Oracle® Fusion Middleware

Oracle E-Business Suite Adapter User's Guide

12*c* (12.2.1)

Part No. E66155-01

October 2015

ORACLE

Oracle Fusion Middleware Oracle E-Business Suite Adapter User's Guide, 12*c* (12.2.1)

Part No. E66155-01

Copyright © 2005, 2015, Oracle and/or its affiliates. All rights reserved.

Primary Author: Melody Yang

Contributing Author: Santiago Bastidas, Sravani Bheemireddy, Vimmika Dinesh, Rajesh Ghosh, Satish Menedi, Nadakuditi Ravindra, Veshaal Singh, Sheela Vasudevan

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 is software or related documentation that 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 END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware 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 that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about 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 unless otherwise set forth in an applicable agreement between you and Oracle. 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, except as set forth in an applicable agreement between you and Oracle.

Contents

Send Us Your Comments

Preface

1 What's New in This Guide

Overview 1-1
Significant Documentation Changes in this Document for 12c (12.2.1) 1-1

2 Introduction to Oracle E-Business Suite Adapter

Overview 2-1
Integration with Oracle Fusion Middleware 2-4
Integration with Oracle WebLogic Server 2-6

3 Oracle E-Business Suite Adapter Features

Overview 3-1
Support for Various Integration Interface Types 3-1
Support for Oracle Integration Repository 3-3
Support for Custom Integration Interfaces in Various Versions of Oracle E-Business Suite 3-4
High Availability 3-6
Monitoring Oracle E-Business Suite Adapter Configurations 3-7

4 Oracle E-Business Suite Adapter Concepts

Overview 4-1
Understanding Applications Context 4-1
Support for Normalized Message Properties 4-7

Support for Multiple Organization Access Control (MOAC)..... 4-11
Support for Multiple Languages..... 4-12
Understanding the Oracle E-Business Suite Module Browser..... 4-14
Understanding Oracle E-Business Suite Adapter Security..... 4-20
Flexfield Support for PL/SQL APIs and Open Interface Tables..... 4-27
Understanding Key Elements in Flexfield Configuration..... 4-29
Accessing Flexfield Mapping User Interface..... 4-34
Adding or Configuring a New Mapping..... 4-37
Configuring Key Flexfield Mappings for a PL/SQL API..... 4-38
Configuring Descriptive Flexfield Mappings for a PL/SQL API..... 4-59
Configuring Key Flexfield Mappings for an Open Interface Table..... 4-79
Configuring Descriptive Flexfield Mappings for an Open Interface Table..... 4-85
Importing an Existing Flexfield Mapping..... 4-115
Modifying Flexfield Mapping Definitions..... 4-119
Reviewing Flexfield Mapping Configurations..... 4-122
Logging..... 4-124
Enhanced Error and Exception Handling..... 4-132
Handling Functional Errors..... 4-132
Example of Error Handling Using APIErrorHandler JCA Property..... 4-133
Creating a New SOA Composite Application..... 4-134
Creating a Partner Link..... 4-136
Creating a Partner Link for File Adapter..... 4-137
Adding a CatchAll Activity on the Main Scope..... 4-139
Creating Invoke Activities..... 4-140
Creating Assign Activities..... 4-143
Validating and Testing SOA Composite Application with BPEL Process..... 4-149
Sample Payload for Creating a Project..... 4-153
Secured Connection Between Oracle E-Business Suite and Oracle Fusion Middleware SOA Suite Using J2EE Data Source Implementation..... 4-155

5 Using XML Gateway

Overview of XML Gateway..... 5-2
Design-Time Tasks for XML Gateway Inbound Messaging..... 5-6
Creating a New SOA Composite Application with BPEL Process..... 5-8
Creating a Partner Link..... 5-13
Adding a Partner Link for the File Adapter..... 5-18
Configuring the Invoke Activities..... 5-24
Configuring the Assign Activity..... 5-31
Run-Time Tasks for XML Gateway Inbound Messaging..... 5-34
Deploying the SOA Composite Application with BPEL Process..... 5-34

Testing the SOA Composite with BPEL Process 5-37
Verifying Records in Oracle E-Business Suite 5-39
Design-Time Task for XML Gateway Outbound Messaging 5-43
Creating a SOA Composite Application with BPEL Process 5-46
Adding a Partner Link 5-52
Adding a Receive Activity 5-57
Adding a Partner Link for File Adapter 5-59
Adding an Invoke Activity 5-64
Adding an Assign Activity 5-65
Run-Time Task for XML Gateway Outbound Messaging 5-67
Deploying the SOA Composite Application with BPEL Process 5-68
Testing the SOA Composite Application with BPEL Process 5-71
Troubleshooting 5-76

6 Using Business Events

Overview of Business Events 6-1
Business Events Concepts 6-2
Business Event Groups 6-3
Design-Time Tasks for Outbound Business Events 6-4
Creating a New SOA Composite Application with BPEL Process 6-6
Creating a Partner Link 6-10
Configuring the Receive Activity 6-20
Adding a Partner Link for the File Adapter 6-22
Configuring an Invoke Activity 6-29
Configuring an Assign Activity 6-31
Run-Time Tasks for Outbound Business Events 6-34
Deploying the SOA Composite Application with BPEL Process 6-34
Testing the SOA Composite Application with BPEL Process 6-38
Troubleshooting 6-46

7 Using Concurrent Programs

Overview of Concurrent Programs 7-1
Design-Time Tasks for Concurrent Programs 7-3
Creating a New SOA Composite Application with BPEL Project 7-4
Adding Partner Links 7-10
Adding Partner Links for File Adapter 7-27
Configuring Invoke Activities 7-39
Configuring Assign Activities 7-46
Run-Time Tasks for Concurrent Programs 7-50
Deploying the SOA Composite Application with BPEL Process 7-50

Testing the Deployed SOA Composite Application with BPEL Process....7-54
Verifying Records in Oracle E-Business Suite.... 7-58
Troubleshooting.... 7-59

8 Using Interface Tables and Views

Overview of Interface Tables and Views.... 8-2
Design-Time Tasks for Interface Tables.... 8-3
Creating a New SOA Composite Application with BPEL Process....8-3
Adding a Partner Link.... 8-8
Adding a Partner Link for File Adapter....8-22
Configuring Invoke Activities.... 8-28
Configuring an Assign Activity.... 8-31
Run-Time Tasks for Interface Tables.... 8-33
Deploying the SOA Composite Application with BPEL Process.... 8-34
Testing the Deployed SOA Composite Application with BPEL Process....8-37
Design-Time Tasks for Views....8-40
Creating a New SOA Composite Application with BPEL Process....8-41
Adding a Partner Link....8-46
Adding a Partner Link for File Adapter....8-58
Configuring Invoke Activities.... 8-64
Configuring an Assign Activity.... 8-68
Run-Time Tasks for Views.... 8-71
Deploying the SOA Composite Application with BPEL Process.... 8-71
Testing the Deployed SOA Composite Application with BPEL Process....8-74

9 Using PL/SQL APIs

Overview of PL/SQL APIs....9-1
Design-Time Tasks for PL/SQL APIs.... 9-2
Creating a New SOA Composite Application with BPEL Process....9-4
Adding Partner Links.... 9-8
Adding a Partner Link for File Adapter....9-19
Defining Wrapper APIs.... 9-25
Declaring Parameters with a DEFAULT Clause....9-28
Configuring the Invoke Activities.... 9-31
Configuring a Transform Activity.... 9-33
Configuring an Assign Activity.... 9-35
Run-Time Tasks for PL/SQL APIs.... 9-39
Deploying the SOA Composite Application with BPEL Process.... 9-39
Testing the Deployed SOA Composite Application with BPEL Process....9-42
Troubleshooting.... 9-46

10 Using e-Commerce Gateway

Overview of e-Commerce Gateway Integration 10-1
Design-Time Tasks for e-Commerce Gateway 10-2
Creating a New SOA Composite Application with BPEL Process 10-3
Adding a Partner Link 10-9
Adding a Partner Link for File Adapter 10-14
Configuring Invoke Activities 10-20
Configuring an Assign Activity 10-25
Run-Time Tasks for e-Commerce Gateway 10-27
Deploying the SOA Composite Application with BPEL Process 10-28
Testing the Deployed SOA Composite Application with BPEL Process 10-31
Verifying Records in Oracle E-Business Suite 10-33

11 Runtime Configuration for Oracle E-Business Suite Adapter

Overview 11-1
Configuring the Data Source in Oracle WebLogic Server 11-1
Creating Connection Factory for Oracle E-Business Suite Adapter 11-7
Creating the Application Server Connection 11-10

A Troubleshooting and Workarounds

General Issues and Workarounds A-1

B Oracle E-Business Suite Adapter Properties

Overview B-1

Index

Send Us Your Comments

Oracle Fusion Middleware Oracle E-Business Suite Adapter User's Guide, 12*c* (12.2.1)

Part No. E66155-01

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 Oracle E-Business Suite 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 Fusion Middleware Oracle E-Business Suite Adapter User's Guide*, 12*c* (12.2.1)

This documentation is written for the technical consultants, implementers and system integration consultants who use Oracle Fusion Middleware Oracle E-Business Suite Adapter (formerly known as 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 Fusion Middleware, Oracle WebLogic Server, and PL/SQL technology.
- Oracle E-Business Suite 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 xii for more Oracle E-Business Suite product information.

Documentation Accessibility

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

Access to Oracle Support

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

Structure

1 What's New in This Guide
2 Introduction to Oracle E-Business Suite Adapter
3 Oracle E-Business Suite Adapter Features
4 Oracle E-Business Suite Adapter Concepts
5 Using XML Gateway
6 Using Business Events
7 Using Concurrent Programs
8 Using Interface Tables and Views
9 Using PL/SQL APIs
10 Using e-Commerce Gateway
11 Runtime Configuration for Oracle E-Business Suite Adapter
A Troubleshooting and Workarounds
B Oracle E-Business Suite Adapter Properties

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 Oracle Fusion Middleware Oracle E-Business Suite Adapter.

Documentation

You may want to refer to other Oracle Fusion Middleware guides when you set up and use Oracle Fusion Middleware Oracle E-Business Suite Adapter. You can read the guides online by reading from the Oracle Fusion Middleware or Oracle Database Documentation Library CD included in your media pack, or on the Oracle Technology Network (OTN).

To download free release notes, installation documentation, white papers, or other collateral, please visit the main OTN page.

Training

Oracle offers a complete set of training courses to help you and your staff master Oracle Fusion Middleware 12*c* 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.

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 Fusion Middleware 12*c* 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 E-Business Suite Data

Oracle STRONGLY RECOMMENDS that you never use SQL*Plus, Oracle Data Browser, database triggers, or any other tool to modify Oracle E-Business Suite 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 E-Business Suite data, you risk destroying the integrity of your data and you lose the ability to audit changes to your data.

Because Oracle E-Business Suite tables are interrelated, any change you make using an Oracle E-Business Suite form can update many tables at once. But when you modify Oracle E-Business Suite data using anything other than Oracle E-Business Suite, 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 E-Business Suite.

When you use Oracle E-Business Suite to modify your data, Oracle E-Business Suite automatically checks that your changes are valid. Oracle E-Business Suite 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

What's New in This Guide

Overview

This chapter introduces significant changes that are described in this guide for Oracle E-Business Suite Adapter and provides pointers to additional information.

For a list of known issues (release notes), see the "Known Issues for Oracle SOA Products and Oracle AIA Foundation Pack" at http://www.oracle.com/technetwork/middleware/docs/soa-aiafp-knownissuesindex-364630.html.

Significant Documentation Changes in this Document for 12*c* (12.2.1)

For Oracle E-Business Suite Adapter 12c (12.2.1), the runtime configuration for Oracle E-Business Suite Adapter that appeared within an Appendix in the 12*c* (12.1.3) version of this guide has been moved to its own chapter called Runtime Configuration for Oracle E-Business Suite Adapter, page 11-1. Therefore, the Appendix has been removed.

2

Introduction to Oracle E-Business Suite Adapter

Overview

Oracle E-Business Suite is a set of integrated business applications that runs entirely on the Internet. Oracle E-Business Suite offers you the following:

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

Oracle E-Business Suite 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 E-Business Suite to non-Oracle applications, you use Oracle Fusion Middleware Oracle E-Business Suite Adapter.

Oracle E-Business Suite Adapter (formerly known as Adapter for Oracle Applications) not only provides comprehensive, bidirectional, multimodal, synchronous, and asynchronous connectivity to Oracle E-Business Suite, but also supports for all modules of Oracle E-Business Suite in Release 12 and Release 11.5.10 including custom integration interfaces in various versions of Oracle E-Business Suite.

> **Important:** The support for various versions of Oracle E-Business Suite has the following conditions:
>
> - The required Oracle E-Business Suite database version for Oracle E-Business Suite Adapter 12*c* (12.2.1) is listed in the following table:

Oracle Database Release	Oracle Database 11*g* Release 1	Oracle Database 11*g* Release 2	Oracle Database 12*c* Release 1
Minimum Required Database Version	11.1.0.7 or higher	11.2.0.3 or higher	12.1.0.1 or higher

- Oracle E-Business Suite Adapter supports Oracle E-Business Suite Release 11.5.10 (OWF.G.Rollup 7 required) as well as Release 12.0 and above.
- 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 My Oracle Support (formerly Oracle*MetaLink*) Knowledge Document 787637.1 for details.

Major Features

Oracle E-Business Suite Adapter provides the following features:

- It leverages the Integration Repository to provide the information from the source of truth on integration.
- 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 JCA extension.

  Note: For more information, see *Oracle Fusion Middleware User's Guide for Technology Adapters*.

- It works under the securely configured connection between Oracle E-Business Suite and Oracle Fusion Middleware using just the FND User name 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

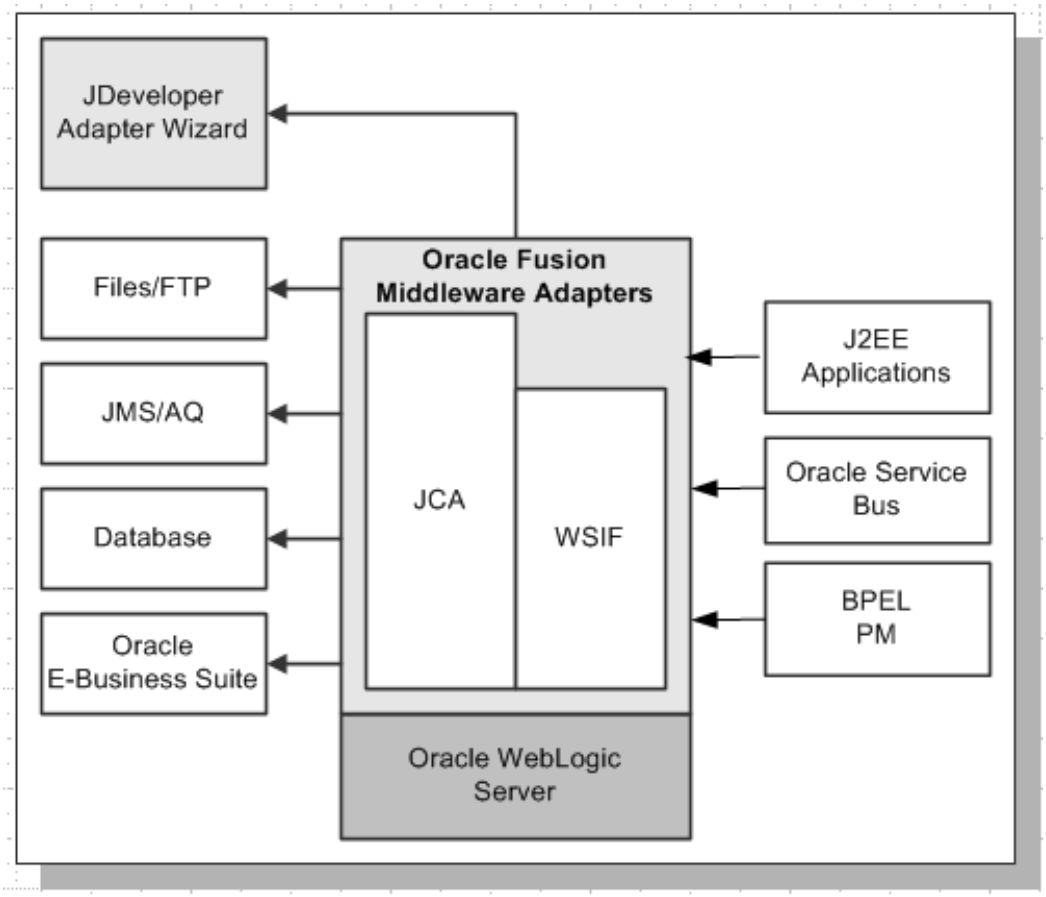
Web services to update Oracle E-Business Suit applications.

- It implicitly takes care of applications context without bothering about the complexities of initiating the same explicitly.
- It supports multiple languages and multiple organization access control (MOAC) setup.
- It uses a JDeveloper based design-time tool for dynamically browsing the Oracle E-Business Suite application interfaces 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 as well as run time.
- It provides flexfield support for PL/SQL APIs and Open Interface Tables so that flexfield information can be displayed at design time and run time.

Architecture

Oracle E-Business Suite Adapter is based on JCA 1.5 standards and deployed as a resource adapter within the Oracle WebLogic Server container. The architecture of Oracle E-Business Suite Adapter is similar to the architecture of technology adapters.

Oracle E-Business Suite Adapter Architecture



For more information on technology adapters, see *Oracle Fusion Middleware User's Guide for Technology Adapters*.

Installing Oracle E-Business Suite Adapter

Oracle E-Business Suite Adapter and Oracle JCA Adapters are available as part of the Oracle Fusion Middleware install. In addition, these adapters support both Oracle WebLogic Server and middle tier deployments.

For more information, see *Oracle Fusion Middleware Installation Guide for Oracle SOA Suite and Oracle Business Process Management Suite*.

Integration with Oracle Fusion Middleware

Oracle E-Business Suite Adapter integrates with the JCA Binding Component of the Oracle Fusion Middleware platform; therefore, it integrates with service engines, such as Oracle BPEL Process Manager (Oracle BPEL PM) and Oracle Mediator.

Oracle E-Business Suite Adapter 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 while integrating with Oracle BPEL PM and Oracle Mediator. At run time, Oracle E-Business Suite integration flows are deployed in the Oracle WebLogic Server for execution of the services to complete the integration.

- **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.

- **Oracle Mediator**

  Oracle Mediator provides a lightweight framework to route data from service providers to external partners. In addition, it can subscribe to and publish business events, as well as transform data using XSL Transformations.

  For example, a Mediator can accept data contained in a text file from an application or service, transform it to a format appropriate for updating a database that serves as a customer repository, and then route and deliver the data to that database.

Design Time

While integrating with Oracle BPEL PM and Oracle Mediator, Oracle E-Business Suite Adapter uses Oracle JDeveloper as the design-time tool to create SOA Composite applications and generate WSDL and JCA files for the Web services.

When you create a partner link in Oracle JDeveloper BPEL Designer, the Adapter Configuration Wizard starts and allows you to select and configure Oracle E-Business Suite Adapter or other adapters. With proper database and service connection setups, you are presented with a functionally organized list of inbound and outbound interfaces available in your Oracle E-Business Suite instance for which you select the one that fulfill your requirements. 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

Oracle E-Business Suite Adapter is based on the JCA 1.5 specification. A Composite application including the BPEL processes, Mediator services, and partner link definitions generated at design time is deployed to the Oracle WebLogic Server. A JCA Binding Component acts as the glue layer that integrates the standard JCA 1.5 resource adapter with the Oracle BPEL Process Manager during run time. The JCA Binding Component acts as a pseudo JCA 1.5 container.

> **Note:** Only the JCA 1.5 integration allows the BPEL PM to receive inbound events (from EIS to J2EE/BPEL PM). The Oracle BPEL Process

Manager acts as a pseudo JCA 1.5 container and implements the JCA 1.5-specific System Contracts. The JCA 1.5 resource adapter and the BPEL PM instance must be deployed in the same Oracle WebLogic Server container.

The Web service invocation launched by the BPEL Invoke activity contained in the SOA Composite is converted to a JCA CCI (Common Client Interface) outbound interaction, and the JCA response is converted back to a Web service response. This end-to-end invocation is synchronous.

Testing the SOA Composite Application at Run Time

After deploying the SOA Composite application, you should validate the design by testing the BPEL process contained in the deployed SOA Composite application to test the interface integration.

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

Integration with Oracle WebLogic Server

Oracle WebLogic Server is a scalable, enterprise-ready Java Platform, Enterprise Edition (Java EE) application server. Its infrastructure enables enterprises to deploy mission-critical applications in a robust, secure, highly available, and scalable environment and is an ideal foundation for building applications based on service-oriented architectures (SOA). SOA is a design methodology aimed at maximizing the reuse of application services.

In addition, Oracle WebLogic Server consists of a JCA container for hosting JCA resource adapters. JCA defines standard Java interfaces for simplifying the integration of a J2EE server with various back-end applications. All client applications run within the Oracle WebLogic Server environment.

Design Time

Oracle JDeveloper is used to create Web services represented in WSDL files and XML Schema Definition (XSD) files for the adapter request-response service.

The Oracle WebLogic Server clients use these XSD files during run time for calling the JCA outbound interaction.

Run Time

Oracle E-Business Suite Adapter is based on the JCA 1.5 specification and is deployed as resource adapter within the Oracle WebLogic Server container. The JCA 1.5 specification addresses the life-cycle management, message-inflow (for Adapter Event publish), and work management contracts.

For more information about using Oracle WebLogic Server with Oracle JDeveloper, see the Using WebLogic Server with Oracle JDeveloper section, *Oracle Fusion Middleware Installation Guide for Oracle JDeveloper*.

3

Oracle E-Business Suite Adapter Features

Overview

Oracle E-Business Suite Adapter enables you to orchestrate discrete data into a meaningful business process and creates Web services for various 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.

Oracle E-Business Suite Adapter provides the following features which are further discussed in this chapter:

- Support for Various Integration Interface Types, page 3-1.
- Support for Oracle Integration Repository, page 3-3
- Support for Custom Integration Interfaces in Various Versions of Oracle E-Business Suite, page 3-4
- High Availability, page 3-6
- Monitoring Oracle E-Business Suite Adapter Configurations, page 3-7

Support for Various Integration Interface Types

Oracle E-Business Suite Adapter acts as a highly flexible integration interface for Oracle E-Business Suite. It supports the following interface types for integrating with Oracle E-Business Suite applications:

- **PL/SQL APIs**

  These APIs enable you to insert and update data in Oracle E-Business Suite 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.

  The Business Event System 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, which lets you register subscriptions to significant events.

  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.

  Business Event Groups

  A business event group is a type of event containing a set of individual business events. Once an event group is defined, a subscription that is registered for an event group will be executed when any of the individual events within it is triggered.

  With the support for business event groups, different business events belonging to an even group can be handled through a single partner link. Service created for an event group would be able to dequeue payloads corresponding to any of the events within the group.

- **Open Interface Tables**

  Interface tables enable you to insert or update data into Oracle E-Business Suite. 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 E-Business Suite. It helps you to insert and retrieve data from Oracle E-Business Suite. XML Gateway is a higher-level interface that exposes OAGIS-formatted XML documents for commonly used Oracle E-Business Suite 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 E-Business Suite. 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 E-Business Suite and third party applications.

- **Interface Views**

  Interface views help you to retrieve data from Oracle E-Business Suite using the application tables.

Please note that Oracle E-Business Suite Adapter also supports the following custom integration interface types that are exposed by the Oracle E-Business Suite Module Browser, not by Oracle Integration Repository:

- Custom PL/SQL APIs
- Custom Business Events

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

- Custom XML Gateway Maps
- Customized Concurrent Programs

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 E-Business Suite. It provides a comprehensive view of the integration interfaces with details for Oracle E-Business Suite.

Oracle Integration Repository can only provide information about an integration interface that has been specifically annotated by the developer to make it public. Oracle E-Business Suite Adapter takes advantage of the annotations that have already been created to make the following integration interface types visible in the Oracle E-Business Suite 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 section, *Oracle E-Business Suite Integrated SOA Gateway User's Guide*. This guide is part of the Oracle E-Business Suite Documentation Library which can be accessed on the Oracle Technology Network (OTN).

> **Note:** Oracle Integration Repository is integral part of Oracle E-Business Suite Release 12.0 and onwards.

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

Oracle E-Business Suite Adapter leverages Integration Repository for Oracle E-Business Suite Release 11.5.10 and Release 12 as the source of truth for the integration content.

Oracle E-Business Suite Adapter supports selecting custom integration interfaces from the Integration Repository for the following versions of Oracle E-Business Suite:

- Release 12, page 3-5
- Release 11.5.10, page 3-6

> **Important:** Please note that the support for various versions of Oracle E-Business Suite has the following conditions:
>
> - The required Oracle E-Business Suite database version for Oracle E-Business Suite Adapter 12*c* (12.2.1) is listed in the following table:
>
> | **Oracle Database Release** | **Oracle Database 11*g* Release 1** | **Oracle Database 11*g* Release 2** | **Oracle Database 12*c* Release 1** |
> |---|---|---|---|
> | Minimum Required Database Version | 11.1.0.7 or higher | 11.2.0.3 or higher | 12.1.0.1 or higher |
>
> - Oracle E-Business Suite Adapter supports Oracle E-Business Suite Release 11.5.10 (OWF.G.Rollup 7 required) as well as Release 12.0 and above.

- 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.

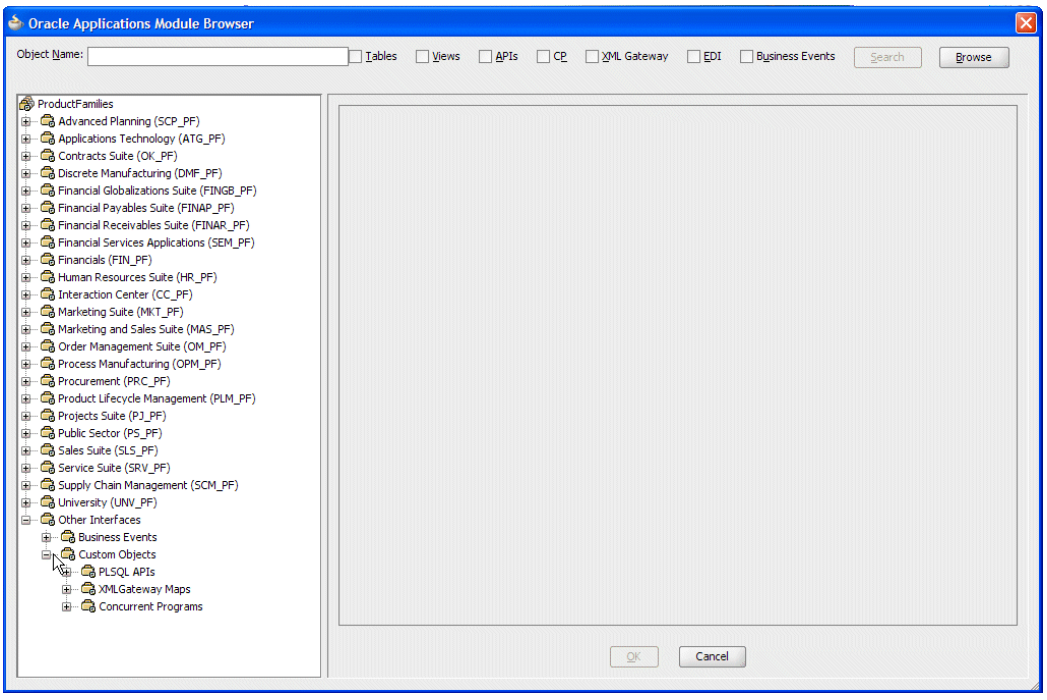
  See My Oracle Support Knowledge Document 787637.1 for details.

From the business service creation and run-time perspectives, Oracle E-Business Suite Adapter supports custom PL/SQL APIs as far as the packages are available in the APPS schema. The Oracle E-Business Suite Module Browser can expose these custom 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 Oracle E-Business Suite Adapter to directly connect to the live database of Oracle Integration Repository querying for the public interfaces and then displaying the list of custom integration interfaces under the `Other Interfaces > Custom Objects` node in the Oracle E-Business Suite Module Browser.

Supporting Custom Integration Interfaces in Release 12



Please note that Oracle E-Business Suite Adapter allows you to extract the Integration Repository data file from the live database you connect to Oracle E-Business Suite 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

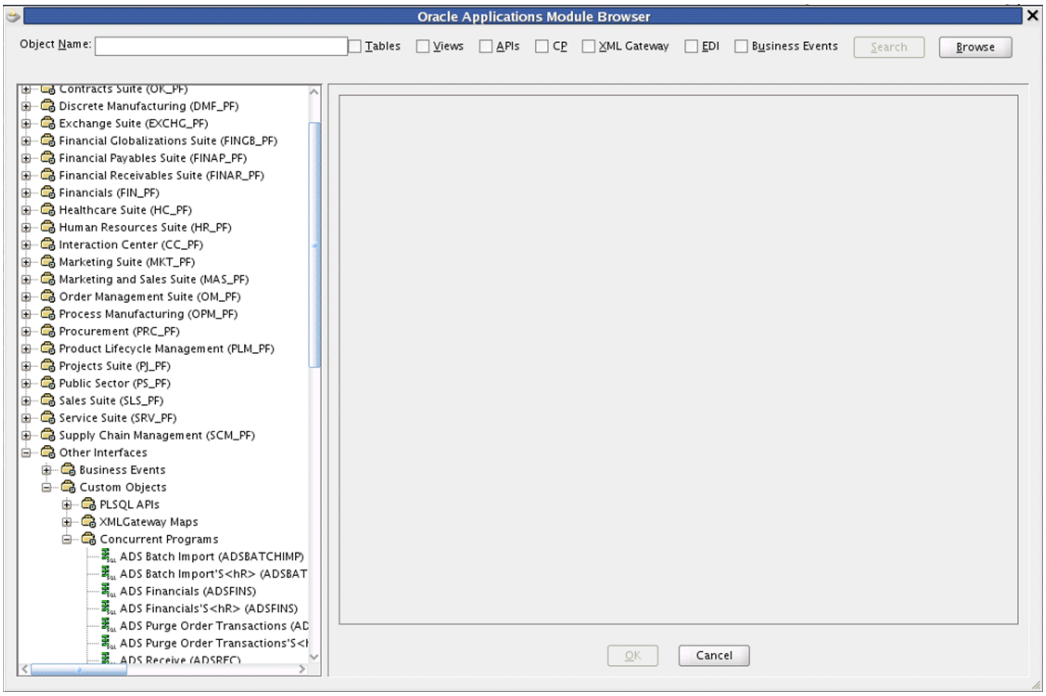
workplace.

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 Oracle E-Business Suite Release 11.5.10, Oracle E-Business Suite Adapter provides the Integration Repository data file bundled as part of the product in xml format. During design time, Oracle E-Business Suite Adapter queries public interfaces from the native XML data file of the Integration Repository shipped with Adapter. Custom integration interfaces are queried directly from the live database and displayed under the `Other Interfaces > Custom Objects` node in the Module Browser.

Supporting Custom Integration Interfaces in Release 11.5.10



High Availability

Oracle E-Business Suite Adapter supports high availability by completely relying on the functionality of the underlying Oracle Database Adapter and Oracle AQ Adapter to support the scenarios, such as Real Application Cluster (RAC) database configuration or clustering configuration.

For information about Oracle Database Adapter and Oracle AQ Adapter, see *Oracle Fusion Middleware User's Guide for Technology Adapters*.

Monitoring Oracle E-Business Suite Adapter Configurations

To effectively diagnose and troubleshoot potential configuration issues, Adapter configuration reports can be enabled and displayed in the Oracle Fusion Middleware Control.

For information about these configuration reports, see the Monitoring Oracle JCA Adapters chapter, *Oracle Fusion Middleware Administering Oracle SOA Suite and Oracle Business Process Management Suite*.

4

Oracle E-Business Suite Adapter Concepts

Overview

This chapter discusses some essential concepts for Oracle E-Business Suite Adapter. It provides basic understanding about how applications context, security, logging, and flexfields are used or supported as well as how errors and exceptions are handled. Detailed concepts described in this chapter are:

- Understanding Applications Context, page 4-1
- Understanding the Oracle E-Business Suite Module Browser, page 4-14
- Understanding Oracle E-Business Suite Adapter Security, page 4-20
- Flexfield Support for PL/SQL APIs and Open Interface Tables, page 4-27
- Logging, page 4-124
- Enhanced Error and Exception Handling, page 4-132
- Secured Connection Between Oracle E-Business Suite and Oracle Fusion Middleware SOA Suite Using J2EE Data Source Implementation, page 4-155

Understanding Applications Context

Applications context is required for invoking interfaces within Oracle E-Business Suite. Applications context is a combination of **Username**, **Responsibility**, **Responsibility Application**, and **Security Group**. These elements may be required for executing an interface successfully within Oracle E-Business Suite.

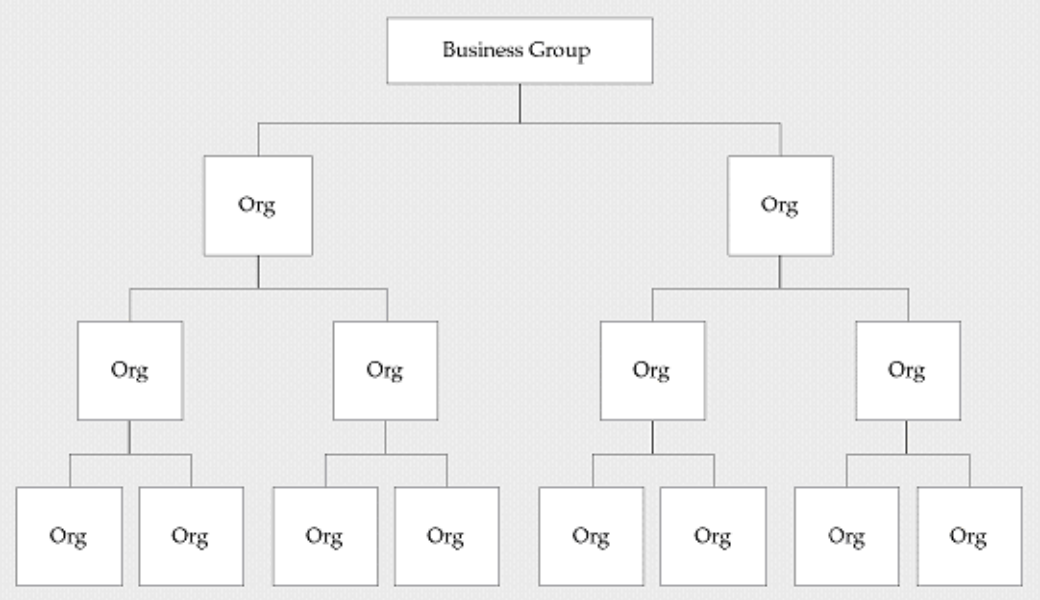
Applications Context in Multiple Organizations

You can define multiple organizations and the relationships between them in a single

installation of Oracle E-Business Suite. 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.

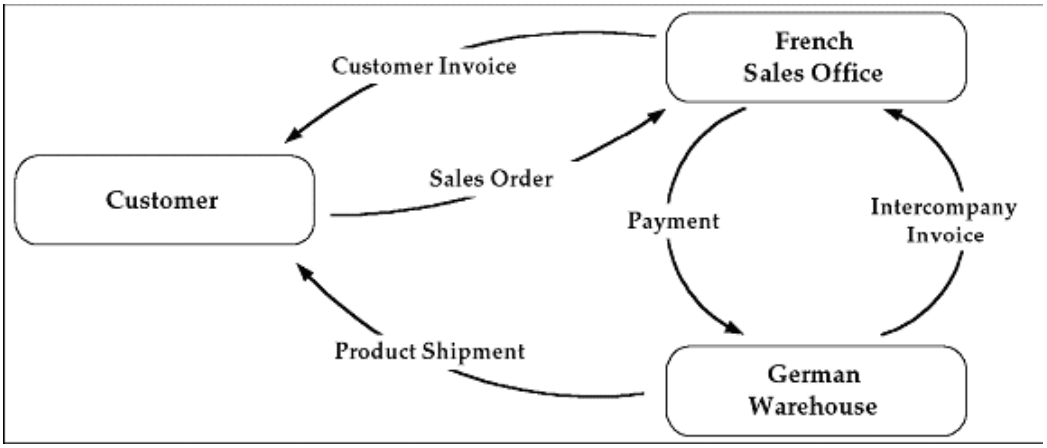
Business Group Hierarchy



Example of a Multiple-Organization Setup

Using the accounting, distribution, and materials management functions in Oracle E-Business Suite, 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

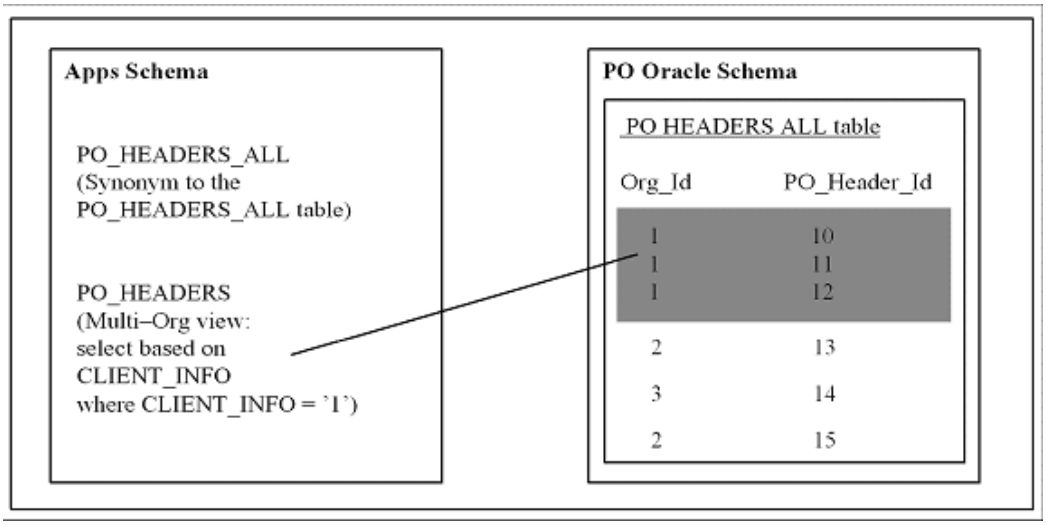


Consider two different organizations in your company: One is a French sales office and the other is a German warehouse. There is a sales order transaction with the customer, and this illustrates how the entire Order-to-Deliver process would work:

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.

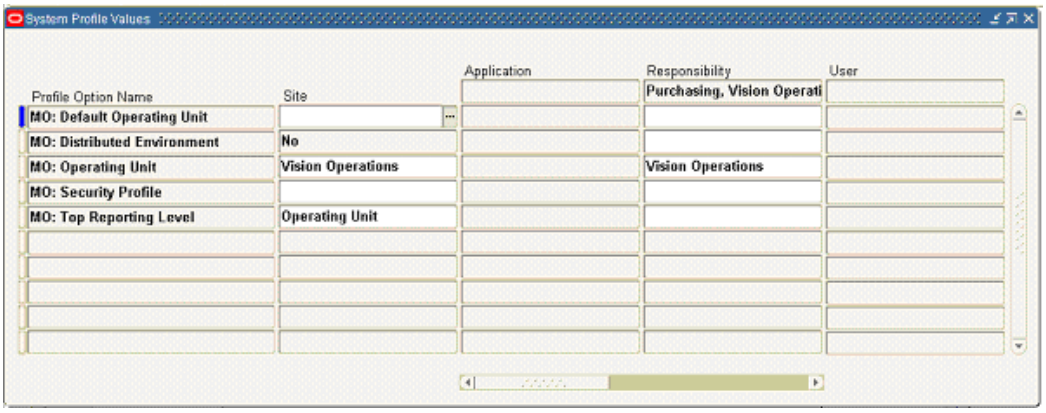
Schema Synonyms



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



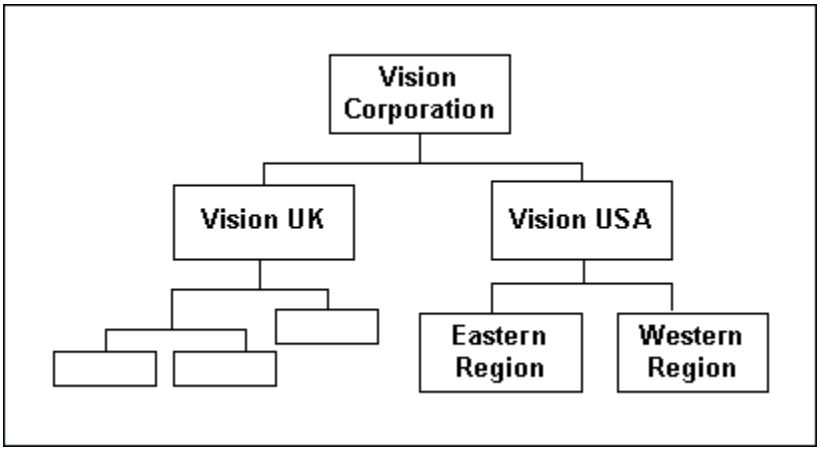
To have a secured way for users to allow access or report on data for the operating units they have access to, many of the interfaces within Oracle E-Business Suite make use of the MOAC security feature to determine the operating unit access privileges and derive the Organization ID based on relevant profile values. Oracle E-Business Suite Adapter can implicitly perform the requisite MOAC setup. If ORG_ID is passed, data access would be set to the passed Organization ID.

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.

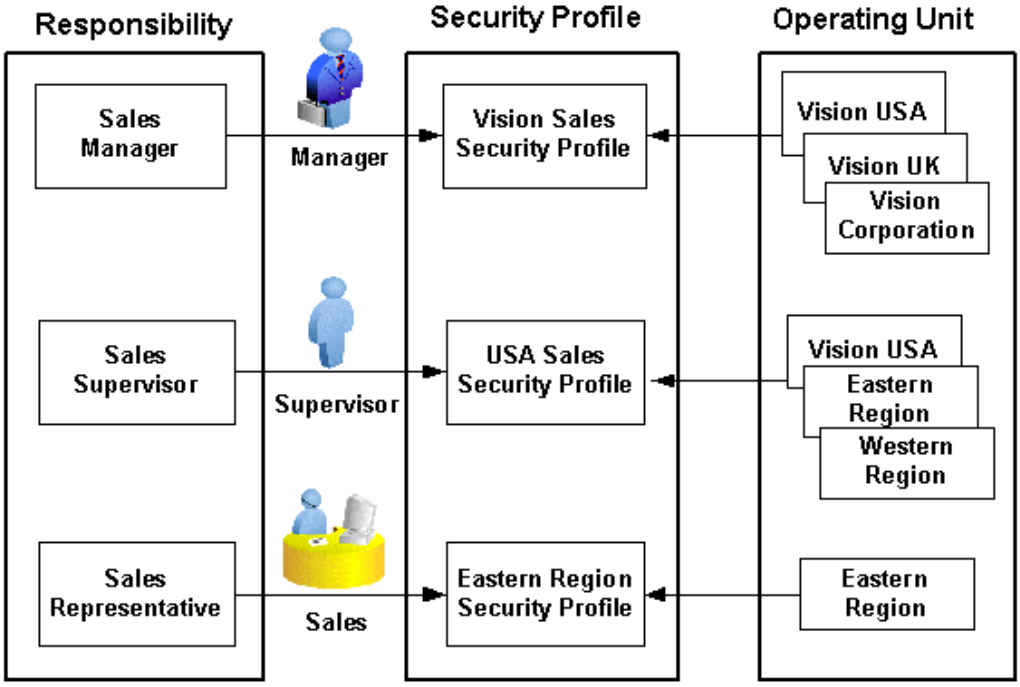
Security profiles are defined based on organization hierarchies. For example, a sales company consists of USA and UK operating units; the USA operating unit has Western Region Sales and East Region Sales. Sales managers are responsible for both USA and UK sales, supervisors are responsible for either 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 sales data access privileges are grouped into the Vision Sales security profile. A USA Sales security profile is created for USA related data, and a regional security profile is created 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.

1. When integrating with Oracle E-Business Suite using PL/SQL APIs or Concurrent Programs, applications context is set based on the values passed for the header properties Username, Responsibility, Responsibility Application and Security Group.

   Note: For backward compatibility reasons, no context related exceptions or errors would be thrown if none of the following header properties are passed - Responsibility Application, Security Group, and NLS Language.

2. MOAC setup is done based on the Responsibility Application to which the user belongs. If Organization ID is passed, the Organization access would be set to the passed Organization.

3. If the user needs the invocation of the interface to happen in a desired language, then the header property NLS Language can be passed. Default value for NLS Language would be the language specified in the user's preference in Oracle

E-Business Suite.

To integrate business processes with Oracle E-Business Suite, it is essential to propagate these applications context values through a flexible mechanism, which is provided by Oracle E-Business Suite Adapter through the header properties.

The following topics are discussed in this section:

- Support for Normalized Message Properties, page 4-7
- Support for Multiple Organization Access Control (MOAC), page 4-11
- Support for Multiple Languages, page 4-12

Support for Normalized Message Properties

To effectively set applications context values required in a BPEL process or to populate mandatory header variables for XML Gateway inbound transactions to be completed successfully, Oracle E-Business Suite Adapter provides a flexible mechanism that allows each context value and header variable to be set and passed in the adapter user interface directly through the Invoke activity. This message normalization feature not only provides a flexible solution on header support, but also simplifies the design-time tasks without using an Assign activity to pass header values.

Setting Message Properties for Applications Context

The following header properties are used in setting applications context for PL/SQL and Concurrent Program interfaces:

- `jca.apps.Username`
- `jca.apps.Responsibility`
- `jca.apps.ORG_ID`
- `jca.apps.RespApplication`
- `jca.apps.SecurityGroup`

> **Note:** Existing header property `jca.apps.Responsibility` used in the earlier releases can now take Responsibility Key as well as Responsibility Name as input. If the header property `jca.apps.NLSLanguage` is set, and Responsibility Name is passed, the value passed for `jca.apps.Responsibility` is expected to be in the same language. However, Responsibility Key as well as all other header properties are language independent.
>
> All these header properties would be used together to set the application 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 Username is `SYSADMIN`, the default Responsibility is `System Administrator`, the default Security Group Key is `Standard`, and the default NLS Language is `US`.

Setting Message Properties for XML Gateway Inbound Transactions

The following header message properties are used in setting XML Gateway information required for XML Gateway inbound/enqueue transactions:

- `jca.apps.ecx.TransactionType`
- `jca.apps.ecx.TransactionSubtype`
- `jca.apps.ecx.PartySiteId`
- `jca.apps.ecx.MessageType`
- `jca.apps.ecx.MessageStandard`
- `jca.apps.ecx.DocumentNumber`
- `jca.apps.ecx.ProtocolType`
- `jca.apps.ecx.ProtocolAddress`
- `jca.apps.ecx.Username`
- `jca.apps.ecx.Password`
- `jca.apps.ecx.Attribute1`
- `jca.apps.ecx.Attribute2`
- `jca.apps.ecx.Attribute3`
- `jca.apps.ecx.Attribute4`
- `jca.apps.ecx.Attribute5`
- `jca.apps.ecx.Payload`

Design-Time Tasks for Message Properties Support

Oracle E-Business Suite Adapter uses the following procedures to complete the design-time tasks to support message normalization:

1. Create a new SOA Composite application with BPEL process, page 5-8
2. Create a partner link, page 5-13

3. Configure an Invoke Activity, page 5-64

   This activity involves the following tasks:

   - *Configure basic information in the General tab:*

     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 a Output Variable for the Invoke activity.

     For detailed instructions, see Configure basic information in the General tab, page 5-25

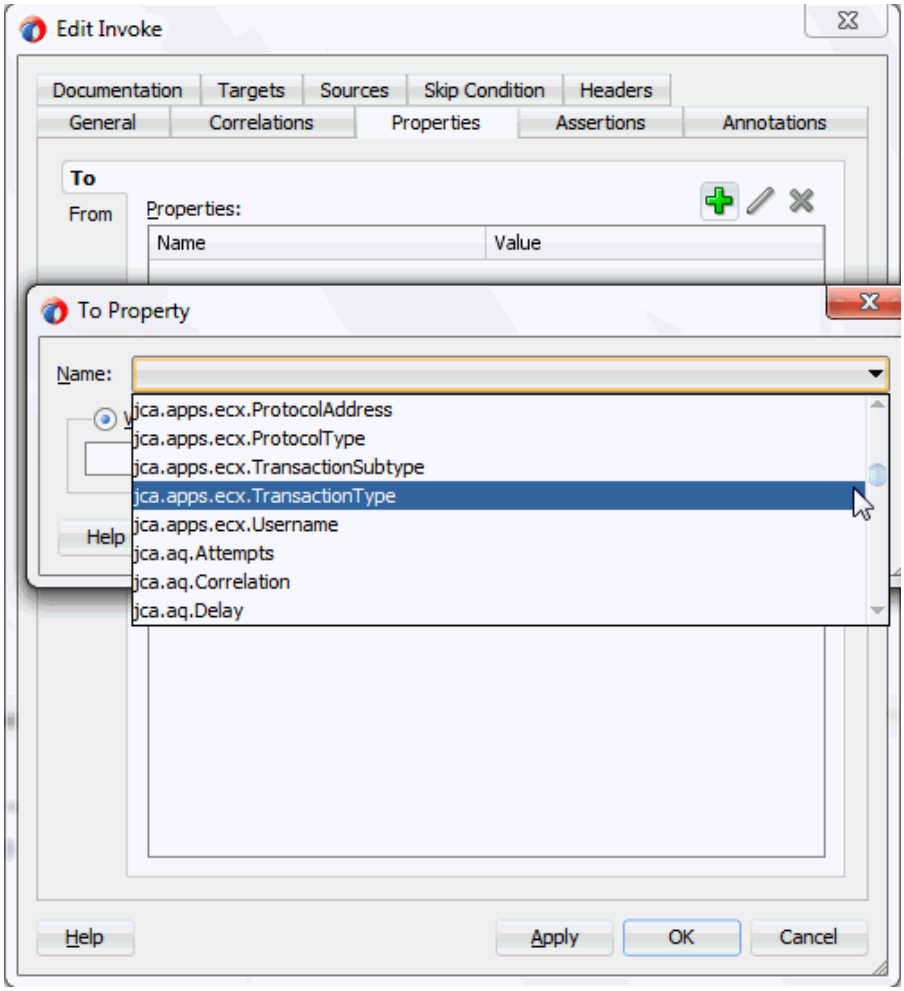
   - *Set the Header Message Properties in the Properties tab:*

     This is to set the necessary message properties for the following purposes:

     - To set XML Gateway header variables for an inbound transaction.

       For example, locate XML Gateway header property `jca.apps.ecx.TransactionType` first and then enter a value (such as `'PO'`) for the selected property.

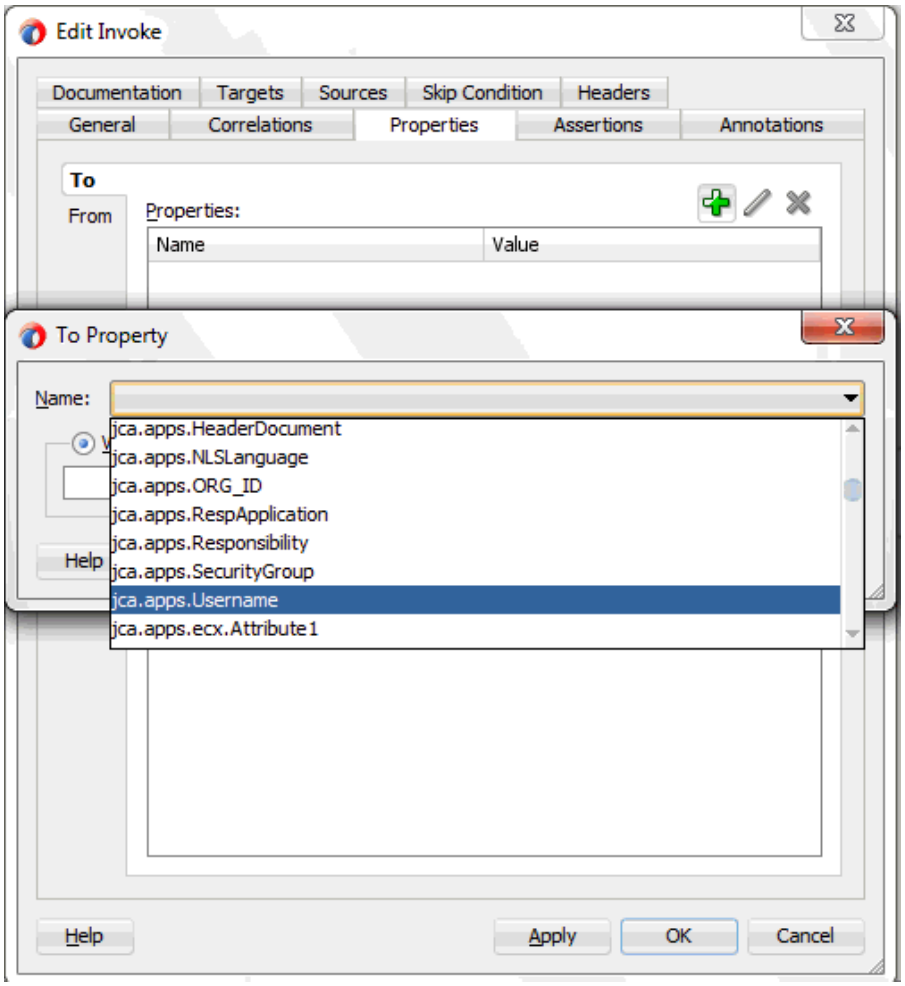
Entering a Selected Property Value



For detailed instructions, see Setting ECX Header Message Properties, page 5-28.

- To set applications context for Oracle E-Business Suite to identify the application user, responsibility, and the user's associated organization information.

  For example, locate the `jca.apps.Username` property from the property list and then enter `'OPERATIONS'` as the property value.



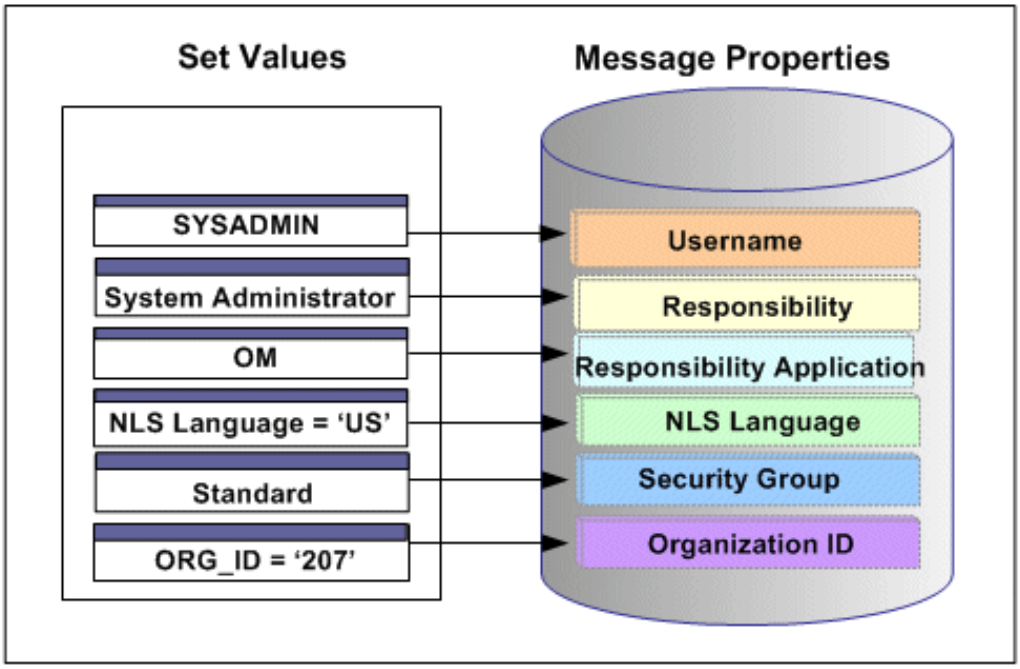
For detailed instructions, see Setting the Header Properties for Applications Context, page 7-44.

Support for Multiple Organization Access Control (MOAC)

Oracle E-Business Suite Adapter allows Organization ID to be passed as a header property which is available as `jca.apps.ORG_ID` within the Properties tab of an Invoke activity during the design time.

On Oracle E-Business Suite subsequent to Release 12.0, MOAC setup is automatically done using the call `MO_GLOBAL.INIT(<Application Name>)`. Subsequently, if ORG_ID is passed, the call to `MO_GLOBAL.set_policy_context('S', <ORG_ID>)` would be made to set the Organization access to the specified ORG_ID.

Setting Header Message Properties



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 has been assigned with a property 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.

Support for Multiple Languages

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

NLS Language Property Value Takes the Priority

When an application user retrieves data from the system for a transaction or receives error messages while executing APIs, the session language of data or error messages can be determined based on the following conditions:

1. The NLS Language value can be set by using the header property `jca.apps.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 property is not passed, Oracle E-Business Suite Adapter 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 (`jca.apps.NLSLanguage`) property 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 property value, then the user's default language, and last determined by the default NLS Language property value 'US'.

For more information about how to set NLS Language property value by using `jca.apps.NLSLanguage`, see Support for Normalized Message Properties, page 4-7.

Determining a User's Default Language

To identify the default language used in the database session for data query and retrieval, Oracle E-Business Suite Adapter 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, Oracle E-Business Suite Adapter 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, Oracle E-Business Suite Adapter will first examine if the NLS Language property 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 property is not passed, Oracle E-Business Suite Adapter 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 E-Business Suite.

> **Note:** The default language set in the General Preference 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 E-Business Suite chapter, *Oracle E-Business Suite User's Guide* for the information on how to set the user preferences.

Understanding the Oracle E-Business Suite Module Browser

Oracle E-Business Suite Module Browser, formerly known as Oracle Applications Module Browser, is a key component of Oracle E-Business Suite Adapter. In addition to the interfaces that are made available through Oracle Integration Repository, Oracle E-Business Suite Adapter enables you to use business events and event groups, custom PL/SQL APIs, and custom XML Gateway maps, all of which you can explore using the Oracle E-Business Suite Module Browser.

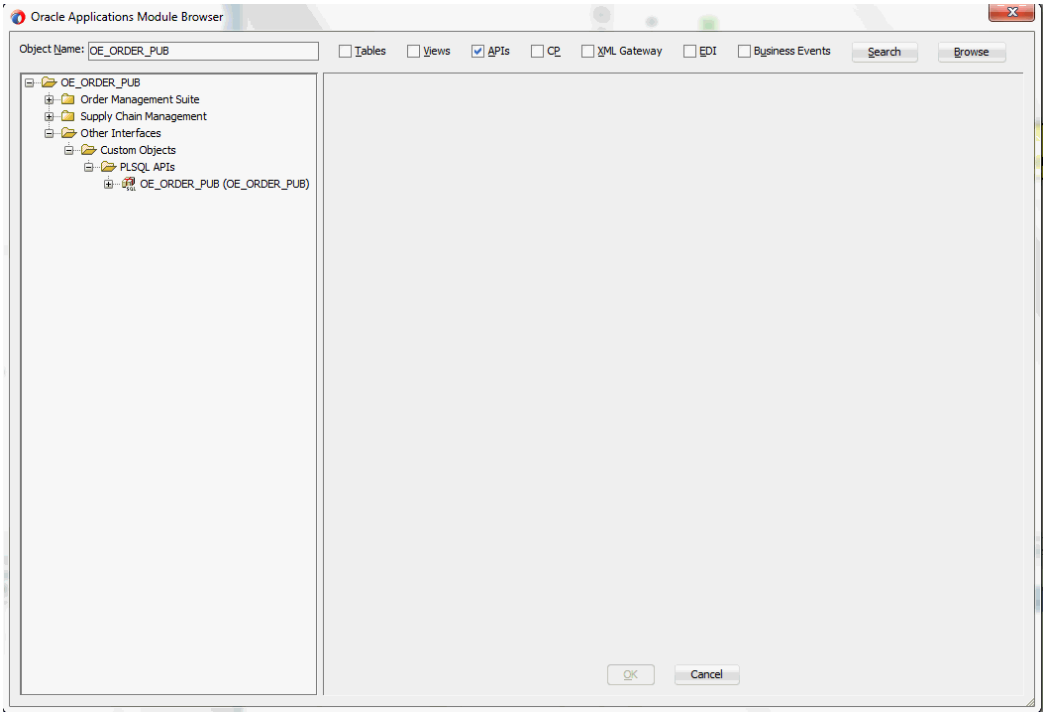
Use the Module Browser to select the interface needed to define a partner link. It can be used in the following ways to locate desired interfaces:

- Locating interfaces by Search
- Locating interfaces by Browse

Searching Interfaces

Enter desired interface name or partial values containing wildcard characters in the Object Name field and click **Search** to locate the desired interface in the Module Browser.

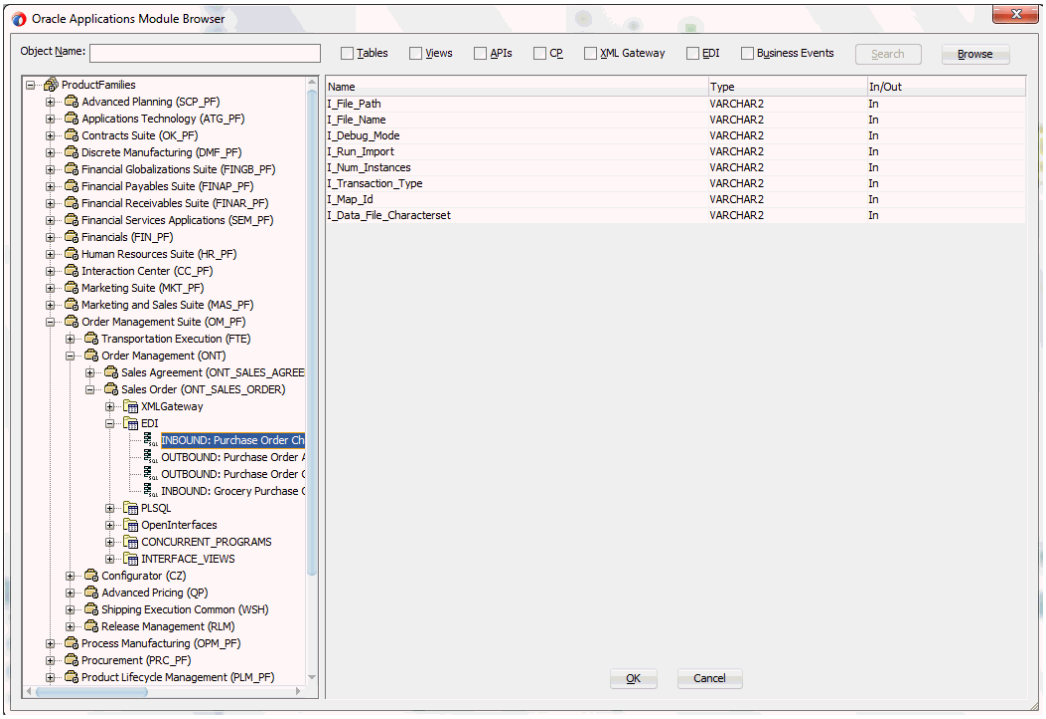
Searching Interfaces Through Oracle E-Business Suite Module Browser



Browsing Interfaces Through Tree Structure

The Module Browser combines interface data from Oracle Integration Repository along with additional interfaces supported by Oracle E-Business Suite Adapter.

Browsing Interfaces by Expanding Tree Structure



The supported interfaces are 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
       |-Inbound
       |-Outbound
          |-Groups
            |-[business event name]
    |-Custom Objects
       |-PLSQL APIs
       |   |-[package_name]
       |-XMLGateway Maps
          |-Inbound
          |-Outbound
       |-Concurrent Programs
      |-[concurrent program name]
```

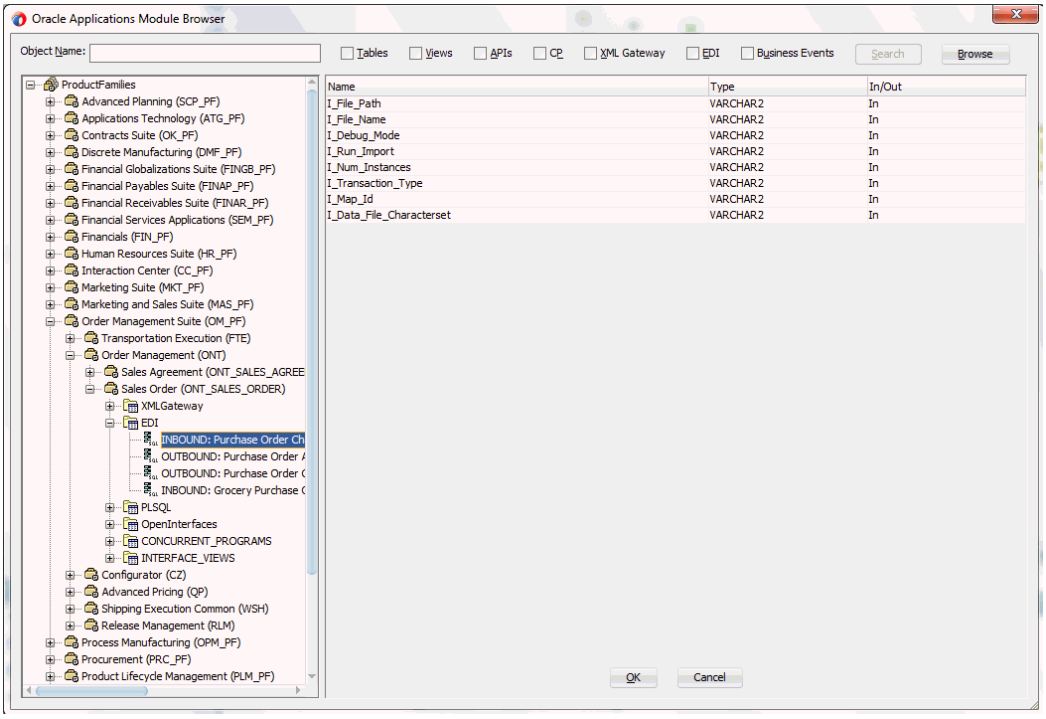
- The items under Other Interfaces, as well as certain PL/SQL APIs and concurrent programs under the *[product family]* hierarchy, are available through Oracle E-Business Suite Adapter, but not through Oracle Integration Repository.

- The number of interfaces indicated by `[n]` only appears in 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 release 12 instance.

Browsing Interfaces by Product Family

The Oracle Integration Repository interface data populates the ***[product_family]*** sections, grouped according to the products and business entities to which they belong.

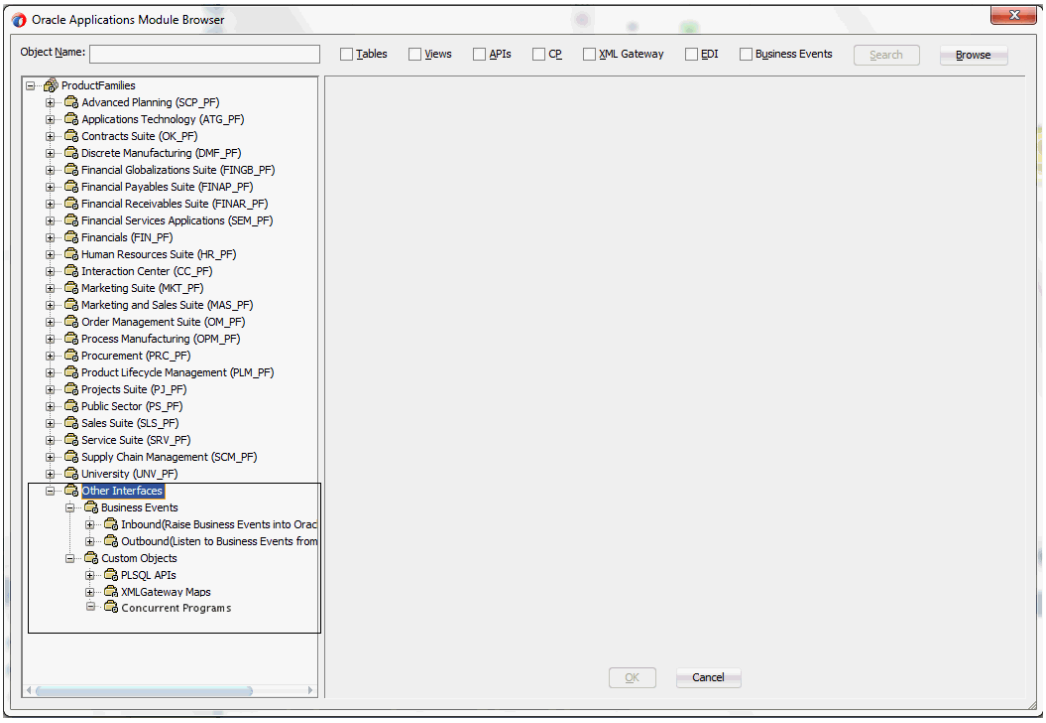
Browsing Interfaces by Product Family



Browsing Business Events and Custom Interfaces Under 'Other Interfaces'

Business events and custom interfaces are displayed under the Other Interfaces note.

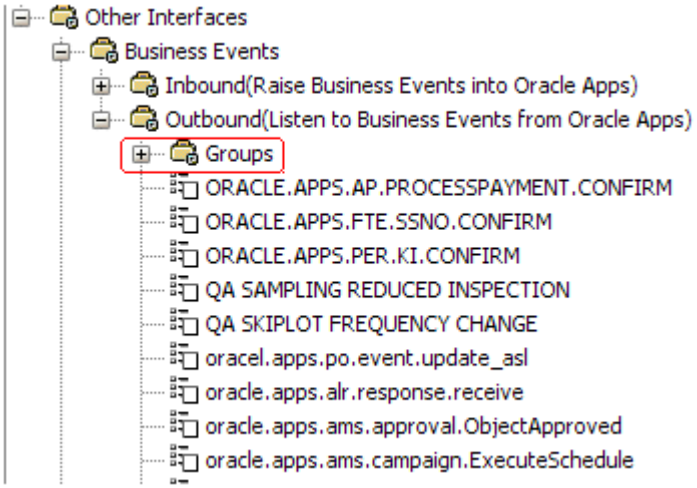
Browsing Other Interfaces



- ***Business Events and Business Event Groups***

  Business events and business event groups appear under the Other Interfaces node from the Oracle E-Business Suite Module Browser.

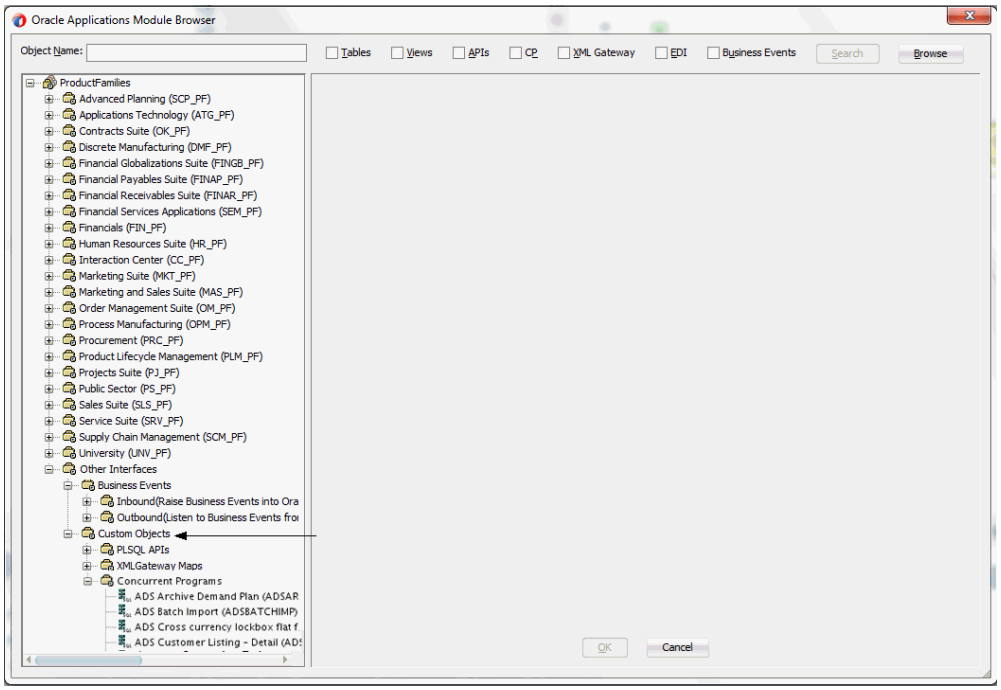
  A business event group is a type of event containing a set of individual business events. Clicking the **Other Interfaces > Business Events > Outbound > Groups** node displays the business event groups.



- *Custom Interfaces*

  Custom interfaces are displayed under the **Other Interfaces > Custom Objects** node from the Oracle E-Business Suite Module Browser.

Browsing Custom Interfaces



The supported custom interfaces are listed as follows:

- Custom PL/SQL APIs

  Custom 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**.

- Custom XML Gateway Maps

  Custom XML Gateway maps are categorized as either Inbound or Outbound. Select either one of the nodes to display custom XML Gateway maps.

- Customized Concurrent Programs

  Clicking the Concurrent Programs node under the Custom Objects node displays all the custom concurrent programs.

Understanding Oracle E-Business Suite Adapter Security

Security is the most critical feature that is designed to guard application content from unauthorized access. By leveraging Oracle User Management security mechanism, Oracle E-Business Suite Adapter 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 E-Business Suite. This protects application programming interfaces (APIs) from unauthorized access or execution without security checks.

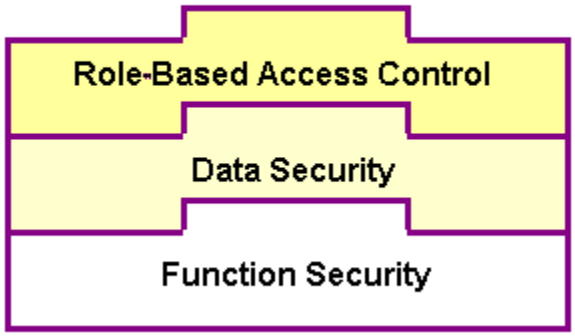
Function Security with Role-Based Access Control (RBAC)

Function security is the basic access control in Oracle E-Business Suite. 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 E-Business Suite, when having the function security layer enforced on the access to an API, it actually implicitly restricts the data access to the Oracle E-Business Suite application.

Building on function security, data security provides another layer of security control to model or enforce security authorizations of specific data records. In other words, data security further refines the security of accessing application records down to the data level.

To provide granular security control and allow only appropriate users with right privileges to execute APIs, Oracle E-Business Suite Adapter leverages Oracle User Management Role-Based Access Control security (RBAC) to reinforce the function security through user roles. Whether a user can access an API is determined by the roles granted to the user. This approach builds upon data security and function security, but it goes beyond both of them.

Function Security with Role-Based Access Control



To better understand how the security works for Oracle E-Business Suite Adapter, the

following topics are included in this section:

- Function Security Support for Overloaded Functions, page 4-21
- Function Security Support through Profile Option, page 4-22

Function Security Support for Overloaded Functions

Oracle E-Business Suite Adapter provides a way to handle access control for overloaded PL/SQL APIs. This feature works in conjunction with the grants created from the Integration Repository user interface, and it does not depend on any profile options.

> **Note:** This security support would work with all interfaces which are available in the Integration Repository. For other interfaces, you need to enable the function security through the profile option mentioned in the section Function Security Support through Profile Option, page 4-22.

For this purpose, two new JCA properties are introduced:

- `DataSecurityCheck`
- `IRepOverloadSeq`

If a user wants the Function Security Authorization check to be performed, the following property information needs to be added in the WSDL file (`XX_apps.jca`):

