

Oracle®

Application Adapter for SAP R/3 User's Guide

WebLogic Integration 10g Release 3 (10.3.1)

January 2010

Oracle Application Adapter for SAP R/3 User's Guide, WebLogic Integration 10g Release 3 (10.3.1)

Copyright © 2008, 2010, Oracle. All rights reserved.

Primary Author: Stefan Kostial

Contributing Authors: Jyothi Jandhyala, Siva Krishnaje

The Programs (which include both the software and documentation) contain proprietary information; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. This document is not warranted to be error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose.

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software--Restricted Rights (June 1987). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and we disclaim liability for any damages caused by such use of the Programs.

Oracle, JD Edwards, PeopleSoft, and Siebel are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

Contents

| | |
|---|------|
| Preface | ix |
| Audience | ix |
| Documentation Accessibility | ix |
| Related Documents | x |
| Conventions | x |
| Help Us to Serve You Better | xiii |
| | |
| 1 Introduction | |
| Adapter Features | 1-1 |
| Supported Versions and Platforms | 1-2 |
| Classical SAP Technologies for ABAP | 1-2 |
| Integration with SAP R/3 | 1-3 |
| Adapter Architecture | 1-4 |
| BSE Versus Oracle Adapter J2CA Deployment | 1-6 |
| Installation Considerations | 1-7 |
| | |
| 2 Configuring Oracle Application Adapter for SAP R/3 | |
| Starting Application Explorer | 2-1 |
| Establishing a Connection (Target) for SAP R/3 | 2-1 |
| Defining a Target to SAP R/3 | 2-2 |
| Connecting to a Defined SAP R/3 Target | 2-8 |
| Managing a Connection to SAP R/3 | 2-8 |
| Viewing Application System Objects | 2-9 |
| Viewing XML Schemas | 2-10 |
| Generating WSDL (J2CA Configurations Only) | 2-10 |
| Creating and Testing a Web Service (BSE Configurations Only) | 2-12 |
| Configuring an Adapter Channel | 2-15 |
| Creating and Editing an Adapter Channel | 2-16 |
| | |
| 3 Configuring Outbound Processing Using Oracle WebLogic Integration (J2CA Configuration) | |
| Creating an Outbound JPD Process Application | 3-1 |
| Deploying the Outbound Process on WebLogic Server | 3-33 |
| Invoking the Input XML Using a Web Browser | 3-38 |

| | | |
|----------|--|------|
| 4 | Configuring Inbound Processing Using Oracle WebLogic Integration (J2CA Configuration) | |
| | Creating an Inbound JPD Process Application | 4-1 |
| | Deploying the Inbound Process on WebLogic Server | 4-17 |
| | Creating an Event Generator and a Channel Subscription | 4-23 |
| | Triggering an Event From SAP GUI | 4-28 |
| | Verifying the Event | 4-31 |
| 5 | Configuring Outbound Processing Using Oracle WebLogic Integration (BSE Configuration) | |
| | Creating an Outbound JPD Process Application | 5-1 |
| | Exporting the Outbound JPD Process Application as an EAR File | 5-32 |
| | Deploying the Outbound JPD Process Application EAR File on WebLogic Server | 5-34 |
| | Creating a Queue to Receive the Response Document in the Test Console | 5-39 |
| | Invoking the Input XML Using a Web Browser | 5-44 |
| 6 | Advanced User Tools | |
| | Web Services Policy-Based Security | 6-1 |
| | Configuring Web Services Policy-Based Security | 6-2 |
| | Migrating Repositories | 6-8 |
| 7 | Troubleshooting and Error Messages | |
| | Troubleshooting | 7-1 |
| | BSE Error Messages | 7-4 |
| | General Error Handling in BSE | 7-5 |
| | Adapter-Specific Error Handling | 7-5 |
| A | Configuring SAP R/3 for Inbound and Outbound Processing | |
| | Configuring SAP R/3 Inbound Processing | A-1 |
| | Configuring a Logical System | A-1 |
| | Configuring a Distribution Model | A-4 |
| | Defining a Partner Profile | A-7 |
| | Configuring SAP R/3 Outbound Processing | A-9 |
| | Related Concepts and Terminology | A-9 |
| | Registering Your Program ID in SAPGUI | A-11 |
| | Testing the SAP R/3 Event Adapter | A-12 |
| | Application Link Embedding Configuration for the Event Adapter | A-14 |
| | Defining a Port | A-14 |
| | Creating a Logical System | A-14 |
| | Creating a Partner Profile | A-16 |
| | Collected IDocs | A-17 |
| | Creating a Distribution Model for the Partner and Message Type | A-17 |
| | Testing the SAP R/3 ALE Configuration | A-19 |

B Oracle WebLogic Integration Deployment

| | |
|---|-----|
| Adapter Integration with Oracle WLI | B-1 |
| Deployment of Adapter | B-1 |
| Updating Adapter Configuration..... | B-2 |

Glossary

Index

Preface

This Preface contains these topics:

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

Audience

Oracle Application Adapter for SAP R/3 (WebLogic Integration 10g Release 3) User's Guide is intended for system integrators who develop client interfaces between SAP R/3 and other applications. It describes how to use the adapter to integrate SAP R/3 IDocs, RFCs, and BAPIs with Oracle WebLogic Integration.

To use this document, you need to know how to install and configure Oracle WebLogic Integration.

Documentation Accessibility

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

<http://www.oracle.com/accessibility/>

Accessibility of Code Examples in Documentation

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

Accessibility of Links to External Web Sites in Documentation

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

TTY Access to Oracle Support Services

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

Related Documents

For more information, refer to these Oracle resources:

- *Oracle Application Adapter Concepts*
- *Oracle Application Adapter for SAP R/3 Installation Guide (WebLogic Integration 10g Release 3)*

Printed documentation is available for sale in the Oracle Store at

<http://oraclestore.oracle.com/>

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

<http://www.oracle.com/technology/membership/>

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

<http://www.oracle.com/technology/documentation/>

Conventions

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

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

Conventions in Text

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

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

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

Conventions in Code Examples

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

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

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

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

| Convention | Meaning | Example |
|----------------|--|---|
| <i>Italics</i> | Italicized text indicates placeholders or variables for which you must provide particular values. | CONNECT SYSTEM/ <i>system_password</i> DB_NAME = <i>database_name</i> |
| UPPERCASE | Uppercase typeface indicates elements supplied by the system. We show these terms in uppercase to distinguish them from terms you define. Unless terms appear in brackets, enter them in the order and with the spelling shown. Because these terms are not case sensitive, you can use them in either UPPERCASE or lowercase. | SELECT last_name, employee_id FROM employees; SELECT * FROM USER_TABLES; DROP TABLE hr.employees; |
| lowercase | Lowercase typeface indicates user-defined programmatic elements, such as names of tables, columns, or files. Note: Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown. | SELECT last_name, employee_id FROM employees; sqlplus hr CREATE USER mjones IDENTIFIED BY ty3MU9; |

Conventions for Windows Operating Systems

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

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

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

Help Us to Serve You Better

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

The following list includes the specifications our consultants require.

- **Platform:**
- **Operating System:**
- **Operating System Version:**
- **Product List:**
- **Adapters:**
- **Adapter Deployment:**
For example, *J2CA* or *Business Services Engine (BSE)*
- **Container Version:**

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

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

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

| JVM Version | Vendor |
|-------------|--------|
| | |

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

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

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

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

Introduction

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

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

Adapter Features

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

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

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

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

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

channel component based on deployment requirements. In the case of this adapter, the channel is an RFC server.

Oracle Application Adapter for SAP R/3 provides:

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

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

See Also: *Oracle Application Adapter Concepts*

Supported Versions and Platforms

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

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

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

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

Classical SAP Technologies for ABAP

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

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

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

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

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

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

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

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

Integration with SAP R/3

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

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

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

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

- Elimination of the requirement for custom coding.
- Consistent data representation.
Provides a standard XML representation of event data and request/response documents for SAP R/3.
The developer is freed from the specific details of the SAP R/3 interface (BAPI, RFC, IDoc) and the specific configuration details of the target SAP R/3 system.
- Adherence to SAP R/3 ABAP serialization rules and SAP R/3 Interface Repository standards published by SAP AG.

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

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

Adapter Architecture

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

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

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

Oracle Adapter Business Services Engine (BSE) Architecture

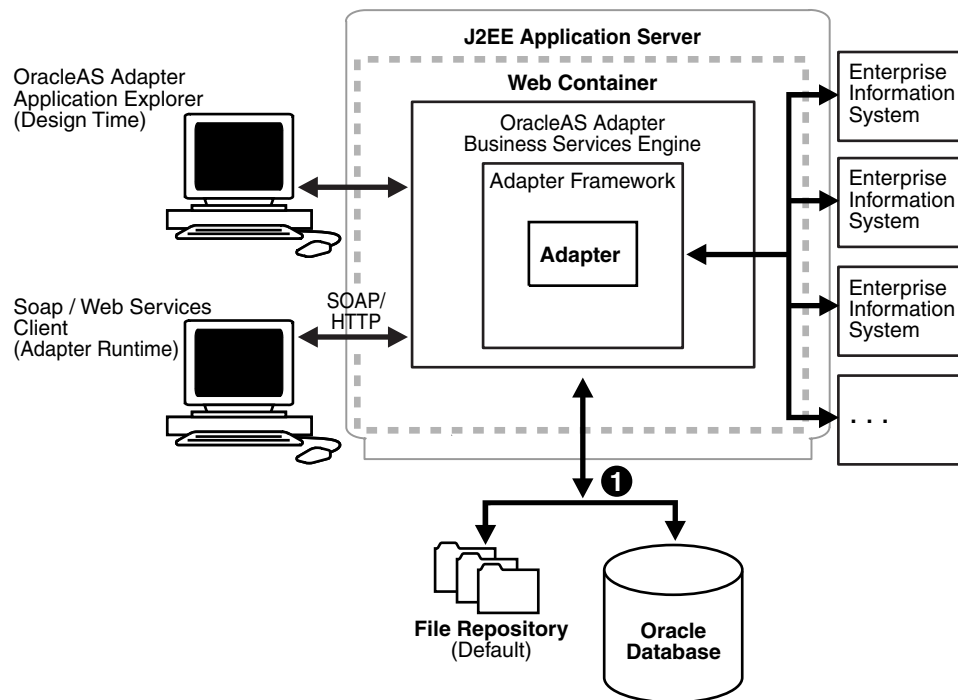
Figure 1–1 shows the generic architecture for BSE for packaged applications. The adapter works with BSE, as deployed to a Web container in a J2EE application server.

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

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

BSE supports both a file-based and an Oracle database repository. The BSE repository stores the SAP R/3 connection information and the Web Service Definition Language (WSDL) files that are used for services.. A single instance of BSE can connect to multiple SAP R/3 applications.

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

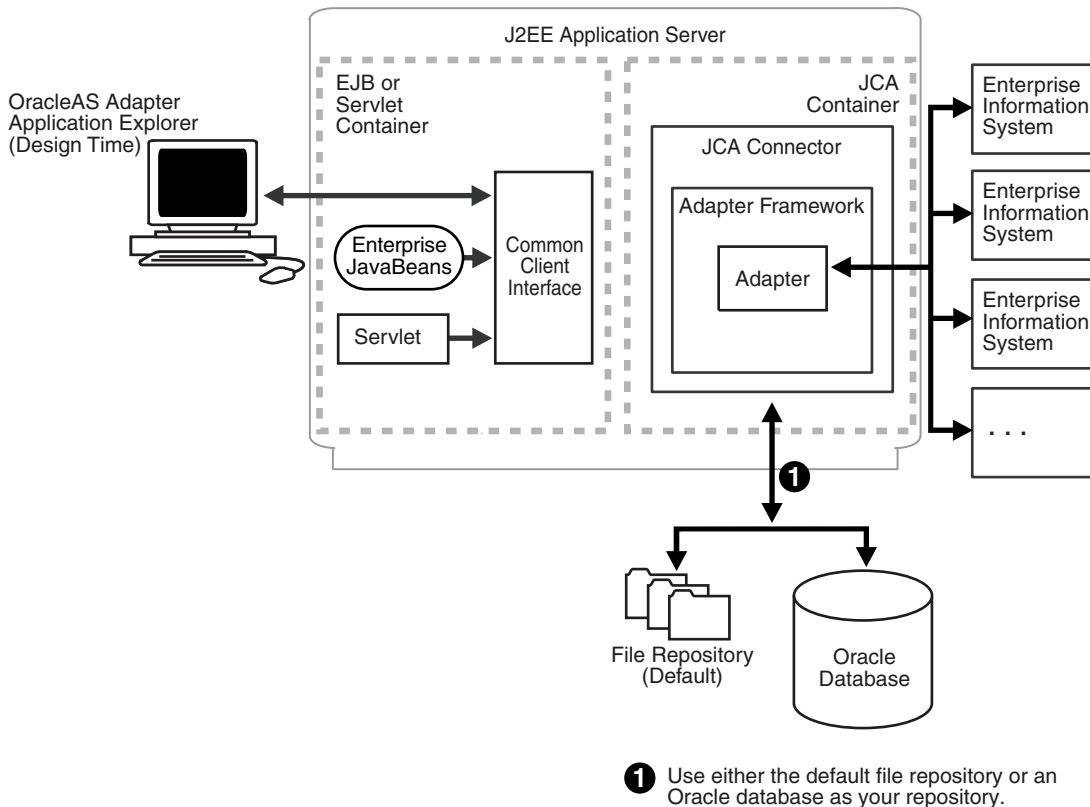
Figure 1–1 Oracle Adapter Business Services (BSE) Architecture

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

Oracle Adapter J2CA Architecture

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

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

Figure 1–2 Oracle Adapter J2CA Architecture**See Also:**

- *Oracle Application Adapter Concepts*
- *Oracle Application Adapter (WebLogic Integration 10g Release 3) Installation Guide*

BSE Versus Oracle Adapter J2CA Deployment

If you are using Oracle Application Adapter for SAP R/3 with Oracle WebLogic Integration (WLI), please note that:

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

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

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

- Conforms more closely to the Service Oriented Architecture (SOA) model for building applications.
- 2. Oracle Adapter J2CA provides slightly better performance.
Oracle Adapter J2CA does provide slightly better performance than BSE. However, the difference decreases as the transaction rate increases.
- 3. Oracle Adapter J2CA and the BSE option both provide identity propagation at run-time.
The BSE option provides the capability to pass identity using the SOAP header. For Oracle Adapter J2CA, user name and password can be passed using the connection specification of the CCI.

Installation Considerations

For information on adapter prerequisites, including how to install and configure the Oracle Application Adapter for SAP R/3, see the *Oracle Application Adapter (WebLogic Integration 10g Release 3) Installation Guide*.

Configuring Oracle Application Adapter for SAP R/3

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

This chapter discusses the following topics:

- [Starting Application Explorer](#)
- [Establishing a Connection \(Target\) for SAP R/3](#)
- [Viewing Application System Objects](#)
- [Viewing XML Schemas](#)
- [Generating WSDL \(J2CA Configurations Only\)](#)
- [Creating and Testing a Web Service \(BSE Configurations Only\)](#)
- [Configuring an Adapter Channel](#)

Starting Application Explorer

To start Application Explorer:

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

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

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

Establishing a Connection (Target) for SAP R/3

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

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

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

Defining a Target to SAP R/3

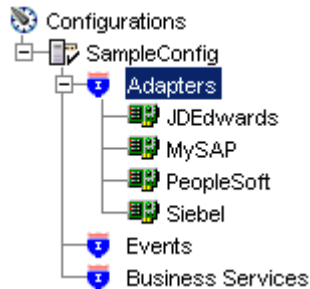
To connect to SAP R/3 for the first time, you must define a new target. Oracle Application Adapter for SAP R/3 supports SAP R/3 standard security and the additional protocol of SNC. Once connected to the SAP R/3 application server, application security is managed by user ID, roles and profiles. For more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

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

To define a target:

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

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



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



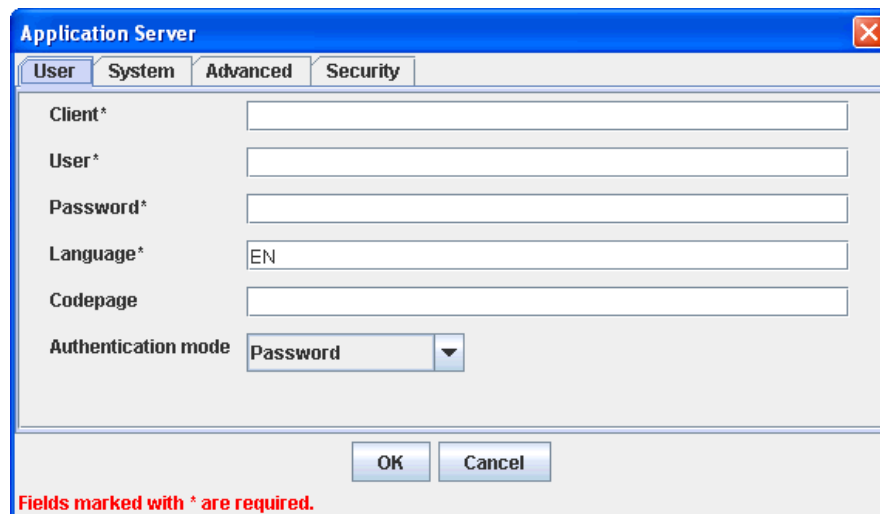
The Add Target dialog box is displayed. Provide the following information:

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

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

1. Click **OK**.

The Application Server dialog box is displayed.



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

The following tabs are available:

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

Table 2–1 User Tab Parameters

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

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

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

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

Table 2–2 System Tab Parameters

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

Table 2–2 (Cont.) System Tab Parameters

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

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

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

Table 2–3 Advanced Tab Parameters

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

Table 2–3 (Cont.) Advanced Tab Parameters

| Target Parameter | Description |
|------------------|---|
| Error Handling | <p>From the list In the event of an exception, you can select Creates Error Document or Throws Exception. To receive more detailed error messages, select Creates Error Document.</p> <p>As a rule:</p> <ul style="list-style-type: none"> ■ If your application is Java centric, select Throws Exception so that code components can catch the exception and react accordingly. ■ If your application is document based, select Creates Document to create an XML document that contains the Java exception. <p>It is up to your application to read the XML document and obtain the error.</p> |
| Commit with Wait | <p>If a high degree of accuracy is required in your application, select the Commit with Wait check box.</p> <p>The adapter waits until all records are physically written to the database before returning from the function call. The “Commit With Wait” has a performance impact on adapter performance, so consider carefully before selecting it. For more information about the commit behavior of BAPIs, see the SAP documentation under “BAPI Programming Guide and Reference (CA-BFA).”</p> <p>All SAP Business Objects that change data must commit work to the database. Some BAPIs developed in version 3.1 of the R/3 system use an internal commit behavior, and their commit behavior cannot be changed by the adapter. As soon as they are called, they commit the work they did.</p> <p>BAPIs developed since release 3.1 use the external commit method. The adapter issues a commit command, and the commit is put in the database queue. If there is an application error in the first part of the commit, the error message “Posting could not be carried out” is returned, and the adapter rolls back the transaction. If in writing to the database, a database error occurs, a short dump is issued in the database records of SAP, but no message is returned to the adapter about the failure.</p> <p>This option is disabled by default.</p> |
| SAP trace | Select this option to enable traces. |

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

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

Table 2–4 Security Tab Parameters

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

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

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

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

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

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



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

See "[Viewing XML Schemas](#)" on page 2-10 for information on how to view schemas for the adapter.

Connecting to a Defined SAP R/3 Target

To connect to an existing target:

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

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

4. Verify your connection parameters.
5. Enter the same password that was specified when this target was created.
6. Right-click the target name and select Connect.

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



Managing a Connection to SAP R/3

To manage SAP R/3 connections, you can:

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

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

- Edit a target.

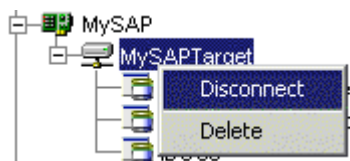
You can modify the connection parameters when your system properties change. After you disconnect, you can modify an existing target.

- Delete a connection that is no longer needed.

Disconnecting from a Connection to SAP R/3

To disconnect a target:

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



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

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

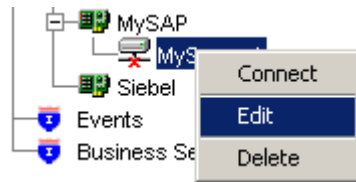


Modifying Connection Parameters

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

To edit a target:

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



The Application Server dialog box displays the target connection information.

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

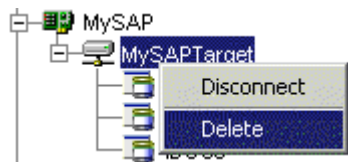
Deleting a Connection to SAP R/3

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

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

To delete a connection to SAP R/3:

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



The node disappears from the list of available connections.

Viewing Application System Objects

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

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

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

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

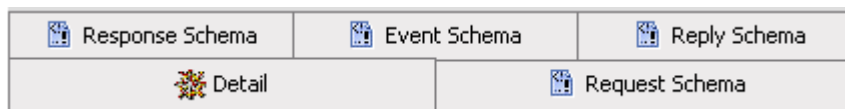
Viewing XML Schemas

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

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

1. Connect to a SAP R/3 target as described in "[Connecting to a Defined SAP R/3 Target](#)" on page 2-8.
2. Expand the **Business Object Repository** node.
3. Click the icon to the left of the **Financial Accounting** node.
4. Scroll down and click the icon to the left of the **CompanyCode** business object.
5. Scroll down and select the BAPI named `GetDetail`.

The following screen appears on the right.



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

Generating WSDL (J2CA Configurations Only)

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

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

Generating a WSDL File for a Request-Response (Outbound) J2CA Service

To generate a WSDL file for a request-response (outbound) J2CA service:

1. After you create a schema, right-click the respective object and select **Create Outbound JCA Service (Request/Response)**.

The Export WSDL dialog box is displayed.

The screenshot shows the 'Export WSDL' dialog box. The 'Name' field contains 'GetDetail_invoke.wsdl'. Under 'Element Form Default', the 'qualified' radio button is selected, and the 'unqualified' radio button is unselected. At the bottom, there are 'OK' and 'Cancel' buttons.

2. Accept the default name for the file.

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

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

The WSDL file is saved and stored in a sub folder of your J2CA configuration directory.

Generating a WSDL File for an Event Notification (Inbound) J2CA Service

To generate a WSDL file for an event notification (inbound) J2CA service:

1. After you create a schema, right-click the respective object and select **Create Inbound JCA Service (Event)**.

The Export WSDL dialog box is displayed.

The screenshot shows the 'Export WSDL' dialog box for an inbound service. The 'Name' field contains 'mysap_MATMASE01_receive_nov9.wsdl'. Under 'Element Form Default', the 'qualified' radio button is selected. The 'Channel' dropdown is set to 'mysap_ch' and the 'Port' dropdown is set to 'no port'. Under 'Validation', the 'Root', 'Namespace', and 'Schema' checkboxes are all unselected. At the bottom, there are 'OK' and 'Cancel' buttons. A red message at the bottom of the dialog reads: '* You must create a separate channel for each inbound service'.

2. Accept the default name for the file.
The .wsdl file extension is added automatically. By default, the names of WSDL files generated for event notification end with `_receive`.
3. Ensure that qualified is selected as the element form, which is the default.
4. Select an available adapter channel from the Channel drop-down list.
5. Select no port from the Port drop-down list.
6. Select the validation type (if required) by selecting either Root, Namespace, or Schema.
7. Click OK.

The WSDL file is saved and stored in a sub folder of your J2CA configuration directory.

Creating and Testing a Web Service (BSE Configurations Only)

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

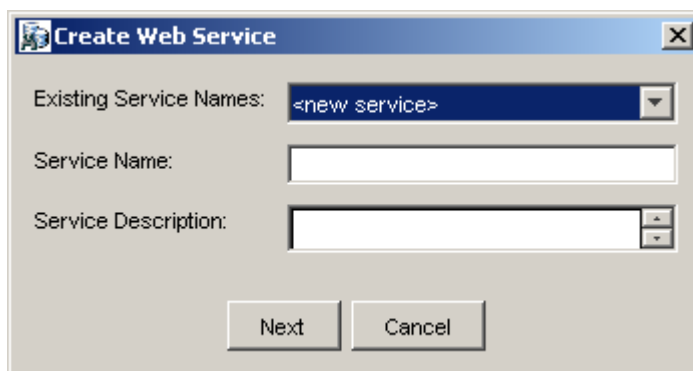
Note: In a J2EE Connector Architecture (J2CA) implementation of the adapter, Web services are not available. When the adapter is deployed to use the Oracle Adapter J2CA, the Common Client Interface provides integration services using the adapter.

Creating a Web Service

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

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

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



Perform the following steps:

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

1. Click **Next**.

The License and Method dialog box is displayed.

Provide the following information:

- a. In the **License Name** field, select one or more license codes to assign to the Web service. To select more than one, hold down the Ctrl key and click the licenses.
- b. In the **Method Name** field, enter a descriptive name for the method.
- c. In the **Method Description** field, enter a brief description of the method.
- d. In the DTD Directory field, specify the location of the DTD you want to use.

2. Click **OK**.

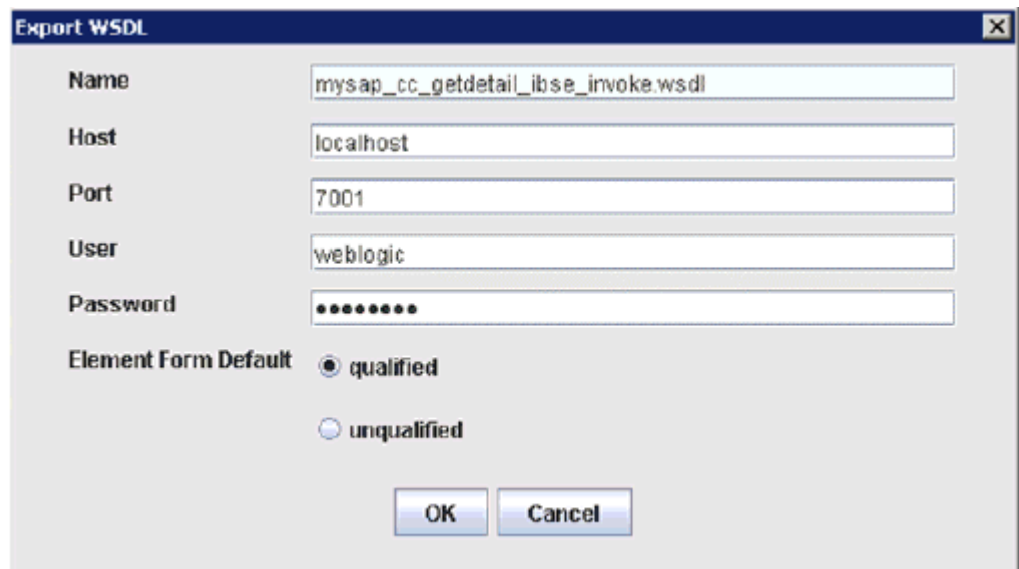
Application Explorer switches the view to the **Business Services** node, and the new Web service appears in the left pane.

Generating a WSDL File for a Request-Response (Outbound) BSE Service

To generate a WSDL file for a request-response (outbound) BSE service:

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

The Export WSDL dialog box is displayed.



2. Accept the default name for the file.

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

3. In the Host field, enter the name of the machine where Oracle WebLogic Integration is running.
4. In the Port field, enter the port for the domain you are using.
5. In the User field, enter your username to access Oracle WebLogic Integration.
6. In the Password field, enter your password to access Oracle WebLogic Integration.
7. Ensure that qualified is selected as the element form, which is the default.
8. Click **OK**.

The WSDL file is saved and stored in a sub folder of your BSE configuration directory

Testing a Web Service

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

To test a Web service:

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

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

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

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

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

The BSE Servlet page displays the results.

Identity Propagation

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

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

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

Oracle ERP Adapter WSDL Definition Limitations

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

Qualified/Unqualified Options

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

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

A correct response document is received.

Namespaces

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

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

A correct response document is received.

Unchecked Data

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

1. Create a Web service for the `RFC_SYSTEM_INFO` function module.
2. Invoke this Web service (`RFC_SYSTEM_INFO`) with XML data pertaining to `BAPI_COMPANYCODE_GETLIST`.

A response document is received with data for `BAPI_COMPANYCODE_GETLIST`.

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

Configuring an Adapter Channel

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

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

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

An **adapter channel** represents configured connections to particular instances of back-end systems. See "[Creating and Editing an Adapter Channel](#)" on page 2-16 for more information.

Creating and Editing an Adapter Channel

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

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

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

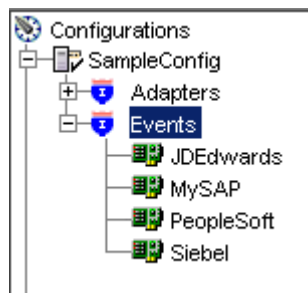
Events are not applicable when using a BSE configuration. You can configure events using a J2CA configuration only.

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

Creating an Adapter Channel

To create an adapter channel:

1. Click the **Events** node.



2. Expand the **MySAP** node.
The ports and channels nodes appear in the left pane.
3. Right-click **Channels** and select **Add Channel**.

The Add Channel dialog box is displayed.

Provide the following information:

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

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

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

Table 2–5 User Tab Parameters

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

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

Table 2–6 System Tab Parameters

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

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

Table 2–7 Security Tab Parameters

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

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

Table 2–8 Advanced Tab Parameters

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

- Click **OK**.

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



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

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

The adapter channel you created becomes active.



The X over the icon disappears.

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

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

Editing an Adapter Channel

To edit an adapter channel:

1. In the left pane, locate the adapter channel you want to edit.
2. Right-click the adapter channel and select **Edit**.
The Edit Channel pane is displayed.
3. Make the required changes to the adapter channel configuration and click **Finish**.

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

Deleting an Adapter Channel

To delete an adapter channel:

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

The adapter channel disappears from the list in the left pane.

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

Configuring Outbound Processing Using Oracle WebLogic Integration (J2CA Configuration)

To integrate with Oracle WebLogic Integration (WLI), Oracle Application Adapter for SAP R/3 must be deployed on the same Oracle WebLogic Server as WLI. The underlying adapter services must be exposed as WSDL files, which are generated during design time in Oracle WebLogic Server Adapter Application Explorer (Application Explorer) for request-response (outbound) services of the adapter. Oracle Application Adapter for SAP R/3 integrates seamlessly with WLI to facilitate Web service integration.

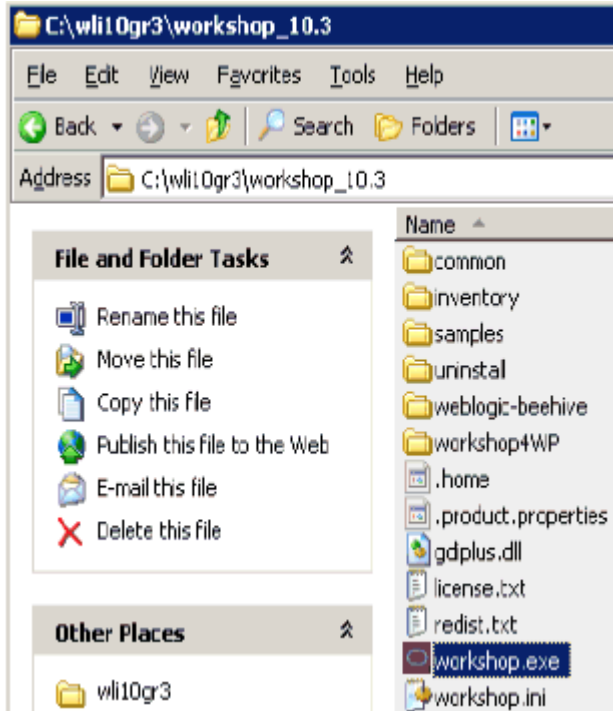
This chapter includes the following topics:

- [Creating an Outbound JPD Process Application](#)
- [Deploying the Outbound Process on WebLogic Server](#)
- [Invoking the Input XML Using a Web Browser](#)

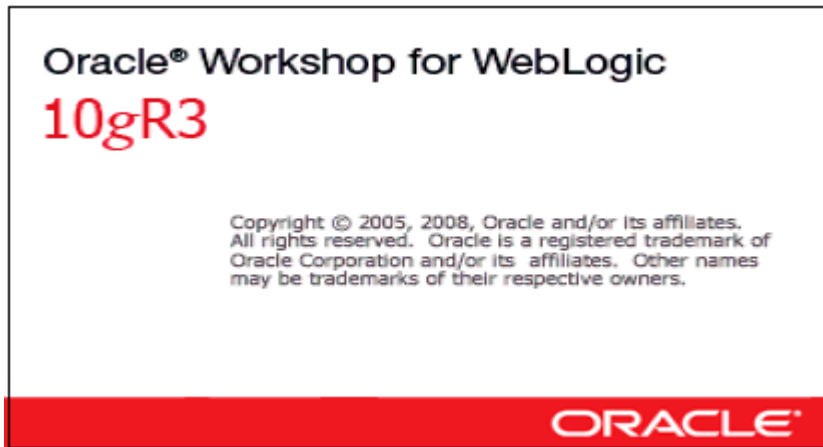
Creating an Outbound JPD Process Application

This section describes how to create an outbound JPD process application.

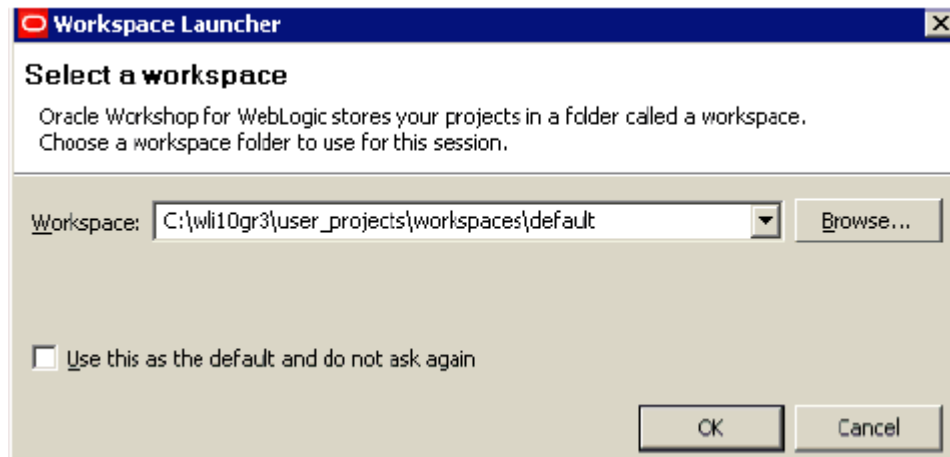
1. Navigate to the WebLogic Integration Workshop location (for example, c:\wli10gr3\workshop_10.3) and double-click the **workshop.exe** file.



The Oracle Workshop for WebLogic screen is displayed.



This is followed by the Workspace Launcher dialog box, as shown in the following image.



2. Specify the workspace location or accept the default path, which is:

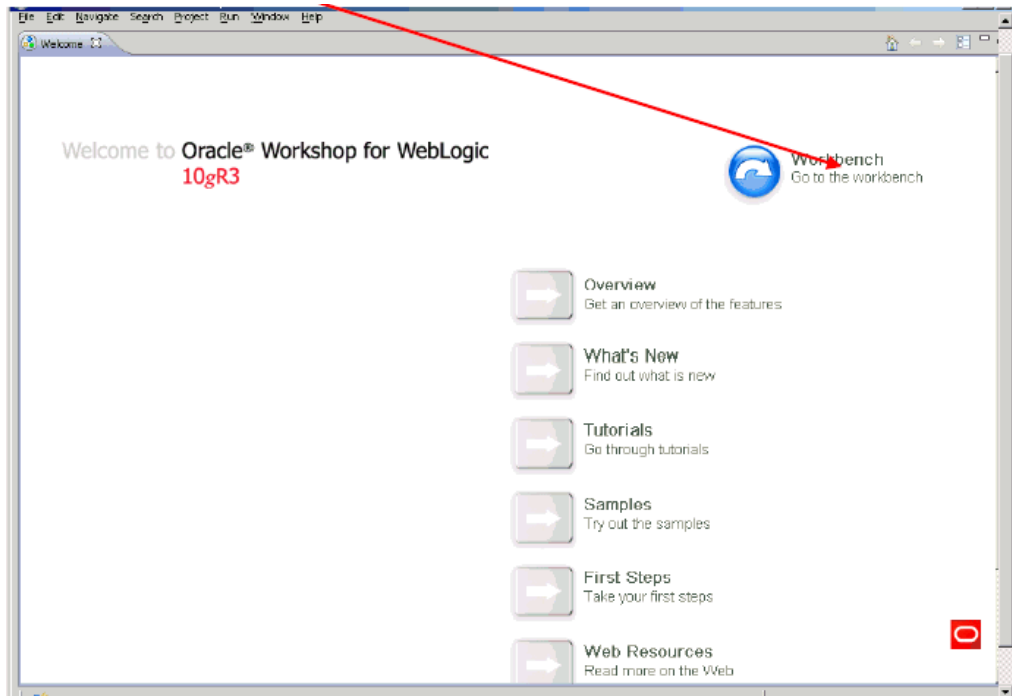
C:\wli10gr3\user_projects\workspaces\default

3. Click **OK**.

The Oracle Workshop for WebLogic loading screen is displayed.

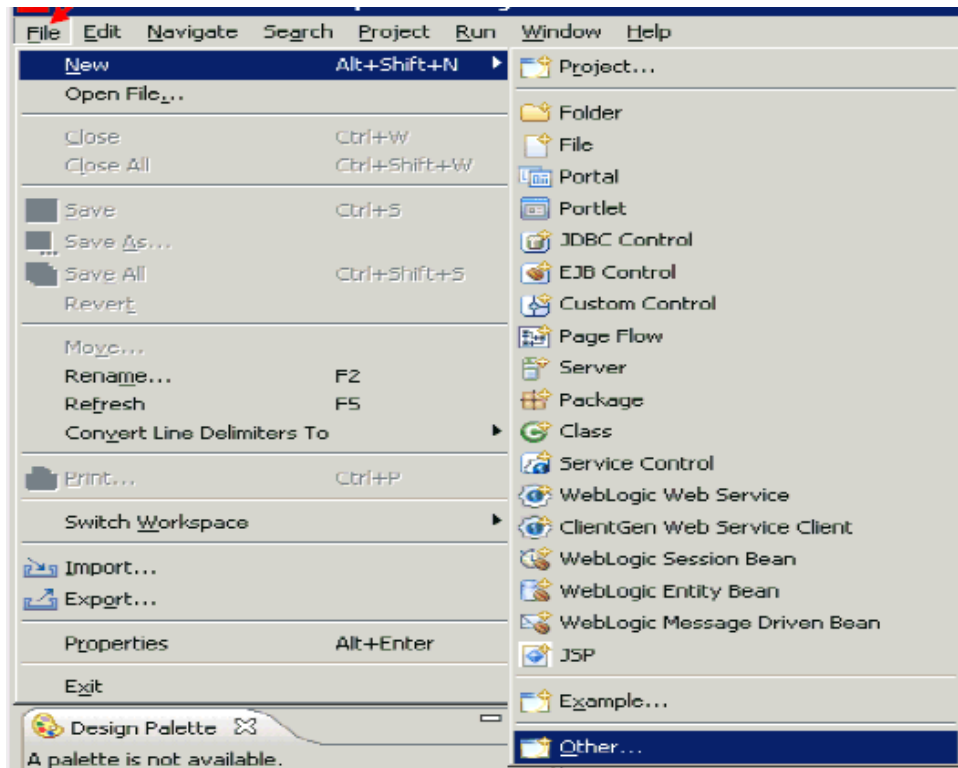


This is followed by the Oracle Workshop for WebLogic Welcome screen, as shown in the following image.



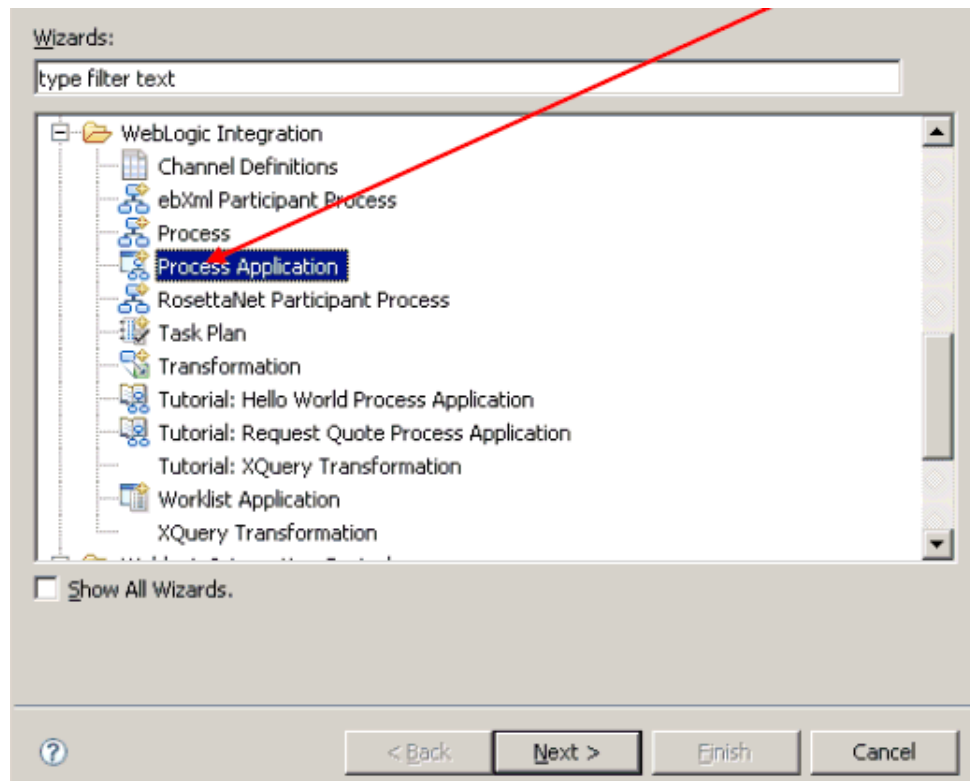
- 4. Click **Workbench**.

The Workbench area is displayed.



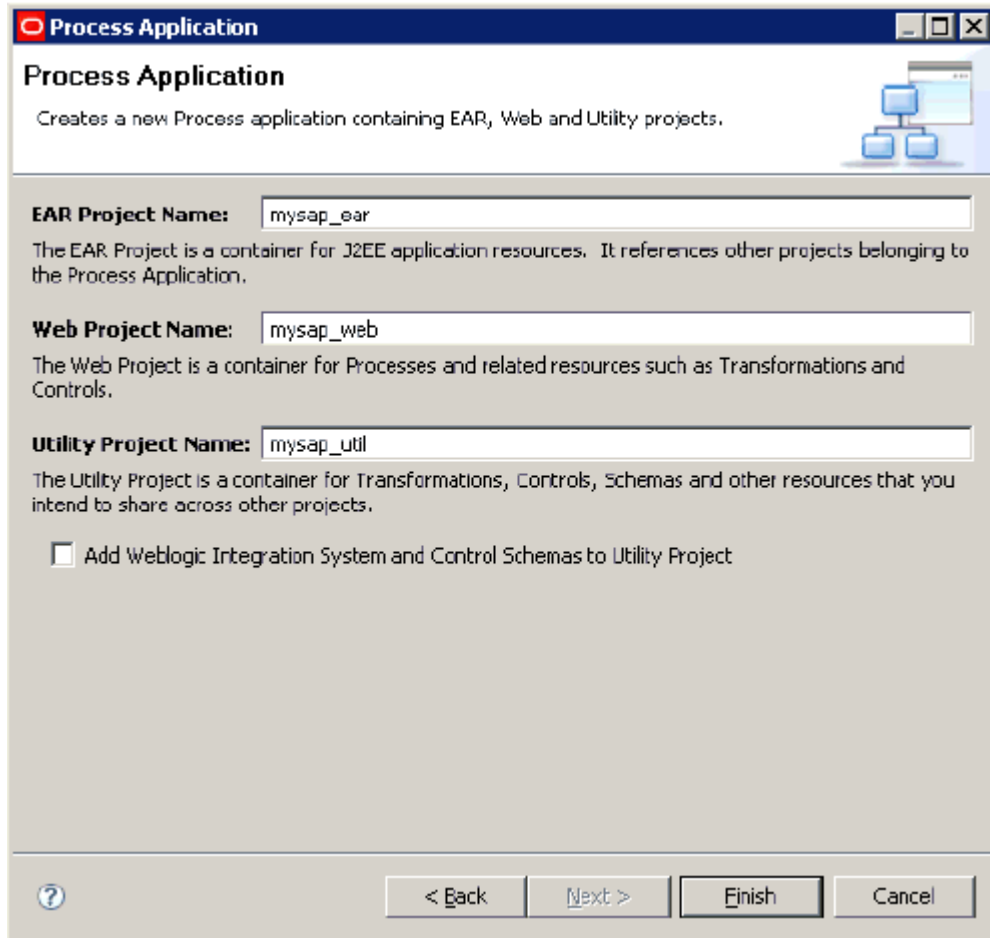
- 5. From the File menu, select **New** and click **Other**.

The following dialog box is displayed.



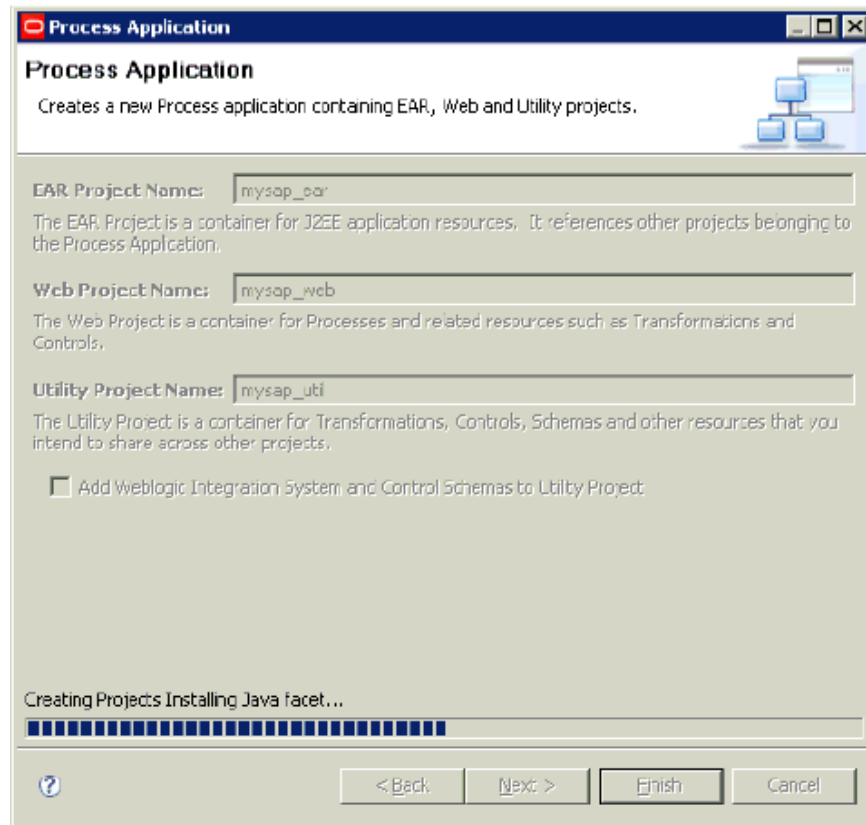
6. Expand the **WebLogic Integration** folder, select **Process Application**, and click **Next**.

The Process Application dialog box is displayed.

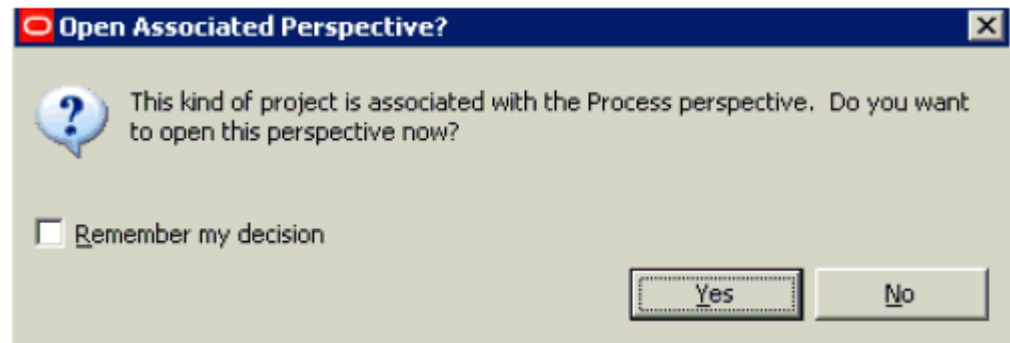


7. Perform the following steps:
 - a. Specify an EAR project name, for example, **mysap_ear**.
 - b. Specify a Web project name, for example, **mysap_web**.
 - c. Specify a Utility project name, for example, **mysap_util**.
8. Click **Finish**.

The EAR, Web, and Utility projects are created by a background process, as shown by the following status indicator.

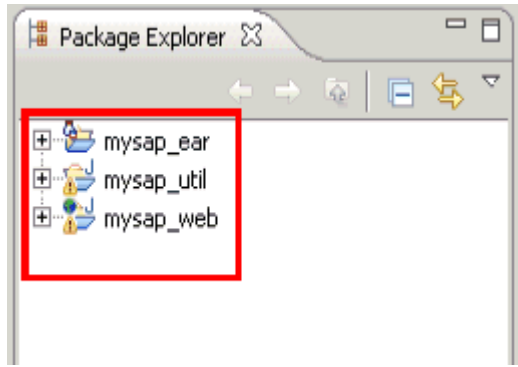


After this process is complete, the following message is displayed.

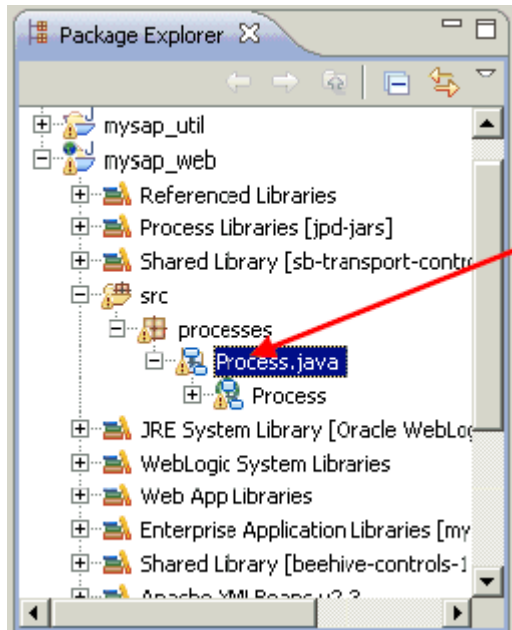


9. Click Yes to open the associated perspective.

The EAR, Web, and Utility projects are created in the Package Explorer area in the left pane.

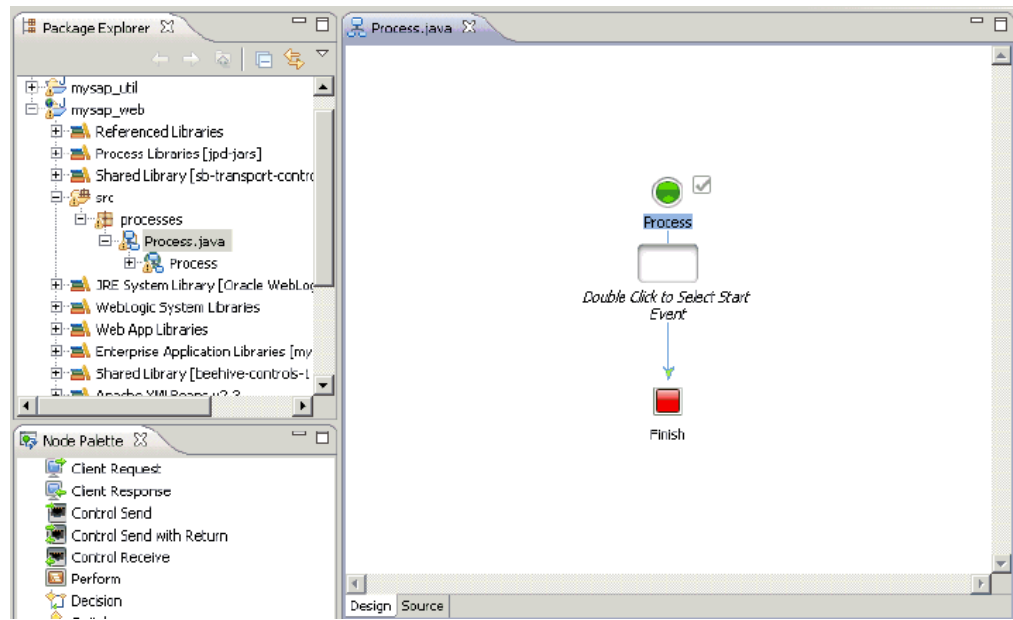


10. Expand the Web project node (for example, **mysap_web**), followed by **src** and **processes**.

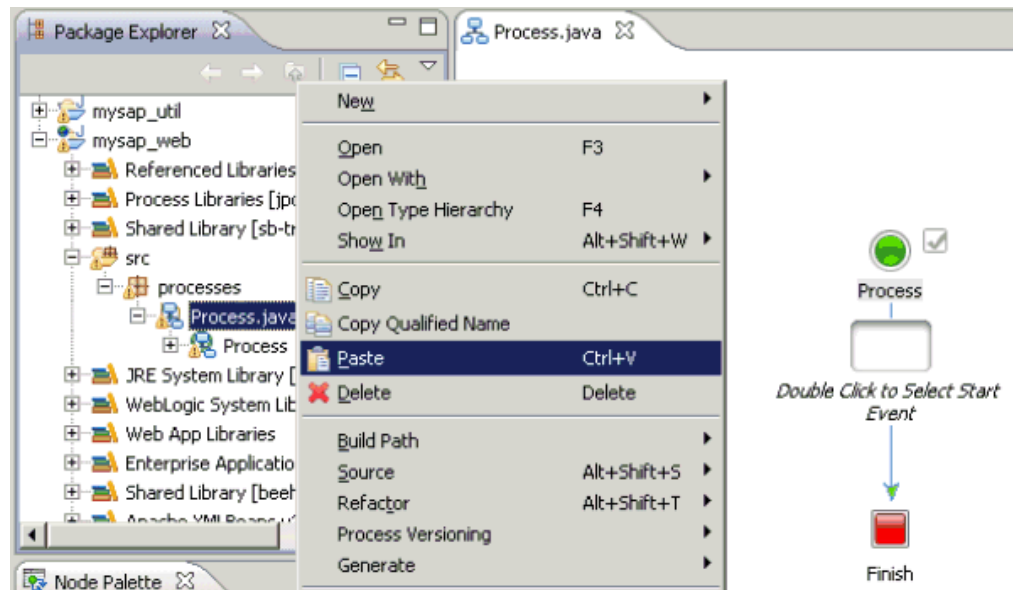


11. Double-click **Process.java**.

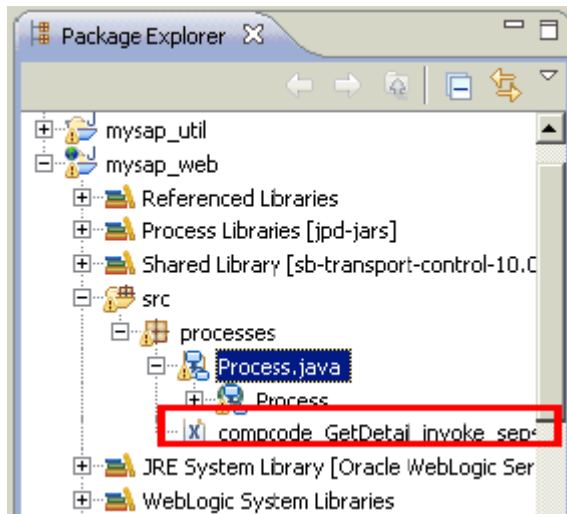
The process design view (**Process.java**) is displayed in the central pane.



12. Copy and paste the generated outbound WSDL file that was created using Application Explorer under the Process.java node.
For more information, see [Generating WSDL \(J2CA Configurations Only\)](#).

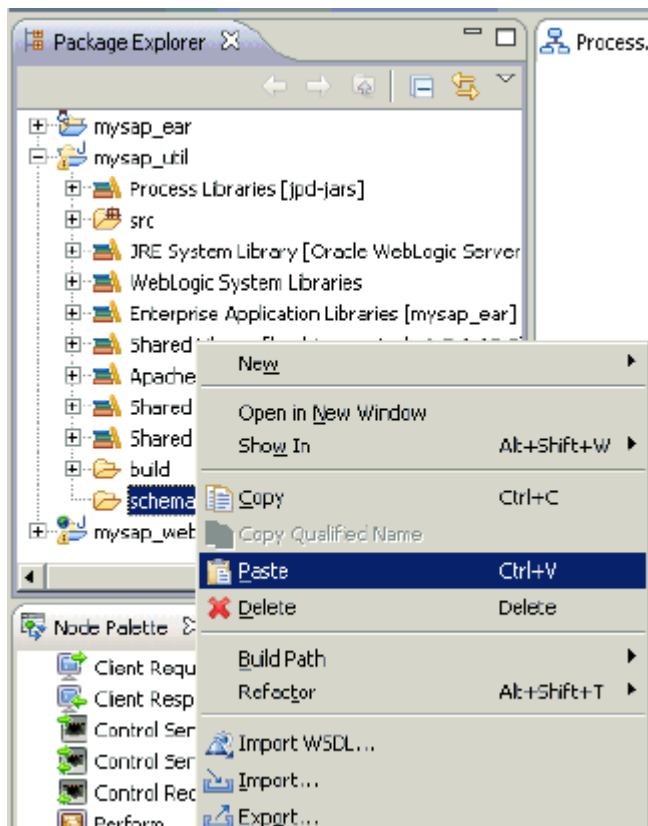


The copied outbound WSDL file is listed under the Process.java node, as shown in the following image.

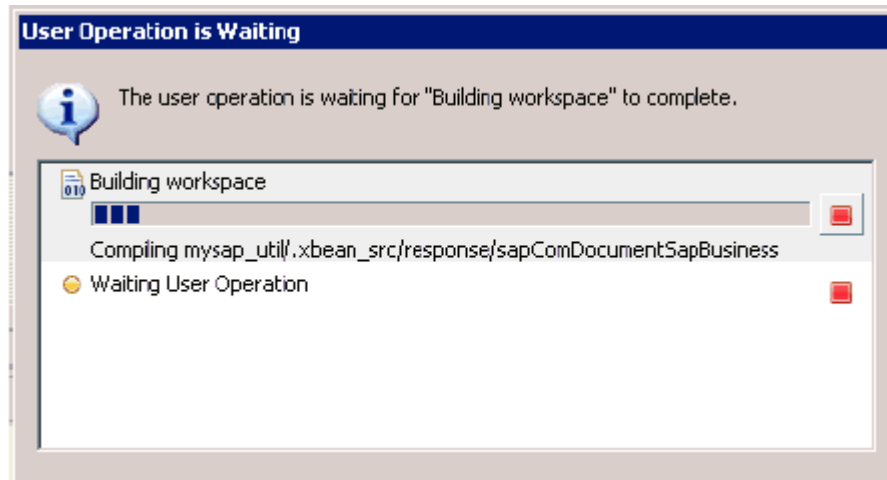


Note: WSDL files and XML schemas (.xsd files) are not required to be specifically placed in these folders. They can be located or placed in different locations. However, in these cases, the WSDL file must be updated with the relative path of the .xsd file.

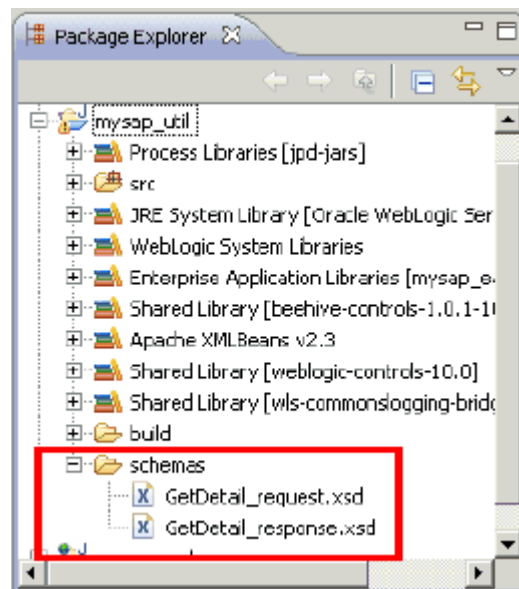
13. Expand the Utility project node (for example, **mysap_util**), and select the schemas folder.
14. Copy and paste the generated request and response XML schemas that were created using Application Explorer in the schemas folder.



