Oracle® Application Server

Adapter for Oracle Applications User's Guide

10*g* (10.1.3.5.0)

Part No. E14293-01

August 2009

Oracle Application Server Adapter for Oracle Applications User's Guide, 10*g* (10.1.3.5.0)

Part No. E14293-01

Copyright © 2005, 2009, Oracle and/or its affiliates. All rights reserved.

Primary Author: David Weld, Melody Yang

Contributing Author: Sravani Bheemireddy, Neeraj Chauhan, Vimmika Dinesh, Anil Kemisetti, Nagaraj Ravinuthala, Nadakuditi Ravindra, Gautam Satpathy, Veshaal Singh, Sheela Vasudevan

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

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS
Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

This software and documentation may provide access to or information on content, products and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third party content, products and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third party content, products or services.

Contents

4 Using XML Gateway for BPEL Process Integration

5 Using Business Events for BPEL Process Integration

Send Us Your Comments

Oracle Application Server Adapter for Oracle Applications User's Guide, 10*g* (10.1.3.5.0)

Part No. E14293-01

Oracle welcomes customers' comments and suggestions on the quality and usefulness of this document. Your feedback is important, and helps us to best meet your needs as a user of our products. For example:

- Are the implementation steps correct and complete?
- Did you understand the context of the procedures?
- Did you find any errors in the information?
- Does the structure of the information help you with your tasks?
- Do you need different information or graphics? If so, where, and in what format?
- Are the examples correct? Do you need more examples?

If you find any errors or have any other suggestions for improvement, then please tell us your name, the name of the company who has licensed our products, the title and part number of the documentation and the chapter, section, and page number (if available).

Note: Before sending us your comments, you might like to check that you have the latest version of the document and if any concerns are already addressed. To do this, access the new Applications Release Online Documentation CD available on My Oracle Support and www.oracle.com. It contains the most current Documentation Library plus all documents revised or released recently.

Send your comments to us using the electronic mail address: appsdoc_us@oracle.com

Please give your name, address, electronic mail address, and telephone number (optional).

If you need assistance with Oracle software, then please contact your support representative or Oracle Support Services.

If you require training or instruction in using Oracle software, then please contact your Oracle local office and inquire about our Oracle University offerings. A list of Oracle offices is available on our Web site at www.oracle.com.

Preface

Intended Audience

Welcome to the *Oracle Application Server Adapter for Oracle Applications User's Guide*, 10*g* (10.1.3.5.0)

This documentation is written for the technical consultants, implementers and system integration consultants who use OracleAS Adapter for Oracle Applications.

This guide assumes you have a working knowledge of the following:

- The principles and customary practices of your business area.
- Oracle E-Business Suite.
- Oracle BPEL Process Manager.
- Oracle JDeveloper.
- Oracle Database, Oracle Application Server, and PL/SQL technology.
- EBS Integration Interfaces.
- Oracle integration technologies, including Web services, WSDL, XML Gateway, EDI Gateway, and the Business Event System.
- B2B, A2A and BP integrations.

If you have never used these products, Oracle suggests that you attend training classes available through Oracle University.

See Related Information Sources on page xi for more Oracle Applications product information.

Deaf/Hard of Hearing Access to Oracle Support Services

To reach Oracle Support Services, use a telecommunications relay service (TRS) to call Oracle Support at 1.800.223.1711. An Oracle Support Services engineer will handle technical issues and provide customer support according to the Oracle service request process. Information about TRS is available at http://www.fcc.gov/cgb/consumerfacts/trs.html, and a list of phone numbers is available at http://www.fcc.gov/cgb/dro/trsphonebk.html.

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.

Structure

Related Information Sources

You can choose from many sources of information, including online documentation, training, and support services, to increase your knowledge and understanding of OracleAS Adapter for Oracle Applications.

Documentation

You may want to refer to other Oracle Application Server guides when you set up and use OracleAS Adapter for Oracle Applications. You can read the guides online by reading from the Oracle Application Server or Oracle Database Documentation Library CD included in your media pack, or on the Oracle Technology Network (OTN) [http://www.oracle.com/technology/documentation/appserver.html]. If you require printed guides, you can purchase them from the Oracle Store [http://oraclestore.oracle.com].

To download free release notes, installation documentation, white papers, or other collateral, please visit the main OTN page [http://www.oracle.com/technology/]. You must register online before using OTN; registration is free and can be done at the OTN registration page [http://www.oracle.com/technology/membership/].

Training

Oracle offers a complete set of training courses to help you and your staff master Oracle Application Server 10*g* and reach full productivity quickly. These courses are organized into functional learning paths, so you take only those courses appropriate to your job or area of responsibility.

You have a choice of educational environments. You can attend courses offered by Oracle University at any one of our many Education Centers, you can arrange for our trainers to teach at your facility, or you can use Oracle Learning Network (OLN), Oracle University's online education utility.

> **Tip:** For information about upcoming instructor-led training, please refer to Oracle University's course offerings [http://education.oracle.com/].

In addition, Oracle training professionals can tailor standard courses or develop custom courses to meet your needs. For example, you may want to use your organization's structure, terminology, and data as examples in a customized training session delivered at your own facility.

Support

From on-site support to central support, our team of experienced professionals provides the help and information you need to keep Oracle Application Server 10*g* working for you. This team includes your Technical Representative, Account Manager, and Oracle's large staff of consultants and support specialists, with expertise in your business area, managing an Oracle Database, and your hardware and software environment.

Do Not Use Database Tools to Modify Oracle Applications Data

Oracle STRONGLY RECOMMENDS that you never use SQL*Plus, Oracle Data Browser, database triggers, or any other tool to modify Oracle Applications data unless otherwise instructed.

Oracle provides powerful tools you can use to create, store, change, retrieve, and maintain information in an Oracle database. But if you use Oracle tools such as SQL*Plus to modify Oracle Applications data, you risk destroying the integrity of your data and you lose the ability to audit changes to your data.

Because Oracle Applications tables are interrelated, any change you make using an Oracle Applications form can update many tables at once. But when you modify Oracle Applications data using anything other than Oracle Applications, you may change a row in one table without making corresponding changes in related tables. If your tables get out of synchronization with each other, you risk retrieving erroneous information and you risk unpredictable results throughout Oracle Applications.

When you use Oracle Applications to modify your data, Oracle Applications automatically checks that your changes are valid. Oracle Applications also keeps track of who changes information. If you enter information into database tables using database tools, you may store invalid information. You also lose the ability to track who has changed your information because SQL*Plus and other database tools do not keep a record of changes.

1

Introduction to OracleAS Adapter for Oracle Applications

This chapter covers the following topics:

- Overview of OracleAS Adapter for Oracle Applications
- New Features in This Release

Overview of OracleAS Adapter for Oracle Applications

Oracle Applications is a set of integrated business applications that runs entirely on the Internet. Oracle Applications offers you the following:

- Reduced costs
- Increased value across front-office and back-office functions
- Access to current, accurate, and consistent data

Oracle Applications are built on a unified information architecture that consolidates data from Oracle and non-Oracle applications and enables a consistent definition of customers, suppliers, partners, and employees across the entire enterprise. This results in a suite of applications that can give you information, such as current performance metrics, financial ratios, profit and loss summaries. To connect Oracle Applications to non-Oracle applications, you use OracleAS Adapter for Oracle Applications.

OracleAS Adapter for Oracle Applications not only provides comprehensive, bidirectional, multimodal, synchronous, and asynchronous connectivity to Oracle Applications, but also supports for all modules of Oracle E-Business Suite in Release 12 and Release 11*i* including custom integration interfaces in various versions of Oracle E-Business Suite.

    - Please note that Adapter for Oracle Applications is also informally known as Oracle E-Business Suite Adapter.

- The support for various versions of Oracle E-Business Suite has the following conditions:
  - OracleAS Adapter for Oracle Applications supports only those versions of Oracle E-Business Suite Release 11*i* which work with OWF.G.Rollup 7 applied.
  - OracleAS Adapter for Oracle Applications 10.1.3.3 onwards supports Oracle E-Business Suite Release 12.0.
  - OracleAS Adapter for Oracle Applications 10.1.3.5 onwards supports Oracle E-Business Suite Release 12.1.
  - To enable the native Oracle E-Business Suite connectivity using J2EE data sources feature, the minimum requirement for Oracle E-Business Suite Release 11*i* is 11*i*.ATG_PF.H.Delta.6 (RUP6) and for Oracle E-Business Suite Release 12 is Release 12.0.4.

    See Secured Connection Between Oracle E-Business Suite and Oracle SOA Suite Using J2EE Data Source Implementation, page 3-22.

Major Features

OracleAS Adapter for Oracle Applications provides the following features:

- It supports open standards, including J2EE Connector Architecture (J2CA), Extensible Markup Language (XML), Web Service Invocation Framework (WSIF), Web Service Inspection Language (WSIL), and Web Service Definition Language (WSDL).
- It supports the widest range of integration interface types. They are PL/SQL APIs, Business Events, Open Interface Tables, Concurrent Programs, XML Gateway Interfaces, e-Commerce Gateway Interface, and Interface Views.
- It generates adapter metadata as WSDL files with J2CA extension.

  Note: See *Oracle Application Server Adapter Concepts* on OTN for more information.

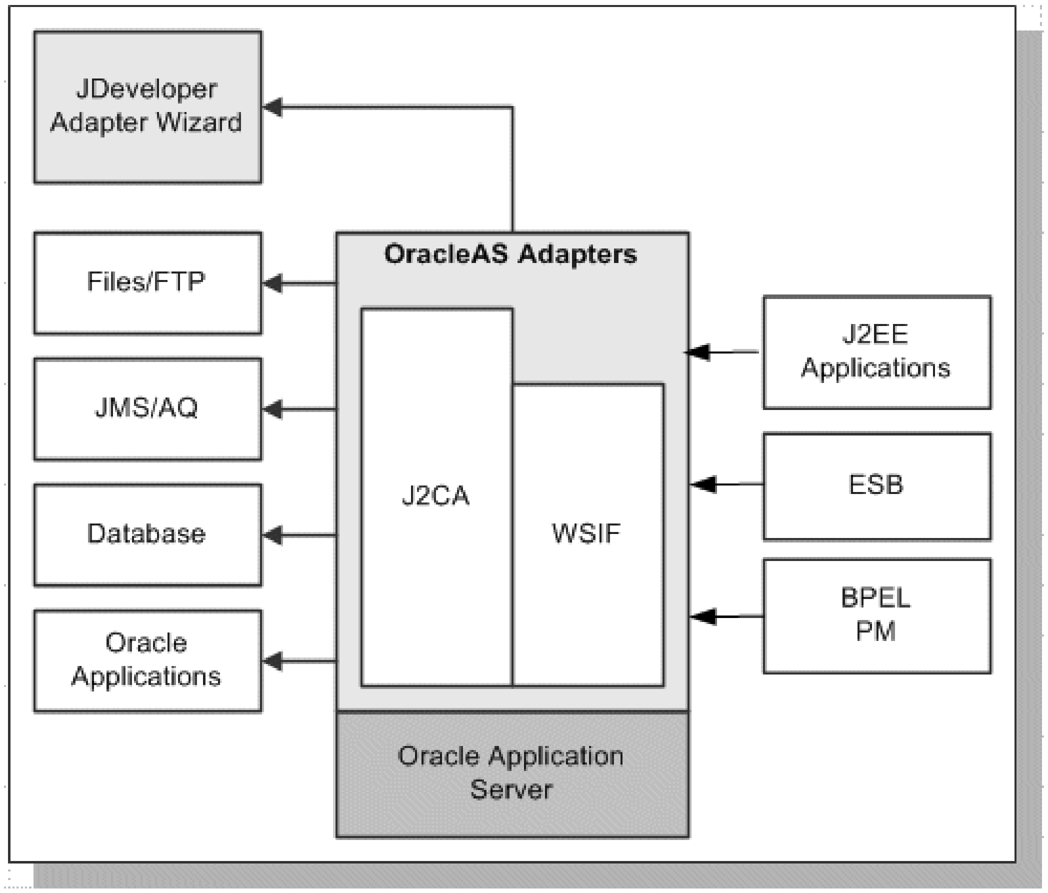
- It works under the securely configured connection between Oracle Applications and Oracle Application Server using just the FND username and password (concept of Oracle Applications username and password) for authentication.

- It leverages and supports Oracle User Management function security feature to allow only authorized users to access and execute APIs that they are exposed as Web services to update Oracle Applications.
- It implicitly takes care of applications context without bothering about the complexities of invoking the same explicitly.
- It supports multiple languages and multiple organization access control (MOAC) setups based on the concept of applications context.
- It uses a JDeveloper based design-time tool for dynamically browsing the Oracle Applications interface and configuring the adapter metadata. The design-time tasks are wizard driven, user-friendly, and intuitive providing superior user experiences.
- It provides the global transaction control support implementing two-phase commit by leveraging the underlying JCA standards compliant framework.
- It supports multiple versions of Oracle E-Business Suite from the same instance of Adapter at design time.

Architecture

OracleAS Adapter for Oracle Applications is based on J2CA 1.0 standards and deployed as a resource adapter in the same Oracle Application Server Containers for J2EE (OC4J) container as BPEL Process Manager. The architecture of OracleAS Adapter for Oracle Applications is similar to the architecture of technology adapters.

OracleAS Adapter for Oracle Applications Architecture



Installing OracleAS Adapter for Oracle Applications

The installation of the Oracle Application Server Adapter for Oracle Applications happens out-of-the-box with the Oracle BPEL Process Manager (PM) product. OracleAS Adapter for Oracle Applications is deployed using the Oracle BPEL PM in Oracle JDeveloper.

> **Note:** Refer to the "Notes on Installing Oracle BPEL Process Manager" section, *Oracle Application Server Integration Business Activity Monitoring User's Guide* for more details about installing Oracle BPEL Process Manager.

Integration with Oracle BPEL Process Manager

Based on the service-oriented architecture (SOA), Oracle BPEL Process Manager (BPEL PM) provides a comprehensive solution for creating, deploying, and managing Oracle

BPEL Process Manager business processes.

OracleAS Adapter for Oracle Applications can easily expose public integration interface within Oracle E-Business Suite as standard Web services. These services can be created and configured in the Oracle JDeveloper at design time either using BPEL Designer or ESB Designer. At run time, Oracle E-Business Suite integration flows are deployed in the BPEL PM Server / ESB Server for execution of the services to complete the integration.

Design Time

During design time, the integration interface is exposed as a Web service represented by WSDL and schema files. This Web service participates in the BPEL PM orchestration process via the process activities like Invoke, Assign, Transform, and other activities if necessary through simple configuration steps.

The Oracle JDeveloper BPEL Designer, a graphical drag and drop environment, is used to design BPEL-based process flows and Web services orchestration.

When you create a partner link in JDeveloper BPEL Designer, the Adapter Configuration Wizard starts which enables you to select and configure the OracleAS Adapters for Oracle Applications. With proper database and service connection setups, you can select an interface in or out from Oracle E-Business Suite and add the XML schema. When configuration is complete, the wizard generates a WSDL file corresponding to the XML schema for the partner link.

Additional process activities are added to the BPEL process if necessary to assign parameters and invoke the service.

Run Time

Before deploying the BPEL process to Oracle BPEL server, necessary run-time parameters, system settings, and application setup must be properly configured for a successful deployment and service integrations.

> **Note:** Agent Listeners should also be up and running if needed for certain interface types used in Oracle E-Business Suite.

While deploying the BPEL process, the WSIF Provider converts the Web service invocation from BPEL PM `Invoke` activity to an outbound interaction call and performs the reverse conversion in the other direction as a Web service response for a synchronous request-response message pattern. The WSIF Provider also supports the one-way asynchronous outbound interaction invocation such as integration with XML Gateway outbound message maps and outbound business events.

Testing the BPEL Process at Run Time

After deploying the BPEL process, you should validate the design by testing the deployed BPEL process. It can be done by manually initiating the BPEL process from the BPEL Console to test the interface integration contained in your BPEL process.

For detailed design-time and run-time tasks for each integration interface, see the individual interface chapter explained later in this book.

New Features in This Release

This section describes the new features that have been added in OracleAS Adapter for Oracle Applications 10*g* (10.1.3.5.0).

Additional Support for Header Variables

Header variables are used to provide applications context information required in the SOA Suite to process Concurrent Programs and PL/SQL APIs.

Adapter for Oracle Applications can now accept the following three new header parameters along with the existing `Username`, `Responsibility`, and `Org_ID` parameters for setting applications context:

- `RespApplication`: Responsibility Application Key which needs to be used to set the context.
- `SecurityGroup`: Security Group Key which defaults to 'Standard' if not passed.
- `NLSLanguage`: It is the language code for the request. NLS context would be set to the language code that is passed if valid. It would default to 'US' if not passed.

> **Note:** Existing header parameter `Responsibility` used in the earlier releases can now take Responsibility Key as well as Responsibility Name as input. If the header parameter `NLSLanguage` is set, and Responsibility Name is passed, the value passed for `Responsibility` is expected to be in the same language. However, Responsibility Key as well as all other header parameters are language independent.
>
> All these header parameters would be used together to set the applications context. Alternatively, passing just the `Username` and `Responsibility` would work as it did in the earlier releases.

For more information, see Supporting for Applications Context Header Variables, page 3-8.

2

OracleAS Adapter for Oracle Applications Features

Overview

OracleAS Adapter for Oracle Applications enables you to orchestrate discrete data into a meaningful business process and creates Web services for various integration interface types within Oracle E-Business Suite. It plays the role of service provider for Oracle E-Business Suite to allow seamless integration between business partners, processes, applications, and end users in heterogeneous environment.

OracleAS Adapter for Oracle Applications provides the following features which are further discussed in this chapter:

- Support for Various Integration Interface Types, page 2-1.
- Support for Oracle Integration Repository, page 2-3
- Support for Custom Integration Interfaces in Various Versions of Oracle E-Business Suite, page 2-4

Support for Various Integration Interface Types

OracleAS Adapter for Oracle Applications acts as a highly flexible integration interface for Oracle Applications. It supports the following interface types for integrating with Oracle Applications:

- **PL/SQL APIs**

  These APIs enable you to insert and update data in Oracle Applications using PL/SQL.

- **Business Events**

A business event is an occurrence in an internet application that might be significant to other objects in a system or to external agents. An example of a business event can be the creation of a new sales order or changes to an existing order.

Oracle Workflow uses the Business Event System that leverages the Oracle Advanced Queuing (AQ) infrastructure to communicate and manage business events between systems. The Business Event System consists of an Event Manager and workflow process event activities. The Event Manager lets you register subscriptions to significant events; event activities representing business events within workflow processes let you model complex business flows or logics within workflow processes.

When a local event occurs, the subscribing code is executed in the same transaction as the code that raised the event. Subscription processing can include executing custom code on the event information, sending event information to a workflow process, and sending event information to other queues or systems.

- **Open Interface Tables**

  Interface tables enable you to insert or update data into Oracle Applications. The associated concurrent program should be running to move the data from the interface tables to base tables.

- **Concurrent Programs**

  Concurrent programs enable you to move data from interface tables to base tables or execute any application logic.

- **Oracle XML Gateway**

  XML Gateway enables bidirectional integration with Oracle Applications. It helps you to insert and retrieve data from Oracle Applications. XML Gateway is a higher-level interface that exposes OAGIS-formatted XML documents for commonly used Oracle Application business objects and business interfaces. XML Gateway integrates with interface tables, Oracle Workflow Business Event System (BES), and interface views to insert and retrieve data from Oracle Applications. It maps the underlying table data to XML and back.

- **Oracle e-Commerce (EDI) Gateway**

  Oracle e-Commerce Gateway provides a common, standards-based approach for Electronic Data Interchange (EDI) integration between Oracle Applications and third party applications.

- **Interface Views**

  Interface views help you to retrieve data from Oracle Applications using the application tables.

Please note that OracleAS Adapter for Oracle Applications also supports the following custom integration interface types that are exposed by the Oracle Applications Module Browser, not by Oracle Integration Repository:

- Customized PL/SQL APIs
- Customized Business Events

  Note: Business events integration interface type is also exposed by Oracle Applications Module Browser, not by Oracle Integration Repository.

- Customized XML Gateway Maps

Support for Oracle Integration Repository

Oracle Integration Repository, an integral part of Oracle E-Business Suite, is a prebuilt catalog of information about the numerous public integration interfaces delivered with Oracle Applications. It provides a comprehensive view of the integration interfaces with details for Oracle E-Business Suite. These interfaces are exposed because their definitions were annotated at design time as required by Oracle Integration Repository.

Oracle Integration Repository can only provide information about an integration interface that has been specifically annotated by the developer to make it public. OracleAS Adapter for Oracle Applications takes advantage of the annotations that have already been created to make the following integration interface types visible in the Oracle Applications Module Browser:

- XML Gateway message maps
- PL/SQL APIs
- Concurrent programs
- Open Interface tables
- Interface views
- e-Commerce Gateway EDI messages

These integration interfaces are exposed as Web services and are available for process orchestration through the Oracle BPEL Process Manager.

For more information about Oracle Integration Repository, see the Navigating Through Oracle Integration Repository chapter, *Oracle E-Business Suite Integrated SOA Gateway User's Guide*. This guide is part of the Oracle Applications documentation library. Oracle Applications documentation can be accessed with the following link:

http://www.oracle.com/technology/documentation/applications.html

Note: Oracle Integration Repository is integral part of Oracle E-Business Suite Release 12.0 onwards; however, it is available as hosted environment for the Release 11.5.10 version at `http://irep.oracle.com`.

Support for Custom Integration Interfaces in Various Versions of Oracle E-Business Suite

OracleAS Adapter for Oracle Applications leverages Integration Repository for Oracle E-Business Suite Release 11.5.10 and Release 12 as the source of truth for the integration content. However, the implementation is based on the version of Oracle E-Business Suite. For pre-Release 11.5.10 instances, OracleAS Adapter for Oracle Applications connects directly to the application database for information on integration interfaces. OracleAS Adapter for Oracle Applications also supports selecting custom integration interfaces and the design-time navigation steps to reach to these custom interfaces depending on the following versions of Oracle E-Business Suite:

- Release 12, page 2-5
- Release 11.5.10, page 2-5
- Pre-Release 11.5.10, page 2-6

Important: Please note that the support for various versions of Oracle E-Business Suite has the following conditions:

- OracleAS Adapter for Oracle Applications supports only those versions of Oracle E-Business Suite Release 11*i* which work with OWF.G.Rollup 7 applied.
- OracleAS Adapter for Oracle Applications version 10.1.3.3 onwards supports Oracle E-Business Suite Release 12.0.
- OracleAS Adapter for Oracle Applications 10.1.3.5 onwards supports Oracle E-Business Suite Release 12.1.
- To enable the native Oracle E-Business Suite connectivity using J2EE data sources feature, the minimum requirement for Oracle E-Business Suite Release 11*i* is 11*i*.ATG_PF.H.Delta.6 (RUP6) and for Oracle E-Business Suite Release 12 is 12.0.4 release.

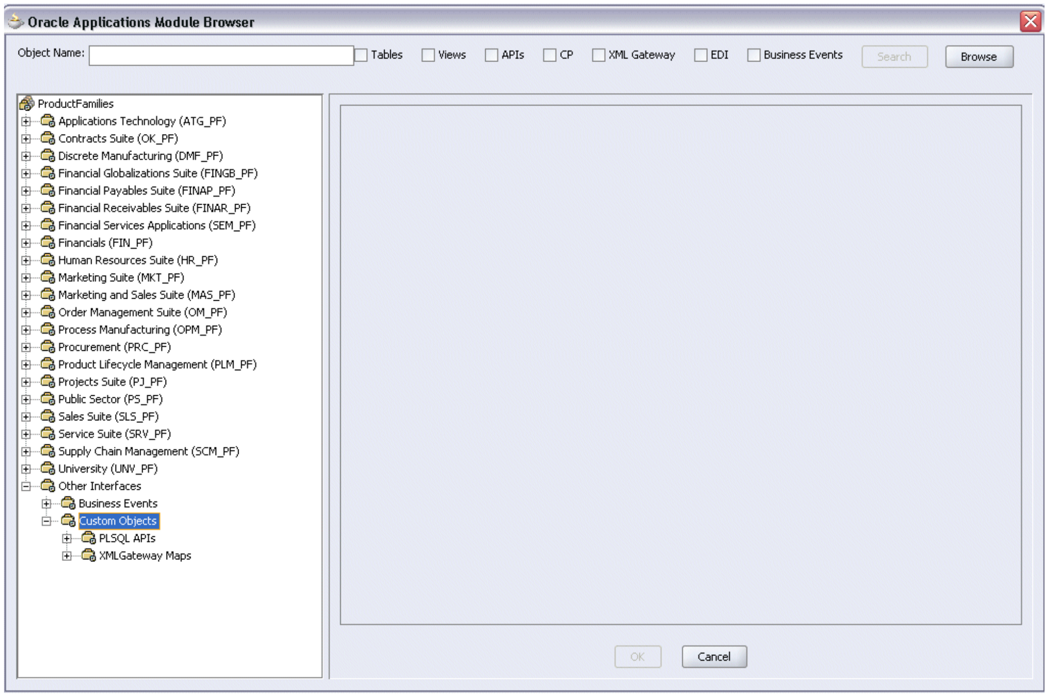
  See Secured Connection Between Oracle E-Business Suite and Oracle SOA Suite Using J2EE Data Source Implementation, page 3-22.

From the business service creation and run-time perspectives, Adapter for Oracle Applications supports customized PL/SQL APIs as far as the packages are available in the APPS schema. The Oracle Applications Module Browser can expose these customized PL/SQL APIs for integration purposes during the design time.

Support for Oracle E-Business Suite Release 12

From Release 12, Oracle Integration Repository is shipped as part of the Oracle E-Business Suite which enables OracleAS Adapter for Oracle Applications to directly connect to the live database of Oracle Integration Repository querying for the public interfaces and then displaying the list of customized PL/SQL APIs under the `Other Interfaces` node in the Oracle Applications Module Browser.

Supporting Custom Integration Interfaces in Release 12



- ProductFamilies
  - Applications Technology (ATG_PF)
  - Contracts Suite (OK_PF)
  - Discrete Manufacturing (DMF_PF)
  - Financial Globalizations Suite (FINGB_PF)
  - Financial Payables Suite (FINAP_PF)
  - Financial Receivables Suite (FINAR_PF)
  - Financial Services Applications (SEM_PF)
  - Financials (FIN_PF)
  - Human Resources Suite (HR_PF)
  - Marketing Suite (MKT_PF)
  - Marketing and Sales Suite (MAS_PF)
  - Order Management Suite (OM_PF)
  - Process Manufacturing (OPM_PF)
  - Procurement (PRC_PF)
  - Product Lifecycle Management (PLM_PF)
  - Projects Suite (PJ_PF)
  - Public Sector (PS_PF)
  - Sales Suite (SLS_PF)
  - Service Suite (SRV_PF)
  - Supply Chain Management (SCM_PF)
  - University (UNV_PF)
  - Other Interfaces
    - Business Events
    - Custom Objects
      - PLSQL APIs
      - XMLGateway Maps

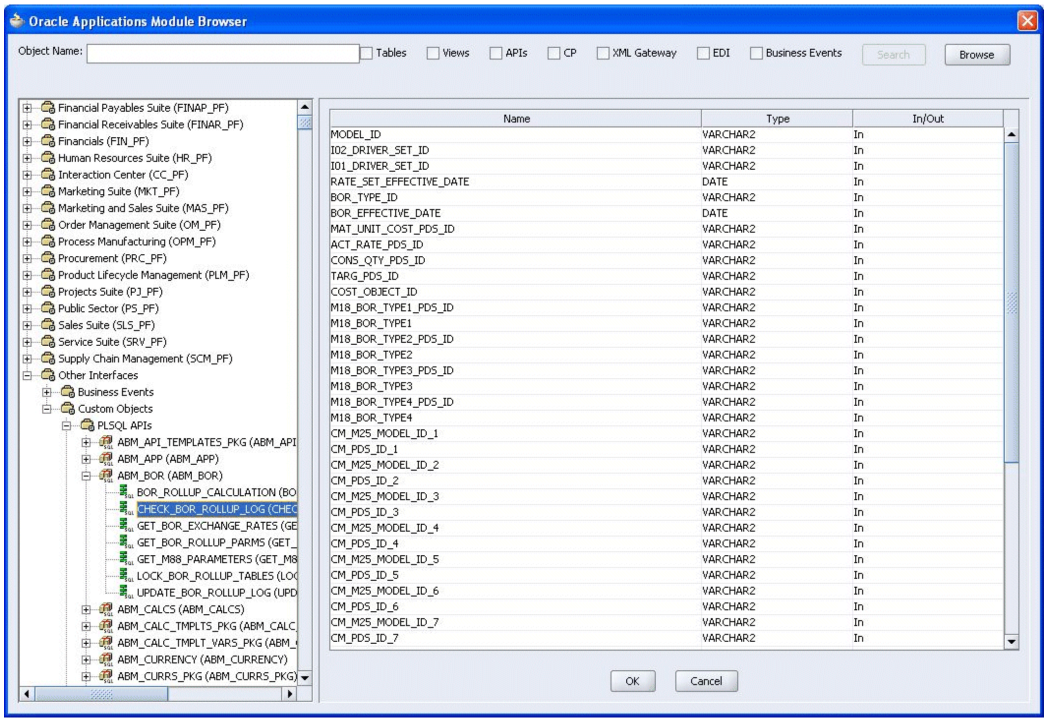
Please note that OracleAS Adapter for Oracle Applications allows you to extract the Integration Repository data file from the live database you connect to Oracle Applications and create a local copy of the Integration Repository data file in your workplace. Next time when you look for public interfaces, the system can retrieve data from the cache in your workplace.

For detailed information about connecting to Oracle E-Business Suite Release 12, please refer to the Creating a Partner Link or Adding a Partner Link design-time task for each integration interface.

Support for Oracle E-Business Suite Release 11.5.10

To support the Release 11.5.10 version of Oracle E-Business Suite, OracleAS Adapter for Oracle Applications provides the Integration Repository data file bundled as part of the product in xml format. At the design time, OracleAS Adapter for Oracle Applications queries public interfaces from the native XML data file of the Integration Repository located in the Adapter and displays the list of custom integration interfaces under the `Other Interfaces` node in the Oracle Applications Module Browser.

Supporting Custom Integration Interfaces in Release 11.5.10



```
Financial Payables Suite (FINAP_PF)
Financial Receivables Suite (FINAR_PF)
Financials (FIN_PF)
Human Resources Suite (HR_PF)
Interaction Center (CC_PF)
Marketing Suite (MKT_PF)
Marketing and Sales Suite (MAS_PF)
Order Management Suite (OM_PF)
Process Manufacturing (OPM_PF)
Procurement (PRC_PF)
Product Lifecycle Management (PLM_PF)
Projects Suite (PJ_PF)
Public Sector (PS_PF)
Sales Suite (SLS_PF)
Service Suite (SRV_PF)
Supply Chain Management (SCM_PF)
Other Interfaces
    Business Events
    Custom Objects
        PLSQL APIs
            ABM_API_TEMPLATES_PKG (ABM_API
            ABM_APP (ABM_APP)
            ABM_BOR (ABM_BOR)
                BOR_ROLLUP_CALCULATION (BO
                CHECK_BOR_ROLLUP_LOG (CHEC
                GET_BOR_EXCHANGE_RATES (GE
                GET_BOR_ROLLUP_PARMS (GET_
                GET_M88_PARAMETERS (GET_M8
                LOCK_BOR_ROLLUP_TABLES (LOC
                UPDATE_BOR_ROLLUP_LOG (UPD
            ABM_CALCS (ABM_CALCS)
            ABM_CALC_TMPLTS_PKG (ABM_CALC
            ABM_CALC_TMPLT_VARS_PKG (ABM_
            ABM_CURRENCY (ABM_CURRENCY)
            ABM_CURRS_PKG (ABM_CURRS_PKG)
```

Name	Type	In/Out
MODEL_ID	VARCHAR2	In
I02_DRIVER_SET_ID	VARCHAR2	In
I01_DRIVER_SET_ID	VARCHAR2	In
RATE_SET_EFFECTIVE_DATE	DATE	In
BOR_TYPE_ID	VARCHAR2	In
BOR_EFFECTIVE_DATE	DATE	In
MAT_UNIT_COST_PDS_ID	VARCHAR2	In
ACT_RATE_PDS_ID	VARCHAR2	In
CONS_QTY_PDS_ID	VARCHAR2	In
TARG_PDS_ID	VARCHAR2	In
COST_OBJECT_ID	VARCHAR2	In
M18_BOR_TYPE1_PDS_ID	VARCHAR2	In
M18_BOR_TYPE1	VARCHAR2	In
M18_BOR_TYPE2_PDS_ID	VARCHAR2	In
M18_BOR_TYPE2	VARCHAR2	In
M18_BOR_TYPE3_PDS_ID	VARCHAR2	In
M18_BOR_TYPE3	VARCHAR2	In
M18_BOR_TYPE4_PDS_ID	VARCHAR2	In
M18_BOR_TYPE4	VARCHAR2	In
CM_M25_MODEL_ID_1	VARCHAR2	In
CM_PDS_ID_1	VARCHAR2	In
CM_M25_MODEL_ID_2	VARCHAR2	In
CM_PDS_ID_2	VARCHAR2	In
CM_M25_MODEL_ID_3	VARCHAR2	In
CM_PDS_ID_3	VARCHAR2	In
CM_M25_MODEL_ID_4	VARCHAR2	In
CM_PDS_ID_4	VARCHAR2	In
CM_M25_MODEL_ID_5	VARCHAR2	In
CM_PDS_ID_5	VARCHAR2	In
CM_M25_MODEL_ID_6	VARCHAR2	In
CM_PDS_ID_6	VARCHAR2	In
CM_M25_MODEL_ID_7	VARCHAR2	In
CM_PDS_ID_7	VARCHAR2	In

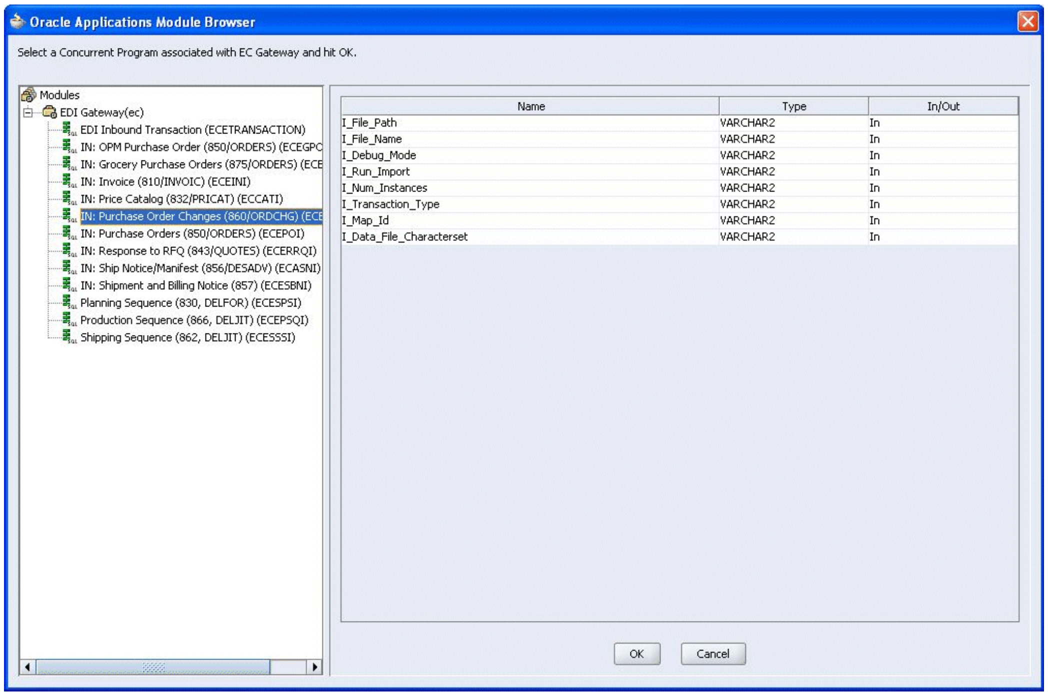
Support for Oracle E-Business Suite Pre-Release 11.5.10

To support the pre-Release 11.5.10 versions of Oracle E-Business Suite, OracleAS Adapter for Oracle Applications connects to the live application database for the integration information on all interface types. Since there is no differentiation between public, private, and customized PL/SQL APIs in the pre-Release 11.5.10 versions of Oracle E-Business Suite, OracleAS Adapter for Oracle Applications displays them all under the node of each module through Oracle Applications Module Browser.

Before making a selection from the browser for the pre-Release 11.5.10, you must select an interface type you want to use in the Adapter Configuration Wizard. All interfaces of your selected type will be displayed in the browser.

For example, you will find a list of concurrent programs associated with e-Commerce (EDI) Gateway displayed in the Oracle Application Module Browser as follows if the EDI Gateway interface type is selected.

Supporting Custom Integration Interfaces in pre-Release 11.5.10


Oracle Applications Module Browser

Select a Concurrent Program associated with EC Gateway and hit OK.

- Modules
  - EDI Gateway(ec)
    - EDI Inbound Transaction (ECETRANSACTION)
    - IN: OPM Purchase Order (850/ORDERS) (ECEGPC
    - IN: Grocery Purchase Orders (875/ORDERS) (ECE
    - IN: Invoice (810/INVOIC) (ECEINI)
    - IN: Price Catalog (832/PRICAT) (ECCATI)
    - IN: Purchase Order Changes (860/ORDCHG) (ECE
    - IN: Purchase Orders (850/ORDERS) (ECEPOI)
    - IN: Response to RFQ (843/QUOTES) (ECERRQI)
    - IN: Ship Notice/Manifest (856/DESADV) (ECASNI)
    - IN: Shipment and Billing Notice (857) (ECESBNI)
    - Planning Sequence (830, DELFOR) (ECESPSI)
    - Production Sequence (866, DELJIT) (ECEPSQI)
    - Shipping Sequence (862, DELJIT) (ECESSSI)

Name	Type	In/Out
I_File_Path	VARCHAR2	In
I_File_Name	VARCHAR2	In
I_Debug_Mode	VARCHAR2	In
I_Run_Import	VARCHAR2	In
I_Num_Instances	VARCHAR2	In
I_Transaction_Type	VARCHAR2	In
I_Map_Id	VARCHAR2	In
I_Data_File_Characterset	VARCHAR2	In

When you make a selection through the module browser at design time, OracleAS Adapter for Oracle Applications validates your selected API against the database. If it exists in the database for a particular version of your instance, then the associated WSDL file will be generated successfully.

3

OracleAS Adapter for Oracle Applications Concepts

This chapter covers the following topics:

- Understanding Applications Context
- Understanding OracleAS Adapter for Oracle Applications Security
- Secured Connection Between Oracle E-Business Suite and Oracle SOA Suite Using J2EE Data Source Implementation
- Understanding the Oracle Applications Module Browser

Understanding Applications Context

Applications context is required for secured transaction processing into and out of Oracle Applications.

Applications context is a combination of **Username**, **Responsibility**, **Responsibility Application**, **NLS Language**, **Security Group**, and **Organization ID**.

To establish applications context, the Organization ID is implicitly derived from the Oracle Applications setup data.

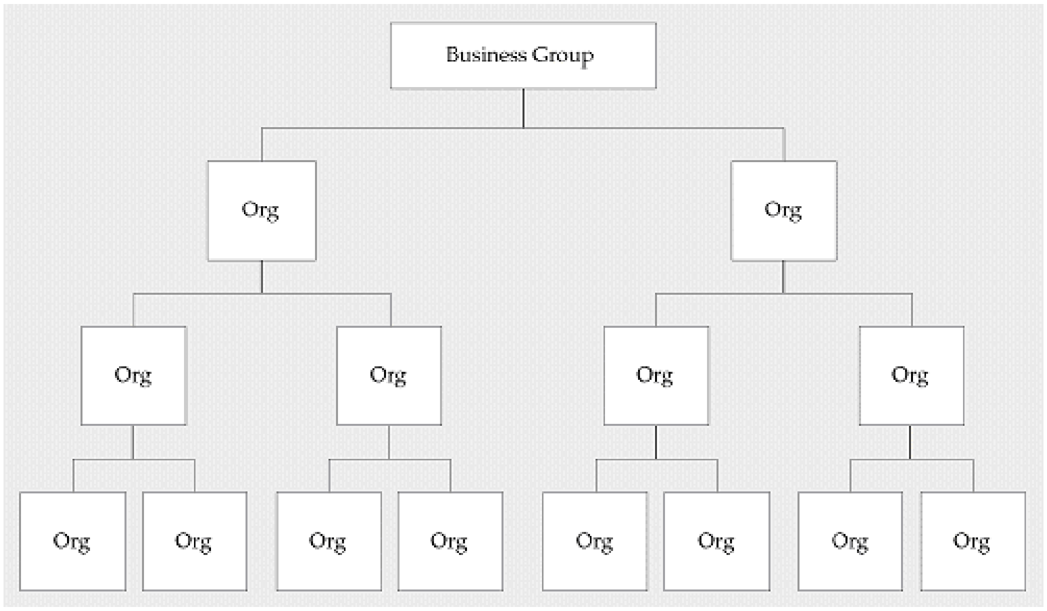
To understand applications context, you need to first understand how Organization ID and multiple organizations are related.

Applications Context in Multiple Organizations

You can define multiple organizations and the relationships between them in a single installation of Oracle Applications. These organizations can be sets of books, business groups, legal entities, operating units, or inventory organizations.

Multilevel organization hierarchies can be defined with a business group at the top of each hierarchy. When you define new organizations, they are automatically assigned to the business group associated with your current session. Each organization is part of a business group. The business group is usually the top box on an enterprise organization chart.

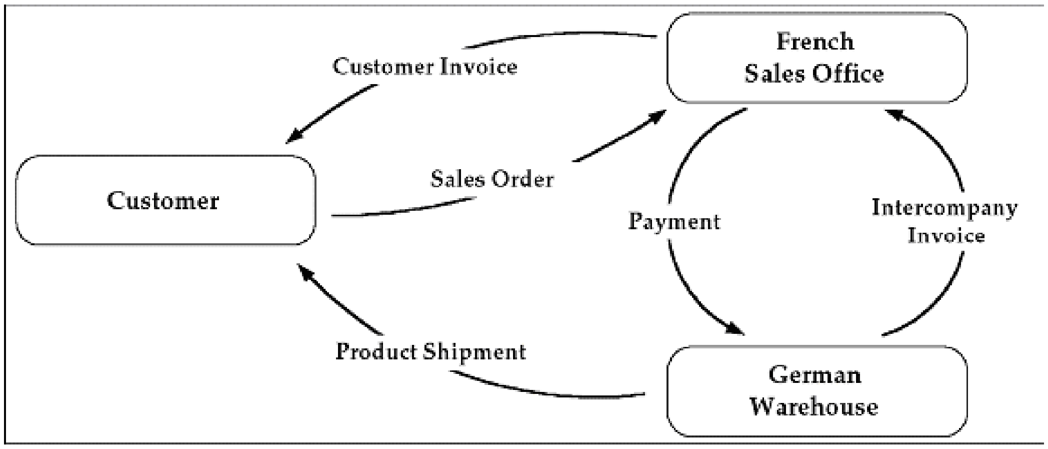
Business Group Hierarchy



Example of a Multiple-Organization Setup

Using the accounting, distribution, and materials management functions in Oracle Applications, you define the relationships among inventory organizations, operating units, legal entities, and sets of books to create a multilevel company structure, as shown in the following diagram.

A Multiple-Organization Transaction

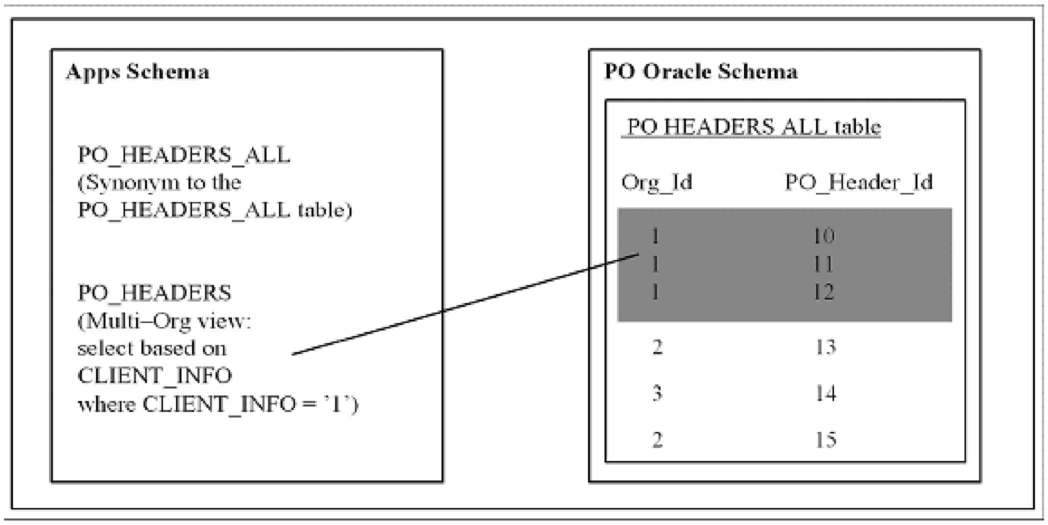


Consider two different organizations in your company: One is a French sales office and the other is a German warehouse. There is a sales order transaction with the customer, and this illustrates how the entire Order-to-Deliver process would work:

1. The customer places a sales order with the French sales office.
2. The German warehouse delivers the product shipment to the customer.
3. The German warehouse issues an inter-company invoice to the French sales office.
4. The French sales office makes the inter-company payment to the German warehouse.
5. The French sales office sends the customer invoice to the customer.
6. The customer makes payment to the French sales office.

The database architecture is the same for a multiple-organization and non-multiple-organization installation, and uses the standard install tools feature that automatically creates synonyms in the APPS schema for each base product table and defines these synonyms with the same name as the base product tables. For example, the PO Oracle schema has a table named PO_HEADERS_ALL and the APPS schema has a corresponding synonym of the same name, PO_HEADERS_ALL. The PO_HEADERS_ALL synonym can be used to access unpartitioned data.

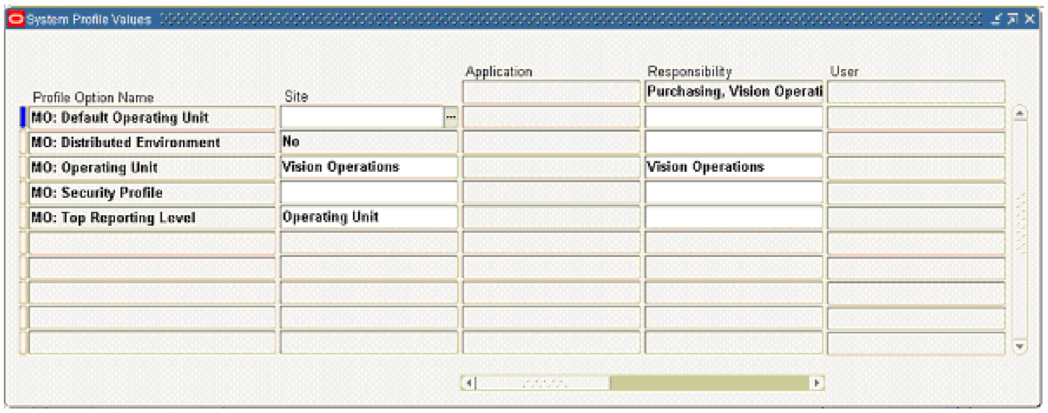
Schema Synonyms



Multi-Organization Access Control (MOAC) Security by Operating Units

While setting up the system profile values, the username and responsibility are tied up with the organization or operating units.

Multiple-Organization System Profiles



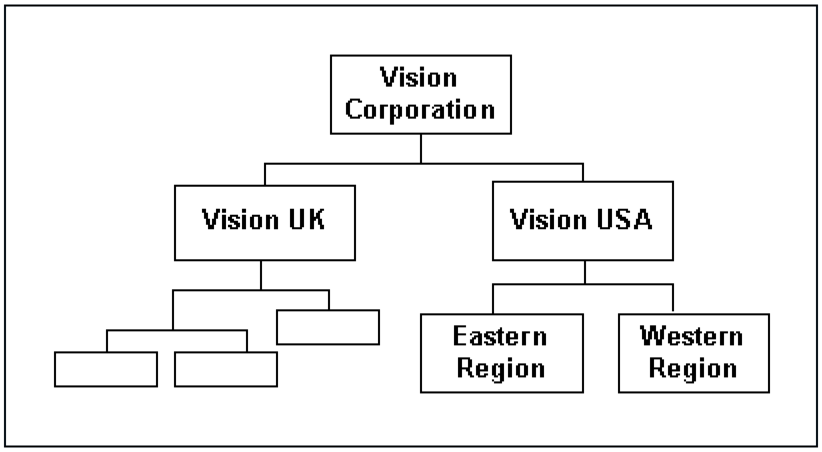
Profile Option Name	Site	Application	Responsibility	User
			Purchasing, Vision Operati	
MO: Default Operating Unit				
MO: Distributed Environment	No			
MO: Operating Unit	Vision Operations		Vision Operations	
MO: Security Profile				
MO: Top Reporting Level	Operating Unit			

To have a secured way for users to only access or report on data for the operating units they have access to, OracleAS Adapter for Oracle Applications uses the MOAC security feature to determine the operating unit access privileges and derive the Organization ID based on relevant profile values.

With MOAC, a system administrator can predefine the scope of access privileges as a security profile, and then use the profile option *MO: Security Profile* to associate the security profile with a responsibility. By using this approach, multiple operating units are associated with a security profile and that security profile is then assigned to a responsibility. Therefore, through the access control of security profiles, users can access data in multiple operating units without changing responsibility.

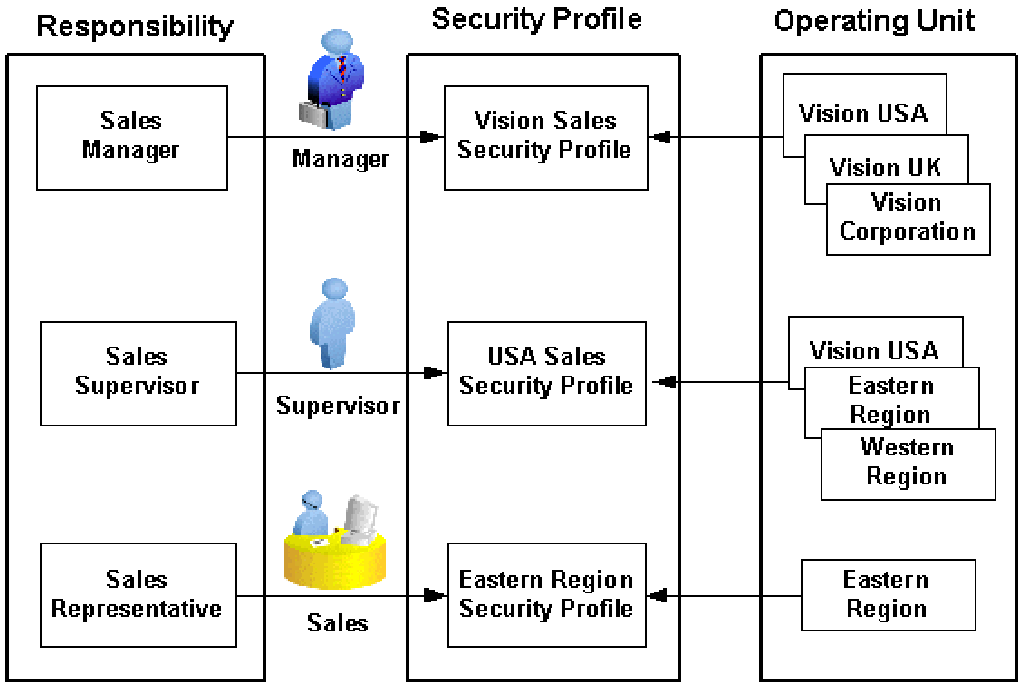
Security profiles are defined based on organization hierarchies. For example, a sales company consists of USA and UK operating units; the USA operating unit has Western Region Sales and East Region Sales. Sales managers are responsible for both USA and UK sales, supervisors are responsible for either USA or UK, and sales representatives are only responsible for their designated sales regions. The Sales organization hierarchy can be illustrated as follows:

Sales Organization Hierarchy



To secure sales data within the company, relevant operating units can be associated with predefined security profiles. For example, all the sales data access privileges are grouped into the Vision Sales security profile. A USA Sales security profile is for USA related data, and a regional security profile is for designated regional data. The system administrator can associate these security profiles containing multiple operating units with users through appropriate *responsibilities*. Therefore, sales supervisors can easily access sales data in the Eastern or Western region without changing their responsibilities. The following diagram illustrates the relationship between security profiles, responsibilities, and operating units for this sales company:

Relationship Diagram Between Security Profiles, Responsibilities, and Operating Units

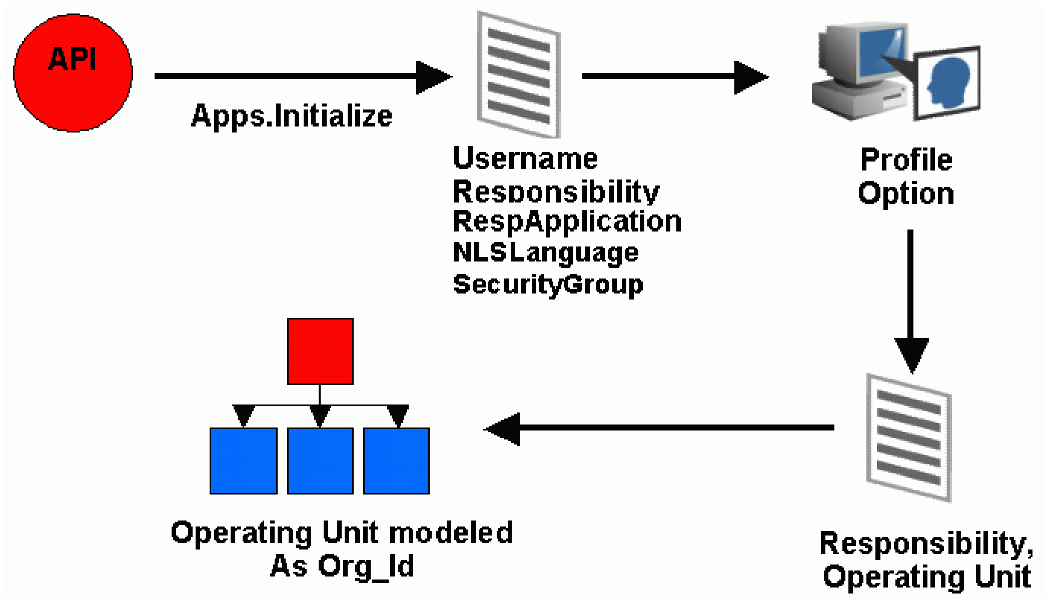


Responsibility Determines Operating Units

Because responsibilities are associated with security profiles that linked to operating units, your responsibility is the key in determining which operating units you will have the access privileges.

The following diagram illustrates how Oracle Applications use the profile options in a multi-organization environment:

Building Applications Context for Multiple Organizations



1. When the system integrator runs, the process achieves the integration with Oracle Applications using PL/SQL APIs.

2. Applications context is set, taking into account the values passed for the header variables Username, Responsibility, Responsibility Application, Security Group, and NLS Language. If the values for the new header variables Responsibility Application, Security Group, and NLS Language are not passed, context information will be determined based on Username and Responsibility.

3. With these parameters, a lookup on all System Profile Values assigned to that responsibility is done to determine the Operating Unit within a multi-organization environment.

4. The Operating Unit is modeled as Organization ID derived from the security profile value.

5. The data is read and written into the Oracle Applications with the parameters of Username, Responsibility, Responsibility Application, Security Group, NLS Language, and Organization ID.

To integrate business processes or invoke BPEL processes, it is essential to propagate these applications context values through a flexible mechanism that allows you to effectively set each individual context value if needed and pass them to complete a BPEL process and meet integration needs. The following topics are discussed in this section:

- Supporting for Applications Context Header Variables, page 3-8

- Support for Multiple Organization Setups, page 3-13
- Support for Multiple Languages, page 3-15

Supporting for Applications Context Header Variables

To effectively set applications context values required in the SOA Suite to process Concurrent Programs and PL/SQL APIs, Adapter for Oracle Applications provides a flexible mechanism that allows each header variable to be set and passed in the Adapter user interface through the Assign activity.

Setting Header Values for Applications Context

The following header values are used in setting applications context required in a business activity or to complete a BPEL process:

- `Username`
- `Responsibility`
- `ORG_ID`
- `RespApplication`
- `SecurityGroup`
- `NLSLanguage`

> **Note:** Existing header variable `Responsibility` used in earlier releases can now take Responsibility Key as well as Responsibility Name as input. If the header variable `NLSLanguage` is set, and Responsibility Name is passed, the value passed for `Responsibility` is expected to be in the same language. However, Responsibility Key as well as all other header variables are language independent.
>
> All these header variables would be used together to set the applications context. Alternatively, passing just the Username and Responsibility would work as it did in the earlier releases.

In the case of a null or empty value, the default value for Username is `SYSADMIN`, the default Responsibility is `System Administrator`, the default Security Group is `Standard`, and the default NLS Language is `US`.

Since the *NLS Language* and *Organization ID* header values are used in supporting applications context, Adapter for Oracle Applications also supports the following features based on the concept of applications context:

- Supporting for Multiple Organization Setups, page 3-13

- Supporting for Multiple Languages, page 3-15

Design-Time Tasks for Assigning Header Variables

Adapter for Oracle Applications uses the following procedures to complete the design-time tasks to assign header variables:

1. Create a New BPEL Project, page 8-3
2. Add a Partner Link, page 8-6
3. Configure an Invoke Activity, page 8-19

   This activity involves the following tasks:

   - Configure basic information in the General tab, page 8-19

     Configure an **Invoke** activity by linking the activity to the Partner Link you just created. This opens the General tab in the Edit Invoke dialog box with Partner Link and Operation information populated.

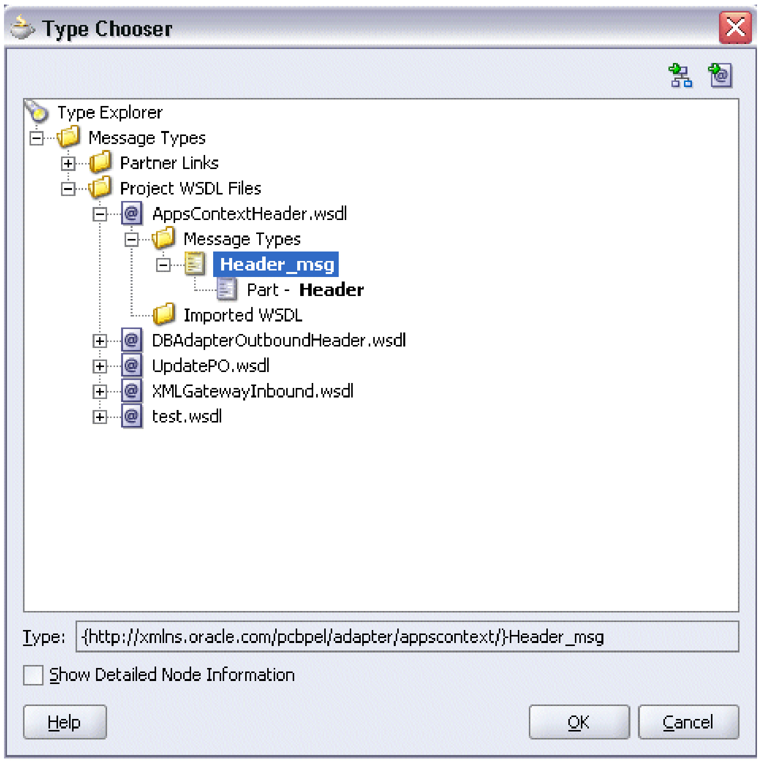
     You can create an Input Variable and an Output Variable for the Invoke activity.

   - Create the Header Variable in the Adapters tab, page 8-21

     Create the header variable used in applications context for Oracle Applications to identify the application user and the associated organization information.

     > **Note:** You can optionally declare the header variable by setting a header variable `{http://xmlns.oracle.com/pcbpel/adapter/appscontext/}Header_msg` to be used in the execution of an API. This declaration provides context information for Oracle Applications to identify the application user that will be executing the API.
     >
     > Since the header variable declaration provides context information for Oracle Applications to identify the application user, Adapter for Oracle Applications uses this variable declaration to support the MLS feature and other features that utilize the concept of applications context including the Organization ID support in multiple organization setups.

Declaring Header Variable



Because the declaration of this header variable is optional, if you do not declare the variable, the default *username* is `SYSADMIN`, the default *responsibility* is `System Administrator`, the default *Security Group* is `Standard`, and the default *NLS Language* is `US`.

For more information, see Declaring Header Variables, page 8-21.

4. Assign Header Values Using an Assign Activity

   After creating the header variable, you need to configure an **Assign** activity by placing it before the **Invoke** activity.

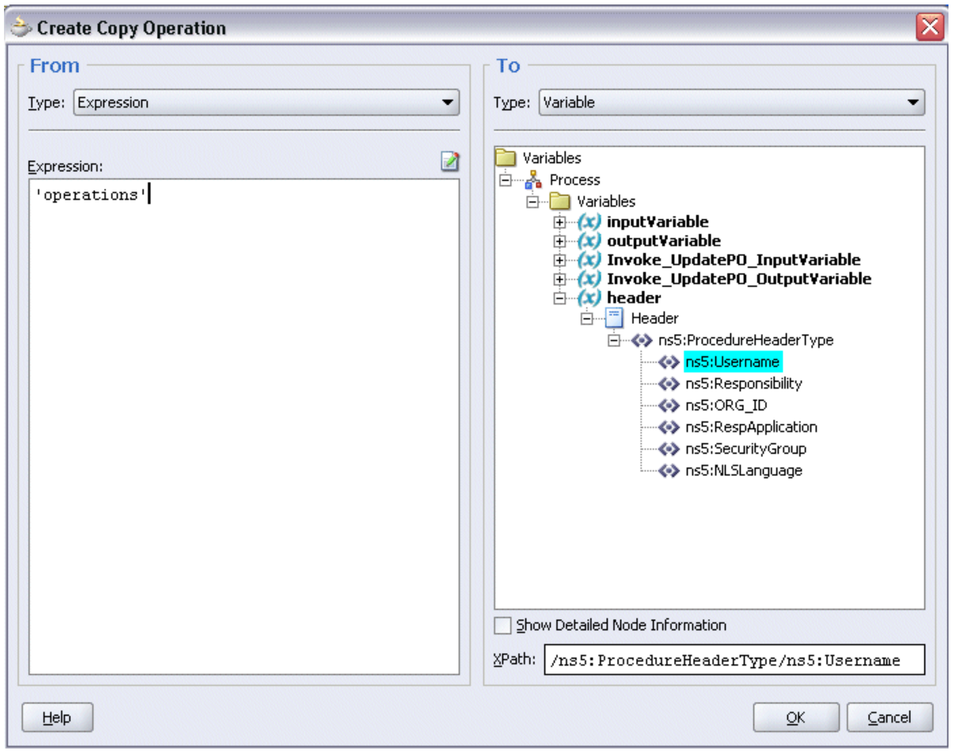
   1. Select **Copy Operation** tab in the Assign dialog box and select **Copy Operation...** from the Create drop-down list.

   2. Enter header variable values in the Assign activity.

      For example, in the From group, select 'Expression' as the type and enter a username variable value such as `'operations'`. In the To group, navigate to **Variables > Process > Variables > header > Header >**

ns5:ProcedureHeaderType and select `ns5:Username`. The XPath field should contain your selected entry.

Assigning a Value to Username Variable



```
'operations'
```

XPath: `/ns5:ProcedureHeaderType/ns5:Username`

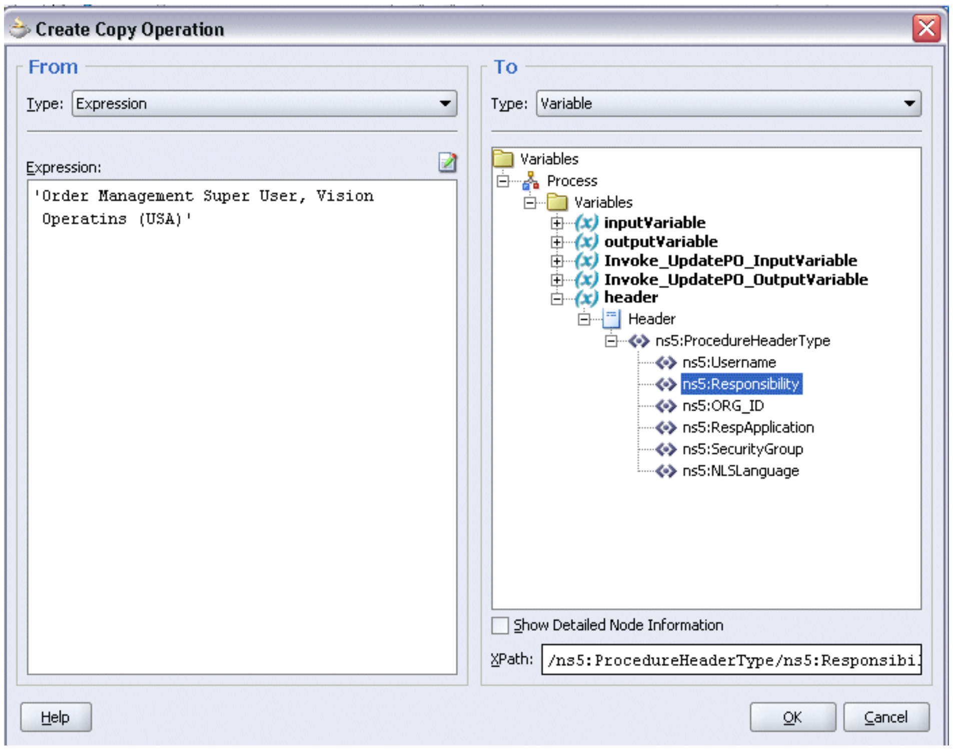
Click **OK**.

3. Repeat the first two steps, and then enter Responsibility value.

   In the From group, select Expression as the Type and enter, for example, `'Order Management Super User, Vision Operations (USA)'`. In the To group, navigate to **Variables > Process > Variables > header > Header > ns5:ProcedureHeaderType** and select `ns5:Responsibility`.

   The XPath field should contain your selected entry.

Assigning a Value to Responsibility Variable



Type: Expression

Expression:

```
'Order Management Super User, Vision
 Operatins (USA)'
```

Type: Variable

```
Variables
  Process
    Variables
      inputVariable
      outputVariable
      Invoke_UpdatePO_InputVariable
      Invoke_UpdatePO_OutputVariable
      header
        Header
          ns5:ProcedureHeaderType
            ns5:Username
            ns5:Responsibility
            ns5:ORG_ID
            ns5:RespApplication
            ns5:SecurityGroup
            ns5:NLSLanguage
```

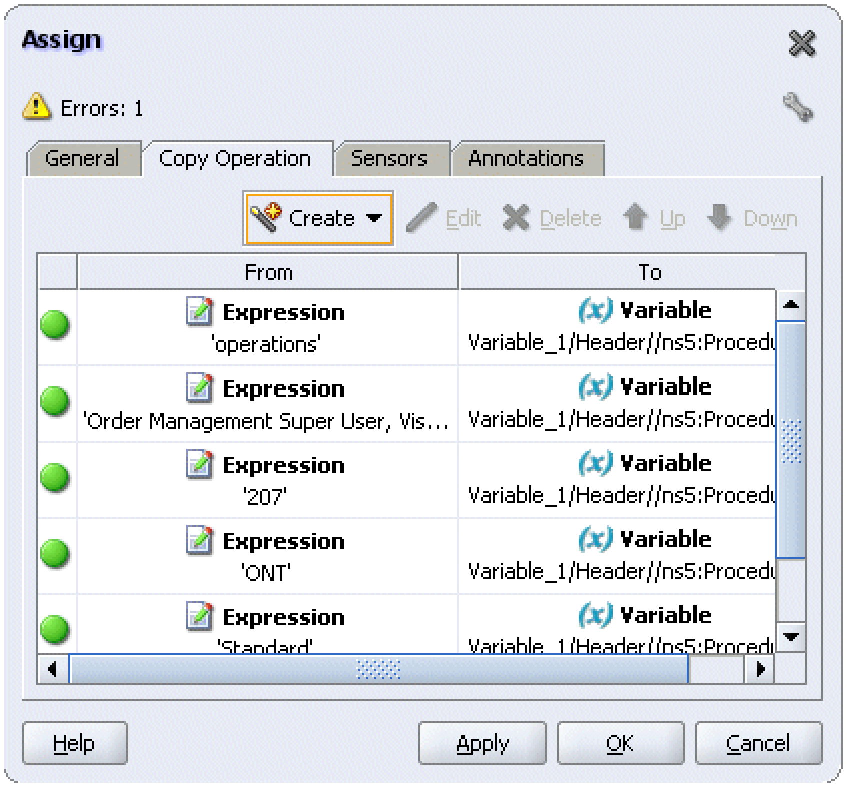
Show Detailed Node Information

XPath: /ns5:ProcedureHeaderType/ns5:Responsibi

Click **OK**.

4. Use the same method to enter the rest of header variables if necessary to complete the Assign activity. For example, enter the following values:

   - `'207'` for `ORG_ID`
   - `'ONT'` for `RespApplication`
   - `'Standard'` for `SecurityGroup`
   - `'US'` for `NLSLanguage`



	From	To
●	Expression 'operations'	Variable Variable_1/Header//ns5:Procedu
●	Expression 'Order Management Super User, Vis...	Variable Variable_1/Header//ns5:Procedu
●	Expression '207'	Variable Variable_1/Header//ns5:Procedu
●	Expression 'ONT'	Variable Variable_1/Header//ns5:Procedu
●	Expression 'Standard'	Variable Variable_1/Header//ns5:Procedu

5. Click **Apply** and **OK** to complete the Assign activity.

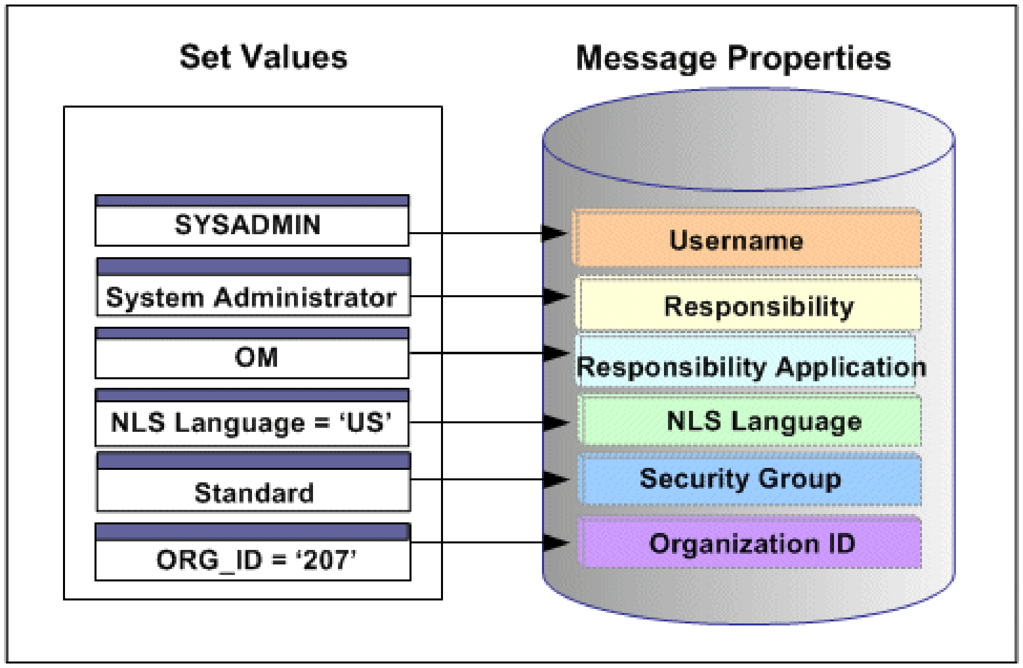
Supporting for Multiple Organization Setups

Instead of implicitly deriving organization information from a profile value during the Oracle Applications setups, OracleAS Adapter for Oracle Applications provides a mechanism which allows Organization ID information to be directly entered through the header variable creation during the design time to support multiple organization setups.

OracleAS Adapter for Oracle Applications uses the header variable to include *Username, Responsibility, Responsibility Application, NLS Language, Security Group*, and *Organization ID*, the essential elements for applications context. Once you declare the header variable and assign appropriate values to each parameter contained in the header for an interface which requires the applications context to be set, these values in the header will be passed and used as an input to the rest of activities in the BPEL process.

Note: Integration interface types that require applications context to be set are PL/SQL APIs, concurrent programs, and EDI programs.

Creating and Assigning Header Variables in a BPEL Process



The advantage of having this header variable mechanism in supporting multiple organization setups is that with only one single BPEL process, organization information can be easily placed into multiple organizations within the Oracle E-Business Suite if the Organization ID value has been specified in the header. While in the past, since Organization ID is implicitly derived from the profile value based on an application logon user's username and responsibility; therefore, only that associated organization for the invocation of the deployed BPEL process can be inserted.

With the example described earlier in the Multiple Organization Setup section, when a change order is placed within the French sales office, a sales manager from the French office logs on to the system to update the order which invokes a PL/SQL API for that change. If the Organization ID contained in the header variable has been assigned with a value, such as `207` for the French sales office, the Organization ID associated with the sales manager will be set to French sales office for the invocation of the API.

> **Note:** For Oracle E-Business Suite Release 12, *Organization ID* parameter is automatically included in the header variable along with *Username*, *Responsibility*, *Responsibility Application*, *Security Group*, and *NLS Language*. For Release 11.5.10, you must apply 11i.ATG_PF.H.delta.5 (RUP5) (patch 5473858) to Oracle Applications instance in order to have Organization ID displayed in the header.

Supporting for Multiple Languages

By leveraging the Multiple Language Support (MLS) feature from Oracle E-Business Suite, OracleAS Adapter for Oracle Applications provides a comprehensive language support mechanism that allows an appropriate language to be dynamically set at run time.

NLS Language Value Sets the Session Language

OracleAS Adapter for Oracle Applications sets the session language for a transaction based on one of the following conditions, whichever is appropriate:

1. The NLS Language value can be set by using the header variable `NLSLanguage`. If a valid language value is passed (i.e. language is enabled in Oracle E-Business Suite instance), the session language is set to the corresponding language.

2. In case the NLS Language header variable is not passed, Adapter for Oracle Applications will use the default language of the user, based on the user preferences, to set the session language.

3. If the user's default language is not found or set, NLS Language (`NLSLanguage`) variable value would be set to `'US'` (American).

This mechanism allows an appropriate session language to be dynamically set at run time first based on the passed NLS Language variable value, then the user's default language, and last determined by the default NLS Language value 'US'.

For more information about how to set NLS Language variable value by using `NLSLanguage`, see Supporting for Applications Context Header Variables, page 3-8.

Determining a User's Default Language

To identify the default language used in the database session for data query and retrieval, Adapter for Oracle Applications will first examine the *ICX: Language* profile value at all levels including user, responsibility, application, and site. If it is not set at any of those levels, OracleAS Adapter for Oracle Applications then takes `NLS_LANGUAGE` parameter from the database instance National Language Support (NLS) parameters. The NLS parameters initialized in the session are:

- `NLS_LANGUAGE`
- `NLS_SORT`
- `NLS_DATE_FORMAT`
- `NLS_DATE_LANGUAGE`
- `NLS_NUMERIC_CHARACTERS`
- `NLS_TERRITORY`

For example, when a user with a default language Japanese logs into the system and performs a transaction through the execution of an API that the user defined in the Partner link of the BPEL process, OracleAS Adapter for Oracle Applications will first examine if the NLS Language variable value is passed. If it is passed with a valid language, then the session language will be set based on the passed value regardless of the default language. If the NLS Language value is not passed, OracleAS Adapter for Oracle Applications will set the session language to the preferred / default language of the user. In case that cannot be found, the language would be set to `'US'` (American). The default language is set in the General Preferences page of Oracle Applications.

> **Note:** The default language set in the General Preferences page updates the *ICX: Language* profile option.

When the applications context is set, user preferences like date formats, time zone information, etc. would be taken care automatically.

Please refer to the Set Preferences section, Getting Started with Oracle Applications chapter, *Oracle Applications User's Guide* for the information on how to set the user preferences.

Understanding OracleAS Adapter for Oracle Applications Security

Security is the most critical feature that is designed to guard application content from unauthorized access. By leveraging Oracle User Management function security, Oracle Application Server Adapter for Oracle Applications provides a security feature which only allows users with authorized privileges to execute APIs that they are exposed through the BPEL process to update Oracle Applications. This protects application programming interfaces (APIs) from unauthorized access or execution without security checks.

Please note that Oracle Application Server Adapter for Oracle Applications provides this security support as an optional feature. If you want all the login users to access and execute APIs without security checks, you can turn the security feature off using the "EBS Adapter for BPEL, Function Security Enabled" (EBS_ADAPTER_FUNCTION_SEC_ENABLED) profile option.

- If it is set to 'Y', then the function security feature is enabled and all API calls for PL/SQL APIs, Oracle e-Commerce Gateway, and concurrent programs will be checked for user security before they are invoked.
- If it is set to 'N' (default value), then the function security feature is disabled. No security check is implemented during the invocation of all API calls.

> **Note:** To have this function security feature available, appropriate patches need to be applied to your environment. See "Oracle Application Server Adapter for Oracle Applications Documentation

Update, Release 10*g*", My Oracle Support Knowledge Document 464164.1 for details.

This section includes the following topics:

- Function Security for OracleAS Adapter for Oracle Applications, page 3-17
- Creating Security Grants, page 3-18

Function Security for OracleAS Adapter for Oracle Applications

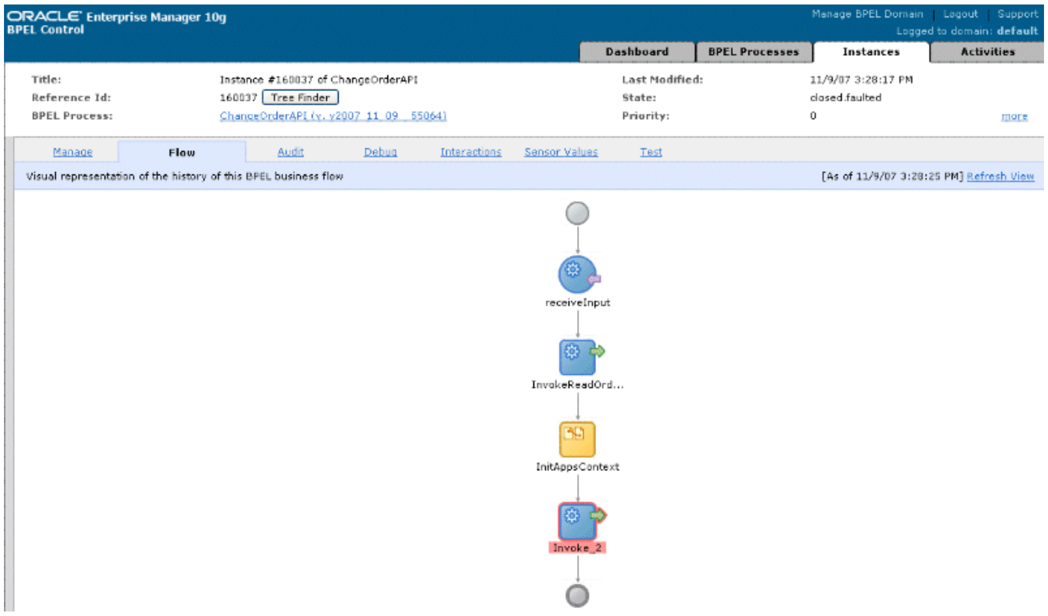
Function security is the basic access control in Oracle Applications. It restricts user access to individual menus and menu options within the system regardless of which application data in the row. Since APIs are stored procedures that enable you to insert and update data in Oracle Applications, when having the function security layer enforced on the access to an API, it actually implicitly restricts the data access to the application.

To allow appropriate users with right privileges to execute APIs, OracleAS Adapter for Oracle Applications leverages Oracle User Management Role-Based Access Control security (RBAC) to reinforce the function security through user roles and whether a user can access an API is determined by the roles granted to the user. A role can be configured to consolidate the responsibilities, permissions, permission sets, and function security policies that users require to perform a specific function. This simplifies mass updates of user permissions because changes can be done through roles which will inherit the new sets of permissions automatically. Based on the job functions, each role can be assigned a specific permission or permission set if needed. For example, a procurement organization may include 'Buyer', 'Purchasing Manager', and 'Purchasing Support' roles. The 'Purchasing Manager' role would include a permission set that contains all Purchase Order (PO) Creation, PO Change, and Contract PO related APIs allowing the manager role to perform a job function while the Buyer or Support role may not have the access privileges.

In OracleAS Adapter for Oracle Applications, all annotated APIs resided in Oracle Integration Repository are registered on the FND_FORM_FUNCTIONS table so that the function security (FND_FORM_FUNCTIONS) can be applied. This allows the creation of a secured function for each API.

By leveraging the concept of permission sets, OracleAS Adapter for Oracle Applications allows related APIs to be grouped and sequenced under one permission set; each permission set can be associated with a function role and then assigned to users through security grants. When a user logs on to Oracle E-Business Suite and tries to access an API exposed through the BPEL process, if the security feature is enabled, the function security API will be invoked to validate whether the user is authorized to have the execution privileges on the API.

For example, if a user does not have the access privileges for a PL/SQL API exposed through a BPEL process, the execution of that BPEL process will fail while trying to invoke the PL/SQL API as shown in the following diagram:



ORACLE Enterprise Manager 10g
BPEL Control
Manage BPEL Domain | Logout | Support
Logged to domain: default
Dashboard
BPEL Processes
Instances
Activities

Title: Instance #160037 of ChangeOrderAPI
Reference Id: 160037
Tree Finder
BPEL Process: ChangeOrderAPI (v. v2007_11_09__55064)
Last Modified: 11/9/07 3:28:17 PM
State: closed.faulted
Priority: 0

Visual representation of the history of this BPEL business flow

[As of 11/9/07 3:28:25 PM] Refresh View

Without the authorized privileges, the Function Security Validation Exception message will be raised indicating that the user does not have the privilege for a specific PL/SQL API.

For more information on Function Security and RBAC security models, see *Oracle Applications System Administrator's Guide - Security* for details.

Creating Security Grants

To secure the API invocation only to a user with appropriate execution privileges, OracleAS Adapter for Oracle Applications uses the following steps to create security grants to users through user roles:

1. Creating a Permission Set, page 3-18
2. Creating a User Role, page 3-21
3. Granting a Permission Set to a User Through a Role, page 3-22

Creating a Permission Set

Use the following steps to create a permission set:

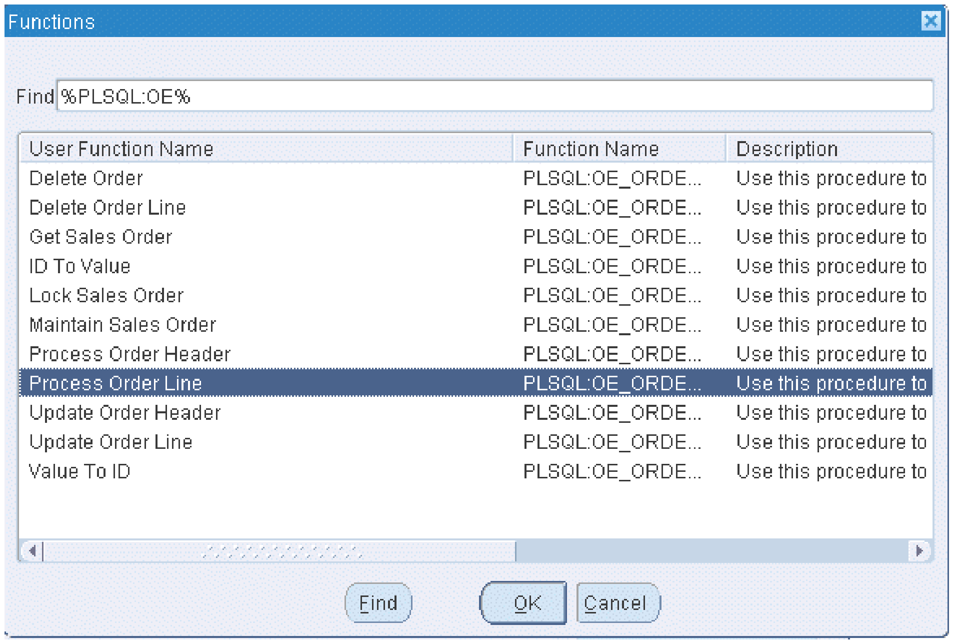
1. Log in to Oracle E-Business Suite using the System Administrator responsibility.
2. Select Application: Menu from the Navigator to access the Menus window.
3. Enter the following menu information:

- Menu: Enter an appropriate menu name (such as 'OE_PROCESS_LINE_PS').
- User Menu Name: Enter an appropriate user menu name (such as 'Order Manager Process Line Permission Set').
- Menu Type: Permission Set
- Description: Enter description information for this menu.

4. Add all the functions that you want to group on this Permission Set by entering values for Seq and Function.
   1. Enter the Seq field.
   2. In the Function column, search for the functions you want to assign to this permission set.

      Select an appropriate function name by performing a search in the Functions window. For example the syntax for searching public PL/SQL APIs is: `PLSQL:<package name>:<procedure name>`. You can enter `%PLSQL:OE%` in the Find field and click **Find** to execute the search.

Searching for Functions



Functions

Find %PLSQL:OE%

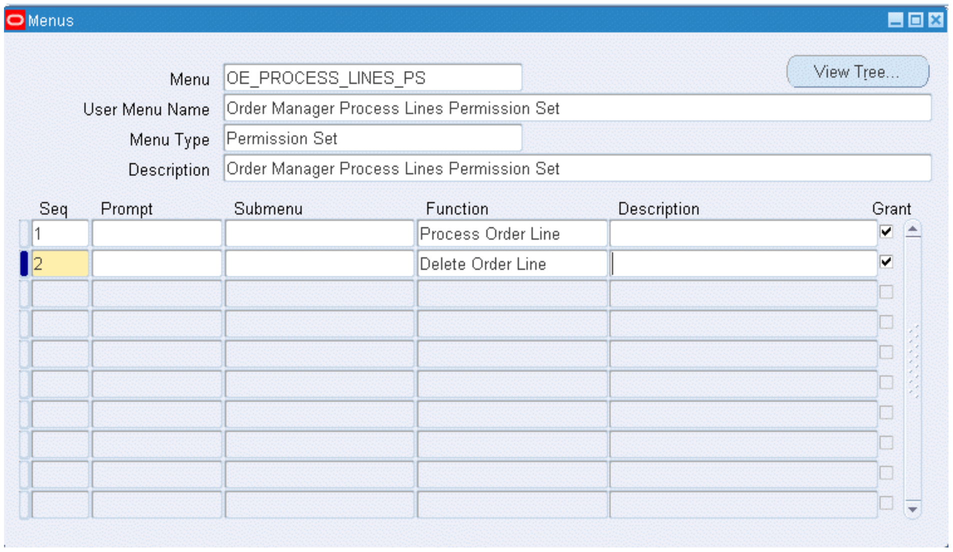
User Function Name	Function Name	Description
Delete Order	PLSQL:OE_ORDE...	Use this procedure to
Delete Order Line	PLSQL:OE_ORDE...	Use this procedure to
Get Sales Order	PLSQL:OE_ORDE...	Use this procedure to
ID To Value	PLSQL:OE_ORDE...	Use this procedure to
Lock Sales Order	PLSQL:OE_ORDE...	Use this procedure to
Maintain Sales Order	PLSQL:OE_ORDE...	Use this procedure to
Process Order Header	PLSQL:OE_ORDE...	Use this procedure to
Process Order Line	PLSQL:OE_ORDE...	Use this procedure to
Update Order Header	PLSQL:OE_ORDE...	Use this procedure to
Update Order Line	PLSQL:OE_ORDE...	Use this procedure to
Value To ID	PLSQL:OE_ORDE...	Use this procedure to

Based on your BPEL process, select appropriate functions and then grant the permissions to the APIs that you will be invoking from the BPEL process.

For example, for a sales order line change BPEL process, you select sales order line change related functions contained in the order change PL/SQL API and group them as a permission set, and then grant the permission set to an appropriate user through a role.

See: Creating a User Role, page 3-21.

Permission Set Menu



Menu	OE_PROCESS_LINES_PS
User Menu Name	Order Manager Process Lines Permission Set
Menu Type	Permission Set
Description	Order Manager Process Lines Permission Set

Seq	Prompt	Submenu	Function	Description	Grant
1			Process Order Line		✓
2			Delete Order Line		✓

5. Save the Permission Set.

Creating a User Role

Permission sets are granted through user roles. Therefore, you must first create a role and then assign the role to a user.

Use the following steps to create a user role:

1. Log in to Oracle E-Business Suite using the User Management responsibility.

2. Select Roles & Role Inheritance from the Navigator to access the Roles & Role Inheritance page.

3. Click **Create Role** to access the Create Role page.

4. Enter the following information to create a role:

   - Category: Select Miscellaneous from the drop-down list.

   - Role Code: Enter an appropriate role code (such as 'EBS_ADAPTER_ROLE').

   - Display Name: Enter appropriate information for the display name (such as 'EBS Adapter Role').

   - Description: Enter appropriate information for the description (such as 'EBS Adapter Role').

- Application: Select an appropriate application (such as 'Application Object Library').

- Active Date: Enter an appropriate date which is earlier than or equal to today's date so that the role can become valid right away.

5. Save the information and click **Create Grant**.

6. Enter the following information in the Create Grant: Define Grant page:
   - Name: Enter an appropriate name (such as 'EBS_ADAPTER_GRANT').

   - Description: Enter description information for this grant.

7. Click **Next**.

8. In the Create Grant: Define Object Parameters and Select Set page, select the Permission Set you created earlier in the Creating a Permission Set section, page 3-18 and click **Next**.

9. Click **Finish**.

Granting a Permission Set to a User Through a Role

Use the following steps to grant a permission set to a user through a role:

1. Log in to Oracle E-Business Suite using the User Management responsibility.

2. Select Users from the Navigator to access the User Maintenance page.

3. Search for the user you want to assign the role and click **Go**.

4. Select the **Update** icon next to the user name that you want to assign the role.

5. In the Update User page, click **Assign Roles** to have the Search window populated which allows you to search for the role that you created earlier.

6. Select the role (such as 'EBS_ADAPTER_ROLE') and save your update.

Secured Connection Between Oracle E-Business Suite and Oracle SOA Suite Using J2EE Data Source Implementation

By implementing the J2EE Data Source for secured connection between Oracle E-Business Suite and Oracle SOA Suite, two distinct advantages can be leveraged. Firstly, to get the secured connection to the Oracle E-Business Suite's application database you do not require the database administrator's username and password, just

FND username and password (concept of Oracle Applications username and password) is sufficient. Secondly, since the password is not stored in the middleware, not only this eliminates the security risk, but also does away the need to keep the password in-sync between Oracle E-Business Suite and SOA Suite.

OracleAS Adapter for Oracle Applications uses a new mechanism to authenticate users at run time and get the connection to Oracle Applications databases through the use of J2EE data sources. Since J2EE data sources are defined in OC4J container that runs the BPEL processes, this approach is native to Oracle E-Business Suite in defining the connection pool to access the application database.

With this new mechanism, account details information including application login user name and password that was required as part of the configuration for database connection is now added together with the dbc file location as input parameters during the J2EE data source creation.

To accomplish this process, the following steps are used to define J2EE data source connection to the Oracle E-Business Suite database:

1. Register your Service-Oriented Architecture (SOA) Suite middle tier node on the Oracle E-Business Suite environment and generate the dbc file used by the data source implementation to instantiate the connections.

2. Copy the dbc file to the middle tier server where your SOA Suite server runs, and place it on a location in the file system to which the SOA Suite owner has access.

3. Create a connection pool where you need to enter the application login user name, password, and dbc file location as the connection factory properties.

4. Create an application data source. This is the step that you associate the application data source with the Java Naming and Directory Interface (JNDI) name for the application database connection for OracleAS Adapter for Oracle Applications.

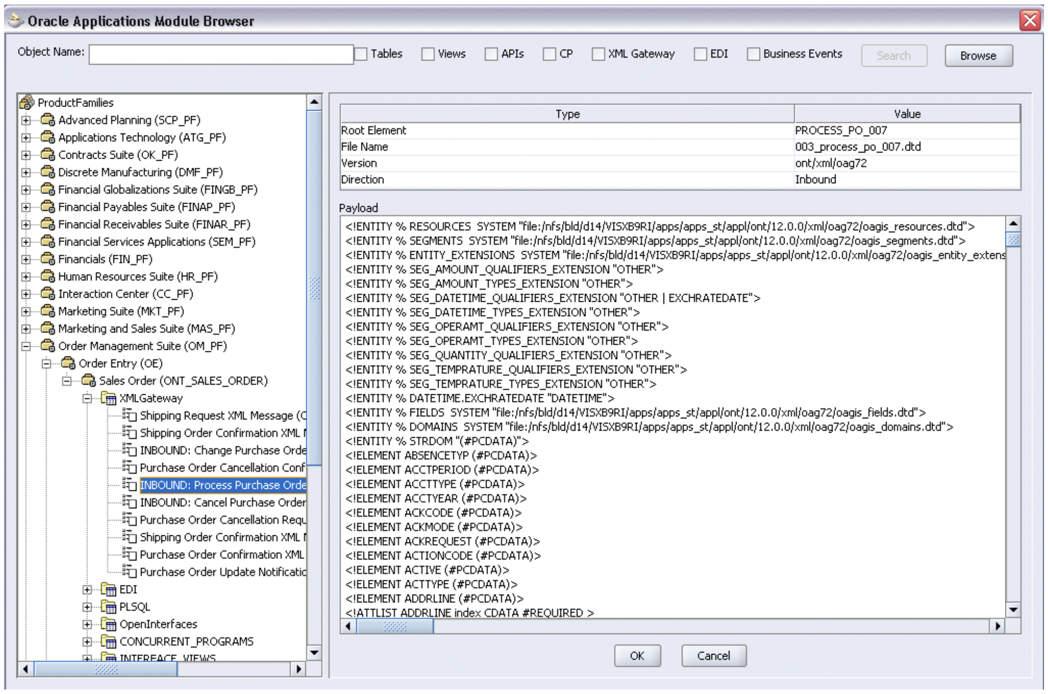
   Note: To have this feature available, you must apply necessary patches and perform appropriate setup tasks to enable the connectivity between Oracle E-Business Suite and an external application server at run time for Oracle E-Business Suite Release 12 and Release 11*i*.

   See "Oracle Application Server Adapter for Oracle Applications Documentation Update, Release 10*g*", My Oracle Support Knowledge Document 464164.1 for details.

Understanding the Oracle Applications Module Browser

In addition to the interfaces that are made available through Oracle Integration Repository, OracleAS Adapter for Oracle Applications enables you to use business events, customized PL/SQL APIs, customized XML Gateway maps, and selected concurrent programs, all of which you can explore using the Oracle Applications Module Browser.

Oracle Applications Module Browser



```
ProductFamilies
  Advanced Planning (SCP_PF)
  Applications Technology (ATG_PF)
  Contracts Suite (OK_PF)
  Discrete Manufacturing (DMF_PF)
  Financial Globalizations Suite (FINGB_PF)
  Financial Payables Suite (FINAP_PF)
  Financial Receivables Suite (FINAR_PF)
  Financial Services Applications (SEM_PF)
  Financials (FIN_PF)
  Human Resources Suite (HR_PF)
  Interaction Center (CC_PF)
  Marketing Suite (MKT_PF)
  Marketing and Sales Suite (MAS_PF)
  Order Management Suite (OM_PF)
    Order Entry (OE)
      Sales Order (ONT_SALES_ORDER)
        XMLGateway
          Shipping Request XML Message (C
          Shipping Order Confirmation XML I
          INBOUND: Change Purchase Orde
          Purchase Order Cancellation Conf
          INBOUND: Process Purchase Orde
          INBOUND: Cancel Purchase Order
          Purchase Order Cancellation Requ
          Shipping Order Confirmation XML I
          Purchase Order Confirmation XML
          Purchase Order Update Notificatic
        EDI
        PLSQL
        OpenInterfaces
        CONCURRENT_PROGRAMS
        INTERFACE_VIEWS
```

Type	Value
Root Element	PROCESS_PO_007
File Name	003_process_po_007.dtd
Version	ont/xml/oag72
Direction	Inbound

```
<!ENTITY % RESOURCES  SYSTEM "file:/nfs/bld/d14/VISXB9RI/apps/apps_st/appl/ont/12.0.0/xml/oag72/oagis_resources.dtd">
<!ENTITY % SEGMENTS  SYSTEM "file:/nfs/bld/d14/VISXB9RI/apps/apps_st/appl/ont/12.0.0/xml/oag72/oagis_segments.dtd">
<!ENTITY % ENTITY_EXTENSIONS  SYSTEM "file:/nfs/bld/d14/VISXB9RI/apps/apps_st/appl/ont/12.0.0/xml/oag72/oagis_entity_extens
<!ENTITY % SEG_AMOUNT_QUALIFIERS_EXTENSION "OTHER">
<!ENTITY % SEG_AMOUNT_TYPES_EXTENSION "OTHER">
<!ENTITY % SEG_DATETIME_QUALIFIERS_EXTENSION "OTHER | EXCHRATEDATE">
<!ENTITY % SEG_DATETIME_TYPES_EXTENSION "OTHER">
<!ENTITY % SEG_OPERAMT_QUALIFIERS_EXTENSION "OTHER">
<!ENTITY % SEG_OPERAMT_TYPES_EXTENSION "OTHER">
<!ENTITY % SEG_QUANTITY_QUALIFIERS_EXTENSION "OTHER">
<!ENTITY % SEG_TEMPRATURE_QUALIFIERS_EXTENSION "OTHER">
<!ENTITY % SEG_TEMPRATURE_TYPES_EXTENSION "OTHER">
<!ENTITY % DATETIME.EXCHRATEDATE "DATETIME">
<!ENTITY % FIELDS  SYSTEM "file:/nfs/bld/d14/VISXB9RI/apps/apps_st/appl/ont/12.0.0/xml/oag72/oagis_fields.dtd">
<!ENTITY % DOMAINS  SYSTEM "file:/nfs/bld/d14/VISXB9RI/apps/apps_st/appl/ont/12.0.0/xml/oag72/oagis_domains.dtd">
<!ENTITY % STRDOM "(#PCDATA)">
<!ELEMENT ABSENCETYP (#PCDATA)>
<!ELEMENT ACCTPERIOD (#PCDATA)>
<!ELEMENT ACCTTYPE (#PCDATA)>
<!ELEMENT ACCTYEAR (#PCDATA)>
<!ELEMENT ACKCODE (#PCDATA)>
<!ELEMENT ACKMODE (#PCDATA)>
<!ELEMENT ACKREQUEST (#PCDATA)>
<!ELEMENT ACTIONCODE (#PCDATA)>
<!ELEMENT ACTIVE (#PCDATA)>
<!ELEMENT ACTTYPE (#PCDATA)>
<!ELEMENT ADDRLINE (#PCDATA)>
<!ATTLIST ADDRLINE index CDATA #REQUIRED >
```

The Oracle Applications Module Browser is a key component of OracleAS Adapter for Oracle Applications. You use the Module Browser to select the interface needed to define a partner link. The Module Browser combines interface data from Oracle Integration Repository with information about the additional interfaces supported by OracleAS Adapter for Oracle Applications, organized in a tree hierarchy as follows:

```
ProductFamilies
 |-[product_family]
 |  |-[product]
 |     |-[business_entity]
 |        |-XML Gateway ([n])
 |        |-EDI ([n])
 |        |-PLSQL ([n])
 |        |  |-[package_name]
 |        |-OpenInterfaces ([n])
 |           |-[OpenInterface_name]
 |              |-Tables ([n])
 |              |-Views ([n])
 |              |-ConcurrentPrograms ([n])
 |-Other Interfaces
    |-Business Events
    |-Custom Objects
       |-PLSQL APIs
       |  |-[package_name]
       |-XMLGateway Maps
          |-Inbound
          |-Outbound
```

- The items under Other Interfaces, as well as certain PL/SQL APIs and concurrent programs under the *[product family]* hierarchy, are available through OracleAS Adapter for Oracle Applications, but not through Oracle Integration Repository.

- The number of interfaces indicated by [n] only appears in the case of an Oracle E-Business Suite 11.5.10 instance is used. It will not be displayed if you are connecting to an Oracle E-Business Suite pre-11.5.10 or Release 12 instance.

The Oracle Integration Repository interface data populates the ***[product_family]*** sections, grouped according to the products and business entities to which they belong. Each interface type heading is followed by a number ***[n]*** indicating how many of that type are listed in that section.

Business events appear under Other Interfaces. Customized XML Gateway maps appear under **Other Interfaces > Custom Objects,** categorized as either inbound or outbound.

Customized PL/SQL APIs appear in two places:

- Procedures within a package that's already exposed via Oracle Integration Repository appear under the package name within a product family hierarchy.

- Procedures within a completely new package appear under the package name, under **Other Interfaces > Custom Objects**.

4

Using XML Gateway for BPEL Process Integration

This chapter covers the following topics:

- Testing the BPEL Process
- Troubleshooting and Debugging

Overview of XML Gateway

The OracleAS Adapter for Oracle Applications provides a bridge between Oracle Applications and third party applications. Inbound and outbound XML data is exchanged between Oracle Applications and third party applications through the XML Gateway.

Oracle XML Gateway provides a common, standards-based approach for XML integration between Oracle Applications and third party applications, both inside and outside your enterprise. XML is key to an integration solution, as it standardizes the way in which data is searched, exchanged, and presented thereby enabling interoperability throughout the supply chain.

Oracle XML Gateway is an XML messaging based integration infrastructure essentially for business partner integration which includes a set of services that allows easy integration between Oracle Applications and third party applications. Oracle Applications utilize the Oracle Workflow Business Event System to support event-based XML message creation and consumption.

OracleAS Adapter for Oracle Applications can be configured to use XML Gateway to interact with third party applications. The tight integration provided by open interface tables is not suitable for those scenarios where trading partners change frequently. XML Gateway is an ideal solution when you need to interact with third party applications that use open standards. Moreover, it is also suitable for scenarios where trading partners change frequently.

Standards-Based Messaging

As a provider of broad based business application solutions to support all industries, Oracle XML Gateway supports all Document Type Definition (DTD) based XML standards. The majority of the Oracle prebuilt messages delivered with Oracle Applications are premapped using the Open Application Group (OAG) standard. Any Oracle prebuilt message map may be remapped to your standard of choice using the XML Gateway Message Designer.

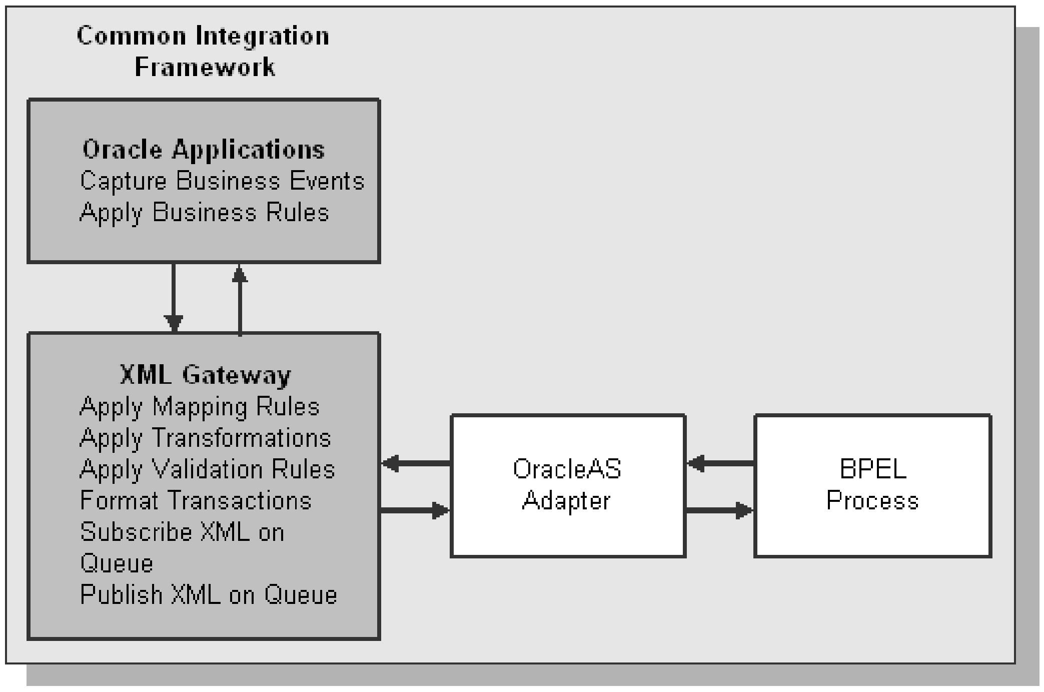
Integration Architecture

XML Gateway provides an application integration infrastructure that is flexible enough to accommodate the integration requirements of any application that needs to integrate with Oracle Applications. XML Gateway enables you to create an efficient and responsive supply chain that links all customers, factories, warehouses, distributors, carriers, and other trading partners. All these entities can seamlessly operate as a single enterprise.

Oracle XML Gateway supports both Business-to-Business (B2B) and Application-to-Application (A2A) initiatives. B2B initiatives include communicating business documents and participating in industry exchanges. An example of an A2A initiative is data integration with legacy and disparate systems.

XML Gateway enables bidirectional integration with Oracle Applications by allowing you to insert and retrieve data from Oracle Applications. The Oracle Applications adapter for XML Gateway supports inbound and outbound XML Gateway message processing. For XML Gateway inbound message processing, the inbound message will be placed in the ECX_INBOUND queue. Agent Listeners running on ECX_INBOUND would enable further processing by the Execution Engine. Oracle XML Gateway picks this XML message, does trading partner validation, and inserts data into Oracle Applications. For XML Gateway outbound message processing, the outbound message will be first enqueued to the ECX_OUTBOUND queue. Oracle BPEL PM listens to ECX_OUTBOUND queue for the message with the same correlation Id BPEL. The message will then be dequeued to retrieve outbound data and then the outbound map will be invoked to update Oracle Applications.

XML Gateway Integration Architecture



Message Queues

The XML Gateway uses queues specifically at two points in the process as well as employing a general error queue. The first point is at the transport agent level between the transport agent module and the XML Gateway. The second point is at the transaction level between base Oracle Applications products and the XML Gateway.

Inbound Queues

Inbound message queues are used for XML messages inbound into Oracle Applications. Inbound message queues are positioned between the Transport Agent and the Oracle Workflow Business Event System.

The messages must be formatted according to the XML Gateway envelope message format. The envelope message format is discussed under XML Gateway Envelope, page 4-4. Oracle Workflow Business Event System copies the inbound messages to the proper inbound Transaction Queue.

Outbound Queues

Outbound message queues are used for XML messages outbound from Oracle Applications. The outbound Message Queue is positioned between the XML Gateway and the Transport Agent.

The XML Gateway creates XML messages, then enqueues them on this queue. The Transport Agent dequeues the message and delivers it to the Trading Partner.

XML Gateway Envelope

In addition to the business document such as a purchase order or invoice in the XML Payload, a set of message attributes are also transmitted. Collectively, these attributes are called the XML Gateway envelope. The following table describes some of these attributes.

Envelope Attributes

Attribute	Description
MESSAGE_TYPE	Payload message format. This defaults to `XML`. Oracle XML Gateway currently supports only `XML`.
MESSAGE_STANDARD	Message format standard as displayed in the Define Transactions form and entered in the Define XML Standards form. This defaults to `OAG`. The message standard entered for an inbound XML document must be the same as the message standard in the trading partner setup.
TRANSACTION_TYPE	External Transaction Type for the business document from the Trading Partner table. The transaction type for an inbound XML document must be the same as the transaction type defined in the Trading Partner form.

Attribute	Description
TRANSACTION_SUBTYPE	External Transaction Subtype for the business document from the Trading Partner table. The transaction subtype for an inbound XML document must be the same as the transaction subtype defined in the Trading Partner form.
DOCUMENT_NUMBER	The document identifier used to identify the transaction, such as a purchase order or invoice number. This field is not used by the XML Gateway, but it may be passed on inbound messages.
PROTOCOL_TYPE	Transmission Protocol as defined in the Trading Partner table.
PROTOCOL_ADDRESS	Transmission address as defined in the Trading Partner table.
USERNAME	USERNAME as defined in the Trading Partner table.
PASSWORD	The password associated with the USERNAME defined in the Trading Partner table.
PARTY_SITE_ID	The party site identifier for an inbound XML document must be the same as the Source Trading Partner location defined in the Trading Partner form.

Attribute	Description
ATTRIBUTE3	For outbound messages, this field has the value from the Destination Trading Partner Location Code in the Trading Partner table. For inbound messages, the presence of this value generates another XML message that is sent to the trading partner identified in the Destination Trading Partner Location Code in the Trading Partner table. This value must be recognized by the hub to forward the XML message to the final recipient of the XML Message. **Note:** For more information, see *Destination Trading Partner Location Code* in the *Oracle XML Gateway User's Guide*. This guide is a part of the Oracle Applications documentation library. Oracle Applications documentation can be accessed from the following link: http://www.oracle.com/technology/documentation/applications.html
PAYLOAD	The XML message.

Parameters defined by the Application

The following parameters may be defined by the base application:

- `ATTRIBUTE1`
- `ATTRIBUTE2`
- `ATTRIBUTE4`
- `ATTRIBUTE5`

Parameters Not Used

The following parameters are not used:

- `PARTYID`
- `PARTYTYPE`

Note: See *Oracle XML Gateway User's Guide* for details on the XML Gateway Execution Engine, Trading Partner validation, and so on. This guide is a part of the Oracle Applications documentation library. Oracle Applications documentation can be accessed from the following link:

http://www.oracle.com/technology/documentation/applications.html

Design-Time Tasks for XML Gateway Inbound Messaging

OracleAS Adapter for Oracle Applications is deployed using the BPEL Process Manager (PM) in Oracle JDeveloper. The BPEL PM creates the WSDL interfaces for the XML Gateway message map.

This section describes the process of configuring OracleAS Adapter for Oracle Applications to use XML Gateway message map. It also describes the tasks required to configure OracleAS Adapter for Oracle Applications using the Adapter Configuration Wizard in Oracle JDeveloper.

Prerequisites to Configure XML Gateway Inbound

Populating XML Gateway Header Variables

The `MESSAGE_TYPE`, `MESSAGE_STANDARD`, `TRANSACTION_TYPE`, `TRANSACTION_SUBTYPE`, and `PARTY_SITE_ID` are the mandatory header variables that you need to populate in order for the XML transactions to complete successfully.

Refer to Configuring the Assign Activity, page 4-32 for more information.

Ensuring Agent Listeners Are Up and Running

You need to configure and schedule two listeners on the Oracle Applications side. These are the ECX Inbound Agent Listener and the ECX Transaction Agent Listener.

Use the following steps to configure these listeners in Oracle Applications:

1. Log in to Oracle Applications with the responsibility of Workflow Administrator.
2. Select the **Workflow Administrator Web Applications** link from the Navigator.
3. Click the **Workflow Manager** link under Oracle Applications Manager.
4. Click the status icon next to **Agent Listeners**.
5. Configure and schedule the **ECX Inbound Agent Listener** and the **ECX Transaction Agent Listener**. Select the listener, and select Start from the **Actions** box. Click **Go**.

Following is a list of the procedures required to accomplish the design-time tasks.