```
<property name="DataSecurityCheck" value="yes"/>
```

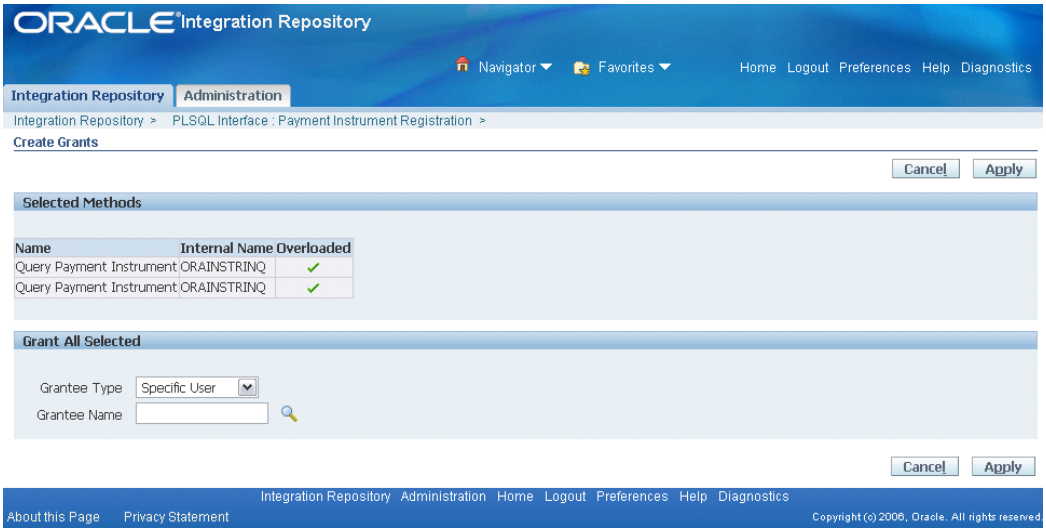
However, the other property `IRepOverloadSeq` is derived automatically by Oracle E-Business Suite Adapter at the design time during creation of the partner link. Based on these two properties, the function security check would be done for the username, which is passed as a header property.

Creating Security Grants through Integration Repository

This type of data security check for overloaded functions works in conjunction with security grants created through the Integration Repository user interface. This security grant can be performed in the Create Grant page for a given interface type to control the method at a very granular level.

An integration repository administrator can select one or more methods in a PL/SQL API and then authorize a user, user group, or all users to execute the selected method(s) by creating appropriate security grants.

Creating Grants in Integration Repository



In the Integration Repository user interface, each overloaded function contained in an interface can be uniquely granted to a specific user, user group, or all users through the create grant feature. If you select more than one overloaded function in the Procedures and Functions region (or the Methods region), in the Create Grant page an Overloaded column appears in the selected methods table indicating more than one overloaded function is selected for the grant.

For more information on how to create security grants using the Integration Repository user interface, see Managing Security Grants, Administering Native Integration Interfaces and Services chapter, *Oracle E-Business Suite Integrated SOA Gateway Implementation Guide* for details.

Function Security Support through Profile Option

Based on the function security with Role-Based Access Control (RBAC), security access control is defined 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 Oracle E-Business Suite Adapter, all annotated APIs that reside 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, Oracle E-Business Suite Adapter 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.

Enabling Function Security

Oracle E-Business Suite Adapter 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.

> **Note:** Using "DataSecurityCheck" described in the earlier section for overloaded functions is the preferred way for interfaces listed in the Integration Repository because it is easy to create the security grants and doesn't depend on any profile options.

- 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.

When a user tries to invoke an API within the Oracle E-Business Suite through a BPEL process, if the function security profile is enabled, the invocation of the API would happen only after validation of the authorization 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. 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 about this feature, see My Oracle Support Knowledge Document 787637.1 for details. For more information on Function Security and RBAC security models, see "Function Security and Role-Based Access Control (RBAC) in Oracle E-Business Suite", My Oracle Support Knowledge Document 1537100.1 and *Oracle E-Business Suite System Administrator's Guide - Security* for details.

Creating Security Grants through User Roles

To secure the API invocation only to a user with appropriate execution privileges, the following steps need to be performed on Oracle E-Business Suite to create security grants for the user using user roles:

1. Creating a Permission Set, page 4-24
2. Creating a User Role, page 4-26
3. Granting a Permission Set to a User Through a Role, page 4-27

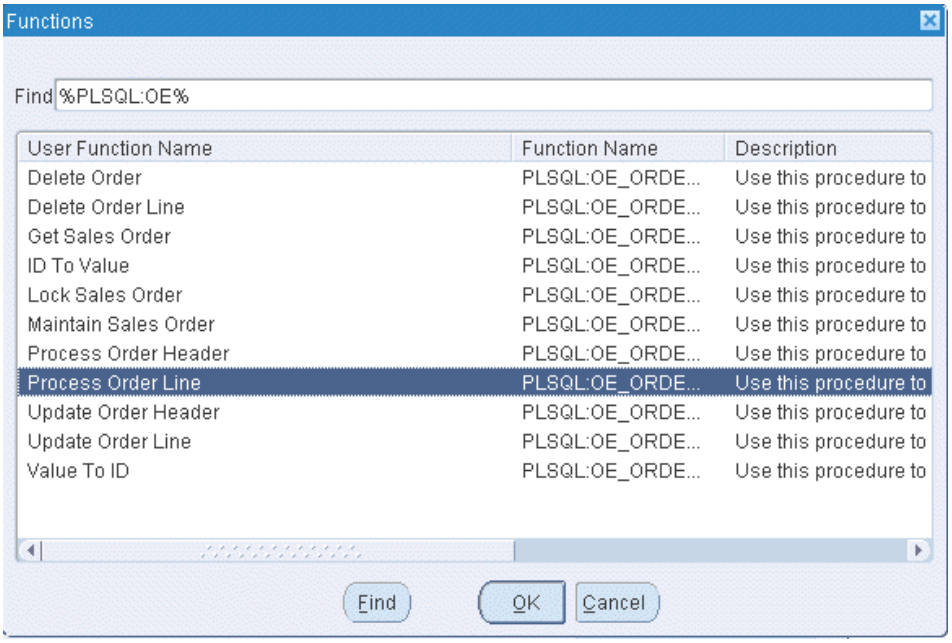
Creating a Permission Set

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

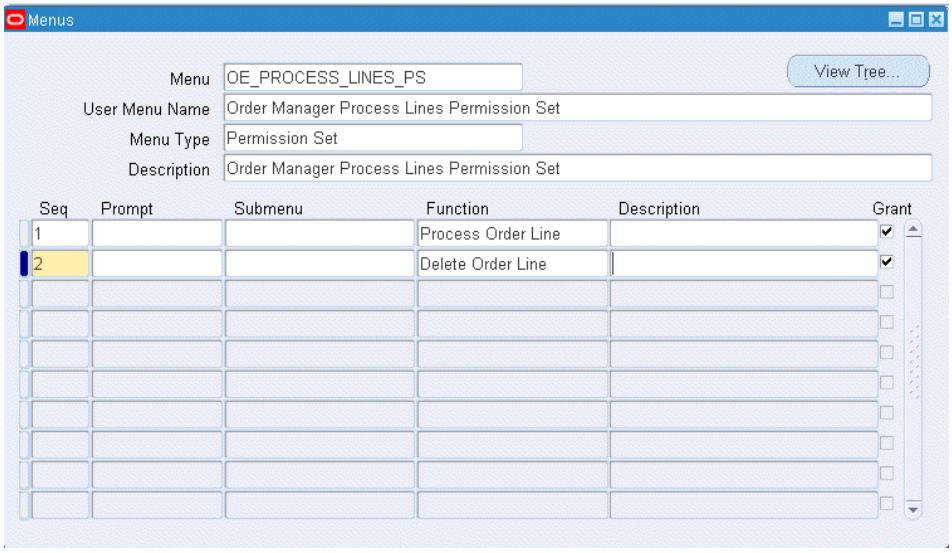


Based on your Composite application, select appropriate functions and then grant permissions to the API that you will be invoking from the BPEL process or Mediator.

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 4-26.

Permission Set Menu



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 4-24 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.

Flexfield Support for PL/SQL APIs and Open Interface Tables

Oracle E-Business Suite Adapter provides flexfield support for PL/SQL APIs and Open Interface Tables (OIT). If a PL/SQL API or an OIT table is configured with flexfields, flexfield information including flexfield data and mapping will not only be displayed at design time, but also be included in the XSD. This helps integration developers to create mappings between third party data elements with more meaningful business object elements. Additionally, this feature allows Oracle E-Business Suite Adapter run time to

work in the context of flexfields.

> **Note:** This feature is available for PL/SQL APIs and Open Interface Tables only and is for Oracle E-Business Suite Release 12 and above.

What is a Flexfield?

Flexfield is one of the essential features in Oracle E-Business Suite. It provides a flexible way to represent objects or to implement context-sensitive fields that appear only when needed. It is a field made up of sub-fields, or segments. Two flexfield types (key and descriptive flexfields) are used to let you customize Oracle E-Business Suite features without additional programming. To provide the unique, customized information that conforms to your business needs, Oracle E-Business Suite Adapter provides flexfield support for PL/SQL APIs and Open Interface Tables that contain flexfields.

How Flexfield Data is Configured at Design Time

At design time during the partner link creation, if a PL/SQL API or an open interface table is configured with flexfields, appropriate flexfield information is displayed along with the API parameters or table columns in the right pane of the Oracle E-Business Suite Module Browser. You can optionally modify or create a new flexfield mapping for the selected interface, or proceed with the partner link creation without configuring the interface with flexfields.

To configure flexfield data, Oracle E-Business Suite Adapter allows you to either use an existing flexfield mapping that has been created earlier, or to create a new mapping if desired. In the case of creating a new mapping, a mapping flexfield wizard appears guiding you through each configuration page where you can add key and descriptive flexfields with desired mapping for the selected interface.

Key Features

Flexfield feature provided in Oracle E-Business Suite Adapter includes the following key features:

- It displays flexfield information which contains the mapping of API parameter or interface table column names and the corresponding flexfield segment names in the Oracle E-Business Suite Module Browser.

- It lets you configure and modify both key flexfields and descriptive flexfields.

- It provides a flexible mapping support mechanism for a selected API or an open interface table that you can configure a new mapping or reuse a previously created mapping through import.

  > **Note:** A flexfield can be mapped only once. Duplicate flexfields are not allowed.

- It allows you to use more than one open interface table during the partner link

creation. Flexfields can be configured individually for each table.

- Once a flexfield mapping is completed and partner link is created, an enhanced schema (XSD) is generated using flexfield names instead of the generic parameter or column names.

- It allows the Oracle E-Business Suite Adapter run time to work in conjunction with the support from Oracle Database Adapter to handle the enhanced schema.

Understanding Key Elements in Flexfield Configuration

Oracle E-Business Suite Adapter supports both key flexfields and descriptive flexfields referenced by an API or open interface table.

- **Key Flexfields**

  Use key flexfields as identifiers for entities. Generally, the identifier you create using a key flexfield is required by the application. For example, an Accounting Flexfield is used to create and display account numbers. This key flexfield is owned by Oracle General Ledger, but its values can be used by many of the applications.

  A key flexfield structure usually consists of multiple segments and each segment contains meaningful information. For example, the Accounting Flexfield structure can consist of five segments such as Company, Department, Account, Sub-Account, and Product.

- **Descriptive Flexfields**

  Use descriptive flexfields to gather additional specialized, important, and unique information required by your business. In other words, this type of flexfield is used to collect information beyond what is collected by Oracle E-Business Suite.

  A descriptive flexfield typically uses multiple structures. Each of these structures can have different segments to gather different data. Each segment's value is stored in a column in one of the application's base tables. The column name reflects the type of flexfield data it holds.

In general, key flexfields store their data in columns called SEGMENTn; descriptive flexfields store their data in columns called ATTRIBUTEn, where n is a number. Use key flexfields to define your own structure for many of the identifiers required by Oracle E-Business Suite; use descriptive flexfields to gather additional information about your business entities beyond the information required by Oracle E-Business Suite.

For more information about flexfields, see the *Oracle E-Business Suite Flexfields Guide*.

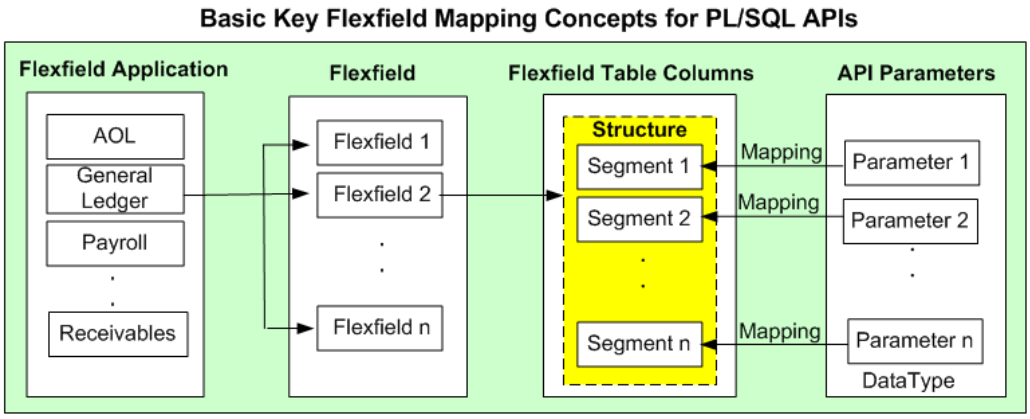
Flexfield Mapping Concepts

Each flexfield regardless of its type has its ***owner application***. For example, an Accounting Flexfield is owned by Oracle General Ledger application. During the flexfield configuration, you need to first select an application to which a flexfield

belongs, and then select a ***flexfield name*** (such as Accounting Flexfield) contained in the selected application (such as General Ledger).

Key Flexfield Mapping for PL/SQL APIs

The following diagram illustrates the high level key flexfield mapping concepts for PL/SQL APIs:



In this diagram, Flexfield 2 (Accounting Flexfield) belongs to General Ledger application, and it contains multiple segments (segment 1 to segment n) which can be grouped by data type.

The main part of flexfield configuration is ***mapping***. Once a flexfield with a desired application is selected, the matching segments (segment 1 to segment n) are derived from the Oracle E-Business Suite database along with the related API parameters based on the flexfield metadata for your mapping.

For PL/SQL APIs, parameters are displayed based on the selected ***data type*** (parent type or record type). API parameters and the flexfield table columns can be mapped either manually or automatically by using Auto Map.

- **Parent type:** Parent type is a pseudo type to allow all the scalar parameters at the procedure level to participate in the mapping. It can be used for simple APIs that just have scalar parameters.

- **Record type:** Record type can be selected if it's for a complex API that has record types in addition to scalar types. When a record type occurs multiple times in the API, the list of paths is displayed for reference. If the parameters from a record type are mapped to the flexfield table columns, then the same mapping would be applicable to all the paths it occurs.

  Supported record types include array, nested tables, and associated arrays.

Note: Automapping feature maps all the table columns to the

parameters which have the column name as part of the parameter name. For examples, parameter P_ATTRIBUTE1 would be mapped to a column name called ATTRIBUTE1, and parameter P_ATTRIBUTE2 would be mapped to a column name called ATTRIBUTE2.

Once mapping is completed, you will need to select a flexfield **structure** for a key flexfield.

Note: A flexfield structure is a specific configuration of segments. If you add or remove segments, or rearrange the order of segments in a flexfield, you get a different structure. The segments that make up a particular structure are logically or functionally related. The structure and segment definition is created in Oracle E-Business Suite through the Application Developer responsibility.

For more information about flexfields, see the *Oracle E-Business Suite Flexfields Guide*.

Key Flexfield Mapping for Open Interface Tables

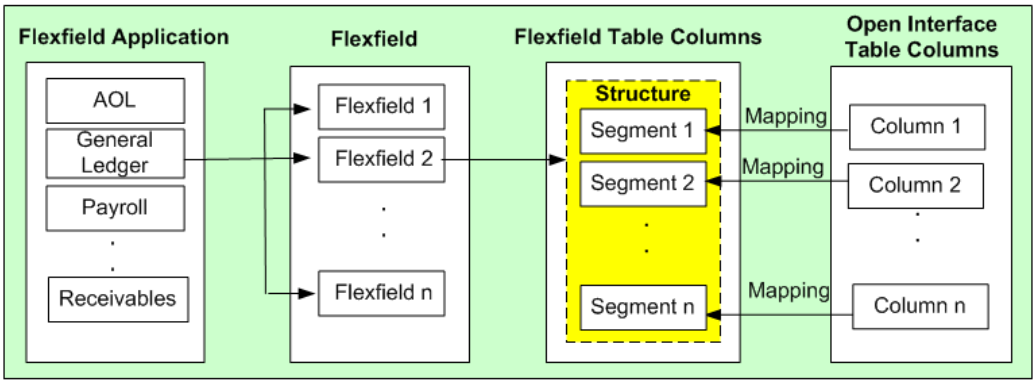
Similar to the mapping for PL/SQL APIs, once a flexfield with a desired application is selected for an Open Interface Table, the matching segments (segment 1 to segment n) are derived from Oracle E-Business Suite database along with the related open interface table columns based on the flexfield metadata for your mapping. Mapping can be performed either manually or automatically by using Auto Map.

Note: Please note that datatype is used for PL/SQL APIs only.

Automapping feature maps all the flexfield table columns to the open interface table columns which have the flexfield table column name as part of the interface table column name. For examples, interface table column name TP_ATTRIBUTE1 would be mapped to a flexfield table column name called ATTRIBUTE1, and interface table column TP_ATTRIBUTE2 would be mapped to a flexfield table column name called ATTRIBUTE2.

The following diagram illustrates the high level key flexfield mapping concepts for Open Interface Tables:

Basic Key Flexfield Mapping Concepts for Open Interface Tables



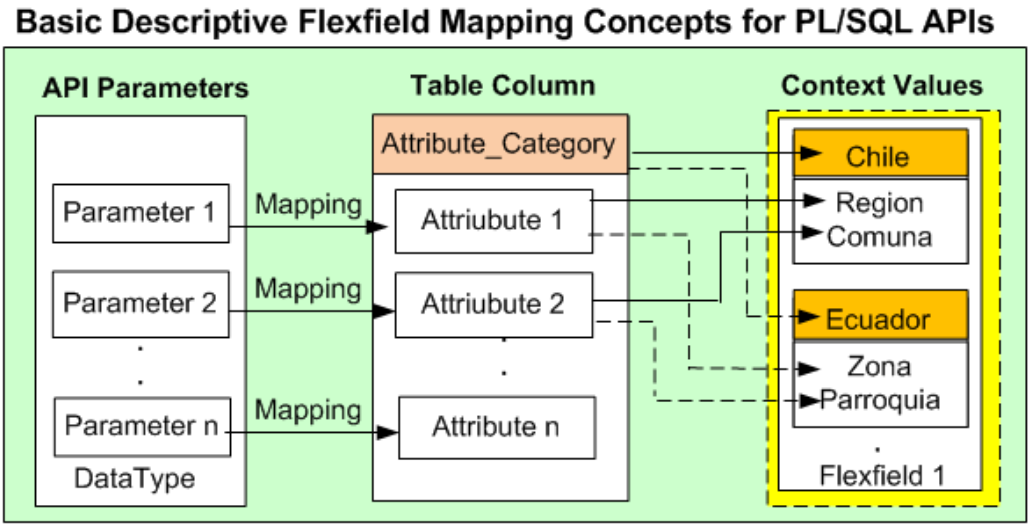
Once mapping is completed, you will need to select a flexfield **structure** for a key flexfield.

Descriptive Flexfield Mapping for PL/SQL APIs

For descriptive flexfields, an additional mapping is required for the **context column**. All the flexfield table columns including the context column should be mapped to the scalar parameters within one record type or parent type. Once mapping is completed, select at least one context value for the selected context column for a descriptive flexfield.

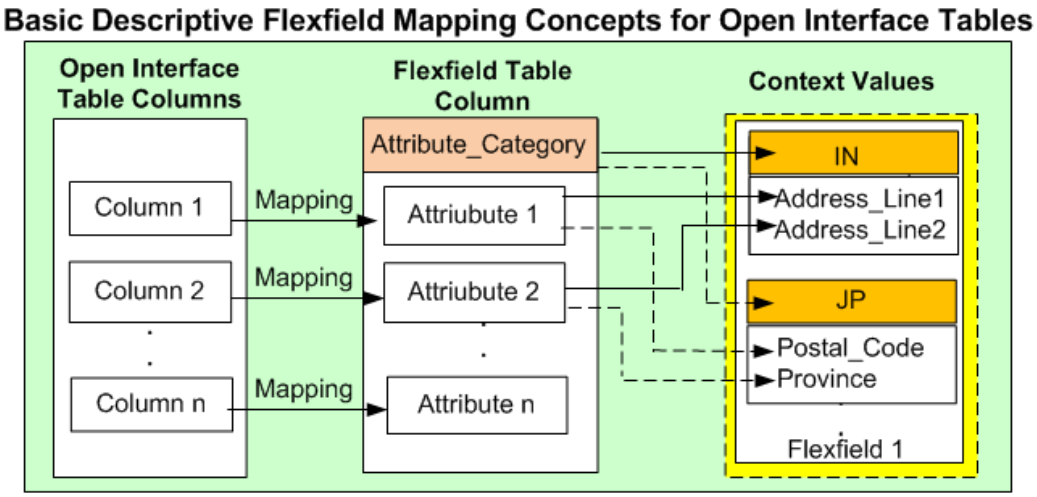
For example, Chile uses 'Region' and 'Comuna' in addition to 'State' and 'County' to represent subdivision or relatively small location geographically. Similarly, Ecuador uses 'Zona' and 'Parroquia' in geography to describe different boundaries. This is a perfect example using a descriptive flexfield to customize the application to meet your business needs. Chile and Ecuador are the context values when defining the descriptive flexfield 'TCA Location Information'. `Parameter1` (or `Parameter2`) is mapped to `ATTRIBUTE1` (or `ATTRIBUTE2`) whose value depends on the selected context value (either 'Chile' or 'Ecuador').

> **Note:** For descriptive flexfields, context value should be available to the runtime through an API parameter.



Descriptive Flexfield Mapping for Open Interface Tables

Similar to PL/SQL APIs, an additional mapping is also required for the **context column** in defining descriptive flexfield mapping for Open Interface Tables. All the flexfield table columns including the context column should be mapped to the open interface table columns. Once mapping is completed, select at least one context value for the selected context column for a descriptive flexfield.



In this diagram, for example, 'Address_Line1' and 'Address_Line2' are selected for 'IN' (India) to describe the address information, while the same attributes (`ATTRIBUTE1` and `ATTRIBUTE2`) are mapped to 'Postal_Code' and 'Province' for 'JP' (Japan).

Oracle E-Business Suite Adapter uses the flexfield mapping defined from the Oracle E-Business Suite Module Browser to generate the enhanced XSD. This XSD will contain flexfield segment names based on the mapping chosen at design time. Oracle E-Business Suite Adapter will then work in conjunction with Oracle Database Adapter

to handle the enhanced schema file for runtime processing.

Guidelines for Defining Flexfield Structure and Context Values

You will not be able to select a context value for a descriptive flexfield or a structure for a key flexfield if one of the following conditions are met:

- A context/structure has no segments or segment mappings.

  Only context/structure values that have at least one segment can be selected.

- A context/structure has segments, but none of them matches with at least one of the table columns mapped in the mapping page.

- A context/structure has all the segments disabled in Oracle E-Business Suite.

- A context/structure itself is disabled in Oracle E-Business Suite.

  Only context/structure that is enabled will be shown in the Flexfield Subflow wizard in step 6 (Flexfield Subflow - Configure Flexfields).

Flexfield mapping will be used to generate an enhanced schema or XSD file.

Flexfield Mapping Configuration Guidelines and Constraints

- Duplicate flexfields are not allowed. A flexfield can be mapped only once.

- Range key flexfields are not supported.

- Valuesets and qualifiers are not supported.

- For PL/SQL APIs, flexfield mapping for all the flexfield segments in one flexfield should happen either at the parent level or within one record type.

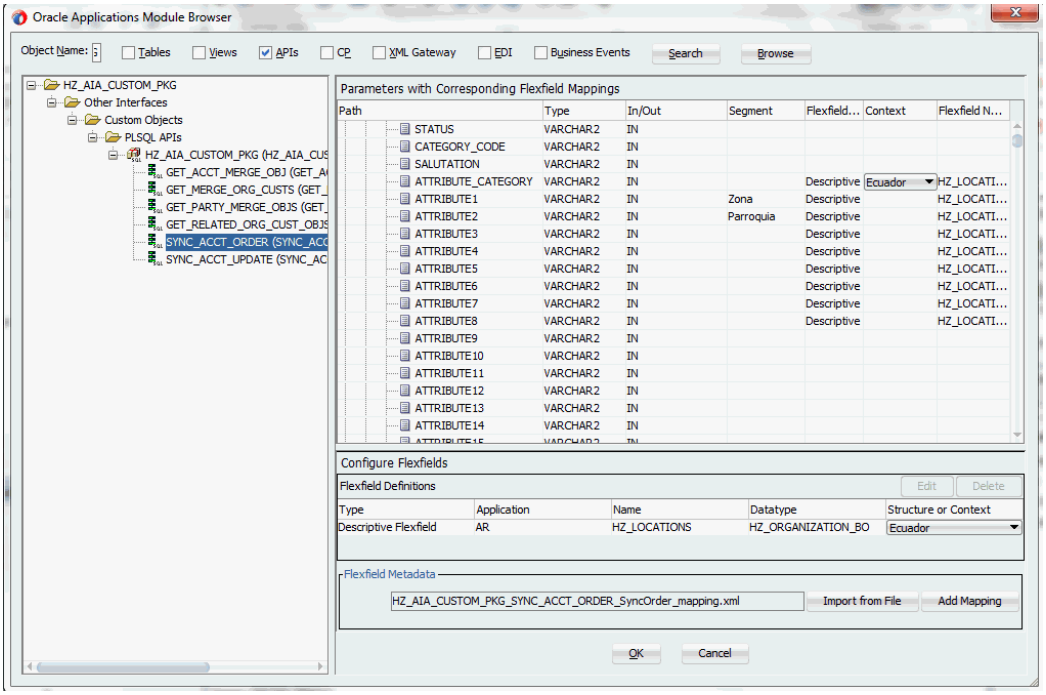
- For PL/SQL APIs, the mapping for the context column should correspond to parameters at the level at which the segments are mapped. In other words, if the descriptive flexfield segments are mapped to a selected data type (either record type or parent type) for PL/SQL APIs, the context column mapping should also happen with the same selected data type.

Accessing Flexfield Mapping User Interface

While creating a partner link at design time, if a selected interface is configured with flexfields, Oracle E-Business Suite Module Browser will display flexfield information in the right pane of the browser window.

For example, a PL/SQL API SYNC_ACCT_ORDER can be retrieved from a search or selected from HZ_AIA_CUSTOM_PKG listed under the **Other Interfaces > Custom Objects > PLSQL APIs** node. The flexfield data if configured in this API is displayed in the Oracle E-Business Suite Module Browser.

API with Flexfield Mapping Information



> **Note:** If a selected API or an interface table has pre-configured flexfield mappings available, a pop-up window appears allowing you to decide if you want to use these pre-configured mappings. Click **Yes** to display these mappings as part of the flexfield definitions in the Configure Flexfields region. You can modify them later if needed.
>
> Click **No** to indicate that you will not use these mappings and the pop-up window disappears.

Understanding Flexfield Mapping Data in Oracle E-Business Suite Module Browser

The flexfield information associated with the selected interface can be populated and configured in the following regions of the Oracle E-Business Suite Module Browser:

> **Note:** Alternatively, you can bypass the flexfield mapping configuration or modification for your selected interface by simply clicking **OK** in the browser to proceed with the partner link creation.

- **Parameters (or Interface Table Columns) with Corresponding Flexfield Mappings Region**

  This region displays the parameters or interface table columns of the selected interface along with the corresponding flexfield mapping information.

For PL/SQL APIs, record type parameters are displayed as a tree structure within the table, allowing you to expand or collapse the tree nodes to locate your desired parameters. For Open Interface Tables, interface table column names are simply listed in the table.

- **Configure Flexfields Region**

  This region allows you to perform the following tasks:

  - *Modifying Flexfield Mapping Definitions*

    No matter if the selected interface is a custom or seeded one, once the flexfield mapping information is populated in the browser, you can modify context values for a descriptive flexfield and structure for a key flexfield.

    Oracle E-Business Suite Adapter will then use the flexfield definitions from the Module Browser to generate the enhanced XSD. This XSD will contain the flexfield segment names chosen at design time based on the mapping information along with the parameter or table column names.

    For more information on how to modify your flexfield mappings, see Modifying Flexfield Mapping Definitions, page 4-119.

  - *Creating a New Mapping*

    You can create a new mapping for a selected interface by clicking **Add Mapping** in the Flexfield Metadata section. A new flexfield mapping wizard appears guiding you through each configuration page where you can add key and descriptive flexfields, as well as configure flexfield mapping between the selected interface and any flexfields defined in the Oracle E-Business Suite instance.

    See: Adding or Configuring a New Custom Mapping, page 4-37.

  - *Importing an Existing Mapping*

    Instead of creating a new mapping, you can use an existing mapping that has been created earlier by clicking **Import from File** in the Flexfield Metadata section. This lets you select a desired flexfield mapping for your selected interface.

    Once a desired mapping is imported, you can modify the context values for a descriptive flexfield and structure for a key flexfield if needed to meet your needs.

    See: Importing an Existing Flexfield Mapping, page 4-115.

Reviewing Flexfield Mapping Configuration Files

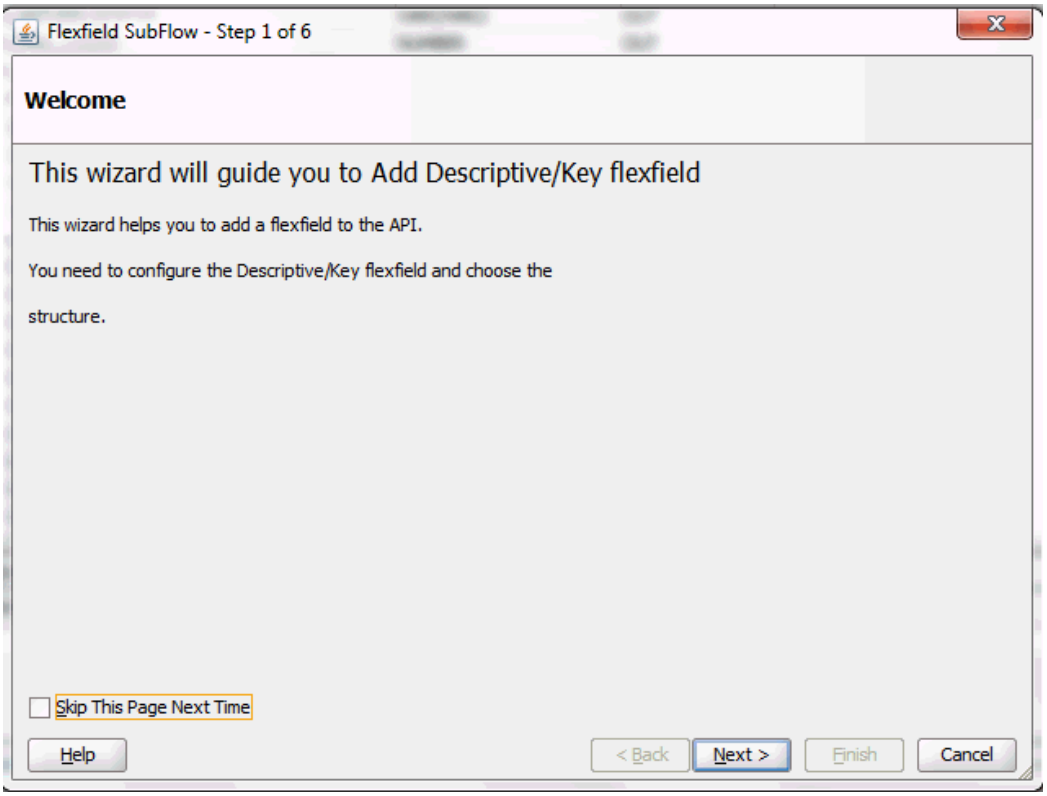
Once a partner link is created, an enhanced XSD file along with flexfield configuration and mapping files are generated. You can review these files for flexfield details. See: Reviewing Flexfield Mapping Configurations, page 4-122.

Adding or Configuring a New Mapping

After performing a search or browsing through the list of interfaces available in Oracle E-Business Suite Module Browser, if a selected PL/SQL interface or an open interface table is configured with flexfields, then flexfield information in the API or the table will be displayed in the right pane of the window.

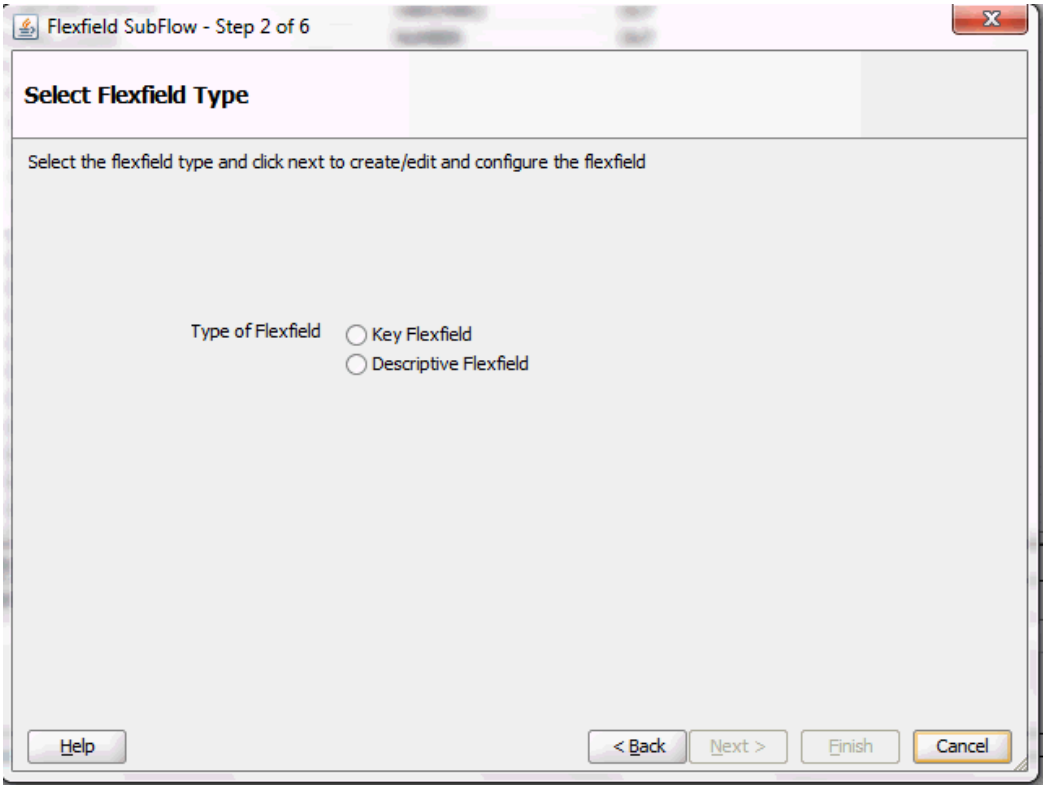
To add or configure a new mapping for the selected API or open interface table, click **Add Mapping** in the Configure Flexfields region. A series of flexfield selection pages in the wizard appear guiding you to configure a new mapping for your selected API or interface table.

Flexfield Subflow - Welcome



After clicking **Next** in the Welcome page, the Select Flexfield Type page appears where you can choose either a key or descriptive flexfield to be configured or added for your flexfield.

Flexfield Subflow - Select Flexfield Type



Select either the Key Flexfield or Descriptive Flexfield radio button for your flexfield configuration.

Please note that you can configure more than one key or descriptive flexfield for the selected API or open interface table to meet your business needs.

See:

- Configuring Key Flexfield Mappings for a PL/SQL API, page 4-38
- Configuring Descriptive Flexfield Mappings for a PL/SQL API, page 4-59
- Configuring Key Flexfield Mappings for an Open Interface Table, page 4-79
- Configuring Descriptive Flexfield Mappings for an Open Interface Table, page 4-85

Configuring Key Flexfield Mappings for a PL/SQL API

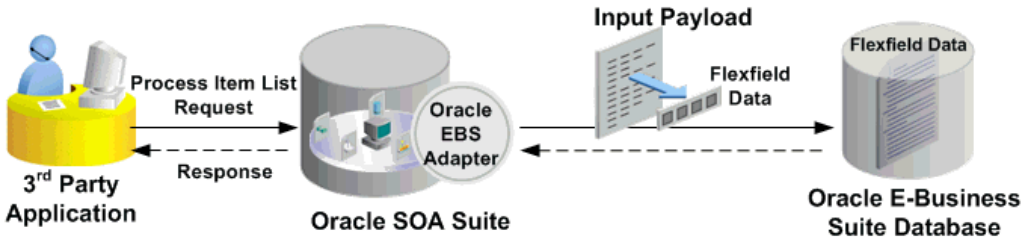
Sample Business Scenario

Take the `INV_EBI_ITEM_PUB.PROCESS_ITEM_LIST` API as an example to explain the inbound web service creation with flexfield data.

When a request is placed for processing inventory item list from a third party

application, the key flexfield data as part of the input payload is routed and the PL/SQL API `INV_EBI_ITEM_PUB.PROCESS_ITEM_LIST` is invoked to update the item list in Oracle E-Business Suite. The updated data will be passed to the client as the response message.

The following diagram illustrates the service invocation process flow:



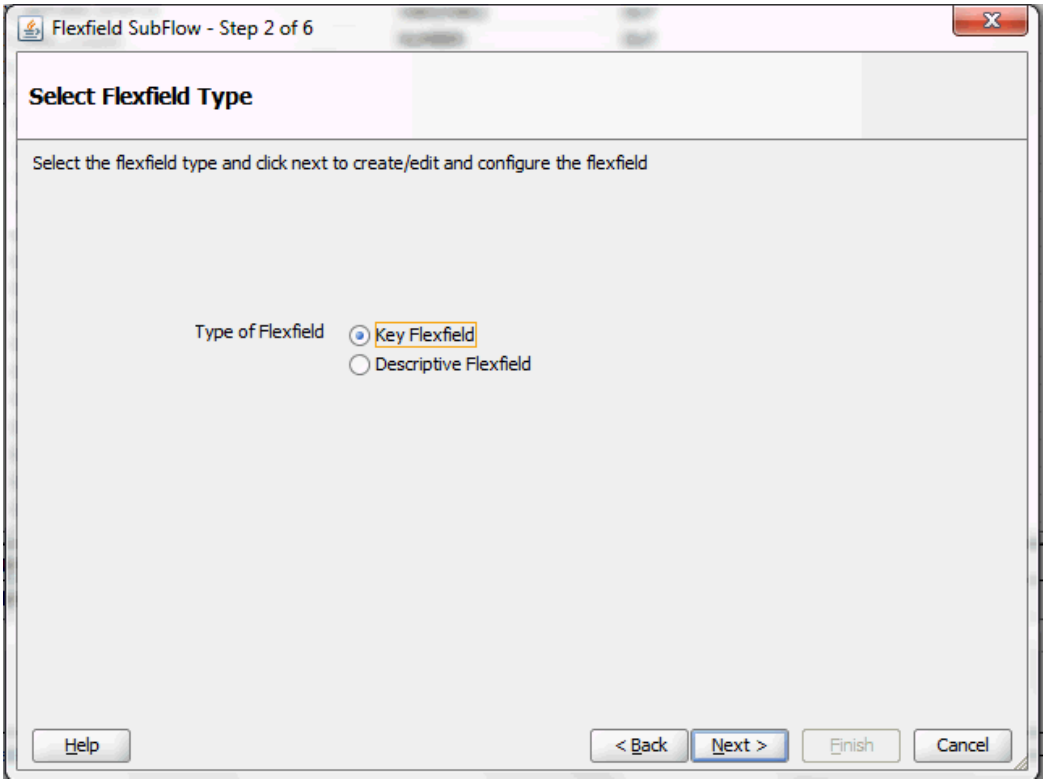
After deploying the service, you can validate the process by querying inventory item list directly from Oracle E-Business Suite application. The retrieved item data should be the same as input from the payload.

Configuring a Key Flexfield Mapping As Part of the Partner Link Creation

To configure a key flexfield mapping for the selected API `INV_EBI_ITEM_PUB.PROCESS_ITEM_LIST`, click **Add Mapping** in the Configure Flexfields region. A series of flexfield selection pages in the wizard appear guiding you to configure a new mapping for your selected API.

Click **Next** in the Welcome page. The Select Flexfield Type page is displayed.

Flexfield Subflow - Select Flexfield Type

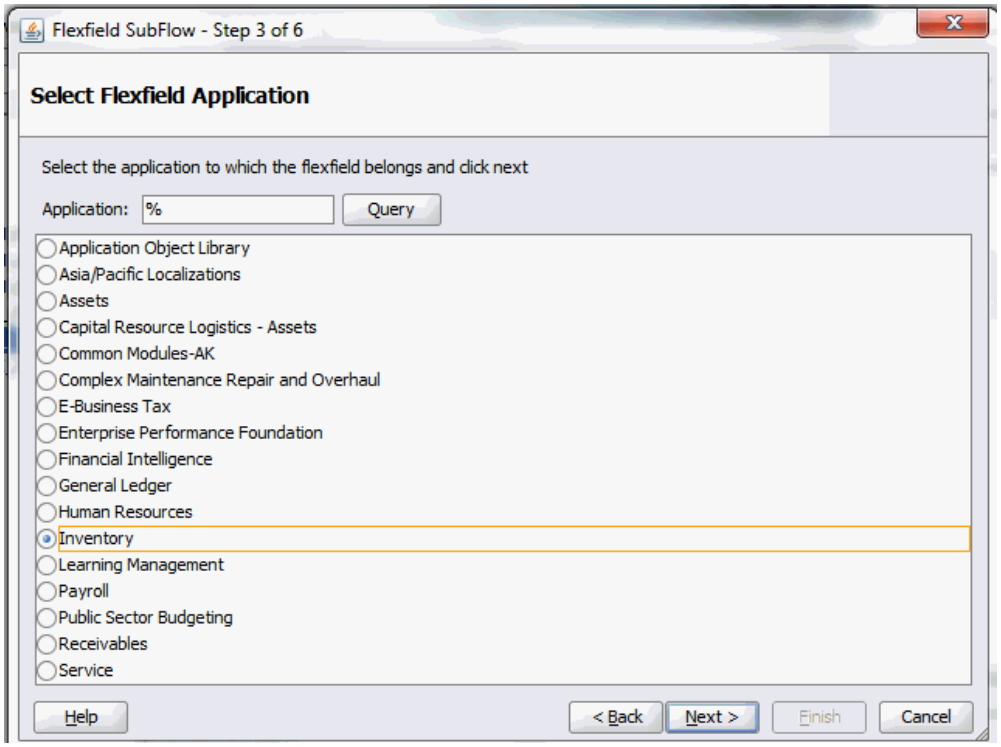


Perform the following steps to configure a key flexfield:

1. Select the **Key Flexfield** radio button in the Select Flexfield Type page. Click **Next**. The Select Flexfield Application page appears.

2. To locate your desired application name, enter keyword search (such as '%App%') and click **Query** to execute the search. All the entries that match your search criteria will be displayed. For example, select the **Inventory** radio button as the application name to which the flexfield belongs.

Flexfield Subflow - Select Flexfield Application

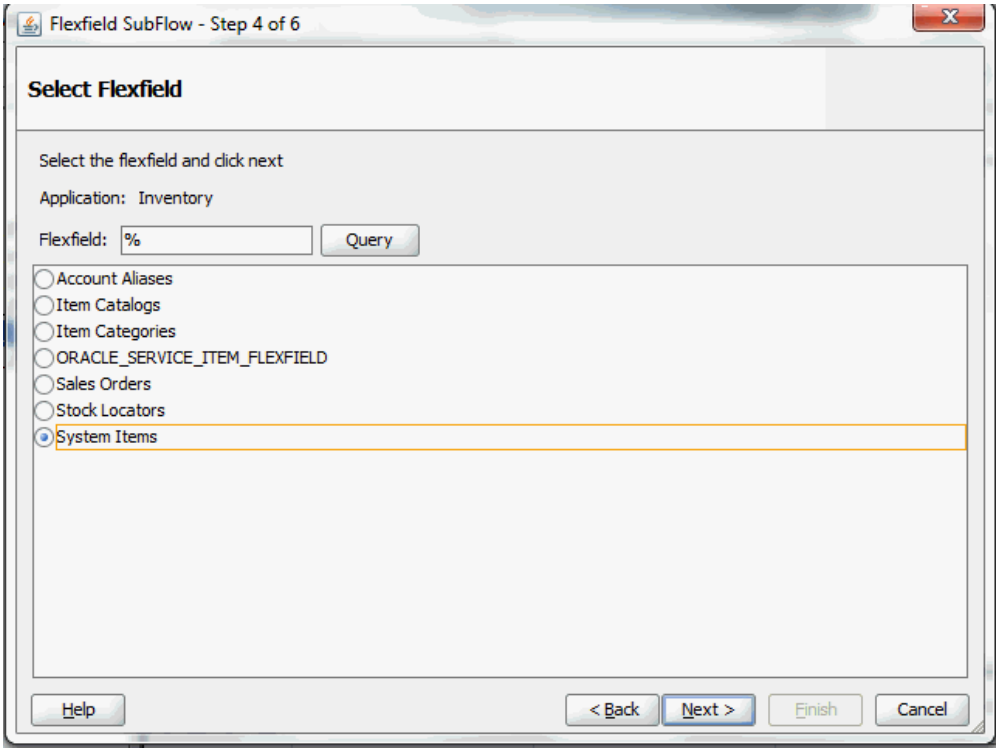


Alternatively, click **Query** directly to execute the blank search without keyword search. This displays all flexfield application names for your selection.

Click **Next**.

3. In the Select Flexfield page, enter keyword search if desired and click **Query** to execute the search. All the matched key flexfields are displayed for your selection. Select your desired key flexfield radio button.

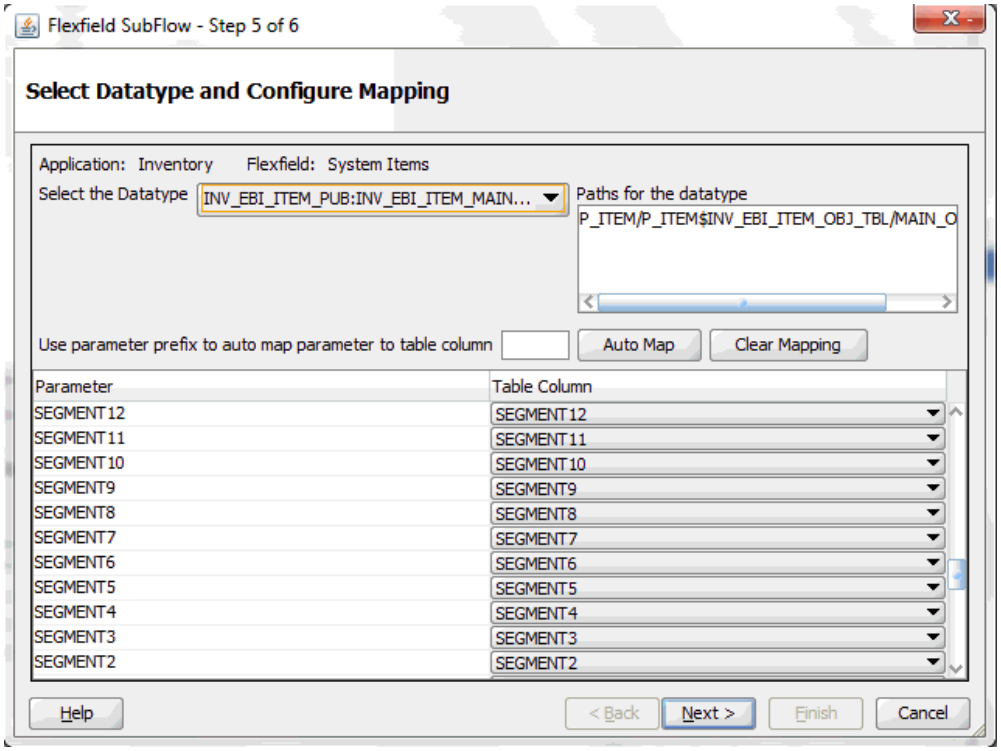
Flexfield Subflow - Select Flexfield



Click **Next** to continue.

4. The Select Datatype and Configure Mapping page appears where you can choose desired mapping for your key flexfield. Notice that your selected key flexfield name is automatically populated as the Flexfield name.

Flexfield Subflow - Select Datatype and Configure Mapping



To configure flexfield mapping, select the <Parent_type> from the Select the Datatype drop-down list if it's a simple API which does not have record types. This retrieves all needed flexfield metadata from Oracle E-Business Suite database and populates parameters at the parent level for your mapping.

> **Note:** Parent_type is a pseudo type to allow all the scalar parameters at the procedure level to participate in the mapping.
>
> You can also select record type if it's for a complex API. When a record type occurs multiple times in the API, the list of paths is displayed for reference. If the parameters from a record type are mapped to the flexfield table columns, then the same mapping would be applicable to all the paths it occurs.

In this mapping page, you have the option to do the mapping manually or use automapping feature.

- Automapping feature maps all the segments to the parameters, which have the column name as part of the parameter name. For example, table column SEGMENT1 would be mapped to parameter SEGMENT1. If there are multiple sets of parameters and the table column names, then the prefix can be specified.

For example, enter 'P_' as the prefix to retrieve all parameters starting with 'P_', such as P_SEGMENT1, P_SEGMENT2, P_SEGMENT3, to P_SEGMENTn where n is a number. If you click **Auto Map**, all the parameters with prefix 'P_' are automatically displayed along with the automapped table column names.

> **Note:** Flexfield values are stored in the application database tables. In general, key flexfields store their data in columns called SEGMENTn and descriptive flexfields store their data in columns called ATTRIBUTEn, where n is a number.

- Manual mapping lets you select a desired table column for a corresponding parameter.

> **Note:** If a parameter has already mapped to a table column, you cannot remap the parameter to other column. In this situation, 'Mapped to other Flexfield' appears instead for the mapped parameter.

Click **Clear Mapping** to clear current mapping you have. All previously mapped columns marked with 'Mapped to other Flexfield' will not be removed from the system.

Click **Next** to continue.

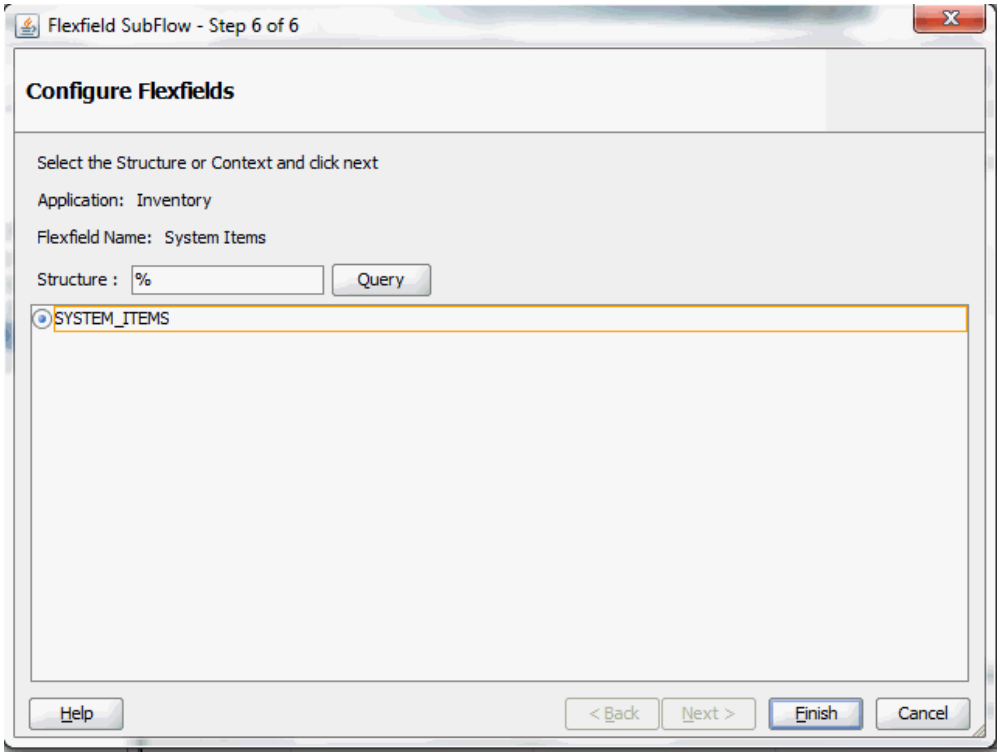
5. The configuration of the flexfield will not be complete without a structure for a key flexfield and at least one context value for a descriptive flexfield.

   In the Configure Flexfields page, enter keyword search if desired in the Structure field or simply click **Query** to execute a blank search. All the matched structures for the selected key flexfield are displayed.

   Select only one structure (such as SYSTEM_ITEMS).

   Please note that there are certain criteria need to be met in order to have your desired structures displayed in the wizard for selection. See: Guidelines for Defining Flexfield Structure and Context Values, page 4-34.

Flexfield Subflow - Configure Flexfields



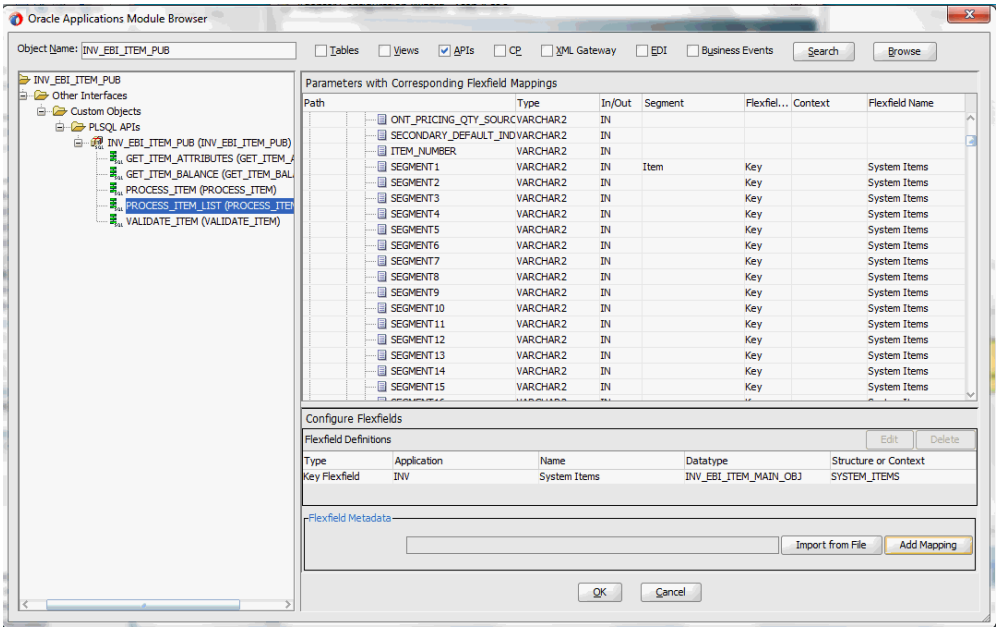
Note: Only one structure for a key flexfield can participate in flexfield mapping. In the case of a descriptive flexfield, multiple context values can participate in the mapping.

Click **Finish**.

6. The Oracle E-Business Suite Module Browser window appears. The newly added key flexfield structure and mapped parameters are displayed in the Parameters with Corresponding Flexfield Mappings region.

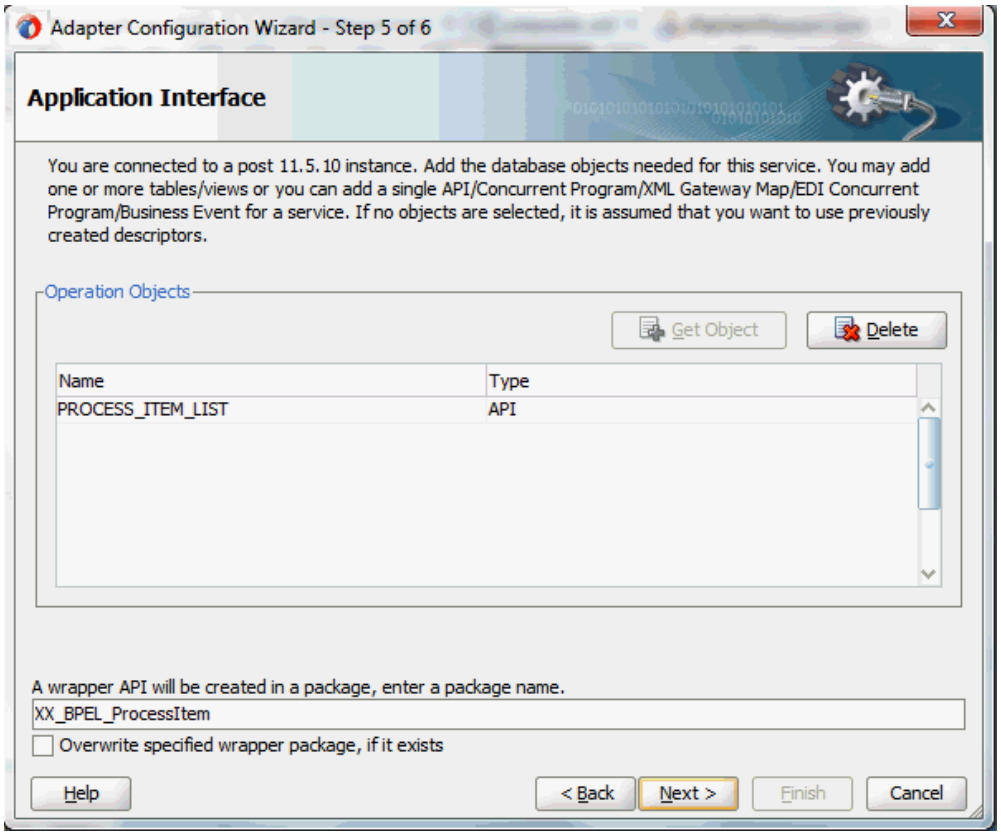
   The key flexfield information is also displayed in the Flexfield Definitions section.

Displaying Selected Key Flexfield Information



Click **OK**.

7. The Application Interface page appears with the selected API.



Click **Next** and then click **Finish**.

The wizard generates the WSDL file corresponding to the partner link. An enhanced XSD file, for example `ProcessItem_KFF_flex.xsd`, is also created. This XSD file contains the schema describing the procedure arguments with the elements representing the parameters that have been replaced with flexfield segment names.

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

8. Additionally, configuration file and flexfield mapping file are created. For more information on these files, see Reviewing Flexfield Mapping Configurations, page 4-122.

To add descriptive flexfield mappings for the selected API, see Configuring Descriptive Flexfield Mappings for a PL/SQL API, page 4-59.

Displaying Mapped Key Flexfields at Design Time

The key flexfield mapping data configured earlier during the partner link creation can be shown at design time.

A Composite Example with Key Flexfield Mappings

Here is an example describing the key flexfield mapping in the BPEL process of a SOA

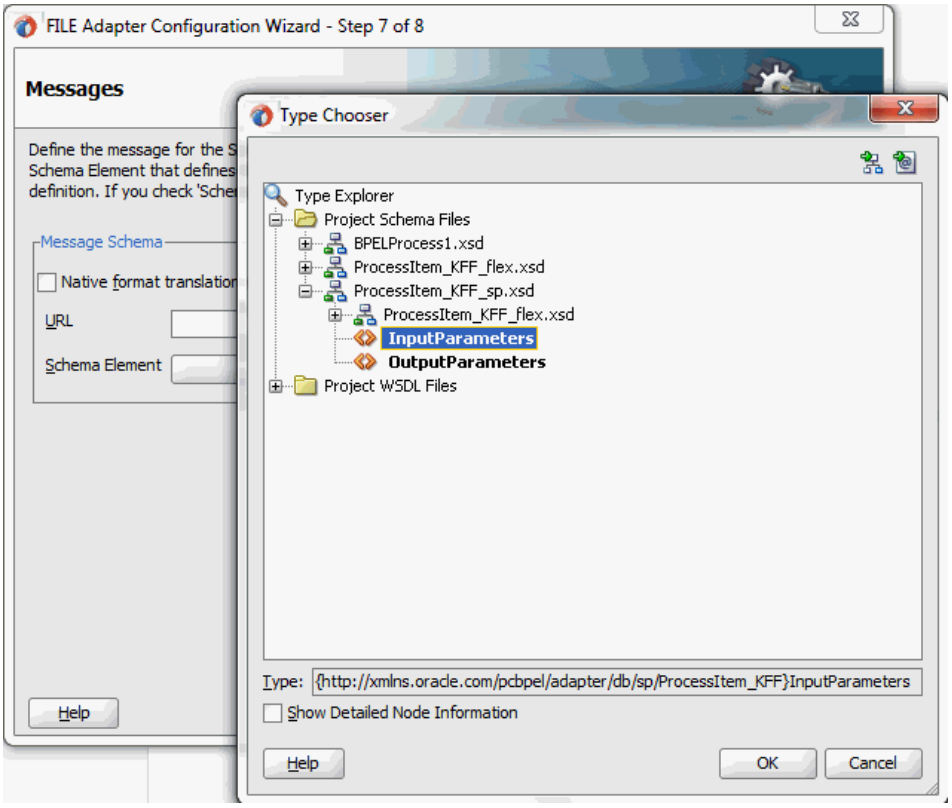
composite:

- Create a new SOA composite application with BPEL process
- Add a partner link service called `ProcesItem_KFF` with key flexfield mapping that we configured earlier for the `INV_EBI_ITEM_PUB.PROCESS_ITEM_LIST` API

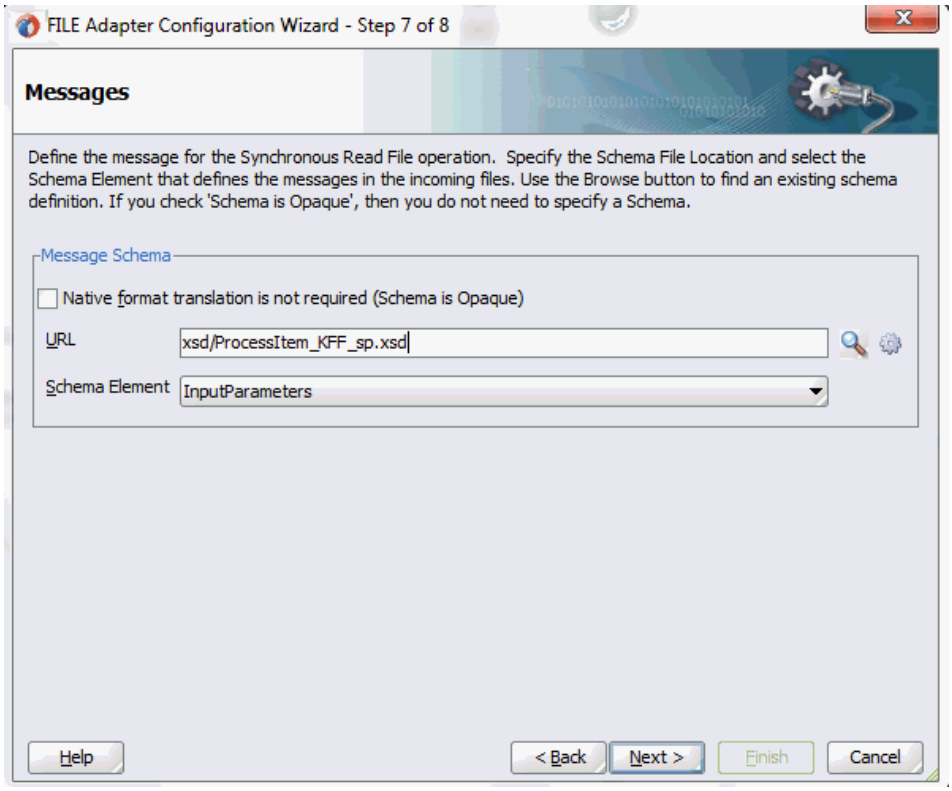
  For information on how to configure key flexfield mappings, see Configuring Key Flexfield Mappings for a PL/SQL API, page 4-38.

- Add a partner link for File Adapter to read input payload that contains flexfield data for service invocation
  - In the File Directories dialog, enter the physical directory (such as `/usr/tmp`) where the input payload xml file (such as `input_payload.xml`) resides.
  - In the Messages dialog, select the 'browse for schema file' icon next to the URL field to open the Type Chooser.

    Click Type Explorer and select **Project Schema Files > ProcessItem_KFF_sp.xsd > ProcessItem_KFF_flex.xsd > InputParameters**.



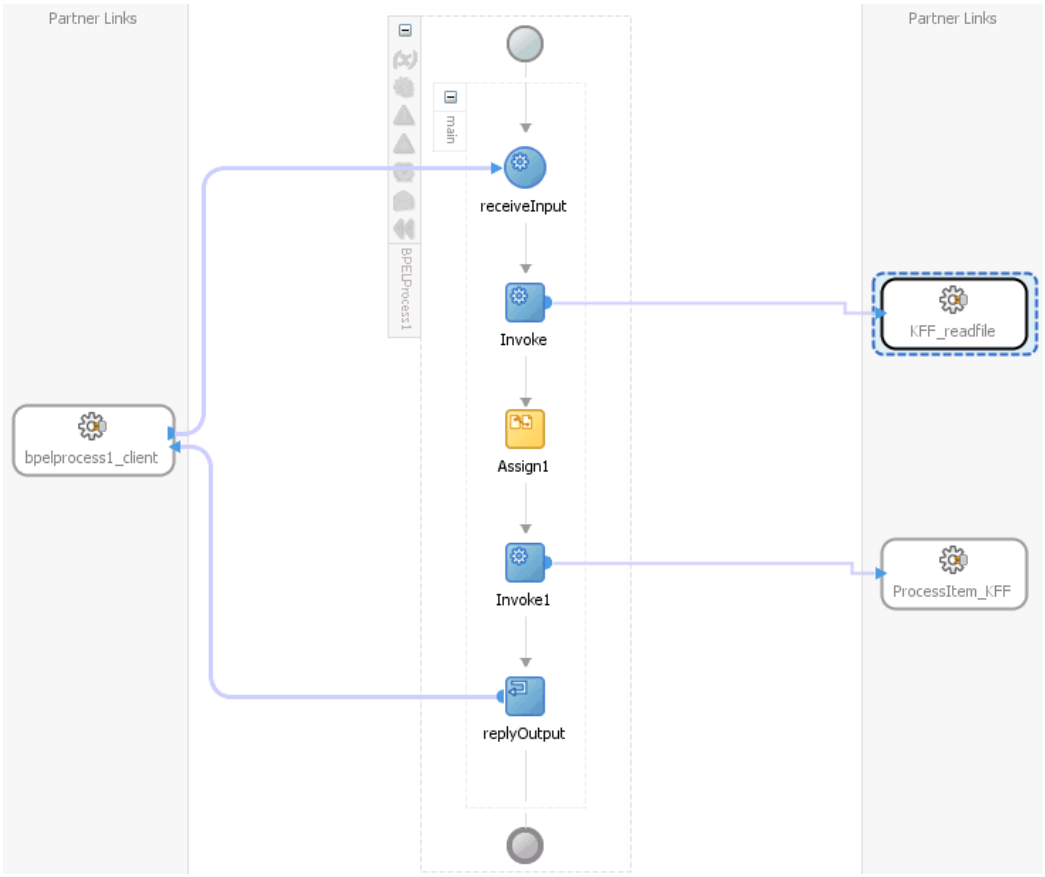
Click **OK**. The selected schema information will be automatically populated in the URL and Schema Element fields.



- Configure two **Invoke** activities
  - Associate the first **Invoke** activity with the File Adapter partner link to synchronously read input payload data
  - Associate the second **Invoke** activity with the partner link `ProcessItem_KFF` that contains flexfield mapping to invoke the service
- Add an **Assign** activity to pass input payload received from the File Adapter to the second **Invoke** activity for service invocation

  Mapped key flexfield column names corresponding to the flexfield matching segments defined in the Oracle E-Business Suite are displayed in the **Assign** activity. See: Assigning Parameters with Key Flexfields in the Assign Activity, page 4-54.

Constructing BPEL Process in SOA Composite

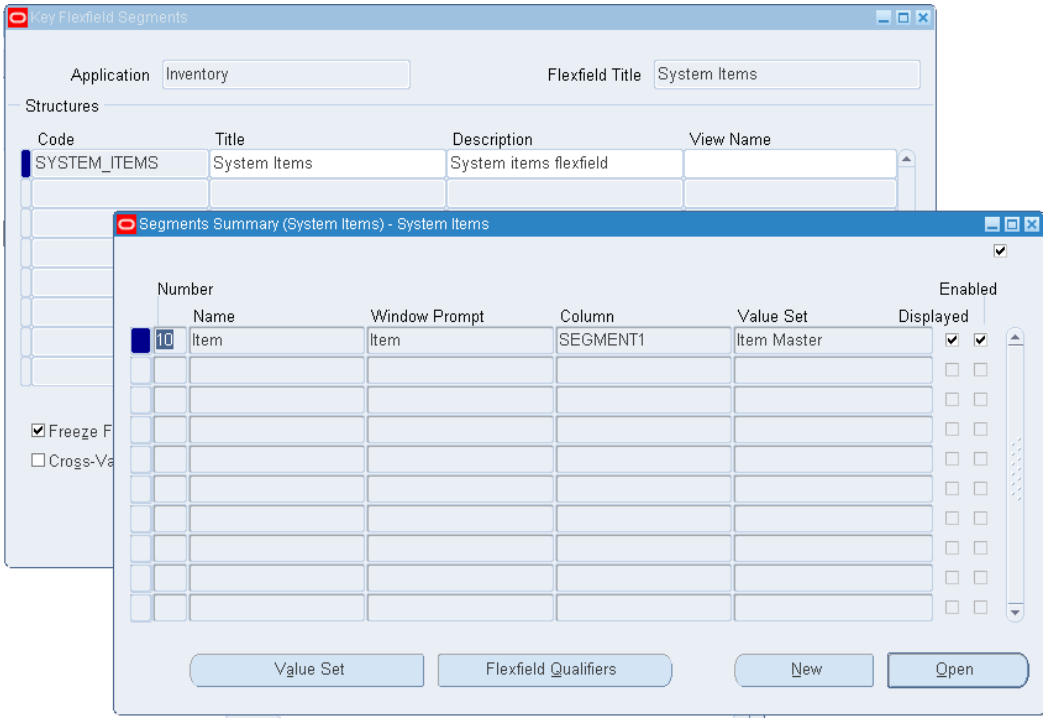


For more information on how to create design time tasks, see Design-Time Tasks for PL/SQL APIs, page 9-2.

Flexfield Validation in Oracle E-Business Suite

When defining key flexfield segments for a key flexfield 'System Items' in Oracle Inventory application (Application Developer responsibility), only SEGMENT1 is specified as 'Item' in the Segments Summary - System Items window.

Key Flexfield Segments Summary for System Items



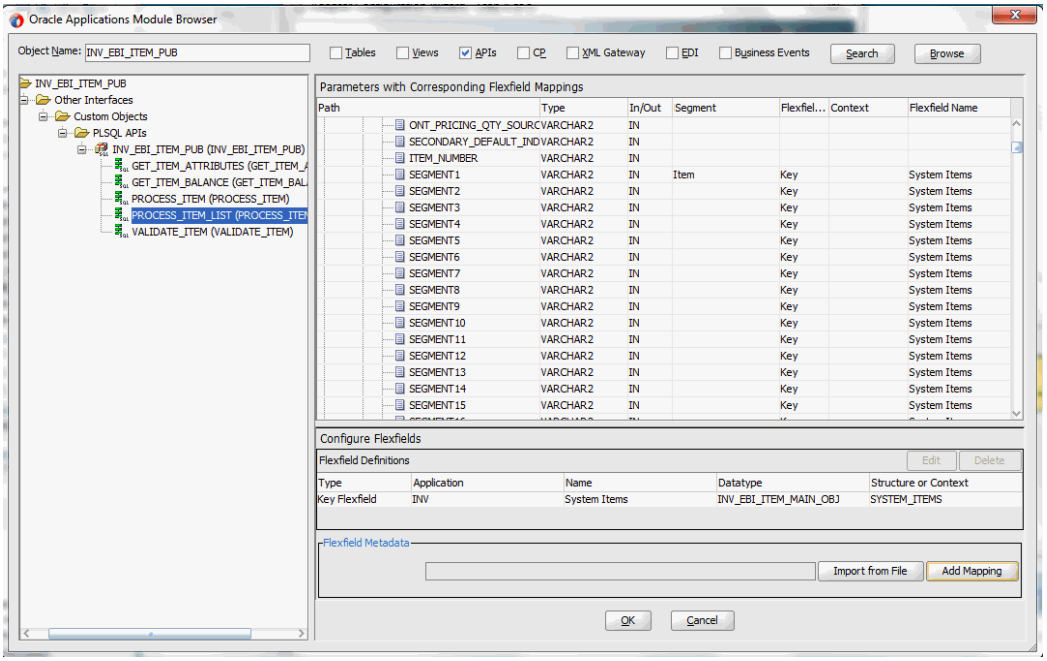
The following table explains the relationship between the key flexfield mapping data and the actual segments defined in Oracle E-Business Suite:

Segment Summary for Key Flexfield 'System Items'

Key Flexfield Segments Mapped During Flexfield Configuration	Key Flexfield Segments Defined in Oracle E-Business Suite
SEGMENT1	Item
SEGMENT2	Not defined
SEGMENTn	Not defined

Mapped segment column names for the key flexfield 'System Items' are displayed in the Oracle E-Business Suite Module Browser once the mapping is complete. In this case, only 'Item', the corresponding column name in SEGMENT1, is shown here.

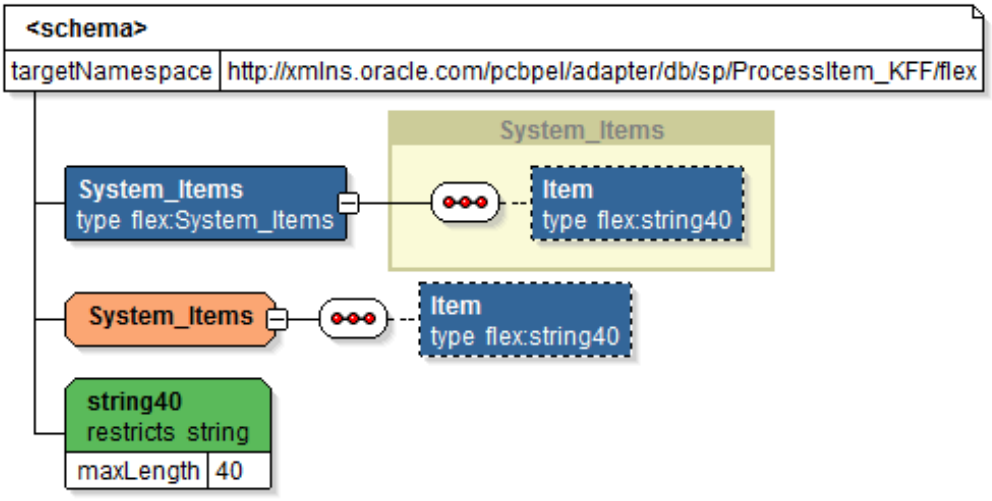
Displaying Mapped Segment Column Name 'Item'



For information on how to define key flexfields and key flexfield segments, refer to the Planning and Defining Key Flexfields chapter, *Oracle E-Business Suite Flexfields Guide*.

Flexfield Validation in the Schema File

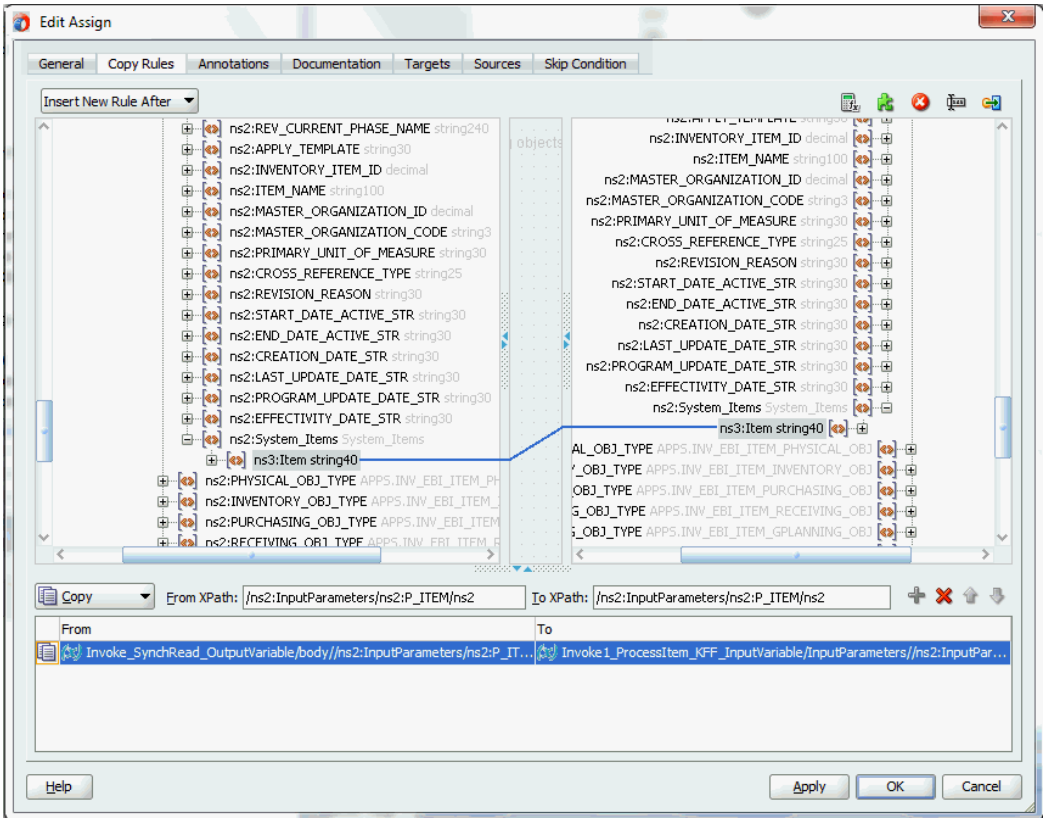
Select the enhanced XSD file (for example `ProcessItem_KFF_flex.xsd`) in Oracle JDeveloper and notice that the mapped key flexfield column (Item) is displayed with type 'flex' under the flexfield name System_Items.



Assigning Parameters with Key Flexfields in the Assign Activity

In the **Assign** activity of the BPEL process diagram within the SOA composite application, the mapped key flexfield can be displayed and used to pass an input variable from payload to the second **Invoke** activity for service invocation.

Displaying Mapped Key Flexfields in the Assign Activity



Use **Assign** activity to provide input payload information to the Item object of the target parameter.

> **Note:** The target parameter `ns2:SEGMENT1` mapped to the column name `SEGMENT1` during the key flexfield configuration earlier is now translated into `ns3:Item` (under `ns2:System_Items`) and displayed in the **Assign** activity.

- In the From navigation tree, navigate to **Variables > Process > Variables > Invoke_SynchRead_OutputVariable > body > ns2:InputParameters > ns2:P_ITEM > ns2:P_ITEM_ITEM > ns2:MAIN_OBJ_TYPE (datatype INV_EBI_ITEM_MAIN_OBJ) > ns2: System_Items** and select **ns3: Item**.

- In the To navigation tree, navigate to **Variables > Process > Variables > Invoke1_ProcessItem_KFF_InputVariable > InputParameters > ns2:InputParameters > ns2:P_ITEM > ns2:P_ITEM_ITEM > ns2:MAIN_OBJ_TYPE (datatype INV_EBI_ITEM_MAIN_OBJ) > ns2: System_Items** and select **ns3: Item**.

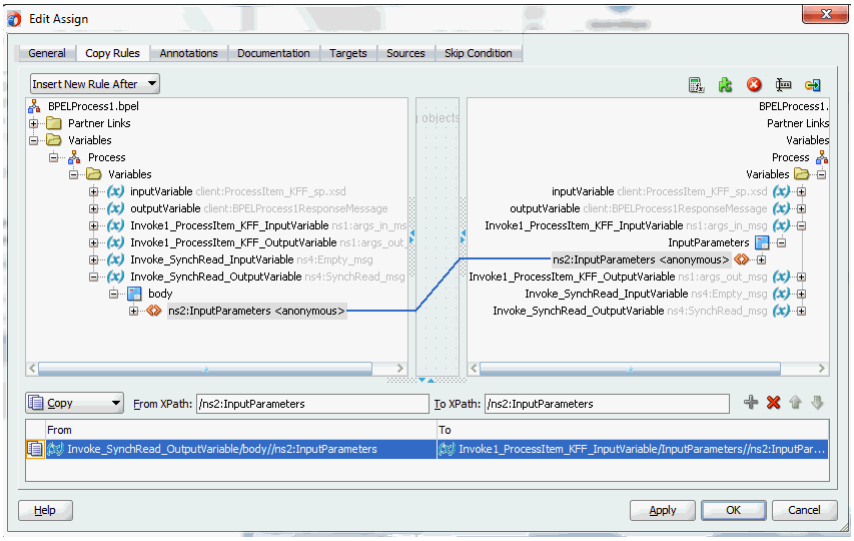
Drag the source node (ns3: Item) to connect to the target node (ns3: Item) that you just specified. This creates a line that connects the source and target nodes. The copy rule is

displayed in the From and To sections at the bottom of the Edit Assign dialog box.

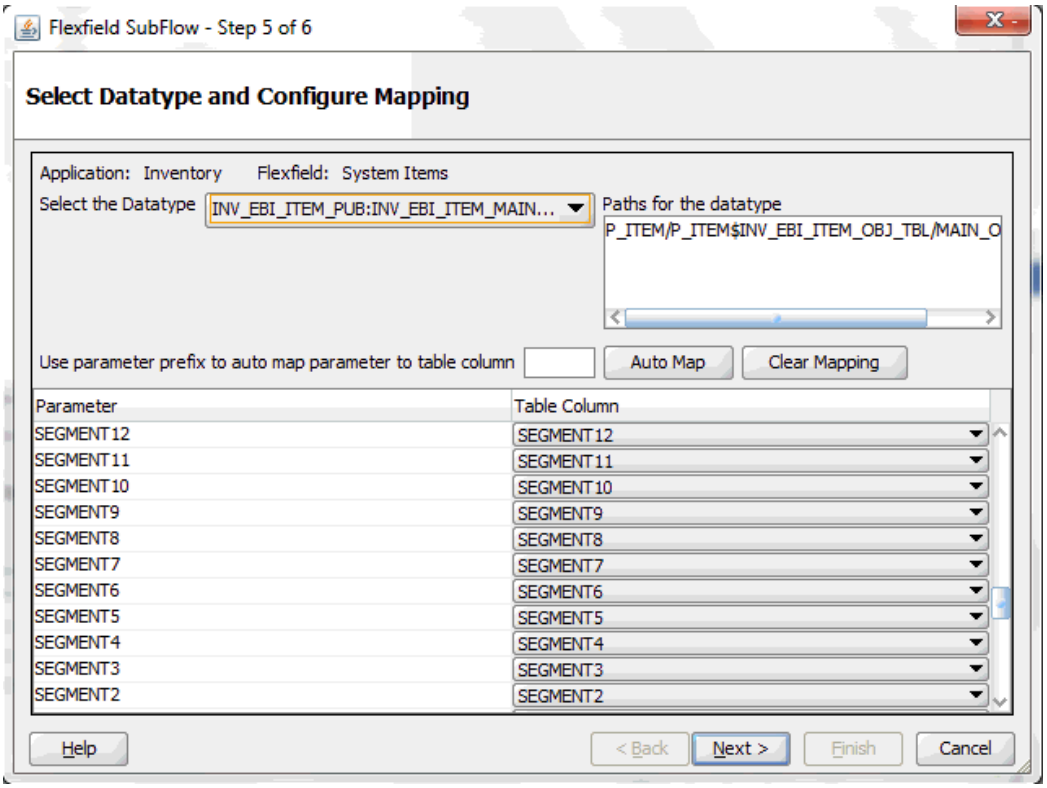
Click **Apply** and **OK** to complete the **Assign** activity.

> **Note:** Alternatively, since 'ns2:InputParameters' variables contained in the Source and Target are identical, you can simply drag the source node (ns2:InputParameters) to connect to the target node (ns2:InputParameters) and complete the **Assign** activity.

Assigning Input Variables to the Invoke Activity



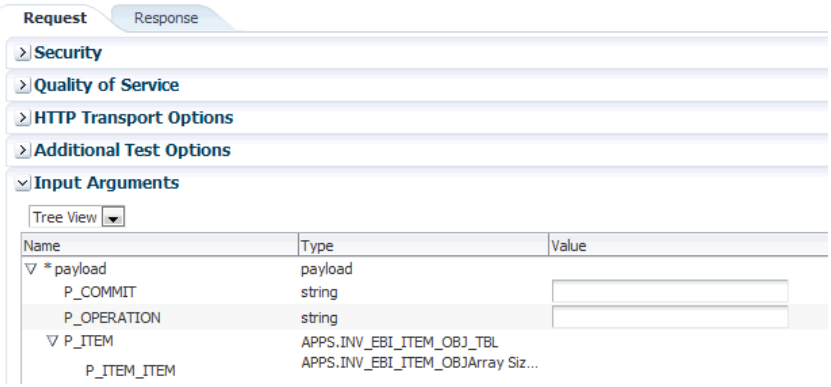
Please note that the datatype INV_EBI_ITEM_MAIN_OBJ shown here is exact the same datatype (INV_EBI_ITEM_PUB:INV_EBI_ITEM_MAIN_OBJ) specified earlier from the drop-down list in the Select Datatype and Configure Mapping page. P_ITEM is listed as the paths for the selected datatype.



Displaying Mapped Key Flexfields at Run Time

After deploying the BPEL process contained in a SOA Composite application (`FlexfieldKFF`) to the Oracle WebLogic managed server (for example 'soa-server1'), the key flexfields we configured earlier are displayed in the soa-server1 server (`http://<servername>:<portnumber>/soa-infra`) allowing you to specify payload data and test the service invocation.

> **Note:** Please note that you can also test and manage the process from the Oracle Enterprise Manager Fusion Middleware Control Console. In this example, the mapped key flexfield used as part of the input payload happens to be located or nested within multiple sub-levels down from the top. This means that we need to drill down to the mapped key flexfield `Item` by expanding the parameter nodes multiple times from the top level (Payload > P_ITEM > P_ITEM_ITEM > MAIN_OBJ_TYPE > System_Items > Item). However, Oracle Enterprise Manager Fusion Middleware Control Console has certain limitations on displaying payload parameters if they are nested too deeply.

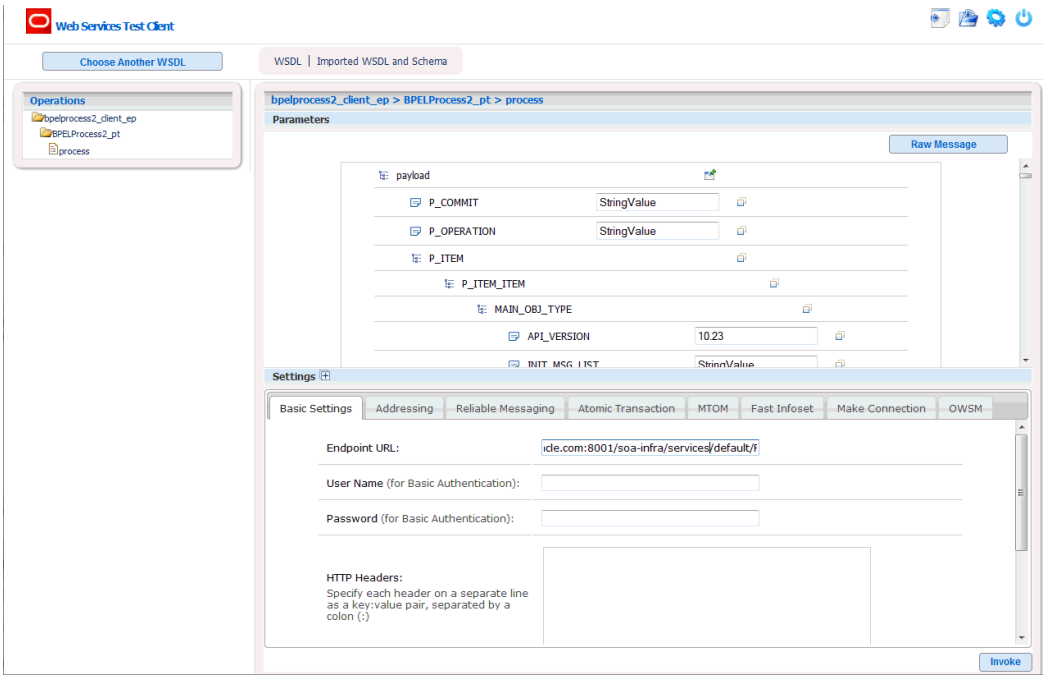


Therefore, in this case we access the mapped key flexfield directly from the `soa-infra` in the soa-server1 server.

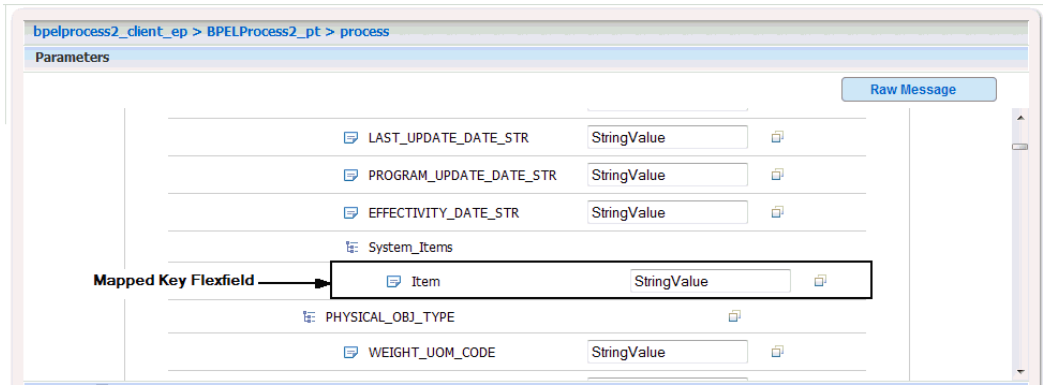
For information on how to test the service invocation from the Oracle Enterprise Manager Fusion Middleware Control Console, see Test the deployed SOA Composite application with BPEL process, page 9-42.

In the Welcome to the Oracle SOA/BPM Platform on WebLogic page, click the deployed key flexfield project.

The Web Services Test Client page is displayed with your selected project. Click **Test**. The key flexfield configured earlier will be displayed as part of the input payload in the Parameters region of the Web Services Test Client page.



In the Parameters region, expand the Payload > P_ITEM > P_ITEM_ITEM > MAIN_OBJ_TYPE > System_Items. Mapped key flexfield 'Item' is displayed as part of the payload.



Enter appropriate data as payload and then click **Invoke** to invoke the service.

Configuring Descriptive Flexfield Mappings for a PL/SQL API

Sample Business Scenario

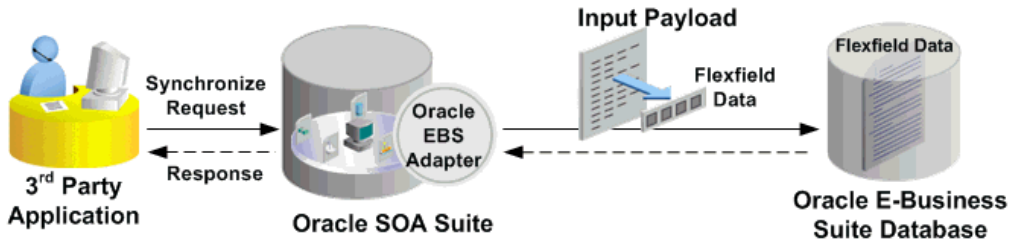
Take the `HZ_AIA_CUSTOM_PKG.SYNC_ACCT_ORDER` API as an example to explain the inbound web service creation with flexfield data.

When a request is placed for synchronizing customer information from a third party application, the descriptive flexfield data as part of the payload is routed and the PL/SQL API `HZ_AIA_CUSTOM_PKG.SYNC_ACCT_ORDER` is invoked to update the

organization and customer accounts in Oracle E-Business Suite. The synchronized customer data will be passed to the client as the response message.

The following diagram illustrates the service invocation process flow:

Invoking an Inbound Service with Flexfield Data



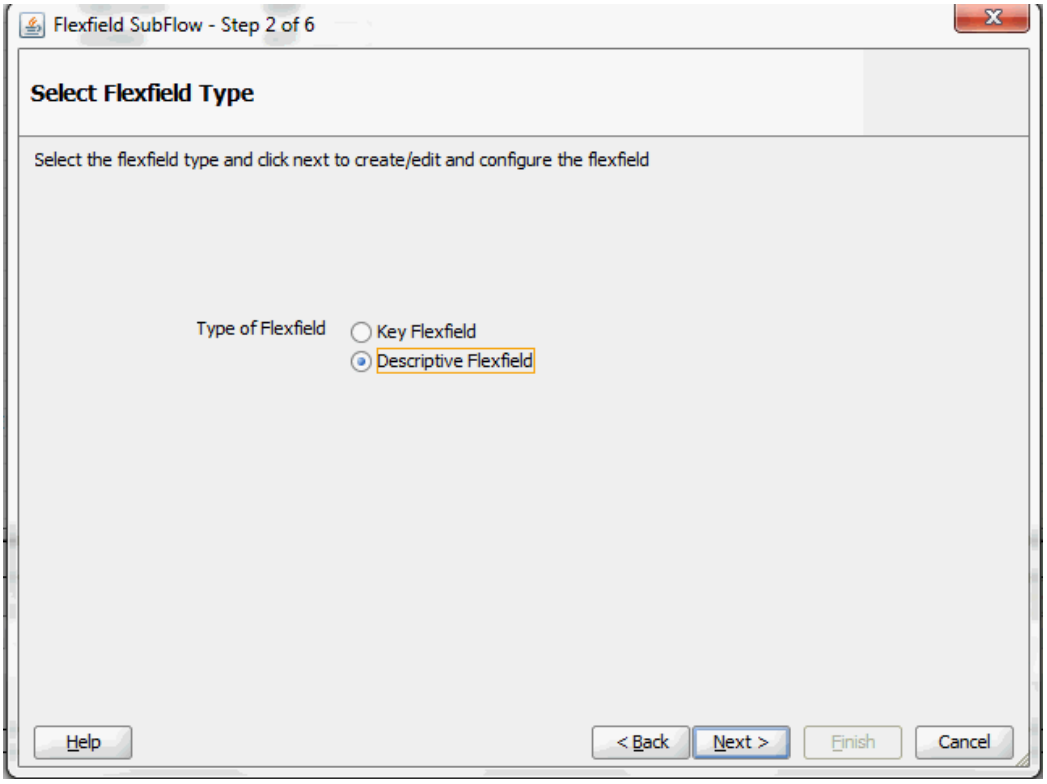
After deploying the service, you can validate the process by querying customer data directly from Oracle Customer Online application. The retrieved customer data should be the same as input from the payload.

Configuring a Descriptive Flexfield Mapping As Part of the Partner Link Creation

To configure a descriptive flexfield mapping for the selected API `HZ_AIA_CUSTOM_PKG.SYNC_ACCT_ORDER`, click **Add Mapping** in the Configure Flexfields region. A series of flexfield selection pages in the wizard appear guiding you to configure a new mapping for your selected API.

Click **Next** in the Welcome page. The Select Flexfield Type page is displayed.

Flexfield Subflow - Select Flexfield Type



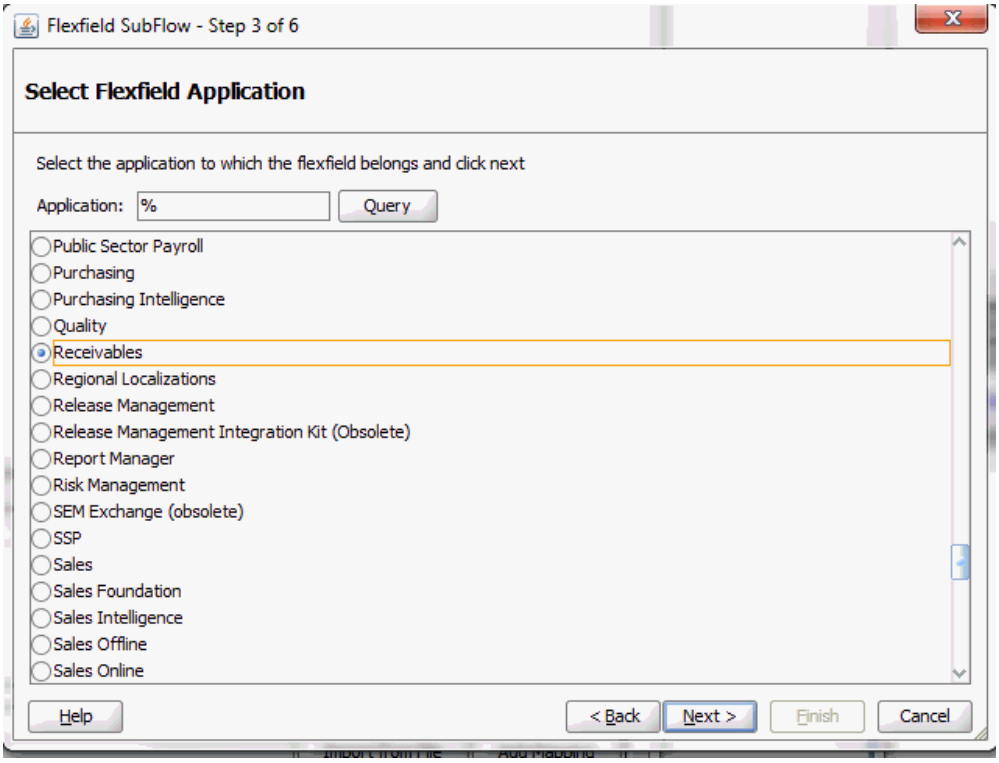
Perform the following steps to configure a descriptive flexfield:

1. Select the **Descriptive Flexfield** radio button in the Select Flexfield Type page, and then use the similar approach as described earlier for key flexfields in the flexfield mapping wizard.

2. Click **Next** in the Select Flexfield Type page. The Select Flexfield Application page is displayed.

3. To locate your desired application, enter keyword search (such as '%App%') or simply click **Query** to execute the search. All the matched applications for descriptive flexfield are displayed. For example, select the **Receivables** radio button as the application name to which the descriptive flexfield belongs.

Flexfield Subflow - Select Flexfield Application

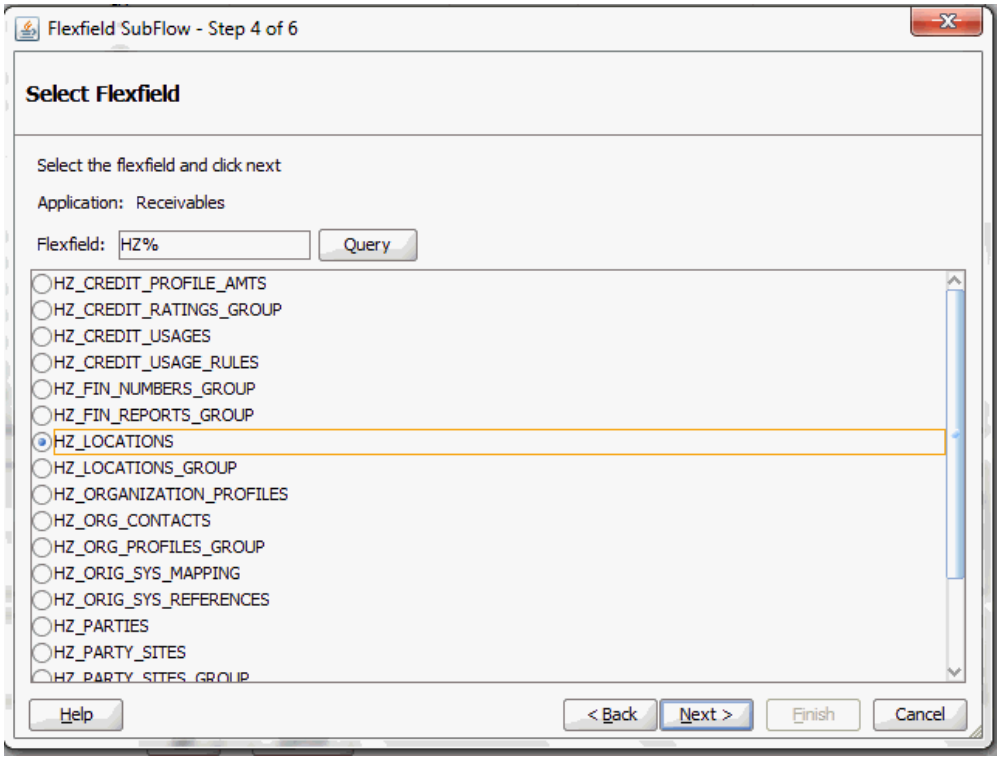


Click **Next**.

4. In the Select Flexfield page, enter keyword search (such as 'HZ%') if desired and click **Query** to execute the search. All the matched descriptive flexfields are displayed for your selection.

   Select your desired descriptive flexfield (such as HZ_LOCATIONS) and click **Next**.

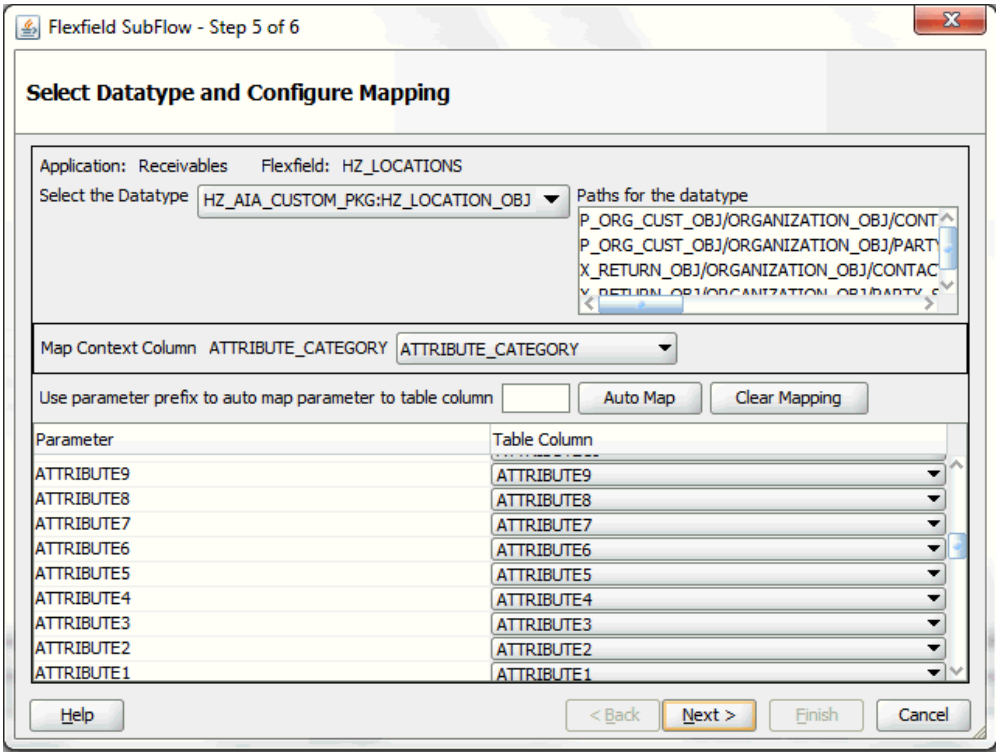
Flexfield Subflow - Select Flexfield



5. The Select Datatype and Configure Mapping page appears where you can choose desired mapping for your descriptive flexfield.

   Notice that your selected descriptive flexfield name HZ_LOCATIONS is automatically populated as the Flexfield name.

Flexfield Subflow - Select Path and Configure Mapping



To configure flexfield mapping, select a desired data type from the Datatype drop-down list. This retrieves all needed flexfield metadata from Oracle E-Business Suite database and populates parameters at the parent level for your mapping.

- **Parent type:** Parent type is a pseudo type to allow all the scalar parameters at the procedure level to participate in the mapping. It can be used for simple APIs that just have scalar parameters.

- **Record type:** Record type can be selected if it's a complex API that has record types in addition to scalar types.

Selecting Context Column for Descriptive Flexfield Mapping

For descriptive flexfields, you need to perform an additional mapping for the context column. This is because flexfield mapping cannot span across record types. All the table columns including the context column should be mapped to the scalar parameters within one record type or parent type.

In the Map Context Column ATTRIBUTE_CATEGORY field, select a desired category name that you want to view the mapping information from the drop-down list. For example, select ATTRIBUTE_CATEGORY from the drop-down list.

Note: If a parameter has already mapped to a table column, you cannot remap the parameter to other column. In this situation, 'Mapped to other Flexfield' appears instead for the mapped parameter.

Similar to key flexfields, the mapping can be performed manually or assisted with the automapping feature by clicking **Auto Map**.

Automapping feature maps all the table columns to the parameters, which have the column name as part of the parameter name. For example, table column names from ATTRIBUTE1 to ATTRIBUTE20 are automatically displayed and mapped to parameters from ATTRIBUTE1 to ATTRIBUTE20 respectively through automapping.

If there are multiple sets of parameters and the table column names, then the prefix needs to be specified for automapping. This would map the potential parameters which need to be considered for the mapping. Enter parameter prefix (such as 'INDUSTRY_') to first locate your desired parameters that you want the automapping to be performed. Next, click **Auto Map**. All the parameters with prefix 'INDUSTRY_' are automatically displayed along with the automapped table column names. Parameter 'INDUSTRY_ATTRIBUTE1' is mapped to column ATTRIBUTE1, and parameter 'INDUSTRY_ATTRIBUTEn' is mapped to column ATTRIBUTEn.

Instead of using automapping feature, you can manually select a desired table column name from the drop-down list for a corresponding parameter for your flexfield mapping.

Click **Clear Mapping** to clear current mapping you have in the table column only. All previously mapped columns marked with 'Mapped to other Flexfield' will not be removed from the system.

Click **Next** to proceed to the next page.

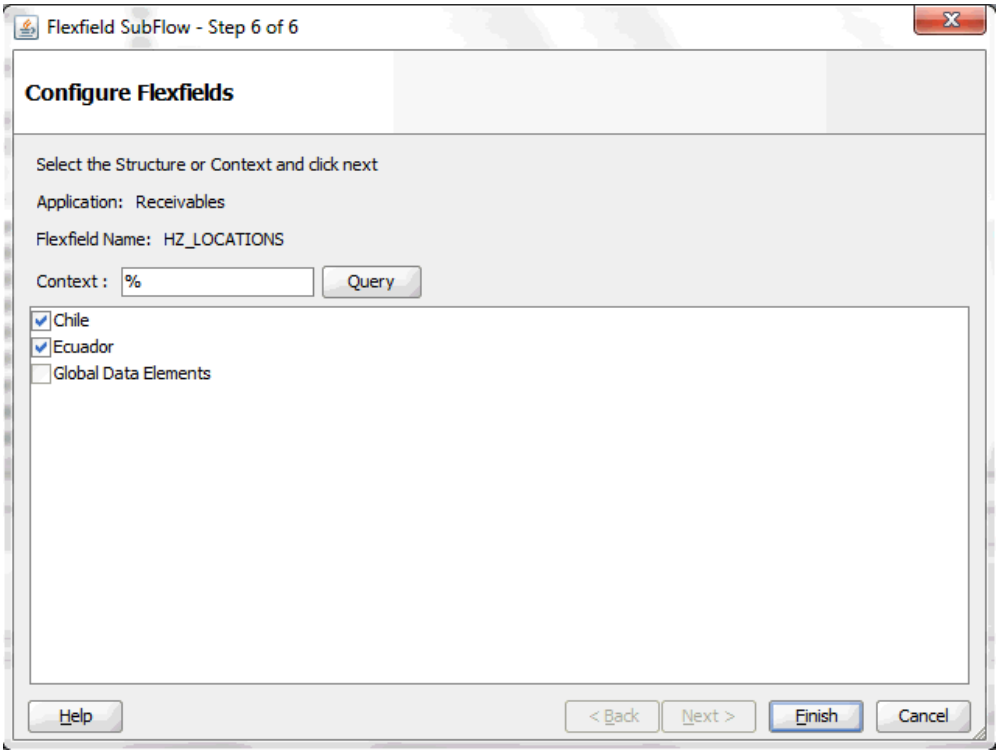
6. In the Configure Flexfields page, select at least one context value for the selected descriptive flexfield. For example, select desired context values (such as Chile and Ecuador).

   You can either enter keyword search in the Context field or simply click **Query** to execute the search before selection.

   For descriptive flexfields, context value should be available to the runtime through an API parameter.

   Please note that there are certain criteria need to be met in order to have your desired context values displayed in the wizard for selection. See: Guidelines for Defining Flexfield Structure and Context Values, page 4-34.

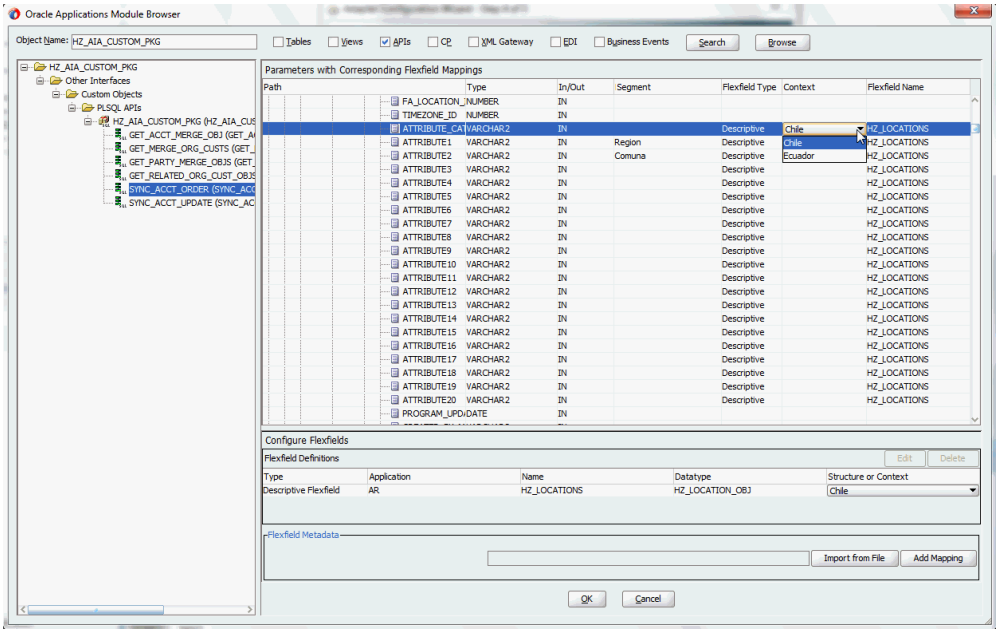
Flexfield Subflow - Configure Flexfields



Click **Finish** to complete the flexfield configuration.

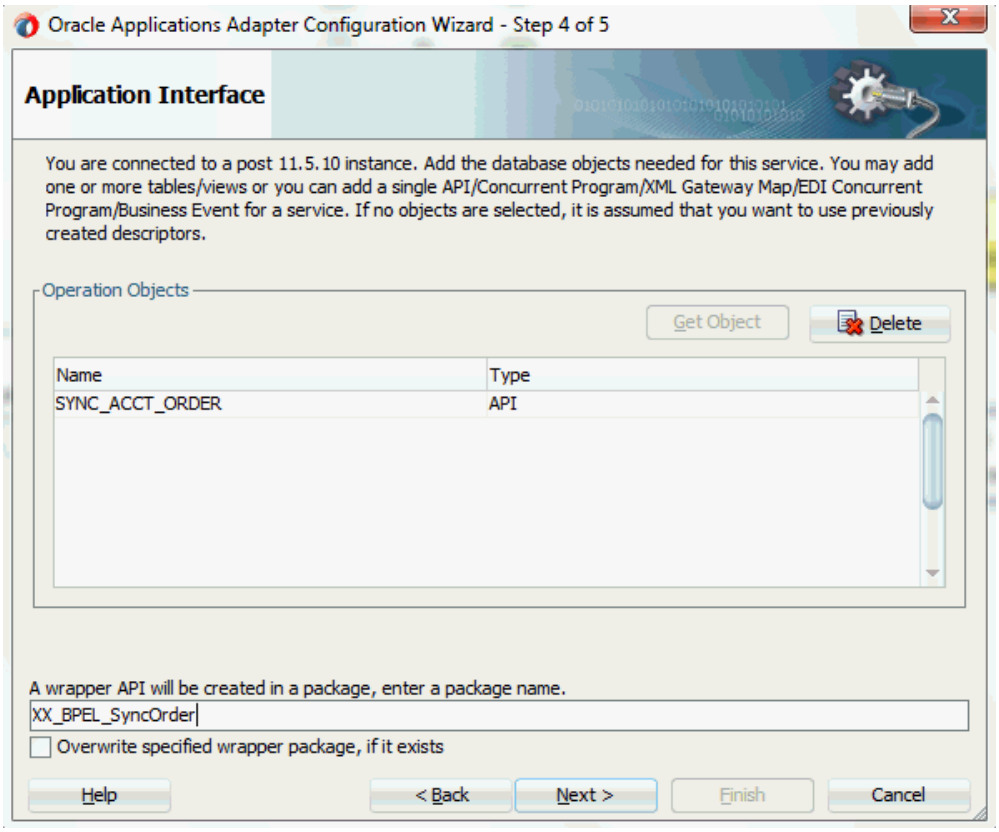
7. The Oracle E-Business Suite Module Browser window appears. The newly added descriptive flexfield information including the segment values for the selected context value is displayed in the right pane of the window.

Displaying Selected Descriptive Flexfield Information



Click **OK**.

8. The Application Interface page appears with the selected API.



Click **Next** and then click **Finish**.

The wizard generates the WSDL file corresponding to the partner link. An enhanced XSD file, for example `SynAccOrder_flex.xsd`, is also created. This XSD file contains the schema describing the procedure arguments with the elements representing the parameters that have been replaced with flexfield segment names.

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

9. Additionally, configuration file and flexfield mapping file are created. For more information on these files, see Reviewing Flexfield Mapping Configurations, page 4-122.

Once a new flexfield mapping has been configured or added, if modification is needed, you can make updates by first selecting desired flexfield type that you want to modify and then make needed changes. For more information on how to modify the selected mapping details, see Modifying Flexfield Mapping Definitions, page 4-119.

Displaying Mapped Descriptive Flexfields at Design Time

Similar to key flexfield mappings, the descriptive flexfield mapping data configured earlier during the partner link creation can be shown at design time along with the following activities orchestrated in the BPEL process within in a SOA composite:

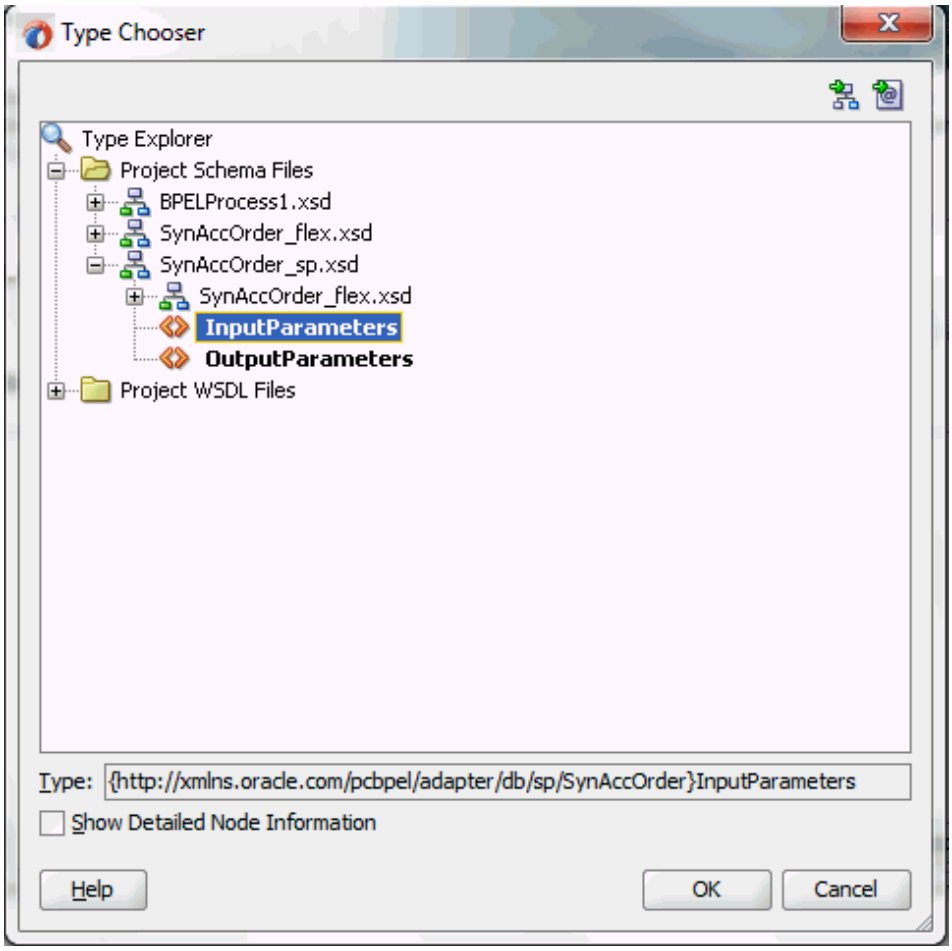
A Composite Example with Descriptive Flexfield Mappings

Here is an example describing the descriptive flexfield mappings in the BPEL process of a SOA composite:

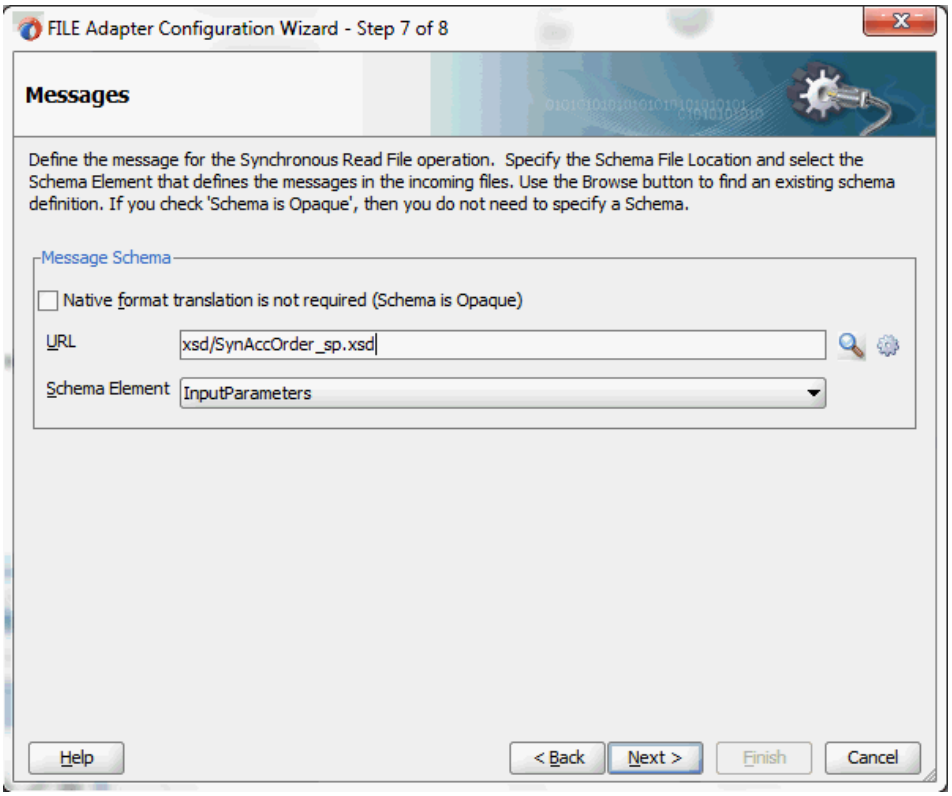
- Create a new SOA composite application with BPEL process
- Add a partner link service called `SynAccOrder` with descriptive flexfield mappings that we configured earlier for the `HZ_AIA_CUSTOM_PKG.SYNC_ACCT_ORDER` API

  For information on how to configure descriptive flexfield mappings, see Configuring Descriptive Flexfield Mappings for a PL/SQL API, page 4-59.
- Add a partner link for File Adapter to read input payload that contains flexfield data for service invocation
  - In the File Directories dialog, enter the physical directory (such as `/usr/tmp`) where the input payload xml file (such as `input_payload.xml`) resides.
  - In the Messages dialog, select the 'browse for schema file' icon next to the URL field to open the Type Chooser.

    Click Type Explorer and select **Project Schema Files > SynAccOrder_sp.xsd > SynAccOrder_flex.xsd > InputParameters**.



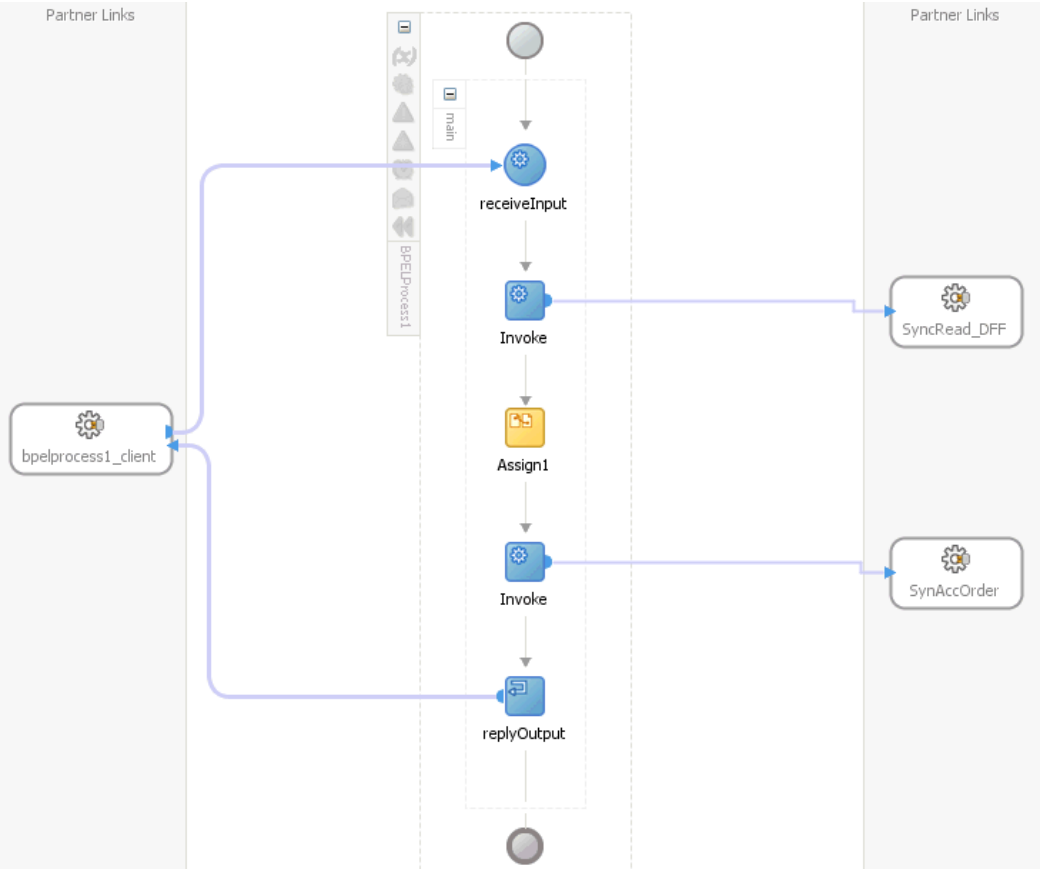
Click **OK**. The selected schema information will be automatically populated in the URL and Schema Element fields.



- Configure two **Invoke** activities
  - Associate the first **Invoke** activity with the File Adapter partner link to synchronously read input payload data
  - Associate the second **Invoke** activity with the partner link `SynAccOrder` that contains flexfield mapping to invoke the service.
- Add an **Assign** activity to pass input payload received from the File Adapter to the second **Invoke** activity for service invocation

  Mapped descriptive flexfield column names corresponding to the flexfield matching segments defined in the Oracle E-Business Suite are displayed in the **Assign** activity. See: Assigning Parameters with Descriptive Flexfields in the Assign Activity, page 4-74.

Constructing BPEL Process in SOA Composite



For more information on how to create design time tasks, see Design-Time Tasks for PL/SQL APIs, page 9-2.

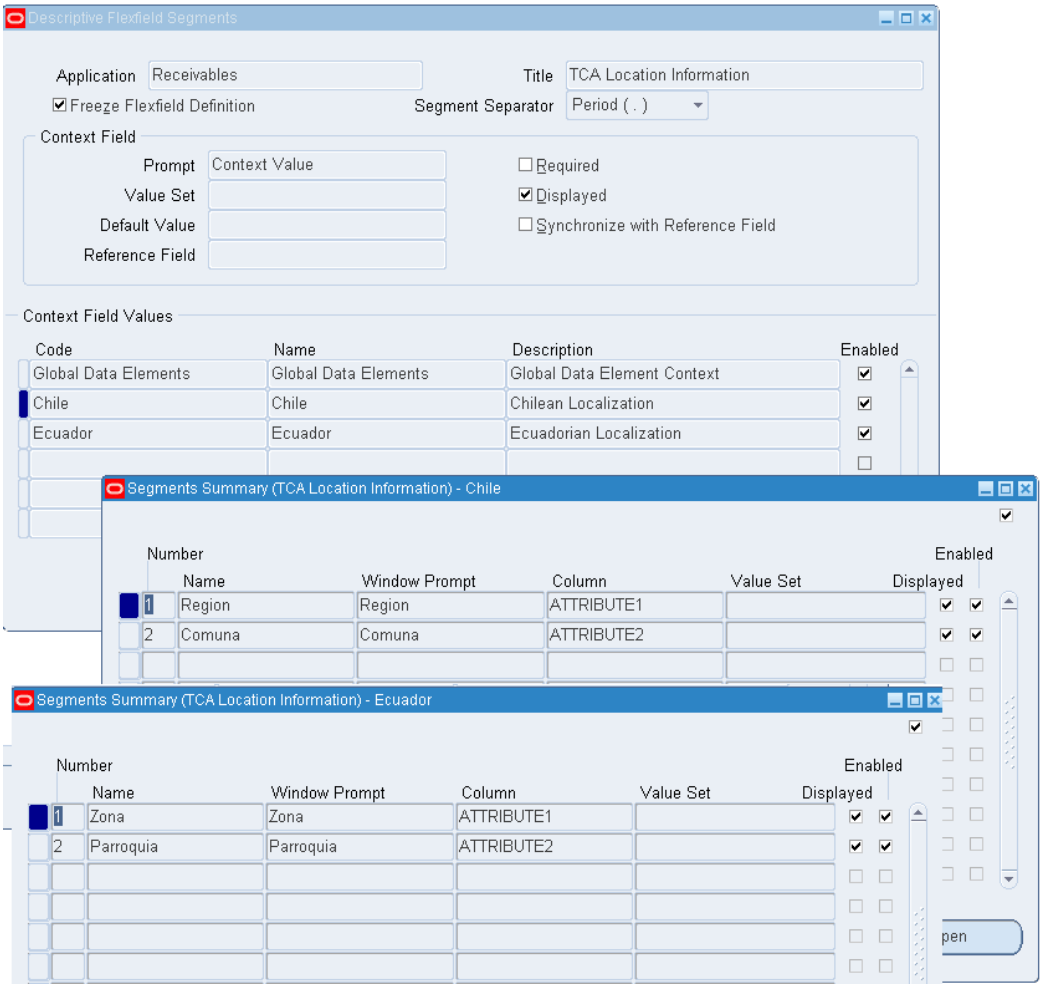
Flexfield Validation in Oracle E-Business Suite

When defining descriptive flexfield attributes for 'TCA Location Information' (HZ_LOCATIONS) in Oracle Receivables application (Application Developer responsibility), the following attributes are specified based on context values:

Descriptive Flexfield Segment Summary for TCA Location Information

Context Value	ATTRIBUTE1	ATTRIBUTE2
Chile	Region	Comuna
Ecuador	Zona	Parroquia

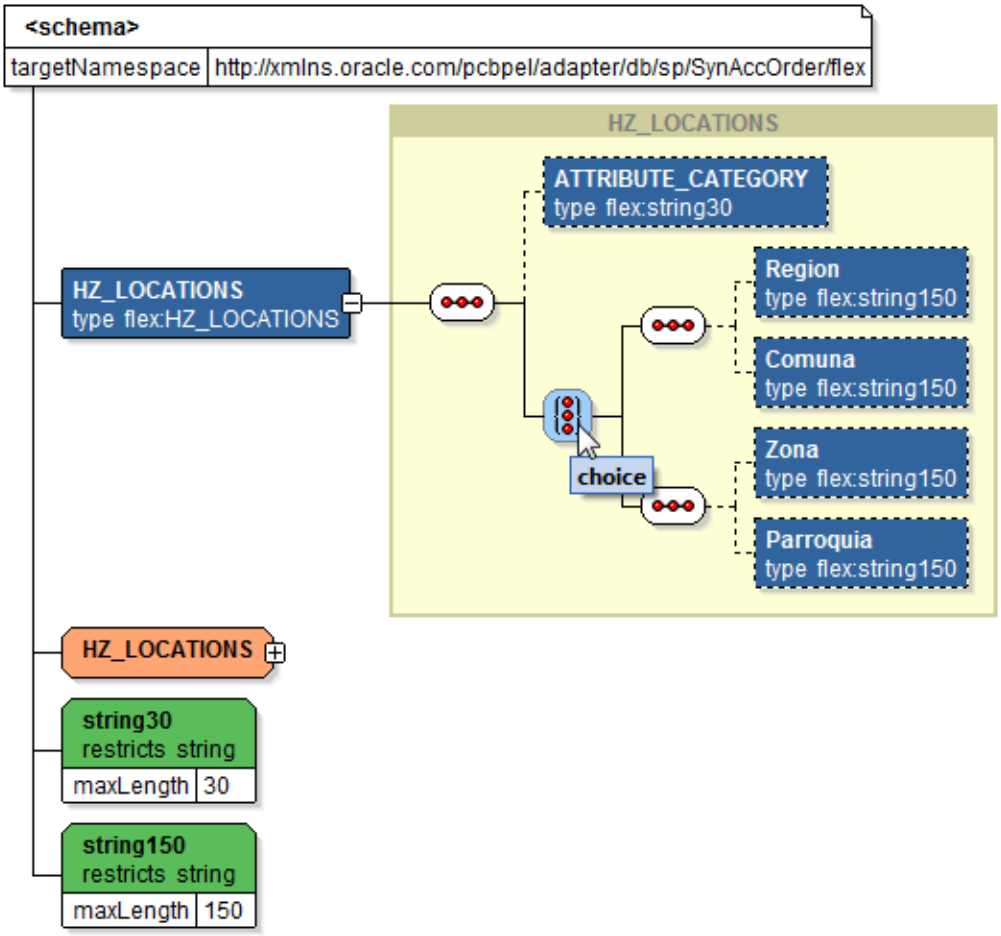
Descriptive Flexfield Segments Summary for System Items



Flexfield Validation in the Schema File

Select the enhanced XSD file (such as `SynAccOrder_flex.xsd`) in Oracle JDeveloper and notice that the mapped descriptive flexfields are displayed with type 'flex' under the HZ_LOCATIONS flexfield as part of the schema.

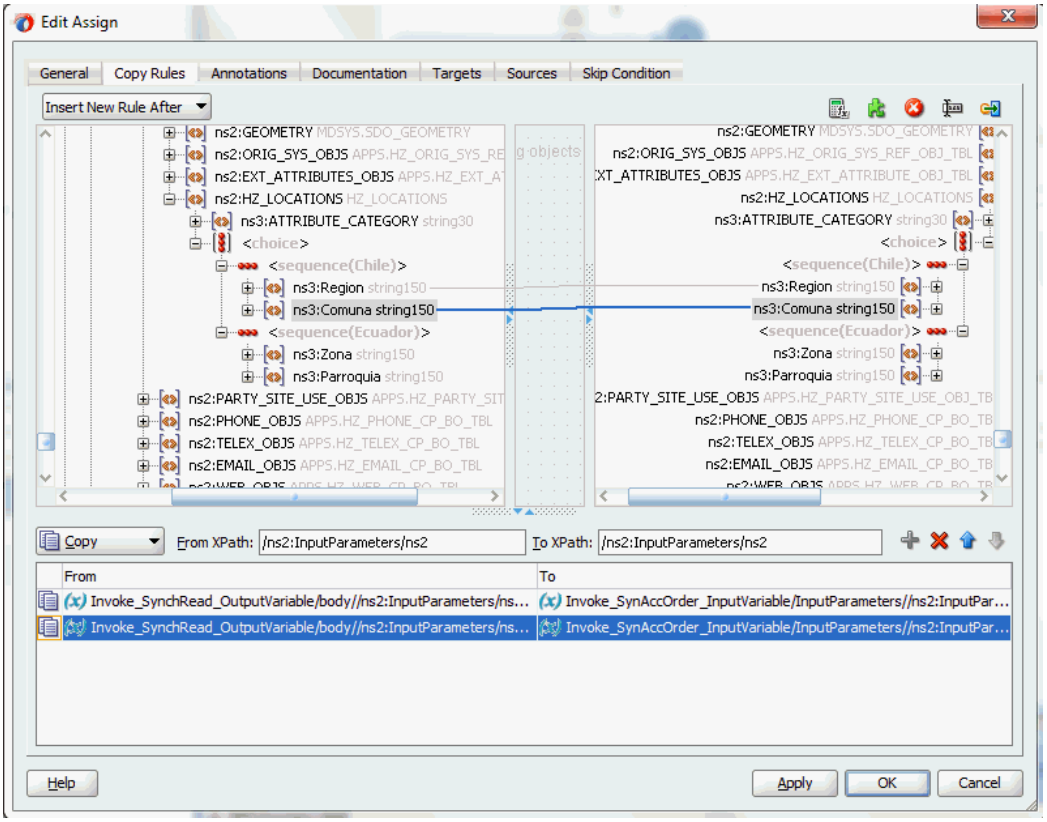
Additionally, the mapped table column names (`Region, Comuna, Zona,` and `Parroquia`) are grouped by context value (`Chile` and `Ecuador`) listed in a tree structure. Context value nodes are connected with the **<choice>** icon in between indicating that only either one of the context values and its associated table columns can be selected.



Assigning Parameters with Descriptive Flexfields in the Assign Activity

In the **Assign** activity of the BPEL process diagram within the SOA composite application, the mapped descriptive flexfields can be displayed and used to pass an input variable received from payload to the **Invoke** activity for `SynAccOrder` service invocation.

Mapped Descriptive Flexfields Displayed in the Assign Activity



Use **Assign** activity to provide input payload information to the TCA Location object of the target parameter.

The target parameters `ns2:ATTRIBUTE1` was mapped to the following descriptive flexfield columns during the flexfield configuration earlier:

- `Region` (under context value `<sequence(Chile)>`)
- `Zona` (under context value `<sequence(Ecuador)>`)

The target parameters `ns2:ATTRIBUTE2` was mapped to the following descriptive flexfield columns during the flexfield configuration earlier:

- `Comuna` (under context value `<sequence(Chile)>`)
- `Parroquia` (under context value `<sequence(Ecuador)>`)

These descriptive flexfield columns grouped by context value are displayed in the **Assign** activity.

- In the From navigation tree, navigate to **Variables > Process > Variables > Invoke_SynchRead_OutputVariable > body > ns2:InputParameters >**

ns2:P_ORG_CUST_OBJ > ns2:ORGANIZATION_OBJ > ns2:PARTY_SITE_OBJS> ns2:PARTY_SITE_OBJS_ITEM > ns2:LOCATION_OBJ >ns2:HZ_LOCATIONS >ns3:ATTRIBUTE_CATEGORY and expand the **<choice>** icon to locate the mapped descriptive flexfield columns `ns3:Region` and `ns3:Comuna` for the selected context value `ns3:Chile` (or select `ns3:Zona` and `ns3:Parroquia` for the selected context value `ns3:Ecuador` instead).

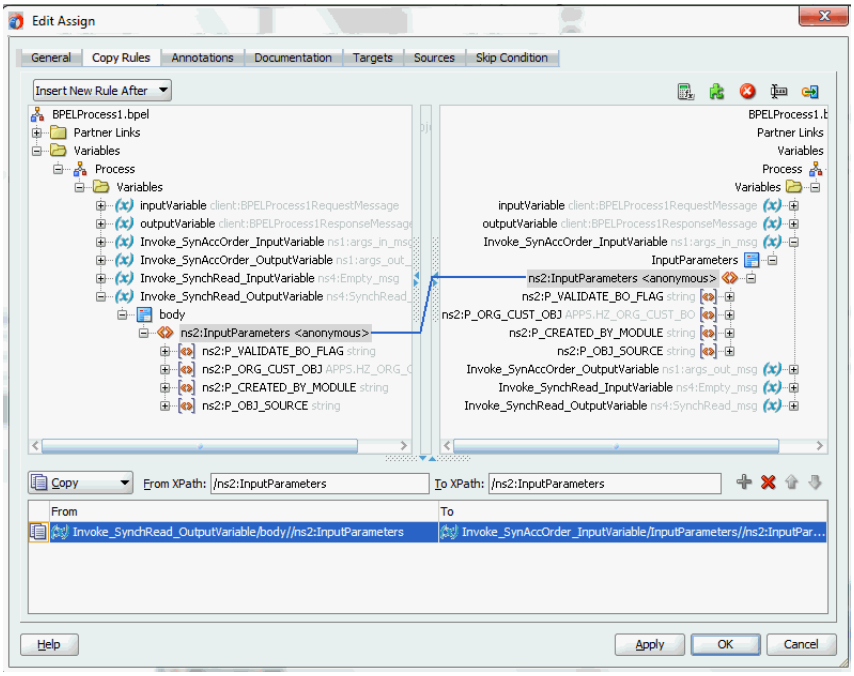
- In the To navigation tree, navigate to **Invoke_SyncOrder_InputVariable > InputParameters > ns2:InputParameters > ns2:P_ORG_CUST_OBJ > ns2:ORGANIZATION_OBJ > ns2:PARTY_SITE_OBJS> ns2:PARTY_SITE_OBJS_ITEM > ns2:LOCATION_OBJ >ns2:HZ_LOCATIONS >ns3:ATTRIBUTE_CATEGORY** and expand the **<choice>** icon to locate the mapped descriptive flexfield columns `ns3:Region` and `ns3:Comuna` for the selected context value `ns3:Chile` (or select `ns3:Zona` and `ns3:Parroquia` for the selected context value `ns3:Ecuador` instead).

To assign the parameters with descriptive flexfields in the **Assign** activity, drag the source node (ns3:Region) to connect to the first target node (ns3:Region) that you just specified. This creates a line that connects the source and target nodes. The copy rule is displayed in the From and To sections at the bottom of the Edit Assign dialog box.

Repeat the procedure to assign the second parameter by dragging the same source node (ns3:Comuna) to connect to the second target node (ns3:Comuna). The second copy rule is displayed in the From and To sections at the bottom of the Edit Assign dialog box.

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

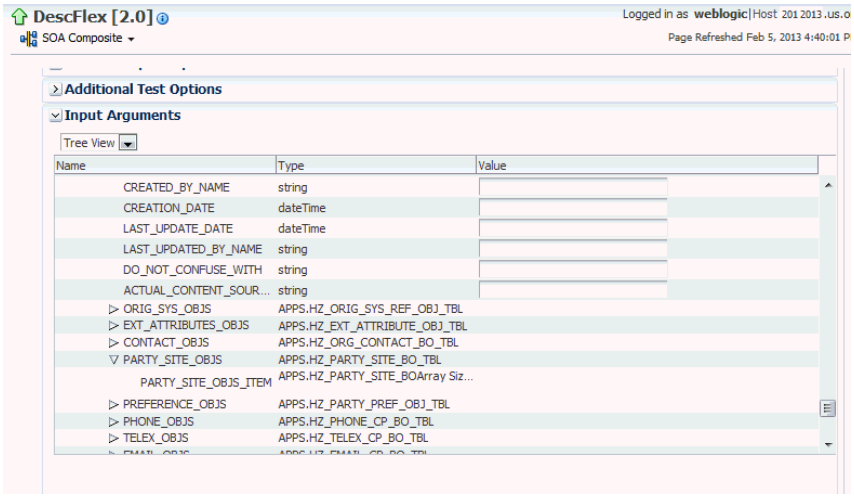
> **Note:** Alternatively, since 'ns2:InputParameters' variables contained in the Source and Target are identical, you can simply drag the source node (ns2:InputParameters) to connect to the target node (ns2:InputParameters) and complete the **Assign** activity.



Displaying Mapped Descriptive Flexfields at Run Time

After deploying the BPEL process contained in a SOA Composite application to the Oracle WebLogic managed server (such as 'soa-server1'), the descriptive flexfields we configured earlier are displayed in the soa-server1 server (`http://<servername>:<portnumber>/soa-infra`) allowing you to specify payload data and test the service invocation. Please note that you also test and manage the process from the Oracle Enterprise Manager Fusion Middleware Control Console.

> **Note:** Please note that you can also test and manage the process from the Oracle Enterprise Manager Fusion Middleware Control Console. In this example, the mapped descriptive flexfields used as part of the input payload happen to be located or nested within multiple sub-levels down from the top. This means that we need to drill down to the mapped descriptive flexfields (`Region` and `Comuna`) by expanding the parameter nodes multiple times from the top level (Payload > P_ORG_CUST_OBJ > ORGANIZATION_OBJ > PARTY_SITE_OBJS > PARTY_SITE_OBJS_ITEM > LOCATION_OBJ > HZ_LOCATIONS > ATTRIBUTE_CATEGORY > HZ_LOCATIONS_1 to view descriptive flexfields). However, Oracle Enterprise Manager Fusion Middleware Control Console has certain limitations on displaying payload parameters if they are nested too deeply.



Therefore, in this case we access the mapped descriptive flexfields directly from the `soa-infra` in the soa-server1 server.

For information on how to test the service invocation from the Oracle Enterprise Manager Fusion Middleware Control Console, see Test the deployed SOA Composite application with BPEL process, page 9-42.

Similar to the key flexfield project described earlier, click the deployed descriptive flexfield project in the Welcome to the Oracle SOA/BPM Platform on WebLogic page.

The Web Services Test Client page is displayed with your selected project. Click **Test**. The descriptive flexfields configured earlier will be displayed as part of the input payload in the Parameters region of the Web Services Test Client page.

In the Parameters region, expand the Payload > P_ORG_CUST_OBJ node and navigate to ORGANIZATION_OBJ > PARTY_SITE_OBJS > PARTY_SITE_OBJS_ITEM > LOCATION_OBJ > HZ_LOCATIONS > ATTRIBUTE_CATEGORY > HZ_LOCATIONS_1.

Two sets of descriptive flexfields configured earlier at the design time are now displayed at the runtime.

1. HZ_LOCATIONS_1_0 (This is the default selection we configured in the Assign activity at the design time.)
   - Region
   - Comuna
2. HZ_LOCATIONS_1_1
   - Zona

- Parroquia

Enter appropriate data as payload and then click **Invoke** to invoke the service.

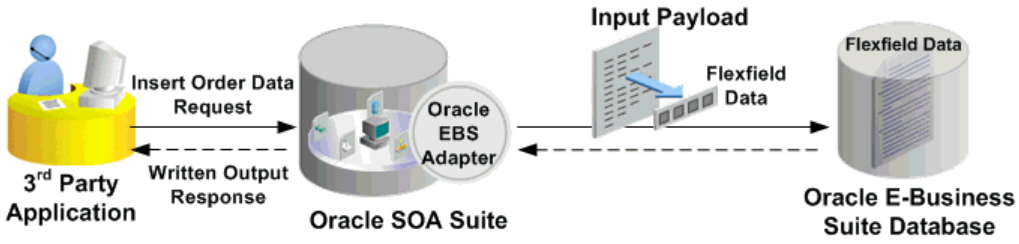
Configuring Key Flexfield Mappings for an Open Interface Table

Sample Business Scenario

Take an open interface table `ONT.OE_HEADERS_IFACE_ALL` as an example to explain the inbound web service creation with flexfield data.

When a request of inserting order data for a specific account is received, order data details can be customized based on the requestor's needs using both key flexfields and descriptive flexfields. This data will then be passed as an input payload and inserted into the open interface table. The inserted data can be selected and written as an output file and returned back as a response message for client verification.

The following diagram illustrates the service invocation process flow:



After deploying the service, you can validate the process directly by viewing the output XML file. This data should be the same as input from the payload.

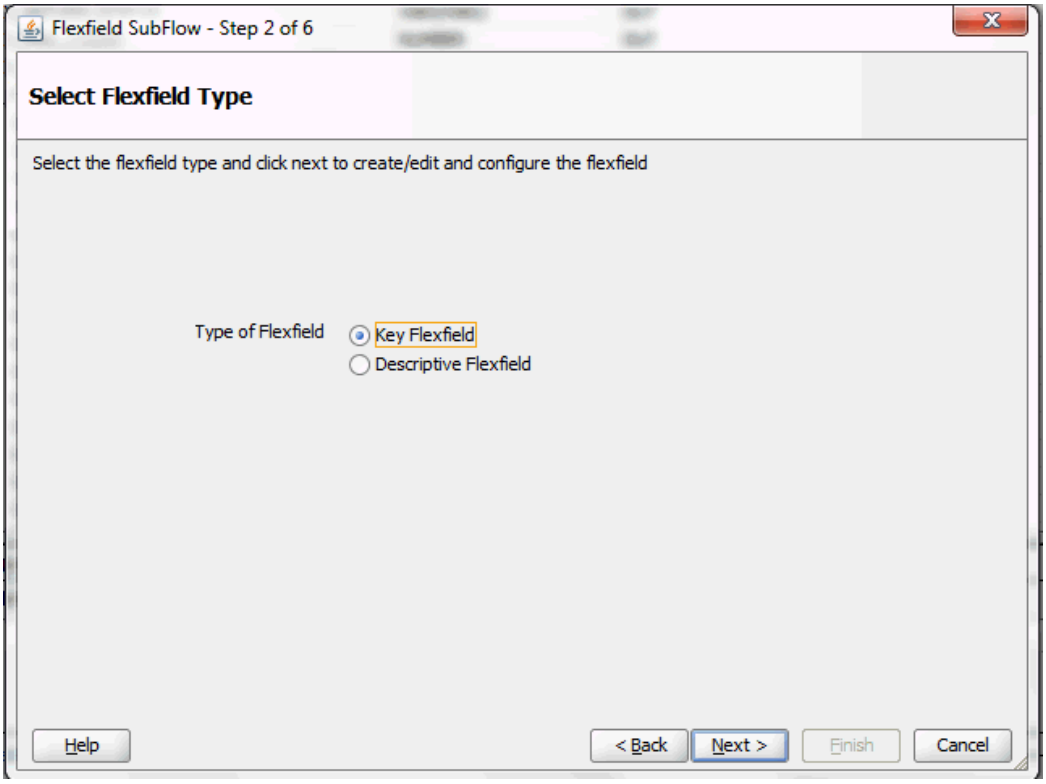
Configuring a Key Flexfield Mapping As Part of the Partner Link Creation

In this example, use key flexfields for a specific order needs (such as a special account number, its associated account name, and so on) for the selected open interface table `ONT.OE_HEADERS_IFACE_ALL`.

To configure a key flexfield mapping, click **Add Mapping** in the Configure Flexfields region. A series of flexfield selection pages in the wizard appear guiding you to configure a new mapping for your selected open interface table.

Click **Next** in the Welcome page. The Select Flexfield Type page is displayed.

Flexfield Subflow - Select Flexfield Type

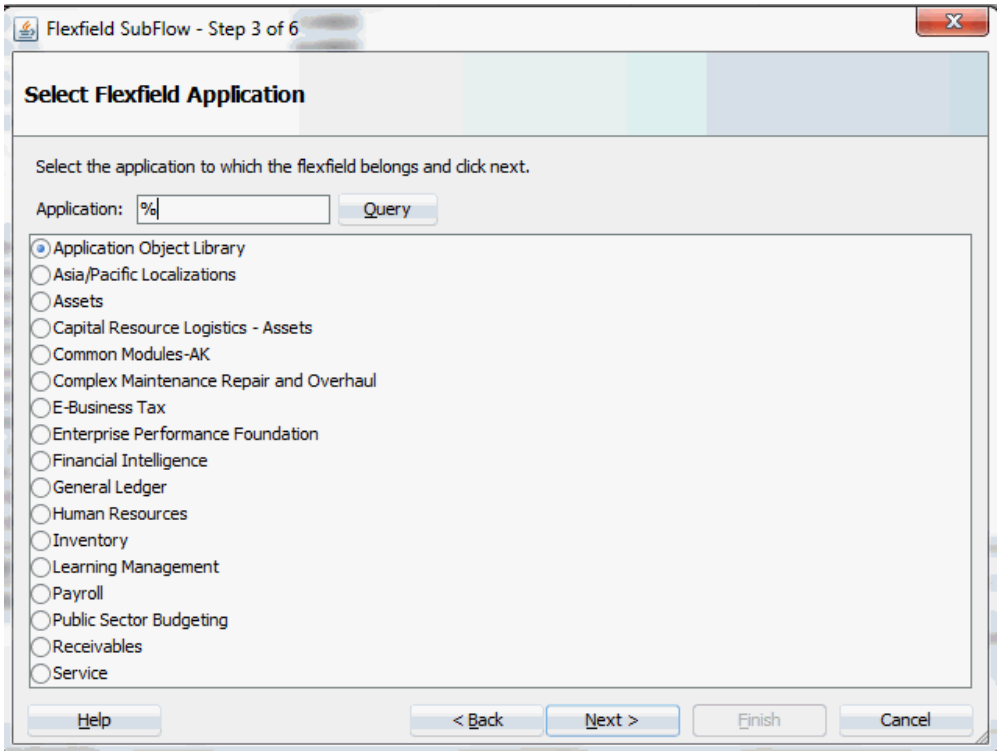


Perform the following steps to configure a key flexfield:

1. Select the **Key Flexfield** radio button in the Select Flexfield Type page. Click **Next**. The Select Flexfield Application page appears.

2. To locate your desired application name, enter keyword search (such as '%App%') and click **Query** to execute the search. All the entries that match your search criteria will be displayed. For example, select the **Application Object Library** radio button as the application name to which the flexfield belongs.

Flexfield Subflow - Select Flexfield Application

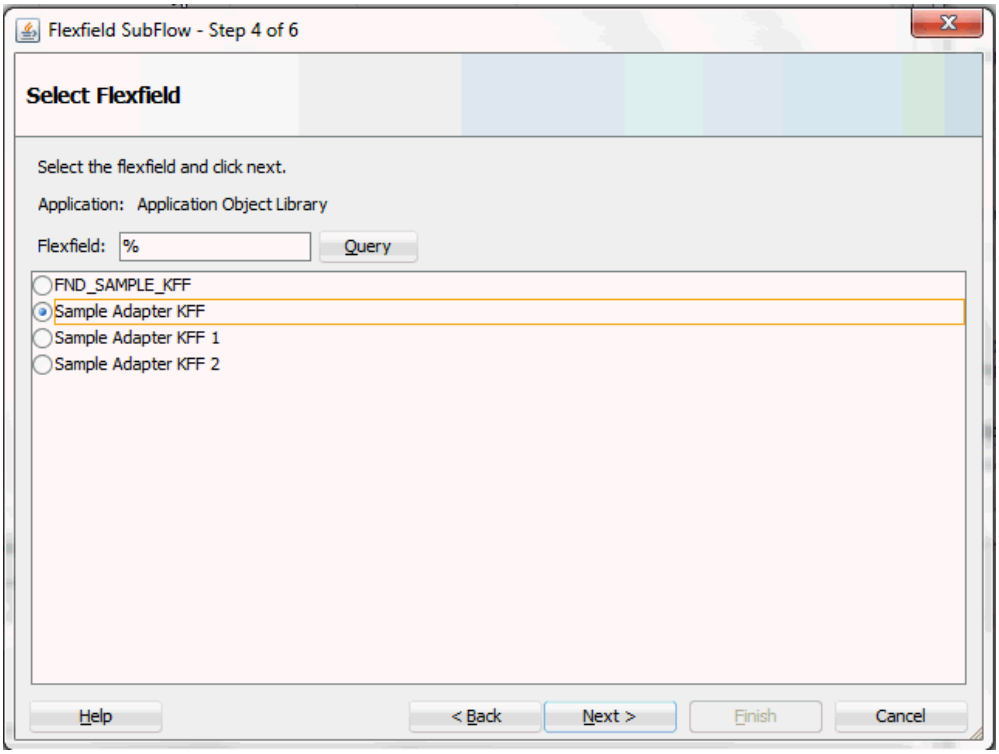


Alternatively, click **Query** directly to execute the blank search without keyword search. This displays all flexfield application names for your selection.

Click **Next**.

3. In the Select Flexfield page, enter keyword search if desired and click **Query** to execute the search. All the matched key flexfields are displayed for your selection. Select your desired key flexfield radio button.

Flexfield Subflow - Select Flexfield



Click **Next** to continue.

4. The Configure Mapping page appears where you can choose desired mapping for your key flexfield. Notice that your selected key flexfield name is automatically populated as the Flexfield name.

   In this mapping page, you have the option to do the mapping manually or use automapping feature.

   - Automapping feature maps all the flexfield segments to the Open Iinterface Table columns, which have the column name as part of the parameter name. For example, flexfield table column SEGMENT1 would be mapped to interface table column TP_SEGMENT1. If there are multiple sets of interface table columns and the flexfield table column names, then the prefix can be specified.

     For example, enter 'TP_' as the prefix to retrieve all interface table columns starting with 'TP_', such as TP_SEGMENT1, TP_SEGMENT2, TP_SEGMENT3, to TP_SEGMENTn where n is a number. If you click **Auto Map**, all the interface table columns with prefix 'TP_' are automatically displayed along with the automapped flexfield table column names.

     Note: Flexfield values are stored in the application database

tables. In general, key flexfields store their data in columns called SEGMENTn and descriptive flexfields store their data in columns called ATTRIBUTEn, where n is a number.

- Manual mapping lets you select a desired flexfield table column for a corresponding interface table column.

  Note: If an interface table column has already mapped to a flexfield table column, you cannot remap that column to other flexfield table column. In this situation, 'Mapped to other Flexfield' appears instead for that mapped interface table column.

Click **Clear Mapping** to clear current mapping you have. All previously mapped columns marked with 'Mapped to other Flexfield' will not be removed from the system.

Click **Next** to continue.

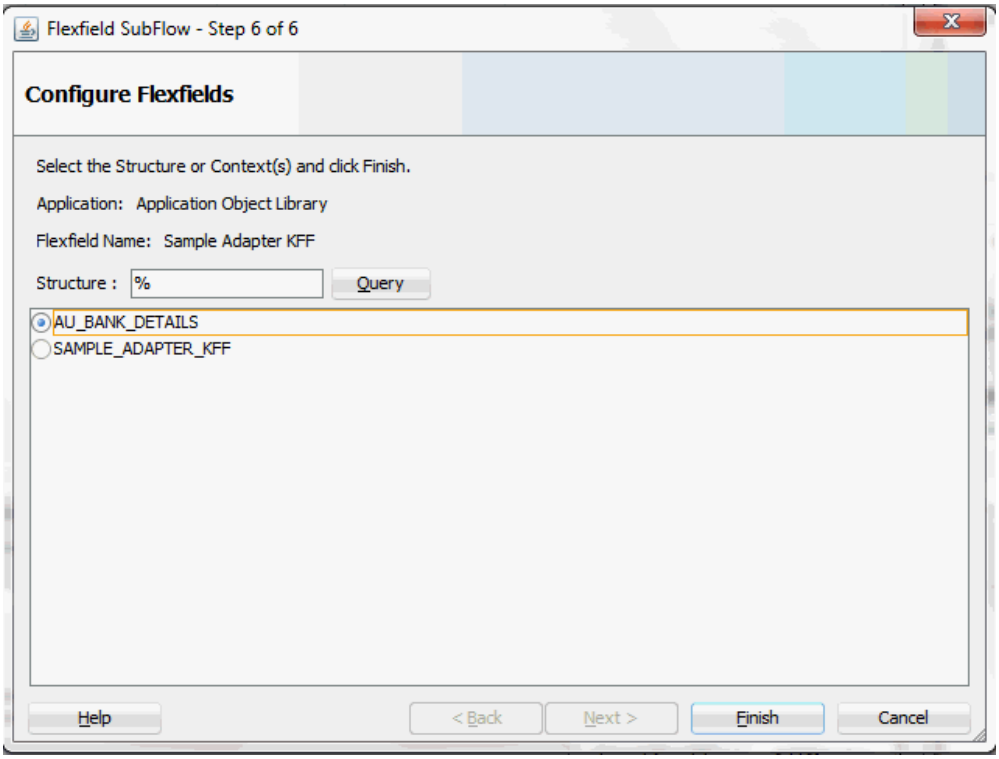
5. The configuration of the flexfield will not be complete without a structure for a key flexfield and at least one context value for a descriptive flexfield.

   In the Configure Flexfields page, enter keyword search if desired in the Structure field or simply click **Query** to execute a blank search. All the matched structures for the selected key flexfield are displayed.

   Select only one structure (such as AU_BANK_DETAILS).

   Please note that there are certain criteria need to be met in order to have your desired structures displayed in the wizard for selection. See: Guidelines for Defining Flexfield Structure and Context Values, page 4-34.

Flexfield Subflow - Configure Flexfields



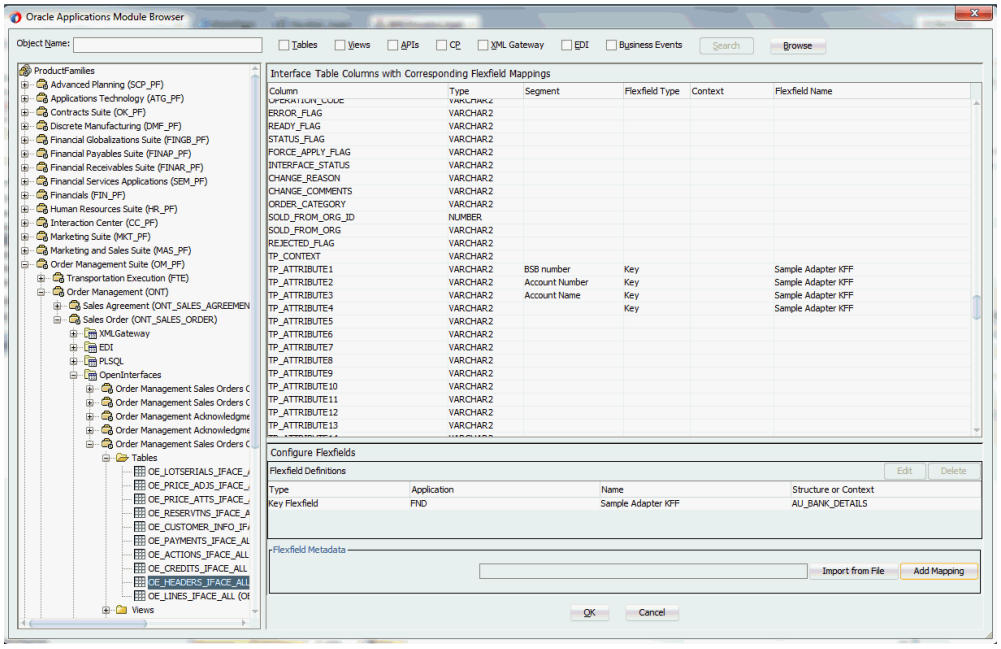
Note: Only one structure for a key flexfield can participate in flexfield mapping. In the case of a descriptive flexfield, multiple context values can participate in the mapping.

Click **Finish**.

6. The Oracle E-Business Suite Module Browser window appears. The newly added key flexfield structure and mapped interface table columns are displayed in the Interface Table Columns with Corresponding Flexfield Mappings region.

   The key flexfield information is also displayed in the Flexfield Definitions section.

Displaying Selected Key Flexfield Information



Click **OK** if you complete your flexfield configuration. The Application Interface page appears with the selected open interface table.

In this example, we will cerate a descriptive flexfield mapping to this selected open interface table by clicking **Add Mapping**. See: Configuring Descriptive Flexfield Mappings for an Open Interface Table, page 4-85.

Configuring Descriptive Flexfield Mappings for an Open Interface Table

Based on the same business scenario described earlier, a special order applying for customers both in India and Japan needs to be inserted to `ONT.OE_HEADERS_IFACE_ALL` open interface table. After the special order data is inserted, a written output file containing those data is requested as part of a response message for the requestor.

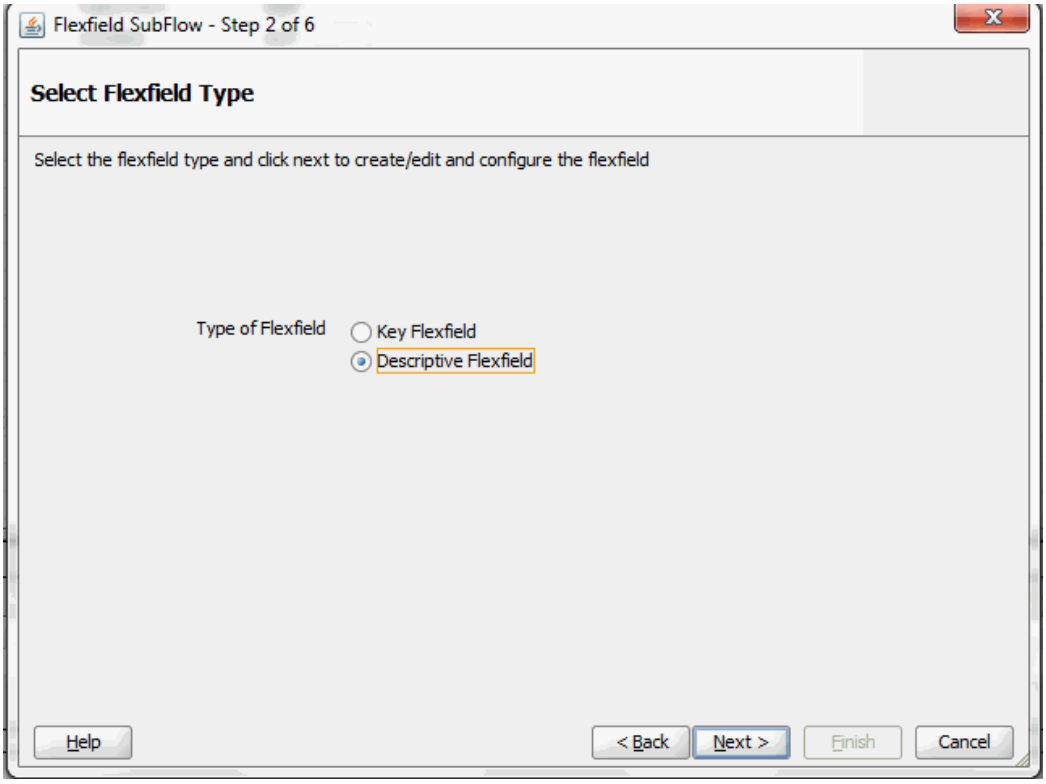
Configuring a Descriptive Flexfield Mapping As Part of the Partner Link Creation

In this example, descriptive flexfields are used to represent address requirements for customers located in different countries.

To configure a descriptive flexfield mapping for the selected open interface table `ONT.OE_HEADERS_IFACE_ALL`, click **Add Mapping** in the Configure Flexfields region. A series of flexfield selection pages in the wizard appear guiding you to configure a new mapping for your selected API.

Click **Next** in the Welcome page. The Select Flexfield Type page is displayed.

Flexfield Subflow - Select Flexfield Type



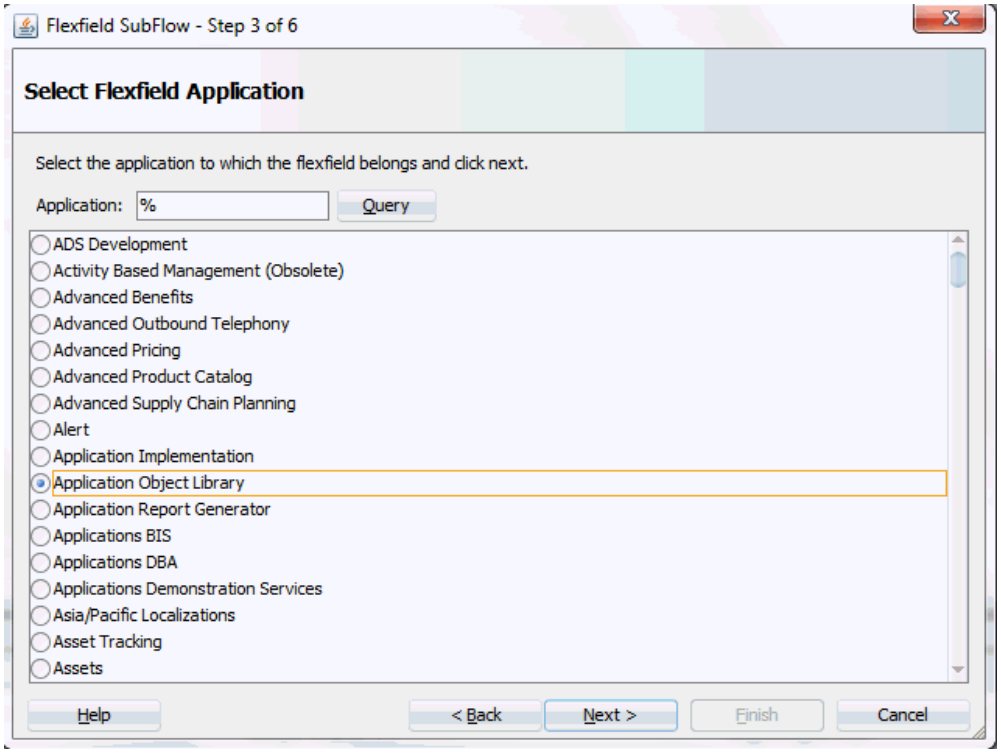
Perform the following steps to configure a descriptive flexfield:

1. Select the **Descriptive Flexfield** radio button in the Select Flexfield Type page, and then use the similar approach as described earlier for key flexfields in the flexfield mapping wizard.

2. Click **Next** in the Select Flexfield Type page. The Select Flexfield Application page is displayed.

3. To locate your desired application, enter keyword search (such as '%App%') or simply click **Query** to execute the search. All the matched applications for descriptive flexfield are displayed. For example, select the **Oracle Application Library** radio button as the application name to which the descriptive flexfield belongs.

Flexfield Subflow - Select Flexfield Application

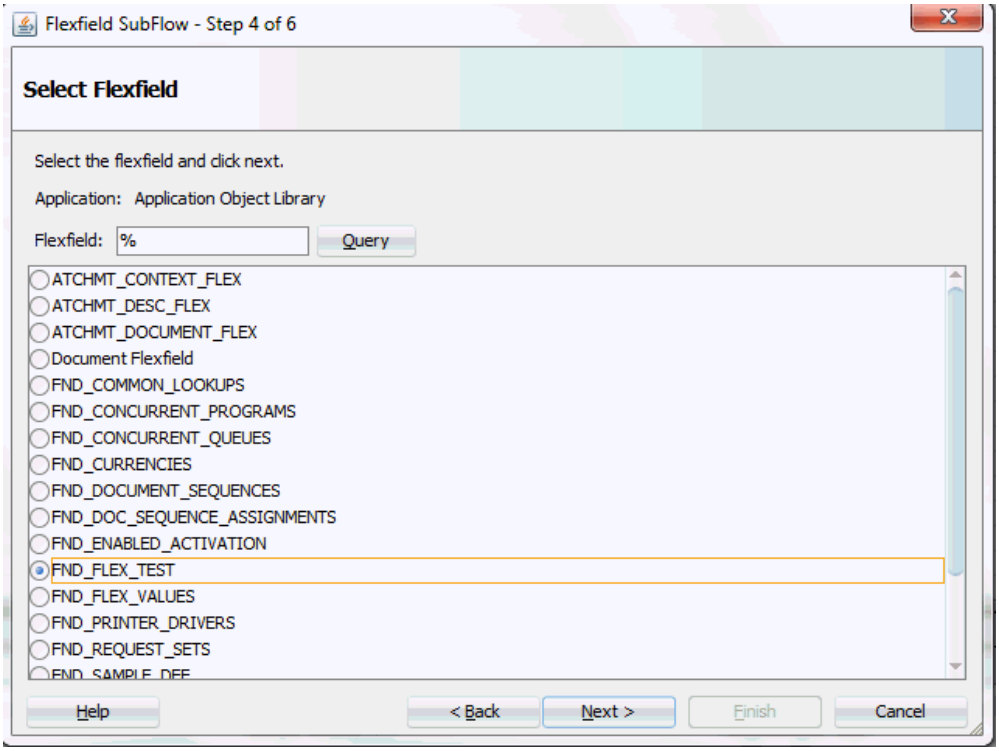


Click **Next**.

4. In the Select Flexfield page, enter keyword search (such as 'FND%') if desired and click **Query** to execute the search. All the matched descriptive flexfields are displayed for your selection.

   Select your desired descriptive flexfield (such as FND_FLEX_TEST) and click **Next**.

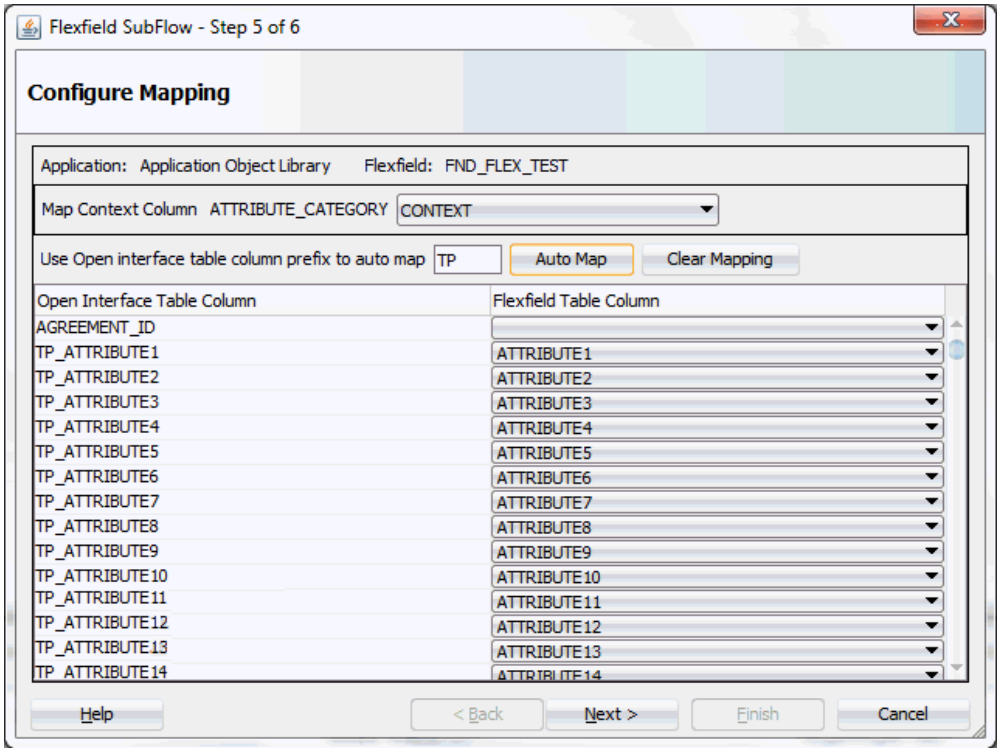
Flexfield Subflow - Select Flexfield



5. The Configure Mapping page appears where you can choose desired mapping for your descriptive flexfield.

   Notice that your selected descriptive flexfield name FND_FLEX_TEST is automatically populated as the Flexfield name.

Flexfield Subflow - Configure Mapping

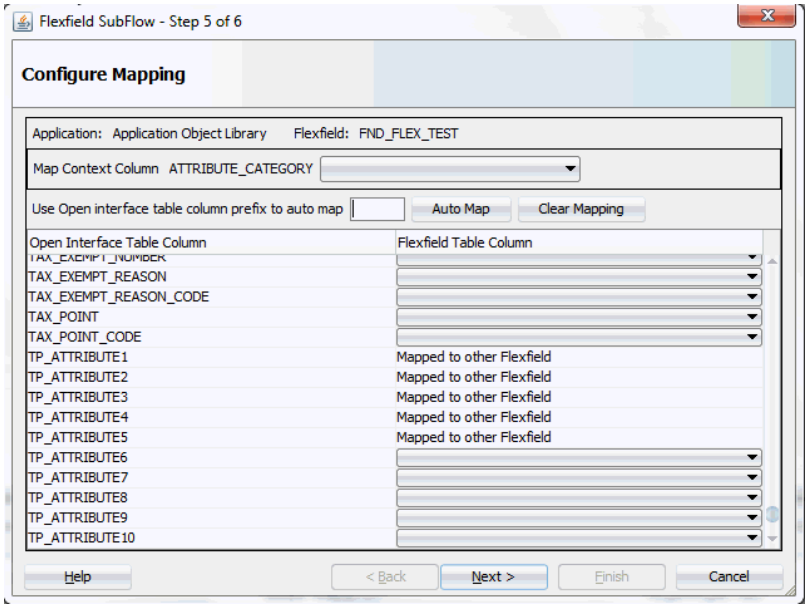


Selecting Context Column for Descriptive Flexfield Mapping

For descriptive flexfields, you need to perform an additional mapping for the context column. This is because flexfield mapping cannot span across open interface tables. All the flexfield table columns including the context column should be mapped to the open interface table columns in the OIT table.

In the Map Context Column ATTRIBUTE_CATEGORY field, select a desired category name that you want to view the mapping information from the drop-down list. For example, select CONTEXT from the drop-down list.

> **Note:** If an open interface table column has already mapped to a flexfield table column, you cannot remap that column to other flexfield table column. In this situation, 'Mapped to other Flexfield' appears instead for that mapped open interface table column.



Similar to key flexfields, the mapping can be performed manually or assisted with the automapping feature by clicking **Auto Map**.

Automapping feature maps all the flexfield table columns to the open interface table columns, which have the column name as part of the interface table column name. For example, flexfield table column names from ATTRIBUTE1 to ATTRIBUTE20 are automatically displayed and mapped to the open interface table columns from ATTRIBUTE1 to ATTRIBUTE20 respectively through automapping.

If there are multiple sets of open interface table columns and the flexfield table column names, then the prefix needs to be specified for automapping. This would map the potential open interface table columns which need to be considered for the mapping. Enter table column prefix (such as 'TP') to first locate your desired interface table columns that you want the automapping to be performed. Next, click **Auto Map**. All the interface table columns with prefix 'TP' are automatically displayed along with the automapped table column names. Interface table column 'TP_ATTRIBUTE1' is mapped to column ATTRIBUTE1, and interface table column 'TP_ATTRIBUTEn' is mapped to column ATTRIBUTEn.

Instead of using automapping feature, you can manually select a desired flexfield table column name from the drop-down list for a corresponding interface table column for your flexfield mapping.

Click **Clear Mapping** to clear current mapping you have in the flexfield table column only. All previously mapped columns marked with 'Mapped to other Flexfield' will not be removed from the system.

Click **Next** to proceed to the next page.

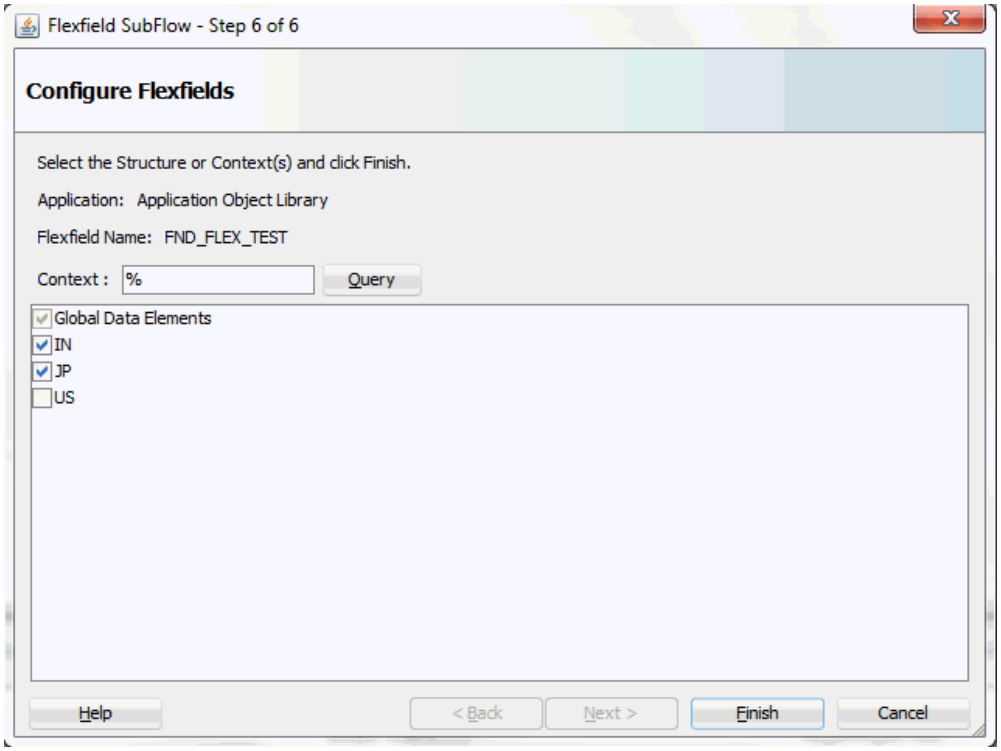
6. In the Configure Flexfields page, select at least one context value for the selected descriptive flexfield. For example, select desired context values (such as 'IN' and 'JP').

   You can either enter keyword search in the Context field or simply click **Query** to execute the search before selection.

   For descriptive flexfields, context value should be available to the run time through an open interface table column.

   Please note that there are certain criteria need to be met in order to have your desired context values displayed in the wizard for selection. See: Guidelines for Defining Flexfield Structure and Context Values, page 4-34.

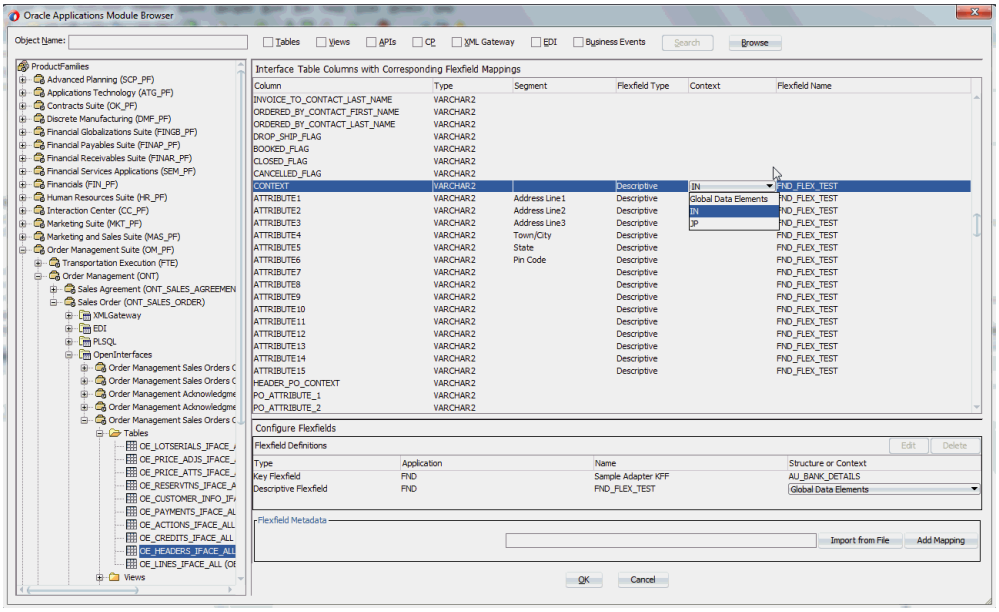
   Flexfield Subflow - Configure Flexfields



   Click **Finish** to complete the flexfield configuration.

7. The Oracle E-Business Suite Module Browser window appears. The newly added descriptive flexfield information including the segment values for the selected context value is displayed in the right pane of the window.

Displaying Selected Descriptive Flexfield Information



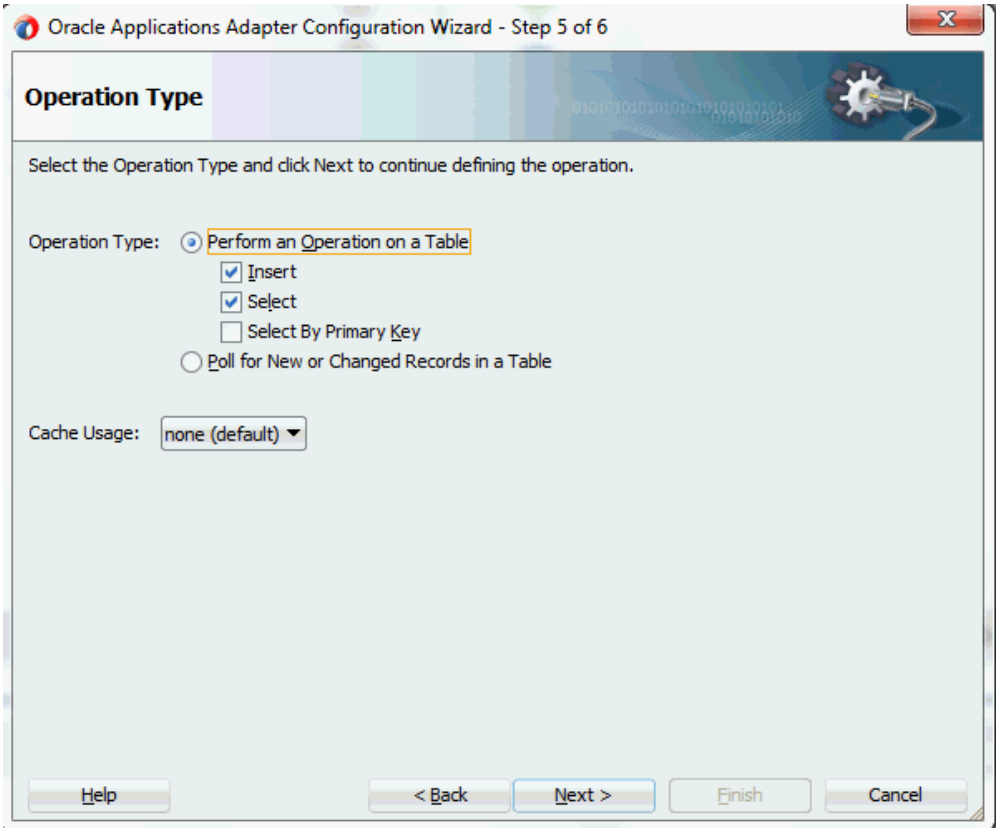
Click **OK**.

8. The Application Interface page appears with the selected open interface table.

Click **Next**.

9. The Operation Type page is displayed.

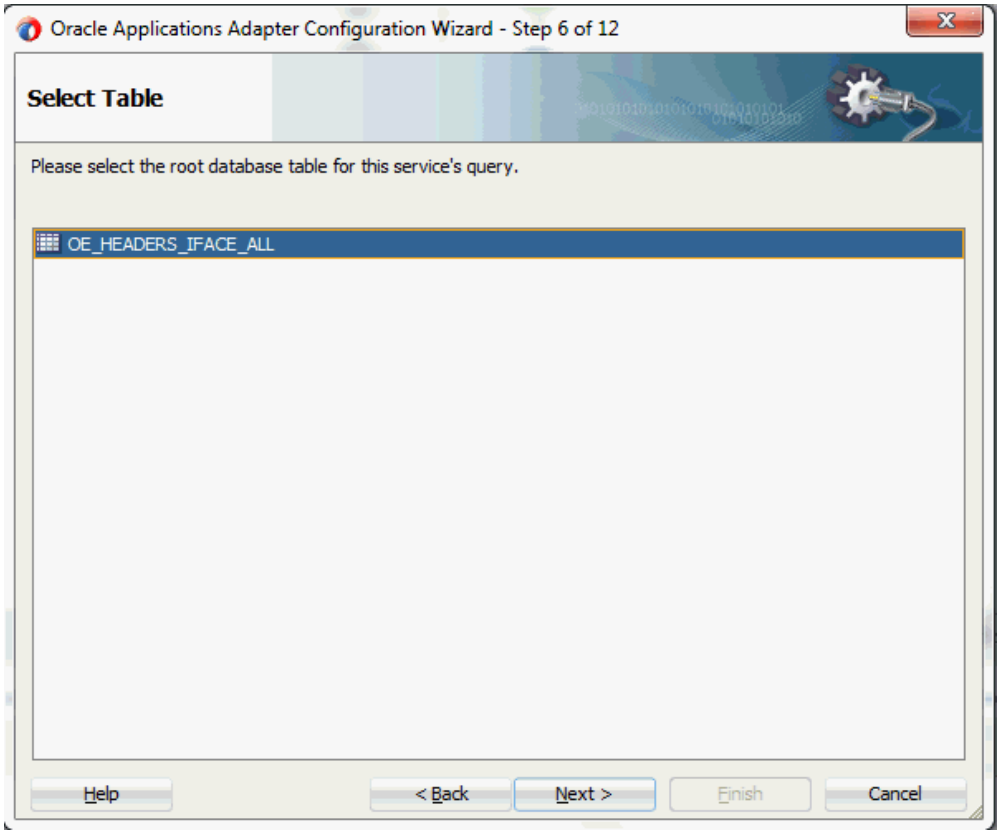
Adapter Configuration Wizard - Operation Type Page



Select the **Perform an Operation on a Table** radio button and then choose the **Select** and **Insert** check boxes.

10. Click **Next**. The Select Table page appears which displays the tables that have been previously imported in this JDeveloper project. Select the OE_HEADER_IFACE_ALL table.

Adapter Configuration Wizard - Select Table Page



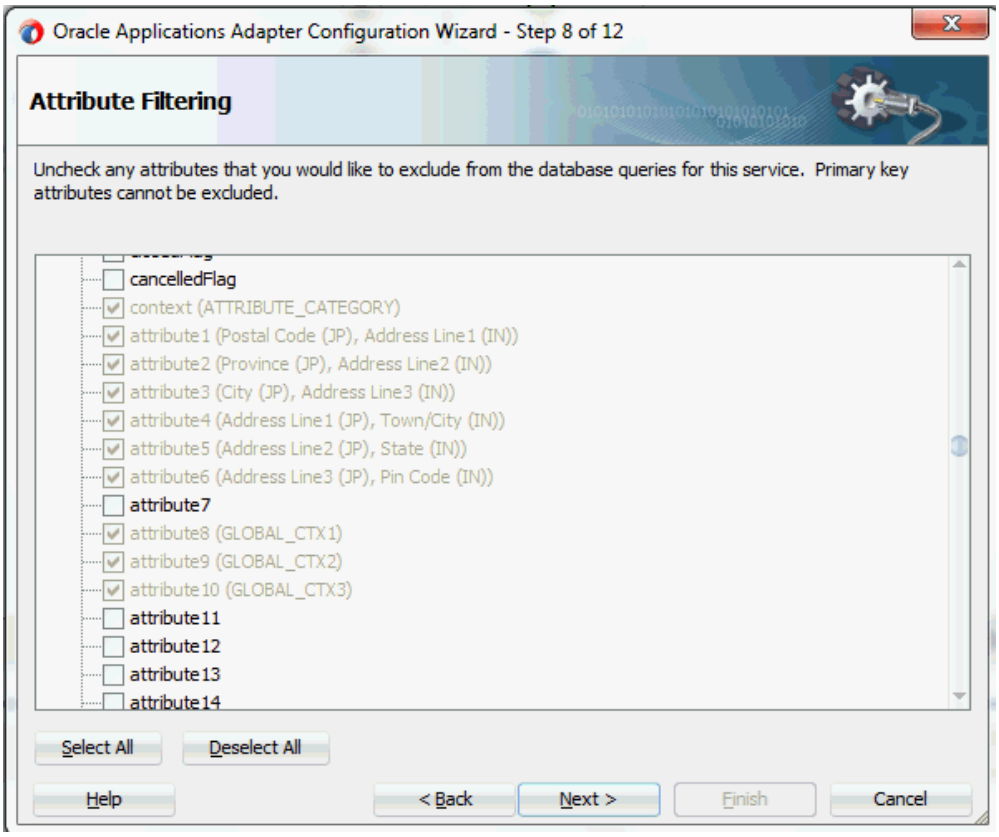
11. Click **Next**. The Relationships page is displayed.

12. Click **Next**. The Attribute Filtering page appears.

    Click **Deselect All** and select the **orderSourceId** check box only.

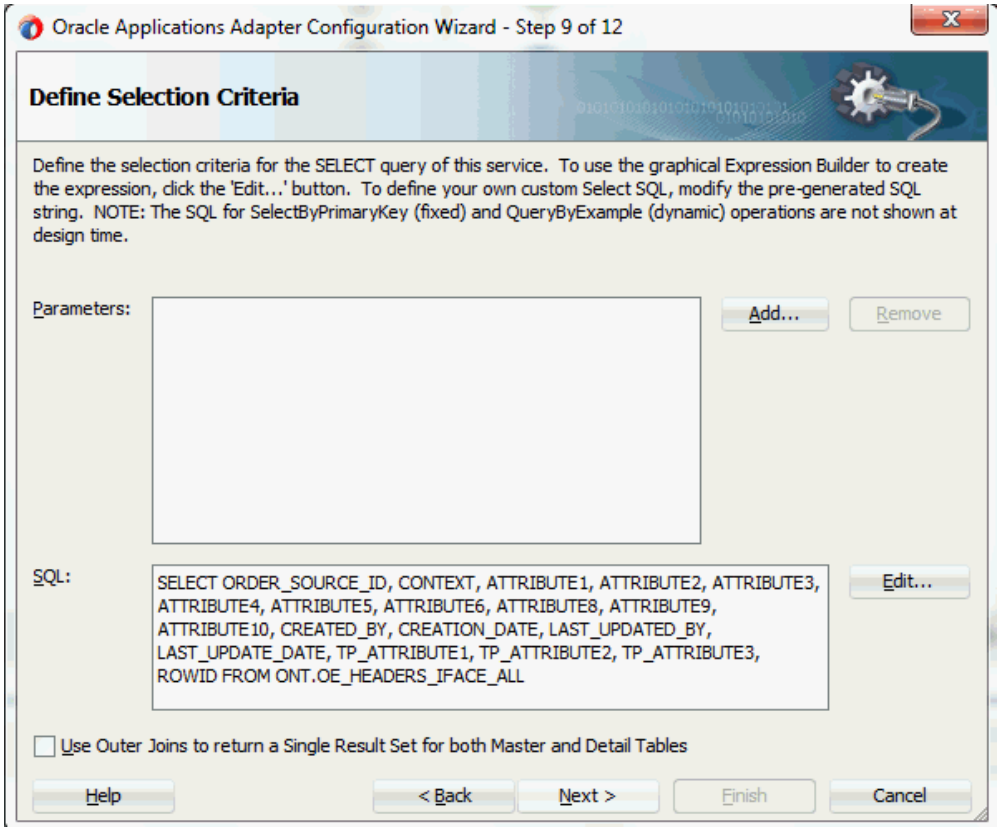
    The mapped flexfield table column names we defined earlier for both key and descriptive flexfields are already selected in this page. However, they appear as grayed-out attributes indicating that you cannot deselect them.

Adapter Configuration Wizard - Attribute Filtering Page with Mapped Flexfield Column Names



13. Click **Next**. The Define Selection Criteria page is displayed with SQL statements populated.

Adapter Configuration Wizard - Define Selection Criteria Page



14. Click **Next**. The Advanced Options page is displayed.

15. Click **Next**. The JCA Endpoint Properties page is displayed.

16. Click **Next** and then click **Finish**.

    The wizard generates the WSDL file corresponding to the partner link. An enhanced XSD file, for example `AppsReference_flex.xsd`, is also created. This XSD file contains the schema describing the procedure arguments with the elements representing the interface table columns that have been replaced with flexfield segment names.

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

17. Additionally, configuration file and flexfield mapping file are created. For more information on these files, see Reviewing Flexfield Configurations, page 4-122.

Once a new flexfield mapping has been configured or added, if modification is needed, you can make updates by first selecting desired flexfield type that you want to modify and then make needed changes. For more information on how to modify the selected mapping details, see Modifying Flexfield Mapping Definitions, page 4-119.

Displaying Mapped Key and Descriptive Flexfields at Design Time

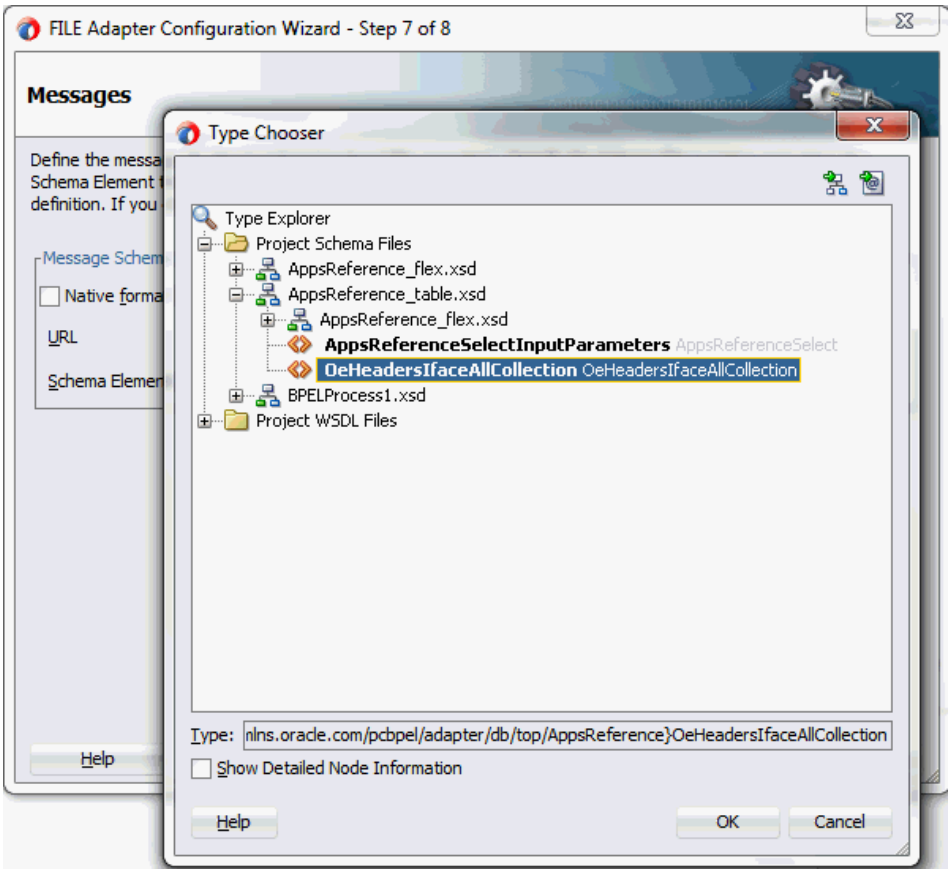
Both key and descriptive flexfield mapping data configured earlier during the partner link creation can be shown at design time along with the following activities orchestrated in the BPEL process within in a SOA composite:

A Composite Example with Key and Descriptive Flexfield Mappings

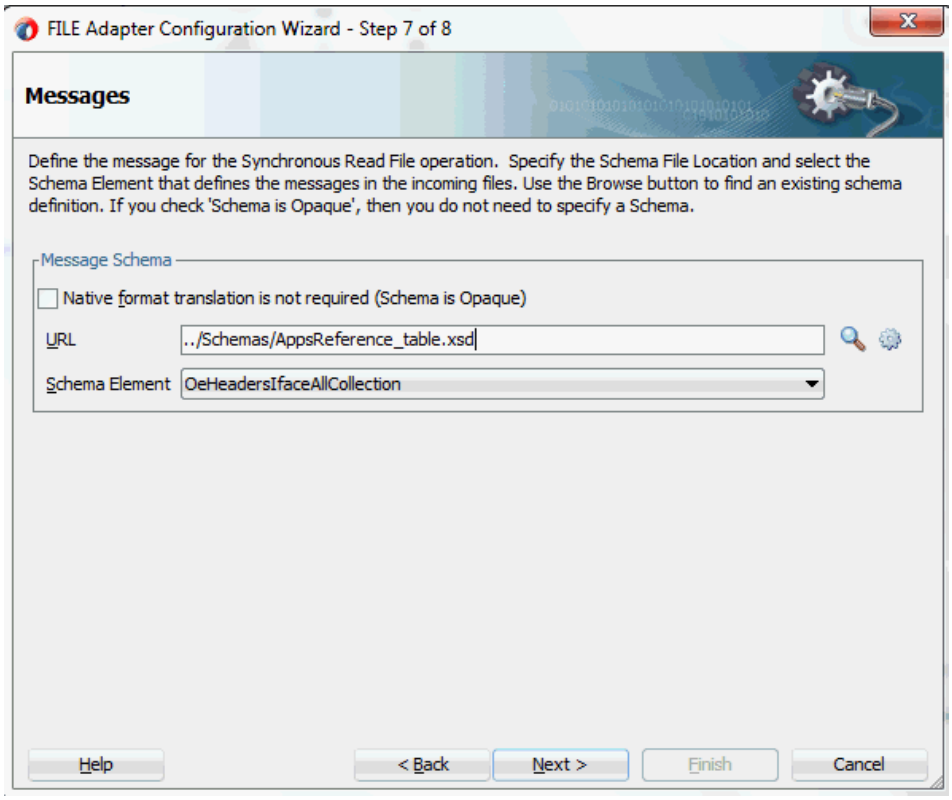
Here is an example describing both key and descriptive flexfield mappings in the BPEL process of a SOA composite:

- Create a new SOA composite application with BPEL process
- Add a partner link service called `AppsReference` with the key and descriptive flexfield mappings that we configured earlier for the `ONT.OE_HEADERS_IFACE_ALL` open interface table.
- Add a partner link for File Adapter `WriteSelectOutput` to write the inserted order data as an output file in XML format.
  - In the File Directories dialog, enter the physical directory (such as `/usr/tmp`) where the output xml file (such as `output_payload.xml`) resides.
  - In the Messages dialog, select the 'browse for schema file' icon next to the URL field to open the Type Chooser.

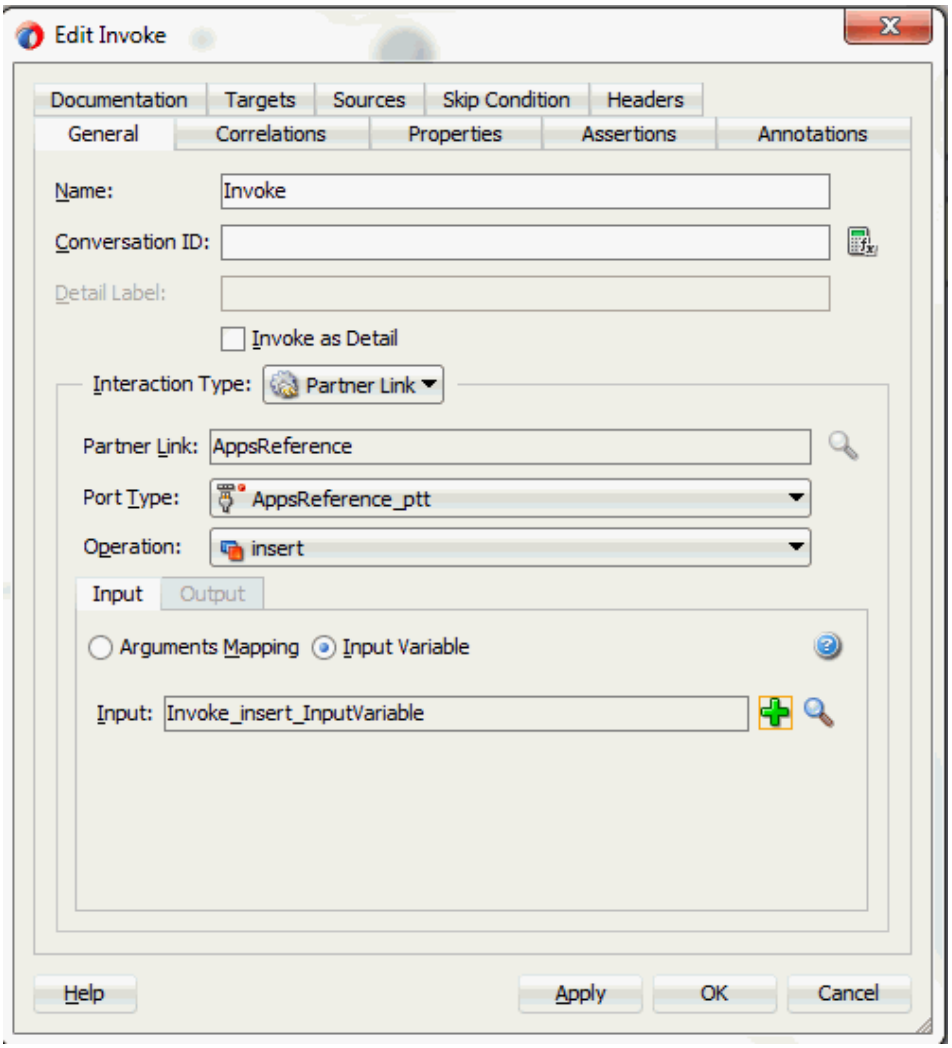
    Click Type Explorer and select **Project Schema Files > AppsReference_table.xsd > AppsReference_flex.xsd > OeHeaderIFaceAllCollection**.



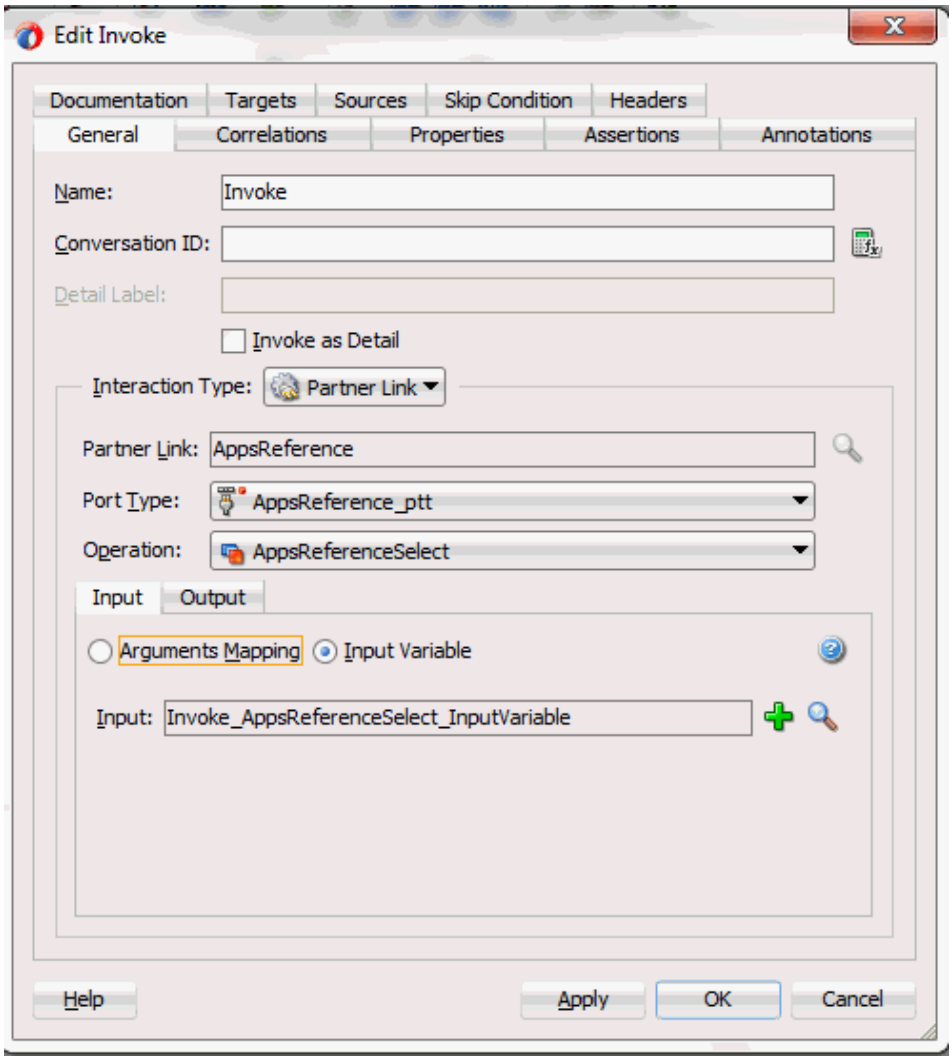
Click **OK**. The selected schema information will be automatically populated in the URL and Schema Element fields.



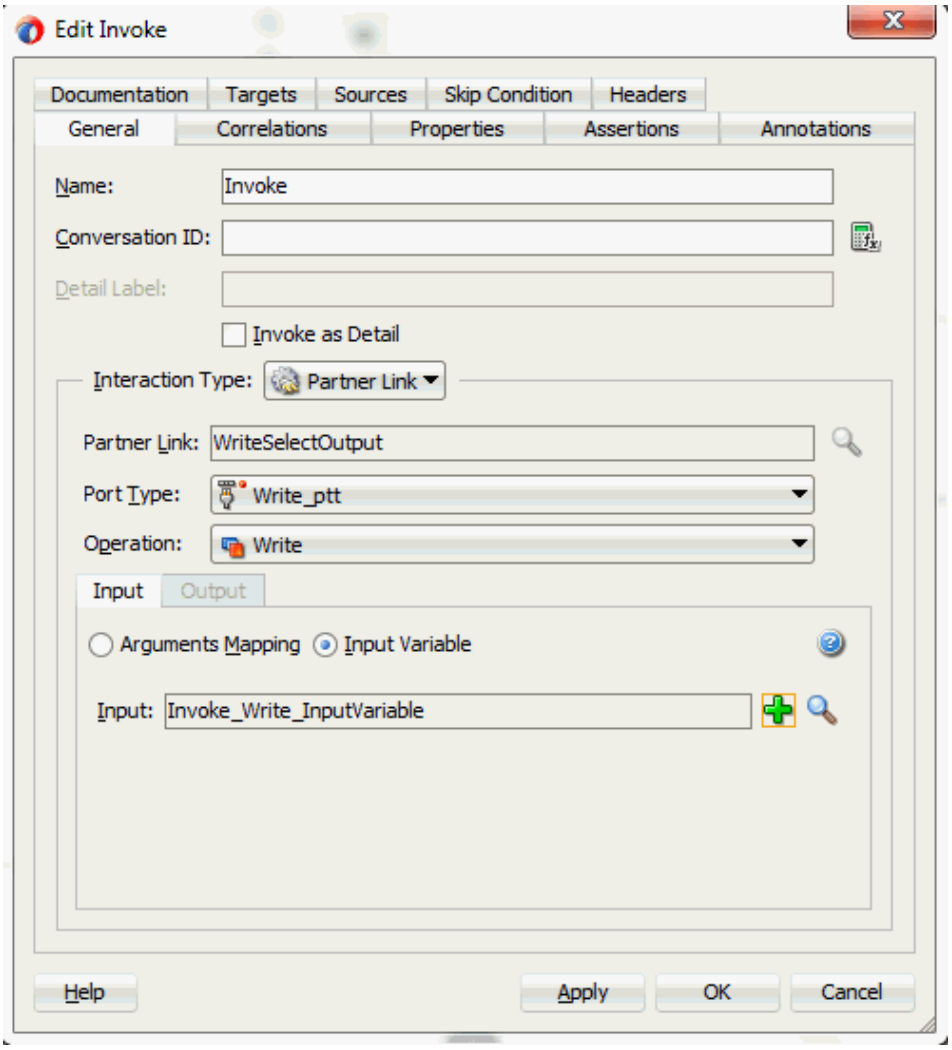
- Configure three **Invoke** activities
  - Associate the first **Invoke** activity with the partner link `AppsReference` to insert data to the `ONT.OE_HEADERS_IFACE_ALL` open interface table columns.



- Associate the second **Invoke** activity with the partner link `AppsReference` that contains flexfield mapping to invoke the service with selected order data. Ensure to click the Output tab to specify the output variable.



- Associate the third **Invoke** activity with the File Adapter partner link `WriteSelectOutput` to write the output file.



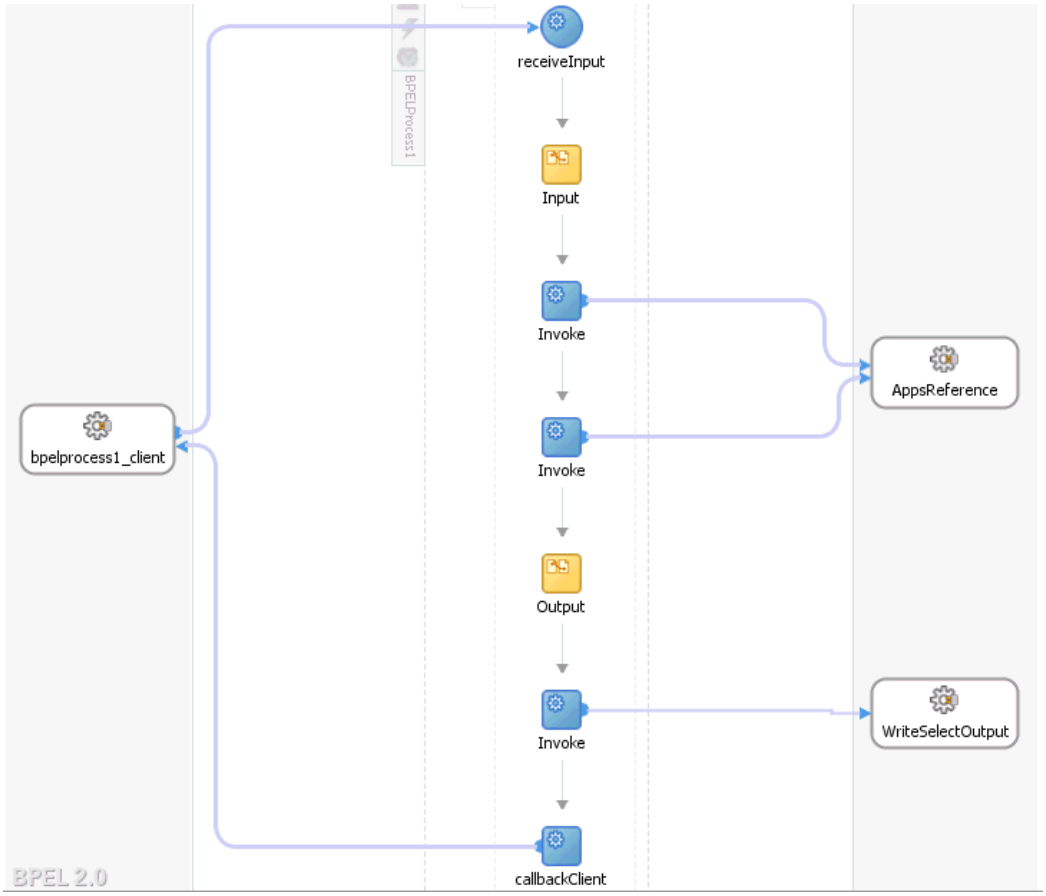
- Add two **Assign** activities:

  1. To pass order data containing both key and descriptive flexfields defined earlier as part of the input payload to the first **Invoke** activity to invoke the service `AppsReference`.

     Mapped key and descriptive flexfield column names corresponding to the flexfield matching segments defined in the Oracle E-Business Suite are displayed in the **Assign** activity. See: Assigning Parameters with Key and Descriptive Flexfields in the Assign Activity, page 4-107.

  2. To pass the output details from the service invocation as input data to the File Adapter.

Constructing BPEL Process in SOA Composite



For more information on how to create design time tasks, see Design-Time Tasks for Open Interface Tables, page 8-3.

Flexfield Validation in the Schema File

Select the enhanced XSD file (such as `AppsReference_flex.xsd`) in Oracle JDeveloper and notice that the mapped key flexfields are displayed with type 'flex' under the Sample_Adapter_KFF, and the mapped descriptive flexfields are displayed with type 'flex' under the FND_FLEX_TEST flexfield as part of the schema.

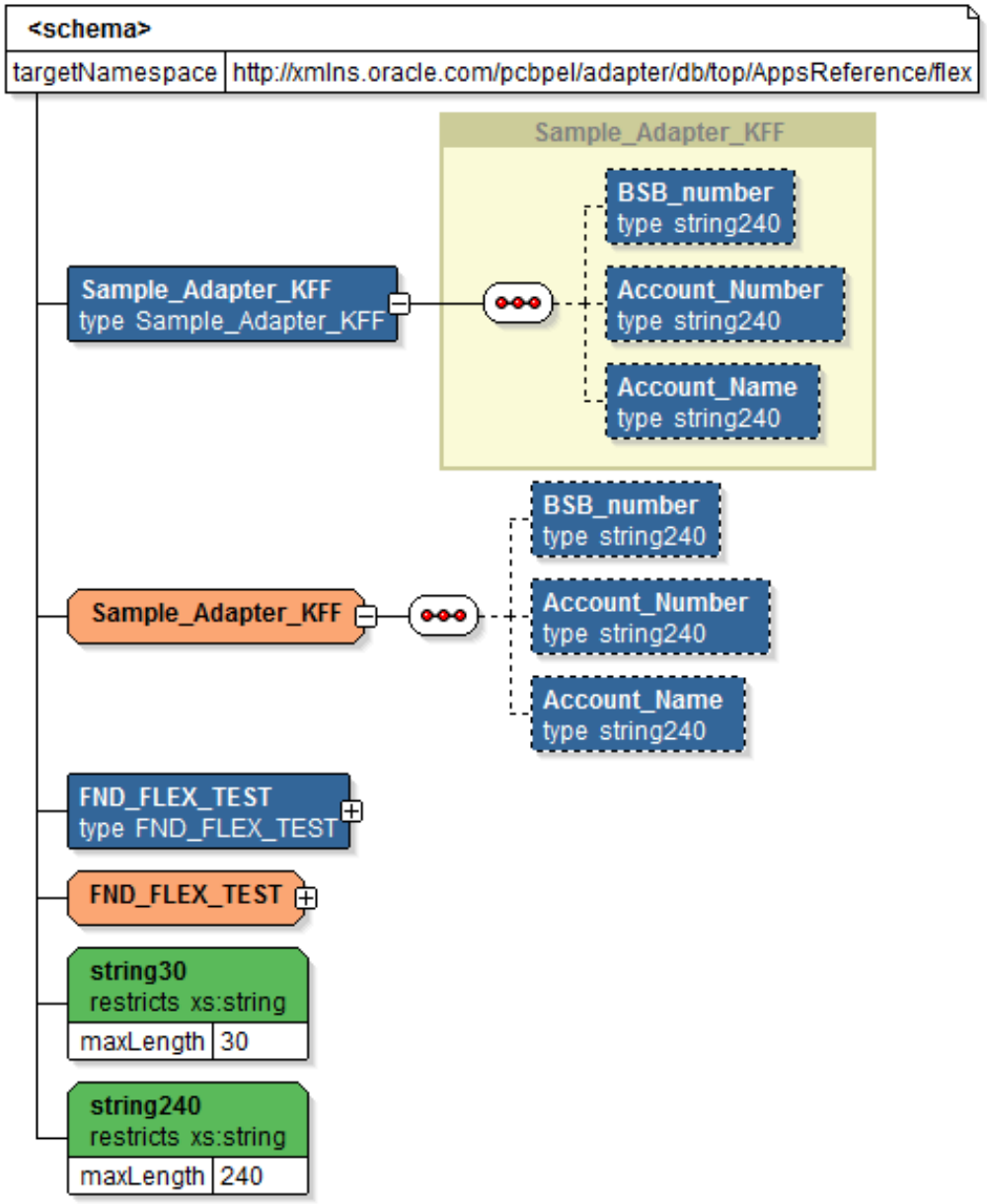
For key flexfields, the following table explains the relationship between the key flexfield mapping data and the actual segments defined in Oracle E-Business Suite:

Segment Summary for Key Flexfield 'Sample_Adapter_KFF'

Key Flexfield Segments Mapped During Flexfield Configuration	Key Flexfield Segments Defined in Oracle E-Business Suite
SEGMENT1	BSB_Number
SEGMENT2	Account_Number
SEGMENT3	Account_Name

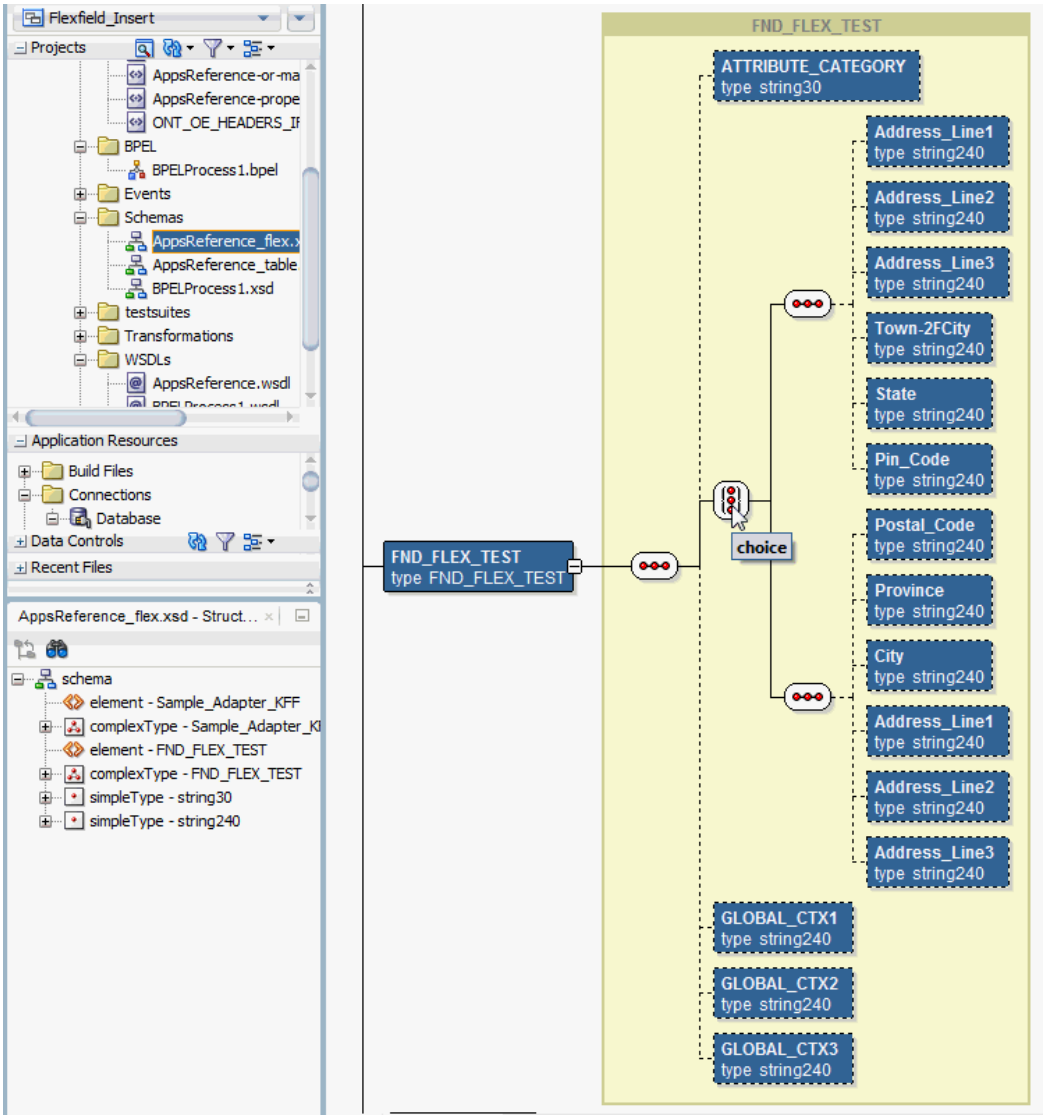
Mapped segment column names for the key flexfield 'BSB_Number', 'Account_Number' and 'Account_Name' are displayed in the Oracle E-Business Suite Module Browser once the mapping is complete.

Mapped Key Flexfield Segments Displayed in the Schema File



For descriptive flexfields, the mapped table column names `Address_Line1`, `Address_Line2`, `Address_Line3`, `Town/City`, `State`, and `Pin_Code` are grouped by context value `IN`, and `Postal_Code`, `Province`, `City`, `Address_Line1`, `Address_Line2`, and `Address_Line3` are grouped by context value `JP` listed in a tree structure. Context value nodes are connected with the **<choice>** icon in between indicating that only either one of the context values and its associated table columns can be selected.

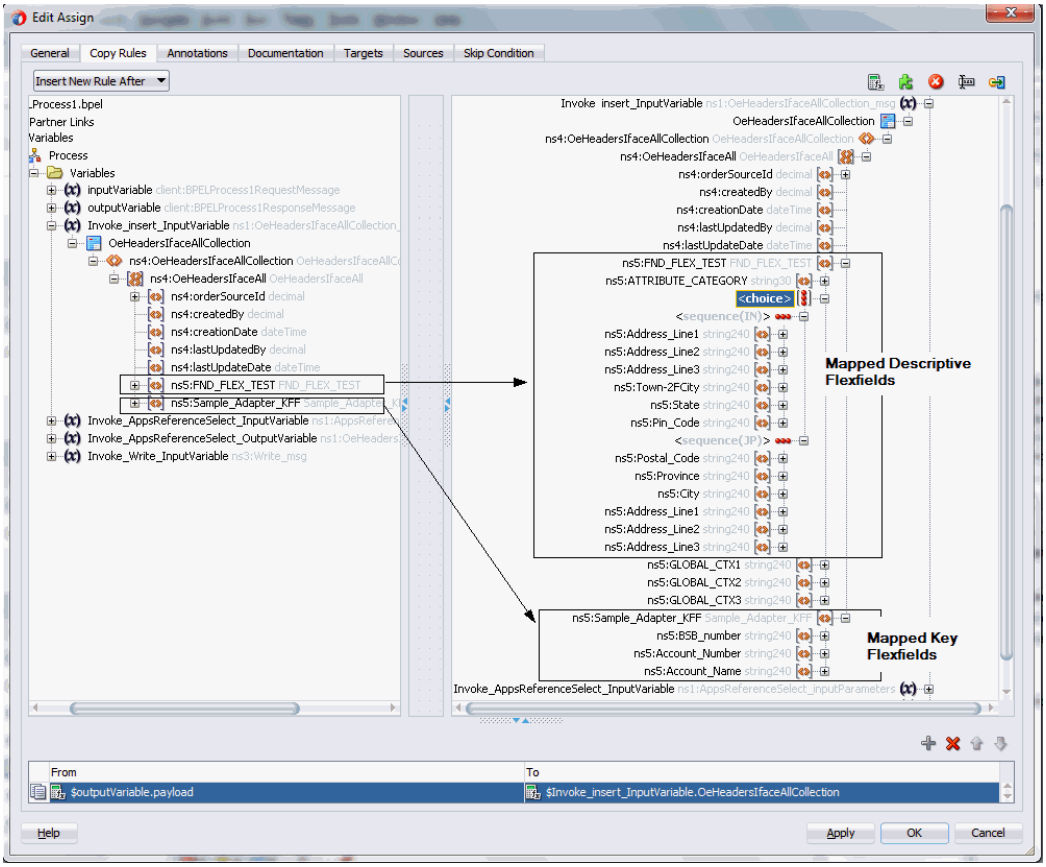
Mapped Descriptive Flexfield Segments Displayed in the Schema File



Assigning Parameters with Key and Descriptive Flexfields in the Assign Activity

In the **Assign** activity of the BPEL process diagram within the SOA composite application, the mapped key and descriptive flexfields can be displayed and used to pass an input variable received from a payload to the **Invoke** activity for `AppsReference` service invocation.

Displaying Mapped Key and Descriptive Flexfields in the Assign Activity



In the Assign activity, the **mapped key flexfields** are displayed under **Variables > Process > Variables > Invoke_Insert_InputVariable > OeHeadersIfaceAllCollection > ns4:OeHeadersIfaceAllCollection > ns4:OeHeadersIfaceAll > ns5:Sample_Adapter_KFF**.

The **mapped descriptive flexfieldd** are displayed under **Variables > Process > Variables > Invoke_Insert_InputVariable > OeHeadersIfaceAllCollection > ns4:OeHeadersIfaceAllCollection > ns4:OeHeadersIfaceAll > ns5:FND_FLEX_TEST**. Expand the **<choice>** node to locate the segment values based on selected context value, either IN or JP.

The target parameters `ns5:ATTRIBUTE1` was mapped to the following descriptive flexfield columns during the descriptive flexfield configuration earlier:

- `Address_Line1` (under context value `<sequence(IN)>`)
- `Postal_Code` (under context value `<sequence(JP)>`)

The target parameters `ns5:ATTRIBUTE2` was mapped to the following descriptive flexfield columns during the flexfield configuration earlier:

- Address_Line2 (under context value <sequence(IN)>)
- Province (under context value <sequence(JP)>)

The target parameters ns5:ATTRIBUTE3 was mapped to the following descriptive flexfield columns during the flexfield configuration earlier:

- Address_Line3 (under context value <sequence(IN)>)
- City (under context value <sequence(JP)>)

The target parameters ns5:ATTRIBUTE4 was mapped to the following descriptive flexfield columns during the flexfield configuration earlier:

- Town/City (under context value <sequence(IN)>)
- Address_Line1 (under context value <sequence(JP)>)

The target parameters ns5:ATTRIBUTE5 was mapped to the following descriptive flexfield columns during the flexfield configuration earlier:

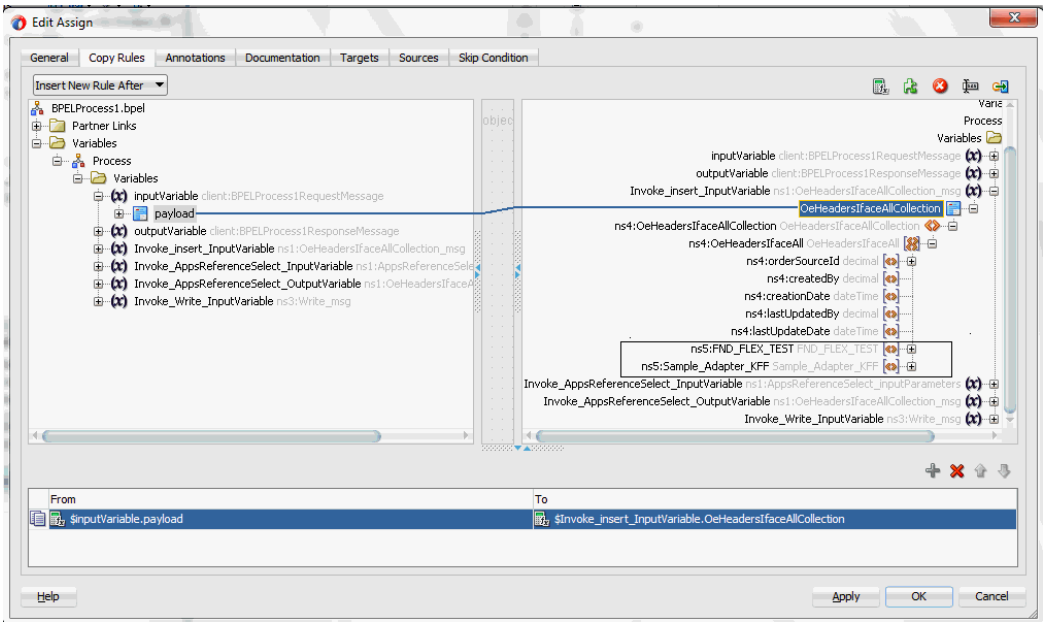
- State (under context value <sequence(IN)>)
- Address_Line2 (under context value <sequence(JP)>)

The target parameters ns5:ATTRIBUTE6 was mapped to the following descriptive flexfield columns during the flexfield configuration earlier:

- Pin_Code (under context value <sequence(IN)>)
- Address_Line3 (under context value <sequence(JP)>)

Assigning Variables

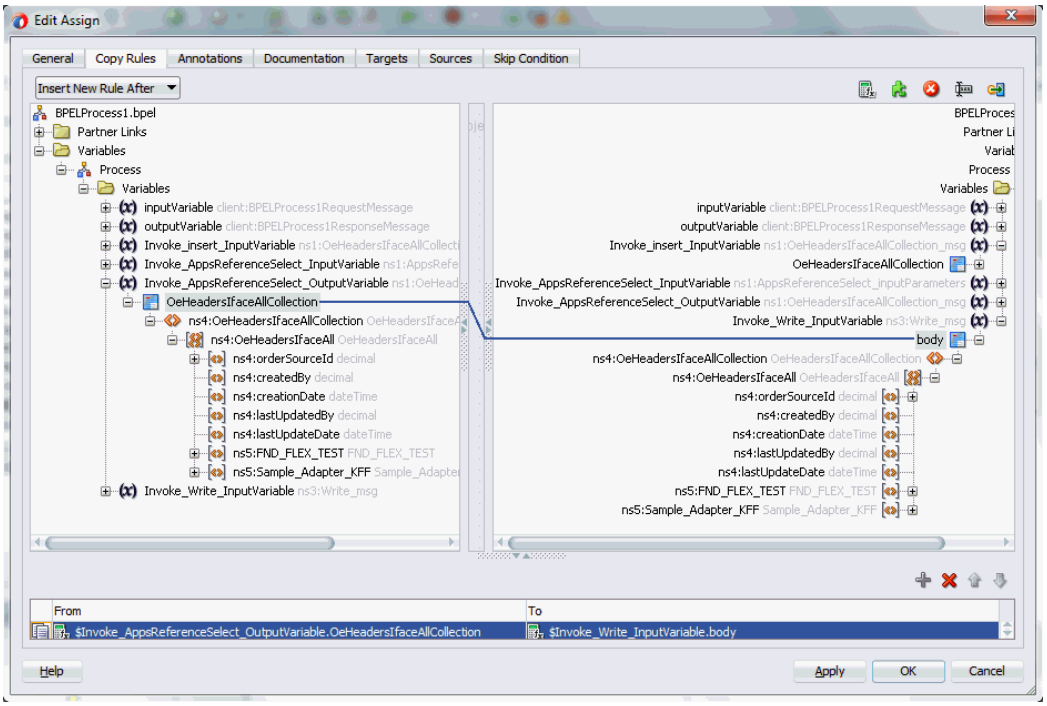
For the first **Assign** activity, use the following steps to assign the variables:



- In the From navigation tree, navigate to **Variables > Process > Variables > inputVariable** and select body.
- In the To navigation tree, navigate to **Variables > Process > Variables > Invoke_Insert_InputVariable** and select OeHeadersIfaceAllCollection

Drag the source node (body) to connect to the target node (OeHeadersIfaceAllCollection) that you just specified. This creates a line that connects the source and target nodes. The copy rule is displayed in the From and To sections at the bottom of the Edit Assign dialog box.

For the second **Assign** activity, use the following steps to assign the variables:



Since the OeHeadersIfaceAllCollection node variables contained in the Source and Target are identical, instead of specifying each flexfield, we can simply drag the source node (OeHeadersIfaceAllCollection) to connect to the target node (OeHeadersIfaceAllCollection) to complete the second **Assign** activity.

- In the From navigation tree, navigate to **Variables > Process > Variables > Invoke_AppsReferenceSelect_OutputVariable** and select OeHeadersIfaceAllCollection.

- In the To navigation tree, navigate to **Variables > Process > Variables > Invoke_Write_InputVariable** and select body.

Drag the source node (OeHeadersIfaceAllCollection) to connect to the target node (body) that you just specified. This creates a line that connects the source and target nodes. The copy rule is displayed in the From and To sections at the bottom of the Edit Assign dialog box.

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

Displaying Mapped Key and Descriptive Flexfields at Run Time

After deploying the BPEL process contained in a SOA Composite application to the Oracle WebLogic managed server (such as 'soa-server1'), the key and descriptive flexfields we configured earlier are displayed in the Oracle Enterprise Manager Fusion Middleware Control Console (`http://<servername>:<portnumber>/em`).

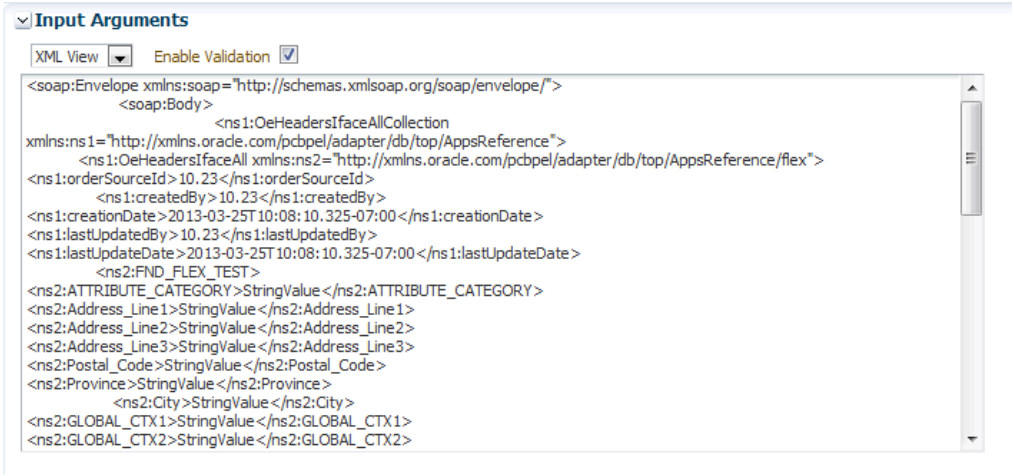
From the Farm base domain, expand the **SOA > soa-infra > soa-infra (soa-server1) > default** to navigate through the SOA Infrastructure home page and menu to access your

deployed SOA composite applications (such as 'Insert [1.0]').

Select 'Insert [1.0]' and click **Test** at the top of the page. The Test Web Service page for initiating an instance appears.

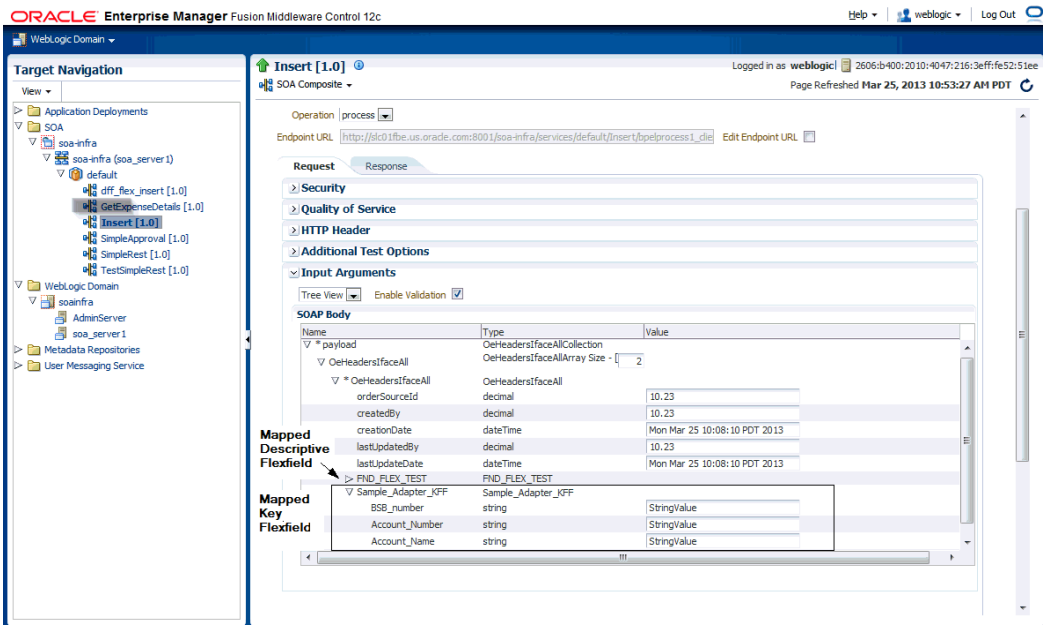
You can specify the XML payload data to use in the Input Arguments section.

Supplying XML Payload Data



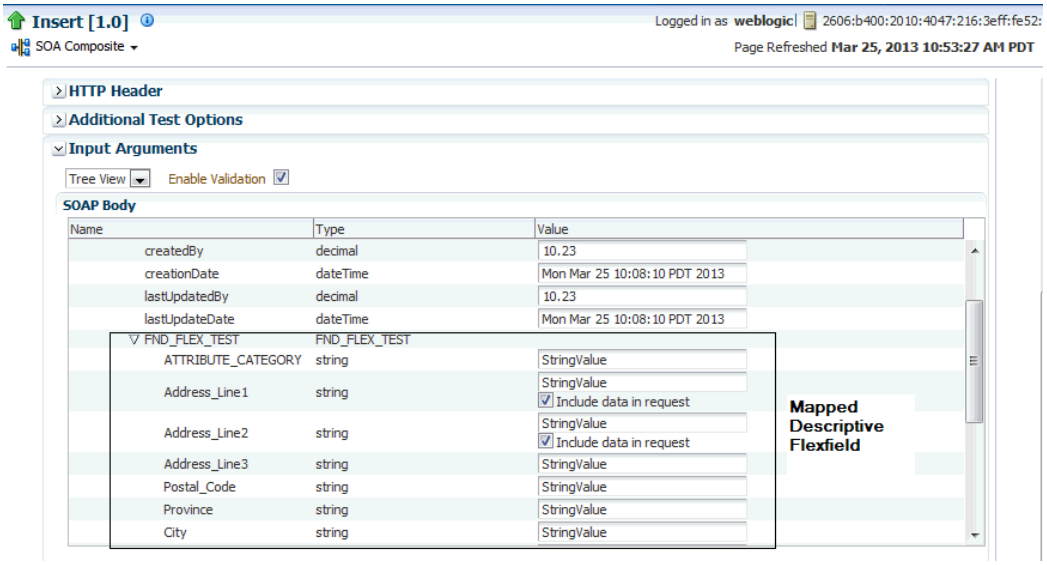
Both key and descriptive flexfield mappings defined earlier are displayed as part of the input payload fields.

Mapped Flexfields are Displayed as Part of the Input Payload



Displaying Mapped Descriptive Flexfields As Part of the Payload

In the SOAP Body section, expand the Payload > OeHeadersIfaceAll > OeHeadersIfaceAll > FND_FLEX_TEST.



One set of descriptive flexfield configured earlier at the design time are passed as part of the payload input and displayed at the run time.

FND_FLEX_TEST (context value 'IN')

- Address_Line1
- Address_Line2
- Address_Line3
- Town/City
- State
- Pin_Code

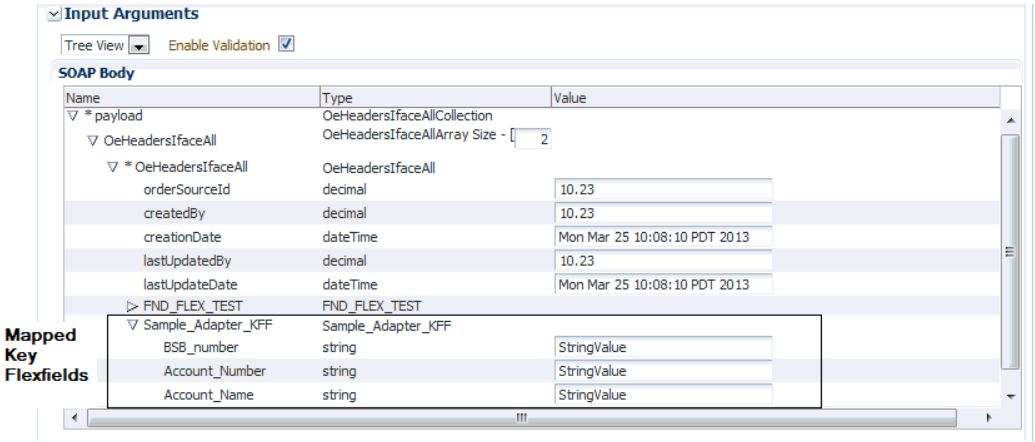
Displaying Mapped Key Flexfields As Part of the Payload

In the SOAP Body section, expand the Payload > OeHeadersIfaceAll > OeHeadersIfaceAll > Sample_Adapter_KFF.

Mapped key flexfield (Sample_Adapter_KFF) with the following mapped segments are displayed as part of the input payload fields:

- BSB_Number
- Account_Number
- Account_Name

Mapped Key Flexfields are Shown as Part of the Input Payload



Enter appropriate data as input payload. Click **Invoke** to invoke the service

In the Instance ID column, click a specific instance ID to show the message flow through the various service components and binding components. The Flow Trace page is displayed.

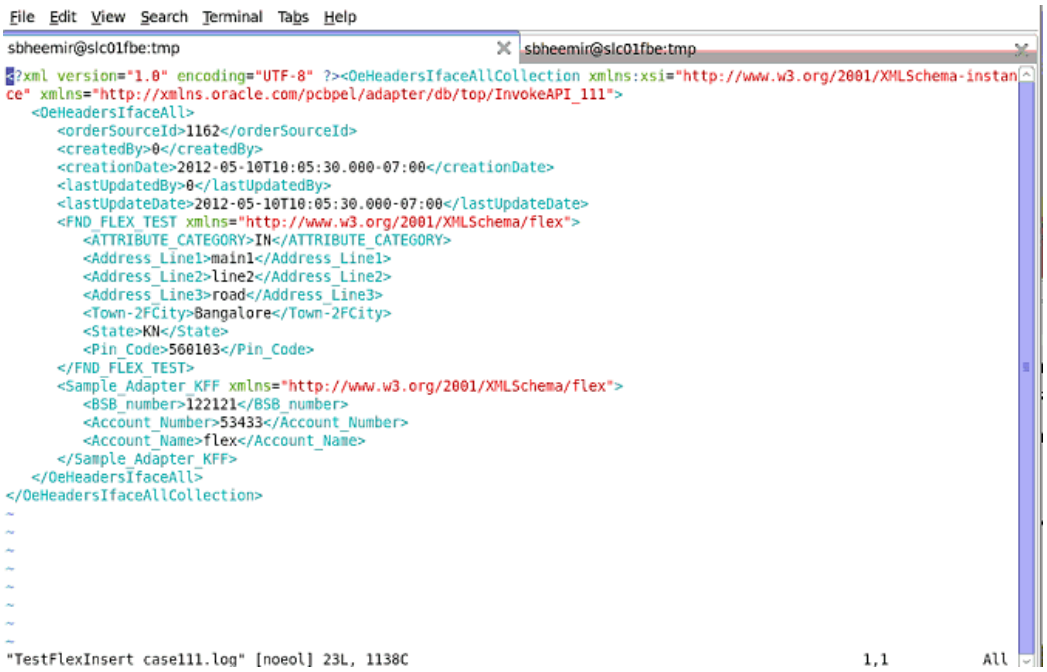
Click the Insert BPEL Component link to open the instance details.

In the Audit Trail tab, click the <payload> link under the first **Invoke** activity. The

payload details is displayed.

Go to the directory (such as \user\tmp) where you specified for the write operation when defining the File Adapter. Open the output XML file. The data should be the same as the input details shown in the payload from the first **Invoke** activity.

The output content is part of a response message for the requestor.

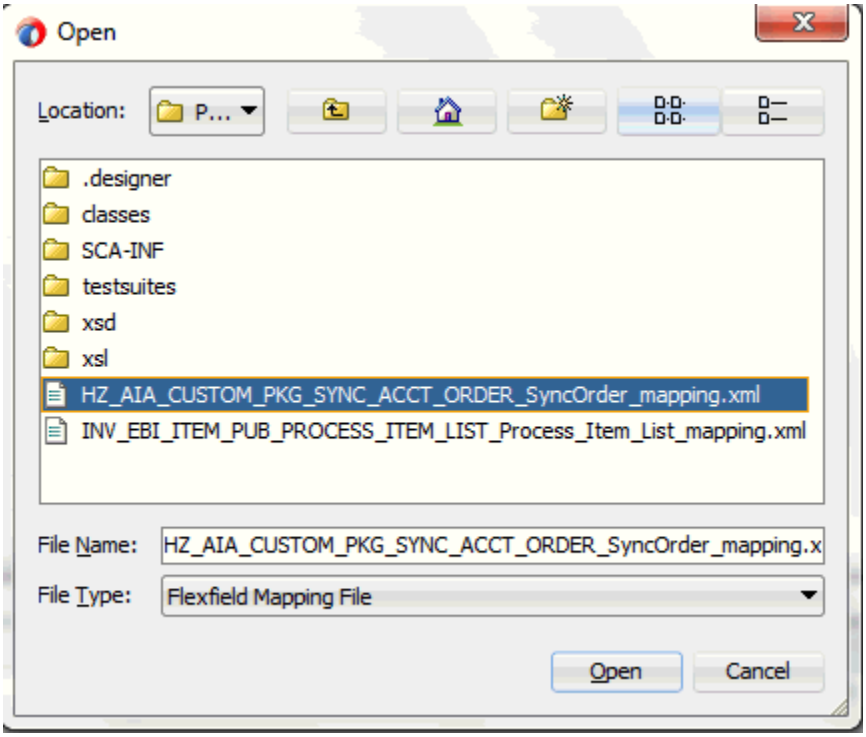


Importing an Existing Flexfield Mapping

Instead of configuring a new mapping, you can simply import an existing mapping for your selected interface. By using a previously configured mapping, you can quickly establish the flexfield mapping for your interface or use it as a base with proper modification if needed.

To import an existing mapping, click **Import from File** in the Configure Flexfields region. The Open window appears where you can select a desired flexfield mapping.

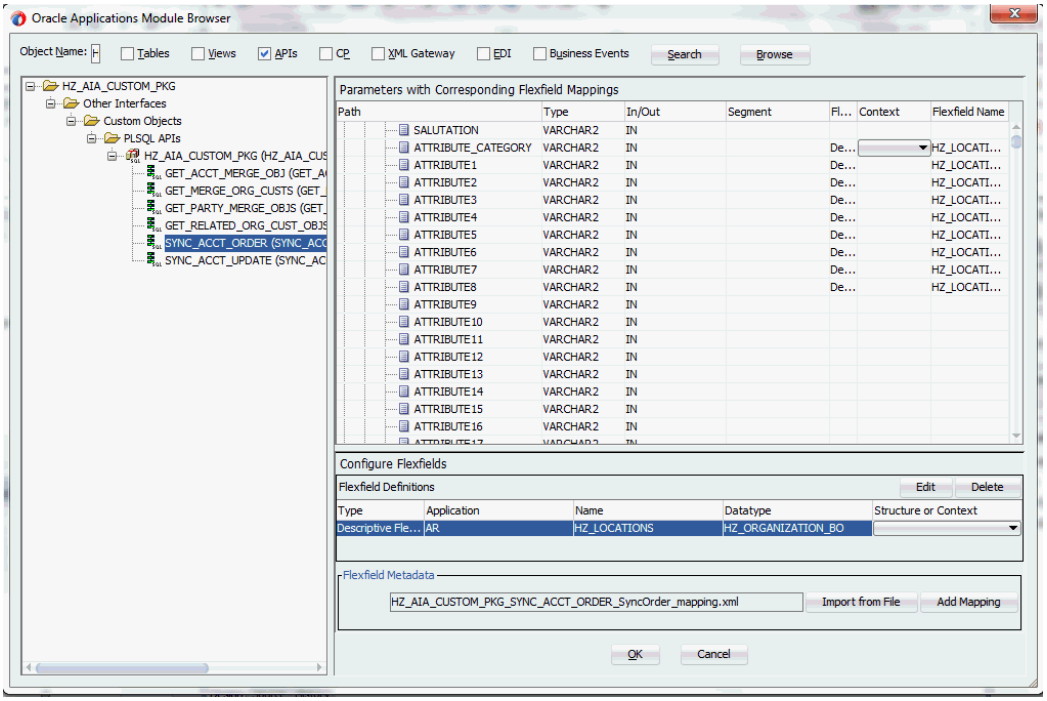
Selecting a desired flexfield mapping



After selecting the desired mapping to be imported, click **Open**. The selected mapping file is automatically displayed in the Import Flexfield Data into the project field.

For example, select 'HZ_AIA_CUSTOM_PKG_SYNC_ACCT_ORDER_SyncOrder_mapping.xml' file and the imported mapping for the selected API is automatically displayed in the Parameters with Corresponding Flexfield Mappings region along with the selected mapping file name.

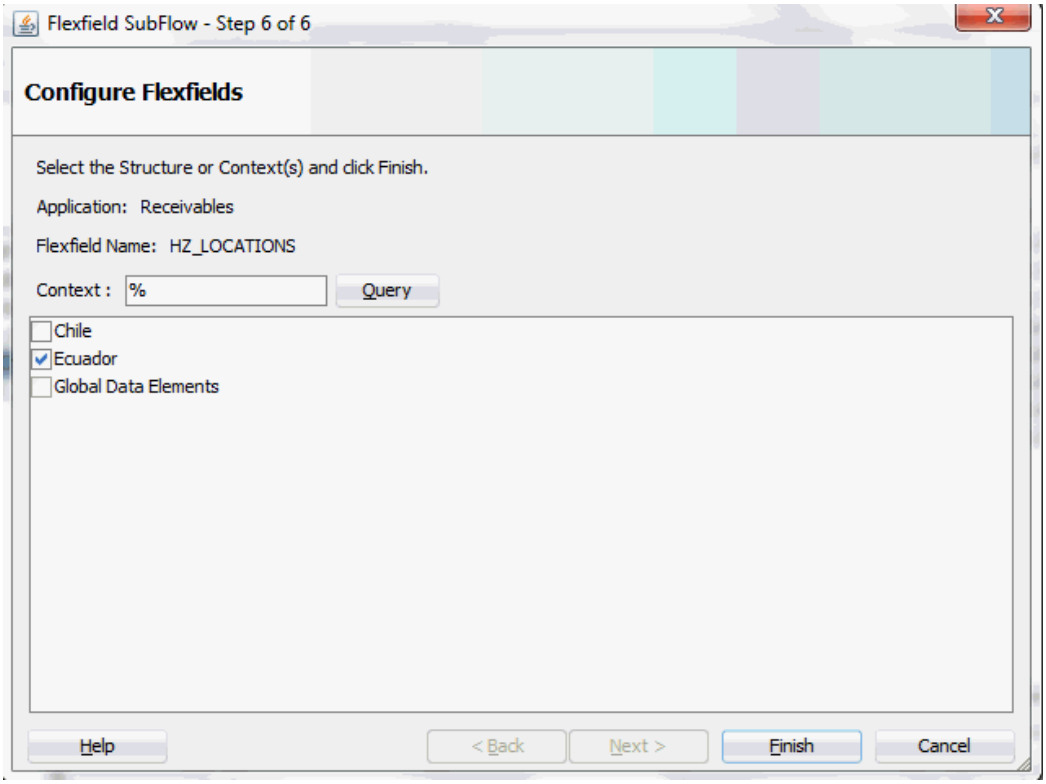
Displaying Imported Flexfield Mapping File



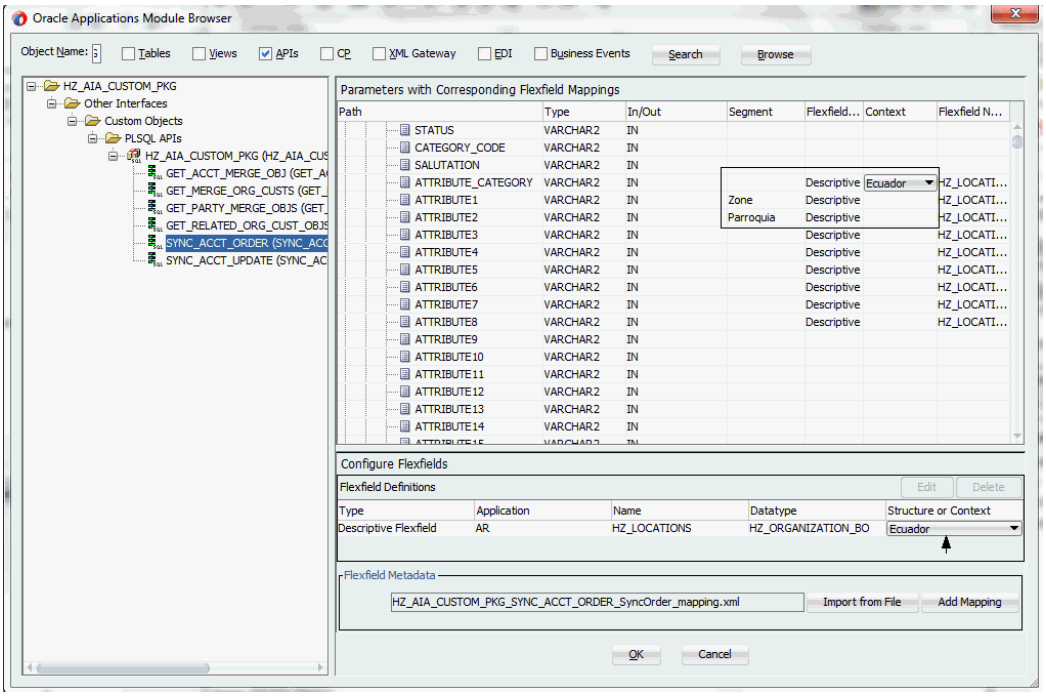
When a desired mapping is imported for the selected interface, you will need to select the structure for a key flexfield and context values for a descriptive flexfields.

First, select the key or descriptive flexfield in the Flexfield Definitions section that requires modification, and then click **Edit**. The Flexfield Subflow - Welcome page is displayed. Click **Next** to select the desired flexfield context values if it's for a descriptive flexfield.

Selecting the Context Values for Imported Mapping



Click **Finish** to complete the descriptive flexfield mapping information. The updated descriptive flexfield context value (Ecuador) is now added both in the Parameters with Corresponding Flexfield Mappings region and Flexfield Definitions section.



Use the same approach to update the key flexfield that requires structure information before clicking **OK**.

For more information on how to modify the selected mapping details, see Modifying Flexfield Mapping Definitions, page 4-119.

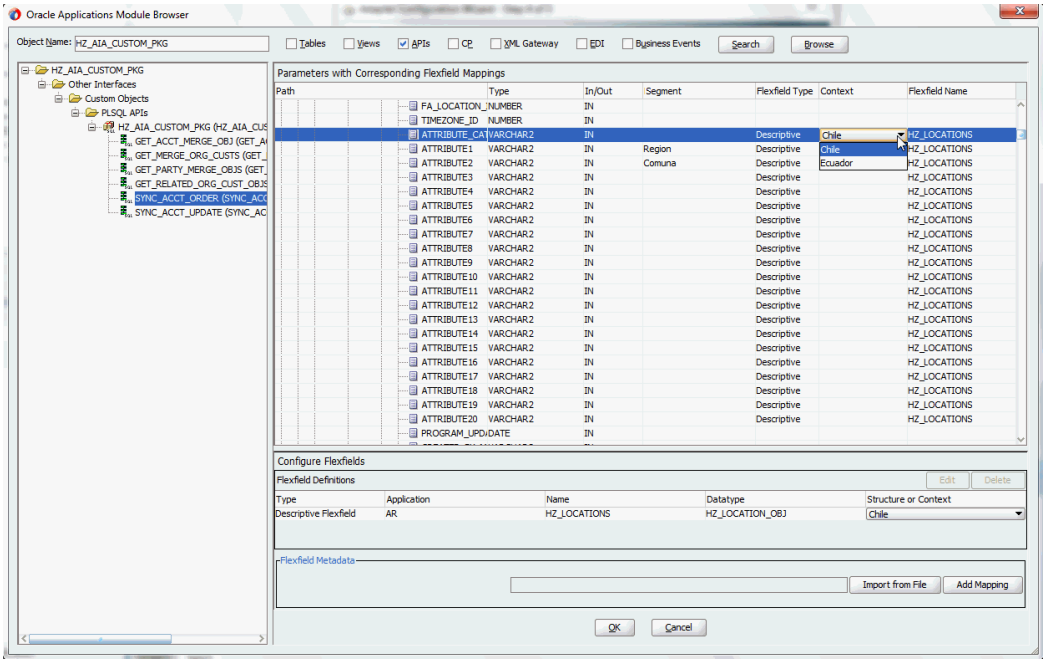
Modifying Flexfield Mapping Definitions

Oracle E-Business Suite Adapter allows you to modify flexfield mapping definitions for both key and descriptive flexfields as often as needed. However, modification is only limited to context values for a descriptive flexfield or flexfield structure for a key flexfield. If further update is needed in addition to modifying structure or context values, you should delete the flexfield mapping first and configure a new one.

Modifying Flexfield Mapping Data

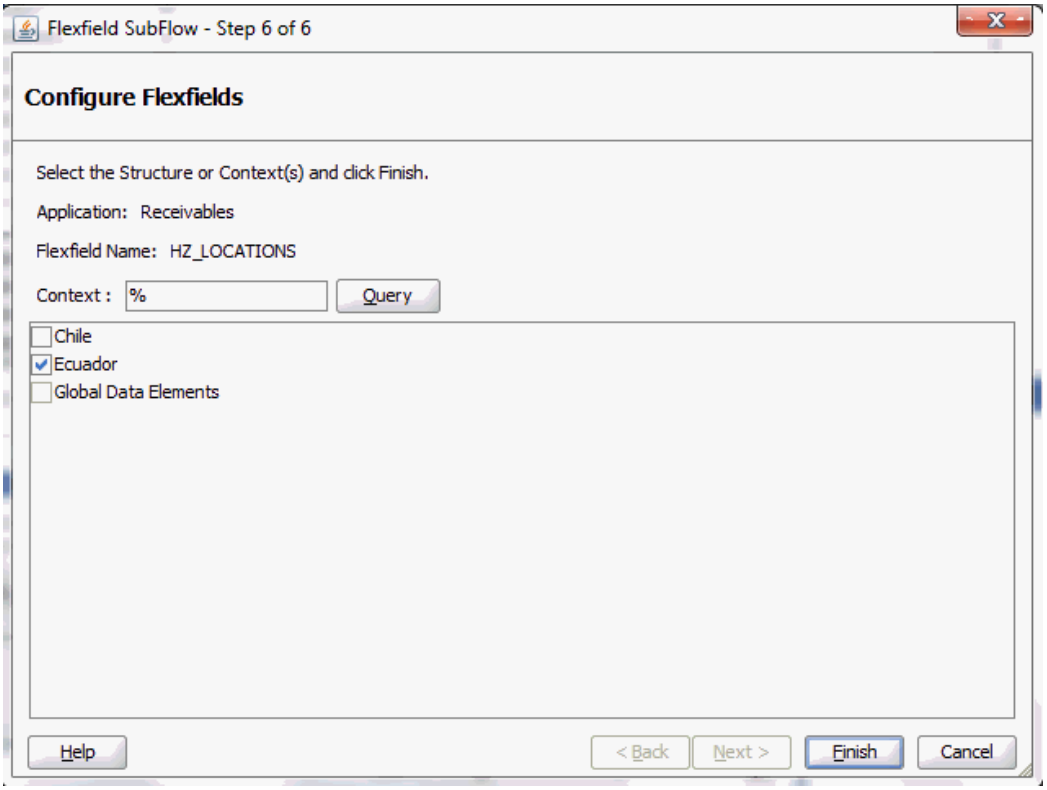
To modify flexfield mapping data, select either the Key Flexfield or Descriptive Flexfield in the Flexfield Definitions section that you want to modify and then click **Edit** . Appropriate flexfield pages corresponding to your selected flexfield type appear where you can modify the flexfield mapping details.

Modifying Flexfield Mapping Data



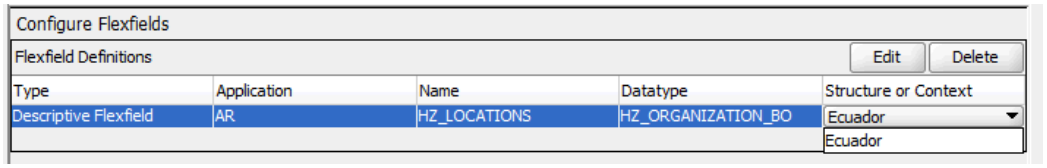
For example, modify the structure of the descriptive flexfield (HZ_LOCATIONS). The Configure Flexfields window appears after you click **Next** in the Flexfield Subflow Welcome wizard.

Flexfield Subflow - Configure Flexfields



Click **Query** and select the 'Ecuador' check box only as the context value. Click **Finish** to complete the modification. The updated flexfield information should be reflected in the Flexfield Definitions section.

Validating Modified Flexfield Mapping Data



You can also validate the changes in the flexfield configuration xml file (`<projectname>_flex-config.xml`). See: Reviewing Flexfield Mapping Configurations, page 4-122.

For more information on how to configure or add each Flexfield Subflow window, see Adding or Configuring a New Flexfield Mapping, page 4-37.

Deleting Flexfield Mapping Data

To delete flexfield mapping data, select either the Key Flexfield or Descriptive Flexfield that you want to delete and then click **Delete**. This removes the selected flexfield

mapping data and the associated structure or values from the database.

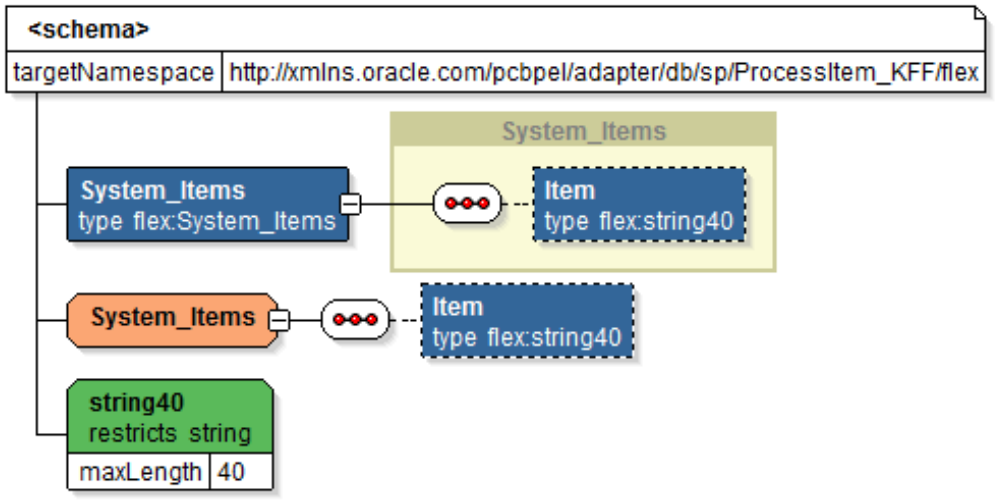
Alternatively, use [Control] and [Shift] keys to select multiple flexfield mappings before the deletion.

Reviewing Flexfield Mapping Configurations

After creating a partner link with flexfield mapping information, from Oracle JDeveloper, expand the **SOA > Schemas** folder and select the enhanced XSD file, for example `ProcessItem_KFF_flex.xsd`, to validate the flexfield mapping information you configured or updated earlier.

The enhanced schema information is displayed with flexfield segment names based on the mapping chosen at the design time.

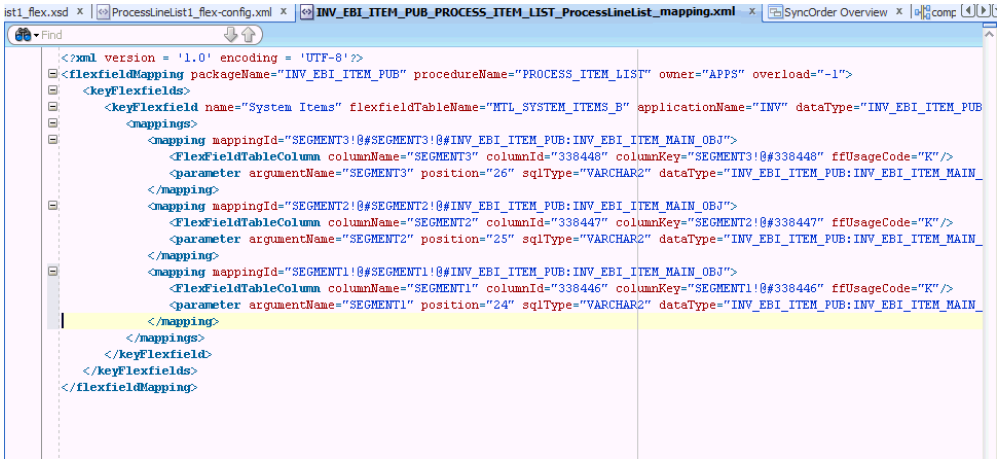
Viewing an Enhanced XSD File



Additionally, two xml files are created along with the XSD file. Expand the **SOA > Adapters** folder to view the file details:

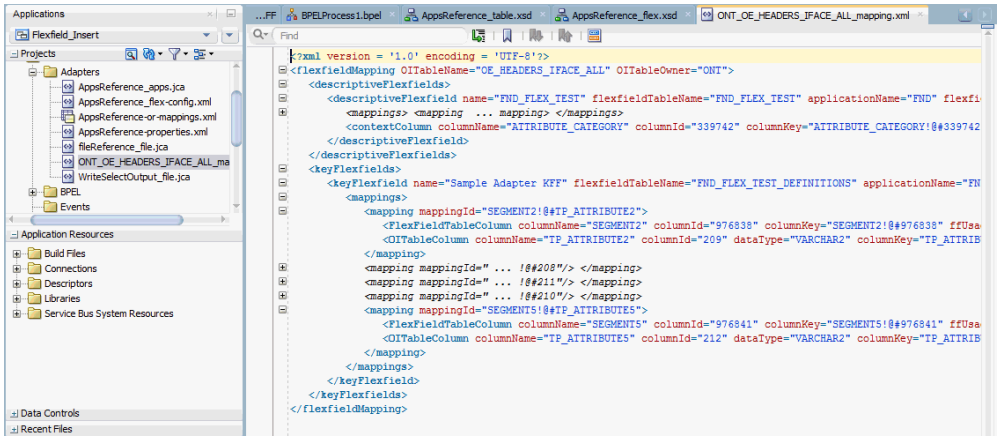
- **Mapping XML:** The mapping xml file (for example `INV_EBI_ITEM_PUB_PROCESS_ITEM_LIST_ProcessItem_KFF_mapping.xml`) would hold the mapping information captured in the mapping wizard panel. This information can be reused for the same API and can be imported from another Partner Link.

Viewing Key Flexfield Mapping Details



Another mapping example (`ONT_OE_HEADERS_IFACE_All_mapping.xml`) is for the open interface table we defined earlier. This example contains the mappings for both key and descriptive flexfields:

Viewing Key and Descriptive Flexfield Mapping Details

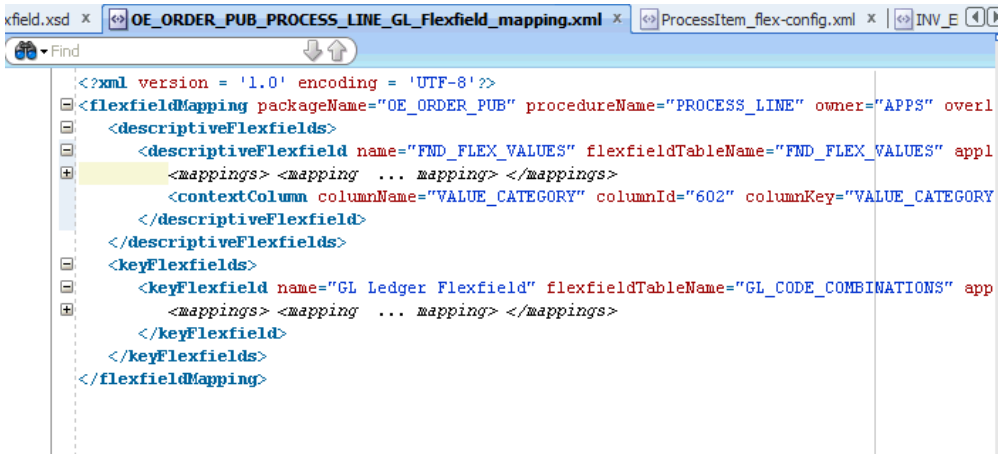


For information on how to import the flexfield mapping for a selected API or open interface table, see Importing an Existing Flexfield Mapping, page 4-115.

- **Config XML:** The config xml (for example `OE_ORDER_PUB_PROCESS_LINE_GL_Flexfield_flex-config.xml`) has the flexfield information including both key and descriptive flexfield mappings if configured. This is not reusable.

  Please note that config xml is internal to the partner link, it should neither be edited nor reused.

Viewing Flexfield Configuration Details



Logging

Oracle E-Business Suite Adapter implements the Oracle SOA Suite's logging framework to write the diagnostic log files in text format. Therefore, whenever the Oracle E-Business Suite services are invoked using Oracle E-Business Suite Adapter, the log messages are recorded which can be accessed by system administrator. This enriches the problem identification mechanism to track any issues during service invocations at run time for Oracle E-Business Suite Adapter.

Oracle E-Business Suite Adapter and technology adapters implement the `LogManager` interface of JCA Binding Component, which redirects the log files for both inbound and outbound interactions to the `soa-diagnostic.log` file in the Oracle Diagnostic Logging (ODL) format. These log files record all types of events including startup and shutdown information, errors and warning messages, access information on HTTP requests, and additional information. It provides a quick, inside-out view about the invocation process which greatly helps administrators identify and resolve potential issues efficiently. With proper log-level configuration in Oracle Enterprise Manager Fusion Middleware Control Console, you can view ODL level log files written to a single file at run time for Oracle E-Business Suite Adapter as well as technology adapters.

To better understand how the logging mechanism work, the following topics are discussed in this chapter:

- Enabling Logging for Adapters, page 4-125
- Searching and Viewing Adapter Logs, page 4-129

For more information about SOA Suite, see *Oracle Fusion Middleware Administrator's Guide for Oracle SOA Suite.*

Enabling Logging for Adapters

All logs of Oracle E-Business Suite Adapter and technology adapters are redirected to the `soa-diagnostic.log` file in the Oracle Diagnostic Logging (ODL) format. To be able to view logs for Oracle E-Business Suite Adapter, you need to set or enable an appropriate message type and its associated log level for the logger `oracle.soa.adapter` using the Oracle Enterprise Manager Fusion Middleware Control Console. This enables the log settings at the Oracle SOA Suite level.

> **Note:** The logger configuration is under `$FMWHOME/user_projects/domains/soainfra/config/fmwconfig/servers/soa_server1/logging.xml` and the corresponding logger name for all Oracle JCA Adapters is called `oracle.soa.adapter`.

The following table lists the diagnostic message types and log levels:

> **Note:** For each message type, possible values for message level are from 1 (highest severity) through 32 (lowest severity). Some components support only some of the levels for each message type. Generally, you need to specify only the type; you do not need to specify the level.
>
> The default message type for each logger is set to NOTIFICATION, level 1.

Message Type	Level	Description
INCIDENT_ERROR	1	A serious problem that may be caused by a bug in the product and that should be reported to Oracle Support. Examples are errors from which you cannot recover or serious problems.

Message Type	Level	Description
ERROR	1	A serious problem that requires immediate attention from the administrator and is not caused by a bug in the product. An example is if Oracle Fusion Middleware cannot process a log file, but you can correct the problem by fixing the permissions on the document.
WARNING	1	A potential problem that should be reviewed by the administrator.
NOTIFICATION	1	A major lifecycle event such as the activation or deactivation of a primary subcomponent or feature.
NOTIFICATION	16	A finer level of granularity for reporting normal events.
TRACE	1	Trace or debug information for events that are meaningful to administrators, such as public API entry or exit points.
TRACE	16	Detailed trace or debug information that can help Oracle Support diagnose problems with a particular subsystem.
TRACE	32	Very detailed trace or debug information that can help Oracle Support diagnose problems with a particular subsystem.

To set the diagnostic message type and log level for Oracle E-Business Suite Adapter in Oracle SOA Suite:

Use the following procedures to set the message type and its associated log level:

1. Navigate to `http://<servername>:<portnumber>/em`.

   The Oracle Enterprise Manager Fusion Middleware Control Console home page is displayed.

2. Enter username and password to log on to the console.

3. Right-click `soa-infra` from the SOA Folder in the Navigator tree.

4. Select **Logs > Log Configuration** from the pop-up menu.

   > **Note:** You can also access the Log Configuration page by clicking the SOA Infrastructure Menu and selecting **Logs > Log Configuration** from the drop-down menu.

   This opens the Log Configuration page where you can view a list of loggers (persistent or active run time) and configure the Oracle Diagnostic Logging (ODL) level for setting the amount and type of information to write to a log file and the log level state.

5. Select the **Log Levels** tab.

6. Select one of the following values from the View drop-down list:

   - **Runtime Loggers (default):** Runtime loggers are automatically created during run time and become active when services are getting executed. For example, `oracle.soa.b2b` or `oracle.soa.bpel` are runtime loggers. Log levels for runtime loggers are not persistent across component restarts.

   - **Loggers with Persistent Log Level State:** Persistent loggers are loggers that are saved in a configuration file and become active when the component is started. The log levels for these loggers are persistent across component restarts.

     > **Note:** By default, the log level is set for Runtime Loggers. Runtime loggers do not persistent across when a component restarts. To ensure that log levels persist across component when it restarts, select **Loggers With Persistent Log Level State** from the View list.

7. Expand the `oracle.soa` node to locate `oracle.soa.adapter` runtime logger in the Logger Name list. Select the logging level from the Oracle Diagnostic Log Level drop-down list. For example, select 'TRACE:32 (FINEST)'.

8. Click **Apply**.

Creating and Editing Log File in the Log Files Tab

You can edit a specific log file by clicking the log handler link displayed in the Log File column. This opens the Log Files tab where you can configure the basic and advanced log configuration settings. These settings include handler's name, the log file in which the log messages are logged, the format of the log messages, the rotation policies used, and other parameters based on the log file configuration class.

For example, select the log handler from the table and click **Edit Configuration**. The Edit Log File dialog box is displayed.

- To change log file location, enter a new path in the Log Path field.
- To configure message levels, select the logging level from the Log Level drop-down list. For example, select 'TRACE:32 (FINEST)'.
- To configure log file rotation, in the Rotation Policy section, select either Size Based or Time Based log file with appropriate information.

Fore more information on how to configure log file, see the Managing Log Files and Diagnostic Data Chapter, *Oracle Fusion Middleware Administrator's Guide*.

For more information about SOA Suite, see *Oracle Fusion Middleware Administrator's Guide for Oracle SOA Suite.*

Automatic FND Logging for PL/SQL APIs and Concurrent Programs Using oracle.soa.adapter Runtime Logger

Based on the log level set for `oracle.soa.adapter` runtime logger in Oracle SOA Suite, logging can be automatically controlled or enabled for PL/SQL APIs and Concurrent Programs while these interfaces are invoked during run-time execution. There is no need to enable the FND logging framework on the Oracle E-Business Suite side separately.

> **Note:** This automatic FND logging would be available in Oracle E-Business Suite Release 12.1.3 and above only.

The following table lists the log level mapping between `oracle.soa.adapter` runtime logger and the corresponding Oracle E-Business Suite:

Runtime Logger oracle.soa.adapter Log Level	Oracle E-Business Suite Log Level
INCIDENT_ERROR	LEVEL_UNEXPECTED
ERROR	LEVEL_ERROR
WARNING	LEVEL_EXCEPTION

Runtime Logger oracle.soa.adapter Log Level	Oracle E-Business Suite Log Level
NOTIFICATION 1, 16, 32	LEVEL_EVENT
TRACE 1	LEVEL_PROCEDURE
TRACE 16, 32	LEVEL_STATEMENT

Searching and Viewing Adapter Logs

Oracle E-Business Suite Adapter and technology adapters implement the `LogManager` interface of JCA Binding Component, which redirects the log files written to a single file at run time in the Oracle Diagnostic Logging (ODL) format.

For both outbound and inbound interactions, the log files are redirected to the single file `soa-diagnostic.log`.

The log files for Oracle SOA Suite that is deployed to the `server-soa` managed server are located in `MW_HOME/user_projects/domains/<domain_name>/servers/server-soa/logs/soa-diagnostic.log`.

To search and view Oracle E-Business Suite Adapter log files, you can use Oracle Enterprise Manager Fusion Middleware Control Console, the WLST `displayLogs` command-line tool, or you can download log files to your local client and view them using another tool (for example a text editor, or another file viewing utility). For information on how to use WLST command-line tool to search and view log files, see the Managing Log Files and Diagnostic Data Chapter, *Oracle Fusion Middleware Administrator's Guide*.

Steps to Search and View Adapter Logs Using Oracle Enterprise Manager Fusion Middleware Control Console

Use the following steps to search Adapter Logs through the Oracle Enterprise Manager Fusion Middleware Control Console:

1. Navigate to `http://<servername>:<portnumber>/em`.

   The Oracle Enterprise Manager Fusion Middleware Control Console home page is displayed.

2. Enter username and password to log on to the console.

3. There are two ways to access the Log Messages page from the Navigator tree in the left pane:

- From the SOA folder, right-click `soa-infra`.
- From the WebLogic Domain folder, right-click `soainfra`.

Select **Logs** and then **View Log Messages** from the pop-up menu. The Log Messages page is displayed with the Search section and a table that shows a summary of the messages with default search criteria.

4. Enter the following search criteria for searching the log messages for Oracle E-Business Suite Adapter:
   - **Date Range:** Select your value from the drop-down list and enter an appropriate number. For example, 'Most Recent' 6 hours.

     If 'Time Interval' is selected from the drop-down list, select the calendar icon for Start Date and then select the date and time. Similarly, select the calendar icon for End Date and then select the date and time.
   - **Message Types:** Select one or more of the message types. For example, select the **Trace** check box if it is the message type configured earlier for Oracle E-Business Suite Adapter.
   - **Message:** Select 'Contains' from the list of values and then enter 'Oracle E-Business Suite Adapter' in the text box.
   - Specify more search criteria if needed in the Search section.

     You can optionally add more search criteria by clicking **Add Fields**. This action allows you to add more criteria, such as Host, which lets you narrow the search to particular hosts. Then click **Add**.
5. Execute the search by clicking **Search**. All messages that match your search criteria will be retrieved and displayed in the table. These messages can be displayed as messages, or can be grouped by message type or message ID depending on the selected value in the Show field.
6. Click one of the log messages from the table. The selected message details, such as message level, component, ECID (Execution Context ID), Relationship ID, actual message, are displayed below the table of messages.

   Clicking the ECID link retrieves related messages with the same ECID in the Related Messages by ECID page. For more information on related messages, see Correlating Messages Across Log Files and Components, page 4-131.
7. Select an appropriate output option from the **Export Messages to File** drop-down list if you want to export log messages as a Oracle Diagnostic Log Text file (.txt) XML file (.xml), or Comma-Separated List (.csv) file.
8. Click **Target Log Files** to open the Log Files page where you can view a list of log

files related to the managed server (`server-soa`).

1. Select a file and click **View Log File**. The View Log File page is displayed for the selected log file where you can view a list of messages contained in this log.

2. To view the details of a message, select the message. The message details, such as message level, component, ECID, Relationship ID, actual message, are displayed below the table of messages.

3. To view messages that are related by time or ECID, click **View Related Messages** and select 'by Time' or by 'ECID (Execution Context ID)'.

   Alternatively, clicking the ECID link directly from the message details also retrieves related messages with the same ECID in the Related Messages by ECID page.

Correlating Messages Across Log Files and Components

Oracle Fusion Middleware components provide **message correlation** information for diagnostic messages. Message correlation information helps those viewing diagnostic messages to determine relationships between messages across components. Each diagnostic message contains an Execution Context ID (ECID) and a Relationship ID.

An ECID is a globally unique identifier associated with the execution of a particular request. An ECID is generated when the request is first processed. A Relationship ID distinguishes the work done in one thread on one process from the work done by any other threads on this and other processes on behalf of the same request.

While viewing log messages in the Oracle Enterprise Manager Fusion Middleware Control Console, you can view correlated messages by selecting a log message first, and then selecting one of the following values from the **View Related Messages** drop-down list:

> **Note:** The **View Related Messages** selection is available only when 'Messages' is chosen in the Show field when displaying all matching messages based on search criteria. If 'Group by Message Type' or 'Group by Message ID' is selected in the Show field, then all matching messages are displayed by groups based on message type or message ID. In this situation, the **View Related Messages** field is not available.

- **by Time:** This displays the Related Messages by Time page where all messages with the same timestamp as the selected message are displayed in this page.

- **by ECID (Execution Context ID):** This displays the Related Messages by ECID page where all messages with the same ECID as the selected message are displayed in this page.

By searching for related messages using the message correlation information, multiple

messages can be examined and the component that first generates a problem can be identified. Message correlation data can help establish a clear path for diagnosing errors generated in the API being invoked.

Enhanced Error and Exception Handling

In order to identify the root cause of an issue or error which might occur during invocation of an Oracle E-Business Suite interface, Oracle E-Business Suite Adapter provides user-friendly, descriptive exceptions and error messages. Especially, these would be available during invocation of any PL/SQL APIs.

Apart from handling errors during adapter preprocessing and the run-time execution by Database Adapter and AQ Adapter, it also provides a way to retrieve the functional errors raised by PL/SQL APIs at run time, without the need for an extra call to retrieve them. These messages would guide you to pinpoint the cause of failures at run time.

Additionally, based on the common logging framework used by other JCA Adapters, all errors and exceptions pertaining to Oracle E-Business Suite Adapter and debug information will be logged. This allows administrators or developers to better understand what happened during the execution and take necessary actions to solve the issue.

Handling Functional Errors

During the invocation of an integrated interface supported by Oracle E-Business Suite Adapter, in case of issues or errors occurred (such as the entered order number does not exist while invoking `OE_ORDER_PUB.CHANGE_ORDER` API), an exception (`ORA-20100: Order number does not exist`) would be thrown and such exception consists of functional errors. This type of exception is used extensively to report functional problems while executing the API.

For most PL/SQL APIs, functional errors can be thrown as `SQLException` caught by Adapter runtime engine along with code and message or as part of API output parameters. However, certain PL/SQL APIs do not throw functional errors during execution, but they keep on accumulating them in the memory stack. The responsibility of fetching the errors is deferred to the caller of the APIs.

For example, HRMS APIs put functional errors in FND_MESSAGE stack using `FND_MSG_PUB.put` package. Upon completion of the API call, the caller must explicitly fetch the errors from this stack using `FND_MSG_PUB.GET()` methods.

> **Note:** Error messages can only be retrieved from the stack `FND_MSG_PUB` in this release.

For such APIs that require explicit handling, a new JCA property called `APIErrorHandler` is introduced to fetch functional errors from the FND_MESSAGE memory stack.

At run time, if the JCA property is set, the error handler will be invoked right after the execution of the PL/SQL API. If this returns one or more errors, then Oracle E-Business Suite Adapter will throw runtime exception. The exception handler, such as `catchAll` fault handler, can be used to process the messages coming out of this function.

Please note that the following functional error handling parameter needs to be added to the WSDL file (`XX_apps.jca`) for fetching the functional errors for such APIs:

