

Oracle® BPEL Process Manager

Order Booking Tutorial

10g (10.1.3.1.0)

B28987-01

September 2006

Oracle BPEL Process Manager Order Booking Tutorial, 10g (10.1.3.1.0)

B28987-01

Copyright © 2005, 2006, Oracle. All rights reserved.

Primary Author: Mark Kennedy

Contributor: SrinivasaGandhi Sampath, Sanjeev Sondur

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

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

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

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

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

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

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

Contents

| | |
|--|------|
| Preface | ix |
| Audience | ix |
| Documentation Accessibility | ix |
| Related Documents | x |
| Conventions | x |
| | |
| 1 Introduction | |
| Order Booking Introduction | 1-1 |
| Business Problem | 1-1 |
| Business Goal | 1-2 |
| Business Solution | 1-2 |
| Using this Tutorial | 1-4 |
| Installing Oracle Application Server SOA Suite | 1-4 |
| Identifying Ports | 1-5 |
| Installing Oracle JDeveloper | 1-5 |
| Starting Oracle BPEL Process Manager Components | 1-5 |
| Creating Oracle Application Server and Integration Server Connections | 1-6 |
| Setting the Hostname in Your Oracle JDeveloper Web Browser Preferences | 1-7 |
| | |
| 2 Invoking a Synchronous Service | |
| Introduction | 2-1 |
| Invoking a Synchronous Service to Receive an Order and Obtain a Credit Rating | 2-1 |
| Starting Oracle BPEL Server and Oracle JDeveloper | 2-2 |
| Starting and Testing Your Services | 2-2 |
| Creating an Application | 2-3 |
| Creating the Order Booking Project and Importing the Schema | 2-3 |
| Validating Your Project During the Design Phase | 2-6 |
| Creating a Credit Rating Service Partner Link | 2-7 |
| Creating Scope and Invoke Activities | 2-9 |
| Creating an Initial Assign Activity to Send a Request to the Credit Rating Service | 2-11 |
| Creating a Second Assign Activity to Receive a Response from the Credit Rating Service | 2-13 |
| Creating a Third Assign Activity | 2-15 |
| Validating, Compiling, and Deploying the Order Booking Process | 2-17 |
| Running the Order Booking Process | 2-18 |
| Summary | 2-20 |

3 Invoking an Asynchronous Service

| | |
|---|-----|
| Introduction | 3-1 |
| Invoking an Asynchronous Service to Receive the Supplier Price for the Order | 3-1 |
| Starting and Testing Your Services..... | 3-2 |
| Creating a Rapid Distributors Partner Link..... | 3-2 |
| Creating Invoke and Receive Activities..... | 3-3 |
| Creating an Assign Activity..... | 3-5 |
| Validating, Compiling, and Deploying the Order Booking Process..... | 3-7 |
| Running the Order Booking Process..... | 3-8 |
| Summary | 3-9 |

4 Adding Parallel Flow

| | |
|---|------|
| Introduction | 4-1 |
| Adding Parallel Flow to the Order Booking Tutorial | 4-1 |
| Starting and Testing Your Services..... | 4-2 |
| Creating a Select Manufacturing Partner Link..... | 4-2 |
| Creating a Flow Activity..... | 4-3 |
| Creating Invoke and Receive Activities..... | 4-4 |
| Creating an Assign Activity..... | 4-6 |
| Validating, Compiling, and Deploying the Order Booking Process..... | 4-8 |
| Running the Order Booking Process..... | 4-9 |
| Summary | 4-11 |

5 Adding Conditional Branching Logic

| | |
|--|-----|
| Introduction | 5-1 |
| Adding Conditional Branching Logic to the Order Booking Tutorial | 5-1 |
| Creating a Switch Activity..... | 5-2 |
| Creating an Assign Activity Under the Case Block for the Rapid Distributors Service..... | 5-2 |
| Creating an Assign Activity Under the Otherwise Block for Select Manufacturing..... | 5-4 |
| Validating, Compiling, and Deploying the Order Booking Process..... | 5-5 |
| Running the Order Booking Process..... | 5-6 |
| Summary | 5-6 |

6 Creating Fault Handling and Exception Management

| | |
|---|-----|
| Introduction | 6-1 |
| Creating Fault Handling and Exception Management | 6-1 |
| Creating a Catch All Branch in the Scope Activity..... | 6-1 |
| Validating, Compiling, and Deploying the Order Booking Process..... | 6-3 |
| Running the Order Booking Process..... | 6-3 |
| Summary | 6-5 |

7 Adding Transformation Logic

| | |
|--|-----|
| Introduction | 7-1 |
| Adding Transformation Logic | 7-1 |
| Creating the POAcknowledge Project and Importing the Schema..... | 7-1 |

| | |
|---------------------------------------|------------|
| Creating a Transform Activity | 7-3 |
| Testing the Transformation Logic..... | 7-8 |
| Summary | 7-9 |

8 Using the File Adapter’s Read Functionality

| | |
|---|------------|
| Introduction..... | 8-1 |
| Designing the File Adapter’s Read Functionality to Activate a BPEL Process | 8-1 |
| Creating a Batch Order Booking Process and Importing the Schema..... | 8-2 |
| Creating a Partner Link that Uses the File Read Functionality of the File Adapter..... | 8-3 |
| Creating an Order Booking Partner Link | 8-4 |
| Creating Receive and Invoke Activities..... | 8-5 |
| Creating an Assign Activity | 8-6 |
| Validating, Compiling, and Deploying the Order Process | 8-7 |
| Running and Verifying the Order Process | 8-7 |
| Summary | 8-8 |

9 Using the File Adapter’s Write Functionality

| | |
|---|-------------|
| Introduction..... | 9-1 |
| Designing the File Adapter’s Write Functionality to Write the Order Acknowledgment | 9-2 |
| Creating a Partner Link that Uses the File Write Functionality of the File Adapter..... | 9-2 |
| Creating an Invoke Activity | 9-4 |
| Creating an Assign Activity | 9-4 |
| Deploying the POAcknowledge Service..... | 9-5 |
| Running and Verifying the POAcknowledge Process..... | 9-5 |
| Creating a Partner Link..... | 9-6 |
| Creating a Scope Activity..... | 9-6 |
| Creating Invoke and Receive Activities..... | 9-7 |
| Creating an Initial Assign Activity | 9-8 |
| Creating a Second Assign Activity | 9-8 |
| Validating, Compiling, and Deploying the Order Booking Process | 9-10 |
| Running and Verifying the Order Booking Process | 9-10 |
| Summary | 9-11 |

10 Designing the Database Adapter to Insert Data

| | |
|--|-------------|
| Introduction..... | 10-1 |
| Designing the Database Adapter with a 1:M Relationship to Insert Order and Order Item Data into Database Tables | 10-2 |
| Creating the Order Fulfillment Process and Importing the Schema | 10-2 |
| Creating a Database Connection and Samples Tables..... | 10-3 |
| Creating a JDeveloper Database Connection..... | 10-4 |
| Creating Sample Tables..... | 10-4 |
| Testing the Database Connection | 10-5 |
| Creating a Partner Link that Uses the Database Adapter | 10-5 |
| Creating an Invoke Activity | 10-7 |
| Creating a Transform Activity and the Database Record Transformation..... | 10-7 |
| Testing the Transformation Logic..... | 10-10 |

| | |
|---|-------|
| Validating, Compiling, and Deploying the FulFillOrder Process..... | 10-11 |
| Running the FulFillOrder Process | 10-11 |
| Verifying the Process | 10-12 |
| Extending the Order Booking Process | 10-12 |
| Creating an Order FulFillment Partner Link | 10-12 |
| Creating a Scope Activity | 10-13 |
| Creating Invoke and Receive Activities..... | 10-13 |
| Creating an Initial Assign Activity | 10-14 |
| Creating a Second Assign Activity | 10-15 |
| Adding Database Exception Handling | 10-16 |
| Validating, Compiling, and Deploying the OrderBooking Process | 10-19 |
| Running the Order Booking Process..... | 10-20 |
| Summary | 10-21 |

11 Designing the Human Task Workflow System

| | |
|--|-------|
| Introduction | 11-1 |
| Designing the Human Workflow System for Manual Approval of an Order | 11-1 |
| Modeling the Order Approval Process | 11-2 |
| Creating the Order Approval Process and Importing the Schema | 11-2 |
| Adding a Human Task to the Order Approval Process..... | 11-3 |
| Assigning Input and Output Parameters for the Human Task | 11-6 |
| Creating a Task Form for the Worklist | 11-8 |
| Modeling the Task Outcome | 11-9 |
| Validating, Compiling, and Deploying the Order Approval Process..... | 11-11 |
| Running the Order Approval Process | 11-11 |
| Creating a Partner Link for the Order Approval Service | 11-13 |
| Creating a Scope Activity..... | 11-14 |
| Creating Invoke and Receive Activities..... | 11-14 |
| Creating an Initial Assign Activity | 11-15 |
| Creating a Second Assign Activity | 11-16 |
| Validating, Compiling, and Deploying the Order Process | 11-18 |
| Running and Verifying the Order Process | 11-18 |
| Summary | 11-20 |

12 Using Sensors

| | |
|--|------|
| Introduction | 12-1 |
| Adding Sensors | 12-1 |
| Creating Sensors..... | 12-1 |
| Running, Validating, and Deploying the OrderBooking Process..... | 12-4 |
| Running the Order Booking Process | 12-5 |
| Summary | 12-6 |

13 Using Notifications

| | |
|---|------|
| Introduction | 13-1 |
| Adding an E-mail Notification to the POAcknowledge Process | 13-1 |
| Configuring Your Company E-Mail Environment | 13-1 |

| | |
|---|-------------|
| Creating an E-mail Service..... | 13-3 |
| Validating, Compiling, and Deploying the POAcknowledge Process..... | 13-3 |
| Running the POAcknowledge Process | 13-4 |
| Summary | 13-4 |

Index

Preface

This manual describes how to use the Oracle BPEL Process Manager Order Booking tutorial.

This preface contains the following topics:

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

Audience

This manual is intended for anyone who is interested in using the Order Booking tutorial.

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 on Oracle BPEL Process Manager, see the following manuals:

- *Oracle BPEL Process Manager Developer's Guide*
- *Oracle BPEL Process Manager Quick Start Guide*

Conventions

The following text conventions are used in this document:

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

Introduction

This tutorial describes how to design and execute a sophisticated process that uses synchronous and asynchronous services, parallel flows of execution, conditional branching logic, fault handling and exceptions management, transformations, file adapter and database adapter functionality, and human workflow, notification, and sensor functionality.

This chapter contains the following topics:

- [Order Booking Introduction](#)
- [Using this Tutorial](#)

Note: This tutorial assumes you have installed the Basic Install installation option of Oracle Application Server SOA Suite 10.1.3.1.0 on a supported Windows operating system. See the *Oracle Application Server Installation Guide 10g Release 3 (10.1.3.1)* for your operating system for instructions.

Order Booking Introduction

GlobalCompany is a large original equipment manufacturer (OEM) that sells assembled widgets and widget parts to their customers through multiple sales channels, including:

- Retail on the Web
- Direct sales through sales representatives
- Large customers through automated business-to-business (B2B) orders

GlobalCompany uses a virtual inventory business model. GlobalCompany has contract suppliers who build and distribute their products. Their core business expertise is product design and product marketing.

GlobalCompany currently deals with two primary suppliers for their order fulfillment, based on order size and geographic location:

- Select Manufacturing (a local contract manufacturer)
- Rapid Distributors (an out-of-state distributor that sources goods from an overseas low-cost manufacturer)

Business Problem

GlobalCompany cannot provide the same services and responses to all customers, irrespective of the channel of order placement and order type.

GlobalCompany has seen sales increase but profit margins fall for the last two quarters and has decided to immediately implement stronger control mechanisms.

GlobalCompany wants their suppliers to compete for business rather than using the current system of fixed quotas and allocations.

Business Goal

GlobalCompany has the following goals:

- Consistent order booking and fulfillment across all sales channels
- Suppliers must bid on all orders and the lowest-priced supplier must be awarded the order
- The ability to override orders approved for fulfillment when margins are low, based on supplier bids
- Completely automated order booking that uses the existing systems and applications
- Automated ability to send an outgoing purchase order acknowledgment to a customer
- The ability to send an e-mail notification to confirm receipt of an order

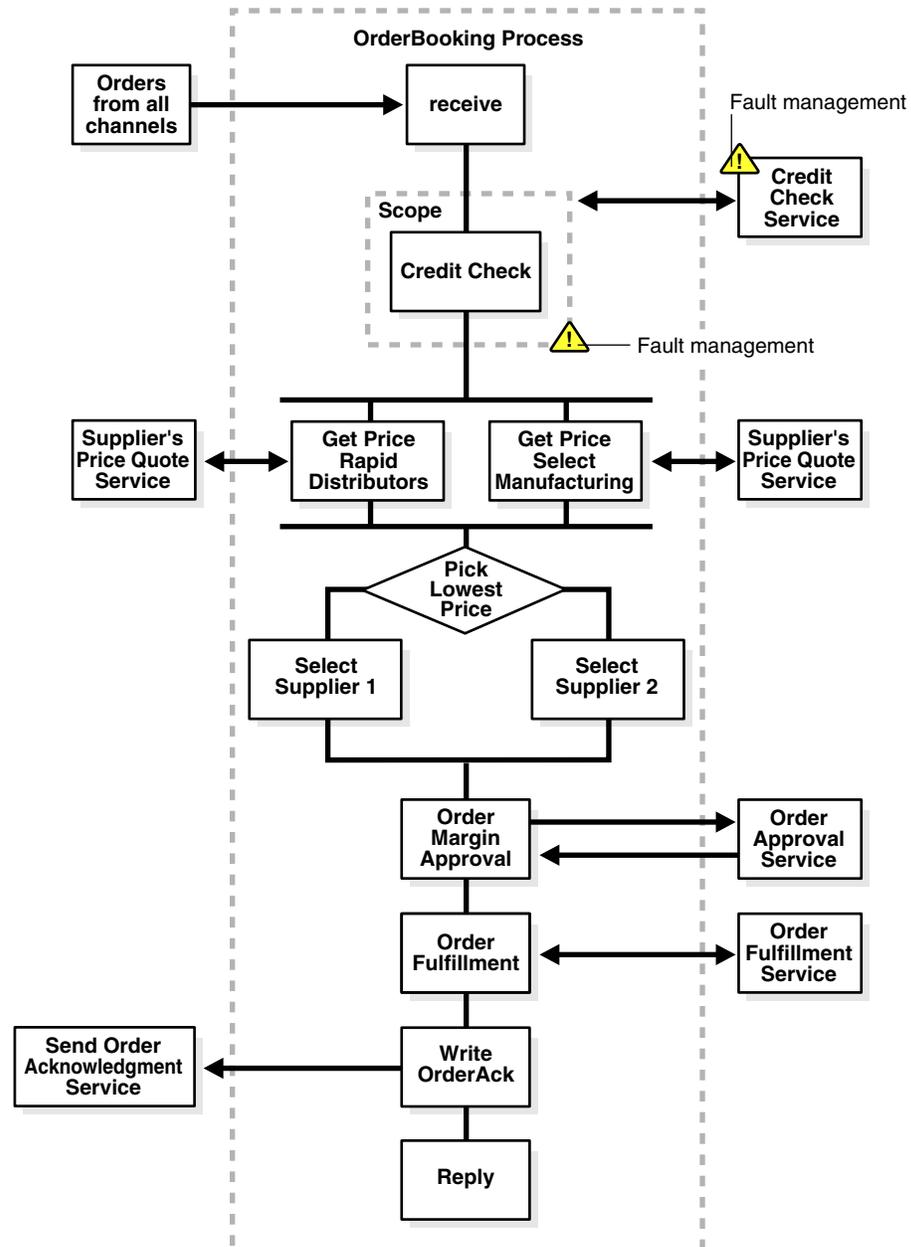
Business Solution

GlobalCompany decides to use Oracle BPEL Process Manager to create an order booking composite application. GlobalCompany uses the Oracle Applications Server platform to enable service-oriented architecture (SOA) in their enterprise.

Oracle BPEL Process Manager is used to implement a business process that receives orders placed to the company through any channels. The same process orchestrates all the existing services in the enterprise for order fulfillment with the right supplier, based on the business rules in the process.

[Figure 1-1](#) provides an overview of this solution:

Figure 1-1 Order Booking Tutorial Overview



1. The OrderBooking process receives sales orders from the following:
 - B2B systems
 - Web systems
 - New customer resource management (CRM) applications
 - Old legacy order capturing systems through files
2. The customer is identified from the order and checked against the existing Credit Rating service. The credit check system returns the relevant rating for the customer.
3. The order's line items are passed to the two suppliers for their price quotes. Select Manufacturing normally quotes lower prices because it is a direct manufacturer,

but has slow turnaround times for a response. Rapid Distributors has an automated response service for price quotes and responds in seconds, but normally quotes higher prices. Both suppliers should respond within thirty minutes. Because of business confidentiality, the order's end user details (customer contact, phone number, e-mail address, and so on) are not sent to the two suppliers. Only the part of the order that is relevant to quoting a price is sent.

4. The process collects the quotes and selects the lowest quoted price as the supplier to which to award the order.
5. An important step is to pass all this information to the business managers to approve the order for fulfillment with all the order details, especially the sales price and the supplier-quoted price, for margin reviews.
6. If the order is approved, it is sent for fulfillment. The old fulfillment service is repackaged to capture orders in the order management system and to send a notification to the supplier for order shipment.
7. An order acknowledgment is then sent to the order placing channel of the sales order to confirm that the order is accepted or rejected.

Using this Tutorial

This tutorial is divided into twelve separate phases. Each phase builds on the previous phase. You must complete each phase in the order described in this tutorial. The files to use with this tutorial are located in the following directory:

`SOA_Oracle_Home\bpel\samples\tutorials\127.OrderBookingTutorial`

A subdirectory in `127.OrderBookingTutorial` named `PracticeFiles` includes key schema files that you copy into the project directories in this tutorial. You are now ready to begin.

Installing Oracle Application Server SOA Suite

1. See the *Oracle Application Server Installation Guide 10g Release 3 (10.1.3.1)* for your operating system for instructions on installing Oracle Application Server SOA Suite.
2. Select **Basic Install** on the Oracle Application Server SOA Suite 10.1.3.1.0 Installation window.

Notes:

- If you want to use Oracle BPEL Process Manager samples, enter **welcome1** in the **AS Administrator Password** field for the **oc4jadmin** user. If you enter a different password, some samples require modification in order to work.
 - If you install the Oracle BPEL Process Manager for Developers installation type and attempt to use this tutorial, note that the ports (9700), Oracle BPEL Server startup commands, and Oracle BPEL Control user name (`bpeladmin`) may be different from those listed in this tutorial.
-
-

3. Click **Install**.
4. Click **Exit** when installation completes.

Identifying Ports

Identify several port numbers that were automatically configured during installation. You use these port numbers when creating application server and integration server connections to Oracle BPEL Server.

1. Use a text editor to open the *SOA_Oracle_Home\bpel\utilities\ant-orabpel.properties* file.
2. Make note of the ports being used by the following properties:
 - `http.port`
 - `opmn.requestport`

These port values are required later when you create connections in "[Creating Oracle Application Server and Integration Server Connections](#)" on page 1-6.

Installing Oracle JDeveloper

You must separately install Oracle JDeveloper 10.1.3.1 Studio.

1. Unzip Oracle JDeveloper into a location on your host.

Note: Do *not* install Oracle JDeveloper into a directory path that includes a space (for example, `C:\Program Files\JDev`). If you do, you receive an error when you compile BPEL processes.

2. Start Oracle JDeveloper by clicking *JDev_Oracle_Home\JDev\bin\jdev.exe* or creating a shortcut.

Starting Oracle BPEL Process Manager Components

Follow these instructions to start key Oracle BPEL Process Manager components:

1. Select **Start > All Programs > Oracle - Oracle_Home > Start SOA Suite** to start the Oracle BPEL Server and other SOA components.
2. Select **Start > All Programs > Oracle - Oracle_Home > Oracle BPEL Process Manager > Developer Prompt** to open up a command prompt at the *SOA_Oracle_Home\bpel\samples* directory. This enables you to access demonstrations and start any required Web services.

Note: Always use the **Developer Prompt** to open an operating system command prompt in this tutorial. This sets all required paths. Opening an operating system command prompt in any other way to run tutorial commands is not supported.

3. Select **Start > All Programs > Oracle - Oracle_Home > Oracle BPEL Process Manager > BPEL Control** to start the Oracle BPEL Control.
4. Select **Start > All Programs > Oracle - Oracle_Home > Oracle BPEL Process Manager > Getting Started with Samples** for details about BPEL samples and additional tutorials available for use.
5. Select **Start > All Programs > Oracle - Oracle_Home > Oracle BPEL Process Manager > Sample Worklist Application** to access the login window for the Oracle BPEL Worklist Application.

Note: You *must* use Internet Explorer version 6 or Mozilla Firefox version 1.0.7 with this tutorial.

Creating Oracle Application Server and Integration Server Connections

1. Select **Connection Navigator** from the **View** main menu in Oracle JDeveloper.
2. Right-click **Application Server**.
3. Select **New Application Server Connection**.
4. Click **Next** on the **Welcome** page.
5. Provide a meaningful name for connecting to the application server. Remember this name, as you select it when creating an integration server connection in Step 19 on page 1-7.
6. Select **Oracle Application Server 10g 10.1.3** from the **Connection Type** list.
7. Click **Next**.
8. Enter the following details:

| Field | Value |
|-----------------|--|
| Username | oc4jadmin |
| Password | Enter the same administrator password you entered in "Installing Oracle Application Server SOA Suite" on page 1-4. |
| Deploy Password | Select this check box; otherwise, you must authenticate each time you deploy a process. |

9. Click **Next**.
10. Enter the following details:

| Field | Value |
|--------------------------|--|
| Host Name | localhost |
| OPMN Port | Enter the same port value used by the <code>opmn.requestport</code> property that you identified in Step 2 of "Identifying Ports" on page 1-5. This is likely port 6003. |
| Group/OC4J Instance Name | home |
| Single Instance | Select this radio button. |
| Group | Leave this radio button unselected. |

11. Click **Next**.
12. Test the connection by clicking **Test Connection**. If the connection is successful, the following message appears:

```
Success!
```
13. Click **Finish**.
14. Right-click **Integration Server** in the **Connection Navigator**.
15. Select **New Integration Server Connection**.

You use this integration server connection to deploy all your BPEL processes.

16. Click **Next** on the **Welcome** page.
17. Provide a meaningful name for connecting to the server.
18. Click **Next**.
19. Select the application server connection name you specified in Step 5 on page 1-6 from the **Application Server** list.
20. Enter the following details:

| Field | Value |
|--|--|
| Host Name | localhost |
| Port | Enter the same port value used by the <code>http.port</code> property that you identified in Step 2 of " Identifying Ports " on page 1-5. This is likely port 8888. |
| Add host name to the list of proxy exceptions | Select this check box to automatically add this host to the list of proxy exceptions shown in the Exceptions field under Tools > Preferences > Web Browser and Proxy . The first time it is started, Oracle JDeveloper by default picks up these proxy exceptions that are set for your browser. |

21. Click **Next**.
22. Test the connection by clicking **Test Connection**. If the connection is successful, the following message appears:


```
Application Server: OK
BPEL Process Manager Server: OK
ESB Server: OK
```

23. Click **Finish**.

Setting the Hostname in Your Oracle JDeveloper Web Browser Preferences

The hostname of your computer was automatically added to the Oracle JDeveloper preference settings during installation. If the hostname does not appear in the preference settings, you can receive parsing errors when selecting a WSDL file on the Service Explorer window while creating a partner link.

1. Select **Preferences** from the **Tools** main menu.
2. Click **Web Browser and Proxy**.
3. Verify that your hostname is in the **Exceptions** field. For example, if your hostname is `myhost-pc`:

```
us.acme.com|*.us.acme.com|localhost|myhost-pc|127.0.0.1
```

4. Ensure also that `localhost` appears in the **Exceptions** field.
5. Click **OK**.

You are now ready to design your BPEL process.

Invoking a Synchronous Service

This chapter of the tutorial describes how to invoke a synchronous service.

This chapter contains the following topics:

- [Introduction](#)
- [Invoking a Synchronous Service to Receive an Order and Obtain a Credit Rating](#)
- [Summary](#)

Introduction

In the first phase of this tutorial, you learn how to build a simple *asynchronous* BPEL process for receiving an order from a client and obtaining a credit rating. You perform the following key tasks:

- Deploy (start) a *synchronous* Credit Rating service.
- Create a new BPEL process named **OrderBooking**.
- Create a partner link to interact with the synchronous Credit Rating service.
- Design the BPEL process to invoke the synchronous Credit Rating service.
- Deploy the BPEL process.
- Run the deployed BPEL process from Oracle BPEL Control and submit a social security card number to the Credit Rating service to check the client's credit history.
- Review the credit rating results returned from the Credit Rating service to the client.

Invoking a Synchronous Service to Receive an Order and Obtain a Credit Rating

This section contains the following topics:

- [Starting Oracle BPEL Server and Oracle JDeveloper](#)
- [Starting and Testing Your Services](#)
- [Creating an Application](#)
- [Creating the Order Booking Project and Importing the Schema](#)
- [Validating Your Project During the Design Phase](#)
- [Creating a Credit Rating Service Partner Link](#)

- [Creating Scope and Invoke Activities](#)
- [Creating an Initial Assign Activity to Send a Request to the Credit Rating Service](#)
- [Creating a Second Assign Activity to Receive a Response from the Credit Rating Service](#)
- [Creating a Third Assign Activity](#)
- [Validating, Compiling, and Deploying the Order Booking Process](#)
- [Running the Order Booking Process](#)

Starting Oracle BPEL Server and Oracle JDeveloper

Ensure that Oracle BPEL Server and Oracle JDeveloper are started. See "[Starting Oracle BPEL Process Manager Components](#)" on page 1-5 for instructions.

Starting and Testing Your Services

During this tutorial, the BPEL process that you design communicates with the Credit Rating service. You must first deploy this service and test that it is running.

1. Select **Start > All Programs > Oracle - Oracle_Home > Oracle BPEL Process Manager > Developer Prompt** to open up an operating system command prompt at the `SOA_Oracle_Home\bpel\samples` directory.

Note: *Always* use the **Developer Prompt** to open an operating system command prompt in all phases of the Order Booking tutorial. Opening an operating system command prompt in any other way to run tutorial commands is not supported.

2. Change directories to the `utils\CreditRatingService` subdirectory:

```
cd utils\CreditRatingService
```

3. Enter the following command:

```
ant
```

This deploys and starts the required services for using this tutorial. If successful, a message similar to the following appears at the end:

```
BUILD SUCCESSFUL
Total time: 3 seconds
oraBPELPM\bpel\samples\utils\CreditRatingService>
```

If you receive an error saying the path was not found, ensure that your environment path is set to the `SOA_Oracle_Home\bpel\bin` directory.

4. Access Oracle BPEL Control using Internet Explorer to see if all required services are running. Select **Start > All Programs > Oracle - Oracle_Home > Oracle BPEL Process Manager > BPEL Control**.
5. Enter the following details to log into Oracle BPEL Control and click **Login**:

| Field | Value |
|----------|-----------|
| Username | oc4jadmin |
| Password | password |

Where *password* is the value you entered during installation in "[Installing Oracle Application Server SOA Suite](#)" on page 1-4.

The services for this tutorial are running if **CreditRatingService**, **TaskActionHandler**, and **TaskManager** display in the **Dashboard** tab.

See Also: "[Creating Oracle Application Server and Integration Server Connections](#)" on page 1-6 for instructions on creating a connection to Oracle BPEL Control

Creating an Application

From Oracle JDeveloper, follow these instructions to create an application before creating the Order Booking project.

1. In the **Application Navigator**, right-click **Applications** and select **New Application**.
2. In the Create Application window, enter the following details and click **OK**:

| Field | Value |
|----------------------------|--------------------------------|
| Application Name | OrderBookapplication |
| Directory Name | Accept the default path. |
| Application Package Prefix | Leave unspecified. |
| Application Template | No Template [All Technologies] |

3. In the Create Project window, click **Cancel**.

Creating the Order Booking Project and Importing the Schema

From Oracle JDeveloper, follow these instructions to create the Order Booking project and import the schema.

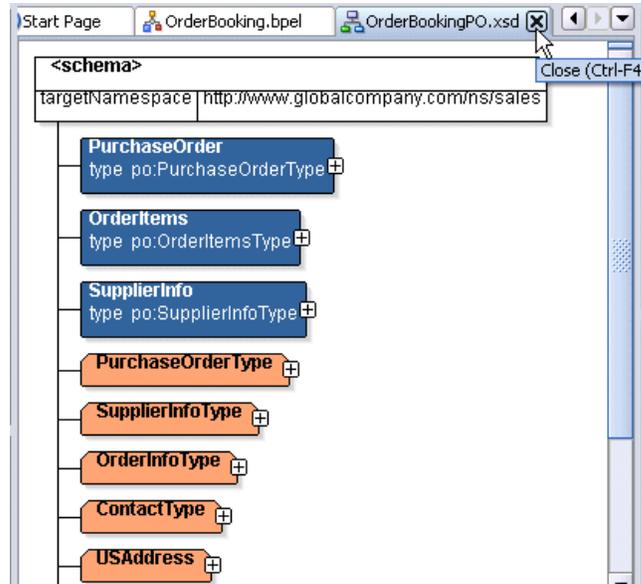
Caution: Do not include any special characters in the project name (such as periods) or in any activity or element names. If you do include special characters, errors appear when you attempt to compile your project.

1. Right-click the application name you just created, **OrderBookapplication**, and select **New Project**.
2. Under **Categories**, expand **General** and click **Projects**.
3. Under **Items**, click **BPEL Process Project** and click **OK**.
4. In the BPEL Project Creation Wizard - Project Settings window, enter the following details and click **Next**.

| Field | Value |
|-----------|--------------------------|
| Name | OrderBooking |
| Namespace | Accept the default path. |

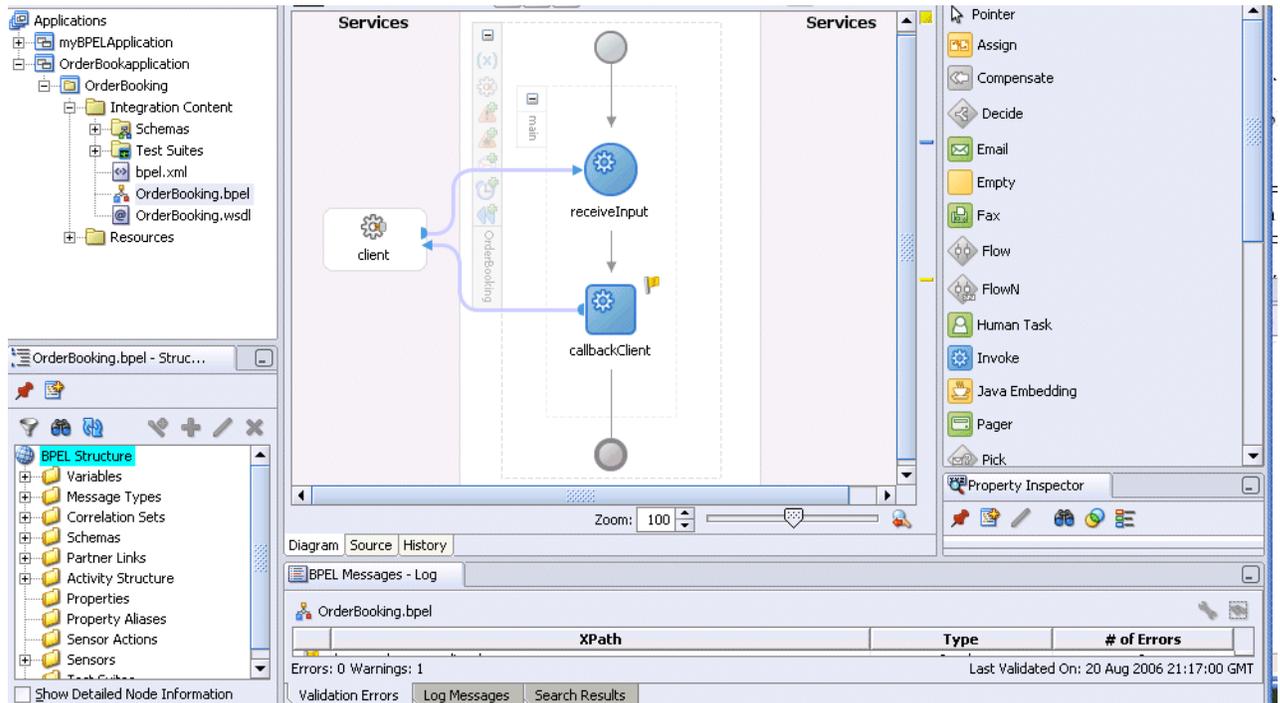
| Field | Value |
|------------------------------|--|
| Use Default Project Settings | Leave this checked to use the same name for the BPEL process and the BPEL project, and to use the default project directory. |
| Template | Asynchronous BPEL Process |

5. Click the **flashlight** icon to the right of the **Input Schema Element** field.
6. In the Select Schema window, select `OrderBookingPO.xsd` from the `SOA_Oracle_Home\bpel\samples\tutorials\127.OrderBookingTutorial\Practice Files` directory and click **Open**.
The Type Chooser window appears.
7. Expand and select **Imported Schemas > OrderBookingPO.xsd > PurchaseOrder**.
8. Click **OK**.
The **OrderBookingRequestMessage** message type is automatically assigned an input message part name of **payload** and message part element of **PurchaseOrder**.
9. Click the **flashlight** icon to the right of the **Output Schema Element** field.
The Type Chooser window appears.
10. Expand and select **Imported Schemas > OrderBookingPO.xsd > PurchaseOrder**.
11. Click **OK**.
The **OrderBookingResponseMessage** message type is automatically assigned an output message part name of **payload** and message part element of **PurchaseOrder**.
12. Click **Finish**.
The GlobalCompany schema is now imported into your project. This schema defines the structure of the purchase order to submit.
The following files appear in the **Application Navigator** section of Oracle JDeveloper:
 - The **bpel.xml**, **OrderBooking.bpel**, and **OrderBooking.wsdl** files appear under **OrderBooking > Integration Content**.
 - The **OrderBookingPO.xsd** file appears under **OrderBooking > Integration Content > Schemas**.
13. If you want, you can double-click **OrderBookingPO.xsd** to view the design and source modes of this file.
14. If you view **OrderBookingPO.xsd**, close the window when done by clicking the **X** button on the **Title** tab.



15. Double-click **OrderBooking.bpel** to display the **Structure** section in the lower left of the window.

Note: The **Diagram** tab must be enabled to see the **Structure** section.



16. Copy **OrderBookingPO_1.xml** from the *SOA_Oracle_Home\bpel\samples\tutorials\127.OrderBookingTutorial\Practice Files* directory to the *JDev_Oracle_Home\jdev\mywork\OrderBookapplication\OrderBooking* directory.
17. Select **Save** from the **File** main menu.

18. If you want to view the message part name and element that were automatically assigned to your message types during BPEL project creation, perform the following steps.
 - a. Expand **Message Types > Process WSDL - OrderBooking.wsdl > OrderBookingRequestMessage > payload** in the **Structure** section.
 - b. Right-click **payload** and select **Edit Message Part** to display the Edit Message Part window.

This defines the (input parameter) message part name (**payload**) of the **PurchaseOrder** element type.
 - c. Click **Cancel**.
 - d. Expand **Message Types > Process WSDL - OrderBooking.wsdl > OrderBookingResponseMessage > payload** in the **Structure** section.
 - e. Right-click **payload** and select **Edit Message Part** to display the Edit Message Part window.

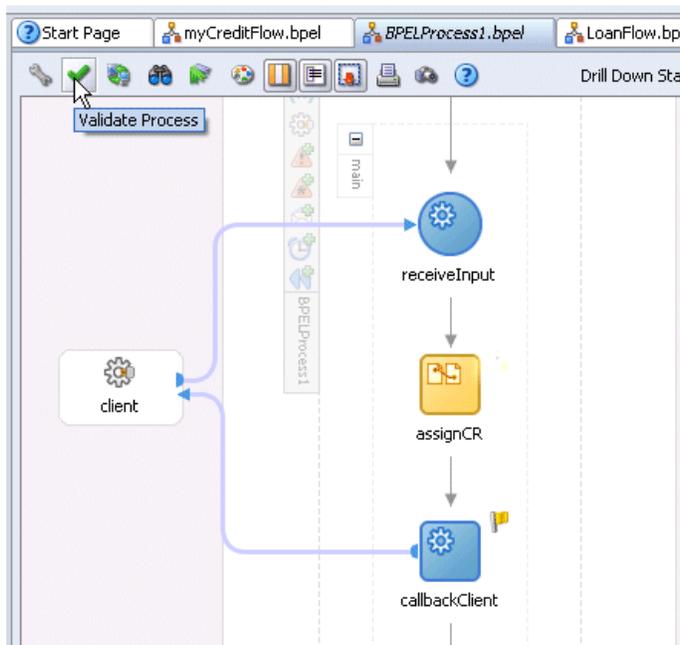
This defines the (output parameter) message part name (**payload**) of the **PurchaseOrder** element type.
 - f. Click **Cancel**.

Validating Your Project During the Design Phase

As you design your BPEL project, small yellow flags can display above activities. You may have correctly configured these activities; they simply may not yet have been validated.