1. Create a new BPEL project, page 4-8
2. Create a partner link, page 4-11
3. Add a partner link for File Adapter, page 4-22

4. Configure the Invoke activity, page 4-30

5. Configure the Assign activity, page 4-32

Creating a New BPEL Project

To create a new BPEL project:

1. Open JDeveloper BPEL Designer.

2. From the **File** menu, select **New**.

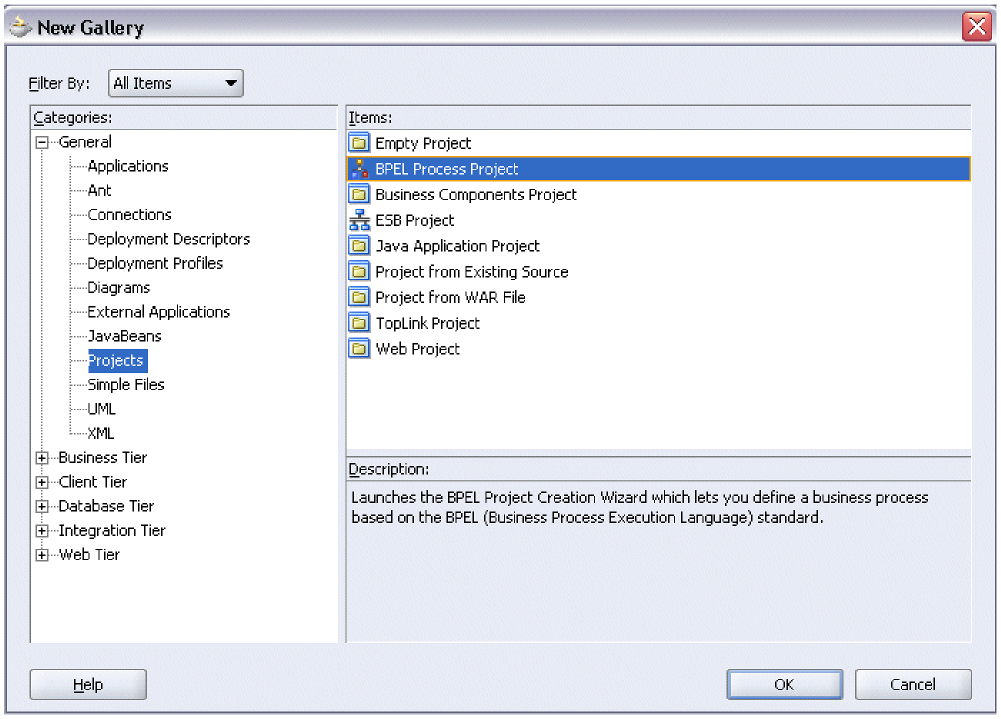
   The New Gallery dialog box appears.

3. Select **All Items** from the **Filter By** box. This displays a list of available categories.

4. Expand the **General** node, and then select **Projects**.

5. Select **BPEL Process Project** from the **Items** group.

Creating a New BPEL Process Project



Launches the BPEL Project Creation Wizard which lets you define a business process based on the BPEL (Business Process Execution Language) standard.

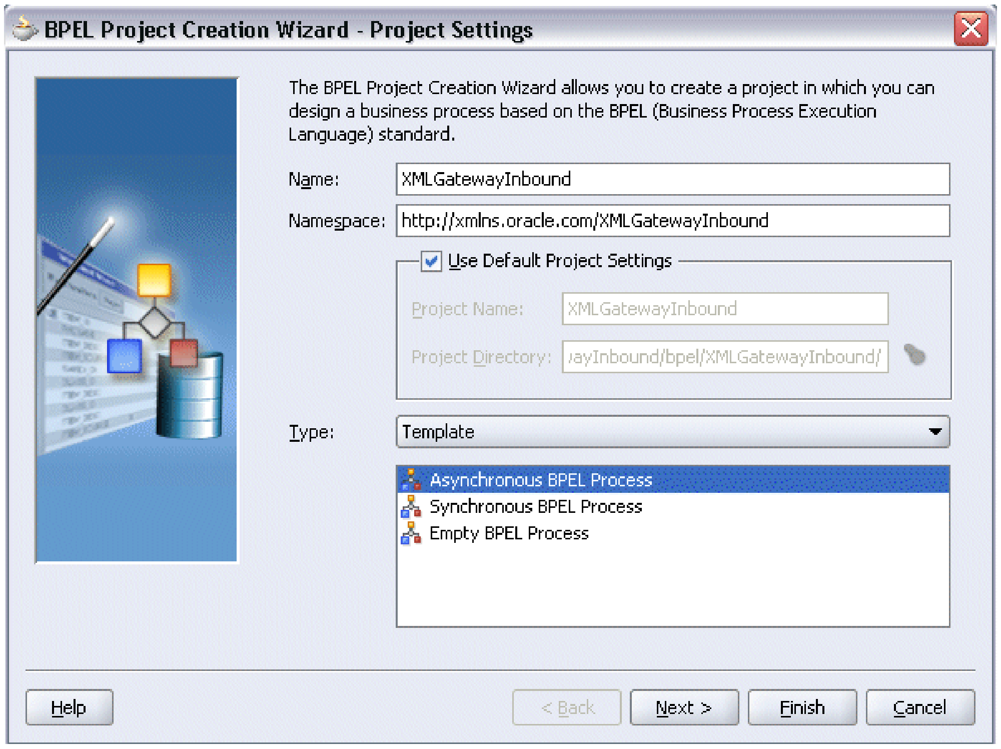
6. Click **OK**. The BPEL Project Creation Wizard - Project Settings dialog box appears.

7. In the **Name** field, enter a descriptive name, for example, `XMLGatewayInbound`.

   Keep the default selection **Use Default Project Settings** unchanged.

8. Keep the default selection **Template** as the Type field. Select **Asynchronous BPEL Process** as the BPEL process type.

Specifying New BPEL Project Settings


BPEL Project Creation Wizard - Project Settings

The BPEL Project Creation Wizard allows you to create a project in which you can design a business process based on the BPEL (Business Process Execution Language) standard.

Name: XMLGatewayInbound

Namespace: http://xmlns.oracle.com/XMLGatewayInbound

[✓] Use Default Project Settings

Project Name: XMLGatewayInbound

Project Directory: ıayInbound/bpel/XMLGatewayInbound/

Type: Template

Asynchronous BPEL Process

Synchronous BPEL Process

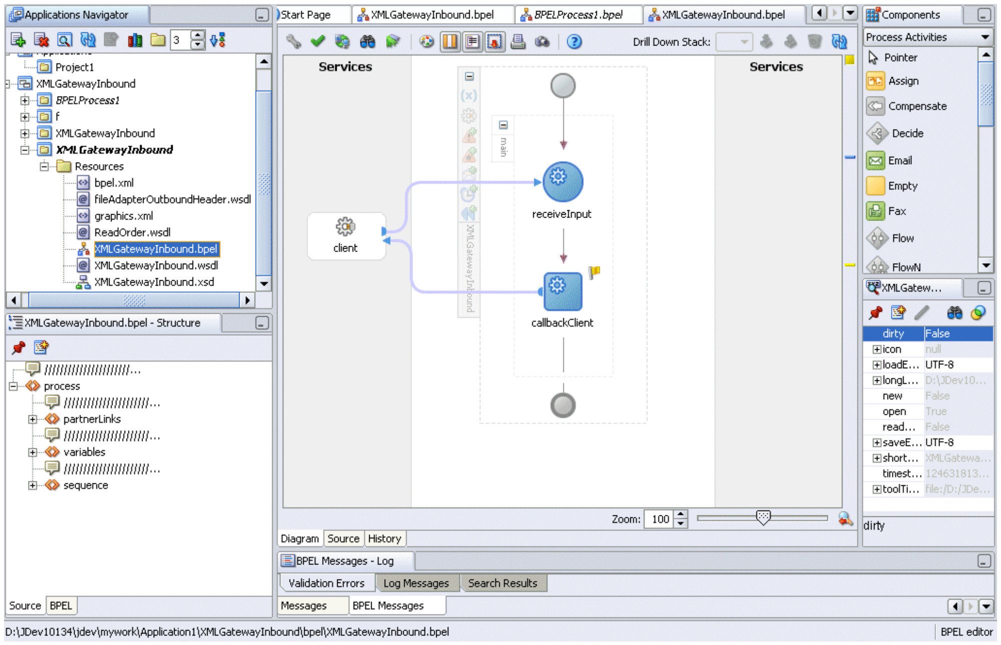
Empty BPEL Process

9. Click **Finish**.

A new asynchronous BPEL process is automatically created with the receiveInput and callbackClient activities.

The required source files including `bpel.xml`, `XMLGInbound.bpel`, and `XMLGInbound.wsdl` are also generated.

New BPEL Process Project



Creating a Partner Link

The next task is to add a partner link to the BPEL process. A partner link defines the link name, type, and the role of the BPEL process that interacts with the partner service.

To create a partner link:

1. Click **Services** in the Component palette.

   Drag and drop **Oracle Applications** from the Component palette, into the border area of the process diagram. The Adapter Configuration Wizard Welcome page appears.

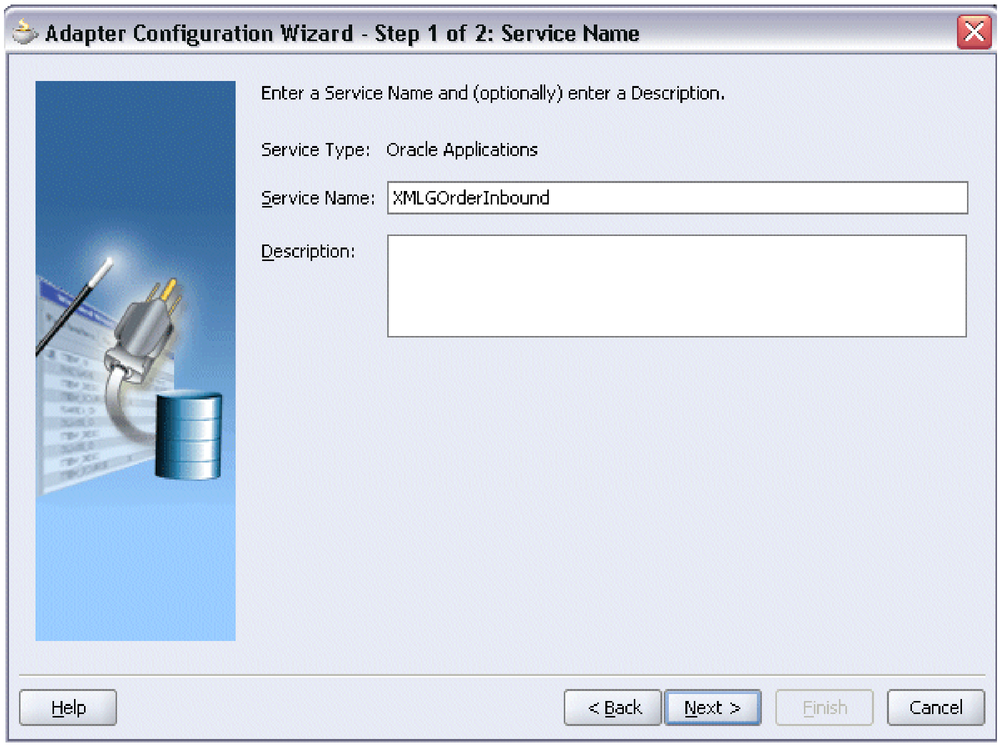
   Click **Next**.

2. The Service Name dialog box appears. Enter the following information:

   1. In the **Service Name** field, enter a service name. For example, enter `XMLGOrderInbound`.

   2. In the **Description** field, enter a description for the service. This is an optional field.

Specifying the Service Name


Adapter Configuration Wizard - Step 1 of 2: Service Name

Enter a Service Name and (optionally) enter a Description.

Service Type: Oracle Applications

Service Name: XMLGOrderInbound

Description:

3. You can either create a new database connection or use an existing connection.

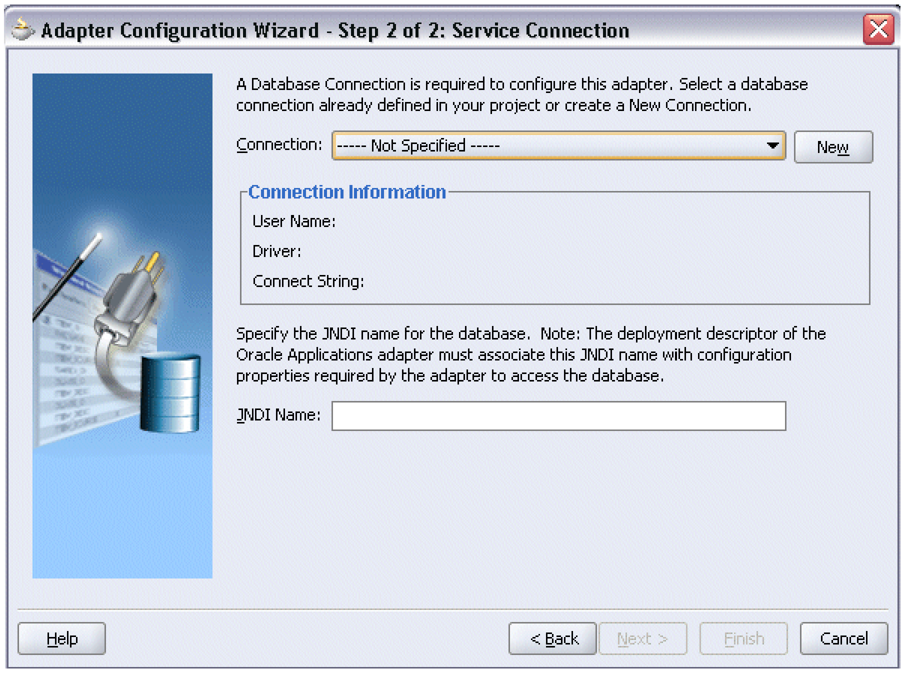
   Note: You need to connect to the database where Oracle E-Business Suite is running.

- **Creating a New Database Connection:**

  Perform the following steps to create a new database connection:

  1. Click **New** in the Service Connection dialog box.

Creating a New Database Connection


Adapter Configuration Wizard - Step 2 of 2: Service Connection

A Database Connection is required to configure this adapter. Select a database connection already defined in your project or create a New Connection.

Connection: ----- Not Specified -----

Connection Information

User Name:

Driver:

Connect String:

Specify the JNDI name for the database. Note: The deployment descriptor of the Oracle Applications adapter must associate this JNDI name with configuration properties required by the adapter to access the database.

JNDI Name:

The Create Database Connection Wizard appears.

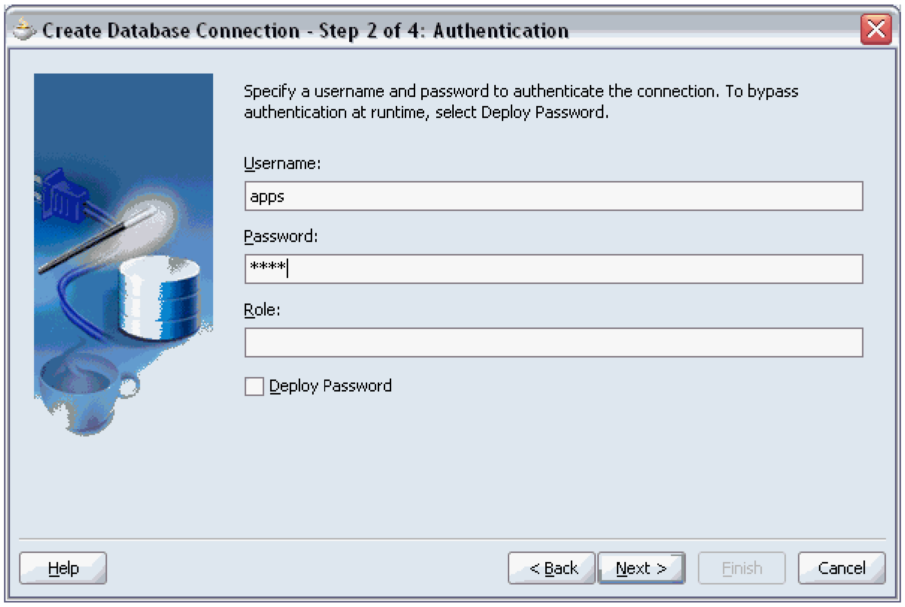
2. Enter the following information in the Type dialog box:

    1. In the **Connection Name** field, specify a unique name for the database connection.

    2. From the **Connection Type** box, select the type of connection for your database connection.

3. Click **Next**. The Authentication dialog box appears.

Authenticating the Database Connection



Create Database Connection - Step 2 of 4: Authentication

Specify a username and password to authenticate the connection. To bypass authentication at runtime, select Deploy Password.

Username: apps

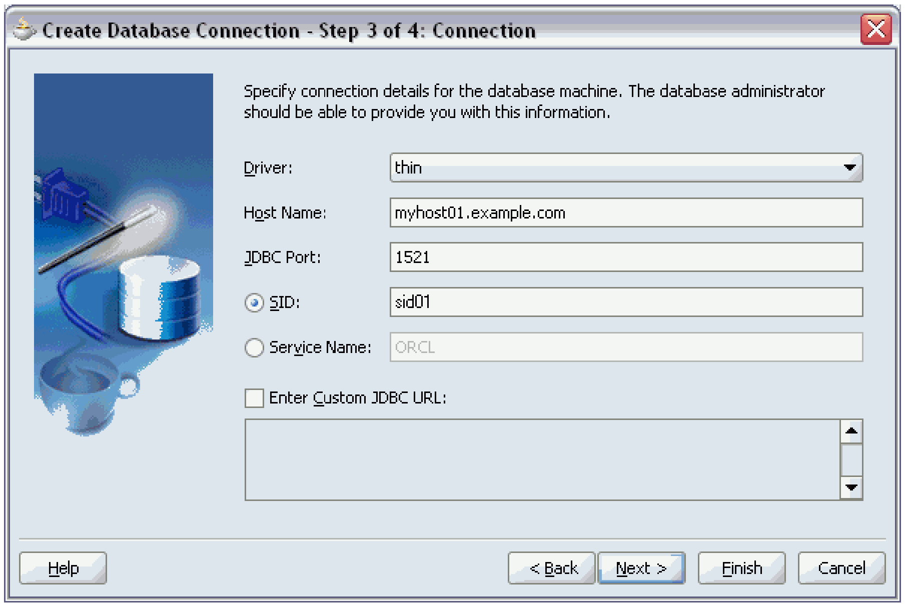
Password: ****

Role:

Deploy Password

4. Enter information in the following fields:
   1. In the **UserName** field, specify a unique name for the database connection.
   2. In the **Password** field, specify a password for the database connection.
5. Click **Next**. The Connection dialog box appears.

Providing Database Connection Details



Specify connection details for the database machine. The database administrator should be able to provide you with this information.

Driver: thin

Host Name: myhost01.example.com

JDBC Port: 1521

(●) SID: sid01

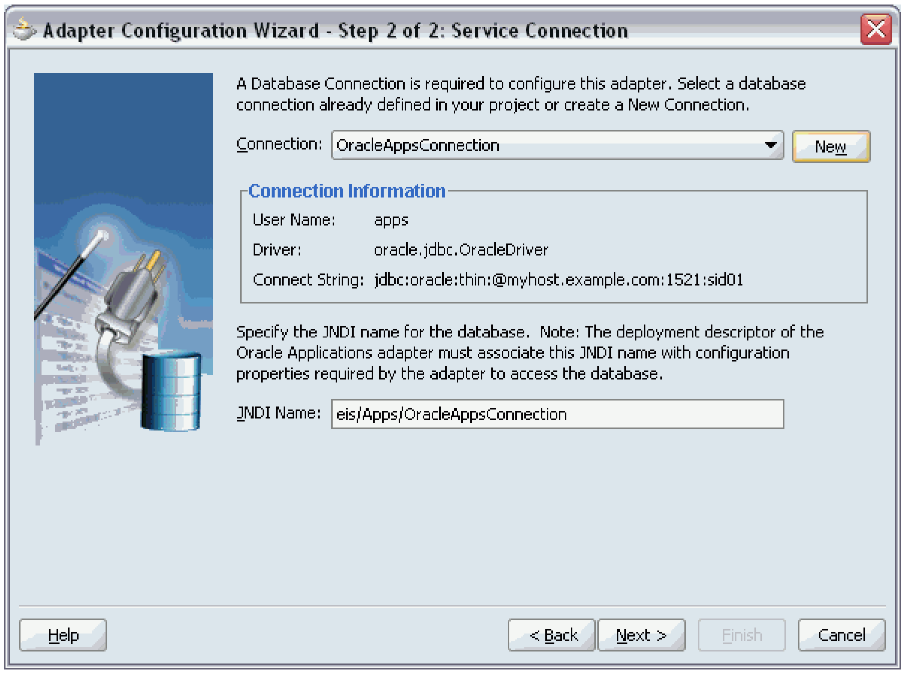
( ) Service Name: ORCL

☐ Enter Custom JDBC URL:

6. Enter information in the following fields:
   - From the **Driver** list, select **Thin**.
   - Enter the host name for the database connection, such as `myhost01.example.com`.
   - Enter the JDBC port number `1521` for the database connection.
   - Select **SID** and specify a unique SID value for the database connection, such as `sid01`.
7. Click **Next**. The Test dialog appears.
8. Click **Test Connection** to determine whether the specified information establishes a connection with the database. The status message "`Success!`" indicates a valid connection.

   Click **Finish** to complete the process of creating a new database connection.
9. The Service Connection dialog box appears, providing a summary of the database connection.

Verifying the Database Service Connection


Adapter Configuration Wizard - Step 2 of 2: Service Connection

A Database Connection is required to configure this adapter. Select a database connection already defined in your project or create a New Connection.

Connection: OracleAppsConnection

Connection Information

User Name: apps

Driver: oracle.jdbc.OracleDriver

Connect String: jdbc:oracle:thin:@myhost.example.com:1521:sid01

Specify the JNDI name for the database. Note: The deployment descriptor of the Oracle Applications adapter must associate this JNDI name with configuration properties required by the adapter to access the database.

JNDI Name: eis/Apps/OracleAppsConnection

The JNDI (Java Naming and Directory Interface) name corresponding to the database connection appears automatically in the **Database Server JNDI Name** field. Alternatively, you can specify a JNDI name.

> **Note:** When you specify a JNDI name, the deployment descriptor of the Adapter for Oracle Applications must associate this JNDI name with configuration properties required by the adapter to access the database.

The JNDI name acts as a placeholder for the connection used when your service is deployed to the BPEL server. This enables you to use different databases for development and later for production.

> **Note:** For more information about JNDI concepts, see *Oracle Application Server Adapter Concepts*.

- **Selecting an Existing Database Connection:**

  Instead of creating a new database connection, you can use an existing database connection that you have configured.

  1. From the Service Connection dialog box, select an existing database

connection from the Connection drop-down list.

2. The selected database connection information is displayed. The JNDI (Java Naming and Directory Interface) name corresponding to the selected database connection also appears automatically in the Database Server JNDI Name field. Alternatively, you can specify a JNDI name.

4. Once you have completed a new database connection or selected an existing connection, you can add an XML Gateway inbound map by browsing through the message maps available in Oracle E-Business Suite.

5. Click **Next** in the Service Connection dialog box.

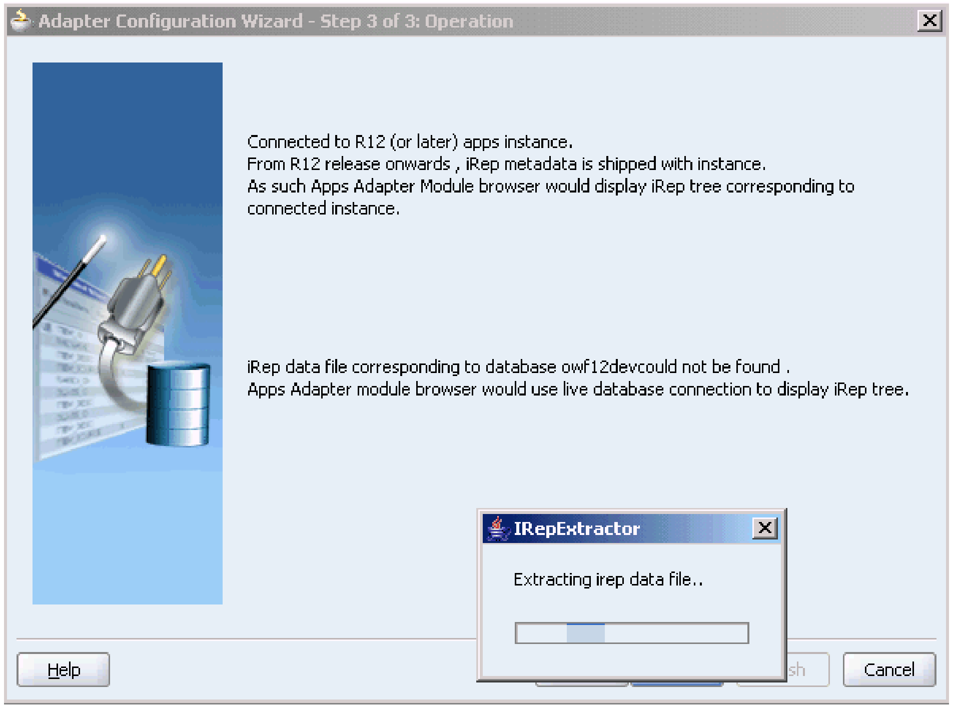
   For Oracle E-Business Suite Release 12:

   If you are connecting to Oracle E-Business Suite Release 12, then the **IREP File not present** dialog box appears indicating that OracleAS Adapter for Oracle Applications could not find the Oracle Integration Repository data file corresponding to the database you are connecting in your workspace. Absence of the data file would make browsing or searching of Integration Repository tree considerably slow. You can choose to extract the data file and create a local copy of the Integration Repository data file. Once it is created successfully, OracleAS Adapter for Oracle Applications will pick it up automatically next time and retrieve data from your local Integration Repository.

   You can select one of the following options:

   - Click **Yes** to extract the Integration Repository data file.

Extracting Integration Repository Data File



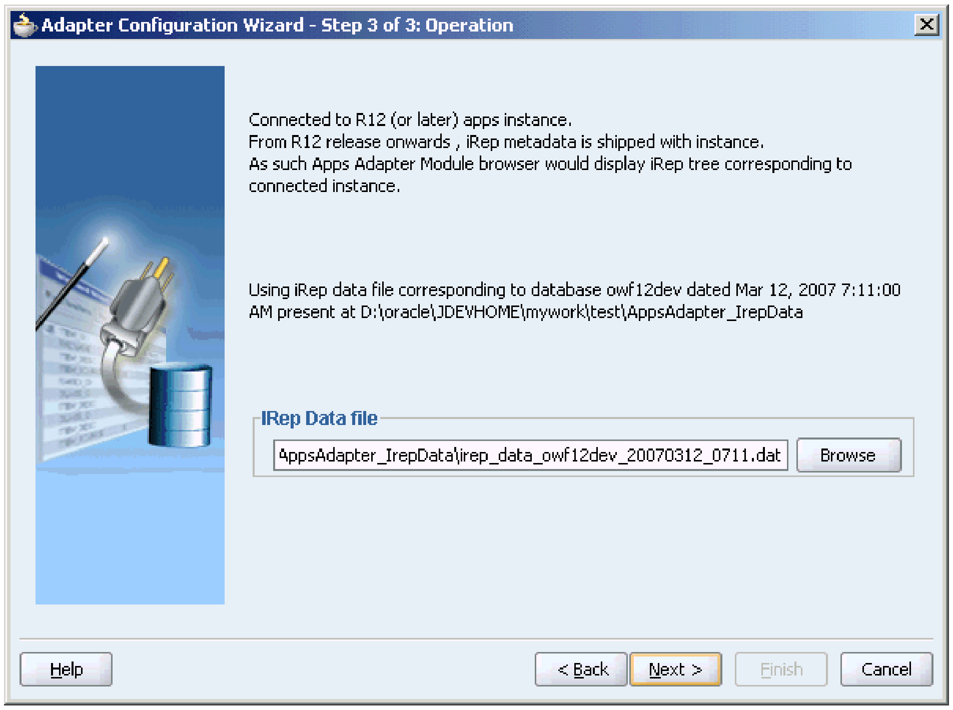
Connected to R12 (or later) apps instance.
From R12 release onwards , iRep metadata is shipped with instance.
As such Apps Adapter Module browser would display iRep tree corresponding to connected instance.

iRep data file corresponding to database owf12devcould not be found .
Apps Adapter module browser would use live database connection to display iRep tree.

Extracting irep data file..

After the system successfully creates a local copy of the Integration Repository data file, next time when you connect to the database, you will find the **IRep Data File** field appears in the Operation dialog box indicating where your local copy exists with the creation date and time as part of the file name.

Using the Local Integration Repository Data File


Adapter Configuration Wizard - Step 3 of 3: Operation

Connected to R12 (or later) apps instance.
From R12 release onwards , iRep metadata is shipped with instance.
As such Apps Adapter Module browser would display iRep tree corresponding to connected instance.

Using iRep data file corresponding to database owf12dev dated Mar 12, 2007 7:11:00 AM present at D:\oracle\JDEVHOME\mywork\test\AppsAdapter_IrepData

IRep Data file: AppsAdapter_IrepData\irep_data_owf12dev_20070312_0711.dat

- Click **No** to query the Integration Repository data file from the live database you are connecting to display the Integration Repository tree.

  Note: It is highly recommended that you create a local copy of the Integration Repository data file so that OracleAS Adapter for Oracle Applications will query the data next time from the local copy in your workspace to enhance the performance.

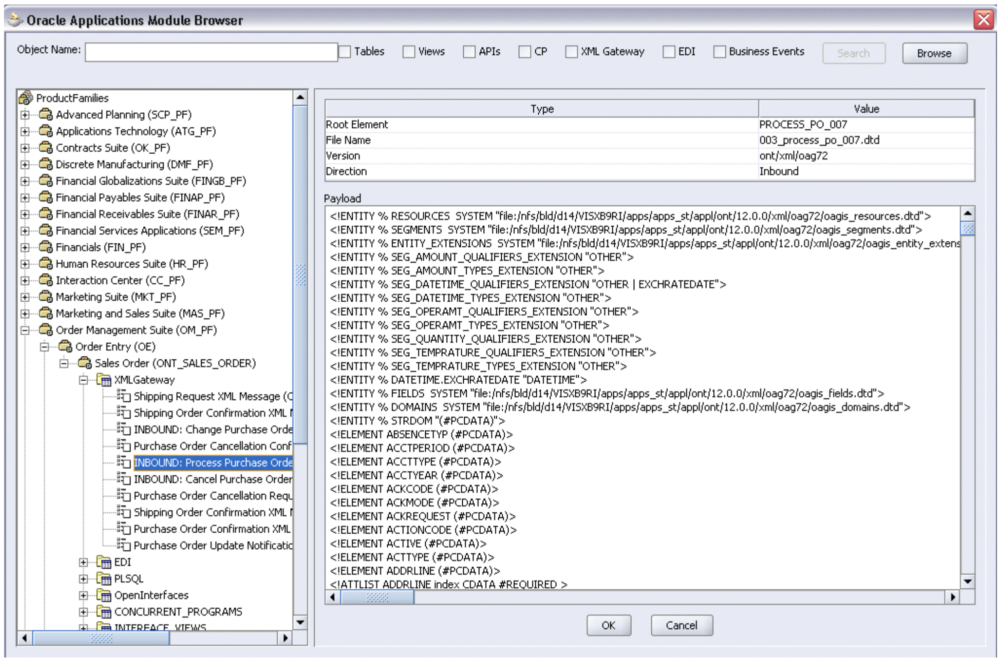
For Oracle E-Business Suite pre-Release 11.5.10:

If you are connecting to a pre-11.5.10 Oracle E-Business Suite instance, you must select the interface type in the Adapter Configuration Wizard. Select **XML Gateway** to proceed.

Click **Add** to open the Oracle Applications Module Browser.

6. The Oracle Applications Module Browser combines interface data from Oracle Integration Repository with information about the additional interfaces supported by OracleAS Adapter for Oracle Applications, organized in a tree hierarchy.

Specifying the XML Gateway Message Map


Oracle Applications Module Browser
Object Name:
Tables
Views
APIs
CP
XML Gateway
EDI
Business Events
Search
Browse
ProductFamilies
Advanced Planning (SCP_PF)
Applications Technology (ATG_PF)
Contracts Suite (OK_PF)
Discrete Manufacturing (DMF_PF)
Financial Globalizations Suite (FINGB_PF)
Financial Payables Suite (FINAP_PF)
Financial Receivables Suite (FINAR_PF)
Financial Services Applications (SEM_PF)
Financials (FIN_PF)
Human Resources Suite (HR_PF)
Interaction Center (CC_PF)
Marketing Suite (MKT_PF)
Marketing and Sales Suite (MAS_PF)
Order Management Suite (OM_PF)
Order Entry (OE)
Sales Order (ONT_SALES_ORDER)
XMLGateway
Shipping Request XML Message (C
Shipping Order Confirmation XML I
INBOUND: Change Purchase Orde
Purchase Order Cancellation Conf
INBOUND: Process Purchase Orde
INBOUND: Cancel Purchase Order
Purchase Order Cancellation Requ
Shipping Order Confirmation XML I
Purchase Order Confirmation XML
Purchase Order Update Notificatio
EDI
PLSQL
OpenInterfaces
CONCURRENT_PROGRAMS
INTERFACE_VIEWS
Payload
OK
Cancel

Type	Value
Root Element	PROCESS_PO_007
File Name	003_process_po_007.dtd
Version	ont/xml/oag72
Direction	Inbound

```
<!ENTITY % RESOURCES  SYSTEM "file:/nfs/bld/d14/VISXB9RI/apps/apps_st/appl/ont/12.0.0/xml/oag72/oagis_resources.dtd">
<!ENTITY % SEGMENTS  SYSTEM "file:/nfs/bld/d14/VISXB9RI/apps/apps_st/appl/ont/12.0.0/xml/oag72/oagis_segments.dtd">
<!ENTITY % ENTITY_EXTENSIONS  SYSTEM "file:/nfs/bld/d14/VISXB9RI/apps/apps_st/appl/ont/12.0.0/xml/oag72/oagis_entity_extens
<!ENTITY % SEG_AMOUNT_QUALIFIERS_EXTENSION "OTHER">
<!ENTITY % SEG_AMOUNT_TYPES_EXTENSION "OTHER">
<!ENTITY % SEG_DATETIME_QUALIFIERS_EXTENSION "OTHER | EXCHRATEDATE">
<!ENTITY % SEG_DATETIME_TYPES_EXTENSION "OTHER">
<!ENTITY % SEG_OPERAMT_QUALIFIERS_EXTENSION "OTHER">
<!ENTITY % SEG_OPERAMT_TYPES_EXTENSION "OTHER">
<!ENTITY % SEG_QUANTITY_QUALIFIERS_EXTENSION "OTHER">
<!ENTITY % SEG_TEMPRATURE_QUALIFIERS_EXTENSION "OTHER">
<!ENTITY % SEG_TEMPRATURE_TYPES_EXTENSION "OTHER">
<!ENTITY % DATETIME.EXCHRATEDATE "DATETIME">
<!ENTITY % FIELDS  SYSTEM "file:/nfs/bld/d14/VISXB9RI/apps/apps_st/appl/ont/12.0.0/xml/oag72/oagis_fields.dtd">
<!ENTITY % DOMAINS  SYSTEM "file:/nfs/bld/d14/VISXB9RI/apps/apps_st/appl/ont/12.0.0/xml/oag72/oagis_domains.dtd">
<!ENTITY % STRDOM "(#PCDATA)">
<!ELEMENT ABSENCETYP (#PCDATA)>
<!ELEMENT ACCTPERIOD (#PCDATA)>
<!ELEMENT ACCTTYPE (#PCDATA)>
<!ELEMENT ACCTYEAR (#PCDATA)>
<!ELEMENT ACKCODE (#PCDATA)>
<!ELEMENT ACKMODE (#PCDATA)>
<!ELEMENT ACKREQUEST (#PCDATA)>
<!ELEMENT ACTIONCODE (#PCDATA)>
<!ELEMENT ACTIVE (#PCDATA)>
<!ELEMENT ACTTYPE (#PCDATA)>
<!ELEMENT ADDRLINE (#PCDATA)>
<!ATTLIST ADDRLINE index CDATA #REQUIRED >
```

Note: The Oracle Applications Module Browser includes the various product families that are available in Oracle E-Business Suite. Each product family contains the individual products. Each product contains the business entities associated with the product. Business entities contain the various application modules that are exposed for integration. These modules are grouped according to the interface they provide.

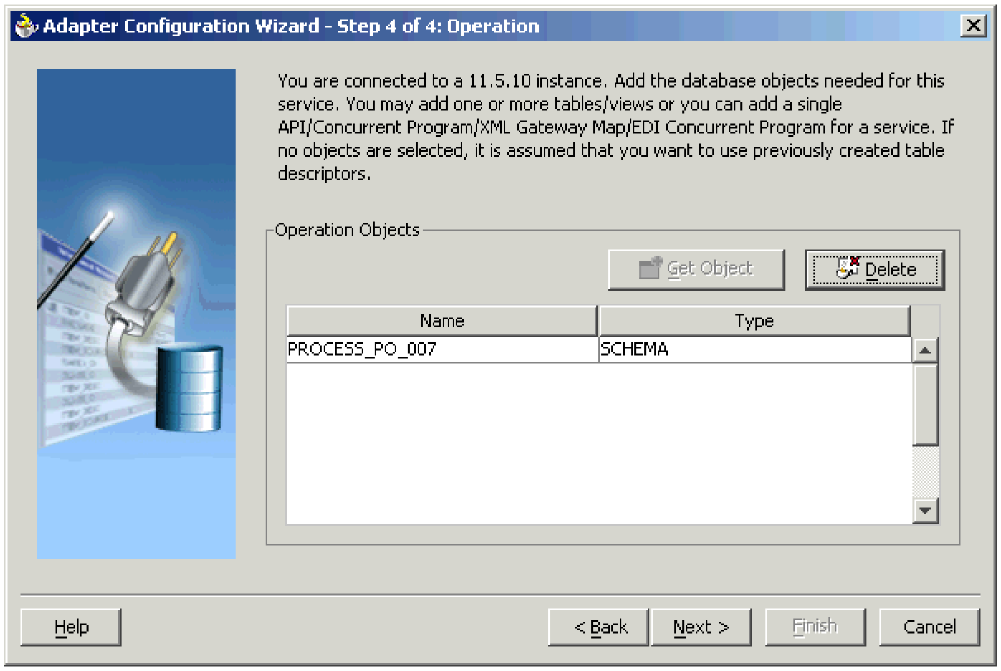
Navigate to *Order Management Suite > Order Management >Sales Order > XML Gateway* to select `Inbound: Process Purchase Order XML Transaction (ONT_3A4R_OAG72_IN).`

- You can also search for an XML Gateway message map by entering the name or part of the name for the message map in the **Object Name** field. Select the **XML Gateway** check box and click **Search**.

- The custom message maps that you might have saved can be found in the **Others** category.

7. Click **OK** to generate the XML schema.

Adding the XML Schema


Adapter Configuration Wizard - Step 4 of 4: Operation

You are connected to a 11.5.10 instance. Add the database objects needed for this service. You may add one or more tables/views or you can add a single API/Concurrent Program/XML Gateway Map/EDI Concurrent Program for a service. If no objects are selected, it is assumed that you want to use previously created table descriptors.

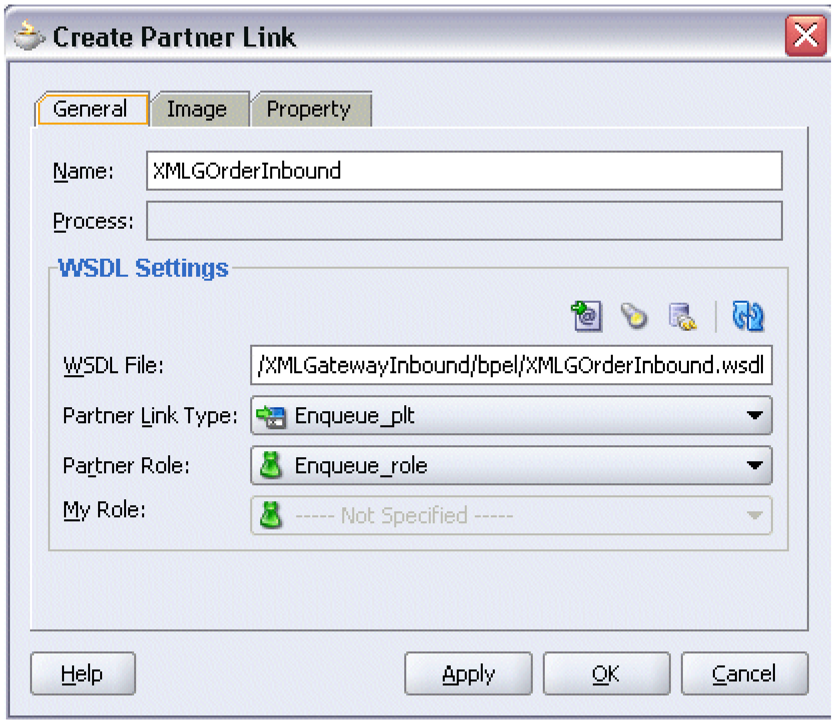
Operation Objects

Name	Type
PROCESS_PO_007	SCHEMA

8. Click **Next** and then click **Finish** to complete the process of configuring Adapter for Oracle Applications.

   The wizard generates the WSDL file corresponding to the XML schema. This WSDL file is now available for the partner link.

Completing the Partner Link Configuration


Create Partner Link

General | Image | Property

Name: XMLGOrderInbound

Process:

WSDL Settings

WSDL File: /XMLGatewayInbound/bpel/XMLGOrderInbound.wsdl

Partner Link Type: Enqueue_plt

Partner Role: Enqueue_role

My Role: ----- Not Specified -----

9. Click **Apply** and then **OK**. The partner link is created with the required WSDL settings.

Adding a Partner Link for File Adapter

Use this step to configure a BPEL process by adding the following two partner links for File Adapter:

1. To pick up an XML file received from the third party application to get the XML message.

2. To get the transaction information for the ECX header.

To add the first Partner Link for File Adapter to get the XML Message:

1. In JDeveloper BPEL Designer, drag and drop the **File Adapter** service from the **Service** section of the Component Palette into the Partner Link area of the process diagram. The Adapter Configuration Wizard welcome page appears.

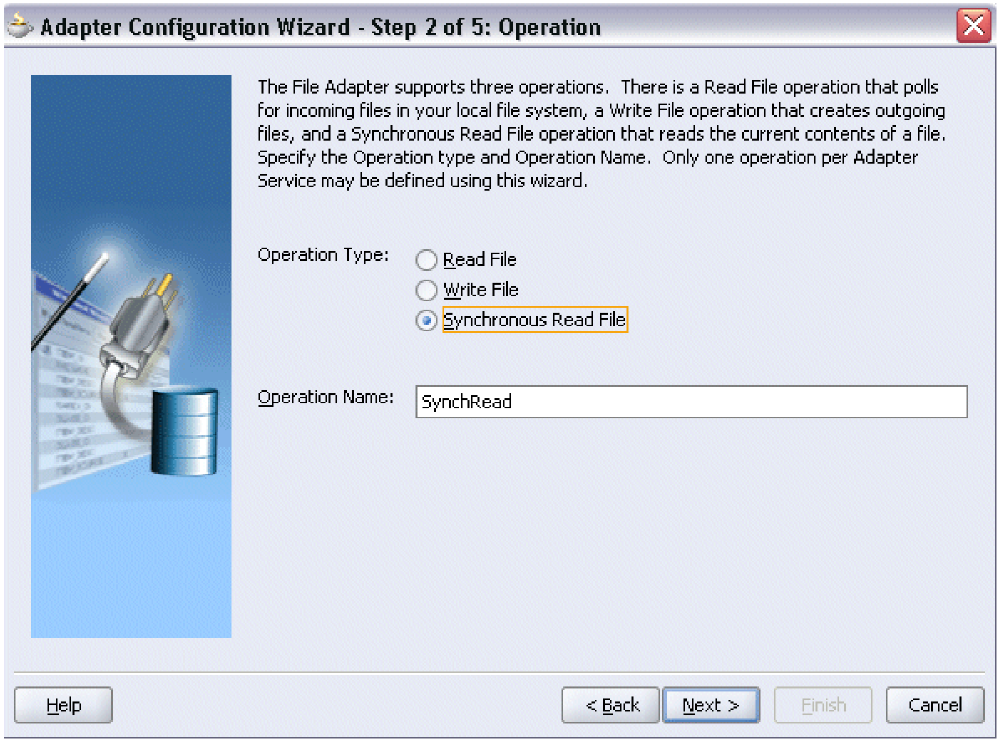
2. Click **Next**. The Service Name dialog box appears.

3. Enter a name for the file adapter service, such as `GetXMLMsg`. You can add an

optional description of the service.

4. Click **Next**, and the Operation dialog box appears.

Specifying the Operation


Adapter Configuration Wizard - Step 2 of 5: Operation

The File Adapter supports three operations. There is a Read File operation that polls for incoming files in your local file system, a Write File operation that creates outgoing files, and a Synchronous Read File operation that reads the current contents of a file. Specify the Operation type and Operation Name. Only one operation per Adapter Service may be defined using this wizard.

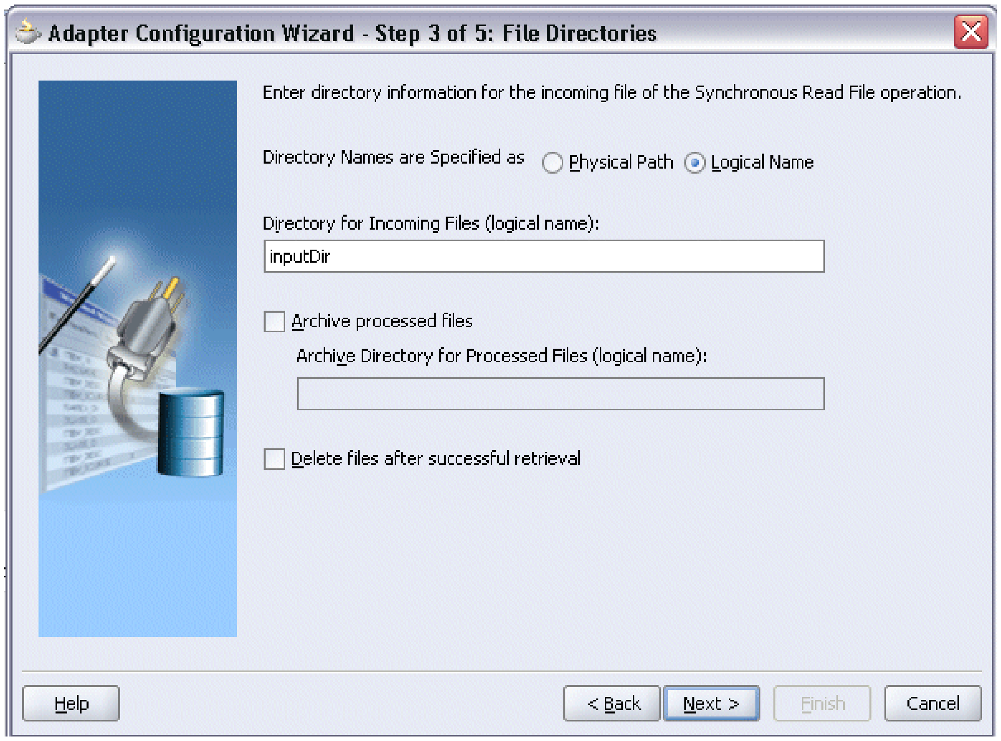
Operation Type: Read File / Write File / [✓ Synchronous Read File]

Operation Name: SynchRead

5. Specify the operation type, for example **Synchronous Read File**. This automatically populates the **Operation Name** field.

   Click **Next** to access the File Directories dialog box.

Configuring the Input File


Adapter Configuration Wizard - Step 3 of 5: File Directories

Enter directory information for the incoming file of the Synchronous Read File operation.

Directory Names are Specified as Physical Path [✓ Logical Name]

Directory for Incoming Files (logical name): inputDir

Archive processed files: ☐

Archive Directory for Processed Files (logical name):

Delete files after successful retrieval: ☐

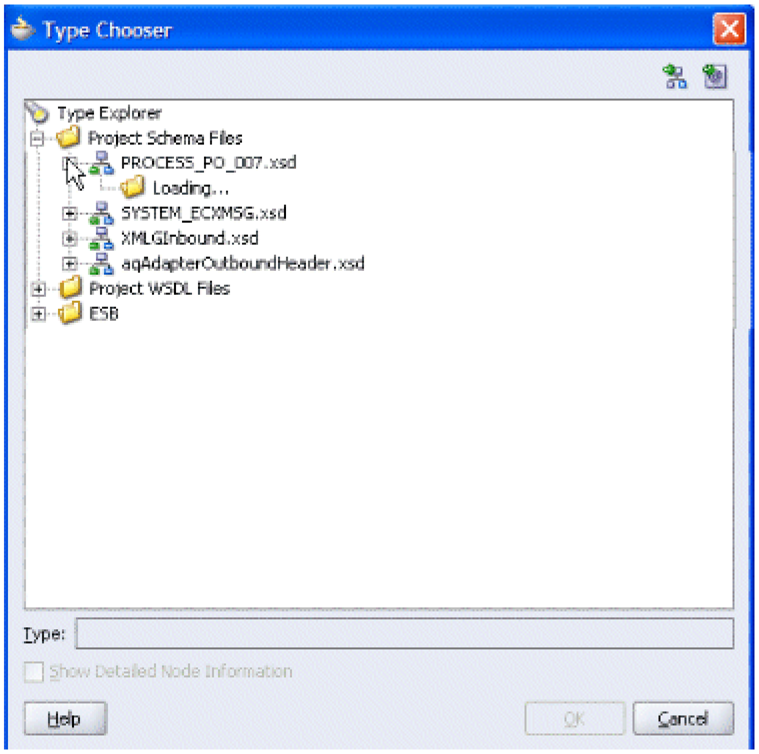
6. Select **Logical Name** check box and specify the logical directory for the incoming file. Uncheck the **Delete files after successful retrieval** check box. Click **Next**.

7. Enter the name of the file for the synchronous read file operation. For example, enter `order_data_xmlg.xml`. Click **Next**. The Messages dialog box appears.

8. Select **Browse** to open the Type Chooser to specify the Schema location and element.

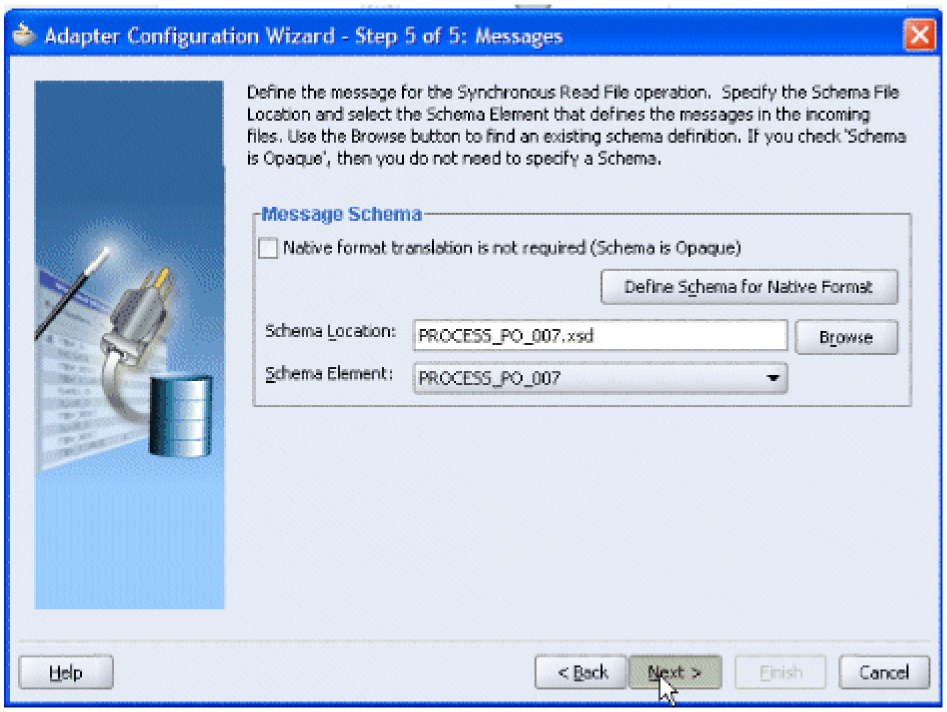
   From the Type Chooser window, expand the node by clicking Project Schema Files > PROCESS_PO_007.xsd > PROCESS_PO_007.

Selecting Schema from the Type Chooser



9. Click **OK** to populate the selected values in the Messages dialog box.

Populating Selected Message Schema and Element



Define the message for the Synchronous Read File operation. Specify the Schema File Location and select the Schema Element that defines the messages in the incoming files. Use the Browse button to find an existing schema definition. If you check 'Schema is Opaque', then you do not need to specify a Schema.

Message Schema

☐ Native format translation is not required (Schema is Opaque)

Schema Location: PROCESS_PO_007.xsd

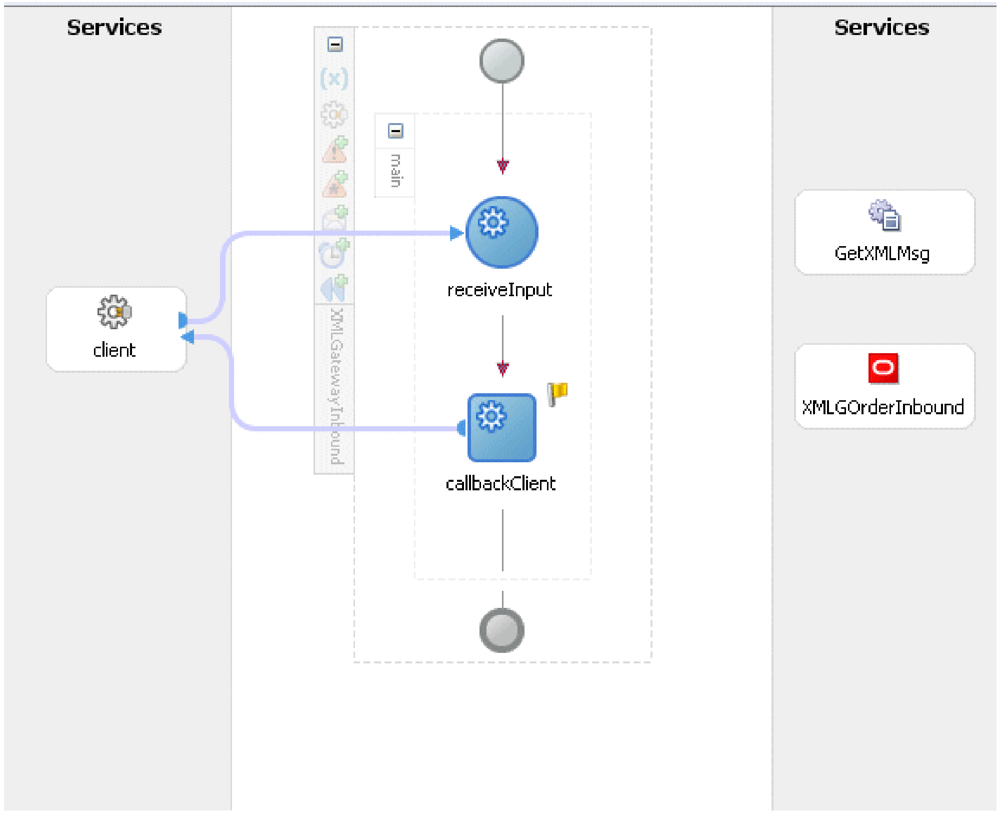
Schema Element: PROCESS_PO_007

10. Click **Next** and then **Finish**. The wizard generates the WSDL file corresponding to the partner link. The main Create Partner Link dialog box appears, specifying the new WSDL file `GetXMLMsg.wsdl`.

    Click **Apply** and then **OK** to complete the configuration and create the partner link with the required WSDL settings for the File Adapter Service.

    The `GetXMLMsg` Partner Link appears in the following BPEL process diagram:

Adding the Partner Link for File Adapter



To add the second Partner Link for File Adapter to get the transaction information for the XML header:

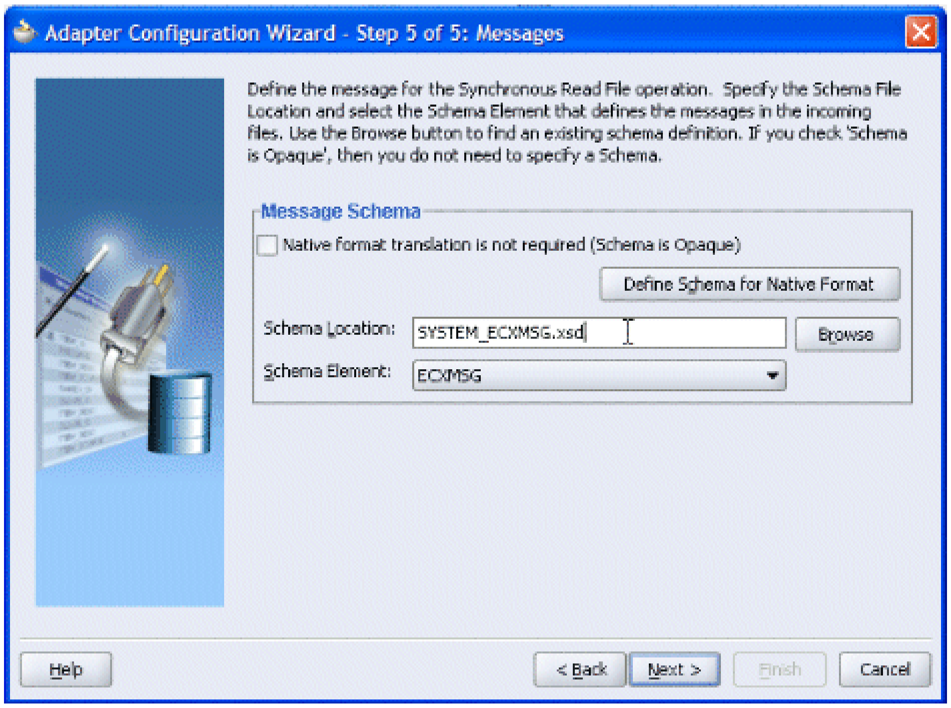
1. Repeat Step 1 to Step 6 mentioned in the first partner link creation for File Adapter to create the second partner link called `GetECXHeader`.

2. Enter the name of the file for the synchronous read file operation. For example, enter `ecx_header_data.xml`. Click **Next**. The Messages dialog box appears.

3. Select **Browse** to open the Type Chooser to specify the Schema location and element.

   From the Type Chooser window, expand the node by clicking Project Schema Files > SYSTEM_ECXMSG.xsd > SYSTEM_ECXMSG. Click **OK** to populate the selected schema location and element in the Messages dialog box.

Populated Selected Schema and Element in the Messages Dialog Box


Adapter Configuration Wizard - Step 5 of 5: Messages

Define the message for the Synchronous Read File operation. Specify the Schema File Location and select the Schema Element that defines the messages in the incoming files. Use the Browse button to find an existing schema definition. If you check 'Schema is Opaque', then you do not need to specify a Schema.

Message Schema

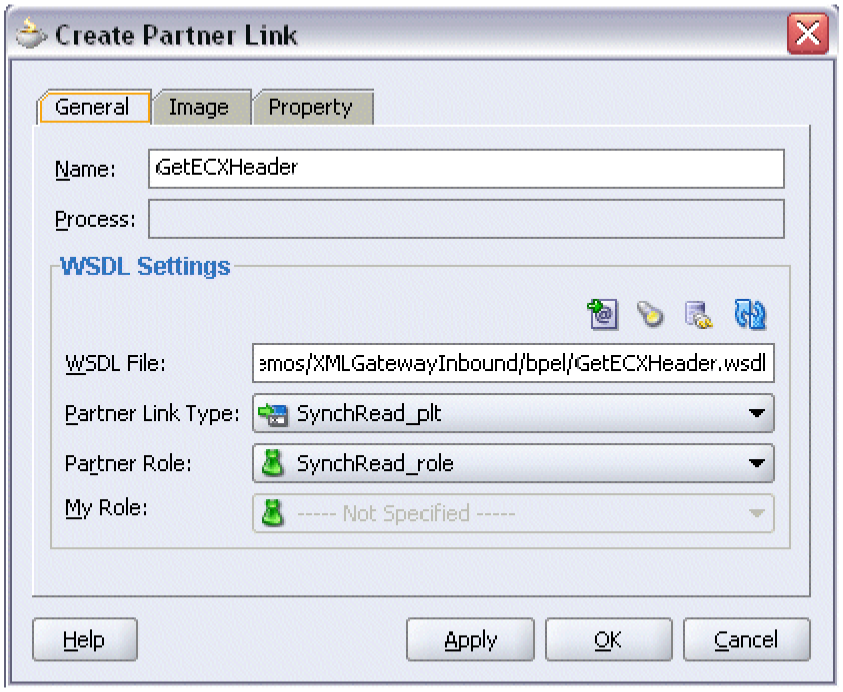
☐ Native format translation is not required (Schema is Opaque)

Schema Location: SYSTEM_ECXMSG.xsd

Schema Element: ECXMSG

4. Click **Next** and then **Finish**. The wizard generates the WSDL file corresponding to the partner link. The main Create Partner Link dialog box appears, specifying the new WSDL file `GetECXHeader.wsdl`.

Adding a Partner Link


Create Partner Link

Name: GetECXHeader

Process:

WSDL Settings

WSDL File: :mos/XMLGatewayInbound/bpel/GetECXHeader.wsdl

Partner Link Type: SynchRead_plt

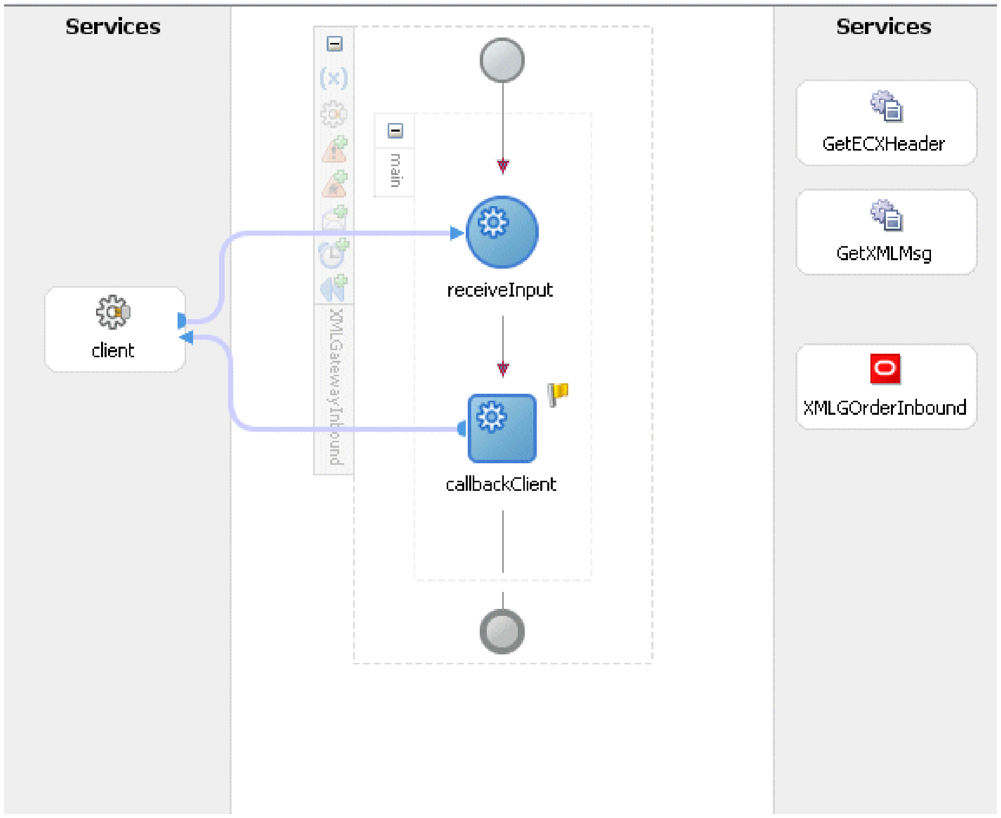
Partner Role: SynchRead_role

My Role: ----- Not Specified -----

Click **Apply** and then **OK** to complete the configuration and create the partner link with the required WSDL settings for the second File Adapter Service.

The `GetECXHeader` Partner Link appears in the following BPEL process diagram:

Adding the Partner Link for File Adapter



Configuring the Invoke Activity

This step is to configure three Invoke activities:

1. To get the XML message details that is received from the Receive activity by invoking the `GetXMLMsg` partner link in an XML file.

2. To get the ECX Header properties details by invoking `GetECXHeader` partner link in an XML file.

3. To enqueue the purchase order information to the ECX_INBOUND queue by invoking `XMLGOrderInbound` partner link in an XML file.

To add the first Invoke activity for a partner link to get XML message:

1. In JDeveloper BPEL Designer, drag and drop the first **Invoke** activity from the Component Palette into the Activity box of the process diagram, between the **Receive** and **Callback** activities.

2. Link the Invoke activity to the `GetXMLMsg` service. The Edit Invoke dialog box

appears.

3. Enter a name for the Invoke activity, and then click the **Create** icon next to the **Input Variable** field to create a new variable. The Create Variable dialog box appears.

4. Select **Global Variable**. Enter a name for the variable. You can also accept the default name. Click **OK**.

5. Enter a name for the Invoke activity, and then click the **Create** icon next to the **Output Variable** field to create a new variable. The Create Variable dialog box appears.

6. Select **Global Variable** and then enter a name for the variable. You can also accept the default name. Click **OK**.

   Click **Apply** and then **OK** in the Edit Invoke dialog box to finish configuring the Invoke activity.

   The Invoke activity appears in the process diagram.

To add the second Invoke activity for a partner link to get ECX Header properties:

1. In JDeveloper BPEL Designer, drag and drop the second **Invoke** activity from the Component Palette into the Activity box of the process diagram, between the first **Invoke** activity and the **Callback** activity.

2. Link the Invoke activity to the `GetECXHeader` service. The Edit Invoke dialog box appears.

3. Repeat Step 3 to Step 6 described in the first Invoke activity creation to complete the second Invoke activity.

To add the third Invoke activity for a partner link to enqueue PO information:

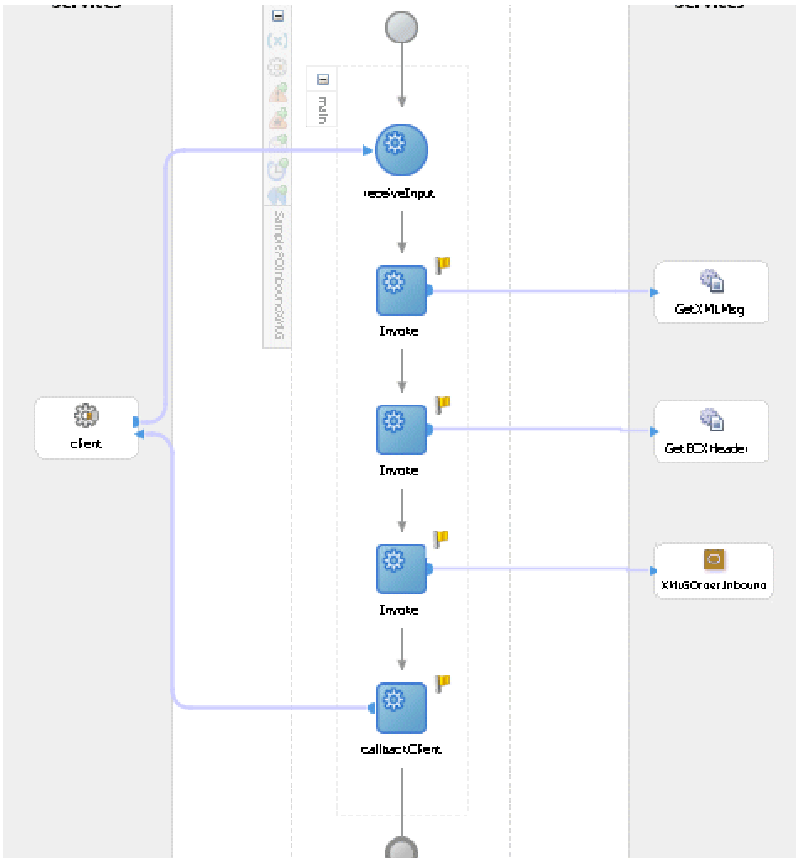
1. In JDeveloper BPEL Designer, drag and drop the third **Invoke** activity from the Component Palette into the process diagram, between the second **Invoke** activity and the **Callback** activity.

2. Link the Invoke activity to the `XMLGOrderInbound` service. The Edit Invoke dialog box appears.

3. Repeat Step 3 to Step 6 described in the first Invoke activity creation to complete the third Invoke activity.

   The three Invoke activities appear in the process diagram.

Adding Invoke Activities



Configuring the Assign Activity

This step is to configure two Assign activities:

1. To pass XML message as an input to the last Invoke activity for enqueuing message.
2. To pass ECX header variables as input variables to the last Invoke activity in order to provide context information for Oracle Applications.

To add the first Assign activity to pass XML message as input to the Invoke activity:

1. In JDeveloper BPEL Designer, drag and drop the **Assign** activity from the Component Palette into the Activity box of the process diagram, between the second and the third **Invoke** activities.
2. Double-click the **Assign** activity to access the Edit Assign dialog box.
3. Click the General tab to enter the name for the Assign activity, such as 'SetXMLMsg'.

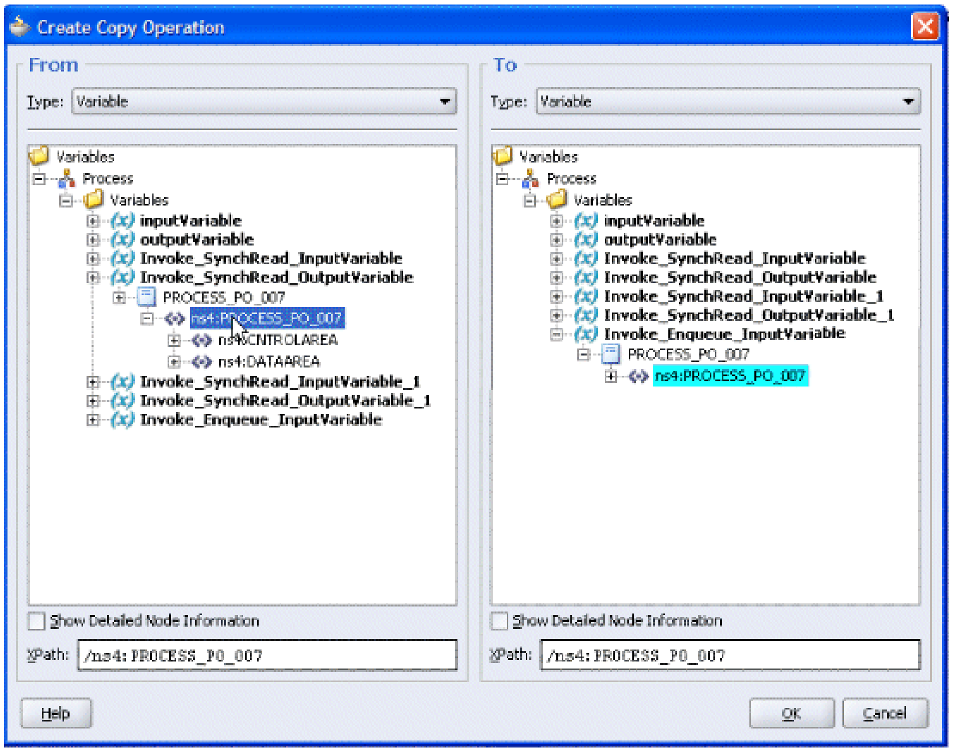
4. On the Copy Operation tab, click **Create**, then select **Copy Operation** from the menu. The Create Copy Operation window appears.

5. Enter the first pair of parameters:

   - In the From navigation tree, select type Variable and then navigate to **Variable > Process > Variables > Invoke_SynchRead_OutputVariable > Process_PO_007** and select **ns4:Process_PO_007**.

     The XPath field should contain your selected entry.

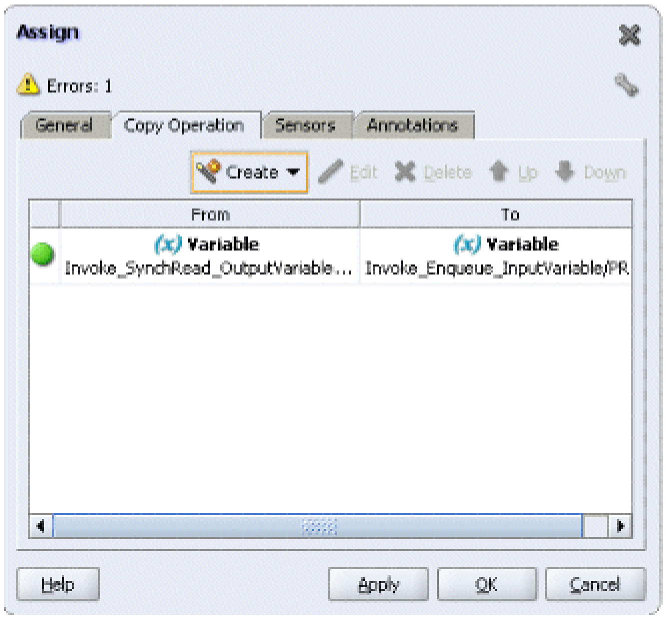
   - In the To navigation tree, select type Variable and then navigate to **Variable > Process > Variables > Invoke_Enqueue_InputVariable > Process_PO_007** and select **ns4:Process_PO_007**. The XPath field should contain your selected entry.


Create Copy Operation
From
Type: Variable
Variables
Process
Variables
inputVariable
outputVariable
Invoke_SynchRead_InputVariable
Invoke_SynchRead_OutputVariable
PROCESS_PO_007
ns4:PROCESS_PO_007
ns4:CNTROLAREA
ns4:DATAAREA
Invoke_SynchRead_InputVariable_1
Invoke_SynchRead_OutputVariable_1
Invoke_Enqueue_InputVariable
Show Detailed Node Information
XPath: /ns4:PROCESS_PO_007
To
Type: Variable
Variables
Process
Variables
inputVariable
outputVariable
Invoke_SynchRead_InputVariable
Invoke_SynchRead_OutputVariable
Invoke_SynchRead_InputVariable_1
Invoke_SynchRead_OutputVariable_1
Invoke_Enqueue_InputVariable
PROCESS_PO_007
ns4:PROCESS_PO_007
Show Detailed Node Information
XPath: /ns4:PROCESS_PO_007
Help
OK
Cancel

   - Click **OK**.

6. The Edit Assign dialog box appears.

Assign Parameters



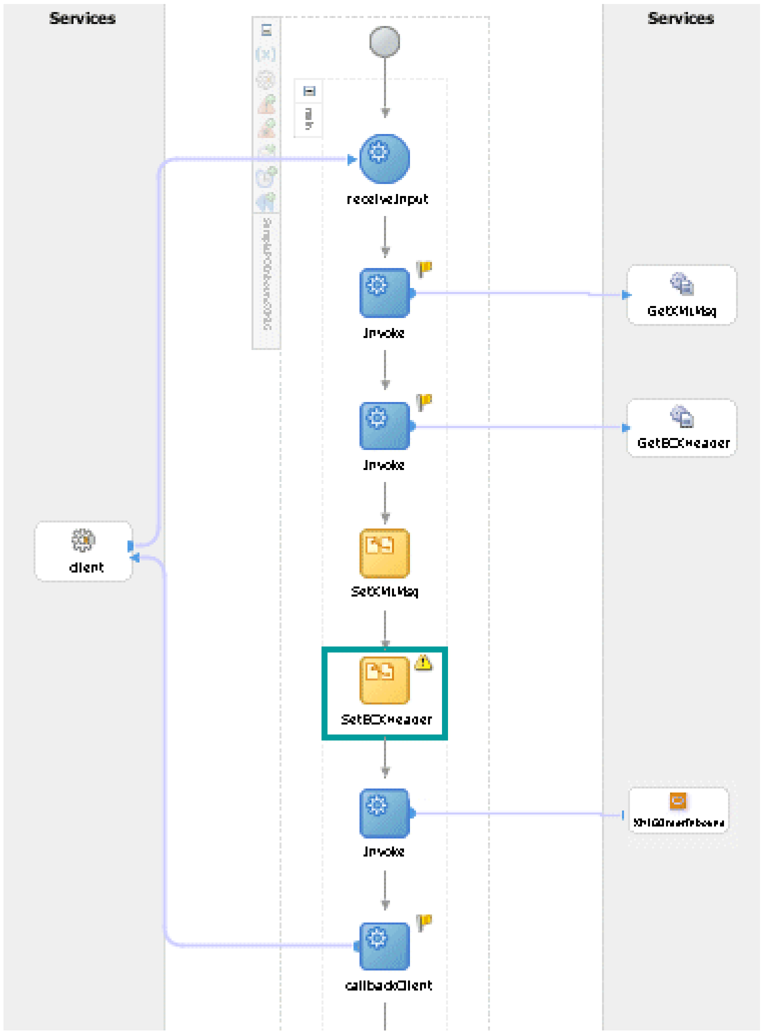
	From	To
●	(x) Variable Invoke_SynchRead_OutputVariable...	(x) Variable Invoke_Enqueue_InputVariable/PR

7. Click **Apply** and then **OK** to complete the configuration of the Assign activity.

To add the second Assign activity to pass ECX header variables to the last Invoke activity:

1. Add the second Assign activity by dragging and dropping the **Assign** activity from the Component Palette into the Activity box of the process diagram, between the `SetXMLMsg` **Assign** activity and the last **Invoke** activity.

Adding an Assign Activity

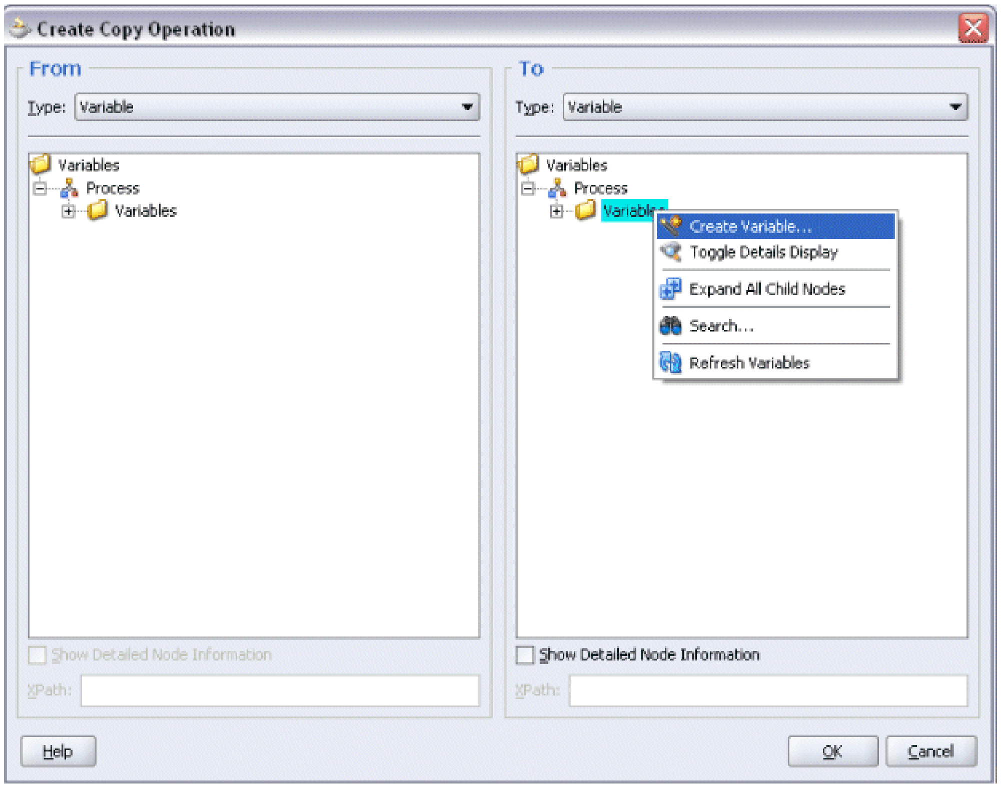


2. Repeat Step 2 to Step 4 described in creating the first Assign activity to add the second Assign activity called 'SetECXHeader'.

3. On the Copy Operation tab, click **Create** and then select **Copy Operation** from the menu. The Create Copy Operation window appears.

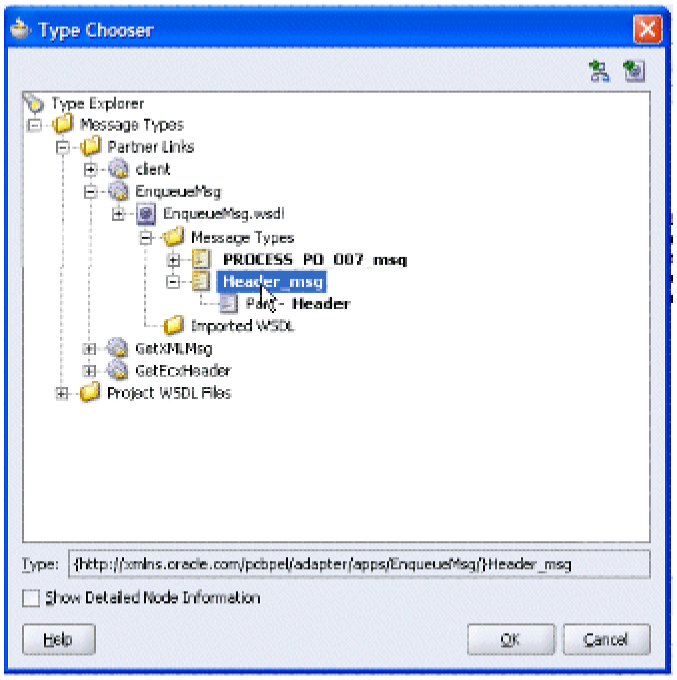
4. In the To navigation tree, select the **Variables** node and right-click on mouse to select 'Create Variable' from the drop-down list.

Creating Variables



The Create Variable dialog box opens. Select the **Message Type** check box and click the **Browse** icon to open the Type Chooser window.

Declaring Header Variables



Type: {http://xmlns.oracle.com/pcbpel/adapter/apps/EnqueueMsg/}Header_msg

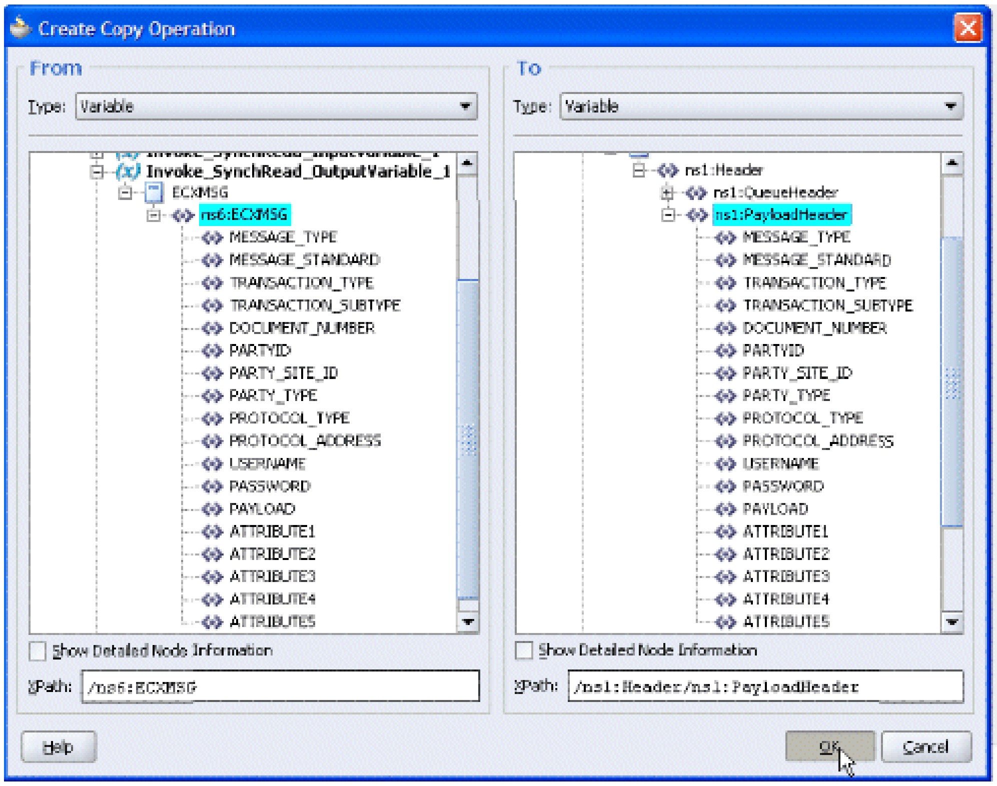
Expand the **Message Type** node and select **Partner Links > EnqueueMsg > EnqueueMsg.wsdl > Message Types > Header_msg**. Click **OK** to return to the Create Copy Operation dialog box.

5. The **ECXHeader** node appears in the Variables node of the To navigation tree. Expand the **ECXHeader** node and select **ns1: Header > ns1: PayloadHeader**.

6. In the From navigation tree, select type Variable. Navigate to **Variable > Process > Variables > Invoke_SynchRead_OutputVariable_1 > ECXMSG** and select **ns6:ECXMSG**.

   The XPath field should contain your selected entry.

Assigning Header Variables

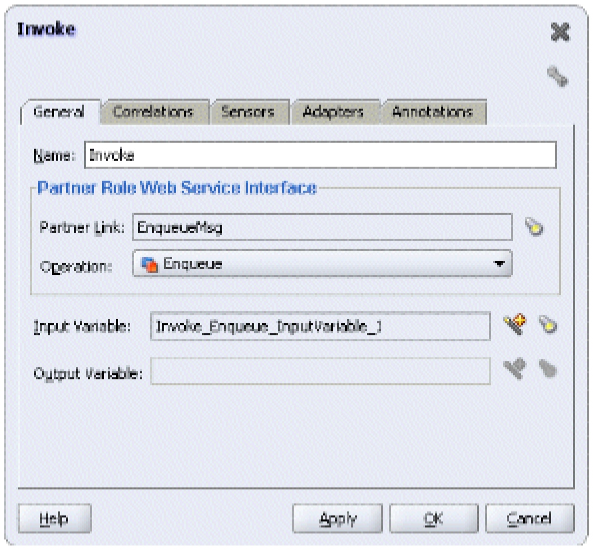


7. Click **OK** to complete the configuration of the Assign activity.

8. Modify the Invoke activity to pass the **ECXHeader** as input header variables you just declared and assigned through the Assign activity.

   1. Double-click the last Invoke activity to open the Invoke edit window.

Editing Invoke Activity



Name: Invoke

Partner Role Web Service Interface

Partner Link: EnqueueMsg

Operation: Enqueue

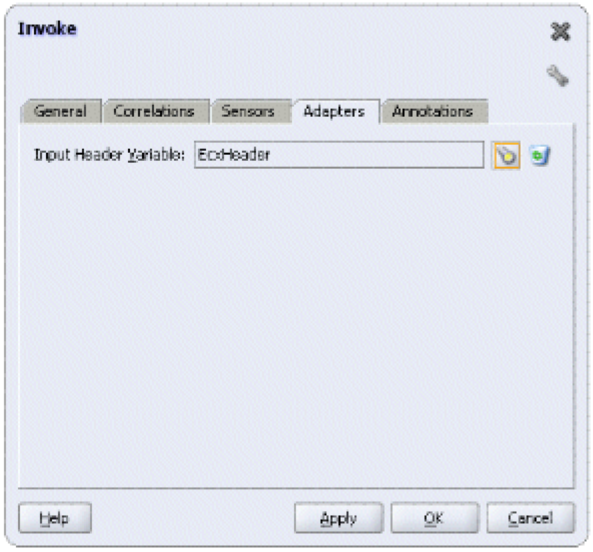
Input Variable: Invoke_Enqueue_InputVariable_1

Output Variable:

2. Select Adapters tab and click the **Browse Variables...** icon next to the Input Header Variables field.

3. Select ECXHeader from the Variable Chooser. Click **OK**.

Entering Input Header Variables



Input Header Variable: EcxHeader

4. The selected ECXHeader is now populated in the Input Header Variables field. Click **Apply** and then **OK**.

Run-Time Tasks for XML Gateway Inbound Messaging

After designing the BPEL process, the next step is to deploy, run and monitor it.

1. Deploy the BPEL process, page 4-40

2. Test the BPEL process, page 4-42

3. Verify records in Oracle Applications, page 4-44

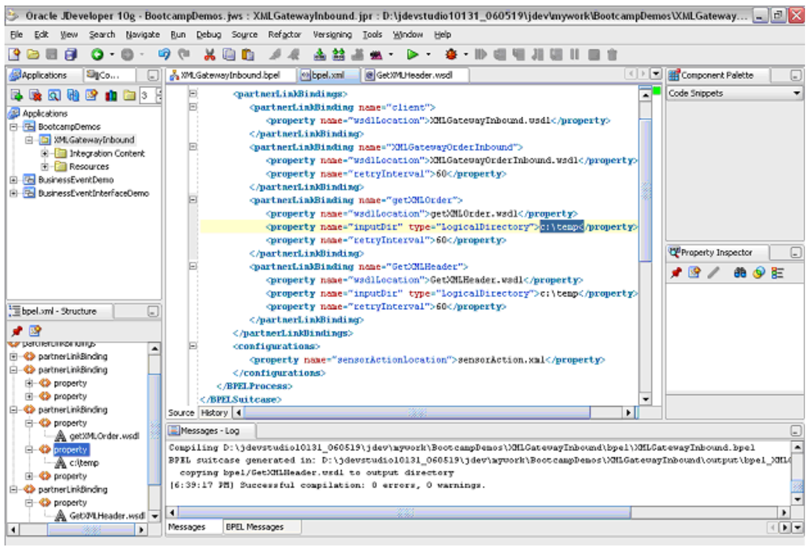
Deploying the BPEL Process

You need to deploy the BPEL process before you can run it. The BPEL process is first compiled and then deployed to the BPEL server.

Note: Before deploying the BPEL Process for XML Gateway Inbound Interface, you should:

- Load the `ecx_header` and `order_data_xmlg.xml` into `C:\Temp` folder.

- Go to the `bpel.xml` file and point the logical directory to `C:\Temp` .



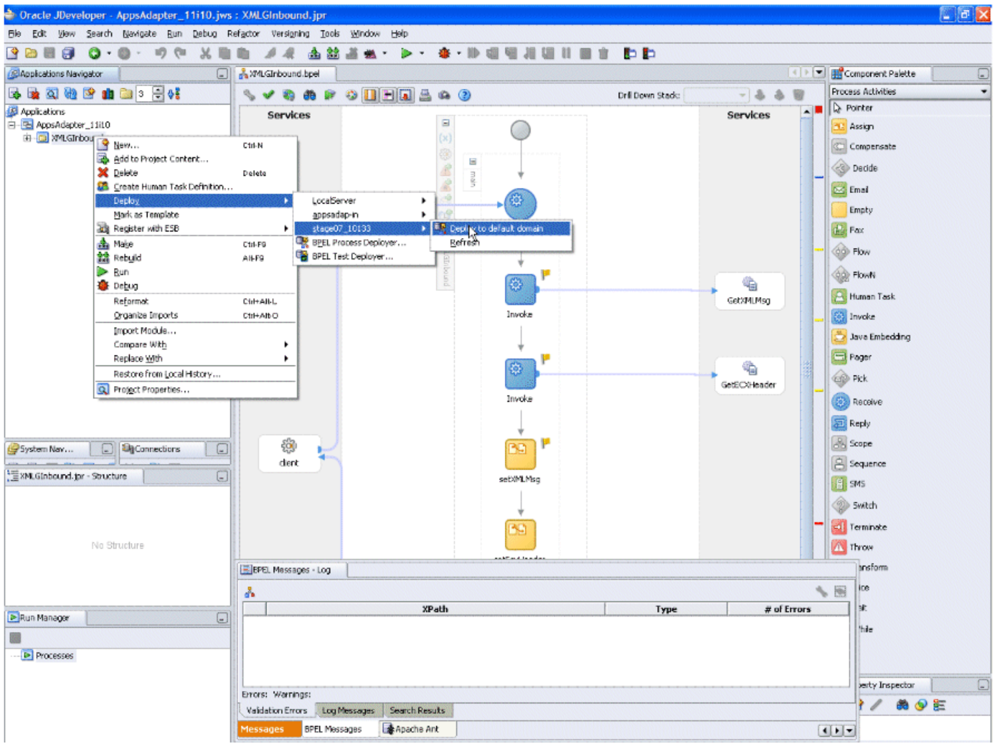
```
      <partnerLinkBindings>
         <partnerLinkBinding name="client">
            <property name="wsdlLocation">XMLGatewayInbound.wsdl</property>
         </partnerLinkBinding>
         <partnerLinkBinding name="XMLGatewayOrderInbound">
            <property name="wsdlLocation">XMLGatewayOrderInbound.wsdl</property>
            <property name="retryInterval">60</property>
         </partnerLinkBinding>
         <partnerLinkBinding name="getXMLOrder">
            <property name="wsdlLocation">getXMLOrder.wsdl</property>
            <property name="inputDir" type="logicalDirectory">c:\temp</property>
            <property name="retryInterval">60</property>
         </partnerLinkBinding>
         <partnerLinkBinding name="GetXMLHeader">
            <property name="wsdlLocation">GetXMLHeader.wsdl</property>
            <property name="inputDir" type="logicalDirectory">c:\temp</property>
            <property name="retryInterval">60</property>
         </partnerLinkBinding>
      </partnerLinkBindings>
      <configurations>
         <property name="sensorActionlocation">sensorAction.xml</property>
      </configurations>
   </BPELProcess>
</BPELSuitcase>
```

```
Compiling D:\jdevstudio10131_060519\jdev\mywork\BootcampDemos\XMLGatewayInbound\bpel\XMLGatewayInbound.bpel
BPEL suitcase generated in: D:\jdevstudio10131_060519\jdev\mywork\BootcampDemos\XMLGatewayInbound\output\bpel_XML
  copying bpel/GetXMLHeader.wsdl to output directory
[6:39:17 PM] Successful compilation: 0 errors, 0 warnings.
```

To deploy the BPEL process:

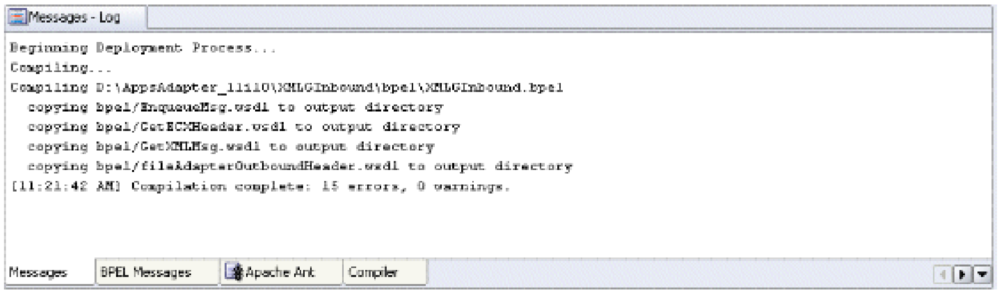
1. Select the BPEL project in the Applications window.
2. Right-click the project and select **Deploy > [Server Connection] > Deploy to Default Domain** from the menu.

Deploying the BPEL Process



3. The BPEL process is compiled and deployed. You can check the progress in the Messages window.

Messages Window



```
Beginning Deployment Process...
Compiling...
Compiling D:\AppsAdapter_11i10\XMLGInbound\bpel\XMLGInbound.bpel
   copying bpel/EnqueueMsg.wsdl to output directory
   copying bpel/GetECXHeader.wsdl to output directory
   copying bpel/GetXMLMsg.wsdl to output directory
   copying bpel/fileAdapterOutboundHeader.wsdl to output directory
[11:21:42 AM] Compilation complete: 15 errors, 0 warnings.
```

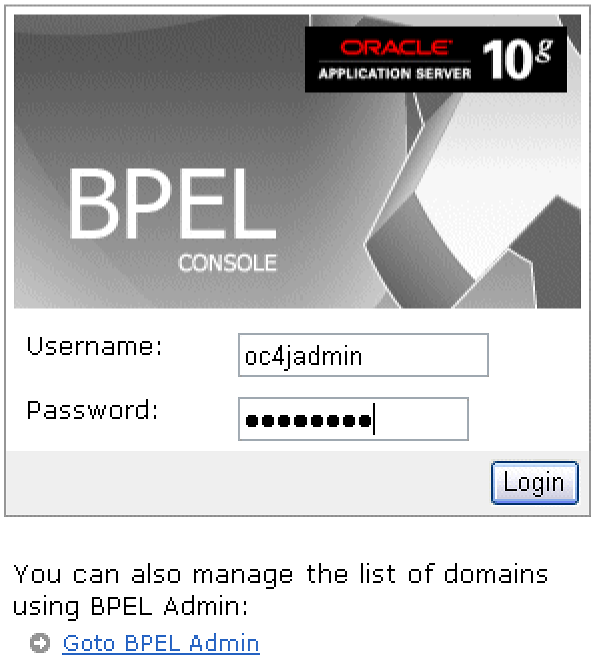
Testing the BPEL Process

Once the BPEL process is deployed, it can be seen in the BPEL Console. You can manage and monitor the process from the BPEL Console. You can also test the process and the integration interface by manually initiating the process.

To test the BPEL process:

1. Navigate to Oracle Application Server 10*g* BPEL Console (`http://<soaSuiteServerHostName>:<port>/BPELConsole`).

   The BPEL Console login page appears.



Username: oc4jadmin

Password: ●●●●●●●●●

You can also manage the list of domains using BPEL Admin:

2. Enter the username and password and click **Login**.

3. The Oracle Enterprise Manager 10*g* BPEL Control appears. The list of deployed processes is shown under Deployed BPEL Processes.

4. Click the BPEL process that you want to initiate. The Initiate page appears. Enter the input values required by the process. You can also specify an XML file for the File Adapter to pick.

5. Click **Post XML Message** to initiate the process.

6. The BPEL process is now initiated. You can check the process flow by clicking the **Visual Flow** icon.

7. The audit trail provides information on the steps that have been executed. The audit trail also records the Reference ID that is returned for the transaction. You can check the audit trail by clicking the **Audit Instance** icon.

   If the BPEL process runs into an error, then a corresponding error code is returned. The following table lists the common error codes, their descriptions, and the fix that you can use for them.

Error Codes for the BPEL Process

Code No.	Code Name	Description	Fix
12400	`APPS_MESSAGE_STANDARD_NOT_FOUND`	Message Standard not set in the header	Set the Message Standard value in the header
12401	`APPS_TRANSACTION_TYPE_NOT_FOUND`	Transaction Type not set in the header	Set the Transaction Type value in the header
12402	`APPS_TRANSACTION_SUBTYPE_NOT_FOUND`	Transaction Subtype not set in the header	Set the Transaction Subtype value in the header
12403	`APPS_PARTY_SITE_ID_NOT_FOUND`	Party Site Id not set in the header	Set the Party Site Id value in the header
12404	`APPS_MESSAGE_TYPE_NOT_FOUND`	Message Type not set in the header	Set the Message Type value in the header
12406	`APPS_CONTEXT_ERROR`	Error in setting Apps Context	Check the username and responsibility values
12407	`APPS_AUTHENTICATION_ERROR`	Invalid FND username/password	Check the username and password values
12408	`APPS_UNKNOWN_EX`	Unknown error in Apps Interaction	Check if all the header values are valid
12409	`APPS_XMLG_HEADER_NULL`	XML Gateway header is null	Pass the required parameters in header

Verifying Records in Oracle Applications

Once the BPEL process is successfully initiated and completed, you can validate it through the relevant module in Oracle Applications. For example, you can check for the creation of a purchase order in Oracle Order Management.

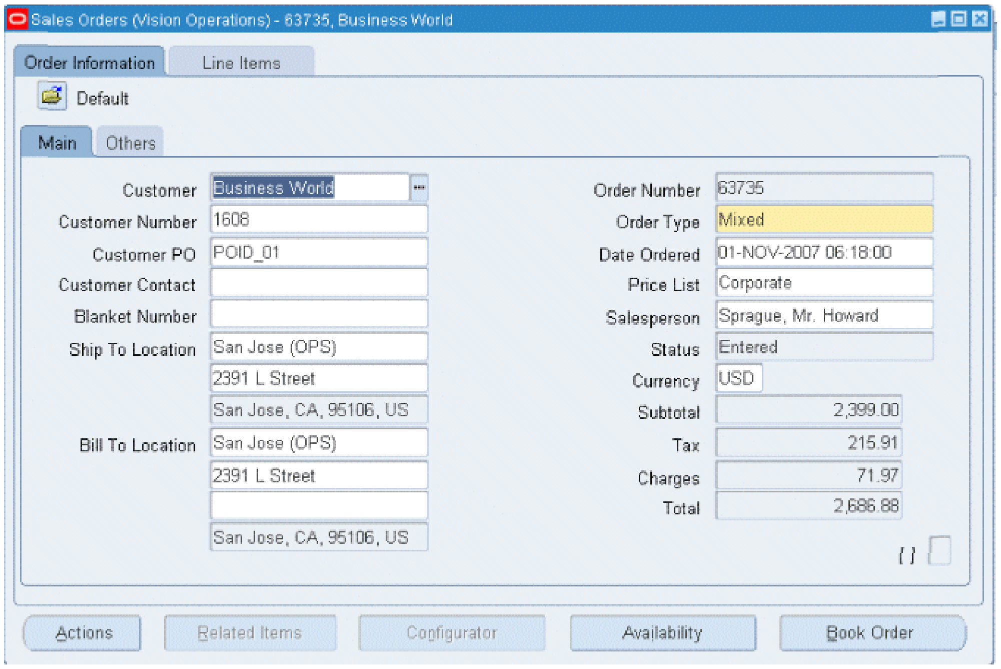
To validate it in Oracle Order Management:

1. Log in to the Forms-based Oracle Applications with the Order Management, Super User responsibility.

2. Select Order Returns > Sales Order. The Sales Order Forms opens up.

3. Search for an order by entering the order number in the Customer PO field. This brings up the details of a newly created order.

   Sales Orders

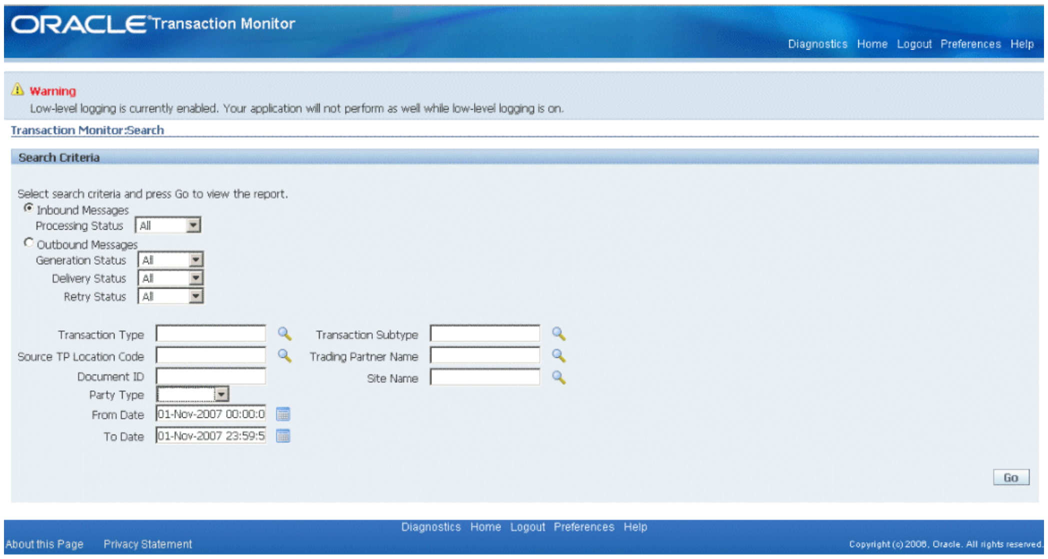
Customer	Business World	Order Number	63735
Customer Number	1608	Order Type	Mixed
Customer PO	POID_01	Date Ordered	01-NOV-2007 06:18:00
Customer Contact		Price List	Corporate
Blanket Number		Salesperson	Sprague, Mr. Howard
Ship To Location	San Jose (OPS)	Status	Entered
	2391 L Street	Currency	USD
	San Jose, CA, 95106, US	Subtotal	2,399.00
Bill To Location	San Jose (OPS)	Tax	215.91
	2391 L Street	Charges	71.97
		Total	2,686.88
	San Jose, CA, 95106, US		

   You can also select the Items tab for the item details.

Additionally, you can validate it from the Transaction Monitor. The Transaction Monitor is a tool for monitoring the status of inbound and outbound transactions originating from and going into Oracle Applications that have been processed by the XML Gateway and delivered or received by the Oracle Transport Agent. It shows a complete history and audit trail of these documents.

To validate it using Oracle Transaction Monitor, you need to log in to Oracle Applications with the Workflow Administrator Web Applications responsibility. Select Transaction Monitor to open the search window to search for the order.

Searching from the Transaction Monitor



ORACLE Transaction Monitor

Diagnostics Home Logout Preferences Help

Warning
Low-level logging is currently enabled. Your application will not perform as well while low-level logging is on.

Transaction Monitor:Search

Search Criteria

Select search criteria and press Go to view the report.

● Inbound Messages	
Processing Status	All
○ Outbound Messages	
Generation Status	All
Delivery Status	All
Retry Status	All
Transaction Type	
Source TP Location Code	
Document ID	
Party Type	
From Date	01-Nov-2007 00:00:0
To Date	01-Nov-2007 23:59:5
Transaction Subtype	
Trading Partner Name	
Site Name	

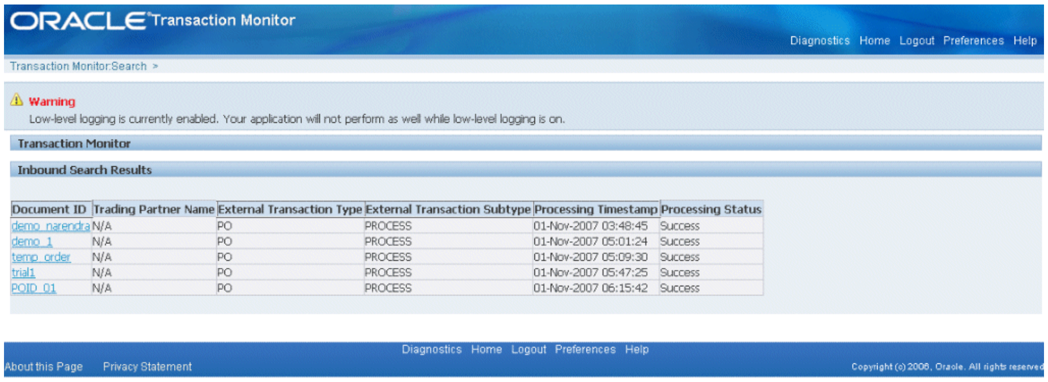
Go

Diagnostics Home Logout Preferences Help

About this Page Privacy Statement Copyright (c) 2006, Oracle. All rights reserved.

Click **Go** to retrieve all XML inbound messages listed in the Inbound Search Results region.

Listing All XML Gateway Inbound Documents



ORACLE Transaction Monitor

Diagnostics Home Logout Preferences Help

Transaction Monitor:Search >

Warning
Low-level logging is currently enabled. Your application will not perform as well while low-level logging is on.

Transaction Monitor

Inbound Search Results

Document ID	Trading Partner Name	External Transaction Type	External Transaction Subtype	Processing Timestamp	Processing Status
demo_narendra	N/A	PO	PROCESS	01-Nov-2007 03:48:45	Success
demo_1	N/A	PO	PROCESS	01-Nov-2007 05:01:24	Success
temp_order	N/A	PO	PROCESS	01-Nov-2007 05:09:30	Success
trial1	N/A	PO	PROCESS	01-Nov-2007 05:47:25	Success
POID_01	N/A	PO	PROCESS	01-Nov-2007 06:15:42	Success

Diagnostics Home Logout Preferences Help

About this Page Privacy Statement Copyright (c) 2006, Oracle. All rights reserved.

See *Oracle XML Gateway User's Guide* for details on using the Transaction Monitor.

Design-Time Task for XML Gateway Outbound Messaging

This section discusses the design-time steps, for an XML Gateway outbound message, that are different from the design-time steps for an inbound message.

Prerequisites to Configure XML Gateway Outbound

Setting Up Correlation Identifier

For invoking an outbound XML Gateway message map, you need to set up the correlation identifier in Oracle Applications. The correlation identifier enables you to label messages meant for a specific agent, in case there are multiple agents listening on the outbound queue. The agent listening for a particular correlation picks up the messages that match the correlation identifier for the agent.

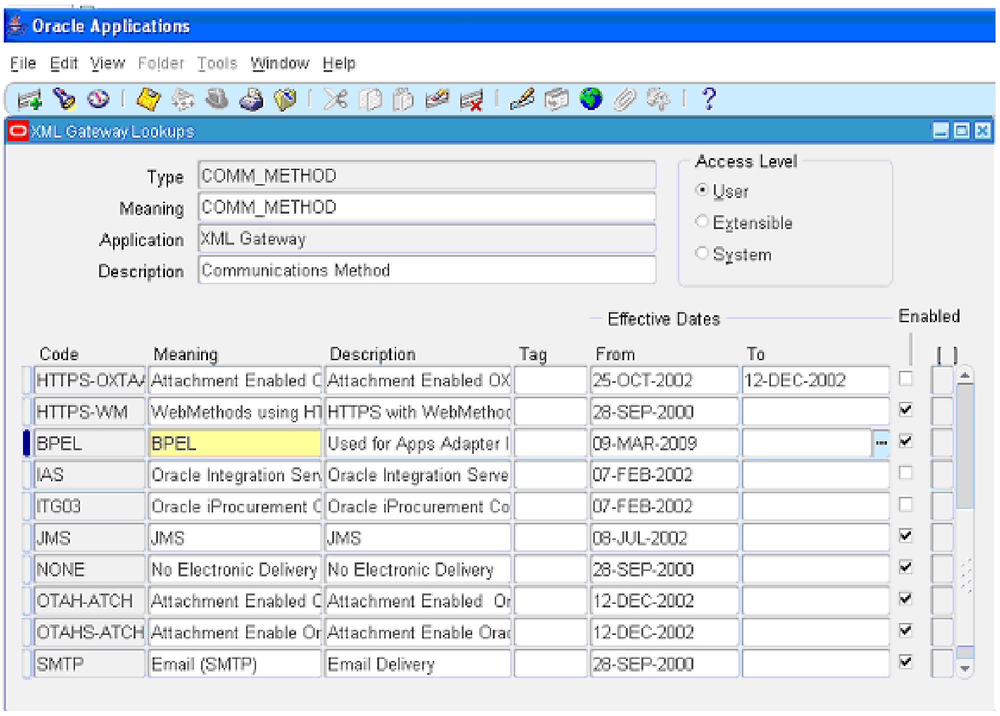
To set up the correlation identifier:

1. Log in to Oracle Applications with the XML Gateway responsibility. The Navigator page appears.

2. Click the **XML Gateway** link.

3. Click the **Define Lookup Values** link under XML Gateway.

XML Gateway Lookups



Type	COMM_METHOD
Meaning	COMM_METHOD
Application	XML Gateway
Description	Communications Method

Access Level: [✓ User] Extensible System

				Effective Dates		Enabled
Code	Meaning	Description	Tag	From	To	
HTTPS-OXTA/	Attachment Enabled C	Attachment Enabled OX		25-OCT-2002	12-DEC-2002	☐
HTTPS-WM	WebMethods using HT	HTTPS with WebMethoc		28-SEP-2000		✓
BPEL	BPEL	Used for Apps Adapter I		09-MAR-2009		✓
IAS	Oracle Integration Sen	Oracle Integration Serve		07-FEB-2002		☐
ITG03	Oracle iProcurement C	Oracle iProcurement Co		07-FEB-2002		☐
JMS	JMS	JMS		08-JUL-2002		✓
NONE	No Electronic Delivery	No Electronic Delivery		28-SEP-2000		✓
OTAH-ATCH	Attachment Enabled C	Attachment Enabled Or		12-DEC-2002		✓
OTAHS-ATCH	Attachment Enable Or	Attachment Enable Orac		12-DEC-2002		✓
SMTP	Email (SMTP)	Email Delivery		28-SEP-2000		✓

4. Enter `COMM_METHOD` for the **Type** field.

5. Enter `BPEL` for the **Code** field.

   Oracle XML Gateway puts the correlation of BPEL when enqueueing the message on the ECX_OUTBOUND queue.