```
<property name="APIErrorHandler"
value="FND_MSG_PUB.GET_DETAIL"/>
```

Example of Error Handling Using APIErrorHandler JCA Property

SOA Composite Application with BPEL Process Scenario

This example uses a PL/SQL API called Create Project (`PA_PROJECT_PUB.CREATE_PROJECT`) to explain how to use the `APIErrorHandler` JCA property along with the `catchAll` fault handler to catch any faults which might occur during the service invocation.

At run time, if the service invocation is executed successfully, a project should be created in Oracle Projects by using an existing project. If any error occurs, the `catchAll` fault handler will catch the error messages. These messages will be written in an xml file.

Functional Requirement

To successfully create a project in Oracle Project using the PL/SQL API (`PA_PROJECT_PUB.CREATE_PROJECT`), the following functional requirement must be in place:

- An existing project is associated with a template in Oracle Project.
- A valid Oracle E-Business Suite user who has the responsibility to create a project in Oracle Project, and that user must have the privilege to execute this `PA_PROJECT_PUB.CREATE_PROJECT` API.

This example includes the following design-time activities:

1. Create a main process to invoke the Create Project service through a partner link
   - Create a partner link for the Create Project API
   - Add an Invoke activity to invoke the API
   - Add an Assign activity to assign the input for the service invocation
   - Add another Assign activity to assign the output for the service invocation
2. Add a CatchAll process on the main process you just created to catch faults if occur during the service invocation

- Create a file adapter with "Write File" as the operation type
- Add an Assign activity to pass the faults from the `catchAll` fault handler
- Add an Invoke activity to invoke the file adapter partner link to write the error messages in an xml file

SOA Composite Application with BPEL Process Creation Flow

Based on the process scenario, the following design-time tasks are explained in this section:

1. Create a new SOA Composite application with BPEL process, page 4-134
2. Create a Partner Link, page 4-136
3. Create a Partner Link for File Adapter, page 4-137
4. Add a CatchAll Activity on the Main Scope, page 4-139
5. Create Invoke Activities, page 4-140
6. Create Assign Activities, page 4-143

Deploying and Testing the Composite Application

Once you complete the design-time activities, deploy the composite application and test it to see if the output file generated by the file adapter contains the faults.

See Validating and Testing SOA Composite Application with BPEL Process, page 4-149.

For the payload information on creating a project, see Sample Payload for Creating a Project, page 4-153.

Creating a New SOA Composite Application

Use this step to create a new SOA composite application that will contain various BPEL process activities.

To create a new SOA composite application with BPEL process:

1. Open Oracle JDeveloper.
2. Click **New Application** in the Application Navigator.

   The "Create SOA Application - Name your application" page is displayed.
3. Enter an appropriate name for the application in the Application Name field and select **SOA Application** from the Application Template list.

   Click **Next**. The "Create SOA Application - Name your project" page is displayed.

4. Enter an appropriate name for the project in the Project Name field, for example, `TestProjectCreation`.

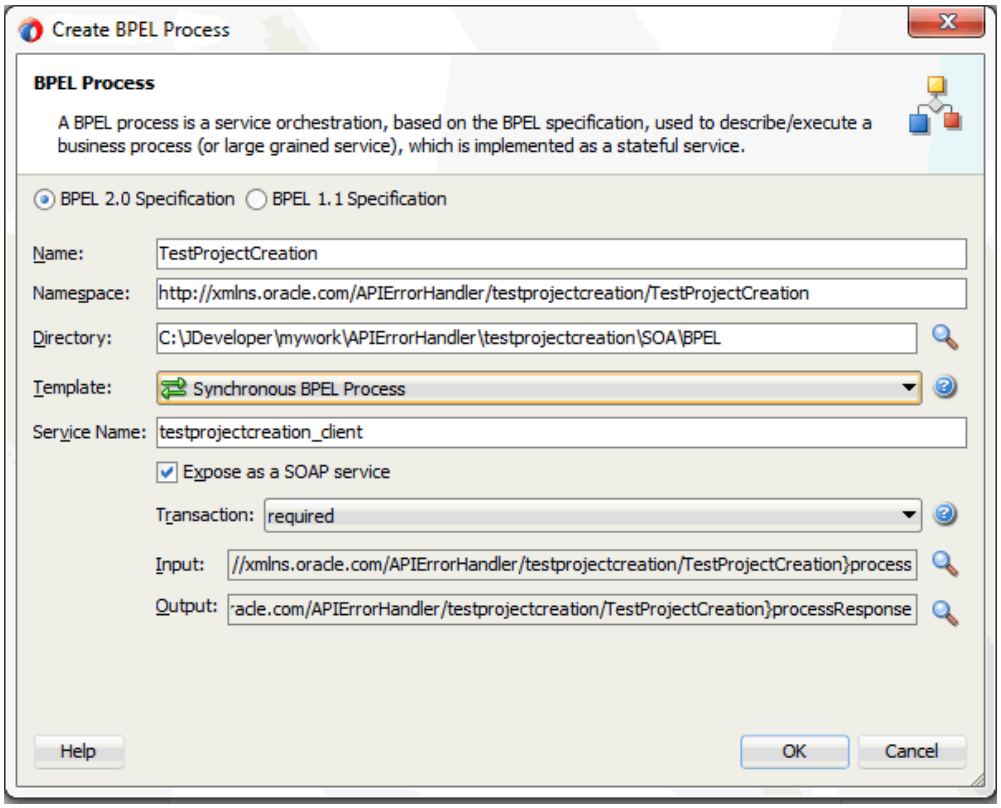
5. In the Project Technologies tab, select 'Web Services' and ensure that **SOA** is selected from the Available technology list to the Selected technology list.

   Click **Next**. The "Create SOA Application - Configure SOA settings" page is displayed.

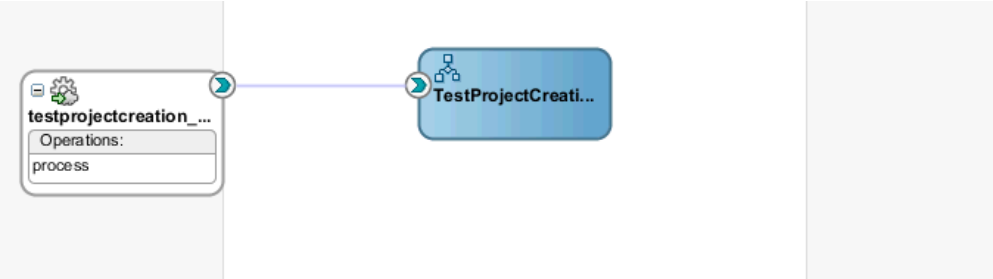
6. Select **Composite With BPEL** from the Composite Template list, and then click **Finish**. You have created a new application, and a SOA project. This automatically creates a SOA composite.

   The Create BPEL Process page appears.

7. Leave the default **BPEL 2.0 Specification** selection unchanged. This creates a BPEL project that supports the BPEL 2.0 specification.



   Enter an appropriate name for the BPEL process in the Name field. Select **Synchronous BPEL Process** in the Template field. Select 'required' from the Transaction drop-down list and click **OK**. A synchronous BPEL process is created with the Receive and Reply activities. The required source files including bpel and wsdl, using the name you specified and the associated composite are also generated.



Creating a Partner Link

Use this step to create a partner link called `CreateProjectApi` for the PL/SQL API that will be invoked later.

To create a partner link:

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

   The Adapter Configuration Wizard Welcome page appears.

2. Enter a service name in the Service Name field. For example, `CreateProjectApi`. Click **Next**. The Service Connection dialog appears.

3. Specify your database connection and proceed to the Oracle Applications Module Browser.

   Locate the PL/SQL API in the Oracle Applications Module Browser by navigating to *Projects Suite (PJ_PF) > Projects (PA) > Project (PA_PROJECT) > PLSQL > Project Definition (PA_PROJECT_PUB)* to select `Create Project (PA_PROJECT_PUB.CREATE_PROJECT)`.

   For information about Oracle Applications Module Browser, see Understanding the Oracle Applications Module Browser, page 4-14.

4. In the Application Interface page, click **Next**.

   In the Finish page, click **Finish** to complete the process of configuring Oracle E-Business Suite Adapter.

   The wizard generates the `CreateProjectApi.wsdl` file corresponding to the `CreateProjectApi_sp.xsd` schema. This WSDL file is now available for the partner link.

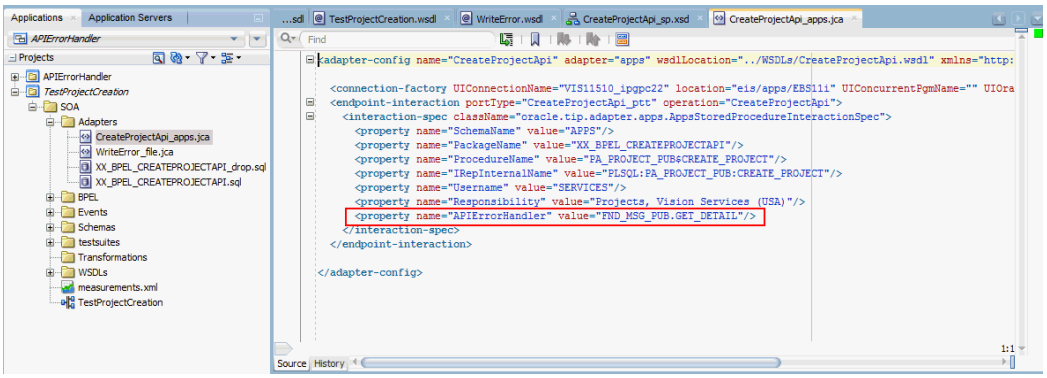
   Select the Partner Link Type and Partner Role fields from the drop-down lists. Click **Apply** and then **OK** to complete the partner link configuration.

To fetch the functional errors for the API you just created as a partner link:

Navigate to SOA > Adapters folder and double click the

`CreateProjectApi_apps.jca`. Add the following functional error handling parameter to the `CreateProjectApi_apps.jca` file and save the changes:

```
<property name="APIErrorHandler"
value="FND_MSG_PUB.GET_DETAIL"/>
```



Creating a Partner Link for File Adapter

Use this step to configure a BPEL process by writing the output to a text file.

1. In Oracle JDeveloper, drag and drop the **File Adapter** service from the **BPEL Services** list into the right Partner Link swim lane 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, for example `WriteError`.

4. Click **Next**. The Adapter Interface dialog box appears.

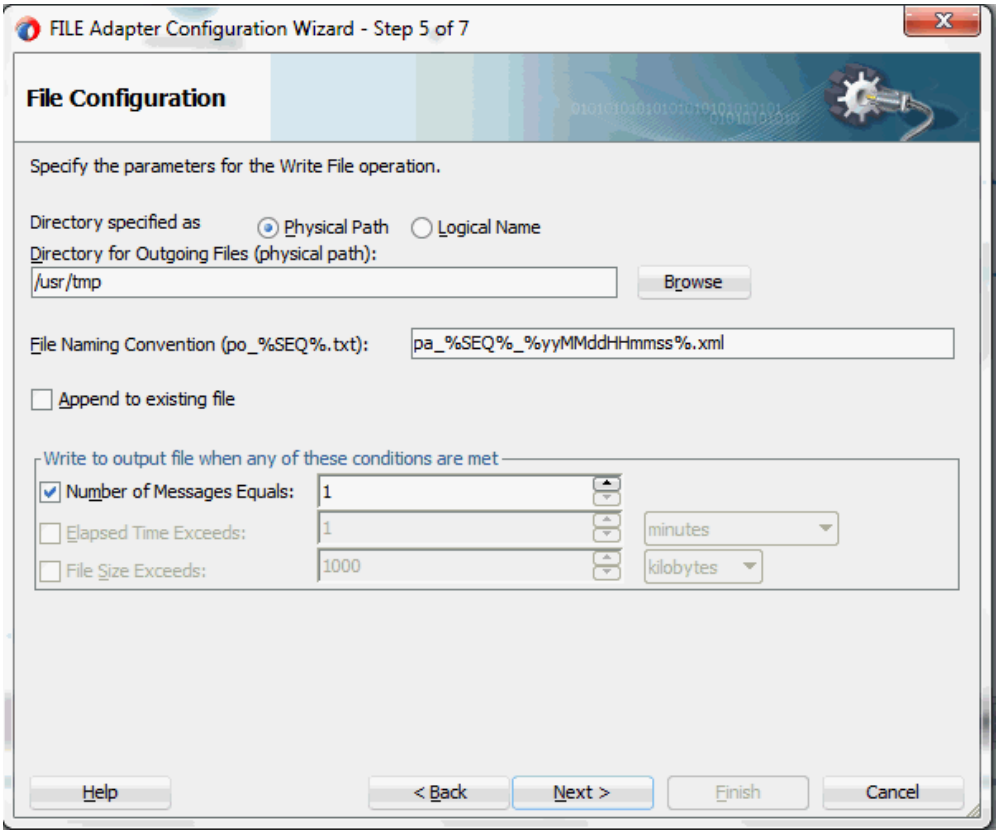
5. Select the **Define from operation and schema (specified later)** radio button and click **Next**. The Operation dialog box appears.

6. Specify the operation type, for example **Write File**.

   This automatically populates the **Operation Name** field.

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

7. For the Directory specified as field, select **Physical Path**. Enter directory path, such as `/usr/tmp/`, in the Directory for Outgoing Files field. Specify a naming convention for the output file, such as pa_%SEQ%_%yyMMddHHmmss%.xml.



Click **Next**. The Messages dialog box appears.

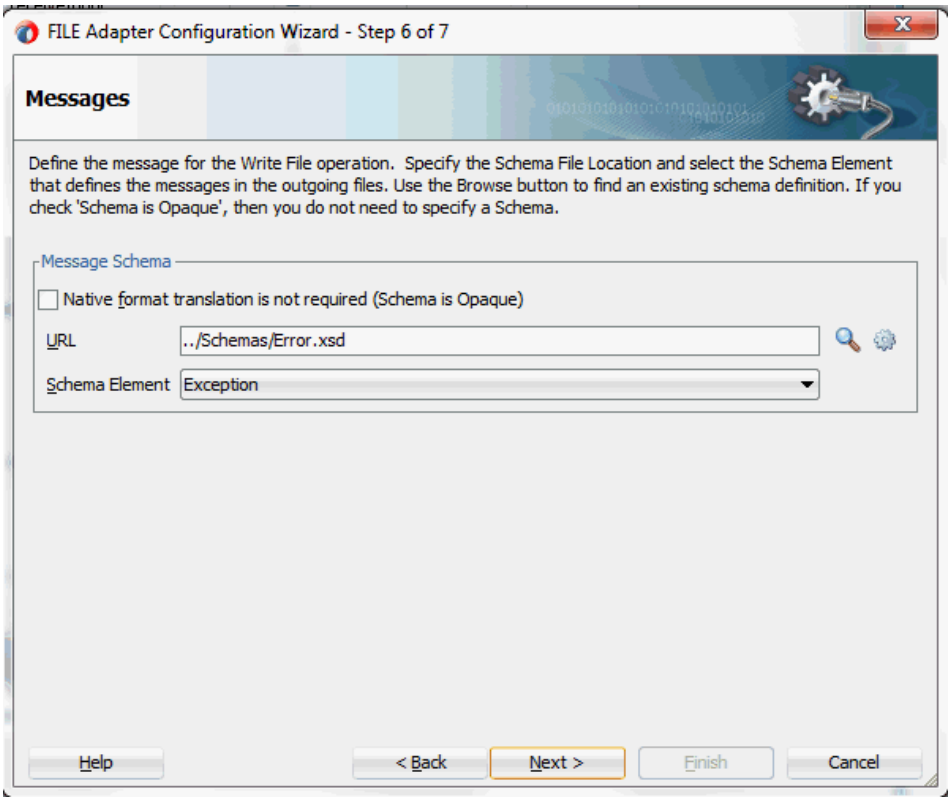
8. Select **Browse for schema file** in front of the URL field. The Type Chooser window is displayed.

    1. Click the **Import Schema Files** button on the top right corner of the Type Chooser window.

    2. Click the **Browse Resources** button in the Import Schema File window to add the Error.xsd file to the schema. Select the **Copy to Project** check box. Click **OK**.

       Select the **Maintain original directory structure for imported files** Copy Options check box and click **OK**.

    3. The selected Error.xsd is displayed as URL and the Exception element is selected as Schema 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 `WriteError.wsdl`.

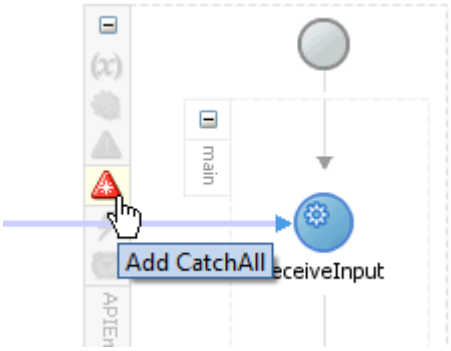
   Click **Apply** and then **OK** to complete the configuration and create the partner link with the required WSDL settings for the File Adapter Service.

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

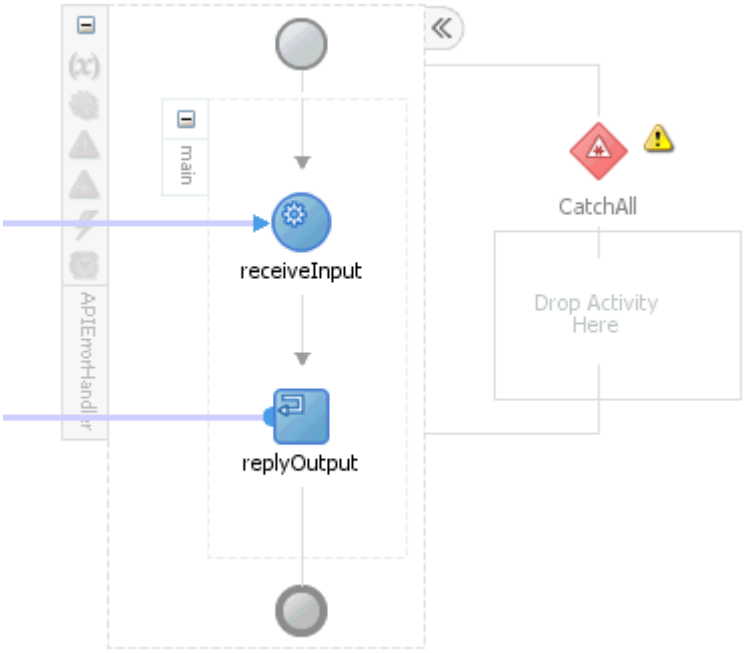
Adding a CatchAll Activity on the Main Scope

This step adds a CatchAll activity to catch any faults if occur during the invocation process.

1. In the expanded Scope activity of the Oracle JDeveloper, click the **Add CatchAll** icon.



2. You can find that a CatchAll activity is created in the right side of the main scope activity.



Creating Invoke Activities

This step is to configure two Invoke activities in order to:

- Invoke the `CreateProjectApi` partner link service.
- Invoke the `WriteError` file adapter partner link that in turn writes the error messages in a output file.

To add an Invoke activity for CreateProjectApi Partner Link:

1. In Oracle JDeveloper, expand the **BPEL Constructs** from the Component Palette.

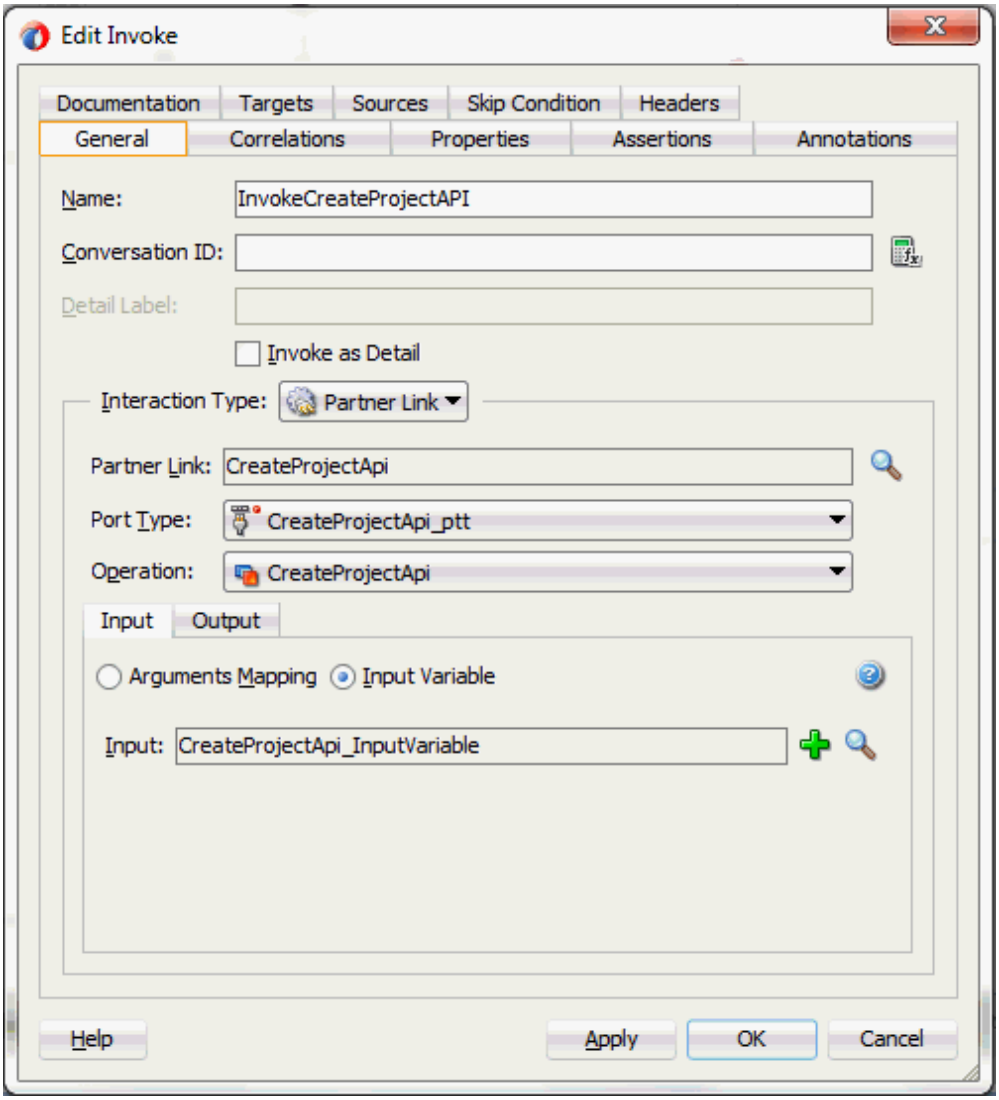
Drag and drop the **Invoke** activity from the Component Palette into the center swim lane of the process diagram, between the **receiveInput** and **replyOutput** activities.

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

3. Enter a name for the Invoke activity, such as InvokeCreateProjectAPI.

4. In the Input tab, click the **Create** icon next to the **Input Variable** field to create a new variable. The Create Variable dialog box appears.

   Enter a name for the variable. You can also accept the default name. Ensure the **Global Variable** radio button is selected and click **OK**.



5. Select the Output tab. Use the same method mentioned in step 4 to create an output variable.

6. Click **Apply** in the Edit Invoke dialog box.

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

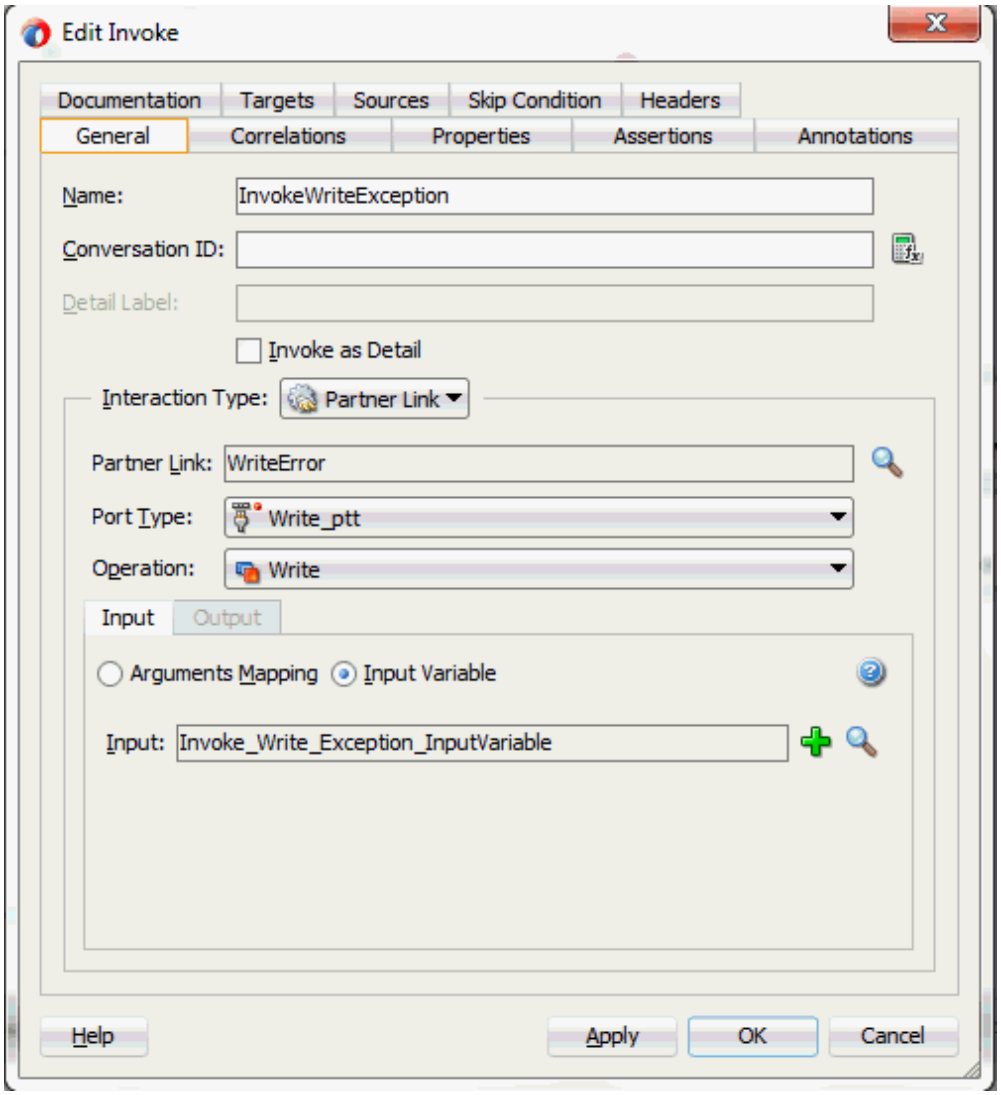
   The Invoke activity appears in the process diagram.

To add the second Invoke activity for WriteError File Adapter Partner Link:

1. In Oracle JDeveloper, expand the **BPEL Constructs** from the Component Palette. Drag and drop the second **Invoke** activity from the Component Palette into the center swim lane of the CatchAll process diagram.

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

3. Follow the instructions described in step 4 of the first Invoke activity to create the input variable for this Invoke activity.



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

The second Invoke activity appears in the process diagram.

Creating Assign Activities

This step is to configure three Assign activities:

1. To pass the input information to the `CreateProjectApi` service through the

Invoke activity.

2. To pass the output information from the `CreateProjectApi` service through the Invoke activity.

3. To pass the faults caught through the CatchAll activity as the input to the `WriteError` file adapter.

To add the first Assign activity in the main scope:

1. In Oracle JDeveloper, expand the **BPEL Constructs** from the Component Palette. Drag and drop the **Assign** activity into the center swim lane of the process diagram, between the **receiveInput** and **Invoke** activities.

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

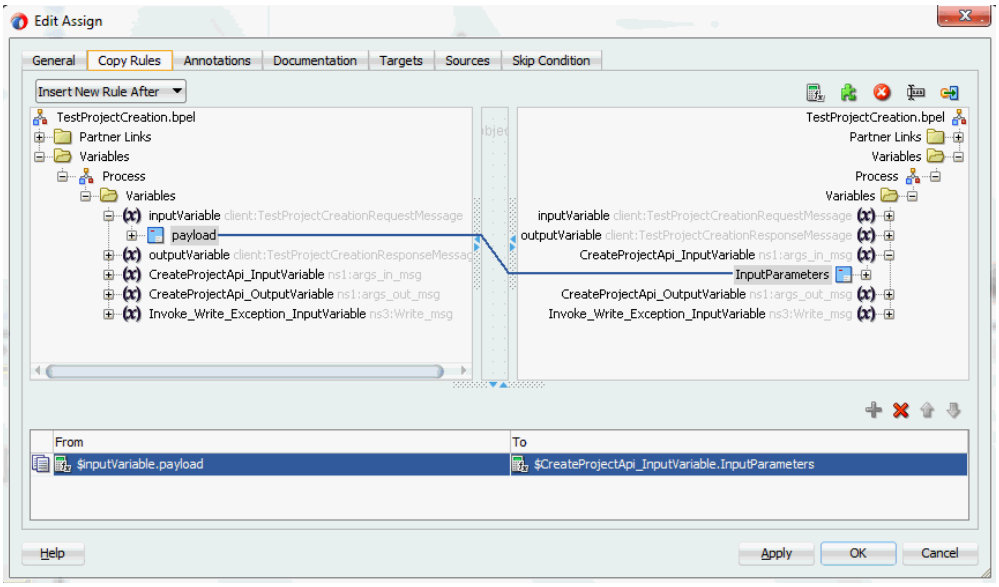
3. Click the General tab to enter the name for the Assign activity, such as 'AssignInput'.

4. Select the Copy Rules tab and expand the source and target trees:

   - In the From navigation tree, navigate to **Variables > Process > Variables > inputVariable** and select **payload**.

   - In the To navigation tree, navigate to **Variables > Process > Variables > CreateProjectApi_InputVariable** and select **InputParameters**.

   Drag the source node (payload) to connect to the target node (InputParameters) that you just identified. This creates a line that connects the source and target nodes. The copy rule is displayed in the From and To sections at the bottom of the Edit Assign dialog box.



5. The Edit Assign dialog box appears.

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

To add the second Assign activity:

1. Add the second Assign activity by dragging and dropping the **Assign** activity from the **BPEL Constracts** section of the Component Palette into the center swim lane of the process diagram, between the **Invoke** and **replyOutput** activities.

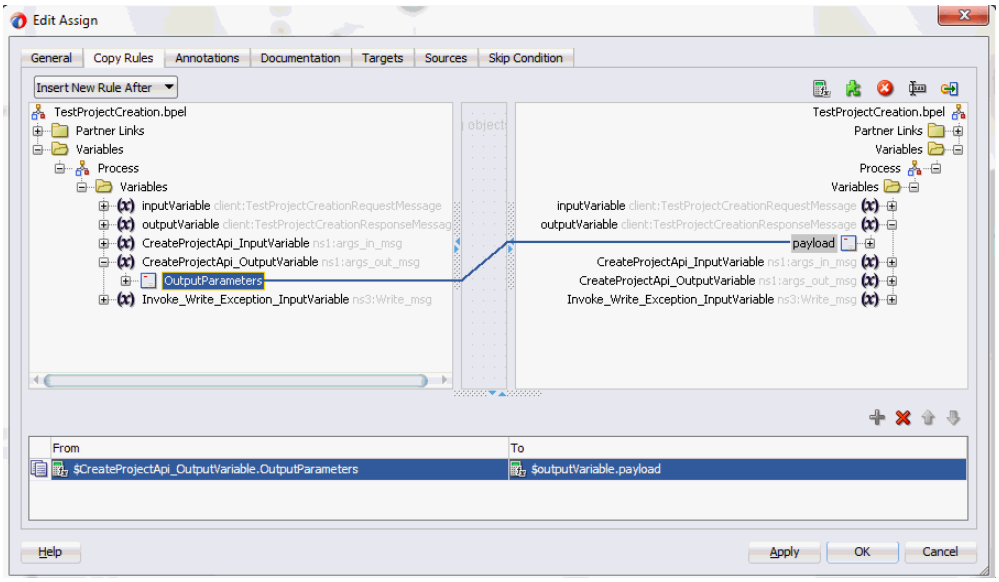
2. Repeat Step 2 to Step 3 described in creating the first Assign activity to add the second Assign activity called 'AssignOuput'.

3. Select the Copy Rules tab and expand the source and target trees:

   - In the From navigation tree, navigate to **Variables > Process > Variables > CreateProjectApi_OutputVariable** and select **OutputParameters**.

   - In the To navigation tree, navigate to **Variables > Process > Variables > outputVariable** and select **payload**.

   Drag the source node (OutputParameters) to connect to the target node (payload) that you just identified. This creates a line that connects the source and target nodes. The copy rule is displayed in the From and To sections at the bottom of the Edit Assign dialog box.



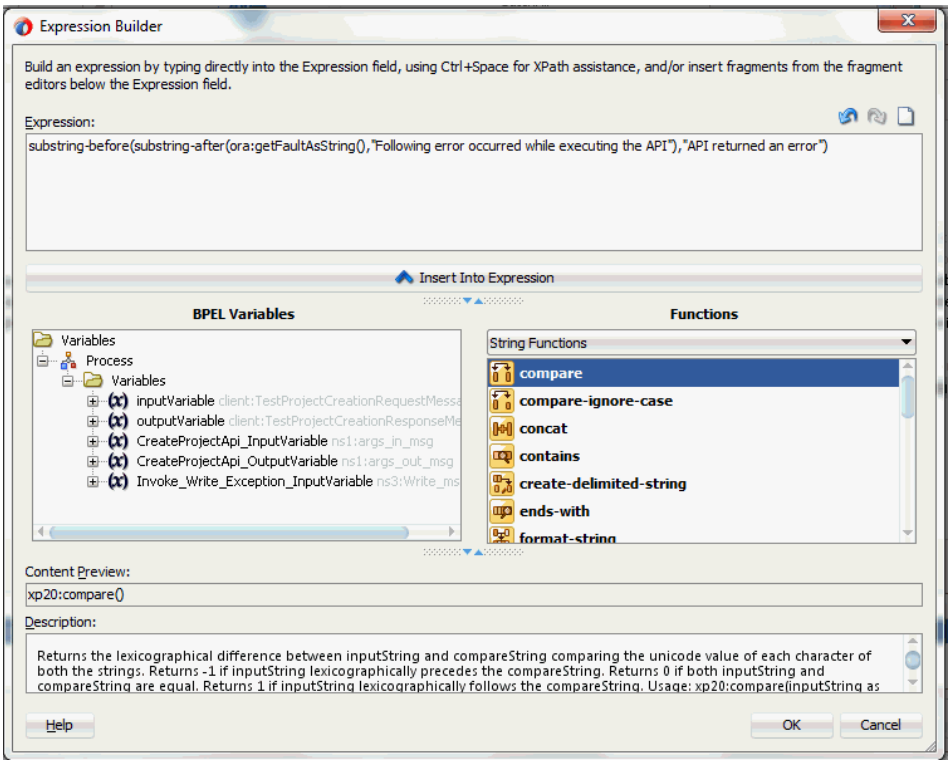
4. Click **Apply** and then **OK** to complete the configuration of the **Assign** activity.

To add the third Assign activity in the CatchAll activity:

1. Add the third Assign activity by dragging and dropping the **Assign** activity from the **BPEL Constracts** section of the Component Palette into the CatchAll process diagram, before the **Invoke** activity.

2. Repeat Step 2 to Step 3 described in creating the first Assign activity to add the second Assign activity.

3. In the Edit Assign dialog, enter the first parameter:
   - Click the **Expression** icon to invoke the Expression Builder dialog.

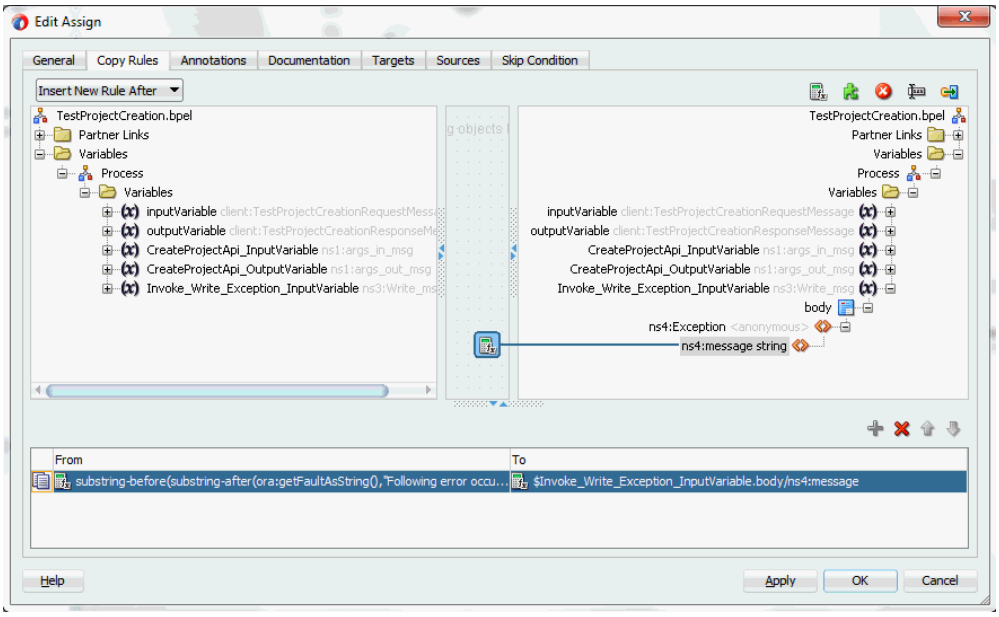


Enter `substring-before(substring-after(ora:getFaultAsString(),"Following error occurred while executing the API"),"API returned an error")` in the Expression box. Click **OK**.

The Expression icon with the expression value appears in the center of the Edit Assign dialog, between the From and To navigation tree nodes.

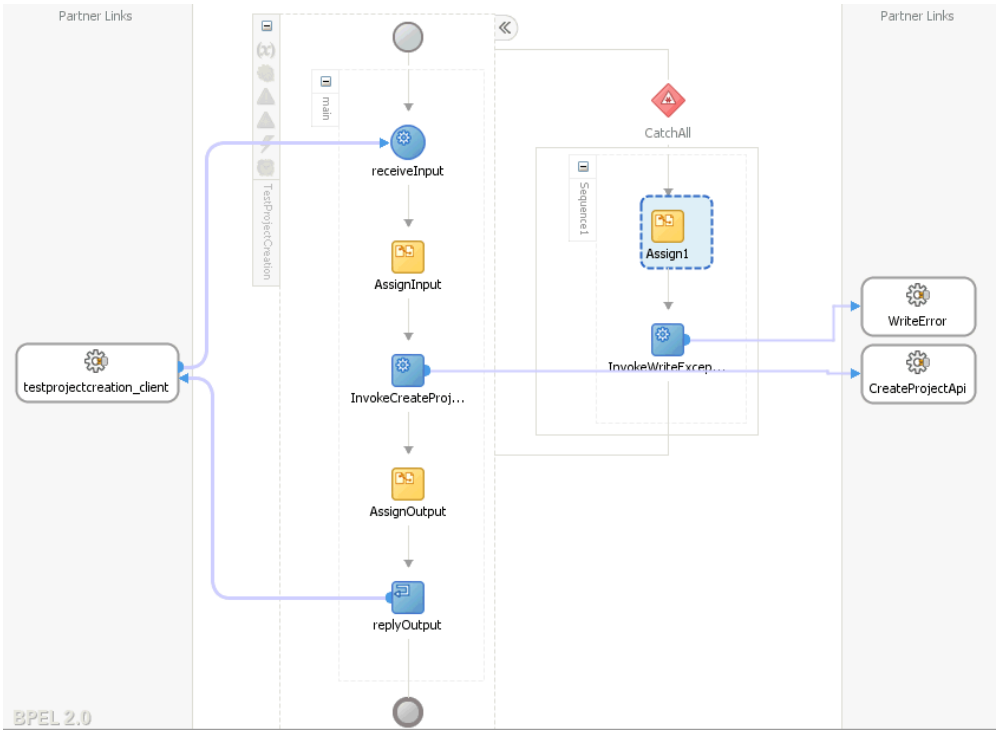
- In the To navigation tree, navigate to **Variables > Process > Variables > Invoke_Write_Exception_InputVariable > body > ns4:Exception** and select **ns4: message string**.

Drag the Expression icon to connect to the target node (ns4: message string) that you just identified. This creates a line that connects the source and target nodes. The copy rule is displayed in the From and To sections at the bottom of the Edit Assign dialog box.

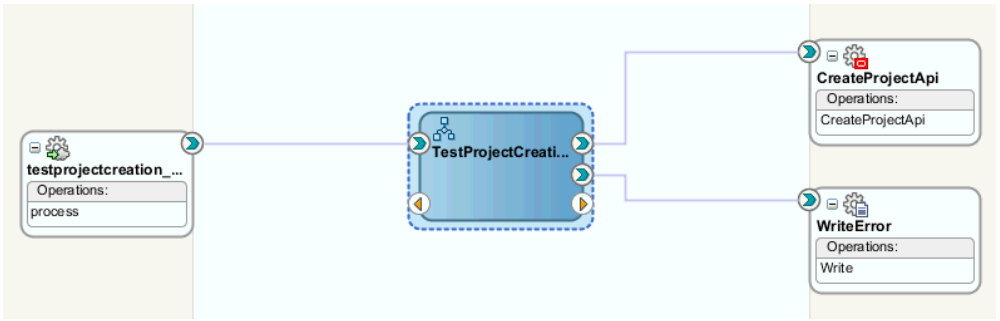


4. Click **Apply** and then **OK** to complete the configuration of the **Assign** activity.

Complete BPEL Process Flow



Click the `TestProjectCreation` to display the Oracle JDeveloper composite diagram:



Validating and Testing SOA Composite Application with BPEL Process

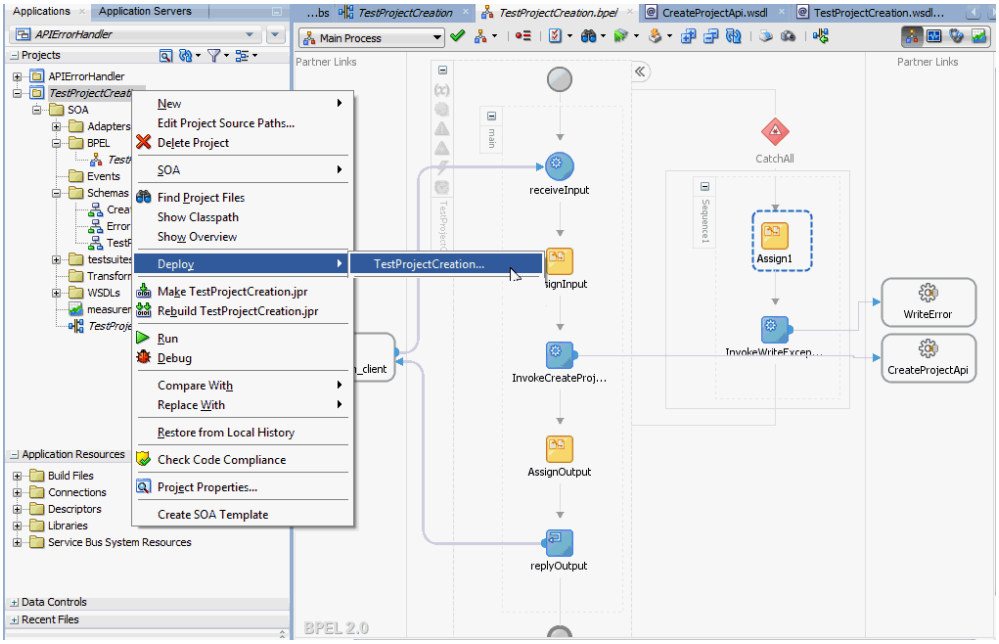
Deploying the SOA Composite Application with BPEL Process

After creating the BPEL process contained in the SOA Composite application at the design time, you need to deploy the SOA Composite with BPEL process first and then manually test the process in the Oracle Enterprise Manager Fusion Middleware Control Console (`http://<servername>:<portnumber>/em`).

Important: Before deploying the SOA Composite with BPEL process using Oracle JDeveloper, you must have established the connectivity between the design-time environment and the runtime server. Ensure that you have configured Oracle E-Business Adapter on your Oracle SOA Suite server to handle the service invocation. For configuration details, see:

- Configuring the Data Source in Oracle WebLogic Server, page 11-1
- Creating Connection Factory for Oracle E-Business Suite Adapter, page 11-7
- Creating the Application Server Connection, page 11-10

1. To deploy the SOA Composite application, in Oracle JDeveloper select the SOA Composite project in the Applications window. Right-click the project name, such as TestProjectCreation, and then select **Deploy > [project name] > [serverConnection]** from the menu that appears.



For example, you can select **Deploy > TestProjectCreation > soa-server1** if you have the server connection set up properly.

2. In the Select Server page, select 'soa-server1' that you have established the server connection earlier. Click **Next**.

3. In the SOA Servers page, accept the default target SOA Server ('soa-server1')

selection.

Click **Next** and **Finish** to start the deployment process.

You can check for successful compilation in the SOA - Log window, and verify that the deployment is successful in the Deployment - Log window.

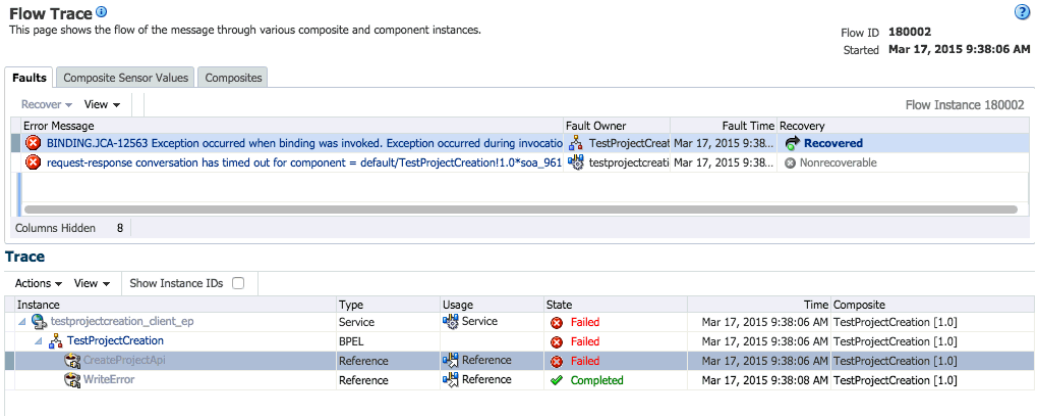
Validating and Testing the Deployed SOA Composite

In the Oracle Enterprise Manager Fusion Middleware Control Console, from the Farm base domain, expand the **SOA >soa-infra** to navigate through the SOA Infrastructure home page and menu to access your deployed SOA composite applications running in the SOA Infrastructure for that managed server. Click the SOA Composite application that you want to initiate from the SOA Infrastructure. Click **Test** at the top of the page.

In the Test Web Service page, specify the XML payload data to use in the Input Arguments section. Enter the input string required by the process and click **Test Web Service** to initiate the process.

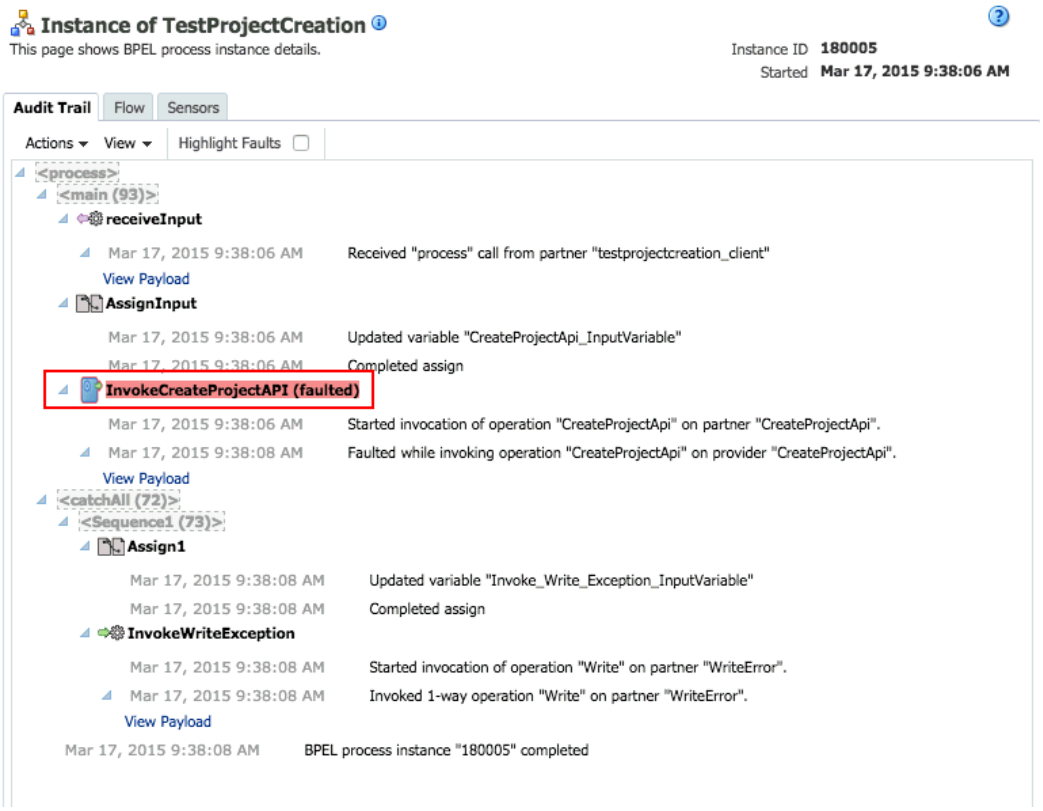
The test results appear in the Response tab upon completion.

In the Response tab, click the **Launch Message Flow Trace** link to view the result details. The Flow Trace page appears.



If any error occurred during the test, you should find it in the Faults tab.

Click the TestProjectCreation link in the Trace region to view the invocation details in the Audit Trail tab.



Click the Flow tab to check the BPEL process flow diagram. Click an activity of the process diagram to view the activity details and flow of the payload through the process.

For example, click the "InvokeCreateProjectAPI (faulted)" to view the faults.

```
BINDING.JCA-12563
Exception occurred when binding was invoked.
Exception occurred during invocation of JCA binding: "JCA Binding
execute of Reference operation 'CreateProjectApi' failed due to: API
Execution Error.
Following error occurred while executing the API
"PA_PROJECT_PUB.CREATE_PROJECT": "1.
PAPA_FUNCTION_SECURITY_ENFORCEDNFND_ERROR_LOCATION_FIELDNFND_MESSAGE_TYP
EE
".
API returned an error.
Correct the inputs for the API call. Contact oracle support if error is
not fixable.
".
The invoked JCA adapter raised a resource exception.
Please examine the above error message carefully to determine a
resolution.
```

Additionally, verify the faults collected through the file adapter partner link during the service invocation. For example, go to the outputDir (`/usr/tmp/`) you specified for the write operation during the file adapter partner link creation. Open the output file (such as pa_%SEQ%_%yyMMddHHmmss%.xml) which contains the following faults:

```
<part name="detail">
<detail>API  Execution Error.
Following error occurred while executing the API
"PA_PROJECT_PUB.CREATE_PROJECT": "
1. PA PA_FUNCTION_SECURITY_ENFORCED N
FND_ERROR_LOCATION_FIELD
FND_MESSAGE_TYPE E".
API returned an error.
Correct the inputs for the API call. Contact oracle support if error is
not fixable.
</detail>
</part>
<part name="code">
<code>EBzA-201</code>
</part>
```

Sample Payload for Creating a Project

The following information shows the sample payload in creating a project in Oracle E-Business Suite:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
   <ns1:InputParameters
xmlns:ns1="http://xmlns.oracle.com/pcbpel/adapter/db/sp/EBSReference">
    <ns1:P_API_VERSION_NUMBER>1.0</ns1:P_API_VERSION_NUMBER>
    <ns1:P_COMMIT>F</ns1:P_COMMIT>
    <ns1:P_INIT_MSG_LIST>T</ns1:P_INIT_MSG_LIST>
    <ns1:P_PM_PRODUCT_CODE>MSPROJECT</ns1:P_PM_PRODUCT_CODE>
    <ns1:P_PROJECT_IN>
     <ns1:PM_PROJECT_REFERENCE>AGL-AMG Project
6</ns1:PM_PROJECT_REFERENCE>
     <ns1:PROJECT_NAME>AGL-AMG Project 6</ns1:PROJECT_NAME>
     <ns1:PA_PROJECT_NUMBER>SAGIR-P6</ns1:PA_PROJECT_NUMBER>
     <ns1:CREATED_FROM_PROJECT_ID>1005</ns1:CREATED_FROM_PROJECT_ID>

<ns1:CARRYING_OUT_ORGANIZATION_ID>478</ns1:CARRYING_OUT_ORGANIZATION_ID>
     <ns1:PROJECT_STATUS_CODE>ACTIVE</ns1:PROJECT_STATUS_CODE>
     <ns1:DESCRIPTION>TaskDesc</ns1:DESCRIPTION>
     <ns1:START_DATE>2014-01-01T00:00:00</ns1:START_DATE>
     <ns1:COMPLETION_DATE>2018-12-31T00:00:00</ns1:COMPLETION_DATE>

<ns1:PROJECT_RELATIONSHIP_CODE>Primary</ns1:PROJECT_RELATIONSHIP_CODE>
    </ns1:P_PROJECT_IN>
    <ns1:P_KEY_MEMBERS>
     <ns1:P_KEY_MEMBERS_ITEM>
      <ns1:PERSON_ID>2960</ns1:PERSON_ID>
      <ns1:PROJECT_ROLE_TYPE>PROJECT MANAGER</ns1:PROJECT_ROLE_TYPE>
     </ns1:P_KEY_MEMBERS_ITEM>
    </ns1:P_KEY_MEMBERS>
    <ns1:P_CLASS_CATEGORIES>
     <ns1:P_CLASS_CATEGORIES_ITEM>
      <ns1:CLASS_CATEGORY>Construction</ns1:CLASS_CATEGORY>
      <ns1:CLASS_CODE>New Building</ns1:CLASS_CODE>
     </ns1:P_CLASS_CATEGORIES_ITEM>
    <ns1:P_CLASS_CATEGORIES>
    <ns1:P_TASKS_IN>
     <ns1:P_TASKS_IN_ITEM>
      <ns1:PM_TASK_REFERENCE>1</ns1:PM_TASK_REFERENCE>
      <ns1:PA_TASK_NUMBER>1</ns1:PA_TASK_NUMBER>
      <ns1:TASK_DESCRIPTION>Plant function</ns1:TASK_DESCRIPTION>
      <ns1:PM_PARENT_TASK_REFERENCE></ns1:PM_PARENT_TASK_REFERENCE>
     </ns1:P_TASKS_IN_ITEM>
       <ns1:P_TASKS_IN_ITEM>
      <ns1:PM_TASK_REFERENCE>1.1</ns1:PM_TASK_REFERENCE>
      <ns1:PA_TASK_NUMBER>1.1</ns1:PA_TASK_NUMBER>
      <ns1:TASK_DESCRIPTION>Plant function</ns1:TASK_DESCRIPTION>
      <ns1:PM_PARENT_TASK_REFERENCE>1</ns1:PM_PARENT_TASK_REFERENCE>
     </ns1:P_TASKS_IN_ITEM>
    </ns1:P_TASKS_IN>
    <ns1:P_ORG_ROLES/>
    <ns1:P_STRUCTURE_IN/>
    <ns1:P_EXT_ATTR_TBL_IN/>
   </ns1:InputParameters>
 </soap:Body>
</soap:Envelope>
```

Secured Connection Between Oracle E-Business Suite and Oracle Fusion Middleware 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 Apps/Apps-equivalent schema's username and password, just FND username and password (concept of Oracle E-Business Suite 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.

Oracle E-Business Suite Adapter authenticates users at run time and gets the connection to Oracle E-Business Suite databases through the use of J2EE data sources. This approach is native to Oracle E-Business Suite in defining the connection pool to access the application database.

Account details information including application login user name and password that was required as part of the configuration for database connection is 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 Oracle E-Business Suite Adapter.

To enable the native Oracle E-Business Suite connectivity using J2EE data sources feature,

> **Note:** To have this feature available, 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.
>
> Additionally, you must apply necessary patches to enable the connectivity between Oracle E-Business Suite and an external

application server. See My Oracle Support Knowledge Document 787637.1 for details.

5

Using XML Gateway

This chapter covers the following topics:

- Overview of XML Gateway
- Design-Time Tasks for XML Gateway Inbound Messaging
- Creating a New SOA Composite Application with BPEL Process
- Creating a Partner Link
- Adding a Partner Link for the File Adapter
- Configuring the Invoke Activities
- Configuring the Assign Activity
- Run-Time Tasks for XML Gateway Inbound Messaging
- Deploying the SOA Composite Application with BPEL Process
- Testing the SOA Composite with BPEL Process
- Verifying Records in Oracle E-Business Suite
- Design-Time Task for XML Gateway Outbound Messaging
- Creating a SOA Composite Application with BPEL Process
- Adding a Partner Link
- Adding a Receive Activity
- Adding a Partner Link for File Adapter
- Adding an Invoke Activity
- Adding an Assign Activity
- Run-Time Task for XML Gateway Outbound Messaging
- Deploying the SOA Composite Application with BPEL Process
- Testing the SOA Composite Application with BPEL Process
- Troubleshooting

Overview of XML Gateway

Oracle E-Business Suite Adapter provides a bridge between Oracle E-Business Suite and third party applications. Inbound and outbound XML data is exchanged between Oracle E-Business Suite and third party applications through the XML Gateway.

Oracle XML Gateway provides a common, standards-based approach for XML integration between Oracle E-Business Suite 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 E-Business Suite and third party applications. Oracle E-Business Suite utilize the Oracle Workflow Business Event System to support event-based XML message creation and consumption.

Oracle E-Business Suite Adapter 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 E-Business Suite 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.

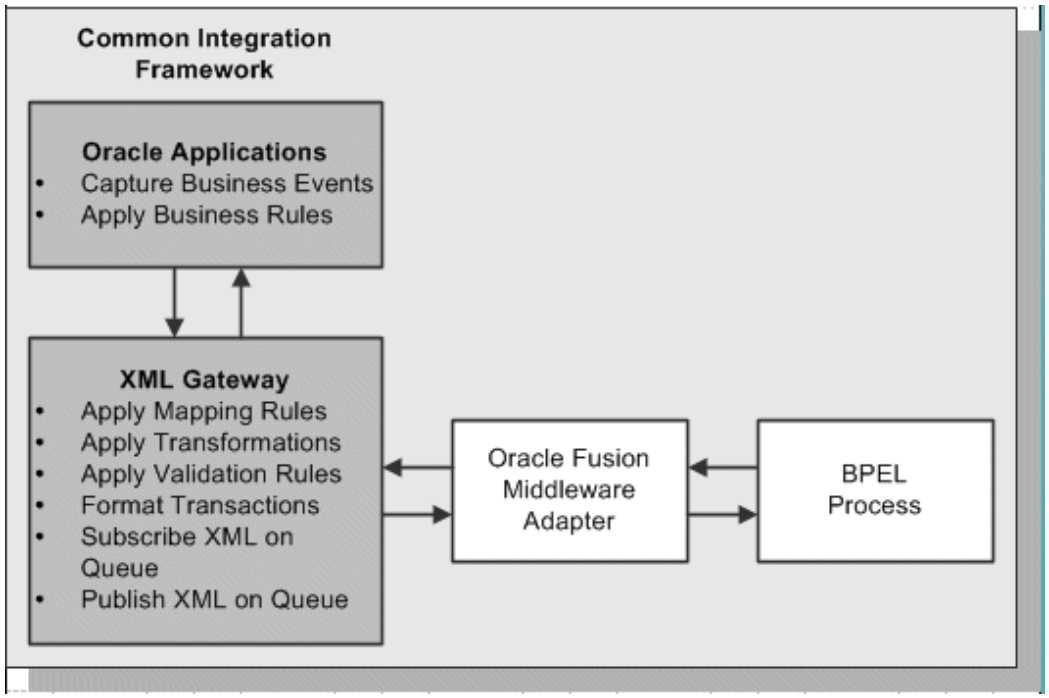
Integration Architecture

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 E-Business Suite. 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 E-Business Suite by allowing you to insert and retrieve data from Oracle E-Business Suite. Oracle E-Business Suite 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 E-Business Suite. 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 from Oracle E-Business Suite. The retrieved data can be passed to trading partners or consumers who subscribed to the message.

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 E-Business Suite application products and the XML Gateway.

Inbound Queues

Inbound message queues are used for XML messages inbound into Oracle E-Business Suite. 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 5-4. 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 E-Business Suite. 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.
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 part of the Oracle E-Business Suite Documentation Library which can be accessed on the Oracle Technology Network (OTN).

Attribute	Description
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 other XML Gateway information. This guide is part of the Oracle E-Business Suite Documentation Library which can be accessed on the Oracle Technology Network (OTN).

Design-Time Tasks for XML Gateway Inbound Messaging

This section describes configuring Oracle E-Business Suite Adapter to use XML Gateway. It describes the tasks required to configure Oracle E-Business Suite Adapter using the Adapter Configuration Wizard in Oracle JDeveloper.

SOA Composite Application with BPEL Process Scenario

Take the XML Gateway Inbound Process PO XML Transaction as an example to explain the BPEL process creation. In this example, the XML Gateway inbound message map is exposed as a Web service through PROCESS_PO_007 inbound map. It allows sales order data including header and line items to be inserted into Order Management system while an associated purchase order is created.

When a purchase order is sent by a trading partner, the purchase order data is used as input to the BPEL process along with ECX Header properties such as

jca.apps.ecx.TransactionType. The BPEL process then pushes this purchase order in 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 order data to Order Management application.

When the SOA Composite application with BPEL process has been successfully executed after deployment, you should get the same order information inserted into the Order Management table once a purchase order is created.

Prerequisites to Configure XML Gateway Inbound

Setting XML Gateway Header Properties

You need to populate certain variables in the BPEL process in order to provide context information for Oracle E-Business Suite applications. This is accomplished by setting individual fields as header properties from the SYSTEM.ECXMSG object used in earlier releases.

The header properties can include, for example, jca.apps.ecx.TransactionType, jca.apps.ecx.TransactionSubtype, jca.apps.ecx.PartySiteId, jca.apps.ecx.MessageType, jca.apps.ecx.MessageStandard, jca.apps.ecx.DocumentNumber that you need to populate for the XML transaction to complete successfully.

These header property values can be defined through the Properties tab of an Invoke activity. See Configure the Invoke activity, page 5-24 for more information.

Setting Up XML Gateway Trading Partner

You need to ensure that you have defined a valid inbound XML Gateway trading partner in the Trading Partner Setup form through XML Gateway responsibility.

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 inbound transaction:

- Transaction type: ONT
- Transaction sub type: POI
- Standard Code: OAG
- External transaction type: PO
- External transaction subtype: PROCESS
- Direction: IN
- Map: ONT_3A4R_OAG72_IN
- Source Trading partner location code: BWSANJOSE

Ensuring Agent Listeners Are All Up and Running

You also need to configure and schedule two listeners on the Oracle E-Business Suite application side. These are the ECX Inbound Agent Listener and the ECX Transaction Agent Listener. Use the following steps to configure these listeners in Oracle E-Business Suite:

1. Log in to Oracle E-Business Suite 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**, **ECX Transaction Agent Listener**, and the **Workflow Deferred Agent Listener**. Select the listener, and select Start from the **Actions** box. Click **Go**.

Based on the XML Gateway Inbound Process PO XML Transaction business scenario, the following design-time tasks are discussed in this chapter:

1. Create a new SOA Composite application with BPEL process, page 5-8.
2. Create a partner link, page 5-13.
3. Add partner links for file adapter, page 5-18.
4. Configure Invoke activities, page 5-24.
5. Configure an Assign activity, page 5-31.

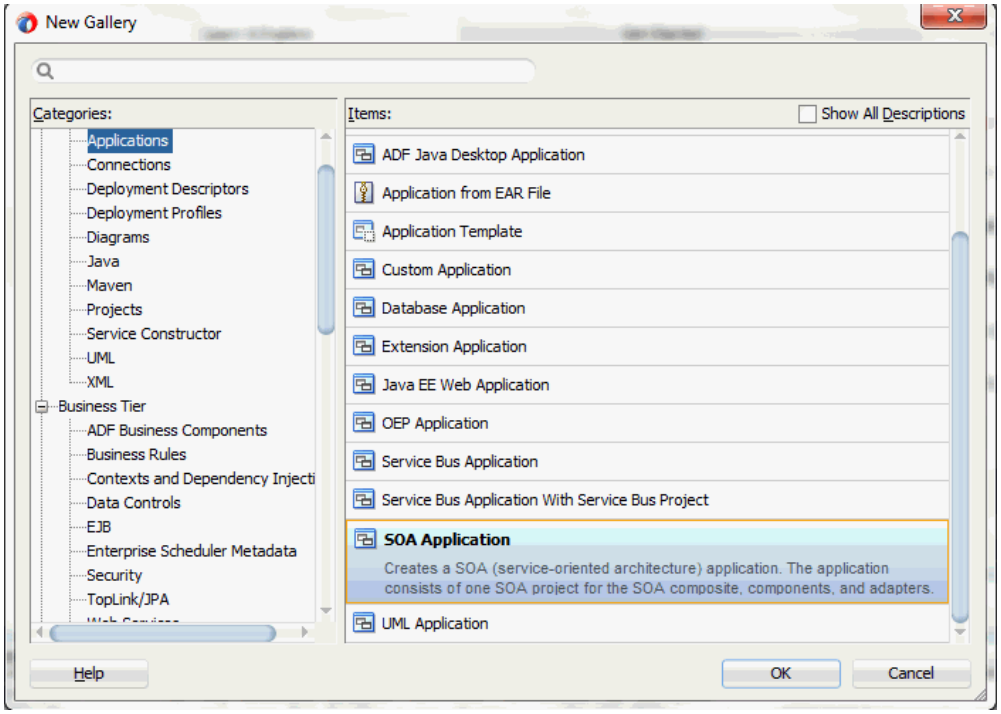
Creating a New SOA Composite Application with BPEL Process

To create a new SOA Composite application with BPEL process:

1. Launch Oracle JDeveloper.
2. Click **New Application** in the Application Navigator.

   The New Gallery page appears. Select **SOA Application** from the Items list.

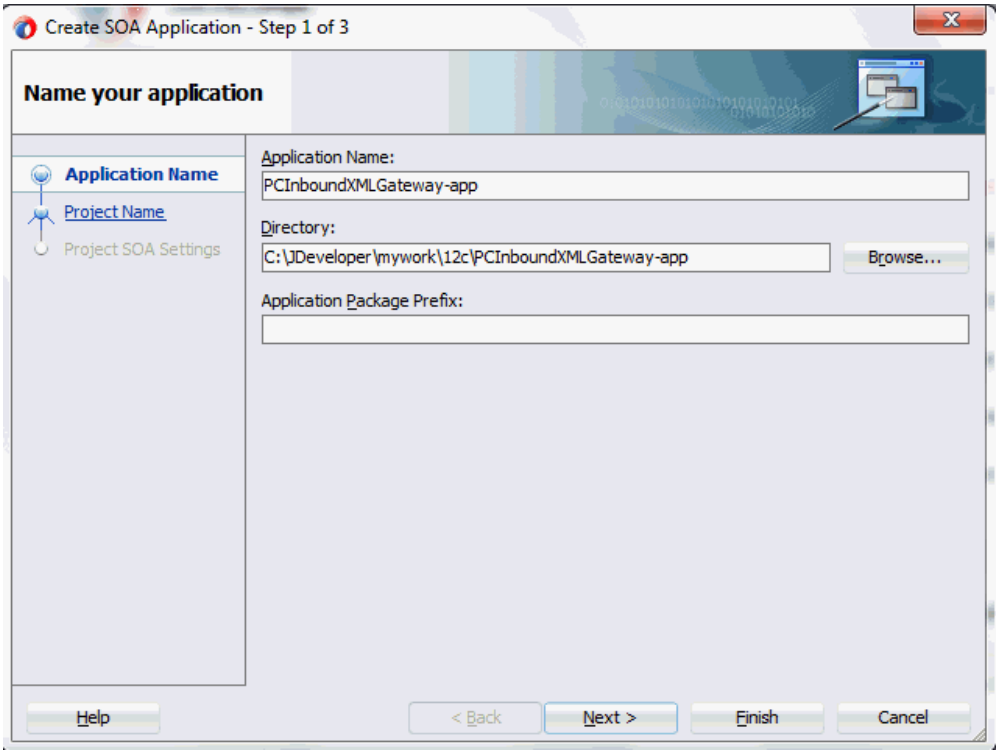
The New Gallery Page



Click **OK**.

3. The Create SOA Application - Name your application page is displayed.

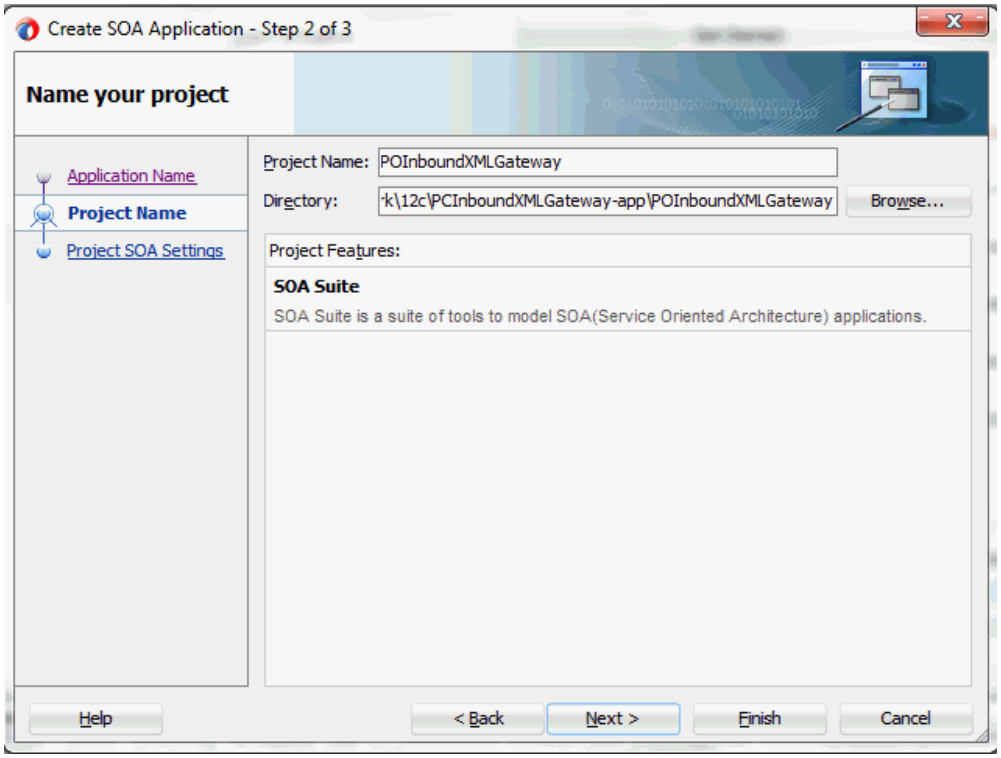
The Create SOA Application - Name your application Page



4. Enter an appropriate name for the application in the **Application Name** field.

   Click **Next**. The Create SOA Application - Name your project page is displayed.

The Create SOA Application - Name your project Page

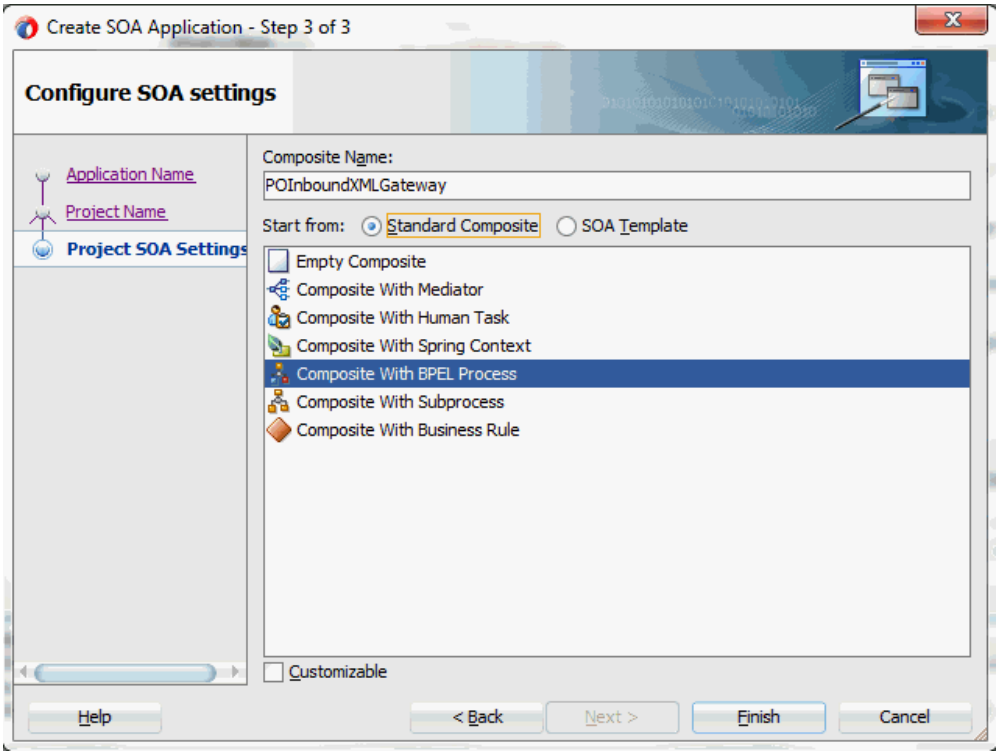


5. Enter an appropriate name for the project in the **Project Name** field. For example, `POInboundXMLGateway`.

   Notice that **SOA Suite** is shown in the Project Features section.

6. Click **Next**. The Create SOA Application - Configure SOA settings page is displayed.

The Create SOA Application - Configure SOA settings Page

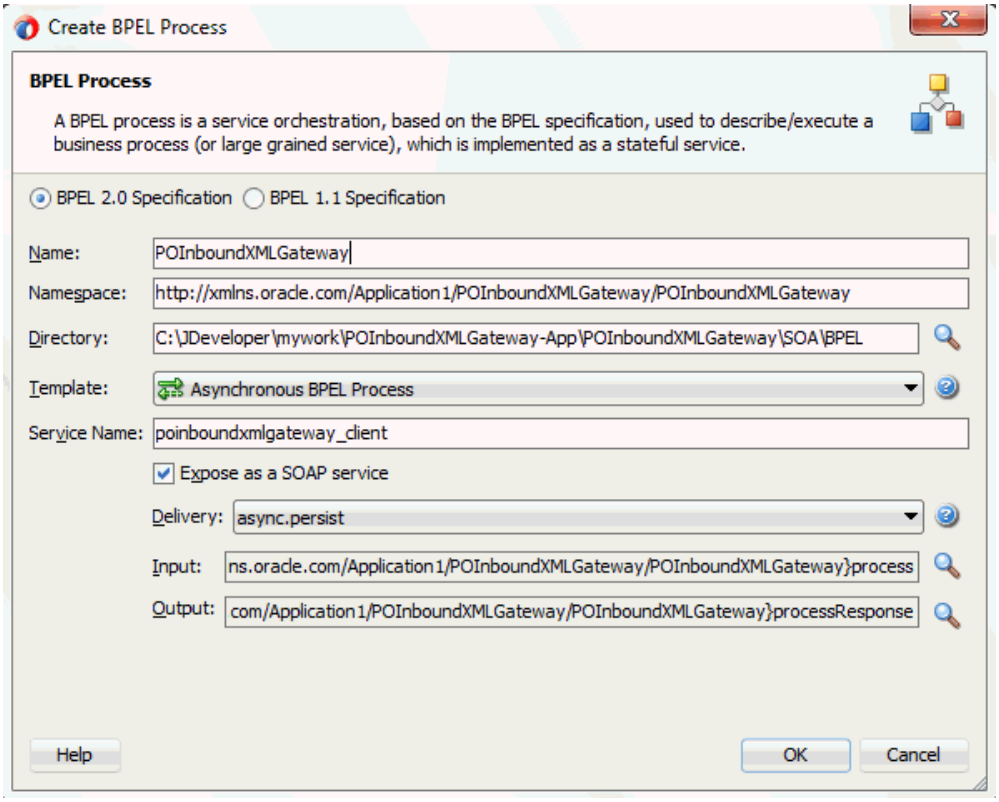


7. Leave the default **Standard Composite** selection unchanged for the Start from field.

   Select **Composite With BPEL Process** from the Standard Composite list, and then click **Finish**. You have created a new application, and a SOA project. This automatically creates a SOA Composite.

   The Create BPEL Process page is displayed.

The Create BPEL Process Page



8. Leave the default **BPEL 2.0 Specification** selection unchanged. This creates a BPEL project that supports the BPEL 2.0 specification.

   Enter an appropriate name for the BPEL process in the **Name** field. For example, `POInboundXMLGateway`.

   Select **Asynchronous BPEL Process** in the **Template** field. Click **OK**.

   An asynchronous BPEL process is created with the Receive and Reply activities. The required source files including bpel and wsdl, using the name you specified (for example, `POInboundXMLGateway.bpel` and `POInboundXMLGateway.wsdl`) and `POInboundXMLGateway` (composite.xml) are also generated.

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 add a partner link:

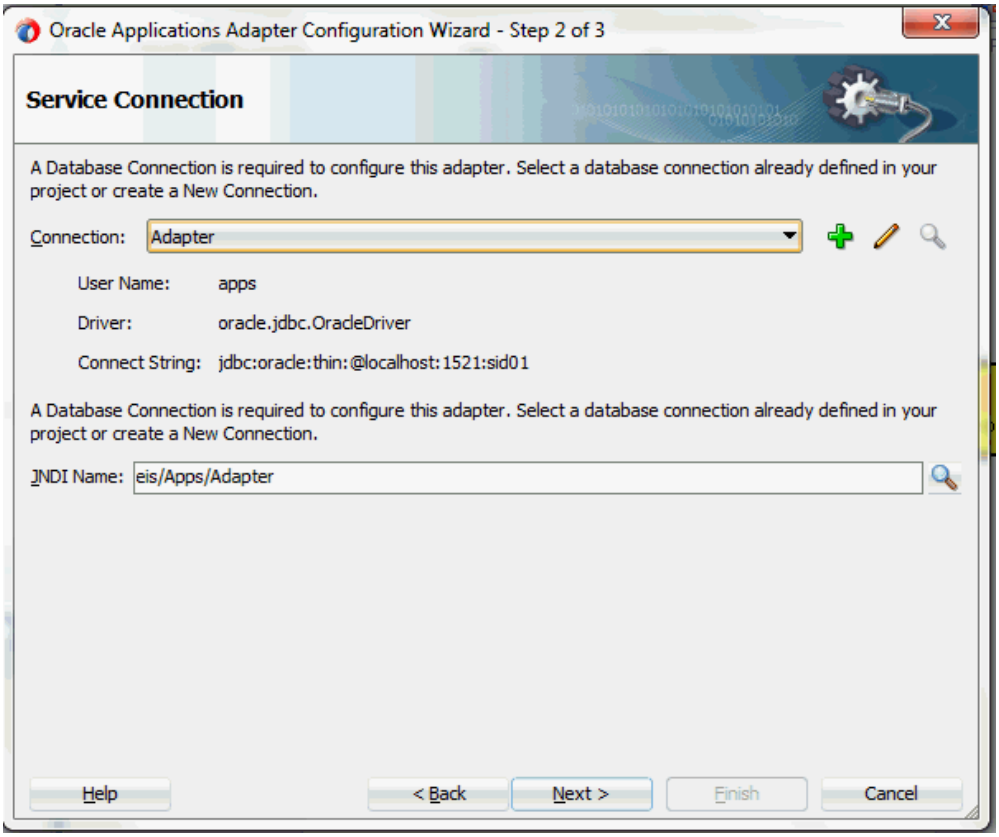
1. Click **BPEL Services** in the Component palette.

Drag and drop **Oracle E-Business Suite** (formerly known as **Oracle Applications**) from the **BPEL Services** list into the right Partner Link swim lane of the process diagram. The Oracle E-Business Suite Reference page appears.

Enter a reference name in the **Name** field. For example, `XMLGatewayOrderInbound`.

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

Specifying a Database Service Connection



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:**

  Complete the following steps to create a new database connection:

  1. From the Service Connection page, click the **Create a New Database Connection** icon.

The Create Database Connection page appears.

2. Enter the following information:

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

    2. From the **Connection Type** list, select the type of connection for your database (such as `Oracle (JDBC)`).

    3. In the **UserName** field, specify a unique name for the database connection.

    4. In the **Password** field, specify a password for the database connection.

    5. In the Oracle (JDBC) Settings, from the **Driver** list, select **Thin**.

    6. In the **Host Name** field, specify the host name for the database connection.

    7. In the **JDBC Port** field, specify the port number for the database connection.

    8. In the **SID** field, specify the unique SID value for the database connection.

3. Click **Test Connection** to determine whether the specified information establishes a connection with the database.

    The status message "`Success!`" indicates a valid connection.

    Click **OK**.

4. The Service Connection dialog box appears, providing a summary of your database connection.

- **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 page, select an appropriate connection from the **Connection** drop-down list.

  2. 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 Oracle E-Business Suite Adapter 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 Fusion Middleware User's Guide for Technology Adapters*.

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

   Click **Next**.

   If you are connecting to Oracle E-Business Suite Release 12, then the **IREP File not present** dialog appears indicating that Adapter 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 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.

     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.

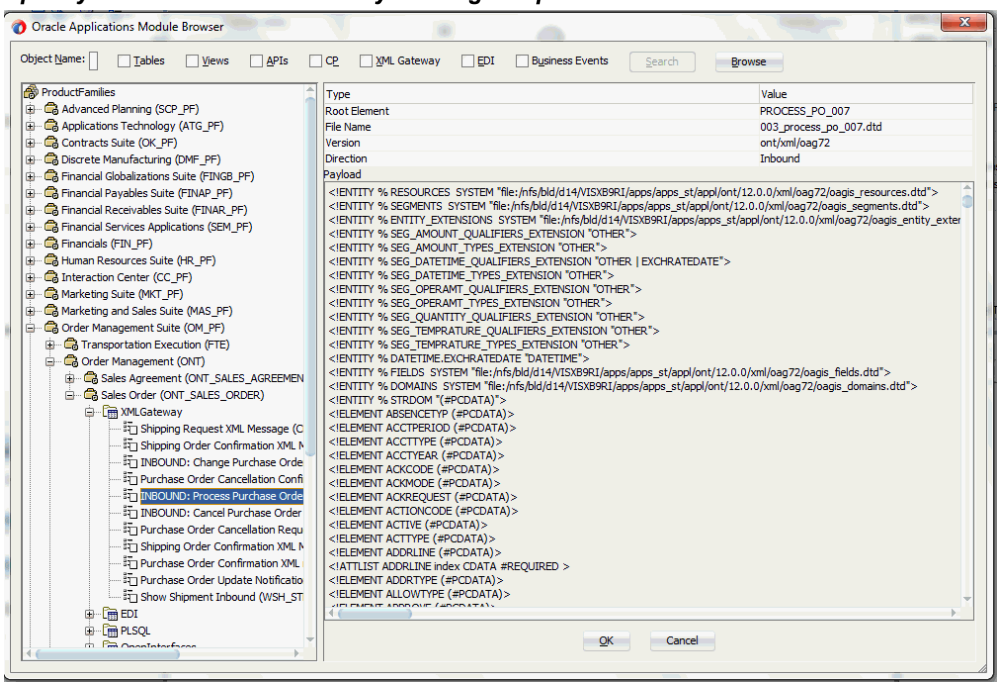
   - Click **No** to query the Integration Repository data file from the live database you are connecting to display the Integration Repository tree.

   In this example, click **No**. Click **Next** in the Operation page to open the Oracle E-Business Suite Module Browser.

5. The Oracle E-Business Suite Module Browser combines interface data from Oracle Integration Repository with information about the additional interfaces supported

by Oracle E-Business Suite Adapter, organized in a tree hierarchy.

Specify an Inbound XML Gateway Message Map from the Module Browser

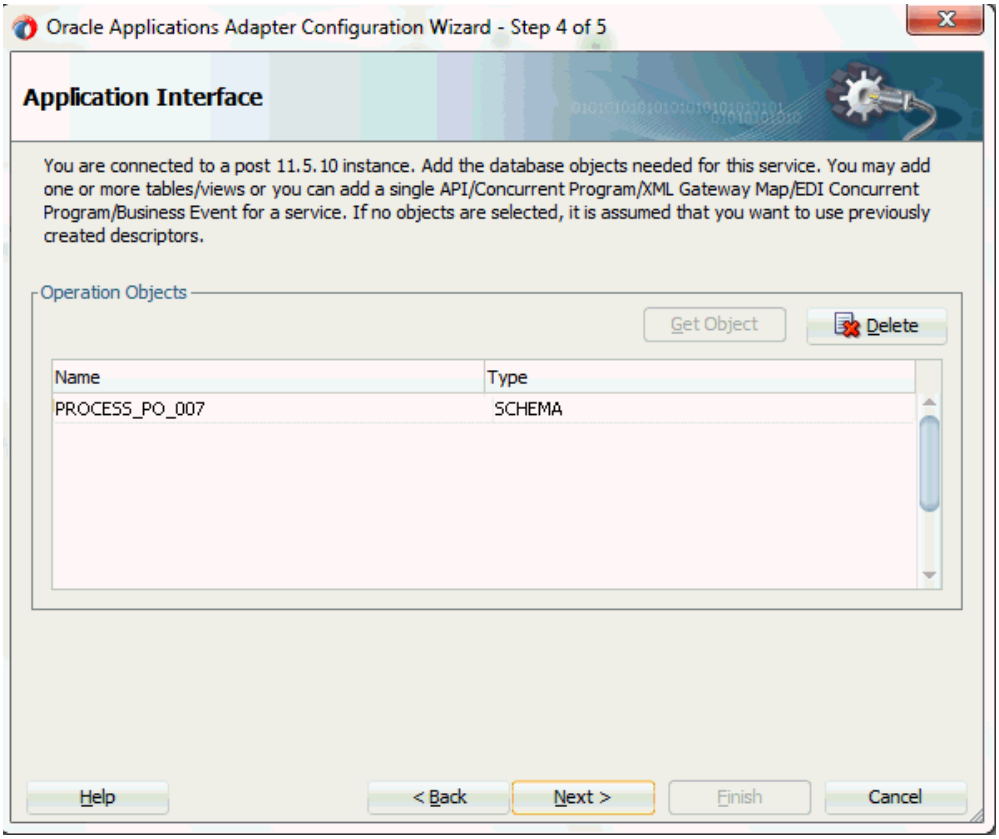


> **Note:** The Oracle E-Business Suite Module Browser includes the various product families that are available in Oracle E-Business Suite. Each product family contains the individual products. Each product contains the business entities associated with the product. Business entities contain the various 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)`.

6. Click **OK**.

Adapter Configuration Wizard - Application Interface Page



Click **Next**.

7. Click **Finish** to complete the process of configuring Oracle E-Business Suite Adapter. The partner link is created with the required WSDL settings. The wizard also generates the WSDL file corresponding to the XML schema. This WSDL file is now available for the partner link.

   Double click the newly generated Partner Link (XMLGatewayOrderInbound) icon to view the partner link details.

Adding a Partner Link for the File Adapter

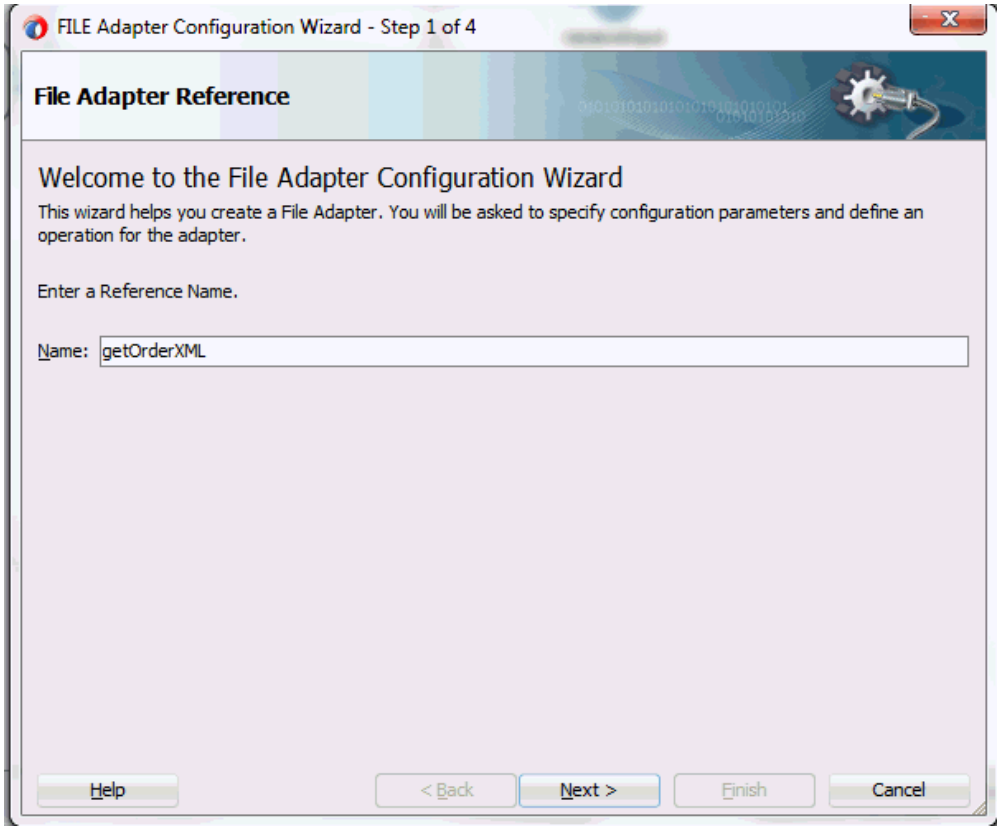
If you are configuring an inbound message, then you would need to add another partner link for the file adapter. This allows the inbound message to pick up an XML file received from the third party application. The data is inserted into Oracle E-Business Suite through the partner link that was configured earlier.

To add a partner link for the file adapter to get the XML Message:

1. In JDeveloper BPEL Designer, click **BPEL Services** in the Component palette.

Drag and drop **File Adapter** from the BPEL Services list into the right Partner Link swim lane of the process diagram. The File Adapter Reference page appears.

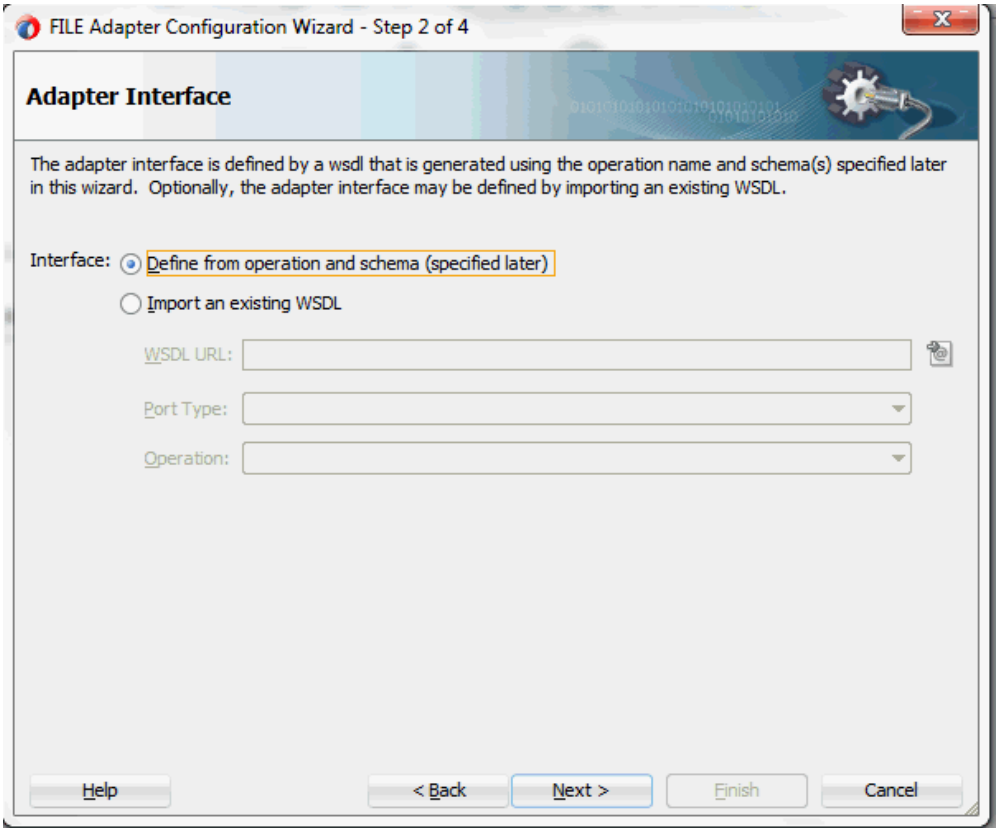
The File Adapter Reference Page



Enter a name for the file adapter service, for example, `getOrderXML`.

2. Click **Next**. The Adapter Interface page appears.

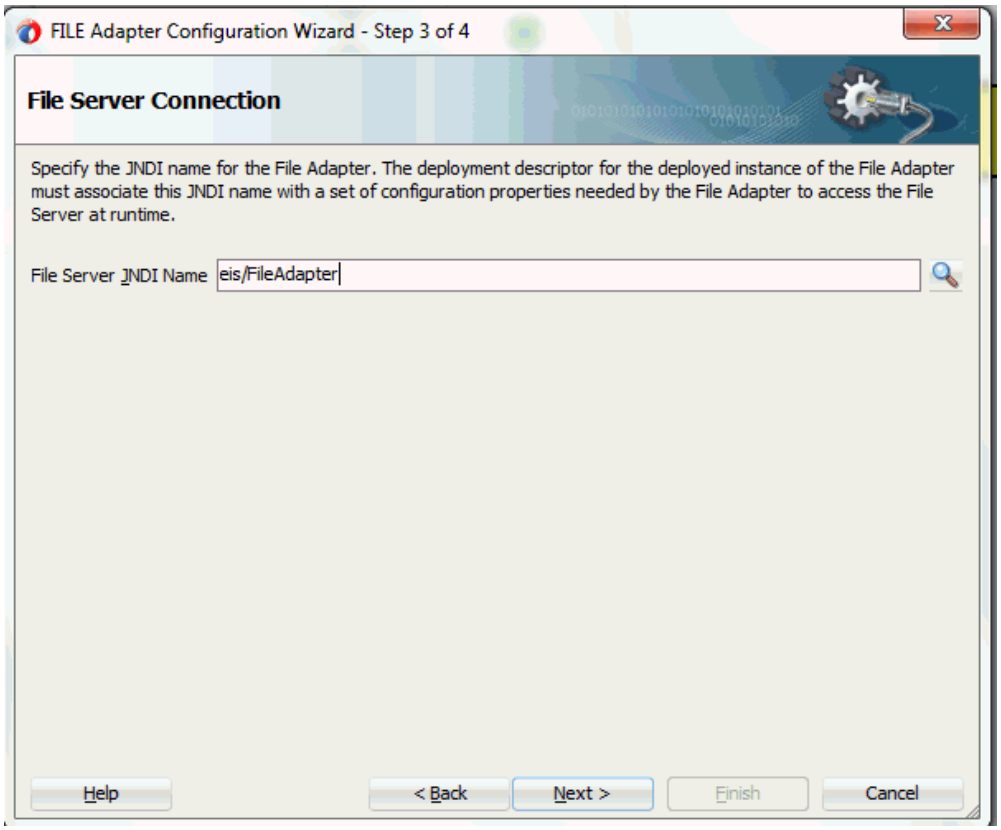
Specifying the Adapter Interface



Select the **Define from operation and schema (specified later)** radio button and click **Next**.

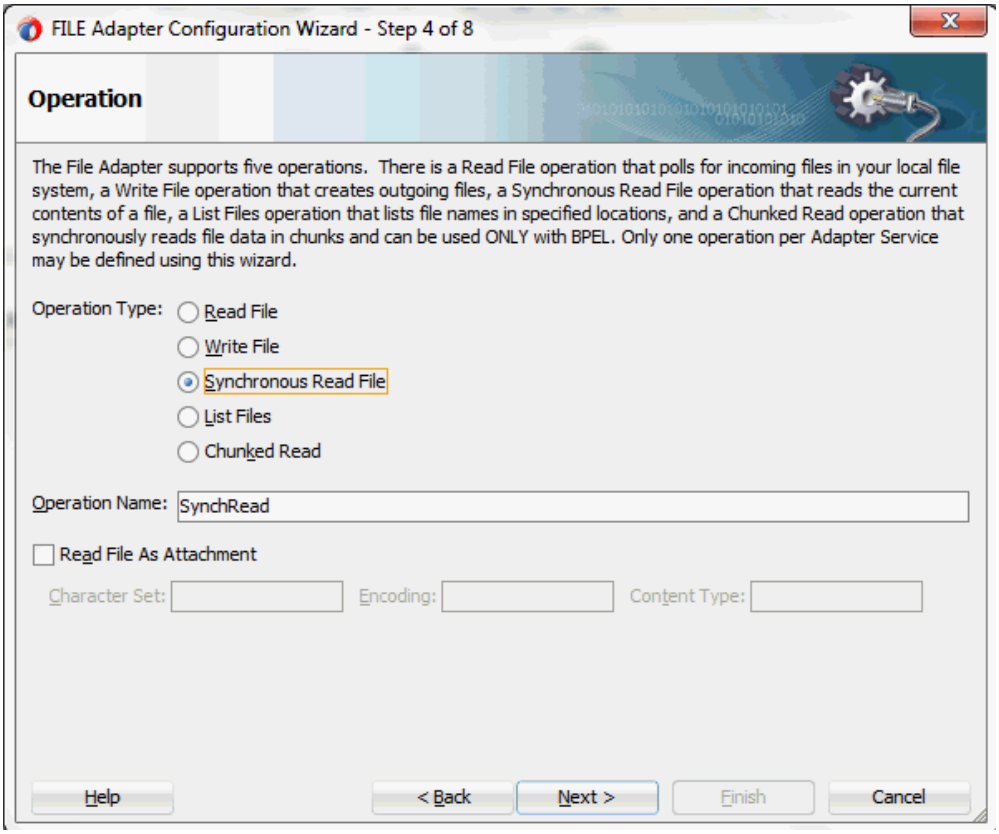
3. The File Server Connection page appears. Leave the default value unchanged for the File Server JNDI Name field.

   Click **Next**



4. In the Operation page, specify the operation type. For example, select the **Synchronous Read File** radio button. This automatically populates the **Operation Name** field.

Specifying the Operation

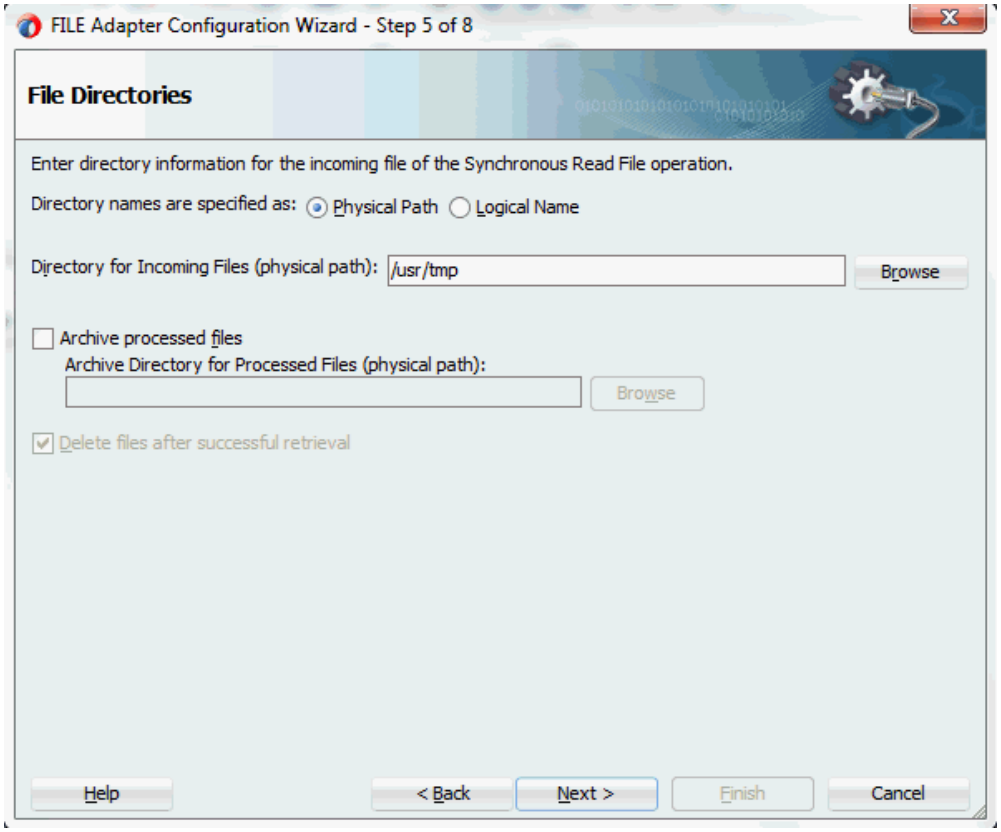


Click **Next** to access the File Directories page.

5. Select the **Physical Name** radio button and specify directory for incoming files, such as `/usr/tmp`.

   Ensure the **Delete files after successful retrieval** check box is not selected.

Configuring the Input File



Click **Next** to open the File Name page.

6. Enter the name of the file for the synchronous read file operation. For example, enter `order_data_xmlg.xml`.

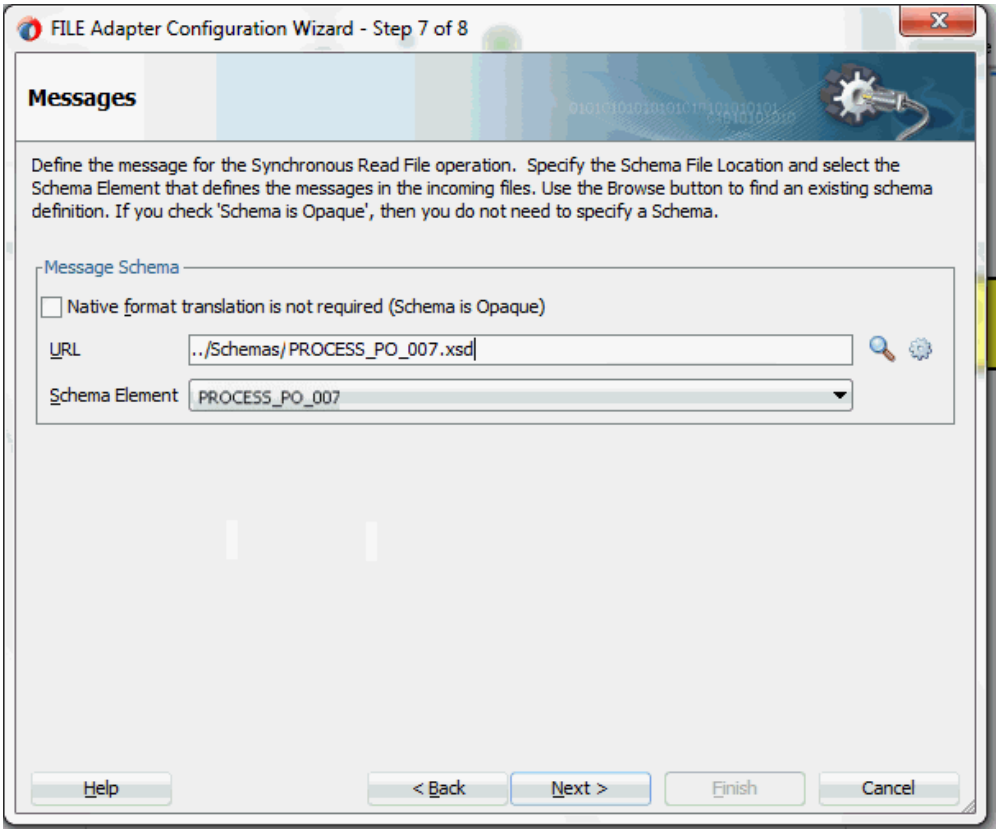
   Click **Next**. The Messages page appears.

7. Select the 'browse for schema file' icon to open the Type Chooser window.

   Expand the node by clicking **Project Schema Files > PROCESS_PO_007.xsd > PROCESS_PO_007**.

   The selected schema information will be automatically populated in the URL and Schema Element fields.

Specifying Message Schema



8. Click **Next** and then **Finish**. This completes the configuration and creates the partner link with the required WSDL settings for the File Adapter service. The wizard also generates the `getOrderXML.wsdl` WSDL file corresponding to the partner link.

   The `getOrderXML` Partner Link appears in the BPEL process diagram. Double click the Partner Link service icon to open the Edit Partner Link dialog and view the details.

Configuring the Invoke Activities

After adding and configuring the partner link, the next task is to configure the BPEL process. You can start by configuring the **Invoke** process activity to enqueue the XML Gateway inbound messages and set XML Gateway header properties.

Based on the scenario described earlier, you need to configure the following two Invoke activities:

1. To get the XML message details from the input XML file through synchronous read operation by invoking the `getOrderXML` partner link.

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

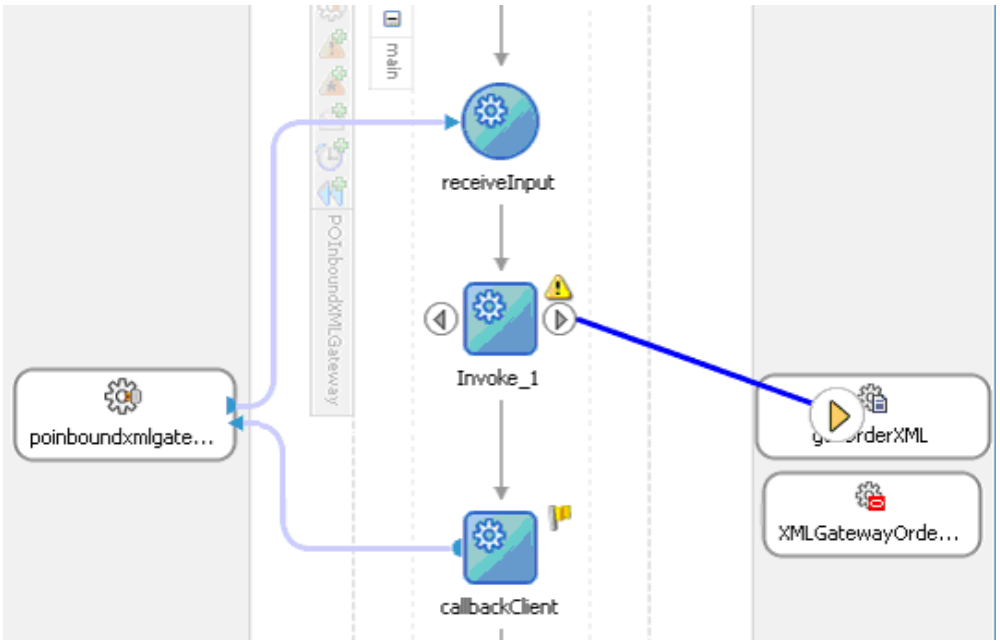
   The ECX Header properties for the XML Gateway inbound service can also be set through the Invoke activity.

To configure the first Invoke activity:

1. In Oracle JDeveloper BPEL Designer, expand the **BPEL Constructs** from the Component Palette. Drag and drop the first **Invoke** activity into the center swim lane of the process diagram, between the **receiveInput** and **callbackClient** activities.

2. Link the Invoke activity to the `getOrderXML` partner link service.

Creating an Invoke Activity



3. In the Edit Invoke dialog, enter a name for the Invoke activity in the General tab.

   The value of the Operation field is automatically selected based on the associated partner link. For example, this Invoke activity is to invoke a File Adapter service to synchronize read an input XML file; thus, the SynchRead is automatically populated in the Operation field.

4. Enter a name for the Invoke activity. In the Input tab, ensure the **Input Variable** radio button is selected. Click the **Create** icon next to the **Input Variable** field to create a new variable. The Create Variable dialog box appears.