1. Click **Validate** above the designer window to validate your process.

Figure 2–1 Validating a Process



If the activity correctly validates, the flag disappears.

2. If the yellow flag remains, you still have validation errors in activities or your process design is not yet complete.
3. Click the yellow flag above the activity to display details about the validation error and corrective actions to follow.

Note: When you deploy a process, as described in "[Validating, Compiling, and Deploying the Order Booking Process](#)" on page 2-17, your process is automatically validated, compiled, and, if successful, deployed together.

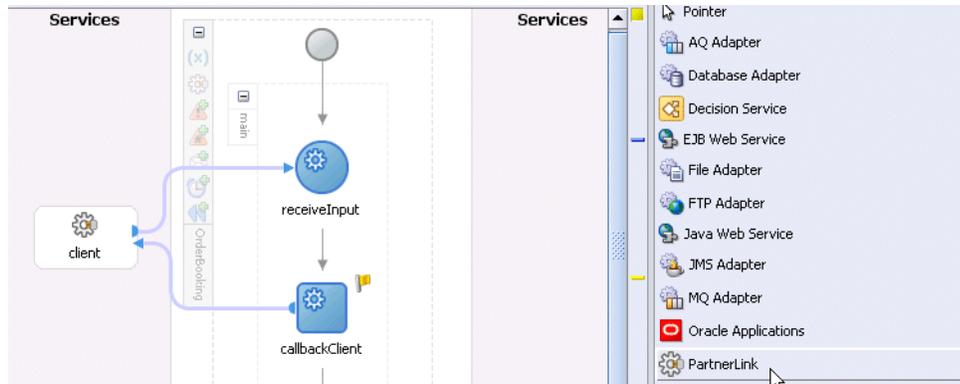
Creating a Credit Rating Service Partner Link

Summary: You now create a partner link for the Credit Rating service. This is the service you started when you ran ant in "[Starting and Testing Your Services](#)" on page 2-2. A partner link enables you to define the external services with which the **OrderBooking** BPEL process is to interact. A partner link type characterizes the conversational relationship between two services by defining the roles played by each service in the conversation and specifying the port type provided by each service to receive messages within the context of the conversation.

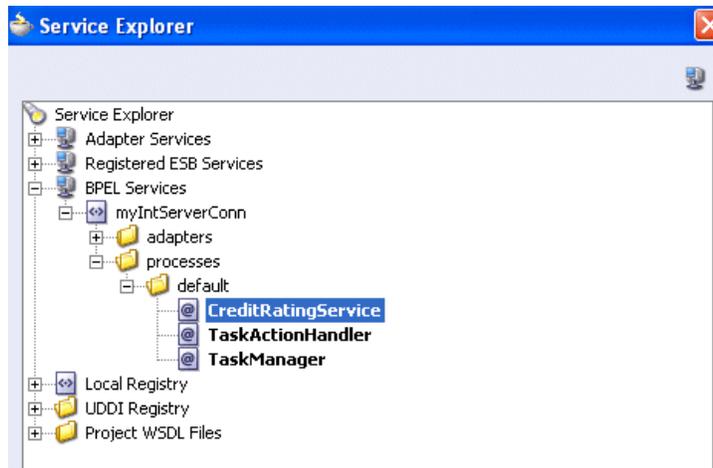
1. Ensure that **Services** is selected in the drop-down list of the **Component Palette** section in the upper right section of Oracle JDeveloper. The **Services** list displays the adapters, partner link, and decision service that you can drag and drop into the designer window.



2. Drag and drop a **PartnerLink** activity onto the right side of the designer window anywhere beneath the header **Services**. Note that both sides of the designer window have sections in which partner links can be placed.



- Enter the following values to create a partner link for the Credit Rating service. For the WSDL File field below, click the **flashlight** (the second icon from the left named **Service Explorer**) to access the Service Explorer window for automatically selecting the Credit Rating service deployed in "[Starting and Testing Your Services](#)" on page 2-2:



| Field | Value |
|-------------------|---|
| Name | CreditRatingService |
| WSDL File | <p>Access this URL by clicking the Service Explorer flashlight icon and expanding and selecting BPEL Services > <i>my_integration_server_connection</i> > processes > default > CreditRatingService</p> <p>where <i>my_integration_server_connection</i> is the server connection name you specified in Step 17 on page 1-7.</p> <p>http://localhost:8888/orabpel/default/CreditRatingService/CreditRatingService?wsdl</p> <p>See Also: "Setting the Hostname in Your Oracle JDeveloper Web Browser Preferences" on page 1-7 if you receive a parsing error when attempting to add a WSDL file in the Service Explorer window.</p> |
| Partner Link Type | CreditRatingService |

| Field | Value |
|--------------|--|
| Partner Role | CreditRatingServiceProvider |
| My Role | Leave unspecified since this is a synchronous service. |

4. Click **OK**.
5. Select **Save** from the **File** main menu.

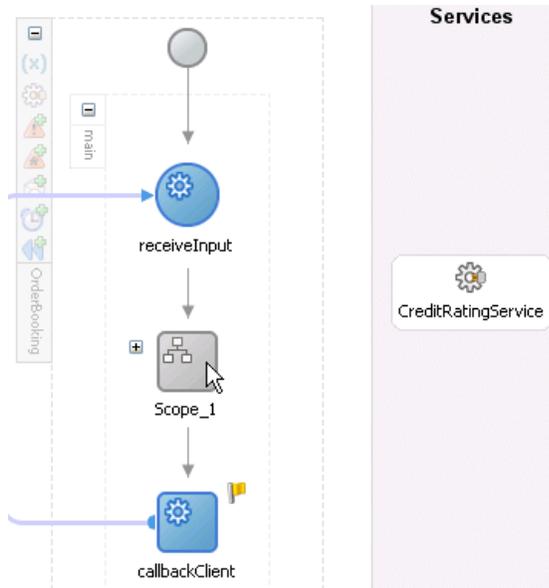
Creating Scope and Invoke Activities

Summary: You now create the **Invoke** activity that sends request data from the **OrderBooking** process to the synchronous Credit Rating service and receives a response. The customer **custID** variable from the Order Booking schema is used to obtain the credit rating. When you run the **OrderBooking** process, you specify a value for **CustID** in Step 4 of "[Running the Order Booking Process](#)" on page 2-18.

1. Ensure that **Process Activities** is selected in the drop-down list of the **Component Palette** section in the upper right section of Oracle JDeveloper. The **Process Activities** list displays the activities that you can drag and drop into the designer window.

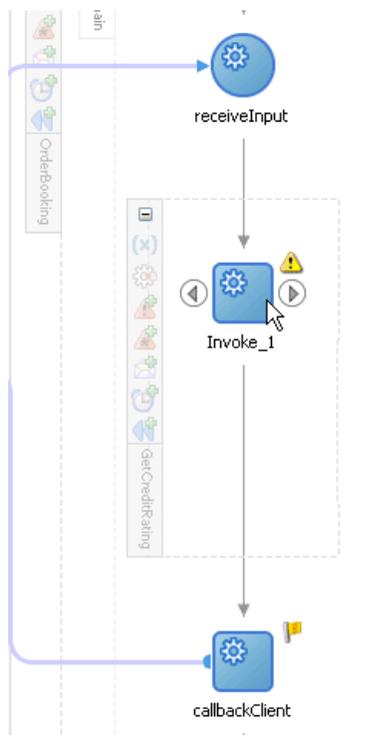


2. Drag and drop a **Scope** activity from the **Component Palette** section to below the **receiveInput Receive** activity. Always pay careful attention to where you place the **Scope** activity; throughout all phases of this tutorial, you drag and drop activities many times.



Note that the first time your drag and drop an activity, it has a generic name such as **Scope_1**.

3. Double-click the **scope** icon to display the Scope window.
4. Enter **GetCreditRating** in the **Name** field of the **General** tab.
5. Click **OK**.
6. Click the + sign to expand the **Scope** activity. You now add an **Invoke** activity and two **Assign** activities to this **Scope** activity.
7. Drag and drop an **Invoke** activity from the **Component Palette** section to inside the **GetCreditRating Scope** activity.

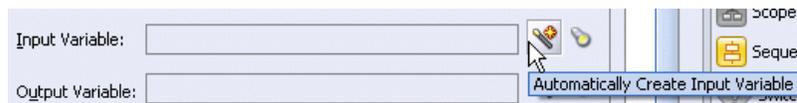


8. Double-click the **Invoke** icon to display the Invoke window.
9. Enter the following details:

| Field | Value |
|--------------|---------------------|
| Name | invokeCR |
| Partner Link | CreditRatingService |

The **Operation (process)** field is automatically filled in.

10. Click the first icon to the right of the **Input Variable** field. This is the automatic variable creation icon.



11. Click **OK** on the Create Variable window.

A variable named **invokeCR_process_InputVariable** is automatically created in the **Input Variable** field. This variable is automatically assigned a message type of **CreditRatingServiceRequestMessage**.

12. Click the first icon to the right of the **Output Variable** field.

13. Click **OK** on the Create Variable window.

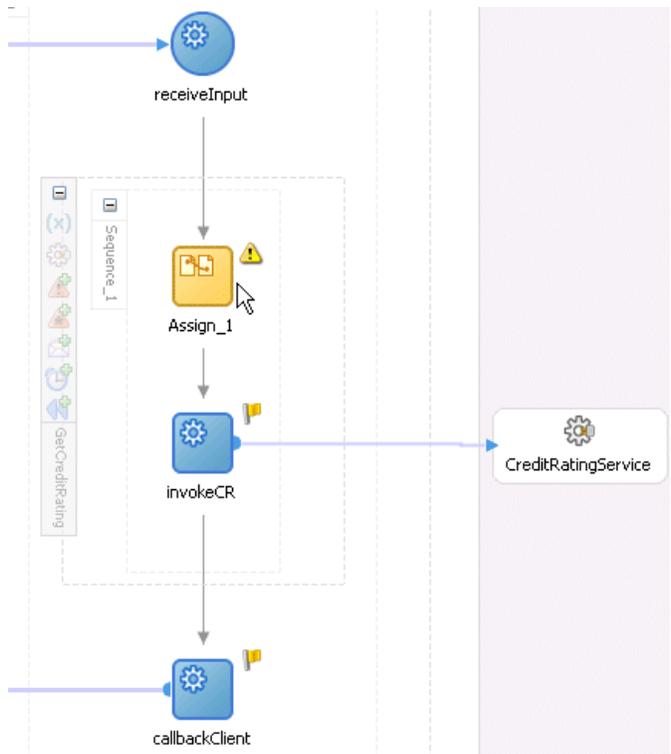
A variable named **invokeCR_process_OutputVariable** is automatically created in the **Output Variable** field. This variable is automatically assigned a message type of **CreditRatingServiceResponseMessage**.

14. Click **OK**.
15. Select **Save** from the **File** main menu.

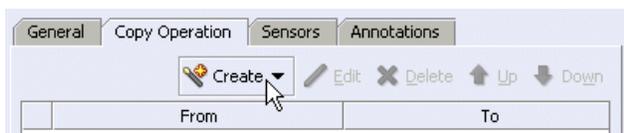
Creating an Initial Assign Activity to Send a Request to the Credit Rating Service

Summary: You now create the first of two **Assign** activities in this phase. This **Assign** activity takes the customer **custID** element from the Order Booking schema and assigns it to **invokeCR_process_InputVariable**. This is the variable used for sending a request to the Credit Rating service.

1. Drag and drop an **Assign** activity from the **Component Palette** section to above the **invokeCR Invoke** activity and inside the **GetCreditRating Scope** activity.



2. Double-click the **assign** icon to display the Assign window.
3. Click the **General** tab.
4. Enter **assignCR** in the **Name** field.
5. Click **Apply**.
6. Click the **Copy Operation** tab.
7. Click **Create** and select **Copy Operation** to display the Create Copy Operation window.

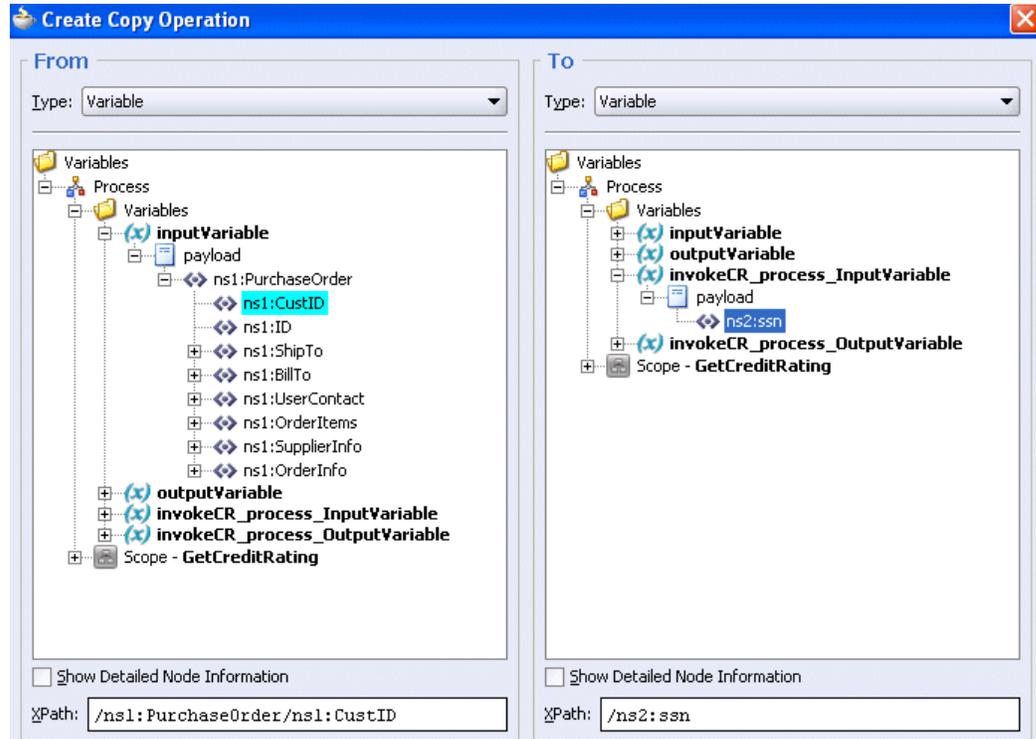


8. Enter the following details:

| Field | Value |
|--------------------|---|
| From | |
| ▪ Type | Variable |
| ▪ Variables | Expand and select Variables > inputVariable > payload > ns1:PurchaseOrder > ns1:CustID |
| | Note: The namespace number values (for example, ns1 , ns2) can vary. Use the namespaces values that automatically appear. |
| To | |
| ▪ Type | Variable |

| Field | Value |
|-----------|--|
| Variables | Expand and select Variables > invokeCR_process_InputVariable > payload > ns2:ssn |

The Create Copy Operation window appears as follows:

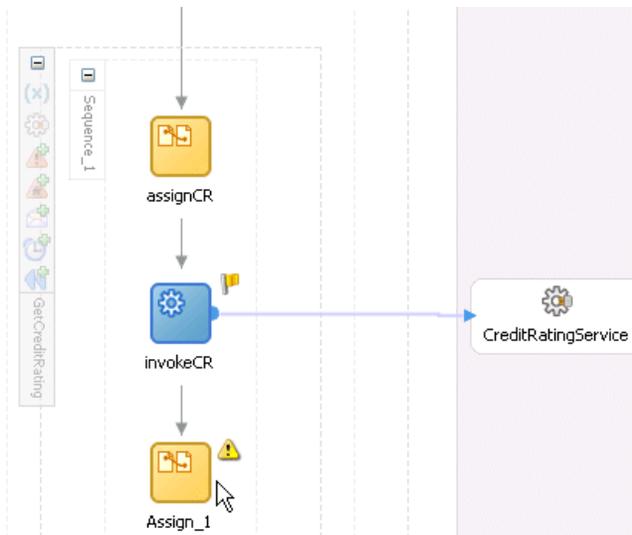


9. Click **OK** to close the Create Copy Operation window and the Assign window.
10. Select **Save** from the **File** main menu.

Creating a Second Assign Activity to Receive a Response from the Credit Rating Service

Summary: This **Assign** activity takes the response received from the Credit Rating service and places it in the comment field.

1. Drag and drop an **Assign** activity from the **Component Palette** section to below the **invokeCR Invoke** activity and inside the **GetCreditRating Scope** activity.



2. Double-click the **assign** icon to display the Assign window.
3. Click the **General** tab.
4. Enter **assignRating** in the **Name** field.
5. Click **Apply**.
6. Click the **Copy Operation** tab.
7. Click **Create** and select **Copy Operation** to display the Create Copy Operation window.
8. Enter the following details:

Notes:

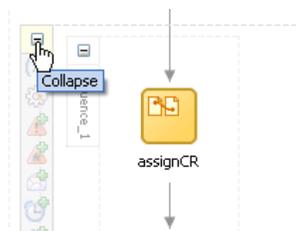
- Instead of manually entering an expression, you can press **Ctrl** and then the space bar in the **Expression** field. Scroll through the list of values that appears and double-click the value you want. Edit the value as necessary. As you enter information, a trailing slash can appear. This means you are being prompted for additional information. Either enter additional information, or press the **Esc** key and delete the trailing slash to complete the input of information.
 - Do not copy and paste the expression syntax shown in this document into Oracle JDeveloper. This is because the single and double quotation marks are sometimes not compatible with Oracle JDeveloper. Instead, use the Expression Builder window or press **Ctrl** and then the space bar to use the XPath Building Assistant.
-
-

| Field | Value |
|--------|------------|
| From | |
| ■ Type | Expression |

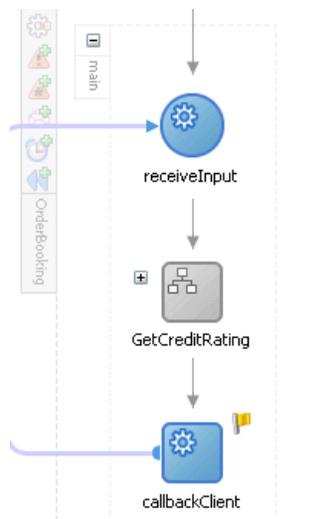
| Field | Value |
|--|---|
| <ul style="list-style-type: none"> Expression | <p><code>concat('Good credit, Rating = ', bpws:getVariableData('invokeCR_process_OutputVariable','payload','/ns2:rating'))</code></p> <p>Note: Press Ctrl and then the space bar to display a list for selecting (double-clicking) this syntax. The namespace number values (for example, ns1, ns2) can vary. Use the namespace values that automatically appear.</p> |
| To | |
| <ul style="list-style-type: none"> Type | Variable |
| <ul style="list-style-type: none"> Variables | Expand and select Variables > inputValue > payload > ns1:PurchaseOrder > ns1:OrderInfo > ns1:OrderComments |

9. Click **OK** to close the Create Copy Operation window and the Assign window.

10. Click the - sign to close the **GetCreditRating Scope** activity.



When closed, the **GetCreditRating Scope** activity looks as follows:



11. Select **Save** from the **File** main menu.

Creating a Third Assign Activity

1. Drag and drop an **Assign** activity from the **Component Palette** section to below the **GetCreditRating Scope** activity and above the **callbackClient** activity.

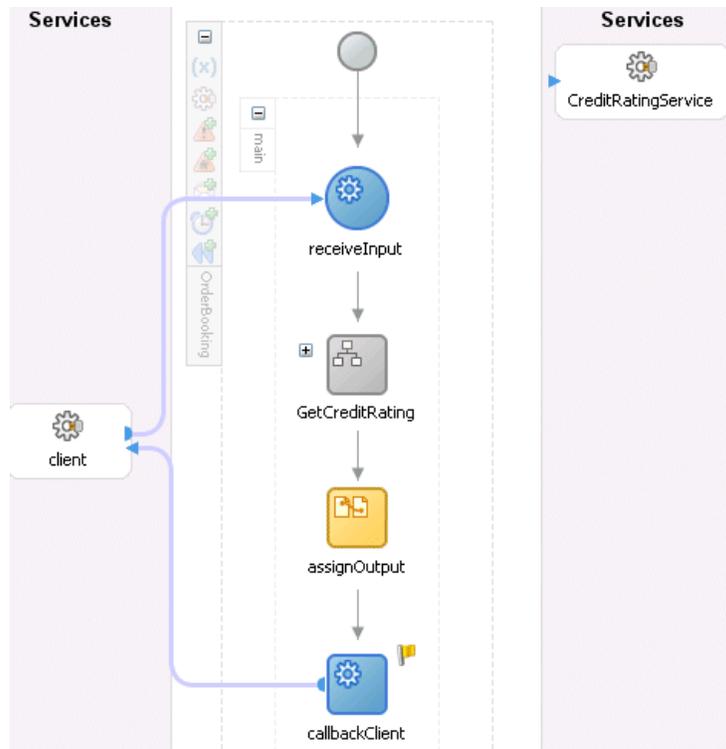


2. Double-click the **assign** icon to display the Assign window.
3. Click the **General** tab.
4. Enter **assignOutput** in the **Name** field.
5. Click **Apply**.
6. Click the **Copy Operation** tab.
7. Click **Create** and select **Copy Operation** to display the Create Copy Operation window.
8. Enter the following details to create a statement:

| Field | Value |
|--------------------|---|
| From | |
| ▪ Type | Variable |
| ▪ Variables | Expand and select Variables > inputVariable > payload |
| To | |
| ▪ Type | Variable |
| ▪ Variables | Expand and select Variables > outputVariable > payload |

9. Click **OK** to close the Create Copy Operation window and the Assign window.
10. Select **Save** from the **File** main menu.

When complete, the designer window looks as follows:



Validating, Compiling, and Deploying the Order Booking Process

1. Go to the **Application Navigator** section.
2. Right-click **OrderBooking**.
3. Select **Deploy > my_integration_server_connection > Deploy to default domain**.

where *my_integration_server_connection* is the integration server connection name you specified in Step 17 on page 1-7.

This compiles the BPEL process.

```

Messages - Log
Beginning Deployment Process...
Compiling...
Compiling C:\jdev\jdev\mywork\OrderBookapplication\OrderBooking\bpel\OrderBooking.bpel
BPEL suitcase generated in: C:\jdev\jdev\mywork\OrderBookapplication\OrderBooking\output\bpel_OrderBooking
[10:54:52 AM] Successful compilation: 0 errors, 0 warnings.
Deploying to http://localhost:9700 domain: default. Please wait...
[10:54:53 AM] Please check Ant log to determine whether the project deployed successfully.
  
```

4. Check for errors by clicking the buttons at the bottom of the window. Errors that do not display under the **Messages** button can sometimes appear under the **Apache Ant** button.

If there are no errors, deployment was successful. If there are errors, double-click the error to display details about the type and location of the error.

5. Make corrections and deploy again.

Running the Order Booking Process

You are now ready to begin the process of applying for a new order.

1. Access Oracle BPEL Control using Internet Explorer or refresh the page if it is already open. Select **Start > All Programs > Oracle - Oracle_Home > Oracle BPEL Process Manager > BPEL Control**.
2. Enter the following details to log into Oracle BPEL Control and click **Login**:

| Field | Value |
|----------|-----------------|
| Username | oc4jadmin |
| Password | <i>password</i> |

Where *password* is the value you entered during installation.

The **Dashboard** tab of Oracle BPEL Control appears.

3. Click **OrderBooking** in the **Deployed BPEL Processes** list.
4. Enter a nine-digit integer value that does *not* begin with zero in the **CustID** field of the HTML Form and click **Post XML Message**. Ignore the other fields.

The **BPEL Processes** tab displays a message similar to the following:

Test Instance Initiated

Instance '39e706a46ad531be:858bf1:fcc240f310:-7ffc' is being processed asynchronously.

5. Click the **Instances** tab at the top.
6. Verify that the **OrderBooking** and **CreditRatingService** instances were created. Both should appear in the **Instance** list. The **Last Modified** list displays the times at which they were created.

| | Dashboard | BPEL Processes | Instances | Activities |
|--------------------------------------|--|------------------------------|--------------------|------------|
| List of BPEL Process Instances 1 - 2 | | | | |
| | Instance | BPEL Process | Last Modified ↑ | |
| ✓ | 1 : Instance #1 of OrderBooking | OrderBooking (v. 1.0) | 7/22/06 7:18:49 PM | |
| ✓ | 2 : Instance #2 of CreditRatingService | CreditRatingService (v. 1.0) | 7/22/06 7:18:48 PM | |

7. Click the **OrderBooking** instance.
A message appears stating the following:
The state of this instance is *Completed*
8. Click the **Audit** link to follow the instance execution process. The **Audit** link displays the following details:

[2006/05/29 11:29:31] New instance of BPEL process "OrderBooking" initiated (# "1").

```

</process>
  <sequence>
    receiveInput
      [2006/05/29 11:29:31] Received "inputVariable" call from partner "client" More...
      <scope name="GetCreditRating">
        <sequence>
          assignCR
            [2006/05/29 11:29:31] Updated variable "invokeCR_process_InputVariable" More...
          invokeCR
            [2006/05/29 11:29:31] Invoked 2-way operation "process" on partner "CreditRatingService"._Mc
          assignRating
            [2006/05/29 11:29:31] Updated variable "inputVariable" More...
        </sequence>
      </scope>
    assignOutput
      [2006/05/29 11:29:31] Updated variable "outputVariable" More...
    callbackClient
      [2006/05/29 11:29:31] Skipped callback "onResult" on partner "client". More...
  </sequence>
  [2006/05/29 11:29:31] BPEL process instance "1" completed
</process>
  
```

9. Click the **Flow** link.

10. Click **callbackClient** at the bottom of the flow to see the results.

Activity Audit Trail -- Web Page Dialog

```

<ns1:State/>
<ns1:Zip/>
<ns1:Country/>
</ns1:Address>
</ns1:ShipTo>
<ns1:BillTo
xmlns:ns1="http://www.globalcompany.com/ns/sales"/>
- <ns1:UserContact
xmlns:ns1="http://www.globalcompany.com/ns/sales">
  <ns1:PhoneNumber/>
  <ns1:EmailAddress/>
</ns1:UserContact>
<ns1:OrderItems
xmlns:ns1="http://www.globalcompany.com/ns/sales"/>
- <ns1:SupplierInfo
xmlns:ns1="http://www.globalcompany.com/ns/sales">
  <ns1:SupplierPrice/>
  <ns1:SupplierName/>
</ns1:SupplierInfo>
- <ns1:OrderInfo
xmlns:ns1="http://www.globalcompany.com/ns/sales">
  <ns1:OrderDate/>
  <ns1:OrderPrice/>
  <ns1:OrderStatus/>
  <ns1:OrderComments>Good credit, Rating =
  560</ns1:OrderComments>
</ns1:OrderInfo>
  
```

11. Verify the credit rating value. The comment field shows that the credit rating is returned as **560**; this is also returned in the variable **invokeCR_process_OutputVariable**.

Summary

You have now built a simple asynchronous BPEL process that received an order from a client and obtained a credit rating from a synchronous Credit Rating service. You performed the following key tasks:

- Deployed a synchronous Credit Rating service.
- Created a new project named **OrderBooking**, which automatically created a BPEL process of the same name.
- Created a partner link that enabled you to define the external service (the deployed Credit Rating service) with which your BPEL process interacted.
- Designed the BPEL process and invoked the Credit Rating service by creating all necessary activities.
- Deployed the BPEL process to Oracle BPEL Control.
- Ran the BPEL process from Oracle BPEL Control and submitted a social security card number to the Credit Rating service, which checked the client's credit history.
- Reviewed the credit rating results returned from the Credit Rating service to the client.

Invoking an Asynchronous Service

This chapter of the tutorial describes how to invoke an asynchronous service.

This chapter contains the following topics:

- [Introduction](#)
- [Invoking an Asynchronous Service to Receive the Supplier Price for the Order](#)
- [Summary](#)

Introduction

This phase of the tutorial adds to the **OrderBooking** process you created in [Chapter 2, "Invoking a Synchronous Service"](#). Ensure that you have successfully completed that phase before performing this one. In this phase, you build a simple asynchronous BPEL process. You perform the following key tasks:

- Deploy an *asynchronous* Rapid Distributors service.
- Create a partner link to interact with the Rapid Distributors service.
- Design the BPEL process to invoke the Rapid Distributors service.
- Deploy the BPEL process.
- Run the deployed BPEL process from Oracle BPEL Control by completing and submitting an order form to the Rapid Distributors service.
- Review the supplier price returned from the Rapid Distributors service to the client.

Invoking an Asynchronous Service to Receive the Supplier Price for the Order

This section contains the following tasks:

- [Starting and Testing Your Services](#)
- [Creating a Rapid Distributors Partner Link](#)
- [Creating Invoke and Receive Activities](#)
- [Creating an Assign Activity](#)
- [Validating, Compiling, and Deploying the Order Booking Process](#)
- [Running the Order Booking Process](#)

Starting and Testing Your Services

During this tutorial, the BPEL process that you design communicates with the Rapid Distributors and Select Manufacturing services. You must first start these services and test that they are running.

1. Select **Start > All Programs > Oracle - Oracle_Home > Oracle BPEL Process Manager > Developer Prompt** to open up an operating system command prompt at the `SOA_Oracle_Home\bpel\samples` directory.

Note: *Always* use the **Developer Prompt** to open an operating system command prompt in all phases of the Order Booking tutorial. Opening an operating system command prompt in any other way to run tutorial commands is not supported.

2. Change directories to the `tutorials\127.OrderBookingTutorial\PriceQuote` subdirectory.


```
cd tutorials\127.OrderBookingTutorial\PriceQuote
```

3. Enter the following command:

```
ant
```

This deploys and starts the Rapid Distributors service required for using this phase of the tutorial. If successful, a message appears at the end:

```
BUILD SUCCESSFUL
```

The Select Manufacturing service and user application are also created. You use these services later in [Chapter 4, "Adding Parallel Flow"](#).

4. Access Oracle BPEL Control to see if all required services are running by selecting **Start > All Programs > Oracle - Oracle_Home > Oracle BPEL Process Manager > BPEL Control**.
5. Enter the following details to log into Oracle BPEL Control and click **Login**:

| Field | Value |
|----------|-----------------|
| Username | oc4jadmin |
| Password | <i>password</i> |

Where *password* is the value you entered during installation.

The services for this phase are running if **RapidDistributors** and **SelectManufacturing** display in the **Dashboard** tab.

Creating a Rapid Distributors Partner Link

Summary: You now create the partner link for the Rapid Distributors service.

1. Return to Oracle JDeveloper.
2. Select **Services** from the **Component Palette** section.

3. Drag and drop a **PartnerLink** activity from the **Component Palette** section onto the *right* side of the designer window.
4. Enter the following values to create a partner link for the Rapid Distributors service:

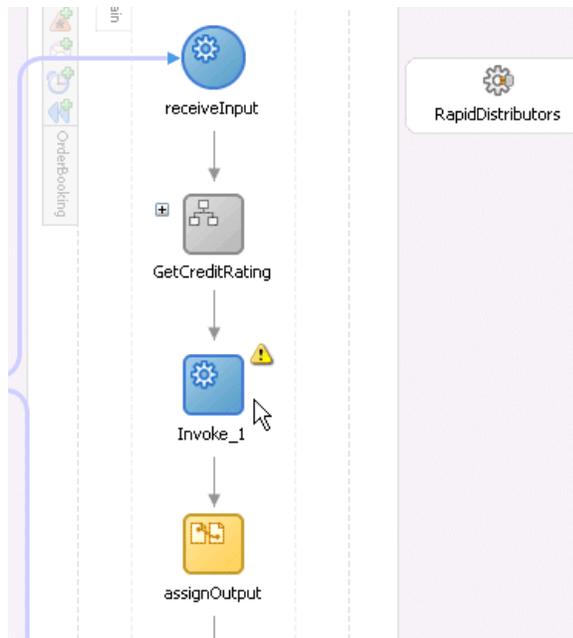
Note: For the WSDL File field below, click the **flashlight** (the second icon from the left named **Service Explorer**) to access the Service Explorer window for automatically selecting the Rapid Distributors service deployed in "[Starting and Testing Your Services](#)" on page 3-2.

| Field | Value |
|-------------------|--|
| Name | RapidDistributors |
| WSDL File | <p>Access this URL by clicking the Service Explorer flashlight icon and expanding and selecting BPEL Services > my_integration_server_connection > processes > default > RapidDistributors</p> <p>http://localhost:8888/orabpel/default/RapidDistributors/RapidDistributors?wsdl</p> <p>See Also: "Setting the Hostname in Your Oracle JDeveloper Web Browser Preferences" on page 1-7 if you receive a parsing error when attempting to add a WSDL file in the Service Explorer window.</p> |
| Partner Link Type | RapidDistributors |
| Partner Role | RapidDistributorsProvider |
| My Role | RapidDistributorsRequester |

5. Click **OK**.
6. Select **Save** from the **File** main menu.

Creating Invoke and Receive Activities

1. Select **Process Activities** from the **Component Palette** section.
2. Drag and drop an **Invoke** activity from the **Component Palette** section to below the **GetCreditRating Scope** activity.

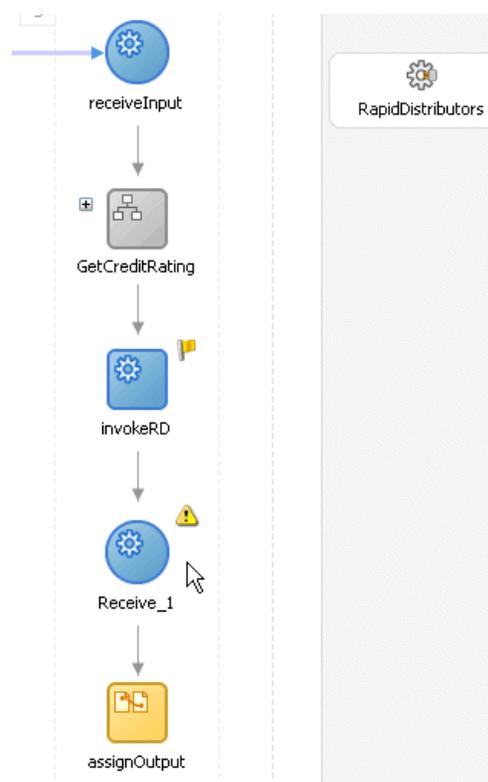


3. Double-click the **Invoke** icon to display the Invoke window.
4. Enter the following details:

| Field | Value |
|--------------|-------------------|
| Name | invokeRD |
| Partner Link | RapidDistributors |

The **Operation (initiate)** field is automatically filled in.

5. Click the first icon to the right of the **Input Variable** field. This is the automatic variable creation icon.
6. Click **OK** on the Create Variable window that appears.
A variable named **invokeRD_initiate_InputVariable** is automatically created in the **Input Variable** field. This variable is automatically assigned a message type of **RapidDistributorsRequestMessage**.
7. Click **OK**.
8. Drag and drop a **Receive** activity below the **invokeRD Invoke** activity you just created.



9. Double-click the **Receive** icon to display the Receive window.

10. Enter the following details:

| Field | Value |
|--------------|-------------------|
| Name | receiveRD |
| Partner Link | RapidDistributors |

The **Operation (onResult)** field is automatically filled in.

11. Click the first icon to the right of the **Variable** field.

12. Click **OK** on the Create Variable window that appears.

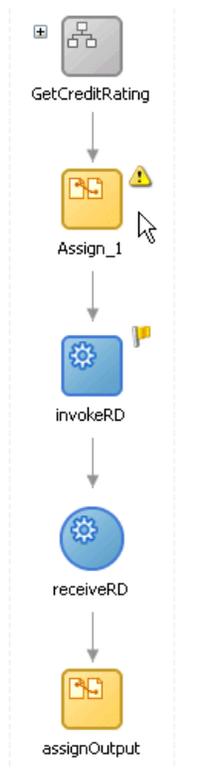
A variable named **receiveRD_onResult_InputVariable** is automatically created in the **Variable** field. This variable is automatically assigned a message type of **RapidDistributorsResponseMessage**.

13. Click **OK**.

14. Select **Save** from the **File** main menu.

Creating an Assign Activity

1. Drag and drop an **Assign** activity from the **Component Palette** section to above the **invokeRD Invoke** activity you just created.



2. Double-click the **assign** icon to display the Assign window.
3. Click the **General** tab.
4. Enter **assignRDIn** in the **Name** field.
5. Click **Apply**.
6. Click the **Copy Operation** tab.
7. Click **Create** and select **Copy Operation** to display the Create Copy Operation window.
8. Enter the following details:

| Field | Value |
|--------------------|---|
| From | |
| ▪ Type | Variable |
| ▪ Variables | Expand and select Variables > inputVariable > payload > ns1:PurchaseOrder > ns1:OrderItems Note: The namespace number values (for example, ns1 , ns2) can vary. Use the namespace values that automatically appear. |
| To | |
| ▪ Type | Variable |
| ▪ Variables | Expand and select Variables > invokeRD_initiate_InputVariable > payload > ns1:OrderItems |

9. Click **OK** to close the Create Copy Operation window and the Assign window.

10. Select **Save** from the **File** main menu. In this phase of the tutorial, you do not create an additional **Assign** activity to explicitly copy the received data of **invokeRD_initiate_InputVariable** into another variable.

When complete, the designer window looks as follows:



Validating, Compiling, and Deploying the Order Booking Process

Note: This tutorial describes how to deploy BPEL processes in Oracle JDeveloper. When you deploy a process with the same name and version, you are prompted to overwrite the currently-deployed BPEL process or deploy to a new version number. The best practice is to always deploy to a new version (as described in this tutorial). Redeploying a process with the same version can have consequences, such as instances being marked as stale.