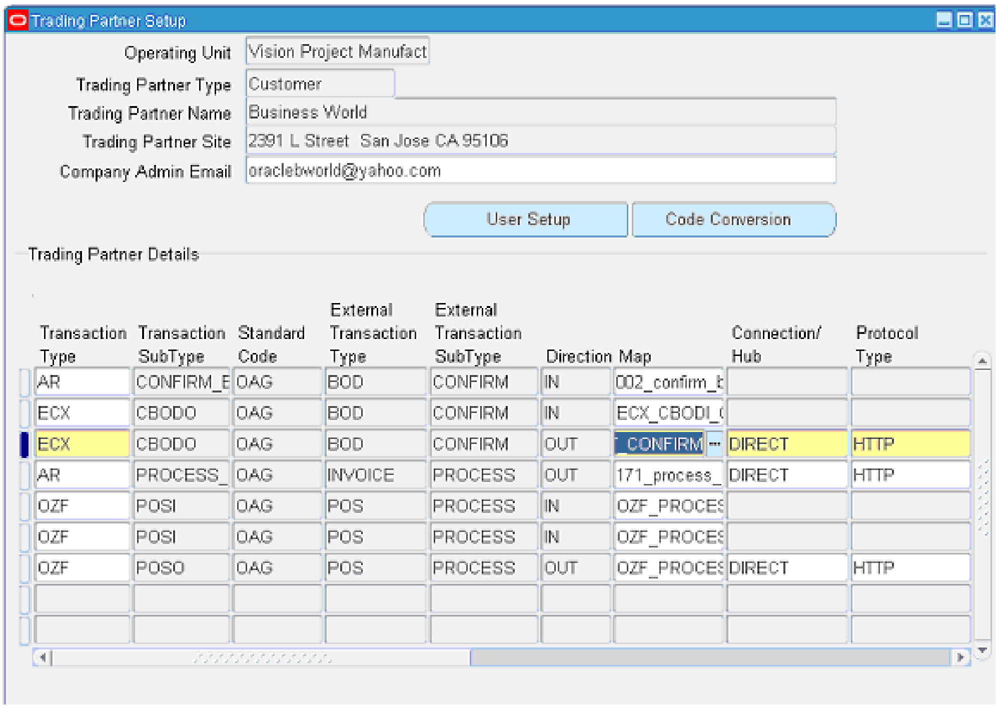
Setting Up XML Gateway Trading Partner

Once you have the correlation identifier set up correctly, you also need to ensure a valid outbound XML Gateway trading partner in the Trading Partner Setup form through XML Gateway responsibility. You want to have the Protocol Type field set to `BPEL`.

For example, a Trading Partner (such as 'Business World' with Site information '2391 L Street San Jose CA 95106' and partner type 'Customer') has the following details for an outbound transaction:

- Transaction type: ECX
- Transaction sub type: CBODO
- Standard Code: OAG
- External transaction type: BOD
- External transaction subtype: CONFIRM
- Direction: OUT
- Map: ECX_CBODO_OAG72_OUT_CONFIRM
- Connection/Hub: DIRECT
- Protocol Type: BPEL
- Source Trading partner location code: BWSANJOSE

Trading Partner Setup



Operating Unit	Vision Project Manufact
Trading Partner Type	Customer
Trading Partner Name	Business World
Trading Partner Site	2391 L Street San Jose CA 95106
Company Admin Email	oraclebworld@yahoo.com

Transaction Type	Transaction SubType	Standard Code	External Transaction Type	External Transaction SubType	Direction	Map	Connection/ Hub	Protocol Type
AR	CONFIRM_E	OAG	BOD	CONFIRM	IN	002_confirm_t		
ECX	CBODO	OAG	BOD	CONFIRM	IN	ECX_CBODI_(		
ECX	CBODO	OAG	BOD	CONFIRM	OUT	_CONFIRM	DIRECT	HTTP
AR	PROCESS_	OAG	INVOICE	PROCESS	OUT	171_process_	DIRECT	HTTP
OZF	POSI	OAG	POS	PROCESS	IN	OZF_PROCES		
OZF	POSI	OAG	POS	PROCESS	IN	OZF_PROCES		
OZF	POSO	OAG	POS	PROCESS	OUT	OZF_PROCES	DIRECT	HTTP

BPEL Process Creation Flow

The following procedure is required to accomplish the design-time task.

1. Create a new BPEL project, page 4-49
2. Create a Partner Link, page 4-51
3. Add a Receive activity, page 4-56
4. Add a Partner Link for File Adapter, page 4-58
5. Add an Invoke activity, page 4-64
6. Add an Assign activity, page 4-66

Creating a New BPEL Project

To create a new BPEL project:

1. Open JDeveloper BPEL Designer.
2. From the **File** menu, select **New**.

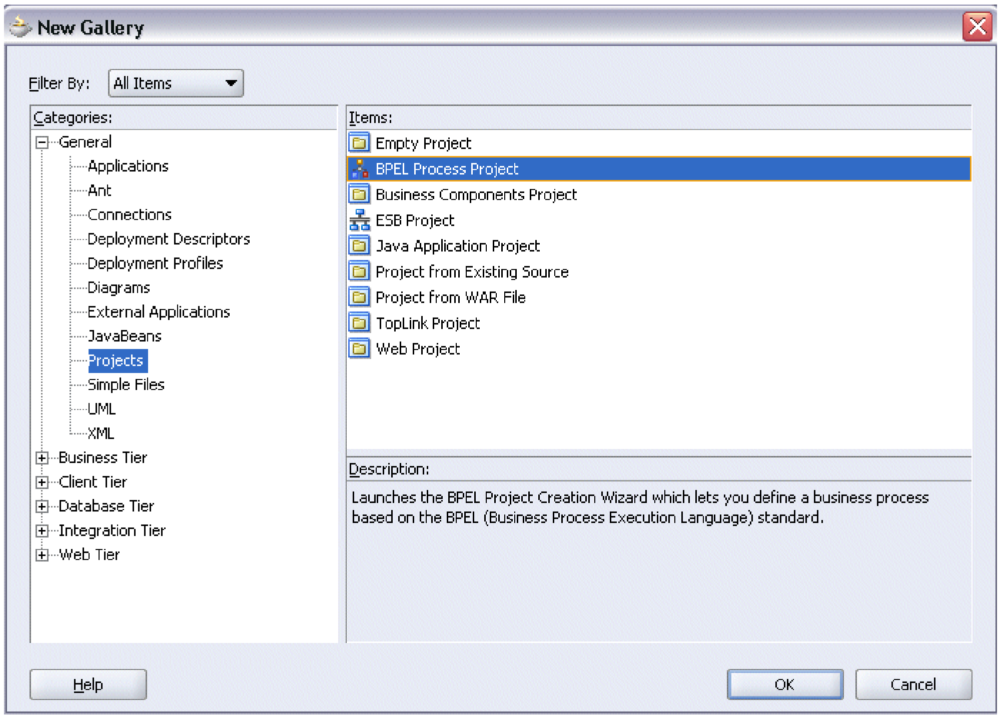
The New Gallery dialog box appears.

3. Select **All Items** from the **Filter By** box. This displays a list of available categories.

4. Expand the **General** node, and then select **Projects**.

5. Select **BPEL Process Project** from the **Items** group.

Creating a New BPEL Process Project



Launches the BPEL Project Creation Wizard which lets you define a business process based on the BPEL (Business Process Execution Language) standard.

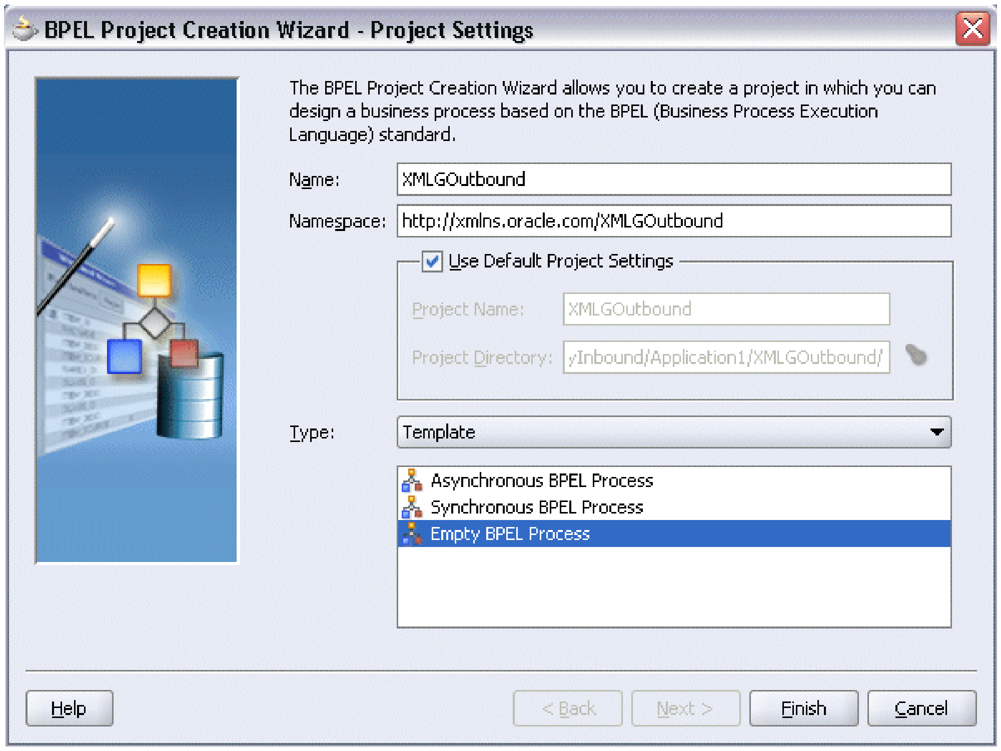
6. Click **OK**. The BPEL Project Creation Wizard - Project Settings dialog box appears.

7. In the **Name** field, enter a descriptive name, such as `XMLGOutbound`.

   Keep the default selection **Use Default Project Settings** unchanged.

8. Keep the default selection **Template** as the Type field. Select **Empty BPEL Process** as the BPEL process type.

Specifying New BPEL Project Settings


BPEL Project Creation Wizard - Project Settings

The BPEL Project Creation Wizard allows you to create a project in which you can design a business process based on the BPEL (Business Process Execution Language) standard.

Name: XMLGOutbound

Namespace: http://xmlns.oracle.com/XMLGOutbound

[✓] Use Default Project Settings

Project Name: XMLGOutbound

Project Directory: y\Inbound\Application1\XMLGOutbound\

Type: Template

- Asynchronous BPEL Process
- Synchronous BPEL Process
- Empty BPEL Process (selected)

9. Click **Finish**.

   An empty BPEL process is created.

   The required source files including `bpel.xml`, `XMLGOutbound.bpel`, and `XMLGOutbound.wsdl` are also generated.

Adding a Partner Link

You need to add a partner link for the outbound XML message in order for the Receive activity to dequeue it later.

To add a partner link:

1. Click **Services** in the Component palette.

   Drag and drop **Oracle Applications** from the BPEL Services list into the right part of the process diagram. The Adapter Configuration Wizard Welcome page appears. Click **Next**.

2. Enter a service name in the **Service Name** field. For example, `GetAck`.

   Click **Next**. The Service Connection dialog appears.

3. You can perform either one of the following options for your database connection:

   > **Note:** You need to connect to the database where Oracle E-Business Suite is running.

   - You can create a new database connection by clicking the **New** icon.

     Detailed instructions on how to create a new database connection, see Creating a New Database Connection, page 4-12.

   - You can select an existing database connection that you have configured earlier from the **Connection** drop-down list.

     The selected database connection information appears in the Service Connection dialog box. The JNDI (Java Naming and Directory Interface) name corresponding to the selected database connection also appears automatically in the **Database Server JNDI Name** field. Alternatively, you can specify a JNDI name.

     > **Note:** When you specify a JNDI name, the deployment descriptor of the Adapter for Oracle Applications must associate this JNDI name with configuration properties required by the adapter to access the database.

     The JNDI name acts as a placeholder for the connection used when your service is deployed to the BPEL server. This enables you to use different databases for development and later for production.

4. Once you have created a new connection or selected an existing connection, you can add an outbound message map by browsing through the list of message maps available in Oracle E-Business Suite.

   Click **Next**.

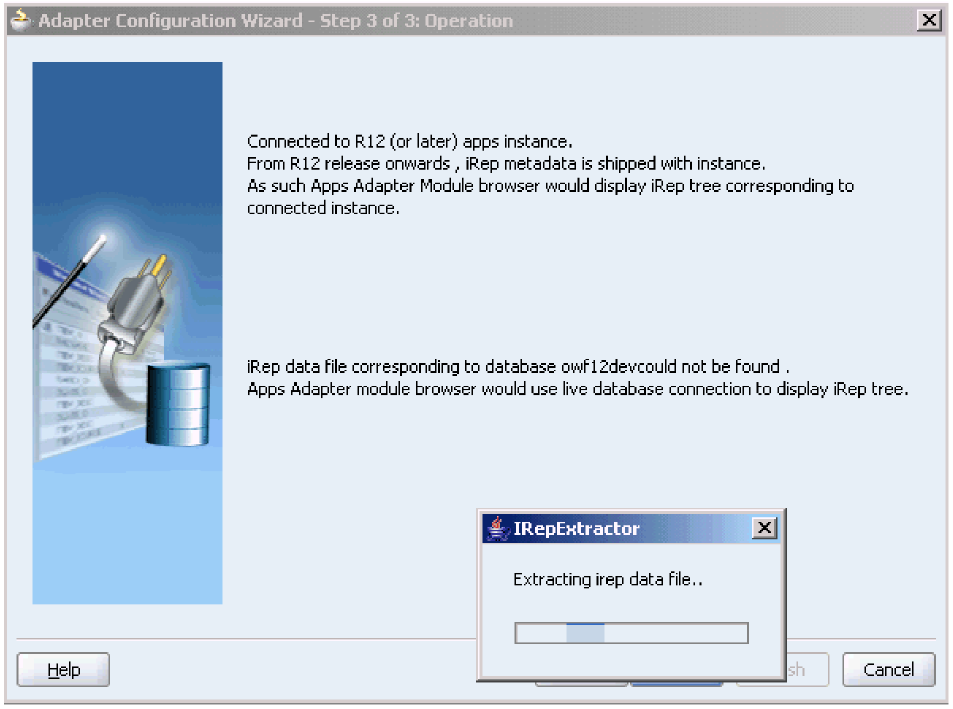
   For Oracle E-Business Suite Release 12:

   If you are connecting to Oracle E-Business Suite Release 12, then the **IREP File not present** dialog appears indicating that OracleAS Adapter for Oracle Applications could not find the Oracle Integration Repository data file corresponding to the database you are connecting in your workspace. Absence of the data file would make browsing or searching of Integration Repository tree considerably slow. You can choose to extract the data file and create a local copy of the Integration Repository data file. Once it is created successfully, OracleAS Adapter for Oracle Applications will pick it up automatically next time and retrieve data from your local Integration Repository.

   You can select one of the following options:

- Click **Yes** to extract the Integration Repository data file.

Extracting Integration Repository Data File



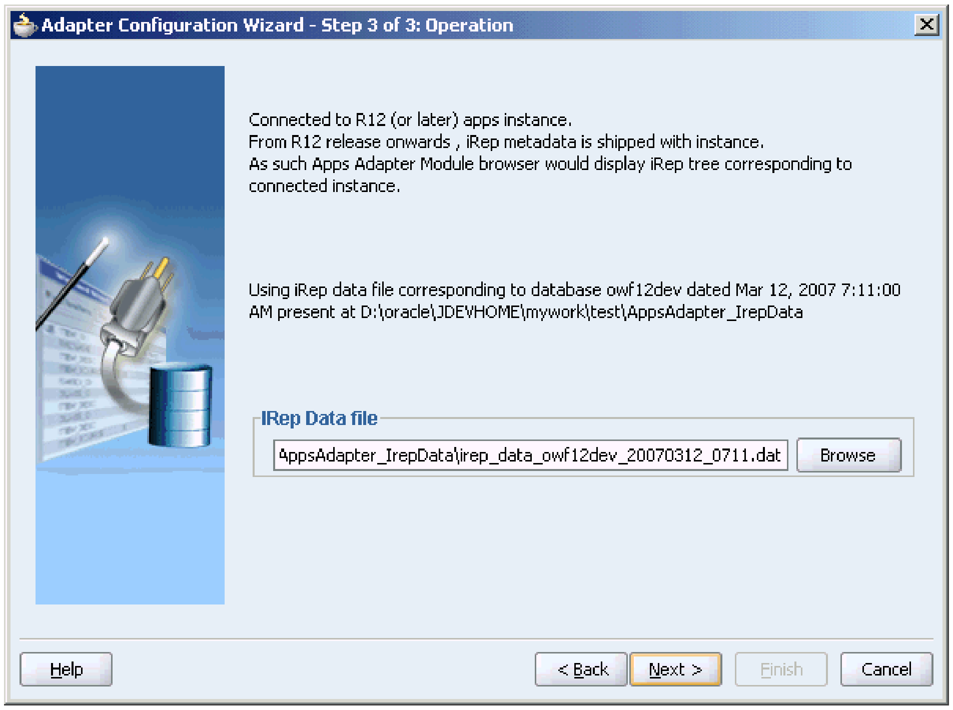
Connected to R12 (or later) apps instance.
From R12 release onwards , iRep metadata is shipped with instance.
As such Apps Adapter Module browser would display iRep tree corresponding to connected instance.

iRep data file corresponding to database owf12devcould not be found .
Apps Adapter module browser would use live database connection to display iRep tree.

Extracting irep data file..

After the system successfully creates a local copy of the Integration Repository data file, next time when you connect to the database, you will find the **IRep Data File** field appears in the Operation dialog indicating where your local copy exists with the creation date and time as part of the file name.

Using the Local Integration Repository Data File



Connected to R12 (or later) apps instance.
From R12 release onwards , iRep metadata is shipped with instance.
As such Apps Adapter Module browser would display iRep tree corresponding to connected instance.

Using iRep data file corresponding to database owf12dev dated Mar 12, 2007 7:11:00 AM present at D:\oracle\JDEVHOME\mywork\test\AppsAdapter_IrepData

AppsAdapter_IrepData\irep_data_owf12dev_20070312_0711.dat

- Click **No** to query the Integration Repository data file from the live database you are connecting to display the Integration Repository tree.

  Note: It is highly recommended that you create a local copy of the Integration Repository data file so that Adapter for Oracle Applications will query the data next time from the local copy in your workspace to enhance the performance.

Click **Next** in the Operation page to open the Oracle Applications Module Browser.

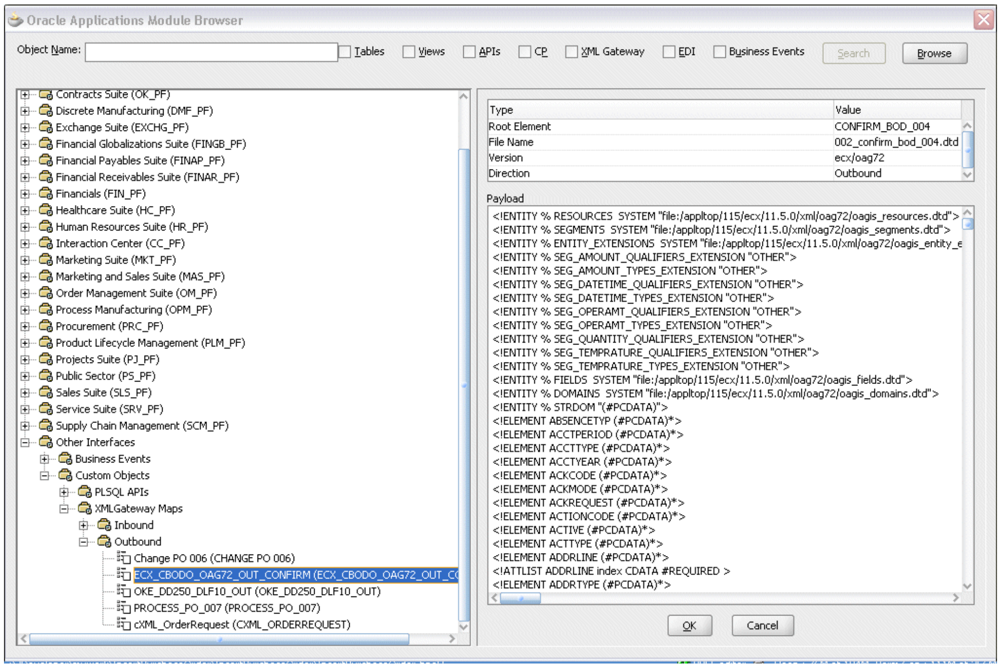
For Oracle E-Business Suite pre-Release 11.5.10:

If you are connecting to a pre-11.5.10 Oracle E-Business Suite instance, you must select the interface type in the Adapter Configuration Wizard. Select **XML Gateway** to proceed.

Click **Add** in the Application Interface dialog to open the Oracle Applications Module Browser.

5. The Oracle Applications Module Browser combines interface data from Oracle Integration Repository with information about the additional interfaces supported by Adapter for Oracle Applications, organized in a tree hierarchy.

Specify an API from The Oracle Applications Module Browser



Oracle Applications Module Browser

Object Name: ☐ Tables ☐ Views ☐ APIs ☐ CP ☐ XML Gateway ☐ EDI ☐ Business Events Search Browse

- Contracts Suite (OK_PF)
- Discrete Manufacturing (DMF_PF)
- Exchange Suite (EXCHG_PF)
- Financial Globalizations Suite (FINGB_PF)
- Financial Payables Suite (FINAP_PF)
- Financial Receivables Suite (FINAR_PF)
- Financials (FIN_PF)
- Healthcare Suite (HC_PF)
- Human Resources Suite (HR_PF)
- Interaction Center (CC_PF)
- Marketing Suite (MKT_PF)
- Marketing and Sales Suite (MAS_PF)
- Order Management Suite (OM_PF)
- Process Manufacturing (OPM_PF)
- Procurement (PRC_PF)
- Product Lifecycle Management (PLM_PF)
- Projects Suite (PJ_PF)
- Public Sector (PS_PF)
- Sales Suite (SLS_PF)
- Service Suite (SRV_PF)
- Supply Chain Management (SCM_PF)
- Other Interfaces
  - Business Events
  - Custom Objects
    - PLSQL APIs
    - XMLGateway Maps
      - Inbound
      - Outbound
        - Change PO 006 (CHANGE PO 006)
        - ECX_CBODO_OAG72_OUT_CONFIRM (ECX_CBODO_OAG72_OUT_C
        - OKE_DD250_DLF10_OUT (OKE_DD250_DLF10_OUT)
        - PROCESS_PO_007 (PROCESS_PO_007)
        - cXML_OrderRequest (CXML_ORDERREQUEST)

Type	Value
Root Element	CONFIRM_BOD_004
File Name	002_confirm_bod_004.dtd
Version	ecx/oag72
Direction	Outbound

Payload

```
<!ENTITY % RESOURCES  SYSTEM "file:/appltop/115/ecx/11.5.0/xml/oag72/oagis_resources.dtd">
<!ENTITY % SEGMENTS  SYSTEM "file:/appltop/115/ecx/11.5.0/xml/oag72/oagis_segments.dtd">
<!ENTITY % ENTITY_EXTENSIONS  SYSTEM "file:/appltop/115/ecx/11.5.0/xml/oag72/oagis_entity_e
<!ENTITY % SEG_AMOUNT_QUALIFIERS_EXTENSION "OTHER">
<!ENTITY % SEG_AMOUNT_TYPES_EXTENSION "OTHER">
<!ENTITY % SEG_DATETIME_QUALIFIERS_EXTENSION "OTHER">
<!ENTITY % SEG_DATETIME_TYPES_EXTENSION "OTHER">
<!ENTITY % SEG_OPERAMT_QUALIFIERS_EXTENSION "OTHER">
<!ENTITY % SEG_OPERAMT_TYPES_EXTENSION "OTHER">
<!ENTITY % SEG_QUANTITY_QUALIFIERS_EXTENSION "OTHER">
<!ENTITY % SEG_TEMPRATURE_QUALIFIERS_EXTENSION "OTHER">
<!ENTITY % SEG_TEMPRATURE_TYPES_EXTENSION "OTHER">
<!ENTITY % FIELDS  SYSTEM "file:/appltop/115/ecx/11.5.0/xml/oag72/oagis_fields.dtd">
<!ENTITY % DOMAINS  SYSTEM "file:/appltop/115/ecx/11.5.0/xml/oag72/oagis_domains.dtd">
<!ENTITY % STRDOM "(#PCDATA)">
<!ELEMENT ABSENCETYP (#PCDATA)*>
<!ELEMENT ACCTPERIOD (#PCDATA)*>
<!ELEMENT ACCTTYPE (#PCDATA)*>
<!ELEMENT ACCTYEAR (#PCDATA)*>
<!ELEMENT ACKCODE (#PCDATA)*>
<!ELEMENT ACKMODE (#PCDATA)*>
<!ELEMENT ACKREQUEST (#PCDATA)*>
<!ELEMENT ACTIONCODE (#PCDATA)*>
<!ELEMENT ACTIVE (#PCDATA)*>
<!ELEMENT ACTTYPE (#PCDATA)*>
<!ELEMENT ADDRLINE (#PCDATA)*>
<!ATTLIST ADDRLINE index CDATA #REQUIRED >
<!ELEMENT ADDRTYPE (#PCDATA)*>
```

OK Cancel

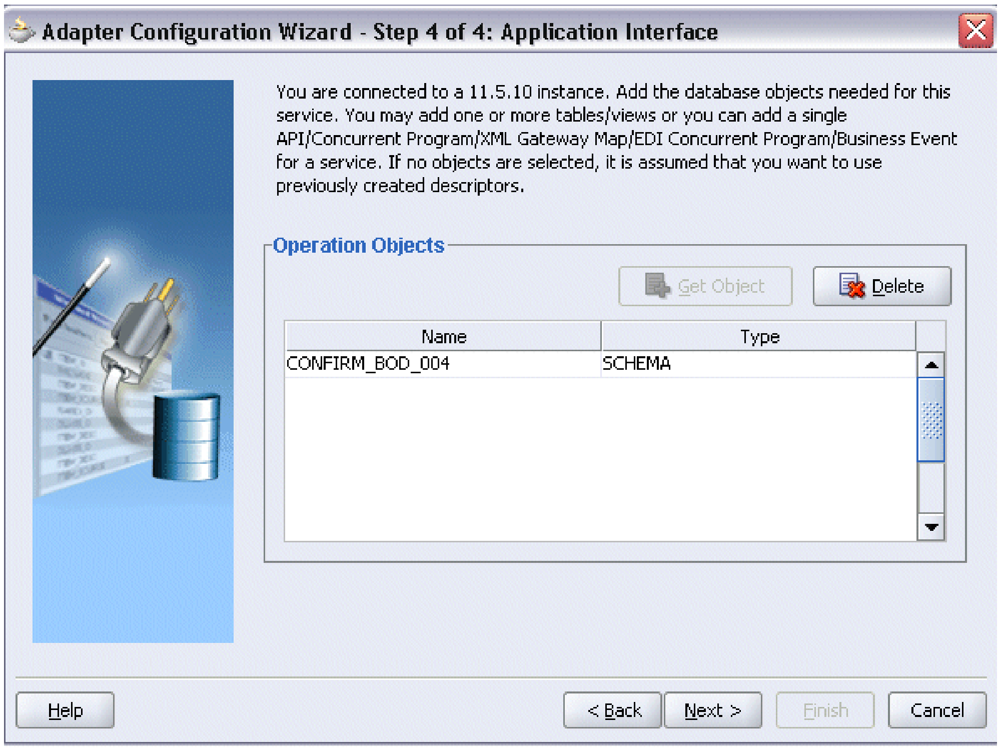
> **Note:** The Oracle Applications Module Browser includes the various product families that are available in Oracle Applications. Each product family contains the individual products. Each product contains the business entities associated with the product. Business entities contain the various application modules that are exposed for integration. These modules are grouped according to the interface they provide.

Navigate to *Other Interfaces* > *Custom Objects* >*XML Gateway Maps*> *Outbound* to select an outbound map `ECX_CBODO_OAG72_OUT_CONFIRM`.

Click **OK**.

6. The Application Interface dialog box appears with the selected XML schema.

Application Interface Dialog


Adapter Configuration Wizard - Step 4 of 4: Application Interface

You are connected to a 11.5.10 instance. Add the database objects needed for this service. You may add one or more tables/views or you can add a single API/Concurrent Program/XML Gateway Map/EDI Concurrent Program/Business Event for a service. If no objects are selected, it is assumed that you want to use previously created descriptors.

Operation Objects

Name	Type
CONFIRM_BOD_004	SCHEMA

7. Click **Next** and then click **Finish** to complete the process of configuring Adapter for Oracle Applications.

   The wizard generates the WSDL file corresponding to the XML schema. This WSDL file is now available for the partner link.

8. Click **Apply** and then **OK**. The partner link is created with the required WSDL settings.

Adding a Receive Activity

When configuring the OracleAS Adapter for Oracle Applications to use an outbound XML Gateway map, you need to configure the Receive activity for the associated partner link. The Receive activity dequeues the outbound XML messages.

To configure the Receive activity:

1. In JDeveloper BPEL Designer, drag and drop the **Receive** activity from the Component Palette into the process map.

2. Link the **Receive** activity to the partner link `GetAck` you created earlier.

The Receive dialog box appears.

3. Enter an appropriate name for the Receive activity.

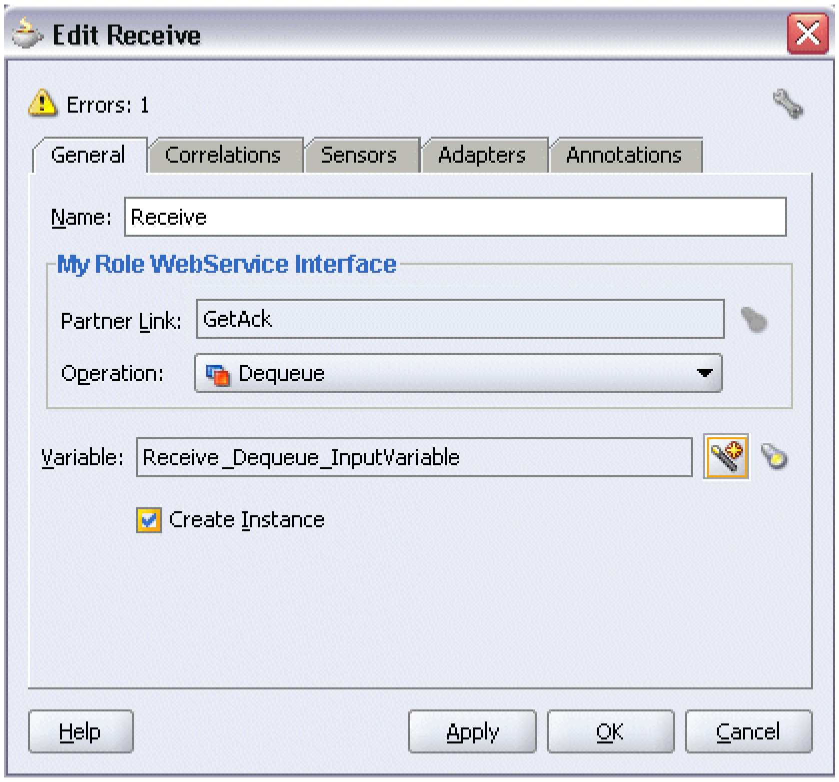
   The Dequeue **Operation** is automatically selected since the partner link has been configured with an outbound XML Gateway map.

4. Specify a **Variable** to receive the message data from the partner link by clicking the **Create** icon to the right of the Variable field. The **Create Variable** dialog box appears.

   Click **OK** to accept the default name.

5. Select the **Create Instance** check box.

Editing the Receive Activity



Edit Receive

Errors: 1

General | Correlations | Sensors | Adapters | Annotations

Name: Receive

My Role WebService Interface

Partner Link: GetAck

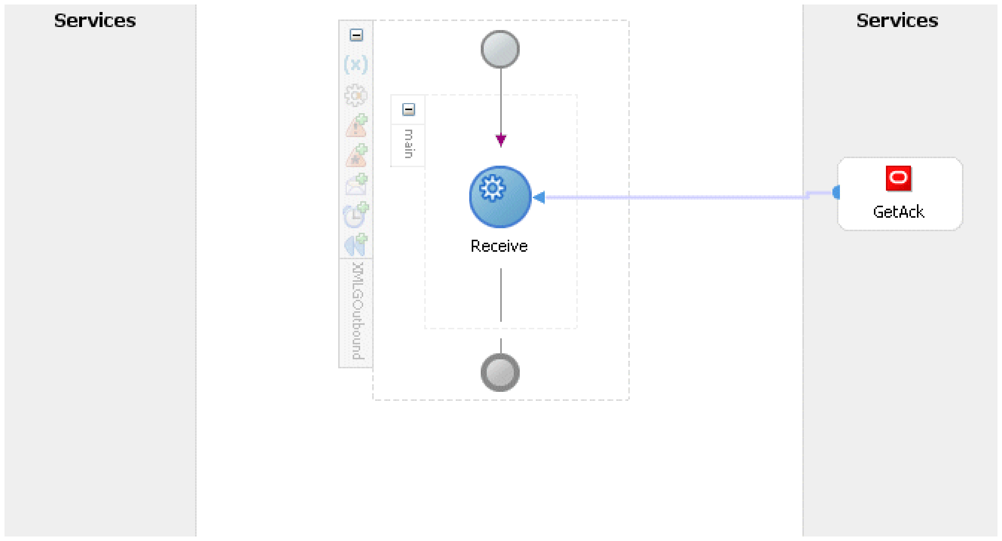
Operation: Dequeue

Variable: Receive_Dequeue_InputVariable

[✓] Create Instance

6. Click **Apply** and then **OK**.

   The Receive activity is added to the BPEL process diagram.



Note: You can define a **Header Variable** under the **Adapters** tab of the Receive dialog box. This header variable is populated with context information from the outbound XML message. Values for fields like **MESSAGE_TYPE**, **MESSAGE_STANDARD** and trading party information like **PARTY_SITE_ID** are returned through this variable.

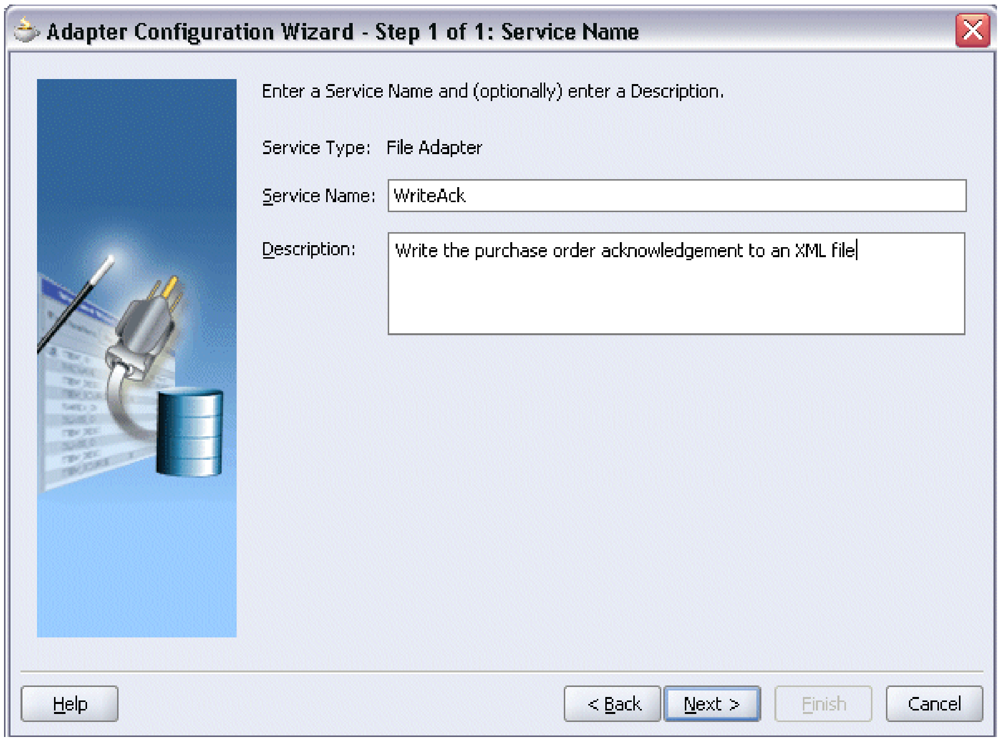
Adding a Partner Link for File Adapter

Use this step to configure a partner link by writing the purchase order acknowledgement to an XML file.

To add a Partner Link for File Adapter:

1. In JDeveloper BPEL Designer, drag and drop the **File Adapter** service from the **Service** section of the Component Palette into the Partner Link area of the process diagram. The Adapter Configuration Wizard appears.
2. The Service Name dialog box appears.

Specifying the Service Name



Adapter Configuration Wizard - Step 1 of 1: Service Name

Enter a Service Name and (optionally) enter a Description.

Service Type: File Adapter

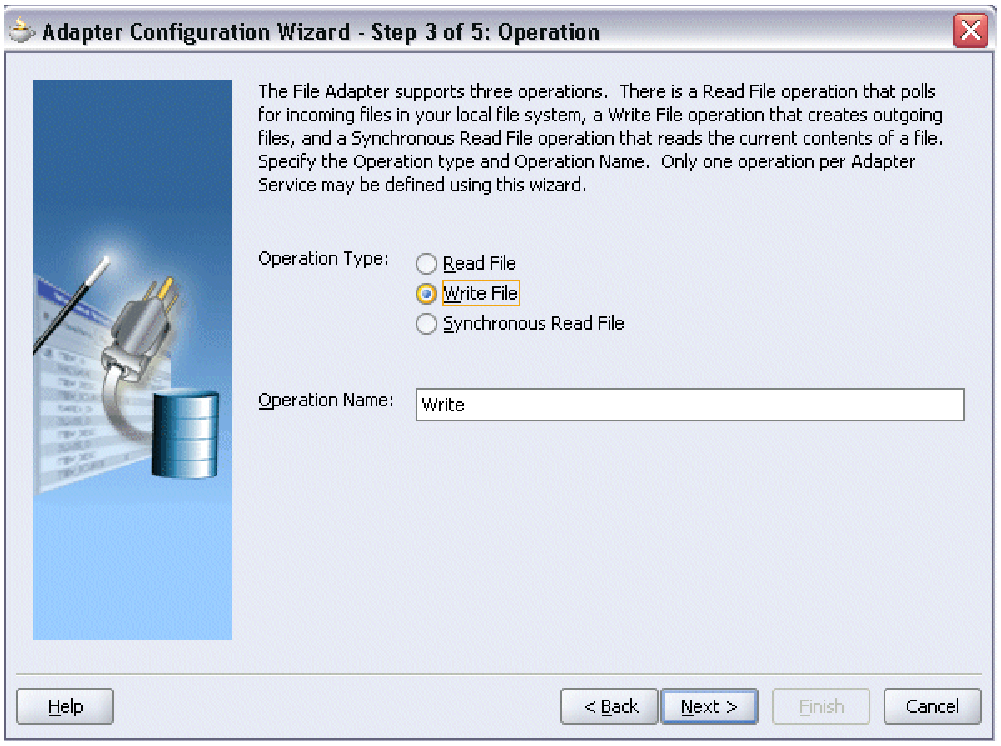
Service Name: WriteAck

Description: Write the purchase order acknowledgement to an XML file

3. Enter a name for the File Adapter service, such as `WriteAck`. You can add an optional description of the service.

4. Click **Next**. The Operation dialog box appears.

Specifying the Operation


Adapter Configuration Wizard - Step 3 of 5: Operation

The File Adapter supports three operations. There is a Read File operation that polls for incoming files in your local file system, a Write File operation that creates outgoing files, and a Synchronous Read File operation that reads the current contents of a file. Specify the Operation type and Operation Name. Only one operation per Adapter Service may be defined using this wizard.

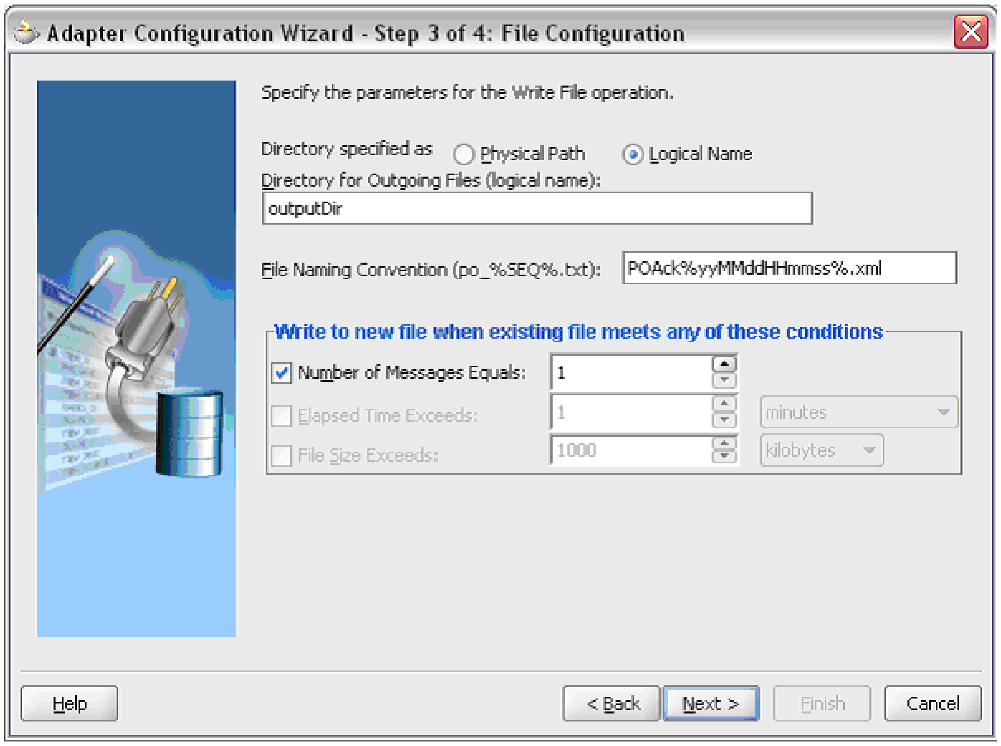
Operation Type: Read File [✓ Write File] Synchronous Read File

Operation Name: Write

5. Specify the operation type, for example **Write File**. This automatically populates the **Operation Name** field.

   Click **Next** to access the File Configuration dialog box.

Configuring the Output File


Adapter Configuration Wizard - Step 3 of 4: File Configuration

Specify the parameters for the Write File operation.

Directory specified as ○ Physical Path [✓ ● Logical Name]

Directory for Outgoing Files (logical name): outputDir

File Naming Convention (po_%SEQ%.txt): POAck%yyMMddHHmmss%.xml

Write to new file when existing file meets any of these conditions

☑ Number of Messages Equals: 1

☐ Elapsed Time Exceeds: 1 minutes

☐ File Size Exceeds: 1000 kilobytes

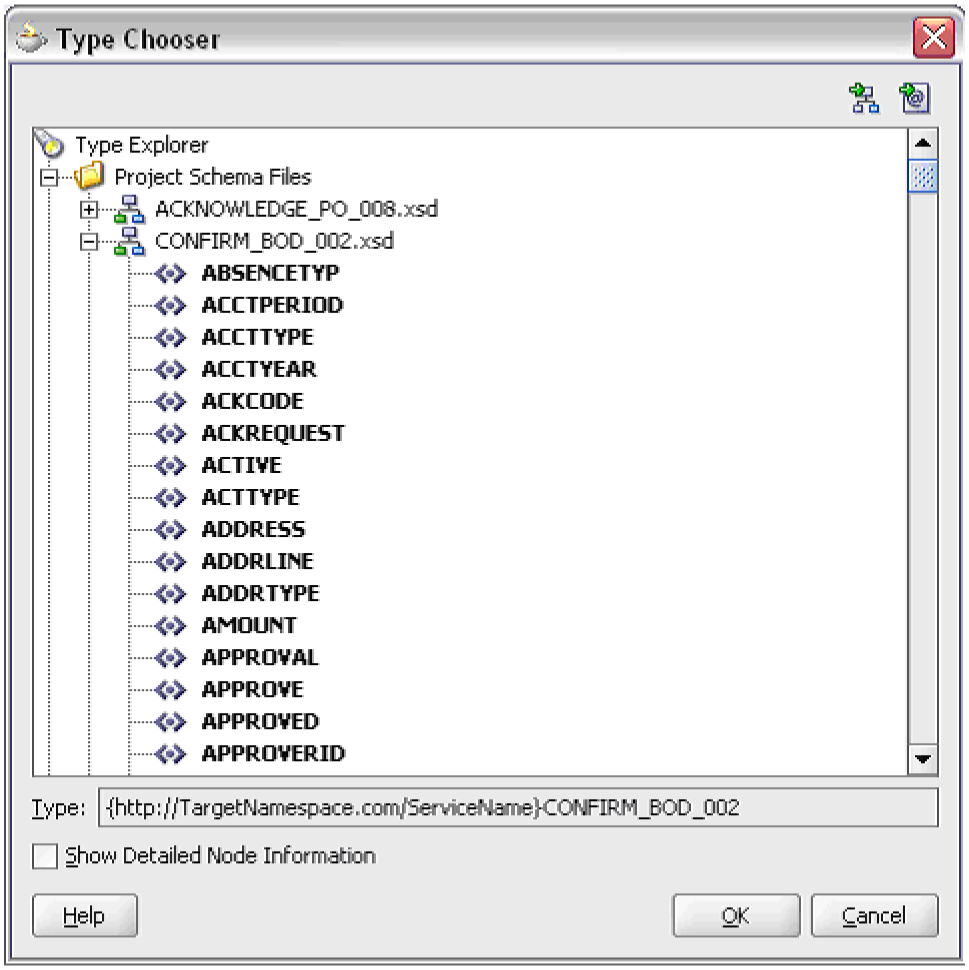
6. For the Directory specified as field, select **Logical Path**. Enter directory path in the Directory for Outgoing Files field, and specify a naming convention for the output file, for example, `POAck%yyMMddJJmmss%.xml`.

7. Confirm the default write condition: Number of Messages Equals 1. Click **Next**. The Messages dialog box appears.

8. Select the **Browse** check box to locate the schema location and schema element.

   The Type Chooser dialog box appears. Expand the path by selecting **Project Schema Files > CONFIRM_BOD_002.xsd**. Select **CONFIRM_BOD_002** as the element.

Type Chooser

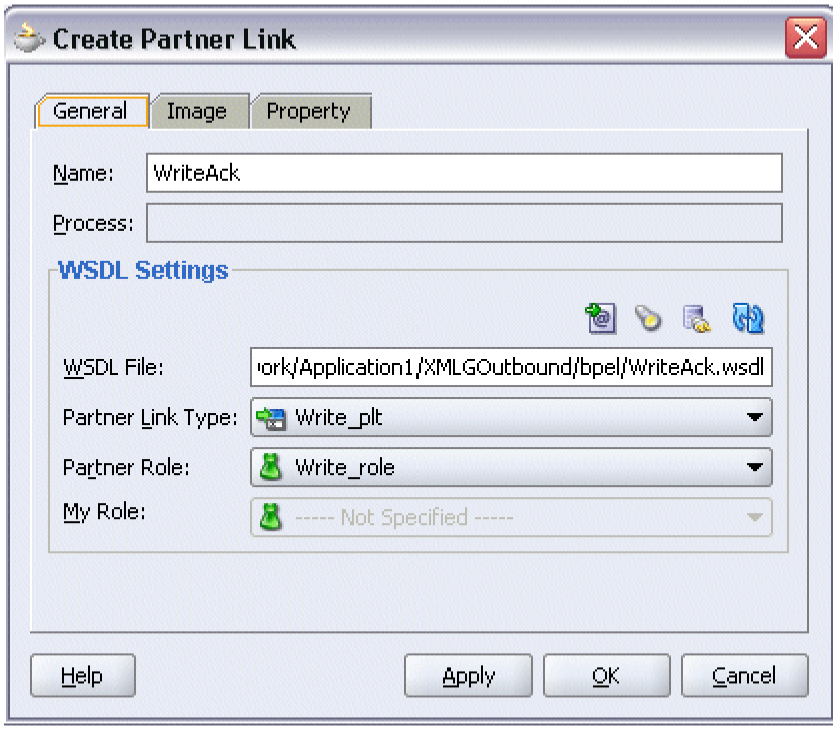


Type: {http://TargetNamespace.com/ServiceName}CONFIRM_BOD_002

Click **OK** to populate the selected schema location and element.

9. Click **Next** and then **Finish**. The wizard generates the WSDL file corresponding to the partner link. The main Create Partner Link dialog box appears, specifying the new WSDL file `WriteAck.wsdl`.

Completing the Partner Link Configuration



Create Partner Link

Name: WriteAck

Process:

WSDL Settings

WSDL File: ork/Application1/XMLGOutbound/bpel/WriteAck.wsdl

Partner Link Type: Write_plt

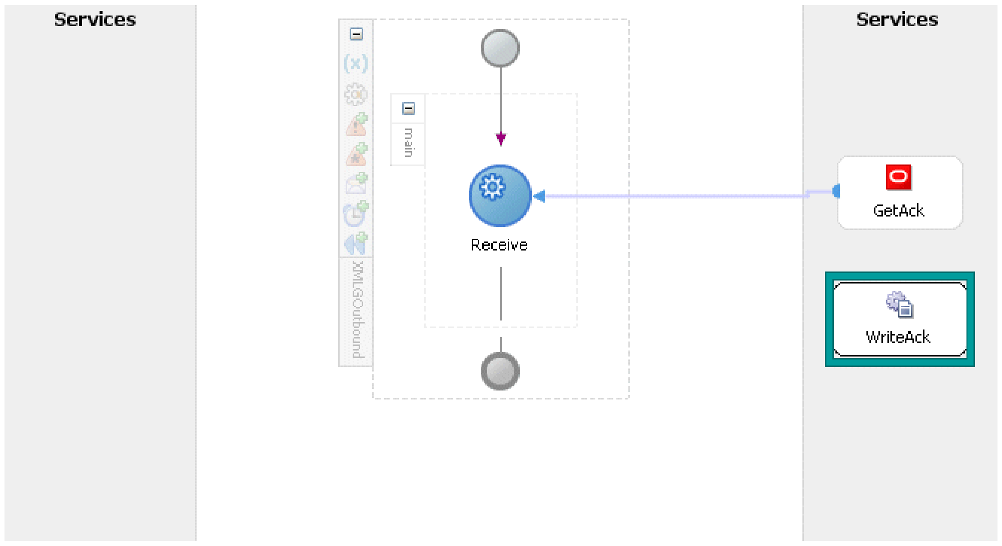
Partner Role: Write_role

My Role: ----- Not Specified -----

Click **Apply** and then **OK** to complete the configuration and create the partner link with the required WSDL settings for the File Adapter service.

The `WriteAck` Partner Link appears in the BPEL process diagram.

Adding the Partner Link for File Adapter



Adding an Invoke Activity

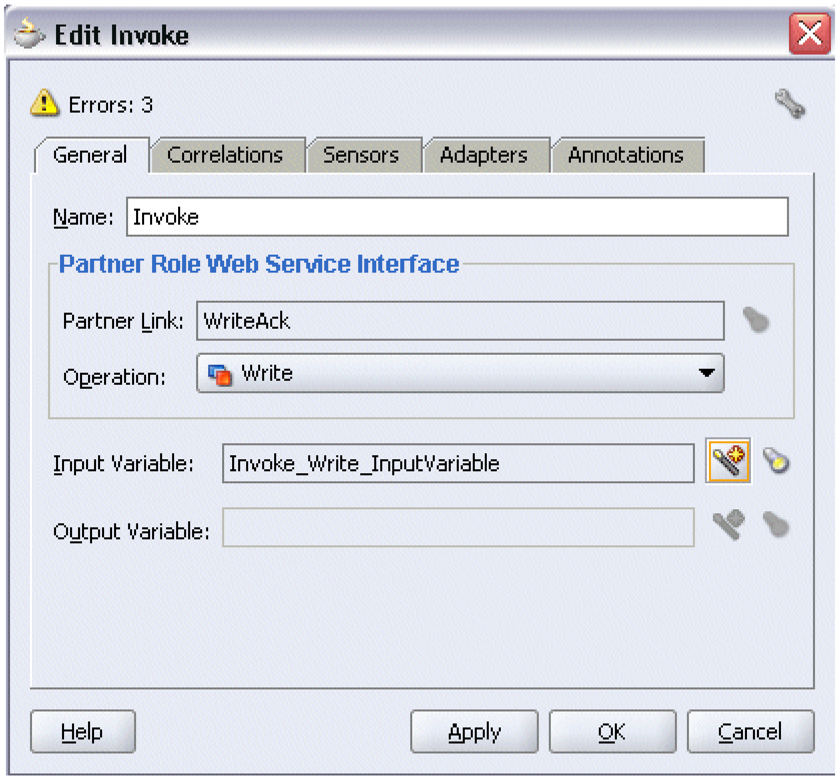
This step is to configure an Invoke activity to send the purchase order acknowledgement that is received from the Receive activity to the `WriteAck` partner link in an XML file.

To add an Invoke activity:

1. In JDeveloper BPEL Designer, drag and drop the **Invoke** activity from the Component Palette into the process diagram, after the **Receive** activity.

2. Link the Invoke activity to the `WriteAck` service. The Invoke activity will send event data to the partner link. The Edit Invoke dialog box appears.

Editing the Invoke Activity


Edit Invoke

Errors: 3

Name: Invoke

Partner Role Web Service Interface

Partner Link: WriteAck

Operation: Write

Input Variable: Invoke_Write_InputVariable

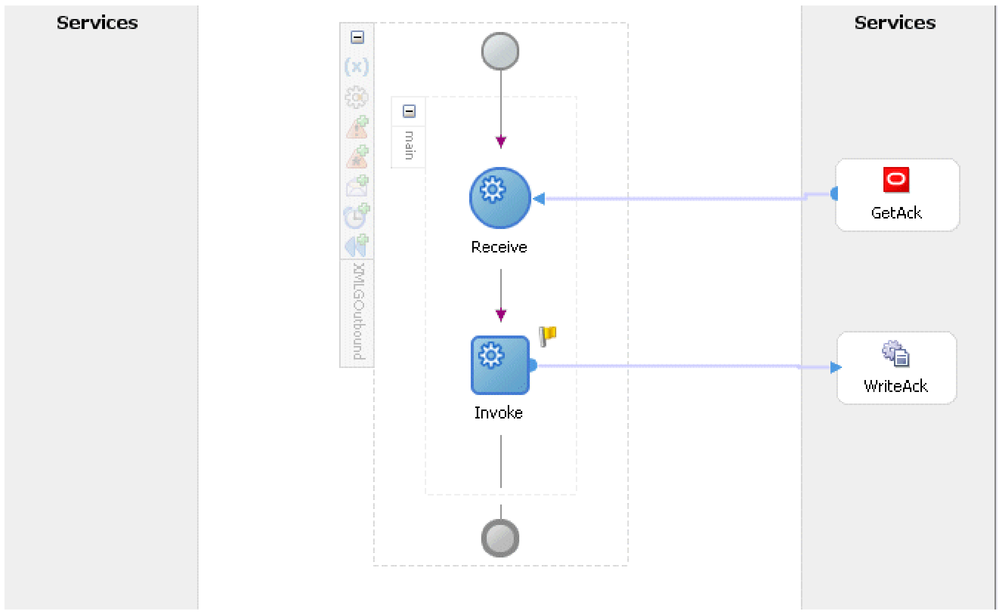
Output Variable:

3. Enter a name for the Invoke activity, and click the **Create** icon next to the **Input Variable** field to create a new variable. The Create Variable dialog box appears.

4. Select **Global Variable** and then enter a name for the variable. You can also accept the default name. Click **OK**.

   Click **Apply** and then **OK** in the Edit Invoke dialog box to finish configuring the Invoke activity.

   The Invoke activity appears in the process diagram.



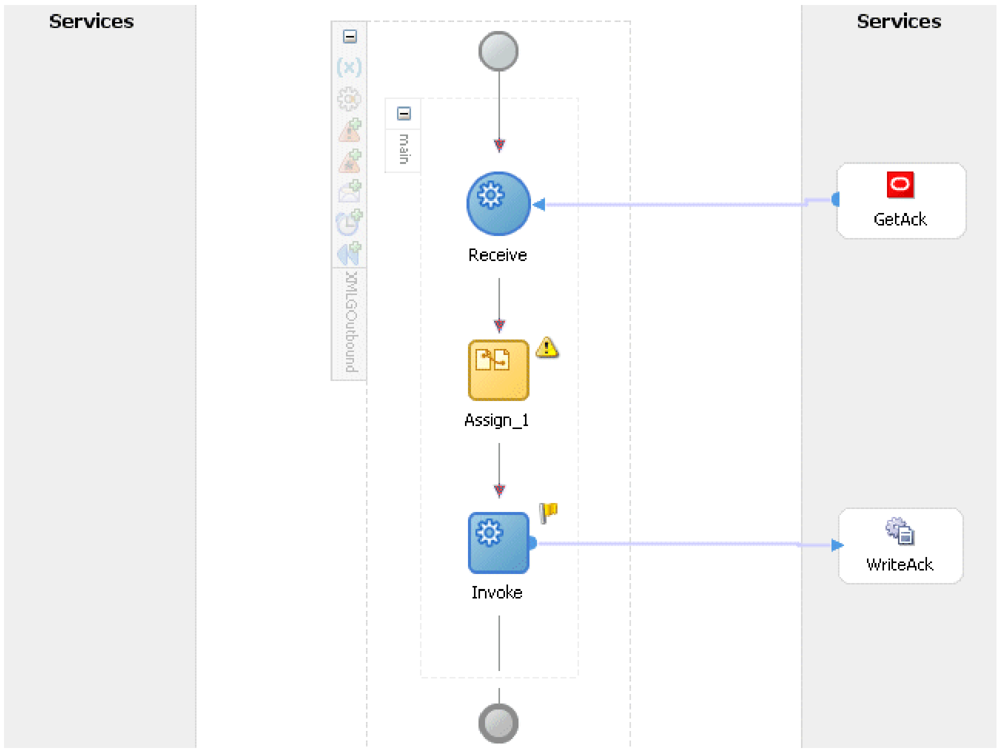
Adding an Assign Activity

Use this step to pass the purchase order acknowledgement details from the Receive activity to the Invoke activity.

To add an Assign activity:

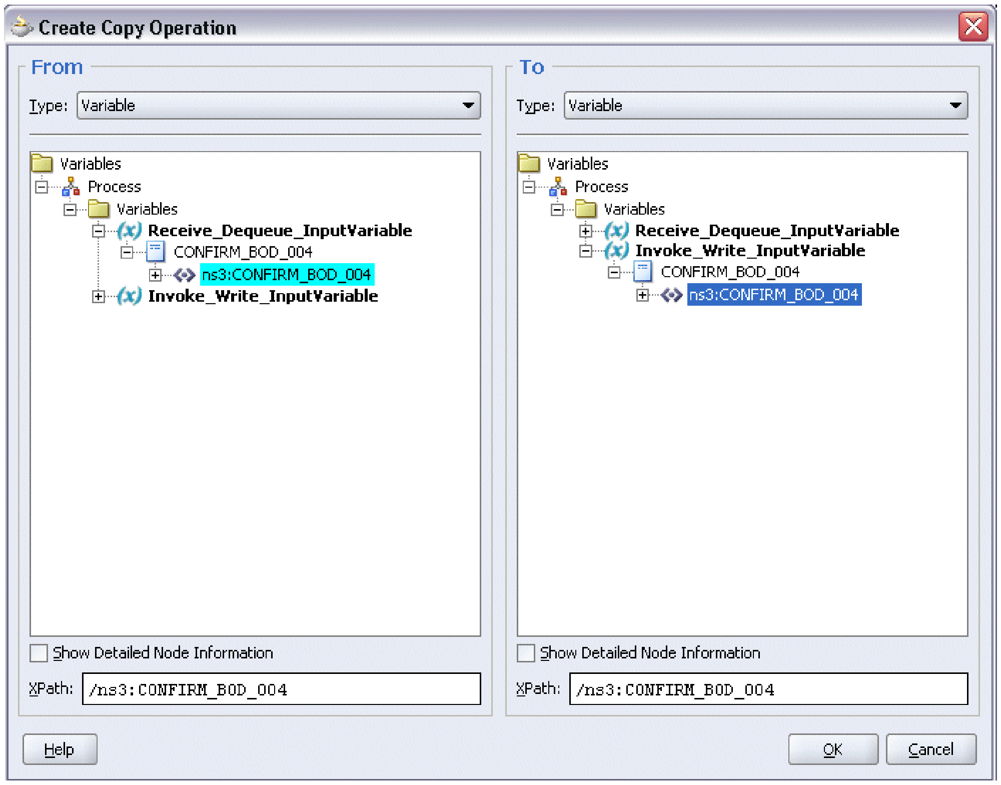
1. In JDeveloper BPEL Designer, drag and drop the **Assign** activity from the Component Palette into the Activity box of the process diagram, between the **Receive** activity and the **Invoke** activity.

Adding an Assign Activity



2. Double-click the **Assign** activity to access the Edit Assign dialog box.
3. On the Copy Operation tab, click **Create** and select **Copy Operation** from the menu. The Create Copy Operation window appears.
4. In the From navigation tree, select type Variable. Navigate to **Variable** > **Process** > **Variables** > **Receive_DEQUEUE_InputVariable** > **CONFIRM_BOD_004** and select **ns3:CONFIRM_BOD_004**. The XPath field should contain your selected entry.
5. In the To navigation tree, select type Variable. Navigate to **Variable** > **Process** > **Variables** > **Invoke_Write_InputVariable** > **CONFIRM_BOD_004** and select **ns3:CONFIRM_BOD_004**. The XPath field should contain your selected entry.

Specifying Assign Parameters



From — Type: Variable

Variables
Process
Variables
Receive_Dequeue_InputVariable
CONFIRM_BOD_004
ns3:CONFIRM_BOD_004
Invoke_Write_InputVariable

Show Detailed Node Information

XPath: /ns3:CONFIRM_BOD_004

To — Type: Variable

Variables
Process
Variables
Receive_Dequeue_InputVariable
Invoke_Write_InputVariable
CONFIRM_BOD_004
ns3:CONFIRM_BOD_004

Show Detailed Node Information

XPath: /ns3:CONFIRM_BOD_004

6. Click **OK**.

   Click **Apply** and then **OK** in the Edit Assign dialog box to complete the configuration of the Assign activity.

Run-Time Task for XML Gateway Outbound Messaging

After designing the BPEL process, you can compile, deploy and test it.

1. Deploy the BPEL process, page 4-68

2. Test the BPEL process, page 4-69

Deploying the BPEL Process

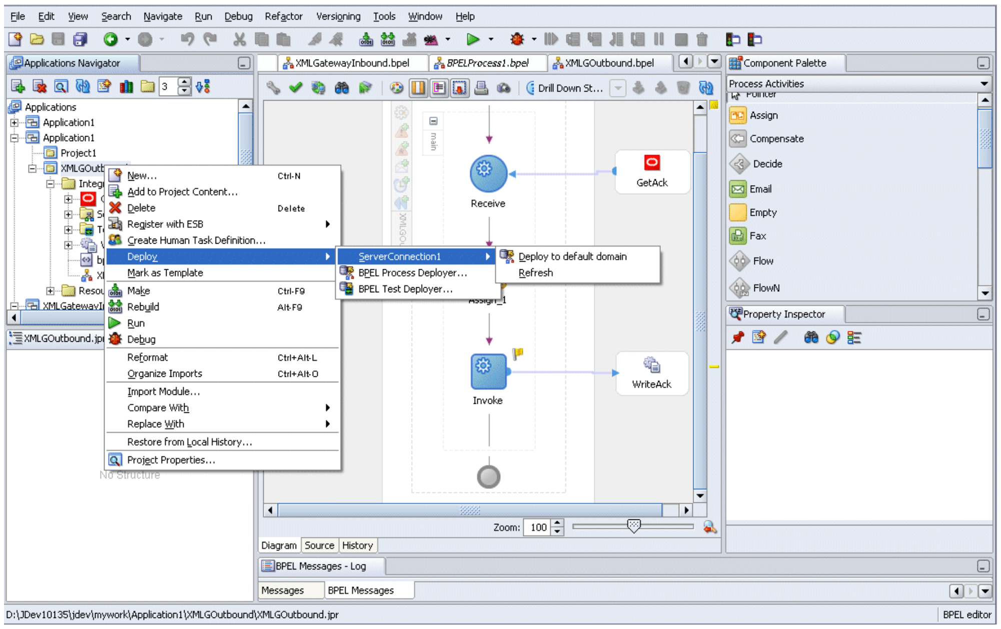
You must deploy the BPEL process before you can run it. The BPEL process is first compiled, and then deployed to the BPEL server.

To deploy the BPEL process:

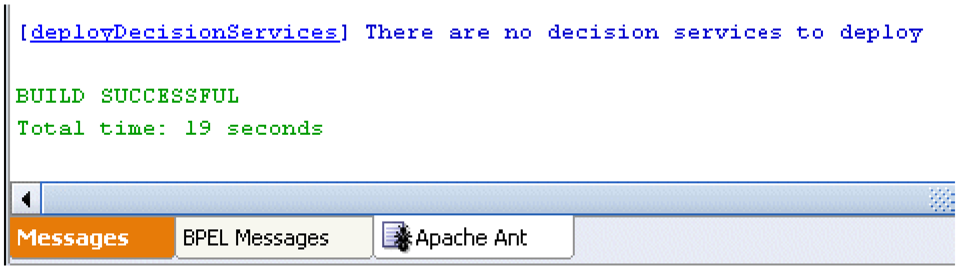
1. Select the BPEL project in the Applications window.

2. Right-click the project name, and then select **Deploy** > **[Server Connection]** > **Deploy to Default Domain** from the menu that appears.

Deploy the BPEL Process



3. The BPEL process is compiled and deployed. You can check if the deployment is successful in the Apache Ant log.



```
[deployDecisionServices] There are no decision services to deploy

BUILD SUCCESSFUL
Total time: 19 seconds
```

Testing the BPEL Process

In Oracle Applications, you can check for outbound transactions that have been processed by the XML Gateway and delivered to the Transaction Agent by using the Transaction Monitor. You can also use the Transaction Monitor to resend an outbound document if necessary.

Note: For details on using the Transaction Monitor, see *Oracle XML Gateway User's Guide.*

If you have used a File Adapter to write outbound messages from Oracle Applications to files, you can check the output directory location for the presence of these files after the BPEL process has run.

To validate the design-time tasks created earlier, you can log in to Oracle E-Business Suite to manually create and book the order as well as generate the order acknowledgement by submitting a Workflow Background Process concurrent request.

To manually test the BPEL process:

1. Log in to Oracle Applications with the XML Gateway responsibility.

   This is to ensure that the XML Gateway trading partner is set up correctly so that a purchase order can have a valid customer that has been defined.

2. Select Define Trading Partner from the navigation menu to access the Trading Partner Setup window.

3. Enter the header values in the Trading Partner Setup form as follows:

   - Trading Partner Type: Customer
   - Trading Partner Name: For example, Business World
   - Trading Partner Site: Enter a trading partner site information. For example, 2391 L Street, San Jose, CA 95106
   - Company Admin Email: Enter a valid e-mail address.

4. Enter the following trading partner details:

   - Transaction Type: ECX
   - Transaction SubType: CBODO
   - Standard Code: OAG
   - External Transaction Type: BOD
   - External Transaction SubType: CONFIRM
   - Direction: Out
   - Map: ECX_CBODO_OAG72_OUT_CONFIRM
   - Connection / Hub: DIRECT

- Protocol Type: BPEL
- Username: 'operation'
- Password: enter 'welcome' twice
- Protocol Address: 'http://ebssoa.sample.com'
- Source Trading Partner Location Code: BWSANJOSE

5. Save your work.

To successfully generated PO Acknowledgement, you need to create an order and then manually book the order through Order Management.

Use the following steps to create an order and then manually book the order:

1. Switch to Order Management Super User, Vision Operations (USA) responsibility and select Customer > Standard from the navigation menu to open the Enter Customer form.
2. Search on the 'Business World' in the Name field and click **Find**.
3. Select the Business World with the following information from the search results:
   - Account Name: Business World
   - Registry ID: 2813
   - Identifying check box: checked
   - Address: 2391 L Street, San Jose, CA 95106
   - Country: United States of America
4. Select row with the following entries:
   - Account Number:1608
   - Account Description: Business World
   - Status: Active
5. Click **Details** to open the Customer Information page.
6. Click **Details** in the row with the Business World with Address '2391 L Street, San Jose, CA 95106' and Country 'United States of America'. This opens the Customer Account page.

7. Enter 'BWSANJOSE' in the EDI Location field.

8. In the Business Purposes tab, create a new row with the following values:

   - Usage: Sold To
   - Check on 'Primary' Check box

   Save your work.

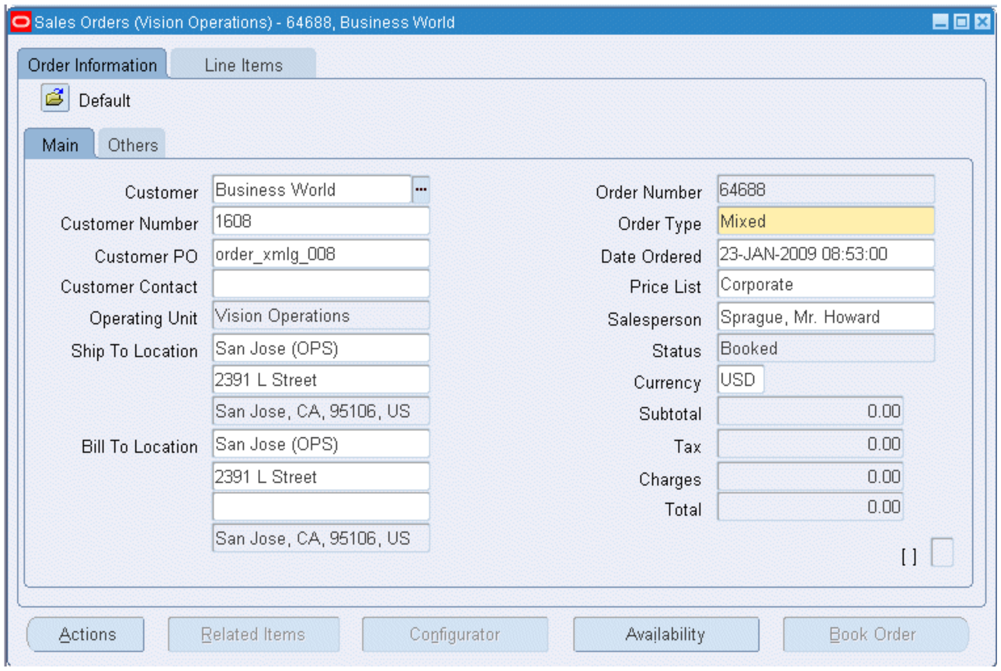
Use the following steps to generate acknowledgement for an already created order:

1. Log in to Oracle Applications with the Order Management Super User, Vision Operations (USA) responsibility. Select Order Returns > Sales Order to open the Sales Order form.

2. Retrieve the order that you have created earlier by entering the order ID in the Customer PO field, such as `order_xmlg_008`.

3. Click **Book Order** to book the order.

Booking an Order



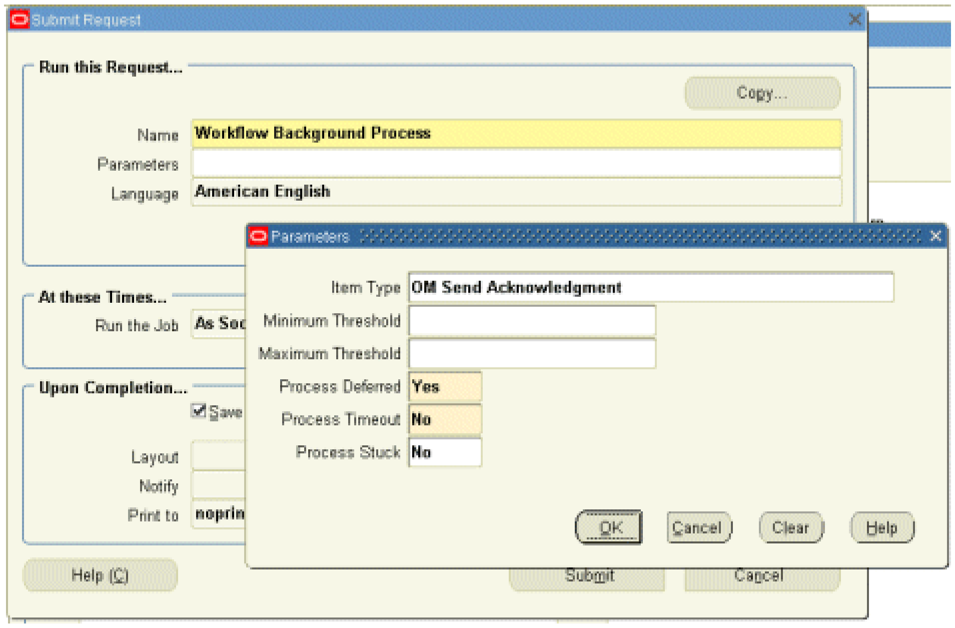
Customer	Business World	Order Number	64688
Customer Number	1608	Order Type	Mixed
Customer PO	order_xmlg_008	Date Ordered	23-JAN-2009 08:53:00
Customer Contact		Price List	Corporate
Operating Unit	Vision Operations	Salesperson	Sprague, Mr. Howard
Ship To Location	San Jose (OPS)	Status	Booked
	2391 L Street	Currency	USD
	San Jose, CA, 95106, US	Subtotal	0.00
Bill To Location	San Jose (OPS)	Tax	0.00
	2391 L Street	Charges	0.00
		Total	0.00
	San Jose, CA, 95106, US		

4. Switch to the System Administrator responsibility and select Request > Run.

5. Select **Single Request** and click **OK**.

6. Enter the following information in the Submit Request form:

Specifying Parameters


Submit Request

Run this Request...

Name: Workflow Background Process

Parameters:

Language: American English

At these Times...

Run the Job: As Soo

Upon Completion...

☑ Save

Layout:

Notify:

Print to: noprin

Parameters

Item Type: OM Send Acknowledgment

Minimum Threshold:

Maximum Threshold:

Process Deferred: Yes

Process Timeout: No

Process Stuck: No

- Name: Workflow Background Process
- Enter the following parameters:
  - Item Type: OM Send Acknowledgement
  - Process Deferred: Y
  - Process Timeout: N
  - Process Stuck: N
- Click **OK**.

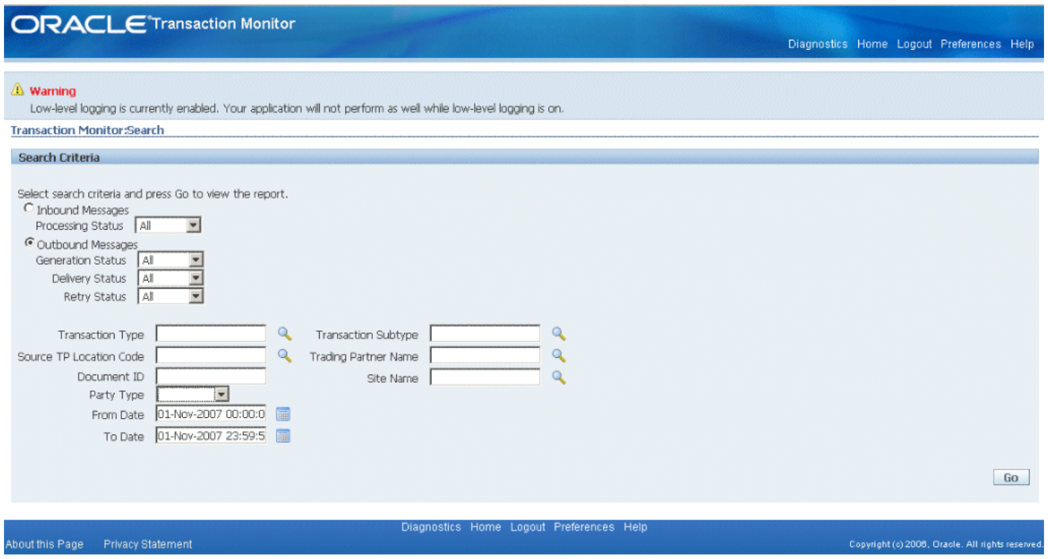
7. Click **Submit** to submit 'send acknowledgement' request.

8. View your request by entering the request ID to ensure its status is 'Success'.

Validating Using Oracle Transaction Monitor

To validate it using Oracle Transaction Monitor, you need to log in to Oracle Applications with the Workflow Administrator Web Applications responsibility. Select Transaction Monitor to open the search window to search for the order.

Searching from the Transaction Monitor


ORACLE Transaction Monitor

Diagnostics Home Logout Preferences Help

Warning
Low-level logging is currently enabled. Your application will not perform as well while low-level logging is on.

Transaction Monitor:Search

Search Criteria

Select search criteria and press Go to view the report.

Inbound Messages
Processing Status: All

(●) Outbound Messages
Generation Status: All
Delivery Status: All
Retry Status: All

Transaction Type		Transaction Subtype	
Source TP Location Code		Trading Partner Name	
Document ID		Site Name	
Party Type			
From Date	01-Nov-2007 00:00:0		
To Date	01-Nov-2007 23:59:5		

Go

Diagnostics Home Logout Preferences Help

About this Page Privacy Statement

Copyright (c) 2006, Oracle. All rights reserved.

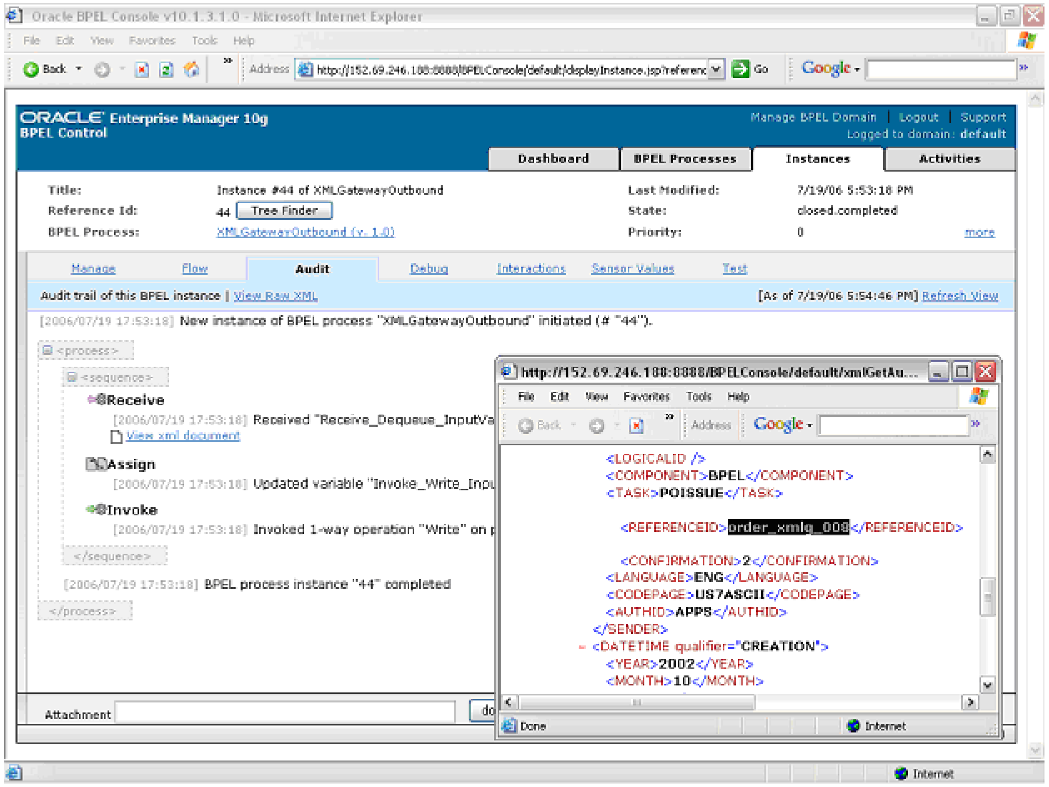
Validating Using Oracle BPEL Console

Log in to Oracle BPEL Console to confirm that the `XMLGOutbound` process has been deployed. This process is continuously polling the ECX_OUTBOUND queue for purchase order acknowledgement.

To verify, select the instance of your deployed process. This opens up in the Instances tab of your selected BPEL process.

Click on the Audit Tab to view the Receive activity. Click the **view xml document** link to open the XML file received. Please note that the Reference ID in the XML file represents the Customer PO in the Sales Orders form.

Viewing XML File for the Receive Activity



Title: Instance #44 of XMLGatewayOutbound
Reference Id: 44
BPEL Process: XMLGatewayOutbound (v. 1.0)
Last Modified: 7/19/06 5:53:18 PM
State: closed.completed
Priority: 0

Audit trail of this BPEL instance | View Raw XML [As of 7/19/06 5:54:46 PM] Refresh View

```
[2006/07/19 17:53:18] New instance of BPEL process "XMLGatewayOutbound" initiated (# "44").
<process>
  <sequence>
    Receive
      [2006/07/19 17:53:18] Received "Receive_Dequeue_InputVa
      View xml document
    Assign
      [2006/07/19 17:53:18] Updated variable "Invoke_Write_Inpu
    Invoke
      [2006/07/19 17:53:18] Invoked 1-way operation "Write" on p
  </sequence>
  [2006/07/19 17:53:18] BPEL process instance "44" completed
</process>
```

```
      <LOGICALID />
      <COMPONENT>BPEL</COMPONENT>
      <TASK>POISSUE</TASK>

        <REFERENCEID>order_xmlg_008</REFERENCEID>

        <CONFIRMATION>2</CONFIRMATION>
      <LANGUAGE>ENG</LANGUAGE>
      <CODEPAGE>US7ASCII</CODEPAGE>
      <AUTHID>APPS</AUTHID>
    </SENDER>
  - <DATETIME qualifier="CREATION">
      <YEAR>2002</YEAR>
      <MONTH>10</MONTH>
```

On approval of the order, Oracle E-Business Suite triggers the workflow that creates the Purchase Order Acknowledgement flow and sends out the PO Acknowledgement as an XML file. The workflow delivers the Confirm BOD as the PO Acknowledgement to the ECX_OUTBOUND queue for delivery to the other system.

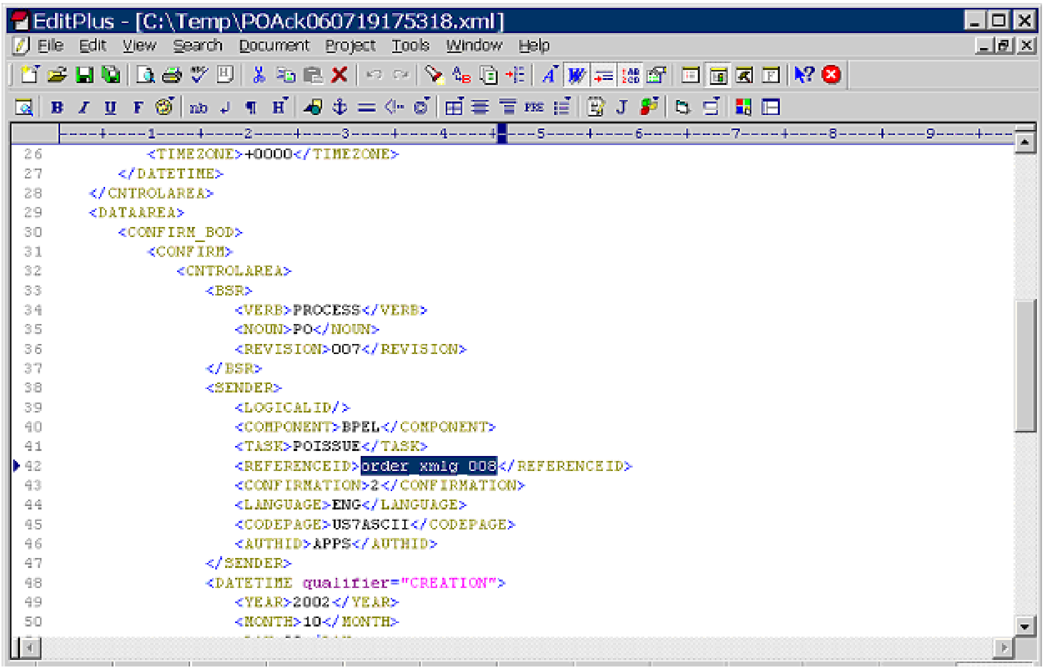
On the other hand, Oracle BPEL PM listens to the ECX_OUTBOUND queue for the message with the correlation Id = "BPEL" which is the same id for this outbound message. The Confirm BOD as the PO Acknowledgement is written as XML file using File Adapter.

Since the BPEL process is deployed, the process is continuously polling the ECX_OUTBOUND queue for PO acknowledgement. It also writes PO Acknowledgement in the directory you specified after receiving from XML Gateway ECX_OUTBOUND queue.

Go to the directory you specified for the write operation, such as `outputDir` logical location (typically `c:\temp`) where the File Adapter has placed the file after writing the PO Acknowledgement in an XML file (`POAck060719175318.xml`).

Open this `POAck060719175318.xml` file. You should find the Reference ID as `order_xmlg_008` (the order booked) for which the acknowledgement is generated.

Validating the Output File


EditPlus - [C:\Temp\POAck060719175318.xml]
File Edit View Search Document Project Tools Window Help

```
      <TIMEZONE>+0000</TIMEZONE>
   </DATETIME>
</CNTROLAREA>
<DATAAREA>
   <CONFIRM_BOD>
      <CONFIRM>
         <CNTROLAREA>
            <BSR>
               <VERB>PROCESS</VERB>
               <NOUN>PO</NOUN>
               <REVISION>007</REVISION>
            </BSR>
            <SENDER>
               <LOGICALID/>
               <COMPONENT>BPEL</COMPONENT>
               <TASK>POISSUE</TASK>
               <REFERENCEID>order xmlg 008</REFERENCEID>
               <CONFIRMATION>2</CONFIRMATION>
               <LANGUAGE>ENG</LANGUAGE>
               <CODEPAGE>US7ASCII</CODEPAGE>
               <AUTHID>APPS</AUTHID>
            </SENDER>
            <DATETIME qualifier="CREATION">
               <YEAR>2002</YEAR>
               <MONTH>10</MONTH>
```

Troubleshooting and Debugging

If you experience problems with your Oracle XML Gateway integration, you can take the following troubleshooting steps:

- Confirm that you have the correct settings for the following elements of the trading partner setup:
  - Standard Code
  - Transaction Type
  - Transaction Subtype
  - Source Trading Partner Location Code (Party Site ID)
- Confirm that the correct transaction is enabled for the trading partner.
- Check the status of the XML transaction in Transaction Monitor.
- Ensure that the document number is unique within this transaction type.
- For inbound transactions, confirm that ECX Listeners are running.

- For outbound transactions, confirm that Background Engine is running.
- In the trading partner setup, ensure that the Protocol Type is set to `BPEL`.
- Specify the same correlation ID for Oracle Applications as for the Adapter.

  The Adapter Configuration Wizard of OracleAS Adapter for Oracle Applications does not specify a correlation ID for XML Gateway transactions for inbound or outbound interfaces. Instead, a default correlation ID of `BPEL` is automatically set in the WSDL file. To make this configuration work, you must configure Oracle Applications to set the same correlation ID value of `BPEL` for the corresponding XML Gateway transactions.

  If you want the Adapter to use a different correlation ID than the default, you need to configure a correlation ID in Oracle Applications, and then edit the `Correlation="BPEL"` line contained in the `<jca:operation>` section of the adapter service WSDL. Replace `BPEL` with the string value of the correlation ID you specified in Oracle Applications.

If you still experience problems with your integration, you can enable debugging.

Enabling Debugging

You can enable debugging for XML Gateway using the BPEL Process Manager.

To enable debugging:

1. Log in to your BPEL Process Manager domain.
2. Select `*yourdomain*.collaxa.cube.ws`
3. Select **Debug.**
4. Enable FND Logging to debug XML Gateway transactions.

Debugging information is output to the log file for your domain. To examine the log file in the BPEL Process Manager, navigate to **Home > BPEL Domains >*yourdomain* > Logs**. The log file is ***yourdomain*.log**.

5 Using Business Events for BPEL Process Integration

This chapter covers the following topics:

- Overview of Business Events
- Design-Time Tasks for Outbound Business Events
- Creating a New BPEL Project
- Creating a Partner Link
- Configuring the Receive Activity
- Adding a Partner Link for the File Adapter
- Configuring the Invoke Activity
- Configuring the Assign Activity
- Run-Time Tasks for Outbound Business Events
- Deploying the BPEL Process
- Testing the BPEL Process
- Troubleshooting and Debugging

Overview of Business Events

The *Oracle Workflow Business Event System* (BES) is an application service that leverages the Oracle Advanced Queuing (AQ) infrastructure to communicate business events between systems. The Business Event System consists of the Event Manager and workflow process event activities.

The Event Manager contains a registry of business events, systems, named communication agents within those systems, and subscriptions indicating that an event is significant to a particular system. Events can be raised locally or received from an external system or the local system through AQ. When a local event occurs, the subscribing code is executed in the same transaction as the code that raised the event, unless the subscriptions are deferred.

Subscriptions can include the following types of processing:

- Executing custom code on the event information
- Sending event information to a workflow process
- Sending event information to other queues or systems

Business events are represented within workflow processes by event activities. By including event activities in a workflow process, you can model complex processing or routing logic for business events beyond the options of directly running a predefined function or sending the event to a predefined recipient.

Each business event represents a ready to use integration or extension point. Oracle E-Business Suite currently ships preconfigured with over 900 business events.

The uses of the Business Event System include:

- **System integration messaging hubs** - Oracle Workflow with the Business Event System can serve as a messaging hub for complex system integration scenarios. The Event Manager can be used to "hard–wire" routing between systems based on event and originator. Workflow process event activities can be used to model more advanced routing, content–based routing, transformations, error handling, and so on.

- **Distributed applications messaging** - Applications can supply Generate and Receive event message handlers for their business entities. For example, message handlers can be used to implement Master/Copy replication for distributed applications.

- **Message-based system integration** - You can set up subscriptions, which cause messages to be sent from one system to another when business events occur. In this way, you can use the Event Manager to implement point–to–point messaging integration.

- **Business–event based workflow processes** - You can develop sophisticated workflow processes that include advanced routing or processing based on the content of business events.

- **Non-invasive customization of packaged applications** - Analysts can register interesting business events for their Internet or intranet applications. Users of those applications can register subscriptions to those events to trigger custom code or workflow processes.

Business Events Concepts

Event

A business event is an occurrence in an Internet or intranet application or program that might be significant to other objects in a system or to external agents. For instance, the creation of a purchase order is an example of a business event in a purchasing application.

Event Key

A string that uniquely identifies an instance of an event. Together, the event name, event key, and event data fully communicate what occurred in the event.

Event Message

A standard Workflow structure for communicating business events, defined by the datatype WF_EVENT_T. The event message contains the event data as well as several header properties, including the event name, event key, addressing attributes, and error information.

Event Activity

A business event modeled as an activity so that it can be included in a workflow process.

Event Data

A set of additional details describing an event. The event data can be structured as an XML document. Together, the event name, event key, and event data fully communicate what occurred in the event.

Event Subscription

A registration indicating that a particular event is significant to a system and specifying the processing to perform when the triggering event occurs. Subscription processing can include calling custom code, sending the event message to a workflow process, or sending the event message to an agent.

Deferred Subscription Processing

If you do not want subscriptions for an event to be executed immediately when the event occurs, you can defer the subscriptions. In this way you can return control more quickly to the calling application and let the Event Manager execute any costly subscription processing at a later time.

Agent

An agent is a named point of communication within a system. Communication within and between systems is accomplished by sending a message from one agent to another. A single system can have several different agents representing different communication alternatives. For example, a system may have different agents to support inbound and outbound communication, communication by different protocols, different propagation frequencies, or other alternatives.

Design-Time Tasks for Outbound Business Events

OracleAS Adapter for Oracle Applications is deployed at design-time using Oracle JDeveloper and at run-time using the BPEL Process Manager.

This section discusses the process of configuring OracleAS Adapter for Oracle Applications to create business event outbound subscriptions. It describes the tasks required to configure OracleAS Adapter for Oracle Applications using the Adapter Configuration Wizard in Oracle JDeveloper.

Multiple BPEL Processes Consuming the Same Business Event

Please note that OracleAS Adapter for Oracle Applications can handle multiple BPEL processes consuming the same business event. OracleAS Adapter for Oracle Applications creates only single subscription for a particular business event regardless of the number of BPEL process consuming it. Internally, this subscription forwards business event message to a multi-consumer AQ. Since each BPEL process is an unique consumer for the event, when the message is placed in the queue, all BPEL processes are notified. Therefore, as a user you do not need to create a separate subscription for each BPEL process. All you need to do is to create the service for the event, and OracleAS Adapter for Oracle Applications will take care of message delivery to each BPEL process.

For example, if there are three BPEL processes (BPEL1, BPEL2, and BPEL3) that want to consume the same business event (such as BE1 event). For each BPEL process, you create a service for the BE1 event using OracleAS Adapter for Oracle Applications. OracleAS Adapter for Oracle Applications in turn creates a single subscription for all the three BPEL processes - BPEL1, BPEL2, and BPEL3. This subscription puts BE1 event message in multi-consumer AQ.

At run time, when a BE1 event is raised, since the subscription is applicable to all the three BPEL processes, all these three deployed BPEL processes will be activated and would receive the same BE1 event message.

Prerequisites to Configure Outbound Business Events

- The agentListener must be running on WF_Deferred queue.
- The event should be enabled for BPEL to subscribe to it. The event should not be in the disabled mode.

Following is a list of the procedures required to accomplish the design-time tasks.

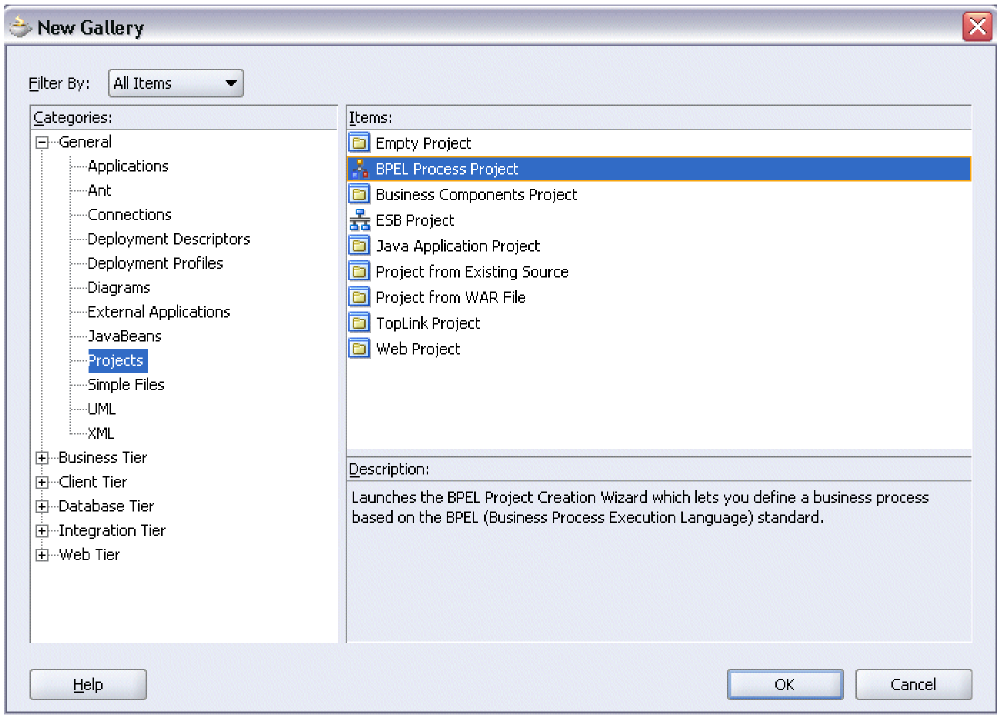
1. Create a new BPEL project, page 5-5
2. Create a partner link, page 5-7
3. Configure the Receive activity, page 5-17
4. Add a partner link for the File Adapter, page 5-19
5. Configure the Invoke activity, page 5-24
6. Configure the Assign activity, page 5-26

Creating a New BPEL Project

To create a new BPEL project:

1. Open JDeveloper BPEL Designer.
2. From the **File** menu, select **New**. The New Gallery dialog box appears.
3. Select **All Items** from the **Filter By** box. This produces a list of available categories.
4. Expand the **General** node, then select **Projects**.
5. Select **BPEL Process Project** from the **Items** group.

Creating a New BPEL Process Project



Launches the BPEL Project Creation Wizard which lets you define a business process based on the BPEL (Business Process Execution Language) standard.

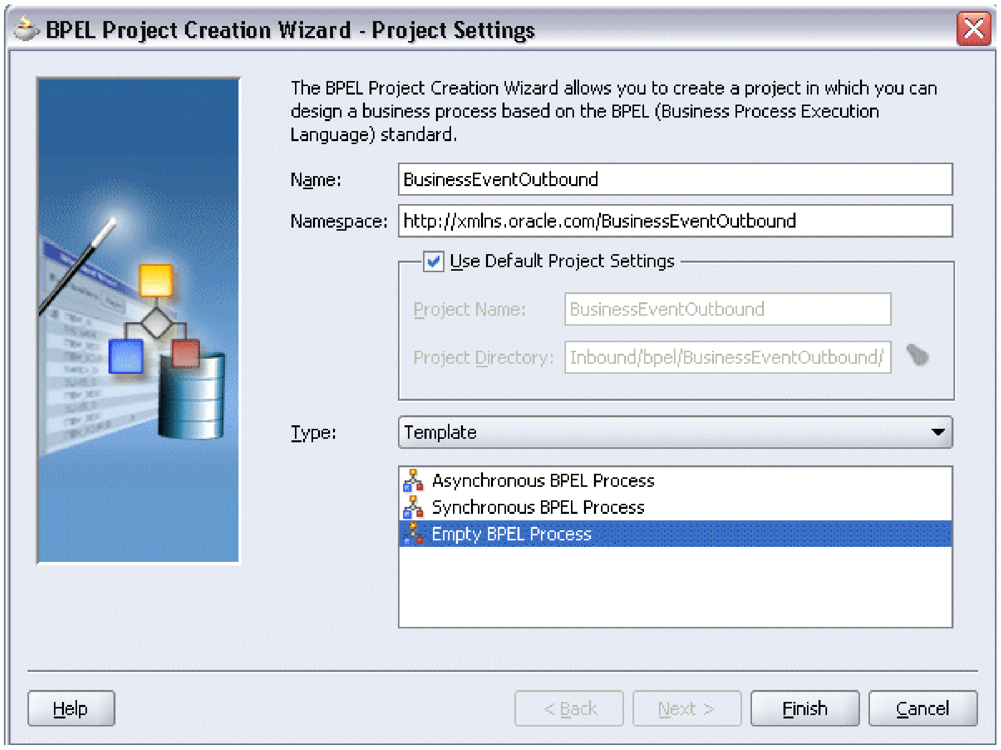
6. Click **OK**. The BPEL Project Creation Wizard - Project Settings dialog box appears.

7. In the **Name** field, enter a descriptive name. For example, enter `BusinessEventOutbound`.

   Keep the default selection **Use Default Project Settings** unchanged.

8. Keep the default selection **Template** as the Type field. Select **Empty BPEL Proces** as the BPEL process type.

Specifying New BPEL Project Settings


BPEL Project Creation Wizard - Project Settings

The BPEL Project Creation Wizard allows you to create a project in which you can design a business process based on the BPEL (Business Process Execution Language) standard.

Name: BusinessEventOutbound

Namespace: http://xmlns.oracle.com/BusinessEventOutbound

[✓] Use Default Project Settings

Project Name: BusinessEventOutbound

Project Directory: Inbound/bpel/BusinessEventOutbound/

Type: Template

Asynchronous BPEL Process
Synchronous BPEL Process
Empty BPEL Process

9. Click **Finish**.

   A new BPEL process is created with the required source files including `bpel.xml`, `BusinessEventOutbound.bpel`, and `BusinessEventOutbound.wsdl`.

Creating a Partner Link

Configuring an outbound business event requires creating a partner link to allow the outbound event to be published to the Oracle BPEL Process Manager.

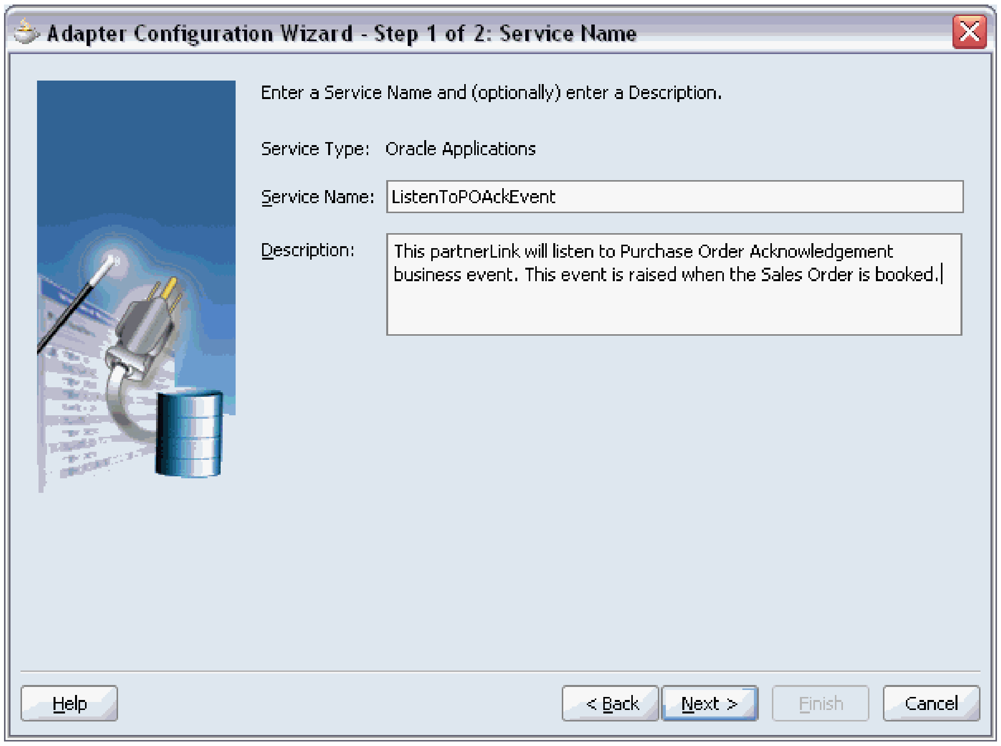
This task adds a partner link to the BPEL process. A partner link defines the link name, type, and the role of the BPEL process that interacts with the partner service.

To create a partner link:

1. In JDeveloper BPEL Designer, drag and drop the **Oracle Applications** adapter service from the Component Palette into the Partner Link border area of the process diagram. The Adapter Configuration Wizard appears.

2. Click **Next**. The Service Name dialog box appears.

Specifying the Service Name



Enter a Service Name and (optionally) enter a Description.

Service Type: Oracle Applications

Service Name: ListenToPOAckEvent

Description: This partnerLink will listen to Purchase Order Acknowledgement business event. This event is raised when the Sales Order is booked.

3. Enter a service name, such as `ListenToPOAckEvent`. You can also add an optional description of the service.

4. Click **Next**. The Service Connection dialog box appears.

5. You can perform either one of the following options for your database connection:

   Note: You need to connect to the database where Oracle E-Business Suite is running.

   - You can create a new database connection by clicking the **New** icon.

     Detailed instructions on how to create a new database connection, see Creating a New Database Connection, page 4-12.

   - You can select an existing database connection that you have configured earlier from the **Connection** drop-down list.

     The Service Connection page will be displayed with the selected connection information. The JNDI (Java Naming and Directory Interface) name corresponding to the database connection appears automatically in the **Database Server JNDI Name** field. Alternatively, you can specify a JNDI name.

Note: When you specify a JNDI name, the deployment descriptor of the Adapter for Oracle Applications must associate this JNDI name with configuration properties required by the adapter to access the database.

The JNDI name acts as a placeholder for the connection used when your service is deployed to the BPEL server. This enables you to use different databases for development and later for production.

6. Once you have created a new connection or selected an existing connection, you can add a business event by browsing through the list of events available in Oracle E-Business Suite.

   Click **Next**.

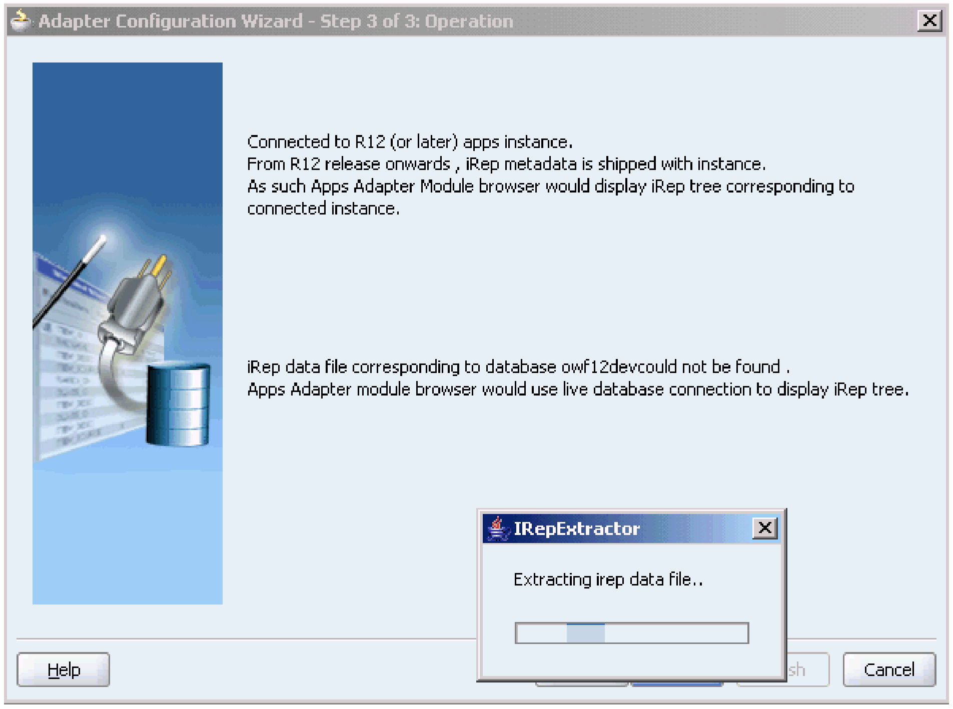
   For Oracle E-Business Suite Release 12:

   If you are connecting to Oracle E-Business Suite Release 12, then the **IREP File not present** dialog box appears indicating that OracleAS Adapter for Oracle Applications could not find the Oracle Integration Repository data file corresponding to the database you are connecting in your workspace. Absence of the data file would make browsing or searching of Integration Repository tree considerably slow. You can choose to extract the data file and create a local copy of the Integration Repository data file. Once it is created successfully, OracleAS Adapter for Oracle Applications will pick it up automatically next time and retrieve data from your local Integration Repository.

   You can select one of the following options:

   - Click **Yes** to extract the Integration Repository data file.

Extracting Integration Repository Data File



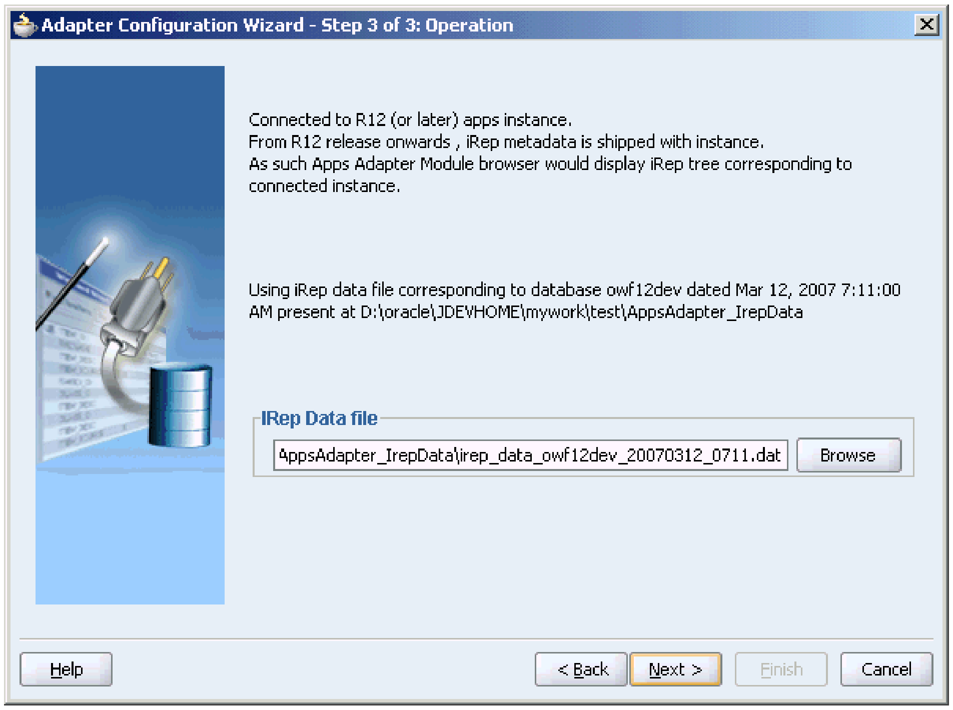
Connected to R12 (or later) apps instance.
From R12 release onwards , iRep metadata is shipped with instance.
As such Apps Adapter Module browser would display iRep tree corresponding to connected instance.

iRep data file corresponding to database owf12devcould not be found .
Apps Adapter module browser would use live database connection to display iRep tree.

Extracting irep data file..

After the system successfully creates a local copy of the Integration Repository data file, next time when you connect to the database, you will find the **IRep Data File** field appears in the Operation dialog box indicating where your local copy exists with the creation date and time as part of the file name.

Using the Local Integration Repository Data File



Connected to R12 (or later) apps instance.
From R12 release onwards , iRep metadata is shipped with instance.
As such Apps Adapter Module browser would display iRep tree corresponding to connected instance.

Using iRep data file corresponding to database owf12dev dated Mar 12, 2007 7:11:00 AM present at D:\oracle\JDEVHOME\mywork\test\AppsAdapter_IrepData

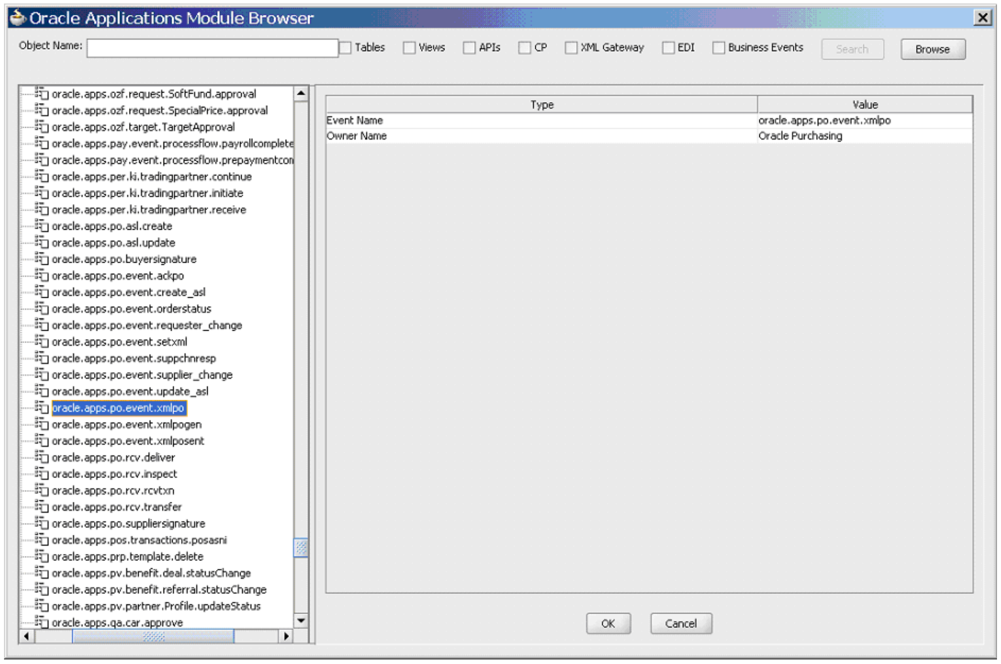
IRep Data file: AppsAdapter_IrepData\irep_data_owf12dev_20070312_0711.dat

- Click **No** to query the Integration Repository data file from the live database you are connecting to display the Integration Repository tree.

