

# **Oracle® SOA Suite**

Quick Start Guide

10g (10.1.3.1.0)

**B28938-01**

September 2006

Oracle SOA Suite Quick Start Guide, 10g (10.1.3.1.0)

B28938-01

Copyright © 2006, Oracle. All rights reserved.

Primary Author: Deborah Steiner

Contributors: Eric Belden, Roman Dobrik, Mark Kennedy, Raghu Kodali, Lynn Munsinger, Robin Whitmore, Robin Zimmerman

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

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

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

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

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

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

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

---

---

# Contents

<b>Preface</b> .....	v
Audience .....	v
Documentation Accessibility .....	v
Related Documents .....	vi
Conventions .....	vi
<b>1 Introduction to the Oracle SOA Suite</b>	
1.1 Why the Shift to Service-Oriented Architecture? .....	1-1
1.2 What Does the Oracle SOA Suite Provide? .....	1-1
1.3 Introduction to SOA Order Booking Application .....	1-5
<b>2 Setting Up the SOA Order Booking Application</b>	
2.1 Checking for and Installing Prerequisites .....	2-1
2.1.1 Task 1: Install an Oracle Database .....	2-1
2.1.2 Task 2: Install JDeveloper Studio 10.1.3.1.0 .....	2-1
2.1.3 Task 3: Install Oracle SOA Suite 10.1.3.1.0 .....	2-2
2.1.4 Task 4: Install the SOA Order Booking Application .....	2-2
2.2 Tasks for Setting Up the SOA Order Booking Application .....	2-2
2.2.1 Task 1: Install Oracle Database Schema .....	2-3
2.2.2 Task 2: Configure Oracle SOA Suite .....	2-3
2.2.3 Task 3: Note the Name of the OC4J Instance Running the Oracle BPEL Server .....	2-6
2.2.4 Task 4: Note the Important Port Numbers .....	2-6
2.2.5 Task 5: Create Connections in JDeveloper .....	2-7
2.2.6 Task 6: Install and Deploy the SOA Order Booking Application .....	2-8
2.2.7 Task 7: Configure the ESB Port .....	2-13
2.2.8 Task 8: Familiarize Yourself with the Schema .....	2-14
<b>3 Running and Monitoring the SOA Order Booking Application</b>	
3.1 Placing an Order for Over \$1000 .....	3-1
3.1.1 Task 1: Start Oracle BPEL Worklist Application .....	3-1
3.1.2 Task 2: Place Order in Web Client .....	3-2
3.1.3 Task 3: View the Order in the Oracle Enterprise Manager 10g BPEL Control Oracle BPEL Control .....	3-4
3.1.4 Task 4: Use the Oracle BPEL Worklist Application to Approve Order .....	3-8
3.1.5 Task 5: View Approval in the Oracle BPEL Control .....	3-10

3.1.6	Task 6: View Instances in Oracle Enterprise Manager 10g ESB Control .....	3-14
3.2	Increasing Approval Order Amount from \$1000 to \$2000 and Resubmitting Order ....	3-16
3.2.1	Task 1: Use Rule Author to Increase Approval Order Amount .....	3-17
3.2.2	Task 2: Place Order in Web Client.....	3-20
3.2.3	Task 3: View the Order Approval in the Oracle BPEL Control .....	3-21

## 4 Adding BPEL and ESB Design Elements with Oracle JDeveloper

4.1	Introduction to the JDeveloper BPEL Designer.....	4-1
4.2	Adding a Credit Rating Service to the SOAOrderBooking Process .....	4-9
4.2.1	Task 1: Create a Database Connection for the Database Containing the Social Security Number .....	4-10
4.2.2	Task 2: Use a Database Adapter As a Service to Access the Database .....	4-10
4.2.3	Task 3: Create getCreditRating Scope Activity .....	4-12
4.2.4	Task 4: Create Invoke Activity to Call getSsn Service.....	4-13
4.2.5	Task 5: Assign Value to Input Variable .....	4-14
4.2.6	Task 6: Install the Credit Rating Service.....	4-17
4.2.7	Task 7: Create CreditRatingService Partner Link .....	4-18
4.2.8	Task 8: Create Invoke Activity to Call CreditRatingService .....	4-19
4.2.9	Task 9: Assign Data to the Input Variable for CreditServiceRating.....	4-21
4.2.10	Task 10: Redeploy SOAOrderBooking .....	4-23
4.2.11	Task 11: Test New Functionality by Placing a New Order.....	4-24
4.2.12	Task 12: View the Order Approval in the Oracle BPEL Control .....	4-25
4.3	Introduction to the JDeveloper ESB Designer .....	4-26
4.4	Adding a New Shipping Target to FulfillmentESB Project .....	4-27
4.4.1	Task 1: Add the DHLShipment Service to ESB .....	4-27
4.4.2	Task 2: Redeploy FulfillmentESB .....	4-32
4.4.3	Task 3: Test New Functionality By Placing a New Order .....	4-32
4.4.4	Task 4: View the Order in the Oracle BPEL Control .....	4-33
4.4.5	Task 5: View the DHLShipment Status in the Oracle ESB Control .....	4-33
4.5	Learning More About Oracle SOA Suite .....	4-35

## Index

---

---

# Preface

This guide provides an overview of the Oracle SOA Suite, as well as a step-by-step tutorial for using an application developed with Oracle SOA Suite components.

This preface includes the following topics:

- [Audience](#)
- [Documentation Accessibility](#)
- [Related Documents](#)
- [Conventions](#)

## Audience

This document is intended for all users who want to quickly understand and get started using Oracle SOA Suite.

## Documentation Accessibility

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

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

### Accessibility of Code Examples in Documentation

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

### Accessibility of Links to External Web Sites in Documentation

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

### TTY Access to Oracle Support Services

Oracle provides dedicated Text Telephone (TTY) access to Oracle Support Services within the United States of America 24 hours a day, seven days a week. For TTY support, call 800.446.2398.

## Related Documents

For more information related to the SOA Order Booking application, see the following documents in the Oracle Application Server 10g Release 3 (10.1.3.1.0) documentation set:

- *Oracle Application Server Tutorial*
- *Oracle SOA Suite Developer's Guide*

## Conventions

The following text conventions are used in this document:

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

---

# Introduction to the Oracle SOA Suite

This chapter provides an overview of Oracle SOA Suite to design, develop, deploy and monitor an SOA application.

This chapter includes the following sections:

- [Section 1.1, "Why the Shift to Service-Oriented Architecture?"](#)
- [Section 1.2, "What Does the Oracle SOA Suite Provide?"](#)
- [Section 1.3, "Introduction to SOA Order Booking Application"](#)

## 1.1 Why the Shift to Service-Oriented Architecture?

Many companies are addressing the complexity of their application and IT environments with Service-Oriented Architecture (SOA). SOA provides an enterprise architecture that supports building connected enterprise applications. SOA facilitates the development of enterprise applications as modular business Web services that can be easily integrated and reused, creating a truly flexible, adaptable IT infrastructure.

Web services provide interoperability of proprietary software. Web services standards, including Web Services Description Language (WSDL), extensible markup language (XML), and Simple Object Access Protocol (SOAP), have emerged as an effective and highly interoperable platform for publishing services. In addition, high performance binding frameworks enable enterprises to access legacy systems and native Java code without having to wrap them in a SOAP interface.

Making Web services work is a two-step process:

1. Publish the service.

Publishing a service involves taking a function within an existing application or system and making it available in a standard way.

2. Compose, or orchestrate, the services into business flows.

Orchestration involves composing multiple services into an end-to-end business process. The business process execution language (BPEL) language supports this orchestration.

## 1.2 What Does the Oracle SOA Suite Provide?

Oracle SOA Suite is a complete set of service infrastructure components for creating, deploying, and managing services. Oracle SOA Suite enables services to be created, managed, and orchestrated into composite applications and business processes. Additionally, you can adopt it incrementally on a project by project basis and still

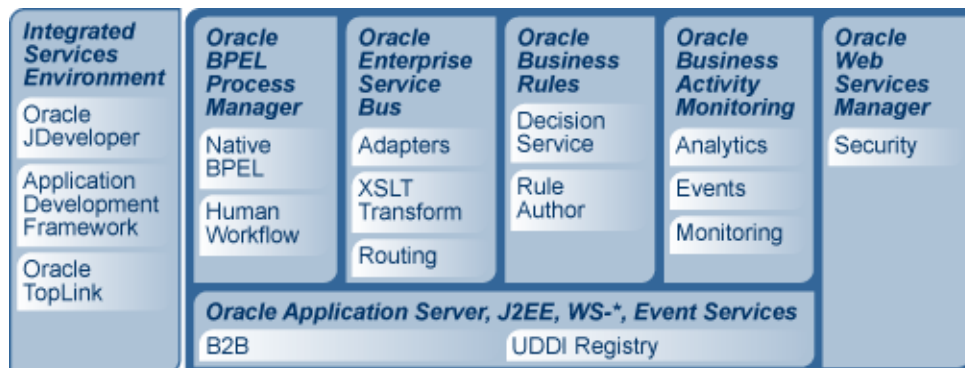
benefit from the common security, management, deployment architecture, and development tools that you get out of the box.

Oracle SOA Suite is a standards-based best-of-breed technology suite that consists of the following:

- Integrated Service Environment (ISE) to develop services
- Oracle BPEL Process Manager to orchestrate services into business processes
- ESB to connect existing IT systems and business partners as a set of services
- Oracle Business Rules for dynamic decisions at runtime that can be managed by business users or business analysts
- OracleAS Integration Business Activity Monitoring to monitor services and disparate events and provide real-time visibility into the state of the enterprise, business processes, people, and systems.
- Oracle Web Services Manager to secure and manage authentication, authorization, and encryption policies on services that is separate from your service logic
- UDDI registry to discover and manage the lifecycle of Web services.
- Oracle Application Server 10g Release 3 (10.1.3) to provide a complete Java 2, Enterprise Edition (J2EE) 1.4-compliant environment for your J2EE applications.

Figure 1–1 shows the Oracle SOA Suite architecture.

**Figure 1–1 Oracle SOA Suite Architecture**



### Integrated Service Environment

Oracle JDeveloper (JDeveloper) is the development component of Oracle SOA Suite. It forms a comprehensive ISE for developing, composing, and orchestrating services into business processes. Business processes can be deployed, registered, and consumed from several types of user interfaces, including desktop clients, browsers, and mobile and telnet devices.

JDeveloper enables developers to model, create, discover, assemble, orchestrate, test, deploy, and maintain composite applications based on services. JDeveloper supports SOA principles and XML Web services standards, as well as traditional Java, J2EE, and PL/SQL component and modular code mechanisms.

Oracle ADF is a model-driven SOA framework that automates and manages businesses and data services and provides a standard data-and-service-binding layer based on JSR 227 that can be used with process flows, page flows, and service invocations. Oracle ADF implements SOA design practices and makes user interfaces as loosely coupled as services themselves.



Oracle TopLink is a data service framework that enables access to relational and XML data. It provides visual mapping tools for facilitating object to relational and object to XML mapping. TopLink and Oracle ADF frameworks simplify creation of business and data services that can be invoked from rich Web interfaces of service-oriented applications.

### **Oracle BPEL Process Manager (including Human Workflow)**

Oracle BPEL Process Manager provides a framework for easily designing, deploying, monitoring, and administering processes based on BPEL standards. Oracle BPEL Process Manager provides support for the following features:

- Web service standards such as XML, SOAP, and WSDL
- Dehydration (enables the states of long-running processes to be automatically maintained in a database) and correlation of asynchronous messages
- Parallel processing of tasks
- Fault handling and exception management during both design time and run time
- Event timeouts and notifications
- Compensation mechanisms for the implementation of long-running transactions
- Scalability and reliability of processes
- Management and administration of processes
- Version control
- Audit trails for tracing business flow history
- Installation on multiple operating systems and integration with multiple application servers (for example, Oracle Application Server, BEA WebLogic, and JBoss) and databases.

Oracle BPEL Process Manager adds value and ease of use to BPEL functionality by providing support for the following in the JDeveloper BPEL Designer:

- Transformations, workflows, worklists, notifications, and sensors
- Technology adapters, including file, FTP, database, advanced queuing (AQ), Java Messaging Service (JMS), Oracle Applications for Oracle E-Business Suite, and WebSphere MQ
- Third-party adapters, including J.D. Edwards OneWorld, PeopleSoft, SAP R/3, Siebel, Tuxedo, CICS, VSAM, IMS/TM, and IMS/DB

### **Oracle Enterprise Service Bus (ESB)**

An enterprise service bus moves data among multiple endpoints, both within and outside of an enterprise. It uses open standards to connect, transform, and route business documents (as Extensible Markup Language (XML) messages), among disparate applications. It enables monitoring and management of business data, with minimal impact on existing applications. An enterprise service bus is the underlying infrastructure for delivering a service-oriented architecture (SOA) and event-driven architecture (EDA).

ESB is the foundation for services using SOA and EDA. At its core, it is a loosely coupled application framework that provides your business with increased flexibility, reusability, and overall responsiveness in a distributed, heterogeneous, message-oriented environment using industry standards.

### **Oracle Business Rules**

Oracle Business Rules enables dynamic decisions at runtime allowing, among other features, applications to rapidly adapt to regulatory and competitive pressures. This increased agility is possible because business analysts using Oracle Business Rules can create and change business rules that are separated from the application code. By using Oracle Business Rules, business analysts can change business rules without stopping business processes. Also, externalizing business rules allows business analysts to manage business rules directly, without involving programmers.

### **OracleAS Integration Business Activity Monitoring**

OracleAS Integration Business Activity Monitoring (BAM) gives business executives the ability to monitor their enterprise business services in real-time and to correlate their KPIs (key performance indicators) to the actual business process. Oracle BAM provides the ability to aggregate service metrics and deliver actionable information on critical business service parameters to user. Oracle BAM delivers information to users through visual dashboards and alerts, improving effectiveness of operations and taking informed decisions. Oracle BAM also provides users the ability to change the business processes and take corrective action if the business environment changes. Oracle BAM is a complete solution for building real-time operational dashboards, monitoring and alerting applications.

### **Oracle Web Services Manager**

Oracle Web Services Manager is a security administrator's environment designed to secure access to Web services and monitor activities performed on protected Web services.

Oracle Web Services Manager includes two main parts: A policy decision point (PDP) and policy enforcement points (PEPs). The PDP includes security and management components accessed through a Web-based management console providing Oracle Enterprise Manager's look and feel. PEPs are interceptors that can be either agents or gateways. Agents run in the same container as the Web services they protect whereas gateways are independent processes analogous to proxy servers. Agents and gateways can be used in combination to ensure end-to-end Web services security.

Typically, a security administrator sets up the environment by registering the Web services to be protected and managed. The administrator then defines declarative security policies (no coding) for each registered service, for example, how credentials are extracted for authentication, how messages are encrypted, decrypted, and signed, and how events are logged.

PEPs intercept requests to the registered Web services and enforce the policies defined by the administrator. In addition, PEPs collect security and management information that they transmit to the PDP. The PDP aggregates that information into management metrics and formats it for display in graphical charts for easy reading. Management metrics is obtained from service-level agreements (SLAs) defined by the administrator using the management console.

Oracle Web Services Manager seamlessly integrates with Oracle BPEL Process Manager and ESB, that is, Oracle Web Services Manager can secure BPEL and ESB processes. In addition, Oracle Web Services Manager integrates with the Oracle Identity Management suite, in particular, Oracle Web Services Manager can extract and transform Oracle Access Manager cookie information thus providing an end-to-end security solution, from browser to networks of Web services.

**OracleAS UDDI Registry**

OracleAS UDDI Registry provides a key component of any SOA with a configurable, scalable, secure repository of Web services that can be managed, discovered and governed by Oracle Fusion Middleware. The OracleAS UDDI Registry meets the core service management needs of any enterprise:

- Enables service providers to publish and advertise their offerings
- Allows service consumers to find, access, and invoke services that meet defined criteria
- Provides critical features for SOA governance

Integration is provided with other products in the Oracle Fusion Middleware family, including Oracle BPEL Control, Oracle Web Services Manager, and JDeveloper, enabling users to query the Registry for published services.

OracleAS UDDI Registry fully supports the latest UDDI V3 specification, and delivers the broadest range of features of any SOA registry available today.

**Oracle Application Server**

Oracle Application Server is a standards-based application server that provides a comprehensive and fully integrated platform for running Web sites, J2EE applications, and Web services.

## 1.3 Introduction to SOA Order Booking Application

The SOA Order Booking application demonstrates how a number of applications, both internal to an enterprise, and external at other sites, can be integrated using the SOA architecture paradigm to create one cohesive ordering system. In this sample application, Global Company is a retail store front for ordering electronic devices through a Web-based client application. Global Company sells electronics, such as MP3 players and televisions, to consumers using a Web-based client application. It utilizes the following Oracle SOA Suite components:

- JDeveloper
- Oracle BPEL Process Manager (including Human Workflow)
- ESB
- Oracle Business Rules
- Oracle Application Server

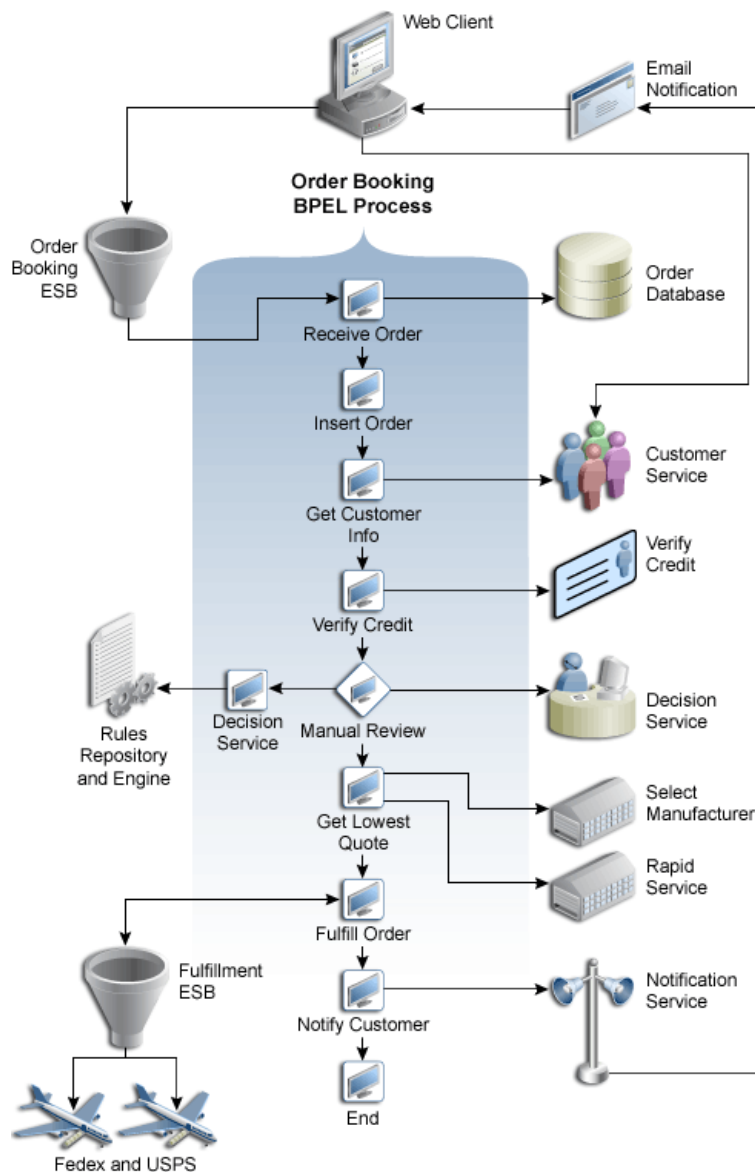
Specifically, the following applications comprise the SOA Order Booking application:

- CreditService: Validates customer credit information
- CustomerService: Represents an existing customer relationship management (CRM) application within the Global Company enterprise. CustomerService is implemented using EJB 3.0.
- DHLShipment: Ships the order using DHL. It is a SOAP service.
- FulfillmentESB: Routes order information to a shipping vendor for shipment. Uses ESB content-based routing to invoke the correct vendor, such as Fedex or USPS, depending on the price of the order. In [Chapter 4](#), you will utilize the DHLShipment service, so orders can be also be shipped by DHL. Also, demonstrates that messages can be routed using a variety of adapters for writing to a database, a file, and a JMS queue.

- OrderBookingESB: Routes order information from the Web client and invokes the Order Booking BPEL process (SOAOrderBooking).
- RapidService: Provides a price for a given order (to compete with the price from SelectManufacturer). This service is used to demonstrate invoking a synchronous process from BPEL.
- SelectManufacturer: Provides a price for a given order (to compete with the price from RapidService). This service is used to demonstrate invoking an asynchronous process from BPEL.
- SOADEMO-CLIENT: Web-based application that enables end users to register themselves as customers, and then order electronics online from Global Company.
- SOAOrderBooking: Processes the order and orchestrates all necessary services within the enterprise to complete the order.

At the center of the SOA Order Booking application is the SOAOrderBooking BPEL process. It orchestrates all the existing services in the enterprise for order fulfillment with the right supplier, based on the business rules in the process.

[Figure 1–2](#) shows an overview of the SOA Order Booking application, followed by a step-by-step description of the application flow.

**Figure 1–2 SOA Order Booking Application Overview**

When a new customer registers in the Web client, the Web client sends the customer's information to the internal customer service application (CustomerService), which then stores the customer information in a database. The authenticated customer can then browse products, add them to their online shopping cart, and place the order.

When a registered customer attempts to log on to the Web client, the customer service application (CustomerService) is invoked and provides authentication.

When a new order is placed, the following occurs:

1. The Web client sends a message to the Order Booking ESB project (OrderBookingESB). ESB then routes the message to any service that has registered an interest in these messages. In this case, it is only the Order Booking BPEL process (SOAOrderBooking) that gets notified, thereby invoking the BPEL process.
2. The BPEL process sets the order to pending, and writes the order to the order database tables.

The registration process sets the default customer status to Gold. The database administrator can apply one of the following statuses to registered customers:

- Platinum
- Gold
- Silver

3. The BPEL process calls the customer service (CustomerService) to retrieve customer ID, name, address, and credit card information.
4. The BPEL process then checks the identified customer against the credit service (CreditService) to verify the customer's credit card is valid. The credit service returns the relevant rating for the customer.

If credit is not approved, the process cancels the order.

If credit is approved, the process takes the order amount, customer status, and runs the Oracle Rules Engine to determine if the order requires approval by management.

5. The BPEL process uses a decision service (Oracle Rules Engine) to determine.

The decision service uses rules in the Oracle Rules Engine that state the following:

- If the customer has a Platinum status, then no manual approval is required.
- If the customer has a Gold status, then manual approval is only required for orders over \$1000.
- If the customer has a Silver status, then manual approval is required for all orders, regardless of the order amount.

These decisions could also have been implemented in the BPEL process, but a rules engine was used instead so that changes could be made at runtime without have to make a design-time change or re-deploy. Moreover, business analysts or non-technical, non-administrator type people can change the rules.

6. For those orders requiring manual approval, the BPEL process invokes the human workflow, which routes a message to a manager.
7. If the order is approved, it is sent to two suppliers for their price quotes.
  - The select manufacturer service (SelectManufacturer) tends to quote a lower price, because it is a direct manufacturer, but has slow turnaround times for a response. It sends the quote asynchronously.
  - The rapid manufacturer service (RapidService) tends to quote a higher price, because has an automated response service for price quotes. It sends the quote synchronously.
8. The BPEL process collects the quotes and selects the lowest quoted price as the supplier to which to award the order.
9. The BPEL process invokes the Fulfillment ESB project (FulfillmentESB), which in turn uses a message flow to complete the order as follows:
  - If the order is less than \$500, the order information is sent to USPS. Their system picks up orders from a directory so we use the file adapter to write the order to a file in the specified directory.
  - If the order is equal to or greater than \$500, the order information is sent to Fedex. The Fedex system waits for rows to appear in a table, so the database adapter is used to write the order to the specified table in their database.

- All orders, regardless of the shipping vendor, are stored in a temporary queue and uploaded to the fulfillment system in batch mode overnight. The JMS adapter is used to write the order information to the specified JMS queue.
10. Once the order is fulfilled, the BPEL process sets the order to complete, and starts a notification service, which sends the customer an email with the purchase order information.





---

# Setting Up the SOA Order Booking Application

This chapter describes installing the required prerequisites for the SOA Order Booking application and setting up the application itself.

This chapter includes the following sections:

- [Section 2.1, "Checking for and Installing Prerequisites"](#)
- [Section 2.2, "Tasks for Setting Up the SOA Order Booking Application"](#)

## 2.1 Checking for and Installing Prerequisites

This section takes you through prerequisite software installation tasks. If you have a prerequisite already installed, then you can skip to the next task.

- [Task 1: Install an Oracle Database](#)
- [Task 2: Install JDeveloper Studio 10.1.3.1.0](#)
- [Task 3: Install Oracle SOA Suite 10.1.3.1.0](#)
- [Task 4: Install the SOA Order Booking Application](#)

### 2.1.1 Task 1: Install an Oracle Database

The SOA Order Booking applications requires a database for its data. The SQL scripts were written for an Oracle database, so you will need some version of an Oracle RDBMS, such as 9i, 10g, or XE. The scripts will not install into Oracle Lite. If you wish to use Oracle Lite or some other database, then you will need to modify the database scripts accordingly.

If you already have an Oracle database installed, then skip to [Task 2: Install JDeveloper Studio 10.1.3.1.0](#). If not, then you can download an Oracle database from:

<http://www.oracle.com/technology/index.html>

Specifically, Oracle Express Edition (XE) is a small footprint database and is recommended. You can download it from:

<http://www.oracle.com/technology/products/xe/index.html>

### 2.1.2 Task 2: Install JDeveloper Studio 10.1.3.1.0

You need JDeveloper to install certain parts of the SOA Order Booking application and use the design-time environment. Because the version of Oracle SOA Suite is 10.1.3.1.0, you need the corresponding version of JDeveloper.

For this quick start, you need JDeveloper 10.1.3.1.0 Studio edition. If you already have it installed, then skip to step [Task 3: Install Oracle SOA Suite 10.1.3.1.0](#). Before you do, ensure you have 10.1.3.1.0 (not to be confused with 10.1.3.0.0) and that it is the Studio edition, not the J2EE or Java edition. You can verify these details in JDeveloper from the **Help > About** menu option.

You can download JDeveloper Studio 10.1.3.1.0 from:

<http://www.oracle.com/technology/products/jdev/index.html>

### 2.1.3 Task 3: Install Oracle SOA Suite 10.1.3.1.0

If you already have Oracle SOA Suite 10.1.3.1.0 installed, then skip to [Section 2.2, "Tasks for Setting Up the SOA Order Booking Application"](#).

This quick start assumes the following:

- The **AS Instance Name** is `soademo`.
- The **AS Administrator Password** is `welcome1`.
- The default port is 8888.