If you deploy BPEL processes in this tutorial using `ant` from the command line, you are not prompted; instead, the process is automatically overwritten. Follow these steps to deploy a process to a new version:

1. Open the `SOA_Oracle_Home\bpel\utilities\ant-orabpel.properties` file.
 2. Change the `rev` property value. For example, change `rev = 1.0` to `rev = 2.0`.
 3. Deploy the process using the `ant` command.
-

1. Go to the **Application Navigator** section.

2. Right-click **OrderBooking**.
3. Select **Deploy > my_integration_server_connection > Deploy to default domain**.
The Deployment Properties window appears.
4. Increment the version number of the project (for example, enter **1.1**) and click **OK**.
When you access Oracle BPEL Control, a new version of the **OrderBooking** process appears, along with the phase 1 **OrderBooking** process version you created in "[Validating, Compiling, and Deploying the Order Booking Process](#)" on page 2-17.

This compiles the BPEL process. Check for errors by clicking the buttons at the bottom of the window. If there are no errors, deployment was successful. If deployment was unsuccessful, see Step 4 on page 2-17.

Running the Order Booking Process

1. Access Oracle BPEL Control by selecting **Start > All Programs > Oracle - Oracle_Home > Oracle BPEL Process Manager > BPEL Control** (or refresh the page if it is already open).
2. Enter the following details to log into Oracle BPEL Control and click **Login**:

| Field | Value |
|----------|-----------|
| Username | oc4jadmin |
| Password | password |

The **Dashboard** tab of Oracle BPEL Control appears. Note that two versions of the **OrderBooking** process appear: **1.0** and **1.1**. The new version, **1.1**, becomes the default version, so an asterisk appears to the right.

3. Click the **OrderBooking** version you increased in Step 4 on page 3-8 (for this example, (v. **1.1**)) in the **Deployed BPEL Processes** list.
4. Enter details in all fields of the HTML Form. You can also use a text editor to copy and paste these details from the contents of the `OrderBookingPO_1.xml` file you copied into the `OrderBooking` directory in Step 16 on page 2-5:

Note: Do *not* copy and paste from Internet Explorer or Mozilla Firefox. You must use a text editor to perform this task.

- a. Click the **XML Source** radio button.
- b. Copy and paste the contents of `OrderBookingPO_1.xml` into the field that appears.
5. Click **Post XML Message**.
The BPEL Processes tab displays a message similar to the following:
Test Instance Initiated
6. Click the **Instances** tab at the top.
7. Verify that the **OrderBooking**, **CreditRatingService**, and **RapidDistributors** instances were created. All should appear in the **Instance** list. The **Last Modified** list displays the times at which they were created.

8. Click the instances.
9. Click the **Audit** and **Flow** links to follow the instance execution process.
10. Click **More** to the right of **receiveRD** under the **Audit** link for the **OrderBooking** instance.
11. Click **view xml document** to see the returned supplier price (returned in the variable **receiveRD_onResult_InputVariable**).

```
<?xml version="1.0" encoding="UTF-8" ?>  
- <SupplierInfo xmlns="http://www.globalcompany.com/ns/sales">  
  <SupplierPrice>5000</SupplierPrice>  
  <SupplierName>RapidDistributors</SupplierName>  
</SupplierInfo>
```

12. Verify other process states and variable values at different stages.

Note: Throughout this tutorial, **RapidDistributors** always quotes a price of 5000.

Summary

You have now built a simple asynchronous BPEL process that received a supplier price for an order from a service. You performed the following key tasks:

- Deployed an *asynchronous* Rapid Distributors service.
- Created a partner link to interact with the Rapid Distributors service.
- Designed the BPEL process to invoke the Rapid Distributors service.
- Deployed the BPEL process.
- Ran the deployed BPEL process from Oracle BPEL Control by completing and submitting an order form to the Rapid Distributors service. Since the Rapid Distributors service was an asynchronous service, an extended time interval between the external service request and response is possible.
- Reviewed the supplier price returned from the Rapid Distributors supplier service to the client.

Adding Parallel Flow

This chapter of the tutorial describes how to build parallel flow logic to call another asynchronous service for receiving a supplier price.

This chapter contains the following topics:

- [Introduction](#)
- [Adding Parallel Flow to the Order Booking Tutorial](#)
- [Summary](#)

Introduction

This phase of the tutorial adds to the **OrderBooking** process you designed in [Chapter 3, "Invoking an Asynchronous Service"](#). Ensure that you have successfully completed that phase before performing this one. In this phase of the tutorial, you build parallel flow logic to call the asynchronous BPEL process to receive supplier prices from two services for a client order. You perform the following key tasks:

- Create a partner link to interact with a second asynchronous service (Select Manufacturing).
- Design parallel flow logic to call the two services (Rapid Distributors and Select Manufacturing).
- Design the BPEL process to invoke the Select Manufacturing service.
- Deploy the BPEL process.
- Run the deployed BPEL process from Oracle BPEL Control by completing and submitting an order form that is sent in parallel to the Rapid Distributors service and the Select Manufacturing service.
- Access the Select Manufacturing URL to manually submit a supplier price (the Rapid Distributors price is automated).
- Wait for both services to return price results.
- Review the supplier prices returned from the Rapid Distributors service and the Select Manufacturing service.

Adding Parallel Flow to the Order Booking Tutorial

This section contains the following tasks:

- [Starting and Testing Your Services](#)
- [Creating a Select Manufacturing Partner Link](#)

- [Creating a Flow Activity](#)
- [Creating Invoke and Receive Activities](#)
- [Creating an Assign Activity](#)
- [Validating, Compiling, and Deploying the Order Booking Process](#)
- [Running the Order Booking Process](#)

Starting and Testing Your Services

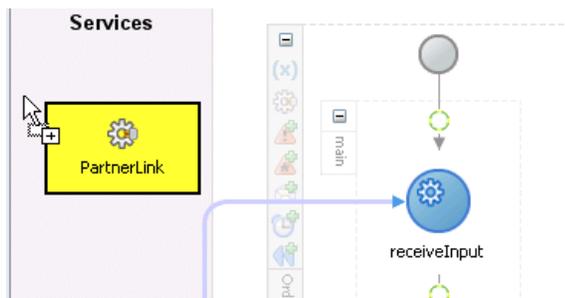
The Select Manufacturing service was automatically started when you started the Rapid Distributors service in "[Starting and Testing Your Services](#)" on page 3-2. If the Select Manufacturing service is not currently running, see that section for instructions on starting it.

1. Access the Select Manufacturing user application using Internet Explorer or Mozilla Firefox to see if it is deployed:
`http://localhost:8888/SelectManufacturingUI`
2. Log in as `jcooper/welcome1`.
 The Orders for Review window appears.
3. Click **Logout**.

Creating a Select Manufacturing Partner Link

Summary: You now create the partner link for the Select Manufacturing service.

1. Return to Oracle JDeveloper.
2. Select **Services** from the **Component Palette** section.
3. Drag and drop a **PartnerLink** activity from the **Component Palette** section onto the *left* side of the designer window (below the **Services** header and above the **client** partner link).



4. Enter the following values to create a partner link for the Select Manufacturing service:

Note: For the **WSDL File** field below, click the **flashlight** (the second icon from the left named **Service Explorer**) to access the Service Explorer window for automatically selecting the Select Manufacturing service deployed in "[Starting and Testing Your Services](#)" on page 3-2.

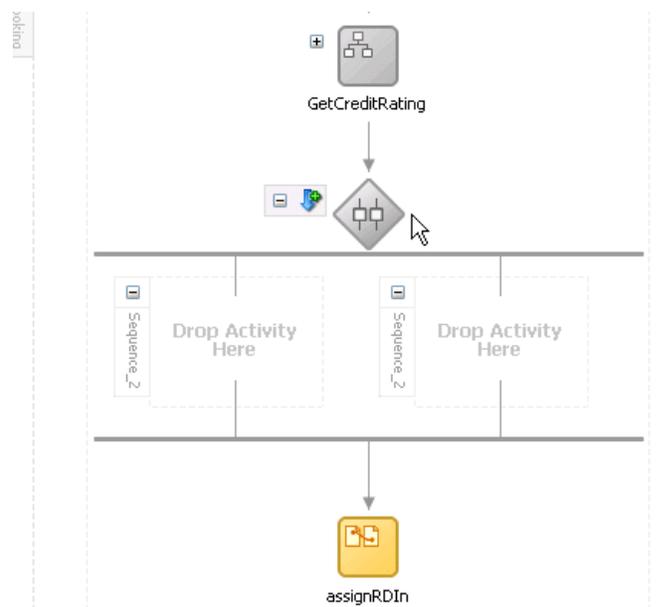
| Field | Value |
|-------------------|---|
| Name | SelectManufacturing |
| WSDL File | Access this URL by clicking the Service Explorer flashlight icon and expanding and selecting BPEL Services > my_integration_server_connection > processes > default > SelectManufacturing http://localhost:8888/orabpel/default/SelectManufacturing/SelectManufacturing?wsdl See Also: " Setting the Hostname in Your Oracle JDeveloper Web Browser Preferences " on page 1-7 if you receive a parsing error when attempting to add a WSDL file in the Service Explorer window. |
| Partner Link Type | SelectManufacturing |
| Partner Role | SelectManufacturingProvider |
| My Role | SelectManufacturingRequester |

5. Click **OK**.
6. Select **Save** from the **File** main menu.

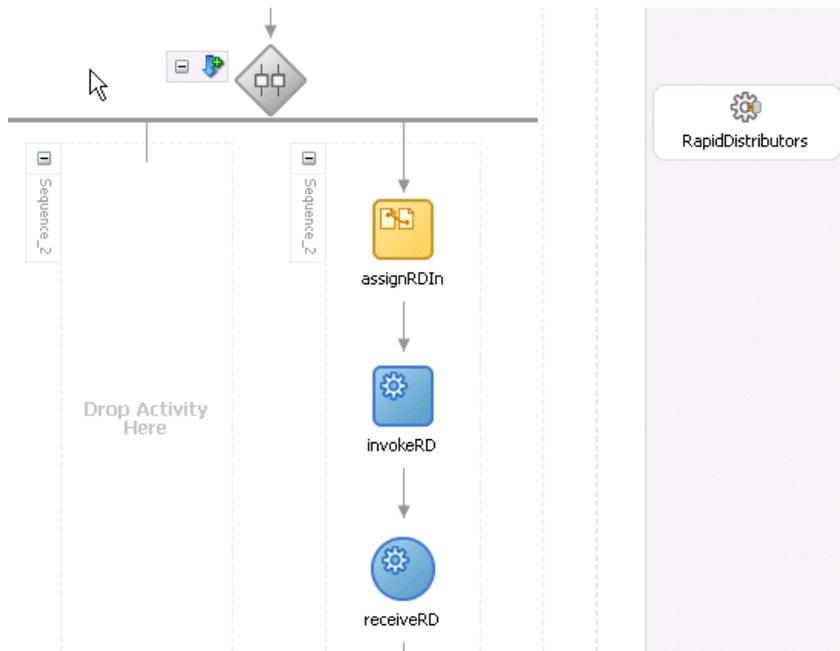
Creating a Flow Activity

Summary: You now create a **Flow** activity into which you place **Invoke** and **Receive** activities for both the Rapid Distributors and Select Manufacturing suppliers. A **Flow** activity is required to invoke the two suppliers in parallel. The order's line items are passed to the two suppliers for their price quotes.

1. Select **Process Activities** from the **Component Palette** section.
2. Drag and drop a **Flow** activity from the **Component Palette** section to below the **GetCreditRating Scope** activity.
3. Click the + sign to expand the Flow activity.



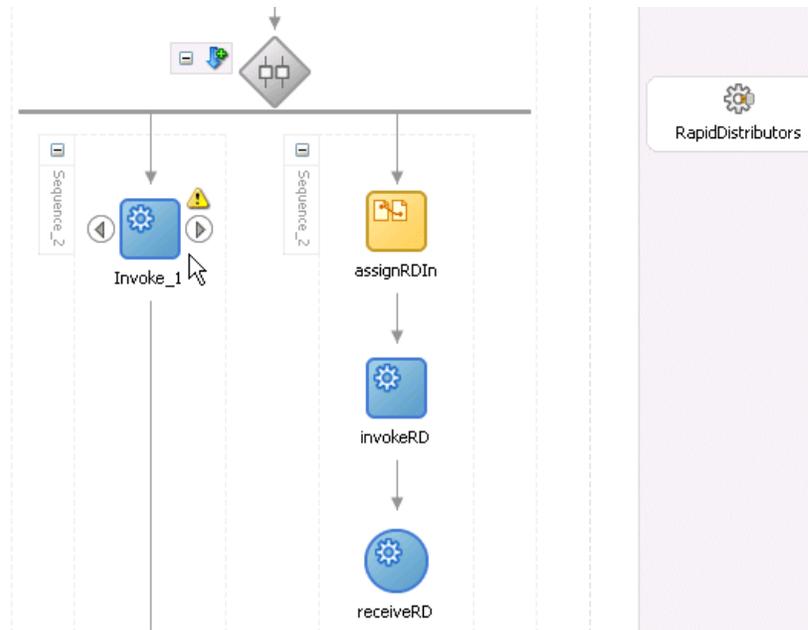
4. Double-click the diamond to display the Flow window.
5. Enter **GetPriceQuote** in the **Name** field.
6. Click **OK**.
7. Drag and drop the following activities that you created in [Chapter 3, "Invoking an Asynchronous Service"](#) into the *right* panel of the **GetPriceQuote Flow** activity. Drag each activity one at a time and place them in the following order.
 - **assignRDIn**
 - **invokeRD**
 - **receiveRD**



8. Select **Save** from the **File** main menu.

Creating Invoke and Receive Activities

1. Drag and drop an **Invoke** activity from the **Component Palette** section into the left panel of the **GetPriceQuote Flow** activity.

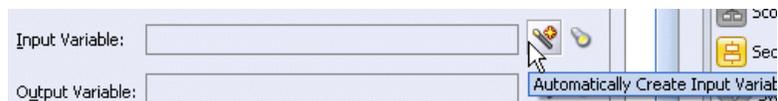


2. Double-click the **Invoke** icon to display the Invoke window.
3. Enter the following details:

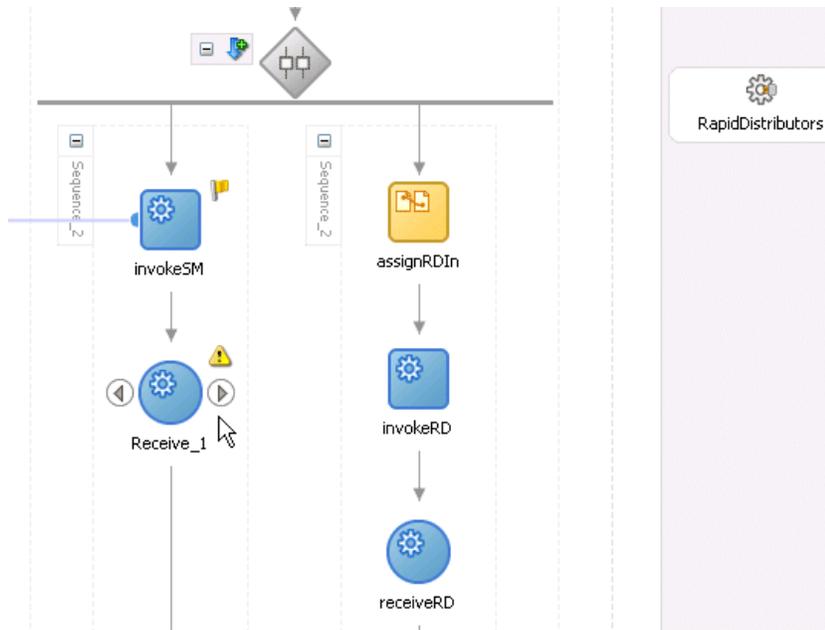
| Field | Value |
|--------------|---------------------|
| Name | invokeSM |
| Partner Link | SelectManufacturing |

The **Operation (initiate)** field is automatically filled in.

4. Click the first icon to the right of the **Input Variable** field. This is the automatic variable creation icon.



5. Click **OK** on the Create Variable window that appears.
 A variable named **invokeSM_initiate_InputVariable** is automatically created in the **Input Variable** field. This variable is automatically assigned a message type of **SelectManufacturingRequestMessage**.
6. Click **OK**.
7. Drag and drop a **Receive** activity below the **invokeSM Invoke** activity you just created.



8. Double-click the **Receive** icon to display the Receive window.
9. Enter the following details:

| Field | Value |
|--------------|---------------------|
| Name | receiveSM |
| Partner Link | SelectManufacturing |

The **Operation (onResult)** field is automatically filled in.

10. Click the first icon to the right of the **Variable** field. This is the automatic variable creation icon.
11. Click **OK** on the Create Variable window that appears.

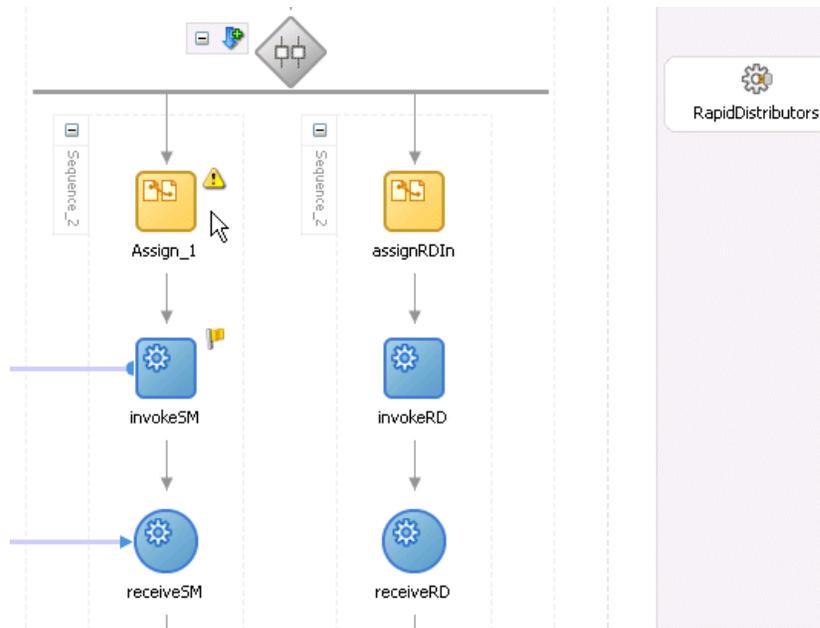
A variable named **receiveSM_onResult_InputVariable** is automatically created in the **Variable** field. This variable is automatically assigned a message type of **SelectManufacturingResponseMessage**.

12. Click **OK**.
13. Select **Save** from the **File** main menu.

Creating an Assign Activity

Summary: You now create an **Assign** activity that passes the customer's order line items to the Select Manufacturing supplier for their price quote.

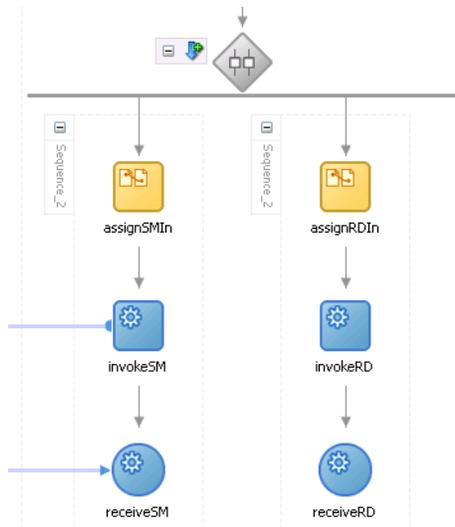
1. Drag and drop an **Assign** activity from the **Component Palette** section into the left panel of the **GetPriceQuote Flow** activity and above the **invokeSM Invoke** activity.



2. Double-click the **assign** icon to display the Assign window.
3. Click the **General** tab.
4. Enter **assignSMIn** in the **Name** field.
5. Click **Apply**.
6. Click the **Copy Operation** tab.
7. Click **Create** and select **Copy Operation** to display the Create Copy Operation window.
8. Enter the following details:

| Field | Value |
|--------------------|---|
| From | |
| ■ Type | Variable |
| ■ Variables | Expand and select Variables > inputVariable > payload > ns1:PurchaseOrder > ns1:OrderItems Note: The namespace number values (for example, ns1 , ns2) can vary. Use the namespace values that automatically appear. |
| To | |
| ■ Type | Variable |
| ■ Variables | Expand and select Variables > invokeSM_initiate_InputVariable > payload > ns1:OrderItems |

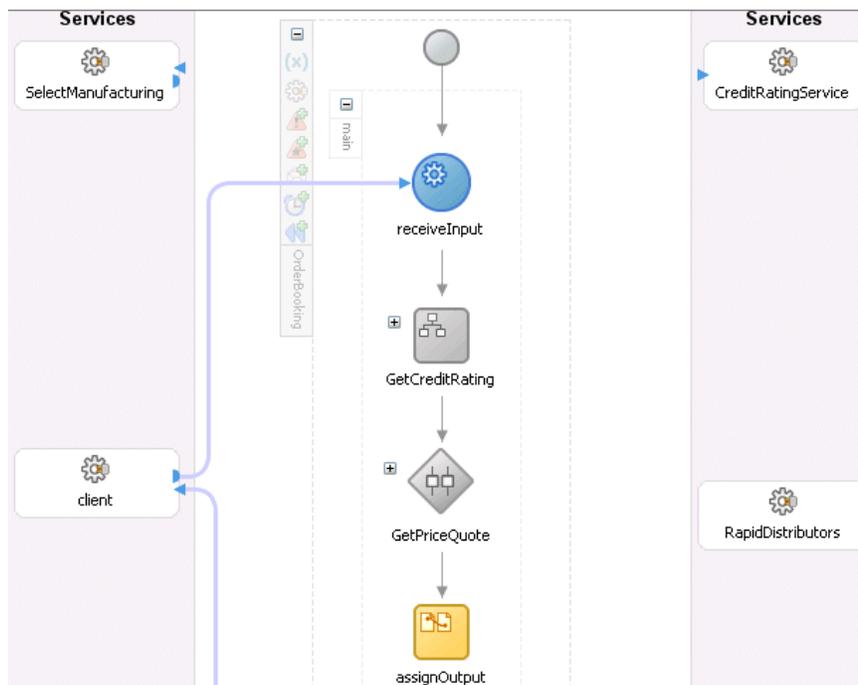
9. Click **OK** to close the Create Copy Operation window and the Assign window.
The designer window looks as follows:



10. Click the - sign to close the **GetPriceQuote Flow** activity. In this phase of the tutorial, you do not create an additional **Assign** activity to explicitly copy the received data of **invokeSM_initiate_InputVariable** into another variable.

11. Select **Save** from the **File** main menu.

When complete, the designer window looks as follows:



Validating, Compiling, and Deploying the Order Booking Process

1. Go to the **Application Navigator** section.
2. Right-click **OrderBooking**.
3. Select **Deploy > my_integration_server_connection > Deploy to default domain**.
4. Increment the version number again of the project when prompted and click **OK** (for example, enter **1.2**).

This compiles the BPEL process. Check for errors by clicking the buttons at the bottom of the window. If there are no errors, deployment was successful. If deployment was unsuccessful, see Step 4 on page 2-17.

Running the Order Booking Process

1. Access Oracle BPEL Control by selecting **Start > All Programs > Oracle - Oracle_Home > Oracle BPEL Process Manager > BPEL Control** (or refresh the page if it is already open).
2. Enter the following details to log into Oracle BPEL Control and click **Login**:

| Field | Value |
|----------|-----------------|
| Username | oc4jadmin |
| Password | <i>password</i> |

Where *password* is the value you entered during installation in "[Installing Oracle Application Server SOA Suite](#)" on page 1-4.

The **Dashboard** tab of Oracle BPEL Control appears.

3. Click the **OrderBooking** version you increased in Step 4 on page 4-8 in the **Deployed BPEL Processes** list.
4. Enter details in *all fields* of the HTML Form. You can also use a text editor to copy and paste these details from the contents of the `OrderBookingPO_1.xml` file:

Note: Do *not* copy and paste from Internet Explorer or Mozilla Firefox. You must use a text editor to perform this task.

- a. Click the **XML Source** radio button.
 - b. Copy and paste the contents of `OrderBookingPO_1.xml` into the field that appears.
5. Click **Post XML Message**.

The BPEL Processes tab displays a message similar to the following:

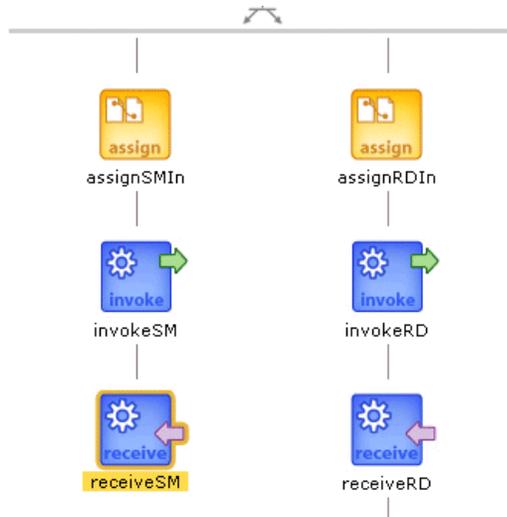
```
Test Instance Initiated
```

6. Click the **Instances** tab at the top.
7. Verify that the **OrderBooking**, **CreditRatingService**, **SelectManufacturing**, and **RapidDistributors** instances were created. All should appear in the **Instance** list. The **Last Modified** list displays the times at which they were created. The **CreditRatingService** and **RapidDistributors** instances complete relatively faster, since they are automated services. **SelectManufacturing** is running and awaiting user input to complete.
8. Click the **OrderBooking** instance you increased in Step 4 on page 4-8 in the **Instance** list.

A message appears stating the following:

```
The state of this instance is Active
```

9. Click the **Audit** link to follow the instance execution process. Note that the **SelectManufacturing** service is pending. The **RapidDistributors** service has completed.
10. Click the **Flow** link. Note that the parallel flow logic has completed on the **RapidDistributors** branch and is waiting on the **SelectManufacturing** branch (highlighted below).



To access this part of the activity, present it to the user for pricing, and accept the user data for task completion and process continuation, you must invoke the task from your browser.

The sample JSP pages and user presentation are prebuilt for you and deployed as `SelectManufacturingUI`. The JSP page displays a list of tasks waiting for user action.

11. Access the prebuilt JSP Select Manufacturing application using Internet Explorer or Mozilla Firefox.

`http://localhost:8888/SelectManufacturingUI`

12. Log in as `jcooper/welcome1`.
13. Click the order under **Title**.

Select Manufacturing
Async. Web Service
"Process On Demand"

You are logged as **jcooper** | [Order List](#) | [Logout](#)

Orders for Review

| Creator | Title | Created Date | Due Date |
|-----------|--|------------------------------------|------------------------------------|
| bpeladmin | Order Booking from Global Company, Order No:11 | Sat Jul 22 20:26:38 PDT 2006 | Mon Jul 24 20:26:38 PDT 2006 |

powered by Oracle

14. Provide a price in the **Price** field, and click **Quote**. This sends a form submit message on a JSP and completes the task. After processing, a message appears saying that the task has completed.

15. Click **Logout**.
16. Return to Oracle BPEL Control.
17. Click the **Instances** tab at the top.
18. Click the **OrderBooking** instance.
A message appears stating the following:
The state of this instance is Completed
19. Click the **Audit** and **Flow** links and observe **OrderBooking** process instance execution to the end (observe each supplier's price quote).

Summary

You have now built parallel flow logic to call the asynchronous BPEL process to receive supplier prices from two services for a client order. You performed the following key tasks:

- Created a partner link that interacted with a second asynchronous service (Select Manufacturing).
- Designed parallel flow logic that called the two services (Rapid Distributors and Select Manufacturing).
- Designed the BPEL process to invoke the Select Manufacturing service.
- Deployed the BPEL process.
- Ran the deployed BPEL process from Oracle BPEL Control by completing and submitting an order form that was sent in parallel to the Rapid Distributors service and the Select Manufacturing service.
- Accessed the Select Manufacturing URL to manually submit a supplier price (the Rapid Distributors price was automated and did not require a manual submittal).
- Waited for both services to return price results.
- Reviewed the supplier price returned from the Rapid Distributors service and the Select Manufacturing service.

Adding Conditional Branching Logic

This chapter of the tutorial describes how to build conditional branching logic.

This chapter contains the following topics:

- [Introduction](#)
- [Adding Conditional Branching Logic to the Order Booking Tutorial](#)
- [Summary](#)

Introduction

This phase of the tutorial adds to the **OrderBooking** process you designed in [Chapter 4, "Adding Parallel Flow"](#). Ensure that you have successfully completed that phase before performing this one. In this phase of the tutorial, you build conditional branching logic to select the lowest priced supplier. You perform the following key tasks:

- Design conditional branching logic to select the lower price from the two services (Rapid Distributors and Select Manufacturing).
- Deploy the BPEL process.
- Run the deployed BPEL process from Oracle BPEL Control by completing and submitting an order form that is sent in parallel to the Rapid Distributors service and the Select Manufacturing service.
- Access the Rapid Distributors URL to manually submit a supplier price.
- Review the lower of the two supplier prices returned from the Rapid Distributors service and the Select Manufacturing service.

Adding Conditional Branching Logic to the Order Booking Tutorial

This section contains the following tasks:

- [Creating a Switch Activity](#)
- [Creating an Assign Activity Under the Case Block for the Rapid Distributors Service](#)
- [Creating an Assign Activity Under the Otherwise Block for Select Manufacturing](#)
- [Validating, Compiling, and Deploying the Order Booking Process](#)
- [Running the Order Booking Process](#)

Creating a Switch Activity

Summary: To compare the two offers and make decisions based on that comparison, the BPEL flow requires a **Switch** activity. A **Switch** activity includes two branches: **<case>** and **<otherwise>**. The first branch is executed if a defined condition (inside the **<case>** branch) is met. If it is not met, the **<otherwise>** branch is executed.

1. Select **Process Activities** from the **Component Palette** section.
2. Drag and drop a **Switch** activity from the **Component Palette** section to below the **GetPriceQuote Flow** activity.
3. Double-click the question mark for the **Switch** activity to display the Switch window.
4. Enter **SelectSupplier** in the Name field.
5. Click **OK**.
6. Click the + sign to expand the Switch activity.
7. Double-click **<case>** to display the Switch Case window.
8. Enter the following expression in the **Name** field:

```
'Is the Rapid Distributors price less than the Select Manufacturing price?'
```

9. Press **Ctrl** and then the space bar in the **Expression** field to display a list for selecting (double-clicking) the following syntax. Edit as necessary. As you enter information, a trailing slash can appear. This means you are being prompted for additional information. Either enter additional information, or press the **Esc** key and delete the trailing slash to complete the input of information.

```
bpws:getVariableData('receiveRD_onResult_
InputVariable','payload','/ns1:SupplierInfo/ns1:SupplierPrice') <
bpws:getVariableData('receiveSM_onResult_
InputVariable','payload','/ns1:SupplierInfo/ns1:SupplierPrice')
```

10. Click **OK**.
11. Select **Save** from the **File** main menu.

Creating an Assign Activity Under the Case Block for the Rapid Distributors Service

Summary: This **Assign** activity assigns the Rapid Distributors supplier to the supplier part of the input purchase order variable.

1. Drag and drop an **Assign** activity from the **Component Palette** section under the **<case>** section of the **SelectSupplier Switch** activity.
2. Double-click the **Assign** activity.
3. Click the **General** tab.
4. Enter **SelectRapidDistributors** in the **Name** field.
5. Click **Apply**.
6. Click the **Copy Operation** tab.
7. Click **Create** and select **Copy Operation** to display the Create Copy Operation window.

8. Enter the following details:

Note: Instead of manually entering an expression, you can press **Ctrl** and then the space bar in the **Expression** field. Scroll through the list of values that appears and double-click the value you want. Edit the value as necessary. As you enter information, a trailing slash can appear. This means you are being prompted for additional information. Either enter additional information, or press the **Esc** key and delete the trailing slash to complete the input of information.

| Field | Value |
|---------------------|--|
| From | |
| ▪ Type | Expression |
| ▪ Expression | concat(bpws:getVariableData('inputVariable','payload','/ns1:PurchaseOrder/ns1:OrderInfo/ns1:OrderComments'),' - Selected: Rapid Distributors') |
| | Note: Press Ctrl and then the space bar to display a list for selecting (double-clicking) this syntax. The namespace number values (for example, ns1 , ns2) can vary. Use the namespace values that automatically appear. Remove any trailing forward slashes (/) that appear at the end. |
| To | |
| ▪ Type | Variable |
| ▪ Variables | Expand and select Variables > inputVariable > payload > ns1:PurchaseOrder > ns1:OrderInfo > ns1:OrderComments |

9. Click **OK** to close the Create Copy Operation window.
10. Click the **Copy Operation** tab.
11. Click **Create** and select **Copy Operation** to display the Create Copy Operation window.
12. Enter the following details to create a *second* statement:

| Field | Value |
|--------------------|--|
| From | |
| ▪ Type | Variable |
| ▪ Variables | Expand and select Variables > receiveRD_onResult_InputVariable > payload > ns1:SupplierInfo |
| | Note: The namespace number values (for example, ns1 , ns2) can vary. Use the namespace values that automatically appear. |
| To | |
| ▪ Type | Variable |
| ▪ Variables | Expand and select Variables > inputVariable > payload > ns1:PurchaseOrder > ns1:SupplierInfo |

13. Click **OK** to close the Create Copy Operation window and the Assign window.
14. Select **Save** from the **File** main menu.

Creating an Assign Activity Under the Otherwise Block for Select Manufacturing

Summary: This **Assign** activity assigns the Select Manufacturing supplier to the supplier part of the input purchase order variable.

1. Drag and drop an **Assign** activity from the **Component Palette** section under the **<otherwise>** section of the **SelectSupplier Switch** activity.
2. Double-click the **Assign** activity.
3. Click the **General** tab.
4. Enter **SelectManufacturing** in the **Name** field.
5. Click **Apply**.
6. Click the **Copy Operation** tab.
7. Click **Create** and select **Copy Operation** to display the Create Copy Operation window.
8. Enter the following details:

Note: Instead of manually entering an expression, you can press **Ctrl** and then the space bar in the **Expression** field. Scroll through the list of values that appears and double-click the value you want. Edit the value as necessary. As you enter information, a trailing slash can appear. This means you are being prompted for additional information. Either enter additional information, or press the **Esc** key and delete the trailing slash to complete the input of information.

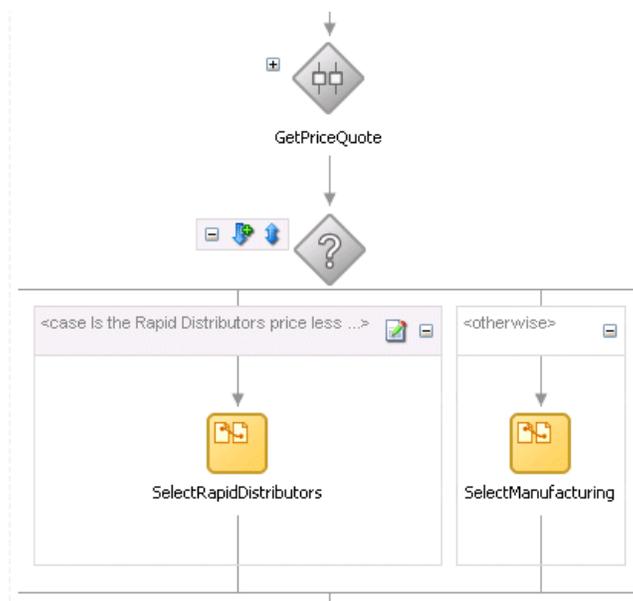
| Field | Value |
|---------------------|--|
| From | |
| ■ Type | Expression |
| ■ Expression | <code>concat(bpws:getVariableData('inputVariable','payload',/ns1:PurchaseOrder/ns1:OrderInfo/ns1:OrderComments'), ' - Selected: SelectManufacturing')</code> Note: Press Ctrl and then the space bar to display a list for selecting (double-clicking) this syntax. The namespace number values (for example, ns1 , ns2) can vary. Use the namespace values that automatically appear. Remove any trailing forward slashes (/) that appear at the end. |
| To | |
| ■ Type | Variable |
| ■ Variables | Expand and select Variables > inputVariable > payload > ns1:PurchaseOrder > ns1:OrderInfo > ns1:OrderComments |

9. Click **OK** to close the Create Copy Operation window.
10. Click the **Copy Operation** tab.
11. Click **Create** and select **Copy Operation** to display the Create Copy Operation window.
12. Enter the following details to create a *second* statement:

| Field | Value |
|-------------|---|
| From | |
| ▪ Type | Variable |
| ▪ Variables | Expand and select Variables > receiveSM_onResult_InputVariable > payload > ns1:SupplierInfo Note: The namespace number values (for example, ns1 , ns2) can vary. Use the namespace values that automatically appear. |
| To | |
| ▪ Type | Variable |
| ▪ Variables | Expand and select Variables > inputVariable > payload > ns1:PurchaseOrder > ns1:SupplierInfo |

- Click **OK** to close the Create Copy Operation window and the Assign window.

When complete, the designer window looks as follows:



- Click the - sign to close the **Switch** activity.
- Select **Save** from the **File** main menu.

Validating, Compiling, and Deploying the Order Booking Process

- Go to the **Application Navigator** section.
- Right-click **OrderBooking**.
- Select **Deploy > my_integration_server_connection > Deploy to default domain**.
- Increment the version number of the project when prompted and click **OK** (for example, enter **1.3**).