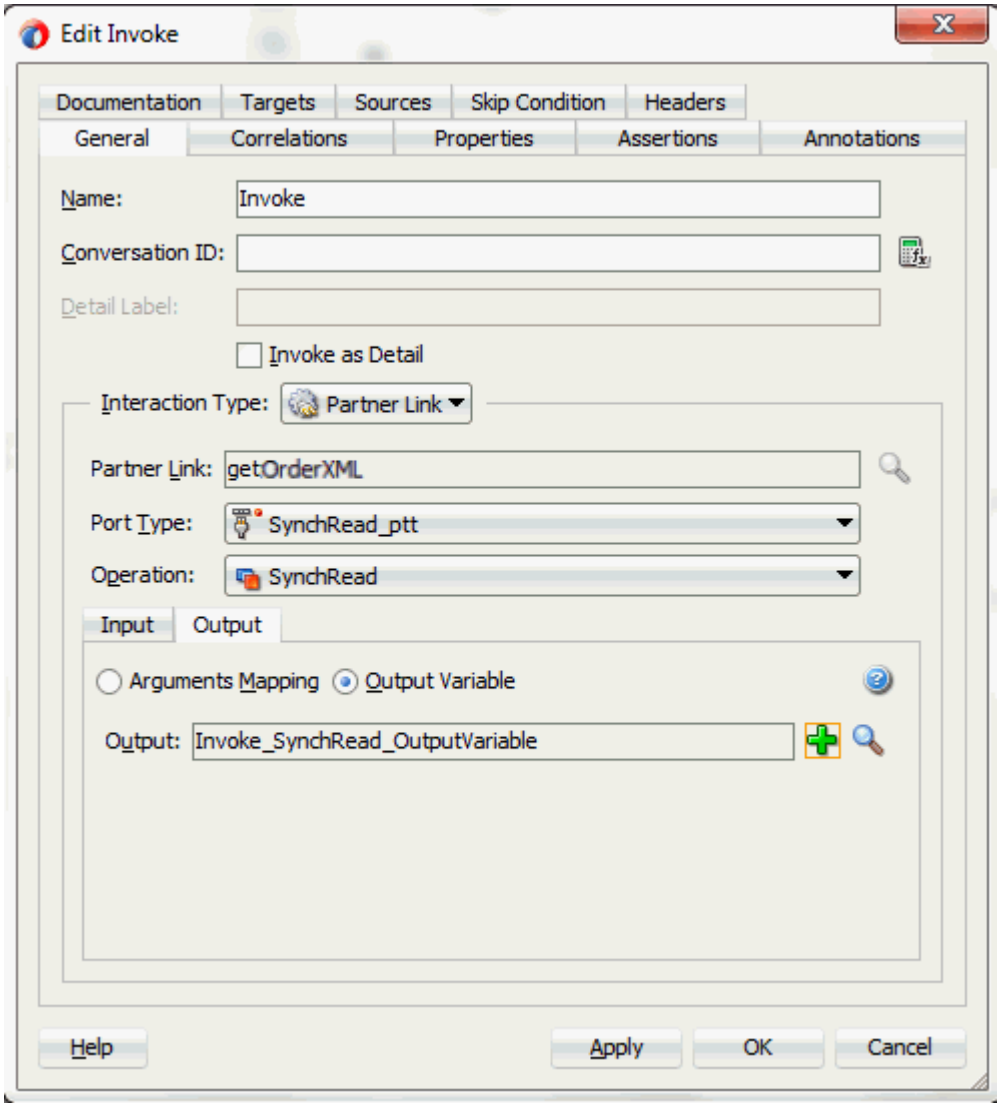
   Select **Global Variable** and enter a name for the variable. You can also accept the

default name. Click **OK**.

5. In the Output tab, ensure the **Output Variable** radio button is selected. Click the **Create** icon next to the **Output Variable** field to create a new variable. 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** to return to the Edit Invoke dialog.

The Edit Invoke Dialog



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

To configure the second Invoke activity:

1. In Oracle JDeveloper BPEL Designer, expand the **BPEL Constructs** from the Component Palette. Drag and drop the second **Invoke** activity into the center swim lane of the process diagram, between the first **Invoke** activity and **callbackClient** activity.

2. Link the Invoke activity to the `XMLGatewayOrderInbound` partner link service.

3. In the Edit Invoke dialog, enter a name for the Invoke activity in the General tab.

   The value of the Operation field is automatically selected based on the associated partner link. Since this Invoke activity is associated with an inbound message map partner link, the Enqueue operation is selected.

4. In the Input tab, ensure the **Input Variable** radio button is selected. Click the **Create** icon next to the **Input Variable** field to create a new variable. 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 **Apply**.

5. **Setting ECX Header Message Properties**

   You need to enter appropriate XML Gateway header property values in order to pass individual message properties required for XML transaction to complete successfully.

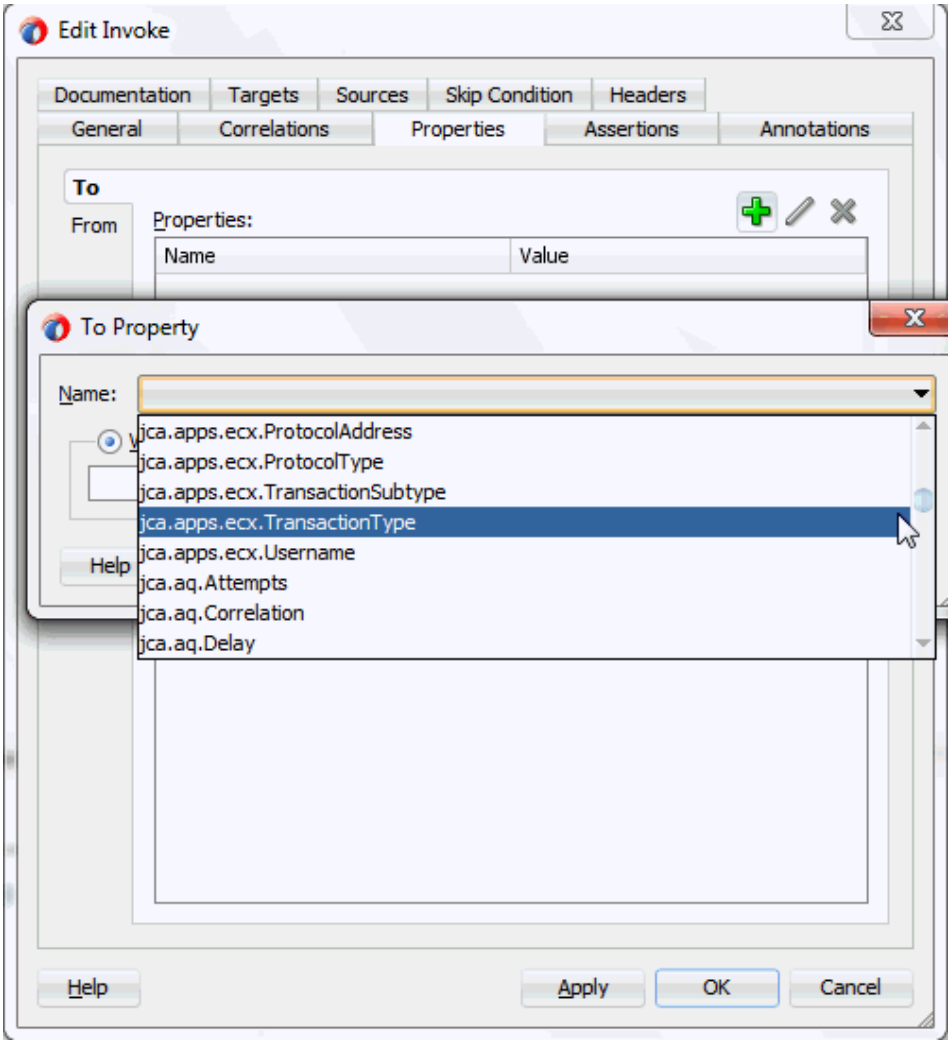
   Use the following steps to set ECX header message properties:

   1. Click the **Properties** tab in the Invoke dialog box. You will find all predefined normalized message properties.

   2. Click the **Create** icon. The To Property dialog appears.

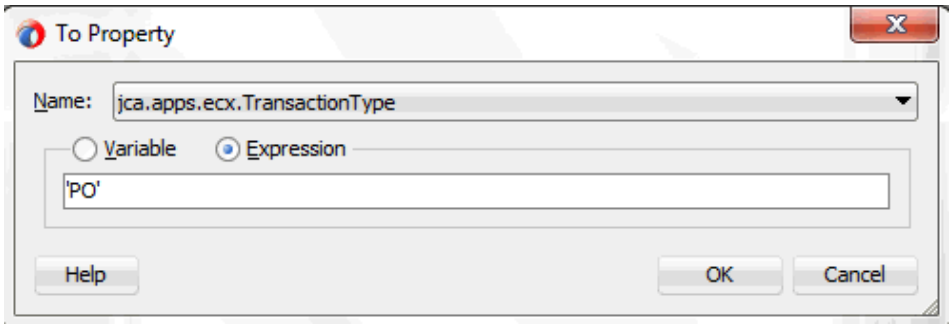
   3. Select the `jca.apps.ecx.TransactionType` property from the Name drop-down list.

Entering a Selected Property Value



4. Select the 'Expression' button as the message type.

   Enter a property value for the selected property. For example, enter `'PO'` for the property `jca.apps.ecx.TransactionType`.



Click **OK**.

5. Repeat Step 2 to Step 4 to set the following property values:

   - `jca.apps.ecx.MessageType: 'XML'`
   - `jca.apps.ecx.TransactionSubType: 'PROCESS'`
   - `jca.apps.ecx.PartySiteId: 'BWSANJOSE'`
   - `jca.apps.ecx.MessageType: 'XML'`
   - `jca.apps.ecx.MessageStandard: 'OAG'`
   - `jca.apps.ecx.DocumentNumber: 'order_xml_01'`

   Note: In earlier releases, XML Gateway header variables including `MESSAGE_TYPE`, `MESSAGE_STANDARD`, `TRANSACTION_TYPE`, `TRANSACTION_SUBTYPE`, `DOUCMENT_NUMBER`, and `PARTY_SITE_ID` are contained in the `SYSTEM.ECXMSG` object. Oracle E-Business Suite Adapter normalizes the `SYSTEM.ECXMSG` object to create and set each individual message property through the Properties tab of the Invoke activity. These header properties are required for the XML transaction to complete successfully.

6. In the Invoke dialog box, click **Apply** and then **OK**.

   Note: If you have configured a partner link for outbound messages, then you need to configure a **Receive** activity in place of the **Invoke** activity. The **Receive** activity is used to dequeue the XML Gateway outbound messages.

Configuring the Assign Activity

Use the Assign activity to pass the output of `getOrderXML` service as an input to the `XMLGatewayOrderInbound` service.

To add an Assign activity:

1. In Oracle JDeveloper BPEL Designer, expand the **BPEL Constructs** from the Component Palette. Drag and drop the **Assign** activity from the **Basic Activities** section into the center swim lane of the process diagram between the two **Invoke** activities you just created earlier.

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

   Click the General tab to enter a name for the Assign activity. For example, `SetOrderXML`.

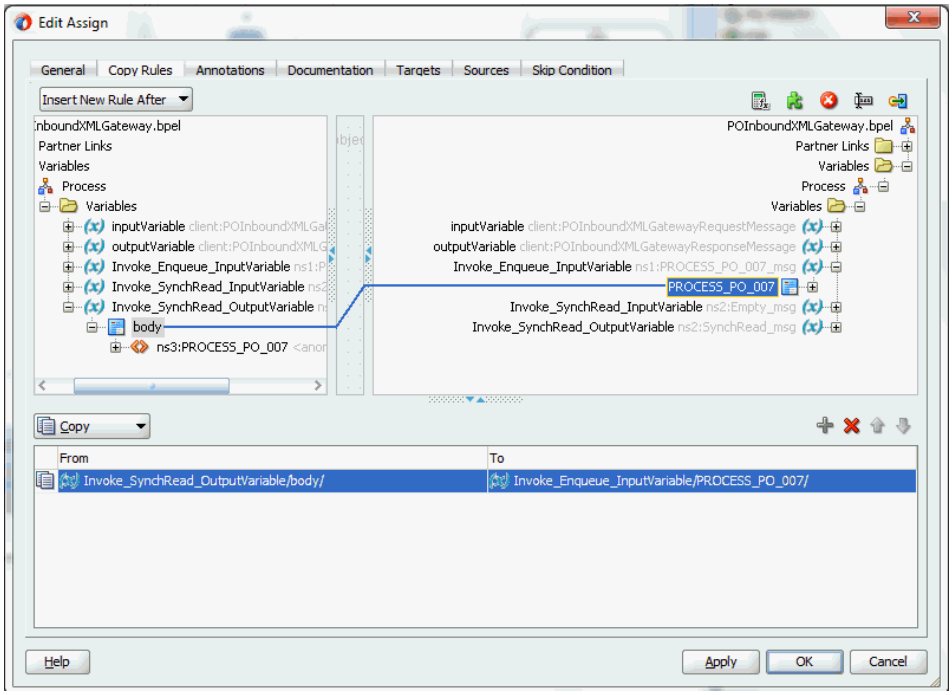
3. Select the Copy Rules tab and expand the target trees:

   - In the From navigation tree, navigate to **Variable > Process > Variables > Invoke_SynchRead_OutputVariable** and select **body**.

   - In the To navigation tree, navigate to **Variable > Process > Variables > Invoke_Enqueue_InputVariable** and select **PROCESS_PO_007**.

     Drag the source node (body) to connect to the target node (PROCESS_PO_007) that you just specified. This creates a line that connects the source and target nodes. The copy rule is displayed in the From and To sections at the bottom of the Edit Assign dialog box.

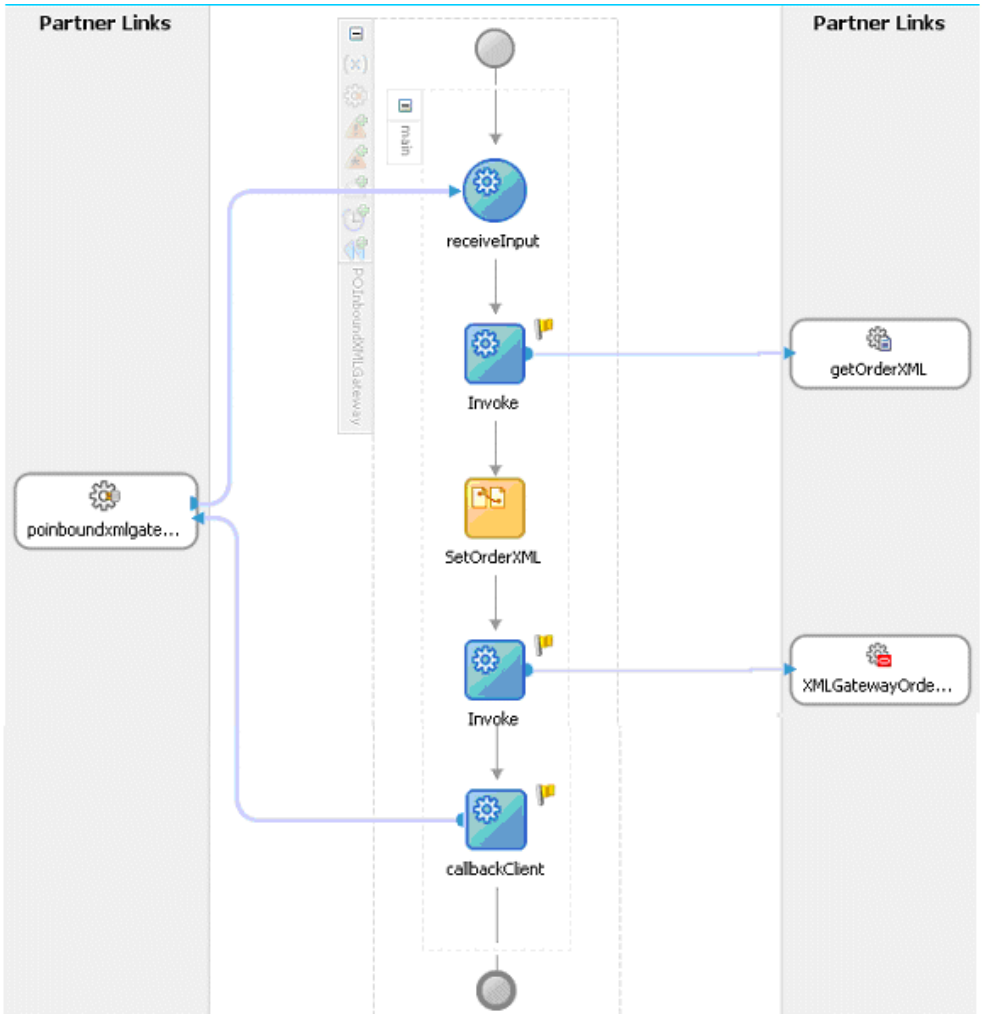
Assigning Parameters



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

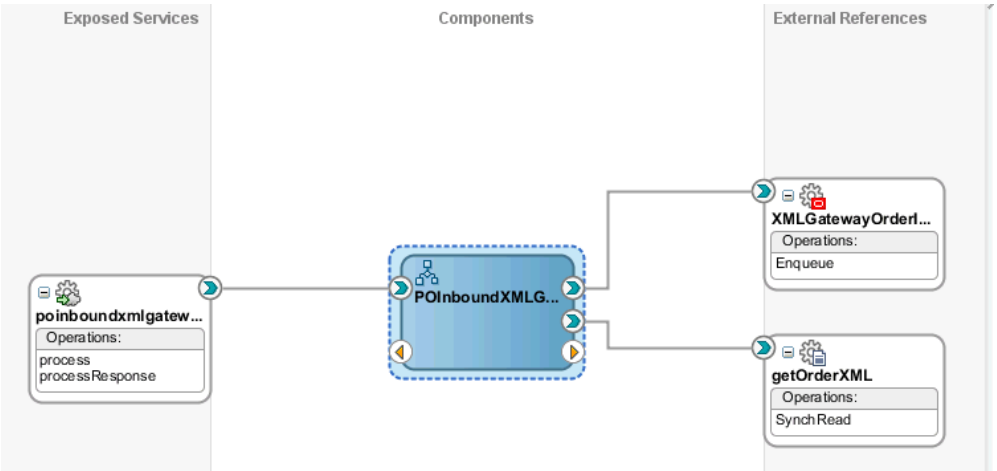
   The following diagram illustrates the complete BPEL process:

BPEL Process Diagram



Click the `POInboundXMLGateway` (composite.xml) to display the Oracle JDeveloper composite diagram:

Oracle JDeveloper composite Diagram



Run-Time Tasks for XML Gateway Inbound Messaging

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

1. Deploy the SOA Composite application with BPEL process, page 5-34
2. Test the SOA Composite application with BPEL process, page 5-37
3. Verify records in Oracle E-Business Suite, page 5-39

Deploying the SOA Composite Application with BPEL Process

To invoke the XML Gateway inbound service from the BPEL client contained in the SOA composite, the SOA composite needs to be deployed to the Oracle WebLogic managed server. This can be achieved using Oracle JDeveloper. Once the composite is deployed, it can be tested from the Oracle Enterprise Manager Fusion Middleware Control Console.

Prerequisites

Before deploying the SOA composite with BPEL process using Oracle JDeveloper, you must have established the connectivity between the design-time environment and the run-time server. For more information, see Configuring the Data Source in Oracle WebLogic Server, page 11-1 and Creating an Application Server Connection, page 11-10.

Note: If a local instance of the WebLogic Server is used, start the

WebLogic Server by selecting **Run > Start Server Instance** from Oracle JDeveloper. Once the WebLogic Admin Server "DefaultServer" instance is successfully started, the <Server started in Running mode> and DefaultServer started message in the Running:DefaultServer and Messages logs should appear.

To deploy the SOA Composite application with BPEL process:

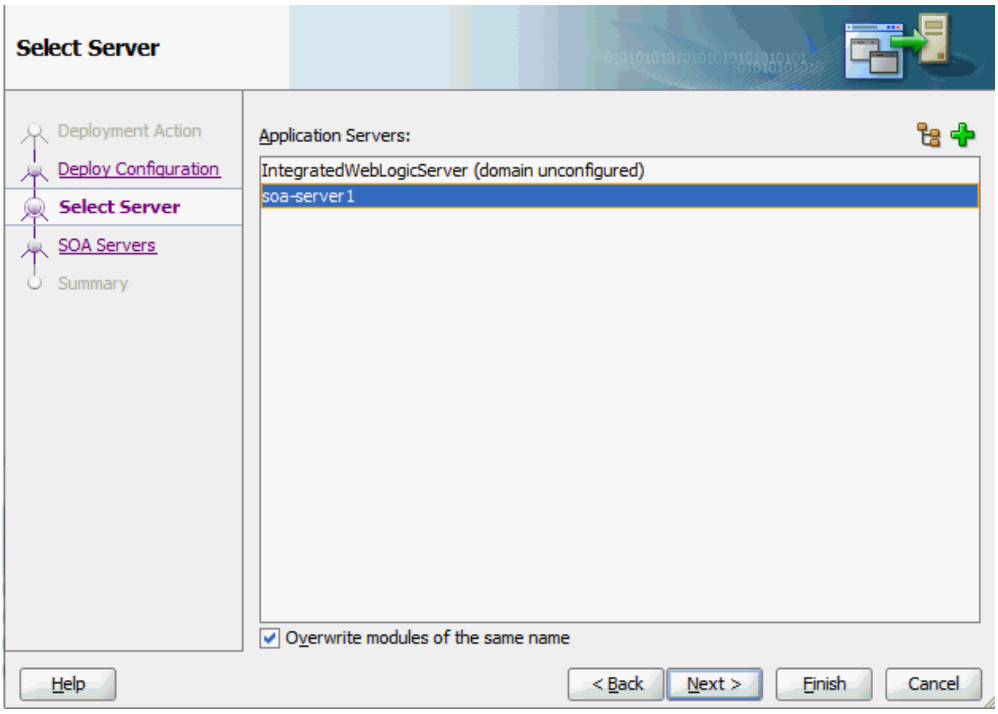
1. Select the SOA Composite project in the Applications Navigator.

2. Right-click the project name. Select **Deploy > [project name] > [serverConnection]** from the menu that appears.

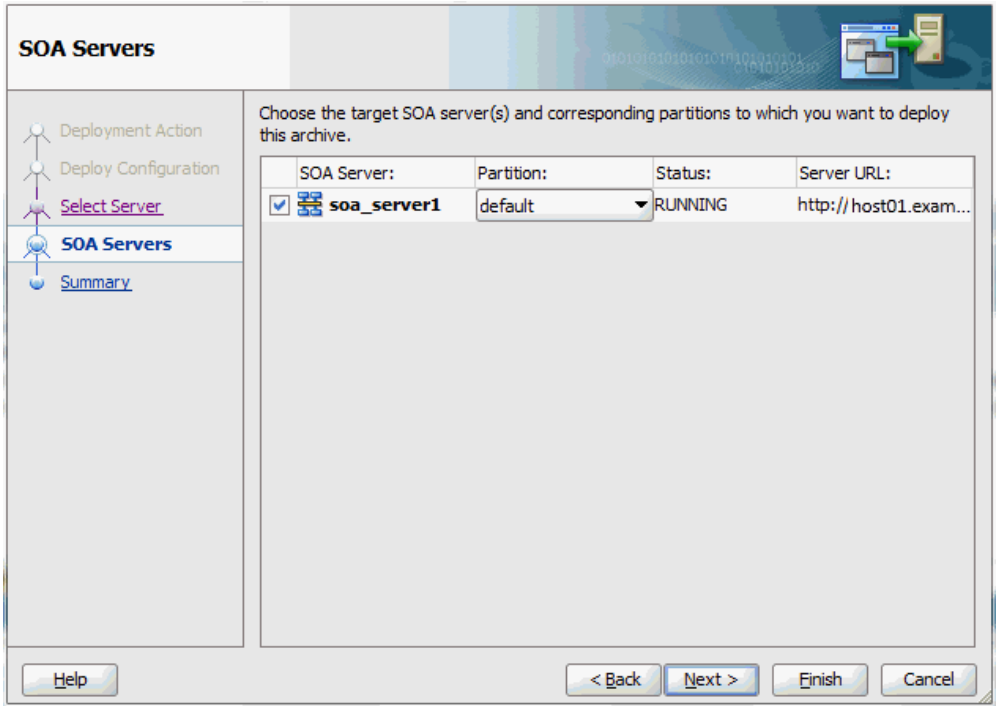
   For example, you can select **Deploy > POInboundXMLGateway > soa-server1** to deploy the process if you have the connection set up appropriately.

   > **Note:** If this is the first time to set up server connection, then the Deployment Action window appears. Select 'Deploy to Application Server' and click **Next**.
   >
   > In the Deploy Configuration window, ensure the following information is selected before clicking **Next** to add a new application server:
   >
   > - New Revision ID: 1.0
   > - Mark composite revision as default: Select this check box.
   > - Overwrite any existing composites with the same revision ID: Select this check box.
   >
   > The steps to create a new Oracle WebLogic Server connection from JDeveloper are covered in Creating an Application Server Connection, page 11-10.

3. In the Select Server page, select 'soa-server1' that you have established the server connection earlier. Click **Next**.



4. In the SOA Servers page, accept the default target SOA Server ('soa-server1') selection.



Click **Next** and **Finish**.

If you are deploying the composite for the first time from your Oracle JDeveloper session, the Authorization Request window appears. Enter username and password information specified during Oracle SOA Suite installation. Click **OK**.

5. Deployment processing starts. Monitor deployment progress and check for successful compilation in the SOA - Log window.

   Verify that the deployment is successful in the Deployment - Log window.

Testing the SOA Composite with BPEL Process

Once the BPEL process contained in the SOA Composite application is deployed, you can manage and monitor the process from the Oracle Enterprise Manager Fusion Middleware Control Console. You can also test the process and the integration interface by manually initiating the process.

To test the deployed SOA Composite with BPEL process:

1. Navigate to Oracle Enterprise Manager Fusion Middleware Control Console (`http://<servername>:<portnumber>/em`). The composite you deployed is displayed in the Applications Navigation tree.

2. Enter username (such as `weblogic`) and password and click **Login** to log in to a farm.

   You may need to select an appropriate target instance farm if there are multiple target Oracle Enterprise Manager Fusion Middleware Control Console farms.

3. From the Farm base domain, expand the **SOA > soa-infra > soa-infra (soa_server1) > default** to navigate through the SOA Infrastructure home page and menu to access your deployed SOA composite applications running in the SOA Infrastructure for that managed server.

   > **Note:** The Farm menu always displays at the top of the navigator. As you expand the SOA folder in the navigator and click the links displayed beneath it, the SOA Infrastructure menu becomes available at the top of the page.

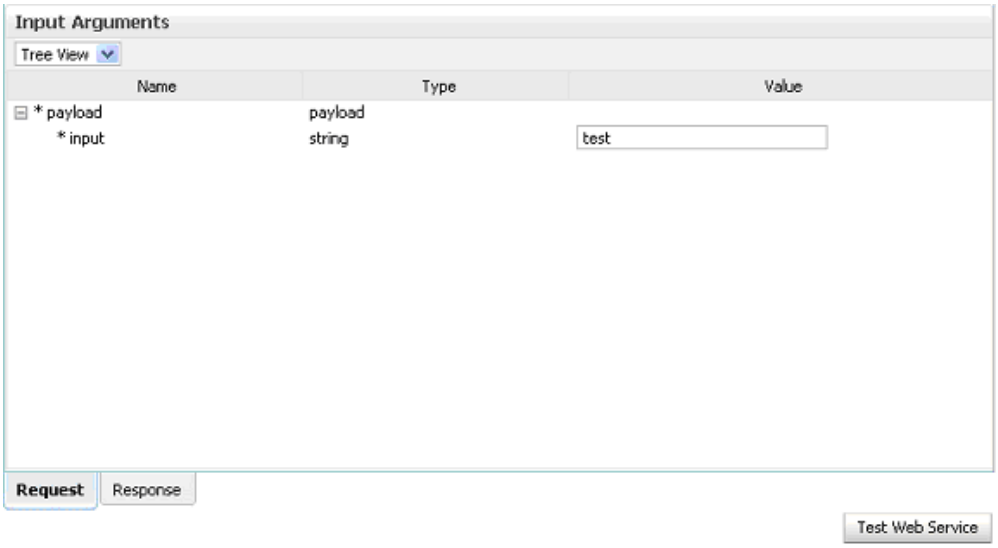
   Click the SOA Composite application that you want to initiate (such as 'POInboundXMLGateway') from the SOA Infrastructure.

   Click **Test** at the top of the page.

4. The Test Web Service page for initiating an instance appears. You can specify the XML payload data to use in the Input Arguments section.

   Enter the input string required by the process and click **Test Web Service** to initiate the process.

Testing Web Service



The test results appear in the Response tab upon completion. Click **Launch Flow Trace** in the Test Web Service page. The Flow Trace page is displayed.

5. In the Trace section, you should find the sequence of the message flow for the service binding component (`poinboundxmlgateway_client_ep`), BPEL component (`POInboundXMLGateway`), and reference binding components (`getOrderXML` and `XMLGatewayOrderInbound`). All involved components have successfully received and processed messages.

   If any error occurred during the test, you should find it in the Faults section.

6. Click your BPEL service component instance link (such as `POInboundXMLGateway`) to display the Instances page where you can view execution details for the BPEL activities in the Audit Trail tab.

   Click the Flow tab to check the BPEL process flow diagram. Click an activity of the process diagram to view the activity details and flow of the payload through the process.

Verifying Records in Oracle E-Business Suite

Once the BPEL process contained in the SOA Composite application is successfully initiated and completed, you can validate it through the relevant module in Oracle E-Business Suite applications.

For example, you can validate it using Oracle Transaction Monitor to ensure the process is successfully completed, and then import the order to Order Management application. You can then verify it in Oracle Order Management to ensure the order does exist.

To validate it Using Oracle Transaction Monitor

Additionally, you can also 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 E-Business Suite 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.

You can navigate to the Transaction Monitor page using the Workflow Administrator Web responsibility. The Transaction Monitor provides the following:

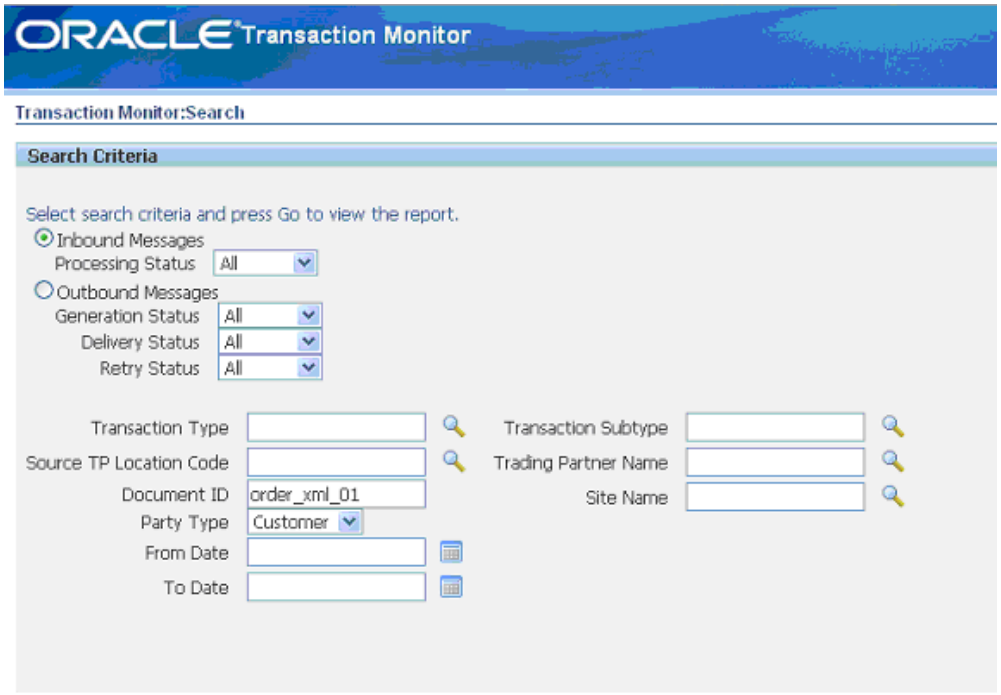
- Flexible search criteria to support access to a specific document or group of documents
- Search results at the document header level with drill down by document ID
- Resend capability for outbound messages
- Viewing capability of the XML message content

> **Note:** For details on using the Transaction Monitor, see *Oracle XML Gateway User's Guide*. This guide is part of the Oracle E-Business Suite Documentation Library which can be accessed on the Oracle Technology Network (OTN).

Use the following steps to validate in Oracle Transaction Monitor:

1. Log on to Oracle E-Business Suite with the Workflow Administrator Web Applications responsibility. Select Transaction Monitor to open the search window to search for the order.
2. Select the **Inbound Messages** radio button.

Searching from the Transaction Monitor



Enter the following information in the search window:

- Party Type: Customer
- Document ID: Enter the document ID you defined for the header message property. For example, enter `oder_xml_01`.

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

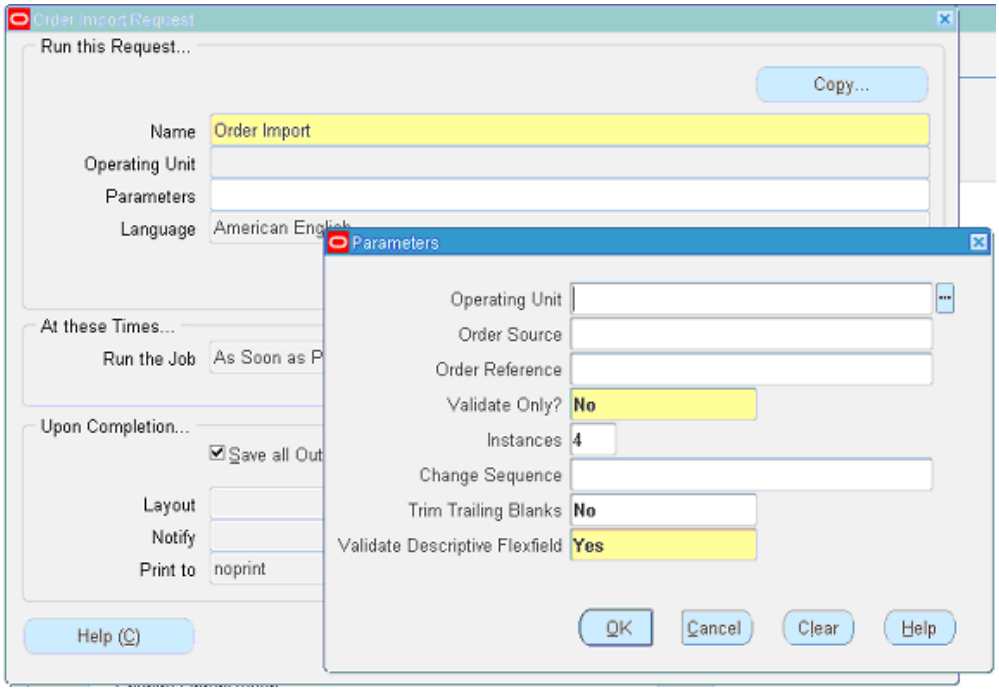
   Confirm that transaction for 'order_xml_01' has the 'Success' status.

To import the order to Oracle Order Management:

1. Log on to the Forms-based Oracle E-Business Suite with the Order Management Super User, Vision Operations (USA) responsibility.

2. Select Orders, Returns : Import Orders > Order Import Request. The Order Import Request window is displayed along with the Parameters dialog.

Order Import Request



3. Click **OK** in the parameters dialog. The Order Import request name is populated automatically in the Import Request window.

4. Click **Submit** to submit the request. This displays a concurrent request number. Record the request number, but click **No** in the Decision dialog that you will not submit another request.

5. From the application menu, select View >Requests to open the Find Requests window.

6. Enter the request number you recorded earlier and click **Find**.

   This would show the status of Order Import request. It should be success and the order should be created in Order Management application.

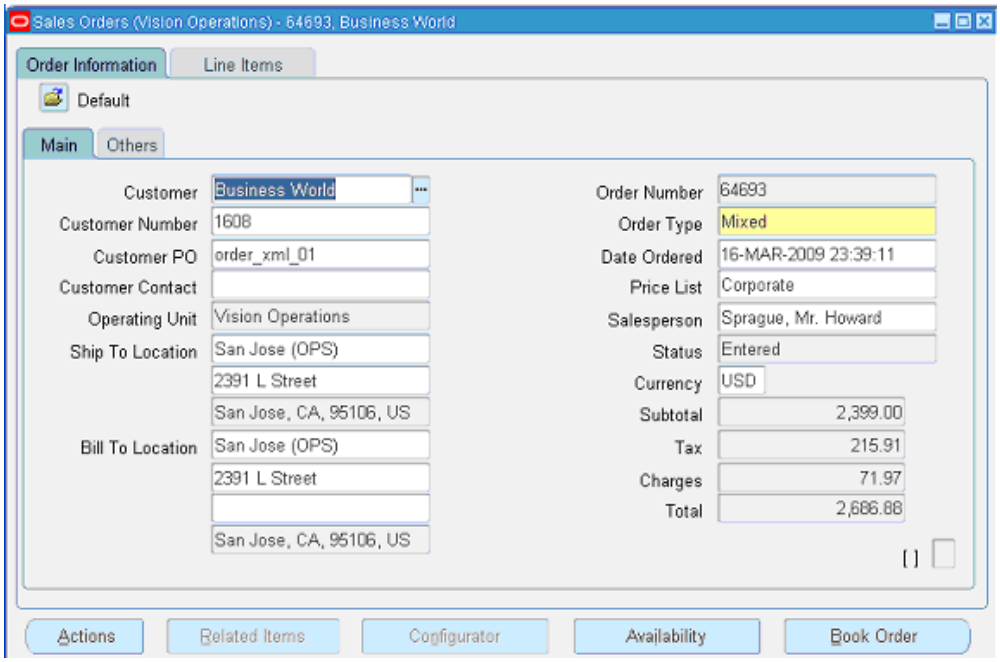
To validate it in Oracle Order Management:

1. Log on to the Forms-based Oracle E-Business Suite with the Order Management, Super User responsibility.

2. Select Order Returns > Sales Order. Sales Order Forms would open up.

3. Search for an order by entering the order number (`order_xml_01`) in the Customer PO field. This would bring up the details of a newly created order using XML Gateway inbound map.

Sales Orders



You can also select the Items tab for item details.

Design-Time Task for XML Gateway Outbound Messaging

For an outbound XML Gateway Map interface, since an outbound message is first enqueued to the ECX_OUTBOUND queue, Oracle BPEL PM listens to ECX_OUTBOUND queue for the message with the same correlation Id. The message will then be dequeued to retrieve outbound data from Oracle E-Business Suite. The retrieved data can be passed to trading partners or consumers who subscribed to the message.

SOA Composite Application with BPEL Process Scenario

Take XML Gateway outbound interface PO acknowledgement XML Transaction as an example. The XML Gateway outbound interface is exposed as a Web service through `ECX_CBODO_OAG72_OUT_CONFIRM` outbound map.

When a purchase order is created and approved, on approval of the purchase order, a workflow will be triggered which 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 ECX_OUTBOUND queue for delivery to the other system.

The correlation Id for this message is set to "BPEL" and the Oracle BPEL PM listens to ECX_OUTBOUND queue for the message with the correlation Id "BPEL". Confirm BOD as the PO Acknowledgement is written as an output XML file using File Adapter.

Prerequisites to Configure XML Gateway Outbound

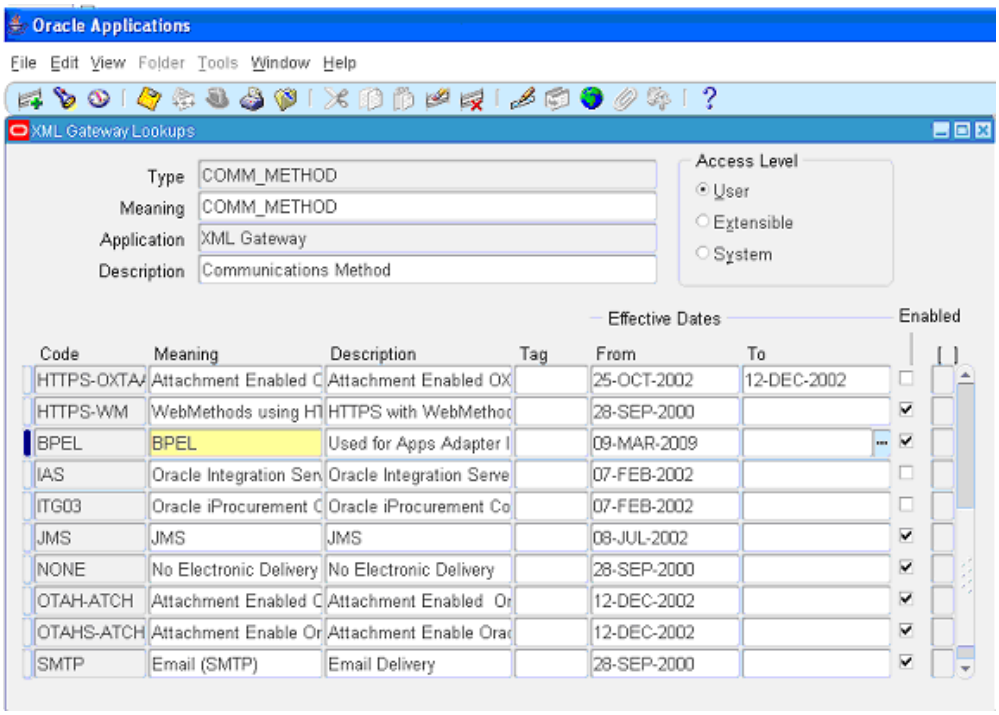
Setting Up Correlation Identifier

For invoking an outbound message map, you need to set up the correlation identifier in Oracle E-Business Suite. 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 E-Business Suite 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 to open the XML Gateway Lookups form.

XML Gateway Lookups



4. Search for `COMM_METHOD` in the **Type** field to see if it exists in the system.
5. Add a new record to the `COMM_METHOD` type by entering `BPEL` for the **Code** field and **Meaning** field. Enter description information and save the record.

   Oracle XML Gateway puts the correlation of BPEL when enqueueing the message

on the ECX_OUTBOUND queue.

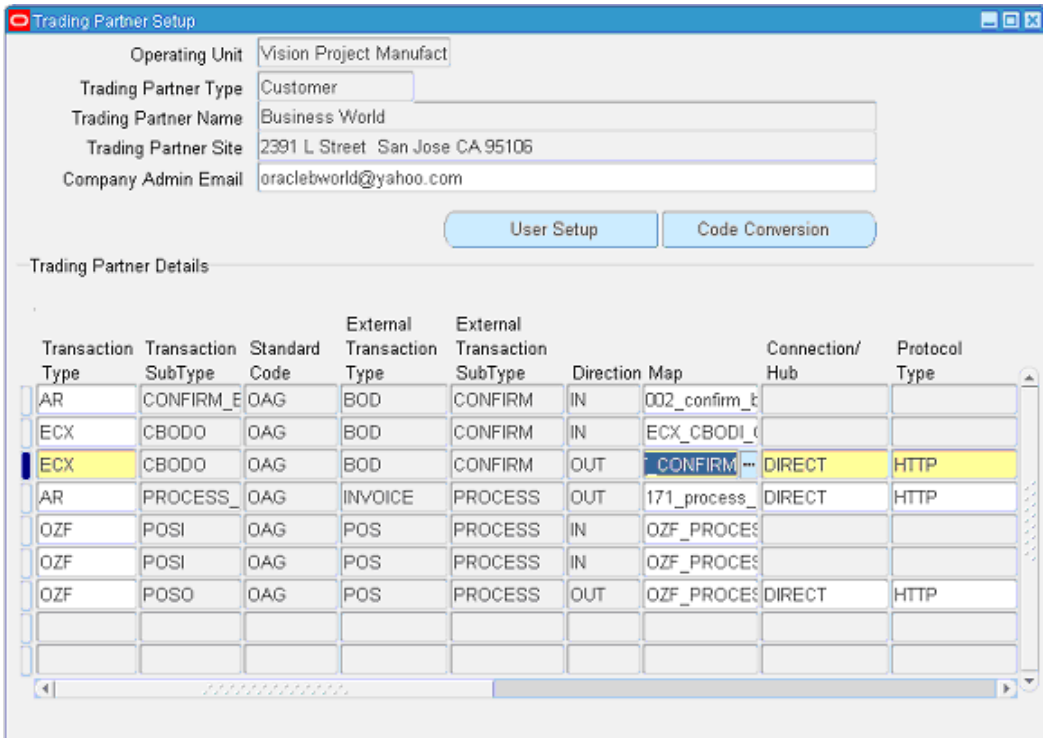
Setting Up XML Gateway Trading Partner

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



SOA Composite Application with BPEL Process Creation Flow

Based on the PO acknowledgement XML Transaction scenario, the following design-time tasks are discussed in this chapter:

1. Create a new SOA Composite application with BPEL process, page 5-46
2. Create a Partner Link, page 5-52
3. Add a Receive activity, page 5-57
4. Add a Partner Link for File Adapter, page 5-59
5. Add an Invoke activity, page 5-64
6. Add an Assign activity, page 5-65

Creating a SOA Composite Application with BPEL Process

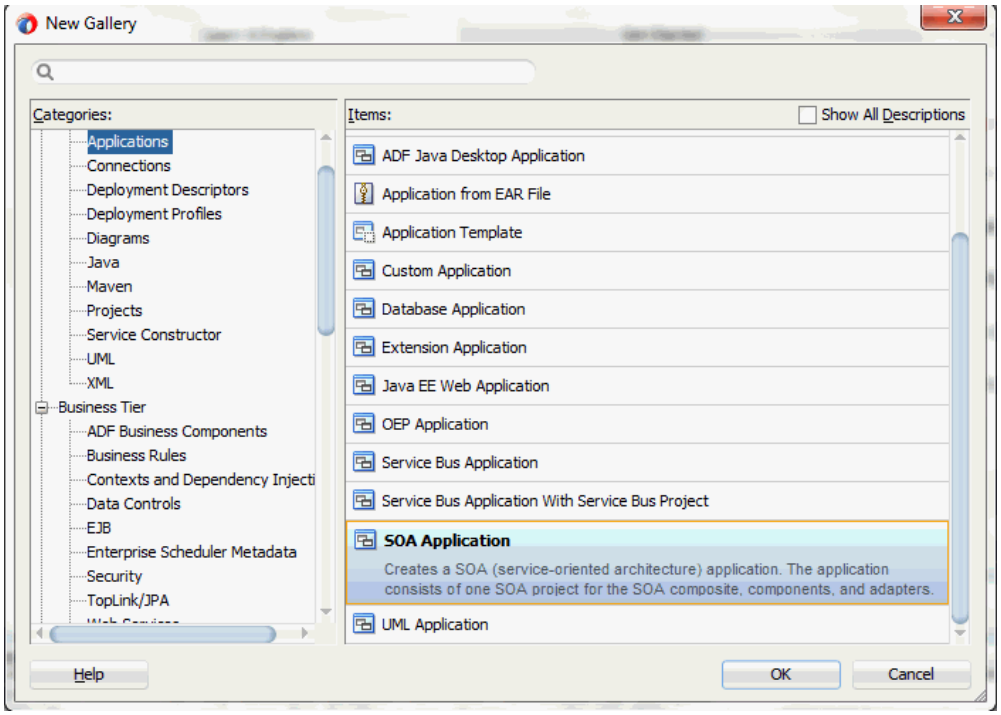
To create a SOA Composite application with BPEL process:

1. Launch Oracle JDeveloper.

2. Click **New Application** in the Application Navigator.

   The New Gallery page appears. Select **SOA Application** from the Items list.

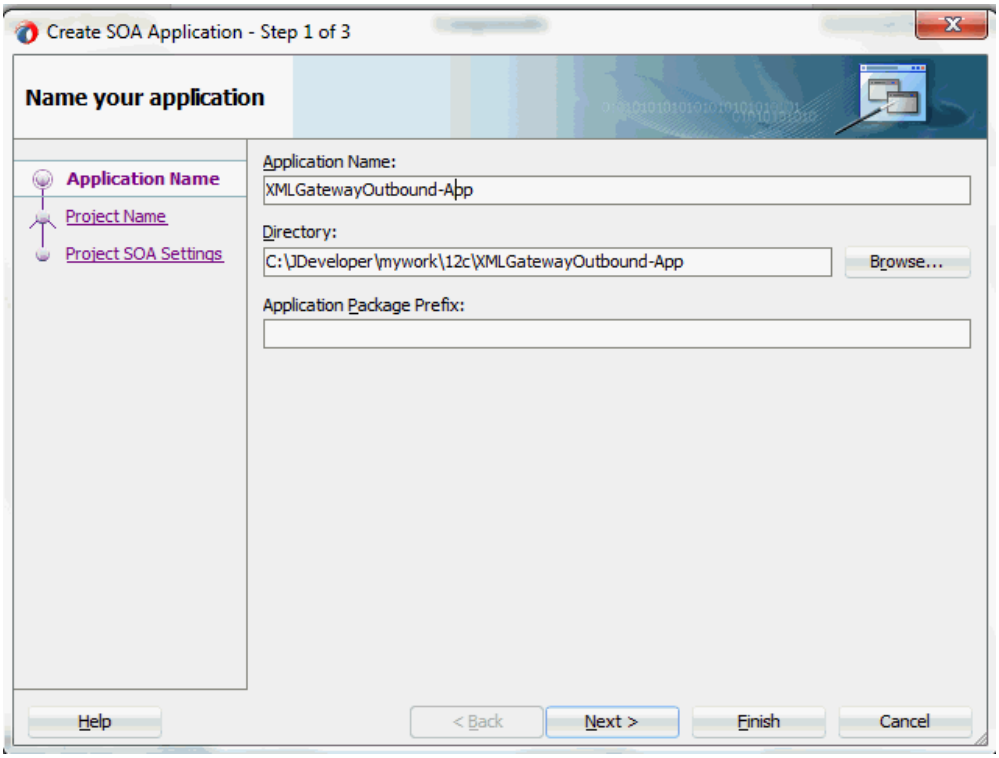
The New Gallery Page



   Click **OK**.

3. The Create SOA Application - Name your application page is displayed.

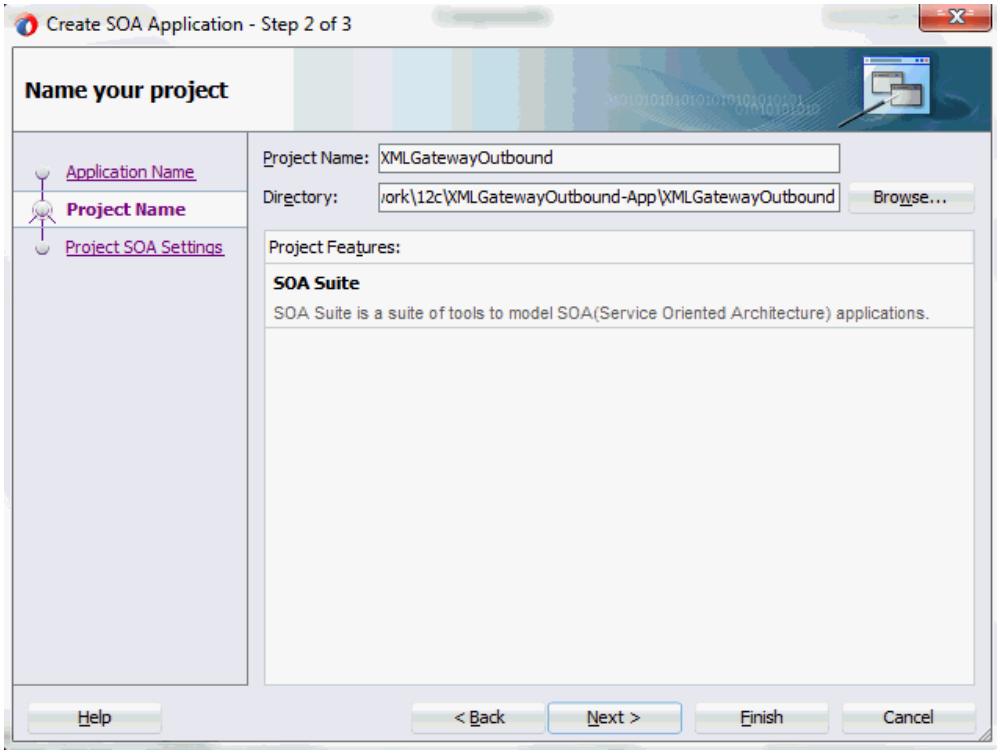
The Create SOA Application - Name your application Page



4. Enter an appropriate name for the application in the **Application Name** field.

   Click **Next**. The Create SOA Application - Name your project page is displayed.

The Create SOA Application - Name your project Page

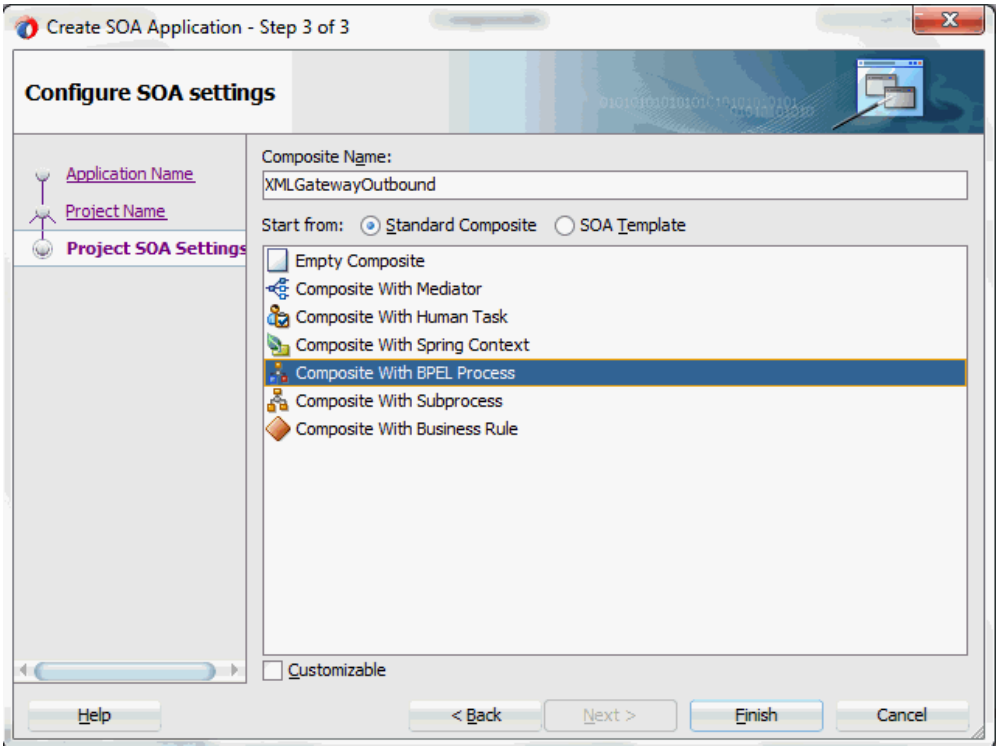


5. Enter an appropriate name for the project in the **Project Name** field. For example, `XMLGatewayOutbound`.

   Notice that **SOA Suite** is shown in the Project Features section.

6. Click **Next**. The Create SOA Application - Configure SOA settings page is displayed.

The Create SOA Application - Configure SOA settings Page



7. Leave the default **Standard Composite** selection unchanged for the Start from field.

   Select **Composite With BPEL Process** from the Standard Composite list, and then click **Finish**. You have created a new application, and a SOA project. This automatically creates a SOA Composite.

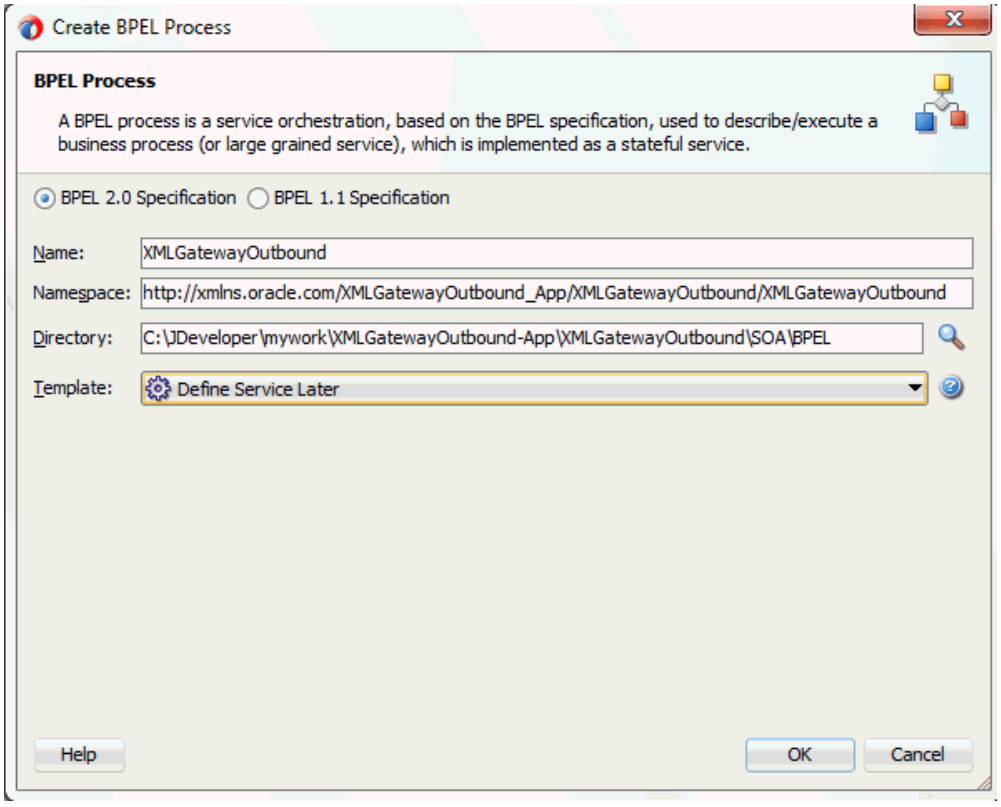
   The Create BPEL Process page is displayed.

8. Leave the default **BPEL 2.0 Specification** selection unchanged. This creates a BPEL project that supports the BPEL 2.0 specification.

   Enter an appropriate name for the BPEL process in the **Name** field. For example, `XMLGatewayOutbound`.

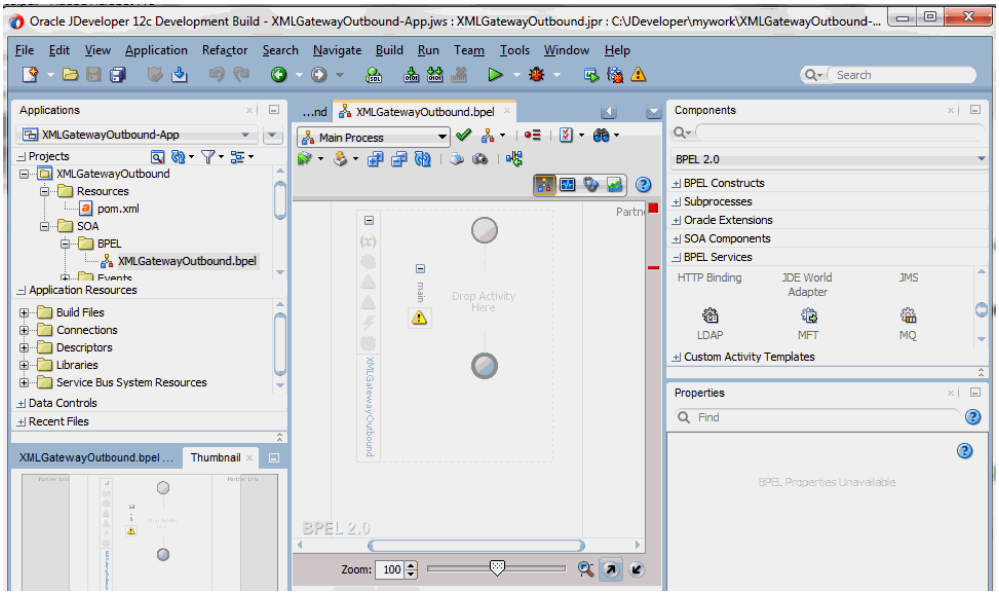
   Select **Define Service Later** in the **Template** field. Click **OK**.

Create BPEL Process



An empty BPEL process is created. The required source files including bpel and wsdl, using the name you specified (for example, `XMLGatewayOutbound.bpel` and `XMLGatewayOutbound.wsdl`) and `XMLGatewayOutbound` (composite.xml) are also generated.

An Empty BPEL Process



Adding a Partner Link

This section describes how to create an Oracle E-Business Suite Adapter for the application service by adding 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.

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 **BPEL Services** in the Component palette.

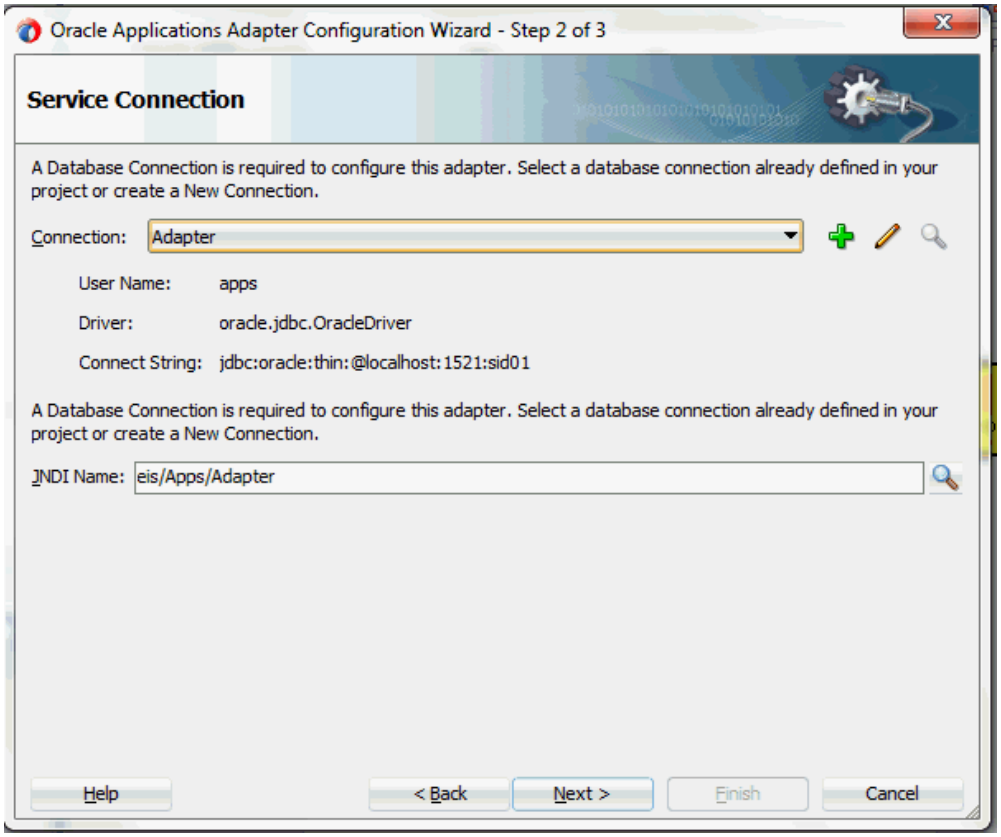
   Drag and drop **Oracle E-Business Suite** (formerly known as **Oracle Applications**) from the **BPEL Services** list into the right Partner Link swim lane of the process diagram. The Oracle E-Business Suite Reference page appears.

   Enter a reference name in the **Name** field. For example, `XMLGatewayPOAckOutbound`.

   Click **Next**.

2. The Service Connection dialog appears.

Specifying a Database Service Connection



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 **Create a New Database Connection** icon.

     How to define a new database connection, see Create a New Database Connection, page 5-14.

   - 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 Oracle E-Business Suite Adapter 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 Fusion Middleware User's Guide for Technology Adapters.*

4. Once you have completed creating a new connection for the service, you can add an outbound message map by browsing through the list of APIs available in Oracle E-Business Suite.

   Click **Next**.

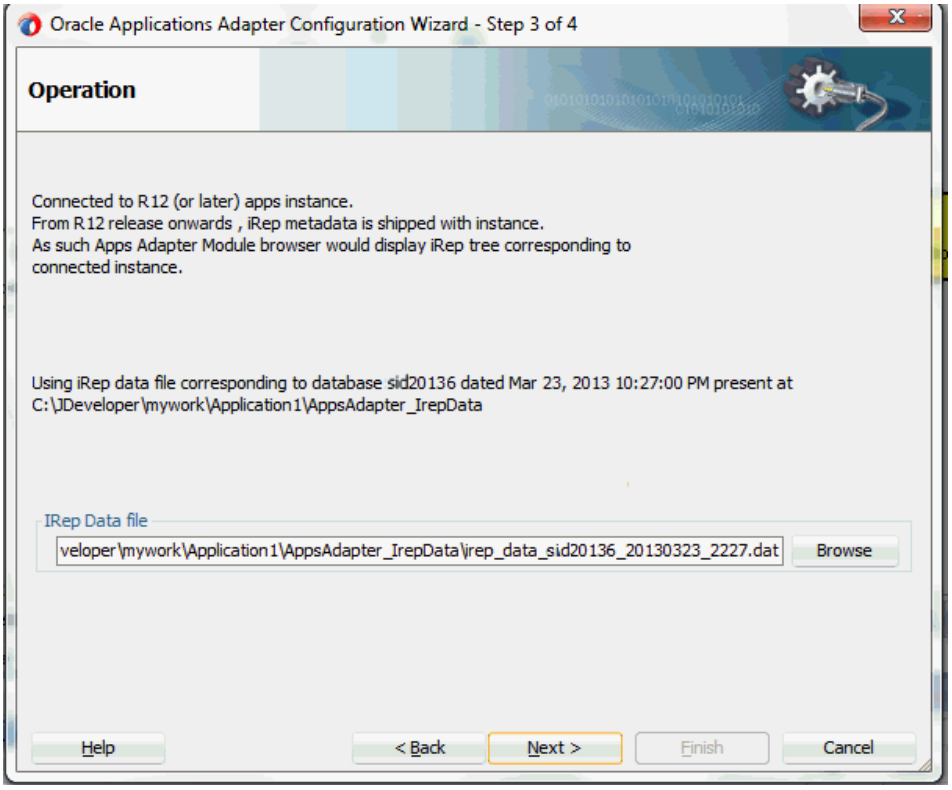
   If you are connecting to Oracle E-Business Suite Release 12, then the **IREP File not present** dialog appears indicating that Adapter could not find the Oracle Integration Repository data file corresponding to the database you are connecting in your workspace. Absence of the data file would make browsing 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 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.

     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

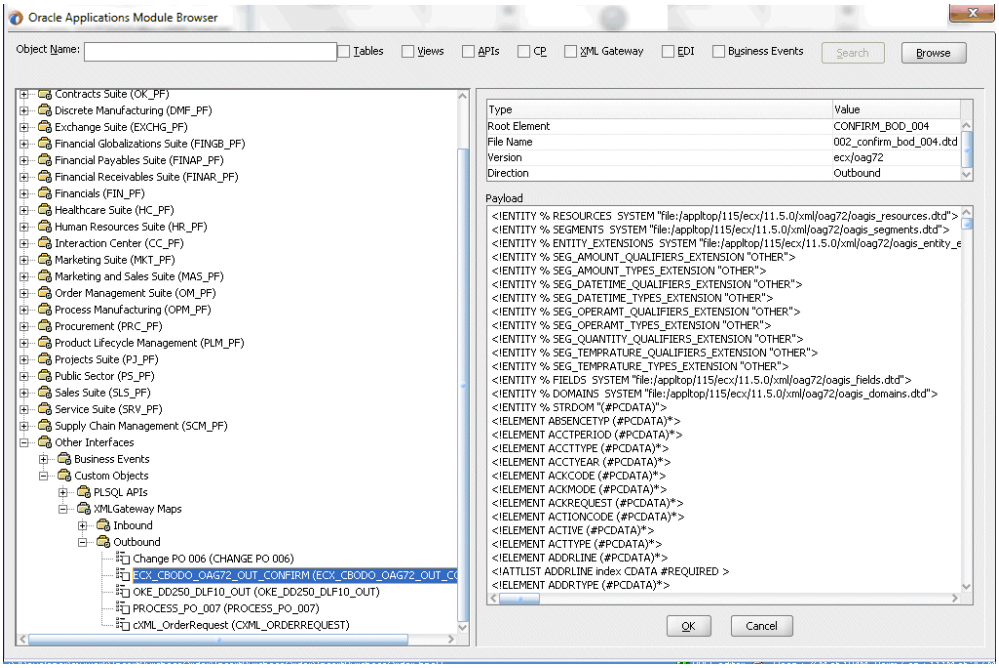


- Click **No** to query the Integration Repository data file from the live database you are connecting to display the Integration Repository tree.

In this example, click **Next** in the Operation page to open the Oracle E-Business Suite Module Browser.

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

Specify an API from the Module Browser

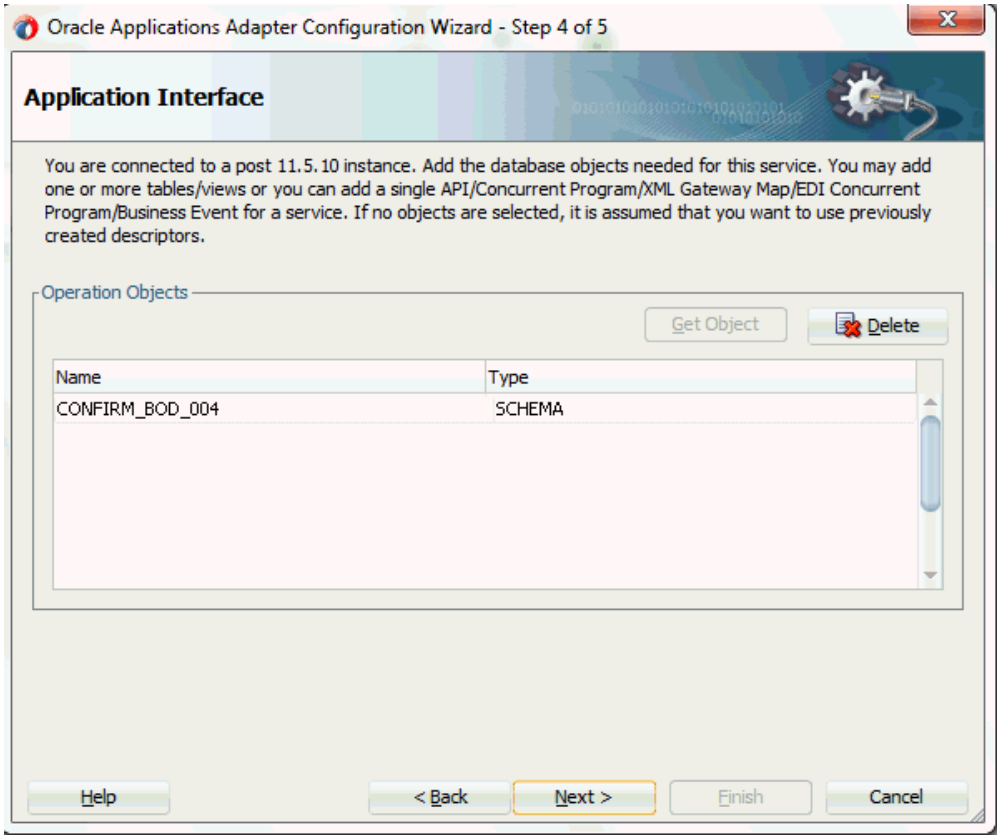


Note: The Oracle E-Business Suite Module Browser includes the various product families that are available in Oracle E-Business Suite. Each product family contains the individual products. Each product contains the business entities associated with the product. Business entities contain the various application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

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

6. Click **OK**.

Adapter Configuration Wizard - Application Interface Page



Click **Next**.

7. Click **Finish** to complete the process. The partner link is created with the required WSDL settings. The wizard also generates the WSDL file corresponding to the XML schema. This WSDL file is now available for the partner link.

Adding a Receive Activity

When configuring the Oracle E-Business Suite Adapter 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 Oracle JDeveloper BPEL Designer, expand the **BPEL Constructs** from the Component Palette. Drag and drop the **Receive** activity from the **Web Service** section into the center swim lane of the process diagram.

2. Link the **Receive** activity to the partner link `XMLGatewayPOAckOutbound` you

created earlier.

The Receive dialog 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.

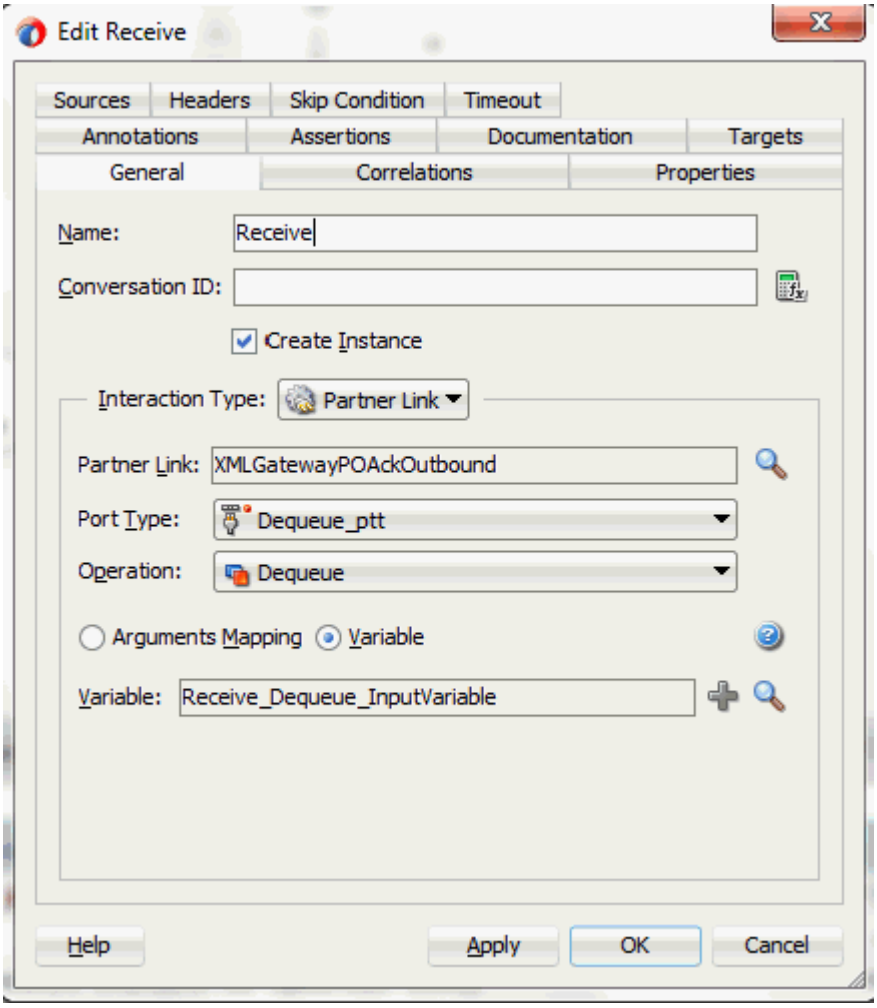
4. Specify a **Variable** to receive the message data from the partner link by clicking the **Create** icon to the right of the Variable field.

   The **Create Variable** dialog box appears.

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

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

Configuring the Receive Activity



6. Click **Apply** in the Receive dialog, then click **OK**.

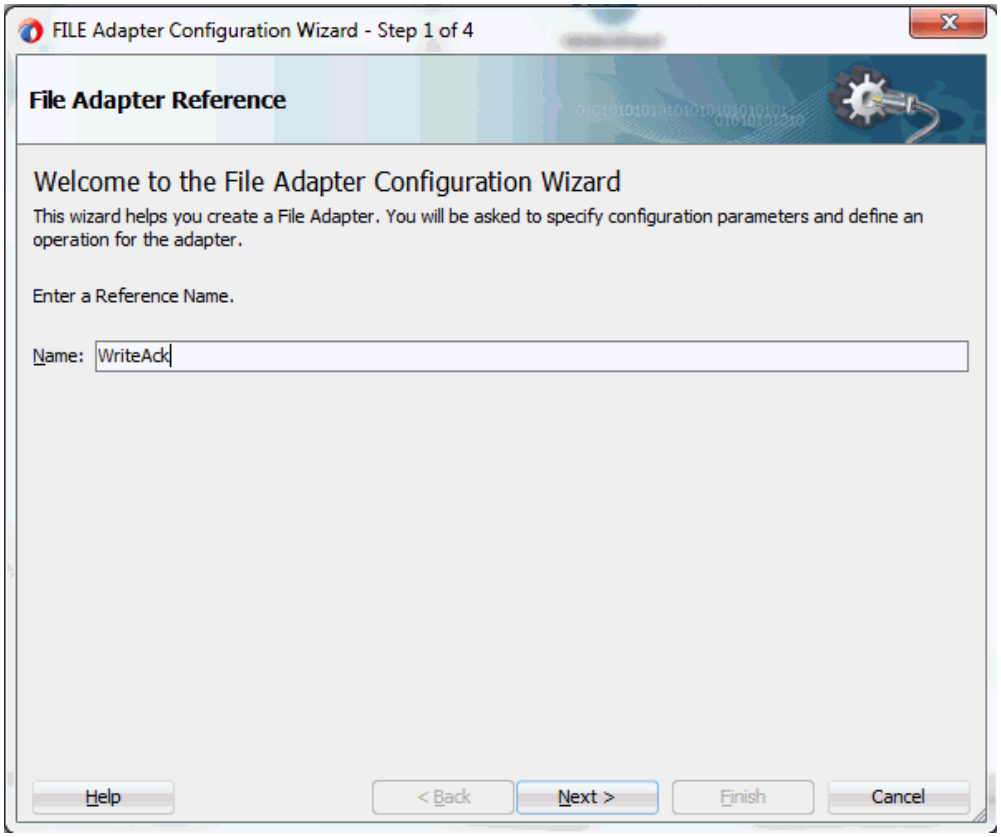
Adding a Partner Link for File Adapter

Use this step to write PO acknowledgement details in an XML file as an output file.

To add a Partner Link for File Adapter:

1. In Oracle JDeveloper BPEL Designer, click **BPEL Services** in the Component palette. Drag and drop **File Adapter** from the BPEL Services list into the right Partner Link swim lane of the process diagram. The File Adapter Reference page appears.

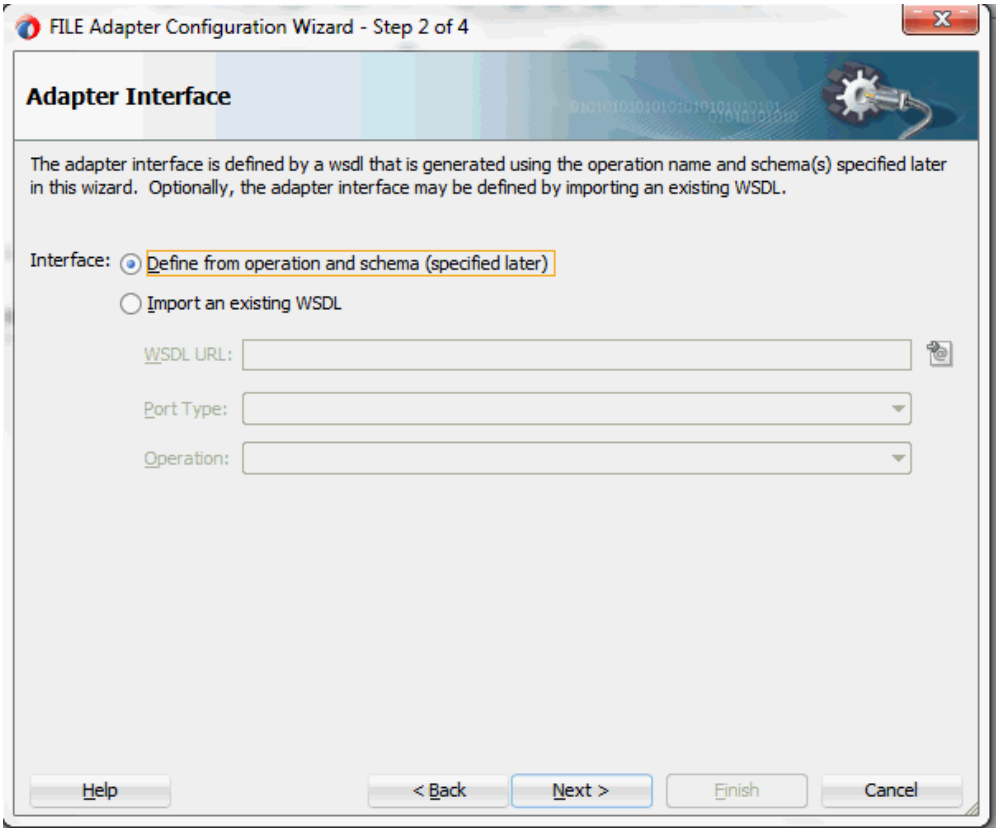
The File Adapter Reference Page



Enter a name for the file adapter service, for example, `WriteAck`.

2. Click **Next**. The Adapter Interface page appears.

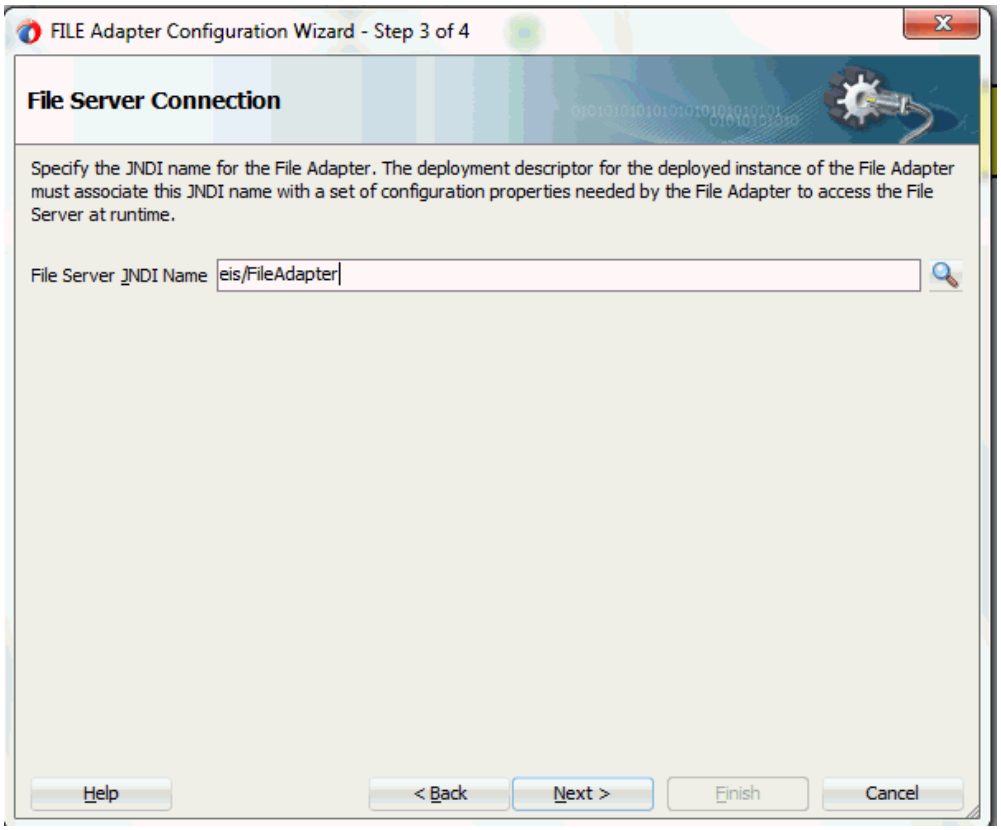
Specifying the Adapter Interface



Select the **Define from operation and schema (specified later)** radio button and click **Next**.

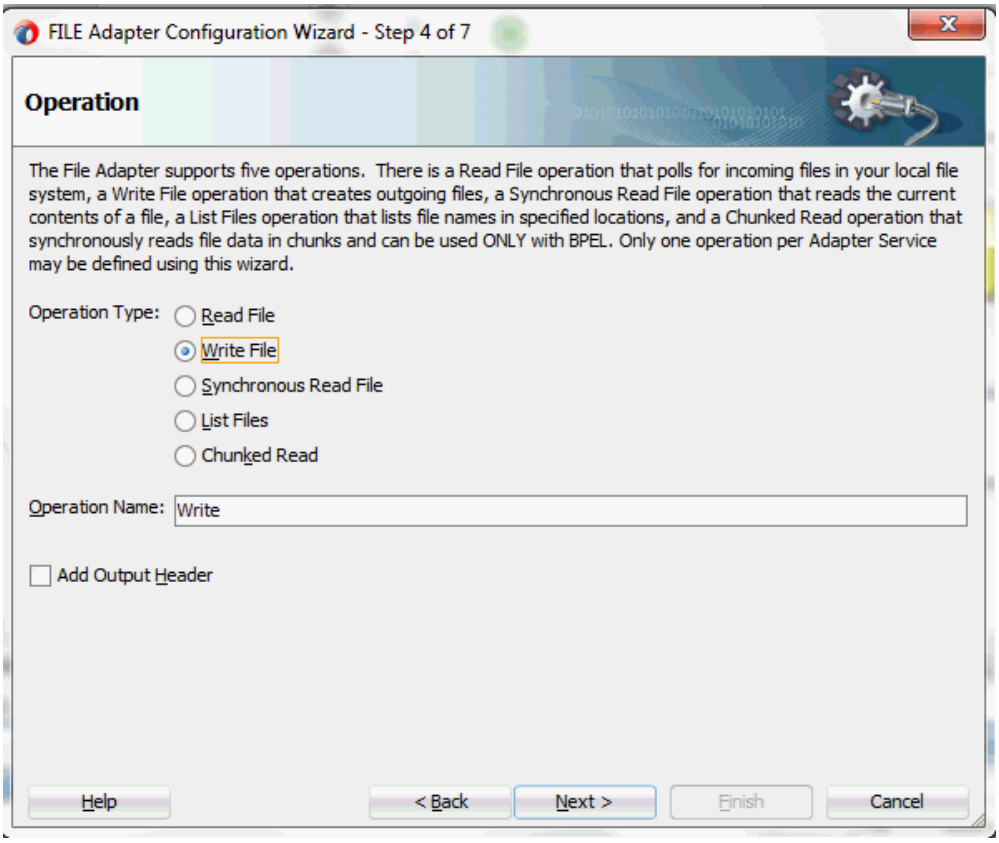
The File Server Connection page appears. Leave the default value unchanged for the File Server JNDI Name field.

Click **Next**



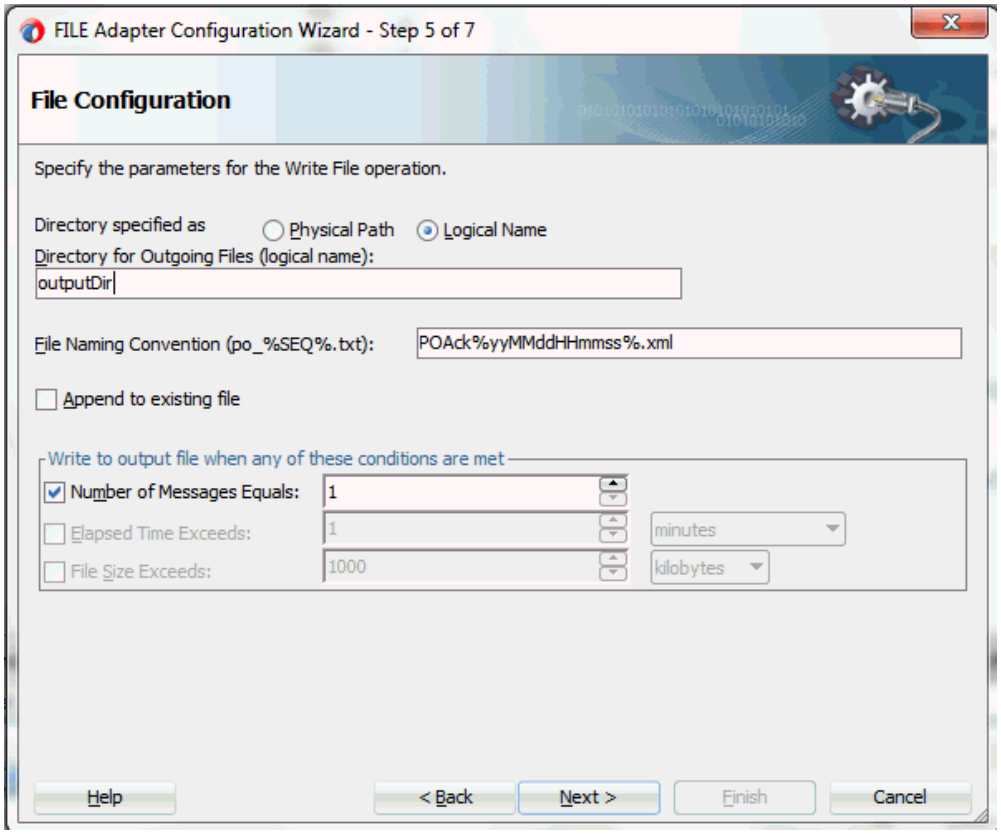
3. In the Operation page, specify the operation type. For example, select the **Write File** radio button. This automatically populates the **Operation Name** field.

Specifying the Operation



Click **Next** to access the File Configuration page.

Adapter Configuration Wizard - File Configuration Page



4. For the Directory specified as field, select **Logical Name**. Enter directory name in the Directory for Outgoing Files (logical name) field, for example, `outputDir`.

   Specify a naming convention for the output file, for example, `POAck%yyMMddJJmmss%.xml`.

   Select the **Number of Messages Equals** check box and set it to '1'.

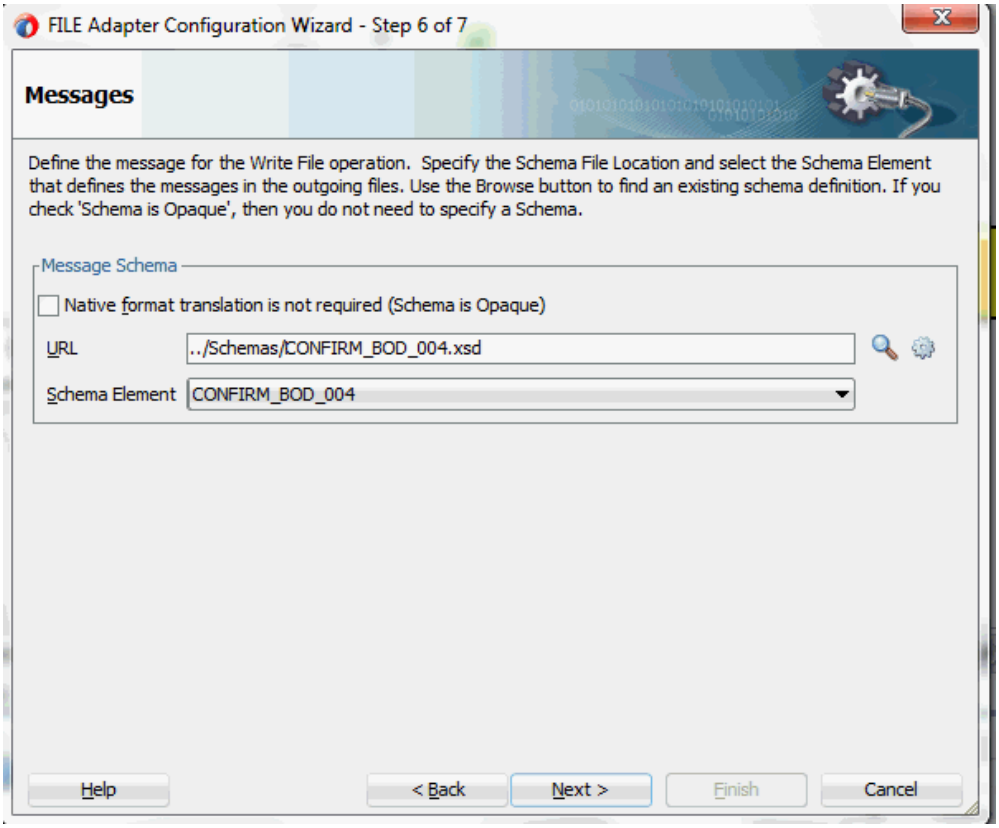
   Click **Next** to open the Messages page.

5. Select the 'browse for file schema' icon next to the URL field to locate the schema location and schema element.

   The Type Chooser dialog box appears. Expand the node by clicking **Project Schema Files > CONFIRM_BOD_004.xsd**. Select **CONFIRM_BOD_004** and click **OK**.

   The selected schema information will be automatically populated in the URL and Schema Element fields.

Adapter Configuration Wizard - Messages Page



6. Click **Next** and then **Finish**. This completes the configuration and creates the partner link with the required WSDL settings for the File Adapter service.

   The wizard also generates the `WriteAck.wsdl` WSDL file corresponding to the partner link.

   The `WriteAck` Partner Link appears in the BPEL process diagram. Double click the Partner Link service icon to open the Edit Partner Link dialog and view the details.

Adding an Invoke Activity

This step is to configure an Invoke activity to write PO acknowledgement information to an XML file through invoking the partner link for File Adapter.

To add an Invoke activity:

1. In Oracle JDeveloper BPEL Designer, expand the **BPEL Constructs** from the Component Palette. Drag and drop the **Invoke** activity from the **Web Service** section into the center swim lane of the process diagram after the **Receive** activity.

2. Link the Invoke activity to the `WriteAck` service.

   The Edit Invoke dialog appears.

3. Enter a name for the Invoke activity. In the Input tab, ensure the **Input Variable** radio button is selected. 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**, then enter a name for the variable. You can also accept the default name. Click **OK**.

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

   The Invoke activity appears in the process diagram.

Adding an Assign Activity

Use the Assign activity to take the output from the Receive activity and to provide input to the Invoke activity.

To add an Assign activity:

1. In Oracle JDeveloper BPEL Designer, expand the **BPEL Constructs** from the Component Palette. Drag and drop the **Assign** activity from the **Basic Activities** section into the center swim lane of the process diagram, between the **Receive** and **Invoke** activities that you just created earlier.

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

   Click the General tab to enter a name for the Assign activity.

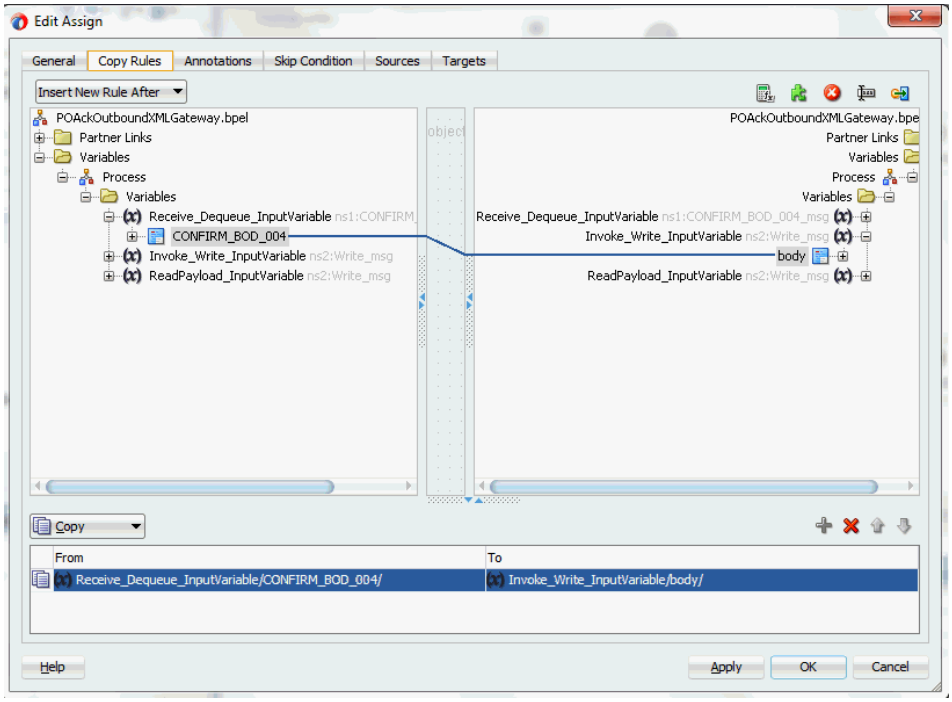
3. Select the Copy Rules tab and expand the target trees:

   - In the From navigation tree, navigate to **Variable > Process > Variables > Receive_DEQUEUE_InputVariable** and select **CONFIRM_BOD_004**.

   - In the To navigation tree, navigate to **Variable > Process > Variables > Invoke_Write_InputVariable** and select **body**.

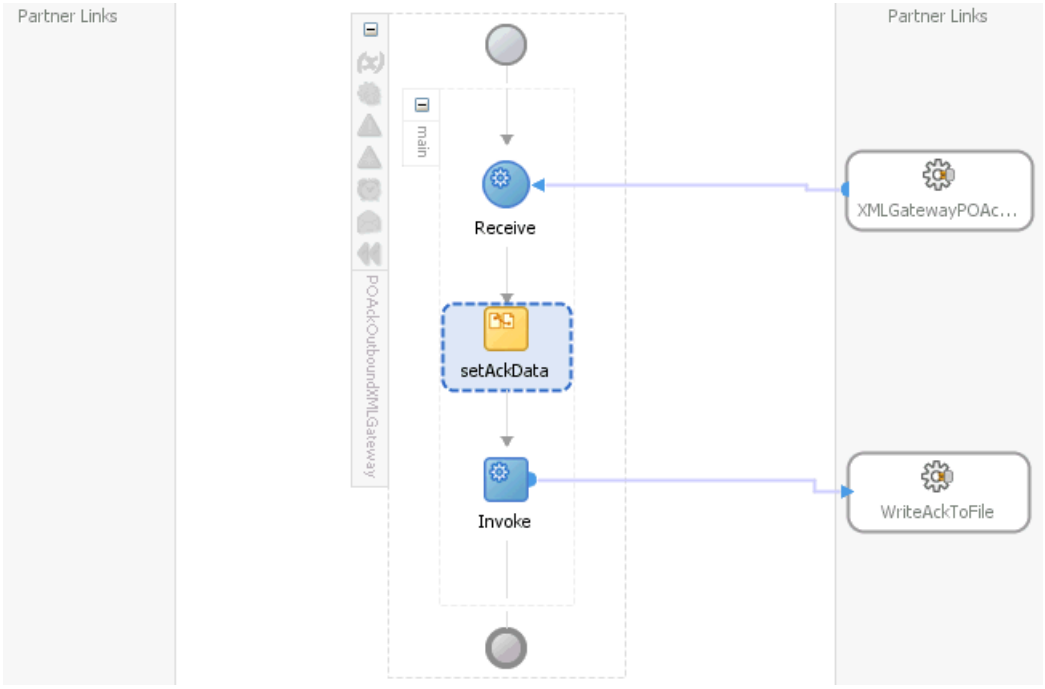
     Drag the source node (CONFIRM_BOD_004) to connect to the target node (body) that you just identified. This creates a line that connects the source and target nodes. The copy rule is displayed in the From and To sections at the bottom of the Edit Assign dialog box.

Assigning Parameters



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

The following diagram illustrates the complete BPEL process diagram:

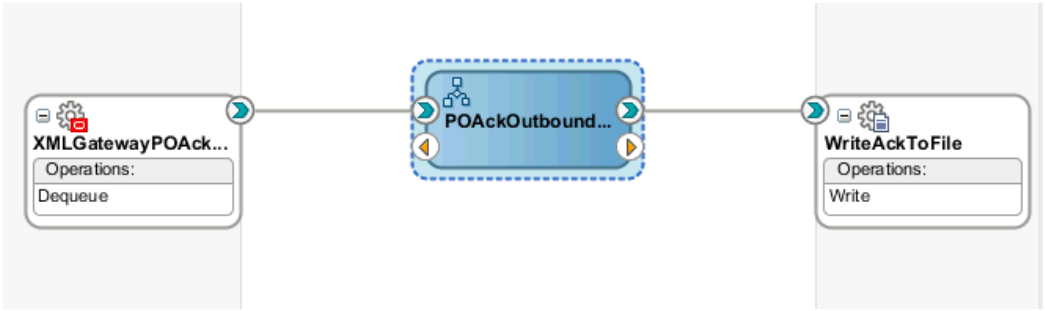


Note: Click the Source tab of `XMLGatewayOutbound` (composite.xml) to enter a value for the physical directory `outputDir` for the reference WriteEventData (such as `/usr/tmp`).

```
<property name="outputDir" type="xs:string"
many="false" override="may">/usr/tmp</property>
```

Click XMLGatewayOutbound (composite.xml) to display the Oracle JDeveloper composite diagram:

Oracle JDeveloper Composite Diagram



Run-Time Task for XML Gateway Outbound Messaging

After designing the SOA Composite application with BPEL process, you can compile,

deploy and test it.

1. Deploy the SOA Composite application with BPEL process, page 5-68
2. Test the SOA Composite with BPEL process, page 5-71

Deploying the SOA Composite Application with BPEL Process

After creating a SOA composite application with BPEL process for XML Gateway outbound message map, you need to deploy it to the Oracle WebLogic managed server. This can be achieved using Oracle JDeveloper. Once the composite is deployed, it can be tested from the Oracle Enterprise Manager Fusion Middleware Control Console.

Prerequisites

Before deploying the SOA composite with BPEL process using Oracle JDeveloper, you must have established the connectivity between the design-time environment and the run-time server. For more information, see Configuring the Data Source in Oracle WebLogic Server, page 11-1 and Creating an Application Server Connection, page 11-10.

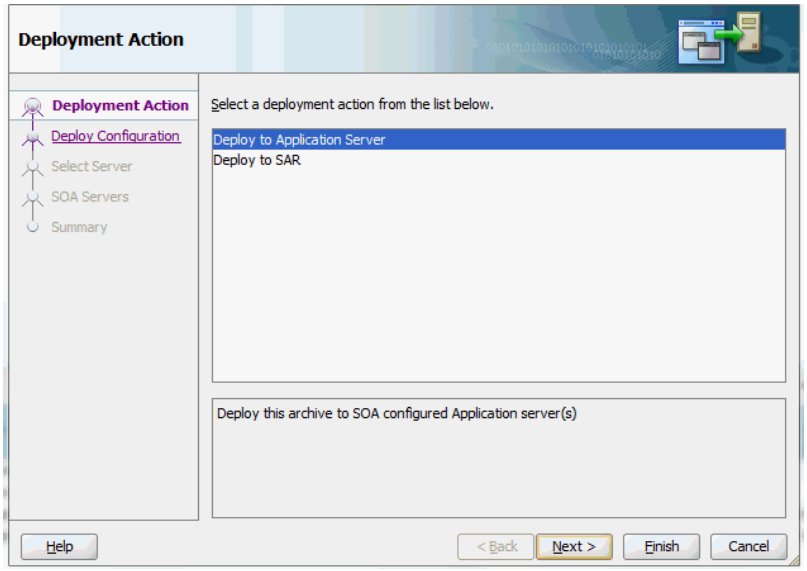
> **Note:** If a local instance of the WebLogic Server is used, start the WebLogic Server by selecting **Run > Start Server Instance** from Oracle JDeveloper. Once the WebLogic Admin Server "DefaultServer" instance is successfully started, the <Server started in Running mode> and DefaultServer started message in the Running:DefaultServer and Messages logs should appear.

To deploy the SOA Composite application with BPEL process:

1. Select the SOA Composite project in the Applications Navigator.
2. Right-click the project name. Select **Deploy > [project name] > [serverConnection]** from the menu that appears.

   For example, you can select **Deploy > XMLGatewayOutbound > soa-server1** to deploy the process if you have the connection set up appropriately.

   > **Note:** If this is the first time to set up server connection, then the Deployment Action window appears. Select 'Deploy to Application Server' and click **Next**.

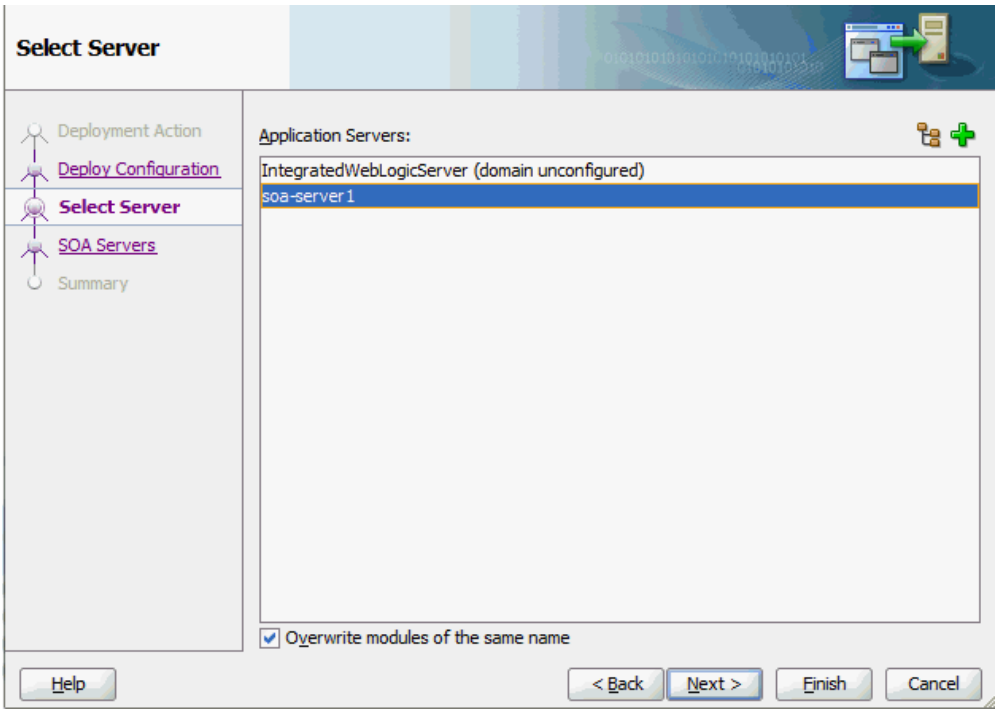


In the Deploy Configuration window, ensure the following information is selected before clicking **Next** to add a new application server:

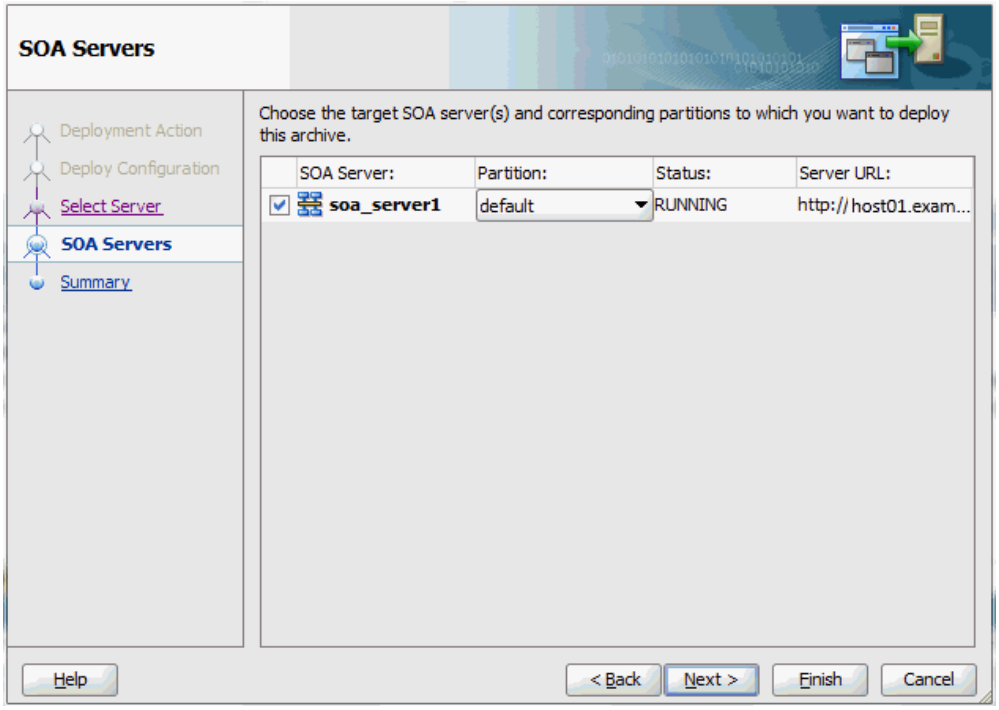
- New Revision ID: 1.0
- Mark composite revision as default: Select this check box.
- Overwrite any existing composites with the same revision ID: Select this check box.

The steps to create a new Oracle WebLogic Server connection from JDeveloper are covered inCreating an Application Server Connection, page 11-10.

3. In the Select Server page, select 'soa-server1' that you have established the server connection earlier. Click **Next**.



4. In the SOA Servers page, accept the default target SOA Server ('soa-server1') selection.



Click **Next** and **Finish**.

If you are deploying the composite for the first time from your Oracle JDeveloper session, the Authorization Request window appears. Enter username and password information specified during Oracle SOA Suite installation. Click **OK**.

5. Deployment processing starts. Monitor deployment progress and check for successful compilation in the SOA - Log window.

   Verify that the deployment is successful in the Deployment - Log window.

Testing the SOA Composite Application with BPEL Process

In Oracle E-Business Suite, 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*. This guide is part of the Oracle E-Business Suite Documentation Library which can be accessed on the Oracle Technology Network (OTN).

If you have used a File Adapter to write outbound messages from Oracle E-Business Suite to files, you can check the output directory location for the presence of these files

after the BPEL process contained in a SOA Composite application has been executed.

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 SOA Composite application with BPEL process:

1. Log in to Oracle E-Business Suite 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 on the Trading Partner Setup form as follows:
   - Trading Partner Type: Customer
   - Trading Partner Name: Business World
   - Trading Partner Site: Enter a trading partner site information, such as 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 the associated password 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 the 'Primary' check box

Save your work.

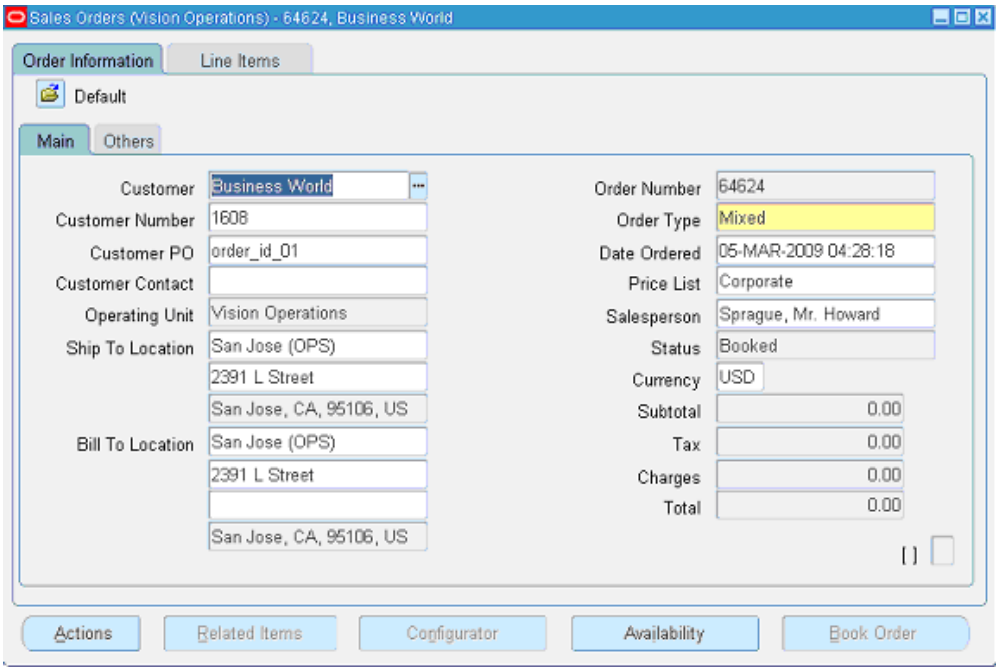
Use the following steps to generate acknowledgement for already created order:

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

2. Retrieve the order that you have created earlier by entering the order ID in the Customer PO field. For example, enter `order_id_01`.

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

Booking an Order

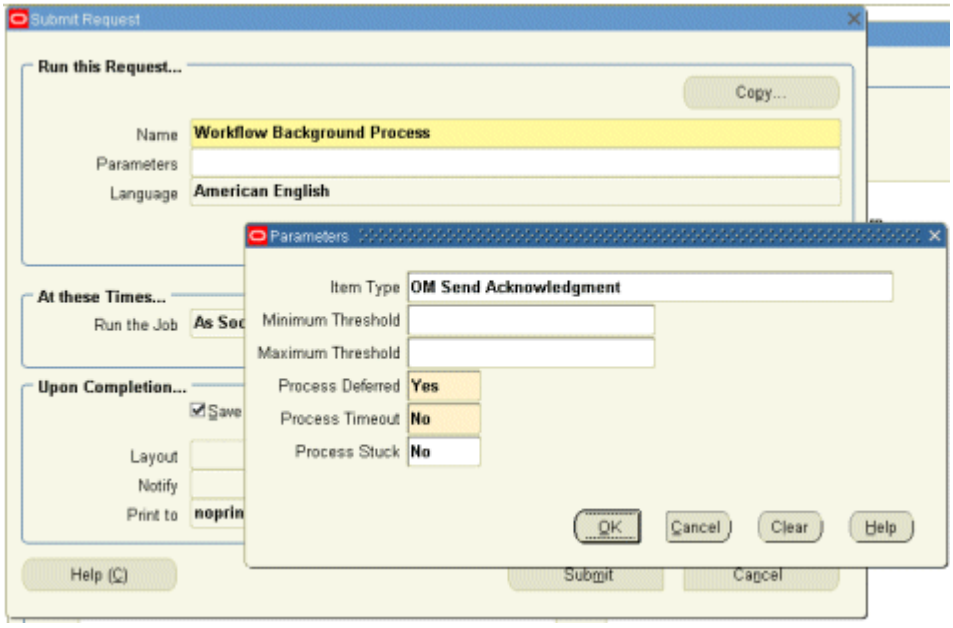


4. Switch to the System Administrator responsibility and select Request > Run.

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

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

Specifying Parameters



- 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'.

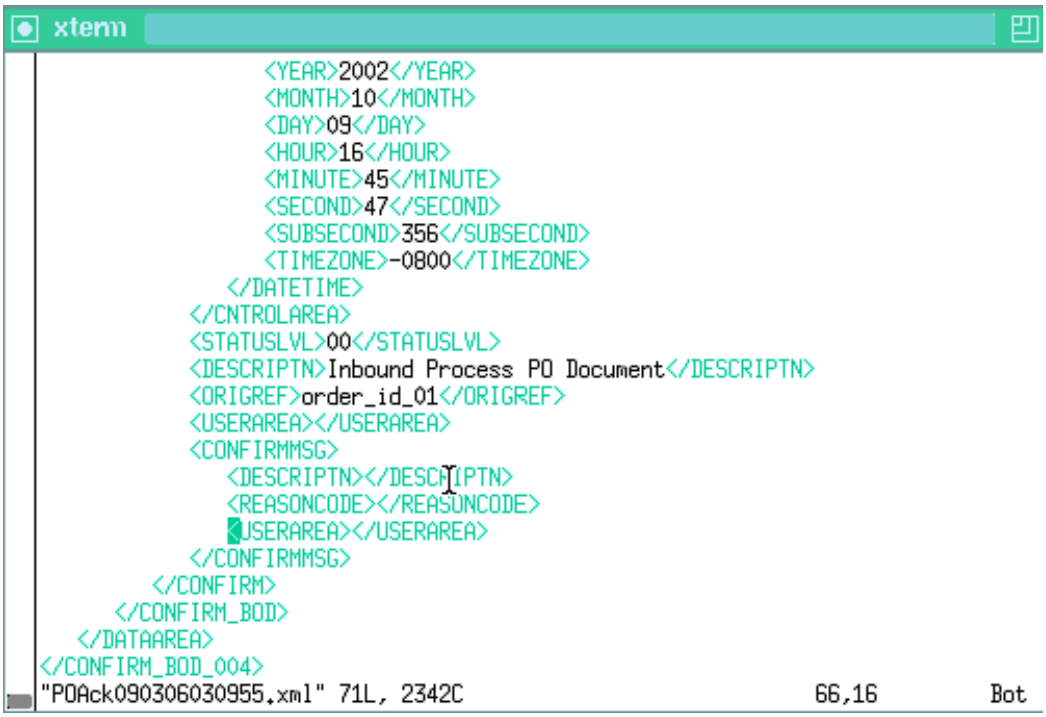
On approval of the order, Oracle E-Business Suite triggers the workflow that creates the Purchase Order Acknowledgement flow and sends out the PO Acknowledgement as an XML file. The workflow delivers the Confirm BOD as the PO Acknowledgement to the ECX_OUTBOUND queue for delivery to the other system.

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 contained in a SOA Composite application is deployed, the process is continuously polling the ECX_OUTBOUND queue for PO acknowledgement. It also writes PO Acknowledgement in the physical directory mentioned in `composite.xml` after receiving from XML Gateway ECX_OUTBOUND queue.

You can open file in text editor and search for ORIGREF element. Its value should be same as order id (`order_id_01` whose order was booked.



Troubleshooting

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 E-Business Suite for the Adapter.

  The Adapter Configuration wizard 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 E-Business Suite 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 E-Business Suite, 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 E-Business Suite.

Enable logging for adapter to see if the issue is on the middleware side. How to enable logging for Oracle E-Business Suite Adapter, see Enabling Logging for Adapters, page 4-125.

6

Using Business Events

This chapter covers the following topics:

- Overview of Business Events
- Design-Time Tasks for Outbound Business Events
- Creating a New SOA Composite Application with BPEL Process
- Creating a Partner Link
- Configuring the Receive Activity
- Adding a Partner Link for the File Adapter
- Configuring an Invoke Activity
- Configuring an Assign Activity
- Run-Time Tasks for Outbound Business Events
- Deploying the SOA Composite Application with BPEL Process
- Testing the SOA Composite Application with BPEL Process
- Troubleshooting

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

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** - 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.
- **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.

Business Event Groups

A business event group is a type of event that contains multiple individual business events. Once an event group is defined, a subscription that is registered for an event group will be executed when any of the individual events within it is triggered.

With the support for business event groups, different business events belonging to an even group can be handled through a single partner link. A service created for an event group would be able to dequeue payloads corresponding to any of the events within the group.

To support existing partner links for outbound business events which are part of an event group, a workaround has to be followed. For detailed information, see Support for Business Events through Existing Partner Links, page A-1.

> **Note:** If individual events under a group are subscribed, then two messages would be enqueued into the WF_BPEL_Q queue. Users requiring only one message would need to disable the subscription for the individual event which enqueues the messages into WF_BPEL_Q.

For information about creating a partner link with a business event group, see Creating a Partner Link with a Business Event Group, page 6-14.

Design-Time Tasks for Outbound Business Events

This section describes the tasks required to configure Oracle E-Business Suite Adapter using the Adapter Configuration Wizard in Oracle JDeveloper.

Creating Service Artifacts for Business Event Consumption

While creating a partner link for listening to a business event, the following tasks are performed behind the scenes:

- Creation of WF_BPEL_Q queue
- Creation of an entry for WF_BPEL_Q in WF_AGENTS table
- Creation of a subscription for the event being listened to

To facilitate re-creating the above entries on a different Oracle E-Business Suite instance, a script is generated in the project folder. In addition, a drop script is created which can be run manually on the corresponding instance to clean up all the tasks performed above. Running the drop script is optional, and is not recommended. The drop script would delete the WF_BPEL_Q queue, which can impact other composites, listening to events on the same Oracle E-Business Suite instance.

Multiple BPEL Processes Consuming the Same Business Event

Oracle E-Business Suite Adapter can handle multiple BPEL processes consuming the same business event. Oracle E-Business Suite Adapter 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 Oracle E-Business Suite Adapter 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 Oracle E-Business Suite Adapter. Oracle E-Business Suite Adapter 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.

SOA Composite Application with BPEL Process Scenario

Take a PO XML Raise business event as an example.

When a purchase order is created and approved, a purchase order approved business event `oracle.apps.po.evnt.xmlpo` is raised. The subscription to this event is created in the background to listen to the business event and get event details. The event data will be passed through BPEL process activities and then written in XML file as an output file.

When the SOA Composite application with BPEL process has been successfully executed after deployment, you should get the same purchase order information from the output file once a purchase order is approved.

Prerequisites to Configure Outbound Business Events

- The WF_Deferred Agent Listener must be up and running on the target instance.
- The event should be enabled for BPEL to subscribe to it. The event should not be in the disabled mode.

SOA Composite Application with BPEL Process Creation Flow

Based on the PO XML Raise business event scenario, the following design-time tasks are discussed in this chapter:

1. Create a new SOA Composite application with BPEL process, page 6-6.
2. Create a partner link, page 6-10.
3. Configure a Receive activity, page 6-20.
4. Add a partner link for the file adapter, page 6-22.
5. Configure an Invoke activity, page 6-29.
6. Configure an Assign activity, page 6-31.

Creating a New SOA Composite Application with BPEL Process

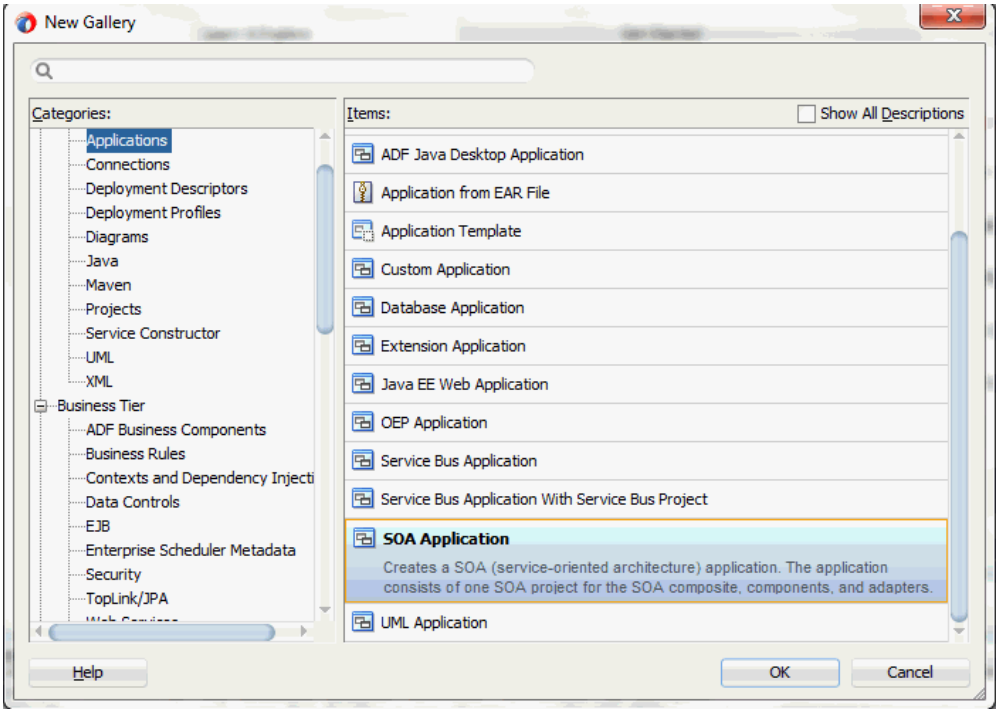
To create a new SOA Composite application with BPEL process:

1. Launch Oracle JDeveloper.

2. Click **New Application** in the Application Navigator.

   The New Gallery page appears. Select **SOA Application** from the Items list.

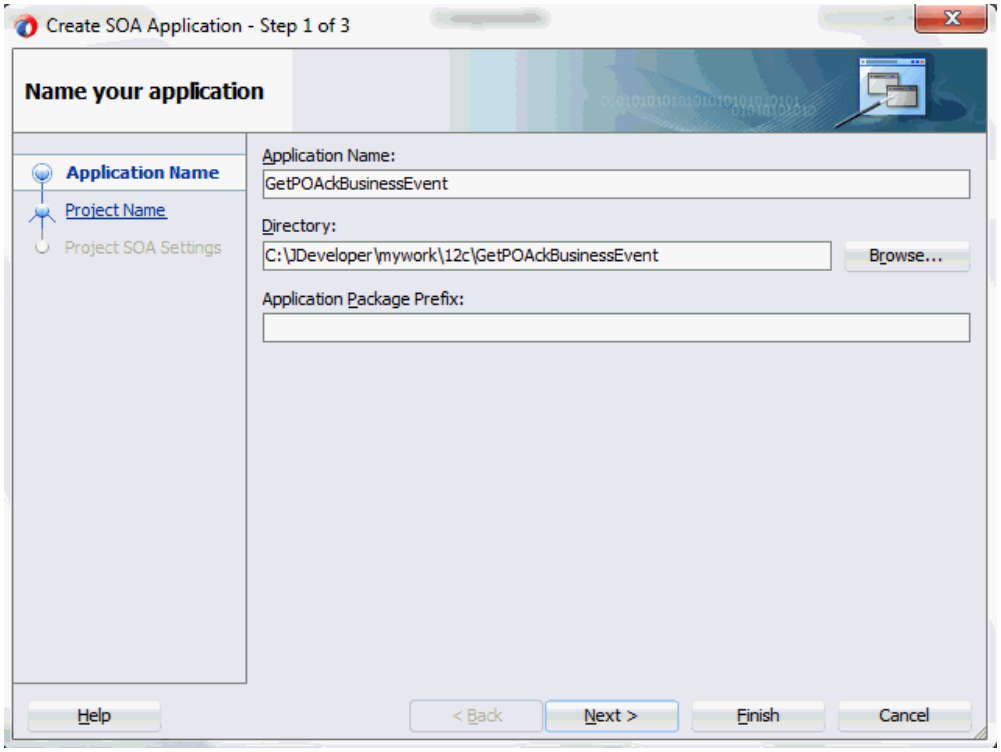
   The New Gallery Page

   Click **OK**.

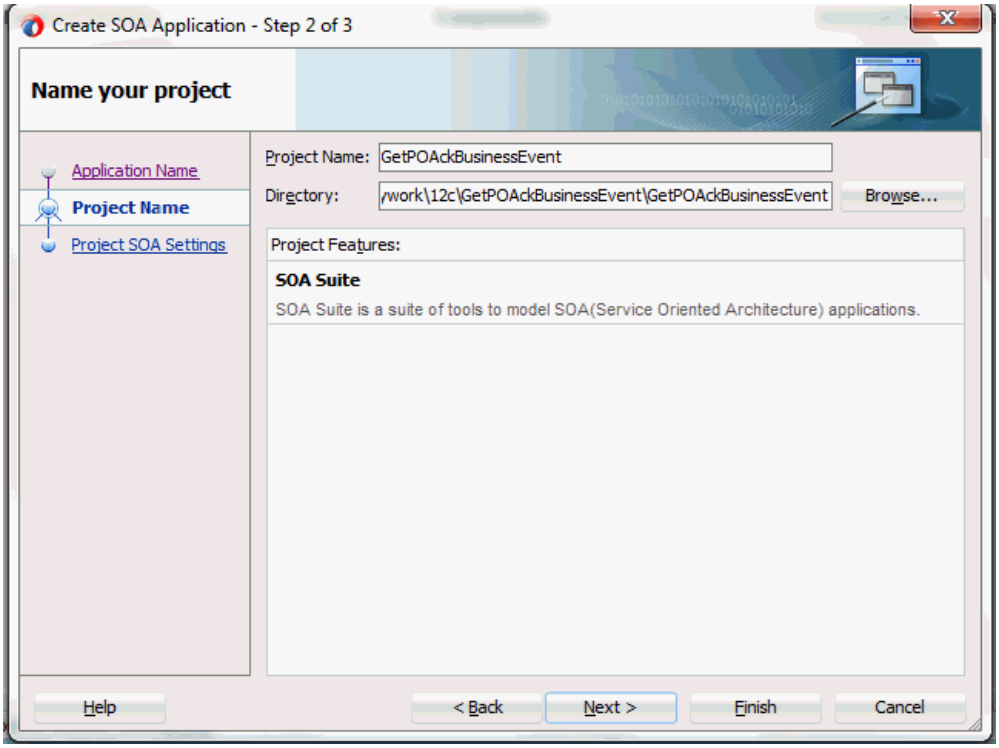
3. The Create SOA Application - Name your application page is displayed.

The Create SOA Application - Name your application Page



4. Enter an appropriate name for the application in the **Application Name** field. Click **Next**. The Create SOA Application - Name your project page is displayed.

The Create SOA Application - Name your project Page

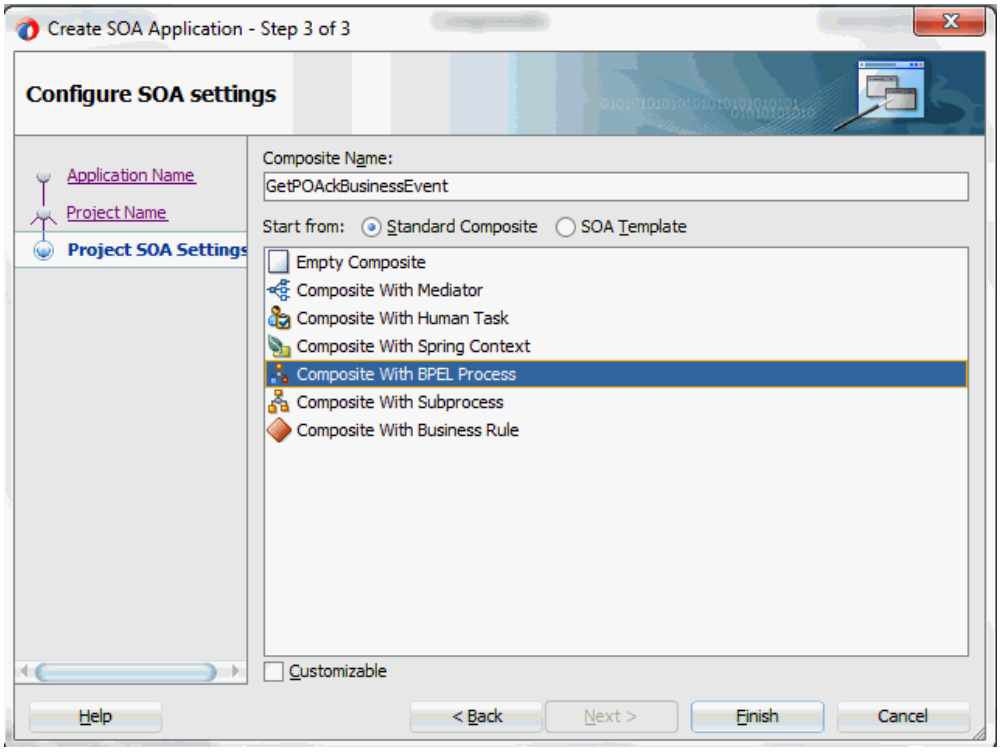


5. Enter an appropriate name for the project in the **Project Name** field. For example, `GetPOAckBusinessEvent`.

   Notice that **SOA Suite** is shown in the Project Features section.

6. Click **Next**. The Create SOA Application - Configure SOA settings page is displayed.

The Create SOA Application - Configure SOA settings Page

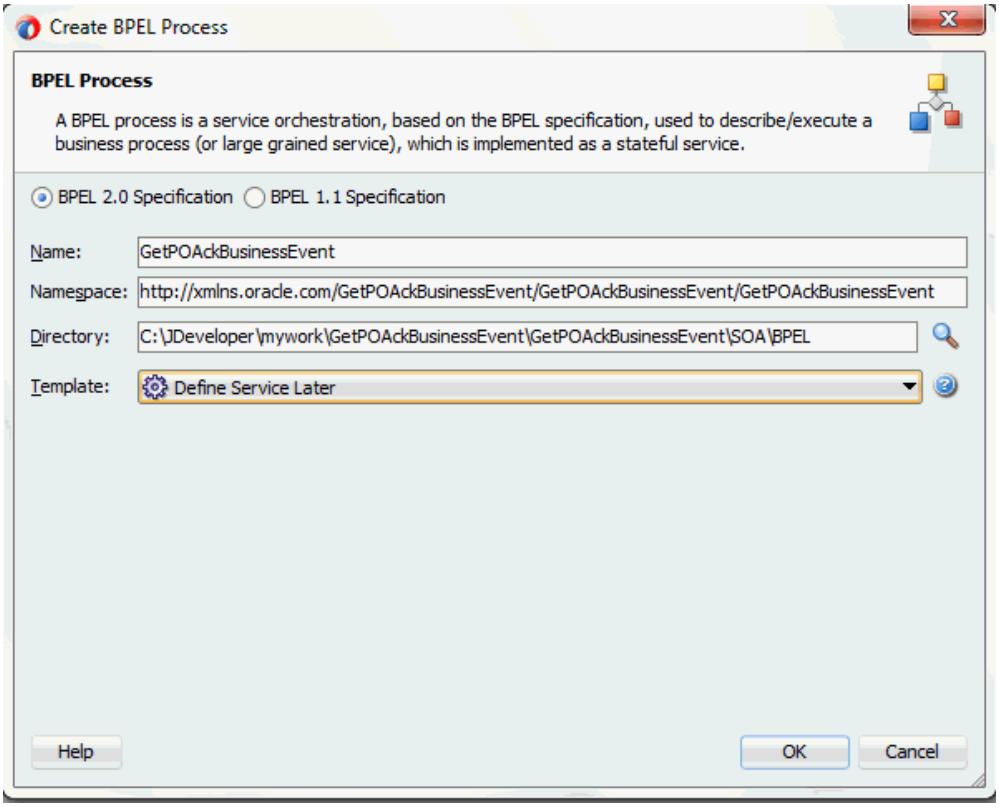


7. Leave the default **Standard Composite** selection unchanged for the Start from field.

   Select **Composite With BPEL Process** from the Standard Composite list, and then click **Finish**. You have created a new application, and a SOA project. This automatically creates a SOA Composite.

   The Create BPEL Process page is displayed.

The Create BPEL Process Page



8. Leave the default **BPEL 2.0 Specification** selection unchanged. This creates a BPEL project that supports the BPEL 2.0 specification.

   Enter an appropriate name for the BPEL process in the **Name** field. For example, `GetPOAckBusinessEvent`.

   Select **Define Service Later** from the **Template** field. Click **OK**.

   An empty BPEL process is created. The required source files including bpel and wsdl, using the name you specified (for example, `GetPOAckBusinessEvent.bpel` and `GetPOAckBusinessEvent.wsdl`) and `GetPOAckBusinessEvent` (composite.xml) are also generated.

Creating a Partner Link

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

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 add a partner link:

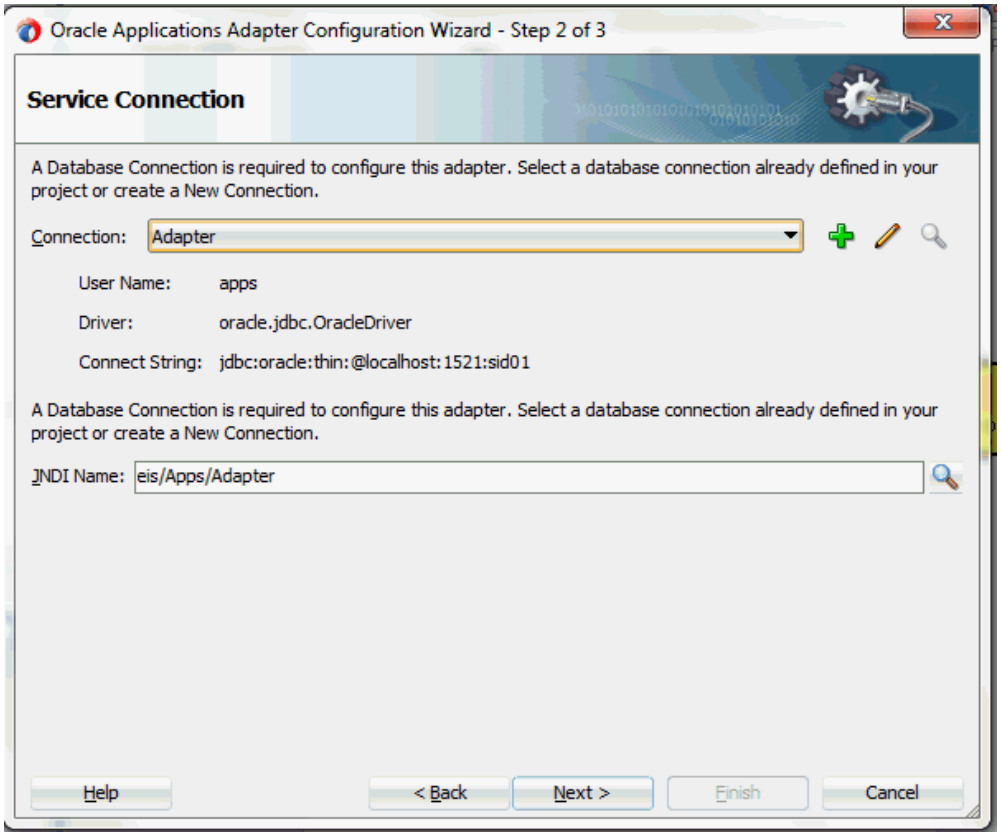
1. Click **BPEL Services** in the Component palette.

   Drag and drop **Oracle E-Business Suite** (formerly known as **Oracle Applications**) from the **BPEL Services** list into the right Partner Link swim lane of the process diagram. The Oracle E-Business Suite Reference page appears.

   Enter a reference name in the **Name** field. For example, `GetPOApprovalEvent`.

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

Specifying a Database Service Connection



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 **Create a New Database Connection** icon.

How to define a new database connection, see Create a New Database Connection, page 5-14.

- 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 Oracle E-Business Suite Adapter 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 Fusion Middleware User's Guide for Technology Adapters.*

4. Once you have completed creating a new connection for the service, you can add a business event by browsing through the list available in Oracle E-Business Suite.

   Click **Next**.

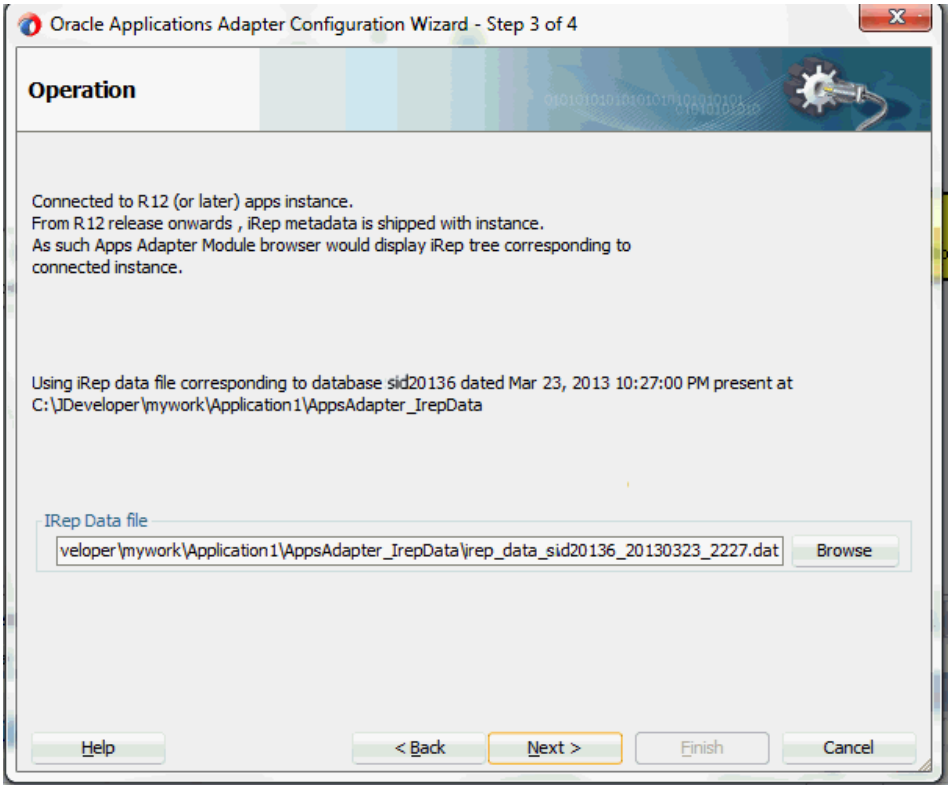
   If you are connecting to Oracle E-Business Suite Release 12, then the **IREP File not present** dialog appears indicating that Adapter could not find the Oracle Integration Repository data file corresponding to the database you are connecting in your workspace. Absence of the data file would make browsing 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 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.

     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



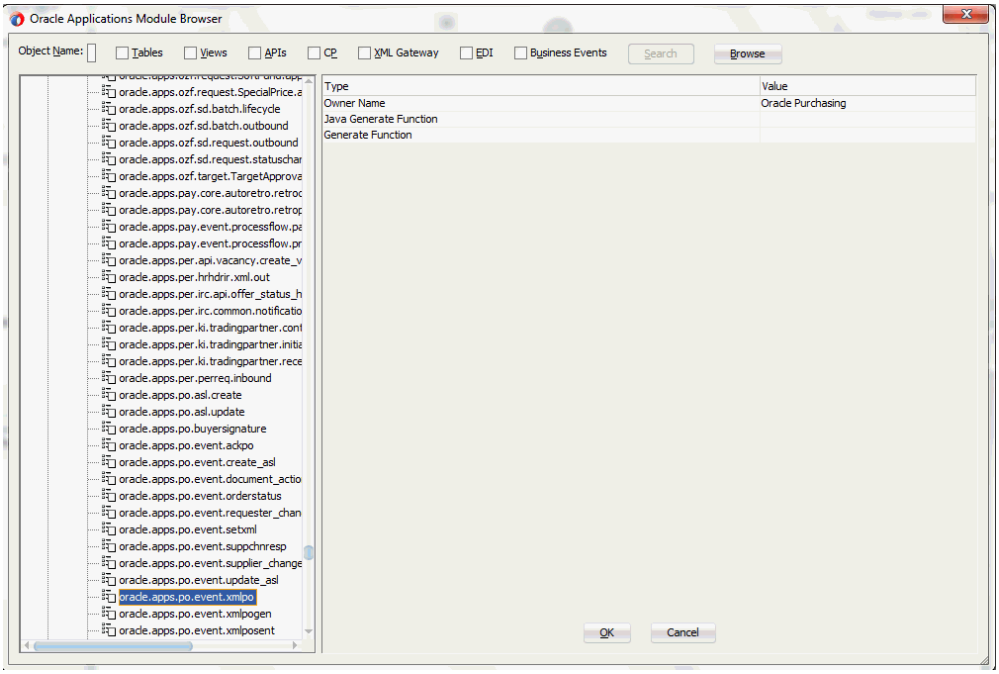
- Click **No** to query the Integration Repository data file from the live database you are connecting to display the Integration Repository tree.

  In this example, click **No**. Click **Next** in the Operation page to open the Oracle E-Business Suite Module Browser.

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

   To select a business event, 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 E-Business Suite. Select the appropriate business event, for example, **oracle.apps.po.event.xmlpo**, and click **OK**. The Application Interface page is displayed with selected business event.

Selecting a Business Event from the Module Browser

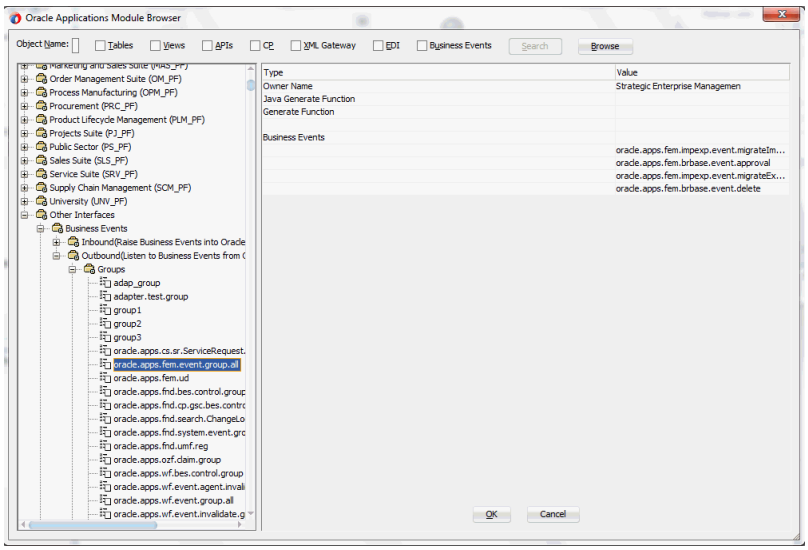


Note: Creating a Partner Link with a Business Event Group

Business event groups appear under the **Other Interfaces > Business Events > Outbound > Groups** node from the Oracle E-Business Suite Module Browser.

If a business event group (such as **oracle.apps.fem.event.group.all**) is selected from the Groups node, detailed event group information including the event group owner name, generate function information if any, and each individual event contained in the selected event group is listed in the right pane of the Oracle E-Business Suite Module Browser window.

Selecting a Business Event Group from the Oracle E-Business Suite Module Browser

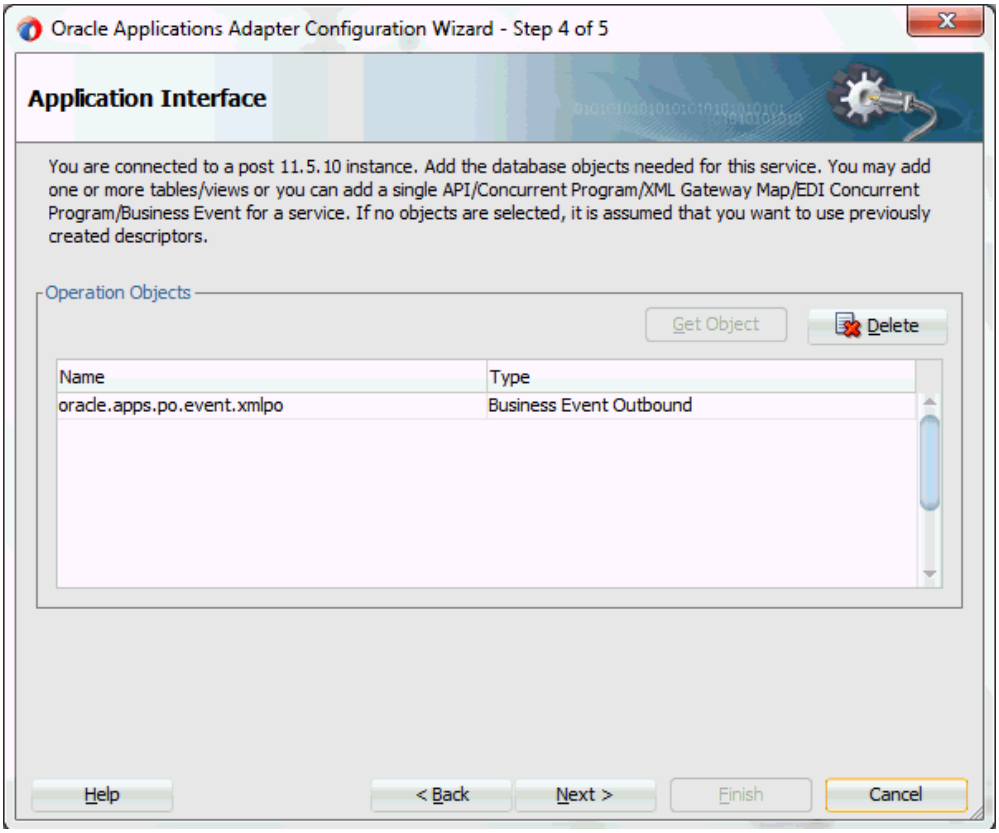


Additionally, select 'Any Schema' in the WF Event Schema Definition page later on for the business event payload. This allows XML payload of any schema to be attached to event payload. You can also verify the JCA Property "MessageSelectorRule" for the selected event group (described in Step 10) once the partner link is created.

This feature applies to Oracle E-Business Suite Release 12 and Release 11.5.10. For more information about business event groups, see Business Event Groups, page 6-3.

6. Click **OK** to display the Application Interface page.

Adapter Configuration Wizard - Application Interface Page



7. Click **Next** in the Application Interface page. The WF Event Schema Definition page for business event payload appears.

8. **Specifying Event Schema**

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

   > **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, and you will proceed to the next step.

   - **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**

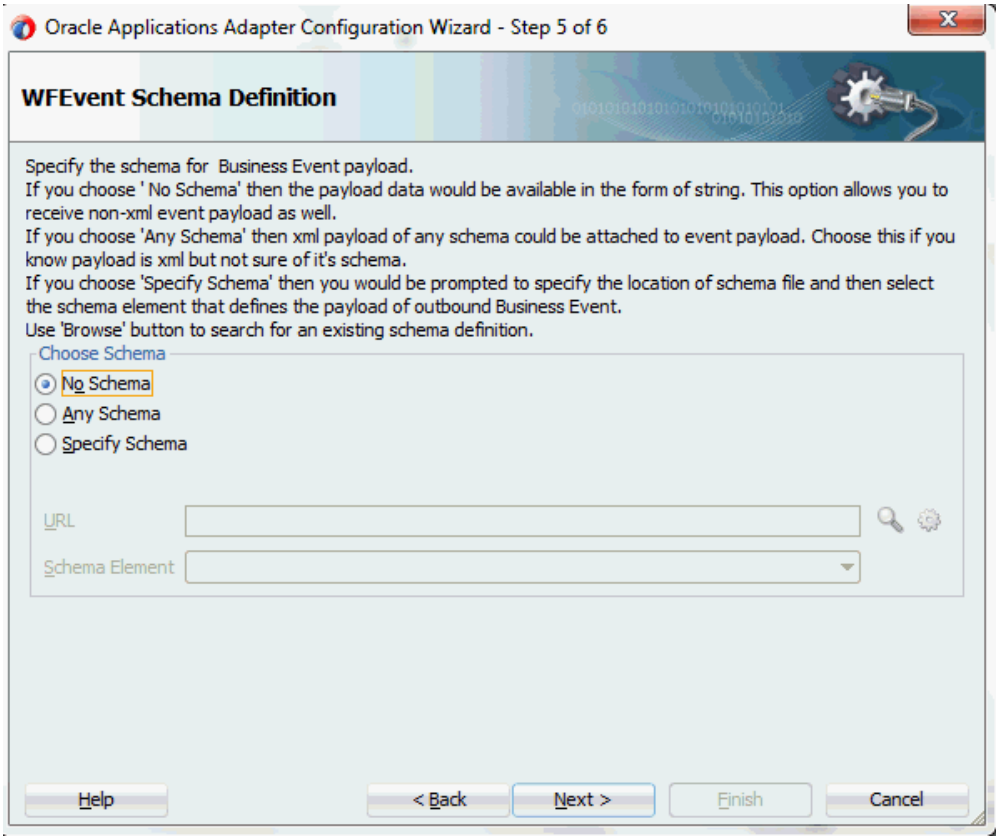
If you select the Any Schema option, then XML payload of any schema could be attached to event payload. You should select this option if you know the payload is XML, but not sure of its schema.

> **Note:** If a business event group is selected for the partner link creation, select the 'Any Schema' option. This allows any schema to be attached to a business event group.

- **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.

Selecting Business Event Payload Schema

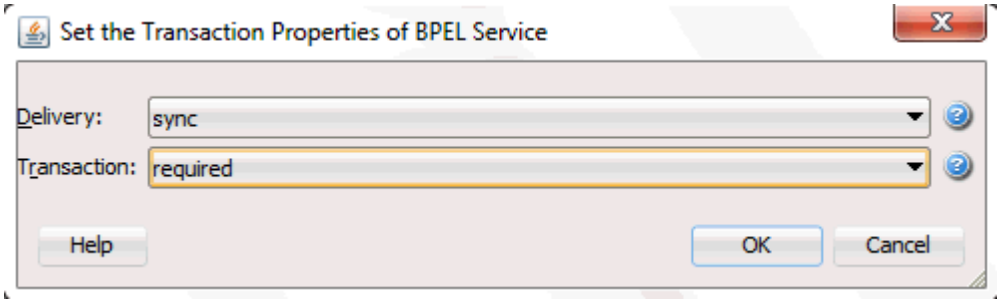


Select 'No Schema' and click **Next**.

9. The Finish page appears indicating that you have finished defining the business event service.

10. The Set Transaction Properties of BPEL Service dialog appears. Ensure that 'sync' is selected in the Delivery field and 'required' is selected in the Transaction field.

Setting the Transaction Properties of BPEL Service



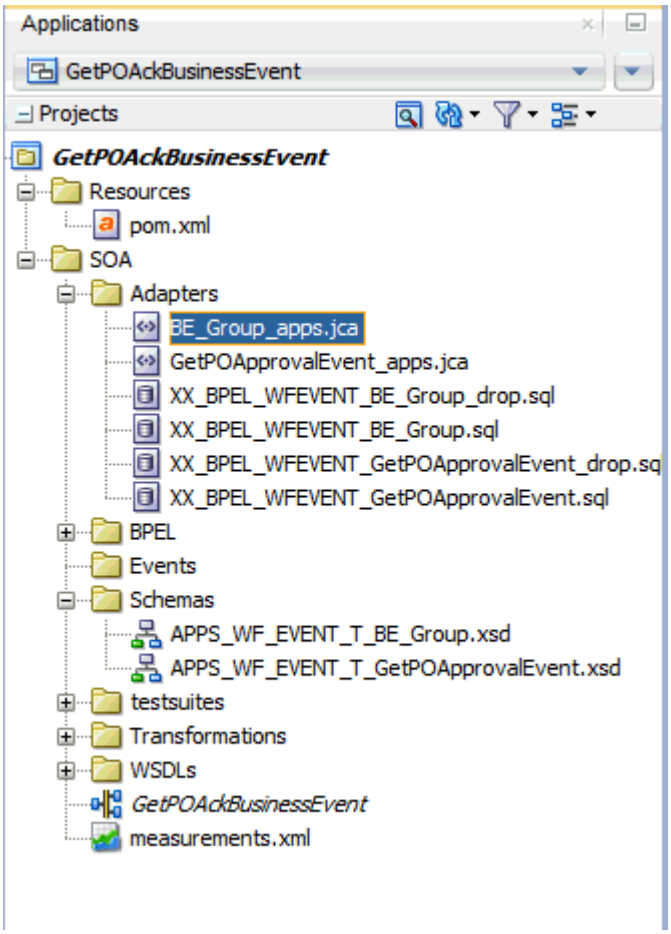
Click **OK** to complete the partner link configuration.

The wizard generates the GetPOApprovalEvent WSDL file corresponding to the **oracle.apps.po.event.xmlpo** business event service.

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.

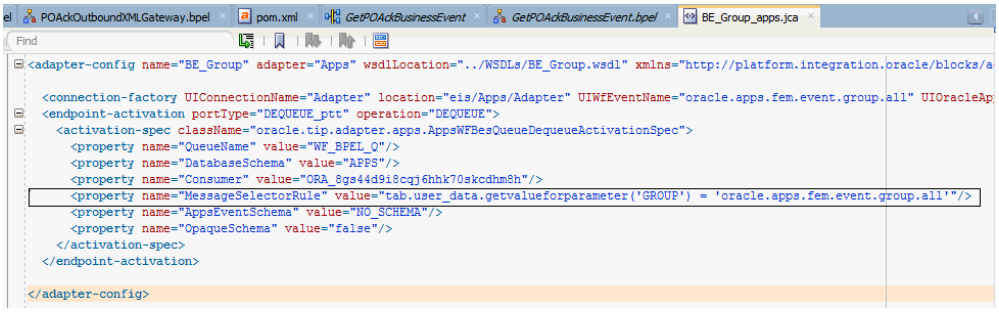
11. **Verifying JCA Property "MessageSelectorRule" for an Event Group Partner Link**

    If a business event group is selected for the partner link creation, you can verify the business event group Jca property by selecting the partner link service (such as `BE_Group_apps.jca` from the **SOA Content > Adapters** folder.



Click the Source tab to display the .jca file. Notice that the JCA property "MessageSelectorRule" contains the value of

```
value="tab.user_data.getvalueforparameter ('GROUP') =
'oracle.apps.fem.event.group.all'".
```



Configuring the Receive Activity

The next task is to configure a Receive activity to receive event details from the partner

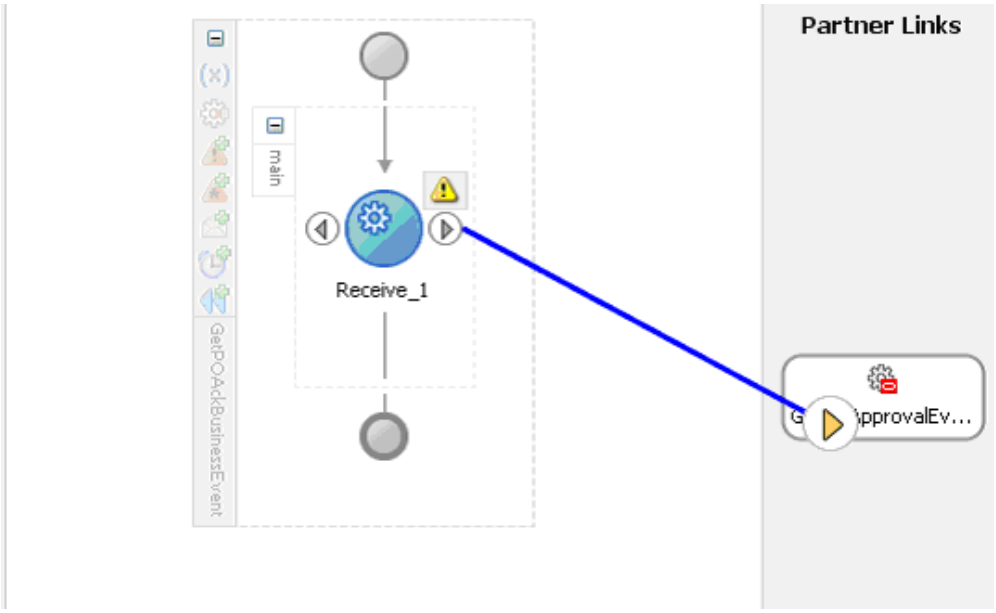
link that you just configured for the adapter service as an input to the Assign activity.

