Oracle® Fusion Middleware Application Adapters

Release Notes for 12*c* (12.2.1.2.0)

E84211-01

December 2016

Provides release notes for the Oracle Application Adapters for Oracle WebLogic Server.

ORACLE

Oracle Fusion Middleware Application Adapters 12*c* (12.2.1.2.0) Release Notes for Oracle WebLogic Server, 12*c* (12.2.1.2.0)

E84211-01

Primary Author: Stefan Kostial

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

Contents

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

Preface

Welcome to *Oracle Fusion Middleware Application Adapters Release Notes for Oracle WebLogic Server*. This document provides release notes for Oracle Application Adapters for Oracle WebLogic Server.

Audience

This document is intended for system administrators who configure and use ERP application adapters.

Documentation Accessibility

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

Access to Oracle Support

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

Related Documents

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

- *Oracle Fusion Middleware Application Adapters Installation Guide for Oracle WebLogic Server*
- *Oracle Fusion Middleware Application Adapter Best Practices Guide for Oracle WebLogic Server*
- *Oracle Fusion Middleware Application Adapter Upgrade Guide for Oracle WebLogic Server*
- *Oracle Fusion Middleware Application Adapter for SAP R/3 User's Guide for Oracle WebLogic Server*
- *Oracle Fusion Middleware Application Adapter for Siebel User's Guide for Oracle WebLogic Server*

- *Oracle Fusion Middleware Application Adapter for PeopleSoft User's Guide for Oracle WebLogic Server*
- *Oracle Fusion Middleware Application Adapter for J.D. Edwards OneWorld User's Guide for Oracle WebLogic Server*
- Oracle's Unified Method (OUM)

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

  For more information about OUM, see the OUM FAQ at

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

Conventions

The following text conventions are used in this document:

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

1

Oracle Application Adapters for Oracle WebLogic Server

This chapter describes issues and workarounds for the following Oracle Application Adapters 12*c* (12.2.1.0.0) for Oracle WebLogic Server 12*c* (12.2.1.0.0):

- Oracle Application Adapter for PeopleSoft
- Oracle Application Adapter for SAP R/3
- Oracle Application Adapter for Siebel
- Oracle Application Adapter for J.D. Edwards OneWorld

Note: Throughout this document, *<ORACLE_HOME>* refers to the 12*c* (12.2.1.0.0) SOA/OSB installed home location.

<ADAPTER_HOME> refers to the following:

- For SOA:

  <ORACLE_HOME>\soa\soa\thirdparty\ApplicationAdapters

- For OSB:

  <ORACLE_HOME>\osb\3rdparty\ApplicationAdapters

Note: As of 12*c* (12.2.1.0.0), the generic .jar installer should be used to install the Oracle Application Adapters for Oracle WebLogic Server on all supported platforms.

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

Note: The 12*c* (12.2.1.0.0) user documentation will also be applicable for the 12*c* (12.2.1.1.0) and 12*c* (12.2.1.2.0) releases.

This chapter includes the following topics:

1.1 Oracle Application Adapters: New Features

The following topic discusses new features that pertain to Oracle Application Adapters 12*c* (12.2.1.0.0) for Oracle WebLogic Server 12*c* (12.2.1.0.0):

1.1.1 Latest Supported Systems

As of Oracle Application Adapters 12*c* (12.2.1.0.0), Oracle Fusion Middleware Application Adapters provide support for the following:

- Oracle Fusion Middleware Application Adapter for SAP R/3:

  Support for SAP Java Connector (SAP JCo) Version 3.0.14

- Oracle Fusion Middleware Application Adapter for PeopleSoft:

  Support for People Tools Version 8.54.

- Oracle Fusion Middleware Application Adapter for Siebel:

  Support for Siebel Public Sector Version 8.2.2

- Oracle Fusion Middleware Application Adapter for J.D. Edwards OneWorld:

  Support for J.D. Edwards OneWorld Version 9.10 with Tools Release 9.1.5.2

For more information, see the corresponding adapter user guide and the *Oracle Fusion Middleware Application Adapters Installation Guide for Oracle WebLogic Server*.

1.1.2 Mandatory 12*c* (12.2.1.0.0) Patch 22026475

In Oracle 12*c* (12.2.1.0.0), the following patch is required to ensure that Oracle Fusion Middleware Application Adapters 12*c* (12.2.1.0.0) for JDeveloper work as designed. Patch 22026475 should be installed on top of 12*c* (12.2.1.0.0) JDeveloper.

For more information on installing Patch 22026475, see the *Oracle Fusion Middleware Application Adapters Installation Guide for Oracle WebLogic Server 12c (12.2.1.0.0)*.

1.1.3 Mandatory 12*c* (12.2.1.1.0) Patch 24364695

In Oracle 12*c* (12.2.1.1.0), the following patch is required to ensure that Oracle Fusion Middleware Application Adapters 12*c* (12.2.1.1.0) for Service-Oriented Architecture (SOA) work as designed. Patch 24364695 should be installed on top of 12*c* (12.2.1.1.0) SOA.

For more information on installing Patch 24364695, see the *Oracle Fusion Middleware Application Adapters Installation Guide for Oracle WebLogic Server 12c (12.2.1.0.0)*.

1.1.4 Mandatory 12*c* (12.2.1.2.0) Patch 25088788

In Oracle 12*c* (12.2.1.2.0), the following patch is required to ensure that Oracle Fusion Middleware Application Adapters 12*c* (12.2.1.2.0) for JDeveloper work as designed. Patch 25088788 should be installed on top of 12*c* (12.2.1.2.0) JDeveloper.

For more information on installing Patch 25088788, see the *Oracle Fusion Middleware Application Adapters Installation Guide for Oracle WebLogic Server 12c (12.2.1.0.0)*.

1.1.5 Configuring Outbound and Inbound Process for Oracle Service Bus Using Sbconsole

With 12*c* (12.2.1.0.0), the sbconsole process creation screens have changed, but the Application Adapter user guides will have the older screenshots which can be ignored. The detailed steps for the Outbound and Inbound process for Oracle Service Bus using sbconsole with 12*c* (12.2.1.0.0) are described below in this section.

For more information, see Chapter 2, "Configuring an Outbound and Inbound Process for Oracle Service Bus Using sbconsole".

1.1.6 Logging Feature

In Oracle Application Adapters 12*c* (12.2.1.0.0), with the Logging feature, J2CA and BSE adapter logs will be updated within Oracle logs in the *{server-name}-diagnostic.log* file. This file is located in the following location:

```
<ORACLE_HOME>\user_projects\domains\base_domain\servers\<server_Name>\logs
```

For more information on the Logging feature, see the corresponding Oracle Application Adapter for Oracle WebLogic Server user documentation, and refer to the *Key Features* section.

1.1.7 Diagnosibility Feature

In Oracle Application Adapters 12*c* (12.2.1.0.0), the Diagnosibility feature captures the endpoint health status (where available) of the adapters, and provides a corresponding alert to the Oracle Adapter Framework, so that it may be displayed in the EM console.

For more information on the Diagnosibility feature, see the corresponding Oracle Application Adapter for Oracle WebLogic Server user documentation, and refer to the *Key Features* section.

Note: The Diagnosibility feature is not supported for the OSB component.

1.1.8 Debugging Feature

In Oracle Application Adapters 12*c* (12.2.1.0.0), the Debugging feature helps you debug a created SOA, BPM, and OSB process in JDeveloper.

For more information on the Debugging feature, see the corresponding Oracle Application Adapter for Oracle WebLogic Server user documentation, and refer to the *Key Features* section.

1.2 Oracle Application Adapters: General Issues and Workarounds

The following topics discuss general issues that pertain to Oracle Fusion Middleware Application Adapters, Oracle WebLogic Server Adapter J2CA, and Oracle WebLogic Server Adapter Business Services Engine (BSE):

1.2.1 Version Change for the Deployment of a SOA (BPEL, Mediator) and BPM Process

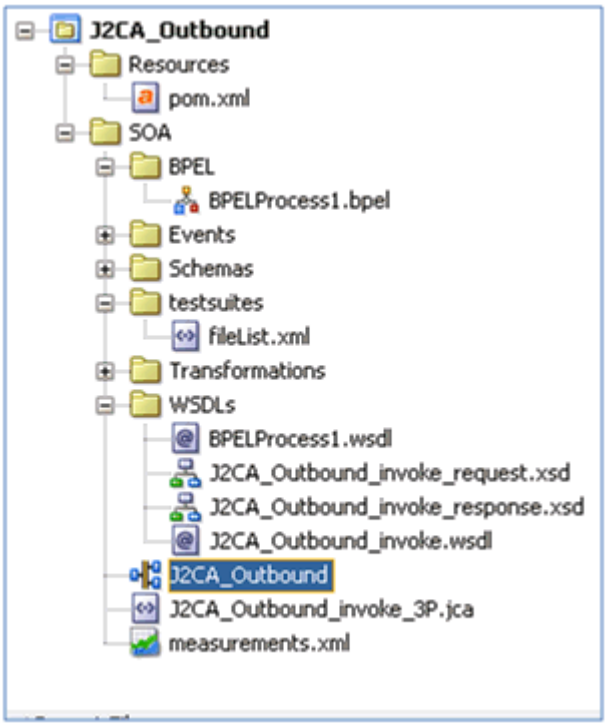
When deploying SOA (BPEL, Mediator) and BPMN J2CA outbound and inbound processes created in 12*c* (12.2.1.0.0) JDeveloper, the following error can be expected:

```
The JCA Binding Component is unable to locate the Resource Adapter specified in
the <connection-factory/> element:
location='eis/OracleJCAAdapter/DefaultConnection'
The reason for this is most likely that either
1) the Resource Adapters RAR file has not been deployed successfully to the
WebLogic J2EE Application server or
2) the JNDI <jndi-name> setting in the WebLogic JCA deployment descriptor has not
been set to eis/OracleJCAAdapter/DefaultConnection. In the last case you might
have to add a new 'connector-factory' entry (connection) to the deployment
descriptor.
Please correct this and then restart the WebLogic Application Server
```

Perform the following steps as a workaround to resolve this issue:

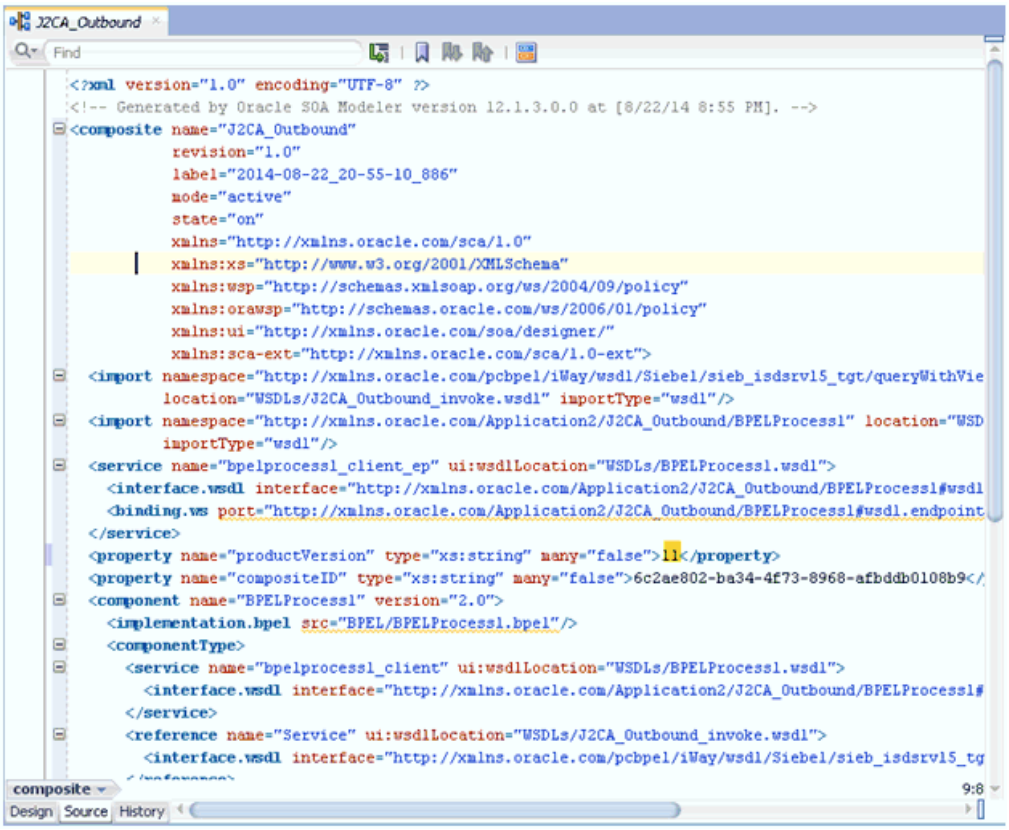
1. Double-click **J2CA_Outbound** (created BPEL process), as shown in Figure 1–1.

Figure 1–1 Created BPEL Process



2. Click the **Source** tab below the opened process.
3. Change the *productVersion* property value from 12.2.1.0.0.150921 to 11, as shown in Figure 1–2.

Figure 1–2 Source Tab

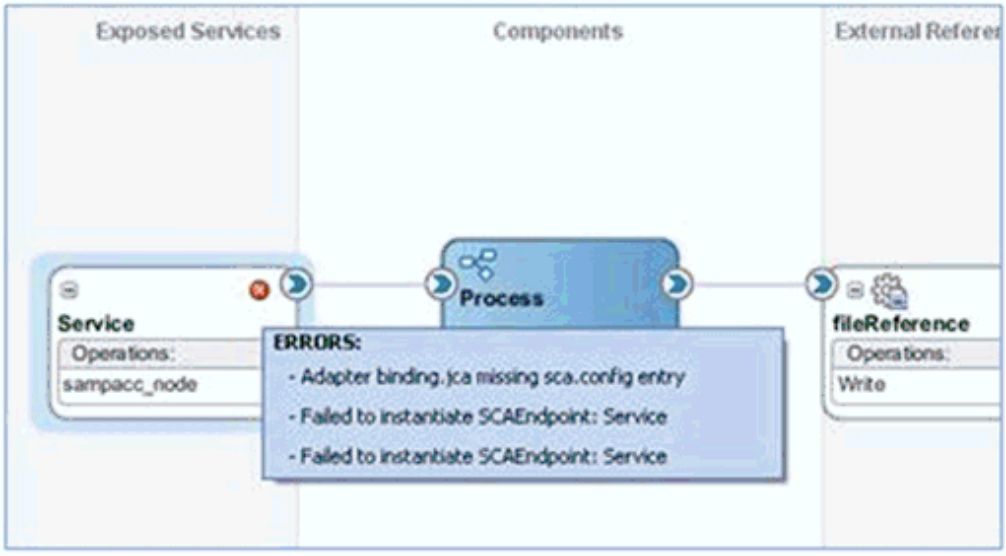


4. Save the changes and proceed with deploying the project.

1.2.2 Warning Message When Creating a Third-Party Service

When creating a third-party adapter for SOA (BPEL, Mediator) and BPM processes in 12*c* (12.2.1.0.0) JDeveloper, warning messages are displayed, as shown in Figure 1–3.

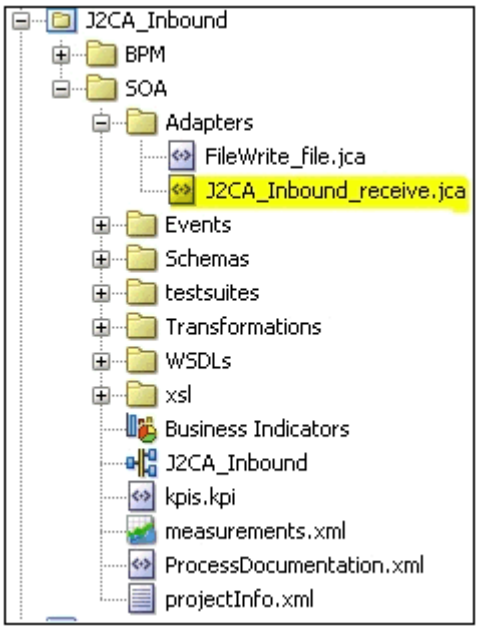
Figure 1–3 Warning Messages



The issue is that the J2CA file has not been converted from 11*g* to 12*c* format. This will not impact the functionality of the adapters. If you require these warning messages to be resolved, then perform the following steps:

1. Open the jca properties file (for example, expand J2CA_Inbound, SOA, Adapters, and then J2CA_Inbound_receive.jca), as shown in Figure 1–4.

Figure 1–4 JCA Properties File



2. Change `adapter="SAP Adapter"` to `adapter="3P"`.

The SAP adapter is being used in this workaround for demonstration purposes. This change is applicable for all other adapters.

1.2.3 JCA and BSE Links in Adapter Folder

Clicking the **JCA_test_link** or **IBSE_test_link** in the <ADAPTER HOME> folder will generate a page not found error.

Perform the following steps as a workaround:

1. Ensure the servers are started and running, and that the adapters are deployed and started.
2. Change the Managed server port in the URL and reload the servlet pages for J2CA and BSE.

1.2.4 Deployment Scripts

This section includes the following topics:

1.2.4.1 Limitation With Deployment Scripts

The deployment scripts do not work for the following environments:

- SOA/OSB cluster environment
- Adapters upgraded from PS6, 12*c* (12.1.3.0.0) to the 12*c* (12.2.1.0.0) environment

1.2.4.2 Deployment Scripts Failed for BSE and J2CA With Errors

Deployment scripts failed for BSE and J2CA with the following errors:

```
The file, '<ADAPTER_HOME>\ibse.war', does not exist.
The file, '<ADAPTER_HOME>\iwafjca.rar', does not exist.
```

Ensure the adapters are installed and available in the specified location.

1.2.4.3 Deployment Scripts Failed With Invalid Port Number

Deployment scripts failed with an invalid port number:

```
Unable to connect to 't3://localhost:7001': Destination localhost, 7001
unreachable; nested exception is:
java.net.ConnectException: Connection refused: connect; No available router to
destination. Ensure the url represents a running admin server and that the
credentials are correct. If using http protocol, tunneling must be enabled on the
admin server.
```

Ensure a valid Admin Server port is provided or ensure the servers are started and running.

1.2.4.4 Deployment Scripts Failed With Incorrect <ORACLE_HOME> Location

Deployment scripts failed with incorrect <ORACLE_HOME> location:

```
The system cannot find the path specified.
Error: Could not find or load main class weblogic.Deployer
```

Ensure the correct <ORACLE_HOME> location is specified.

1.2.4.5 Deployment Scripts Failed With Invalid WebLogic User Name

Deployment scripts failed with an invalid WebLogic user name:

```
Unable to connect to 't3://localhost:7101': User: weblogic123, failed to be
authenticated.. Ensure the url represents a running admin server and that the
credentials are correct. If using http protocol, tunneling must be enabled on the
admin server.
```

Enter the correct WebLogic user name.

1.2.4.6 Deployment Scripts Failed With Invalid WebLogic Password

Deployment scripts failed with an invalid WebLogic password:

```
Unable to connect to 't3://localhost:7101': User: weblogic, failed to be
authenticated.. Ensure the url represents a running admin server and that the
credentials are correct. If using http protocol, tunneling must be enabled on the
admin server.
```

Enter the correct WebLogic password.

1.2.4.7 Deployment Scripts Failed With Incorrect Server Name

Deployment scripts failed with an incorrect server name (for example, test):

```
'test' is not a configured target.
```

Ensure the server name provided is valid.

1.2.4.8 Undeployment Failed When Providing Incorrect Application Name for BSE/J2CA

Undeployment failed when providing incorrect application name for BSE/J2CA:

```
[Deployer:149001]
No application named "ibse1" exists for operation "undeploy".
```

Provide a valid application name for BSE/J2CA to be undeployed.

1.2.5 MS SQL Server and DB2 Support

With Oracle Application Adapters 12*c* (12.2.1.0.0), MS SQL Server and DB2 databases are not supported.

1.2.6 Supported ojdbc.jar File

When an Oracle enterprise database is configured as a repository with Oracle Application Adapters for Oracle WebLogic Server, only the *ojdbc7.jar* file is currently certified.

1.2.7 Supported Modes

The following section lists the supported modes for the Oracle Fusion Middleware Application Adapters.

Oracle Service-Oriented Architecture (SOA) and Oracle Business Process Management (BPM)

Supported modes:

- Managed mode

> **Note:** Oracle Fusion Middleware Application Adapters must be deployed to the Managed Server during deployment.

- Integrated Server mode

> **Note:** By default, the Oracle Fusion Middleware Application Adapters are deployed to the Integrated Server during deployment.

Oracle Service Bus (OSB)

Supported modes:

- Managed mode

> **Note:** Oracle Fusion Middleware Application Adapters must be deployed to the Managed Server during deployment.

- Integrated Server mode

> **Note:** By default, the Oracle Fusion Middleware Application Adapters are deployed to the Integrated Server during deployment.

Combined Oracle Service-Oriented Architecture (SOA) and Oracle Service Bus (OSB)

Supported modes:

- Managed mode

> **Note:**
> - Installing Oracle Fusion Middleware Application Adapters to the SOA location (for example, *<ORACLE_HOME>\soa\soa\thirdparty\ApplicationAdapters*) is sufficient to work with SOA and OSB in a combined SOA and OSB environment.
> - Oracle Fusion Middleware Application Adapters must be deployed to the Managed Server during deployment.

1.2.8 Testing Outbound BPEL and Mediator Processes

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

The Mediator data flows can be tested using the EM console. When creating a Mediator data flow and interactions, the Web services are created and registered with the Oracle Application Server. For more information about testing Web services, see your Oracle Application Server administrator and the following documentation:

- *Oracle Fusion Middleware Application Adapter for SAP R/3 User's Guide for Oracle WebLogic Server*
- *Oracle Fusion Middleware Application Adapter for Siebel User's Guide for Oracle WebLogic Server*
- *Oracle Fusion Middleware Application Adapter for PeopleSoft User's Guide for Oracle WebLogic Server*
- *Oracle Fusion Middleware Application Adapter for J.D. Edwards OneWorld User's Guide for Oracle WebLogic Server*

1.2.9 Certified Repository

Oracle Application Adapters have been certified with the Oracle enterprise database as the repository. The version that was certified is Oracle Database 12*c* Enterprise Edition (12.1.0.2.0).

Other versions of the Oracle enterprise database are also supported as long as they are supported by the Oracle SOA Suite. Except for the Oracle enterprise database, Oracle Application Adapters do not support any other database, including Oracle XE, Oracle Berkeley Database, or databases from other vendors.

1.2.10 HTTP Repository Connection

HTTP repository connections are not supported for Oracle Application Adapters 12*c* (12.2.1.0.0), which prevents you from connecting to the Oracle Application Adapters instance from a remote machine. As a workaround, you must install Application Explorer on every platform where integration with Oracle Application Adapters using Application Explorer is required.

1.2.11 File Repository Usage

Do not use a File repository in development, testing, and production environments. Only use the Oracle database repository.

1.2.12 Using Business Services Engine for Inbound Processing

Using Business Services Engine (BSE) for inbound processing is not supported. BSE only supports services (outbound).

1.2.13 Synchronous Events

Synchronous event handling is not supported for the Oracle Application Adapter for Siebel, Oracle Application Adapter PeopleSoft, and Oracle Application Adapter J.D. Edwards OneWorld.

1.2.14 Ports Option for Inbound Processing

The ports option for inbound processing is not supported for J2CA events.

1.2.15 Supported Custom Objects

Oracle continues to support the custom objects of an EIS. However, Oracle cannot guarantee the support for all custom objects at the customer environment. Support to the custom object will be considered on a case by case option. Custom objects could fall under any of the following category:

- **SAP**

  BAPIs, RFCs, and ALE/IDocs

- **Siebel**

  Business Objects, Business Services, and Integration Objects

- **PeopleSoft**

  Component Interfaces and Messages

- **J.D. Edwards OneWorld**

  Business Functions and Transaction Types

It is recommended that customers who wish to troubleshoot an issue with the custom objects of an EIS, provide the following to Oracle:

1. Data and definition of custom objects.
2. Request and response XML documents for the custom objects.
3. Reproduction steps for the custom object.

1.2.16 Adapter Compatibility

You cannot have multiple versions of the client library files in the same `lib` directory. Therefore, you cannot have one instance of the adapter configured to connect to different EIS versions at the same time by using different versions of the client library files.

1.2.17 Encoding Support

The Oracle Application Adapters support only UTF-8 encoding.

1.2.18 J2CA Components

If you are working with a J2CA configuration and you create, update, or delete an adapter target or channel using Application Explorer, then you must restart Oracle WebLogic Server. This is required to refresh the repository and recognize the new/updated target or channel in the J2CA Test Servlet, BPEL process, Mediator process, BPM process, and OSB process. The default URL for the J2CA Test Servlet is:

`http://`*`hostname`*`:`*`port`*`/iwafjca`

This applies to J2CA targets and channels only, not BSE targets. This also applies when a target or a channel parameter is modified using Application Explorer.

1.2.19 Invoking a Run-Time Message When BSE Is Unavailable

When invoking a run-time message while BSE is unavailable, a "Whitespace required" exception error occurs.

The error message "unable to connect to URL http://*host*:*port*/ibse..." appears.

1.2.20 Invoking Outbound BPEL Process at Run Time Using BSE

When invoking an outbound BPEL process at run time using BSE, a "Not able to find SoapRouter" message appears.

Workaround: Add the following element in the respective XML file:

```
<property name "optSoapShortcut">false</property>
```

1.2.21 Using the J2CA Test Tool for Outbound Only

The J2CA Test Tool must be used only for outbound (services). Inbound (events) will list only the channels available for the corresponding adapters.

1.2.22 BSE Web Services Browser Pages Do Not Support DBCS Input

When using a BSE configuration, the browser-based test tools add encodings to the content being sent. The BSE test tool causes characters to become garbled due to these added encodings. Therefore, you cannot perform a test using input that contains Japanese characters. A possible workaround is to use a tool that can send pure SOAP requests.

1.2.23 Adapter Language Certification

Oracle Application Adapters are certified with the English language. The following Oracle Application Adapters 12*c* (12.2.1.0.0) are certified with the Japanese language:

- Oracle Application Adapter for SAP R/3 (using SAP JCo 3.x)
- Oracle Application Adapter for Siebel
- Oracle Application Adapter for PeopleSoft

The Oracle Application Adapter for J.D. Edwards OneWorld is not certified with the Japanese language, but it is supported. If you encounter any issues with other languages, then contact Oracle Customer Support for a workaround.

1.2.24 File Channel

The File channel for events is not supported in production environments for the Oracle Application Adapter for PeopleSoft, Oracle Application Adapter for J.D. Edwards OneWorld, and Oracle Application Adapter for Siebel. The File channel is available only for testing purposes in a non-network environment. As a best practice, it is not recommended to use the File channel.

1.2.25 Unsupported Adapter Functionality

The Oracle Application Adapter for SAP R/3, Oracle Application Adapter for PeopleSoft, Oracle Application Adapter for Siebel, and Oracle Application Adapter for J.D. Edwards OneWorld does not support functionality such as Transaction, XA, and two-phase commit.