To download Oracle SOA Suite 10.1.3.1.0:

1. From the following page, select the **Download and install Oracle SOA Suite 10.1.3.1** link:

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

2. On the download page, select **Oracle SOA Suite 10.1.3.1**.

When you install Oracle SOA Suite, ensure you do the following:

- Select **Basic Install**.
- Set the **AS Instance Name** to `soademo`.
- Set the **AS Administrator Password** to `welcome1`.

The Oracle SOA Suite installation sets the `ORACLE_HOME` environment variable for your computer. In some cases, this setting can cause a conflict with the Oracle Database. Specifically you may get errors from your Oracle Net Listener. If this is the case, then reset the `ORACLE_HOME` environment variable to your database location. On Windows, you may need to restart your PC.

### 2.1.4 Task 4: Install the SOA Order Booking Application

Download the SOA Order Booking application ZIP file (`soademo_101310_prod.zip`) and extract it to a working directory.

To access the ZIP file:

1. From the following page, select the **Download and install Oracle SOA Suite 10.1.3.1** link:

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

2. On the download page, select **SOA Order Booking Demo application**.

## 2.2 Tasks for Setting Up the SOA Order Booking Application

To use the SOA Order Booking application, you must complete the following tasks:

- [Task 1: Install Oracle Database Schema](#)
- [Task 2: Configure Oracle SOA Suite](#)
- [Task 3: Note the Name of the OC4J Instance Running the Oracle BPEL Server](#)
- [Task 4: Note the Important Port Numbers](#)
- [Task 5: Create Connections in JDeveloper](#)
- [Task 6: Install and Deploy the SOA Order Booking Application](#)
- [Task 7: Configure the ESB Port](#)
- [Task 8: Familiarize Yourself with the Schema](#)

This guide uses the following references:

- *JDEV\_HOME* refers to the location of the your JDeveloper installation
- *ORACLE\_HOME* refers to the location of the your Oracle SOA Suite installation.
- *DEMO\_HOME* refers to the location where you unzipped the SOA Order Booking ZIP file.

### 2.2.1 Task 1: Install Oracle Database Schema

The SOA Order Booking application tables are contained within one schema.

To install the schema:

1. Connect to your database and create a user named `soademo`.

For example:

```
SQL> CREATE USER soademo IDENTIFIED BY soademo;  
SQL> GRANT CONNECT, RESOURCE TO soademo;
```

2. Connect to the database as user `soademo`:

```
SQL> CONNECT soademo/soademo
```

3. Run the following scripts from the *DEMO\_HOME* directory:

- `createSchemaObjects.sql`
- `populateSchemaTables.sql`

---

**Note:** Ignore any issues the first time you run these scripts when objects are dropped. These errors are fine, as the objects have not been created yet.

---

### 2.2.2 Task 2: Configure Oracle SOA Suite

Once you install Oracle SOA Suite, you need to create a connection pool, data source, and database adapter for the SOA Order Booking application.

To configure the container:

1. Point your browser to the Welcome to Oracle SOA Suite (10.1.3.1.0) page:

```
http://localhost:8888
```

8888 is the default HTTP port. Use the port number that the installer selected for your environment. If port 8888 does not work for you and you are not sure what your port number is, use a text editor and open:

`ORACLE_HOME\install\readme.txt`

Look for the line **Access the Oracle SOA Suite 10g Application Server Welcome Page via the following URL and the next line will show you the URL for your welcome page.**

2. From the right-hand side **Manage Your SOA Suite** portlet, select the **Application Server Control** link.

The Oracle Enterprise Manager 10g Application Server Control Console displays the administrator logon dialog box.

3. Enter `oc4jadmin` as the username and enter the password created during installation, `welcome1`.
4. In the **Members** section, expand the **home OC4J** container.
5. Ensure the applications all have a status of **Up**.
6. Create a connection pool and JDBC data source for the OC4J instance.

A data source enables you to retrieve a connection to a database server. A managed data source uses a connection pool to efficiently manage connections. You must define one connection pool and its connection factory.

- a. Select the **home OC4J** link.

The OC4J: home page appears.

- b. Click the **Administration** tab.

- c. In **Services > JDBC Resources**, click the **Go To Task** icon.

The JDBC Resources page appears.

- d. In the **Connection Pools** section, click the **Create** button.

The Create Connection Pool - Application page appears.

- e. Accept defaults, and click **Continue**.

The Create Connection Pool page appears.

- f. Enter the following values, leaving the defaults for the other fields:

Element	Value
Name	soademo_pool
JDBC URL	URL for your database. For example: <code>jdbc:oracle:thin:@mydb.company.com:1521:orcl</code>
Username	soademo
Password	soademo or customized password entered when creating the soademo user in the database

- g. Click **Test Connection**.

The Test Connection page appears.

- h. Click **Test**.

The JDBC Resources page updates with a successful connection confirmation message. If you get an error message, check the URL and credentials to ensure you entered the right information.

- i. Click **Finish**.
- j. Back in the JDBC Resources page, under **Data Sources**, click **Create**.  
The Create Data Source - Application & Type page appears.
- k. Accept the defaults, and click **Continue**.  
The Create Data Source - Managed Data Source page appears.
- l. Enter the following values, leaving the defaults for the other fields:

Element	Value
Name	soademoDS
JNDI Location	jdbc/soademoDS
Connection Pool	soademo_pool

- m. Click **Finish**.
  - n. Back on the **Data Sources** section, locate the soademoDS data source, and click the **Test Connection** icon.  
The Test Connection page appears.
  - o. Click **Test**.  
The JDBC Resources page updates with a successful connection confirmation message.
7. Create a database adapter connection factory:
- a. At the top of the JDBC Resources page, click the **OC4J:home** breadcrumb link.  
The OC4J: home page appears.
  - b. Click the **Applications** tab.
  - c. In the table of applications, click the **default** link.  
The Application: default page appears.
  - d. In the **Modules** section, click the **DbAdapter** link.  
The Resource Adapter: Database Adapter page appears.
  - e. Click the **Connection Factories** tab.
  - f. At the top of the page, click the **Create** button.

---

**Note:** Do not click the **Create** button in the **Shared Connection Pools** section.

---

The Create Connection Factory: Select Interface page appears.

- g. Accept the default interface, and click **Continue**.  
The Create Connection Factory page appears.
- h. In the **JNDI Location** field, enter `eis/DB/soademo`.
- i. In the **Configuration Properties** section, for **xADDataSourceName**, enter `jdbc/soademoDS`.
- j. Leave the defaults for the other fields.

- k. Click **Finish**.
- l. Leave the Application Server Control Console running, as you need it for the next task.

### 2.2.3 Task 3: Note the Name of the OC4J Instance Running the Oracle BPEL Server

In several upcoming procedures you will be promoted to specify an OC4J instance name. If you did the recommended Basic Install, then one instance is installed. By default, the instance name is `home`. If you did an Advanced Install, then two instances are installed. By default, they are `home` and `oc4j_soa`. The `oc4j_soa` instance runs the Oracle BPEL Server, which is the one you must specify.

To find the OC4J instance name:

1. From Application Server Control Console, click the **Cluster Topology** breadcrumb link.
2. In the **Members** section, note the instance names.
3. If you did an Advanced Install, expand `oc4j_soa` and verify that `orabpel` is one of the services.

### 2.2.4 Task 4: Note the Important Port Numbers

The SOA Order Booking Application has some configuration files that use the default port numbers. Your installation may not be using the default port numbers and may need to be reconfigured. This section explains how to find the port numbers being used in your environment.

To find the port numbers:

1. In the Cluster Topology page of the Application Server Control Console, in the **Administration** section, select **Runtime Ports**.

The Runtime Ports page shows the ports being used by your installation.

Name	Host	Port In Use	Port Range	Configure Port
<b>Expand All   Collapse All</b>				
▼ All Application Servers				
▼ soademo.dsteiner-pc.us.oracle.com	dsteiner-pc			
▼ OPMN				
Local		6100	6100	
Remote		6200	6200	
Request		6003	6003	
▼ OC4J : home				
JMS		12601	12601-12700	
HTTP		8888		
RMIS		12701	12701-12800	
RMI		12401	12401-12500	

These are the important ones to note for the SOA Order Booking Application installation:

- OPMN Request (defaults to 6003)
- OC4J:home HTTP (defaults to 8888)

- OC4J:home RMI (defaults to 12401)

If your ports are the same as the preceding defaults, then you don't need to worry since the configuration files already use those defaults. If one or more of your ports are different, then note them down for now. In some later installation steps you will be asked to refer to these port numbers.

## 2.2.5 Task 5: Create Connections in JDeveloper

Before you can use JDeveloper to deploy the application to the server, JDeveloper needs to know how to connect to the server. This includes the application server and the database.

To create these connections in JDeveloper:

1. Start JDeveloper by running `JDEV_HOME\jdeveloper.exe`.
2. Create a connection to the database using the `SOADEMO` schema:
  - a. Click the **Connections** tab, or if it not currently displayed, choose **View > Connection Navigator**.
  - b. Double-click the **Database** folder to open the Create Database Connection Wizard.
  - c. Complete the wizard, ensuring you complete the following:
    - On the Step 1 of 4: Type 1 page, enter `soademo` for the **Connection Name**.
    - On the Step 2 of 4: Authentication page, enter `soademo` for the **Username** and **Password** fields, and click **Deploy Password**.
    - Enter the appropriate values for where you installed the schema.
3. Create the application server connection:
  - a. Double-click **Application Server** to open the Create Application Server Connection wizard.
  - b. On the Step 1 of 4: Type page, perform the following and then click **Next**.

Element	Action
Connection Name	Enter <code>OrderBookingAS</code> .
Connection Type	Select <b>Oracle Application Server 10g 10.1.3</b> .

- c. On the Step 2 of 4: Authentication page, perform the following and then click **Next**.

Element	Action
User	Enter <code>oc4j_admin</code> .
Password	Enter <code>welcome1</code> .
Deploy Password	Click this checkbox.

- d. On the Step 3 of 4: Connection page, perform the following and then click **Next**.

Element	Action
Host Name	Enter the host name in which you installed Oracle SOA Suite.
OPMN	Enter the OPMN request port. The default is 6003.  <b>See Also:</b> <a href="#">"Task 4: Note the Important Port Numbers"</a> on page 2-6 to find out what the port number is for your environment
Group/OC4J Instance Name	Enter home.

- e. On the Step 4 of 4: Test page, click **Test the Connection**.  
If the test does not succeed, use the **Back** button to verify and change values, if needed.
  - f. Click **Finish**.
4. Create a connection to the Oracle BPEL Process Manager and Enterprise Service Bus integration server.  
  
Creating this connection enables you to deploy business processes to Oracle BPEL Process Manager and services to the Enterprise Service Bus.
    - a. Double-click **Integration Server** to open the Create Integration Server Connection Wizard.
    - b. On Step 1 of 3: Name, enter `OrderBookingIS` in the **Name** field.
    - c. On Step 2 of 3: Connection, perform the following and then click **Next**.

Element	Action
Application Server	Select <code>OrderBookingAS</code> , which is the name of the application server you created in Step 3.
Host Name	Enter the host name in which you installed Oracle SOA Suite
Port	Enter the port for the Oracle HTTP Server installed with the SOA Suite. The default is 8888.  <b>See Also:</b> <a href="#">"Task 4: Note the Important Port Numbers"</a> on page 2-6 to find out what the port number is for your environment

- d. On Step 3 of 3, Test, click **Test the Connection**.  
If the test does not succeed, use the **Back** button to verify and change values, if needed.
- e. Click **Finish**.

## 2.2.6 Task 6: Install and Deploy the SOA Order Booking Application

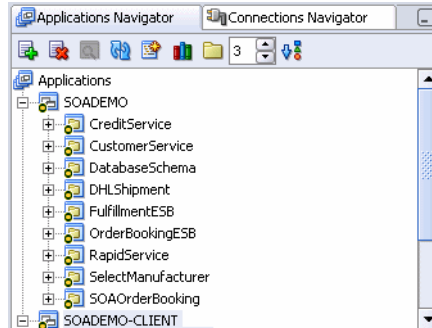
To deploy the individual SOA Order Booking applications to the container using JDeveloper:

1. Open the applications in JDeveloper:
  - a. Choose **File > Open**.
  - b. Navigate to the `DEMO_HOME` directory and open the `SOADEMO.jws` file.
  - c. Choose **File > Open** again.



- d. Navigate to the `DEMO_HOME/SOADEMO-CLIENT` directory and open the `SOADEMO-CLIENT.jws` file.

Both applications appear in the Application Navigator. Review [Section 1.3, "Introduction to SOA Order Booking Application"](#) on page 1-5 for a description of these applications.

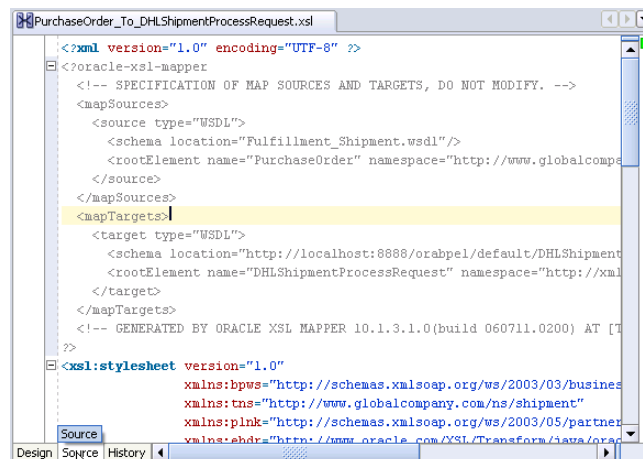


## 2. Deploy the FulfillmentESB project:

- a. Expand the **SOADEMO** application.
- b. If you are using port 8888 as your HTTP port, then proceed to Step g. If you are not using the default HTTP port, expand the **Resources** node and double-click the `PurchaseOrder_To_DHLShipmentProcessRequest.xml` file to open it.

Disregard any errors that appear.

- c. In the `PurchaseOrder_To_DHLShipmentProcessRequest.xml` page, replace all instances of 8888 with your port number. See ["Task 4: Note the Important Port Numbers"](#) on page 2-6 to find out what the port number is for your environment.
- d. Click the **Source** sub-tab, located at the bottom of the `PurchaseOrder_To_DHLShipmentProcessRequest.xml` page, to switch to the **Source** view.



- e. Replace all instances of 8888 with your port number. See ["Task 4: Note the Important Port Numbers"](#) on page 2-6 to find out what the port number is for your environment.
- f. Choose **File > Save** to save your work.
- g. Right-click **FulfillmentESB**, and select **Register with ESB > OrderBookingIS**.

- h. Click **OK** in the ESB Registration Summary dialog.
3. Deploy the SelectManufacturer BPEL process using Ant:
  - a. Expand **SelectManufacturer > Resources**.
  - b. Open the `build.properties` file and verify values are set as follows.
 

`admin.user`: Set to `oc4jadmin`.

`admin_password`: Set to `welcome1`.

`hostname`: Set to host name in which you installed Oracle SOA Suite. You can also use `localhost` or `127.0.0.1`.

`http.port`: Set to the Oracle HTTP Server. The default for this port is 8888.

`rmi.port`: Set to 12401. See ["Task 4: Note the Important Port Numbers"](#) on page 2-6 to find out what the port number is for your environment.

`oc4jinstancename`: Set to the name to the OC4J instance running the Oracle BPEL Server. For a Basic Install, the default is `home`. See ["Task 3: Note the Name of the OC4J Instance Running the Oracle BPEL Server"](#) on page 2-6 to find out what the instance name is for your environment.

`asinstancename`: You do not need to set this property.

`opmn.requestport`: Set to the OPMN request port. The default for this port is 6003. See ["Task 4: Note the Important Port Numbers"](#) on page 2-6 to find out what the port number is for your environment.

`platform`: Set to `ias_10g`.
  - c. In the Application Navigator, right-click `build.xml`, and select **Run Ant**.  
The Run Ant dialog appears.
  - d. Click the **Properties** tab. If there are any properties in the **Properties** section, located in the upper part of the dialog, then remove them.
  - e. In the **Property Files** section, located in the lower part of the window, click **Add**.  
The Add Ant Property File dialog appears.
  - f. Select the `build.properties` file and then click **Open**.
  - g. In the Run Ant dialog, click **OK**.  
This action starts the deployment process, which may take anywhere from 30 to 60 seconds. You can monitor the progress in the **Messages** pane, in the **Apache Ant - Log** tab, in the **Apache Ant** sub-tab. You will know the deployment is complete when you see the text `BUILD SUCCESSFUL`.
4. Deploy the CreditService application:
  - a. Expand **CreditService > Resources**.
  - b. Right-click the `WebServices.deploy` file, and select **Deploy to > OrderBookingAS**.
  - c. When the Configure Application dialog appears, click **OK**.  
When deployment completes, the Deployment - Log displays a Deployment finished message.
5. Deploy the CustomerService application:
  - a. Expand **CustomerService > Resources**.

- b. Right-click the `CustomerService.deploy` file, and select **Deploy to > OrderBookingAS**.
  - c. When the Configure Application dialog appears, click **OK**.  
When deployment completes, the Deployment - Log displays a Deployment finished message.
6. Deploy the RapidService application:
  - a. Expand **RapidService > Resources**.
  - b. Right-click the `WebServices.deploy` file, and select **Deploy to > OrderBookingAS**.
  - c. When the Configure Application dialog appears, click **OK**.  
When deployment completes, the Deployment - Log displays a Deployment finished message.
7. Deploy the SOAOrderBooking BPEL process using Ant:
  - a. Expand **SOAOrderBooking > Resources**.
  - b. Open the `build.properties` file and verify values are set as follows.  
`admin.user`: Set to `oc4jadmin`.  
`admin_password`: Set to `welcome1`.  
`hostname`: Set to host name in which you installed Oracle SOA Suite. You can also use `localhost` or `127.0.0.1`.  
`http.port`: Set to the Oracle HTTP Server. The default for this port is 8888.  
`rmi.port`: Set to 12401. See ["Task 4: Note the Important Port Numbers"](#) on page 2-6 to find out what the port number is for your environment.  
`oc4jinstancename`: Set to the name to the OC4J instance running the Oracle BPEL Server. For a Basic Install, the default is `home`. See ["Task 3: Note the Name of the OC4J Instance Running the Oracle BPEL Server"](#) on page 2-6 to find out what the instance name is for your environment.  
`asinstancename`: You do not need to set this property.  
`opmn.requestport`: Set to the OPMN request port. The default for this port is 6003. See ["Task 4: Note the Important Port Numbers"](#) on page 2-6 to find out what the port number is for your environment.  
`platform`: Set to `ias_10g`.
  - c. If you are using port 8888 as your HTTP port, then proceed to Step k. If you are not using the default HTTP port, expand the **Integration Content** node and double-click the `bpe1.xml` file to open it.
  - d. In the `bpe1.xml` page, replace all instances of 8888 with your port number. See ["Task 4: Note the Important Port Numbers"](#) on page 2-6 to find out what the port number is for your environment.
  - e. Choose **File > Save** to save your work, and then restart JDeveloper.
  - f. Under the **Integration Content** node, double-click the `CreditValidatingService.wsdl` file to open it.
  - g. Click the **Source** sub-tab to switch to the **Source** view.

- h. Replace all instances of 8888 with your port number. See ["Task 4: Note the Important Port Numbers"](#) on page 2-6 to find out what the port number is for your environment.
  - i. Choose **File > Save** to save your work.
  - j. Repeat Steps f through i for `CustomerSvc.wsdl` and `RapidService.wsdl`.
  - k. In the Application Navigator, right-click `build.xml`, and select **Run Ant**.  
The Run Ant dialog appears.
  - l. Click the **Properties** tab. If there are any properties in the **Properties** section, located in the upper part of the dialog, then remove them.
  - m. In the **Property Files** section, located in the lower part of the window, click **Add**.  
The Add Ant Property File dialog appears.
  - n. Select the `build.properties` file and then click **Open**.
  - o. In the Run Ant dialog, click **OK**.  
This action starts the deployment process, which may take anywhere from 30 to 60 seconds. You can monitor the progress in the **Messages** pane, in the **Apache Ant - Log** tab, in the **Apache Ant** sub-tab. You will know the deployment is complete when you see the text `BUILD SUCCESSFUL`.
8. Deploy the DHLShipment service using Ant:
- a. Expand **DHLShipment > Resources**.
  - b. Open the `build.properties` file and verify values are set as follows.  
`admin.user`: Set to `oc4jadmin`.  
`admin_password`: Set to `welcome1`.  
`hostname`: Set to host name in which you installed Oracle SOA Suite. You can also use `localhost` or `127.0.0.1`.  
`http.port`: Set to the Oracle HTTP Server. The default for this port is 8888.  
`rmi.port`: Set to 12401. See ["Task 4: Note the Important Port Numbers"](#) on page 2-6 to find out what the port number is for your environment.  
`oc4jinstancename`: Set to the name to the OC4J instance running the Oracle BPEL Server. For a Basic Install, the default is `home`. See ["Task 3: Note the Name of the OC4J Instance Running the Oracle BPEL Server"](#) on page 2-6 to find out what the instance name is for your environment.  
`asinstancename`: You do not need to set this property.  
`opmn.requestport`: Set to the OPMN request port. The default for this port is 6003. See ["Task 4: Note the Important Port Numbers"](#) on page 2-6 to find out what the port number is for your environment.  
`platform`: Set to `ias_10g`.
  - c. In the Application Navigator, right-click `build.xml`, and select **Run Ant**.  
The Run Ant dialog appears.
  - d. Click the **Properties** tab. If there are any properties in the **Properties** section, located in the upper part of the dialog, then remove them.

- e. In the **Property Files** section, located in the lower part of the window, click **Add**.  
The Add Ant Property File dialog appears.
  - f. Select the `build.properties` file and then click **Open**.
  - g. In the Run Ant dialog, click **OK**.  
This action starts the deployment process, which may take anywhere from 30 to 60 seconds. You can monitor the progress in the **Messages** pane, in the **Apache Ant - Log** tab, in the **Apache Ant** sub-tab. You will know the deployment is complete when you see the text `BUILD SUCCESSFUL`.
9. Deploy the OrderBookingESB project:
    - a. If you are using port 8888 as your HTTP port, then skip this step and proceed to Step c. Using a text editor, open the following file and change all references of 8888 with your port number. See "[Task 4: Note the Important Port Numbers](#)" on page 2-6 to find out the HTTP port for your environment.  
`DEMO_HOME\OrderBookingESB\OrderBooking.esbsys`  
`DEMO_HOME\OrderBookingESB\OrderBooking_OrderBookingProcess.esbsvc`
    - b. Choose **File > Save** to save your work.
    - c. Right-click **OrderBookingESB**, and select **Register with ESB > OrderBookingIS**.
    - d. Click **OK** in the Summary dialog.
  10. Deploy the SOADEMO-CLIENT Web client:
    - a. If you are using port 8888 as your HTTP port, then skip this step and proceed to Step c. Using a text editor, open the following files and change all references of 8888 with your port number. See "[Task 4: Note the Important Port Numbers](#)" on page 2-6 to find out the HTTP port for your environment.  
`DEMO_HOME\SOADEMO-CLIENT\CustomerService\src\oracle\soademo\view\services\runtime\CustomerServiceSoapHttp_Stub.java`  
`DEMO_HOME\SOADEMO-CLIENT\OrderService\src\oracle\soademo\view\services\runtime\__soap_initiate_ppt_Stub.java`
    - b. Choose **File > Save** to save your work.
    - c. Expand **SOADEMO-CLIENT > Assembly > Application Sources**.
    - d. Right-click the `SOADEMO.deploy` file, and select **Deploy to > OrderBookingAS**.
    - e. When the Configure Application dialog appears, click **OK**.

### 2.2.7 Task 7: Configure the ESB Port

If your HTTP port is 8888, which is the default, then skip this task and proceed to "[Task 8: Familiarize Yourself with the Schema](#)" on page 2-14. Otherwise, follow these steps:

1. Point your browser to the Welcome to Oracle SOA Suite (10.1.3.1.0) page:  
`http://localhost:http_port`

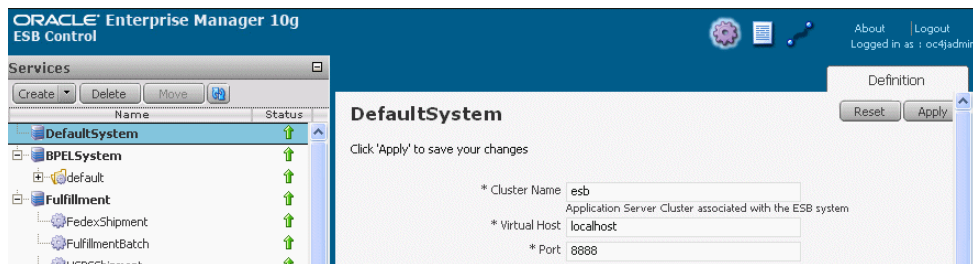
**See Also:** ["Task 4: Note the Important Port Numbers"](#) on page 2-6 to find out what the port number is for your environment

- From the right-hand side **Manage Your SOA Suite** portlet, select the **ESB Control** link.

The Oracle Enterprise Manager 10g ESB Control displays.

- In the **Services** pane, select the **DefaultSystem** group.

Configuration information appears in the right-hand pane.



- In the **Port** field, change the HTTP port to the HTTP port number used in your environment.

**See Also:** ["Task 4: Note the Important Port Numbers"](#) on page 2-6 to find out what the port number is for your environment

- Click **Apply**.
- Repeat Steps 3 through 5 for other groups in the **Services** pane: **BPELSystem**, **OrderBooking**, and **Fulfillment**.
- Close the Oracle Enterprise Manager 10g ESB Control.

## 2.2.8 Task 8: Familiarize Yourself with the Schema

The SOADEMO schema consists of the following tables:

For the SOADEMO-CLIENT Web client:

- **PRODUCT:** Contains the products offered for sale on the Web site.

For the SOAOrderBooking BPEL flow:

- **ORDERS:** Contains information regarding an order
- **ITEMS:** Contains information regarding the items contained in an order
- **SSN:** Contains credit score based on the customer ID.

For the CustomerService application:

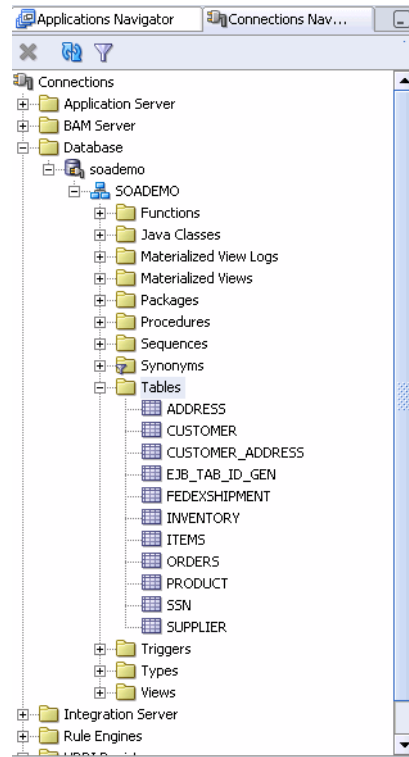
- **CUSTOMER:** Contains customer information, such as name, email, and credit card number
- **ADDRESS:** Contains address information
- **CUSTOMER\_ADDRESS:** Maps a customer to an address. This join table allows a customer to have more than one address, and an address to belong to more than one customer.
- **EJB\_TAB\_ID\_GEN:** Contains the IDs used in the CustomerService application

For the FulfillmentESB:

- **FEDEXSHIPMENT**: Contains shipping details for an order. In reality, this table would exist at FedEx.