To configure the Receive activity:

1. In JDeveloper BPEL Designer, expand the **BPEL Constructs** from the Component Palette. Drag and drop **Receive** from the **Web Service** section into the center swim lane of the process diagram.

2. Link the **Receive** activity to the partner link `GetPOApprovalEvent` that you just created earlier.

Associating the Receive Activity with the Partner Link



The Receive dialog appears.

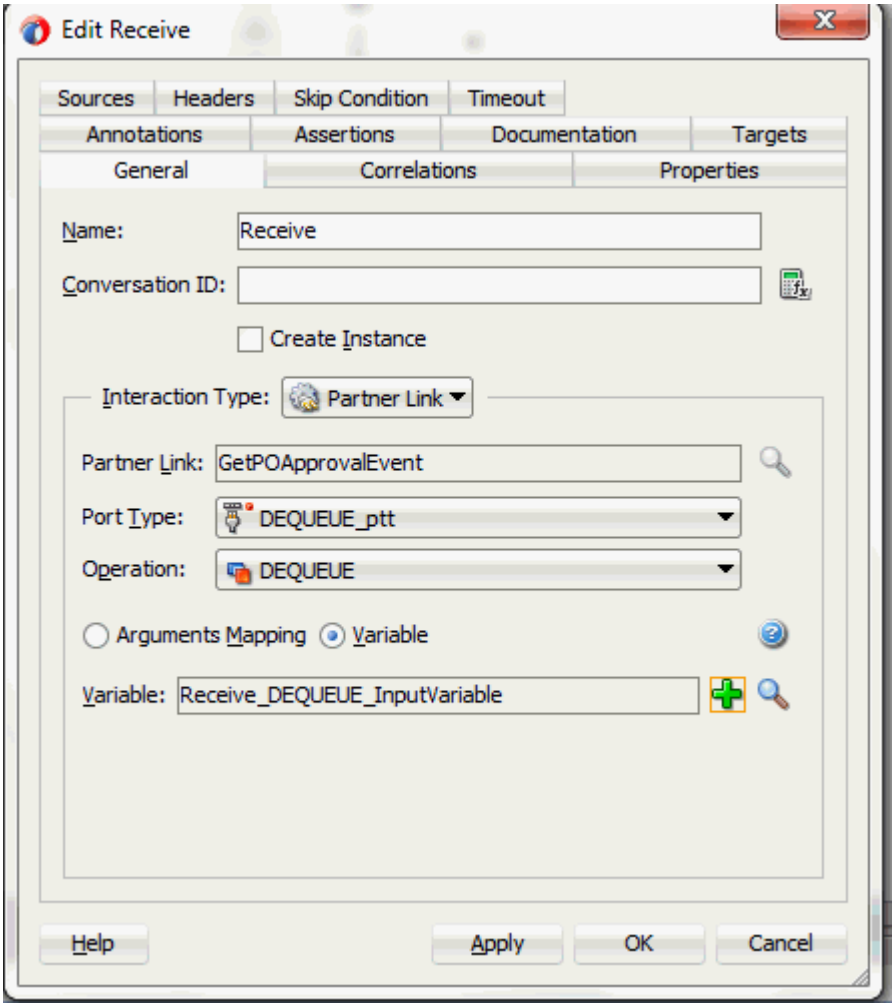
3. Enter an appropriate name for the Receive activity.

   The Dequeue **Operation** is automatically selected since the partner link has been configured with an outbound business event.

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.

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

Configuring the Receive Activity



6. Click **Apply** in the Receive dialog, then click **OK**.

Adding a Partner Link for the File Adapter

If you are configuring an outbound business event, you need to add another partner link for the file adapter. This allows the outbound business event to write the data to the XML file.

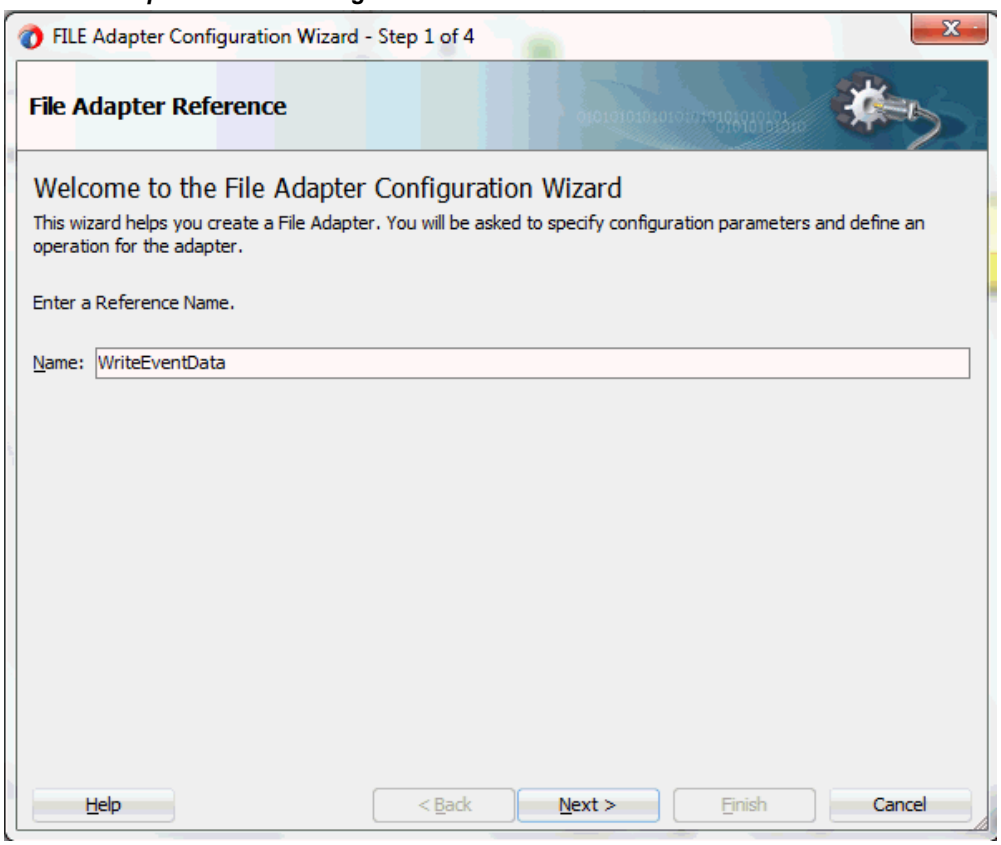
To add a partner link for the file adapter:

1. In Oracle JDeveloper BPEL Designer, click **BPEL Services** in the Component palette.

   Drag and drop **File Adapter** from the BPEL Services list into the right Partner Link

swim lane of the process diagram. The File Adapter Reference page appears.

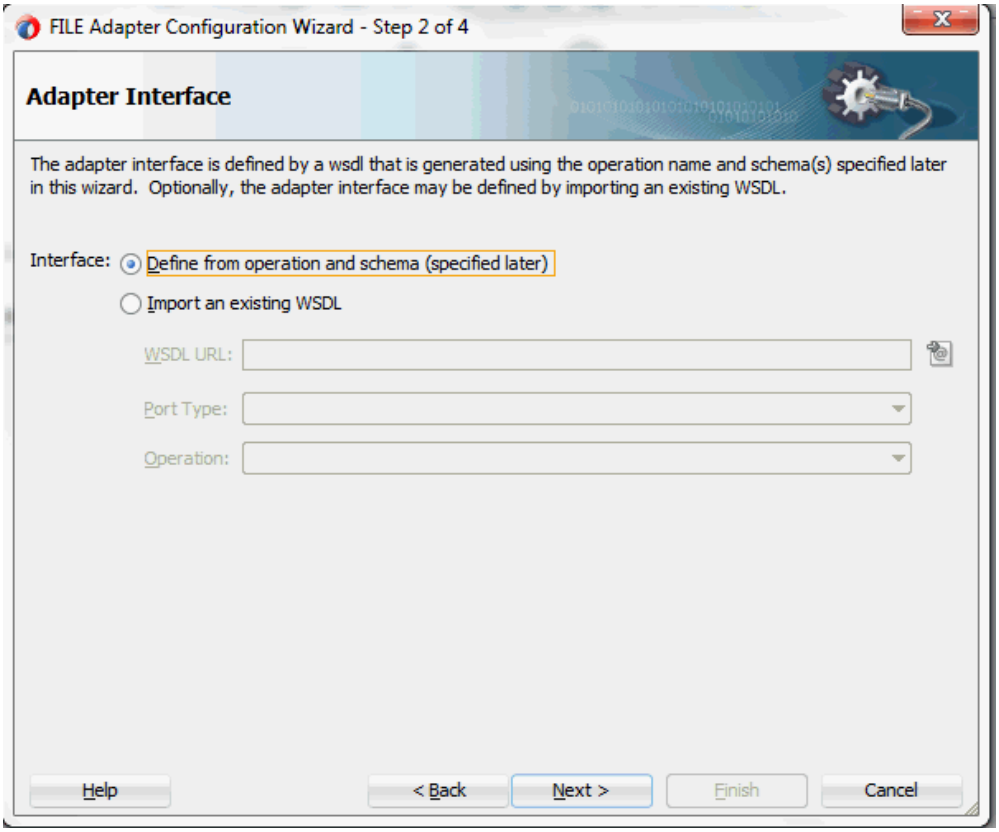
The File Adapter Reference Page



Enter a name for the file adapter service. For example, enter `WriteEventData`.

2. Click **Next**. The Adapter Interface page appears.

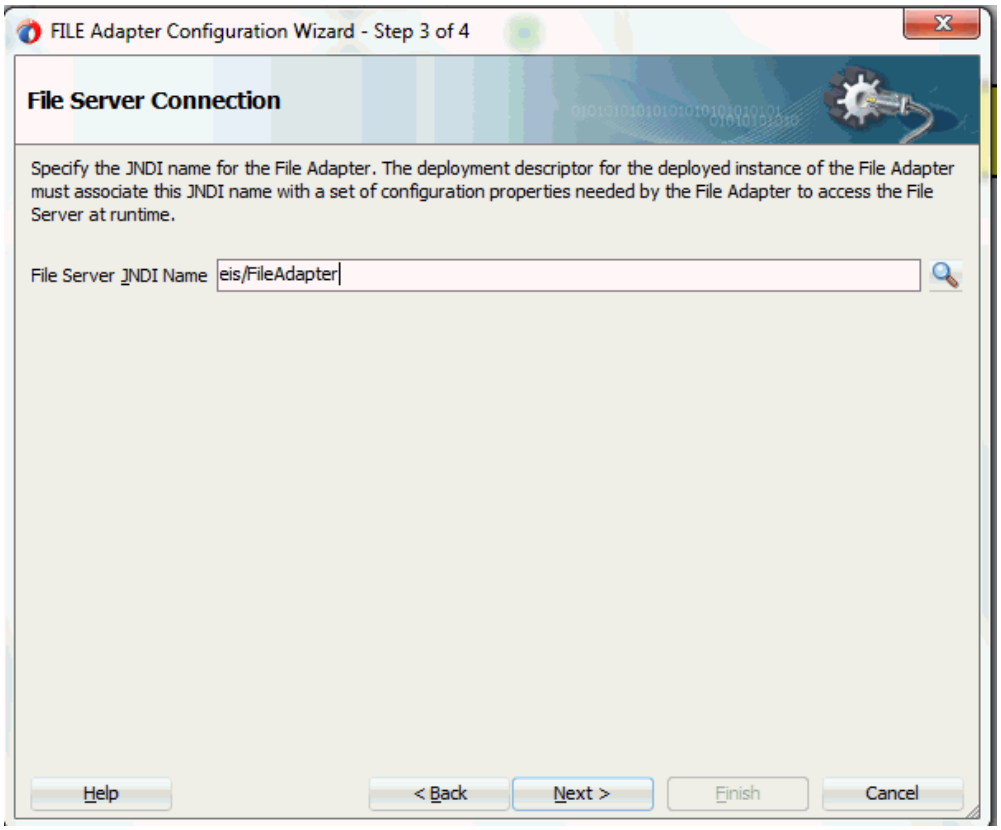
Specifying the Adapter Interface



Select the **Define from operation and schema (specified later)** radio button and click **Next**.

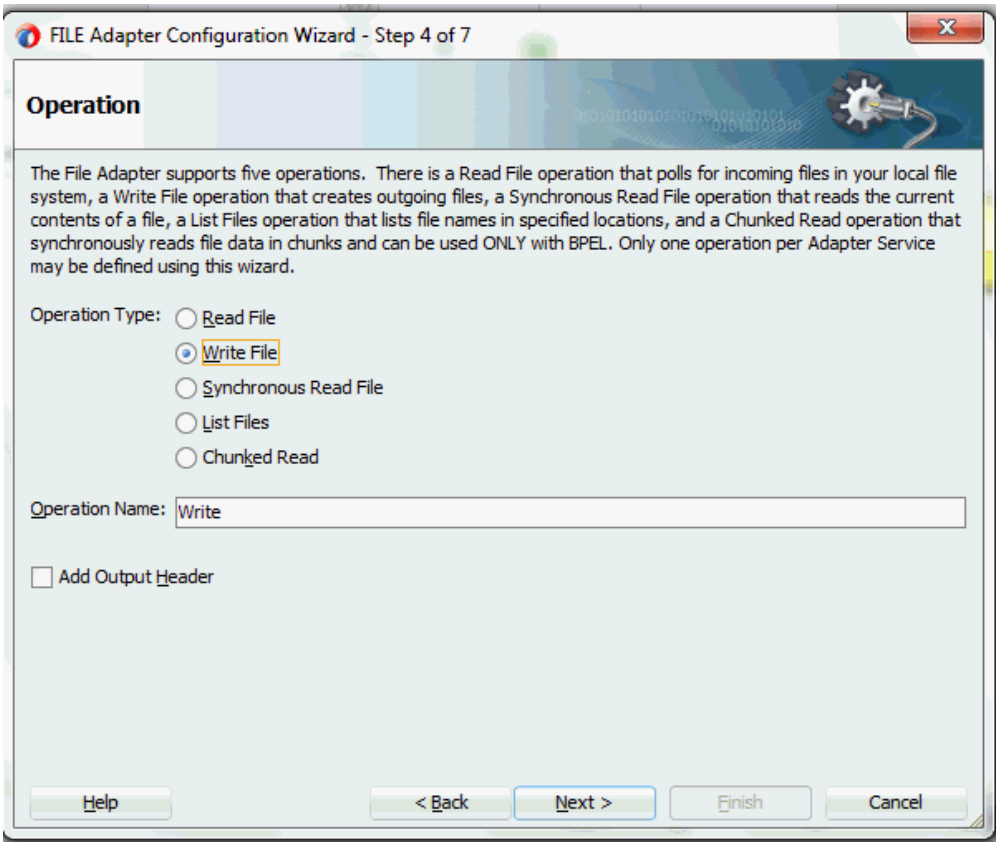
The File Server Connection page appears. Leave the default value unchanged for the File Server JNDI Name field.

Click **Next**



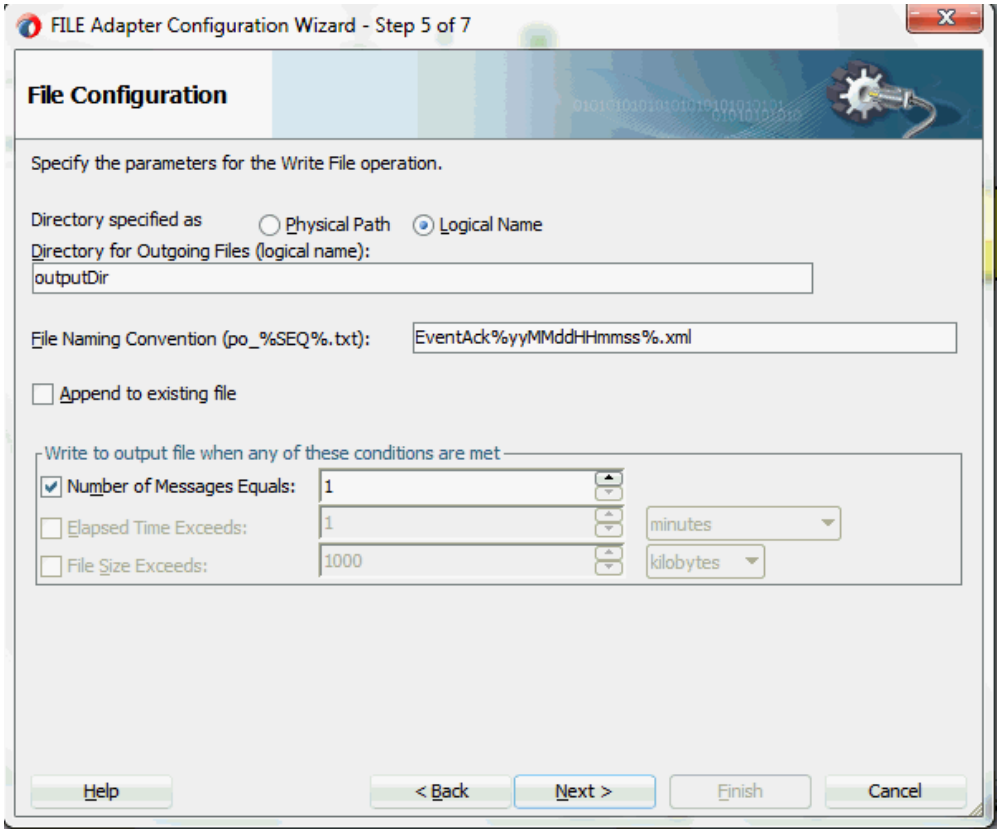
3. In the Operation page, specify the operation type. For example, select the **Write File** radio button. This automatically populates the **Operation Name** field.

Specifying the Operation



4. Click **Next** to access the File Configuration page.

Configuring the Output File



5. For the **Directory specified as** field, select the **Logical Name** radio button. Enter `outputDir` as the **Directory for Outgoing Files (logical name)** and specify a naming convention for the output file, such as `EventAck%yyMMddHHmmss%.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**.

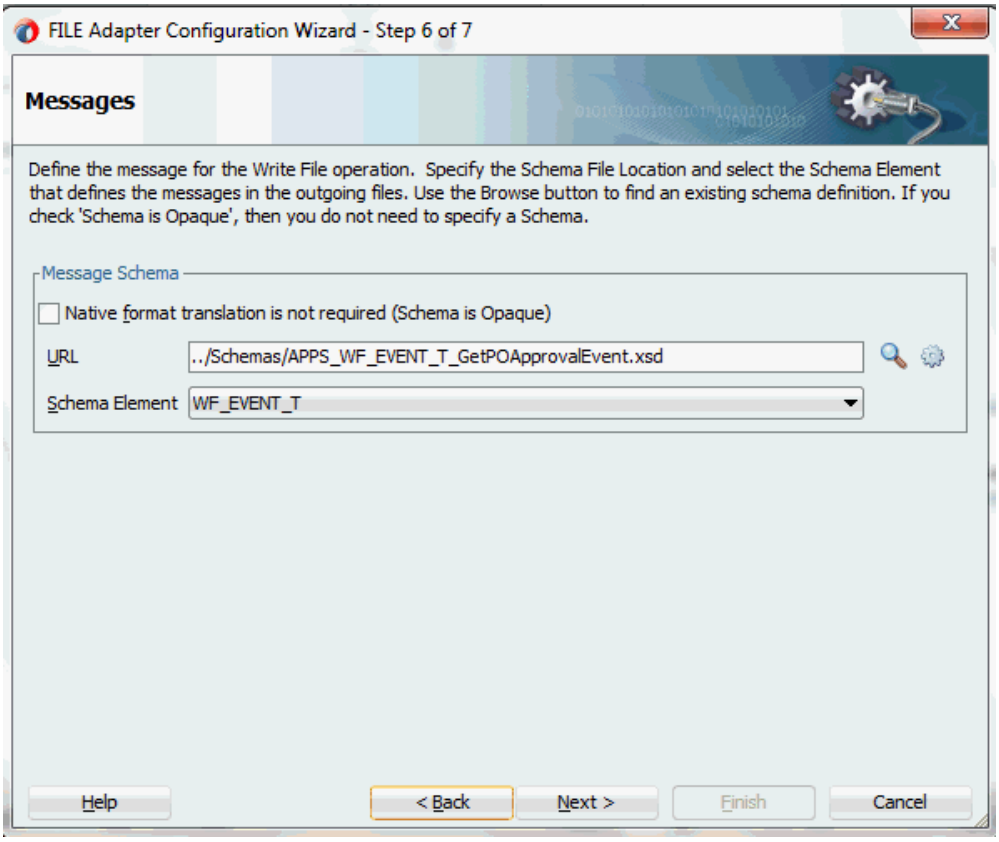
6. Click **Next**, and the Messages page appears. For the output file to be written, you must provide a schema.

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

8. Expand the node by clicking **Project Schema Files > WF_EVENT_T_GetPOApprovalEvent.xsd**. Select **WF_EVENT_T** as the element and click **OK**.

The selected schema information will be automatically populated in the URL and Schema Element fields.

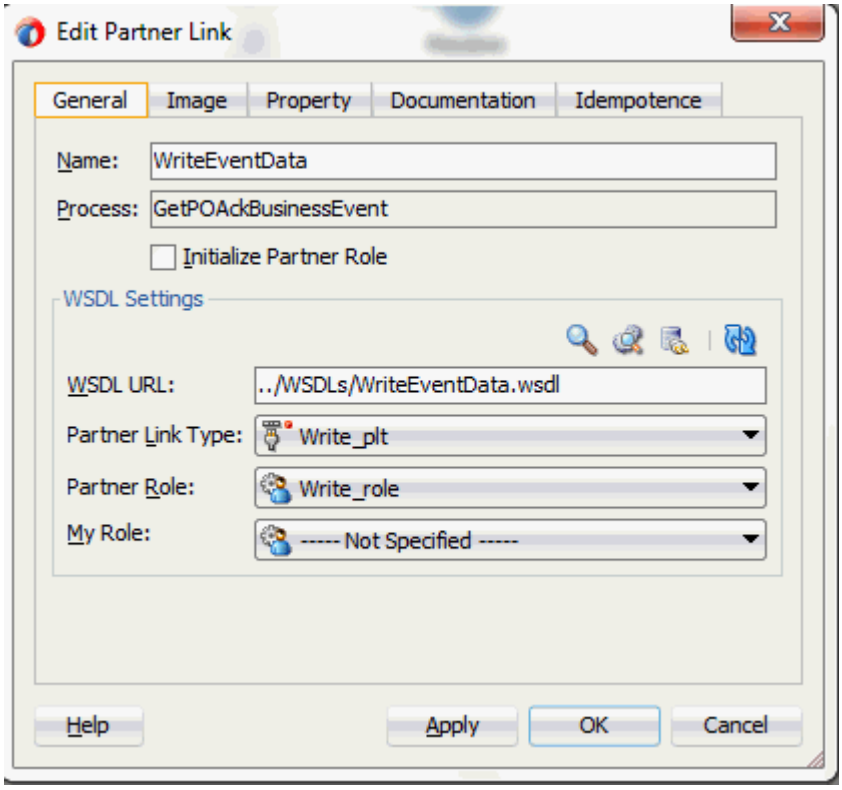
Populating the Selected Message Schema



9. Click **Next** and then **Finish**. This completes the configuration and creates the partner link with the required WSDL settings for the File Adapter service. The wizard also generates the `WriteEventData.wsdl` WSDL file corresponding to the partner link.

   The `WriteEventData` Partner Link appears in the BPEL process diagram. Double click the Partner Link service icon to open the Edit Partner Link dialog and view the details.

Completing the Partner Link Configuration



Configuring an Invoke Activity

After adding the File Adapter partner link, you need to configure an Invoke activity to associate it with the File Adapter link.

Through the Invoke activity, the business event information can be written to the XML file you specified as the output directory.

To configure an Invoke activity:

1. In Oracle JDeveloper BPEL Designer, expand the **BPEL Constructs** from the Component Palette. Drag and drop an **Invoke** activity from the **Web Service** section into the center swim lane of the process diagram after the **Receive** activity.

2. Link the Invoke activity to the `WriteEventData` File Adapter service.

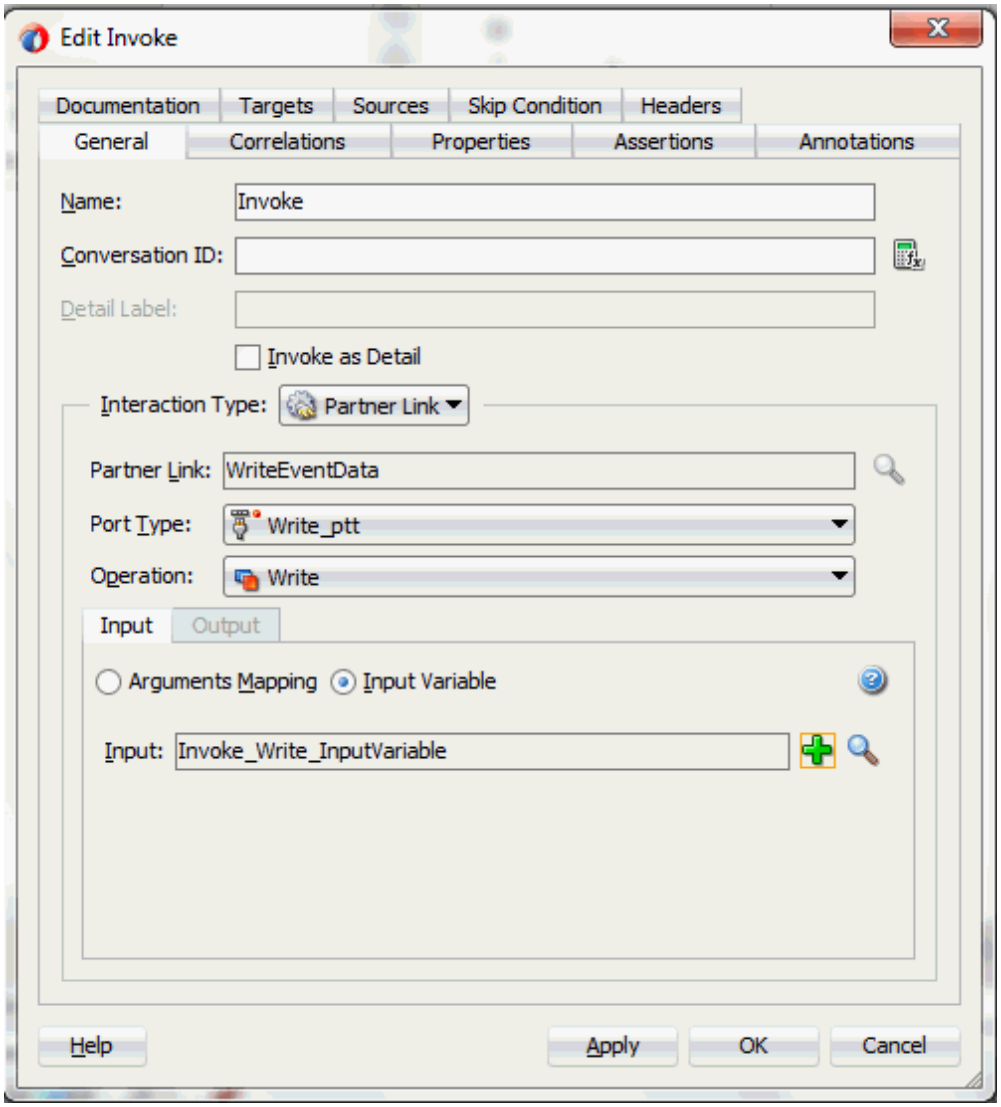
   The Invoke activity will send event data to the partner link. The Edit Invoke dialog appears.

3. Enter a name for the Invoke activity. In the Input tab, ensure the **Input Variable** radio button is selected. 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**, then enter a name for the variable. You can also accept the default name. Click **OK** to return to the Edit Invoke dialog.

Editing the Invoke Activity



Click **Apply** and then **OK** to finish configuring the Invoke activity.

Configuring an Assign Activity

Use the Assign activity to take the output from the Receive activity and to provide input to the invoke activity.

To configure an Assign activity:

1. In Oracle JDeveloper BPEL Designer, expand the **BPEL Constructs** from the Component Palette. Drag and drop the **Assign** activity from the **Basic Activities**

section into the center swim lane of the process diagram, between the **Receive** activity and the **Invoke** activity.

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

   Click the General tab to enter a name for the Assign activity. For example, `setEventData`.

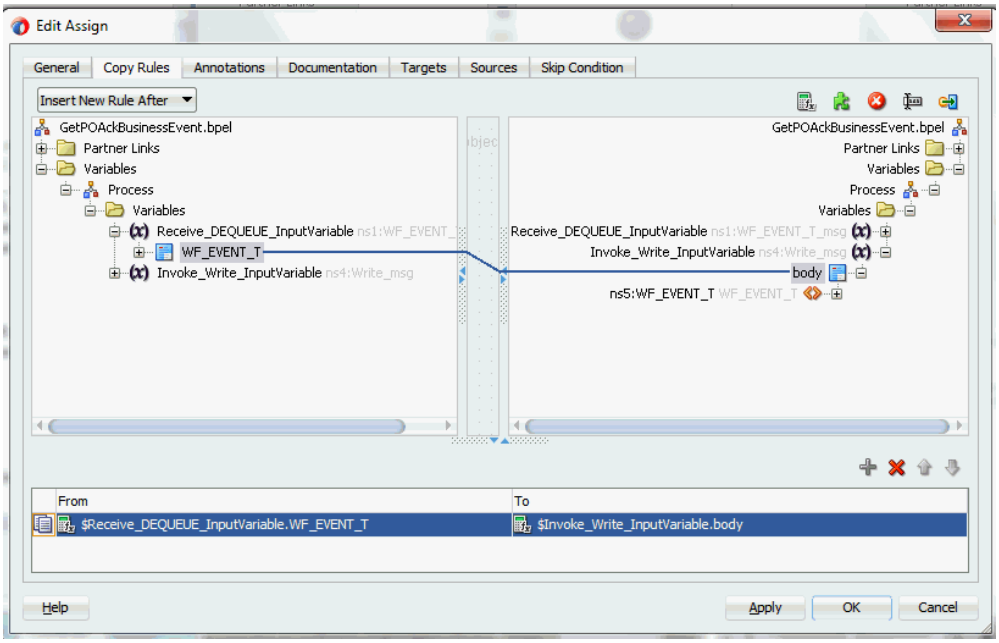
3. Select the Copy Rules tab and expand the target trees:

   - In the From navigation tree, navigate to **Variable > Process > Variables > Receive_DEQUEUE_InputVariable** and select **WF_EVENT_T**.

   - In the To navigation tree, navigate to **Variable > Process > Variables > Invoke_Write_InputVariable > body > ns3:WF_EVENT_T**.

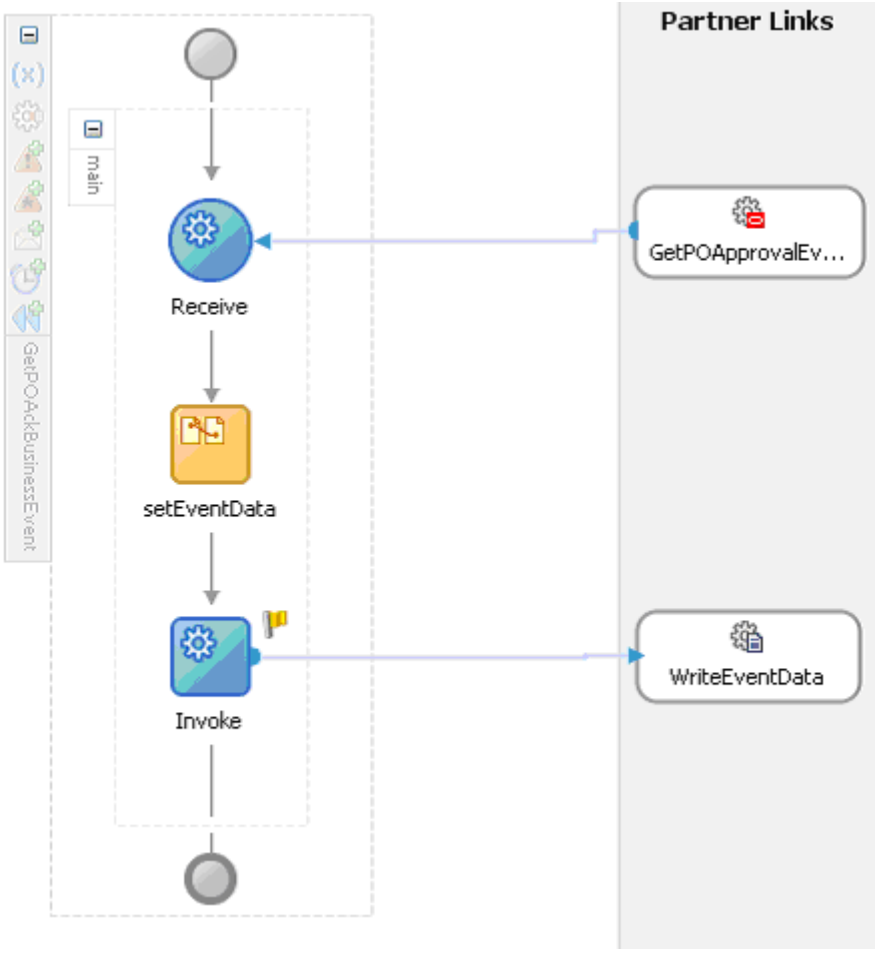
   Drag the source node (WF_EVENT_T) to connect to the target node (body) that you just specified. This creates a line that connects the source and target nodes. The copy rule is displayed in the From and To sections at the bottom of the Edit Assign dialog box.

Assigning Parameters



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

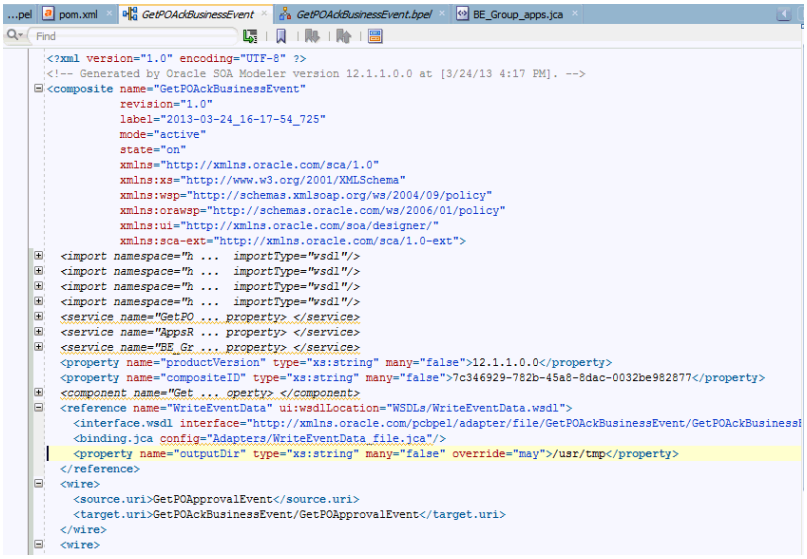
Completed Outbound Business Event BPEL Process Project



Click the `GetPOApprovalEvent` (composite.xml) to display the Oracle JDeveloper composite diagram:

> **Note:** Click the Source tab of GetPOApprovalEvent (composite.xml) to enter a value for the physical directory `outputDir` for the reference WriteEventData (such as `/usr/tmp`).
>
> ```
> <property name="outputDir" type="xs:string"
> many="false" override="may">/usr/tmp</property>
> ```

Specifying the Physical Directory for the Property



Oracle JDeveloper Composite Diagram

Run-Time Tasks for Outbound Business Events

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

1. Deploy the SOA Composite application with BPEL process, page 6-34.

2. Test the SOA Composite application with BPEL process, page 6-38.

Deploying the SOA Composite Application with BPEL Process

To invoke the service (GetPOApprovalEvent) from the BPEL client contained in the SOA composite, the SOA composite needs to be deployed to the Oracle WebLogic managed server. This can be achieved using Oracle JDeveloper. Once the composite is

deployed, it can be tested from the Oracle Enterprise Manager Fusion Middleware Control Console.

Prerequisites

Before deploying the SOA composite with BPEL process using Oracle JDeveloper, you must have established the connectivity between the design-time environment and the run-time server. For more information, see Configuring the Data Source in Oracle WebLogic Server, page 11-1 and Creating an Application Server Connection, page 11-10.

> **Note:** If a local instance of the WebLogic Server is used, start the WebLogic Server by selecting **Run > Start Server Instance** from Oracle JDeveloper. Once the WebLogic Admin Server "DefaultServer" instance is successfully started, the <Server started in Running mode> and DefaultServer started message in the Running:DefaultServer and Messages logs should appear.

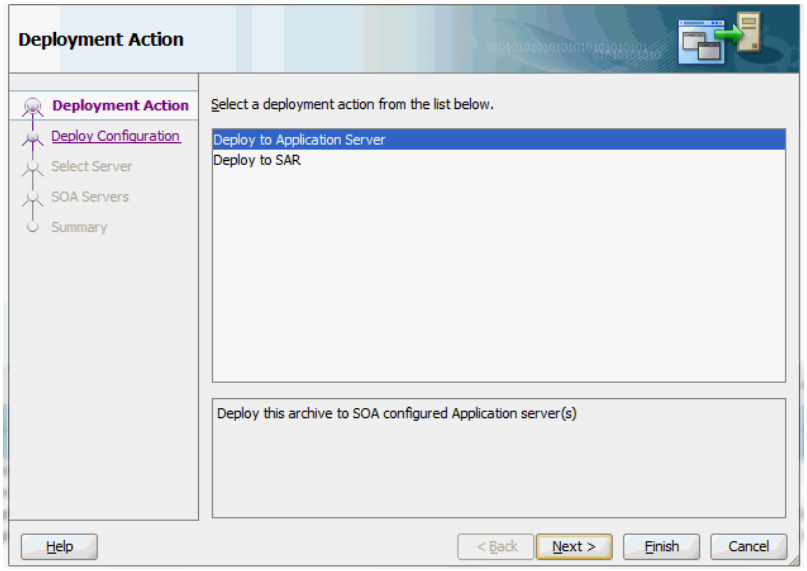
To deploy the SOA Composite application with BPEL process:

1. In the Applications Navigator of JDeveloper BPEL Designer, select your SOA Composite project name (such as **GetPOAckBusinessEvent**).

2. Right-click the project name, and then select **Deploy > [project name] > [serverConnection]** from the menu that appears.

   For example, you can select **Deploy > GetPOAckBusinessEvent > soa-server1** to deploy the process if you have the connection set up appropriately.

   > **Note:** If this is the first time to set up server connection, then the Deployment Action window appears. Select 'Deploy to Application Server' and click **Next**.

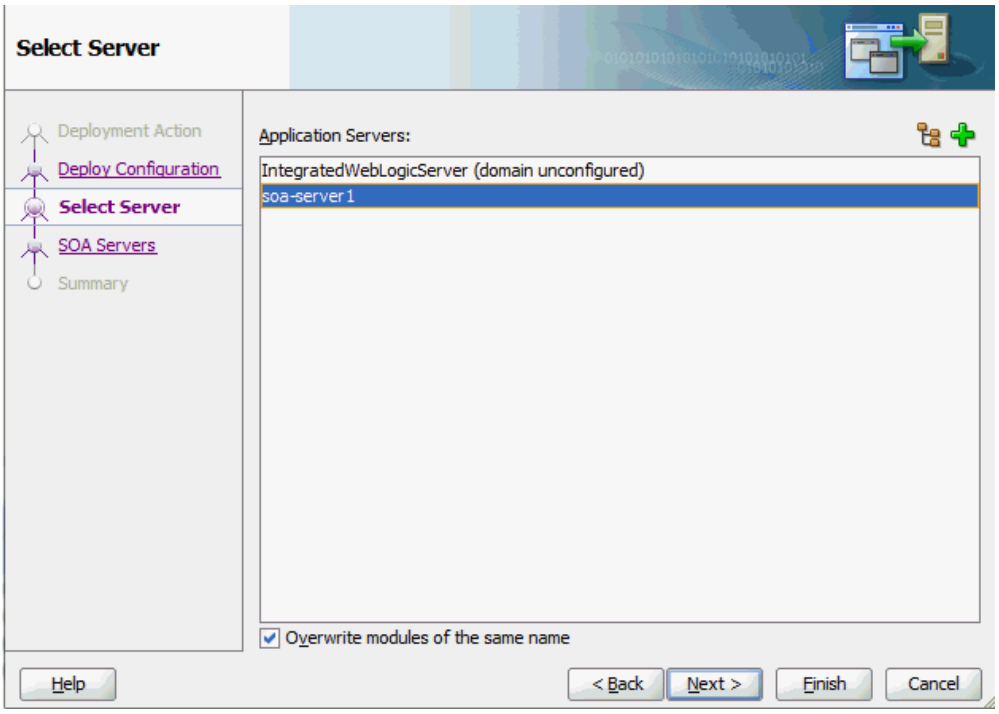


In the Deploy Configuration window, ensure the following information is selected before clicking **Next** to add a new application server:

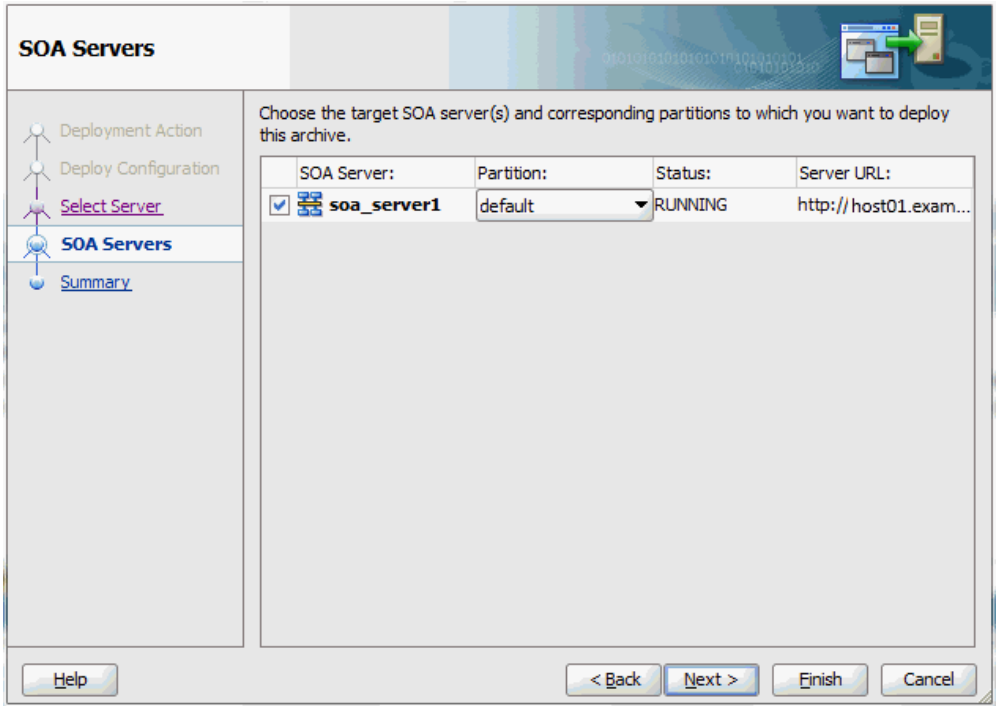
- New Revision ID: 1.0
- Mark composite revision as default: Select this check box.
- Overwrite any existing composites with the same revision ID: Select this check box.

The steps to create a new Oracle WebLogic Server connection from JDeveloper are covered inCreating an Application Server Connection, page 11-10.

3. In the Select Server page, select 'soa-server1' that you have established the server connection earlier. Click **Next**.



4. In the SOA Servers page, accept the default target SOA Server ('soa-server1') selection.



Click **Next** and **Finish**.

If you are deploying the composite for the first time from your Oracle JDeveloper session, the Authorization Request window appears. Enter username and password information specified during Oracle SOA Suite installation. Click **OK**.

5. Deployment processing starts. Monitor deployment progress and check for successful compilation in the SOA - Log window as well as in the Deployment - Log window.

Testing the SOA Composite Application with BPEL Process

Once the SOA Composite application with BPEL process is deployed, you can manage and monitor the process from the Oracle Enterprise Manager Fusion Middleware Control Console. You can also test the process by manually initiating it.

To test the SOA Composite application with BPEL process:

1. Navigate to Oracle Enterprise Manager Fusion Middleware Control Console (`http://<servername>:<portnumber>/em`). The composite you deployed is displayed in the Applications Navigation tree.

2. Enter username (such as `weblogic`) and password and click **Login** to log in to a farm.

   You may need to select an appropriate target instance farm if there are multiple target Oracle Enterprise Manager Fusion Middleware Control Console farms.

3. From the Farm base domain, expand the **SOA > soa-infra > soa-infra (soa_server1) > default** to navigate through the SOA Infrastructure home page and menu to access your deployed SOA Composite applications running in the SOA Infrastructure for that managed server.

   > **Note:** The Farm menu always displays at the top of the navigator. When you expand the SOA folder in the navigator and click the links displayed beneath it, the SOA Infrastructure menu becomes available at the top of the page.

4. Log on to Oracle E-Business Suite 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 supplier that has been defined.

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

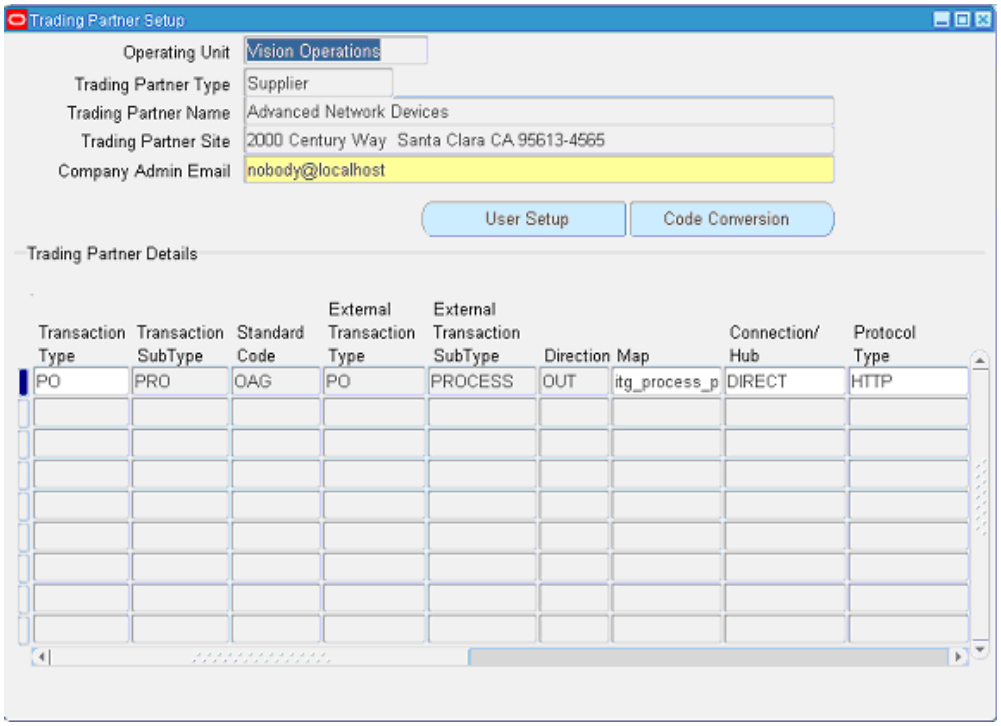
6. Enter the header values on the Trading Partner Setup form as follows:

   - Trading Partner Type: Supplier
   - Trading Partner Name: Advanced Network Devices

- Trading Partner Site: Enter a trading partner site information, such as 2000 Century Way, Santa Clara, CA 95613-4565.
- Company Admin Email: Enter a valid e-mail address.

7. Enter the following trading partner details:
   - Transaction Type: PO
   - 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: 'operation'
   - Password: Enter the associated password twice.
   - Protocol Address: 'http://appsadapter.sample.com'
   - Source Trading partner location code: STPLC

Trading Partner Setup Form



Save the trading partner details.

8. Switch responsibility by selecting the Purchasing, Vision Operations (USA) and select Purchase Order from the navigation menu.

   The Purchase Order forms is displayed.

9. Create a purchase order with the header values reflecting the trading partner you previously defined in the Purchase Order window:

   - Supplier: Enter a supplier information, such as 'Advanced Network Devices'.
   - Site: Select a site information, such as 'SANTA CLARA-ERS'.

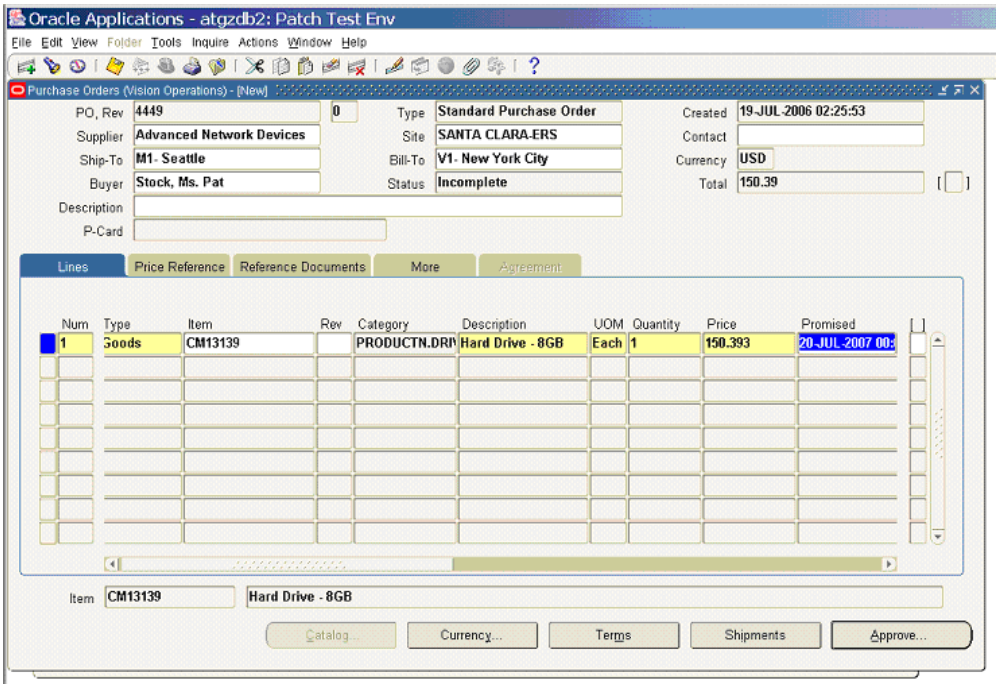
10. On the Lines tab, enter a data row with the following values:

    - Type: Goods
    - Item: CM13139
    - Quantity: 1
    - Description: Hard Drive - 8GB

- Promised: Enter any future date in the format of dd-mmm-yyyy (such as 23-JUN-2009).

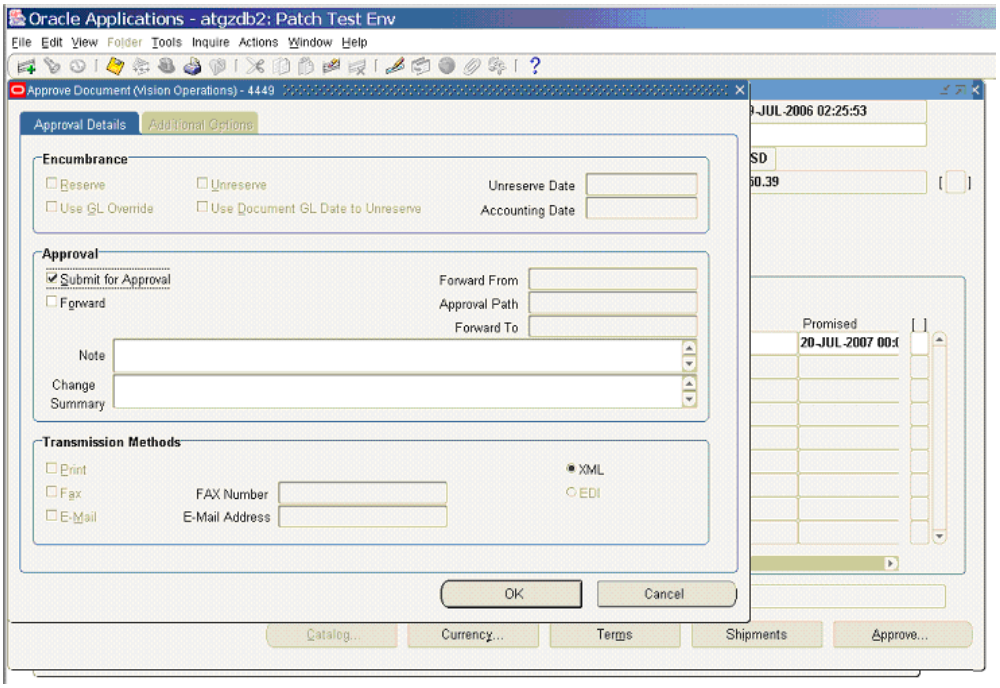
11. Save your purchase order. The status of the purchase order is 'Incomplete'.

Setting Up a Purchase Order



12. Click **Approve**. The Approve Document form appears.

Approving the Purchase Order



Click **OK** to confirm the approval.

> **Note:** Because the trading partner is set up and valid, the transmission method is automatically set to **XML**.

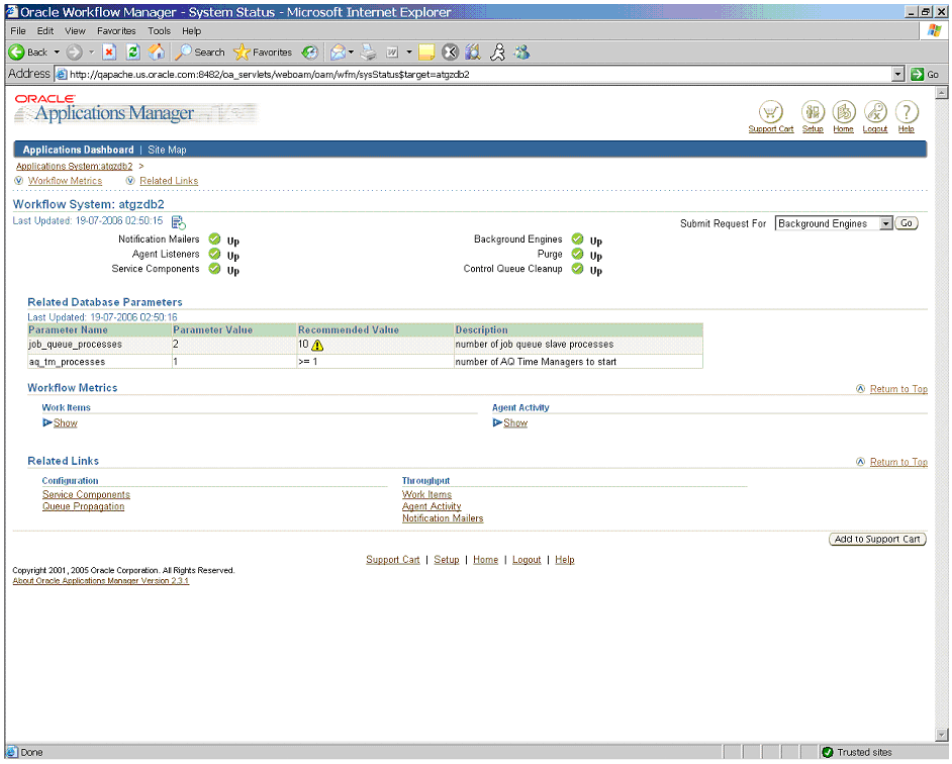
The status of the purchase order is now changed to 'Approved'. For future reference, record the value of the PO, Rev field (for example, the PO number 4449 in this case).

Once the purchase order is approved, the business event `oracle.apps.po.event.xmlpo` is raised.

13. Use the following steps to ensure that the **WF_Deferred Agent Listener** is running on the target database.

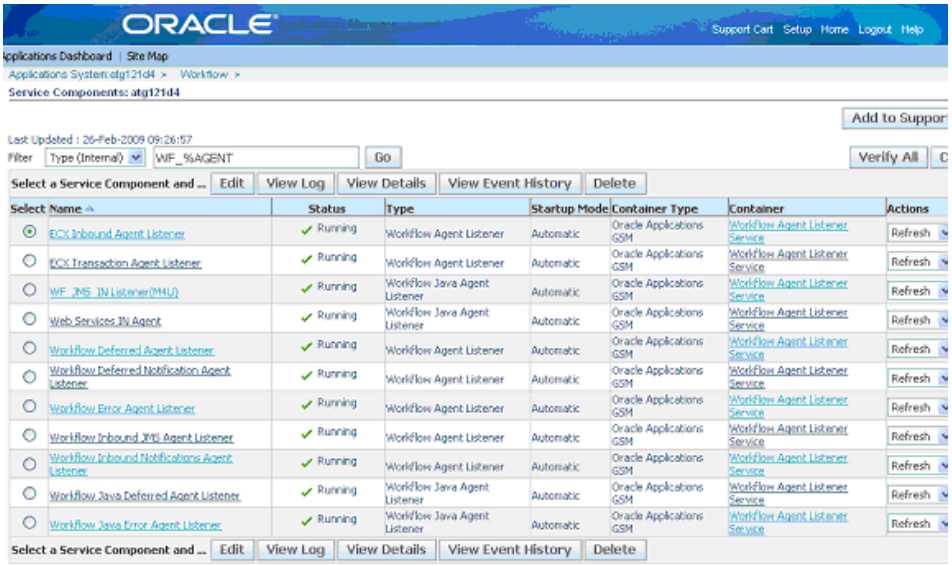
    1. Log on to Oracle E-Business Suite with the System Administrator responsibility.

    2. Select the Workflow Administrator Web Applications responsibility and choose **Oracle Applications Manager > Workflow Manager** from the menu.



3. 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

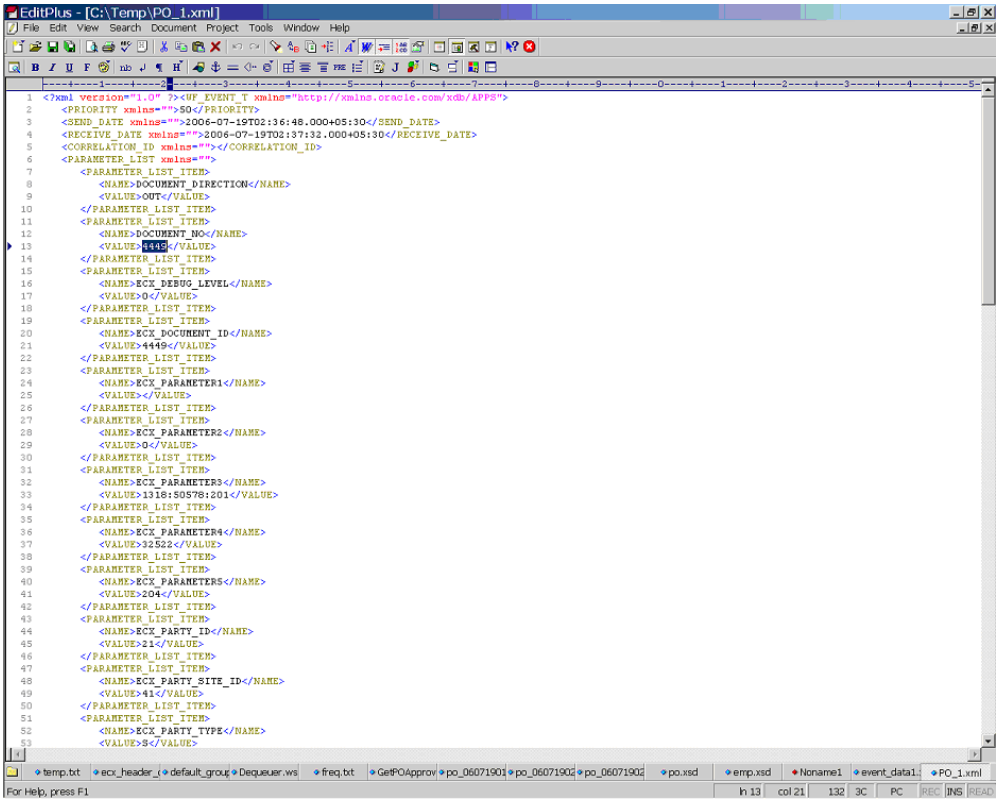


4. Confirm that the **Workflow Deferred Agent Listener** is in `Running` status.

14. At this time, your deployed BPEL process contained in a SOA Composite is listening for `oracle.apps.po.event.xmlpo` business event. Please allow 2 to 3 minutes for the BPEL process to activate after the event is raised.

15. Go to the directory, for example *outputDir* (typically under `c:\temp`) you specified for the write operation. Open the output file (such as `EventAck%yyMMddHHmmss.xml`), and confirm that the order number is same as that of the approved purchase order.

Confirming the Output Order Number



Troubleshooting

If you experience problems with your Business Event System integration, you can check the following troubleshooting steps:

- Confirm that Workflow Deferred Agent Listener is up and running.
- Ensure that business events are raised after the Composite is deployed.
- If the partner link of Oracle E-Business Suite Adapter is created on one instance of Oracle E-Business Suite and deployed on another, ensure the following on the target database:
  - WF_BPEL_Q queue exists
  - A custom subscription for the business event being raised is present

    Note: The above two can be easily re-created by running the `XX_BPEL_WFEVENT_<parnerlink_name>.sql` (located in the

project folder) on the target Oracle E-Business Suite database.

- Enable logging for Adapter to see if the issue is on the middleware side. For information on enabling logging for Oracle E-Business Suite Adapter, see Enabling Logging for Adapters, page 4-125.

7

Using Concurrent Programs

This chapter covers the following topics:

- Overview of Concurrent Programs
- Design-Time Tasks for Concurrent Programs
- Creating a New SOA Composite Application with BPEL Project
- Adding Partner Links
- Adding Partner Links for File Adapter
- Configuring Invoke Activities
- Configuring Assign Activities
- Run-Time Tasks for Concurrent Programs
- Deploying the SOA Composite Application with BPEL Process
- Testing the Deployed SOA Composite Application with BPEL Process
- Verifying Records in Oracle E-Business Suite
- Troubleshooting

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.

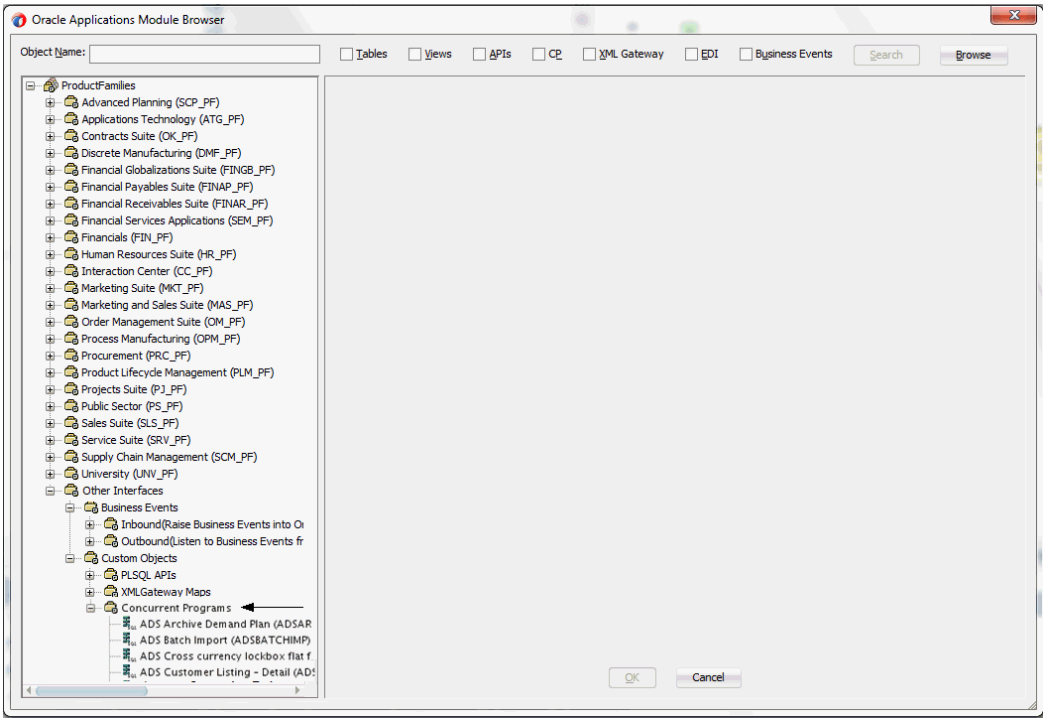
Accessing Concurrent Programs

Use Oracle E-Business Suite Module Browser to access concurrent programs either by browsing through product family or by performing a search to locate your desired concurrent programs.

Custom Concurrent programs

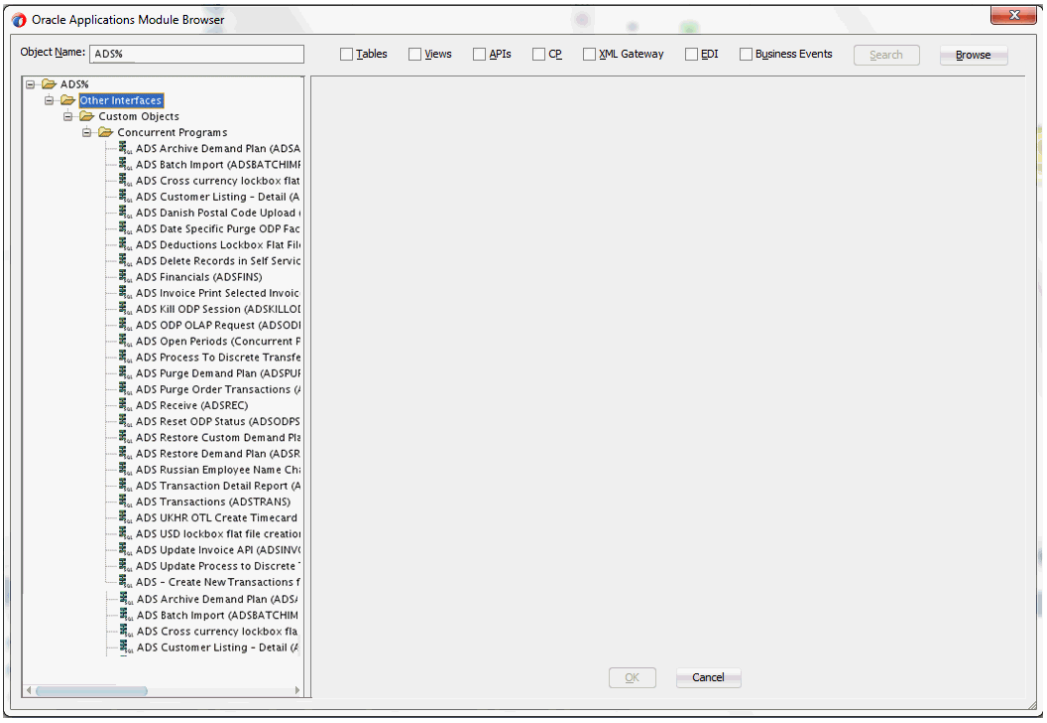
For custom concurrent programs, they are displayed in the Oracle E-Business Suite Module Browser under the **Other Interfaces > Custom Objects > Concurrent Programs** node.

Browsing and Selecting a Custom Concurrent Program



Alternatively, you can perform a search to locate your desired custom concurrent programs.

Searching for a Custom Concurrent Program



Design-Time Tasks for Concurrent Programs

This section describes how to configure the Oracle E-Business Suite Adapter to use concurrent programs.

SOA Composite Application with BPEL Process Scenario

Take Open Interface: Order Import concurrent program (`OEOIMP`) as an example. This concurrent program allows you to run the Order Import process to retrieve order details from the Order Management open interface tables to create sales orders.

When a request of creating an order is received, order details will be passed and inserted into Open Interface tables (`OE_HEADER_IFACE_ALL` and `OE_LINES_IFACE_ALL`). Once the insertion is completed, the concurrent program is invoked to import order data from Open Interface tables to Order Management base tables.

When the SOA Composite application with BPEL process has been successfully executed after deployment, you should find the order is created in Oracle E-Business Suite. The purchase order ID should be the same as the payload input value.

Prerequisites to Configuring Concurrent Programs

Populating Applications Context Header Variables

You need to populate certain variables in the BPEL PM in order to provide context

information for Oracle E-Business Suite. The context information is required for a transaction in order for an Oracle E-Business Suite user that has sufficient privileges to run the program.

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 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. 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 Invoke activity.

For more information about applications context, see Supporting for Normalized Message Properties, page 4-7.

SOA Composite Application with BPEL Process Creation Flow

Based on the process scenario, the following design-time tasks are explained in this chapter:

1. Create a new SOA Composite application with BPEL process, page 7-4
2. Add partner links, page 7-10
3. Add a partner link for File Adapter, page 7-27
4. Configure Invoke activities, page 7-39
5. Configure Assign activities, page 7-46

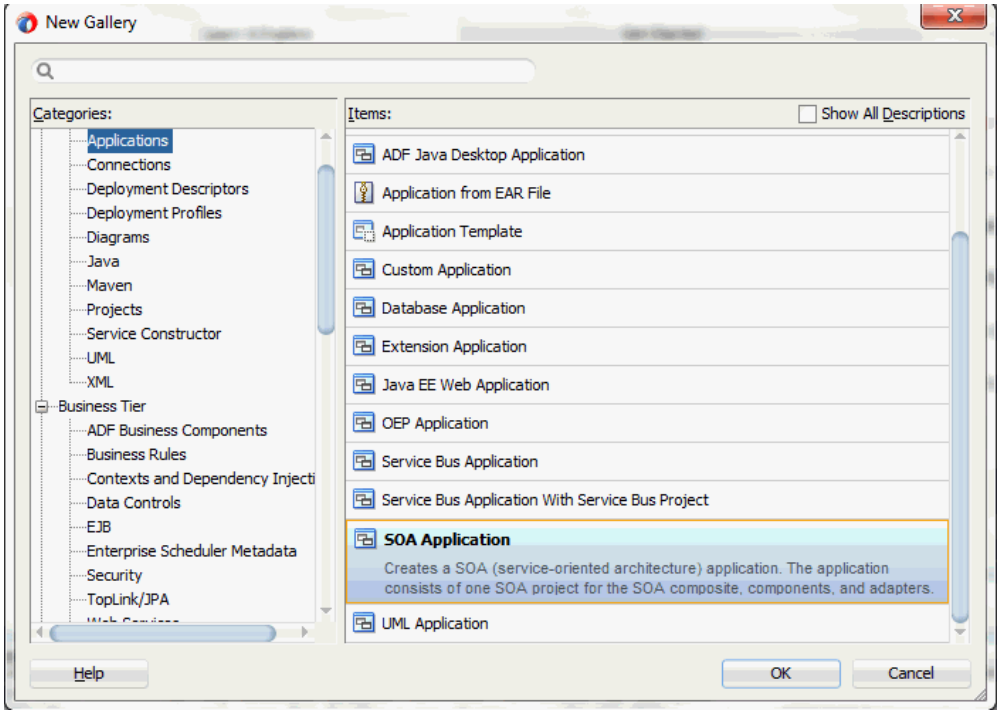
Creating a New SOA Composite Application with BPEL Project

To create a new SOA Composite application with BPEL project:

1. Open JDeveloper BPEL Designer.
2. Click **New Application** in the Application Navigator.

   The New Gallery page appears. Select **SOA Application** from the Items list.

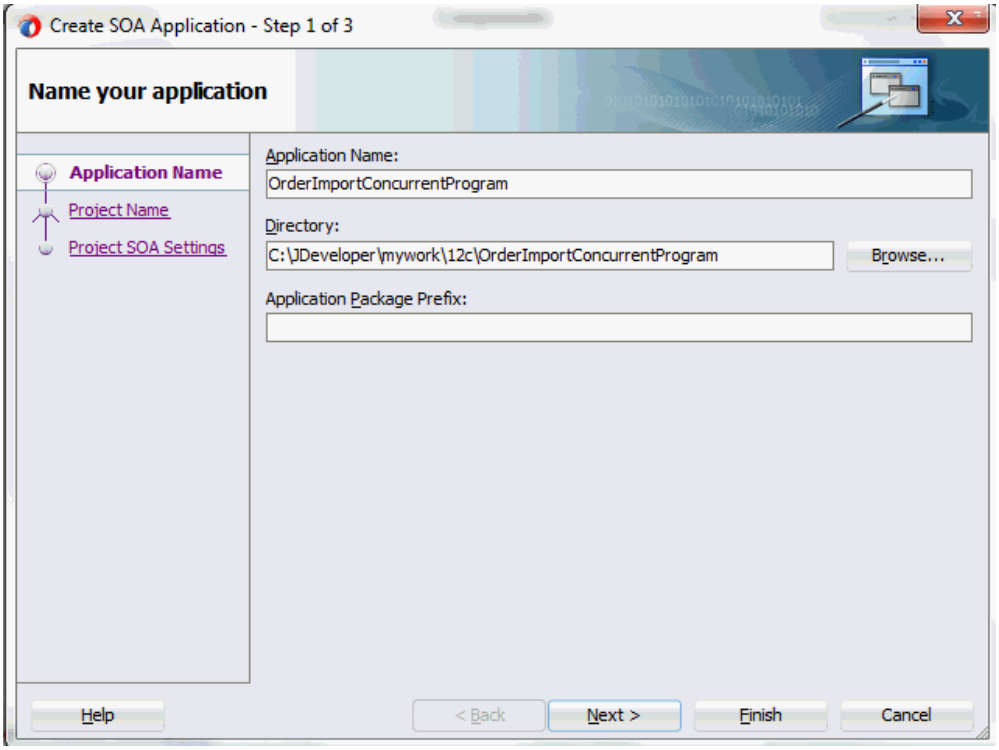
The New Gallery Page



Click **OK**.

3. The Create SOA Application - Name your application page is displayed.

The Create SOA Application - Name your application Page



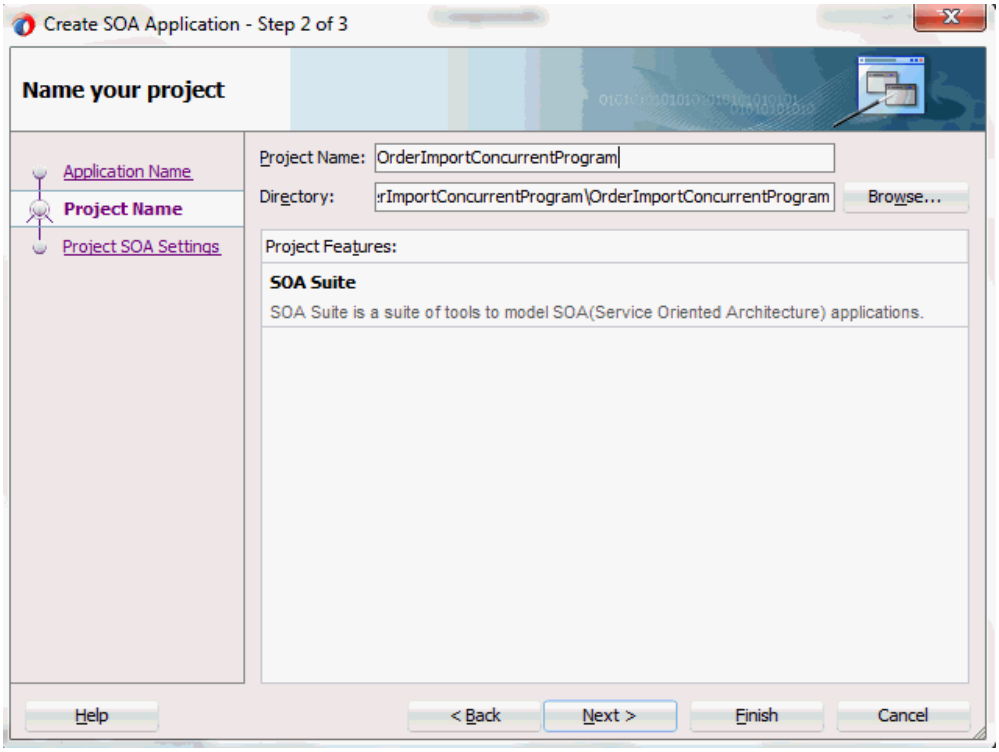
Enter an appropriate name for the application in the **Application Name** field.

4. Click **Next**. The Create SOA Application - Name your project page is displayed.

   Enter an appropriate name for the project in the **Project Name** field. For example, `OrderImportConcurrentProgram`.

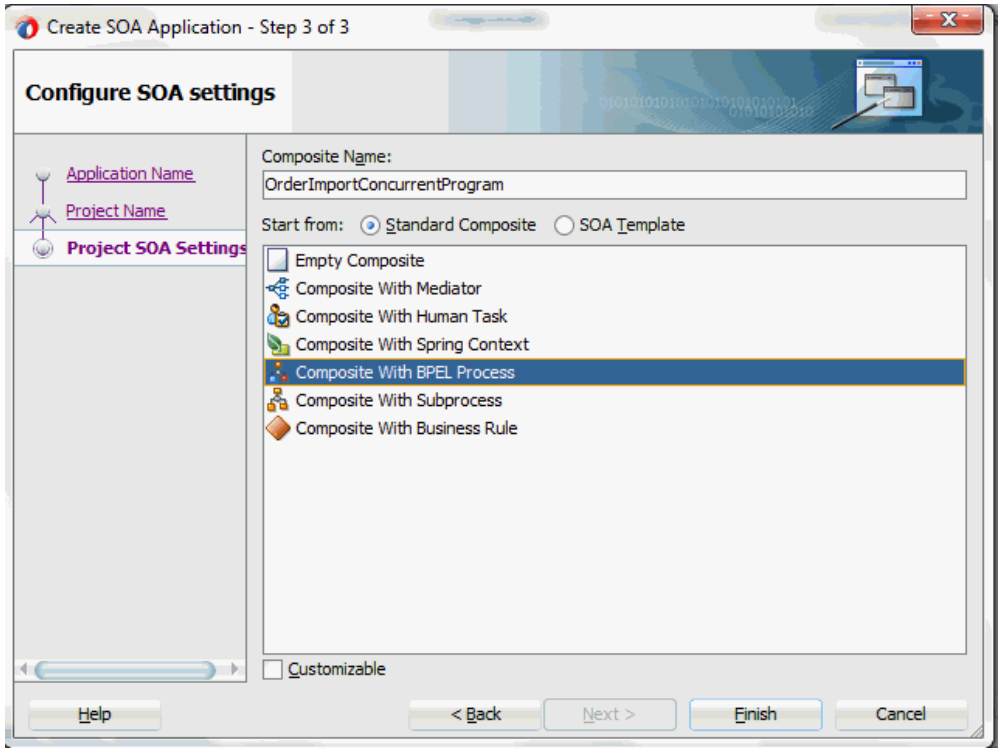
   Notice that **SOA Suite** is shown in the Project Features section.

The Create SOA Application - Name your project Page



5. Click **Next**. The Create SOA Application - Configure SOA settings page is displayed.

The Create SOA Application - Configure SOA settings Page

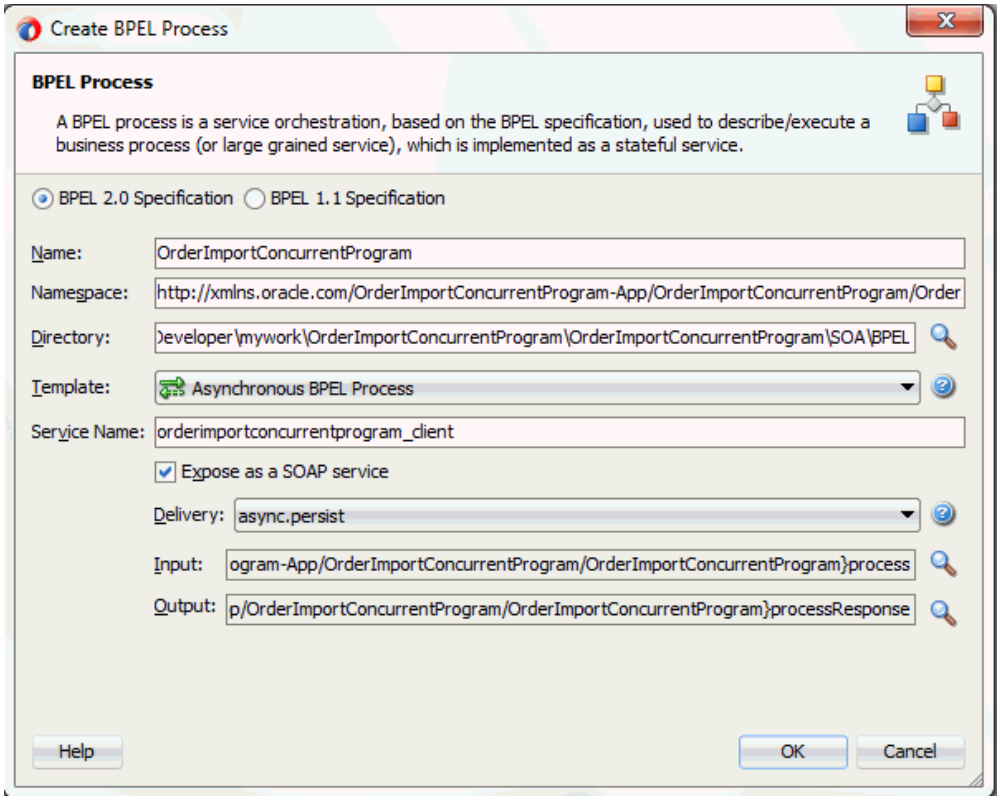


6. Leave the default **Standard Composite** selection unchanged for the Start from field.

   Select **Composite With BPEL Process** from the Standard Composite list, and then click **Finish**. You have created a new application, and a SOA project. This automatically creates a SOA Composite.

   The Create BPEL Process page is displayed.

The Create BPEL Process Page



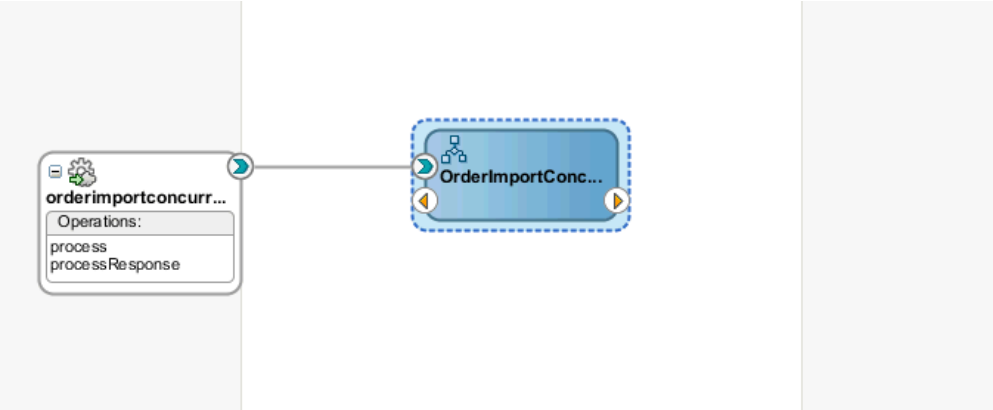
7. Leave the default **BPEL 2.0 Specification** selection unchanged. This creates a BPEL project that supports the BPEL 2.0 specification.

   Enter an appropriate name for the BPEL process in the **Name** field. For example, `OrderImportConcurrentProgram`.

   Select **Asynchronous BPEL Process** in the **Template** field. Click **OK**.

   An asynchronous BPEL process is created with the Receive and Reply activities. The required source files including bpel and wsdl, using the name you specified (for example, `OrderImportConcurrentProgram.bpel` and `OrderImportConcurrentProgram.wsdl`) and `OrderImportConcurrentProgram` (composite.xml) are also generated.

8. Navigate to SOA folder and click `OrderImportConcurrentProgram` (composite.xml) to view the composite diagram.



Double click on the `OrderImportConcurrentProgram` component to open the BPEL process.

Adding Partner Links

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.

Based on the BPEL process scenario discussed earlier, the following two partner links need to be configured:

- Add the first partner link `InsertOrder` to insert order details into selected Open Interface tables
- Add the second partner link `ImportOrderCP` to import purchase order data from the Open Interface tables to Order Management base tables

To add the first partner link:

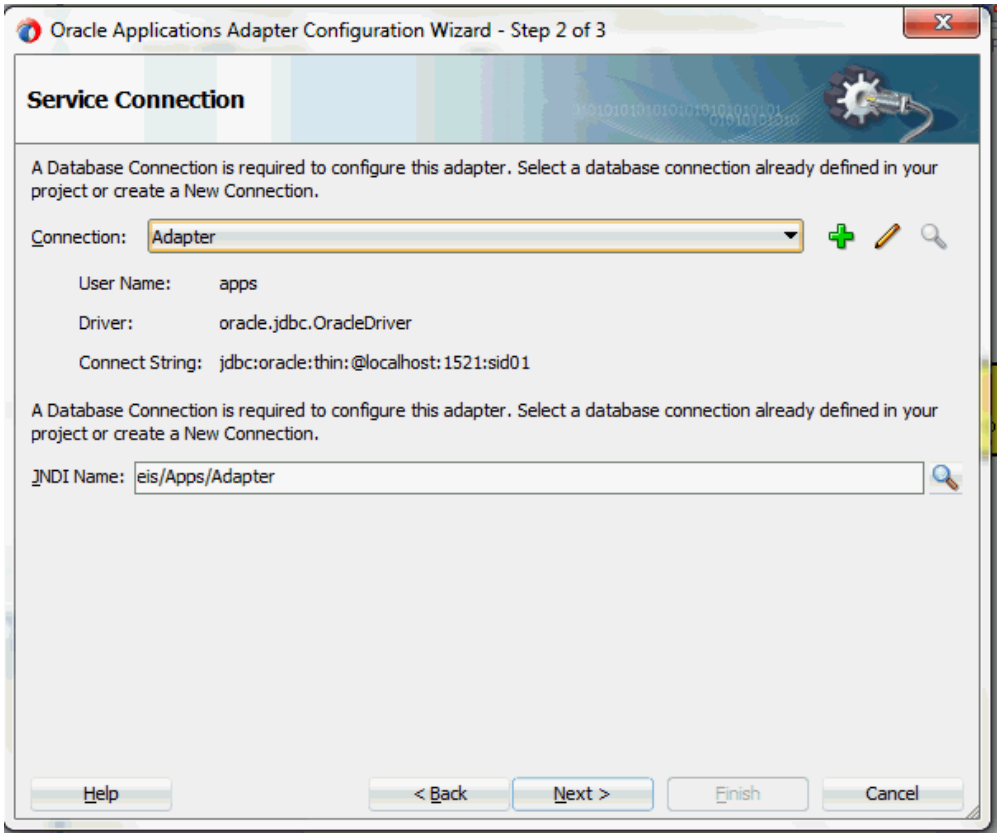
1. Click **BPEL Services** in the Component palette.

   Drag and drop **Oracle E-Business Suite** (formerly known as **Oracle Applications**) from the **BPEL Services** list into the right Partner Link swim lane of the process diagram. The Oracle E-Business Suite Reference page appears.

   Enter a service name in the **Name** field. For example, `InsertOrder`.

2. Click **Next**. The Service Connection page appears.

Specifying a Database Service Connection



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 **Create a New Database Connection** icon.

     How to define a new database connection, see Create a New Database Connection, page 5-14.

   - 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 Oracle E-Business Suite Adapter 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 Fusion Middleware User's Guide for Technology Adapters.*

4. Click **Next** in the Service Connection dialog box. You can add a concurrent program by browsing through the list of concurrent programs available in Oracle E-Business Suite.

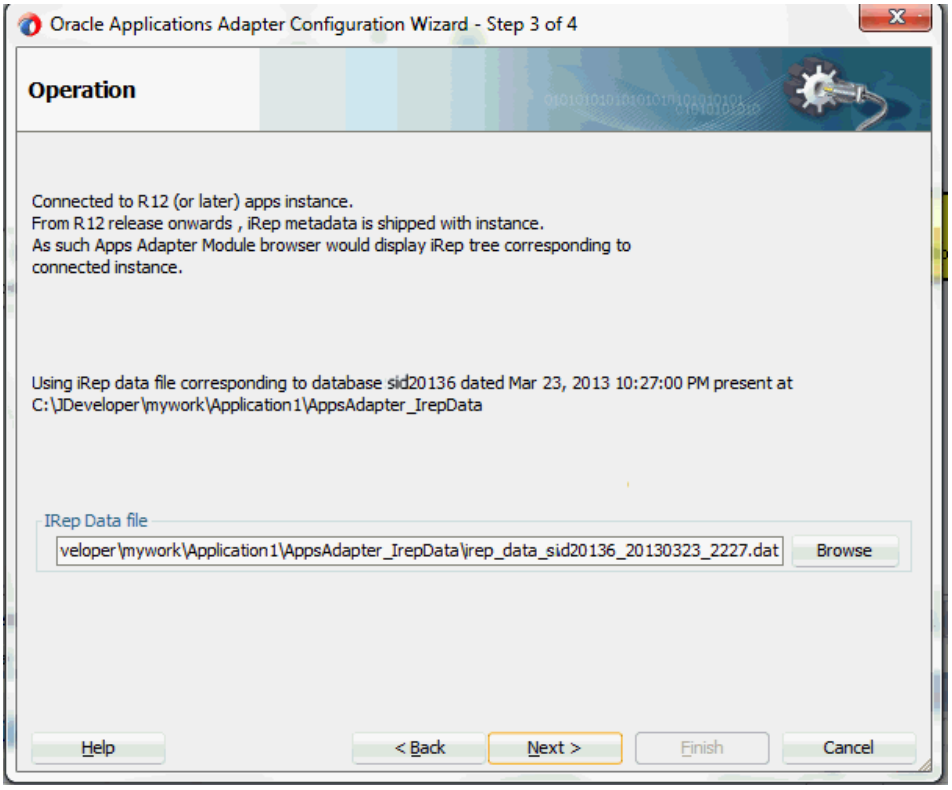
   If you are connecting to Oracle E-Business Suite Release 12, then the **IREP File not present** dialog box appears indicating that Adapter could not find the Oracle Integration Repository data file corresponding to the database you are connecting in your workspace. Absence of the data file would make 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 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.

     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



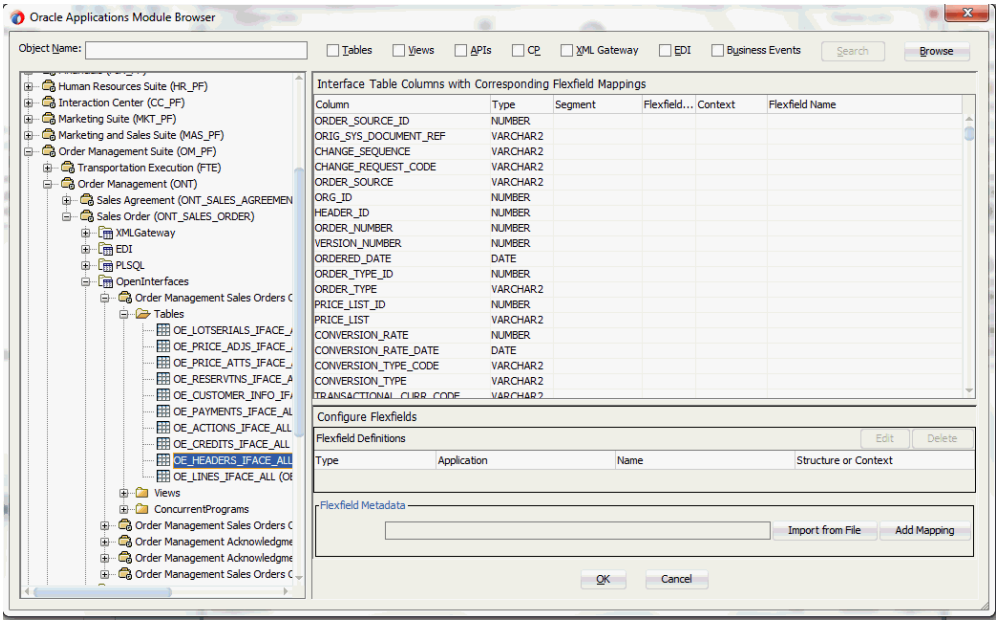
- Click **No** to query the Integration Repository data file from the live database you are connecting to display the Integration Repository tree.

In this example, click **No**. Click **Next** in the Operation page to open the Oracle E-Business Suite Module Browser.

5. Oracle E-Business Suite Module Browser includes the various product families that are available in Oracle E-Business Suite. Expand the tree nodes to locate your desired interface. Concurrent programs can be found under the Open Interfaces 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`.

Selecting a Concurrent Program from the Module Browser



Note: If a selected interface table has pre-configured flexfield mappings available, a pop-up window appears allowing you to decide if you want to use these pre-configured mappings. Click **Yes** to display these mappings as part of the flexfield definitions in the Configure Flexfields region. You can modify them later if needed.

Click **No** to indicate that you will not use these mappings and the pop-up window disappears.

Configuring Flexfield Mapping Data for the Selected Open Interface Table

Please note that an open interface table is selected in this example as part of the partner link creation. Oracle E-Business Suite Adapter provides flexfield support for open interface tables. Once the table is selected in the module browser, you can either configure flexfield mappings for the table that has flexfields defined in Oracle E-Business Suite or use current data without further flexfield mapping configuration.

Perform either one of the following tasks in the Oracle E-Business Suite Module Browser:

- Use the current data without further configuration by clicking **OK** to continue.
- Configure flexfield mapping data by either using an existing flexfield mapping or creating a new mapping through a flexfield wizard if desired.

  For more information on flexfield support for open interface tables and how to

modify existing mapping configuration, see:

- Configuring Key Flexfield Mappings for an Open Interface Table, page 4-79
- Configuring Descriptive Flexfield Mappings for an Open Interface Table, page 4-85
- Flexfield Support for PL/SQL APIs and Open Interface Tables, page 4-27.

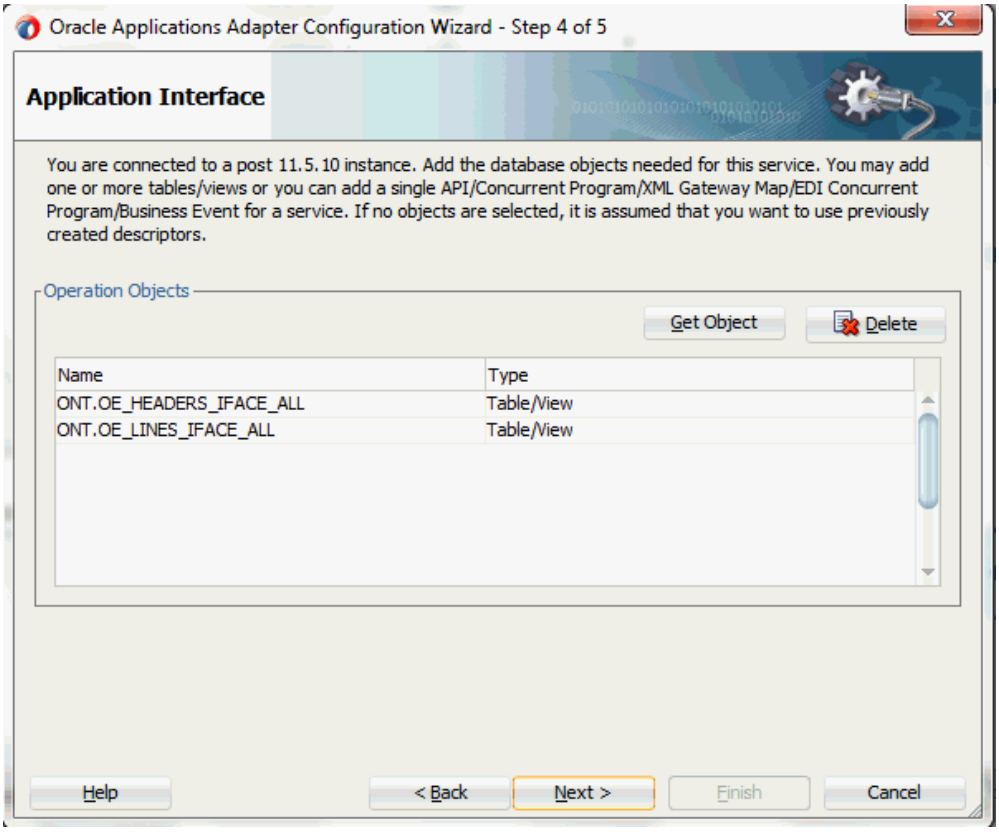
6. In this example, click **OK**. The Application Interface page appears with the selected open table.

Adapter Configuration Wizard - Application Interface Page

7. Click **Get Object** to open the 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* to select `OE_LINES_IFACE_ALL`.

   Click **OK**. The Application Interface page appears with the two selected tables.

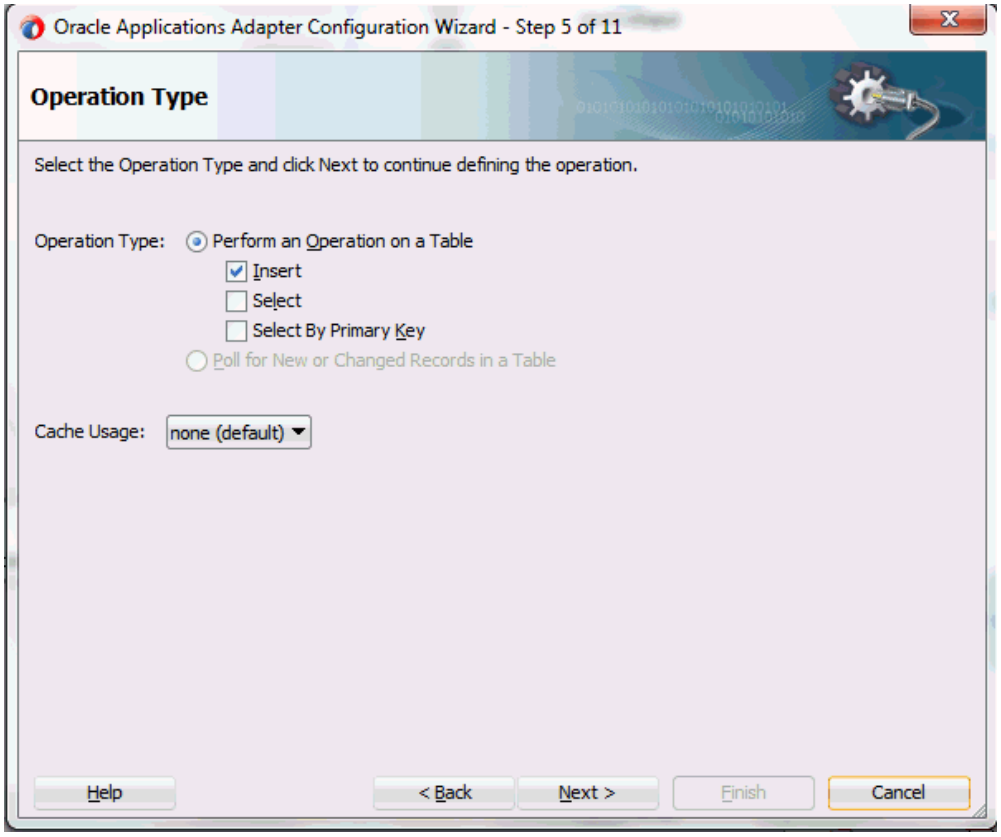
Adapter Configuration Wizard - Application Interface Page



Click **OK**.

8. Click **Next**. The Operation Type page is displayed.

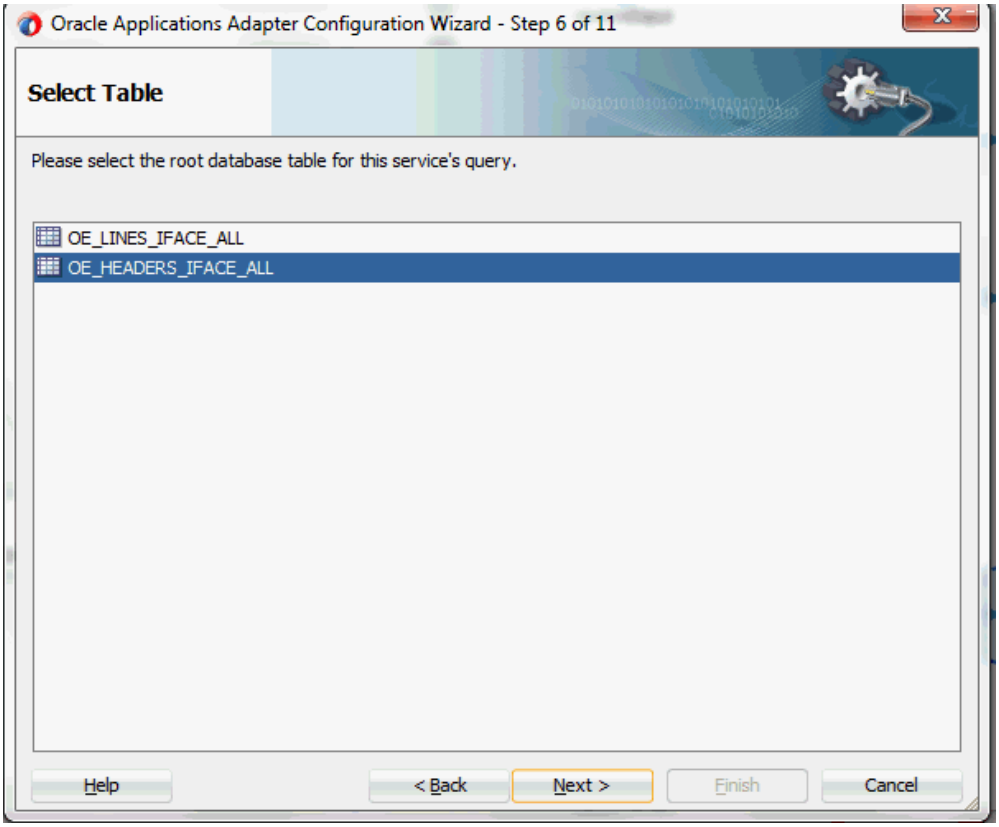
Adapter Configuration Wizard - Operation Type Page



Select the **Perform an Operation on a Table** radio button and then choose the **Insert** check box. Ensure that the **Select** check box is deselected.

9. Click **Next**. The Select Table page is displayed.

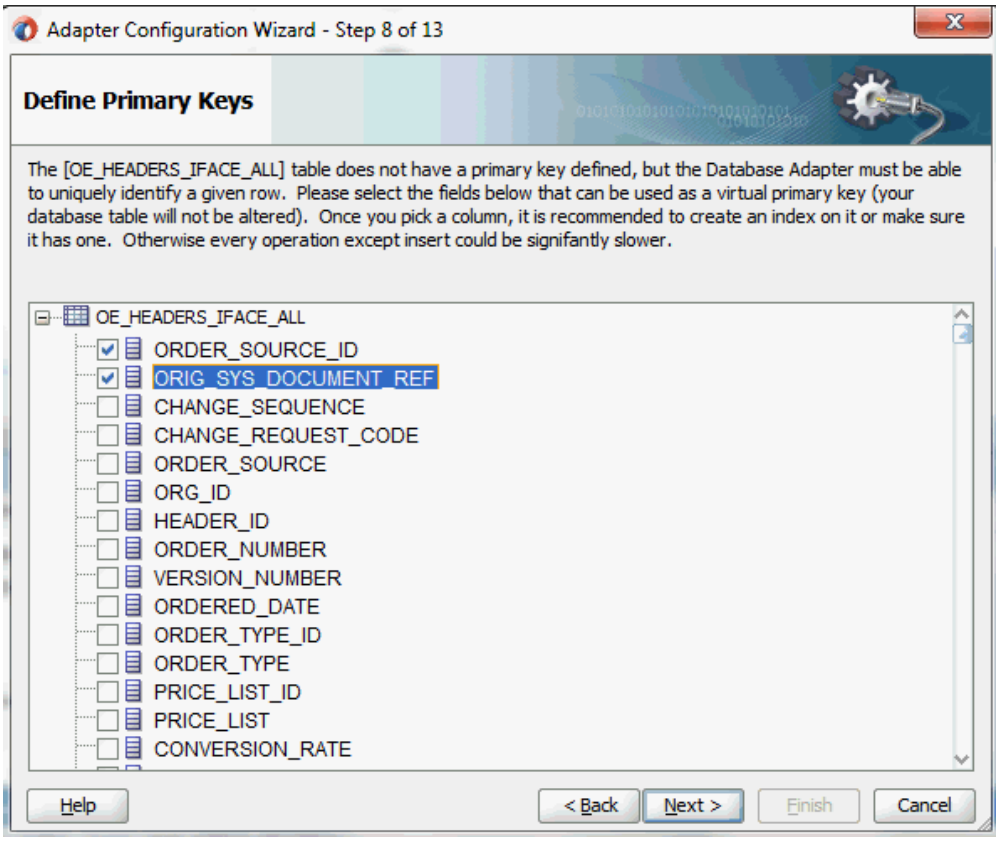
Adapter Configuration Wizard - Select Table Page



Select OE_HEADERS_IFACE_ALL as the root database table for this service's query.

10. Click **Next**. The Define Primary Keys page is displayed.

Adapter Configuration Wizard - Define Primary Keys Page for OE_HEADERS_IFACE_ALL



Select the following primary keys for the OE_HEADERS_IFACE_ALL table:

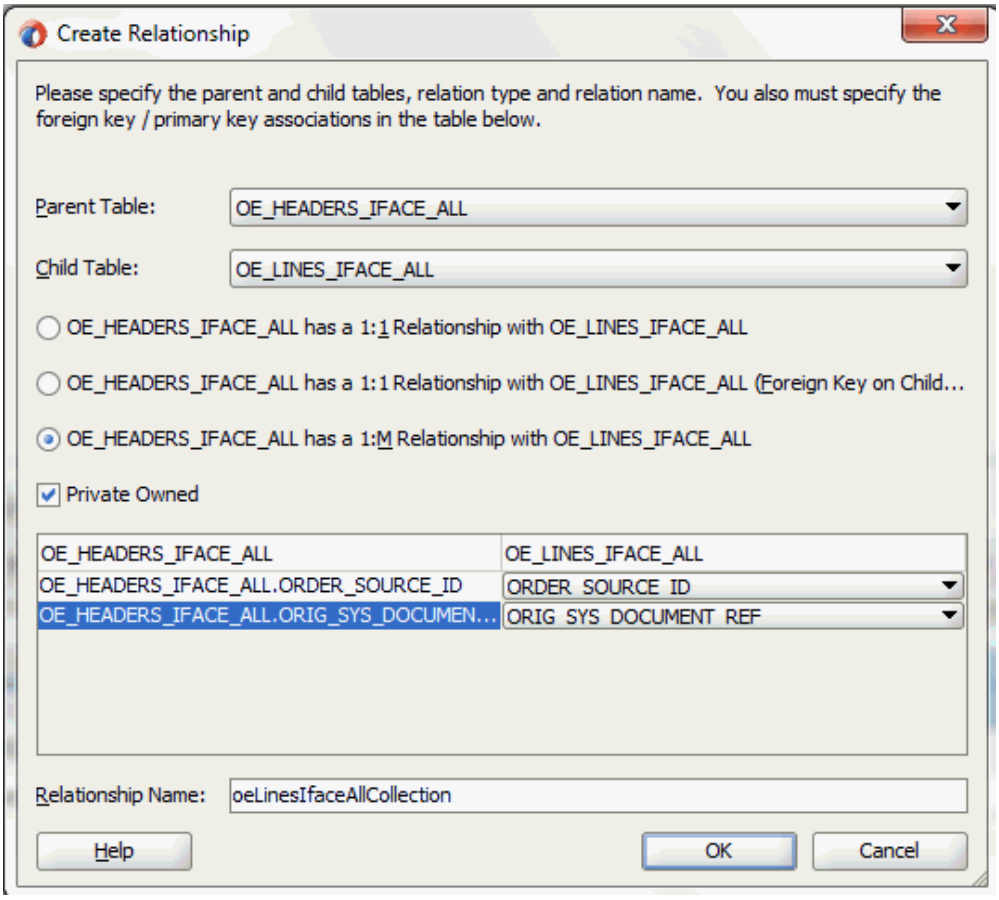
- ORDER_SOURCE_ID
- ORIG_SYS_DOCUMENT_REF

Click **Next**.

Select the same primary keys for the OE_LINES_IFACE_ALL table.

11. Click **Next**. The Relationships page appears. Click **Create** to open the Create Relationship dialog.

Defining Relationships



Enter the following information to define the relationship between the header and the detail table:

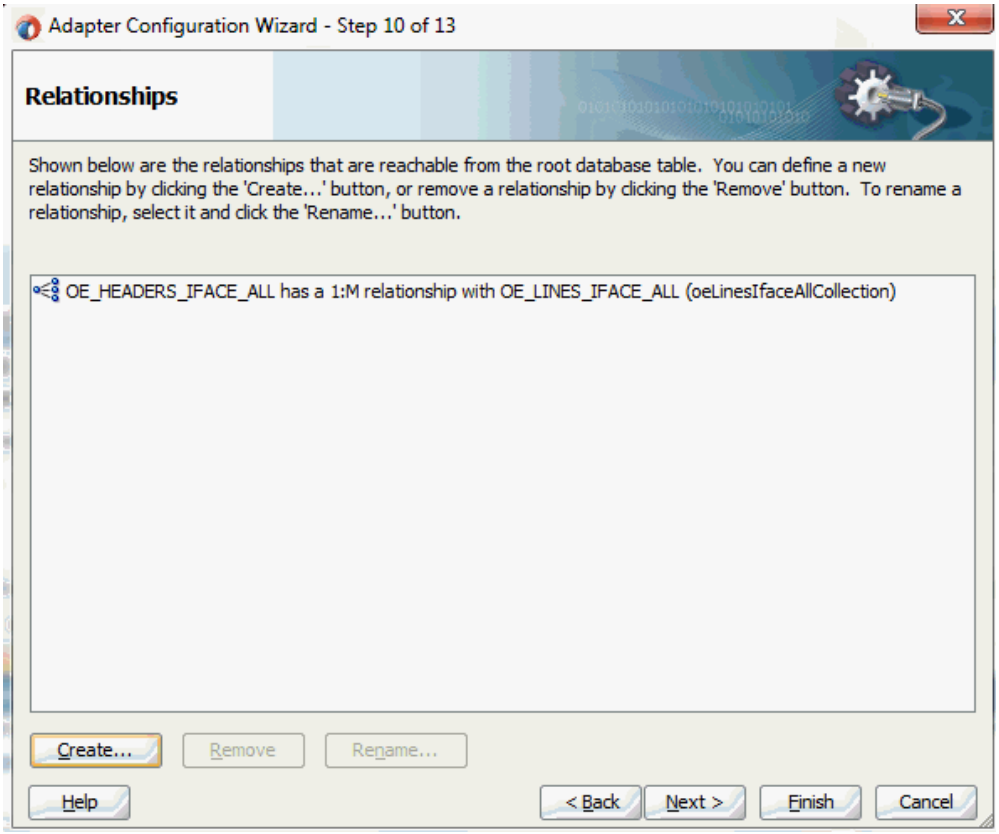
- Select the OE_HEADERS_IFACE_ALL as the parent table and OE_LINES_IFACE_ALL as the child table.
- Select the mapping type: OE_HEADERS_IFACE_ALL has a 1:M Relationship with OE_LINES_IFACE_ALL

> **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.

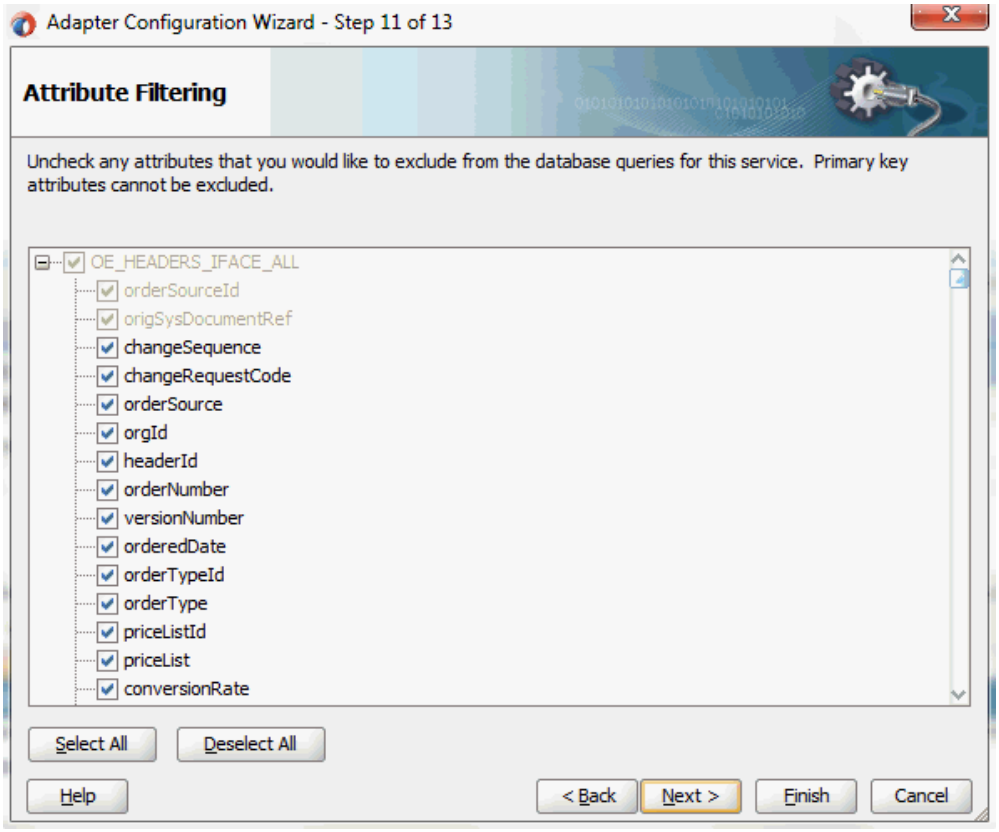
Click **OK**. The newly created relationship information appears in the Relationships page.

Adapter Configuration Wizard - Relationships Page



12. Click **Next**. The Attribute Filtering page appears. Leave default selections unchanged.

Adapter Configuration Wizard - Attribute Filtering Page



Click **Next**. The Advanced Options page appears.

Click **Next**.

13. Click **Finish**. The wizard generates the WSDL file corresponding to the selected interface. This WSDL file is now available for the partner link.

To add the second partner link:

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

   Drag and drop **Oracle E-Business Suite** (formerly known as **Oracle Applications**) from the **BPEL Services** list into the right Partner Link swim lane of the process diagram. The Oracle E-Business Suite Reference page appears.

   Enter a service name in the **Name** field. For example, `ImportOrderCP`.

2. 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 **Create a New Database Connection** icon.

  How to define a new database connection, see Create a New Database Connection, page 5-14.

- 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 Oracle E-Business Suite Adapter 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 Fusion Middleware User's Guide for Technology Adapters.*

4. Once you have completed creating a new connection for the service, you can add a concurrent program by browsing through the list available in Oracle E-Business Suite.

   Click **Next**.

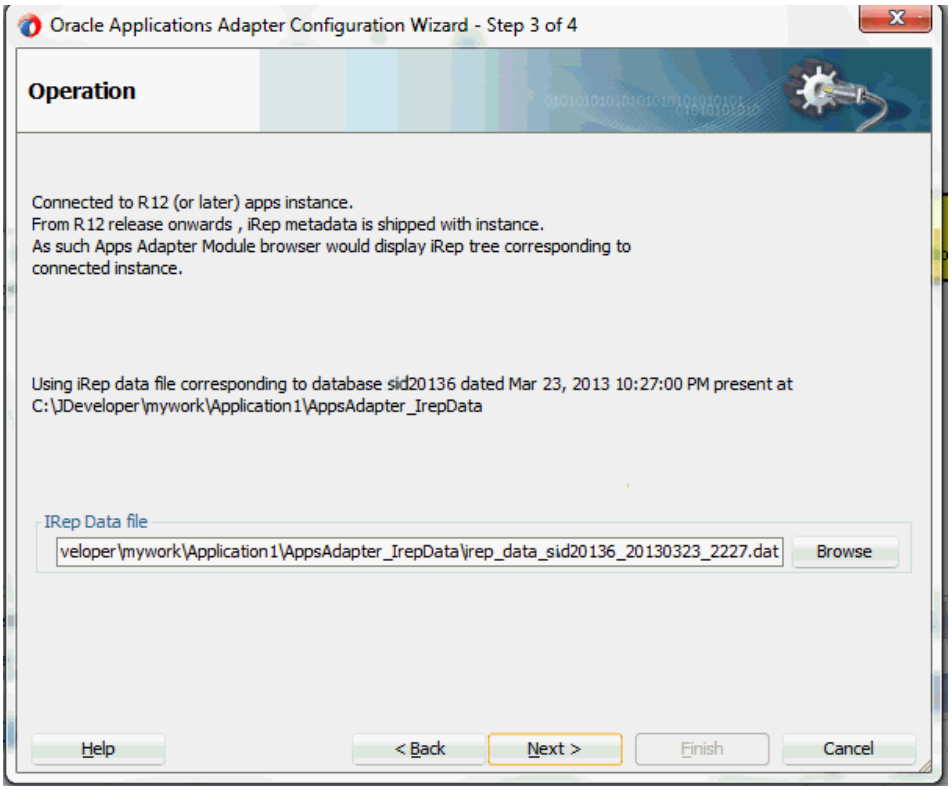
   If you are connecting to Oracle E-Business Suite Release 12, then the **IREP File not present** dialog appears indicating that Adapter could not find the Oracle Integration Repository data file corresponding to the database you are connecting to Oracle E-Business Suite in your workspace. Absence of the 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 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.

  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

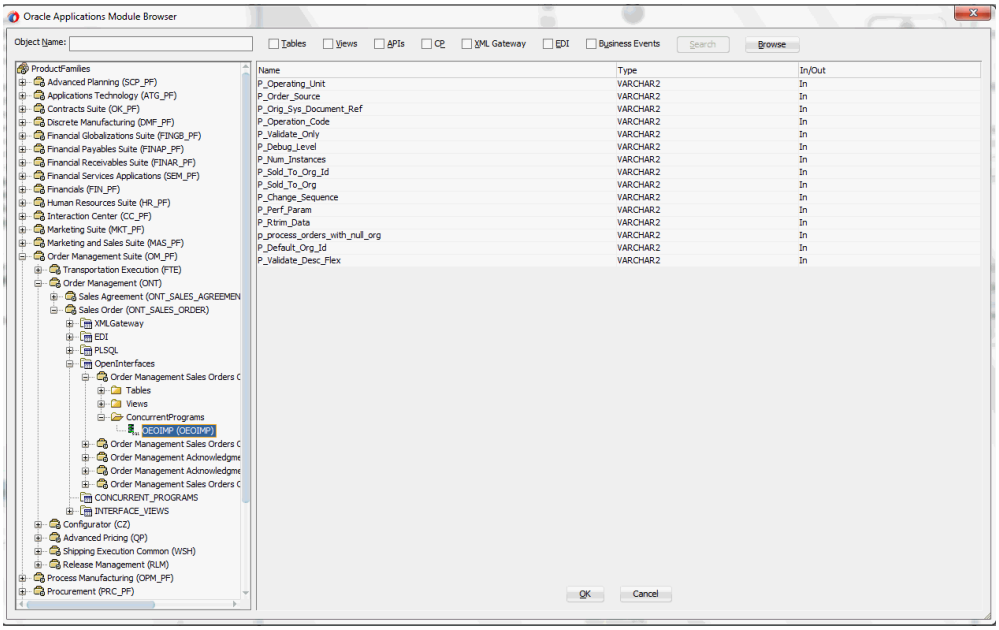


- Click **No** to query the Integration Repository data file from the live database you are connecting to display the Integration Repository tree.

In this example, click **Next** in the Operation page to open the Oracle E-Business Suite Module Browser.

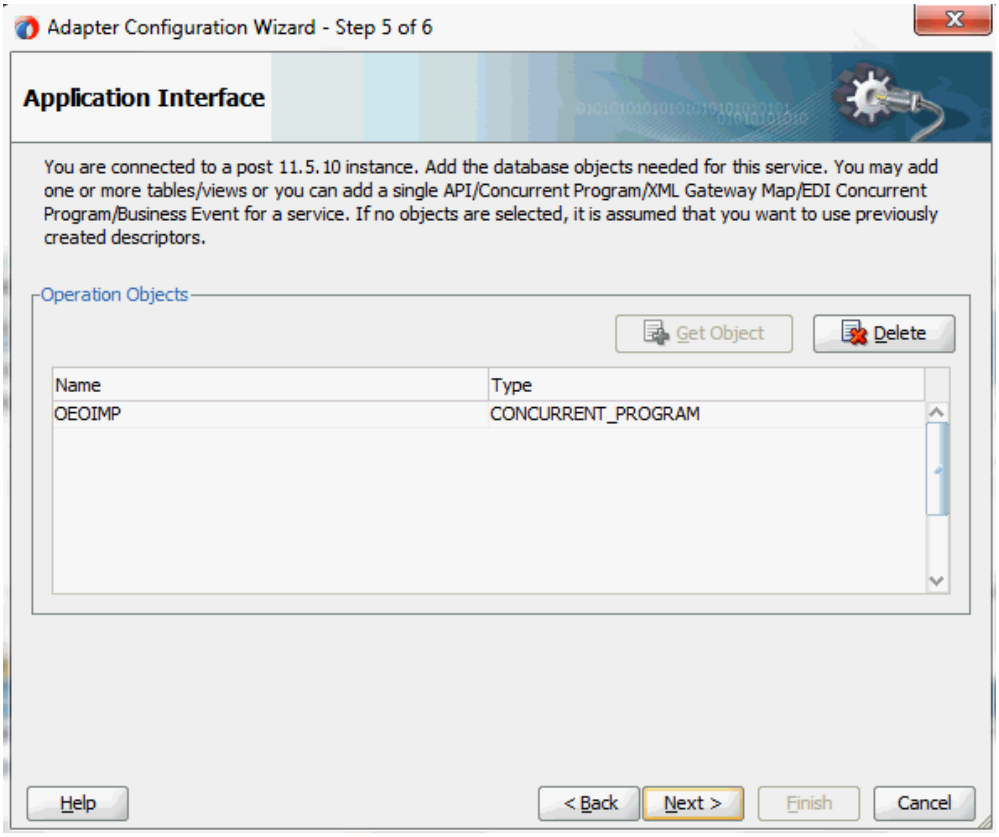
5. The Oracle E-Business Suite Module Browser combines interface data from Oracle Integration Repository with information about the additional interfaces supported by Oracle E-Business Suite Adapter, organized in a tree hierarchy.

   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)`.



6. Click **OK**. The Application Interface page appears.

Adapter Configuration Wizard - Application Interface Page



7. Click **Next**, then click **Finish** to complete the process. The partner link is created with the required WSDL settings. The wizard also generates the WSDL file corresponding to the XML schema. This WSDL file is now available for the partner link.

Adding Partner Links for File Adapter

Use this step to configure a BPEL process by synchronously reading payload from an input file to obtain the order details and concurrent program details.

Configure the following two partner links:

1. To obtain order details from an input file through Synchronous Read operation.
2. To obtain concurrent program details from an input file through Synchronous Read operation.

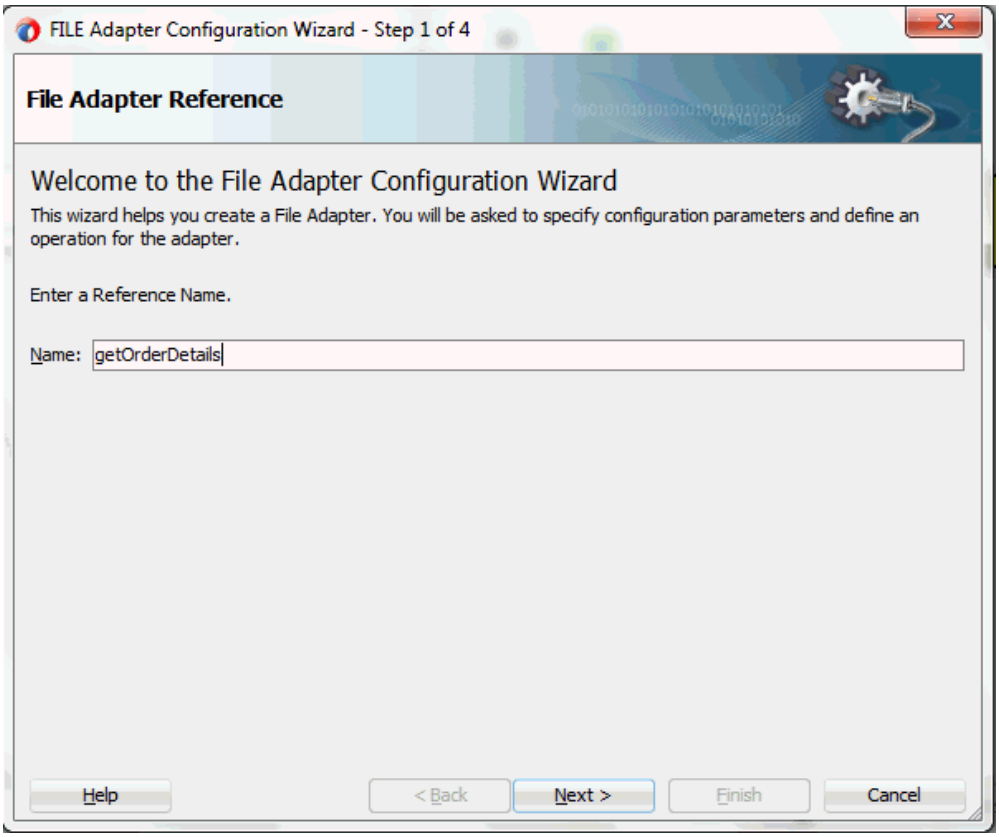
To add the first Partner Link for File Adapter to read order details:

1. In Oracle JDeveloper BPEL Designer, click **BPEL Services** in the Component

palette.

Drag and drop **File Adapter** from the BPEL Services list into the right Partner Link swim lane before the first partner link `InsertOrder`. The File Adapter Reference page appears.

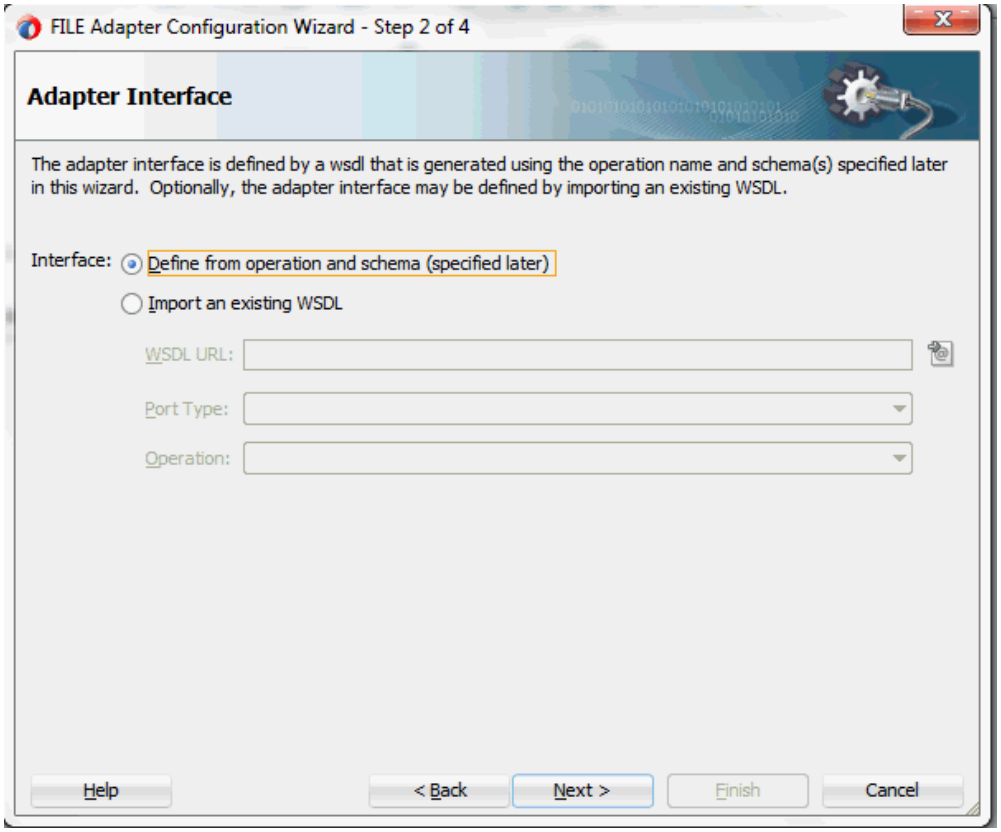
Specifying a File Reference Name



Enter a name for the File Adapter service, for example, `getOrderDetails`.

2. Click **Next**. The Adapter Interface page appears.

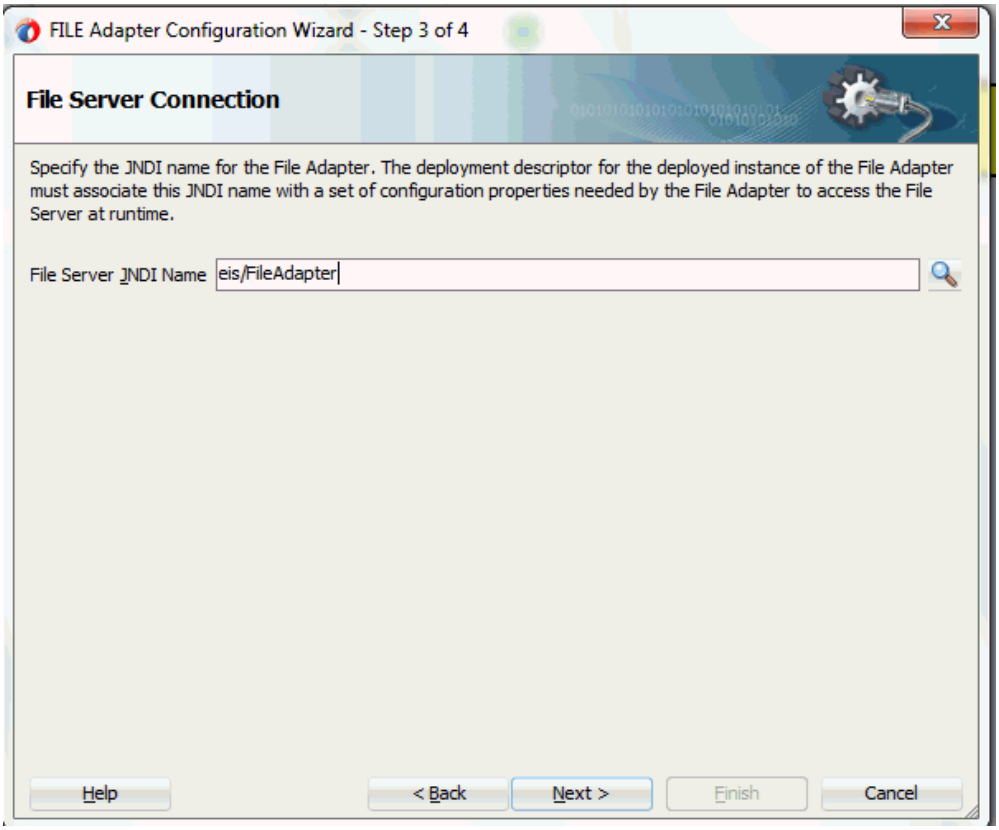
Specifying the Adapter Interface



Select the **Define from operation and schema (specified later)** radio button and click **Next**.

The File Server Connection page appears. Leave the default value unchanged for the File Server JNDI Name field.

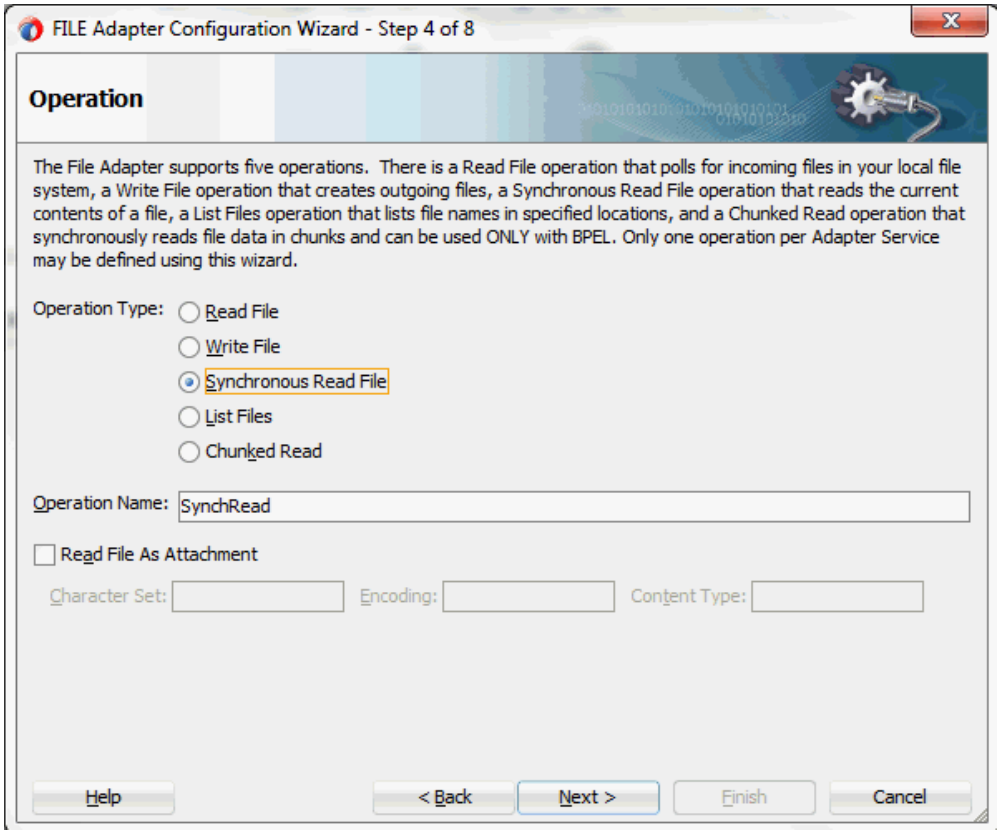
Specifying File Server JNDI Name



Click **Next**

3. In the Operation page, specify the operation type. For example, select the **Synchronous Read File** radio button. This automatically populates the **Operation Name** field.

Specifying the Operation

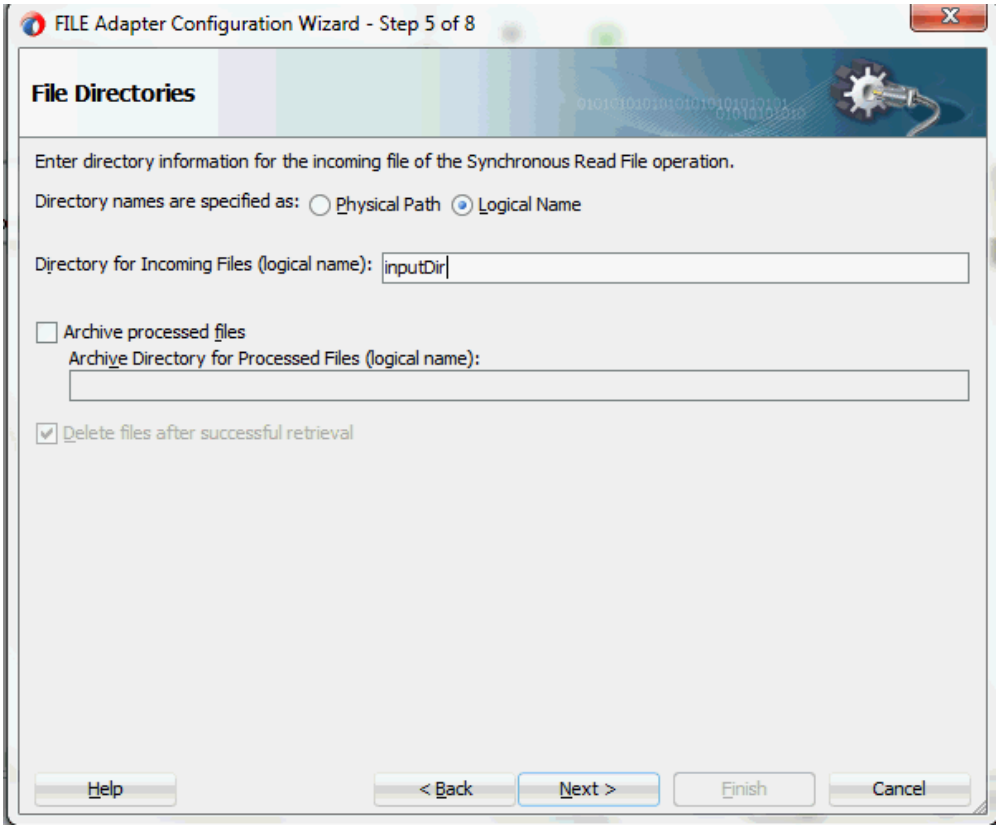


Click **Next** to access the File Directories page.

4. Select the **Logical Name** radio button and specify directory for incoming files, such as `inputDir`.

   Ensure the **Delete files after successful retrieval** check box is not selected.

Configuring the Input File



Click **Next** to open the File Name page.

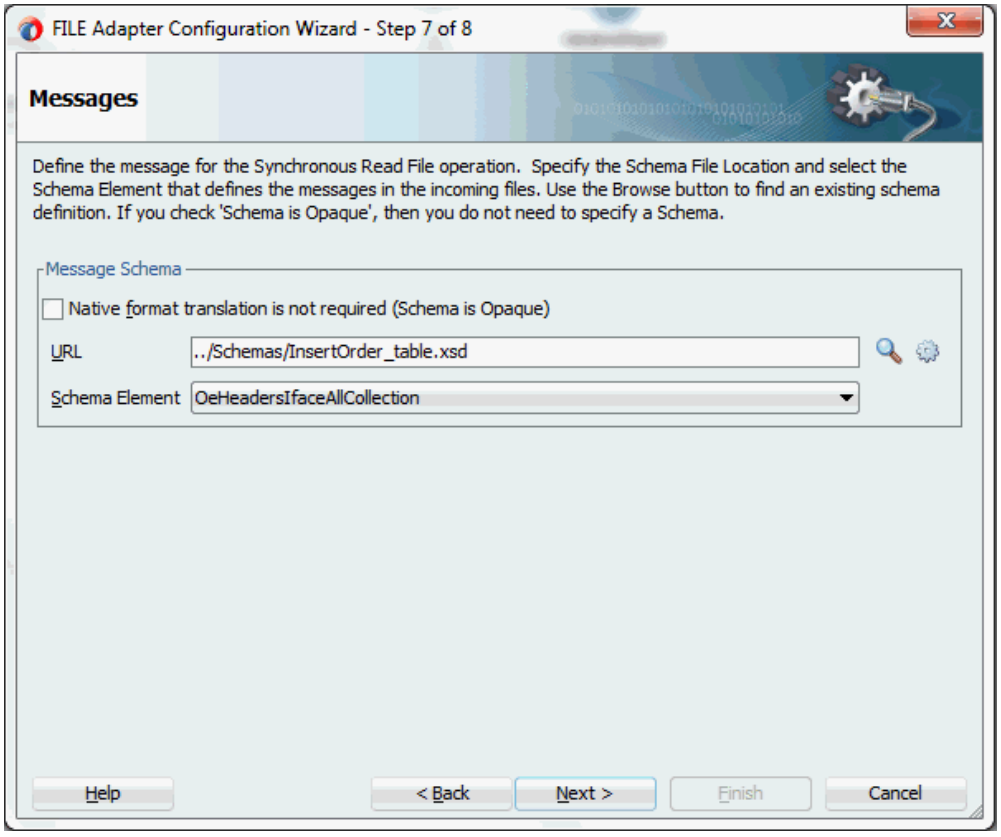
5. Enter the name of the file for the synchronous read file operation. For example, enter `order_data.xml`. Click **Next**. The Messages page appears.

6. Select the 'browse for schema file' icon next to the URL field to open the Type Chooser.

   Click Type Explorer and select *Project Schema Files* > *InsertOrder_table.xsd* > *OeHeadersIfaceAllCollection*. Click **OK**.

   The selected schema information will be automatically populated in the URL and Schema Element fields.

Specifying Message Schema



7. Click **Next** and then **Finish**. This completes the configuration and create the partner link with the required WSDL settings for the File Adapter service. The wizard also generates the WSDL file corresponding to the partner link. The main Create Partner Link dialog box appears, specifying the new WSDL file `getOrderDetails.wsdl`.

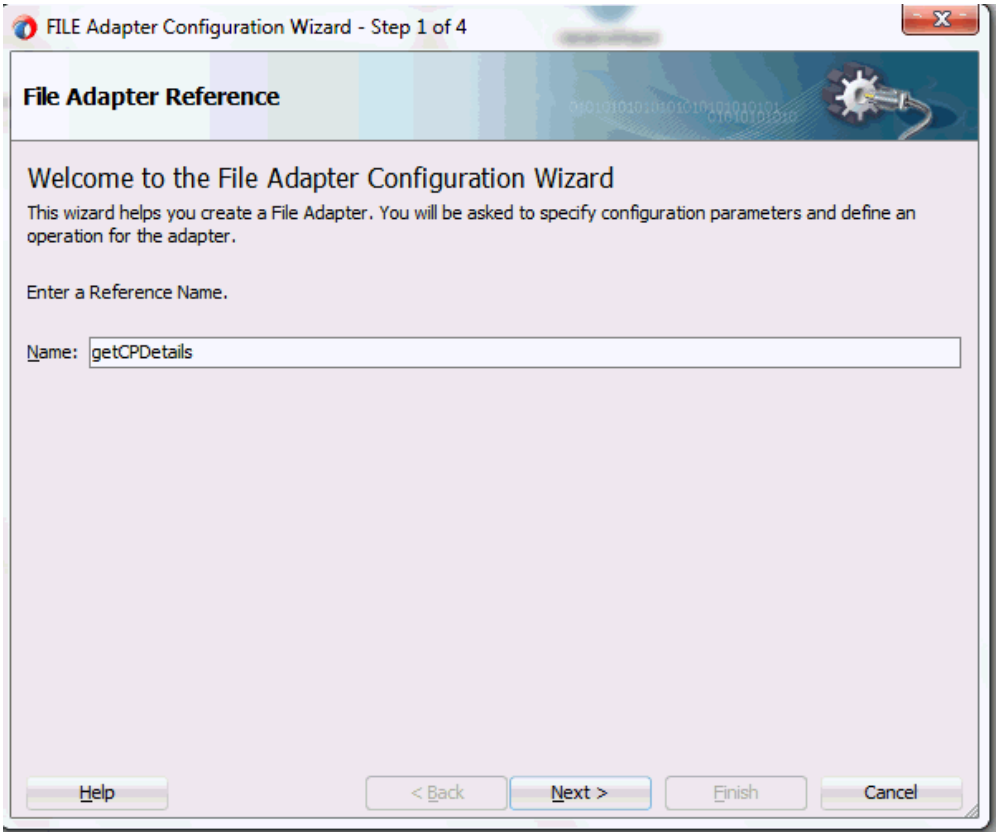
   The `getOrderDetails` Partner Link appears in the BPEL process diagram.

To add the second Partner Link for File Adapter to read concurrent program details:

1. In Oracle JDeveloper BPEL Designer, click **BPEL Services** in the Component palette.

   Drag and drop **File Adapter** from the **BPEL Services** list into the right Partner Link swim lane after the first File Adapter partner link `getOrderDetails`. The File Adapter Reference page appears.

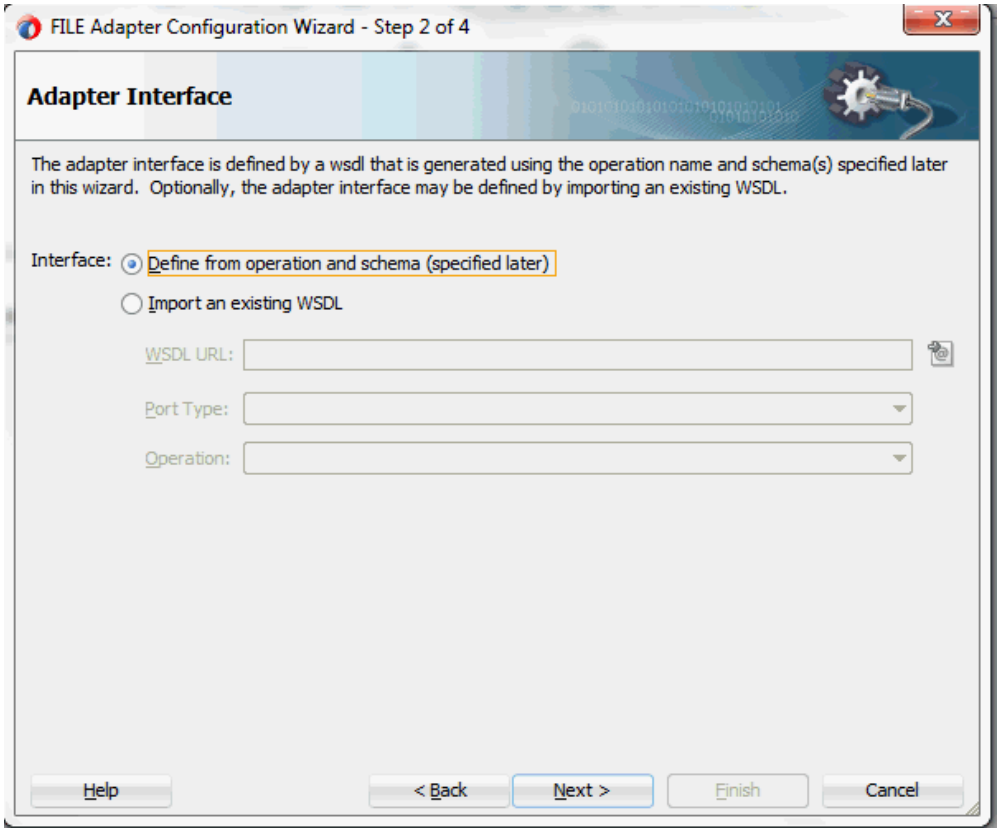
Specifying a File Adapter Reference Name



Enter a name for the File Adapter service, such as `getCPDetails`.

2. Click **Next**. The Adapter Interface page appears.

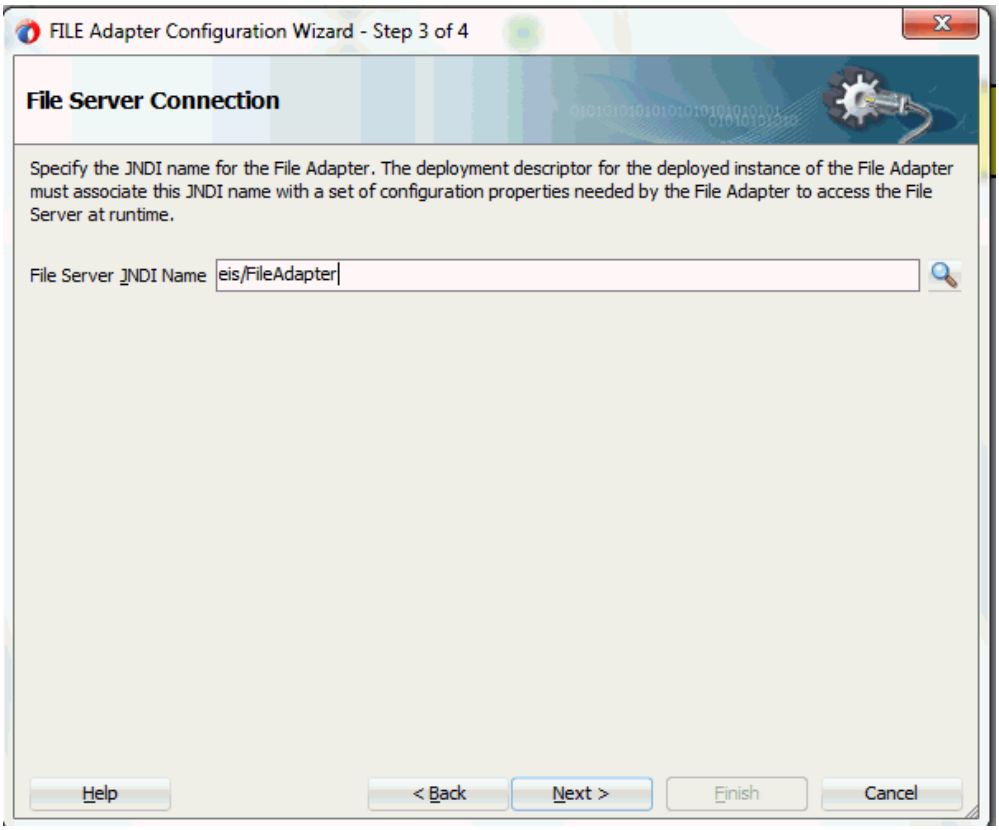
Specifying the Adapter Interface



Select the **Define from operation and schema (specified later)** radio button and click **Next**.

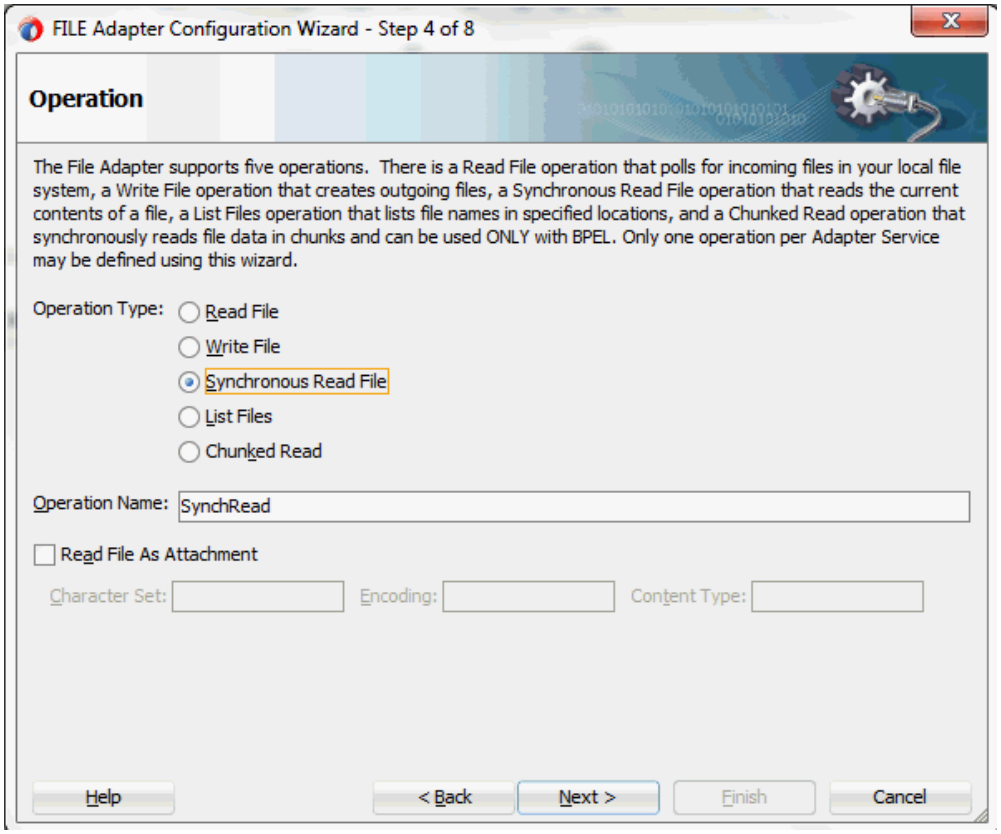
The File Server Connection page appears. Leave the default value unchanged for the File Server JNDI Name field.

Click **Next**



3. In the Operation page, specify the operation type. For example, select the **Synchronous Read File** radio button. This automatically populates the **Operation Name** field.

Specifying the Operation

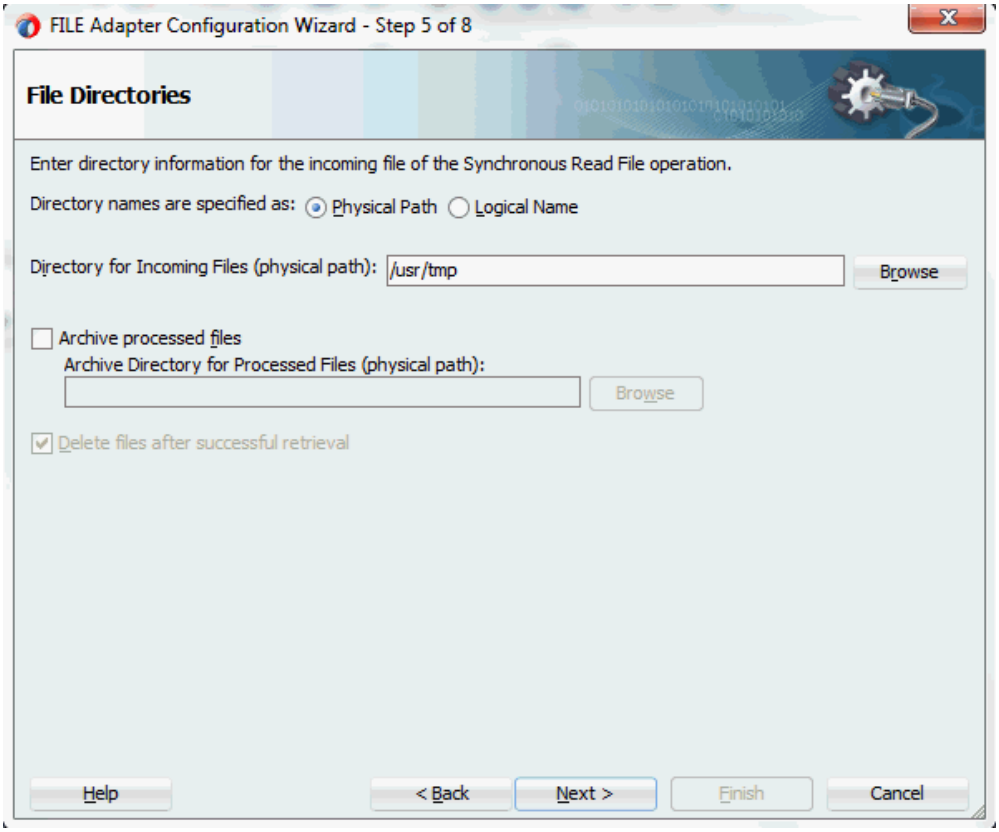


Click **Next** to access the File Directories page.

4. Select the **Logical Name** radio button and specify directory for incoming files, such as `inputDir`.

   Ensure the **Delete files after successful retrieval** check box is not selected.

Configuring the Input File



Click **Next** to open the File Name page.

5. Enter the name of the file for the synchronous read file operation. For example, enter `cp_data.xml`. Click **Next** to open the Messages page.

6. Select the 'browse for schema file' icon next to the URL field to open the Type Chooser.

   Click Type Explorer and select *Project Schema Files* > *ImportOrderCP_sp.xsd* > *InputParameters*. Click **OK**.

   The selected schema information will be automatically populated in the URL and Schema Element fields.

7. Click **Next** and then **Finish**. his completes the configuration and create the partner link with the required WSDL settings for the File Adapter service. The wizard also generates the WSDL file corresponding to the partner link. The main Create Partner Link dialog box appears, specifying the new WSDL file `getCPDetails.wsdl`.

   The `getCPDetails` Partner Link appears in the BPEL process diagram. Double click the Partner Link service icon to open the Edit Partner Link dialog and view the

details.

