Oracle® BPEL Process Manager

Order Booking Tutorial

10*g* Release 2 (10.1.2)

Part No. B15813-01

August 17, 2005

ORACLE®

Oracle BPEL Process Manager Order Booking Tutorial, 10*g* Release 2 (10.1.2)

Part No. B15813-01

Primary Author: Mark Kennedy

Contributor: Prashant Nema, SrinivasaGandhi Sampath, Sanjeev Sondur

Contents

3 Invoking an Asynchronous Service

4 Adding Parallel Flow

5 Adding Conditional Branching Logic

6 Creating Fault Handling and Exception Management

7 Adding Transformation Logic

8 Using the File Adapter's Read Functionality

9 Using the File Adapter's Write Functionality

10 Designing the Database Adapter to Insert Data

11 Designing the Human Workflow System

12 Using Sensors

13 Using Notifications

Index

Send Us Your Comments

Oracle BPEL Process Manager Order Booking Tutorial, 10*g* Release 2 (10.1.2)

Part No. B15813-01

Oracle welcomes your comments and suggestions on the quality and usefulness of this publication. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most about this manual?

If you find any errors or have any other suggestions for improvement, please indicate the title and part number of the documentation and the chapter, section, and page number (if available). You can send comments to us in the following ways:

- Electronic mail: appserverdocs_us@oracle.com
- FAX: 650-506-7375. Attn: Oracle Application Server Documentation Manager
- Postal service:

  Oracle Corporation
  Oracle Application Server Documentation
  500 Oracle Parkway, Mailstop 1op4
  Redwood Shores, CA 94065
  USA

If you would like a reply, please give your name, address, telephone number, and electronic mail address (optional).

If you have problems with the software, please contact your local Oracle Support Services.

Preface

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

This preface contains the following topics:

Intended 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.

Structure

This guide consists of the following chapters:

Chapter 1, "Introduction"
This chapter describes the Order Booking tutorial scenario and how to get started.

Chapter 2, "Invoking a Synchronous Service"
This chapter of the tutorial describes how to invoke a synchronous service.

Chapter 3, "Invoking an Asynchronous Service"
This chapter of the tutorial describes how to invoke an asynchronous service.

Chapter 4, "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.

Chapter 5, "Adding Conditional Branching Logic"
This chapter of the tutorial describes how to build conditional branching logic.

Chapter 6, "Creating Fault Handling and Exception Management"

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

Chapter 7, "Adding Transformation Logic"

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

Chapter 8, "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.

Chapter 9, "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.

Chapter 10, "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.

Chapter 11, "Designing the Human Workflow System"

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

Chapter 12, "Using Sensors"

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

Chapter 13, "Using Notifications"

This chapter of the tutorial describes how to add notifications.

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

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

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

Conventions in Text

We use various conventions in text to help you more quickly identify special terms. The following table describes those conventions and provides examples of their use.

Convention	Meaning	Example
Bold	Bold typeface indicates terms that are defined in the text or terms that appear in a glossary, or both.	When you specify this clause, you create an **index-organized table**.
Italics	Italic typeface indicates book titles or emphasis.	*Oracle Database Concepts* Ensure that the recovery catalog and target database do *not* reside on the same disk.
`UPPERCASE monospace (fixed-width) font`	Uppercase monospace typeface indicates elements supplied by the system. Such elements include parameters, privileges, datatypes, RMAN keywords, SQL keywords, SQL*Plus or utility commands, packages and methods, as well as system-supplied column names, database objects and structures, usernames, and roles.	You can specify this clause only for a `NUMBER` column. You can back up the database by using the `BACKUP` command. Query the `TABLE_NAME` column in the `USER_TABLES` data dictionary view. Use the `DBMS_STATS.GENERATE_STATS` procedure.
`lowercase monospace (fixed-width) font`	Lowercase monospace typeface indicates executables, filenames, directory names, and sample user-supplied elements. Such elements include computer and database names, net service names, and connect identifiers, as well as user-supplied database objects and structures, column names, packages and classes, usernames and roles, program units, and parameter values. **Note:** Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown.	Enter `sqlplus` to start SQL*Plus. The password is specified in the `orapwd` file. Back up the datafiles and control files in the `/disk1/oracle/dbs` directory. The `department_id`, `department_name`, and `location_id` columns are in the `hr.departments` table. Set the `QUERY_REWRITE_ENABLED` initialization parameter to `true`. Connect as `oe` user. The `JRepUtil` class implements these methods.
`lowercase italic monospace (fixed-width) font`	Lowercase italic monospace font represents placeholders or variables.	You can specify the *`parallel_clause`*. Run *`old_release`*`.SQL` where *`old_release`* refers to the release you installed prior to upgrading.

Conventions in Code Examples

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

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

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

Convention	Meaning	Example
`[ ]`	Brackets enclose one or more optional items. Do not enter the brackets.	`DECIMAL (`*`digits`* `[ , `*`precision`* `])`
`{ }`	Braces enclose two or more items, one of which is required. Do not enter the braces.	`{ENABLE | DISABLE}`
`|`	A vertical bar represents a choice of two or more options within brackets or braces. Enter one of the options. Do not enter the vertical bar.	`{ENABLE | DISABLE}` `[COMPRESS | NOCOMPRESS]`

Convention	Meaning	Example
...	Horizontal ellipsis points indicate either: ■ That we have omitted parts of the code that are not directly related to the example ■ That you can repeat a portion of the code	CREATE TABLE ... AS *subquery*; SELECT *col1*, *col2*, ... , *coln* FROM employees;
. . .	Vertical ellipsis points indicate that we have omitted several lines of code not directly related to the example.	SQL> SELECT NAME FROM V$DATAFILE; NAME ---------- /fsl/dbs/tbs_01.dbf /fs1/dbs/tbs_02.dbf . . . /fsl/dbs/tbs_09.dbf 9 rows selected.
Other notation	You must enter symbols other than brackets, braces, vertical bars, and ellipsis points as shown.	acctbal NUMBER(11,2); acct CONSTANT NUMBER(4) := 3;
Italics	Italicized text indicates placeholders or variables for which you must supply particular values.	CONNECT SYSTEM/*system_password* DB_NAME = *database_name*
UPPERCASE	Uppercase typeface indicates elements supplied by the system. We show these terms in uppercase in order to distinguish them from terms you define. Unless terms appear in brackets, enter them in the order and with the spelling shown. However, because these terms are not case sensitive, you can enter them in lowercase.	SELECT last_name, employee_id FROM employees; SELECT * FROM USER_TABLES; DROP TABLE hr.employees;
lowercase	Lowercase typeface indicates programmatic elements that you supply. For example, lowercase indicates names of tables, columns, or files. **Note:** Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown.	SELECT last_name, employee_id FROM employees; sqlplus hr/hr CREATE USER mjones IDENTIFIED BY ty3MU9;

Conventions for Windows Operating Systems

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

Convention	Meaning	Example
Choose Start >	How to start a program.	To start the Database Configuration Assistant, choose Start > Programs > Oracle - *HOME_NAME* > Configuration and Migration Tools > Database Configuration Assistant.

Convention	Meaning	Example
File and directory names	File and directory names are not case sensitive. The following special characters are not allowed: left angle bracket (<), right angle bracket (>), colon (:), double quotation marks ("), slash (/), pipe (|), and dash (-). The special character backslash (\) is treated as an element separator, even when it appears in quotes. If the file name begins with \\, then Windows assumes it uses the Universal Naming Convention.	`c:\winnt"\"system32 is the same as C:\WINNT\SYSTEM32`
`C:\>`	Represents the Windows command prompt of the current hard disk drive. The escape character in a command prompt is the caret (^). Your prompt reflects the subdirectory in which you are working. Referred to as the *command prompt* in this manual.	`C:\oracle\oradata>`
Special characters	The backslash (\) special character is sometimes required as an escape character for the double quotation mark (") special character at the Windows command prompt. Parentheses and the single quotation mark (') do not require an escape character. Refer to your Windows operating system documentation for more information on escape and special characters.	`C:\>exp scott/tiger TABLES=emp QUERY=\"WHERE job='SALESMAN' and sal<1600\" C:\>imp SYSTEM/`*`password`*` FROMUSER=scott TABLES=(emp, dept)`
HOME_NAME	Represents the Oracle home name. The home name can be up to 16 alphanumeric characters. The only special character allowed in the home name is the underscore.	`C:\> net start Oracle`*`HOME_NAME`*`TNSListener`

1 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:

Note: This tutorial assumes you have installed the following components:

- The BPEL Process Manager for Developers installation option on a supported Windows operating system

  This option provides an Oracle JDeveloper 10*g*-based environment for modeling, editing, and designing business processes using BPEL. Oracle BPEL Server (to which to deploy BPEL processes) and Oracle BPEL Console (from which to run BPEL processes) are also installed. See *Oracle BPEL Process Manager Quick Start Guide* or *Oracle Application Server Integration Installation Guide* for installation instructions.

- Patch 1

  See the following URL for instructions on downloading and applying Patch 1:

  `http://www.oracle.com/technology/bpel`

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

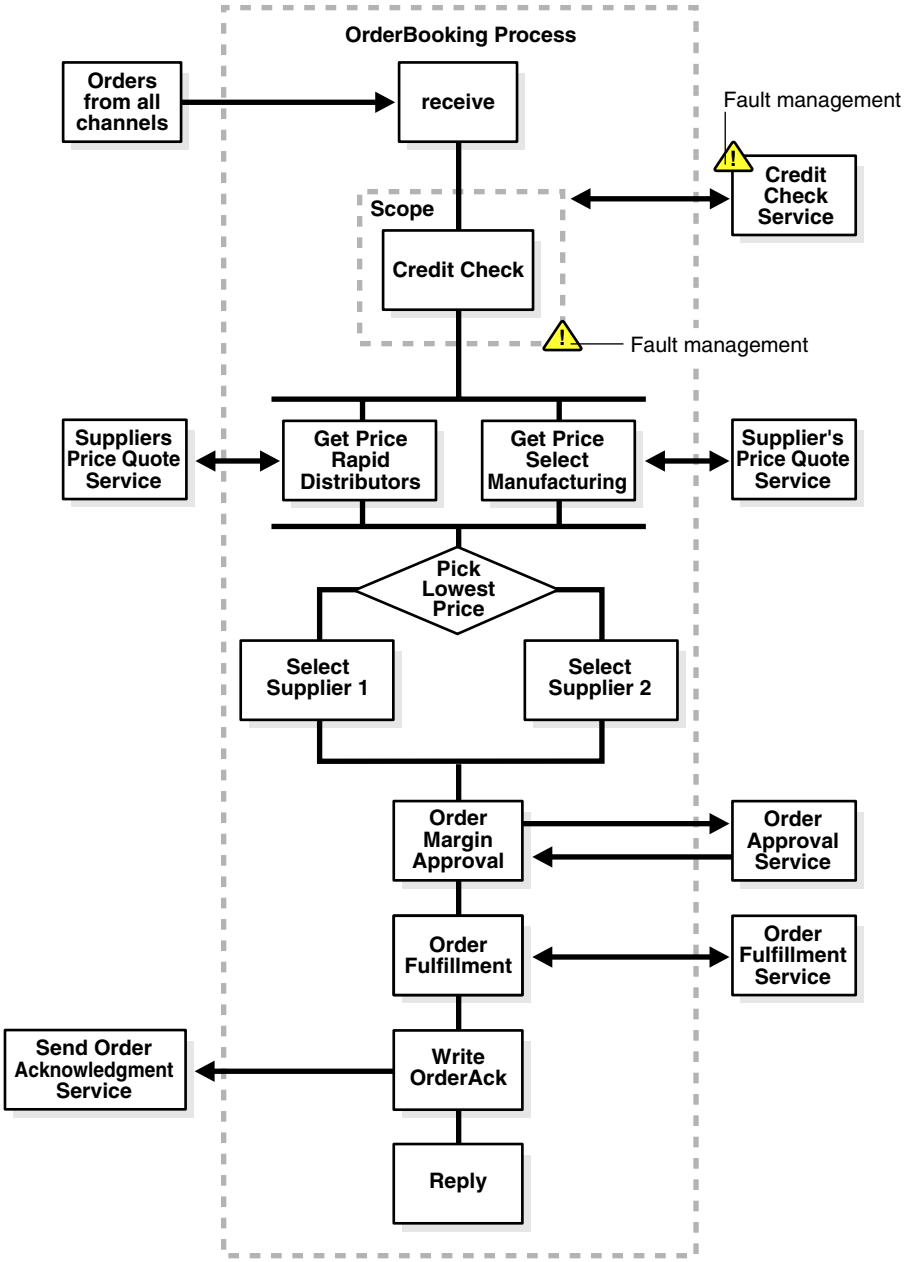
GlobalCompany decides to use Oracle BPEL Process Manager to create an order booking composite application. GlobalCompany uses the Oracle Applications Server platform to enable service-oriented architecture (SOA) in their enterprise.

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 acknowledgement 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:

Oracle_Home\integration\orabpel\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.

Starting Oracle BPEL Process Manager Components

Follow the instructions to start Oracle BPEL Process Manager components.

1. Select **Start** > **All Programs** > **Oracle - *Oracle_Home*** > **Oracle BPEL Process Manager 10.1.2** > **Start BPEL PM Server** to start the Oracle BPEL Server.
2. Select **Start** > **All Programs** > **Oracle - *Oracle_Home*** > **Oracle BPEL Process Manager 10.1.2** > **JDeveloper BPEL Designer** to start the JDeveloper BPEL Designer.
3. Select **Start** > **All Programs** > **Oracle - *Oracle_Home*** > **Oracle BPEL Process Manager 10.1.2** > **Developer Prompt** to open up a command prompt at the *Oracle_Home*\integration\orabpel\samples directory. This enables you to easily access demonstrations and start any required Web services.
4. Select **Start** > **All Programs** > **Oracle - *Oracle_Home*** > **Oracle BPEL Process Manager 10.1.2** > **BPEL Console** to start the Oracle BPEL Console.
5. Select **Start** > **All Programs** > **Oracle - *Oracle_Home*** > **Oracle BPEL Process Manager 10.1.2** > **Getting Started with Samples** for details about BPEL samples and additional tutorials available for use.

6. Select **Start** > **All Programs** > **Oracle -** *Oracle_Home* > **Oracle BPEL Process Manager 10.1.2** > **Sample Worklist Application** to access the login window for the Oracle BPEL Worklist Application.

> **Note:** You *must* use Internet Explorer version 6 with this tutorial.

Connecting to Oracle BPEL Server

When you start JDeveloper BPEL Designer for the first time, a connection named **LocalBPELServer** is automatically created. This connection enables you to deploy your BPEL process from JDeveloper BPEL Designer to Oracle BPEL Server. The **LocalBPELServer** connection is sufficient for completing all phases of this tutorial.

If you want to create additional connections, follow these instructions.

1. Select **Connection Navigator** from the **View** main menu in JDeveloper BPEL Designer.
2. Right click **BPEL Process Manager Server**.
3. Select **New BPEL Process Manager Connection**.
4. Click **Next** on the Welcome page.
5. Provide a name for connecting to the server.
6. Click **Next**.
7. Enter the following details:

Field	Value
Host Name	**localhost** (default value)
Port Number	**9700** (default value)

8. Click **Next**.
9. Test the connection by clicking **Test Connection**. If the connection was successful, the following message appears:

   ```
   Success.
   ```

10. Click **Finish**.

Setting the Hostname in Your JDeveloper BPEL Designer Web Browser Preferences

Add the hostname of your computer to JDeveloper BPEL Designer preference settings. If you do not do this, you can receive parsing errors when selecting a WSDL file on the WSDL Chooser window while creating a partner link. Perform this task even if you do not have a proxy server.

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

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

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

5. Click **OK**.

2

Invoking a Synchronous Service

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

This chapter contains the following topics:

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 Credit Rating service.
- Design the BPEL process to invoke the Credit Rating service.
- Deploy the BPEL process.
- Run the deployed BPEL process from Oracle BPEL Console 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 JDeveloper BPEL Designer

Ensure that Oracle BPEL Server and JDeveloper BPEL Designer are started. See "Starting Oracle BPEL Process Manager Components" on page 1-4 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 10.1.2** > **Developer Prompt** to open up an operating system command prompt at the *Oracle_Home*\integration\orabpel\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:

   ```
   obant
   ```

   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\integration\orabpel\samples\utils\CreditRatingService>ENDLOCAL
   ```

   If you receive an error saying the path was not found, ensure that your environment path is set to the *Oracle_Home*\integration\orabpel\bin directory.

4. Access Oracle BPEL Console using Internet Explorer to see if all required services are running. Select **Start** > **All Programs** > **Oracle -** ***Oracle_Home*** > **Oracle BPEL Process Manager 10.1.2** > **BPEL Console**.

5. Click **Login** with **default** selected from the **Domain** list. The password is **bpel**.

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

   See Also: "Connecting to Oracle BPEL Server" on page 1-5 for instructions on creating a connection to Oracle BPEL Console

Creating a New Workspace and an Order Booking Project

Follow these instructions to create a new workspace and Order Booking project.

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. Return to JDeveloper BPEL Designer.
2. Select **File** > **New** from the main menu.
3. Double-click **Workspace** in the **Items** window to display the Create Workspace window.
4. Enter **OrderBookworkspace** in the **Workspace Name** field and accept the default path in the **Directory Name** field.

5. Deselect the **Add a New Empty Project** check box.
6. Click **OK**.
7. Right-click **OrderBookworkspace** in the **Applications Navigator** section of the designer window. If the **Applications Navigator** section does not currently appear, select **Application Navigator** from the **View** main menu.
8. Select **New Project**.
9. Double-click **BPEL Process Project** in the **Items** window to display the BPEL Process Project window.
10. Enter **OrderBooking** in the **BPEL Process Name** field. All other fields default to the correct values for creating an asynchronous BPEL process.
11. Click **OK**.

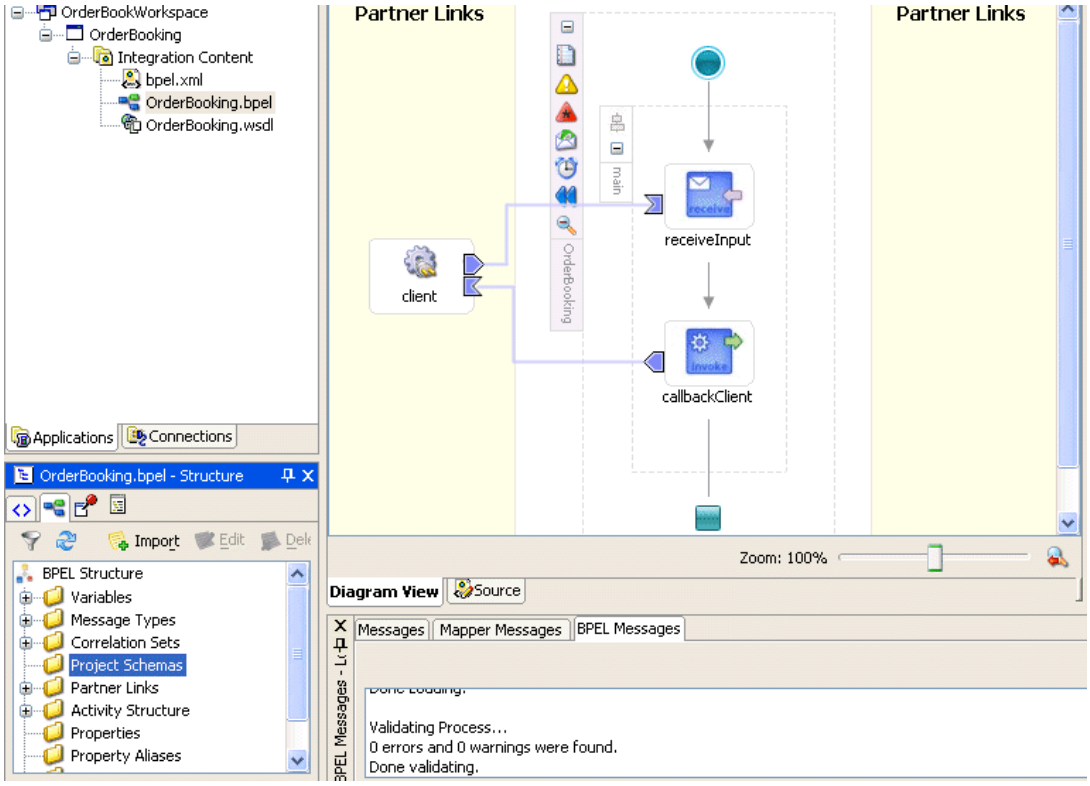
    The **bpel.xml**, **OrderBooking.bpel**, and **OrderBooking.wsdl** files are created.

Importing the Order Booking Schema

Summary: You now import the GlobalCompany schema into your project. This schema defines the structure of the purchase order to submit.

1. Copy `OrderBookingPO.xsd` and `OrderBookingPO_1.xml` from the *`Oracle_Home`*`\integration\orabpel\samples\tutorials\127.OrderBookingTutorial\PracticeFiles` directory to the *`Oracle_Home`*`\integration\jdev\jdev\mywork\OrderBookworkspace\OrderBooking` directory.
2. Double-click **OrderBooking.bpel** in the **Applications Navigator** section of the designer window to display the **Structure** section in the lower left of the window.

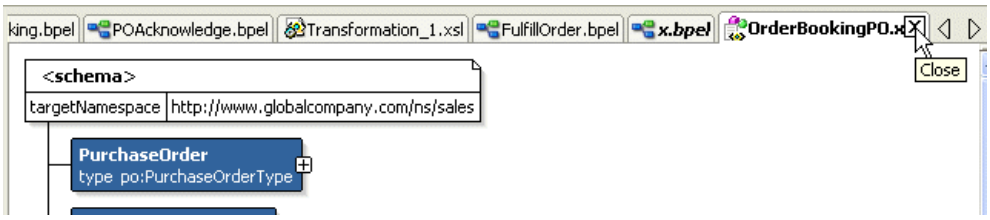
Note: The **Diagram View** tab must be enabled to see the **Structure** section.



3. Select and right-click **Project Schemas** in the **Structure** section.
4. Select **Import Schema**.

   The Import Schema window appears.
5. Click the **flashlight** icon to access the Open window.
6. Select **OrderBookingPO.xsd** from the **OrderBooking** directory in which you placed it in Step 1 on page 2-4 and click **Open**.

   The file is added to the **URL** field of the Import Schema window.
7. Click **OK**.
8. Verify that **OrderBookingPO.xsd** appears under **OrderBooking** > **Web Content** > **Miscellaneous Files** in the **Applications Navigator** section of the designer window. **OrderBookingPO.xsd** also appears under **Project Schemas** in the **Structure** section. If you want, you can double-click **OrderBookingPO.xsd** under **Miscellaneous Files** to view the design and source modes of this file.
9. If you view **OrderBookingPO.xsd**, close the window when done by clicking the **x** button on the **Title** tab.



10. Double-click **OrderBooking.bpel** in the **Applications Navigator** section of the designer window to display the **Structure** section in the lower left section of the designer window.
11. Expand **Message Types** > **Process WSDL - OrderBooking.wsdl** > **OrderBookingRequestMessage** > **payload** in the **Structure** section.
12. Right-click **payload** and select **Edit Message Part** to display the Edit Message Part window.
13. Select **Element** and click the **flashlight** icon to the right of the **Element** field.
14. Expand **Project Schema Files** > **OrderBookingPO.xsd**.
15. Select **PurchaseOrder**.
16. Click **OK** to close the Type Chooser window and the Edit Message Part window.

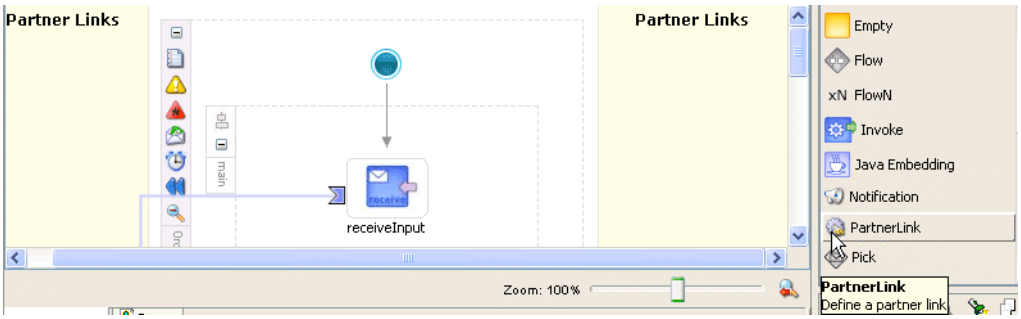
    This defines the input parameter (payload) of the **PurchaseOrder** type.
17. Expand **Message Types** > **Process WSDL - OrderBooking.wsdl** > **OrderBookingResponseMessage** > **payload** in the **Structure** section.
18. Right-click **payload** and select **Edit Message Part** to display the Edit Message Part window.
19. Select **Element** and click the **flashlight** icon to the right of the **Element** field.
20. Expand **Project Schema Files** > **OrderBookingPO.xsd**.
21. Select **PurchaseOrder**.
22. Click **OK** to close the Type Chooser window and the Edit Message Part window.

    This defines the output parameter (payload) of the **PurchaseOrder** type.
23. Select **Save** from the **File** main menu.

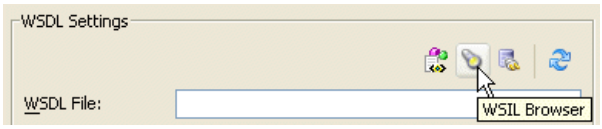
Creating a Credit Rating Service Partner Link

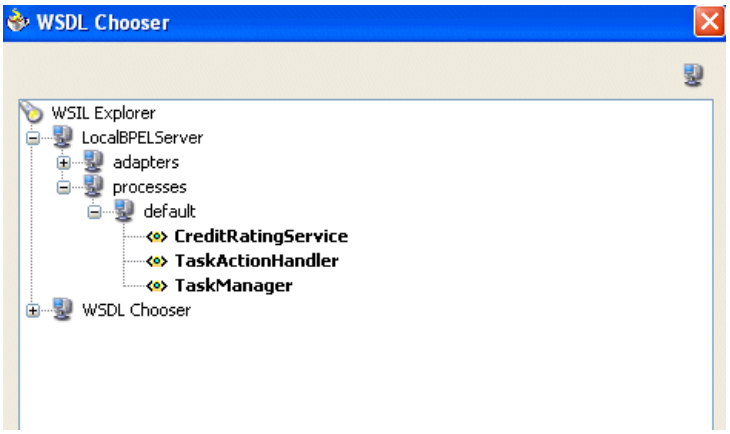
Summary: You now create a partner link for the Credit Rating service. This is the service you started when you ran `obant` in "Starting and Testing Your Services" on page 2-2. A partner link enables you to define the external services with which the **OrderBooking** BPEL process is to interact. A partner 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 **Process Activities** is selected in the drop-down list of the **Component Palette** section in the upper right section of JDeveloper BPEL Designer.
2. Drag and drop a **PartnerLink** activity onto the right side of the designer window anywhere beneath the header **Partner Links**. Note that both sides of the designer window have sections in which partner links can be placed.



3. 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 **WSIL Browser**) to access the WSDL Chooser 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	Access this URL by clicking the **WSIL Browser flashlight** icon and expanding and selecting **LocalBPELServer** > **processes** > **default** > **CreditRatingService**. **http://localhost:9700/orabpel/default/CreditRatingService/CreditRatingService?wsdl** **See Also:** "Setting the Hostname in Your JDeveloper BPEL Designer Web Browser Preferences" on page 1-5 if you receive a parsing error when attempting to add a WSDL file in the WSDL Chooser window.
Partner Link Type	**CreditRatingService**
My Role	Leave unspecified since this is a synchronous service.
Partner Role	**CreditRatingServiceProvider**

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

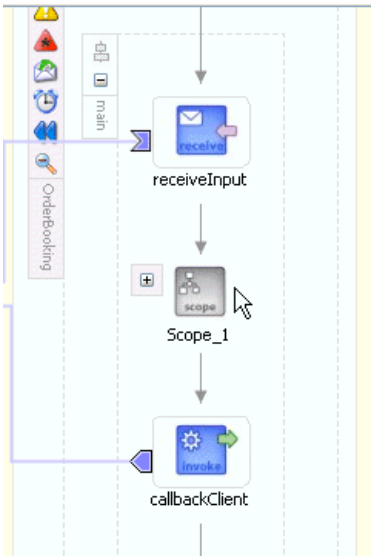
Creating Scope and Invoke Activities

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 3 of "Running the Order Booking Process" on page 2-16.

Note: As you create and open activities such as **Scope**, **Invoke**, and **Assign** for the first time, the message `Invalid Settings` can appear at the top. This is because you have not yet entered details. You can ignore this message. After you enter and apply your details, the message disappears.

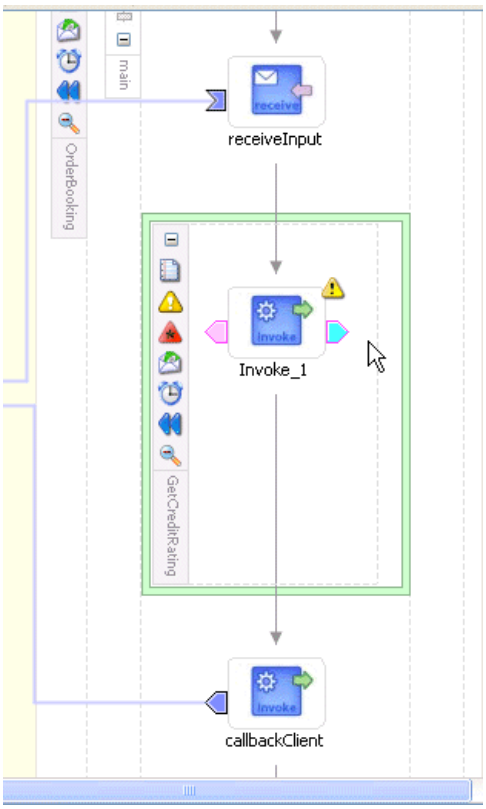
---

1. 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**.

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



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

Field	Value
Name	**invokeCR**
Partner Link	**CreditRatingService**

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

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

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

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

11. Click the first icon to the right of the **Output Variable** field.
12. Click **OK** on the Create Variable window that appears.

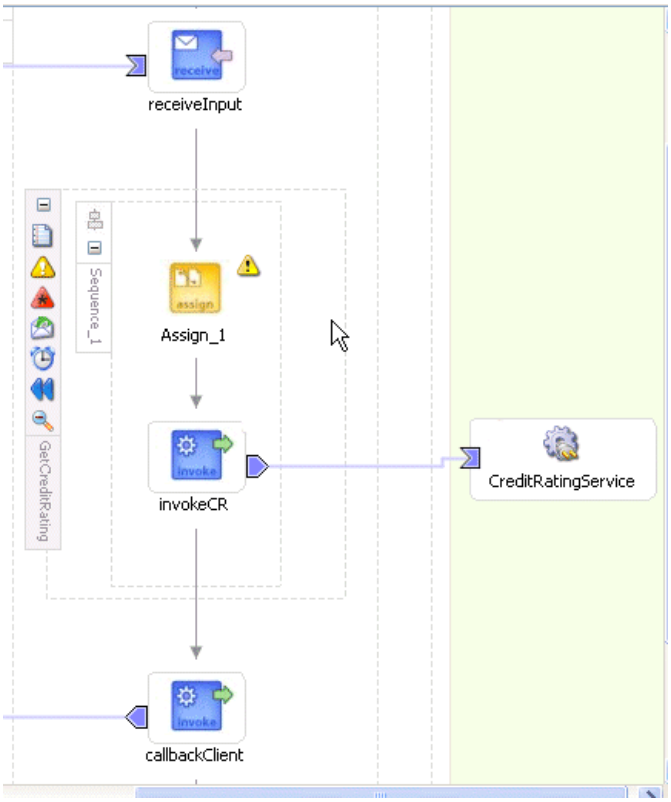
    A variable named **invokeCR_process_OutputVariable** is automatically created in the **Output Variable** field. This variable is automatically assigned a message type of **CreditRatingServiceResponseMessage**.

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

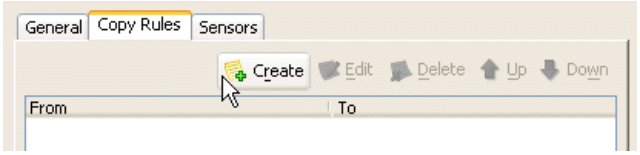
Creating an Initial Assign Activity to Send Request Data to the Credit Rating Service

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.



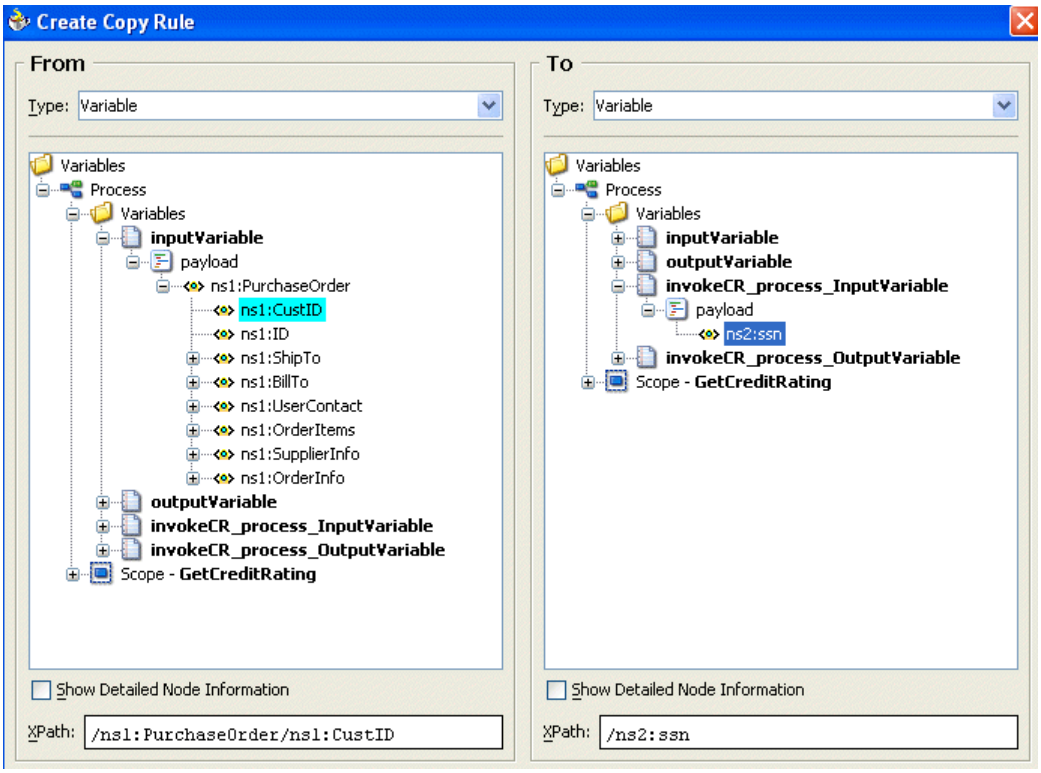
2. Double-click the **assign** icon to display the Assign window.
3. Click the **General** tab.
4. Enter **assignCR** in the **Name** field.
5. Click **Apply**.
6. Click the **Copy Rules** tab.
7. Click **Create** to display the Create Copy Rule window.



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.
To	
▪ **Type**	**Variable**
▪ **Variables**	Expand and select **Variables** > **invokeCR_process_InputVariable** > **payload** > **ns2:ssn**

The Create Copy Rule window appears as follows:

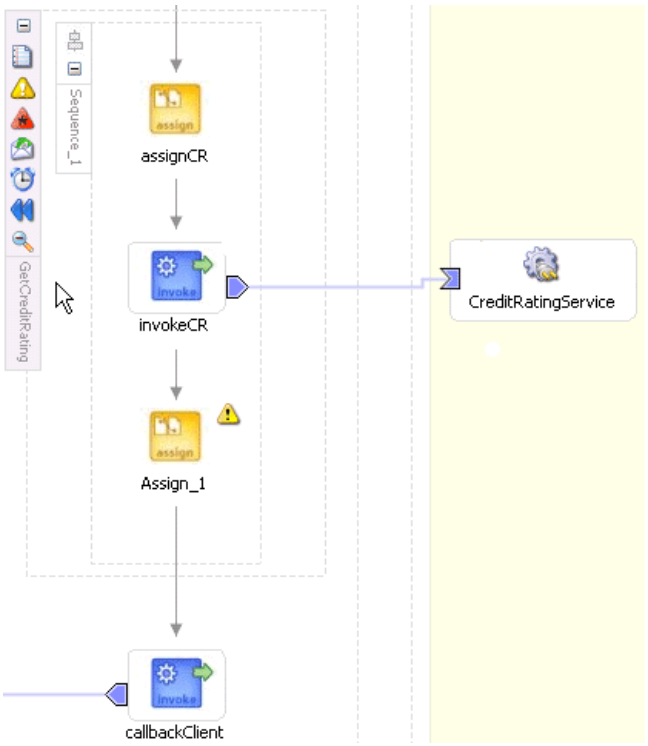


9. Click **OK** to close the Create Copy Rule window and the Assign window.
10. Select **Save** from the **File** main menu.

Creating a Second Assign Activity to Receive Response Data 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.



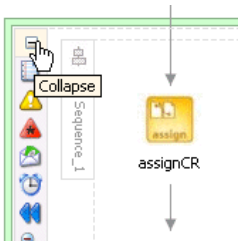
2. Double-click the **assign** icon to display the Assign window.
3. Click the **General** tab.
4. Enter **assignRating** in the **Name** field.
5. Click **Apply**.
6. Click the **Copy Rules** tab.
7. Click **Create** to display the Create Copy Rule window.
8. Enter the following details:

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.

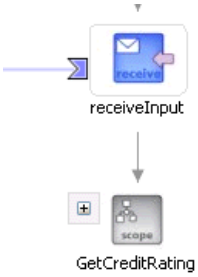
Field	Value
From	
▪ **Type**	**Expression**
▪ **Expression**	**concat('Good credit, Rating = ', bpws:getVariableData('invokeCR_process_OutputVariable','payload','/ns2:rating'))** **Note**: Press **Ctrl** and then the space bar to display a list for selecting (double-clicking) this syntax. The namespace number values (for example, **ns1**, **ns2**) can vary. Use the namespace values that automatically appear.

Field	Value
To	
▪ **Type**	**Variable**
▪ **Variables**	Expand and select **Variables** > **inputVariable** > **payload** > **ns1:PurchaseOrder** > **ns1:OrderInfo** > **ns1:OrderComments**

9. Click **OK** to close the Create Copy Rule window and the Assign window.
10. Click the **-** sign to close the **GetCreditRating Scope** activity.



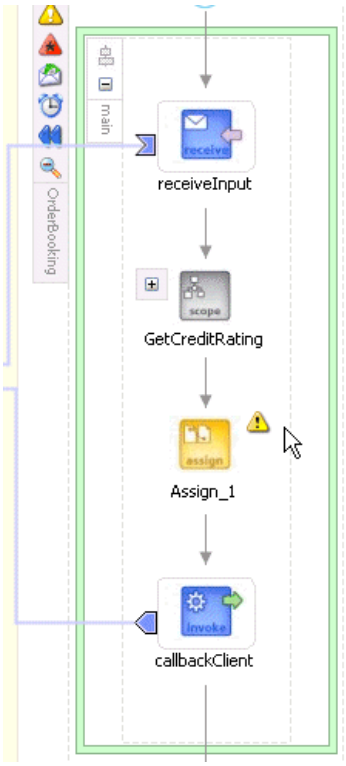
When closed, the **GetCreditRating Scope** activity looks as follows:



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.

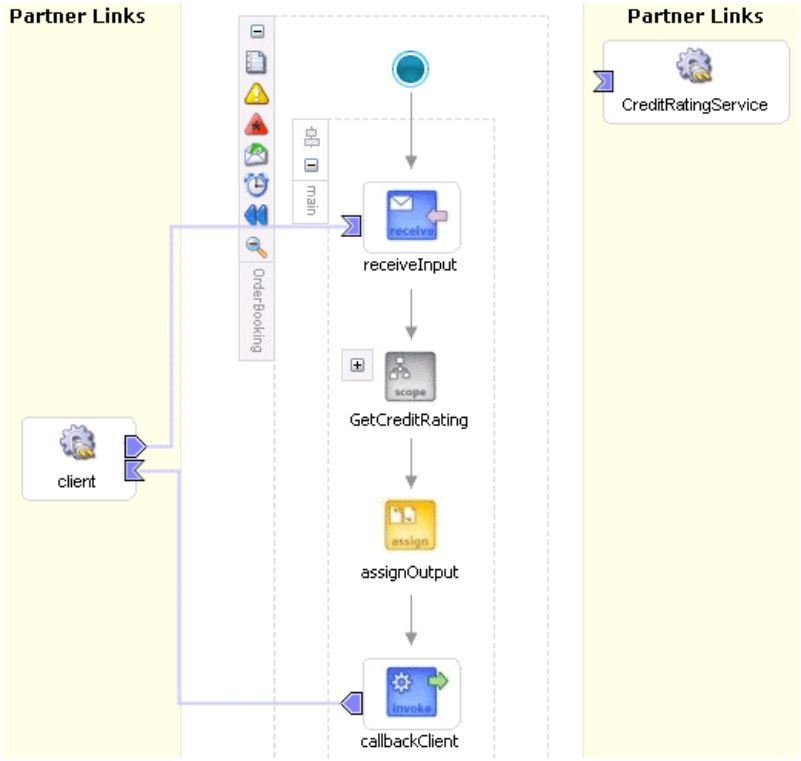


2. Double-click the **assign** icon to display the Assign window.
3. Click the **General** tab.
4. Enter **assignOutput** in the **Name** field.
5. Click **Apply**.
6. Click the **Copy Rules** tab.
7. Click **Create** to display the Create Copy Rule window.
8. Enter the following details to create a 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 Rule window and the Assign window.
10. Select **Save** from the **File** main menu.

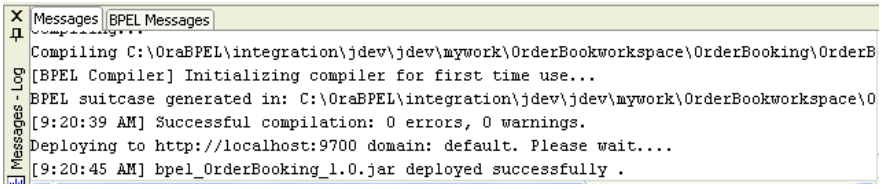
    When complete, the designer window looks as follows:



Validating, Compiling, and Deploying the Order Booking Process

1. Go to the **Applications Navigator** section.
2. Right-click **OrderBooking**.
3. Select **Deploy** > **LocalBPELServer** > **Deploy to default domain**.
4. Enter the domain password of **bpel** when prompted.

   This compiles the BPEL process. Review the bottom of the window for any errors. If there are no errors, deployment was successful.



5. If there are errors, click **BPEL Validation Errors** to display details about the type and location of the error.
6. 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. Log into Oracle BPEL Console using Internet Explorer or refresh the page if it is already open. Select **Start** > **All Programs** > **Oracle -** *Oracle_Home* > **Oracle BPEL Process Manager 10.1.2** > **BPEL Console**.

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

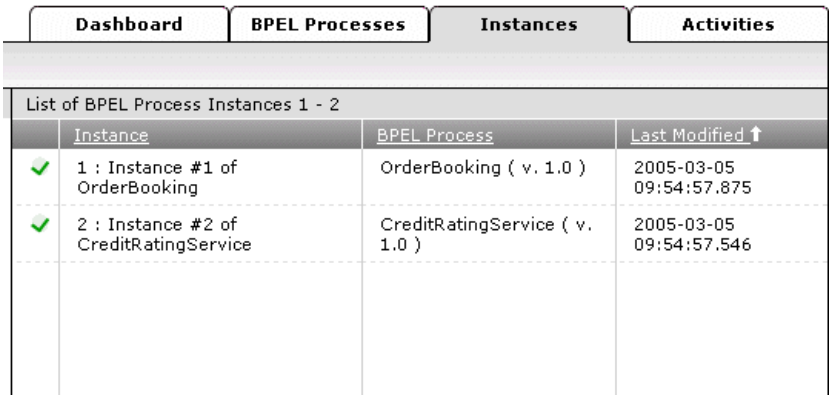
2. Click **OrderBooking** in the **Deployed BPEL Processes** list.
3. Enter a nine-digit integer value that does not begin with zero in the **CustID** field of the HTML Form and click **Post XML Message**. Ignore the other fields.

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

   ```
   Test Instance Initiated