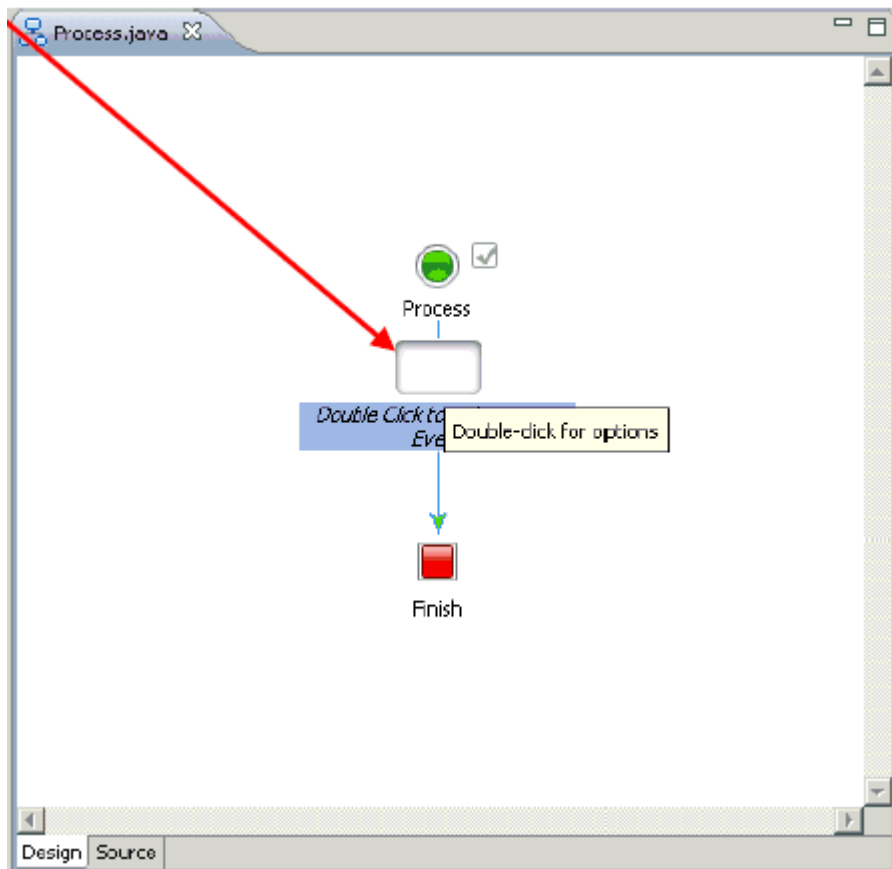
The new workspace is created by a background process, as shown by the following status indicator.



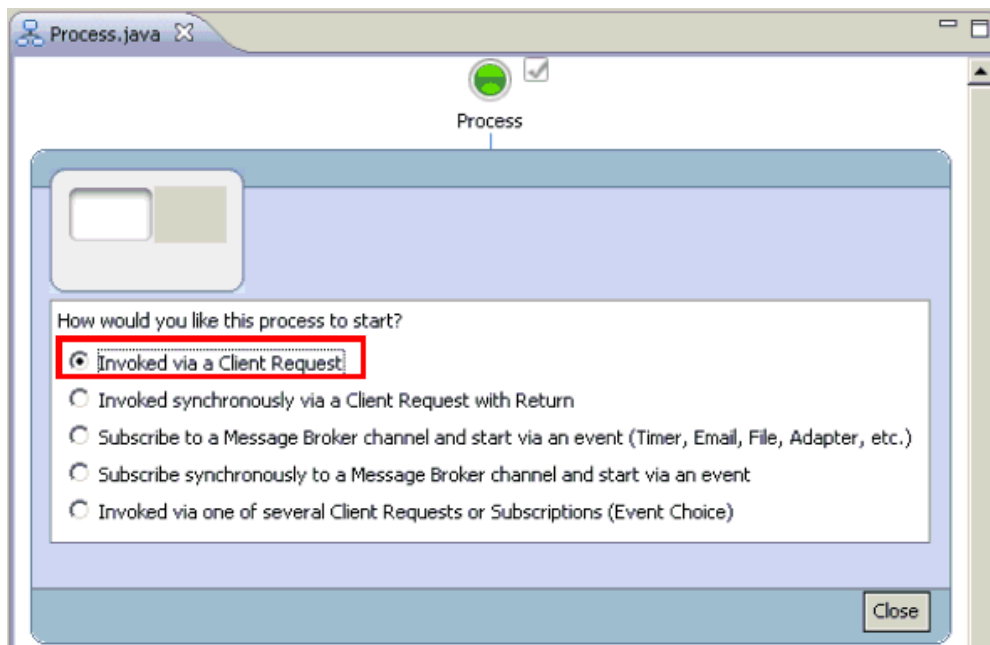
After this process is complete, the copied schemas are listed under the schemas folder.



15. Double-click the main node in the process design view (Process.java), as shown in the following image.

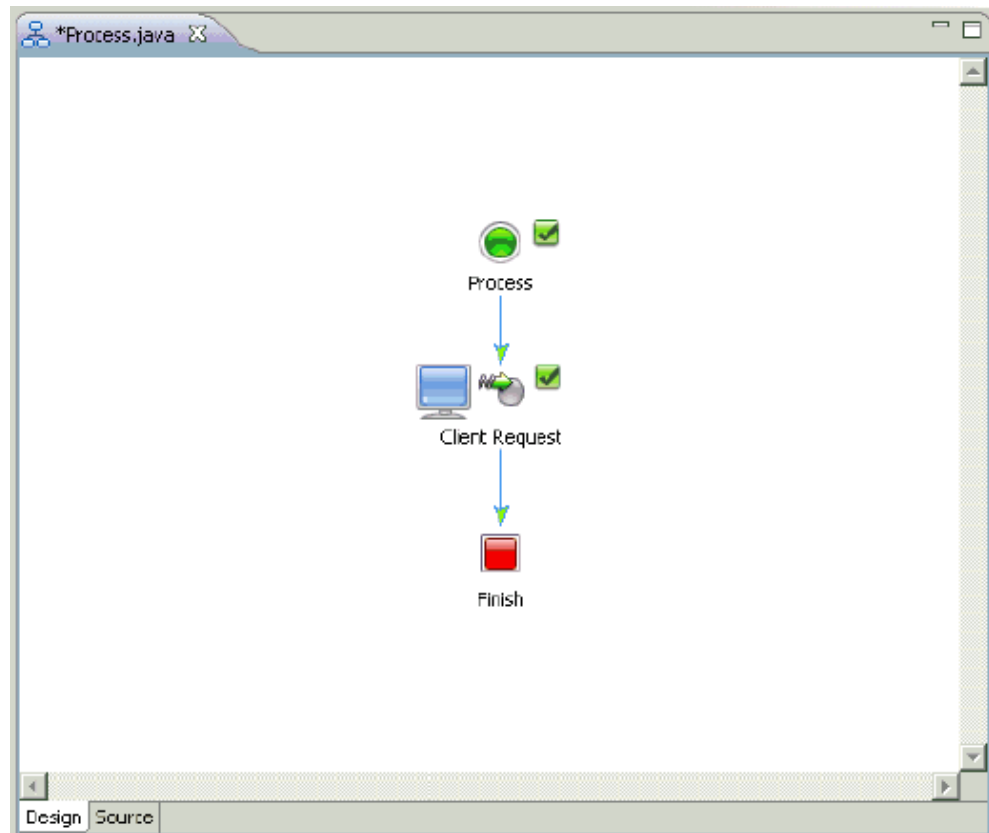


The following dialog box is displayed.

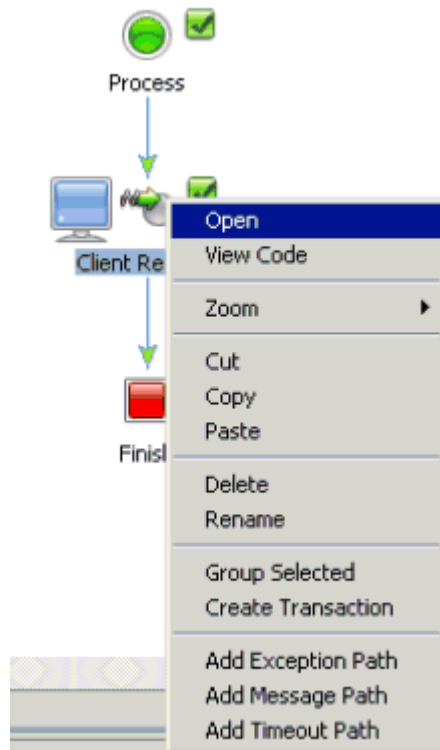


16. Select Invoked via a Client Request and click Close.

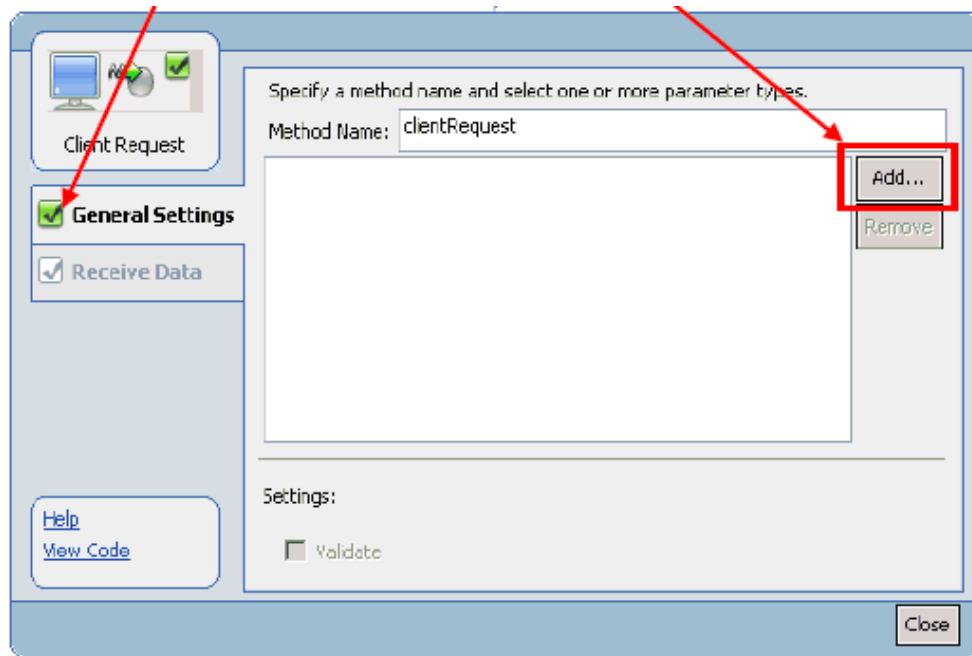
The Client Request node is added to the process design view (Process.java).



17. Right-click the Client Request node and select Open from the context menu.

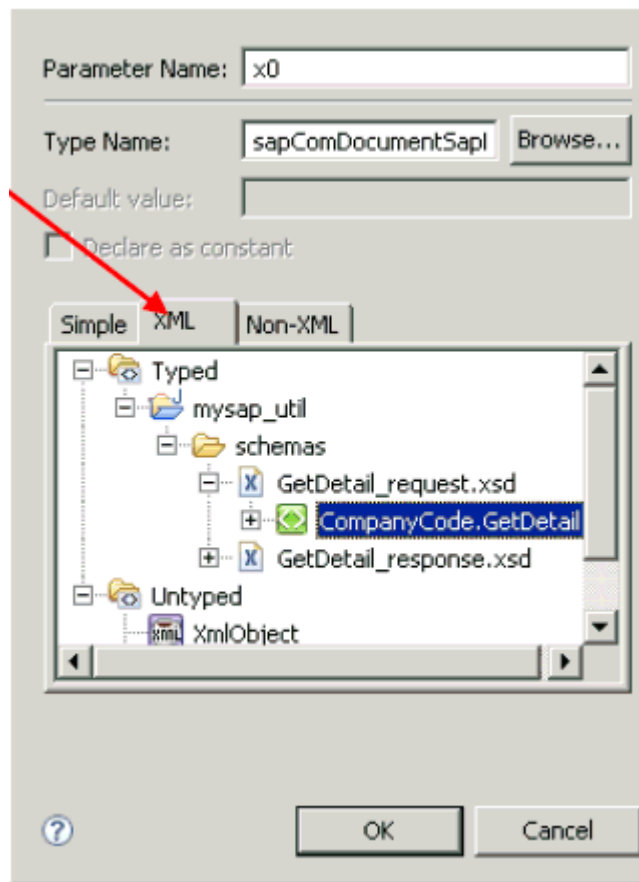


The Client Request configuration properties dialog box is displayed.



18. Click General Settings in the left pane and then click Add.

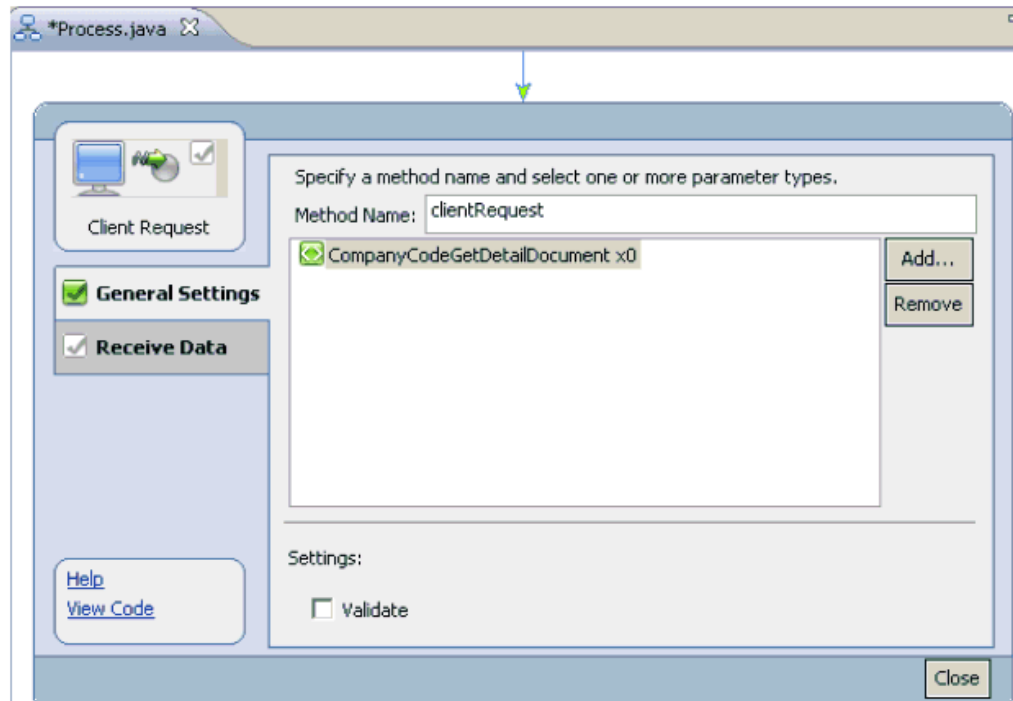
The following dialog box is displayed.



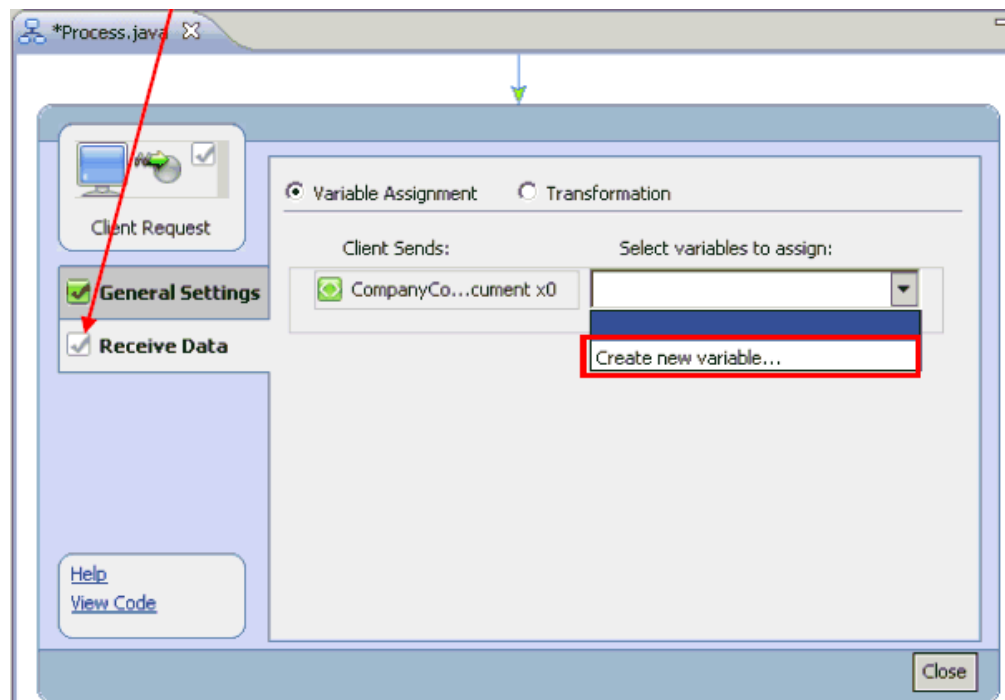
19. Under the XML tab, expand Typed, followed by mysap_util, schemas, GetDetail_request.xsd, and select CompanyCode.GetDetail.

20. Click OK.

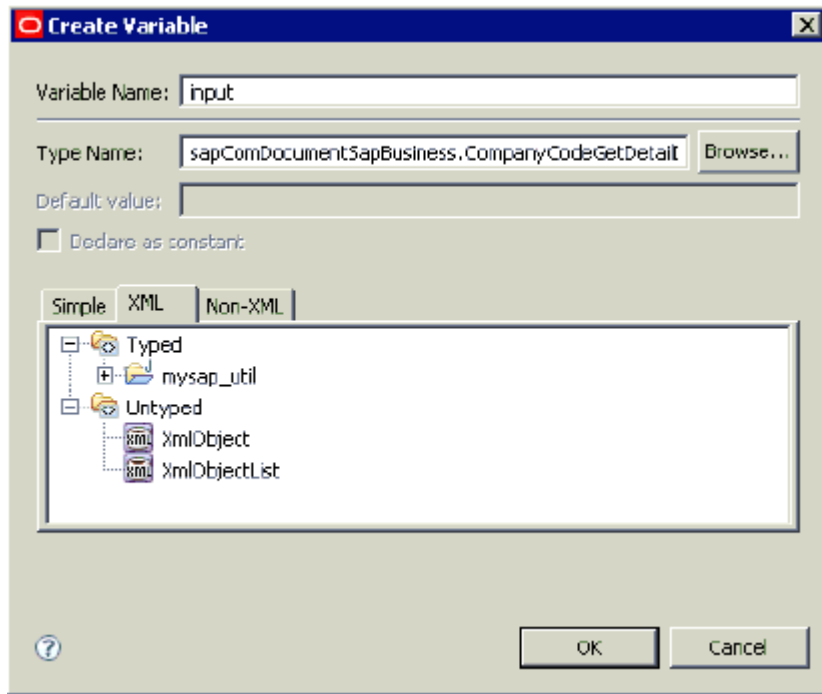
You are returned to the Client Request configuration properties dialog box.



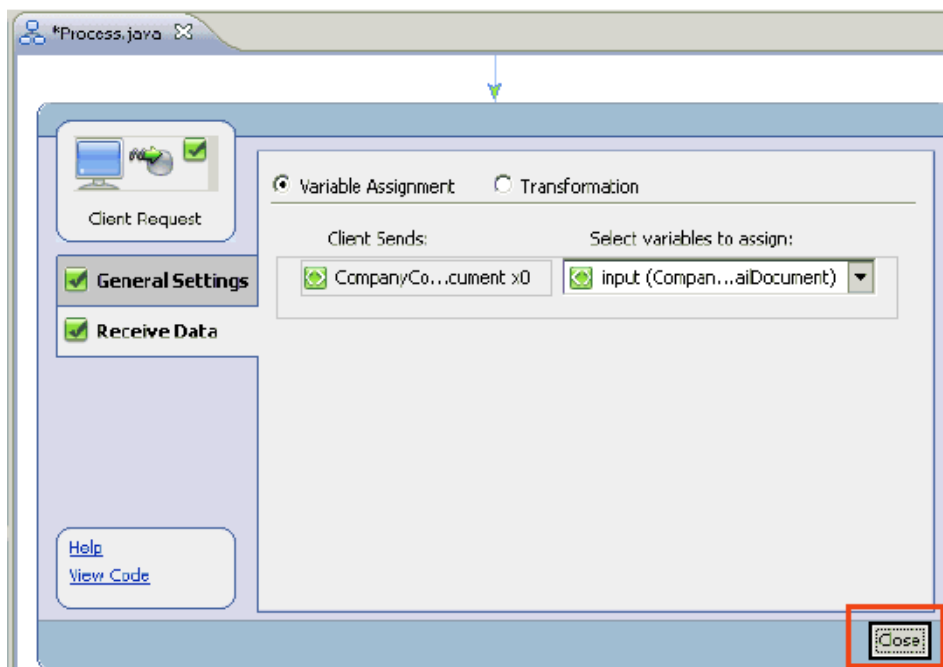
21. Click Receive Data in the left pane and select Create new variable from the drop-down list.



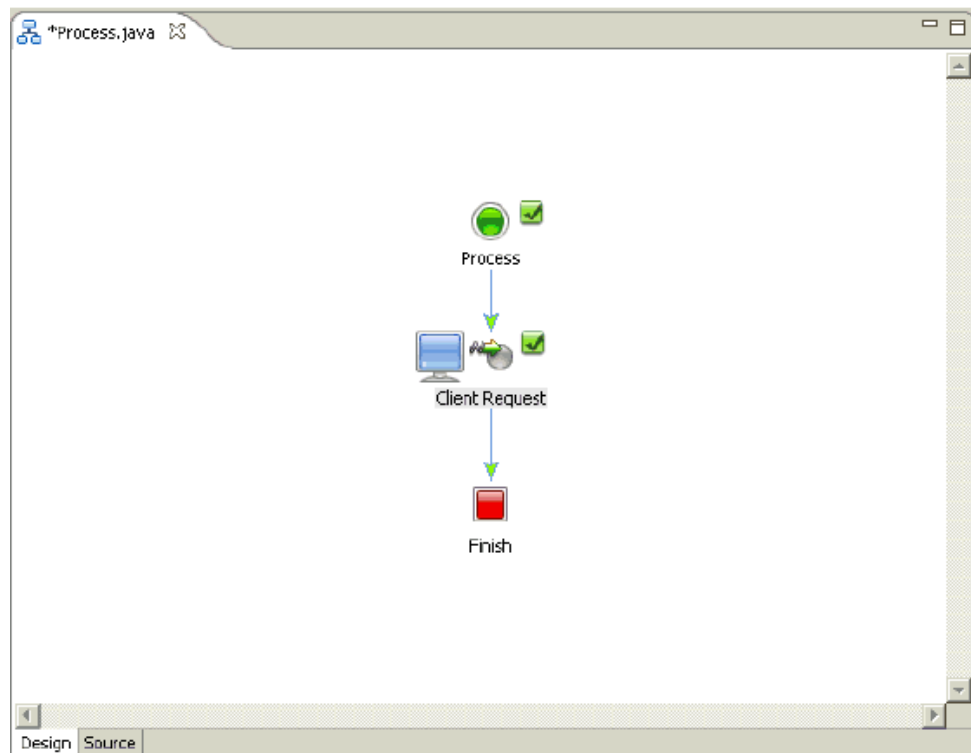
The Create Variable dialog box is displayed.



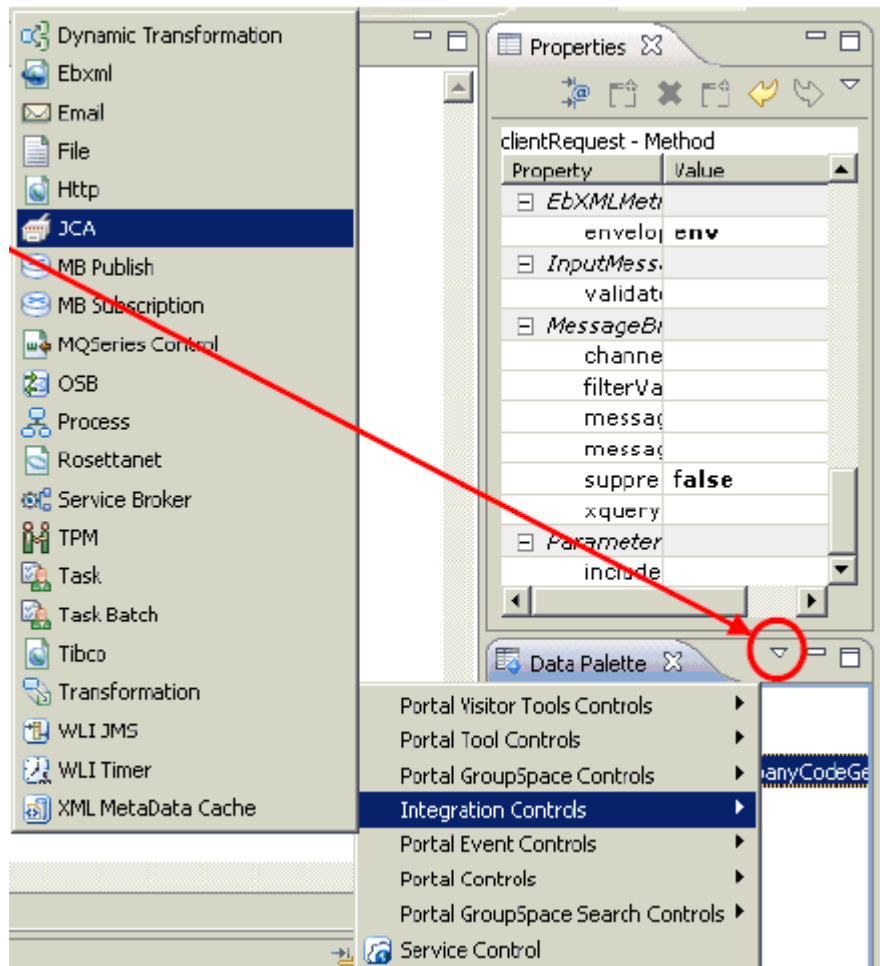
22. Specify a new variable name for the input data for example, input, and click OK.
 You are returned to the Client Request configuration properties dialog box, which has been refreshed with the new information.



23. Click Close.
 You are returned to the process design view (Process.java).



24. Click the down arrow in the Data Palette area, select Integration Controls, and select JCA.



The Insert Control: JCA Control dialog box is displayed.

Insert Control : JCA Control

Field Name
Enter the name of the field for this control's declaration

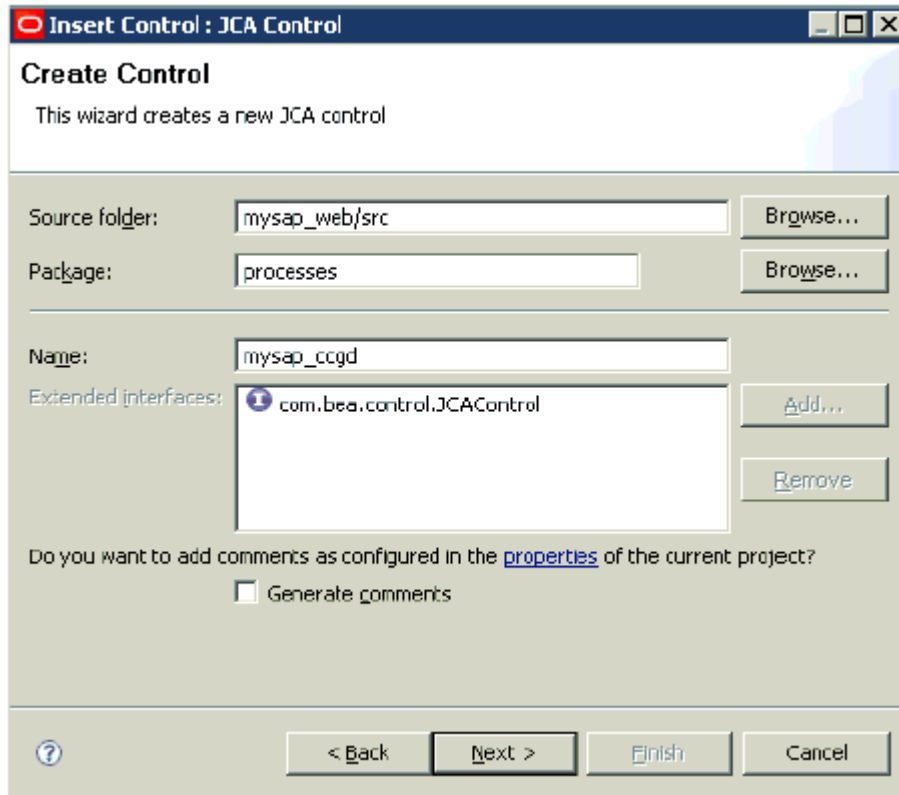
Field Name:

Insertion point:

Make this a control factory that can create multiple instances at runtime

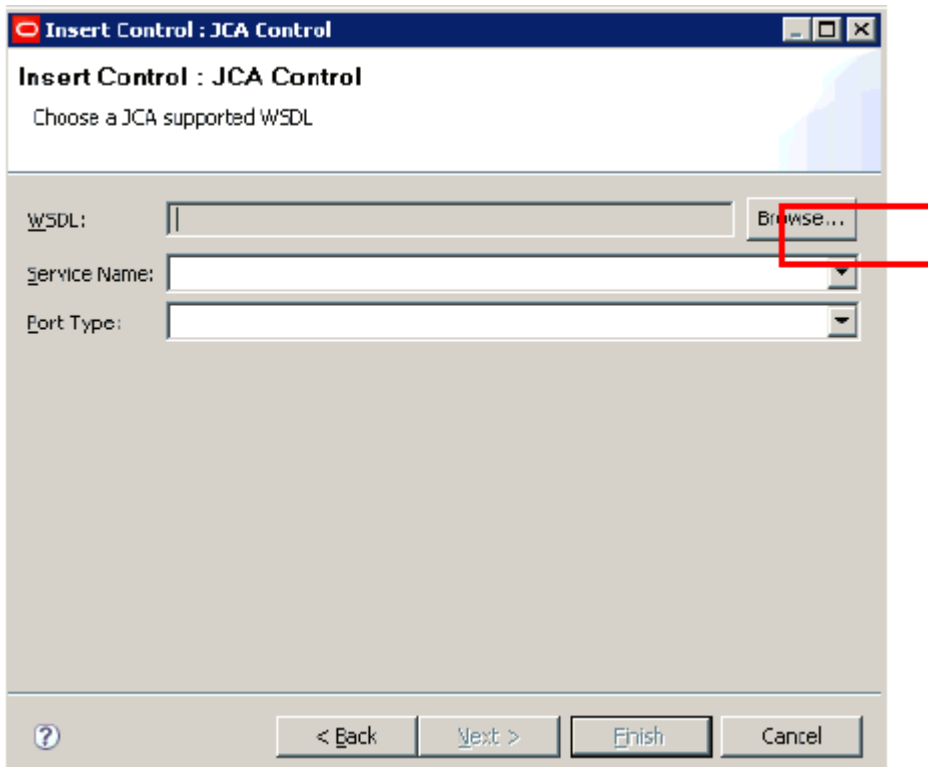
? < Back Next > Finish Cancel

25. Specify the field name (for example, JCAControl1) and click Next.
The Create Control pane is displayed.



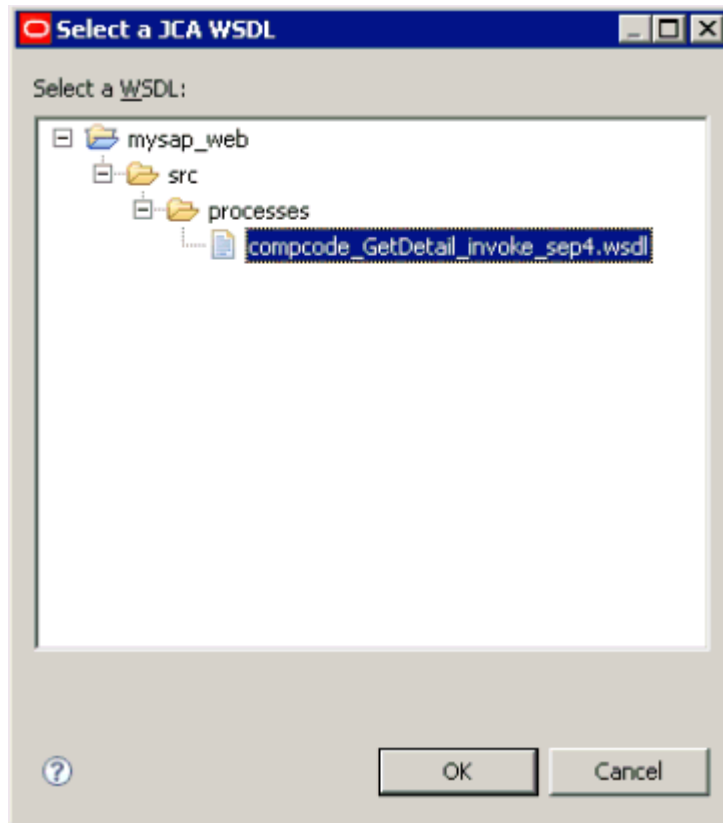
- 26. In the Name field, specify a name for the new JCA control (for example, mysap_ccgd) and click Next.

The Insert Control : JCA Control pane is displayed.



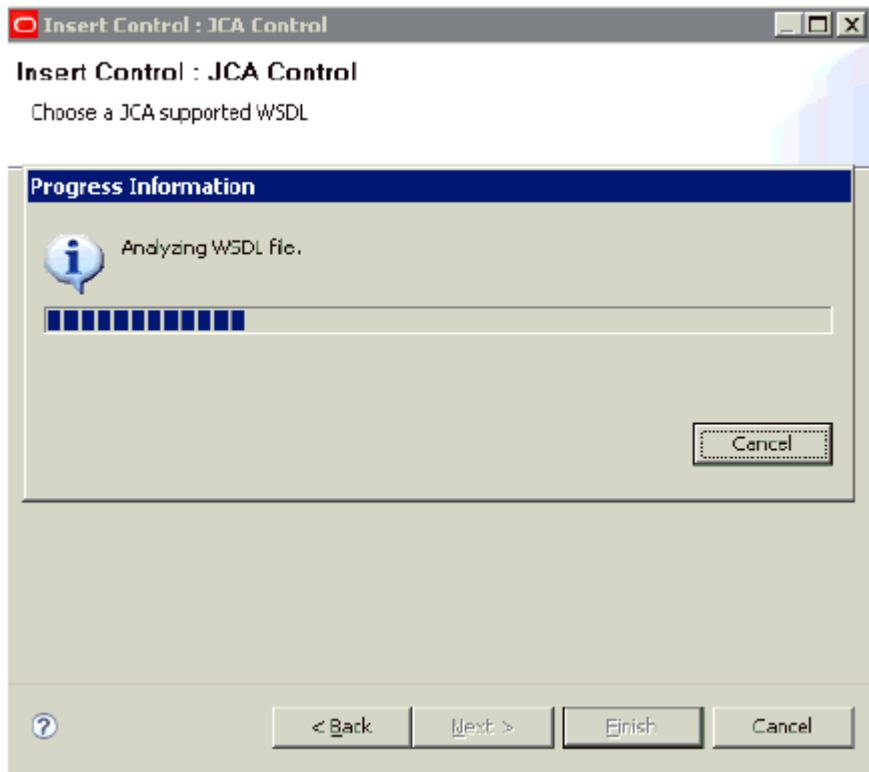
27. Click the Browse button to the right of the WSDL field.

The Select a JCA WSDL dialog box is displayed.

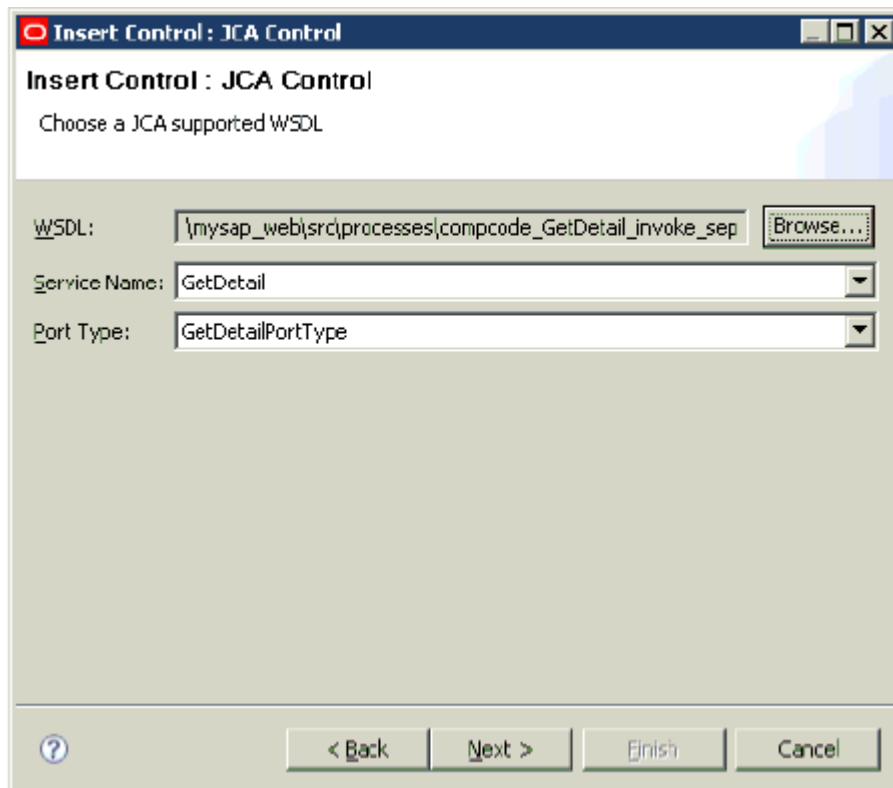


28. Expand mysap_web, followed by src and processes.
29. Select an available WSDL file (for example, CompanyCode_GetDetail_invoke_sep4.wsdl) and click OK.

The WSDL file is analyzed by a background process, as shown by the following status indicator.

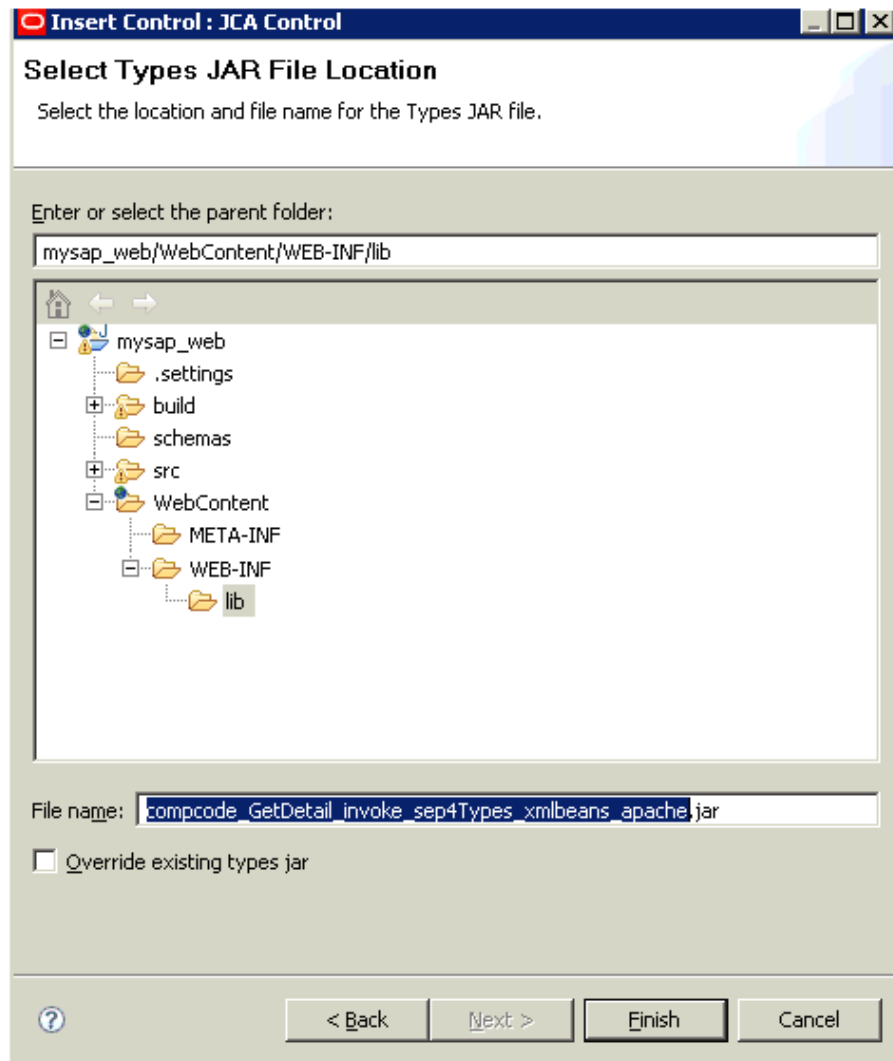


After this process is complete, you are returned to the Insert Control: JCA Control dialog box.



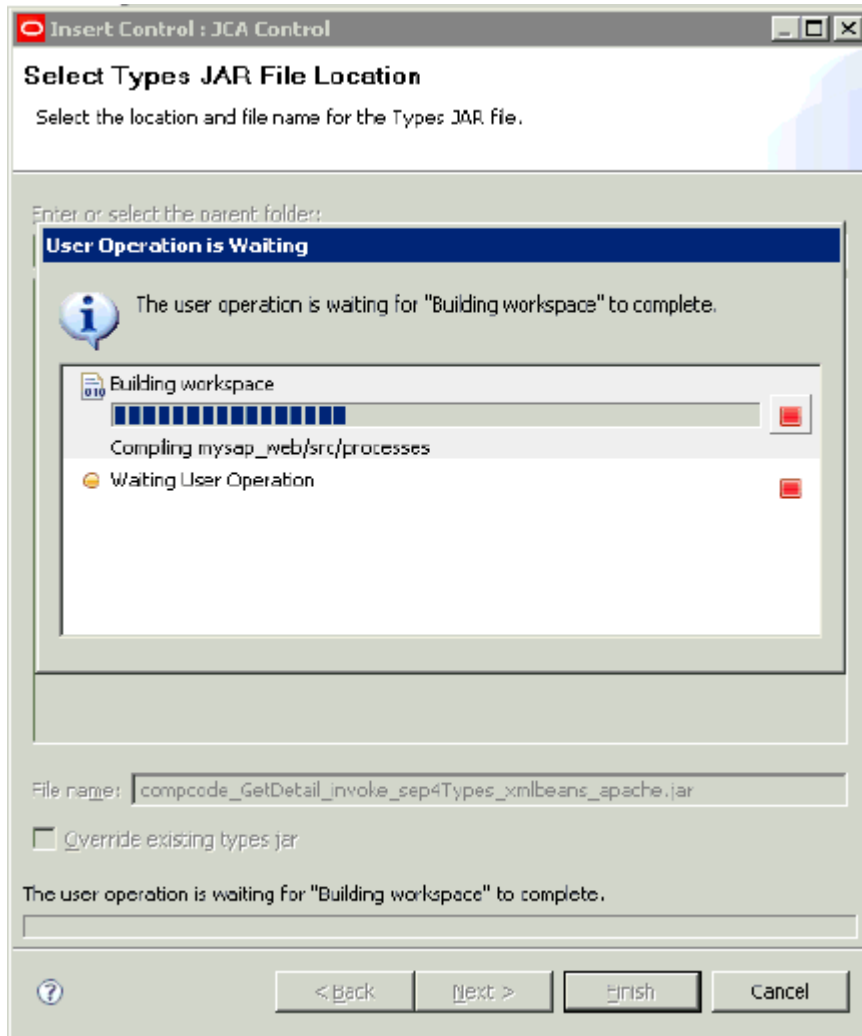
30. Click Next.

The Select Types JAR File Location pane is displayed.



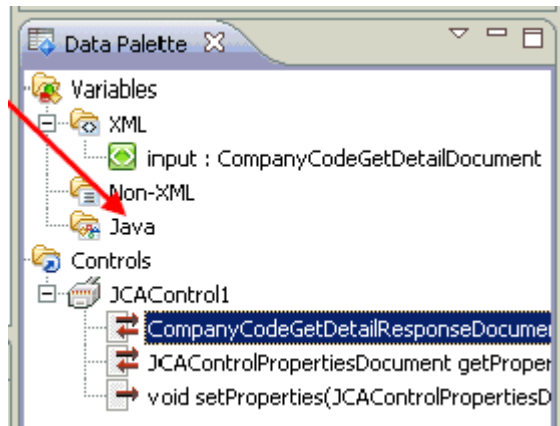
31. Click Finish.

The required operation is performed by a background process, as shown by the following status indicator.

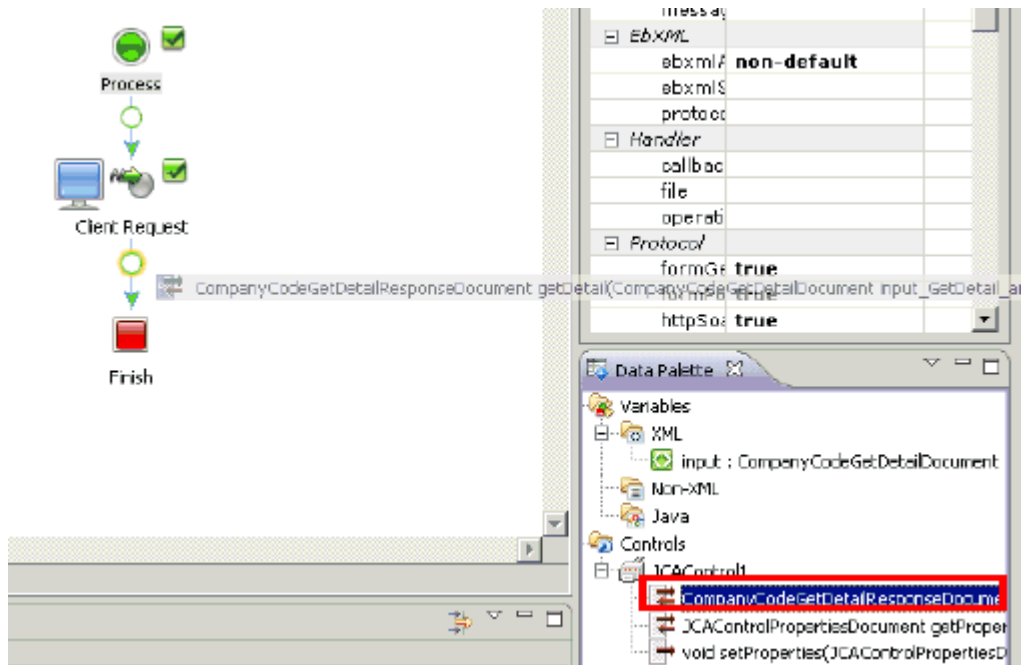


After this process is complete, the new JCA control (JCAControl1) is added in the Data Palette area under the Controls folder.

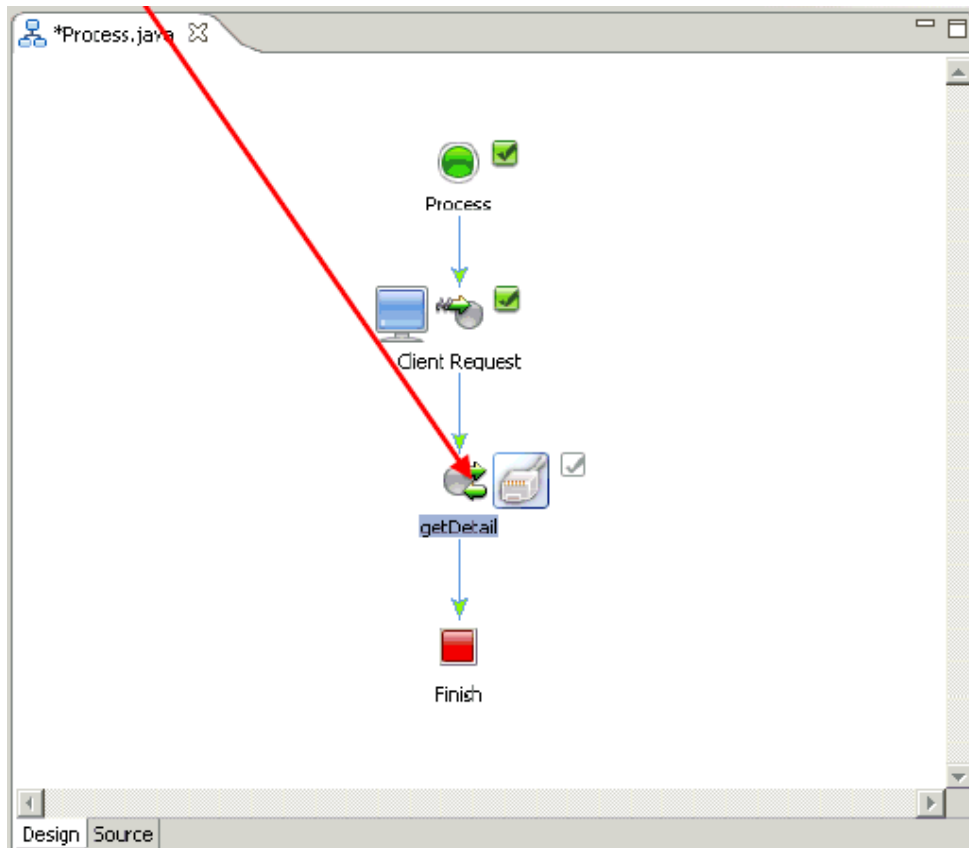
Note: When you use the JCA control with the Oracle Application Adapter for SAP R/3, you must use the JCAControl.Transaction annotation. This annotation is a class level or type level annotation that can be used in the JCA Control Interface source code. It has only one attribute suspend whose default value is "no". You must set the value of this attribute to "yes" to ensure that existing global transaction in the business process is suspended prior to invoking operations on the SAP Adapter via JCA Control and resumed after the call.



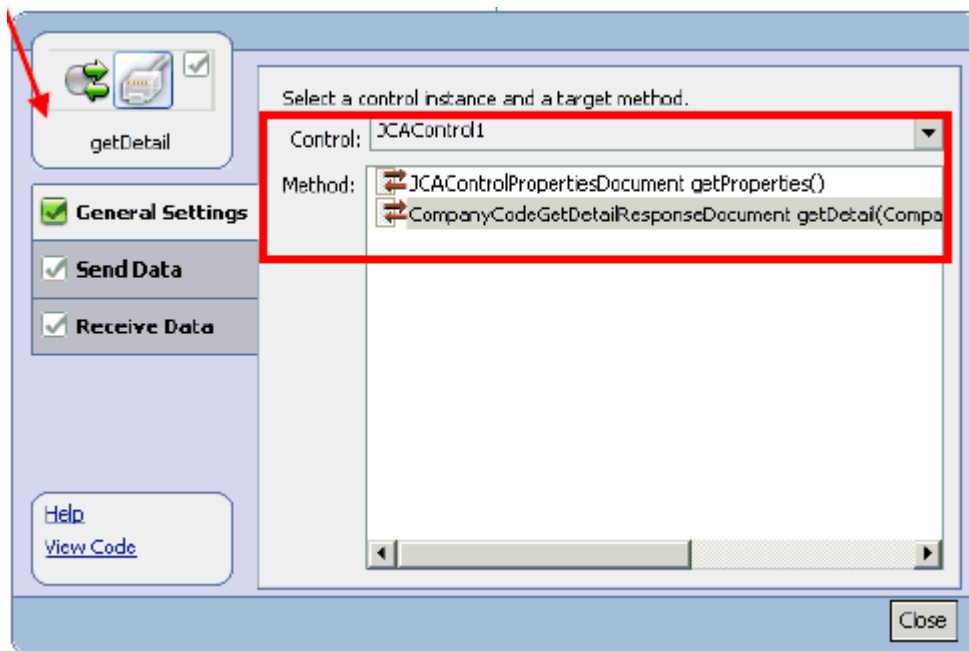
32. Click and drag `CompanyCode GetDetail` from the Data Palette area to the circle between the Client Request node and Finish node in the process design view (Process.java).



33. Right-click the `getDetail` node in the process design view (Process.java) and select `Open` from the context menu.

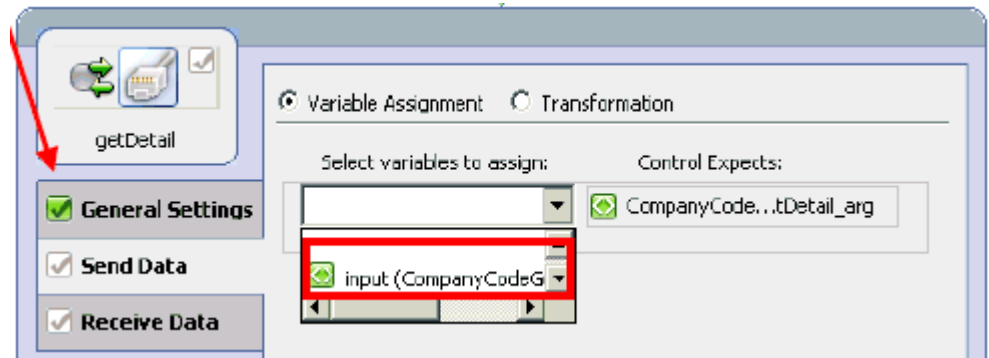


The getDetail configuration properties dialog box is displayed.



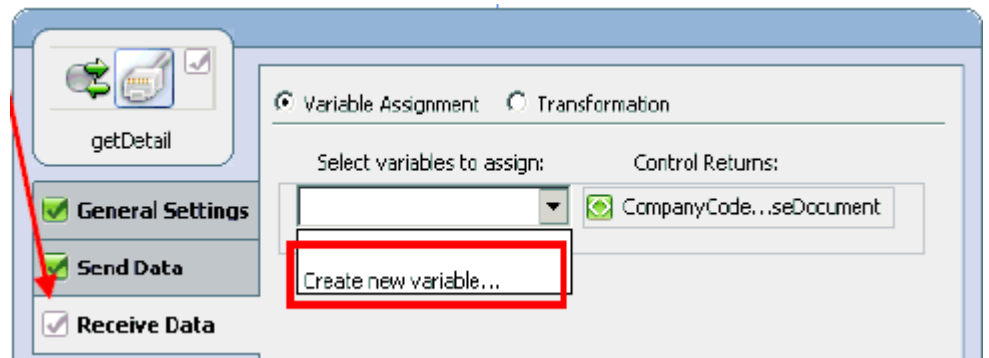
34. Click General Settings in the left pane.
35. Select JCAControl1 from the Control drop-down list.
36. Select CompanyCode GetDetail from the Method list.

37. Click Send Data in the left pane.

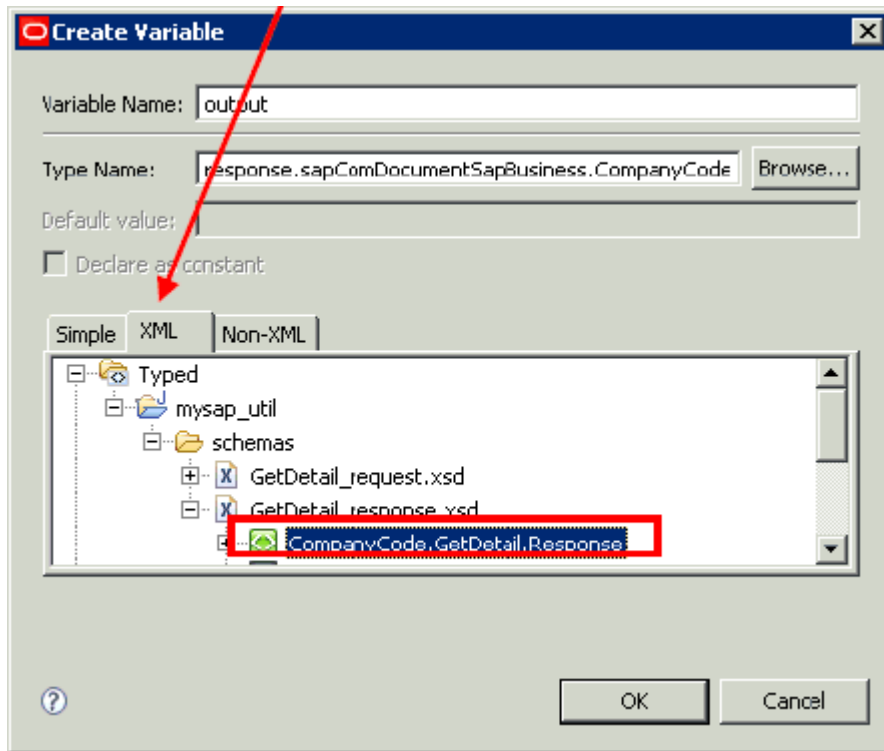


38. From the Select variables to assign drop-down list, select the defined input variable that is available.

39. Click Receive Data in the left pane.

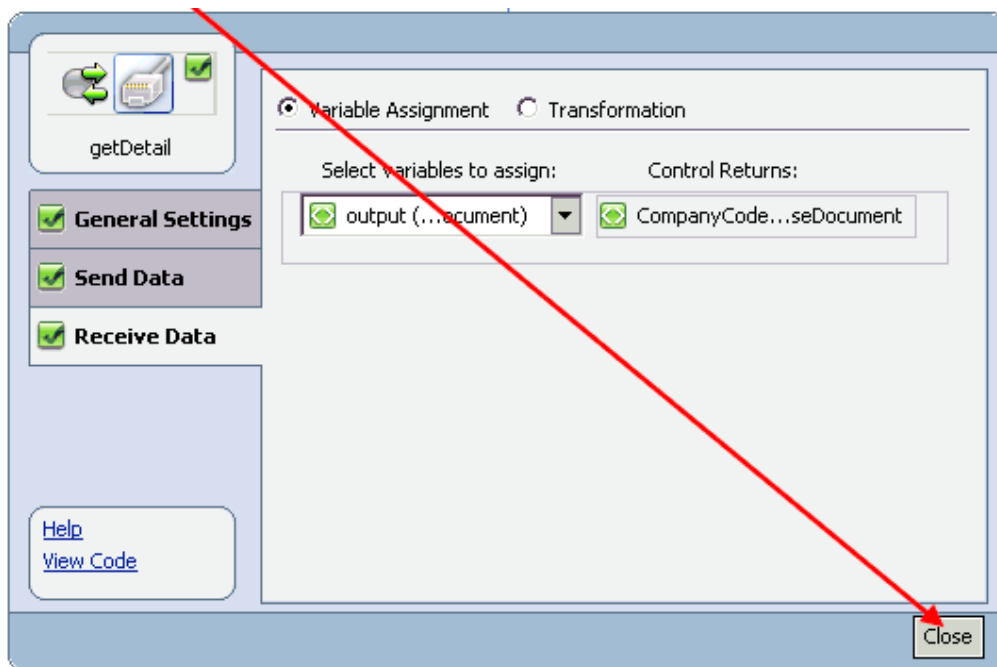


40. From the Select variables to assign drop-down list, select Create new variable. The Create Variable dialog box is displayed.



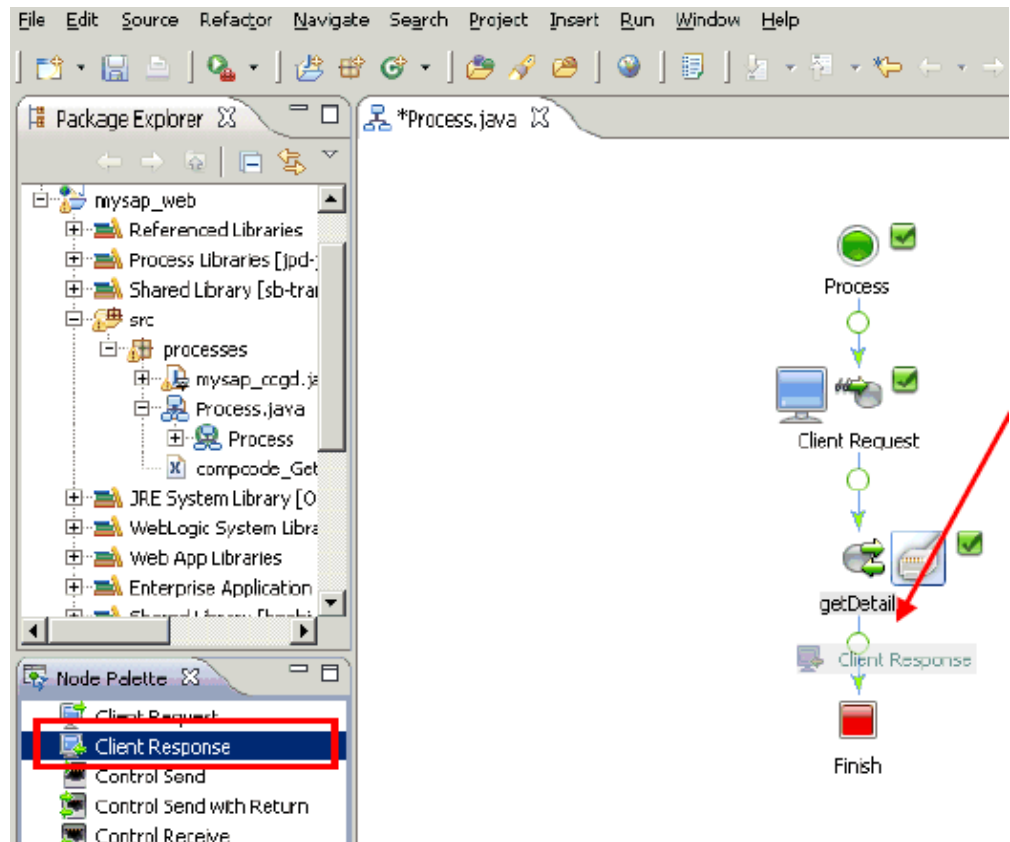
41. In the Variable Name field, enter a new variable name for the output data (for example, output).
42. In the XML tab, expand Typed, mysap_util, schemas, GetDetail_response.xsd, and select Companycode.GetDetail.Response.
43. Click OK.

You are returned to the getDetail configuration properties dialog box.