  > **Note:** It is highly recommended that you create a local copy of the Integration Repository data file so that OracleAS Adapter for Oracle Applications will query the data next time from the local copy in your workspace to enhance the performance.

7. The Oracle Applications Module Browser opens.

Oracle Applications Module Browser



oracle.apps.ozf.request.SoftFund.approval
oracle.apps.ozf.request.SpecialPrice.approval
oracle.apps.ozf.target.TargetApproval
oracle.apps.pay.event.processflow.payrollcomplete
oracle.apps.pay.event.processflow.prepaymentcon
oracle.apps.per.ki.tradingpartner.continue
oracle.apps.per.ki.tradingpartner.initiate
oracle.apps.per.ki.tradingpartner.receive
oracle.apps.po.asl.create
oracle.apps.po.asl.update
oracle.apps.po.buyersignature
oracle.apps.po.event.ackpo
oracle.apps.po.event.create_asl
oracle.apps.po.event.orderstatus
oracle.apps.po.event.requester_change
oracle.apps.po.event.setxml
oracle.apps.po.event.suppchnresp
oracle.apps.po.event.supplier_change
oracle.apps.po.event.update_asl
oracle.apps.po.event.xmlpo
oracle.apps.po.event.xmlpogen
oracle.apps.po.event.xmlposent
oracle.apps.po.rcv.deliver
oracle.apps.po.rcv.inspect
oracle.apps.po.rcv.rcvtxn
oracle.apps.po.rcv.transfer
oracle.apps.po.suppliersignature
oracle.apps.pos.transactions.posasni
oracle.apps.prp.template.delete
oracle.apps.pv.benefit.deal.statusChange
oracle.apps.pv.benefit.referral.statusChange
oracle.apps.pv.partner.Profile.updateStatus
oracle.apps.qa.car.approve

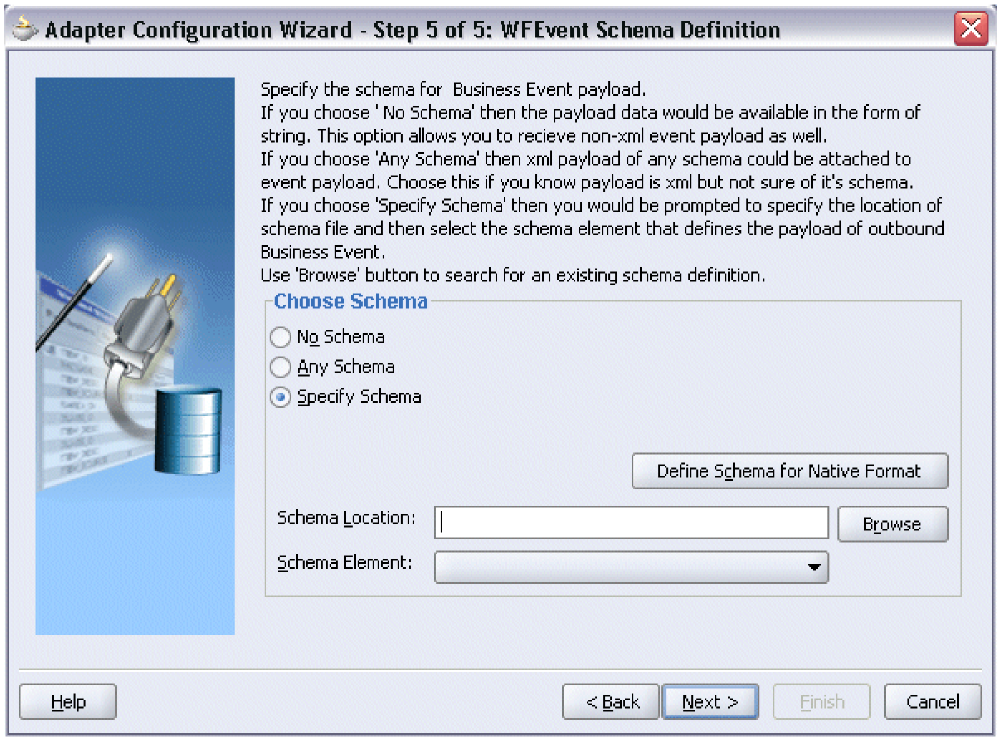
Type	Value
Event Name	oracle.apps.po.event.xmlpo
Owner Name	Oracle Purchasing

Note: Business Events can be found in the Other Interfaces node, which is at the product family level. For more information, see Using the Oracle Applications Module Browser, page 3-23.

8. Expand the navigation tree to **Product Families > Other Interfaces > Business Events > Outbound**. The direction outbound is from the Oracle E-Business Suite perspective, in this case listening to business events from Oracle Applications. Select the appropriate business event, for example, **oracle.apps.po.event.xmlpo**, and click **OK**.

9. Click **Next** in the Operation dialog box. The WF Event Schema Definition dialog box for business event payload appears.

Selecting Business Event Payload Schema


Adapter Configuration Wizard - Step 5 of 5: WFEvent Schema Definition

Specify the schema for Business Event payload.
If you choose ' No Schema' then the payload data would be available in the form of string. This option allows you to recieve non-xml event payload as well.
If you choose 'Any Schema' then xml payload of any schema could be attached to event payload. Choose this if you know payload is xml but not sure of it's schema.
If you choose 'Specify Schema' then you would be prompted to specify the location of schema file and then select the schema element that defines the payload of outbound Business Event.
Use 'Browse' button to search for an existing schema definition.

Choose Schema

No Schema [✓ Specify Schema] Any Schema

Schema Location:

Schema Element:

10. You must specify one of the following options to be used for the business event payload:

No Schema

If you select the No Schema option, then the payload data would be available in the form of string. This option also allows you to receive non-XML event payload.

Any Schema

If you select the Any Schema option, then XML payload of any schema could be attached to event payload. You should select this option if you know the payload is XML, but not sure of its schema.

> **Note:** When you select either the 'No Schema' or 'Any Schema' option, there is no need to further specify the schema information for your business event service, and you will proceed to the next step.

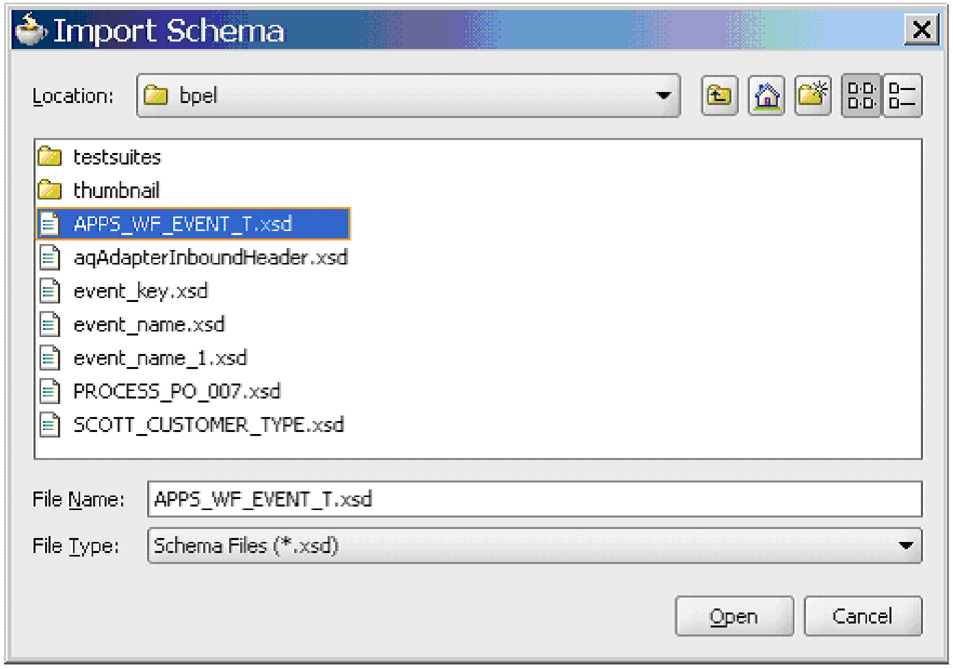
Specify Schema

If you select the Specify Schema option, then the Schema Location and Schema Element fields become visible. You must specify the location of schema file and then select the schema element that defines the payload of outbound business event.

To specify schema location and element

1. Click **Browse** to search for an existing schema definition in the Type Chooser.
2. Click the **Import Schema File** icon at the upper right of the Type Chooser, then click the **Browse File System** icon in the Import Schema File dialog box.

Selecting a Schema File



Import Schema

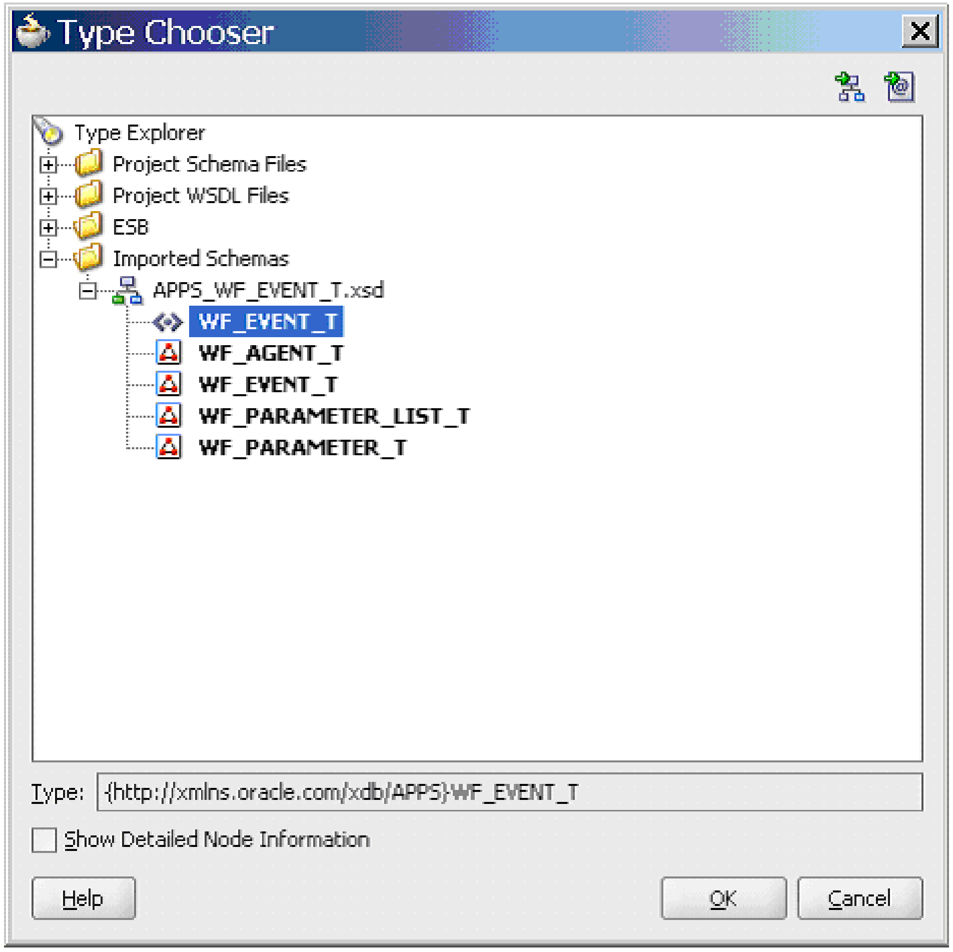
Location: bpel

testsuites
thumbnail
APPS_WF_EVENT_T.xsd
aqAdapterInboundHeader.xsd
event_key.xsd
event_name.xsd
event_name_1.xsd
PROCESS_PO_007.xsd
SCOTT_CUSTOMER_TYPE.xsd

File Name: APPS_WF_EVENT_T.xsd
File Type: Schema Files (*.xsd)

3. In the Import Schema dialog box, navigate to the schema file APPS_WF_EVENT_T.xsd and open it. The Type Chooser reappears with the selected schema in the **Imported Schemas** section.

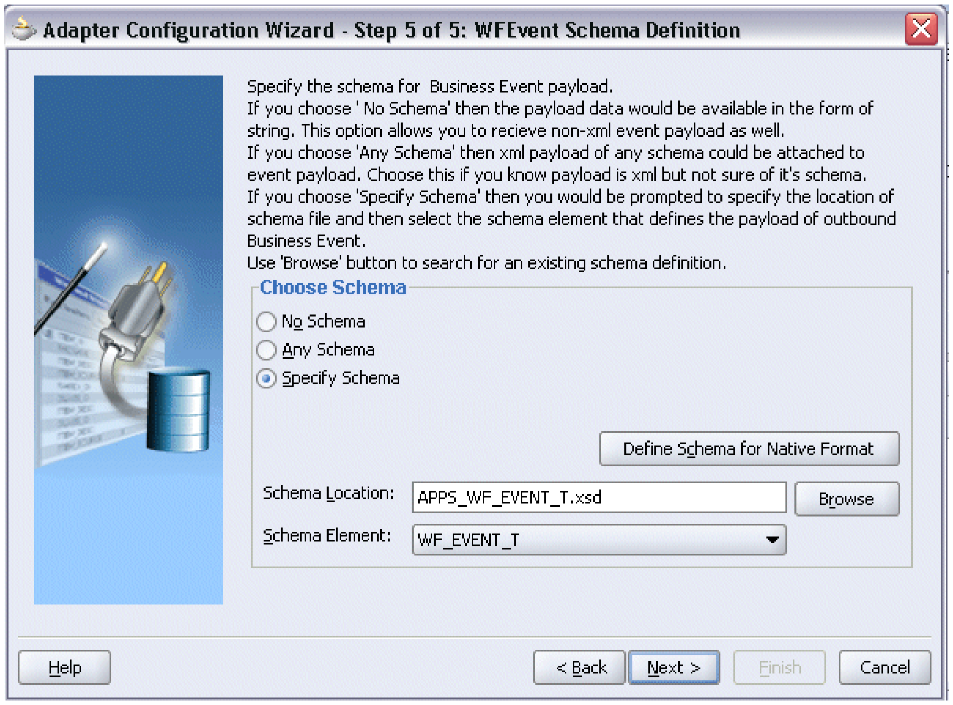
Choosing the Schema



Type: {http://xmlns.oracle.com/xdb/APPS}WF_EVENT_T

4. Select the schema element **WF_EVENT_T** for the business event and click **OK**. The WF Event Schema Definition dialog box reappears with your selected schema location and element information populated. Click **Next**.

Populating Selected Business Event Payload Schema



Adapter Configuration Wizard - Step 5 of 5: WFEvent Schema Definition

Specify the schema for Business Event payload.
If you choose ' No Schema' then the payload data would be available in the form of string. This option allows you to recieve non-xml event payload as well.
If you choose 'Any Schema' then xml payload of any schema could be attached to event payload. Choose this if you know payload is xml but not sure of it's schema.
If you choose 'Specify Schema' then you would be prompted to specify the location of schema file and then select the schema element that defines the payload of outbound Business Event.
Use 'Browse' button to search for an existing schema definition.

Choose Schema

No Schema [✓ Specify Schema] Any Schema

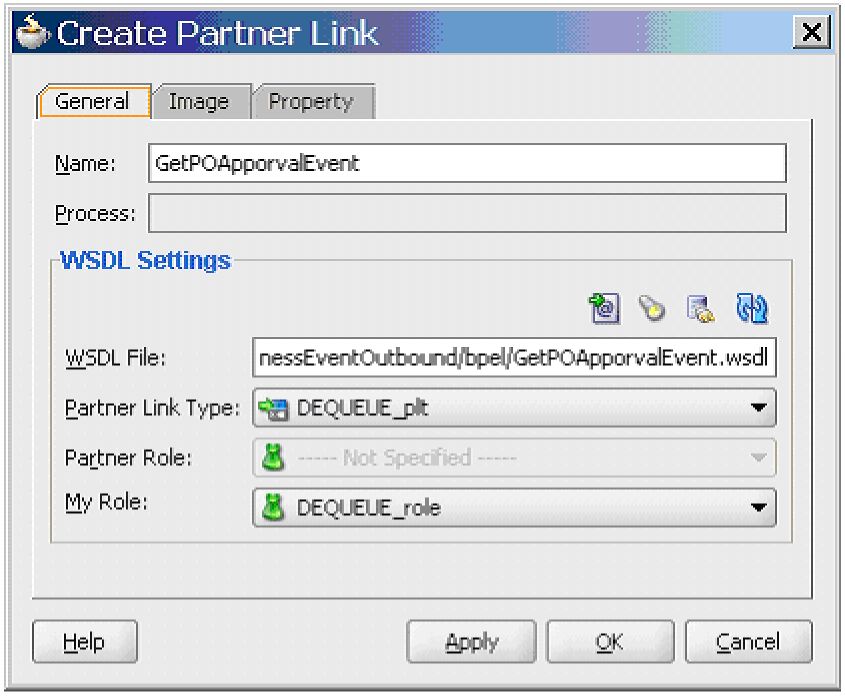
Schema Location: APPS_WF_EVENT_T.xsd

Schema Element: WF_EVENT_T

11. The Finish dialog box appears indicating that you have finished defining the business event service. The wizard generates the GetPOApprovalEvent WSDL file corresponding to the **oracle.apps.po.event.xmlpo** business event service.

    The main Create Partner Link dialog box appears, specifying the new WSDL file.

Completing the Partner Link Configuration



Name: GetPOApporvalEvent

Process:

WSDL File: nessEventOutbound/bpel/GetPOApporvalEvent.wsdl

Partner Link Type: DEQUEUE_plt

Partner Role: ----- Not Specified -----

My Role: DEQUEUE_role

12. Click **Apply** and then **OK** to complete the partner link configuration. The partner link is created with the required WSDL settings, and is represented in the BPEL project by a new icon in the border area of the process diagram.

Configuring the Receive Activity

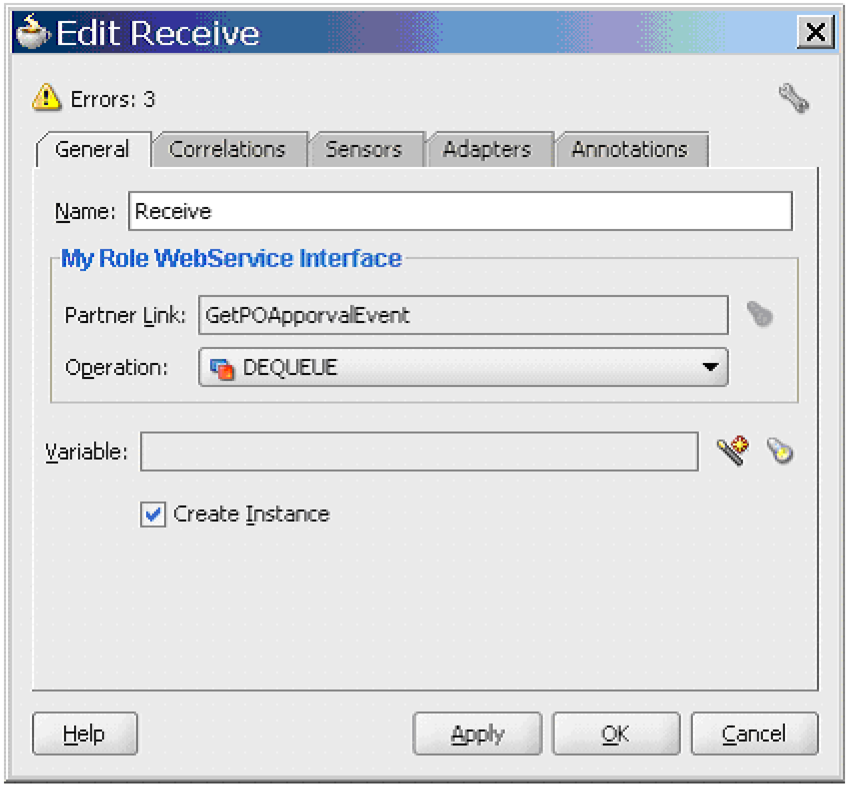
The next task is to configure a Receive activity to receive XML data from the partner link that you configured for the Oracle Applications Adapter service for business events.

To configure the Receive activity:

1. In JDeveloper BPEL Designer, drag and drop the **Receive** activity from the Process Activities of the Component Palette into the process diagram.

2. Link the Receive activity to the GetPOApprovalEvent partner link. The Receive activity will take event data from the partner link. The Edit Receive dialog box appears.

Editing the Receive Activity


Edit Receive

Errors: 3

General | Correlations | Sensors | Adapters | Annotations

Name: Receive

My Role WebService Interface

Partner Link: GetPOApporvalEvent

Operation: DEQUEUE

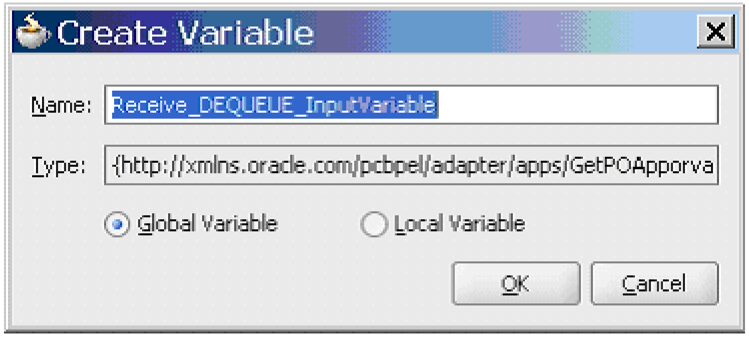
Variable:

[✓ Create Instance]

Help | Apply | OK | Cancel

3. Enter a name for the Receive activity, then click the **Create** icon next to the **Variable** field to create a new variable. The Create Variable dialog box appears.

Creating a Variable


Create Variable

Name: Receive_DEQUEUE_InputVariable

Type: {http://xmlns.oracle.com/pcbpel/adapter/apps/GetPOApporva

[✓ Global Variable] Local Variable

OK | Cancel

4. Select **Global Variable** and enter a name for the variable. You can also accept the default name. Click **OK** to return to the Edit Receive dialog box.

5. Select **Create Instance**.

Click **Apply** and **OK** to finish configuring the Receive activity.

Adding a Partner Link for the File Adapter

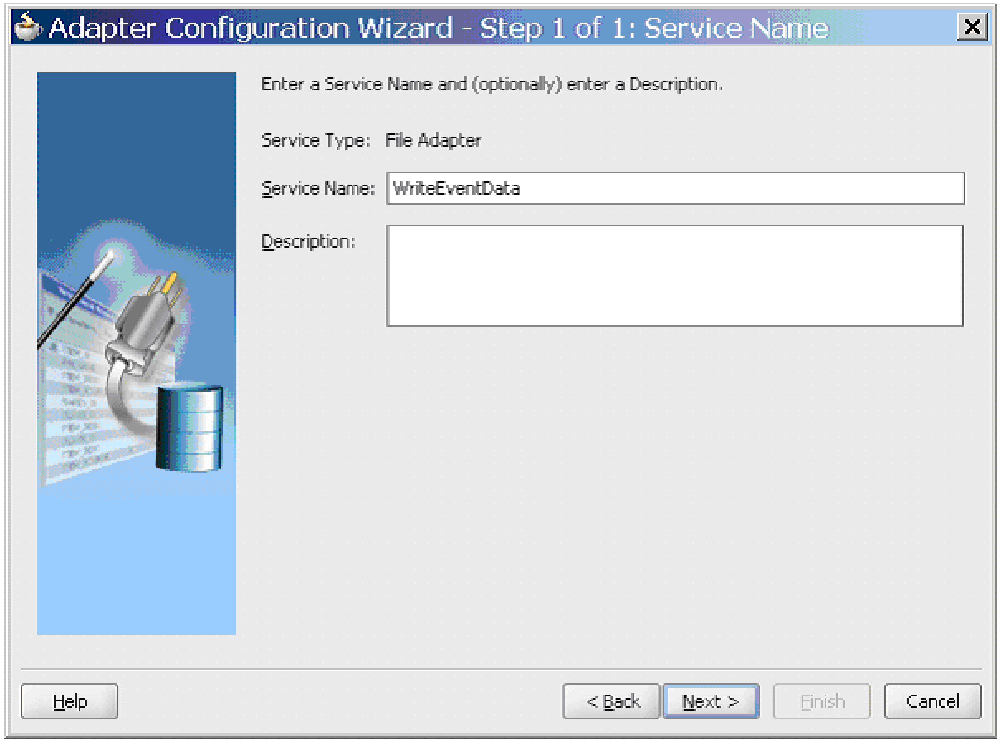
If you need the payload of an outbound business event to be written to an XML file, a partner link for the File Adapter may be added.

To add a partner link for the file adapter:

1. In JDeveloper BPEL Designer, drag and drop the **File Adapter** service from the **Service** section of the Component Palette into the Partner Link area of the process diagram. The Adapter Configuration Wizard appears.

2. Click **Next**. The Service Name dialog box appears.

Specifying the Service Name


Adapter Configuration Wizard - Step 1 of 1: Service Name

Enter a Service Name and (optionally) enter a Description.

Service Type: File Adapter

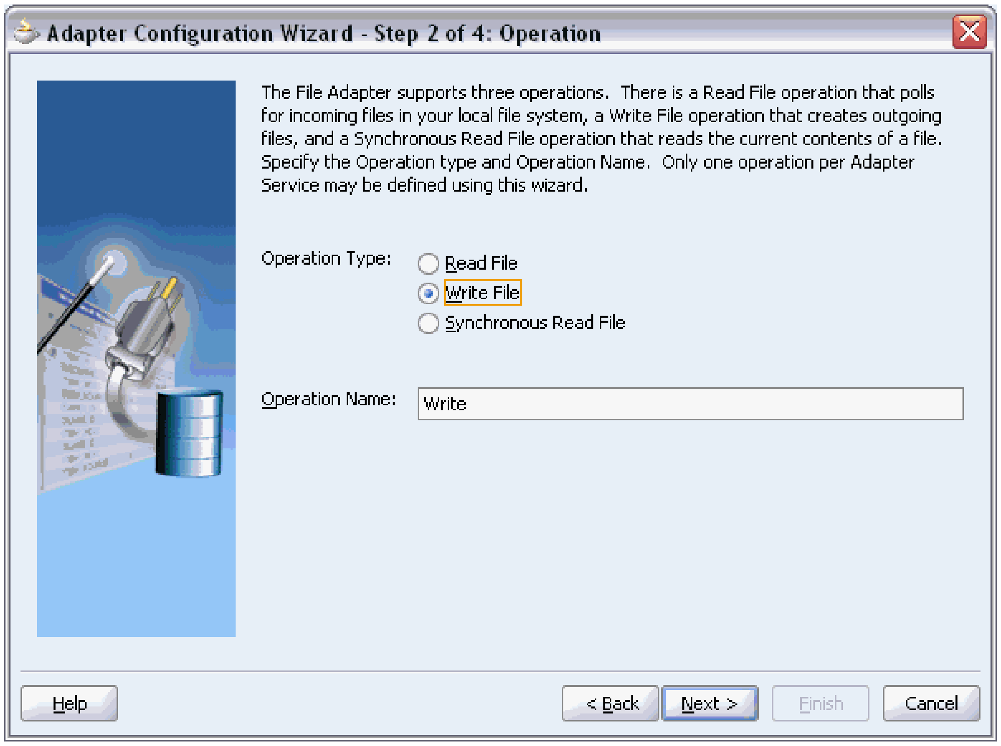
Service Name: WriteEventData

Description:

3. Enter a name for the File Adapter service, such as `WriteEventData`. You can also add an optional description of the service.

4. Click **Next** and the Operation dialog box appears.

Specifying the Operation



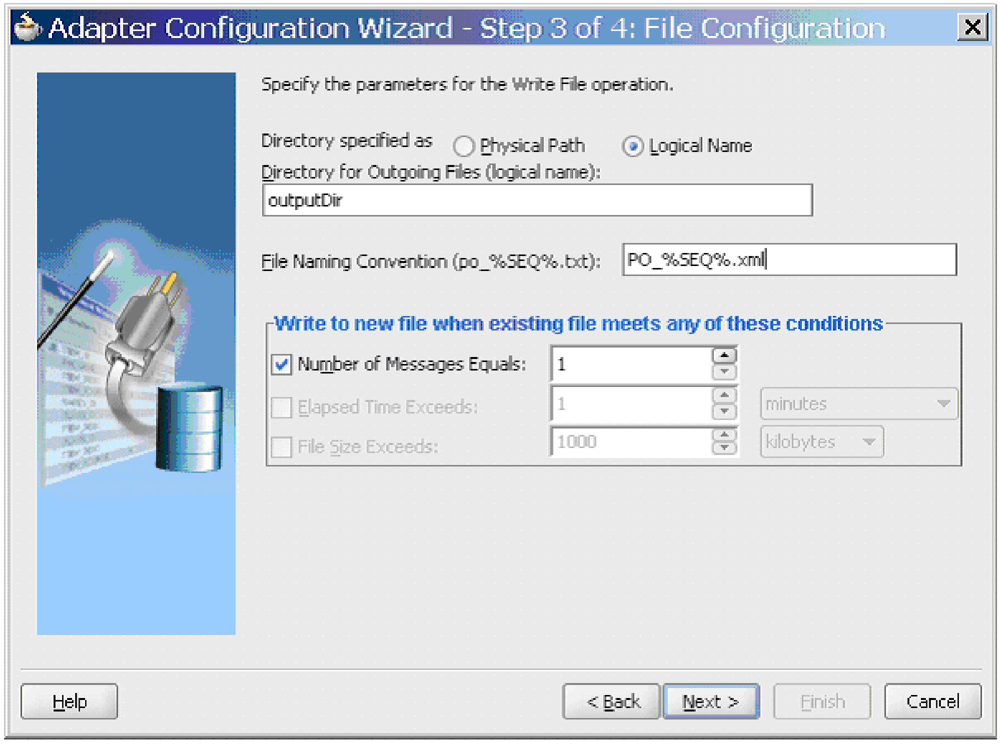
The File Adapter supports three operations. There is a Read File operation that polls for incoming files in your local file system, a Write File operation that creates outgoing files, and a Synchronous Read File operation that reads the current contents of a file. Specify the Operation type and Operation Name. Only one operation per Adapter Service may be defined using this wizard.

Operation Type: Read File [✓ Write File] Synchronous Read File

Operation Name: Write

5. Select **Write File** as the operation type. This automatically populates the **Operation Name** field. Click **Next** to access the File Configuration dialog box.

Configuring the Output File


Adapter Configuration Wizard - Step 3 of 4: File Configuration

Specify the parameters for the Write File operation.

Directory specified as ( ) Physical Path [✓ Logical Name]

Directory for Outgoing Files (logical name): outputDir

File Naming Convention (po_%SEQ%.txt): PO_%SEQ%.xml

Write to new file when existing file meets any of these conditions

- [✓] Number of Messages Equals: 1
- [ ] Elapsed Time Exceeds: 1 minutes
- [ ] File Size Exceeds: 1000 kilobytes

6. For **Directory specified as**, select **Logical Name**.

   Enter `outputDir` as the **Directory for Outgoing Files**, and specify a naming convention for the output file, such as `PO_%SEQ%.xml`.

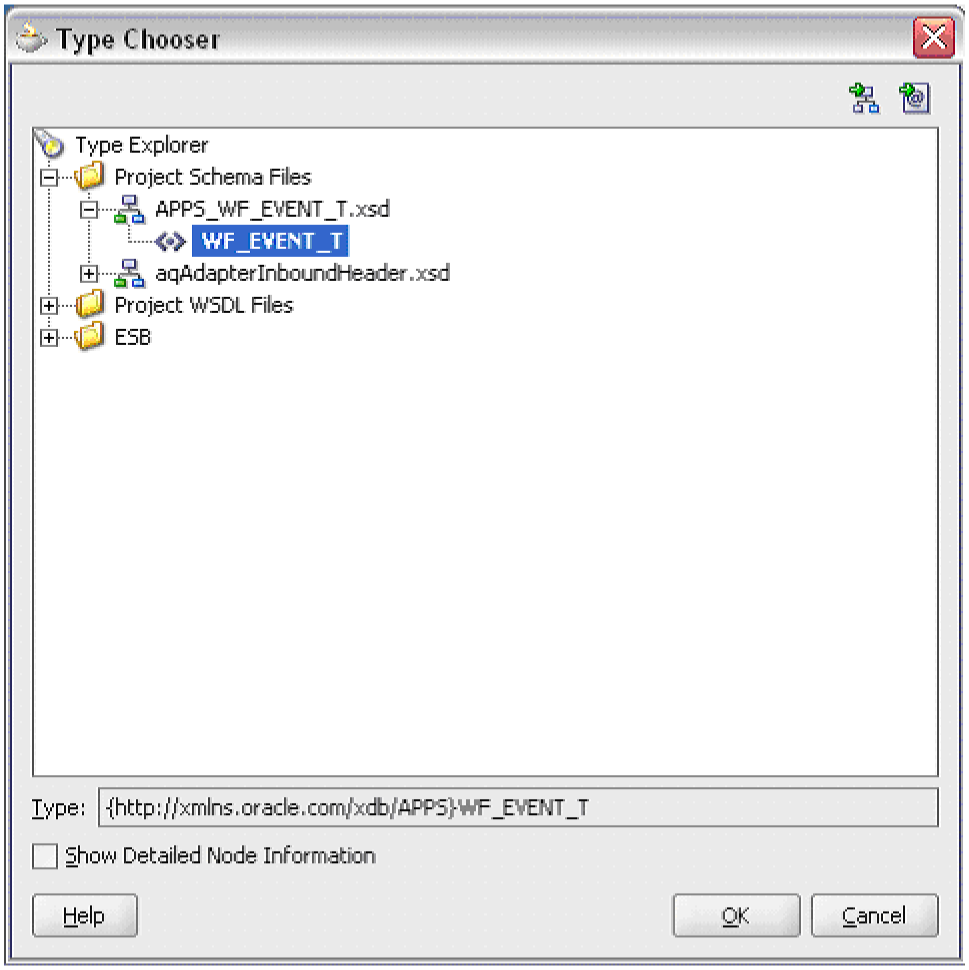
   > **Tip:** When you type a percent sign (`%`), you can choose from a list of date variables or a sequence number variable (`SEQ`) as part of the filename.

   Confirm the default write condition: **Number of Messages Equals 1**.

7. Click **Next** and the Messages dialog box appears. For the output file to be written, you must provide a schema.

8. Click **Browse** to access the Type Chooser.

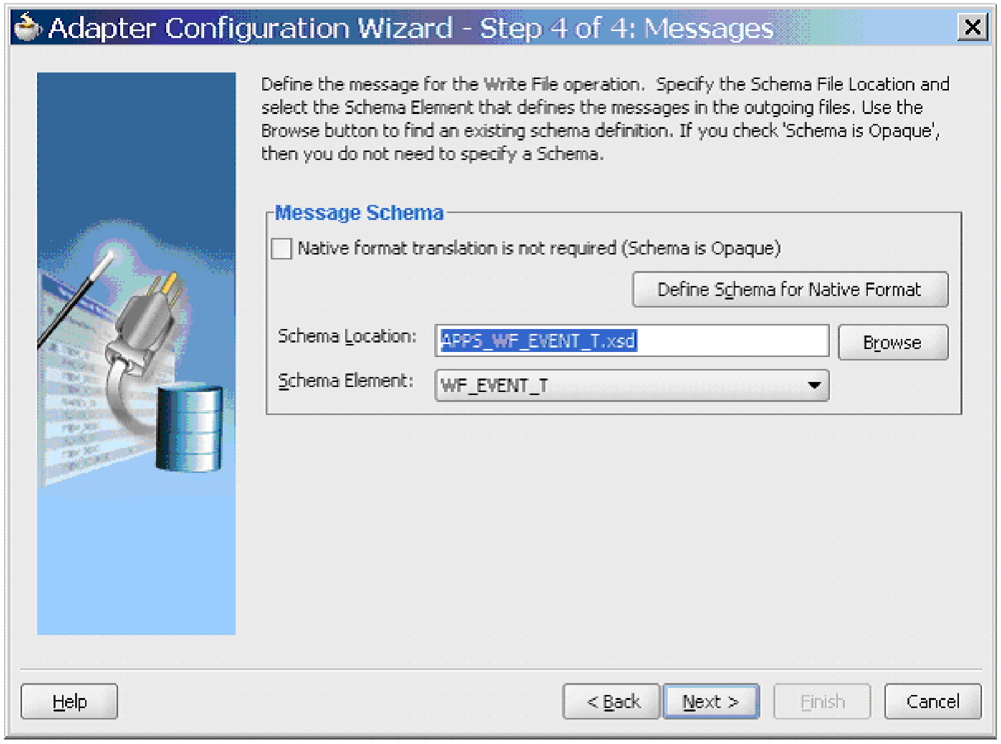
Choosing the Message Schema



Type: {http://xmlns.oracle.com/xdb/APPS}WF_EVENT_T

9. In the Type Chooser, navigate to **Project Schema Files > APPS_WF_EVENT_T.xsd** and select the **WF_EVENT_T** schema. Click **OK** to return to the Messages dialog box.

Specifying the Message Schema


Adapter Configuration Wizard - Step 4 of 4: Messages

Define the message for the Write File operation. Specify the Schema File Location and select the Schema Element that defines the messages in the outgoing files. Use the Browse button to find an existing schema definition. If you check 'Schema is Opaque', then you do not need to specify a Schema.

Message Schema

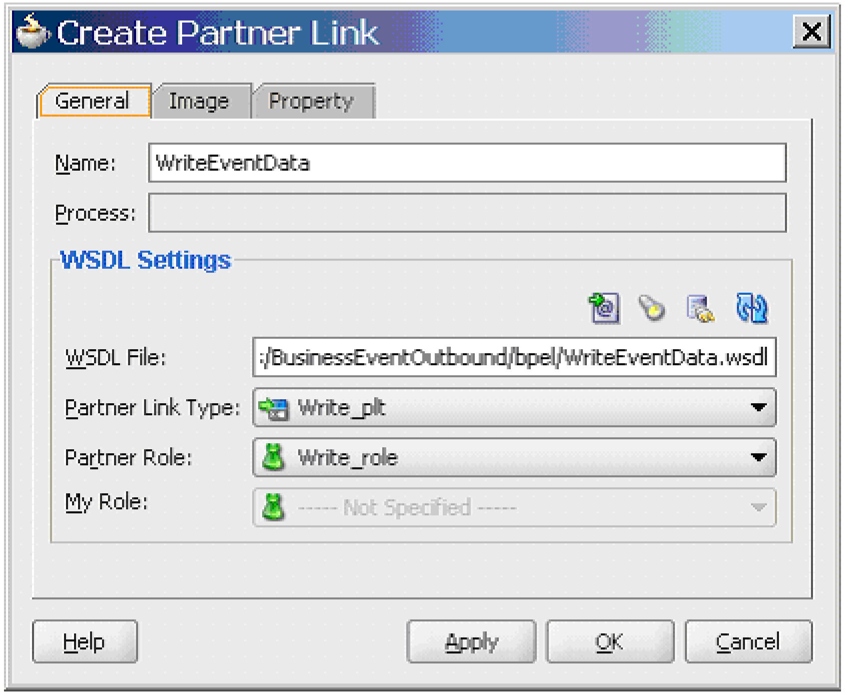
☐ Native format translation is not required (Schema is Opaque)

Schema Location: APPS_WF_EVENT_T.xsd

Schema Element: WF_EVENT_T

10. Click **Next** and then **Finish**. The wizard generates the WSDL file corresponding to the partner link. The main Create Partner Link dialog box appears, specifying the new WSDL file.

Completing the Partner Link Configuration



Name: WriteEventData

Process:

WSDL File: :/BusinessEventOutbound/bpel/WriteEventData.wsdl

Partner Link Type: Write_plt

Partner Role: Write_role

My Role: ----- Not Specified -----

11. Click **Apply** and **OK** to complete the configuration and create the partner link with the required WSDL settings for the File Adapter service.

Configuring the Invoke Activity

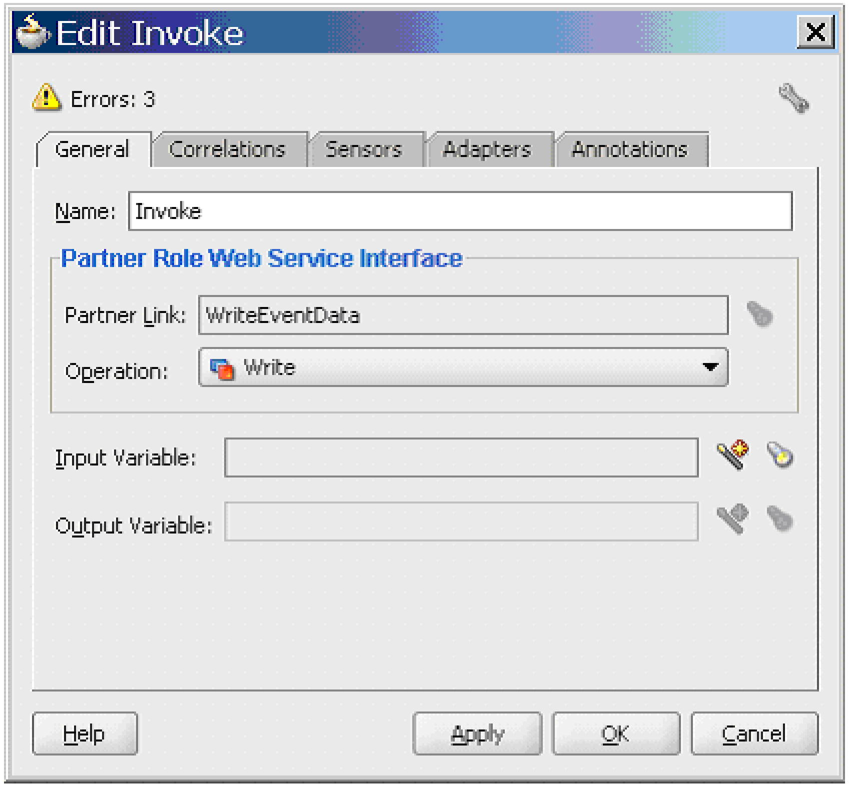
After adding and configuring the partner link, the next task is to configure the Invoke activity to write the business event information to the file.

To configure the Invoke activity:

1. In JDeveloper BPEL Designer, drag and drop the **Invoke** activity from the Component Palette into the process diagram after the **Receive** activity.

2. Link the Invoke activity to the WriteEventData File Adapter service. The Invoke activity will send event data to the partner link. The Edit Invoke dialog box appears.

Editing the Invoke Activity


Edit Invoke

Errors: 3

Name: Invoke

Partner Role Web Service Interface

Partner Link: WriteEventData

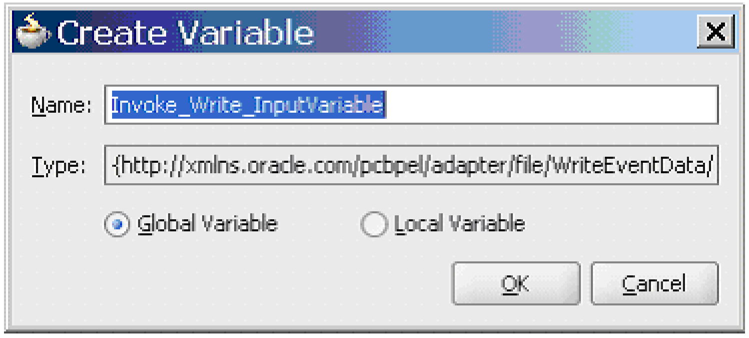
Operation: Write

Input Variable:

Output Variable:

3. Enter a name for the Invoke activity and click the **Create** icon next to the **Input Variable** field to create a new variable. The Create Variable dialog box appears.

Creating a Variable


Create Variable

Name: Invoke_Write_InputVariable

Type: {http://xmlns.oracle.com/pcbpel/adapter/file/WriteEventData/

[✓ Global Variable] Local Variable

4. Select **Global Variable** and enter a name for the variable. You can also accept the default name. Click **OK** in the Create Variable dialog box.

   Click **Apply** and **OK** in the Edit Invoke dialog box to finish configuring the Invoke activity.

Configuring the Assign Activity

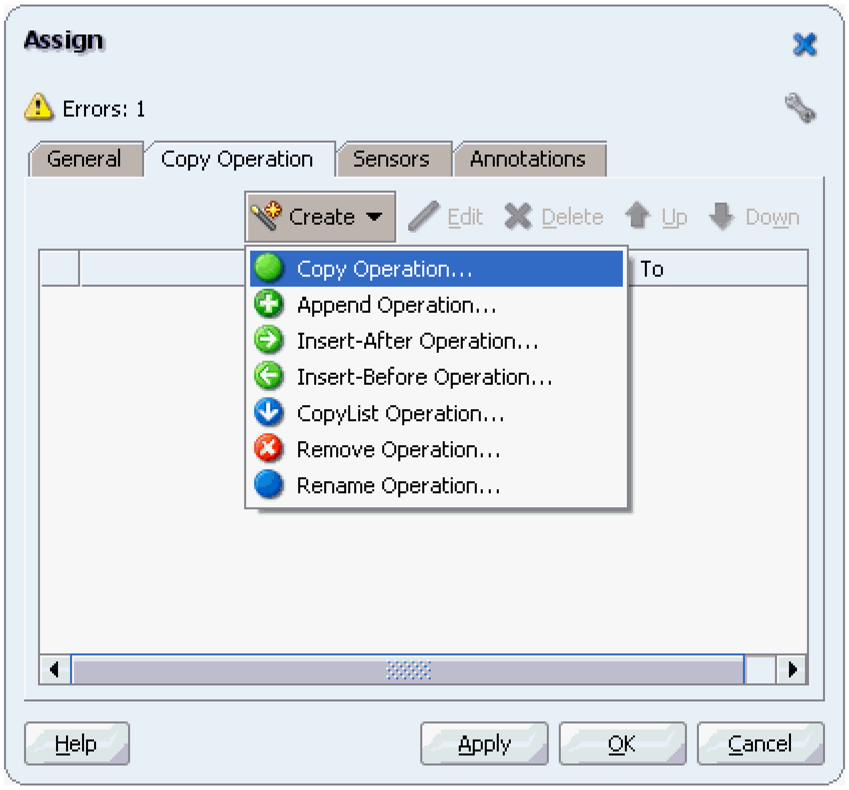
The next task is to add an Assign activity to the process map. This activity is configured to assign the variable values to invoke activity.

To configure the Assign activity:

1. In JDeveloper BPEL Designer, drag and drop the **Assign** activity from the Component Palette into the Activity box of the process diagram, between the **Receive** activity and the **Invoke** activity.

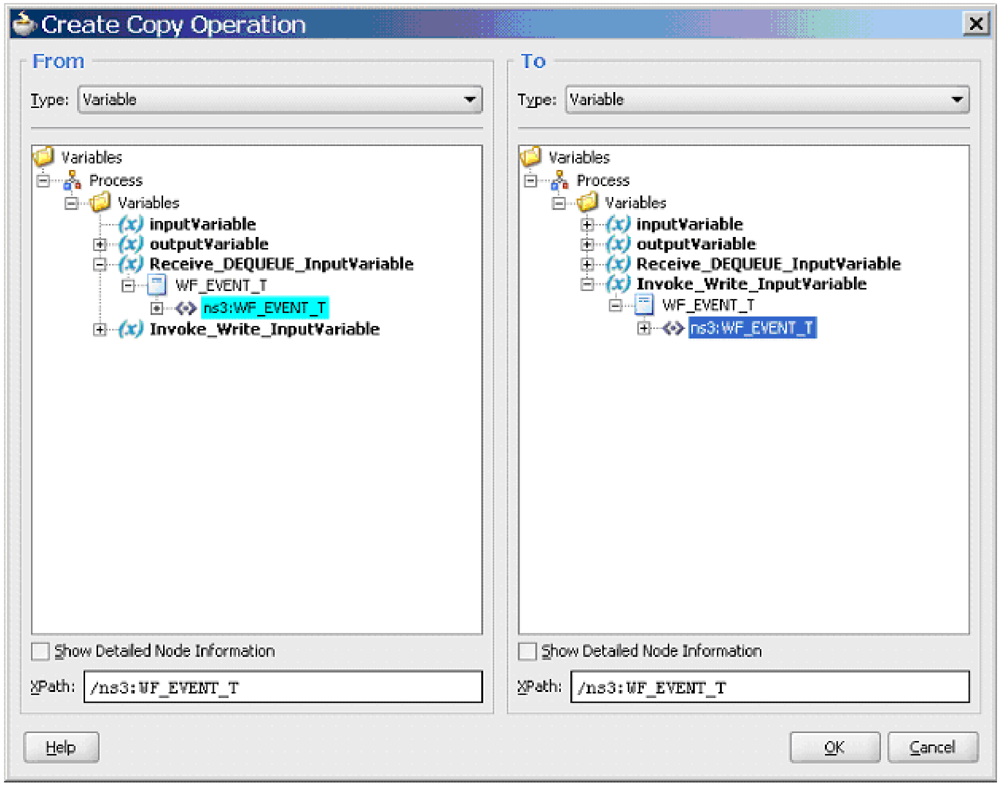
2. Double-click the **Assign** activity to access the Edit Assign dialog box.

Specifying a Copy Operation Action



3. On the Copy Operation tab, click **Create** and select **Copy Operation** from the menu. The Create Copy Operation window appears.

Defining the Copy Operation



4. In the From navigation tree, select type Variable. Navigate to **Variable** > **Process** > **Variables** > **Receive_DEQUEUE_InputVariable** > **WF_EVENT_T** and select **ns3:WF_EVENT_T**. The XPath field should contain your selected entry.

5. In the To navigation tree, select type Variable. Navigate to **Variable** > **Process** > **Variables** > **Invoke_Write_InputVariable** > **WF_EVENT_T** and select **ns3:WF_EVENT_T**. The XPath field should contain your selected entry.

6. Click **OK**.

   Click **Apply** and **OK** in the Edit Assign dialog box to complete the configuration of the Assign activity.

Run-Time Tasks for Outbound Business Events

After designing the BPEL process, you can compile, deploy and test it.

1. Deploy the BPEL process, page 5-28.

2. Test the BPEL process, page 5-29.

Deploying the BPEL Process

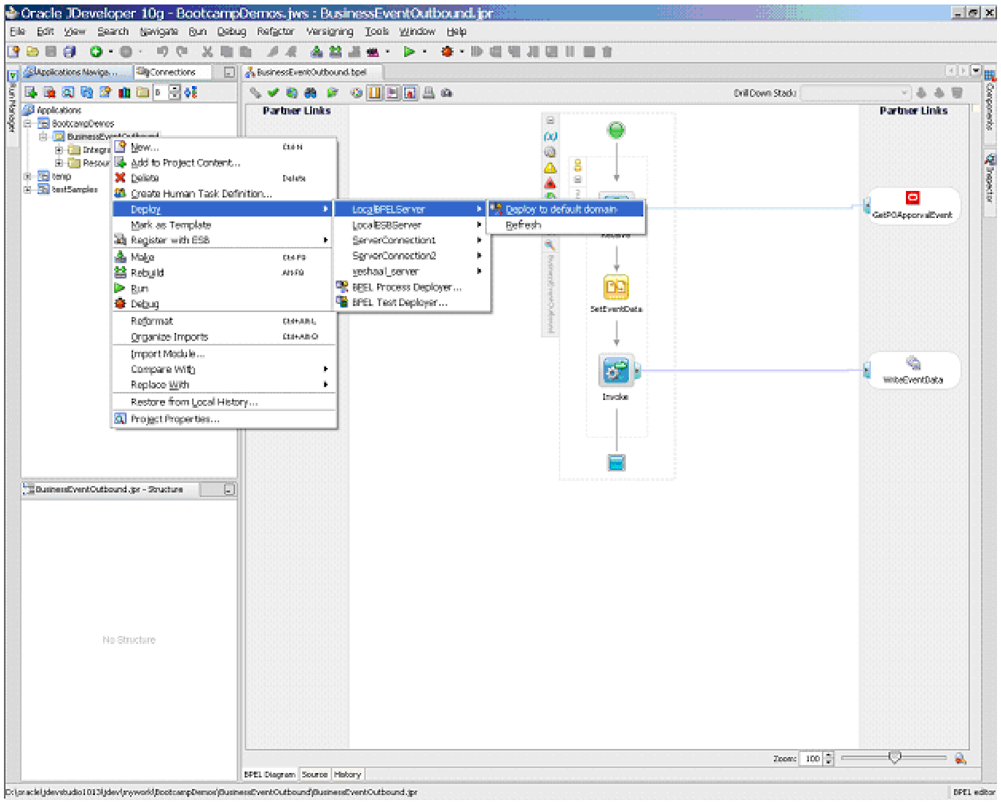
You need to deploy the BPEL process before you can run it. The BPEL process is first compiled and then deployed to the BPEL server.

To deploy the BPEL process:

1. In the Applications Navigator of JDeveloper BPEL Designer, select the **BusinessEventOutbound** project.

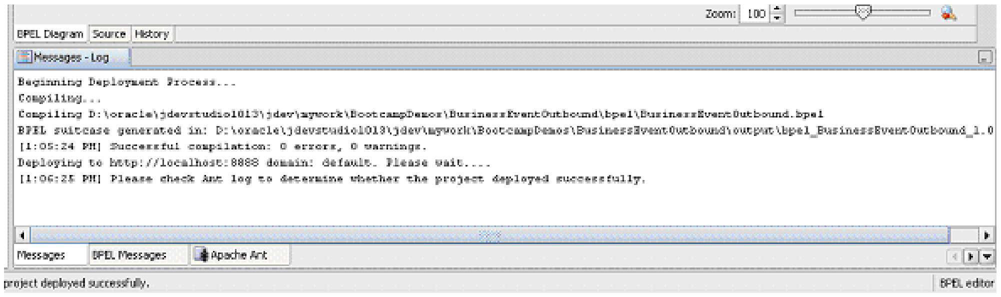
2. Right-click the project and select **Deploy > [Server Connection] > Deploy to Default Domain** from the menu.

Deploying the BPEL Process



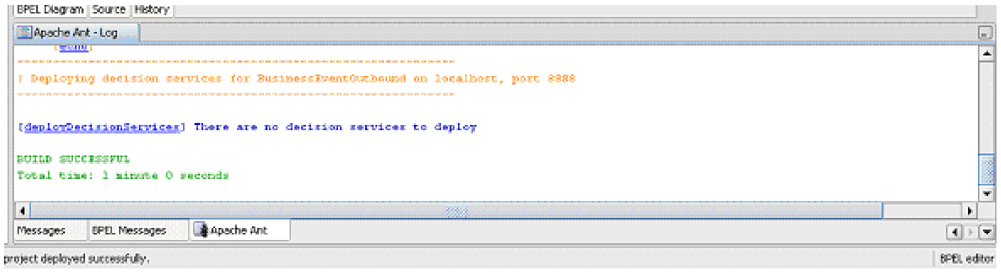
3. The BPEL project is compiled and successfully deployed.

Compilation and Deployment Message Logs



```
Beginning Deployment Process...
Compiling...
Compiling D:\oracle\jdevstudio1013\jdev\mywork\BootcampDemos\BusinessEventOutbound\bpel\BusinessEventOutbound.bpel
BPEL suitcase generated in: D:\oracle\jdevstudio1013\jdev\mywork\BootcampDemos\BusinessEventOutbound\output\bpel_BusinessEventOutbound_1.0
[1:05:24 PM] Successful compilation: 0 errors, 0 warnings.
Deploying to http://localhost:8888 domain: default. Please wait....
[1:06:25 PM] Please check Ant log to determine whether the project deployed successfully.
```

project deployed successfully.



```
--------------------------------------------------------------
| Deploying decision services for BusinessEventOutbound on localhost, port 8888
--------------------------------------------------------------

[deployDecisionServices] There are no decision services to deploy

BUILD SUCCESSFUL
Total time: 1 minute 0 seconds
```

project deployed successfully.

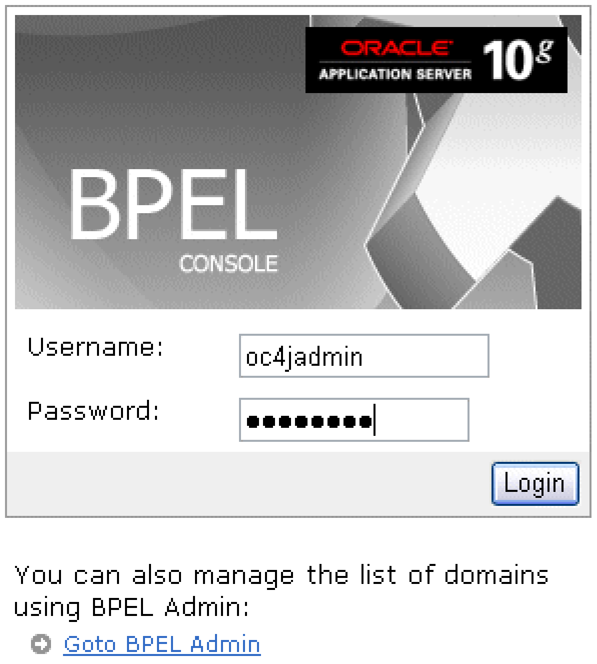
Testing the BPEL Process

Once the BPEL process is deployed, it can be seen in the BPEL Console. You can manage and monitor the process from the BPEL Console. You can also test the process and the integration interface by manually initiating the process.

To test the BPEL process:

1. Navigate to Oracle Application Server 10*g* BPEL Console (`http://<soaSuiteServerHostName>:<port>/BPELConsole`).

   The BPEL Console login page appears.



Username: oc4jadmin

Password: ●●●●●●●●●

You can also manage the list of domains using BPEL Admin:

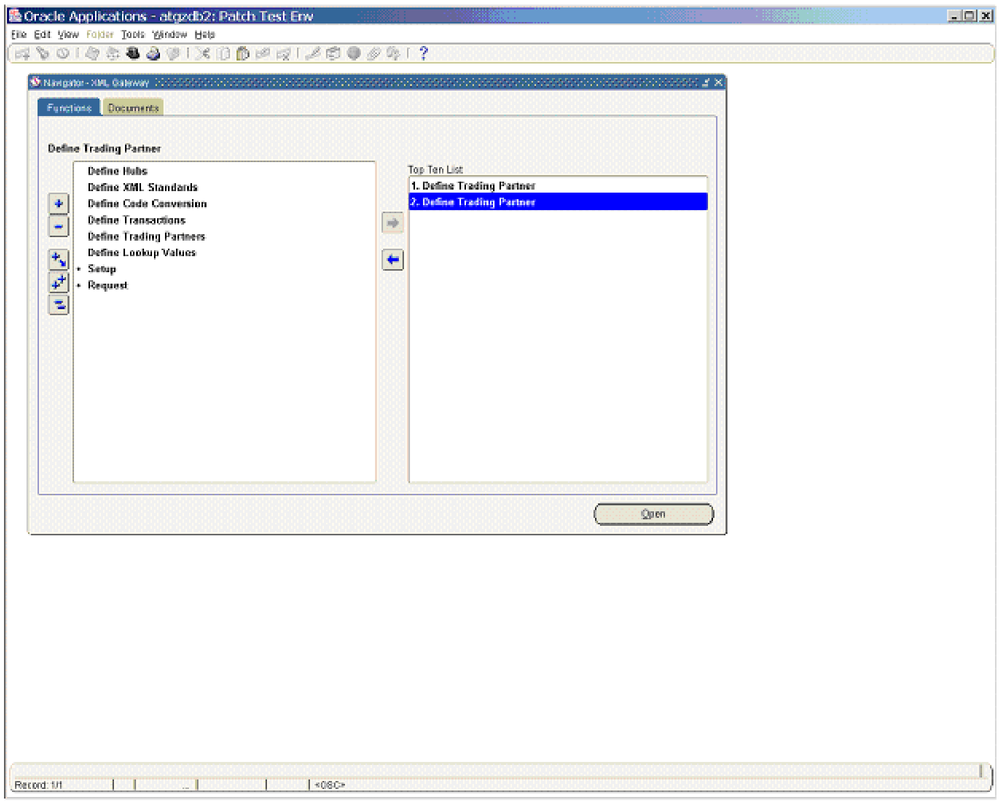
2. Enter the username and password and click **Login**.
3. The Oracle Enterprise Manager 10*g* BPEL Control appears. The list of deployed processes is shown under Deployed BPEL Processes.

   In the BPEL Console, confirm that **BusinessEventOutbound** has been deployed.
4. Log in to Oracle E-Business Suite.

   Select the Purchasing, Vision Operations (USA) responsibility, and then click **Purchase Orders**.

   The Purchase Order form opens.
5. Close the form and select **File > Switch Responsibility**. Select the XML Gateway responsibility.

Selecting XML Gateway Functions


Oracle Applications - atgzdb2: Patch Test Env
File Edit View Folder Tools Window Help
Navigator - XML Gateway
Functions
Documents
Open
Record: 1/1
<OSC>

Define Trading Partner

- Define Hubs
- Define XML Standards
- Define Code Conversion
- Define Transactions
- Define Trading Partners
- Define Lookup Values
- + Setup
- + Request

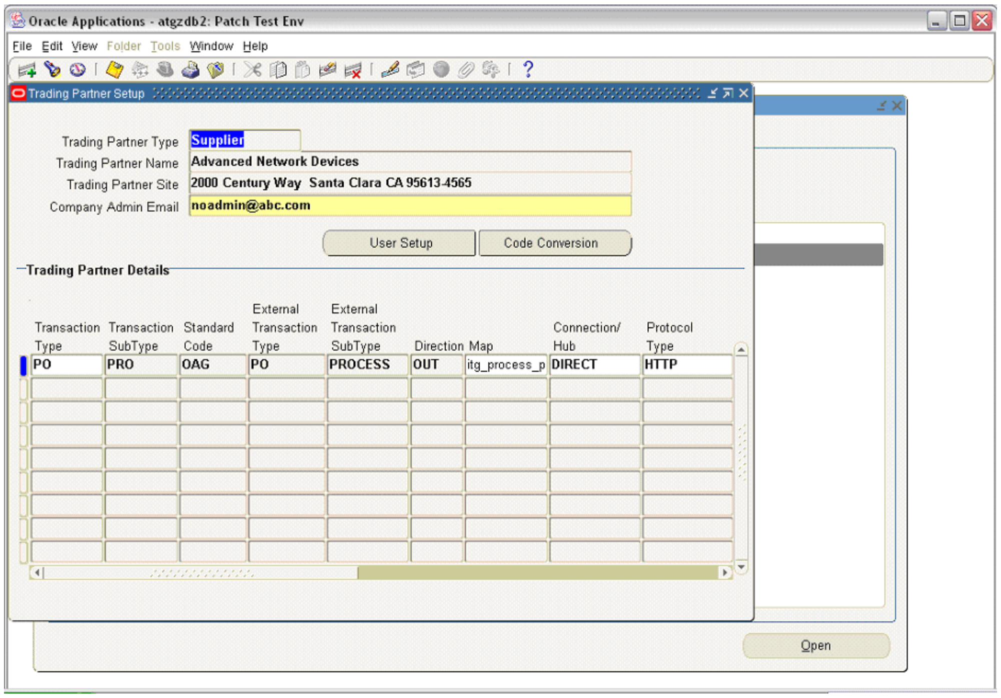
Top Ten List

1. Define Trading Partner
2. Define Trading Partner

6. Select **Define Trading Partner** to access the Trading Partner form.

   Press **<F11>** to access the Trading Partner Setup form.

Setting Up Trading Partners


Oracle Applications - atgzdb2: Patch Test Env
File Edit View Folder Tools Window Help
Trading Partner Setup
User Setup
Code Conversion
Open

Trading Partner Type: Supplier
Trading Partner Name: Advanced Network Devices
Trading Partner Site: 2000 Century Way Santa Clara CA 95613-4565
Company Admin Email: noadmin@abc.com

Trading Partner Details

Transaction Type	Transaction SubType	Standard Code	External Transaction Type	External Transaction SubType	Direction	Map	Connection/ Hub	Protocol Type
PO	PRO	OAG	PO	PROCESS	OUT	itg_process_p	DIRECT	HTTP

7. Enter the header values in the Trading Partner Setup form as follows:

   - **Trading Partner Type** - `Supplier`
   - **Trading Partner Name** - for example, `Advanced Network Devices`
   - **Trading Partner Site** - *trading_partner_site_address*
   - **Company Admin Email** - *valid_email_id*

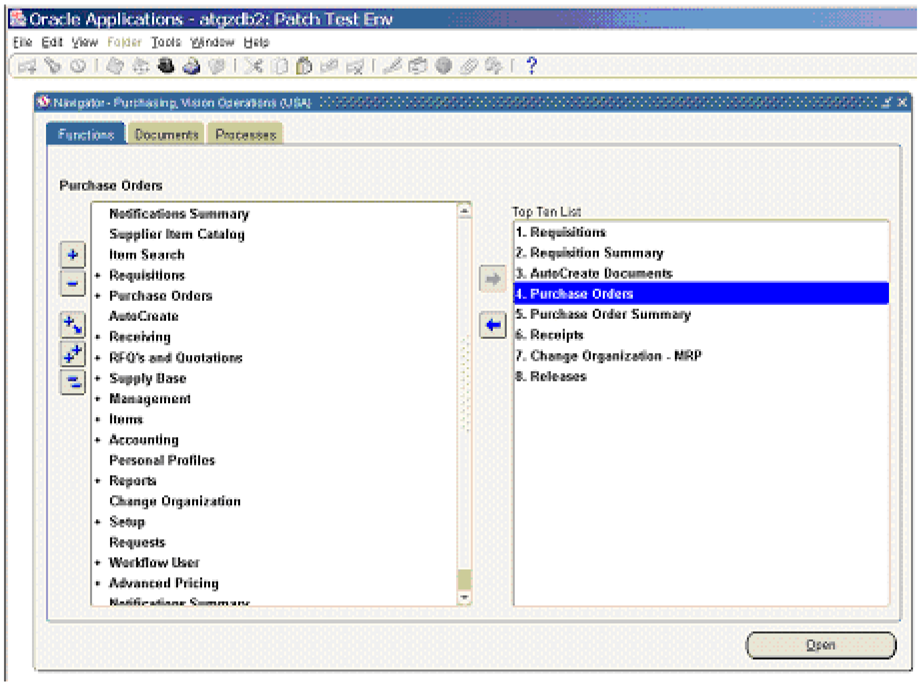
8. Enter a data row with the following values:

   - **Transaction SubType** - `PRO`
   - **Standard Code** - `OAG`
   - **External Transaction Type** - `PO`
   - **External Transaction SubType** - `PROCESS`
   - **Direction** - `OUT`
   - **Map** - `itg_process_po_007_out`

- **Connection / Hub** - `DIRECT`
- **Protocol Type** - `HTTP`
- **Username** - *username*
- **Password** - *password*
- **Protocol Address** - *valid_protocol_address*
- **Source Trading Partner Location Code** - *valid_location_code*

9. Save the trading partner details and switch back to the Purchasing, Vision Operations (USA) responsibility.

Selecting Purchasing Functions



Oracle Applications - atgzdb2: Patch Test Env

File Edit View Folder Tools Window Help

Navigator - Purchasing, Vision Operations (USA)

Functions | Documents | Processes

Purchase Orders

Notifications Summary
Supplier Item Catalog
Item Search
+ Requisitions
+ Purchase Orders
AutoCreate
+ Receiving
+ RFQ's and Quotations
+ Supply Base
+ Management
+ Items
+ Accounting
Personal Profiles
+ Reports
Change Organization
+ Setup
Requests
+ Workflow User
+ Advanced Pricing
Notifications Summary

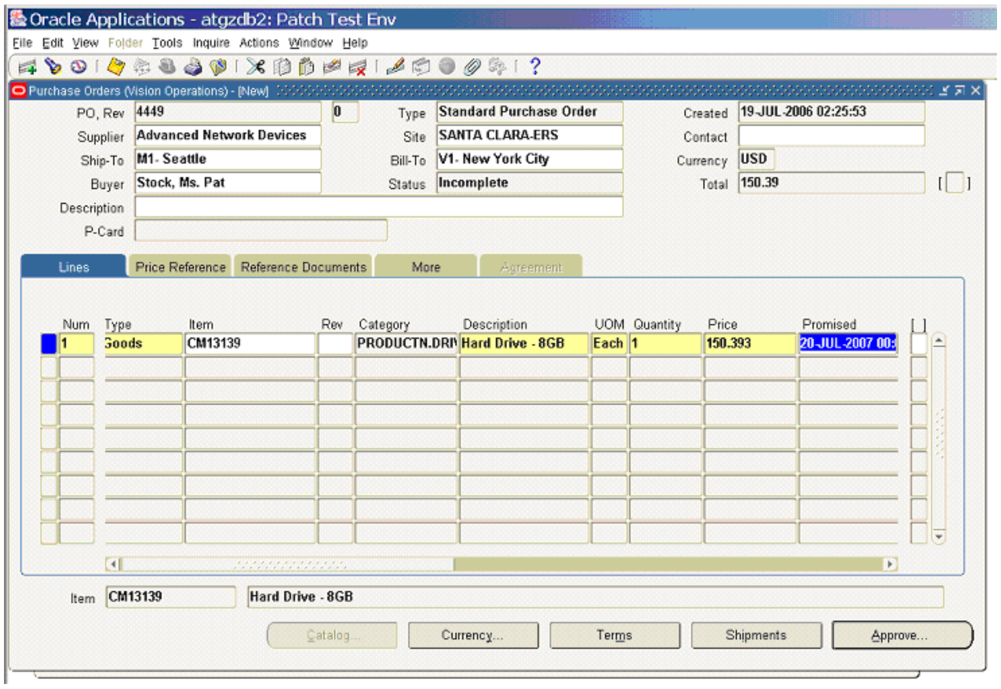
Top Ten List

1. Requisitions
2. Requisition Summary
3. AutoCreate Documents
4. Purchase Orders
5. Purchase Order Summary
6. Receipts
7. Change Organization - MRP
8. Releases

Open

10. Select **Purchase Orders** to access the Purchase Orders form.

Setting Up a Purchase Order


Oracle Applications - atgzdb2: Patch Test Env
File Edit View Folder Tools Inquire Actions Window Help
Purchase Orders (Vision Operations) - [New]
Lines
Price Reference
Reference Documents
More
Agreement
Catalog...
Currency...
Terms
Shipments
Approve...

PO, Rev	4449 | 0	Type	Standard Purchase Order	Created	19-JUL-2006 02:25:53
Supplier	Advanced Network Devices	Site	SANTA CLARA-ERS	Contact	
Ship-To	M1- Seattle	Bill-To	V1- New York City	Currency	USD
Buyer	Stock, Ms. Pat	Status	Incomplete	Total	150.39
Description					
P-Card					

Num	Type	Item	Rev	Category	Description	UOM	Quantity	Price	Promised
1	Goods	CM13139		PRODUCTN.DRI	Hard Drive - 8GB	Each	1	150.393	20-JUL-2007 00:

Item CM13139 Hard Drive - 8GB

11. Create a purchase order with header values reflecting the trading partner you previously defined:

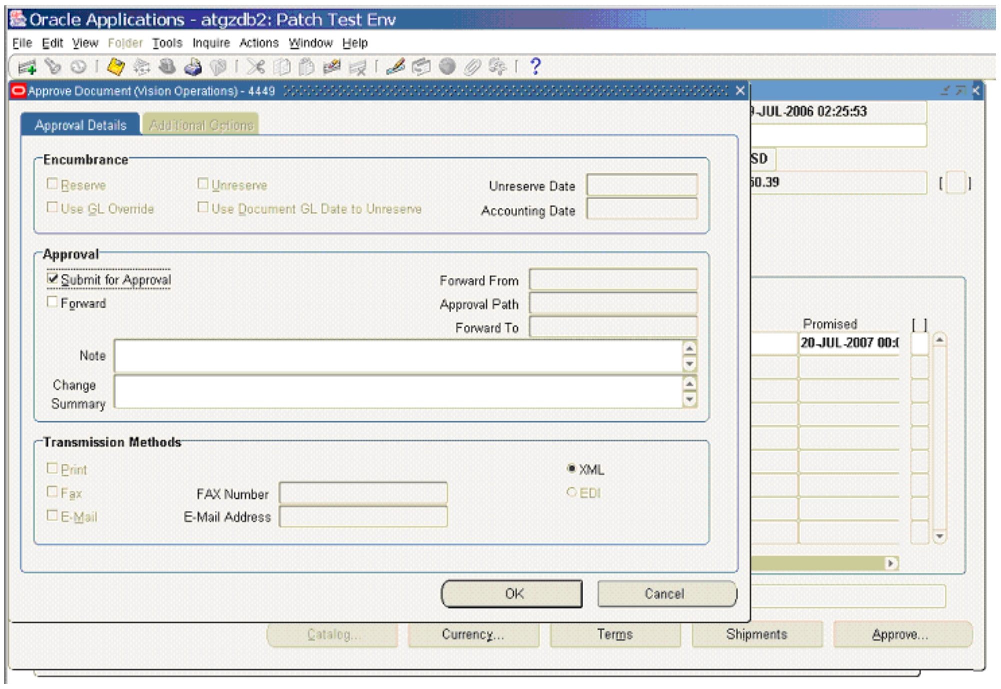
    - **Supplier** - `Advanced Network Devices`
    - **Site** - `SANTA CLARA-ERS`

12. On the Lines tab, enter a data row with the following values:

    - **Item** - `CM13139`
    - **Quantity** - `1`
    - **Promised** - *some_future_date*

13. Save the purchase order. The Approve Document form appears.

Approving the Purchase Order



Submit for Approval: ✓

Transmission Methods: [● XML] ○ EDI

-JUL-2006 02:25:53

SD

60.39

20-JUL-2007 00:(

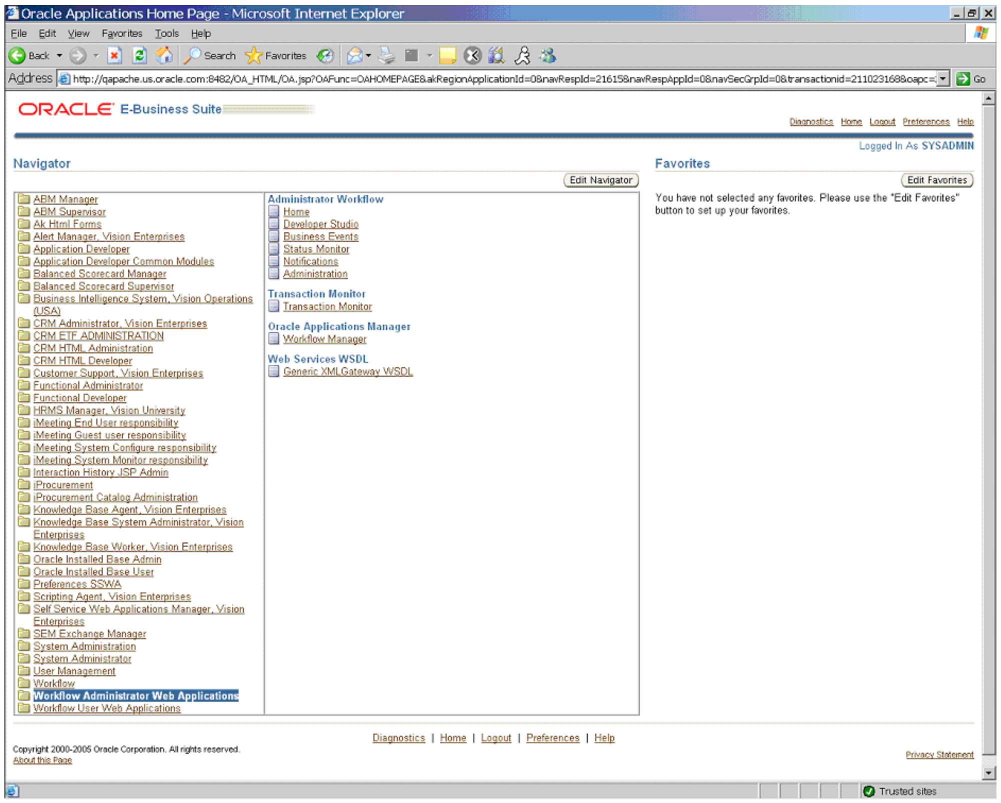
14. Because the trading partner is set up and valid, the transmission method is automatically set to **XML**.

    Click **OK** to approve the purchase order. The Purchase Orders form reappears. The status of the purchase order is now `Approved`. For future reference, note the value of the **PO, Rev** field. Once the purchase order is approved, the business event **oracle.apps.po.event.xmlpo** is raised.

    You should ensure that the **WF_Deferred** agent listener is running on the target database.

15. Log in to Oracle Applications as System Administrator.

Accessing the Oracle Applications Home Page



Oracle Applications Home Page - Microsoft Internet Explorer

Address http://qapache.us.oracle.com:8482/OA_HTML/OA.jsp?OAFunc=OAHOMEPAGE&akRegionApplicationId=0&navRespId=21615&navRespAppId=0&navSecGrpId=0&transactionid=211023168&oapc=

ORACLE E-Business Suite

Diagnostics Home Logout Preferences Help

Logged In As SYSADMIN

Navigator

Edit Navigator

- ABM Manager
- ABM Supervisor
- Ak Html Forms
- Alert Manager, Vision Enterprises
- Application Developer
- Application Developer Common Modules
- Balanced Scorecard Manager
- Balanced Scorecard Supervisor
- Business Intelligence System, Vision Operations (USA)
- CRM Administrator, Vision Enterprises
- CRM ETF ADMINISTRATION
- CRM HTML Administration
- CRM HTML Developer
- Customer Support, Vision Enterprises
- Functional Administrator
- Functional Developer
- HRMS Manager, Vision University
- iMeeting End User responsibility
- iMeeting Guest user responsibility
- iMeeting System Configure responsibility
- iMeeting System Monitor responsibility
- Interaction History JSP Admin
- iProcurement
- iProcurement Catalog Administration
- Knowledge Base Agent, Vision Enterprises
- Knowledge Base System Administrator, Vision Enterprises
- Knowledge Base Worker, Vision Enterprises
- Oracle Installed Base Admin
- Oracle Installed Base User
- Preferences SSWA
- Scripting Agent, Vision Enterprises
- Self Service Web Applications Manager, Vision Enterprises
- SEM Exchange Manager
- System Administration
- System Administrator
- User Management
- Workflow
- Workflow Administrator Web Applications
- Workflow User Web Applications

Administrator Workflow
- Home
- Developer Studio
- Business Events
- Status Monitor
- Notifications
- Administration

Transaction Monitor
- Transaction Monitor

Oracle Applications Manager
- Workflow Manager

Web Services WSDL
- Generic XMLGateway WSDL

Favorites

Edit Favorites

You have not selected any favorites. Please use the "Edit Favorites" button to set up your favorites.

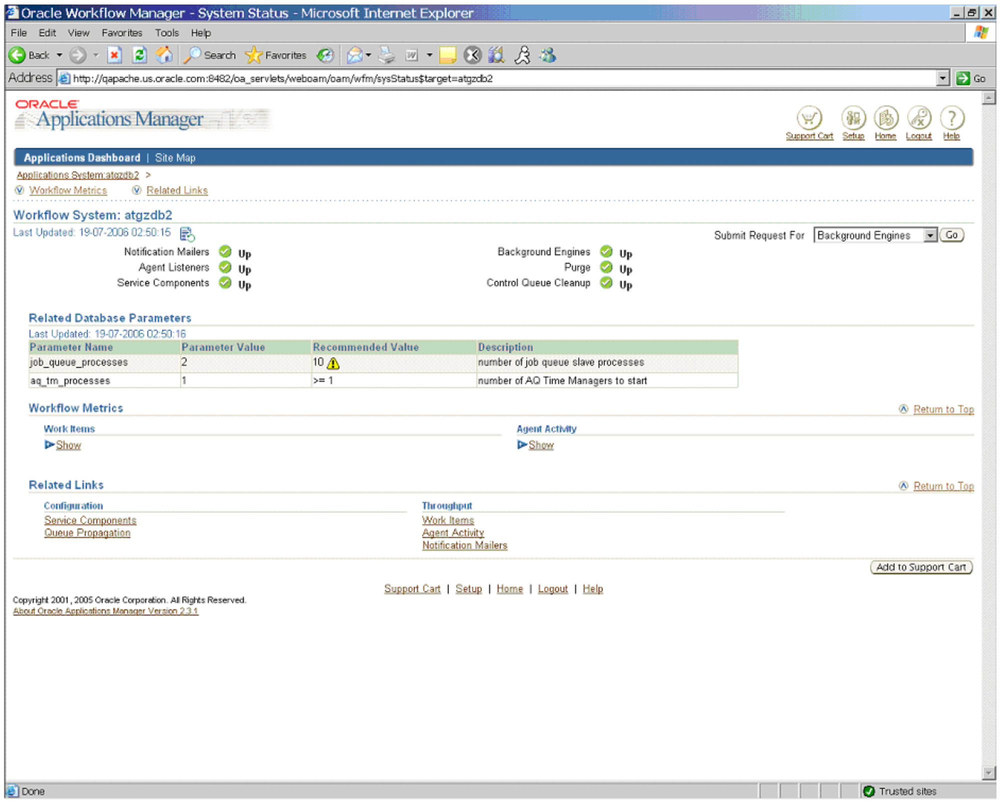
Diagnostics | Home | Logout | Preferences | Help

Copyright 2000-2005 Oracle Corporation. All rights reserved.
About this Page

Privacy Statement

Trusted sites

16. On the Oracle Applications home page, select the Workflow Administrator Web Applications responsibility. Click **Oracle Applications Manager > Workflow Manager** to open the Applications Manager page.



Oracle Workflow Manager - System Status - Microsoft Internet Explorer

Address: http://qapache.us.oracle.com:8482/oa_servlets/weboam/oam/wfm/sysStatus$target=atgzdb2

ORACLE Applications Manager

Applications Dashboard | Site Map

Applications System:atgzdb2 >

Workflow Metrics Related Links

Workflow System: atgzdb2

Last Updated: 19-07-2006 02:50:15

Submit Request For: Background Engines

Notification Mailers	✓	Up	Background Engines	✓	Up
Agent Listeners	✓	Up	Purge	✓	Up
Service Components	✓	Up	Control Queue Cleanup	✓	Up

Related Database Parameters

Last Updated: 19-07-2006 02:50:16

Parameter Name	Parameter Value	Recommended Value	Description
job_queue_processes	2	10 ⚠	number of job queue slave processes
aq_tm_processes	1	>= 1	number of AQ Time Managers to start

Workflow Metrics

Work Items: Show

Agent Activity: Show

Related Links

Configuration: Service Components, Queue Propagation

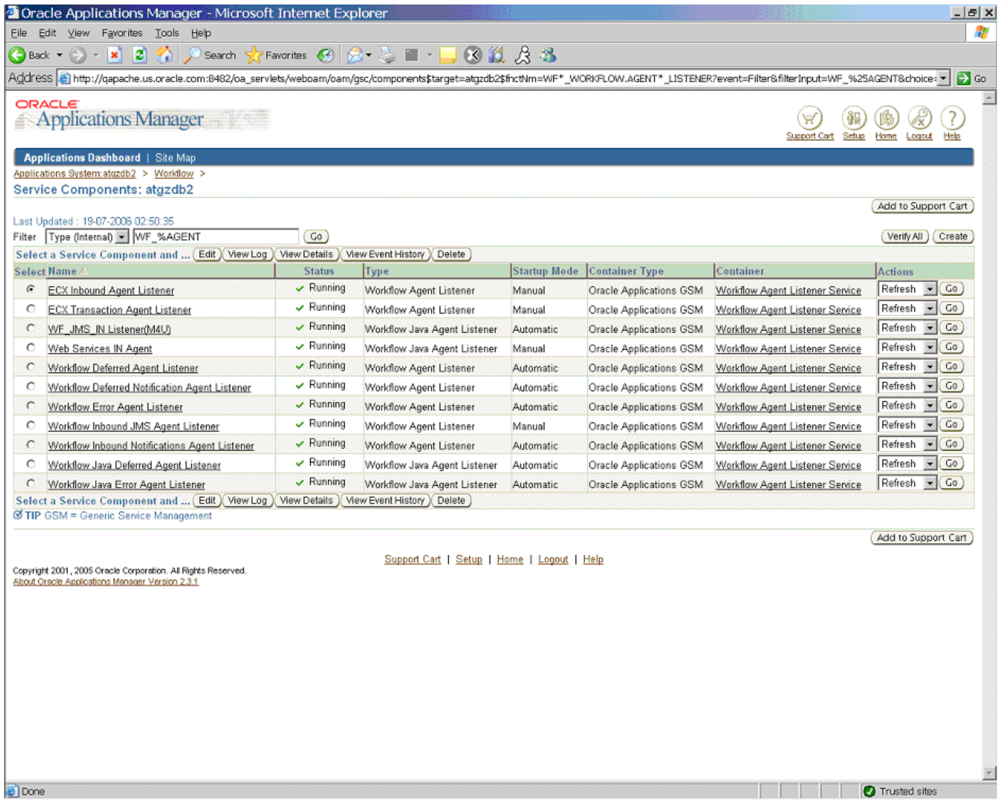
Throughput: Work Items, Agent Activity, Notification Mailers

Copyright 2001, 2005 Oracle Corporation. All Rights Reserved.

About Oracle Applications Manager Version 2.3.1

17. On the Applications Manager page, click the **Agent Listeners** icon. The Service Components page appears containing a list of the installed agent listeners.

Reviewing Agent Listener status



Oracle Applications Manager - Microsoft Internet Explorer

Address http://qapache.us.oracle.com:8482/oa_servlets/weboam/oam/gsc/components$target=atgzdb2$fnctNm=WF*_WORKFLOW.AGENT*_LISTENER?event=Filter&filterInput=WF_%25AGENT&choice=

Applications Dashboard | Site Map

Applications System:atgzdb2 > Workflow >

Service Components: atgzdb2

Last Updated : 19-07-2006 02:50:35

Filter Type (Internal) WF_%AGENT

Select a Service Component and ... Edit View Log View Details View Event History Delete

Select	Name	Status	Type	Startup Mode	Container Type	Container	Actions
●	ECX Inbound Agent Listener	✓ Running	Workflow Agent Listener	Manual	Oracle Applications GSM	Workflow Agent Listener Service	Refresh Go
○	ECX Transaction Agent Listener	✓ Running	Workflow Agent Listener	Manual	Oracle Applications GSM	Workflow Agent Listener Service	Refresh Go
○	WF_JMS_IN Listener(M4U)	✓ Running	Workflow Java Agent Listener	Automatic	Oracle Applications GSM	Workflow Agent Listener Service	Refresh Go
○	Web Services IN Agent	✓ Running	Workflow Java Agent Listener	Manual	Oracle Applications GSM	Workflow Agent Listener Service	Refresh Go
○	Workflow Deferred Agent Listener	✓ Running	Workflow Agent Listener	Automatic	Oracle Applications GSM	Workflow Agent Listener Service	Refresh Go
○	Workflow Deferred Notification Agent Listener	✓ Running	Workflow Agent Listener	Automatic	Oracle Applications GSM	Workflow Agent Listener Service	Refresh Go
○	Workflow Error Agent Listener	✓ Running	Workflow Agent Listener	Automatic	Oracle Applications GSM	Workflow Agent Listener Service	Refresh Go
○	Workflow Inbound JMS Agent Listener	✓ Running	Workflow Agent Listener	Manual	Oracle Applications GSM	Workflow Agent Listener Service	Refresh Go
○	Workflow Inbound Notifications Agent Listener	✓ Running	Workflow Agent Listener	Automatic	Oracle Applications GSM	Workflow Agent Listener Service	Refresh Go
○	Workflow Java Deferred Agent Listener	✓ Running	Workflow Java Agent Listener	Automatic	Oracle Applications GSM	Workflow Agent Listener Service	Refresh Go
○	Workflow Java Error Agent Listener	✓ Running	Workflow Java Agent Listener	Automatic	Oracle Applications GSM	Workflow Agent Listener Service	Refresh Go

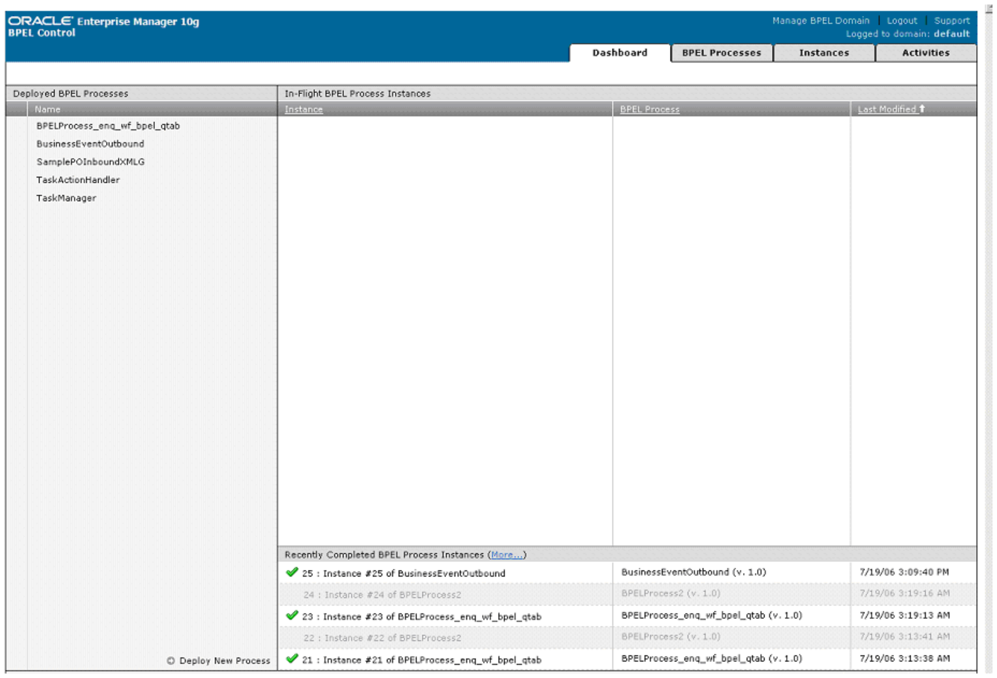
TIP GSM = Generic Service Management

Copyright 2001, 2005 Oracle Corporation. All Rights Reserved.
About Oracle Applications Manager Version 2.3.1

18. Confirm that the **Workflow Deferred Agent Listener** is in `Running` status.

19. Log in to Oracle BPEL Process Manager and return to the BPEL Console.

Confirming Completed BPEL Processes


ORACLE Enterprise Manager 10g
BPEL Control
Manage BPEL Domain | Logout | Support
Logged to domain: default
Dashboard
BPEL Processes
Instances
Activities
Deploy New Process

Deployed BPEL Processes

Name
BPELProcess_enq_wf_bpel_qtab
BusinessEventOutbound
SamplePOInboundXMLG
TaskActionHandler
TaskManager

In-Flight BPEL Process Instances

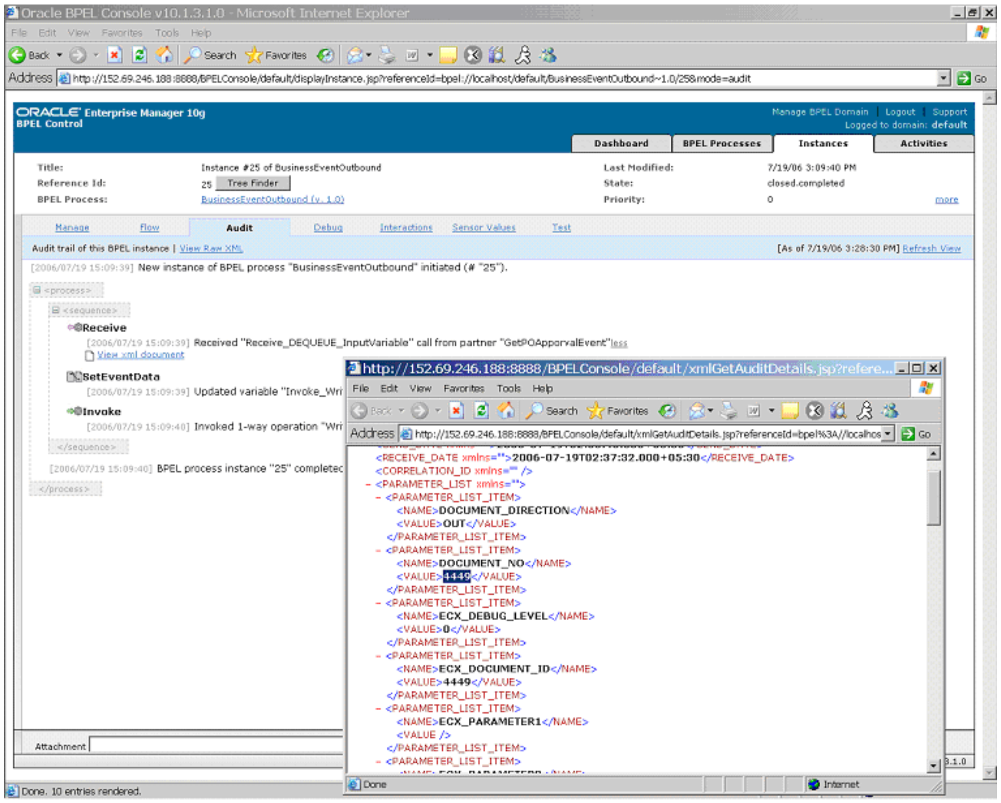
Instance	BPEL Process	Last Modified

Recently Completed BPEL Process Instances (More...)

✓ 25 : Instance #25 of BusinessEventOutbound	BusinessEventOutbound (v. 1.0)	7/19/06 3:09:40 PM
24 : Instance #24 of BPELProcess2	BPELProcess2 (v. 1.0)	7/19/06 3:19:16 AM
✓ 23 : Instance #23 of BPELProcess_enq_wf_bpel_qtab	BPELProcess_enq_wf_bpel_qtab (v. 1.0)	7/19/06 3:19:13 AM
22 : Instance #22 of BPELProcess2	BPELProcess2 (v. 1.0)	7/19/06 3:13:41 AM
✓ 21 : Instance #21 of BPELProcess_enq_wf_bpel_qtab	BPELProcess_enq_wf_bpel_qtab (v. 1.0)	7/19/06 3:13:38 AM

20. Confirm that the **BusinessEventOutbound** process has completed. Select the instance which opens up in the Instance tab. Select the Audit link.

Auditing the BusinessEventOutbound Instance Receive Activity



Oracle BPEL Console v10.1.3.1.0 - Microsoft Internet Explorer

Address http://152.69.246.188:8888/BPELConsole/default/displayInstance.jsp?referenceId=bpel://localhost/default/BusinessEventOutbound~1.0/25&mode=audit

Title:	Instance #25 of BusinessEventOutbound	Last Modified:	7/19/06 3:09:40 PM
Reference Id:	25	State:	closed.completed
BPEL Process:	BusinessEventOutbound (v. 1.0)	Priority:	0

Audit trail of this BPEL instance | View Raw XML [As of 7/19/06 3:28:30 PM] Refresh View

```
[2006/07/19 15:09:39] New instance of BPEL process "BusinessEventOutbound" initiated (# "25").
<process>
  <sequence>
    Receive
      [2006/07/19 15:09:39] Received "Receive_DEQUEUE_InputVariable" call from partner "GetPOApporvalEvent"less
      View xml document
    SetEventData
      [2006/07/19 15:09:39] Updated variable "Invoke_Wri
    Invoke
      [2006/07/19 15:09:40] Invoked 1-way operation "Wri
  </sequence>
  [2006/07/19 15:09:40] BPEL process instance "25" completec
</process>
```

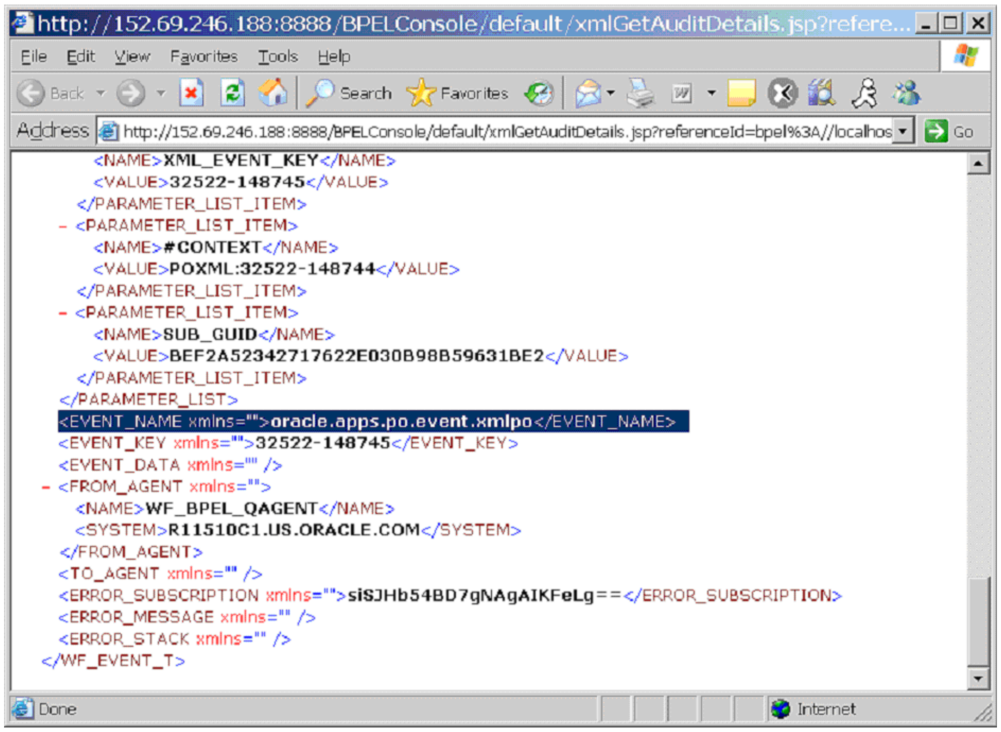
http://152.69.246.188:8888/BPELConsole/default/xmlGetAuditDetails.jsp?refere...

Address http://152.69.246.188:8888/BPELConsole/default/xmlGetAuditDetails.jsp?referenceId=bpel%3A//localhos

```
  <RECEIVE_DATE xmlns="">2006-07-19T02:37:32.000+05:30</RECEIVE_DATE>
  <CORRELATION_ID xmlns="" />
- <PARAMETER_LIST xmlns="">
  - <PARAMETER_LIST_ITEM>
      <NAME>DOCUMENT_DIRECTION</NAME>
      <VALUE>OUT</VALUE>
    </PARAMETER_LIST_ITEM>
  - <PARAMETER_LIST_ITEM>
      <NAME>DOCUMENT_NO</NAME>
      <VALUE>4449</VALUE>
    </PARAMETER_LIST_ITEM>
  - <PARAMETER_LIST_ITEM>
      <NAME>ECX_DEBUG_LEVEL</NAME>
      <VALUE>0</VALUE>
    </PARAMETER_LIST_ITEM>
  - <PARAMETER_LIST_ITEM>
      <NAME>ECX_DOCUMENT_ID</NAME>
      <VALUE>4449</VALUE>
    </PARAMETER_LIST_ITEM>
  - <PARAMETER_LIST_ITEM>
      <NAME>ECX_PARAMETER1</NAME>
      <VALUE />
    </PARAMETER_LIST_ITEM>
  - <PARAMETER_LIST_ITEM>
```

21. Open the XML file for the **Receive** activity and notice the name of the event raised, `oracle.apps.po.event.xmlpo`.

Examining the Receive Event Name

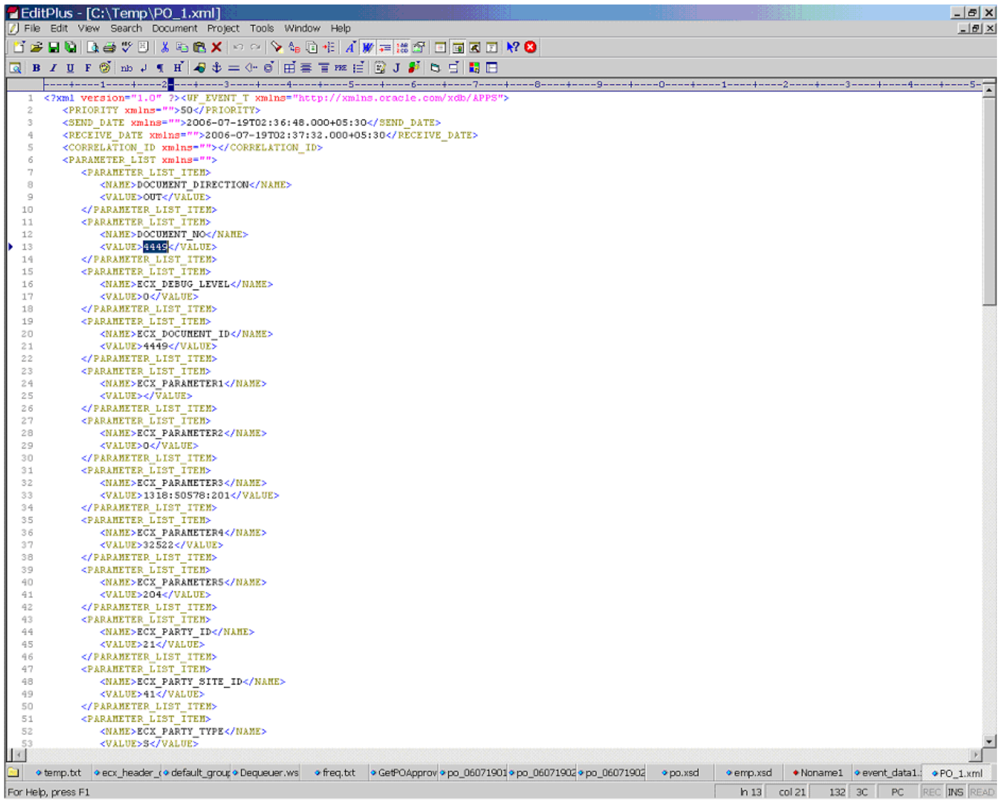

http://152.69.246.188:8888/BPELConsole/default/xmlGetAuditDetails.jsp?refere...
File Edit View Favorites Tools Help
Back Search Favorites
Address http://152.69.246.188:8888/BPELConsole/default/xmlGetAuditDetails.jsp?referenceId=bpel%3A//localhos
Go
Done
Internet

```
        <NAME>XML_EVENT_KEY</NAME>
        <VALUE>32522-148745</VALUE>
      </PARAMETER_LIST_ITEM>
    - <PARAMETER_LIST_ITEM>
        <NAME>#CONTEXT</NAME>
        <VALUE>POXML:32522-148744</VALUE>
      </PARAMETER_LIST_ITEM>
    - <PARAMETER_LIST_ITEM>
        <NAME>SUB_GUID</NAME>
        <VALUE>BEF2A52342717622E030B98B59631BE2</VALUE>
      </PARAMETER_LIST_ITEM>
    </PARAMETER_LIST>
    <EVENT_NAME xmlns="">oracle.apps.po.event.xmlpo</EVENT_NAME>
    <EVENT_KEY xmlns="">32522-148745</EVENT_KEY>
    <EVENT_DATA xmlns="" />
  - <FROM_AGENT xmlns="">
      <NAME>WF_BPEL_QAGENT</NAME>
      <SYSTEM>R11510C1.US.ORACLE.COM</SYSTEM>
    </FROM_AGENT>
    <TO_AGENT xmlns="" />
    <ERROR_SUBSCRIPTION xmlns="">siSJHb54BD7gNAgAIKFeLg==</ERROR_SUBSCRIPTION>
    <ERROR_MESSAGE xmlns="" />
    <ERROR_STACK xmlns="" />
  </WF_EVENT_T>
```

22. Examine the **Assign** and **Invoke** activities as well for the event raised and document number.

23. Go to the directory you specified for the write operation, such as **outputDir** (typically `c:\temp`). Open the output file (for example, `PO_1.xml`) and confirm that the order number is the same as that of the approved purchase order.

Confirming the Output Order Number



```
<?xml version="1.0" ?><WF_EVENT_T xmlns="http://xmlns.oracle.com/xdb/APPS">
   <PRIORITY xmlns="">50</PRIORITY>
   <SEND_DATE xmlns="">2006-07-19T02:36:48.000+05:30</SEND_DATE>
   <RECEIVE_DATE xmlns="">2006-07-19T02:37:32.000+05:30</RECEIVE_DATE>
   <CORRELATION_ID xmlns=""></CORRELATION_ID>
   <PARAMETER_LIST xmlns="">
      <PARAMETER_LIST_ITEM>
         <NAME>DOCUMENT_DIRECTION</NAME>
         <VALUE>OUT</VALUE>
      </PARAMETER_LIST_ITEM>
      <PARAMETER_LIST_ITEM>
         <NAME>DOCUMENT_NO</NAME>
         <VALUE>4449</VALUE>
      </PARAMETER_LIST_ITEM>
      <PARAMETER_LIST_ITEM>
         <NAME>ECX_DEBUG_LEVEL</NAME>
         <VALUE>0</VALUE>
      </PARAMETER_LIST_ITEM>
      <PARAMETER_LIST_ITEM>
         <NAME>ECX_DOCUMENT_ID</NAME>
         <VALUE>4449</VALUE>
      </PARAMETER_LIST_ITEM>
      <PARAMETER_LIST_ITEM>
         <NAME>ECX_PARAMETER1</NAME>
         <VALUE></VALUE>
      </PARAMETER_LIST_ITEM>
      <PARAMETER_LIST_ITEM>
         <NAME>ECX_PARAMETER2</NAME>
         <VALUE>0</VALUE>
      </PARAMETER_LIST_ITEM>
      <PARAMETER_LIST_ITEM>
         <NAME>ECX_PARAMETER3</NAME>
         <VALUE>1318:50578:201</VALUE>
      </PARAMETER_LIST_ITEM>
      <PARAMETER_LIST_ITEM>
         <NAME>ECX_PARAMETER4</NAME>
         <VALUE>32522</VALUE>
      </PARAMETER_LIST_ITEM>
      <PARAMETER_LIST_ITEM>
         <NAME>ECX_PARAMETER5</NAME>
         <VALUE>204</VALUE>
      </PARAMETER_LIST_ITEM>
      <PARAMETER_LIST_ITEM>
         <NAME>ECX_PARTY_ID</NAME>
         <VALUE>21</VALUE>
      </PARAMETER_LIST_ITEM>
      <PARAMETER_LIST_ITEM>
         <NAME>ECX_PARTY_SITE_ID</NAME>
         <VALUE>41</VALUE>
      </PARAMETER_LIST_ITEM>
      <PARAMETER_LIST_ITEM>
         <NAME>ECX_PARTY_TYPE</NAME>
         <VALUE>S</VALUE>
```

Troubleshooting and Debugging

If you experience problems with your Business Event System integration, you can take the following troubleshooting steps:

- Confirm that WF_Listener is up and running.
- Ensure that business events are raised only after the BPEL process is deployed.
- If the BPEL process is created on one instance of Oracle Applications and deployed on another instance, ensure the following:
  - WF_BPEL_Q, WF_BPEL_QTab, and WF_BPEL_QAgent should be present on the target database.
  - A custom subscription for the raised business event should be present on the target database.

If you still experience problems with your integration, you can enable debugging.

Enabling Debugging

You can enable debugging for business events using the BPEL Process Manager.

To enable debugging:

1. Log into your BPEL Process Manager domain.
2. Select `*yourdomain*.collaxa.cube.ws`
3. Select **Debug.**

Debugging information is output to the log file for your domain. To examine the log file in the BPEL Process Manager, navigate to **Home > BPEL Domains >*yourdomain* > Logs**. The log file is ***yourdomain*.log**.

6

Using Concurrent Programs for BPEL Process Integration

This chapter covers the following topics:

- Overview of Concurrent Programs
- Design-Time Tasks for Concurrent Programs
- Creating a New BPEL Project
- Adding a Partner Link
- Configuring the Invoke Activity
- Configuring the Assign Activity
- Run-Time Tasks for Concurrent Programs
- Deploying the BPEL Process
- Testing the BPEL Process
- Verifying Records in Oracle Applications
- Troubleshooting and Debugging

Overview of Concurrent Programs

A concurrent program is an instance of an execution file. Concurrent programs use a concurrent program executable to locate the correct execution file. Several concurrent programs may use the same execution file to perform their specific tasks, each having different parameter defaults. Concurrent manager runs in the background waiting for a concurrent program to be submitted. As soon as a concurrent program is submitted, it is put into an execution queue by concurrent manager. Concurrent manager also manages the concurrent execution of concurrent programs.

Concurrent programs associated with the Open Interface Table move the data from interface table to base tables. While other concurrent programs execute the business logic and application-level processing for Oracle E-Business Suite.

Design-Time Tasks for Concurrent Programs

This section describes how to configure the OracleAS Adapter for Oracle Applications to use concurrent programs. It describes the tasks required to configure OracleAS Adapter for Oracle Applications using the Adapter Configuration Wizard in Oracle JDeveloper.

Prerequisites to Configuring Concurrent Programs

OracleAS Adapter for Oracle Applications is deployed using the BPEL Process Manager (PM) in Oracle JDeveloper. The BPEL PM creates the WSDL interfaces for the concurrent programs.

Populating Applications Context Header Variables

You need to populate certain variables in the SOA Suite to provide context information required in concurrent programs in order for an Oracle Applications user that has sufficient privileges to run the programs.

The context is set taking into account the values passed for the header parameters including *Username, Responsibility, Responsibility Application, Security Group*, and *NLS Language*. If the values for the new header parameters *Responsibility Application, Security Group*, and *NLS Language* are not passed, context information will be determined based on *Username* and *Responsibility*.

The default value for the *Username* is `SYSADMIN`, the default value for *Responsibility* is `SYSTEM ADMINISTRATOR`, the default *Security Group Key* is `STANDARD`, and the default *NLS Language* is `US`.

You can change the default values specified in the generated WSDL. This is a static way of changing the context information. These values would apply to all invocations of the deployed business process. However, if you need to provide different context information for different invocations of the business process, then you can dynamically populate the header values. The context information can be specified by configuring an Assign activity before the Invoke activity in the BPEL PM.

Detailed instructions on how to assign header variables, see Design-Time Tasks for Assigning Header Variables, page 3-9.

Following is a list of the procedures required to accomplish the design-time tasks.

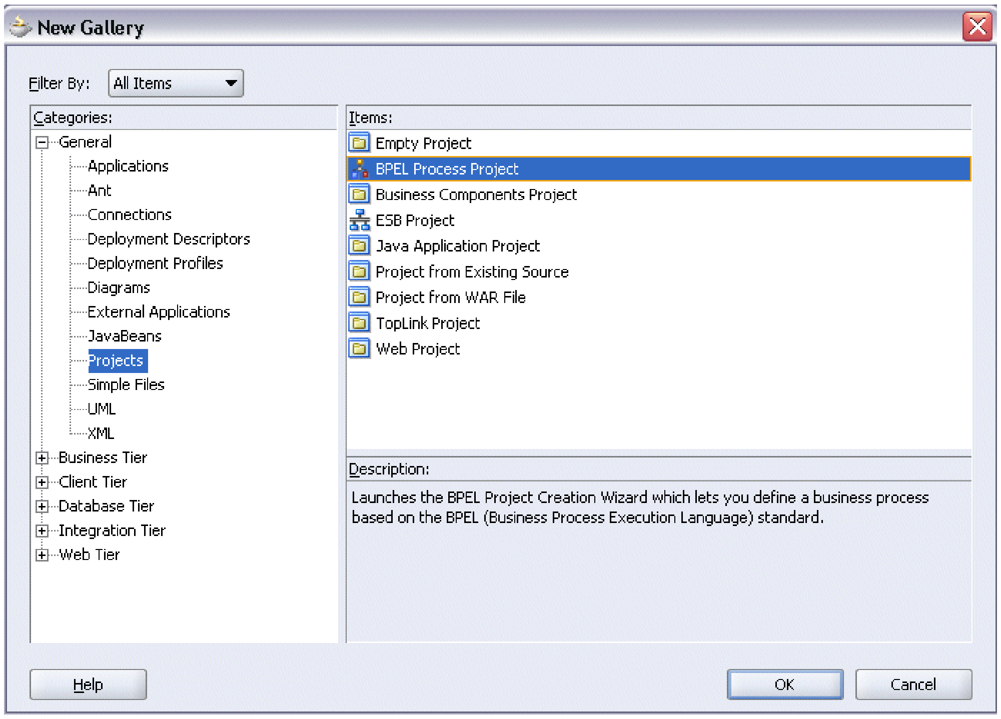
1. Create a new BPEL project, page 6-3
2. Add a partner link, page 6-4
3. Configure the Invoke activity, page 6-11
4. Configure the Assign activity, page 6-18

Creating a New BPEL Project

To create a new BPEL project:

1. Open JDeveloper BPEL Designer.
2. From the **File** menu, select **New**. The New Gallery dialog box appears.
3. Select **All Items** from the **Filter By** box. This displays a list of available categories.
4. Expand the **General** node, and then select **Projects**.
5. Select **BPEL Process Project** from the **Items** group.