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

4. Click the **Instances** tab at the top.
5. 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.

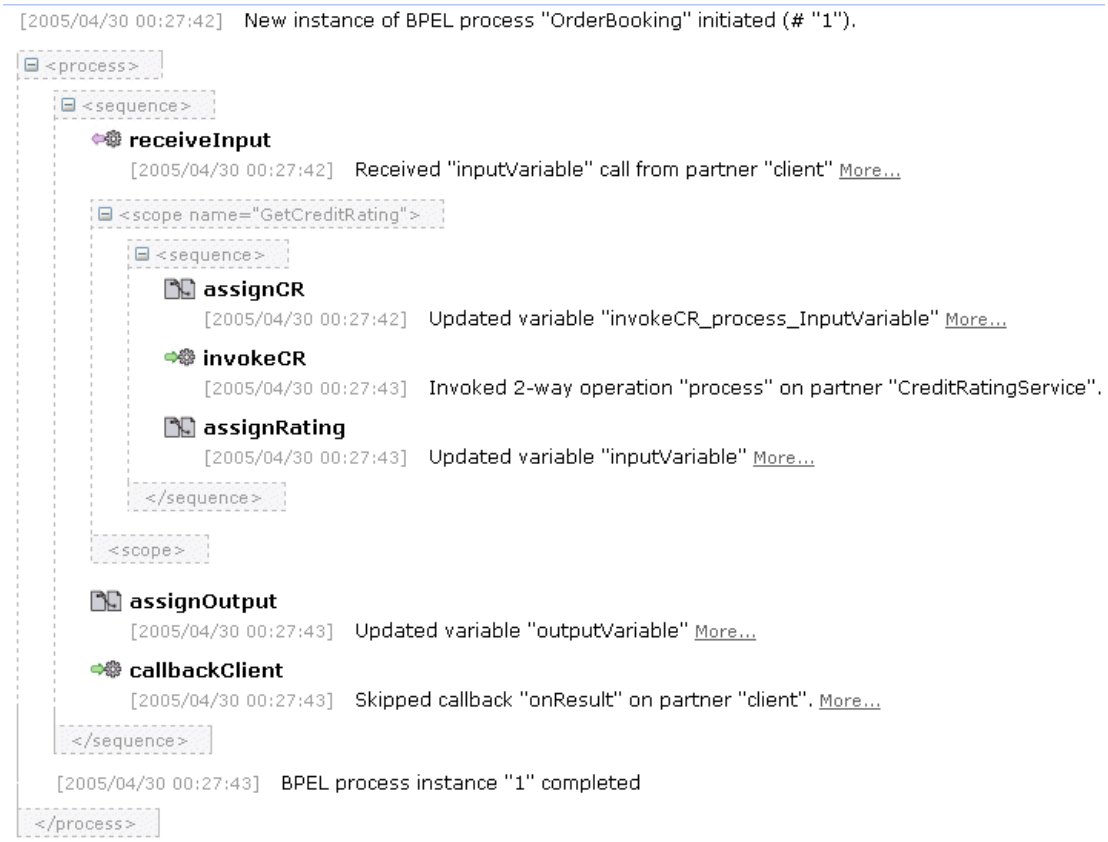


6. Click the **OrderBooking** instance.

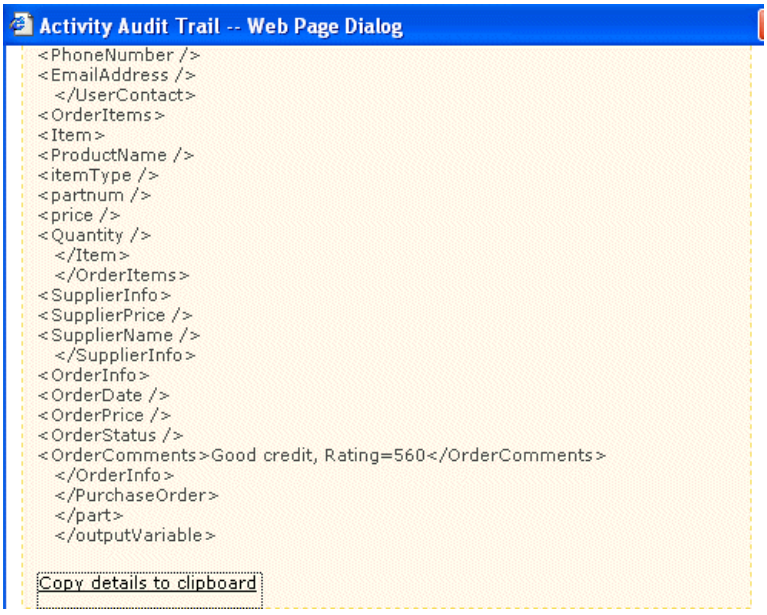
   A message appears stating the following:

   ```
   The state of this instance is Completed
   ```

7. Click the **Audit** link to follow the instance execution process. The **Audit** link displays the following details:



8. Click the **Flow** link.
9. Click **callbackClient** at the bottom of the flow to see the results.



10. 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 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 Console.
- Ran the BPEL process from Oracle BPEL Console 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.

3

Invoking an Asynchronous Service

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

This chapter contains the following topics:

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 Console 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 10.1.2** > **Developer Prompt** to open up an operating system command prompt at the `Oracle_Home\integration\orabpel\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:

   ```
   obant
   ```

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 Console using Internet Explorer to see if all required services are running by selecting **Start** > **All Programs** > **Oracle -** ***Oracle_Home*** > **Oracle BPEL Process Manager 10.1.2** > **BPEL Console**.
5. Click **Login** with **default** selected from the **Domain** list. The password is **bpel**.

   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 JDeveloper BPEL Designer.
2. Ensure that **Process Activities** is selected in the drop-down list of the **Component Palette** section.
3. Drag and drop a **PartnerLink** activity onto the *right* side of the designer window.
4. Enter the following values to create a partner link for the Rapid Distributors service:

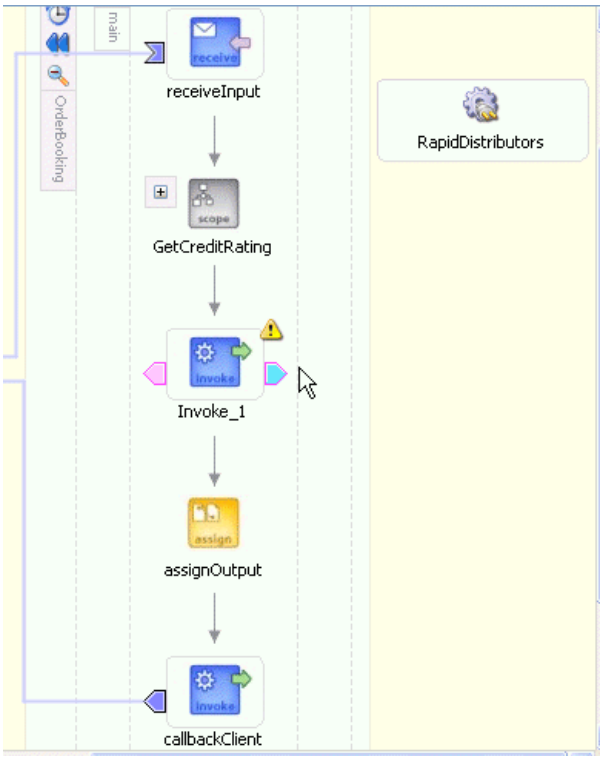
Note: For the **WSDL File** field below, click the **flashlight** (the second icon from the left named **WSIL Browser**) to access the WSDL Chooser window for automatically selecting the Rapid Distributors service deployed in "Starting and Testing Your Services" on page 3-2.

Field	Value
Name	**RapidDistributors**
WSDL File	Access this URL by clicking the **WSIL Browser flashlight** icon and expanding and selecting **LocalBPELServer** > **processes** > **default** > **RapidDistributors**. **http://localhost:9700/orabpel/default/RapidDistributors/RapidDistributors?wsdl** **See Also:** "Setting the Hostname in Your JDeveloper BPEL Designer Web Browser Preferences" on page 1-5 if you receive a parsing error when attempting to add a WSDL file in the WSDL Chooser window.
Partner Link Type	**RapidDistributors**
My Role	**RapidDistributorsRequester**
Partner Role	**RapidDistributorsProvider**

5. Click **OK**.
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 to below the **GetCreditRating Scope** activity.



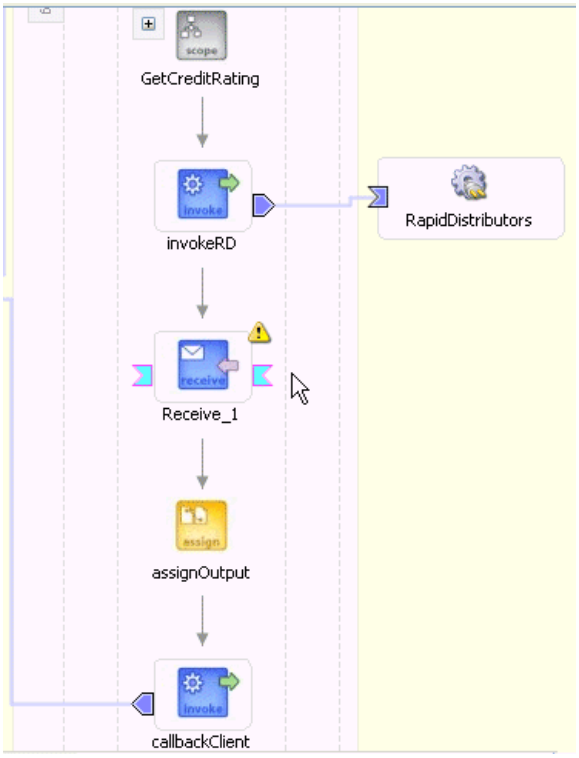
2. Double-click the **Invoke** icon to display the Invoke window.
3. Enter the following details:

Field	Value
Name	**invokeRD**
Partner Link	**RapidDistributors**

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 **invokeRD_initiate_InputVariable** is automatically created in the **Input Variable** field. This variable is automatically assigned a message type of **RapidDistributorsRequestMessage**.
6. Click **OK**.
7. Drag and drop a **Receive** activity below the **invokeRD Invoke** activity you just created.



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

Field	Value
Name	**receiveRD**
Partner Link	**RapidDistributors**

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

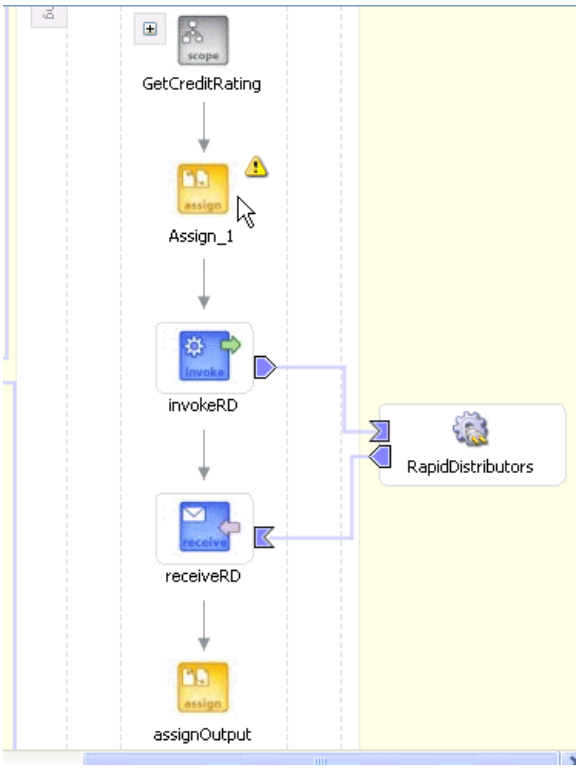
10. Click the first icon to the right of the **Variable** field.
11. Click **OK** on the Create Variable window that appears.

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

12. Click **OK**.
13. 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.

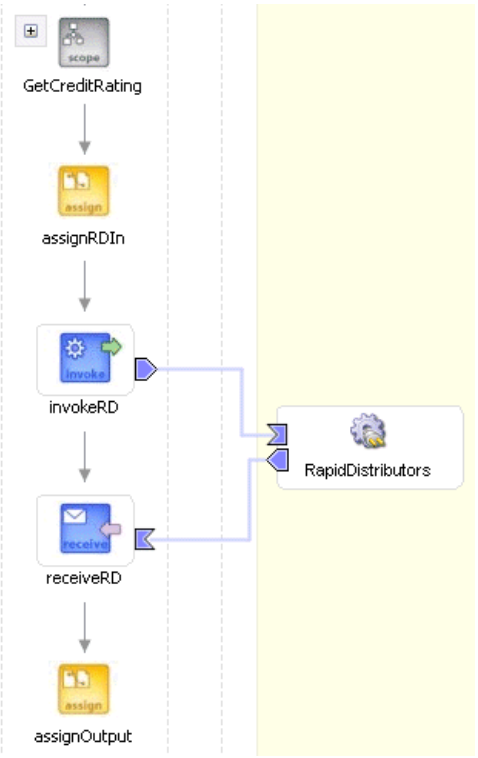


2. Double-click the **assign** icon to display the Assign window.
3. Click the **General** tab.
4. Enter **assignRDIn** in the **Name** field.
5. Click **Apply**.
6. Click the **Copy Rules** tab.
7. Click **Create** to display the Create Copy Rule window.
8. Enter the following details:

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 Rule 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

1. Go to the **Applications Navigator** section.
2. Right-click **OrderBooking**.
3. Select **Deploy** > **LocalBPELServer** > **Deploy to default domain**.
4. Enter the domain password of **bpel** when prompted.

   The Deploy Properties window appears.
5. Increment the version number of the project (for example, enter **1.1**) and click **OK**. When you access Oracle BPEL Console, 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 2-15.

   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 5 on page 2-15.

Running the Order Booking Process

1. Log into Oracle BPEL Console using Internet Explorer by selecting **Start** > **All Programs** > **Oracle - *Oracle_Home*** > **Oracle BPEL Process Manager 10.1.2** > **BPEL Console** (or refresh the page if it is already open).

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

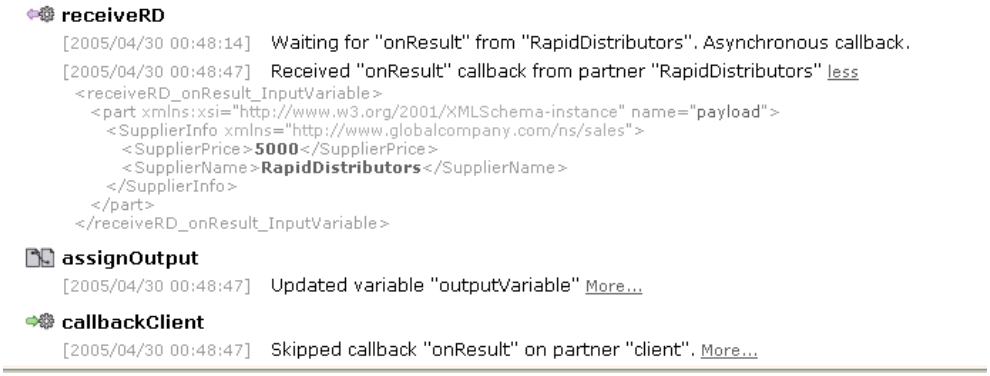
2. Click the **OrderBooking** version you increased in Step 5 on page 3-7 (for this example, **(v. 1.1)**) in the **Deployed BPEL Processes** list.
3. Enter details in all fields of the HTML Form. You can also use a text editor to copy and paste these details from the contents of the `OrderBookingPO_1.xml` file you copied into the `OrderBooking` directory in Step 1 on page 2-4:

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

   a. Select **XML Source** from the **Initiating a test instance** list.
   b. Copy and paste the contents of `OrderBookingPO_1.xml` into the field that appears.
4. Click **Post XML Message**.