To view the schema for the SOA Order Booking application using JDeveloper:

1. Click the **Connections Navigator** tab.
2. Expand **Database > soademo > SOADEMO > Tables**.



3. Double-click **CUSTOMER**, and then click the **Data** tab to view the customer information.

CUSTOMER									
	CUSTID	FNAME	LNAME	PHONENUM...	EMAIL	PASSWORD	CREDITCAR...	CREDITCAR...	ST
1	10	steve	king	18008888000	sking@soad...	welcome1	AMEX	12345678	Gold
2	11	john	chen	18008888000	jchen@soad...	welcome1	Visa	12345670	Platin
3	12	guy	himuro	18008888000	ghimuro@so...	welcome1	AMEX	12345678	Silver

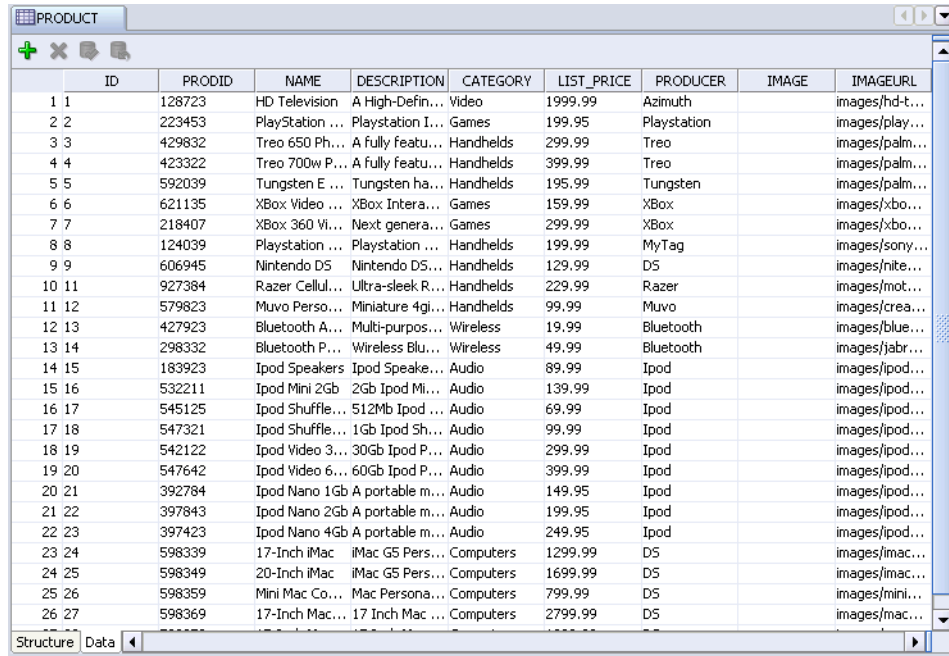
The **CUSTOMER** table shows data for preregistered customers, as described in [Table 2-1](#).

**Table 2-1 Pre-Registered Customers in the SOA Order Booking Application**

Email	Password	Customer Status	Notes
sking@soademo.org	welcome1	Gold	Requires manual approval for orders over \$1,000.
jchen@soademo.org	welcome1	Platinum	Has an invalid credit card number.
ghimuro@soademo.org	welcome1	Silver	Requires manual approval for all orders.

In [Chapter 3](#), you will place an order through the Web client as `sking`, and monitor that order process.

4. Double-click **PRODUCT**, and then click the **Data** tab to view the product inventory.



ID	PRODID	NAME	DESCRIPTION	CATEGORY	LIST_PRICE	PRODUCER	IMAGE	IMAGEURL
1 1	128723	HD Television	A High-Defin...	Video	1999.99	Azimuth		images/hd-t...
2 2	223453	PlayStation ...	Playstation I...	Games	199.95	Playstation		images/play...
3 3	429832	Treo 650 Ph...	A fully featu...	Handhelds	299.99	Treo		images/palm...
4 4	423322	Treo 700w P...	A fully featu...	Handhelds	399.99	Treo		images/palm...
5 5	592039	Tungsten E ...	Tungsten ha...	Handhelds	195.99	Tungsten		images/palm...
6 6	621135	XBox Video ...	XBox Intera...	Games	159.99	XBox		images/xbo...
7 7	218407	XBox 360 Vi...	Next genera...	Games	299.99	XBox		images/xbo...
8 8	124039	Playstation ...	Playstation ...	Handhelds	199.99	MyTag		images/sony...
9 9	606945	Nintendo DS	Nintendo DS...	Handhelds	129.99	DS		images/nite...
10 11	927384	Razer Cellul...	Ultra-sleek R...	Handhelds	229.99	Razer		images/mot...
11 12	579823	Muvo Perso...	Miniature 4gi...	Handhelds	99.99	Muvo		images/crea...
12 13	427923	Bluetooth A...	Multi-purpos...	Wireless	19.99	Bluetooth		images/blue...
13 14	298332	Bluetooth P...	Wireless Blu...	Wireless	49.99	Bluetooth		images/jabr...
14 15	183923	Ipod Speakers	Ipod Speake...	Audio	89.99	Ipod		images/ipod...
15 16	532211	Ipod Mini 2Gb	2Gb Ipod Mi...	Audio	139.99	Ipod		images/ipod...
16 17	545125	Ipod Shuffle...	512Mb Ipod ...	Audio	69.99	Ipod		images/ipod...
17 18	547321	Ipod Shuffle...	1Gb Ipod Sh...	Audio	99.99	Ipod		images/ipod...
18 19	542122	Ipod Video 3...	30Gb Ipod P...	Audio	299.99	Ipod		images/ipod...
19 20	547642	Ipod Video 6...	60Gb Ipod P...	Audio	399.99	Ipod		images/ipod...
20 21	392784	Ipod Nano 1Gb	A portable m...	Audio	149.95	Ipod		images/ipod...
21 22	397843	Ipod Nano 2Gb	A portable m...	Audio	199.95	Ipod		images/ipod...
22 23	397423	Ipod Nano 4Gb	A portable m...	Audio	249.95	Ipod		images/ipod...
23 24	598339	17-Inch iMac	iMac G5 Pers...	Computers	1299.99	DS		images/imac...
24 25	598349	20-Inch iMac	iMac G5 Pers...	Computers	1699.99	DS		images/imac...
25 26	598359	Mini Mac Co...	Mac Persona...	Computers	799.99	DS		images/mini...
26 27	598369	17-Inch Mac...	17 Inch Mac ...	Computers	2799.99	DS		images/mac...

The **PRODUCT** table shows the electronic devices available through the Web-client interface.

5. View any of the other tables you are interested in.



---

## Running and Monitoring the SOA Order Booking Application

This chapter describes how to use the SOA Order Booking application. It explains three different order scenarios for the Web client and how to monitor orders process through the SOA Order Booking business flow.

This chapter includes the following sections:

- [Section 3.1, "Placing an Order for Over \\$1000"](#)
- [Section 3.2, "Increasing Approval Order Amount from \\$1000 to \\$2000 and Resubmitting Order"](#)

### 3.1 Placing an Order for Over \$1000

In this scenario, you will place an order as preregistered user `sking` for over \$1000 and watch it process. As you will see, by default, orders for over \$1000 for Gold status customers like `sking` require manual approval. The approval is routed to `jcooper`, an internal Global Company supervisor with approval privileges.

In this scenario, you will perform the following tasks:

- [Task 1: Start Oracle BPEL Worklist Application](#)
- [Task 2: Place Order in Web Client](#)
- [Task 3: View the Order in the Oracle Enterprise Manager 10g BPEL Control Oracle BPEL Control](#)
- [Task 4: Use the Oracle BPEL Worklist Application to Approve Order](#)
- [Task 5: View Approval in the Oracle BPEL Control](#)
- [Task 6: View Instances in Oracle Enterprise Manager 10g ESB Control](#)

#### 3.1.1 Task 1: Start Oracle BPEL Worklist Application

For orders over \$1000, the Order Booking BPEL process (`SOAOrderBooking`) invokes the human workflow, which routes a message to the manager of the person who entered the order. In this case, the approval is routed to `jcooper` through the Oracle BPEL Worklist Application.

To start the Worklist Application:

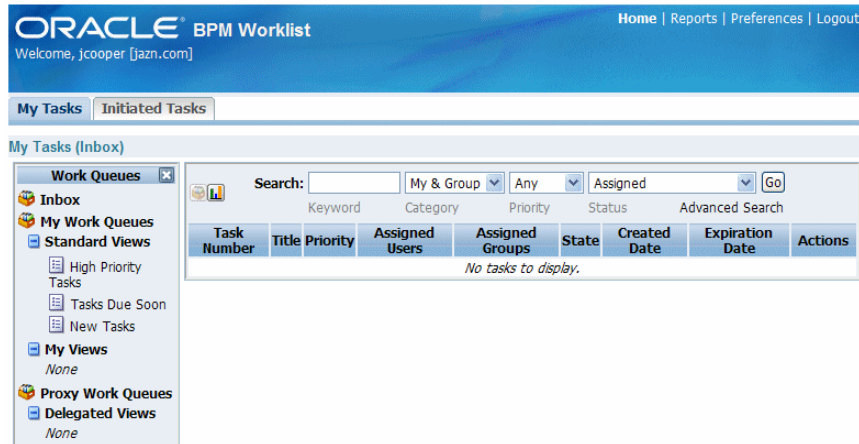
1. Select **Start > All Programs > Oracle - Oracle - soademo > Oracle BPEL Process Manager > Worklist Application**.

- When prompted, enter `jcooper` in the **Username** field and `welcome1` in the **Password** field.

`jcooper` is a defined Oracle BPEL Process Manager administrator.

- Click **Login**.

The **My Tasks** tab shows that no worklist tasks are currently assigned.



- Keep the Worklist Application running, as you will need it for future tasks.

### 3.1.2 Task 2: Place Order in Web Client

The ordering process begins in the Web client application where a user shops for and orders products. The Web application kicks off the Order Booking ESB (OrderBookingESB) flow, which in turn invokes SOAOrderBooking process flow. The BPEL flow handles the actual ordering process.

To place a new order, run the Web client:

- Point to the following URL:

`http://localhost:8888/soademo`

The Application Login page for the Web client appears.

- Enter `sking@soademo.org` in the **Email** field and `welcome1` in the **Password** field.

`sking` is a preregistered user with a Gold status. Customers with a Gold status require manual approval for orders over \$1000.

- Click **Login**.

The welcome page appears.

**Global Company**  
Your Online Shopping MegaStore

Help Logout

Home Browse Items Orders Cart

Userid: sking@soademo.org

**SOA Release Demo Web Client**

What do you want to do?

[Browse products and create a new order](#)  
[View current orders](#)

- Click the **Browse products and create a new order** link.

The Browse and Select Items page appears.

- Click **Next 10**, and select **Ipod Mini 2 Gb**.

**Browse and Select Items**

Select	name	prodid	description	category	listPrice	producer
<input type="radio"/>	Muvo Personal MP3 Player	579823	Miniature 4gig Sterero MP3 Player	Handhelds	\$99.00	Muvo
<input type="radio"/>	Bluetooth Adaptor	427923	Multi-purpose Mini BlueTooth Adaptor	Wireless	\$19.00	Bluetooth
<input type="radio"/>	Bluetooth Phone Headset	298332	Wireless Bluetooth Phone Headset	Wireless	\$49.00	Bluetooth
<input type="radio"/>	Ipod Speakers	183923	Ipod Speaker Addon System	Audio	\$89.00	Ipod
<input checked="" type="radio"/>	Ipod Mini 2Gb	532211	2Gb Ipod Mini Personal Audio Player	Audio	\$139.00	Ipod
<input type="radio"/>	Ipod Shuffle 512Mb	545125	512Mb Ipod Shuffle Personal Audio Player	Audio	\$69.00	Ipod
<input type="radio"/>	Ipod Shuffle 1Gb	547321	1Gb Ipod Shuffle Personal Audio Player	Audio	\$99.00	Ipod
<input type="radio"/>	Ipod Video 30Gb	542122	30Gb Ipod Personal Audio and Video Player	Audio	\$299.00	Ipod
<input type="radio"/>	Ipod Video 60Gb	547642	60Gb Ipod Personal Audio and Video Player	Audio	\$399.00	Ipod
<input type="radio"/>	Ipod Nano 1Gb	392784	A portable media entertainment device.	Audio	\$149.00	Ipod

- Click **View Details**.

The Item Details page appears. It displays detailed information about the product, and enables the user to select a quantity to add to their cart.

**Item Details**

Item Details

name	Ipod Mini 2Gb
description	2Gb Ipod Mini Personal Audio Player
category	Audio
listPrice	\$139.00
prodid	532211
producer	Ipod

[Continue Shopping](#)

Add to cart?

Quantity?

[Add to cart](#)

[Go to Shopping Cart](#)

- From the **Quantity** list, select 10, and click **Add to cart**.

- Click **Go to Shopping Cart**.

The Shopping Cart Contents page appears.

**Shopping Cart Contents**

Product Name	Part No.	Category	Price	Quantity
Ipod Mini 2Gb	532211	Audio	\$139.00	10
Empty Shopping Cart		Continue Shopping	Place Order	Order Total: \$1,390.00

- Click **Place Order** to submit the order.

The welcome page updates with an Order Submitted message.

**SOA Release Demo Web Client**

<a href="#">Information</a>
Order Submitted!

### 3.1.3 Task 3: View the Order in the Oracle Enterprise Manager 10g BPEL Control Oracle BPEL Control

Once you placed the order, a message was sent to OrderBookingESB, which initiated a the SOAOrderBooking process. You can monitor the progress of that BPEL process from the Oracle Enterprise Manager 10g BPEL Control (Oracle BPEL Control).

To view monitor the SOAOrderBooking process:

- Log into the Oracle BPEL Control by selecting **Start > All Programs > Oracle - Oracle - soademo > Oracle BPEL Process Manager > BPEL Control**.
- When prompted, enter oc4jadmin in the **Username** field and welcome1 in the **Password** field.

The **Dashboard** tab of the Oracle BPEL Control appears. The **Name** column lists all of the BPEL processes that are deployed to this server. You could click those to perform some management activities of initiate them for testing. For now, you will view a particular running instance of one of those processes.

ORACLE® Enterprise Manager 10g

BPEL Control

Manage BPEL Domain | Logout | Support

Logged to domain: default

Dashboard

BPEL Processes

Instances

Activities

Deployed BPEL Processes

In-Flight BPEL Process Instances 1 - 1

Name	Instance	BPEL Process	Last Modified ↑
DHLShipment	1 : Instance #1 of SOAOrderBooking	SOAOrderBooking (v. 1.0)	8/7/06 12:39:57 PM
SOAOrderBooking			
SelectManufacturer			
TaskActionHandler			
TaskManager			

Deploy New Process

- Click the **Instances** tab.

This tab lists all of the instances. Those with a green checkmark are completed; those without a green checkmark are still in progress. The order you just submitted is not complete.

ORACLE® Enterprise Manager 10g

BPEL Control

Manage BPEL Domain | Logout | Support

Logged to domain: default

Dashboard

BPEL Processes

Instances

Activities

Locate Instances

List of BPEL Process Instances 1 - 1

Instance Id#

Title

Priority

BPEL Process  

All Processes

Creation Date  

All Times

State  

All States

Test Filter  

Show All

Go

Instance	BPEL Process	Last Modified ↑
1 : Instance #1 of SOAOrderBooking	SOAOrderBooking (v. 1.0)	8/7/06 12:39:57 PM

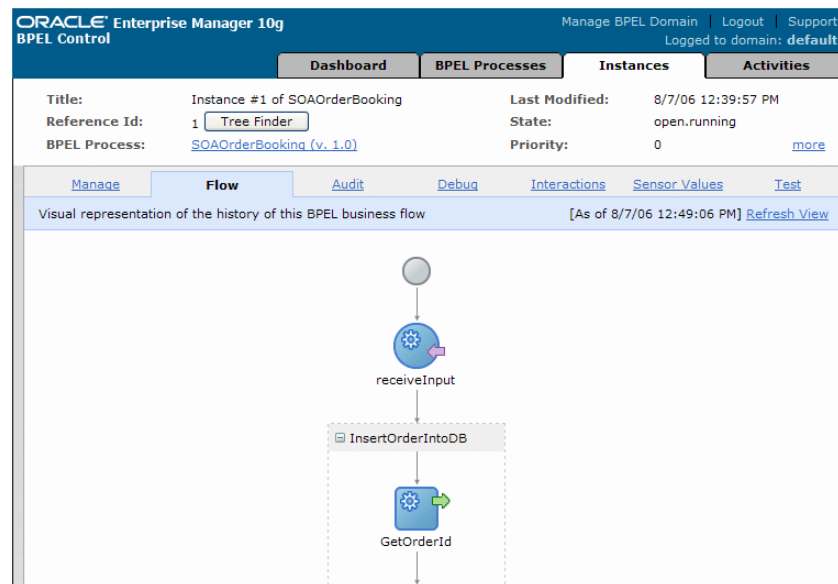
- Find the most recent instance of SOAOrderBooking by using the timestamp in the **Last Modified** column, and click the *nnn*: Instance #*nnn* of SOAOrderBooking (where *nnn* is some number) in the **Instance** column to select the instance. You may see more than instance if other orders were previously submitted.

If you do not see the link, refresh the page. The first time you run a BPEL process after starting the server can sometimes be a little slower, as everything needs to initialize.

The instance information for the selected instance displays. There are various tasks and actions that you can perform for an instance.

- Click the **Flow** sub-tab for a visual representation of this instance.

This view enables you to view the progress of the process.



The icons in the flow are referred to as activities. You can click them to view their details.

- Click the first activity, the blue, circle **receive** activity labelled **receiveInput**.

The Activity Audit window displays with the XML associated with the activity.



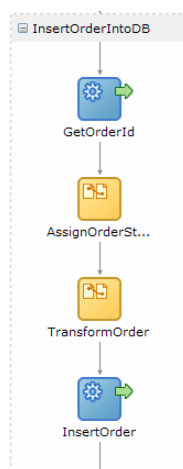
In this case, the audit trail displays the XML input to this BPEL process instance that came from the client Web page through ESB. Note the following:

- **CustId** is 10: Represents the ID of the customer submitting the order
- **EmailAddress** is sking@soademo.org: Identifies the email address you used to login
- **<OrderItems>** element: Shows the item you ordered

The green text shows the XML namespaces being used, and can be ignored.

7. Close the Activity Audit Trail window to return to the flow diagram.

Under **receiveInput**, you will see a grey box labelled **InsertOrderIntoDB**, which contains several activities. **InsertOrderIntoDB** is a scope and it groups activities together into logical chunks. Therefore, all of the activities in **InsertOrderIntoDB** are related to writing the order into the database.



8. Click the invoke activity labelled **GetOrderId** to open the XML information Activity Audit window.

An invoke activity is used to invoke a service. In this case, a service is called to generate a unique order number for this order. Notice that a number is being

returned in the `<order_seq_id_gen.nextval>` element. This is the order ID that has been generated.

9. Close the Activity Audit Trail window to return to the flow diagram.
10. Click the **InsertOrder** invoke activity to open the XML information Activity Audit window.

This invoke activity is invoking the database adapter to actually write the record to the database. The database adapter exposes this functionality as a service, which is why the invoke activity is used.

The XML that appears in the Activity Audit Trail window shows the data being written to the database. Notice that `<ordid>` is set to the number you saw in the previous step. Also note that the other data came from the input data received by the process.

11. Close the Activity Audit Trail window to return to the flow diagram.
12. In the **CustomerService** scope, click the **GetCustInfo** invoke activity.

In the `<customerServiceRequest>` element, you will see the data passed to the service as input to that service. In this case, `<custid>` is 10, which is the ID of the customer submitting this order.

The reason you see both the input and output to this service is because it is being invoked synchronously (as opposed to asynchronously).

13. Close the Activity Audit Trail window to return to the flow diagram.
14. In the **CreditService** scope, click the **InvokeCreditService** invoke activity

Recall that BPEL is used to define your business process. This service takes credit card information as input and returns true or false depending on the validity of the credit card.

Since this is a synchronous invocation, you see the request and response data. The request data is in the `<validateRequest>` element. As you can see, the credit card information obtained in the previous step is submitted.

In the `<validateResponse>` element, notice that the service returns true, meaning this customer's credit card is valid.

15. Close the Activity Audit Trail window to return to the flow diagram.
16. Scroll down to the **DecisionService** activity, near the end:



This decision service activity represents the invocation of the rules engine for a dynamic decision at runtime. As previously explained, orders over \$1000 require manager approval, unless this customer is a Platinum customer. Because sking is a Gold customer, manual approval is required.

17. Click the + icon to expand the decision service scope, and then click the invoke icon labelled **Invoke**, which is where the rules engine service is actually invoked.

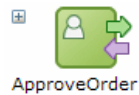
The `<dsIn>` element contains the input data that was sent to the service. Within that element, you see a sub-element called `<approve>`, which shows the price and customer status. The decision service needs this information to make its decision.

Since the order price is over \$1000 and this customer is not Platinum, the decision service returns **true**, signifying that approval is required. You can see this return value from the service in the `<dsOut>` element:

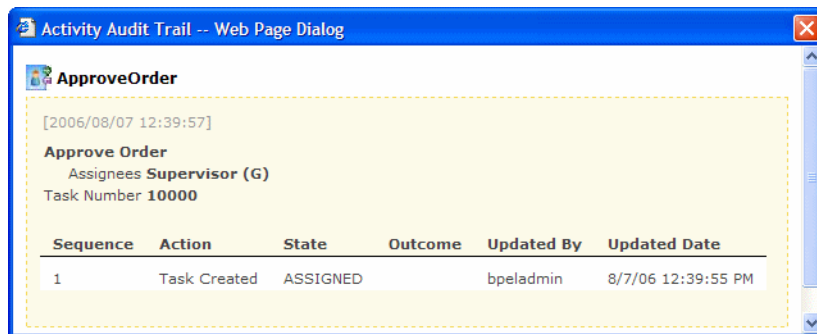
```
<approvalRequired>true</approvalRequired>
```

In a later task, you will modify the business rule to change its behavior.

18. Close the Activity Audit Trail window to return to the flow diagram.
19. Scroll to the bottom of the process flow, and click the **ApproveOrder** activity.



The Activity Audit Trail window shows information pertaining to this human workflow activity rather than XML.



In the **Assignees** field, notice that this task has been assigned to the Supervisor group.

20. Close the Activity Audit Trail window to return to the flow diagram.
- In the next task, you will approve this order by "changing hats" to a user that has approval privileges.
21. Keep the Oracle BPEL Control running, as you will need it for future tasks.

### 3.1.4 Task 4: Use the Oracle BPEL Worklist Application to Approve Order

To approve the order, you will use the worklist application supplied when you installed Oracle SOA Suite.

Once you are familiar with Oracle SOA Suite, you can use the workflow engine, which has a rich API, to create your own custom GUIs to let your users perform approval and management tasks.

To use the Worklist Application to approve the order:

1. Go to the browser running the Worklist Application, and then click the **Refresh** button in your browser.

The **My Tasks** tab shows the **Approve Order** tasks assigned to the Supervisor group. `jcooper` is a defined member of that group.



ORACLE<sup>®</sup> BPM Worklist  
Welcome, jcooper [jazzn.com]

Home | Reports | Preferences | Logout

My Tasks | Initiated Tasks

My Tasks (Inbox)

Work Queues

Inbox

My Work Queues

Standard Views

- High Priority Tasks
- Tasks Due Soon
- New Tasks

My Views

None

Proxy Work Queues

Delegated Views

Search:  My & Group  Any  Assigned  Go

Keyword Category Priority Status Advanced Search

Task Number	Title	Priority	Assigned Users	Assigned Groups	State	Created Date	Expiration Date	Actions
10000	Approve Order	3		Supervisor	Assigned	Aug 7, 2006 12:39 PM		-- Select an Action -- Go

The Worklist Application enables users to manage their tasks, create escalations, setup routing rules to automatically handle tasks, create vacation rules to say how tasks should be handled while away, run reports, and so on. All of this functionality is based on the engine APIs. Therefore, you can also add this functionality to any customized screens you create.

Because this task was assigned to a group, it needs to be acquired by someone from the group to ensure that only one person at a time makes any changes.

- From the **Actions** list, on the right-hand side, select **Claim**, and then click **Go**.

Expiration Date

Actions

-- Select an Action -- Go

-- Select an Action --

Claim

The Tasks Details page appears, displaying details of the order to help you make your decision about whether or not to give approval. You can add comments or attachments. In this case none of the other fields can be edited, but that is just in this application. It is possible to make the data fields editable.

ORACLE<sup>®</sup> BPM Worklist  
Welcome, jcooper [jazzn.com]

Home | Reports | Preferences | Logout

✓ Your request was processed successfully.

My Tasks | Initiated Tasks

My Tasks > Task Details (Approve Order)

Task Action: -- Select an Action -- Go

Delegate...

Save

[10000] Approve Order

State : ASSIGNED

Outcome:

Priority : 3

Created Date : 08/07/06

Updated Date : 08/07/06 02:46 PM

Expiration Date:

Creator :

Acquired By: jcooper

Assignees : Supervisor(U)

Purchase Order

- From **Task Action** list, at the top left of the page, select **APPROVE**, and then click **Go**.

The **My Tasks** tab updates to show that no worklist tasks are currently assigned.

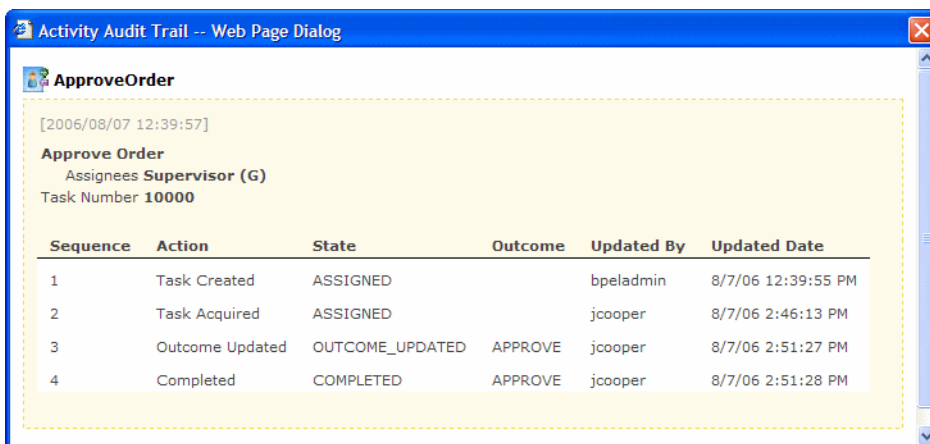
### 3.1.5 Task 5: View Approval in the Oracle BPEL Control

- Go back to the Oracle BPEL Control browser window, and then click the **Refresh** button in your browser.
- In the **Flow** view of the **Instances** tab, scroll down to the **ApproveOrder** human task that the process was previously stopped at:

Notice that there are more activities beneath it, indicating that the BPEL process has progressed.