1.2.26 Upgrading Adapters (BSE Configuration) on Windows Platforms

For the Oracle Application Adapters 12*c* (12.2.1.0.0), after upgrading the adapters on a Windows platform, if you update the adapters for a BSE configuration and click **Activate Changes**, the following error may be generated:

```
An error occurred during activation of changes, please see the log for details.
[Deployer:149258]Server failed to remove the staged files 'C:\Oracle_Home\user_
projects\domains\base_domain\servers\soa_server1\stage\ibse\ibse.war' for
application 'ibse' completely. Check the directory and make sure no other
application using this directory. This will result in inappropriate results when
this server gets partitioned and trying to deploy this application.
```

Perform the following steps as a workaround:

1. Release the configuration from the WebLogic Administration Console.
2. Restart the Managed Server.

1.2.27 Application Explorer Online Help

Application Explorer online help is supported only on Microsoft Internet Explorer and Mozilla FireFox browsers. It is not supported on Google Chrome browsers. If your default browser is Google Chrome, then use Microsoft Internet Explorer or Mozilla FireFox if you would like to view the online help.

1.3 Application Explorer

The following topics discuss issues that pertain to Application Explorer:

- Section 1.3.1, "Unable to Connect to Remote Machines Using File or DB Repositories for J2CA Configurations"
- Section 1.3.2, "Unable to Connect to a J2CA Configuration"

1.3.1 Unable to Connect to Remote Machines Using File or DB Repositories for J2CA Configurations

Application Explorer cannot be used to connect to remote machines for a J2CA configuration when using a File or DB repository. When using a J2CA configuration, SOA Suite must be installed on the same machine that is being used as the container for the adapters.

1.3.2 Unable to Connect to a J2CA Configuration

When connecting to a J2CA configuration in Application Explorer, the following error message may be generated:

```
oracle/tip/adapter/api/exception/PCResourceException
```

Perform the following steps to resolve this issue:

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

   <ORACLE_HOME>\user_projects\domains\base_domain\bin

2. Run **setDomainEnv.cmd** (Windows) or **. ./setDomainEnv.sh** (UNIX/Linux).

   The `setDomainEnv` command sets the class path and other environment variables for Application Explorer in the Oracle WebLogic Server environment.

3. Do not close the command prompt window.
4. Navigate to the following directory:

   `<ADAPTER_HOME>\tools\iwae\bin`

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

   You can now connect to a J2CA configuration.

1.4 Oracle Application Adapter for PeopleSoft: Issues and Workarounds

The following topics discuss issues that pertain to Oracle Application Adapter for PeopleSoft:

1.4.1 Platform Support

For 12*c* (12.2.1.0.0), Oracle Application Adapter for PeopleSoft is certified only with Windows and Oracle Enterprise Linux platforms. If support on any other platforms is required for this adapter, then contact Oracle customer support.

1.4.2 Limitation With PeopleTools 8.40 Support

With Oracle Application Adapter for PeopleSoft 12*c* (12.2.1.0.0), configuring and using adapter targets and channels is not supported for PeopleTools 8.40. However existing processes will continue working without any issues.

1.4.3 PeopleSoft LDAP Authentication Incompatible

PeopleSoft LDAP authentication relies on sign-on PeopleCode. However, the authentication services that PeopleSoft provides with Component Interfaces do not invoke sign-on PeopleCode, so you cannot use PeopleSoft LDAP authentication with Oracle Application Adapter for PeopleSoft.

Workaround

None.

1.4.4 Automatic Reconnect to PeopleSoft

Oracle Application Adapter for PeopleSoft does not automatically reconnect when a connection to PeopleSoft becomes unavailable.

Workaround

None.

1.4.5 HTTPS Protocol

Oracle Application Adapter for PeopleSoft does not support the HTTPS protocol for events.

Workaround

None.

1.4.6 PeopleSoft Messages

Oracle Application Adapter for PeopleSoft uses messages only for events and not for services.

Workaround

None.

1.4.7 Limitation with Level 2 Scrolls

PeopleSoft has acknowledged a limitation with Component Interfaces that contain level 2 scrolls. If you try to insert a new row on a level 2 scroll, then a Null Pointer Exception error occurs.

If you receive this error, then you must upgrade your PeopleSoft release level.

This limitation is fixed in PeopleTools Version 8.16.08, and PeopleTools Version 8.17.02 in the 8.1x code line. It is tracked by PeopleSoft Incident T-MZYGAR-2C5YS.

In the 8.4x code line, this limitation is tracked by PeopleSoft Incident T-TCHURY-YZ9FR and is fixed in PeopleSoft 8.41.

Workaround

None.

1.4.8 Limitation with Level 3 Scrolls

PeopleSoft has acknowledged a limitation with Component Interfaces that contain level 3 scrolls. If you try to insert a new row on a level 3 scroll, then a Null Pointer Exception error occurs.

If you receive this error, then you must upgrade your PeopleSoft release level.

This limitation is fixed in PeopleTools Version 8.18 and is tracked by PeopleSoft Incident T-MZYGAR-D2529. However, this is still a limitation in PeopleSoft 8.41 and 8.42 (PeopleSoft Incident T-MZYGAR-3F72X). PeopleSoft has reported that this will be fixed in 8.43 (PeopleSoft incident report, 562734000).

Workaround

None.

1.4.9 Limitation with Effective Dated Scrolls

PeopleSoft has acknowledged that there is a limitation with effective dating and multiple transactions. If you want to insert multiple effective dated rows for the same primary keys, then you must use two separate transactions. This limitation is tracked by PeopleSoft Incident T-ACESAR-BS362.

Workaround

None.

1.4.10 Limitation When Inserting a Second Row for Level 1, 2, or 3 Scrolls

If you are trying to insert a level 1, 2, or 3 scroll and the following conditions exist:

- There is exactly one row for the level 1, 2, or 3 scroll.
- There are required field names that end in a numeric value.

An error message appears, in the following format:

```
This is an invalid property {ADDRESS_1} (91,15)
```

This example was created using the VNDR_ID Component Interface from the Financials application. Note that the actual property name is ADDRESS1.

Workaround

As a workaround, perform the following steps:

1. In the PeopleSoft Application Designer, open the Component Interface with which you are working.
2. Select the property that ends in a numeric value.
3. Right-click and select Edit Name from the context menu.
4. Change the name of the property.

   You can select a name that does not end in a number (for example, ADDRESSA), or you can add an underscore (for example, ADDRESS_1).
5. Save the Component Interface.
6. Regenerate the Java APIs for the Component Interface.
7. Use the revised property name in your XML transaction.

1.4.11 PeopleTools Date Format

The YYYY-MM-DD date format does not work for a Component Interface key.

PeopleSoft has acknowledged this limitation for most releases of PeopleTools and has addressed it in the newest releases. For more information, see PeopleSoft Resolution ID 200730918.

Workaround

Use the MM/DD/YYYY format. Alternatively, you can write a PeopleSoft method that takes a date format of YYYY-MM-DD, changes the date to a string, reformats it to DD/MM/YYYY, and passes it to the Component Interface date.

1.4.12 Generating Java APIs

It is possible to create Component Interfaces within PeopleSoft that are internally inconsistent. Inconsistencies have been found in some Component Interface templates that have been delivered by PeopleSoft. Indicators of this problem include errors when generating the Java APIs in the PeopleSoft Application Designer.

If you encounter errors while generating the Java APIs, then the Component Interface is likely to malfunction and can possibly corrupt your database. The correct operation of Component Interfaces when errors appear during API generation cannot be guaranteed. In addition, it is strongly recommended that the source of the errors is fixed before continuing.

Workaround

Fix the Component Interface using PeopleTools.

1.4.13 Differences Between Component Interface Functionality and Adapter Functionality

The differences between Component Interfaces and standard application functionality relating to panel processing have been observed. Among the possible symptoms of these differences are messages such as, "First Operand of . is NULL." Oracle Application Adapter for PeopleSoft should replicate the functionality of a PeopleSoft Component Interface, but only when the Component Interface is run through the PeopleSoft application server in three-tier mode.

If you notice differences between expected Component Interface functionality and adapter functionality, then you must verify that the differences are real by running the Component Interface with the PeopleTools Component Interface test tool in three-tier mode.

Workaround

Test the Component Interface using the PeopleTools Component Interface testing tool in three-tier mode only.

1.4.14 Missing Field Errors When Using a Component Interface

When using a Component Interface, it is difficult to determine which of the required fields are missing when you receive a PeopleSoft error message that states, "The highlighted field is required."

Workaround

You can edit the message in the PeopleSoft message catalog to pass a variable for the field name. For more information, see PeopleSoft Resolution 200731449.

1.4.15 Support for Related Display Fields

Related display fields are not supported by Component Interfaces.

Workaround

For more information, see PeopleSoft Resolution 200731974, which offers several workarounds.

1.4.16 Differences Between Component Interface Functionality and Adapter Functionality

PeopleSoft has acknowledged problems in the Component Interface back-end processor, which will cause the adapter to act differently than the Component Interface test tool in three-tier mode for certain Component Interfaces.

PeopleSoft Case 1965239 describes a problem with the CI_JOB_DATA_HIRE Component Interface in HR 8.1x. In this situation, the NAME field is not populated by PeopleCode correctly. The workaround is to manually populate the NAME field through the XML.

PeopleSoft Resolution ID 200728981 describes a problem with the JOBCODE Component Interface, which does not allow the REG_TEMP field to be changed to empty. As a workaround, customers must upgrade to a later release of PeopleTools.

Workaround

The workaround is dependent on the Component Interface.

1.4.17 Multiple Effective Dated Scrolls

A failure occurs when inserting multiple effective dated rows.

Workaround

If you want to insert multiple effective dated rows for the same primary keys, then you must use two separate transactions. This limitation is tracked by PeopleSoft Incident T-ACESAR-BS362.

1.4.18 Debug Message

For certain releases of PeopleTools 8.4x, you may receive the following message in your debug window:

```
PSProperties not yet initialized!
```

This is a PeopleSoft warning message you can ignore.

Workaround

None.

1.4.19 LOCATION Component Interface

When you attempt to access the LOCATION Component Interface using the Human Resources application, release 8.80.000, a failure occurs during runtime and the following message is displayed:

```
Component Interface Not Found
```

This is due to a problem in the way the PeopleSoft application has been delivered and is not related to the PeopleTools release.

Workaround

Perform the following steps:

1. Open the Component Interface in the PeopleTools Application Designer.
2. Make a small change to the Component Interface.
3. Undo the change.
4. Save the Component Interface.

This procedure resets certain internal PeopleSoft data structures, which enables Oracle Application Adapter for PeopleSoft to find the Component Interface. This has been observed in the LOCATION Component Interface running on the Human Resource applications release 8.8.000 on several different releases of PeopleTools, but it may occur in other Component Interfaces as well.

1.4.20 Component Interface Names

Although PeopleSoft permits Component Interface names that begin with certain special characters (such as an underscore), Application Explorer does not recognize such names.

Workaround

Begin Component Interface names with the letters A-Z or the integers 0-9.

1.4.21 Component Interface Java API Compilation Errors (People Tools 8.46)

When you compile all Java APIs for the Component Interfaces using People Tools 8.46, you may encounter compilation errors with some of the Component Interfaces.

Workaround

You can manually correct the Java source code of the failing Component Interfaces. Alternatively, if the failing Component Interfaces are not going to be used, then remove them from the API build process and do not include them in the build.

1.5 Oracle Application Adapter for SAP R/3: Issues and Workarounds

The following topics discuss issues that pertain to Oracle Application Adapter for SAP R/3 (using SAP JCo 3.x).

1.5.1 Starting Multiple Channels With the Same Program ID

Multiple instances are not created for a SAP JCo server implementation. Only one instance is created. As a result, you cannot start multiple channels with the same Program ID in Application Explorer. You can create multiple channels with the same Program ID, but the channels cannot be started simultaneously. For example, if two channels are using the same Program ID, then you must stop the first channel before you can start the second channel.

This limitation is not applicable for runtime scenarios. During runtime, the adapter handles multiplexing channels and Program IDs through the inbound schema validation method. However, this is not supported for Application Explorer runtime.

1.5.2 Supported Versions and Platforms for SAP JCo 3.0.14

The following SAP ERP platforms are supported by the Oracle Application Adapter for SAP R/3 (using SAP JCo 3.0.14):

- 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

The following operating systems are supported by the Oracle Application Adapter for SAP R/3 (using SAP JCo 3.0.14):

- Windows 64-bit Windows Server 2012
- Linux (Intel processor only) - (64-bit)
- HP-UX PA-RISC - (64-bit only)
- HP-UX Itanium - (64-bit only)
- Solaris - (64-bit only)
- AIX - (64-bit only)

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

1.5.3 Native IDoc Format Support

The native IDoc format for inbound processing is not currently supported with Oracle Application Adapter for SAP 12*c* (12.2.1.0.0). This issue will be resolved by applying a patch in a future release.

1.5.4 SAP R/3 4.6C Support

Oracle Application Adapter for SAP R/3 uses SAP JCo APIs that are provided by SAP. The support for this adapter is paired with the official supported versions of SAP JCo.

SAP R/3 4.6C is out of general maintenance by SAP. Oracle Software can access SAP 4.6C systems using the current SAP JCo API. Adapter service is only provided by the SAP JCo API via the SAP RFC interface. If there are any application processing issues arising because of release compatibility, then the adapter user (customer) is responsible to resolve those issues by consulting with SAP. Oracle Software does not provide support for SAP application or communication compatibility issues for SAP JCo and SAP R/3 4.6C systems.

1.5.5 Date and Time Field Mapping

Oracle Application Adapter for SAP R/3 (using SAP JCo 3.x) only handles date of format "yyyy-MM-dd" and time of format "HH:mm:ss".

Many remotely callable functions have a field that has a DATE format. The DATE field object of an adapter is equivalent to the ABAP DATE object as an 8 byte string with the format YYMMDD. In the SAP GUI, other profiles may be executed that transform the data display in the SAP GUI environment. However, the data is always stored in the DATE object format. For ease of use, the SAP Java Connector (JCo) converts data for DATE objects that also have the format YYYY-MM-DD to YYYYMMDD. A field must have format DATE (ABAP type D) to enable these transformations. RFC and BAPI functions employ fields with type D objects. All data for IDocs is type C (Character) by definition of the EDI_DD40 structure in the ABAP dictionary. Therefore, IDocs only accept the YYYYMMDD format as no conversions are performed on the field.

The TIME field object of an adapter is equivalent to the ABAP TIME object as a 6 byte string with the format HHMMSS. In the SAP GUI, other profiles may be executed that transform the time display in the SAP GUI environment. However, the data is always stored in the TIME object format. For ease of use, the SAP Java Connector (JCo) converts data for TIME objects that also have the format HH:MM:SS to HHMMSS. A field must have format TIME (ABAP type T) to enable these transformations. RFC and BAPI functions employ fields with type T objects. All data for IDocs is type C (Character) by definition of the EDI_DD40 structure in the ABAP dictionary. Therefore, IDocs will only accept HHMMSS as no conversions are performed on the field.

1.5.6 Intermediate Documents (IDocs) During Inbound Processing

When using collected IDocs during inbound processing (service mode) where one XML file contains multiple IDocs, a unique sequence number must be provided to identify each individual IDoc in the sequence. If the SAP application server cannot determine a correct sequence from the collected IDocs, then data from the first IDoc is taken and added (duplicated) to each subsequent IDoc. All data segments in subsequent IDocs in the collected IDoc file are ignored. For more information on this topic, consult the SAP ALE reference documentation.

Encoding

Encoding is a general term encompassing communications, hardware, software and instance documents. Encoding is usually not noticed until it is incorrect, when a document contains invalid characters with blank spaces or “?” or “#” characters. There are several areas to examine when investigating an encoding issue:

- Communications Channel

The communications channel encoding between the SAP server and the client machine is managed by the SAP server. The relevant client libraries query the client machine for the encoding setting and configure the communication with this information

- Hardware

  The client machine encoding can be determined via the Windows Control Panel Applet "Region and Language" or the "Set_Locale" environment variable on Unix or Linux machines. When using the "Set_Locale" variable, the actual Locale files must be present on the client machine. On Windows machines, the required language pack must be installed.

- Software

  Character sets are the characters or symbols of a given language or languages, and the numeric value that is used by the client machine to map data to the correct character or symbol. Data is represented in character sets and encoding is the means of transmitting this information to another character set, either local or remote. If a given character has a correct numeric value, but a missing symbol, then this results in the common convention of using "?" for the character. The resolution may be as simple as switching to a different encoding. There is also a chance that fonts, which are used to render characters or symbols on a screen or paper, do not have the correct representation or any representation for a given numeric value. This also must be checked.

  A source of encoding confusion that is often overlooked is the JVM encoding, which can prevail if a document is missing an encoding statement. Missing encoding declarations can occur and unless a specific application catches this, often the JVM encoding is used as the encoding, Always check the JVM encoding parameters if an encoding issue occurs to ensure that this issue is not the cause.

- Instance Document

  The XML header declaration contains an encoding statement. It should be remembered that it is a declaration, not a promise. A document may state "UTF-8" but be saved in "ASCII" or some other format. Mismatched document encoding is a common mistake that usually can be resolved by opening it in an editor that supports saving in different encodings and save to the declaration statement. However, always ensure when saving to a given encoding that all characters are present and correct after the save.

1.5.7 SAP IDoc Data With DBCS (Japanese, Chinese, etc.) Characters Overflows and Truncates Characters

SAP IDoc data with Japanese DBCS characters overflows and truncates characters in all BSE and J2CA events and services.

Explanation:

This issue only occurs on non-Unicode SAP MDMP environments, where one character can be two or more bytes. As an example of this issue, when using Japanese, the SAP field length is four characters. The English word "ball" fits correctly into the field because one character equals one byte. The Japanese word for ball in Shift-Jis encoding is three characters, but two bytes per character, so the last character is truncated and the last character appears in the next field. Since IDocs are positional delimited, this can cause errors in processing. This occurs because SAP uses character length, not byte length for all non-Unicode field lengths. There is no work around on this issue other than using Unicode or using shorter text in IDocs in DBCS.

1.5.8 No Values Set for Code and Details in Binding Fault

If you use the wrong request XML to invoke a SAP outbound process in Oracle BPEL Console, then the instance will be faulted and a binding fault will be thrown. Log in to the Oracle BPEL Console, select the faulted instance, and then click **Audit**. No values are set for Code and Details in the binding fault.

Workaround

None.

1.5.9 SAP R/3 Adapter Exceptions

During outbound processing, instead of receiving adapter exceptions from the BPEL or Mediator layer in the Enterprise Manager (EM) console, you can perform the following steps to receive adapter exceptions in XML format:

1. Open Application Explorer and connect to a configuration.
2. Expand the MySAP adapter node to view the available targets.
3. Right-click an available MySAP target node and select **Edit**.

   The Application Server dialog displays the target connection information.
4. Click the **Advanced** tab.
5. From the Error Handling list, select **Creates Error Document**.
6. Click the **User** tab.
7. In the Password field, type a valid password for the SAP R/3 application.
8. Click **OK**.
9. Close Application Explorer.

   Oracle BPEL or Mediator generates the error message in the response XML document. For example, if you use the GetDetail method for the CompanyCode SAP BAPI in your outbound processing, the following error message is shown in the XML response:

```
<companycode_get_detail_
oct24ProcessResponseurn:sap-com:document:sap:business.responsehttp://xmlns.orac
le.com/companycode_get_detail_oct24>
<COMPANYCODE_ADDRESS> </COMPANYCODE_ADDRESS>
<COMPANYCODE_DETAIL> </COMPANYCODE_DETAIL>
<RETURN>
<TYPE>E</TYPE>
<CODE>FN020</CODE>
<MESSAGE>Company code 1010 does not exist</MESSAGE>
<LOG_MSG_NO>000000</LOG_MSG_NO>
<MESSAGE_V1>1010</MESSAGE_V1>
</RETURN>
</companycode_get_detail_oct24ProcessResponse>
```

1.5.10 Behavior With Multiple Channels

When different channels are created using identical connection parameters (server, gateway, and program ID) the SAP gateway automatically enters load balancing mode. The load balancing algorithm is determined by the gateway profile configuration during gateway installation. Typically these include least used, lightest load, or several other SAP selections. If you are intentionally setting load balancing,

then consult the gateway manager for the proper procedure to configure the adapter and Oracle instances to take advantage of load balancing features on the system. If load balancing is accidentally triggered by configuration errors, then it can result in missing messages (sent to anther server) or wrong destination.

For example:

Shipping selects server A1 with gateway 01 and program ID MyProg.

Purchasing selects server A1 with gateway 01 and program ID MyProg.

Shipping sends ten messages and only six appear at the BPEL process.

Purchasing sends ten messages and only five appear at the BPEL process.

Shipping cannot understand what happened to four missing messages.

Shipping cannot understand why they have five purchasing messages.

The same applies for Purchasing.

Workaround

Change the program ID to a unique key for each department, as this is not a load balanced scenario.

1.5.11 Connections Not Listed in SAP Gateway Monitor

There are some occurrences when connections being made to SAP R/3 using Application Explorer are not listed in the SAP Gateway Monitor (transaction SMGW). This issue pertains to Oracle Application Adapter for SAP R/3 (using SAP JCo 3.0).

The adapter does not connect directly to the SAP R/3 system, but to the SAP JCo. Since the SAP JCo manages the connections to SAP R/3, and most connections are rapid and transient, most client (inbound) connections to SAP R/3 do not display in the SAP Gateway Monitor (transaction SMGW). Regular SAP JCo and RFC client traces are available from the client tracing options as described in the *Oracle Fusion Middleware Application Adapter for SAP R/3 (SAP JCo 3.0) User's Guide*. Server connections are persistent and are listed under "logged on Clients" in transaction SMGW.

1.5.12 Multithreading Functionality With the Outbound Listener

The adapter usually attempts to start three threads for each SAP outbound event channel. This is performance consistent with the SAP Gateway model, where only one thread is active at a time to emit. However, in the adapter one thread is emitting, one thread is writing, and one thread is in cleanup. If the gateway administrator enables multithreading on a particular program ID, then multiply the threads by three when setting the threads for maximum performance.

1.5.13 Single Sign-On

SAP supports single sign-on by means of authentication tickets created in its own programs. Currently, there is no single sign-on support between Oracle Application Servers and SAP ERP systems.

1.5.14 Metadata Support

SAP ERP Version 6.0 introduced many new data structures, such as deep (multi-level) structures, nested structures (structures where each column is itself a structure), and

tables of such data structures. Currently, only deep and nested structures are supported. Nested tables or line types are currently not supported by the adapter.

1.5.15 Nested and Deep Structure Support for the Test Run Facility

The Application Explorer Test Run facility supports all of the SAP basic types, flat structures, and SAP tables with a single input row. However, *Nested* and *Deep Structures* are not supported from the Application Explorer Test Run facility. It is a run time functionality only.

1.6 Oracle Application Adapter for Siebel: Issues and Workarounds

The following topics discuss issues that pertain to Oracle Application Adapter for Siebel.

1.6.1 Additional Configuration in the Siebel Environment

Some out-of-the-box Siebel business services may require additional setup steps in the Siebel environment before they can be run successfully. For example:

- When using a business service such as EAI XML Converter, before generating the XSD for the integration object, you should use Siebel tools to remove the 'xml container element' tag in the integration components you plan to use.
- When building a solution with the EAI dispatch service business service, you may need to set up a named subsystem to handle HTTP requests.

1.6.2 Service and Integration Nodes

Service nodes and Integration node can be created using only alphanumeric characters and "-" "_". No other special characters are allowed.

1.6.3 Automatic Reconnect to Siebel

When connecting to Siebel using the Java Data Bean Interface, you cannot reconnect after initial connection loss. This might occur when Application Explorer experiences a brief loss of network connection or if the Siebel Server or Gateway Service is restarted while Application Explorer is logged into the Siebel application.

Workaround

To log in successfully to the Siebel application, restart your application server and Application Explorer. This is a known Siebel API issue. For more information, see Siebel Alert 984.

1.6.4 Updating or Deleting Siebel Records Using Oracle Application Adapter for Siebel

If you are logged in as a Siebel user that does not belong to the team that owns the record being updated or deleted, then you cannot perform the action. By default, the adapter is set to 'My' view. However, in Siebel Access Control there are other views, such as 'All' view and 'Organization' view. Therefore, even if the user does not belong to the team and the record is not visible in 'My' view, the user may be able to update or delete the record through another view in the Siebel front end. This is not possible through the adapter. The adapter requires that the user is part of the team of the record being updated or deleted.

Workaround

There are two possible workarounds:

- Log in through the Siebel adapter as a user that is already part of the team that owns the record you need to update or delete.
- Add the user to the team that owns the record you need to update or delete.

1.6.5 Adapter Exception Error If Siebel Request Document Contains Japanese Characters

An adapter exception error is returned if the Siebel request document contains Japanese characters in J2CA. The same request works in BSE.

Workaround

This will be fixed in a future release.

1.6.6 HTTPS Protocol

Oracle Application Adapter for Siebel does not support the HTTPS protocol for services and events.

Workaround

None.

1.6.7 Multi-Value Groups

Oracle Application Adapter for Siebel does not support Multi-Value Groups (MVG) with join specifications.

Workaround

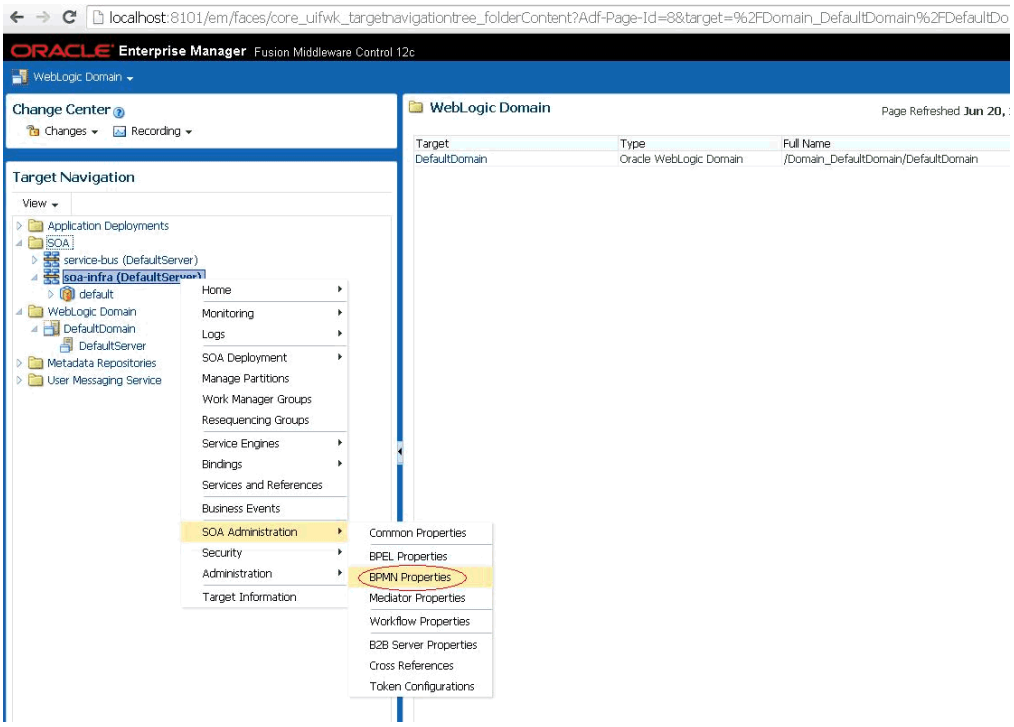
None.