Creating a New BPEL Process Project

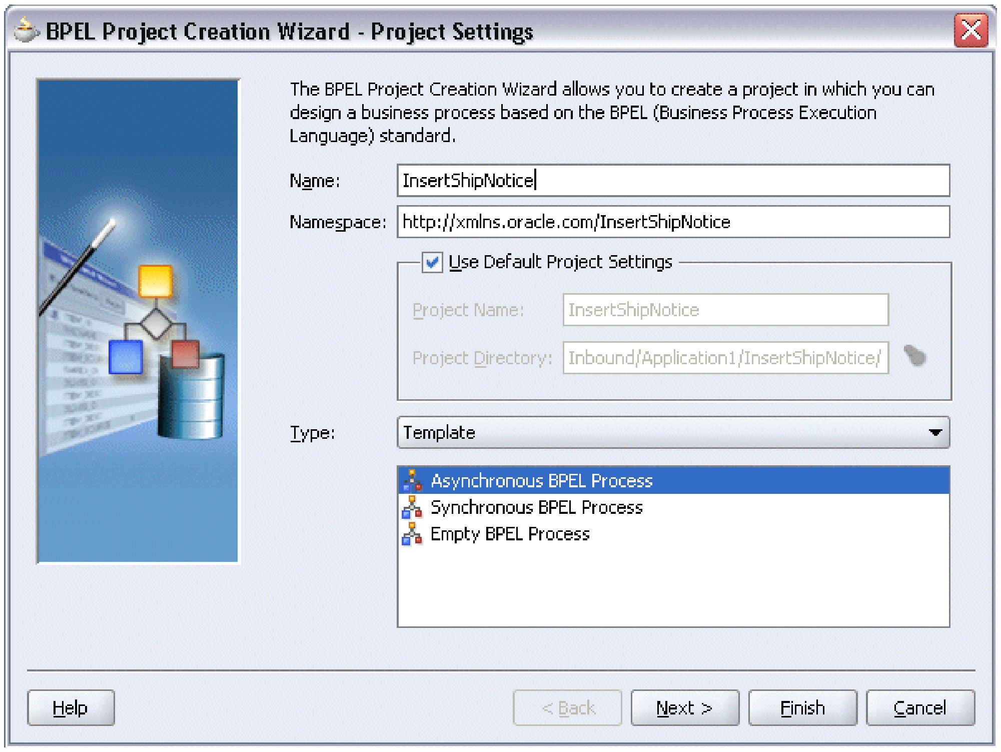


Launches the BPEL Project Creation Wizard which lets you define a business process based on the BPEL (Business Process Execution Language) standard.

6. Click **OK**. The BPEL Project Creation Wizard - Project Settings dialog box appears.
7. In the **BPEL Process Name** field, enter a descriptive name. For example, enter `InsertShipNotice`.

   Keep the default selection **Use Default Project Settings** unchanged.
8. Keep the default selection **Template** as the Type field. Select **Asynchronous BPEL Process** as the BPEL process type.

Specifying New BPEL Project Settings



The BPEL Project Creation Wizard allows you to create a project in which you can design a business process based on the BPEL (Business Process Execution Language) standard.

Name: InsertShipNotice

Namespace: http://xmlns.oracle.com/InsertShipNotice

Use Default Project Settings ✓

Project Name: InsertShipNotice

Project Directory: Inbound/Application1/InsertShipNotice/

Type: Template

Asynchronous BPEL Process

Synchronous BPEL Process

Empty BPEL Process

9. Click **OK**.

   A new asynchronous BPEL process is created with the receiveInput and callbackClient activities.

   The required source files including `bpel.xml`, `InsertShipNotice.bpel`, and `InsertShipNotice.wsdl` are also created.

Adding a Partner Link

The next task is to add a partner link to the BPEL process. A partner link defines the link name, type, and the role of the BPEL process that interacts with the partner service.

To add a partner link:

1. In JDeveloper BPEL Designer, click **Services** in the Component palette.

   Drag and drop **Oracle Applications** icon into the border area of the process diagram.

   The Adapter Configuration Wizard appears.

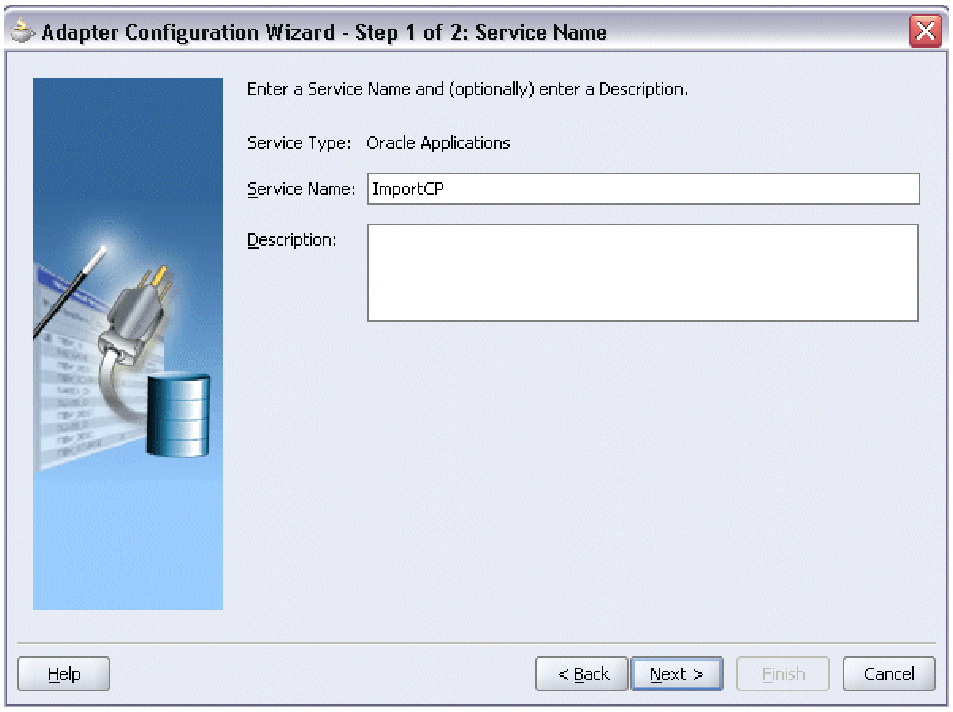
2. Click **Next** and the Service Name dialog box appears.

3. Enter the following information:

   1. In the **Service Name** field, enter a service name, such as `ImportCP`.

   2. In the **Description** field, enter a description for the service. This is an optional field. Click **Next**.

Specifying the Service Name



Adapter Configuration Wizard - Step 1 of 2: Service Name

Enter a Service Name and (optionally) enter a Description.

Service Type: Oracle Applications

Service Name: ImportCP

Description:

Click **Next**. The Service Connection dialog box appears.

4. You can perform either one of the following options for your database connection:

   Note: You need to connect to the database where Oracle E-Business Suite is running.

   - You can create a new database connection by clicking the **New** icon.

     Detailed instructions on how to create a new database connection, see Creating a New Database Connection, page 4-12.

   - Instead of creating a new database connection, you can select an existing connection that you have configured earlier from the **Connection** drop-down list.

     The selected database connection information appears in the Service

Connection dialog box. The JNDI (Java Naming and Directory Interface) name corresponding to the selected database connection also appears automatically in the **Database Server JNDI Name** field. Alternatively, you can specify a JNDI name.

> **Note:** When you specify a JNDI name, the deployment descriptor of the Adapter for Oracle Applications must associate this JNDI name with configuration properties required by the adapter to access the database.

The JNDI name acts as a placeholder for the connection used when your service is deployed to the BPEL server. This enables you to use different databases for development and later for production.

> **Note:** For more information about JNDI concepts, refer to *Oracle Application Server Adapter Concepts.*

5. Once you have created a new connection or selected an existing connection, you can add a concurrent program by browsing through the list of concurrent programs available in Oracle E-Business Suite.

6. Click **Next** in the Service Connection dialog box.

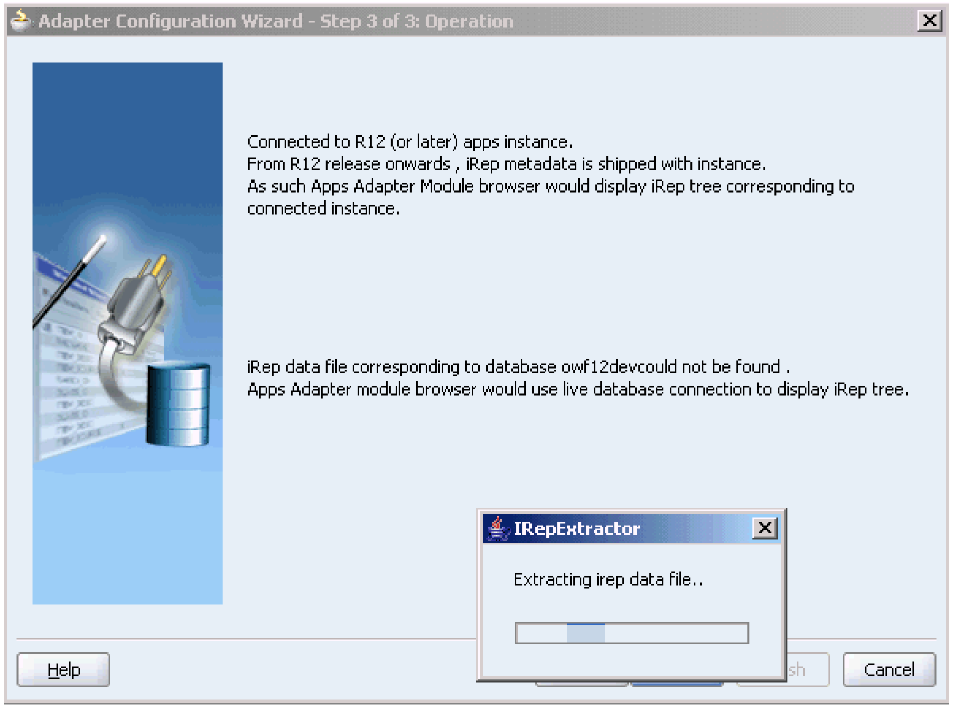
   For Oracle E-Business Suite Release 12:

   If you are connecting to Oracle E-Business Suite Release 12, then the **IREP File not present** dialog box appears indicating that OracleAS Adapter for Oracle Applications could not find the Oracle Integration Repository data file corresponding to the database you are connecting in your workspace. Absence of the data file would make browsing or searching of Integration Repository tree considerably slow. You can choose to extract the data file and create a local copy of the Integration Repository data file. Once it is created successfully, OracleAS Adapter for Oracle Applications will pick it up automatically next time and retrieve data from your local Integration Repository.

   You can select one of the following options:

   - Click **Yes** to extract the Integration Repository data file.

Extracting Integration Repository Data File



Adapter Configuration Wizard - Step 3 of 3: Operation

Connected to R12 (or later) apps instance.
From R12 release onwards , iRep metadata is shipped with instance.
As such Apps Adapter Module browser would display iRep tree corresponding to connected instance.

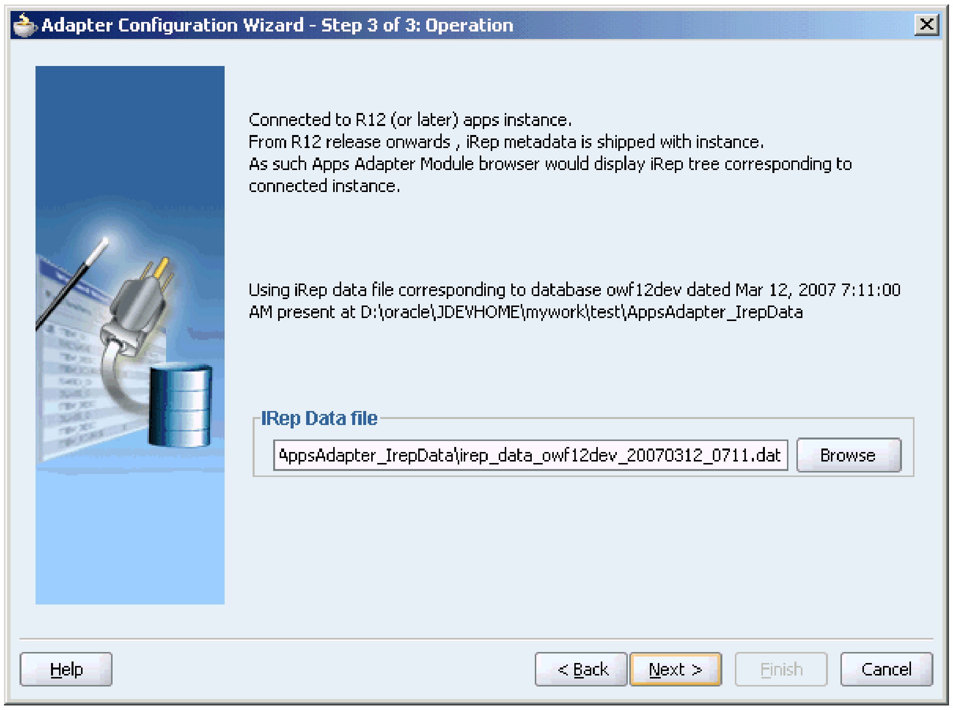
iRep data file corresponding to database owf12devcould not be found .
Apps Adapter module browser would use live database connection to display iRep tree.

IRepExtractor

Extracting irep data file..

After the system successfully creates a local copy of the Integration Repository data file, next time when you connect to the database, you will find the **IRep Data File** field appears in the Operation dialog box indicating where your local copy exists with the creation date and time as part of the file name.

Using the Local Integration Repository Data File



Connected to R12 (or later) apps instance.
From R12 release onwards , iRep metadata is shipped with instance.
As such Apps Adapter Module browser would display iRep tree corresponding to connected instance.

Using iRep data file corresponding to database owf12dev dated Mar 12, 2007 7:11:00 AM present at D:\oracle\JDEVHOME\mywork\test\AppsAdapter_IrepData

IRep Data file: AppsAdapter_IrepData\irep_data_owf12dev_20070312_0711.dat

- Click **No** to query the Integration Repository data file from the live database you are connecting to display the Integration Repository tree.

  Note: It is highly recommended that you create a local copy of the Integration Repository data file so that OracleAS Adapter for Oracle Applications will query the data next time from the local copy in your workspace to enhance the performance.

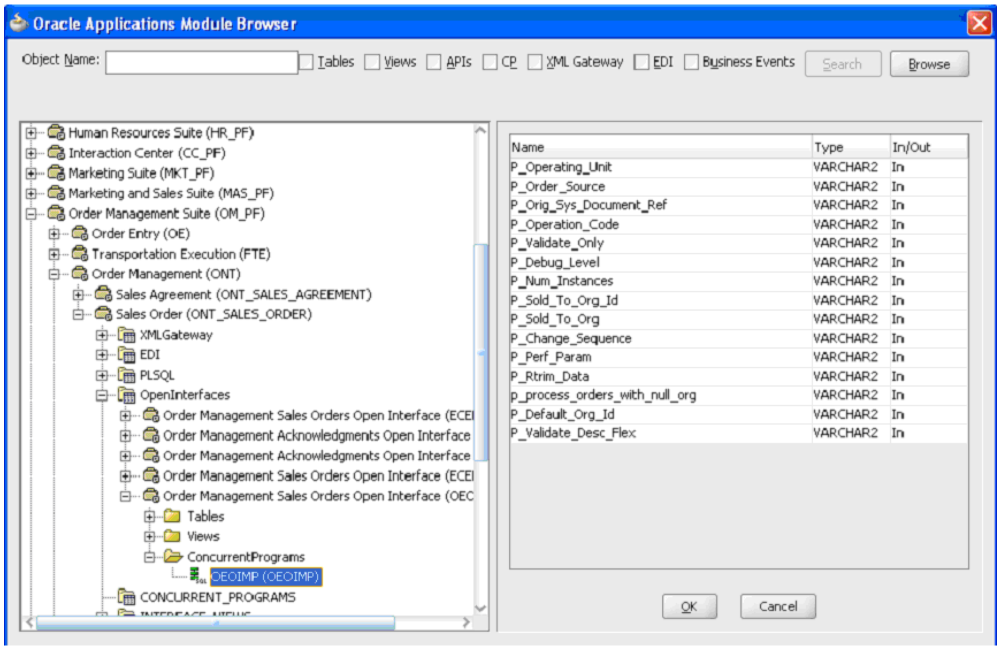
For Oracle E-Business Suite pre-Release 11.5.10:

If you are connecting to a pre-11.5.10 Oracle E-Business Suite instance, you must select the interface type in the Adapter Configuration Wizard. Select **Tables/Views/APIs/Concurrent Programs** to proceed.

Click **Add** to open the Oracle Applications Module Browser.

7. The Oracle Applications Module Browser combines interface data from Integration Repository with information about the additional interfaces supported by Adapter for Oracle Applications, organized in a tree hierarchy.

Specify a Concurrent Program


Oracle Applications Module Browser
Object Name:
Tables
Views
APIs
CP
XML Gateway
EDI
Business Events
Search
Browse
Human Resources Suite (HR_PF)
Interaction Center (CC_PF)
Marketing Suite (MKT_PF)
Marketing and Sales Suite (MAS_PF)
Order Management Suite (OM_PF)
Order Entry (OE)
Transportation Execution (FTE)
Order Management (ONT)
Sales Agreement (ONT_SALES_AGREEMENT)
Sales Order (ONT_SALES_ORDER)
XMLGateway
EDI
PLSQL
OpenInterfaces
Order Management Sales Orders Open Interface (ECEI
Order Management Acknowledgments Open Interface
Order Management Acknowledgments Open Interface
Order Management Sales Orders Open Interface (ECEI
Order Management Sales Orders Open Interface (OEC
Tables
Views
ConcurrentPrograms
OEOIMP (OEOIMP)
CONCURRENT_PROGRAMS
OK
Cancel

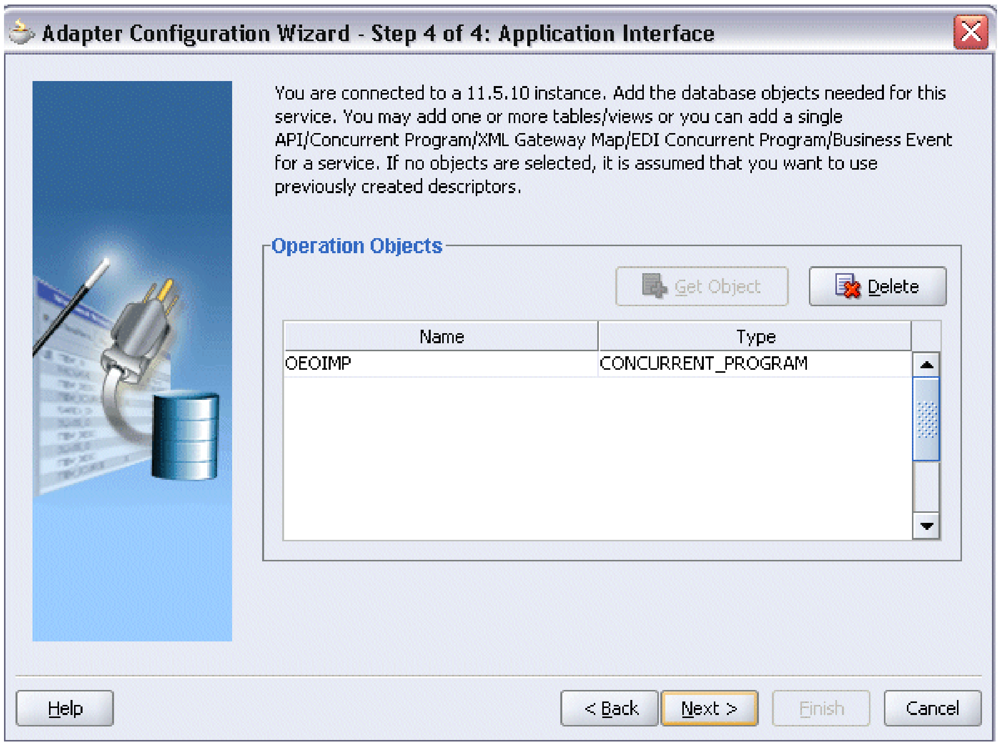
Name	Type	In/Out
P_Operating_Unit	VARCHAR2	In
P_Order_Source	VARCHAR2	In
P_Orig_Sys_Document_Ref	VARCHAR2	In
P_Operation_Code	VARCHAR2	In
P_Validate_Only	VARCHAR2	In
P_Debug_Level	VARCHAR2	In
P_Num_Instances	VARCHAR2	In
P_Sold_To_Org_Id	VARCHAR2	In
P_Sold_To_Org	VARCHAR2	In
P_Change_Sequence	VARCHAR2	In
P_Perf_Param	VARCHAR2	In
P_Rtrim_Data	VARCHAR2	In
p_process_orders_with_null_org	VARCHAR2	In
P_Default_Org_Id	VARCHAR2	In
P_Validate_Desc_Flex	VARCHAR2	In

Navigate to *Order Management Suite (OM_PF) >Order Management (ONT) > Sales Order (ONT_SALES_ORDER) > OpenInterfaces > Order Management Sales Orders Open Interface > ConcurrentPrograms* to select a concurrent program `OEOIMP (OEOIMP)`.

> **Note:** You can also search for a concurrent program by entering the name of the program in the **Object Name** field. Select the **CP** check box, and then click **Search**.

8. Click **OK**. The Application Interface page appears.

Application interface



Adapter Configuration Wizard - Step 4 of 4: Application Interface

You are connected to a 11.5.10 instance. Add the database objects needed for this service. You may add one or more tables/views or you can add a single API/Concurrent Program/XML Gateway Map/EDI Concurrent Program/Business Event for a service. If no objects are selected, it is assumed that you want to use previously created descriptors.

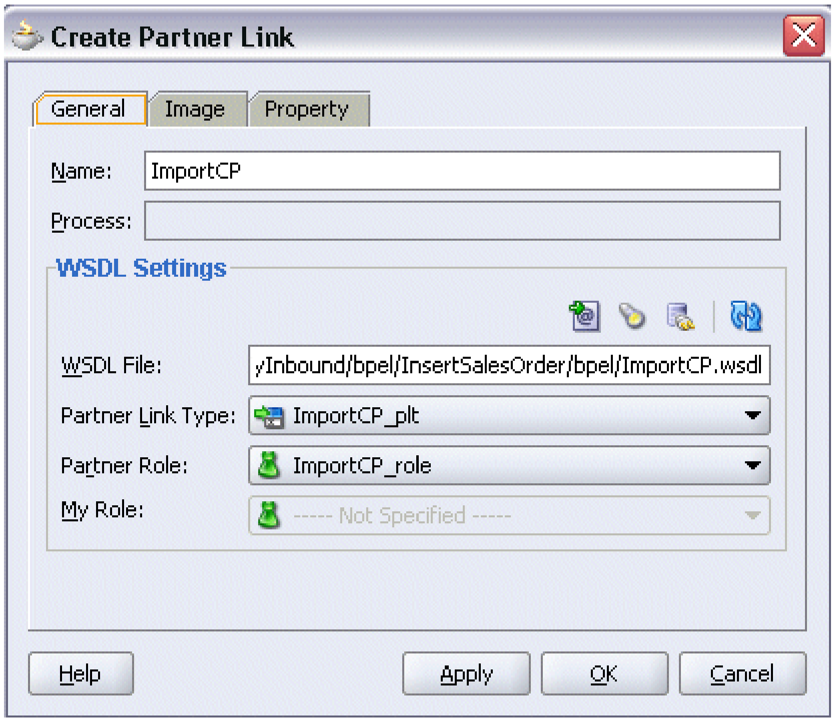
Operation Objects

Name	Type
OEOIMP	CONCURRENT_PROGRAM

9. Click **Next** and then click **Finish** to complete the process of configuring Adapter for Oracle Applications.

   The wizard generates the WSDL file corresponding to the XML schema. This WSDL file is now available for the partner link.

Create Partner Link



Name: ImportCP

Process:

WSDL File: yInbound/bpel/InsertSalesOrder/bpel/ImportCP.wsdl

Partner Link Type: ImportCP_plt

Partner Role: ImportCP_role

My Role: ----- Not Specified -----

10. Click **Apply** and **OK**. The partner link is created with the required WSDL settings.

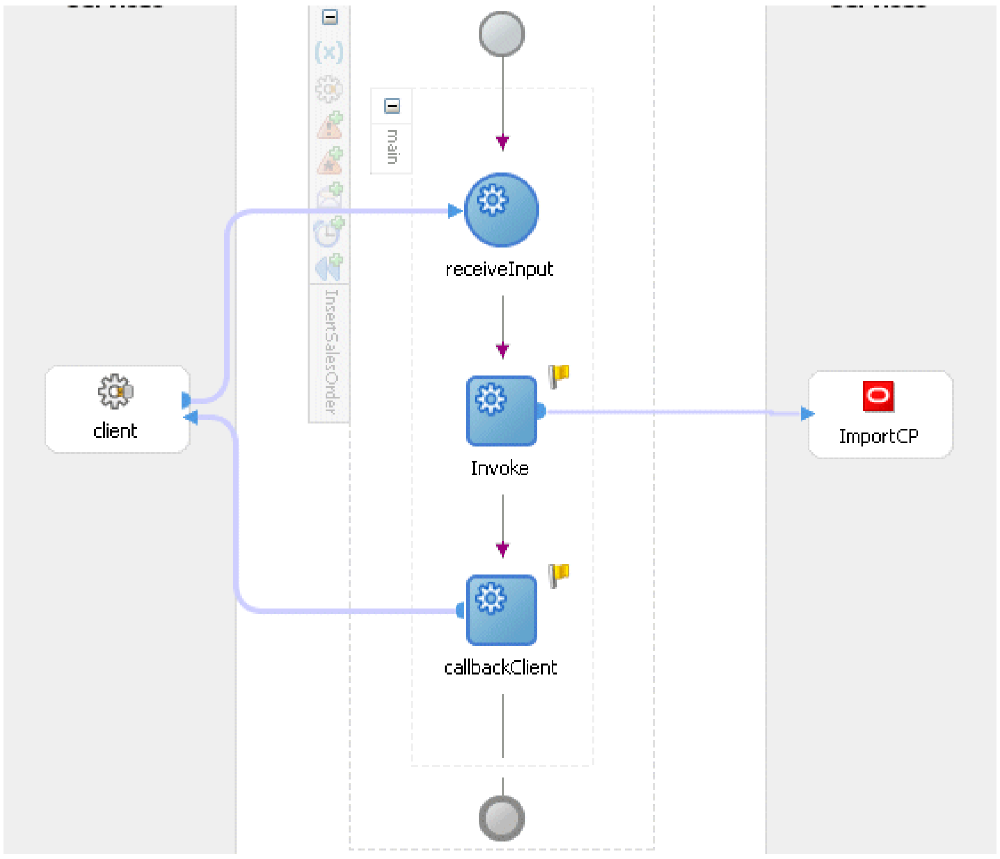
Configuring the Invoke Activity

After adding and configuring the partner link, the next task is to configure the Invoke activity.

To configure the Invoke activity:

1. Drag and drop the **Invoke** activity from the Component Palette into the process diagram.

Adding an Invoke Activity



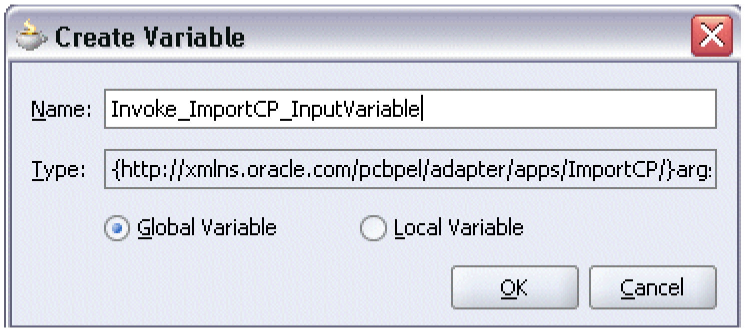
2. Link the Invoke activity to the `ImportCP` partner link. This opens the Invoke dialog box. The **General** tab is selected by default.

   Notice that the **Operation** field is automatically selected with the concurrent program that you chose.

3. Click the **Create** icon next to the **Input Variable** field. Enter a descriptive name for the variable in the Create Variable dialog box that appears. You can also accept the default name.

   Select **Global Variable** and click **OK**.

Creating the Input Variable


Create Variable

Name: Invoke_ImportCP_InputVariable

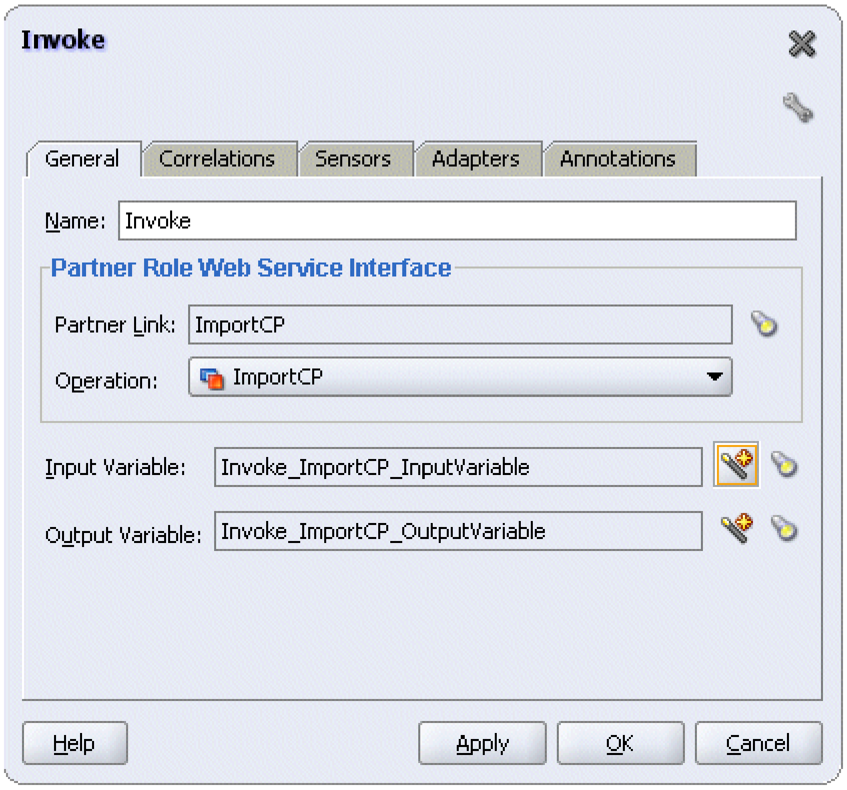
Type: {http://xmlns.oracle.com/pcbpel/adapter/apps/ImportCP/}arg:

[✓ Global Variable] Local Variable

4. Click the **Create** icon next to the **Output Variable** field. Enter a descriptive name for the variable in the Create Variable dialog box that appears. You can also accept the default name.

   Select **Global Variable** and click **OK**.

General Tab of the Invoke Activity


Invoke

Name: Invoke

Partner Role Web Service Interface

Partner Link: ImportCP

Operation: ImportCP

Input Variable: Invoke_ImportCP_InputVariable

Output Variable: Invoke_ImportCP_OutputVariable

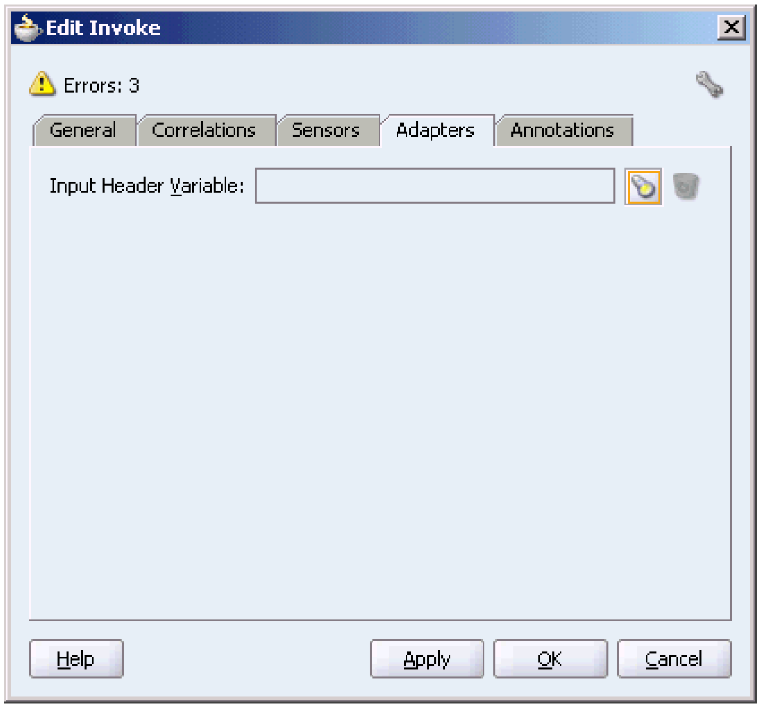
5. In the Invoke dialog box, click **Apply** and then **OK**.

Note: You can define an **Input Header Variable** under the **Adapters** tab of the Invoke dialog box. This variable can be used to provide context information for Oracle Applications in supporting multiple languages and multiple organization setups.

To Create the Header Variable:

1. Click the Adapters tab in the Edit Invoke dialog box and click **Browse Variable...** icon for the Input Header Variable field.

Specifying Input Header Variable



2. In the Variable Chooser dialog box, right-click on the Variables node and select Create Variable option from the menu.

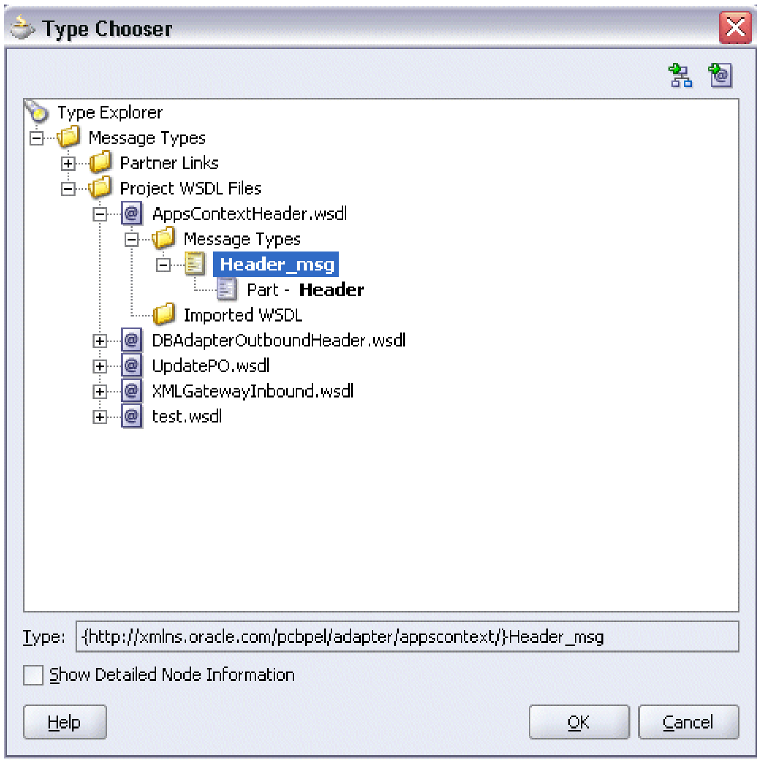
3. In the Create Variable dialog box, enter `header` in the Name field.

4. Select **Message Type** and click **Browse Message Type...** icon to open Type Chooser dialog box.

5. Expand the Partner Link node to locate the Header_msg node `{http://xmlns.oracle.com/pcbpel/adapter/appscontext/}Header_msg` for your partner link. The Header_msg node should be under the path, *Your Partner Link WSDL* >

AppsContextHeader.wsdl > Message Types > Header_msg.

Declaring Header Variable

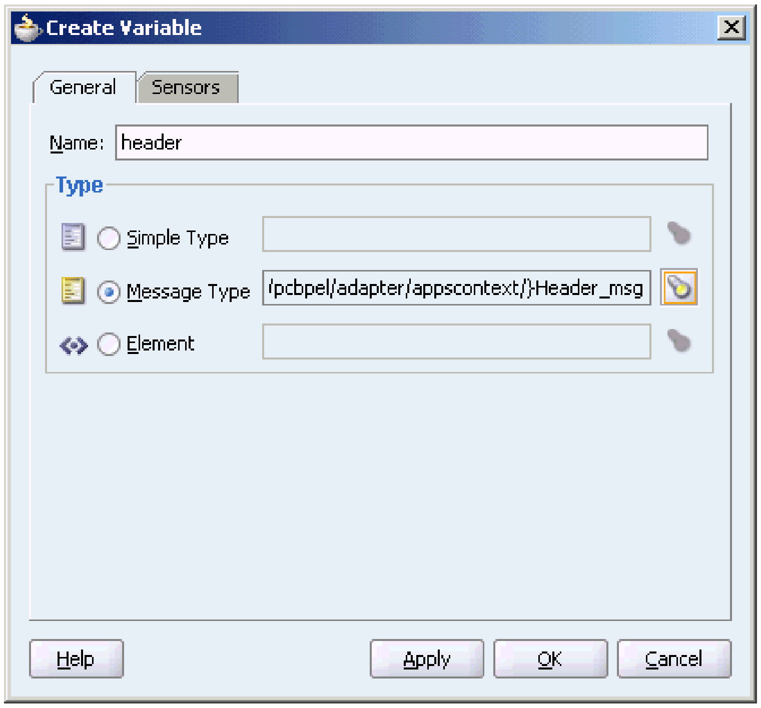


Type: {http://xmlns.oracle.com/pcbpel/adapter/appscontext/}Header_msg

☐ Show Detailed Node Information

6. Click **OK** to return to the Create Variable dialog box with your selected message type populated.

Populating Selected Header Variable



Name: header

Type

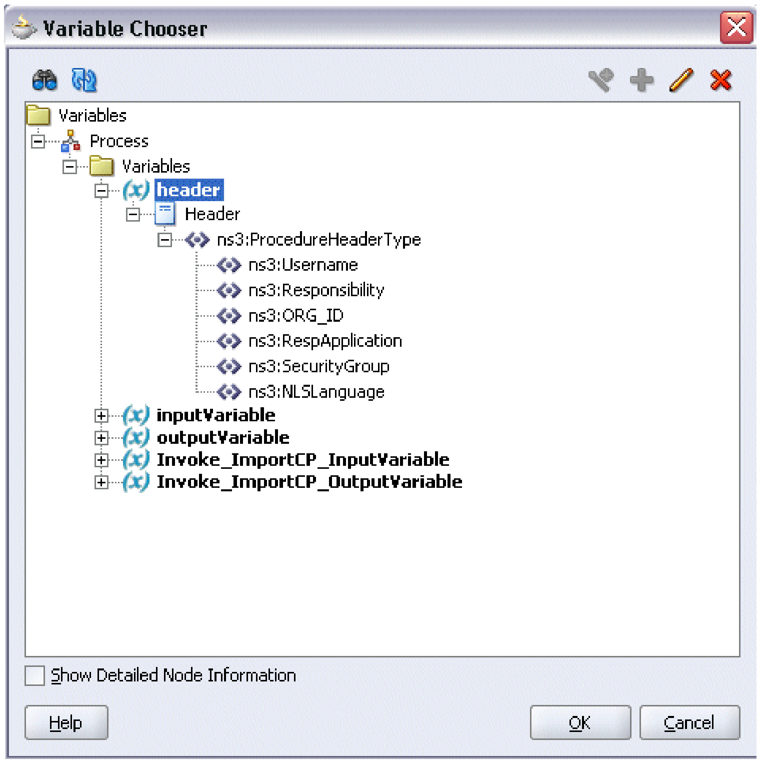
Simple Type

(●) Message Type /pcbpel/adapter/appscontext/}Header_msg

Element

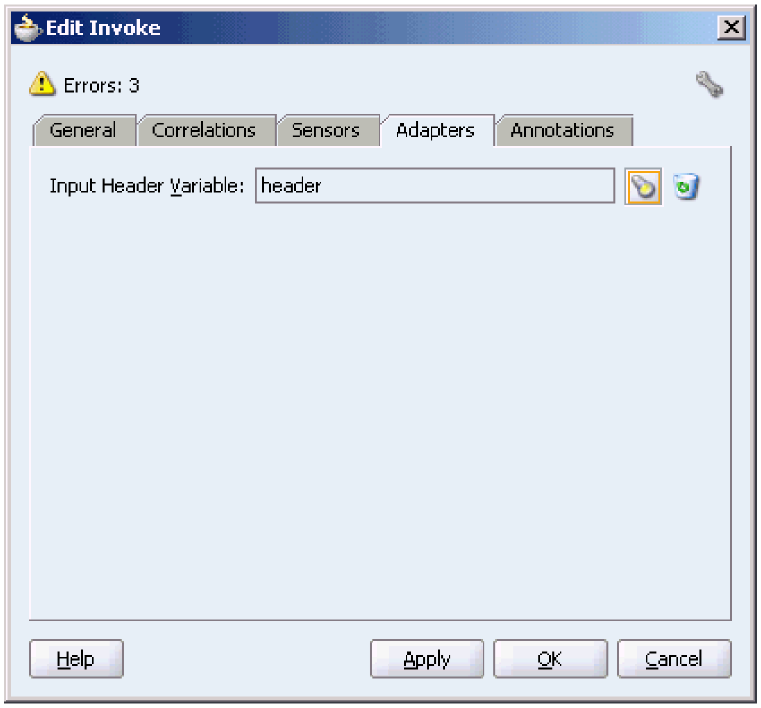
7. To view your header variable including Username, Responsibility, ORG_ID, Responsibility Application, Security Group, and NLS Language elements, click the **Browse Message Type...** icon to open Variable Chooser dialog box. Locate the header variable to view the variable hierarchical structure with these parameters needed for applications context.

Viewing Header Variable Structure



8. Click **OK** to return to the Edit Invoke dialog with the selected header variable populated for the Input Header Variable field. Click **Apply** to complete the header creation.

Populating Input Header Variable



Input Header Variable: header

Configuring the Assign Activity

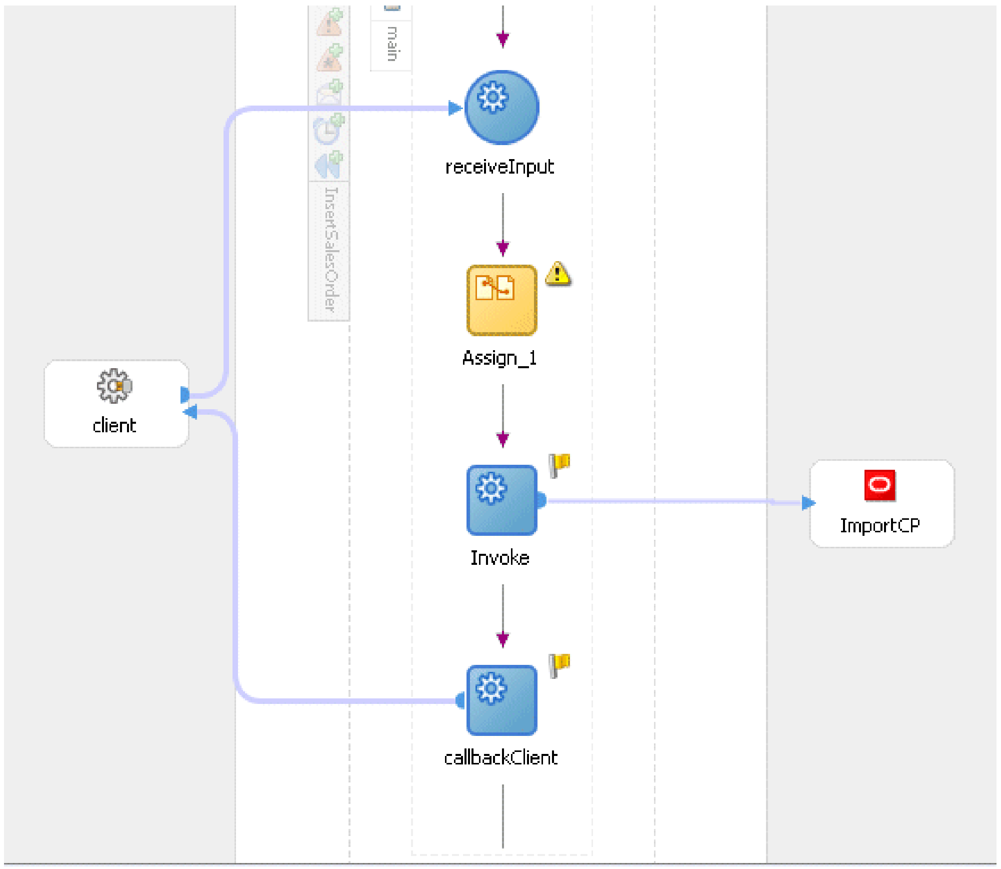
This step is to configure two Assign activities in order to:

- Set the applications context information.
- Set the payload for the `ImportCP` service.

To add the first Assign activity to set applications context information:

1. In JDeveloper BPEL Designer, drag and drop the **Assign** activity from the Component Palette into the Activity box of the process diagram before the **Invoke** activity.

Adding an Assign Activity



2. Double-click the **Assign** activity to access the Edit Assign dialog box.

3. Click the General tab to enter the name for the Assign activity, such as 'SetAppsContext'.

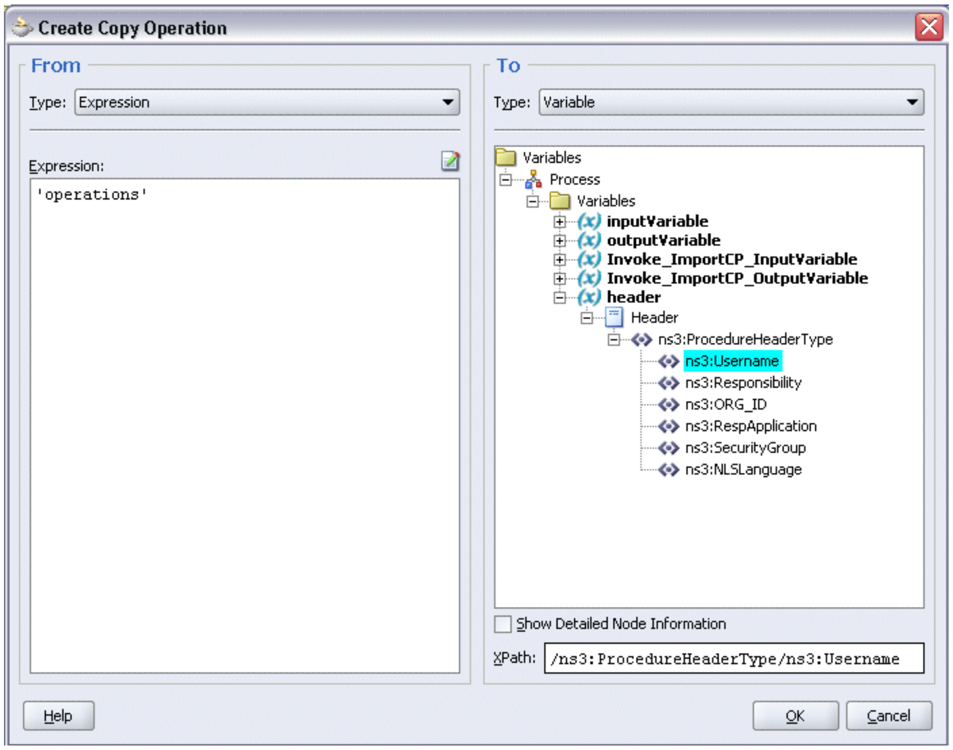
4. On the Copy Operation tab, click **Create** and then select **Copy Operation** from the menu. The Create Copy Operation window appears.

5. Enter the first pair of parameters:

   - In the From navigation tree, select type Expression and enter `'operations'` in the Expression box.

   - In the To navigation tree, select type Variable. Navigate to **Variable > Process > Variables > header > Header > ns3:ProcedureHeaderType** and select **ns3:Username**. The XPath field should contain your selected entry.

Assign Username Parameter



```
'operations'
```

XPath: `/ns3:ProcedureHeaderType/ns3:Username`

- Click **OK**.

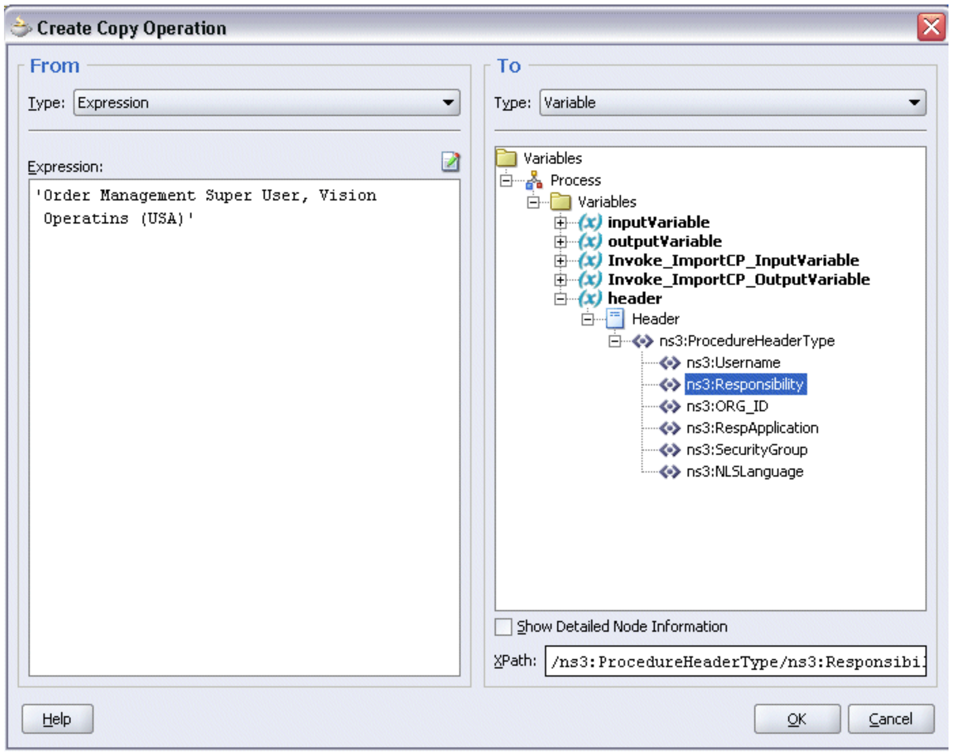
6. Enter the second pair of parameters:

   On the Copy Operation tab, click **Create** and select **Copy Operation** from the menu. The Create Copy Operation window appears.

   - In the From navigation tree, select type Expression and enter `'Order Management Super User, Vision Operatins (USA)'` in the Expression box.

   - In the To navigation tree, select type Variable. Navigate to **Variable** > **Process** > **Variables** > **header** > **Header**> **ns3:ProcedureHeaderType** and select **ns3:Responsibility**. The XPath field should contain your selected entry.

Assign Responsibility Parameter



From — Type: Expression

Expression:

```
'Order Management Super User, Vision
 Operatins (USA)'
```

To — Type: Variable

```
Variables
  Process
    Variables
      inputVariable
      outputVariable
      Invoke_ImportCP_InputVariable
      Invoke_ImportCP_OutputVariable
      header
        Header
          ns3:ProcedureHeaderType
            ns3:Username
            ns3:Responsibility
            ns3:ORG_ID
            ns3:RespApplication
            ns3:SecurityGroup
            ns3:NLSLanguage
```

☐ Show Detailed Node Information

XPath: `/ns3:ProcedureHeaderType/ns3:Responsibi`

- Click **OK**.

7. Enter the third pair of parameters:

   On the Copy Operation tab, click **Create** and select **Copy Operation** from the menu. The Create Copy Operation window appears.

   - In the From navigation tree, select type Expression and enter `'204'` in the Expression box.

   - In the To navigation tree, select type Variable. Navigate to **Variable** > **Process** > **Variables** > **header** > **Header**> **ns3:ProcedureHeaderType** and select **ns3:ORG_ID**. The XPath field should contain your selected entry.

   - Click **OK**.

8. Enter the fourth pair of parameters:

   On the Copy Operation tab, click **Create** and select **Copy Operation** from the menu. The Create Copy Operation window appears.

   - In the From navigation tree, select type Expression and enter `'ONT'` in the

Expression box.

- In the To navigation tree, select type Variable. Navigate to **Variable > Process > Variables > header > Header> ns3:ProcedureHeaderType** and select **ns3:RespApplication**. The XPath field should contain your selected entry.

- Click **OK**.

9. Enter the fifth pair of parameters:

   On the Copy Operation tab, click **Create** and select **Copy Operation** from the menu. The Create Copy Operation window appears.

   - In the From navigation tree, select type Expression and enter `'Standard'` in the Expression box.

   - In the To navigation tree, select type Variable. Navigate to **Variable > Process > Variables > header > Header> ns3:ProcedureHeaderType** and select **ns3:SecurityGroup**. The XPath field should contain your selected entry.

   - Click **OK**.

10. Enter the sixth pair of parameters:

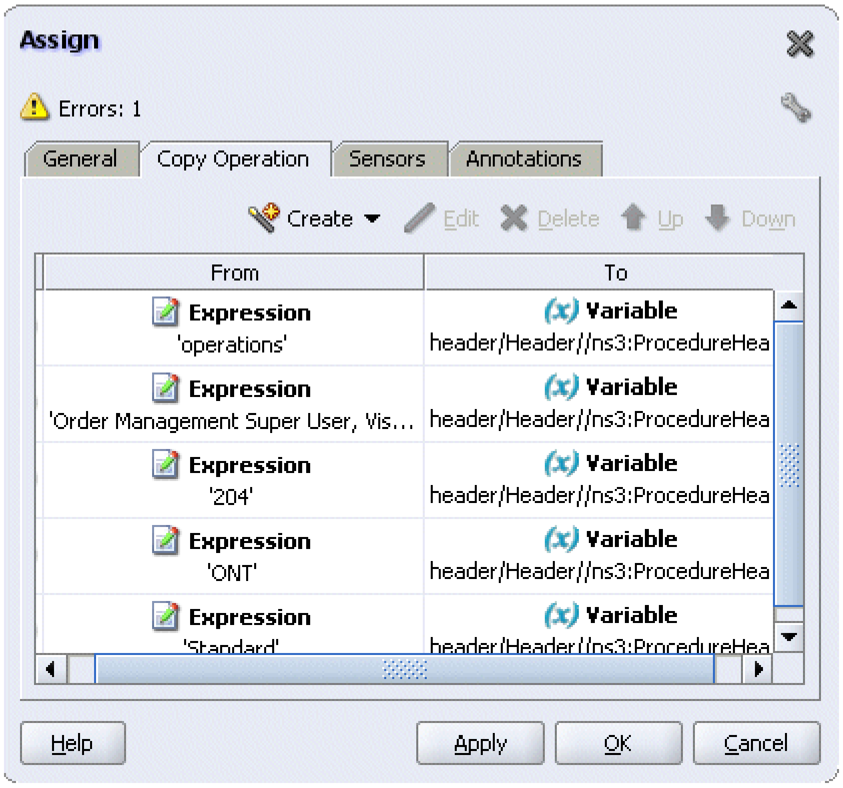
    On the Copy Operation tab, click **Create** and select **Copy Operation** from the menu. The Create Copy Operation window appears.

    - In the From navigation tree, select type Expression and enter `'US'` in the Expression box.

    - In the To navigation tree, select type Variable. Navigate to **Variable > Process > Variables > header > Header> ns3:ProcedureHeaderType** and select **ns3:NLSLanguage**. The XPath field should contain your selected entry.

    - Click **OK**.

Assign Activity



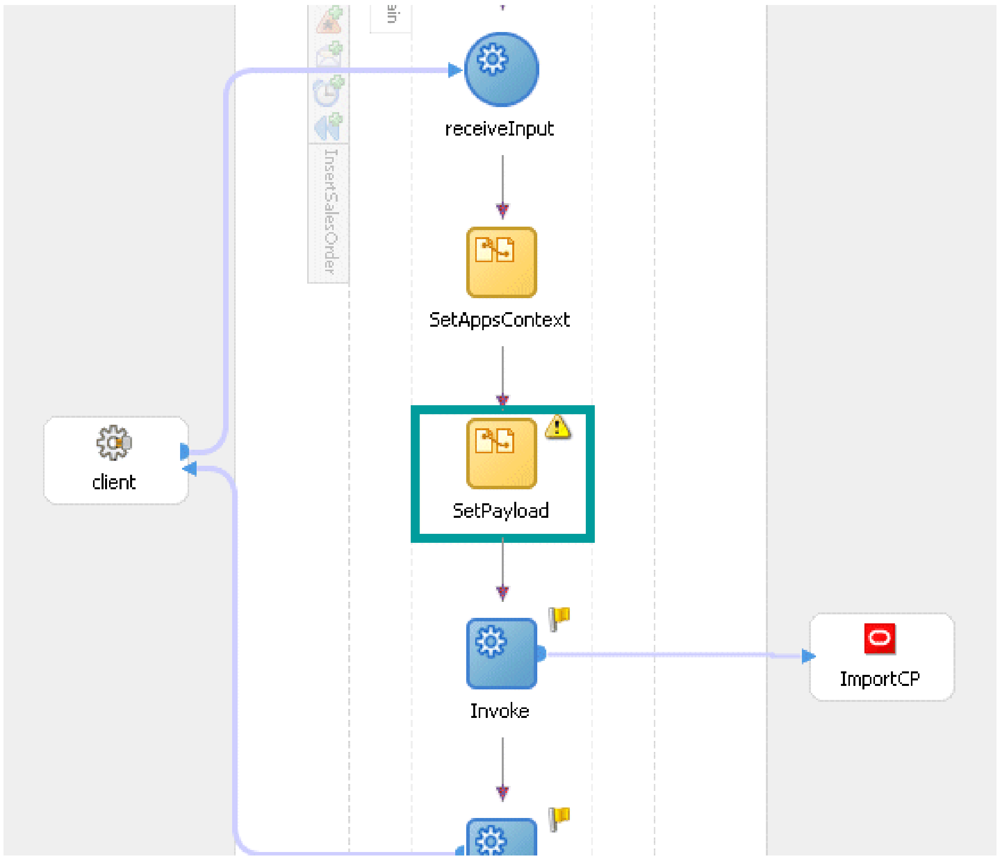
From	To
Expression 'operations'	Variable header/Header//ns3:ProcedureHea
Expression 'Order Management Super User, Vis...	Variable header/Header//ns3:ProcedureHea
Expression '204'	Variable header/Header//ns3:ProcedureHea
Expression 'ONT'	Variable header/Header//ns3:ProcedureHea
Expression 'Standard'	Variable header/Header//ns3:ProcedureHea

11. Click **Apply** and then **OK** to complete the configuration of the Assign activity.

To add the second Assign activity to set payload for the ImportCP service:

1. In JDeveloper BPEL Designer, drag and drop the **Assign** activity from the Component Palette into the Activity box of the process diagram after the first **Assign** activity.

Adding an Assign Activity



2. Double-click the **Assign** activity to access the Edit Assign dialog box.

3. Click the General tab to enter the name for the Assign activity, such as 'SetPayload'.

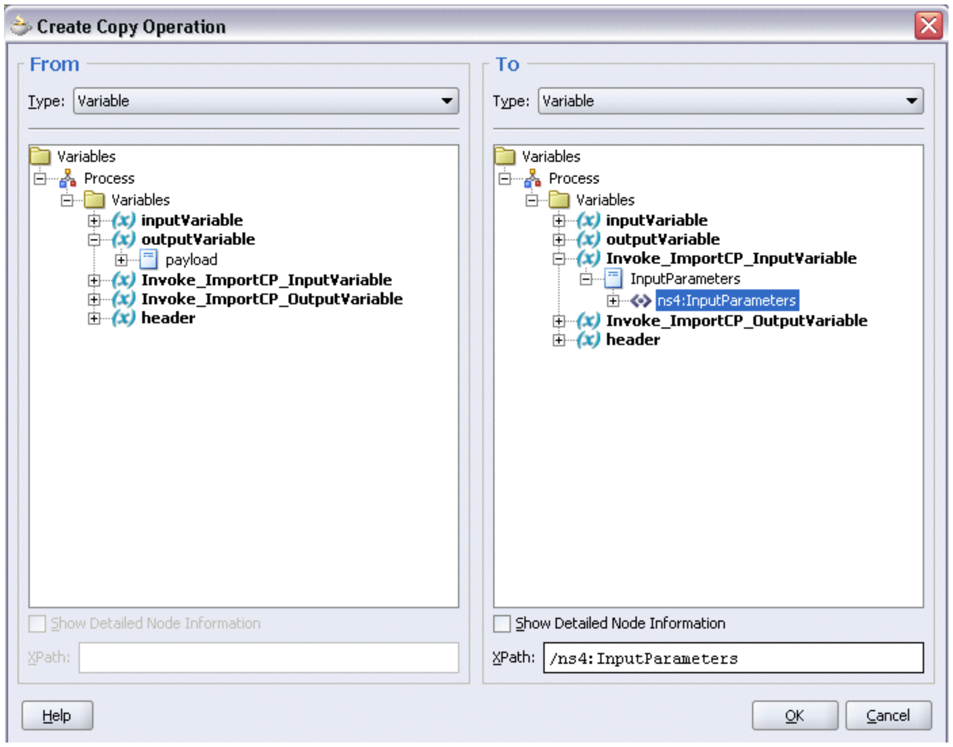
4. On the Copy Operation tab, click **Create** and select **Copy Operation** from the menu. The Create Copy Operation window appears.

5. Enter the following information to the Assign activity:

   - In the From navigation tree, select type Variable. Navigate to **Variable > Process > Variables > outputVariable** and select **payload**.

   - In the To navigation tree, select type Variable. Navigate to **Variable > Process > Variables > Invoke_ImportCP_InputVariable > InputParameters** and select **ns4:InputParameters**. The XPath field should contain your selected entry.

Assign InputParameters



XPath: `/ns4:InputParameters`

- Click **OK**.

6. Click **Apply** and then **OK** to complete the configuration of the Assign activity.

Run-Time Tasks for Concurrent Programs

After designing the BPEL process, the next step is to deploy, run and monitor it.

1. Deploy the BPEL process, page 6-25
2. Test the BPEL process, page 6-26
3. Verify records in Oracle Applications, page 6-29

Deploying the BPEL Process

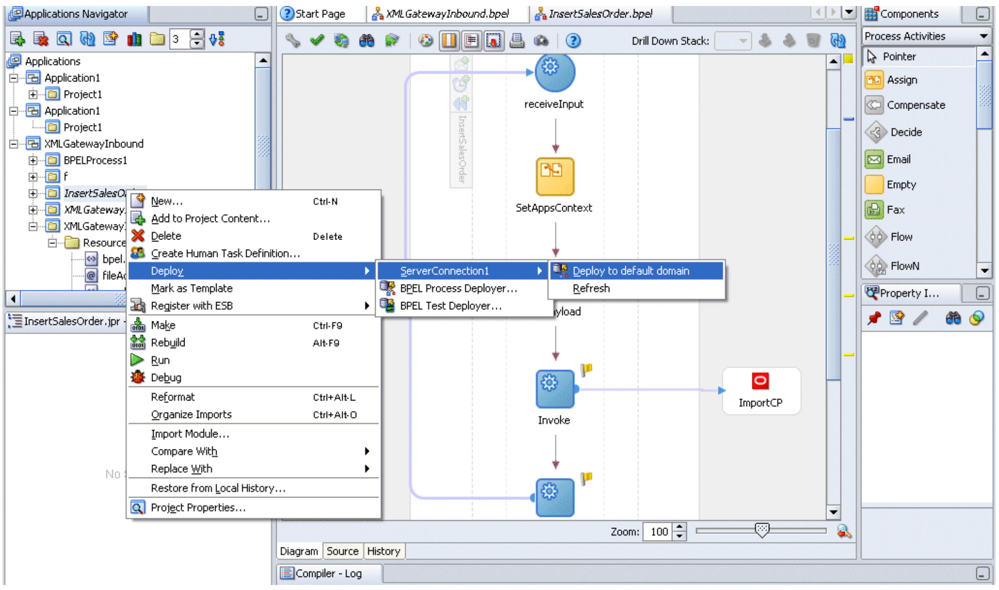
You must deploy the BPEL process before you can run it. The BPEL process is first compiled, and then deployed to the BPEL server.

To deploy the BPEL process:

1. Select the BPEL project in the Applications window.

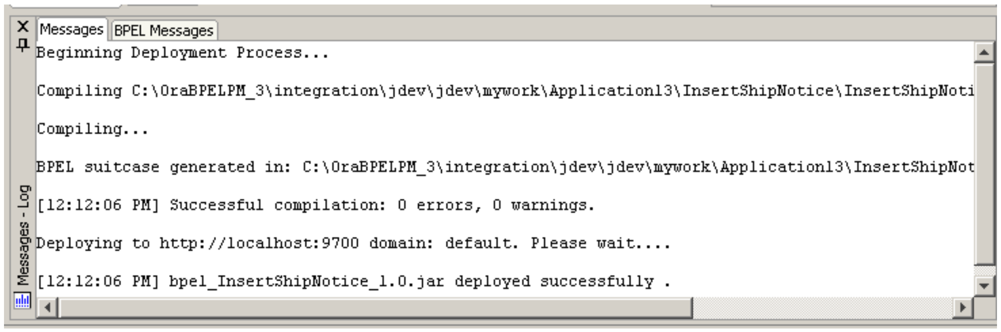
2. Right-click the project name, then select **Deploy > [Server Connection] > Deploy to Default Domain** from the menu that appears.

Deploying the BPEL Process



3. The BPEL process is compiled and deployed. You can check the progress in the Messages window.

Messages Window



```
Beginning Deployment Process...

Compiling C:\OraBPELPM_3\integration\jdev\jdev\mywork\Application13\InsertShipNotice\InsertShipNoti

Compiling...

BPEL suitcase generated in: C:\OraBPELPM_3\integration\jdev\jdev\mywork\Application13\InsertShipNot

[12:12:06 PM] Successful compilation: 0 errors, 0 warnings.

Deploying to http://localhost:9700 domain: default. Please wait....

[12:12:06 PM] bpel_InsertShipNotice_1.0.jar deployed successfully .
```

Testing the BPEL Process

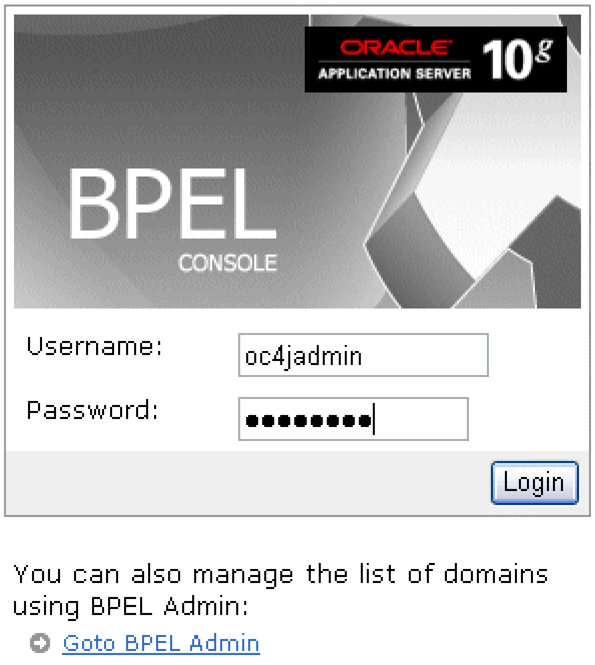
Once the BPEL process is deployed, you can manage and monitor the process from the

BPEL Console. You can also test the process and the integration interface by manually initiating the BPEL process.

To test the BPEL process:

1. Navigate to Oracle Application Server 10*g* BPEL Console (`http://<soaSuiteServerHostName>:<port>/BPELConsole`).

   The BPEL Console login page appears.

   Username: oc4jadmin

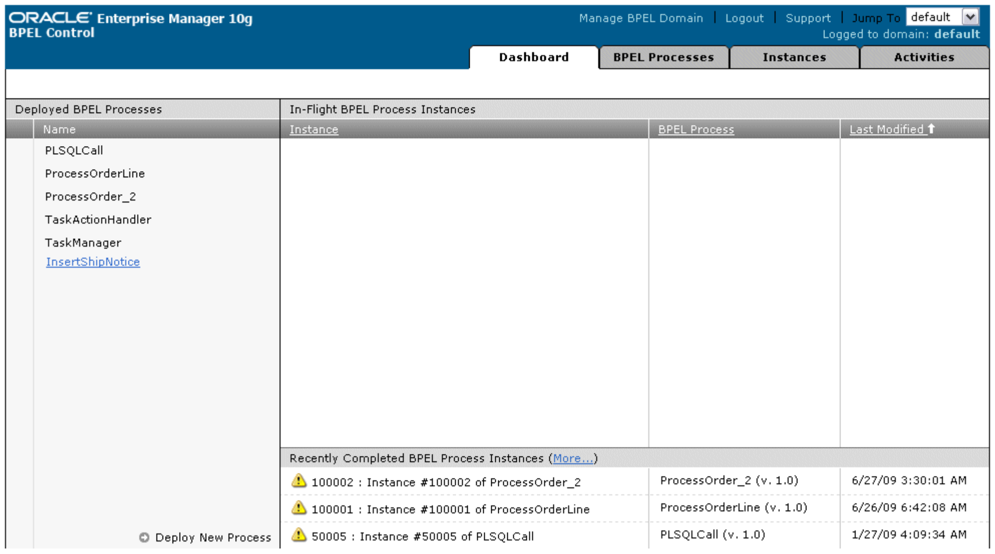
   Password: ●●●●●●●●●

   You can also manage the list of domains using BPEL Admin:

2. Enter the username and password and click **Login**.

3. The Oracle Enterprise Manager 10*g* BPEL Control appears. The list of deployed processes is shown under Deployed BPEL Processes.

Deployed BPEL Processes


ORACLE Enterprise Manager 10g
BPEL Control
Manage BPEL Domain | Logout | Support | Jump To default
Logged to domain: default
Dashboard
BPEL Processes
Instances
Activities

Deployed BPEL Processes
Name
PLSQLCall
ProcessOrderLine
ProcessOrder_2
TaskActionHandler
TaskManager
InsertShipNotice

Deploy New Process

In-Flight BPEL Process Instances

Instance	BPEL Process	Last Modified

Recently Completed BPEL Process Instances (More...)

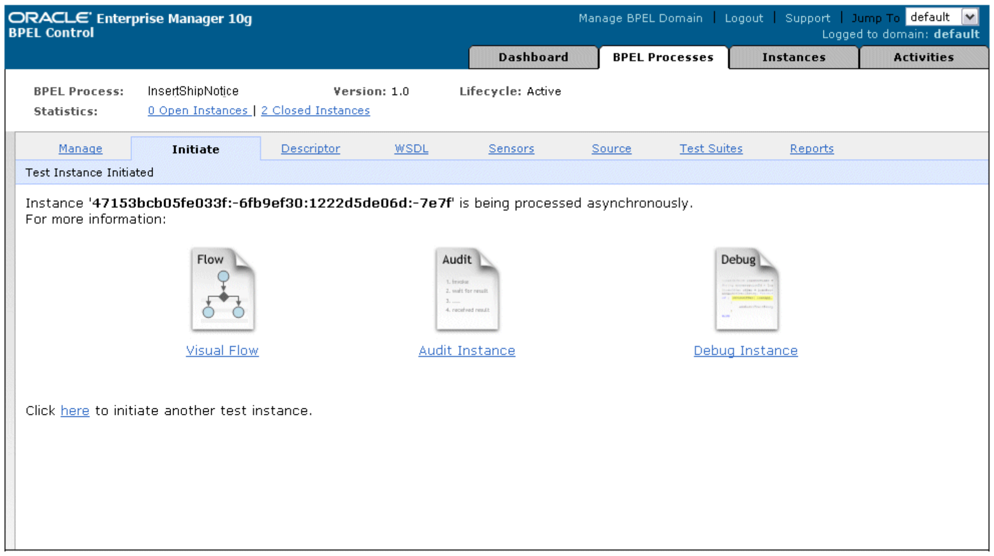
100002 : Instance #100002 of ProcessOrder_2	ProcessOrder_2 (v. 1.0)	6/27/09 3:30:01 AM
100001 : Instance #100001 of ProcessOrderLine	ProcessOrderLine (v. 1.0)	6/26/09 6:42:08 AM
50005 : Instance #50005 of PLSQLCall	PLSQLCall (v. 1.0)	1/27/09 4:09:34 AM

4. Click the BPEL process that you want to initiate. The Initiate page appears. Enter the input string required by the process.

5. Click **Post XML Message** to initiate the process.

6. The BPEL process is now initiated. You can check the process flow by clicking the **Visual Flow** icon.

BPEL Console Initiate Page


ORACLE Enterprise Manager 10g
BPEL Control
Manage BPEL Domain | Logout | Support | Jump To default
Logged to domain: default
Dashboard
BPEL Processes
Instances
Activities

BPEL Process: InsertShipNotice Version: 1.0 Lifecycle: Active
Statistics: 0 Open Instances | 2 Closed Instances

Test Instance Initiated

Instance '**47153bcb05fe033f:-6fb9ef30:1222d5de06d:-7e7f**' is being processed asynchronously.
For more information:

Click here to initiate another test instance.

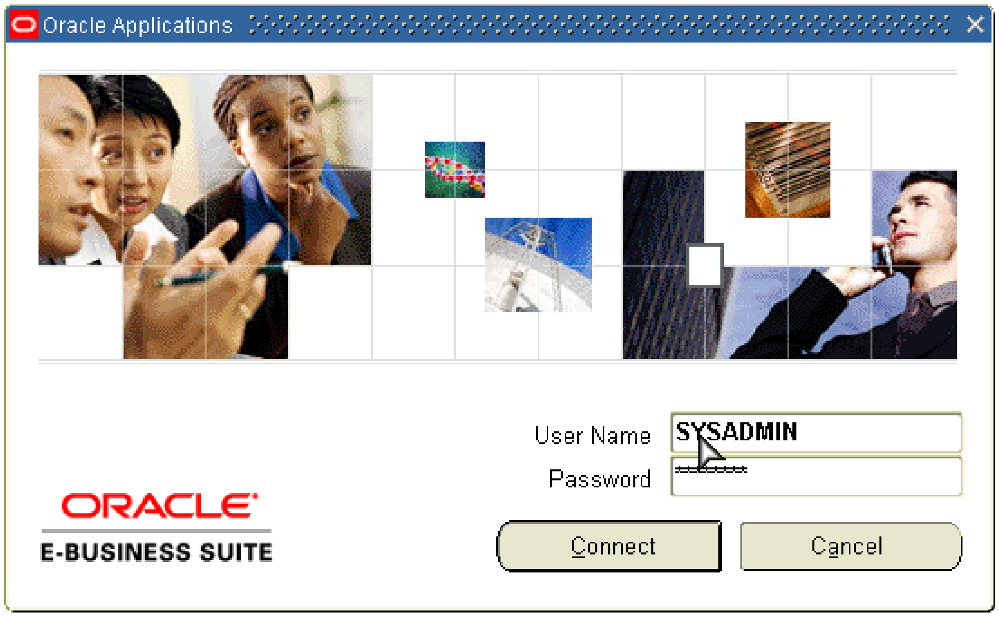
7. The audit trail provides information about the steps that have been executed. The audit trail also records the Request ID that is returned for the transaction. You can check the audit trail by clicking the **Audit Instance** icon.

Verifying Records in Oracle Applications

To verify records in Oracle Applications:

1. Log in to Oracle Applications as the System Administrator.

Oracle Applications Login Dialog Box



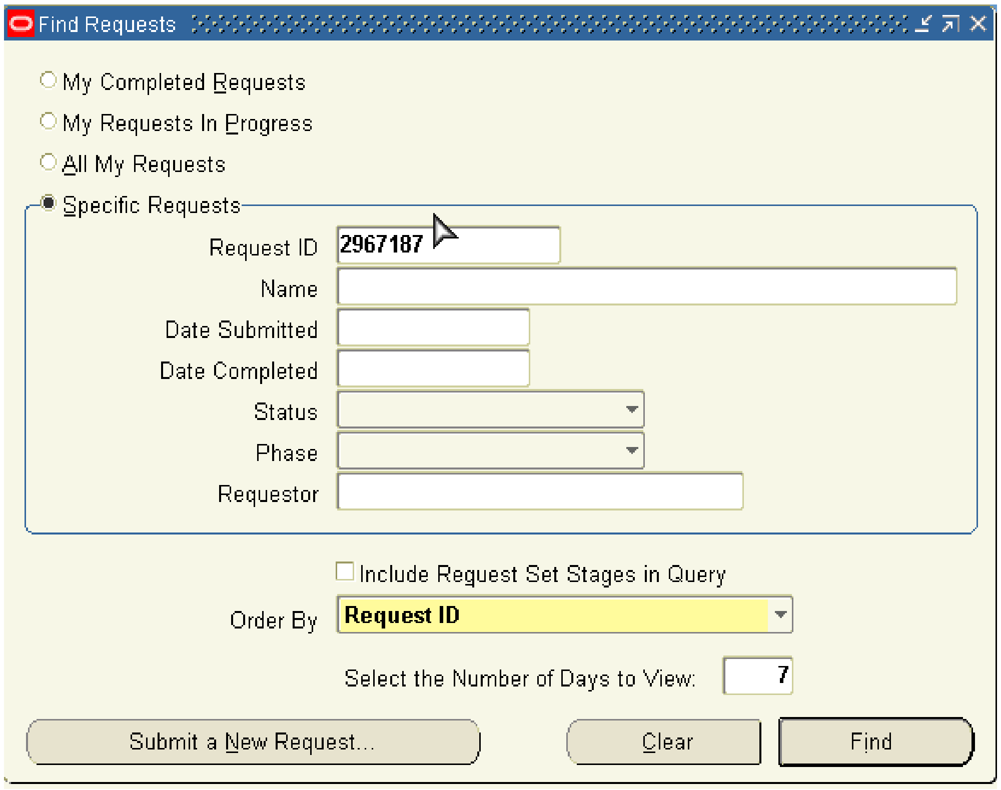
User Name SYSADMIN

Password ********

2. Select **Requests** from the **View** menu.

3. In the Find Requests form, search for the request by entering the Request ID that you got from the audit trail, and then click **Find**.

Find Requests Dialog Box


Find Requests

- ○ My Completed Requests
- ○ My Requests In Progress
- ○ All My Requests
- ● Specific Requests

Request ID	2967187
Name	
Date Submitted	
Date Completed	
Status	
Phase	
Requestor	

☐ Include Request Set Stages in Query

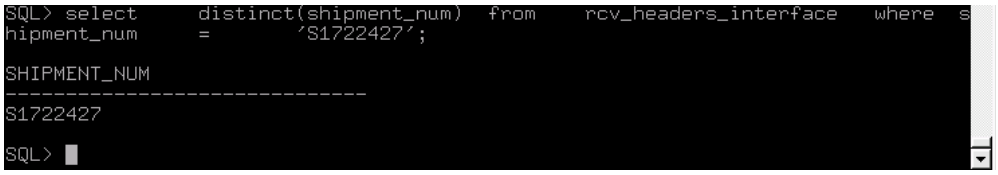
Order By: Request ID

Select the Number of Days to View: 7

4. The Request details are displayed. You can check for details such as the Phase and Status of the request.

5. If the **Status** of the request is `Complete`, you can also query the appropriate table in Oracle Applications to search for the relevant records that have been inserted.

Querying Oracle Applications for a Record



```
SQL> select     distinct(shipment_num)  from    rcv_headers_interface   where   s
hipment_num     =       'S1722427';

SHIPMENT_NUM
------------------------------
S1722427

SQL> 
```

Troubleshooting and Debugging

If you experience problems with your concurrent program integration, you can take the following troubleshooting steps:

- If you're using complex (that is, Boolean) datatypes, ensure that PL/SQL wrappers are applied on the target instance of the concurrent program.
- Examine your transaction data in the Open Interface Tracking Form.
- Check the status of the concurrent programs by Request ID.

If you still experience problems with your integration, you can enable debugging.

Enabling Debugging

You can enable debugging using the BPEL Process Manager.

To enable debugging:

1. Log into your BPEL Process Manager domain.
2. Select `yourdomain.collaxa.cube.ws`
3. Select **Debug**.

Debugging information is output to the log file for your domain. To examine the log file in the BPEL Process Manager, navigate to **Home > BPEL Domains >*yourdomain* > Logs**. The log file is ***yourdomain*.log**.

7

Using Interface Tables and Views for BPEL Process Integration

This chapter covers the following topics:

Overview of Interface Tables and Views

OracleAS Adapter for Oracle Applications uses interface tables to insert and update data in Oracle Applications, and uses views to retrieve data from Oracle Applications. This chapter describes the following interfaces:

- Interface Tables
- Interface Views

Interface Tables

OracleAS Adapter for Oracle Applications use interface tables to insert data in Oracle Applications. For example, by using interface tables, you can insert a purchase order into Oracle Applications to generate the sales order automatically. Data is never loaded directly into Oracle Applications base tables. Instead, data is first loaded into interface tables, and then Oracle-supplied concurrent programs move data from interface tables to base tables. This ensures that all business logic and processing is handled using Oracle components.

OracleAS Adapter for Oracle Applications use open interface tables to integrate with Oracle Applications through direct database access. The OracleAS Adapter for Oracle Applications inserts data into the open interface tables. These interface tables can be used only for insert operations and support only an inbound connection into Oracle Applications.

Interface tables are intermediate tables into which the data is inserted first. Once the data gets inserted into the interface tables, the data is validated, and then transferred to the base tables. Base tables are real application tables that reside in the application database. The data that resides in the interface tables is transferred to the base tables using concurrent programs. A concurrent program is an instance of an execution file. Concurrent programs are scheduled in Oracle Applications to move data from interface tables to base tables. These programs perform the application-level checks and run validation before inserting data into base tables.

Interface Views

OraclesAS Adapter for Oracle Applications uses interface views to retrieve data from Oracle Applications. For example, by using the views, you can retrieve information about your customers from the required tables in Oracle Applications.

OracleAS Adapter for Oracle Applications uses interface views to retrieve data from Oracle Applications. Views allow only simple definition. By using the views, you can get synchronous data access to Oracle Applications. In OracleAS Adapter for Oracle Applications, interface views are created on base tables as well as interface tables. These views can be used *only* for select operations.

In the Oracle Applications 11.5.10 release, you cannot work on multiple interface views. A work around to address this would be to create a view that spans multiple views.

Design-Time Tasks for Interface Tables

This section describes how to configure the OracleAS Adapter for Oracle Applications to use interface tables. It describes the steps to configure OracleAS Adapter for Oracle Applications using the Adapter Configuration Wizard in Oracle JDeveloper.

Prerequisites to Configure Interface Tables

- Define primary keys on all the interface tables being used.
- Define parent-child relationships among all the interface tables used.

Following is a list of the procedures required to accomplish the design-time tasks.

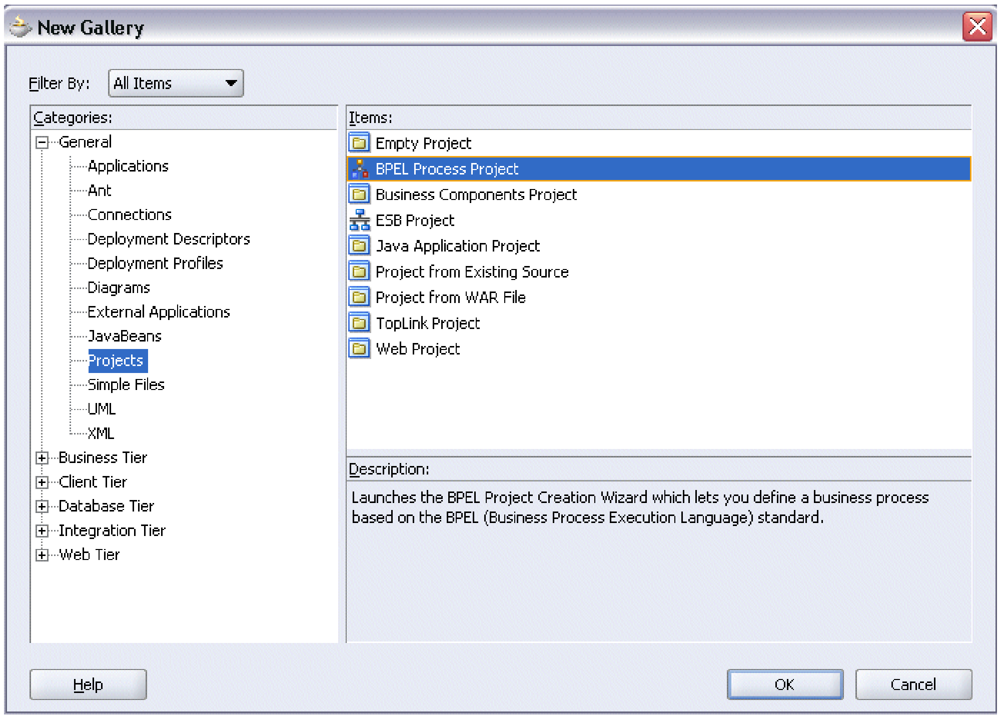
1. Create a new BPEL project, page 7-3
2. Add a partner link, page 7-6
3. Configure the Invoke activity, page 7-19
4. Configure the Assign activity, page 7-21

Creating a New BPEL Project

To create a new BPEL project:

1. Open JDeveloper BPEL Designer.
2. From the **File** menu, select **New**. The New Gallery dialog box appears.
3. Select **All Items** from the **Filter By** list. This displays a list of available categories.
4. Expand the **General** node, and then select **Projects**.
5. Select **BPEL Process Project** from the **Items** list.

Creating a New BPEL Process Project



Launches the BPEL Project Creation Wizard which lets you define a business process based on the BPEL (Business Process Execution Language) standard.

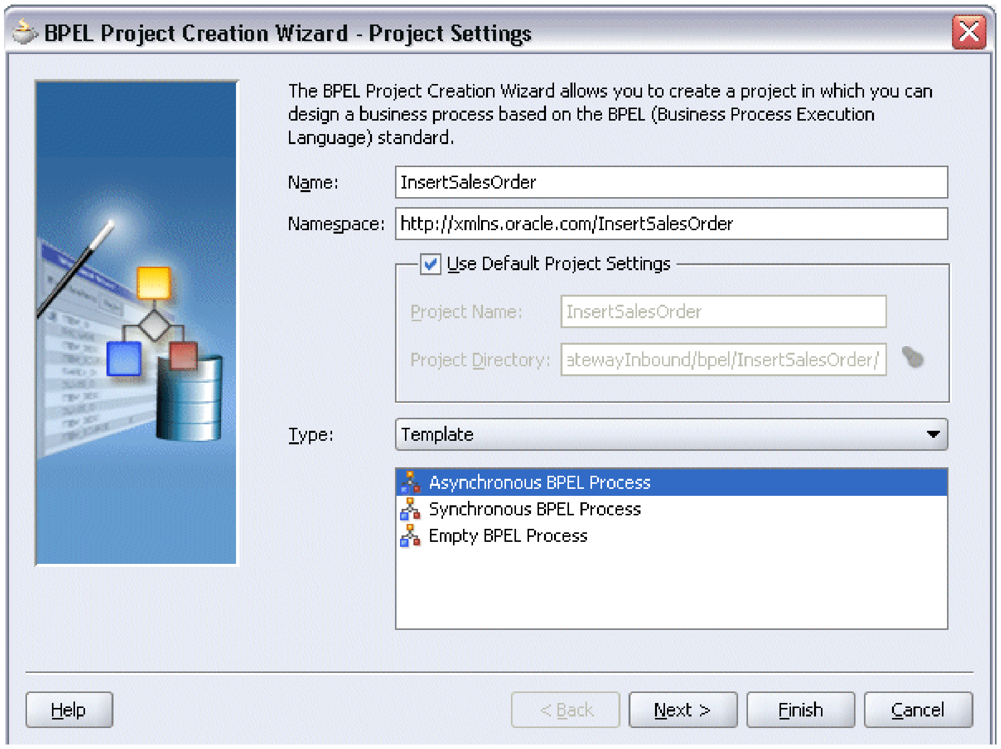
6. Click **OK**. The BPEL Project Creation Wizard - Project Settings dialog box appears.

7. In the **Name** field, enter a descriptive name. For example, enter `InsertSalesOrder`.

   Keep the default selection **Use Default Project Settings** unchanged.

8. Keep the default selection **Template** as the Type field. Select **Asynchronous BPEL Process** as the BPEL process type.

Specifying New BPEL Project Settings


BPEL Project Creation Wizard - Project Settings

The BPEL Project Creation Wizard allows you to create a project in which you can design a business process based on the BPEL (Business Process Execution Language) standard.

Name: InsertSalesOrder

Namespace: http://xmlns.oracle.com/InsertSalesOrder

[✓] Use Default Project Settings

Project Name: InsertSalesOrder

Project Directory: atewayInbound/bpel/InsertSalesOrder/

Type: Template

Asynchronous BPEL Process

Synchronous BPEL Process

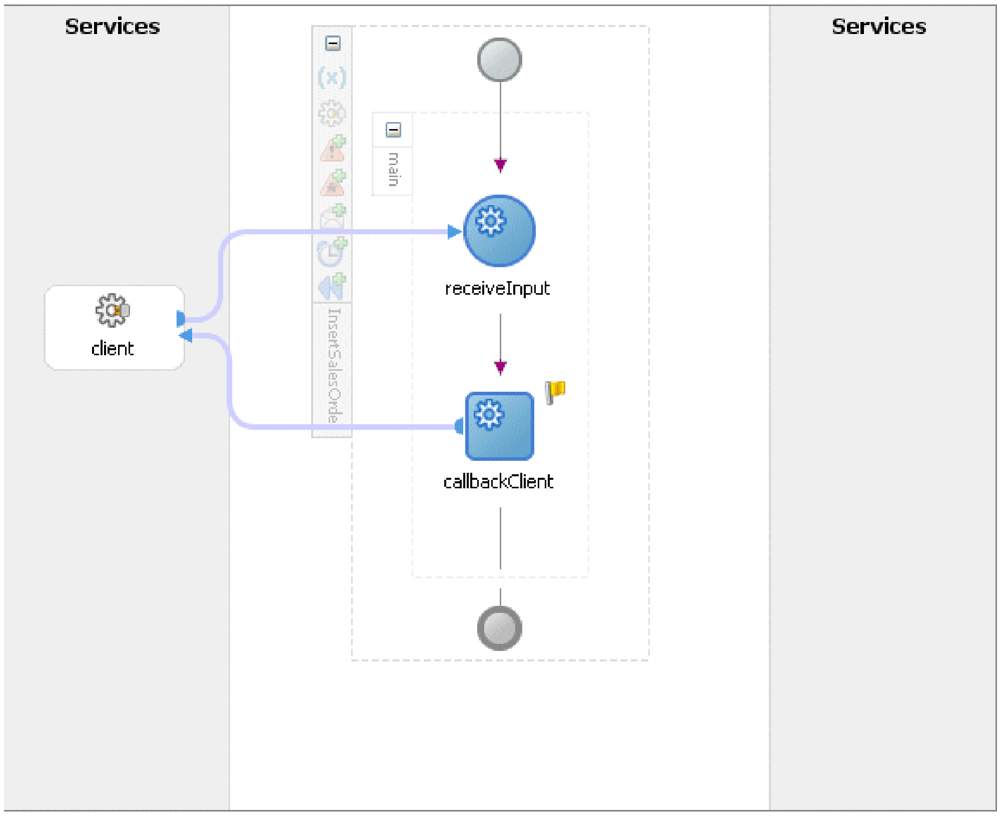
Empty BPEL Process

9. Click **OK**.

   A new asynchronous BPEL process is created with the receiveInput and callbackClient activities.

   The required source files including `bpel.xml`, `InsertSalesOrder.bpel`, and `InsertSalesOrder.wsdl` are also generated.

New BPEL Process Project



Adding a Partner Link

This section describes how to add a partner link to your BPEL process. A partner link defines the link name, type, and the role of the BPEL process that interacts with the partner service.

To add a partner link:

1. In JDeveloper BPEL Designer, click **Services** in the Component palette.

   Drag and drop **Oracle Applications** icon into the border area of the process diagram.

   The Adapter Configuration Wizard appears.

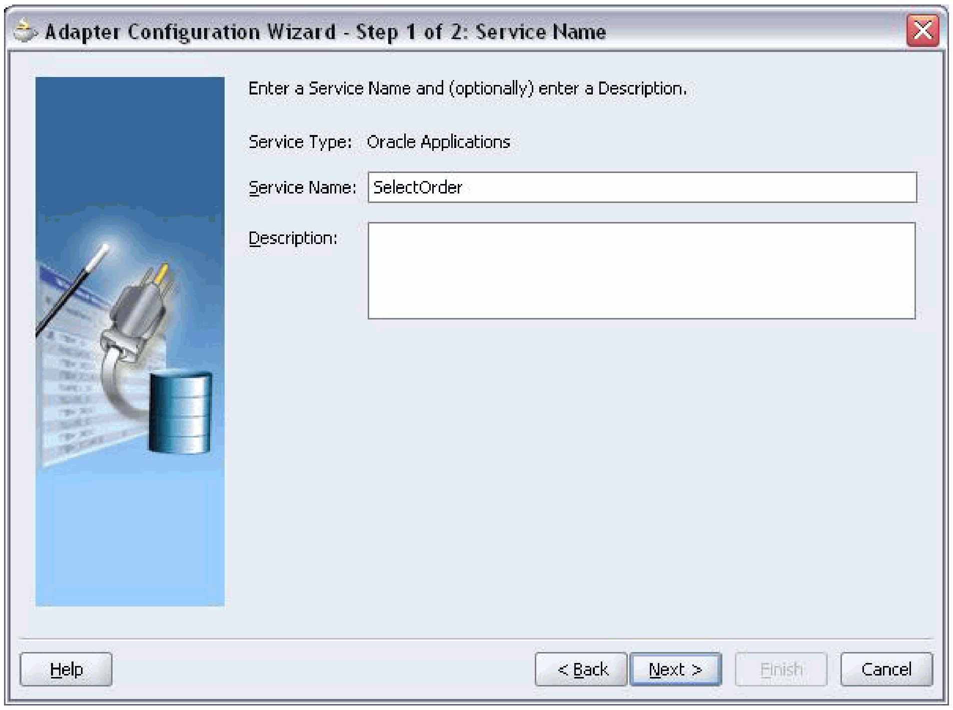
2. Click **Next**. The Service Name dialog box appears.

3. Enter the following information:

   1. In the **Service Name** field, enter a service name.

2. In the **Description** field, enter a description for the service. This is an optional field.

Specifying the Service Name



Enter a Service Name and (optionally) enter a Description.

Service Type: Oracle Applications

Service Name: SelectOrder

Description:

4. Click **Next**. The Service Connection dialog box appears.

5. You can perform either one of the following options for your database connection:

   Note: You need to connect to the database where Oracle E-Business Suite is running.

   - You can create a new database connection by clicking the **New** icon.

     Detailed instructions on how to create a new database connection, see Creating a New Database Connection, page 4-12.

   - You can select an existing database connection that you have configured earlier from the **Connection** drop-down list.

     The selected database connection information appears in the Service Connection dialog box. The JNDI (Java Naming and Directory Interface) name corresponding to the selected database connection also appears automatically in the **Database Server JNDI Name** field. Alternatively, you can specify a JNDI name.

Note: When you specify a JNDI name, the deployment descriptor of the Adapter for Oracle Applications must associate this JNDI name with configuration properties required by the adapter to access the database.

The JNDI name acts as a placeholder for the connection used when your service is deployed to the BPEL server. This enables you to use different databases for development and later for production.

Note: For more information about JNDI concepts, refer to *Oracle Application Server Adapter Concepts.*

6. Once you have created a new connection or selected an existing connection, you can add an interface table available in Oracle E-Business Suite.

7. Click **Next**.

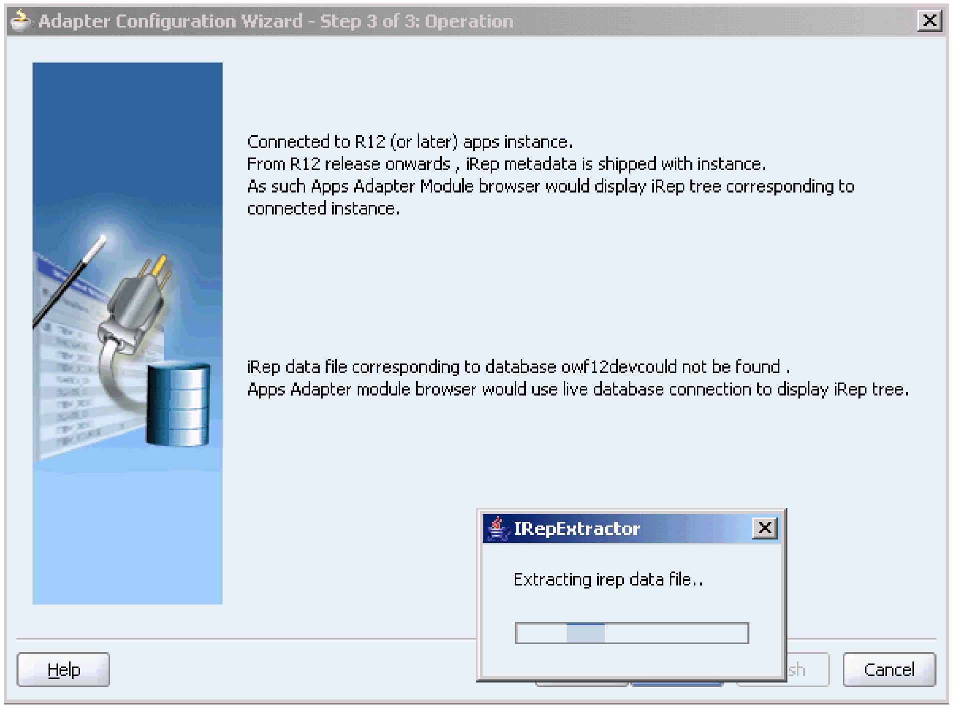
   For Oracle E-Business Suite Release 12:

   If you are connecting to Oracle E-Business Suite Release 12, then the **IREP File not present** dialog box appears indicating that OracleAS Adapter for Oracle Applications could not find the Oracle Integration Repository data file corresponding to the database you are connecting in your workspace. Absence of the data file would make browsing or searching of Integration Repository tree considerably slow. You can choose to extract the data file and create a local copy of the Integration Repository data file. Once it is created successfully, OracleAS Adapter for Oracle Applications will pick it up automatically next time and retrieve data from your local Integration Repository.

   You can select one of the following options:

   - Click **Yes** to extract the Integration Repository data file.

Extracting Integration Repository Data File



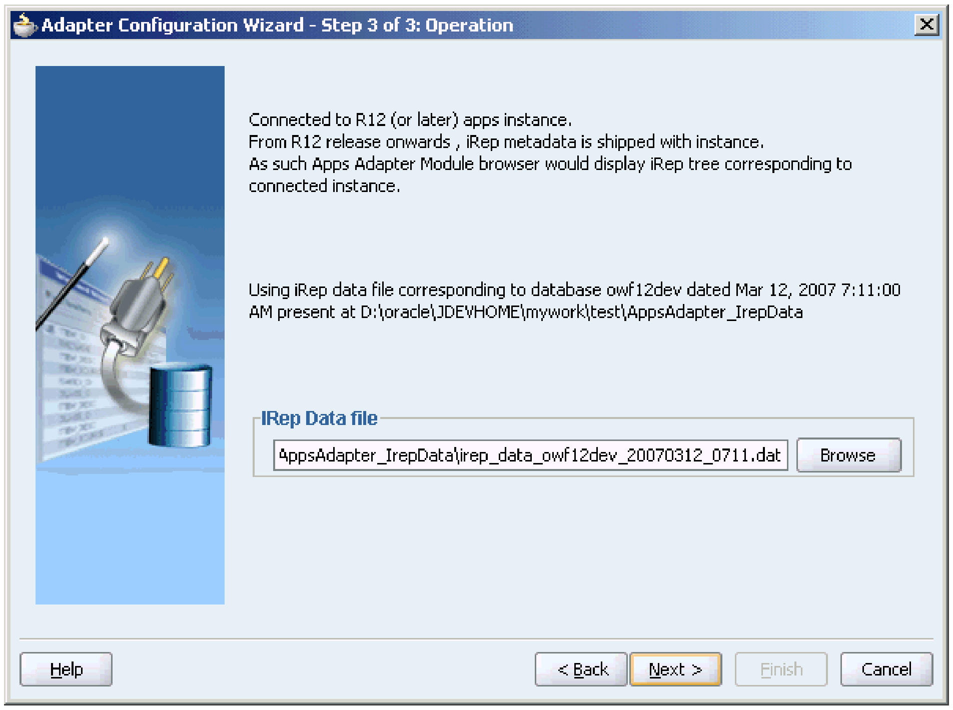
Connected to R12 (or later) apps instance.
From R12 release onwards , iRep metadata is shipped with instance.
As such Apps Adapter Module browser would display iRep tree corresponding to connected instance.

iRep data file corresponding to database owf12devcould not be found .
Apps Adapter module browser would use live database connection to display iRep tree.

Extracting irep data file..

After the system successfully creates a local copy of the Integration Repository data file, next time when you connect to the database, you will find the **IRep Data File** field appears in the Operation dialog box indicating where your local copy exists with the creation date and time as part of the file name.

Using the Local Integration Repository Data File



Connected to R12 (or later) apps instance.
From R12 release onwards , iRep metadata is shipped with instance.
As such Apps Adapter Module browser would display iRep tree corresponding to connected instance.

Using iRep data file corresponding to database owf12dev dated Mar 12, 2007 7:11:00 AM present at D:\oracle\JDEVHOME\mywork\test\AppsAdapter_IrepData

IRep Data file: AppsAdapter_IrepData\irep_data_owf12dev_20070312_0711.dat

- Click **No** to query the Integration Repository data file from the live database you are connecting to display the Integration Repository tree.

  Note: It is highly recommended that you create a local copy of the Integration Repository data file so that OracleAS Adapter for Oracle Applications will query the data next time from the local copy in your workspace to enhance the performance.

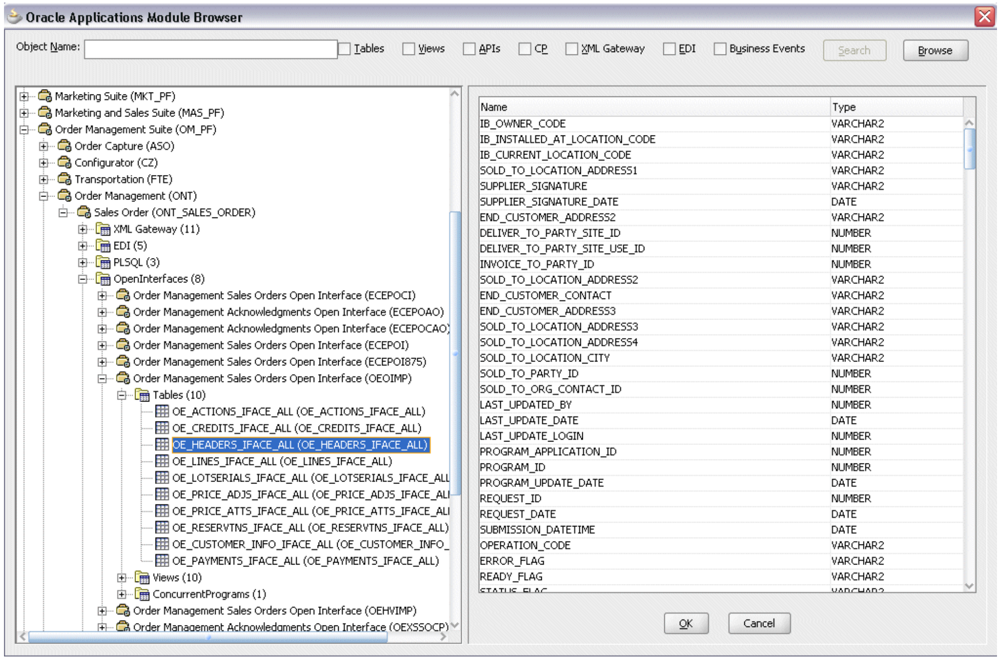
For Oracle E-Business Suite pre-Release 11.5.10:

If you are connecting to a pre-11.5.10 Oracle E-Business Suite instance, you must select the interface type in the Adapter Configuration Wizard. Select **Tables/Views/APIs/Concurrent Programs** to proceed.

Click **Add** to open the Oracle Applications Module Browser.

8. The Oracle Applications Module Browser combines interface data from Integration Repository with information about the additional interfaces supported by Adapter for Oracle Applications, organized in a tree hierarchy.

Adding a Table from the Oracle Applications Module Browser



- Marketing Suite (MKT_PF)
- Marketing and Sales Suite (MAS_PF)
- Order Management Suite (OM_PF)
  - Order Capture (ASO)
  - Configurator (CZ)
  - Transportation (FTE)
  - Order Management (ONT)
    - Sales Order (ONT_SALES_ORDER)
      - XML Gateway (11)
      - EDI (5)
      - PLSQL (3)
      - OpenInterfaces (8)
        - Order Management Sales Orders Open Interface (ECEPOCI)
        - Order Management Acknowledgments Open Interface (ECEPOAO)
        - Order Management Acknowledgments Open Interface (ECEPOCAO)
        - Order Management Sales Orders Open Interface (ECEPOI)
        - Order Management Sales Orders Open Interface (ECEPOI875)
        - Order Management Sales Orders Open Interface (OEOIMP)
          - Tables (10)
            - OE_ACTIONS_IFACE_ALL (OE_ACTIONS_IFACE_ALL)
            - OE_CREDITS_IFACE_ALL (OE_CREDITS_IFACE_ALL)
            - OE_HEADERS_IFACE_ALL (OE_HEADERS_IFACE_ALL)
            - OE_LINES_IFACE_ALL (OE_LINES_IFACE_ALL)
            - OE_LOTSERIALS_IFACE_ALL (OE_LOTSERIALS_IFACE_ALL
            - OE_PRICE_ADJS_IFACE_ALL (OE_PRICE_ADJS_IFACE_ALI
            - OE_PRICE_ATTS_IFACE_ALL (OE_PRICE_ATTS_IFACE_ALI
            - OE_RESERVTNS_IFACE_ALL (OE_RESERVTNS_IFACE_ALL)
            - OE_CUSTOMER_INFO_IFACE_ALL (OE_CUSTOMER_INFO_
            - OE_PAYMENTS_IFACE_ALL (OE_PAYMENTS_IFACE_ALL)
          - Views (10)
          - ConcurrentPrograms (1)
        - Order Management Sales Orders Open Interface (OEHVIMP)
        - Order Management Acknowledgments Open Interface (OEXSSOCP)

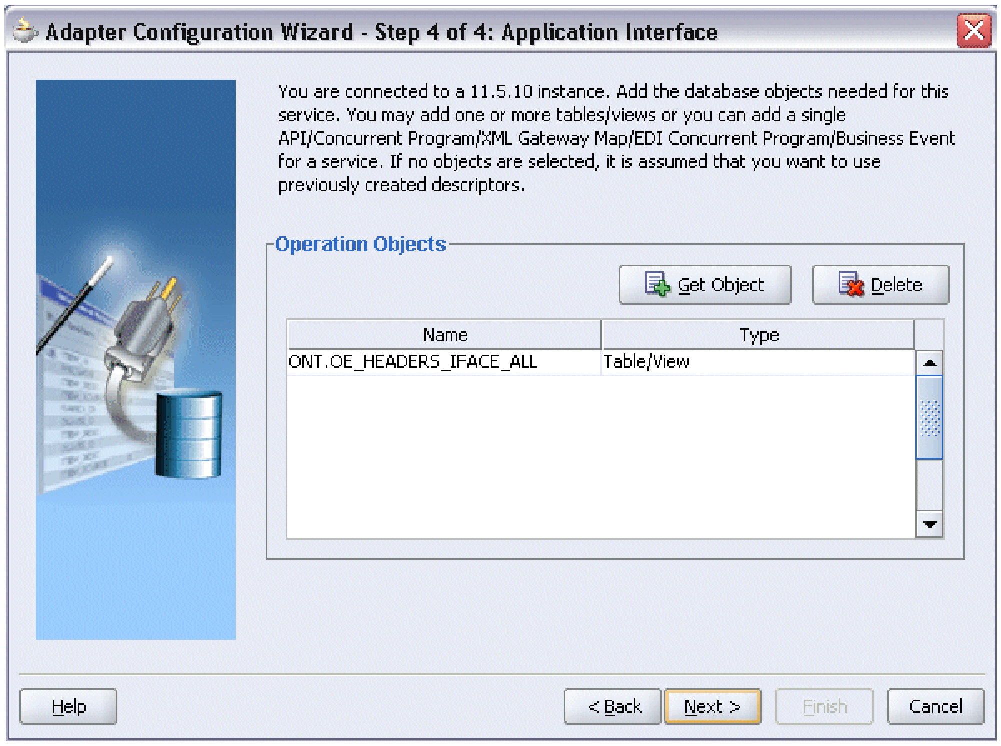
Name	Type
IB_OWNER_CODE	VARCHAR2
IB_INSTALLED_AT_LOCATION_CODE	VARCHAR2
IB_CURRENT_LOCATION_CODE	VARCHAR2
SOLD_TO_LOCATION_ADDRESS1	VARCHAR2
SUPPLIER_SIGNATURE	VARCHAR2
SUPPLIER_SIGNATURE_DATE	DATE
END_CUSTOMER_ADDRESS2	VARCHAR2
DELIVER_TO_PARTY_SITE_ID	NUMBER
DELIVER_TO_PARTY_SITE_USE_ID	NUMBER
INVOICE_TO_PARTY_ID	NUMBER
SOLD_TO_LOCATION_ADDRESS2	VARCHAR2
END_CUSTOMER_CONTACT	VARCHAR2
END_CUSTOMER_ADDRESS3	VARCHAR2
SOLD_TO_LOCATION_ADDRESS3	VARCHAR2
SOLD_TO_LOCATION_ADDRESS4	VARCHAR2
SOLD_TO_LOCATION_CITY	VARCHAR2
SOLD_TO_PARTY_ID	NUMBER
SOLD_TO_ORG_CONTACT_ID	NUMBER
LAST_UPDATED_BY	NUMBER
LAST_UPDATE_DATE	DATE
LAST_UPDATE_LOGIN	NUMBER
PROGRAM_APPLICATION_ID	NUMBER
PROGRAM_ID	NUMBER
PROGRAM_UPDATE_DATE	DATE
REQUEST_ID	NUMBER
REQUEST_DATE	DATE
SUBMISSION_DATETIME	DATE
OPERATION_CODE	VARCHAR2
ERROR_FLAG	VARCHAR2
READY_FLAG	VARCHAR2

Note: Oracle Applications Module Browser includes the various product families that are available in Oracle Applications. For example, Applications Technology or Order Management Suite are product families in Oracle Applications. The product families contain the individual products. For example, Order Management Suite contains the Order Management product. The product contains the business entities associated with the product. For example, the Order Management product contains the Sales Order business entity. Business entities contain the various application modules that are exposed for integration. These modules are grouped according to the interface they provide. Tables can be found under the OpenInterfaces category.