This compiles the BPEL process. Check for errors by clicking the buttons at the bottom of the window. If there are no errors, deployment was successful. If deployment was unsuccessful, see Step 4 on page 2-17.

Running the Order Booking Process

1. Access Oracle BPEL Control by selecting **Start > All Programs > Oracle - Oracle_Home > Oracle BPEL Process Manager > BPEL Control** (or refresh the page if it is already open).
2. Enter the following details to log into Oracle BPEL Control and click **Login**:

| Field | Value |
|----------|-----------------|
| Username | oc4jadmin |
| Password | <i>password</i> |

Where *password* is the value you entered during installation.

The **Dashboard** tab of Oracle BPEL Control appears.

3. Start the **OrderBooking** instance version you increased in Step 4 on page 5-5 in *one of the following ways*:
 - Click **OrderBooking** in the **Deployed BPEL Processes** list and enter details in all fields of the HTML Form and click **Post XML Message**.
 - Click **OrderBooking** in the **Deployed BPEL Processes** list and click the **XML Source** radio button. Use a text editor to copy and paste the contents of the `OrderBookingPO_1.xml` file into the field that appears. Click **Post XML Message**.

Note: Do *not* copy and paste from Internet Explorer or Mozilla Firefox. You must use a text editor to perform this task.

The **BPEL Processes** tab displays a message similar to the following:

```
Test Instance Initiated
```

4. Complete the manual price quote task for **SelectManufacturing** by using the graphical user interface JSP page, as described in Step 11 on page 4-10.
5. Return to Oracle BPEL Control.
6. Click the **Instances** tab at the top.
7. Verify that the **OrderBooking**, **CreditRatingService**, **SelectManufacturing**, and **RapidDistributors** instances were created. All should appear in the **Instance** list. The **Last Modified** list displays the times at which they were created.
8. Click the instances.
9. Click the **Audit** link to follow the instance execution process.
10. Click the **More...** link for each sequence to view the activity details.
11. Verify the results of conditional branching, the value of the price quote, and the execution path of the condition logic. (Similar instance management can also be obtained in a graphical fashion under the **Flow** link.)

Summary

You have now built conditional branching logic to select the lowest priced supplier. You performed the following key tasks:

- Designed conditional branching logic to select the lower price from the two services (Rapid Distributors and Select Manufacturing).
- Deployed the BPEL process.
- Ran the deployed BPEL process from Oracle BPEL Control by completing and submitting an order form that was sent in parallel to the Rapid Distributors service and the Select Manufacturing service.
- Accessed the Rapid Distributors URL to manually submit a supplier price.
- Reviewed the lower of the two supplier prices returned from the Rapid Distributors service and the Select Manufacturing service. After the two supplier services (Rapid Distributors and Select Manufacturing) returned with a price quote, a conditional branching decision was taken to select the lowest priced supplier.

Creating Fault Handling and Exception Management

This chapter of the tutorial describes how to build fault handling and exception management capabilities to catch exceptions.

This chapter contains the following topics:

- [Introduction](#)
- [Creating Fault Handling and Exception Management](#)
- [Summary](#)

Introduction

This phase of the tutorial adds to the **OrderBooking** process you designed in [Chapter 5, "Adding Conditional Branching Logic"](#). Ensure that you have successfully completed that phase before performing this one. In this phase of the tutorial, you build fault handling and exception management capabilities to identify a bad credit history. You perform the following key tasks:

- Design fault handling and exception management capabilities.
- Deploy the BPEL process.
- Run the deployed BPEL process from Oracle BPEL Control and submit an *invalid* social security card number to the Credit Rating service to check the client's credit history.
- Review the credit rating results returned from the Credit Rating service to the client to see if the invalid social security number is identified.

Creating Fault Handling and Exception Management

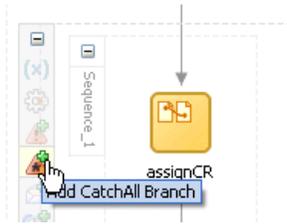
This section contains these tasks:

- [Creating a Catch All Branch in the Scope Activity](#)
- [Validating, Compiling, and Deploying the Order Booking Process](#)
- [Running the Order Booking Process](#)

Creating a Catch All Branch in the Scope Activity

1. Click the + sign to expand the **GetCreditRating Scope** activity.

- Click **Add Catch All Branch** in the icons (fourth from the top) on the left side of **GetCreditRating**.



- Click the **+** sign to expand the new branch.
A new box displays to the right with the words **Drop Activity Here**.
- Select **Process Activities** from the **Component Palette** section.
- Drag and drop an **Assign** activity from the **Component Palette** section into the **Catch** branch you just created.
- Double-click the **assign** icon to display the Assign window.
- Click the **General** tab.
- Enter **assignZeroCR** in the Name field.
- Click **Apply**.
- Click the **Copy Operation** tab.
- Click **Create** and select **Copy Operation** to display the Create Copy Operation window. You now create two fault handling rules.
- Enter the following values:

| Field | Value |
|--------------|--|
| From | |
| ▪ Type | Expression |
| ▪ Expression | '0' |
| To | |
| ▪ Type | Variable |
| ▪ Variables | Expand and select Variables > invokeCR_process_OutputVariable > payload > ns1:rating Note: The namespace number values (for example, ns1 , ns2) can vary. Use the namespace values that automatically appear. |

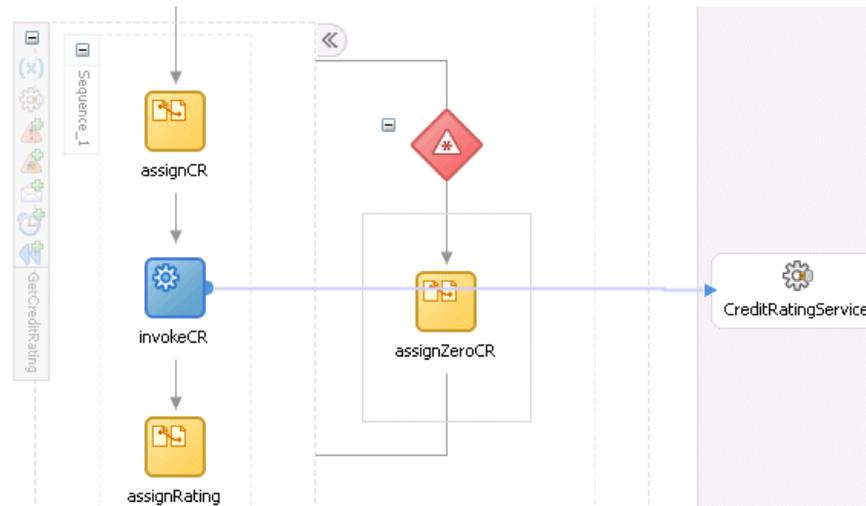
- Click **OK**.
- Click the **Copy Operation** tab.
- Click **Create** and select **Copy Operation** to display the Create Copy Operation window.
- Enter the following details to create a *second* statement:

| Field | Value |
|-------------|-------|
| From | |

| Field | Value |
|--------------|---|
| ■ Type | Expression |
| ■ Expression | ('Bad Credit, Rating = 0') |
| To | |
| ■ Type | Variable |
| ■ Variables | Expand and select Variables > inputVariable > payload > ns1:PurchaseOrder > ns1:OrderInfo > ns1:OrderComments |

17. Click **OK** to close the Create Copy Operation window and the Assign window.

When complete, the designer window looks as follows:



18. Click the - sign to close the **Scope** activity.

19. Select **Save** from the **File** main menu.

Validating, Compiling, and Deploying the Order Booking Process

1. Go to the **Application Navigator** section.
2. Right-click **OrderBooking**.
3. Select **Deploy > my_integration_server_connection > Deploy to default domain**.
4. Increment the version number of the project when prompted and click **OK** (for example, enter **1.4**).

This compiles the BPEL process. Check for errors by clicking the buttons at the bottom of the window. If there are no errors, deployment was successful. If deployment was unsuccessful, see Step 4 on page 2-17.

Running the Order Booking Process

You are now ready to test the fault handling and exception management capabilities.

1. Access Oracle BPEL Control by selecting **Start > All Programs > Oracle - Oracle_Home > Oracle BPEL Process Manager > BPEL Control** (or refresh the page if it is already open).
2. Enter the following details to log into Oracle BPEL Control and click **Login**:

| Field | Value |
|----------|-----------|
| Username | oc4jadmin |
| Password | password |

Where *password* is the value you entered during installation.

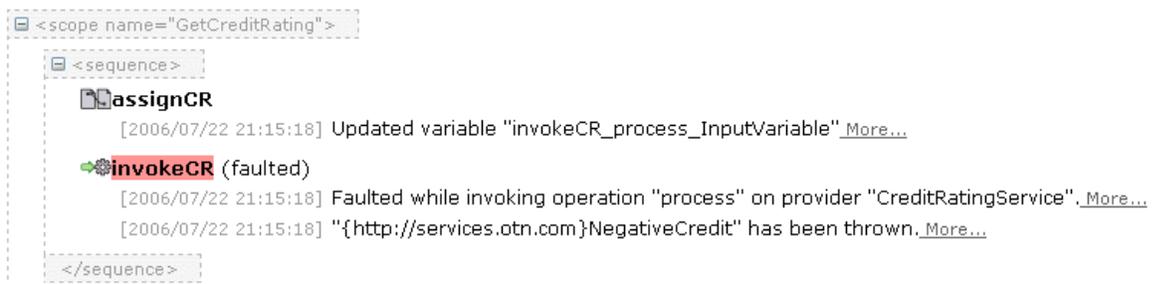
The **Dashboard** tab of Oracle BPEL Control appears.

3. Start the **OrderBooking** instance version you increased in Step 4 on page 6-3 in one of the following ways:
 - Click **OrderBooking** in the **Deployed BPEL Processes** list. Enter a nine-digit integer value that begins with zero in the **CustID** field of the HTML Form and click **Post XML Message** to post a message. This action invokes an exception in the Credit Rating service.
 - Open the provided CreateOrderBookingUI application at `http://localhost:8888/CreateOrderBookingUI` and provide inputs to initiate a process. Ensure that you overwrite the default value with zero in the **CustID** field.

The BPEL Processes tab displays a message similar to the following:

```
Test Instance Initiated
```

4. Complete the manual price quote task for **SelectManufacturing** by using the graphical user interface JSP page, as described in Step 11 on page 4-10.
5. Return to Oracle BPEL Control.
6. Click the **Instances** tab at the top.
7. Verify that the **OrderBooking** instance was created. The instance should appear in the **Instance** list. The **Last Modified** list displays the time at which the instance was created.
8. Click the **OrderBooking** instance.
9. Click the **Audit** link to follow the instance execution process.
10. Observe that the steps were executed and the exception was caught and handled in the process flow. Inside the scope block, a bad credit exception was raised, the exception was processed in the exception block, a default value of **0** was assigned, and processing continued.



11. Click the **Flow** link to observe additional details.

Summary

You have now built fault handling and exception management capabilities to catch exceptions in the Credit Rating service. You performed the following key tasks:

- Designed fault handling and exception management capabilities in the BPEL process.
- Deployed the BPEL process.
- Ran the deployed BPEL process from Oracle BPEL Control and deliberately submitted an invalid social security card number to the Credit Rating service to check the client's credit history.
- Reviewed the credit rating results returned from the Credit Rating service to the client. The invalid social security card number that you entered invoked the fault handling and exception management capabilities that you designed. A bad credit history message was returned to the client. The BPEL process was terminated before the Rapid Distributors and Select Manufacturing services were invoked.

Adding Transformation Logic

This chapter of the tutorial describes how to add transformation logic to a BPEL process.

This chapter contains the following topics:

- [Introduction](#)
- [Adding Transformation Logic](#)
- [Summary](#)

Introduction

This phase of the tutorial adds to the process you designed in [Chapter 6, "Creating Fault Handling and Exception Management"](#). Ensure that you have successfully completed that phase before performing this one. In this phase of the tutorial, you add transformation logic to a new BPEL process. You perform the following key tasks:

- Create a new BPEL process named **POAcknowledge**.
- Design the BPEL process to add transformation logic to transform incoming purchase order data into outgoing purchase order acknowledgment data.
- Test the transformation logic. This is a standalone test, and does not involve designing or deploying the complete process. You simply build and test the transformation logic from Oracle JDeveloper.

Adding Transformation Logic

This section contains the following tasks:

- [Creating the POAcknowledge Project and Importing the Schema](#)
- [Creating a Transform Activity](#)
- [Testing the Transformation Logic](#)

Creating the POAcknowledge Project and Importing the Schema

From Oracle JDeveloper, follow these instructions to create the POAcknowledge project and import the schema.

Caution: Do not include any special characters in the project name (such as periods) or in any activity or element names. If you do include special characters, errors appear when you attempt to compile your project.

1. Right-click **OrderBookapplication** and select **New Project**.
2. Under **Categories**, expand **General** and click **Projects**.
3. Under **Items**, click **BPEL Process Project** and click **OK**.
4. In the BPEL Project Creation Wizard - Project Settings window, enter the following details and click **Next**.

| Field | Value |
|------------------------------|--|
| Name | POAcknowledge |
| Namespace | Accept the default path. |
| Use Default Project Settings | Leave this checked to use the same name for the BPEL process and the BPEL project, and to use the default project directory. |
| Template | Asynchronous BPEL Process |

5. Click the **flashlight** icon to the right of the **Input Schema Element** field.
6. In the Select Schema window, select `OrderBookingPO.xsd` from the `SOA_Oracle_Home\bpel\samples\tutorials\127.OrderBookingTutorial\Practice Files` directory and click **Open**.
This imports the purchase order schema into your project.
The Type Chooser window appears.
7. Expand and select **Imported Schemas > OrderBookingPO.xsd > PurchaseOrder**.
8. Click **OK**.
The **POAcknowledgeRequestMessage** message type is automatically assigned an input message part name of **payload** and message part element of **PurchaseOrder**.
9. Click the **flashlight** icon to the right of the **Output Schema Element** field.
The Type Chooser window appears.
10. Click the first icon in the upper right corner (named **Import Schema File**).
The Import Schema window appears.
11. Click the **flashlight** icon to the right of the **URL** field.
12. In the Select Schema window, select `POAcknowledge.xsd` from the `SOA_Oracle_Home\bpel\samples\tutorials\127.OrderBookingTutorial\Practice Files` directory and click **Open**.
13. Click **OK** on the Import Schema File window.
This imports the purchase order schema into your project.
The Type Chooser window appears.

14. Expand and select the second **Imported Schemas > POAcknowledge.xsd > POAcknowledge**.

15. Click **OK**.

The **POAcknowledgeResponseMessage** message type is automatically assigned an output message part name of **payload** and message part element of **POAcknowledge**.

16. Click **Finish**.

The following files appear in the **Application Navigator**:

- The **bpel.xml**, **POAcknowledge.bpel**, and **POAcknowledge.wsdl** files appear under **POAcknowledge > Integration Content**.
- The **POAcknowledge.xsd** and **OrderBookingPO.xsd** files appear under **POAcknowledge > Integration Content > Schemas**.

17. Select **Save** from the **File** main menu.

18. If you want to view the message part name and element that were automatically assigned to your message types during BPEL project creation, ensure that **POAcknowledge.bpel** is displayed and perform the following steps.

- a. Expand **Message Types > Process WSDL - POAcknowledge.wsdl > POAcknowledgeRequestMessage > payload** in the **Structure** section.
- b. Right-click **payload** and select **Edit Message Part** to display the Edit Message Part window.
- c. Note that this defines the (input parameter) message part name (**payload**) of the **PurchaseOrder** element type.
- d. Click **Cancel**.
- e. Expand **Message Types > Process WSDL - POAcknowledge.wsdl > POAcknowledgeResponseMessage > payload** in the **Structure** section.
- f. Right-click **payload** and select **Edit Message Part** to display the Edit Message Part window.
- g. Note that this defines the (output parameter) message part name (**payload**) of the **POAcknowledge** element type.
- h. Click **Cancel**.

Creating a Transform Activity

Summary: You now use a **Transform** activity to create a transformation that maps incoming purchase order data into outgoing purchase order acknowledgment data.

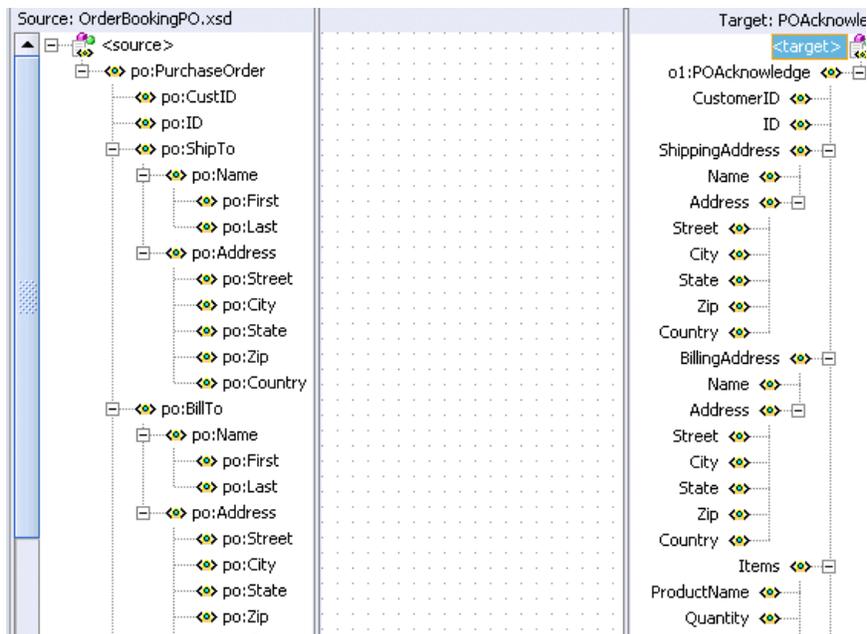
1. Drag and drop a **Transform** activity between the **receiveInput Receive** activity and the **callbackClient Invoke** activity.
2. Double-click the **Transform** activity.
3. Click the **General** tab.
4. Enter **transformPO** in the **Name** field.
5. Click the **Transformation** tab to display the Transformation window.

6. Select **inputVariable** in the **Source Variable** field (this automatically selects **payload** in the **Source Part** field).
7. Select **outputVariable** in the **Target Variable** field (this automatically selects **payload** in the **Target Part** field).
8. Leave the **Mapper File** field selection as **Transformation_1**.
9. Click the second icon (the **Create Mapping** icon) to the right of the **Mapper File** field.



A window for graphically mapping source and target elements appears. This window enables you to drag (map) a source element to a target element. If you instead receive a message saying the file already exists, click the third icon to access the transformation window.

10. Right-click the source and target schemas (the top-most elements in both lists) and select **Expand All**.



11. Drag and drop the following source elements to the following target elements. As you perform this task, a line connects each source to its target.

Note: The namespace values that appear for the source (for example, **ns1, po**) can vary. Use the namespace values that automatically appear.

| Drag the Source... | To the Target... |
|--------------------|------------------|
| po:CustID | CustomerID |

| Drag the Source... | To the Target... |
|--------------------|------------------|
| po:ID | ID |
| po:ShipTo | ShippingAddress |

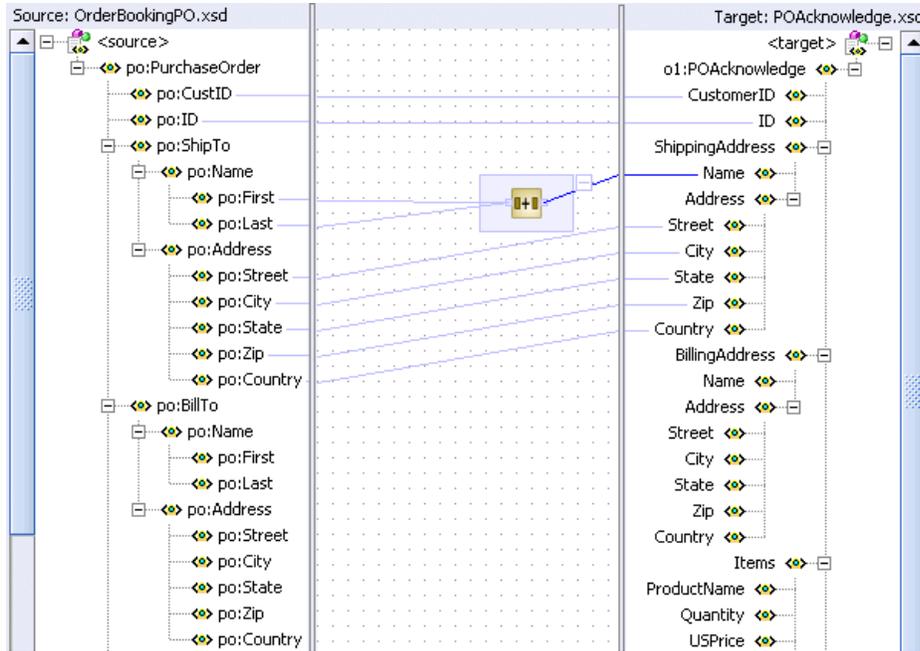
Note: Click **OK** when prompted to confirm your mapping. Connecting this source and target automatically connects all subelements.

The Transformation window appears as follows:

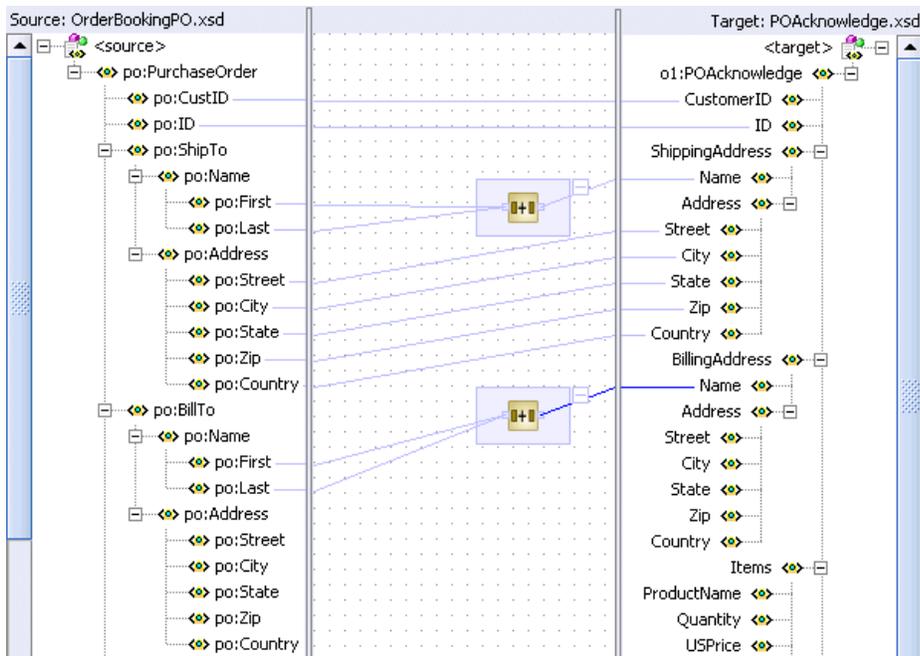


Note: If you drag and drop a source element to the wrong target element, right-click the connecting line and select **Delete**.

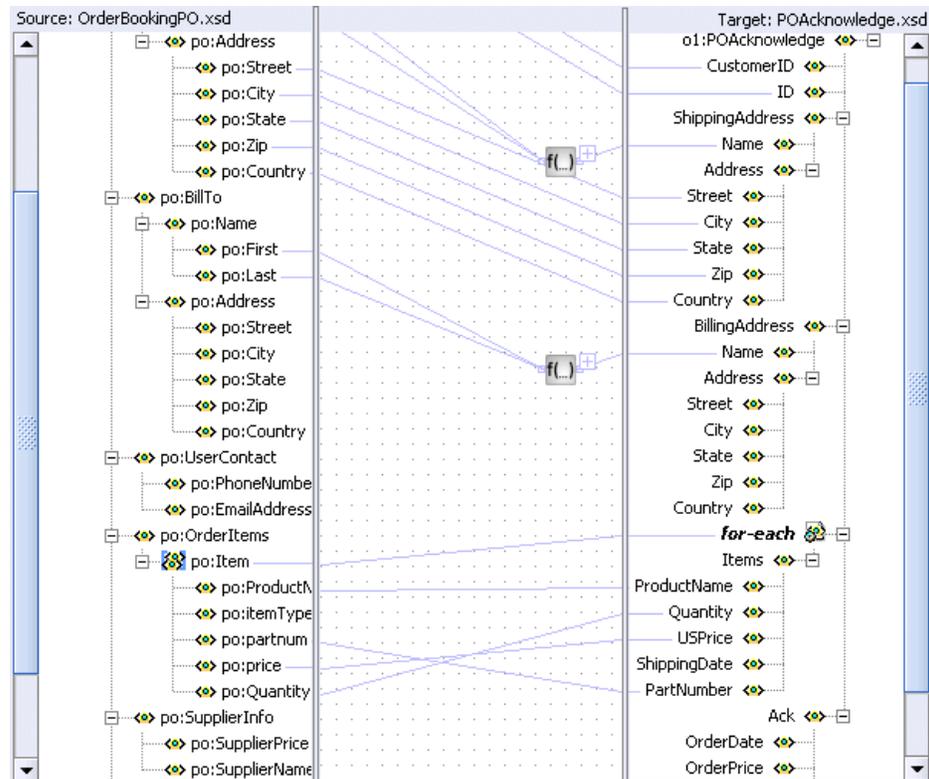
12. Select **String Functions** from the **Component Palette** section.
13. Drag and drop a **concat** function into the middle of the transformation window.
14. Go to the **po:ShipTo > po:Name** source section.
15. Drag the **po:First** and **po:Last** source elements below **po:ShipTo > po:Name** into the left side of the **concat** function. This connects two source lines to the **concat** function.
16. Go to the **ShippingAddress** target section.
17. Drag the **Name** target element into the right side of the **concat** function. This connects the target line to the **concat** function and concatenates the first and last names from the source elements into a single name in the target element.



18. Drag and drop another **concat** function into the middle of the transformation window to map source **BillTo** to target **BillingAddress**.
19. Go to the **po:BillTo > po:Name** source section.
20. Drag the **po:First** and **po:Last** source elements below **po:BillTo > po:Name** into the left side of the **concat** function. This connects two source lines to the **concat** function.
21. Go to the **BillingAddress** target section.
22. Drag the **Name** target element into the right side of the **concat** function. This connects the target line to the **concat** function and concatenates the first and last names from the source elements into a single name in the target element.



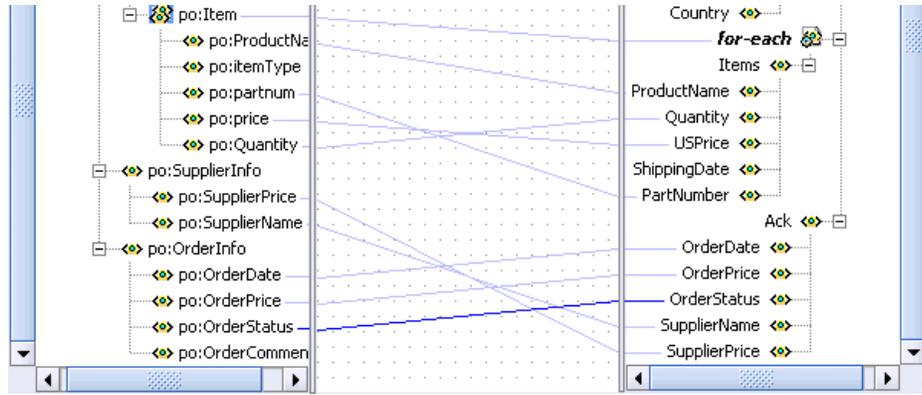
23. Select and right-click the target **Items**. This displays a list of XSL functions.
24. Select **Add XSL Node > for-each**.
The **for-each** node appears above the target **Items**.
25. Go to the source **po:OrderItems** section.
26. Drag and drop the source **po:Item** onto the **for-each** link.
27. Drag and drop the source **po:Item** onto the target **Items** (below the **for-each** link).
28. Click **OK** when prompted to confirm your mapping. This automatically maps all subelements.



This defines a relationship that specifies a one-to-many (1:M) mapping between the source and target.

29. Drag and drop the remaining source purchase order elements to the target purchase order acknowledge elements:

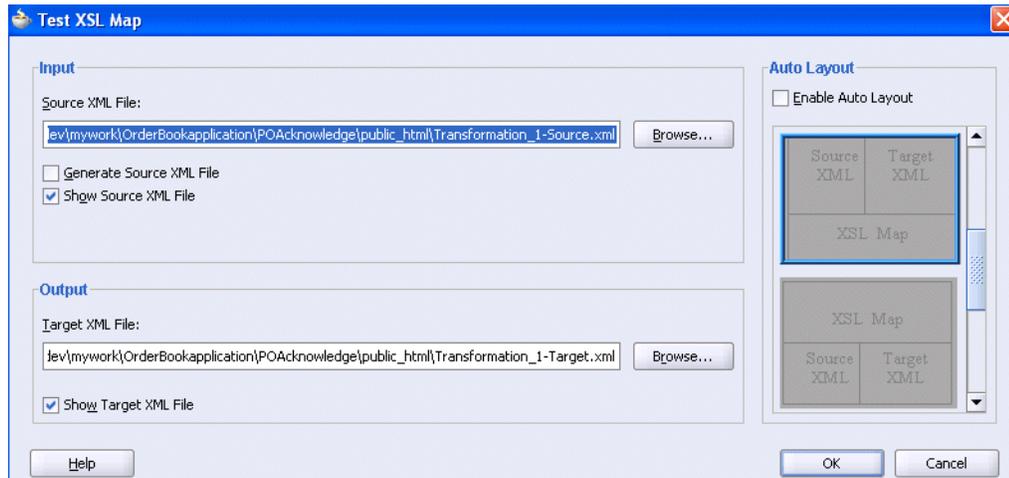
| Drag the Source... | To the Target... |
|--------------------|------------------|
| po:SupplierPrice | SupplierPrice |
| po:SupplierName | SupplierName |
| po:OrderDate | OrderDate |
| po:OrderPrice | OrderPrice |
| po:OrderStatus | OrderStatus |



30. Select **Save** from the **File** main menu.

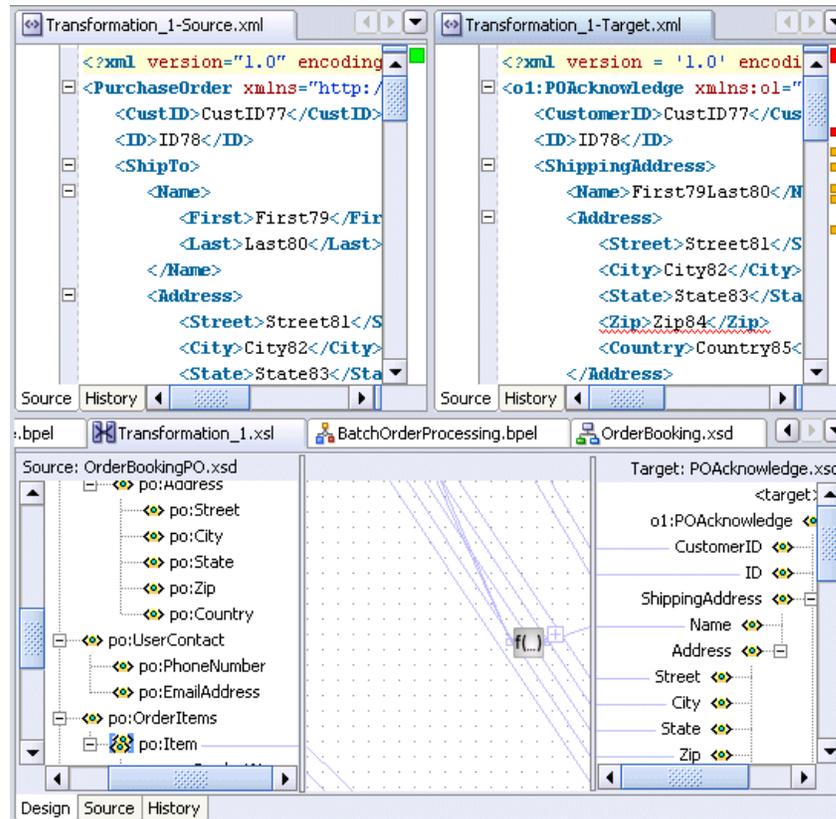
Testing the Transformation Logic

1. Right-click in the transformation mapping window and select **Test**.
This displays the Test XSL Map window.
2. Ensure that **Generate Source XML Data** is selected.
This creates sample XML test data.
3. Accept all other default settings.

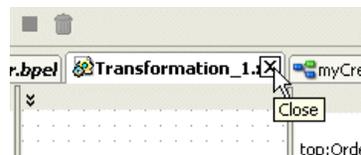


4. Click **OK** to test the translation logic.

Sample test results appear. The source XML and target XML display data side-by-side, with the XSL map data underneath. This is the default layout of test results.



5. Verify the results.
6. Select **Save** from the **File** main menu.
7. Close the **Transformation_1-Source.xml**, **Transformation_1-Target.xml**, and **Transformation_1.xsl** files by clicking the x button for each in the **Title** tab.



A translation file named **Transformation_1.xsl** appears under **POAcknowledge > Integration Content** in the **Application Navigator** section.

Summary

You have now added transformation logic to a BPEL process. You performed the following key tasks:

- Created a new BPEL process named **POAcknowledge**.
- Designed the BPEL process to add transformation logic that transformed incoming purchase order data into outgoing purchase order acknowledgment data.
- Tested the transformation logic and reviewed the results. You did not design or deploy the complete process. That comes in later phases of this tutorial.

Using the File Adapter's Read Functionality

This chapter of the tutorial describes how to add a file adapter to read data from a file and call the **OrderBooking** BPEL process as an external service.

This chapter contains the following topics:

- [Introduction](#)
- [Designing the File Adapter's Read Functionality to Activate a BPEL Process](#)
- [Summary](#)

Introduction

This phase of the tutorial adds to the process you designed in [Chapter 7, "Adding Transformation Logic"](#). Ensure that you have successfully completed that phase before performing this one. In this phase of the tutorial, you add a file adapter to read data from a file and call the previously created **OrderBooking** BPEL process as a service. You perform the following key tasks:

- Create a new BPEL process named **BatchOrderProcessing**.
- Create a partner link that uses the file adapter (named **FileRead**) to read and then delete files from a directory (in this case, an incoming purchase order).
- Create a partner link with which to interact (in this case, the **OrderBooking** BPEL process you previously designed is now used as a service).
- Design the BPEL process to invoke both **OrderBooking** and **FileRead**.
- Deploy the **BatchOrderProcessing** BPEL process.
- Run the deployed BPEL process from Oracle BPEL Control.
- Place a file in a directory and observe that the file is read and then deleted after **OrderBooking** returns a response.

Designing the File Adapter's Read Functionality to Activate a BPEL Process

This section contains the following tasks:

- [Creating a Batch Order Booking Process and Importing the Schema](#)
- [Creating a Partner Link that Uses the File Read Functionality of the File Adapter](#)
- [Creating an Order Booking Partner Link](#)
- [Creating Receive and Invoke Activities](#)

- [Creating an Assign Activity](#)
- [Validating, Compiling, and Deploying the Order Process](#)
- [Running and Verifying the Order Process](#)

Creating a Batch Order Booking Process and Importing the Schema

Follow these instructions to create a new Batch Order Booking project and import the schema. Note that in this project, you create an empty BPEL process, and *not* an asynchronous process.

Caution: Do not include any special characters in the project name (such as periods) or in any activity or element names. If you do include special characters, errors appear when you attempt to compile your project.

1. Right-click **OrderBookapplication** and select **New Project**.
2. Under **Categories**, expand **General** and click **Projects**.
3. Under **Items**, click **BPEL Process Project** and click **OK**.
4. In the BPEL Project Creation Wizard - Project Settings window, enter the following details and click **Finish**.

| Field | Value |
|------------------------------|--|
| Name | BatchOrderProcessing |
| Namespace | Accept the default path. |
| Use Default Project Settings | Leave this checked to use the same name for the BPEL process and the BPEL project, and to use the default project directory. |
| Template | Empty BPEL Process |

Note: Ensure that you select **Empty BPEL Process** from the **Template** list. This project is different from previous projects that you created.

This creates the **bpel.xml** and **BatchOrderProcessing.bpel** files in the **Application Navigator** section of the designer window.

5. Double-click **BatchOrderProcessing.bpel** in the **Application Navigator** section of the designer window to display the **Structure** section in the lower left section of the designer window.
6. Select and right-click **Schemas** in the **Structure** section.
7. Select **Import Schema**.
The Import Schema window appears.
8. Click the **flashlight** icon next to the URL field.
9. Select **OrderBookingPO.xsd** from the *SOA_Oracle_Home\bpel\samples\tutorials\127.OrderBookingTutorial\Practice Files* directory and click **Open**.

The file is added to the **URL** field of the Import Schema window.

10. Click **OK**.
11. Verify that **OrderBookingPO.xsd** appears under **BatchOrderProcessing > Integration Content > Schemas** in the **Application Navigator** section of the designer window. **OrderBookingPO.xsd** also appears under **Schemas** in the **Structure** section.

Creating a Partner Link that Uses the File Read Functionality of the File Adapter

Summary: You now create a partner link that uses the file adapter in the inbound direction. You configure the file adapter to check a specific directory every 15 seconds for files with a specified extension. This partner link serves as the input point for the process, and reads the files and provides input data (create instance) to the process.