- Click the **ApproveOrder** activity to again see the workflow information.

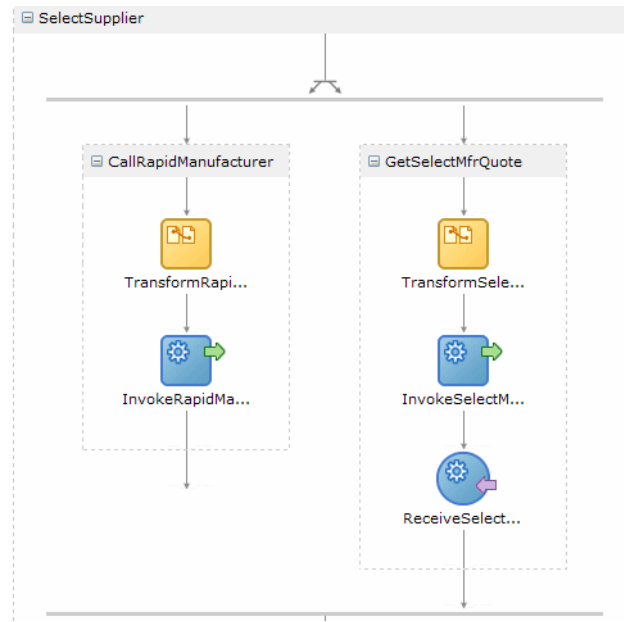
The Activity Audit Trail window shows information pertaining to this human workflow activity rather than XML.



Sequence	Action	State	Outcome	Updated By	Updated Date
1	Task Created	ASSIGNED		bpeladmin	8/7/06 12:39:55 PM
2	Task Acquired	ASSIGNED		jcooper	8/7/06 2:46:13 PM
3	Outcome Updated	OUTCOME_UPDATED	APPROVE	jcooper	8/7/06 2:51:27 PM
4	Completed	COMPLETED	APPROVE	jcooper	8/7/06 2:51:28 PM

Notice the audit trail of the activities that took place on the task. In this case it was a straightforward approval task. But you can have very complex workflow tasks that involve sophisticated routing rules and approval chains. You can monitor the progress of a workflow from the Oracle BPEL Control or the Worklist Application.

- Close the Activity Audit Trail window to return to the flow diagram.
- Scroll down to the large scope labelled **SelectSupplier**.



This scope contains a flow activity which lets a BPEL process perform tasks in parallel. In this case, it solicits the rapid and select manufacturer partners to determine who can sell the required items for the cheapest price. It does not make sense to call one partner, wait for their response, then call another partner and wait for their response, and so on. One slow response can cause delays to your process.

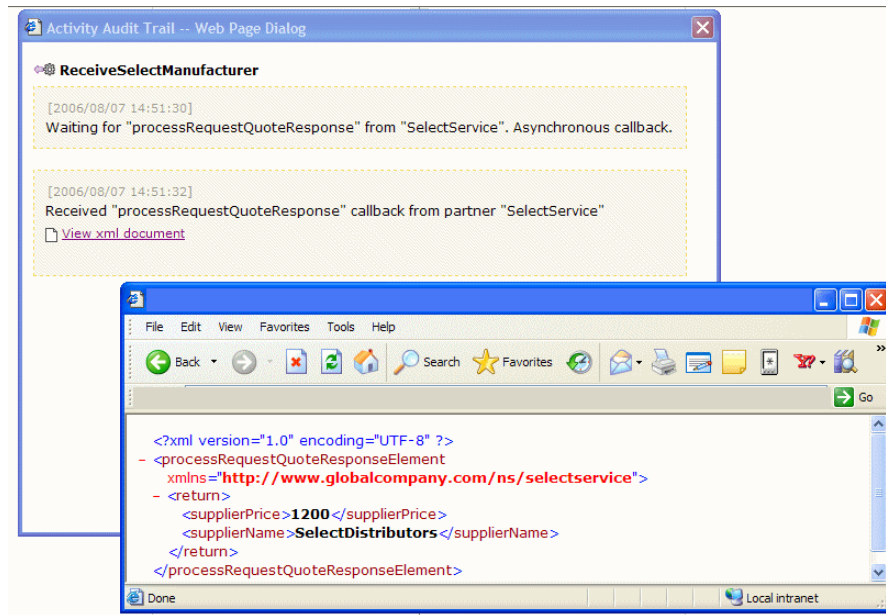
By using a flow activity, BPEL can execute these requests in parallel, and wait for all responses before continuing. In this case, the flow has just two parallel groups.

The rapid manufacturer service is synchronous, which is why it only has a single invoke. The select manufacturer service is asynchronous because it can take a relatively long time to respond. Asynchronous services are invoked in the same way as a synchronous service, but the return data is received using a receive activity.

6. Click the **InvokeSelectManufacturer** invoke activity to open the XML information Activity Audit window.

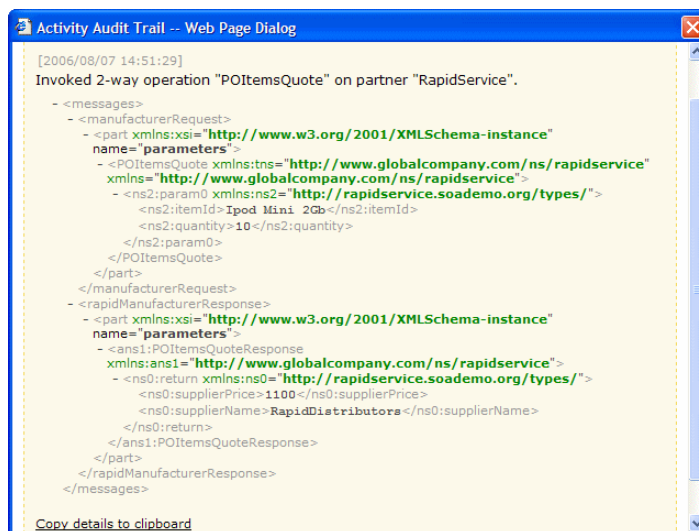
In the Activity Audit window, notice that the only input to the service being invoked is shown. Further, only the minimal information the service needs is sent: the items to buy and their quantity. Private customer data is not sent to external partners. The partners only need to know the quantity and items to provide a quote.

7. Close the Activity Audit Trail window to return to the flow diagram.
8. Click the **ReceiveSelectManufacturer** invoke activity to open the XML information Activity Audit window, and then click the **View xml Document** link.



The return data from the select manufacturer service displays, the supplier name and their price. Note the price \$1,200 price, as you will compare it to the price returned by the rapid manufacturer service in the following steps.

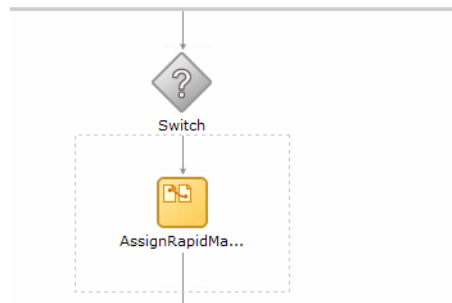
9. Close the XML window, and then close the Activity Audit Trail window to return to the flow diagram.
10. Click the **InvokeRapidManufacturer** invoke activity to open the XML information Activity Audit window.



Notice the input to the service in the **<manufacturerRequest>** element. Again, it provides only enough information for the partner to provide a quote.

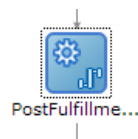
The response data is in the **<rapidManufacturerResponse>** element. The value from the rapid manufacturer service is \$1,100, which is lower than the select manufacturer price.

11. Close the Activity Audit Trail window to return to the flow diagram.
12. Look at the **Switch** activity under the flow activity:



A switch activity in BPEL is like an if-then-else or case or switch statement in other program languages. In this case, the switch activity compares the two prices and selects the lowest one.

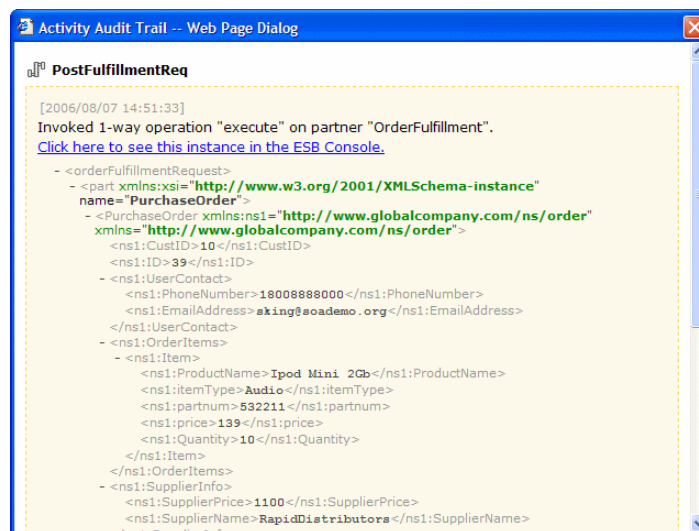
13. Scroll to the next scope labelled **PostFulfillmentReq**.



Now that the order is approved and the manufacturing partner is selected, the next step in the business process is to fulfill the order, that is actually send it to the customer.

The SOAOrderBooking process defines that FulfillmentESB is responsible for routing the order to the correct fulfillment destination. The PostFulfillmentReq activity invokes FulfillmentESB.

14. Click the **PostFulfillmentReq** activity to open the XML information Activity Audit window to see ESB being invoked.

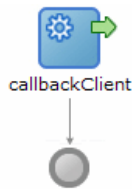


While it is possible to click link **Click here to see this instance in the ESB Console**, in this scenario, it is important to complete viewing the SOAOrderBooking process flow first.

15. Close the Activity Audit Trail window to return to the flow diagram.
16. Scroll to the next two scopes, **SetFinalOrderStatus** and **NotifyCustomer**.

In these scopes, the database is updated with supplier information and invokes a notification service which emails the customer details of the order.

17. Scroll to the **callbackClient** invoke activity.



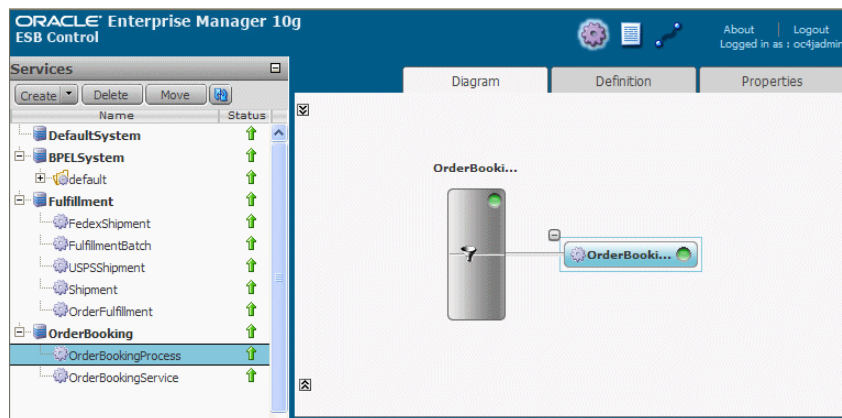
This activity is the final step. Remember, anytime you have a BPEL process it is also a service. Some services return data and some do not. In this case, the BPEL process does not return data. However, sometimes a BPEL process returns data, and you can see that data by clicking on the final activity.

### 3.1.6 Task 6: View Instances in Oracle Enterprise Manager 10g ESB Control

1. Log into the Oracle Enterprise Manager 10g ESB Control (Oracle ESB Control) by selecting **Start > All Programs > Oracle - Oracle - soademo > Oracle ESB > ESB Console**.
2. When prompted, enter `oc4jadmin` in the **Username** field and `welcome1` in the **Password** field.

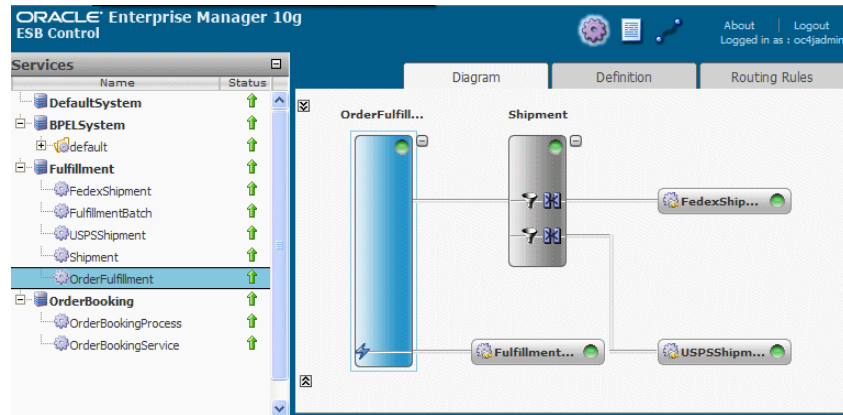
The initial screen shows the available services and their definitions. They are grouped into logical groups.

3. Under **OrderBooking**, click **OrderBookingProcess**.



The Oracle ESB Control shows the BPEL process was initiated. The grey box labelled **OrderBooking** represents a routing service that routes messages. In this case, it is only routing one message, to start the BPEL process, indicated in the blue box labelled **OrderBooking**.

4. Under **Fulfillment**, click **OrderFulfillment**.



This service is more complex. A message arrives from the SOAOrderBooking BPEL process to process an order. **OrderFulfillment** routes the message to two places, the **Shipment** routing service and **FulfillmentBatch**. **FulfillmentBatch** is a JMS queue that is responsible for storing all fulfillment orders for overnight batch processing.

The **Shipment** routing service is responsible for making sure the right shipping service is used. For orders over \$500, Fedex (**FedexShipment**) is used. Otherwise, USPS (**USPSShipment**) is used. Oracle ESB supports content-based routing and can inspect the message to use the order price to route the message the correct destination.

5. Click the **Instances** icon, at the top of the screen, toward the right-hand side.



The **Instances** pane appears on the left-hand side.

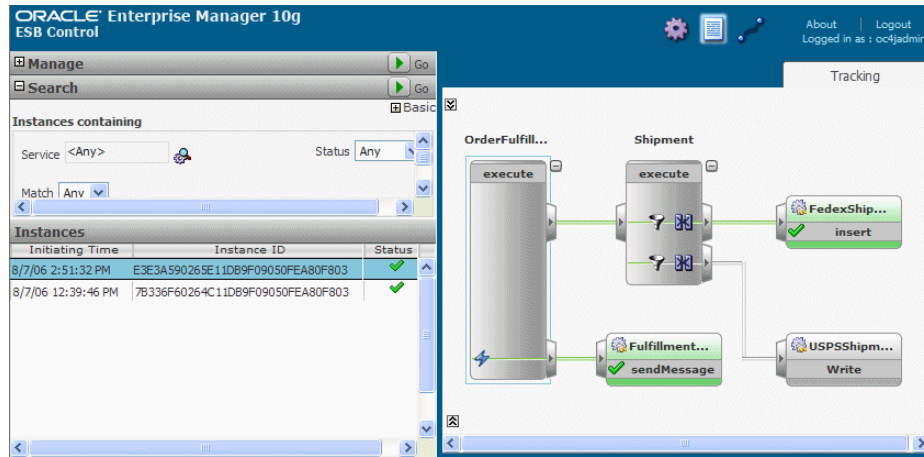
The screenshot shows the 'Instances' pane in the Oracle Enterprise Manager 10g ESB Control interface. The pane has a search bar and a table of instances. The table has columns for 'Initiating Time', 'Instance ID', and 'Status'. Two instances are listed, both with a status of 'Success' (indicated by a green checkmark).

Initiating Time	Instance ID	Status
8/7/06 2:51:32 PM	E3E3A590265E11D89F09050FEA80F803	Success
8/7/06 12:39:46 PM	7B336F60264C11D89F09050FEA80F803	Success

6. Click the instance at the top of the list.

The Fulfillment flow displays. It shows the runtime instance of the ESB service and which path the messages took, as represented by the green. In this case, notice that **OrderFulfillment** invoked both the **FulfillmentBatch** and **FedexShipment** services.





7. Hover your mouse over the filter icon in the **Shipment** routing service for USPS.  
It shows `OrderPrice < 500`, meaning that this service should be invoked if the order price is less than \$500. Since the order price was greater than 500, this service was not invoked, as represented by the grey.
8. Hover your mouse over one of the transformation icons in the **Shipment** routing service.



This icon represents a data transformation. When ESB routes a message to FedexShipment or USPSShipment, they expect the data to be in a certain format. Data is transformed from Global Company's format to that which the target service expecting. In this case, that format is XSL.

XSL stands for XML Stylesheet and is the standard way to do XML data transformations. You can create XSL files using JDeveloper or third party tools.

9. Keep the respective browser windows for the Web client, Oracle BPEL Control, and Oracle ESB Control running.

You will use these applications in the next task.

## 3.2 Increasing Approval Order Amount from \$1000 to \$2000 and Resubmitting Order

As explained in [Section 1.3, "Introduction to SOA Order Booking Application"](#) on page 1-5, the SOAOrderBooking process uses a decision service to determine whether or not the order requires manual management approval. This service, in turn, uses rules in the Oracle Rules Engine to determine if approval is required.

In this scenario, you will increase the approval amount from \$1000 to \$2000 without redeploying the SOA Order Booking application. You will then submit an order for over \$1000 and watch it process without manual approval.

In this scenario, you will perform the following tasks:

- [Task 1: Use Rule Author to Increase Approval Order Amount](#)
- [Task 2: Place Order in Web Client](#)
- [Task 3: View the Order Approval in the Oracle BPEL Control](#)



### 3.2.1 Task 1: Use Rule Author to Increase Approval Order Amount

In this task, you change the approval amount of \$1000 to \$2000 in the Oracle Business Rules Rule Author. The Rule Author enables you to create or modify rules and to create a data model that describes the business objects that you use with rules.

1. Open the Rule Author by pointing to the following URL:

`http://localhost:8888/ruleauthor`

2. When prompted, enter `oc4jadmin` in the **Username** field and `welcome1` in the **Password** field.

The Welcome page appears.

3. Click the **Repository** tab.

4. Select **File** for the **Repository Type** and set the **File Location** to:

`ORACLE_HOME\j2ee\home\applications\rules_default_SOAOrderBooking_1_0_  
DecisionService\DecisionService-web\WEB-INF\repository\sample_repository`

5. Click **Connect**.

A Confirmation message displays with the connection details.

6. Click the **Load** subtab.

The Load Dictionary page appears.

7. Select `OrderBookingRules` from the **Existing Dictionaries** list and `INITIAL` from **Version** list.

The Load Dictionary page appears.

8. Click **Load**.

A Confirmation message displays with loaded dictionary details.

9. Click the **Rulesets** tab.

The screenshot shows the Oracle Rule Author interface with the **Rulesets** tab selected. The **RuleSet Summary** section displays a table with one rule set: `ApproveOrderRequired`. The left-hand margin shows a tree view of the loaded dictionary `OrderBookingRules`, with `ApproveOrderRequired` expanded to show its three rules: `platinumMember`, `overLimit`, and `belowLimit`.

RuleSet Summary	
<a href="#">Delete</a>   <a href="#">Create</a>	
<a href="#">Select All</a>   <a href="#">Select None</a>	
Select Name	Edit
<input type="checkbox"/> ApproveOrderRequired	<a href="#">Edit</a>

It shows a single rule set called **ApproveOrderRequired**, which has three rules: **platinumMember**, **overLimit**, and **belowLimit**.

10. Select the **platinumMember** link from the left-hand margin.

The screenshot shows the Oracle Rule Author interface. The left-hand margin displays a tree structure under 'Dictionary loaded: OrderBookingRules'. The tree includes 'Ruleset', 'ApproveOrderRequ', 'platinumMember' (selected), 'overLimit', and 'belowLimit'. The main area is titled 'RuleSet Summary > Ruleset > Rule'. It contains fields for '\* Name' (platinumMember), 'Description', and 'Priority' (0). Below these fields are 'New Pattern' and 'Delete' buttons. The 'If' section contains a condition: 'approve is a ApproveType and approve.status == "Platinum"'. The 'Then' section is empty, with 'New Action' and 'Delete' buttons. At the bottom, there is a navigation bar with links: Home, Repository, Definitions, Rulesets, Customization, and RL.

The **Rule** section the rule definition for **platinumMember**. The **If** section specifies the condition and the **Then** section specifies the result. In this case, if the customer status is Platinum, then **approvalRequired** is **false**, meaning no approval is required.

11. Select the **overLimit** link from the left-hand margin.

The screenshot shows the Oracle Rule Author interface. The left-hand margin displays a tree structure under 'Dictionary loaded: OrderBookingRules'. The tree includes 'Ruleset', 'ApproveOrderRequ', 'platinumMember', 'overLimit' (selected), and 'belowLimit'. The main area is titled 'RuleSet Summary > Ruleset > Rule'. It contains fields for '\* Name' (overLimit), 'Description', and 'Priority' (0). Below these fields are 'New Pattern' and 'Delete' buttons. The 'If' section contains a condition: 'approve is a ApproveType and approve.price >= AUTOMATED\_ORDER\_LIMIT and approve.status != "Platinum"'. The 'Then' section contains an action: 'Assign approve.approvalRequired = true'. At the bottom, there is a navigation bar with links: Home, Repository, Definitions, Rulesets, Customization, and RL.

The **Rule** section specifies that if the price is over a certain limit and the customer status is not Platinum, approval is required. Notice that the limit is specified by the constant **AUTOMATED\_ORDER\_LIMIT**.

12. Select the **belowLimit** link from the left-hand margin.

The screenshot shows the Oracle Rule Author interface. The top navigation bar includes 'Home', 'Repository', 'Definitions', 'Rulesets', 'Customization', and 'RL'. The 'Rulesets' tab is active. On the left, a tree view shows the 'OrderBookingRules' dictionary with a 'Ruleset' folder containing 'ApproveOrderRequ', 'platinumMember', 'overLimit', and 'belowLimit'. The 'belowLimit' rule is selected. The main area shows the 'Rule' configuration for 'belowLimit'. The 'If' section contains the condition: 'approve is a **ApproveType** and approve.price < AUTOMATED\_ORDER\_LIMIT'. The 'Then' section contains the action: 'Assign approve.approvalRequired = false'.

This rule specifies that if the price is under a certain limit, no approval is required. The constant **AUTOMATED\_ORDER\_LIMIT** is the same one used in the **overLimit** rule.

13. Click the **Definitions** tab.

The navigation tree shows the **Definitions** folder, which contains the available definitions.

14. Click the **Variable (1)** folder.

The Variable Summary page displays.

The screenshot shows the Oracle Rule Author interface with the 'Definitions' tab active. The left tree view shows the 'Definitions' folder containing 'Fact', 'Constraint', 'Variable (1)', and 'RLFunction (2)'. The 'Variable (1)' folder is selected. The main area shows the 'Variable Summary' page for 'DM.AUTOMATED\_ORDER\_LIMIT'. It includes a table with columns 'Select Name', 'Alias', and 'Edit'. The table contains one row: 'DM.AUTOMATED\_ORDER\_LIMIT' with alias 'AUTOMATED\_ORDER\_LIMIT' and an 'Edit' icon.

15. Click the **Edit** icon next to **DM.AUTOMATED\_ORDER\_LIMIT**.

16. Change the amount from 1000 to 2000 in the **Expression** text entry area.

The screenshot shows the Oracle Rule Author interface with the 'Definitions' tab active. The left tree view shows the 'Definitions' folder with 'Variable (1)' selected. The main area shows the 'Variable' configuration page for 'DM.AUTOMATED\_ORDER\_LIMIT'. The 'Name' is 'DM.AUTOMATED\_ORDER\_LIMIT', the 'Alias' is 'AUTOMATED\_ORDER\_LIMIT', and the 'Final' checkbox is checked. The 'Type' is 'float'. The 'Expression' text area contains '2000.00'. The 'OK', 'Cancel', and 'Apply' buttons are visible at the top right.

17. Click **Apply**.

A Confirmation message displays with the update status.

18. Click **Save Dictionary**, located at the top of the Rule Author window.19. When prompted to save the dictionary, click **Save**.

A Confirmation message displays with the dictionary save status.

## 20. Close the Rule Author window.

### 3.2.2 Task 2: Place Order in Web Client

To place the order:

1. In the welcome page, click the **Browse products and create a new order** link.

The Browse and Select Items page appears.

2. Click the **Next 10** link, and select **Ipod Mini 2 Gb**.

#### Browse and Select Items

Select and <a href="#">View Details</a>		<a href="#">Previous 10</a> 11-20 of 27 <a href="#">Next 7</a>				
Select	name	prodid	description	category	listPrice	producer
<input type="radio"/>	Muvo Personal MP3 Player	579823	Miniature 4gig Steroo MP3 Player	Handhelds	\$99.00	Muvo
<input type="radio"/>	Bluetooth Adaptor	427923	Multi-purpose Mini BlueTooth Adaptor	Wireless	\$19.00	Bluetooth
<input type="radio"/>	Bluetooth Phone Headset	298332	Wireless Bluetooth Phone Headset	Wireless	\$49.00	Bluetooth
<input type="radio"/>	Ipod Speakers	183923	Ipod Speaker Addon System	Audio	\$89.00	Ipod
<input checked="" type="radio"/>	Ipod Mini 2Gb	532211	2Gb Ipod Mini Personal Audio Player	Audio	\$139.00	Ipod
<input type="radio"/>	Ipod Shuffle 512Mb	545125	512Mb Ipod Shuffle Personal Audio Player	Audio	\$69.00	Ipod
<input type="radio"/>	Ipod Shuffle 1Gb	547321	1Gb Ipod Shuffle Personal Audio Player	Audio	\$99.00	Ipod
<input type="radio"/>	Ipod Video 30Gb	542122	30Gb Ipod Personal Audio and Video Player	Audio	\$299.00	Ipod
<input type="radio"/>	Ipod Video 60Gb	547642	60Gb Ipod Personal Audio and Video Player	Audio	\$399.00	Ipod
<input type="radio"/>	Ipod Nano 1Gb	392784	A portable media entertainment device.	Audio	\$149.00	Ipod

3. Click **View Details**.

The Item Details page appears. It displays detailed information about the product, and enables the user to select a quantity to add to their cart.