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

   ```
   Test Instance Initiated
   ```
5. Click the **Instances** tab at the top.
6. 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.
7. Click the instances.
8. Click the **Audit** and **Flow** links to follow the instance execution process.
9. Click **More** to the right of **receiveRD** under the **Audit** link for the **OrderBooking** instance to see the returned supplier price (returned in the variable **receiveRD_onResult_InputVariable**).



10. 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 Console 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.

4

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

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 Console 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

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 to see if it is deployed:

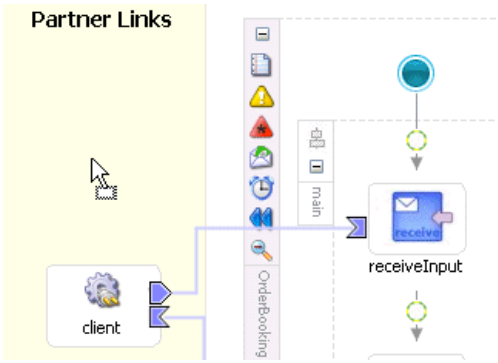
   ```
   http://localhost:9700/SelectManufacturingUI
   ```

   The Order List to Review and Quote Price window appears.

Creating a Select Manufacturing Partner Link

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

1. Return to JDeveloper BPEL Designer.
2. Ensure that **Process Activities** is selected in the drop-down list of the **Component Palette** section.
3. Drag and drop a **PartnerLink** activity onto the *left* side of the designer window (below the **Partner Links** 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 **WSIL Browser**) to access the WSDL Chooser window for automatically selecting the Select Manufacturing service deployed in "Starting and Testing Your Services" on page 3-2.

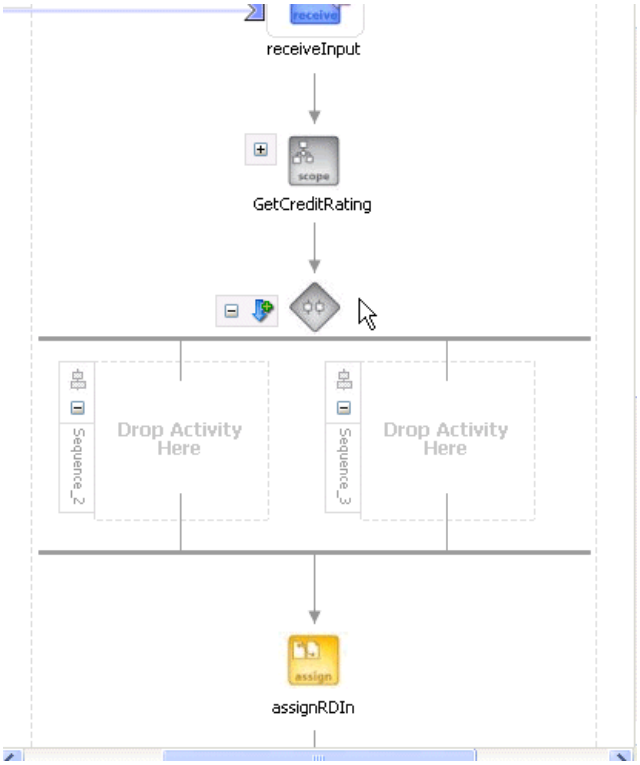
Field	Value
Name	**SelectManufacturing**
WSDL File	Access this URL by clicking the **WSIL Browser flashlight** icon and expanding and selecting **LocalBPELServer** > **processes** > **default** > **SelectManufacturing**. **http://localhost:9700/orabpel/default/SelectManufacturing/SelectManufacturing?wsdl** **See Also:** "Setting the Hostname in Your JDeveloper BPEL Designer Web Browser Preferences" on page 1-5 if you receive a parsing error when attempting to add a WSDL file in the WSDL Chooser window.
Partner Link Type	**SelectManufacturing**
My Role	**SelectManufacturingRequester**
Partner Role	**SelectManufacturingProvider**

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

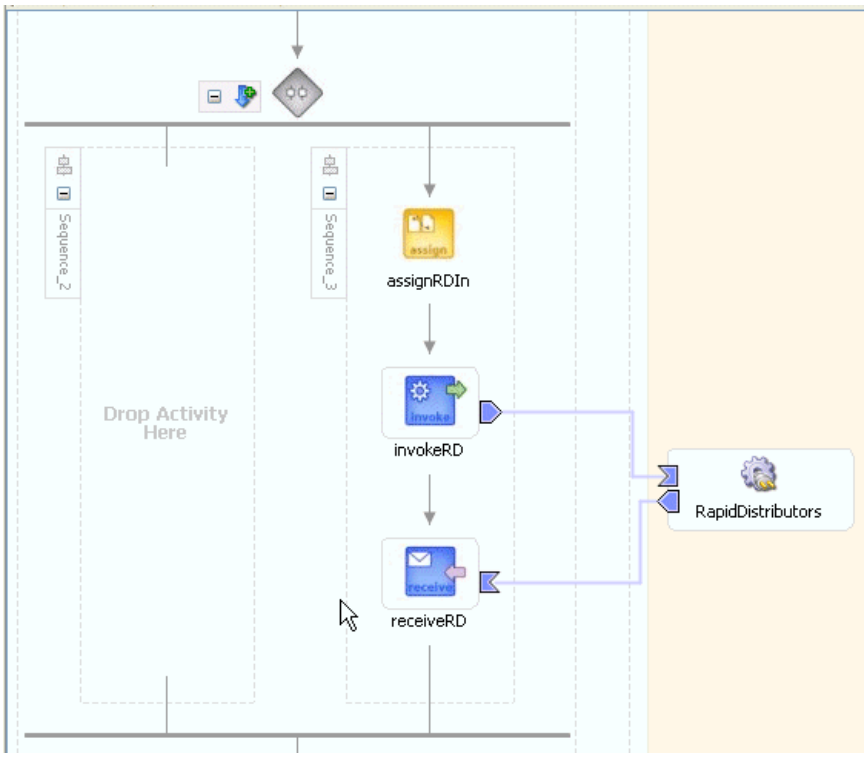
Creating a Flow Activity

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. Drag and drop a **Flow** activity from the **Component Palette** section to below the **GetCreditRating Scope** activity.



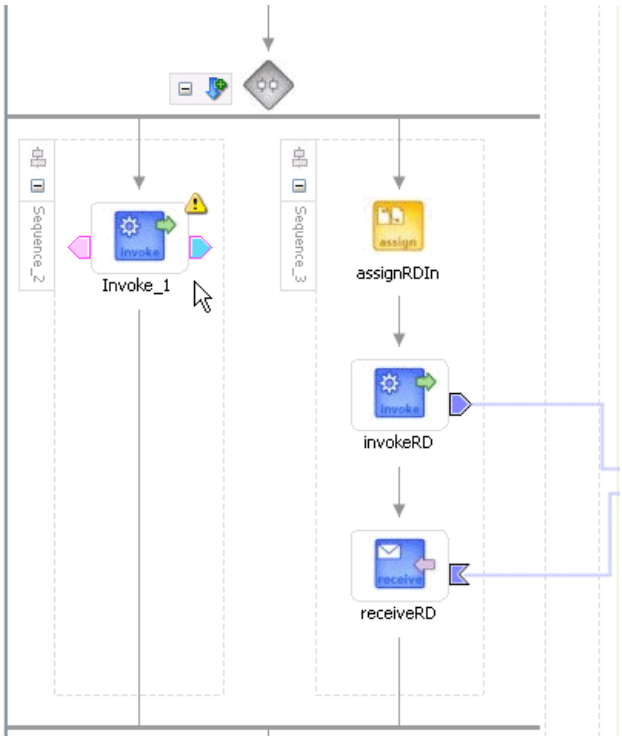
2. Double-click the diamond to display the Flow window.
3. Enter **GetPriceQuote**.
4. Click **OK**.
5. Drag and drop the following activities that you created in Chapter 3, "Invoking an Asynchronous Service" into the *right* panel of the **GetPriceQuote Flow** activity. Drag each activity one at a time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



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 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**

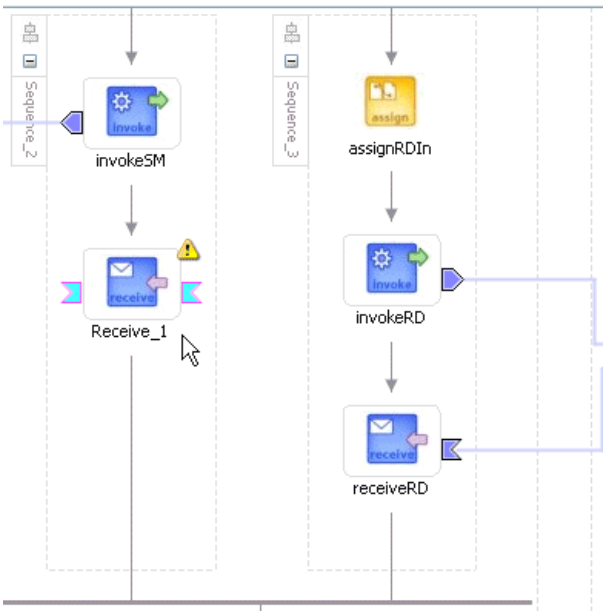
The **Operation** (**initiate**) field is automatically filled in.

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.

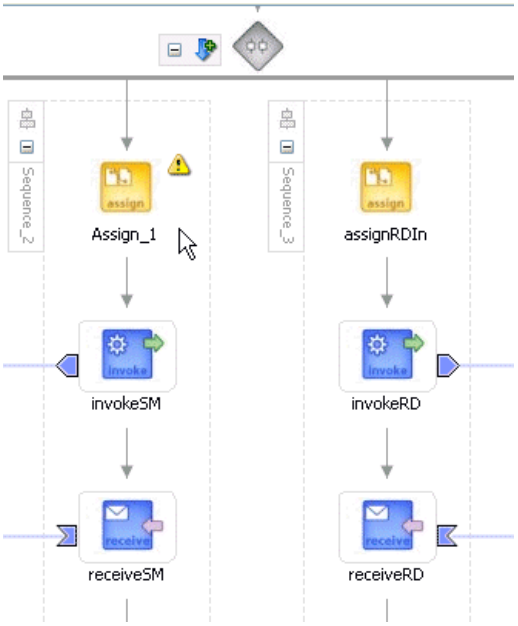
10. Click the first icon to the right of the **Variable** field. This is the automatic variable creation icon.
11. Click **OK** on the Create Variable window that appears.

    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.

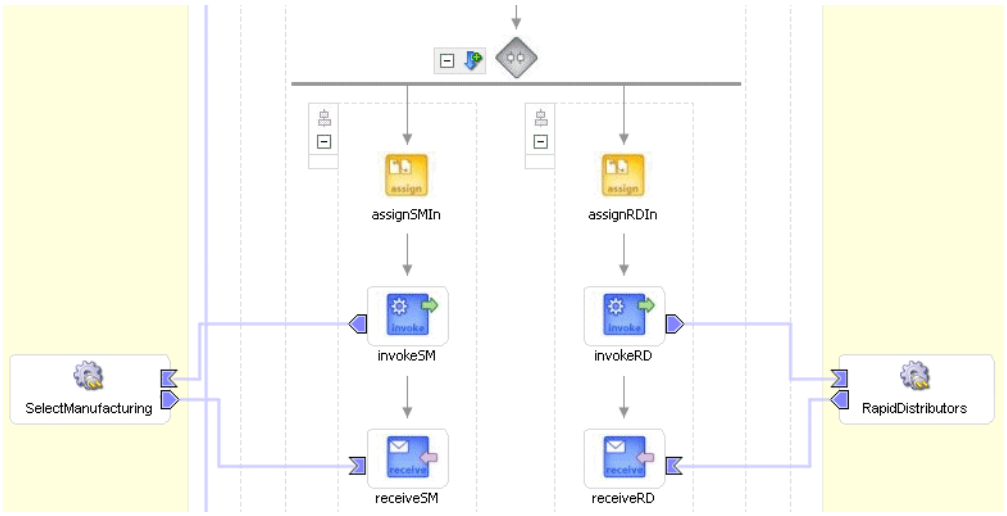


2. Double-click the **assign** icon to display the Assign window.
3. Click the **General** tab.
4. Enter **assignSMIn** in the **Name** field.
5. Click **Apply**.
6. Click the **Copy Rules** tab.
7. Click **Create** to display the Create Copy Rule window.
8. Enter the following details:

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**

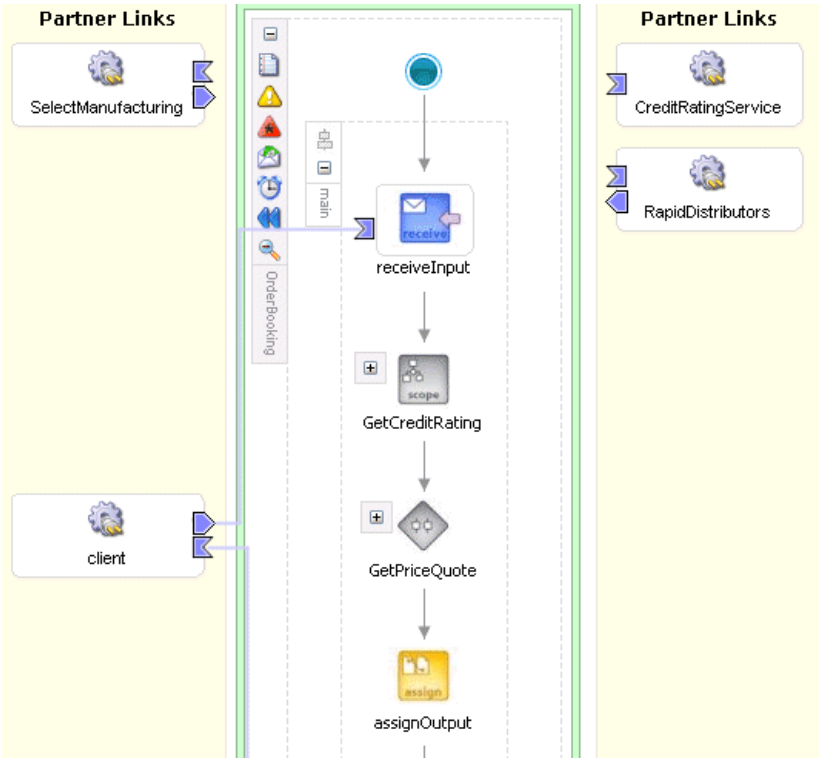
9. Click **OK** to close the Create Copy Rule window and the Assign window.

   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 **Applications Navigator** section.
2. Right-click **OrderBooking**.
3. Select **Deploy** > **LocalBPELServer** > **Deploy to default domain**.
4. Enter the domain password of **bpel** when prompted.
5. Increment the version number again of the project when prompted and click **OK** (for example, enter **1.2**).

   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 5 on page 2-15.

Running the Order Booking Process

1. Log into Oracle BPEL Console using Internet Explorer by selecting **Start** > **All Programs** > **Oracle - *Oracle_Home*** > **Oracle BPEL Process Manager 10.1.2** > **BPEL Console** (or refresh the page if it is already open).

   The **Dashboard** tab of Oracle BPEL Console appears.
2. Click the **OrderBooking** version you increased in Step 5 on page 4-9 in the **Deployed BPEL Processes** list.
3. 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. You must use a text editor to perform this task.

a. Select **XML Source** from the **Initiating a test instance** list.

b. Copy and paste the contents of `OrderBookingPO_1.xml` into the field that appears.

4. Click **Post XML Message**.

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

   ```
   Test Instance Initiated
   ```

5. Click the **Instances** tab at the top.

6. Verify that the **OrderBooking**, **CreditRatingService**, **TaskManager**, **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. This is defined in the **TaskManager** service, which is also awaiting completion.

7. Click the **OrderBooking** instance you increased in Step 5 on page 4-9 in the **Instance** list.

   A message appears stating the following:

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

8. Click the **Audit** link to follow the instance execution process. Note that the **SelectManufacturing** service is pending. The **RapidDistributors** service has completed.

9. 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).

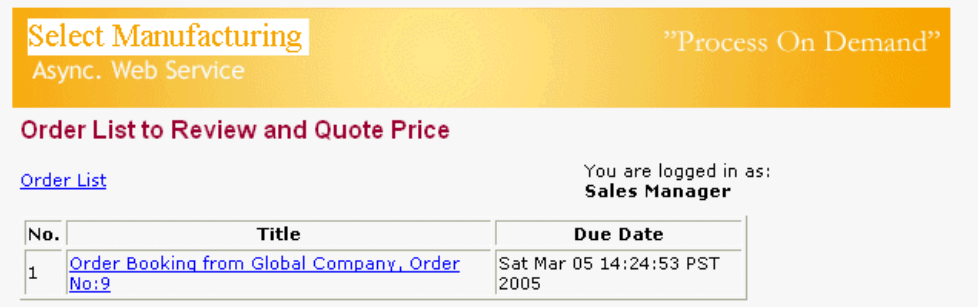
To access this part of the activity, present it to the user for pricing, and accept the user data for task completion and process continuation, you must invoke the task from your browser.

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.

10. Access the prebuilt JSP Select Manufacturing application using Internet Explorer.

    ```
    http://localhost:9700/SelectManufacturingUI
    ```

11. Click the order under **Title**.



12. 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.
13. Return to Oracle BPEL Console.
14. Click the **Instances** tab at the top.
15. Click the **OrderBooking** instance.

    A message appears stating the following:

    ```
    The state of this instance is Completed
    ```

16. 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 Console 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.