Navigate to *Order Management Suite (OM_PF)* > *Order Management (ONT)* > *Sales Order (ONT_SALES_ORDER)* > *OpenInterfaces* >*Order Management Sales Orders Open Interface (OEOIMP)* > *Tables* to select `OE_HEADERS_IFACE_ALL`.

Click **OK**. The Application Interface dialog appears with the selected open table.

Application Interface Dialog



Adapter Configuration Wizard - Step 4 of 4: Application Interface

You are connected to a 11.5.10 instance. Add the database objects needed for this service. You may add one or more tables/views or you can add a single API/Concurrent Program/XML Gateway Map/EDI Concurrent Program/Business Event for a service. If no objects are selected, it is assumed that you want to use previously created descriptors.

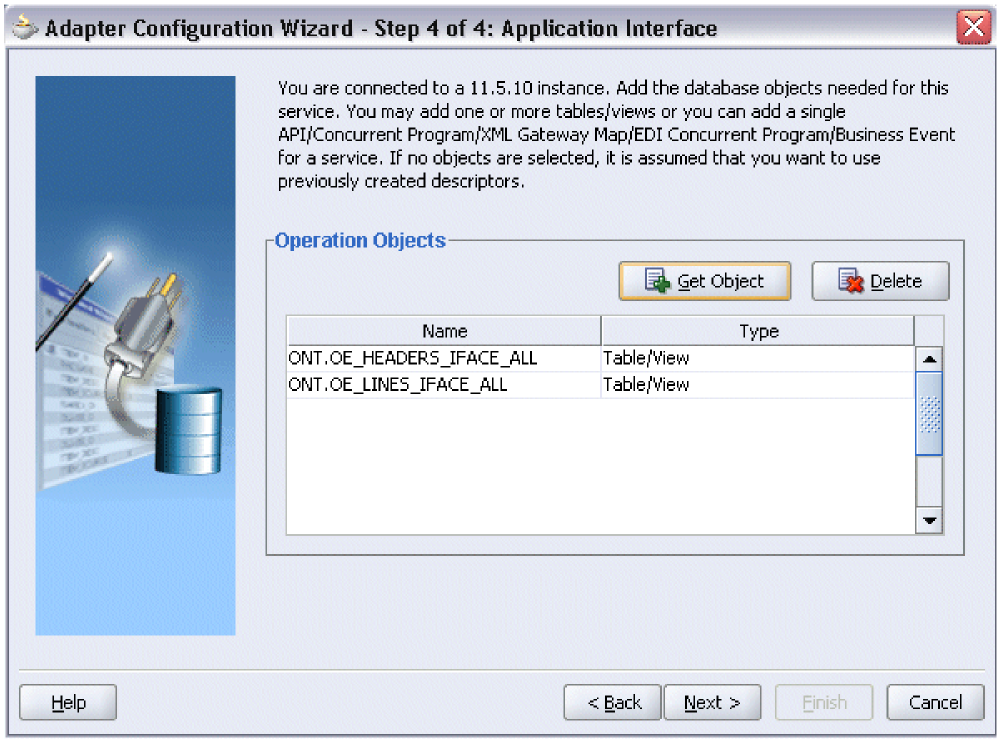
Operation Objects

Name	Type
ONT.OE_HEADERS_IFACE_ALL	Table/View

9. Click **Get Object** to open the Oracle Applications Module Browser again to select another open interface table `OE_LINES_IFACE_ALL` using the same navigation path *Order Management Suite (OM_PF) > Order Management (ONT) > Sales Order (ONT_SALES_ORDER) > OpenInterfaces >Order Management Sales Orders Open Interface (OEOIMP) > Tables*.

   Click **OK**. The Application Interface dialog appears with the two selected tables.

Application Interface Dialog



Adapter Configuration Wizard - Step 4 of 4: Application Interface

You are connected to a 11.5.10 instance. Add the database objects needed for this service. You may add one or more tables/views or you can add a single API/Concurrent Program/XML Gateway Map/EDI Concurrent Program/Business Event for a service. If no objects are selected, it is assumed that you want to use previously created descriptors.

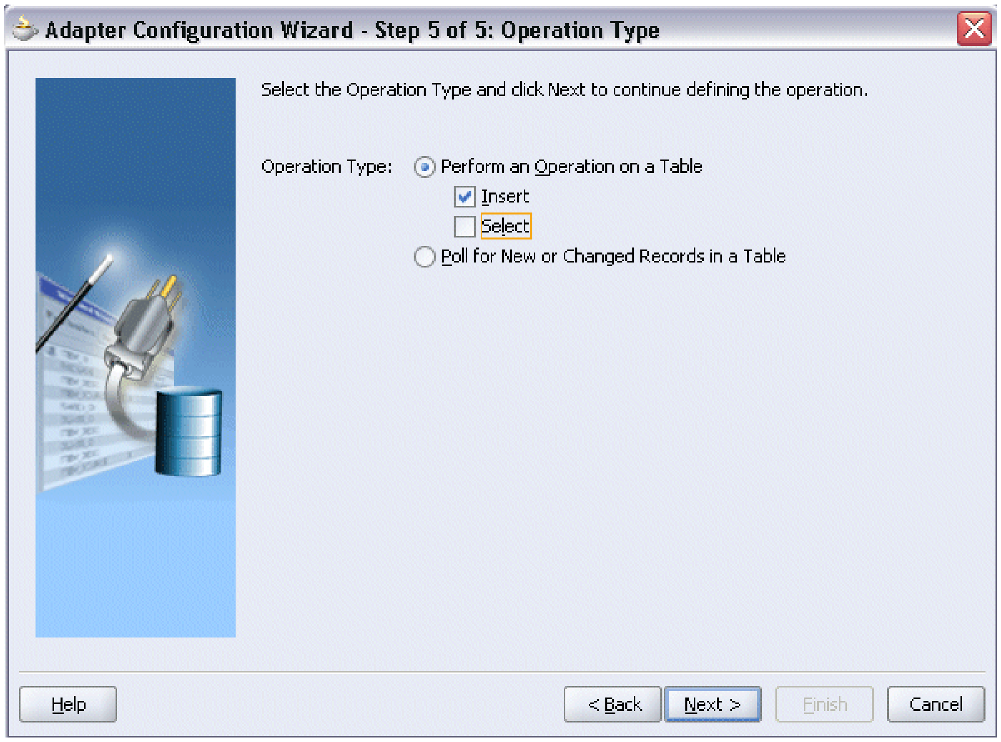
Operation Objects

Name	Type
ONT.OE_HEADERS_IFACE_ALL	Table/View
ONT.OE_LINES_IFACE_ALL	Table/View

10. Click **Next**.

11. In the Operation Type dialog, select **Perform an Operation on a Table**, and then select the **Insert** check box.

Selecting the Type of Operation



Select the Operation Type and click Next to continue defining the operation.

Operation Type: [✓ Perform an Operation on a Table]

[✓ Insert]

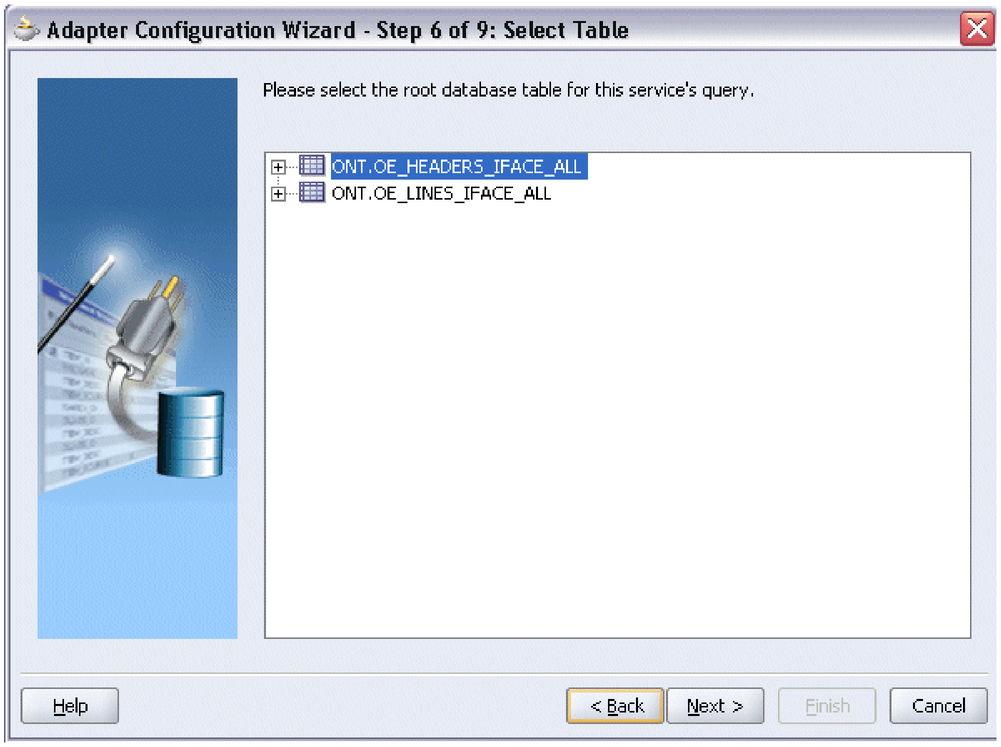
Select

Poll for New or Changed Records in a Table

> **Note:** You can only insert data into the interface tables even though the select option is enabled.

12. Click **Next**. The Select Table dialog box appears.
13. Select OE_HEADERS_IFACE_ALL as the root database table in the Select Table dialog box, which displays the tables that have been previously imported in this JDeveloper project (including tables that were imported for other partner links). This enables you to reuse configured table definitions in multiple partner links.

Selecting a Root Table


Adapter Configuration Wizard - Step 6 of 9: Select Table

Please select the root database table for this service's query.

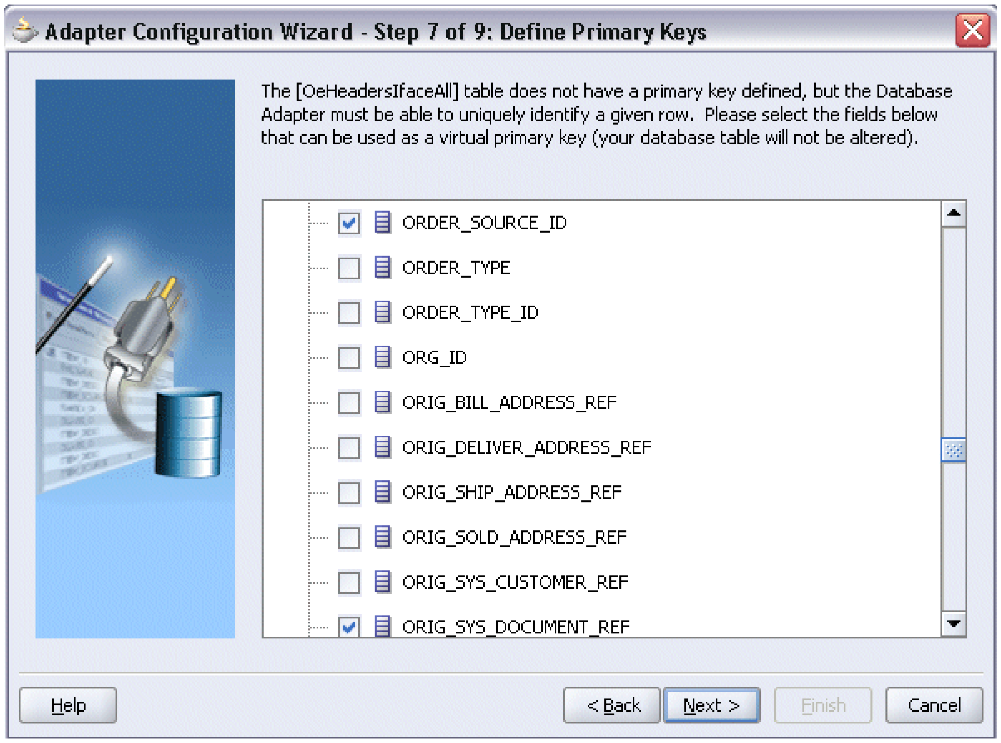
ONT.OE_HEADERS_IFACE_ALL
ONT.OE_LINES_IFACE_ALL

14. Click **Next**. The Define Primary Keys dialog box appears.

15. Select the following primary keys for the OE_HEADERS_IFACE_ALL table:

    - ORDER_SOURCE_ID
    - ORIG_SYS_DOCUMENT_REF

Defining Primary Keys Page for OE_HEADERS_IFACE_ALL


Adapter Configuration Wizard - Step 7 of 9: Define Primary Keys

The [OeHeadersIfaceAll] table does not have a primary key defined, but the Database Adapter must be able to uniquely identify a given row. Please select the fields below that can be used as a virtual primary key (your database table will not be altered).

- [✓] ORDER_SOURCE_ID
- [ ] ORDER_TYPE
- [ ] ORDER_TYPE_ID
- [ ] ORG_ID
- [ ] ORIG_BILL_ADDRESS_REF
- [ ] ORIG_DELIVER_ADDRESS_REF
- [ ] ORIG_SHIP_ADDRESS_REF
- [ ] ORIG_SOLD_ADDRESS_REF
- [ ] ORIG_SYS_CUSTOMER_REF
- [✓] ORIG_SYS_DOCUMENT_REF

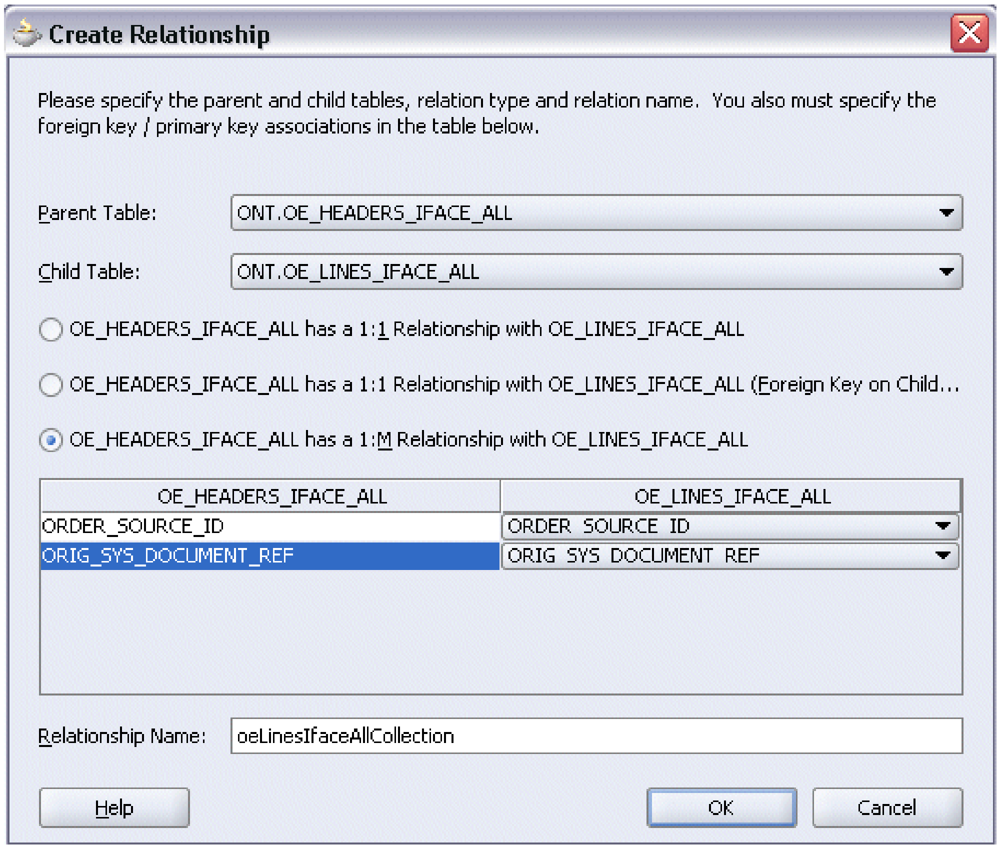
Click **Next**.

Select the same primary keys for the OE_LINES_IFACE_ALL table.

16. Click **Next**. The Relationships dialog box appears.

17. Click **Create** to define a new relationship. The Create Relationship dialog box appears.

Defining Relationships



Create Relationship

Please specify the parent and child tables, relation type and relation name. You also must specify the foreign key / primary key associations in the table below.

Parent Table: ONT.OE_HEADERS_IFACE_ALL

Child Table: ONT.OE_LINES_IFACE_ALL

( ) OE_HEADERS_IFACE_ALL has a 1:1 Relationship with OE_LINES_IFACE_ALL

( ) OE_HEADERS_IFACE_ALL has a 1:1 Relationship with OE_LINES_IFACE_ALL (Foreign Key on Child...

[✓ OE_HEADERS_IFACE_ALL has a 1:M Relationship with OE_LINES_IFACE_ALL]

OE_HEADERS_IFACE_ALL	OE_LINES_IFACE_ALL
ORDER_SOURCE_ID	ORDER_SOURCE_ID
ORIG_SYS_DOCUMENT_REF	ORIG_SYS_DOCUMENT_REF

Relationship Name: oeLinesIfaceAllCollection

18. Enter the following information to define the relationship between the header and the detail table:

- Select the OE_HEADERS_IFACE_ALL as the parent table and OE_LINES_IFACE_ALL as the child table.

- Select the mapping type: OE_HEADERS_IFACE_ALL has a 1:M Relationship with OE_LINES_IFACE_ALL

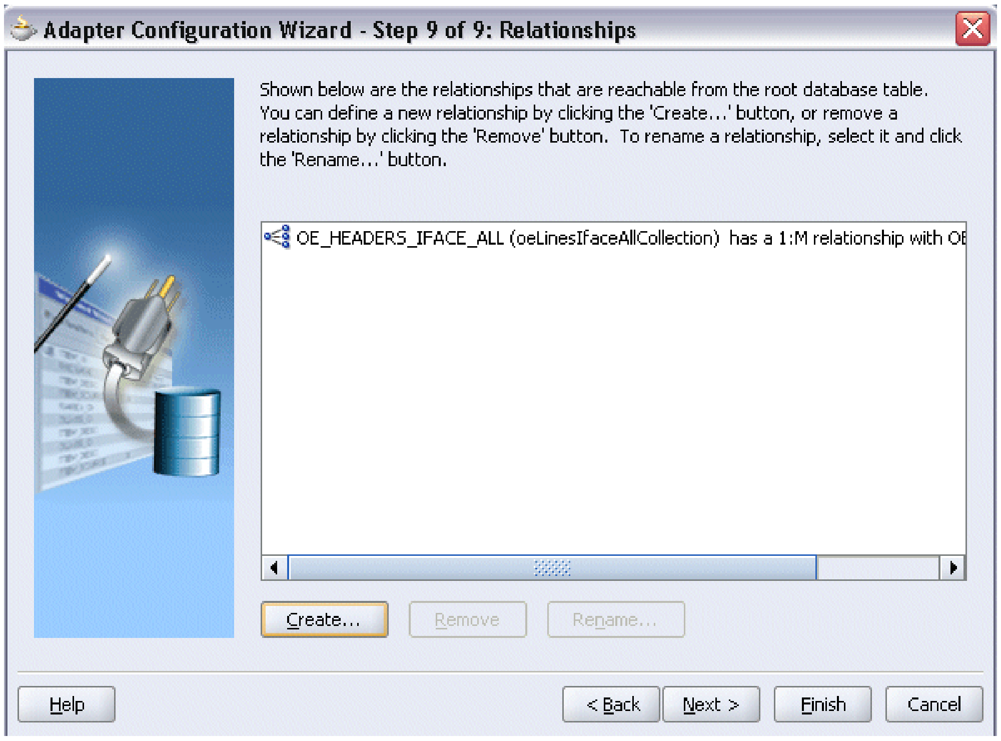
  Note: If foreign key constraints between tables already exist in the database, then two relationships are created automatically while importing tables. One of the relationships is 1:M relationship from the source table, which is the table containing the foreign key constraints, to the target table. The other relationship is a 1:1 back pointer from the target table to the source table.

- Select the **Private Owned** check box.

- Associate the foreign key fields with the primary key fields:
    - OE_HEADERS_IFACE_ALL.ORDER_SOURCE_ID: ORDER_SOURCE_ID
    - OE_HEADERS_IFACE_ALL.ORIG_SYS_DOCUMENT_REF: ORIG_SYS_DOCUMENT_REF
- The Relationship Name field is populated automatically by default. You can optionally specify a new name for the relationship you are creating.

19. Click **OK**. The newly created relationship information appears in the Relationships dialog.

Relationships Dialog

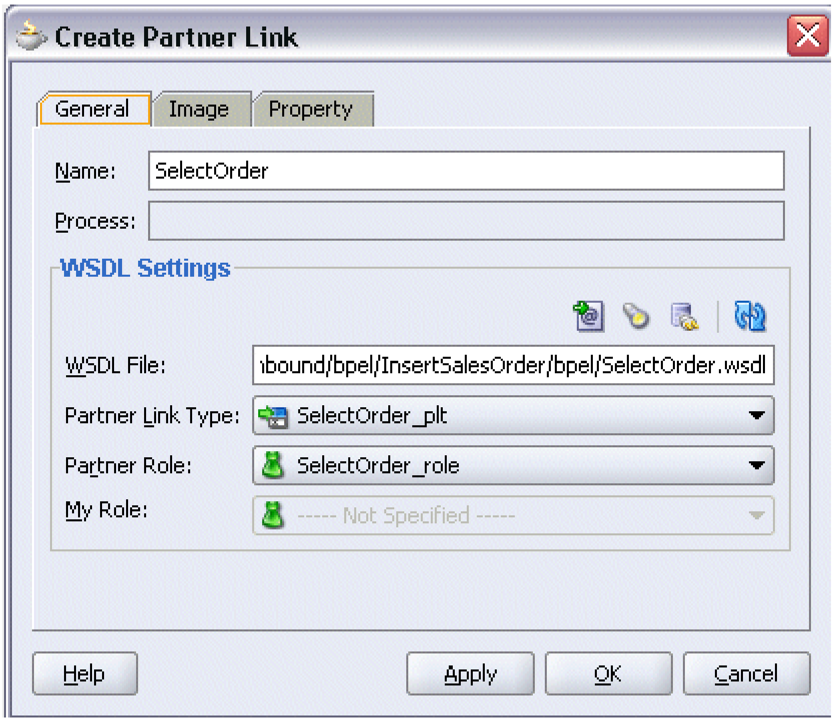


Shown below are the relationships that are reachable from the root database table. You can define a new relationship by clicking the 'Create...' button, or remove a relationship by clicking the 'Remove' button. To rename a relationship, select it and click the 'Rename...' button.

OE_HEADERS_IFACE_ALL (oeLinesIfaceAllCollection) has a 1:M relationship with OE

20. Click **Next** and then click **Finish** to complete the process.

    The wizard generates the WSDL file corresponding to the selected interface. This WSDL file is now available for the partner link.


Create Partner Link

General | Image | Property

Name: SelectOrder

Process:

WSDL Settings

WSDL File: ıbound/bpel/InsertSalesOrder/bpel/SelectOrder.wsdl

Partner Link Type: SelectOrder_plt

Partner Role: SelectOrder_role

My Role: ----- Not Specified -----

Help | Apply | OK | Cancel

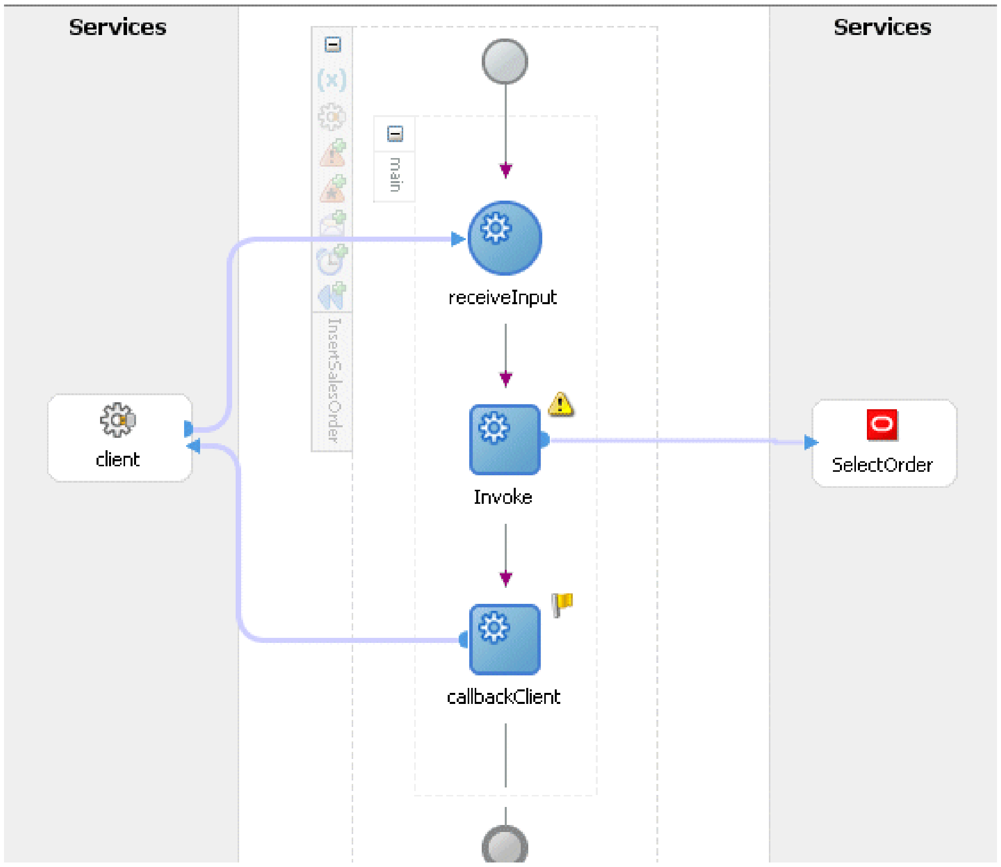
21. Click **Apply** and then **OK**. The partner link is created with the required WSDL settings.

Configuring the Invoke Activity

To configure the Invoke activity:

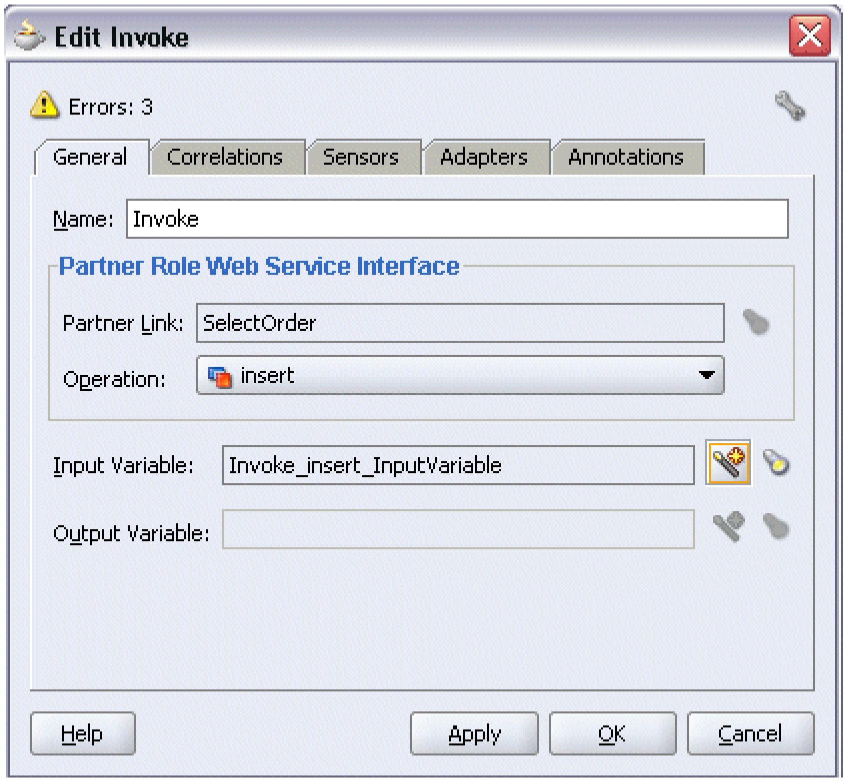
1. In JDeveloper BPEL Designer, drag and drop the **Invoke** activity from the Component palette into the process diagram between the Receive and Callback activities.

Adding an Invoke Activity



2. Link the Invoke activity to the `SelectOrder` service. The Edit Invoke dialog appears.

Configuring the Invoke Activity



Edit Invoke

Errors: 3

Name: Invoke

Partner Role Web Service Interface

Partner Link: SelectOrder

Operation: insert

Input Variable: Invoke_insert_InputVariable

Output Variable:

3. Click the **Create** icon next to the **Input Variable** field. Enter a descriptive name for the variable. You can also accept the default name. Select **Global Variable** and click **OK**.

4. In the Invoke dialog box, click **Apply** and then **OK** to complete the configuration of the Invoke activity.

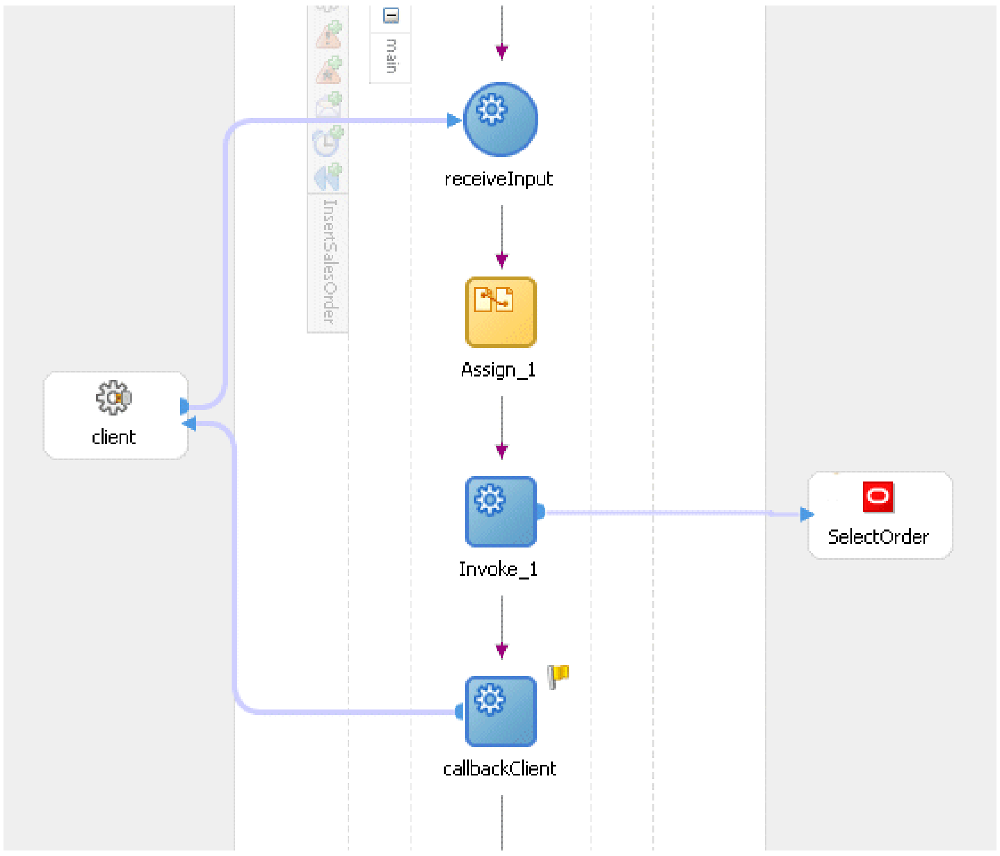
Configuring the Assign Activity

The next task is to add an Assign activity to provide values to the input variables.

To configure the Assign activity:

1. Drag and drop the Assign activity to the process map. The Assign activity must be dropped between the Receive and Invoke activities.

Adding an Assign Activity



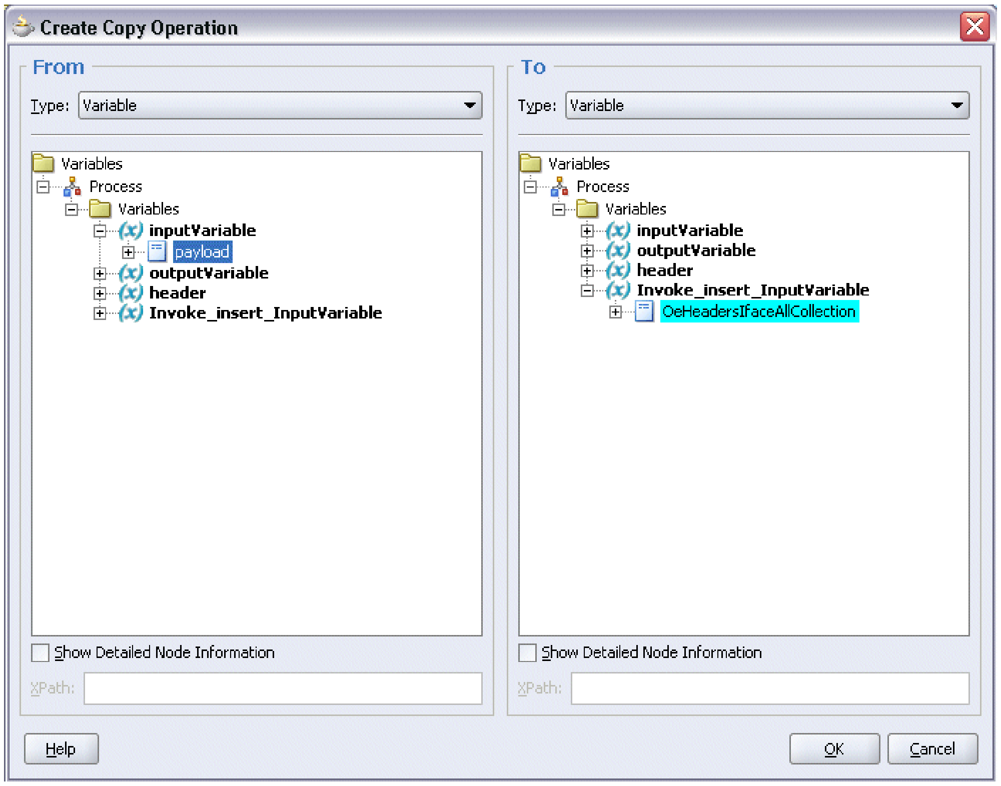
2. Double-click the **Assign** activity in the process map. The Assign dialog box appears.

3. Click **Create** and select **Copy Operation...** to open the Create Copy Operation dialog box.

4. In the From navigation tree, select type Variable. Navigate to **Variable > Process > Variables > inputVariable** and select **Payload**.

   In the To navigation tree, select type Variable. Navigate to **Variable > Process > Variables > Invoke_insert_InputVariable** and select **OeHeadersIfaceAllCollection**
   .

Assigning Values to Input Variables



5. After assigning values to the input variables, click **OK**.

6. Click **Apply** and **OK** to complete the Assign activity.

Run-Time Tasks for Interface Tables

After designing the BPEL process, the next step is to deploy, run, and monitor it.

1. Deploy the BPEL process, page 7-23

2. Test the BPEL process, page 7-24

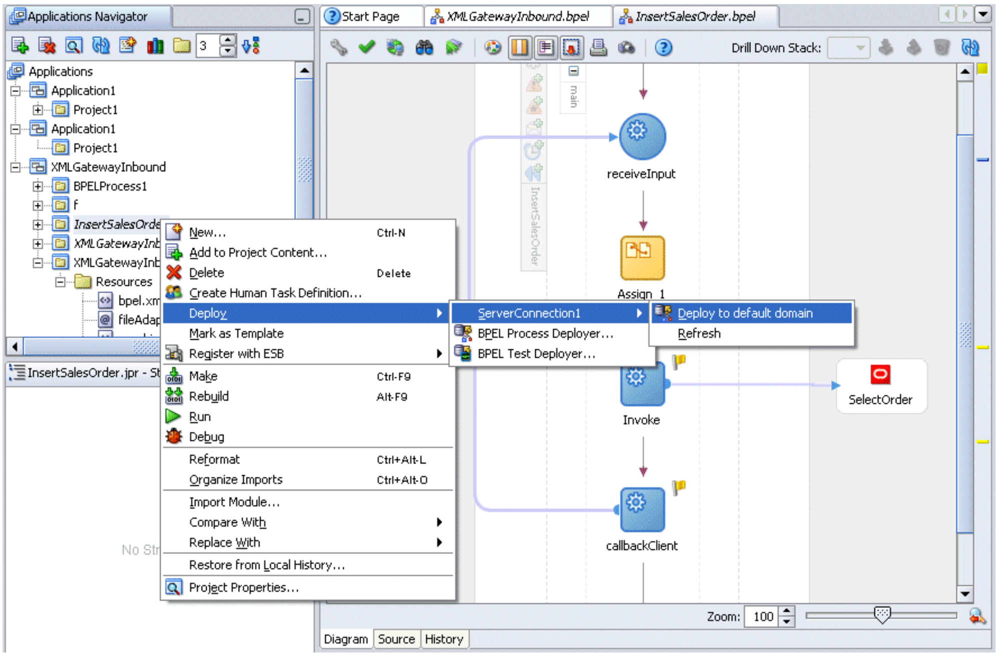
Deploying the BPEL Process

You need to deploy the BPEL process to a BPEL server before you can run it. The BPEL process is first compiled, and then deployed to the BPEL server.

To deploy the BPEL process:

1. Select the BPEL project in the Applications window.
2. Right-click the project name. Select **Deploy > [Server Connection] > Deploy to Default Domain** from the menu that appears.

Deploying the BPEL Process



3. The BPEL process is compiled and deployed. You can check the progress in the Messages window.

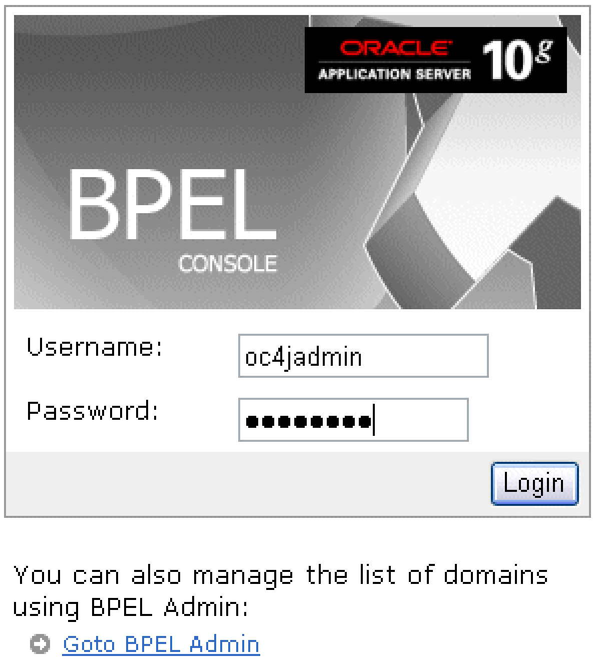
Testing the BPEL Process

Once the BPEL process is deployed, it can be seen in the BPEL console. You can manage and monitor the process from the BPEL console. You can also test the process and the integration interface by manually initiating the process.

To manually initiate and monitor the BPEL process:

1. Navigate to Oracle Application Server 10*g* BPEL Console (`http://<soaSuiteServerHostName>:<port>/BPELConsole`).

   The BPEL Console login page appears.



Username: oc4jadmin

Password: •••••••••

You can also manage the list of domains using BPEL Admin:

2. Enter the username and password and click **Login**.

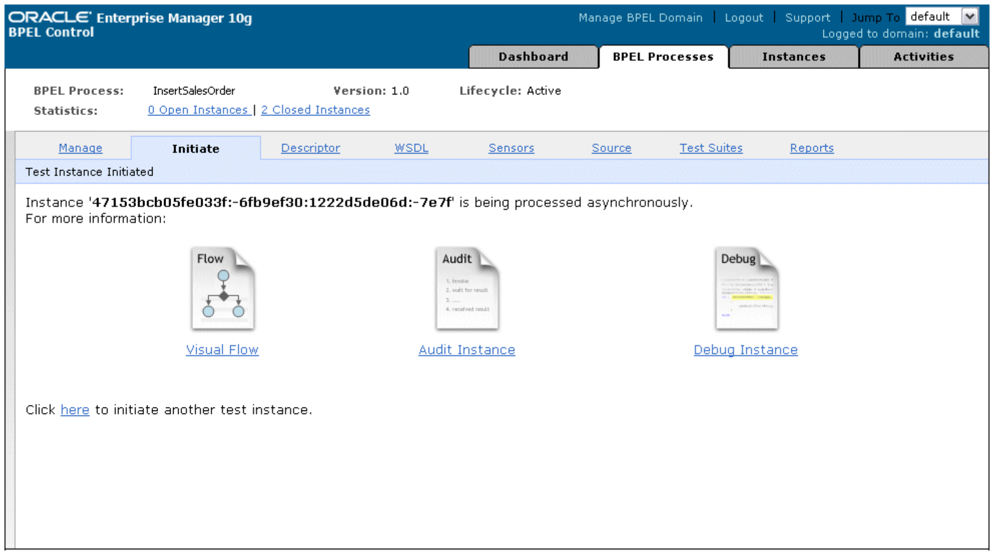
3. The Oracle Enterprise Manager 10*g* BPEL Control appears. The list of deployed processes is shown under Deployed BPEL Processes.

4. Click the BPEL process that you want to initiate. The Initiate page appears. Enter the input values required by the process.

5. Click **Post XML Message** to initiate the process.

6. The BPEL process is now initiated. You can check the process flow by clicking the **Visual Flow** icon.

BPEL Console Initiate Page


ORACLE Enterprise Manager 10g
BPEL Control
Manage BPEL Domain | Logout | Support | Jump To default
Logged to domain: default
Dashboard
BPEL Processes
Instances
Activities

BPEL Process: InsertSalesOrder Version: 1.0 Lifecycle: Active
Statistics: 0 Open Instances | 2 Closed Instances

Test Instance Initiated

Instance '47153bcb05fe033f:-6fb9ef30:1222d5de06d:-7e7f' is being processed asynchronously.
For more information:

Click here to initiate another test instance.

7. The audit trail provides information about the steps that have been executed. You can check the audit trail by clicking the **Audit Instance** icon.

   Note: To confirm that the records have been written into the open interface tables, you can write the SQL `SELECT` statements and fetch the results showing the latest records inserted into the open interface tables. Alternatively, open the forms of the module for which the record has been inserted and then check for changes in the form.

Design-Time Tasks for Interface Views

This section describes the steps to configure OracleAS Adapter for Oracle Applications using the Adapter Configuration Wizard in Oracle JDeveloper.

Prerequisites to Configure Interface Views

You need to define a uniqueness criteria, which could be a single key or composite keys.

Following is a list of the procedures required to accomplish the design-time tasks.

1. Create a new BPEL project, page 7-27
2. Add a partner link, page 7-29
3. Configure the Invoke activity, page 7-42

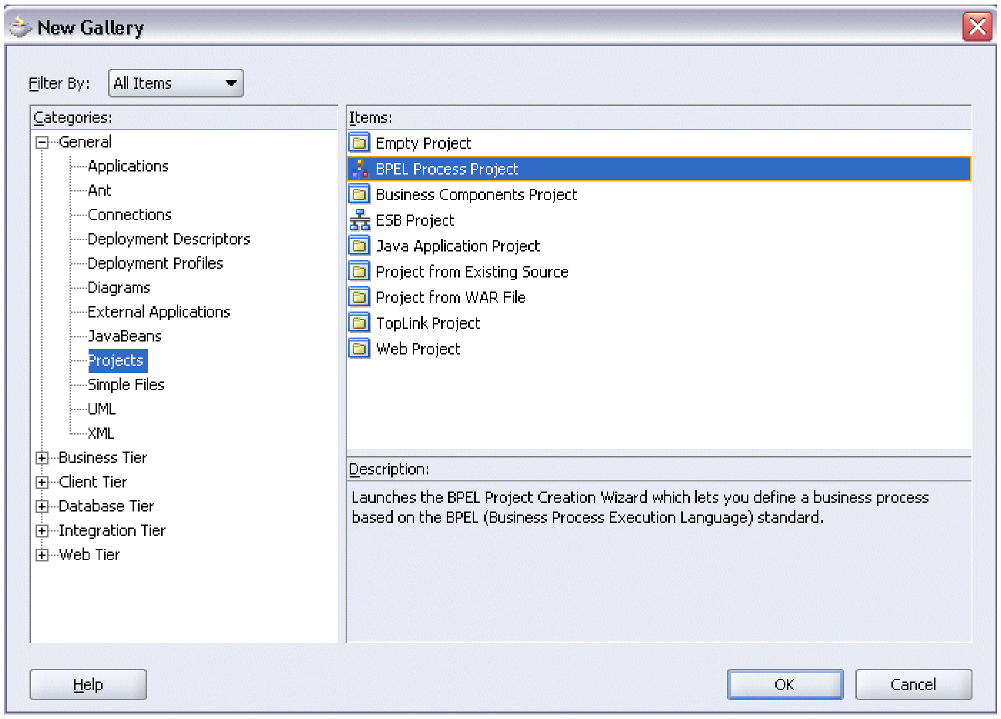
4. Configure the Assign activity, page 7-43

Creating a New BPEL Project

To create a new BPEL project:

1. Open JDeveloper BPEL Designer.
2. From the **File** menu, select **New**. The New Gallery dialog box appears.
3. Select **All Items** from the **Filter By** list. This displays a list of available categories.
4. Expand the **General** node, and then select **Projects**.
5. Select **BPEL Process Project** from the **Items** list.

Creating a New BPEL Process Project



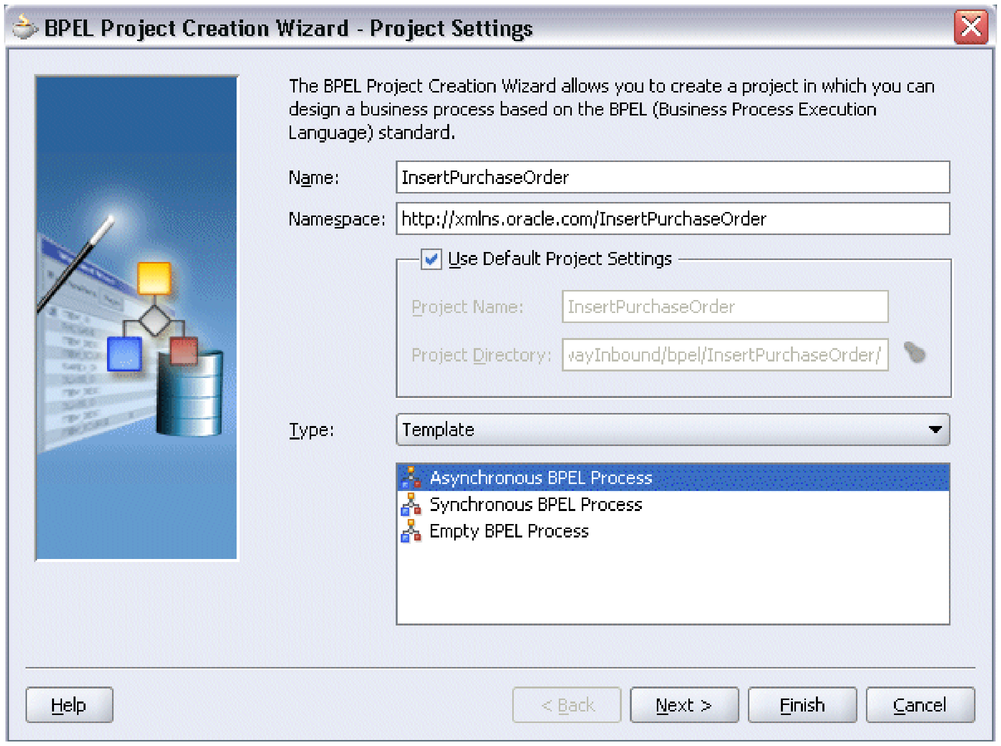
Launches the BPEL Project Creation Wizard which lets you define a business process based on the BPEL (Business Process Execution Language) standard.

6. Click **OK**. The BPEL Project Creation Wizard - Project Settings dialog box appears.
7. In the **Name** field, enter a descriptive name. For example, enter `SelectCustomer`.

   Keep the default selection **Use Default Project Settings** unchanged.
8. Keep the default selection **Template** as the Type field. Select **Asynchronous BPEL**

Process as the BPEL process type.

Specifying a Name for the New BPEL Process Project


BPEL Project Creation Wizard - Project Settings

The BPEL Project Creation Wizard allows you to create a project in which you can design a business process based on the BPEL (Business Process Execution Language) standard.

Name: InsertPurchaseOrder

Namespace: http://xmlns.oracle.com/InsertPurchaseOrder

☑ Use Default Project Settings

Project Name: InsertPurchaseOrder

Project Directory: vayInbound/bpel/InsertPurchaseOrder/

Type: Template

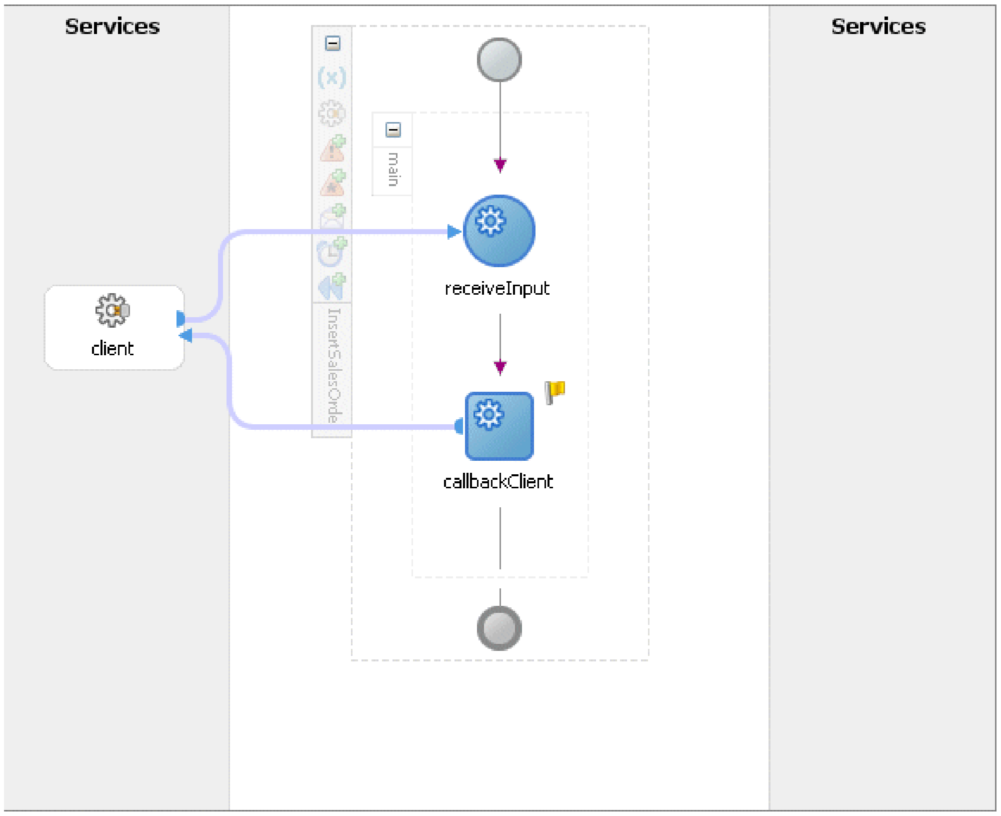
- Asynchronous BPEL Process
- Synchronous BPEL Process
- Empty BPEL Process

9. Click **OK**.

   A new asynchronous BPEL process is created with the receiveInput and callbackClient activities.

   The required source files including `bpel.xml`, `InsertPurchaseOrder.xml`, and `InsertPurchaseOrder.wsdl` are created.

New BPEL Process



Adding a Partner Link

This section describes how to add a partner link to the BPEL process. A BPEL partner link defines the link name, type, and the role of the BPEL process that interacts with the partner service.

To add a partner link:

1. In JDeveloper BPEL Designer, click **Services** in the Component palette.

   Drag and drop **Oracle Applications** icon into the border area of the process diagram.

   The Adapter Configuration Wizard appears.

2. Click **Next**. The Service Name dialog box appears.

3. Enter the following information:

   1. In the **Service Name** field, enter a service name, such as `OrderAck`.

2. In the **Description** field, enter a description for the service. This is an optional field.

4. Click **Next**. The Service Connection dialog box appears.

5. You can perform either one of the following options for your database connection:

   Note: You need to connect to the database where Oracle E-Business Suite is running.

   - You can create a new database connection by clicking the **New** icon.

     Detailed instructions on how to create a new database connection, see Creating a New Database Connection, page 4-12.

   - You can select an existing database connection that you have configured earlier from the **Connection** drop-down list.

     The selected database connection information appears in the Service Connection dialog box. The JNDI (Java Naming and Directory Interface) name corresponding to the selected database connection also appears automatically in the **Database Server JNDI Name** field. Alternatively, you can specify a JNDI name.

     Note: When you specify a JNDI name, the deployment descriptor of the Adapter for Oracle Applications must associate this JNDI name with configuration properties required by the adapter to access the database.

     The JNDI name acts as a placeholder for the connection used when your service is deployed to the BPEL server. This enables you to use different databases for development and later for production.

     Note: *Oracle Application Server Adapter Concepts* for understanding JNDI concepts.

6. Once you have created a new connection or selected an existing connection, you can add an interface view available in Oracle E-Business Suite.

7. Click **Next** in the Service Connection dialog box.

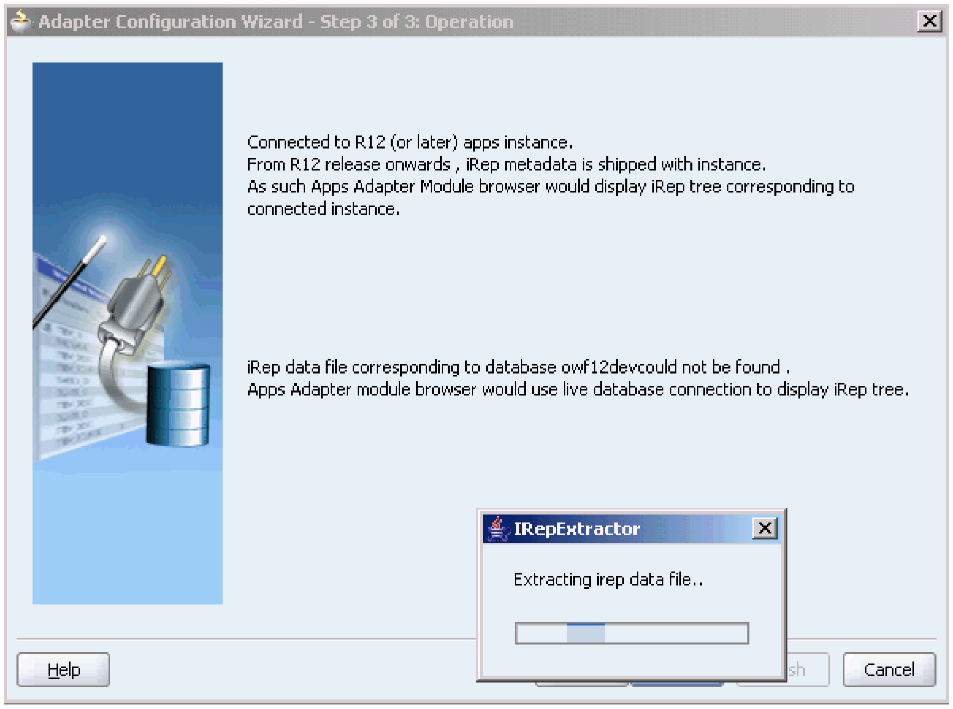
   For Oracle E-Business Suite Release 12:

   If you are connecting to Oracle E-Business Suite Release 12, then the **IREP File not present** dialog box appears indicating that OracleAS Adapter for Oracle Applications could not find Oracle the Integration Repository data file corresponding to the database you are connecting in your workspace. Absence of the data file would make browsing or searching of Integration Repository tree considerably slow. You can choose to extract the data file and create a local copy of the Integration Repository data file. Once it is created successfully, OracleAS Adapter for Oracle Applications will pick it up automatically next time and retrieve data from your local Integration Repository.

You can select one of the following options:

- Click **Yes** to extract the Integration Repository data file.

Extracting Integration Repository Data File



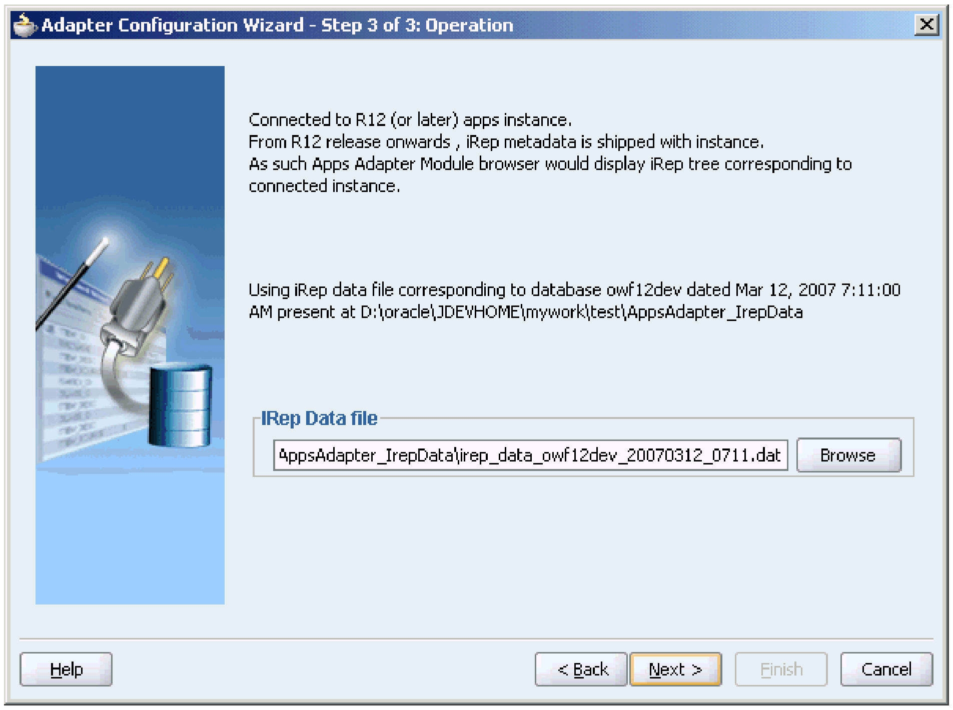
Connected to R12 (or later) apps instance.
From R12 release onwards , iRep metadata is shipped with instance.
As such Apps Adapter Module browser would display iRep tree corresponding to connected instance.

iRep data file corresponding to database owf12devcould not be found .
Apps Adapter module browser would use live database connection to display iRep tree.

Extracting irep data file..

After the system successfully creates a local copy of the Integration Repository data file, next time when you connect to the database, you will find the **IRep Data File** field appears in the Operation dialog box indicating where your local copy exists with the creation date and time as part of the file name.

Using the Local Integration Repository Data File



Connected to R12 (or later) apps instance.
From R12 release onwards , iRep metadata is shipped with instance.
As such Apps Adapter Module browser would display iRep tree corresponding to connected instance.

Using iRep data file corresponding to database owf12dev dated Mar 12, 2007 7:11:00 AM present at D:\oracle\JDEVHOME\mywork\test\AppsAdapter_IrepData

IRep Data file: AppsAdapter_IrepData\irep_data_owf12dev_20070312_0711.dat

- Click **No** to query the Integration Repository data file from the live database you are connecting to display the Integration Repository tree.

  Note: It is highly recommended that you create a local copy of the Integration Repository data file so that OracleAS Adapter for Oracle Applications will query the data next time from the local copy in your workspace to enhance the performance.

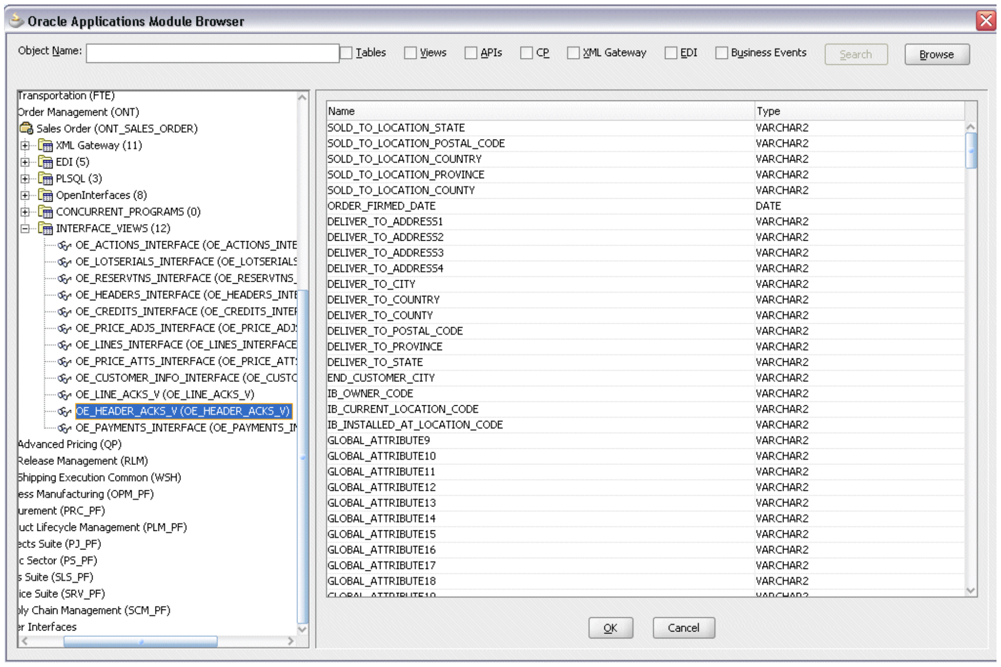
For Oracle E-Business Suite pre-Release 11.5.10:

If you are connecting to a pre-11.5.10 Oracle E-Business Suite instance, you must select the interface type in the Adapter Configuration Wizard. Select **Tables/Views/APIs/Concurrent Programs** to proceed.

Click **Add** to open the Oracle Applications Module Browser.

8. The Oracle Applications Module Browser combines interface data from Integration Repository with information about the additional interfaces supported by Adapter for Oracle Applications, organized in a tree hierarchy.

Selecting a View from the Oracle Applications Module Browser



```
Transportation (FTE)
Order Management (ONT)
 Sales Order (ONT_SALES_ORDER)
  XML Gateway (11)
  EDI (5)
  PLSQL (3)
  OpenInterfaces (8)
  CONCURRENT_PROGRAMS (0)
  INTERFACE_VIEWS (12)
    OE_ACTIONS_INTERFACE (OE_ACTIONS_INTE
    OE_LOTSERIALS_INTERFACE (OE_LOTSERIALS
    OE_RESERVTNS_INTERFACE (OE_RESERVTNS_
    OE_HEADERS_INTERFACE (OE_HEADERS_INTE
    OE_CREDITS_INTERFACE (OE_CREDITS_INTER
    OE_PRICE_ADJS_INTERFACE (OE_PRICE_ADJ
    OE_LINES_INTERFACE (OE_LINES_INTERFACE
    OE_PRICE_ATTS_INTERFACE (OE_PRICE_ATT
    OE_CUSTOMER_INFO_INTERFACE (OE_CUSTO
    OE_LINE_ACKS_V (OE_LINE_ACKS_V)
    OE_HEADER_ACKS_V (OE_HEADER_ACKS_V)
    OE_PAYMENTS_INTERFACE (OE_PAYMENTS_IN
Advanced Pricing (QP)
Release Management (RLM)
Shipping Execution Common (WSH)
ess Manufacturing (OPM_PF)
urement (PRC_PF)
uct Lifecycle Management (PLM_PF)
ects Suite (PJ_PF)
c Sector (PS_PF)
s Suite (SLS_PF)
ice Suite (SRV_PF)
ly Chain Management (SCM_PF)
r Interfaces
```

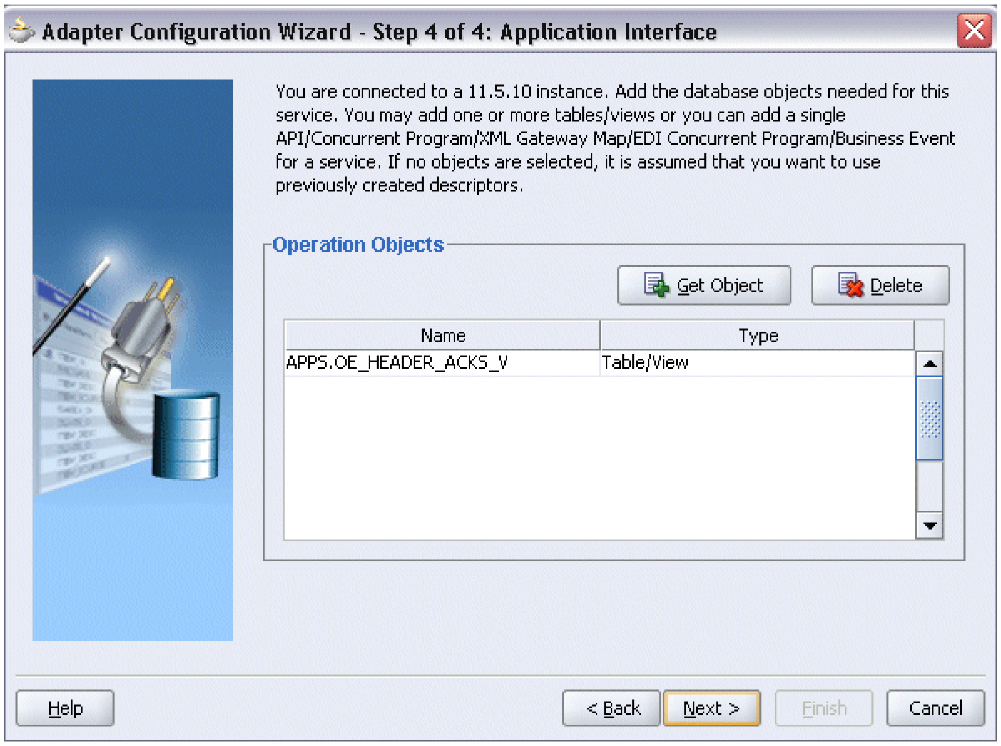
Name	Type
SOLD_TO_LOCATION_STATE	VARCHAR2
SOLD_TO_LOCATION_POSTAL_CODE	VARCHAR2
SOLD_TO_LOCATION_COUNTRY	VARCHAR2
SOLD_TO_LOCATION_PROVINCE	VARCHAR2
SOLD_TO_LOCATION_COUNTY	VARCHAR2
ORDER_FIRMED_DATE	DATE
DELIVER_TO_ADDRESS1	VARCHAR2
DELIVER_TO_ADDRESS2	VARCHAR2
DELIVER_TO_ADDRESS3	VARCHAR2
DELIVER_TO_ADDRESS4	VARCHAR2
DELIVER_TO_CITY	VARCHAR2
DELIVER_TO_COUNTRY	VARCHAR2
DELIVER_TO_COUNTY	VARCHAR2
DELIVER_TO_POSTAL_CODE	VARCHAR2
DELIVER_TO_PROVINCE	VARCHAR2
DELIVER_TO_STATE	VARCHAR2
END_CUSTOMER_CITY	VARCHAR2
IB_OWNER_CODE	VARCHAR2
IB_CURRENT_LOCATION_CODE	VARCHAR2
IB_INSTALLED_AT_LOCATION_CODE	VARCHAR2
GLOBAL_ATTRIBUTE9	VARCHAR2
GLOBAL_ATTRIBUTE10	VARCHAR2
GLOBAL_ATTRIBUTE11	VARCHAR2
GLOBAL_ATTRIBUTE12	VARCHAR2
GLOBAL_ATTRIBUTE13	VARCHAR2
GLOBAL_ATTRIBUTE14	VARCHAR2
GLOBAL_ATTRIBUTE15	VARCHAR2
GLOBAL_ATTRIBUTE16	VARCHAR2
GLOBAL_ATTRIBUTE17	VARCHAR2
GLOBAL_ATTRIBUTE18	VARCHAR2

Navigate to *Order Management Suite (OM_PF) > Order Management (ONT) > Sales Order (ONT_SALES_ORDER) > Interface_Views* to select `OE_HEADER_ACKS_V`.

Click **OK**.

> **Note:** You can also search for a view by entering the name of the program in the **Object Name** field. Select the **Views** check box, and then click **Search**.

9. The Application Interface dialog appears with the selected interface view.


Adapter Configuration Wizard - Step 4 of 4: Application Interface

You are connected to a 11.5.10 instance. Add the database objects needed for this service. You may add one or more tables/views or you can add a single API/Concurrent Program/XML Gateway Map/EDI Concurrent Program/Business Event for a service. If no objects are selected, it is assumed that you want to use previously created descriptors.

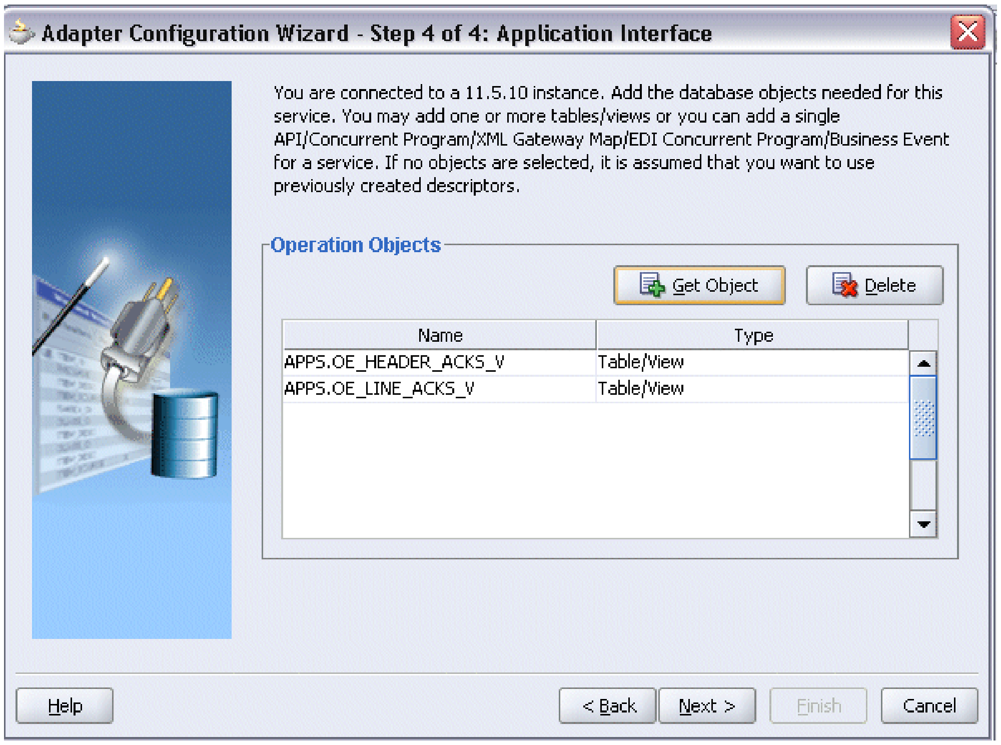
Operation Objects

Name	Type
APPS.OE_HEADER_ACKS_V	Table/View

Click **Get Object** to open the Oracle Applications Module Browser again to select another open interface table `OE_LINES_ACKS_V` using the same navigation path.

Click **OK**. The Application Interface dialog appears with the two selected views.

Application Interface Dialog



Adapter Configuration Wizard - Step 4 of 4: Application Interface

You are connected to a 11.5.10 instance. Add the database objects needed for this service. You may add one or more tables/views or you can add a single API/Concurrent Program/XML Gateway Map/EDI Concurrent Program/Business Event for a service. If no objects are selected, it is assumed that you want to use previously created descriptors.

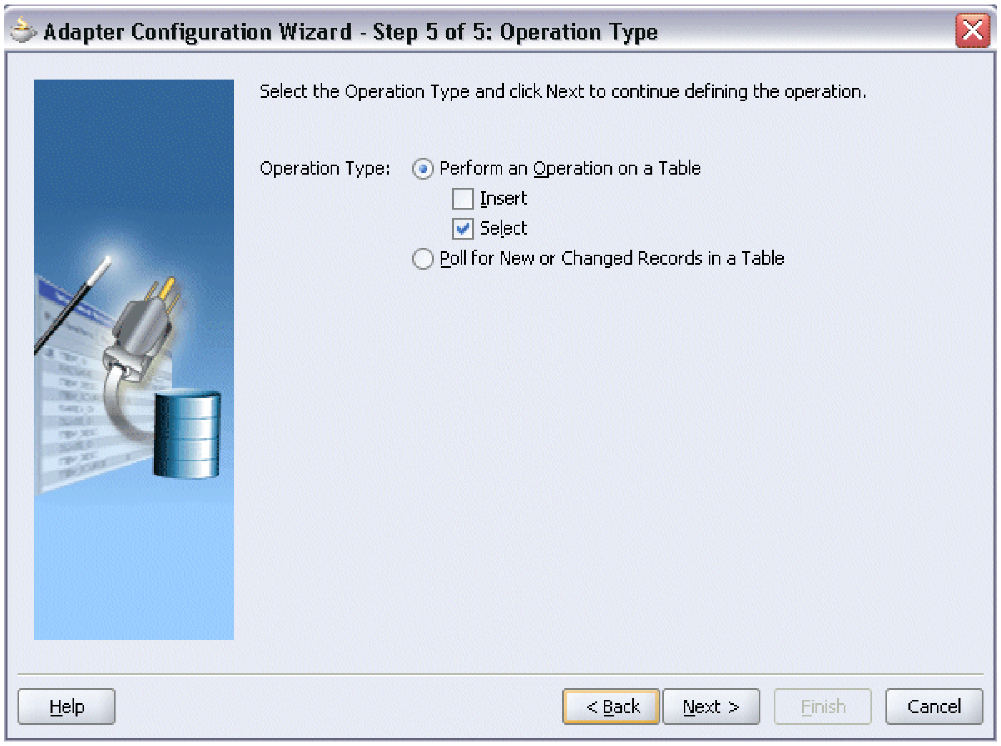
Operation Objects

Name	Type
APPS.OE_HEADER_ACKS_V	Table/View
APPS.OE_LINE_ACKS_V	Table/View

10. Click **Next** in the Operation dialog box. The Operation Type dialog box appears.

11. Select **Perform an Operation on a Table**, then select **Select**.

Selecting the Type of Operation


Adapter Configuration Wizard - Step 5 of 5: Operation Type

Select the Operation Type and click Next to continue defining the operation.

Operation Type: [● Perform an Operation on a Table]

- [ ] Insert
- [x] Select

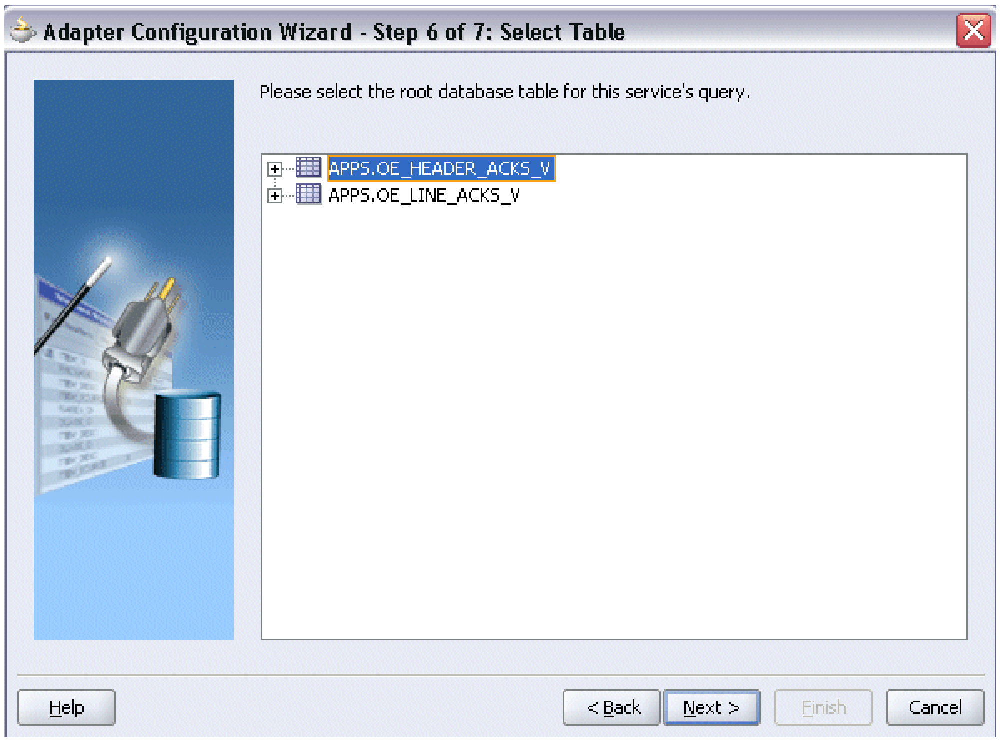
○ Poll for New or Changed Records in a Table

Note: You can perform only the *Select* operation on views.

12. Click **Next**. The Select Table dialog box appears.

13. Select APPS.OE_HEADERS_ACKS_V as the root database table in the Select Table dialog box, which displays the tables that have been previously imported in this JDeveloper project (including tables that were imported for other partner links). This enables you to reuse configured table definitions in multiple partner links.

Selecting a Root Table



Adapter Configuration Wizard - Step 6 of 7: Select Table

Please select the root database table for this service's query.

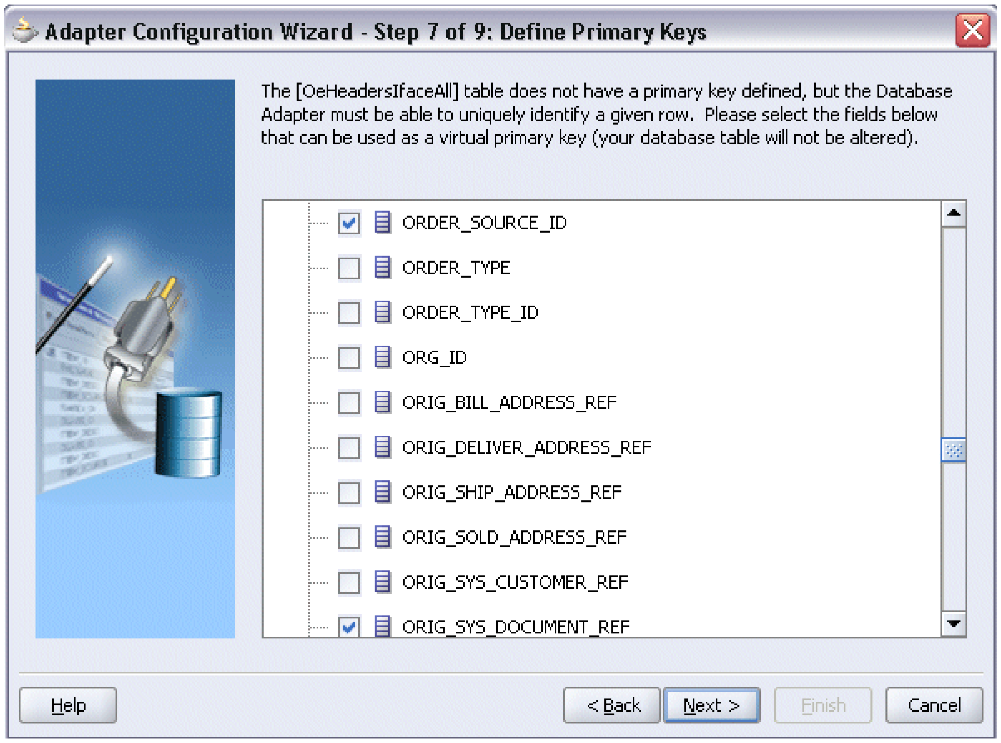
APPS.OE_HEADER_ACKS_V
APPS.OE_LINE_ACKS_V

Select the required database table, and then click **Next**. The Define Primary Keys dialog box appears.

14. Select the following primary keys for the APPS.OE_HEADERS_ACKS_V view:

- ORDER_SOURCE_ID
- ORIG_SYS_DOCUMENT_REF

Defining Primary Keys Page for APPS.OE_HEADERS_ACKS_V



Adapter Configuration Wizard - Step 7 of 9: Define Primary Keys

The [OeHeadersIfaceAll] table does not have a primary key defined, but the Database Adapter must be able to uniquely identify a given row. Please select the fields below that can be used as a virtual primary key (your database table will not be altered).

- [✓] ORDER_SOURCE_ID
- [ ] ORDER_TYPE
- [ ] ORDER_TYPE_ID
- [ ] ORG_ID
- [ ] ORIG_BILL_ADDRESS_REF
- [ ] ORIG_DELIVER_ADDRESS_REF
- [ ] ORIG_SHIP_ADDRESS_REF
- [ ] ORIG_SOLD_ADDRESS_REF
- [ ] ORIG_SYS_CUSTOMER_REF
- [✓] ORIG_SYS_DOCUMENT_REF

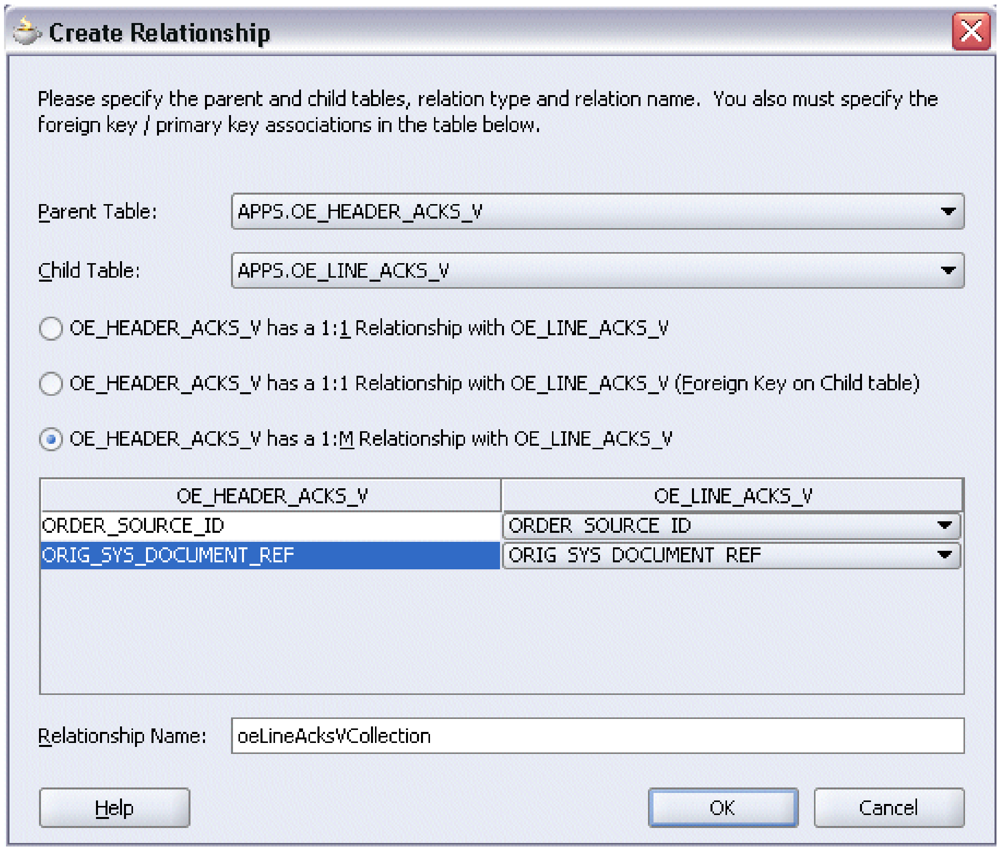
Click **Next**.

Select the same primary keys for the APPS.OE_LINES_ACKS_V view.

15. Click **Next**. The Relationships dialog box appears.

16. Click **Create** to define a new relationship. The Create Relationship dialog box appears.

Defining Relationships



Please specify the parent and child tables, relation type and relation name. You also must specify the foreign key / primary key associations in the table below.

Parent Table: APPS.OE_HEADER_ACKS_V

Child Table: APPS.OE_LINE_ACKS_V

( ) OE_HEADER_ACKS_V has a 1:1 Relationship with OE_LINE_ACKS_V

( ) OE_HEADER_ACKS_V has a 1:1 Relationship with OE_LINE_ACKS_V (Foreign Key on Child table)

(✓) OE_HEADER_ACKS_V has a 1:M Relationship with OE_LINE_ACKS_V

OE_HEADER_ACKS_V	OE_LINE_ACKS_V
ORDER_SOURCE_ID	ORDER_SOURCE_ID
ORIG_SYS_DOCUMENT_REF	ORIG_SYS_DOCUMENT_REF

Relationship Name: oeLineAcksVCollection

17. Enter the following information to define the relationship between the header and the detail table:

   - Select the APPS.OE_HEADERS_ACKS_V as the parent table and APPS.OE_LINES_ACKS_V as the child table.

   - Select the mapping type: APPS.OE_HEADERS_ACKS_V has a 1:M Relationship with APPS.OE_LINES_ACKS_V

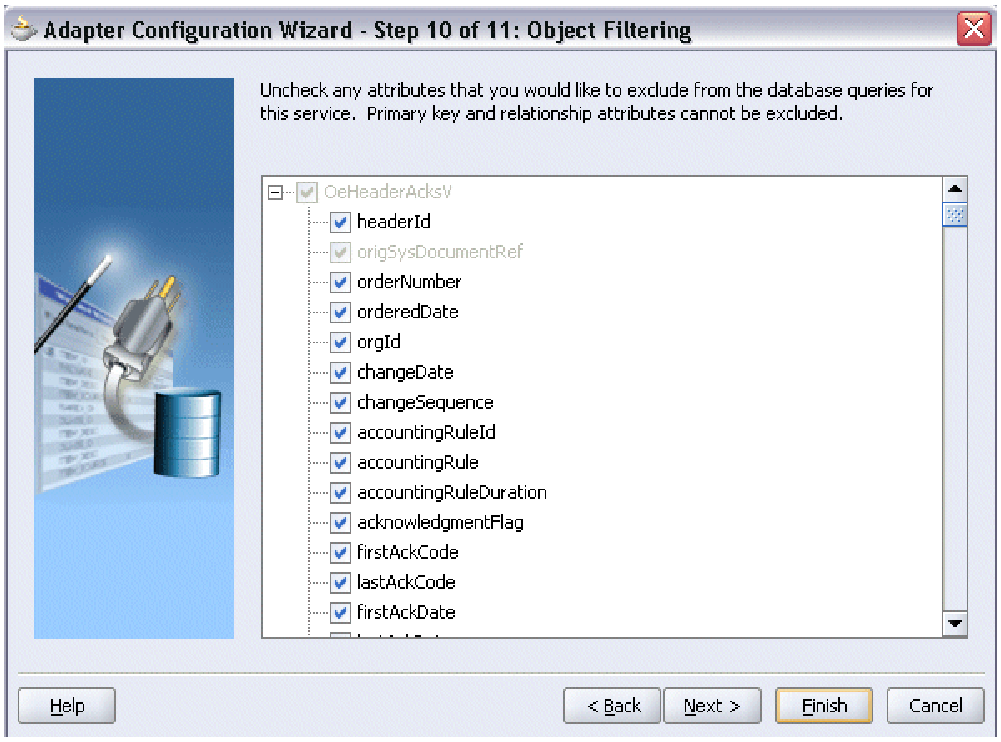
     Note: If foreign key constraints between tables already exist in the database, then two relationships are created automatically while importing tables. One of the relationships is 1:M relationship from the source table, which is the table containing the foreign key constraints, to the target table. The other relationship is a 1:1 back pointer from the target table to the source table.

   - Select the **Private Owned** check box.

- Associate the foreign key fields with the primary key fields:
  - OE_HEADERS_ACKS_V.ORDER_SOURCE_ID: OE_LINES_ACKS_V.ORDER_SOURCE_ID
  - OE_HEADERS_ACKS_V.ORIG_SYS_DOCUMENT_REF: OE_LINES_ACKS_V.ORIG_SYS_DOCUMENT_REF
- The Relationship Name field is populated automatically by default. You can optionally specify a new name for the relationship you are creating.

18. Click **Next**. The Object Filtering dialog box appears.

Object Filtering

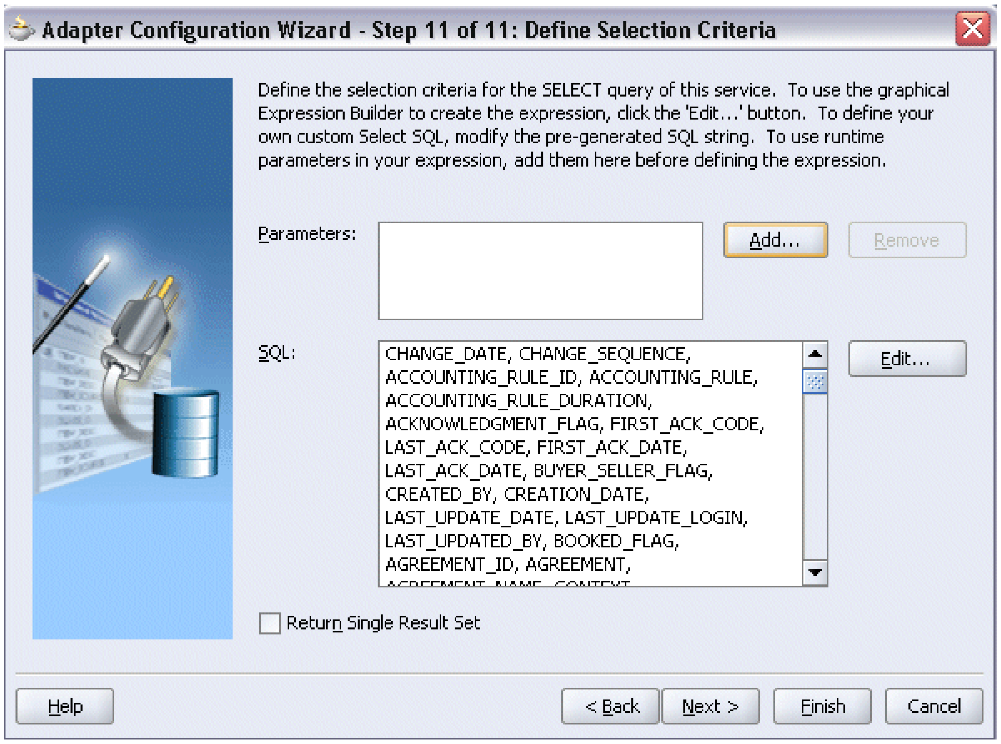

Adapter Configuration Wizard - Step 10 of 11: Object Filtering

Uncheck any attributes that you would like to exclude from the database queries for this service. Primary key and relationship attributes cannot be excluded.

- OeHeaderAcksV
  - ✓ headerId
  - ✓ origSysDocumentRef
  - ✓ orderNumber
  - ✓ orderedDate
  - ✓ orgId
  - ✓ changeDate
  - ✓ changeSequence
  - ✓ accountingRuleId
  - ✓ accountingRule
  - ✓ accountingRuleDuration
  - ✓ acknowledgmentFlag
  - ✓ firstAckCode
  - ✓ lastAckCode
  - ✓ firstAckDate

19. Click **Next** to open the Define Selection Criteria dialog box.

Define Selection Criteria Dialog


Adapter Configuration Wizard - Step 11 of 11: Define Selection Criteria

Define the selection criteria for the SELECT query of this service. To use the graphical Expression Builder to create the expression, click the 'Edit...' button. To define your own custom Select SQL, modify the pre-generated SQL string. To use runtime parameters in your expression, add them here before defining the expression.

```
CHANGE_DATE, CHANGE_SEQUENCE,
ACCOUNTING_RULE_ID, ACCOUNTING_RULE,
ACCOUNTING_RULE_DURATION,
ACKNOWLEDGMENT_FLAG, FIRST_ACK_CODE,
LAST_ACK_CODE, FIRST_ACK_DATE,
LAST_ACK_DATE, BUYER_SELLER_FLAG,
CREATED_BY, CREATION_DATE,
LAST_UPDATE_DATE, LAST_UPDATE_LOGIN,
LAST_UPDATED_BY, BOOKED_FLAG,
AGREEMENT_ID, AGREEMENT,
```

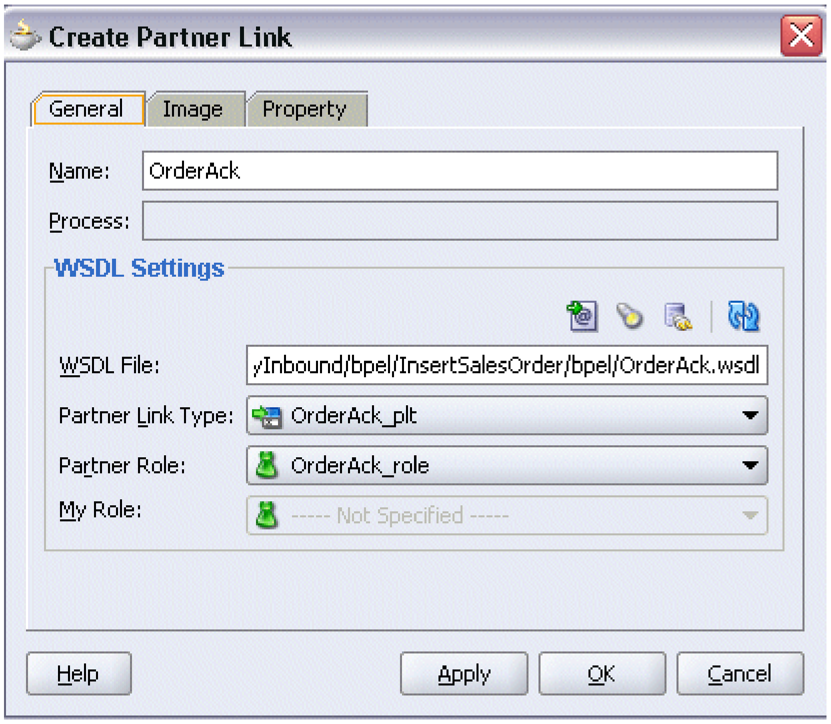
Return Single Result Set [ ]

The Define WHERE Clause dialog box appears. If your service contains a `SELECT` query, then you can customize the `WHERE` clause of the `SELECT` statement.

You can click **Add** to add a new parameter.

20. Click **Next**. The Finish dialog box appears.

21. Click **Finish**. The Create Partner Link dialog box appears.



Name: OrderAck

Process:

WSDL Settings

WSDL File: /Inbound/bpel/InsertSalesOrder/bpel/OrderAck.wsdl

Partner Link Type: OrderAck_plt

Partner Role: OrderAck_role

My Role: ----- Not Specified -----

22. Click **Apply** and **OK**. The partner link is created with the required WSDL settings.

Configuring the Invoke Activity

To configure the Invoke activity:

1. In JDeveloper BPEL Designer, drag and drop the **Invoke** activity from the Component Palette into the process diagram between the receiveInput and callbackClient activities.

2. Link the Invoke activity to the `OrderAck` service. The Edit Invoke dialog box appears.

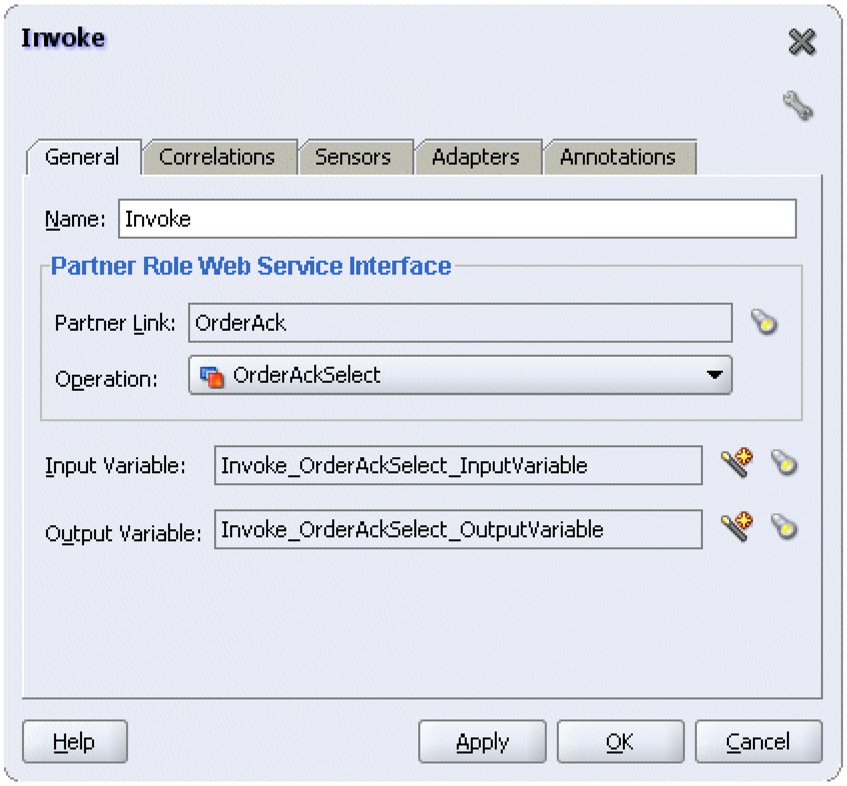
3. Click the **Create** icon next to the **Input Variable** field. The Create Variable dialog box appears.

   Select **Global Variable** and enter a name for the variable. You can also accept the default name. Click **OK**.

   Click the **Create** icon next to the **Output Variable** field. The Create Variable dialog box appears.

   Select **Global Variable** and enter a name for the variable. You can also accept the default name. Click **OK**.

Creating a Variable



Name: Invoke

Partner Link: OrderAck

Operation: OrderAckSelect

Input Variable: Invoke_OrderAckSelect_InputVariable

Output Variable: Invoke_OrderAckSelect_OutputVariable

4. In the Invoke dialog box, click **Apply** and then **OK**. The BPEL process is configured with the Invoke activity.

Configuring the Assign Activity

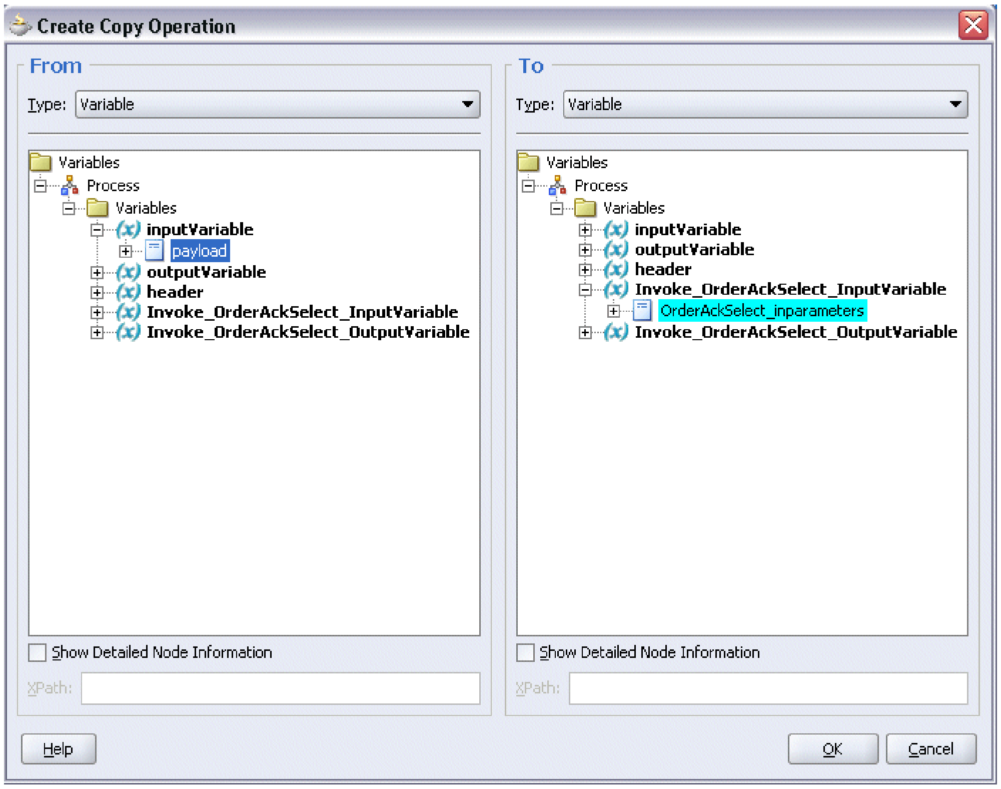
The next task is to add an Assign activity to the process map. This is used to provide values to the input variables.

To configure the Assign activity:

1. Drag and drop the Assign activity to the process map. The Assign activity must be added between the Receive and Invoke activities.
2. Double-click the **Assign** activity in the process map. The Assign dialog box appears.
3. Click **Create** and select **Copy Operation...** to open the Create Copy Operation dialog box.
4. In the From navigation tree, select type Variable. Navigate to **Variable > Process > Variables > inputVariable** and select **Payload**.

In the To navigation tree, select type Variable. Navigate to **Variable > Process > Variables > Invoke_OrderAckSelect_InputVariable** and select **OrderAckSelect_inputparameters**.

Assigning Values to the Input Variables


Create Copy Operation
From
Type: Variable
Variables
Process
Variables
inputVariable
payload
outputVariable
header
Invoke_OrderAckSelect_InputVariable
Invoke_OrderAckSelect_OutputVariable
Show Detailed Node Information
XPath:
To
Type: Variable
Variables
Process
Variables
inputVariable
outputVariable
header
Invoke_OrderAckSelect_InputVariable
OrderAckSelect_inparameters
Invoke_OrderAckSelect_OutputVariable
Show Detailed Node Information
XPath:
Help
OK
Cancel

Click **OK**.

5. After assigning values to the input variables, click **Apply** and **OK** to complete the Assign activity.

Run-Time Tasks for Views

After designing the BPEL process, the next steps are to deploy, run and monitor it.

1. Deploy the BPEL process, page 7-44

2. Test the BPEL process, page 7-45

Deploying the BPEL Process

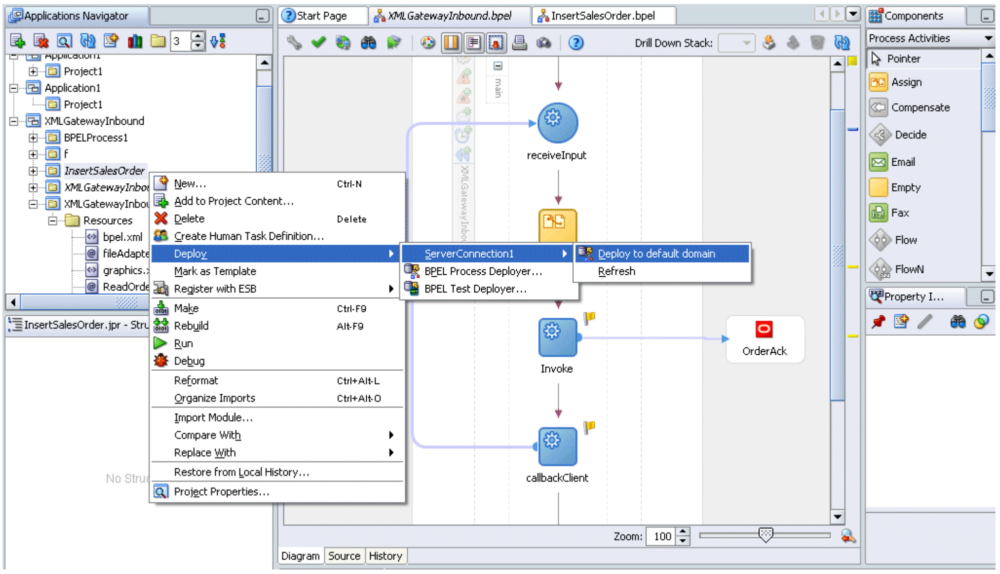
You need to deploy the BPEL process before you can run it. The BPEL process is first compiled, and then deployed to the BPEL server. The following steps discuss deploying the BPEL process to a BPEL server:

To deploy the BPEL process:

1. Select the BPEL project in the Applications window.

2. Right-click the project name. Select **Deploy > [Server Connection] > Deploy to Default Domain** from the menu that appears.

Deploying the BPEL Process



3. The BPEL process is compiled and deployed. You can check the progress of the compilation in the Messages window.

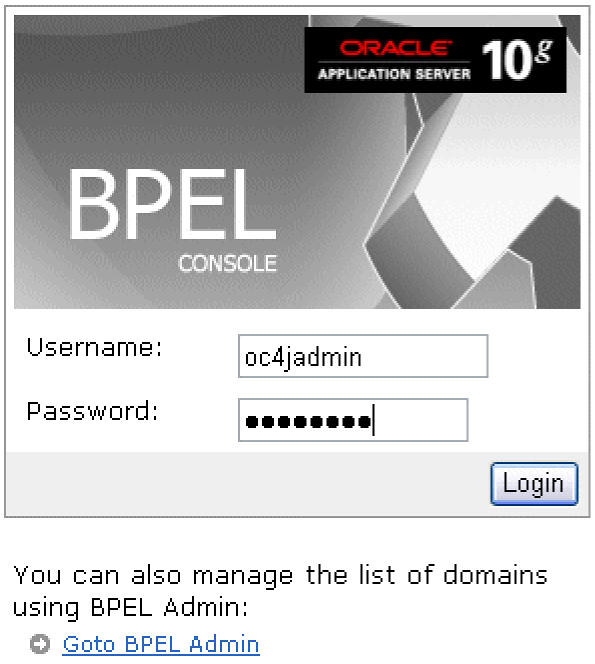
Testing the BPEL Process

Once the BPEL process is deployed, you can manage and monitor the process from the BPEL Console. You can also test the process and the integration interface by manually initiating the process. The following steps discuss manually initiating and monitoring the BPEL process:

To manually initiate and monitor the BPEL process:

1. Navigate to Oracle Application Server 10*g* BPEL Console (`http://<soaSuiteServerHostName>:<port>/BPELConsole`).

   The BPEL Console login page appears.



Username: oc4jadmin

Password: ●●●●●●●●●

You can also manage the list of domains using BPEL Admin:

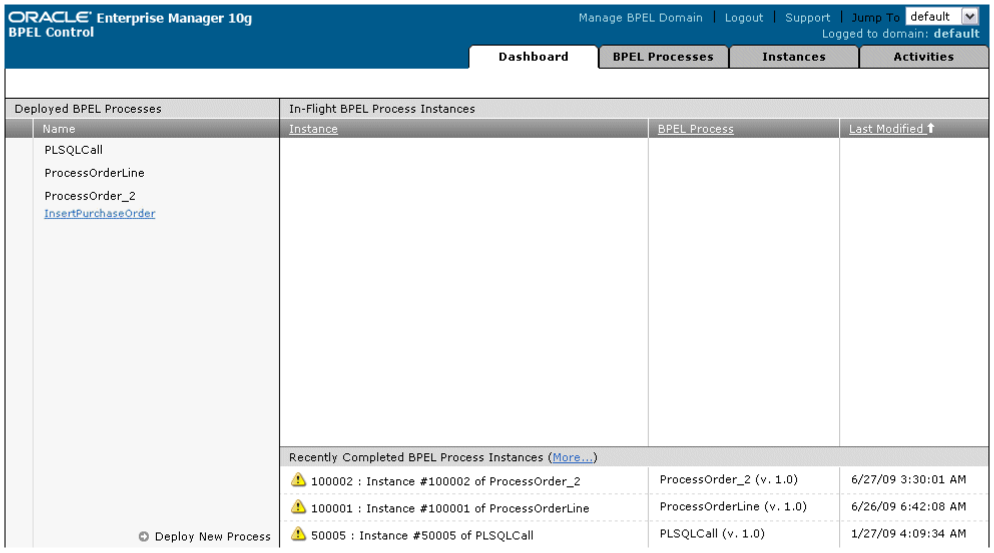
Goto BPEL Admin

2. Enter the username and password and click **Login**.

3. The Oracle Enterprise Manager 10*g* BPEL Control appears. The list of deployed processes is shown under Deployed BPEL Processes.

4.

Deployed BPEL Processes


ORACLE Enterprise Manager 10g BPEL Control

Manage BPEL Domain | Logout | Support | Jump To default

Logged to domain: default

Dashboard | BPEL Processes | Instances | Activities

Deployed BPEL Processes

Name
PLSQLCall
ProcessOrderLine
ProcessOrder_2
InsertPurchaseOrder

Deploy New Process

In-Flight BPEL Process Instances

Instance	BPEL Process	Last Modified

Recently Completed BPEL Process Instances (More...)

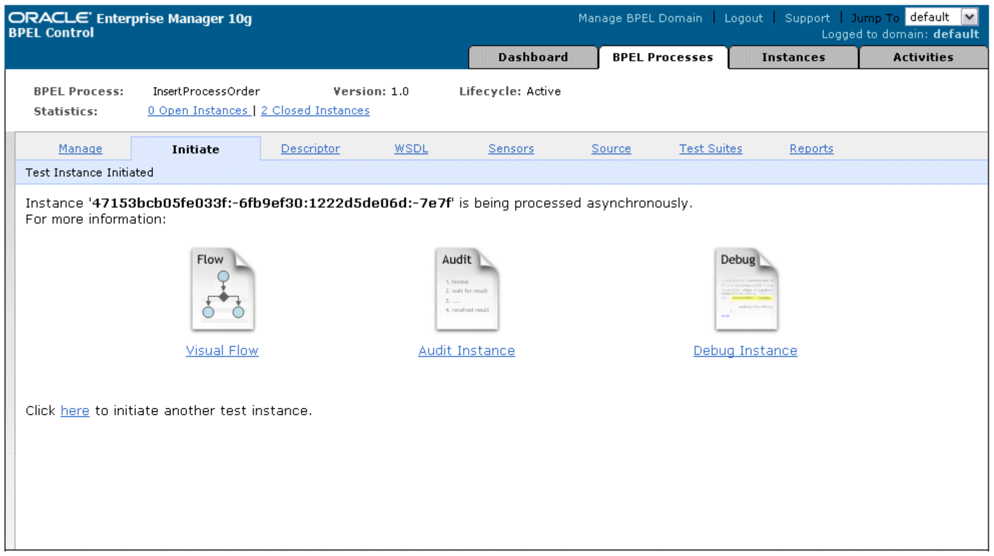
100002 : Instance #100002 of ProcessOrder_2	ProcessOrder_2 (v. 1.0)	6/27/09 3:30:01 AM
100001 : Instance #100001 of ProcessOrderLine	ProcessOrderLine (v. 1.0)	6/26/09 6:42:08 AM
50005 : Instance #50005 of PLSQLCall	PLSQLCall (v. 1.0)	1/27/09 4:09:34 AM

5. Click the BPEL process that you want to initiate. The Initiate page appears. Enter the input values required by the process.

6. Click **Post XML Message** to initiate the process.

7. The BPEL process is now initiated. You can check the process flow by clicking the **Visual Flow** icon on the BPEL Console Initiate page.

BPEL Console Initiate Page


ORACLE Enterprise Manager 10g
BPEL Control
Manage BPEL Domain | Logout | Support | Jump To default
Logged to domain: default
Dashboard
BPEL Processes
Instances
Activities

BPEL Process: InsertProcessOrder Version: 1.0 Lifecycle: Active

Statistics: 0 Open Instances | 2 Closed Instances

Test Instance Initiated

Instance '**47153bcb05fe033f:-6fb9ef30:1222d5de06d:-7e7f**' is being processed asynchronously.
For more information:

Click here to initiate another test instance.

8. The audit trail provides information about the steps that have been executed. You can check the audit trail by clicking the **Audit Instance** icon.

8

Using PL/SQL APIs for BPEL Process Integration

This chapter covers the following topics:

- Overview of PL/SQL APIs
- Design-Time Tasks for PL/SQL APIs
- Creating a New BPEL Project
- Adding a Partner Link
- Defining Wrapper APIs
- Declaring Parameters with a DEFAULT Clause
- Configuring the Invoke Activity
- Configuring the Assign Activity
- Run-Time Tasks for PL/SQL APIs
- Deploying the BPEL Process
- Testing the BPEL Process
- Troubleshooting and Debugging

Overview of PL/SQL APIs

OracleAS Adapter for Oracle Applications uses PL/SQL application programming interfaces (APIs) to insert and update data in Oracle Applications. APIs are stored procedures that enable you to insert and update data in Oracle Applications. Additionally, you can use PL/SQL APIs to retrieve data. For example, by using PL/SQL APIs, you can insert a customer record in Oracle Applications.