1.6.8 BPMN Properties Adjustment

For inbound processing, if the size of the XML response event message is close to 1 MB or greater, then perform the following steps.

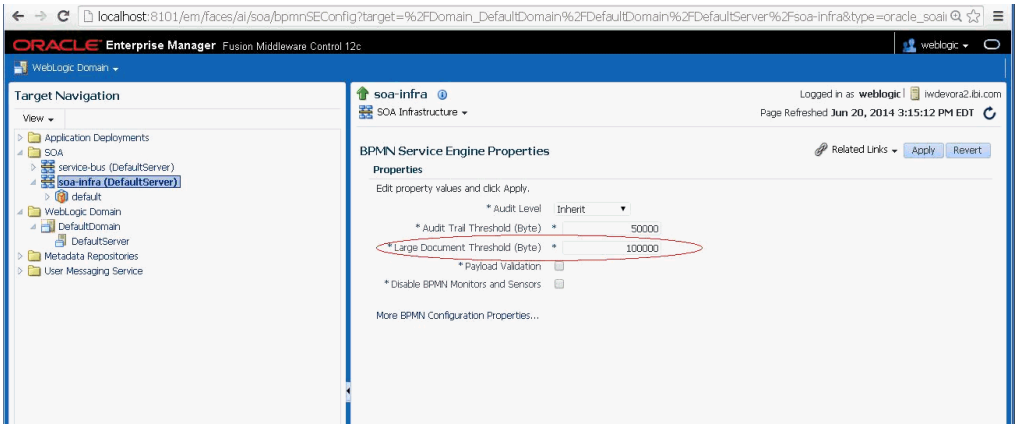
1. Log on to the Enterprise Manager console.
2. Right-click **soa-infra (DefaultServer)** in the left pane, select **SOA Administration**, and then click **BPMN Properties**, as shown in Figure 1–5.

Figure 1–5 Enterprise Manager Console



3. In the BPMN Service Engine Properties pane, provide a value for Large Document Threshold (Byte) parameter, as shown in Figure 1–6.

Figure 1–6 BPMN Service Engine Properties Pane



1.7 Oracle Application Adapter for J.D. Edwards OneWorld: Issues and Workarounds

The following topic discusses issues that pertain to Oracle Application Adapter for J.D. Edwards OneWorld.

- Section 1.7.1, "Platform Support"
- Section 1.7.2, "J.D. Edwards OneWorld Unit Of Work (UOW)"

1.7.1 Platform Support

Oracle Application Adapter 12*c* (12.2.1.0.0) for J.D. Edwards OneWorld is certified only with Windows and Oracle Enterprise Linux platforms. If support on any other platforms is required for this adapter, then contact Oracle customer support.

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

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

1. Oracle recommends you to generate individual business functions of J.D. Edwards OneWorld and then group them together.
2. Generating the individual business functions and grouping them together is completely based on the experience and knowledge related to the business functions of J.D. Edwards OneWorld. There is no documentation on generating the UOW.
3. To create the XML Schema Document (XSD) files that can be used for UOW, perform the following steps:
   a. Trigger the appropriate event from the J.D. Edwards OneWorld GUI and generate an XML output file based on the event.
   b. Using the XML file, create an XSD file using an XML editor, such as XMLSPY.
   c. When creating the XSD, make sure that the XSD satisfies the SOA 11*g* namespace requirements. Manually add the namespace, target namespace, and other items that are required for SOA 11*g*.
4. Once generated, copy the XSD files for the UOW into the repository folder. This repository folder is automatically configured on your file system when a J.D. Edwards OneWorld target is created using Application Explorer.

2

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

Note: This chapter describes how to configure an outbound and inbound process for Oracle Service Bus using sbconsole for Oracle Application Adapter for SAP R/3.

All of the steps in this chapter are similar for Oracle Application Adapter for Siebel, Oracle Application Adapter PeopleSoft, and Oracle Application Adapter J.D. Edwards OneWorld.

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

This chapter contains the following sections:

2.1 Overview of Application Adapter Integration with Oracle Service Bus

To integrate with Oracle Service Bus (OSB), Oracle Application Adapter for SAP R/3 must be deployed in the same Oracle WebLogic Server as OSB. The underlying adapter services must be exposed as WSDL files, which are generated during design

time in Oracle Adapter Application Explorer (Application Explorer) for both request-response (outbound) and event notification (inbound) services of the adapter.

2.2 Configuring an Outbound Process Using sbconsole (J2CA Configuration)

This section describes how to configure an outbound process using sbconsole for J2CA configurations.

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

`<ADAPTER_HOME>\etc\sample\SAP_Samples.zip\SAP_Samples\OSB\J2CA\SAP_Sample_J2CA_OSB_Outbound_Project`

This section includes the following topics:

- Section 2.2.1, "Starting Oracle Service Bus and Creating Project Folders"
- Section 2.2.2, "Setting the Class Path for Application Explorer to Integrate With Oracle Service Bus"
- Section 2.2.3, "Publishing a WSDL From Application Explorer to Oracle Service Bus"
- Section 2.2.4, "Configuring a WSDL-based Business Service"
- Section 2.2.5, "Configuring a File Type Business Service"
- Section 2.2.6, "Configuring a Pipeline With Proxy Service"

2.2.1 Starting Oracle Service Bus and Creating Project Folders

This section describes how to start Oracle Service Bus (OSB) and create project folders.

Perform the following steps to start Oracle Service Bus and create project folders:

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

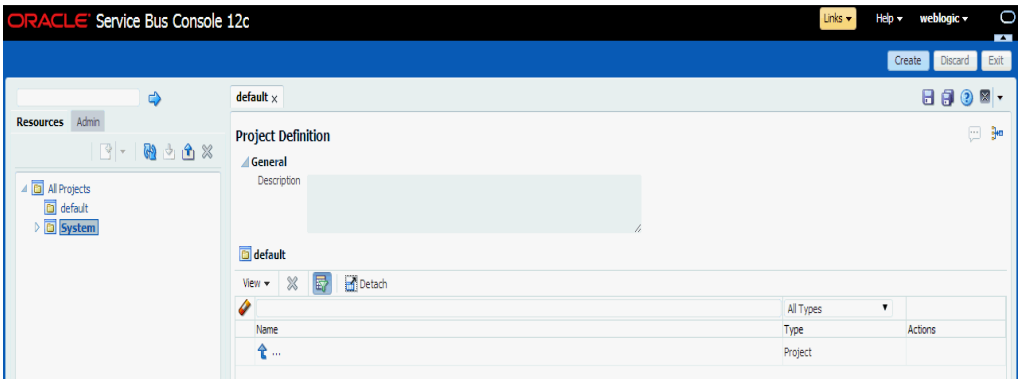
   `http://hostname:port/sbconsole`

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

   The Oracle Service Bus Console logon page is displayed.
3. Log on to the Oracle Service Bus Console using a valid user name and password.

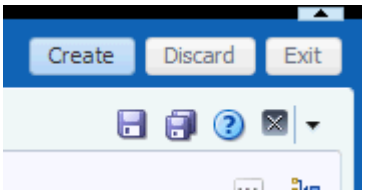
   The Oracle Service Bus Console home page is displayed, as shown in Figure 2–1.

Figure 2–1 Oracle Service Bus Console Home Page



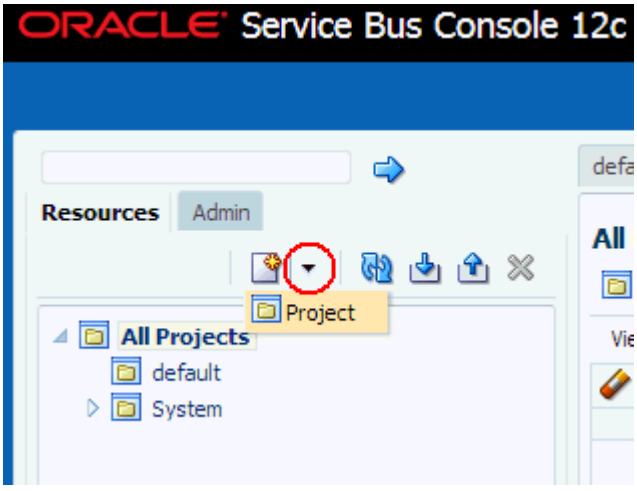
4. Click **Create** in the right pane of the Oracle Service Bus session, as shown in Figure 2–2.

Figure 2–2 Oracle Service Bus Session



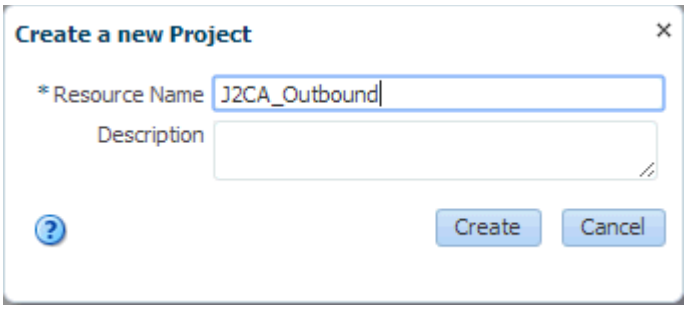
5. Select **All Projects**, click the down arrow in the left pane, and select **Project**, as shown in Figure 2–3.

Figure 2–3 All Projects Folder



The Create a new Project window is displayed, as shown in Figure 2–4.

Figure 2–4 Create New Project Window

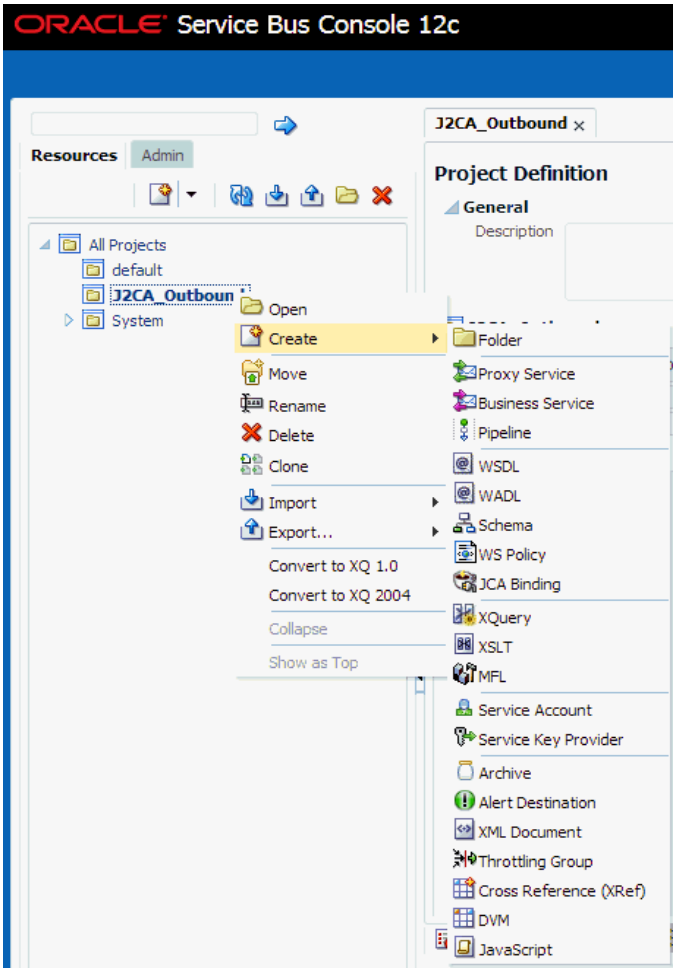


6. Provide a valid name for the new project (for example, J2CA_Outbound) in the Resource Name field, and click **Create**.

   The new project is successfully created and listed.

7. Right-click the newly created project, select **Create**, and click **Folder**, as shown in Figure 2–5.

Figure 2–5 Create Option



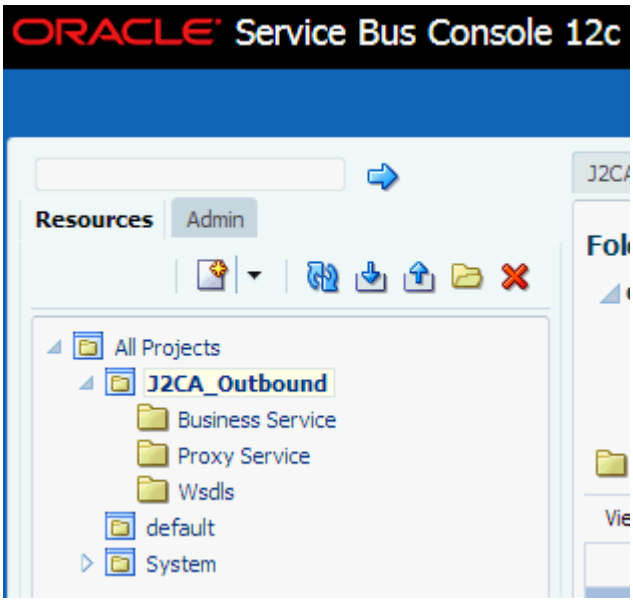
   The Create a new Folder window is displayed.

8. In the Resource Name field, type **Business Service** and click **Create**.

9. Repeat steps 7 and 8 to create folders with the names **Proxy Service** and **Wsdls**.

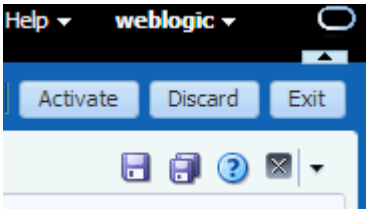
   The Business Service, Proxy Service, and Wsdls folders are listed in the left pane below the project node, as shown in Figure 2–6.

Figure 2–6 Project Node



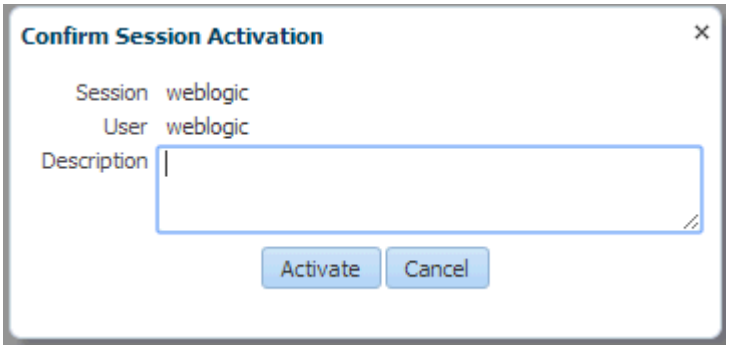
10. Click **Activate** in the right pane of the Oracle Service Bus session, as shown in Figure 2–7.

Figure 2–7 Activate Button



11. In the Confirm Session Activation page, click **Activate** to save the changes, as shown in Figure 2–8.

Figure 2–8 Confirm Session Activation Window



2.2.2 Setting the Class Path for Application Explorer to Integrate With Oracle Service Bus

Before starting and using Application Explorer to publish a WSDL directly to the Oracle Service Bus (OSB) Console (project/folder), OSB users must perform the following steps:

1. Open the command prompt window.
2. Navigate to the following directory:

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

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

   This command sets the class path for Application Explorer to access the Oracle WebLogic Server APIs to publish the WSDLs to the OSB Console.
4. Do not close the command prompt window.
5. Navigate to the following directory:

   `<ADAPTER_HOME>\tools\iwae\bin`

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

   You are now ready to publish WSDLs from Application Explorer to the OSB Console.

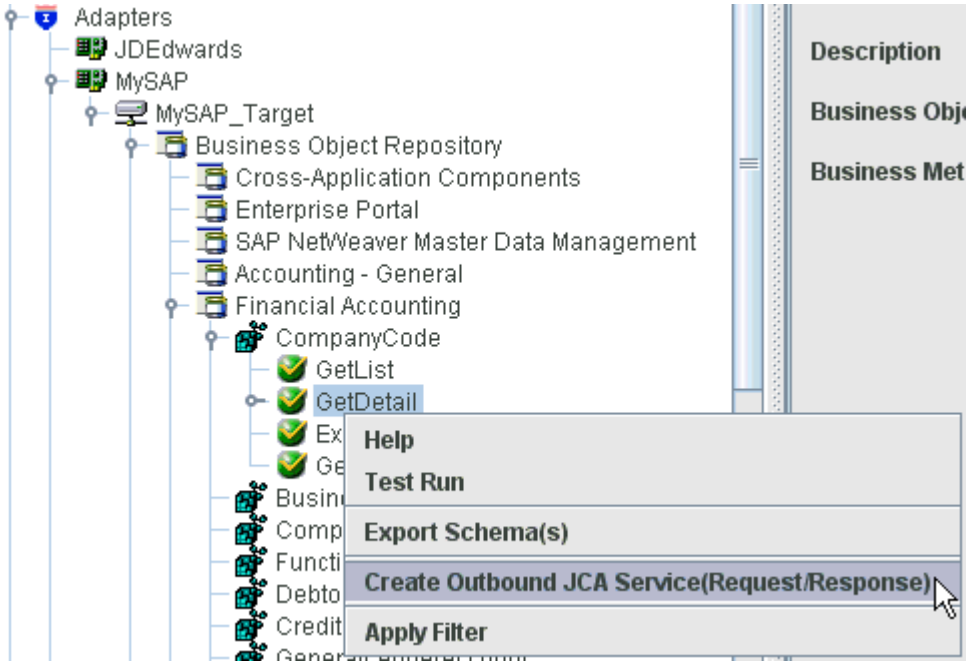
2.2.3 Publishing a WSDL From Application Explorer to Oracle Service Bus

Perform the following steps to publish a WSDL from Application Explorer to Oracle Service Bus:

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

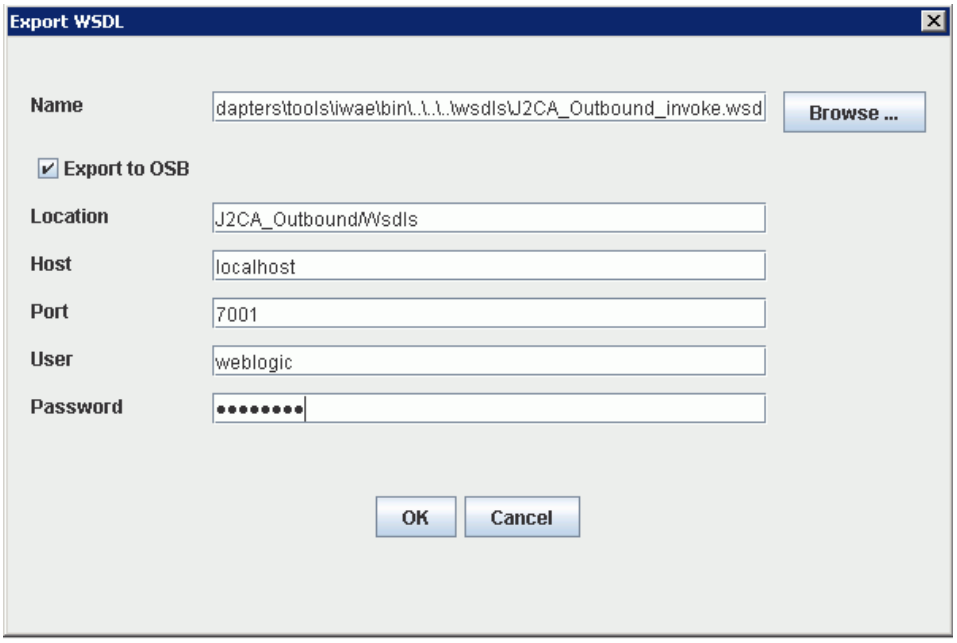
   For more information, see *Chapter 4, "Configuring Oracle Application Adapter for SAP R/3* in the *Oracle Fusion Middleware Application Adapter for SAP R/3 User's Guide for Oracle WebLogic Server*.
2. Expand **Business Object Repository**, **Financial Accounting**, and then **CompanyCode**.
3. Right-click the **GetDetail** method and select **Create Outbound JCA Service(Request/Response)** from the menu, as shown in Figure 2–9.

Figure 2–9 Create Outbound JCA Service (Request/Response) Option



The Export WSDL dialog is displayed, as shown in Figure 2–10.

Figure 2–10 Export WSDL Dialog



4. In the Name field, a default file name for the WSDL file is provided. You can accept the default or provide your own.
5. Select the **Export to OSB** option.
6. In the Location field, enter the folder name in Oracle Service Bus where you want to publish the WSDL document.

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

7. In the Host field, enter the name of the machine where Oracle Service Bus is installed.
8. In the Port field, enter the port that is being used by Oracle Service Bus.
9. In the User field, enter your username to access Oracle Service Bus.
10. In the Password field, enter your password to access Oracle Service Bus.
11. Click **OK**.

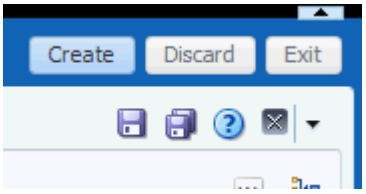
    The WSDL is published to the location specified in the Export WSDL dialog and is now available for use with a Business Service or Proxy Service in Oracle Service Bus.

2.2.4 Configuring a WSDL-based Business Service

Perform the following steps to configure a WSDL-based Proxy Service:

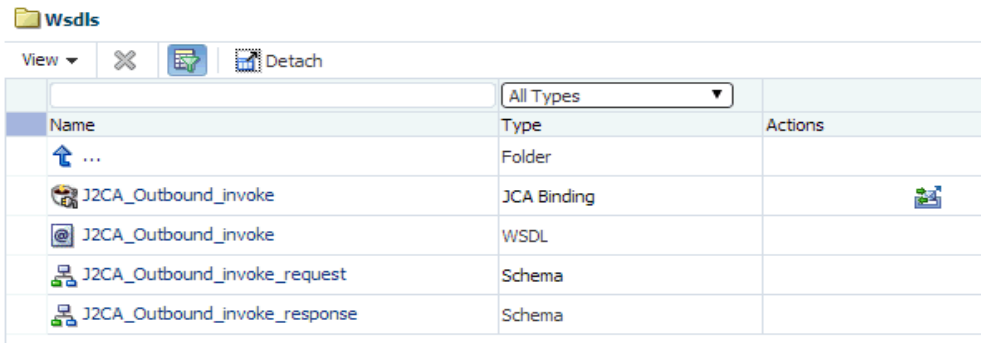
1. Open the Oracle Service Bus Console and click **Create** in the right pane of the Oracle Service Bus session, as shown in Figure 2–11.

Figure 2–11 Create Button



2. Double-click the created WSDL folder in the left pane (for example, Wsdls) and ensure that the exported WSDL is listed in the right pane, as shown in Figure 2–12.

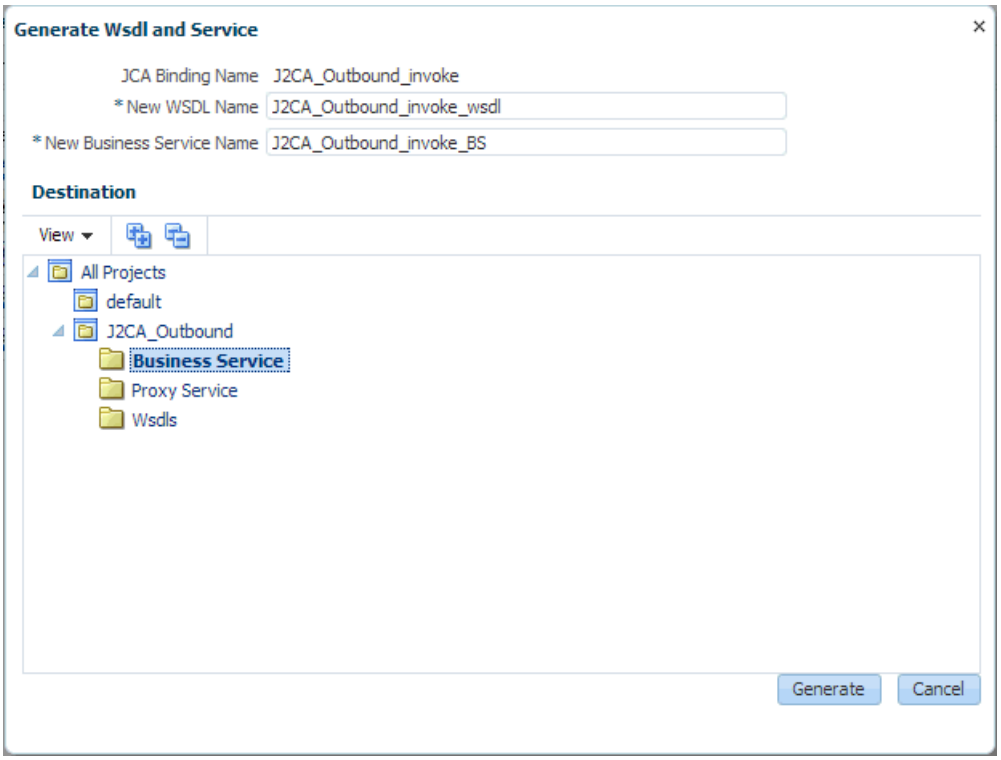
Figure 2–12 Wsdls Folder



3. Click the icon that corresponds to the JCA Binding in the Actions column.

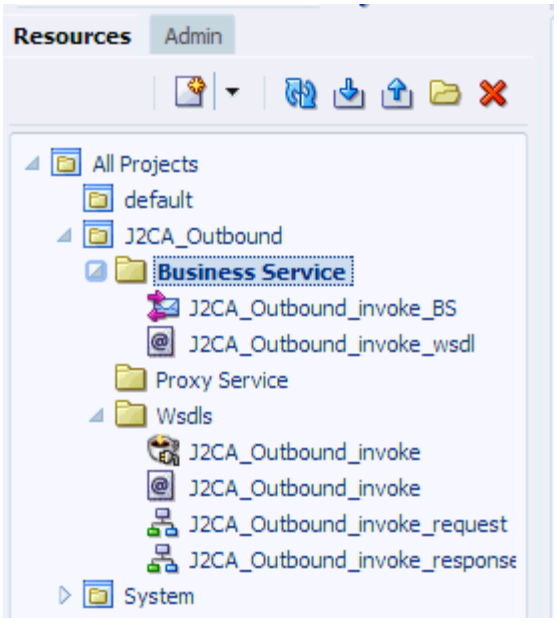
   The Generate WSDL and Service window is displayed, as shown in Figure 2–13.