5

Adding Conditional Branching Logic

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

This chapter contains the following topics:

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 Console 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

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

```
Is the Rapid Distributors price less than the Select Manufactuing price?
```

7. 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')
```

8. Click **OK**.
9. 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 Rules** tab.
7. Click **Create** to display the Create Copy Rule 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.

Field	Value
To	
▪ **Type**	**Variable**
▪ **Variables**	Expand and select **Variables** > **inputVariable** > **payload** > **ns1:PurchaseOrder** > **ns1:OrderInfo** > **ns1:OrderComments**

9. Click **OK** to close the Create Copy Rule window.
10. Click the **Copy Rules** tab.
11. Click **Create** to display the Create Copy Rule 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 Rule window and the Assign window.
14. Select **Save** from the **File** main menu.

Creating an Assign Activity Under the Otherwise Block for the Select Manufacturing Service

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 Rules** tab.
7. Click **Create** to display the Create Copy Rule 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.

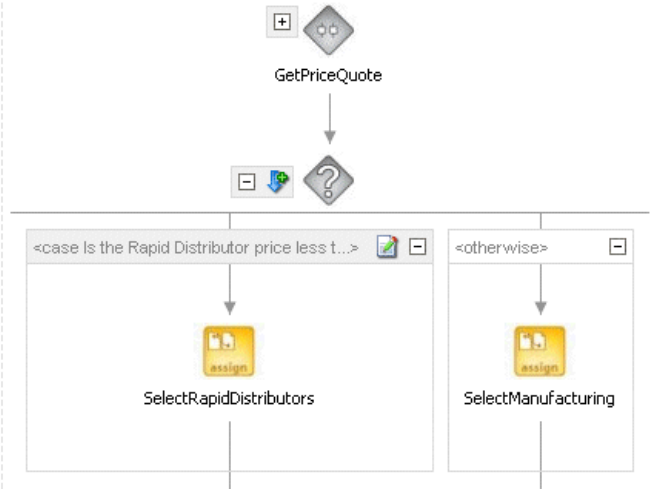
Field	Value
From	
▪ **Type**	**Expression**
▪ **Expression**	**concat(bpws:getVariableData('inputVariable','payload','/ns1:PurchaseOrder/ns1:OrderInfo/ns1:OrderComments'), ' - Selected: SelectManufacturing')** **Note**: Press **Ctrl** and then the space bar to display a list for selecting (double-clicking) 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 Rule window.
10. Click the **Copy Rules** tab.
11. Click **Create** to display the Create Copy Rule 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**

13. Click **OK** to close the Create Copy Rule window and the Assign window.

    When complete, the designer window looks as follows:



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

Validating, Compiling, and Deploying the Order Booking Process

1. Go to the **Applications Navigator** section.
2. Right-click **OrderBooking**.
3. Select **Deploy** > **LocalBPELServer** > **Deploy to default domain**.
4. Enter the domain password of **bpel** when prompted.
5. Increment the version number of the project when prompted and click **OK** (for example, enter **1.3**).

   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 5 on page 2-15.

Running the Order Booking Process

1. Log into Oracle BPEL Console using Internet Explorer by selecting **Start** > **All Programs** > **Oracle - *Oracle_Home*** > **Oracle BPEL Process Manager 10.1.2** > **BPEL Console** (or refresh the page if it is already open).

   The **Dashboard** tab of Oracle BPEL Console appears.
2. Start the **OrderBooking** instance version you increased in Step 5 on page 5-6 in one of the following ways:
   - Open the **CreateOrderBookingUI** application at `http://localhost:9700/CreateOrderBookingUI` and provide inputs to initiate a process.
   - 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 select **XML Source** from the **Initiating a test instance** list. 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.

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

```
Test Instance Initiated
```

3. Complete the manual price quote task for **SelectManufacturing** by using the graphical user interface JSP page, as described in Step 10 on page 4-11.
4. Return to Oracle BPEL Console.
5. Click the **Instances** tab at the top.
6. Verify that the **OrderBooking**, **CreditRatingService**, **TaskManager**, **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.
7. Click the instances.
8. Click the **Audit** link to follow the instance execution process.
9. Click the **More...** link for each sequence to view the activity details.
10. 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 Console 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.

6

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

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 Console 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.

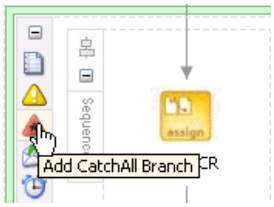
Creating Fault Handling and Exception Management

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



   A new box displays to the right with the words **Drop Activity Here**.
3. Drag and drop an **Assign** activity from the **Component Palette** section into the **Catch** branch you just created.
4. Double-click the **assign** icon to display the Assign window.
5. Click the **General** tab.
6. Enter **assignZeroCR** in the **Name** field.
7. Click **Apply**.
8. Click the **Copy Rules** tab.
9. Click **Create** to display the Create Copy Rule window. You now create two fault handling rules.
10. Enter the following values:

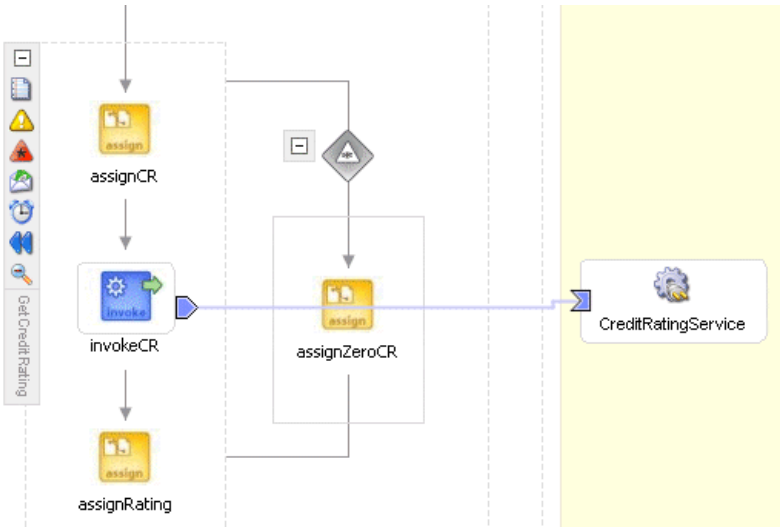
Field	Value
From	
▪ **Type**	**Expression**
▪ **Expression**	**'0'**
To	
▪ **Type**	**Variable**
▪ **Variables**	Expand and select **Variables > invokeCR_process_OutputVariable** > **payload** > **ns1:rating** **Note:** The namespace number values (for example, **ns1**, **ns2**) can vary. Use the namespace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

11. Click **OK**.
12. Click the **Copy Rules** tab.
13. Click **Create** to again display the Create Copy Rule window.
14. Enter the following details to create a second statement:

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

15. Click **OK** to close the Create Copy Rule window and the Assign window.

    When complete, the designer window looks as follows:



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

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

Validating, Compiling, and Deploying the Order Booking Process

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

   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 5 on page 2-15.

Running the Order Booking Process

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

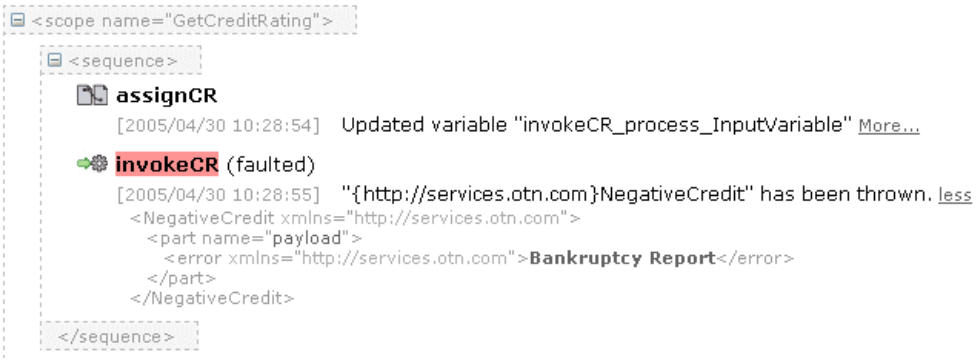
1. Log into Oracle BPEL Console using Internet Explorer by selecting **Start** > **All Programs** > **Oracle - *Oracle_Home*** > **Oracle BPEL Process Manager 10.1.2** > **BPEL Console** (or refresh the page if it is already open).

   The **Dashboard** tab of Oracle BPEL Console appears.
2. Start the **OrderBooking** instance version you increased in Step 5 on page 6-4 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:9700/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
   ```
3. Complete the manual price quote task for **SelectManufacturing** by using the graphical user interface JSP page, as described in Step 10 on page 4-11.
4. Return to Oracle BPEL Console.
5. Click the **Instances** tab at the top.
6. 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.
7. Click the **OrderBooking** instance.
8. Click the **Audit** link to follow the instance execution process.
9. 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.



10. 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 Console 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.

7

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

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 acknowledgement 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 JDeveloper BPEL Designer.

Adding Transformation Logic

This section contains the following tasks:

- Creating a POAcknowledge Project
- Importing Schema Files
- Creating a Transform Activity
- Testing the Transformation Logic

Creating a POAcknowledge Project

You must create a **POAcknowledge** project.

> **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 **OrderBookworkspace** in the **Applications Navigator** section of the designer window.
2. Select **New Project**.
3. Double-click **BPEL Process Project** in the **Items** window to display the BPEL Process Project window.
4. Enter **POAcknowledge** in the **BPEL Process Name** field. All other fields default to the correct values for creating an asynchronous BPEL process.
5. Click **OK**.

   This creates the **bpel.xml**, **POAcknowledge.bpel**, and **POAcknowledge.wsdl** files in the **Applications Navigator** section of the designer window.

Importing Schema Files

1. Copy `OrderBookingPO.xsd` and `POAcknowledge.xsd` from the *`Oracle_Home`*`\integration\orabpel\samples\tutorials\127.OrderBookingTutorial\PracticeFiles` directory to the *`Oracle_Home`*`\integration\jdev\jdev\mywork\OrderBookworkspace\POAcknowledge` directory.

2. Double-click **POAcknowledge.bpel** in the **Applications Navigator** section of the designer window to display the **Structure** section in the lower left section of JDeveloper BPEL Designer.
3. Select and right-click **Project Schemas** in the **Structure** section of the designer window.
4. Select **Import Schema**.

   The Import Schema window appears.
5. Click the **flashlight** icon to access the Open window.
6. Select **OrderBookingPO.xsd** from the **POAcknowledge** directory in which you placed it in Step 1 on page 7-2 and click **Open**.

   The file is added to the **URL** field of the Import Schema window.
7. Click **OK**.
8. Verify that **OrderBookingPO.xsd** appears under **POAcknowledge** > **Web Content** > **Miscellaneous Files** in the **Applications Navigator** section of the designer window. **OrderBookingPO.xsd** also appears under **Project Schemas** in the **Structure** section.
9. Select **Save** from the **File** main menu.
10. Expand **Message Types** > **Process WSDL - POAcknowledge.wsdl** > **POAcknowledgeRequestMessage** > **payload** in the **Structure** section.
11. Right-click **payload** and select **Edit Message Part** to display the Edit Message Part window.
12. Select **Element** and click the **flashlight** icon to the right of the **Element** field.
13. Expand **Project Schema Files** > **OrderBookingPO.xsd**.
14. Select **PurchaseOrder**.
15. Click **OK** to close the Type Chooser window and the Edit Message Part window.

    This defines the input parameter (payload) of the **PurchaseOrder** type.
16. Select and right-click **Project Schemas** in the **Structure** section of the designer window.
17. Select **Import Schema**.

    The Import Schema window appears.
18. Click the **flashlight** icon to access the Open window.
19. Select **POAcknowledge.xsd** from the **POAcknowledge** directory in which you placed it in Step 1 on page 7-2 and click **Open**.

    The file is added to the **URL** field of the Import Schema window.
20. Click **OK**.
21. Verify that **POAcknowledge.xsd** appears under **POAcknowledge** > **Web Content** > **Miscellaneous Files** in the **Applications Navigator** section of the designer window. **POAcknowledge.xsd** also appears under **Project Schemas** in the **Structure** section.
22. Select **Save** from the **File** main menu.
23. Expand **Message Types** > **Process WSDL - POAcknowledge.wsdl** > **POAcknowledgeResponseMessage** > **payload** in the **Structure** section.

24. Right-click **payload** and select **Edit Message Part** to display the Edit Message Part window.
25. Select **Element** and click the **flashlight** icon to the right of the **Element** field.
26. Expand **Project Schema Files** > **POAcknowledge.xsd**.
27. Select **POAcknowledge**.
28. Click **OK** to close the Type Chooser window and the Edit Message Part window.
29. Select **Save** from the **File** main menu.

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 acknowledgement 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.

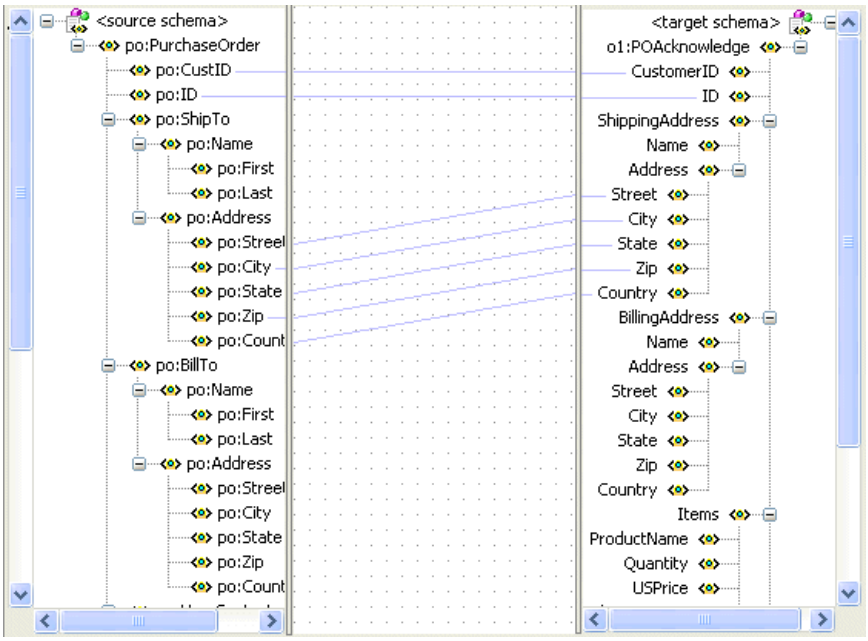
10. Right-click the source and target schemas (the top-most elements in both lists) and select **Expand All**.

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

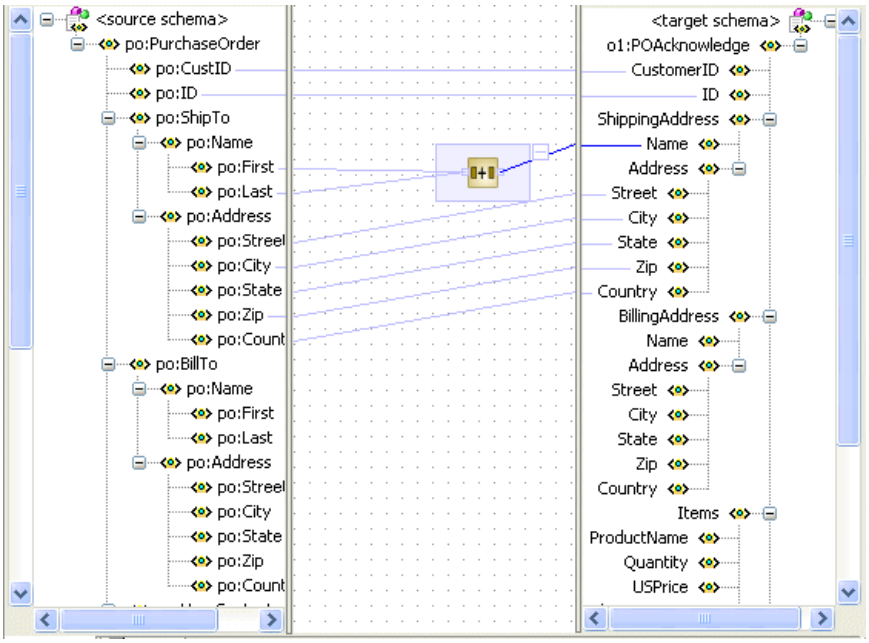
The Transformation window appears as follows:



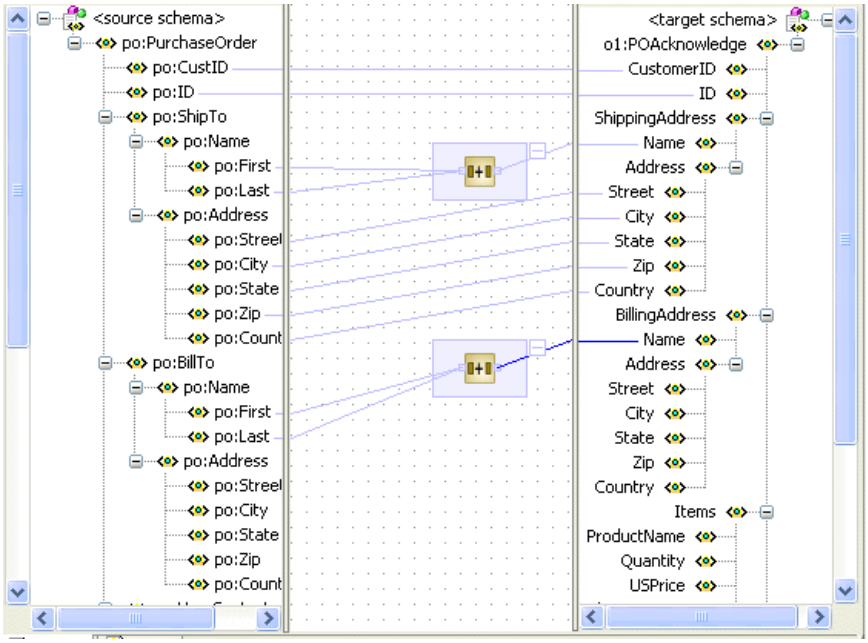
Note: If you drag and drop a source element to the wrong target element, right-click the connecting line and select **Delete**.

---

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 **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 **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 **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 **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**.

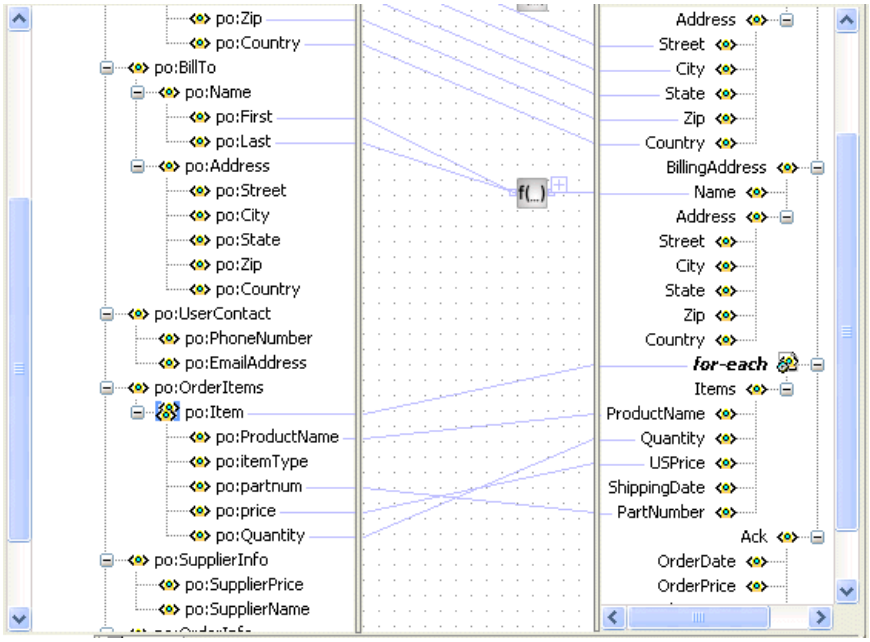
    The **for-each** node appears above the target **Items**.

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).

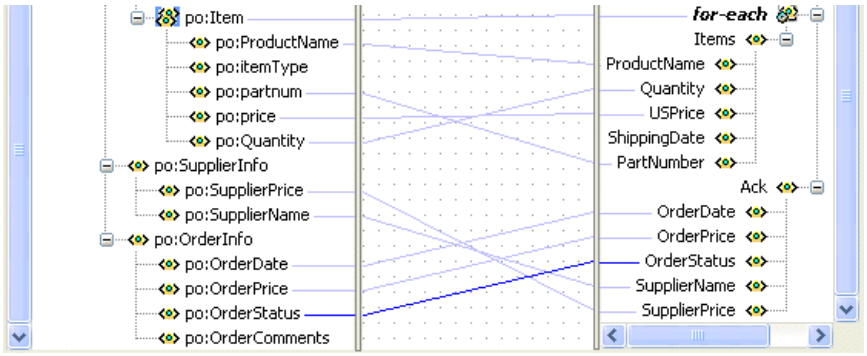
28. Click **OK** when prompted to confirm your mapping. This automatically maps all subelements.



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**.