Note: For more information about PL/SQL procedure limitations, refer to *Oracle Application Server Adapters for Files, FTP, Databases, and*

Enterprise Messaging User's Guide.

Design-Time Tasks for PL/SQL APIs

This section describes how to configure the OracleAS Adapter for Oracle Applications to use PL/SQL APIs. It describes the tasks required to configure OracleAS Adapter for Oracle Applications using the Adapter Configuration Wizard in Oracle JDeveloper.

Prerequisites to Configure PL/SQL APIs

OracleAS Adapter for Oracle Applications is deployed using the BPEL Process Manager (PM) in Oracle JDeveloper. The BPEL PM creates the WSDL interfaces for the API.

Populating Applications Context Header Variables

You need to populate certain variables in the SOA Suite to provide context information required in an API transaction in order for an Oracle Applications user that has sufficient privileges to run the program.

The context is set taking into account the values passed for the header parameters including *Username, Responsibility, Responsibility Application, Security Group*, and *NLS Language*. If the values for the new header parameters *Responsibility Application, Security Group*, and *NLS Language* are not passed, context information will be determined based on *Username* and *Responsibility*.

The default value for the *Username* is `SYSADMIN`, the default value for *Responsibility* is `SYSTEM ADMINISTRATOR`, the default *Security Group Key* is `STANDARD`, and the default *NLS Language* is `US`.

You can change the default values specified in the generated WSDL. This is a static way of changing the context information. These values would apply to all invocations of the deployed business process. However, if you need to provide different context information for different invocations of the business process, then you can dynamically populate the header values. The context information can be specified by configuring an Assign activity before the Invoke activity in the BPEL PM.

Detailed instructions on how to assign header variables, see Design-Time Tasks for Assigning Header Variables, page 3-9.

Populating Default Values for Record Types

Certain PL/SQL APIs exposed from Oracle E-Business Suite take record types as input. Such APIs expect default values to be populated for parameters within these record types for successful execution.

The default values are `FND_API.G_MISS_CHAR` for characters, `FND_API.G_MISS_DATE` for dates, and `FND_API.G_MISS_NUM` for numbers. OracleAS Adapter for Oracle Applications can default these values when the parameters within the record type are passed as nil values, for example, as shown below:

```
<PRICE_LIST_REC>
<ATTRIBUTE1 xsi:nil="true"/>
<ATTRIBUTE2 xsi:nil="true"/>
<ATTRIBUTE3 xsi:nil="true"/>
...
</PRICE_LIST_REC>
```

This can be achieved with the help of a function in a Transform activity, or by directly passing the XML input with nil values and then assigning them to the record types within an Assign activity.

Following is a list of the procedures required to accomplish the design-time tasks.

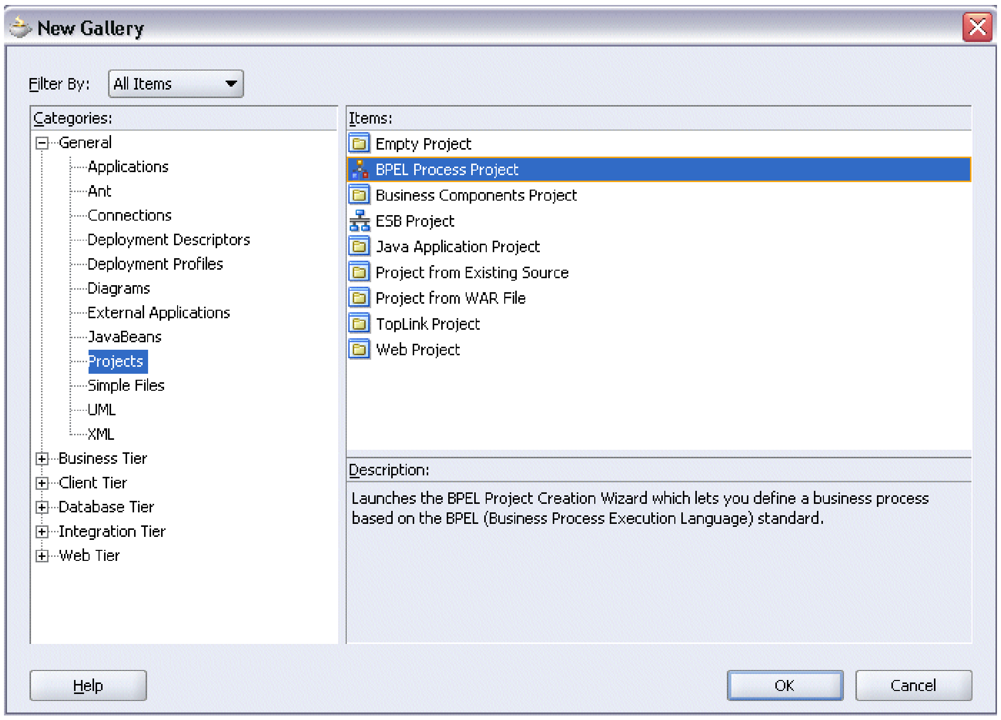
1. Create a new BPEL project, page 8-3
2. Add a partner link, page 8-6
3. Define wrapper APIs, page 8-13
4. Declare parameters with a DEFAULT clause, page 8-16
5. Configure the Invoke Activity, page 8-19
6. Configure the Assign Activity, page 8-26

Creating a New BPEL Project

To create a new BPEL project:

1. Open BPEL Designer.
2. From the **File** menu, select **New**. The New Gallery dialog box appears.
3. Select **All Items** from the **Filter By** list. This displays a list of available categories.
4. Expand the **General** node, and then select **Projects**.
5. Select **BPEL Process Project** from the **Items** list.

Creating a New BPEL Process Project



Launches the BPEL Project Creation Wizard which lets you define a business process based on the BPEL (Business Process Execution Language) standard.

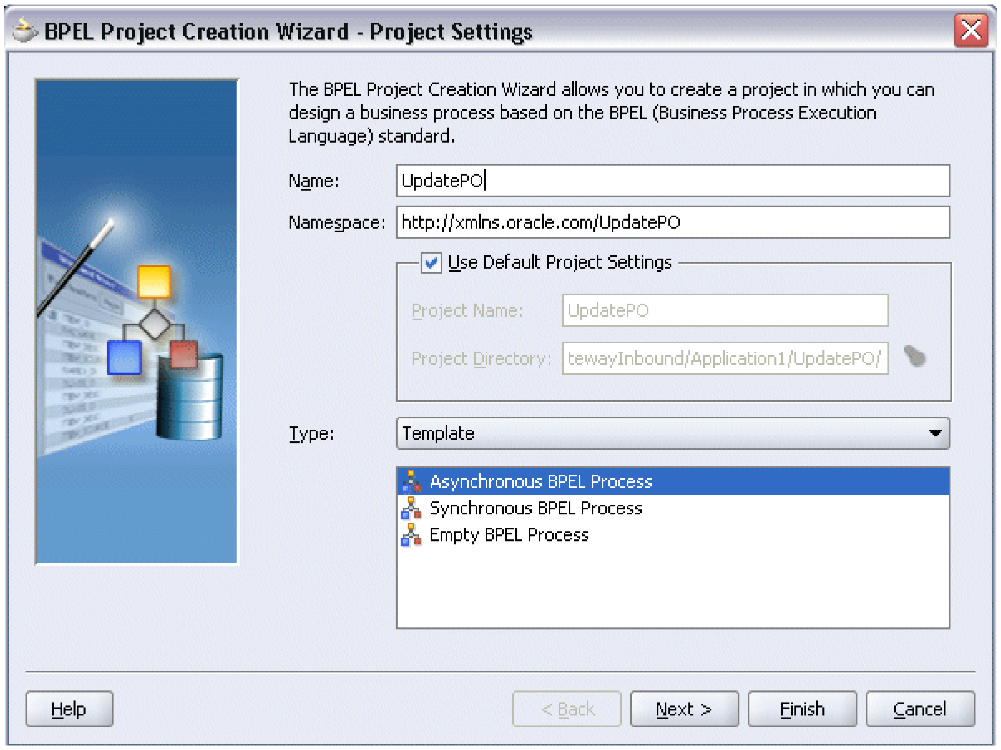
6. Click **OK**. The BPEL Project Creation Wizard - Project Settings dialog box appears.

7. In the **Name** field, enter a descriptive name. For example, enter `UpdatePO`.

   Keep the default selection **Use Default Project Settings** unchanged.

8. Keep the default selection **Template** as the Type field. Select **Asynchronous BPEL Process** as the BPEL process type.

Specifying a Name for the New BPEL Process Project


BPEL Project Creation Wizard - Project Settings

The BPEL Project Creation Wizard allows you to create a project in which you can design a business process based on the BPEL (Business Process Execution Language) standard.

Name: UpdatePO

Namespace: http://xmlns.oracle.com/UpdatePO

☑ Use Default Project Settings

Project Name: UpdatePO

Project Directory: tewayInbound/Application1/UpdatePO/

Type: Template

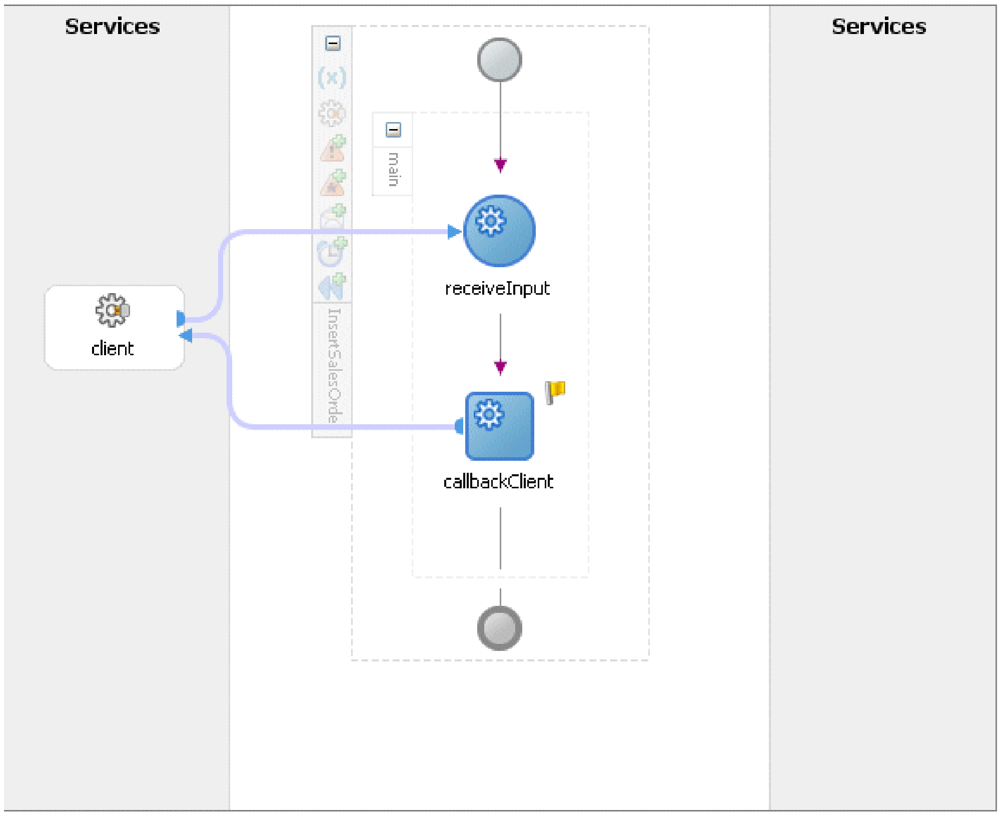
Asynchronous BPEL Process
Synchronous BPEL Process
Empty BPEL Process

9. Click **OK**.

   A new asynchronous BPEL process is created with the receiveInput and callbackClient activities.

   The required source files including `bpel.xml`, `UpdatePO.bpel`, and `UpdatePO.wsdl` are also generated.

New BPEL Process



Adding a Partner Link

This section describes how to add a partner link to your BPEL process. A BPEL partner link defines the link name, type, and the role of the BPEL process that interacts with the partner service.

To add a partner link:

1. In JDeveloper BPEL Designer, click **Services** in the Component palette.

   Drag and drop **Oracle Applications** icon into the border area of the process diagram.

   The Adapter Configuration Wizard appears.

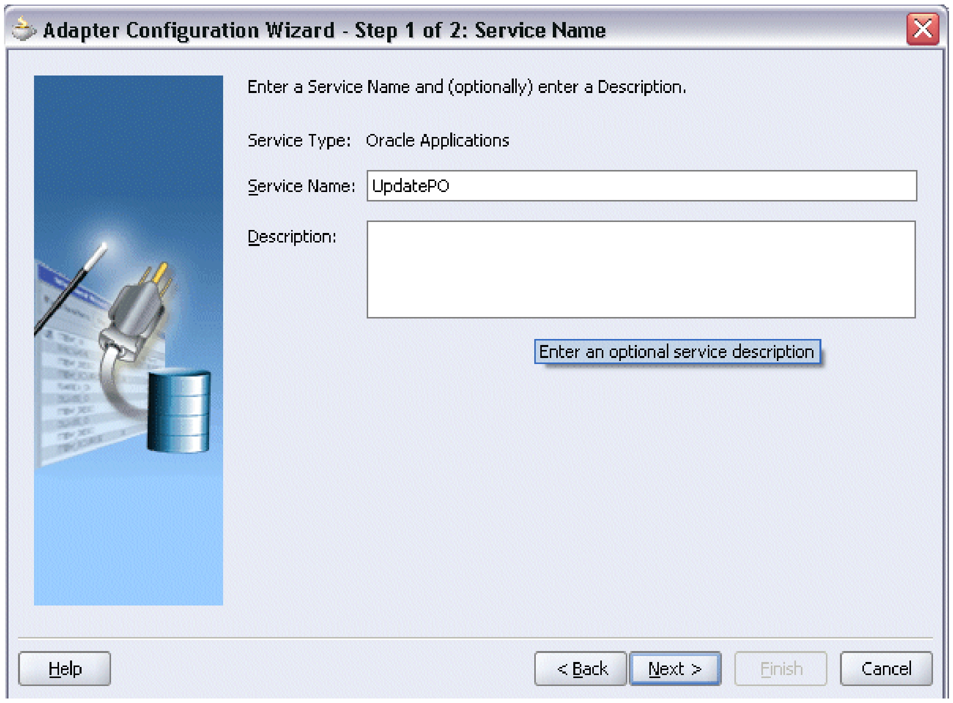
2. Click **Next**. The Service Name dialog box appears.

3. Enter the following information:

   1. In the **Service Name** field, enter a service name, such as `UpdatePO`.

2. In the **Description** field, enter a description for the service. This is an optional field. The Service Name dialog box appears.

Specifying the Service Name and Description


Adapter Configuration Wizard - Step 1 of 2: Service Name

Enter a Service Name and (optionally) enter a Description.

Service Type: Oracle Applications

Service Name: UpdatePO

Description:

4. Click **Next**. The Service Connection dialog box appears.

5. You can perform either one of the following options for your database connection:

   Note: You need to connect to the database where Oracle E-Business Suite is running.

   - You can create a new database connection by clicking the **New** icon.

     Detailed instructions on how to create a new database connection, see Creating a New Database Connection, page 4-12.

   - You can select an existing database connection that you have configured earlier from the **Connection** drop-down list.

     The selected database connection information appears in the Service Connection dialog box. The JNDI (Java Naming and Directory Interface) name corresponding to the selected database connection also appears automatically in the **Database Server JNDI Name** field. Alternatively, you can specify a JNDI name.

Note: When you specify a JNDI name, the deployment descriptor of the Adapter for Oracle Applications must associate this JNDI name with configuration properties required by the adapter to access the database.

The JNDI name acts as a placeholder for the connection used when your service is deployed to the BPEL server. This enables you to use different databases for development and later for production.

Note: For more information about JNDI concepts, refer to *Oracle Application Server Adapter Concepts.*

6. Once you have created a new connection or selected an existing connection, you can add a PL/SQL API by browsing through the list of APIs available in Oracle E-Business Suite.

7. Click **Next**.

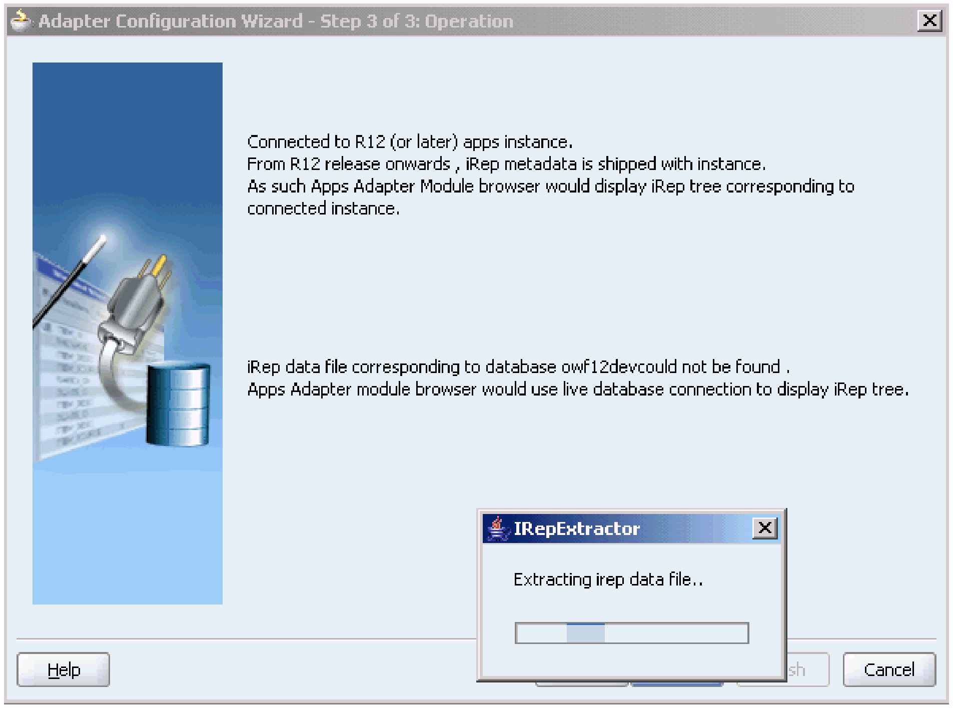
   For Oracle E-Business Suite Release 12:

   If you are connecting to Oracle E-Business Suite Release 12, then the **IREP File not present** dialog box appears indicating that Adapter for Oracle Applications could not find the Oracle Integration Repository data file corresponding to the database you are connecting in your workspace. Absence of the data file would make browsing or searching of Integration Repository tree considerably slow. You can choose to extract the data file and create a local copy of the Integration Repository data file. Once it is created successfully, Adapter for Oracle Applications will pick it up automatically next time and retrieve data from your local Integration Repository.

   You can select one of the following options:

   - Click **Yes** to extract the Integration Repository data file.

Extracting Integration Repository Data File



Adapter Configuration Wizard - Step 3 of 3: Operation

Connected to R12 (or later) apps instance.
From R12 release onwards , iRep metadata is shipped with instance.
As such Apps Adapter Module browser would display iRep tree corresponding to connected instance.

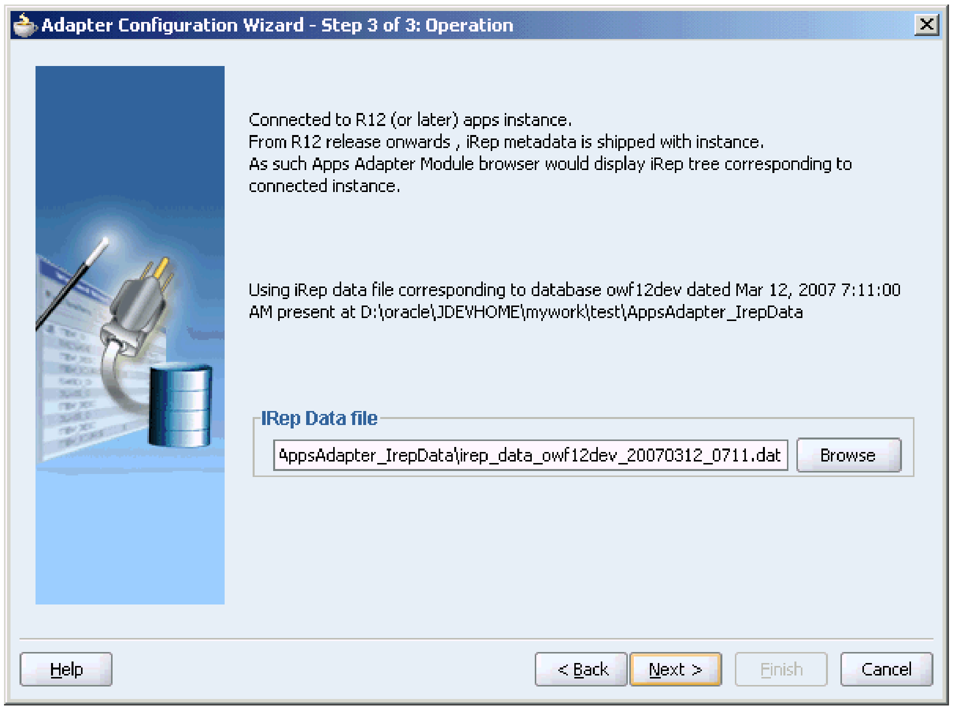
iRep data file corresponding to database owf12devcould not be found .
Apps Adapter module browser would use live database connection to display iRep tree.

IRepExtractor

Extracting irep data file..

After the system successfully creates a local copy of the Integration Repository data file, next time when you connect to the database, you will find the **IRep Data File** field appears in the Operation dialog box indicating where your local copy exists with the creation date and time as part of the file name.

Using the Local Integration Repository Data File


Adapter Configuration Wizard - Step 3 of 3: Operation

Connected to R12 (or later) apps instance.
From R12 release onwards , iRep metadata is shipped with instance.
As such Apps Adapter Module browser would display iRep tree corresponding to connected instance.

Using iRep data file corresponding to database owf12dev dated Mar 12, 2007 7:11:00 AM present at D:\oracle\JDEVHOME\mywork\test\AppsAdapter_IrepData

IRep Data file: AppsAdapter_IrepData\irep_data_owf12dev_20070312_0711.dat

- Click **No** to query the Integration Repository data file from the live database you are connecting to display the Integration Repository tree.

  Note: It is highly recommended that you create a local copy of the Integration Repository data file so that Adapter for Oracle Applications will query the data next time from the local copy in your workspace to enhance the performance.

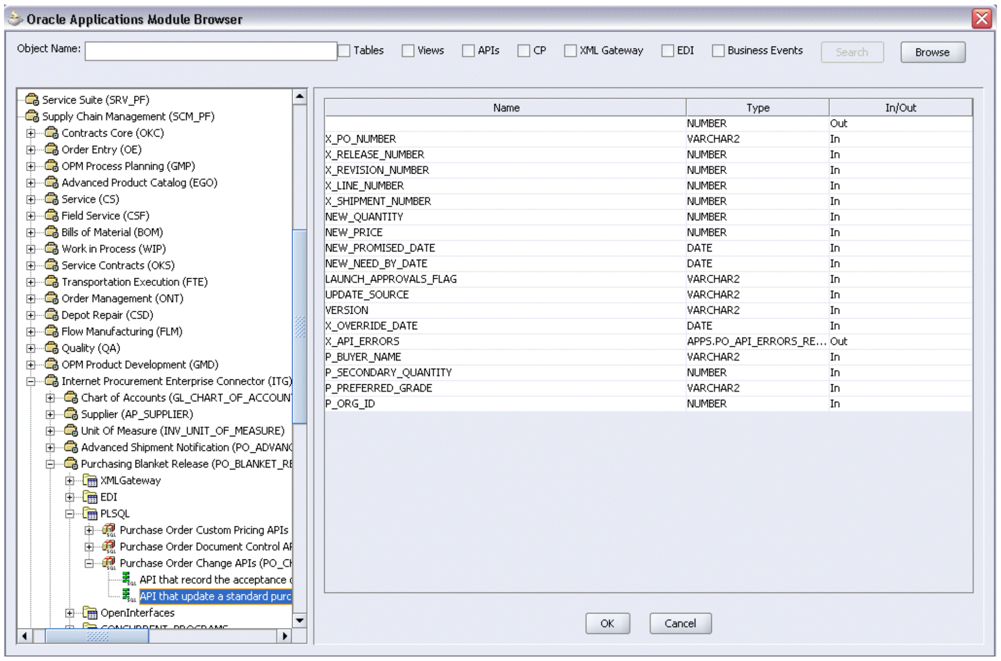
For Oracle E-Business Suite pre-Release 11.5.10:

If you are connecting to a pre-11.5.10 Oracle E-Business Suite instance, you must select the interface type in the Adapter Configuration Wizard. Select **Tables/Views/APIs/Concurrent Programs** to proceed.

Click **Add** in the Application Interface dialog box to open the Oracle Applications Module Browser.

8. The Oracle Applications Module Browser combines interface data from Oracle Integration Repository with information about the additional interfaces supported by Adapter for Oracle Applications, organized in a tree hierarchy.

Specifying the PL/SQL API



```
Service Suite (SRV_PF)
Supply Chain Management (SCM_PF)
  Contracts Core (OKC)
  Order Entry (OE)
  OPM Process Planning (GMP)
  Advanced Product Catalog (EGO)
  Service (CS)
  Field Service (CSF)
  Bills of Material (BOM)
  Work in Process (WIP)
  Service Contracts (OKS)
  Transportation Execution (FTE)
  Order Management (ONT)
  Depot Repair (CSD)
  Flow Manufacturing (FLM)
  Quality (QA)
  OPM Product Development (GMD)
  Internet Procurement Enterprise Connector (ITG)
    Chart of Accounts (GL_CHART_OF_ACCOUN
    Supplier (AP_SUPPLIER)
    Unit Of Measure (INV_UNIT_OF_MEASURE)
    Advanced Shipment Notification (PO_ADVANC
    Purchasing Blanket Release (PO_BLANKET_RE
      XMLGateway
      EDI
      PLSQL
        Purchase Order Custom Pricing APIs
        Purchase Order Document Control AF
        Purchase Order Change APIs (PO_CH
          API that record the acceptance c
          API that update a standard purc
      OpenInterfaces
```

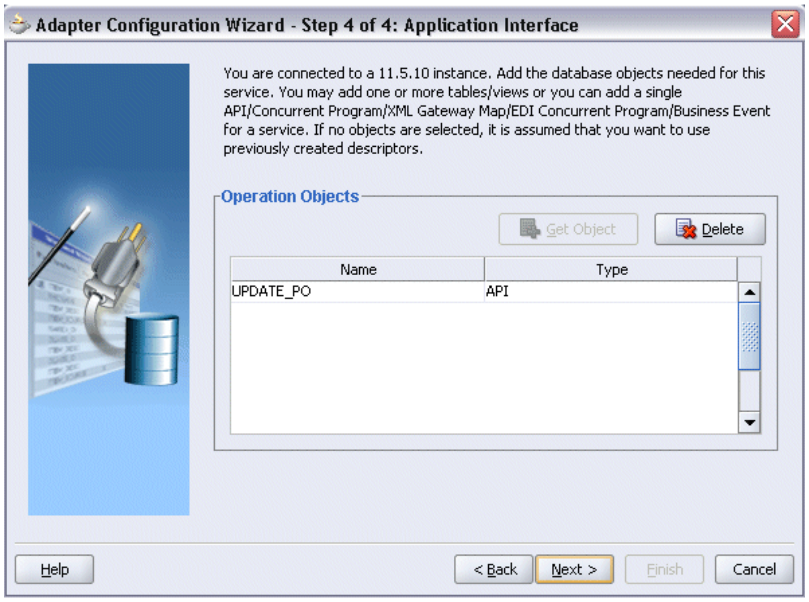
Name	Type	In/Out
	NUMBER	Out
X_PO_NUMBER	VARCHAR2	In
X_RELEASE_NUMBER	NUMBER	In
X_REVISION_NUMBER	NUMBER	In
X_LINE_NUMBER	NUMBER	In
X_SHIPMENT_NUMBER	NUMBER	In
NEW_QUANTITY	NUMBER	In
NEW_PRICE	NUMBER	In
NEW_PROMISED_DATE	DATE	In
NEW_NEED_BY_DATE	DATE	In
LAUNCH_APPROVALS_FLAG	VARCHAR2	In
UPDATE_SOURCE	VARCHAR2	In
VERSION	VARCHAR2	In
X_OVERRIDE_DATE	DATE	In
X_API_ERRORS	APPS.PO_API_ERRORS_RE...	Out
P_BUYER_NAME	VARCHAR2	In
P_SECONDARY_QUANTITY	NUMBER	In
P_PREFERRED_GRADE	VARCHAR2	In
P_ORG_ID	NUMBER	In

> **Note:** The Oracle Applications Module Browser includes the various product families that are available in Oracle Applications. Each product family contains the individual products. Each product contains the business entities associated with the product. Business entities contain the various application modules that are exposed for integration. These modules are grouped according to the interface they provide PL/SQL APIs can be found under the PL/SQL category.

Navigate to *Supplier Chain Management (SCM_PF)* > *Internet Procurement Enterprise Connector (ITG)* >*Purchase Blanket Release* > *PLSQL* > *Purchase Order Change APIs* > *UPDATE_PO*. The signature of that given API appears.

> **Note:** You can use the Search option to quickly find the required objects. Enter the required database object name in the **Object Name** field and select **APIs**. Click **Search** to retrieve the required database objects. When searching for a PL/SQL API, enter the package name, and *not* the procedure name. In contrast, when using the DB Adapter Wizard, the user *must* enter the name of the PL/SQL API while searching.

Adding the PL/SQL API


Adapter Configuration Wizard - Step 4 of 4: Application Interface

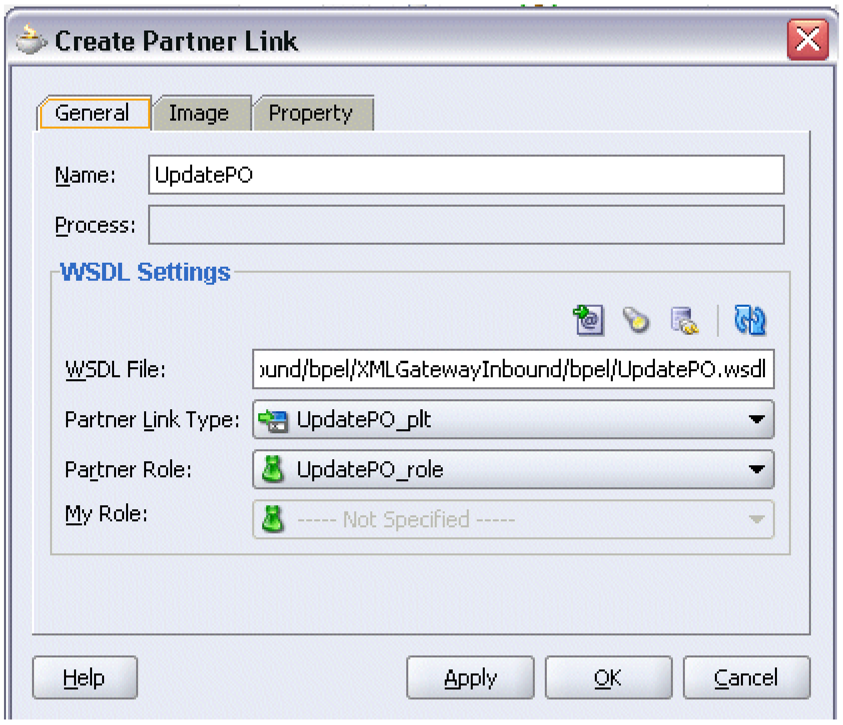
You are connected to a 11.5.10 instance. Add the database objects needed for this service. You may add one or more tables/views or you can add a single API/Concurrent Program/XML Gateway Map/EDI Concurrent Program/Business Event for a service. If no objects are selected, it is assumed that you want to use previously created descriptors.

Operation Objects

Name	Type
UPDATE_PO	API

9. Click **Next** and then click **Finish** to complete the process of configuring Adapter for Oracle Applications. The wizard generates the WSDL file corresponding to the XML schema. This WSDL file is now available for the partner link.

Create Partner Link



Name: UpdatePO

Process:

WSDL File: ›und/bpel/XMLGatewayInbound/bpel/UpdatePO.wsdl

Partner Link Type: UpdatePO_plt

Partner Role: UpdatePO_role

My Role: ----- Not Specified -----

10. Click **Apply** and **OK**. The partner link is created with the required WSDL settings.

Defining Wrapper APIs

The Adapter Configuration Wizard generates a wrapper API when a PL/SQL API has arguments of data types, such as PL/SQL Boolean, PL/SQL Table, or PL/SQL Record. For generating the wrapper API, Oracle JPublisher is automatically invoked in the background. When a wrapper API is created, besides the WSDL and XSD files, two SQL files are created: one for creating the wrapper API and necessary datatypes, and another for deleting it. The two SQL files are saved in the same directory where the WSDL and XSD files are stored, and are available in the Project view.

Following is a sample code of a PL/SQL API that would require a wrapper to be generated:

```
package pkg is
    type rec is record (...);
    type tbl is table of .. index by ..;
    procedure proc(r rec, t tbl, b boolean);
end;
```

If the preceding PL/SQL API is selected in the wizard, then a wrapper API will be created and loaded into the database. This wrapper API will be used instead of the originally selected PL/SQL API. For this reason, the content of the WSDL and XSD files represent the wrapper procedure, not the procedure originally selected.

The following are the types that will be created for the wrapper API:

- Object type for PL/SQL RECORD
- Nested table of the given type for PL/SQL TABLE, for example, the nested table of NUMBER
- INTEGER substituted for PL/SQL BOOLEAN

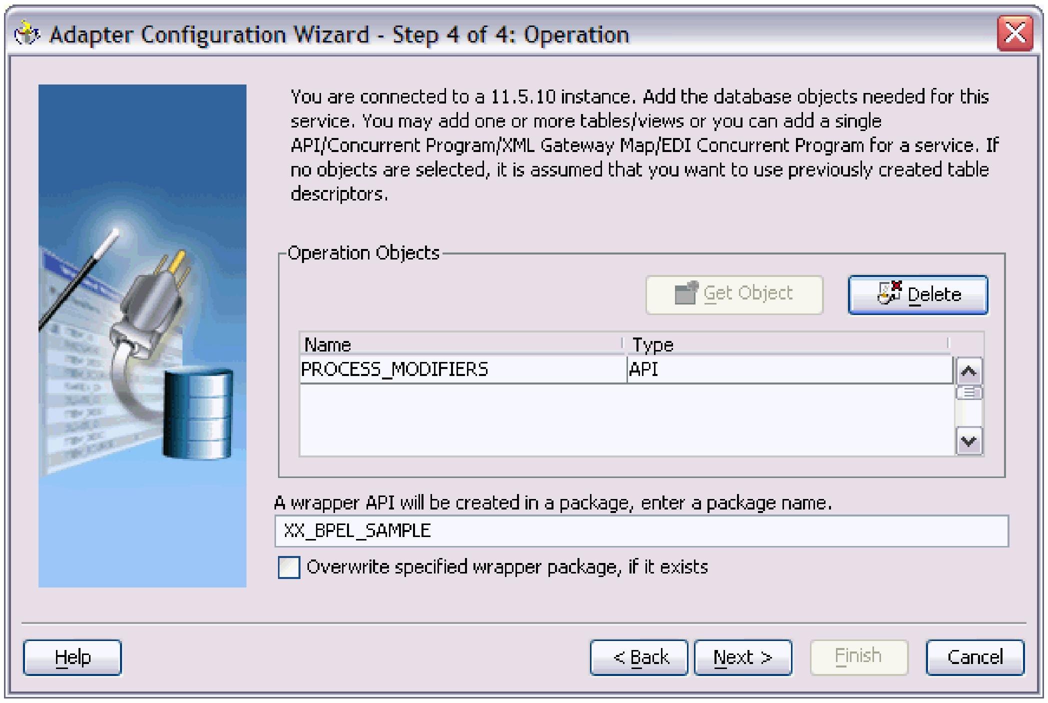
The generated SQL file that creates the wrapper API also creates the required schema objects. The types of the wrapper API's parameters will be those of the new schema object types. The wrapper package will contain conversion APIs to convert between the base PL/SQL type and the new schema object types.

> **Note:** The `SQLJUTL` package contains the BOOL2INT and INT2BOOL conversion functions used for PL/SQL BOOLEAN arguments whose data types have been changed to INTEGER.

The wrapper API is created in a package. This package is named XX_BPEL_*servicename. servicename* is the name of the service that you entered in step 6 of Adding a Partner Link, page 8-6. If this package already exists, then the wizard prompts for a different package name, or to select a checkbox, to overwrite the existing package. Overwriting an existing package causes all PL/SQL APIs in the specified package to be lost. When the wizard creates a package for the wrapper, only one API, that is, the wrapper API, is contained in it.

> **Note:** Despite specifying to overwrite an existing package, if the wrapper API already exists in the specified package, the wizard will not re-create the wrapper API, as it would take some time. This means no SQL files will be created, Oracle JPublisher will not be run, and the WSDL and XSD files will be for the existing wrapper API. The Finish page of the wizard will indicate that these actions will take place, but it is possible that they will not, depending on whether the wrapper already exists.

Entering a Package Name for the Wrapper API



Adapter Configuration Wizard - Step 4 of 4: Operation

You are connected to a 11.5.10 instance. Add the database objects needed for this service. You may add one or more tables/views or you can add a single API/Concurrent Program/XML Gateway Map/EDI Concurrent Program for a service. If no objects are selected, it is assumed that you want to use previously created table descriptors.

Operation Objects

Name	Type
PROCESS_MODIFIERS	API

A wrapper API will be created in a package, enter a package name.

XX_BPEL_SAMPLE

Overwrite specified wrapper package, if it exists

Note: The package name for the wrapper has a limit of 30 characters, and the wrapper API name has a limit of 29 characters. Thus, if the package name or the wrapper API name is longer than the maximum limit, it will be truncated accordingly.

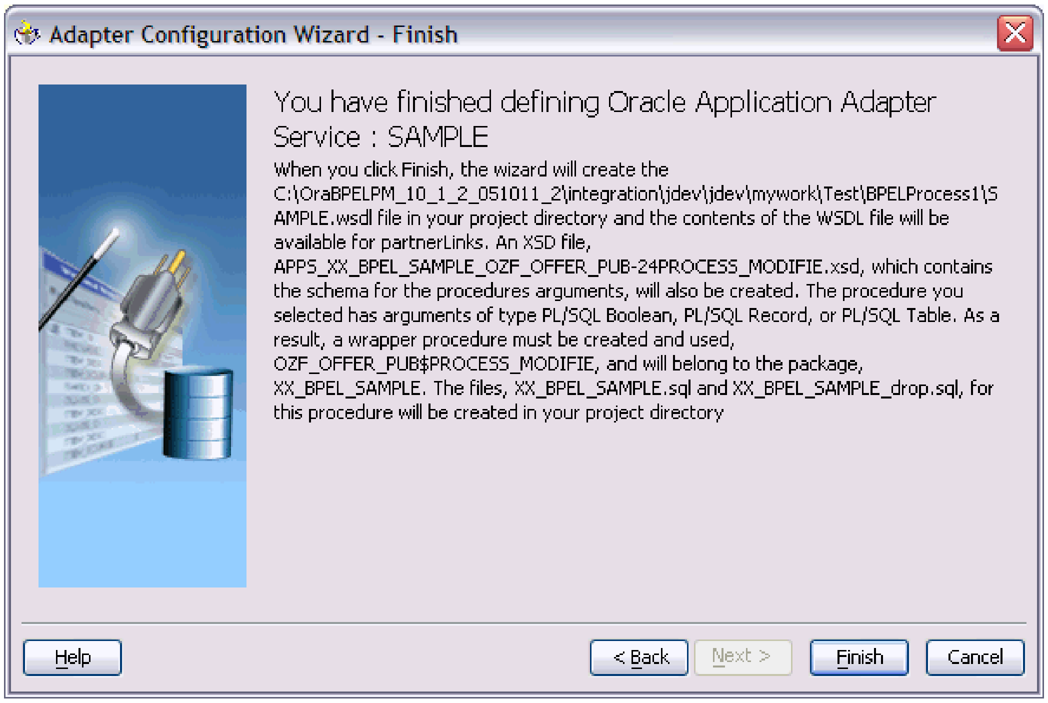
The name of the wrapper API depends on whether the PL/SQL API that was originally selected is in a package or not. If the original PL/SQL API is a root-level API, that is, it does not belong in a package, then the name of the wrapper API will be, TOPLEVEL$ *original_api_name.* If the originally selected PL/SQL API is in a package, then the name of the wrapper API will be *original_package_name*$*original_api_name.*

Wrapper APIs follow the naming convention of Oracle JPublisher. For example, if the original PL/SQL API was called `CREATE_EMPLOYEE` and was a root-level API, then the wrapper API would be named `TOPLEVEL$CREATE_EMPLOYEE`. If the original PL/SQL API is in a package called `EMPLOYEE`, then the wrapper API would be named `EMPLOYEE$CREATE_EMPLOYEE`.

The Finish page of the wizard is different when a wrapper API needs to be created. The Finish page informs you that a wrapper API is needed, and in addition, lists the name of the wrapper package, wrapper API, and the SQL files that will be created.

Note: When a wrapper API needs to be created, it may take a while before the wizard completes. However, the processing time for subsequent PL/SQL APIs in the same package would be much shorter.

The Finish Page



You have finished defining Oracle Application Adapter Service : SAMPLE

When you click Finish, the wizard will create the C:\OraBPELPM_10_1_2_051011_2\integration\jdev\jdev\mywork\Test\BPELProcess1\SAMPLE.wsdl file in your project directory and the contents of the WSDL file will be available for partnerLinks. An XSD file, APPS_XX_BPEL_SAMPLE_OZF_OFFER_PUB-24PROCESS_MODIFIE.xsd, which contains the schema for the procedures arguments, will also be created. The procedure you selected has arguments of type PL/SQL Boolean, PL/SQL Record, or PL/SQL Table. As a result, a wrapper procedure must be created and used, OZF_OFFER_PUB$PROCESS_MODIFIE, and will belong to the package, XX_BPEL_SAMPLE. The files, XX_BPEL_SAMPLE.sql and XX_BPEL_SAMPLE_drop.sql, for this procedure will be created in your project directory

Note: The `REF CURSOR` type is not supported out of the box. However, for detailed steps to generate an adapter service for a PL/SQL API which takes `REF CURSOR` type, see the section "Support for REF CURSOR" in *Oracle BPEL Process Manager Developer's Guide* on OTN.

Overloaded APIs are not supported in release 11.5.10.

Declaring Parameters with a DEFAULT Clause

You can declare parameters of a stored procedure with a DEFAULT clause, so that when you invoke the procedure without that parameter, BPEL Process Manager will supply a default value for that parameter. For example:

```
PROCEDURE addEmployee (name VARCHAR2, country VARCHAR2 DEFAULT 'US')
```

This procedure can be invoked in the following two ways:

- `addEmployee ('John Smith') // country => 'US'`
- `addEmployee ('John Smith', 'France') // country => 'France'`

Omitting Parameters With a DEFAULT Clause

You can omit elements for parameters with default values in the instance XML. The procedure will be invoked without these parameters, allowing their default values to be used, as shown in the following example of input and runtime invocation.

Input

```
<db:InputParameters xmlns:db="…">
    <name>John Smith</name>
</db:InputParameters>
```

Runtime Invocation

```
BEGIN addEmployee (name=>?); END; // country => 'US'
```

If the input includes a value for the defaulted parameter, the value in the input will be used, rather than the default, as follows:

Input

```
<db:InputParameters xmlns:db="…">
    <name>John Smith</name>
    <country>France</country>
</db:InputParameters>
```

Runtime Invocation

```
BEGIN addEmployee (name=>?, country=>?); END; // country => 'France'
```

Omitting Parameters Without a DEFAULT Clause

The element in the XSD for parameters with a default clause is annotated with a special tag to indicate that the parameter has a default clause, as shown in the following example.

```
<element name="country" … db:default="true" …/>
```

This new functionality allows elements for parameters without a default clause also to be omitted in the instance XML. In these cases, the parameter is still included in the invocation of the stored procedure. A value of NULL is bound by default. Following is an example of a declaration where neither parameter has a DEFAULT clause:

```
PROCEDURE addEmployee (name VARCHAR2, country VARCHAR2)
```

In BPEL Process Manager release 10.1.2, elements for both parameters were required in the instance XML. If an element was omitted, it was presumed to have a DEFAULT clause, so the parameter was not included in the invocation of the procedure. In this case, the missing parameter resulted in a PL/SQL error stating that an incorrect number of arguments was passed to the procedure.

In the current BPEL Process Manager release, the missing parameter will be included in the invocation of the procedure. A `NULL` value will be bound, as shown in the following example:

Input

```
<db:InputParameters xmlns:db="…">
    <name>John Smith</name>
</db:InputParameters>
```

Runtime Invocation

```
BEGIN addEmployee (name =>?, country=>?); END; // country => NULL
```

Even though the element for *country* was not provided in the instance XML, it still appears in the call to the procedure. In this case, *country* will be `NULL`.

DEFAULT Clause Handling in Wrapper Procedures:

If a procedure contains a special type requiring a wrapper to be generated, the `default` clauses on any of the original parameters will *not* be carried over to the wrapper, as shown in the following example:

```
PROCEDURE needsWrapper(isTrue BOOLEAN, value NUMBER DEFAULT 0)
```

Assuming that the procedure in the preceding example was defined at the top level, outside of a package, the generated wrapper will appear, as shown in the following example:

```
TOPLEVEL$NEEDSWRAPPER (isTrue INTEGER, value NUMBER)
```

In the preceding example, the BOOLEAN type has been replaced by INTEGER. The default clause on the value parameter is missing. In the current release, parameters of generated wrapper procedures will never have a default clause, even if these parameters did in the original procedure. If the element is missing in the instance XML, instead of defaulting to `0`, then the value of the parameter will be `NULL`, as shown in the following example:

Input

```
<db:InputParameters xmlns:db="…">
    <isTrue>1</isTrue>
</db:InputParameters>
```

Runtime Invocation

```
BEGIN TOPLEVEL$NEEDSWRAPPER (isTrue =>?, value =>?); END; // value =>
NULL
```

To fix this, you can edit the generated SQL file, restoring the default clauses. You should then, run the SQL file to reload the wrapper definitions into the database schema. In addition, you should modify the generated XSD.

Following are the steps to fix the default clause with the wrapper generated for `needsWrapper()`:

1. Change the signature in the following manner in the generated wrapper SQL file, *from:*

   ```
   TOPLEVEL$NEEDSWRAPPER (isTrue INTEGER, value NUMBER)
   ```

   To:

   ```
   TOPLEVEL$NEEDSWRAPPER (isTrue INTEGER, value NUMBER DEFAULT 0)
   ```

2. Reload the modified wrapper SQL file mentioned in the preceding example into the appropriate database schema. For BOOLEAN parameters with DEFAULT clause, you need to map as follows:

   ```
   DEFAULT TRUE(base) to DEFAULT 1 (wrapper)
   DEFAULT FALSE (base) to DEFAULT 0 (wrapper)
   ```

   For example, if the base stored procedure is `PROCEDURE needsWrapper(isTrue BOOLEAN TRUE, value NUMBER DEFAULT 0)`, then the generated wrapper would be `TOPLEVEL$NEEDSWRAPPER (isTrue INTEGER, value NUMBER).`

You should manually fix the store procedure to be `TOPLEVEL$NEEDSWRAPPER (isTrue INTEGER DEFAULT 1, value NUMBER DEFAULT 0)`

3. If a parameter has a default clause, then its corresponding element in the XSD must have an extra attribute, `db:default="true"`. For example, if a parameter has a default clause `TOPLEVEL$NEEDSWRAPPER(isTrue INTEGER DEFAULT 1, value NUMBER DEFAULT 0)`, then the elements in the XSD for `isTrue` and `value` need to have the following new attributes:

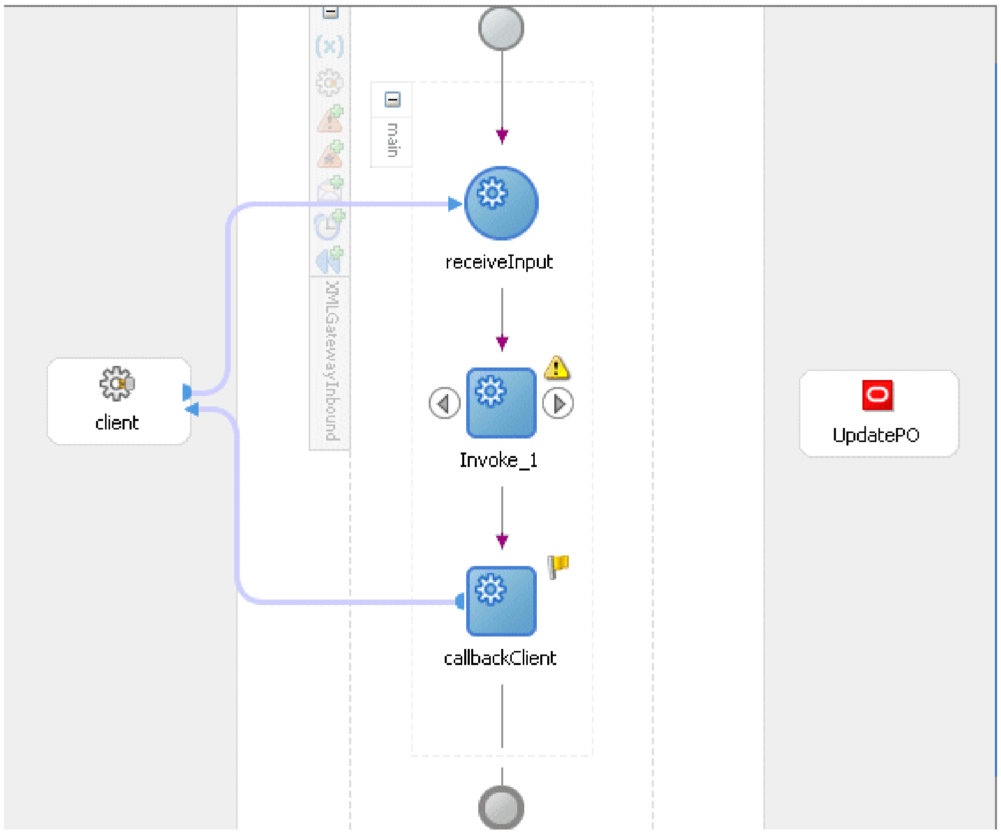
```
<element name="ISTRUE" ... db:default="true" .../>
<element name="VALUE" ... db:default="true" .../>
```

Configuring the Invoke Activity

To configure the Invoke activity:

1. In JDeveloper BPEL Designer, drag and drop the **Invoke** activity from the Component Palette into the process diagram between the receiveInput and callbackClient activities.

Adding the Invoke Activity



2. Link the Invoke activity to the `UpdatePO` service. The Edit Invoke dialog appears.

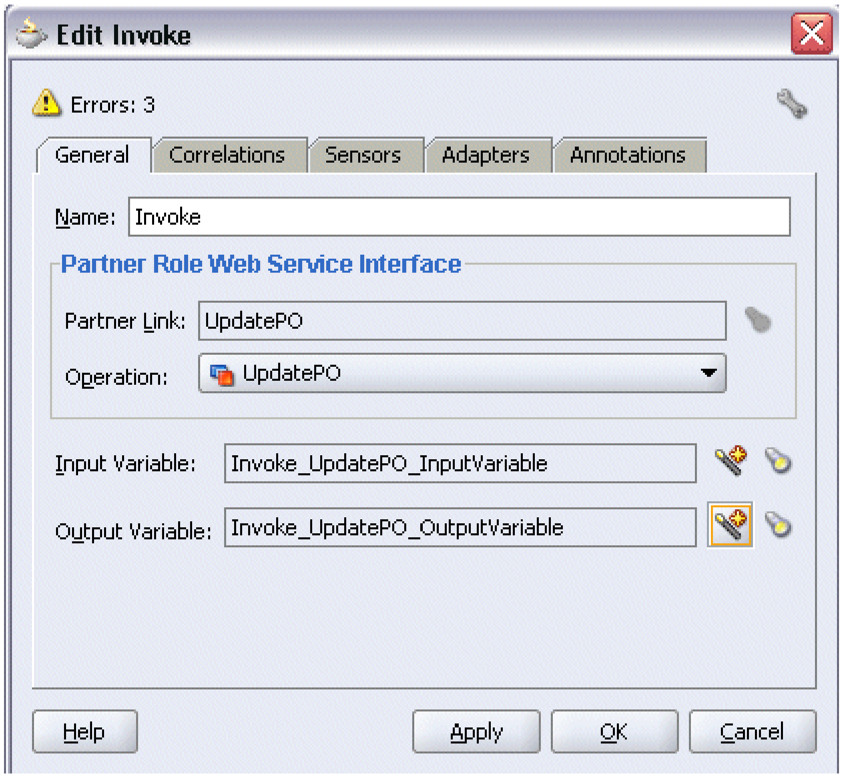
3. In the General tab, enter a name for the Invoke activity and click the **Create** icon next to the **Input Variable** field to create a new variable. The Create Variable dialog box appears.

4. Select **Global Variable** and enter a name for the variable. You can also accept the default name. Click **OK**.

5. In the Edit Invoke dialog,

   click the **Create** icon next to the **Output Variable** field. Select **Global Variable** and enter a name for the variable. You can also accept the default name. Click **OK**.

Edit Invoke Dialog


Edit Invoke

Errors: 3

Name: Invoke

Partner Role Web Service Interface

Partner Link: UpdatePO

Operation: UpdatePO

Input Variable: Invoke_UpdatePO_InputVariable

Output Variable: Invoke_UpdatePO_OutputVariable

6. In the Edit Invoke dialog box, click **Apply** and then **OK**.

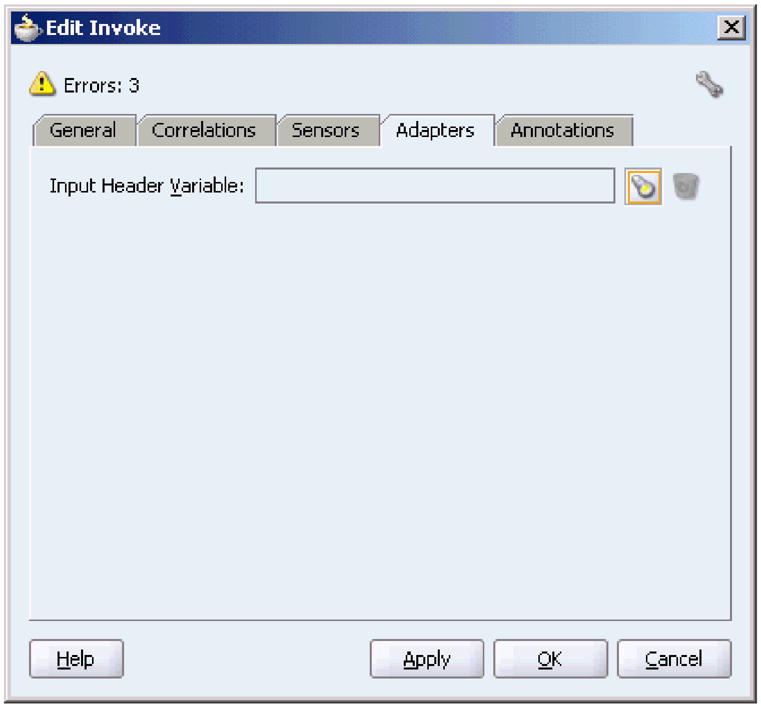
Note: Declaring Header Variables

You can define an **Input Header Variable** under the **Adapters** tab of the Edit Invoke dialog box. This variable can be used to provide applications context information required in the SOA Suite to process Concurrent Programs and PL/SQL APIs and to support multiple languages and multiple organization setups.

To Create the Header Variable:

1. Click on the Adapters tab in the Edit Invoke dialog box and click the **Browse Variable...** icon for the Input Header Variable field.

Specifying Input Header Variable



Input Header Variable:

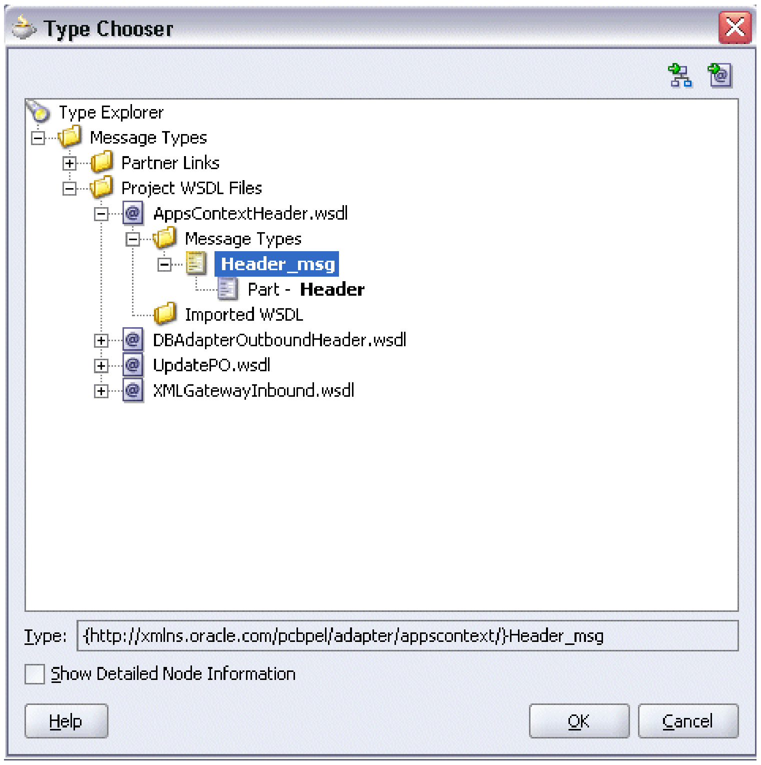
2. In the Variable Chooser dialog box, right-click on the Variables node and select Create Variable option from the menu.

3. In the Name field, enter `header` for the new variable.

4. Select **Message Type** and click **Browse Message Type...** icon to open Type Chooser dialog box.

5. Expand the Partner Link node to locate the Header_msg node `{http://xmlns.oracle.com/pcbpel/adapter/appscontext/}Header_msg` for your partner link. The Header_msg node should be under the path, *Your Partner Link WSDL* > AppsContextHeader.wsdl > Message Types > Header_msg.

Declaring Header Variable

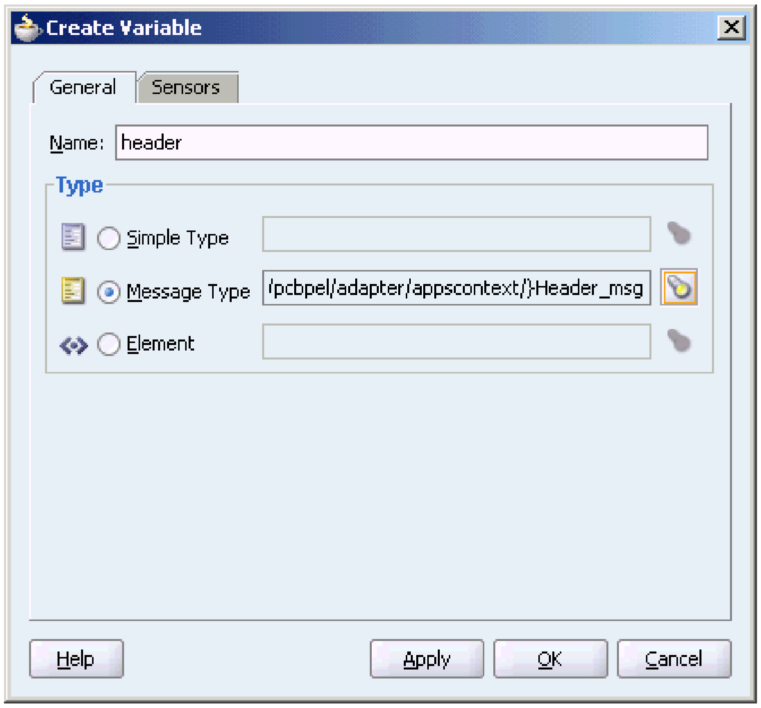


Type: {http://xmlns.oracle.com/pcbpel/adapter/appscontext/}Header_msg

Show Detailed Node Information

6. Click **OK** to return to the Create Variable dialog box with your selected message type populated.

Populating Selected Header Variable


Create Variable

General | Sensors

Name: header

Type

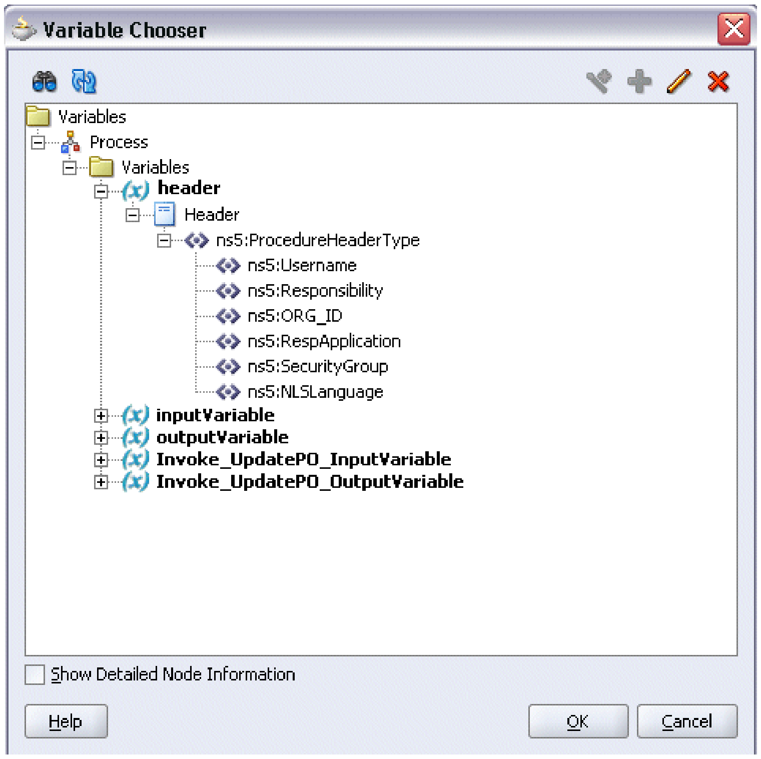
( ) Simple Type

[✓ (•) Message Type] /pcbpel/adapter/appscontext/}Header_msg

( ) Element

7. To view your header variable including Username, Responsibility, ORG_ID, Responsibility Application, Security Group, and NLS Language, click the **Browse Message Type...** icon to open Variable Chooser dialog box. Locate the header variable to view the variable hierarchical structure with these parameters needed for applications context.

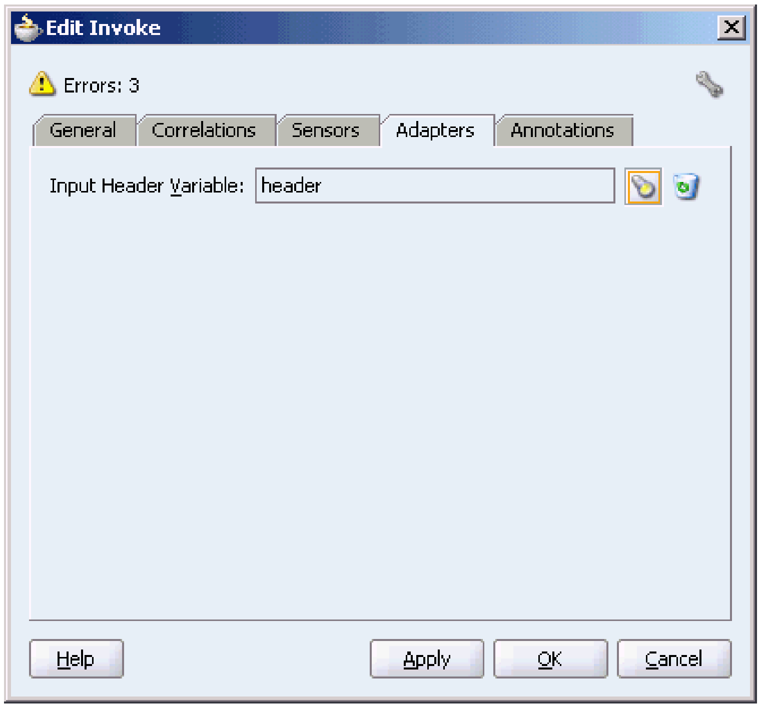
Viewing Header Variable Structure


Variable Chooser

Variables
Process
Variables
header
Header
ns5:ProcedureHeaderType
ns5:Username
ns5:Responsibility
ns5:ORG_ID
ns5:RespApplication
ns5:SecurityGroup
ns5:NLSLanguage
inputVariable
outputVariable
Invoke_UpdatePO_InputVariable
Invoke_UpdatePO_OutputVariable

8. Click **OK** to return to the Edit Invoke dialog with the selected header variable populated for the Input Header Variable field. Click **Apply** to complete the header creation.

Populating Input Header Variable



Input Header Variable: header

Configuring the Assign Activity

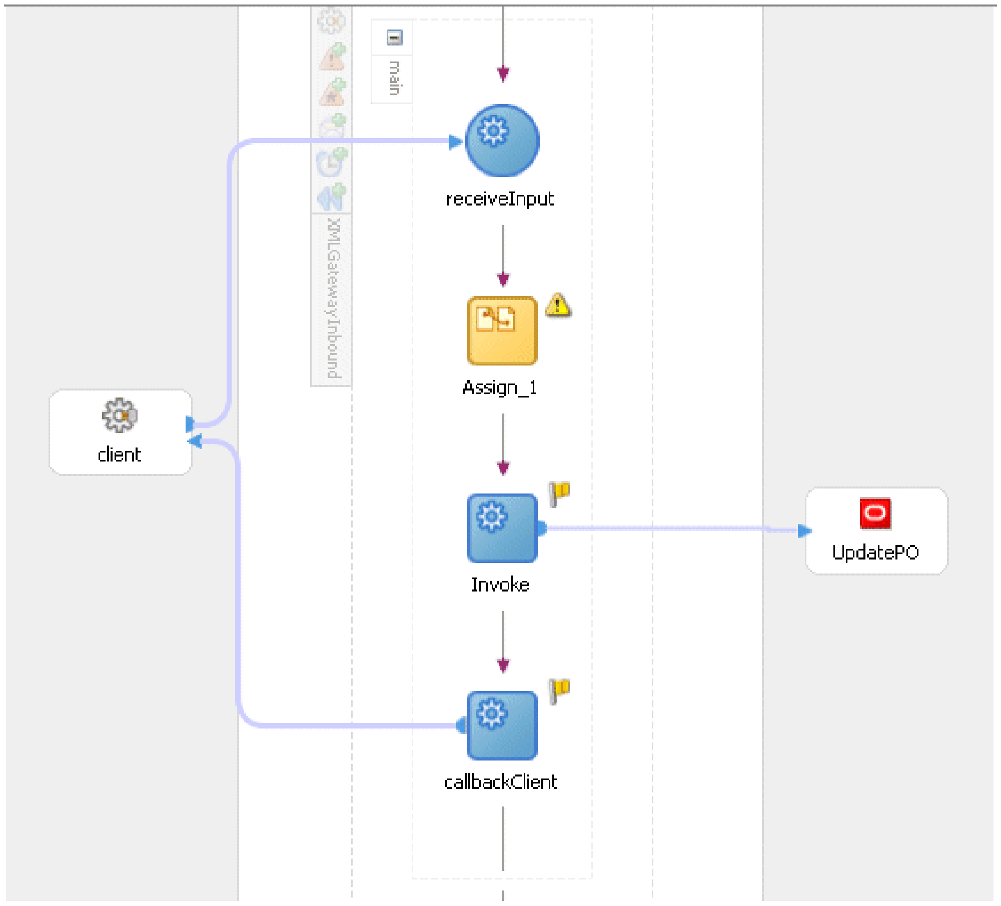
This step is to configure two Assign activities in order to:

- Set the applications context information.
- Set the payload for the `UpdatePO` service.

To add the first Assign activity to set applications context information:

1. In JDeveloper BPEL Designer, drag and drop the **Assign** activity from the Component Palette into the process map before the **Invoke** activity.

Adding an Assign Activity



2. Double-click the **Assign** activity to access the Edit Assign dialog box.

3. Click the General tab to enter the name for the Assign activity, such as 'SetAppsContext'.

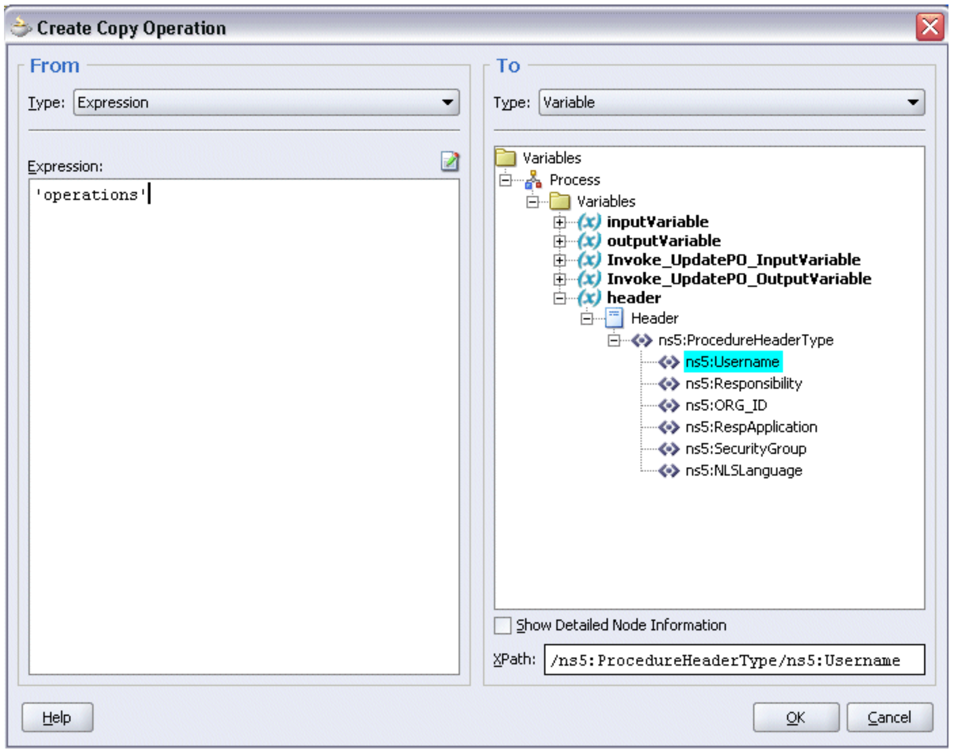
4. On the Copy Operation tab, click **Create** and select **Copy Operation** from the menu. The Create Copy Operation window appears.

5. Enter the first pair of parameters:

   - In the From navigation tree, select type Expression and enter `'operations'` in the Expression box.

   - In the To navigation tree, select type Variable. Navigate to **Variable > Process > Variables > header > Header> ns5:ProcedureHeaderType** and select **ns5:Username**. The XPath field should contain your selected entry.

Assign Username Parameter



Expression:

```
'operations'
```

XPath: `/ns5:ProcedureHeaderType/ns5:Username`

- Click **OK**.

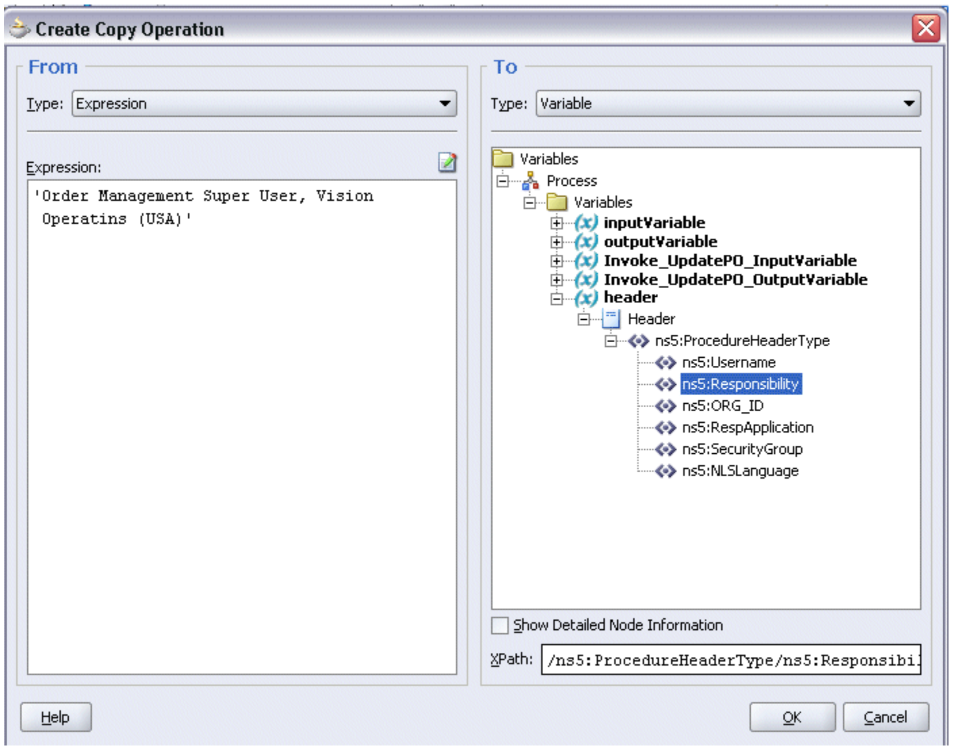
6. Enter the second pair of parameters:

   On the Copy Operation tab, click **Create** and select **Copy Operation** from the menu. The Create Copy Operation window appears.

   - In the From navigation tree, select type Expression and enter `'Order Management Super User, Vision Operatins (USA)'` in the Expression box.

   - In the To navigation tree, select type Variable. Navigate to **Variable > Process > Variables > header > Header> ns5:ProcedureHeaderType** and select **ns5:Responsibility**. The XPath field should contain your selected entry.

Assign Responsibility Parameter



Type: Expression

Expression:

```
'Order Management Super User, Vision
 Operatins (USA)'
```

Type: Variable

```
Variables
  Process
    Variables
      inputVariable
      outputVariable
      Invoke_UpdatePO_InputVariable
      Invoke_UpdatePO_OutputVariable
      header
        Header
          ns5:ProcedureHeaderType
            ns5:Username
            ns5:Responsibility
            ns5:ORG_ID
            ns5:RespApplication
            ns5:SecurityGroup
            ns5:NLSLanguage
```

Show Detailed Node Information

XPath: /ns5:ProcedureHeaderType/ns5:Responsibi

- Click **OK**.

7. Enter the third pair of parameters:

   On the Copy Operation tab, click **Create** and select **Copy Operation** from the menu. The Create Copy Operation window appears.

   - In the From navigation tree, select type Expression and enter `'207'` in the Expression box.

   - In the To navigation tree, select type Variable. Navigate to **Variable > Process > Variables > header > Header> ns5:ProcedureHeaderType** and select **ns5:ORG_ID**. The XPath field should contain your selected entry.

   - Click **OK**.

8. Enter the fourth pair of parameters:

   On the Copy Operation tab, click **Create** and select **Copy Operation** from the menu. The Create Copy Operation window appears.

   - In the From navigation tree, select type Expression and enter `'ONT'` in the

Expression box.

- In the To navigation tree, select type Variable. Navigate to **Variable > Process > Variables > header > Header> ns5:ProcedureHeaderType** and select **ns5:RespApplication**. The XPath field should contain your selected entry.

- Click **OK**.

9. Enter the fifth pair of parameters:

   On the Copy Operation tab, click **Create** and select **Copy Operation** from the menu. The Create Copy Operation window appears.

   - In the From navigation tree, select type Expression and enter `'Standard'` in the Expression box.

   - In the To navigation tree, select type Variable. Navigate to **Variable > Process > Variables > header > Header> ns5:ProcedureHeaderType** and select **ns5:SecurityGroup**. The XPath field should contain your selected entry.

   - Click **OK**.

10. Enter the sixth pair of parameters:

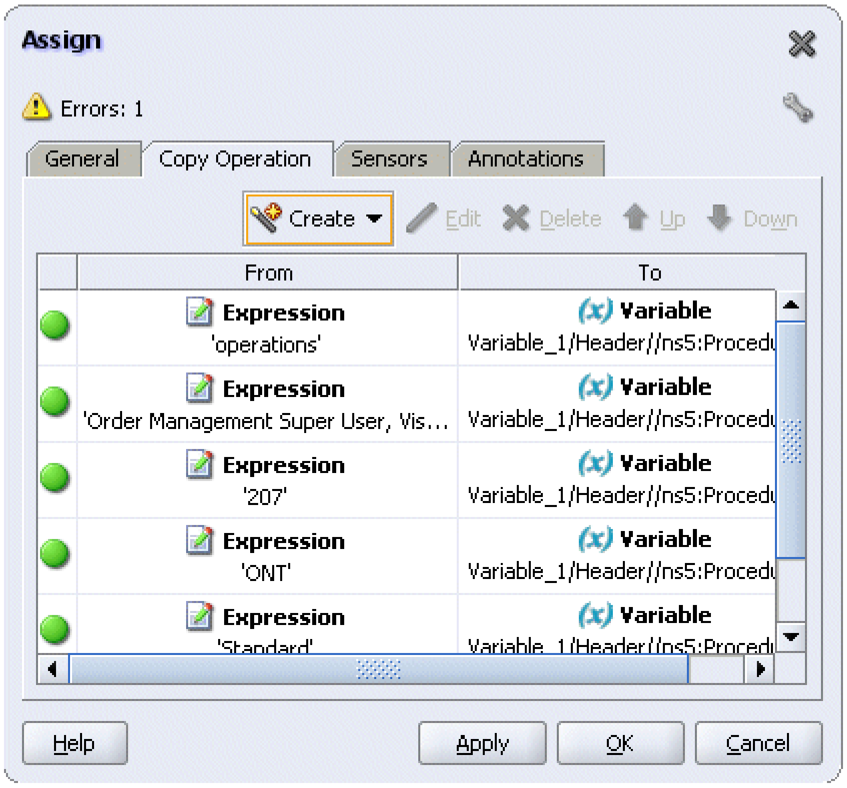
    On the Copy Operation tab, click **Create** and select **Copy Operation** from the menu. The Create Copy Operation window appears.

    - In the From navigation tree, select type Expression and enter `'US'` in the Expression box.

    - In the To navigation tree, select type Variable. Navigate to **Variable > Process > Variables > header > Header> ns5:ProcedureHeaderType** and select **ns5:NLSLanguage**. The XPath field should contain your selected entry.

    - Click **OK**.

Assign Activity



	From	To
●	Expression 'operations'	Variable Variable_1/Header//ns5:Procedu
●	Expression 'Order Management Super User, Vis...	Variable Variable_1/Header//ns5:Procedu
●	Expression '207'	Variable Variable_1/Header//ns5:Procedu
●	Expression 'ONT'	Variable Variable_1/Header//ns5:Procedu
●	Expression 'Standard'	Variable Variable_1/Header//ns5:Procedu

11. Click **Apply** and then **OK** to complete the configuration of the Assign activity.

To add the second Assign activity to set payload for the UpdatePO service:

1. In JDeveloper BPEL Designer, drag and drop the **Assign** activity from the Component Palette into the Activity box of the process diagram after the first **Assign** activity.

2. Double-click the **Assign** activity to access the Edit Assign dialog box.

3. Click the General tab to enter the name for the second Assign activity, such as 'SetPayload'.

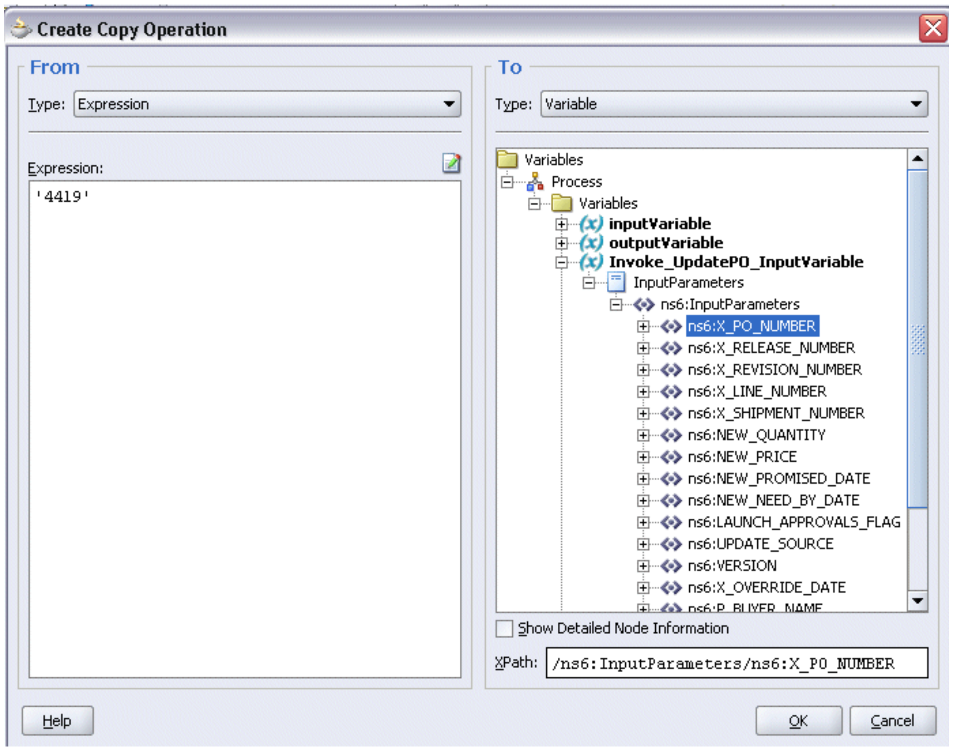
4. On the Copy Operation tab, click **Create** and select **Copy Operation** from the menu. The Create Copy Operation window appears.

5. Enter the first pair of parameters:

   - In the From navigation tree, select type Expression and enter `'4419'` in the Expression box.

- In the To navigation tree, select type Variable. Navigate to **Variable > Process > Variables > Invoke_UpdatePO_InputVariable > InputParameters> ns6:InputParameters** and select **ns6:X_PO_NUMBER**. The XPath field should contain your selected entry.

Assign PO Number Parameter



Expression: `'4419'`

XPath: `/ns6:InputParameters/ns6:X_PO_NUMBER`

- Click **OK**.

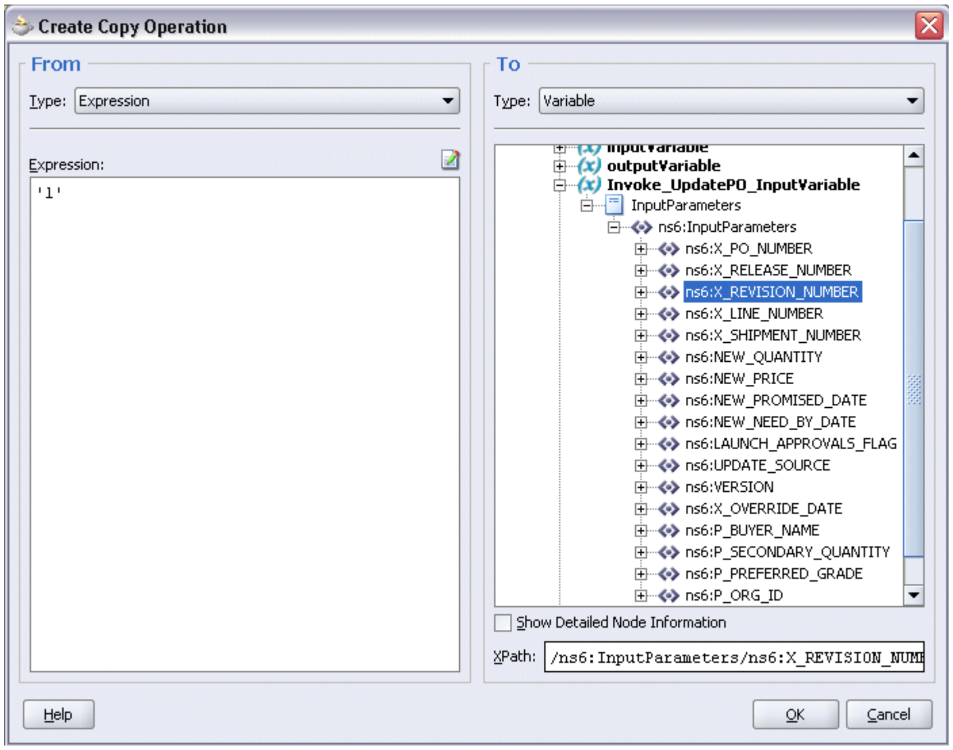
6. Enter the second pair of parameters:

   On the Copy Operation tab, click **Create** and select **Copy Operation** from the menu. The Create Copy Operation window appears.

   - In the From navigation tree, select type Expression and enter `'1'` in the Expression box.

   - In the To navigation tree, select type Variable. Navigate to **Variable > Process > Variables > Invoke_UpdatePO_InputVariable > InputParameters> ns6:InputParameters** and select **ns6:X_REVISION_NUMBER**. The XPath field should contain your selected entry.

Assign Revision Number Parameter



From — Type: Expression

Expression:

```
'1'
```

To — Type: Variable

```
inputVariable
outputVariable
Invoke_UpdatePO_InputVariable
  InputParameters
    ns6:InputParameters
      ns6:X_PO_NUMBER
      ns6:X_RELEASE_NUMBER
      ns6:X_REVISION_NUMBER
      ns6:X_LINE_NUMBER
      ns6:X_SHIPMENT_NUMBER
      ns6:NEW_QUANTITY
      ns6:NEW_PRICE
      ns6:NEW_PROMISED_DATE
      ns6:NEW_NEED_BY_DATE
      ns6:LAUNCH_APPROVALS_FLAG
      ns6:UPDATE_SOURCE
      ns6:VERSION
      ns6:X_OVERRIDE_DATE
      ns6:P_BUYER_NAME
      ns6:P_SECONDARY_QUANTITY
      ns6:P_PREFERRED_GRADE
      ns6:P_ORG_ID
```

Show Detailed Node Information

XPath: /ns6:InputParameters/ns6:X_REVISION_NUMB

- Click **OK**.

7. Enter the third pair of parameters:

   On the Copy Operation tab, click **Create** and select **Copy Operation** from the menu. The Create Copy Operation window appears.

   - In the From navigation tree, select type Expression and enter `'1'` in the Expression box.

   - In the To navigation tree, select type Variable. Navigate to **Variable > Process > Variables > Invoke_UpdatePO_InputVariable > InputParameters> ns6:InputParameters** and select **ns6:X_LINE_NUMBER**. The XPath field should contain your selected entry.

   - Click **OK**.

8. Enter the fourth pair of parameters:

   On the Copy Operation tab, click **Create** and select **Copy Operation** from the menu. The Create Copy Operation window appears.

- In the From navigation tree, select type Expression and enter `'6'` in the Expression box.

  This is the new quantity number.

- In the To navigation tree, select type Variable. Navigate to **Variable > Process > Variables > Invoke_UpdatePO_InputVariable > InputParameters> ns6:InputParameters** and select **ns6:NEW_QUANTITY**. The XPath field should contain your selected entry.

- Click **OK**.

9. Enter the fifth pair of parameters:

   On the Copy Operation tab, click **Create** and select **Copy Operation** from the menu. The Create Copy Operation window appears.

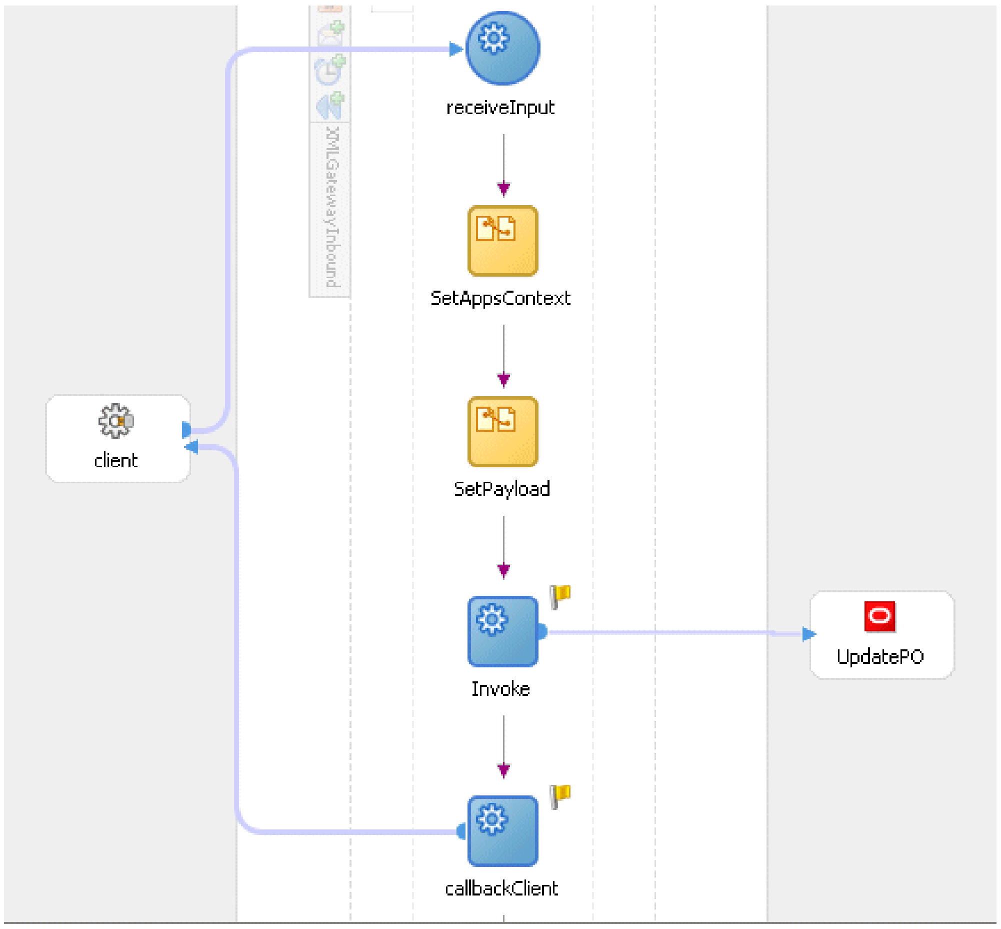
   - In the From navigation tree, select type Expression and enter `'1.0'` in the Expression box.

   - In the To navigation tree, select type Variable. Navigate to **Variable > Process > Variables > Invoke_UpdatePO_InputVariable > InputParameters> ns6:InputParameters** and select **ns6:VERSION**. The XPath field should contain your selected entry.

   - Click **OK**.

10. Click **Apply** and then **OK** to complete the configuration of the second Assign activity.

BPEL Process Diagram



Run-Time Tasks for PL/SQL APIs

After designing the BPEL process, the next step is to deploy, run and monitor it.

1. Deploy the BPEL Process, page 8-35
2. Test the BPEL Process, page 8-36

Deploying the BPEL Process

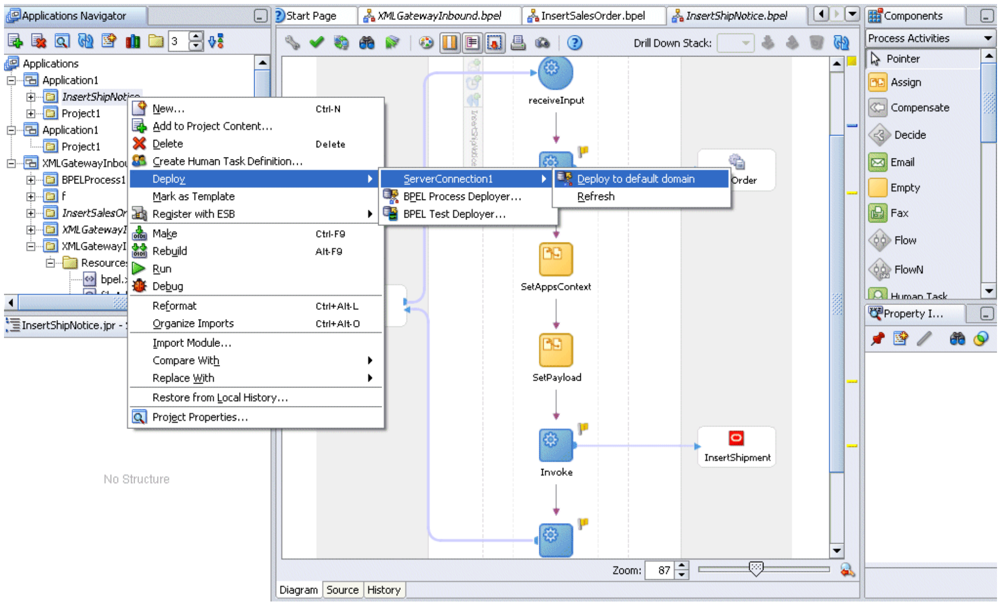
You must deploy the BPEL process before you can run it. The BPEL process is first compiled, and then deployed to the BPEL server.

To deploy the BPEL process:

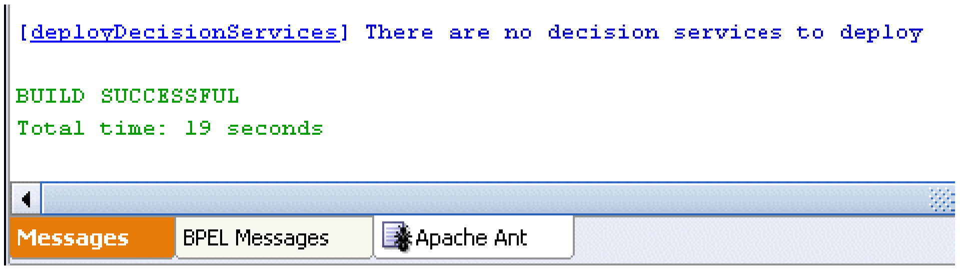
1. Select the BPEL project in the Applications window.

2. Right-click the project name, and then select **Deploy > [Server Connection] > Deploy to Default Domain** from the menu that appears.

Deploy the BPEL Process



3. The BPEL process is compiled and deployed. You can check if the deployment is successful in the Apache Ant log.



```
[deployDecisionServices] There are no decision services to deploy

BUILD SUCCESSFUL
Total time: 19 seconds
```

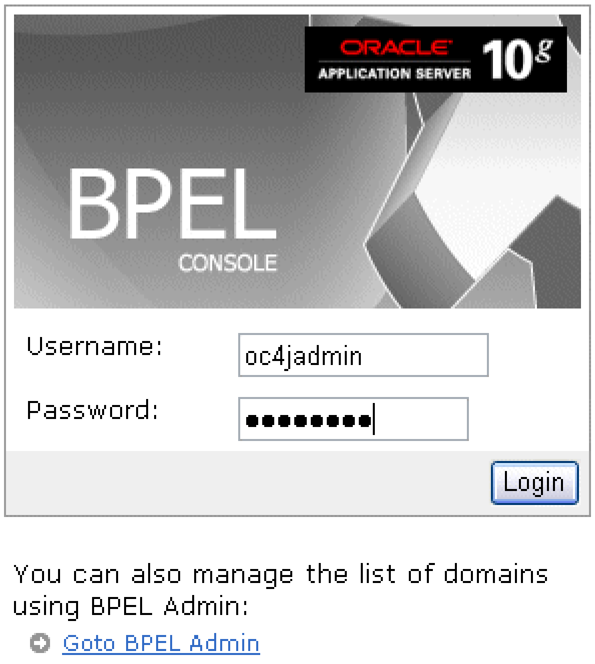
Testing the BPEL Process

Once the BPEL process is deployed, you can manage and monitor the process from the BPEL Console. You can also test the process and the integration interface by manually initiating the process.

To test the BPEL process:

1. Navigate to Oracle Application Server 10*g* BPEL Console (`http://<soaSuiteServerHostName>:<port>/BPELConsole`).

   The BPEL Console login page appears.



Username: oc4jadmin

Password: ●●●●●●●●●

You can also manage the list of domains using BPEL Admin:

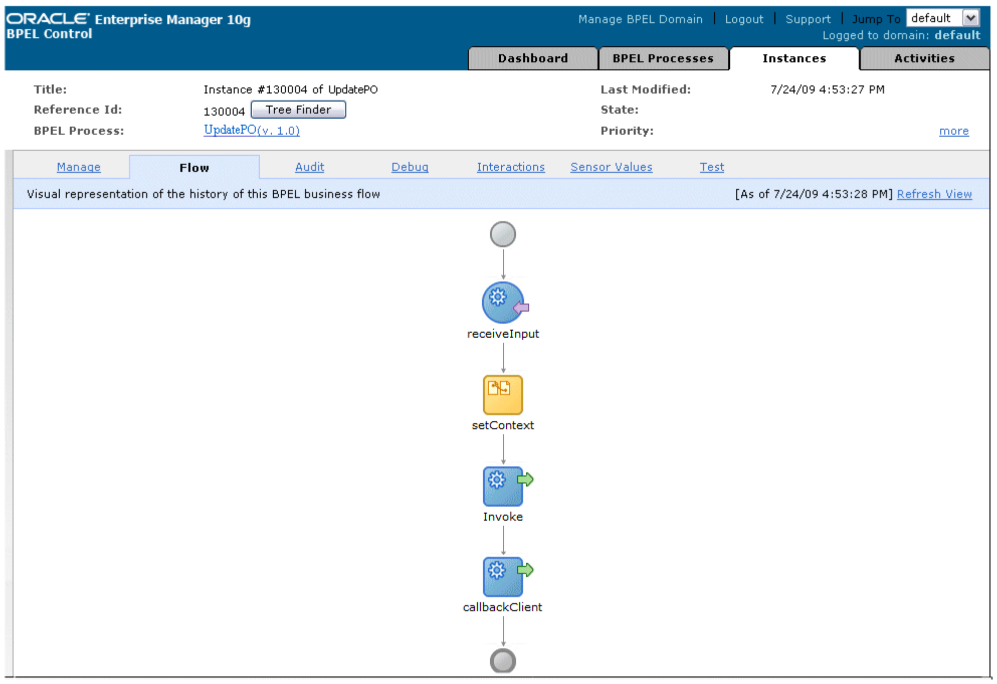
2. Enter the username and password and click **Login**.

3. The Oracle Enterprise Manager 10*g* BPEL Control appears. The list of deployed processes is shown under Deployed BPEL Processes.

4. Click the 'UpdatePO' BPEL process that you want to initiate. The Initiate page appears. Enter the input string required by the process.

5. Click **Post XML Message** to initiate the process.

6. The BPEL process is now initiated. You can check the process flow by clicking the **Visual Flow** icon. This shows the flow of the BPEL process.


ORACLE Enterprise Manager 10g BPEL Control

Manage BPEL Domain | Logout | Support | Jump To default
Logged to domain: default

Dashboard | BPEL Processes | Instances | Activities

Title: Instance #130004 of UpdatePO
Reference Id: 130004 Tree Finder
BPEL Process: UpdatePO(v. 1.0)
Last Modified: 7/24/09 4:53:27 PM
State:
Priority:
more

Manage | Flow | Audit | Debug | Interactions | Sensor Values | Test

Visual representation of the history of this BPEL business flow
[As of 7/24/09 4:53:28 PM] Refresh View

7. Click the Invoke activity in the flow.

8. Notice that <UPDATE_PO>1</UPDATE_PO> in the output parameters. This shows that the update is successful.

9. To validate the BPEL process, you can log on to Oracle Applications forms to check if the quantity field is updated in the Purchase Order form.

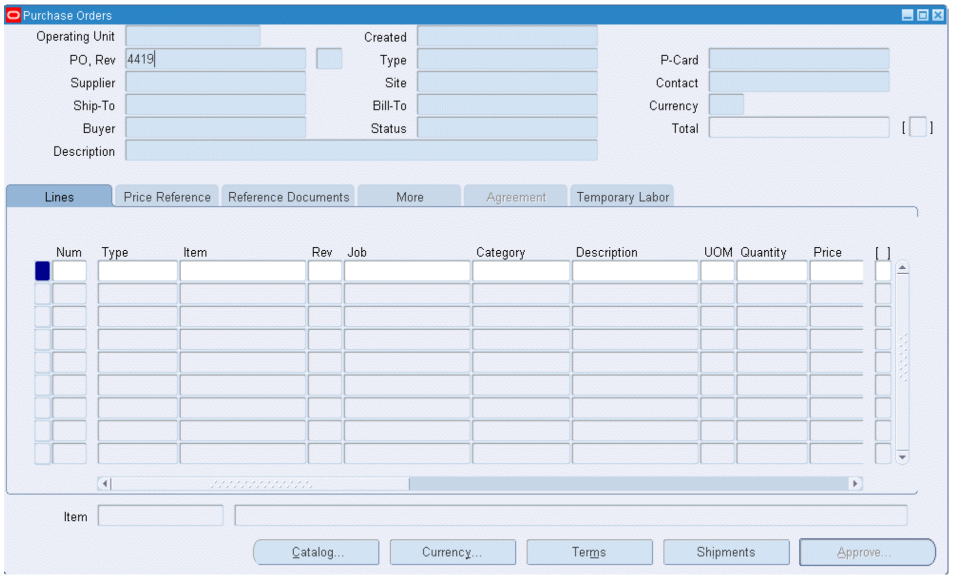
    1. Log in to Oracle E-Business Suite with the Order Management Super User, Vision Operations (USA) responsibility.

    2. Select **Purchasing: Purchase Orders > Purchase Orders** to open the Purchase Orders form.

    3. Query your purchase order number by selecting **View > Query by Example > Enter**.

    4. In the PO, Rev field, enter your order number '4419'.

Query an Existing Order

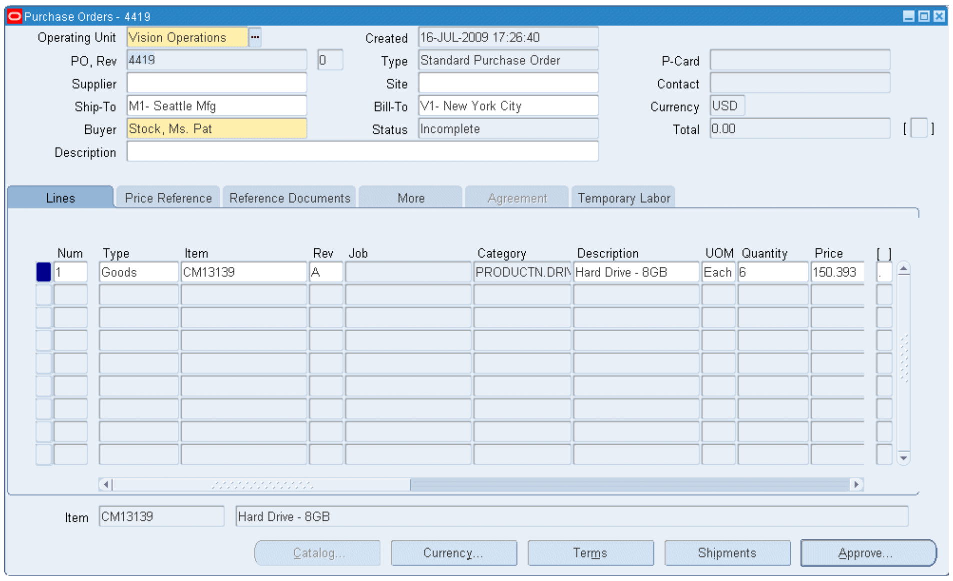


PO, Rev: 4419

5. Execute the query by selecting **View > Query by Example > Run**.

6. You should find the quantity number is updated to number '6'.

Locating an Existing Order



Purchase Orders - 4419

Operating Unit	Vision Operations	Created	16-JUL-2009 17:26:40		
PO, Rev	4419 0	Type	Standard Purchase Order	P-Card	
Supplier		Site		Contact	
Ship-To	M1- Seattle Mfg	Bill-To	V1- New York City	Currency	USD
Buyer	Stock, Ms. Pat	Status	Incomplete	Total	0.00
Description					

Num	Type	Item	Rev	Job	Category	Description	UOM	Quantity	Price
1	Goods	CM13139	A		PRODUCTN.DRI\	Hard Drive - 8GB	Each	6	150.393

Item CM13139 Hard Drive - 8GB

Troubleshooting and Debugging

If you experience problems with your PL/SQL API integration, you can take the following troubleshooting steps:

- If you designed your PL/SQL API integration in one instance of your Oracle Applications environment, and plan to run it in another instance, be sure to rerun the SQL scripts in the second instance to create the necessary PL/SQL wrappers.
- Ensure that the JNDI location in **oc4j-ra.xml** is properly configured.
- Use **encrypt.bat** to encrypt the passwords in **oc4j-ra.xml** if needed.

If you still experience problems with your integration, you can enable debugging.

Enabling Debugging

You can enable debugging for PL/SQL APIs using the BPEL Process Manager.

To enable debugging:

1. Log into your BPEL Process Manager domain.
2. Select `*yourdomain*.collaxa.cube.ws`
3. Select **Debug**.

Debugging information is output to the log file for your domain. To examine the log file in the BPEL Process Manager, navigate to **Home > BPEL Domains >*yourdomain* > Logs**. The log file is ***yourdomain*.log**.

9

Using e-Commerce Gateway for BPEL Process Integration

This chapter covers the following topics:

- Overview of e-Commerce Gateway Integration
- Design-Time Tasks for e-Commerce Gateway
- Creating a New BPEL Project
- Adding a Partner Link
- Adding a Partner Link for File Adapter
- Configuring the Invoke Activity
- Configuring the Assign Activity
- Run-Time Tasks for e-Commerce Gateway
- Deploying the BPEL Process
- Testing the BPEL Process
- Verifying Records in Oracle Applications

Overview of e-Commerce Gateway Integration

Oracle e-Commerce Gateway provides a common, standards-based approach for Electronic Data Interchange (EDI) integration between Oracle Applications and third party applications. EDI is the computer to computer exchange of business documents in a standard format. The EDI format is commonly used for e-commerce transactions between businesses.

Oracle e-Commerce Gateway is the EDI integration enabler for Oracle Applications. It provides a single integration framework for you to conduct e-business using EDI standards with everyone in your global supply chain. Oracle e-Commerce Gateway provides an application integration infrastructure that is flexible enough to accommodate the integration requirements of any and all applications that must integrate with Oracle Applications. This allows for seamless flow of information in an ever expanding trading partner base.

Oracle e-Commerce Gateway includes pre-built transactions of key business documents that can be implemented simply by defining a trading partner and enabling the transaction in test or production mode. You can implement a single transaction, a group of transactions, or a business flow. You can implement the pre-built transactions as is or configure them to meet your specific industry needs.

Oracle e-Commerce Gateway uses a metadata driven approach to dynamically generate outbound and consume inbound flat files based on user defined trading partner, mapping, transformation, and data validation rules. You can change a rule by changing the metadata stored in the repository. The updated rule takes effect at run-time without any code modifications.

> **Note:** For detailed information about Oracle e-Commerce Gateway, see *Oracle e-Commerce Gateway User's Guide.* This guide is a part of the Oracle Applications documentation library. Oracle Applications documentation can be accessed from the following link:
>
> http://www.oracle.com/technology/documentation/applications.html

OracleAS Adapter for Oracle Applications can be configured to use e-Commerce Gateway to interact with third party applications. e-Commerce Gateway, like XML Gateway, is primarily used for Business-to-Business (B2B) integration. While XML transactions are mostly based on a single transaction and are event based, EDI transactions are more batch oriented.

Design-Time Tasks for e-Commerce Gateway

OracleAS adapter for Oracle Applications is deployed using the BPEL Process Manager (PM) in Oracle JDeveloper. The BPEL PM creates the WSDL interfaces for the e-Commerce Gateway.

This section describes configuring the OracleAS Adapter for Oracle Applications to use e-Commerce Gateway. It describes the tasks required to configure OracleAS Adapter for Oracle Applications using the Adapter Configuration Wizard in Oracle JDeveloper.

Prerequisites to Configuring e-Commerce Gateway

Populating Applications Context Header Variables

You need to populate certain variables in the SOA Suite to provide context information required in EDI/concurrent programs in order for an Oracle Applications user that has sufficient privileges to run the programs.

The context is set taking into account the values passed for the header properties including *Username, Responsibility, Responsibility Application, Security Group*, and *NLS Language*. If the values for the new header properties *Responsibility Application, Security Group*, and *NLS Language* are not passed, context information will be determined based on *Username* and *Responsibility*.

The default value for the *Username* is `SYSADMIN`, the default value for *Responsibility* is `SYSTEM ADMINISTRATOR`, the default *Security Group Key* is `STANDARD`, and the default *NLS Language* is `US`.

You can change the default values specified in the generated WSDL. This is a static way of changing the context information. These values would apply to all invocations of the deployed business process. However, if you need to provide different context information for different invocations of the business process, then you can dynamically populate the header variable with values. The context information can be specified by configuring an Assign activity before the Invoke activity in the BPEL PM.

Following is a list of the procedures required to accomplish the design-time tasks.

1. Create a new BPEL project, page 9-3
2. Add a partner link, page 9-6
3. Add a partner link for File Adapter, page 9-13
4. Configure the Invoke activity, page 9-19
5. Configure the Assign activity, page 9-25

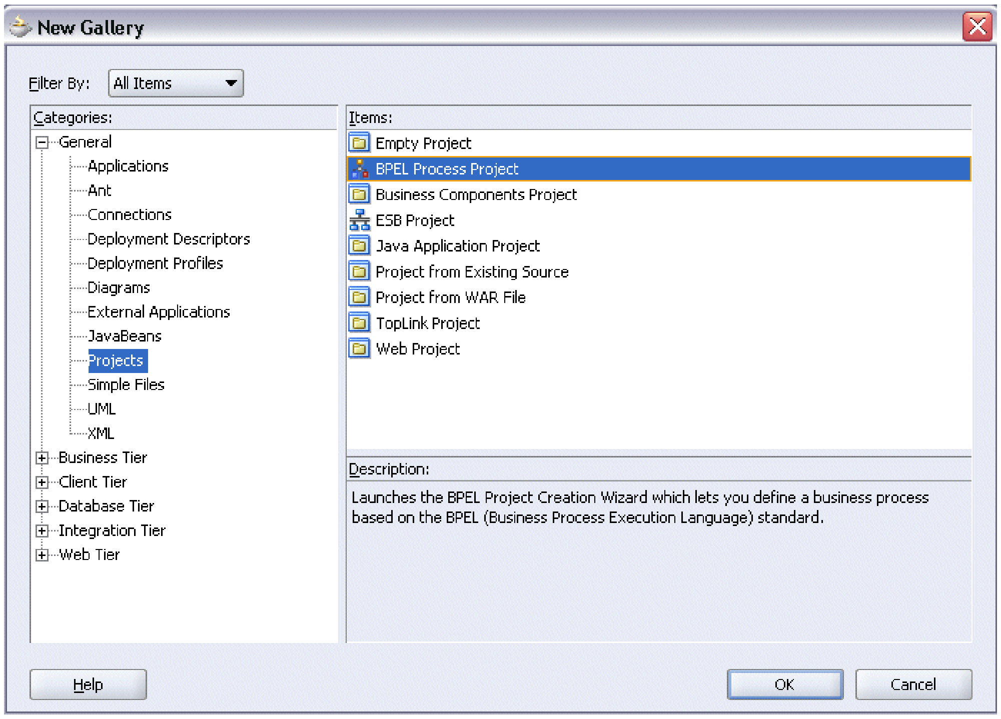
Creating a New BPEL Project

The first configuration task is to create a new BPEL project.

To create a new BPEL project:

1. Open JDeveloper BPEL Designer.
2. From the **File** menu, select **New**. The New Gallery dialog box appears.
3. Select **All Items** from the **Filter By** box. This displays a list of available categories.
4. Expand the **General** node, and then select **Projects**.
5. Select **BPEL Process Project** from the **Items** group.