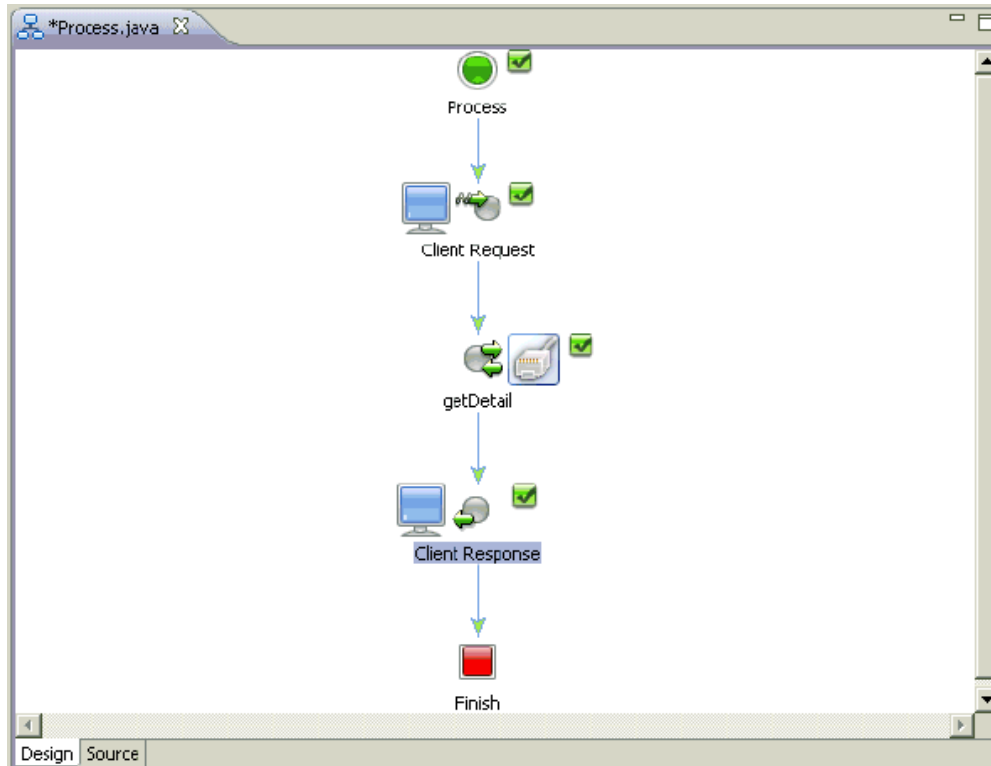
44. Click Close.

You are returned to the process design view (Process.java).

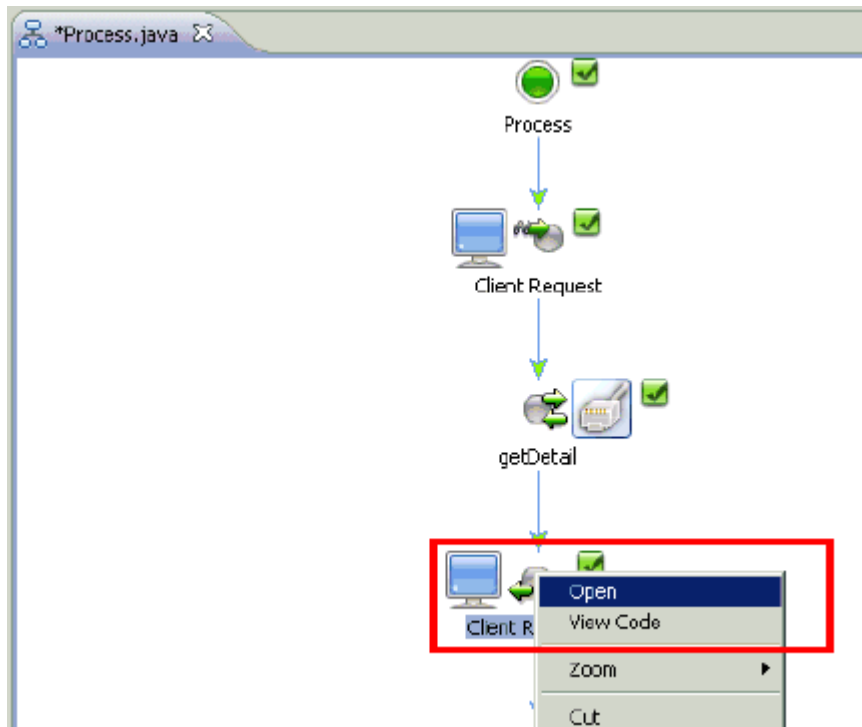


45. From the Node Palette area in the left pane, click the Client Response node and drag it to the circle between the getDetail node and the Finish node.

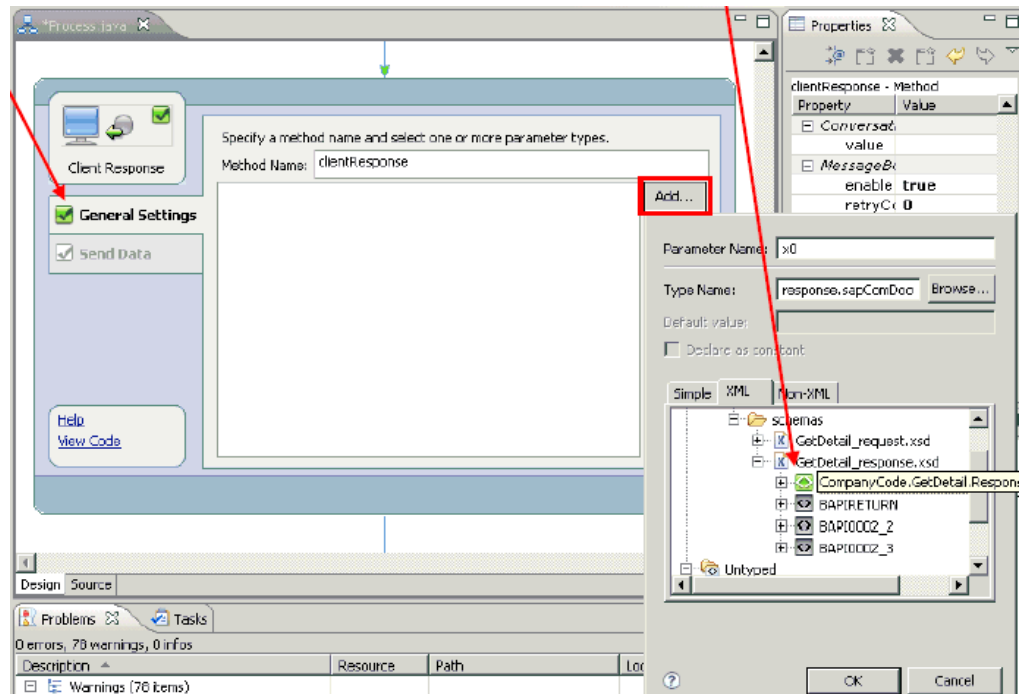
The process design view (Process.java) is updated to reflect the new addition of the Client Response node.



46. Right-click the Client Response node and select Open from the context menu.



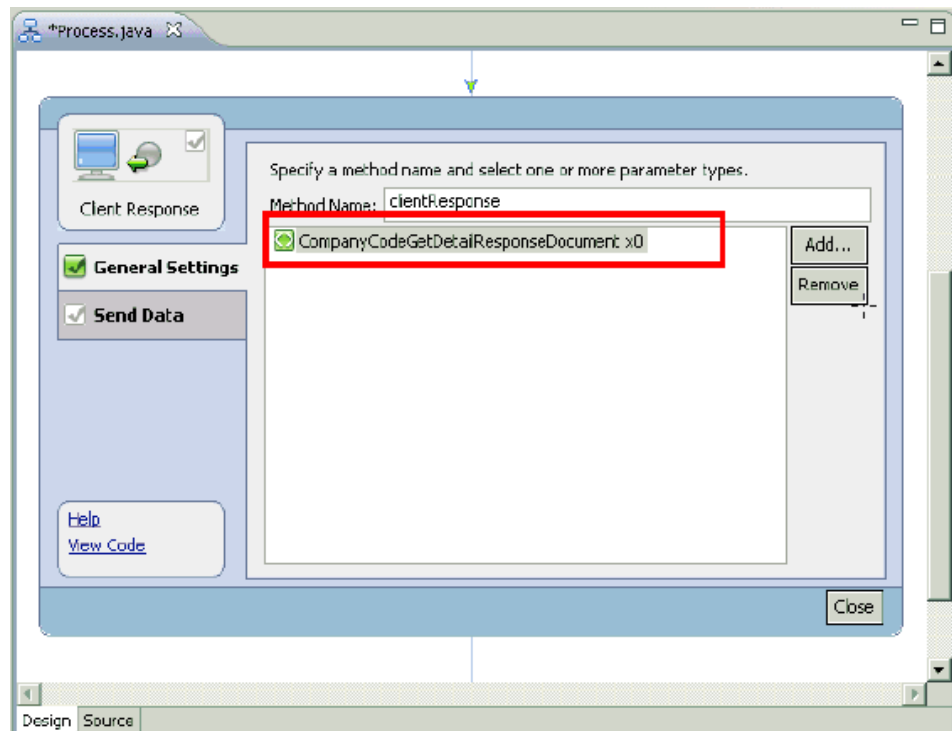
The Client Response configuration properties dialog box is displayed.



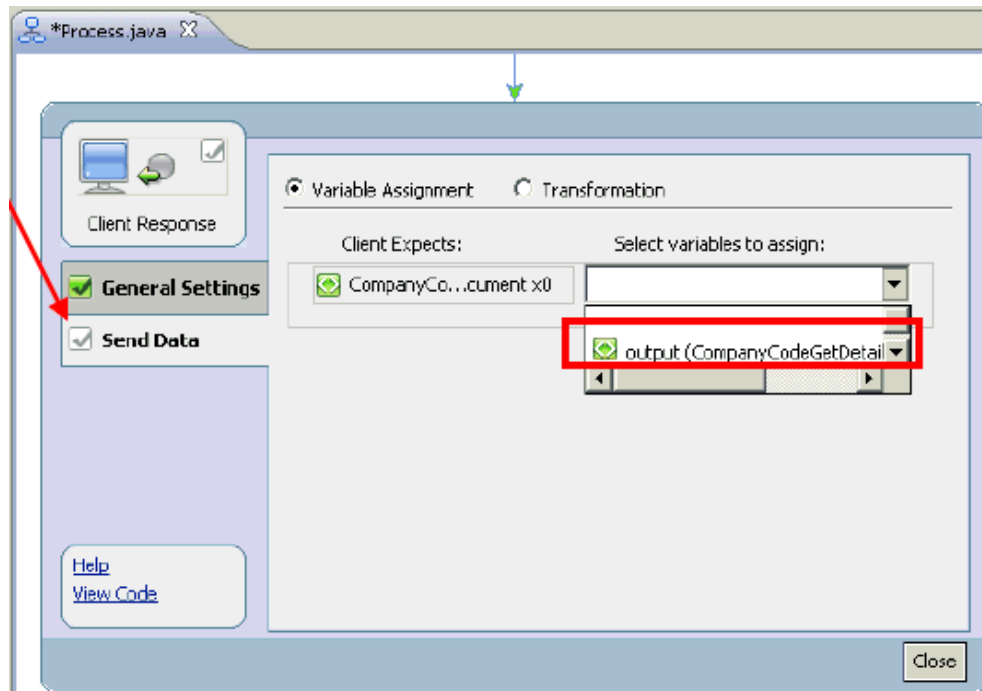
47. Click Add in the General Settings area.

48. In the XML tab, select the XML response schema (Companycode.GetDetail.Response) and click OK.

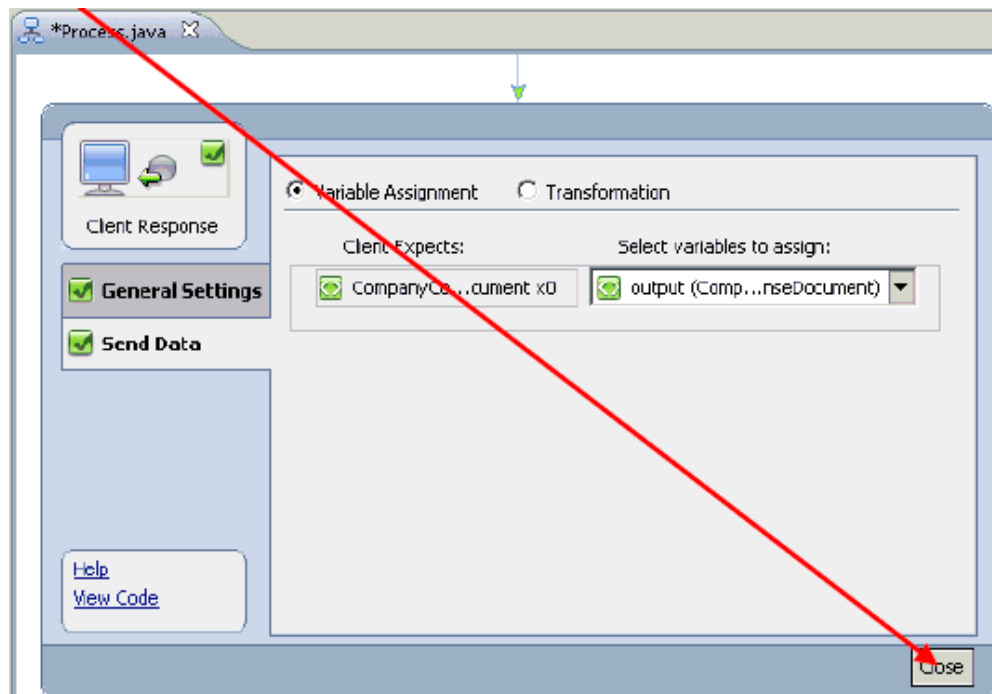
The response XML schema is added to the General Settings area.



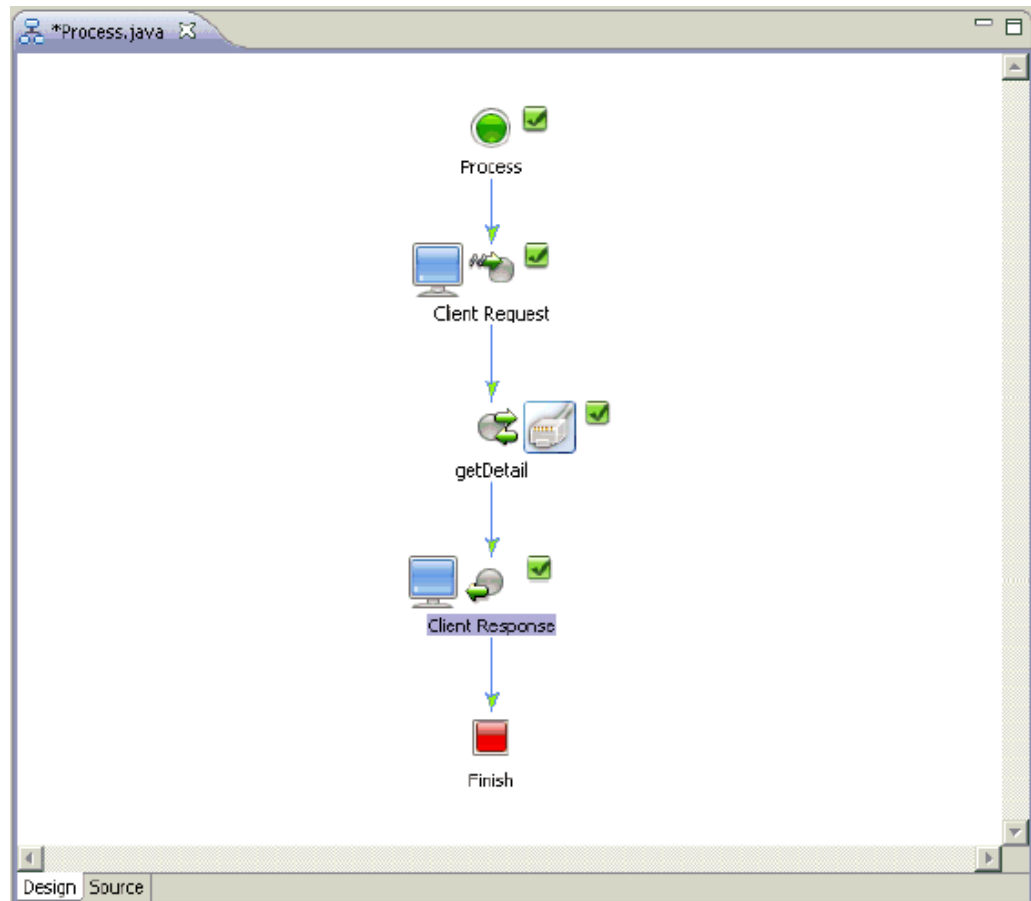
49. Click Send Data in the left pane.



50. From the Select variables to assign drop-down list, select the defined output variable that is available.



51. Click Close.
You are returned to the process design view (Process.java), which is now finished.



52. Save the outbound process (Process.java).

You are now ready to deploy the outbound process on WebLogic Server.

Deploying the Outbound Process on WebLogic Server

This section describes how to deploy the outbound process on WebLogic Server.

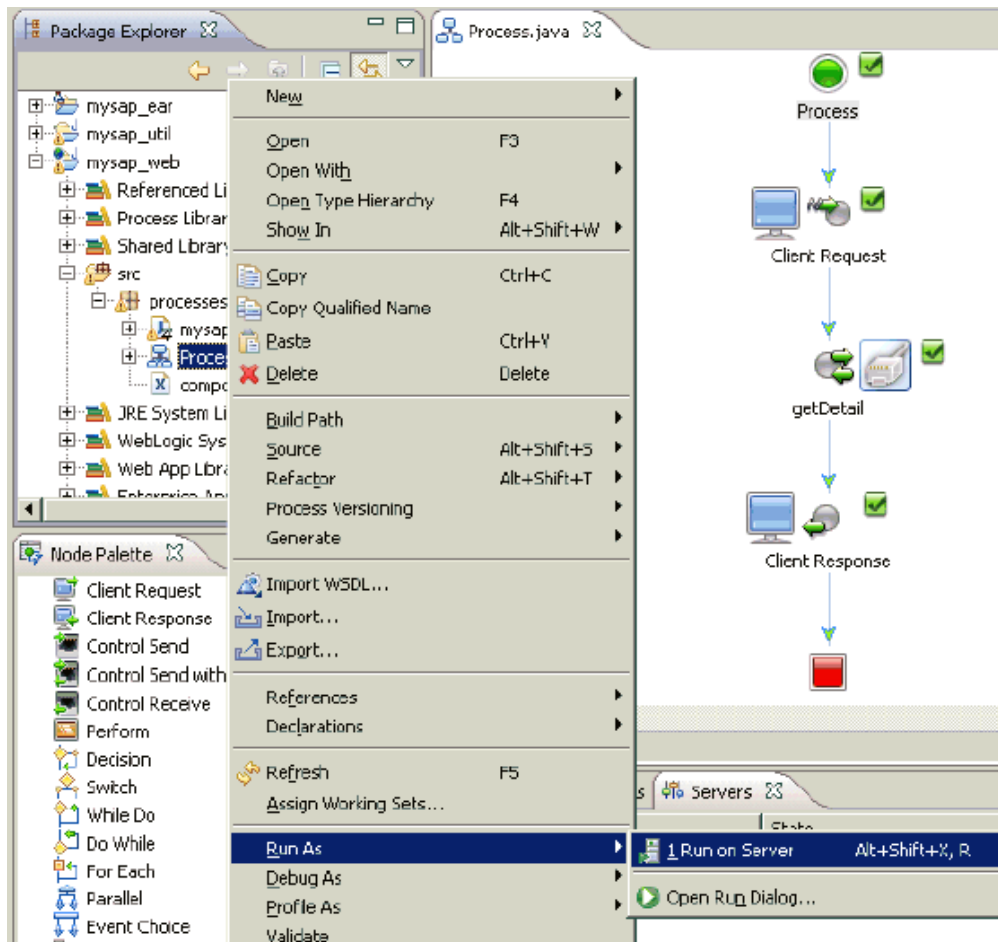
Note: Depending on your platform (for example, Windows or Solaris), there are a few differences in the outbound deployment process.

On Windows platforms, you can deploy the outbound process directly to the WebLogic Server. If you are using Windows, follow the procedure that is described in this section.

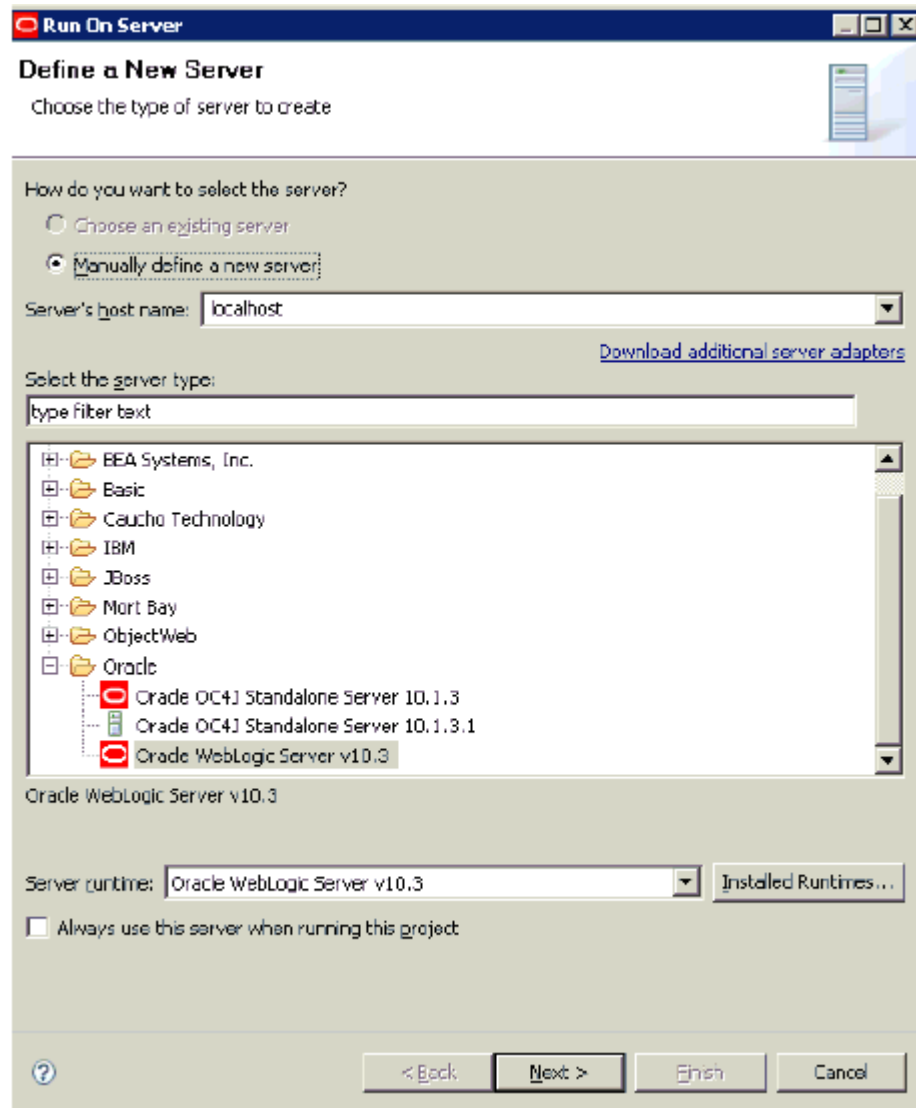
On Solaris platforms, you must first export the outbound process as an EAR file, deploy the outbound process on WebLogic Server, and then create a queue to receive the response document in the Test Console. For more information, see:

- [Exporting the Outbound JPD Process Application as an EAR File](#)
- [Deploying the Outbound JPD Process Application EAR File on WebLogic Server](#)
- [Creating a Queue to Receive the Response Document in the Test Console](#)

1. In the Package Explorer, right-click **Process**, select **Run As** from the context menu and click **Run on Server**.



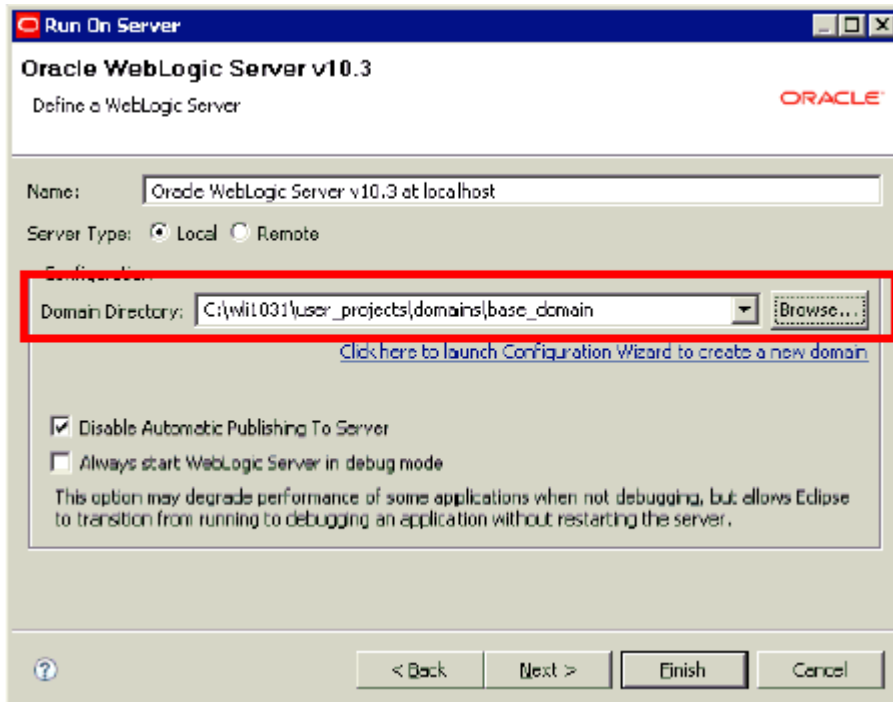
The Run On Server (Define a New Server) dialog box is displayed.



You can use an existing server or define a new server if required.

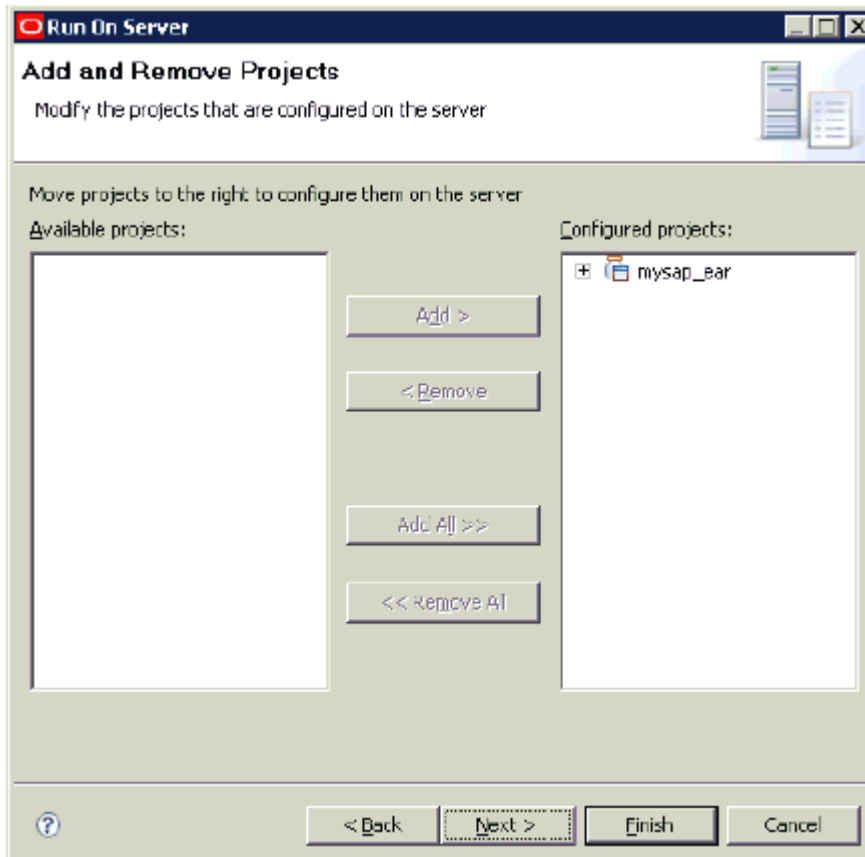
2. Verify that **Oracle WebLogic Server v10.3** is selected in the server type area and click **Next**.

The Run On Server (Oracle WebLogic Server v10.3) dialog box is displayed.



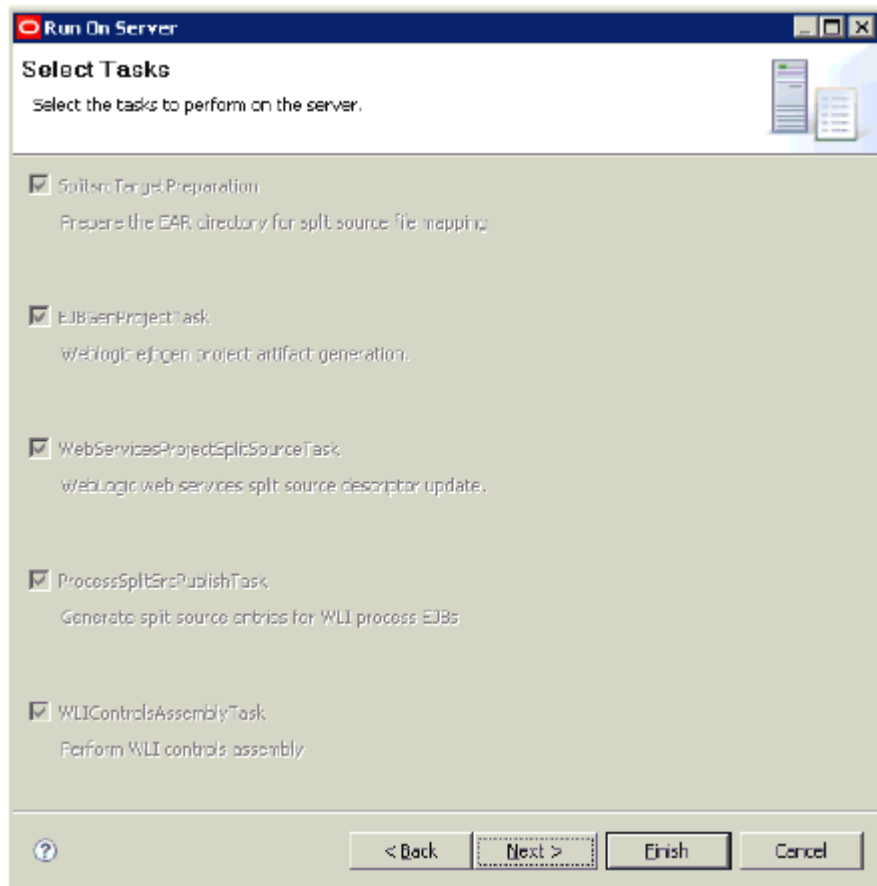
3. In the Domain Directory field, enter the path to your WebLogic Domain.
4. Click Next.

The Run On Server (Add and Remove Projects) dialog box is displayed.



5. Click Next.

The Run On Server (Select Tasks) dialog box is displayed.

**6. Click Finish.**

Once deployed, a Web browser is opened, as shown in the following image.

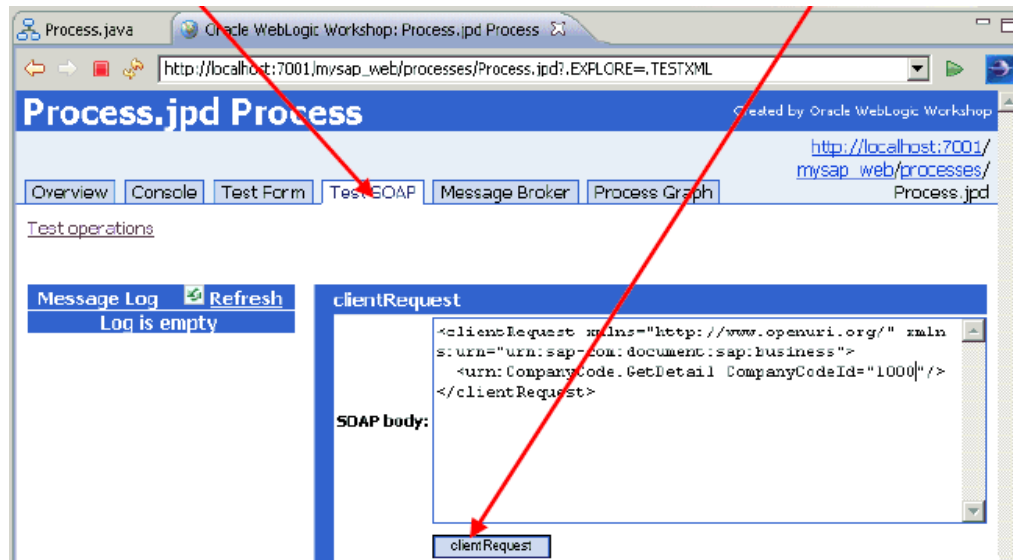


You are now ready to invoke the input XML using a Web browser.

Invoking the Input XML Using a Web Browser

This section describes how to invoke the input XML using a Web browser.

1. Click the **TestSOAP** tab.



2. In the clientRequest area, provide the input value and click **clientRequest**.
The request is processed, as shown in the following image.

The screenshot shows the Oracle WebLogic Workshop interface. The browser address bar displays the URL: `http://localhost:7001/mysap_web/processes/Process.jsp?EXPLORE=.TESTXML&.LOGENTRY=0`. The interface includes tabs for Overview, Console, Test Form, Test SOAP, Message Broker, and Process Graph. The Message Log is active, showing a message log for 'clientRequest' with a 'Refresh' and 'Clear Log' button. The log contains two entries:

Service Request clientRequest
Submitted at Tuesday, September 15, 2009 5:16:26 PM GMT+05:30

```
<?xml version="1.0" encoding="UTF-8" ?>
<SOAP-ENV:Envelope xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Header>
    <StartHeader xmlns="http://www.openuri.org/2002/04/soap/conversation/">
      <callbackLocation>http://TESTUI</callbackLocation>
    </StartHeader>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <clientRequest xmlns="http://www.openuri.org/" xmlns:urn="urn:sap-com:document:sap:business">
      <urn:CompanyCode.GetDetail CompanyCodeId="1000">
        </urn:CompanyCode.GetDetail>
      </clientRequest>
    </SOAP-ENV:Body>
  </SOAP-ENV:Envelope>
```

Service Response
Submitted at Tuesday, September 15, 2009 5:16:27 PM GMT+05:30

```
<?xml version="1.0" encoding="UTF-8" ?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Body>
    <ns:clientRequestResponse xmlns:ns="http://www.openuri.org/">
      </ns:clientRequestResponse>
    </SOAP-ENV:Body>
  </SOAP-ENV:Envelope>
```

3. In the server traces after processing the request the following lines confirm that the output XML is returned successfully:

```
Tue Sep 15 17:18:55 GMT+05:30 2009 <FINEST> Returning outbound XMLRecord for
output_GetDetail
Tue Sep 15 17:18:55 GMT+05:30 2009 <INFO> Sent message..
```

For example:

```

C:\WINDOWS\system32\cmd.exe - startWebLogic.cmd
ver state changed to ADMIN
<Sep 14, 2009 12:52:24 PM GMT+05:30> <Notice> <WebLogicServer> <BEA-000365> <Server state changed to RESUMING>
<Sep 14, 2009 12:52:24 PM GMT+05:30> <Warning> <Server> <BEA-002611> <Hostname "AMTEX-CH-QA163.AMTEXPDC", maps to multiple IP addresses: 192.168.128.163, 127.0.0.1>
<Sep 14, 2009 12:52:24 PM GMT+05:30> <Warning> <Server> <BEA-002611> <Hostname "127.0.0.1", maps to multiple IP addresses: 192.168.128.163, 127.0.0.1>
<Sep 14, 2009 12:52:24 PM GMT+05:30> <Notice> <Server> <BEA-002613> <Channel "Default" is now listening on 192.168.128.163:7001 for protocols iiop, t3, ldap, snmp, http.>
<Sep 14, 2009 12:52:24 PM GMT+05:30> <Notice> <Server> <BEA-002613> <Channel "Default[1]" is now listening on 127.0.0.1:7001 for protocols iiop, t3, ldap, snmp, http.>
<Sep 14, 2009 12:52:24 PM GMT+05:30> <Notice> <WebLogicServer> <BEA-000331> <Started WebLogic Admin Server "AdminServer" for domain "base_domain" running in Development Mode>
<Sep 14, 2009 12:52:27 PM GMT+05:30> <Notice> <WebLogicServer> <BEA-000365> <Server state changed to RUNNING>
<Sep 14, 2009 12:52:27 PM GMT+05:30> <Notice> <WebLogicServer> <BEA-000360> <Server started in RUNNING mode>
Tue Sep 15 17:16:34 GMT+05:30 2009 <FINEST> Instantiating XMLRecord converter: com.ibm.afjca.oracle.XMLRecordConverterImpl
Tue Sep 15 17:16:34 GMT+05:30 2009 <FINEST> Instantiating outbound JCA interactionSpec com.ibm.afjca.cci.IVAFInteractionSpec
Tue Sep 15 17:16:34 GMT+05:30 2009 <FINEST> Populating outbound JCA interactionSpec com.ibm.afjca.cci.IVAFInteractionSpec with properties: <FunctionName=PROCESS
>
Tue Sep 15 17:17:31 GMT+05:30 2009 <INFO> zip:C:\wli10gr3\user_projects\domains\base_domain\servers\AdminServer\tnp\WL_user\mysap_ear\n5sqjw\war\WEB-INF\lib\w_lcls_gen.jar!/processes/comcode_GetDetail_invoke_sep4.wsdl | GetDetailPortTypepe::GetDetail(input_GetDetail,output_GetDetail) | - Using JCA Connection Pool - max size = <unbounded>
Tue Sep 15 17:17:31 GMT+05:30 2009 <FINEST> zip:C:\wli10gr3\user_projects\domains\base_domain\servers\AdminServer\tnp\WL_user\mysap_ear\n5sqjw\war\WEB-INF\lib\w_lcls_gen.jar!/processes/comcode_GetDetail_invoke_sep4.wsdl | GetDetailPortTypepe::GetDetail(input_GetDetail,output_GetDetail) | - Looking up Resource Adapter JNDI location 'eis/OracleJCAAdapter/DefaultConnection'
Tue Sep 15 17:18:47 GMT+05:30 2009 <FINEST> zip:C:\wli10gr3\user_projects\domains\base_domain\servers\AdminServer\tnp\WL_user\mysap_ear\n5sqjw\war\WEB-INF\lib\w_lcls_gen.jar!/processes/comcode_GetDetail_invoke_sep4.wsdl | GetDetailPortTypepe::GetDetail(input_GetDetail,output_GetDetail) | - Starting JCA LocalTransaction
Tue Sep 15 17:18:47 GMT+05:30 2009 <FINEST> zip:C:\wli10gr3\user_projects\domains\base_domain\servers\AdminServer\tnp\WL_user\mysap_ear\n5sqjw\war\WEB-INF\lib\w_lcls_gen.jar!/processes/comcode_GetDetail_invoke_sep4.wsdl | GetDetailPortTypepe::GetDetail(input_GetDetail,output_GetDetail) | - Invoking JCA Outbound Interaction
Tue Sep 15 17:18:55 GMT+05:30 2009 <FINEST> zip:C:\wli10gr3\user_projects\domains\base_domain\servers\AdminServer\tnp\WL_user\mysap_ear\n5sqjw\war\WEB-INF\lib\w_lcls_gen.jar!/processes/comcode_GetDetail_invoke_sep4.wsdl | GetDetailPortTypepe::GetDetail(input_GetDetail,output_GetDetail) | - Committing JCA LocalTransaction
Tue Sep 15 17:18:55 GMT+05:30 2009 <FINEST> Returning outbound XMLRecord for output_GetDetail
Tue Sep 15 17:18:55 GMT+05:30 2009 <INFO> Sent message..
    
```

4. Return to the Web browser and click Refresh.

The screenshot shows the Oracle WebLogic Workshop interface. The 'Test SOAP' tab is active, displaying a SOAP response in the 'Client Callback' window. The response is an XML document with the following structure:

```

<SOAP-ENV:Envelope xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Body>
    <ns:clientResponse xmlns:ns="http://www.openurl.org/">
      <CompanyCode_GetDetail.Response xmlns="urn:sap-com:document:sap.business.response">
        <COMPANYCODE_ADDRESS>
          <ADDR_NO>000000121</ADDR_NO>
          <FORMOFADDR>Firma</FORMOFADDR>
          <NAME>Ides AG</NAME>
          <NAME_2></NAME_2>
          <NAME_3></NAME_3>
          <NAME_4></NAME_4>
          <C_O_NAME></C_O_NAME>
          <CITY>Frankfurt</CITY>
          <DISTRICT></DISTRICT>
          <CITY_NO></CITY_NO>
          <POSTL_CDD1>60441</POSTL_CDD1>
          <POSTL_CDD2>60070</POSTL_CDD2>
          <POSTL_CDD3></POSTL_CDD3>
          <PO_BOX>160529</PO_BOX>
          <PO_BOX_CIT></PO_BOX_CIT>
          <DELIW_DIS></DELIW_DIS>
          <STREET>Lyoner Stern 231</STREET>
          <STREET_NO></STREET_NO>
          <STR_ADDR></STR_ADDR>
          <HOUSE_NO></HOUSE_NO>
          <STR_SUPPL1></STR_SUPPL1>
          <STR_SUPPL2></STR_SUPPL2>
          <LOCATION></LOCATION>
          <BUILDING></BUILDING>
          <FLOOR></FLOOR>
        
```

5. To view the output response, click **callback.clientResponse**.

Configuring Inbound Processing Using Oracle WebLogic Integration (J2CA Configuration)

To integrate with Oracle WebLogic Integration (WLI), Oracle Application Adapter for SAP R/3 must be deployed on the same Oracle WebLogic Server as WLI. The underlying adapter services must be exposed as WSDL files, which are generated during design time in Oracle WebLogic Server Adapter Application Explorer (Application Explorer) for event notification (inbound) services of the adapter. Oracle Application Adapter for SAP R/3 integrates seamlessly with WLI to facilitate Web service integration.

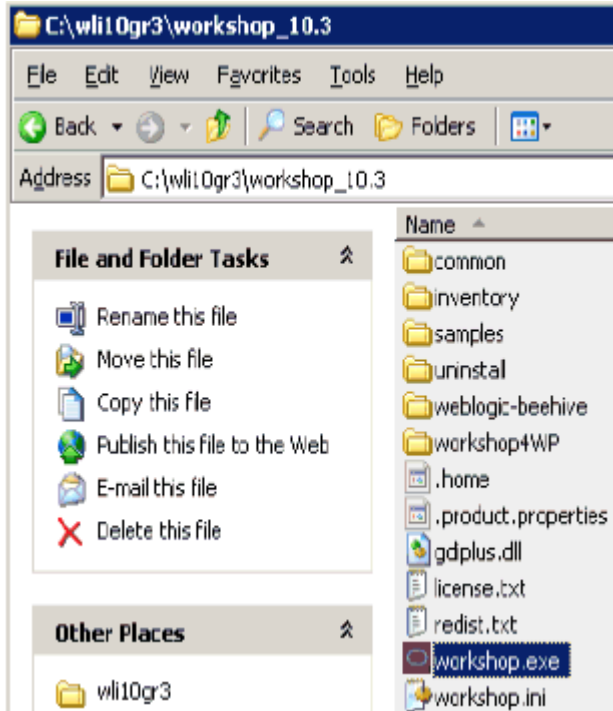
This chapter includes the following topics:

- [Creating an Inbound JPD Process Application](#)
- [Deploying the Inbound Process on WebLogic Server](#)
- [Creating an Event Generator and a Channel Subscription](#)
- [Triggering an Event From SAP GUI](#)
- [Verifying the Event](#)

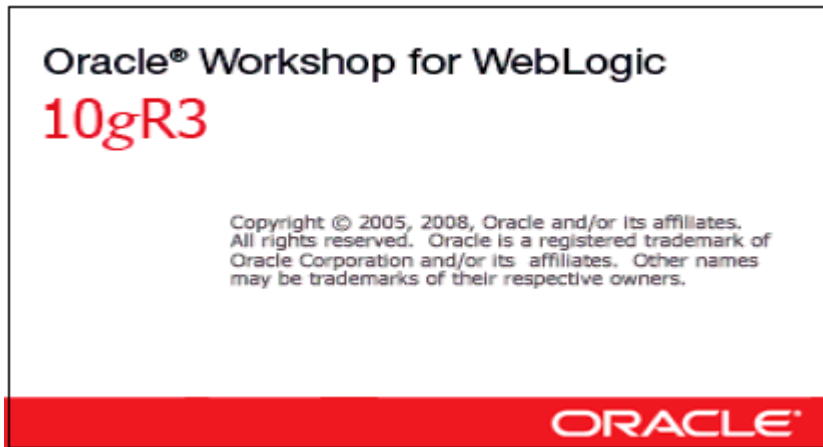
Creating an Inbound JPD Process Application

This section describes how to create an inbound JPD process application.

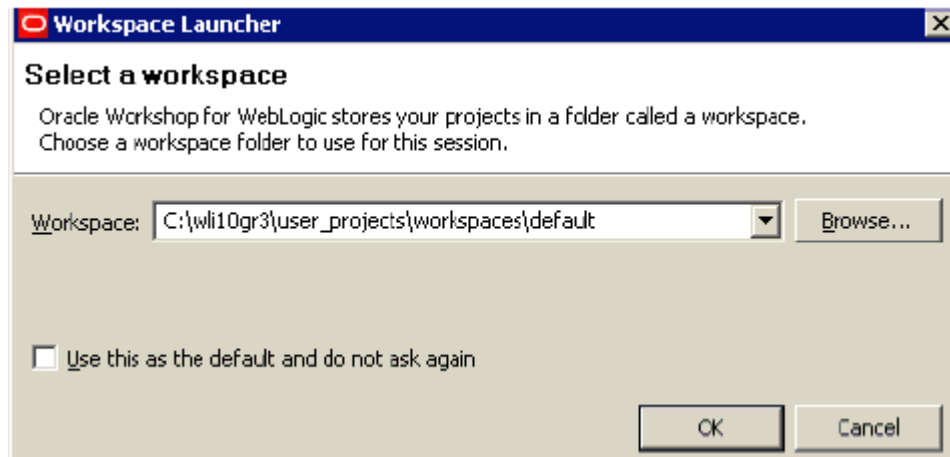
1. Navigate to the WebLogic Integration Workshop location (for example, `c:\wli10gr3\workshop_10.3`) and double-click the **workshop.exe** file.



The Oracle Workshop for WebLogic screen is displayed.



This is followed by the Workspace Launcher dialog box, as shown in the following image.



2. Specify the workspace location or accept the default path, which is:

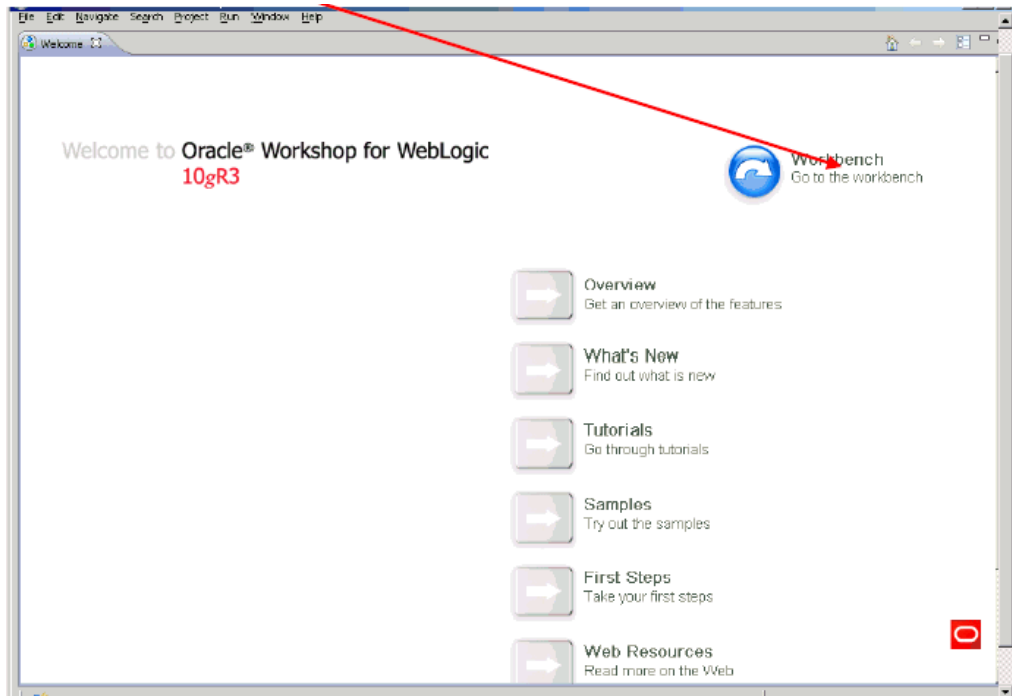
C:\wli10gr3\user_projects\workspaces\default

3. Click **OK**.

The Oracle Workshop for WebLogic loading screen is displayed.

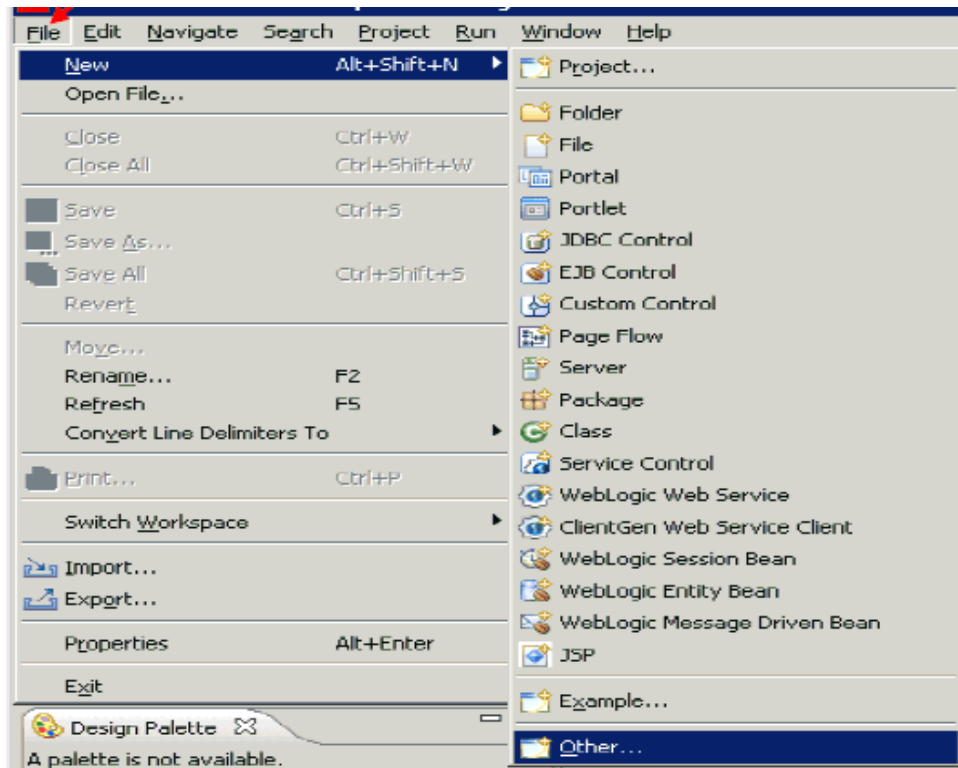


This is followed by the Oracle Workshop for WebLogic Welcome screen, as shown in the following image.



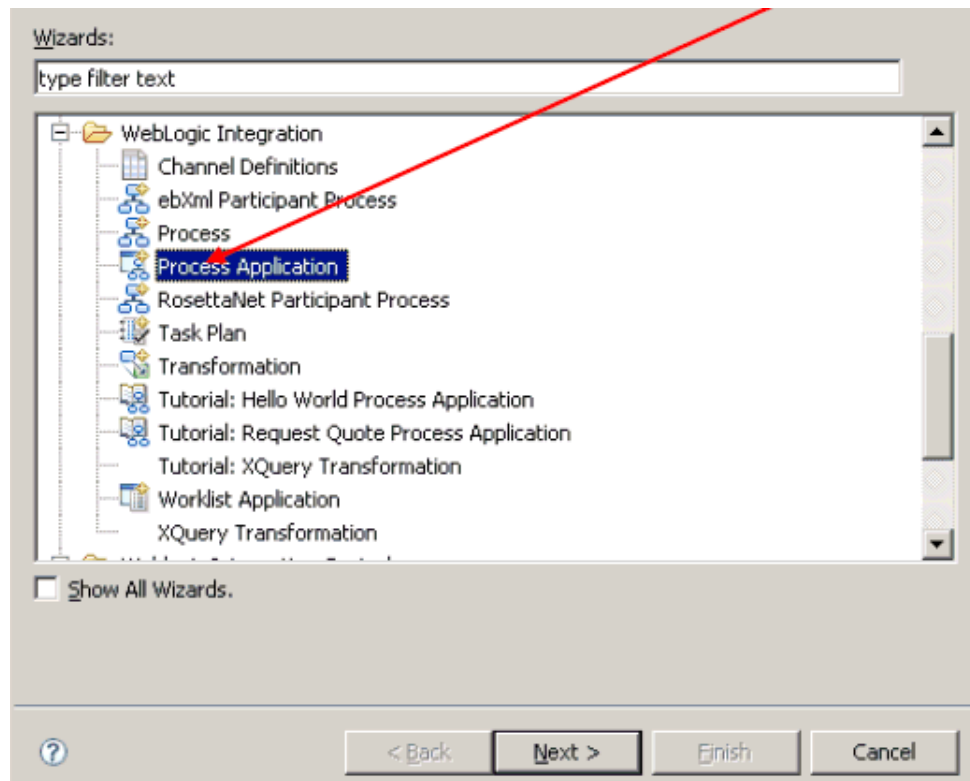
4. Click **Workbench**.

The Workbench area is displayed.



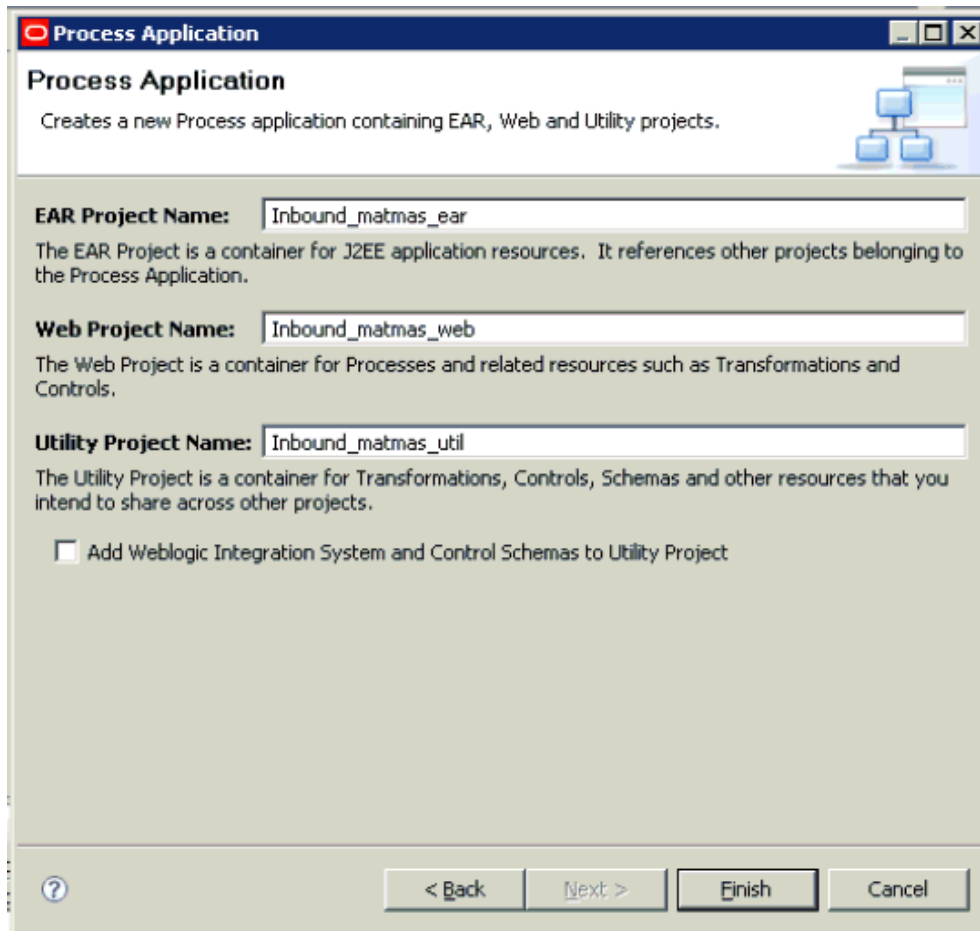
5. From the File menu, select **New** and click **Other**.

The following dialog box is displayed.



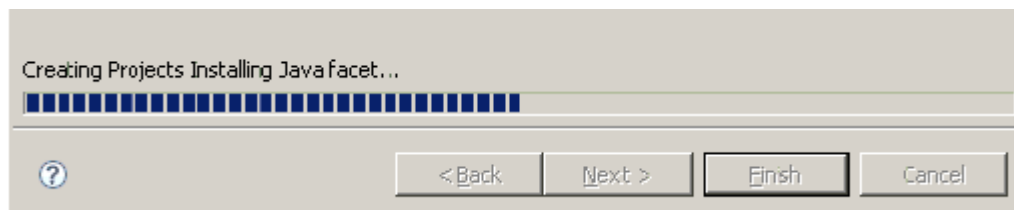
6. Expand the **WebLogic Integration** folder, select **Process Application**, and click **Next**.

The Process Application dialog box is displayed.

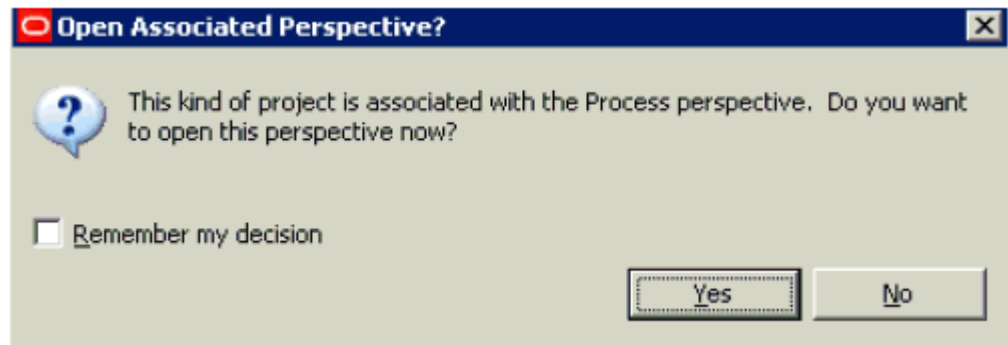


7. Perform the following steps:
 - a. Specify an EAR project name, for example, **Inbound_matmas_ear**.
 - b. Specify a Web project name, for example, **Inbound_matmas_web**.
 - c. Specify a Utility project name, for example, **Inbound_matmas_util**.
8. Click **Finish**.

The EAR, Web, and Utility projects are created by a background process, as shown by the following status indicator.



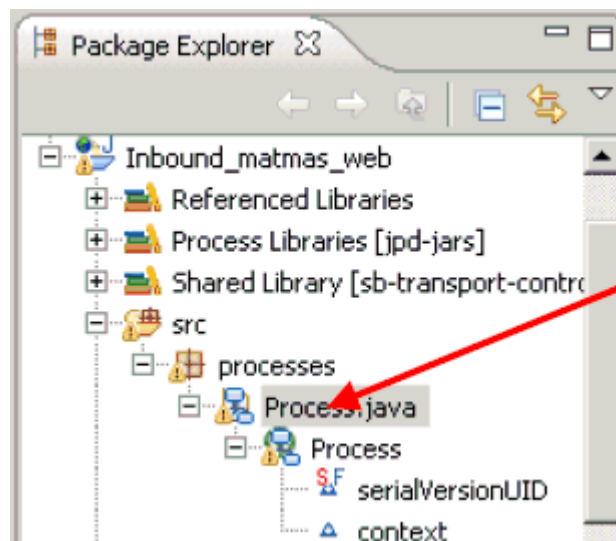
After this process is complete, the following message is displayed.



9. Click Yes to open the associated perspective.

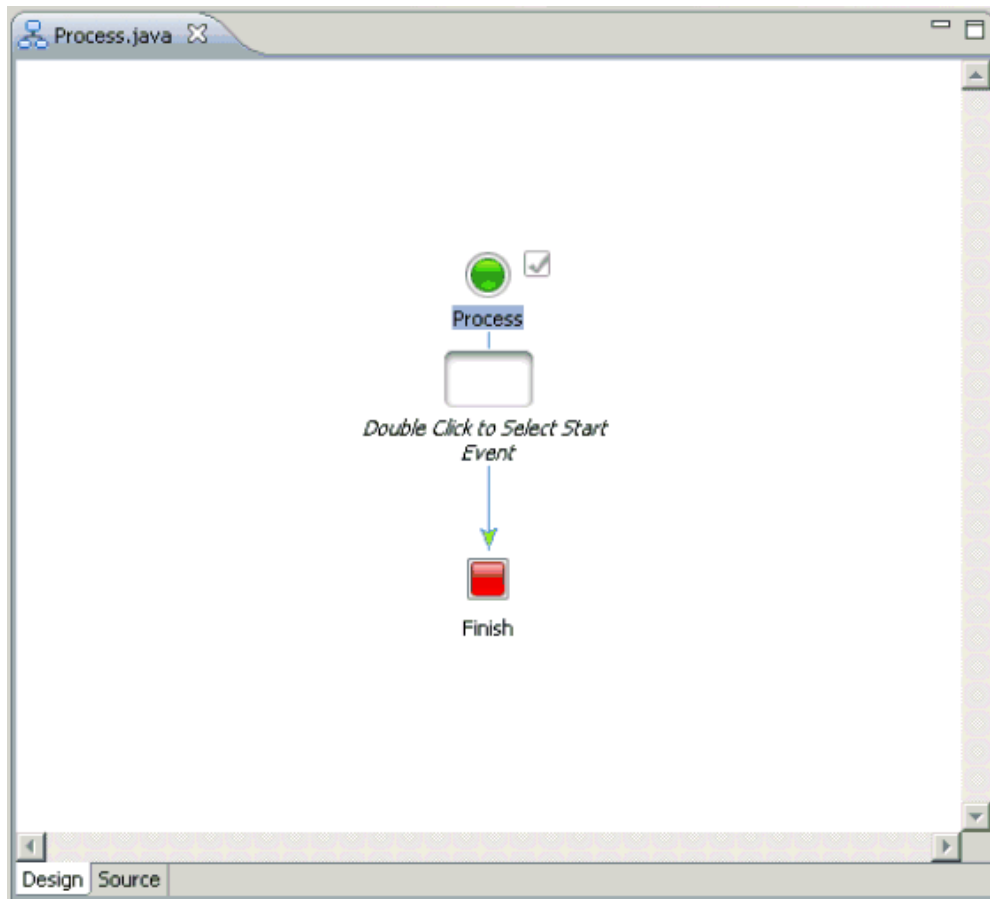
The EAR, Web, and Utility projects are created in the Package Explorer area in the left pane.