Figure 2–13 Generate WSDL and Service Window



4. Provide a new WSDL name and a new Business Service name in the corresponding fields.
5. In the Destination area, select an available project and the sub-folder that is designated for Business Services.
6. Click **Generate**.
7. Expand **Business Service** under the project folder and check if the generated WSDL and Business Service are listed, as shown in Figure 2–14.

Figure 2–14 Business Service Folder

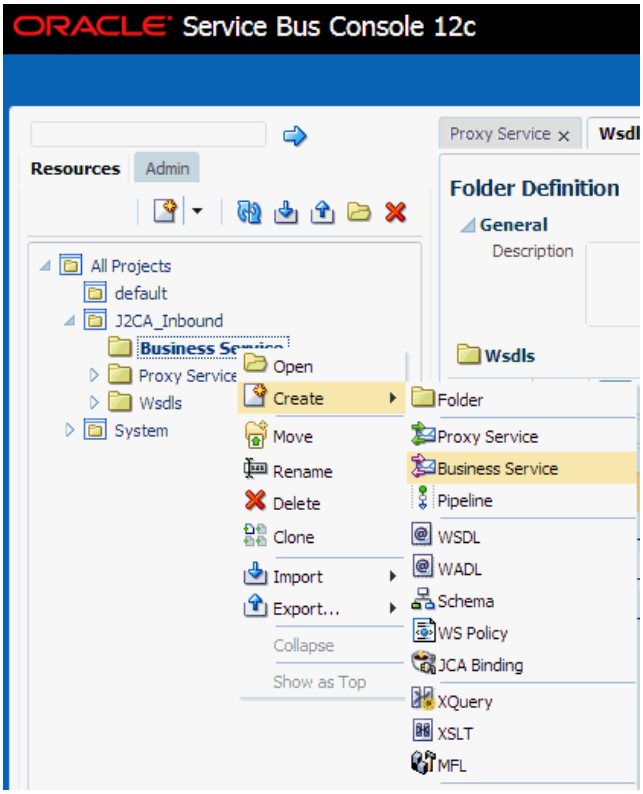


2.2.5 Configuring a File Type Business Service

Perform the following steps to configure a File type Business Service:

1. Right-click the **Business Service** folder you created in the left pane, select **Create**, and click **Business Service** as shown in Figure 2–15.

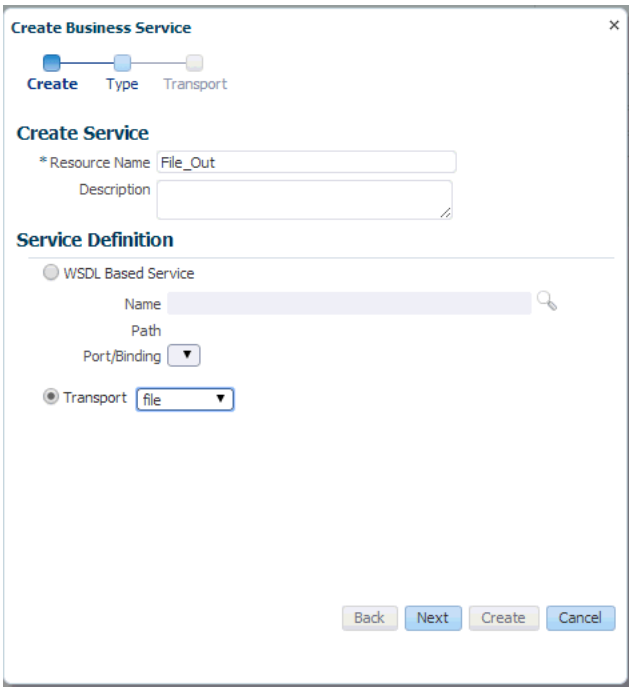
Figure 2–15 Business Service Folder



The Create Business Service window is displayed.

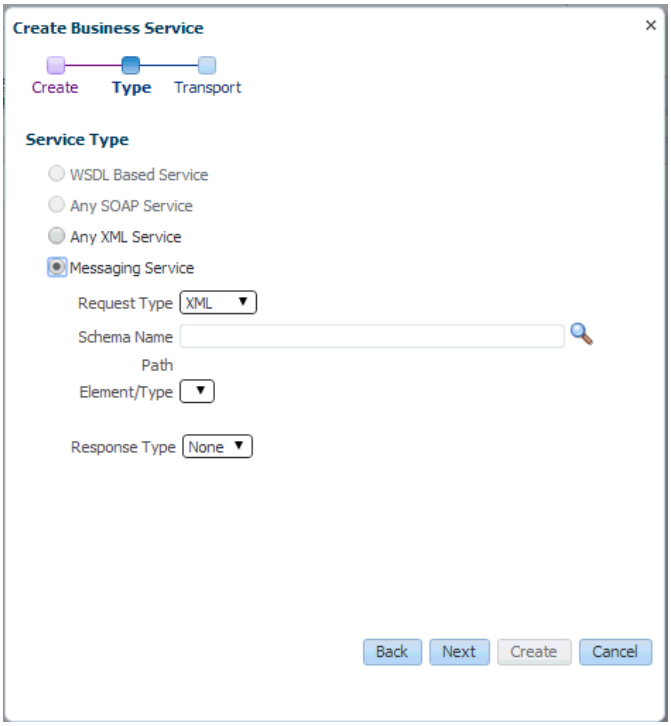
2. In the Resource Name field, provide a name for the Business Service, select the **File** option in the Transport section under Service Definition, and click **Next**, as shown in Figure 2–16.

Figure 2–16 Service Definition



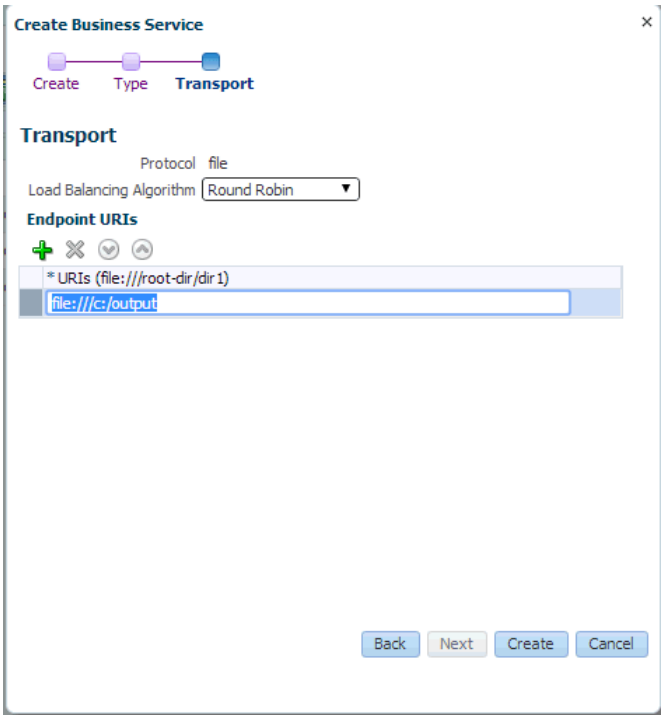
3. In the Service Type section, select **Messaging Service**. By default, the Request Type is set to XML, and the Response Type is set to None. Then click **Next**, as shown in Figure 2–17.

Figure 2–17 Service Type Configuration Page



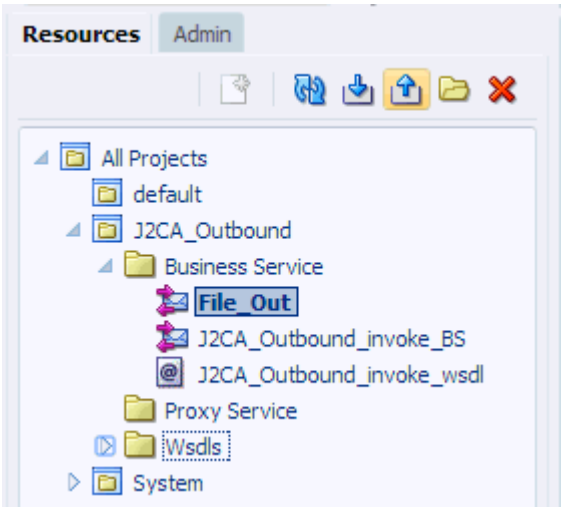
4. Enter the path to a destination folder on your file system in the Endpoint URI field.
5. Click **Create**, as shown in Figure 2–18.

Figure 2–18 Transport Page



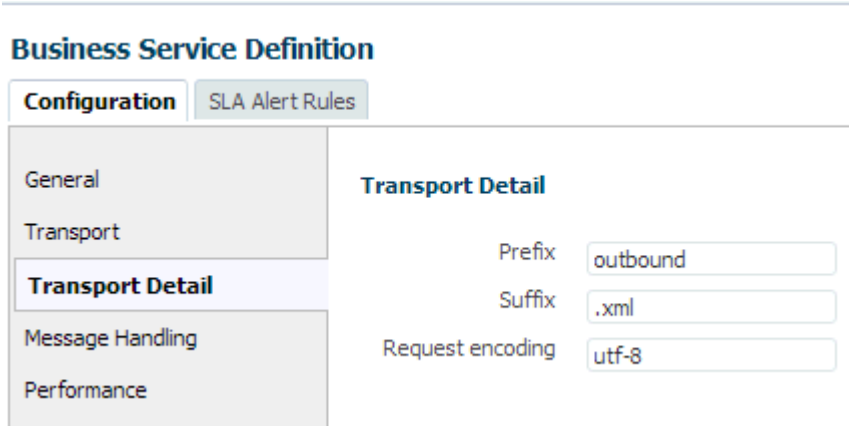
The Business Service **File_Out** is created and listed under Business Service, as shown in Figure 2–19.

Figure 2–19 File_Out Business Service



6. Double-click **File_Out**, click **Transport Detail** in the left pane, and enter the prefix and suffix for the output file to be received, as shown in Figure 2–20.

Figure 2–20 Transport Detail



7. Click the **Save** or **Save All** icon in the right corner, as shown in Figure 2–21.

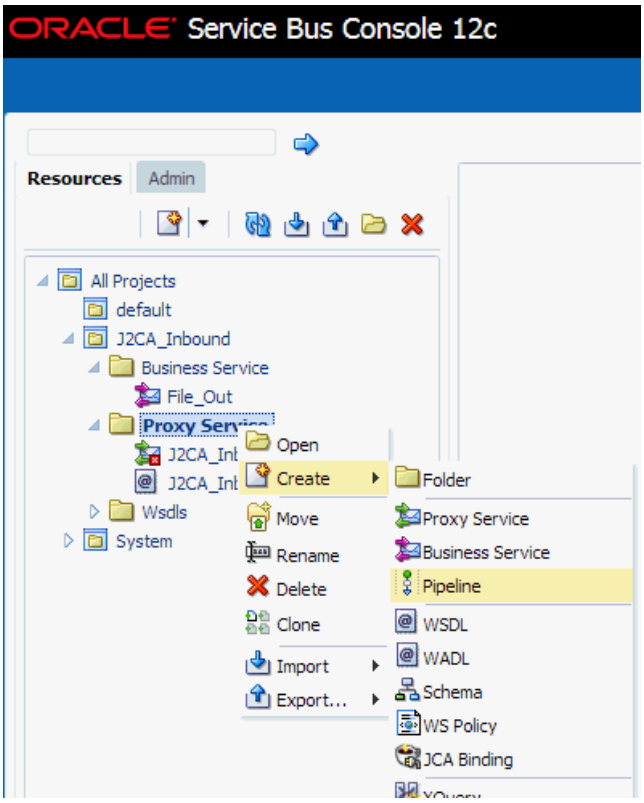
Figure 2–21 Save/Save All Icons

2.2.6 Configuring a Pipeline With Proxy Service

Perform the following steps to configure a Pipeline:

1. Right-click the Proxy Service folder, select **Create** and click **Pipeline**, as shown in Figure 2–22.

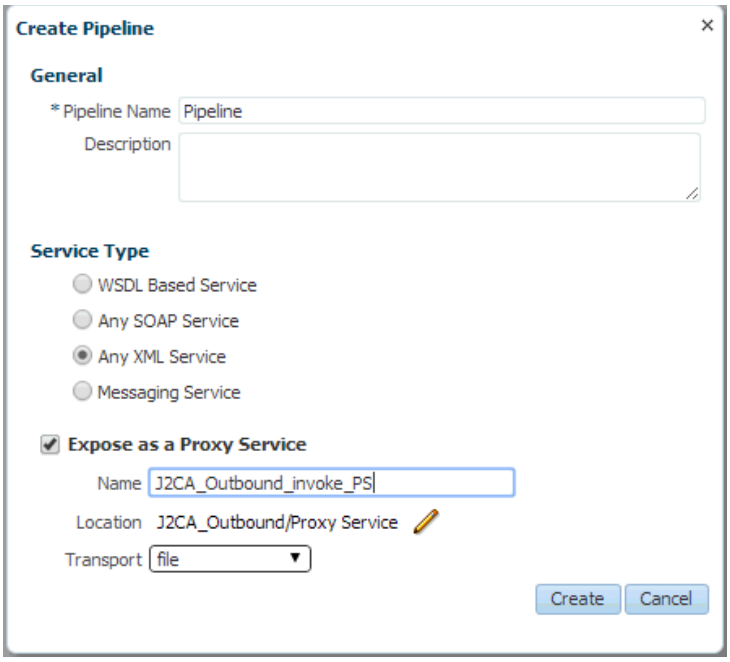
Figure 2–22 Pipeline Option



The Create Pipeline window is displayed.

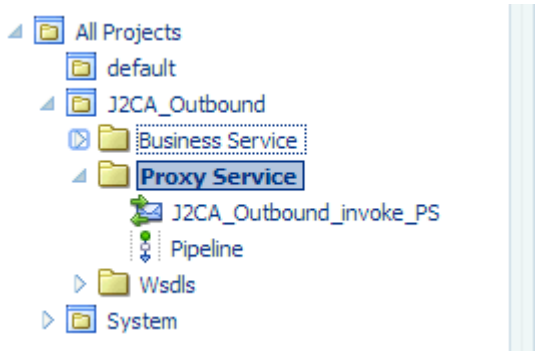
2. Enter a name in the Pipeline Name field. By default, **Expose as a Proxy Service** is selected. If you wish to change the Proxy Service Name, change it and set Transport as **file**, and click **Create** as shown in Figure 2–23.

Figure 2–23 Create Pipeline Window



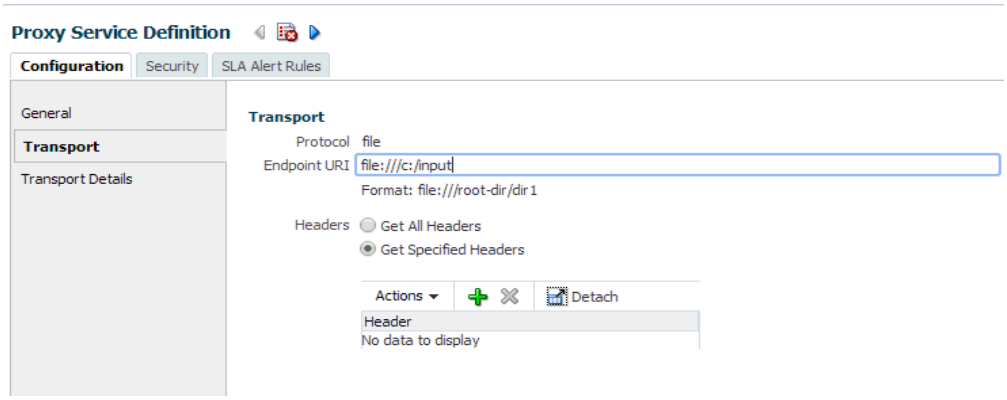
The created Pipeline and the Proxy Service is listed under Proxy Service, as shown in Figure 2–24.

Figure 2–24 Pipeline Node



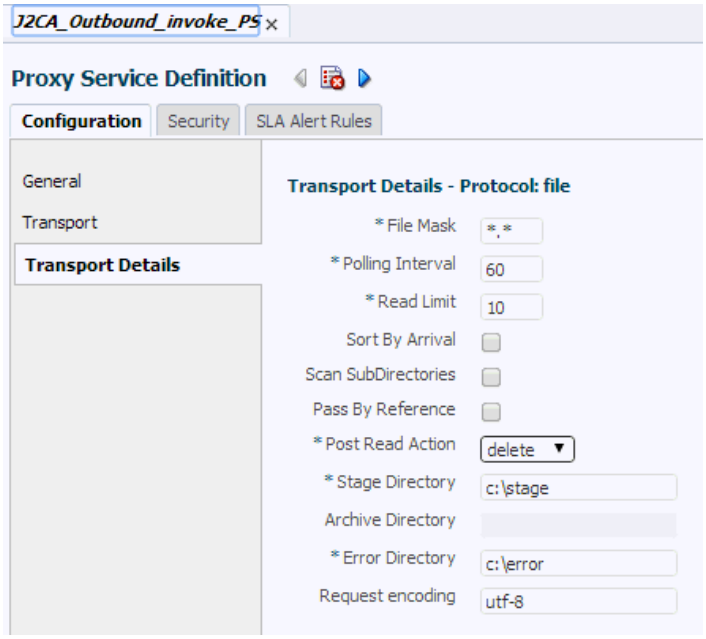
3. Double-click the created proxy service and click **Transport** in the left pane. Provide the input location in the Endpoint URI field, as shown in Figure 2–25.

Figure 2–25 Transport



4. Click **Transport Details** in the left pane and provide the location for the Stage Directory and the Error Directory fields, as shown in Figure 2–26.

Figure 2–26 Transport Details



5. Click the **Save All** icon in the right corner, as shown in Figure 2–27.

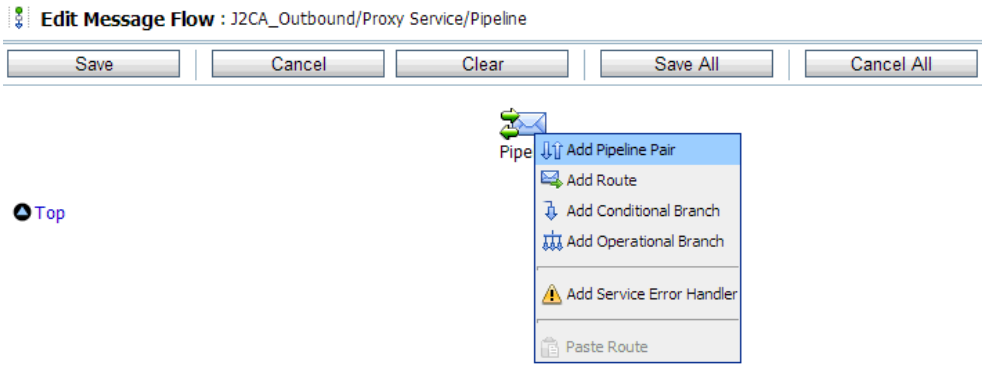
Figure 2–27 Save All Icon

6. Double-click the **Pipeline** node and click the **Open Message Flow** icon on the right pane to open the message flow, as shown in Figure 2–28.

Figure 2–28 Open Message Flow Icon

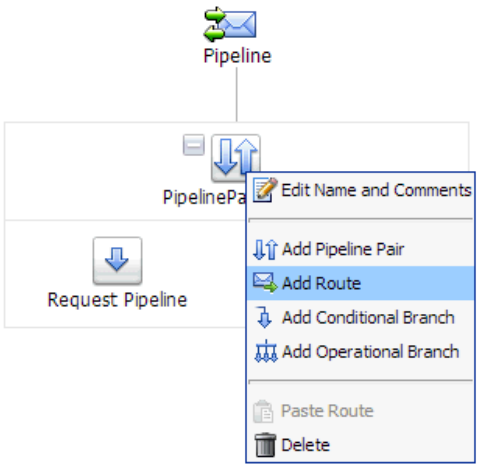
7. Click the Proxy Service icon and select **Add Pipeline Pair** from the menu, as shown in Figure 2–29.

Figure 2–29 Add Pipeline Pair Option



8. Click the **PipelinePairNode1** icon and select **Add Route** from the menu, as shown in Figure 2–30.

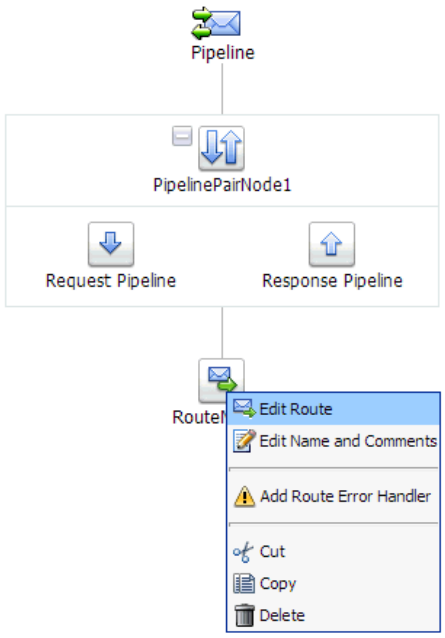
Figure 2–30 Add Route Option



The RouteNode1 icon is added below the PipelinePairNode1 icon.

9. Click the RouteNode1 icon and select **Edit Route** from the menu, as shown in Figure 2–31.

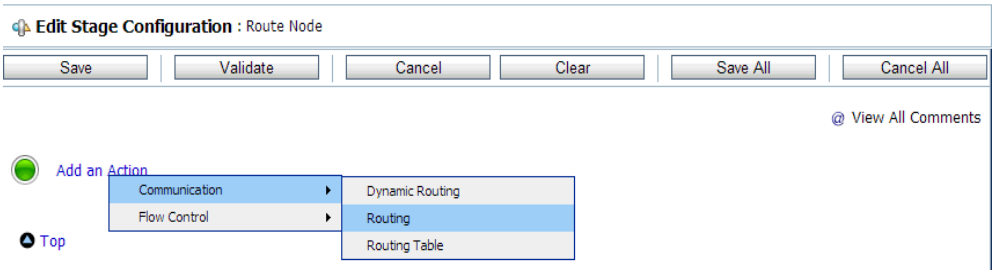
Figure 2–31 Edit Route Option



The Edit Stage Configuration workspace area is displayed.

10. Click **Add an Action**, select **Communication** and click **Routing**, as shown in Figure 2–32.

Figure 2–32 Edit Stage Configuration Workspace Area



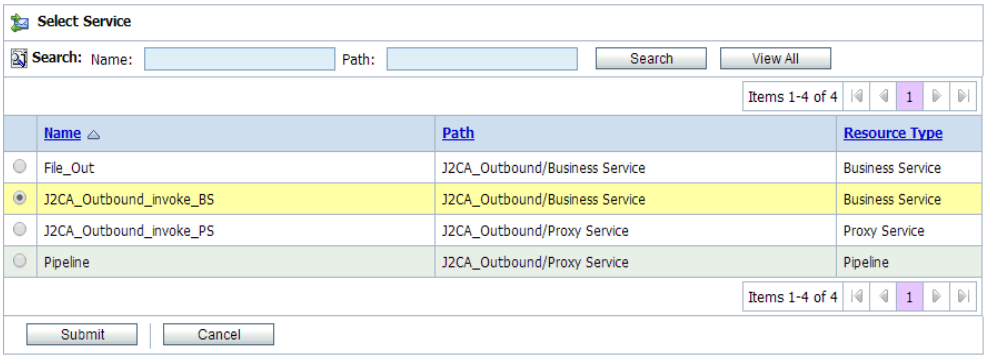
11. Click **<Service>**, as shown in Figure 2–33.

Figure 2–33 Actions

The Select Service dialog is displayed.

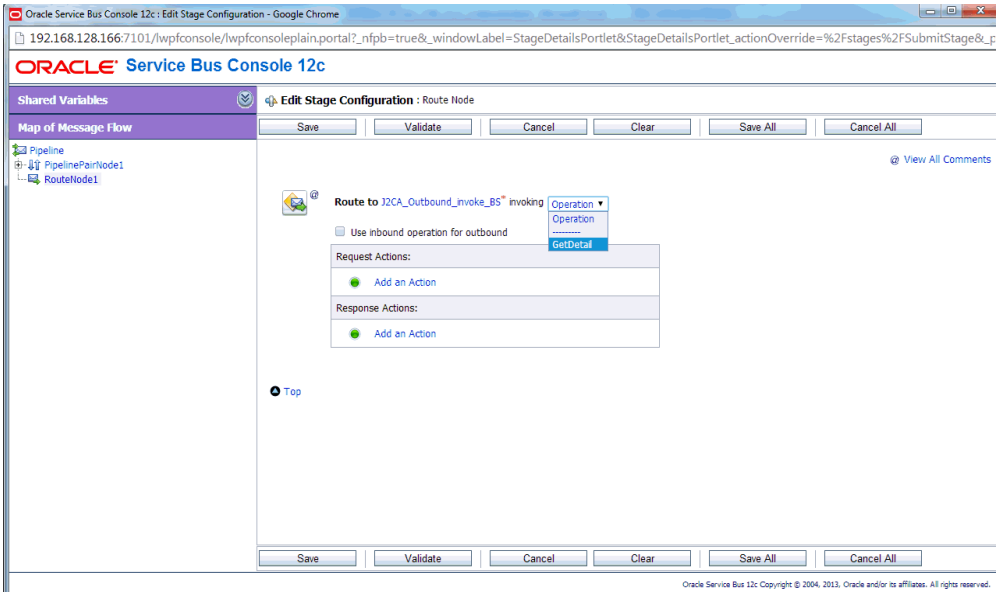
12. Select the WSDL type Business Service configured for SAP R/3 and click on **Submit**, as shown in Figure 2–34.

Figure 2–34 Select Service Dialog



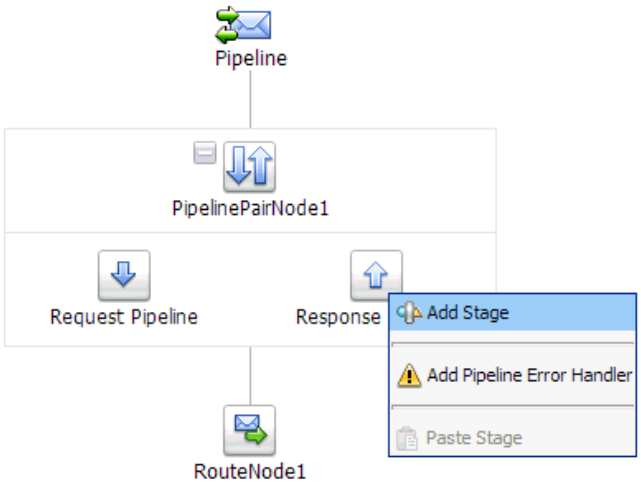
13. Select the name of the SAP R/3 business object (for example, GetDetail) as the operational attribute from the list, and click **Save**, as shown in Figure 2–35.

