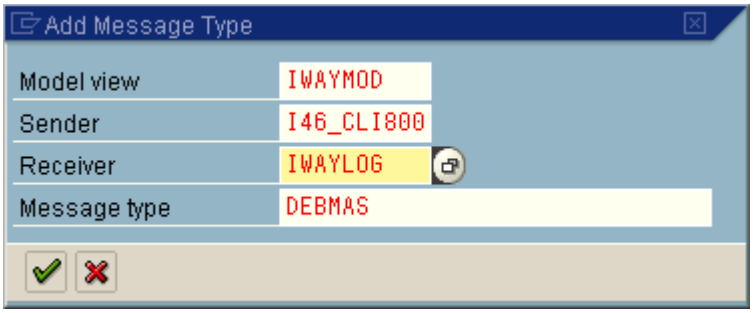


Perform the following steps:

a. In the Distribution Model tree, select a new model view.

b. On the right, select **Add message type**.

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



Provide the following information:

a. In the **Sender** field, provide the sender that points to the SAP R/3 system, which sends the IDoc, for example, `I46_CLI800`.

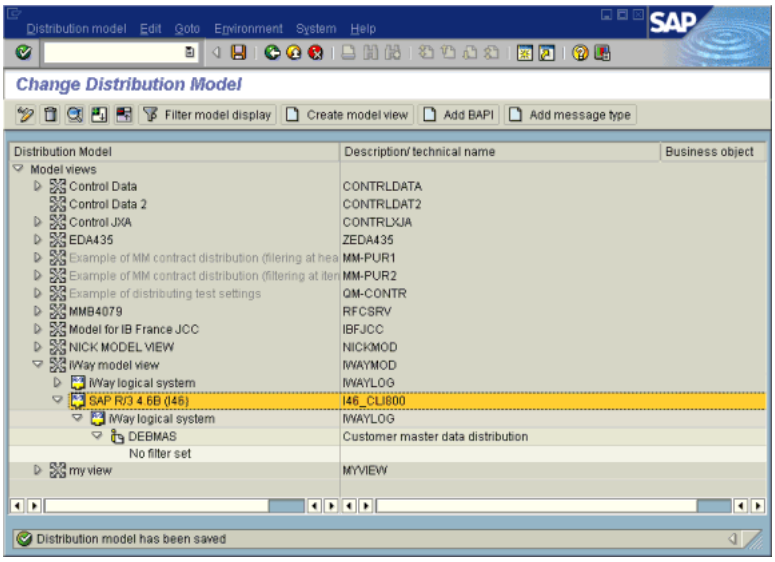
In this case, the sender is an SAP 4.6B system.

b. In the **Receiver** field, provide the logical system, for example, ORACLETDS.

c. In the **Message type** field, provide the type of IDoc, for example, DEBMAS.

5. Click the **check mark** icon.
6. Click **Save**.

The Change Distribution Model window displays the new model view to use to send message type, DEBMAS, from the I46_CLI800 SAP system to the ORACLETDS logical system.



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

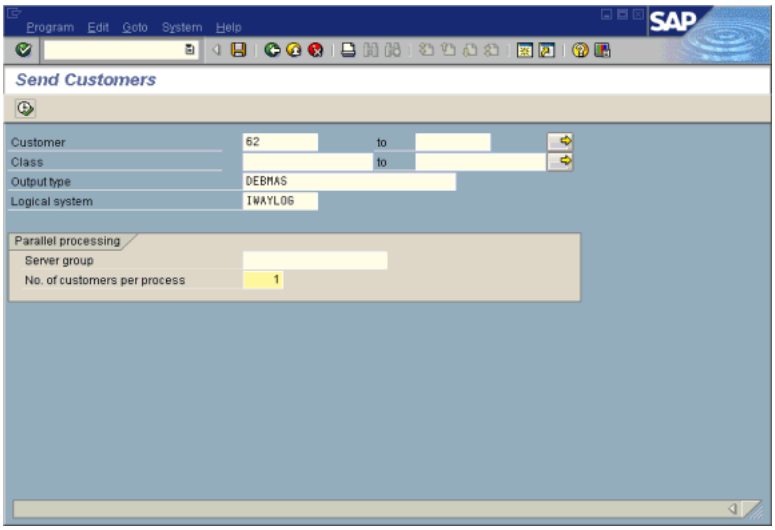
Testing the SAP R/3 ALE Configuration

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

Testing the SAP R/3 ALE Configuration

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

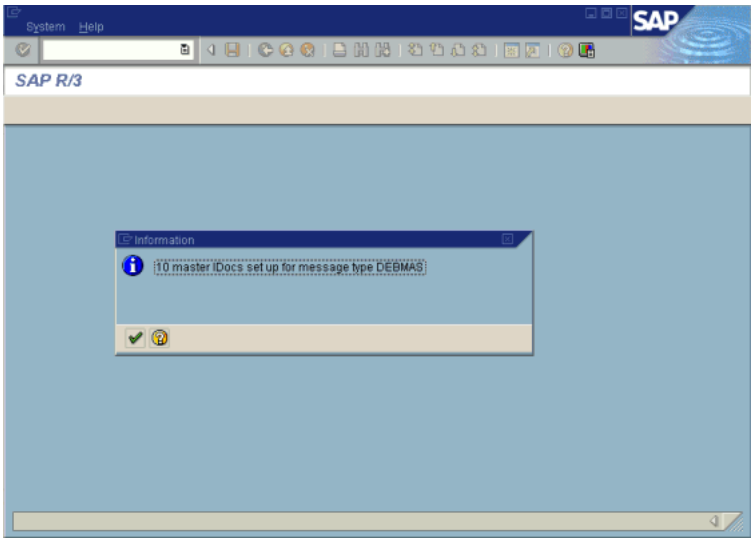
1. In the Send Customers window, enter the IDoc message type, for example, DEBMAS in the **Output type** field.



2. In the **Logical system** field, enter the logical system, for example, ORACLETDS.
3. Click **Run**.

   The SAP R/3 event adapter receives the IDoc in XML format. No response is expected from the event adapter.

   A confirmation window is displayed.


System Help
SAP
SAP R/3
Information
10 master IDocs set up for message type DEBMAS

Glossary

adapter

Provides universal connectivity by enabling an electronic interface to be accommodated (without loss of function) to another electronic interface.

agent

Supports service protocols in listeners and documents.

business service

Also known as a Web service. A Web service is a self-contained, modularized function that can be published and accessed across a network using open standards. It is the implementation of an interface by a component and is an executable entity.

channel

Represents configured connections to particular instances of back-end systems. A channel binds one or more event ports to a particular listener managed by an adapter.

listener

A component that accepts requests from client applications.

port

Associates a particular business object exposed by the adapter with a particular disposition. A disposition is a URL that defines the protocol and location of the event data. The port defines the end point of the event consumption.

Index

A

B

C

D

E

F

G

H

I

J

L

M

N

O

P

R

S

T

U

W

X