10. Expand the Web project node (for example, **Inbound_matmas_web**), followed by **src** and **processes**.

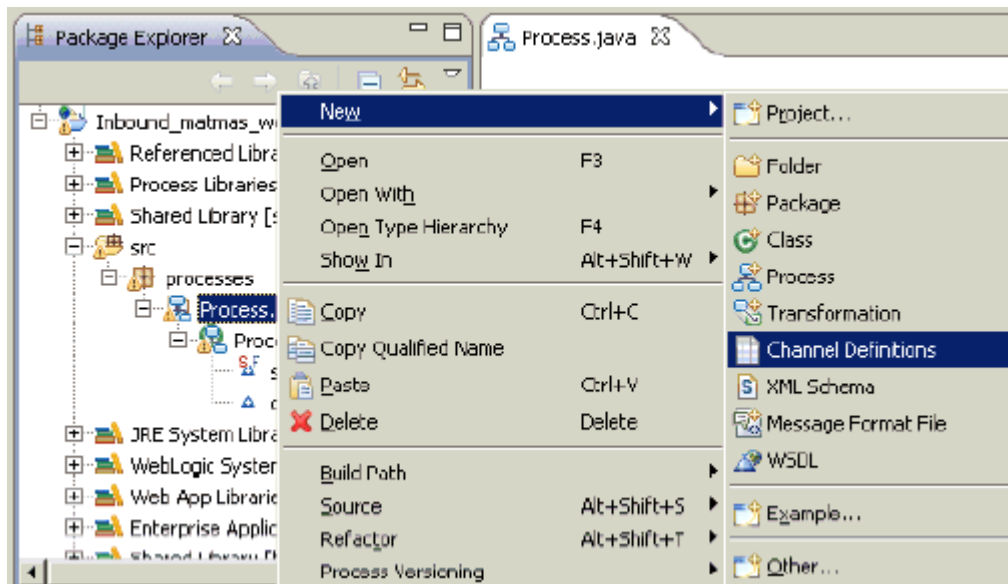


11. Double-click Process.java.

The process design view (Process.java) is displayed in the central pane.

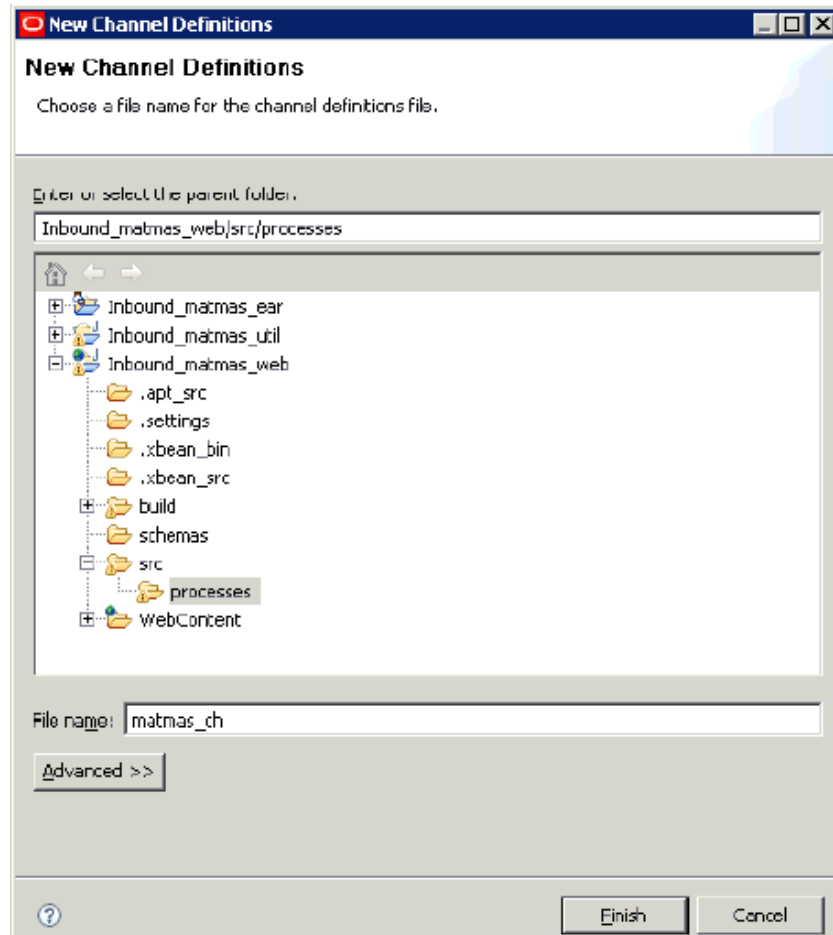


You are now ready to create a Message Broker channel with XML data.



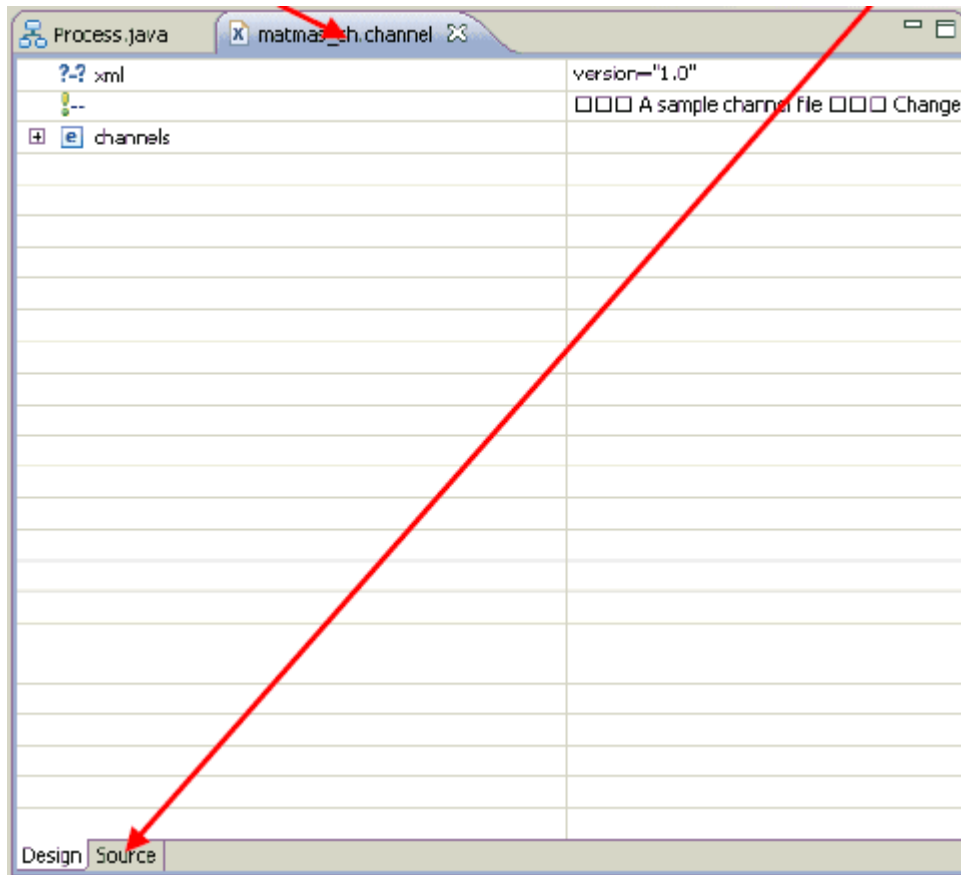
12. In the left pane, right-click **Process.java**, select **New**, and click **Channel Definitions** from the context menu.

The **New Channel Definitions** dialog box is displayed.

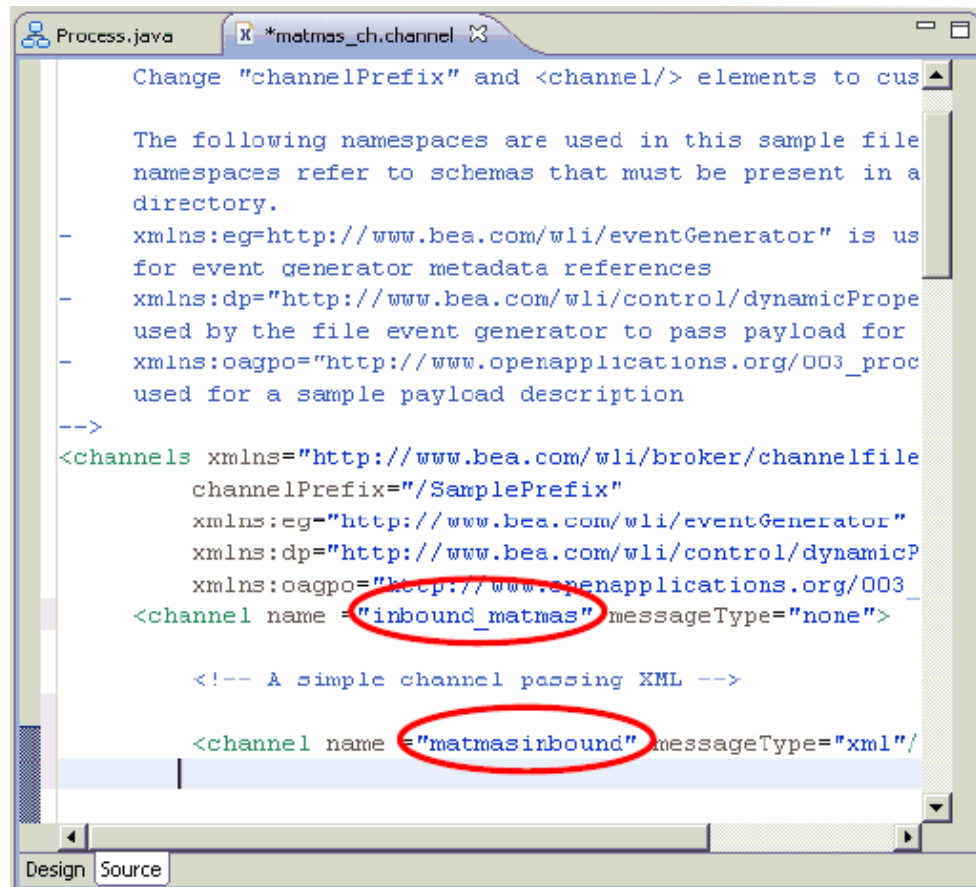


13. Specify a name for the new channel definition in the File name field, for example, matmas_ch.
14. Click Finish.

A new channel definition is created in the workspace area as a new tab (for example, matmas_ch.channel).



15. Click the Source tab to change the channel definition view.



16. Make the following changes in the source view so that after the application is deployed, the Message Broker channel is available for subscription to any event generator:

```
<channels xmlns="http://www.bea.com/wli/broker/channelfile"
  channelPrefix="/SamplePrefix" xmlns:eg="http://www.bea.com/wli/eventGenerator"
  xmlns:dp="http://www.bea.com/wli/control/dynamicProperties"
  xmlns:oagpo="http://www.openapplications.org/003_process_po_007">
  <channel name ="inbound_matmas" messageType="none">

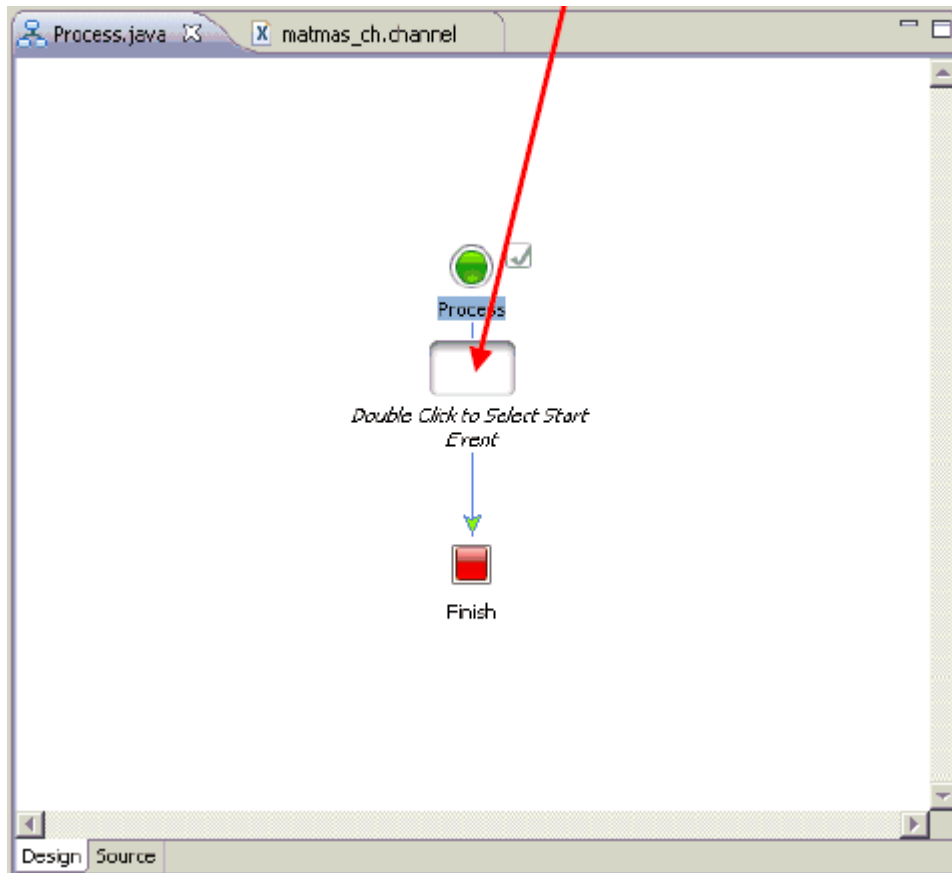
    <!-- A simple channel passing XML -->

    <channel name ="matmasinbound" messageType="xml"/>
```

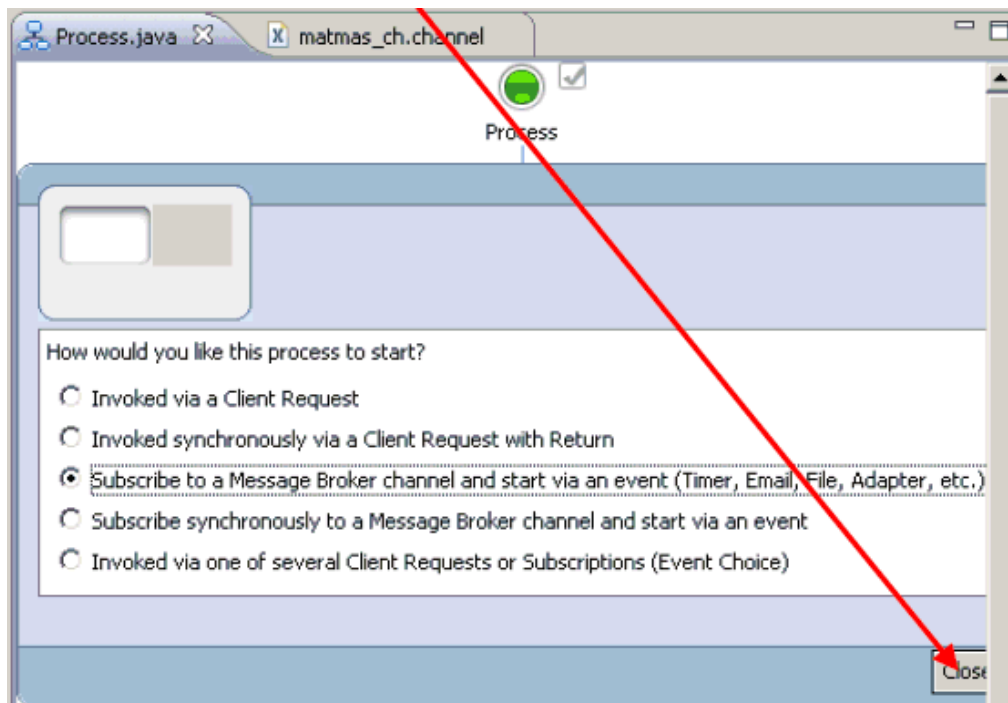
17. Click **Save**.

Please note that it is important to save your project at this stage, otherwise the channel will not be displayed for the subscription node.

18. Click the Process.java tab in the workspace area.

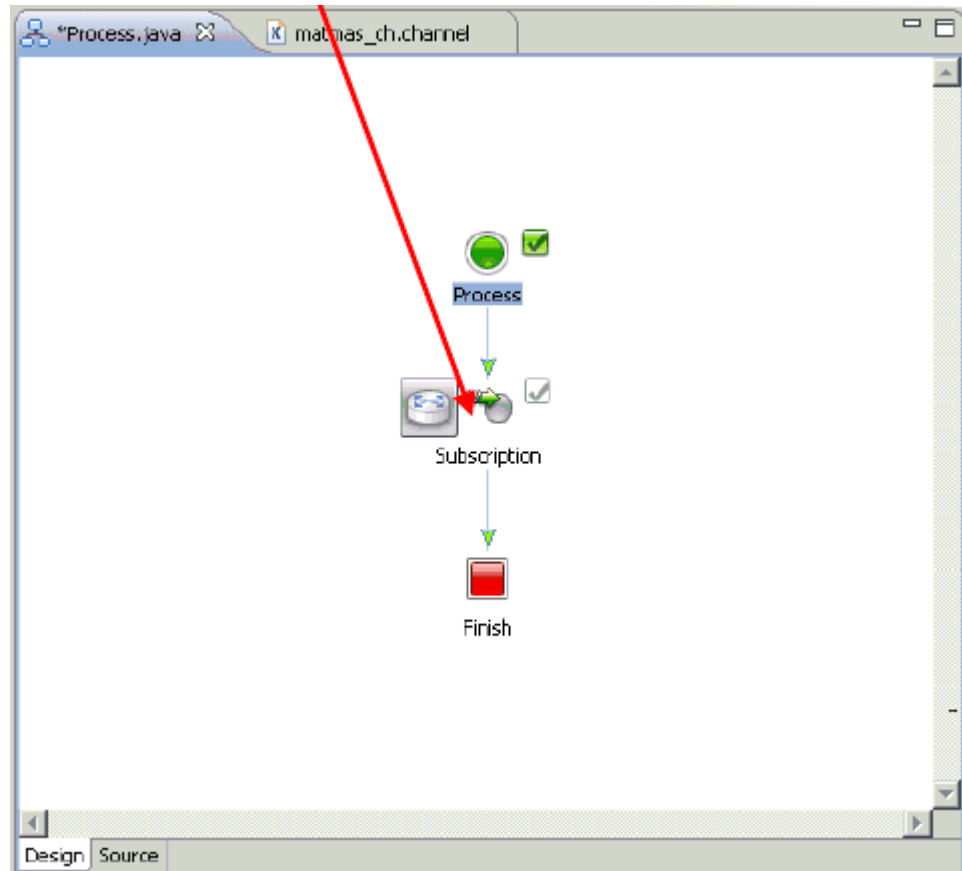


19. Double-click the main node in the process design view (Process.java).
The following dialog box is displayed.



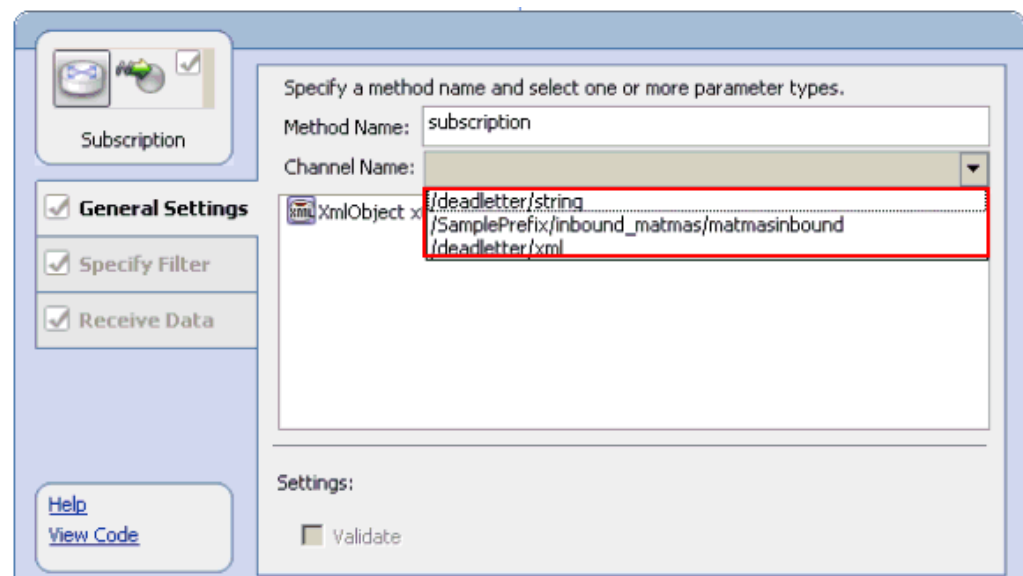
20. Select Subscribe to a Message Broker channel and start via an event (Timer, Email, File, Adapter, etc.) and click Close.

The Subscription node is added to the process design view (Process.java).

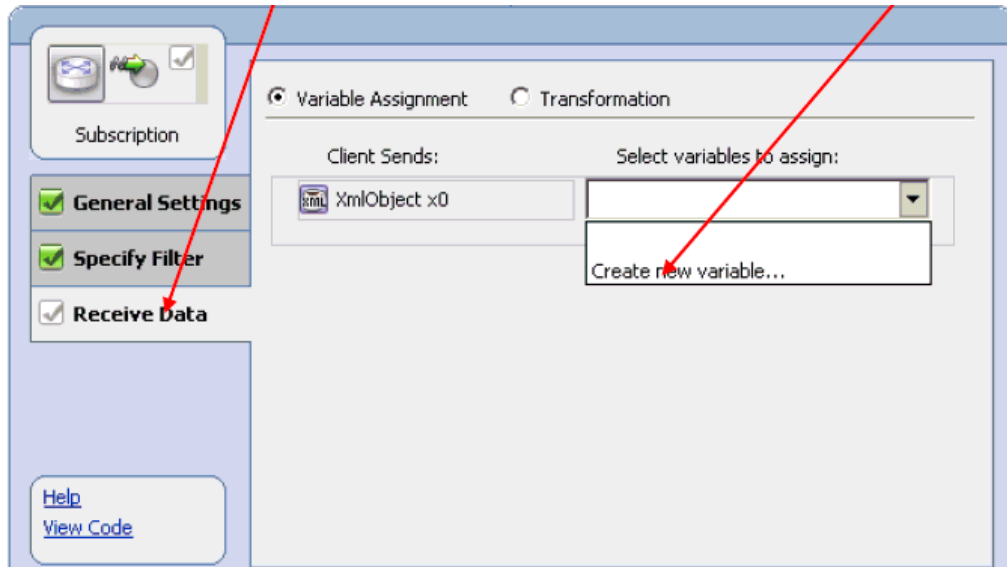


21. Double-click the Subscription node.

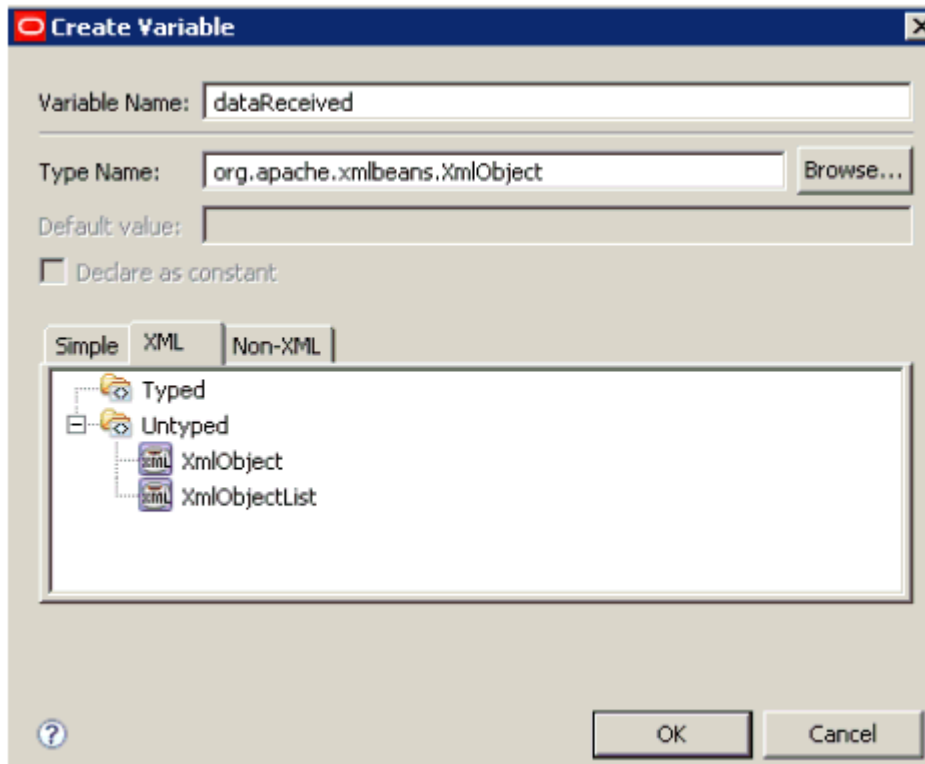
The Subscription configuration properties dialog box is displayed.



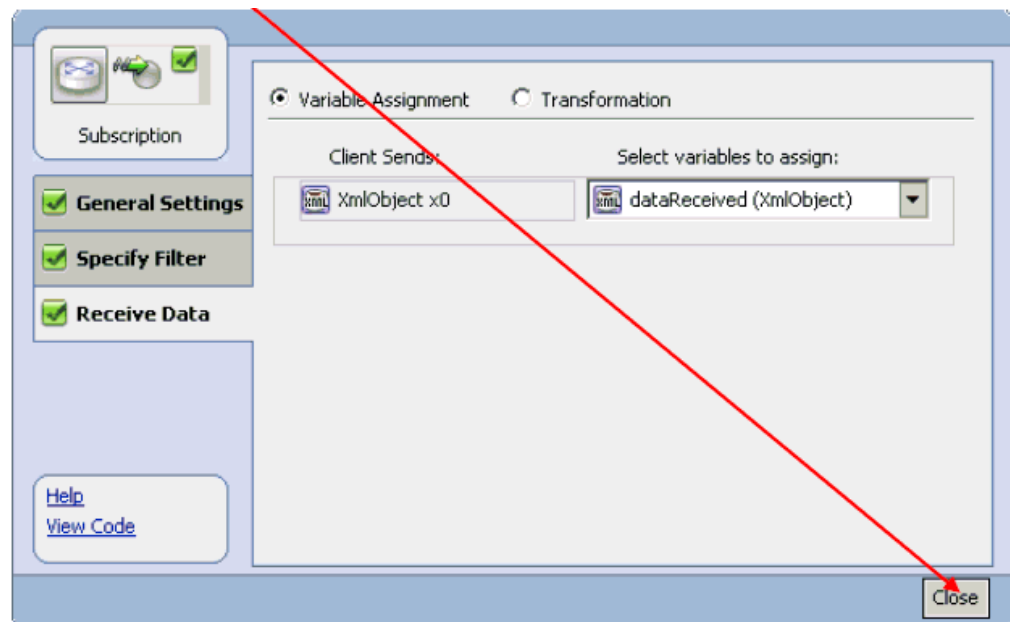
22. From the Channel Name drop-down list, select /samplePrefix/inbound_matmas/matmasinbound.
23. Click Receive Data in the left pane.



24. From the Select variables to assign drop-down list, select Create new variable. The Create Variable dialog box is displayed.

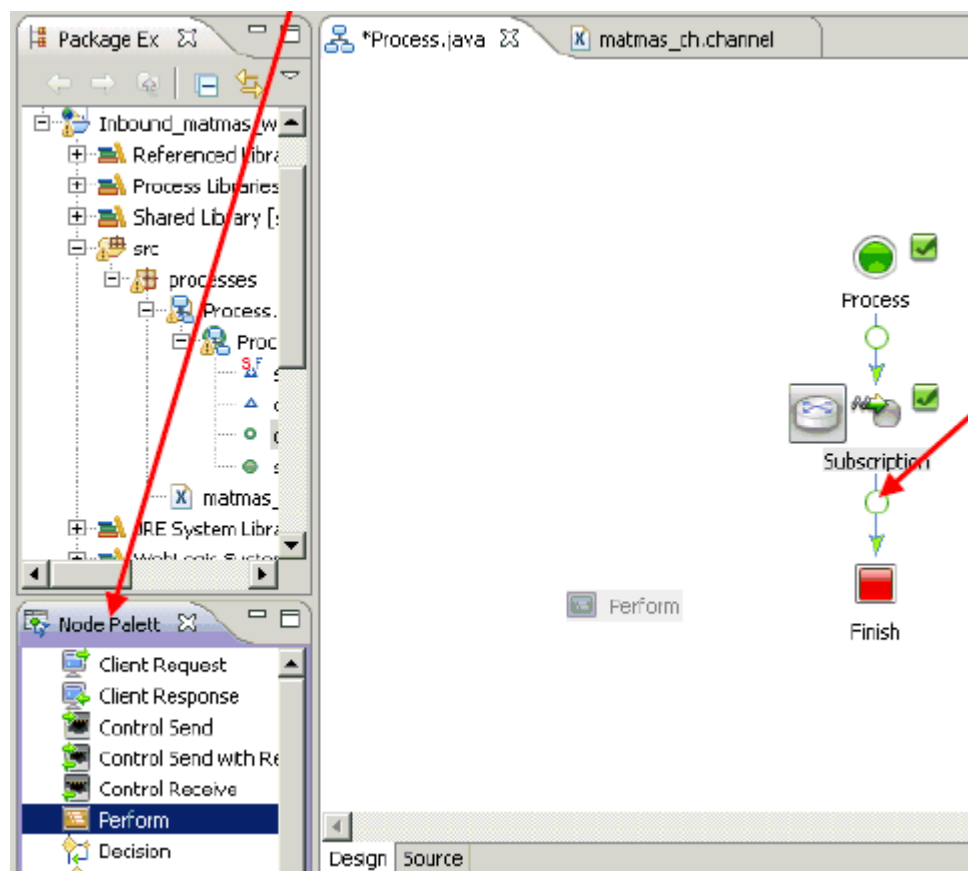


25. Specify a new variable name for the input data for example, dataReceived, and click OK. You are returned to the Subscription configuration properties dialog box, which has been refreshed with the new information.



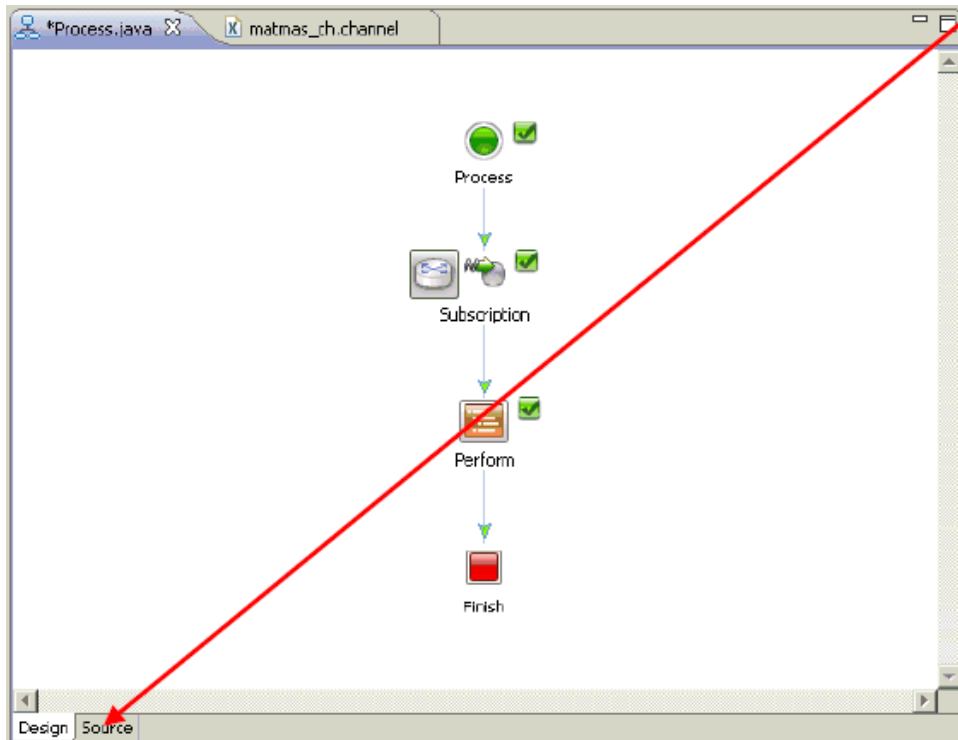
26. Click Close.

You are returned to the process design view (Process.java).



27. From the Node Palette area in the left pane, click the Perform node and drag it to the circle between the Subscription node and the Finish node.

The process design view (Process.java) is updated to reflect the new addition of the Perform node.



28. Click the Source tab.

```

public class Process implements ProcessDefinition {

    public org.apache.xmlbeans.XmlObject dataReceived;

    @com.bea.wli.jpd.Context
    JpdContext context;

    static final long serialVersionUID = 1;

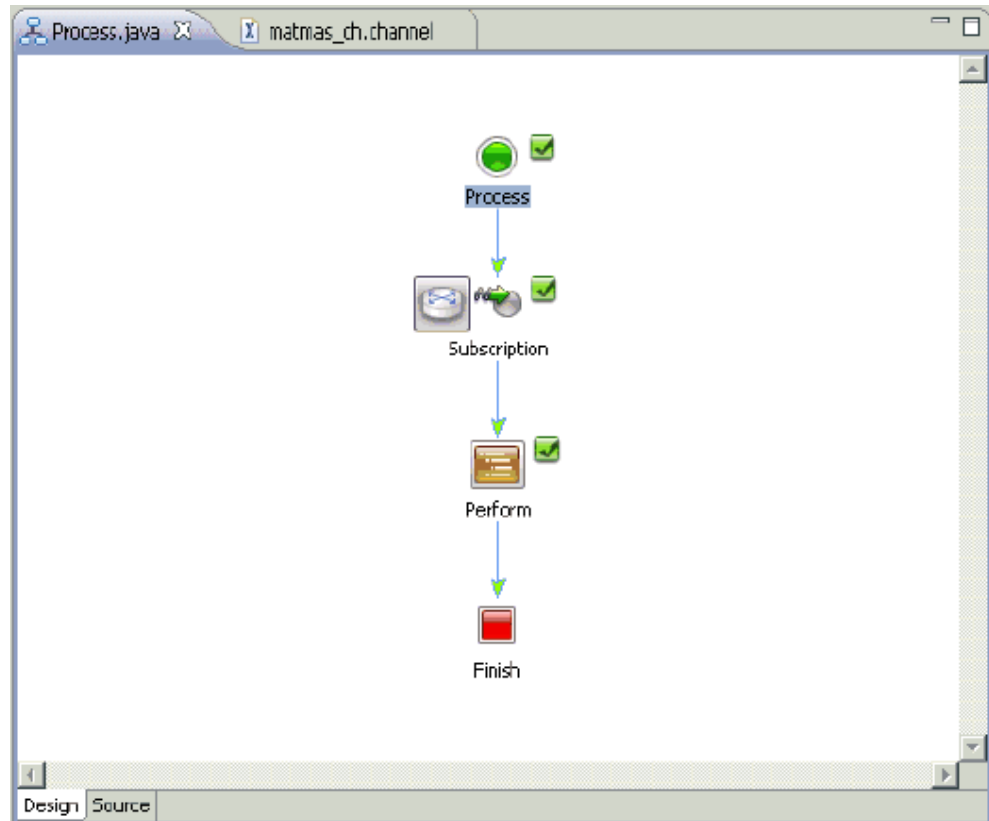
    @com.bea.wli.control.broker.MessageBroker.StaticSubscription(xqu
    public void subscription(org.apache.xmlbeans.XmlObject xO) {
        // #START: CODE GENERATED - PROTECTED SECTION - you can safe
        // input transform
        // parameter assignment
        this.dataReceived = xO;
        // #END : CODE GENERATED - PROTECTED SECTION - you can safe
    }

    public void perform() throws Exception {
        System.out.println(" Data Received .." +this.dataReceived);
    }
}
    
```

29. Make the following changes in the syntax to print the triggered IDOCs in the server traces:

```
public void perform() throws Exception {
    System.out.println(" Data Received .." +this.dataReceived);
}
```

30. Return to the process design view (Process.java), which is now finished.



31. Save the inbound process (Process.java).

You are now ready to deploy the inbound process on WebLogic Server.

Deploying the Inbound Process on WebLogic Server

This section describes how to deploy the inbound process on WebLogic Server.

Note: Depending on your platform (for example, Windows or Solaris), there are a few differences in the inbound deployment process.

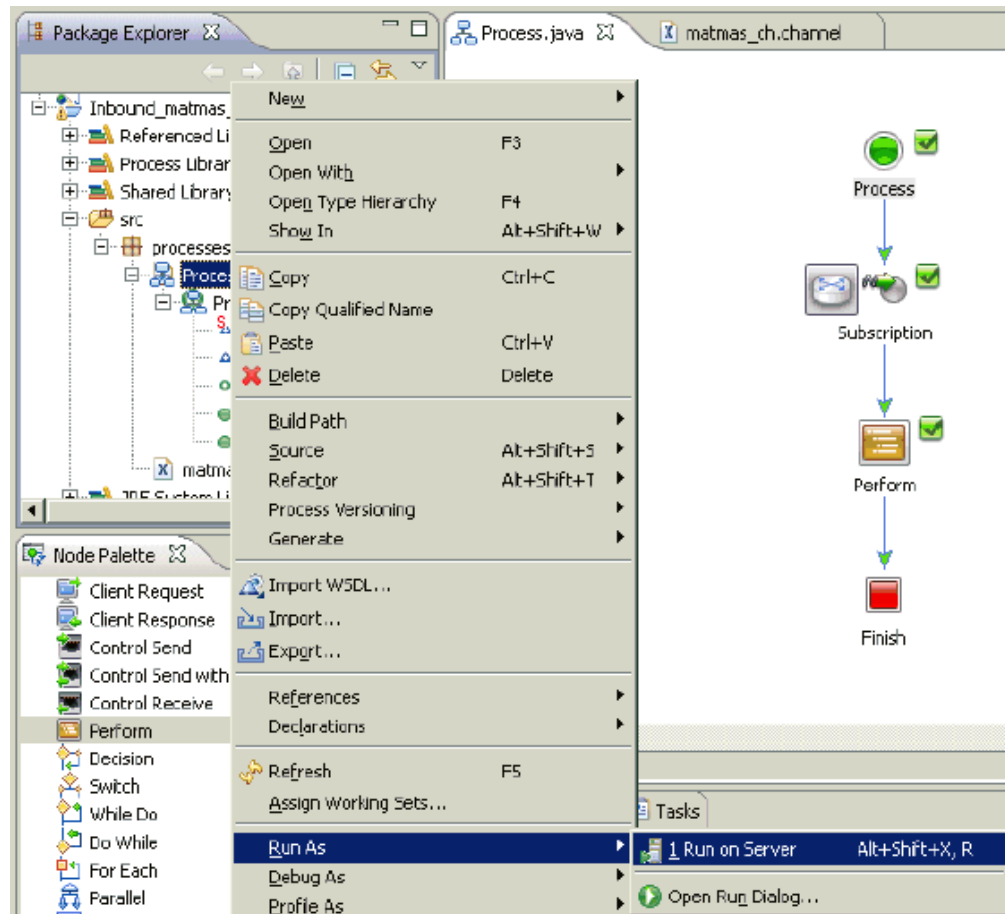
On Windows platforms, you can deploy the inbound process directly to the WebLogic Server. If you are using Windows, follow the procedure that is described in this section.

On Solaris platforms, you must first export the inbound process as an EAR file, deploy the inbound process on WebLogic Server, and then create a queue to receive the response document in the Test Console. The following sections can be used for reference purposes:

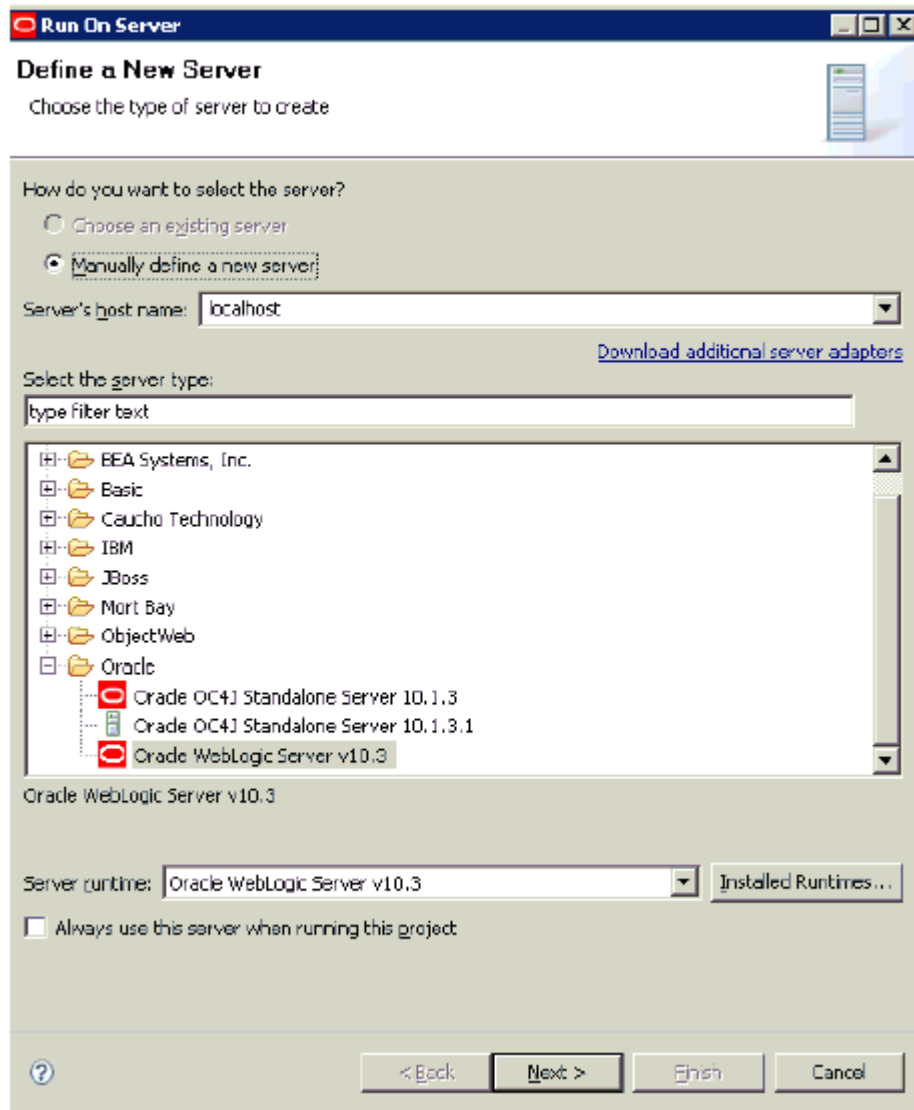
- [Exporting the Outbound JPD Process Application as an EAR File](#)
- [Deploying the Outbound JPD Process Application EAR File on WebLogic Server](#)
- [Creating a Queue to Receive the Response Document in the Test Console](#)

The procedures that are described in these sections use an outbound JPD process as an example. However, these procedures are also identical and valid for inbound JPD processes.

1. In the Package Explorer, right-click **Process**, select **Run As** from the context menu and click **Run on Server**.

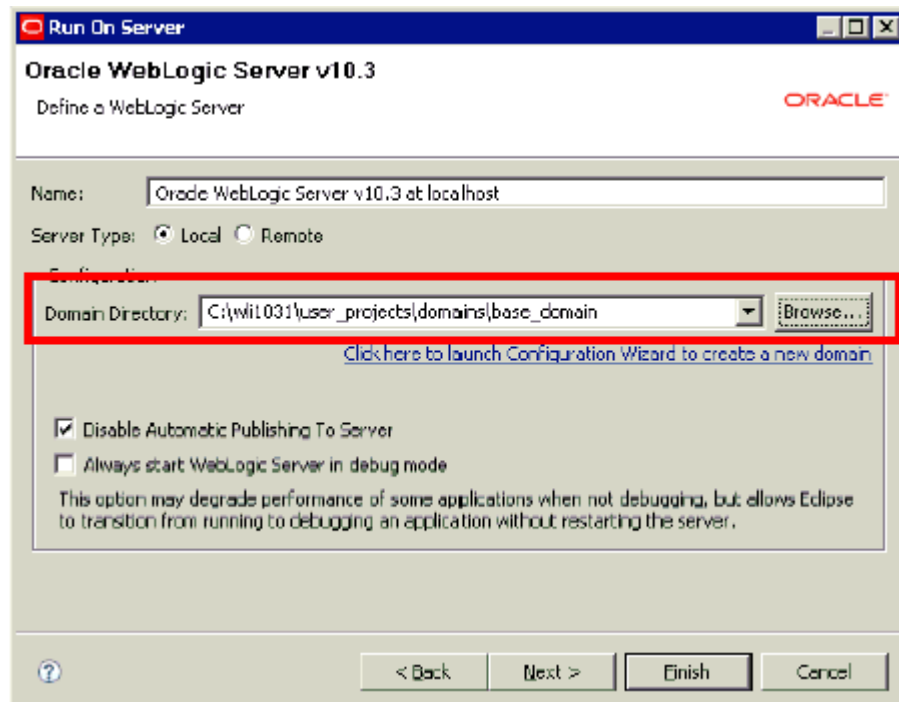


The Run On Server (Define a New Server) dialog box is displayed.



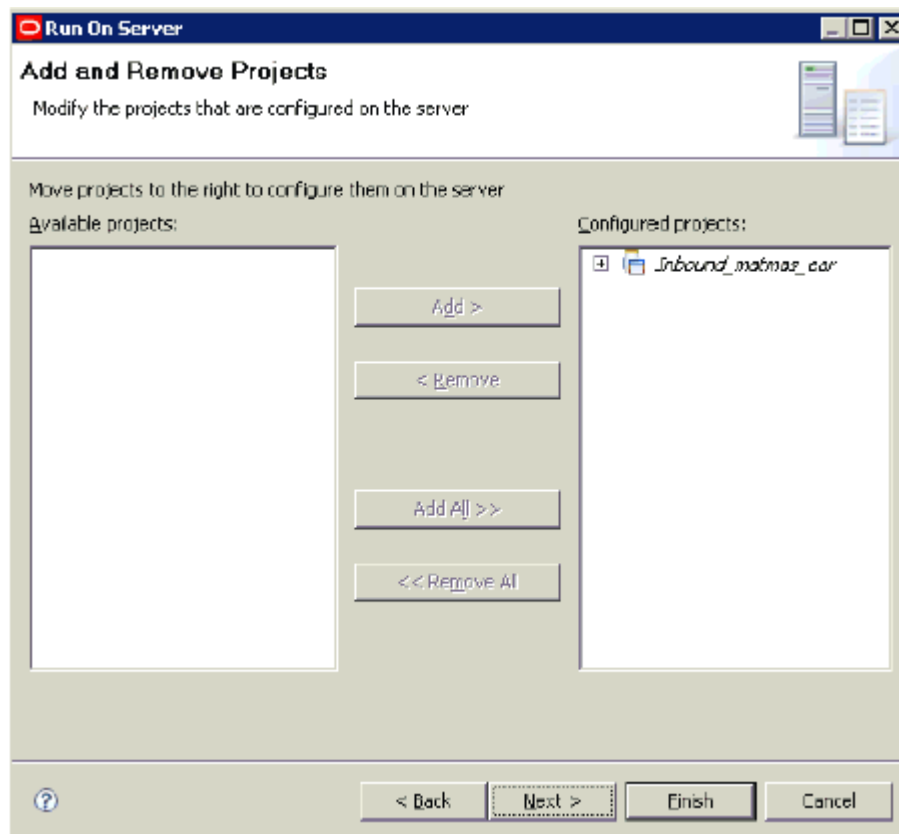
2. Verify that **Oracle WebLogic Server v10.3** is selected in the server type area and click **Next**.

The Run On Server (Oracle WebLogic Server v10.3) dialog box is displayed.



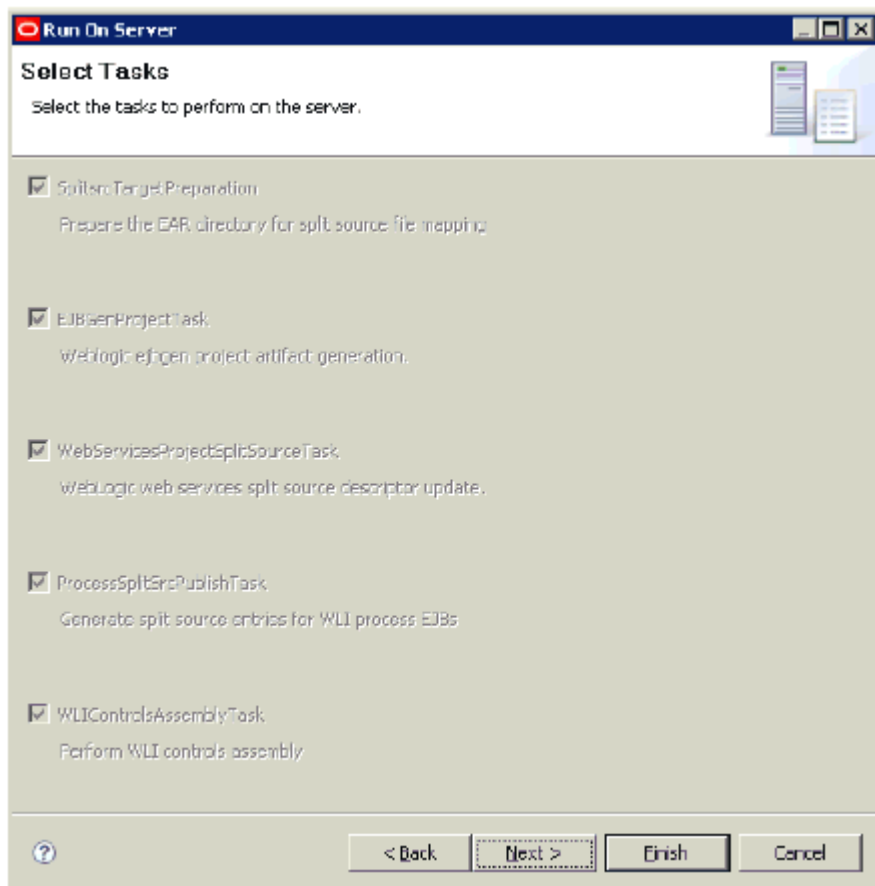
3. In the Domain Directory field, enter the path to your WebLogic Domain.
4. Click Next.

The Run On Server (Add and Remove Projects) dialog box is displayed.



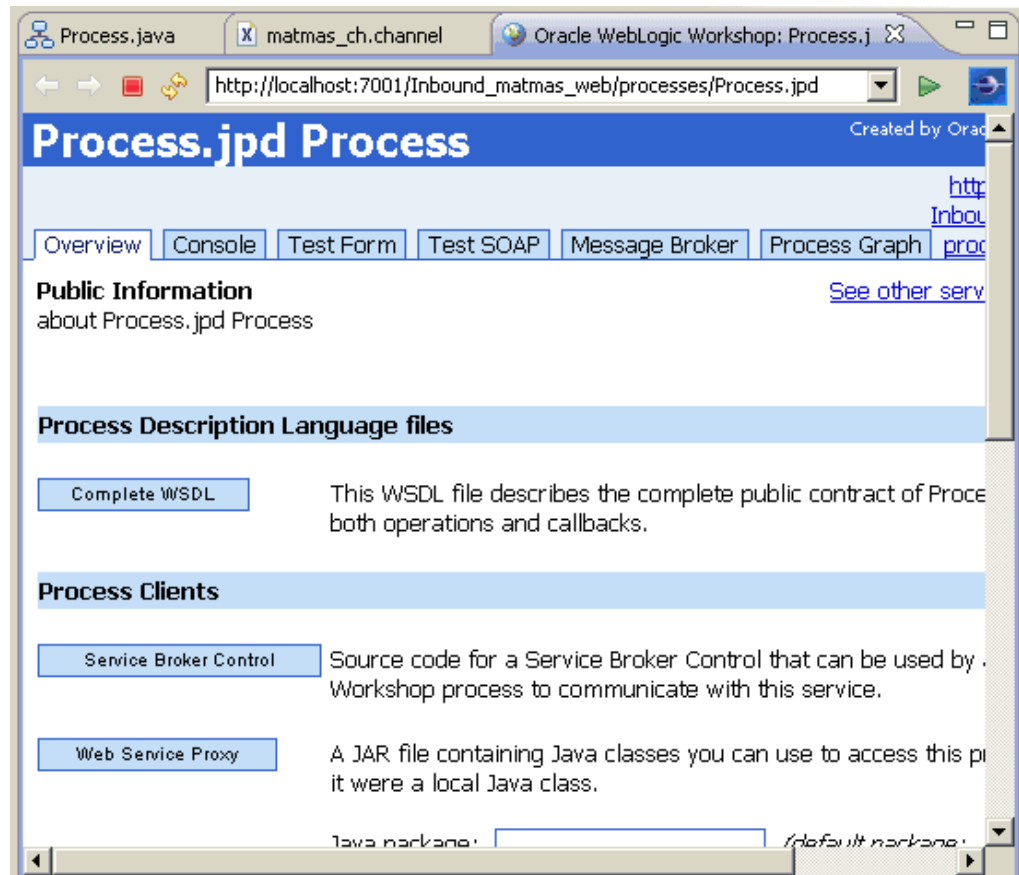
5. Click **Next**.

The Run On Server (Select Tasks) dialog box is displayed.



6. Click **Finish**.

Once deployed, a Web browser is opened, as shown in the following image.



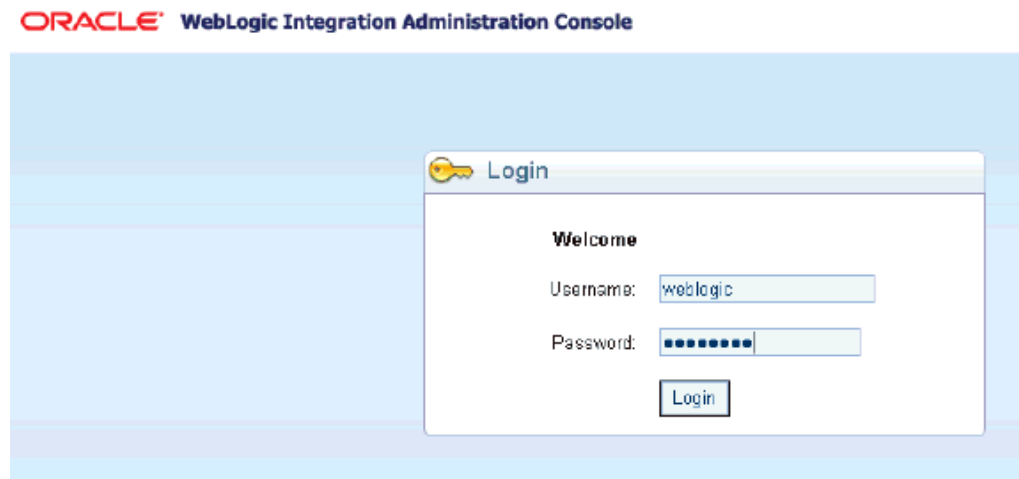
You are now ready to create an Event Generator and Channel Subscription.

Creating an Event Generator and a Channel Subscription

This section describes how to create an Event Generator and a Channel Subscription.

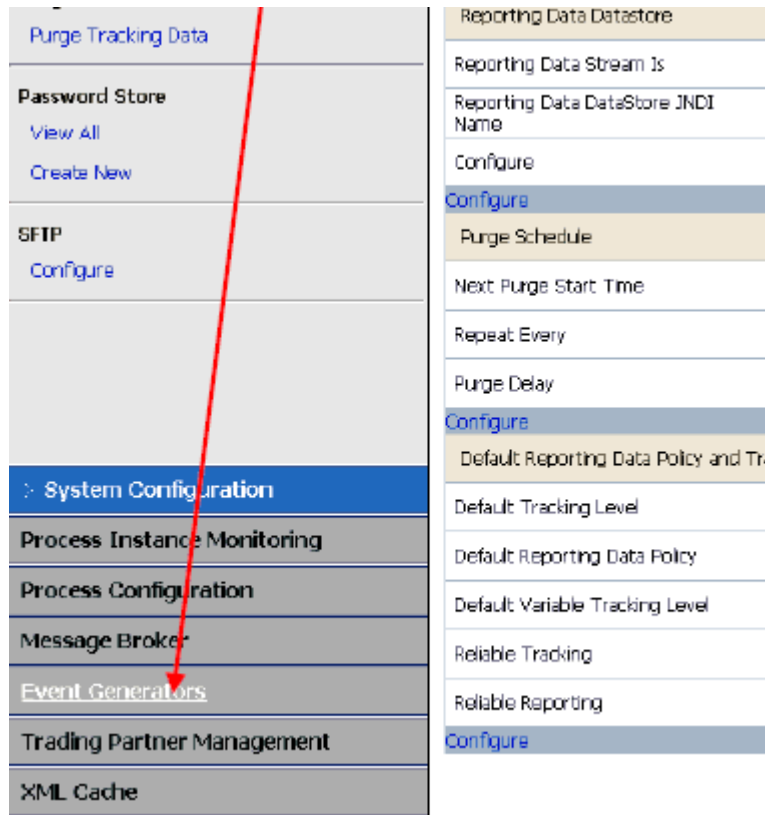
1. Enter the following URL in your Web browser to logon to the Oracle WebLogic Integration Administration Console:

`http://localhost:7001/wliconsole`

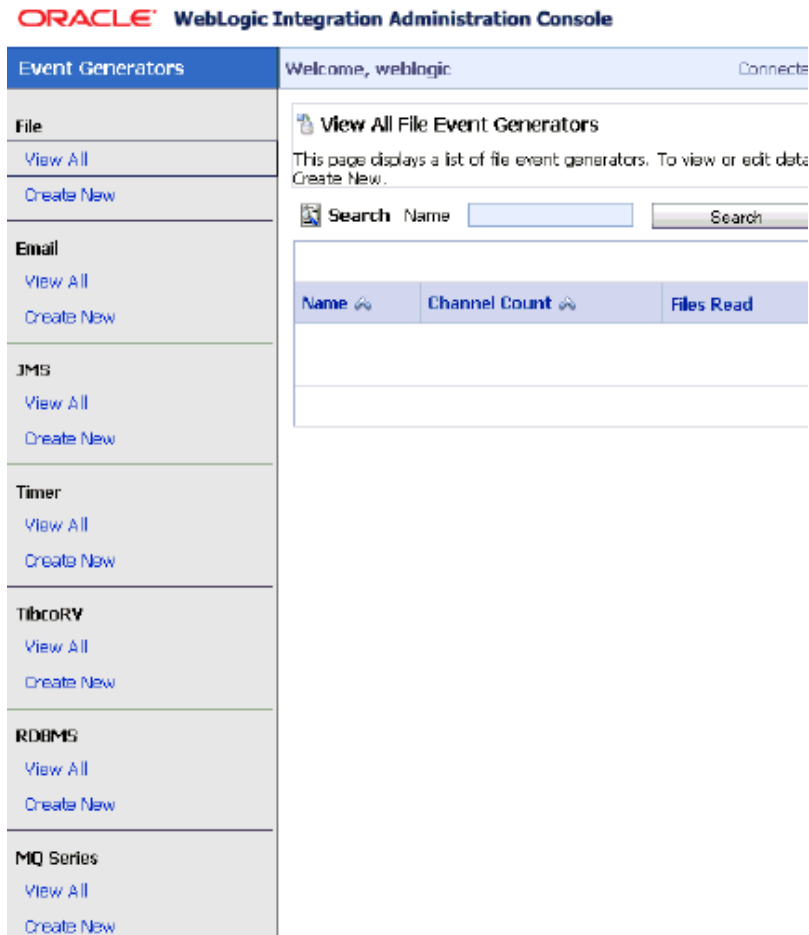


2. Enter **weblogic** in the Username and Password fields.
3. Click Login.

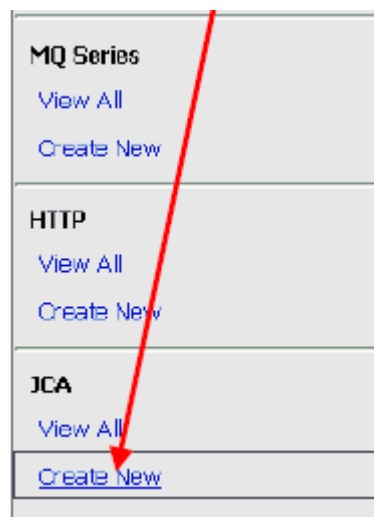
The Oracle WebLogic Integration Administration Console is displayed, as shown in the following image.



4. Click Event Generators in the left pane.
The Event Generators page is displayed.



5. Scroll down the list of available protocols in the left pane until you find JCA.



6. Click Create New in the JCA area.
The Create a New JCA Event Generator page is displayed.

Welcome, weblogic Connected to : base_domain Home WLS Console LOGOUT Help AskOracle

Create a New JCA Event Generator

Use this page to create a new JCA Event Generator. Although new event generators are deployed immediately, they do not have channel rules. You can add rules after you create the new event generator.