Configuring Invoke Activities

After adding and configuring partner links, you need to configure the following four Invoke activities:

1. To get the order details by invoking the `getOrderDetails` partner link through Synchronous Read operation to read the order from an input file.

2. To get the concurrent program details by invoking the `getCPDetails` partner link through Synchronous Read operation to read the concurrent program information from an input file.

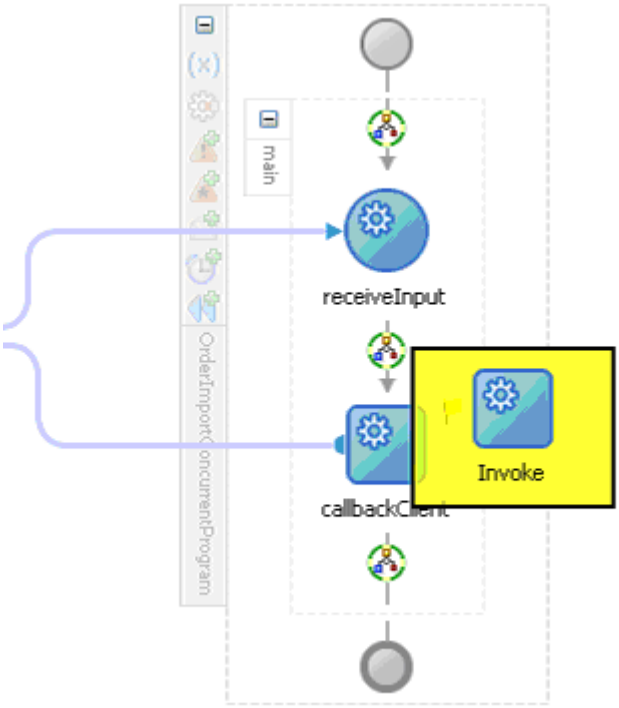
3. To insert order data into Open Interface tables by invoking `InsertOrder` partner link.

4. To import order data from Open Interface tables to Oracle E-Business Suite by invoking `ImportOrderCP` concurrent program partner link.

To add the first Invoke activity for a partner link to get order details:

1. In Oracle JDeveloper BPEL Designer, expand the **BPEL Constructs** from the Component Palette. Drag and drop the first **Invoke** activity from the **Web Service** section into the center swim lane of the process diagram, between the **receiveInput** and **callbackClient** activities.

Adding an Invoke Activity



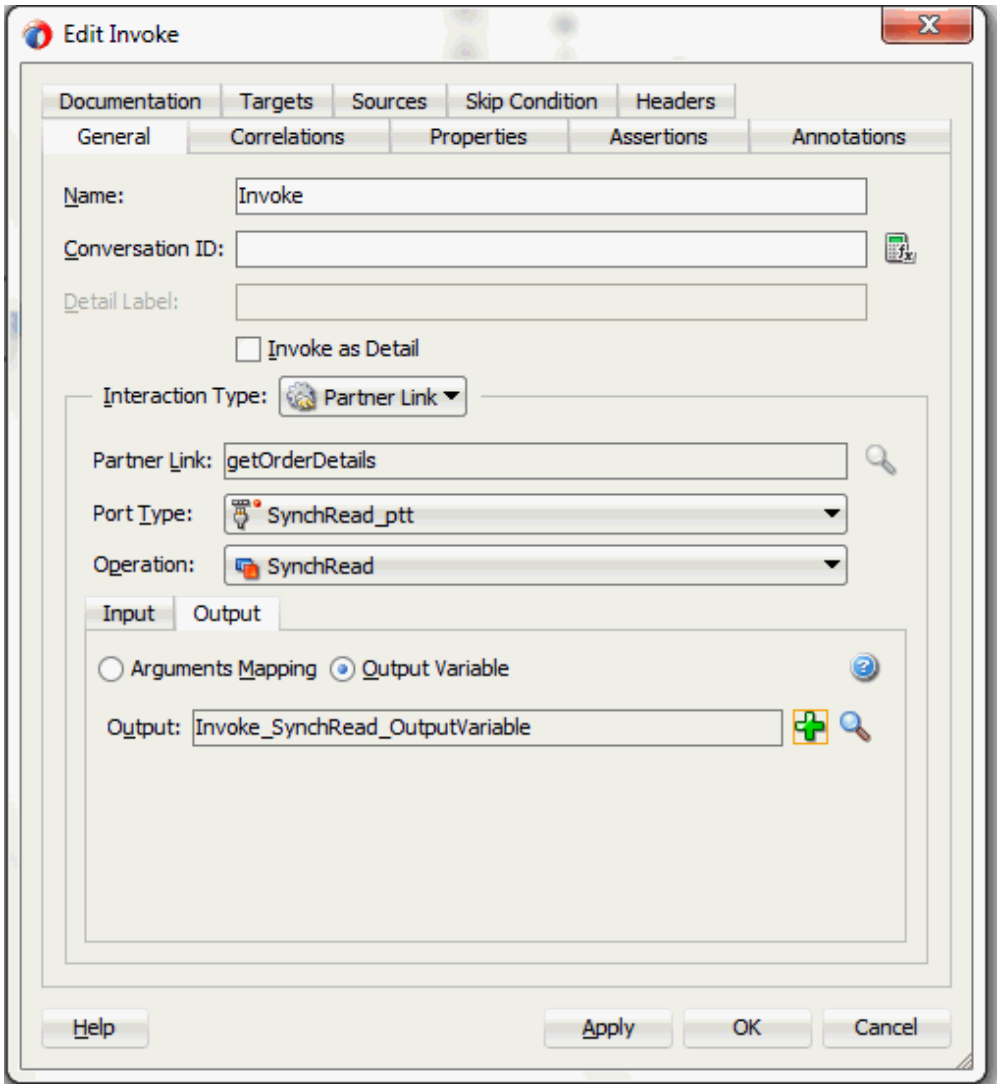
2. Link the Invoke activity to the `getOrderDetails` service. The Edit Invoke dialog appears.

3. Enter a name for the Invoke activity. In the Input tab, ensure the **Input Variable** radio button is selected. 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**, then enter a name for the variable. You can also accept the default name. Click **OK**.

5. In the Output tab, ensure the **Output Variable** radio button is selected. 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**, then enter a name for the variable. You can also accept the default name. Click **OK** to return to the Edit Invoke dialog box.



Click **Apply** and then **OK** to finish configuring the Invoke activity.

The Invoke activity appears in the process diagram.

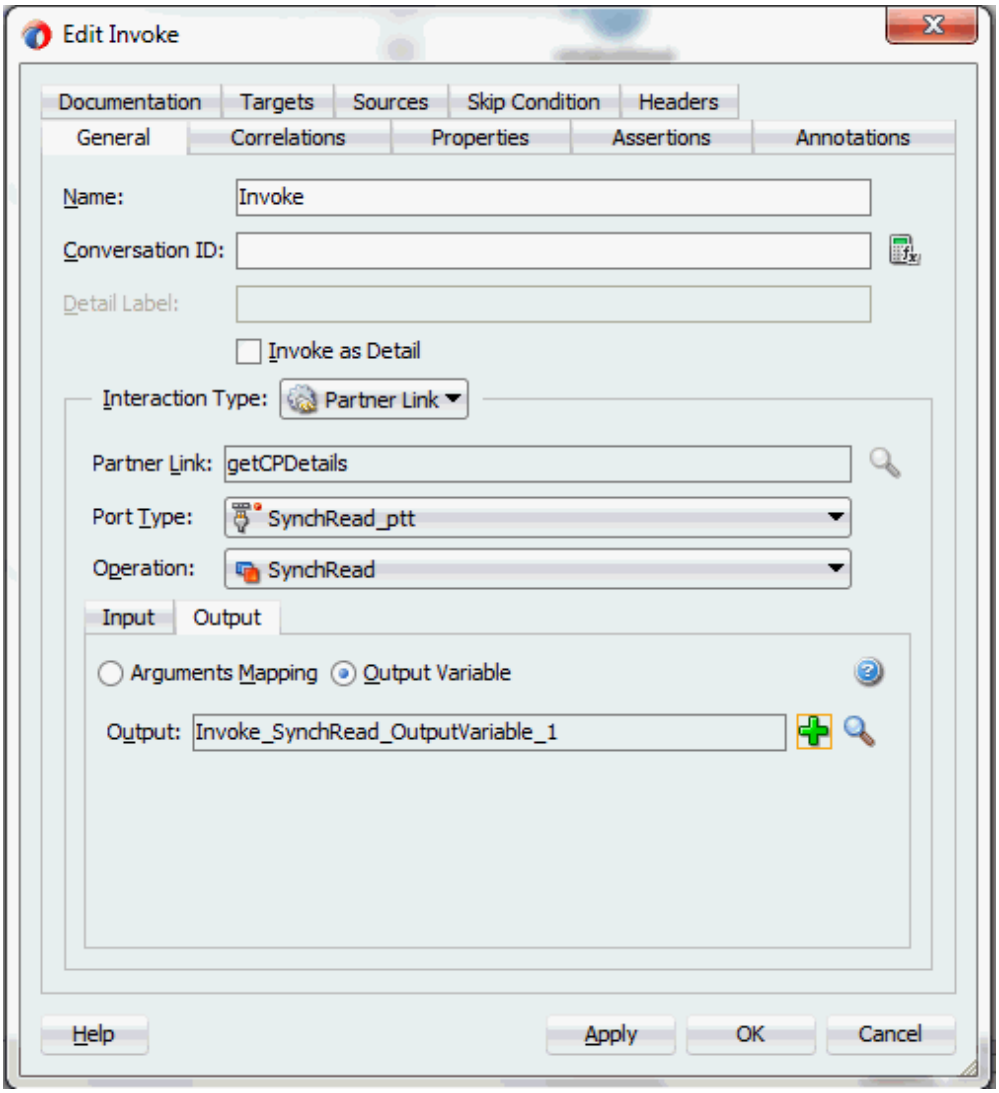
To add the second Invoke activity for a partner link to get concurrent program details:

1. In JDeveloper BPEL Designer, expand the **BPEL Constructs** from the Component Palette. Drag and drop the second **Invoke** activity from the **Web Service** section into the center swim lane of the process diagram right after the first **Invoke** activity.

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

3. Repeat Step 3 to Step 6 described in the first Invoke activity procedure.

The input and output variables should be populated in the Edit Invoke dialog box.

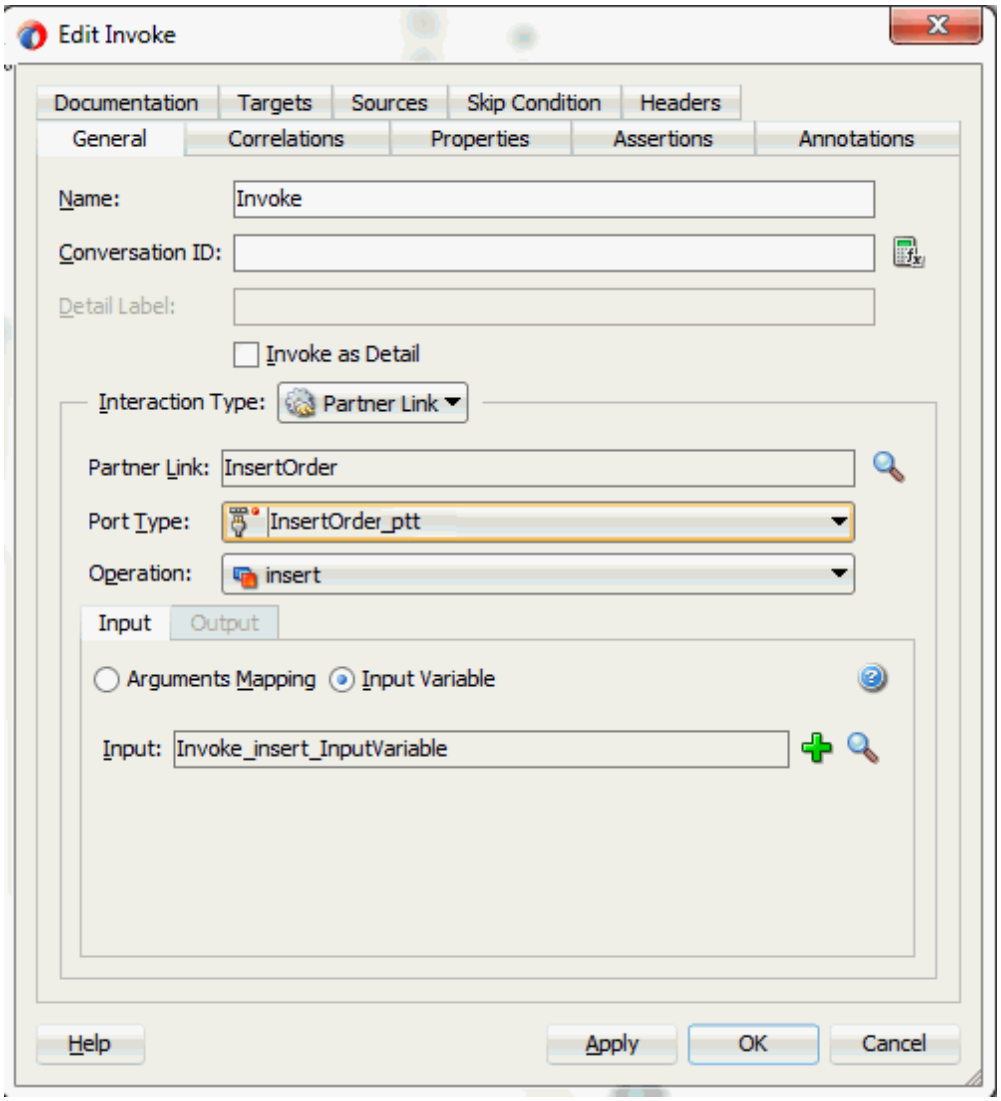


To add the third Invoke activity for a partner link to insert order data into Open Interface tables:

1. In JDeveloper BPEL Designer, expand the **BPEL Constructs** from the Component Palette. Drag and drop the third **Invoke** activity from the **Web Service** section into the center swim lane of the process diagram, between the second **Invoke** and **callbackClient** activities.

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

3. Repeat Step 3 to Step 4 described in the first Invoke activity procedure. Click **OK** in the Edit Invoke dialog box to finish configuring the second Invoke activity.



To add the fourth Invoke activity for a partner link to import the order information to Oracle E-Business Suite:

1. In JDeveloper BPEL Designer, expand the **BPEL Constructs** from the Component Palette. Drag and drop the fourth **Invoke** activity from the **Web Service** section into the center swim lane of the process diagram, between the third **Invoke** and **callbackClient** activities.

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

3. Repeat Step 3 to Step 6 described in the first Invoke activity procedure.

   The input and output variables should be populated in the Edit Invoke dialog box.

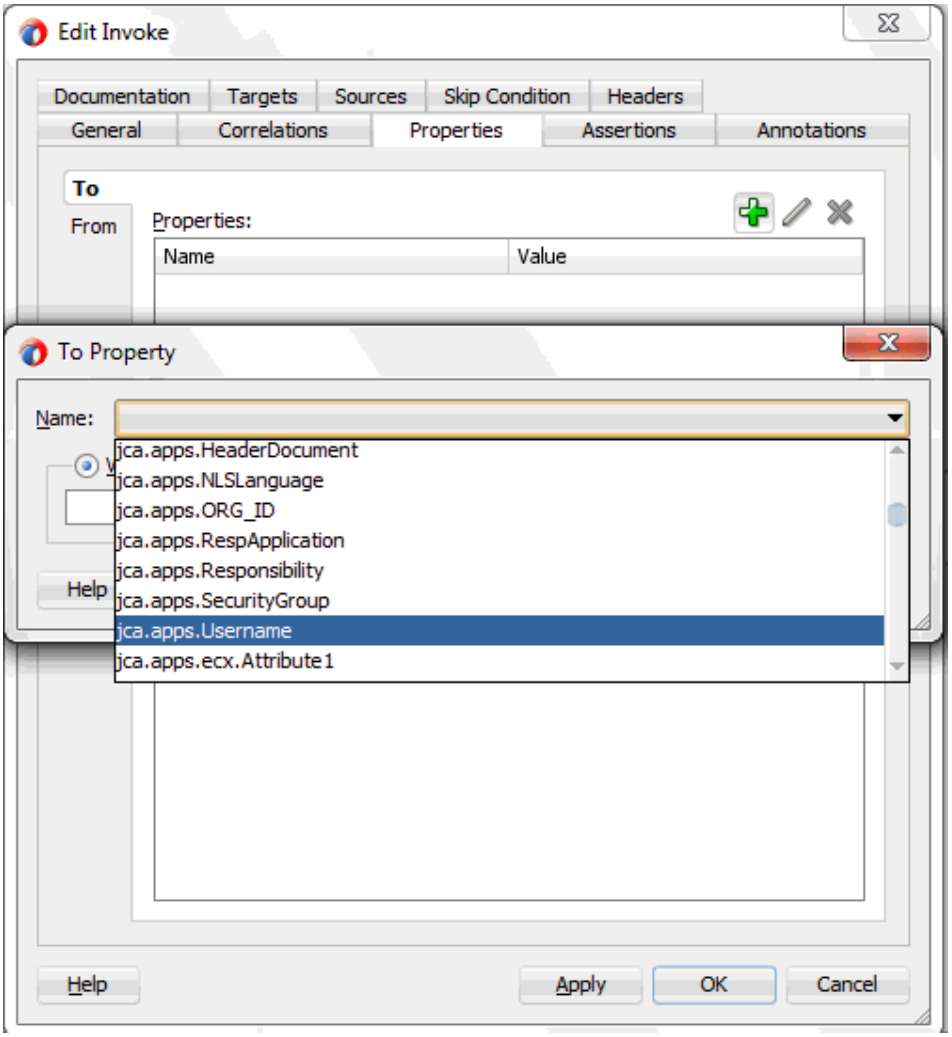
4. **Setting Header Properties for Applications Context**

   Use the following steps to set the header message properties required to pass applications context required to complete the BPEL process:

   1. Select the Properties tab in the Edit Invoke dialog box.

   2. Click the **Create** icon. The To Property dialog appears.

   3. Select the `jca.apps.Username` property from the Name drop-down list.



4. Select the **Expression** radio button and enter `'OPERATIONS'` as the property value.

   Click **OK**.

5. Repeat Step 2 to Step 4 to assign the following properties:

   - `'Order Management Super User, Vision Operations (USA)'` for `jca.apps.Responsibility`.
   - `204` for `jca.apps.ORG_ID`

5. Click **Apply** and then **OK** to complete the Invoke activity.

Configuring Assign Activities

Based on the BPEL process scenario, you need to configure the following two Assign activities:

1. To pass the output of File Adapter's Synchronous Read (`getOrderDetails`) service as an input to the Open Interface`InsertOrder` service.

2. To set the payload of the Concurrent Program (`ImportOrderCP`) service.

To add the first Assign activity:

1. In JDeveloper BPEL Designer, expand the **BPEL Constructs** from the Component Palette.

   Drag and drop the first **Assign** activity from the **Basic Activities** section into the center swim lane of the process diagram, between the first and second **Invoke** activities that you just created earlier.

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

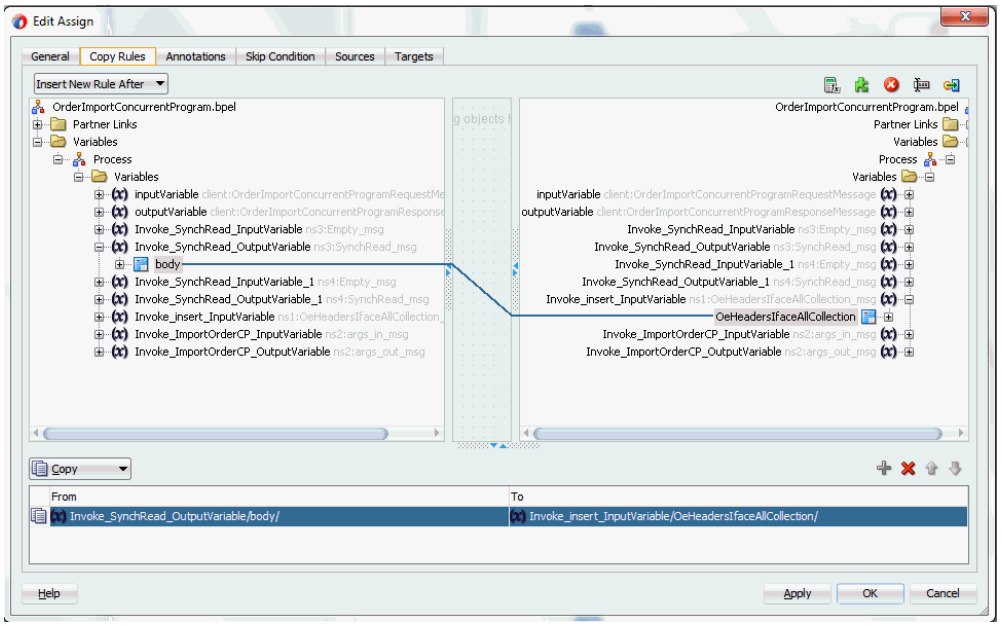
   Click the General tab to enter a name for the Assign activity. For example, `SetOrderDetails`.

3. Select the Copy Rules tab and expand the target trees:

   - In the From navigation tree, navigate to **Variable > Process > Variables > Invoke_SynchRead_OutputVariable** and select **body**.

   - In the To navigation tree, navigate to **Variable > Process > Variables > Invoke_insert_InputVariable** and select **OeHeadersIfaceAllCollection**.

   Drag the source node (body) to connect to the target node (OeHeadersIfaceAllCollection) that you just specified. This creates a line that connects the source and target nodes. The copy rule is displayed in the From and To sections at the bottom of the Edit Assign dialog box.



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

To add the second Assign activity:

1. In JDeveloper BPEL Designer, expand the **BPEL Constructs** from the Component Palette.

   Drag and drop the second **Assign** activity from the **Basic Activities** section into the center swim lane of the process diagram, between the third and fourth **Invoke** activities that you just created earlier.

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

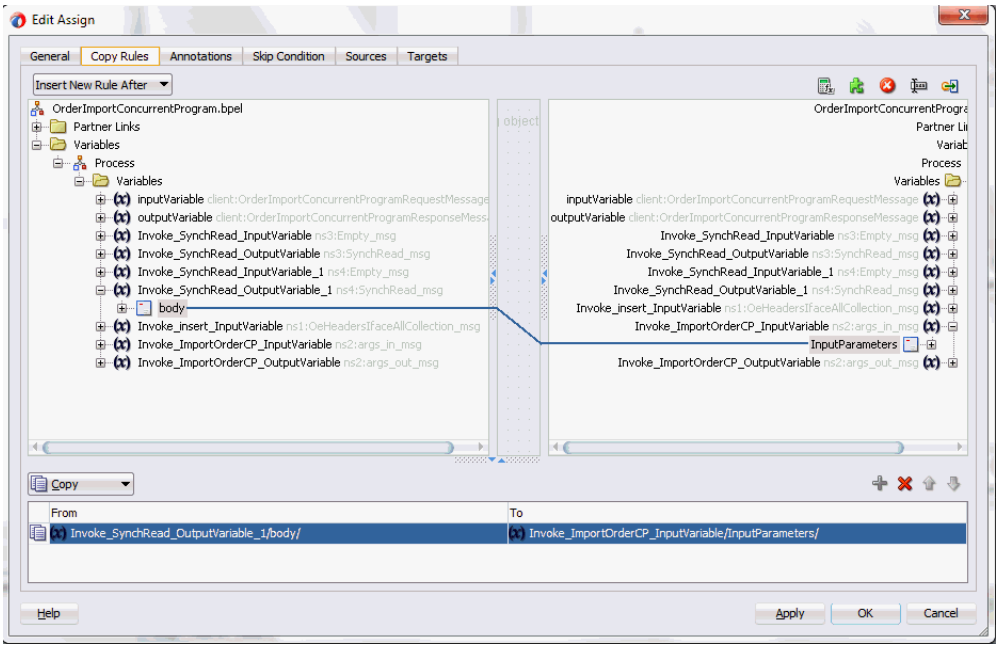
   Click the General tab to enter a name for the Assign activity. For example, `SetCPDetails`.

3. Select the Copy Rules tab and expand the target trees:

   - In the From navigation tree, navigate to **Variable > Process > Variables > Invoke_SynchRead_OutputVariable_1** and select **body**.

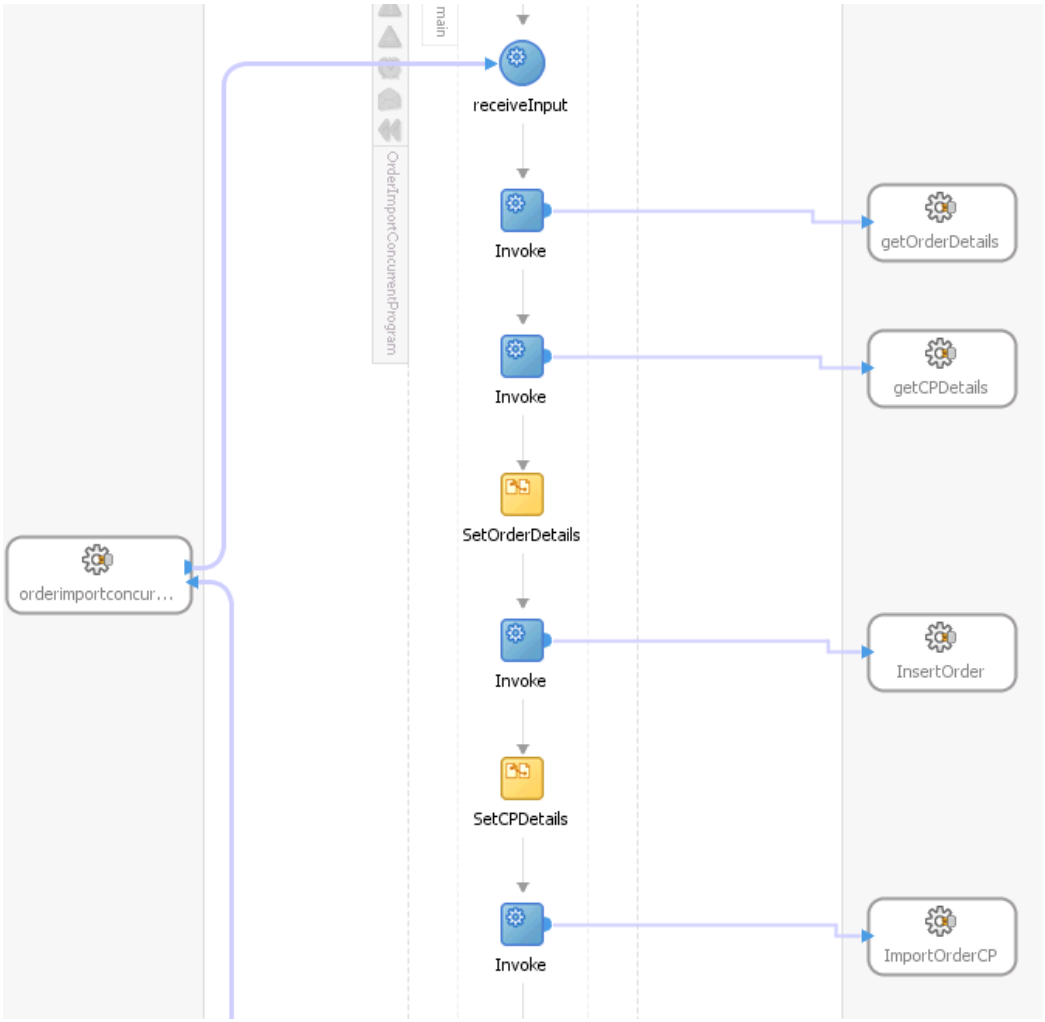
   - In the To navigation tree, navigate to **Variable > Process > Variables > Invoke_ImportOrderCP_InputVariable** and select **InputParameters**.

   Drag the source node (body) to connect to the target node (InputParameters) that you just specified. This creates a line that connects the source and target nodes. The copy rule is displayed in the From and To sections at the bottom of the Edit Assign dialog box.

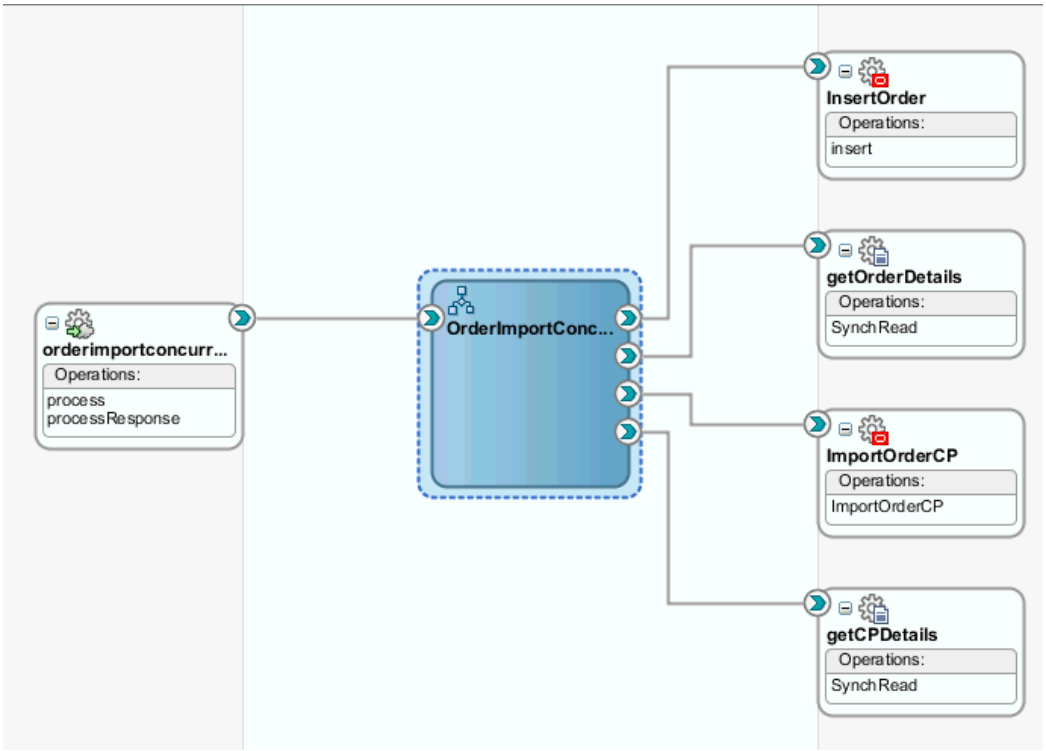


4. Click **Apply** and then **OK** to complete the configuration of the Assign activity.

Completed Concurrent Program BPEL Process Project



Click the `OrderImportConcurrentProgram` (composite.xml) to display the Oracle JDeveloper composite diagram:



Note: Click the Source tab of `OrderImportConcurrentProgram` (composite.xml) to enter a value for the physical directory `inputDir` for the reference `getOrderDetails` and `getCPDetails` (such as `/usr/tmp`).

```
<property name="inputDir" type="xs:string"
many="false" override="may">/usr/tmp</property>
```

Run-Time Tasks for Concurrent Programs

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

1. Deploy the SOA Composite application with BPEL process, page 7-50
2. Test the Deployed SOA Composite application with BPEL process, page 7-54
3. Verify records in Oracle E-Business Suite, page 7-58

Deploying the SOA Composite Application with BPEL Process

To invoke the services from the BPEL client contained in the SOA composite, the SOA composite needs to be deployed to the Oracle WebLogic managed server. This can be

achieved using Oracle JDeveloper. Once the composite is deployed, it can be tested from the Oracle Enterprise Manager Fusion Middleware Control Console.

Prerequisites

Before deploying the SOA composite with BPEL process using Oracle JDeveloper, you must have established the connectivity between the design-time environment and the run-time server. For more information, see Configuring the Data Source in Oracle WebLogic Server, page 11-1 and Creating an Application Server Connection, page 11-10.

> **Note:** If a local instance of the WebLogic Server is used, start the WebLogic Server by selecting **Run > Start Server Instance** from Oracle JDeveloper. Once the WebLogic Admin Server "DefaultServer" instance is successfully started, the <Server started in Running mode> and DefaultServer started message in the Running:DefaultServer and Messages logs should appear.

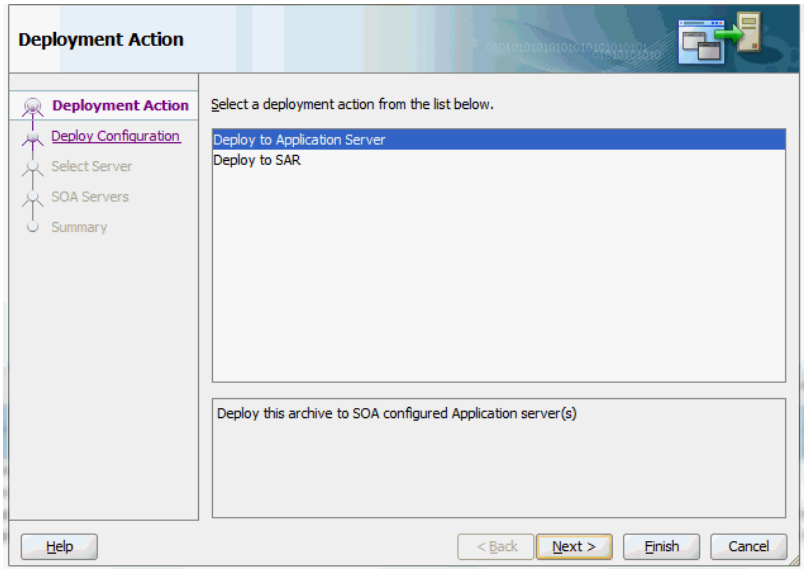
To deploy the SOA Composite application with BPEL process:

1. Select the SOA Composite project in the Applications Navigator.

2. Right-click the project name, then select **Deploy > [project name] > [serverConnection]** from the menu that appears.

   For example, you can select **Deploy > OrderImportConcurrentProgram > soa-server1** to deploy the process if you have the connection set up appropriately.

   > **Note:** If this is the first time to set up server connection, then the Deployment Action window appears. Select 'Deploy to Application Server' and click **Next**.

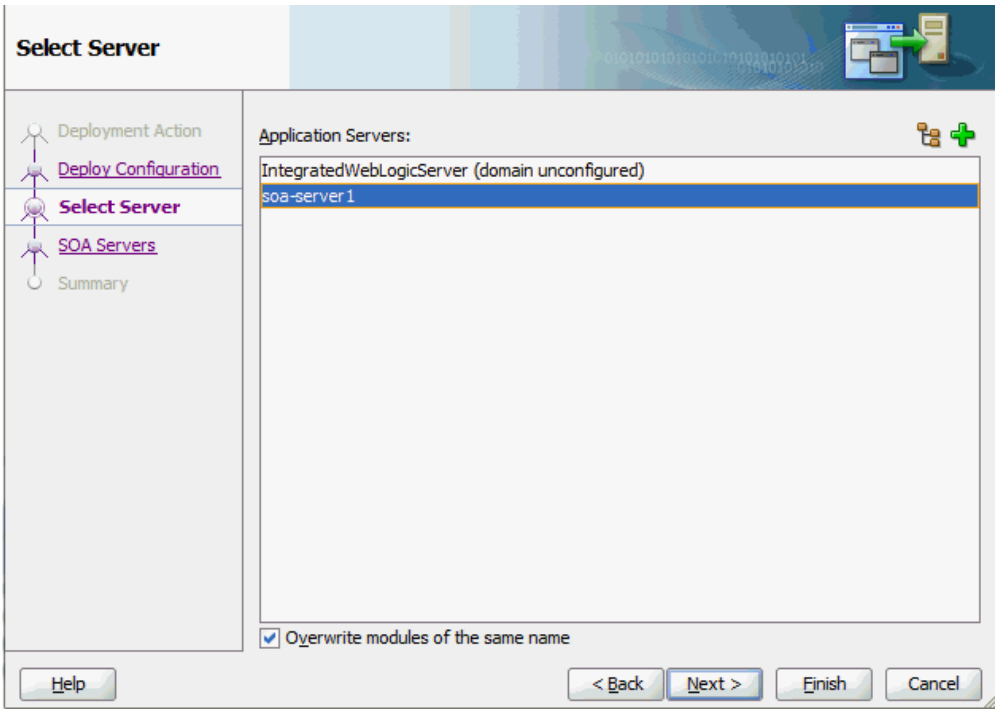


In the Deploy Configuration window, ensure the following information is selected before clicking **Next** to add a new application server:

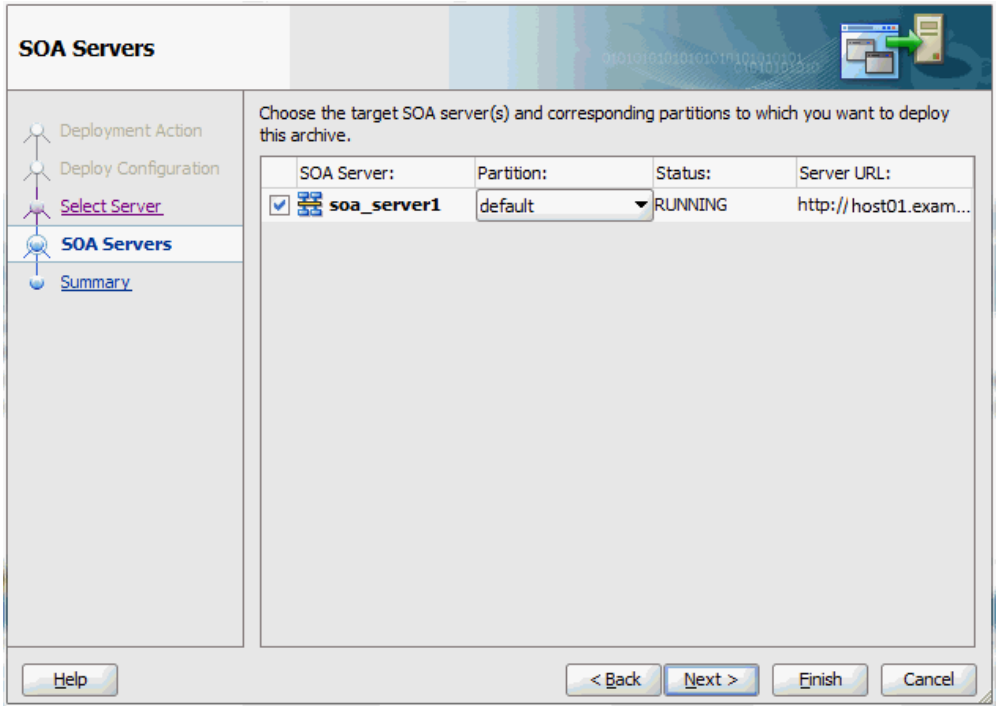
- New Revision ID: 1.0
- Mark composite revision as default: Select this check box.
- Overwrite any existing composites with the same revision ID: Select this check box.

The steps to create a new Oracle WebLogic Server connection from JDeveloper are covered in Creating an Application Server Connection, page 11-10.

3. In the Select Server page, select 'soa-server1' that you have established the server connection earlier. Click **Next**.



4. In the SOA Servers page, accept the default target SOA Server ('soa-server1') selection.



Click **Next** and **Finish**.

If you are deploying the composite for the first time from your Oracle JDeveloper session, the Authorization Request window appears. Enter username and password information specified during Oracle SOA Suite installation. Click **OK**.

5. Deployment processing starts. Monitor deployment progress and check for successful compilation in the SOA - Log window.

   Verify that the deployment is successful in the Deployment - Log window.

Testing the Deployed SOA Composite Application with BPEL Process

Once the BPEL process contained in a SOA Composite application is deployed, you can manage and monitor the process from the Oracle Enterprise Manager Fusion Middleware Control Console. You can also test the process and the integration interface by manually initiating the process.

To test the deployed SOA Composite application with BPEL process:

1. Navigate to Oracle Enterprise Manager Fusion Middleware Control Console (`http://<servername>:<portnumber>/em`). The composite you deployed is displayed in the Applications Navigation tree.

2. Enter username (such as `weblogic`) and password and click **Login** to log in to a farm.

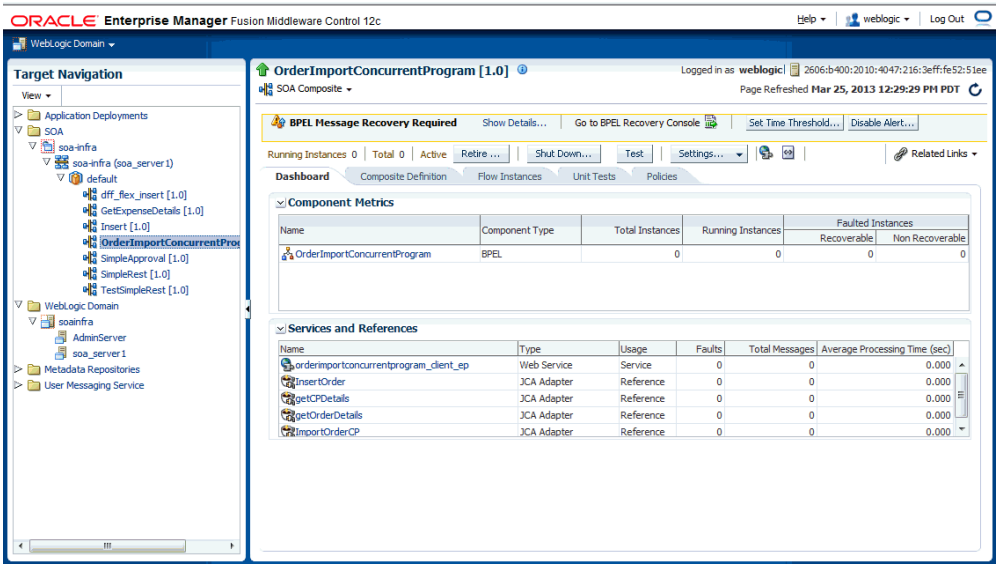
   You may need to select an appropriate target instance farm if there are multiple target Oracle Enterprise Manager Fusion Middleware Control Console farms.

3. From the Farm base domain, expand the **SOA > soa-infra > soa-infra (soa_server1) > default** to navigate through the SOA Infrastructure home page and menu to access your deployed SOA composite applications running in the SOA Infrastructure for that managed server.

   > **Note:** The Farm menu always displays at the top of the navigator. As you expand the SOA folder in the navigator and click the links displayed beneath it, the SOA Infrastructure menu becomes available at the top of the page.

   Click the SOA composite application that you want to initiate (such as 'OrderImportConcurrentProgram') from the SOA Infrastructure.

Viewing Deployed SOA Composites

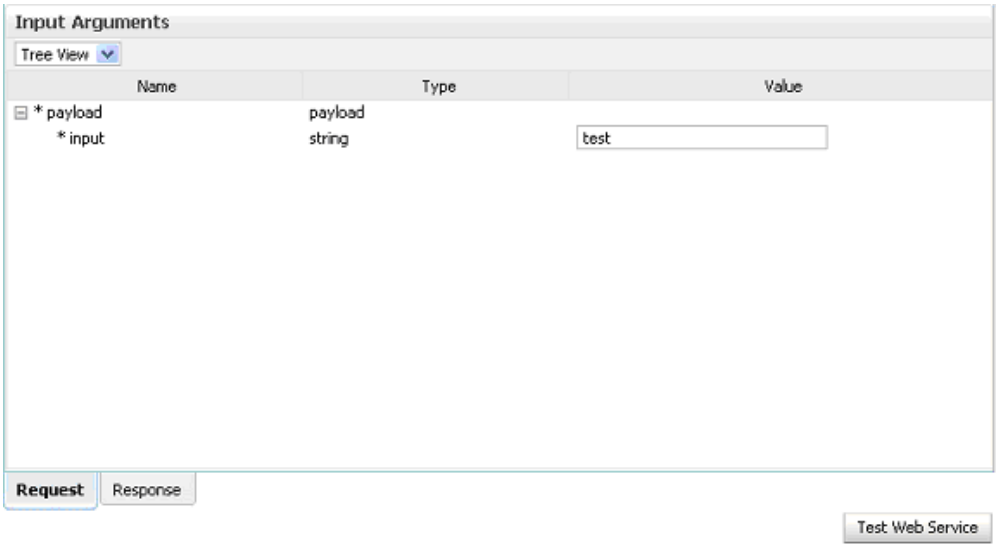


Click **Test** at the top of the page.

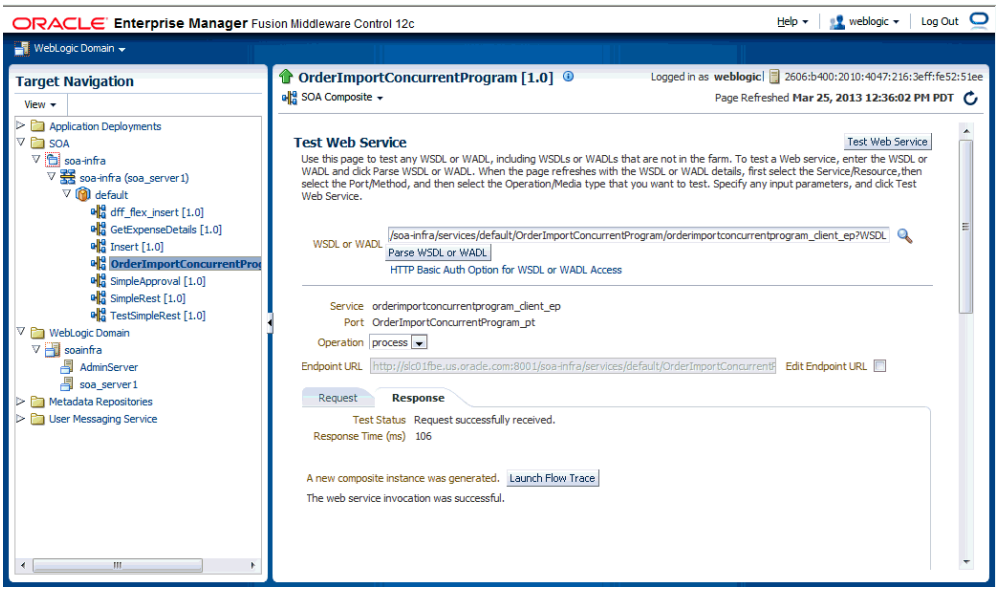
4. The Test Web Service page for initiating an instance appears. You can specify the XML payload data to use in the Input Arguments section.

   Enter the input string required by the process and click **Test Web Service** to initiate the process.

Testing Web Service



The test results appear in the Response tab upon completion.

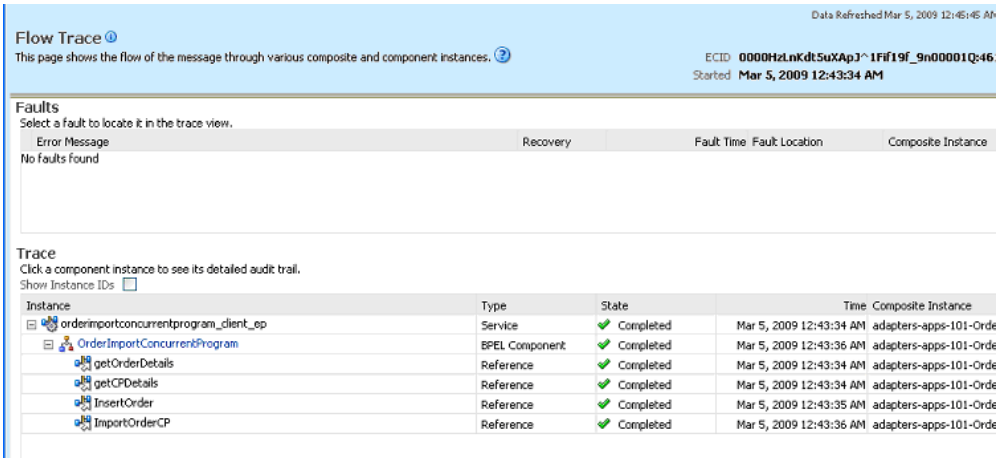
5. 

Click **Launch Flow Trace** in the Test Web Service page. The Flow Trace page is displayed.

In the Trace section, you should find the sequence of the message flow for the service binding component (`orderimportconcurrentprogram_client_ep`), BPEL component (`OrderImportConcurrentProgram`), and reference binding components (`getOrderDetails`, `getCPDetails`, `InsertOrder` and `ImportOrderCP`). All involved components have successfully received and

processed messages.

Flow Trace Page



If any error occurred during the test, you should find it in the Faults section.

6. Click your BPEL service component instance link (such as `OrderImportConcurrentProgram`) to display the Instances page where you can view execution details for the BPEL activities in the Audit Trail tab.

   Click the Flow tab to check the BPEL process flow diagram. Click an activity of the process diagram to view the activity details and flow of the payload through the process.

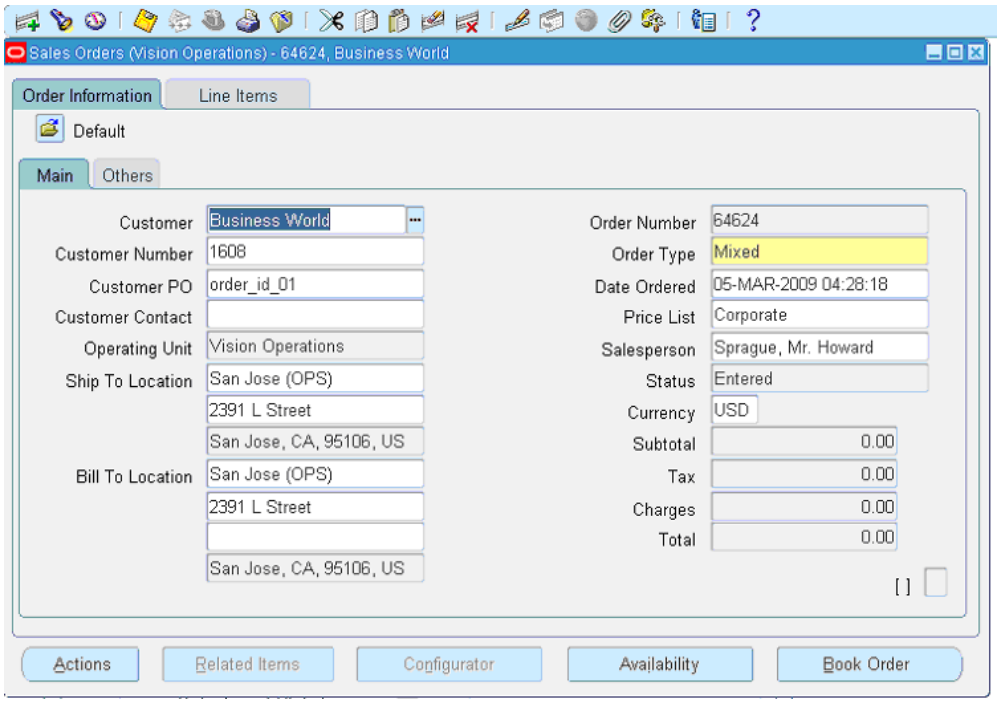
Verifying Records in Oracle E-Business Suite

To validate the BPEL process, log on to Oracle E-Business Suite forms to check the status of the Concurrent Program execution. Once it is successfully executed, the purchase order should also be created in Oracle E-Business Suite applications.

To verify records in Oracle E-Business Suite applications:

1. Log in to Oracle E-Business Suite with the Order Management Super User, Vision Operations (USA) responsibility.

2. Choose Orders, Return > Sales Order to open the Sales Orders form.



3. Search for an order by pressing **F11** key. In the Customer PO field, enter the order ID present in the xml file. For example, enter `order_id_01` and press **CTRL+F11** keys to execute the query.

4. You should find the newly created order appears with order ID `order_id_01`.

   Select the Line Items tab to verify the quantity and item type of the order which should be the same as that present in the xml file

5. Alternatively, you can also validate the result by querying the database with the following SQL statement:

   ```
   Select orig_sys_document_ref from oe_order_headers_all where
   orig_sys_document_ref = 'order_id_01';
   ```

Troubleshooting

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.

Enable logging for Adapter to see if the issue is on the middleware side. For information on how to enable logging, see Enabling Logging for Adapters, page 4-125.

8

Using Interface Tables and Views

This chapter covers the following topics:

- Overview of Interface Tables and Views
- Design-Time Tasks for Interface Tables
- Creating a New SOA Composite Application with BPEL Process
- Adding a Partner Link
- Adding a Partner Link for File Adapter
- Configuring Invoke Activities
- Configuring an Assign Activity
- Run-Time Tasks for Interface Tables
- Deploying the SOA Composite Application with BPEL Process
- Testing the Deployed SOA Composite Application with BPEL Process
- Design-Time Tasks for Views
- Creating a New SOA Composite Application with BPEL Process
- Adding a Partner Link
- Adding a Partner Link for File Adapter
- Configuring Invoke Activities
- Configuring an Assign Activity
- Run-Time Tasks for Views
- Deploying the SOA Composite Application with BPEL Process
- Testing the Deployed SOA Composite Application with BPEL Process

Overview of Interface Tables and Views

Interface tables can be used to insert and update data in Oracle E-Business Suite, and interface views can be used to retrieve data from Oracle E-Business Suite. This chapter describes the following interfaces:

- Interface Tables
- Interface Views

Interface Tables

Interface tables can be used to insert and update data in Oracle E-Business Suite. For example, you can insert a purchase order into Oracle E-Business Suite to generate the sales order automatically. Data is never loaded directly into Oracle E-Business Suite application 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.

Oracle E-Business Suite Adapter uses open interface tables to integrate with Oracle E-Business Suite applications through direct database access. Oracle E-Business Suite Adapter 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 E-Business Suite.

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 E-Business Suite 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.

Views

Oracle E-Business Suite Adapter uses views to retrieve data from Oracle E-Business Suite applications. For example, you can retrieve information about your customers from the required tables in Oracle E-Business Suite.

Views allow only simple definition. By using views, you can get synchronous data access to Oracle E-Business Suite. In Oracle E-Business Suite Adapter, views are created on base tables as well as interface tables. These views can be used *only* for select operations.

In the Oracle E-Business Suite 11.5.10 release, you cannot work on multiple 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 Oracle E-Business Suite Adapter to use interface tables. It describes the steps to configure the adapter using the Adapter Configuration Wizard in Oracle JDeveloper.

SOA Composite Application with BPEL Process Scenario

Take Open Interface tables (`OE_HEADER_IFACE_ALL` and `OE_LINES_IFACE_ALL`) as examples.

When a request of inserting order data into Open Interface tables is received, order details will be retrieved through synchronous read operation from an input file and then inserted into Open Interface tables (`OE_HEADER_IFACE_ALL` and `OE_LINES_IFACE_ALL`).

When the SOA Composite application with BPEL process has been successfully executed after deployment, you can validate the result by fetching the inserted data using SQL statement.

Prerequisites to Configure Interface Tables

- Define primary keys on all the interface tables being used.
- Define parent-child relationships among all the selected interface tables.

SOA Composite Application with BPEL Process Flow

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

1. Create a new SOA Composite application with BPEL process, page 8-3
2. Add a partner link, page 8-8
3. Add a partner link for File Adapter, page 8-22
4. Configure the Invoke activities, page 8-28
5. Configure the Assign activity, page 8-31

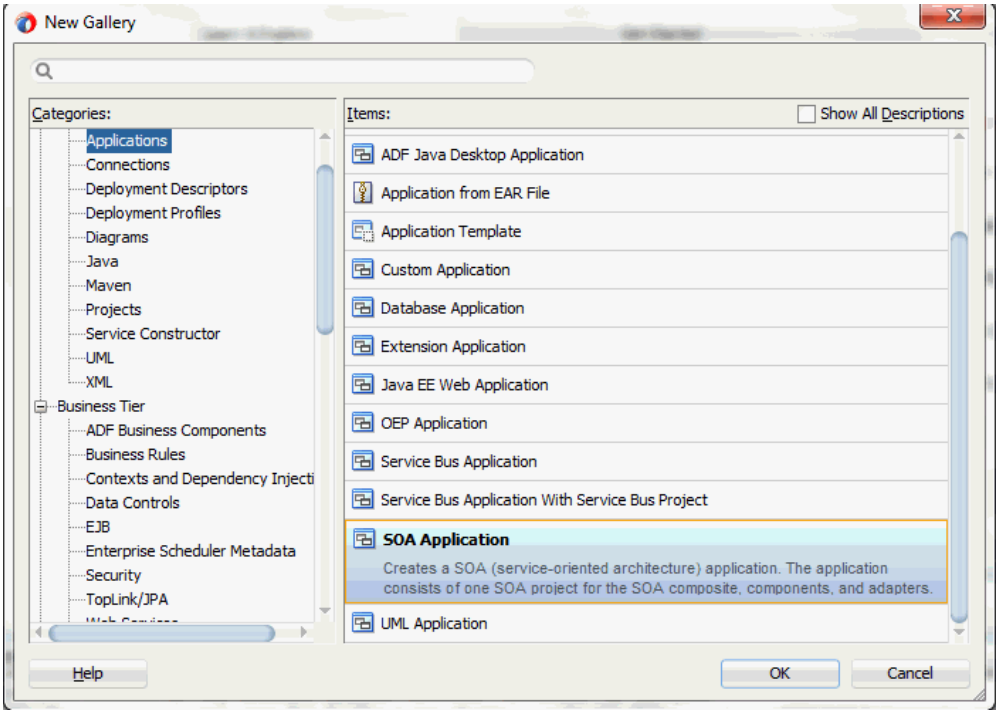
Creating a New SOA Composite Application with BPEL Process

To create a new SOA Composite application with BPEL process:

1. Open JDeveloper BPEL Designer.
2. Click **New Application** in the Application Navigator.

   The New Gallery page appears. Select **SOA Application** from the Items list.

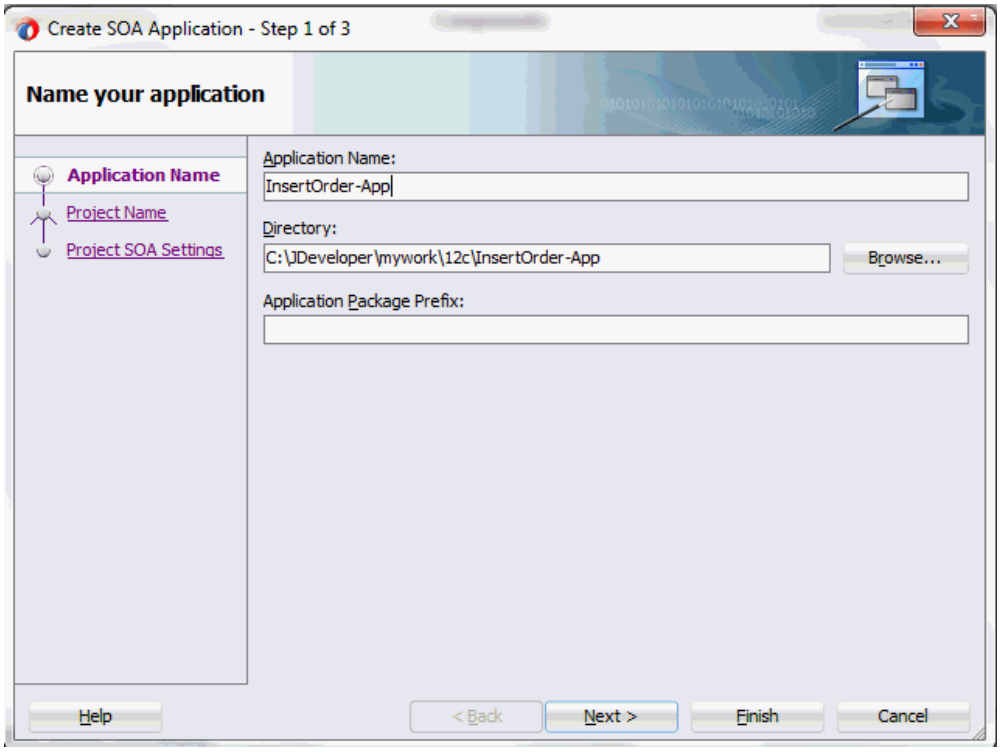
The New Gallery Page



Click **OK**.

3. The Create SOA Application - Name your application page is displayed.

The Create SOA Application - Name your application Page



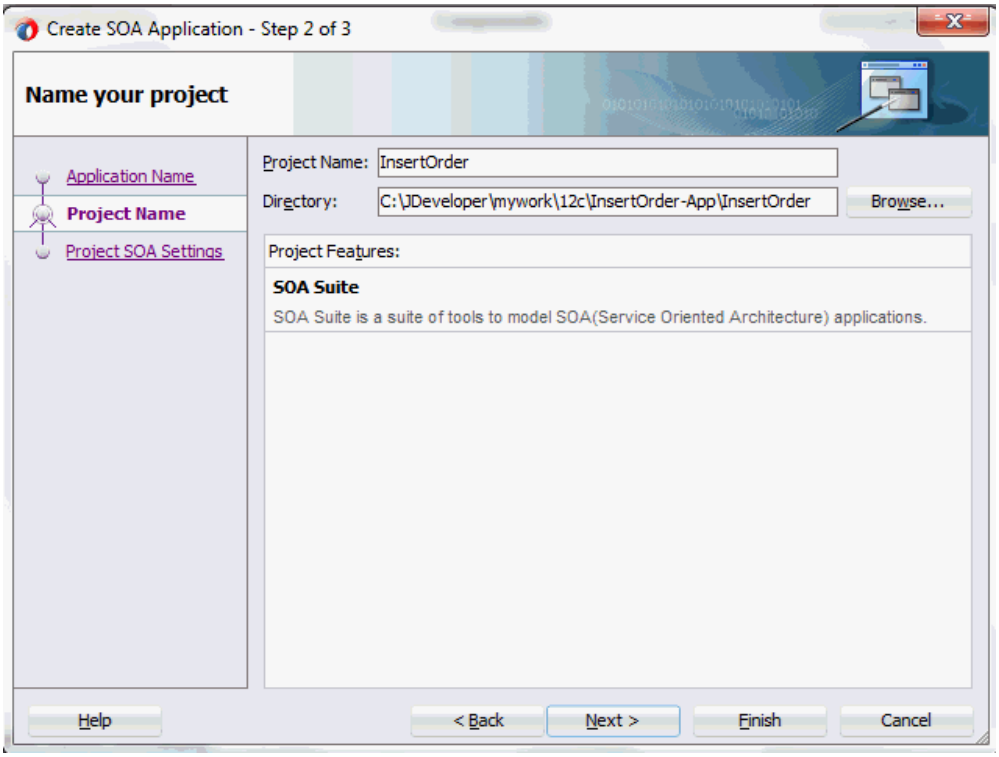
Enter an appropriate name for the application in the **Application Name** field.

Click **Next**. The Create SOA Application - Name your project page is displayed.

4. Enter an appropriate name for the project in the **Project Name** field. For example, `InsertOrder`.

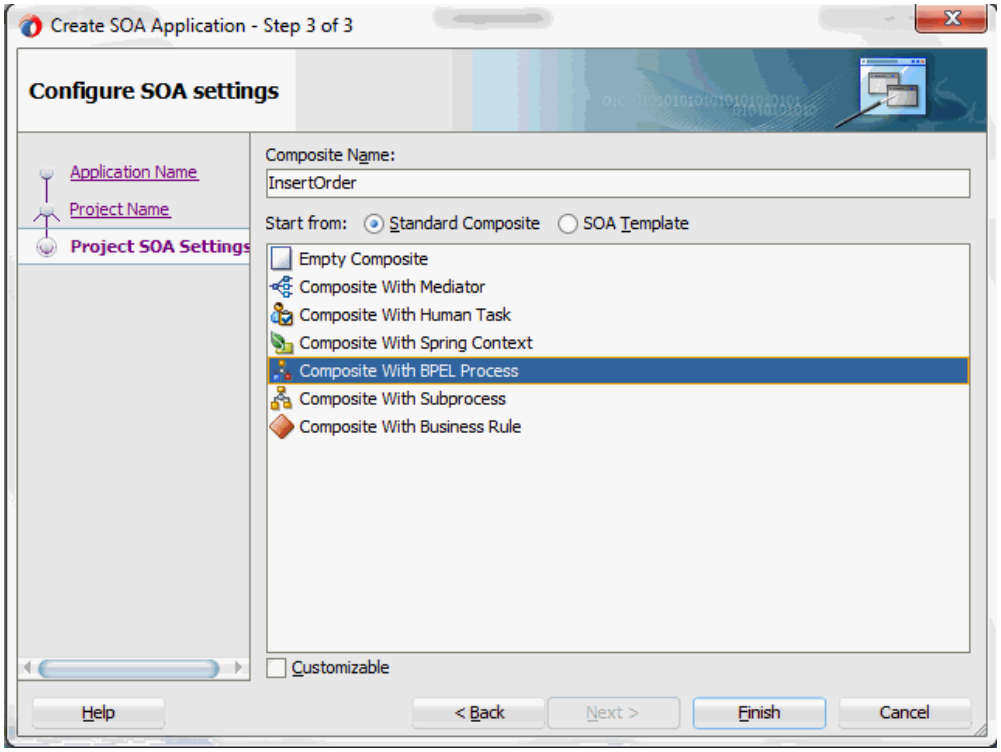
   Notice that **SOA Suite** is shown in the Project Features section.

The Create SOA Application - Name your project Page



5. Click **Next**. The Create SOA Application - Configure SOA settings page is displayed.

The Create SOA Application - Configure SOA settings Page



6. Leave the default **Standard Composite** selection unchanged for the Start from field.

   Select **Composite With BPEL Process** from the Standard Composite list, and then click **Finish**. You have created a new application, and a SOA project. This automatically creates a SOA Composite.

   The Create BPEL Process page is displayed.

7. Leave the default **BPEL 2.0 Specification** selection unchanged. This creates a BPEL project that supports the BPEL 2.0 specification.

   Enter an appropriate name for the BPEL process in the **Name** field. For example, `InsertOrder`.

   Select **Asynchronous BPEL Process** in the **Template** field. Click **OK**.

   An asynchronous BPEL process is created with the Receive and Reply activities. The required source files including bpel and wsdl, using the name you specified (for example, `InsertOrder.bpel` and `InsertOrder.wsdl`) and `InsertOrder` (composite.xml) are also generated.

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 to insert order details into selected Open Interface tables:

1. Click **BPEL Services** in the Component Palette.

   Drag and drop **Oracle E-Business Suite** (formerly known as **Oracle Applications**) from the **BPEL Services** list into the right Partner Link swim lane of the process diagram. The Oracle E-Business Suite Reference page appears.

   Enter a service name in the **Name** field. For example, `InsertOrder`.

2. 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 **Create a New Database Connection** icon.

     How to define a new database connection, see Create a New Database Connection, page 5-14.

   - 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 Oracle E-Business Suite Adapter 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 Fusion Middleware User's Guide for Technology Adapters.*

4. Click **Next** in the Service Connection dialog box. You can add an interface table by browsing through the list of interface tables available in Oracle E-Business Suite.

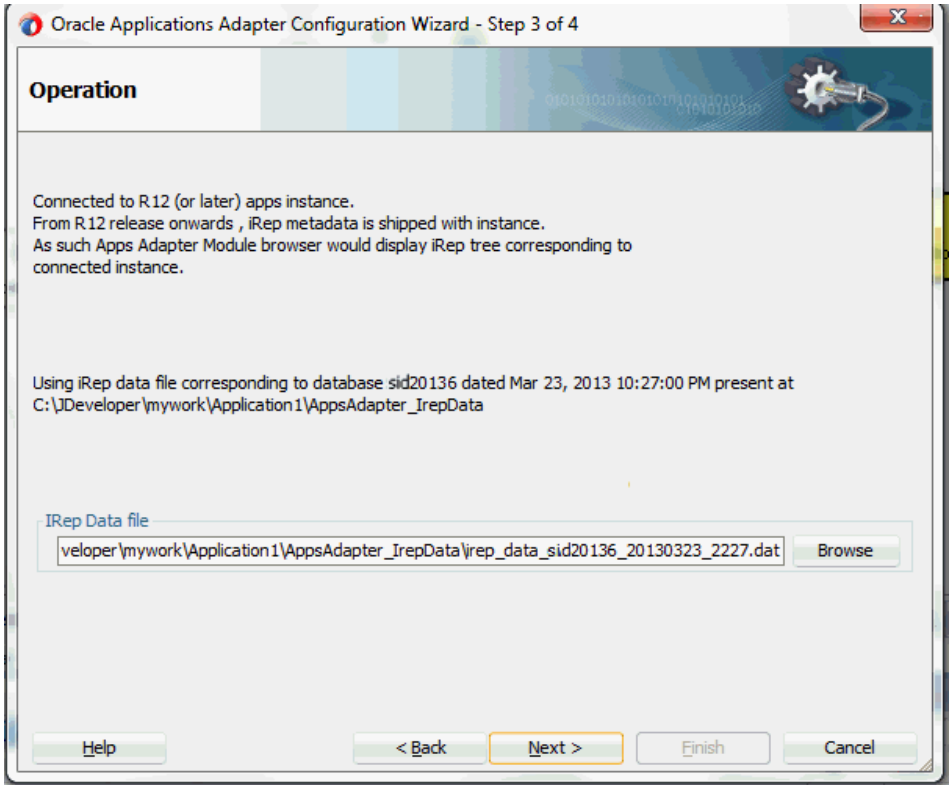
   If you are connecting to Oracle E-Business Suite Release 12, then the **IREP File not present** dialog box appears indicating that Adapter could not find the Oracle Integration Repository data file corresponding to the database you are connecting in your workspace. Absence of the data file would make 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 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.

     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



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

In this example, click **No**. Click **Next** in the Operation page to open the Oracle E-Business Suite Module Browser.

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

> **Note:** Oracle E-Business Suite Module Browser includes the various product families that are available in Oracle E-Business Suite. For example, Applications Technology or Order Management Suite are product families in Oracle E-Business Suite. The product families contain the individual products. For example, Order Management Suite contains the Order Management product. The individual products contain the business entities associated with the product. For example, the Order Management product contains the Sales Order business entity.

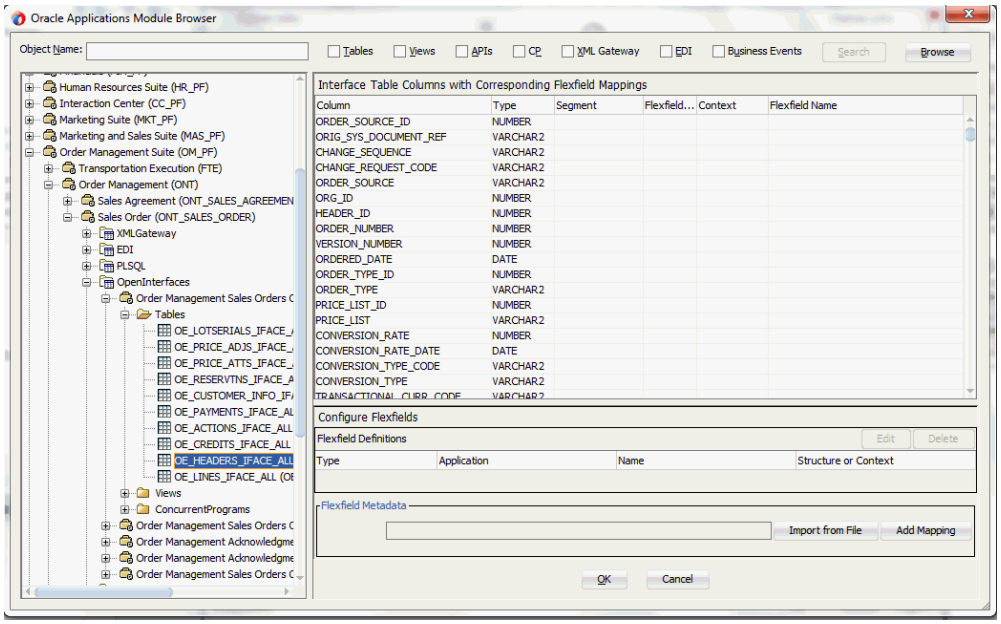
Business entities contain the various application modules that are exposed for integration. These modules are grouped according to the interface they provide. concurrent programs can be found under the Open Interfaces category.

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`.

Note: You can also search for a table by entering the name of the program in the **Object Name** field. Select the **Tables** check box, then click **Search**.

6. Oracle E-Business Suite Adapter provides flexfield support for Open Interface Tables. Once the selected table is displayed in the Oracle E-Business Suite Module Browser, you can either configure flexfield mappings for the table that has flexfields defined in Oracle E-Business Suite or use current data without further flexfield configuration.

Displaying a Selected Interface Table from the Module Browser



Note: If a selected interface table has pre-configured flexfield mappings available, a pop-up window appears allowing you to decide if you want to use these pre-configured mappings. Click **Yes**

to display these mappings as part of the flexfield definitions in the Configure Flexfields region. You can modify them later if needed.

Click **No** to indicate that you will not use these mappings and the pop-up window disappears.

Configuring Flexfield Mapping Data for the Selected Open Interface Table

Perform either one of the following tasks in the Oracle E-Business Suite Module Browser:

- Use the current data without further configuration by clicking **OK** to continue.
- Configure flexfield mapping data by either using an existing flexfield mapping or creating a new mapping through a flexfield wizard if desired.
  - **Creating a New Mapping**

    You can create a new mapping for a selected open interface table by clicking **Add Mapping** in the Flexfield Metadata section. A new flexfield mapping wizard appears guiding you through each configuration page where you can add key and descriptive flexfields, as well as configure flexfield mapping between the selected interface table columns and flexfields defined in the Oracle E-Business Suite instance.

    For more information on how to create a new mapping and configure key and descriptive flexfields, see:

    - Adding or Configuring a New Mapping, page 4-37
    - Configuring Key Flexfield Mappings for an Open Interface Table, page 4-79
    - Configuring Descriptive Flexfield Mappings for an Open Interface Table, page 4-85

  - **Importing an Existing Mapping**

    Instead of creating a new mapping, you can use an existing mapping that has been created earlier by clicking **Import from File** in the Flexfield Metadata section. This lets you select a desired flexfield mapping for your selected interface table. Once a desired mapping is imported, you can modify the context values for a descriptive flexfield, and modify structure for a key flexfield if needed to meet your needs.

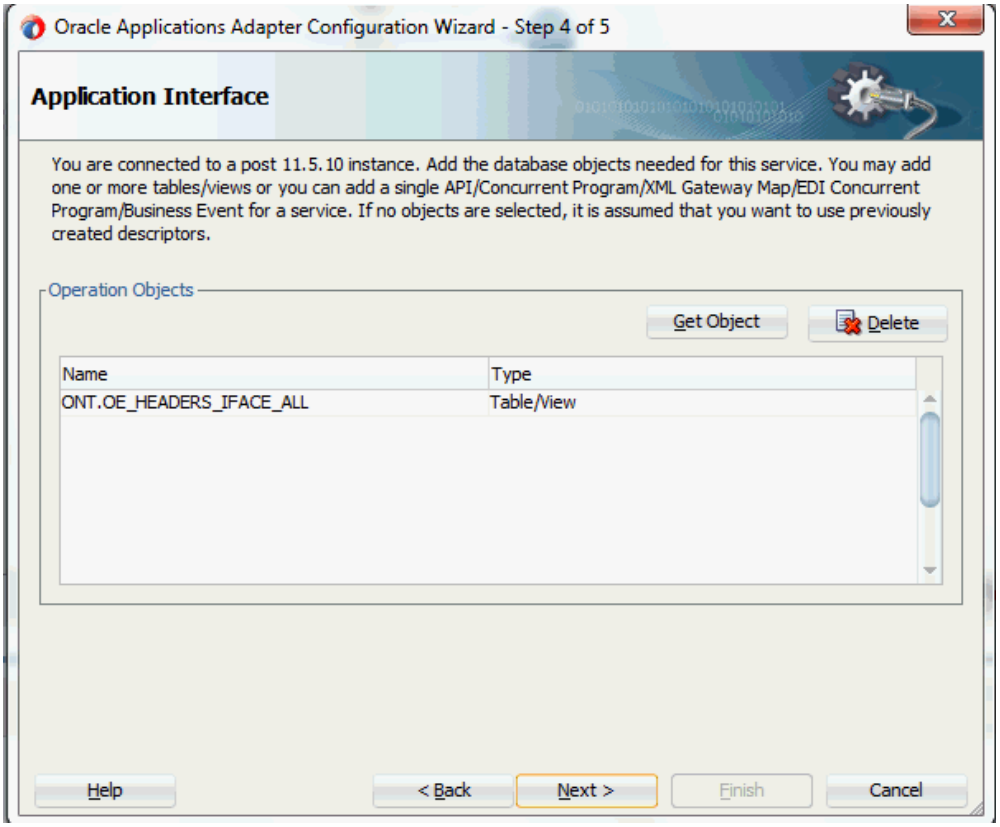
    See: Importing an Existing Flexfield Mapping, page 4-115.

For more information on flexfield support for open interface tables and how to modify existing mapping configuration, see Flexfield Support for PL/SQL APIs

and Open Interface Tables, page 4-27.

In this example, click **OK** to continue. The Application Interface page appears with the selected open table.

Adapter Configuration Wizard - Application Interface Page



7. Click **Get Object** to open the Oracle E-Business Suite 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*.

   In this example, click **OK** to continue the partner link creation without configuring flexfields for the selected interface table.

   Note: Configuring Flexfield Mapping Data for the Selected Open Interface Table

   If a selected interface table has pre-configured flexfield mappings available, a pop-up window appears allowing you to decide if you want to use these pre-configured mappings.

- Click **Yes** to display these mappings as part of the flexfield definitions in the Configure Flexfields region. You can modify them later if needed.

- Click **No** to indicate that you will not use these mappings and the pop-up window disappears.

If you want to configure flexfield mappings without using the pree-configured flexfield ones, you can create a new mapping through a flexfield wizard if desired. For information on how to configure flexfield data for the selected open interface table, see:

- Adding or Configuring a New Mapping, page 4-37

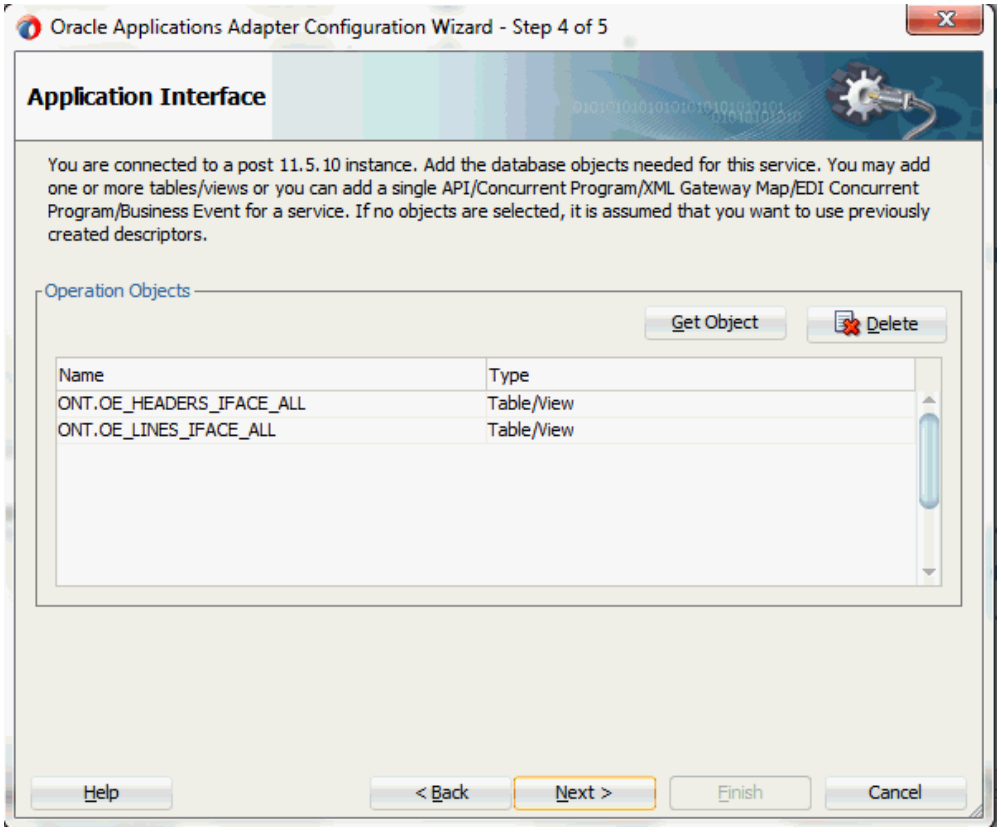
- Configuring Key Flexfield Mappings for an Open Interface Table, page 4-79

- Configuring Descriptive Flexfield Mappings for an Open Interface Table, page 4-85

- Importing an Existing Flexfield Mapping, page 4-115

The Application Interface page appears with the two selected tables.

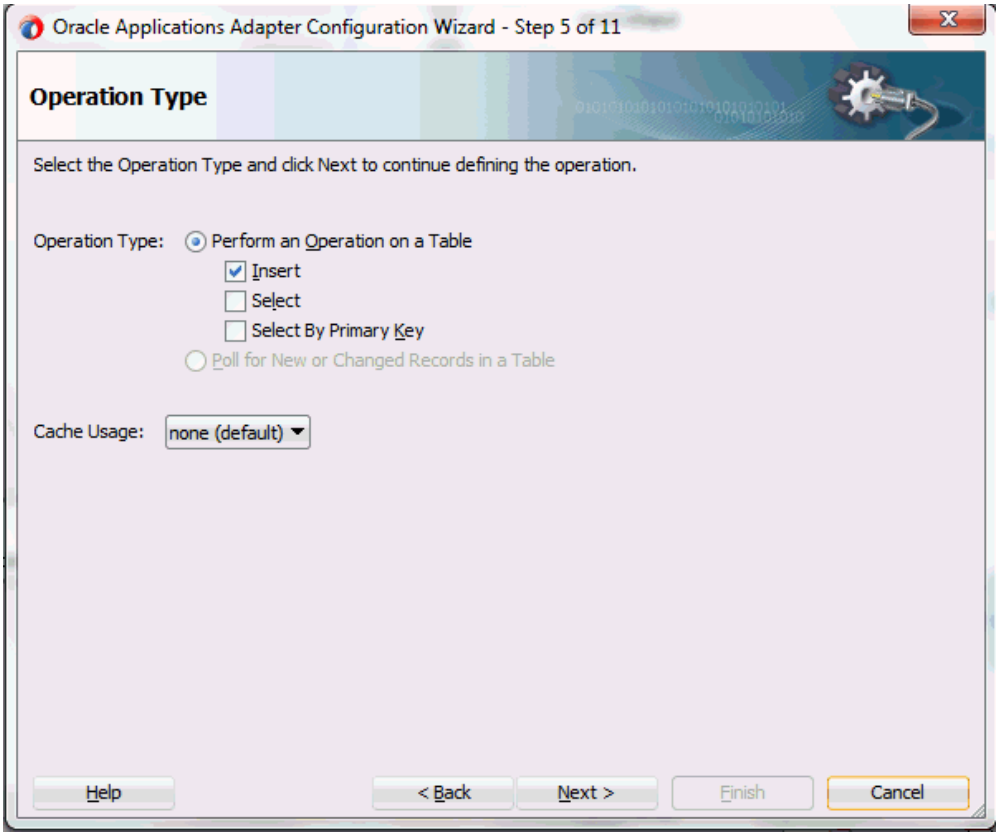
Adapter Configuration Wizard - Application Interface Page



Click **OK**.

8. Click **Next**. The Operation Type page is displayed.

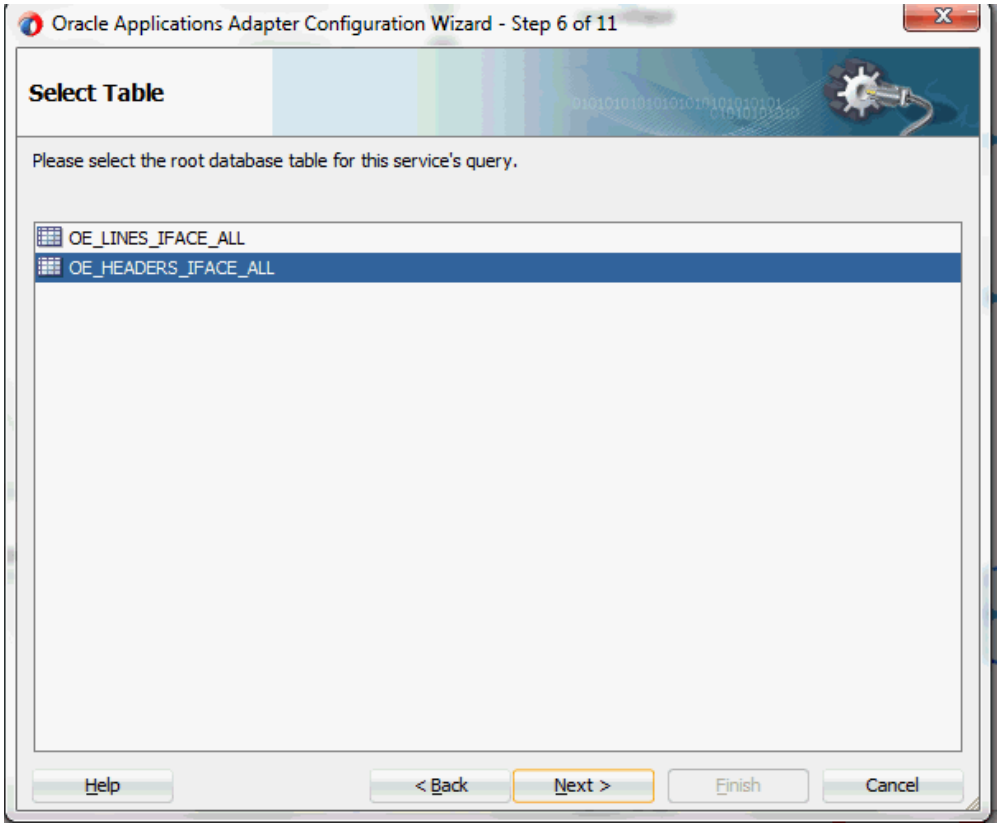
Adapter Configuration Wizard - Operation Type Page



Select the **Perform an Operation on a Table** radio button and then choose the **Insert** check box. Ensure that the **Select** check box is deselected.

9. Click **Next**. The Select Table page appears 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.

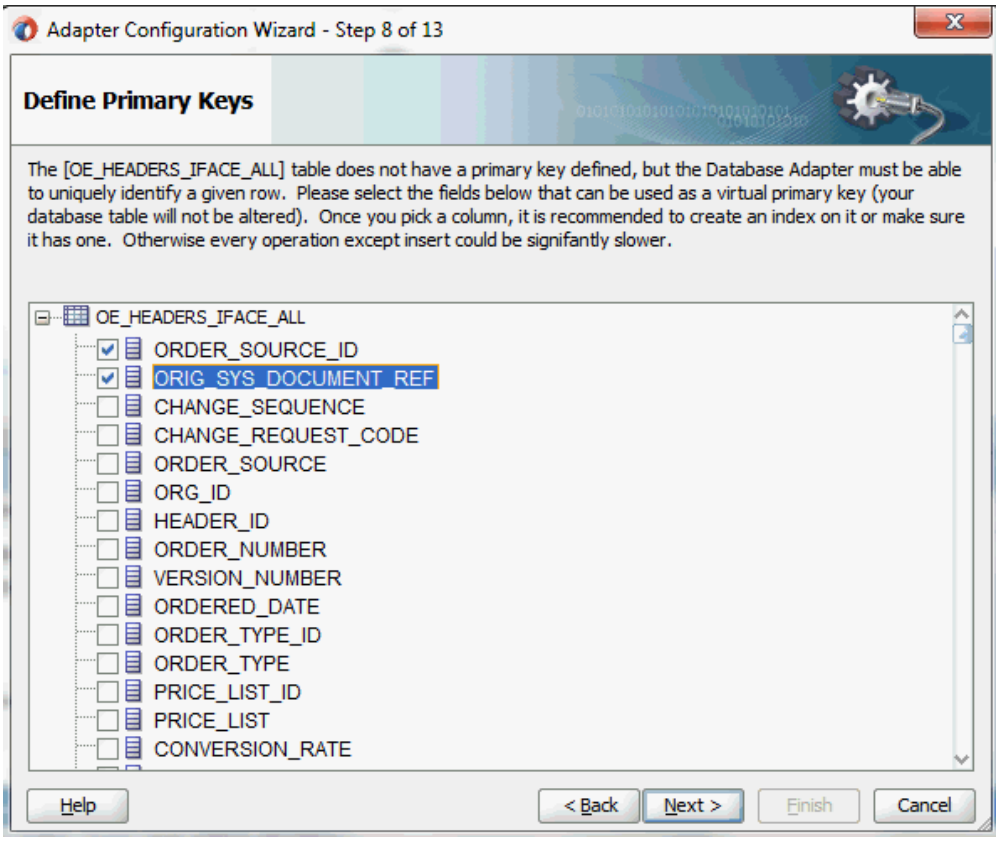
Adapter Configuration Wizard - Select Table Page



Select OE_HEADERS_IFACE_ALL as the root database table for this service's query.

10. Click **Next**. The Define Primary Keys page is displayed.

Adapter Configuration Wizard - Define Primary Keys Page for OE_HEADERS_IFACE_ALL



Select the following primary keys for the OE_HEADERS_IFACE_ALL table:

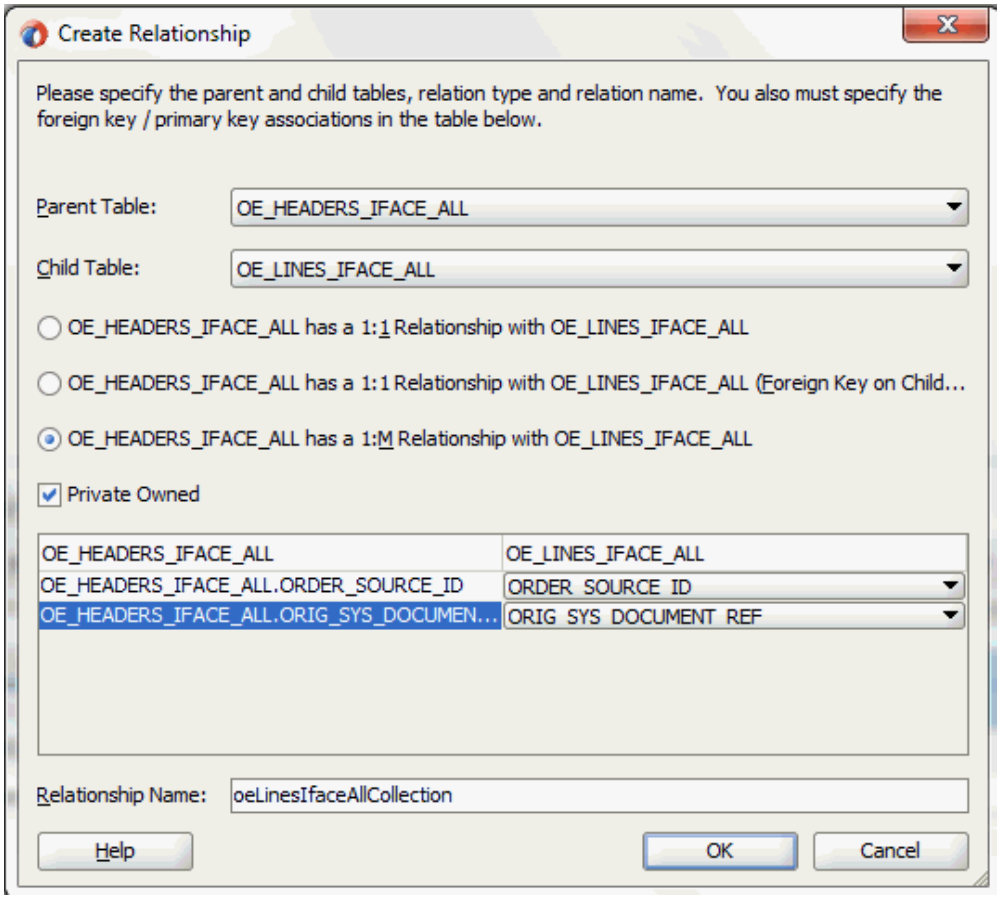
- ORDER_SOURCE_ID
- ORIG_SYS_DOCUMENT_REF

Click **Next**.

Select the same primary keys for the OE_LINES_IFACE_ALL table.

11. Click **Next**. The Relationships page appears. Click **Create** to open the Create Relationship dialog.

Defining Relationships



Enter the following information to define the relationship between the header and the detail table:

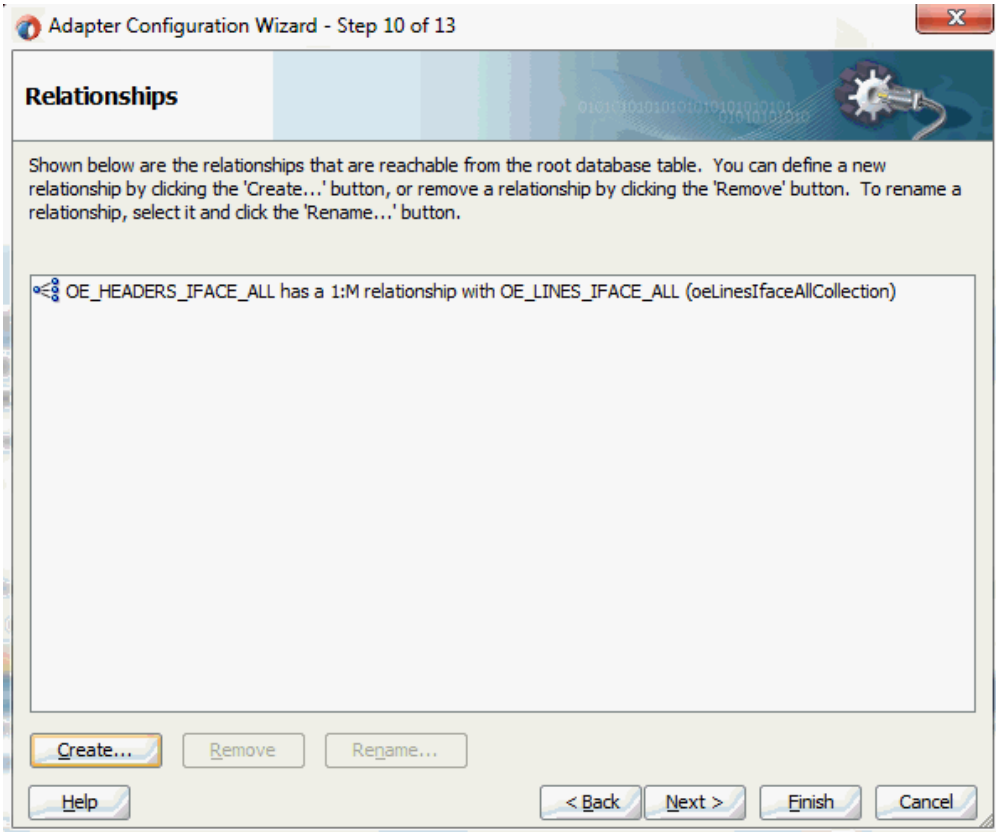
- Select the OE_HEADERS_IFACE_ALL as the parent table and OE_LINES_IFACE_ALL as the child table.
- Select the mapping type: OE_HEADERS_IFACE_ALL has a 1:M Relationship with OE_LINES_IFACE_ALL

> **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.

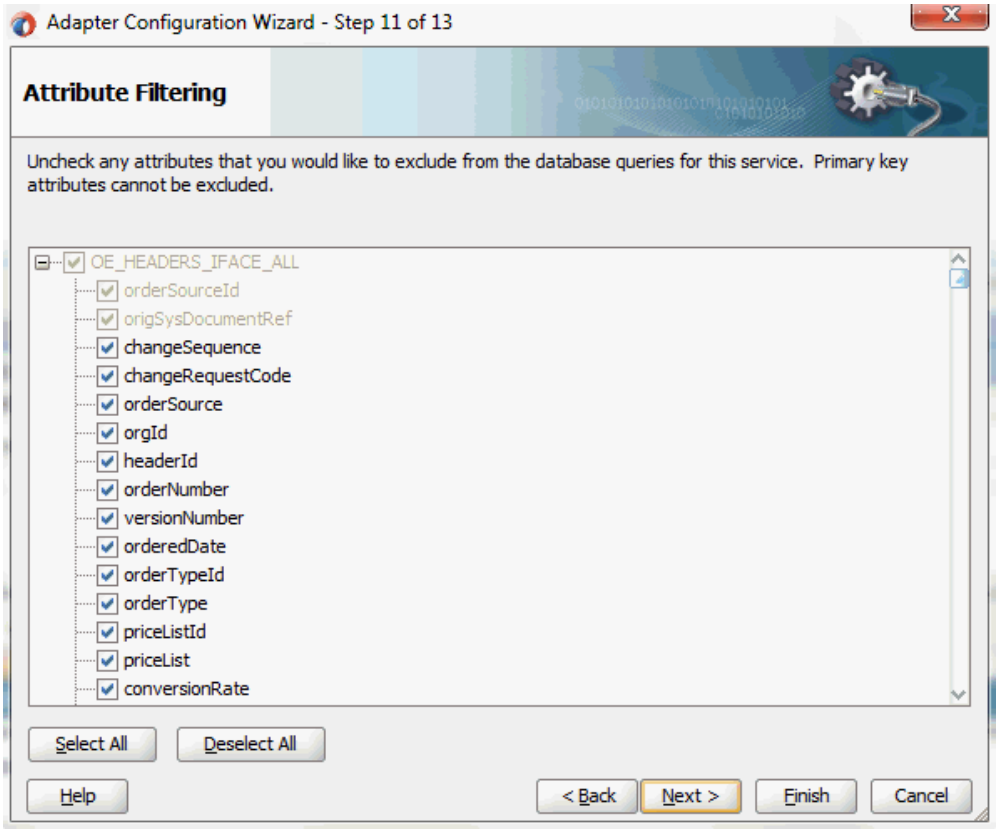
Click **OK**. The newly created relationship information appears in the Relationships page.

Adapter Configuration Wizard - Relationships Page



12. Click **Next**. The Attribute Filtering page appears. Leave default selections unchanged.

Adapter Configuration Wizard - Attribute Filtering Page



Click **Next**. The Advanced Options page appears.

Click **Next**.

13. Click **Finish**. The wizard generates the WSDL file corresponding to the selected interface. This WSDL file is now available for the partner link.

Adding a Partner Link for File Adapter

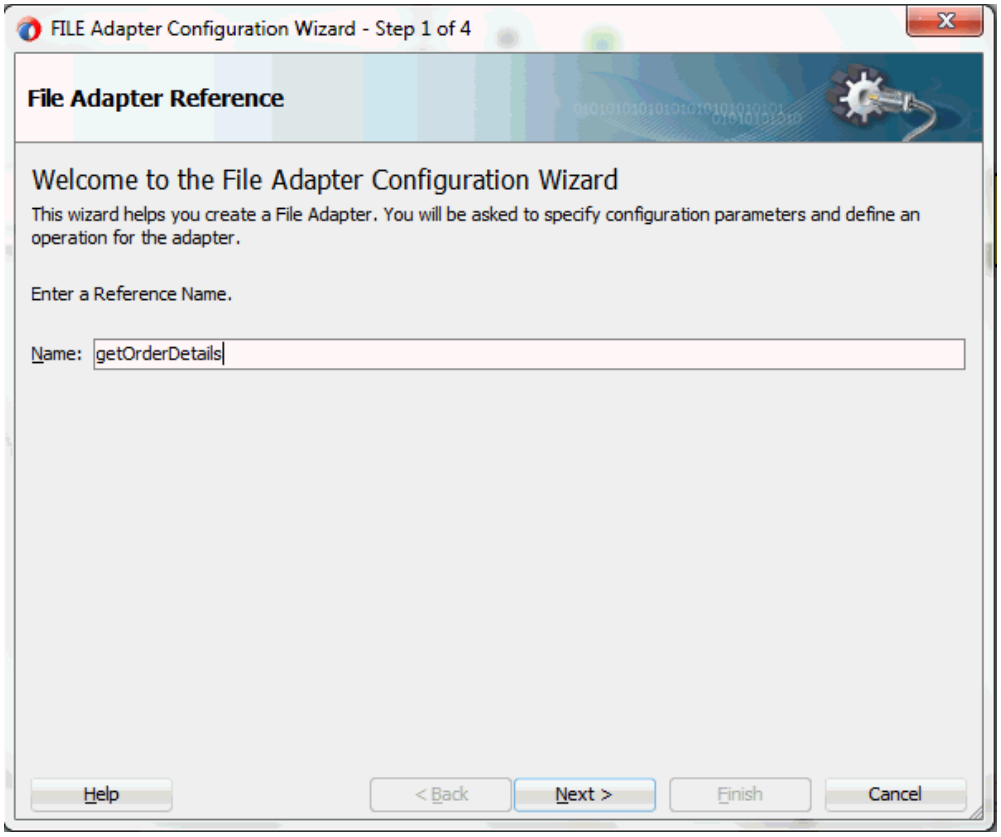
Use this step to configure a BPEL process by synchronously reading payload from an input file to obtain the purchase order details.

To add a Partner Link for File Adapter to read order details:

1. In Oracle JDeveloper BPEL Designer, click **BPEL Services** in the Component palette.

   Drag and drop **File Adapter** from the **BPEL Services** list into the right Partner Link swim lane before the first partner link `InsertOrder`. The File Adapter Reference page appears.

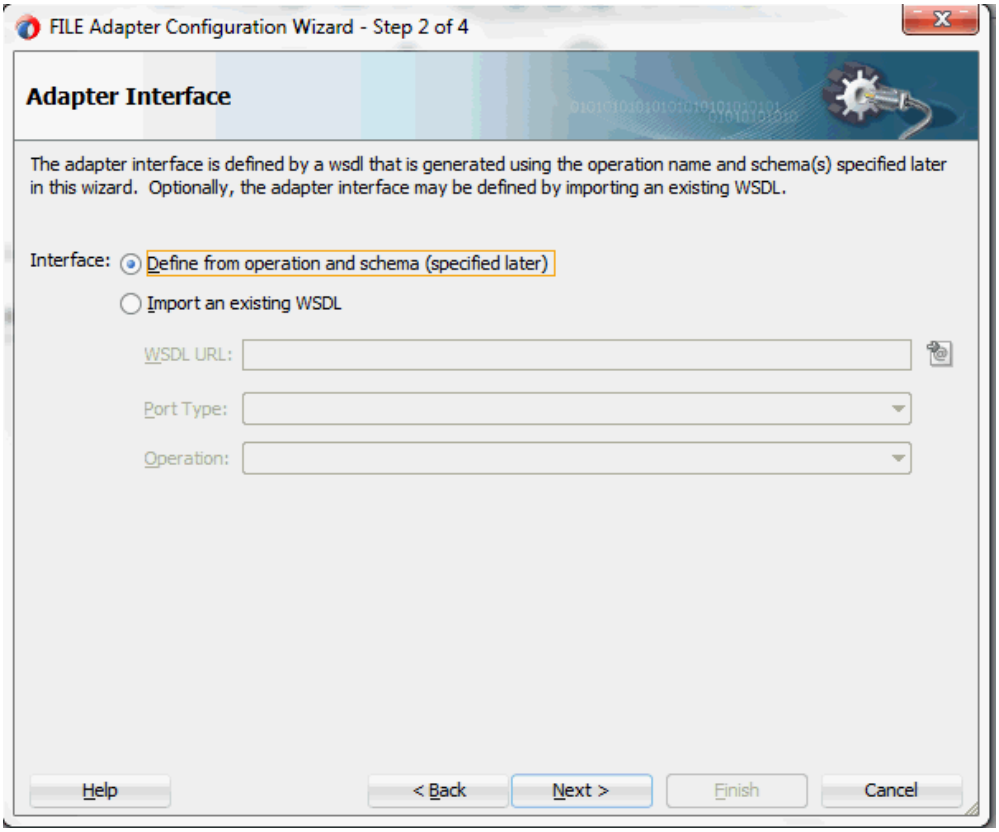
Specifying a File Reference Name



Enter a name for the File Adapter service, for example, `getOrderDetails`.

2. Click **Next**. The Adapter Interface page appears.

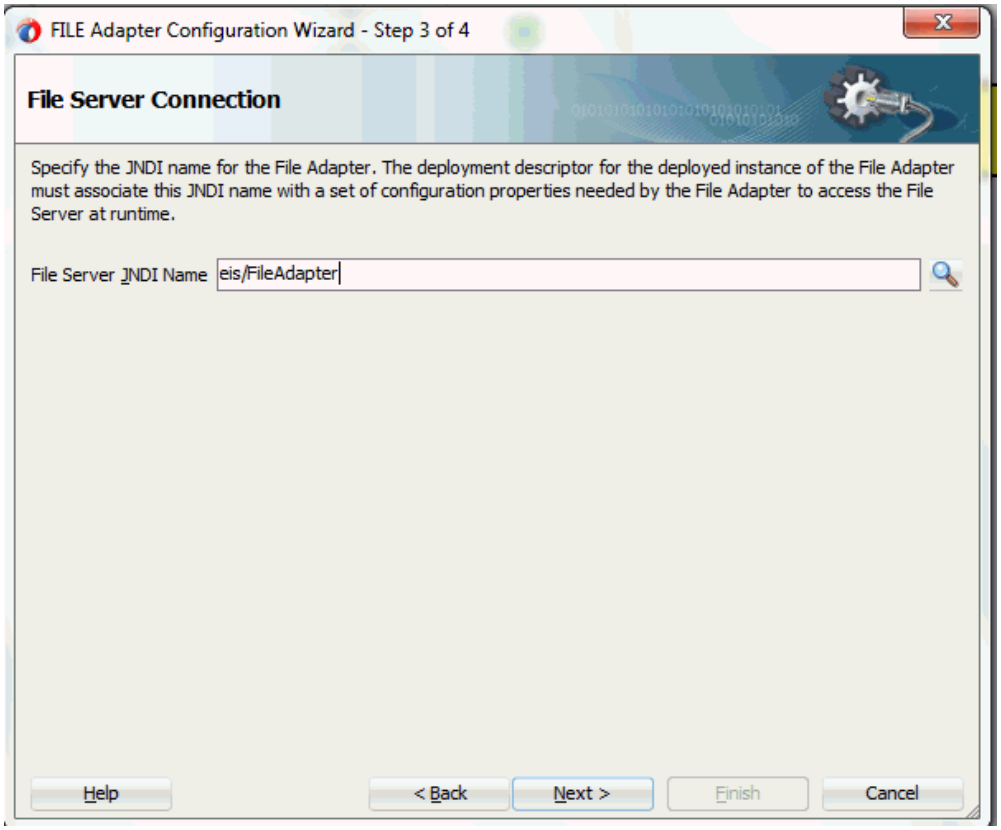
Specifying the Adapter Interface



Select the **Define from operation and schema (specified later)** radio button and click **Next**.

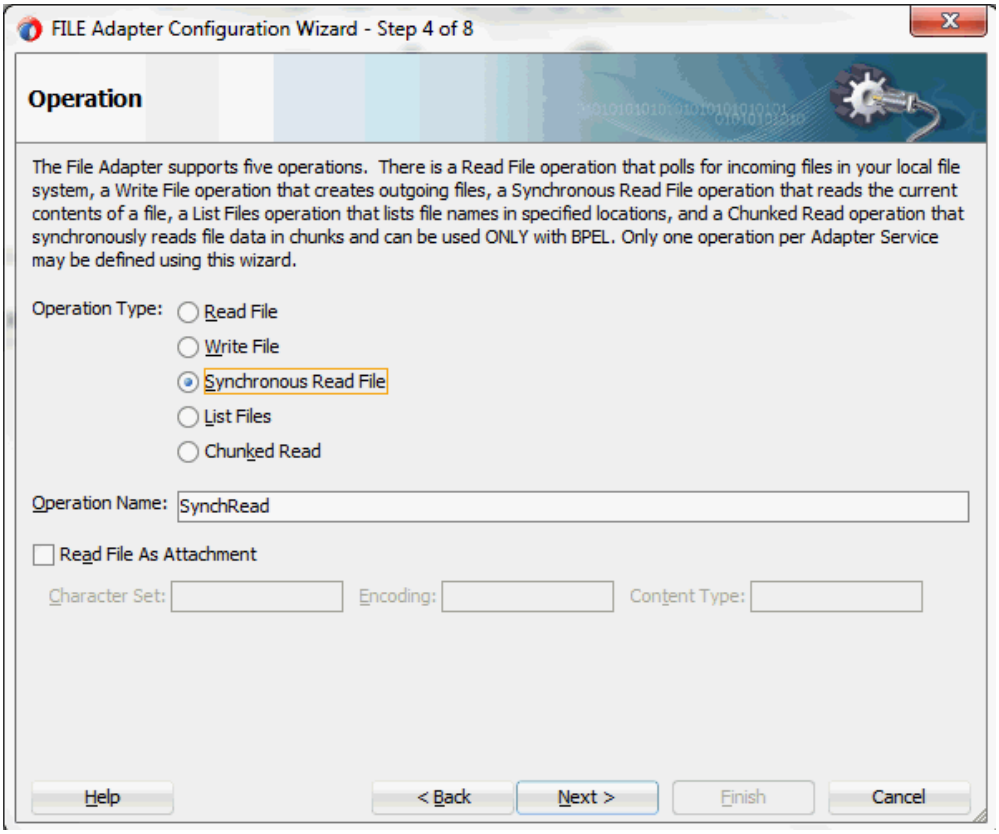
The File Server Connection page appears. Leave the default value unchanged for the File Server JNDI Name field.

Specifying File Server JNDI Name



3. In the Operation page, specify the operation type. For example, select the **Synchronous Read File** radio button. This automatically populates the **Operation Name** field.

Specifying the Operation

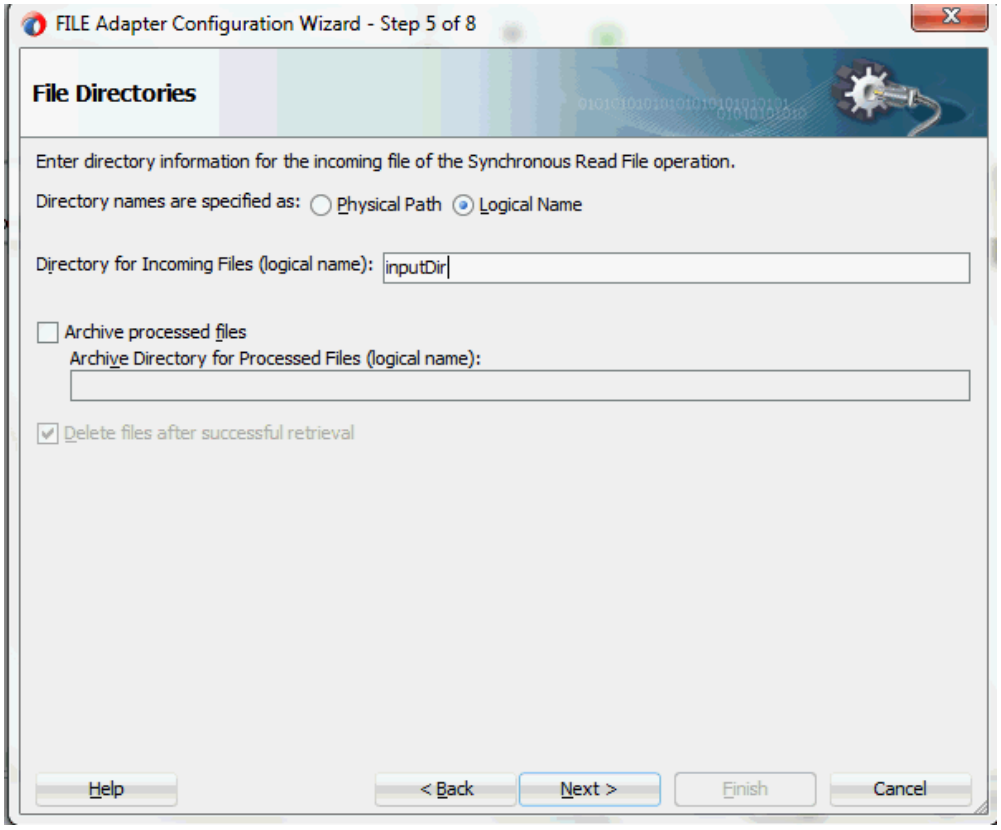


Click **Next** to access the File Directories page.

4. Select the **Logical Name** radio button and specify directory for incoming files, such as `inputDir`.

   Ensure the **Delete files after successful retrieval** check box is not selected.

Configuring the Input File



Click **Next** to open the File Name page.

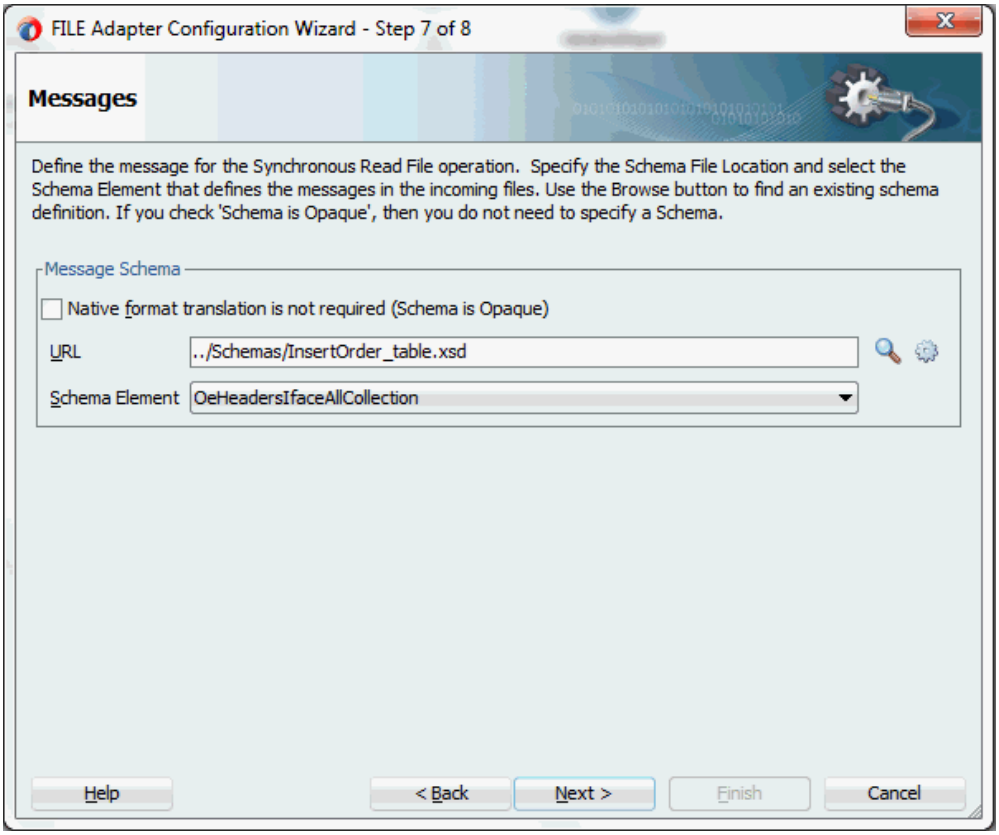
5. Enter the name of the file for the synchronous read file operation. For example, enter `order_data.xml`. Click **Next** to open the Messages page.

6. Select the 'browse for schema file' icon next to the URL field to open the Type Chooser.

   Click Type Explorer and select *Project Schema Files* > *InsertOrder_table.xsd* > *OeHeadersIfaceAllCollection*. Click **OK**.

   The selected schema information will be automatically populated in the URL and Schema Element fields.

Specifying Message Schema



7. Click **Next** and then **Finish**. This completes the configuration and create the partner link with the required WSDL settings for the File Adapter service. The wizard also generates the WSDL file corresponding to the partner link. The main Create Partner Link dialog box appears, specifying the new WSDL file `getOrderDetails.wsdl`.

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

Configuring Invoke Activities

After configuring partner links, you need to configure the following two Invoke activities:

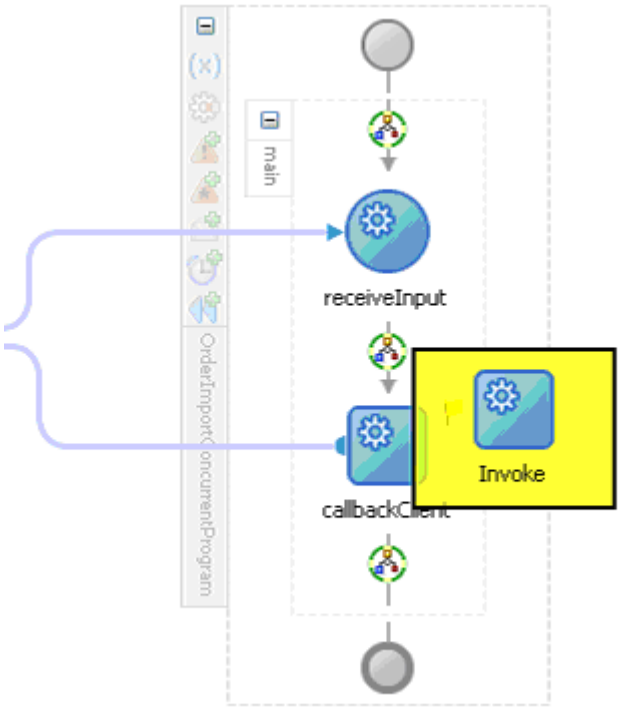
1. To get the acknowledgement data by invoking the `ViewAck` partner link.
2. To write acknowledgement data to an XML file by invoking `WriteAckdata` partner link for File Adapter.

To add the first Invoke activity for a partner link to get acknowledgement data:

1. In Oracle JDeveloper BPEL Designer, expand the **BPEL Constructs** from the

Component Palette. Drag and drop the first **Invoke** activity from the **Web Service** section into the center swim lane of the process diagram, between the **receiveInput** and **callbackClient** activities.

Adding an Invoke Activity



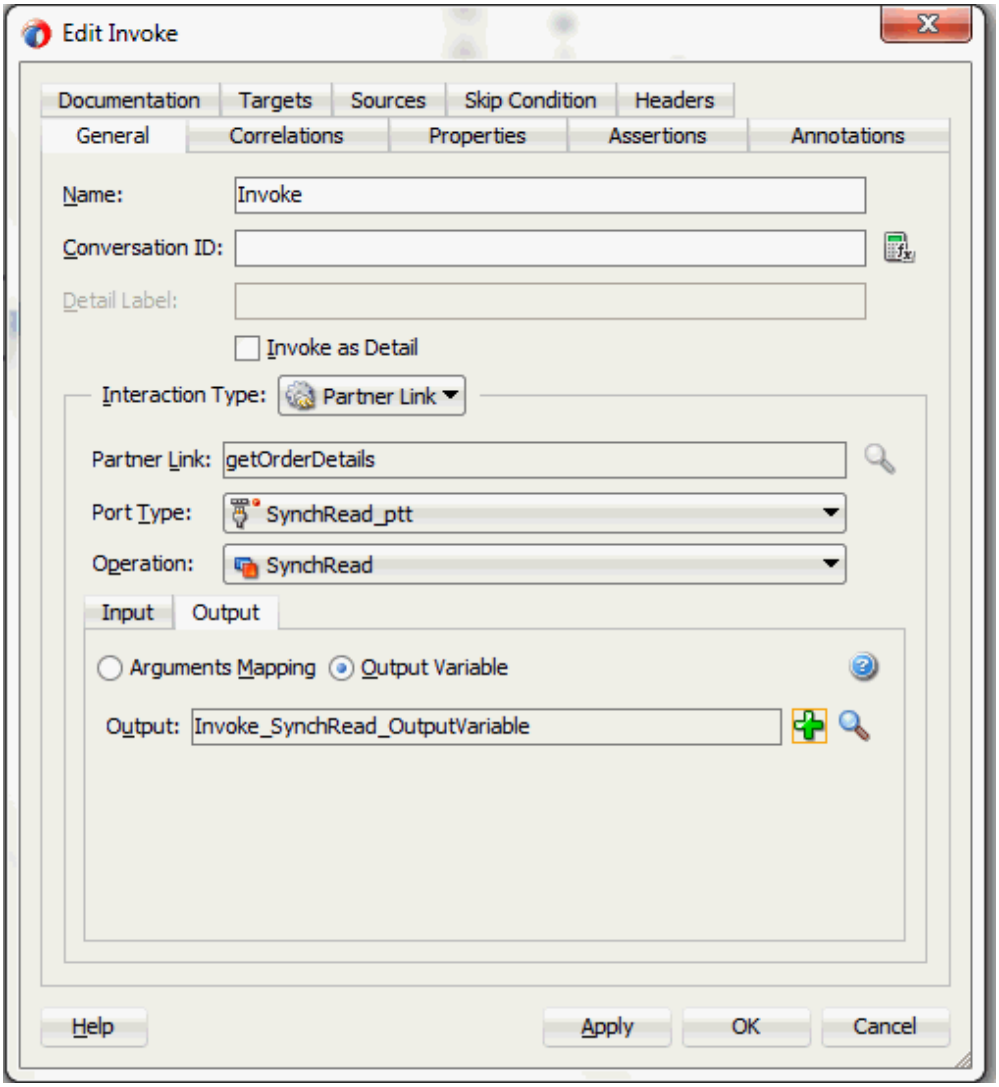
2. Link the Invoke activity to the `ViewAck` service. The Edit Invoke dialog appears.

3. Enter a name for the Invoke activity. In the Input tab, ensure the **Input Variable** radio button is selected. 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**, then enter a name for the variable. You can also accept the default name. Click **OK**.

5. In the Output tab, ensure the **Output Variable** radio button is selected. 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**, then enter a name for the variable. You can also accept the default name. Click **OK**.



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

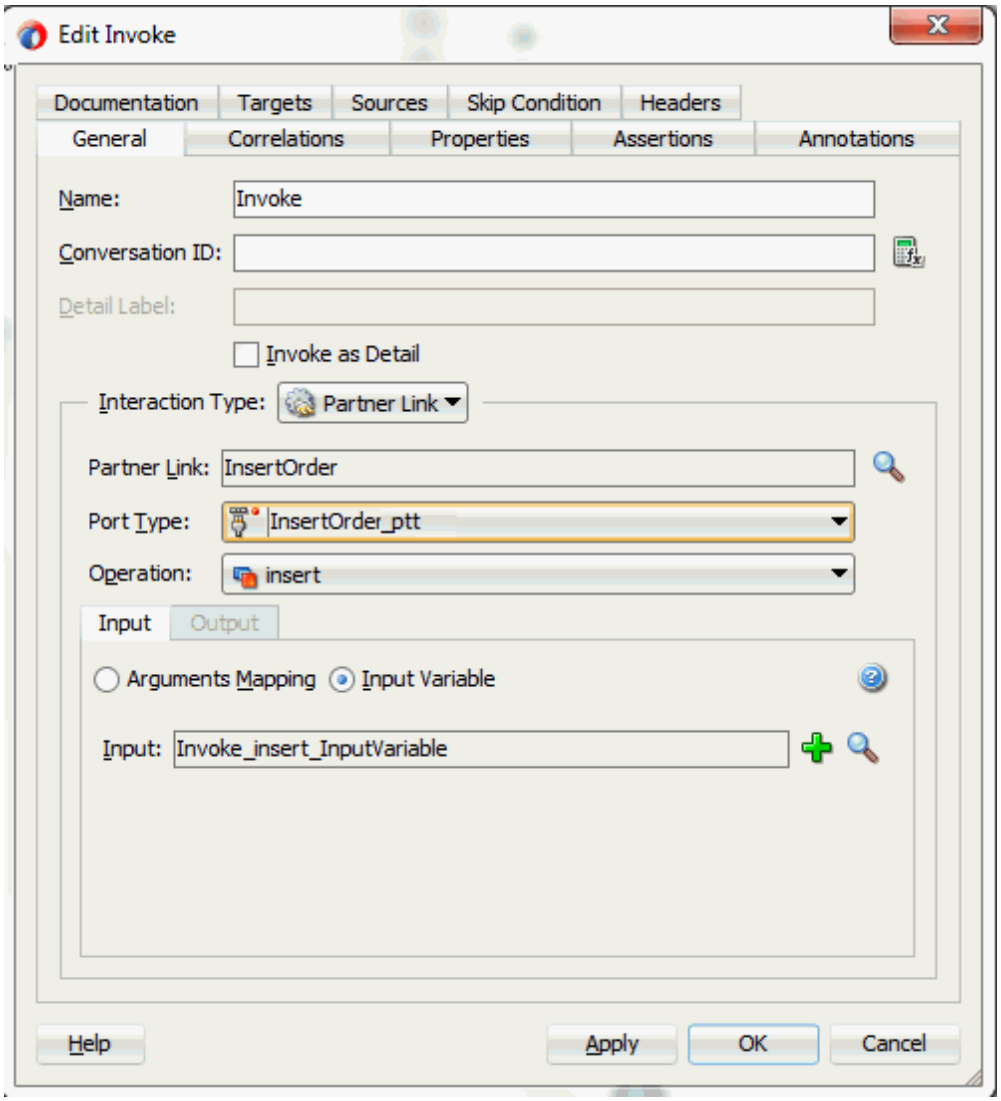
The Invoke activity appears in the process diagram.

To add the second Invoke activity for a partner link to insert order data into Open Interface tables:

1. In Oracle JDeveloper BPEL Designer, expand the **BPEL Constructs** from the Component Palette. Drag and drop the second **Invoke** activity from the **Web Service** section into the center swim lane of the process diagram, between the first **Invoke** and **callbackClient** activities.

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

3. Repeat Step 3 and Step 4 described in the first Invoke activity procedure. Click **Apply** and then **OK** in the Edit Invoke dialog box to finish configuring the second Invoke activity.



Configuring an Assign Activity

The next task is to add an Assign activity to the process map. This is used to pass the output of File Adapter's Synchronous Read (`getOrderDetails`) service as an input to the Open Interface`InsertOrder` service.

To add an Assign activity:

1. In JDeveloper BPEL Designer, expand the **BPEL Constructs** from the Component Palette.

Drag and drop the **Assign** activity from the **Basic Activities** section into the center swim lane of the process diagram, between the first and second **Invoke** activities that you just created earlier.

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

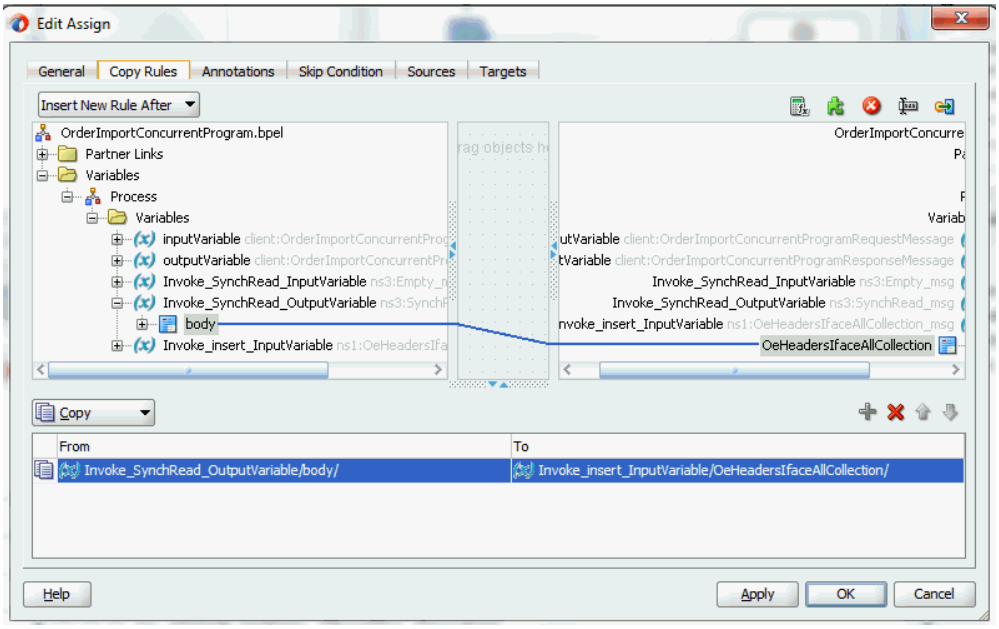
   Click the General tab to enter a name for the Assign activity. For example, `SetOrderDetails`.

3. Enter the following parameter information:

   - In the From navigation tree, navigate to **Variable > Process > Variables > Invoke_SynchRead_OutputVariable** and select **body**.

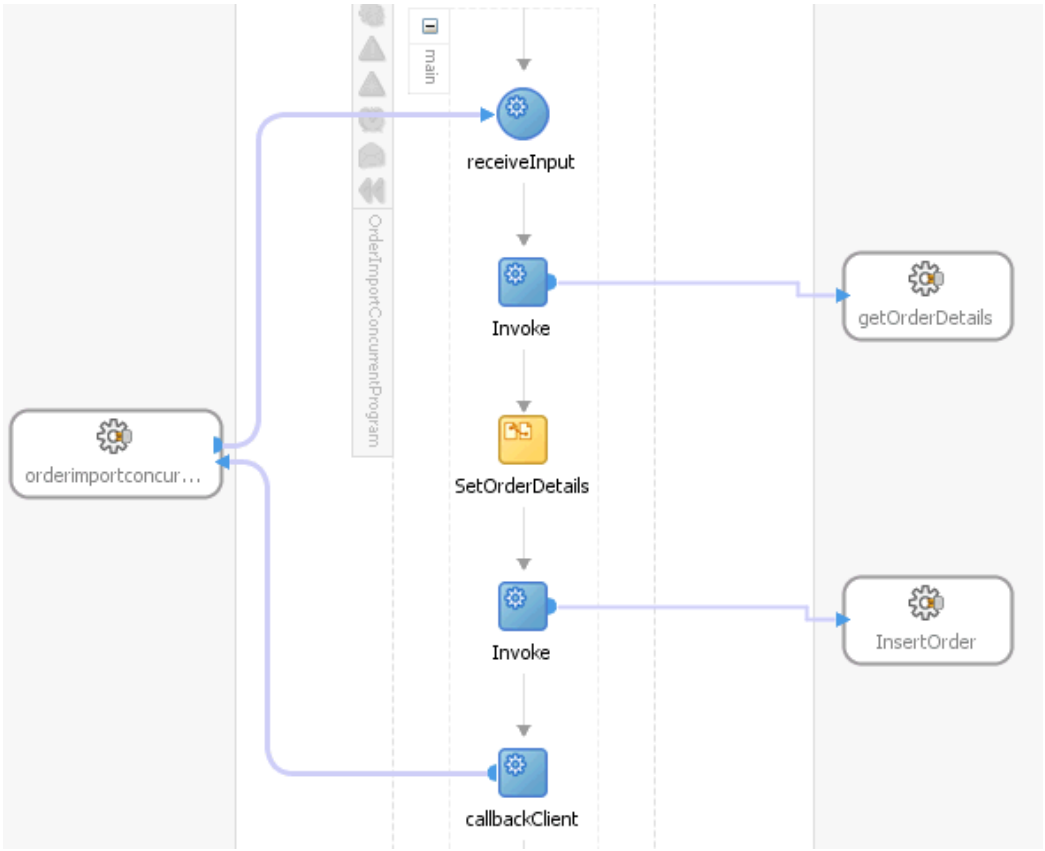
   - In the To navigation tree, navigate to **Variable > Process > Variables > Invoke_insert_InputVariable** and select **OeHeadersIfaceAllCollection**.

   Drag the source node (body) to connect to the target node (OeHeadersIfaceAllCollection) that you just specified. This creates a line that connects the source and target nodes. The copy rule is displayed in the From and To sections at the bottom of the Edit Assign dialog box.



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

Completed BPEL Process Diagram



Click the `InsertOrder` (composite.xml) to display the Oracle JDeveloper composite diagram:

> **Note:** Click the Source tab of `InsertOrder` (composite.xml) to enter a value for the physical directory `inputDir` for the reference `getOrderDetails` (such as `/usr/tmp`).
>
> ```
> <property name="inputDir" type="xs:string"
> many="false" override="may">/usr/tmp</property>
> ```

Run-Time Tasks for Interface Tables

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

1. Deploy the SOA Composite application with BPEL process, page 8-34
2. Test the deployed SOA Composite application with BPEL process, page 8-37

Deploying the SOA Composite Application with BPEL Process

To invoke the services from the BPEL client contained in the SOA composite, the SOA composite needs to be deployed to the Oracle WebLogic managed server. This can be achieved using Oracle JDeveloper. Once the composite is deployed, it can be tested from the Oracle Enterprise Manager Fusion Middleware Control Console.

Prerequisites

Before deploying the SOA composite with BPEL process using Oracle JDeveloper, you must have established the connectivity between the design-time environment and the run-time server. For more information, see Configuring the Data Source in Oracle WebLogic Server, page 11-1 and Creating an Application Server Connection, page 11-10.

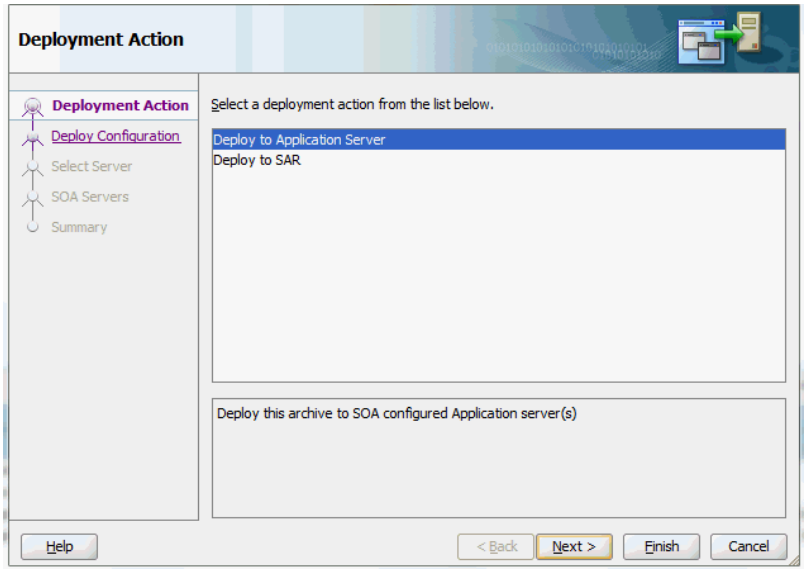
> **Note:** If a local instance of the WebLogic Server is used, start the WebLogic Server by selecting **Run > Start Server Instance** from Oracle JDeveloper. Once the WebLogic Admin Server "DefaultServer" instance is successfully started, the <Server started in Running mode> and DefaultServer started message in the Running:DefaultServer and Messages logs should appear.

To deploy the SOA Composite application with BPEL process:

1. Select the SOA Composite project in the Applications Navigator.
2. Right-click the project name, and then select **Deploy > [project name] > [serverConnection]** from the menu that appears.

   For example, you can select **Deploy > InsertOrder > soa-server1** to deploy the process if you have the connection set up appropriately.

   > **Note:** If this is the first time to set up server connection, then the Deployment Action window appears. Select 'Deploy to Application Server' and click **Next**.

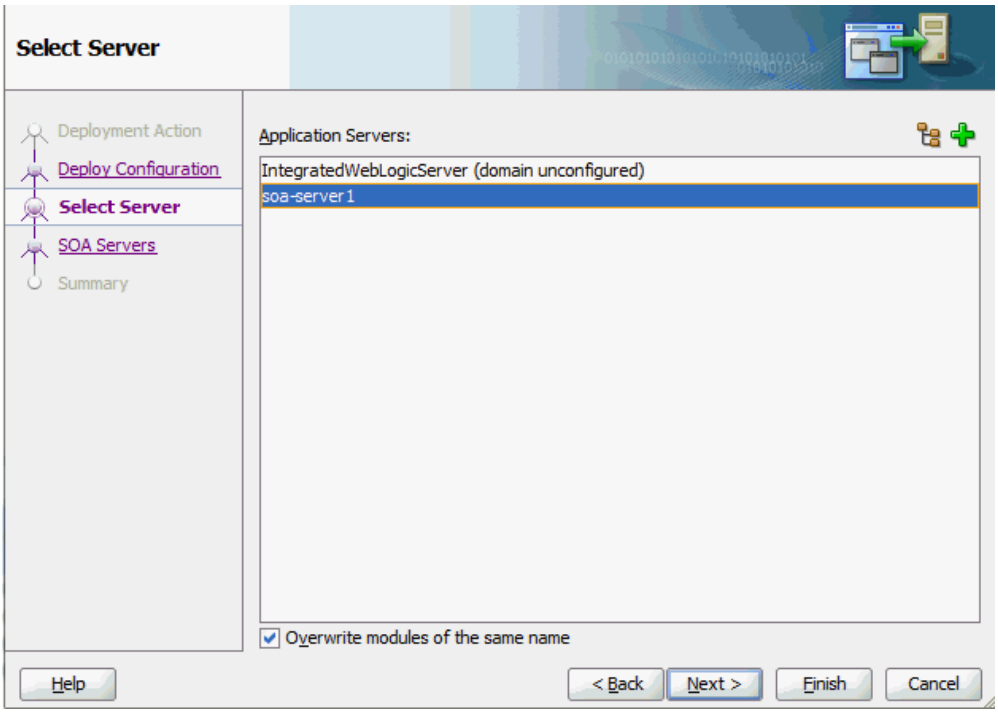


In the Deploy Configuration window, ensure the following information is selected before clicking **Next** to add a new application server:

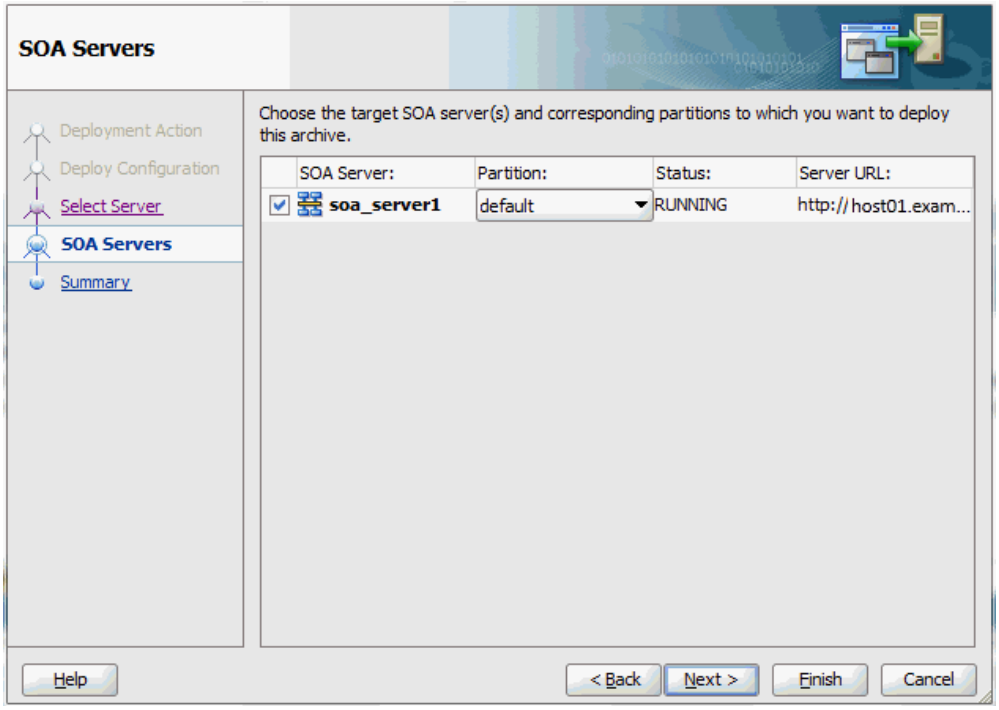
- New Revision ID: 1.0
- Mark composite revision as default: Select this check box.
- Overwrite any existing composites with the same revision ID: Select this check box.

The steps to create a new Oracle WebLogic Server connection from JDeveloper are covered in Creating an Application Server Connection, page 11-10.

3. In the Select Server page, select 'soa-server1' that you have established the server connection earlier. Click **Next**.



4. In the SOA Servers page, accept the default target SOA Server ('soa-server1') selection.



Click **Next** and **Finish**.

If you are deploying the composite for the first time from your Oracle JDeveloper session, the Authorization Request window appears. Enter username and password information specified during Oracle SOA Suite installation. Click **OK**.

5. Deployment processing starts. Monitor deployment progress and check for successful compilation in the SOA - Log window as well as in the Deployment - Log window.

Testing the Deployed SOA Composite Application with BPEL Process

Once the SOA Composite application with BPEL process is deployed, you can manage and monitor the process from the Oracle Enterprise Manager Fusion Middleware Control Console. You can also test the process and the integration interface by manually initiating the process.

To manually initiate and monitor the deployed SOA Composite application with BPEL process:

1. Navigate to Oracle Enterprise Manager Fusion Middleware Control Console (`http://<servername>:<portnumber>/em`). The composite you deployed is displayed in the Applications Navigation tree.

2. Enter username (such as `weblogic`) and password and click **Login** to log in to a farm.

   You may need to select an appropriate target instance farm if there are multiple target Oracle Enterprise Manager Fusion Middleware Control Console farms.

3. From the Farm base domain, expand the **SOA > soa-infra > soa-infra (soa_server1) > default** to access your deployed SOA composite applications running in the SOA Infrastructure for that managed server.

   > **Note:** The Farm menu always displays at the top of the navigator. As you expand the SOA folder in the navigator and click the links displayed beneath it, the SOA Infrastructure menu becomes available at the top of the page.

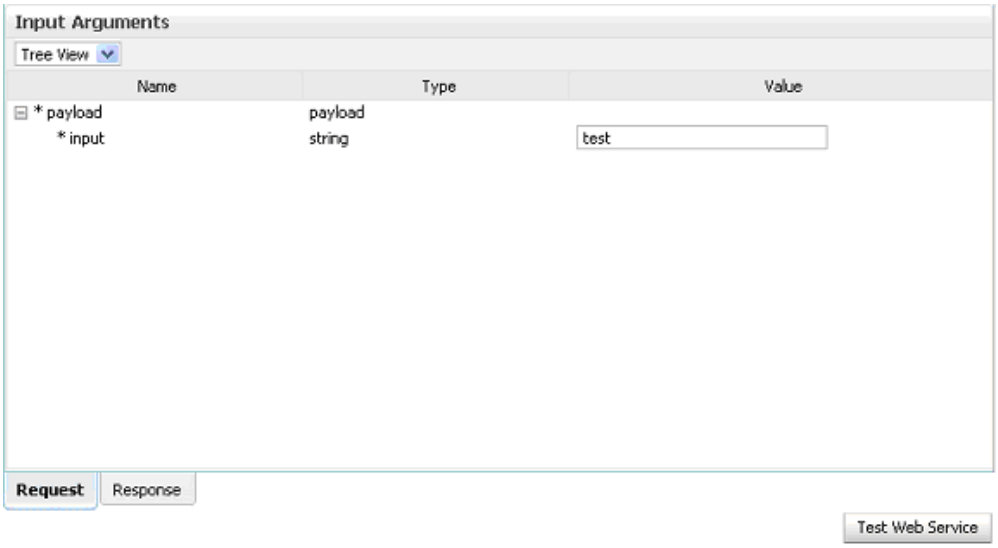
   Click the SOA Composite application that you want to initiate (such as 'OrderInsert') from the SOA Infrastructure.

   Click **Test** at the top of the page.

4. The Test Web Service page for initiating an instance appears. You can specify the XML payload data to use in the Input Arguments section.

   Enter the input string required by the process and click **Test Web Service** to initiate the process.

Testing Web Service



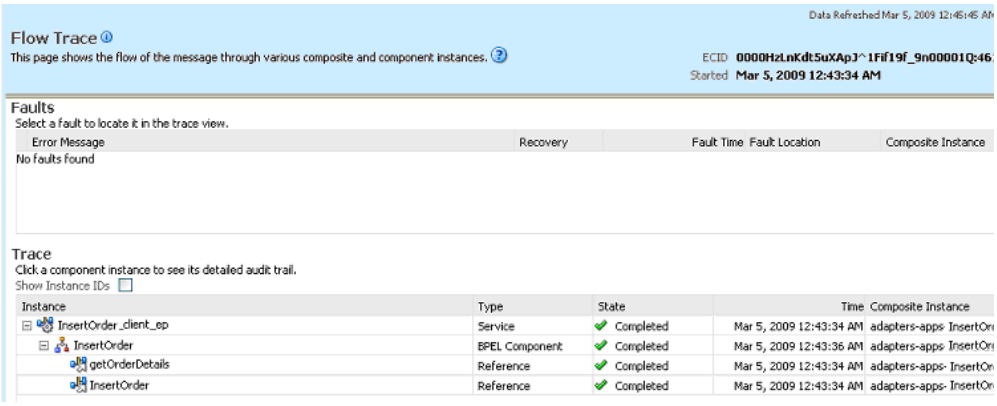
The test results appear in the Response tab upon completion.

5. Click **Launch Flow Trace** in the Test Web Service page. The Flow Trace page is displayed.

   In the Trace section, you should find the sequence of the message flow for the service binding component (`insertorder_client_ep`), BPEL component (`InsertOrder`), and reference binding components (`getOrderDetails` and `InsertOrder`). All involved components have successfully received and processed messages.

   If any error occurred during the test, you should find it in the Faults section.

Flow Trace Page



6. Click your BPEL service component instance link (such as `InsertOrder`) to display the Instances page where you can view execution details for the BPEL activities in the Audit Trail tab.

   Click the Flow tab to check the BPEL process flow diagram. Click an activity of the process diagram to view the activity details and flow of the payload through the process.

7. **Validating Records Using SQL**

   To confirm that the records have been inserted into the selected Open Interface tables, you can write SQL `SELECT` statements and fetch the results showing the latest records inserted into the open interface tables.

Design-Time Tasks for Views

This section describes the steps to configure Oracle E-Business Suite Adapter using the Adapter Configuration Wizard in Oracle JDeveloper.

SOA Composite Application with BPEL Process Scenario

In this example, Oracle E-Business Suite Adapter retrieves purchase order acknowledgement information from Interface Views `OE_HEADER_ACKS_V` and `OE_LINE_ACKS_V`. The acknowledgement information will then be written to an XML file as an output for confirmation.

When the SOA Composite application with BPEL process has been successfully executed after deployment, you can validate the output XML file with the same order book reference ID once an order is approved.

Prerequisites to Configure Views

You need to define a uniqueness criteria, which could be a single key or composite keys.

SOA Composite Application with BPEL Process Flow

Based on the scenario, the following design-time tasks are discussed in this chapter.

1. Create a new SOA Composite application with BPEL process, page 8-41
2. Add a partner link, page 8-46
3. Add a partner link for File Adapter, page 8-58
4. Configure the Invoke activity, page 8-64
5. Configure the Assign activity, page 8-68

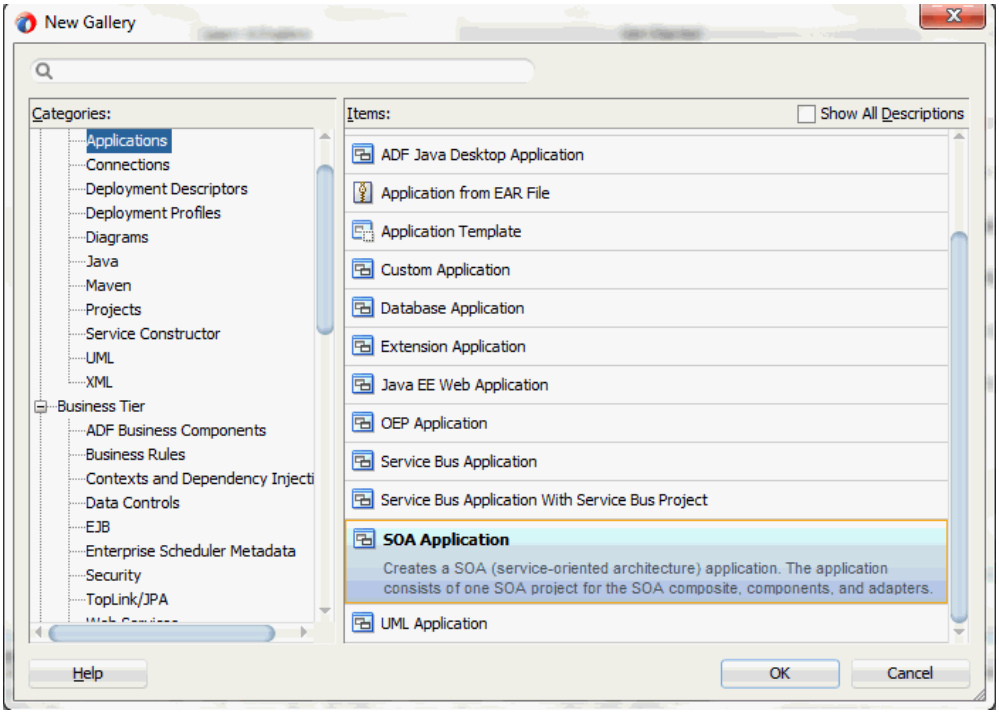
Creating a New SOA Composite Application with BPEL Process

To create a new SOA Composite application with BPEL process:

1. Launch Oracle JDeveloper.
2. Click **New Application** in the Application Navigator.

   The New Gallery page appears. Select **SOA Application** from the Items list.

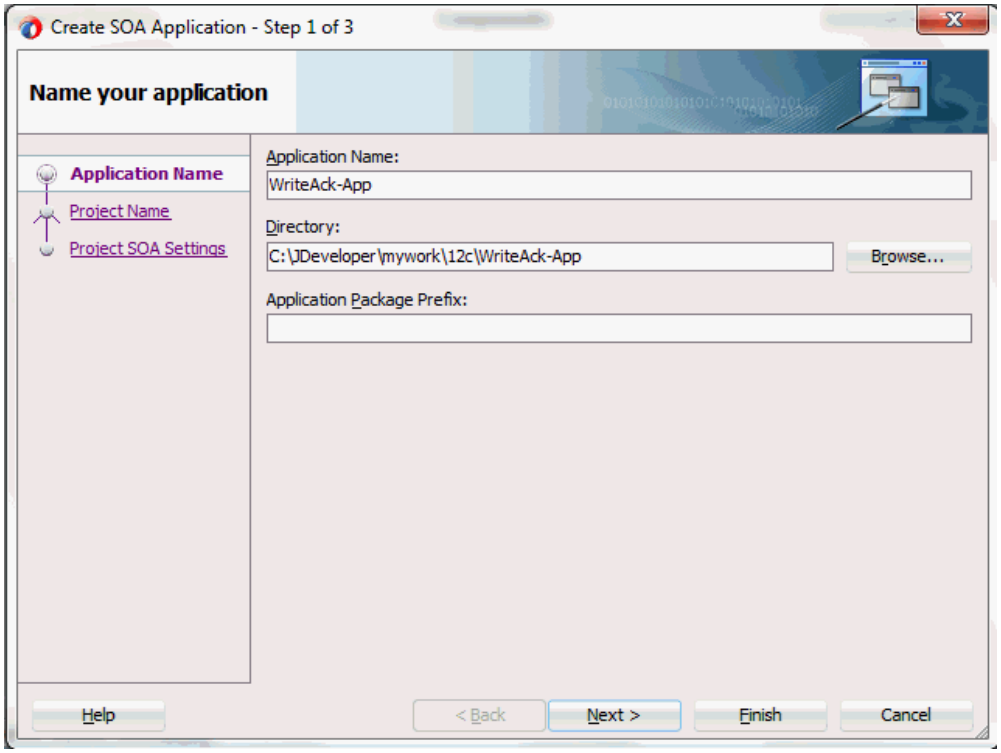
The New Gallery Page



Click **OK**.

3. The Create SOA Application - Name your application page is displayed.

The Create SOA Application - Name your application Page

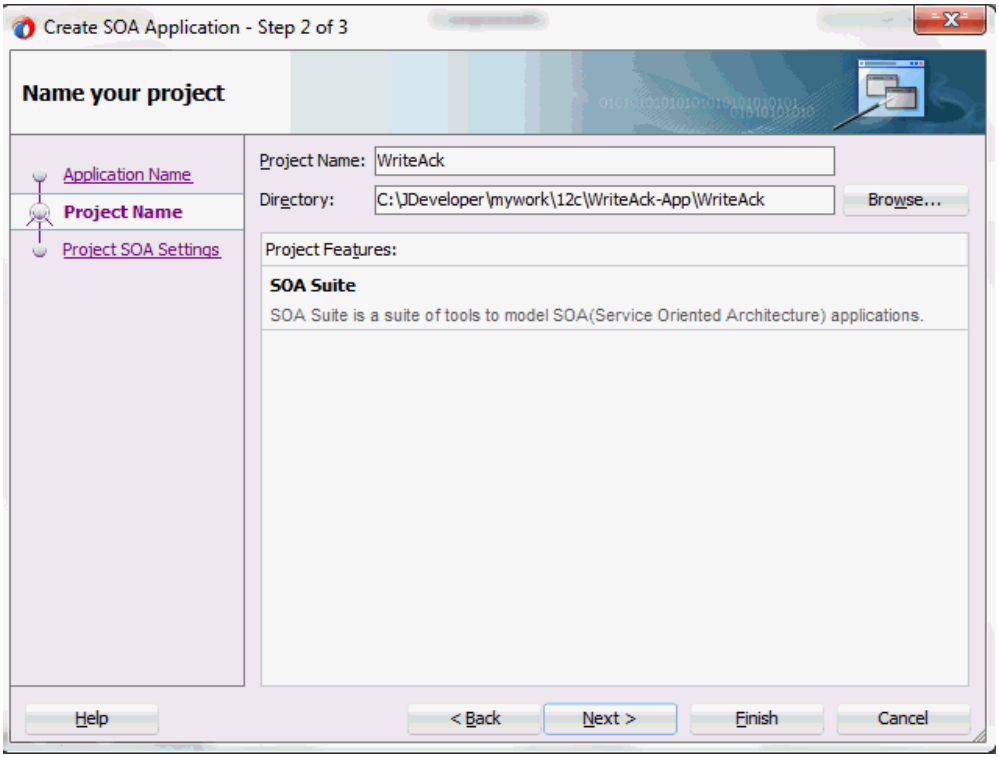


Enter an appropriate name for the application in the **Application Name** field.

Click **Next**. The Create SOA Application - Name your project page is displayed.

4. Enter an appropriate name for the project in the **Project Name** field. For example, `WriteAck`.

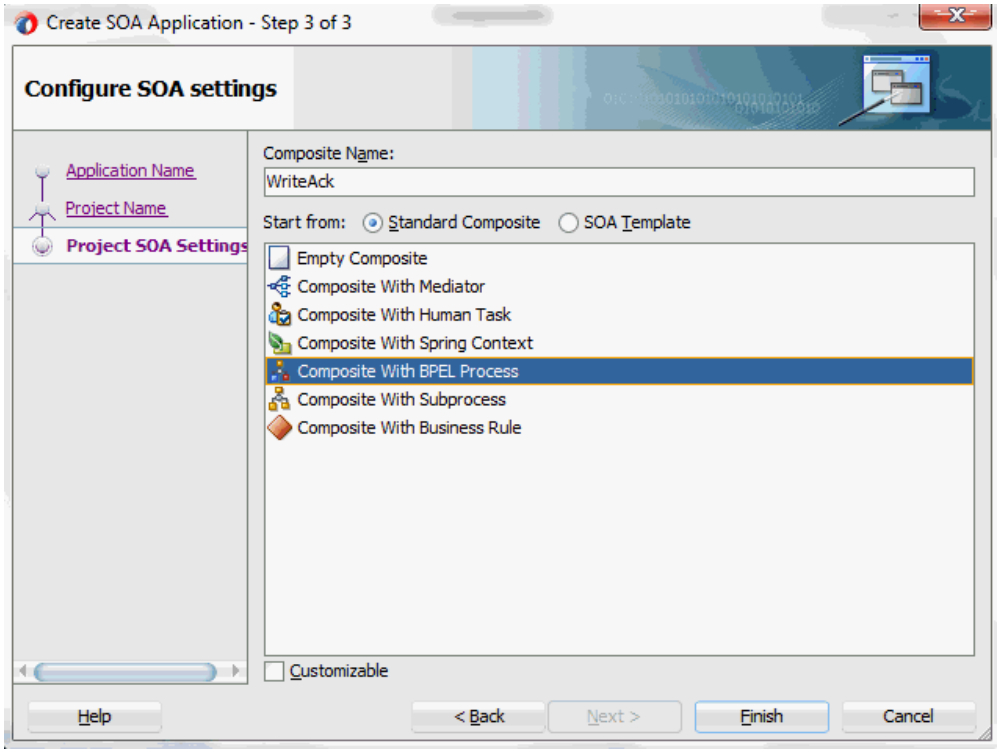
The Create SOA Application - Name your project Page



Notice that **SOA Suite** is shown in the Project Features section.