1. Ensure that **Services** is selected in the drop-down list of the **Component Palette** section.
2. Drag and drop a **PartnerLink** activity onto the right side of the designer window.
3. Leave the **Name** field empty. A name is automatically created when you run the Adapter Configuration Wizard based on the service name you enter in Step 7.
4. Click the third icon at the top (the **Define Adapter Service** icon). This starts the Adapter Configuration Wizard.



5. Click **Next** on the Welcome window.
6. Select **File Adapter** on the Adapter Type window and click **Next**.
7. Enter **FileRead** in the **Service Name** field on the Service Name window and click **Next**.
8. Select **Read File** as the **Operation Type** on the Operation window and click **Next**. Note that the **Operation Name** field is automatically filled in with **Read**.
9. Select **Directory Names are Specified as Physical Path**.
10. Click **Browse** next to the **Directory for Incoming Files (physical path)** field.
11. Select the directory from which to read files (*drive_letter:\temp* on Windows operating systems or */temp* on UNIX operation systems). You must create this directory with the correct permissions if it does not exist.
12. Click **Select**.
13. Ensure that the **Archive processed files** check box is *not* selected.
14. Ensure that the **Delete files after successful retrieval** check box is selected.
15. Click **Next**.
16. Enter **OrderBooking*.xml** in the **Include Files with Name Pattern** field. This convention specifies that any input files with this pattern are selected for processing.
17. Click **Next**.

18. Enter **15 seconds** in the **Polling Frequency** field.
19. Click **Next**.
20. Click **Browse** next to the **Schema Location** field on the Messages window to display the Type Chooser window.
21. Select **Project Schema Files > OrderBookingPO.xsd > PurchaseOrder**.
22. Click **OK**.

The **Schema Location** field (**OrderBookingPO.xsd**) and the **Schema Element** field (**PurchaseOrder**) are filled in.

23. Click **Next**.
24. Click **Finish**.

You are returned to the Partner Link window. All other fields are automatically completed. The window looks as follows:

| Field | Value |
|-------------------|--|
| Name | FileRead |
| WSDL File | file:/c:/oraJDev/jdev/mywork/OrderBookapplication/BatchOrderProcessing/bpel/FileRead.wsdl. Note: This directory path with a drive letter represents an example on Windows operating systems. If running this tutorial on Unix operating systems, your directory path varies. In addition, oraJDev represents the Oracle JDeveloper home directory for this example. |
| Partner Link Type | Read_plt |
| Partner Role | Leave unspecified. |
| My Role | Read_role |

25. Click **OK**.
26. Select **Save** from the **File** main menu.

Note that **FileRead.wsdl** now appears under **BatchOrderProcessing > Integration Content > FileRead**. This file contains the parameters you specified with the Adapter Configuration Wizard.

Creating an Order Booking Partner Link

Summary: You now create a partner link to call the Order Booking process that you previously created as a BPEL process.

1. Drag and drop a **PartnerLink** activity from the **Component Palette** section onto the right side of the designer window.
2. Enter the following values to create a partner link for the Order Booking process:

Note: For the **WSDL File** field below, click the **flashlight** (the second icon from the left named **Service Explorer**) to access the Service Explorer window for automatically selecting the **OrderBooking** service.

| Field | Value |
|-------------------|--|
| Name | OrderBooking |
| WSDL File | Access this URL by clicking the Service Explorer flashlight icon and expanding and selecting BPEL Services > my_integration_server_connection > processes > default > OrderBooking http://localhost:8888/orabpel/default/OrderBooking/OrderBooking?wsdl See Also: " Setting the Hostname in Your Oracle JDeveloper Web Browser Preferences " on page 1-7 if you receive a parsing error when attempting to add a WSDL file in the Service Explorer window. |
| Partner Link Type | OrderBooking |
| Partner Role | OrderBookingProvider |
| My Role | OrderBookingRequester |

3. Click **OK**.
4. Select **Save** from the **File** main menu.

Creating Receive and Invoke Activities

Summary: You now create **Receive** and **Invoke** activities. The **Receive** activity reads data from a file. The **Invoke** activity initiates a process in **OrderBooking**.

1. Select **Process Activities** from the **Component Palette** section.
2. Drag and drop a **Receive** activity from the **Component Palette** section into the designer window. Because you created an empty BPEL process in "[Creating a Batch Order Booking Process and Importing the Schema](#)" on page 8-2, the designer window does not include or require the **receiveInput Receive** and **callbackClient Invoke** activities that are automatically created with an asynchronous BPEL process.
3. Double-click the **Receive** icon to display the Receive window.
4. Enter the following details:

| Field | Value |
|-----------------|------------------------|
| Name | receive_FileIn |
| Partner Link | FileRead |
| Create Instance | Select this check box. |

The **Operation (Read)** field is automatically filled in.

5. Click the first icon to the right of the **Variable** field. This is the automatic variable creation icon.
6. Click **OK** on the Create Variable window that appears.

A variable named **receive_FileIn_Read_InputVariable** is automatically created in the **Input Variable** field. This variable is automatically assigned a message type of **FileReadPurchaseOrder_msg**.
7. Ensure that you selected the **Create Instance** check box, as mentioned in Step 4.

8. Click **OK**.
9. Drag and drop an **Invoke** activity from the **Component Palette** section to below the **receive_FileIn Receive** activity.
10. Double-click the **Invoke** icon to display the Invoke window.
11. Enter the following details:

| Field | Value |
|--------------|---------------------|
| Name | invoke_OrderBooking |
| Partner Link | OrderBooking |

The **Operation (initiate)** field is automatically filled in.

12. Click the first icon to the right of the **Input Variable** field. This is the automatic variable creation icon.
13. Click **OK** on the Create Variable window that appears.
A variable named **invoke_OrderBooking_initiate_InputVariable** is automatically created in the **Input Variable** field. This variable is automatically assigned a message type of **OrderBookingRequestMessage**.
14. Click **OK**.
15. Select **Save** from the **File** main menu.

Creating an Assign Activity

Summary: You now create an **Assign** activity that passes the data read from a file to **invoke_OrderBooking_initiate_InputVariable**, the variable used for initiating activity with the **OrderBooking** process.

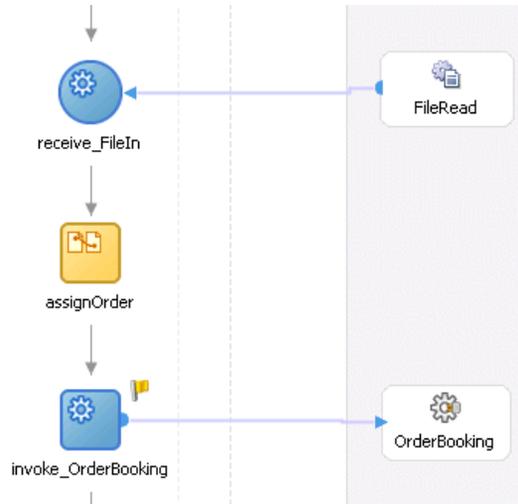
1. Drag and drop an **Assign** activity from the **Component Palette** section to below the **receive_FileIn Receive** activity.
2. Double-click the **assign** icon to display the Assign window.
3. Click the **General** tab.
4. Enter **assignOrder** in the **Name** field.
5. Click **Apply**.
6. Click the **Copy Operation** tab.
7. Click **Create** and select **Copy Operation** to display the Create Copy Operation window.
8. Enter the following details:

| Field | Value |
|-------------|--|
| From | |
| ▪ Type | Variable |
| ▪ Variables | Expand and select Variables > receive_FileIn_Read_InputVariable > PurchaseOrder |
| To | |

| Field | Value |
|-------------|--|
| ■ Type | Variable |
| ■ Variables | Expand and select Variables > invoke_OrderBooking_initiate_ InputVariable > payload |

- Click **OK** to close the Create Copy Operation window and the Assign window.
- Select **Save** from the **File** main menu.

When complete, the designer window looks as follows:



Validating, Compiling, and Deploying the Order Process

- Go to the **Application Navigator** section.
- Right-click **BatchOrderProcessing**.
- Select **Deploy** > *my_integration_server_connection* > **Deploy to the default domain**.

This compiles the BPEL process. Check for errors by clicking the buttons at the bottom of the window. If there are no errors, deployment was successful. If deployment was unsuccessful, see Step 4 on page 2-17.

Running and Verifying the Order Process

- Log into Oracle BPEL Control by selecting **Start** > **All Programs** > **Oracle - Oracle_Home** > **Oracle BPEL Process Manager** > **BPEL Control** (or refresh the page if it is already open).
- Enter the following details to log into Oracle BPEL Control and click **Login**:

| Field | Value |
|-----------------|-----------------|
| Username | oc4jadmin |
| Password | <i>password</i> |

Where *password* is the value you entered during installation in "Installing Oracle Application Server SOA Suite" on page 1-4.

The **Dashboard** tab of Oracle BPEL Control appears. Note that your BPEL process, **BatchOrderProcessing** and a series of subprocesses on which it depends, now appears in the Deployed BPEL Processes list.

3. Copy the `OrderBookingPO_1.xml` file from the `SOA_Oracle_Home\bpel\samples\tutorials\127.OrderBookingTutorial\PracticeFiles` directory to the directory you specified in Step 11 on page 8-3 (*drive_letter:\temp* on Windows operating systems or `/temp` on Unix operation systems).
4. Refresh Oracle BPEL Control after allowing the polling frequency of 15 seconds that you set in Step 18 on page 8-4 to elapse. After the process starts, the file is read, deleted (if reading was successful), and sent to the **OrderBooking** process. Processing completes after **OrderBooking** returns a response.
5. Complete the manual price quote task for **SelectManufacturing** by using the graphical user interface JSP page, as described in Step 11 on page 4-10.
6. Return to Oracle BPEL Control.
7. Click the **Instance** tab at the top.
8. Refresh Oracle BPEL Control and observe process instance creation and completion.
9. Click the instance.
10. Click the **Audit** link and click through other instances (of subprocesses) and follow the execution.
11. Click the **More** link for each sequence to view the activity details. (Similar instance management can also be obtained in a graphical fashion under the **Flow** link.)

Summary

You have now added a file adapter to read data from a file and call the previously created **OrderBooking** BPEL process as a service. You performed the following key tasks:

- Created a new BPEL process named **BatchOrderProcessing**.
- Created a partner link that used the file adapter (named **FileRead**) to read and then delete files from a directory (in this case, an incoming purchase order).
- Created a partner link with which to interact (in this case, the **OrderBooking** BPEL process you previously designed is now used as a service).
- Designed the BPEL process to invoke both **OrderBooking** and **FileRead**.
- Deployed the **BatchOrderProcessing** BPEL process.
- Ran the deployed BPEL process from Oracle BPEL Control.
- Placed a file in a directory and observed that the file was read and then deleted after **OrderBooking** returned a response.

Using the File Adapter's Write Functionality

This chapter of the tutorial describes how to add write functionality to the file adapter to accept input data in the form of a purchase order.

This chapter contains the following topics:

- [Introduction](#)
- [Designing the File Adapter's Write Functionality to Write the Order Acknowledgment](#)
- [Summary](#)

Introduction

This phase of the tutorial adds to the process you designed in [Chapter 8, "Using the File Adapter's Read Functionality"](#). Ensure that you have successfully completed that phase before performing this one. In this phase of the tutorial, you add write functionality to the file adapter and use the transformation logic you designed in [Chapter 7, "Adding Transformation Logic"](#). You perform the following key tasks:

- Return to the **POAcknowledge** BPEL process and create a partner link (named **FileOutInbound**) that uses the file adapter to write files to a directory (in this case, an outgoing purchase order acknowledgment).
- Design **POAcknowledge** to invoke **FileOutInbound**.
- Deploy **POAcknowledge**.
- Return to the **OrderBooking** BPEL process and create a partner link (named **WritePOA**) that interacts with **POAcknowledge**.
- Design **OrderBooking** to invoke **WritePOA**.
- Deploy the **OrderBooking** BPEL process.
- Run **OrderBooking** from Oracle BPEL Control by completing and submitting an order form to the Rapid Distributors and Select Manufacturing services.
- Copy the purchase order file to be read to a directory.
- Access the Rapid Distributors URL to manually submit a supplier price.
- Observe that the purchase order file is deleted and a purchase order acknowledgment file is written to the directory. Observe that the contents of the incoming purchase order file are transformed into an outgoing purchase order acknowledgment.

Designing the File Adapter's Write Functionality to Write the Order Acknowledgment

This section contains these tasks:

- [Creating a Partner Link that Uses the File Write Functionality of the File Adapter](#)
- [Creating an Invoke Activity](#)
- [Creating an Assign Activity](#)
- [Deploying the POAcknowledge Service](#)
- [Running and Verifying the POAcknowledge Process](#)
- [Creating a Partner Link](#)
- [Creating a Scope Activity](#)
- [Creating Invoke and Receive Activities](#)
- [Creating an Initial Assign Activity](#)
- [Creating a Second Assign Activity](#)
- [Validating, Compiling, and Deploying the Order Booking Process](#)
- [Running and Verifying the Order Booking Process](#)

Creating a Partner Link that Uses the File Write Functionality of the File Adapter

Summary: You now create a partner link that uses the file adapter in the outbound direction. You configure the file adapter to write a file to a specified directory. In this phase, you work from the **POAcknowledge** project.

1. Click the **POAcknowledge** project in the **Application Navigator** section of the designer window.
2. Double-click **POAcknowledge.bpel**.
3. Select **Services** from the **Component Palette** section.
4. Drag and drop a **PartnerLink** activity from the **Component Palette** section onto the right side of the designer window.
5. Leave the **Name** field empty. A name is automatically created when you run the Adapter Configuration Wizard based on the service name you enter in Step 9.
6. Click the third icon at the top (the **Define Adapter Service** icon). This starts the Adapter Configuration Wizard.



7. Click **Next** on the Welcome window.
8. Select **File Adapter** on the Adapter Type window and click **Next**.
9. Enter **FileWrite** in the **Service Name** field on the Service Name window and click **Next**.

10. Select **Write File** as the **Operation Type** on the Operation window and click **Next**. Note that the **Operation Name** field is automatically filled in with **Write**.
11. Select **Directory Specified as Physical Path**.
12. Click **Browse** next to the **Directory for Outgoing Files (physical path)** field.
13. Select the directory in which to create files (*drive_letter*: \temp on Windows operating systems or /temp on Unix operation systems). You must create this directory with the correct permissions if it does not exist.
14. Click **Select**.
15. Enter **po_%** in the **File Naming Convention** field.
A list of naming conventions automatically appears.
16. Select **yyMMddHHmmss** from the list that appears.
A closing % sign is automatically added.
17. Enter **.xml** to the end of your selection.
A name of **po_%yyMMddHHmmss%.xml** now appears in the **File Naming Convention** field.
18. Unselect the **Number of Messages Equal, Elapsed Time Exceeds, and File Size Exceeds** check boxes and click **Next**.
19. Click **Browse** next to the **Schema Location** field on the Messages window.
20. Select **Project Schema Files > POAcknowledge.xsd > POAcknowledge** in the Type Chooser window.
21. Click **OK**.
The **Schema Location** field (**POAcknowledge.xsd**) and the **Schema Element** field (**POAcknowledge**) are filled in.
22. Click **Next**.
23. Click **Finish**.
You are returned to the Partner Link window. All other fields are automatically completed. The window looks as follows:

| Field | Value |
|-------------------|---|
| Name | FileWrite |
| WSDL File | file:/c:/oraJDev/jdev/mywork/OrderBookapplication/POAcknowledge/bpel/FileWrite.wsdl Note: This directory path with a drive letter represents an example on Windows operating systems. If running this tutorial on Unix operating systems, your directory path varies. In addition, oraJDev represents the Oracle JDeveloper home directory for this example. |
| Partner Link Type | Write_plt |
| Partner Role | Write_role |
| My Role | Leave unspecified. |

24. Click **OK**.
25. Select **Save** from the **File** main menu.

Note that **FileWrite.wsdl** now appears under **POAcknowledge > Integration Content > FileWrite**. This file contains the parameters you specified with the Adapter Configuration Wizard.

Creating an Invoke Activity

Summary: This **Invoke** activity writes order acknowledgment data results into a file.

1. Select **Process Activities** from the **Component Palette** section.
2. Drag and drop an **Invoke** activity from the **Component Palette** section to below the **transformPO Transform** activity.
3. Double-click the **Invoke** icon to display the Invoke window.
4. Enter the following details:

| Field | Value |
|--------------|---------------|
| Name | invokeFileOut |
| Partner Link | FileWrite |

The **Operation (Write)** field is automatically filled in.

5. Click the first icon to the right of the **Input Variable** field. This is the automatic variable creation icon.
6. Click **OK** on the Create Variable window that appears.

A variable named **invokeFileOut_write_InputVariable** is automatically created in the **Input Variable** field. This variable is automatically assigned a message type of **FileWritePOAcknowledge_msg**.

7. Click **OK**.
8. Select **Save** from the **File** main menu.

Creating an Assign Activity

1. Drag and drop an **Assign** activity from the **Component Palette** section to above the **invokeFileOut Invoke** activity.
2. Double-click the **assign** icon to display the Assign window.
3. Click the **General** tab.
4. Enter **assignFileWriteIn** in the **Name** field.
5. Click **Apply**.
6. Click the **Copy Operation** tab.
7. Click **Create** and select **Copy Operation** to display the Create Copy Operation window.
8. Enter the following values:

| Field | Value |
|-------|-------|
| From | |

| Field | Value |
|-------------|--|
| ■ Type | Variable |
| ■ Variables | Expand and select Variables > outputVariable > payload |
| To | |
| ■ Type | Variable |
| ■ Variables | Expand and select Variables > invokeFileOut_Write_InputVariable > POAcknowledge |

9. Click **OK** to close the Create Copy Operation window and the Assign window.
10. Select **Save** from the **File** main menu.

Deploying the POAcknowledge Service

1. Go to the **Application Navigator** section.
2. Right-click **POAcknowledge**.
3. Select **Deploy** > *my_integration_server_connection* > **Deploy to default domain**.

This compiles the BPEL process. Review the bottom of the window for any errors. If there are no errors, deployment was successful. If deployment was unsuccessful, see Step 4 on page 2-17.

Running and Verifying the POAcknowledge Process

1. Log into Oracle BPEL Control by selecting **Start** > **All Programs** > **Oracle - Oracle_Home** > **Oracle BPEL Process Manager** > **BPEL Control** (or refresh the page if it is already open).
2. Enter the following details to log into Oracle BPEL Control and click **Login**:

| Field | Value |
|----------|-----------------|
| Username | oc4jadmin |
| Password | <i>password</i> |

Where *password* is the value you entered during installation.

The **Dashboard** tab of Oracle BPEL Control appears. Note that your BPEL process, **POAcknowledge**, now appears in the **Deployed BPEL Processes** list.

3. Click **POAcknowledge**.
4. Enter details in *all fields* of the HTML Form and click **Post XML Message**.
5. Refresh Oracle BPEL Control.
6. Click the **Instances** tab.
7. Click the **POAcknowledge** instance.
8. Click the **Audit** link to view the transformation logic.
9. View the output file written in the directory you selected in Step 13 on page 9-3 (*drive_letter*:\temp on Windows operating systems or /temp on Unix operation systems).

Creating a Partner Link

Summary: You now extend the original **OrderBooking** process to call the newly-created **POAcknowledge** service to write a PO acknowledgment in a file. You perform these steps in the **OrderBooking** process.

1. Expand the **OrderBooking** process in the **Application Navigator** section.
2. Double-click **OrderBooking.bpel**.
3. Select **Services** from the **Component Palette** section.
4. Drag and drop a **PartnerLink** activity from the **Component Palette** section onto the right side of the designer window.
5. Enter the following values to create a partner link for the **OrderBooking** process:

Note: For the **WSDL File** field below, click the **flashlight** (the second icon from the left named **Service Explorer**) to access the Service Explorer window for automatically selecting a purchase order acknowledgment service.

| Field | Value |
|-------------------|---|
| Name | WritePOA Note: Enter WritePOA after selecting the WSDL File. This is because the name automatically defaults to POAcknowledge based on your selection in the WSDL File field. |
| WSDL File | Access this URL by clicking the Service Explorer flashlight icon and expanding and selecting BPEL Services > my_integration_server_connection > processes > default > POAcknowledge http://localhost:8888/orabpel/default/POAcknowledge/POAcknowledge?wsdl See Also: " Setting the Hostname in Your Oracle JDeveloper Web Browser Preferences " on page 1-7 if you receive a parsing error when attempting to add a WSDL file in the Service Explorer window. |
| Partner Link Type | POAcknowledge |
| Partner Role | POAcknowledgeProvider |
| My Role | POAcknowledgeRequester |

6. Click **OK**.
7. Select **Save** from the **File** main menu.

Creating a Scope Activity

Summary: You now create a **Scope** activity to group all activities that form a logical step to be executed.

1. Select **Process Activities** from the **Component Palette** section.
2. Drag and drop a **Scope** activity from the **Component Palette** section to below the **SelectSupplier Switch** activity.

3. Double-click the **scope** icon to display the Scope window.
4. Enter **WritePOAcknowledgment** in the **Name** field of the **General** tab.
5. Click **OK**.

In this phase of the tutorial, you create the **Invoke**, **Receive**, and **Assign** activities inside this **Scope** activity.

6. Click the + sign to expand the **Scope** activity.
7. Select **Save** from the **File** main menu.

Creating Invoke and Receive Activities

Summary: You now create **Invoke** and **Receive** activities. The **Invoke** activity provide an initiate operation that sends input data. The **Receive** activity provides a callback interface to receive data after the external service is processed.

1. Drag and drop an **Invoke** activity from the **Component Palette** section to inside the **WritePOAcknowledgment Scope** activity.
2. Double-click the **Invoke** icon to display the Invoke window.
3. Enter the following details:

| Field | Value |
|--------------|-----------------|
| Name | invokeFileWrite |
| Partner Link | WritePOA |

The **Operation (initiate)** field is automatically filled in.

4. Click the first icon to the right of the **Input Variable** field. This is the automatic variable creation icon.
5. Click **OK** on the Create Variable window that appears.

A variable named **invokeFileWrite_initiate_InputVariable** is automatically created in the **Input Variable** field. This variable is automatically assigned a message type of **POAcknowledgeRequestMessage**.

6. Click **OK**.
7. Drag and drop a **Receive** activity from the **Component Palette** section to below the **invokeFileWrite Invoke** activity inside the **WritePOAcknowledgment Scope** activity.
8. Double-click the **Receive** icon to display the Receive window.
9. Enter the following details:

| Field | Value |
|--------------|------------------|
| Name | receiveFileWrite |
| Partner Link | WritePOA |

The **Operation (onResult)** field is automatically filled in.

10. Click the first icon to the right of the **Variable** field. This is the automatic variable creation icon.
11. Click **OK** on the Create Variable window that appears.
 A variable named **receiveFileWrite_onResult_InputVariable** is automatically created in the **Output Variable** field. This variable is automatically assigned a message type of **POAcknowledgeResponseMessage**.
12. Click **OK**.
13. Select **Save** from the **File** main menu.

Creating an Initial Assign Activity

1. Drag and drop an **Assign** activity from the **Component Palette** section to above the **invokeFileWrite Invoke** activity inside the **WritePOAcknowledgment Scope** activity.
2. Double-click the **assign** icon to display the Assign window.
3. Click the **General** tab.
4. Enter **assignFileWriteIn** in the **Name** field.
5. Click **Apply**.
6. Click the **Copy Operations** tab.
7. Click **Create** and select **Copy Operation** to display the Create Copy Operation window.
8. Enter the following values:

| Field | Value |
|--------------------|---|
| From | |
| ▪ Type | Variable |
| ▪ Variables | Expand and select Variables > inputVariable > payload |
| To | |
| ▪ Type | Variable |
| ▪ Variables | Expand and select Variables > invokeFileWrite_initiate_InputVariable > payload |

9. Click **OK** to close the Create Copy Operation window and the Assign window.
10. Select **Save** from the **File** main menu.

Creating a Second Assign Activity

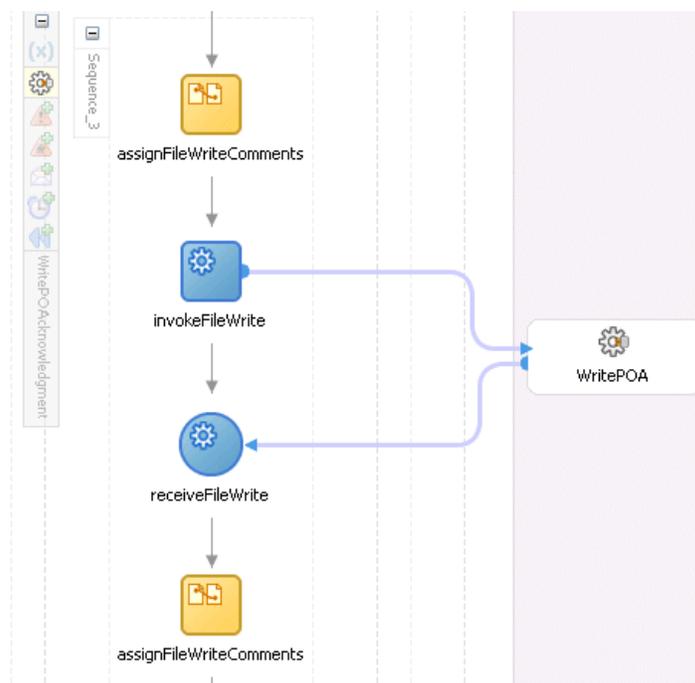
1. Drag and drop an **Assign** activity from the **Component Palette** section to below the **receiveFileWrite Receive** activity inside the **WritePOAcknowledgment Scope** activity.
2. Double-click the **assign** icon to display the Assign window.
3. Click the **General** tab.
4. Enter **assignFileWriteComments** in the **Name** field.
5. Click **Apply**.

6. Click the **Copy Operation** tab.
7. Click **Create** and select **Copy Operation** to display the Create Copy Operation window.
8. Enter the following values:

Note: Instead of manually entering an expression, you can press **Ctrl** and then the space bar in the **Expression** field. Scroll through the list of values that appears and double-click the value you want. Edit the value as necessary. As you enter information, a trailing slash can appear. This means you are being prompted for additional information. Either enter additional information, or press the **Esc** key and delete the trailing slash to complete the input of information.

| Field | Value |
|---------------------|---|
| From | |
| ▪ Type | Expression |
| ▪ Expression | <code>concat(bpws:getVariableData('inputVariable','payload','/ns1:PurchaseOrder/ns1:OrderInfo/ns1:OrderComments'),' - Write POAck complete')</code> |
| To | |
| ▪ Type | Variable |
| ▪ Variables | Expand and select Variables > inputVariable > payload > ns1:PurchaseOrder > ns1:OrderInfo > ns1:OrderComments |

9. Click **OK** to close the Create Copy Operation window and the Assign window.
When complete, the designer window looks as follows:



10. Click the - sign to close the **WritePOAcknowledgment Scope** activity.
11. Select **Save** from the **File** main menu.

Validating, Compiling, and Deploying the Order Booking Process

1. Go to the **Application Navigator** section.
2. Right-click **OrderBooking**.
3. Select **Deploy > my_integration_server_connection > Deploy to default domain**.
4. Increment the version number of the project when prompted and click **OK** (for example, enter 1.5).

This compiles the BPEL process. Check for errors by clicking the buttons at the bottom of the window. If there are no errors, deployment was successful.

Running and Verifying the Order Booking Process

1. Log into Oracle BPEL Control by selecting **Start > All Programs > Oracle - Oracle_Home > Oracle BPEL Process Manager > BPEL Control** (or refresh the page if it is already open).
2. Enter the following details to log into Oracle BPEL Control and click **Login**:

| Field | Value |
|----------|-----------|
| Username | oc4jadmin |
| Password | password |

The **Dashboard** tab of Oracle BPEL Control appears. Note that your BPEL processes, **POAcknowledge** and **OrderBooking**, appear in the **Deployed BPEL Processes** list.

3. Click the **OrderBooking** instance version you increased in Step 4 on page 9-10 and start it in one of the following ways:
 - Open the provided **CreateOrderBookingUI** application at `http://localhost:8888/CreateOrderBookingUI` and provide inputs to initiate a process.
 - Enter details in all fields of the HTML Form and click **Post XML Message**.
 - Click the **XML Source** radio button. Use a text editor to copy and paste the contents of the `OrderBookingPO_1.xml` file into the field that appears. Click **Post XML Message**.

Note: Do *not* copy and paste from Internet Explorer. You must use a text editor to perform this task.

4. Copy the `OrderBookingPO_1.xml` file to the directory you previously selected to invoke the process instance (`drive_letter:\temp` on Windows operating systems or `/temp` on Unix operation systems). This step is optional if the **BatchOrderProcessing** process of section "[Running and Verifying the Order Process](#)" on page 8-7 is complete.
5. Complete the manual price quote task for **SelectManufacturing** by using the graphical user interface JSP page, as described in Step 11 on page 4-10.

6. Return to Oracle BPEL Control.
7. Refresh Oracle BPEL Control.
8. Click the **Instances** tab at the top.

Verify that the instances were created. All should appear in the **Instance** list. The **Last Modified** list displays the times at which they were created.
9. Click the instance.
10. Check the contents of the directory (*drive_letter*:\temp on Windows operating systems or /temp on Unix operation systems). An output file named *po_timestamp.xml* should be created. Verify the attributes (for example the concatenation of the first and last names, and so on).

Summary

You have now added write functionality to the file adapter to accept input data in the form of a purchase order. This input data was transformed into a purchase order acknowledgment and the results were written to a file. You performed the following key tasks:

- Returned to the **POAcknowledge** BPEL process and created a partner link (named **FileOutInbound**) that used the file adapter to write files to a directory (in this case, an outgoing purchase order acknowledgment).
- Designed **POAcknowledge** to invoke **FileOutInbound**.
- Deployed **POAcknowledge**.
- Returned to the **OrderBooking** BPEL process and created a partner link (named **WritePOA**) that interacted with **POAcknowledge**.
- Designed **OrderBooking** to invoke **WritePOA**.
- Deployed the **OrderBooking** BPEL process.
- Ran **OrderBooking** from Oracle BPEL Control by completing and submitting an order form to the Rapid Distributors and Select Manufacturing services.
- Copied the purchase order file to be read to a directory.
- Accessed the Rapid Distributors URL to manually submit a supplier price.
- Observed that the purchase order file was deleted and a purchase order acknowledgment file was written to the directory. Observed that the contents of the incoming purchase order file were transformed into an outgoing purchase order acknowledgment.

Designing the Database Adapter to Insert Data

This chapter of the tutorial describes how to use a database adapter to commit data to database tables in a one-to-many (1:M) master-detail relationship.

This chapter contains the following topics:

- [Introduction](#)
- [Designing the Database Adapter with a 1:M Relationship to Insert Order and Order Item Data into Database Tables](#)
- [Summary](#)

Introduction

This phase of the tutorial adds to the process you designed in [Chapter 9, "Using the File Adapter's Write Functionality"](#). Ensure that you have successfully completed that phase before performing this one. In this phase of the tutorial, you use a database adapter to commit data to database tables in a one-to-many (1:M) master-detail relationship. You perform the following key tasks:

- Create a new BPEL process named **FulfillOrder**.
- Configure database connections and create sample database tables.
- Create a partner link that uses the database adapter (named **DBInsert**) to commit data to database tables in a 1:M relationship.
- Design the BPEL process to invoke **DBInsert**.
- Design the BPEL process to add transformation logic to transform data required by the database adapter.
- Return to the **OrderBooking** BPEL process and create a partner link (named **FulfillOrder**) that interacts with the **FulfillOrder** BPEL process.
- Design the BPEL process to invoke **FulfillOrder**.
- Add fault handling and exception management.
- Deploy the **OrderBooking** BPEL process.
- Run **OrderBooking** from Oracle BPEL Control to receive and send data to the database.

Designing the Database Adapter with a 1:M Relationship to Insert Order and Order Item Data into Database Tables

This section contains these tasks:

- [Creating the Order Fulfillment Process and Importing the Schema](#)
- [Creating a Database Connection and Samples Tables](#)
- [Creating a Partner Link that Uses the Database Adapter](#)
- [Creating an Invoke Activity](#)
- [Creating a Transform Activity and the Database Record Transformation](#)
- [Testing the Transformation Logic](#)
- [Validating, Compiling, and Deploying the FulFillOrder Process](#)
- [Running the FulFillOrder Process](#)
- [Verifying the Process](#)
- [Extending the Order Booking Process](#)

Creating the Order Fulfillment Process and Importing the Schema

From Oracle JDeveloper, follow these instructions to create the Order Fulfillment process and import the schema.

Caution: Do not include any special characters in the project name (such as periods) or in any activity or element names. If you do include special characters, errors appear when you attempt to compile your project.

1. Right-click **OrderBookapplication** and select **New Project**.
2. Under **Categories**, expand **General** and click **Projects**.
3. Under **Items**, click **BPEL Process Project** and click **OK**.
4. In the BPEL Project Creation Wizard - Project Settings window, enter the following details and click **Next**.

| Field | Value |
|------------------------------|--|
| Name | FulfillOrder |
| Namespace | Accept the default path. |
| Use Default Project Settings | Leave this checked to use the same name for the BPEL process and the BPEL project, and to use the default project directory. |
| Template | Asynchronous BPEL Process |

The **bpel.xml**, **FulfillOrder.bpel**, and **FulfillOrder.wsdl** files are created.

5. Click the **flashlight** icon to the right of the **Input Schema Element** field.
6. In the Select Schema window, select **OrderBookingPO.xsd** from the **SOA_Oracle_**

Home\bpel\samples\tutorials\127.OrderBookingTutorial\Practice Files directory and click **Open**.

The Type Chooser window appears.

7. Expand and select **Imported Schemas > OrderBookingPO.xsd > Purchase Order**.
8. Click **OK**.

The **OrderBookingRequestMessage** message type is automatically assigned an input message part name of **payload** and message part element of **PurchaseOrder**.

9. Click the **flashlight** icon to the right of the **Output Schema Element** field.

The Type Chooser window appears.

10. Expand and select **Imported Schemas > OrderBookingPO.xsd > Purchase Order**.
11. Click **OK**.

The **OrderBookingResponseMessage** message type is automatically assigned an output message part name of **payload** and message part element of **PurchaseOrder**.

12. Click **Finish**.

The GlobalCompany schema is now imported into your project. This schema defines the structure of the purchase order to submit.

The following files appear in the **Application Navigator** section of Oracle JDeveloper:

- The **bpel.xml**, **FulfillOrder.bpel**, and **FulfillOrder.wsdl** files appear under **FulfillOrder > Integration Content**.
 - The **OrderBookingPO.xsd** file appears under **FulfillOrder > Integration Content > Schemas**.
13. Double-click **OrderBooking.bpel** to display the **Structure** section in the lower left of the window.

Note: The **Diagram** tab must be enabled to see the **Structure** section.

14. Select **Save** from the **File** main menu.

Creating a Database Connection and Samples Tables

Summary: You now configure Oracle JDeveloper and Oracle BPEL Server to connect to the database instance installed with Oracle BPEL Process Manager.

You must create a database connection and database sample tables to use this portion of the tutorial.

This section contains these tasks:

- [Creating a JDeveloper Database Connection](#)
- [Creating Sample Tables](#)
- [Testing the Database Connection](#)

Creating a JDeveloper Database Connection

1. Select **Connection Navigator** from the **View** main menu.
2. Right-click **Database** in the **Connections - Navigator** window and select **New Database Connection**.
This starts the Create Database Connection Wizard.
3. Click **Next** on the Welcome window.
4. Enter **myConnection** in the **Connection Name** field of the Type window.
5. Select **Third Party JDBC Driver** from the **Connection Type** list, and click **Next**.
6. Enter **system** in the **Username** field of the Authentication window.
7. Enter **manager** in the **Password** field.
8. Leave the remaining fields as they are, and click **Next**.
9. Click **New** to the right of the **Driver Class** field on the Connection window.
The Register JDBC Driver window appears.
10. Enter **oracle.lite.poljdbc.POLJDBCdriver** in the **Driver Class** field. Ensure that there are no blank spaces at the end of this entry.
11. Click **Browse** to the right of the **Library** field.
The Select Library window appears.
12. Click **New**.
The Create Library window appears.
13. Click **Add Entry**.
The Select Path Entry window appears.
14. Navigate to **SOA_Oracle_Home/bpel/lib/olite40.jar**.
15. Click **Select**.
16. Click **OK** to close the Select Library and Register JDBC Driver windows.
17. Enter **jdbc:polite4@localhost:1531:orabpel** in the **URL** field of the Connection window.
18. Click **Next**.
19. Click **Test Connection** on the Test window.
If the connection was successful, the following message appears:
Success!
20. Click **Finish**.

Creating Sample Tables

1. Ensure that Oracle BPEL Server is running.
2. Select **Start > All Programs > Oracle - Oracle_Home > Oracle BPEL Process Manager > Developer Prompt** to open up an operating system command prompt.

Note: *Always* use the **Developer Prompt** to open an operating system command prompt in all phases of the Order Booking tutorial. Opening an operating system command prompt in any other way to run tutorial commands is not supported.

3. Change directories to the `tutorials\127.OrderBookingTutorial` directory:

```
cd tutorials\127.OrderBookingTutorial
```

4. Enter the following command:

```
polsql.cmd @PracticeFiles\insertTable.sql
```

Messages appear indicating table creation and sample row insertion into the default Oracle Lite instance.