Generator Name The name of the event generator must be unique.

WSDL File Please provide JCA WSDL File Server location.

JMS Connection Factory JNDI Name

Next > Reset Cancel

7. Perform the following steps:
 - a. In the Generator Name field, specify a name for the new JCA Event Generator (for example, matmas_inbound).
 - b. In the WSDL File field, specify the path to the inbound WSDL file that was created using Application Explorer. For example:
 c:\wli1031\erp-adapters\wsdls\MATMAS01_receive_sep10.wsdl
 - c. In the JMS Connection Factory JNDI Name drop-down list, accept the default value that is selected.
8. Click Next.

The following page is displayed.

Welcome, weblogic Connected to : base_domain Home WLS Console LOGOUT Help AskOracle

Create a New JCA Event Generator

Use this page to create a new JCA Event Generator. Although new event generators are deployed immediately, they do not have channel rules. You can add rules after you create the new event generator.

Generator Name The name of the event generator must be unique.

Port Name

Operation

Dispatch Policy

Submit < Back Reset Cancel

9. Accept the default values and click Submit.

The new JCA event generator (matmas_inbound) is created.

Welcome, weblogic Connected to : base_domain

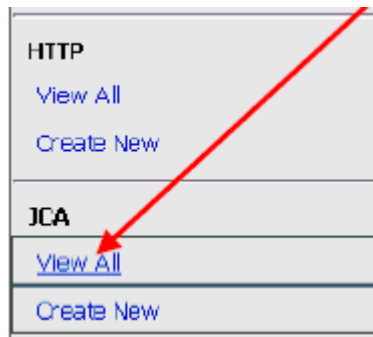
JCA Event Generator Definition

This page displays the channel rules defined for this JCA event generator. To add a new channel rule, click the Channel Name for the rule.

Generator Name matmas_inbound

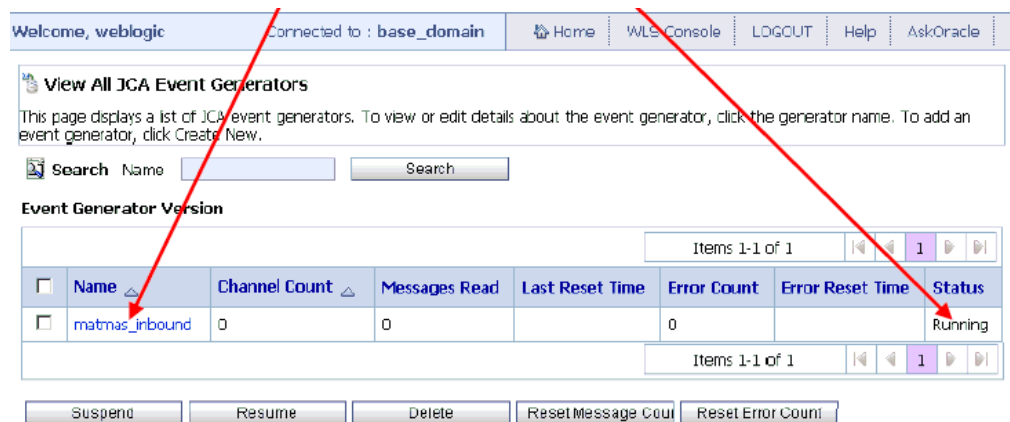
| Channel Name |
|----------------|
| No matching di |

10. Scroll down the list of available protocols in the left pane until you find JCA.



11. Click View All.

A list of all the available JCA Event Generators is displayed.

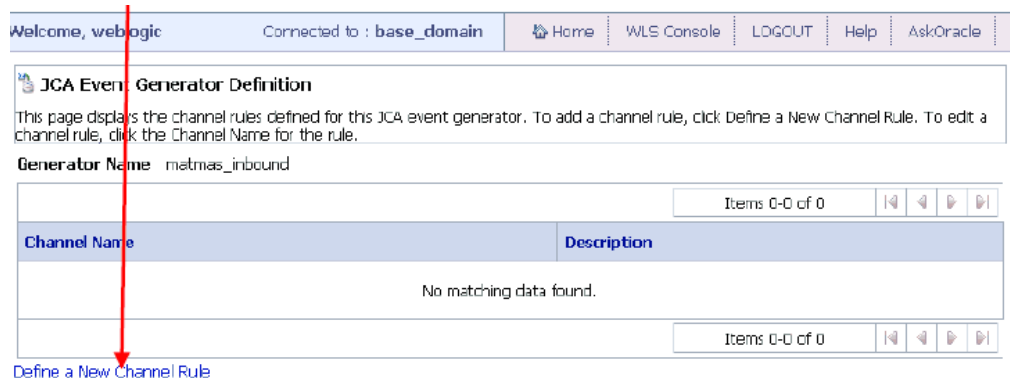


12. In the Status column, verify that the Event Generator status is Running.

This means that the JCA Event Generator was created successfully.

13. In the Name column, click the name of the JCA Event Generator (for example, matmas_inbound).

The JCA Event Generator Definition page is displayed.



14. Click Define a New Channel Rule.

The JCA Generator Channel Rule Definition page is displayed.

15. Perform the following steps:

- a. From the Channel Name drop-down list, select the channel that you created earlier during the inbound JPD process creation procedure (for example, matmasinbound). This same channel is also available in the Oracle WebLogic Integration Administration Console.
- b. In the Description field, specify an optional description for the channel rule.
- c. From the Publish As drop-down list, select weblogic.

16. Click Submit.

The new channel rule definition for the JCA Event Generator (matmas_inbound) has been successfully created.

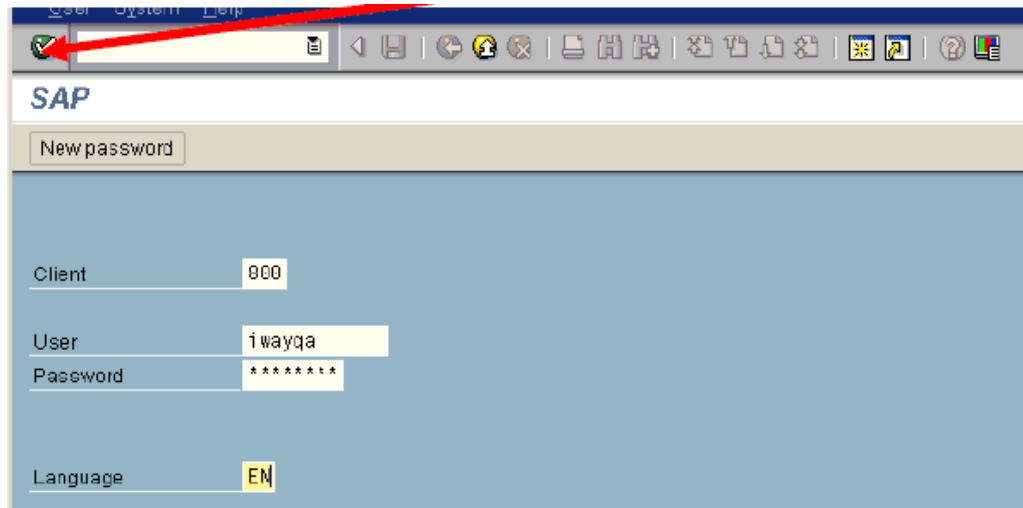
Any event that is now triggered for an IDOC from SAP GUI will be received by the JCA Event Generator (matmas_inbound). This in turn will send the event message over the Message Broker channel that is configured. As a result, the Message broker channel will execute the inbound JPD process. This is a typical scenario in which an inbound JPD process will be executed.

You are now ready to trigger an event from SAP GUI.

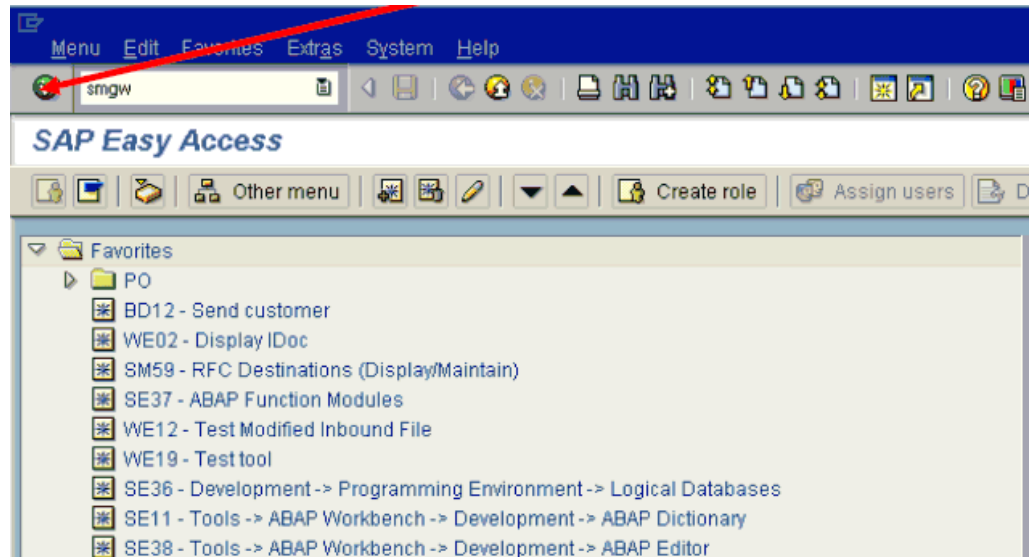
Triggering an Event From SAP GUI

This section describes how to trigger an event from SAP GUI.

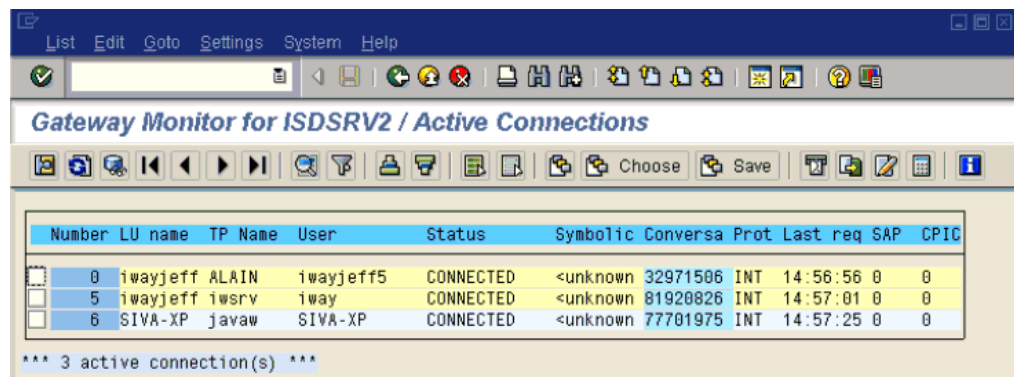
1. Launch the SAP Workbench and logon to the SAP system.



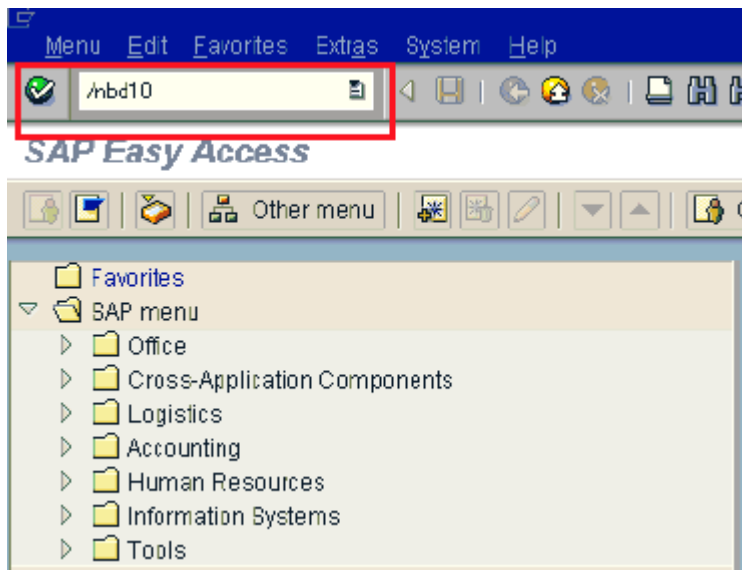
- Click the green check mark icon in the upper-left corner to continue.
The SAP Easy Access screen is displayed.



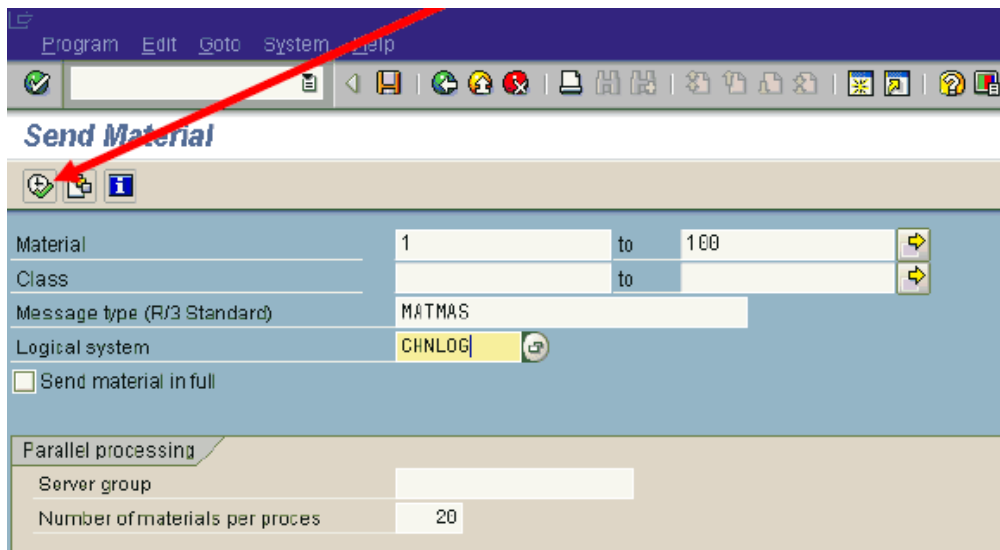
- Enter smgw in the navigation menu and click the green check mark icon.
The Gateway Monitor for ISDSRV2 / Active Connections screen is displayed.



4. Verify that your channel is active and listening. If it is listening, then you can trigger an event.
5. To trigger MATMAS, enter /nbd10 in the navigation menu and click the green check mark icon.



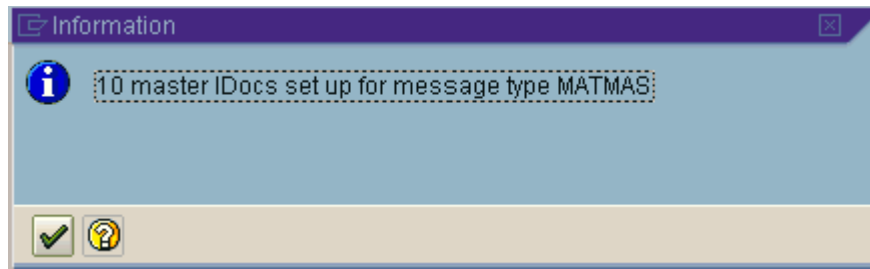
The Send Material screen is displayed.



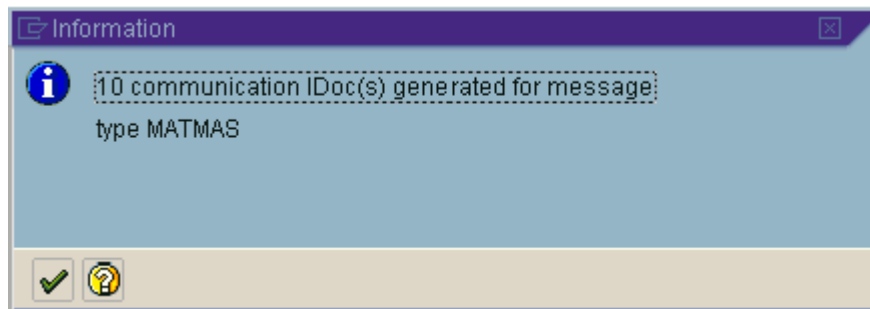
6. Perform the following steps:
 - a. In the Material field, specify a range of 1 to 100.
 - b. In the Message Type field, type MATMAS.
 - c. In the Logical System field, type CHNLOG.
7. Click the following icon:



The following message is displayed.



8. Click the green check mark.
The following message is displayed.

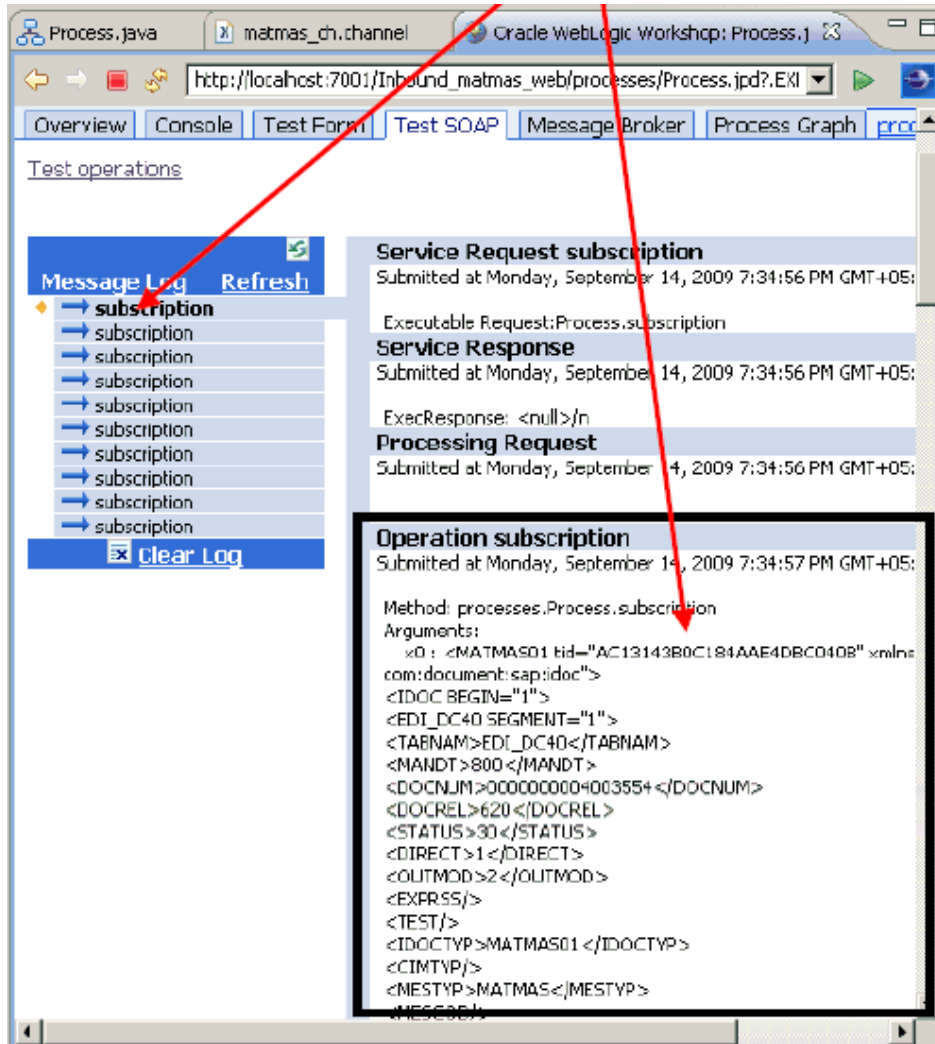


9. Click the green check mark.
You are now ready to verify the event.

Verifying the Event

This section describes how to verify the event.

1. Once the SAP IDOCs are triggered, you can check the instances from the Test SOAP tab (for the inbound JPD process deployed) by clicking on the subscription in the left pane.
The IDOCs received can be seen in the Operation subscription area.



2. You can monitor and determine the status of the inbound JPD process from the Process Instance Statistics pane.

The screenshot shows the 'Process Instance Statistics' page. It includes a search bar and a table with the following data:

| Display Name | Average Elapsed | Running | Susp. | Aborted | Frozen | Term. | Comp. |
|------------------------------------|-----------------|---------|-------|---------|--------|-------|-------|
| /inbound_web/processes/Process.jsp | 25 msec | 0 | 0 | 0 | 0 | 0 | 10 |
| Process | 36 msec | N/A | 0 | 0 | 0 | 0 | 10 |

A red arrow points from the 'Process' row in the table to the 'Process Instance Statistics' header.

3. Verify that every instance is complete for the inbound JPD process.
If any instance is *Terminated* or *Aborted*, then some exceptions have occurred during the process.
4. Click Process to view the instances.

The Process Instance Summary pane is displayed where all the completed instances are listed.

Process Instance Summary

This page displays a summary of process instances. Use the search boxes to filter the displayed instances. To view instance details, click the Instance ID.

Search

Search Instance ID

Search Process Label

Number of Instances Displayed Per Page

Items 1-10 of 10

| <input type="checkbox"/> | ID | Display Name | Process Label | Start Time | Elapsed Time | Status |
|--------------------------|--|--------------|---------------|-----------------|--------------|-----------|
| <input type="checkbox"/> | 192.168.128.164--fe402cb.123b8b7ad2c.-7fdc | Process | | 9/14/09 7:35 PM | 15 msecs | Completed |
| <input type="checkbox"/> | 192.168.128.164--fe402cb.123b8b7ad2c.-7fde | Process | | 9/14/09 7:35 PM | 0 ms | Completed |
| <input type="checkbox"/> | 192.168.128.164--fe402cb.123b8b7ad2c.-7fe0 | Process | | 9/14/09 7:35 PM | 15 msecs | Completed |
| <input type="checkbox"/> | 192.168.128.164--fe402cb.123b8b7ad2c.-7fe2 | Process | | 9/14/09 7:35 PM | 15 msecs | Completed |
| <input type="checkbox"/> | 192.168.128.164--fe402cb.123b8b7ad2c.-7fe4 | Process | | 9/14/09 7:35 PM | 0 ms | Completed |
| <input type="checkbox"/> | 192.168.128.164--fe402cb.123b8b7ad2c.-7fe6 | Process | | 9/14/09 7:35 PM | 15 msecs | Completed |
| <input type="checkbox"/> | 192.168.128.164--fe402cb.123b8b7ad2c.-7fe8 | Process | | 9/14/09 7:34 PM | 47 msecs | Completed |
| <input type="checkbox"/> | 192.168.128.164--fe402cb.123b8b7ad2c.-7fea | Process | | 9/14/09 7:34 PM | 0 ms | Completed |
| <input type="checkbox"/> | 192.168.128.164--fe402cb.123b8b7ad2c.-7fec | Process | | 9/14/09 7:34 PM | 15 msecs | Completed |
| <input type="checkbox"/> | 192.168.128.164--fe402cb.123b8b7ad2c.-7fee | Process | | 9/14/09 7:34 PM | 0.2 secs | Completed |

- Click on a particular instance to view the SAP IDOC.

In addition, the triggered SAP IDOCs can be viewed in the server traces (for example, received through the *dataReceived* variable).

```

Data Received ..<MATMAS01 tid="AC13143B0C184AAF433B04F3" xmlns="urn:sap-com:doc
ument:sap:idoc">
  <IDOC BEGIN="1">
    <EDI_DC40 SEGMENT="1">
      <TABNAM>EDI_DC40</TABNAM>
      <MANDT>800</MANDT>
      <DOCNUM>0000000004003723</DOCNUM>
      <DOCREL>620</DOCREL>
      <STATUS>30</STATUS>
      <DIRECT>1</DIRECT>
      <OUTMOD>2</OUTMOD>
      <EXPRS />
      <TEST />
      <IDOCTYP>MATMAS01</IDOCTYP>
      <CIMTYP />
      <MESTYP>MATMAS</MESTYP>
      <MESCOD />
      <MESFCT />
      <STD />
      <STDURS />
      <STDMES />
      <SNDPOR>SAPP47</SNDPOR>
      <SNDPRT>LS</SNDPRT>
      <SNDPFC />
      <SNDPRN>T90CLNT090</SNDPRN>
      <SNDSAD />
      <SNDLAD />
      <RCUPOR>A000000000</RCUPOR>
      <RCUPRT>LS</RCUPRT>
      <RCUPFC />
      <RCUPRN>CHNLOG</RCUPRN>
      <RCUSAD />
      <RCULAD />
      <CREDAT>20090915</CREDAT>
      <CRETIM>033315</CRETIM>
      <REFINT />
      <REFGRP />
      <REFMES />
      <ARCKEY />
      <SERIAL>20090915033315</SERIAL>
    </EDI_DC40>
  <E1MARAM SEGMENT="1">

```

Configuring Outbound Processing Using Oracle WebLogic Integration (BSE Configuration)

To integrate with Oracle WebLogic Integration (WLI), Oracle Application Adapter for SAP R/3 must be deployed on the same Oracle WebLogic Server as WLI. The underlying adapter services must be exposed as WSDL files, which are generated during design time in Oracle WebLogic Server Adapter Application Explorer (Application Explorer) for request-response (outbound) services of the adapter. Oracle Application Adapter for SAP R/3 integrates seamlessly with WLI to facilitate Web service integration.

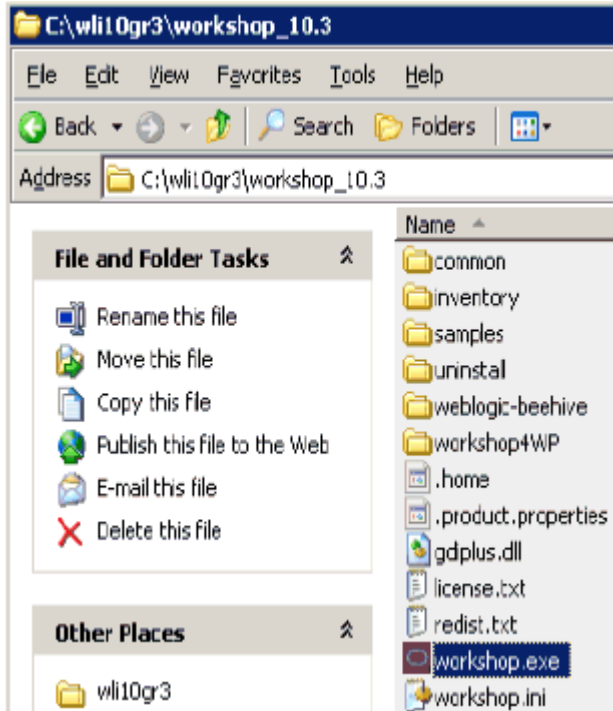
This chapter includes the following topics:

- [Creating an Outbound JPD Process Application](#)
- [Exporting the Outbound JPD Process Application as an EAR File](#)
- [Deploying the Outbound JPD Process Application EAR File on WebLogic Server](#)
- [Creating a Queue to Receive the Response Document in the Test Console](#)
- [Invoking the Input XML Using a Web Browser](#)

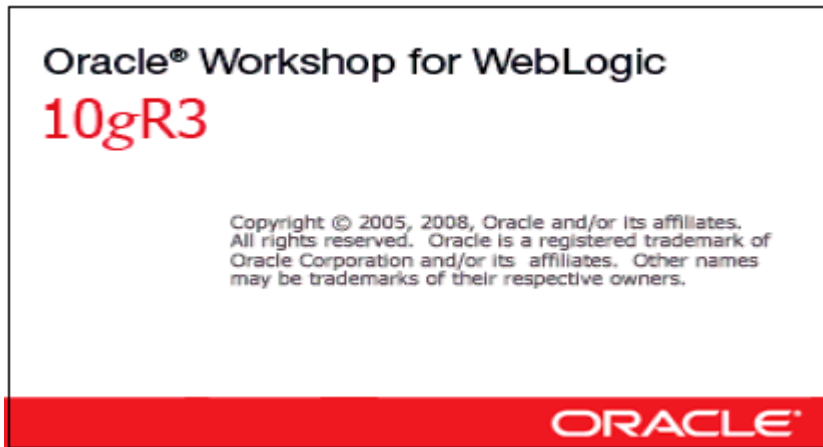
Creating an Outbound JPD Process Application

This section describes how to create an outbound JPD process application.

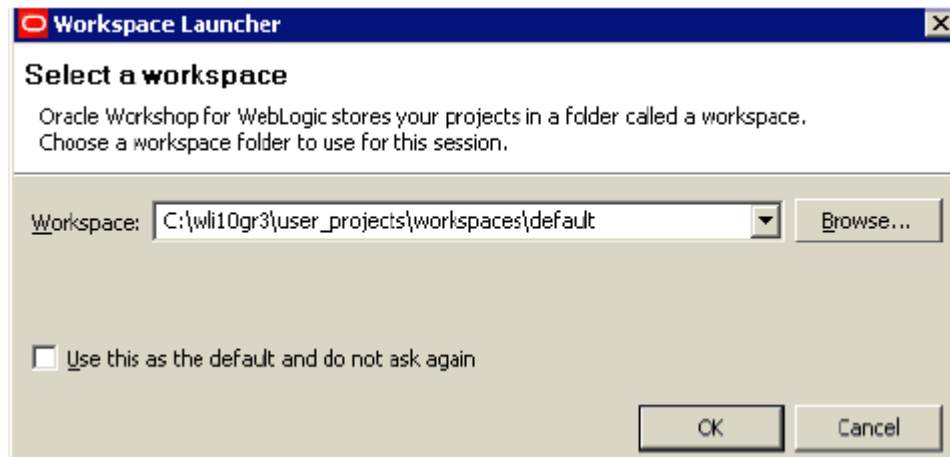
1. Navigate to the WebLogic Integration Workshop location (for example, `c:\wli10gr3\workshop_10.3`) and double-click the **workshop.exe** file.



The Oracle Workshop for WebLogic screen is displayed.



This is followed by the Workspace Launcher dialog box, as shown in the following image.



2. Specify the workspace location or accept the default path, which is:

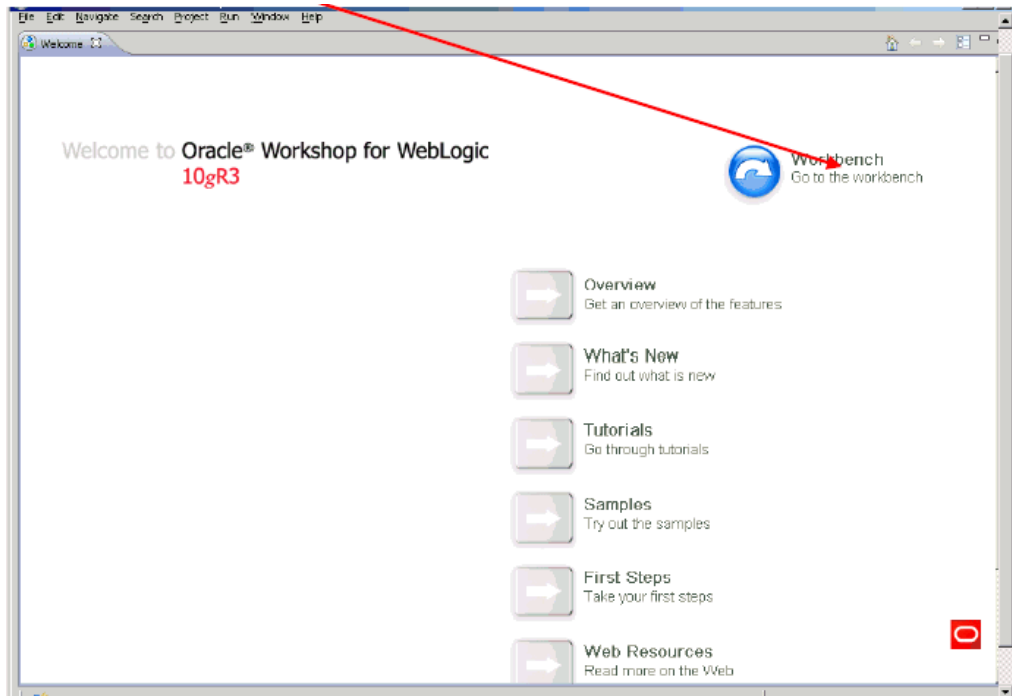
C:\wli10gr3\user_projects\workspaces\default

3. Click **OK**.

The Oracle Workshop for WebLogic loading screen is displayed.

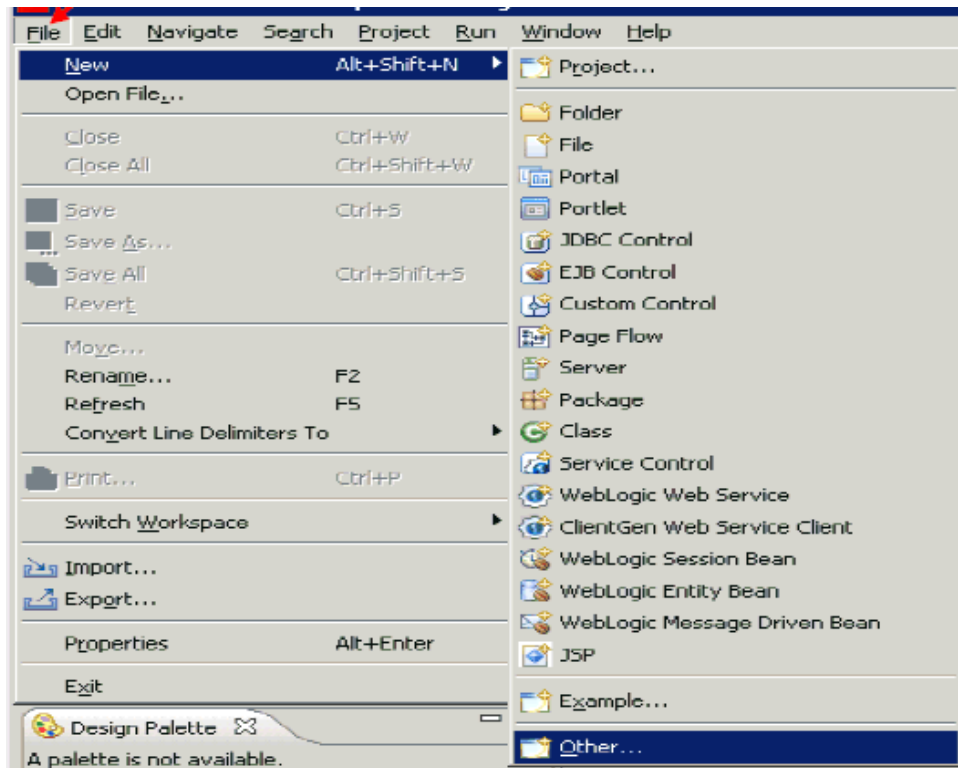


This is followed by the Oracle Workshop for WebLogic Welcome screen, as shown in the following image.



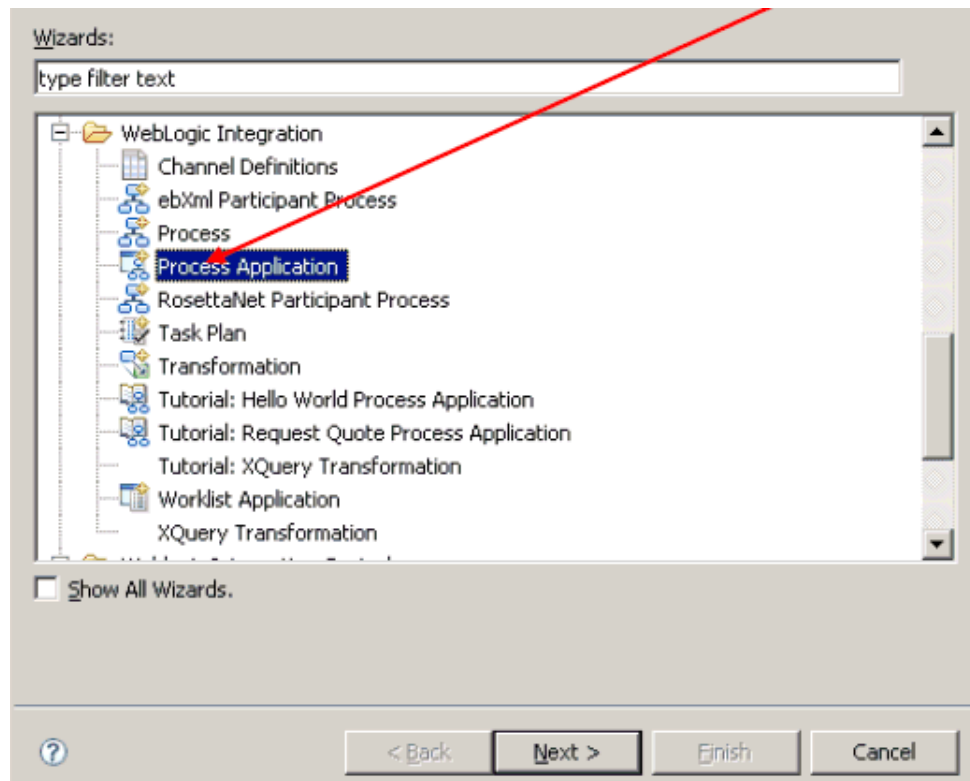
4. Click **Workbench**.

The Workbench area is displayed.



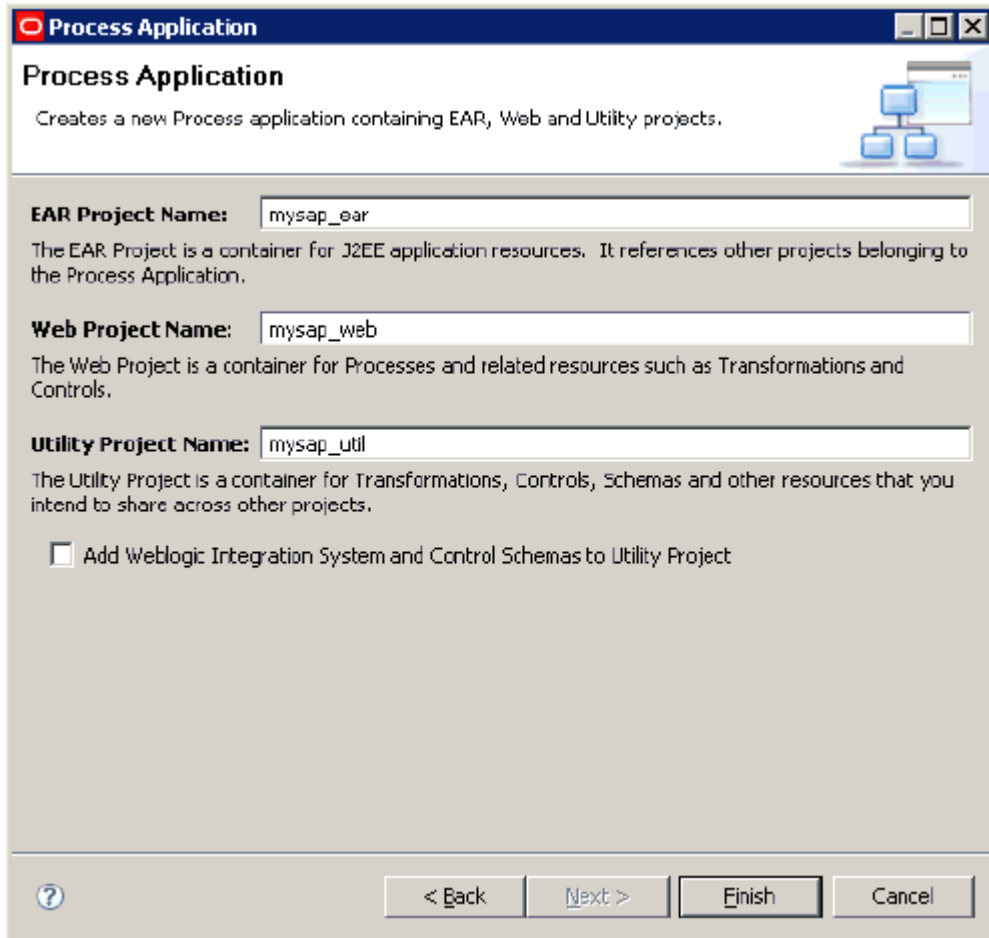
5. From the File menu, select **New** and click **Other**.

The following dialog box is displayed.



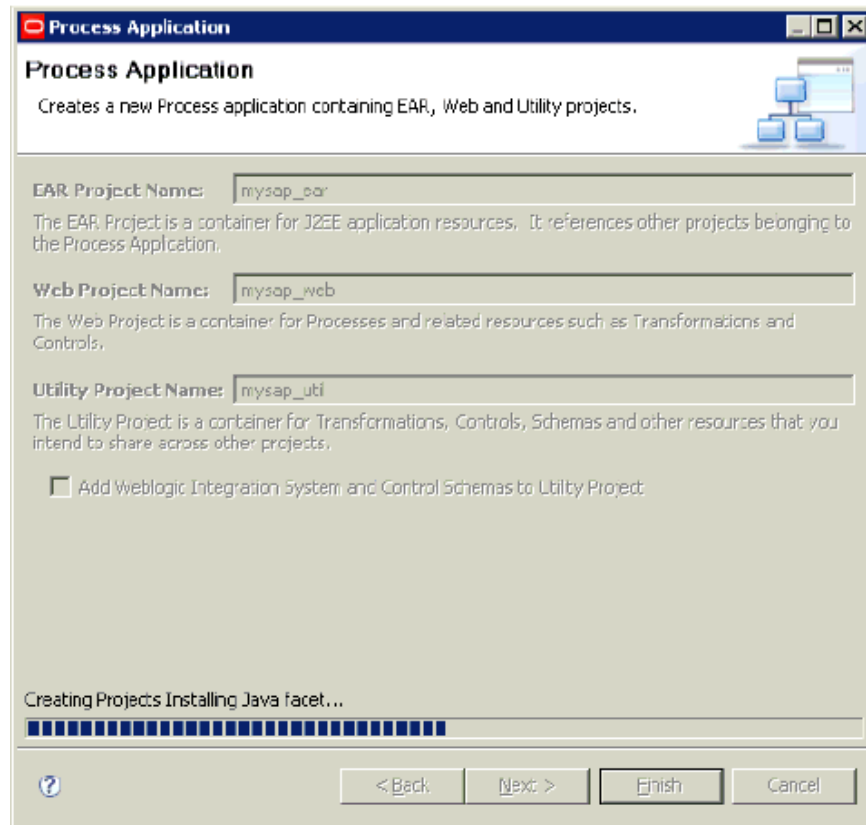
6. Expand the **WebLogic Integration** folder, select **Process Application**, and click **Next**.

The Process Application dialog box is displayed.

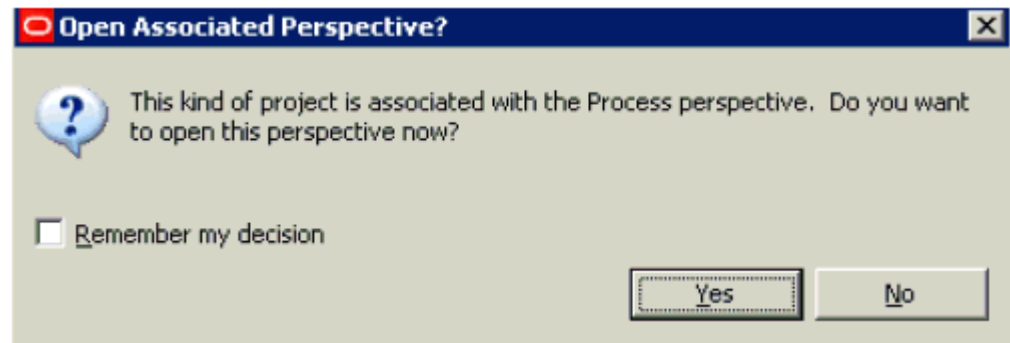


7. Perform the following steps:
 - a. Specify an EAR project name, for example, **mysap_ear**.
 - b. Specify a Web project name, for example, **mysap_web**.
 - c. Specify a Utility project name, for example, **mysap_util**.
8. Click **Finish**.

The EAR, Web, and Utility projects are created by a background process, as shown by the following status indicator.

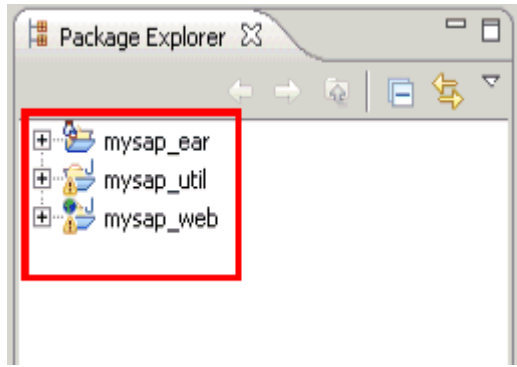


After this process is complete, the following message is displayed.

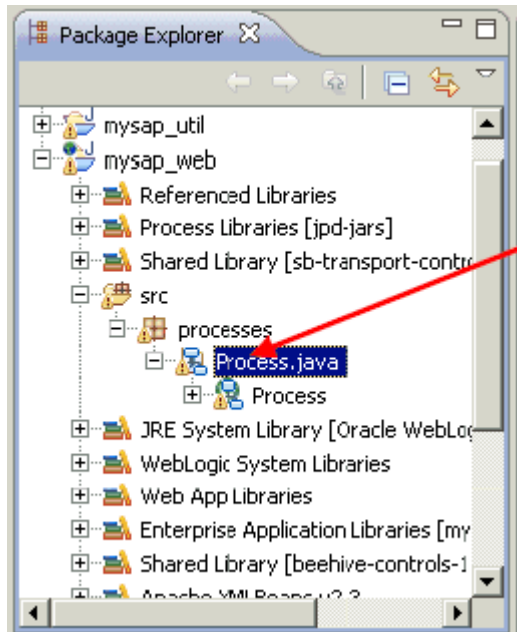


9. Click Yes to open the associated perspective.

The EAR, Web, and Utility projects are created in the Package Explorer area in the left pane.

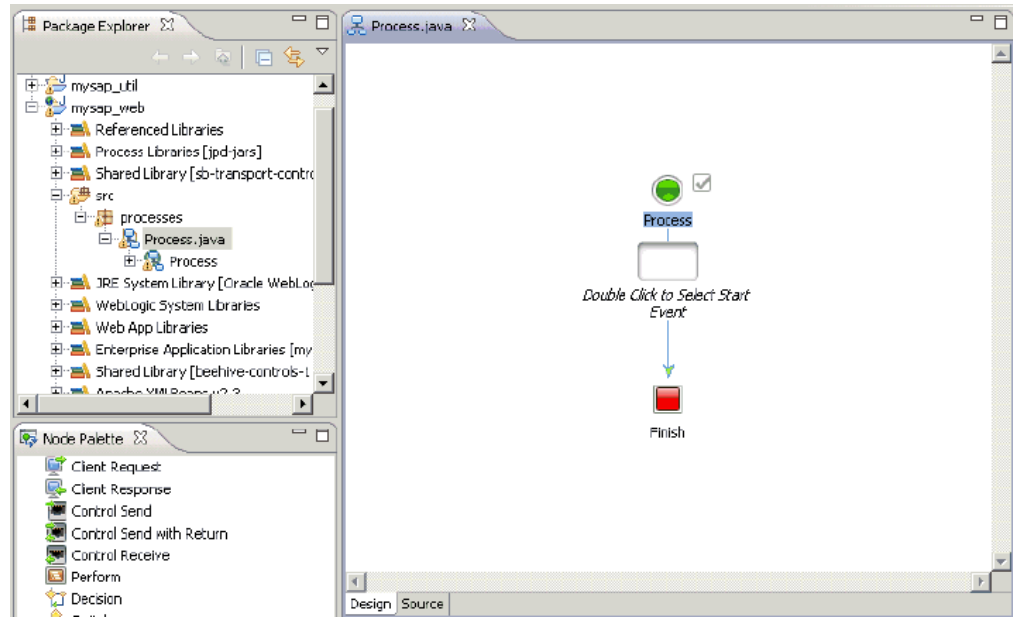


10. Expand the Web project node (for example, **mysap_web**), followed by **src** and **processes**.

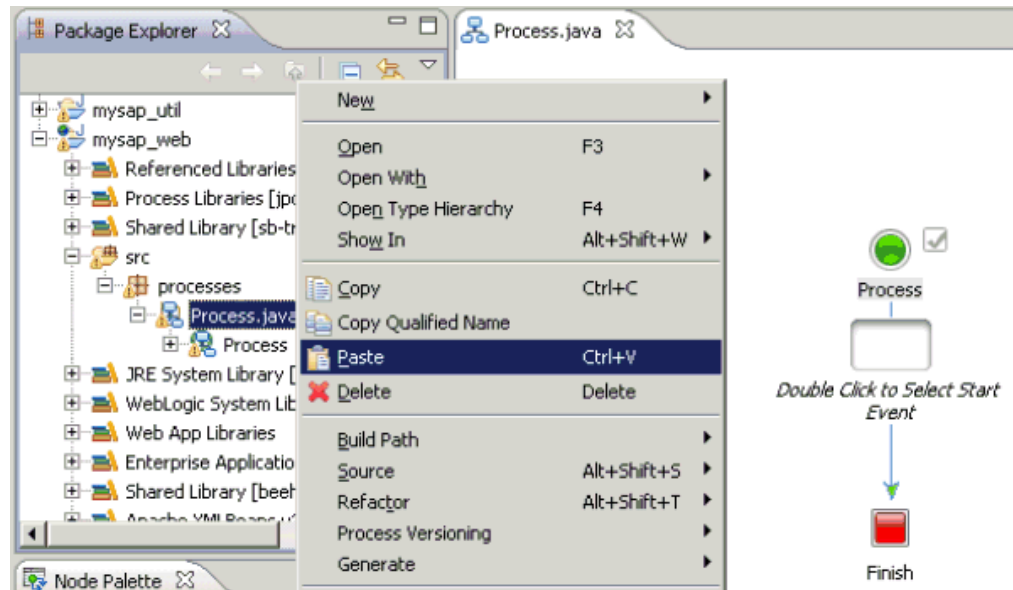


11. Double-click **Process.java**.

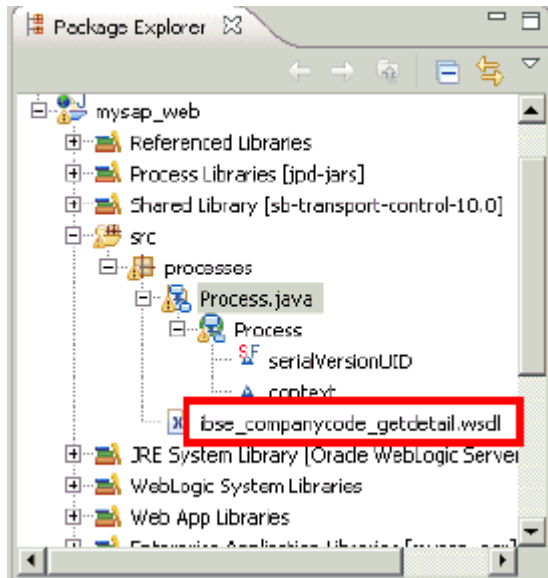
The process design view (**Process.java**) is displayed in the central pane.



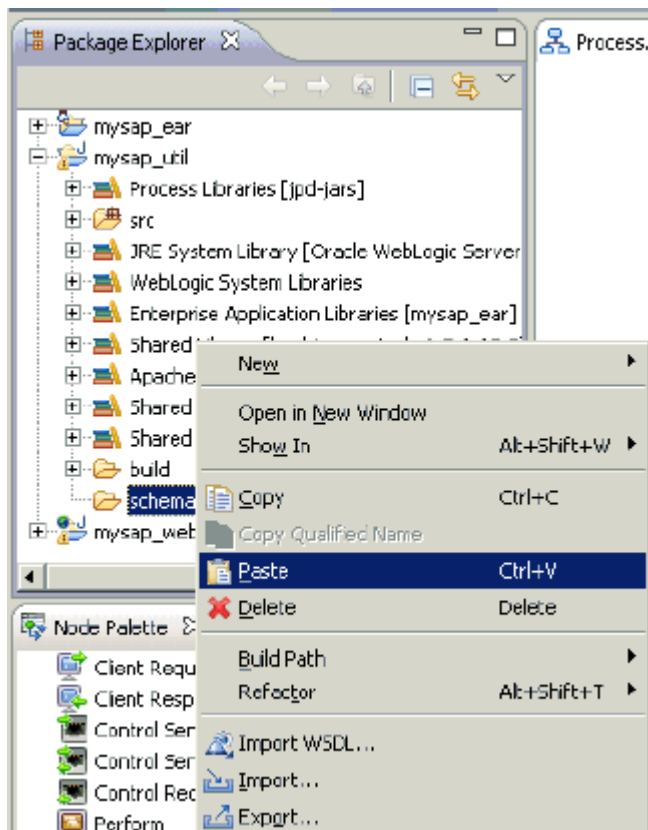
- Copy and paste the generated outbound WSDL file that was created using Application Explorer under the Process.java node.



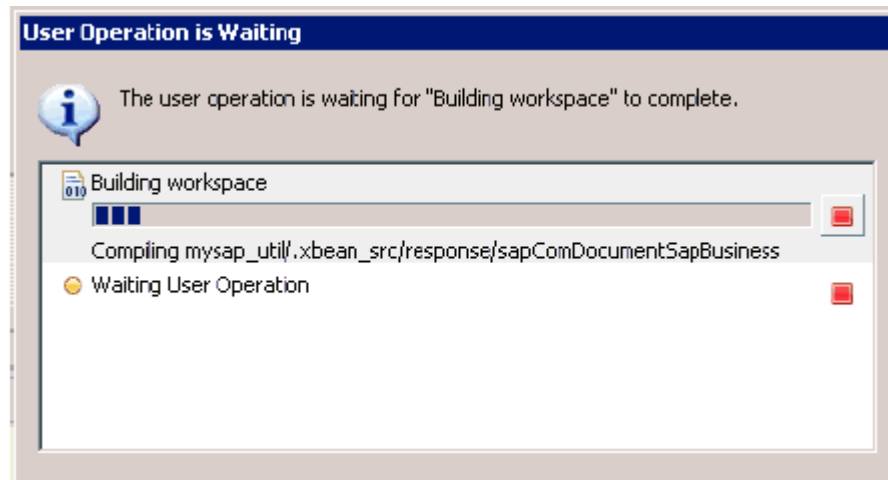
The copied outbound WSDL file is listed under the Process.java node, as shown in the following image.



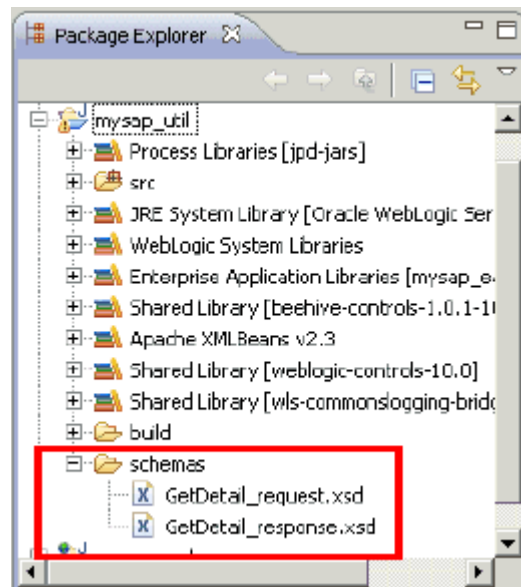
13. Expand the Utility project node (for example, **mysap_util**), and select the schemas folder.
14. Copy and paste the generated request and response XML schemas that were created using Application Explorer in the schemas folder.



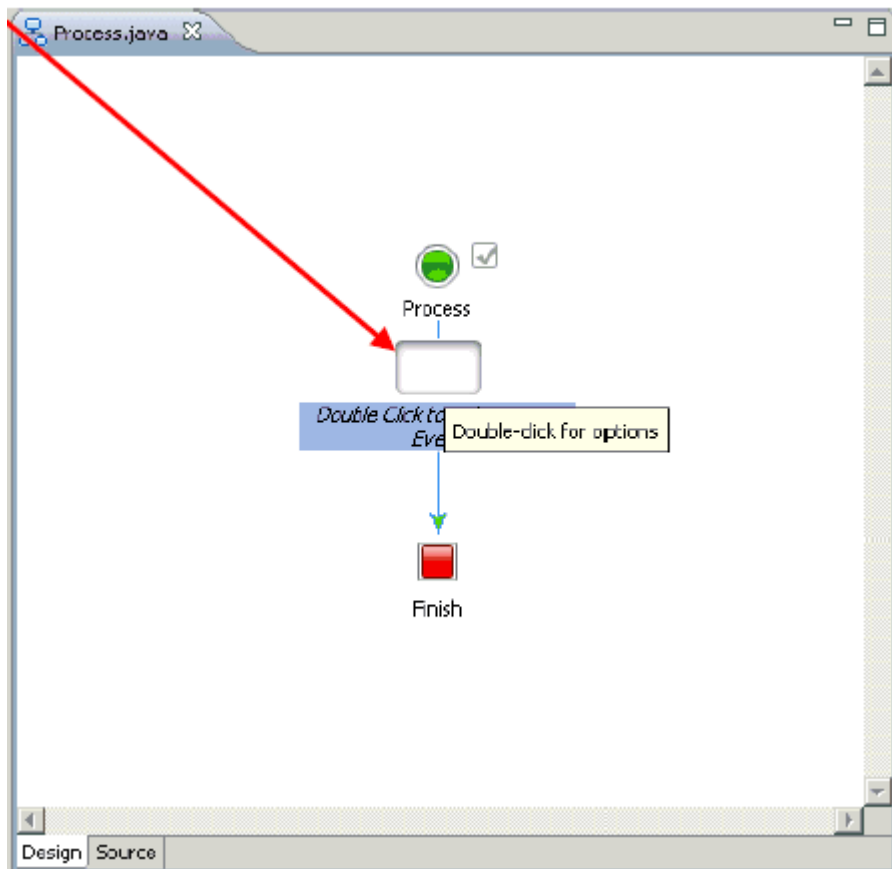
The new workspace is created by a background process, as shown by the following status indicator.



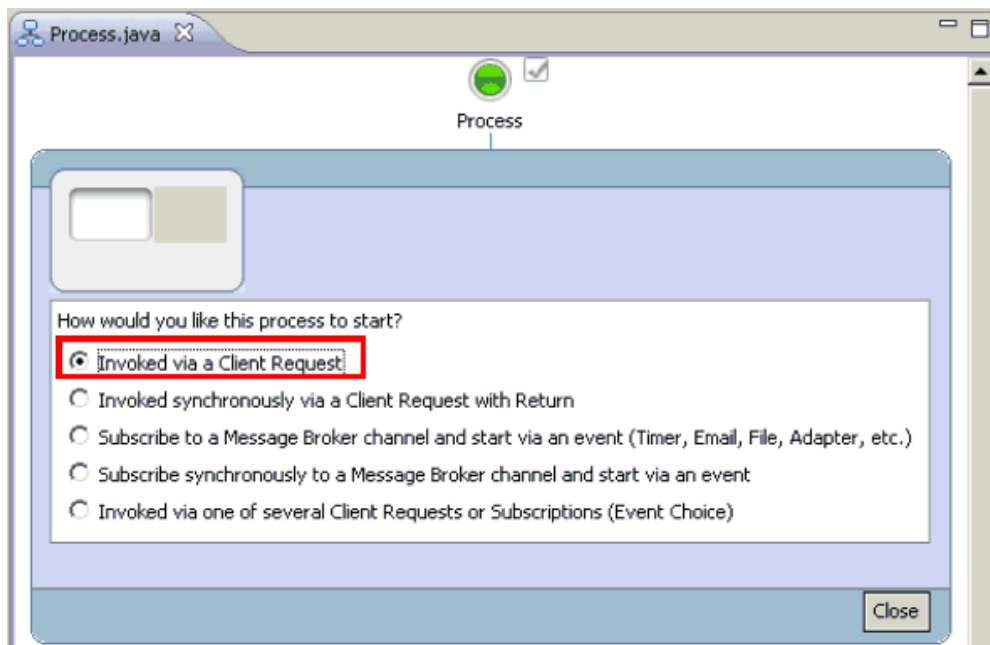
After this process is complete, the copied schemas are listed under the schemas folder.



15. Double-click the main node in the process design view (Process.java), as shown in the following image.

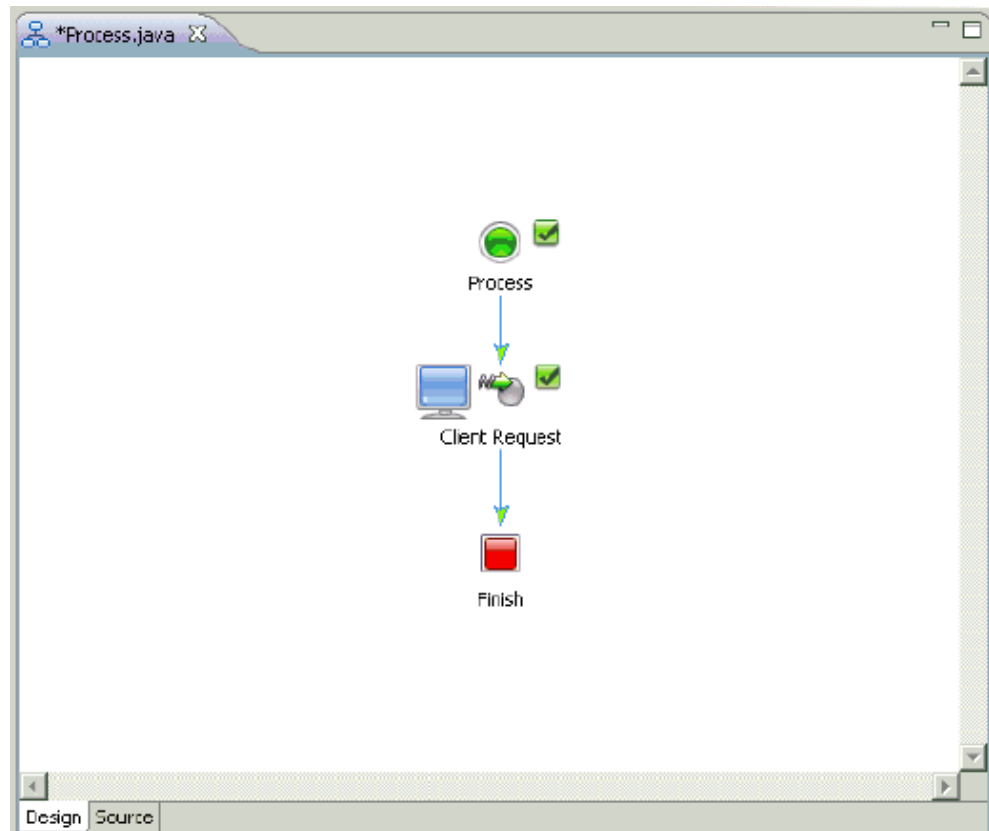


The following dialog box is displayed.