5. Click **Next**. The Create SOA Application - Configure SOA settings page is displayed.

The Create SOA Application - Configure SOA settings Page



6. Leave the default **Standard Composite** selection unchanged for the Start from field.

   Select **Composite With BPEL Process** from the Standard Composite list, and then click **Finish**. You have created a new application, and a SOA project. This automatically creates a SOA Composite.

   The Create BPEL Process page is displayed.

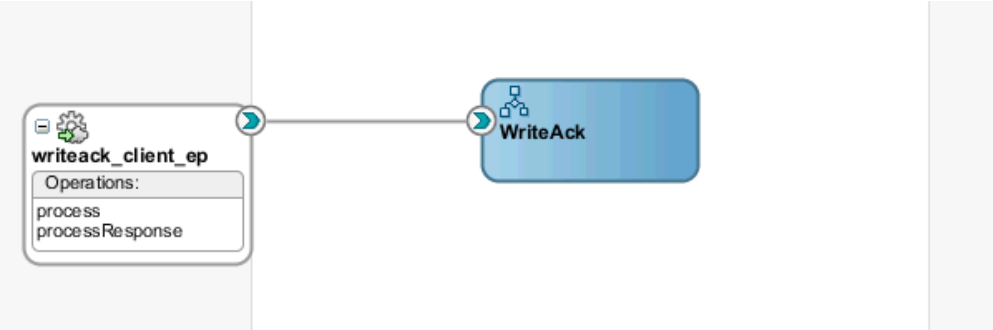
7. Leave the default **BPEL 2.0 Specification** selection unchanged. This creates a BPEL project that supports the BPEL 2.0 specification.

   Enter an appropriate name for the BPEL process in the **Name** field. For example, `WriteAck`.

   Select **Asynchronous BPEL Process** in the **Template** field. Click **OK**.

   An asynchronous BPEL process is created with the Receive and Reply activities. The required source files including bpel and wsdl, using the name you specified (for example, `WriteAck.bpel` and `WriteAck.wsdl`) and `WriteAck.bpel` (composite.xml) are also generated.

8. Navigate to SOA folder and click the `WriteAck.bpel` (composite.xml) to view the composite diagram.



Double click on the `WriteAck`component to open the BPEL process.

Adding a Partner Link

A BPEL partner link defines the link name, type, and the role of the BPEL process that interacts with the partner service.

In the scenario described earlier, you need to add a partner link to retrieve order acknowledgement information from selected Open Interface Views.

To add a partner link:

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

   Drag and drop **Oracle E-Business Suite** (formerly known as **Oracle Applications**) from the **BPEL Services** list into the right Partner Link swim lane of the process diagram. The Oracle E-Business Suite Reference page appears.

   Enter a reference name in the **Name** field. For example, `ViewAck`.

2. 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 **Create a New Database Connection** icon.

     How to define a new database connection, see Create a New Database Connection, page 5-14.

   - 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 Oracle E-Business Suite Adapter 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 Fusion Middleware User's Guide for Technology Adapters.*

4. Click **Next** in the Service Connection dialog box. You can add an interface view by browsing through the list of open interface views available in Oracle E-Business Suite.

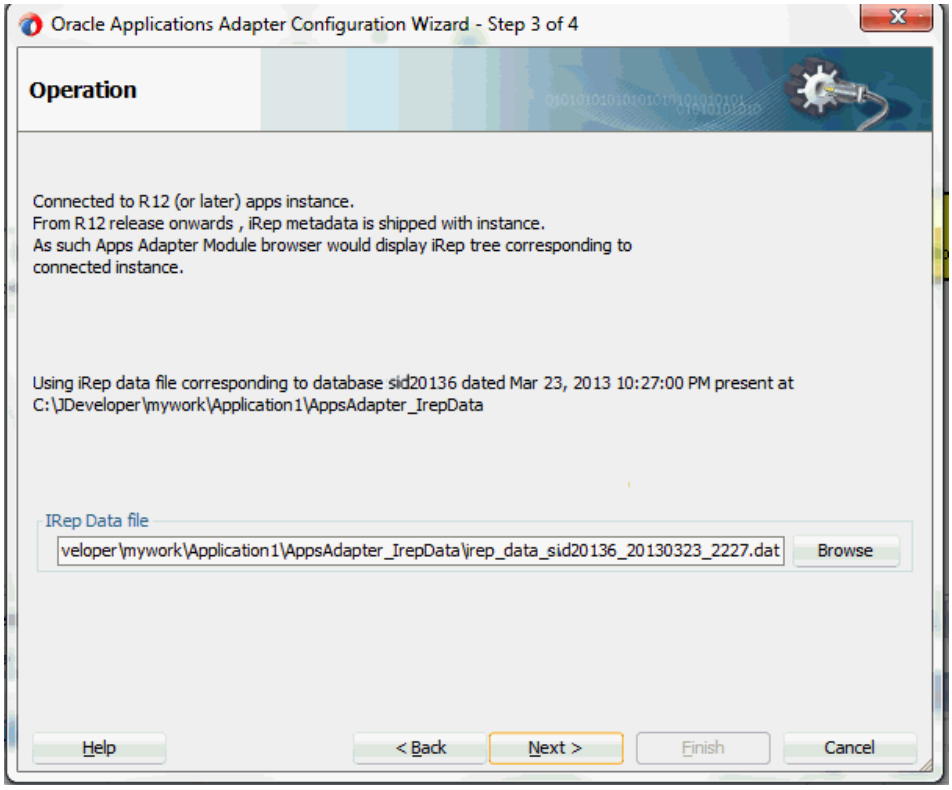
   If you are connecting to Oracle E-Business Suite Release 12, then the **IREP File not present** dialog box appears indicating that Adapter could not find the Oracle Integration Repository data file corresponding to the database you are connecting in your workspace. Absence of the data file would make 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 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.

     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



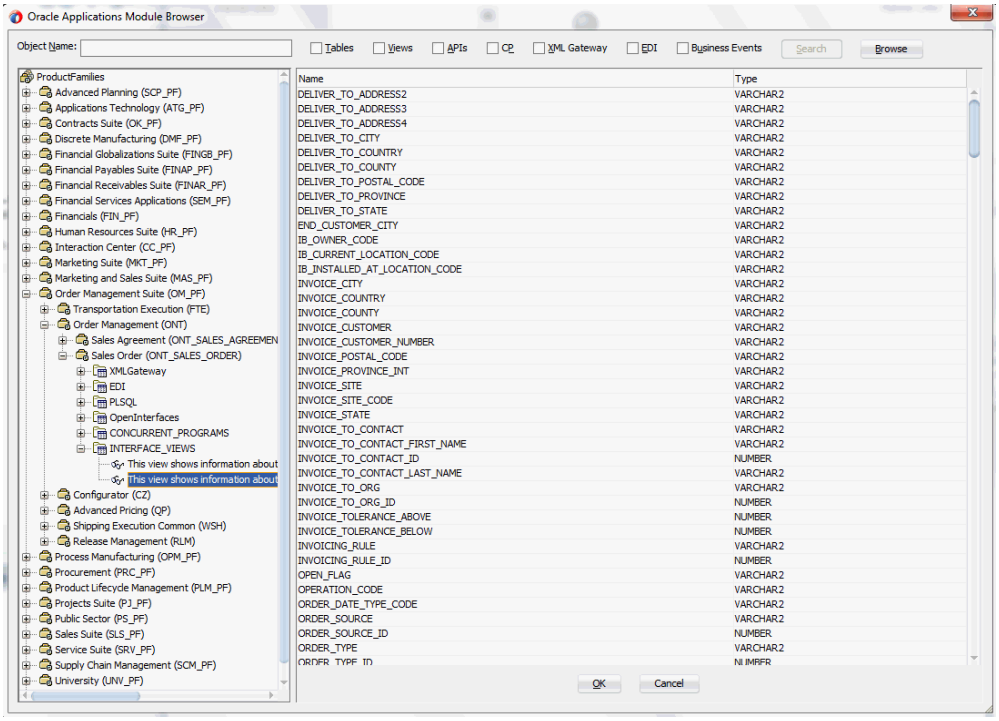
- Click **No** to query the Integration Repository data file from the live database you are connecting to display the Integration Repository tree.

In this example, click **No**. Click **Next** in the Operation page to open the Oracle E-Business Suite Module Browser.

5. Oracle E-Business Suite Module Browser includes the various product families that are available in Oracle E-Business Suite.

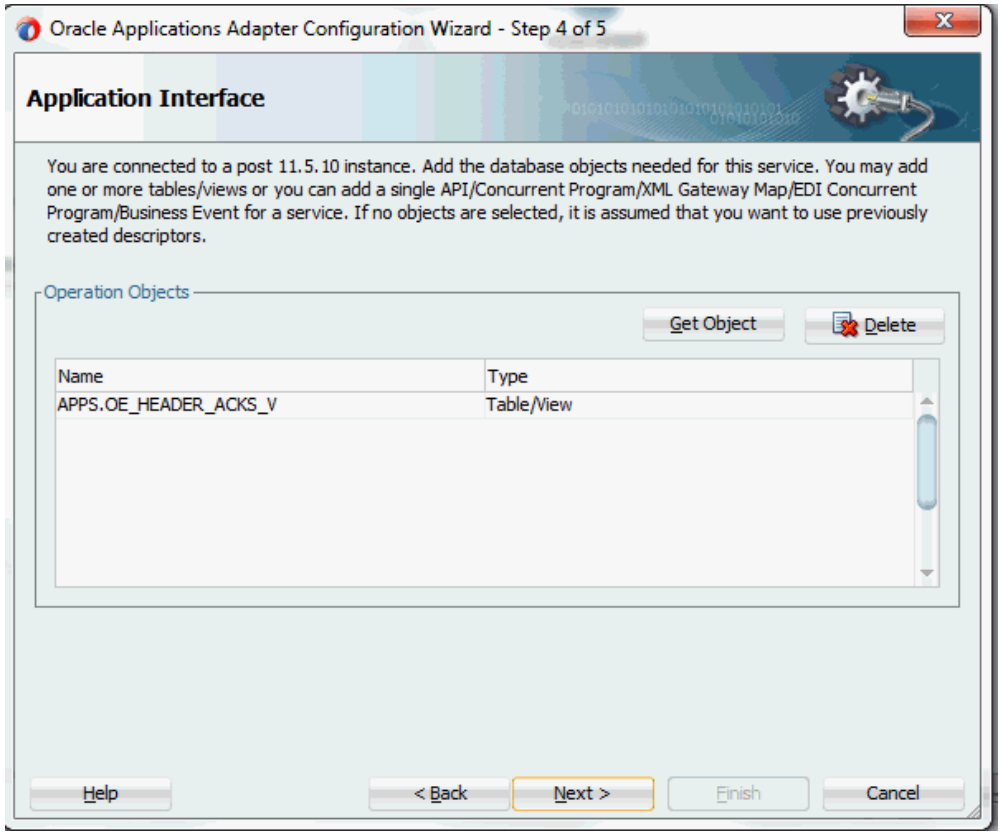
   Navigate to *Order Management Suite (OM_PF) > Order Management (ONT) > Sales Order (ONT_SALES_ORDER) > Interface_Views* to select `OE_HEADER_ACKS_V`.

Selecting a view from the Module Browser



6. Click **OK**. The Application Interface page appears with the selected interface view.

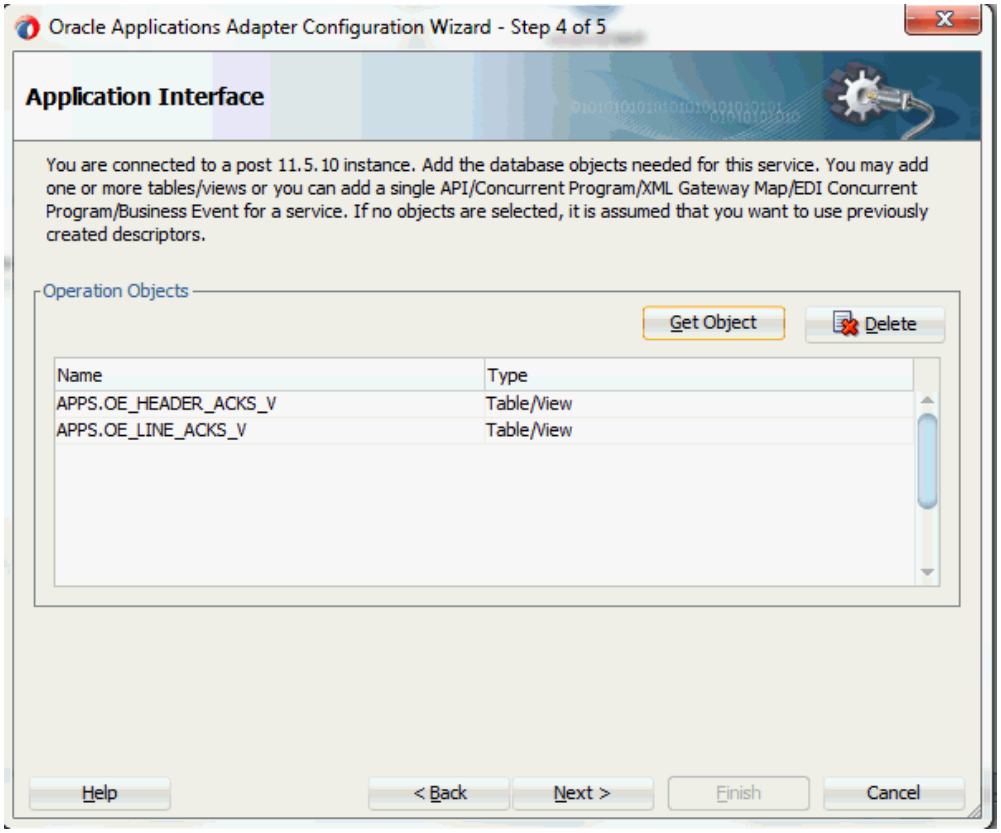
Adapter Configuration Wizard - Application Interface Page



7. Click **Get Object** to open the Oracle E-Business Suite Module Browser again to select another interface view`OE_LINE_ACKS_V` using the same navigation path *Order Management Suite (OM_PF) > Order Management (ONT) > Sales Order (ONT_SALES_ORDER) > Interface_Views*.

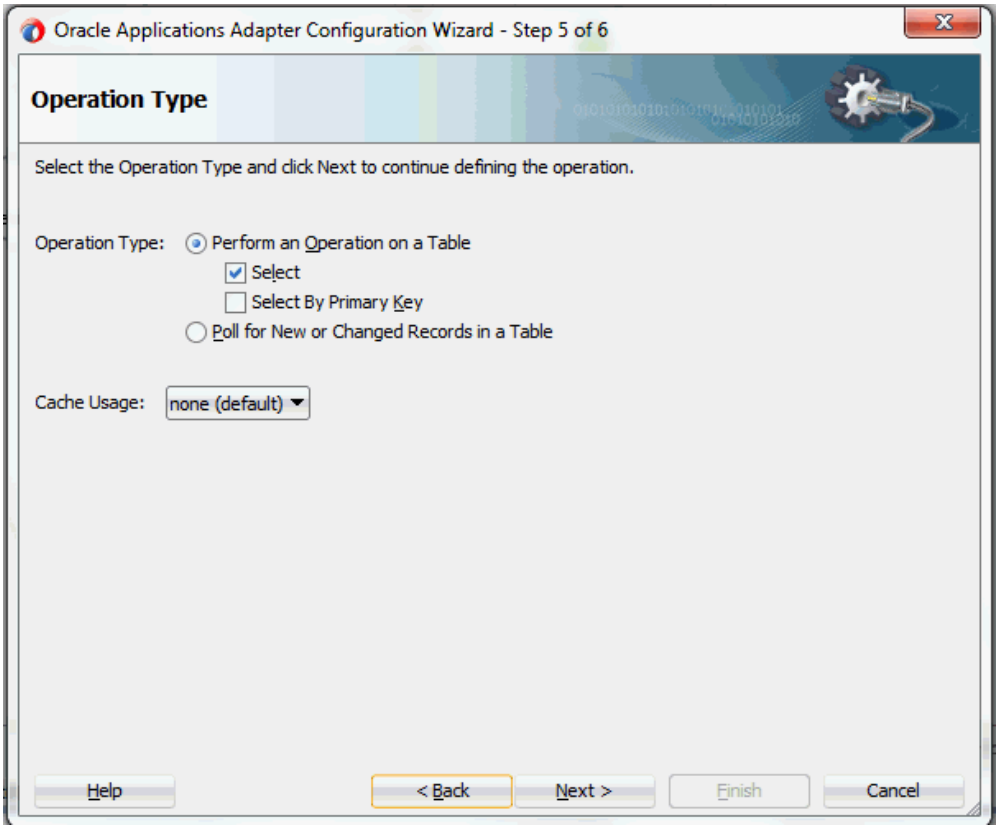
   Click **OK**. The Application Interface page appears with the two selected views.

Adapter Configuration Wizard - Application Interface Page



8. Click **Next**. The Operation Type page is displayed.

Adapter Configuration Wizard - Operation Type Page

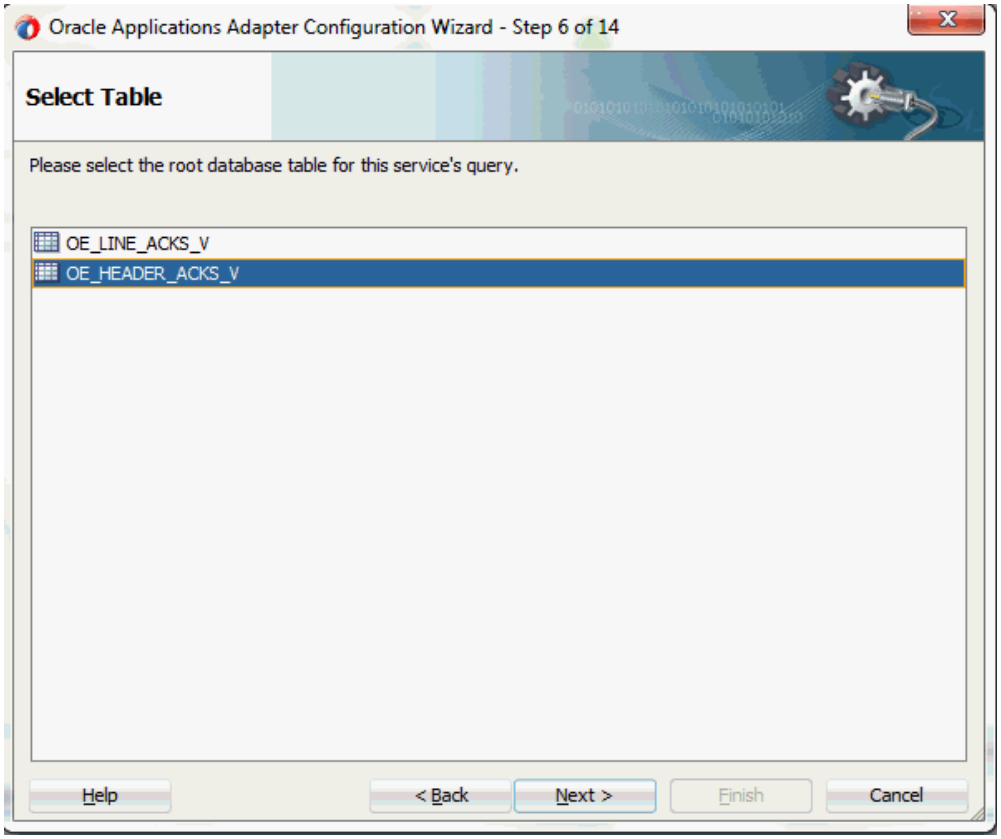


Select the **Perform an Operation on a Table** radio button and then choose the **Select** check box.

Note: Interface views can be used only for Select operations.

9. Click **Next**. The Select Table page is displayed.

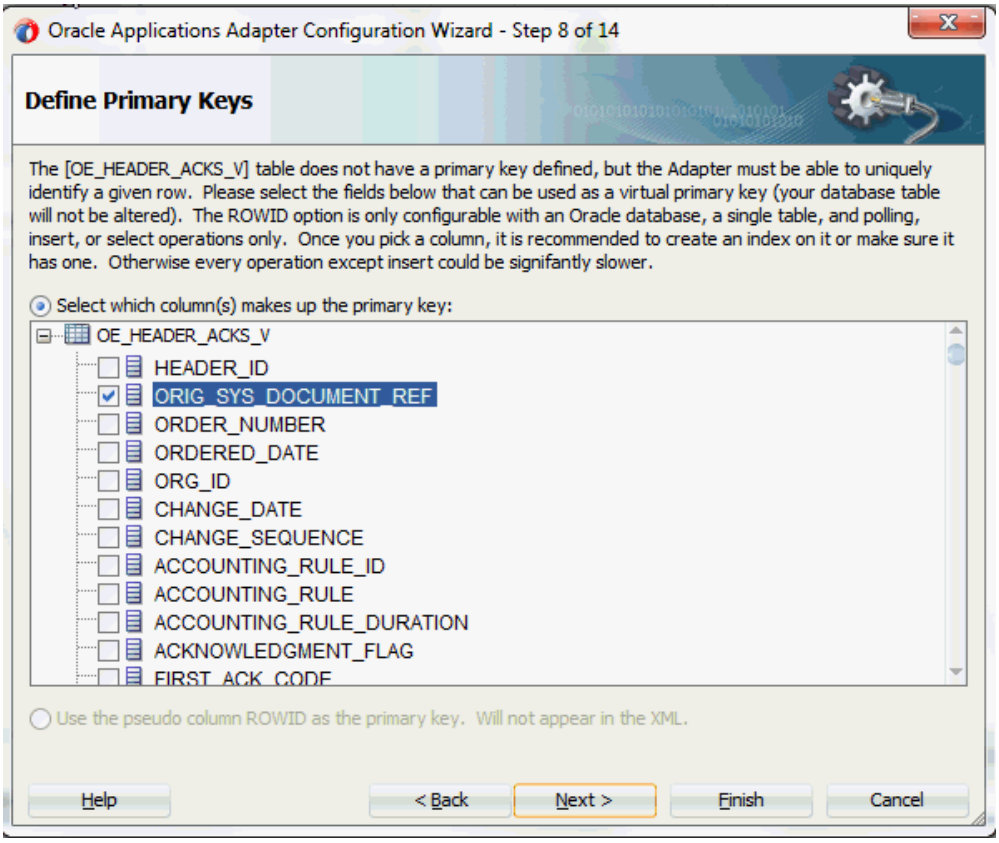
Adapter Configuration Wizard - Select Table Page



Select OE_HEADERS_ACKS_V as the root database table for this service's query.

10. Click **Next**. The Define Primary Keys page is displayed.

Adapter Configuration Wizard - Define Primary Keys Page for OE_HEADERS_ACKS_V



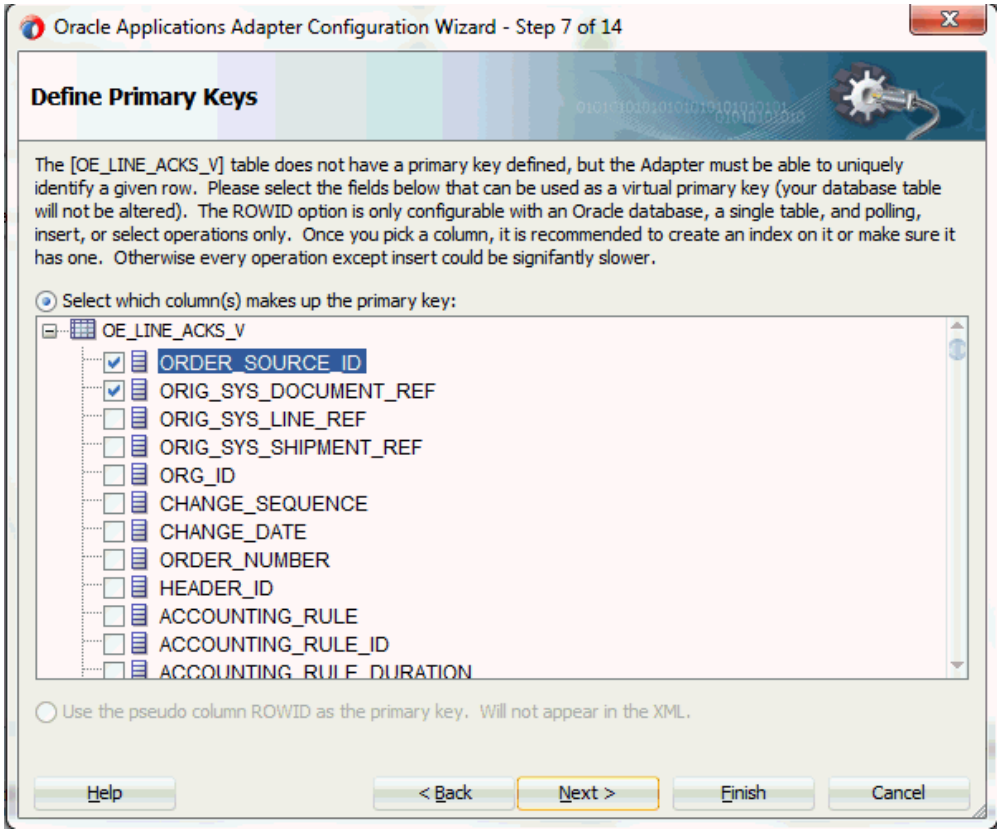
Select the following primary keys for the OE_HEADERS_ACKS_V table:

- ORIG_SYS_DOCUMENT_REF
- ORDER_SOURCE_ID

Click **Next**.

Select the same primary keys for the OE_LINE_ACKS_V table.

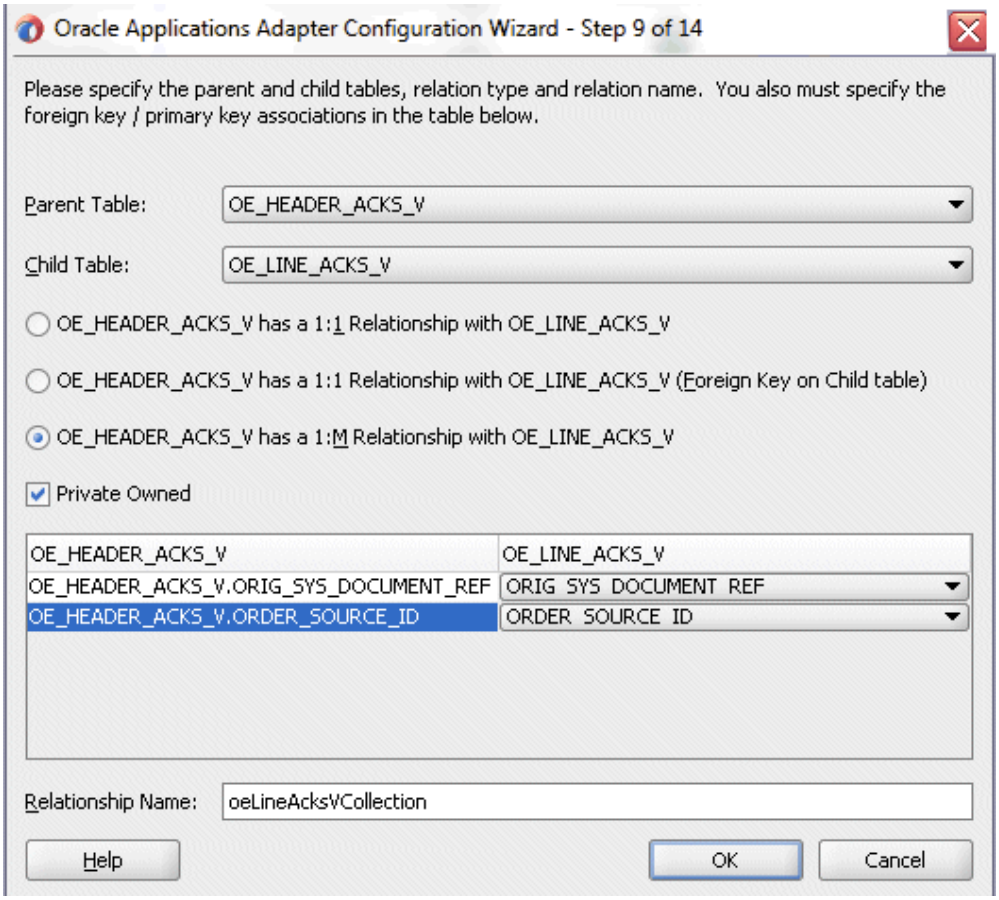
Adapter Configuration Wizard - Define Primary Keys Page for OE_LINE_ACKS_V



11. Click **Next**. The Relationships page appears.

    Click **Create** to open the Create Relationship dialog.

Defining Relationships



Enter the following information to define the relationship between the header and the detail table:

- Select the OE_HEADERS_ACKS_V as the parent table and OE_LINES_ACKS_V as the child table.
- Select the mapping type: OE_HEADERS_ACKS_V has a 1:M Relationship with OE_LINES_ACKS_V

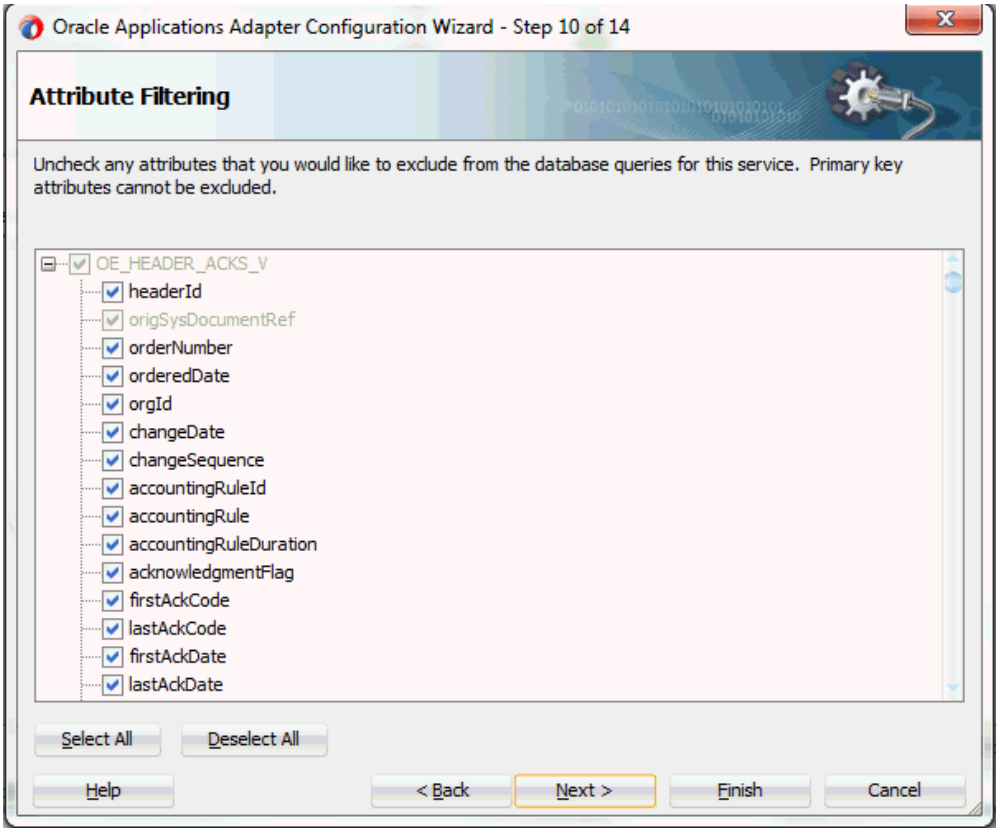
> **Note:** If foreign key constraints between tables already exist in the database, then two relationships are created automatically while importing tables. One of the relationships is 1:M relationship from the source table, which is the table containing the foreign key constraints, to the target table. 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: ORDER_SOURCE_ID
  - OE_HEADERS_ACKS_V.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.

Click **OK**. The newly created relationship information appears in the Relationships page.

12. Click **Next**. The Attribute Filtering page appears. Leave default selections unchanged.

Adapter Configuration Wizard - Attribute Filtering Page



Click **Next**. The Define Selection Criteria page appears.

Click **Add** to add a new parameter if necessary. Click **Edit** to use the graphical expression builder to create the expression. You can define your own custom Select SQL by modifying the predefined SQL string.

13. Click **Next**. The Advanced Options page appears.

    Click **Next**.

14. Click **Finish**. The wizard generates the WSDL file corresponding to the selected interface. This WSDL file is now available for the partner link.

Adding a Partner Link for File Adapter

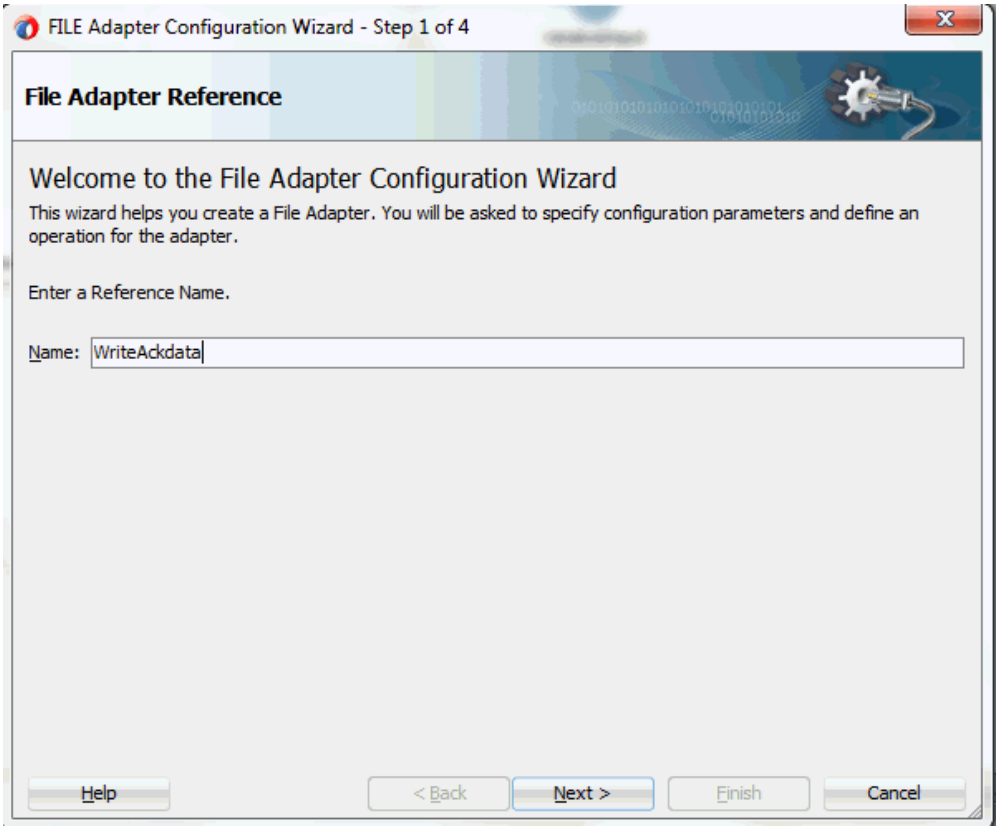
Use this step to configure a BPEL process by adding a partner link for File Adapter. This allows the acknowledgement data to be written to an XML file.

To add a partner link for the file adapter:

1. In Oracle JDeveloper BPEL Designer, click **BPEL Services** in the Component palette.

   Drag and drop **File Adapter** from the **BPEL Services** list into the right Partner Link swim lane of the process diagram right after the partner link you just created. The File Adapter Reference page appears.

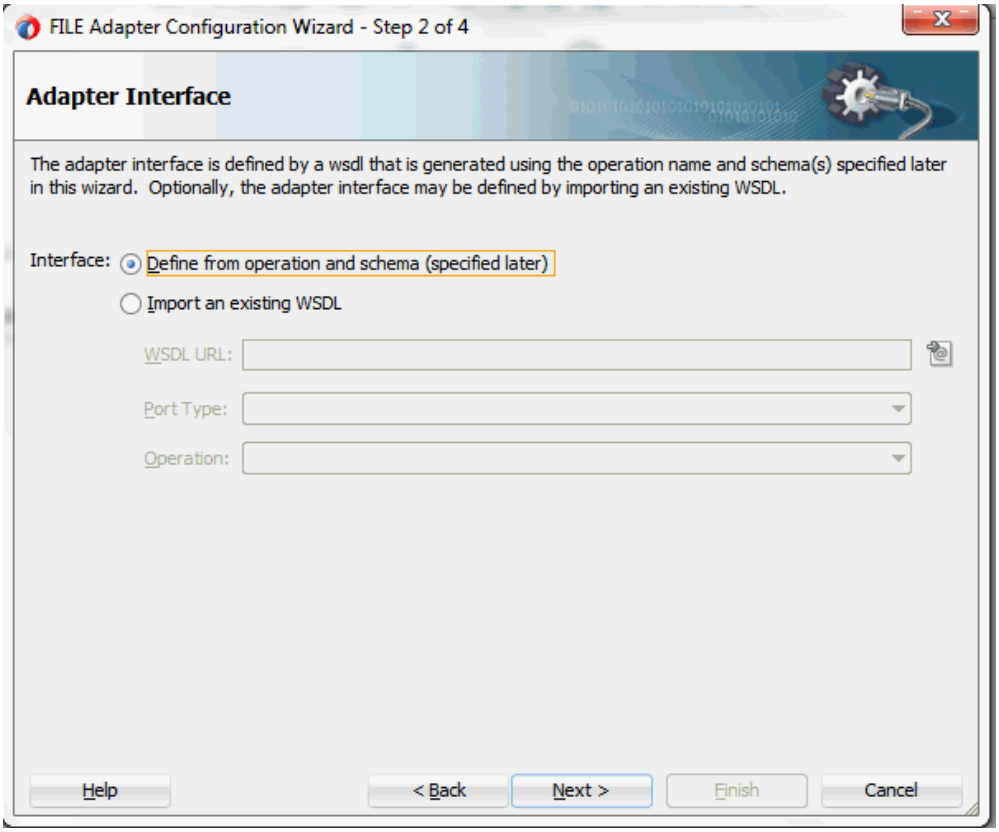
   Specifying a File Adapter Reference Name

   Enter a name for the File Adapter service, for example, `WriteAckdata`.

2. Click **Next**. The Adapter Interface page appears.

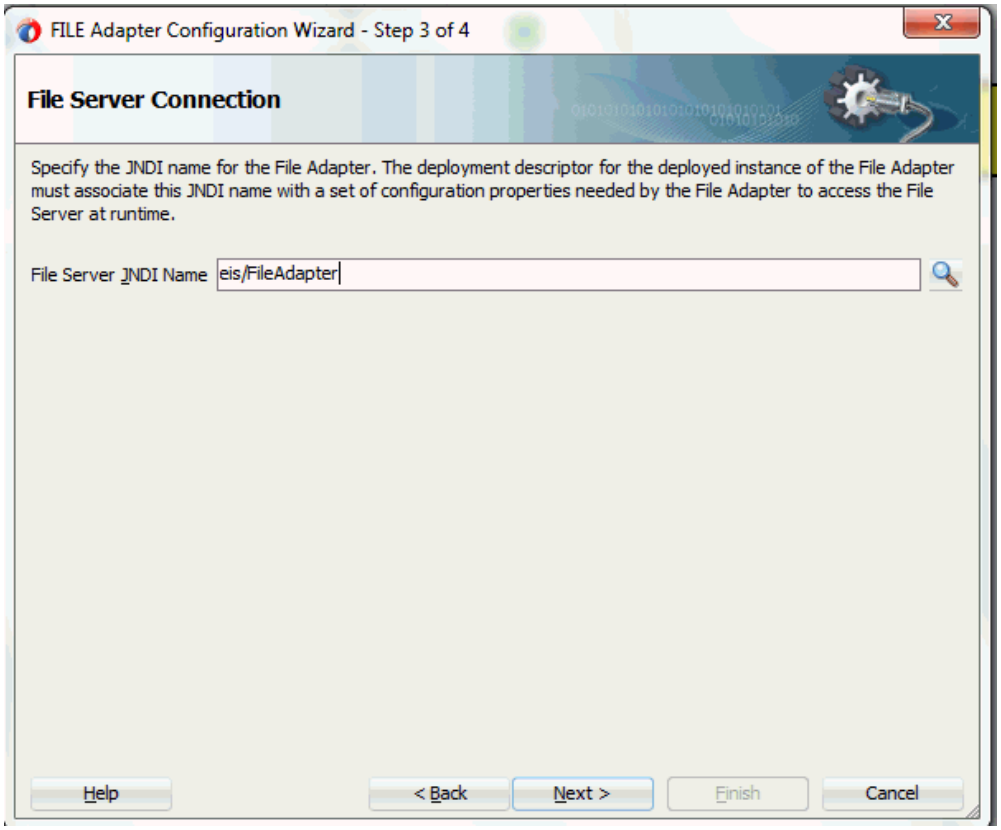
Specifying the Adapter Interface



Select the **Define from operation and schema (specified later)** radio button and click **Next**.

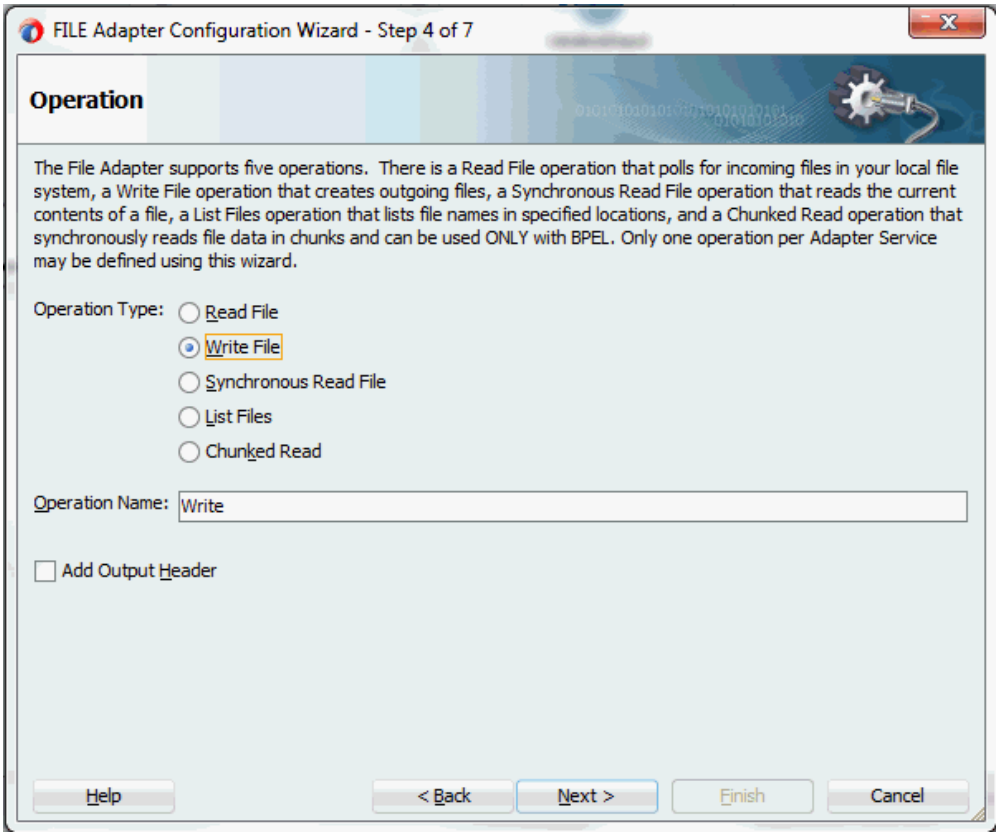
The File Server Connection page appears. Leave the default value unchanged for the File Server JNDI Name field.

Specifying File Server JNDI Name



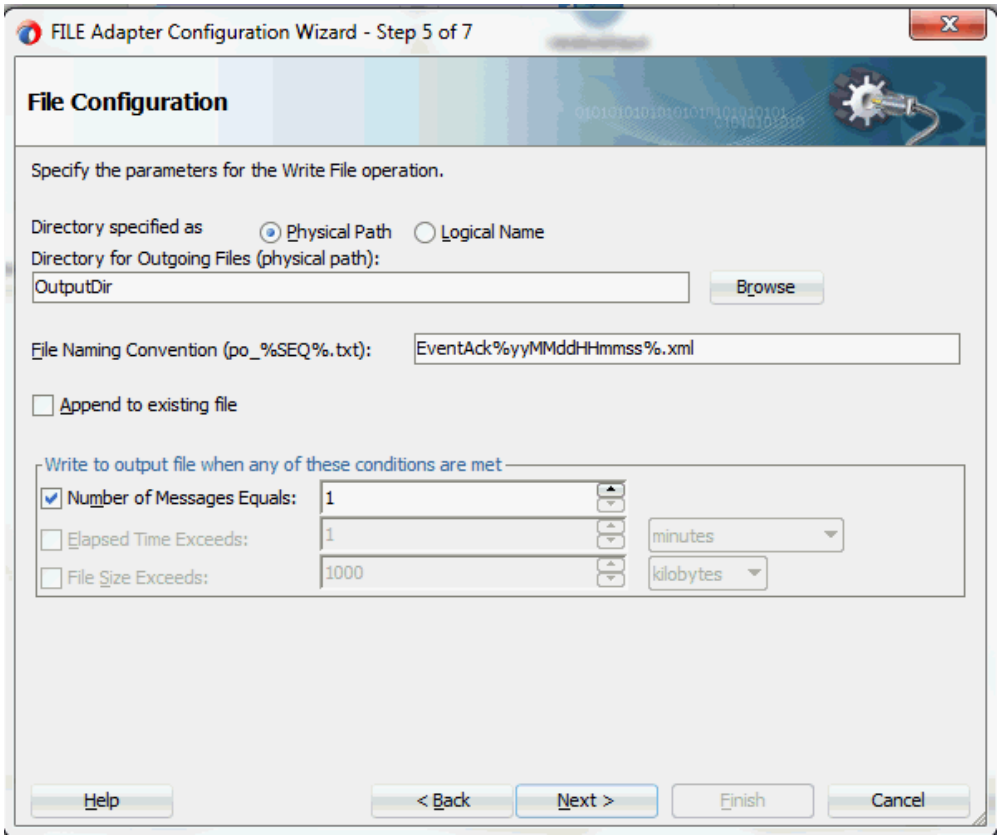
3. In the Operation page, specify the operation type. For example, select the **Write File** radio button. This automatically populates the **Operation Name** field.

Specifying the Operation



4. Click **Next** to access the File Configuration page.

Configuring the Output File



5. For the **Directory specified as** field, select the **Logical Name** radio button. Enter `OutputDir` as the **Directory for Outgoing Files (logical name)** and specify a naming convention for the output file, such as `EventAck%yyMMddHHmmss%.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**.

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

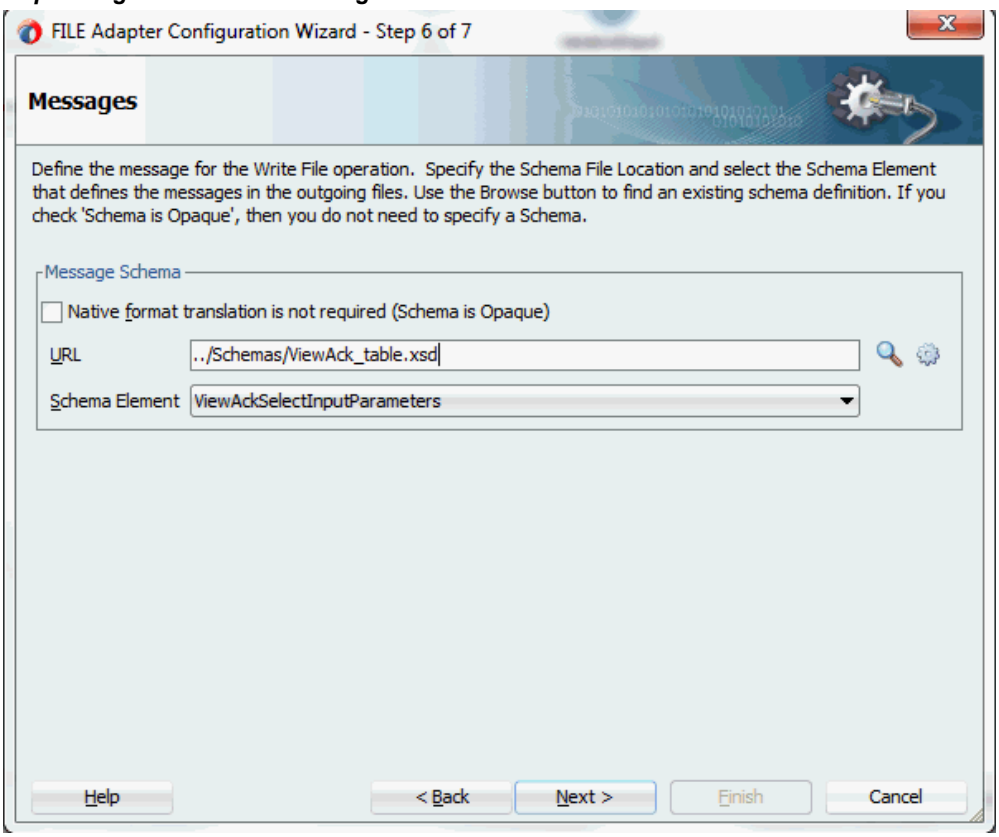
7. Click **Browse** to access the Type Chooser.

8. Expand the node by clicking **Project Schema Files > ViewAck_table.xsd** and selecting **OeHeaderAcksSelectInputParameters**. Click **OK**.

   The selected schema information will be automatically populated in the URL and

Schema Element fields.

Populating the Selected Message Schema



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.

Configuring Invoke Activities

Based on the scenario described earlier, you need to configure the following two Invoke activities:

1. To get the purchase order acknowledgement details by invoking the `ViewAck` partner link.

2. To write the purchase order acknowledgement information to an XML file by invoking `WriteAckdata` partner link for File Adapter.

To add the first Invoke activity for a partner link to get acknowledgement details:

1. In Oracle JDeveloper BPEL Designer, expand the **BPEL Constructs** from the

Component Palette. Drag and drop the first **Invoke** activity from the **Web Service** section into the center swim lane of the process diagram, between the **receiveInput** and **callbackClient** activities.

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

   The value of the Operation field is automatically selected based on the associated partner link. For example, this Invoke activity is associated with an interface table partner link for 'select' operation only, the 'ViewAckSelect' service name with 'Select' operation is populated as the value.

3. Enter a name for the Invoke activity. In the Input tab, ensure the **Input Variable** radio button is selected. Click the **Create** icon next to the **Input Variable** field to create a new variable. The Create Variable dialog box appears.

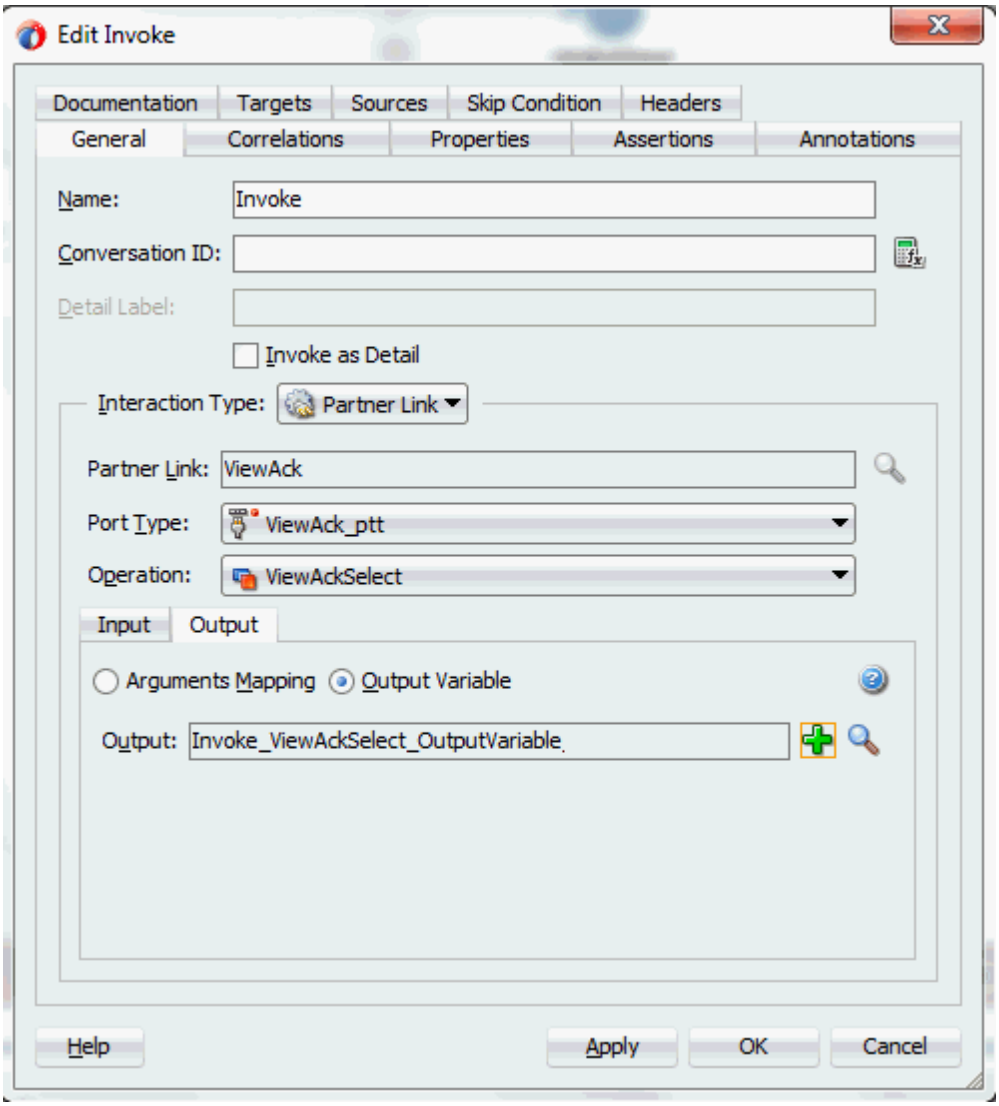
4. Select **Global Variable**, then enter a name for the variable. You can also accept the default name.

   Click **OK**.

5. In the Output tab, ensure the **Output Variable** radio button is selected. 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**, then enter a name for the variable. You can also accept the default name.

   Click **OK** to close the Create Variable dialog.



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.

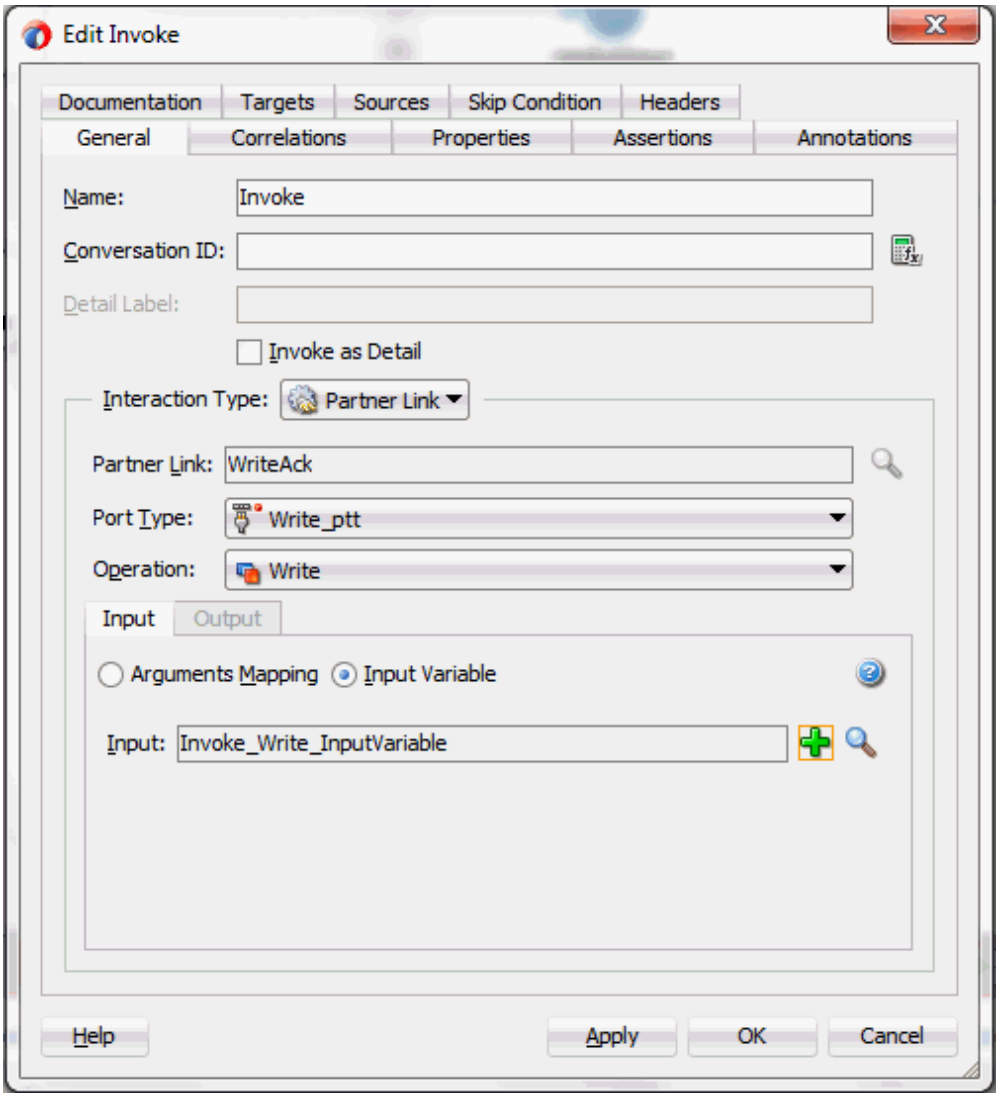
To add the second Invoke activity for a File Adapter partner link to write acknowledgement details in an XML file:

1. In Oracle JDeveloper BPEL Designer, expand the **BPEL Constructs** from the Component Palette. Drag and drop the first **Invoke** activity from the **Web Service** section into the center swim lane of the process diagram, right after the first **Invoke** activity.

2. Link the Invoke activity to the `WriteAckdata` service for File Adapter. The Edit Invoke dialog appears.

3. Enter a name for the Invoke activity. In the Input tab, ensure the **Input Variable** radio button is selected. 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**, then enter a name for the variable. You can also accept the default name. Click **OK** to return to the Edit Invoke dialog box.



Click **Apply** and then **OK** to finish configuring the Invoke activity.

The second Invoke activity appears in the process diagram.

Configuring an 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 an Assign activity:

1. In Oracle JDeveloper BPEL Designer, expand the **BPEL Constructs** from the Component Palette.

   Drag and drop the **Assign** activity from the **Basic Activities** section into the center

swim lane of the process diagram between the two **Invoke** activities that you just created earlier.

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

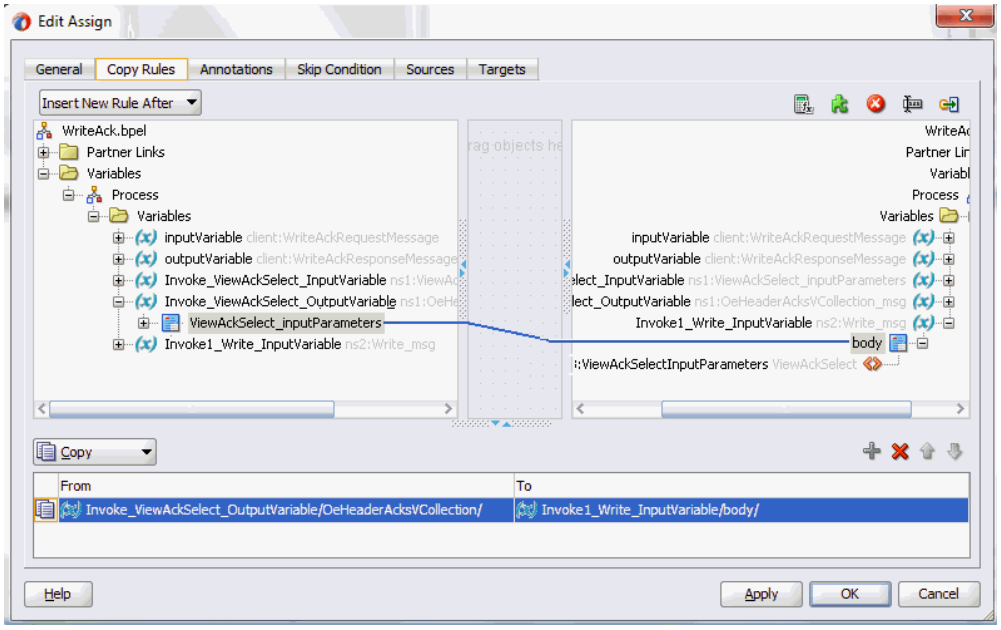
   Click the General tab to enter a name for the Assign activity. For example, `setAckdata`.

3. Select the Copy Rules tab and expand the target trees:

   - In the From navigation tree, navigate to **Variable > Process > Variables > Invoke_ViewAckSelect_OutputVariaable** and select **ViewAckSelect_inputParameters**.

   - In the To navigation tree, navigate to **Variable > Process > Variables > Invoke1_Write_InputVariable** and select **body**.

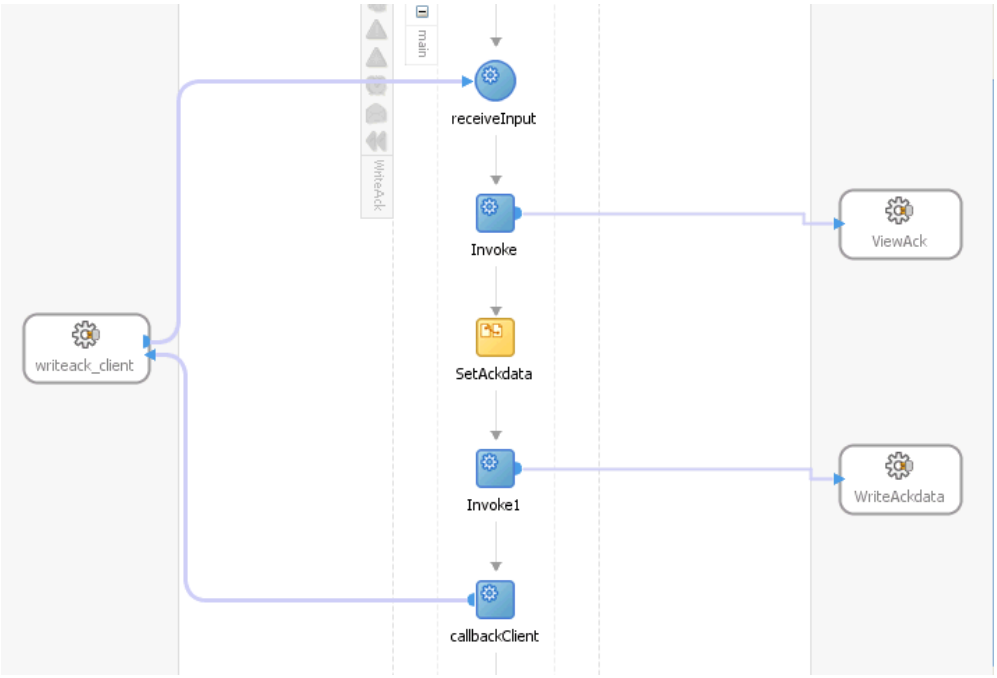
   Drag the source node (ViewAckSelect_inputParameters) to connect to the target node (body) that you just specified. This creates a line that connects the source and target nodes. The copy rule is displayed in the From and To sections at the bottom of the Edit Assign dialog box.



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

The BPEL diagram is completed.

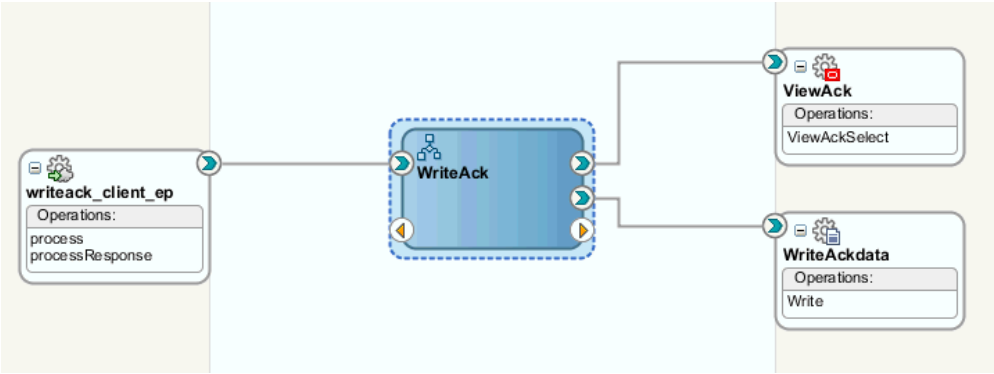
BPEL Process Diagram



Click the `WriteAck` (composite.xml) to display the Oracle JDeveloper composite diagram:

> **Note:** Click the Source tab of `WriteAck` (composite.xml) to enter a value for the physical directory `outputDir` for the reference WriteAckdata (such as `/usr/tmp`).
>
> ```
> <property name="outputDir" type="xs:string"
> many="false" override="may">/usr/tmp</property>
> ```

Oracle JDeveloper Composite Diagram



Run-Time Tasks for Views

After designing the SOA Composite application with BPEL process, the next steps are to deploy, run and monitor it.

1. Deploy the SOA Composite application with BPEL process, page 8-71
2. Test the deployed SOA Composite application with BPEL process, page 8-74

Deploying the SOA Composite Application with BPEL Process

To invoke the service from the BPEL client contained in the SOA composite, the SOA composite needs to be deployed to the Oracle WebLogic managed server. This can be achieved using Oracle JDeveloper. Once the composite is deployed, it can be tested from the Oracle Enterprise Manager Fusion Middleware Control Console.

Prerequisites

Before deploying the SOA composite with BPEL process using Oracle JDeveloper, you must have established the connectivity between the design-time environment and the run-time server. For more information, see Configuring the Data Source in Oracle WebLogic Server, page 11-1 and Creating an Application Server Connection, page 11-10.

> **Note:** If a local instance of the WebLogic Server is used, start the WebLogic Server by selecting **Run > Start Server Instance** from Oracle JDeveloper. Once the WebLogic Admin Server "DefaultServer" instance is successfully started, the <Server started in Running mode> and DefaultServer started message in the Running:DefaultServer and Messages logs should appear.

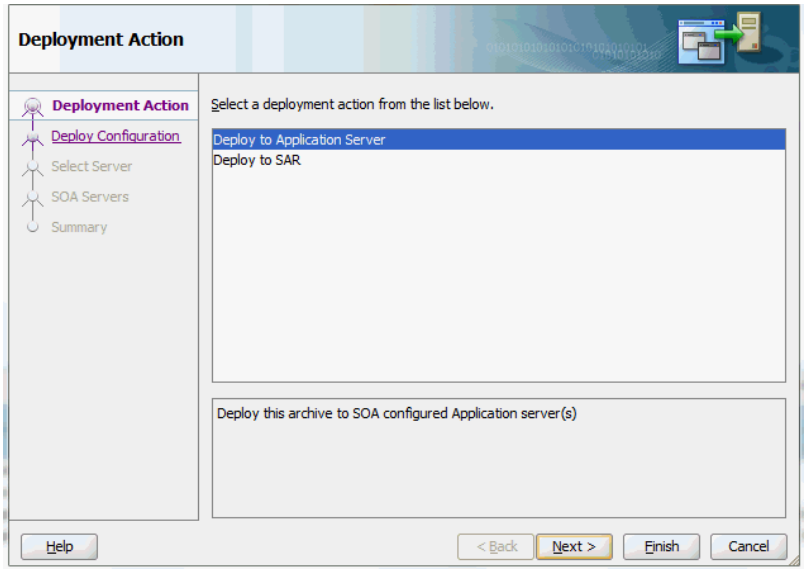
To deploy the SOA Composite application with BPEL process:

1. Select the SOA Composite project in the Applications Navigator.

2. Right-click the project name, and then select **Deploy > [project name] > [serverConnection]** from the menu that appears.

   For example, you can select **Deploy >WriteAck > soa-server1** to deploy the process if you have the connection set up appropriately.

   Note: If this is the first time to set up server connection, then the Deployment Action window appears. Select 'Deploy to Application Server' and click **Next**.



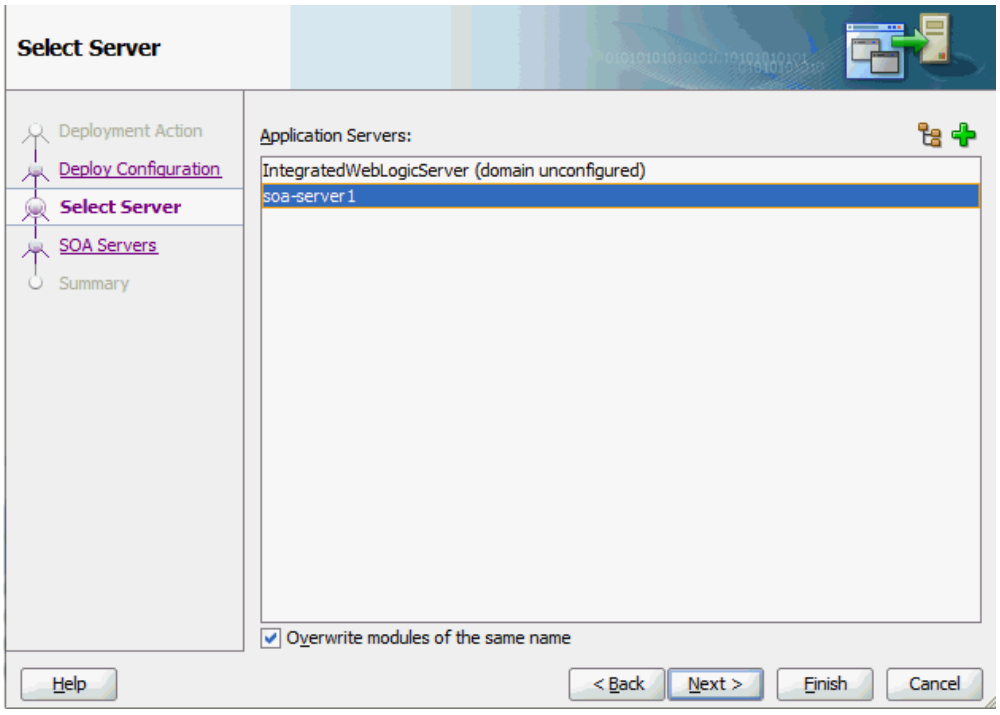
   In the Deploy Configuration window, ensure the following information is selected before clicking **Next** to add a new application server:

   - New Revision ID: 1.0
   - Mark composite revision as default: Select this check box.
   - Overwrite any existing composites with the same revision ID: Select this check box.

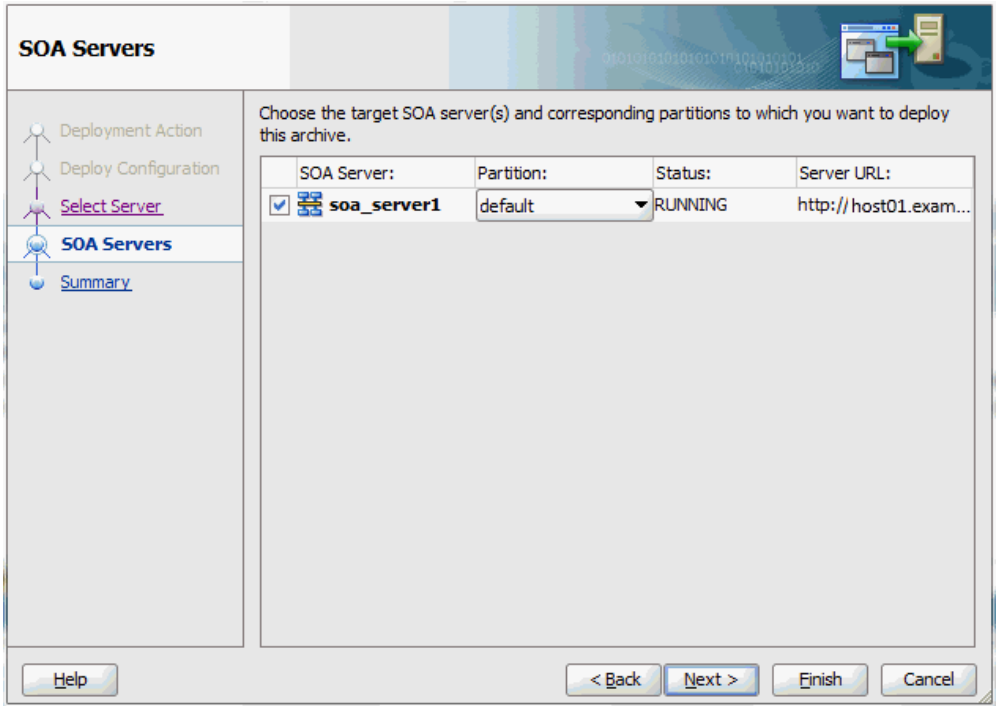
   The steps to create a new Oracle WebLogic Server connection from JDeveloper are covered in Creating an Application Server Connection, page 11-10.

3. In the Select Server page, select 'soa-server1' that you have established the server

connection earlier. Click **Next**.



4. In the SOA Servers page, accept the default target SOA Server ('soa-server1') selection.



Click **Next** and **Finish**.

If you are deploying the composite for the first time from your Oracle JDeveloper session, the Authorization Request window appears. Enter username and password information specified during Oracle SOA Suite installation. Click **OK**.

5. The BPEL process contained in the SOA Composite application is compiled and deployed. You can check the progress of the compilation in the Messages window.

Testing the Deployed SOA Composite Application with BPEL Process

Once the SOA Composite application is deployed, you can manage and monitor the process from the Oracle Enterprise Manager Fusion Middleware Control Console. You can also test the process and the integration interface by manually initiating the process.

To manually initiate and monitor the deployed SOA Composite application with BPEL process:

1. Navigate to Oracle Enterprise Manager Fusion Middleware Control Console (`http://<servername>:<portnumber>/em`). The composite you deployed is displayed in the Applications Navigation tree.

2. Enter username (such as `weblogic`) and password and click **Login** to log in to a farm.

   You may need to select an appropriate target instance farm if there are multiple target Oracle Enterprise Manager Fusion Middleware Control Console farms.

3. From the Farm base domain, expand the **SOA > soa-infra > soa-infra (soa_server1) > default** to navigate through the SOA Infrastructure home page and menu to access your deployed SOA Composite applications running in the SOA Infrastructure for that managed server.

   > **Note:** The Farm menu always displays at the top of the navigator. As you expand the SOA folder in the navigator and click the links displayed beneath it, the SOA Infrastructure menu becomes available at the top of the page.

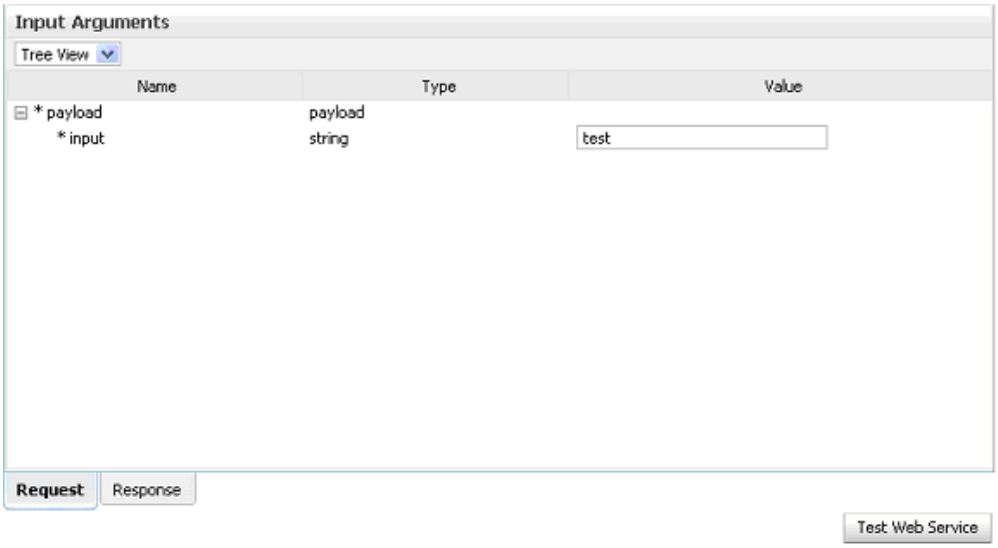
   Click the SOA Composite application that you want to initiate (such as 'WriteAck') from the SOA Infrastructure.

   Click **Test** at the top of the page.

4. The Test Web Service page for initiating an instance appears. You can specify the XML payload data to use in the Input Arguments section.

   Enter the input string required by the process and click **Test Web Service** to initiate the process.

Testing Web Service



The test results appear in the Response tab upon completion.

5. Click **Launch Flow Trace** in the Test Web Service page. The Flow Trace page is displayed.

   In the Trace section, you should find the sequence of the message flow for the service binding component (`writeack_client_ep`), BPEL component (`WriteAck`), and reference binding components (`ViewAck` and `WriteAckdata`). All involved components have successfully received and processed messages.

   If any error occurred during the test, you should find it in the Faults section.

6. Click your BPEL service component instance link (such as `WriteAck`) to display the Instances page where you can view execution details for the BPEL activities in the Audit Trail tab.

   Click the Flow tab to check the BPEL process flow diagram. Click an activity of the process diagram to view the activity details and flow of the payload through the process.

7. **Verifying Records in Oracle E-Business Suite**

   Log on to Oracle E-Business Suite with Purchasing, Vision Operations (USA) responsibility and select Purchase Order from the navigation menu.

   The Purchase Order forms is displayed.

8. Create a purchase order with the following header values:

   - Supplier: Enter a supplier information, such as 'Advanced Network Devices'.

- Site: Select a site information, such as 'SANTA CLARA-ERS'.

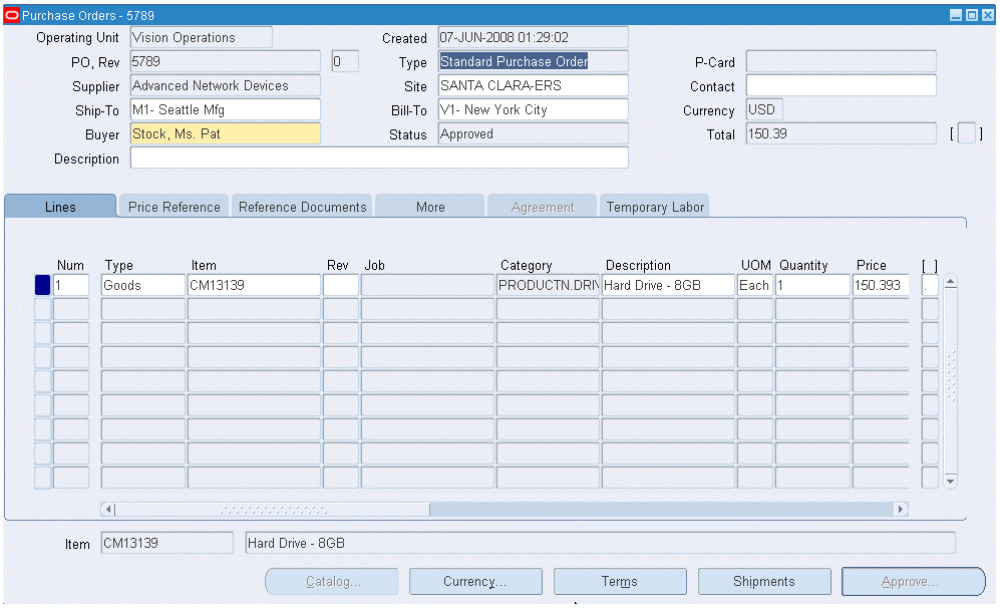
9. On the Lines tab, enter a data row with the following values:
   - Type: Goods
   - Item: CM13139
   - Quantity: 1
   - Description: Hard Drive - 8GB
   - Promised: Enter any future date in the format of dd-mmm-yyyy (such as 23-JUN-2009)

10. Save your purchase order. The status of the purchase order is 'Incomplete'.

11. Click **Approve**. The Approve Document form appears.

    Click **OK** to confirm the approval.

    Note: Because the trading partner is set up and valid, the transmission method is automatically set to **XML**.



The status of the purchase order is now changed to 'Approved'. For future reference, note the value of the PO, Rev field. For example, the PO number 5789.

Once the purchase order is approved, the order details should be recorded in the system.

After deploying the BPEL process contained in a SOA Composite application, the order acknowledgement information should be retrieved from the `OE_HEADER_ACKS_V` and `OE_LINE_ACKS_V` interface views.

Validate the output file `EventAck%yyMMddHHmmss%.xml` in the specified output directory by opening the xml file to confirm the purchase order details. The document number in the xml file should be the same number you just approved in Oracle E-Business Suite.

9

Using PL/SQL APIs

This chapter covers the following topics:

- Overview of PL/SQL APIs
- Design-Time Tasks for PL/SQL APIs
- Creating a New SOA Composite Application with BPEL Process
- Adding Partner Links
- Adding a Partner Link for File Adapter
- Defining Wrapper APIs
- Declaring Parameters with a DEFAULT Clause
- Configuring the Invoke Activities
- Configuring a Transform Activity
- Configuring an Assign Activity
- Run-Time Tasks for PL/SQL APIs
- Deploying the SOA Composite Application with BPEL Process
- Testing the Deployed SOA Composite Application with BPEL Process
- Troubleshooting

Overview of PL/SQL APIs

Oracle E-Business Suite Adapter uses PL/SQL application programming interfaces (APIs) to insert and update data in Oracle E-Business Suite. APIs are stored procedures that enable you to insert and update data in 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 E-Business Suite.

Note: For more information about PL/SQL procedure limitations, refer

to *Oracle Fusion Middleware User's Guide for Technology Adapters.*

Design-Time Tasks for PL/SQL APIs

This section describes how to configure the Oracle E-Business Suite Adapter to use PL/SQL APIs.

SOA Composite Application with BPEL Process Scenario

In this example, Oracle E-Business Suite Adapter exposes the following stored procedures as Web services in a BPEL process to update the 'quantity' field of an existing purchase order based on user input.

- A custom Order Management stored procedure `GET_G_MISS_LINE_REC (GET_G_MISS_LINE_REC)` to initialize the purchase order
- A PL/SQL Process Order Line (`PROCESS_LINE`) to update the existing purchase order

When a change order request is received, the purchase order information including order quantity and other line item details will be retrieved. Based on user input, a new order quantity will be updated in Oracle Order Management.

When the SOA Composite application with BPEL process has been successfully executed after deployment, you can validate the process by querying it directly from Order Management tables or validate it from the Formed-based Oracle Order Management application. The retrieved `ordered_quantity` value from the query table or you find in the Order Management application should be the same as the quantity value given by the user through `changeorder_data.xml` file.

Prerequisites to Configure PL/SQL APIs

Oracle E-Business Suite Adapter 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 BPEL PM in order to provide context information for Oracle E-Business Suite. The context information is required for an API transaction in order for an Oracle E-Business Suite user that has sufficient privileges to run the program.

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 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. 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 Invoke activity.

For more information about applications context, see Supporting for Normalized Message Properties, page 4-7.

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. Oracle E-Business Suite Adapter 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 SOA Composite application with BPEL process, page 9-4
2. Add partner links, page 9-8
3. Add a partner link for File Adapter, page 9-19
4. Define wrapper APIs, page 9-25
5. Declare parameters with a DEFAULT clause, page 9-28
6. Configure Invoke activities, page 9-31
7. Configure a Transform activity, page 9-33
8. Configure an Assign activity, page 9-35

Creating a New SOA Composite Application with BPEL Process

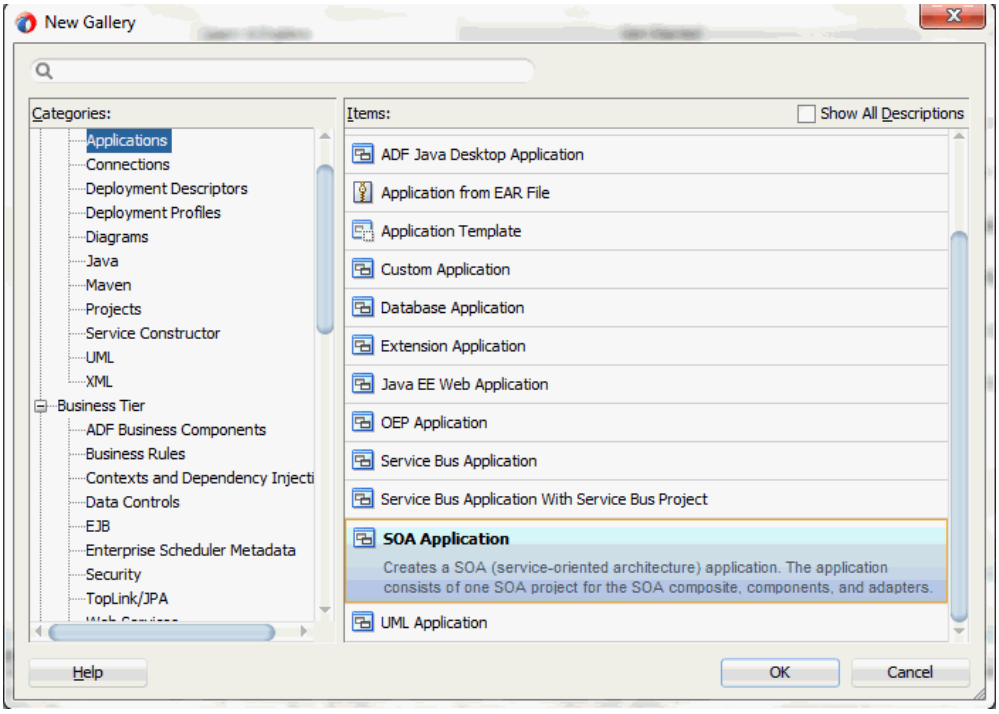
To create a new SOA Composite application with BPEL process:

1. Launch Oracle JDeveloper.

2. Click **New Application** in the Application Navigator.

   The New Gallery page appears. Select **SOA Application** from the Items list.

   The New Gallery Page

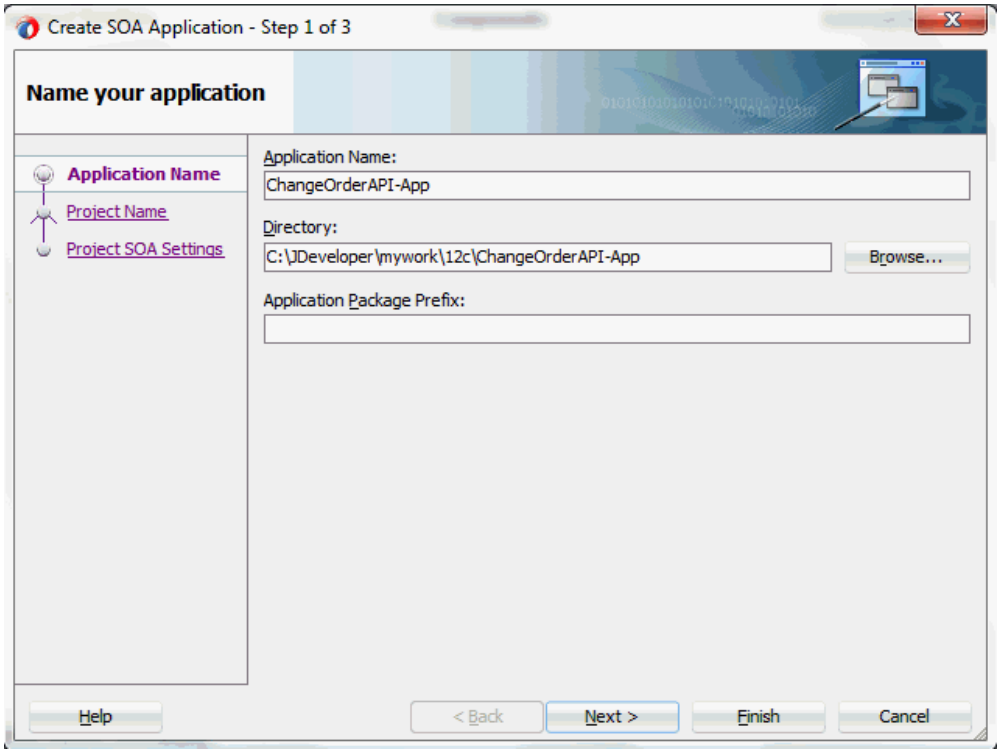
   

   Click **OK**.

3. Click **New Application** in the Application Navigator.

   The Create SOA Application - Name your application page is displayed.

The Create SOA Application - Name your application Page



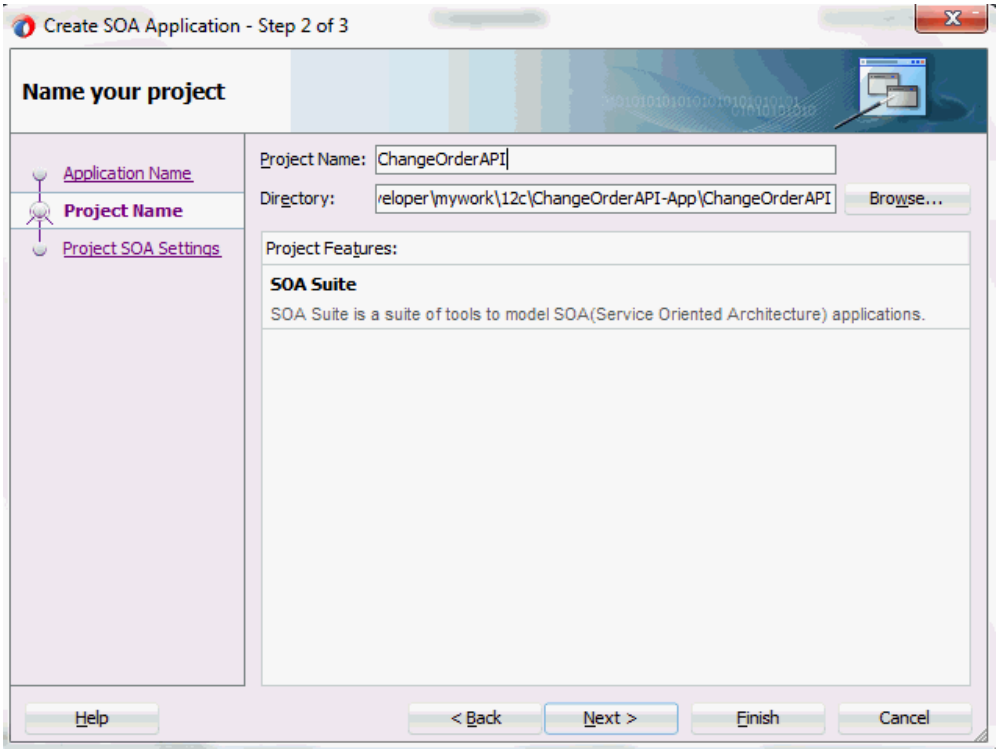
4. Enter an appropriate name for the application in the **Application Name** field.

   Click **Next**. The Create SOA Application - Name your project page is displayed.

5. Enter an appropriate name for the project in the **Project Name** field. For example, `ChangeOrderAPI`.

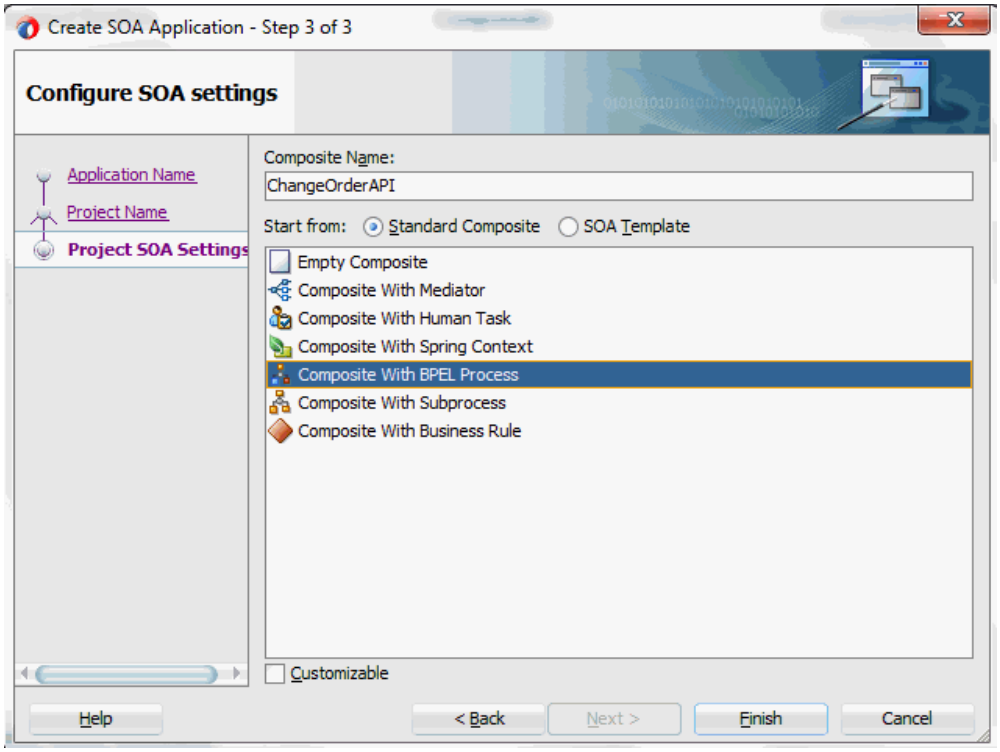
   Notice that **SOA Suite** is shown in the Project Features section.

The Create SOA Application - Name your project Page



6. Click **Next**. The Create SOA Application - Configure SOA settings page is displayed.

The Create SOA Application - Configure SOA settings Page



7. Leave the default **Standard Composite** selection unchanged for the Start from field.

   Select **Composite With BPEL Process** from the Standard Composite list, and then click **Finish**. You have created a new application, and a SOA project. This automatically creates a SOA Composite.

   The Create BPEL Process page is displayed.

8. Leave the default **BPEL 2.0 Specification** selection unchanged. This creates a BPEL project that supports the BPEL 2.0 specification.

   Enter an appropriate name for the BPEL process in the **Name** field. For example, `ChangeOrderAPI`.

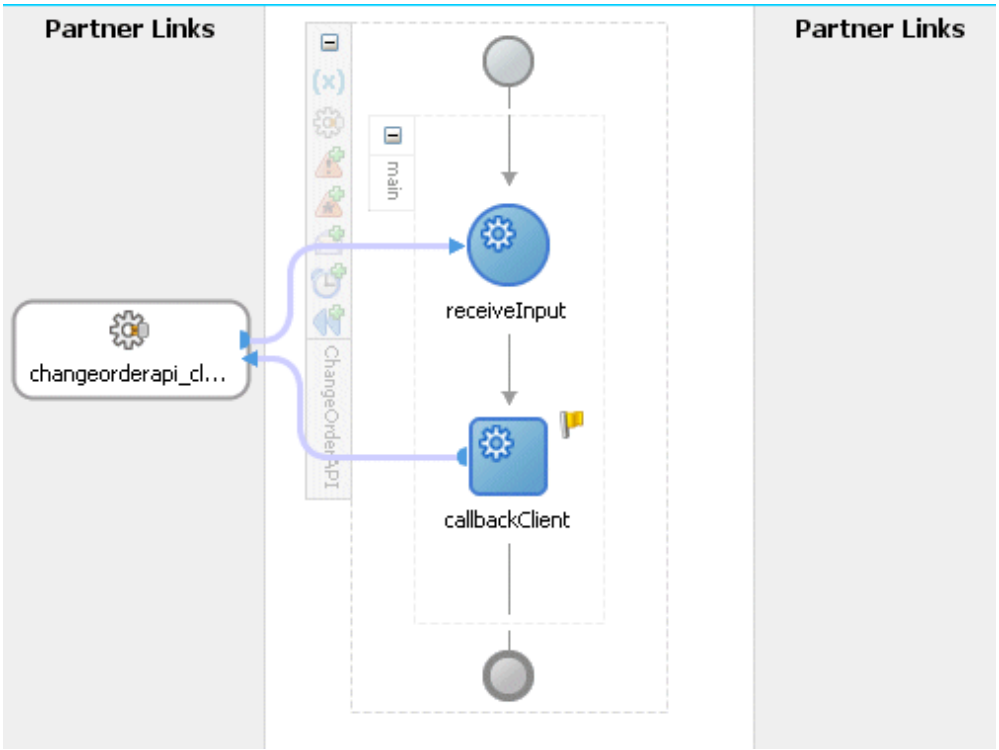
   Select **Asynchronous BPEL Process** in the **Template** field. Click **OK**.

   An asynchronous BPEL process is created with the Receive and Reply activities. The required source files including bpel and wsdl, using the name you specified (for example, `ChangeOrderAPI.bpel` and `ChangeOrderAPI.wsdl`) and `ChangeOrderAPI` (composite.xml) are also generated.

9. Navigate to SOA folder and click `ChangeOrderAPI` (composite.xml) to view the composite diagram.

   Double click on the `ChangeOrderAPI` component to open the BPEL process.

New BPEL Process



Adding Partner Links

The next task is to add a partner link to the BPEL process. This section describes how to create a partner link for the application service using a BPEL process. A BPEL partner link defines the link name, type, and the role of the BPEL process that interacts with the partner service.

Based on the BPEL process scenario discussed earlier, the following two partner links need to be configured:

- Add the first partner link `(initLineRec)` to initialize the purchase order
- Add the second partner link `(OrderManagement)` to update the existing purchase order

To add the first partner link:

1. Click **BPEL Services** in the Component Palette.

   Drag and drop **Oracle E-Business Suite** (formerly known as **Oracle Applications**) from the **BPEL Services** list into the right Partner Link swim lane of the process diagram. The Oracle E-Business Suite Reference page appears.

Enter a service name in the **Name** field. For example, `initLineRec`.

2. 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 **Create a New Database Connection** icon.

     How to define a new database connection, see Create a New Database Connection, page 5-14.

   - 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 Oracle E-Business Suite Adapter 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 Fusion Middleware User's Guide for Technology Adapters.*

4. Once you have completed creating a new connection for the service, you can add a PL/SQL API by browsing through the list of APIs available in Oracle E-Business Suite.

   Click **Next**.

   If you are connecting to Oracle E-Business Suite Release 12, then the **IREP File not present** dialog appears indicating that Adapter could not find the Oracle Integration Repository data file corresponding to the database you are connecting to Oracle E-Business Suite in your workspace. Absence of the data file would make browsing or searching of Integration Repository tree considerably slow. You can

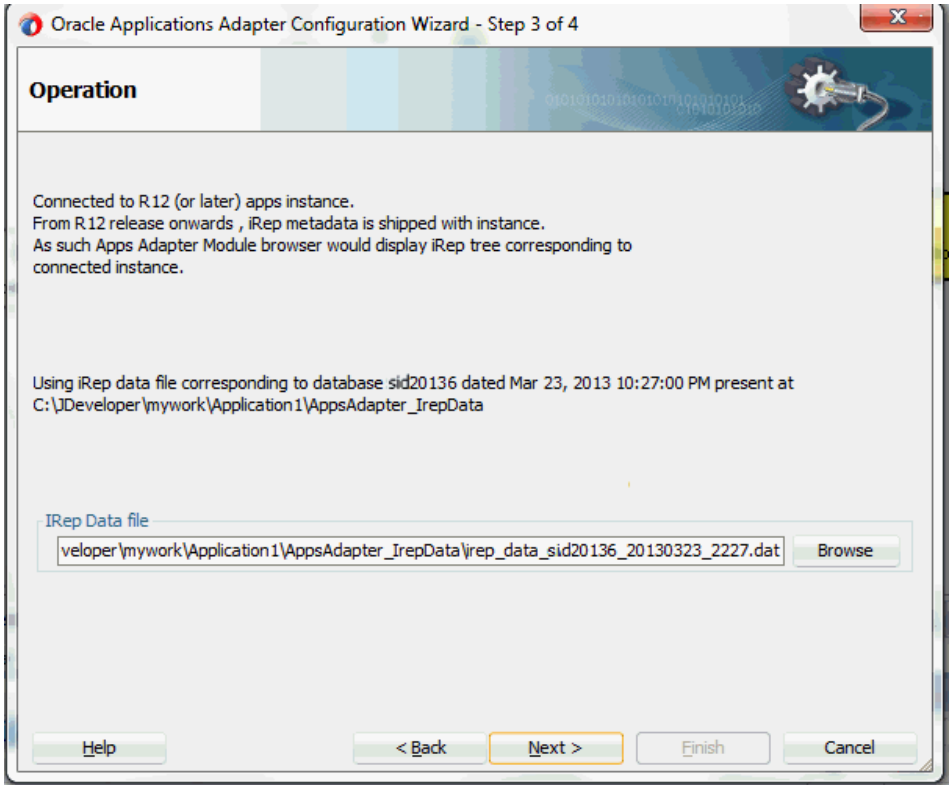
choose to extract the data file and create a local copy of the Integration Repository data file. Once it is created successfully, Adapter will pick it up automatically next time and retrieve data from your local Integration Repository.

You can select one of the following options:

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

  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



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

In this example, click **No**. Click **Next** in the Operation page to open the Oracle E-Business Suite Module Browser.

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

Note: The Oracle E-Business Suite 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. PL/SQL APIs can be found under the PL/SQL category.

Navigate to *Other Interfaces* > *Custom Objects* >*PLSQL APIs* > *OE_ORDER_PUB* to select a custom stored procedure `GET_G_MISS_LINE_REC`.

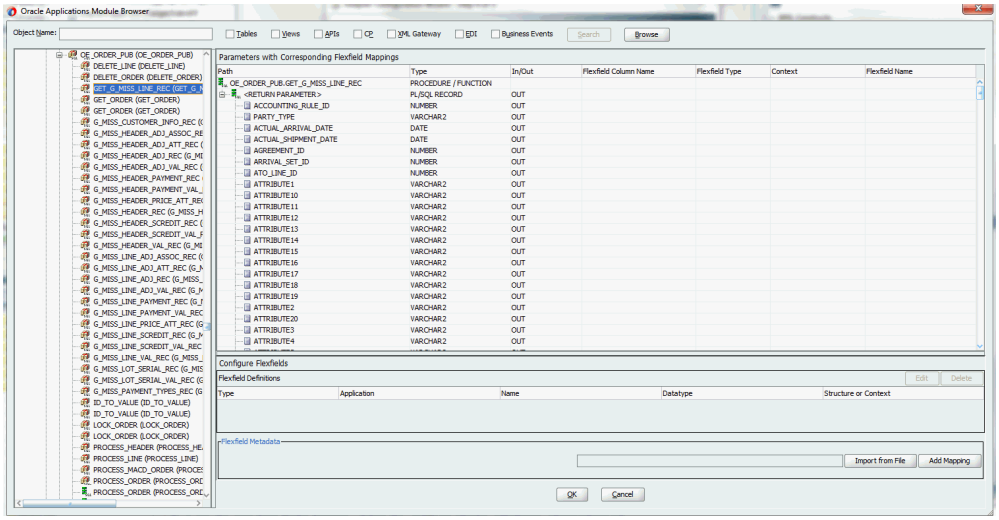
Note: The `GET_G_MISS_LINE_REC (GET_G_MISS_LINE_REC)` stored procedure does not take any input but has a complex data type (PL/SQL Table) as output. Oracle E-Business Suite Adapter design-time automatically generates a wrapper stored procedure and loads it on the underlying database.

6. Oracle E-Business Suite Adapter provides flexfield support for PL/SQL APIs. Once the selected API is displayed in the module browser, you can either configure flexfield mappings for the API that has flexfields defined in Oracle E-Business Suite or use current data without further flexfield configuration.

   Note: If a selected API has pre-configured flexfield mappings available, a pop-up window appears allowing you to decide if you want to use these pre-configured mappings. Click **Yes** to display these mappings as part of the flexfield definitions in the Configure Flexfields region. You can modify them later if needed.

   Click **No** to indicate that you will not use these mappings and the pop-up window disappears.

Displaying Selected API in the Oracle E-Business Suite Module Browser



Configuring Flexfield Mapping Data for the Selected API

Perform either one of the following tasks in the Oracle E-Business Suite Module Browser:

- Use the current data without further configuration by clicking **OK** to continue.

- Configure flexfield mapping data by either using an existing flexfield mapping or creating a new mapping through a flexfield wizard if desired.

  - **Creating a New Mapping**

    You can create a new mapping for a selected API by clicking **Add Mapping** in the Flexfield Metadata section. A new flexfield mapping wizard appears guiding you through each configuration page where you can add key and descriptive flexfields, as well as configure flexfield mapping between the selected API parameters and flexfields defined in the Oracle E-Business Suite instance. For more information on how to create a new mapping and configure flexfields for PL/SQL APIs, see:

    - Adding or Configuring a New Mapping, page 4-37

    - Configuring Key Flexfield Mappings for a PL/SQL API, page 4-38

    - Configuring Descriptive Flexfield Mappings for a PL/SQL API, page 4-59

  - **Importing an Existing Mapping**

    Instead of creating a new mapping, you can use an existing mapping that

has been created earlier by clicking **Import from File** in the Flexfield Metadata section. This lets you select a desired flexfield mapping for your selected API. Once a desired mapping is imported, you can modify the context values for a descriptive flexfield, and modify structure for a key flexfield if needed to meet your needs.

See: Importing an Existing Flexfield Mapping, page 4-115.

For more information on flexfield support for PL/SQL APIs and how to modify existing mappings, see Flexfield Support for PL/SQL APIs and Open Interface Tables, page 4-27.

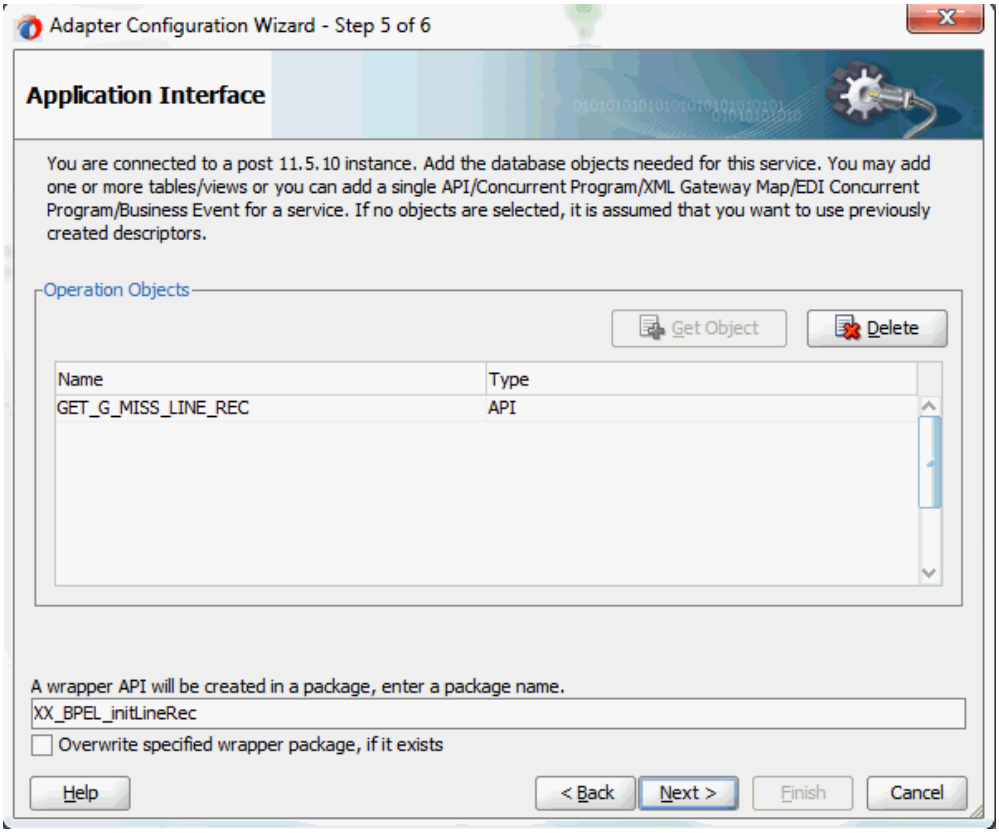
In this example, click **OK** to continue.

Please note that the input and the output parameters of the Stored Procedure contains complex data types that are not readily mapped to JDBC types. The Adapter wizard provides a mechanism that detects when these types are used and then invokes Oracle JPublisher to generate the necessary wrappers automatically. Oracle JPublisher generates two SQL files, one to create schema objects, and another to drop them. The SQL that creates the schema objects is automatically executed from within the wizard to create the schema objects in the database schema before the XSD is generated.

The Adapter Service points to the wrapper Stored Procedure instead of the Process Order Line Stored Procedure. The Adapter design-time automatically loads the wrapper procedure onto the underlying database.

7. In the Application Interface page, click **Next**.

Application Interface Page



In the Finish page, click **Finish** to complete the partner link creation.

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

To add the second partner link:

1. Click **BPEL Services** in the Component Palette.

   Drag and drop **Oracle E-Business Suite** (formerly known as **Oracle Applications**) from the **BPEL Services** list into the right Partner Link swim lane of the process diagram. The Oracle E-Business Suite Reference page appears.

   Enter a service name in the **Name** field. For example, `OrderManagement`.

2. 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 **Create a New Database Connection** icon.

  How to define a new database connection, see Create a New Database Connection, page 5-14.

- 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 Oracle E-Business Suite Adapter 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 Fusion Middleware User's Guide for Technology Adapters.*

4. Once you have completed creating a new connection for the service, you can add a PL/SQL API by browsing through the list of APIs available in Oracle E-Business Suite.

   Click **Next**.

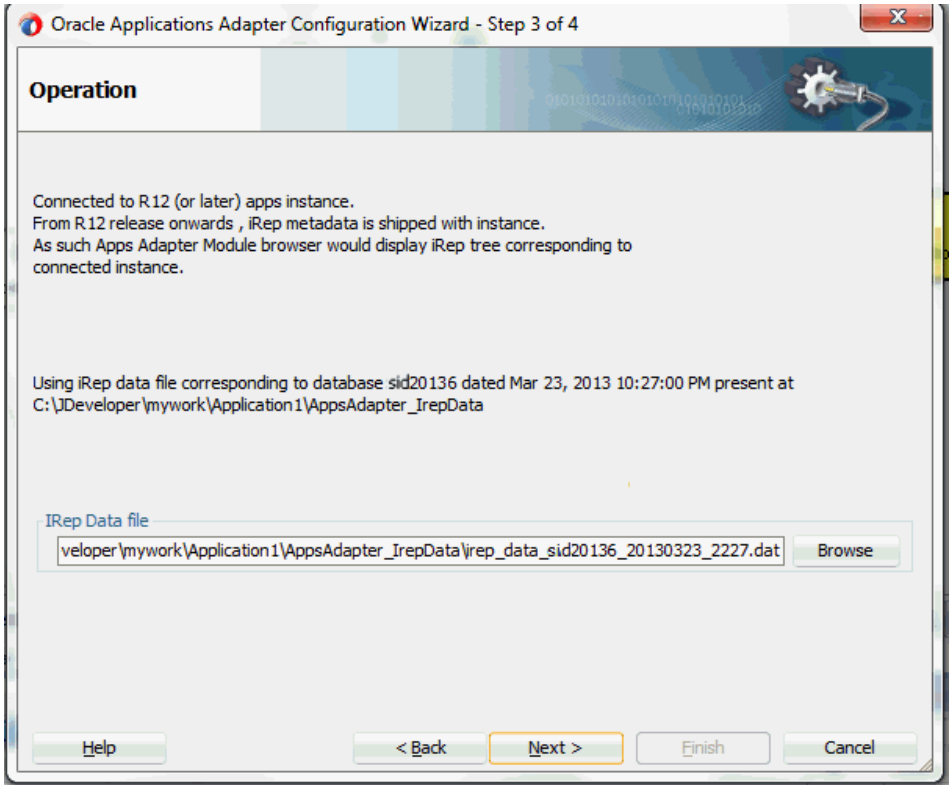
   If you are connecting to Oracle E-Business Suite Release 12, then the **IREP File not present** dialog appears indicating that Adapter could not find the Oracle Integration Repository data file corresponding to the database you are connecting in your workspace. Absence of the data file would make browsing 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 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.

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



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

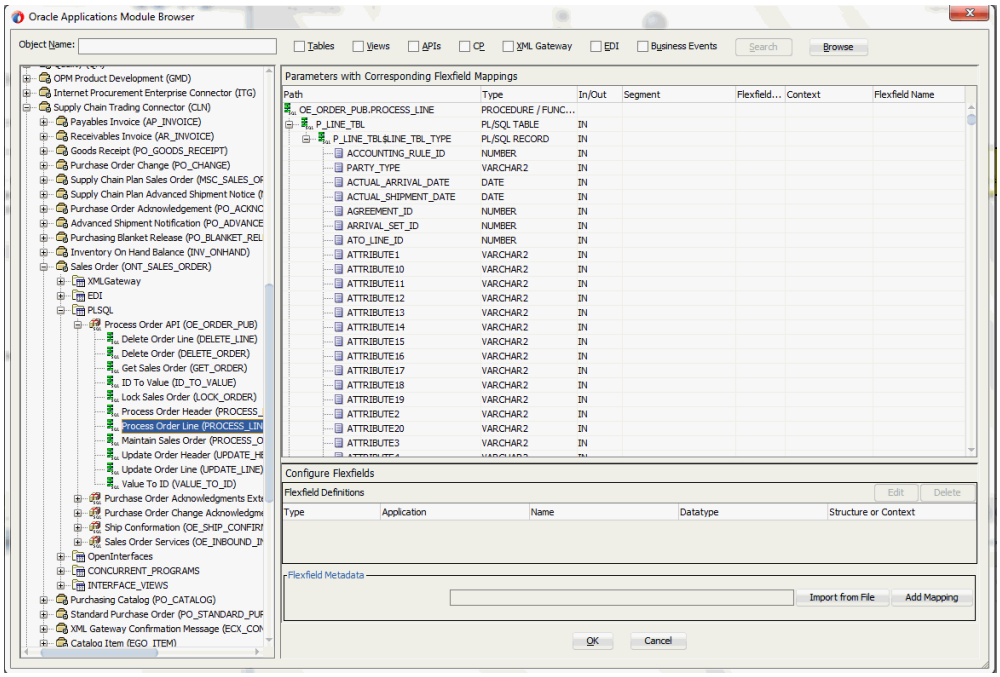
In this example, click **Next** in the Operation page to open the Oracle E-Business Suite Module Browser.

5. Select a PL/SQL API. For example, select `Process Order Line (PROCESS_LINE)` API from *Product Families > Supply Chain Management (SCM_PF)> Supply Chain Trading Connector(CLN) > Sales Order (ONT_SALES_ORDER) > PLSQL > Process Order API(OE_ORDER_PUB) > Process Order Line (PROCESS_LINE)*.

   Note: Use the Search option to quickly find the required objects. Enter the required database object name in the **Object Name** field and select **APIs**. And then, 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.

Selected API Displayed in the Oracle E-Business Suite Module Browser



Please note that Oracle E-Business Suite Adapter provides flexfield support for PL/SQL APIs. Once the selected API is displayed in the module browser, you can either configure flexfield mappings for the API that has flexfields defined in Oracle E-Business Suite or use current data without further flexfield configuration.

Configuring Flexfield Mapping Data for the Selected API

Perform either one of the following tasks in the Oracle E-Business Suite Module Browser:

- Use the current data without flexfield mapping configuration by clicking **OK** to continue.

- Configure flexfield mapping data by either using an existing flexfield mapping or creating a new mapping through a flexfield wizard if desired.

  Creating a New Mapping

  You can create a new mapping for a selected API by clicking **Add Mapping** in the Flexfield Metadata section. A new flexfield mapping wizard appears

guiding you through each configuration page where you can add key and descriptive flexfields, as well as configure flexfield mapping between the selected API parameters and flexfields defined in the Oracle E-Business Suite instance. For more information on how to create a new mapping for PL/SQL APIs, see:

- Adding or Configuring a New Mapping, page 4-37
- Configuring Key Flexfield Mappings for a PL/SQL API, page 4-38
- Configuring Descriptive Flexfield Mappings for a PL/SQL API, page 4-59

For more information on flexfield support for PL/SQL APIs and how to modify existing mapping configuration, see Flexfield Support for PL/SQL APIs and Open Interface Tables, page 4-27.

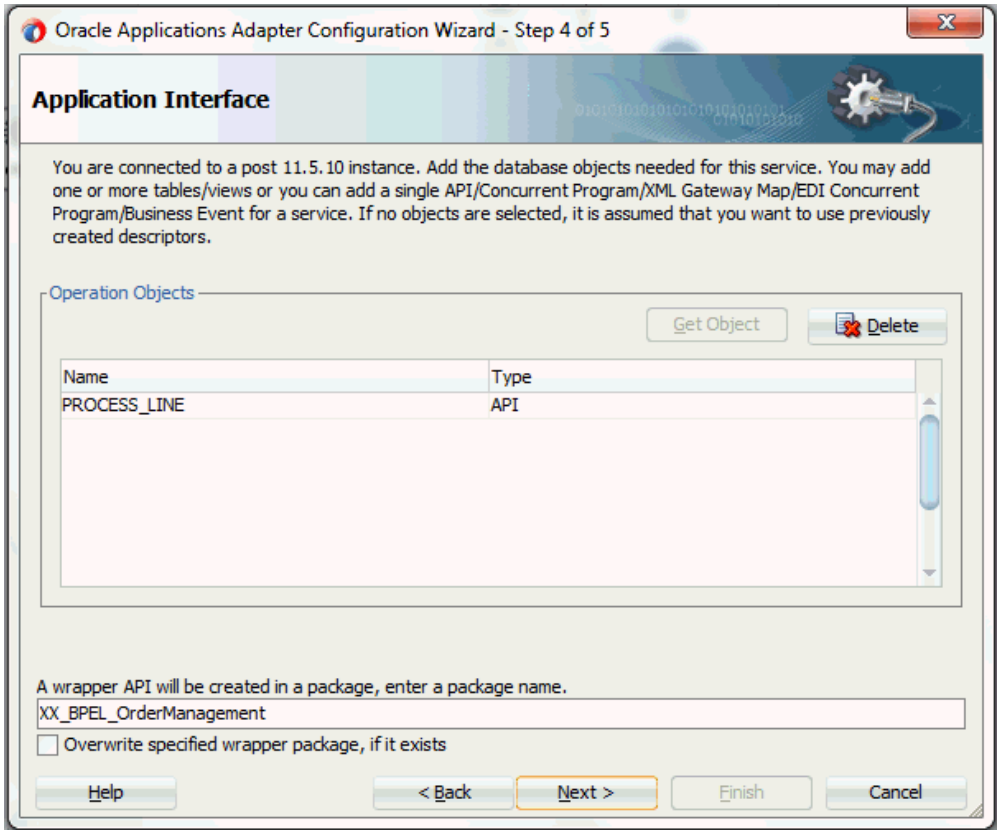
6. In this example, click **OK** to continue.

   Please note that the input and the output parameters of the Stored Procedure contains complex data types that are not readily mapped to JDBC types. The Adapter wizard provides a mechanism that detects when these types are used and then invokes Oracle JPublisher to generate the necessary wrappers automatically. Oracle JPublisher generates two SQL files, one to create schema objects, and another to drop them. The SQL that creates the schema objects is automatically executed from within the wizard to create the schema objects in the database schema before the XSD is generated.

   The Adapter Service points to the wrapper Stored Procedure instead of the Process Order Line Stored Procedure. The Adapter design-time automatically loads the wrapper procedure onto the underlying database.

7. In the Application Interface page, click **Next**.

Adapter Configuration Wizard - Application Interface Page



8. In the Finish page, click **Finish** to complete the process of configuring a partner link for an application service.

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

Adding a Partner Link for File Adapter

Use this step to configure a BPEL process by synchronously reading an existing purchase order to obtain the order details.

To add a Partner Link for File Adapter to read order details:

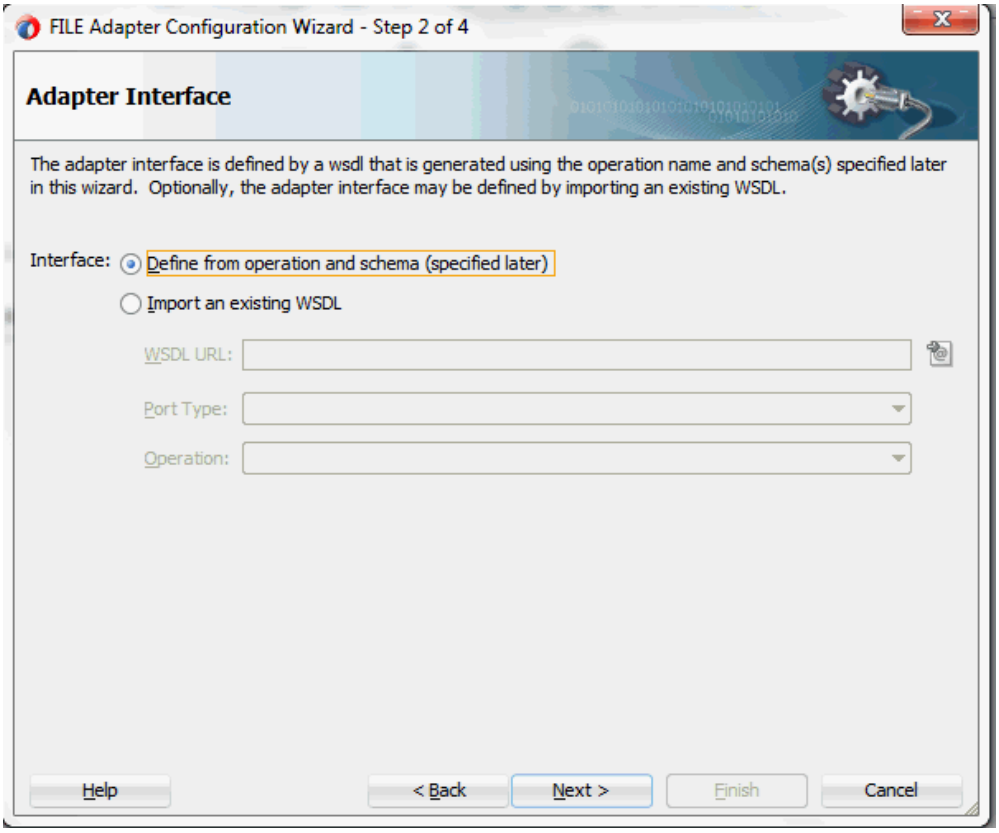
1. In Oracle JDeveloper BPEL Designer, click **BPEL Services** in the Component Palette.

   Drag and drop **File Adapter** from the **BPEL Services** list into the right Partner Link swim lane of the process diagram. The File Adapter Reference page appears.

   Enter a name for the file adapter service, such as `getOrderDetails`.

2. Click **Next**. The Adapter Interface page appears.

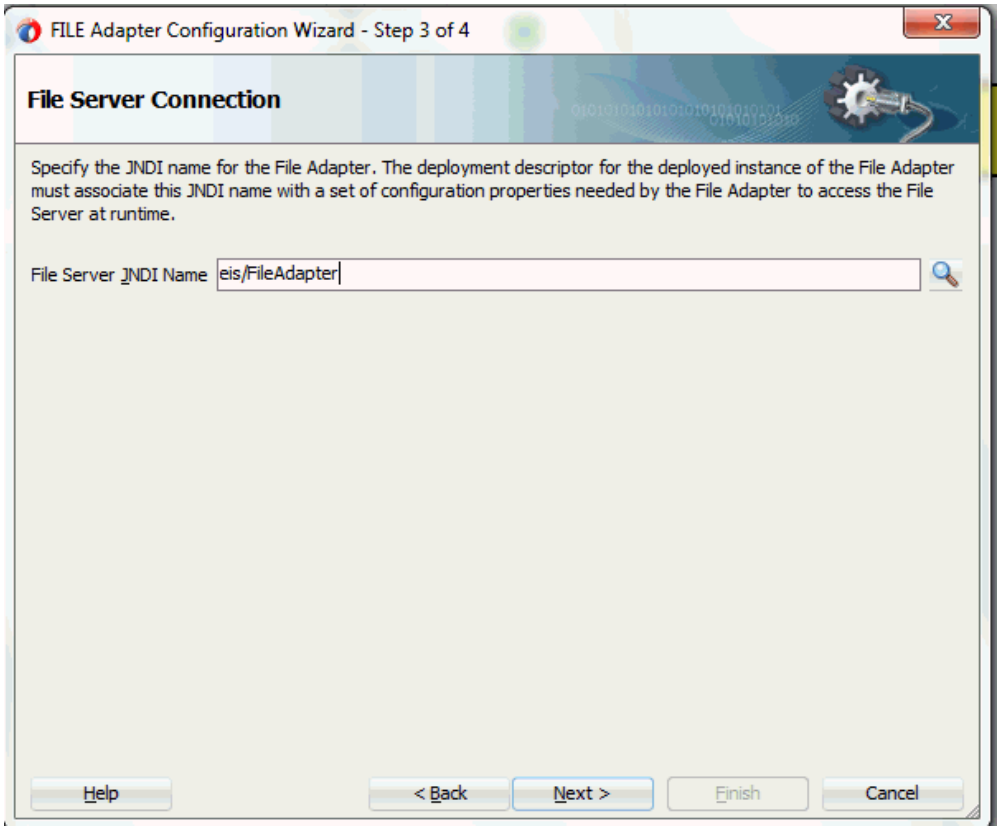
Specifying the Adapter Interface



Select the **Define from operation and schema (specified later)** radio button and click **Next**.

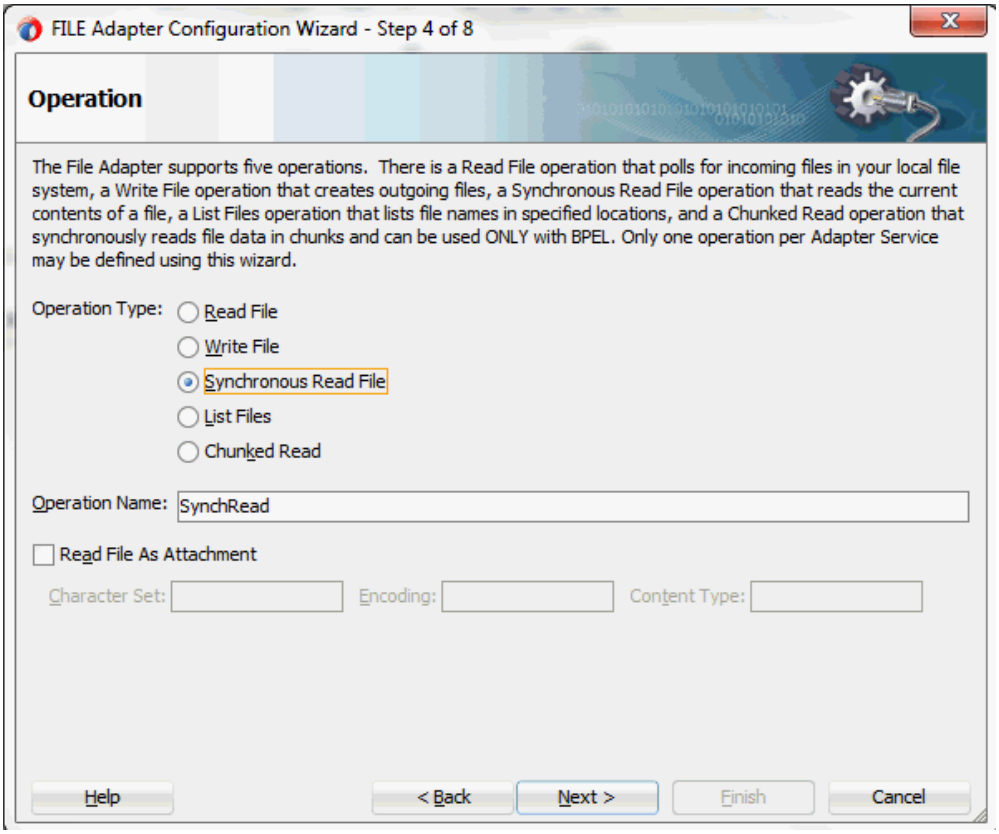
The File Server Connection page appears. Leave the default value unchanged for the File Server JNDI Name field.

Specifying File Server JNDI Name



3. In the Operation page, specify the operation type. For example, select the **Synchronous Read File** radio button. This automatically populates the **Operation Name** field.

Specifying the Operation

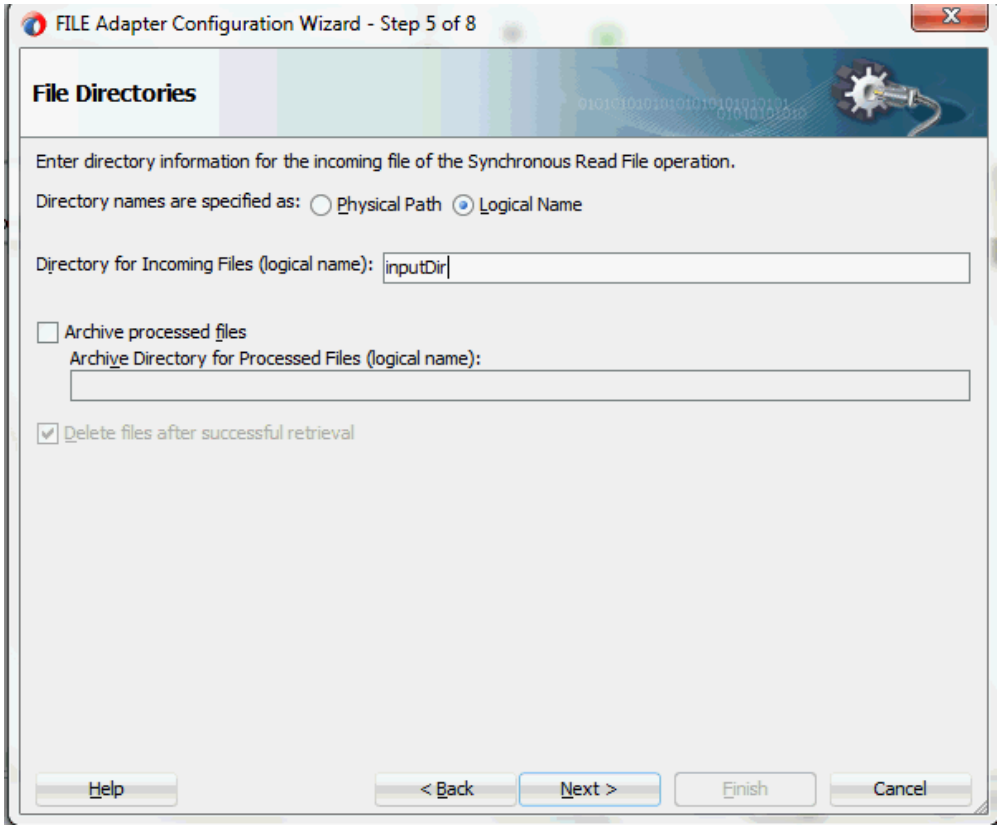


Click **Next** to access the File Directories page.

4. Select the **Logical Name** radio button and specify directory for incoming files, such as `inputDir`.

   Ensure the **Delete files after successful retrieval** check box is not selected.

Configuring the Input File



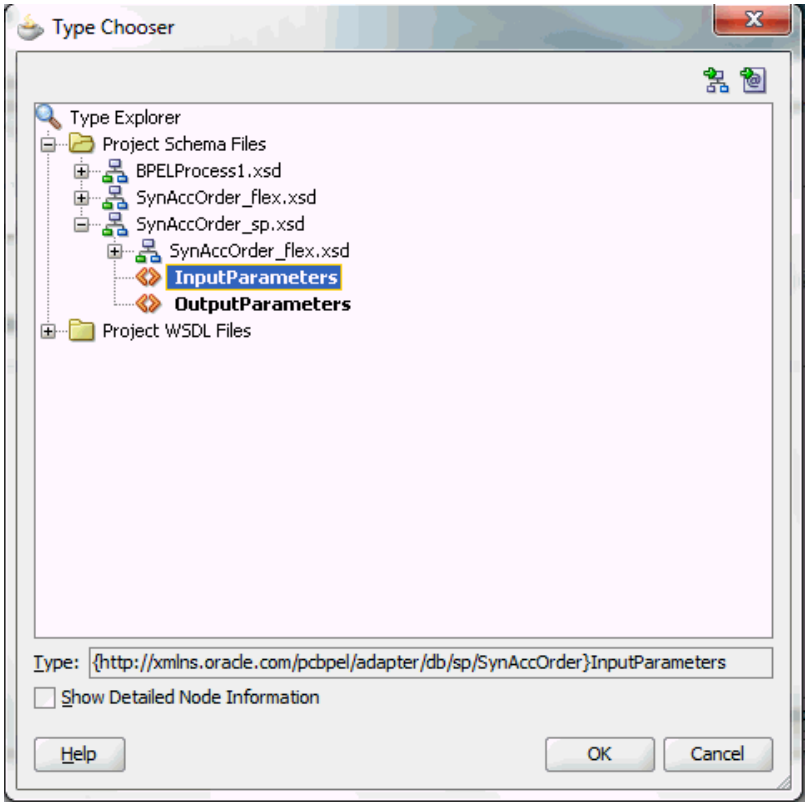
Click **Next** to open the File Name page.

5. Enter the name of the file for the synchronous read file operation. For example, enter `changeorder_data.xml`. Click **Next** to open the Messages page.

6. Select the 'browse for schema file' icon on the top right corner to open the Type Chooser window.

   Click Type Explorer and select *Project WSDL Files* > *OrderManagement.wsdl* > *Inline Schemas* > *schema* > *InputParameters*.

   > **Note:** If the selected API is configured with flexfields, select the flexfield schema (`<servicename>_sp.xsd`) instead from the Type Chooser.
   >
   > For example, `SynAccOrder` partner link service is created with descriptive flexfields. Select **Project Schema Files > SynAccOrder_sp.xsd > SynAccOrder_flex.xsd > InputParameters**.

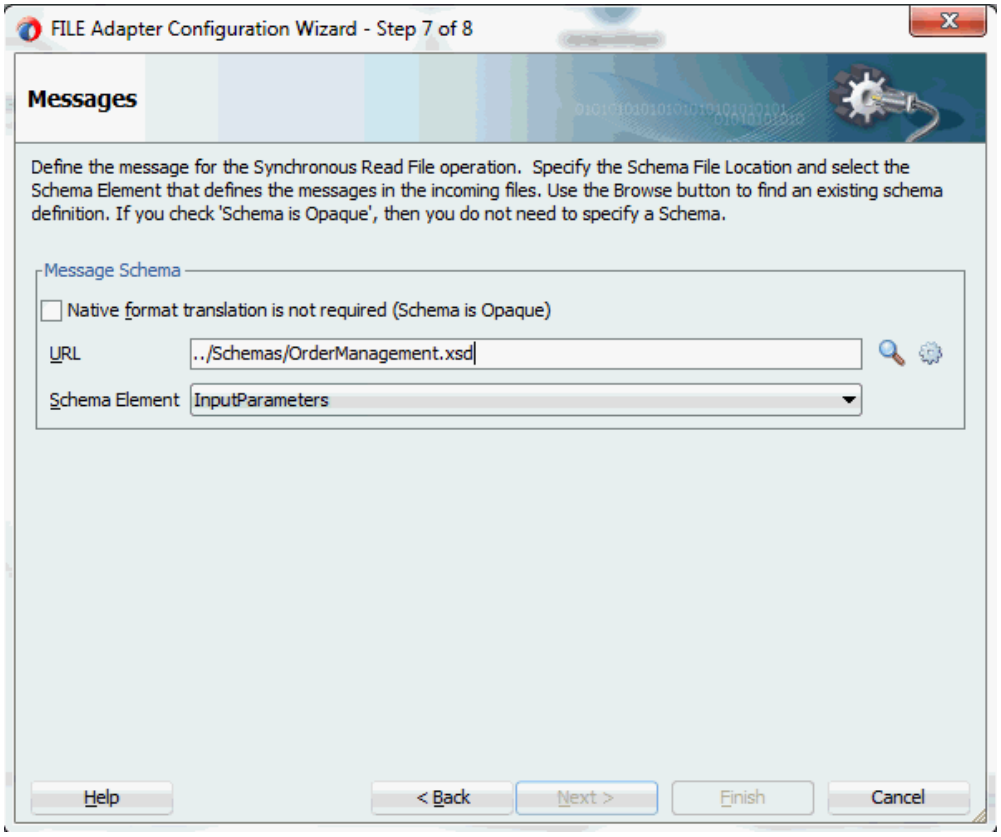


Click **OK**. The selected `SynAccOrder_sp.xsd` schema is displayed as URL and `InputParameters` is displayed as Schema Element.

For more information on flexfield support for PL/SQL APIs and how to modify existing configuration, see Flexfield Support for PL/SQL APIs, page 4-27.

The selected schema information will be automatically populated in the URL and Schema Element fields.

Specifying Message Schema



7. 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 `getOrderDetails.wsdl`.

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

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

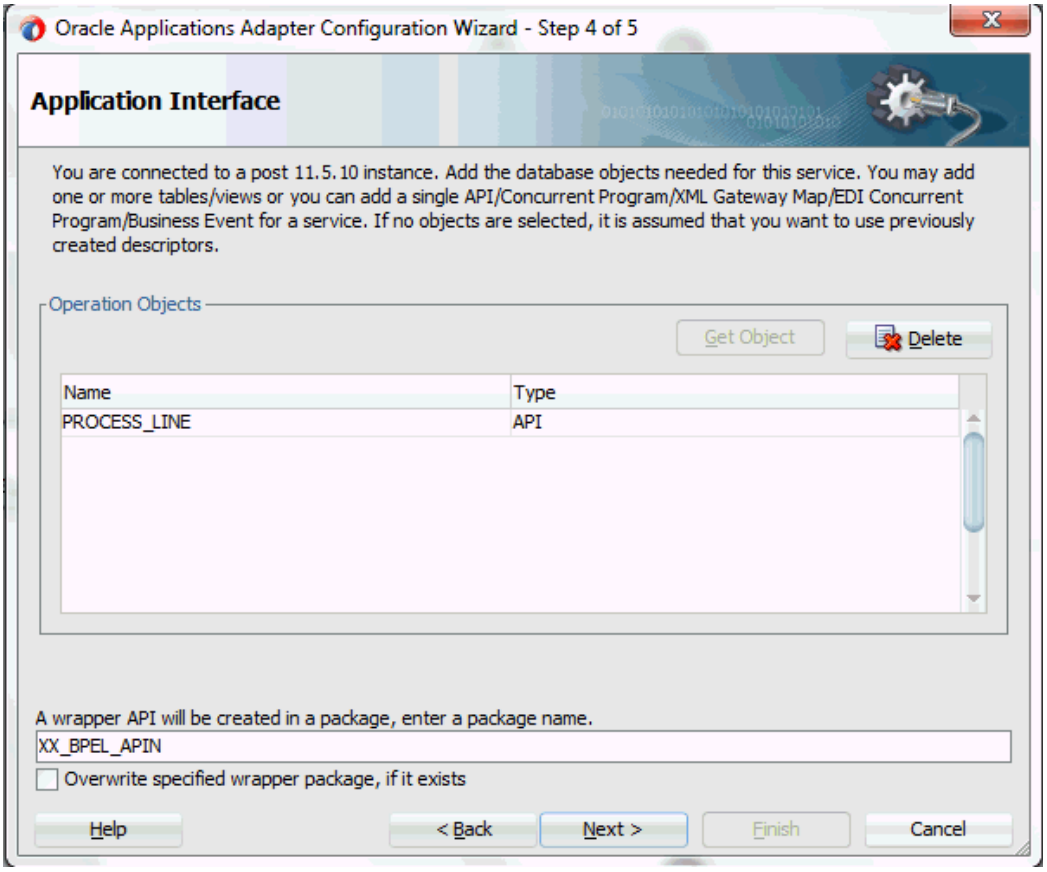
The generated SQL file that creates the wrapper API also creates the required schema objects. The types of the wrapper API's parameters will be those of the new schema object types. The wrapper package will contain conversion APIs to convert between the base PL/SQL type and the new schema object types.

> **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 2 of Adding a Partner Link, page 9-9. 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



> **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.*

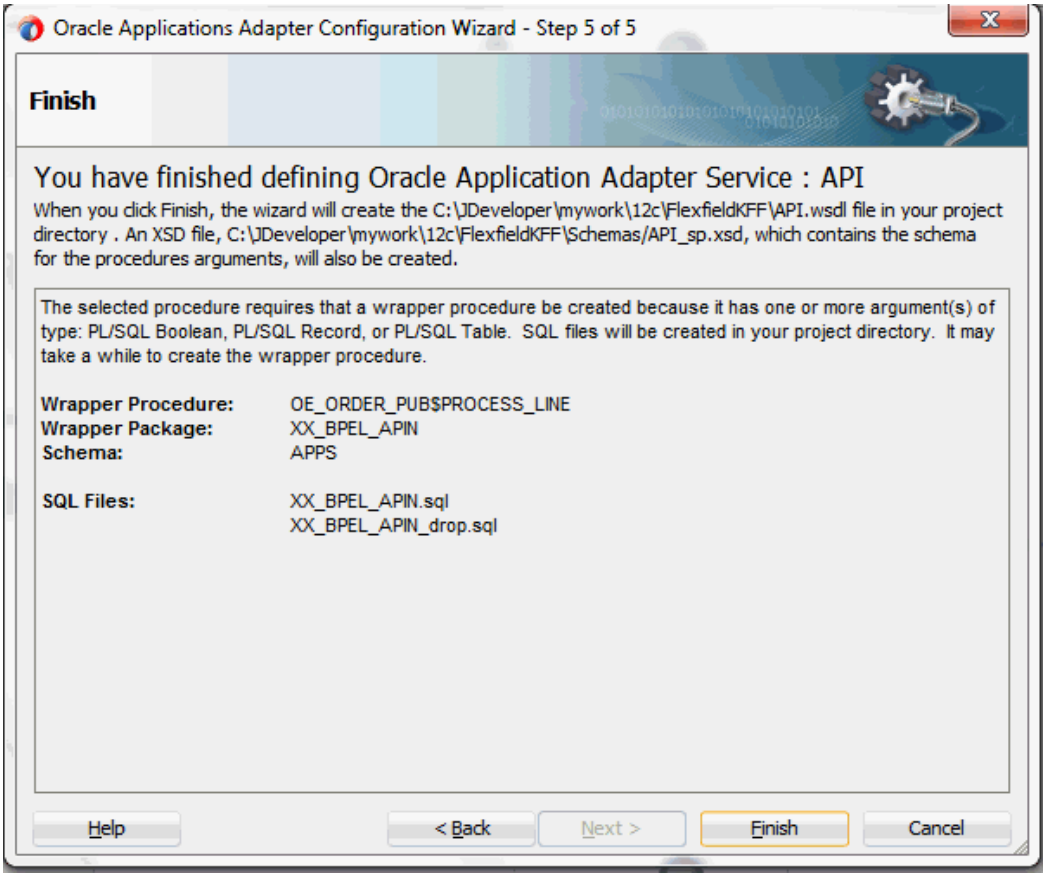
Wrapper APIs follow the naming convention of Oracle JPublisher. For example, if the original PL/SQL API was called `CREATE_EMPLOYEE` and was a root-level API, then the wrapper API would be named `TOPLEVEL$CREATE_EMPLOYEE`. If the original PL/SQL API is in a package called `EMPLOYEE`, then the wrapper API would be named `EMPLOYEE$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



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 Activities

This step is to configure three Invoke activities:

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

2. To initialize the existing purchase order by invoking `initLineRec` partner link.

3. To update the purchase order quantity information to Oracle E-Business Suite by invoking `OrderManagement` partner link.

To add the first Invoke activity for a partner link to get order details:

1. In Oracle JDeveloper BPEL Designer, expand the **BPEL Constructs** from the Component Palette. Drag and drop the first **Invoke** activity from the **Web Service** section into the center swim lane of the process diagram, between the **receiveInput** and **callbackClient** activities.

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

3. Enter a name for the Invoke activity. In the Input tab, ensure the **Input Variable** radio button is selected. 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**, then enter a name for the variable. You can also accept the default name. Click **OK**.

5. In the Output tab, ensure the **Output Variable** radio button is selected. Click the **Create** icon next to the **Output Variable** field to create a new variable. The Create Variable dialog box appears.

   Select **Global Variable**, 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 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 initialize the existing purchase order:

1. In Oracle JDeveloper BPEL Designer, expand the **BPEL Constructs** from the Component Palette. Drag and drop the second **Invoke** activity from the **Web Service** section into the center swim lane of the process diagram, between the first **Invoke** and **callbackClient** activities.

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

3. Repeat Step 3 to Step 5 described in the first Invoke activity procedure to complete the second Invoke activity.

To add the third Invoke activity for a partner link to update the order quantity information to Oracle E-Business Suite:

1. In Oracle JDeveloper BPEL Designer, expand the **BPEL Constructs** from the Component Palette. Drag and drop the third **Invoke** activity from the **Web Service** section into the center swim lane of the process diagram, between the second **Invoke** and **callbackClient** activities.

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

3. Repeat Step 3 to Step 5 described in the first Invoke activity procedure.

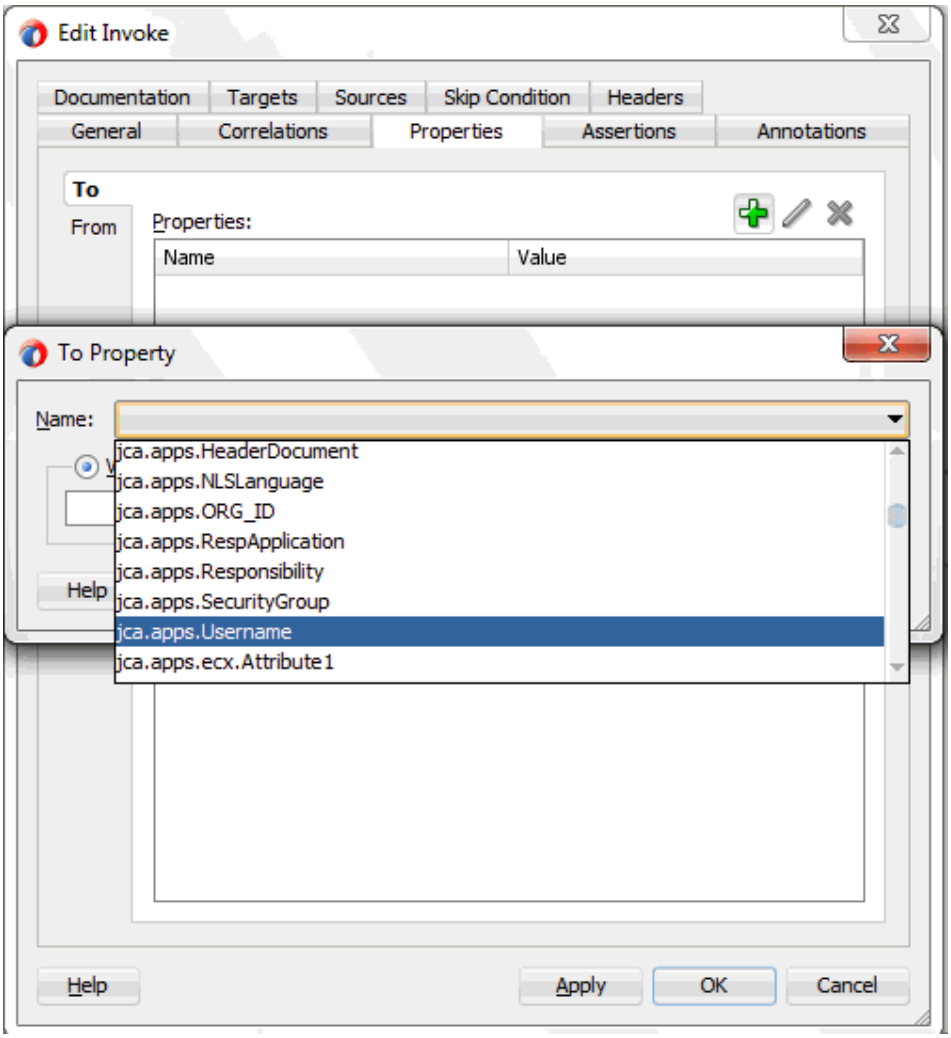
4. **Setting Header Properties for Applications Context**

   Use the following steps to set the header message properties required to pass applications context required to complete the BPEL process:

   1. Select the Properties tab in the Edit Invoke dialog box.

   2. Click the **Create** icon. The To Property dialog appears.

   3. Select the `jca.apps.Username` property from the Name drop-down list.



4. Select the **Expression** radio button and enter `'OPERATIONS'` as the property value.

   Click **OK**.

5. Repeat Step 2 to Step 4 to assign `'Order Management Super User, Vision Operations (USA)'` for `jca.apps.Responsibility`.

5. Click **Apply** and then **OK** to complete the **Invoke** activity.

Configuring a Transform Activity

The Transform activity can be used to configure the parameters for the input and output variables. The Transform activity can also be used if variable values need to be transformed before updating them in Oracle E-Business Suite.

To configure a Transform activity:

Before add a Transform activity, use the following steps to add variables:

1. From the **structure** panel, select `ChangeOrderAPI.bpel` > Variables > Process > Variables folder. Right click on Variables folder and select **Create Variable** from the drop-down menu.

2. In the Create Variable dialog, enter `temp_miss_line` in the variable Name field.

3. Select the **Message Type** radio button and click the **Browse Message Type** icon to open the Type Chooser.

   Select `args_in_msg` from *Message Types* > *Project WSDL Files* > *OrderManagement.wsdl* > *Message Types*. Click OK to close the Type Chooser.

   The selected information will be populated in the Message Type field in the Create Variable dialog.

Use the following step to configure the Transform activity:

1. In Oracle JDeveloper BPEL Designer, expand the **Oracle Extensions** from the Component Palette. Drag and drop **Transform** from the **Transform** section into center swim lane of the process map window. The **Transform** activity should be placed in between the second and third**Invoke** activities.

   This Transform activity is used to set the output of `initLineRec` service to the input of the `OrderManagement` service.

2. Double-click **Transform** in the process map to open the Transform dialog. The Transformation tab is selected by default.

   Click the General tab to name this Transform activity as `set_Miss_Line`.

3. In the Transformation tab, click the **Create** icon to open the Source Variable dialog.

   Select the `Invoke_initLineRec_OutputVariable` from the Source Variable drop-down list, and select `OutputParameters` element as the Source Part value.

   Click **OK**.

4. Select the `Invoke_OrderManagement_InputVariable` from the Target Variable drop-down list. The Target Part value of the variable is also selected.

5. Click the **Create** icon next to the Mapper File field to create a new transformation mapping file. Mapper File specifies the file in which you create the mappings using the XSLT Mapper Transformation tool.

6. The transformation mapping file appears. The **Design** view appears by default.

7. You can define the parameter values in the **Design** view. Drag a string function to

the Design area. Connect the function to the appropriate parameter for which you want to define a value.

For example, link `db:OE_ORDER_PUB-24GET_G_MISS_LINE` from the source variable to `db:P_LINE_TBL_ITEM` as the target variable.

> **Note:** You can use an input parameter value from the source variable, transform it using a string function, and use it as the input parameter value for the target variable.

8. Click **OK** to create the mapping from source variable to target variable.

Configuring an Assign Activity

Use the Assign activity to pass the output of `getOrderDetails` service and `initLineRec` service as an input to the `OrderManagement` service.

To add an Assign activity:

1. In Oracle JDeveloper BPEL Designer, expand the **BPEL Constructs** from the Component Palette. Drag and drop the **Assign** activity from the **Basic Activities** section into the center swim lane of the process diagram right after the **Transform** activity that you just created earlier.

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

   Click the General tab to enter a name for the Assign activity. For example, `setOrderChanges`.

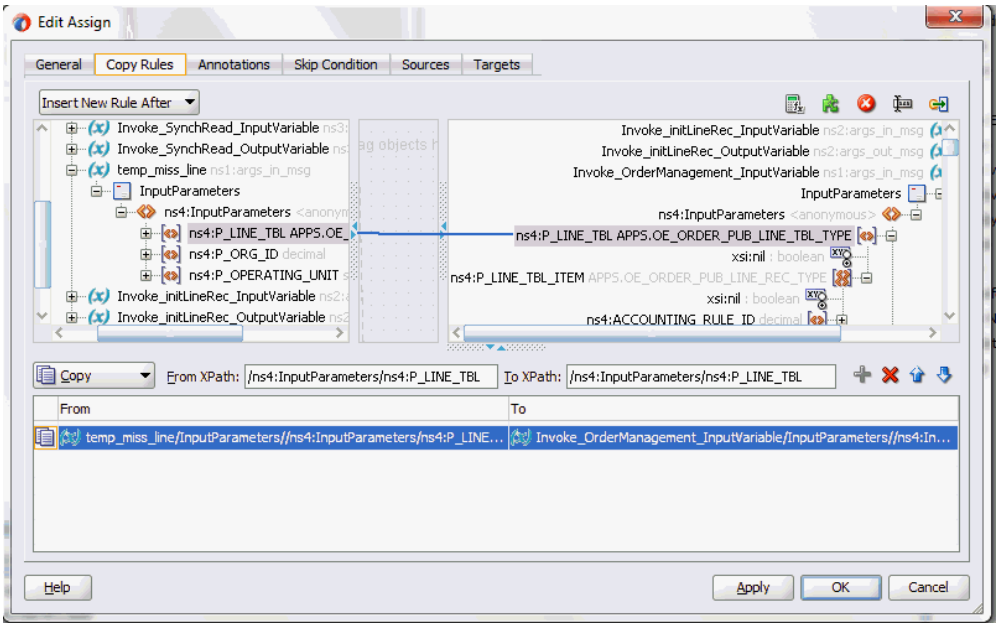
3. Select the Copy Rules tab and expand the target trees:

   - In the From navigation tree, navigate to **Variable > Process > Variables > Temp_miss_line >Input Parameters** and select **ns4:P_LINE_TBL**.

   - In the To navigation tree, navigate to **Variable > Process > Variables > Invoke_OrderManagement_InputVariable > Input Parameters** and select **ns4:P_LINE_TBL**.

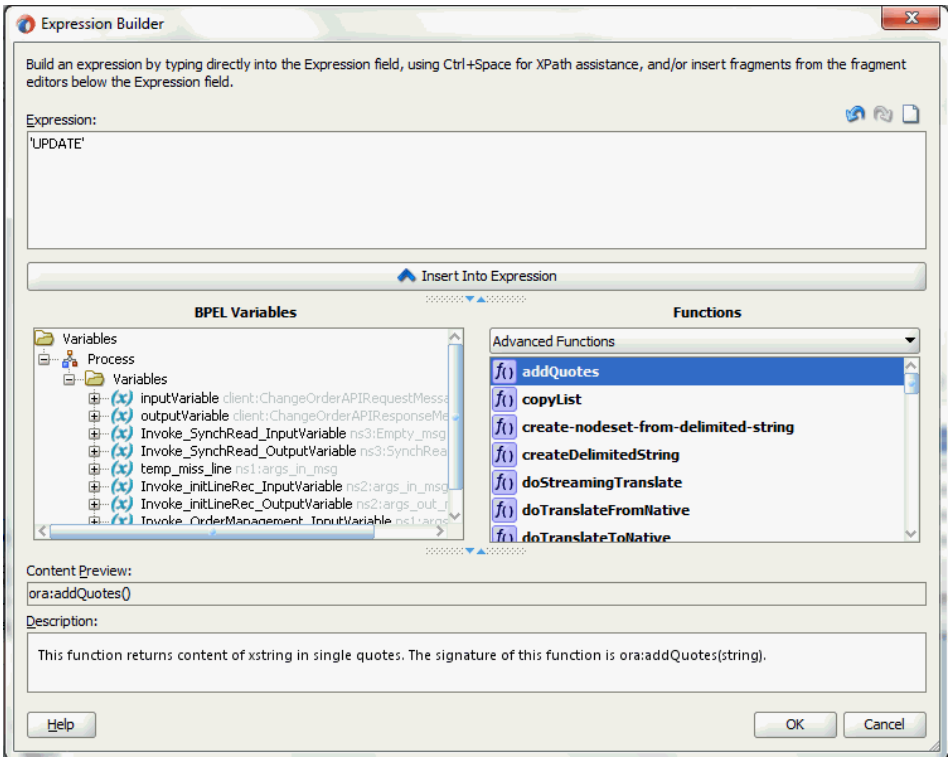
   - Click **OK**. The Edit Assign dialog box appears.