Creating a New BPEL Process Project

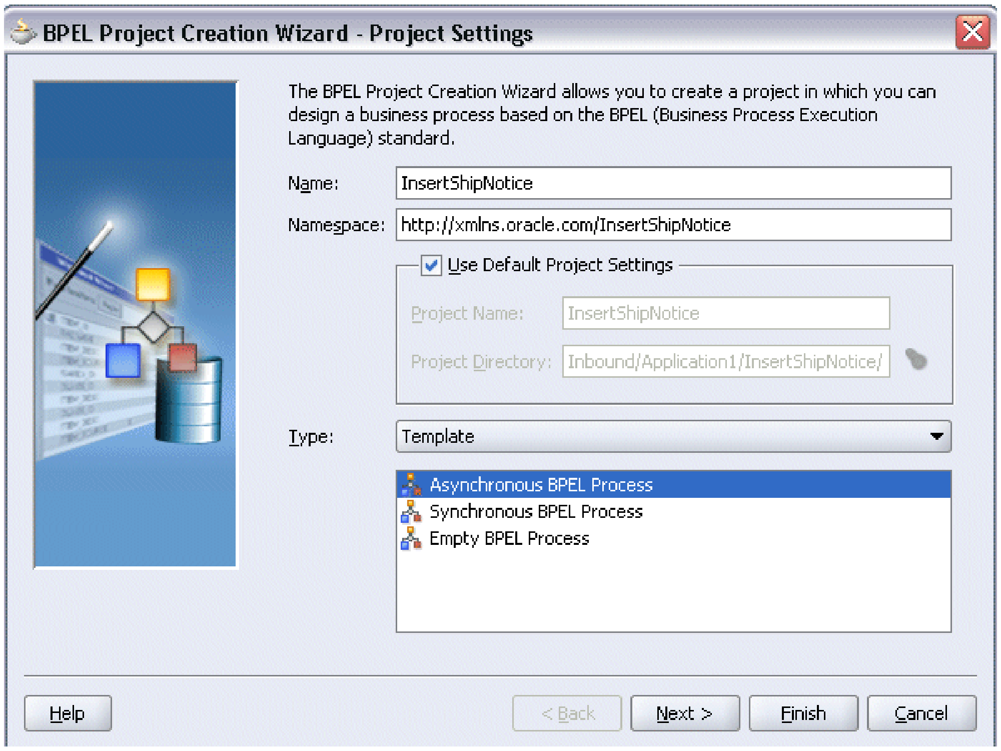


Launches the BPEL Project Creation Wizard which lets you define a business process based on the BPEL (Business Process Execution Language) standard.

6. Click **OK**. The BPEL Project Creation Wizard - Project Settings dialog box appears.
7. In the **Name** field, enter a descriptive name. For example, enter `InsertShipNotice`.

   Keep the default selection **Use Default Project Settings** unchanged.
8. Keep the default selection **Template** as the Type field. Select **Asynchronous BPEL Process** as the BPEL process type.

Specifying a Name for the New BPEL Process Project



BPEL Project Creation Wizard - Project Settings

The BPEL Project Creation Wizard allows you to create a project in which you can design a business process based on the BPEL (Business Process Execution Language) standard.

Name: InsertShipNotice

Namespace: http://xmlns.oracle.com/InsertShipNotice

[✓] Use Default Project Settings

Project Name: InsertShipNotice

Project Directory: Inbound/Application1/InsertShipNotice/

Type: Template

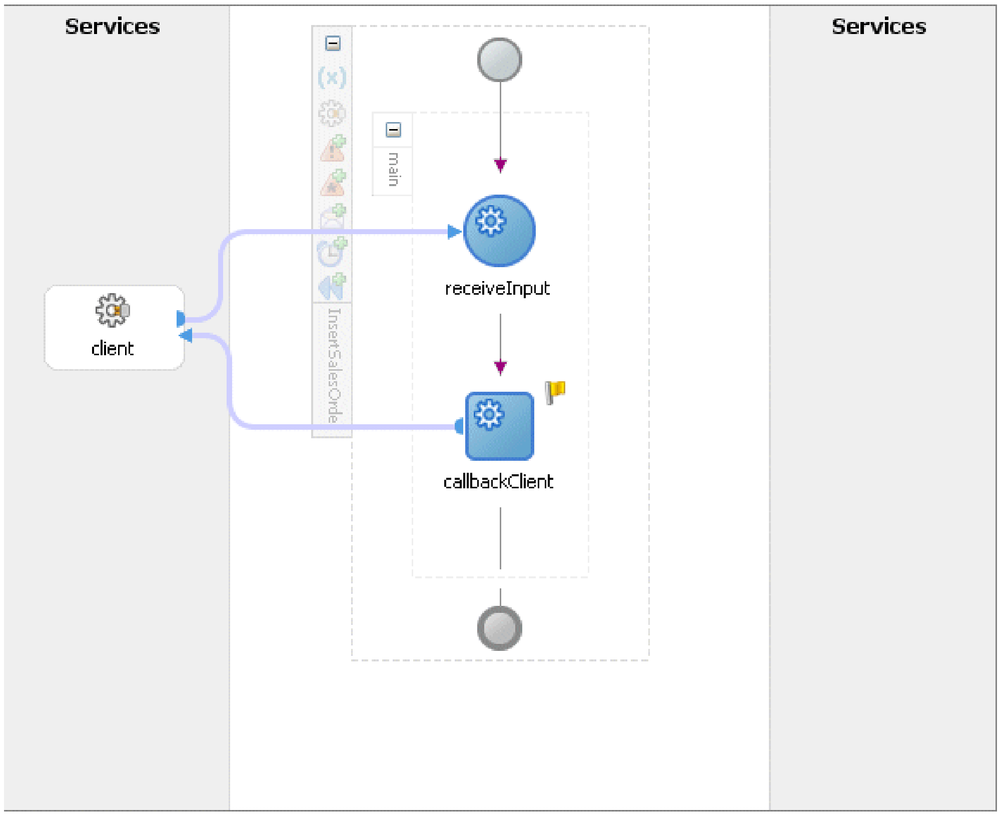
- Asynchronous BPEL Process
- Synchronous BPEL Process
- Empty BPEL Process

9. Click **OK**.

   A new asynchronous BPEL process is created with the receiveInput and callbackClient activities.

   The required source files including `bpel.xml`, `InsertShipNotice.bpel`, and `InsertShipNotice.wsdl` are also generated.

New BPEL Process Project



Adding a Partner Link

A partner link defines the link name, type, and the role of the BPEL process that interacts with the partner service.

To add a partner link:

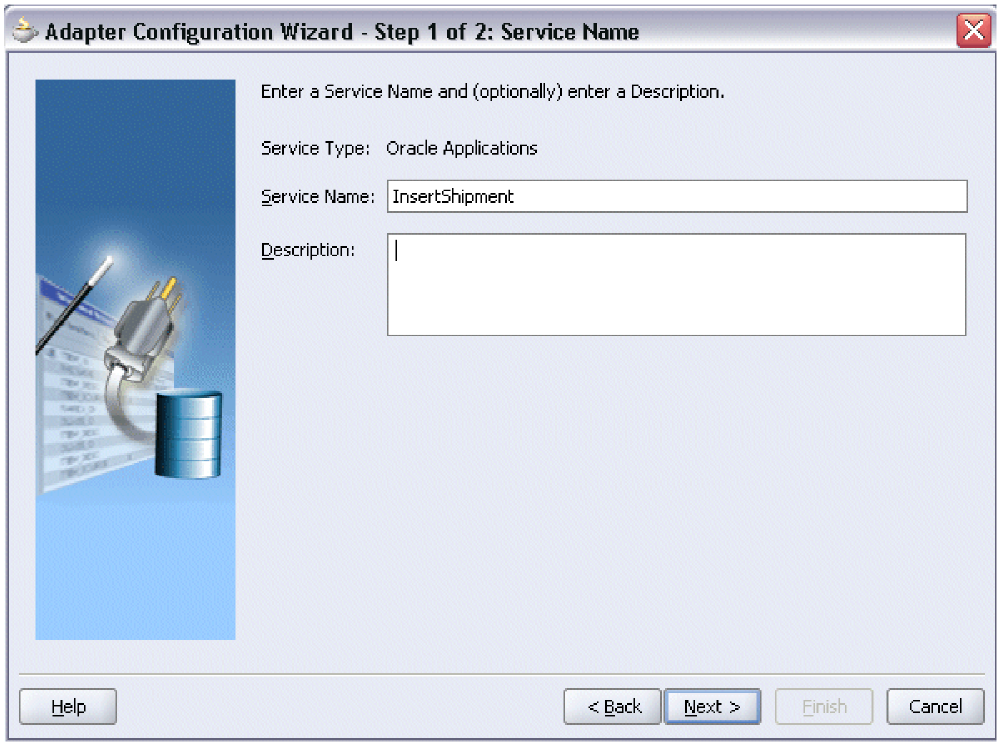
1. In JDeveloper BPEL Designer, click **Services** in the Component palette.

   Drag and drop **Oracle Applications** icon into the border area of the process diagram.

   The Adapter Configuration Wizard appears.

2. Click **Next**. The Service Name dialog box appears. Enter the following information:

Specifying a Service Name


Adapter Configuration Wizard - Step 1 of 2: Service Name

Enter a Service Name and (optionally) enter a Description.

Service Type: Oracle Applications

Service Name: InsertShipment

Description:

1. In the **Service Name** field, enter a service name, such as `InsertShipment`.

2. In the **Description** field, enter a description for the service. This is an optional field.

3. Click **Next**. The Service Connection dialog box appears.

4. You can perform either one of the following options for your database connection:

   Note: You need to connect to the database where Oracle E-Business Suite is running.

   - You can create a new database connection by clicking the **New** icon.

     Detailed instructions on how to create a new database connection, see Creating a New Database Connection, page 4-12.

   - You can select an existing database connection that you have configured earlier from the **Connection** drop-down list.

     The selected database connection information appears in the Service Connection dialog box. The JNDI (Java Naming and Directory Interface) name corresponding to the selected database connection also appears automatically in the **Database Server JNDI Name** field. Alternatively, you can specify a JNDI name.

> **Note:** When you specify a JNDI name, the deployment descriptor of the Adapter for Oracle Applications must associate this JNDI name with configuration properties required by the adapter to access the database.

The JNDI name acts as a placeholder for the connection used when your service is deployed to the BPEL server. This enables you to use different databases for development and later for production.

> **Note:** For more information about JNDI concepts, see *Oracle Application Server Adapter Concepts.*

5. Once you have created a new connection or selected an existing connection, you can select an e-Commerce Gateway interface by browsing through the modules available in Oracle E-Business Suite.

6. Click **Next** in the Service Connection dialog box.

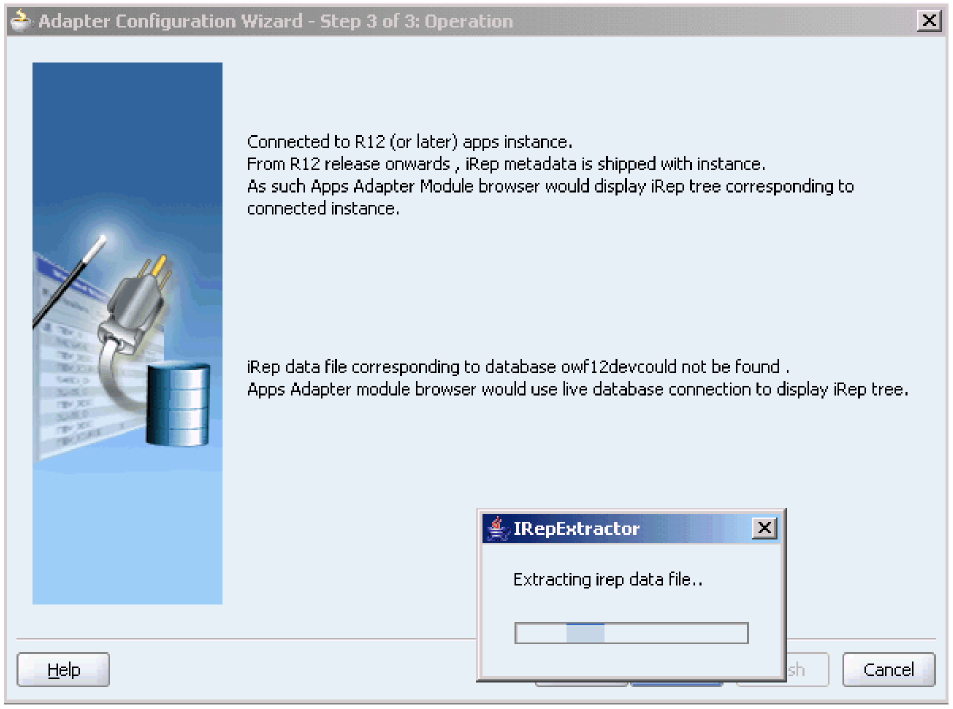
   For Oracle E-Business Suite Release 12:

   If you are connecting to Oracle E-Business Suite Release 12, then the **IREP File not present** dialog box appears indicating that Adapter for Oracle Applications could not find the Oracle Integration Repository data file corresponding to the database you are connecting in your workspace. Absence of the data file would make browsing or searching of Integration Repository tree considerably slow. You can choose to extract the data file and create a local copy of the Integration Repository data file. Once it is created successfully, Adapter for Oracle Applications will pick it up automatically next time and retrieve data from your local Integration Repository.

   You can select one of the following options:

   - Click **Yes** to extract the Integration Repository data file.

Extracting Integration Repository Data File



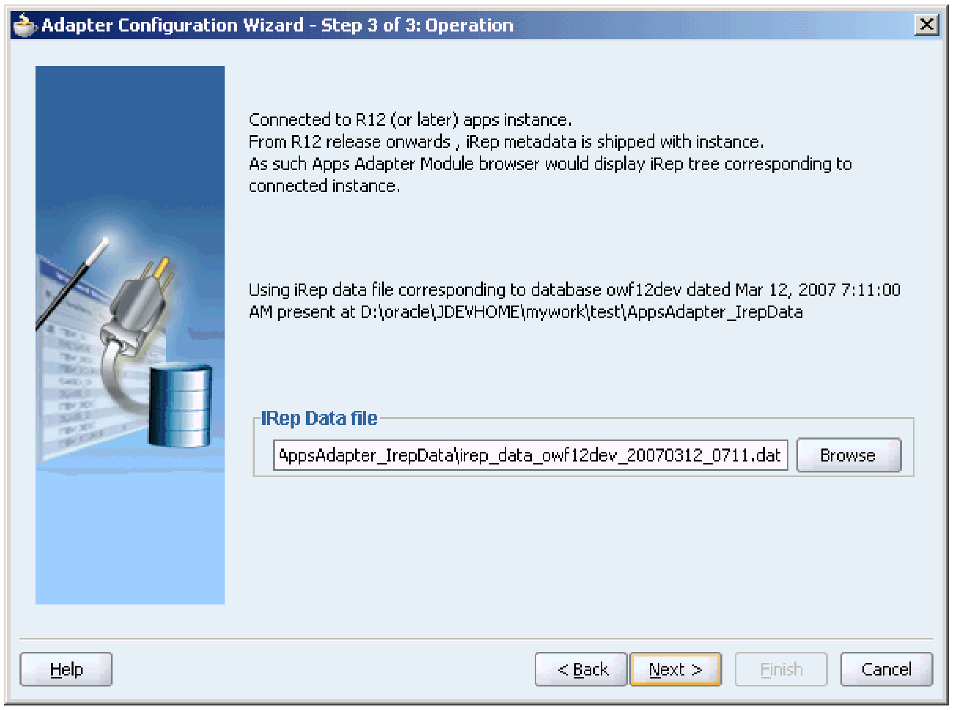
Connected to R12 (or later) apps instance.
From R12 release onwards , iRep metadata is shipped with instance.
As such Apps Adapter Module browser would display iRep tree corresponding to connected instance.

iRep data file corresponding to database owf12devcould not be found .
Apps Adapter module browser would use live database connection to display iRep tree.

Extracting irep data file..

After the system successfully creates a local copy of the Integration Repository data file, next time when you connect to the database, you will find the **IRep Data File** field appears in the Operation dialog box indicating where your local copy exists with the creation date and time as part of the file name.

Using the Local Integration Repository Data File



Connected to R12 (or later) apps instance.
From R12 release onwards , iRep metadata is shipped with instance.
As such Apps Adapter Module browser would display iRep tree corresponding to connected instance.

Using iRep data file corresponding to database owf12dev dated Mar 12, 2007 7:11:00 AM present at D:\oracle\JDEVHOME\mywork\test\AppsAdapter_IrepData

AppsAdapter_IrepData\irep_data_owf12dev_20070312_0711.dat

- Click **No** to query the Integration Repository data file from the live database you are connecting to display the Integration Repository tree.

  Note: It is highly recommended that you create a local copy of the Integration Repository data file so that Adapter for Oracle Applications will query the data next time from the local copy in your workspace to enhance the performance.

Click **Next** in the Operation page to open the Oracle Applications Module Browser.

For Oracle E-Business Suite pre-Release 11.5.10:

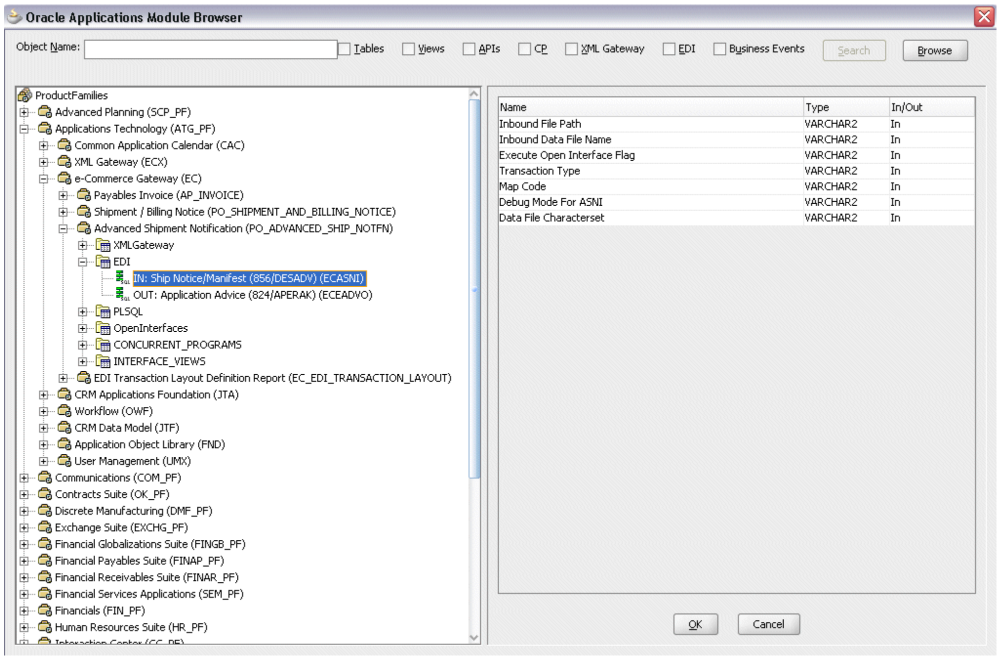
If you are connecting to a pre-11.5.10 Oracle E-Business Suite instance, you must select the interface type in the Adapter Configuration Wizard. Select **Inbound into Oracle Applications** or **Outbound from Oracle Applications** depending on whether data is inbound into Oracle Applications or outbound from Oracle Applications. Next, click **Get CP** to choose the EDI concurrent program from the Oracle Applications Module Browser. Select **EDI Gateway** to proceed.

Click **Add** to open the Oracle Applications Module Browser.

7. The Oracle Applications Module Browser combines interface data from Oracle

Integration Repository with information about the additional interfaces supported by Adapter for Oracle Applications, organized in a tree hierarchy.

Specify an Interface from The Oracle Applications Module Browser



```
ProductFamilies
+ Advanced Planning (SCP_PF)
- Applications Technology (ATG_PF)
    + Common Application Calendar (CAC)
    + XML Gateway (ECX)
    - e-Commerce Gateway (EC)
        + Payables Invoice (AP_INVOICE)
        + Shipment / Billing Notice (PO_SHIPMENT_AND_BILLING_NOTICE)
        - Advanced Shipment Notification (PO_ADVANCED_SHIP_NOTFN)
            + XMLGateway
            - EDI
                IN: Ship Notice/Manifest (856/DESADV) (ECASNI)
                OUT: Application Advice (824/APERAK) (ECEADVO)
            + PLSQL
            + OpenInterfaces
            + CONCURRENT_PROGRAMS
            + INTERFACE_VIEWS
        + EDI Transaction Layout Definition Report (EC_EDI_TRANSACTION_LAYOUT)
    + CRM Applications Foundation (JTA)
    + Workflow (OWF)
    + CRM Data Model (JTF)
    + Application Object Library (FND)
    + User Management (UMX)
+ Communications (COM_PF)
+ Contracts Suite (OK_PF)
+ Discrete Manufacturing (DMF_PF)
+ Exchange Suite (EXCHG_PF)
+ Financial Globalizations Suite (FINGB_PF)
+ Financial Payables Suite (FINAP_PF)
+ Financial Receivables Suite (FINAR_PF)
+ Financial Services Applications (SEM_PF)
+ Financials (FIN_PF)
+ Human Resources Suite (HR_PF)
```

Name	Type	In/Out
Inbound File Path	VARCHAR2	In
Inbound Data File Name	VARCHAR2	In
Execute Open Interface Flag	VARCHAR2	In
Transaction Type	VARCHAR2	In
Map Code	VARCHAR2	In
Debug Mode For ASNI	VARCHAR2	In
Data File Characterset	VARCHAR2	In

Select an inbound or outbound EDI program. You can select only one EDI program for each adapter service.

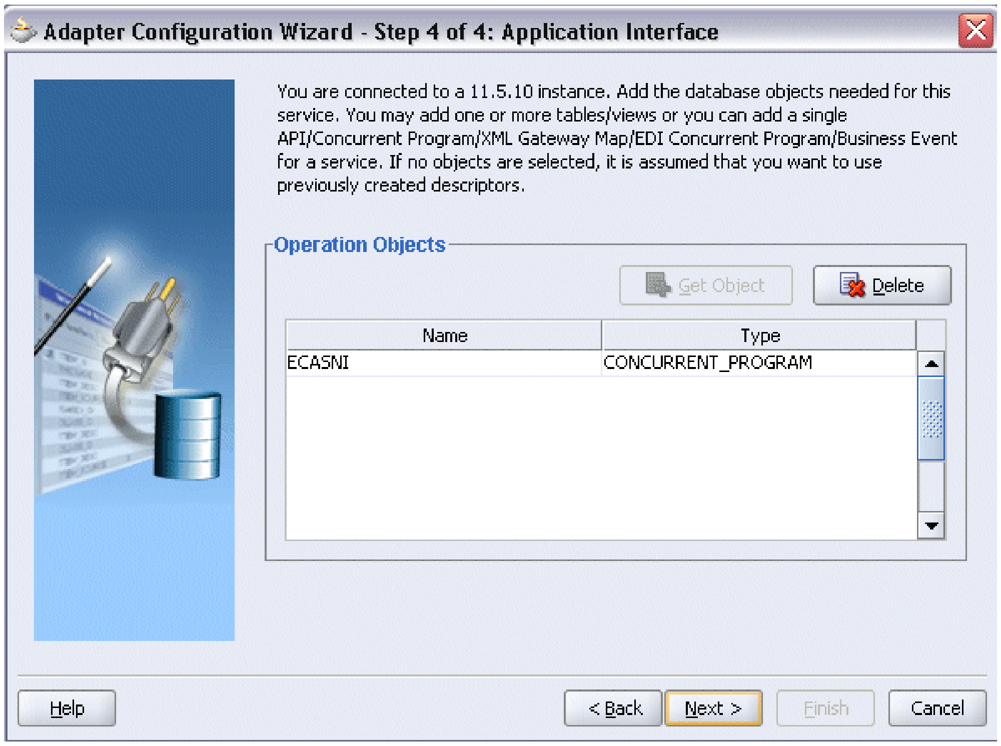
For example, navigate to *Product Families* > *Applications Technology* > *e-Commerce Gateway* > *Advanced Shipment Notification* > *EDI* to select `EDI > IN: Ship Notice/Manifest (856/DESADV) (ECASNI)` EDI concurrent program.

> **Note:** You can also search for an EDI program by entering the name of the program in the **Object Name** field. Select the **EDI** check box and click **Search**.

8. Click **OK**.

   The selected EDI concurrent program is added to Operation Objects.

Application Interface Dialog



Adapter Configuration Wizard - Step 4 of 4: Application Interface

You are connected to a 11.5.10 instance. Add the database objects needed for this service. You may add one or more tables/views or you can add a single API/Concurrent Program/XML Gateway Map/EDI Concurrent Program/Business Event for a service. If no objects are selected, it is assumed that you want to use previously created descriptors.

Operation Objects

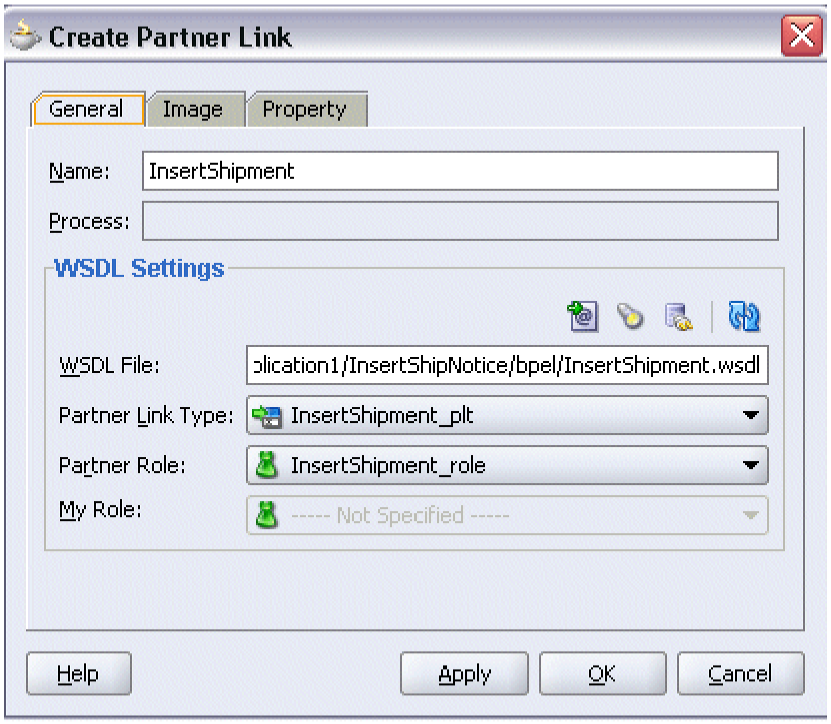
Name	Type
ECASNI	CONCURRENT_PROGRAM

Click **Next**.

9. Click **Finish** to complete the process of configuring Adapter for Oracle Applications. The wizard generates the WSDL file corresponding to the selected interface. This WSDL file is now available for the partner link.

   Note: When you click **Finish**, two SQL files may be added to the project if a wrapper does not exist for the function. A wrapper is generated the very first time you create the e-Commerce Gateway based service. Subsequent services reuse the same wrapper.

Completing the Partner Link Configuration



Name: InsertShipment

Process:

WSDL File: plication1/InsertShipNotice/bpel/InsertShipment.wsdl

Partner Link Type: InsertShipment_plt

Partner Role: InsertShipment_role

My Role: ----- Not Specified -----

10. Click **Apply** and **OK**. The partner link is created with the required WSDL settings.

Adding a Partner Link for File Adapter

Use this step to configure a BPEL process by adding a partner link for File Adapter to synchronously read an existing order to obtain the shipping details.

To add a Partner Link for File Adapter to read shipping details:

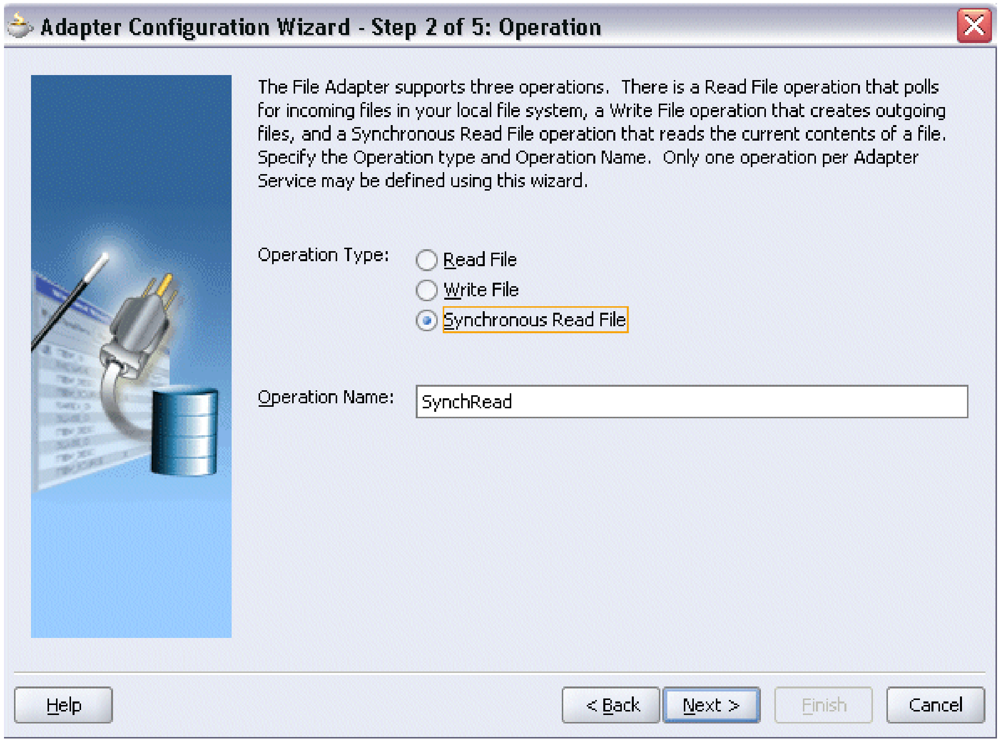
1. In JDeveloper BPEL Designer, drag and drop the **File Adapter** service from the **Service** section of the Component Palette into the Partner Link area of the process diagram. The Adapter Configuration Wizard welcome page appears.

2. Click **Next**. The Service Name dialog box appears.

3. Enter a name for the File Adapter service, such as `GetOrder`. You can add an optional description of the service.

4. Click **Next** and the Operation dialog box appears.

Specifying the Operation


Adapter Configuration Wizard - Step 2 of 5: Operation

The File Adapter supports three operations. There is a Read File operation that polls for incoming files in your local file system, a Write File operation that creates outgoing files, and a Synchronous Read File operation that reads the current contents of a file. Specify the Operation type and Operation Name. Only one operation per Adapter Service may be defined using this wizard.

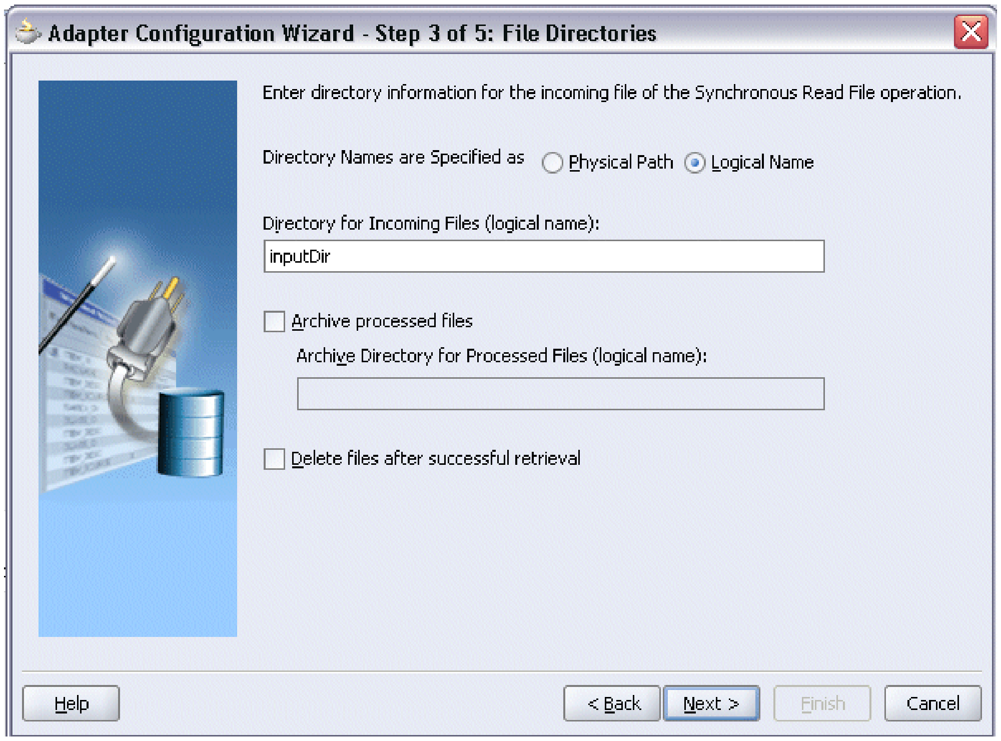
Operation Type: Read File / Write File / [✓ Synchronous Read File]

Operation Name: SynchRead

5. Specify the operation type, for example **Synchronous Read File**. This automatically populates the **Operation Name** field.

   Click **Next** to access the File Directories dialog box.

Configuring the Input File


Adapter Configuration Wizard - Step 3 of 5: File Directories

Enter directory information for the incoming file of the Synchronous Read File operation.

Directory Names are Specified as ( ) Physical Path [✓ Logical Name]

Directory for Incoming Files (logical name): inputDir

☐ Archive processed files

Archive Directory for Processed Files (logical name):

☐ Delete files after successful retrieval

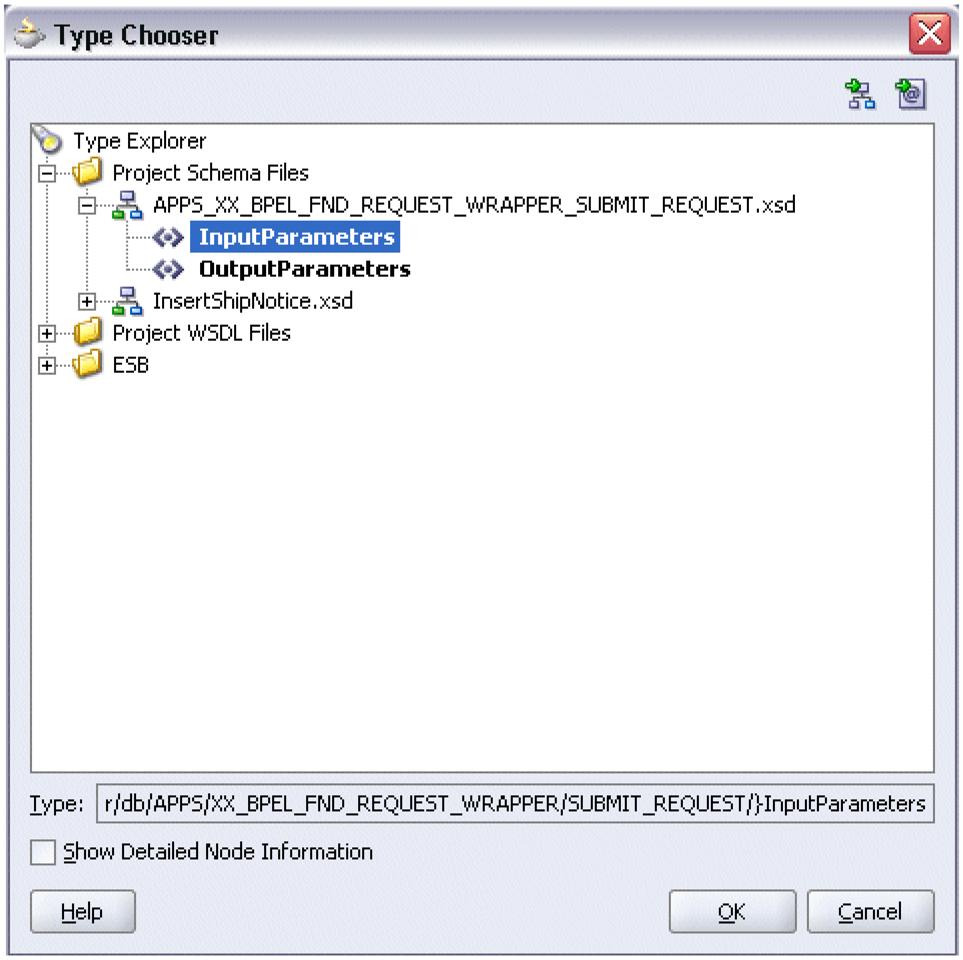
6. Select the **Logical Name** check box and specify the logical directory for the incoming file. Uncheck the **Delete files after successful retrieval** check box. Click **Next**.

7. Enter the name of the file for the synchronous read file operation, for example, enter 'order_data_xmlg.xml'. Click **Next**. The Messages dialog box appears.

8. Select **Browse** to open the Type Chooser to specify the Schema location and element.

   From the Type Chooser window, expand the node by selecting **Project Schema Files > APPS_XX_BPEL_FND_REQUEST_WRAPPER_SUBMIT_REQUEST.xsd > InputParameters**.

Selecting Schema from the Type Chooser

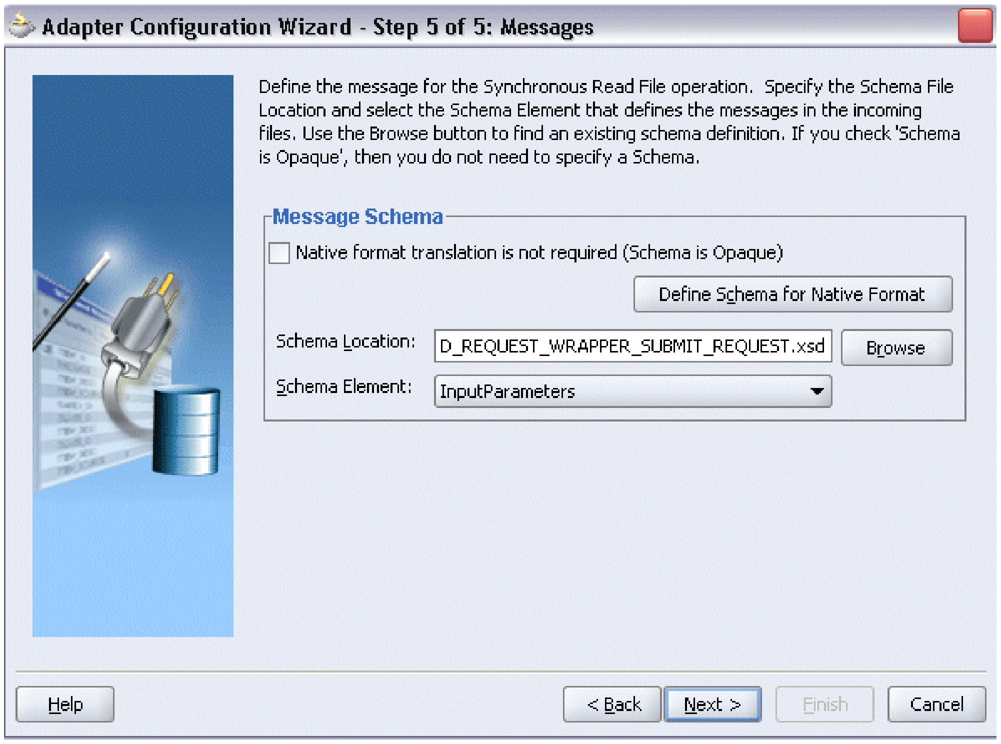


Type: r/db/APPS/XX_BPEL_FND_REQUEST_WRAPPER/SUBMIT_REQUEST/}InputParameters

☐ Show Detailed Node Information

9. Click **OK** to populate the selected values in the Messages dialog box.

Populating Selected Message Schema and Element


Adapter Configuration Wizard - Step 5 of 5: Messages

Define the message for the Synchronous Read File operation. Specify the Schema File Location and select the Schema Element that defines the messages in the incoming files. Use the Browse button to find an existing schema definition. If you check 'Schema is Opaque', then you do not need to specify a Schema.

Message Schema

Native format translation is not required (Schema is Opaque)

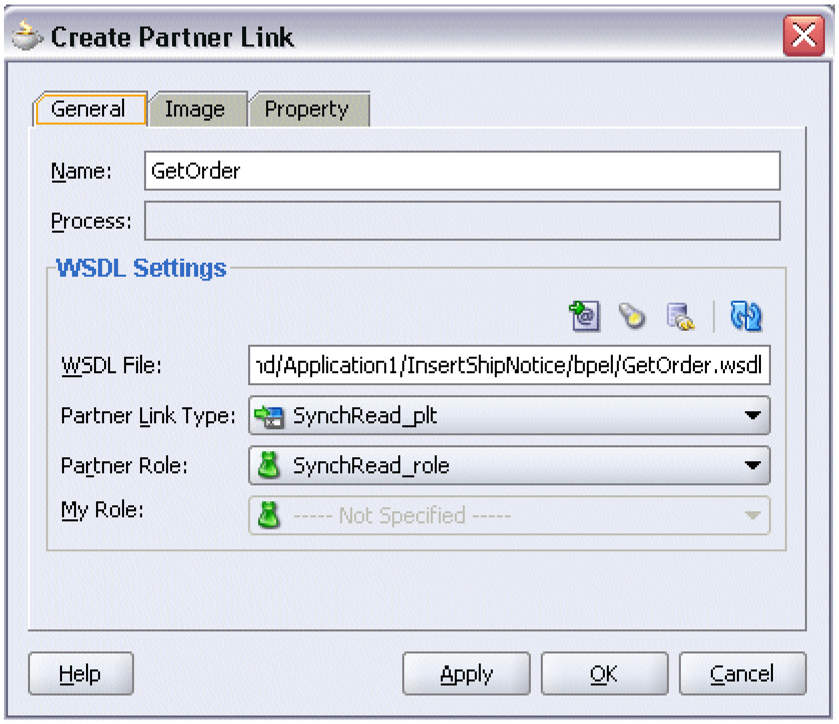
Schema Location: D_REQUEST_WRAPPER_SUBMIT_REQUEST.xsd

Schema Element: InputParameters

10. Click **Next** and then **Finish**. The wizard generates the WSDL file corresponding to the partner link. The main Create Partner Link dialog box appears, specifying the new WSDL file `GetOrder.wsdl`.

    Click **Apply** and then **OK** to complete the configuration and create the partner link with the required WSDL settings for the File Adapter service.

Create Partner Link



Create Partner Link

General | Image | Property

Name: GetOrder

Process:

WSDL Settings

WSDL File: nd/Application1/InsertShipNotice/bpel/GetOrder.wsdl

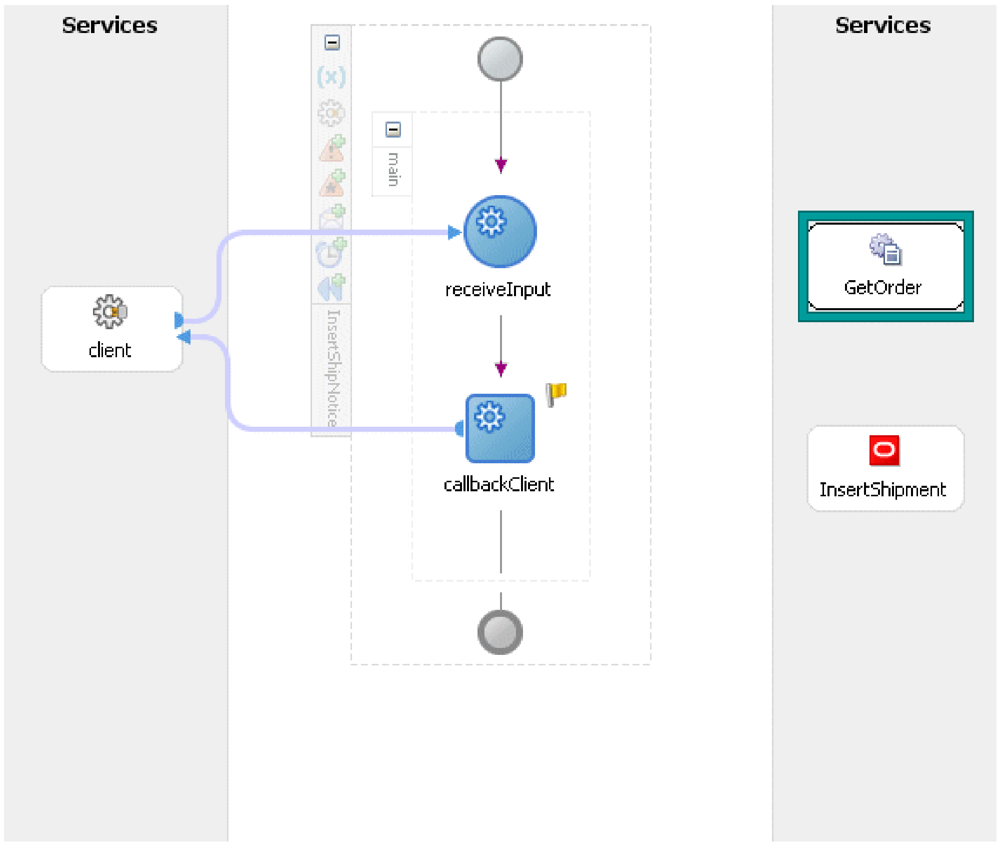
Partner Link Type: SynchRead_plt

Partner Role: SynchRead_role

My Role: ----- Not Specified -----

The `GetOrder` Partner Link appears in the BPEL process diagram.

Adding the Partner Link for File Adapter



Configuring the Invoke Activity

After adding and configuring the partner link, you can configure the following two **Invoke** activities to invoke the EDI concurrent program and File Adapter partner link:

1. To get the shipping details that is received from the Receive activity by invoking the `GetOrder` partner link in an XML file.

2. To create a shipment notice by invoking `InsertShipment` partner link.

To add the first Invoke activity for a partner link to get shipping details:

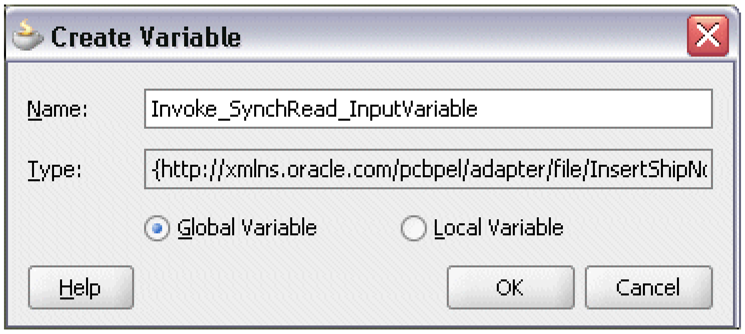
1. In JDeveloper BPEL Designer, select **Process Activities** in the Component Palette. Drag and drop the first **Invoke** activity into the process diagram, between the **receiveInput** and **callbackClient** activities.

2. Link the Invoke activity to the `GetOrder` service. The Edit Invoke dialog appears.

3. Enter a name for the Invoke activity. Click the **Create** icon next to the **Input**

Variable field to create a new variable. The Create Variable dialog box appears.

Create Input Variable


Create Variable

Name: Invoke_SynchRead_InputVariable

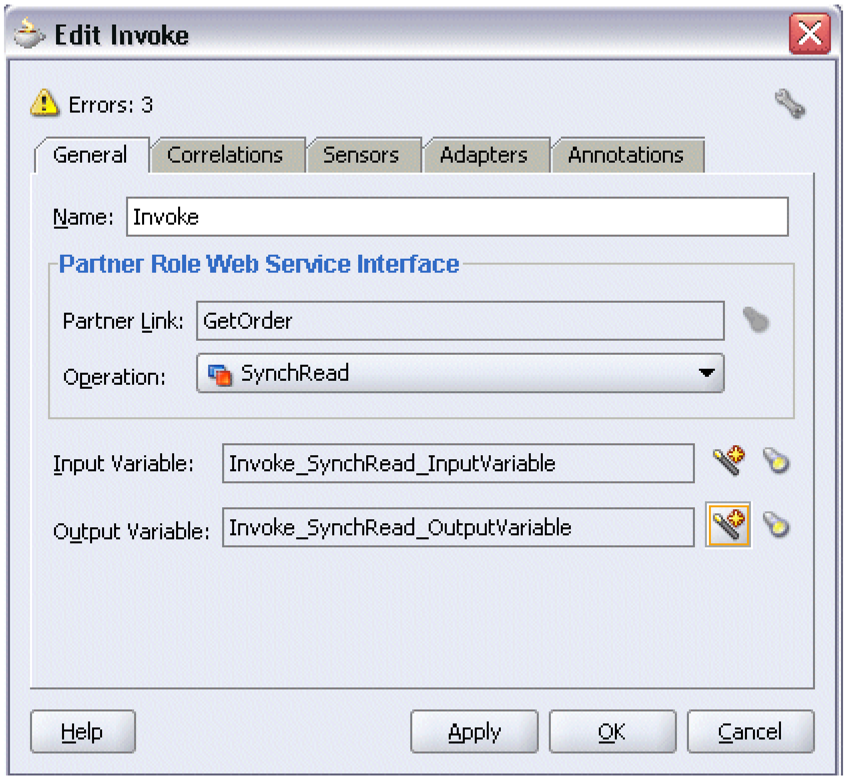
Type: {http://xmlns.oracle.com/pcbpel/adapter/file/InsertShipNo

[✓ Global Variable] Local Variable

4. Select **Global Variable** and then enter a name for the variable. You can also accept the default name. Click **OK**.

5. Click the **Create** icon next to the **Output Variable** field to create a new variable. The Create Variable dialog box appears.

6. Select **Global Variable** and then enter a name for the variable. You can also accept the default name. Click **OK**.


Edit Invoke

Errors: 3

Name: Invoke

Partner Role Web Service Interface

Partner Link: GetOrder

Operation: SynchRead

Input Variable: Invoke_SynchRead_InputVariable

Output Variable: Invoke_SynchRead_OutputVariable

7. Click **Apply** and **OK** in the Edit Invoke dialog box to finish configuring the Invoke activity.

   The Invoke activity appears in the BPEL process diagram.

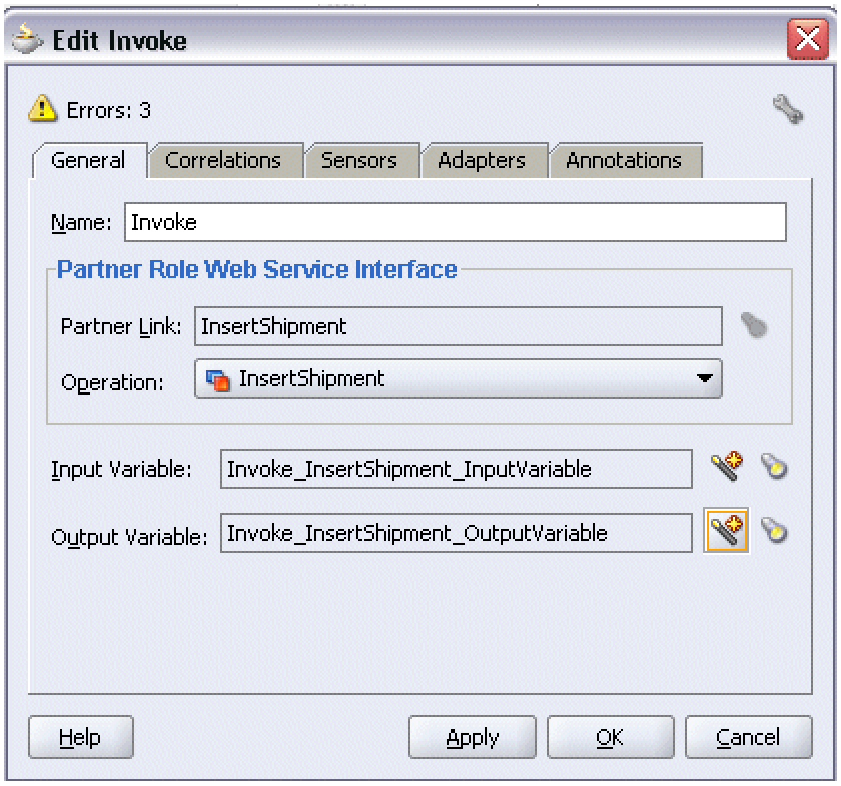
To add the second Invoke activity for a partner link to create a shipment notice:

1. In JDeveloper BPEL Designer, select **Process Activities** in the Component Palette. Drag and drop the second **Invoke** activity into the process diagram, between the first **Invoke** and **callbackClient** activities.

2. Link the Invoke activity to the `InsertShipment` service. The Edit Invoke dialog box appears.

   The **Operation** is automatically selected, depending on the EDI concurrent program that you chose when configuring the partner link.

3. Repeat Step 3 to Step 6 described in the first Invoke activity.



Name: Invoke

Partner Link: InsertShipment

Operation: InsertShipment

Input Variable: Invoke_InsertShipment_InputVariable

Output Variable: Invoke_InsertShipment_OutputVariable

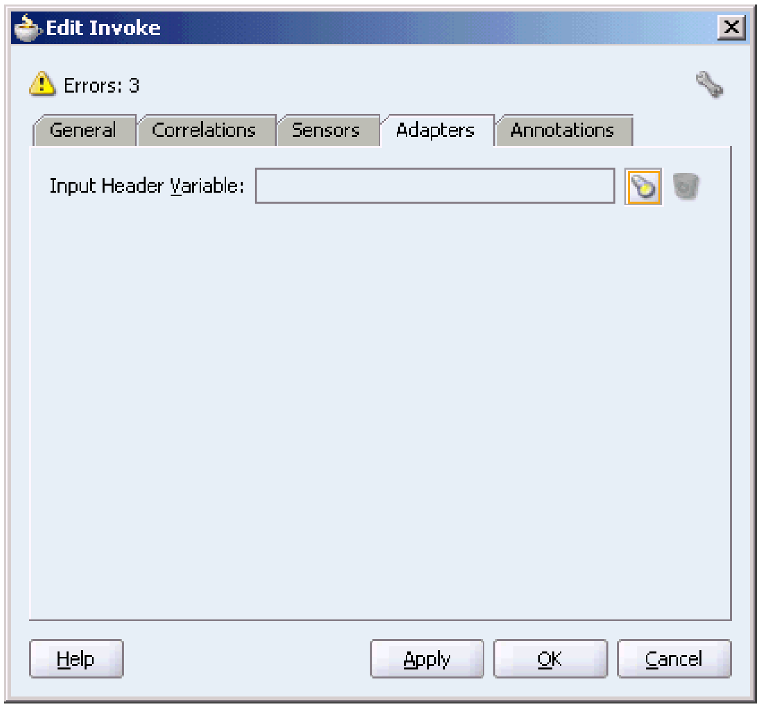
4. Click **Apply** and then **OK** to complete the Invoke activity.

   > **Note:** You can define an **Input Header Variable** on the **Adapters** tab of the Invoke dialog box. This variable can be used to provide context information for Oracle Applications.

To Create the Header Variable:

1. Click on the Adapters tab in the Edit Invoke dialog box and click **Browse Variable...** icon for the Input Header Variable field.

Specifying Input Header Variable



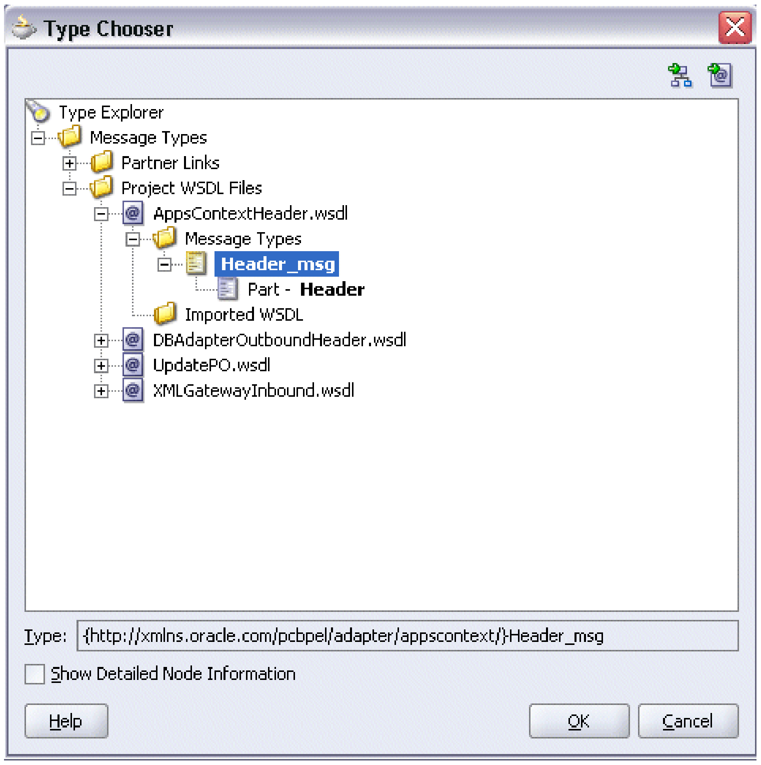
2. In the Variable Chooser dialog box, right-click on the Variables node and select Create Variable option from the menu.

3. Enter `header` in the Name field.

4. Select **Message Type** and click the **Browse Message Type...** icon to open Type Chooser dialog box.

5. Expand the Partner Link node to locate the Header_msg node `{http://xmlns.oracle.com/pcbpel/adapter/appscontext/}Header_msg` for your partner link. The Header_msg node should be under the path, *Your Partner Link WSDL* > AppsContextHeader.wsdl > Message Types > Header_msg.

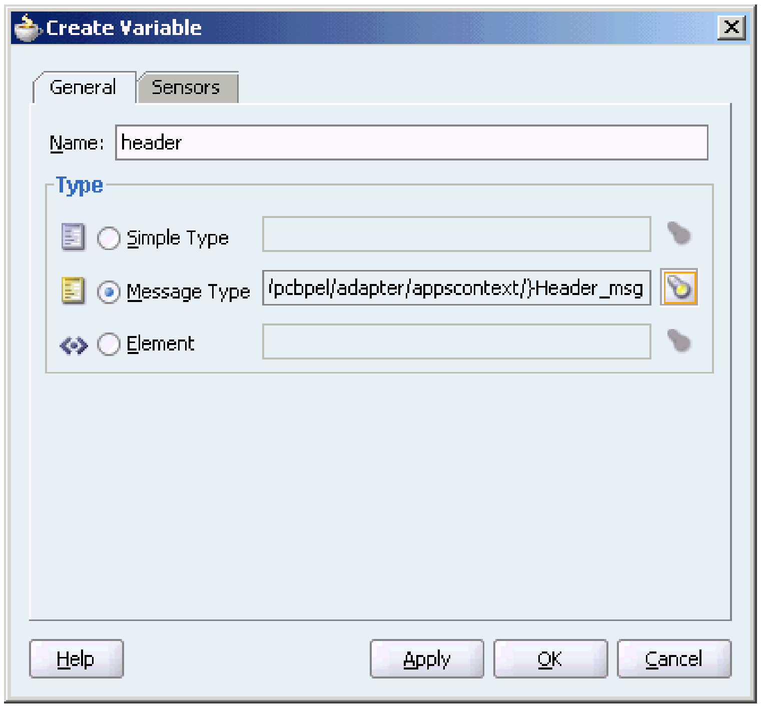
Declaring Header Variable



Type: {http://xmlns.oracle.com/pcbpel/adapter/appscontext/}Header_msg

6. Click **OK** to return to the Create Variable dialog box with your selected message type populated.

Populating Selected Header Variable



Name: header

Type

( ) Simple Type

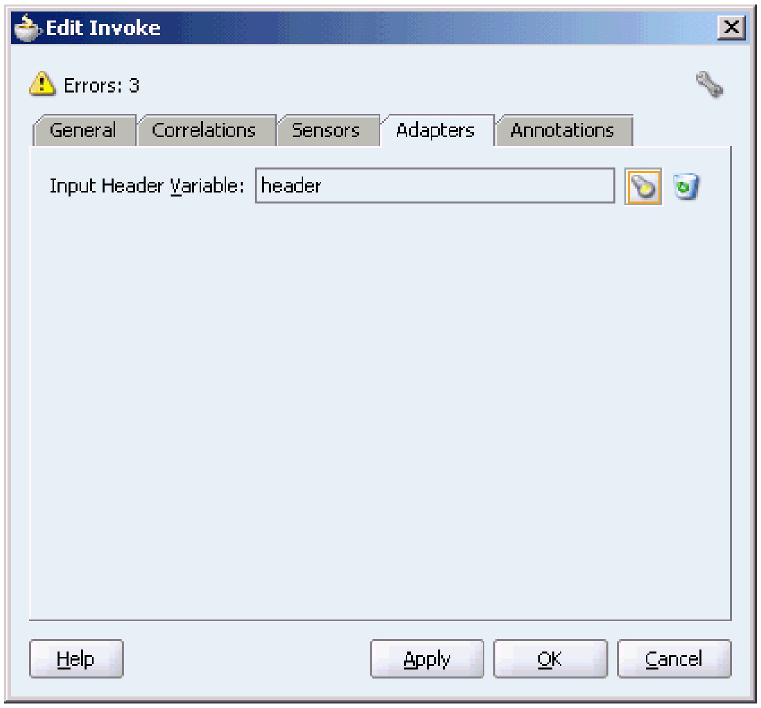
[✓ Message Type] /pcbpel/adapter/appscontext/}Header_msg

( ) Element

7. To view your header variable with Username, Responsibility, ORG_ID, Responsibility Application, Security Group, and NLS Language, click the **Browse Message Type...** icon to open Variable Chooser dialog box. Locate the header variable to view the variable hierarchical structure with these parameters needed for applications context.

8. Click **OK** to return to the Edit Invoke dialog with the selected header variable populated for the Input Header Variable field. Click **Apply** to complete the header creation.

Populating Input Header Variable



Input Header Variable: header

How to assign header variables, see Configure the Assign activity, page 9-25.

Configuring the Assign Activity

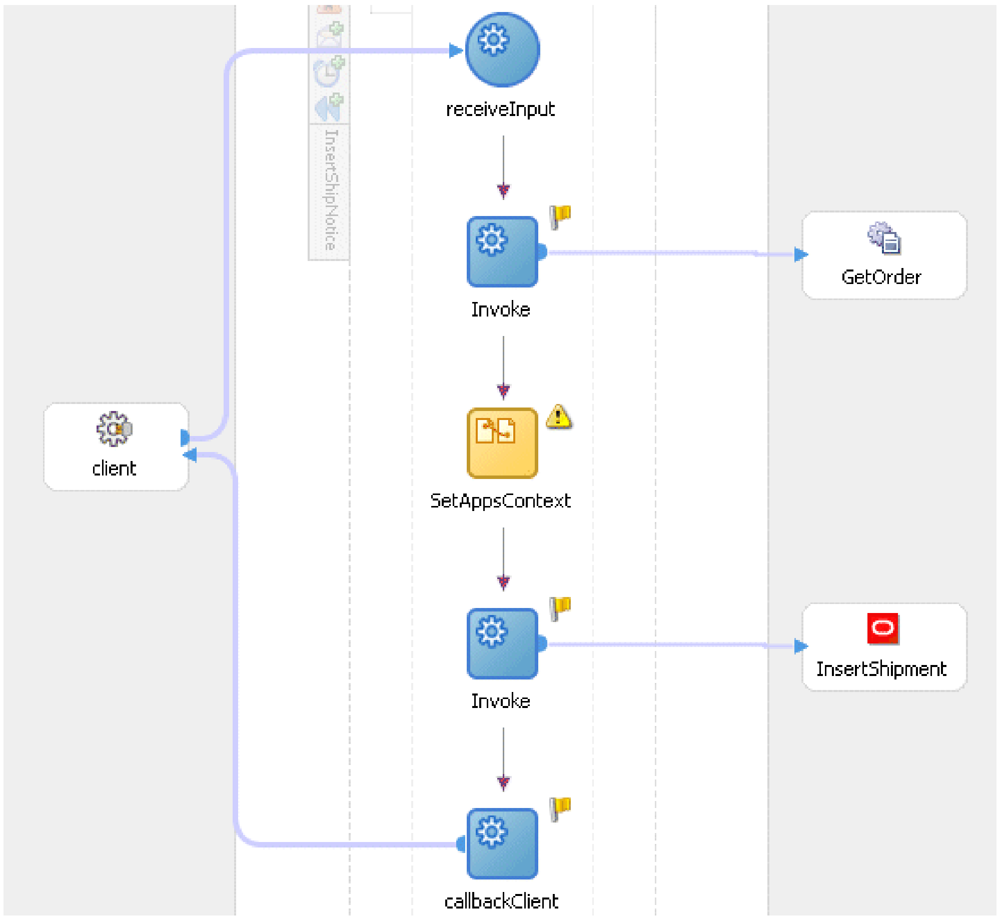
This step is to configure two Assign activities in order to:

- Set the applications context information.
- Pass the output of `GetOrder` service as an input to the `InsertShipment` service.

To add the first Assign activity to set applications context information:

1. In JDeveloper BPEL Designer, drag and drop the **Assign** activity from the Component Palette into the process diagram between the two **Invoke** activities.

Adding an Assign Activity



2. Double-click the **Assign** activity to access the Edit Assign dialog box.

3. Click the General tab to enter the name for the Assign activity, such as 'SetAppsContext'.

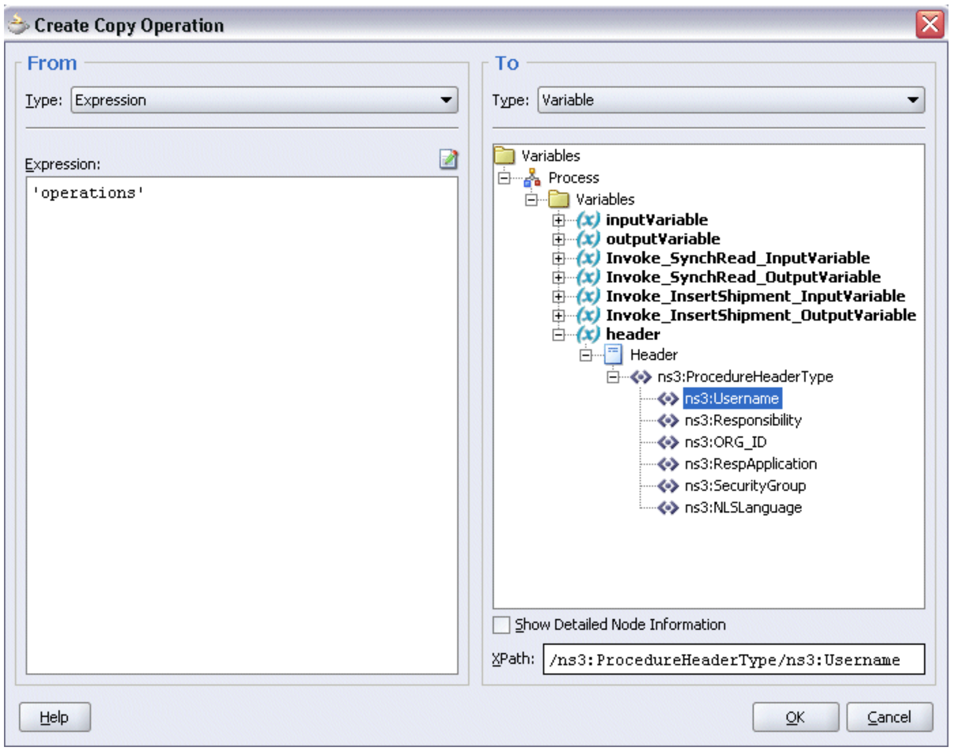
4. On the Copy Operation tab, click **Create** and select **Copy Operation** from the menu. The Create Copy Operation window appears.

5. Enter the first pair of parameters:

   - In the From navigation tree, select type Expression and enter `'operations'` in the Expression box.

   - In the To navigation tree, select type Variable. Navigate to **Variable** > **Process** > **Variables** > **header** > **Header**> **ns5:ProcedureHeaderType** and select **ns5:Username**. The XPath field should contain your selected entry.

Assign Username Parameter


Create Copy Operation

From

Type: Expression

Expression:

```
'operations'
```

To

Type: Variable

XPath: /ns3:ProcedureHeaderType/ns3:Username

- Click **OK**.

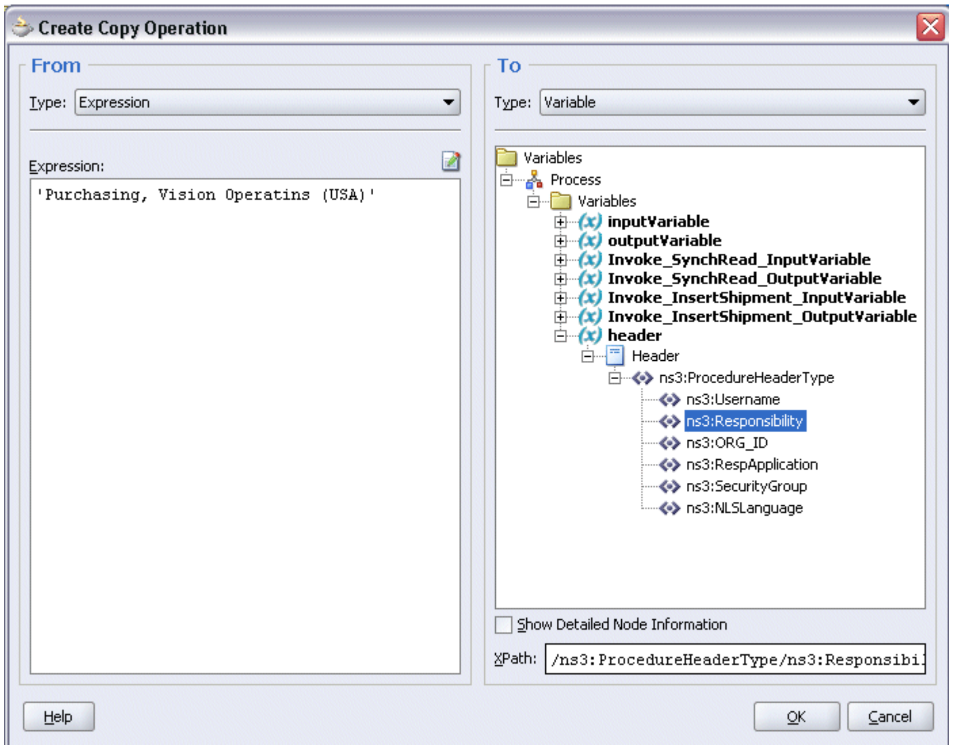
6. Enter the second pair of parameters:

   On the Copy Operation tab, click **Create** and select **Copy Operation** from the menu. The Create Copy Operation window appears.

   - In the From navigation tree, select type Expression and enter `'Purchasing, Vision Operatins (USA)'` in the Expression box.

   - In the To navigation tree, select type Variable. Navigate to **Variable** > **Process** > **Variables** > **header** > **Header**> **ns5:ProcedureHeaderType** and select **ns5:Responsibility**. The XPath field should contain your selected entry.

Assign Responsibility Parameter



```
'Purchasing, Vision Operatins (USA)'
```

```
/ns3:ProcedureHeaderType/ns3:Responsibi
```

- Click **OK**.

7. Enter the third pair of parameters:

   On the Copy Operation tab, click **Create** and select **Copy Operation** from the menu. The Create Copy Operation window appears.

   - In the From navigation tree, select type Expression and enter `'207'` in the Expression box.

   - In the To navigation tree, select type Variable. Navigate to **Variable > Process > Variables > header > Header> ns5:ProcedureHeaderType** and select **ns5:ORG_ID**. The XPath field should contain your selected entry.

   - Click **OK**.

8. Enter the fourth pair of parameters:

   On the Copy Operation tab, click **Create** and select **Copy Operation** from the menu. The Create Copy Operation window appears.

   - In the From navigation tree, select type Expression and enter `'Purchasing'` in the Expression box.

- In the To navigation tree, select type Variable. Navigate to **Variable > Process > Variables > header > Header> ns5:ProcedureHeaderType** and select **ns5:RespApplication**. The XPath field should contain your selected entry.

- Click **OK**.

9. Enter the fifth pair of parameters:

   On the Copy Operation tab, click **Create** and select **Copy Operation** from the menu. The Create Copy Operation window appears.

   - In the From navigation tree, select type Expression and enter `'Standard'` in the Expression box.

   - In the To navigation tree, select type Variable. Navigate to **Variable > Process > Variables > header > Header> ns5:ProcedureHeaderType** and select **ns5:SecurityGroup**. The XPath field should contain your selected entry.

   - Click **OK**.

10. Enter the sixth pair of parameters:

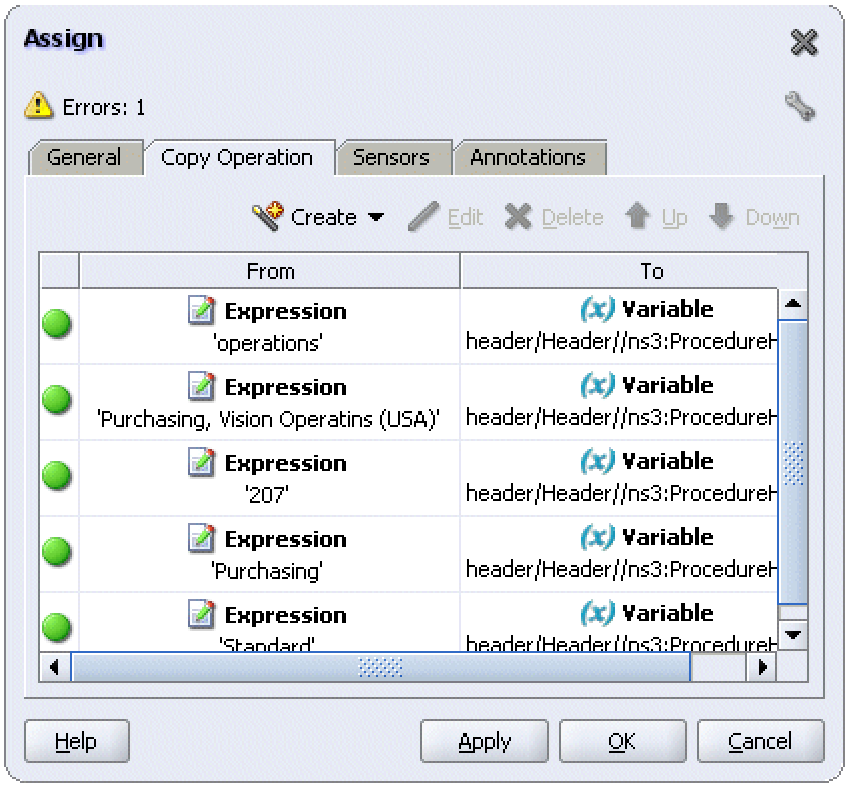
    On the Copy Operation tab, click **Create** and select **Copy Operation** from the menu. The Create Copy Operation window appears.

    - In the From navigation tree, select type Expression and enter `'US'` in the Expression box.

    - In the To navigation tree, select type Variable. Navigate to **Variable > Process > Variables > header > Header> ns5:ProcedureHeaderType** and select **ns5:NLSLanguage**. The XPath field should contain your selected entry.

    - Click **OK**.

Assign Activity



	From	To
●	Expression 'operations'	Variable header/Header//ns3:ProcedureH
●	Expression 'Purchasing, Vision Operatins (USA)'	Variable header/Header//ns3:ProcedureH
●	Expression '207'	Variable header/Header//ns3:ProcedureH
●	Expression 'Purchasing'	Variable header/Header//ns3:ProcedureH
●	Expression 'Standard'	Variable header/Header//ns3:ProcedureH

11. Click **Apply** and then **OK** to complete the configuration of the Assign activity.

To add the second Assign activity to set payload for the InsertShipment service:

1. In JDeveloper BPEL Designer, drag and drop the **Assign** activity from the Component Palette into the process diagram after the first **Assign** activity.

2. Double-click the **Assign** activity to access the Edit Assign dialog box.

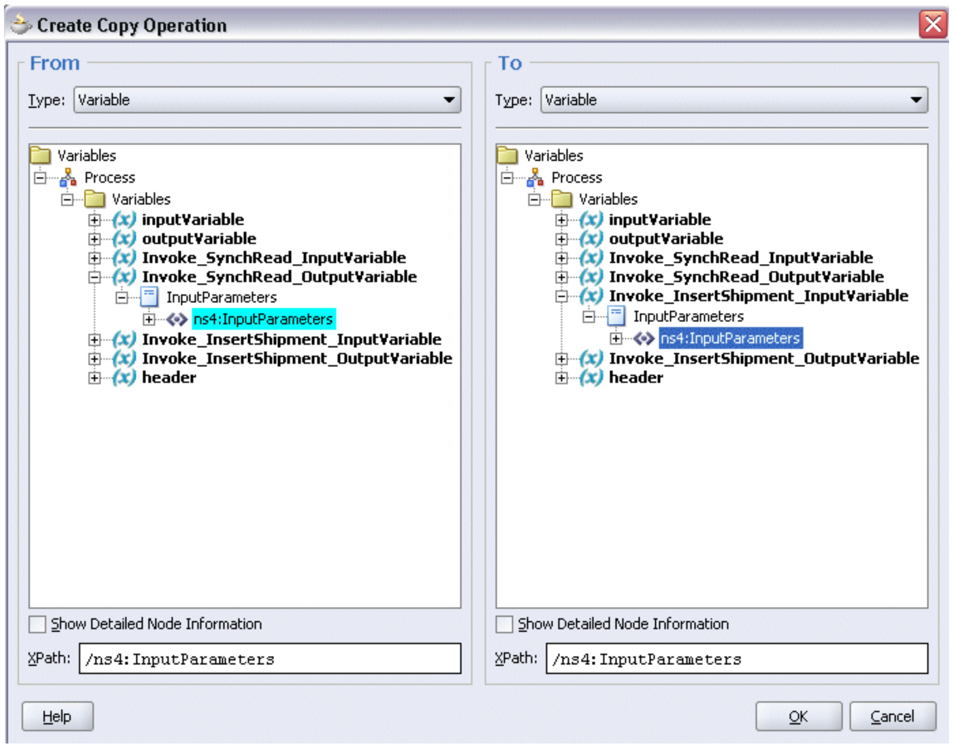
3. Click the General tab to enter the name for the second Assign activity, such as 'SetPayload'.

4. On the Copy Operation tab, click **Create** and select **Copy Operation** from the menu. The Create Copy Operation window appears.

5. Enter the parameter information:

   - In the From navigation tree, select type Variable. Navigate to **Variable > Process > Variables > Invoke_SynchRead_OutputVariable > InputParameters** and select **ns4:InputParameters**. The XPath field should contain your selected entry.

- In the To navigation tree, select type Variable. Navigate to **Variable > Process > Variables > Invoke_InsertShipment_InputVariable > InputParameters** and select **ns4:InputParameters**. The XPath field should contain your selected entry.



Create Copy Operation

From

Type: Variable

```
Variables
  Process
    Variables
      inputVariable
      outputVariable
      Invoke_SynchRead_InputVariable
      Invoke_SynchRead_OutputVariable
        InputParameters
          ns4:InputParameters
      Invoke_InsertShipment_InputVariable
      Invoke_InsertShipment_OutputVariable
      header
```

Show Detailed Node Information

XPath: /ns4:InputParameters

To

Type: Variable

```
Variables
  Process
    Variables
      inputVariable
      outputVariable
      Invoke_SynchRead_InputVariable
      Invoke_SynchRead_OutputVariable
      Invoke_InsertShipment_InputVariable
        InputParameters
          ns4:InputParameters
      Invoke_InsertShipment_OutputVariable
      header
```

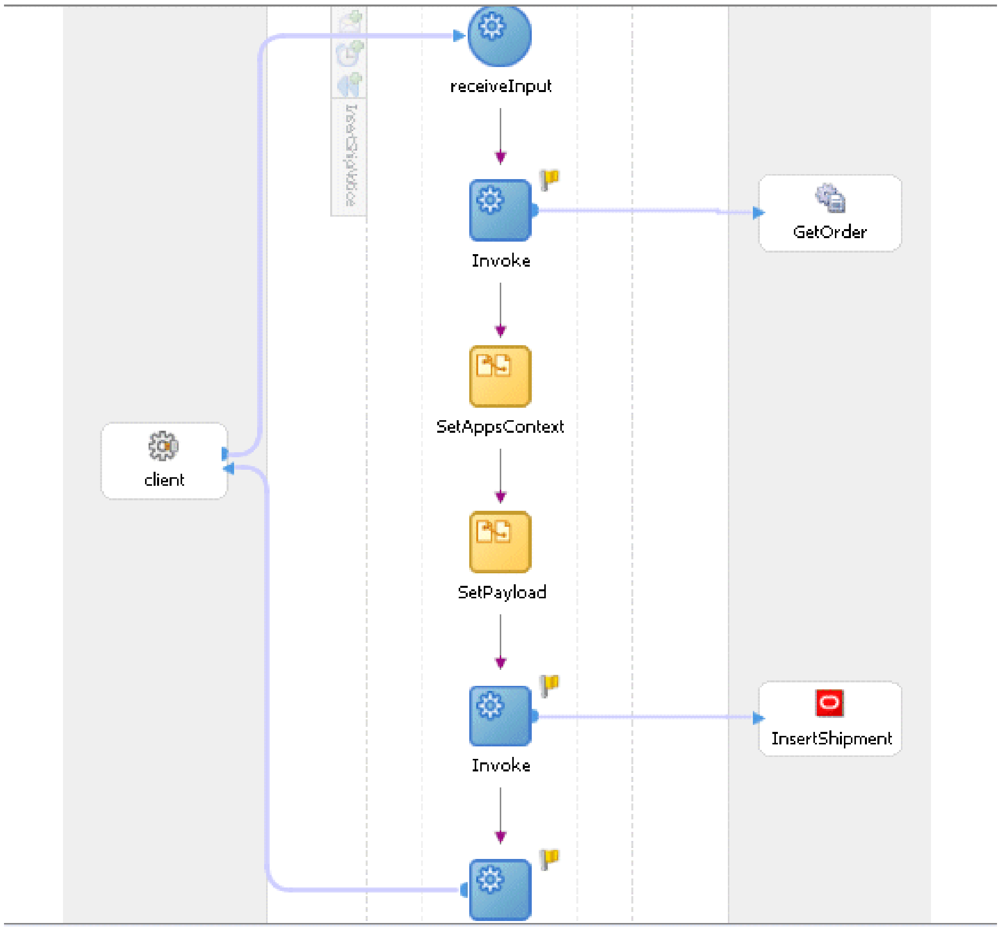
Show Detailed Node Information

XPath: /ns4:InputParameters

- Click **OK**. The Edit Assign dialog box appears.

6. Click **Apply** and then **OK** to complete the configuration of the second Assign activity.

BPEL Process Diagram



Run-Time Tasks for e-Commerce Gateway

After designing the BPEL process, the next step is to deploy, run and monitor it.

1. Deploy the BPEL process, page 9-32
2. Test the BPEL process, page 9-33
3. Verify records in Oracle Applications., page 9-36

Deploying the BPEL Process

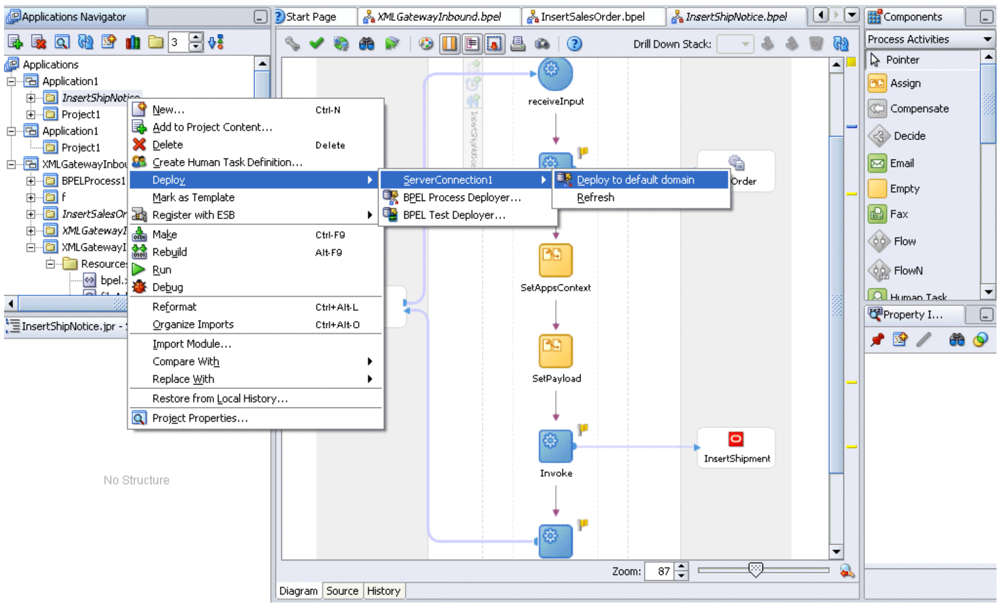
You need to deploy the BPEL process before you can run it. The BPEL process is first compiled and then deployed to the BPEL server.

To deploy the BPEL process:

1. Select the BPEL project in the Applications window.

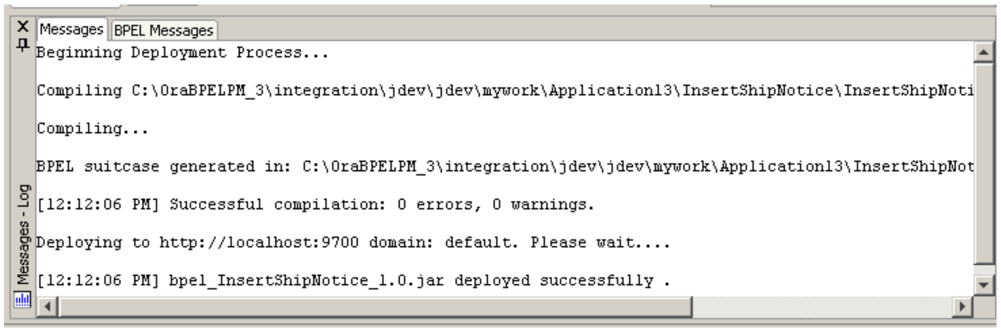
2. Right-click the project name, and then select **Deploy > [Server Connection] > Deploy to Default Domain** from the menu that appears.

Deploying the BPEL Process



3. The BPEL process is compiled and deployed. You can check the progress in the Messages window.

Messages Window



```
Beginning Deployment Process...

Compiling C:\OraBPELPM_3\integration\jdev\jdev\mywork\Application13\InsertShipNotice\InsertShipNoti

Compiling...

BPEL suitcase generated in: C:\OraBPELPM_3\integration\jdev\jdev\mywork\Application13\InsertShipNot

[12:12:06 PM] Successful compilation: 0 errors, 0 warnings.

Deploying to http://localhost:9700 domain: default. Please wait....

[12:12:06 PM] bpel_InsertShipNotice_1.0.jar deployed successfully .
```

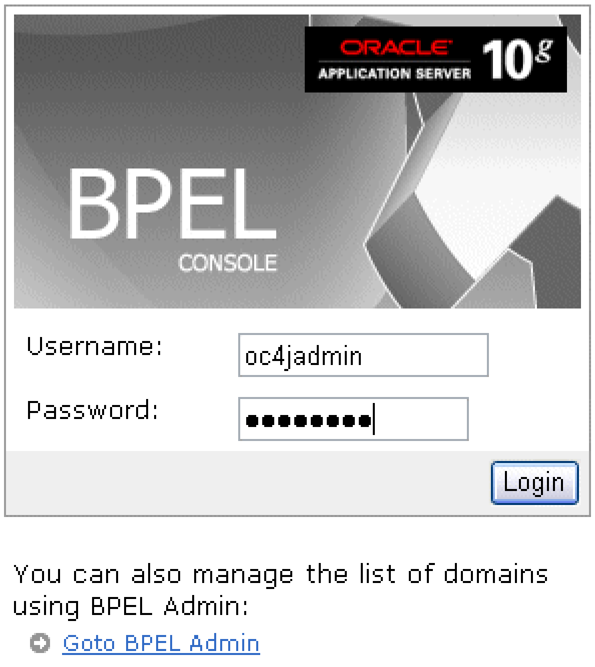
Testing the BPEL Process

Once the BPEL process is deployed, it can be seen in the BPEL Console. You can manage and monitor the process from the BPEL Console. You can also test the process and the integration interface by manually initiating the process.

To test the BPEL process:

1. Navigate to Oracle Application Server 10*g* BPEL Console (`http://<soaSuiteServerHostName>:<port>/BPELConsole`).

   The BPEL Console login page appears.



Username: oc4jadmin

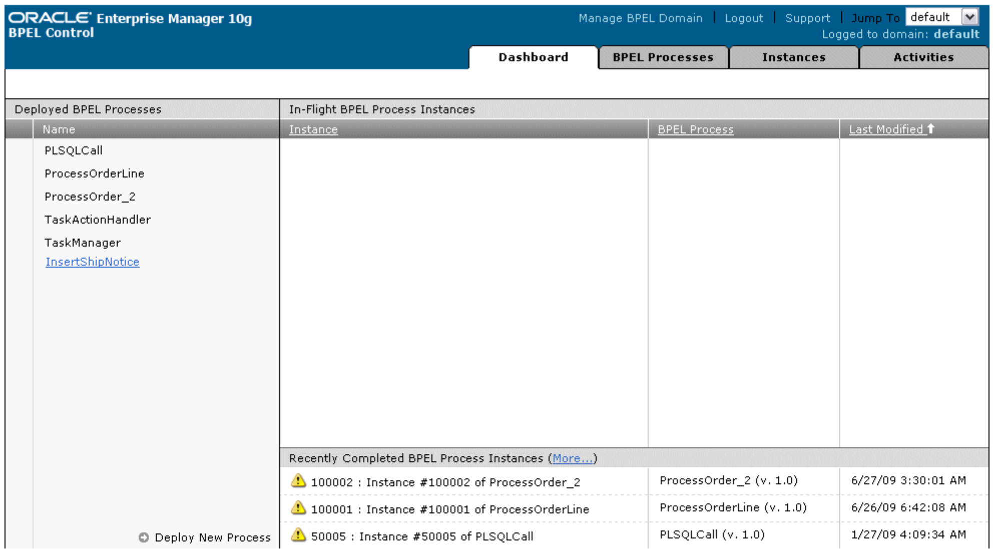
Password: ●●●●●●●●●

You can also manage the list of domains using BPEL Admin:

2. Enter the username and password and click **Login**.

3. The Oracle Enterprise Manager 10*g* BPEL Control appears. The list of deployed processes is shown under Deployed BPEL Processes.

Deployed BPEL Processes


ORACLE Enterprise Manager 10g
BPEL Control
Manage BPEL Domain | Logout | Support | Jump To default
Logged to domain: default
Dashboard
BPEL Processes
Instances
Activities
Deploy New Process

Deployed BPEL Processes

Name
PLSQLCall
ProcessOrderLine
ProcessOrder_2
TaskActionHandler
TaskManager
InsertShipNotice

In-Flight BPEL Process Instances

Instance	BPEL Process	Last Modified

Recently Completed BPEL Process Instances (More...)

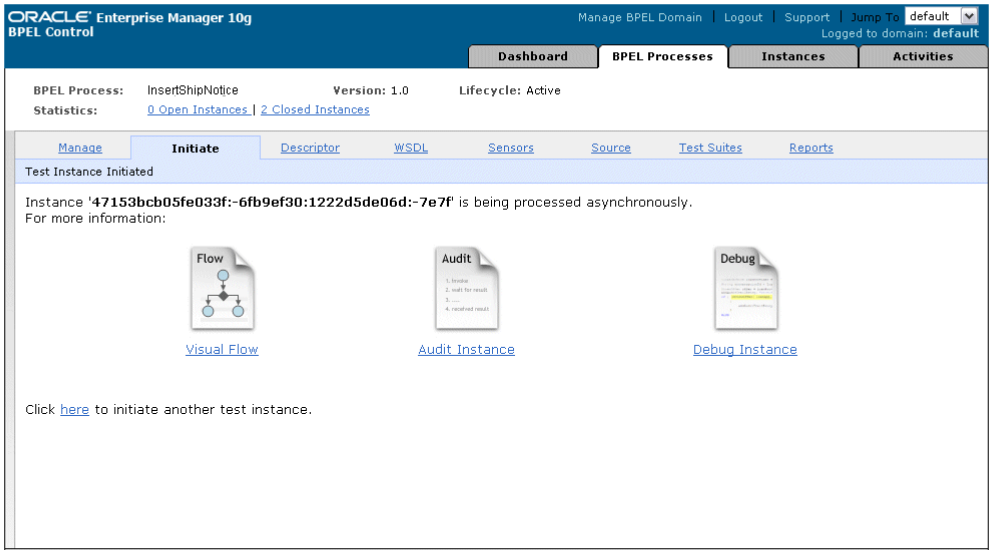
100002 : Instance #100002 of ProcessOrder_2	ProcessOrder_2 (v. 1.0)	6/27/09 3:30:01 AM
100001 : Instance #100001 of ProcessOrderLine	ProcessOrderLine (v. 1.0)	6/26/09 6:42:08 AM
50005 : Instance #50005 of PLSQLCall	PLSQLCall (v. 1.0)	1/27/09 4:09:34 AM

4. Click the BPEL process that you want to initiate. The Initiate page appears. Enter the input string required by the process.

5. Click **Post XML Message** to initiate the process.

6. The BPEL process is now initiated. You can check the process flow by clicking the **Visual Flow** icon.

BPEL Console Initiate Page



BPEL Process: InsertShipNotice Version: 1.0 Lifecycle: Active

Statistics: 0 Open Instances | 2 Closed Instances

Test Instance Initiated

Instance '47153bcb05fe033f:-6fb9ef30:1222d5de06d:-7e7f' is being processed asynchronously.
For more information:

Click here to initiate another test instance.

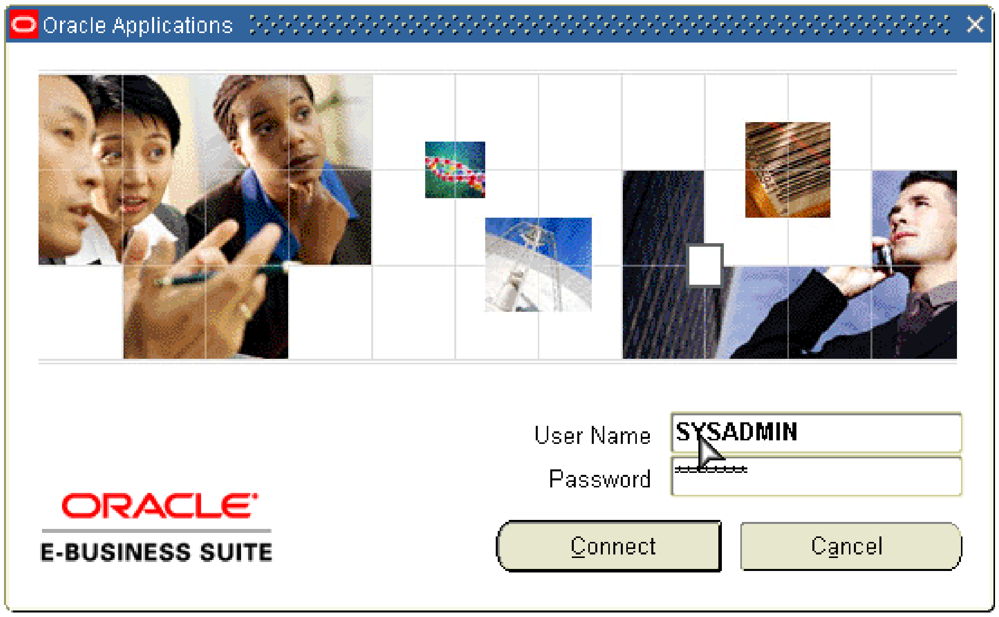
7. The audit trail provides information on the steps that have been executed. The audit trail also records the Request ID that is returned for the transaction. You can check the audit trail by clicking the **Audit Instance** icon.

Verifying Records in Oracle Applications

To verify records in Oracle Applications:

1. Log in to Oracle Applications as the System Administrator.

Oracle Applications Login Dialog Box



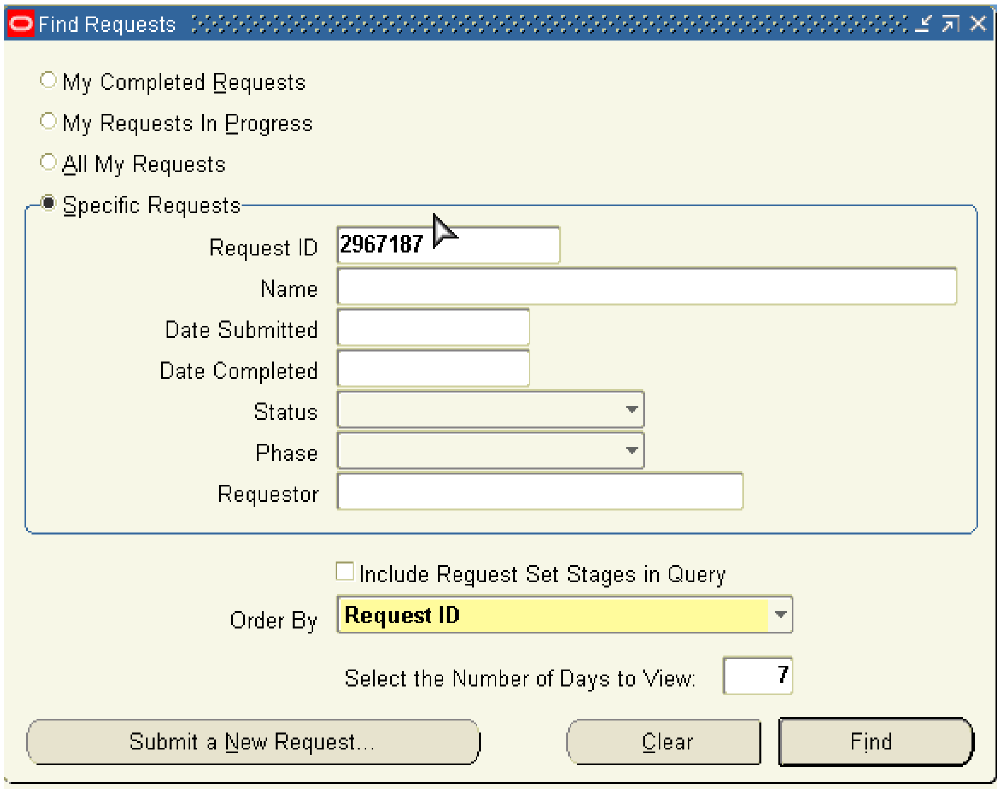
User Name: SYSADMIN

Password: ********

2. Select **Requests** from the **View** menu.

3. Search for the Request by entering the Request Id that you got from the audit trail, then click **Find**.

Find Requests Dialog Box



Request ID: 2967187

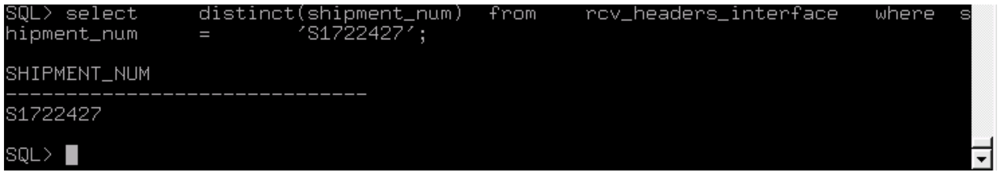
Order By: Request ID

Select the Number of Days to View: 7

4. The Request details are displayed. You can check for details such as the Phase and Status of the request.

5. If the **Status** of the request is `Complete`, you can also query the appropriate table in Oracle Applications to search for the relevant records that have been inserted.

Querying Oracle Applications for a Record



```
SQL> select     distinct(shipment_num)  from    rcv_headers_interface   where  s
hipment_num     =       'S1722427';

SHIPMENT_NUM
------------------------------
S1722427

SQL> 
```

A

WSDL Definition File and Connection Information Details

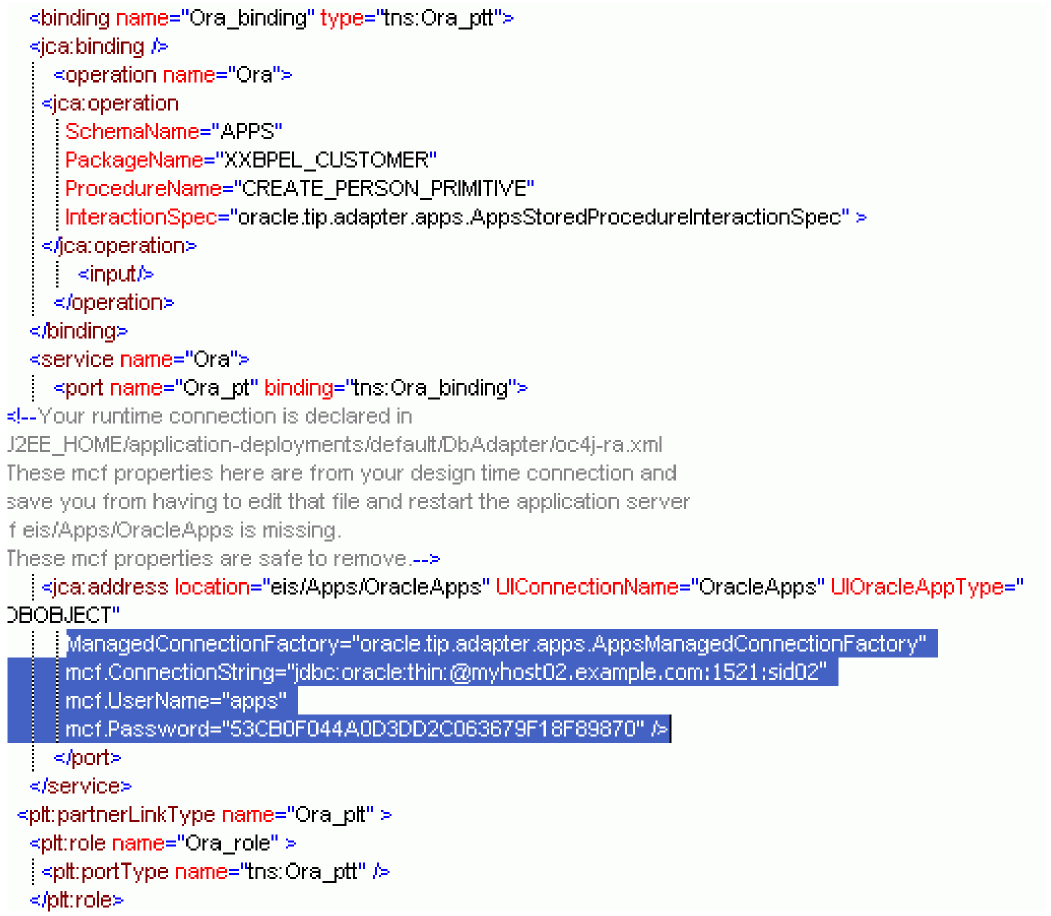
This appendix covers the following topics:

- WSDL Definition File
- Configuring Connection Information

WSDL Definition File

Web Service Definition Language (WSDL) is generated by the JDeveloper BPEL Designer during design time. The WSDL file generated by the Adapter Wizard is the adapter service definition. In addition, it specifies various operations exposed by the service. The operations are based on user input to the Adapter Wizard. An operation either retrieves or inserts data to Oracle Applications and is represented by a JCA activation or interaction spec.

Example of a WSDL File



```
  <binding name="Ora_binding" type="tns:Ora_ptt">
  <jca:binding />
    <operation name="Ora">
   <jca:operation
     SchemaName="APPS"
     PackageName="XXBPEL_CUSTOMER"
     ProcedureName="CREATE_PERSON_PRIMITIVE"
     InteractionSpec="oracle.tip.adapter.apps.AppsStoredProcedureInteractionSpec" >
   </jca:operation>
      <input/>
    </operation>
  </binding>
  <service name="Ora">
    <port name="Ora_pt" binding="tns:Ora_binding">
<!--Your runtime connection is declared in
J2EE_HOME/application-deployments/default/DbAdapter/oc4j-ra.xml
These mcf properties here are from your design time connection and
save you from having to edit that file and restart the application server
f eis/Apps/OracleApps is missing.
These mcf properties are safe to remove.-->
   <jca:address location="eis/Apps/OracleApps" UIConnectionName="OracleApps" UIOracleAppType="
DBOBJECT"
     ManagedConnectionFactory="oracle.tip.adapter.apps.AppsManagedConnectionFactory"
     mcf.ConnectionString="jdbc:oracle:thin:@myhost02.example.com:1521:sid02"
     mcf.UserName="apps"
     mcf.Password="53CB0F044A0D3DD2C063679F18F89870" />
    </port>
  </service>
 <plt:partnerLinkType name="Ora_plt" >
  <plt:role name="Ora_role" >
   <plt:portType name="tns:Ora_ptt" />
  </plt:role>
```

Configuring Connection Information

OracleAS Adapter for Oracle Applications is deployed as J2CA 1.5 resource adapters within the same OC4J container as BPEL Process Manager during installation.

Although OracleAS Adapter for Oracle Applications is physically deployed as J2CA 1.5 resource adapters, the logical deployment of the Adapter involves creating the connection entries for the J2CA 1.0 resource adapter. This connection information is for tying up the database connection information provided when you create a partner link or service and is used by iAS at run time in the background.

Use Enterprise Manager to modify the connection configuration so that the proper information is inserted into `oc4j-ra.xml` automatically.

> **Note:** The oc4j-ra.xml is the file that is administered through the Enterprise Manager console.

The following example of connection configuration for OracleAS Adapter for Oracle Applications can be used with the Oracle Applications adapter tutorials packaged with the product. For the logical deployment changes to take effect, the OC4J container process must be restarted.

To configure the connection information:

1. Log in to Enterprise Manager at the URL for your Web server host machine:

   `http://`*`servername`*`:`*`port`*`/em`

   > **Note:** The default port number is `8888`. The default login user ID and password are `oc4jadmin` and `welcome1` respectively.

2. Create a connection pool and a data source:

   1. Click **Home**, then **Administration**.
   2. Click the **Go To Task** icon for Services/JDBC Resources.
   3. Under Connection Pools, click the **Create** button.
   4. Accept the defaults, and click **Continue**.
   5. Enter the following values (leave defaults for the rest):

Field	Value
Name	`appsSample_pool`
JDBC URL	The URL for your database. For example: `jdbc:oracle:thin:@`*`host`*`:1521:`*`SID`*
Username	`apps`
Password	`apps`

   6. Click **Finish**, then click the **Test Connection** icon for your new connection pool.
   7. On the page that appears, click **Test**.

      Back on the main page, a successful connection message is displayed. If you see an error message, check the URL and credentials to ensure you've entered the right information.
   8. Click **Finish**.

9. Under Data Sources, click **Create**.

10. Accept the defaults and click **Continue**.

11. Enter the following values (leave defaults for the rest):

Field	**Value**
Name	`appsDemoDS`
JNDI Location	`jdbc/appsSampleDataSource`
Connection Pool	`appsSample_pool`

12. Click **Finish**.

3. Create an Oracle Applications adapter connection:

   1. At the top of the page, click the **OC4J:home** breadcrumb link, then the **Applications** link.

   2. In the tree of applications, click the **default** link.

   3. Under Modules, click the **AppsAdapter** link, then the **Connection Factories** link.

   4. Under Connection Factories, click **Create**.

      Note: Use the **Create** button near the top of the screen, not the one in the Shared Connection Pools section.

   5. Accept all the defaults and click **Continue**.

   6. For **JNDI Location**, enter `eis/Apps/appsSample`.

   7. Under Configuration Properties, for **xADataSourceName**, enter `jdbc/appsSampleDataSource`. Keep the default entries for all of the other fields.

   8. Click **Finish**.

B

Troubleshooting and Workarounds

This appendix covers the following topics:

- General Issues and Workarounds

General Issues and Workarounds

This section describes the following issues and workarounds:

- **Applications Context Information Default Values**

  Applications context is used in passing header variables that may be required in the SOA Suite to process Concurrent Programs and PL/SQL APIs. When setting the context, it takes into account the values passed for the header variables including *Username*, *Responsibility*, *Responsibility Application*, *Security Group*, and *NLS Language*.

  If the values for the new header properties *Responsibility Application*, *Security Group*, and *NLS Language* are not passed, context information will be determined based on *Username* and *Responsibility*.

  In the case of a null or empty value, the default *Username* is `SYSADMIN`, the default *Responsibility* is `SYSTEM ADMINISTRATOR`, the default *Security Group Key* is `Standard`, and the default *NLS Language* is `US`.

  You can change the default values specified in the generated WSDL. If you need to provide different context information for different invocations of the business process, then you can dynamically populate the header values. The context information can be passed through an Assign activity.

- **Correlation ID Defaults to BPEL for XML Gateway Transactions**

  The Adapter Configuration Wizard of OracleAS Adapter for Oracle Applications does not specify a correlation ID for XML Gateway transactions for inbound or outbound interfaces. Instead, a default correlation ID of `BPEL` is automatically set in the WSDL file. To make this configuration work, you must configure Oracle Applications to set the same correlation ID value of `BPEL` for the corresponding

XML Gateway transactions.

If you want the Adapter to use a different correlation ID than the default, you need to configure a correlation ID in Oracle Applications, and then edit the `Correlation="BPEL"` line contained in the `<jca:operation>` section of the adapter service WSDL. Replace `BPEL` with the string value of the correlation ID you specified in Oracle Applications.

- **Workaround for Stored Procedures Using Complex Types and the DEFAULT Clause**

  When working with stored procedures for which the Adapter Configuration Wizard must generate wrapper SQL stored procedures, there is a current limitation on DEFAULT clauses not being carried over to the generated wrapper stored procedures.

  As a workaround, perform the following steps one time only for a given stored procedure:

  1. Open the generated wrapper SQL script.

  2. Copy all default clauses from the base-stored procedure into the corresponding wrapper.

  3. Use SQL*Plus to reload the wrapper SQL script into the database.

  4. Edit the generated XSD. If a parameter has a DEFAULT clause, its corresponding element in the XSD must have the extra attribute: `db:default="true"`

     For example, with the following SQL:

     ```
     FINANCE$INVOICE(isTrue INTEGER DEFAULT 1, value NUMBER DEFAULT 0)
     ```

     The elements in the XSD for `isTrue` and `value` must have the new attribute:

     ```
     <element name="ISTRUE" ... db:default="true" .../>
     <element name="VALUE" ... db:default="true" .../>
     ```

- **One-time Workaround for Concurrent Programs and E-Commerce Gateway Interfaces**

  When working with Concurrent Programs and E-Commerce Gateway interfaces, you must perform the following workaround exactly once for a given Oracle E-Business Suite instance.

  > **Note:** This is to work around the known issue with the Adapter Configuration wizard being unable to preserve DEFAULT clauses for PL/SQL wrappers that it generates underneath the covers.

  Load the following SQL file into the apps schema (using SQL*Plus) before launching the Oracle Applications adapter of the Adapter Configuration Wizard to create services for either Concurrent Programs or E-Commerce Gateway Interfaces.

```
ORACLE_HOME
\bpel\samples\tutorials\150.AppsAdapter\OrderImportConcurrentProgram
\bpel\XX_BPEL_FND_REQUEST_SUBMIT_REQUEST.sql
```

- **Cannot Create a Partner Link If the Underlying API Has Been Recreated**

  The generation of a wrapper for an API that was recreated with the same name, but with a different set of parameters, will fail.

  > **Note:** This can happen for both packaged procedures and top-level or root procedures that require generated wrappers.

  The following example illustrates the problem:

  1. Create the initial API that, in this case, is defined at the top level:

     ```
     SQL> create procedure test (a number, b varchar2, c BOOLEAN)
     ```

     The `BOOLEAN` parameter indicates that a wrapper is necessary.

  2. Use the database adapter for stored procedures in the Adapter Configuration Wizard to generate and load the wrapper for this API.

  3. Drop the API, then recreate it with a different set of parameters:

     ```
     SQL> drop procedure test
     SQL> create procedure test (a number, b varchar2, c number, d
     BOOLEAN)
     ```

  4. An attempt to generate a partner link for this API using the Adapter Configuration Wizard will fail with the following message:

     ```
     The wrapper procedure, TOPLEVEL$TEST, could not be found
     ```

  5. As a workround, exit JDeveloper BPEL Designer and restart it after recreating the stored procedure, but before attempting to create the second partner link.

Index