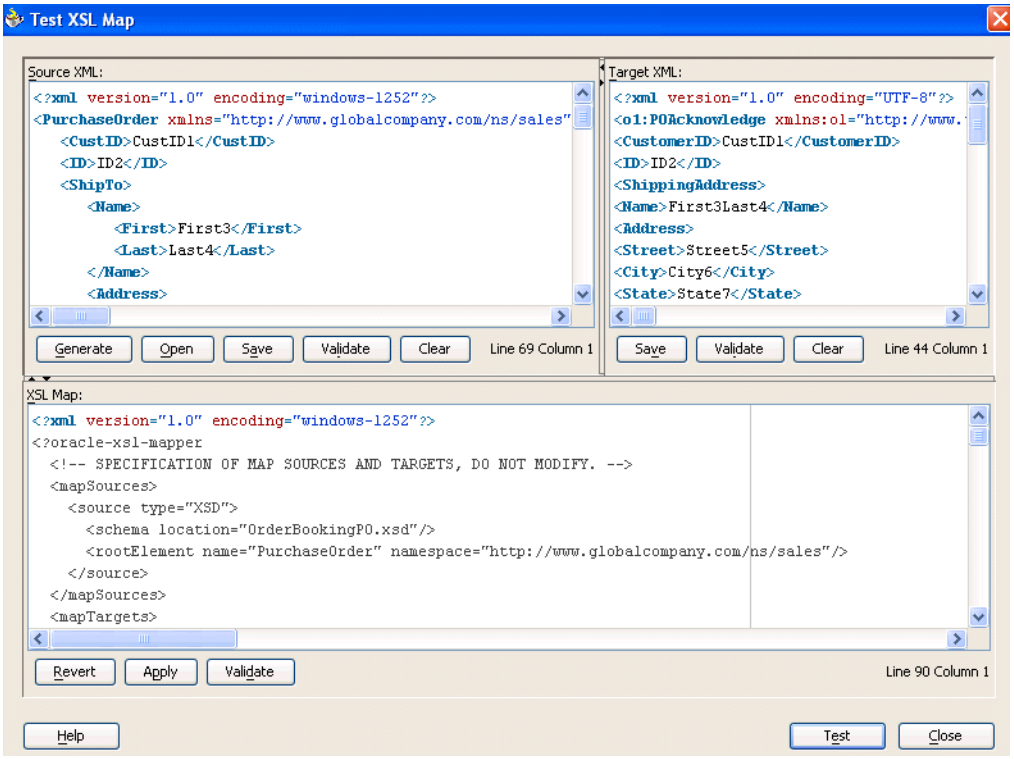
   This displays the Test XSL Map window.

2. Select **Generate** to create sample XML test data.

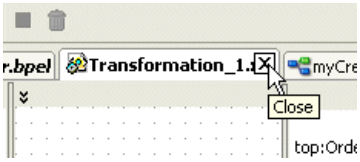
   Test data appears in the upper left corner in the **Source XML** section.

3. Click **Test** at the bottom right to test the translation logic.

   Sample test results appear in the upper right corner in the **Target XML** section.



4. Verify the results.
5. Click **Close** to close the test window.
6. Click **Yes** if prompted to save your transformation details.
7. Close the Transformation window by clicking the **x** button on the **Title** tab.



A translation file named **Transformation_1.xsl** appears under **POAcknowledge** > **Integration Content** in the **Applications 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 acknowledgement 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.

8

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

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 **FileReadAdapter**) 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 **FileReadAdapter**.
- Deploy the **BatchOrderProcessing** BPEL process.
- Run the deployed BPEL process from Oracle BPEL Console.
- 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

You must create a new project. Follow these instructions to create a new Batch Order Booking project. 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 **OrderBookworkspace** in the **Applications Navigator** section of the designer window.
2. Select **New Project**.

3. Double-click **BPEL Process Project** in the **Items** window to display the BPEL Process Project window.
4. Enter **BatchOrderProcessing** in the **BPEL Process Name** field.
5. Select **Empty BPEL Process** from the **Template** list.

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

6. Ensure that the **Use Default** check box is selected.
7. Click **OK**.

   This creates the **bpel.xml**, **BatchOrderProcessing.bpel**, and **BatchOrderProcessing.wsdl** files in the **Applications Navigator** section of the designer window.
8. Copy `OrderBookingPO.xsd` and `OrderBookingPO_1.xml` from the *`Oracle_Home`*`\integration\orabpel\samples\tutorials\127.OrderBookingTutorials\PracticeFiles` directory to the *`Oracle_Home`*`\integration\jdev\jdev\mywork\OrderBookworkspace\BatchOrderProcessing` directory.
9. Double-click **BatchOrderProcessing.bpel** in the **Applications Navigator** section of the designer window to display the **Structure** section in the lower left section of the designer window.
10. Select and right-click **Project Schemas** in the **Structure** section.
11. Select **Import Schema**.

    The Import Schema window appears.
12. Click the **flashlight** icon to access the Open window.
13. Select **OrderBookingPO.xsd** from the **BatchOrderProcessing** directory in which you placed it in Step 8 on page 8-3 and click **Open**.

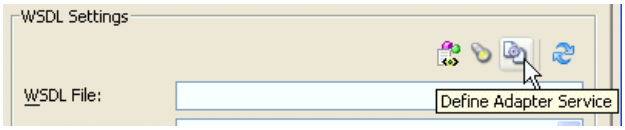
    The file is added to the **URL** field of the Import Schema window.
14. Click **OK**.
15. Verify that **OrderBookingPO.xsd** appears under **BatchOrderProcessing** > **Web Content** > **Miscellaneous Files** in the **Applications Navigator** section of the designer window. **OrderBookingPO.xsd** also appears under **Project 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 **Process Activities** 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. Enter **FileReadAdapter** in the **Name** field.
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. Click **Next**.
15. 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.
16. Click **Next**.
17. Enter **15 seconds** in the **Polling Frequency** field.
18. Ensure that the **Delete files after successful retrieval** check box is selected.
19. Click **Next**.
20. Click **Browse** next to the **Schema File URL** field on the Messages window to display the Type Chooser window.
21. Select **Project Schema Files** > **OrderBookingPO.xsd** > **PurchaseOrder**.
22. Click **OK**.

    The **Schema File URL** 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	**FileReadAdapter**

Field	Value
WSDL File	**file:/c:/oraBPEL/integration/jdev/jdev/mywork/OrderBookworkspace/BatchOrderProcessing/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.
Partner Link Type	**Read_plt**
My Role	**Read_role**
Partner Role	Leave unspecified.

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

    Note that **FileRead.wsdl** now appears under **BatchOrderProcessing** > **Integration Content**. 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 **WSIL Browser**) to access the WSDL Chooser window for automatically selecting the **OrderBooking** service.

Field	Value
Name	**OrderBooking**
WSDL File	Access this URL by clicking the **WSIL Browser** flashlight icon and expanding and selecting **LocalBPELServer** > **processes** > **default** > **OrderBooking** **http://localhost:9700/orabpel/default/OrderBooking/OrderBooking?wsdl** **See Also:** "Setting the Hostname in Your JDeveloper BPEL Designer Web Browser Preferences" on page 1-5 if you receive a parsing error when attempting to add a WSDL file in the WSDL Chooser window.
Partner Link Type	**OrderBooking**
My Role	**OrderBookingRequester**
Partner Role	**OrderBookingProvider**

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

Field	Value
Name	**receive_FileIn**
Partner Link	**FileReadAdapter**
Create Instance	Select this check box.

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

4. Click the first icon to the right of the **Variable** field. This is the automatic variable creation icon.
5. 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**.
6. Ensure that you selected the **Create Instance** check box, as mentioned in Step 3.
7. Click **OK**.
8. Drag and drop an **Invoke** activity from the **Component Palette** section to below the **receive_FileIn Receive** activity.
9. Double-click the **Invoke** icon to display the Invoke window.
10. Enter the following details:

Field	Value
Name	**invoke_OrderBooking**
Partner Link	**OrderBooking**

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

11. Click the first icon to the right of the **Input Variable** field. This is the automatic variable creation icon.
12. 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**.

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

Creating an Assign Activity

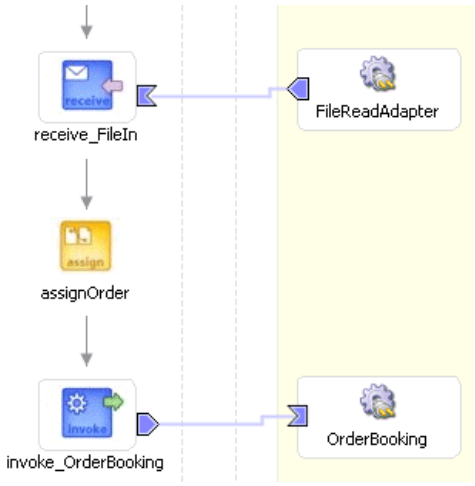
Summary: You now create an **Assign** activity that passes the data read from a file to **invoke_OrderBooking_initiate_InputVariable**, the variable used for initiating activity with the **OrderBooking** process.

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 Rules** tab.
7. Click **Create** to display the Create Copy Rule window.
8. Enter the following details:

Field	Value
From	
▪ **Type**	**Variable**
▪ **Variables**	Expand and select **Variables > receive_FileIn_Read_InputVariable** > **PurchaseOrder**
To	
▪ **Type**	**Variable**
▪ **Variables**	Expand and select **Variables > invoke_OrderBooking_initiate_InputVariable** > **payload**

9. Click **OK** to close the Create Copy Rule 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 Process

1. Go to the **Applications Navigator** section.
2. Right-click **BatchOrderProcessing**.
3. Select **Deploy** > **LocalBPELServer** > **Deploy to default domain**.
4. Enter the domain password of **bpel** when prompted.

   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 5 on page 2-15.

Running and Verifying the Order Process

1. Log into Oracle BPEL Console using Internet Explorer by selecting **Start** > **All Programs** > **Oracle - *Oracle_Home*** > **Oracle BPEL Process Manager 10.1.2** > **BPEL Console** (or refresh the page if it is already open).

   The **Dashboard** tab of Oracle BPEL Console appears. Note that your BPEL process, **BatchOrderProcessing** and a series of subprocesses on which it depends, now appears in the **Deployed BPEL Processes** list.
2. Copy the `OrderBookingPO_1.xml` file from the *`Oracle_Home`*`\integration\jdev\jdev\mywork\OrderBookworkspace\BatchOrderProcessing` directory to the directory you specified in Step 11 on page 8-4 (*`drive_letter`*`:\temp` on Windows operating systems or `/temp` on Unix operation systems).
3. Refresh Oracle BPEL Console after allowing the polling frequency of 15 seconds that you set in Step 17 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.
4. Complete the manual price quote task for **SelectManufacturing** by using the graphical user interface JSP page, as described in Step 10 on page 4-11.
5. Return to Oracle BPEL Console.
6. Click the **Instance** tab at the top.
7. Refresh Oracle BPEL Console and observe process instance creation and completion.

8. Click the instance.
9. Click the **Audit** link and click through other instances (of subprocesses) and follow the execution.
10. 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 **FileReadAdapter**) 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 **FileReadAdapter**.
- Deployed the **BatchOrderProcessing** BPEL process.
- Ran the deployed BPEL process from Oracle BPEL Console.
- Placed a file in a directory and observed that the file was read and then deleted after **OrderBooking** returned a response.

9

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

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 Console 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 acknowledgement file is written to the directory. Observe that the contents of the incoming purchase order file are transformed into an outgoing purchase order acknowledgement.

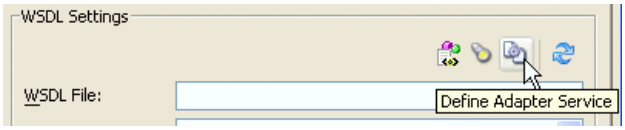
Designing the File Adapter's Write Functionality to Write the Order Acknowledgment

This section contains these tasks:

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

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 **Applications Navigator** section of the designer window.
2. Double-click **POAcknowledge.bpel**.
3. Drag and drop a **PartnerLink** activity from the **Component Palette** section onto the right side of the designer window.
4. Enter **FileOutbound** 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 **File Adapter** on the Adapter Type window and click **Next**.
8. Enter **FileWrite** in the **Service Name** field on the Service Name window and click **Next**.
9. 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**.
10. Select **Directory Specified as Physical Path**.
11. Click **Browse** next to the **Directory for Outgoing Files (physical path)** field.
12. 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.
13. Click **Select**.
14. Enter **po_%yyMMddHHmmss%.xml** in the **File Naming Convention** field.
15. Unselect the **Number of Messages Equal, Elapsed Time Exceeds**, and **File Size Exceeds** check boxes and click **Next**.
16. Click **Browse** next to the **Schema File URL** field on the Messages window.
17. Select **Project Schema Files** > **POAcknowledge.xsd** > **POAcknowledge** in the Type Chooser window.
18. Click **OK**.

    The **Schema File URL** field (**POAcknowledge.xsd**) and the **Schema Element** field (**POAcknowledge**) are filled in.
19. Click **Next**.
20. Click **Finish**.

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

Field	Value
Name	**FileOutbound**
WSDL File	**file:/c:/oraBPEL/integration/jdev/jdev/mywork/OrderBookworkspace/POAcknowledge/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.
Partner Link Type	**Write_plt**
My Role	Leave unspecified.
Partner Role	**Write_role**

21. Click **OK**.
22. Select **Save** from the **File** main menu.

    Note that **FileWrite.wsdl** now appears under **POAcknowledge** > **Integration Content**. 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. Drag and drop an **Invoke** activity from the **Component Palette** section to below the **transformPO Transform** activity.
2. Double-click the **Invoke** icon to display the Invoke window.
3. Enter the following details:

Field	Value
Name	**invokeFileOut**
Partner Link	**FileOutbound**

The **Operation** (**Write**) 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 **invokeFileOut_write_InputVariable** is automatically created in the **Input Variable** field. This variable is automatically assigned a message type of **FileWritePOAcknowledge_msg**.
6. Click **OK**.
7. 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 Rules** tab.
7. Click **Create** to display the Create Copy Rule window.
8. Enter the following values:

Field	Value
From	
▪ **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 Rule window and the Assign window.
10. Select **Save** from the **File** main menu.

Deploying the POAcknowledge Service

1. Go to the **Applications Navigator** section.
2. Right-click **POAcknowledge**.
3. Select **Deploy** > **LocalBPELServer** > **Deploy to default domain**.
4. Enter the domain password of **bpel** when prompted.

   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 5 on page 2-15.

Running and Verifying the POAcknowledge Process

1. Log into Oracle BPEL Console using Internet Explorer by selecting **Start** > **All Programs** > **Oracle - *Oracle_Home*** > **Oracle BPEL Process Manager 10.1.2** > **BPEL Console** (or refresh the page if it is already open).

   The **Dashboard** tab of Oracle BPEL Console appears. Note that your BPEL process, **POAcknowledge**, now appears in the **Deployed BPEL Processes** list.
2. Click **POAcknowledge**.
3. Enter details in all fields of the HTML Form and click **Post XML Message**.
4. Refresh Oracle BPEL Console.

5. Click the **Instances** tab.
6. Click the **POAcknowledge** instance.
7. Click the **Audit** link to view the transformation logic.
8. View the output file written in the directory you selected in Step 12 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 **Applications Navigator** section.
2. Double-click **OrderBooking.bpel**.
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 **OrderBooking** process:

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

Field	Value
Name	**WritePOA**
WSDL File	Access this URL by clicking the **WSIL Browser flashlight** icon and expanding and selecting **LocalBPELServer** > **processes** > **default** > **POAcknowledge**. **http://localhost:9700/orabpel/default/POAcknowledge/POAcknowledge?wsdl** **See Also:** "Setting the Hostname in Your JDeveloper BPEL Designer Web Browser Preferences" on page 1-5 if you receive a parsing error when attempting to add a WSDL file in the WSDL Chooser window.
Partner Link Type	**POAcknowledge**
My Role	**POAcknowledgeRequester**
Partner Role	**POAcknowledgeProvider**

5. Click **OK**.
6. 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 **WritePOAcknowledgment** 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

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 Rules** tab.
7. Click **Create** to display the Create Copy Rule 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 Rule 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**.

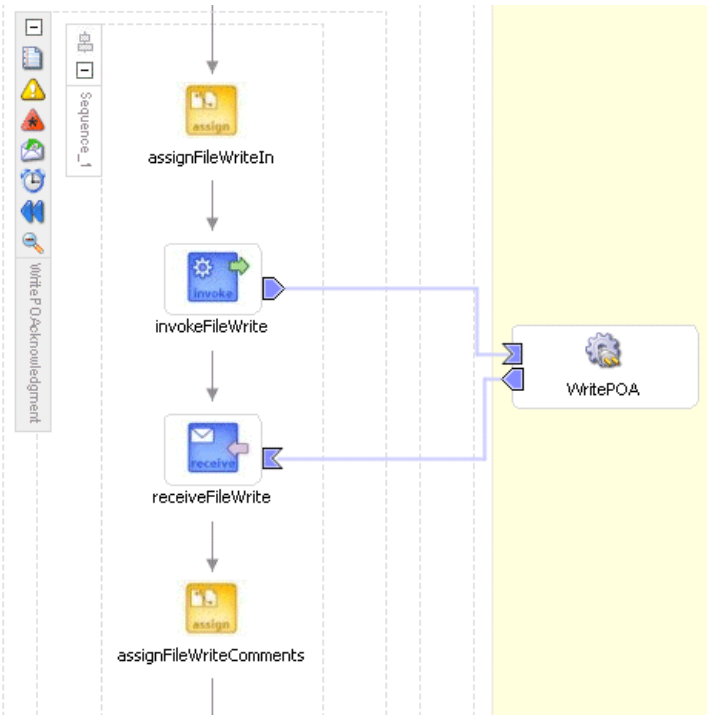
6. Click the **Copy Rules** tab.
7. Click **Create** to display the Create Copy Rule window.
8. Enter the following values:

> **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**	**concat(bpws:getVariableData('inputVariable','payload','/ns1:PurchaseOrder/ns1:OrderInfo/ns1:OrderComments'), ' - Write POAck complete')**
To	
▪ **Type**	**Variable**
▪ **Variables**	Expand and select **Variables** > **inputVariable** > **payload** > **ns1:PurchaseOrder** > **ns1:OrderInfo** > **ns1:OrderComments**

9. Click **OK** to close the Create Copy Rule 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 **Applications Navigator** section.
2. Right-click **OrderBooking**.
3. Select **Deploy** > **LocalBPELServer** > **Deploy to default domain**.
4. Enter the domain password of **bpel** when prompted.
5. Increment the version number of the project when prompted and click **OK** (for example, enter **1.5**).

   This compiles the BPEL process. Review the bottom of the window for any errors. If there are no errors, deployment was successful.

Running and Verifying the Order Booking Process

1. Log into Oracle BPEL Console using Internet Explorer by selecting **Start** > **All Programs** > **Oracle - *Oracle_Home*** > **Oracle BPEL Process Manager 10.1.2** > **BPEL Console** (or refresh the page if it is already open).

   The **Dashboard** tab of Oracle BPEL Console appears. Note that your BPEL processes, **POAcknowledge** and **OrderBooking**, appear in the **Deployed BPEL Processes** list.
2. Click the **BPEL Processes** tab.
3. Note which **OrderBooking** version has an asterisk next to the version number. This is the default version.
4. 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.
5. Scroll down and click **Mark as Default** to make this the default version of **OrderBooking**.
6. 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.
7. If you have installed Patch 1, go to Step 8. Otherwise, you must first perform Steps 7a through 7c. See Chapter 1, "Introduction" for information about where to obtain Patch 1.
   a. Click **Clear WSDL Cache** under **Related Tasks** in the lower-left section.
   b. Click **Back to Processes** when prompted.
   c. Go to Step 8.
8. Click the **Dashboard** tab.
9. Click the **OrderBooking** instance version you increased in Step 5 on page 9-10 and start it in one of the following ways:
   - Open the provided **CreateOrderBookingUI** application at `http://localhost:9700/CreateOrderBookingUI` and provide inputs to initiate a process.

- Enter details in all fields of the HTML Form and click **Post XML Message**.
- Select **XML Source** from the **Initiating a test instance** list. 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.

---

10. 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 8-8 is complete.
11. Complete the manual price quote task for **SelectManufacturing** by using the graphical user interface JSP page, as described in Step 10 on page 4-11.
12. Return to Oracle BPEL Console.
13. Refresh Oracle BPEL Console.
14. 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.
15. Click the instance.
16. 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 Console 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 acknowledgement file was written to the directory. Observed that the contents of the incoming purchase order file were transformed into an outgoing purchase order acknowledgement.

10

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

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 Console 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 an Order Fulfillment Process

1. Right-click **OrderBookworkspace** in the **Applications Navigator** section of the designer window.
2. Select **New Project**.

3. Double-click **BPEL Process Project** in the **Items** window to display the BPEL Process Project window.
4. Enter **FulfillOrder** in the **BPEL Process Name** field. All other fields default to the correct values for creating an asynchronous BPEL process.
5. Click **OK**.