   Drag the source node (ns4:P_LINE_TBL) to connect to the target node (ns4:P_LINE_TBL) that you just specified. This creates a line that connects the source and target nodes. The copy rule is displayed in the From and To sections at the bottom of the Edit Assign dialog box.



4. Enter the second pair of parameters:

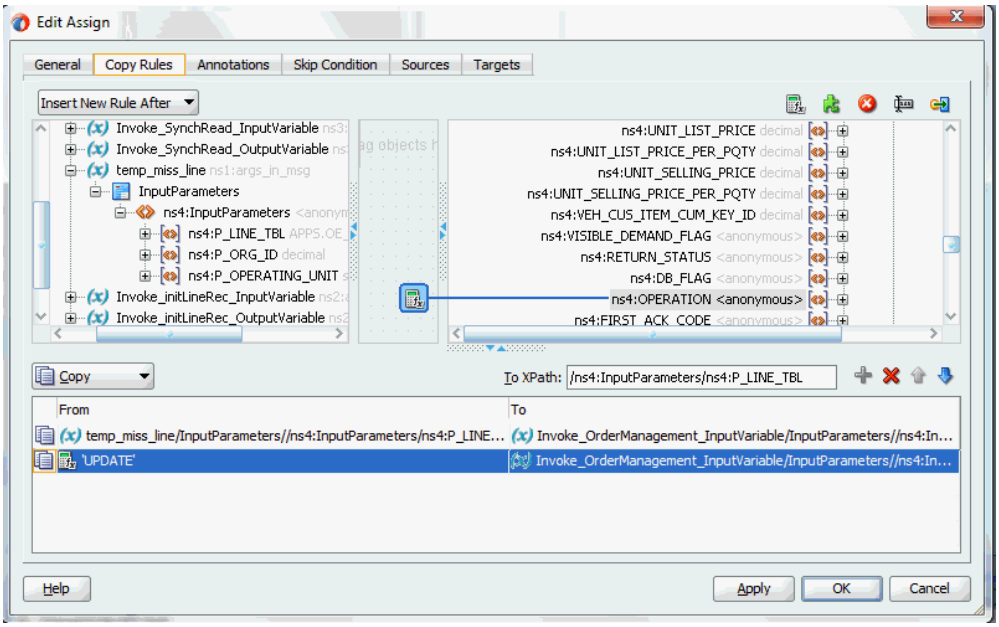
   - Click the Expression icon to invoke the Expression Builder dialog.



Enter 'UPDATE' in the Expression box. Click **OK**. The Expression icon with the expression value ('UPDATE') appears in the center of the Edit Assign dialog, between the From and To navigation tree nodes.

- In the To navigation tree, navigate to **Variable > Process > Variables > Invoke_OrderManagement_InputVariable > Input Parameters > P_LINE_TBL > P_LINE_TBL_ITEM** and select **ns4:OPERATION**.

Drag the Expression icon to connect to the target node (ns4:OPERATION) that you just identified. This creates a line that connects the source and target nodes. The copy rule is displayed in the From and To sections at the bottom of the Edit Assign dialog box.



5. Enter the third pair of parameters:

   - Click the Expression icon to invoke the Expression Builder dialog. Enter 'Update Order Quantity' in the Expression box. Click **OK**. The Expression icon with the expression value ('Update Order Quantity') appears in the center of the Edit Assign dialog, between the From and To navigation tree nodes.

   - In the To navigation tree, navigate to **Variable > Process > Variables > Invoke_OrderManagement_InputVariable > P_LINE_TBL > P_LINE_TBL_ITEM** and select **ns4:CHANGE_REASON**.

   Drag the Expression icon to connect to the target node (ns4:CHANGE_REASON) that you just identified. This creates a line that connects the source and target nodes. The copy rule is displayed in the From and To sections at the bottom of the Edit Assign dialog box.

6. Enter the fourth pair of parameters:

   - In the From navigation tree, navigate to **Variable > Process > Variables > Invoke_SyncRead_OutputVariable>Input Parameters > P_LINE_TBL >P_LINE_TBL_ITEM** and select **ns4:LINE_ID**.

   - In the To navigation tree, navigate to **Variable > Process > Variables > Invoke_OrderManagement_InputVariable > P_LINE_TBL > P_LINE_TBL_ITEM** and select **ns4:LINE_ID**.

   Drag the source node (ns4:LINE_ID) to connect to the target node (ns4:LINE_ID) that you just specified. This creates a line that connects the source and target nodes.

The copy rule is displayed in the From and To sections at the bottom of the Edit Assign dialog box.

7. Enter the fifth pair of parameters:

   - In the From navigation tree, navigate to **Variable > Process > Variables > Invoke_SyncRead_OutputVariable>Input Parameters > P_LINE_TBL >P_LINE_TBL_ITEM** and select **ns4:ORDERED_QUANTITY**.

   - In the To navigation tree, navigate to **Variable > Process > Variables > Invoke_OrderManagement_InputVariable > P_LINE_TBL > P_LINE_TBL_ITEM** and select **ns4:ORDERED_QUANTITY**.

   Drag the source node (ns4:ORDERED_QUANTITY) to connect to the target node (nns4:ORDERED_QUANTITY) that you just specified. This creates a line that connects the source and target nodes. The copy rule is displayed in the From and To sections at the bottom of the Edit Assign dialog box.

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

Run-Time Tasks for PL/SQL APIs

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

1. Deploy the SOA Composite application with BPEL process, page 9-39

2. Test the deployed SOA Composite application with BPEL process, page 9-42

Deploying the SOA Composite Application with BPEL Process

To invoke the services from the BPEL client contained in the SOA composite, the SOA composite needs to be deployed to the Oracle WebLogic managed server. This can be achieved using Oracle JDeveloper. Once the composite is deployed, it can be tested from the Oracle Enterprise Manager Fusion Middleware Control Console.

Prerequisites

Before deploying the SOA composite with BPEL process using Oracle JDeveloper, you must have established the connectivity between the design-time environment and the run-time server. For more information, see Configuring the Data Source in Oracle WebLogic Server, page 11-1 and Creating an Application Server Connection, page 11-10.

> **Note:** If a local instance of the WebLogic Server is used, start the WebLogic Server by selecting **Run > Start Server Instance** from Oracle

JDeveloper. Once the WebLogic Admin Server "DefaultServer" instance is successfully started, the <Server started in Running mode> and DefaultServer started message in the Running:DefaultServer and Messages logs should appear.

To deploy the SOA Composite application with BPEL process:

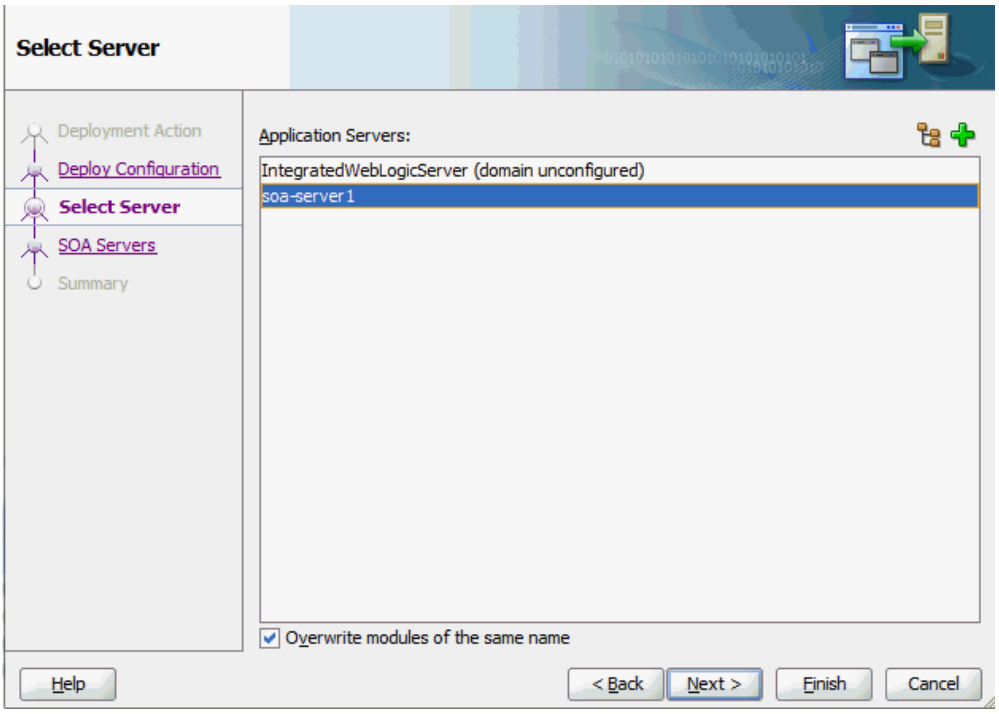
1. Select the SOA Composite project in the Applications window.

2. Right-click the project name, and then select **Deploy > [project name] > [serverConnection]** from the menu that appears.

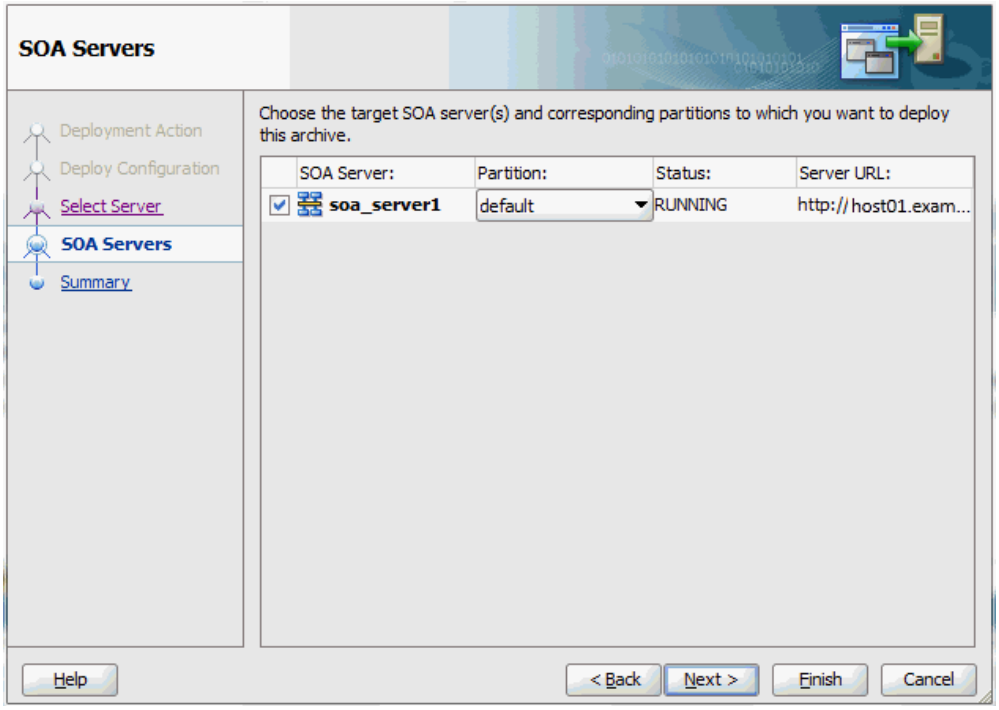
   For example, you can select **Deploy > ChangeOrderAPI> soa-server1** to deploy the process if you have the connection set up appropriately.

   > **Note:** If this is the first time to set up server connection, then the Deployment Action window appears. Select 'Deploy to Application Server' and click **Next**.
   >
   > In the Deploy Configuration window, ensure the following information is selected before clicking **Next** to add a new application server:
   >
   > - New Revision ID: 1.0
   > - Mark composite revision as default: Select this check box.
   > - Overwrite any existing composites with the same revision ID: Select this check box.
   >
   > The steps to create a new Oracle WebLogic Server connection from JDeveloper are covered in Creating an Application Server Connection, page 11-10.

3. In the Select Server page, select 'soa-server1' that you have established the server connection earlier. Click **Next**.



4. In the SOA Servers page, accept the default target SOA Server ('soa-server1') selection.



Click **Next** and **Finish**.

If you are deploying the composite for the first time from your Oracle JDeveloper session, the Authorization Request window appears. Enter username and password information specified during Oracle SOA Suite installation. Click **OK**.

5. Deployment processing starts. Monitor deployment progress and check for successful compilation in the SOA - Log window as well as in the Deployment - Log window.

Testing the Deployed SOA Composite Application with BPEL Process

Once the BPEL process contained in a SOA Composite application is deployed, you can manage and monitor the process from the Oracle Enterprise Manager Fusion Middleware Control Console. You can also test the process and the integration interface by manually initiating the process.

To test the deployed SOA Composite application with BPEL process:

1. Navigate to Oracle Enterprise Manager Fusion Middleware Control Console (`http://<servername>:<portnumber>/em`). The composite you deployed is displayed in the Applications Navigation tree.

2. Enter username (such as `weblogic`) and password and click **Login** to log in to a farm.

   You may need to select an appropriate target instance farm if there are multiple target Oracle Enterprise Manager Fusion Middleware Control Console farms.

3. From the Farm base domain, expand the **SOA > soa-infra > soa-infra (soa_server1) > default** to navigate through the SOA Infrastructure home page and menu to access your deployed SOA Composite applications running in the SOA Infrastructure for that managed server.

   > **Note:** The Farm menu always displays at the top of the navigator. As you expand the SOA folder in the navigator and click the links displayed beneath it, the SOA Infrastructure menu becomes available at the top of the page.

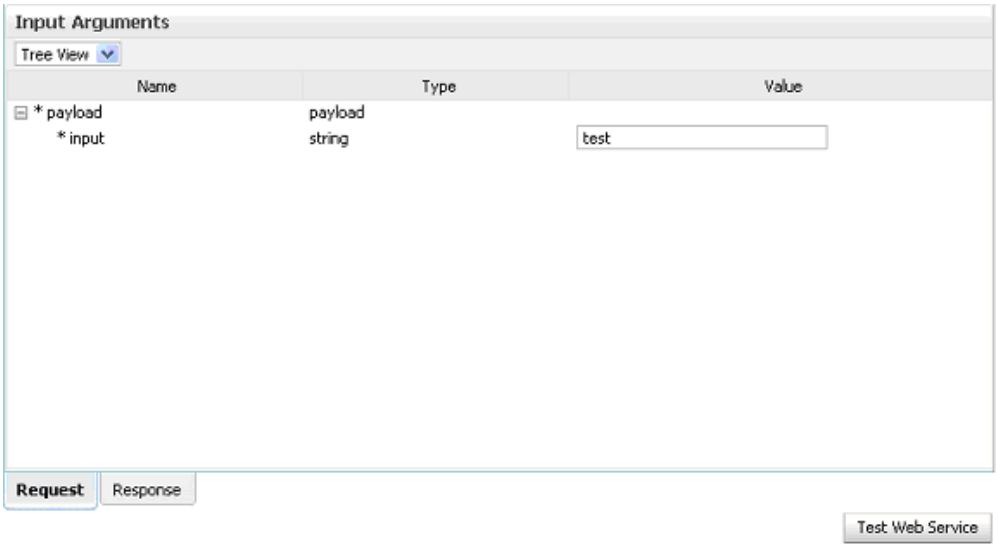
   Click the SOA Composite application that you want to initiate (such as 'ChangeOrderAPI') from the SOA Infrastructure.

   Click **Test** at the top of the page.

4. The Test Web Service page for initiating an instance appears. You can specify the XML payload data to use in the Input Arguments section.

   Enter the input string required by the process and click **Test Web Service** to initiate the process.

Testing Web Service

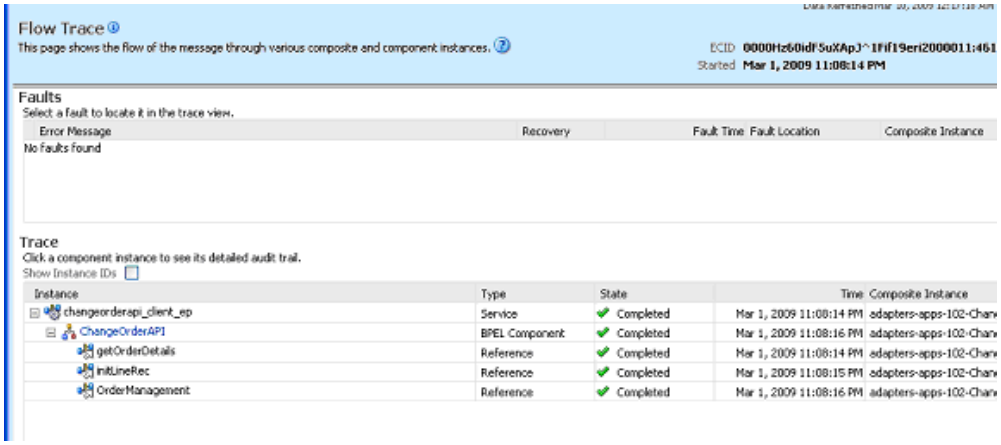


The test results appear in the Response tab upon completion.

5. Click **Launch Flow Trace** in the Test Web Service page. The Flow Trace page is displayed.

   In the Trace section, you should find the sequence of the message flow for the service binding component (`changeorderapi_client_ep`), BPEL component (`ChangeOrderAPI`), and reference binding components (`getOrderDetails`, `initLineRec`, and `OrderManagement`). All involved components have successfully received and processed messages.

Flow Trace Page



Click your BPEL service component instance link (such as `ChangeOrderAPI`) to display the Instances page where you can view execution details for the BPEL activities in the Audit Trail tab.

Click the Flow tab to check the BPEL process flow diagram. Click an activity of the process diagram to view the activity details and flow of the payload through the process.

Validating Records Using SQL

To confirm that the records have been written into the Oracle E-Business Suite applications, you can write SQL `select l.ordered_quantity from oe_order_lines_all l,oe_order_headers_all h where h.orig_sys_document_ref='<ORDER_ID> and h.header_id=l.header_id;` statements and fetch the results showing the `ordered_quantity` should be the same as the value provided in the `changeorder_data.xml` document.

Verifying Records in Oracle E-Business Suite

Alternatively, you can go to the specific module in Oracle E-Business Suite with Oracle Management Super User, Vision Operation (USA) responsibility.

To validate it in Oracle Order Management:

1. Log on to the Forms-based Oracle E-Business Suite applications with the Order Management, Super User responsibility.

2. Select Order Returns > Sales Order. Sales Order Forms would open up.

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

4. Select the Line Items tab to check the quantity of the order. It should be the

same as it is set in `changeorder_data.xml` file.

Troubleshooting

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 E-Business Suite 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 **weblogic-ra.xml** is properly configured.

Enable logging for Adapter to see if the issue is on the middleware side. For information on how to enable logging, see Enabling Logging for Adapters, page 4-125.

10

Using e-Commerce Gateway

This chapter covers the following topics:

- Overview of e-Commerce Gateway Integration
- Design-Time Tasks for e-Commerce Gateway
- Creating a New SOA Composite Application with BPEL Process
- Adding a Partner Link
- Adding a Partner Link for File Adapter
- Configuring Invoke Activities
- Configuring an Assign Activity
- Run-Time Tasks for e-Commerce Gateway
- Deploying the SOA Composite Application with BPEL Process
- Testing the Deployed SOA Composite Application with BPEL Process
- Verifying Records in Oracle E-Business Suite

Overview of e-Commerce Gateway Integration

Oracle e-Commerce Gateway provides a common, standards-based approach for Electronic Data Interchange (EDI) integration between Oracle E-Business Suite 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 E-Business Suite. 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 E-Business Suite. This allows for seamless flow of information in an ever expanding trading partner base.

Oracle e-Commerce Gateway includes prebuilt 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 prebuilt 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 part of the Oracle E-Business Suite Documentation Library which can be accessed on the Oracle Technology Network (OTN).

Oracle E-Business Suite Adapter 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

Oracle E-Business Suite Adapter 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 the tasks to configure Oracle E-Business Suite Adapter to use e-Commerce Gateway in Oracle JDeveloper.

SOA Composite Application with BPEL Process Scenario

This example uses IN: Ship Notice/Manifest (ECASNI) EDI concurrent program exposed as a Web service to support the inbound Ship Notice/Manifest transaction.

When a shipment request is received, the shipping details will be retrieved and then imported to the Purchasing system to create an electronic shipment notice as a pre-receipt.

When the SOA Composite application with BPEL process has been successfully executed after deployment, you should be able to validate if the shipment notice has been inserted in Oracle E-Business Suite with the same shipment identifier number.

Prerequisites to Configure e-Commerce Gateway

Populating Applications Context Header Variables

You need to populate certain variables in the BPEL PM in order to provide context

information for Oracle E-Business Suite. The context information is required for an EDI transaction in order for an Oracle E-Business Suite user that has sufficient privileges to run the program.

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 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. 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 Invoke activity.

For more information about applications context, see Supporting for Normalized Message Properties, page 4-7.

Based on the scenario described earlier, the following design-time tasks are discussed in this chapter.

1. Create a new SOA Composite application with BPEL process, page 10-3
2. Add a partner link, page 10-9
3. Add a partner link for File Adapter, page 10-14
4. Configure Invoke activities, page 10-20
5. Configure an Assign activity, page 10-25

Creating a New SOA Composite Application with BPEL Process

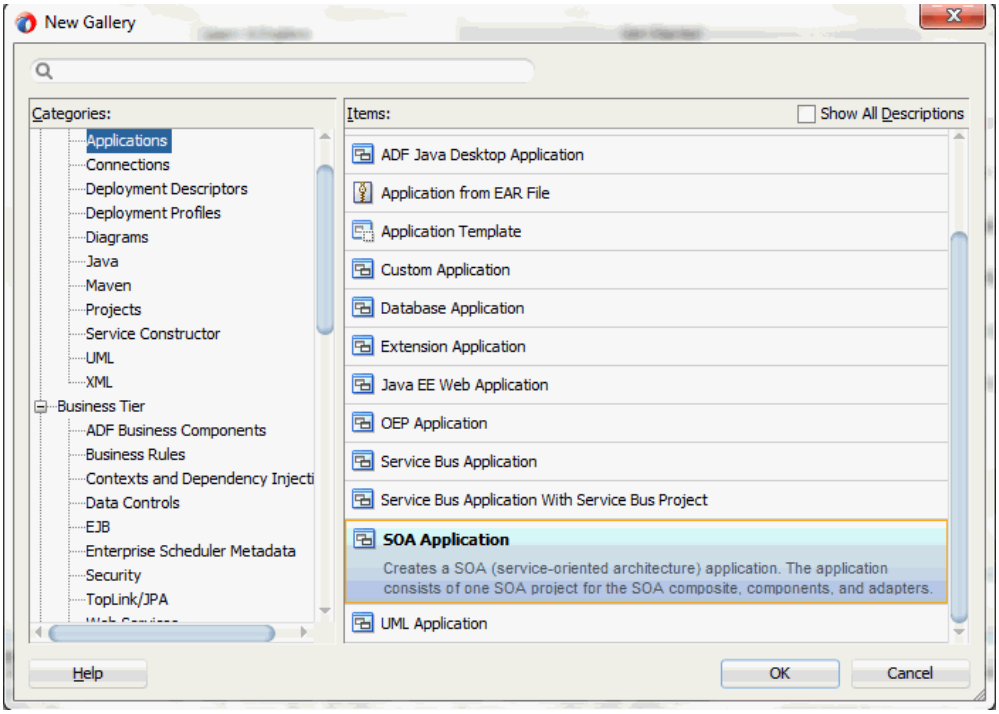
The first configuration task is to create a new SOA Composite application with BPEL process.

To create a new SOA Composite application with BPEL process:

1. Launch Oracle JDeveloper.
2. Click **New Application** in the Application Navigator.

   The New Gallery page appears. Select **SOA Application** from the Items list.

The New Gallery Page

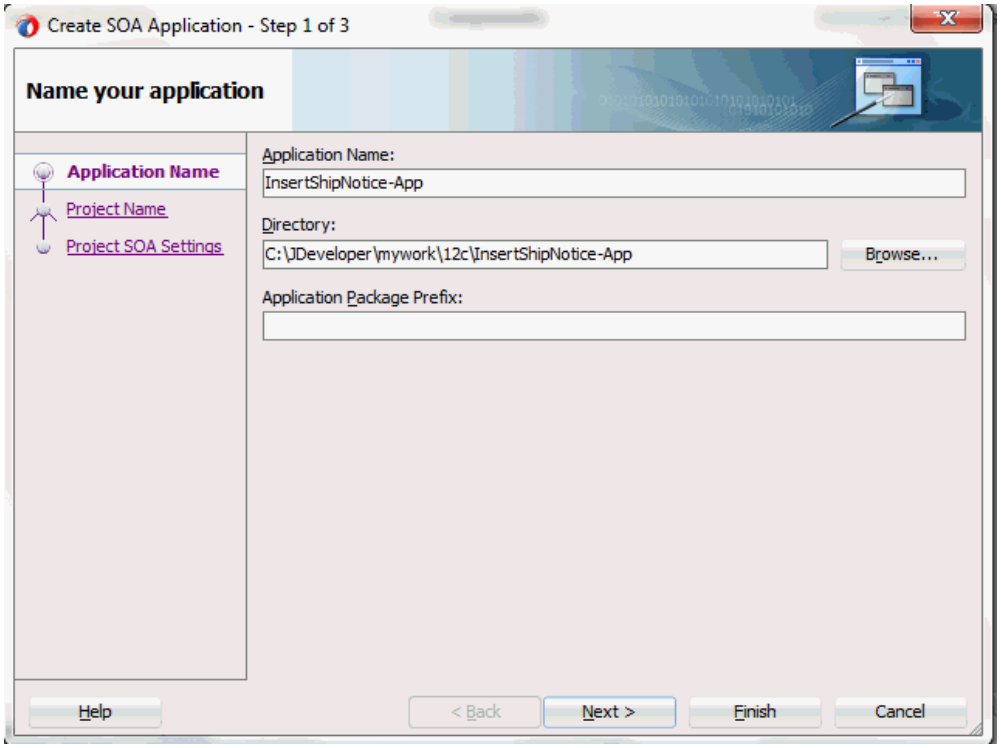


Click **OK**.

3. Click **New Application** in the Application Navigator.

   The Create SOA Application - Name your application page is displayed.

The Create SOA Application - Name your application Page



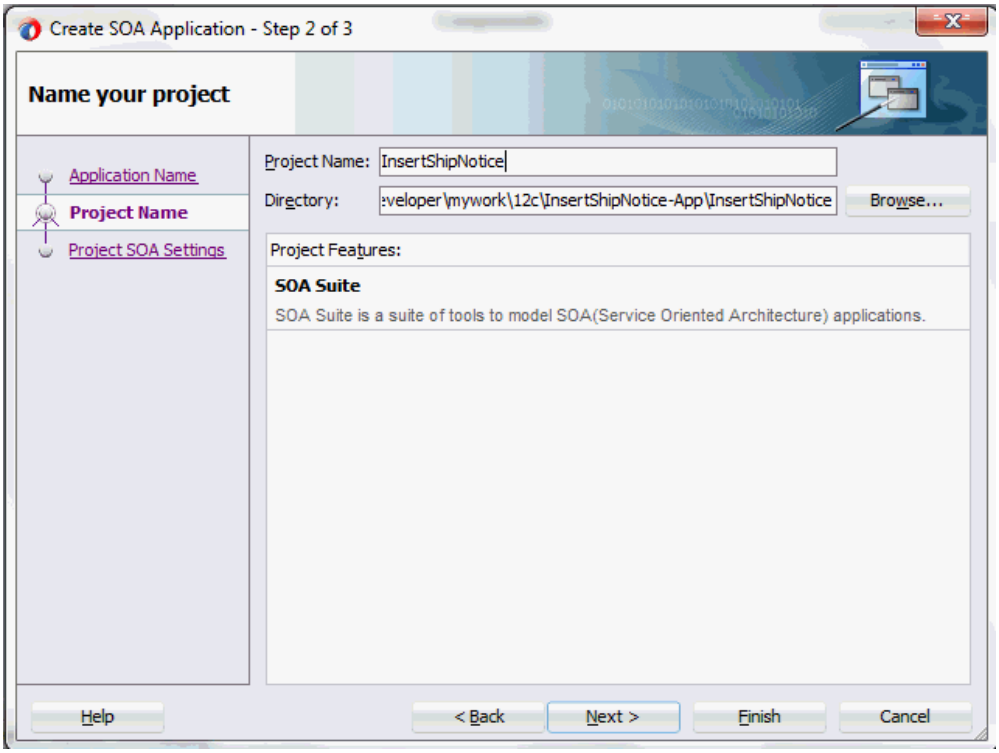
4. Enter an appropriate name for the application in the **Application Name** field.

   Click **Next**. The Create SOA Application - Name your project page is displayed.

5. Enter an appropriate name for the project in the **Project Name** field. For example, `InsertShipNotice`.

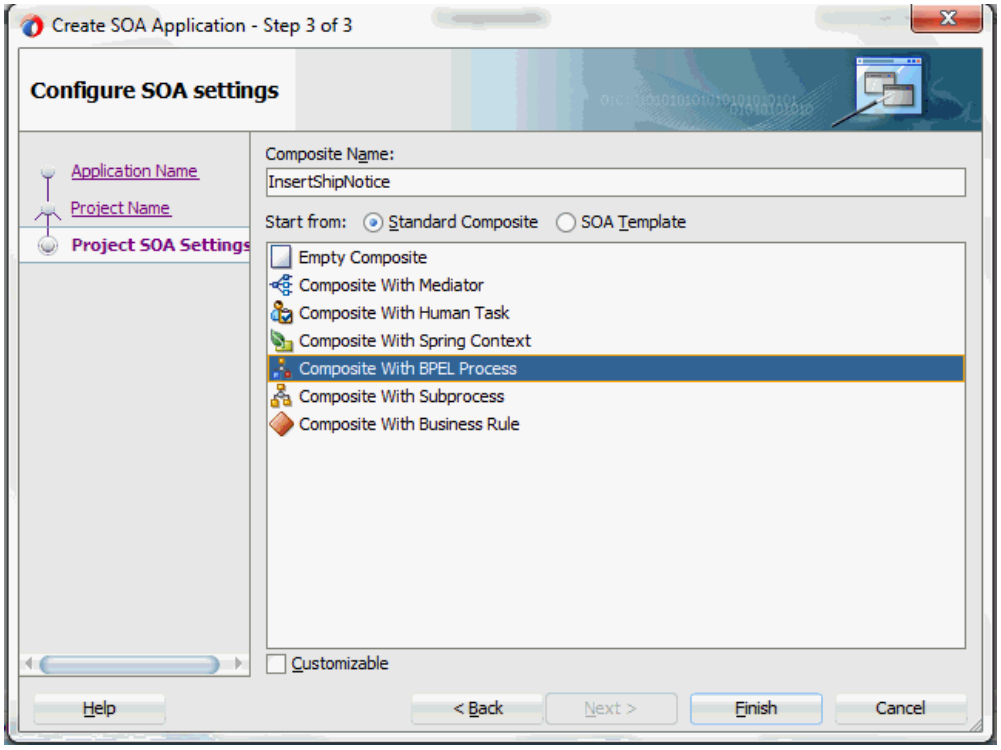
   Notice that **SOA Suite** is shown in the Project Features section.

The Create SOA Application - Name your project Page



6. Click **Next**. The Create SOA Application - Configure SOA settings page is displayed.

The Create SOA Application - Configure SOA settings Page

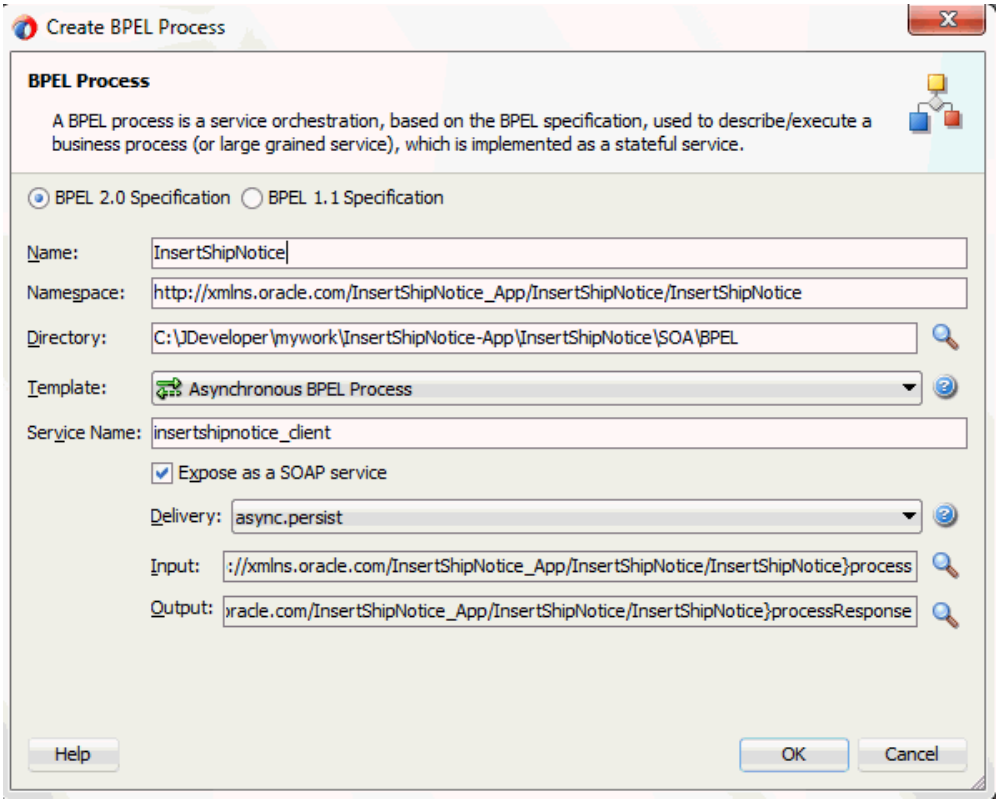


7. Leave the default **Standard Composite** selection unchanged for the Start from field.

   Select **Composite With BPEL Process** from the Standard Composite list, and then click **Finish**. You have created a new application, and a SOA project. This automatically creates a SOA Composite.

   The Create BPEL Process page is displayed.

The Create BPEL Process Page

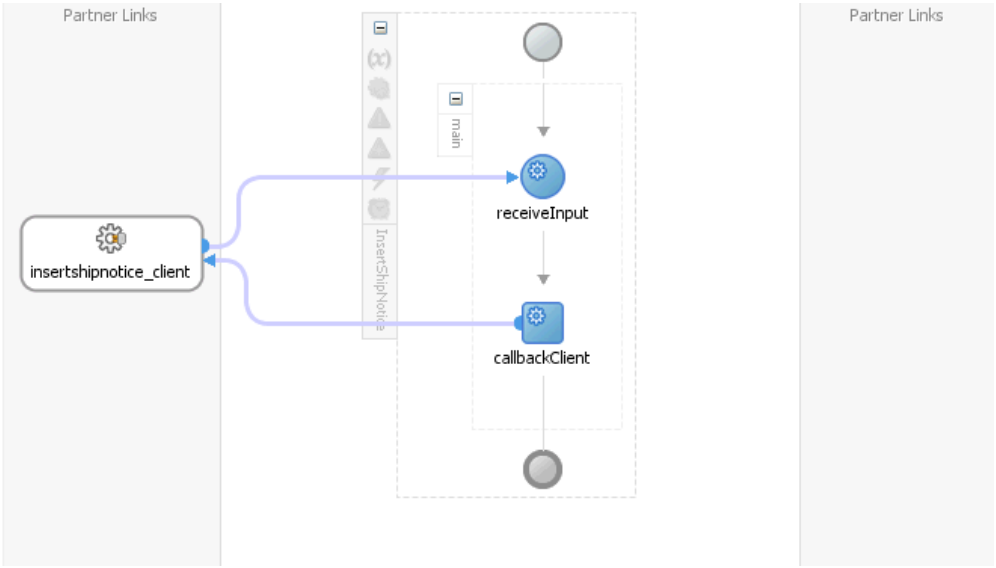


8. Leave the default **BPEL 2.0 Specification** selection unchanged. This creates a BPEL project that supports the BPEL2.0 specification.

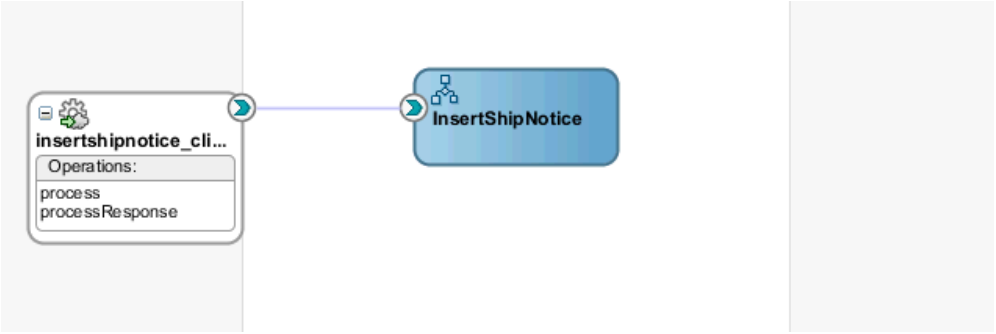
   Enter an appropriate name for the BPEL process in the **Name** field. For example, `InsertShipNotice`.

   Select **Asynchronous BPEL Process** in the **Template** field. Click **OK**.

   An asynchronous BPEL process is created with the Receive and Reply activities.



The required source files including bpel and wsdl, using the name you specified (for example, `InsertShipNotice.bpel` and `InsertShipNotice.wsdl`) and `InsertShipNotice` (composite.xml) are also generated.



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.

Based on the BPEL process scenario, you need to create a partner link to create a ship notice.

To add a partner link:

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

   Drag and drop **Oracle E-Business Suite** (formerly known as **Oracle Applications**) from the **BPEL Services** list into the right Partner Link swim lane of the process diagram. The Oracle E-Business Suite Reference page appears.

Enter a reference name in the **Name** field. For example, `ShipNotice`.

2. Click **Next**. The Service Connection page 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 **Create a New Database Connection** icon.

     How to define a new database connection, see Create a New Database Connection, page 5-14.

   - 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 Oracle E-Business Suite Adapter 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 Fusion Middleware User's Guide for Technology Adapters.*

4. Once you have completed creating a new connection for the service, you can add an interface by browsing through the available list in Oracle E-Business Suite.

   Click **Next**.

   If you are connecting to Oracle E-Business Suite Release 12, then the **IREP File not present** dialog appears indicating that Adapter 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 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.

  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.

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

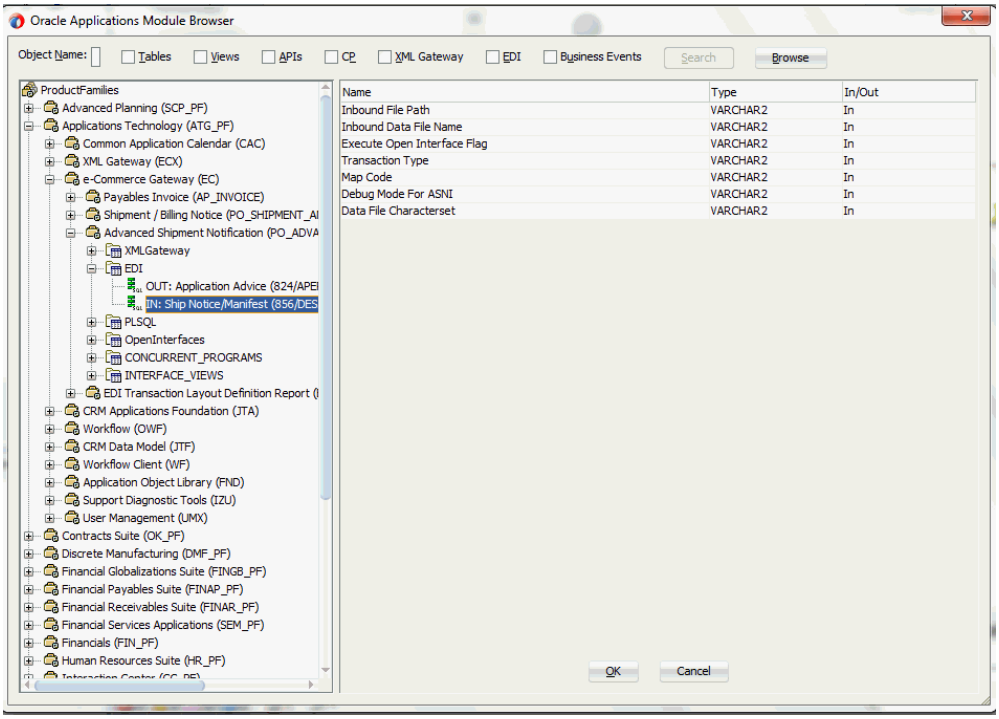
In this example, click **No**. Click **Next** in the Operation page to open the Oracle E-Business Suite Module Browser.

5. Once you have identified an existing database connection or completed a new connection for the service, you can select an e-Commerce Gateway interface through the Module Browser.

   Note: Oracle E-Business Suite Module Browser includes the various product families that are available in Oracle E-Business Suite. For example, Applications Technology or Order Management Suite are product families in Oracle E-Business Suite. The product families contain the individual products. For example, Applications Technology contains the e-Commerce Gateway product. The product contains the business entities associated with the product. For example, the e-Commerce Gateway product contains the Advanced Shipment Notification business entity.

   Business entities contain the various application modules that are exposed for integration. These modules are grouped according to the interface they provide. EDI programs can be found under the EDI category.

Specify an Interface from The Oracle E-Business Suite Module Browser



Select an inbound or outbound EDI program. You can select only one EDI program for each adapter service.

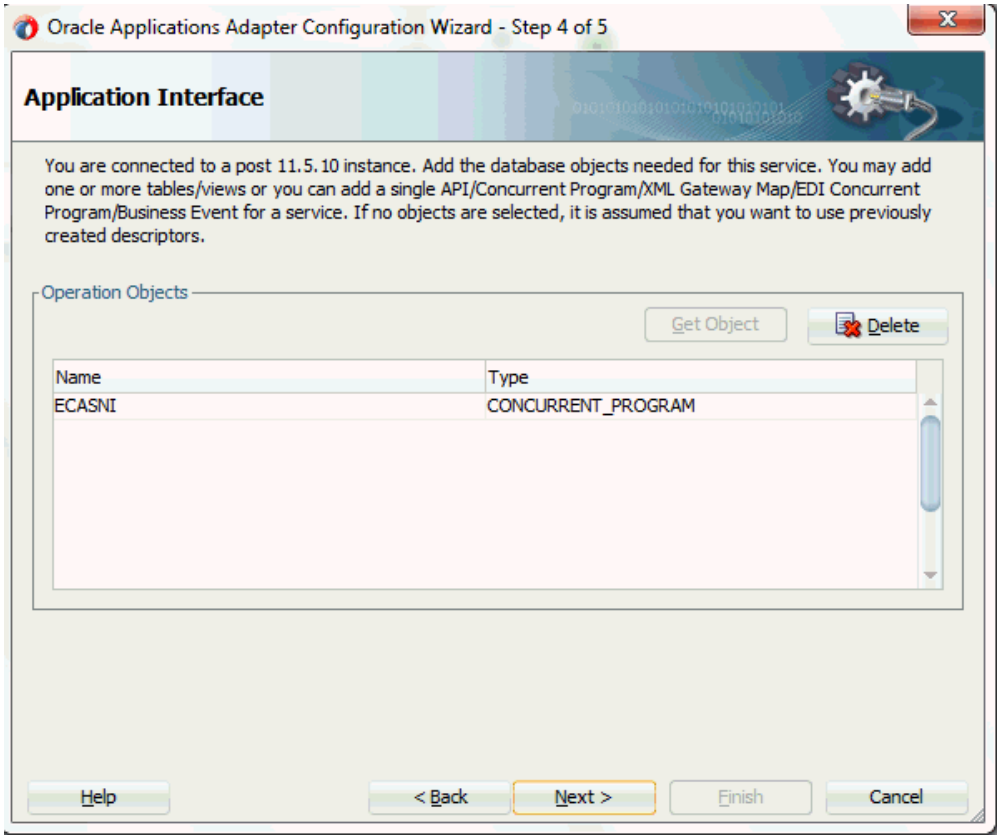
For example, navigate to *Product Families* > *Applications Technology* > *e-Commerce Gateway* > *Advanced Shipment Notification* > *EDI* to select `EDI > IN: Ship Notice/Manifest (856/DESADV) (ECASNI)` EDI concurrent program.

> **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**.

6. Click **OK**.

   The selected EDI concurrent program is added to Operation Objects.

Adapter Configuration Wizard - Application Interface Page



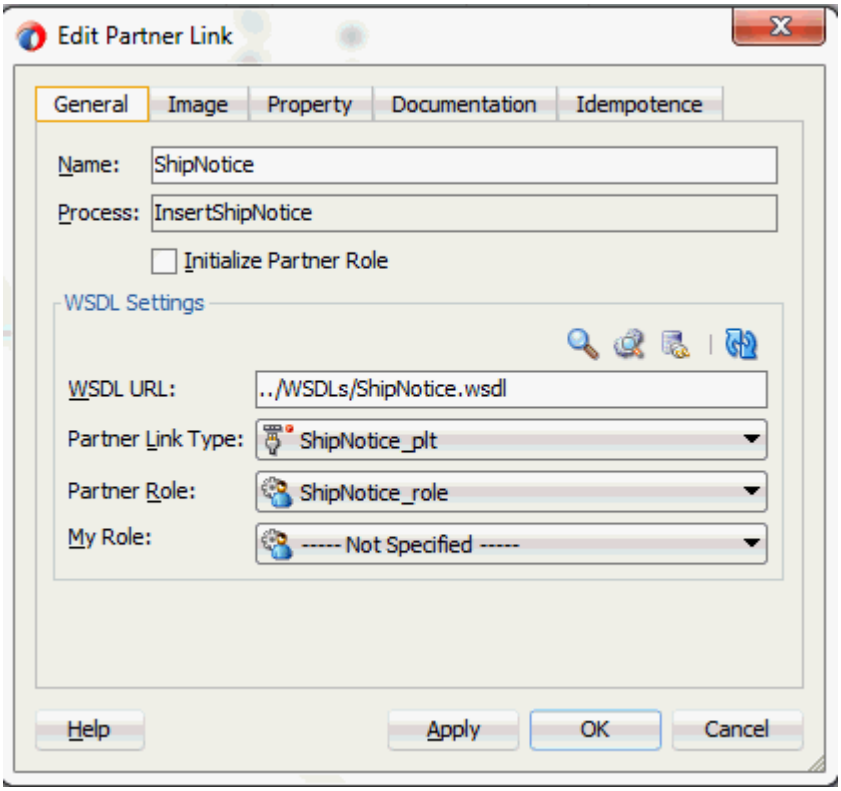
Click **Next**.

7. Click **Finish** to complete the process of configuring a partner link for an application service. The partner link is created with the required WSDL settings. The wizard also 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.

Double click the newly generated Partner Link (ShipNotice) icon to view the details.

Partner Link Configuration Details



Adding a Partner Link for File Adapter

Use this step to configure a BPEL process by synchronously reading 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, click **BPEL Services** in the Component palette.

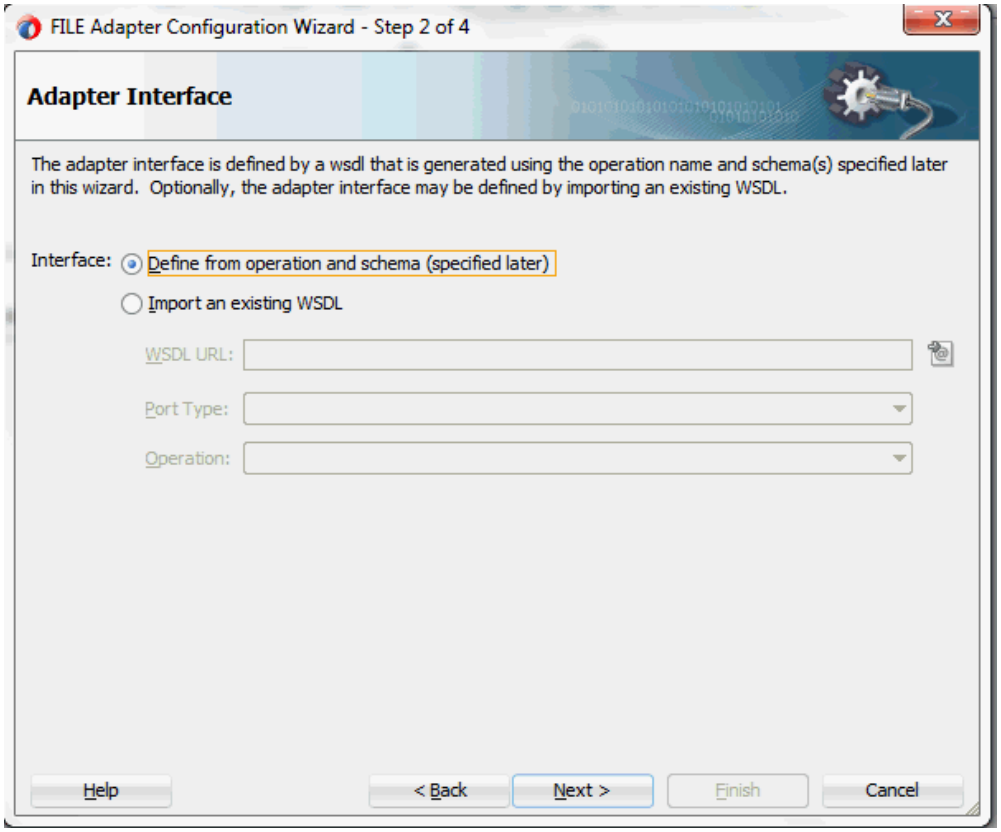
   Drag and drop **File Adapter** from the BPEL Services list into the right Partner Link swim lane of the process diagram before the partner link `InsertShipment`. The Adapter Configuration Wizard Welcome page appears.

   Click **Next**.

2. In the Service Name dialog, enter a name for the file adapter service, such as `getShippingDetails`.

3. Click **Next**. The Adapter Interface page appears.

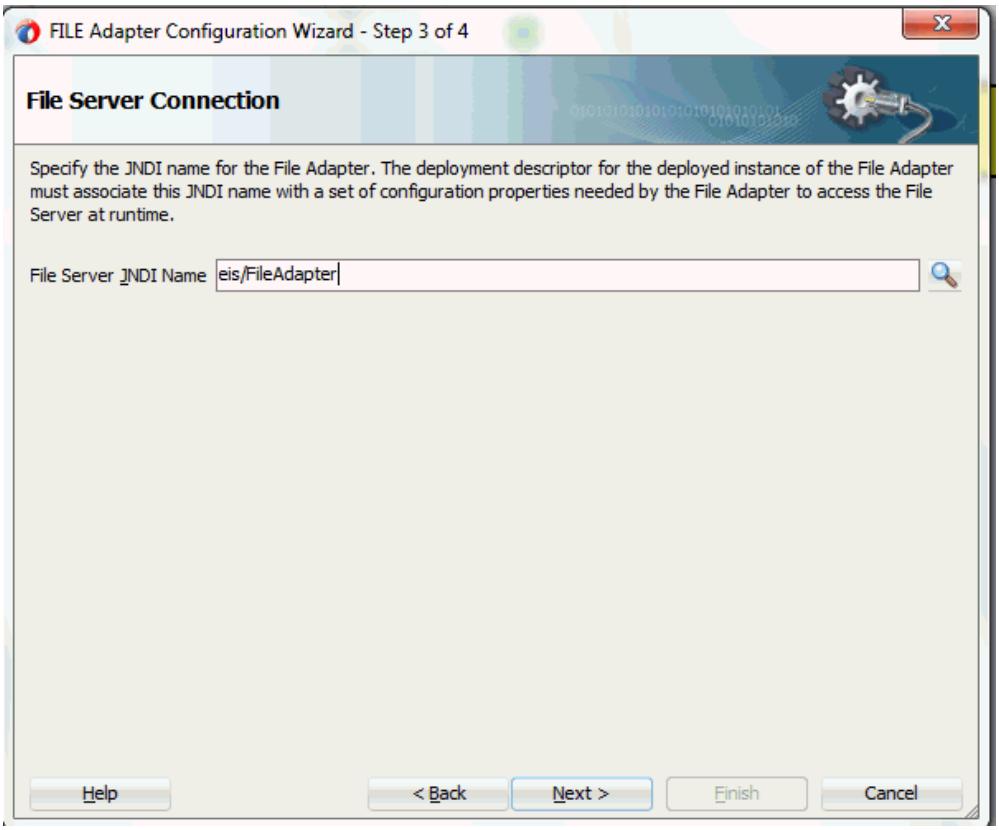
Specifying the Adapter Interface



Select the **Define from operation and schema (specified later)** radio button and click **Next**.

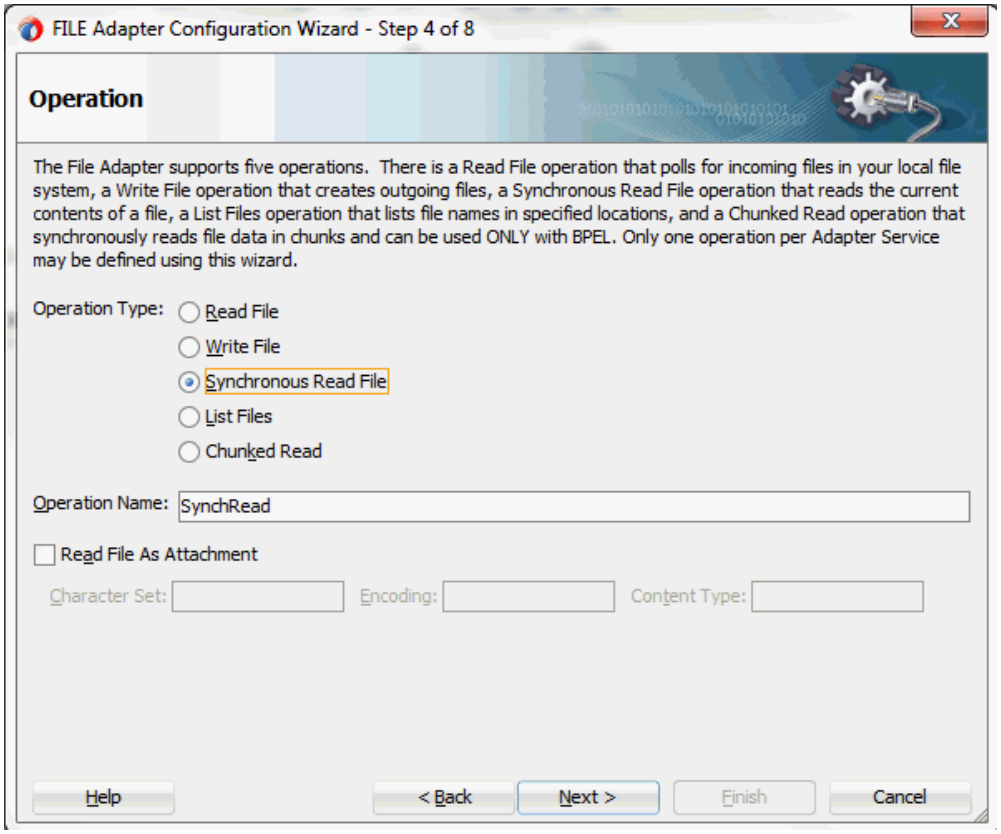
The File Server Connection page appears. Leave the default value unchanged for the File Server JNDI Name field.

Click **Next**



4. In the Operation dialog, specify the operation type. For example, select the **Synchronous Read File** radio button. This automatically populates the **Operation Name** field.

Specifying the Operation

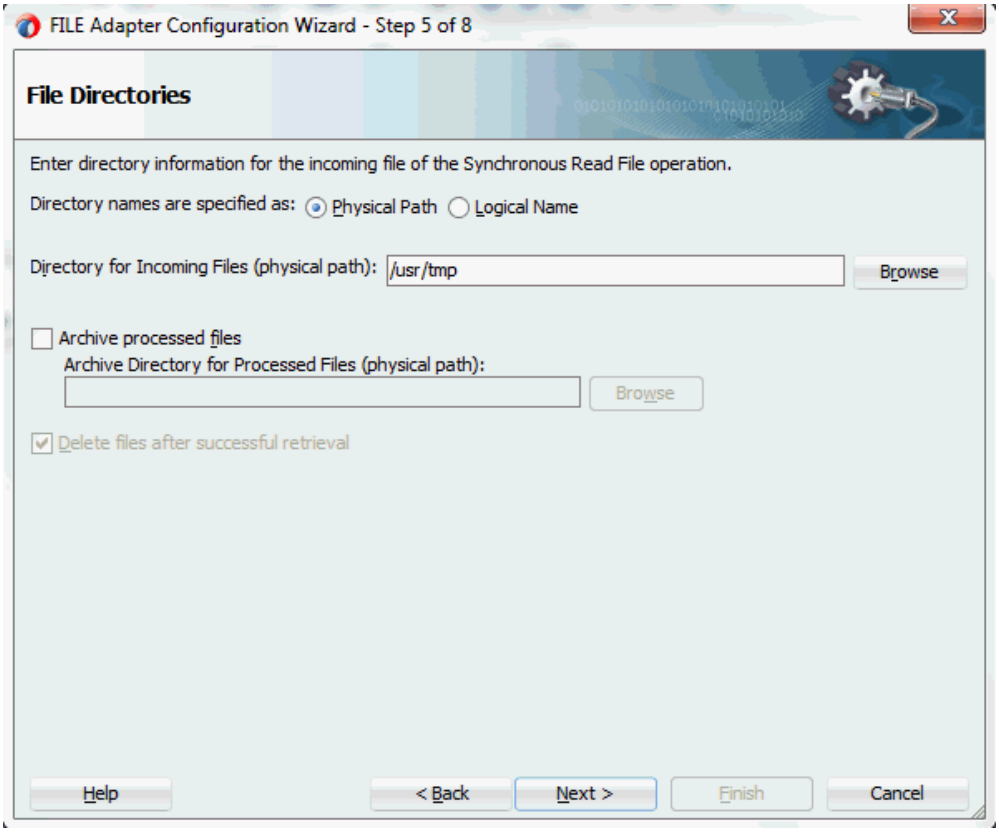


Click **Next** to access the File Directories page.

5. Select the **Physical Name** radio button and specify directory for incoming files, such as `/usr/tmp`.

   Ensure the **Delete files after successful retrieval** check box is not selected.

Configuring the Input File



Click **Next** to open the File Name page.

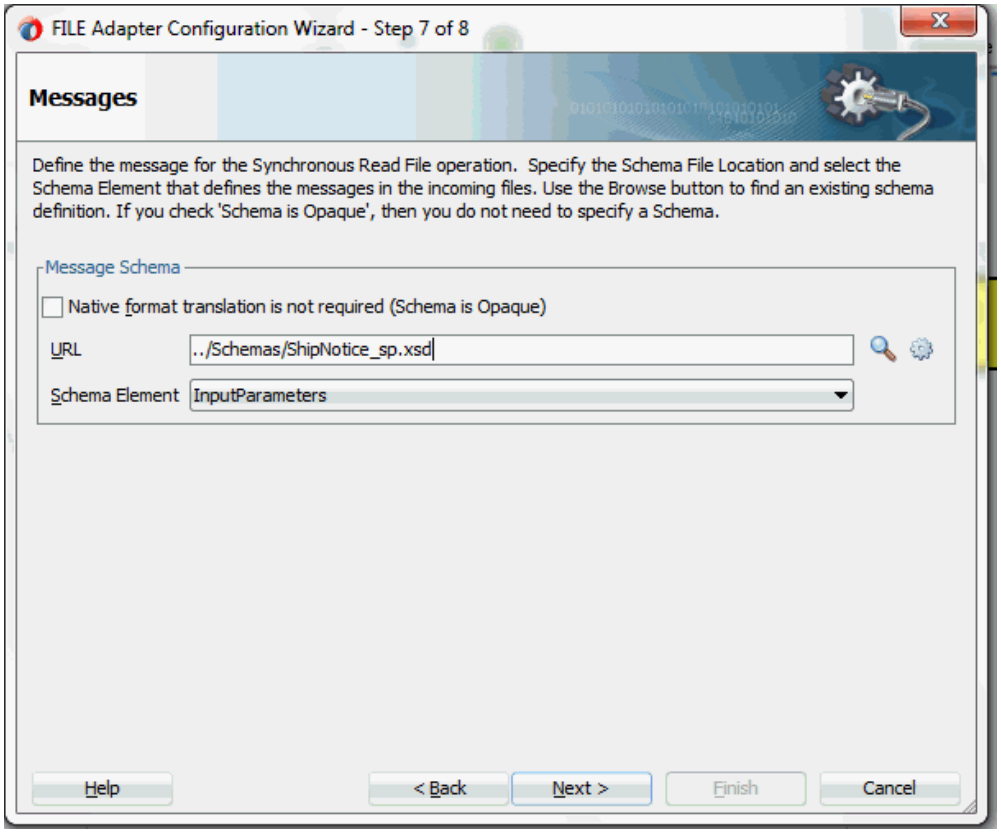
6. Enter the name of the file for the synchronous read file operation. For example, enter `order_data.xml`. Click **Next** to open the Messages dialog.

7. Select the 'browse for schema file' icon to open the Type Chooser.

   Click Type Explorer and select *Project Schema Files* > *ShipNotice_sp.xsd* > *InputParameters*.

   The selected schema information will be automatically populated in the URL and Schema Element fields.

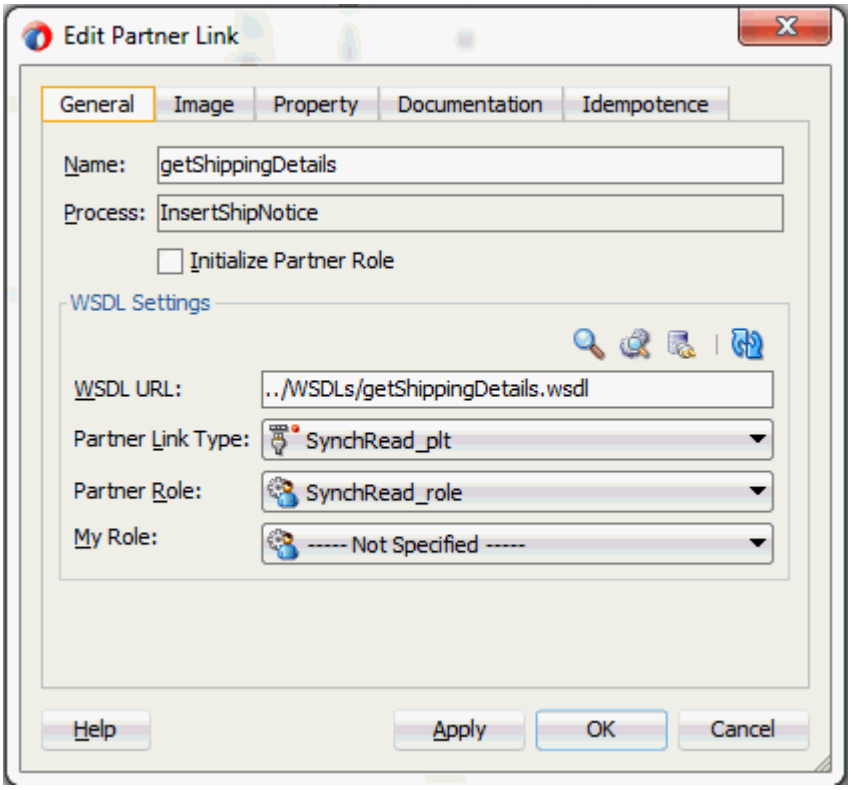
Specifying Message Schema



8. Click **Next** and then **Finish**. This completes the configuration and creates the partner link with the required WSDL settings for the File Adapter service. The wizard also generates the `getOrderXML.wsdl` WSDL file corresponding to the partner link.

   The `getShippingDetails` Partner Link appears in the BPEL process diagram. Double click the Partner Link service icon to open the Edit Partner Link dialog and view the details.

Partner Link Details



Configuring Invoke Activities

After adding and configuring the partner link, you can configure the following two **Invoke** process 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 `getShippingDetails` partner link in an XML file.

2. To create a shipment notice by invoking `ShipNotice` partner link.

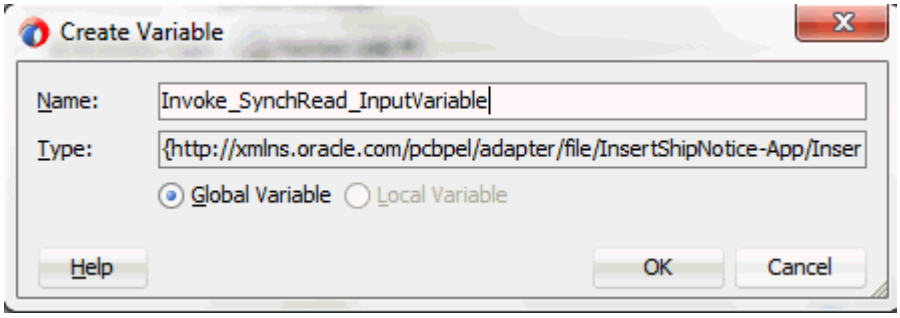
To add the first Invoke activity for a partner link to get shipping details:

1. In Oracle JDeveloper BPEL Designer, expand the **BPEL Constructs** from the Component Palette. Drag and drop the first **Invoke** activity into the center swim lane of the process diagram, between the **receiveInput** and **callbackClient** activities.

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

3. Enter a name for the Invoke activity. In the Input tab, ensure the **Input Variable** radio button is selected. Click the **Create** icon next to the **Input Variable** field to create a new variable. The Create Variable dialog box appears.

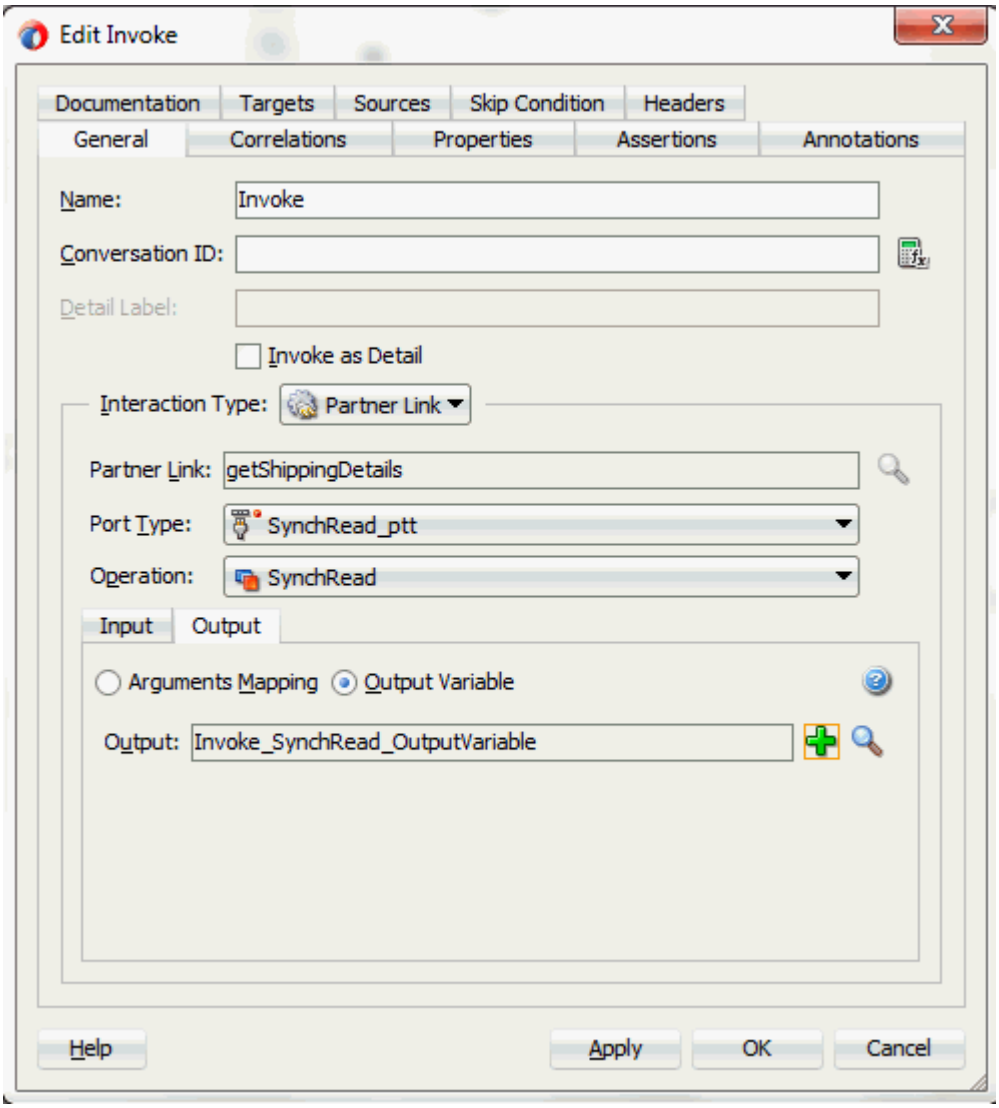
Create Input Variable



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

5. In the Output tab, ensure the **Output Variable** radio button is selected. 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**, then enter a name for the variable. You can also accept the default name. Click **OK**.



7. Click **Apply** and **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 create a shipment notice:

1. In Oracle JDeveloper BPEL Designer, expand the **BPEL Constructs** from the Component Palette. Drag and drop the third **Invoke** activity into the center swim lane of the process diagram, between the first **Invoke** and **callbackClient** activities.

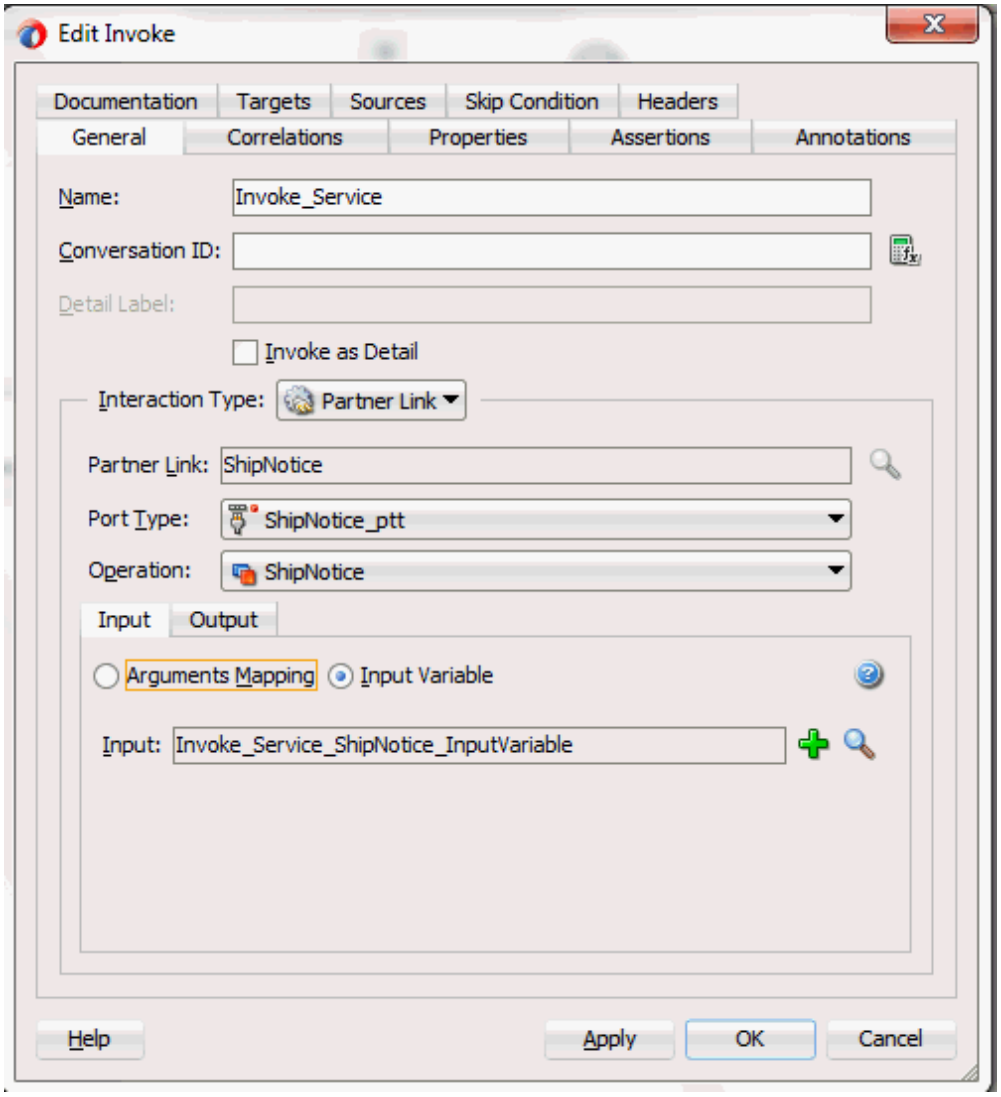
2. Link the Invoke activity to the `ShipNotice` service. The Edit Invoke dialog box appears.

   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 procedure.

   Click **Apply** and **OK**.



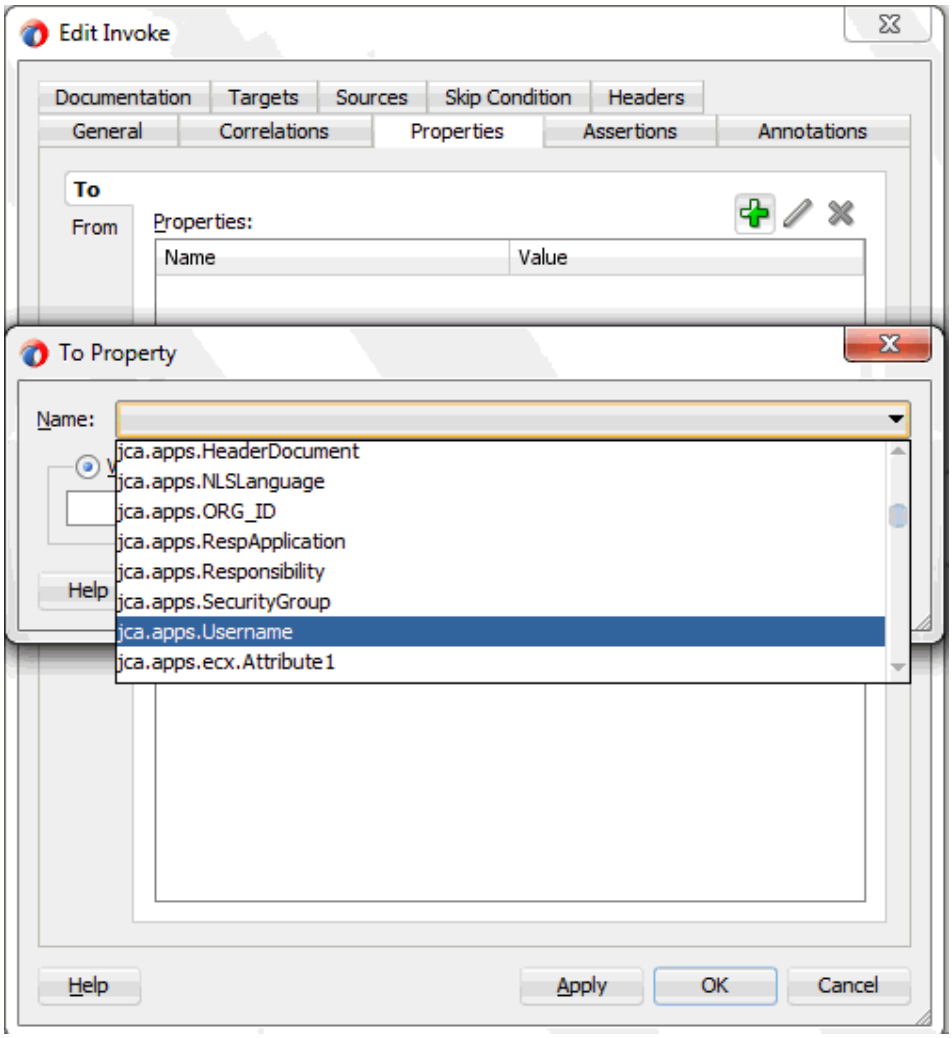
4. **Setting Header Properties for Applications Context**

   Use the following steps to set the header message properties required to pass applications context required to complete the BPEL process:

   1. Select the Properties tab to set the Header message properties required to pass applications context required to complete the BPEL process.

   2. Click the **Create** icon. The To Property dialog appears.

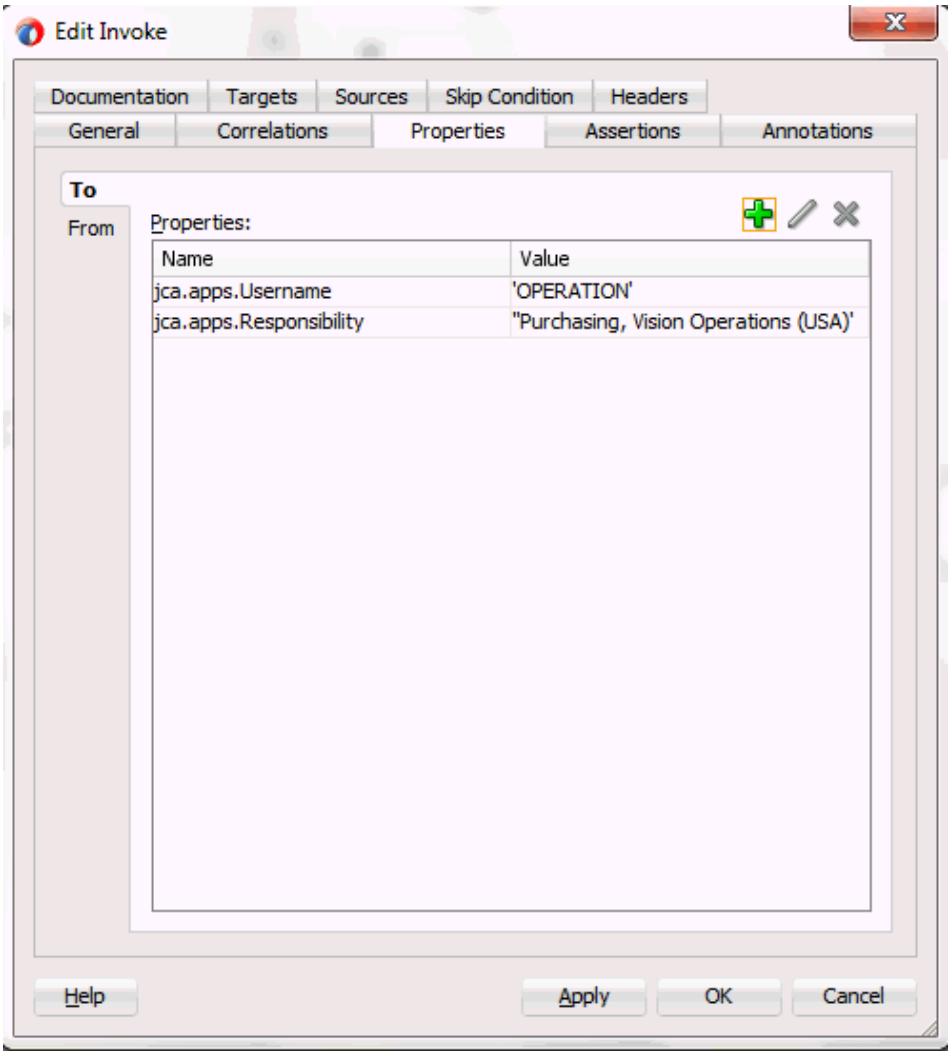
3. Select the `jca.apps.Username` property from the Name drop-down list.



4. Select the **Expression** radio button and enter `'OPERATIONS'` as the property value.

   Click **OK**.

5. Repeat Step 2 to Step 4 to assign `'Purchasing, Vision Operations (USA)'` for `jca.apps.Responsibility`.



5. Click **Apply** and then **OK** to complete the Invoke activity.

Configuring an Assign Activity

Use the Assign activity to pass the output of `getShippingDetails` service as an input to the `InsertShipment` service.

To add an Assign activity:

1. In Oracle JDeveloper BPEL Designer, expand the **BPEL Constructs** from the Component Palette. Drag and drop the **Assign** activity from the Basic Activities section into the center swim lane of the process diagram between the two **Invoke** activities that you just created earlier.

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

Click the General tab to enter a name for the Assign activity. For example, `setShipInfo`.

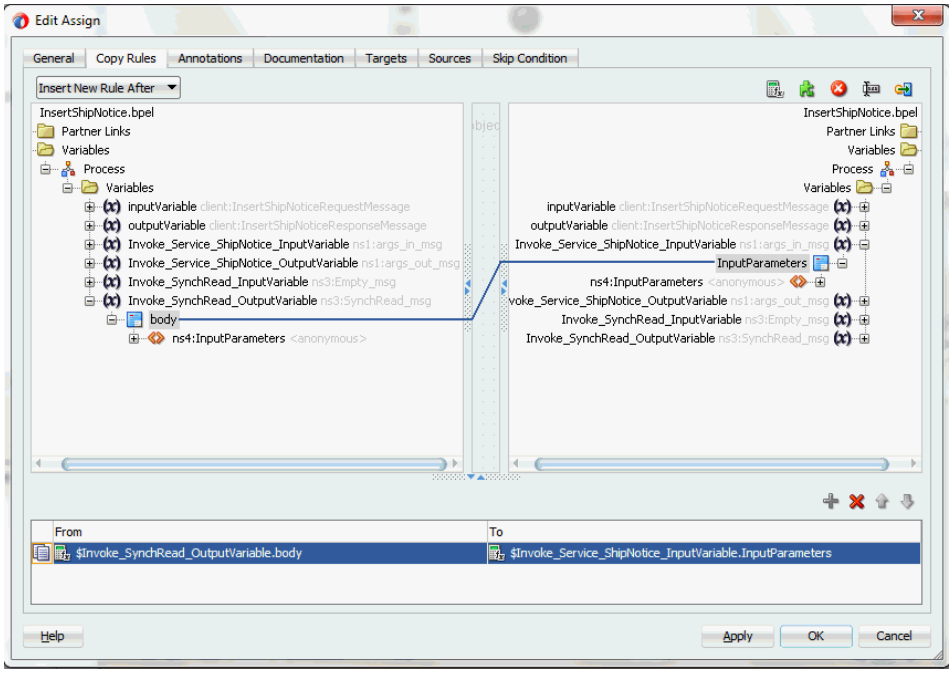
3. Select the Copy Rules tab and expand the source and target trees:

   - In the From navigation tree, navigate to **Variable > Process > Variables > Invoke_SynchRead_OutputVariable > body** and select **ns3:InputParameters**.

   - In the To navigation tree, navigate to **Variable > Process > Variables > Invoke_Service_ShipNotice_InputVariable > Input Parameters** and select **ns3:InputParameters**.

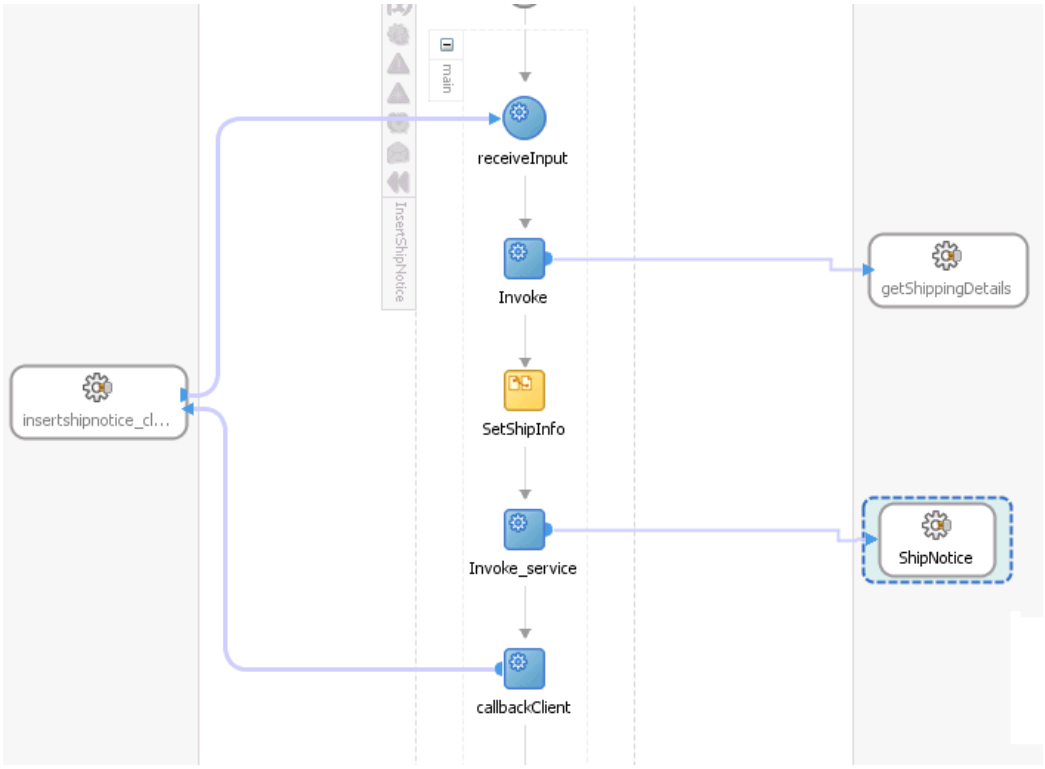
     Drag the source node (ns3:InputParameters) to connect to the target node (ns3:InputParameters) that you just identified. This creates a line that connects the source and target nodes. The copy rule is displayed in the From and To sections at the bottom of the Edit Assign dialog box.

Assigning Parameters

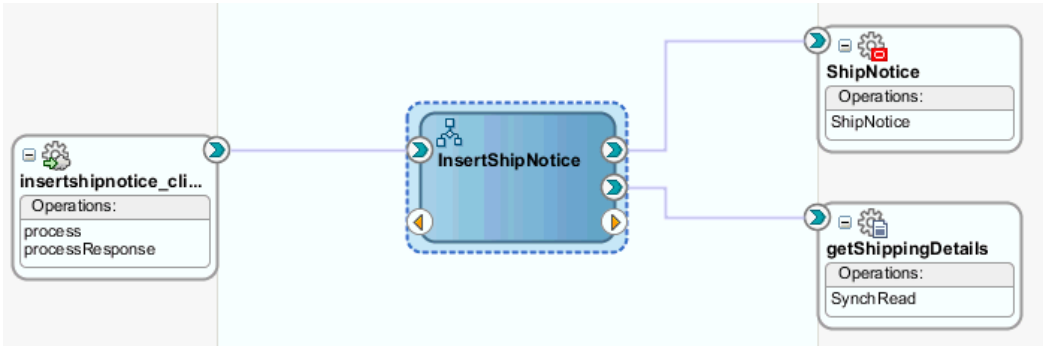


4. Click **Apply** and **OK** to complete the configuration of the Assign activity.

Completed Concurrent Program BPEL Process Project



Click the `composite.xml` to display the Oracle JDeveloper composite diagram:



Run-Time Tasks for e-Commerce Gateway

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

1. Deploy the SOA Composite application with BPEL process, page 10-28
2. Test the deployed SOA Composite application with BPEL process, page 10-31

3. Verify records in Oracle E-Business Suite, page 10-33

Deploying the SOA Composite Application with BPEL Process

To invoke the service (InsertShipment) from the BPEL client contained in the SOA composite, the SOA composite needs to be deployed to the Oracle WebLogic managed server. This can be achieved using Oracle JDeveloper. Once the composite is deployed, it can be tested from the Oracle Enterprise Manager Fusion Middleware Control Console.

Prerequisites

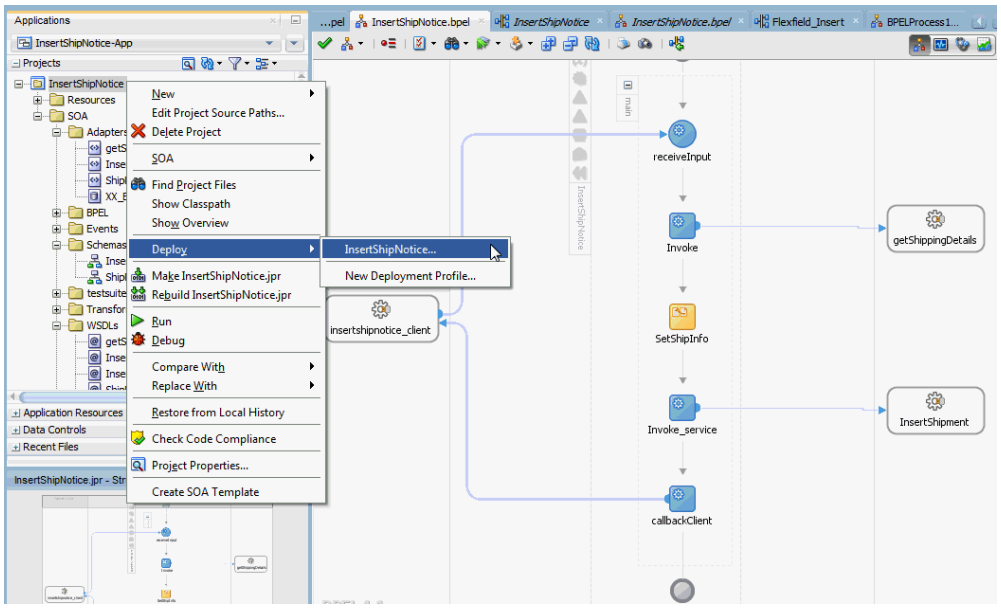
Before deploying the SOA composite with BPEL process using Oracle JDeveloper, you must have established the connectivity between the design-time environment and the run-time server. For more information, see Configuring the Data Source in Oracle WebLogic Server, page 11-1 and Creating an Application Server Connection, page 11-10.

> **Note:** If a local instance of the WebLogic Server is used, start the WebLogic Server by selecting **Run > Start Server Instance** from Oracle JDeveloper. Once the WebLogic Admin Server "DefaultServer" instance is successfully started, the <Server started in Running mode> and DefaultServer started message in the Running:DefaultServer and Messages logs should appear.

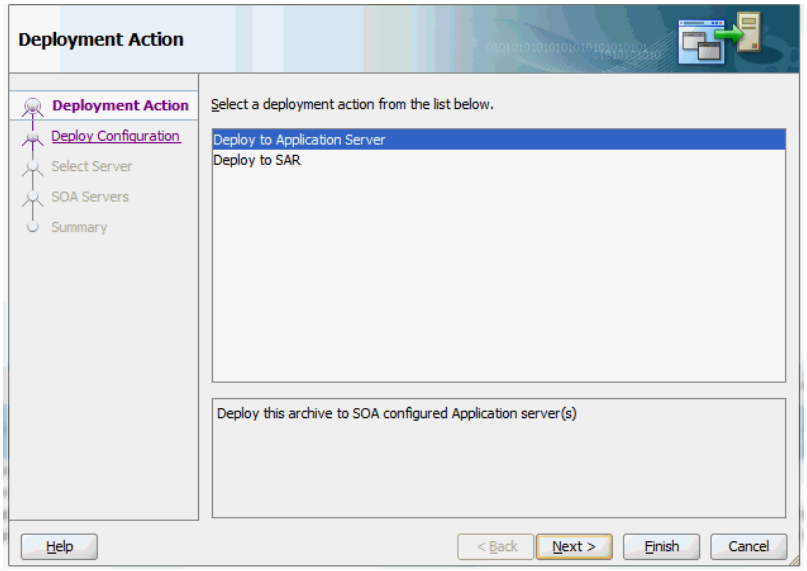
To deploy the SOA Composite application with BPEL process:

1. Select the SOA Composite project in the Applications Navigator.
2. Right-click the project name, and then select **Deploy > [project name] > [serverConnection]** from the menu that appears.

   For example, you can select **Deploy > InsertShipNotice > soa-server1** to deploy the process if you have the connection set up appropriately.



Note: If this is the first time to set up server connection, then the Deployment Action window appears. Select 'Deploy to Application Server' and click **Next**.

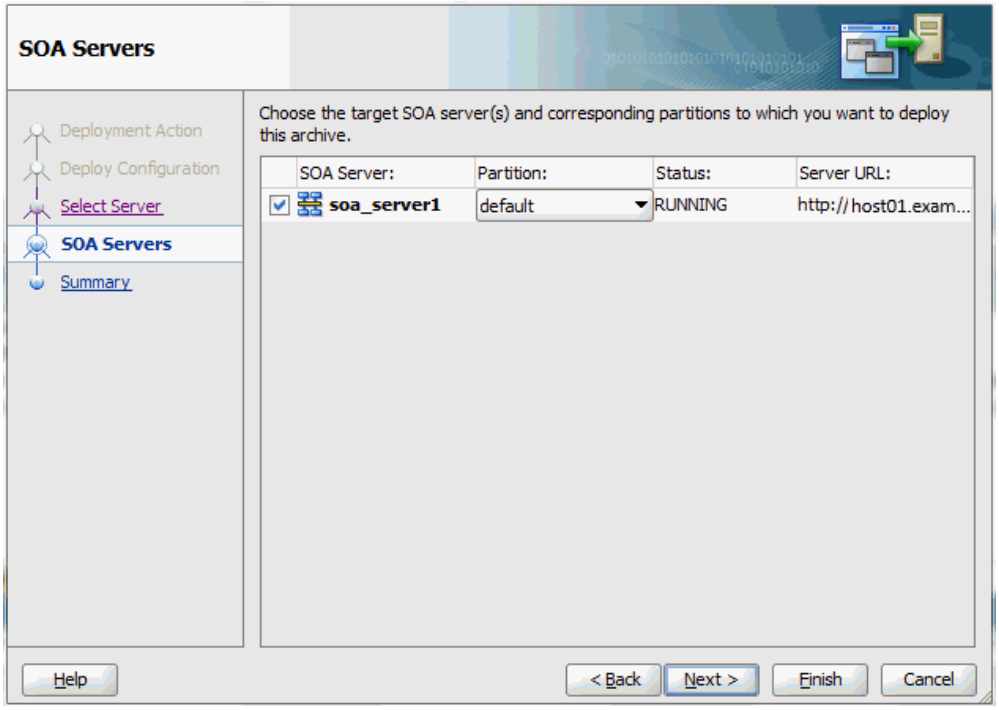


In the Deploy Configuration window, ensure the following information is selected before clicking **Next** to add a new application server:

- New Revision ID: 1.0
- Mark composite revision as default: Select this check box.
- Overwrite any existing composites with the same revision ID: Select this check box.

The steps to create a new Oracle WebLogic Server connection from JDeveloper are covered in Creating an Application Server Connection, page 11-10.

3. In the Select Server page, select 'soa-server1' that you have established the server connection earlier. Click **Next**.
4. In the SOA Servers page, accept the default target SOA Server ('soa-server1') selection.



Click **Next** and **Finish**.

If you are deploying the composite for the first time from your Oracle JDeveloper session, the Authorization Request window appears. Enter username and password information specified during Oracle SOA Suite installation. Click **OK**.

5. Deployment processing starts. Monitor deployment progress and check for successful compilation in the SOA - Log window.

   Verify that the deployment is successful in the Deployment - Log window.

Testing the Deployed SOA Composite Application with BPEL Process

Once the BPEL process contained in a SOA Composite application is deployed, you can manage and monitor the process from the Oracle Enterprise Manager Fusion Middleware Control Console. You can also test the integration by manually initiating the process.

To test the deployed SOA Composite application with BPEL process:

1. Navigate to Oracle Enterprise Manager Fusion Middleware Control Console (`http://<servername>:<portnumber>/em`). The composite you deployed is displayed in the Applications Navigation tree.

2. Enter username (such as `weblogic`) and password and click **Login** to log in to a farm.

   You may need to select an appropriate target instance farm if there are multiple target Oracle Enterprise Manager Fusion Middleware Control Console farms.

3. From the Farm base domain, expand the **SOA > soa-infra > soa-infra (soa_server1) > default** to navigate through the SOA Infrastructure home page and menu to access your deployed SOA Composite applications running in the SOA Infrastructure for that managed server.

   > **Note:** The Farm menu always displays at the top of the navigator. As you expand the SOA folder in the navigator and click the links displayed beneath it, the SOA Infrastructure menu becomes available at the top of the page.

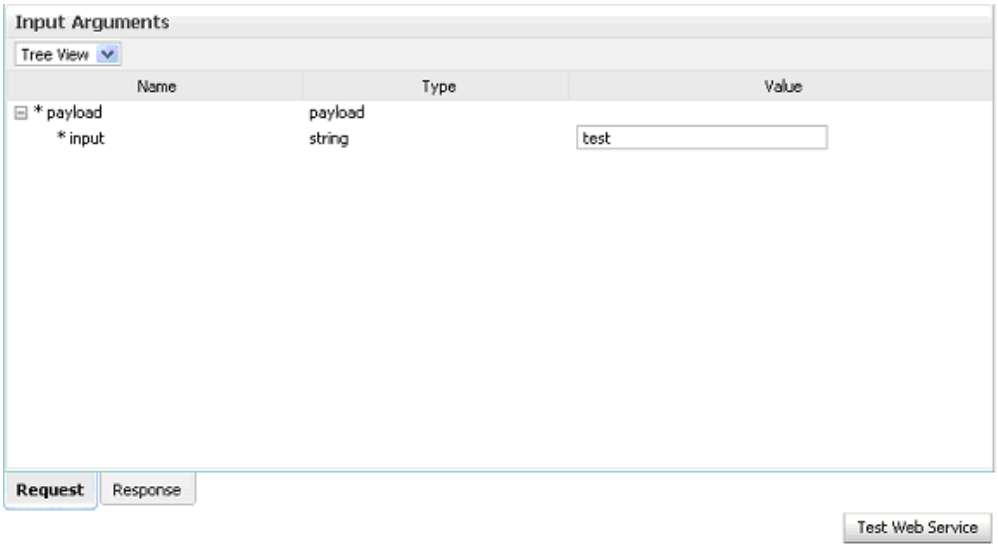
   Click the SOA Composite application that you want to initiate (such as 'InsertShipment') from the SOA Infrastructure.

   Click **Test** at the top of the page.

4. The Test Web Service page for initiating an instance appears. You can specify the XML payload data to use in the Input Arguments section.

   Enter the input string required by the process and click **Test Web Service** to initiate the process.

Testing Web Service



The test results appear in the Response tab upon completion.

5. Click **Launch Flow Trace** in the Test Web Service page. The Flow Trace page is displayed.

   In the Trace section, you should find the sequence of the message flow for the service binding component (`insertshipnotice_client_ep`), BPEL component (`ShipNotice`), and reference binding components (`getShippingDetails` and `ShipNotice`). All involved components have successfully received and processed messages.

   If any error occurred during the test, you should find it in the Faults section.

6. Click your BPEL service component instance link (such as `ShipNotice`) to display the Instances page where you can view execution details for the BPEL activities in the Audit Trail tab.

   Click the Flow tab to check the BPEL process flow diagram. Click an activity of the process diagram to view the activity details and flow of the payload through the process.

Verifying Records in Oracle E-Business Suite

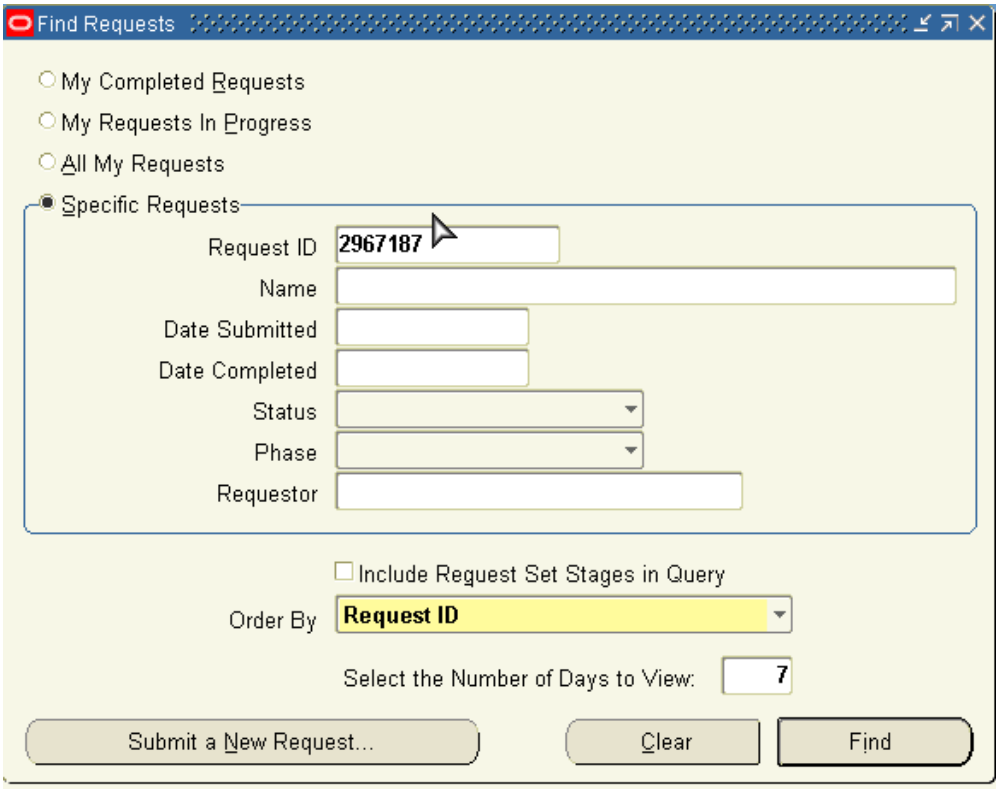
To verify retcords in Oracle E-Business Suite:

1. Log in to Oracle E-Business Suite as the System Administrator.

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.**

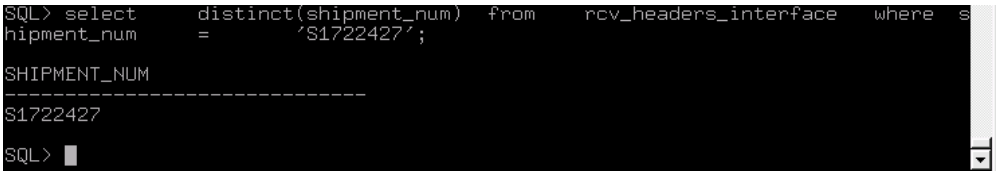
Find Requests Dialog Box



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 E-Business Suite to search for the relevant records that have been inserted.

Querying Oracle E-Business Suite for a Record



11

Runtime Configuration for Oracle E-Business Suite Adapter

Overview

Oracle E-Business Suite Adapter is deployed as JCA 1.5 resource adapter within the Oracle WebLogic Server container.

Although Oracle E-Business Suite Adapter is physically deployed as JCA 1.5 resource adapter, the logical deployment of the Adapter involves creating the connection entries for the JCA 1.5 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.

For the logical deployment changes to take effect, the Oracle WebLogic Server container process must be restarted.

The following topics are included in this section:

- Configuring the Data Source in Oracle WebLogic Server, page 11-1
- Creating Connection Factory for Oracle E-Business Suite Adapter, page 11-7
- Creating the Application Server Connection, page 11-10

Configuring the Data Source in Oracle WebLogic Server

Oracle WebLogic Server consists of a JCA container for hosting JCA resource adapters. JCA defines standard Java interfaces for simplifying the integration of a J2EE server with various back-end applications. All client applications run within the Oracle WebLogic Server environment.

The JCA resource adapter is a WebLogic Server component contained in a Resource Adapter Archive (RAR) file within the applications/directory. An RAR file contains a correctly formatted deployment descriptor. In addition, it contains declarative

information about the contract between the Oracle WebLogic Server and resource adapter.

The `weblogic-ra.xml` file is the deployment descriptor for a WebLogic resource adapter. It contains deployment configurations for deploying resource adapters to Oracle WebLogic Server, which includes the back-end application connection information as specified in the deployment descriptor of the resource adapter, Java Naming and Directory Interface (JNDI) name to be used, connection pooling parameters, security identities, Work Manager properties, logging, and resource principal mapping mechanism and configurations.

> **Note:** A WebLogic resource adapter also uses `ra.xml` descriptor that describes the resource adapter-related attributes type and its deployment properties using the standard XML schema specified by the JCA 1.5 Specification.

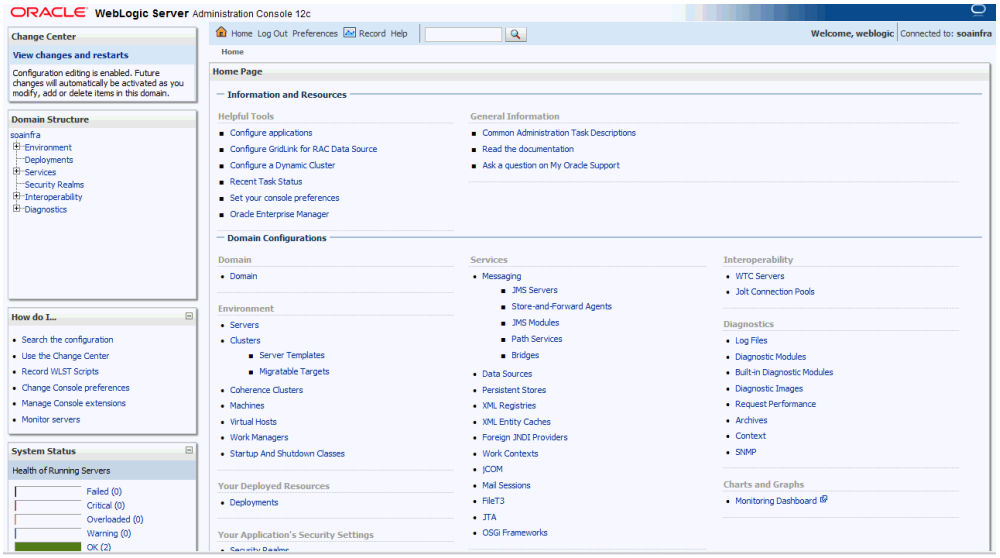
After successfully installing Oracle WebLogic Server, use the following steps to configure the data sources in the Oracle WebLogic Server Administration Console:

> **Note:** How to install Oracle WebLogic Server and create connection factory, see *Oracle WebLogic Server Installation Guide* for details.

1. Navigate to `http://servername:portnumber/console`.
2. Enter the server administrator's username and password to log on to the Oracle WebLogic Server Administration Console.

   The Home page of the Oracle WebLogic Server Administration Console appears.

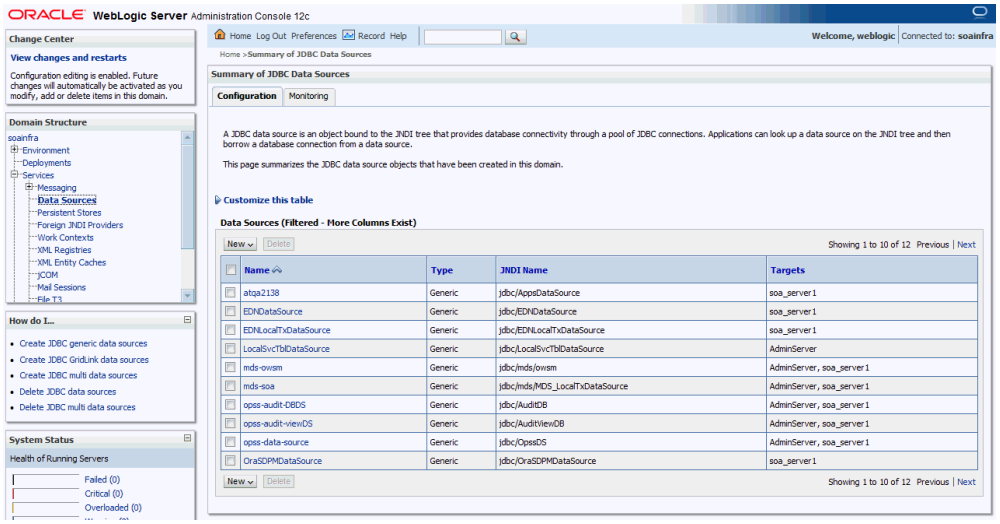
Oracle WebLogic Server Administration Console Home Page



3. Under the Domain Structure section, select **Services** and then **DataSources**.

   The Summary of JDBC Data Sources page is displayed.

The Summary of JDBC Data Sources Page

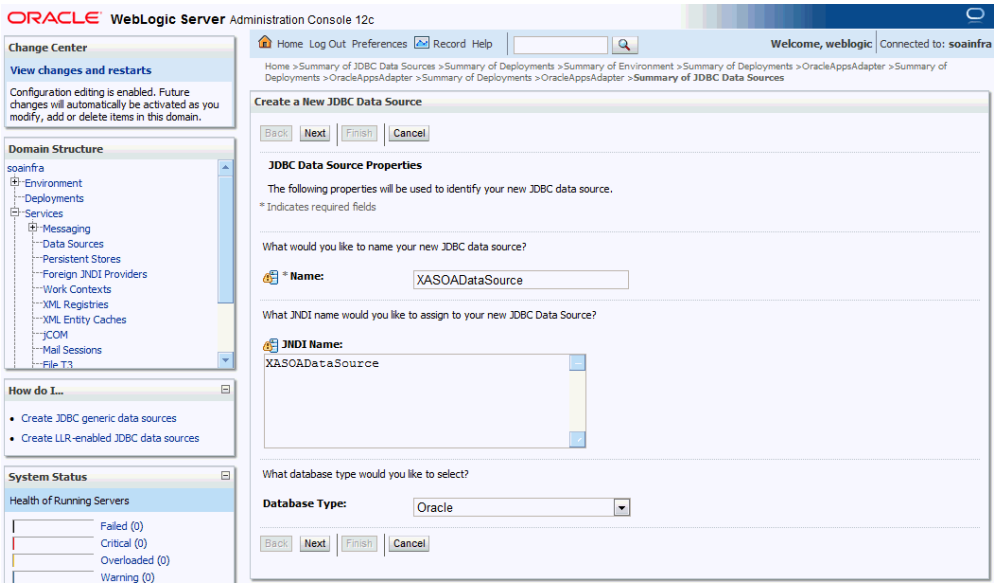


4. In the Summary of JDBC Data Sources section, click **New** and select the option **Generic Data Source** to create a new data source. The Create a New JDBC Data Source page is displayed.

5. Enter the values for the properties to be used to identify your new JDBC data

source.

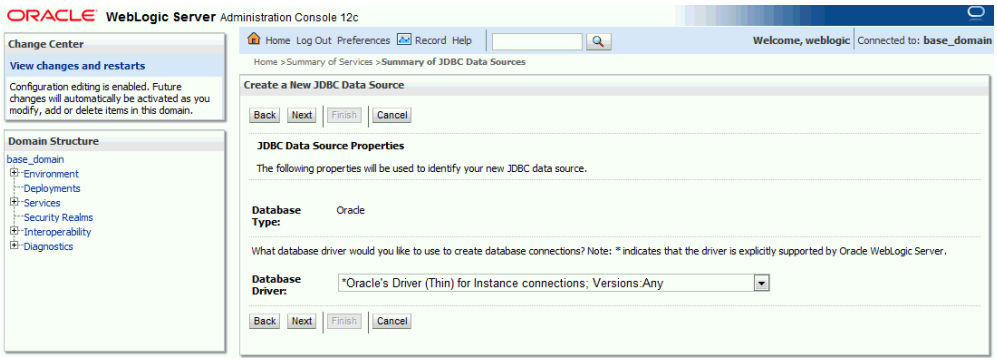
The Create a New JDBC Data Source - JDBC Data Source Properties Page



- Name: Enter a desired new JDBC data source name.
- JNDI Name: Enter a JNDI name that you like to assign to your new JDBC Data Source.
- Database Type: Select `Oracle` as database type from the drop-down list.

Click **Next**.

6. The selected value for Database Type field is displayed. In the Database Driver field, select `Oracle's Driver` (Oracle's Driver (Thin or Thin XA) for instance connections) from the drop-down list.



Note: A driver name with '*' indicates that the driver is explicitly supported by Oracle WebLogic Server.

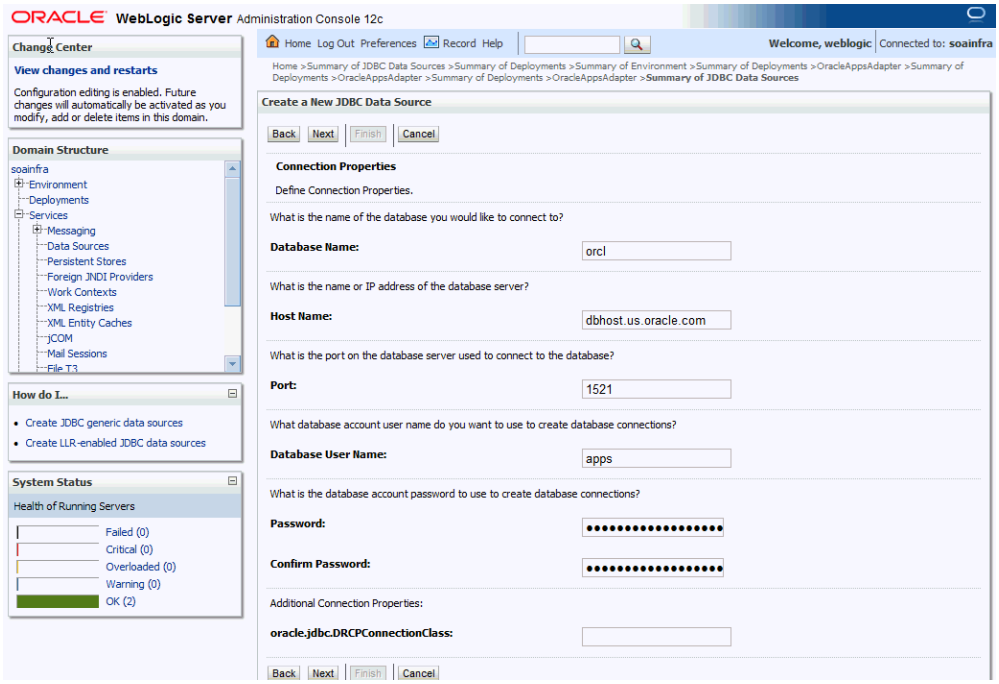
The required Oracle E-Business Suite database version for Oracle E-Business Suite Adapter 12*c* (12.2.1) is listed in the following table:

Oracle Database Release	Oracle Database 11*g* Release 1	Oracle Database 11*g* Release 2	Oracle Database 12*c* Release 1
Minimum Required Database Version	11.1.0.7 or higher	11.2.0.3 or higher	12.1.0.1 or higher

7. Click **Next**. The Create a New JDBC Data Source - Transaction Options page is displayed.

8. Click **Next**. The Create a New JDBC Data Source - Connection Properties page is displayed.

The Create a New JDBC Data Source - Connection Properties page



9. Enter the connection properties in the Connection Properties page.
   - Database Name: Enter a database name that you would like to connect.
   - Host Name: Enter a host name or IP address of the database server.
   - Port: Enter a database port on the database server used to connect to the database.
   - Database User Name: Enter the database account username that you want to use to create the database connections.
   - Password: Enter the database account password that you want to use to create the database connections.
   - Confirm Password: Enter the same password for confirmation.

10. Click **Next**. The Create a New JDBC Data Source - Test Database Connection page is displayed.

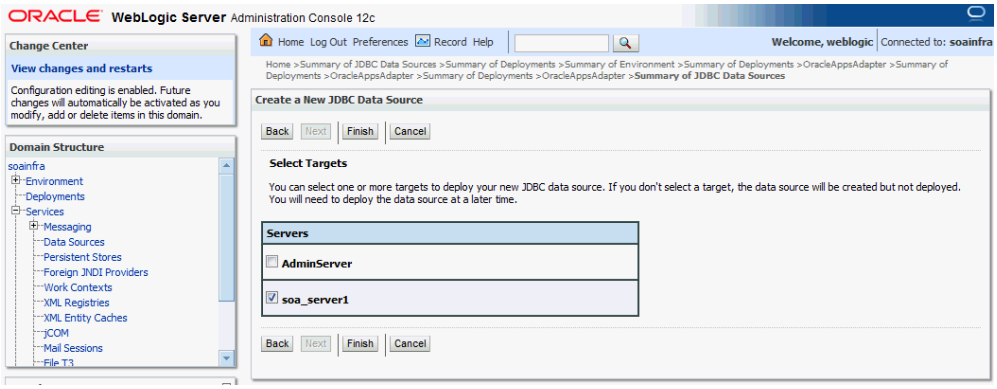
11. Click **Test Configuration** to test the database availability and the connection properties you provided.

    A 'Connection test succeeded' message appears at the top of the Create a New JDBC

Data Source - Test Database Connection page.

12. Click **Next**. The Create a New JDBC Data Source - Select Targets page is displayed.

The Create a New JDBC Data Source - Select Targets Page



13. Select the 'soa_server1' check box, and then click **Finish**.

    The Summary of JDBC Data Sources page is displayed. This page summarizes the JDBC data source objects that have been created in this domain. The Data Source that you created appears in this list.

Creating Connection Factory for Oracle E-Business Suite Adapter

The deployment descriptor of the Oracle E-Business Suite Adapter must associate with the Java Naming and Directory Interface (JNDI) name with configuration properties required by the adapter to access the database.

Use the following steps to configure the JNDI and additional settings for Oracle E-Business Suite Adapter through the Oracle WebLogic Server Administration Console:

1. Navigate to `http://servername:portnumber/console`.

2. Enter the server administrator's username and password to log on to the Oracle WebLogic Server Administration Console.

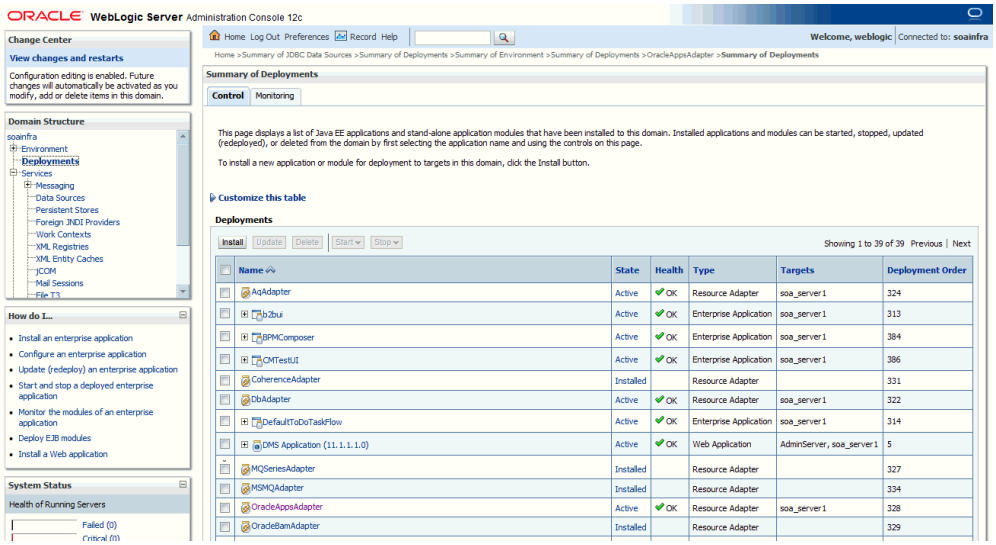
   The Home page of the Oracle WebLogic Server Administration Console appears.

3. Under the Domain Structure section, select **Deployment**.

   The Summary of Deployments page appears where you can find a list of Java EE applications and standalone application modules that have been installed to this domain.

4. Locate an appropriate link for Oracle E-Business Suite Adapter from the summary of deployment table first and then click the link.

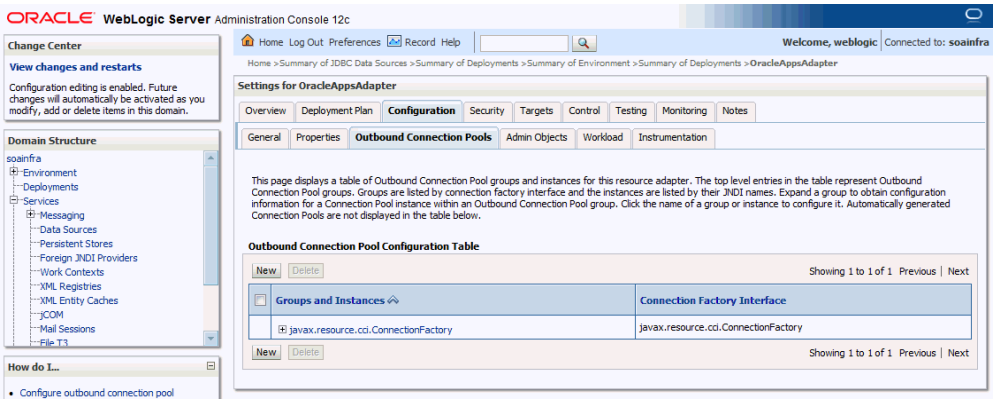
Summary of Deployments



The Overview Settings for the selected application name appears.

5. Click the **Configuration** tab and the **Outbound Connection Pools** subtab to display the Outbound Connection Pool groups and instances for this selected resource adapter.

Outbound Connection Pool Configuration Table



You can click the `javax.resource.cci.ConnectionFactory` link to expand this group and obtain configuration information for a Connection Pool instance.

6. Click **New** and select the `javax.resource.cci.ConnectionFactory` radio button to create an instance. Click **Next** to create a new outbound connection.

7. Enter a JNDI name that you want to use to obtain the new connection instance. For example, enter `eis/Apps/Ebiz`.

The Outbound Connection instance represents a connection pool. The JNDI name can be used to obtain the pool at run time.

Click **Finish**.

8. This opens the Save Deployment Plan Assistant page where you can find the new deployment plan information including plan path with an XML extension (`.xml`), recently used paths, and current location.

   > **Note:** The plan path must have an XML extension (`.xml`). It is recommended that this file be called `Plan.xml`, and that each plan be located in a unique directory. If the plan file exists it will be overwritten. Other files in the plan directory may be overwritten as well.

9. Click **OK** to save the configuration change for a deployment plan.

   You may find some unexpected exception messages populated while processing the request. Use the following steps to resolve the issue:

   1. Click the Adapter link that you selected earlier on the breadcrumb navigation path.

   2. Click the **Configuration** tab and the **Outbound Connection Pools** subtab to display the Outbound Connection Pool groups and instances.

      Click the `javax.resource.cci.ConnectionFactory` expand node to display the newly configured JNDI (such as `eis/Apps/Ebiz`) for the outbound instance in the configuration table.

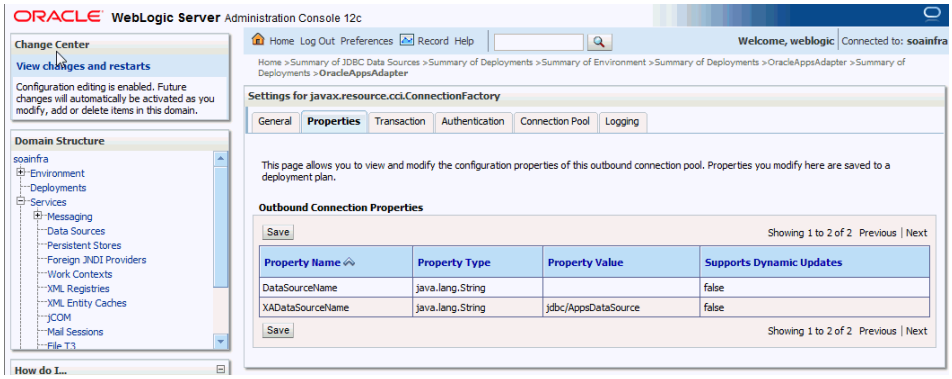
   3. Click the outbound instance name (such as `eis/Apps/Ebiz`) to display the outbound connection properties.

   4. Place your mouse in the Property Value column for the 'XADataSourceName' property name and click to enable the input text box.

      Enter `jdbc/AppsDataSource` in the text box, press [Enter] key, and then click **Save** to update the deployment plan.

      The Summary of JDBC Data Source page appears with `jdbc/AppsDataSource` listed in the table.

Outbound Connection Properties Page



5. Select 'OracleAppsAdapter' from Deployments and click **Update** to complete the update wizard. The new Connection Factory settings would be applied without restarting.

Creating the Application Server Connection

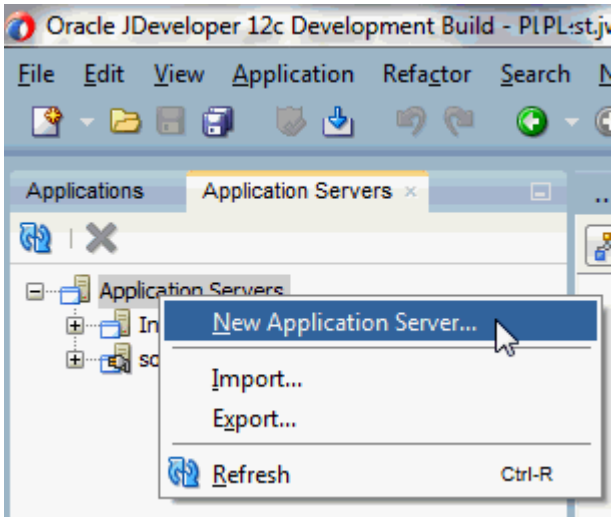
You must establish a connectivity between the design-time environment and the server you want to deploy it to. In order to establish such a connectivity, you must create the application server connection.

Use the following steps to create the application server connection:

1. From Oracle JDeveloper, select **Window > Application Servers** to open the Application Servers tab.

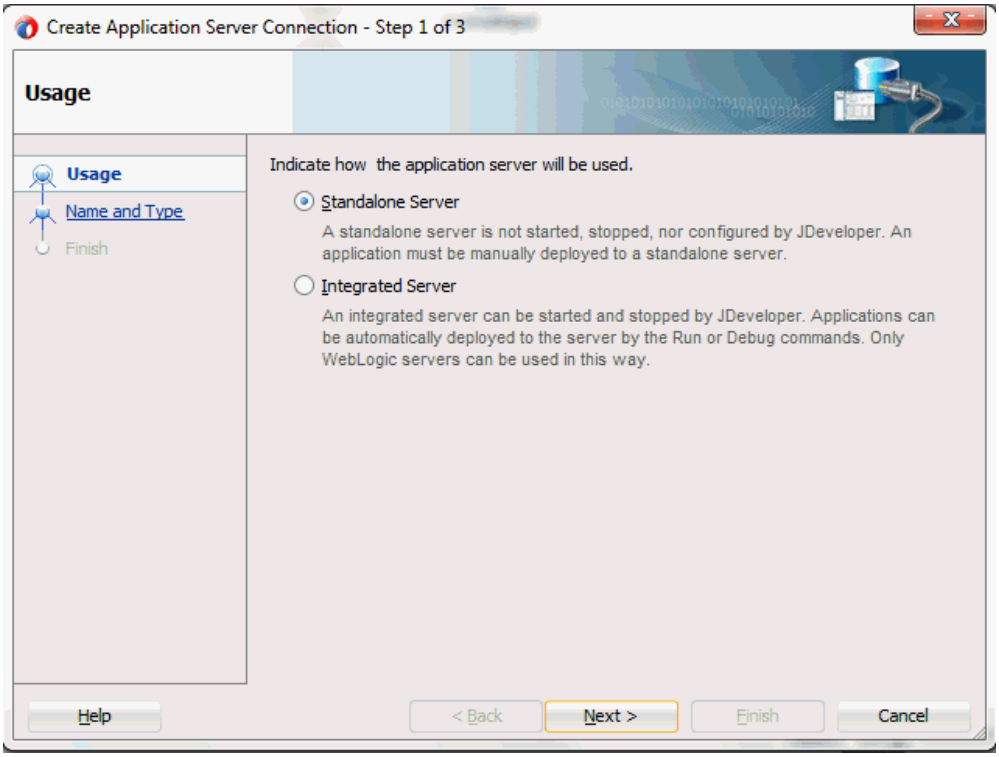
2. Right-click on the Application Servers node and select **New Application Server** to open the Create Application Server Connection wizard.

Create a New Application Server Connection



3. In the Usage page, leave the default **Standard Server** selection unchanged.

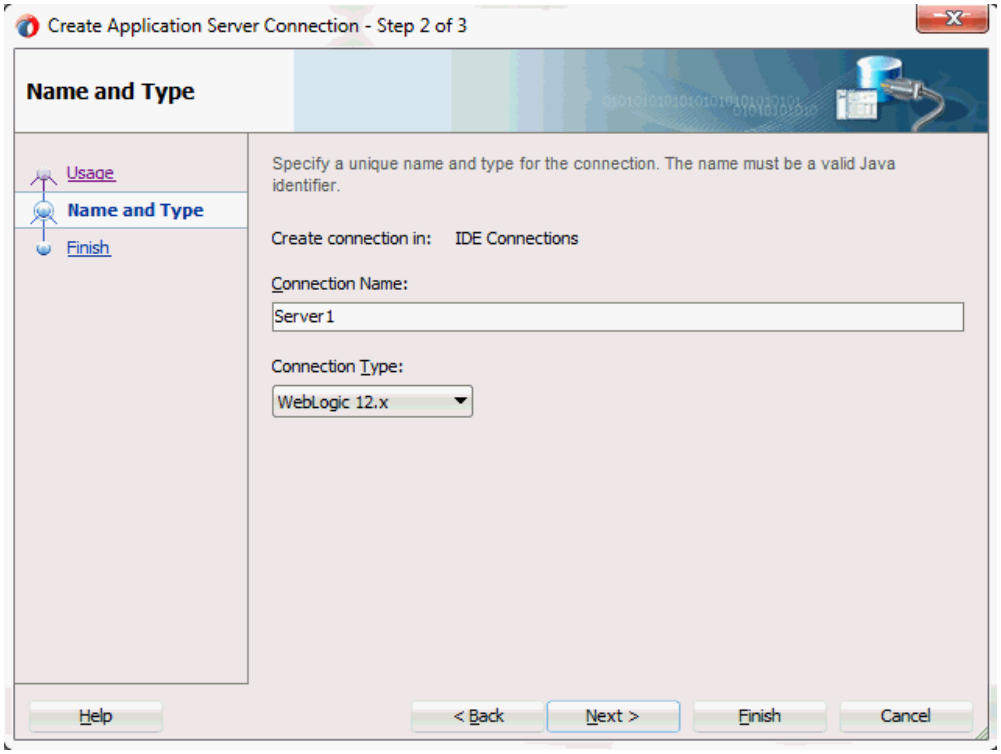
Select the Server Usage



Click **Next**.

4. Enter an appropriate name in the Connection Name field. Select **WebLogic 12.x** as the connection type.

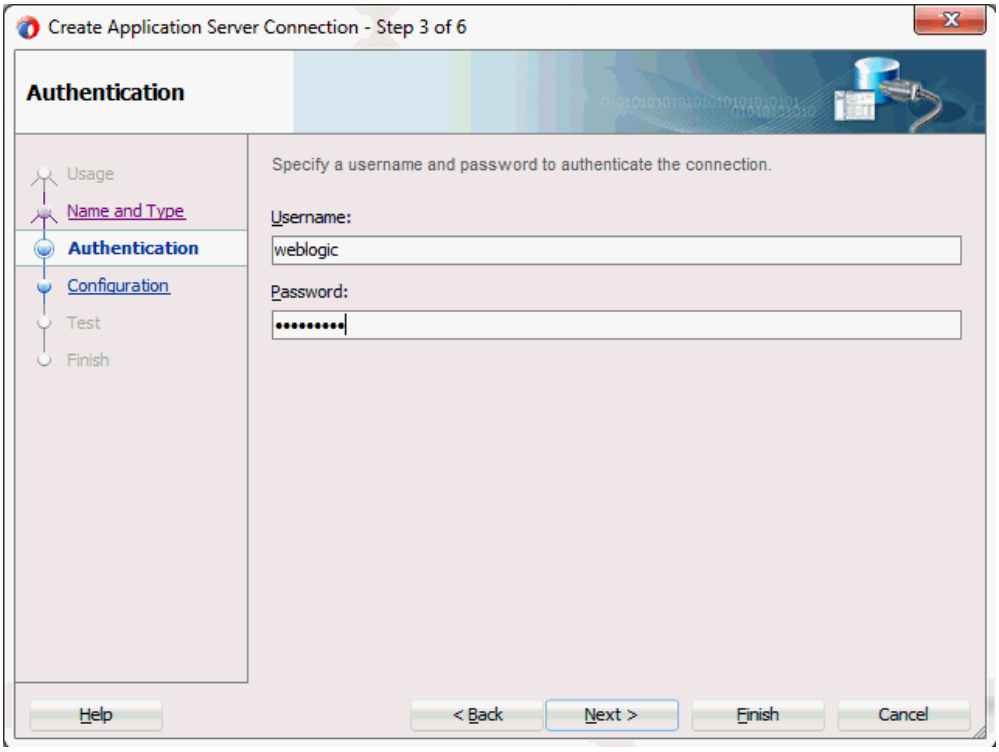
Select the Server Name and Type



Click **Next**.

5. Enter the username (such as `weblogic`) and password specified during installation and click **Next**.

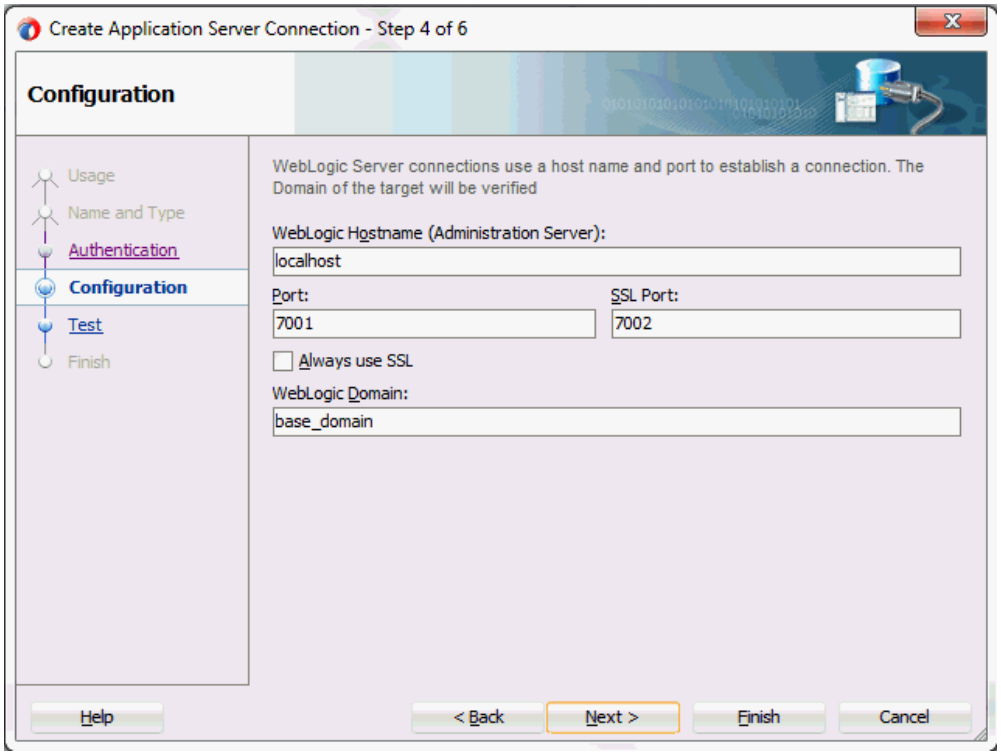
Enter Server Authentication Information



6. Enter the WebLogic Server connection host name and port information. Leave the default domain name unchanged in the WebLogic Domain field.

   Click **Next**.

Specifying Server Configuration Information



Click **Next**.

7. Click **Test Connection** to validate your server configuration. You should find success messages populated in the Status window.

   Click **Next** and **Finish** to complete the application server connection.

A

Troubleshooting and Workarounds

General Issues and Workarounds

This section describes the following issues and workarounds:

- **Sequence of Elements in XML and XSD Files Should Match**

  Order of elements passed as input to an API using Oracle E-Business Suite Adapter should strictly be the same as the order in which those elements appear under the `sequence` tag of the corresponding schema.

  Please note that in previous releases, Oracle E-Business Suite Adapter was non-restrictive and these elements could be passed in any order. However, the sequence of elements should now match the sequence in the corresponding schema.

- **Support for Business Events through Existing Partner Links**

  With the additional support for business event groups, perform the following steps to modify all existing partner links for outbound business events contained in an event group that is being listened to:

  > **Important:** This workaround is applicable only if Oracle E-Business Suite Adapter is being used for listening to Event Groups. It is NOT needed if the Business Event is not part of any Event Group that is being listened to.

  1. Updating the JCA property "MessageSelectorRule" for the existing Business Event partner links. This can be achieved by performing either one of the following steps:

     - Rerunning the Adapter Configuration (Recommended)

       Rerunning or editing the configuration wizard for an existing partner link of an outbound business event would update the artifacts with the

appropriate MessageSelectorRule.

- Manually Updating the JCA Properties

  Alternatively, modify the JCA property "MessageSelectorRule" for the business event partner link located in the file `XXX_apps.jca`, as stated below:

  > **Note:** `<EVENT_NAME>` is to be replaced with the name of the Business Event.

  Change existing property from:

  ```
  <property name="MessageSelectorRule"
  value="tab.user_data.event_name = '<EVENT_NAME>'"/>
  ```

  To the following:

  ```
  <property name="MessageSelectorRule"
  value="tab.user_data.event_name = '<EVENT_NAME>' AND
  tab.user_data.getvalueforparameter('GROUP') IS NULL"/>
  ```

2. Delete the existing subscriber on WF_BPEL_Q of the corresponding Oracle E-Business Suite applications.

   1. Undeploy the SOA Composite application containing the Business Event partner link.

   2. Delete the subscriber using the following snippet:

      > **Note:** `<CONSUMER_NAME>` is to be replaced with the name of the subscriber. It can be found in the `XXX_apps.jca` file as the value for the JCA property 'Consumer'.

      ```
      begin
      subscriber := sys.aq$_agent('<CONSUMER_NAME>', NULL, NULL);
      dbms_aqadm.remove_subscriber(queue_name => 'WF_BPEL_Q',
      subscriber => subscriber);
      end;
      ```

   3. Redeploy the SOA Composite application.

For more information about business event groups, see Business Event Groups, page 6-3.

- **"DataSecurityCheck" Requires "IRepOverloadSeq" to be Passed for Overloaded Functions**

  Oracle E-Business Suite Adapter uses the following two new JCA properties to handle access control for overloaded PL/SQL APIs. This feature works in

conjunction with the grants created from the Integration Repository user interface, and it does not depend on any profile options.

- `DataSecurityCheck`
- `IRepOverloadSeq`

If a user wants the Function Security Authorization check to be performed, the following property information needs to be added in the WSDL file `XX_apps.jca`:

```
<property name="DataSecurityCheck" value="yes"/>
```

However, the other property `IRepOverloadSeq` is derived automatically by Oracle E-Business Suite Adapter at the design time during creation of the partner link. Based on these two properties, the function security check would be performed for the username, which is passed as a header property.

Please note that this security support would work with all interfaces which are available in the Integration Repository. For other interfaces, you need to enable the function security through the profile option "EBS Adapter for BPEL, Function Security Enabled" (EBS_ADAPTER_FUNCTION_SEC_ENABLED).

For more information about security support for overloaded functions, see Function Security Support for Overloaded Functions, page 4-21.

- **Populating Default Values for Record Types While Using PL/SQL APIs**

  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. Oracle E-Business Suite Adapter 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>
```

  When such a PL/SQL API is invoked at run time, the values being passed for the record type parameters should have the attribute `xsi:nil="true"`. This can be achieved in more than one way at design time by using 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.

  Additionally, the value for each parameter can be checked at design time. For example, if the attribute value is not passed in a Transform activity, the parameter can be set with `xsi:nil="true"` using a function. If the value `xsi:nil="true"` is passed in the payload, Oracle E-Business Suite Adapter automatically populates the default value for the attribute. This applies to the parameters within or outside a

record type.

- **Re-creating Wrapper Packages While Using Existing PL/SQL SOA Composites Against a Different Release Instance**

  When a user has a SOA composite of a PL/SQL API created against an Oracle E-Business Suite Release 11i instance and intends to use it against the Release 12 instance or vice versa, for the compatibility in the target instance, the wrapper package of the SOA composite must be re-created. This approach updates the signature in the generated wrapper SQL file for the target instance and avoids the possible confusion whether the signature is the same or has changed in the target instance.

- **WSDL Context Information Default Values**

  Applications context is used in passing header variables that may be required in a business activity or to complete a BPEL process. When setting the context, it takes 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 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. 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 Invoke activity.

  For more information about applications context, see Supporting for Normalized Message Properties, page 4-7.

- **Correlation ID Defaults to BPEL for XML Gateway Transactions**

  The Adapter Configuration wizard of Oracle E-Business Suite Adapter 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 E-Business Suite 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 E-Business Suite, 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 E-Business Suite.

- **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" .../>
     ```

- **Cannot Create a Partner Link If the Underlying API Has Been Re-created**

  The generation of a wrapper for an API that was re-created 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 re-create 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 workaround, exit JDeveloper BPEL Designer and restart it after recreating the stored procedure, but before attempting to create the second partner link.

B

Oracle E-Business Suite Adapter Properties

Overview

This appendix describes the properties applicable to the Oracle E-Business Suite Adapter, including:

- JCA Properties for Oracle E-Business Suite Adapter, page B-1
- JCA Properties for Oracle E-Business Suite Adapter: Normalized Properties, page B-4
- Binding Properties for Oracle E-Business Suite Adapter, page B-7

For more information about JCA properties, see:

- Section A.7, Generic Oracle JCA Adapter Properties, *Oracle Fusion Middleware User's Guide for Technology Adapters* for properties applicable to all Oracle JCA Adapters
- Section 33.1.2.8, Oracle JCA Adapters Endpoint Properties, *Oracle Fusion Middleware Administrator's Guide for Oracle SOA Suite and Oracle Business Process Management Suite*
- Appendix H, Normalized Message Properties, *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*

JCA Properties for Oracle E-Business Suite Adapter

The following table lists the JCA properties used for Oracle E-Business Suite Adapter:

Property	Description
SchemaName	The schema of the Oracle E-Business Suite. Usually points to the "Apps" or Apps-equivalent schema.
PackageName	Name of the PL/SQL package or the wrapper package.
ProcedureName	Name of the procedure or function to be invoked. Points to Wrapper procedure / function if present.
DataSecurityCheck	Passing this property with a value of "yes" enables function security check for PL/SQL APIs and Concurrent Programs.
QueryTimeout	This property specifies the maximum number of seconds that the JDBC driver should wait for the specified stored procedure or function to execute. When the threshold is exceeded, SQLException is thrown. If the value is zero, then the driver waits indefinitely.
APIErrorHandler	Passing this property enables the caller to retrieve error messages from FND_MSG_PUB message stack, without the need for another explicit call to retrieve them.
GetActiveUnitOfWork	If it is set to "true", this advanced setting forces all invoked activities in the same global transaction to use the same SQL connection if going to the same database. This guarantees that later invoked activities can see the changes of earlier invoked activities. (However, this may not be needed if using emulated two-phase commit, which automatically uses the same connection.)

Property	Description
IRepOverloadSeq	This is automatically populated and is not user-configurable. It works in conjunction with "DataSecurityCheck" property. **Note:** This property should not be altered by end users and modification might lead to undesirable results.
IRepInternalName	Holds an internal value used by Oracle E-Business Suite Adapter. **Note:** This property should not be altered by end users and modification might lead to undesirable results.

The following JCA properties are specific to Enqueue and Dequeue operations that are applicable for XML Gateway and Business Event Outbound interfaces:

Property	Description
QueueName	The name of the AQ Queue being read from or written to.
DatabaseSchema	The schema of the Oracle E-Business Suite. Usually points to the "Apps" or Apps-equivalent schema.
Consumer	Name of the AQ Consumer, which dequeues the messages from the AQ Queue.
MessageSelectorRule	Specifies the condition based on payload or queue header values, to screen the messages and accept only those that meet the condition. **Note:** This property should not be altered by end users and modification might lead to undesirable results.

Property	Description
AppsEventSchema	Passing one of the following event schema selection options to be used for the business event payload: • No Schema • Any Schema • Specify Schema For information on each schema selection option, see Specifying Event Schema, page 6-16.

JCA Properties for Oracle E-Business Suite Adapter: Normalized Properties

The following table lists the normalized message properties used for Oracle E-Business Suite Adapter:

Property	Description
jca.apps.Username	This property specifies the Oracle E-Business Suite application username information. In the case of a null or empty value, the default Username is SYSADMIN,
jca.apps.Responsibility	This property specifies the Oracle E-Business Suite application responsibility information for setting applications context. The default Responsibility is System Administrator.
jca.apps.ORG_ID	This property specifies the organization ID used in Oracle E-Business Suite application.
jca.apps.RespApplication	This property specifies the responsibility application short name information for setting applications context. It accepts Application Short Name (such as FND) as its value.

Property	Description
jca.apps.SecurityGroup	This property specifies the security group information for setting applications context. The default Security Group Key is Standard,
jca.apps.NLSLanguage	This property specifies the NLS language information. The default NLS Language is US, It is used for multiple language support (MLS) feature for Oracle E-Business Suite.
jca.apps.ecx.TransactionType	This property specifies 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. This property sets XML Gateway message header information required for XML Gateway inbound transactions.
jca.apps.ecx.TransactionSubtype	This property specifies 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. Similar to jca.apps.ecx.TransactionType, this property sets XML Gateway message header for XML Gateway inbound transactions
jca.apps.ecx.PartySiteId	This property specifies the party site identifier for an inbound XML document. The party site must be the same as the Source Trading Partner location defined in the Trading Partner form. This property is used to set XML Gateway message header.
jca.apps.ecx.MessageType	This property sets XML Gateway message header for payload message format. This defaults to XML.

Property	Description
jca.apps.ecx.MessageStandard	This property sets XML Gateway 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.
jca.apps.ecx.DocumentNumber	This property sets document number for XML Gateway. The document identifier used to identify the transaction, such as a purchase order or invoice number.
jca.apps.ecx.ProtocolType	This property specifies the protocol type for setting XML Gateway message header.
jca.apps.ecx.ProtocolAddress	This property specifies the protocol address for setting XML Gateway message header.
jca.apps.ecx.Username	This property specifies the username for setting XML Gateway message header. USERNAME is defined in the Trading Partner table.
jca.apps.ecx.Password	This property specifies the password associated with the USERNAME that is defined in the Trading Partner table.
jca.apps.ecx.Attribute1	This property specifies the Attribute1 parameter that may be defined by the base application.
jca.apps.ecx.Attribute2	This property specifies the Attribute2 parameter that may be defined by the base application.

Property	Description
jca.apps.ecx.Attribute3	This property specifies the Attribute3 parameter. For outbound messages, this parameter 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. For more information, see Destination Trading Partner Location Code, *Oracle XML Gateway User's Guide*.
jca.apps.ecx.Attribute4	This property specifies the Attribute4 parameter that may be defined by the base application.
jca.apps.ecx.Attribute5	This property specifies the Attribute5 parameter that may be defined by the base application.
jca.apps.ecx.Payload	This property specifies the payload information for XML Gateway transactions.

Binding Properties for Oracle E-Business Suite Adapter

The following table lists the binding properties used for Oracle E-Business Suite Adapter:

Property	Description
DequeueTimeOut	It is the interval after which the dequeue() API will time out if no message is received on the inbound queue. The default value is 1s.

Property	Description
ConnectionRetryDelay	The time for which Oracle E-Business Suite Adapter will wait before trying to re-create a connection after a connection is lost. The default value is 15s

Index

A

agent
  See business events concepts
APIs
  See PL/SQL APIs
applications context
  example, 4-2
Applications Context
  Design-Time Tasks, 4-8
  Multiple Language Support, 4-12
  multiple organization access control (MOAC), 4-11
  Normalized Message Properties, 4-7

B

B2B, 5-2
base tables, 8-2
business events, 6-1
  troubleshooting, 6-46
business events concepts, 6-2
  agent, 6-3
  deferred subscription processing, 6-3
  event, 6-2
  event activity, 6-3
  event data, 6-3
  Event Groups, 6-3
  event key, 6-2
  event message, 6-3
  event subscription, 6-3
business events outbound
  Assign activity, 6-31
  configuring Oracle E-Business Suite Adapter, 6-4
  creating a new SOA Composite application, 6-6
  creating a partner link, 6-10
  deploying the SOA Composite application, 6-34
  design-time prerequisites, 6-5
  design-time tasks, 6-4
  file adapter partner link, 6-22
  Invoke activity, 6-29
  Receive activity, 6-20
  run-time tasks, 6-34
  testing the SOA Composite application , 6-38

C

concurrent program
  adding a new partner link for file adapter, 7-27
concurrent programs
  adding a partner link, 7-10
  configuring Oracle E-Business Suite Adapter, 7-3
  configuring the Assign activity, 7-46
  configuring the Invoke activity, 7-39
  creating a new SOA Composite application, 7-4
  deploying the SOA Composite application, 7-50
  design-time steps, 7-3
  overview, 7-1
  prerequisite to configure, 7-3

run-time steps, 7-50
testing the SOA Composite application, 7-54
troubleshooting, 7-59
verifying records, 7-58
configure flexfield mappings at design time
modifying flexfield mapping, 4-119
review flexfield mapping configuration, 4-122
configure flexfields at design time
configure a new custom mapping, 4-37
importing a flexfield mapping, 4-115
connection information
Applications Database Connection, 11-7
data source configuration, 11-1
WebLogic Server Connection, 11-10

D

deferred subscription processing, 6-3
See also business events concepts
design-time tasks
business events outbound, 6-4

E

EDI
adding a new partner link for file adapter, 10-14
adding a partner link, 10-9
configuring Oracle E-Business Suite Adapter, 10-2
configuring the Assign activity, 10-25
configuring the Invoke activity, 10-20
creating a new SOA Composite application, 10-3
deploying the SOA Composite application, 10-28
design-time steps, 10-2
overview, 10-1
prerequisites to configuring Oracle E-Business Suite Adapter, 10-2
run-time steps, 10-27
testing the SOA Composite application, 10-31
verifying records in Oracle Applications, 10-33
Errors and Exceptions Handling
Enabling Function Error Handling at Design Time, 4-132
Example of Error Handling Using APIErrorHandler JCA property, 4-133
event
See business events concepts
event activity, 6-3
event data, 6-3
event key, 6-2
event message, 6-3
event subscription, 6-3
Example of Error Handling Using APIErrorHandler JCA property
Create a new SOA Composite application, 4-134
Creating a CatchAll Activity, 4-139
Creating a Partner Link, 4-136
Creating a Partner Link for File Adapter, 4-137
Creating Assign Activities, 4-143
Creating Invoke Activities, 4-140
Deploying and Testing SOA Composite Application , 4-149
Sample Payload, 4-153

F

Features, 3-1
flexfield Support
configure flexfield mappings
Key Flexfield mappings for APIs, 4-38
configure flexfield mappings at design time
Descriptive Flexfield mappings for APIs, 4-59
Key and Descriptive Flexfield mappings for Interface Tables, 4-85
configure flexfields
Key Flexfields for Interface Tables, 4-79
configure flexfields at design time overview, 4-34
Flexfield Support
key elements, 4-29

I

integration architecture, 5-2
interface tables, 8-2
adding a file adapter partner link, 8-22
adding a partner link, 8-8
configuring Oracle E-Business Suite Adapter, 8-3
configuring the Assign activity, 8-31
configuring the Invoke activity, 8-28

creating a new SOA Composite application, 8-3
deploying the SOA Composite application, 8-34
design-time steps, 8-3
overview, 8-2
prerequisites to configure, 8-3
run-time steps, 8-33
testing the SOA Composite application, 8-37

J

JCA Properties
Overview, B-1

L

Logging
Correlating Messages, 4-131
enabling logging, 4-125
overview, 4-124
Searching and Viewing Logs, 4-129

M

message queues, 5-3
inbound queues, 5-3
outbound queues, 5-4

O

Oracle E-Business Suite Adapter
Applications Context, 4-1
features, 2-2
Flexfield Support Overview, 4-27
High Availability, 3-6
integration with BPEL PM, 2-4
interfaces, 3-1
issues and workarounds, A-1
Monitoring Oracle E-Business Suite Adapter Configurations, 3-7
Oracle E-Business Suite Support, 3-4
Oracle Integration Repository, 3-3
overview, 2-1
Oracle E-Business Suite Adapter
architecture, 2-3
installing, 2-4
integration with Oracle WebLogic Server, 2-6
Oracle E-Business Suite Adapter Concepts
Error and Exception Handling, 4-132
Function Security, 4-20
Module Browser, 4-14
overview, 4-1
Using J2EE Data Source, 4-155
Oracle E-Business Suite Adapter Properties
Binding Properties, B-7
JCA properties, B-1
Normalized Properties, B-4

P

PL/SQL APIs
adding a new partner link, 9-8
adding a new partner link for file adapter, 9-19
configuring Oracle E-Business Suite Adapter, 9-2
configuring the Assign activity, 9-35
configuring the Invoke activity, 9-31
configuring the Transform activity, 9-33
creating a new SOA Composite application, 9-4
deploying the SOA Composite application, 9-39
design-time steps, 9-2
overview, 9-1
prerequisite to configure, 9-2
run-time steps, 9-39
testing the SOA Composite application, 9-42
troubleshooting, 9-46
prerequisites
business events outbound, 6-5

R

Runtime Configuration for Oracle E-Business Suite Adapter , 11-1
run-time tasks
business events outbound, 6-34

S

standards-based messaging, 5-2

V

views, 8-2
adding a new partner link for file adapter, 8-

58
adding a partner link, 8-46
configuring Oracle E-Business Suite Adapter, 8-40
configuring the Assign activity, 8-68
configuring the Invoke activity, 8-64
creating a new SOA Composite application, 8-41
deploying the SOA Composite application, 8-71
design-time steps, 8-40
overview, 8-2
prerequisites to configure, 8-40
run-time steps, 8-71
testing the SOA Composite application, 8-74

W

wrapper APIs, 9-25
DEFAULT clause handling in wrapper procedures, 9-30
describing parameters with DEFAULT clause, 9-28

X

XML Gateway
overview, 5-2
troubleshooting, 5-76
XML gateway envelope, 5-4
ATTRIBUTE3, 5-5
DOCUMENT_NUMBER, 5-5
MESSAGE_STANDARD, 5-4
MESSAGE_TYPE, 5-4
parameters defined by the application, 5-6
parameters not used, 5-6
PARTY_SITE_ID, 5-5
PASSWORD, 5-5
PAYLOAD, 5-6
PROTOCOL_ADDRESS, 5-5
PROTOCOL_TYPE, 5-5
TRANSACTION_SUBTYPE, 5-5
TRANSACTION_TYPE, 5-4
USERNAME, 5-5
XML gateway inbound
adding a partner link for the file adapter, 5-18
configuring Oracle E-Business Suite Adapter, 5-6
configuring the Assign activity, 5-31
configuring the Invoke activity, 5-24
creating a new SOA Composite application, 5-8
creating a partner link, 5-13
deploying the SOA Composite application, 5-34
design-time steps, 5-6
prerequisites to configure, 5-7
run-time steps, 5-34
testing the SOA Composite application, 5-37
verifying records, 5-39
XML gateway outbound
design-time steps, 5-43
prerequisites to configure, 5-44
run-time steps, 5-67
testing the SOA Composite application, 5-71
XML Gateway outbound
deploying the SOA Composite application, 5-68
XML Gateway Outbound
adding a new partner link, 5-52
adding a new partner link for file adapter, 5-59
Adding a Receive Activity, 5-57
configuring the Assign activity, 5-65
configuring the Invoke activity, 5-64
creating a new SOA Composite application, 5-46