Figure 2–35 Edit Stage Configuration



14. Click the Response Pipeline icon and select **Add Stage** from the menu, as shown in Figure 2–36.

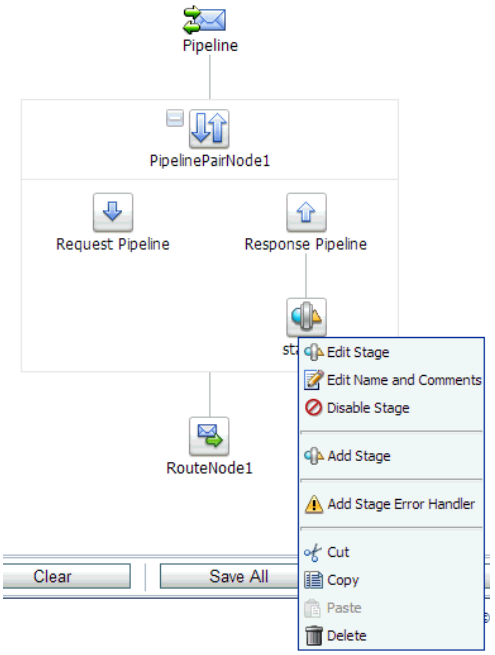
Figure 2–36 Response Pipeline Icon



The Stage1 icon is added below the Response Pipeline icon.

15. Click the Stage1 icon and select **Edit Stage** from the menu, as shown in Figure 2–37.

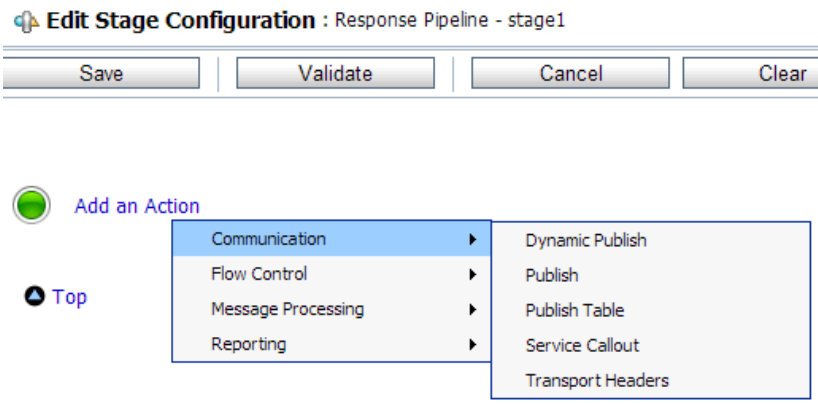
Figure 2–37 Edit Stage Option



The Edit Stage Configuration workspace area is displayed.

16. Click **Add an Action**, select **Communication**, and then click **Publish**, as shown in Figure 2–38.

Figure 2–38 Edit Stage Configuration Workspace Area

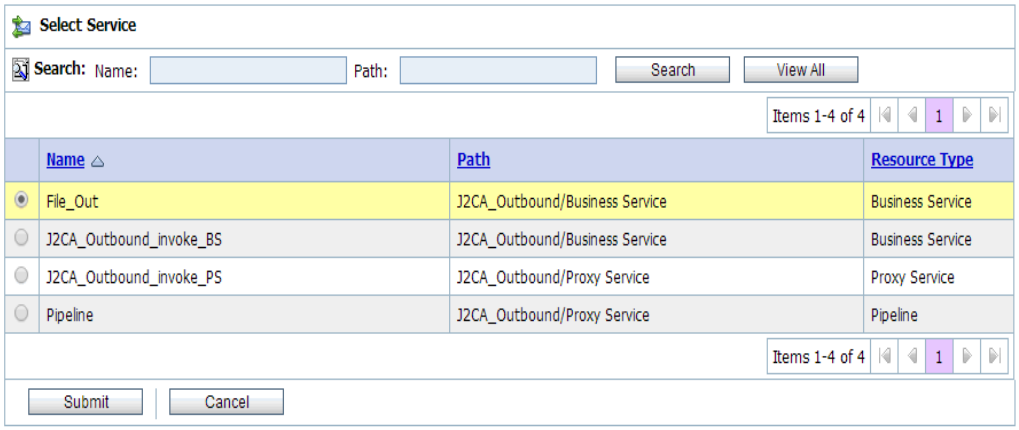


17. Click **<Service>**, as shown in Figure 2–39.

Figure 2–39 <Service> Action

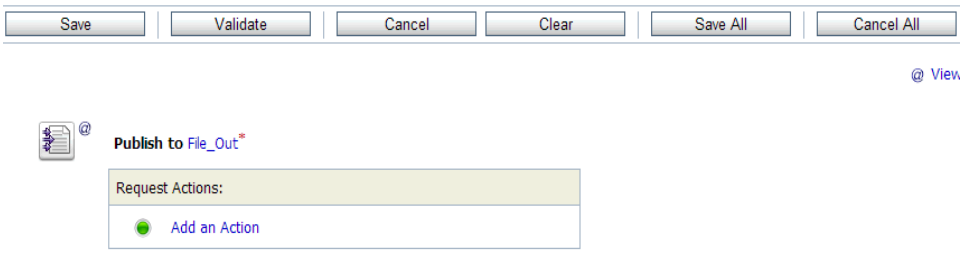
18. In the Select Service dialog, select a File type Business Service and click **Submit**, as shown in Figure 2–40.

Figure 2–40 Select Service Dialog



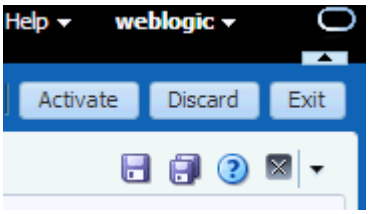
19. Click **Save All**, as shown in Figure 2–41.

Figure 2–41 Save All Button



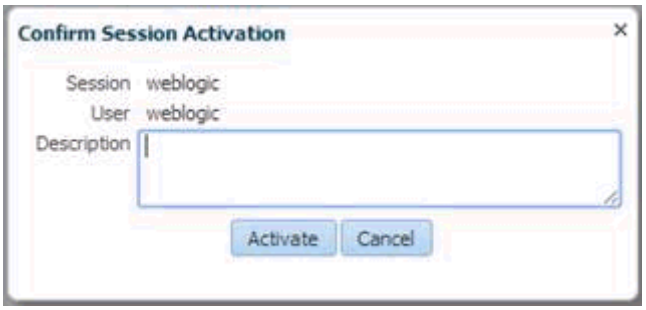
20. Click **Activate** in the right pane of the Oracle Service Bus session, as shown in Figure 2–42.

Figure 2–42 Activate Button



21. Click **Activate** to save the changes, as shown in Figure 2–43.

Figure 2–43 Confirm Session Activation



22. Copy and paste an input XML file in the input folder you have configured (for example, C:\input). Output is received in the configured output location (for example, C:\output).

2.3 Configuring an Inbound Process Using sbconsole (J2CA Configuration)

This section describes how to configure an inbound process using sbconsole for J2CA configurations.

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

`<ADAPTER_HOME>\etc\sample\SAP_Samples.zip\SAP_Samples\OSB\J2CA\SAP_Sample_J2CA_OSB_Inbound_Project`

This section includes the following topics:

2.3.1 Starting Oracle Service Bus and Creating Project Folders

For more information on starting Oracle Service Bus and creating project folders, see Section 2.2.1, "Starting Oracle Service Bus and Creating Project Folders" on page 2-2.

2.3.2 Setting the Class Path for Application Explorer to Integrate With Oracle Service Bus

For more information on setting the class path for Application Explorer to integrate with Oracle Service Bus, see Section 2.2.2, "Setting the Class Path for Application Explorer to Integrate With Oracle Service Bus" on page 2-6.

2.3.3 Publishing a WSDL From Application Explorer to Oracle Service Bus

Perform the following steps to publish a WSDL from Application Explorer to Oracle Service Bus:

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

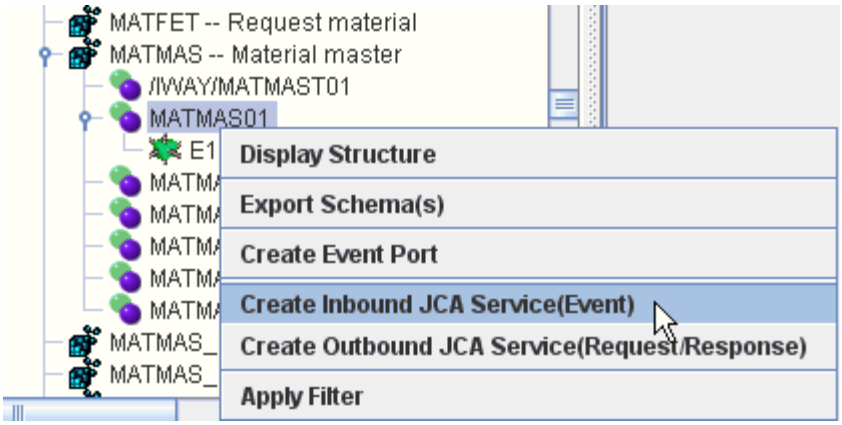
   For more information, see *Chapter 4, "Configuring Oracle Application Adapter for SAP R/3* in the *Oracle Fusion Middleware Application Adapter for SAP R/3 User's Guide for Oracle WebLogic Server*.

2. Create a SAP R/3 channel.

   For more information, see *Section 4.10, "Configuring an Event Adapter"* on page 4-22 in the *Oracle Fusion Middleware Application Adapter for SAP R/3 User's Guide for Oracle WebLogic Server*.

3. Expand the **ALE (IDOCs)** node and then the **MATMAS -- Material master** node in the target.

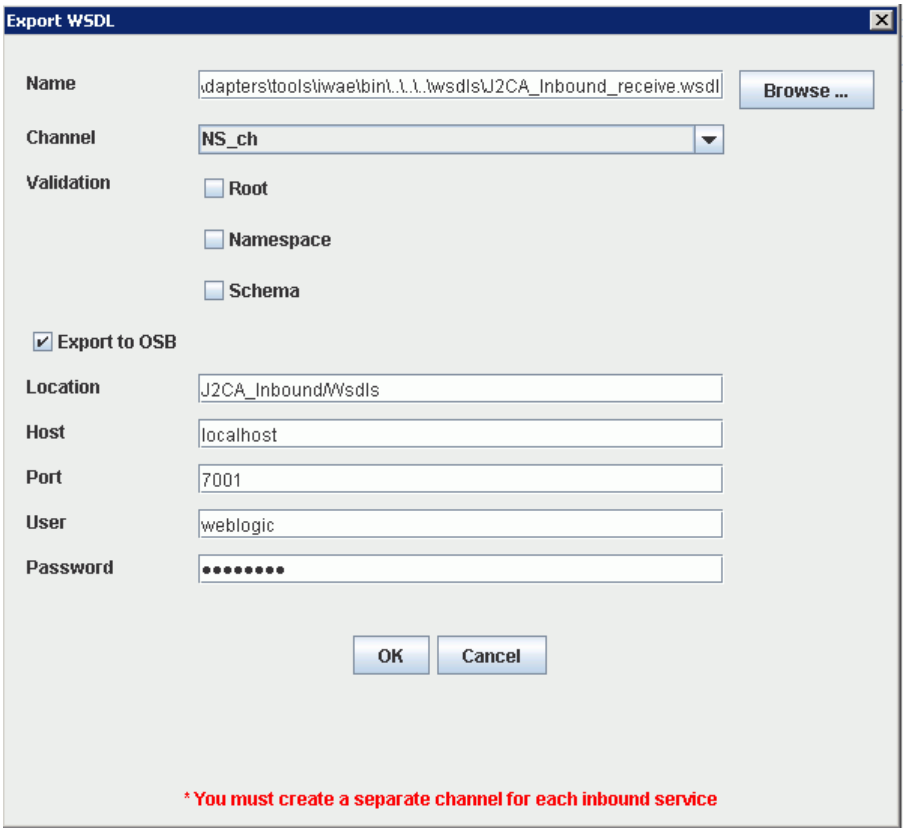
Figure 2–44 Create Inbound JCA Service(Event) Option



4. Right-click **MATMAS01** and select **Create Inbound JCA Service(Event)** from the menu, as shown in Figure 2–44.

   The Export WSDL dialog is displayed, as shown in Figure 2–45.

Figure 2–45 Export WSDL Dialog



5. In the Name field, a default file name for the WSDL file is provided. You can accept the default or provide your own.
6. From the Channel list, select the channel you created for this inbound service.
7. Three check boxes for Root, Namespace, and Schema validation are also available. Selection of multiple validation options is allowed.

8. Select the **Export to OSB** option.
9. In the Location field, enter the folder name in Oracle Service Bus where you want to publish the WSDL document.

   The location is composed of an Oracle Service Bus project name and optionally, one or more folder names. The project name and any folder names must be separated by a forward slash character "/".
10. In the Host field, enter the name of the machine where Oracle Service Bus is installed.
11. In the Port field, enter the port that is being used by Oracle Service Bus.
12. In the User field, enter your username to access Oracle Service Bus.
13. In the Password field, enter your password to access Oracle Service Bus.
14. Click **OK**.

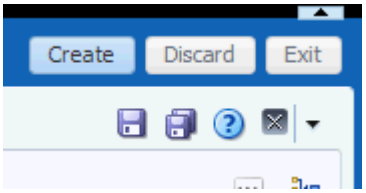
    The inbound WSDL is published to the location specified in the Export WSDL dialog and is now available for use with a Proxy Service in Oracle Service Bus.

2.3.4 Configuring a WSDL-based Proxy Service

Perform the following steps to configure a WSDL-based Proxy Service:

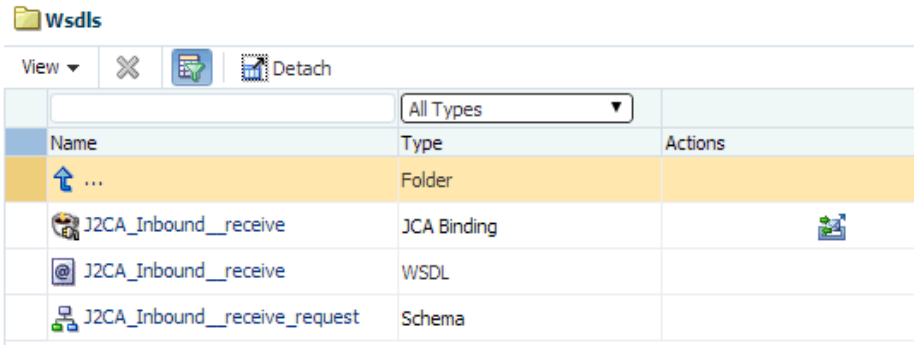
1. Open the Oracle Service Bus Console and click **Create** in the right pane of the Oracle Service Bus session, as shown in Figure 2–46.

Figure 2–46 Create Button



2. Double-click the created WSDL folder in the left pane (for example, Wsdls), and ensure that the exported WSDL is listed in the right pane, as shown in Figure 2–47.

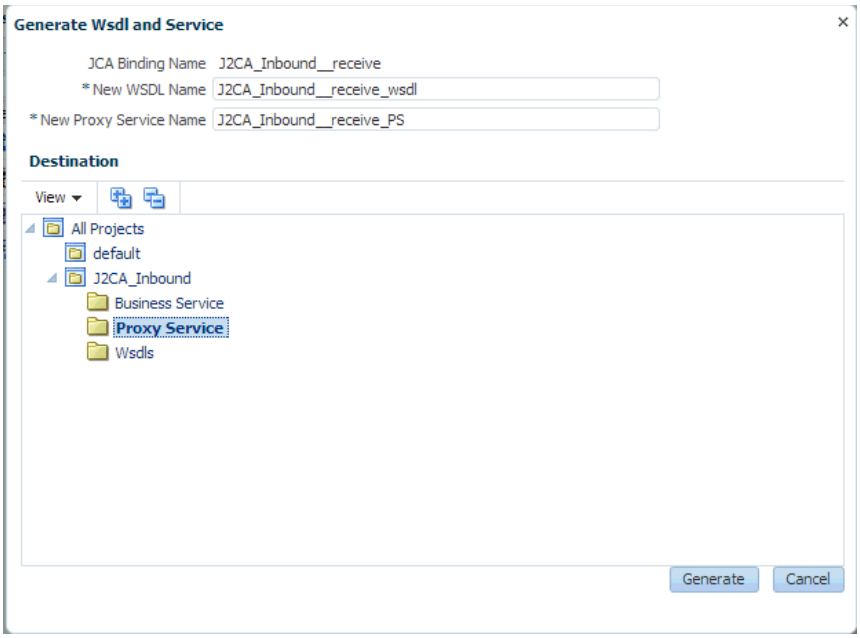
Figure 2–47 Exported WSDL



3. Click the icon that corresponds to the JCA Binding in the Actions column.

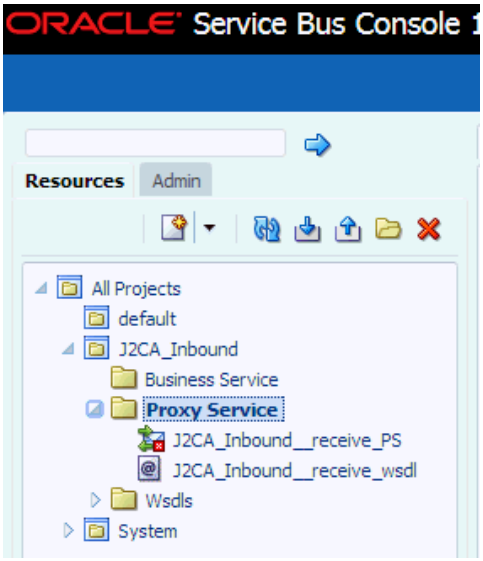
   The Generate WSDL and Service page is displayed, as shown in Figure 2–48.

Figure 2–48 Generate WSDL and Service Page



4. Provide a new WSDL name and a new Proxy Service name in the corresponding fields.
5. In the Destination area, select an available project and the sub-folder that is designated for Proxy Services.
6. Click **Generate**.
7. Expand **Proxy Service** under Project Explorer and check if the generated WSDL and Proxy Service are listed, as shown in Figure 2–49.

Figure 2–49 Generated WSDL

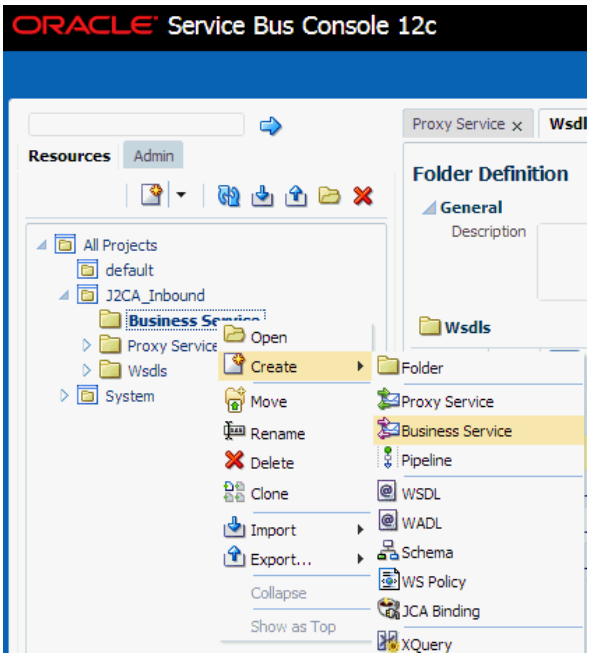


2.3.5 Configuring a File Type Business Service

Perform the following steps to configure a File type Business Service:

1. Right-click the Business Service folder you created in the left pane, select **Create**, and click **Business Service**, as shown in Figure 2–50.

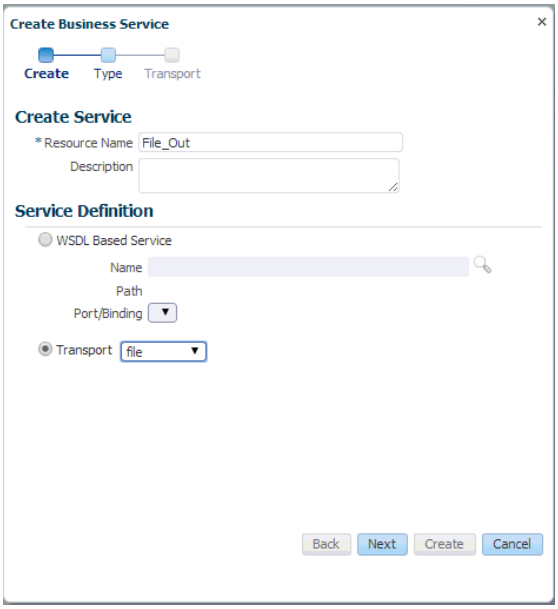
Figure 2–50 Business Service Folder



The Create Business Service window is displayed.

2. In the Resource Name field, provide a name for the Business Service and select the **File** option from the Transport drop-down list in the Service Definition area, as shown in Figure 2–51.

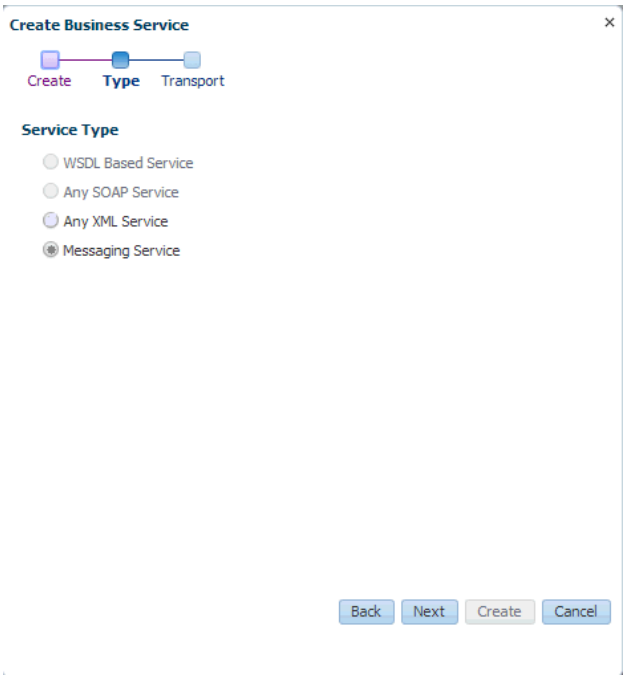
Figure 2–51 Create Business Service Window



3. Click **Next**.

4. In the Service Type area, select **Messaging Service** as the service type, as shown in Figure 2–52.

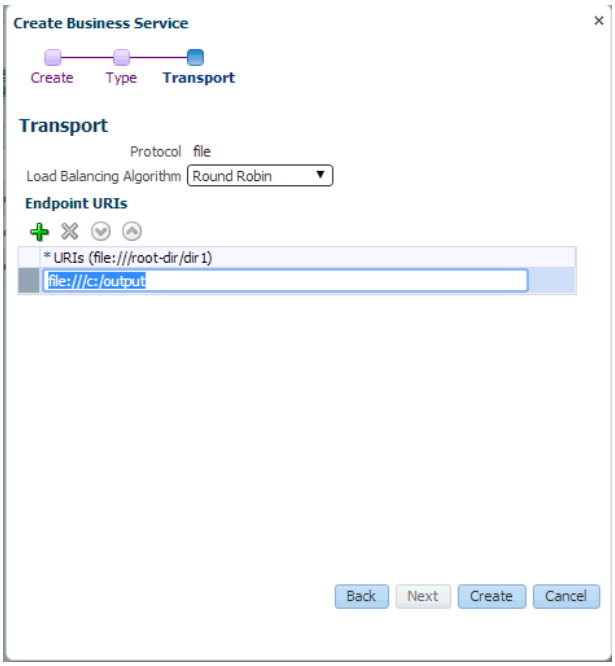
Figure 2–52 Service Type Area



5. Click **Next**.

   The Transport page is displayed, as shown in Figure 2–53.

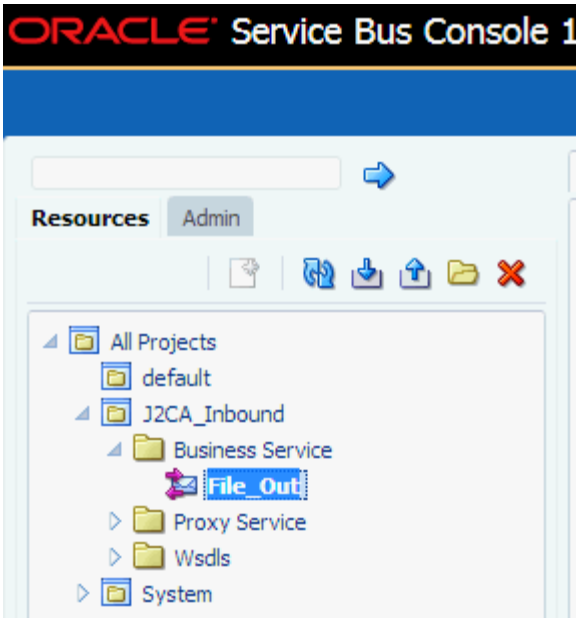
Figure 2–53 Transport Page



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

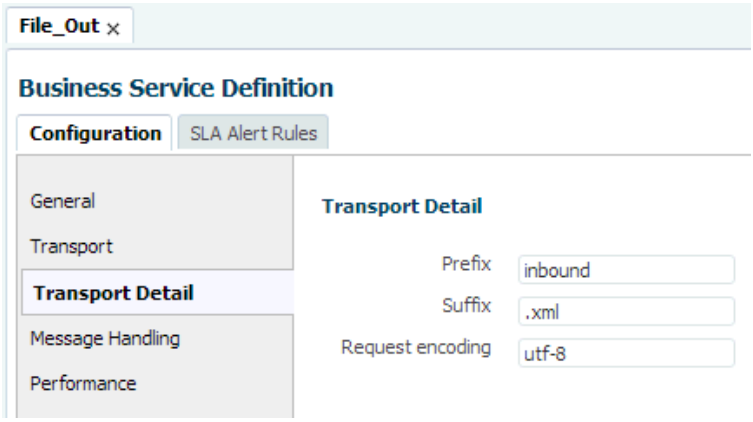
   The Business Service File_Out is created and listed under Business Service, as shown in Figure 2–54.

Figure 2–54 File_Out Business Service



7. Double-click **File_Out**, click **Transport Detail** in the left pane, and enter the prefix and suffix for the output file to be received, as shown in Figure 2–55.