Testing the Database Connection

1. Select **Connection Navigator** from the **View** main menu.
2. Expand **Database**.
3. Right-click **myConnection** and select **SQL Worksheet**.
4. Enter `SELECT * FROM system.orders;` in the **Enter SQL Statement** field.
5. Right-click the statement and select **Execute SQL Statement**.

The sample table created with data appears in the **Results** field.

The screenshot shows the 'Enter SQL Statement' field containing the query `select * from system.orders;`. Below it, the 'Results' field displays a table with the following data:

| | ID | COMPANY | ORDERDATE | ORDERPRICE | SUPPLIERNA... | SUPPLIERPR... | ORDERSTATUS |
|---|-----|------------|----------------|------------|---------------|---------------|-------------|
| 1 | 200 | 200Company | 1999-12-31 ... | 200 | 200Supplier | 200 | reject |
| 2 | 100 | 100Company | 1999-12-31 ... | 100 | 100Supplier | 100 | reject |
| 3 | 300 | 300Company | 1999-12-31 ... | 300 | 2300Supplier | 300 | reject |

Creating a Partner Link that Uses the Database Adapter

Summary: You now create a partner link that uses the database adapter to commit data into database tables in a one-to-many (1:M) relationship.

1. Select **Application Navigator** from the **View** main menu.
2. Double-click `FulFillOrder.bpel`.
3. Drag and drop a **PartnerLink** activity onto the right side of the designer window.
4. Enter **DBInsert** in the **Name** field.
5. Click the third icon at the top (the **Define Adapter Service** icon). This starts the Adapter Configuration Wizard.



6. Click **Next** on the Welcome window.
7. Select **Database Adapter** on the Adapter Type window and click **Next**.
8. Enter **writeDB** in the **Service Name** field on the Service Name window and click **Next**.
9. Select the **myConnection** database connection setting you created in "[Creating a Database Connection and Samples Tables](#)" on page 10-3 from the list.
10. Ensure that **eis/DB/BPELSamples** displays in the **Database Server JNDI Name** field. The name is case sensitive. Ensure that it correctly matches the case of the connection name.
11. Click **Next**.
12. Select **Perform an Operation on a Table**.
13. Unselect **Delete**.
14. Click **Next**.
15. Click **Import Tables**.
16. Click **Query**.
17. Select **ORDERS** and **ORDERITEMS**.
18. Click **>** to move both tables to the **Selected** field.
19. Click **OK**. You are returned to the Select Table window of the wizard.
20. Select **SYSTEM.ORDERS** and click **Next**.

The relationship definitions are automatically created and displayed as 1:M.

21. Select **ORDERITEMS (orders) has a 1:1 Relationship with ORDERS** and click **Remove Relationship**. This ensures that only **ORDERS(orderitemsCollection) has a 1:M relationship with ORDERITEMS** appears.
22. Click **Next**.
23. Leave all attributes as default for the attributes to return from the query on the page.
24. Click **Next** twice.
25. Click **Finish**.

The Partner Links window is automatically completed. The window looks as follows:

| Field | Value |
|-----------|---|
| Name | DBInsert |
| WSDL File | file:/c:/oraJDev/jdev/mywork/OrderBookapplication/FulFillOrder/writeDB.wsdl |

Note: This directory path with a drive letter represents an example on Windows operating systems. If running this tutorial on Unix operating systems, your directory path varies. In addition, **oraJDev** represents the Oracle JDeveloper home directory for this example.

| Field | Value |
|-------------------|--------------------|
| Partner Link Type | writeDB_plt |
| Partner Role | writeDB_role |
| My Role | Leave unspecified. |

26. Click **OK**.

27. Select **Save** from the **File** main menu.

Note that **writeDB.wsdl** now appears under **FulFillOrder > Integration Content > writeDB**. This file contains the parameters you specified with the Adapter Configuration Wizard.

Creating an Invoke Activity

1. Drag and drop an **Invoke** activity below the **receiveInput Receive** activity.
2. Double-click the **Invoke** activity to display the Invoke window.
3. Enter the following details:

| Field | Value |
|--------------|---------------|
| Name | writeDBrecord |
| Partner Link | DBInsert |

The **Operation (merge)** field is automatically filled in.

4. Click the first icon to the right of the **Input Variable** field. This is the automatic variable creation icon.
5. Click **OK** on the Create Variable window that appears.

A variable named **writeDBrecord_merge_InputVariable** is automatically created in the **Input Variable** field. This variable is automatically assigned a message type of **OrdersCollection_msg**.

6. Click **OK**.
7. Select **Save** from the **File** main menu.

Creating a Transform Activity and the Database Record Transformation

Summary: You now define your transformation logic.

1. Drag and drop a **Transform** activity below the **receiveInput Receive** activity.
2. Double-click the **Transform** activity to display the Transform window.
3. Click the **General** tab.
4. Enter **transformDBRecord** in the **Name** field.
5. Click the **Transformation** tab.
6. Select **inputVariable** in the **Source Variable** field and **payload** in the **Source Part** field.

7. Select **wroteDBrecord_merge_InputVariable** in the **Target Variable** field and **OrdersCollection** in the **Target Part** field.
8. Leave the **Mapper File** field selection as **Transformation_1**.
9. Click the second icon (the **Create Mapping** icon) to the right of the **Mapper File** field.

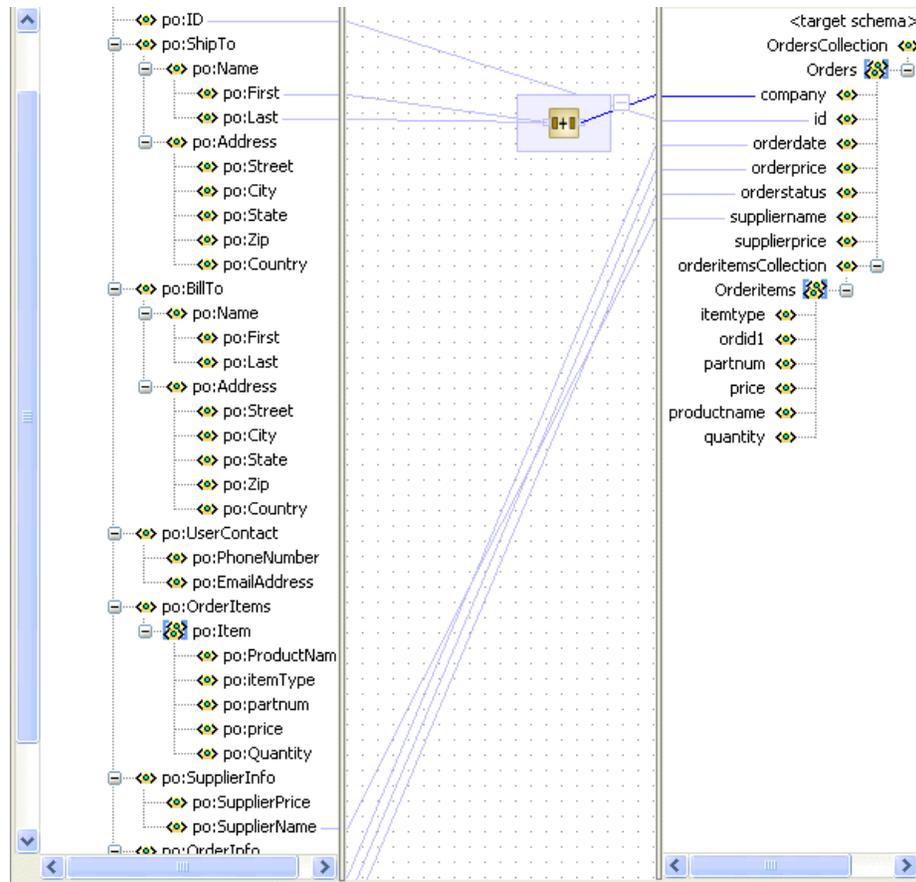


A transformation window for graphically mapping source and target elements appears. This window enables you to drag (map) a source element to a target element. If you instead receive a message saying the file already exists, click the third icon to access the transformation window.

10. Right-click the source and target schemas (the top-most elements in both lists) and select **Expand All**.
11. Drag and drop the following source elements to the following target elements:

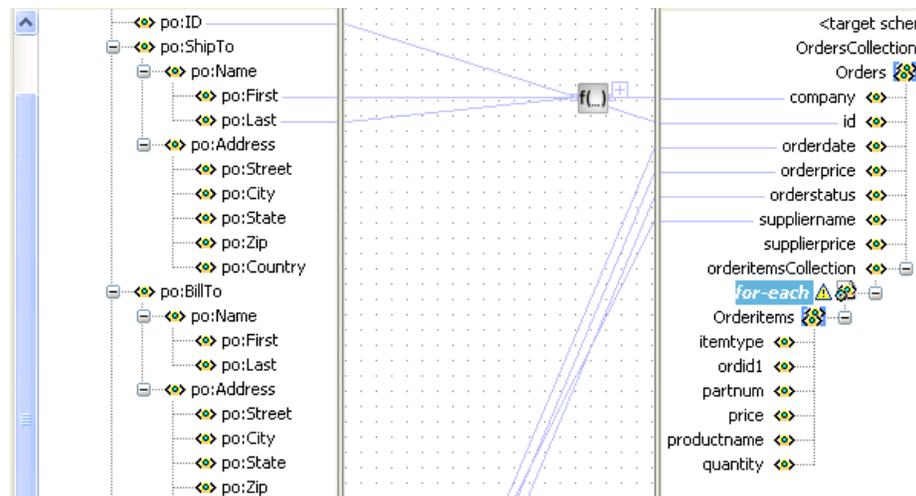
| Source | Target |
|------------------|---------------|
| po:ID | id |
| po:SupplierName | suppliername |
| po:SupplierPrice | supplierprice |
| po:OrderDate | orderdate |
| po:OrderPrice | orderprice |
| po:OrderStatus | orderstatus |

12. Select **String Functions** from the **Component Palette** section.
13. Drag and drop a **concat** function into the middle of the transformation window.
14. Go to the **po:ShipTo** source section.
15. Go to the **po:Name** subelement of this section.
16. Drag the **po:First** and **po>Last** source subelements into the **concat** function. This connects two source lines to the **concat** function.
17. Go to the **Orders** target section.
18. Drag the **company** target subelement into the **concat** function. This connects the target line to the **concat** function and concatenates the first and last names from the source elements into a single name in the target element.



19. Right-click the target **Orderitems** and select **Add XSL Node > for-each**.

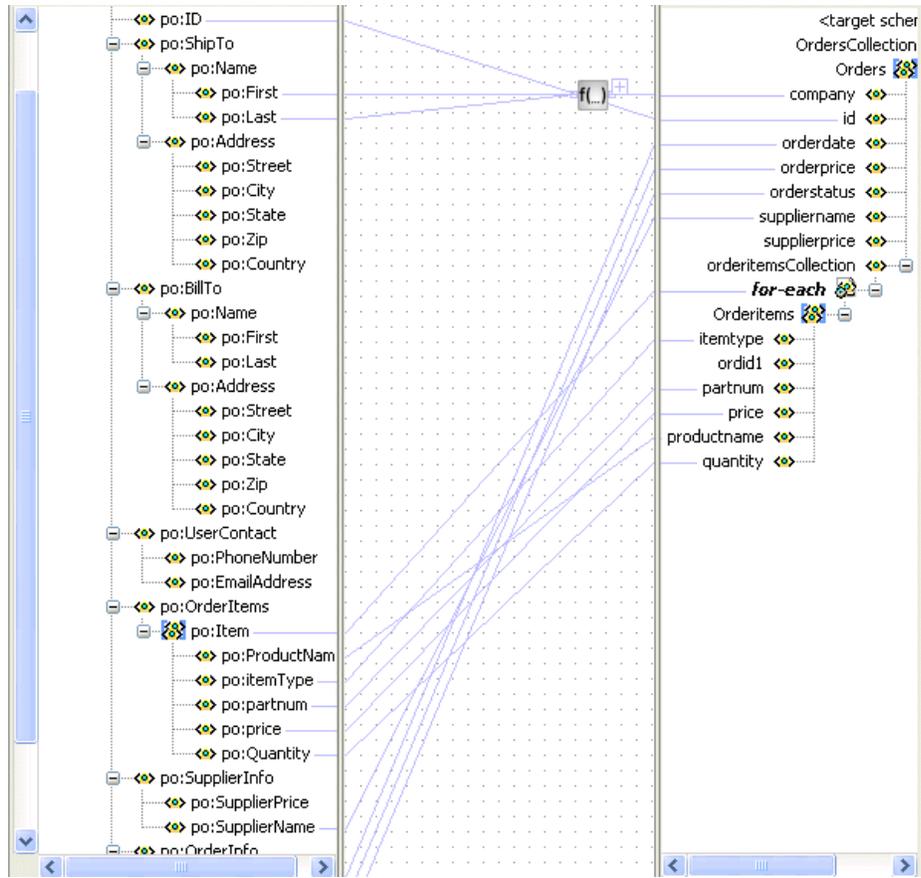
The **for-each** node appears between **orderitemsCollection** and **OrderItems** in the target schema.



20. Drag and drop the source **po:OrderItems:Item** onto the **for-each** node. This creates a 1:M relationship.

21. Drag and drop the source **po:OrderItems:Item** onto the target **Orderitems**.

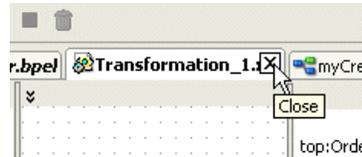
22. Click **OK** when prompted to confirm your mapping. This creates an automatic mapping of all matching subelements.



23. Select **Save** from the **File** main menu.

Testing the Transformation Logic

1. Right-click in the transformation mapping window and select **Test**.
This displays the Test XSL Map window.
2. Ensure that **Generate Source XML Data** is selected.
This creates sample XML test data.
3. Accept all other default settings, and click **OK**.
Sample test results appear. The source XML and target XML display data side-by-side, with the XSL map data underneath. This is the default layout of test results.
4. Verify the results.
5. Select **Save** from the **File** main menu.
6. Click **Yes** if prompted to save your transformation details.
7. Close the **Transformation_1-Source.xml**, **Transformation_1-Target.xml**, and **Transformation_1.xsl** files by clicking the x button for each in the **Title** tab.



A translation file named **Transformation_1.xsl** appears under **POAcknowledge > Integration Content** in the **Application Navigator** section.

Validating, Compiling, and Deploying the FulFillOrder Process

1. Go to the **Application Navigator** section.
2. Right-click **FulfillOrder**.
3. Select **Deploy > my_integration_server_connection > Deploy to default domain**.

This compiles the BPEL process. Check for errors by clicking the buttons at the bottom of the window. If there are no errors, deployment was successful. If deployment was unsuccessful, see Step 4 on page 2-17.

Running the FulFillOrder Process

1. Log into Oracle BPEL Control by selecting **Start > All Programs > Oracle - Oracle_Home > Oracle BPEL Process Manager > BPEL Control** (or refresh the page if it is already open).
2. Enter the following details to log into Oracle BPEL Control and click **Login**:

| Field | Value |
|----------|-----------|
| Username | oc4jadmin |
| Password | password |

Where *password* is the value you entered during installation.

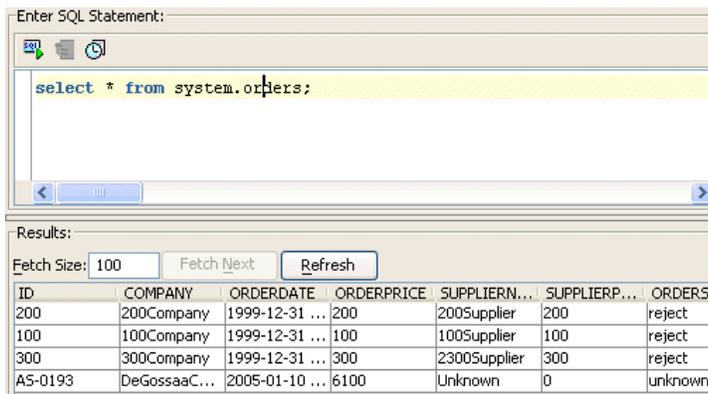
The **Dashboard** tab of Oracle BPEL Control appears. Note that your BPEL process, **FulfillOrder**, now appears in the Deployed BPEL Processes list. Refresh if necessary.

3. Click **FulfillOrder**.
4. Enter information in *one of the following ways*:
 - Enter details in all the fields of the HTML Form and click **Post XML Message**.
 - Click the **XML Source** radio button. Use a text editor to copy and paste the contents of `OrderBookingPO_1.xml` into the field that appears. Click **Post XML Message**.
5. Refresh Oracle BPEL Control.
6. Click the **Instances** tab at the top.
7. Click the **FulfillOrder** instance.
8. Click the **Audit** link to view the transformation logic and the output written into the database tables.
9. Click the **More** link for each sequence to view the activity details. (Similar instance management can also be obtained in a graphical fashion under the **Flow** link.)

Verifying the Process

1. Return to Oracle JDeveloper.
2. Select **Connection Navigator** from the **View** main menu.
3. Expand **Database**.
4. Right-click **myConnection** and select **SQL Worksheet**. Note that right clicking on the connection name fails if the case of the connection name is incorrect.
5. Right-click the SQL statement you entered in Step 4 on page 10-5 and select **Execute SQL Statement** (for example, **SELECT * FROM system.orders;**) or manually enter **SELECT * FROM orderitems WHERE ordid = 'your_order_number'**.

This verifies that the database table is properly inserted.



Extending the Order Booking Process

In this phase of the tutorial, you extend the main **OrderBooking** process to call the newly created **FulFillOrder** process as a web service for inserting database tables (through the defined WSDL file).

This section contains these tasks:

- [Creating an Order FulFillment Partner Link](#)
- [Creating a Scope Activity](#)
- [Creating Invoke and Receive Activities](#)
- [Creating an Initial Assign Activity](#)
- [Creating a Second Assign Activity](#)
- [Adding Database Exception Handling](#)
- [Validating, Compiling, and Deploying the OrderBooking Process](#)
- [Running the Order Booking Process](#)

Creating an Order FulFillment Partner Link

Summary: You now create a partner link for the FulfillOrder service.

1. Go to the **OrderBooking** process in the **Application Navigator** section.
2. Double-click **OrderBooking.bpel**.

3. Select **Services** from the **Component Palette** section.
4. Drag and drop a **PartnerLink** activity from the **Component Palette** section onto the right side of the designer window.
5. Enter the following values to create a partner link for the Order Booking process:

Note: For the **WSDL File** field below, click the **flashlight** (the second icon from the left named **Service Explorer**) to access the Service Explorer window for automatically selecting the **FulFillOrder** process as a service.

| Field | Value |
|-------------------|--|
| Name | FulfillOrder |
| WSDL File | Access this URL by clicking the Service Explorer flashlight icon and expanding and selecting BPEL Services > my_integration_server_connection > processes > default > FulfillOrder http://localhost:8888/orabpel/default/FulfillOrder/FulfillOrder?wsdl See Also: " Setting the Hostname in Your Oracle JDeveloper Web Browser Preferences " on page 1-7 if you receive a parsing error when attempting to add a WSDL file in the Service Explorer window. |
| Partner Link Type | FulfillOrder |
| Partner Role | FulfillOrderProvider |
| My Role | FulfillOrderRequester |

6. Click **OK**.
7. Select **Save** from the **File** main menu.

Creating a Scope Activity

Summary: You now create a **Scope** activity to group all activities that form a logical step to be executed.

1. Select **Process Activities** from the **Component Palette** section.
2. Drag and drop a **Scope** activity from the **Component Palette** section to below the **SelectSupplier Switch** activity.
3. Double-click the **scope** icon to display the Scope window.
4. Enter **OrderFulFillment** in the **Name** field of the **General** tab.
5. Click **OK**.

In this phase of the tutorial, you create the **Invoke**, **Receive**, and **Assign** activities inside this **Scope** activity.

6. Click the + sign to expand the **Scope** activity.
7. Select **Save** from the **File** main menu.

Creating Invoke and Receive Activities

Summary: You now create **Invoke** and **Receive** activities.

1. Drag and drop an **Invoke** activity from the **Component Palette** section into the **OrderFulFillment Scope** activity.
2. Double-click the **Invoke** icon to display the Invoke window.
3. Enter the following details:

| Field | Value |
|--------------|--------------------|
| Name | invokeFulfillOrder |
| Partner Link | FulfillOrder |

The **Operation (initiate)** field is automatically filled in.

4. Click the first icon to the right of the **Input Variable** field. This is the automatic variable creation icon.
5. Click **OK** on the Create Variable window that appears.
A variable named **invokeFulfillOrder_initiate_InputVariable** is automatically created in the **Input Variable** field. This variable is automatically assigned a message type of **FulfillOrderRequestMessage**.
6. Click **OK**.
7. Drag and drop a **Receive** activity from the **Component Palette** section to below the **invokeFulfillOrder Invoke** activity inside the **OrderFulFillment Scope** activity.
8. Double-click the **Receive** icon to display the Receive window.
9. Enter the following details:

| Field | Value |
|--------------|---------------------|
| Name | receiveFulfillOrder |
| Partner Link | FulfillOrder |

The **Operation (onResult)** field is automatically filled in.

10. Click the first icon to the right of the **Variable** field. This is the automatic variable creation icon.
11. Click **OK** on the Create Variable window that appears.
A variable named **receiveFulfillOrder_onResult_InputVariable** is automatically created in the **Output Variable** field. This variable is automatically assigned a message type of **FulfillOrderResponseMessage**.
12. Click **OK**.
13. Select **Save** from the **File** main menu.

Creating an Initial Assign Activity

1. Drag and drop an **Assign** activity from the **Component Palette** section to above the **invokeFulfillOrder Invoke** activity inside the **OrderFulFillment Scope** activity.
2. Double-click the **assign** icon to display the Assign window.
3. Click the **General** tab.

4. Enter **assignDBIn** in the **Name** field.
5. Click **Apply**.
6. Click the **Copy Operation** tab.
7. Click **Create** and select **Copy Operation** to display the Create Copy Operation window.
8. Enter the following values:

| Field | Value |
|--------------------|---|
| From | |
| ▪ Type | Variable |
| ▪ Variables | Expand and select Variables > inputVariable > payload |
| To | |
| ▪ Type | Variable |
| ▪ Variables | Expand and select Variables > invokeFulfillOrder_initiate_ InputVariable > payload |

9. Click **OK** to close the Create Copy Operation window and the Assign window.
10. Select **Save** from the **File** main menu.

Creating a Second Assign Activity

1. Drag and drop an **Assign** activity from the **Component Palette** section to below the **receiveFulfillOrder Receive** activity inside the **OrderFulfillment Scope** activity.
2. Double-click the **assign** icon to display the Assign window.
3. Click the **General** tab.
4. Enter **assignDBWriteComments** in the **Name** field.
5. Click **Apply**.
6. Click the **Copy Operation** tab.
7. Click **Create** and select **Copy Operation** to display the Create Copy Operation window.
8. Enter the following values:

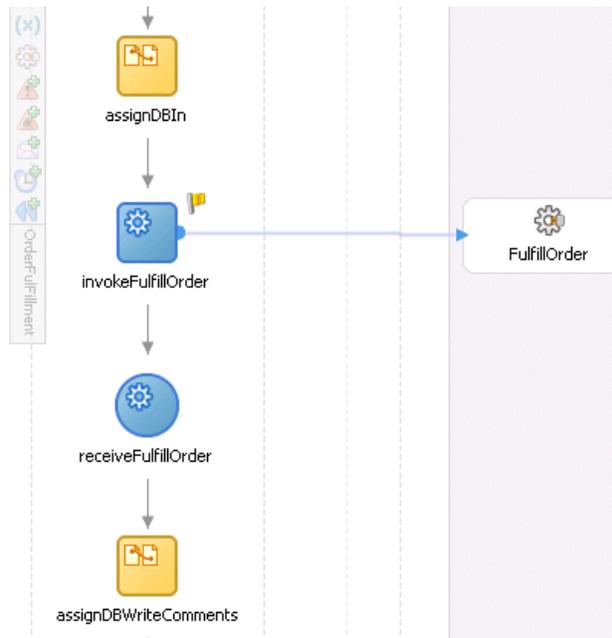
Note: Instead of manually entering an expression, you can press **Ctrl** and then the space bar in the **Expression** field. Scroll through the list of values that appears and double-click the value you want. Edit the value as necessary. As you enter information, a trailing slash can appear. This means you are being prompted for additional information. Either enter additional information, or press the **Esc** key and delete the trailing slash to complete the input of information.

| Field | Value |
|---------------|-------------------|
| From | |
| ▪ Type | Expression |

| Field | Value |
|--|--|
| <ul style="list-style-type: none"> Expression | concat(bpws:getVariableData('inputVariable','payload','/ns1:PurchaseOrder/ns1:OrderInfo/ns1:OrderComments'),' - Order Fulfillment Complete') |
| To | |
| <ul style="list-style-type: none"> Type | Variable |
| <ul style="list-style-type: none"> Variables | Expand and select > Variables > inputVariable > payload > ns1:PurchaseOrder > ns1:OrderInfo > ns1:OrderComments |

- Click **OK** to close the Create Copy Operation window and the Assign window.

When complete, the designer window looks as follows:



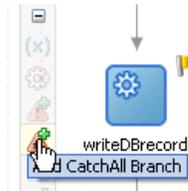
- Click the - sign to close the **OrderFulfillment Scope** activity.
- Select **Save** from the **File** main menu.

Adding Database Exception Handling

Summary: You perform database exception handling tasks in the **FulFillOrder** process by creating a local variable and placing the **writeDBRecord Invoke** activity in a **Scope** activity. The local variable is set to false (represented by 0). You attempt to call the external partner service in a **While** loop activity until the local variable is satisfied (set to 1). The **While** activity is set to loop a maximum of five times. In the case of an exception, you reset the flag to false (0).

- Ensure that you are in the **FulFillOrder** process. If not, double-click **FulFillOrder.bpel** in the **Application Navigator** section of the designer window.
- Drag and drop a **Scope** activity from the **Component Palette** section to below the **transformDBRecord Transform** activity.
- Double-click the **Scope** icon to display the Scope window.

4. Enter **db_write_scope** in the **Name** field of the **General** tab.
5. Click **OK**.
6. Click the + sign to expand the **Scope** activity.
7. Drag and drop the **writeDBrecord Invoke** activity into the **db_write_scope** Scope activity.
8. Click **Add Catch All Branch** in the icons on the left side of **db_write_scope**.



9. Click the + sign to expand the new branch.
A new box displays to the right with an asterisk and the words **Drop Activity Here**.
10. Select **Variables > Process > Variables** in the **Structure** section.
11. Right-click **Variables** and select **Create Variable**.
12. Enter **dbStatus** in the **Name** field.
13. Select **Simple Type** and click the **flashlight** icon to the right of the **Simple Type** field.
14. Select **int**.
15. Click **OK** to close the Type Chooser window and the Create Variable window.
16. Drag and drop an **Assign** activity from the **Component Palette** section to above the **db_write_scope** Scope activity.
17. Double-click the **Assign** icon to display the Assign window.
18. Click the **General** tab.
19. Enter **assign_db_0** in the **Name** field.
20. Click **Apply**.
21. Click the **Copy Operation** tab.
22. Click **Create** and select **Copy Operation** to display the Create Copy Operation window.
23. Enter the following values:

| Field | Value |
|--------------|--|
| From | |
| ■ Type | Expression |
| ■ Expression | '0' |
| To | |
| ■ Type | Variable |
| ■ Variables | Expand and select Variables > dbStatus |

24. Click **OK** to close the Create Copy Operation window and the Assign window.
25. Drag and drop another **Assign** activity from the **Component Palette** section to below the **writeDBrecord Invoke** activity inside the **db_write_scope Scope** activity.
26. Double-click the **Assign** icon to display the Assign window.
27. Click the **General** tab.
28. Enter **assign_db_1** in the **Name** field.
29. Click **Apply**.
30. Click the **Copy Operation** tab.
31. Click **Create** and select **Copy Operation** to display the Create Copy Operation window.
32. Enter the following values:

| Field | Value |
|--------------|--|
| From | |
| ▪ Type | Expression |
| ▪ Expression | '10' |
| To | |
| ▪ Type | Variable |
| ▪ Variables | Expand and select Variables > dbStatus |

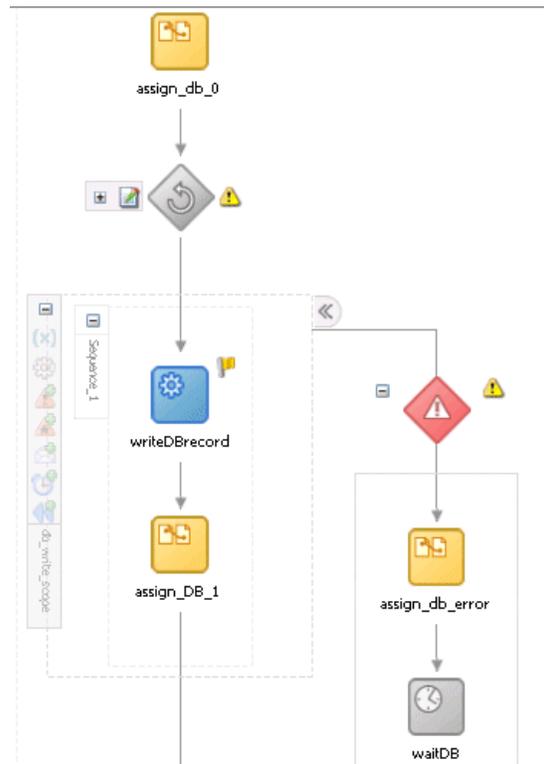
33. Click **OK** to close the Create Copy Operation window and the Assign window.
34. Drag and drop a third **Assign** activity from the **Component Palette** section into the **Catch All** branch on the right side of the **db_write_scope Scope** activity.
35. Double-click the **Assign** icon to display the Assign window.
36. Enter **assign_db_error** in the **Name** field of the **General** tab.
37. Click **Apply**.
38. Click the **Copy Operation** tab.
39. Click **Create** and select **Copy Operation** to display the Create Copy Operation window.
40. Enter the following values:

| Field | Value |
|--------------|--|
| From | |
| ▪ Type | Expression |
| ▪ Expression | bpws:getVariableData('dbStatus') + 1 |
| To | |
| ▪ Type | Variable |
| ▪ Variables | Expand and select Variables > dbStatus |

41. Click **OK** to close the Create Copy Operation window and the Assign window.

42. Drag and drop a **Wait** activity from the **Component Palette** section into the **Catch** branch below the **assign_db_error** **Assign** activity.
43. Double-click the **Wait** activity to display the Wait window.
44. Enter **waitDB** in the **Name** field.
45. Set the value to **30** seconds in the **HH:mm:ss** field. This means this block is retried every 30 seconds until the **FulFillOrder** process is successful (that is, **dbStatus** is 1).
46. Click **OK**.
47. Drag and drop a **While** activity from the **Component Palette** section to immediately above the **db_write_scope** **Scope** activity.

The designer window looks as follows:



48. Click the - sign to close the **db_write_scope** **Scope** activity.
49. Expand the **While** activity.
50. Drag the **db_write_scope** **Scope** activity inside the **While** activity.
51. Double-click the **While** activity and press **Ctrl** and then the space bar to enter the following in the **Expression** field:


```
bpws:getVariableData('dbStatus') < 5
```
52. Click **OK**.
53. Select **Save** from the **File** main menu.

Validating, Compiling, and Deploying the OrderBooking Process

1. Go to the **Application Navigator** section.

2. Right-click **OrderBooking**.
3. Select **Deploy > my_integration_server_connection > Deploy to default domain**.
4. Increment the version number of the project when prompted (for example, enter 1.6) and click **OK**.

This compiles the BPEL process. Check for errors by clicking the buttons at the bottom of the window. If there are no errors, deployment was successful. If deployment was unsuccessful, see Step 4 on page 2-17.

Running the Order Booking Process

1. Log into Oracle BPEL Control by selecting **Start > All Programs > Oracle - Oracle_Home > Oracle BPEL Process Manager > BPEL Control** (or refresh the page if it is already open).
2. Enter the following details to log into Oracle BPEL Control and click **Login**:

| Field | Value |
|----------|-----------|
| Username | oc4jadmin |
| Password | password |

The **Dashboard** tab of Oracle BPEL Control appears. Note that your BPEL process, **OrderBooking**, now appears in the **Deployed BPEL Processes** list.

3. Click the **BPEL Processes** tab.
4. Note which **OrderBooking** version has an asterisk next to the version number. This is the default version.
5. Click the latest version number of the OrderBooking process. If you have been incrementing the version numbers as described in previous chapters, this is the highest number.

The Managing this BPEL Process window appears.

6. Scroll down and click **Mark as Default** to make this the default version of OrderBooking.
7. Click **Done** when prompted and note that this **OrderBooking** version now has an asterisk next to the version number, indicating that it is the default version.
8. Click the **Dashboard** tab.
9. Start the correct **OrderBooking** version instance in *one of the following ways*:
 - Click **OrderBooking** in the **Deployed BPEL Processes** list.
 - Enter details in all fields of the HTML Form and click **Post XML Message**.
 - or
 - Click the **XML Source** radio button. Use a text editor to copy and paste the contents of the OrderBookingPO_1.xml file into the field that appears. Click **Post XML Message**.

Note: Do *not* copy and paste from Internet Explorer or Mozilla Firefox. You must use a text editor to perform this task.

- *Optionally* copy the `OrderBookingPO_1.xml` file to the `temp` directory to invoke the process instance (this assumes you have completed [Chapter 8, "Using the File Adapter's Read Functionality"](#)).
- 10. Complete the manual price quote task for **SelectManufacturing** by using the graphical user interface JSP page, as described in Step 11 on page 4-10.
- 11. Return to Oracle BPEL Control.
- 12. Refresh Oracle BPEL Control.
- 13. Click the **Instances** tab at the top.
- 14. Click the **OrderBooking** instance.
- 15. Click the **Audit** link to view the transformation logic and the output file written in the temporary directory you specified (*drive_letter*: \temp on Windows operating systems or /temp on Unix operation systems).
- 16. Click the **More** link for each sequence to view the activity details. (Similar instance management can also be obtained in a graphical fashion under the **Flow** link.)
- 17. Check the contents of the `temp` directory. An output file named `po_timestamp.xml` should be created. Verify the attributes (for example the concatenation of the first and last names, and so on).
- 18. Select **Connection Navigator** from the **View** main menu in Oracle JDeveloper.
- 19. Expand **Database**.
- 20. Right-click **myConnection** and select **SQL Worksheet**.
- 21. Enter the statement **SELECT * FROM system.orderitems;** and right-click and select **Execute SQL Statement** to see your orders persisted in the database.

The screenshot shows the 'Enter SQL Statement:' window with the query `select * from system.orderitems;` entered. Below the window, the 'Results:' section displays a table with the following data:

| ORDID | ITEMTYPE | PRODUCTN... | QUANTITY | PRICE | PARTNUM |
|---------|------------|---------------|----------|-------|------------|
| 100 | 100item | 100Product... | 100 | 100 | 100partnum |
| 200 | 200item | 200Product... | 200 | 200 | 200partnum |
| 300 | 300item | 300Product... | 300 | 300 | 300partnum |
| 100 | 101item | 101Product... | 101 | 101 | 100partnum |
| 200 | 202item | 202Product... | 202 | 202 | 200partnum |
| 300 | 303item | 303Product... | 303 | 303 | 300partnum |
| AS-0193 | Entertainm | Sony Camco... | 5 | 500 | TRV-32H |
| AS-0193 | Audio-Wide | Grundi TV | 4 | 900 | 239-FRH |

Summary

You have now used the database adapter to commit data to database tables in a 1:M master-detail relationship. You performed the following key tasks:

- Created a new BPEL process named **FulfillOrder**.
- Configured database connections and created sample database tables.
- Created a partner link that used the database adapter (named **DBInsert**) to commit data to database tables in a 1:M relationship.

- Designed the BPEL process to invoke **DBInsert**.
- Designed the BPEL process to add transformation logic to transform data required by the database adapter.
- Returned to the **OrderBooking** BPEL process and created a partner link (named **FulfillOrder**) that interacts with the **FulfillOrder** BPEL process.
- Designed the BPEL process to invoke **FulfillOrder**.
- Added fault handling and exception management.
- Deployed the **OrderBooking** BPEL process.
- Ran **OrderBooking** from Oracle BPEL Control to receive and send data to the database

Designing the Human Task Workflow System

This chapter of the tutorial describes how to add user tasks to handle the manual approval or rejection of a purchase order.

This chapter contains the following topics:

- [Introduction](#)
- [Designing the Human Workflow System for Manual Approval of an Order](#)
- [Summary](#)

Introduction

This phase of the tutorial adds to the process you designed in [Chapter 10, "Designing the Database Adapter to Insert Data"](#). Ensure that you have successfully completed that phase before performing this one. This phase extends the previous project steps by calling an external manual service through human task capabilities. It includes a call out to another Web service that handles human tasks and uses predefined (default) graphical user interface screens for user actions. You perform the following key tasks:

- Create a new BPEL process named **OrderApproval**.
- Design the BPEL process to add workflow functionality to handle the manual approval or rejection of a purchase order.
- Deploy the **OrderApproval** BPEL process.
- Run the **OrderApproval** BPEL process from Oracle BPEL Control.
- Access the Oracle BPEL Worklist Application URL to approve the order.
- Return to the **OrderBooking** BPEL process and create a partner link (named **POAService**) that interacts with the **OrderApproval** BPEL process.
- Design the BPEL process to invoke **POAService**.
- Deploy the **OrderBooking** BPEL process.
- Run the **OrderBooking** BPEL process from Oracle BPEL Control.
- Access the Oracle BPEL Worklist Application URL to approve the order.