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

Importing the Schema

1. Copy `OrderBookingPO.xsd` from the *`Oracle_Home`*`\integration\orabpel\samples\tutorials\127.OrderBookingTutorial\PracticeFiles` directory to the *`Oracle_Home`*`\integration\jdev\jdev\mywork\OrderBookworkspace\FulFillOrder` directory.
2. Double-click **FulfillOrder.bpel** in the **Applications Navigator** section of the designer window to display the **Structure** section in the lower left section of the designer window.
3. Select and right-click **Project Schemas** in the **Structure** section.
4. Select **Import Schema**.

   The Import Schema window appears.
5. Click the **flashlight** icon to access the Open window.
6. Select **OrderBookingPO.xsd** from the **FulFillOrder** directory in which you placed it in Step 1 on page 10-3 and click **Open**.

   The file is added to the **URL** field of the Import Schema window.
7. Click **OK**.
8. Verify that **OrderBookingPO.xsd** appears under **FulfillOrder** > **Web Content** > **Miscellaneous Files** in the **Applications Navigator** section of the designer window. **OrderBookingPO.xsd** also appears under **Project Schemas** in the **Structure** section.
9. Expand **Message Types** > **Process WSDL - FulfillOrder.wsdl** > **FulfillOrderRequestMessage** > **payload** in the **Structure** section.
10. Right-click **payload** and select **Edit Message Part** to display the Edit Message Part window.
11. Select **Element** and click the **flashlight** icon to the right of the **Element** field.
12. Expand **Project Schema Files** > **OrderBookingPO.xsd**.
13. Select **PurchaseOrder**.
14. Click **OK** to close the Type Chooser window and the Edit Message Part window.

    This defines the input parameter (payload) of the **PurchaseOrder** type.
15. Select **Save** from the **File** main menu.

Creating a Database Connection and Samples Tables

Summary: You now configure JDeveloper BPEL Designer 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:

- Configuring a Database Instance Design-Time Connection
- Configuring a BPEL Server Runtime Connection
- Creating a JDeveloper Database Connection
- Creating Sample Tables
- Testing the Database Connection

Configuring a Database Instance Design-Time Connection

Follow these steps to configure and connect to the database instance.

1. Select **Manage Libraries** from the **Tools** main menu.
2. Select **System Libraries** > **Oracle8i Lite**.

   Oracle8i Lite appears in the **Library Name** field.
3. Enter ***Oracle_Home*\integration\orabpel\lib\olite40.jar** in the **Class Path** field or click **Edit** to select the path.
4. Leave the remaining fields as they are.
5. Click **OK**.

Configuring a BPEL Server Runtime Connection

1. Open a text editor.
2. Open the *Oracle_Home*\integration\orabpel\system\appserver\oc4j\j2ee\home\application-deployments\default\DbAdapter\oc4j-ra.xml file.
3. Locate the following connection factory settings in the file and ensure that the values in bold appear. If not, change them to these values.

```
<connector-factory location="eis/DB/BPELSamples" connector-name="Database
Adapter">
   <config-property name="driverClassName"
value="oracle.lite.poljdbc.POLJDBCDriver"/>
   <config-property name="connectionString"
value="jdbc:polite4@localhost:100:orabpel"/>
   <config-property name="userName" value="system"/>
   <config-property name="password" value="manager"/>
   <config-property name="minConnections" value="5"/>
   <config-property name="maxConnections" value="5"/>
   <config-property name="minReadConnections" value="1"/>
   <config-property name="maxReadConnections" value="1"/>
   <config-property name="usesExternalConnectionPooling" value="false"/>
   <config-property name="dataSourceName" value=""/>
   <config-property name="usesExternalTransactionController" value="false"/>
   <config-property name="platformClassName"
value="oracle.toplink.internal.databaseaccess.DatabasePlatform"/>
   <config-property name="usesNativeSequencing" value="true"/>
   <config-property name="sequencePreallocationSize" value="50"/>
   <config-property name="tableQualifier" value=""/>
</connector-factory>
```

4. Stop and restart Oracle BPEL Server from the Start Menu, as described in "Starting Oracle BPEL Process Manager Components" on page 1-4.

Creating a JDeveloper Database Connection

1. Return to JDeveloper BPEL Designer.
2. Select **Connection Navigator** from the **View** main menu.
3. Right-click **Database** in the **Connections - Navigator** window and select **New Database Connection**.

   This starts the Create Database Connection Wizard.
4. Click **Next** on the Welcome window.
5. Enter **myConnection** in the **Connection Name** field of the Type window.
6. Select **Third Party JDBC Driver** from the **Connection Type** list, and click **Next**.
7. Enter **system** in the **Username** field of the Authentication window.
8. Enter **manager** in the **Password** field.
9. Leave the remaining fields as they are, and click **Next**.
10. Click **New** to the right of the **Driver Class** field on the Connection window.

    The Register JDBC Driver window appears.
11. Enter **oracle.lite.poljdbc.POLJDBCDriver** in the **Driver Class** field. Ensure that there are no blank spaces at the end of this entry.
12. Click **New** to the right of the **Library** field.

    The Create Library window appears.
13. Enter a new library name in the **Library Name** field (for example, **newOracle8Lite**).
14. Enter *Oracle_Home*/**integration/orabpel/lib/olite40.jar** in the **Class Path** field or click **Edit** to select the path.
15. Leave the remaining fields as they are.
16. Click **OK** to close the Create Library and Register JBDC Driver windows.
17. Enter **jdbc:polite4@localhost:100: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 10.1.2** > **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:

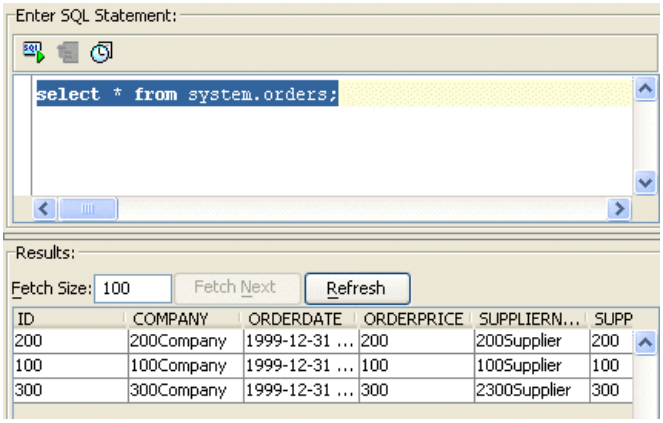
   ```
   polsql.cmd  @PracticeFiles\insertTable.sql
   ```

   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.

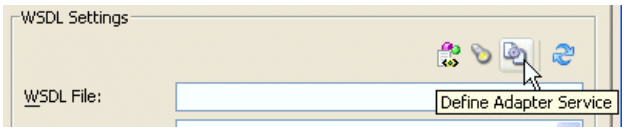


Creating a Partner Link that Uses the Database Adapter

> **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 **Applications 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. Select **ORDERS** and **ORDERITEMS**.
17. Click **>** to move both tables to the **Selected** field.
18. Click **OK**. You are returned to the Select Table window.
19. Select **SYSTEM.ORDERS** and click **Next**.

    The relationship definitions are automatically created and displayed as 1:M.
20. Select **ORDERITEMS (ordid) 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.
21. Click **Next**.
22. Leave all attributes as default for the attributes to return from the query on the Select Attributes page.
23. Click **Next** twice.
24. Click **Finish**.

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

Field	Value
Name	**DBInsert**

Field	Value
WSDL File	**file:/c:/orabpel/integration/jdev/jdev/mywork/OrderBookworkspace/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.
Partner Link Type	**writeDB_plt**
My Role	Leave unspecified.
Partner Role	**writeDB_role**

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

    Note that **writeDB.wsdl** now appears under **FulFillOrder** > **Integration Content**. 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.

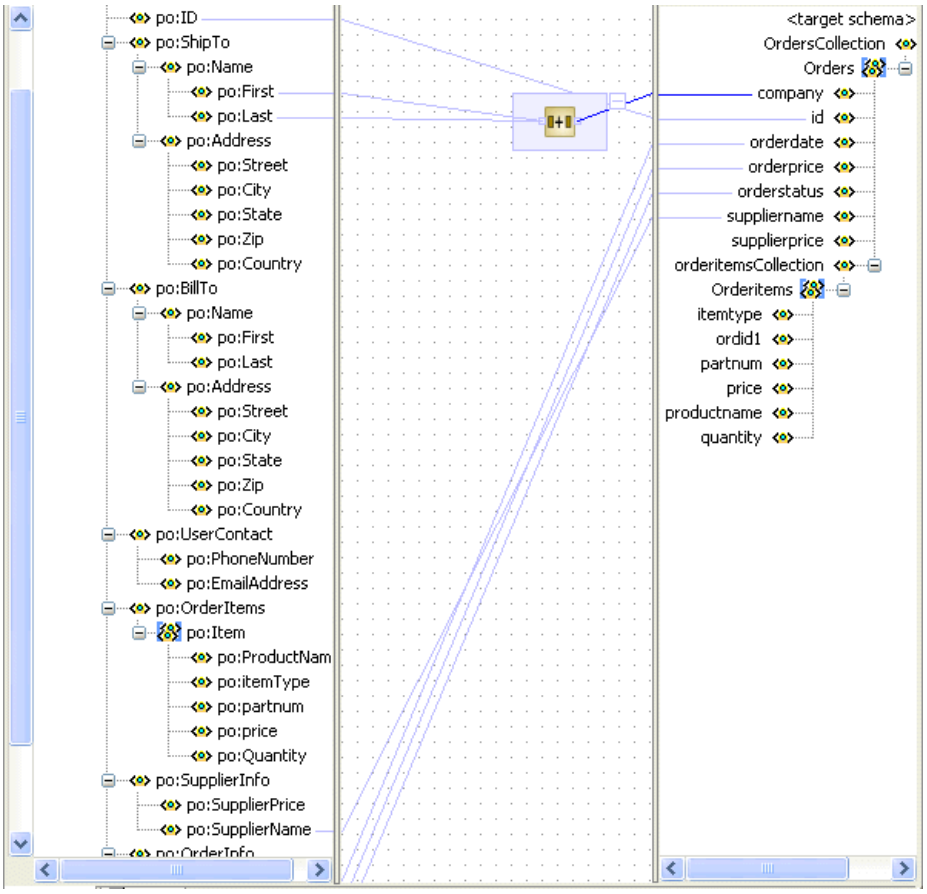
5. Click the **Transformation** tab.
6. Select **inputVariable** in the **Source Variable** field and **payload** in the **Source Part** field.
7. Select **writeDBrecord_merge_InputVariable** in the **Target Variable** field and **OrdersCollection** in the **Target Part** field.
8. Leave the **Mapper 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:

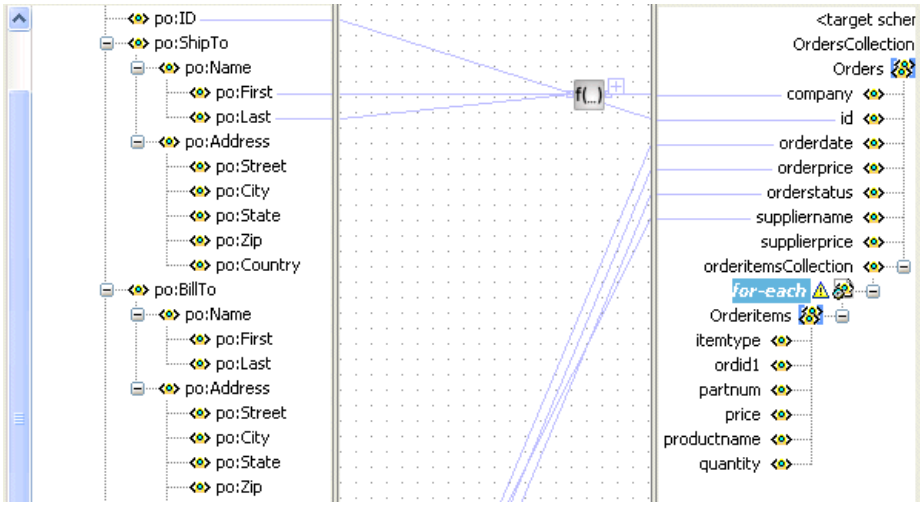
Source	Target
po:ID	**id**
po:SupplierName	**suppliername**
po:SupplierPrice	**supplierprice**
po:OrderDate	**orderdate**
po:OrderPrice	**orderprice**
po:OrderStatus	**orderstatus**

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.

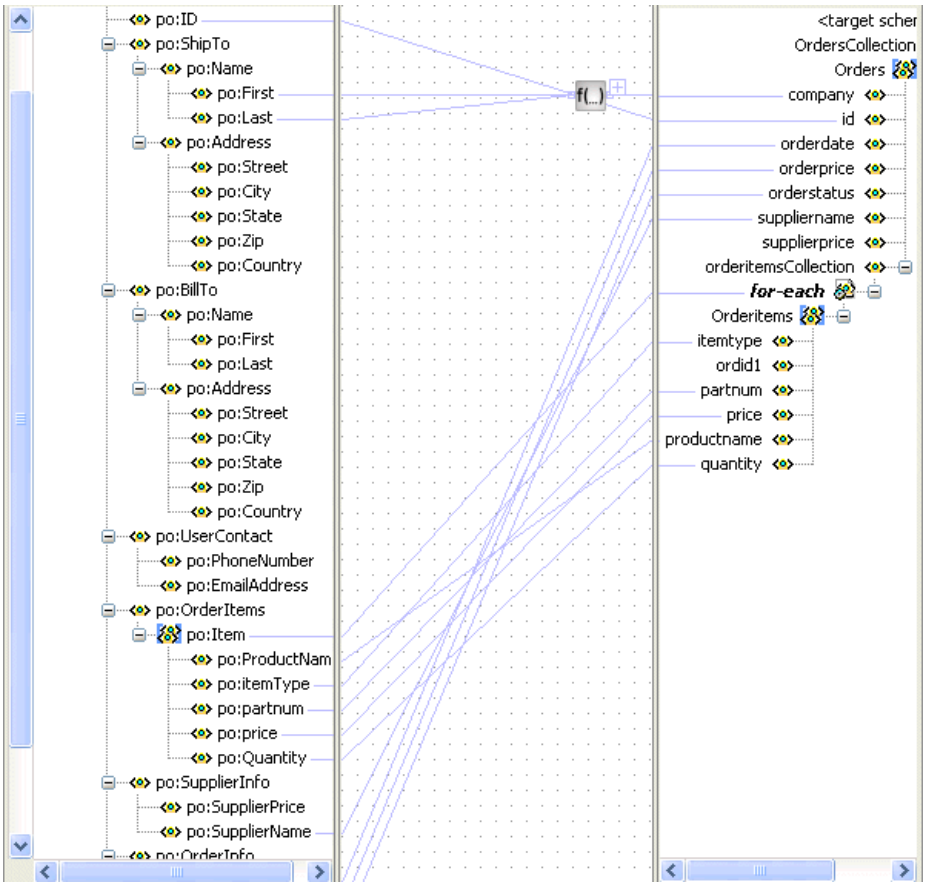


19. Right-click the target **Orderitems** and select **Add XSL Node** > **for-each**.

    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. Select **Generate** to create sample test data.

   Test data appears in the upper left corner in the **Source XML** section.

3. Click **Test** to test the transformation logic.

   Sample test results appear in the upper right corner in the **Target XML** section.

4. Verify the results. For example, note that the first and last names in the source code now appear as a single company name.

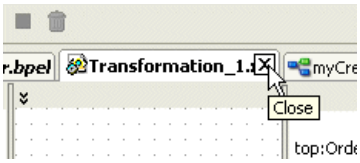
   Note: If you click **Validate**, the following message appears:

   ```
   Target XML is invalid for the target schema.
   <Line 6, Column 34>: XML-24525: (Error) Invalid text '2005-05-23'
    in element: 'orderdate'
   ```

   You can ignore this message. This message displayed because the source **PurchaseOrder:OrderInfo:OrderDate** is a **date** field and the target **orderCollections:orders:orderdate** is a **dateTime** field.

5. Click **Close** to close the test window.

6. Click **Yes** if prompted to save your transformation details.
7. Close the Transformation window by clicking the **x** button on the **Title** tab.



A file named **Transformation_1.xsd** appears under **FulfillOrder** > **Integration Content** in the **Applications Navigator** section.

Validating, Compiling, and Deploying the FulFillOrder Process

1. Go to the **Applications Navigator** section.
2. Right-click **FulfillOrder**.
3. Select **Deploy** > **LocalBPELServer** > **Deploy to default domain**.
4. Enter the domain password of **bpel** when prompted.

   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 5 on page 2-15.

Running the FulFillOrder Process

1. Log into Oracle BPEL Console using Internet Explorer by selecting **Start** > **All Programs** > **Oracle - *Oracle_Home*** > **Oracle BPEL Process Manager 10.1.2** > **BPEL Console** (or refresh the page if it is already open).

   The **Dashboard** tab of Oracle BPEL Console appears. Note that your BPEL process, **FulfillOrder**, now appears in the **Deployed BPEL Processes** list.
2. 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**.

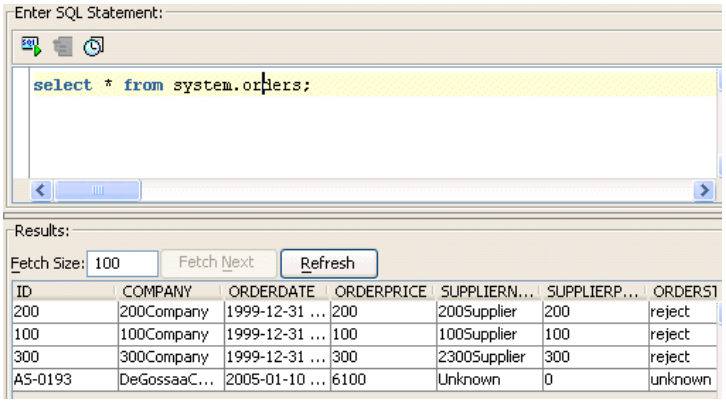
   or

   - Select **XML Source** from the **Initiating a test instance** list. 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 Console.
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. Select **Connection Navigator** from the **View** main menu.
2. Expand **Database**.
3. 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.
4. Right-click the SQL statement you entered in Step 4 on page 10-6 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 **Applications Navigator** section.
2. Double-click **OrderBooking.bpel**.
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 Order Booking process:

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

Field	Value
Name	**FulfillOrder**
WSDL File	Access this URL by clicking the **WSIL Browser flashlight** icon and expanding and selecting **LocalBPELServer** > **processes** > **default** > **FulfillOrder**. **http://localhost:9700/orabpel/default/FulfillOrder/FulfillOrder?wsdl** **See Also:** "Setting the Hostname in Your JDeveloper BPEL Designer Web Browser Preferences" on page 1-5 if you receive a parsing error when attempting to add a WSDL file in the WSDL Chooser window.
Partner Link Type	**FulfillOrder**
My Role	**FulfillOrderRequester**
Partner Role	**FulfillOrderProvider**

5. Click **OK**.
6. 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 **OrderFulFillment** 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

> **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 Rules** tab.
7. Click **Create** to display the Create Copy Rule 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 Rule 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 Rules** tab.
7. Click **Create** to display the Create Copy Rule window.
8. Enter the following values:

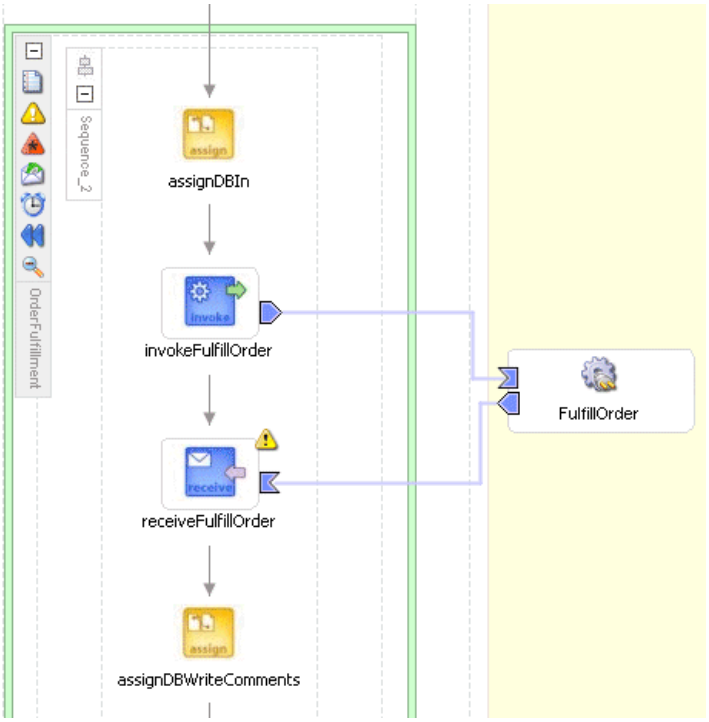
> **Note:** Instead of manually entering an expression, you can press **Ctrl** and then the space bar in the **Expression** field. Scroll through the list of values that appears and double-click the value you want. Edit the value as necessary. As you enter information, a trailing slash can appear. This 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**	**concat(bpws:getVariableData('inputVariable','payload','/ns1:Purchase Order/ns1:OrderInfo/ns1:OrderComments'), ' - Order FulFillment Complete')**
To	
▪ **Type**	**Variable**

Field	Value
▪ **Variables**	Expand and select > **Variables** > **inputVariable** > **payload** > **ns1:PurchaseOrder** > **ns1:OrderInfo** > **ns1:OrderComments**

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

   When complete, the designer window looks as follows:



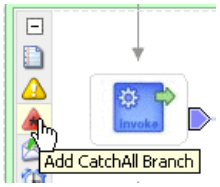
10. Click the **-** sign to close the **OrderFulfillment Scope** activity.
11. Select **Save** from the **File** main menu.