Figure 2–55 Transport Detail Page



8. Click the Save or Save All icon in the right corner, as shown in Figure 2–56.

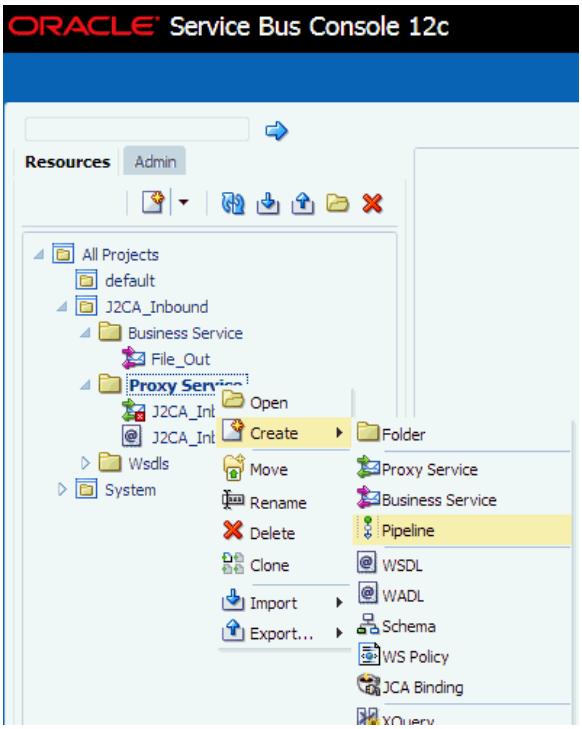
Figure 2–56 Save Icons

2.3.6 Configuring a Pipeline

Perform the following steps to configure a Pipeline:

1. Right-click the proxy service you created and select **Create**, and then click **Pipeline**, as shown in Figure 2–57.

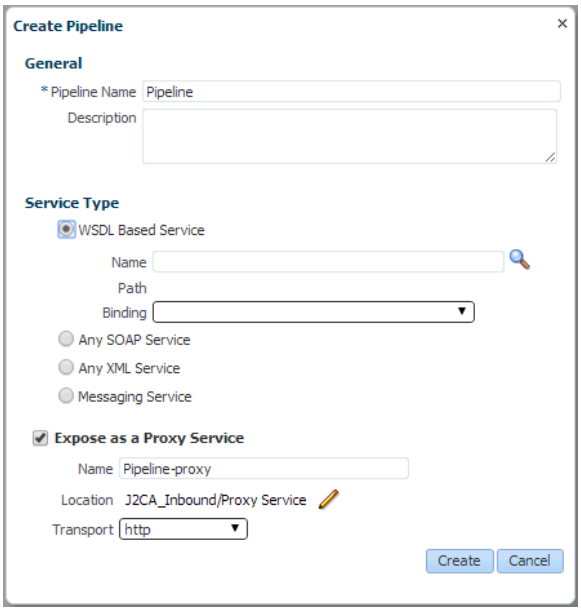
Figure 2–57 Pipeline Option



The Create Pipeline window is displayed.

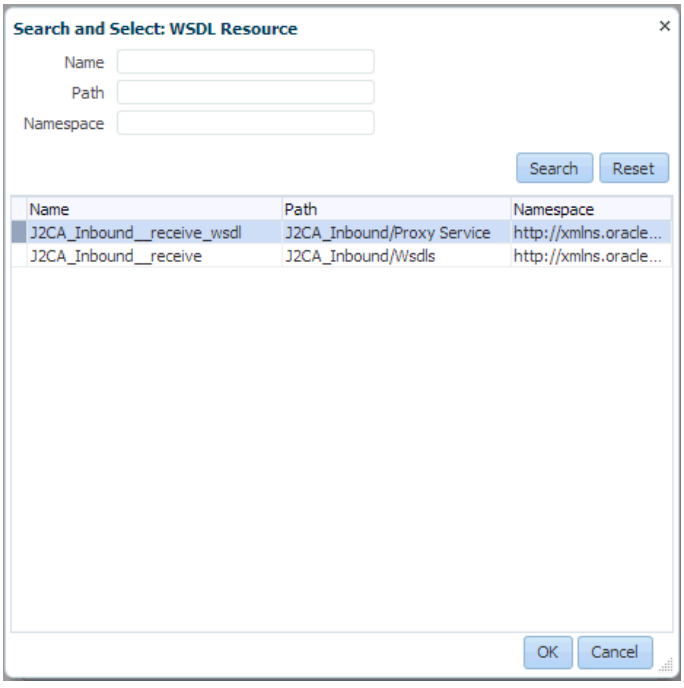
2. In the Pipeline Name field, enter a name and select the Service Type as **WSDL Based Service**, as shown in Figure 2–58.

Figure 2–58 Create Pipeline Window



3. Click the Search icon, and in the displayed Search and Select: WSDL Resource window, select **J2CA_Inbound_receive_wsdl**, and click **OK**, as shown in Figure 2–59.

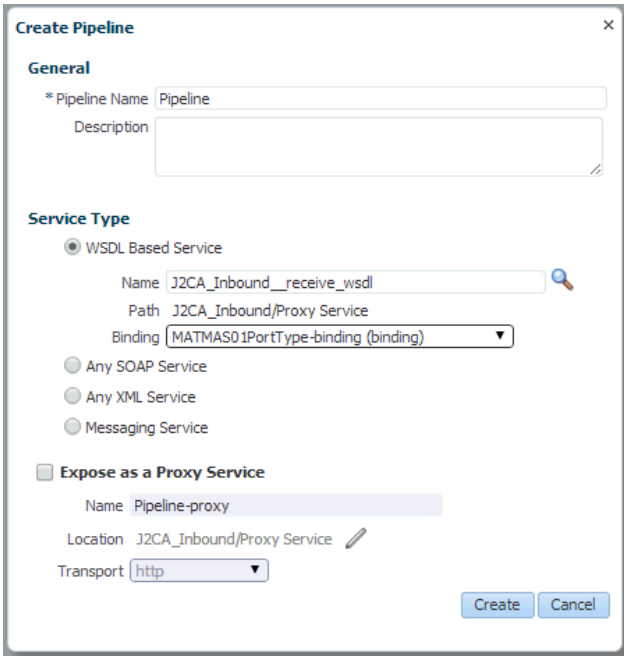
Figure 2–59 Search and Select: WSDL Resource Window



The Create Pipeline window opens.

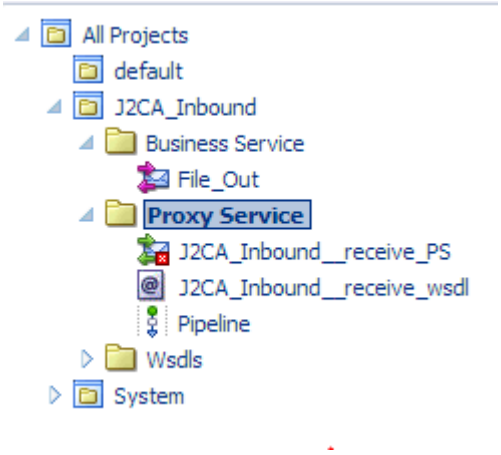
4. Clear the check box for **Expose as a Proxy Service**, and click **Create**, as shown in Figure 2–60.

Figure 2–60 Create Pipeline Window



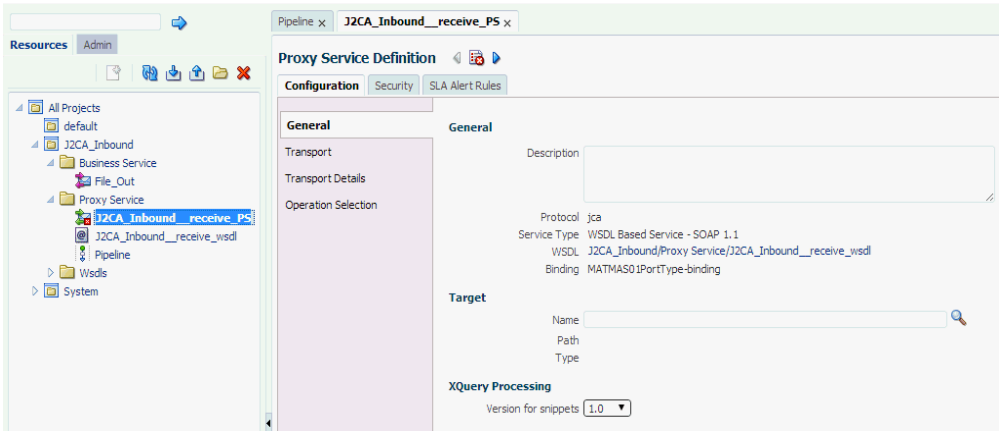
The pipeline is created and listed under Proxy Service, as shown in Figure 2–61.

Figure 2–61 Proxy Service Pipeline



5. Double-click the **J2CA_Inbound_receive_PS** node under Proxy Service in the left pane and click the **Search** icon in the Target area in right pane, as shown in Figure 2–62.

Figure 2–62 Proxy Service Definition Window

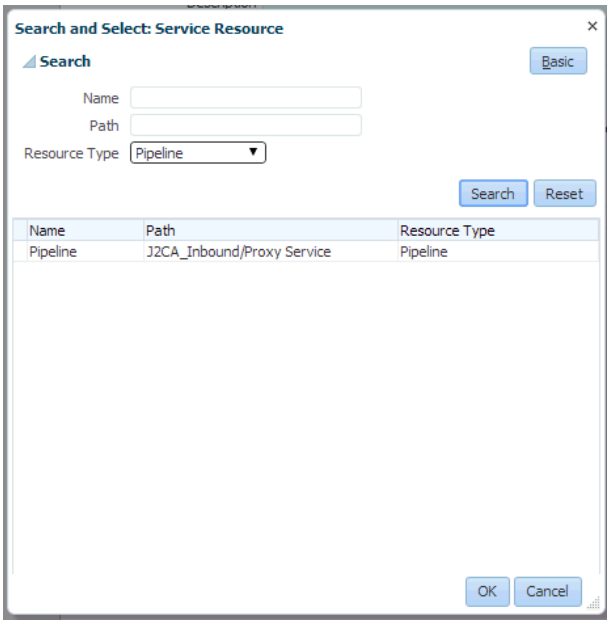


The Search and Select: Service Resource window appears.

6. From the Resource Type drop-down list, select **Pipeline** and then click the **Search** button.

   The Pipeline is listed, as shown in Figure 2–63.

Figure 2–63 Search and Select: Service Resource Window

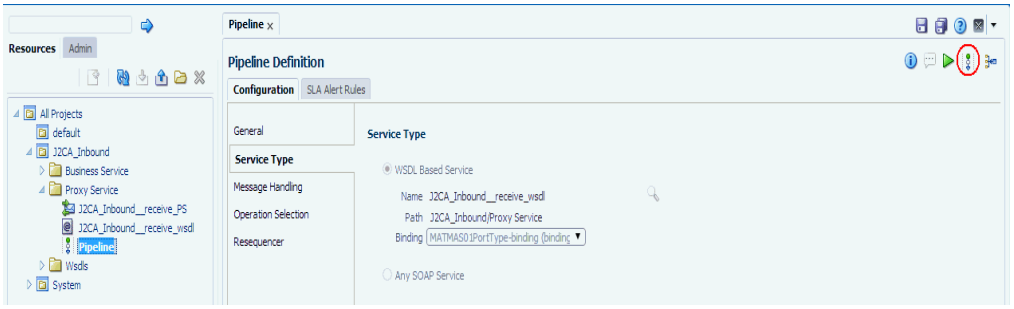


7. Select the Pipeline and click **OK**.
8. Click the Save or Save All icon in the right corner, as shown in Figure 2–64.

Figure 2–64 Save and Save All Icons

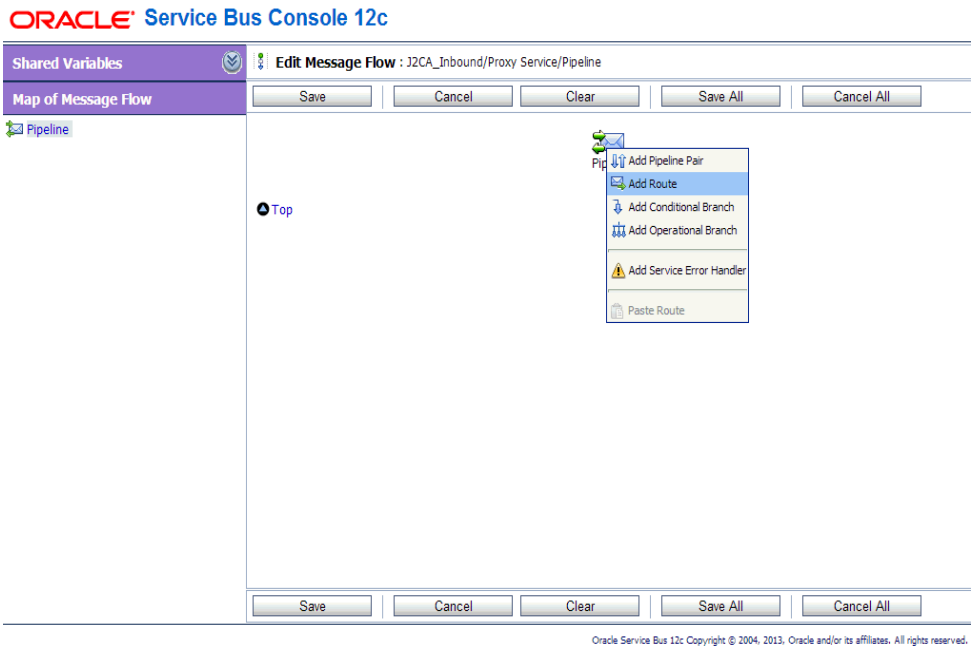
9. In the left pane, double-click **Pipeline** under the Proxy Service folder and click the down-pointing icon on the right pane to open the message flow, as shown in Figure 2–65.

Figure 2–65 Message Flow



10. Click the displayed Proxy service icon and select **Add Route** from the menu, as shown in Figure 2–66.

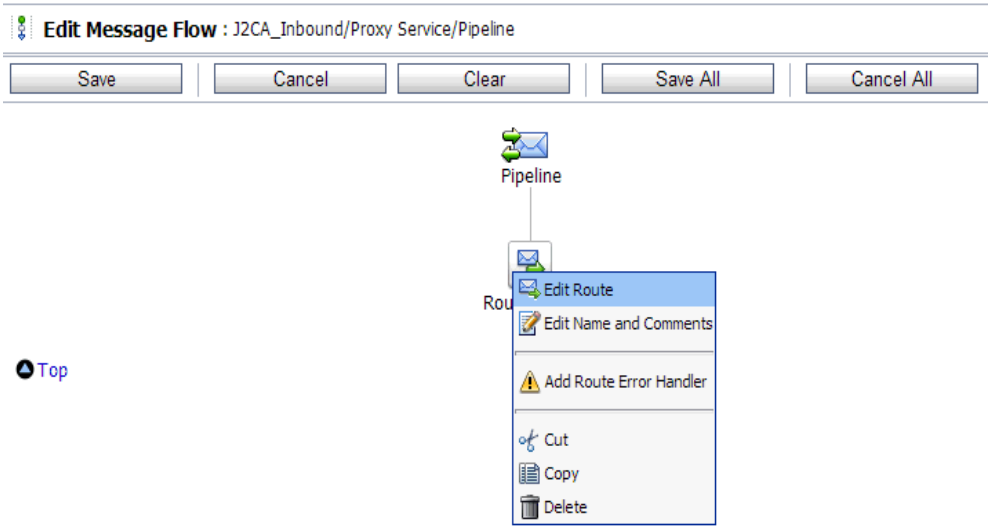
Figure 2–66 Add Route Option



The RouteNode1 icon is added.

11. Click the RouteNode1 icon and select **Edit Route** from the menu, as shown in Figure 2–67.

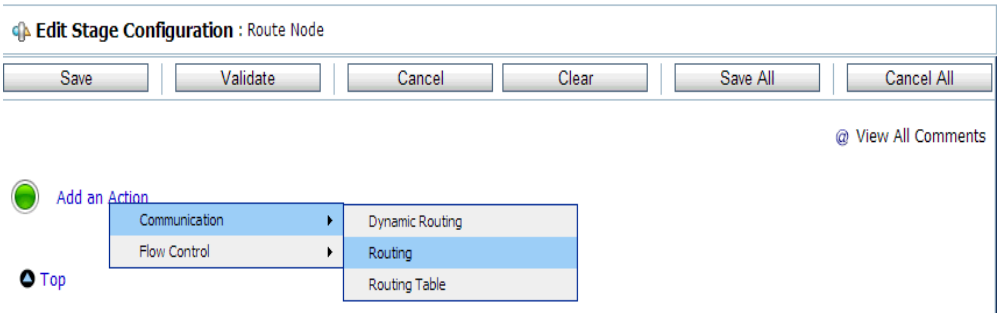
Figure 2–67 Edit Route Option



The Edit Stage Configuration workspace area is displayed.

12. Click **Add an Action**, select **Communication** from the menu, and then click **Routing**, as shown in Figure 2–68.

Figure 2–68 Edit Stage Configuration Workspace



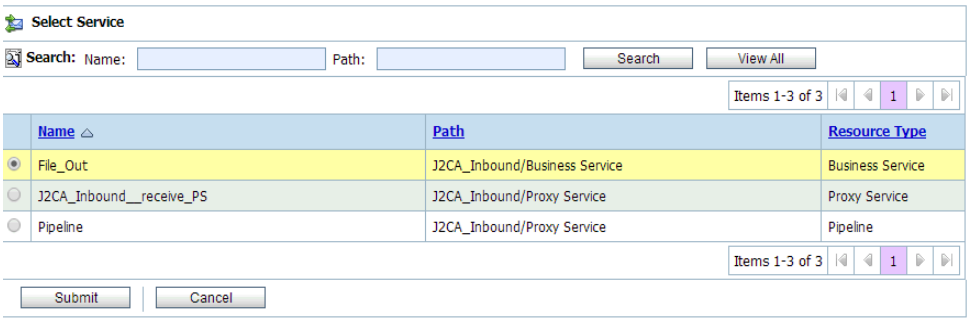
13. Click **<Service>**, as shown in Figure 2–69.

Figure 2–69 Service Route Actions

The Select Service dialog is displayed.

14. Select the **File_Out** Business service and click **Submit** as shown in Figure 2–70.

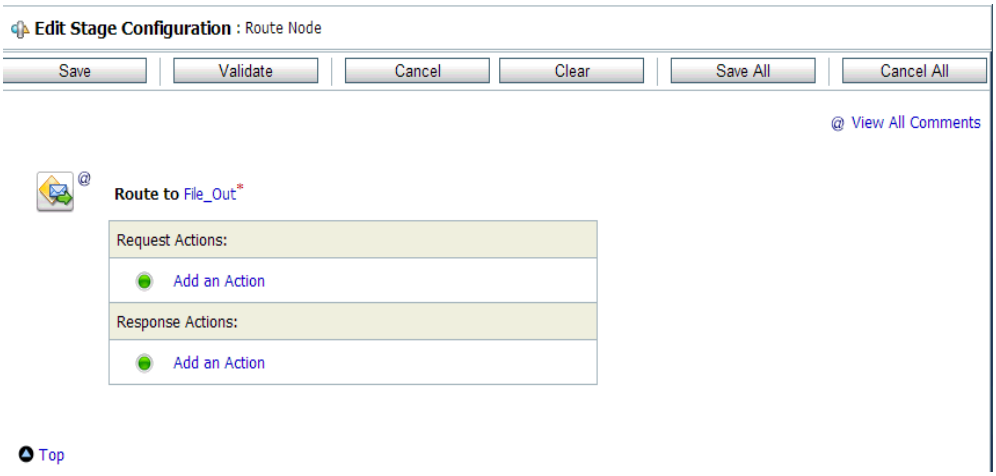
Figure 2–70 Select Service Dialog



You are returned to the Edit Stage Configuration workspace area.

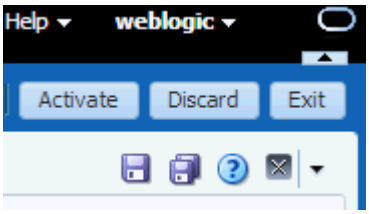
15. Click Save All, as shown inFigure 2–71.

Figure 2–71 Edit Stage Configuration Workspace Area



16. Click **Activate** in the right pane of the Oracle Service Bus session, as shown in Figure 2–72.

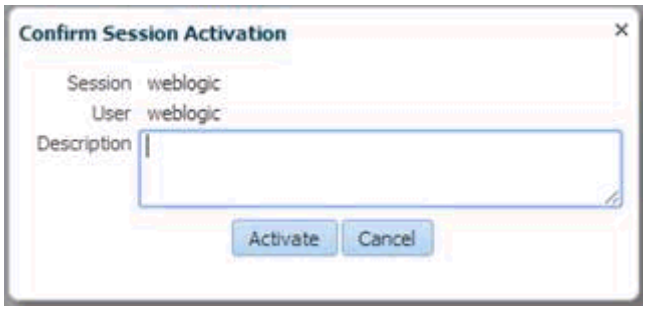
Figure 2–72 Activate Button



The Confirm Session Activation window appears.

17. Click **Activate** to save the changes, as shown in Figure 2–73.

Figure 2–73 Confirm Session Activation Window



18. Trigger an event from the SAP R/3 system and check if the output is received in the configured output location.

2.4 Configuring an Outbound Process Using sbconsole (BSE Configuration)

This section describes how to configure an outbound process using sbconsole for BSE configurations.

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

`<ADAPTER_HOME>\etc\sample\SAP_Samples.zip\SAP_Samples\OSB\BSE\SAP_Sample_BSE_OSB_Outbound_Project`

This section includes the following topics:

- Section 2.4.1, "Starting Oracle Service Bus and Creating Project Folders"
- Section 2.4.2, "Setting the Class Path for Application Explorer to Integrate With Oracle Service Bus"
- Section 2.4.3, "Publishing a WSDL From Application Explorer to Oracle Service Bus"
- Section 2.4.4, "Configuring a File Type Business Service"
- Section 2.4.5, "Configuring a WSDL-based Business Service"
- Section 2.4.6, "Configuring a Pipeline With Proxy Service"

2.4.1 Starting Oracle Service Bus and Creating Project Folders

For more information on starting Oracle Service Bus and creating project folders, see Section 2.2.1, "Starting Oracle Service Bus and Creating Project Folders" on page 2-2.

2.4.2 Setting the Class Path for Application Explorer to Integrate With Oracle Service Bus

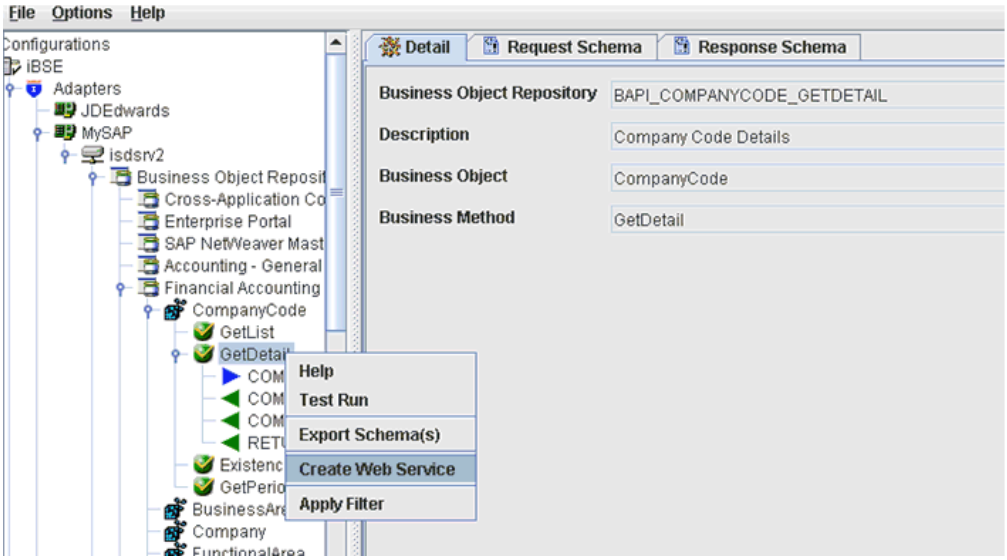
For more information on setting the class path for Application Explorer to integrate with Oracle Service Bus, see Section 2.2.2, "Setting the Class Path for Application Explorer to Integrate With Oracle Service Bus" on page 2-6.

2.4.3 Publishing a WSDL From Application Explorer to Oracle Service Bus

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

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

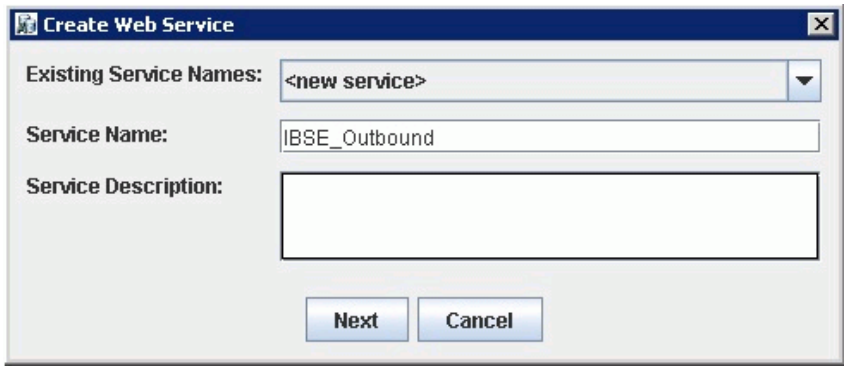
Figure 2–74 Create Web Service Option



2. Expand the **Business Object Repository** node, **Financial Accounting** node, and the **CompanyCode** business object.
3. Right-click the **GetDetail** method and select **Create Web Service** from the menu, as shown in Figure 2–74.

   The Create Web Service dialog is displayed, as shown in Figure 2–75.

Figure 2–75 Create Web Service Dialog



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

   Application Explorer switches the view to the Business Services node, and the new Web service appears in the left pane.
6. Right-click the new Web service and select **Export WSDL** from the menu.

The Export WSDL dialog is displayed, as shown in Figure 2–76.

Figure 2–76 Export WSDL Dialog

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

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

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

2.4.4 Configuring a File Type Business Service

For more information on configuring a file type business service, see Section 2.2.5, "Configuring a File Type Business Service" on page 2-10.

2.4.5 Configuring a WSDL-based Business Service

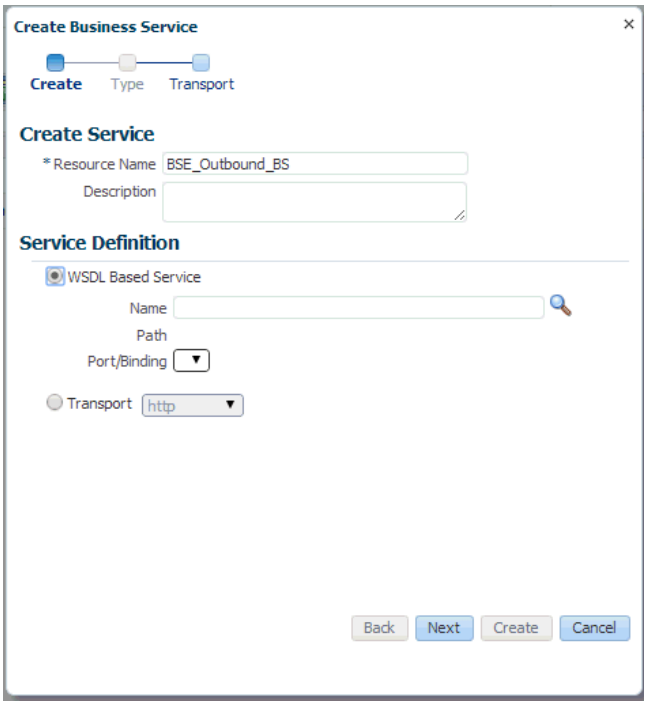
This section describes how to configure a WSDL type Business Service using the Oracle Service Bus Console.