#### Item Details

<input checked="" type="checkbox"/> Item Details		<input type="button" value="Add to cart?"/>
name	Ipod Mini 2Gb	Quantity? <input type="text" value="10"/> <input type="button" value="Add to cart"/> <a href="#">Go to Shopping Cart</a>
description	2Gb Ipod Mini Personal Audio Player	
category	Audio	
listPrice	\$139.00	
prodid	532211	
producer	Ipod	
<input type="button" value="Continue Shopping"/>		

4. In the **Quantity** list, select 10, and click **Add to cart**.5. Click **Go to Shopping Cart**.

The Shopping Cart Contents page appears.

**Shopping Cart Contents**

Product Name	Part No.	Category	Price	Quantity
Ipod Mini 2Gb	532211	Audio	\$139.00	10
Empty Shopping Cart		Continue Shopping	Place Order	Order Total: \$1,390.00

- Click **Place Order** to submit the order.

The welcome page updates with an Order Submitted message.

**SOA Release Demo Web Client**

[Information](#)

Order Submitted!

**3.2.3 Task 3: View the Order Approval in the Oracle BPEL Control**

- Go back to the Oracle BPEL Control browser window.
- Click the **Refresh** button in your browser.

The **Dashboard** tab shows the SelectManufacturer and SOAOrderBooking flows completed without manual approval.

ORACLE Enterprise Manager 10g BPEL Control		Manage BPEL Domain   Logout   Support	
Dashboard		BPEL Processes	Instances
Deployed BPEL Processes		In-Flight BPEL Process Instances	
Name	Instance	BPEL Process	Last Modified ↑
DHLShipment			
SOAOrderBooking			
SelectManufacturer			
TaskActionHandler			
TaskManager			
Recently Completed BPEL Process Instances <a href="#">(More...)</a>			
✓ 3 : Instance #3 of SOAOrderBooking		SOAOrderBooking (v. 1.0)	8/7/06 11:38:02 PM
✓ 4 : Instance #4 of SelectManufacturer		SelectManufacturer (v. 1.0)	8/7/06 11:38:00 PM
✓ 1 : Instance #1 of SOAOrderBooking		SOAOrderBooking (v. 1.0)	8/7/06 2:51:33 PM
✓ 2 : Instance #2 of SelectManufacturer		SelectManufacturer (v. 1.0)	8/7/06 2:51:31 PM

© Deploy New Process

Oracle BPEL Console v10.1.3.1.0



---

## Adding BPEL and ESB Design Elements with Oracle JDeveloper

In [Chapter 3](#), you placed orders and watched them process through the SOA Order Booking business flow. In this chapter, you will learn how to use the JDeveloper BPEL and ESB Designers to add BPEL and ESB elements.

This chapter includes the following sections:

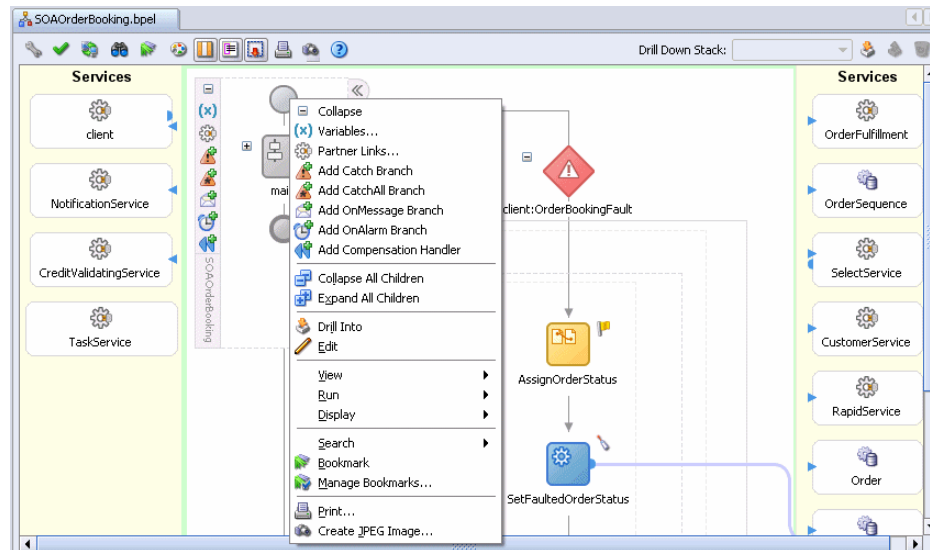
- [Section 4.1, "Introduction to the JDeveloper BPEL Designer"](#)
- [Section 4.2, "Adding a Credit Rating Service to the SOAOrderBooking Process"](#)
- [Section 4.3, "Introduction to the JDeveloper ESB Designer"](#)
- [Section 4.4, "Adding a New Shipping Target to FulfillmentESB Project"](#)
- [Section 4.5, "Learning More About Oracle SOA Suite"](#)

### 4.1 Introduction to the JDeveloper BPEL Designer

This section provides a brief introduction to the JDeveloper BPEL designer. In the next section, [Section 4.2, "Adding a Credit Rating Service to the SOAOrderBooking Process"](#), you will make modifications to the SOAOrderBooking BPEL process.

To familiarize yourself with the JDeveloper BPEL designer:

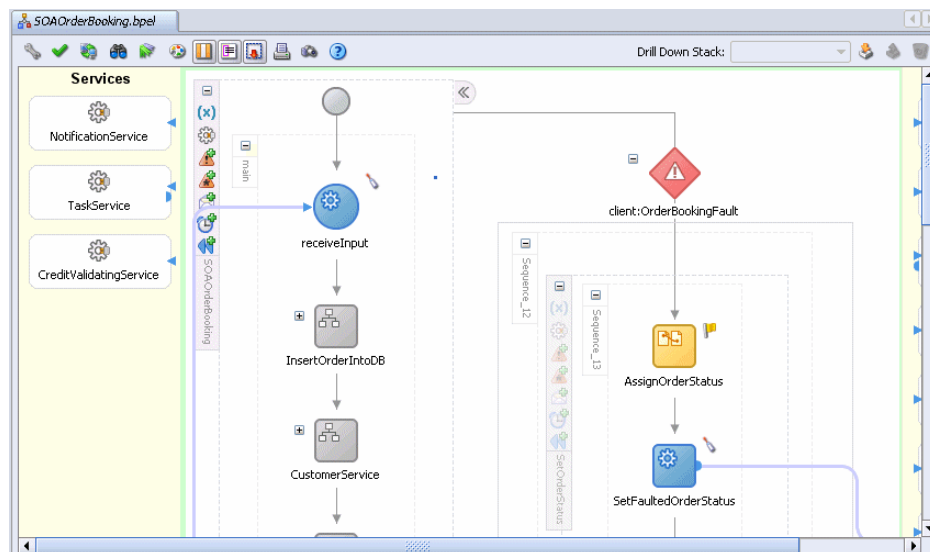
1. In the Applications Navigator of JDeveloper, expand **SOADEMO > SOAOrderBooking > Integration Content**.
2. Double-click `SOAOrderBooking.bpel`.  
The process displays in the JDeveloper BPEL Designer.
3. Right-click the **Start** icon at the top of the scope and select **Collapse All Children**.



The main logic of this BPEL process collapses into a scope called **main**. The other activities to the right under **client:OrderBookingFault** is the main exception handler for the process and can be ignored for now.

The yellow swim lanes on the left and right-hand sides of the main window are where the services reside. These are the services that are invoked at various stages of the BPEL process. The white area in the middle is where the BPEL logic resides.

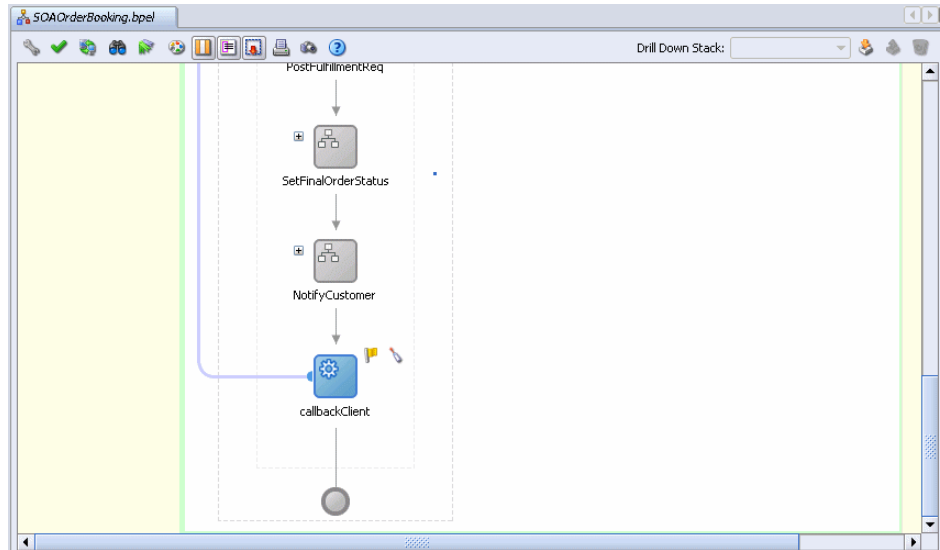
4. Expand the **main** scope by clicking the **Expand** icon (+).



The first round blue icon named **receiveInput** represents when something invokes this BPEL process. The input to this process is passed to this receive activity.

5. Scroll all the way to the bottom of the **main** scope to see a square blue invoke activity called **callbackClient**.





This activity represents the end of the BPEL process and returning the result data to the client that invoked this process. (BPEL processes do not have to return data. BPEL processes can simply end.)

Everything between **receiveInput** and **callbackClient** contains the logic of the BPEL process.

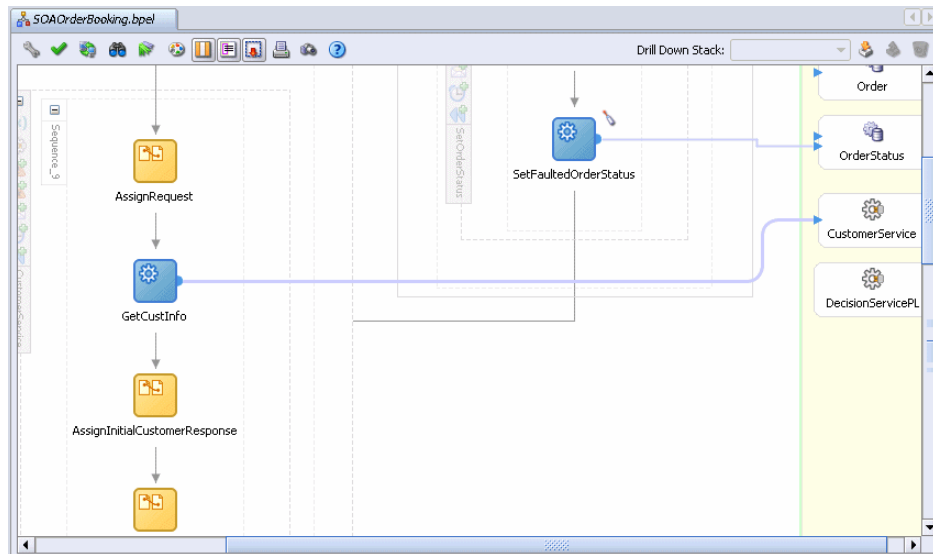
The BPEL language has several container activities, that is, activities that can contain other activities. The most commonly used is the scope activity. A scope activity does not actually execute or do anything, it simply holds other activities, including other scopes. Scopes are analogous to curly braces in Java. You can use them to break up your process into logical chunks.

6. Expand the **CustomerService** scope, located toward the beginning of the process, by clicking the **Expand** icon (+).

**Sequence\_9** appears. Many scopes contain an inner sequence. If you come across one, simply expand it to see the activities it contains.

7. Expand the **Sequence\_9** sequence by clicking the **Expand** icon (+).

When this BPEL process was invoked, part of the input to the process is the customer ID of the customer submitting the order. The business process takes that ID and retrieves the customer's details, such as address and credit card number. You can see this process with the **GetCustInfo** invoke activity. An invoke activity invokes a service. If you follow the blue line coming out of **GetCustInfo**, you will see that it invokes **CustomerService**.

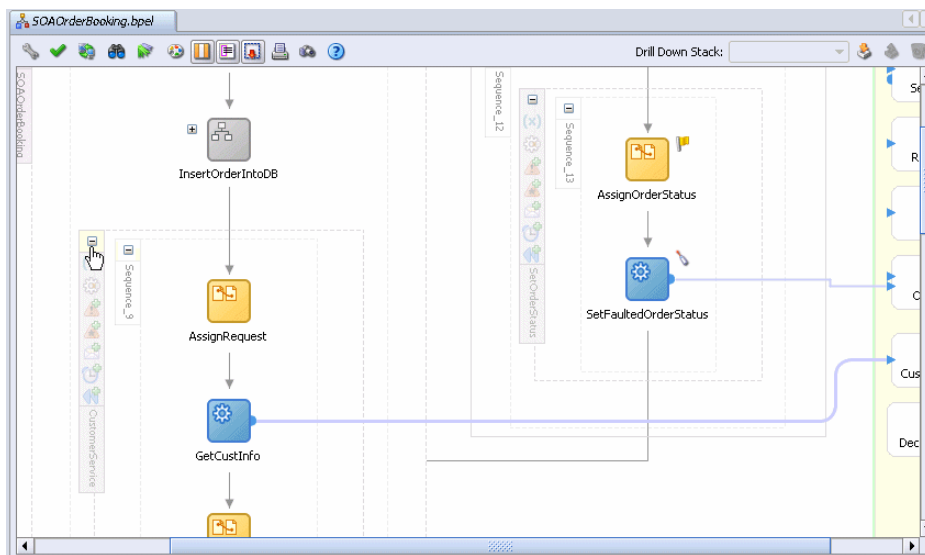


In this case, **CustomerService** is being invoked synchronously, so the BPEL process will wait until the service returns the results before continuing.

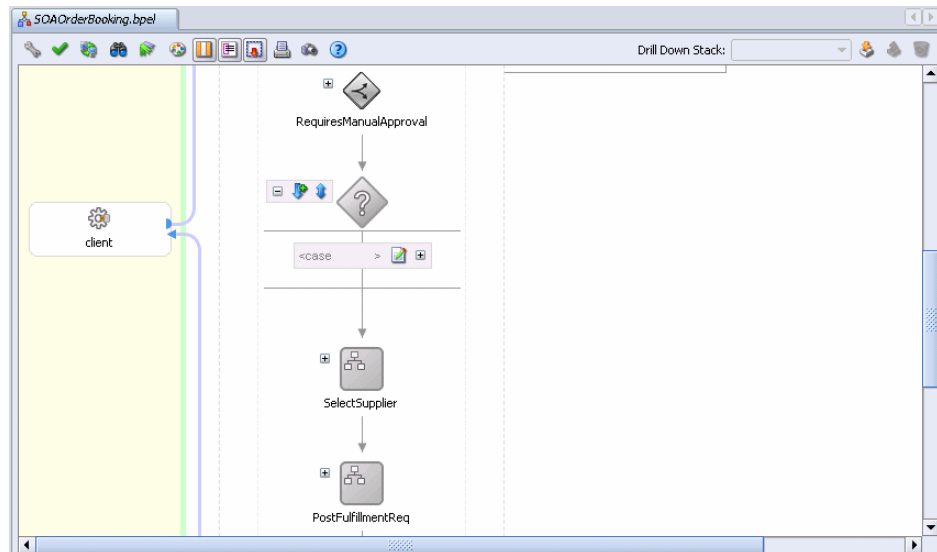
Before and after the **GetCustInfo** invoke activity, there are assign activities. An assign activity assigns data to variables. In this case, the customer ID is assigned to the variable that is sent as input to **CustomerService**, and another assign activity copies the return data from that service to a variable that can be used and referred to later in the BPEL process.

All of the activities in this scope are loosely involved in retrieving the customer information, which is why they are grouped together in a scope. Scopes break up the process into logical chunks to simplify development and make the process easier to read and work with.

8. Click the **Collapse** icon (-) icon on the **CustomerService** scope to collapse it.

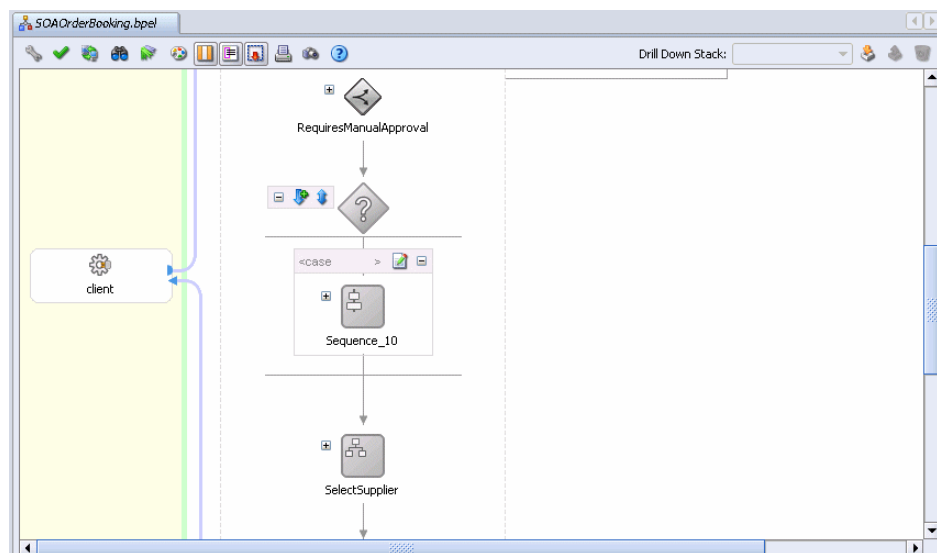


9. Scroll down to the **requiresApproval** switch activity and click the **Expand** icon (+).

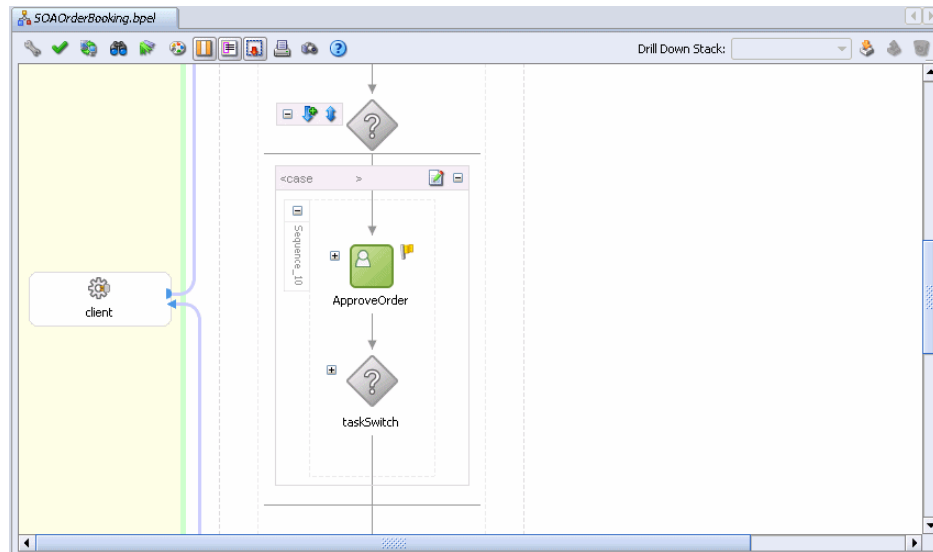


10. Click the **Expand** icon (+) on the **case** statement.

The sequence appears.



11. Click the **Expand** icon (+) on the sequence.



The green **ApproveOrder** activity represents invoking a human workflow activity as part of this BPEL process.

**12. Double-click **ApproveOrder**.**

The Human Task dialog appears.

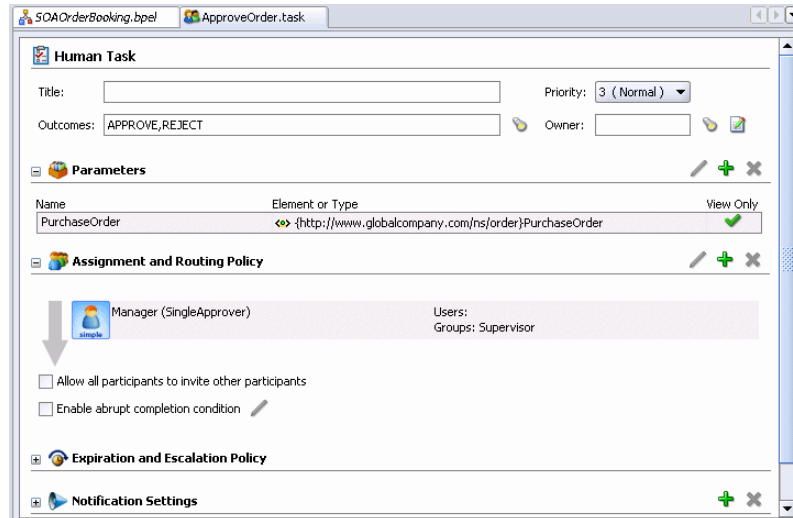
The screenshot shows the 'Human Task' dialog box with the 'General' tab selected. The dialog has several fields and sections:

- Task Definition:** A text field containing 'ApproveOrder'.
- Task Title:** A text field containing 'Approve Order'.
- Initiator:** An empty text field.
- Priority:** A dropdown menu set to '3'.
- Task Parameters:** A section with a table-like structure. It has two columns: 'BPEL Variable' and an empty column. The first row contains the text 'PurchaseOrder /client:SOAOrderBookingProcessRequest/ns4:PurchaseOrder'.

At the bottom of the dialog are buttons for 'Help', 'Apply', 'OK', and 'Cancel'.

**13. Click the **Edit Task Definition** icon. It is the third icon to the right of the **Task Definition** field that looks like a pencil.**

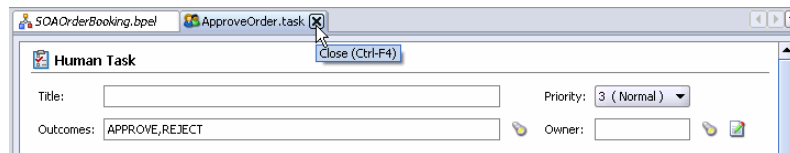
The ApproveOrder.task page displays. This screen enables you to define the behavior of your human workflow.



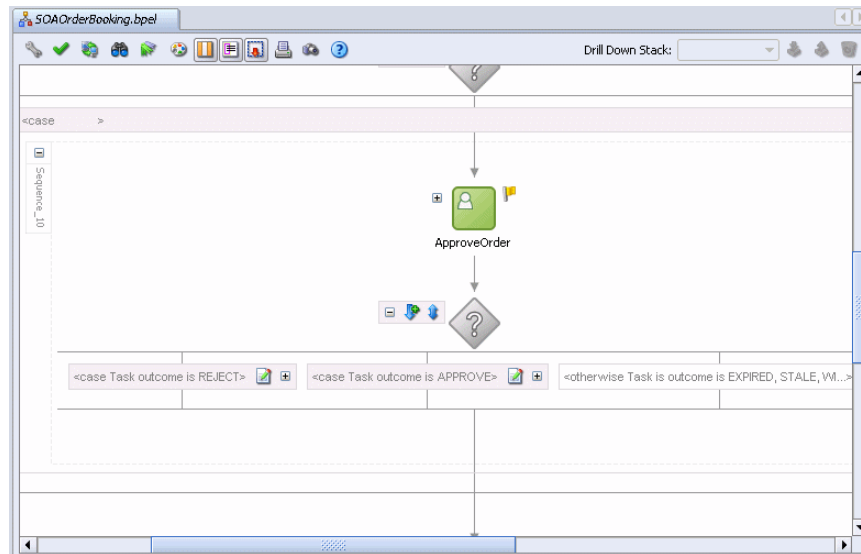
In this case, the **Outcomes** field shows that a request can be approved or rejected. You can define routing policies in the **Assignment and Routing Policy** section. In this case, a task is submitted to the Supervisor group for approval. For your process, you can specify a complex series of routing rules. For example, a workflow could specify to get approval from Sales, then from the Sales Manager, and then obtain final approval from a person named Fred.

There are several other settings that can be specified, such as how long until the task expires, how to notify participants in the workflow, escalation rules, and so on.

14. Close the ApproveOrder.task page by hovering your mouse over the tab and pressing X.



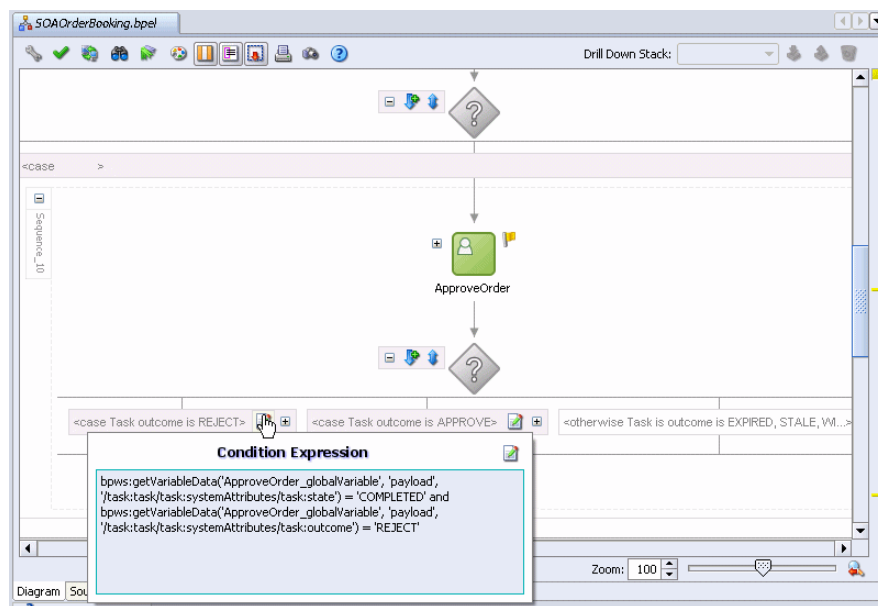
15. Back in the Human Task dialog, click **Cancel**.
16. Click the **Expand** icon (+) on the **taskSwitch** switch activity.



A switch activity is like an if-then-else or case or switch statement in other languages.

Notice that the **taskSwitch** activity has cases for handling if the human workflow task is rejected, approved, or something else, such as expired.

17. Click the **View Condition Expression** icon for the **REJECT** case to see the XPath evaluation expression, and then press the Escape key when done viewing.



18. Click the **Expand** icon (+) on the same **REJECT** case, and then the inner sequence.

It shows the activities that execute if the workflow task is rejected. In this case, data is assigned to some variable and an exception is thrown.

19. Scroll all the way to the top of the BPEL process, right-click the **Start** icon at the top of the scope, and select **Collapse All Children**.