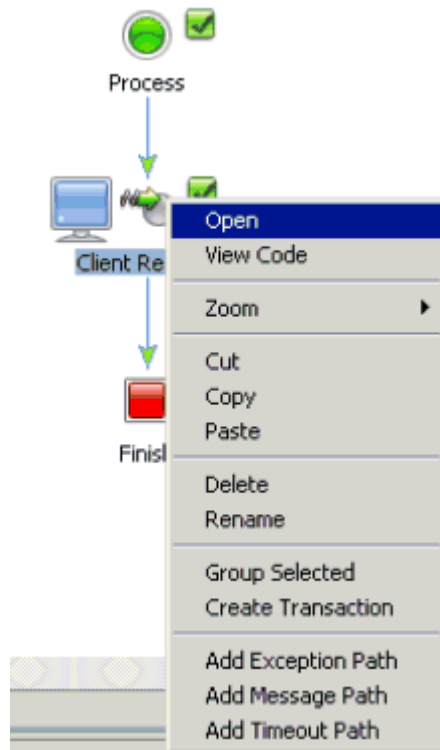


16. Select Invoked via a Client Request and click Close.

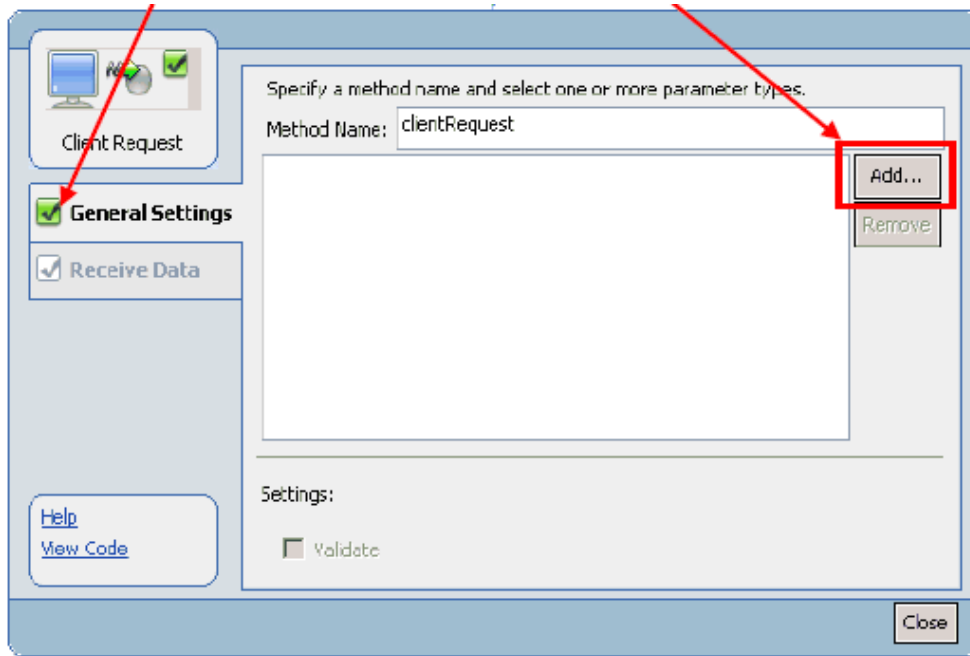
The Client Request node is added to the process design view (Process.java).



17. Right-click the Client Request node and select Open from the context menu.

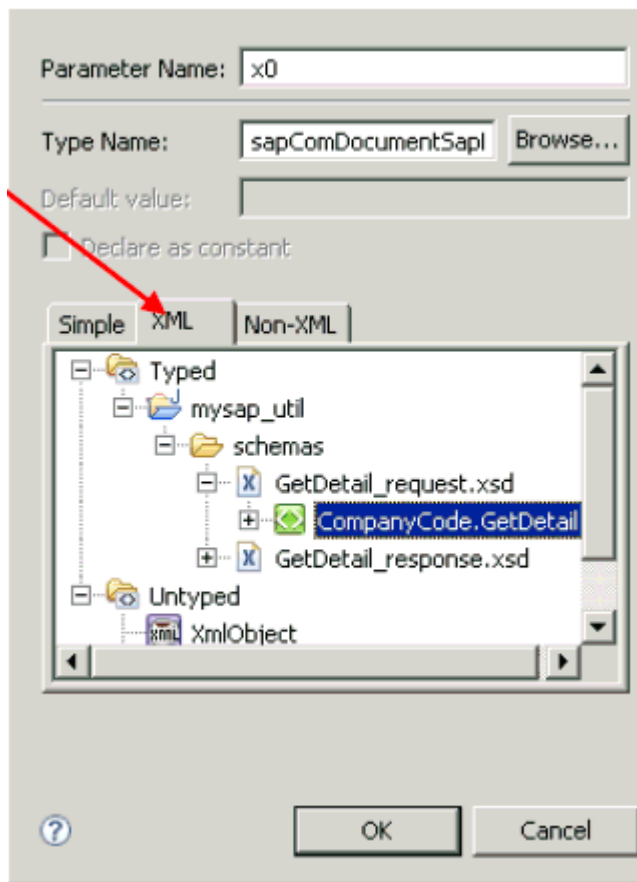


The Client Request configuration properties dialog box is displayed.



18. Click General Settings in the left pane and then click Add.

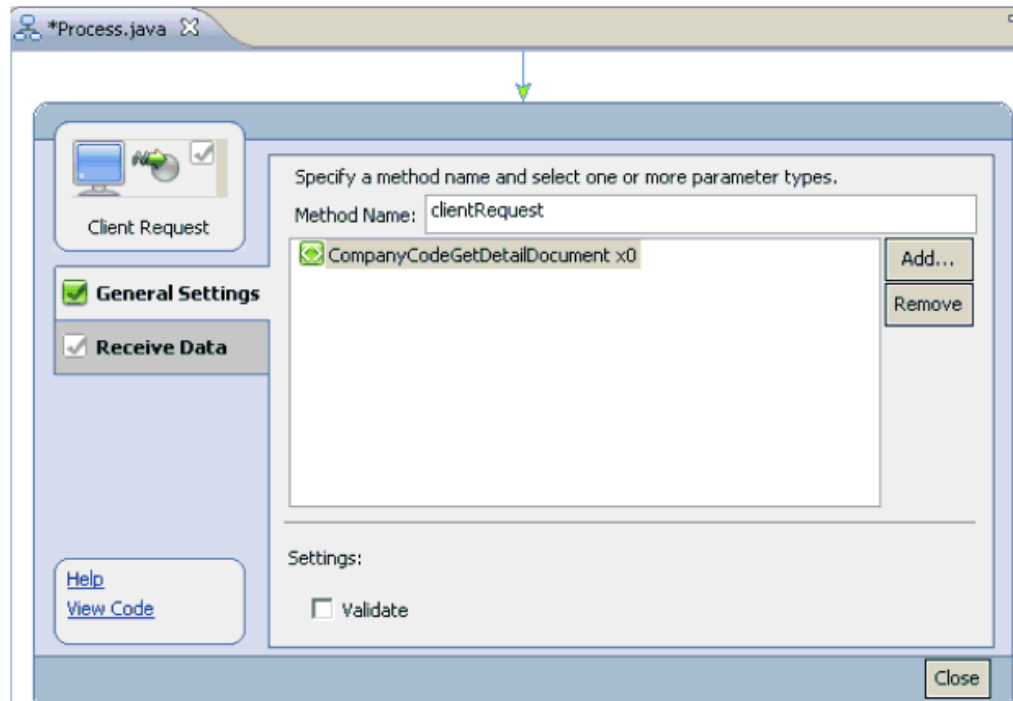
The following dialog box is displayed.



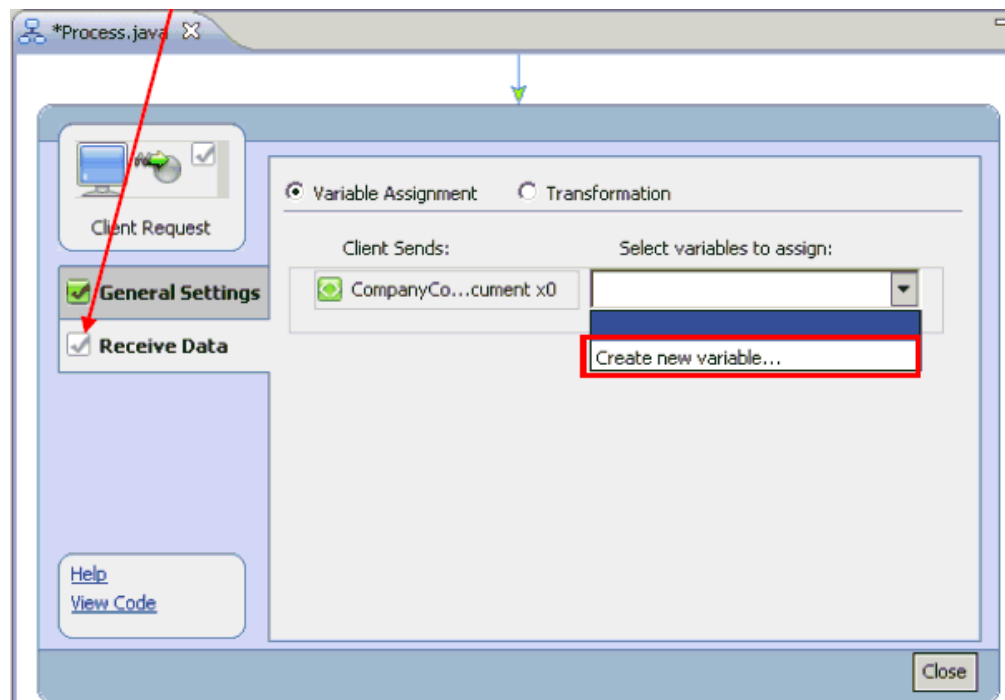
19. Under the XML tab, expand Typed, followed by mysap_util, schemas, GetDetail_request.xsd, and select CompanyCode.GetDetail.

20. Click OK.

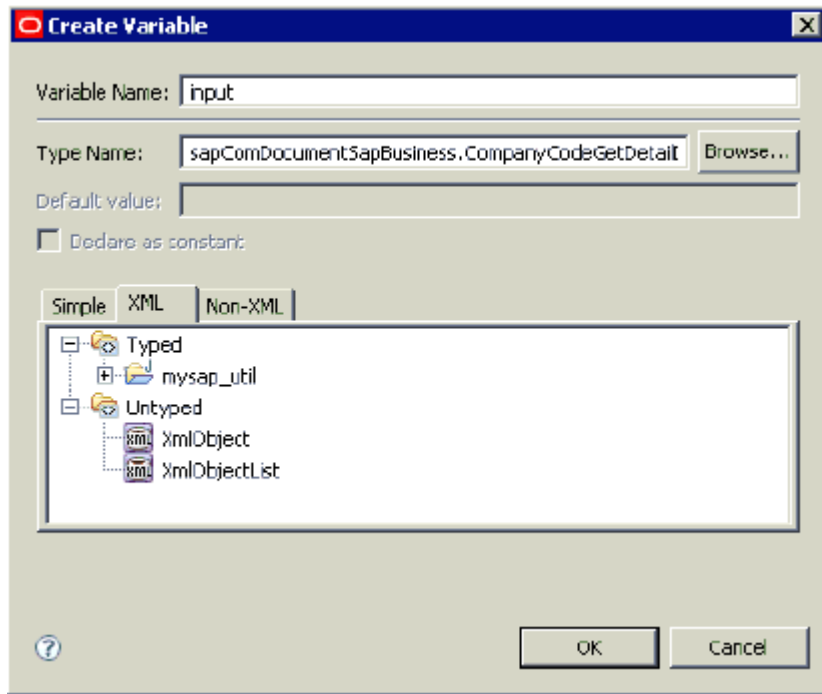
You are returned to the Client Request configuration properties dialog box.



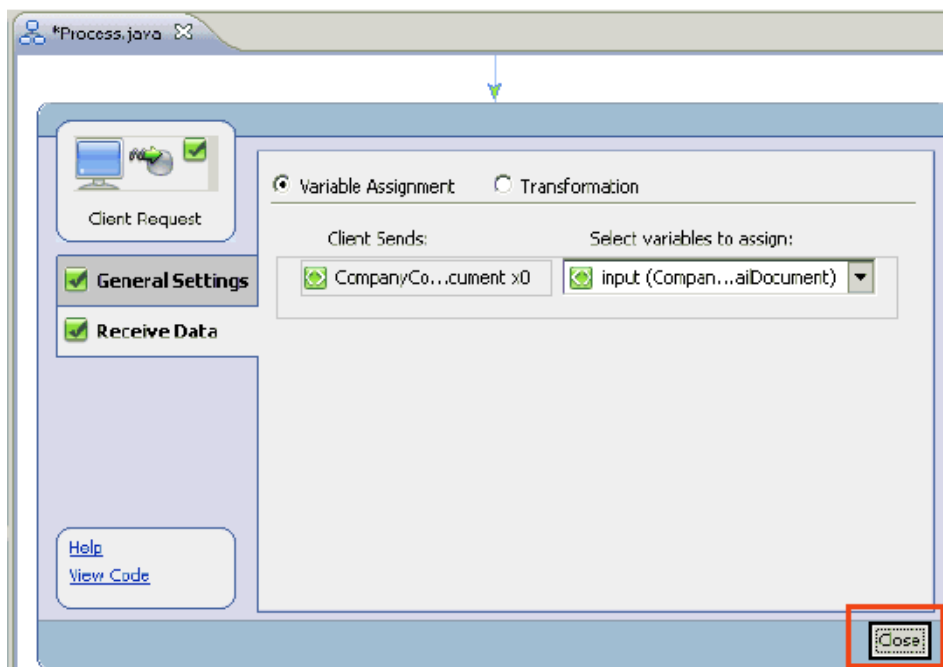
21. Click Receive Data in the left pane and select Create new variable from the drop-down list.



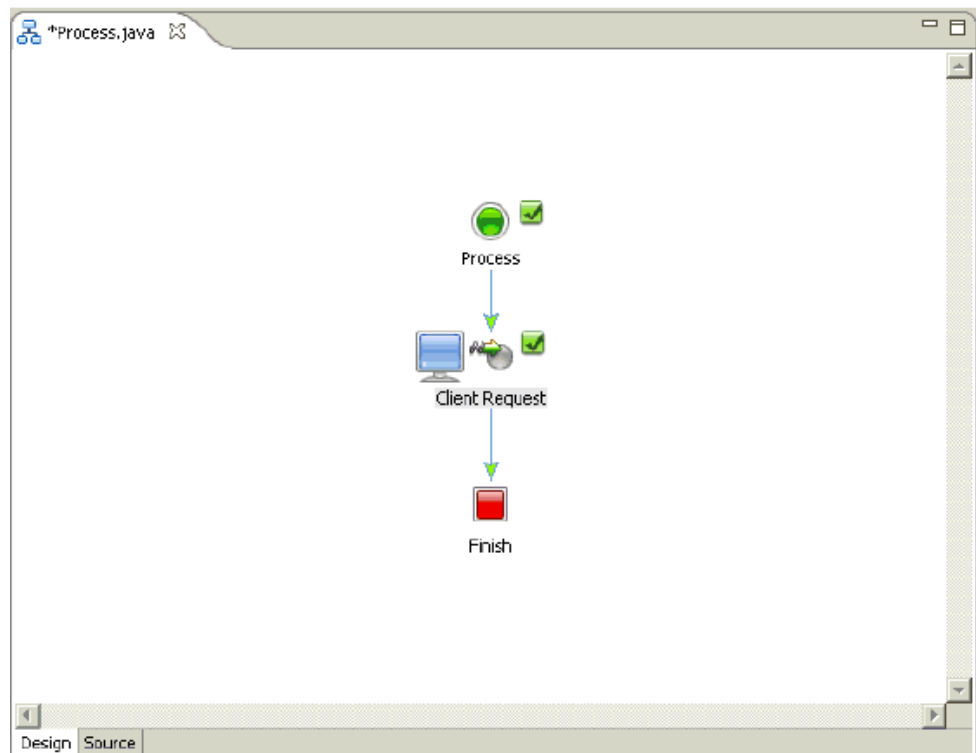
The Create Variable dialog box is displayed.



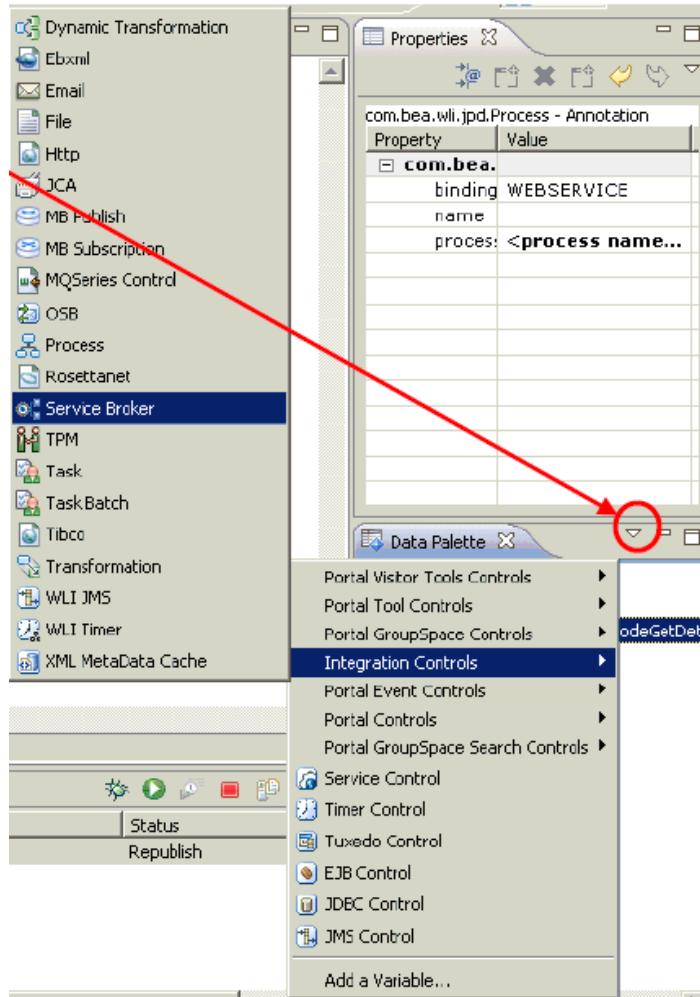
22. Specify a new variable name for the input data for example, input, and click OK.
 You are returned to the Client Request configuration properties dialog box, which has been refreshed with the new information.



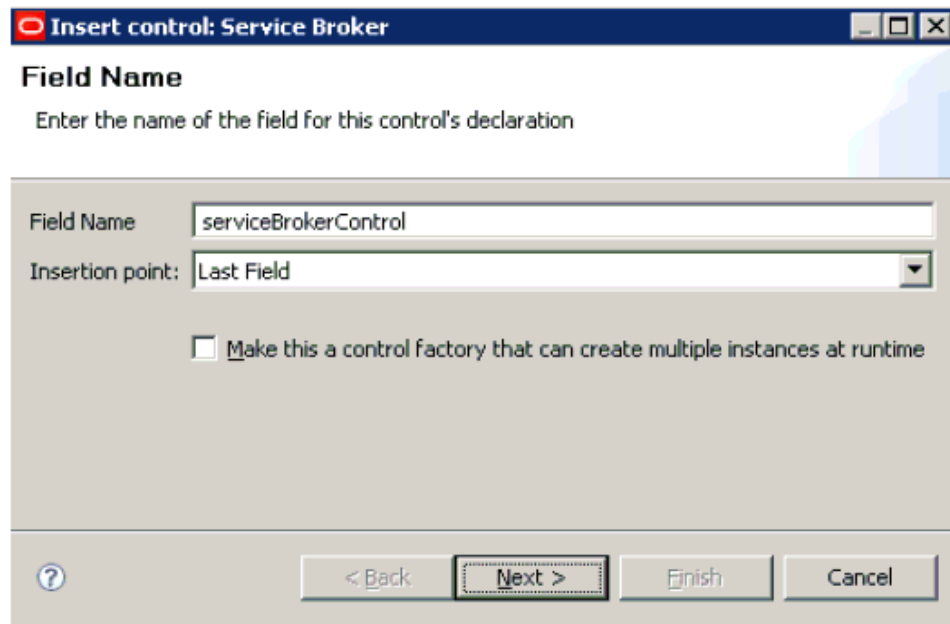
23. Click Close.
 You are returned to the process design view (Process.java).



24. Click the down arrow in the Data Palette area, select Integration Controls, and select Service Broker.

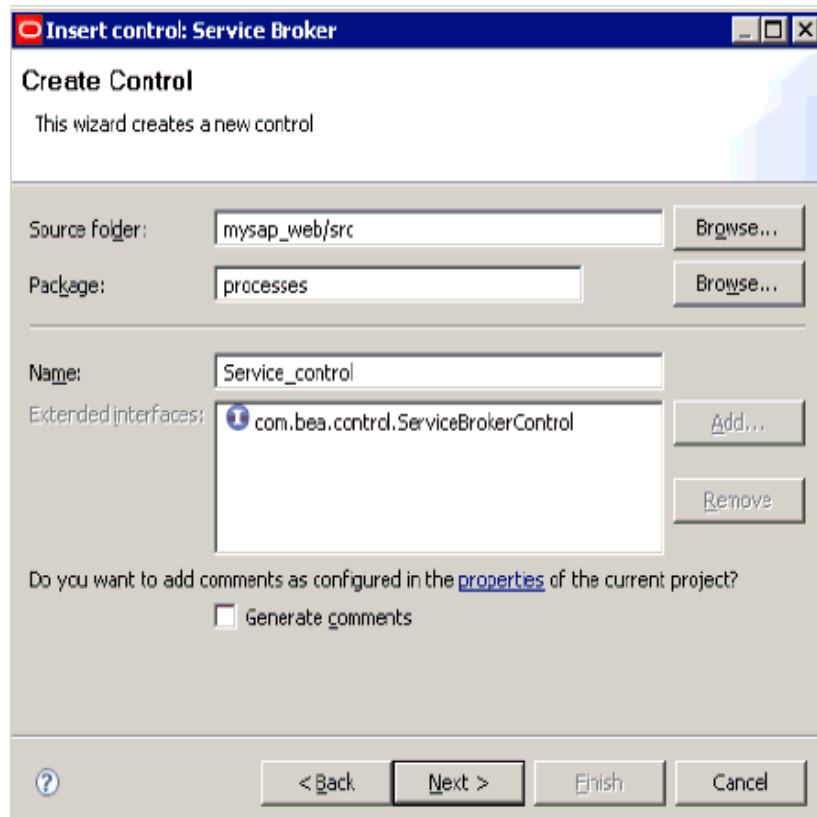


The Insert Control: Service Broker dialog box is displayed.



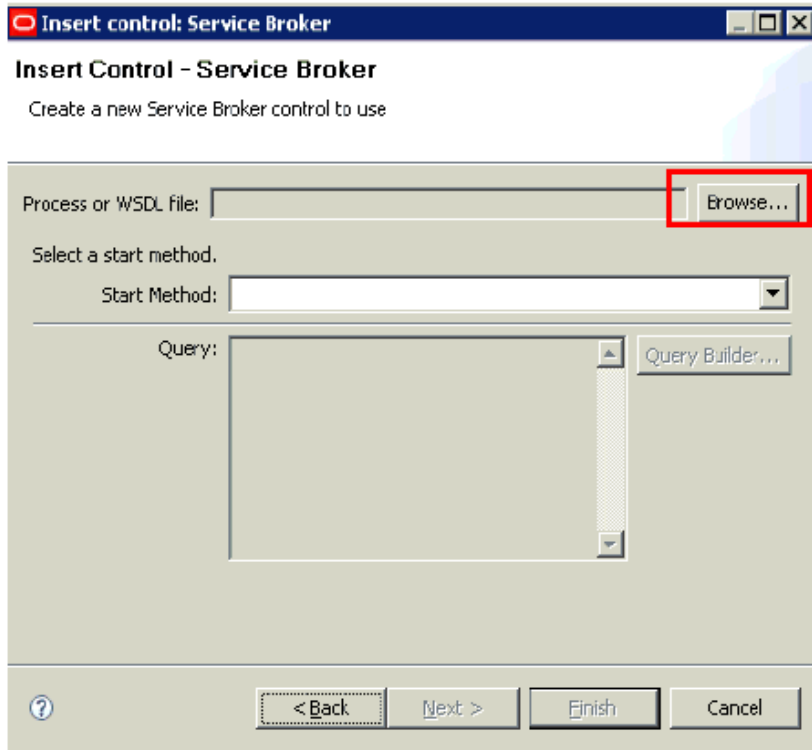
25. Specify the field name (for example, serviceBrokerControl) and click Next.

The Create Control pane is displayed.

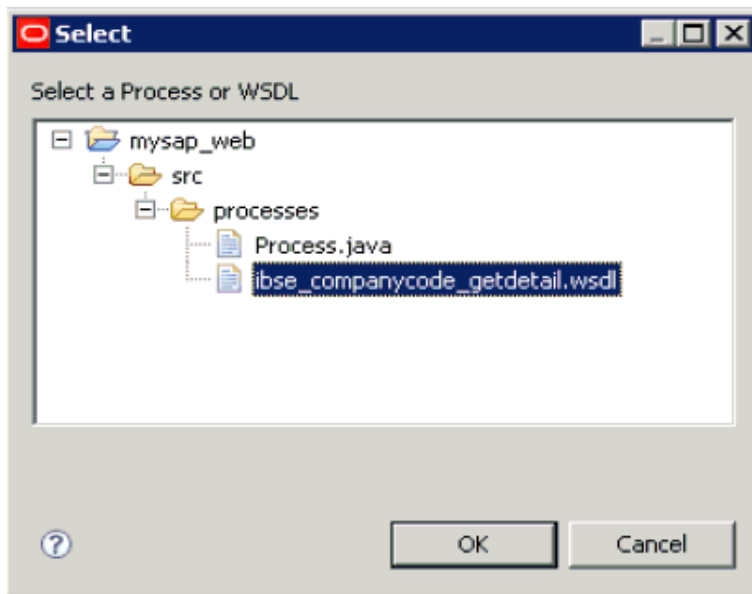


26. In the Name field, specify a name for the new Service Broker control (for example, Service_control) and click Next.

The Insert Control : Service Broker pane is displayed.

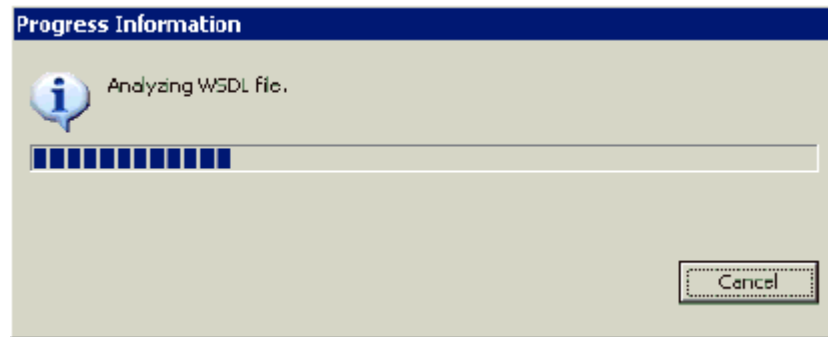


27. Click the Browse button to the right of the WSDL field.
The Select a Process or WSDL dialog box is displayed.

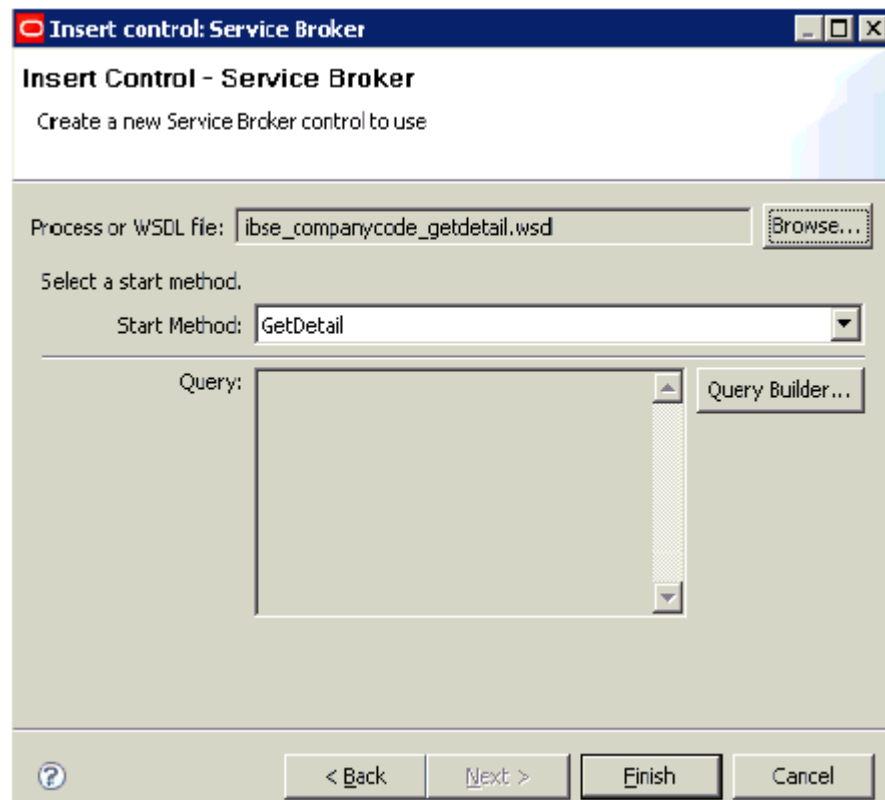


28. Expand mysap_web, followed by src and processes.
29. Select `ibse_companycode_getdetail.wsdl` and click OK.

The WSDL file is analyzed by a background process, as shown by the following status indicator.



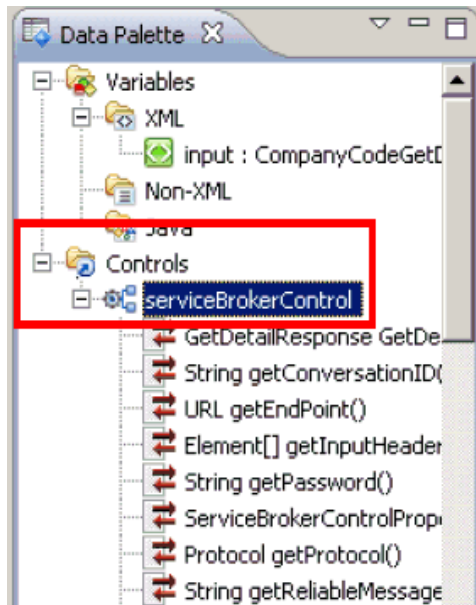
After this process is complete, you are returned to the Insert Control: Service Broker dialog box.



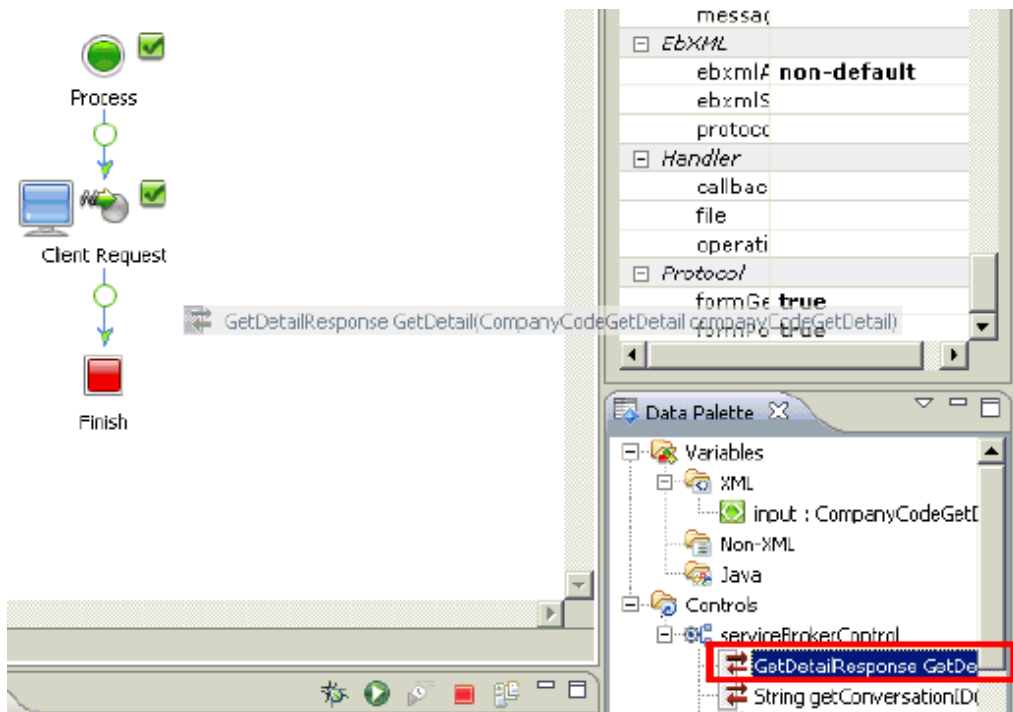
30. Click Finish.

The required operation is performed by a background process.

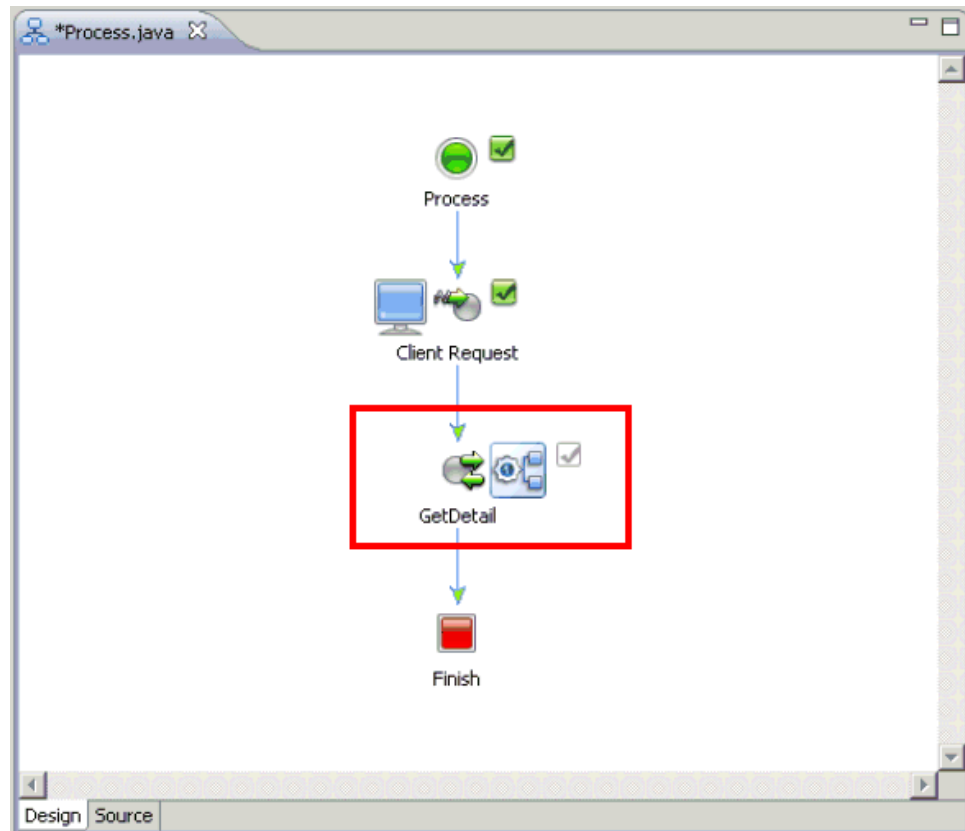
After this process is complete, the new Service Broker control (serviceBrokerControl) is added in the Data Palette area under the Controls folder.



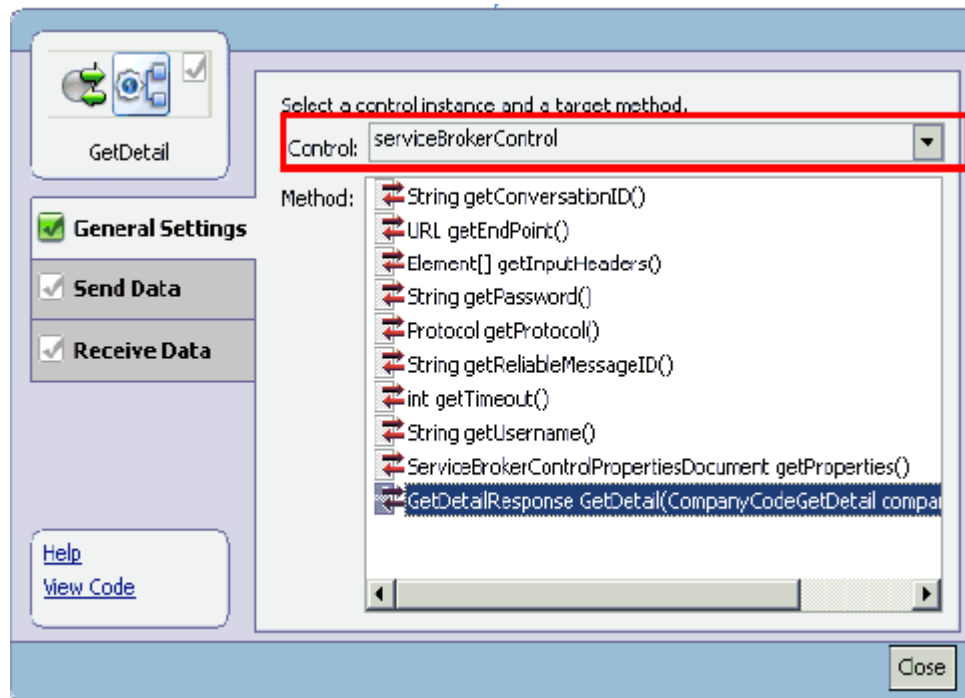
31. Click and drag GetDetailResponse from the Data Palette area to the circle between the Client Request node and Finish node in the process design view (Process.java).



32. Right-click the getDetail node in the process design view (Process.java) and select Open from the context menu.

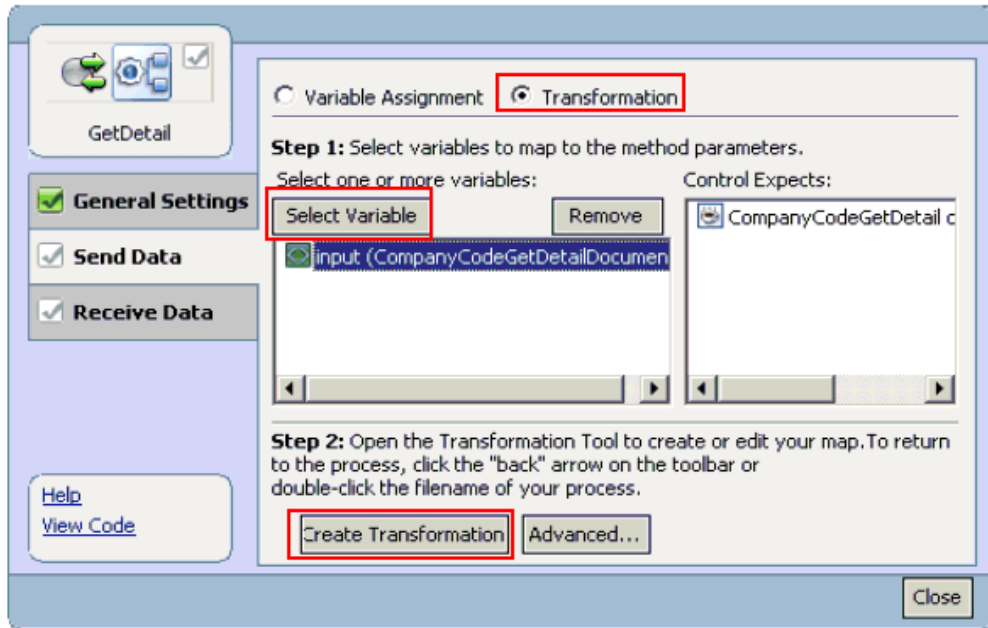


The getDetail configuration properties dialog box is displayed.

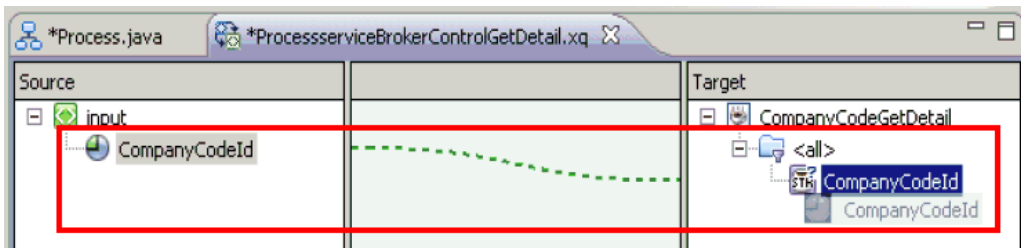


33. Click General Settings in the left pane.
34. Select serviceBrokerControl from the Control drop-down list.

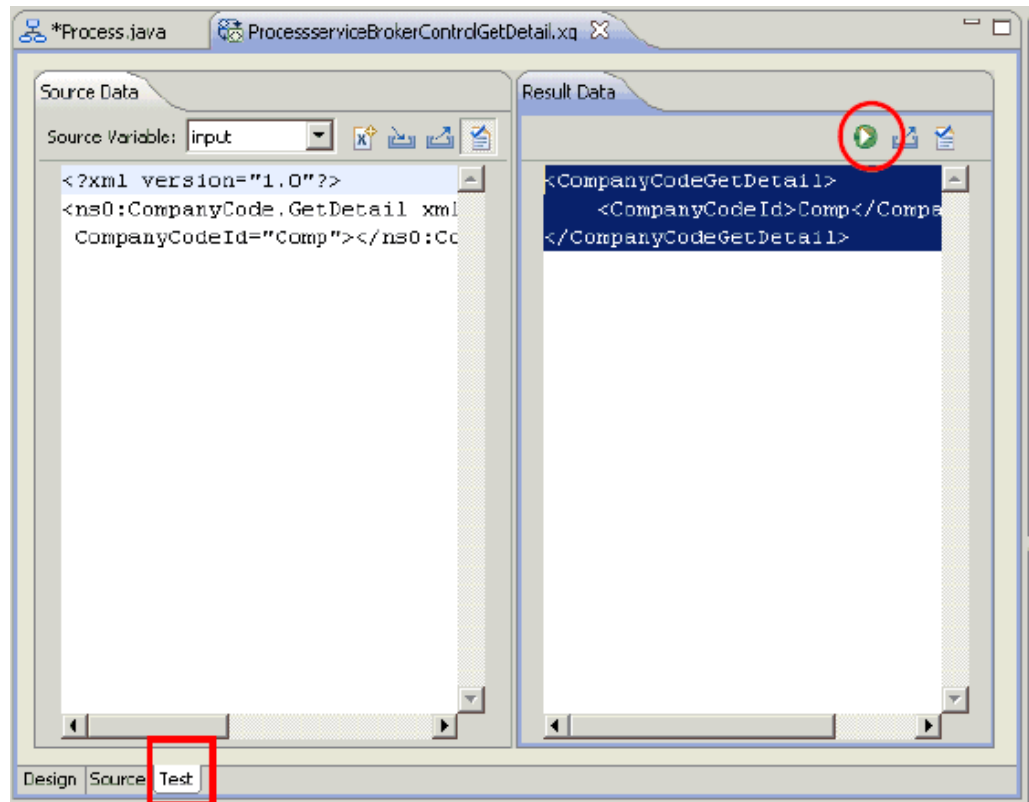
35. Select GetDetailResponse from the Method list.
36. Click Send Data in the left pane.



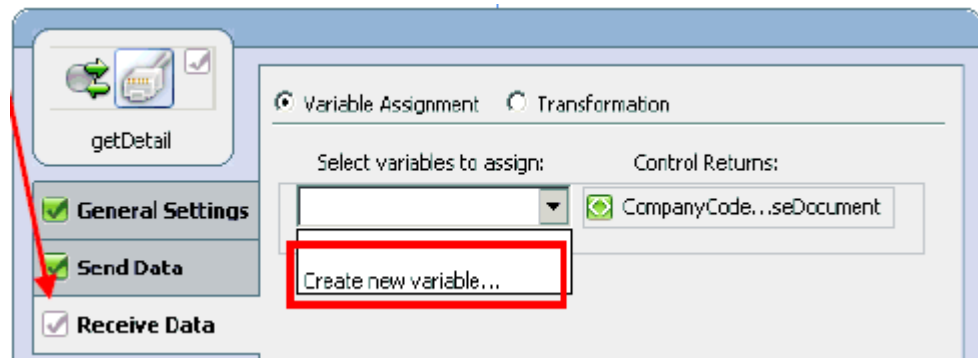
37. Select the **Transformation** option.
38. Click **Select Variable** and select the available input variable.
39. Click **Create Transformation**.



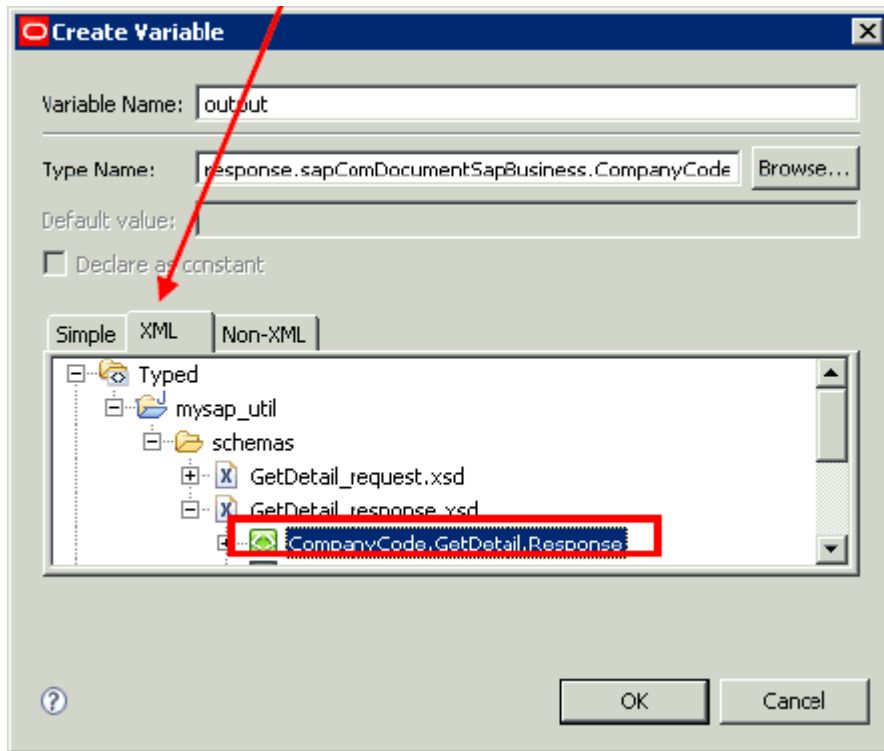
40. Map the **CompanyCodeId** field from the Source column to the Target column.



41. Click the Test tab and then click TestRun to check the mapping.
42. From the getDetail configuration properties dialog box, click Receive Data in the left pane.

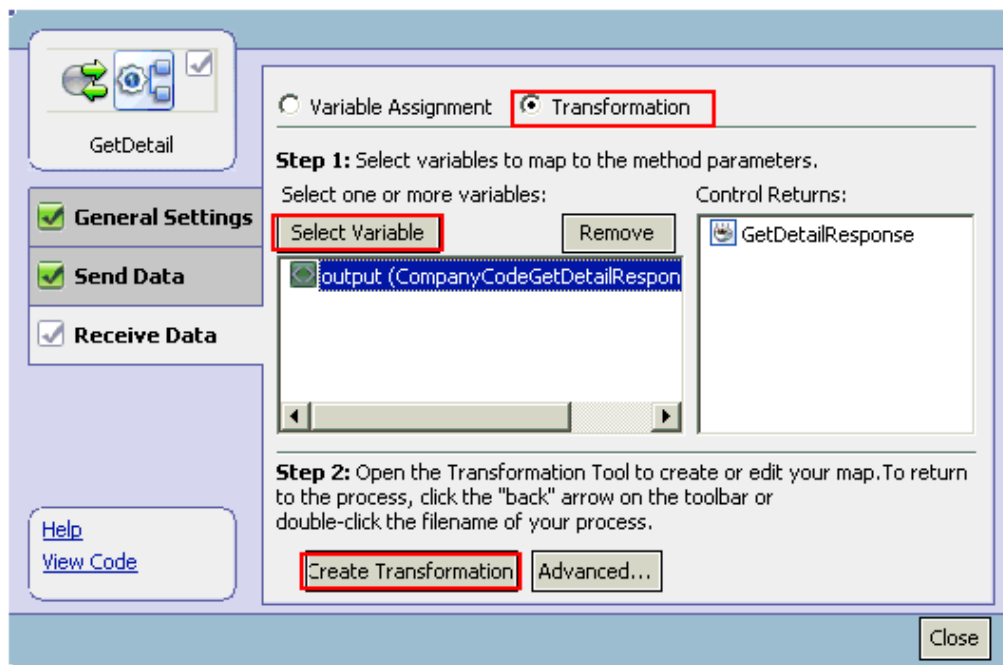


43. From the Select variables to assign drop-down list, select Create new variable. The Create Variable dialog box is displayed.



44. In the Variable Name field, enter a new variable name for the output data (for example, output).
45. In the XML tab, expand Typed, mysap_util, schemas, GetDetail_response.xsd, and select Companycode.GetDetail.Response.
46. Click OK.

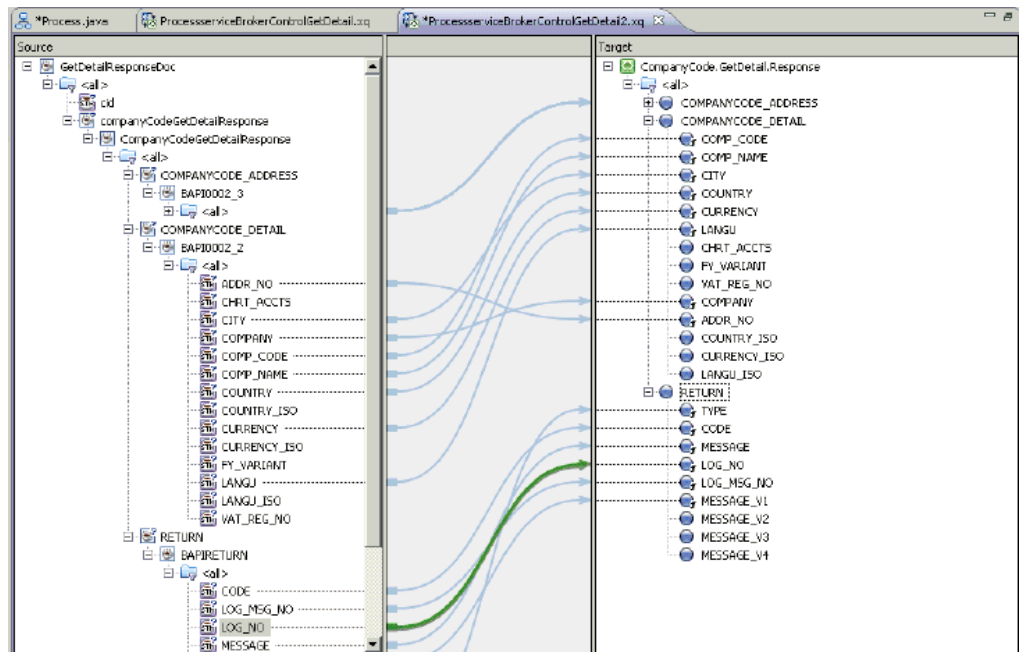
You are returned to the getDetail configuration properties dialog box.



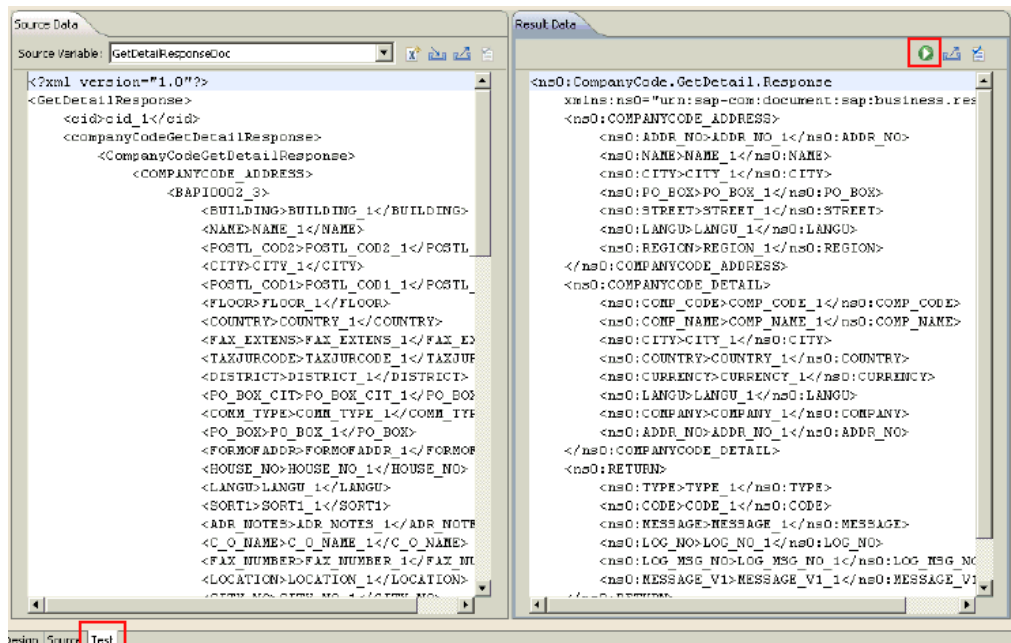
47. Select the **Transformation** option.

48. Click **Select Variable** and select the available input variable.

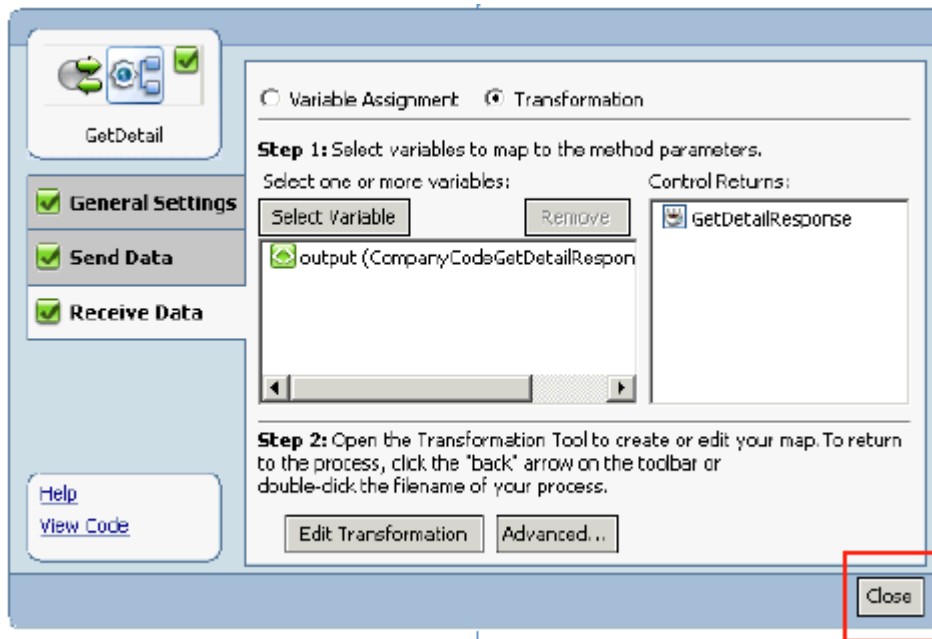
49. Click **Create Transformation**.



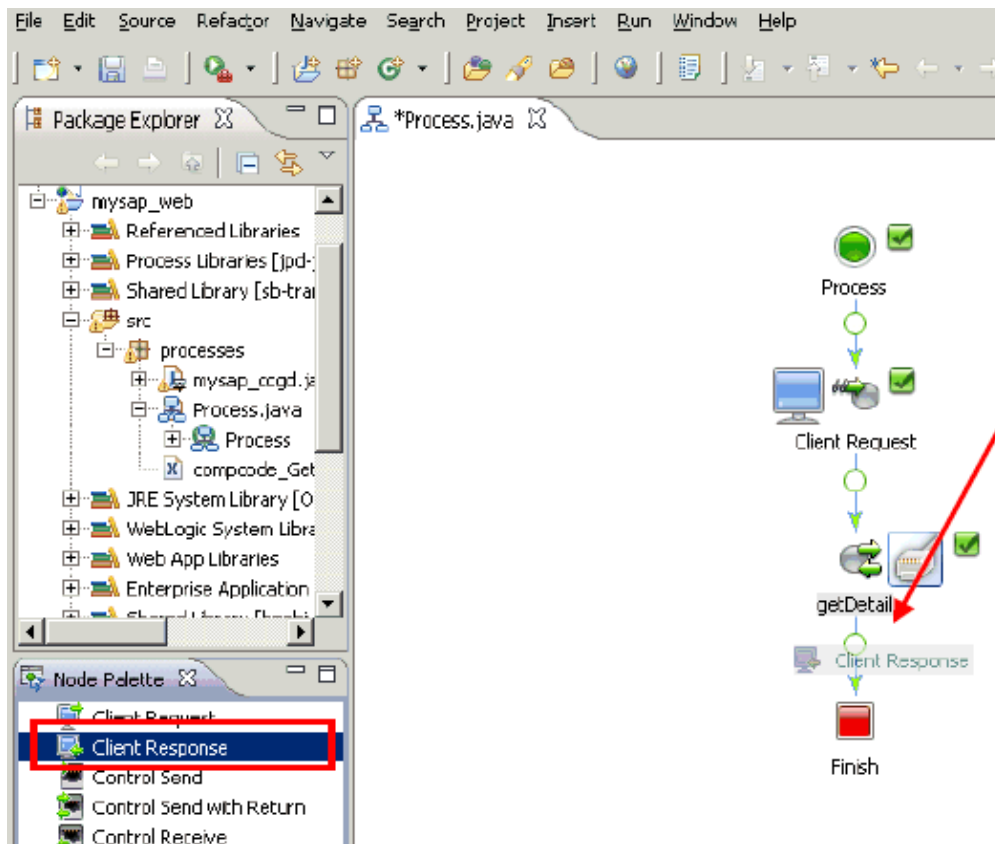
50. Map the fields under **COMPANYCODE_ADDRESS**, **COMPANYCODE_DETAIL**, and **RETURN** from the Source column to the Target column.



51. Click the **Test** tab and then click **TestRun** to check the mapping.

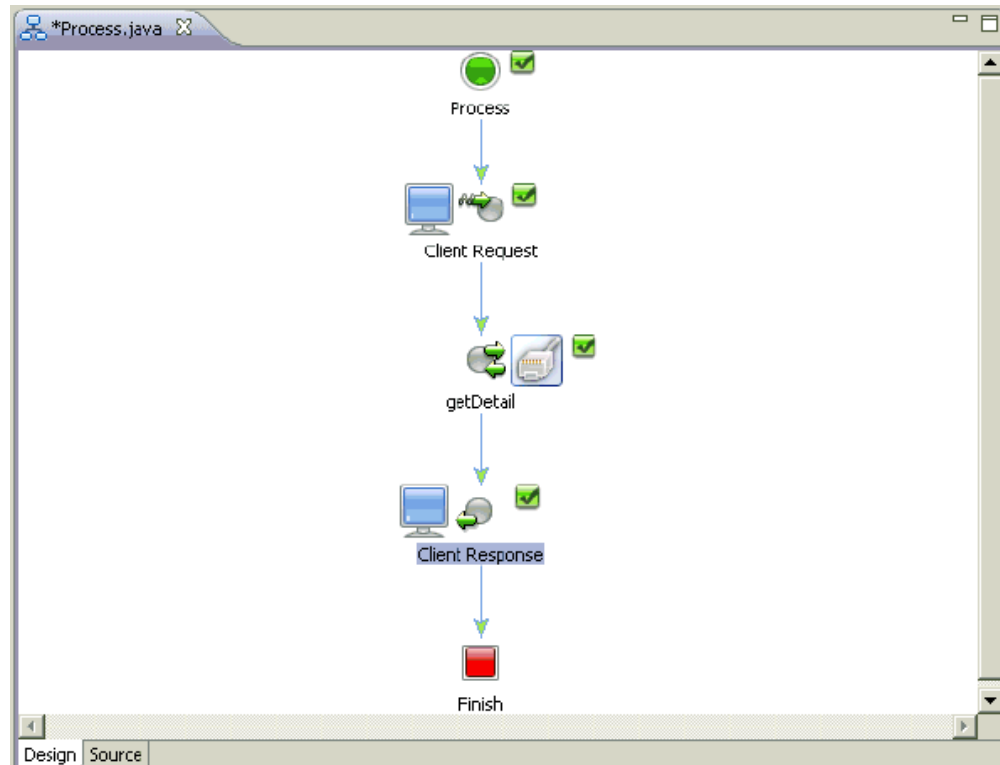


52. From the getDetail configuration properties dialog box, click Close.
You are returned to the process design view (Process.java).