Designing the Human Workflow System for Manual Approval of an Order

This section contains these topics:

- [Modeling the Order Approval Process](#)

- [Creating a Partner Link for the Order Approval Service](#)
- [Creating a Scope Activity](#)
- [Creating Invoke and Receive Activities](#)
- [Creating an Initial Assign Activity](#)
- [Creating a Second Assign Activity](#)
- [Validating, Compiling, and Deploying the Order Process](#)
- [Running and Verifying the Order Process](#)

Modeling the Order Approval Process

In this phase of the tutorial, you create a new project, Order Approval, and define the human workflow process, a sequential workflow in which the order is approved or rejected. The order is first assigned to the Supervisor role. After a user with the Supervisor role approves the order, it is sent to the user's manager for final approval.

This section contains these tasks:

- [Creating the Order Approval Process and Importing the Schema](#)
- [Adding a Human Task to the Order Approval Process](#)
- [Assigning Input and Output Parameters for the Human Task](#)
- [Creating a Task Form for the Worklist](#)
- [Modeling the Task Outcome](#)
- [Validating, Compiling, and Deploying the Order Approval Process](#)
- [Running the Order Approval Process](#)

Creating the Order Approval Process and Importing the Schema

Summary: After creating the BPEL process, you import the `OrderBookingPO.xsd` schema.

1. In **OrderBookingapplication**, create an asynchronous BPEL process with the name **OrderApproval**.

See "[Creating the Order Booking Project and Importing the Schema](#)" on page 2-3 for a refresher on how to create a project.

2. Click **Next**.
3. Click the **flashlight** next to **Input Schema Element** to browse for **OrderBookingPO.xsd** in

```
SOA_Oracle_Home\bpel\samples\tutorials\127.OrderBookingTutorial\
OrderApproval\bpel
```

The Type Chooser window appears.

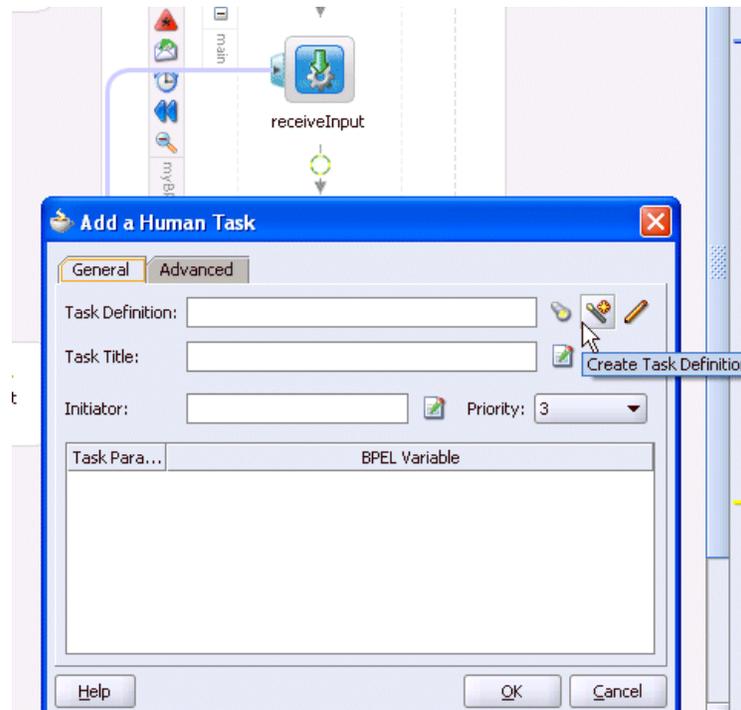
4. Expand and select **Imported Schemas > OrderBookingPO.xsd > PurchaseOrder**.
5. Click **OK**.
6. Click the **flashlight** next to **Output Schema Element** and repeat Step 4.
7. Click **Finish**.

The schemas are now imported into the project. **OrderBookingPO.xsd** appears under **Order Approval > Integration Content > Schemas** in the **Application Navigator**, and under **Schemas** in the **Structure** section. The BPEL process—a Receive activity (**receiveInput**) and an Invoke activity (**callbackClient**)—is displayed.

Adding a Human Task to the Order Approval Process

Summary: When you define the human task, the `OrderApproval.task` file—the task configuration metadata file—is created.

1. Select **Process Activities** from the **Component Palette** section.
2. Drag and drop a **Human Task** activity from the **Component Palette** to between **receiveInput** and **callbackClient**.
3. Click the **Create Task Definition** icon (second icon).



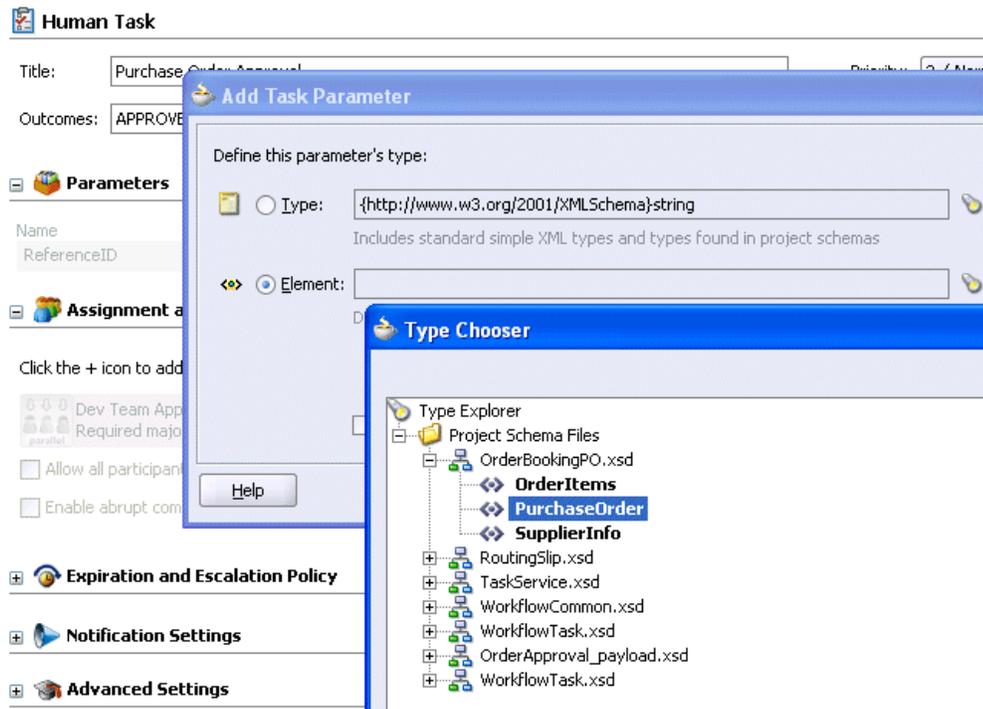
4. Enter **OrderApproval** for the human task name and click **OK**. (Accept the default location.)

The `OrderApproval.task` file is created.

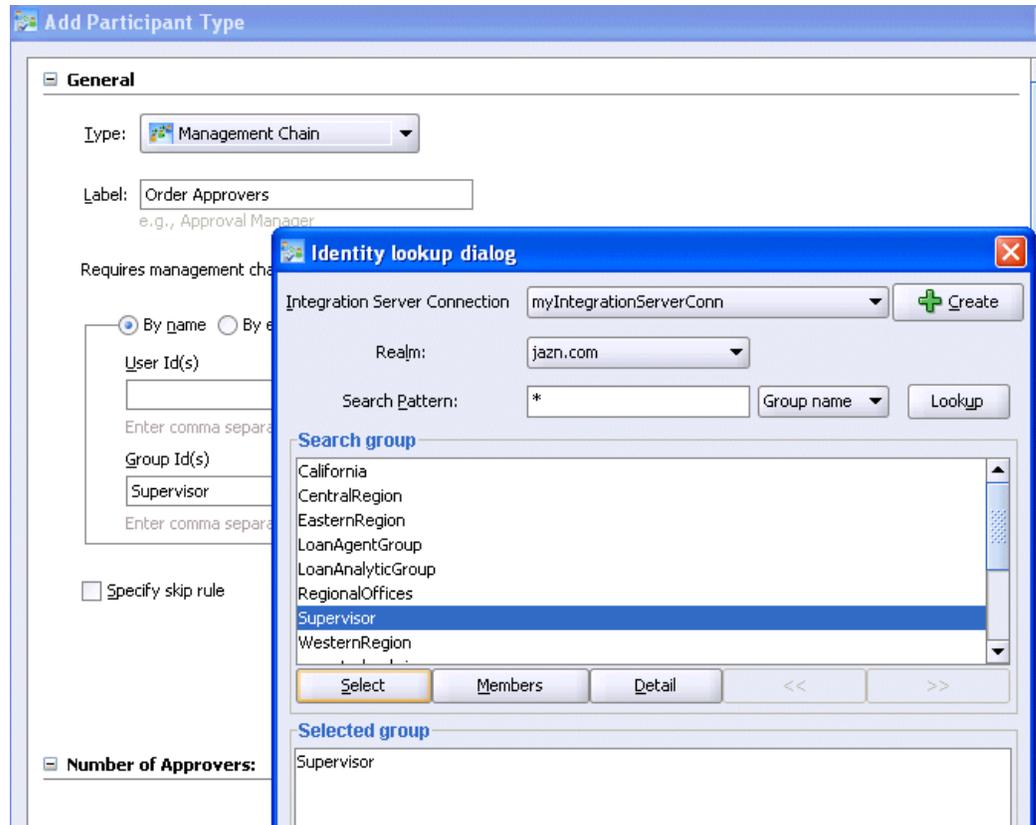
The Human Task editor is displayed.

5. For **Title**, enter **Purchase Order Approval**.
6. Accept the default values for **Priority** and **Outcomes**.
7. For **Parameters**, click the + icon on the right side of the window.
The Add Task Parameter window is displayed.
8. Click **Element** and then the **flashlight** icon.

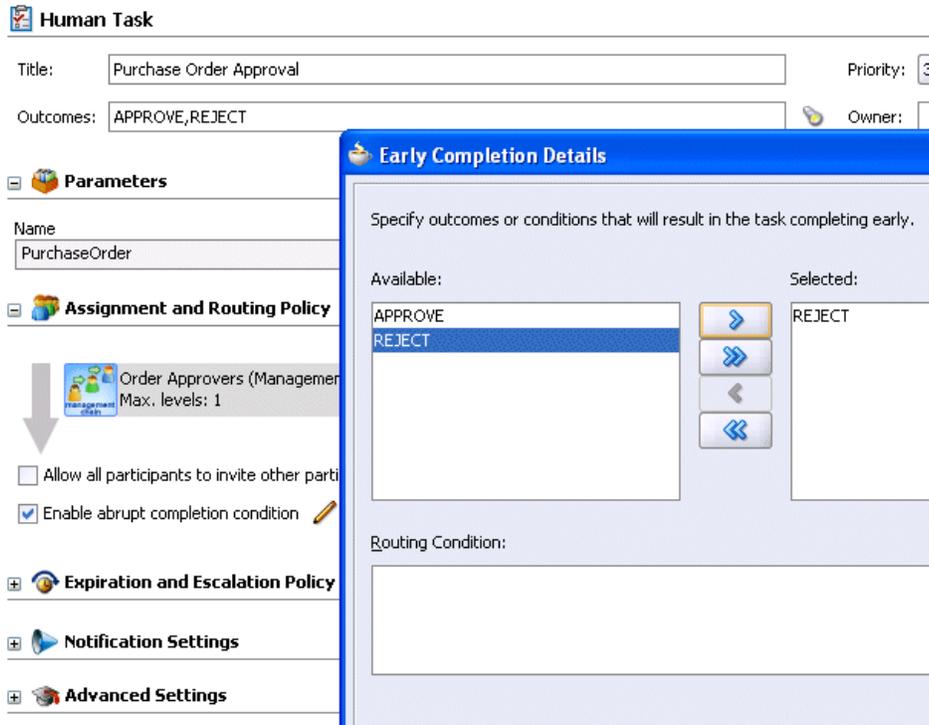
- In the Type Chooser window, expand and select **Project Schema Files > OrderBookingPO.xsd > PurchaseOrder**, and click **OK**.



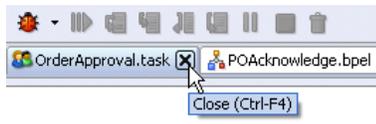
- In the Add a Parameter window, click **Modifiable via worklist** and click **OK**.
This ensures that you can modify task data using the Oracle BPEL Worklist Application.
- In the **Assignment and Routing Policy** section, click the + icon on the right side of the window.
The Add Participant Type window is displayed.
- For **Type**, select **Management Chain**.
This participant type routes tasks for approval to multiple users in a management chain hierarchy.
- For **Label**, enter **Order Approvers**.
- Click **By name**.
In this example, you assign the task to the Supervisor group and specify one level of the management chain above the user who acts on the task.
- For **Group Id(s)**, click the **flashlight** to display the Identity lookup dialog window.
Ensure that your integration server connection is selected.
- Ensure that Oracle BPEL Server is running and click **Lookup** to see a list of all groups currently in the system.
- Select **Supervisor**, click **Select**, and click **OK**.



18. In the **Number of Approvers** section of the Add Participant Type window, enter 1 in the **Maximum Number of Chain Levels Up** field. Click **OK**.
19. In the **Assignment and Routing Policy** section, click **Enable abrupt completion condition**.
The Early Completion Details window is automatically displayed.
20. Select **REJECT** and move it to the **Selected** list. Click **OK**.



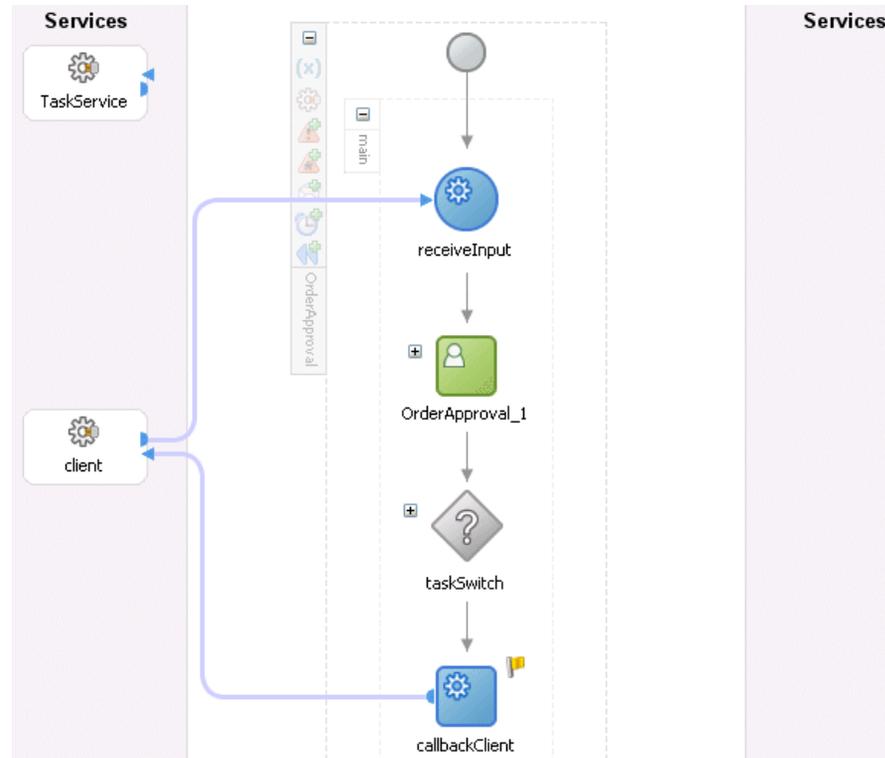
21. Select **Save** from the **File** main menu.
22. Click the **X** next to **OrderApproval.task** to close the Human Task editor.



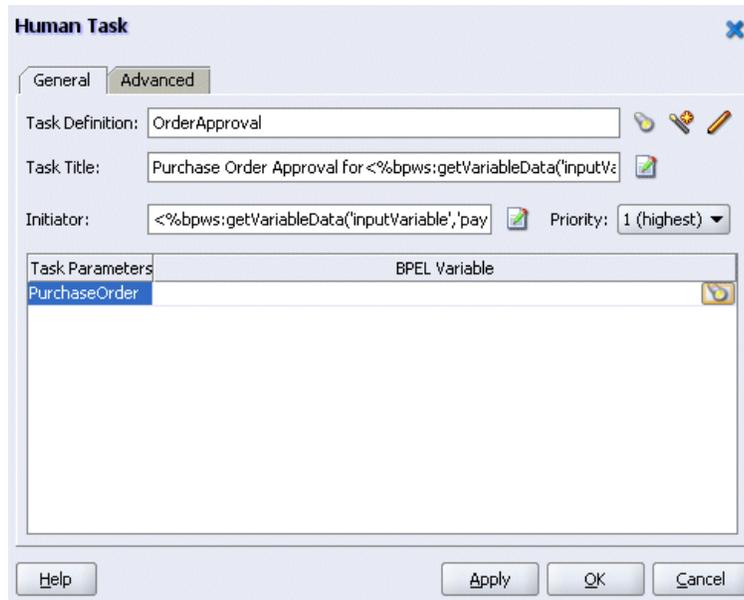
Assigning Input and Output Parameters for the Human Task

Summary: Map the fields to the variables in the BPEL process.

1. Double-click **OrderApproval.bpel** to see the Human Task activity, **OrderApproval_1**, in the BPEL process.



2. Double-click **OrderApproval_1**.
3. In the **Task Title** field, add the word *for* to the end of the expression *Purchase Order Approval*.
4. Click the icon at the right to use the XPath Expression Builder.
5. In the **BPEL Variables** section, expand and select **inputVariable** > **payload** > **ns1:PurchaseOrder** > **ns1:CustID**.
6. Click **Insert Into Expression**.
The XPath expression corresponding to **ns1:CustID** appears in the **Expression** section.
7. Click **OK**.
The XPath expression is appended to the task title.
8. Click the icon at the right to use the XPath Expression Builder for the **Initiator** field.
9. Repeat Steps 5 through 7 to insert the same XPath expression in the **Initiator** field.



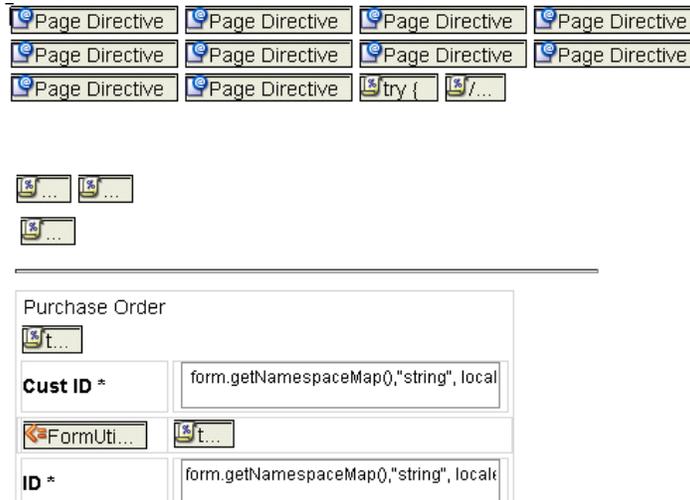
10. Click the **flashlight** icon in the **BPEL Variable** column.
The Task Parameters window appears.
11. In the Task Parameters window, expand and select **Variables > inputVariable > payload > ns1:PurchaseOrder**.
12. Click **Insert Into Expression**, and click **OK**.
13. In the Human Task window, click **OK**.
14. Select **Save** from the **File** main menu.

Creating a Task Form for the Worklist

Summary: An autogenerated task form, `payload-body.jsp`, is created.

1. In the **Application Navigator**, right-click the **OrderApproval** folder that includes the `.task` file and select **Auto Generate Simple Task Form**.

The autogenerated task form file appears:



2. Close `payload-body.jsp` by clicking the X sign at the top.

Modeling the Task Outcome

Summary: The Switch activity reflects the possible outcomes, or cases, specified previously, Approve and Reject. It also has an Otherwise case to represent other outcomes, such as errored, stale, or expired. Inside each of the cases, you can add activities to complete modeling of the business process. The `copyPayloadFromTask` Assign activities copy the payload back to its source.

1. Double-click `OrderApproval.bpel`.
2. Expand the `taskSwitch` Switch activity.
3. Drag and drop an **Assign** activity to below the `copyPayloadFromTask` Assign activity in the `<case Task outcome is APPROVE>` section of the **Switch** activity.
4. Double-click the **Assign** icon to display the Assign window.
5. Click the **General** tab.
6. Enter `assignOrderApproval1` in the **Name** field.
7. Click **Apply**.
8. Click the **Copy Operation** tab.
9. Click **Create** and select **Copy Operation**.
10. Enter the following details:

| Field | Value |
|--------------|---------------------------------|
| From | |
| ▪ Type | Expression |
| ▪ Expression | <code>string('Accepted')</code> |
| To | |
| ▪ Type | Variable |

| Field | Value |
|---|---|
| <ul style="list-style-type: none"> Variables | Select and expand Variables > inputVariable > payload > ns1:PurchaseOrder > ns1:OrderInfo > ns1:OrderComments Note: The namespace number values (for example, ns1 , ns2) can vary. Use the namespace values that automatically appear. |

Note that the **XPath** field shows that the expression is set to

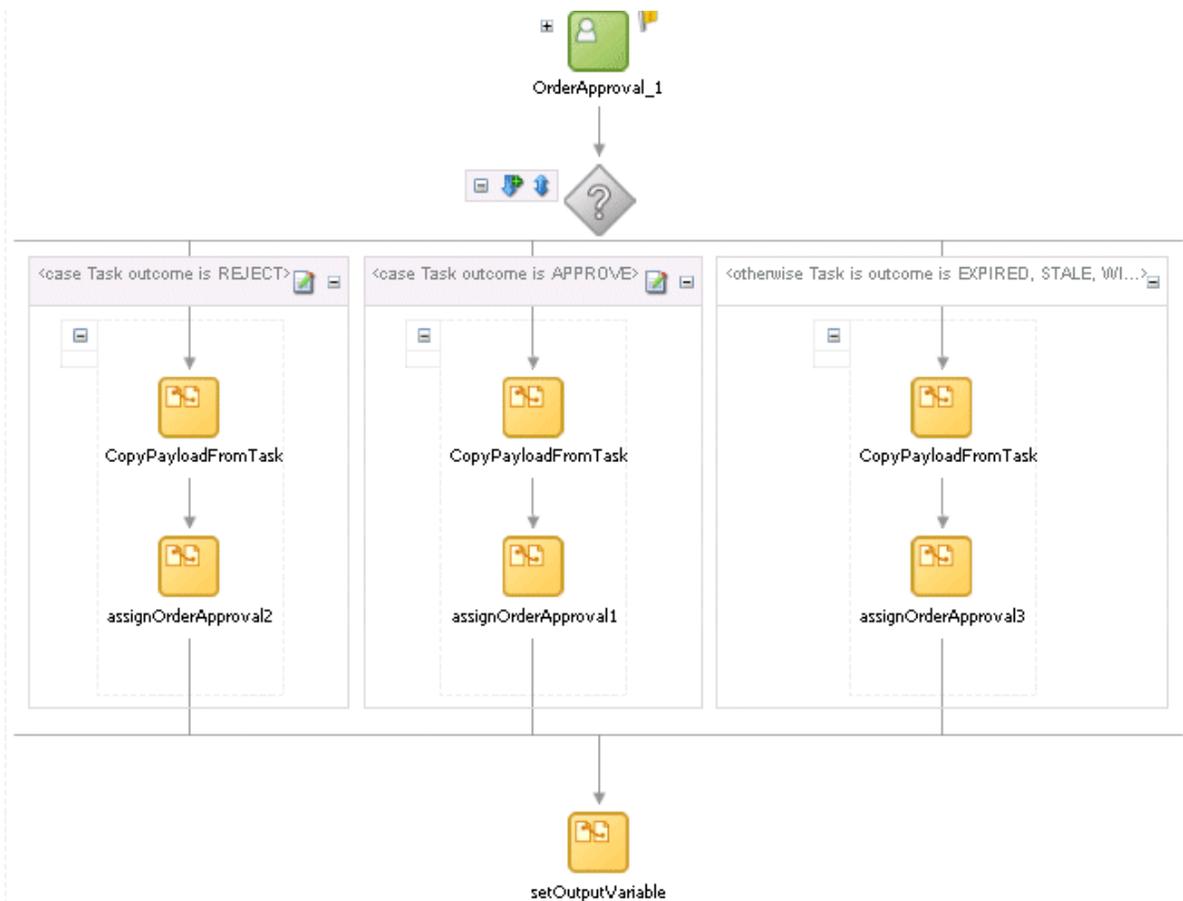
`/ns1:PurchaseOrder/ns1:OrderInfo/ns1:OrderComments`

- Click **OK** to close the Create Copy Operation window and the Assign window.
- Repeat Steps 3 through 11 to create an **Assign** activity below the **copyPayloadFromTask Assign** activity in the **<case Task outcome is REJECT>** section. Enter the same details as described above, with the following exceptions:
 - Name it **assignOrderApproval2**
 - Set the **Expression** field to **string('Rejected')**
- Repeat Steps 3 through 11 to create an **Assign** Activity below the **copyPayloadFromTask Assign** activity in the **<otherwise>** section. Enter the same details as described above, with the following exceptions:
 - Name it **assignOrderApproval3**
 - Set the **Expression** field to **string('Rejected')**
- Drag and drop an **Assign** activity from the **Component Palette** section to below (outside) the **Switch** activity.
- Double-click the **Assign** icon to display the Assign window.
- Click the **General** tab.
- Enter **setOutputVariable** in the **Name** field.
- Click **Apply**.
- Click the **Copy Operation** tab.
- Click **Create** and select **Copy Operation**.
- Enter the following details:

| Field | Value |
|---|---|
| From | |
| <ul style="list-style-type: none"> Type | Variable |
| <ul style="list-style-type: none"> Variables | Expand and select Variables > inputVariable > payload > ns1:PurchaseOrder Note: The namespace number values (for example, ns1 , ns2) can vary. Use the namespace values that automatically appear. |
| To | |
| <ul style="list-style-type: none"> Type | Variable |
| <ul style="list-style-type: none"> Variables | Expand and select Variables > outputVariable > payload > ns1:PurchaseOrder |

- Click **OK** to close the Create Copy Operation window and the Assign window.

The process looks as follows:



23. Select **Save** from the **File** main menu.
24. Click the - sign to close the **taskSwitch Switch** activity.

Validating, Compiling, and Deploying the Order Approval Process

1. Go to the **Application Navigator** section.
2. Right-click **OrderApproval**.
3. Select **Deploy > my_integration_server_connection > Deploy to default domain**.

This compiles the BPEL process. Check for errors by clicking the buttons at the bottom of the window. If there are no errors, deployment was successful. If deployment was unsuccessful, see Step 4 on page 2-17.

Running the Order Approval Process

1. Log into Oracle BPEL Control by selecting **Start > All Programs > Oracle - Oracle_Home > Oracle BPEL Process Manager > BPEL Control** (or refresh the page if it is already open).
The **Dashboard** tab of Oracle BPEL Control appears.
2. Enter the following details to log into Oracle BPEL Control and click **Login**:

| Field | Value |
|----------|-----------|
| Username | oc4jadmin |
| Password | password |

Where *password* is the value you entered during installation.

All services are running if the **TaskActionHandler** and **TaskManager** services appear in the Dashboard tab.

3. Click **OrderApproval** in the **Deployed BPEL Processes** list.
4. Enter details in all fields of the HTML Form. You can also use a text editor to copy and paste these details from the contents of the `OrderBookingPO_1.xml` file:

Note: Do *not* copy and paste from Internet Explorer or Mozilla Firefox. You must use a text editor to perform this task.

- a. Click the **XML Source** radio button.
 - b. Copy and paste the contents of `OrderBookingPO_1.xml` into the field that appears.
5. Click **Post XML Message**.

The **BPEL Processes** tab displays a message similar to the following:

```
Test Instance Initiated
```

6. Click the **Instances** tab at the top.
7. Click the **OrderApproval** instance.
A message appears indicating that the instance is active.
8. Select **Start > All Programs > Oracle - Oracle_Home > Oracle BPEL Process Manager > Sample Worklist Application** to access the login window for Oracle BPEL Worklist Application:
9. Log in as `jcooper/welcome1`.
This displays Oracle BPEL Worklist Application. A task waiting to be approved appears.
10. Select **Claim** in the **Actions** list for the task to approve.

11. Click **Go**.

The task details and payload information appear. Since this task is configured to be approved by two users, the task is routed to `jstein`, the manager of `jcooper` if it is accepted. If the task is rejected, the **OrderApproval** process identifies the task as rejected and completes processing.

12. Review the information. For example, the following information appears if you copied and pasted in the contents of `OrderBookingPO_1.xml`.

ORACLE BPM Worklist Home | Reports | Preferences | Log
 Welcome, jcooper [jazn.com]

✓ Your request was processed successfully.

[10000] Purchase Order Approval for Antilla Electronics

| | | |
|-------------------------|---|--------------------------------------|
| State : ASSIGNED | Created Date : 05/19/06 | Creator : Antilla Electronics |
| Outcome : | Updated Date : 05/19/06 09:32 AM | Acquired By : jcooper |
| Priority : 3 | Expiration Date : | Assignees : Supervisor(U) |

Purchase Order

Cust ID : Antilla Electronics string

ID : AS-0193 string

Name

First : DeGossaa string

Last : Calro string

Address

Street : 12, Burhon Street string

City : Danzig string

State : Godanzk string

Zip : PO-20398 string

Country : Poland string

Name

First : Mrs string

Last : Castro string

13. Select **Approve** from the **Task Action** list and click **Go**.
14. Log out as user jcooper.
15. Log into Oracle BPEL Worklist Application as jstein/welcome1.
16. Select **Approve** from the **Actions** list and click **Go**.
 After processing, no tasks appear in Oracle BPEL Worklist Application.
17. Log out.
18. Return to Oracle BPEL Control.
19. Click the **Instances** tab at the top.
20. Click the **OrderApproval** instance.
 A message appears indicating that the instance has completed.
21. Click the **Audit** and **Flow** links to observe additional details about the completed **OrderApproval** process.

Creating a Partner Link for the Order Approval Service

Summary: You now create a partner link for the **OrderApproval** service.

1. Ensure that you are in the **OrderBooking** process.
2. Double-click **OrderBooking.bpel**.

3. Select **Services** from the **Component Palette** section.
4. Drag and drop a **PartnerLink** activity from the **Component Palette** section onto the right side of the designer window.
5. Enter the following values to create a partner link for the **OrderApproval** service:

Note: For the **WSDL File** field below, click the **flashlight** (the second icon from the left named **Service Explorer**) to access the Service Explorer window for automatically selecting the **OrderApproval** service.

| Field | Value |
|-------------------|--|
| Name | POAService |
| WSDL File | <p>Access this URL by clicking the Service Explorer flashlight icon and expanding and selecting BPEL Services > my_integration_server_connection > processes > default > OrderApproval</p> <p>http://localhost:8888/orabpel/default/OrderApproval/OrderApproval?wsdl</p> <p>See Also: "Setting the Hostname in Your Oracle JDeveloper Web Browser Preferences" on page 1-7 if you receive a parsing error when attempting to add a WSDL file in the Service Explorer window.</p> |
| Partner Link Type | OrderApproval |
| Partner Role | OrderApprovalProvider |
| My Role | OrderApprovalRequester |

6. Click **OK**.
7. Select **Save** from the **File** main menu.

Creating a Scope Activity

Summary: You now create a **Scope** activity to group all activities that form a logical step to be executed.

1. Drag and drop a **Scope** activity from the **Component Palette** section to below the **SelectSupplier Switch** activity.
2. Double-click the **scope** icon to display the Scope window.
3. Enter **UserApproval** in the **Name** field of the **General** tab.
4. Click **OK**.

In this phase of the tutorial, you create the **Invoke**, **Receive**, and **Assign** activities inside this Scope activity.

5. Click the + sign to expand the **Scope** activity.
6. Select **Save** from the **File** main menu.

Creating Invoke and Receive Activities

1. Drag and drop an **Invoke** activity from the **Component Palette** section into the **UserApproval Scope** activity.

2. Double-click the **Invoke** icon to display the Invoke window.
3. Enter the following details:

| Field | Value |
|--------------|------------------|
| Name | invokePOAService |
| Partner Link | POAService |

The **Operation (initiate)** field is automatically filled in.

4. Click the first icon to the right of the **Input Variable** field. This is the automatic variable creation icon.
5. Click **OK** on the Create Variable window that appears.
A variable named **invokePOAService_initiate_InputVariable** is automatically created in the **Input Variable** field. This variable is automatically assigned a message type of **OrderApprovalRequestMessage**.
6. Click **OK**.
7. Drag and drop a **Receive** activity to below the **invokePOAService Invoke** activity inside the **UserApproval Scope** activity.
8. Double-click the **Receive** icon to display the Receive window.
9. Enter the following details:

| Field | Value |
|--------------|-------------------|
| Name | receivePOAService |
| Partner Link | POAService |

The **Operation (onResult)** field is automatically filled in.

10. Click the first icon to the right of the **Variable** field. This is the automatic variable creation icon.
11. Click **OK** on the Create Variable window that appears.
A variable named **receivePOAService_onResult_InputVariable** is automatically created in the **Variable** field. This variable is automatically assigned a message type of **OrderApprovalResponseMessage**.
12. Click **OK**.
13. Select **Save** from the **File** main menu.

Creating an Initial Assign Activity

1. Drag and drop an **Assign** activity from the **Component Palette** section to above the **invokePOAService Invoke** activity and inside the **UserApproval Scope** activity.
2. Double-click the **Assign** icon to display the Assign window.
3. Enter **assignPOADatIn** in the **Name** field of the **General** tab.
4. Click **Apply**.
5. Click the **Copy Operation** tab.

6. Click **Create** and select **Copy Operation** to display the Create Copy Operation window.
7. Enter the following details:

| Field | Value |
|-------------|--|
| From | |
| ▪ Type | Variable |
| ▪ Variables | Expand and select Variables > inputVariable > payload |
| To | |
| ▪ Type | Variable |
| ▪ Variables | Expand and select Variables > invokePOAService_initiate_inputVariable > payload |

8. Click **OK** to close the Create Copy Operation window and the Assign window.
9. Select **Save** from the **File** main menu.

Creating a Second Assign Activity

1. Drag and drop an **Assign** activity from the **Component Palette** section to below the **receivePOAService Receive** activity and inside the **UserApproval Scope** activity.
2. Double-click the **Assign** icon to display the Assign window.
3. Enter **UserResponse** in the **Name** field of the **General** tab.
4. Click **Apply**.
5. Click the **Copy Operation** tab.
6. Click **Create** and select **Copy Operation** to display the Create Copy Operation window.
7. Enter the following details:

| Field | Value |
|-------------|--|
| From | |
| ▪ Type | Variable |
| ▪ Variables | Expand and select Variables > receivePOAService_onResult_InputVariable > payload > ns1:PurchaseOrder > ns1:OrderInfo > ns1:OrderComments Note: The namespace number values (for example, ns1 , ns2) can vary. Use the namespace values that automatically appear. |
| To | |
| ▪ Type | Variable |
| ▪ Variables | Expand and select Variables > inputVariable > payload > ns1:PurchaseOrder > ns1:OrderInfo > ns1:OrderStatus |

8. Click **OK**.
9. Click **Create** and select **Copy Operation** in the **Copy Operation** tab again to display the Create Copy Operation window.

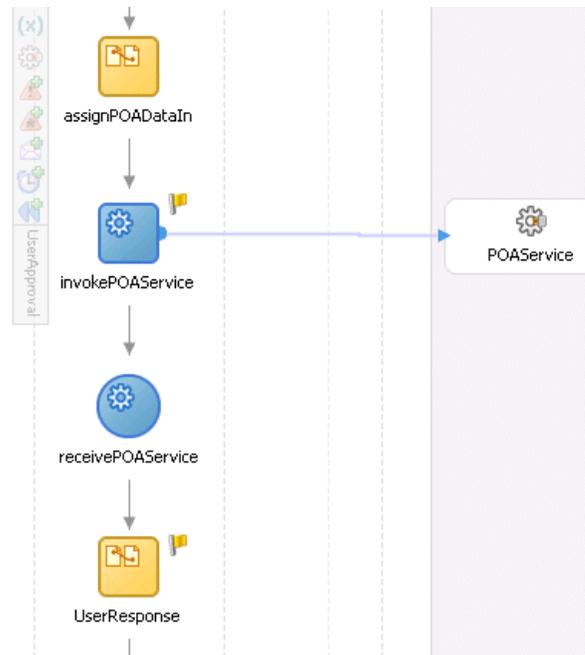
10. Enter the following details to create a *second* statement:

Note: Instead of manually entering an expression, you can press **Ctrl** and then the space bar in the **Expression** field. Scroll through the list of values that appears and double-click the value you want. Edit the value as necessary. As you enter information, a trailing slash can appear. This means you are being prompted for additional information. Either enter additional information, or press the **Esc** key and delete the trailing slash to complete the input of information.

| Field | Value |
|--------------|---|
| From | |
| ■ Type | Expression |
| ■ Expression | concat(bpws:getVariableData('inputVariable','payload','/ns1:PurchaseOrder/ns1:OrderInfo/ns1:OrderComments'),',', User Review Complete') |
| To | |
| ■ Type | Variable |
| ■ Variables | Expand and select Variables > inputVariable > payload > ns1:PurchaseOrder > ns1:OrderInfo > ns1:OrderComments |

11. Click **OK** to close the Create Copy Operation window and the Assign window.

When complete, the designer window looks as follows:



12. Click the - sign to close the **UserApproval Scope** activity.

13. Select **Save** from the **File** main menu.

Validating, Compiling, and Deploying the Order Process

1. Go to the **Application Navigator** section.
2. Right-click **OrderBooking**.
3. Select **Deploy > my_integration_server_connection > Deploy to default domain**.
4. Increment the version number of the project when prompted (for example, enter 1.7) and click **OK**.