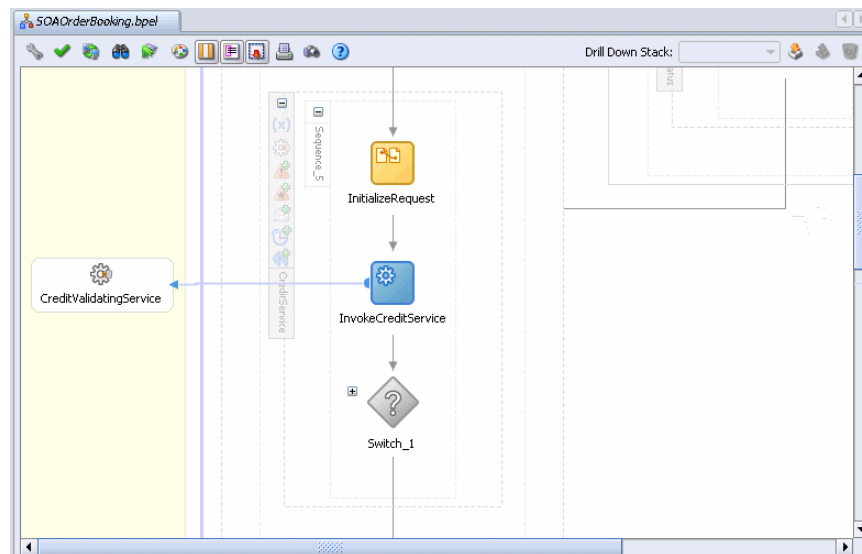
## 4.2 Adding a Credit Rating Service to the SOAOrderBooking Process

One of the strengths of BPEL is that it is an implementation-independent language for describing a business process. Because the logic of your business process is separate from the code, it is easy to re-orchestrate your BPEL flows when your business process changes.

For example, after deploying the existing BPEL process, you discover that Global Company has been selling its products to customers with a bad credit rating, resulting in a loss of money from these types. You can adjust the business process to check the credit rating of the buyer before selling the product.

The SOAOrderBooking BPEL process already does a credit check on the customer by checking their credit card. You can see this by following these steps:

1. Click the **Expand** icon (+) on the **main** scope.
2. Scroll down to the **CreditService** scope and click the **Expand** icon (+).
3. Expand the inner sequence, called **Sequence\_5**.



Notice there is an invoke activity called **InvokeCreditService**, which invokes the service called **CreditValidatingService**. In this case, the process is sending the credit card information to the Credit Validation service, and the service responds by determining if the credit card is valid or not.

4. Click the **Collapse** icon (-) icon on the **CreditService** scope to collapse it

In addition to checking if the customer's credit card is valid, you can modify the process to check the credit rating of customers. A Web service called **CreditRatingService** exists that the process can utilize. The Web service takes a Social Security number as its input, and returns the credit rating for that Social Security number.

The BPEL process needs to retrieve the Social Security number for the customer. Early in the BPEL process, **CustomerService** is called to retrieve several details about the customer, but Social Security number is not included. The Social Security number information is stored in a different database. In typical deployments, it is common for companies to store data in multiple places. Using BPEL, you can easily retrieve that different data and pull it together for your business processes.

In this scenario, you will modify the existing BPEL process to retrieve the Social Security number for customer `sking` from a different database. The BPEL process will then pass that Social Security number to the `CreditRatingService` service to get the customer's credit rating. You will perform the following tasks:

- [Task 1: Create a Database Connection for the Database Containing the Social Security Number](#)
- [Task 2: Use a Database Adapter As a Service to Access the Database](#)
- [Task 3: Create `getCreditRating` Scope Activity](#)
- [Task 4: Create Invoke Activity to Call `getSsn` Service](#)
- [Task 5: Assign Value to Input Variable](#)
- [Task 6: Install the Credit Rating Service](#)
- [Task 7: Create `CreditRatingService` Partner Link](#)
- [Task 8: Create Invoke Activity to Call `CreditRatingService`](#)
- [Task 9: Assign Data to the Input Variable for `CreditServiceRating`](#)
- [Task 10: Redeploy `SOAOrderBooking`](#)
- [Task 11: Test New Functionality by Placing a New Order](#)
- [Task 12: View the Order Approval in the Oracle BPEL Control](#)

#### 4.2.1 Task 1: Create a Database Connection for the Database Containing the Social Security Number

To access the Social Security number information, create a database connection to the database containing it. In this case, you will create another connection to the `SOADEMO` schema, which connection `soademo` is already using. In a real-world deployment, this connection would point to a different database. However, using a different table in the same database is adequate for this demonstration.

To create a database connection:

1. Start JDeveloper by running `JDEV_HOME\jdeveloper.exe`.
2. Create a connection to the database using the `SOADEMO` schema:
  - a. Click the **Connections** tab, or if it not currently displayed, choose **View > Connection Navigator**.
  - b. Double-click the **Database** folder to open the Create Database Connection Wizard.
  - c. Complete the wizard, ensuring you complete the following:
    - On the Step 1 of 4: Type 1 page, enter `soademo_ssn` for the **Connection Name**.
    - On the Step 2 of 4: Authentication page, enter `soademo` for the **Username** and **Password** fields, and click **Deploy Password**.
    - Enter the appropriate values for where you installed the schema.

#### 4.2.2 Task 2: Use a Database Adapter As a Service to Access the Database

In this task, you create a service that uses a database adapter to query the `SSN` table in the `SOADEMO` schema.



1. In the Component Palette, select **Services** from the dropdown.
2. Click and then drag the **Database Adapter** icon from the Component Palette and drop it in one of the yellow **Services** swim lanes of the SOAOrderBooking.bpel page.  
The Adapter Configuration Wizard appears.
3. On the Welcome page, click **Next**.
4. On the Step 1 of 2: Service Name page, in the **Name** field, enter `getSsn`, and then click **Next**.
5. On the Step 2 of 2: Service Connection page:
  - **Connection:** Select the `soademo_ssn` database connection.
  - **JNDI Name:** Ensure the name is set to `eis/DB/soademo_ssn`. The name is case sensitive. Ensure that it correctly matches the case of the connection name.
6. Click **Next**.
7. On the Step 3 of 3: Operation Type page, select **Perform an Operation on a Table**, deselect all operations, except for **Select**, which is required.
8. Click **Next**.
9. In the Select Table page, click **Import Tables**.  
The Import Tables dialog appears.
10. Click **Query** to display the tables.
11. Select **SSN**, and click **>** to move this table to the **Selected** area.
12. Click **OK** to save the setting, and return to the wizard.
13. On the Step 4 of 7: Select Table page, the **SOADEMO.SSN** table displays. Click **Next**.
14. On the Step 5 of 8: Define Primary Key page, select **CUSTOMERID**, and click **Next**.
15. On the Step 6 of 8: Relationships, click **Next**, as there are no relationships.
16. On the Step 7 of 8: Object Filtering page, leave the settings as they are, and click **Next**.  
When the Step 8 of 8: Define Selection Criteria page displays, notice the SQL query is set to:  

```
SELECT CUSTOMERID, SSN FROM SSN
```
17. On the Step 8 of 8: Define Selection Criteria page, in the **Parameters** section, click **Add** to add a parameter.  
The Parameter Name dialog appears.
18. Enter `custIdToGet`, and then click **OK**.  
At runtime, this parameter gets populated.
19. Back on the Step 8 of 8: Define Selection Criteria page, in the **SQL** section, click **Edit**.  
The Expression Builder dialog appears.
20. Click **Add** to add a `where` clause.

21. In the **Second Argument** section, change the selection from **Literal** to **Parameter**, making sure the populated list underneath specifies `custIdToGet`, and then click **OK**.

22. Back on the Step 8 of 8: Define Selection Criteria page, notice the SQL query changes to:

```
SELECT CUSTOMERID, SSN FROM SSN WHERE (CUSTOMERID =#custIdToGet)
```

23. Click **Next**.

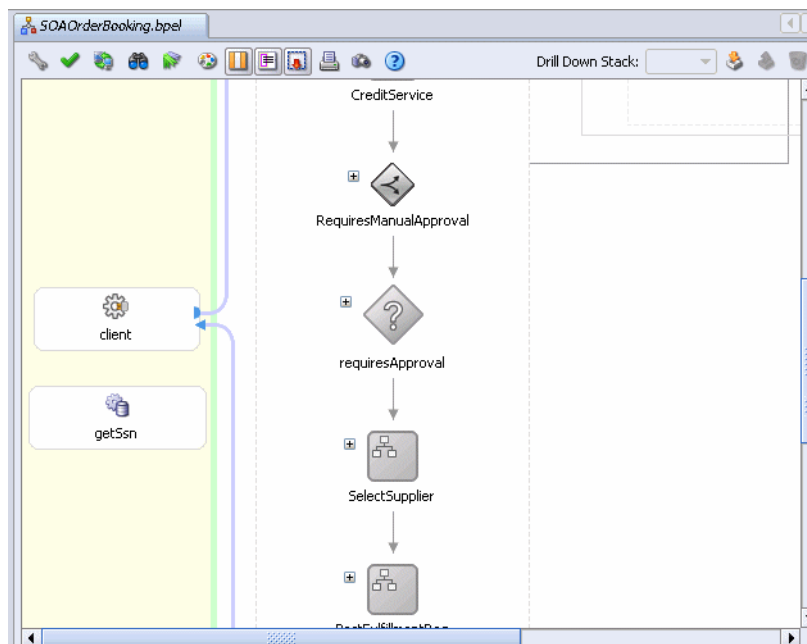
24. On the Finish page, click **Finish** to create the database adapter partner link.

The Create Partner Link dialog appears and is automatically completed as follows:

Element	Value
Name	getSsn
WSDL File	file:DEMO_HOME/SOAOrderBooking/bpel/getSsn.wsdl
Partner Link Type	getSsn_plt
Partner Role	getSsn_role
My Role	unspecified

25. Click **OK**.

The SOAOrderBooking.bpel page updates with the **getSsn** service. By making this database adapter into a service, you have made it appear as a Web service to the SOAOrderBooking process. You can now invoke this service just as you would with any Web service, and it will return a row from the database.



### 4.2.3 Task 3: Create getCreditRating Scope Activity

You now create a scope activity to group all activities that form a logical step to be executed.

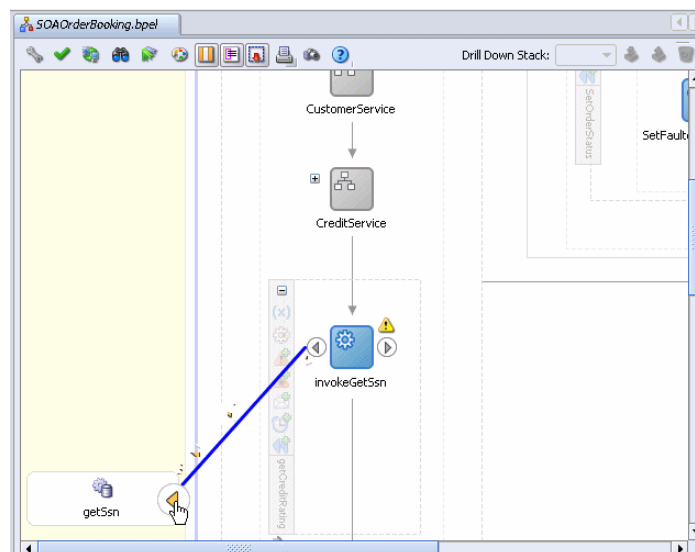
1. In the Component Palette, select **Process Activities** from the dropdown.
2. Drag and drop a **Scope** activity from the **Component Palette** section below the **CreditService** activity, but above the **RequiresManualApproval** activity
3. Rename this activity by double-clicking the name underneath the icon. Do not double-click the activity icon itself.
4. In the edit field, change the name to `getCreditRating`.
5. Click the Expand (+) icon to expand the **getCreditRating** scope.

#### 4.2.4 Task 4: Create Invoke Activity to Call getSsn Service

To call a service from BPEL, you use the invoke activity. The invoke activity will call the service and pass it data, and in this case, wait for a response from the service with the return data.

To create an invoke activity:

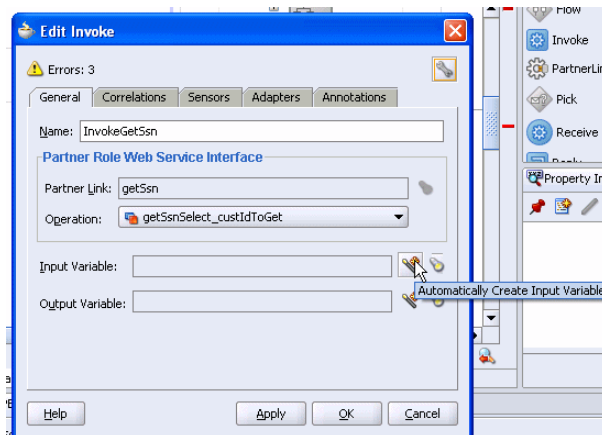
1. Drag and drop an **Invoke** activity from the **Component Palette** section inside the **getCreditRating** scope activity.
2. Rename this activity by double-clicking name underneath the icon. Do not double-click the invoke icon itself.
3. In the edit field, change the name to `invokeGetSsn`.
4. Drag the mouse from the left side of **invokeGetSsn** to the **getSsn** database adapter service.



The Edit Invoke dialog appears and is automatically filled in with the following information:

Element	Value
Name	invokeGetSsn
Partner Link	getSsn
Operation	getSsnSelect_custIdToGet

- Click the **Automatically Create Input Variable** icon. It is the first icon to the right of the **Input Variable** field.



The Create Variable dialog appears with the input variable. A variable named **invokeGetSsn\_getSsnSelect\_custIdToGet\_InputVariable** is automatically created in the **Name** field. This variable is automatically assigned a message type.

The variable name is based on the invoke activity (invokeGetSsn), database adapter (getSsn), and database adapter parameter (custIdToGet).

- Leave **Global Variable** selected, and click **OK**.

The Edit Invoke dialog populates with the variable in the **Input Variable** field.

- Click the **Automatically Create Output Variable** icon. It is the first icon to the right of the **Output Variable** field.

The Create Variable dialog appears with the output variable. A variable named **invokeGetSsn\_getSsnSelect\_custIdToGet\_OutputVariable** is automatically created in the **Name** field. This variable is automatically assigned a message type.

- Leave **Global Variable** selected, and click **OK**.

The Edit Invoke dialog populates with the variable in the **Output Variable** field.

These variables provide input and output data for the getSsn service.

- Click **OK** to save the variable settings.

## 4.2.5 Task 5: Assign Value to Input Variable

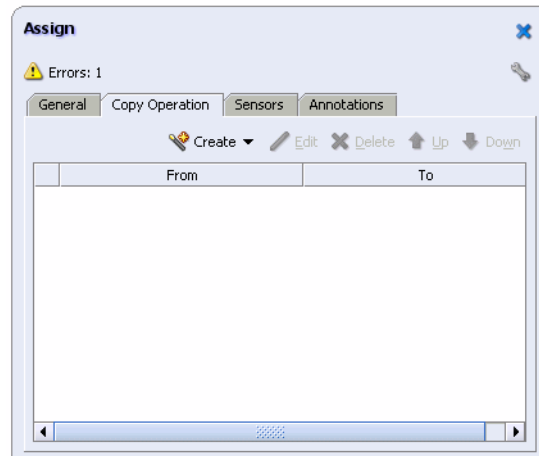
You now create an assign activity to take the customer ID and assign it to the input variable **invokeGetSsn\_getSsnSelect\_custIdToGet\_InputVariable**, which in turn sends a request to the getSsn service.

- Drag and drop an **Assign** activity from the **Component Palette** section to above the **invokeGetSsn** invoke activity and inside the **getCreditRating** scope activity.

An assign activity references variables in an invoke activity, which is why you create it after creating the invoke activity.

- Rename this activity by double-clicking name underneath the icon. Do not double-click the invoke icon itself.
- In the edit field, enter **assignCustID**.
- Double-click the **assignCustID** activity.

The Assign window displays.

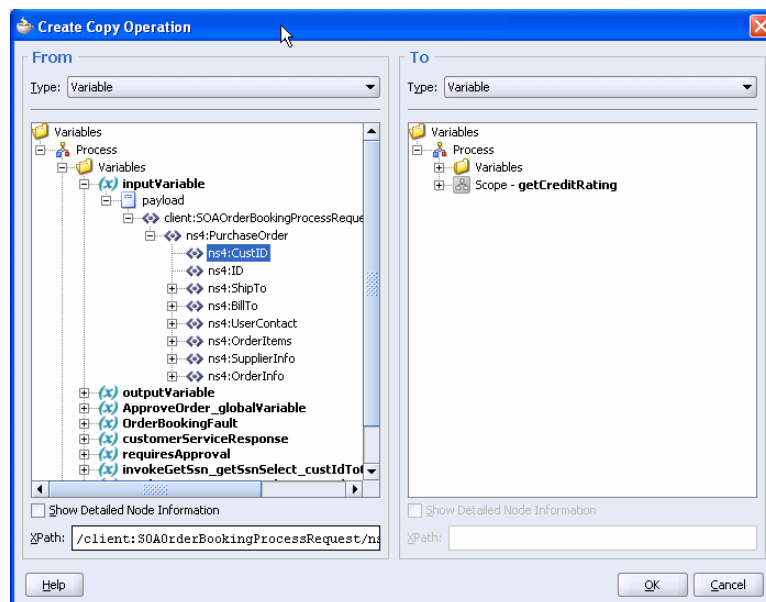


**Tip:** When using these type of windows in JDeveloper, do not click the X icon to apply changes. The X icon removes your changes. You must click OK to apply changes. If you cannot see the OK button, scroll down until you see it.

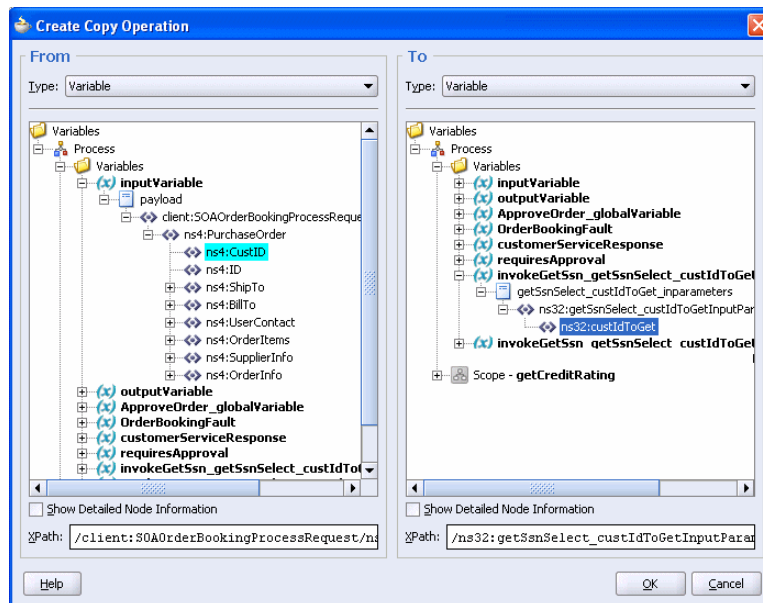
5. From the **Create** menu, select **Copy Operation**.

The Create Copy Operation dialog appears. It enables you to create a copy rule. In this dialog, you will specify a rule for populating the database adapter parameter (custIdToGet) with CustID in the input variable invokeGetSsn\_getSsnSelect\_custIdToGet\_InputVariable.

6. On the **From** side, expand **Variables > inputVariable > payload > client:SOAOrderBookingProcessRequest > ns4:PurchaseOrder**.
7. Select **ns4:CustID**. Your system may show a different prefix than **ns4**.

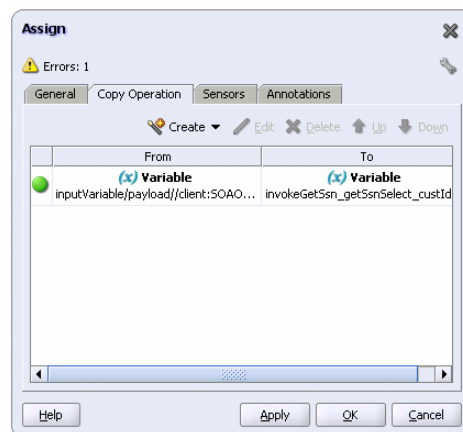


8. On the **To** side, expand **Variables > invokeGetSsn\_getSsnSelect\_custIdToGet\_InputVariable > getSsnSelect\_custIdToGet\_inparameters > ns32:getSsnSelect\_custIdToGetInputParameters**.
9. Select **ns32:custIdToGet**. Your system may show a different prefix than **ns32**.



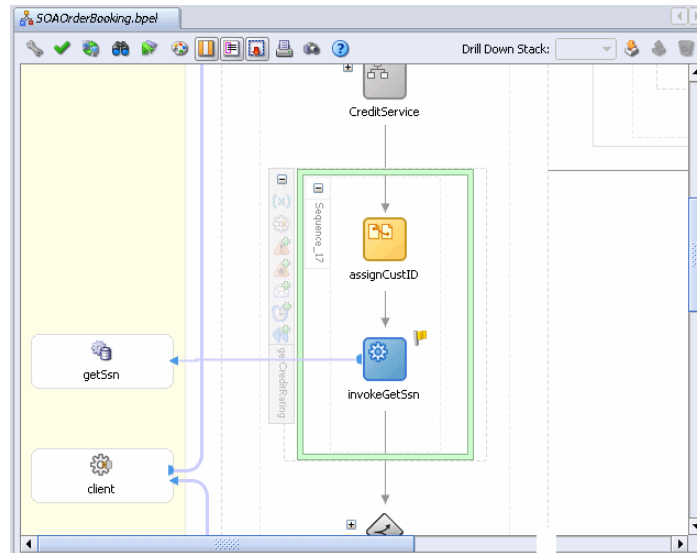
10. Click **OK**.

The Copy Operation tab in the Assign window updates to show the rule.



11. Back in the Assign window, scroll-down and click **OK**.

The SOAOrderBooking.bpel page should now look like the following:



**Summary:** You created a `getCreditRating` scope activity that takes the customer ID provided by the `SOAOrderBooking` process and assigns it to the input variable, `invokeGetSsn_getSsnSelect_custIdToGet_InputVariable`. The input variable is passed as input to the `getSsn` service, which in turn returns the Social Security number to the output variable, `invokeGetSsn_getSsnSelect_custIdToGet_OutputVariable`. Next, you will create a `CreditRatingService`, and assign this output variable as input to that Web service.

#### 4.2.6 Task 6: Install the Credit Rating Service

At this point in the procedure, you have modified the BPEL process to extract the customer's Social Security number from the database. The next step is to pass the Social Security number to the credit rating service, which returns the credit rating.

In this task, you install the credit rating service. The service is called `CreditRatingService` and is one of the samples shipped when you install Oracle SOA Suite. The following steps can also be used to install any of the many other samples that also shipped.

1. Select **Start > All Programs > Oracle - Oracle - soademo > Oracle BPEL Process Manager > Developer Prompt**.

A command prompt displays. Notice you are in the `samples` directory.

2. Enter the following command:

```
cd utils\CreditRatingService
```

3. Enter following command:

```
ant
```

This command runs the `ant` script to install the `CreditRatingService` service. It takes 30 to 60 seconds to run the script.

When you run `ant`, the script looks in the current directory for a `build.xml` file, which contains `ant` scripts for installing the sample. You can explore the `samples` directory and any directories under it. Anywhere there is a `build.xml` file, you can simply run `ant` to install that sample.

4. Enter `exit` to close the command window.

`CreditRatingService` is now implemented as a BPEL process. Whenever you create a BPEL process, it is exposed as a service. This means there is now a service called `CreditRatingService` that can be invoked by anything that can invoke a service. In the next task, you will invoke that service from the `SOAOrderBooking` process.

If you receive an authentication error when you run `ant`, it is probably because you are using a password other than `welcome1`. To resolve this issue:

1. Use a text editor and open:

```
ORACLE_HOME\bpel\utilities\ant-orabpel.properties
```

2. Modify the value of `admin.password` from `welcome1` to the password you are using.

### 4.2.7 Task 7: Create `CreditRatingService` Partner Link

In this task, you create a partner link to a Web service that maintains ratings for Social Security numbers.

1. Drag and drop **PartnerLink** into one of the yellow **Services** swim lanes of the `SOAOrderBooking.bpel` page.

The Create Partner Link dialog appears.

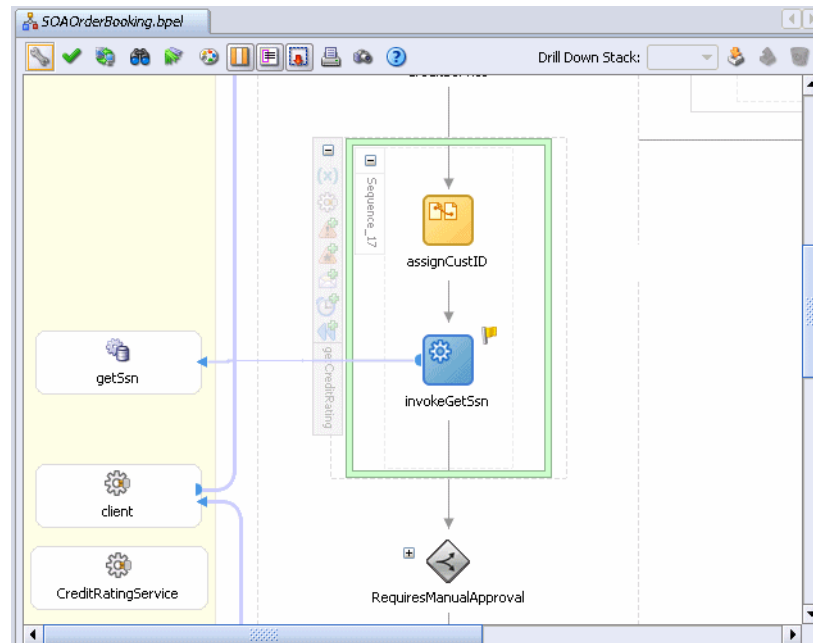
2. Provide values for the elements as follows:

Element	Value
Name	<code>CreditRatingService</code>
WSDL File	<ol style="list-style-type: none"><li>1. Click the <b>Service Explorer</b> flashlight icon.</li><li>2. In the Service Explorer dialog, expand <b>BPEL Services</b> &gt; <b>OrderBookingIS</b> &gt; <b>processes</b> &gt; <b>default</b>.</li><li>3. Select <b>CreditRatingService</b>.</li><li>4. Click <b>OK</b>.</li></ol>
Partner Link Type	<code>CreditRatingService</code>
Partner Role	<code>CreditRatingServiceProvider</code>
My Role	<code>unspecified</code>

3. Click **OK**.

The `SOAOrderBooking.bpel` page updates with the **CreditRatingService** partner link.