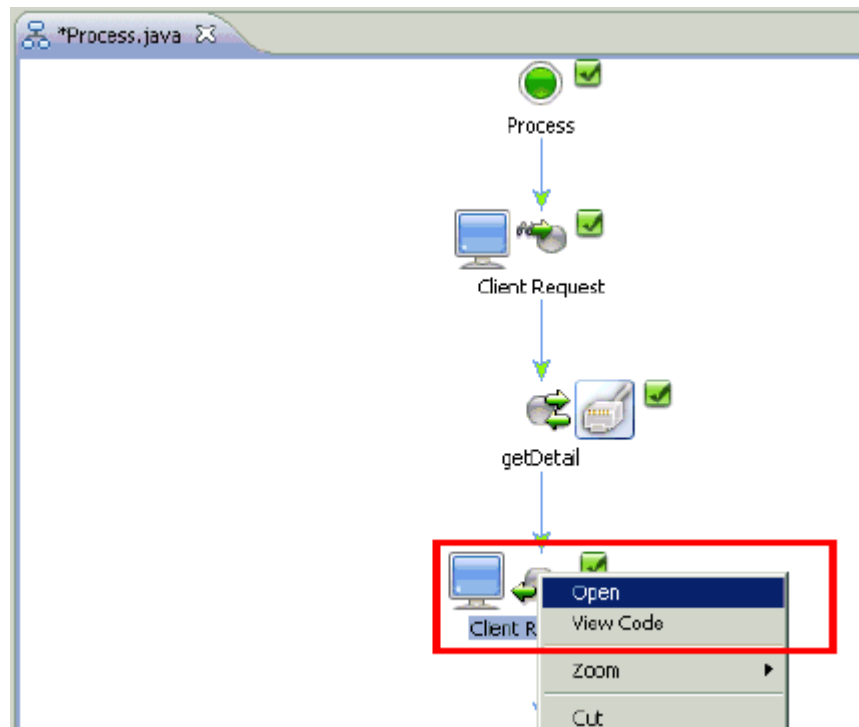


53. From the Node Palette area in the left pane, click the Client Response node and drag it to the circle between the getDetail node and the Finish node.

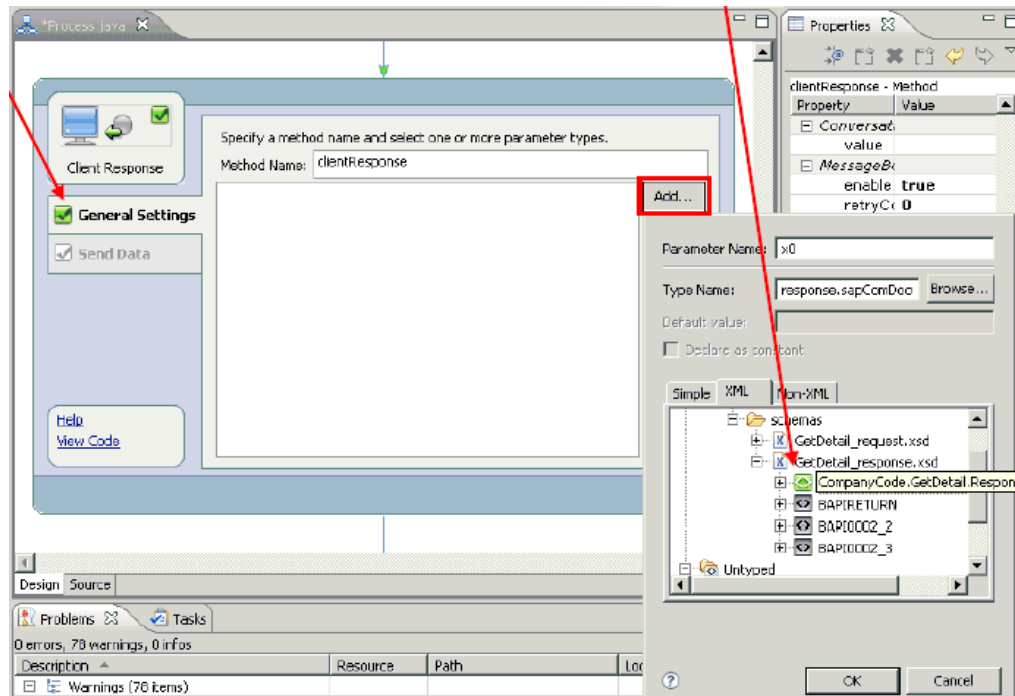
The process design view (Process.java) is updated to reflect the new addition of the Client Response node.



54. Right-click the Client Response node and select Open from the context menu.



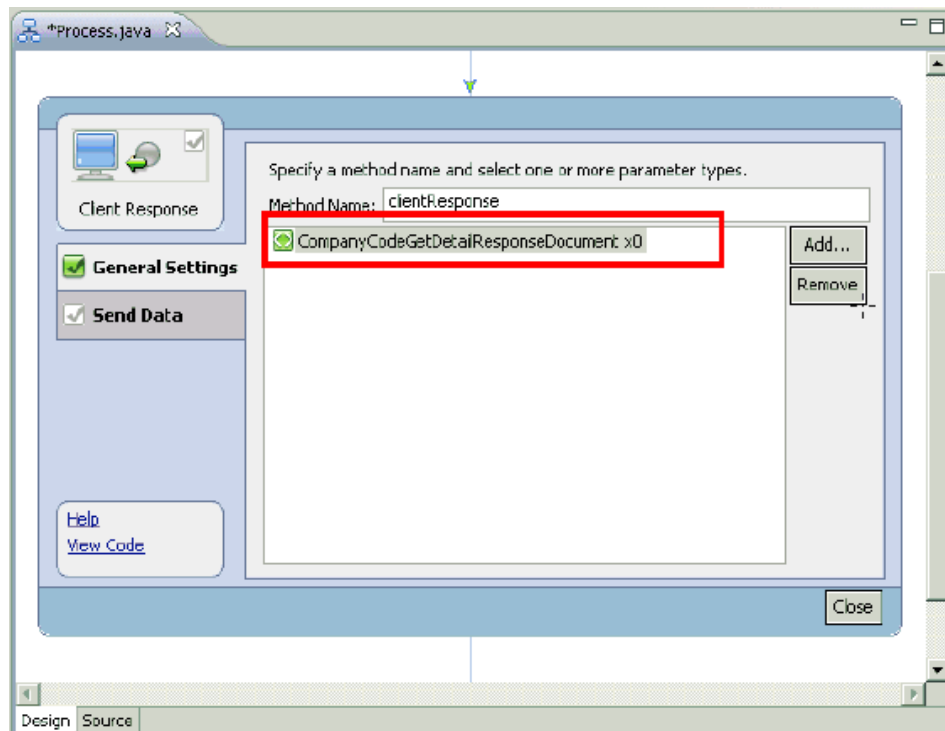
The Client Response configuration properties dialog box is displayed.



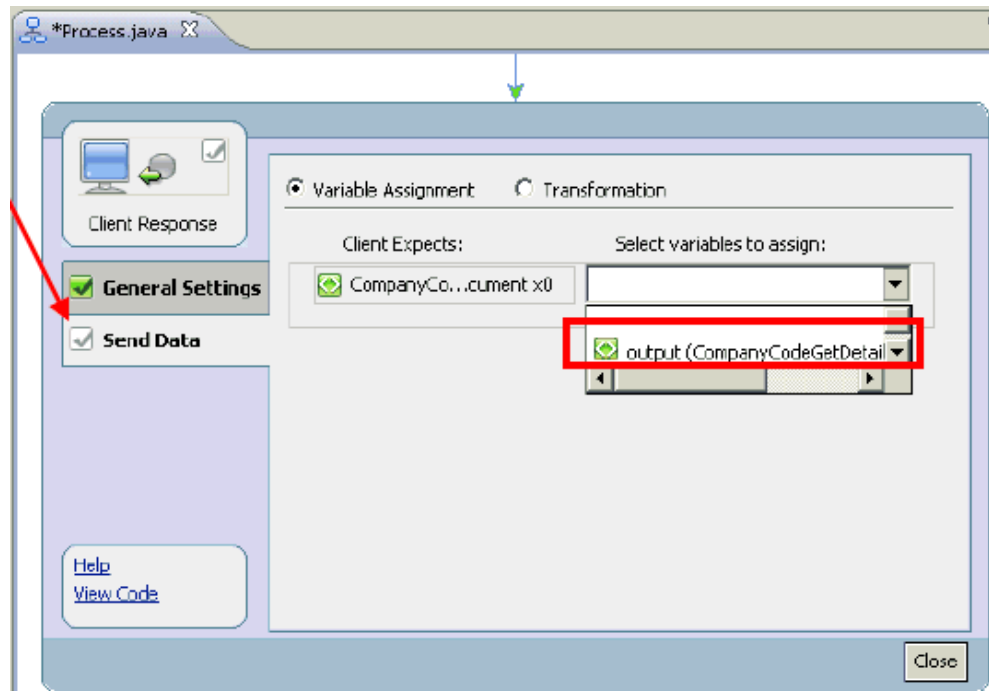
55. Click Add in the General Settings area.

56. In the XML tab, select the XML response schema (Companycode.GetDetail.Response) and click OK.

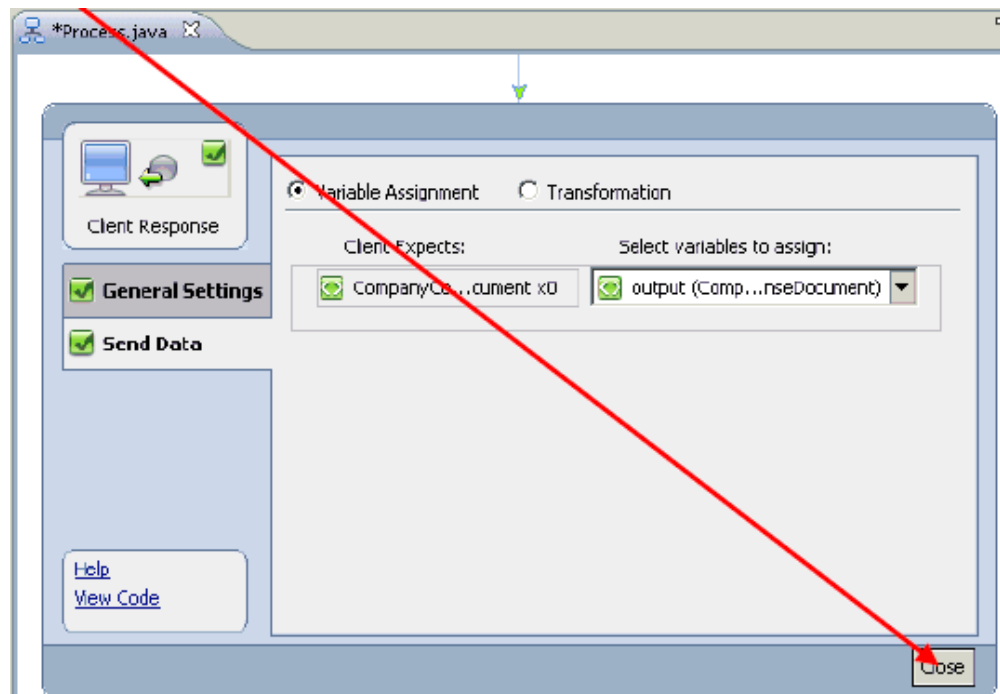
The response XML schema is added to the General Settings area.



57. Click Send Data in the left pane.

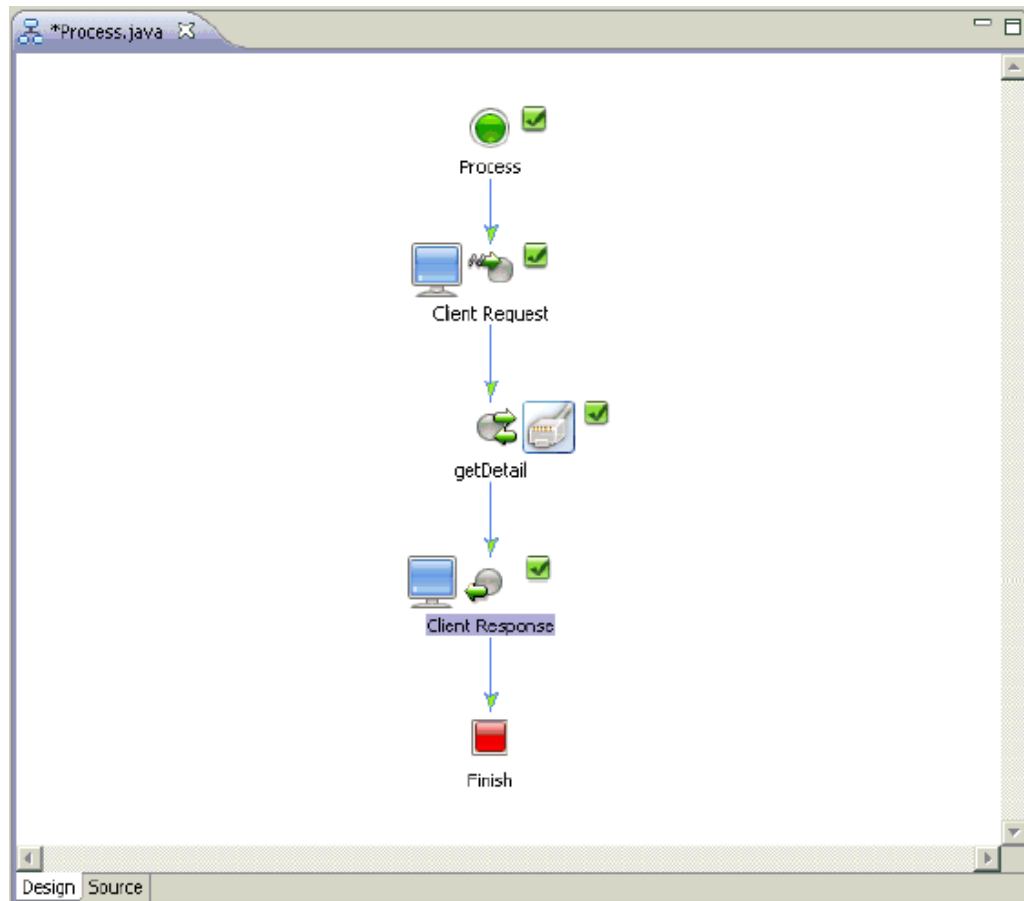


58. From the Select variables to assign drop-down list, select the defined output variable that is available.



59. Click Close.

You are returned to the process design view (Process.java), which is now finished.



60. Save the outbound process (Process.java).

You are now ready to export the outbound JPD process application as an EAR file.

Exporting the Outbound JPD Process Application as an EAR File

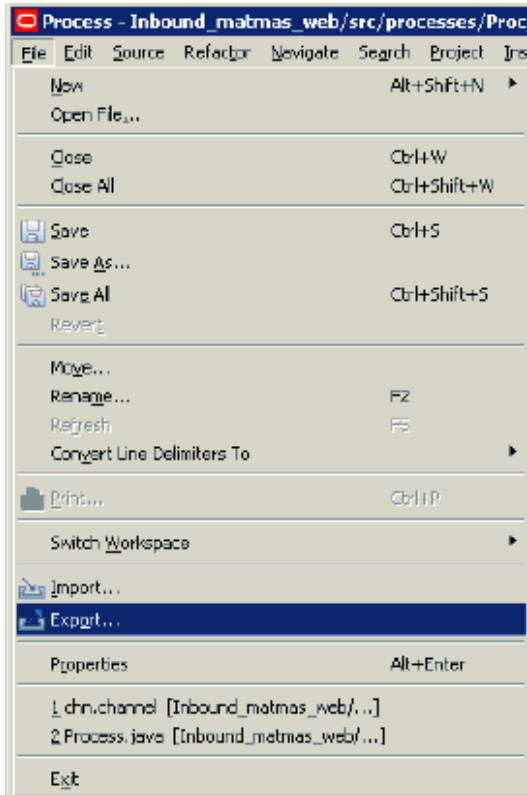
This section describes how to export the outbound JPD process application as an EAR file.

Note: Depending on your platform (for example, Windows or Solaris), there are a few differences in the outbound deployment process.

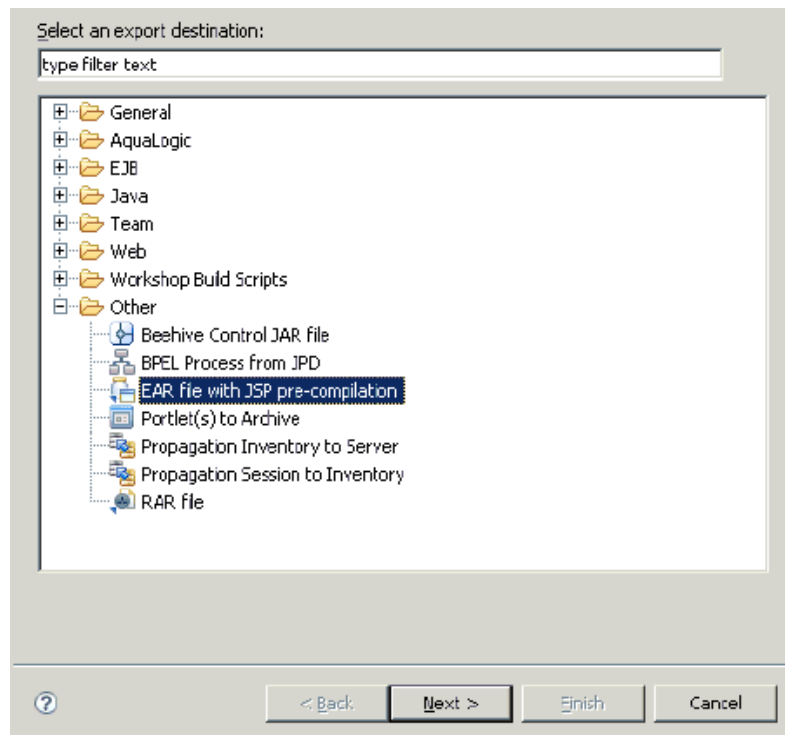
On Windows platforms, you can deploy the outbound process directly to the WebLogic Server. For more information, see [Deploying the Outbound Process on WebLogic Server](#).

On Solaris platforms, you must first export the outbound process as an EAR file, deploy the outbound process on WebLogic Server, and then create a queue to receive the response document in the Test Console. If you are using Solaris, follow the corresponding procedures that are described in this chapter.

1. From the menu bar, click File and select Export.

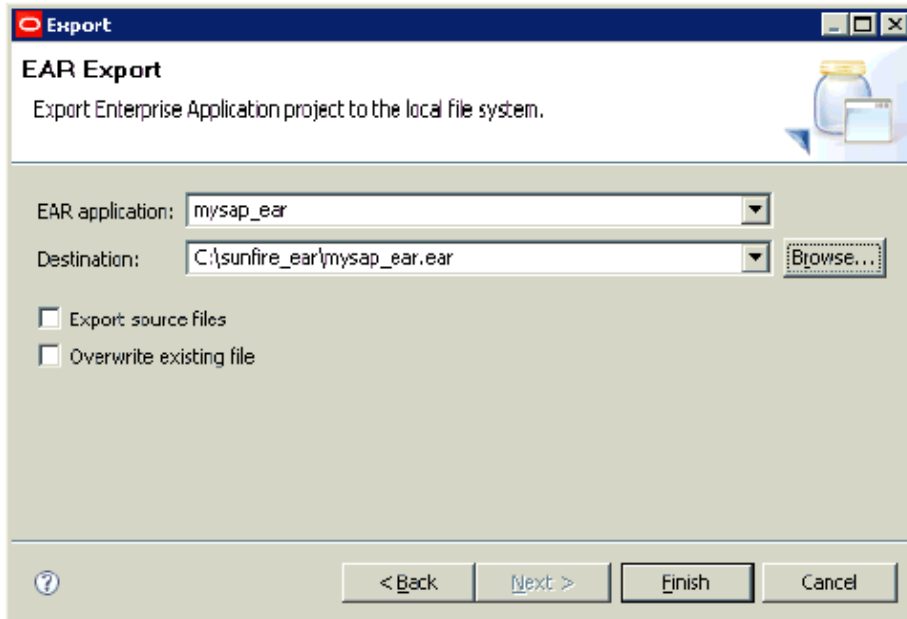


The Export dialog box is displayed.

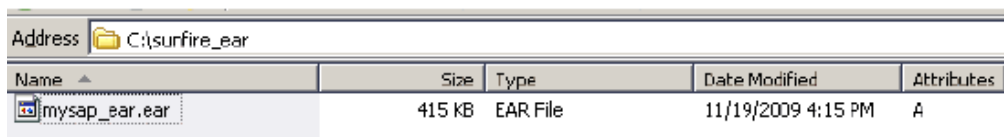


2. Select EAR file with JSP pre-compilation and click Next.

The EAR Export dialog box is displayed.



3. Select the EAR application from the drop-down list and specify a destination where the EAR file should be exported.
4. Click Finish.
5. Ensure that the EAR file has been exported properly.



You are now ready to deploy the outbound JPD process application EAR file on WebLogic Server.

Deploying the Outbound JPD Process Application EAR File on WebLogic Server

This section describes how to deploy the outbound process on WebLogic Server. You can also deploy applications using Oracle Workshop for WebLogic. For more information, see *Deploying Applications to WebLogic Server* in *Oracle Workshop for WebLogic User Guide*.

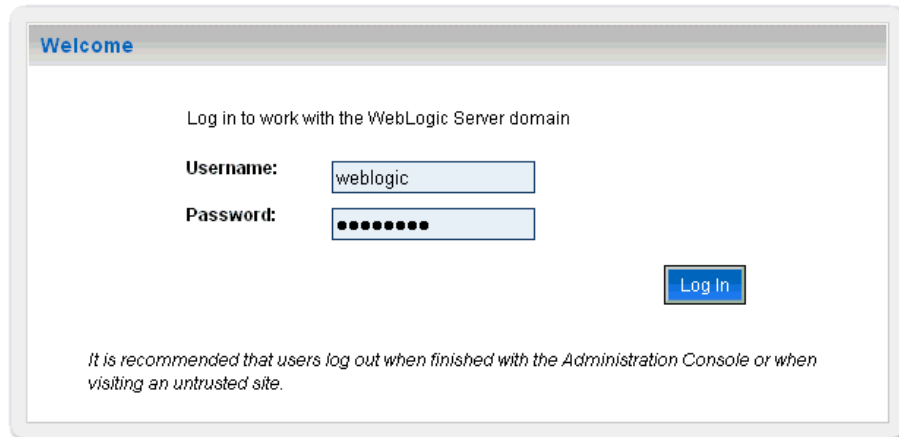
1. Start the Oracle WebLogic Server for the Oracle WebLogic Server domain that you have configured.
2. Open the Oracle WebLogic Server Administration Console in a Web browser by entering the following URL:

```
http://hostname:port/console
```

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

The Oracle WebLogic Server Administration Console logon page is displayed.

ORACLE® WebLogic Server® Administration Console



WebLogic Server Version: 10.3.0.0

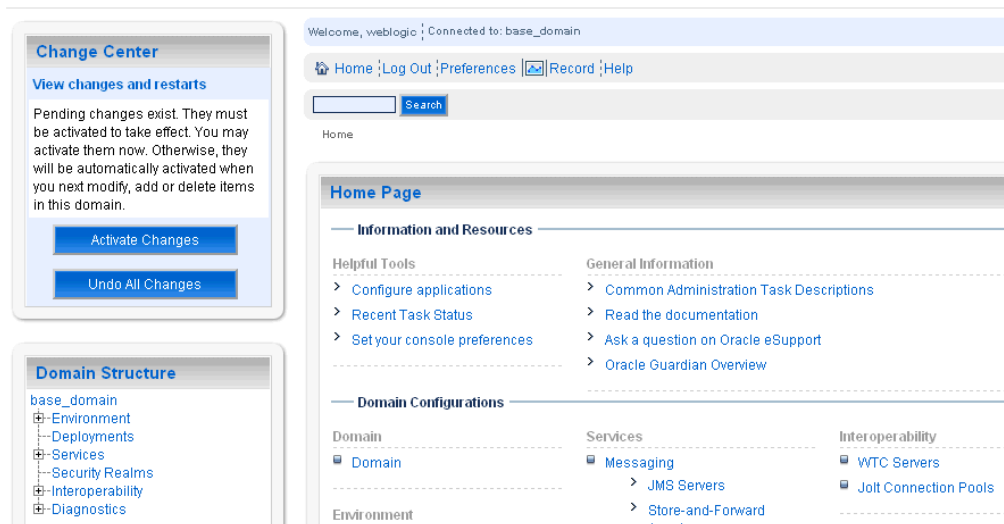
Copyright © 1996,2008, Oracle and/or its affiliates. All rights reserved.

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

3. Log on to the Oracle WebLogic Server Administrative Console using an account that has administrator privileges.

The Oracle WebLogic Server Administration Console home page is displayed.

ORACLE® WebLogic Server® Administration Console



4. In the Domain Structure section in the left pane, click **Deployments**

The Deployments page is displayed.

Deployments

Install Update Delete Start Stop Showing 1 to 10 of 84 Previous Next

| <input type="checkbox"/> | Name | State | Health | Type | Deployment Order |
|--------------------------|---|--------|--------|------------------------|------------------|
| <input type="checkbox"/> | ALDSP Transport Provider | Active | OK | Web Application | 161 |
| <input type="checkbox"/> | aldsp_transport-110n(3.0,3.0) | Active | | Library | 160 |
| <input type="checkbox"/> | ALSB Cluster Singleton Marker Application | Active | OK | Enterprise Application | 80 |

5. Click **Install**.

The Install Application Assistant page is displayed.

Install Application Assistant

Back Next Finish Cancel

Locate deployment to install and prepare for deployment

Select the file path that represents the application root directory, archive file, exploded archive directory, or application module descriptor that you want to install. You can also enter the path of the application directory or file in the Path field.

Note: Only valid file paths are displayed below. If you cannot find your deployment files, [upload your file\(s\)](#) and/or confirm that your application contains the required deployment descriptors.

Path: /rdbsms/ora102/Chennai_QA

Recently Used Paths: [rdbsms/ora102/Chennai_QA](#)
[rdbsms/ora102/wii1031/erp-adapters](#)

Current Location: 172.19.250.196 / rdbsms / ora102 / Chennai_QA

- WLI
- lock
- ear
- jde_repository
- qtp9.5
- siebel_XDR_files
- soa_schemas
- mysap_ear.ear

Back Next Finish Cancel

6. Browse to the folder that contains the exported EAR file (for example, mysap_ear.ear).

7. Select the radio button next to **mysap_ear.ear** and click **Next**.

The Choose Targeting Style page is displayed.

Install Application Assistant

Back Next Finish Cancel

Choose targeting style

Targets are the servers, clusters, and virtual hosts on which this deployment will run. There are several ways you can target an application.

Install this deployment as an application

The application and its components will be targeted to the same locations. This is the most common usage.

Install this deployment as a library

Application libraries are deployments that are available for other deployments to share. Libraries should be available on all of the targets running their referencing applications.

Back Next Finish Cancel

8. Leave the default **Install this deployment as an application** selected and click **Next**.
The **Optional Settings** page is displayed.

Install Application Assistant

Back Next Finish Cancel

Optional Settings

You can modify these settings or accept the defaults

General

What do you want to name this deployment?

Name:

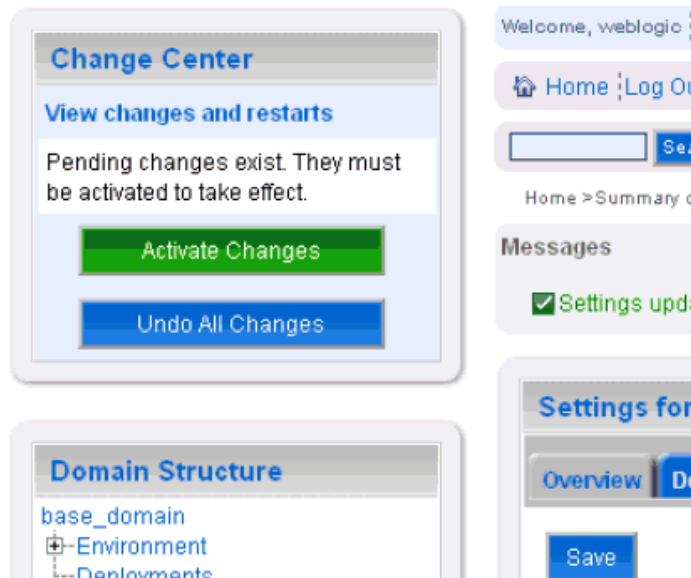
Security

What security model do you want to use with this application?

DD Only: Use only roles and policies that are defined in the deployment descriptors.

Custom Roles: Use roles that are defined in the Administration Console; use policies that are defined in the deployment descriptor.

9. Click **Next** again leaving the default values.
The **Summary** page is displayed.
10. Click **Finish**.
The **Settings** page for the application (`mysap_ear`) is displayed.



11. Click Save and then Activate Changes.
12. In the Domain Structure section in the left pane, click **Deployments**.

| | | | | | |
|--------------------------|-------------------------|----------|--|------------------------|-----|
| <input type="checkbox"/> | jsf(1.2,1.2.3.2) | Active | | Library | 1 |
| <input type="checkbox"/> | jsf-myfaces(1.1,1.1.1) | Active | | Library | 1 |
| <input type="checkbox"/> | jsf-myfaces(1.1,1.1.3) | Active | | Library | 1 |
| <input type="checkbox"/> | jsf-ri(1.1,1.1.1) | Active | | Library | 1 |
| <input type="checkbox"/> | jstl(1.1,1.1.2) | Active | | Library | 1 |
| <input type="checkbox"/> | jstl(1.2,1.2.0.1) | Active | | Library | 1 |
| <input type="checkbox"/> | JWSQueueTransport | Active | <input checked="" type="checkbox"/> OK | EJB | 100 |
| <input type="checkbox"/> | mqEG_ear | Active | <input checked="" type="checkbox"/> OK | Enterprise Application | 100 |
| <input type="checkbox"/> | mysap_dynamic_nov19_ear | Active | <input type="checkbox"/> Warning | Enterprise Application | 100 |
| <input type="checkbox"/> | mysap_ear | Prepared | <input checked="" type="checkbox"/> OK | Enterprise Application | 100 |

Showing 1 to 25 of 71 Previous | Next

13. Navigate through the table that lists all the deployed applications until you find the deployed EAR application (mysap_ear).
14. Select the check box next to **mysap_ear**.
15. Click the **Start** submenu (down arrow) and select **Servicing all requests**.
16. Ensure that the deployed EAR application (mysap_ear) is set to Active.
You are now ready to create a queue to receive the response document in the Test Console.

Creating a Queue to Receive the Response Document in the Test Console

This section describes how to create a queue to receive the response document in the Test Console.

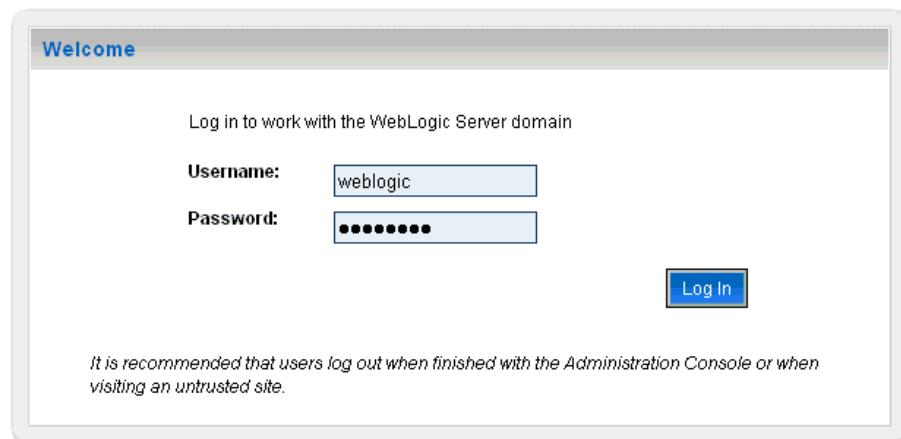
1. Start the Oracle WebLogic Server for the Oracle WebLogic Server domain that you have configured.
2. Open the Oracle WebLogic Server Administration Console in a Web browser by entering the following URL:

```
http://hostname:port/console
```

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

The Oracle WebLogic Server Administration Console login page is displayed.

ORACLE WebLogic Server®
Administration Console



Welcome

Log in to work with the WebLogic Server domain

Username:

Password:

It is recommended that users log out when finished with the Administration Console or when visiting an untrusted site.

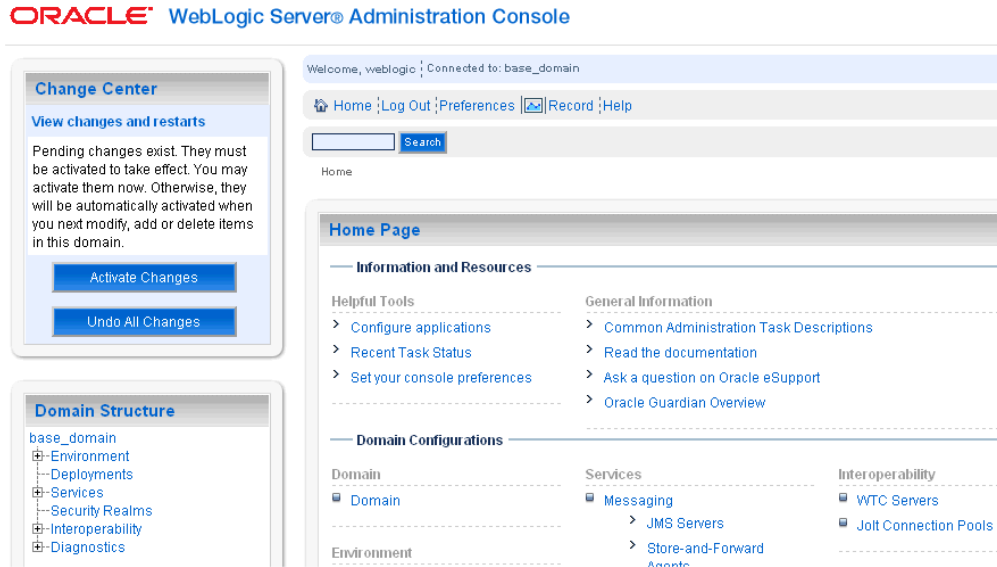
WebLogic Server Version: 10.3.0.0

Copyright © 1996,2008, Oracle and/or its affiliates. All rights reserved.

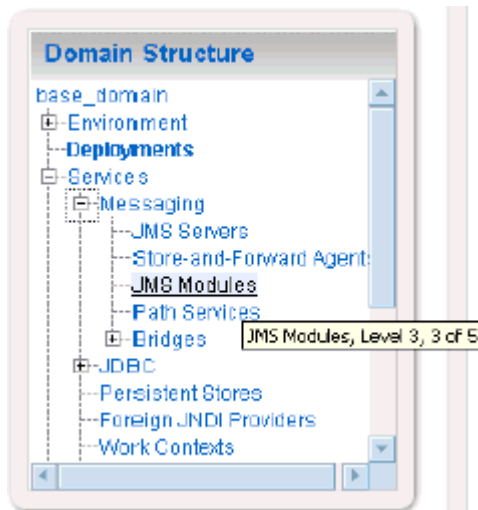
Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

3. Log on to the Oracle WebLogic Server Administrative Console using an account that has administrator privileges.

The Oracle WebLogic Server Administration Console home page is displayed.



4. In the Domain Structure section in the left pane, expand Services, Messaging, and click JMS Modules.



The JMS Modules page is displayed.

JMS Modules

JMS system resources are configured and stored as modules similar to standard J2EE modules. Such resources include queues, topics, connection factories, templates, destination keys, quota, distributed queues, distributed topics, foreign servers, and JMS store-and-forward (SAF) parameters. You can administratively configure and manage JMS system modules as global system resources.

This page summarizes the JMS system modules that have been created for this domain.

[Customize this table](#)

JMS Modules

New Delete Showing 1 to 2 of 2 Previous | Next

| <input type="checkbox"/> | Name ↕ | Type |
|--------------------------|--------------------|--------|
| <input type="checkbox"/> | conversational-jms | System |
| <input type="checkbox"/> | wseeJmsModule | System |

New Delete Showing 1 to 2 of 2 Previous | Next

5. Click Lock & Edit.

Change Center

View changes and restarts

Click the Lock & Edit button to modify, add or delete items in this domain.

Lock & Edit

Release Configuration

6. Click conversational-jms.

This page displays general information about a JMS system module and its resources. It also allows you to configure new resources and access existing resources.

Name: conversational-jms The name of this JMS system module. [More Info...](#)

Descriptor File Name: jms/conversational-jms.xml The name of the JMS module descriptor file. [More Info...](#)

This page summarizes the JMS resources that have been created for this JMS system module, including queue and topic destinations, connection factories, JMS templates, destination sort keys, destination quota, distributed destinations, foreign servers, and store-and-forward parameters.

[Customize this table](#)

Summary of Resources

 Showing 1 to 10 of 37 [Previous](#) | [Next](#)

| <input type="checkbox"/> | Name | Type | JNDI Name | Subc |
|--------------------------|---|--------------------|---|------|
| <input type="checkbox"/> | cgJMSQueue | Queue | jws.queue | cgJW |
| <input type="checkbox"/> | cgQueue | Connection Factory | weblogic.jms.QueueConnectionFactory | cgQL |
| <input type="checkbox"/> | com.bea.wli.b2b.server.TopicConnectionFactory | Connection Factory | com.bea.wli.b2b.server.TopicConnectionFactory | com. |
| <input type="checkbox"/> | dynamic_queue | Queue | inbnd_web_queue.AsyncDispatcher | cgJW |

7. Click New.

The Create a New JMS System Module Resource page is displayed.

Create a New JMS System Module Resource

Choose the type of resource you want to create.

Use these pages to create resources in a JMS system module, such as queues, topics, templates, and connection factories.

Depending on the type of resource you select, you are prompted to enter basic information for creating the resource. For targetable resources, like stand-alone queues and topics, connection factories, distributed queues and topics, foreign servers, and JMS BAF destinations, you can also proceed to targeting pages for selecting appropriate server targets. You can also associate targetable resources with subdeployments, which is an advanced mechanism for grouping JMS module resources and the members to server resources.

Connection Factory Defines a set of connection configuration parameters that are used to create connections for JMS clients. [More Info...](#)

Queue Defines a point-to-point destination type, which are used for asynchronous peer communications. A message delivered to a queue is distributed to only one consumer. [More Info...](#)

Topic Defines a publish/subscribe destination type, which are used for asynchronous peer communications. A message delivered to a topic is distributed to all topic consumers. [More Info...](#)

8. Select the **Queue** option and click Next.

The Create a New JMS System Module Resource page is displayed.

9. Enter a queue name and JNDI name in the corresponding fields.

10. Click Next.

The following page is displayed.

11. From the Subdeployments drop-down list, select cgJMSServer.

12. Click Finish.

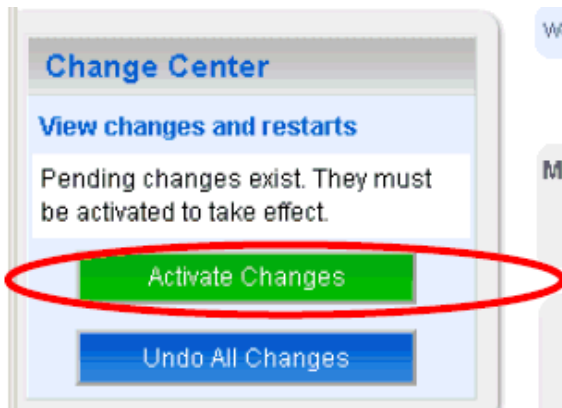
The new queue is created.

Summary of Resources

New Delete Showing 1 to 10 of 36 Previous

| <input type="checkbox"/> | Name ↕ | Type | JNDI Name |
|--------------------------|---|--------------------|---|
| <input type="checkbox"/> | cgJWSQueue | Queue | jws.queue |
| <input type="checkbox"/> | cgQueue | Connection Factory | weblogic.jms.QueueConnectionFactory |
| <input type="checkbox"/> | com.bea.wli.b2b.server.TopicConnectionFactory | Connection Factory | com.bea.wli.b2b.server.TopicConnectionFac |
| <input type="checkbox"/> | mysap_web.queue.AsyncDispatcher | Queue | mysap_web.queue.AsyncDispatcher |
| <input type="checkbox"/> | wli.b2b.failedmessage.queue | Queue | wli.b2b.failedmessage.queue |
| <input type="checkbox"/> | wli.b2b.mt.event.stream | Queue | wli.b2b.mt.event.stream |
| <input type="checkbox"/> | wli.b2b.mt.event.stream_error | Queue | wli.b2b.mt.event.stream_error |
| <input type="checkbox"/> | wli.internal.b2b.ebxmlencoder.queue | Queue | wli.internal.b2b.ebxmlencoder.queue |
| <input type="checkbox"/> | wli.internal.b2b.events.topic | Topic | wli.internal.b2b.events.topic |
| <input type="checkbox"/> | wli.internal.b2b.rosettanetencoder.queue | Queue | wli.internal.b2b.rosettanetencoder.queue |

13. Click Activate Changes to activate the new queue.



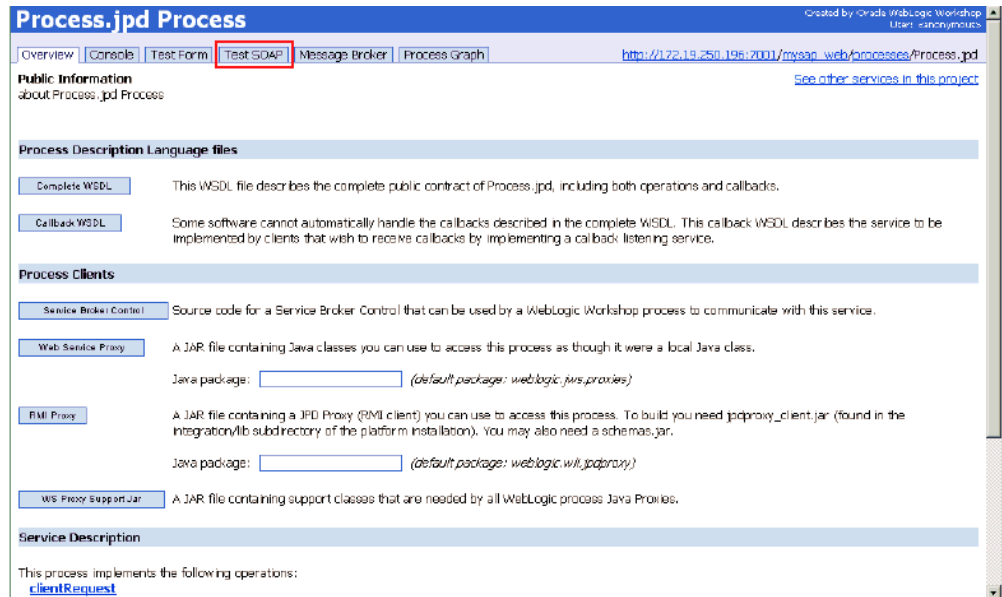
Invoking the Input XML Using a Web Browser

This section describes how to invoke the input XML using a Web browser.

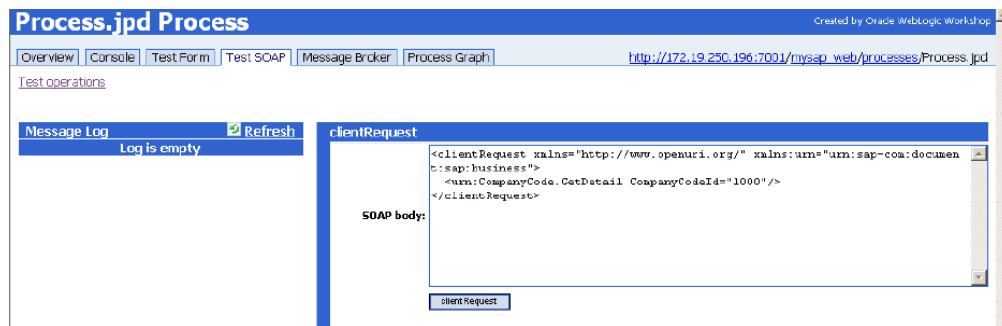
1. Enter the following URL in a Web browser:

`http://172.19.250.196:7001/mysap_web/processes/Process.jspd?`

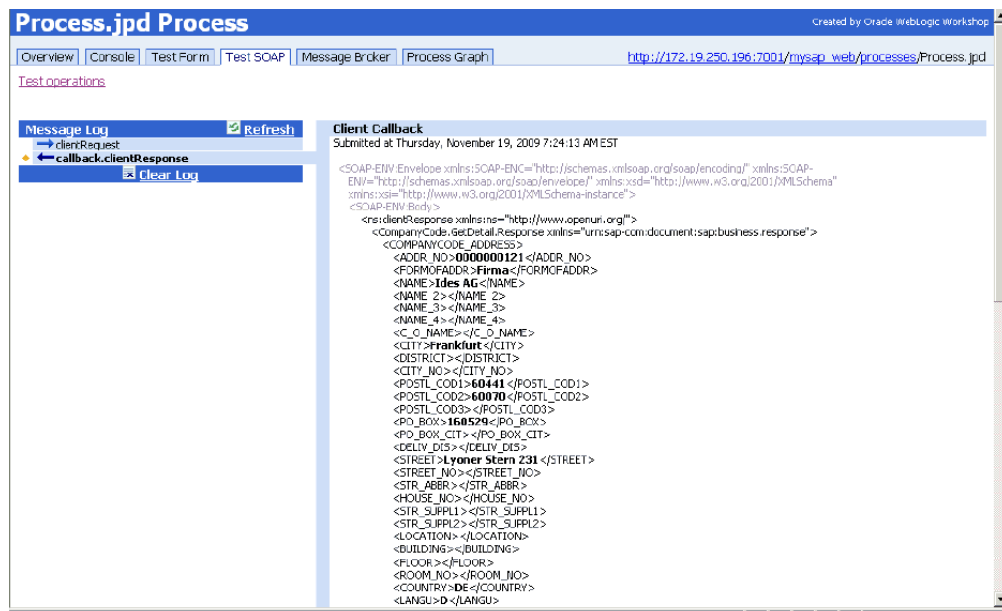
2. Click the **TestSOAP** tab.



3. In the clientRequest area, provide the input value and click **client Request**.



The request is processed, as shown in the following image.



4. To view the output response, click **callback.clientResponse**.

Advanced User Tools

This chapter includes the following topics:

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

Web Services Policy-Based Security

Application Explorer provides a security model called Web services policy-based security for Business Services Engine (BSE) configurations. The following topics describe how the feature works and how to configure it.

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

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

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

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

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

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

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

policy is associated to the Business Service. At that time, only those granted execution permissions, or users not part of the group that has been denied execution permissions, have access to the Business Service.

Configuring Web Services Policy-Based Security

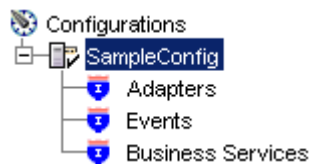
The following procedures describe how to configure Web services policy-based security.

Creating and Associating a User with a Policy

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

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

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



Perform the following steps:

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



1. Right-click Users and click New User.

The New User dialog box is displayed.

Provide the following information:

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

1. Click OK.



The new user is added under the Users node.

Creating a Group to Use With a Policy

To create a group to use with a policy:

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

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



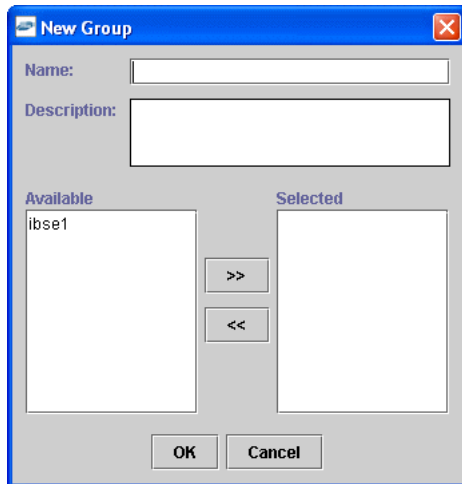
Perform the following steps:

- a. Expand the Business Services node.
- b. Expand the Configuration node.
- c. Expand the Security node.

- d. Expand the Users and Groups node.
4. Right-click Groups and select New Group.



The New Group dialog box is displayed.



Provide the following information:

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

The new group is added under the Group node.

Creating an Execution Policy

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

To create an execution policy:

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

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

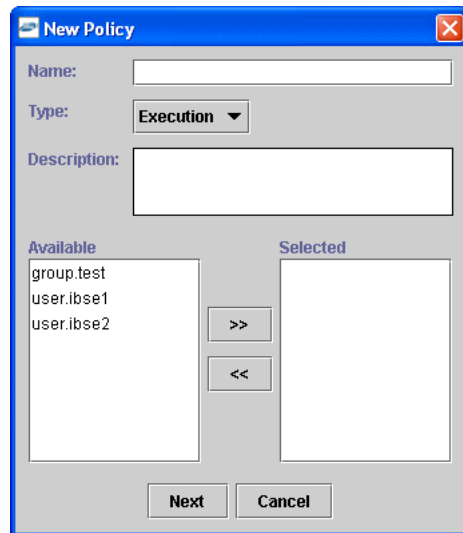


Perform the following steps:

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



The New policy dialog box is displayed.



Provide the following information:

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

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

1. When you have selected at least one user selected, click OK.
2. Click Next.

The New Policy permissions dialog box is displayed.



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

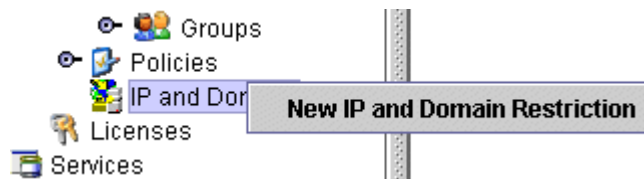
The following pane summarizes your configuration.

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

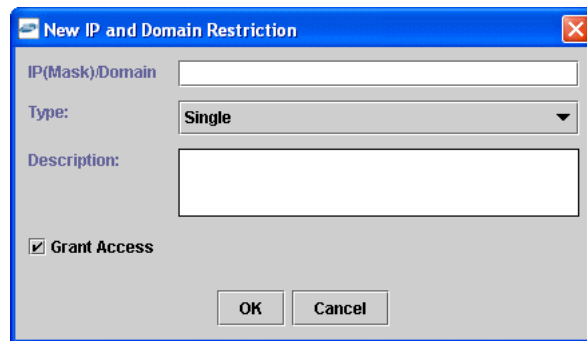
Using the IP and Domain Restrictions Policy Type

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

1. Open Application Explorer.
2. Right-click the configuration to which you want to connect, for example, SampleConfig. See [Chapter 2, "Configuring Oracle Application Adapter for SAP R/3"](#) for information on creating a new configuration.
3. Select Connect.
 - Nodes appear for Adapters, Events, and Business Services (also known as Web services).
 - a. Expand the Business Services node.
 - b. Expand the Configuration node.
 - c. Expand the Security node.
1. Right-click IP and Domain and select New IP and Domain Restriction.



The New IP and Domain Restriction dialog box is displayed.



Provide the following information:

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

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

The following pane summarizes your configuration.

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

Migrating Repositories

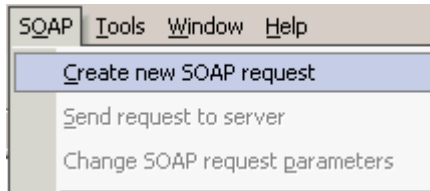
During design time, the Oracle repository is used to store metadata created when using Application Explorer to configure adapter connections, browse EIS objects, configure services, and configure listeners to listen for EIS events. The information in the repository is also referenced at run-time. For management purposes, you can migrate BSE and J2CA repositories that are configured for Oracle to new destinations without affecting your existing configuration. For example, you may want to migrate a repository from a test environment to a production environment.

Migrating a BSE Repository

To migrate a BSE repository:

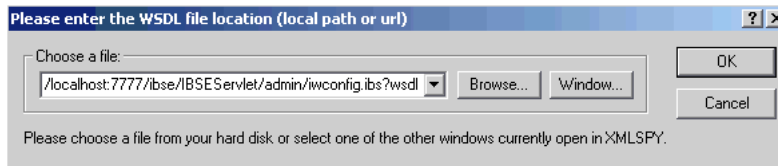
1. Copy the BSE control service URL, for example:
`http://localhost:7777/ibse/IBSEServlet/admin/iwcontrol.ibs`
2. Open a third-party XML editor, for example, XMLSPY.
3. From the menu bar, click SOAP.

A list of options appears.



4. Select Create new SOAP request.

The WSDL file location dialog box is displayed.



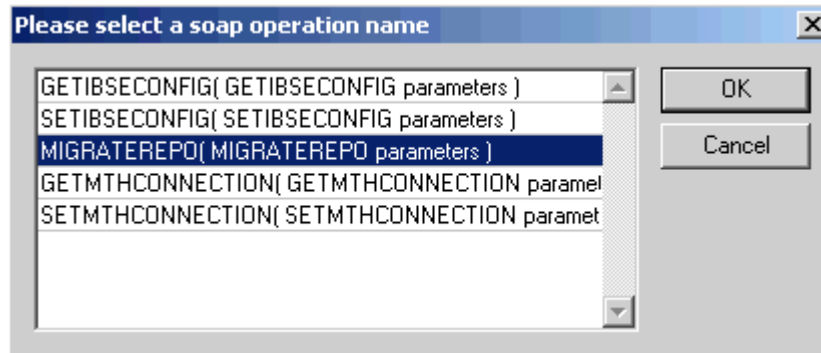
Perform the following steps:

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

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

5. Click OK.

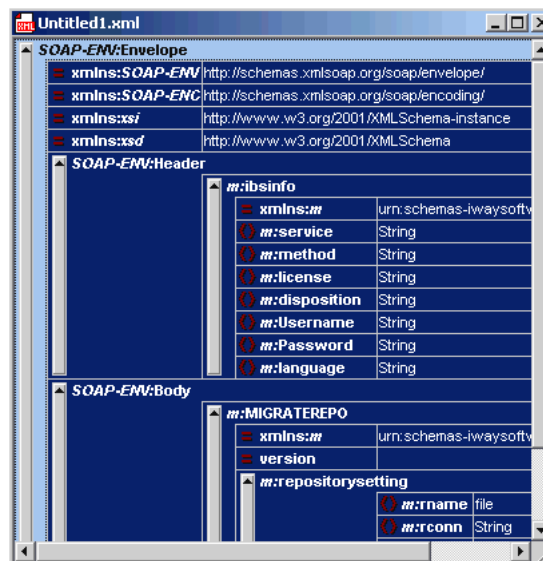
The soap operation name dialog box is displayed and the available control methods are listed.



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

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

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



7. Locate the Text view icon in the toolbar.



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

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

9. Locate the following section:

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

Perform the following steps:

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

The Oracle repository URL has the following format:

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

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

10. Perform one of the following migration options.

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

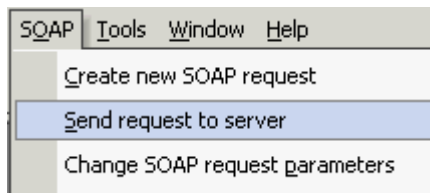
```
<m:servicename>SAPService1</m:servicename>
```

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

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

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

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



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

Migrating a J2CA Repository

To migrate a J2CA repository:

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

```
wls_home\erp-adapters\config\JCA_CONFIG
```

Where *JCA_CONFIG* is the name of your J2CA configuration.

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

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

Troubleshooting and Error Messages

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

- [Troubleshooting](#)
- [BSE Error Messages](#)

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

Troubleshooting

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

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

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

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

Application Explorer

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

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

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

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

| Error | Solution |
|---|--|
| <p>SAP R/3 does not appear in the Application Explorer Adapter node list.</p> | <p>Ensure that the <code>sapjco.jar</code> and <code>sapjcorfc.dll</code> files are added to the <code>lib</code> directory. Ensure the <code>librfc32.dll</code> file is added to the Windows <code>system32</code> folder.</p> |
| <p>Cannot connect to Oracle Application Adapter for SAP R/3 from Application Explorer. Problem activating adapter</p> | <p>Ensure that:</p> <ul style="list-style-type: none"> ■ SAP R/3 is running. ■ The Application Server name, System Number, and Client Number are correct. ■ The SAP R/3 user ID and password are correct. |
| <p>Cannot connect to the SAP R/3 target through Application Explorer. The following error message appears:</p> <pre>Error getting target [SAP] - java.lang.Exception: Error Logon to SAP System</pre> | <p>Ensure that you enter the correct connection parameters when connecting to the SAP R/3 target.</p> |
| <p>Cannot connect to your SAP R/3 system through Application Explorer. The following error message appears:</p> <pre>Problem activating adapter. (com.ibi.sapr3.SapAdapterException : com.sap.mw.jco.JCO\$Exception: (102) RFC_ERROR_COMMUNICATION: Connect to SAP gateway failed Connect_PM GWHOST=isdsvr8, GWSERV=sapgw00, ASHOST=isdsvr8, SYSNR=00 LOCATION CPIC (TCP/IP) on local host ERROR partner not reached (host isdsvr8, service 3300) TIME Fri Aug 27 11:49:14 2004 RELEASE 620 COMPONENT NI (network interface) VERSION 36 RC -10 MODULE ninti.c LINE 979 DETAIL NipConnect2 SYSTEM CALL SO_ERROR ERRNO 10061 ERRNO TEXT WSAECONNREFUSED: Connection refused COUNTER 1). Check logs for more information</pre> | <p>Ensure that SAP R/3 is running and that you are using the correct parameter values to connect to your application server.</p> |
| <p>Cannot connect to your SAP R/3 system through Application Explorer even though SAP R/3 is running. The following error message appears:</p> <pre>Problem activating adapter. (com.ibi.sapr3.SapAdapterException : java.lang.ExceptionInInitializerEr ror: JCO.classInitialize(): Could not load middleware layer 'com.sap.mw.jco.rfc.MiddlewareRFC' JCO.nativeInit(): Could not initialize dynamic link library sapjcorfc [no sapjcorfc in java.library.path]. java.library.path</pre> | <p>Ensure that the <code>sapjcorfc.dll</code> file is added to the <code>lib</code> directory and the <code>librfc32.dll</code> file is added to the Windows <code>system32</code> folder.</p> |

| Error | Solution |
|--|--|
| <p>The dll is loaded in another classloader (BSE and J2CA are installed on the same server). The following error message appears:</p> <pre>com.ibi.sapr3.SapAdapterException: java.lang.ExceptionInInitializerEr ror: JCO.classInitialize(): Could not load middleware layer 'com.sap.mw.jco.rfc.MiddlewareRFC' JCO.nativeInit(): Could not initialize dynamic link library sapjcorfc [Native Library F:\iWay55.008.0628\lib\sapjcorfc.d ll already loaded in another classloader]. java.library.path</pre> | <p>Add <code>sapjco.jar</code> to the server classpath.</p> |
| <p>Unable to start Application Explorer in a Solaris environment. The following exception is thrown in the console:</p> <pre>javax.resource.ResourceException: IWAFManagedConnectionFactory: License violation.at com.ibi.afjca.spi.IWAFManagedConne ctionFactory.createConnectionFacto ry(IWAFManagedConnectionFactory.ja va:98)at com.iwaysoftware.iwae.common.JCATr ansport.getConnectionFactory(JCATr ansport.java:133) at com.iwaysoftware.iwae.common.JCATr ansport.initJCA(JCATransport.java: 69)at com.iwaysoftware.iwae.common.JCATr ansport.<init>(JCATransport.java:6 2)at com.iwaysoftware.iwae.common.Adapt erClient.<init>(AdapterClient.java :85)at com.ibi.bse.ConfigWorker.run(Confi gWorker.java:41) at java.lang.Thread.run(Thread.java:5 34) Could not create the connection factory.</pre> | <p>JAVACMD is not set on the user system. Before starting Application Explorer, export JAVACMD as follows:</p> <p>JAVACMD=<code><jdk_home>/bin/java</code>, where <code><jdk_home></code> is the directory where JDK is installed on your system.</p> |
| <p>Logon failure error at run-time</p> | <p>If the password for connecting to your SAP R/3 system is not specified when creating a target or with the Edit option in Application Explorer, you will be unable to connect to SAP R/3. The connection password is not saved in <code>repository.xml</code>. Update the password using the Edit option in Application Explorer, then restart the application server.</p> |
| <p>The following exception occurs when you start Application Explorer by activating <code>ae.bat</code> (not <code>iaexplorer.bat</code>):</p> <pre>java.lang.ClassNotFoundException: org.bouncycastle.jce.provider.Boun cyCastleProvider</pre> | <p>This is a benign exception. It does not affect adapter functionality. Download BouncyCastle files from:</p> <p>ftp://ftp.bouncycastle.org/pub</p> |

SAP R/3

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

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

Oracle Adapter J2CA

| Error | Solution |
|--|---|
| <p>In Application Explorer, the following error message appears when you attempt to connect to an Oracle Adapter J2CA configuration:</p> <pre>Could not initialize JCA</pre> | <p>In the Details tab in the right pane, ensure that the directory specified in the Home field points to the correct directory, for example,</p> <pre>wls_ home\erp-adapters\tools\iwae\bin\ .\..\..\</pre> |

BSE Error Messages

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

General Error Handling in BSE

BSE serves as both a SOAP gateway into the adapter framework and as the engine for some of the adapters. In both design-time and run-time, various conditions can cause errors in BSE when Web services that use adapters run. Some of these conditions and resulting errors are exposed the same way, regardless of the specific adapter; others are exposed differently, based on the adapter being used. This topic explains what you can expect if you encounter some of the more common error conditions on an adapter-specific basis. Usually the SOAP gateway (**agent**) inside BSE passes a SOAP request message to the adapter required for the Web service. If an error occurs, how it is exposed depends on the adapter and the API or interfaces that the adapter uses. A few scenarios cause the SOAP gateway to generate a SOAP fault. In general, anytime the SOAP agent inside BSE receives an invalid SOAP request, a SOAP fault element is generated in the SOAP response. The SOAP fault element contains fault string and fault code elements. The fault code contains a description of the SOAP agent error.