Perform the following steps to configure a WSDL-based Proxy Service:

1. Right-click on the Business Service folder in the left pane and select **Business Service**.

   The Create Business Service window is displayed, as shown in Figure 2–77.

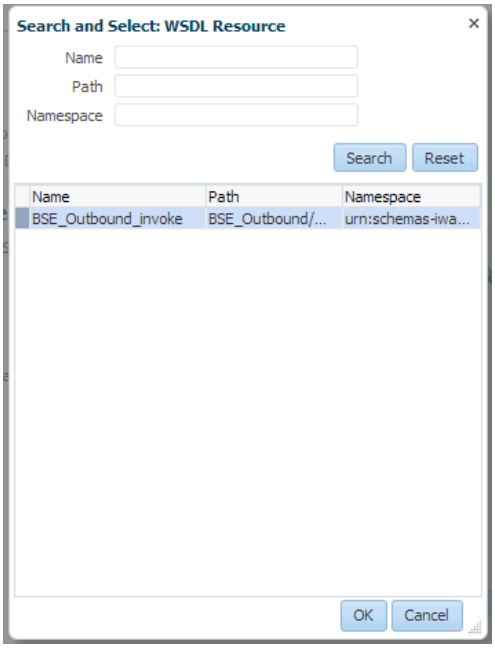
Figure 2–77 Create Business Service Window



2. Provide a name for the Business Service, and in Service Definition area, select the WSDL Based Service option and click the search icon.

   The Search and Select: WSDL Resource window is displayed, as shown in Figure 2–78.

Figure 2–78 Search and Select: WSDL Resource Window

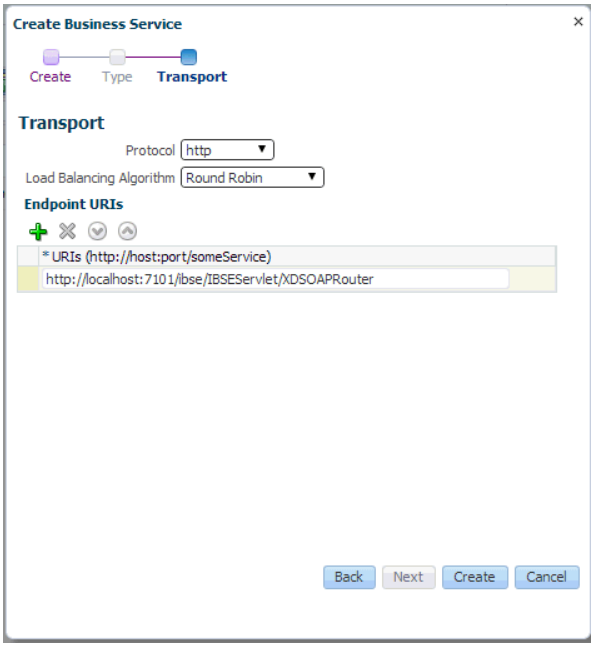


3. Click the **Search** button, select the BSE Outbound WSDL, and click **OK**.

   You are returned to the Create Business Service window.

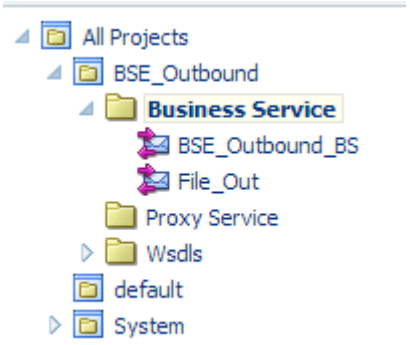
4. Click **Next**.
5. Accept the default values and click the **Create** button, as shown in Figure 2–79.

Figure 2–79 Create Business Service Window



The created WSDL-based Business Service is listed under the Business Service folder, as shown in Figure 2–80.

Figure 2–80 WSDL-based Business Service

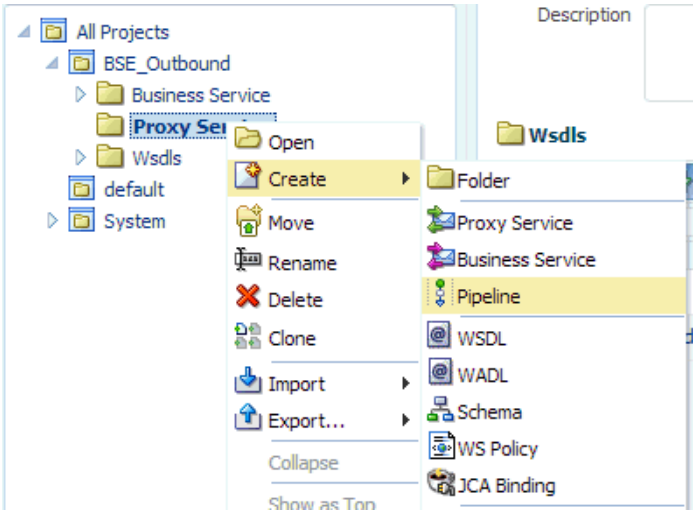


2.4.6 Configuring a Pipeline With Proxy Service

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

1. Right-click the Proxy Service folder, select **Create** and click **Pipeline**, as shown in Figure 2–81.

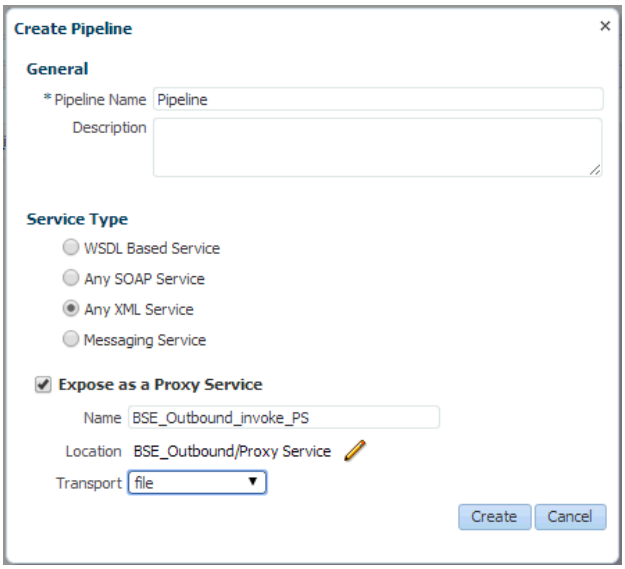
Figure 2–81 Pipeline Option



The Create Pipeline window is displayed.

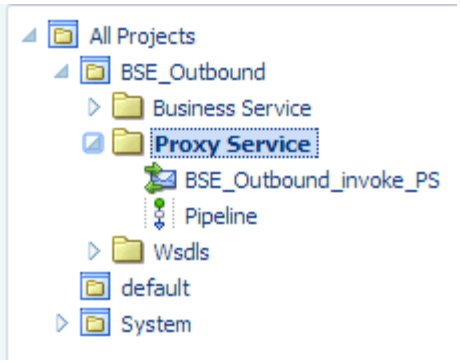
2. Enter a name in the Pipeline Name field. By default, **Expose as a Proxy Service** is selected. If you wish to change the Proxy Service Name, change it and set Transport to **file**, and click **Create** as shown in Figure 2–82.

Figure 2–82 Create Pipeline Window



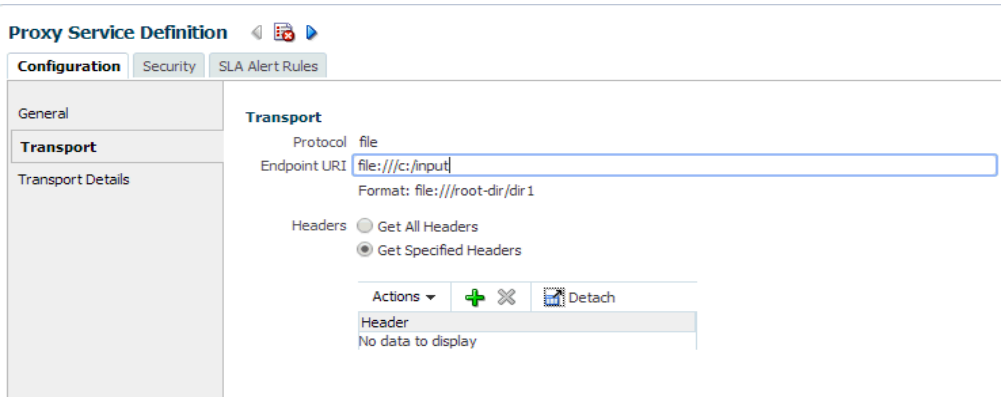
The created Pipeline and the Proxy Service is listed under Proxy Service, as shown in Figure 2–83.

Figure 2–83 Pipeline Node



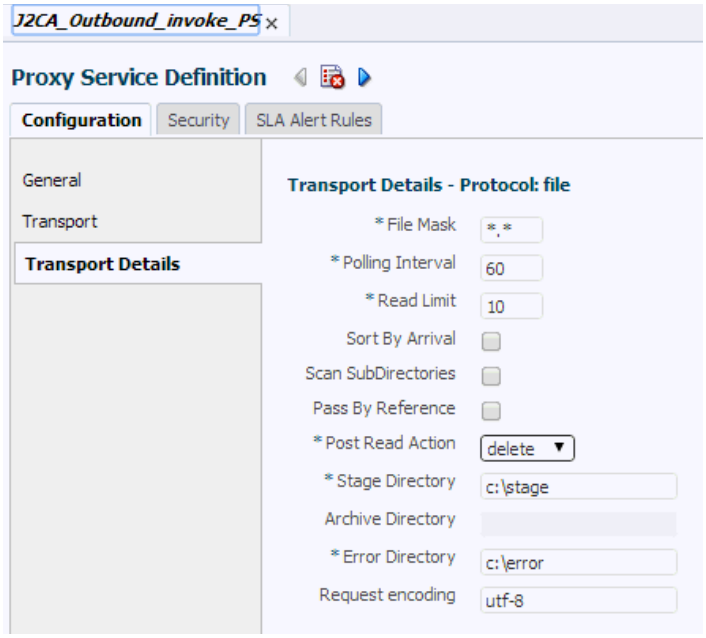
3. Double-click the created proxy service and click **Transport** in the left pane. Provide the input location in the Endpoint URI field, as shown in Figure 2–84.

Figure 2–84 Transport



4. Click **Transport Details** in the left pane and provide the location for the Stage Directory and the Error Directory fields, as shown in Figure 2–85.

Figure 2–85 Transport Details



5. Click the **Save All** icon in the right corner, as shown in Figure 2–86.

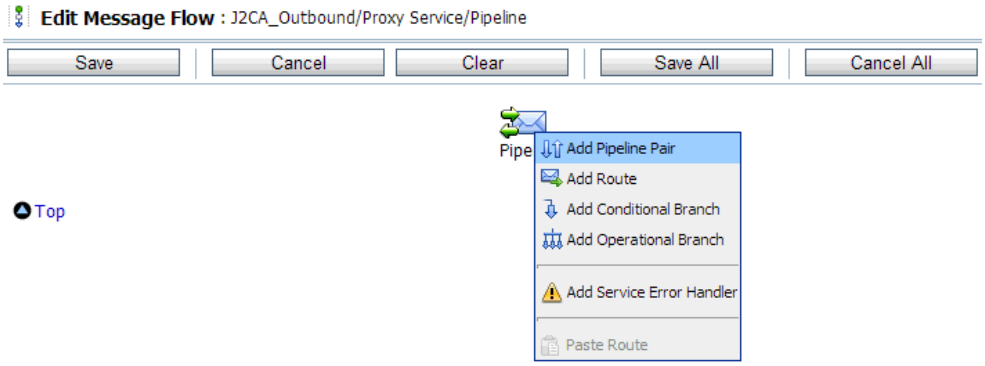
Figure 2–86 Save All Icon

6. Double-click the **Pipeline** node and click the **Open Message Flow** icon on the right pane to open the message flow, as shown in Figure 2–87.

Figure 2–87 Open Message Flow Icon

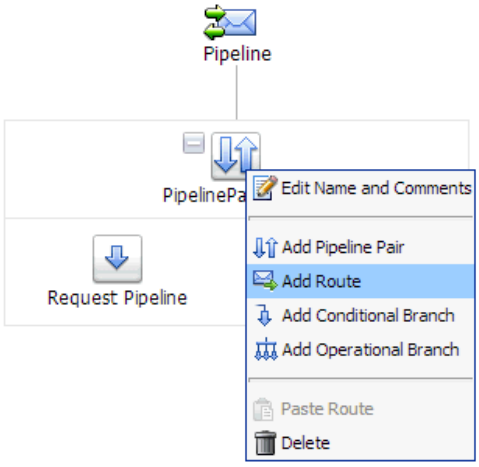
7. Click the Proxy Service icon and select **Add Pipeline Pair** from the menu, as shown in Figure 2–88.

Figure 2–88 Add Pipeline Pair Option



8. Click the **PipelinePairNode1** icon and select **Add Route** from the menu, as shown in Figure 2–89.

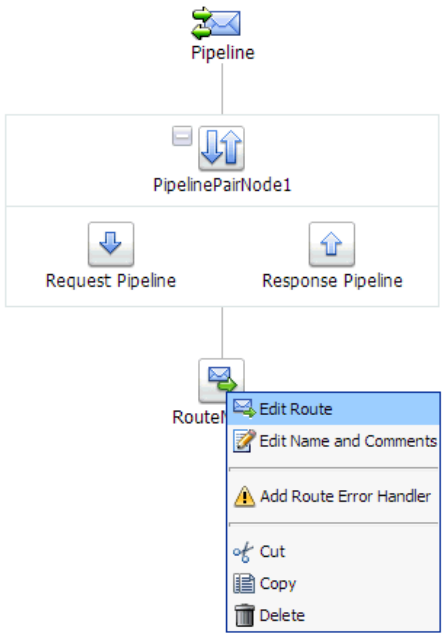
Figure 2–89 Add Route Option



The RouteNode1 icon is added below the PipelinePairNode1 icon.

9. Click the RouteNode1 icon and select **Edit Route** from the menu, as shown in Figure 2–90.

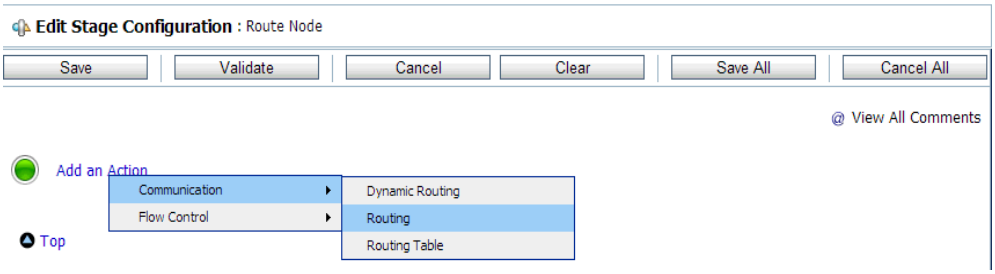
Figure 2–90 Edit Route Option



The Edit Stage Configuration workspace area is displayed.

10. Click **Add an Action**, select **Communication** and click **Routing**, as shown in Figure 2–91.

Figure 2–91 Edit Stage Configuration Workspace Area



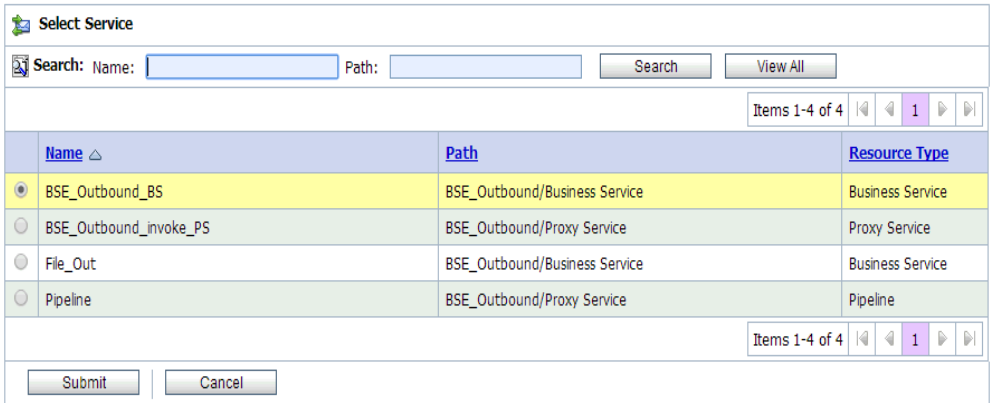
11. Click **<Service>**, as shown in Figure 2–92.

Figure 2–92 Actions

The Select Service dialog is displayed.

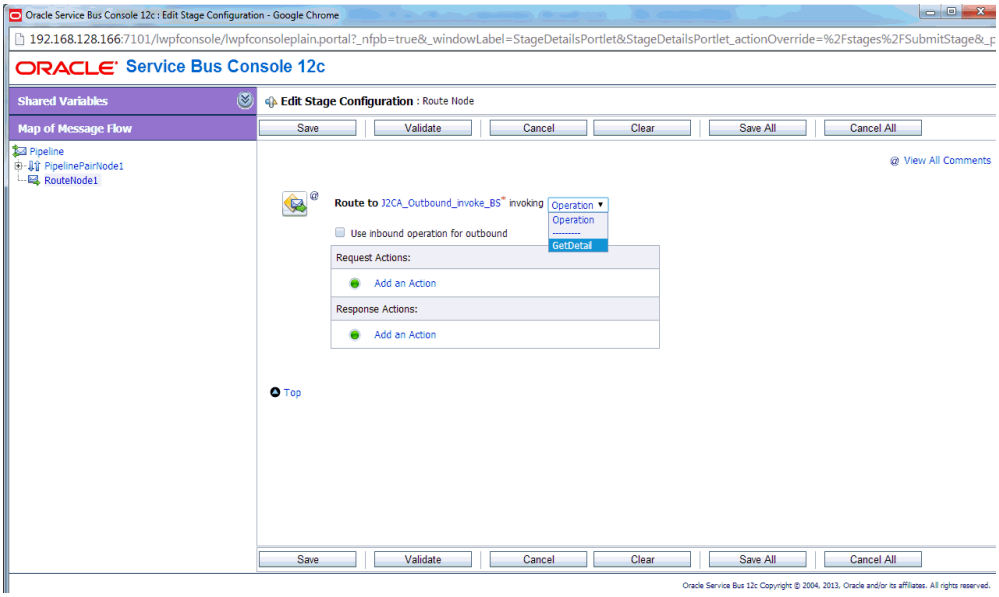
12. Select the WSDL type Business Service configured for SAP R/3 and click on **Submit**, as shown in Figure 2–93.

Figure 2–93 Select Service Dialog



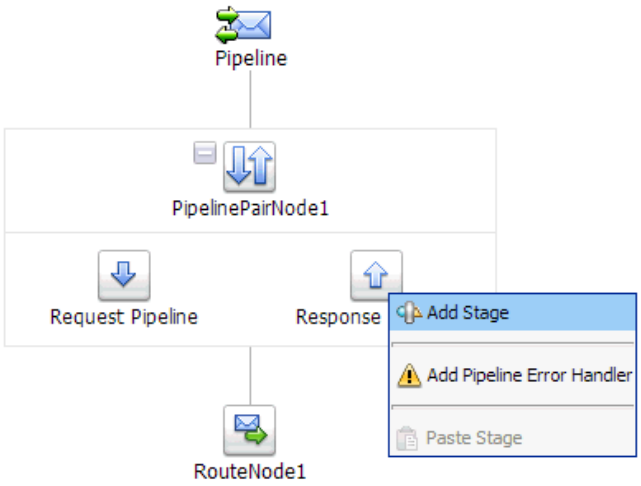
13. Select the name of the SAP R/3 business object (for example, GetDetail) as the operational attribute from the list, and click **Save**, as shown in Figure 2–94.

Figure 2–94 Edit Stage Configuration



14. Click the Response Pipeline icon and select **Add Stage** from the menu, as shown in Figure 2–95.

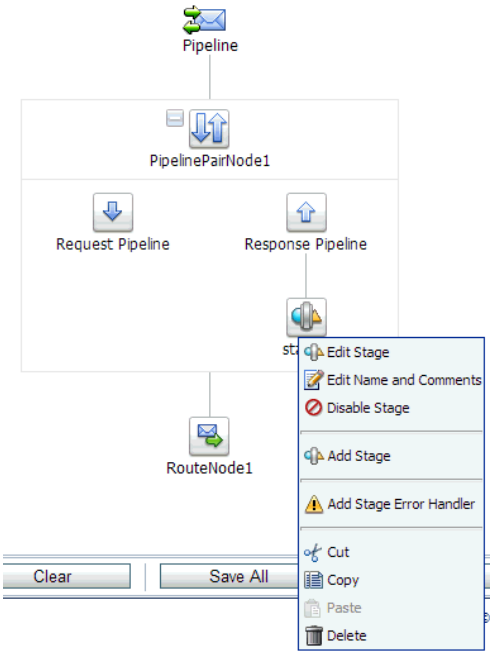
Figure 2–95 Response Pipeline Icon



The Stage1 icon is added below the Response Pipeline icon.

15. Click the Stage1 icon and select **Edit Stage** from the menu, as shown in Figure 2–96.

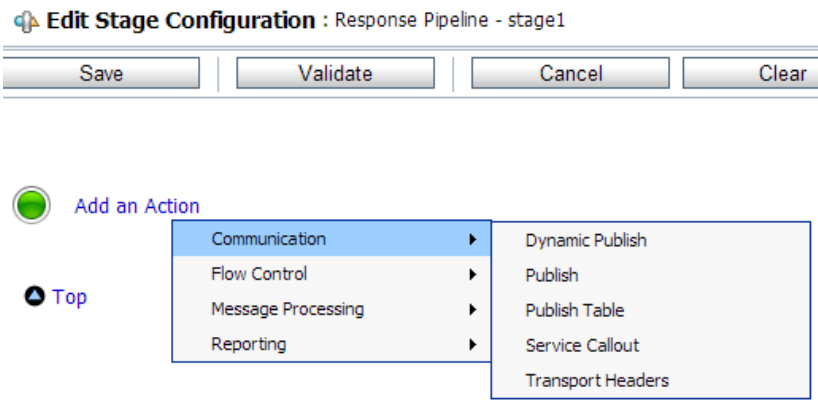
Figure 2–96 Edit Stage Option



The Edit Stage Configuration workspace area is displayed.

16. Click **Add an Action**, select **Communication**, and then click **Publish**, as shown in Figure 2–97.

Figure 2–97 Edit Stage Configuration Workspace Area

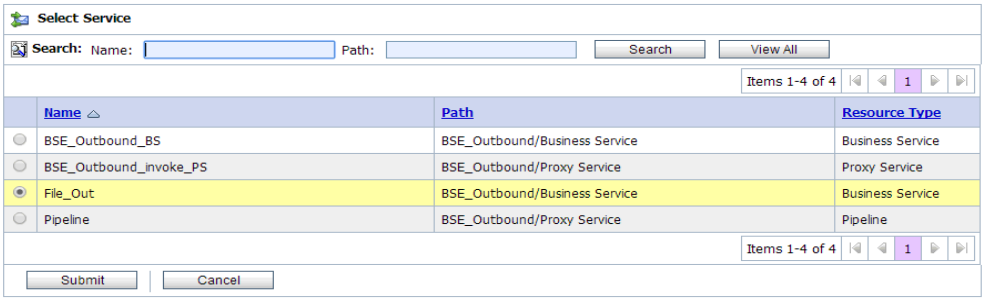


17. Click **<Service>**, as shown in Figure 2–98.

Figure 2–98 <Service> Action

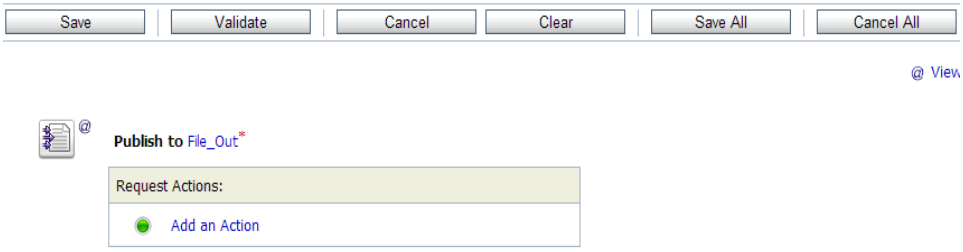
18. In the Select Service dialog, select a File type Business Service and click **Submit**, as shown in Figure 2–99.

Figure 2–99 Select Service Dialog



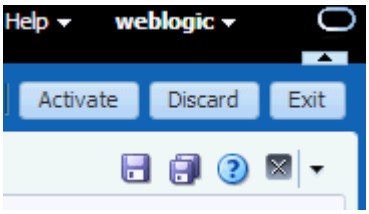
19. Click **Save All**, as shown in Figure 2–100.

Figure 2–100 Save All Button



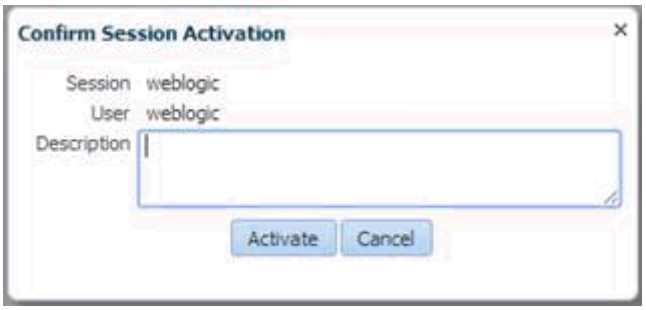
20. Click **Activate** in the right pane of the Oracle Service Bus session, as shown in Figure 2–101.

Figure 2–101 Activate Button



21. Click **Activate** to save the changes, as shown in Figure 2–102.

Figure 2–102 Confirm Session Activation



22. Copy and paste an input XML file in the input folder you have configured (for example, C:\input).

    Output is received in the configured output location (for example, C:\output).

2.5 Working With Synchronous SAP R/3 Events Using Oracle Service Bus (J2CA Configuration)

This section describes how to work with synchronous SAP R/3 events using Oracle Service Bus. For more information on synchronous SAP R/3 events, see *Section 6.6.1, "Overview"* in the *Oracle Fusion Middleware Application Adapter for SAP R/3 User's Guide for Oracle WebLogic Server*.