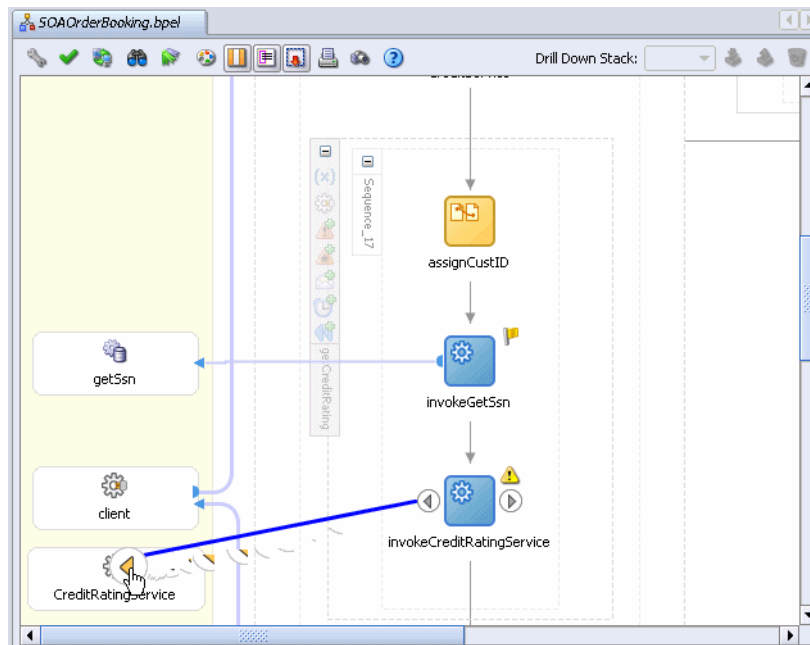


#### 4.2.8 Task 8: Create Invoke Activity to Call CreditRatingService

In this task, you create an invoke activity to send request data from the SOAOrderBooking process to the CreditRatingService partner link and receive a response.

To create the invoke activity:

1. Drag and drop an **Invoke** activity from the **Component Palette** section below the **invokeGetSsn** activity inside the **getCreditRating** scope activity.
2. Rename this activity by double-clicking name underneath the icon. Do not double-click the invoke icon itself.
3. In the edit field, change the name to **invokeCreditRatingService**.
4. Drag the mouse from the left side of **invokeCreditRatingService** to the **CreditRatingService** partner link.



The Edit Invoke dialog appears and is automatically filled in with the following information:

Element	Value
Name	invokeCreditRatingService
Partner Link	CreditRatingService
Operation	process

- Click the **Automatically Create Input Variable** icon. It is the first icon to the right of the **Input Variable** field.

The Create Variable dialog appears with the input variable. A variable named **invokeCreditRatingService\_process\_InputVariable** is automatically created in the **Name** field. This variable is automatically assigned a message type.

The variable name is based on the invoke activity (invokeCreditRatingService) and operation (process). This variable provides input data to the CreditRatingService service.

- Leave **Global Variable** selected, and click **OK**.

The Edit Invoke dialog populates with the variable in the **Input Variable** field.

- Click the **Automatically Create Output Variable** icon. It is the first icon to the right of the **Output Variable** field.

The Create Variable dialog appears with the output variable. A variable named **invokeCreditRatingService\_process\_OutputVariable** is automatically created in the **Name** field. This variable is automatically assigned a message type.

- Leave **Global Variable** selected, and click **OK**.

The Edit Invoke dialog populates with the variable in the **Output Variable** field.

This variable provides output data from the CreditRatingService service.

9. In the Edit Invoke dialog, click **OK** to save the variable settings.

#### 4.2.9 Task 9: Assign Data to the Input Variable for CreditServiceRating

You now create an assign activity to take the Social Security number from the `invokeGetSsn_getSsnSelect_custIdToGet_OutputVariable` variable and assign it to the input variable `invokeCreditRatingService_process_InputVariable`, which in turn sends a request to `CreditRatingService` for the Social Security number.

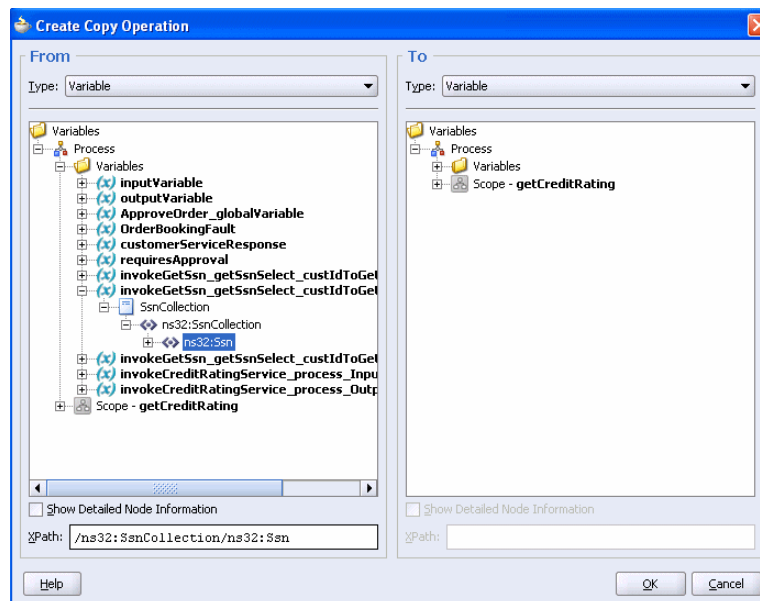
1. Drag and drop an **Assign** activity from the **Component Palette** section to above the `invokeCreditRatingService` invoke activity, but below the `invokeGetSsn` invoke activity.
2. Rename this activity by double-clicking name underneath the icon. Do not double-click the assign icon itself.
3. In the edit field, enter `assignSsn`.
4. Double-click the `assignSsn` activity.

The Assign window displays.

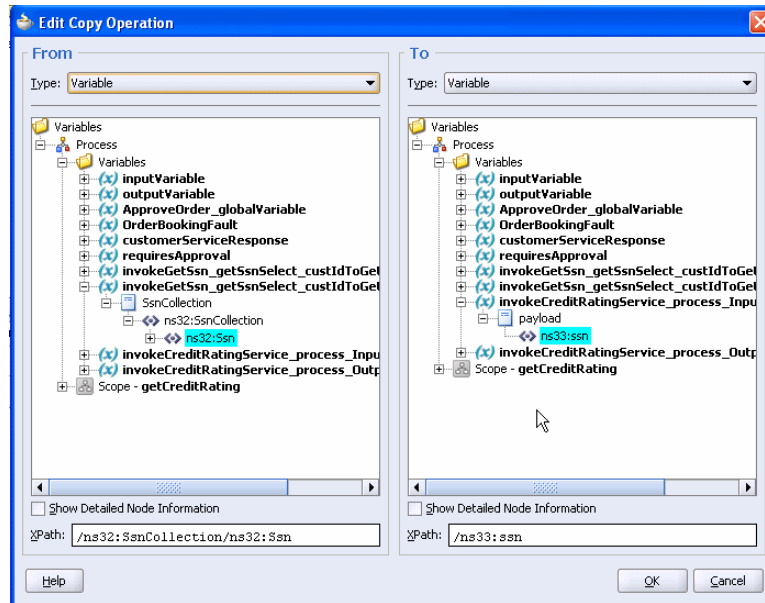
5. From the **Create** menu, select **Copy Operation**.

The Create Copy Operation dialog appears. Previously, you used the database adapter to retrieve the Social Security number from the database. To use that Social Security number and pass it to `CreditRatingService`, on the **From** side, you specify to take the Social Security number from the variable that was returned by `getSsn`.

6. On the **From** side, expand **Variables** > `invokeGetSsn_getSsnSelect_custIdToGet_OutputVariable` (of the database adapter) > `SsnCollection` > `ns32:SsnCollection`. Your system may show a different prefix than `ns32`.
7. Select `ns32:ssn`, where `ssn` represents the `ssn` column in the `SSN` table of the `SOADEMO` schema.

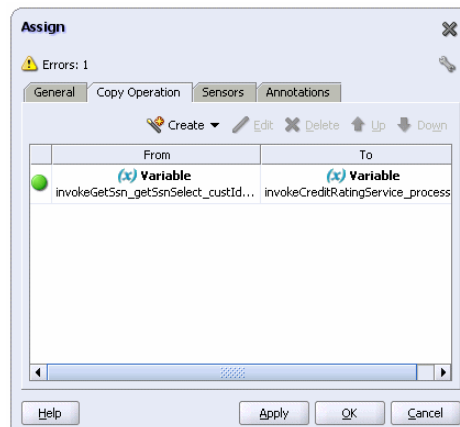


8. On the **To** side, expand **Variables** > `invokeCreditRatingService_process_InputVariable` > `payload`.
9. Select `ns33:ssn`. Your system may show a different prefix than `ns32`.



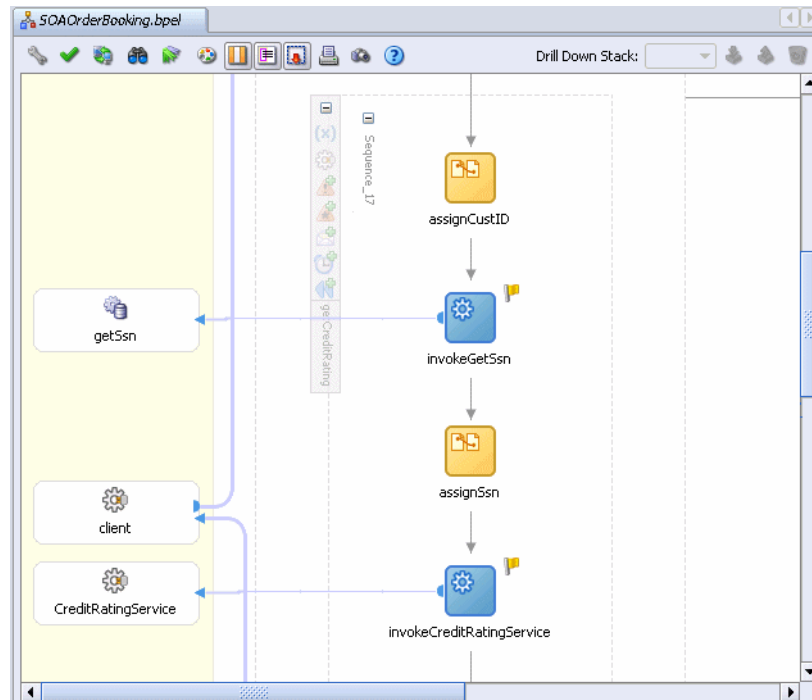
10. Click **OK**.

The Copy Operation tab in the Assign window updates to show the rule.



11. Back in the Assign window, scroll-down and click **OK**.

The SOAOrderBooking.bpel page should now look like the following:



12. Choose **File > Save** to save your work.

**Summary:** You created a `getCreditRating` scope activity that takes the customer ID provided by the `SOAOrderBooking` process and assigns it to the input variable, `invokeGetSsn_getSsnSelect_custIdToGet_InputVariable`. The input variable is passed as input to the `getSsn` service, which in turn returns the Social Security number to the output variable, `invokeGetSsn_getSsnSelect_custIdToGet_OutputVariable`. Next, you assigned this variable as input to a `CreditRatingService` Web service, which in turn returns the rating for the Social Security number to the output variable, `invokeCreditRatingService_process_OutputVariable`.

#### 4.2.10 Task 10: Redeploy SOAOrderBooking

Now that you have modified and saved the BPEL process, it is time to test it. To do that, deploy it to the server, and then place a new order, which will use the updated BPEL process.

To redeploy `SOAOrderBooking`:

1. In the Application Navigator, expand **SOAOrderBooking > Resources**.
2. In the Application Navigator, right-click `build.xml`, and select **Run Ant**.  
The Run Ant dialog appears.
3. Click the **Properties** tab, and if any properties exist, remove them.
4. In the **Property Files** section, click **Add** and select the `build.properties` file.
5. Click **Open** and click **OK**.

This action starts the deployment process, which may take anywhere from 30 to 60 seconds. You can monitor the progress in the **Messages** pane, in the **Apache Ant** -

**Log** tab, in the **Apache Ant** sub-tab. You will know the deployment is complete when you see the text **BUILD SUCCESSFUL**.

## 4.2.11 Task 11: Test New Functionality by Placing a New Order

To place an order in the Web client:

1. Click the **Browse products and create a new order** link.

The Browse and Select Items page appears. It lists the electronic products available for sale.

### Browse and Select Items

Select and <a href="#">View Details</a>						
		<a href="#">Previous</a>		1-10 of 27	<a href="#">Next 10</a> <a href="#">»</a>	
Select	name	prodid	description	category	listPrice	producer
<input type="radio"/>	HD Television	128723	A High-Definition Flat Screen Television that uses DLP Technology.	Video	\$1999.00	Azimuth
<input checked="" type="radio"/>	PlayStation 2 Video Game System	223453	Playstation II Interactive Video Gaming System.	Games	\$199.00	Playstation
<input type="radio"/>	Treo 650 Phone/PDA	429832	A fully featured Treo Phone and PDA combination.	Handhelds	\$299.00	Treo
<input type="radio"/>	Treo 700w Phone/PDA	423322	A fully featured Treo Phone and Windows CE Handheld.	Handhelds	\$399.00	Treo
<input type="radio"/>	Tungsten E PDA	592039	Tungsten handheld Personal Digital Assistant.	Handhelds	\$195.00	Tungsten
<input type="radio"/>	XBox Video Game System	621135	XBox Interactive Video Gaming System	Games	\$159.00	XBox
<input type="radio"/>	XBox 360 Video Game System	218407	Next generation XBox 360 Interactive Video Gaming System.	Games	\$299.00	XBox
<input type="radio"/>	Playstation Portable	124039	Playstation Portable HandHeld Video Gaming System.	Handhelds	\$199.00	MyTag
<input type="radio"/>	Nintendo DS	606945	Nintendo DS Handheld Gaming System	Handhelds	\$129.00	DS
<input type="radio"/>	Razer Cellular Phone	927384	Ultra-sleek Razer Cellular Phone	Handhelds	\$229.00	Razer

2. Select the PlayStation 2 Video Game System, priced at \$199.00, and click **View Details**.

The Item Details page appears. It displays detailed information about the product, and enables the user to select a quantity to add to their cart.

### Item Details

Item Details	
name	PlayStation 2 Video Game System
description	Playstation II Interactive Video Gaming System.
category	Games
listPrice	\$199.00
prodid	223453
producer	Playstation
<a href="#">Continue Shopping</a>	

**Add to cart?**

Quantity?

[Add to cart](#)

[Go to Shopping Cart](#)

3. In the **Quantity** list, select 1, and click **Add to cart**.
4. Click **Go to Shopping Cart**.

The Shopping Cart Contents page appears.

### Shopping Cart Contents

Product Name	Part No.	Category	Price	Quantity
PlayStation 2 Video Game System	223453	Games	\$199.00	1
<a href="#">Empty Shopping Cart</a>		<a href="#">Continue Shopping</a>	<a href="#">Place Order</a>	Order Total: \$199.00

5. Click **Place order** to submit the order.

The welcome page updates with an `Order Submitted` message.

SOA Release Demo Web Client

Information

Order Submitted!

4.2.12 Task 12: View the Order Approval in the Oracle BPEL Control

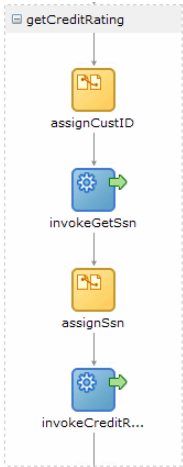
- 1. If the Oracle BPEL Control is not running, select **Start > All Programs > Oracle - Oracle - soademo > Oracle BPEL Process Manager > BPEL Control**.
- 2. When prompted, enter `oc4jadmin` in the **Username** field and `welcome1` in the **Password** field.

The **Dashboard** tab updates to show the order has completed.

ORACLE Enterprise Manager 10g BPEL Control			
		Manage BPEL Domain	Logout   Support
		Logged to domain: default	
Dashboard		BPEL Processes	Instances
Activities			
Deployed BPEL Processes		In-Flight BPEL Process Instances	
Name		Instance	BPEL Process
CreditRatingService			
DHLShipment			
SOAOrderBooking			
SelectManufacturer			
TaskActionHandler			
TaskManager			
		Recently Completed BPEL Process Instances (More...)	
		101 : Instance #101 of SOAOrderBooking	SOAOrderBooking (v. 1.0) 8/9/06 5:18:25 PM
		103 : Instance #103 of SelectManufacturer	SelectManufacturer (v. 1.0) 8/9/06 5:18:23 PM
		102 : Instance #102 of CreditRatingService	CreditRatingService (v. 1.0) 8/9/06 5:18:17 PM
		7 : Instance #7 of SelectManufacturer	SelectManufacturer (v. 1.0) 8/9/06 4:17:10 PM
		5 : Instance #5 of SOAOrderBooking	SOAOrderBooking (v. 1.0) 8/9/06 4:17:08 PM
Deploy New Process		Oracle BPEL Console v10.1.3.1.0	

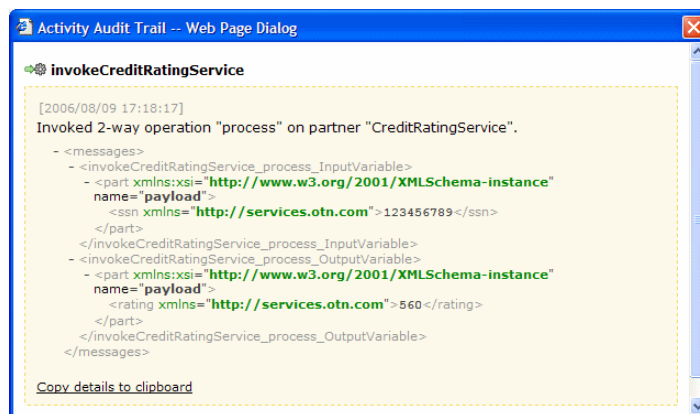
Notice the addition of the **CreditRatingService**, which you just created.

- 3. Click the **Instances** tab.
- 4. Click the last instance of **SOAOrderBooking**.
- 5. Click the **Flow** subtab, and the scroll to the **getCreditRating** scope.



6. Click **invokeCreditRatingService**.

The Activity Audit Trail window displays. It shows Social Security number (**ssn**) 123456789 was returned for customer ID (**customerid**) 10 from the previous activity. For this activity, it shows CreditRatingService returned a rating of 560.



7. Close the Activity Audit Trail window.

8. Click the **Dashboard** tab to navigate back to the main view.

## 4.3 Introduction to the JDeveloper ESB Designer

This section provides a brief introduction to the JDeveloper ESB Designer. In the next section, [Section 4.4](#), you will make modifications to the FulfillmentESB project.

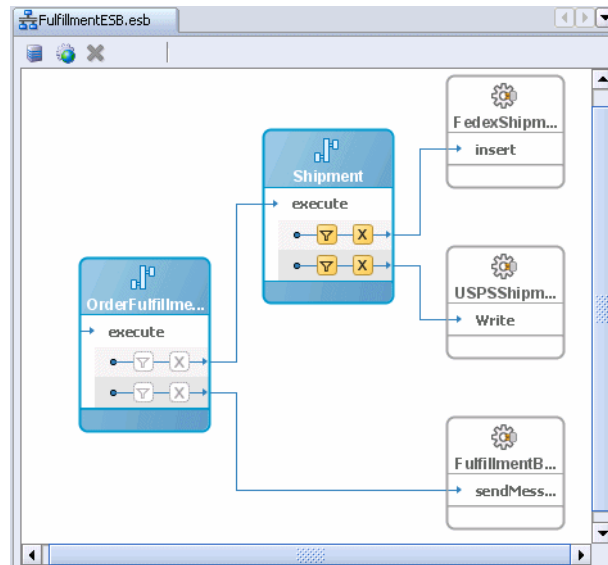
To familiarize yourself with the JDeveloper ESB Designer:

1. In the Applications Navigator of JDeveloper, expand **SOADEMO > FulfillmentESB > Resources**.
2. Double-click **FulfillmentESB.esb**.

The JDeveloper ESB Designer displays the following in the flow:

- **OrderFulfillment** routing service, which routes messages to **Shipment** and **FulfillmentBatch**
- **Shipment** routing service, which is responsible for sending shipment information to either USPS (**USPSShipment**) or Fedex (**FedexShipment**).
- **USPSShipment** file adapter
- **FedexShipment** database adapter
- **FulfillmentBatch** JMS adapter, which is responsible for storing all fulfillment orders for overnight batch processing





3. In **Shipment**, hover your mouse over the filter icons.

The first filter shows the following for orders to be shipped by FedEx:

```
/inpl:PurchaseOrder/inpl:OrderInfo/inpl:OrderPrice >= 500
```

The second filter shows the following for orders to be shipped by USPS

```
/inpl:PurchaseOrder/inpl:OrderInfo/inpl:OrderPrice < 500
```

As described and shown in previous chapters, after an order has been approved, Fulfillment sends the order to the appropriate shipping target. Orders under \$500 are routed to USPS; orders \$500 and over are routed to FedEx using these filters. USPS orders are sent through a file adapter and written to a file. FedEx orders are sent through database adapter and written to the FEDEX table in the database.

## 4.4 Adding a New Shipping Target to FulfillmentESB Project

To familiarize yourself with designing ESB elements in the JDeveloper ESB Designer, you will create a third shipping target for DHL that uses a SOAP service for orders over \$1000.

In this scenario, you will perform the following tasks:

- [Task 1: Add the DHLShipment Service to ESB](#)
- [Task 2: Redeploy FulfillmentESB](#)
- [Task 3: Test New Functionality By Placing a New Order](#)
- [Task 4: View the Order in the Oracle BPEL Control](#)
- [Task 5: View the DHLShipment Status in the Oracle ESB Control](#)

### 4.4.1 Task 1: Add the DHLShipment Service to ESB

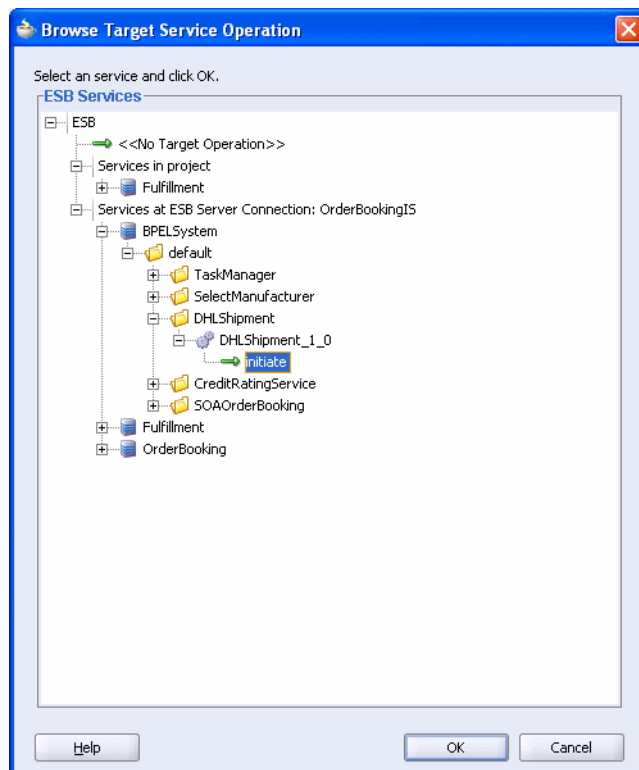
To create the DHLShipment service:

1. Double-click the **Shipment** routing service

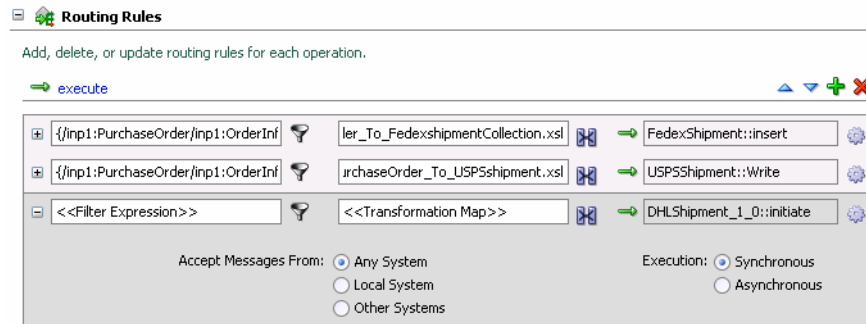
The Fulfillment\_Shipment.esbsvc page displays with **Routing Service** information.



2. In the **Routing Rules** section, click the **Expand the Routing Rule** icon (+) sign to expand it.
3. Scroll to the far right, and click the green + icon to add another routing rule.  
The **Browse Target Service Operation** dialog appears.
4. Expand **Services at ESB Server Connection: OracleBookingIS > BPEL System > default > DHLShipment > DHLShipment\_1\_0**.  
DHLShipment is a service that you deployed in [Chapter 2](#).
5. Under **DHLShipment\_1\_0**, click **initiate**.



6. Click **OK** in the **Browse Target Service Operation** dialog.  
The **Routing Rules** section now looks like this:



7. Create a filter to direct orders \$1000 and over to DHLShipment.

- a. Click the filter icon for the DHLShipment rule.

The Expression Builder dialog appears

- b. In the **WSDL Message** section, expand **PurchaseOrder** > **inp1:PurchaseOrder** > **inp1:OrderInfo**, and then select **inp1:OrderPrice**.

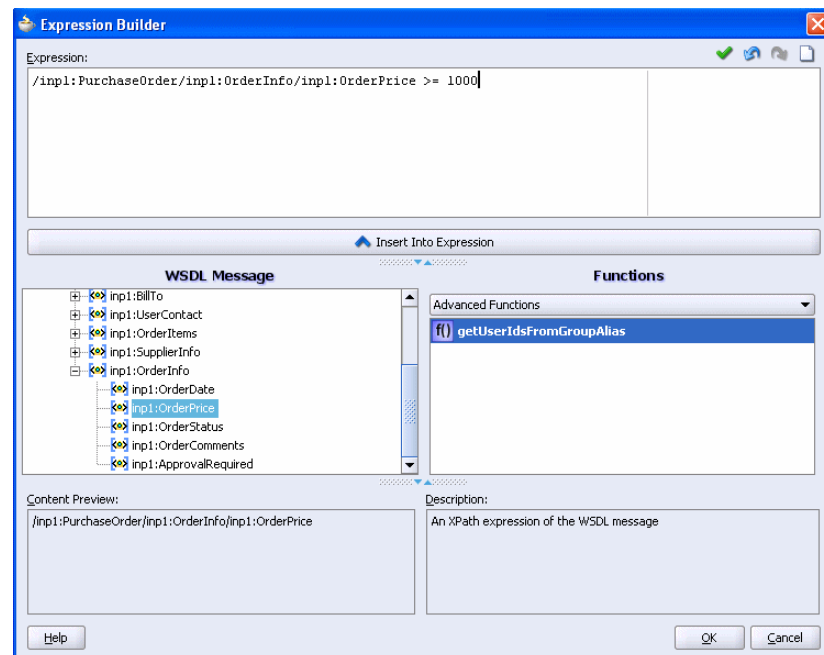
The Content Preview box also shows the path:

/inp1:PurchaseOrder/inp1:OrderInfo/inp1:OrderPrice

- c. Click **Insert Into Expression**.