The following SOAP response document results when BSE receives an invalid SOAP request:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <faultcode>SOAP-ENV:Client</faultcode>
      <faultstring>Parameter node is missing</faultstring>
    </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

In this example, BSE did not receive an element in the SOAP request message that is mandatory for the WSDL for this Web service.

Adapter-Specific Error Handling

When an adapter raises an exception during run-time, the SOAP agent in BSE produces a SOAP fault element in the generated SOAP response. The SOAP fault element contains fault code and fault string elements. The fault string contains the native error description from the adapter target system. Since adapters use the target system interfaces and APIs, whether an exception is raised depends on how the target systems interface or API treats the error condition. If a SOAP request message is passed to an adapter by the SOAP agent in BSE and that request is invalid based on the WSDL for that service, the adapter may raise an exception yielding a SOAP fault. While it is almost impossible to anticipate every error condition that an adapter may encounter, the following is a description of how adapters handle common error conditions and how they are then exposed to the Web services consumer application.

Oracle Application Adapter for SAP R/3 Invalid SOAP Request

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

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

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

Empty Result From SOAP Request

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

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

Failure to Connect to SAP R/3

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

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Body>
  <SOAP-ENV:Fault>
    <faultcode>SOAP-ENV:Server</faultcode>
    <faultstring>Error processing agent [XDSapIfrAgent] - XD[RETRY]
Connect to SAP gateway failed Connect_PM GWHOST=ESDSUN, GWSERV=sapgw00,
ASHOST=ESDSUN,
SYSNR=00 LOCATION CPIC (TCP/IP) on local host ERROR partner not reached (host
ESDSUN, service 3300)
TIME Mon Jun 30 16:01:02 2003 RELEASE 620 COMPONENT NI (network interface) VERSION
36 RC -10 MODULE ninti.c LINE 976 DETAIL NipConnect2
SYSTEM CALL SO_ERROR ERRNO 10061 ERRNO TEXT WSAECONNREFUSED: Connection refused
COUNTER 1</faultstring>
  </SOAP-ENV:Fault>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Invalid SOAP Request

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

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

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

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

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

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

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

Configuring SAP R/3 for Inbound and Outbound Processing

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

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

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

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

Configuring SAP R/3 Inbound Processing

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

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

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

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

Configuring a Logical System

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

Defining a Logical System

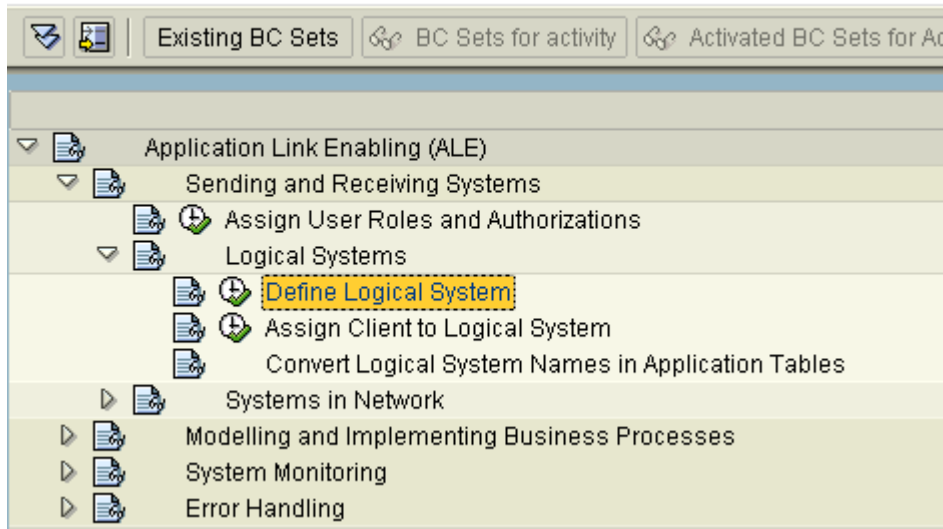
To define a logical system:

1. Run the **sale** transaction.



The Display IMG window is displayed.

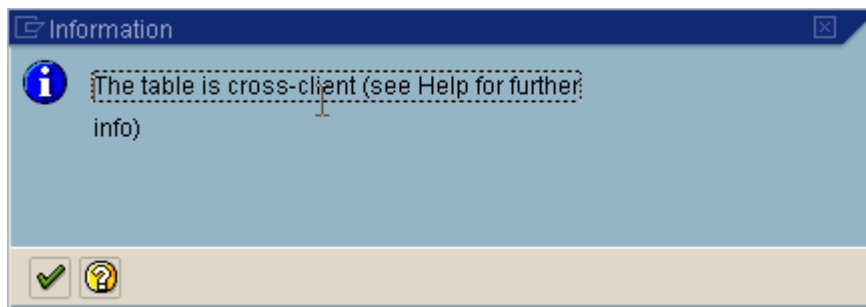
Display IMG



Perform the following steps:

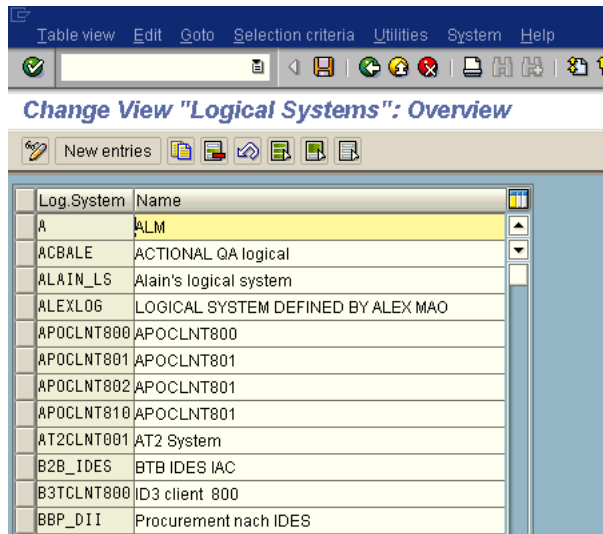
- a. Expand Sending and Receiving Systems.
 - b. Expand **Logical Systems**.
 - c. Select Define Logical System.
2. Click the IMG - Activity icon.

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



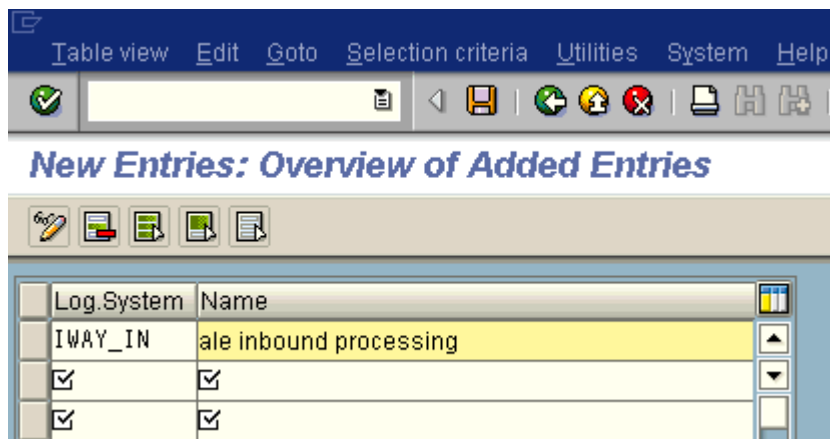
3. Click the **check mark** icon to continue.

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



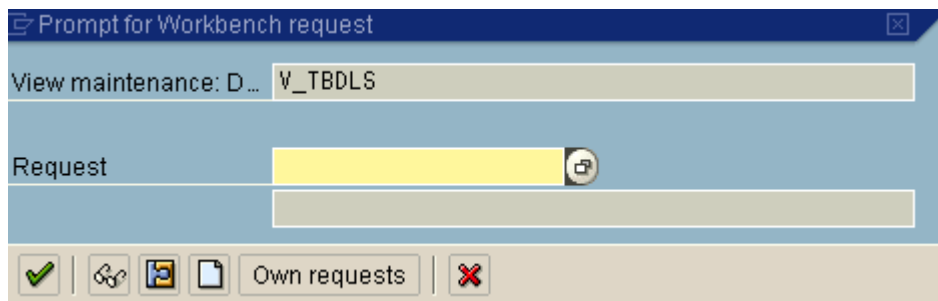
4. Click New Entries.

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



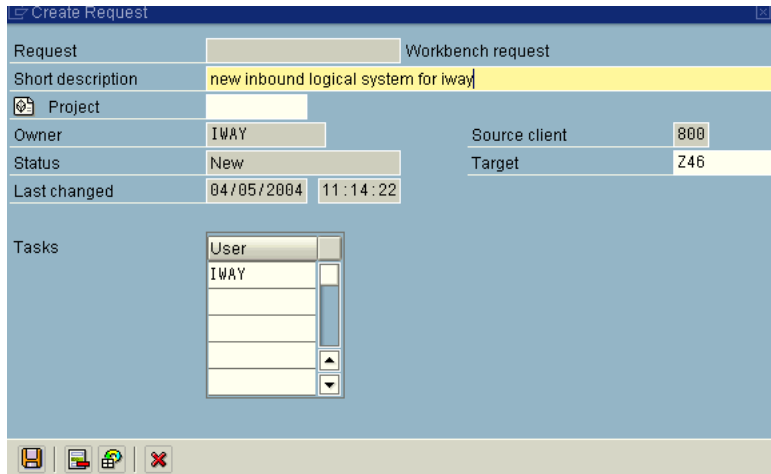
5. Enter the Logical System, for example, ORACLETDS, in the **Log.System** column and provide a description in the **Name** column.
6. Click Save.

The Prompt for Workbench request dialog box is displayed.



7. Click the Create Request icon.

The Create Request dialog box is displayed.



- Enter a name and description for your request and click Save.
The logical system you configured, for example, ORACLETDS, is now added to the list.

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

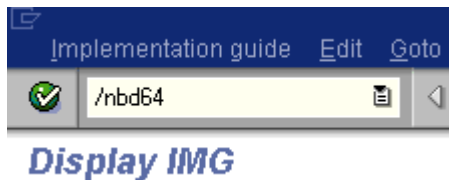
Configuring a Distribution Model

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

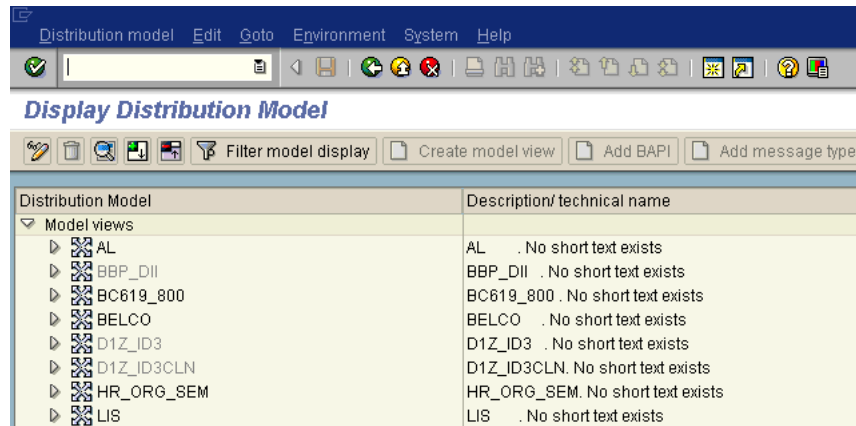
Defining a Distribution Model

To define a distribution model:

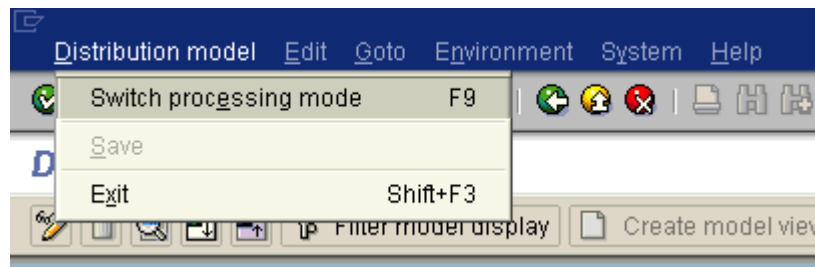
- Run the **bd64** transaction.



The Display Distribution Model window is displayed.

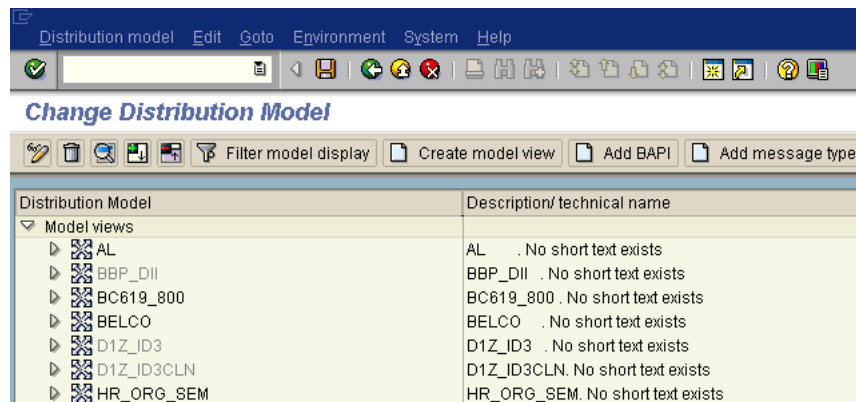


2. Click Distribution Model from the menu bar.



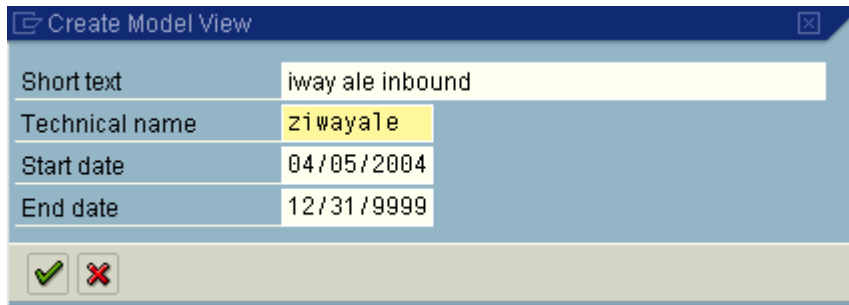
3. Select **Switch processing mode**.

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



4. Click Create model view.

The Create Model View dialog box is displayed.



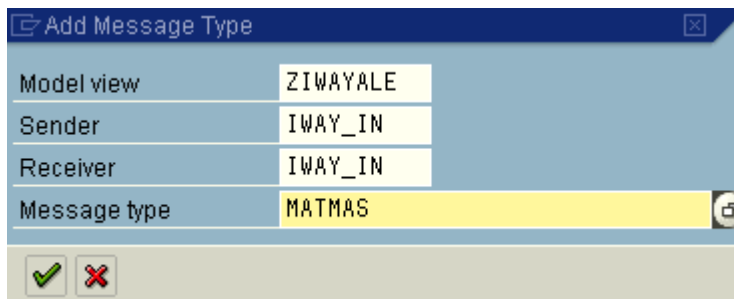
5. Enter a model view name in the **Short text** field and a name in the **Technical name** field, which also serves as a description.
6. Click the **check mark** icon to enter the information.

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

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

7. Click Add message type.

The Add Message Type dialog box is displayed.



Perform the following steps:

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

Defining a Partner Profile

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

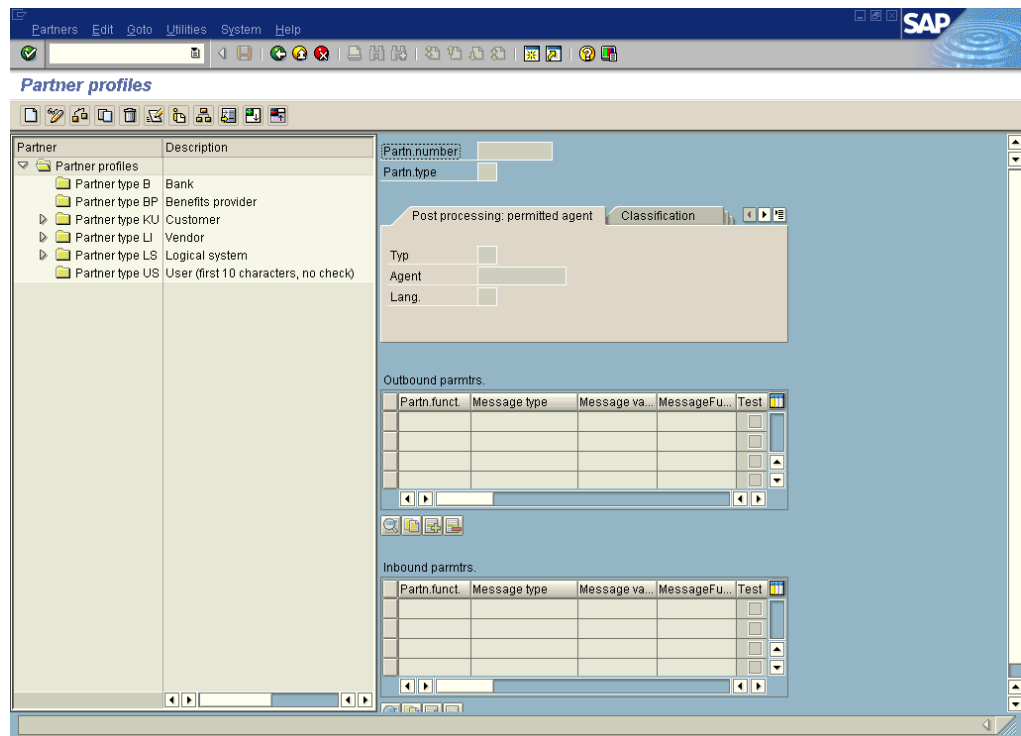
Defining a Partner Profile

To define a partner profile:

1. Run the **we20** transaction.

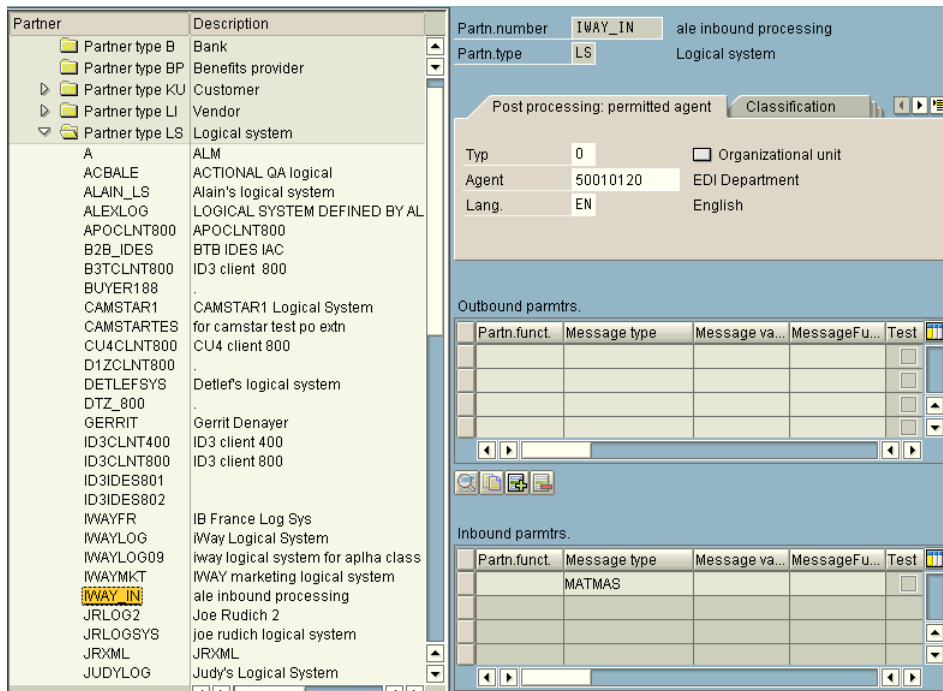


The Partner profiles window is displayed.



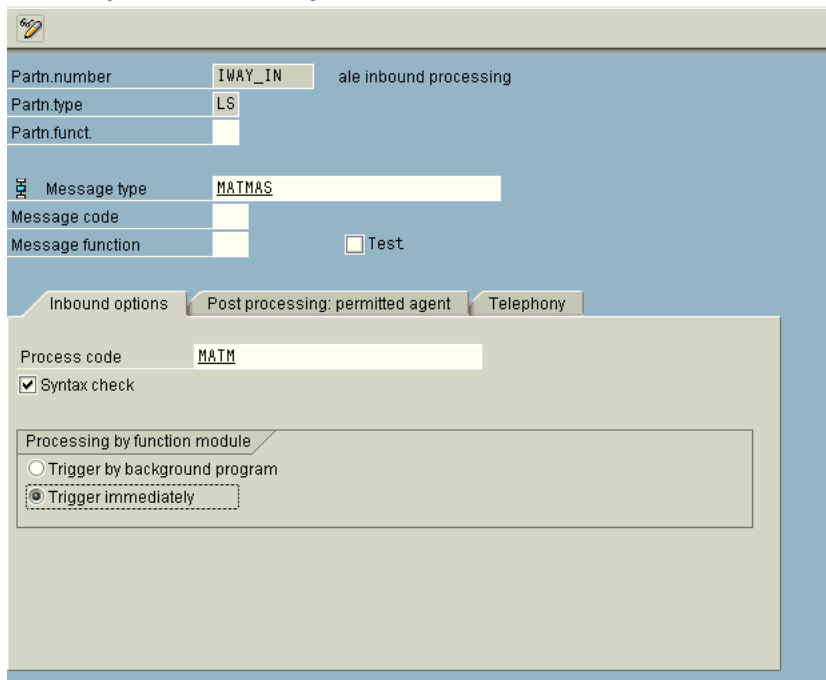
2. In the left pane, expand Partner type LS and select the logical system you configured from the list, for example, ORACLETDS.

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



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

Partner profiles: Inbound parameters



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

The Inbound options tab is selected by default.

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

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

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

Configuring SAP R/3 Outbound Processing

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

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

Related Concepts and Terminology

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

Client and Server Programs

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

- RFC Client
- RFC Server

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

SAP R/3 Gateway

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

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

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

Program IDs and Load Balancing

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

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

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

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

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

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

Connection Pooling

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

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

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

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

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

Registering Your Program ID in SAPGUI

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

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

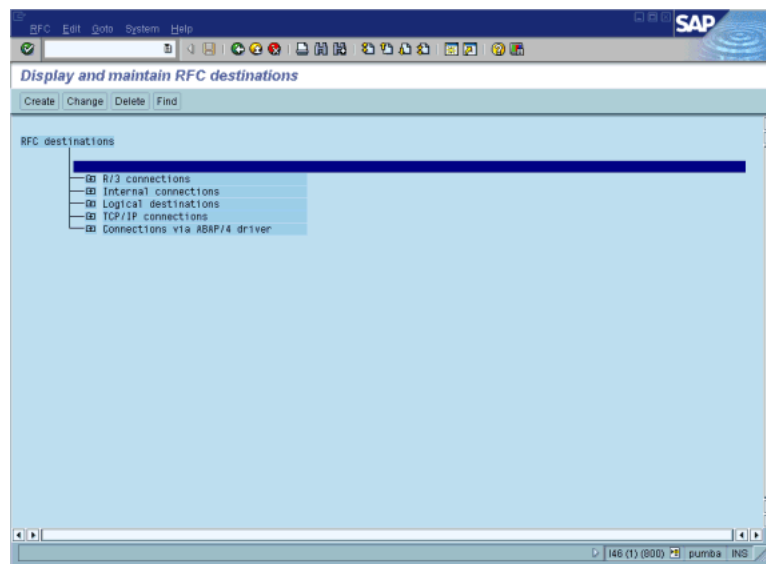
The RFC destination is a symbolic name (for example, ORACLETDS) that is used to direct events to a target system, masking the program ID. The Program ID is configured in both SAPGUI and the event adapter.

Registering Your Program ID

To register your program ID:

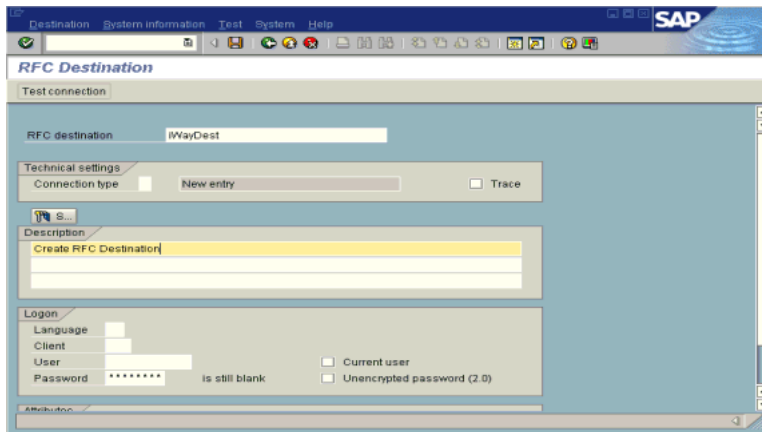
1. Launch the SAP GUI and log in to the SAP R/3 system.
2. Select **Tools, Administration, Network**, and then RFC destination.
3. Run the **SM59** transaction.

The Display and maintain RFC destinations window is displayed.



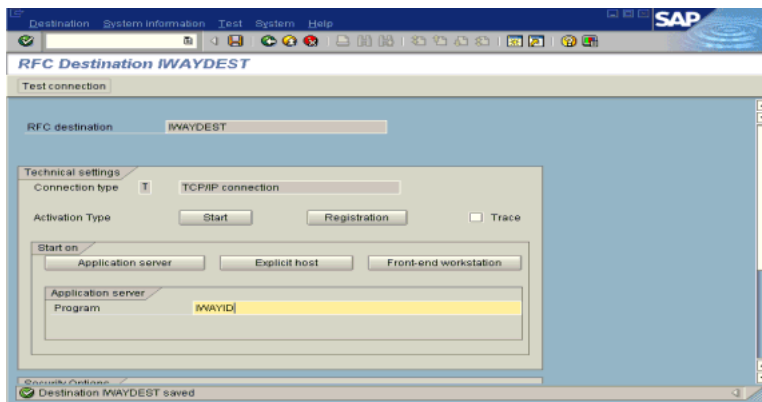
4. Select TCP/IP connections and click Create.

The RFC Destination window is displayed.



Provide the following information:

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



Perform the following steps:

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

Testing the SAP R/3 Event Adapter

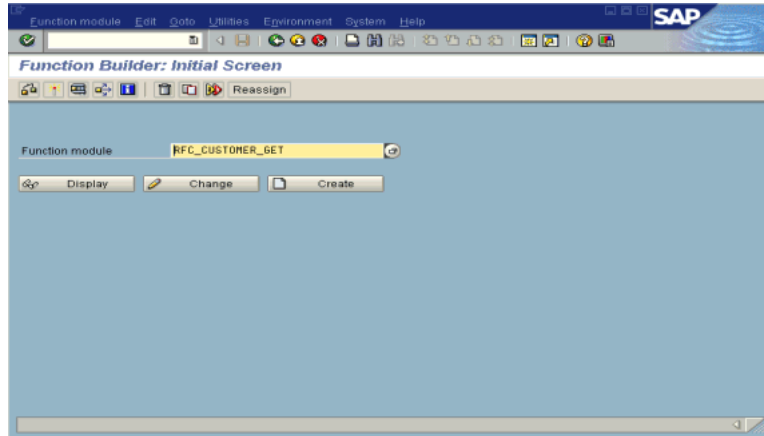
In the SAP Server, the SE37 transaction enables you to send an RFC (Remote Function Call) or a BAPI (Business Application Programming Interface) to any RFC destination.

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

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

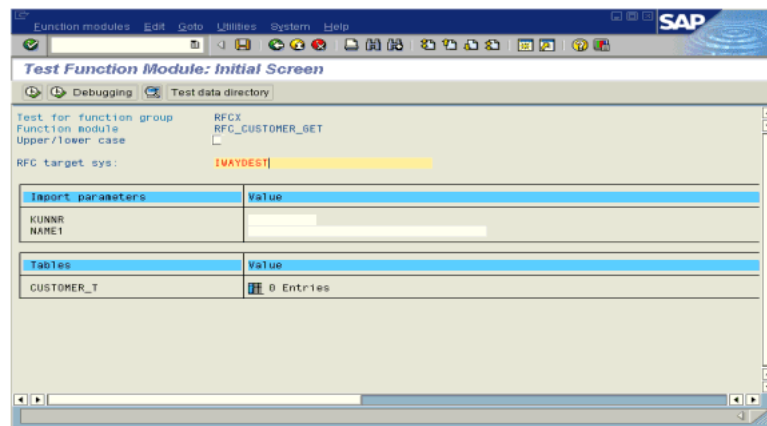
To test the SAP R/3 event adapter:

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



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

The Test Function Module: Initial Screen window is displayed.



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

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

Application Link Embedding Configuration for the Event Adapter

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

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

Defining a Port

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

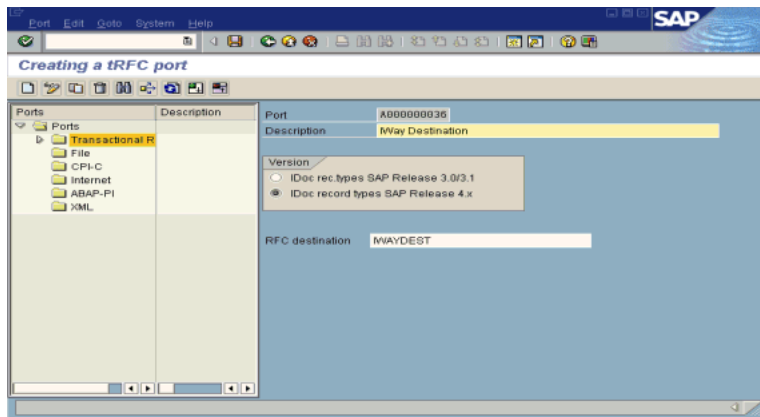
Defining a Port

To define a port:

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

You can also run the WE21 transaction.

The Creating a tRFC port window is displayed.



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

Creating a Logical System

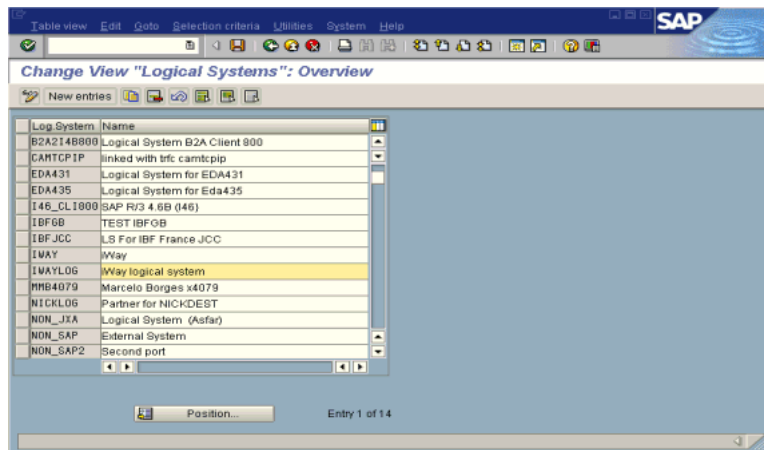
One type of partner is a logical system. A logical system manages one or more RFC destinations.

Creating a Logical System

To create a logical system called ORACLETDS:

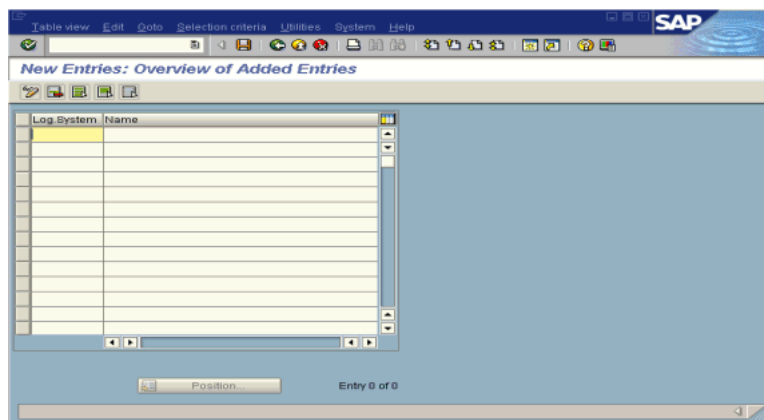
1. In the ALE configuration, enter the area menu selection SALE transaction.
2. Select SAP Reference IMG.
3. Expand the following nodes: **Basis Components, Application Link Enabling (ALE), Sending and Receiving Systems, Logical Systems,** and Define Logical System.
4. Click the check mark beside Define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

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



5. Click New entries.

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



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

Creating a Partner Profile

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

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

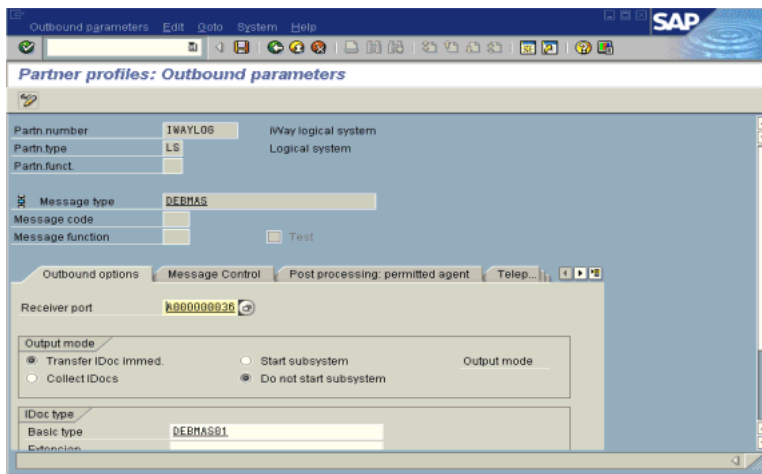
Creating a Partner Profile

To create a partner profile:

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

You can also run the WE21 transaction.

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



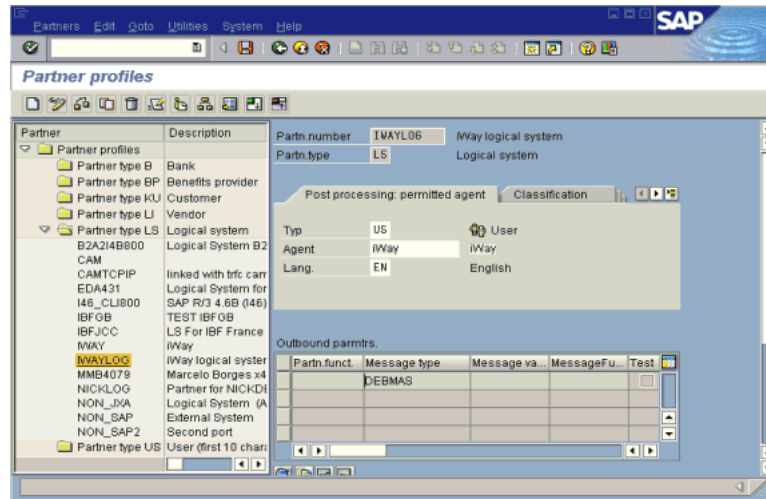
Perform the following steps:

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

Provide the following information:

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

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



Collected IDocs

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

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

Creating a Distribution Model for the Partner and Message Type

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

Creating a Distribution Model

To create a distribution model called ORAMOD:

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

You can also run the BD64 transaction.

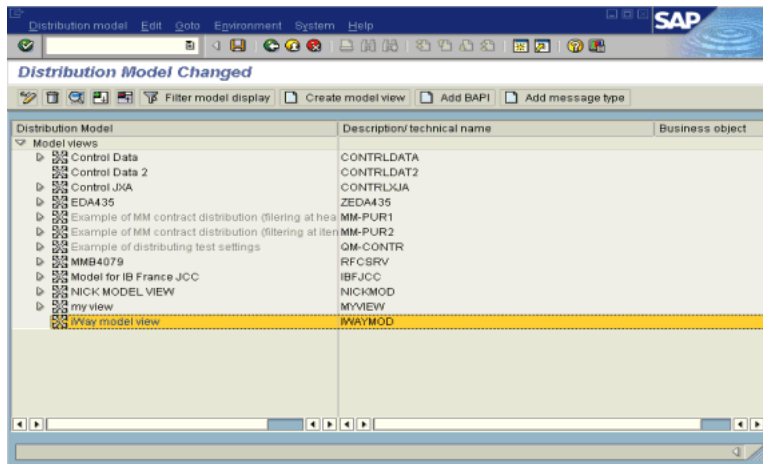
The Display Distribution Model window is displayed.

2. Select Create model view.

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

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

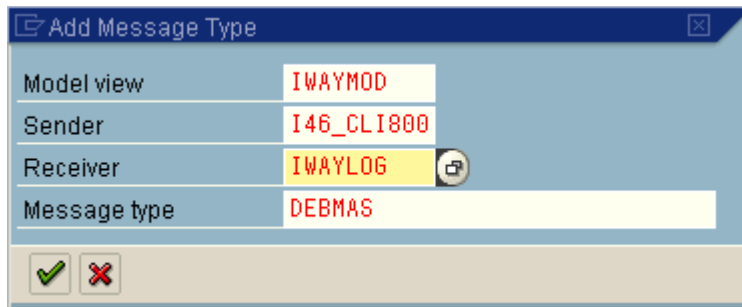
The Distribution Model Changed window is displayed, showing a tree structure of the distribution model.



Perform the following steps:

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

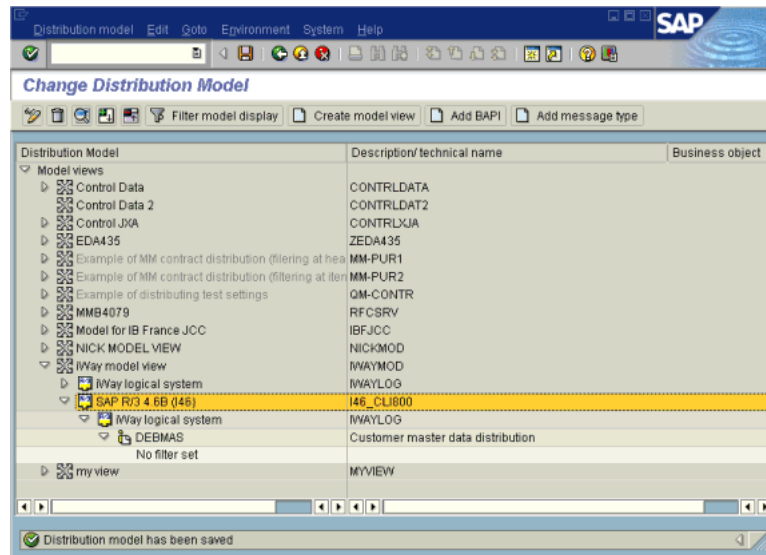
The Add Message Type box is displayed. It contains fields for specifying the sender and receiver of the message and the message type.



Provide the following information:

- a. In the **Sender** field, provide the sender that points to the SAP R/3 system, which sends the IDoc, for example, I46_CLI800.
In this case, the sender is an SAP 4.6B system.
 - b. In the **Receiver** field, provide the logical system, for example, ORACLETDS.
 - c. In the **Message type** field, provide the type of IDoc, for example, DEBMAS.
5. Click the **check mark** icon.
 6. Click **Save**.

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



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

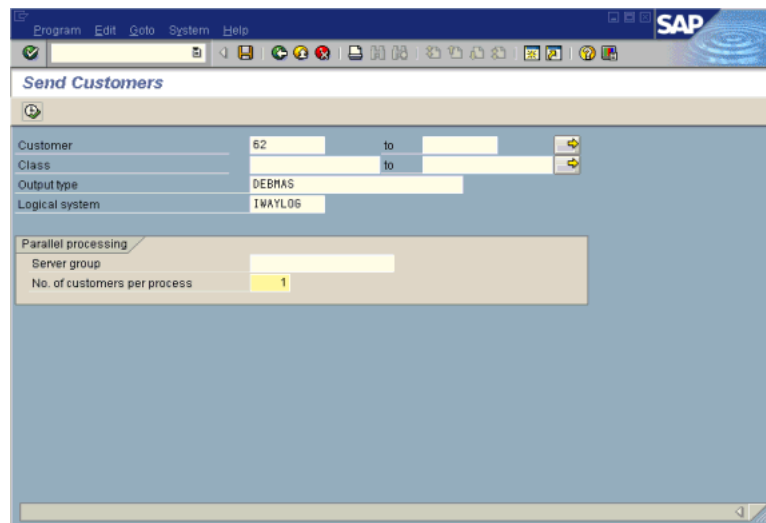
Testing the SAP R/3 ALE Configuration

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

Testing the SAP R/3 ALE Configuration

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

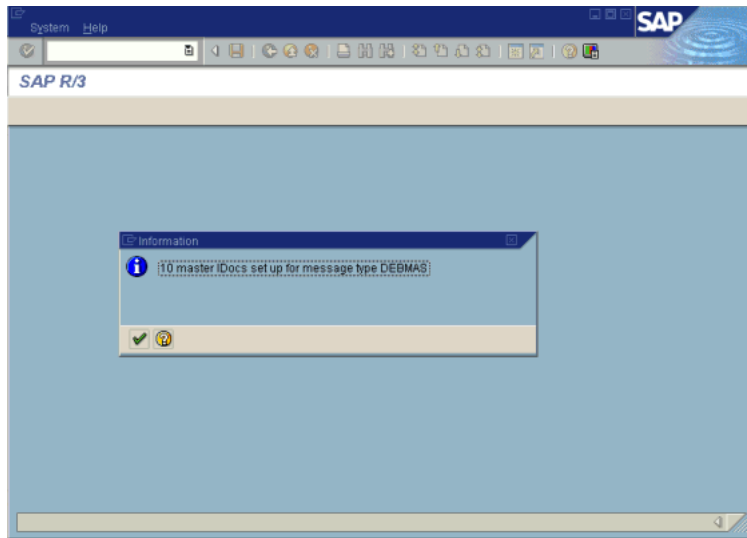
1. In the Send Customers window, enter the IDoc message type, for example, DEBMAS in the **Output type** field.



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

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

A confirmation window is displayed.



Oracle WebLogic Integration Deployment

This appendix describes Oracle WebLogic Integration (Oracle WLI) deployment and integration with Oracle Application Adapter for SAP R/3.

This chapter discusses the following topics:

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

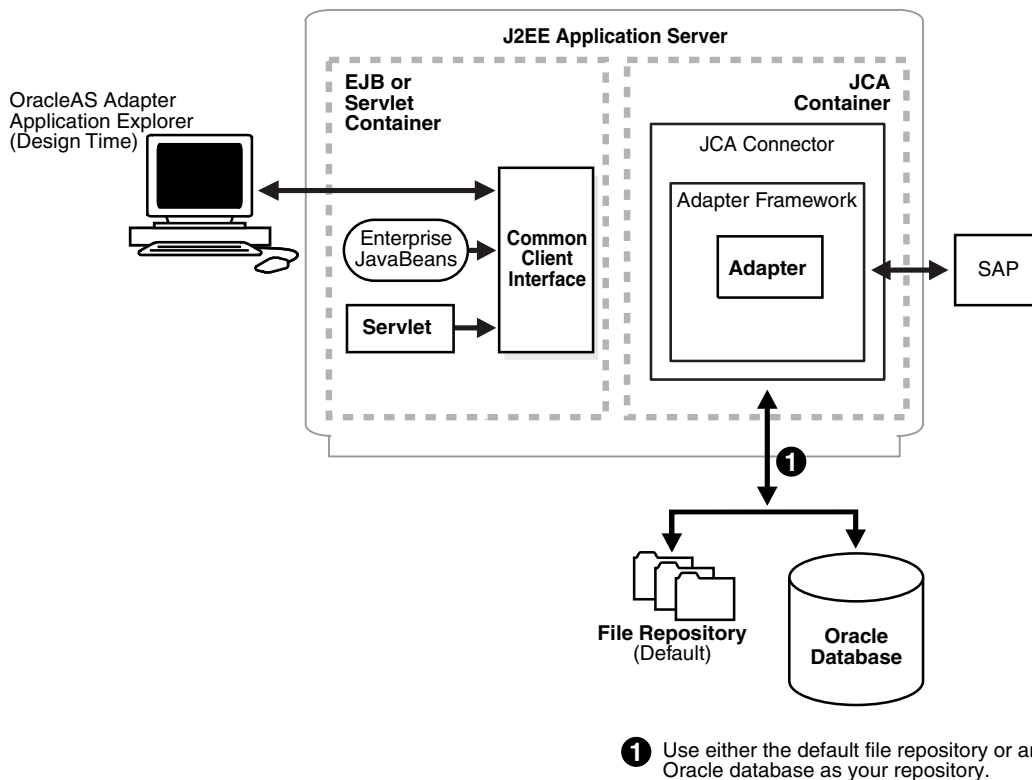
Adapter Integration with Oracle WLI

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

Deployment of Adapter

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

Figure B–1 Oracle Application Server J2CA Architecture

**See Also:**

- *Oracle Application Adapter Concepts*

Updating Adapter Configuration

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

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

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

Creating a Managed Connector Factory Object

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

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE connector PUBLIC "-//Sun Microsystems, Inc.//DTD Connector 1.0//EN"
'http://java.sun.com/dtd/connector_1_0.dtd'>
<connector>
  <display-name>IWAFJCA10</display-name>
```

```

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

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

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

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

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

```

```

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

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

```

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

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

Creating Multiple Managed Connector Factory Objects

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

```

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

```



```

    <connection-definition-group>

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

```

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

```

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

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

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

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

```

```

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

```

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

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

Modifying WSDL Files for Additional Connection Factory Values

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

Notice that the outbound WSDL has the following default connection factory:

```

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

```

The connection factory value must be changed to the following:

```

eis/OracleJCAAdapter/DefaultConnection1

```

For example:

```

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

```

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

Glossary

adapter

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

agent

Supports service protocols in listeners and documents.

business service

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

channel

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

listener

A component that accepts requests from client applications.

port

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

Index

A

- ABAP programs, A-9
- access rights, 6-1
- Activation Type, A-12
- adapter configuration
 - updating, B-2
- adapter exceptions, 7-4 to 7-5
- adapter types
 - resource, 1-1, B-1 to B-2
- adapters
 - configuring, 2-1 to 2-19, B-2
 - deploying, 1-4, B-1 to B-2
 - troubleshooting, 7-1 to 7-7
- Add Channel dialog box, 2-16
- Add Message Type dialog box, A-6
- Add Target dialog box, 2-2
- Advanced tab, 2-5, 2-18
- ae batch script, 7-1
- ALE (Application Link Embedding)
 - documents, A-9, A-14
- application adapters
 - configuring, 2-1 to 2-19, B-2
 - deploying, B-1 to B-2
 - troubleshooting, 7-1 to 7-7
- Application Explorer, 1-2
 - application systems and, 2-1
 - channels and, 2-16
 - debugging and, 7-1 to 7-3
 - J2CA configuration and, B-2
 - OracleAS Adapter J2CA and, 7-4
 - schemas and, 2-10
 - security and, 6-1, 6-6
 - testing and, 7-1 to 7-3
 - troubleshooting, 7-2 to 7-3
- Application Link Embedding (ALE)
 - documents, A-9, A-14
- Application Server dialog box, 2-3
- Application Server parameter, 2-4
- application systems
 - Application Explorer and, 2-1
 - supported, 2-2
- Available list, 6-4 to 6-5

B

- BAPI (Business Application Programming Interfaces), 7-4
 - registering, A-11
- BAPI Exception parameter, 2-5, 2-18
- BAPI requests, 1-4
- BSE (OracleAS Adapter Business Services Engine), 1-1, 1-4
 - connection access to, 6-6
 - troubleshooting, 7-4
- BSE control service URL, 6-8
- BSE repositories
 - migrating, 6-8
- Business Application Programming Interfaces (BAPI), 7-4
 - registering, A-11
- business events, 1-1
- business functions, 2-2, 2-10 to 2-13
 - creating schemas for, 2-10
- business processes, 1-3
- business services
 - creating, 2-12 to 2-13
 - deploying, 6-1
 - testing, 2-14
- Business Services node, 6-2 to 6-5, 6-7

C

- CCI calls, B-1
- Change Distribution Model window, A-6
- channels, 2-15
 - Application Explorer and, 2-16
 - creating, 2-16
 - deleting, 2-19
 - editing, 2-19
 - starting, 2-19
 - stopping, 2-19
- Channels node, 2-16
- channels. *See also* listeners
- Client parameter, 2-3, 2-17
- client programs
 - RFC Client, A-9
- Code page parameter, 2-3, 2-17
- Commit with wait parameter, 2-5, 2-18
- Configuration node under Business

- Services, 6-2 to 6-5, 6-7
- configuration parameters, B-4
 - IWayConfig, B-4
 - IWayHome, B-4
 - IWayRepoPassword, B-4
 - IWayRepoURL, B-4
 - IWayRepoUser, B-4
 - Loglevel, B-4
- configurations
 - overwriting, B-4 to ??
- configuring adapters, 2-1 to 2-19, B-2
- connecting to MySAP ERP, 2-1 to 2-8, 7-2, 7-6
- connecting to OracleAS Adapter J2CA, 7-4
- Connection dialog box, 2-8
- connection parameters, 2-3 to 2-8, 2-17 to 2-18, 7-2
 - Advanced, 2-5, 2-18
 - Application Server, 2-4
 - BAPI Exception Handling, 2-5
 - Client, 2-3, 2-17
 - Code page, 2-3
 - Commit with wait, 2-5
 - Connection pool name, 2-5
 - Connection pool size, 2-5
 - EDI Version, 2-4
 - Language, 2-3
 - Logon ticket (SSO2), 2-6
 - Logon ticket (X509), 2-6
 - MySAP ERP Trace, 2-3
 - Password, 2-3, 2-17
 - Security, 2-6
 - SNC level, 2-6
 - SNC library path, 2-7, 2-18
 - SNC mode, 2-6
 - SNC name, 2-6
 - SNC partner, 2-6
 - System, 2-4, 2-17
 - System Number, 2-4
 - User, 2-3, 2-17
- Connection pool name parameter, 2-5, 2-18
- Connection pool size parameter, 2-5, 2-18
- Connection type field, A-11
- connections
 - closing, 2-8
 - deleting, 2-8
 - establishing, 2-1 to 2-8
- Connector
 - deploying to Oracle Application Server, B-1
- connector factories, B-2
- connector factory objects, B-2
 - multiple, B-4
- control methods, 6-9
- Create Model View dialog box, A-5
- Create Request dialog box, A-3
- Create Web Service dialog box, 2-12

D

- data exchange
 - partner profiles and, A-7
- database connections

Index-2

- opening, B-4
- databases
 - connecting to, B-4
 - Oracle, B-4
- deploying adapters, B-1 to B-2
- Description field, 2-2, 2-13, 6-3 to 6-5, 6-7, A-11
- design time, 6-8
- Destination parameter, A-9
- disconnecting from MySAP ERP, 2-8
- Display and maintain RFC Destination
 - window, A-11
- Display Distribution Model window, A-4
- Display IMG window, A-2
- DNS Lookup option, 6-7
- DNS name, 6-7
- Domain Name System (DNS), 6-7
- domain names, 6-7
- Domain option, 6-7

E

- EDI Version parameter, 2-4
- editing channels, 2-19
- editing targets, 2-9
- EJB (Enterprise Java Beans), B-1
- Enterprise Connector for J2EE Connector Architecture (J2CA), 1-4
- Enterprise Java Beans (EJB), B-1
- error messages, 7-2 to 7-7
 - target systems and, 7-5
- event adapters
 - configuring, 2-15
 - testing, A-12
- event handling, A-1, A-9
- Event Manager, A-9
- events, 1-1, 2-15, A-1
 - configuring, 2-15
 - creating, A-9
- Execution Denied list, 6-6
- Execution Granted list, 6-6
- Existing Service Names list, 2-13

F

- fault code elements, 7-5
- fault string elements, 7-5
- Function Builder, A-13
- functions, A-9

G

- Gateway Server, A-10
- Grant Access check box, 6-7
- Group (of Computers) option, 6-7
- Group node, 6-4
- groups
 - creating, 6-3

I

- IDoc (Intermediate Documents), 7-4, A-1

- defining, A-1
- registering, A-11
- saving, A-1
- transferring, A-1
- IDoc interface, A-1
- IDoc requests, 1-4
- in, 2-18
- inbound IDoc
 - configuring, A-1
- inbound MySAP ERP processing, A-1
 - configuring, A-1
- Inbound options tab, A-8
- input XML documents, 7-4
- installation directories, B-4
- instances of policy types, 6-1
- Intermediate Documents (IDoc), 7-4, A-1
 - defining, A-1
 - registering, A-11
 - saving, A-1
 - transferring, A-1
- IP (Mask)/Domain field, 6-7
- IP addresses, 6-7
- IP and Domain Restriction policy type, 6-6
- IWayConfig parameter, B-4
- IWayHome parameter, B-4
- IWayRepoPassword parameter, B-4
- IWayRepoURL parameter, B-4
- IWayRepoUser parameter, B-4

J

- J2CA (Enterprise Connector for J2EE Connector Architecture), 1-4
 - OracleAS Adapter and, B-1
- J2CA configuration
 - Application Explorer and, B-2
- J2CA repositories
 - migrating, 6-11
- J2CA resource adapters, 1-1, B-1
- Java program clients, B-1

L

- Language parameter, 2-3, 2-17
- librfc32.dll file, 7-2
- License and Method dialog box, 2-13
- License field, 2-13
- licenses, 2-13
- list of nodes, 7-2
- listeners, 1-4, A-10
- listeners. *See also* channels
- load balancing, A-10
- load balancing algorithms, A-10
- log files, 7-1
- log levels
 - overwriting, B-4 to ??
- logical systems, A-1 to A-6
 - configuring, A-1
 - defining, A-2
 - unique IDs, A-1 to A-2

- Loglevel parameter, B-4 to ??
- Logon ticket (SS02) parameter, 2-6
- Logon ticket (X509) parameter, 2-6
- Log.System column, A-3

M

- managed connector factories, B-2
- managed connector factory objects, B-2
 - multiple, B-4
- ManagedConnectionFactory parameter, B-4 to ??
- Message Server dialog box, 2-16
- Message Type field, A-8
- messages, 1-1
- metadata
 - storing, 6-8
- Method Name field, 2-13
- methods, 6-1
- migrating repositories, 6-8
- migrating Web services, 6-10
- My SAP ERP System port, A-1
- MySAP ERP
 - connecting to, 1-3, 2-1 to 2-8, 7-2, 7-6
 - disconnecting from, 2-8
 - integrating, 1-3
- MySAP ERP business functions, 2-2, 2-10 to 2-13
 - creating schemas for, 2-10
- MySAP ERP business processes, 1-3
- MySAP ERP event adapters
 - configuring, A-14
 - testing, A-13
- MySAP ERP event handling, A-9 to A-10
- MySAP ERP Event Manager, A-9
- MySAP ERP function modules
 - routing, A-9
- MySAP ERP Gateway server, A-9
- MySAP ERP inbound processing, A-1
- MySAP ERP outbound processing, A-9
- MySAP ERP platforms, 1-3
- MySAP ERP program ID (identifiers)
 - BAPI and, A-10
 - multiple remote function calls and, A-10
 - registering, A-11 to A-12
 - RFC Destination and, A-9
- MySAP ERP Trace parameter, 2-3, 2-17

N

- Name field, 2-2, 6-3 to 6-5
- New Group dialog box, 6-4
- New Policy permissions dialog box, 6-6
- New User dialog box, 6-3
- Node list, 7-2
- nodes
 - Business Services, 6-2 to 6-5
 - Channels, 2-16
 - Configuration under Business Services, 6-2 to 6-5, 6-7
 - connected, 2-8
 - disconnected, 2-9

- Group, 6-4
- Policies, 6-5
- Security, 6-5, 6-7
- Users, 6-3
- Users and Groups, 6-2 to 6-3

O

- Object type, 7-4
- OC4J-ra.xml file, B-2 to B-4
- Oracle Application Server
 - deployment of Connector to, B-1
- Oracle databases, B-4
- Oracle repositories
 - migrating, 6-8
- OracleAS Adapter
 - installation directory and, B-4
 - J2CA and, B-1
- OracleAS Adapter Application Explorer. *See* Application Explorer
- OracleAS Adapter Business Services Engine (BSE), 1-1, 1-4
 - troubleshooting, 7-4
- OracleAS Adapter for MySAP ERP
 - configuring, 2-1 to 2-19
 - deploying, 1-1
 - troubleshooting, 7-1 to 7-7
- OracleAS Adapter J2CA
 - Application Explorer and, 7-4
 - connecting to, 7-4
- outbound MySAP ERP processing, A-1 to A-9
 - configuring, A-9

P

- parameter types
 - channel configuration, 2-19
 - configuration, B-4
 - connection, 2-17 to 2-18, 7-2
 - repository migration, 6-9
- partner profiles
 - data exchange and, A-7
 - defining, A-7
- Partner profiles window, A-7
- Partner type LS, A-7
- Partn.number field, A-7
- Password parameter, 2-3, 2-17, 6-3
- passwords, 2-3, 2-17, 6-3, B-4 to ??
- permissions, 6-1
 - configuring, 7-4
 - denying, 6-6
 - granting, 6-6
- policies, 6-1
 - applying, 6-1
 - creating, 6-4
- Policies node, 6-5
- policy types
 - instances of, 6-1
 - IP and Domain Restriction, 6-6
- policy-based security, 6-1 to 6-8

- port definitions, A-1
- privileges, 6-1
 - setting, 6-1
- Process code field, A-8
- Processing by function module area, A-9
- Program field, A-12
- program ID (identifiers)
 - BAPI and, A-10
 - multiple remote function calls and, A-10
 - registering, A-11 to A-12
 - RFC Destination and, A-9
- Prompt for Workbench request dialog box, A-3
- properties, B-4
- pseudo events
 - Remote Function Calls (RFC) and, A-9

R

- registering program ID (identifiers), A-11 to A-12
- Remote Function Calls (RFC)
 - pseudo events and, A-9
 - registering, A-11
- Remote Function module, 7-4
- repositories
 - migrating, 6-8
- repository migration parameters, 6-9
- repository URLs, 6-10
- repository.xml file, 6-11
- request processing
 - BAPI, 1-4
 - IDoc, 1-4
 - RFC, 1-4
- request schemas, 2-10
- requests
 - executing, 7-4
- resource adapters, B-1 to B-2
 - deploying, 1-1
- Resource Execution policy type, 6-1
- response schemas, 2-10
- RFC (Remote Function Calls)
 - pseudo events and, A-9
 - registering, A-11
- RFC API
 - RFC calls and, A-9
- RFC Client program, A-9
- RFC Destination, A-9 to A-11
- RFC Destination field, A-11
- RFC Destination window, A-11
- RFC functions, A-9
- RFC programs, A-9
 - Client, A-9
 - Server, A-9
- RFC requests, 1-4
- RFC target systems, A-13
- runtime, 6-8

S

- SAP GUI, A-11, A-14
- sapjco.jar file, 7-2 to 7-3

- sapjcorfc.dll file, 7-2
- schemas, A-10
 - Application Explorer and, 2-10
 - creating, 2-10
- SE37 transaction, A-12
- security, 6-1 to 6-8
 - Application Explorer and, 6-1, 6-6
 - configuring, 6-2
- Security node, 6-2 to 6-5, 6-7
- security policies
 - applying, 6-1
 - creating, 6-4
- Security Policy option, 6-6
- Security tab, 2-6
- Selected list, 6-4 to 6-5
- server programs
 - RFC Client, A-9
 - RFC Server, A-9
- service names, 2-13
- services, 1-1
 - creating, 2-12 to 2-13
 - testing, 2-14
- servlets, B-1
- Single (Computer) option, 6-7
- SM59 transaction, A-9 to A-11
- SNC level parameter, 2-6
- SNC library path parameter, 2-6
- SNC mode parameter, 2-6
- SNC name parameter, 2-6
- SNC partner parameter, 2-6
- SOAP agents, 7-5
- SOAP envelopes, 6-9
- SOAP faults, 7-5
- soap operation name dialog box, 6-9
- SOAP requests, 6-1, 6-6, 7-5 to 7-7
 - creating, 6-8
 - errors and, 7-5 to 7-6
- storing metadata, 6-8
- System Number parameter, 2-4
- System tab, 2-4, 2-17

T

- target systems
 - errors and, 7-5
- Target Type list, 2-2
- targets
 - connecting to, 2-8, 7-6
 - defining, 2-2
 - deleting, 2-8
 - disconnecting from, 2-8
 - editing, 2-9
- Test Function Module Initial Screen window, A-13
- testing event adapters, A-13
- testing Web services, 2-14
- trace information, 7-1
- transaction processing, 1-1
- trigger by background program option, A-9
- trigger immediately option, A-9
- troubleshooting, 7-1 to 7-7

- Application Explorer, 7-2 to 7-3
- BSE, 7-4
- Web services, 7-4 to 7-6
- Type list, 6-7

U

- unique IDs, A-1
- updating adapter configuration, B-2
- User parameter, 2-3, 2-17, 7-4
- User tab, 2-3, 2-17
- users
 - associating, 6-2
- Users and Groups node, 6-2 to 6-3
- Users node, 6-3

W

- Web Service Definition Language (WSDL), 2-10 to 2-11, ?? to 2-12
- Web service names, 2-13
- Web services, 1-1
 - creating, 2-12 to 2-13
 - deploying, 6-1, 7-5 to 7-7
 - migrating, 6-10
 - testing, 2-14
 - troubleshooting, 7-4 to 7-6
- Web services policy-based security, 6-1 to 6-8
- WSDL (Web Service Definition Language), 2-10 to 2-11, ?? to 2-12
- WSDL file location dialog box, 6-8

X

- XML documents
 - input, 7-4
- XML messages, 1-1
- XML schemas
 - creating, 2-10