This compiles the BPEL process. Check for errors by clicking the buttons at the bottom of the window. If there are no errors, deployment was successful. If deployment was unsuccessful, see Step 4 on page 2-17.

Running and Verifying the Order Process

1. Log into Oracle BPEL Control by selecting **Start > All Programs > Oracle - Oracle_Home > Oracle BPEL Process Manager > BPEL Control** (or refresh the page if it is already open).
2. Enter the following details to log into Oracle BPEL Control and click **Login**:

| Field | Value |
|----------|-----------|
| Username | oc4jadmin |
| Password | password |

3. Verify that the processes are deployed (**OrderApproval**, **OrderBooking**, and all other processes, including **TaskActionHandler**). **OrderApproval** uses **TaskActionHandler** with a more sophisticated user interaction flow compared to the simpler **TaskManager** used by **SelectManufacturing**.
4. Start the correct **OrderBooking** version instance in one of the following ways:
 - Copy *SOA_Oracle_Home\bpel\samples\tutorials\127.OrderBookingTutorial\PracticeFiles\OrderBookingPO_1.xml* to a temporary directory (*drive_letter:\temp* on Windows operating systems or */temp* on Unix operation systems).
 - Click the process and provide input to the default graphical user interface page in either of the following ways:
 - Enter details in all fields of the HTML Form and click **Post XML Message**.
 - Click the **XML Source** radio button. Use a text editor to copy and paste the contents of the *OrderBookingPO_1.xml* file into the field that appears. Click **Post XML Message**.

Note: Do *not* copy and paste from Internet Explorer or Mozilla Firefox. You must use a text editor to perform this task.

- Open the provided **CreateOrderBookingUI** application at <http://localhost:8888/CreateOrderBookingUI> and provide inputs to initiate a process.
5. Verify that the **OrderBooking** process has started.

6. Provide a price quote for **SelectManufacturing** from the following graphical user interface:

<http://localhost:8888/SelectManufacturingUI>

After providing a quote, watch the process instance execute and invoke the user approval service at **OrderApproval**. Go to Oracle BPEL Control and verify that the **OrderApproval** process and **TaskActionHandler** process have both started.

7. Select **Start > All Programs > Oracle - Oracle_Home > Oracle BPEL Process Manager > Sample Worklist Application** to access the login window for Oracle BPEL Worklist Application.
8. Log in as `jcooper/welcome1`. Default users are provided at install time in the following configuration file:

`SOA_Oracle_Home\bpel\system\appserver\oc4j\j2ee\home\config\jazn-data.xml`

ORACLE
BPEL Worklist

User: [jcooper](#) | [Home](#) | [Logout](#)

Search: My & Group Any Assigned Go

Keyword Category Priority Status [Advanced Search](#)

Home (My & Group Tasks)

| Number ↓ | Title | Priority | Status | Assignee | Expiration Date | Modified Date | Actions |
|----------|---|----------|----------|--------------------------------|----------------------|---------------------|--|
| 10001 | Order For Antilla Electronics | 3 | Assigned | Supervisor (G) | May 9, 2005 12:27 AM | May 8, 2005 2:27 PM | -- Select an Action -- Go |

Data refreshed on May 8, 2005 2:29 PM

The list shows the task awaiting user actions.

9. Select **Claim** in the **Actions** list and click **Go** to claim a task.
10. View the task details, the comments section, and so on.
11. Approve or reject the task by selecting **Accept** or **Reject** from the **Task Action** list, and click **Go**.
12. Log out of user `jcooper`. Rejects go directly back to the process, while approved tasks require the next level of approval.
13. Log in as `jstein/welcome1` (the default manager of `jcooper` in the hierarchy defined in the `jazn-data.xml` configuration file listed in Step 8).

The list shows the task awaiting user actions.

Home (My & Group Tasks)

| Number ↓ | Title | Priority | Status | Assignee | Expiration Date | Modified Date | Actions |
|----------|---|----------|----------|----------------------------|----------------------|---------------------|--|
| 10001 | Order For Antilla Electronics | 3 | Assigned | jstein (U) | May 9, 2005 12:27 AM | May 8, 2005 2:34 PM | -- Select an Action -- Go |

14. View the task details, the comments section, and so on.
15. Approve or reject the task in the **Actions** list and click **Go**.

After the `jstein` comments, the approved tasks of `jcooper` are returned to the process.

16. View Oracle BPEL Control, database tables, or temporary directory you specified (*drive_letter*: \temp on Windows operating systems or /temp on Unix operation systems) for completion of the process (as examined in previous tutorial phases).

Summary

You have now added user tasks to handle the manual approval or rejection of a purchase order. This phase extended the previous project steps by calling an external manual service through human task capabilities. It included a call out to another service that handled human tasks and used predefined (default) graphical user interface screens for user actions. You performed the following key tasks:

- Created a new BPEL process named **OrderApproval**.
- Designed the BPEL process to add workflow functionality to handle the manual approval or rejection of a purchase order.
- Deployed the **OrderApproval** BPEL process.
- Ran the **OrderApproval** BPEL process from Oracle BPEL Control.
- Accessed the Oracle BPEL Worklist Application URL to approve the order.
- Returned to the **OrderBooking** BPEL process and created a partner link (named **POAService**) that interacted with the **OrderApproval** BPEL process.
- Designed the BPEL process to invoke **POAService**.
- Deployed the **OrderBooking** BPEL process.
- Ran the **OrderBooking** BPEL process from Oracle BPEL Control.
- Accessed the Oracle BPEL Worklist Application URL to approve the order.

Using Sensors

This chapter of the tutorial describes how to insert sensors on activities in the **OrderBooking** process.

This chapter contains the following topics:

- [Introduction](#)
- [Adding Sensors](#)
- [Summary](#)

Introduction

This phase of the tutorial adds to the process you designed in [Chapter 11, "Designing the Human Task Workflow System"](#). Ensure that you have successfully completed that phase before performing this one. In this phase of the tutorial, you insert sensors on activities in the **OrderBooking** process. Sensors capture and publish business event information to Java Message Service (JMS) topics. This provides business analysts with real-time access to critical business performance indicators. Business analysts can use this information to analyze and improve the speed and effectiveness of critical business activities and remove delays in managing and executing critical business processes. You perform the following key tasks:

- Create sensors on activities in the **OrderBooking** BPEL process.
- Deploy the **OrderBooking** BPEL process.
- Run the **OrderBooking** BPEL process from Oracle BPEL Control.
- Observe the sensor results.

Adding Sensors

This section contains these tasks:

- [Creating Sensors](#)
- [Running, Validating, and Deploying the OrderBooking Process](#)
- [Running the Order Booking Process](#)

Creating Sensors

1. Double-click **OrderBooking.bpel** in the **Application Navigator** section.
2. Double-click the **receiveInput Receive** activity.

3. Click the **Sensors** tab.
4. Click **Create** to access the Create Activity Sensor window.
5. Enter **InstanceStart** in the Name field.
6. Select **Completion** from the **Evaluation Time** list.
7. Click **Create** in the **Activity Variable Sensors** section.
The Create Activity Variable Sensor window appears.
8. Click the icon to the right of the **Variable XPath** field.
The Variable XPath Builder window appears.
9. Expand and select **Variables > inputVariable > payload**.
10. Click **OK**.

The Create Activity Variable Sensor window displays the following details:

Variable XPath: 

Output Namespace:

Output Datatype:

11. Click **OK** to return to the Create Activity Sensor window.
12. Click **Add** in the **Sensor Actions** section.
The Sensor Action Chooser window appears.
13. Select **Sensor Actions** and click the **Create** icon.
14. Select **Sensor Action** from the list that appears.
The Create Sensor Action window appears.
15. Enter **ReportsPublisher** in the **Name** field.
16. Select **Database** from the **Publish Type** list.
17. Click **OK**.
The Sensor Action Chooser window appears.
18. Select **ReportsPublisher** and click **OK** to return to the Create Activity Sensor window.
19. Click **Add** in the **Sensor Actions** section.
The Sensor Action Chooser window appears.
20. Select **Sensor Actions** and click the **Create** icon.
21. Select **Sensor Action** from the list that appears.
The Create Sensor Action window appears.
22. Enter the following details.

| Field | Description |
|--------------|--------------|
| Name | BAMPublisher |
| Publish Type | JMS Topic |

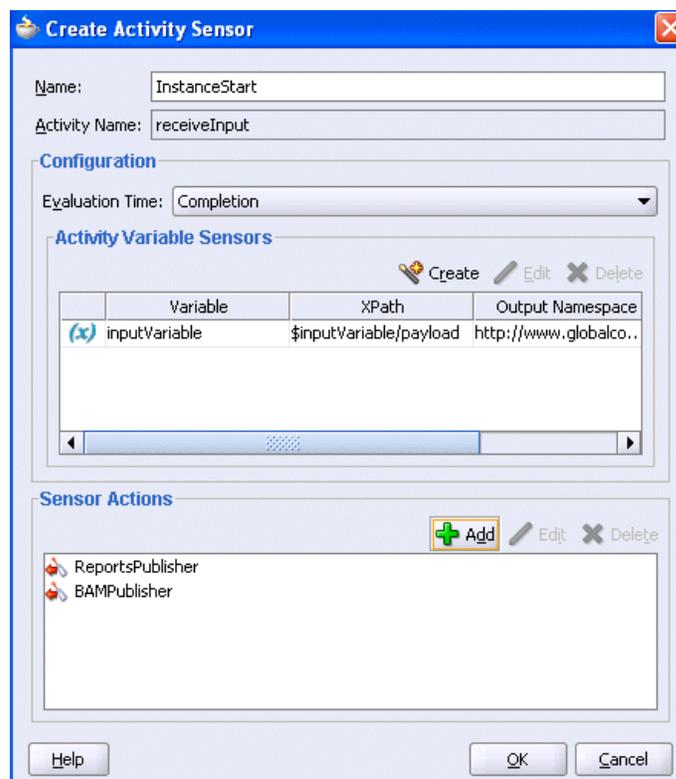
| Field | Description |
|------------------------|----------------------------|
| JMS Connection Factory | jms/TopicConnectionFactory |
| Publish Target | jms/demoTopic |
| Filter | Leave empty. |
| Enable | Leave selected. |

23. Click **OK**.

The Sensor Action Chooser window appears.

24. Select **BAMPublisher** and click **OK** to return to the Create Activity Sensor window.

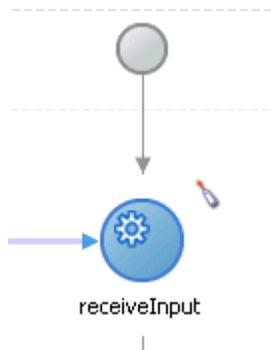
The Create Activity Sensor window appears as follows:



25. Click **OK** in the Create Activity Sensor window and the Receive window.

26. Select **Save** from the **File** main menu.

The **receiveInput Receive** activity now includes a sensor icon in the upper right.



27. Repeat the steps in this section to create additional sensors on the following **OrderBooking** activities. For all these activities, add both existing sensor actions: **ReportsPublisher** and **BAMPublisher**. You do not need to create additional sensor actions.

| Create Sensor on Activity... | Name to Give Sensor... | Evaluation Time To Select... | Variable XPath To Select... |
|------------------------------|------------------------|------------------------------|--|
| receiveInput | InstanceStart | Completion | \$inputVariable/payload |
| receiveSM | ReceivedSMPrice | Completion | \$receiveSM_onResult_InputVariable/payload |
| receiveRD | ReceivedRDPrice | Completion | \$receiveRD_onResult_InputVariable/payload |
| invokePOAService | UserReviewStart | Completion | \$inputVariable/payload |
| receivePOAService | UserReviewComplete | Completion | \$inputVariable/payload |
| callbackClient | InstanceCompletion | Completion | \$inputVariable/payload |

28. Select **Save** from the **File** main menu.

Running, Validating, and Deploying the OrderBooking Process

1. Log into Oracle BPEL Control by selecting **Start > All Programs > Oracle - Oracle_Home > Oracle BPEL Process Manager > BPEL Control** (or refresh the page if it is already open).
2. Enter the following details to log into Oracle BPEL Control and click **Login**:

| Field | Value |
|----------|-----------------|
| Username | oc4jadmin |
| Password | <i>password</i> |

Where *password* is the value you entered during installation.

3. Click the latest **OrderBooking** instance process to undeploy in the **Dashboard** tab.
4. Click the **Manage** link.
5. Click **Undeploy** at the bottom of the window.
6. Click **OK** when prompted to mark all open instances for this process as stale.
7. Click **Done** when prompted.

8. Go to the **Application Navigator** section in Oracle JDeveloper.
9. Right-click **OrderBooking**.
10. Select **Deploy > my_integration_server_connection > Deploy to default domain**.
11. Increment the version number of the project when prompted and click **OK**.

This compiles the BPEL process. Check for errors by clicking the buttons at the bottom of the window. If there are no errors, deployment was successful. If deployment was unsuccessful, see Step 4 on page 2-17.

Running the Order Booking Process

1. Log into Oracle BPEL Control by selecting **Start > All Programs > Oracle - Oracle_Home > Oracle BPEL Process Manager > BPEL Control** (or refresh the page if it is already open).
2. Enter the following details to log into Oracle BPEL Control and click **Login**:

| Field | Value |
|----------|-----------|
| Username | oc4jadmin |
| Password | password |

3. Verify that the **OrderBooking** instance appears under the **Dashboard** tab.
4. Click the **OrderBooking** instance.
5. Click the **XML Source** radio button.
6. Use a text editor to copy and paste the contents of `OrderBookingPO_1.xml` into the field that appears.

Note: Do *not* copy and paste from Internet Explorer or Mozilla Firefox. You must use a text editor to perform this task.

7. Click **Post XML Message**.
8. Click **Audit Instance**.
Note that the **SelectManufacturing** process is pending and waiting for a price quote.
9. Provide a price quote for **SelectManufacturing** from the following graphical user interface:
`http://localhost:8888/SelectManufacturingUI`
10. Refresh Oracle BPEL Control and note that the process is pending on **POAService**.
11. Select **Start > All Programs > Oracle - Oracle_Home > Oracle BPEL Process Manager > Sample Worklist Application** to access the login window for Oracle BPEL Worklist Application.
12. Log in as `jcooper/welcome1`.
The list shows the task awaiting user actions.
13. Select **Claim** in the **Actions** list and click **Go** to claim a task.
14. Select **Approve** from the **Task Action** list, and click **Go**.

15. Log out as user `jcooper`.
16. Log in as `jstein/welcome1` (the default manager of `jcooper` in the hierarchy).
17. Select **Approve** in the **Task Action** list, and click **Go**.
18. Refresh Oracle BPEL Control and confirm that the instance has completed.
19. Click the **Sensor Values** link to confirm the sensor results.

Summary

You have now added sensors to the **OrderBooking** process. You performed the following key tasks:

- Created sensors on activities in the **OrderBooking** BPEL process.
- Deployed the **OrderBooking** BPEL process.
- Ran **OrderBooking** BPEL process from Oracle BPEL Control.
- Observed the sensor results.

Using Notifications

This chapter of the tutorial describes how to add notifications.

This chapter contains the following topics:

- [Introduction](#)
- [Adding an E-mail Notification to the POAcknowledge Process](#)
- [Summary](#)

Introduction

This phase of the tutorial adds notifications to the **POAcknowledge** process. Ensure that you have successfully completed the phases in [Chapter 9, "Using the File Adapter's Write Functionality"](#) and [Chapter 12, "Using Sensors"](#) before performing this phase. Notifications enable you to send notification about an event to a user, group, or destination address. You can send a notification by e-mail, voice mail, or Short Message Service (SMS). In this phase, you send an e-mail notification to confirm receipt of an order. You perform the following tasks:

- Configure your company e-mail environment.
- Create a notification activity in the **POAcknowledge** process.
- Deploy the **POAcknowledge** BPEL process.
- Run **POAcknowledge** from Oracle BPEL Control to send an e-mail notification.

Adding an E-mail Notification to the POAcknowledge Process

This section contains the following tasks:

- [Configuring Your Company E-Mail Environment](#)
- [Creating an E-mail Service](#)
- [Validating, Compiling, and Deploying the POAcknowledge Process](#)
- [Running the POAcknowledge Process](#)

Configuring Your Company E-Mail Environment

1. Go to the `SOA_Oracle_Home\bpel\system\services\config` directory.
2. Make a backup copy of the `ns_emails.xml` file.
3. Open a text editor.

- Open `ns_emails.xml` and replace `NONE` with `EMAIL` for the `NotificationMode` parameter. Replace the other parameter values shown in bold with ones appropriate to your company environment.

```
<EmailAccounts xmlns="http://xmlns.oracle.com/ias/pcbpel/NotificationService"
  EmailMimeCharset=" "
  NotificationMode="NONE">
  <EmailAccount>
    <Name>Default</Name>
    <GeneralSettings>
      <FromName>Oracle BPM</FromName>
      <FromAddress>accountId@yourdomain.com</FromAddress>
    </GeneralSettings>
    <OutgoingServerSettings>
      <SMTPHost>yourdomain.com</SMTPHost>
      <SMTPPort>25</SMTPPort>
    </OutgoingServerSettings>
    <IncomingServerSettings>
      <Server>yourdomain.com</Server>
      <Port>110</Port>
      <Protocol>pop3</Protocol>
      <UserName>accountId</UserName>
      <Password ns0:encrypted="false"
        xmlns:ns0="http://xmlns.oracle.com/ias/pcbpel/NotificationService">
        password</Password>
      <UseSSL>>false</UseSSL>
      <Folder>Inbox</Folder>
      <PollingFrequency>1</PollingFrequency>
      <PostReadOperation>
        <MarkAsRead/>
      </PostReadOperation>
    </IncomingServerSettings>
  </EmailAccount>
<!--EmailAccount>
```

- Change the e-mail addresses for the users `jcooper` and `jstein` to e-mail accounts you can access. If an XML-based JAZN provider is used, these properties can be changed in

```
SOA_Oracle_Home\bpel\system\services\config\users-properties.xml
```

The following code example shows where to change e-mail addresses:

```
<userObject>
  <name>jstein</name>
  <description>Demo User</description>
  <email>user2@hotmail.us.com</email>
  <title>Manager2</title>
  <firstName>John</firstName>
  <lastName>Steinback</lastName>
  <manager>wfaulk</manager>
  <timeZone>America/Los_Angeles</timeZone>
  <languagePreference>en-US</languagePreference>
  <notificationPreferences>Mail</notificationPreferences>
</userObject>
```

- Save your changes and exit the file.
- Restart Oracle JDeveloper.

Creating an E-mail Service

1. Double-click **POAcknowledge.bpel** in the **Application Navigator**.
2. Select **Process Activities** from the **Component Palette** section.
3. Drag and drop an **Email** service from the **Component Palette** section to below the **InvokeFileOut Invoke** activity.
4. Enter the following details:

Note: Instead of manually entering an expression, you can click the icon to the right of the field to display the XPath Expression Builder to dynamically build your expression.

| Field | Description |
|---------------------|--|
| From Account | Default |
| To: | Enter a valid e-mail address. |
| Cc: | Enter a valid e-mail address. |
| Bcc: | Enter a valid e-mail address. |
| Reply To: | Enter a valid e-mail address. |
| Subject: | PO Acknowledgment for Items Ordered and Price |
| Body: | <p>Press the icon to the right of this field to display the XPath Expression Builder window. Use this window to build the following expression, and click OK when complete.</p> <pre>concat(string('Dear Customer, Your order number '),bpws:getVariableData('inputVariable','payload',/ns1:PurchaseOrder/ns1:ID'),string(' for customer '),bpws:getVariableData('inputVariable','payload',/ns1:PurchaseOrder/ns1:CustID'),string(' has been confirmed. The order status is '),bpws:getVariableData('inputVariable','payload',/ns1:PurchaseOrder/ns1:OrderInfo/ns1:OrderStatus'),string('". Thanks for using Order Booking Application from Oracle BPEL Process Manager. Administrator.')</pre> |

5. Click **OK**.

Validating, Compiling, and Deploying the POAcknowledge Process

1. Go to the **Application Navigator** section.
2. Right-click **POAcknowledge**.
3. Select **Deploy > my_integration_server_connection > Deploy to default domain**.
4. Increment the version number again of the project when prompted and click **OK** (for example, enter **1.1**).

This compiles the BPEL process. Check for errors by clicking the buttons at the bottom of the window. If there are no errors, deployment was successful. If deployment was unsuccessful, see Step 4 on page 2-17.

Running the POAcknowledge Process

1. Log into Oracle BPEL Control by selecting **Start > All Programs > Oracle - Oracle_Home > Oracle BPEL Process Manager > BPEL Control** (or refresh the page if it is already open).
2. Enter the following details to log into Oracle BPEL Control and click **Login**:

| Field | Value |
|-----------------|------------------------|
| Username | <code>oc4jadmin</code> |
| Password | <code>password</code> |

Where `password` is the value you entered during installation.

The **Dashboard** tab of Oracle BPEL Control appears. Note that your BPEL process, **POAcknowledge**, now appears in the **Deployed BPEL Processes** list.

3. Click the **POAcknowledge** version you increased in Step 4 on page 13-3 in the **Deployed BPEL Processes** list.
4. Enter details in all fields of the HTML Form and click **Post XML Message**.
5. Refresh Oracle BPEL Control.
6. Click the **Instances** tab.
7. Click the **POAcknowledge** instance.
8. Click the **Audit** link to view progress.
9. Check your e-mail account to verify that e-mail has been delivered (there can be an e-mail server-based delivery delay).

Summary

You have now added notification logic to the process.

With the completion of this phase, the **OrderBooking** process is now completely configured for order acknowledgment through e-mail notification.

A

- activity sensors
 - creating, 12-3
- Adapter Configuration Wizard
 - configuring the database adapter, 10-5, 10-6
 - configuring the file adapter, 8-3, 9-2
- ant
 - deploying, 3-7
 - running, 2-2, 3-2
- application
 - creating, 2-3, 7-1, 8-2, 10-2
- assign activities
 - creating, 2-11, 2-13, 2-15, 3-5, 4-6, 5-2, 5-4, 8-6, 9-4, 9-8, 10-14, 10-15, 11-15, 11-16
- asynchronous services, 3-1, 3-9
 - creating a partner link, 3-2
 - creating an assign activity, 3-5
 - creating invoke and receive activities, 3-3
 - running the OrderBooking process, 3-8
 - starting and testing your services, 3-2
 - summary, 3-9
 - validating, compiling, and deploying, 3-7
- audit link
 - following the instance execution process, 2-18, 3-9, 4-10, 5-6, 6-4, 8-8, 9-5, 10-11, 10-21, 13-4

B

- BatchOrderProcessing process
 - deploying, 8-7
 - running, 8-7
- BPEL errors section
 - errors during validation, compilation, and deployment, 2-17
- browser
 - requirements, 1-6
 - setting browser preferences in Oracle JDeveloper, 1-7

C

- catch all branches
 - creating, 6-1, 10-17
- compiling
 - BatchOrderProcessing process, 8-7

- FulfillOrder process, 10-11
- OrderApproval process, 11-11
- OrderBooking process, 2-17, 3-7, 4-8, 5-5, 6-3, 9-10, 10-19, 11-18, 12-4
- POAcknowledge process, 9-5, 13-3
- concat function
 - using in transformations, 7-5, 7-6, 10-8
- conditional branching logic, 5-1
 - creating a switch activity, 5-2
 - creating an assign activity under the switch activity case block, 5-2
 - creating an assign activity under the switch activity otherwise block, 5-4
 - running the OrderBooking process, 5-6
 - summary, 5-6
 - validating, compiling, and deploying a process, 5-5
- connecting
 - to a database, 10-3, 10-5
- connections
 - configuring a JDeveloper database connection, 10-4
 - testing the database connection, 10-5
- console
 - starting, 1-5
- copy operation
 - creating, 2-12, 2-14, 2-16, 3-6, 5-3, 8-6, 9-4, 9-8, 9-9, 10-15, 10-17, 10-18, 11-10, 11-16, 11-17
- CreateOrderBookingUI application, 9-10
- CreditRatingService process, 2-3

D

- Dashboard tab
 - viewing deployed processes, 3-8, 8-8, 9-5, 9-10, 10-20, 11-11, 13-4
 - viewing started services, 2-3, 3-2
- database adapter, 10-1
 - adding exception handling, 10-16
 - creating a database connection and sample tables, 10-3
 - creating a JDeveloper database connection, 10-4
 - creating a partner link, 10-5, 10-12
 - creating a scope activity, 10-13
 - creating a transform activity, 10-7
 - creating an assign activity, 10-14

- creating an invoke activity, 10-7
- creating invoke and receive activities, 10-13
- creating sample tables, 10-4
- running the FulfillOrder process, 10-11
- running the OrderBooking process, 10-20
- summary, 10-21
- testing the database connection, 10-5
- testing transformation logic, 10-10
- validating, compiling, and deploying, 10-11, 10-19
- verifying that the database table is properly inserted, 10-12
- database tables, 10-3
 - creating, 10-4
- DBInsert process
 - selecting for a partner link, 10-6
- deploying
 - BatchOrderProcessing process, 8-7
 - FulfillOrder process, 10-11
 - OrderApproval process, 11-11
 - OrderBooking process, 2-17, 3-7, 4-8, 5-5, 6-3, 9-10, 10-19, 11-18, 12-4
 - POAcknowledge process, 9-5, 13-3
 - using ant, 3-7
- developer prompt
 - starting, 1-5
 - using, 2-2, 3-2, 10-5

E

- e-mail
 - configuration for notifications, 13-1
- errors
 - during validation, compilation, and deployment, 2-17
 - parsing errors when creating a partner link, 2-8, 3-3, 4-3, 8-5, 9-6, 10-13, 11-14
- Esc key
 - stopping creation of XPath expressions, 2-14, 5-3, 9-8, 9-9, 10-15, 10-17, 10-18, 11-10, 11-16, 11-17
- exception management, 6-1
- Expression field
 - using keyboard shortcuts to enter information, 2-14, 5-3, 5-4, 9-9, 10-15, 11-17, 13-3

F

- fault handling and exception management, 6-1, 10-16
 - creating a catch all branch, 6-1, 10-17
 - running the OrderBooking process, 6-3
 - summary, 6-5
 - validating, compiling, and deploying, 6-3
- file adapter
 - read functionality, 8-1
 - creating a partner link, 8-3, 8-4
 - creating an assign activity, 8-6
 - creating receive and invoke activities, 8-5
 - running, 8-7

- summary, 8-8
- validating, compiling, and deploying, 8-7
- write functionality, 9-1
 - creating a partner link, 9-2, 9-6
 - creating a scope activity, 9-6
 - creating an assign activity, 9-4, 9-8
 - creating an invoke activity, 9-4
 - creating invoke and receive activities, 9-7
 - POAcknowledge process, 9-5
 - running the OrderBooking process, 9-10
 - summary, 9-11
 - validating, compiling, and deploying, 9-5, 9-10
- FileReadAdapter process
 - selecting for a partnerlink, 8-4
- flow activities
 - creating, 4-3
- flow link
 - following the instance execution process, 2-19, 3-9, 4-10, 5-6, 6-4, 8-8, 10-21
- for-each node
 - using in transformations, 7-7, 10-9
- FulfillOrder process
 - deploying, 10-11
 - running, 10-11
 - selecting for a partner link, 10-13

H

- human workflow, 11-1
 - creating a new project, 11-2
 - creating a partner link, 11-13
 - creating a scope activity, 11-14
 - creating a user task activity, 11-3
 - creating an assign activity, 11-15, 11-16
 - creating invoke and receive activities, 11-14
 - running the OrderApproval process, 11-11
 - running the OrderBooking process, 11-18
 - summary, 11-20
 - validating, compiling, and deploying, 11-11, 11-18

I

- insertTable.sql, 10-5
- installation, 1-1
- instances
 - state, 2-18, 4-9, 4-11
 - verifying creation of process instances, 2-18, 3-8, 4-9, 5-5, 5-6, 6-4, 8-8, 9-11
- integration server connection
 - using to deploy a process, 3-8, 4-8, 5-5, 6-3, 8-7, 9-5, 9-10, 10-11, 10-20, 11-11, 11-18, 12-5, 13-3
- Internet Explorer
 - do not copy and paste with, 3-8, 4-9, 5-6, 9-10, 10-20, 11-12, 11-18, 12-5
 - supported browser version, 1-6
- introduction
 - adding conditional branching logic, 5-1
 - adding parallel flow, 4-1

- adding transformation logic, 7-1
- creating fault handling and exception management, 6-1
- designing human workflow, 11-1
- designing the database adapter to insert data, 10-1
- invoking a synchronous service, 2-1
- invoking an asynchronous service, 3-1
- using notifications, 13-1
- using sensors, 12-1
- using the file adapter's read functionality, 8-1
- using the file adapter's write functionality, 9-1

invoke activities

- creating, 2-9, 3-3, 4-4, 8-5, 9-4, 9-7, 10-7, 10-13, 11-14

K

keyboard shortcuts

- using to enter information in Expression fields, 2-14, 5-3, 5-4, 9-9, 10-15, 11-17, 13-3

M

mapper file

- creating transformation mappings in, 7-4, 7-9, 10-8

Mozilla Firefox

- supported browser version, 1-6

N

notification activities

- creating, 13-3

notifications, 13-1

- creating a notification activity, 13-3
- running, 13-4
- summary, 13-4
- through e-mail, 13-1
- validating, compiling, and deploying, 13-3

ns_emails.xml file

- configuring for e-mail notifications, 13-1

O

Oracle BPEL Console

- accessing, 2-2, 2-18, 3-8, 4-9, 5-6, 6-3, 8-7, 9-5, 9-10, 10-11, 10-20, 11-11, 11-18, 12-4, 12-5, 13-4
- running a process, 2-18, 3-8, 4-9, 5-6, 6-3, 8-7, 9-5, 9-10, 10-11, 10-20, 11-11, 11-18, 12-4, 12-5, 13-4
- starting, 1-5

Oracle BPEL Process Manager

- starting components, 1-5

Oracle BPEL Server

- starting, 1-5

Oracle BPEL Worklist Application

- accessing, 11-12, 11-19, 12-5
- approving or rejecting a task, 11-12, 11-19, 12-5

Oracle JDeveloper

- setting hostname in Web browser preferences, 1-7

Order Booking

- business goal, 1-2
- business problem, 1-1
- business solution, 1-2
- deploying, 2-17
- how to use this tutorial, 1-4
- overview of tutorial, 1-1

OrderApproval process

- deploying, 11-11
- running, 11-11

OrderBooking process

- compiling, 5-5
- deploying, 3-7, 4-8, 6-3, 9-10, 10-19, 11-18, 12-4
- running, 2-18, 3-8, 4-9, 5-6, 6-3, 9-10, 10-20, 11-18, 12-5

OrderBookingPO_1.xml file

- copying and pasting content of, 3-8, 4-9, 5-6, 9-10, 10-20, 11-12, 11-18, 12-5

overview

- of tutorial, 1-1

P

parallel flow, 4-1

- creating a flow activity, 4-3
- creating a partner link, 4-2
- creating an assign activity, 4-6
- creating invoke and receive activities, 4-4
- running the OrderBooking process, 4-9
- starting and testing your services, 4-2
- summary, 4-11
- validating, compiling, and deploying, 4-8

partner links

- creating, 2-7, 4-2, 8-3, 8-4, 9-2, 9-6, 10-5, 10-12, 11-13
- creating asynchronous services, 3-2

phase 1

- invoking a synchronous service, 2-1

phase 10

- designing human workflow, 11-1

phase 11

- using sensors, 12-1

phase 12

- using notifications, 13-1

phase 2

- invoking an asynchronous service, 3-1

phase 3

- adding parallel flow, 4-1

phase 4

- adding conditional branching logic, 5-1

phase 5

- creating fault handling and exception management, 6-1

phase 6

- adding transformation logic, 7-1

phase 7

- using the file adapter's read functionality, 8-1

phase 8

- using file adapter's write functionality, 9-1

phase 9

- designing the database adapter to insert data, 10-1
- POAcknowledge process
 - deploying, 9-5, 13-3
 - running, 9-5, 13-4
- POAService process
 - selecting for a partner link, 11-14
- preface
 - heading
 - PH PrefaceHead, ix
 - PT PrefaceTitle, ix
- process instances
 - verifying creation of, 2-18, 3-8, 4-9, 5-5, 5-6, 6-4, 8-8, 9-11
- processes
 - incrementing versions, 6-3, 9-10, 10-20, 11-18
- projects
 - creating, 2-3, 7-1, 8-2, 10-2, 11-2
 - special characters in project names, 2-3, 7-2, 8-2, 10-2
- PT PrefaceTitle, ix

R

- RapidDistributors process, 3-2
- receive activities
 - creating, 3-3, 4-4, 8-5, 9-7, 10-13, 11-14
- running
 - BatchOrderProcessing, 8-7
 - FulfillOrder, 10-11
 - OrderApproval, 11-11
 - OrderBooking, 2-18, 3-8, 4-9, 5-6, 6-3, 9-10, 10-20, 11-18, 12-5
 - POAcknowledge, 9-5, 13-4

S

- scope activities
 - creating, 2-9, 9-6, 10-13, 11-14
 - creating a catch all branch inside a scope activity, 6-1, 10-17
- Select Manufacturing user application
 - accessing to enter a price quote, 4-2, 4-10, 5-6, 6-4, 8-8, 10-21, 11-19, 12-5
- SelectManufacturing process, 3-2
- sensor actions
 - creating, 12-2
- sensors, 12-1
 - creating, 12-1
 - creating activity sensors, 12-3
 - creating sensor actions, 12-2
 - running the OrderBooking process, 12-5
 - summary, 12-6
 - validating, compiling, and deploying, 12-4
- server
 - starting, 1-5
- Service Explorer
 - selecting a service, 2-8, 3-3, 4-2, 8-4, 9-3, 9-6, 10-6, 10-13, 11-14
- special characters

- in project or element names, 2-3, 7-2, 8-2, 10-2
- SQL Worksheet
 - running, 10-12
- starting
 - developer prompt, 1-5
 - Oracle BPEL Console, 1-5
 - Oracle BPEL Process Manager components, 1-5
 - Oracle BPEL Server, 1-5
 - server, 1-5
- summary, 3-9
 - conditional branching logic, 5-6
 - database adapter, 10-21
 - fault handling and exception management, 6-5
 - file adapter
 - read functionality, 8-8
 - write functionality, 9-11
 - human workflow, 11-20
 - notifications, 13-4
 - parallel flow, 4-11
 - sensors, 12-6
 - synchronous services, 2-20
 - transformation logic, 7-9
- switch activities
 - creating, 5-2
- synchronous services, 2-1
 - creating a partner link, 2-7
 - creating an application and project, 2-3, 7-1, 8-2, 10-2
 - creating an assign activity, 2-11, 2-13, 2-15
 - creating scope and invoke activities, 2-9
 - running the OrderBooking process, 2-18
 - starting and testing your services, 2-2
 - summary, 2-20
 - validating, compiling, and deploying, 2-17

T

- TaskActionHandler process, 2-3
- TaskManager process, 2-3
- testing
 - transformation logic, 7-8, 10-10
- text editor
 - use for copying and pasting, 3-8, 4-9, 5-6, 9-10, 10-20, 11-12, 11-18, 12-5
- transform activities
 - creating, 7-3, 10-7
- transformation logic, 7-1
 - creating a transform activity, 7-3, 10-7
 - summary, 7-9
 - testing, 7-8, 10-10
- transformation_1 mapper file, 7-4, 7-9, 10-8
- transformations
 - mapping sources and targets, 7-4, 10-8
 - using a concat function, 7-5, 7-6, 10-8
 - using a for-each mode, 7-7, 10-9
- tutorial
 - how to use, 1-4

U

user task activities
 creating, 11-3

V

validating
 BatchOrderProcessing process, 8-7, 10-19
 FulfillOrder process, 10-11, 11-18, 12-4
 OrderApproval process, 11-11
 OrderBooking process, 2-17, 3-7, 4-8, 5-5, 6-3, 9-10
 POAcknowledge process, 9-5, 13-3
variables
 automatically creating, 2-11, 3-4, 3-5, 4-5, 4-6, 8-5,
 8-6, 9-7, 9-8, 10-7, 10-14, 11-15
 manually creating, 10-17
versions
 of processes, 6-3, 9-10, 10-20, 11-18

W

wait activities
 creating, 10-19
Web browser
 requirements, 1-6
 setting browser preferences in Oracle
 JDeveloper, 1-7
while activities
 creating, 10-19
Workflow Wizard
 running, 11-3
WritePOA process
 selecting for a partner link, 9-6
WSDL file
 selecting for a partner link, 2-8, 3-3, 4-2, 8-4, 9-3,
 9-6, 10-6, 10-13, 11-14

X

XPath expressions
 creating, 2-12, 2-14, 2-16, 3-6, 5-2, 5-3, 8-6, 9-4, 9-8,
 9-9, 10-15, 10-17, 10-18, 11-10, 11-16, 11-17