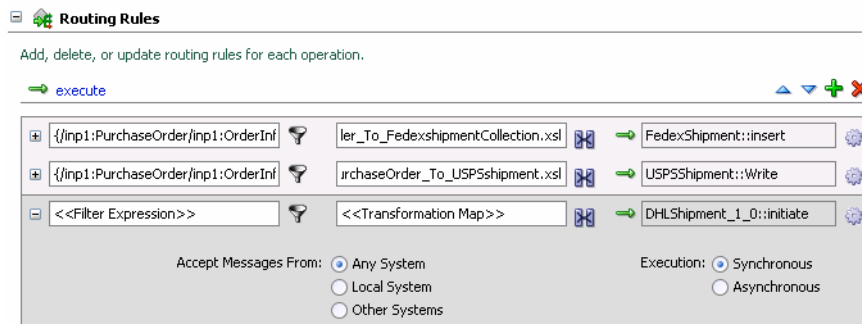
The path appears in the **Expression** box at the top of the dialog.

- d. In the **Expression** box, append `>= 1000` to the path, so that it now reads `/inp1:PurchaseOrder/inp1:OrderInfo/inp1:OrderPrice >= 1000`.

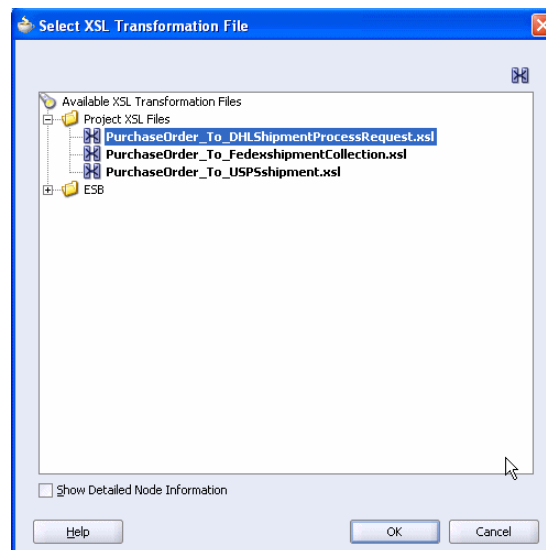


- e. Click **OK** in the Expression Builder.

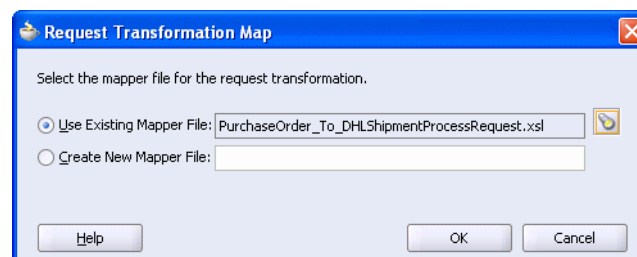
The **Routing Rules** section now looks like this:



8. Create a transformation so that the DHLShipment SOAP service gets the proper information:
  - a. In the **Routing Rules** section, click the transformation icon.  
The Request Transformation Map dialog appears.
  - b. Select **Use Existing Mapper File**, and click the flashlight icon.  
The Select XSL Transformation File dialog appears.
  - c. Expand **Project XSL Files**, and select **PurchaseOrder\_To\_DHLShipmentProcessRequest.xsl**.

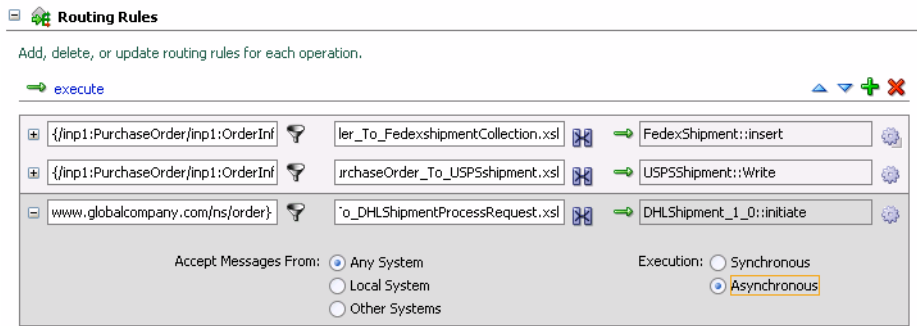


- d. Click **OK** to return back to the Request Transformation Map dialog.



- e. In the Request Transformation Map dialog, click **OK**.
- f. Scroll to the far lower right, and click **Asynchronous**.

The **Routing Rules** section now looks like this:

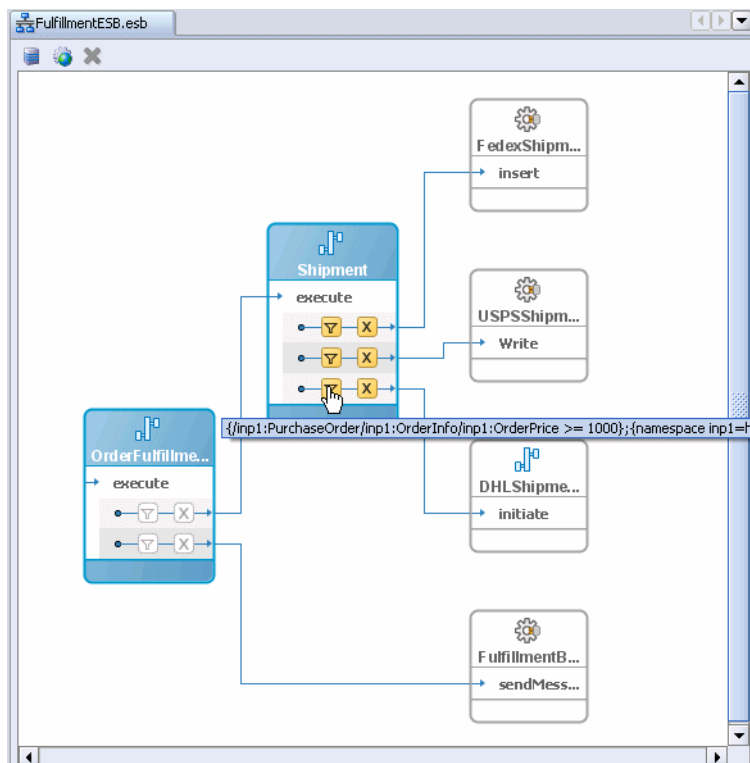


- g. Select **File > Save** to save your work.
- h. Close the Fulfillment\_Shipment.esbsvc window by hovering your mouse over the tab and pressing **X**.

The FulfillmentESB.esb page updates. From **Shipment**, you should now see three arrows: one going to **FedexShipment**, a second one going to **USPSShipment**, and a third one going to **DHLShipment**.

If you hover your mouse over the filter icon third filter in **Shipment**, it shows the filter used for orders to be shipped by DHL:

```
/inp1:PurchaseOrder/inp1:OrderInfo/inp1:OrderPrice >= 1000
```



## 4.4.2 Task 2: Redeploy FulfillmentESB

Now that you have modified and saved the **FulfillmentESB**, it is time to test it. To do that, deploy it to the server, and then place a new order for over \$1000 to see if it is sent DHL.

To redeploy **FulfillmentESB**:

1. In the Application Navigator, right-click **FulfillmentESB**, and select **Register to ESB > OrderBookingIS**.
2. Click **OK** in the Summary dialog.

## 4.4.3 Task 3: Test New Functionality By Placing a New Order

To place an order in the Web client:

1. In the welcome page, click the **Browse products and create a new order** link.  
The Browse and Select Items page appears.
2. Click the **Next 10** link, and select **Ipod Mini 2 Gb**.

### Browse and Select Items

Select and		View Details		Previous 10		11-20 of 27		Next 7	
Select	name	prodid	description	category	listPrice	producer			
<input type="radio"/>	Muvo Personal MP3 Player	579823	Miniature 4gig Steroo MP3 Player	Handhelds	\$99.00	Muvo			
<input type="radio"/>	Bluetooth Adaptor	427923	Multi-purpose Mini BlueTooth Adaptor	Wireless	\$19.00	Bluetooth			
<input type="radio"/>	Bluetooth Phone Headset	298332	Wireless Bluetooth Phone Headset	Wireless	\$49.00	Bluetooth			
<input type="radio"/>	Ipod Speakers	183923	Ipod Speaker Addon System	Audio	\$89.00	Ipod			
<input checked="" type="radio"/>	Ipod Mini 2Gb	532211	2Gb Ipod Mini Personal Audio Player	Audio	\$139.00	Ipod			
<input type="radio"/>	Ipod Shuffle 512Mb	545125	512Mb Ipod Shuffle Personal Audio Player	Audio	\$69.00	Ipod			
<input type="radio"/>	Ipod Shuffle 1Gb	547321	1Gb Ipod Shuffle Personal Audio Player	Audio	\$99.00	Ipod			
<input type="radio"/>	Ipod Video 30Gb	542122	30Gb Ipod Personal Audio and Video Player	Audio	\$299.00	Ipod			
<input type="radio"/>	Ipod Video 60Gb	547642	60Gb Ipod Personal Audio and Video Player	Audio	\$399.00	Ipod			
<input type="radio"/>	Ipod Nano 1Gb	392784	A portable media entertainment device.	Audio	\$149.00	Ipod			

3. Click **View Details**.

The Item Details page appears. It displays detailed information about the product, and enables the user to select a quantity to add to their cart.

### Item Details

Item Details	
name	Ipod Mini 2Gb
description	2Gb Ipod Mini Personal Audio Player
category	Audio
listPrice	\$139.00
prodid	532211
producer	Ipod
<input type="button" value="Continue Shopping"/>	

Add to cart?	
Quantity?	10
<input type="button" value="Add to cart"/>	
<a href="#">Go to Shopping Cart</a>	

4. In the **Quantity** list, select 10, and click **Add to cart**.
5. Click **Go to Shopping Cart**.

The Shopping Cart Contents page appears.

**Shopping Cart Contents**

Product Name	Part No.	Category	Price	Quantity
Ipod Mini 2Gb	532211	Audio	\$139.00	10
Empty Shopping Cart		Continue Shopping	Place Order	Order Total: \$1,390.00

6. Click **Place order** to submit the order.

The welcome page updates with an Order Submitted message.

**SOA Release Demo Web Client**

<a href="#">Information</a> Order Submitted!
---

**4.4.4 Task 4: View the Order in the Oracle BPEL Control**

1. If the Oracle BPEL Control is not running, select **Start > All Programs > Oracle - Oracle - soademo > Oracle BPEL Process Manager > BPEL Control**.
2. When prompted, enter `oc4jadmin` in the **Username** field and `welcome1` in the **Password** field.
3. The **Dashboard** tab shows the order completed.

In [Section 3.2, "Increasing Approval Order Amount from \\$1000 to \\$2000 and Resubmitting Order"](#) on page 3-16, you changed the approval amount. Therefore, the order should complete without approval. If you did not complete [Section 3.2](#), then you must use the Worklist Application to approve the order. See [Section 3.2](#) for further information on using the Worklist Application.

**4.4.5 Task 5: View the DHLShipment Status in the Oracle ESB Control**

1. If the Oracle ESB Control is not running, select **Start > All Programs > Oracle - Oracle - soademo > Oracle ESB > ESB Control**.
2. When prompted, enter `oc4jadmin` in the **Username** field and `welcome1` in the **Password** field.
3. Click the **Instances** icon, at the top of the screen, toward the right-hand side.



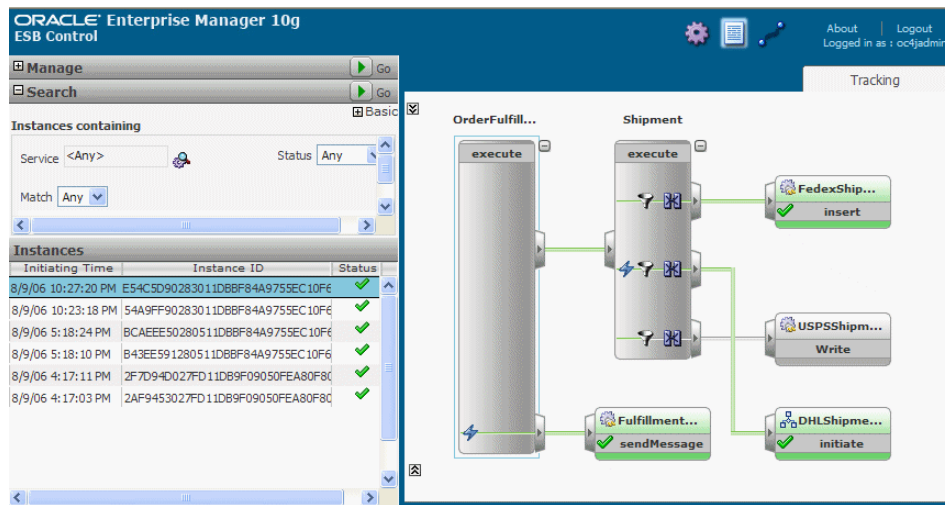
The **Instances** pane appears on the left-hand side. The **Status** column shows the **OrderBooking** and **Fulfillment** instances were successfully completed.

Initiating Time	Instance ID	Status
8/9/06 10:27:20 PM	E54C5D90283011DBBF84A9755EC10F60	✓
8/9/06 10:23:18 PM	54A9FF90283011DBBF84A9755EC10F60	✓
8/9/06 5:18:24 PM	BCAEEE50280511DBBF84A9755EC10F60	✓
8/9/06 5:18:10 PM	B43EE591280511DBBF84A9755EC10F60	✓
8/9/06 4:17:11 PM	2F7D94D027FD11DB9F09050FEA80F803	✓
8/9/06 4:17:03 PM	2AF9453027FD11DB9F09050FEA80F803	✓

- Click the instance at the top of the list.

This view shows the **Shipment** service routed the over \$1000 order to the new **DHLShipment** service, as shown by the green line. However, also notice the green line extends from **Shipment** to **FedexShipment**. In other words, the product will be shipped by two different shipment vendors. This quick start did not ask you to adjust the routing rule for FedEx, which accepts orders for \$500 or above. Therefore, you need to be careful when adding ESB elements to avoid unexpected results.

In a real-world enterprise, you would need to alter the filter rule for FedEx, so orders \$500 and up to \$1000 are routed to FedEx. By making this change, orders for \$1000 and over would route only to DHL. To gain further experience with ESB, feel free to make this change yourself.

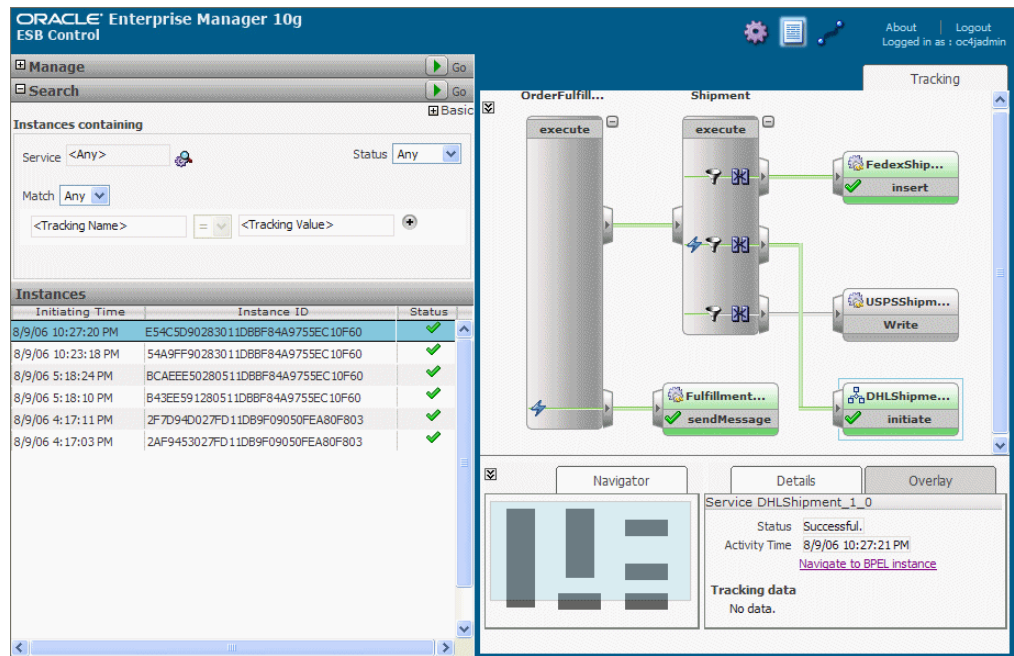


- At the bottom of the **Fulfillment** instance, click the **Expand** icon.



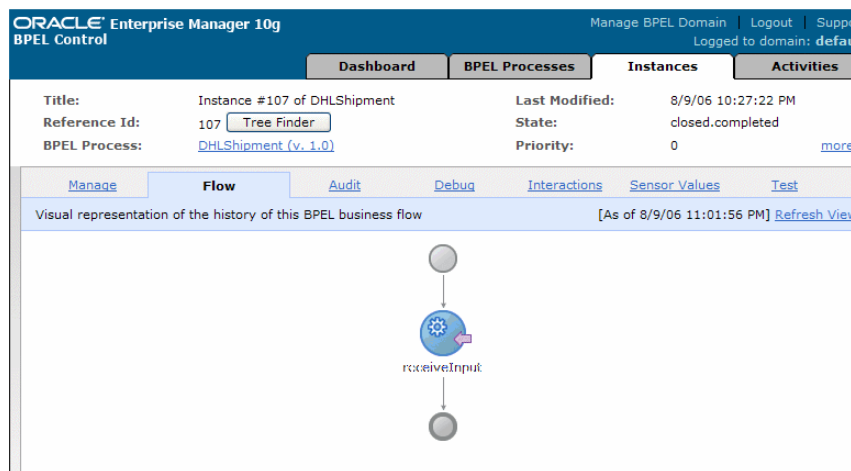
- Click the **DHLShipment** service.





- Click the **Navigate to BPEL instance** link to see the integration with the Oracle BPEL Control.

The Oracle BPEL Control displays with the **Flow** view for the DHLShipment service.



## 4.5 Learning More About Oracle SOA Suite

To learn about Oracle SOA Suite, refer to the following resources:

- Oracle's Service-Oriented Architecture Technology Center:  
<http://www.oracle.com/technology/soa>
- Oracle Application Server Tutorial* for a step-by-step approach for building the SOA Order Booking application yourself
- Oracle SOA Suite Developer's Guide* for an in-depth description of designing and developing an SOA application, using the SOA Order Booking application as an example



---

---

# Index

## A

---

### activities

- assign, 4-4
- invoke, 3-6
- receive, 3-5
- scope, 3-6
- switch, 3-12

ADDRESS table, 2-14

ant script, 4-17

application server connections, 2-7

ApproveOrderRequired rule set, 3-17

ApproveOrder.task file, 4-6

assign activities, 4-4

audit trail, viewing in Oracle BPEL Control, 3-6

## B

---

bpel.xml file, 2-11

build.xml file, 2-10

### business rules

- defined, 3-17
  - Oracle BPEL Process Manager, used with decision services, 3-7
  - SOA Order Booking application, used in, 1-8
- business rules, rule sets and, 3-17

## C

---

connection factories, 2-5

connection pools, 2-4

createSchemaObjects.sql script, 2-3

CreditRatingService Web service, 4-9

CreditService application

- deploying, 2-10
- introduced, 1-8

CreditService scope, 3-7, 4-9

CUSTOMER table, 2-14

CUSTOMER\_ADDRESS table, 2-14

CustomerService application

- deploying, 2-10
- introduced, 1-8

CustomerService scope, 3-7, 4-3

## D

---

Dashboard tab, Oracle BPEL Control, 3-4

data sources, 2-4

database connections, 2-7

decision services, 1-8, 3-7

Definitions tab, Oracle Business Rules Rule

Author, 3-19

### deploying

applications for the SOA Order Booking application, 2-8

CreditService application, 2-10

CustomerService application, 2-10

DHLShipment SOAP service, 2-12

FulfillmentESB project, 2-9

OrderBookingESB project, 2-13

RapidService application, 2-11

SelectManufacturer, 2-10

SOADEMO-CLIENT, 2-13

SOAOrderBooking process, 2-11

DHLShipment SOAP service

adding to FulfillmentESB flow, 4-27

deploying, 2-12

introduced, 1-5

## E

---

EJB\_TAB\_ID\_GEN table, 2-14

## F

---

FedexShipment database adapter, 3-15, 4-26

FEDEXSHIPMENT table, 2-15

FulfillmentBatch routing service, 3-15, 4-26

FulfillmentESB project

deploying, 2-9

DHLShipment SOAP service, 4-27

FedexShipment database adapter, 3-15, 4-26

FulfillmentBatch JMS adapter, 3-15, 4-26

introduced, 1-8

OrderFulfillment routing service, 4-26

Shipment routing service, 4-26

USPSShipment file adapter, 3-15, 4-26

Fullfillent\_Shipment.esbsvc file, 4-27

## G

---

getCreditRating scope, 4-13

## I

---

- InsertOrderIntoDB scope, 3-6
- Instances icon, Oracle ESB Control, 3-15
- Instances tab, Oracle BPEL Control, 3-4
- Integrated Service Environment
  - introduced, 1-2
  - Oracle ADF, 1-2
  - Oracle JDeveloper, 1-2
  - Oracle TopLink, 1-2
- integration server connections, 2-8
- invoke activities, 3-6
- ITEMS table, 2-14

## J

---

- jcooper user, 3-1
- JDBC data sources, 2-4

## M

---

- My Tasks tab, Oracle BPEL Worklist Application, 3-8

## N

---

- Navigate to BPEL Instance link, 4-35
- NotifyCustomer scope, 3-13

## O

---

- Oracle Application Server
  - introduced, 1-5
- Oracle BPEL Control
  - audit trail, 3-6
  - Dashboard tab, 3-4
  - Flow view of Instances tab, 3-5
  - Instances tab, 3-4
  - introduced, 3-4
- Oracle BPEL Process Manager
  - business rules, 3-7
  - decision services in, 1-8, 3-7
  - introduced, 1-3
  - Oracle BPEL Worklist Application, 3-1
  - switch activities in, 3-12
- Oracle BPEL Worklist Application
  - approving orders with, 3-8
  - introduced, 3-1
  - My Tasks tab, 3-8
- Oracle Business Activity Monitoring
  - introduced, 1-4
- Oracle Business Rules
  - business rules, 3-17
  - introduced, 1-4
  - Oracle Business Rules Rule Author, 3-17
  - rule sets, 3-17
- Oracle Business Rules Rule Author
  - Definitions tab, 3-19
  - introduced, 3-17
  - Repository tab, 3-17
  - Rulesets tab, 3-17

- Oracle Enterprise Manager Application Server Control
  - configuring Oracle SOA Suite, 2-3
  - determining port numbers, 2-6
- Oracle Enterprise Service Bus
  - configuring ports for the SOA Order Booking application, 2-13
  - introduced, 1-3
- Oracle ESB Control
  - Instances icon, 3-15
  - Instances view, 3-15
  - introduced, 3-14
  - Navigate to BPEL Instance link, 4-35
  - Services view, 3-14
  - viewing DHLShipment SOAP service, 4-33
- Oracle JDeveloper BPEL designer
  - adding a new scope, 4-9
  - introduced, 4-1
- Oracle JDeveloper ESB Designer
  - adding the DHLShipment SOAP service, 4-27
  - introduced, 4-26
- Oracle SOA Suite
  - architectural overview, 1-2
  - components, 1-2
  - introduced, 1-1
- Oracle Web Services Manager
  - introduced, 1-4
- OracleAS UDDI Registry
  - introduced, 1-5
- OrderBookingESB project
  - deploying, 2-13
  - introduced, 1-7
- OrderFulfillment routing service, 4-26
- ORDERS table, 2-14

## P

---

- populateSchemaTables.sql script, 2-3
- port numbers, determining with Application Server Control, 2-6
- PostFulfillmentReq scope, 3-13
- PRODUCT table, 2-14

## R

---

- RapidService application
  - deploying, 2-11
  - introduced, 1-8
- receive activities, 3-5
- Repository tab, Oracle Business Rules Rule Author, 3-17
- rule sets
  - ApproveOrderRequired, 3-17
  - business rules and, 3-17
- Rulesets tab, Oracle Business Rules Rule Author, 3-17

## S

---

- sample\_repository file, 3-17
- scope activities, 3-6

- SelectManufacturer process
  - deploying, 2-10
  - introduced, 1-8
- SelectSupplier scope, 3-10
- sequences, 4-3
- Service-Oriented Architecture
  - benefits of, 1-1
  - overview, 1-1
- Services view, Oracle ESB Control, 3-14
- SetFinalOrderStatus scope, 3-13
- Shipment routing service, 4-26
- Simple Object Access Protocol, 1-1
- sking user, 3-1
- SOA Order Booking application
  - ADDRESS table, 2-14
  - application server connections for, 2-7
  - business rules, used in, 1-8
  - connection pool for, 2-4
  - CreditRatingService Web service, 4-9
  - CreditService scope, 3-7, 4-9
  - CUSTOMER table, 2-14
  - CUSTOMER\_ADDRESS table, 2-14
  - CustomerService scope, 3-7, 4-3
  - database connection factory for, 2-5
  - database connections for, 2-7
  - database schema
    - installing, 2-3
  - decision services in, 1-8
  - deploying, 2-8
  - EJB\_TAB\_ID\_GEN table, 2-14
  - ESB ports, configuring, 2-13
  - FEDEXSHIPMENT table, 2-15
  - getCreditRating scope, 4-13
  - InsertOrderIntoDB scope, 3-6
  - installing, 2-1
  - installing the schema, 2-3
  - integration server connections for, 2-8
  - introduced, 1-5
  - ITEMS table, 2-14
  - jcooper user, 3-1
  - JDBC data source for, 2-4
  - NotifyCustomer scope, 3-13
  - ORDERS table, 2-14
  - port numbers, determining, 2-6
  - PostFulfillmentReq scope, 3-13
  - predefined users for, 2-15
  - PRODUCT table, 2-14
  - SelectSupplier scope, 3-10
  - SetFinalOrderStatus scope, 3-13
  - sking user, 3-1
  - SSN table, 2-14
  - switch activities, 3-12
  - URL for launching, 2-15
- SOADEMO schema, 2-14
- soademo\_101310\_prod.zip file, 2-2
- SOADEMO-CLIENT application
  - deploying, 2-13
  - introduced, 1-7
  - ordering, 3-2
- SOAOrderBooking process

- deploying, 2-11
  - introduced, 1-7
- SSN table, 2-14
- switch activities, 3-12

## U

---

- USPSShipment file adapter, 3-15, 4-26

## W

---

- Web services, 1-1
- Web Services Description Language, 1-1