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

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

`<ADAPTER_HOME>\etc\sample\SAP_Samples.zip\SAP_Samples\OSB\J2CA\SAP_Sample_J2CA_OSB_Synchronous_Events`

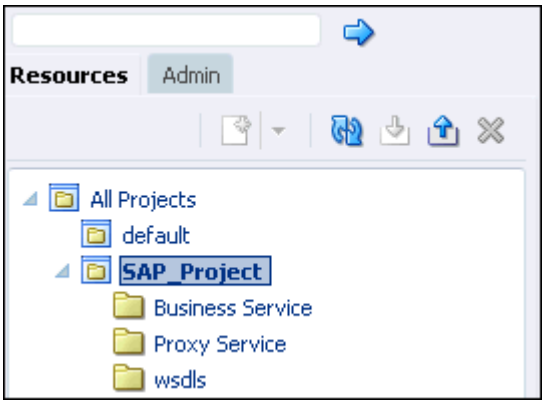
This section includes the following topics:

- Section 2.5.1, "Starting Oracle Service Bus and Creating Project Folders"
- Section 2.5.2, "Configuring Adapter Targets and Channels"
- Section 2.5.3, "Generating a WSDL File for a Message"
- Section 2.5.4, "Creating a Business Service in the Oracle Service Bus Console"
- Section 2.5.5, "Creating a Proxy Service in the Oracle Service Bus Console"
- Section 2.5.6, "Configuring a Pipeline in the Proxy Service"

2.5.1 Starting Oracle Service Bus and Creating Project Folders

Start Oracle Service Bus and create the required project folders, as shown in Figure 2–103.

Figure 2–103 Project Explorer



For more information, see Section 2.2.1, "Starting Oracle Service Bus and Creating Project Folders".

2.5.2 Configuring Adapter Targets and Channels

Start Application Explorer and create two adapter targets and one channel for SAP R/3 using a J2CA configuration.

For more information, see *Chapter 4, "Configuring Oracle Application Adapter for SAP R/3* in the *Oracle Fusion Middleware Application Adapter for SAP R/3 User's Guide for Oracle WebLogic Server*.

2.5.3 Generating a WSDL File for a Message

Perform the following steps to generate a WSDL file for a message:

1. Ensure that the channel is not started before generating the WSDL file.
2. Connect to the first SAP adapter target (target1).

3. Generate an inbound WSDL file for BAPI_COMPANYCODE GETLIST (select **Remote Function Modules**, **Financial Accounting**, and then **0002 - Company Code Business Object**).
4. Connect to the second SAP adapter target (target2).
5. Generate an outbound WSDL file for BAPI_COMPANYCODE_GETLIST (select **Remote Function Modules**, **Financial Accounting**, and then **0002 - Company Code Business Object**).

   For more information, see Section 2.2.3, "Publishing a WSDL From Application Explorer to Oracle Service Bus" and Section 2.3.3, "Publishing a WSDL From Application Explorer to Oracle Service Bus".

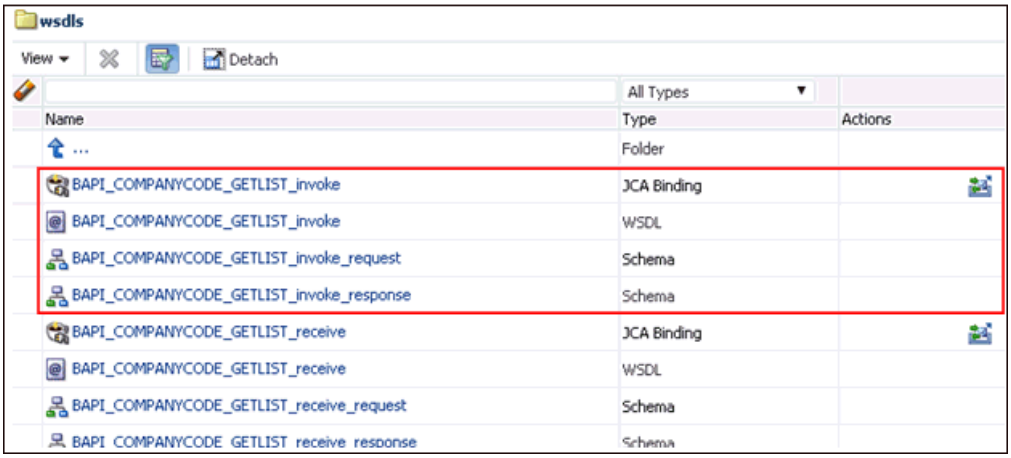
2.5.4 Creating a Business Service in the Oracle Service Bus Console

This section describes how to create a Business Service in the Oracle Service Bus console.

1. In the Oracle Service Bus console, click **Create**, which is located on the top and select the created WSDL folder in the left pane.

   WSDL, JCA binding, and XSD files are listed in the right pane, as shown in Figure 2–104.

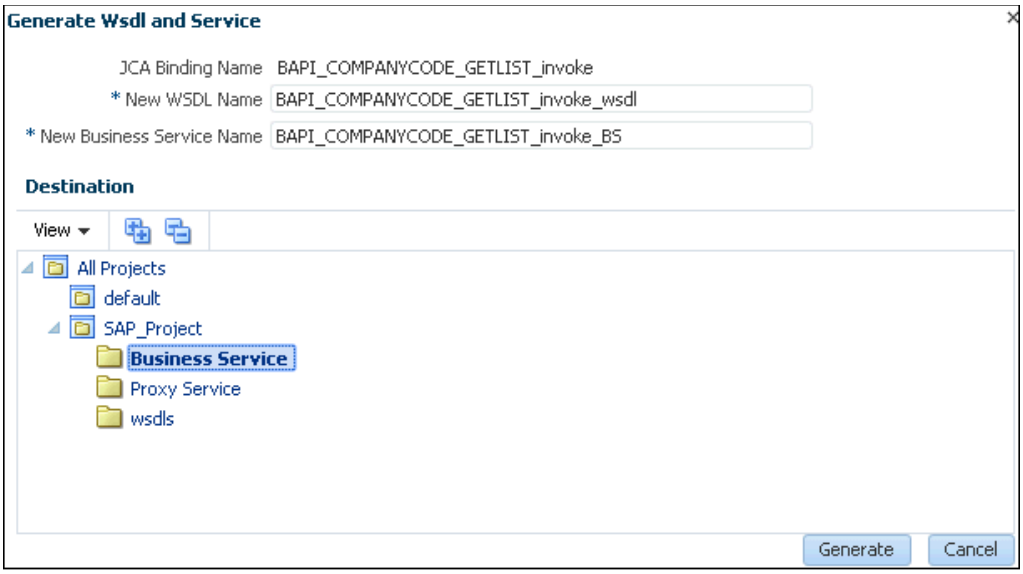
Figure 2–104 Contents of the WSDL Folder



2. Click the icon corresponding to the exported outbound JCA binding.

   The Generate WSDL and Service page opens, as shown in Figure 2–105.

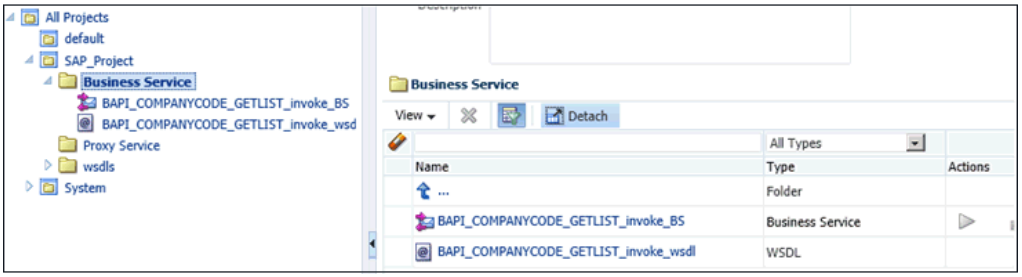
Figure 2–105 Generate WSDL and Service Page



3. Provide a new WSDL name and a new Business Service name in the corresponding fields.
4. In the Destination area, select the **Business Service** subfolder from the corresponding project.
5. Click **Generate**.

   Select the Business Service folder in the left pane under the project folder (for example, SAP_Project) and check that the WSDL and service resources are successfully created, as shown in Figure 2–106.

Figure 2–106 Checking the WSDL and Service Resources



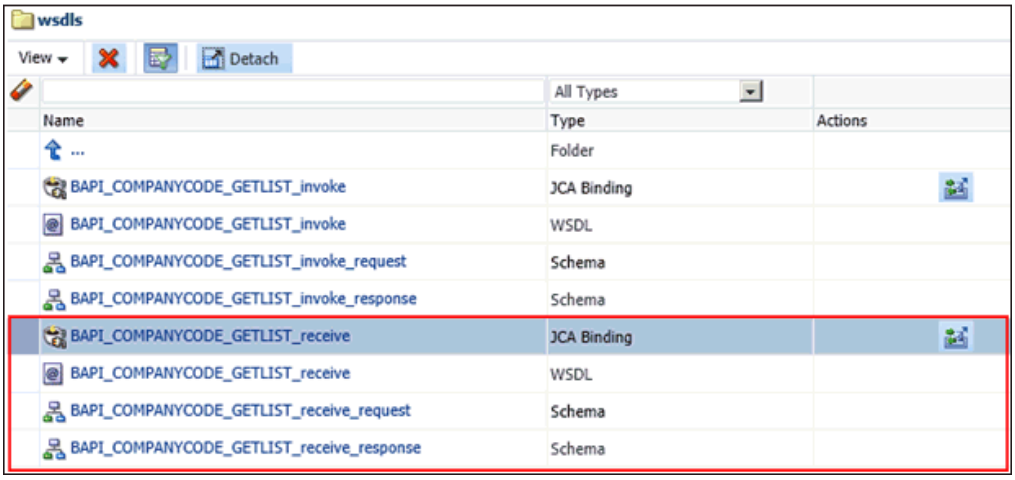
2.5.5 Creating a Proxy Service in the Oracle Service Bus Console

This section describes how to create a Proxy Service in the Oracle Service Bus console.

1. In the Oracle Service Bus console, select the created WSDL folder in the left pane.

   WSDL, JCA binding, and XSD files are listed in the right pane, as shown in Figure 2–107.

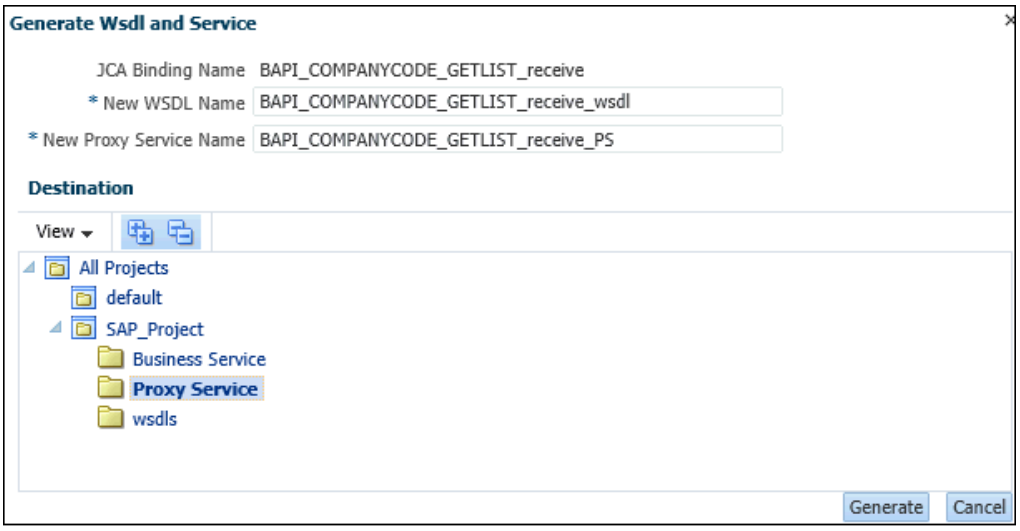
Figure 2–107 Contents of the WSDL Folder



2. Click the icon corresponding to the exported inbound JCA binding.

   The Generate WSDL and Service page is displayed, as shown in Figure 2–108.

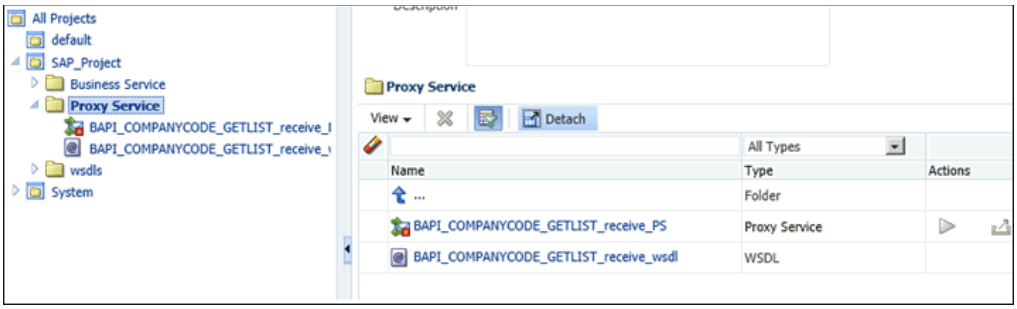
Figure 2–108 Generate WSDL and Service Page



3. Provide a new WSDL name and a new Proxy Service name in the corresponding fields.
4. In the Destination area, select the **Proxy Service** subfolder from the corresponding project.
5. Click **Generate**.

   Select the Proxy Service folder in the left pane under the project folder (for example, SAP_Project) and check that the WSDL and service resources are successfully created, as shown in Figure 2–109.

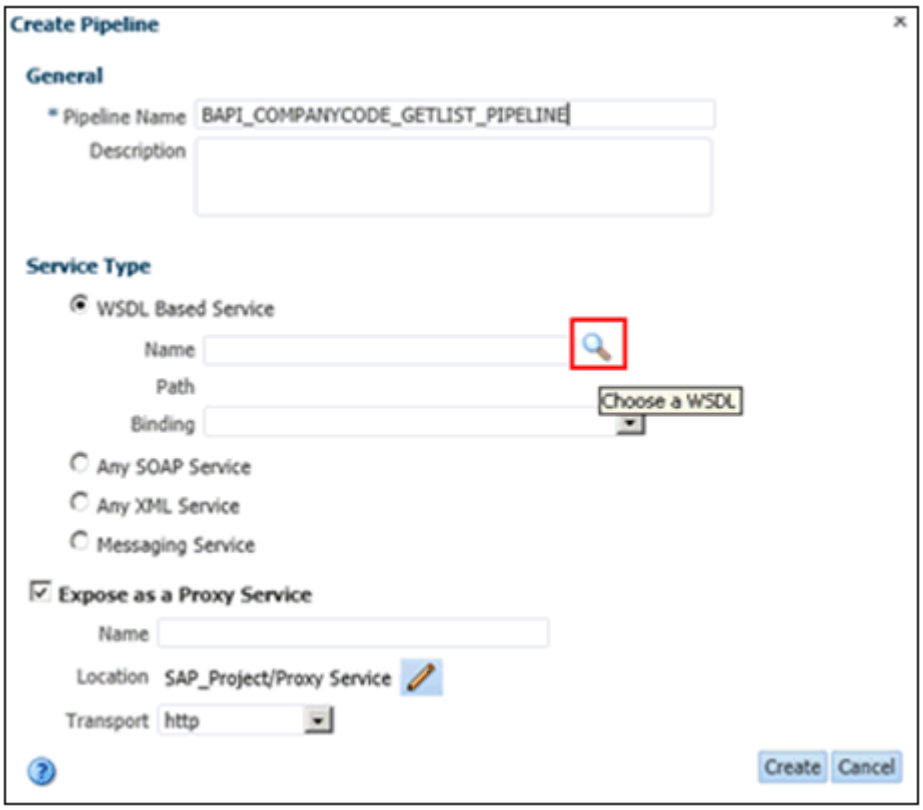
Figure 2–109 Checking the WSDL and Service Resources



6. Right-click the Proxy Service, select **Create** and then click **Pipeline**.

   The Create Pipeline window is displayed, as shown in Figure 2–110.

Figure 2–110 Create Pipeline Window



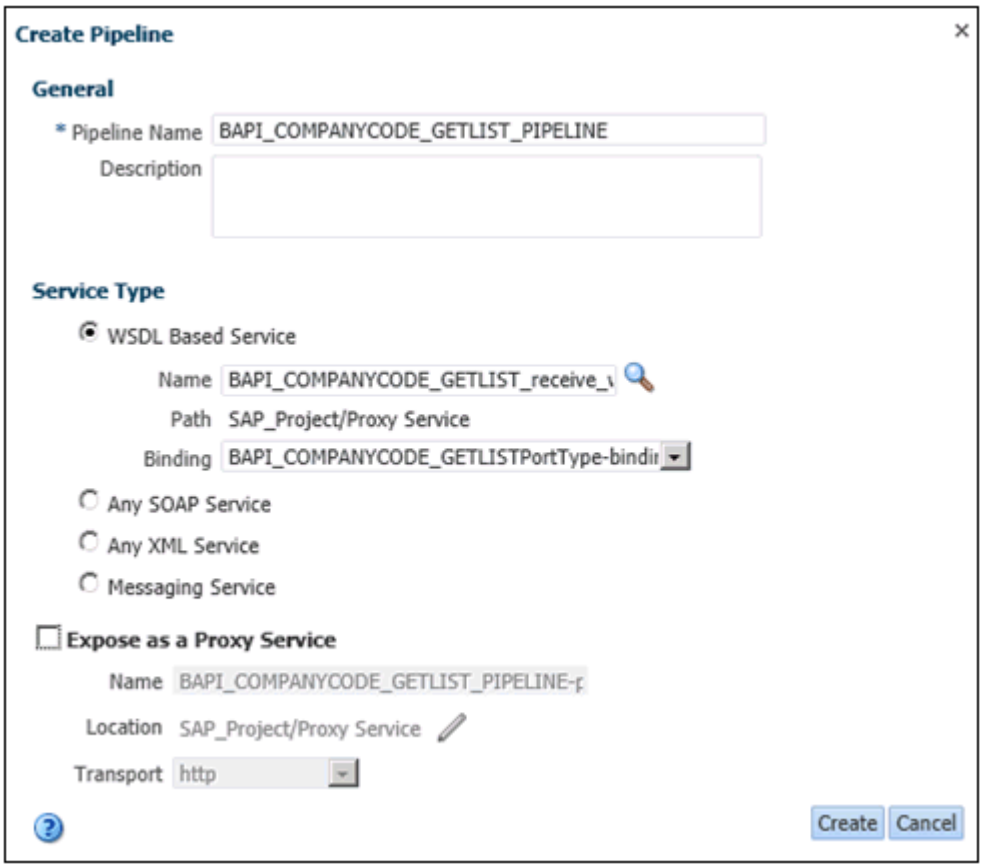
7. Enter a Pipeline name, select **WSDL Based Service** as the Service Type, and then click the **Search** icon next to the Name field.

   The Search and Select: WSDL Resource window is displayed.

8. Click **Search** and in the displayed list, select **BAPI_COMPANYCODE GETLIST_receive_wsdl**, which is available under the Proxy service, and then click **Ok**.

   You are returned to the Create Pipeline window, as shown Figure 2–111.

Figure 2–111 Create Pipeline Window

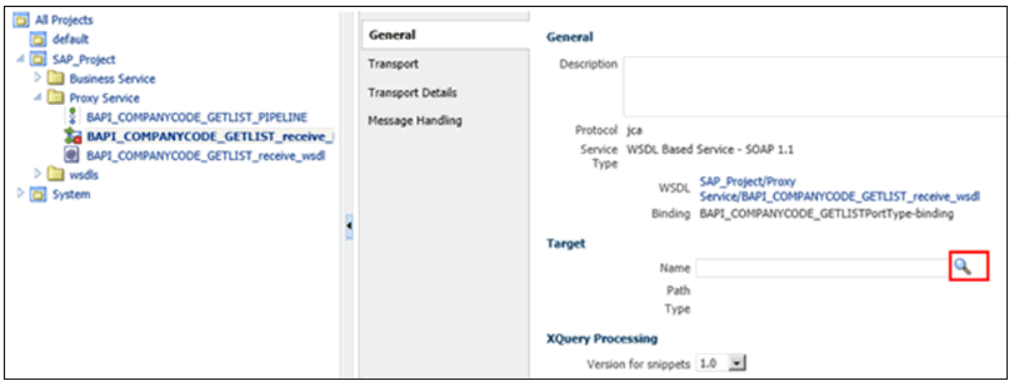


9. Uncheck the **Expose as a Proxy Service** option and then click **Create**.

   The Pipeline will be created and listed under the Proxy Service folder.

10. Double-click **BAPI_COMPANYCODE_GETLIST_receive_PS** under the Proxy Service folder and then click the **Search** icon in the Target area in left pane, as shown in Figure 2–112.

Figure 2–112 Search Icon



The Search and Select: Service Resource window is displayed.

11. Click **Search** and in the displayed list select **BAPI COMPANYCODE_GETLIST_PIPELINE**, which is available under the Proxy Service, and then click **Ok**.

12. Click **Save/Save All** in the left corner.

2.5.6 Configuring a Pipeline in the Proxy Service

This section describes how to configure a Pipeline in the Proxy Service.

1. Click the created Pipeline under the Proxy Service folder and click the **Open Message Flow** icon at the top of right pane.
2. In the Edit Message flow workspace area, click the displayed Proxy Service icon and select **Add Route** from the menu.
3. Click the **RouteNode1** icon and select **Edit Route** from the menu.
4. Click **Add an Action**, select **Communication** from the menu, and then click **Routing**.
5. Click **<Service>**.
6. Select the created Business Service and click **Submit**.
7. Select the **BAPI_COMPANYCODE_GETLIST** operation from the list.
8. Under **Request Actions**, click **Add an Action - Reporting**, and select **Log**.
9. Click the Log **<Expression>**.
10. In the XQuery/XSLT Expression Editor page, type **$body**, click **Validate**, and then click **Save**.
11. Select **Error** from the Severity Level list under the Log area in the Request Actions section.
12. Under **Response Actions**, click **Add an Action - Reporting**, and select **Log**.
13. Click on the Log **<Expression>**.
14. In the XQuery/XSLT Expression Editor page, type **$body**, click **Validate**, and then click **Save**.
15. Select **Error** from the Severity Level list under the Log area in the Response Actions section.

    The final pipeline structure is shown in Figure 2–113.

Figure 2–113 Final Pipeline Structure

16. Click **Save**.

    You are returned to the Edit Message Flow workspace area.

17. Click **Save**.

18. Click **Activate** located at the top of the screen.

    In the Confirm Session Activation window that is displayed, click **Activate**.

2.6 Configuring JMS Proxy Services Using Oracle Service Bus (J2CA Configuration)

This section describes how to configure JMS Proxy Services using Oracle Service Bus for a J2CA configuration.

1. Start Oracle Service Bus and create the required project folder.

   For more information, see Section 2.2.1, "Starting Oracle Service Bus and Creating Project Folders".

2. Generate and publish the WSDL from Application Explorer to the created project folder. Using the published WSDL, create a Business Service.

   For more information, see Section 2.2.3, "Publishing a WSDL From Application Explorer to Oracle Service Bus".

3. Open the Service Bus Console page.

4. In the Project Explorer, right-click the **Proxy Service** folder, select **Create**, and then click **Resource**.

   The Resource Gallery is displayed.

5. Click **Proxy Service**.
6. Click **SOAP** and then click **OK**.
7. Provide a name for the Proxy Service. In the Service Definition area, select the **WSDL Based Service** option and then click the **Search** icon.

   The Search and Select: WSDL Resource window is displayed.
8. Click **Search**, select the existing business service, and then uncheck the **Generate Pipeline** option.
9. Click **Next**.
10. Select **jms** from the Protocol list and then click **Create**.
11. Double-click the created Proxy Service and select the existing business service under the Target section.
12. In Transport details section, provide the following parameters.
    a. Select **Queue** in the Destination Type section.
    b. Enable the **Is Response Required** check box.
    c. Select **Text** in the Response Message Type section.
    d. In the Response URI field, provide the Endpoint URI used in the Transport Configuration and change `Request` to `Response`.

       For example:

       jms://localhost:7003/weblogic.jms.XAConnectionFactory/Adapter_outbound_PSResponse
13. In the Message Handling section, ensure the **SOAP Body Type** is selected and then click **Next**.
14. Click **Save**.
15. In the left pane, click **Activate**, and then **Submit**.
16. In the left pane, click **ProxyService** under the Projects folder.
17. Click the **Launch Test Console** icon for the created Proxy Service.
18. Provide the input values for **Payload**, uncheck the **Direct Call** box, and click **Execute**.
19. Review the Response document, and then click **Close**.
20. Click the **Oracle WLS Console** tab.
21. In the Oracle WLS Console, expand **Services**, expand **Messaging**, and click **JMS Modules**.
22. Click **jmsResources**.
23. Click **Lock & Edit**.
24. Click the appropriate request link, for example, **Adapter_outbound_PSRequest**.
25. Click the **Monitoring** tab.
26. Enable the check box and click **Show Messages**
27. Click **New**.
28. Provide the input payload in the **Body** field and click **OK**.

    A Success message appears.

29. In the Oracle WLS Console, expand **Services**, expand **Messaging**, and click **JMS Modules**.
30. Click **jmsResources**.
31. Click the appropriate response link, for example, **Adapter_outbound_PSResponse**.
32. Click the **Monitoring** tab.
33. Enable the check box and click **Show Messages**.
34. Click the **ID** link.

    The response document is shown under the Text field.

2.7 Configuring HTTP Proxy Services Using Oracle Service Bus (J2CA Configuration)

This section describes how to configure HTTP Proxy Services using Oracle Service Bus for a J2CA configuration.

1. Start the Oracle Service Bus and create the required project folders.

   For more information, see Section 2.2.1, "Starting Oracle Service Bus and Creating Project Folders".
2. Generate and publish the WSDL from Application Explorer to the created project folder, and create a Business Service using the published WSDL.

   For more information, see Section 2.2.3, "Publishing a WSDL From Application Explorer to Oracle Service Bus".
3. Open the Service Bus Console page.
4. In the Project Explorer, right-click the **Proxy Service** folder, select **Create**, and then click **Resource**.

   The Resource Gallery is displayed.
5. Click **Proxy Service**.
6. Click **SOAP** and then click **OK**.
7. Provide a name for the Proxy Service. In the Service Definition area, select the **WSDL Based Service** option and then click the **Search** icon.

   The Search and Select: WSDL Resource window is displayed.
8. Click **Search**, select the existing business service, and then uncheck the **Generate Pipeline** option.
9. Select **http** in the Protocol list and click **Create**.
10. Double-click the created Proxy Service and select the existing business service under the Target section.
11. Click **Save**.

    The created Proxy Service is saved.
12. Click **Activate** in the left pane, and then **Submit** on the right pane.
13. Click **ProxyService** in the Projects folder on the left pane.
14. Click the **Launch Test Console** icon for the created Proxy Service.

15. Provide the input values for the Payload, and then click **Execute**.
16. Review the **Response Document**.