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).

1. Ensure that you are in the **FulFillOrder** process. If not, double-click **FulFillOrder.bpel** in the **Applications Navigator** section of the designer window.
2. Drag and drop a **Scope** activity from the **Component Palette** section to below the **transformDBRecord Transform** activity.
3. 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**.



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

Field	Value
From	
▪ **Type**	**Expression**
▪ **Expression**	**'0'**
To	
▪ **Type**	**Variable**
▪ **Variables**	Expand and select **Variables** > **dbStatus**

23. Click **OK** to close the Create Copy Rule window and the Assign window.
24. Drag and drop another **Assign** activity from the **Component Palette** section to below the **writeDBrecord Invoke** activity inside the **db_write_scope Scope** activity.
25. Double-click the **Assign** icon to display the Assign window.

26. Click the **General** tab.
27. Enter **assign_db_1** in the **Name** field.
28. Click **Apply**.
29. Click the **Copy Rules** tab.
30. Click **Create** to display the Create Copy Rule window.
31. Enter the following values:

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

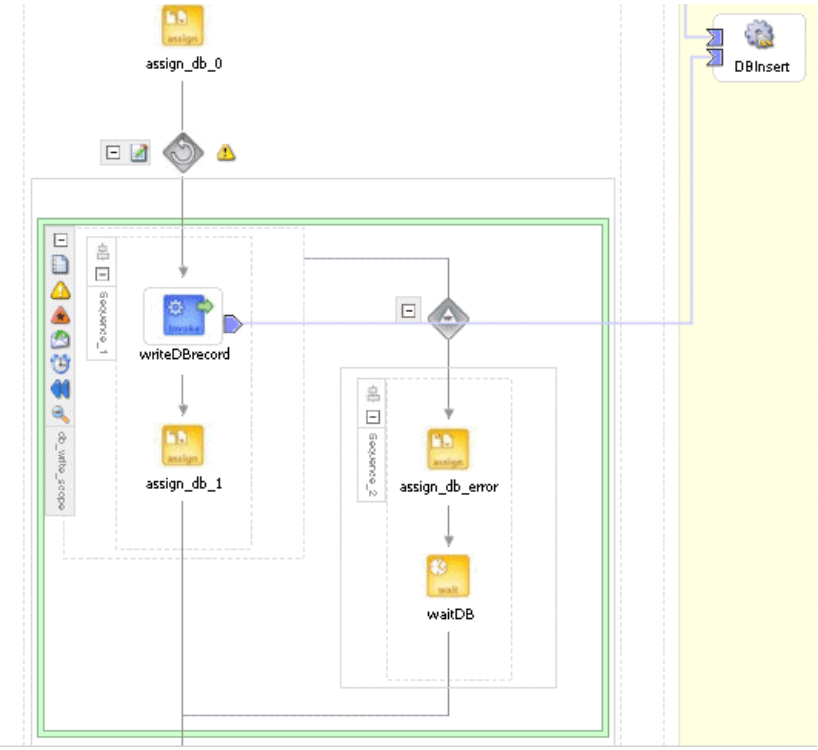
32. Click **OK** to close the Create Copy Rule window and the Assign window.
33. Drag and drop a third **Assign** activity from the **Component Palette** section into the **Catch All** branch on the right side of the **db_write_scope Scope** activity.
34. Double-click the **Assign** icon to display the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

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

40. Click **OK** to close the Create Copy Rule window and the Assign window.
41. Drag and drop a **Wait** activity from the **Component Palette** section into the **Catch** branch below the **assign_db_error Assign** activity.
42. Double-click the **Wait** activity to display the Wait window.
43. Enter **waitDB** in the Name field.
44. 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**).

45. Click **OK**.
46. 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:



47. Click the **-** sign to close the **db_write_scope Scope** activity.
48. Expand the **While** activity.
49. Drag the **db_write_scope Scope** activity inside the **While** activity.
50. 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
    ```

51. Click **OK**.
52. Select **Save** from the **File** main menu.

Validating, Compiling, and Deploying the OrderBooking Process

1. Go to the **Applications Navigator** section.
2. Right-click **OrderBooking**.
3. Select **Deploy** > **LocalBPELServer** > **Deploy to default domain**.
4. Enter the domain password of **bpel** when prompted.
5. Increment the version number of the project when prompted (for example, enter **1.6**) and click **OK**.

   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 5 on page 2-15.

Running the Order Booking Process

1. Log into Oracle BPEL Console using Internet Explorer by selecting **Start** > **All Programs** > **Oracle - *Oracle_Home*** > **Oracle BPEL Process Manager 10.1.2** > **BPEL Console** (or refresh the page if it is already open).

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

2. Click the **BPEL Processes** tab.

3. Note which **OrderBooking** version has an asterisk next to the version number. This is the default version.

4. 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.

5. Scroll down and click **Mark as Default** to make this the default version of **OrderBooking**.

6. 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.

7. If you have installed Patch 1, go to Step 8. Otherwise, you must first perform Steps 7a through 7b on page 9-10. See Chapter 1, "Introduction" for information about where to obtain Patch 1.

8. Click the **Dashboard** tab.

9. Start the correct **OrderBooking** version instance in one of the following ways:

   - Open the provided **CreateOrderBookingUI** application at `http://localhost:9700/CreateOrderBookingUI` and provide inputs to initiate a process.
   - Click **OrderBooking** in the **Deployed BPEL Processes** list.
     - Enter details in all fields of the HTML Form and click **Post XML Message**.

       or

     - Select **XML Source** from the **Initiating a test instance** list. 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.

   - *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 10 on page 4-11.

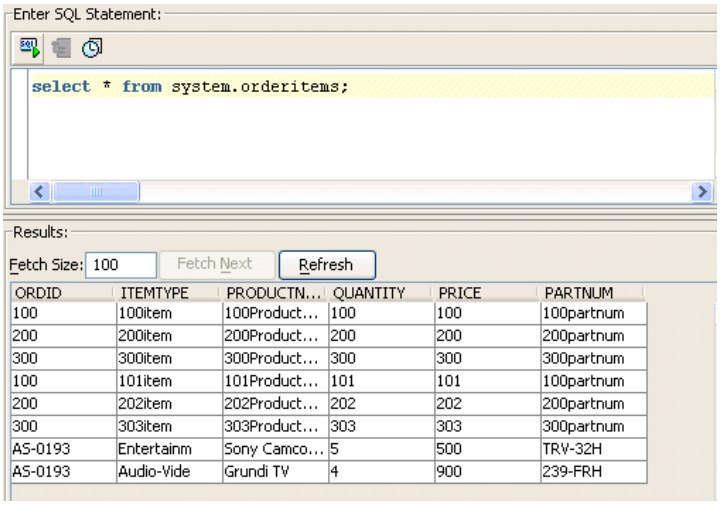
11. Return to Oracle BPEL Console.

12. Refresh Oracle BPEL Console.

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.
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.



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 Console to receive and send data to the database

11

Designing the Human 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

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 Console.
- 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 Console.
- 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:

Defining the Purchase Order Approval Service

In this phase of the tutorial, you create a new project and define the human workflow service project.

This section contains these tasks:

- Running the Order Approval Process

Creating a New Project

1. Right-click **OrderBookworkspace** in the **Applications Navigator** section of the designer window.
2. Select **New Project**.
3. Double-click **BPEL Process Project** in the **Items** window to display the BPEL Process Project window.
4. Enter **OrderApproval** in the **BPEL Process Name** field. All other fields default to the correct values for creating an asynchronous BPEL process.
5. Click **OK**.

Importing a Project Schema

1. Copy `OrderBookingPO.xsd`, `OrderBookingPO_1.xml`, and `Order.xsl` from the *ORACLE_HOME*`\integration\orabpel\samples\tutorials\127.OrderBookingTutorial\PracticeFiles` directory to the *ORACLE_HOME*`\integration\jdev\jdev\mywork\OrderBookworkspace\OrderApproval` directory.
2. Double-click **OrderApproval.bpel** in the **Applications Navigator** section of the designer window.
3. Select and right-click **Project Schemas** in the **Structure** section.
4. Select **Import Schema**.

   The Import Schema window appears.
5. Click the **flashlight** icon to access the Open window.
6. Select **OrderBookingPO.xsd** from the **OrderApproval** directory in which you placed it in Step 1 on page 11-3 and click **Open**.

   The file is added to the **URL** field of the Import Schema window.
7. Click **OK**.
8. Verify that **OrderBookingPO.xsd** appears under **OrderApproval** > **Web Content** > **Miscellaneous Files** in the **Applications Navigator** section of the designer window. **OrderBookingPO.xsd** also appears under **Project Schemas** in the **Structure** section.
9. Expand **Message Types** > **Process WSDL - OrderApproval.wsdl** > **OrderApprovalRequestMessage** > **payload** in the **Structure** section.
10. Right-click **payload** and select **Edit Message Part** to display the Edit Message Part window.
11. Select **Element** and click the **flashlight** icon to the right of the **Element** field.
12. Expand **Project Schema Files** > **OrderBookingPO.xsd**.
13. Select **PurchaseOrder**.
14. Click **OK** to close the Type Chooser window and the Edit Message Part window.

    This defines the input parameter (payload) of the PurchaseOrder type.
15. Expand **Message Types** > **Process WSDL - OrderApproval.wsdl** > **OrderApprovalResponseMessage** > **payload** in the **Structure** section.

16. Right-click payload and select **Edit Message Part** to display the Edit Message Part window.
17. Select **Element** and click the **flashlight** icon to the right of the **Element** field.
18. Expand **Project Schema Files** > **OrderBookingPO.xsd**.
19. Select **PurchaseOrder**.
20. Click **OK** to close the Type Chooser window and the Edit Message Part window.

    This defines the output parameter (payload) of the **PurchaseOrder** type.
21. Select **Save** from the **File** main menu.

Creating a User Task Activity

1. Ensure that **Process Activities** is selected in the drop-down list of the **Component Palette** section.
2. Drag and drop a **User Task** activity from the **Component Palette** section to below the **receiveInput Receive** activity.

    This starts the Workflow Wizard.
3. Leave the **Create New Workflow** check box selected and click **Next**.

    The Workflow Pattern window appears.
4. Select **Sequential Workflow** from the **Workflow Patterns** list and click **Next**.

    The Task Details window appears.
5. Click the icon to the right of the **Task Title** field to display the XPath Expression Builder window.
6. Press **Ctrl** and then the space bar to add the following in the **Expression Builder** field:

```
concat(string('Order No:'),bpws:getVariableData('inputVariable','payload','
 /ns1:PurchaseOrder/ns1:ID'),string(' for '),bpws:getVariableData
 ('inputVariable','payload','/ns1:PurchaseOrder/ns1:CustID'))
```

7. Click **OK** when complete to return to the Task Details window.
8. Click the **flashlight** icon to the right of the **Payload** field to display the Variable XPath Builder window.
9. Expand and select **Process** > **Variables** > **inputVariable** > **payload** > **ns1:PurchaseOrder**.
10. Click **OK**.

    The following information displays in the **Payload** field:

```
bpws:getVariableData('inputVariable','payload','/ns1:PurchaseOrder')
```

11. Click **XSL File** and select **Order.xsl**. This is the XML style sheet for the payload display.
12. Leave the **Task Creator** field empty.
13. Select **10** for the **Minutes** field of the **Expiration Duration** section.
14. Click **Next**.

    The Task Outcomes window appears.
15. Leave the two default outcomes as **ACCEPT** and **REJECT**.

16. Click **Next**.

    The Task Notifications window appears.

17. Accept the default settings and click **Next**.

    The Assignment Policy window appears.

18. Select **Management chain** and click **Next**.

    The Assignees window appears.

19. Select **Assign to Group**.

20. Select **Group** and click the **flashlight** icon.

    The Identity Lookup dialog window appears.

21. Click **Lookup**. Note that it can take a few seconds for names to appear.

22. Select **Supervisor** and click **Select**.

    The **Supervisor** role appears in the **Selected group** field.

23. Click **OK**.

    Supervisor appears in the **Group** field.

24. Click **Next**.

25. Enter **1** in the **Number of levels** field.

26. Leave the remaining field blank on this window.

27. Click **Next**.

    The Routing policy window appears.

28. Select **ACCEPT** as the conclusion that results in the task being routed.

29. Leave the remaining field blank on this window.

30. Click **Next**.

    The Finish window displays your selections for workflow name, pattern, and outcome.

31. Click **Finish**.

    A **Scope** activity and **Switch** activity are automatically created.

    Both activities contain the details you entered in the Workflow Wizard.

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

Creating Assign Activities Inside the Switch Activity

> **Summary:** You now create an **Assign** activity that copies the payload back to its source.

1. Drag and drop an **Assign** activity from the **Component Palette** section to below the **copyPayloadFromTask Assign** activity in the **<case Task outcome is ACCEPT> section** of the **Switch** activity.

2. Double-click the **assign** icon to display the Assign window.

3. Click the **General** tab.

4. Enter **assignOrderApproval1** in the **Name** field.

5. Click **Apply**.
6. Click the **Copy Rules** tab.
7. Click **Create** to display the Create Copy Rule 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**	**string('Accepted')**
To	
▪ **Type**	**Variable**
▪ **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.

9. Click **OK** to close the Create Copy Rule window and the Assign window.
10. Repeat Steps 1 through 9 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')**
11. Repeat Steps 1 through 9 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')**
12. Drag and drop an **Assign** activity from the **Component Palette** section to below (outside) the **Switch** activity.
13. Double-click the **Assign** icon to display the Assign window.
14. Click the **General** tab.
15. Enter **setOutputVariable** in the **Name** field.
16. Click **Apply**.
17. Click the **Copy Rules** tab.

18. Click **Create** to display the Create Copy Rule window.
19. Enter the following details:

Field	Value
From	
▪ **Type**	**Variable**
▪ **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	
▪ **Type**	**Variable**
▪ **Variables**	Expand and select **Variables** > **outputVariable** > **payload** > **ns1:PurchaseOrder**

20. Click **OK** to close the Create Copy Rule window and the Assign window.

    When complete, the designer window looks as follows:

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

Validating, Compiling, and Deploying the Order Approval Process

1. Go to the **Applications Navigator** section.
2. Right-click **OrderApproval**.
3. Select **Deploy** > **LocalBPELServer** > **Deploy to default domain**.
4. Enter the domain password of **bpel** when prompted.

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 5 on page 2-15.

Running the Order Approval Process

1. Log into Oracle BPEL Console using Internet Explorer by selecting **Start** > **All Programs** > **Oracle -** *Oracle_Home* > **Oracle BPEL Process Manager 10.1.2** > **BPEL Console** (or refresh the page if it is already open).

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

2. Click **OrderApproval** in the **Deployed BPEL Processes** list.

3. 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. You must use a text editor to perform this task.

   a. Select **XML Source** from the **Initiating a test instance** list.

   b. Copy and paste the contents of `OrderBookingPO_1.xml` into the field that appears.

4. Click **Post XML Message**.

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

   ```
   Test Instance Initiated
   ```

5. Click the **Instances** tab at the top.

6. Click the **OrderApproval** instance.

   A message appears indicating that the instance is active.

7. Select **Start** > **All Programs** > **Oracle -** *Oracle_Home* > **Oracle BPEL Process Manager 10.1.2** > **Sample Worklist Application** to access the login window for Oracle BPEL Worklist Application:

8. Log in as `jcooper/welcome`.

   This displays Oracle BPEL Worklist Application. A task waiting to be approved appears.

9. Select **Acquire** in the **Actions** list for the task to approve.

10. 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.

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

State: Assigned		**Priority:** 3		**Assignees:** Supervisor (G)	
Sub State:		**Creator:** bpeladmin		**Process:** OrderApproval	
Conclusion:		**Created:** May 8, 2005 2:06 PM		**Owner:** bpeladmin	
Expiration: May 9, 2005 12:06 AM		**Modified:** May 8, 2005 2:08 PM		**Task Key:**	
Acquirer: jcooper		**Modifier:** jcooper		**Task Number:** 10000	
Pattern: Sequential Workflow					Update Fields...

Customer Id	Antilla Electronics
Id	AS-0193
Order Date	2005-01-10
Order Price	6100.00
Order Status	unknown
Order Comments	...
Ship To	DeGossaa Calro 12, Burhon Street Danzig Godanzk PO-20398 Poland
Bill To	Mrs Castro 23, Uudemankatu Hyvinkaa Finland SF-05800 Finland

User Contact

Phone Number	01-358-918-89043
Phone Number	HansonRep@semiconductors.po

Supplier Info

Supplier Price	0.0
Supplier Name	Unknown

Items

Product Name	Item Type	Part Number	Price	Quantity
Grundi TV	Audio-Video	239-FRH	900.00	4
Sony Camcorder	Entertainment	TRV-32H	500.00	5

12. Select **Accept** from the **Task Action** list and click **Go**.
13. Log out as user `jcooper`.
14. Log into Oracle BPEL Worklist Application as `jstein/welcome`.
15. Select **Accept** from the **Actions** list and click **Go**.

    After processing, no tasks appear in Oracle BPEL Worklist Application.
16. Return to Oracle BPEL Console.
17. Click the **Instances** tab at the top.
18. Click the **OrderApproval** instance.

    A message appears indicating that the instance has completed.
19. 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. Ensure that **Process Activities** is selected in the drop-down list of the **Component Palette** section.
4. Drag and drop a **PartnerLink** activity 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 **WSIL Browser**) to access the WSDL Chooser window for automatically selecting the **OrderApproval** service.

Field	Value
Name	**POAService**
WSDL File	Access this URL by clicking the **WSIL Browser flashlight** icon and expanding and selecting **LocalBPELServer** > **processes** > **default** > **OrderApproval**. **http://localhost:9700/orabpel/default/OrderApproval/OrderApproval?wsdl** **See Also:** "Setting the Hostname in Your JDeveloper BPEL Designer Web Browser Preferences" on page 1-5 if you receive a parsing error when attempting to add a WSDL file in the WSDL Chooser window.
Partner Link Type	**OrderApproval**
My Role	**OrderApprovalRequester**
Partner Role	**OrderApprovalProvider**

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 **assignPOADataIn** in the **Name** field of the **General** tab.
4. Click **Apply**.
5. Click the **Copy Rules** tab.

6. Click **Create** to display the Create Copy Rule 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 Rule 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 Rules** tab.
6. Click **Create** to display the Create Copy Rule 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**

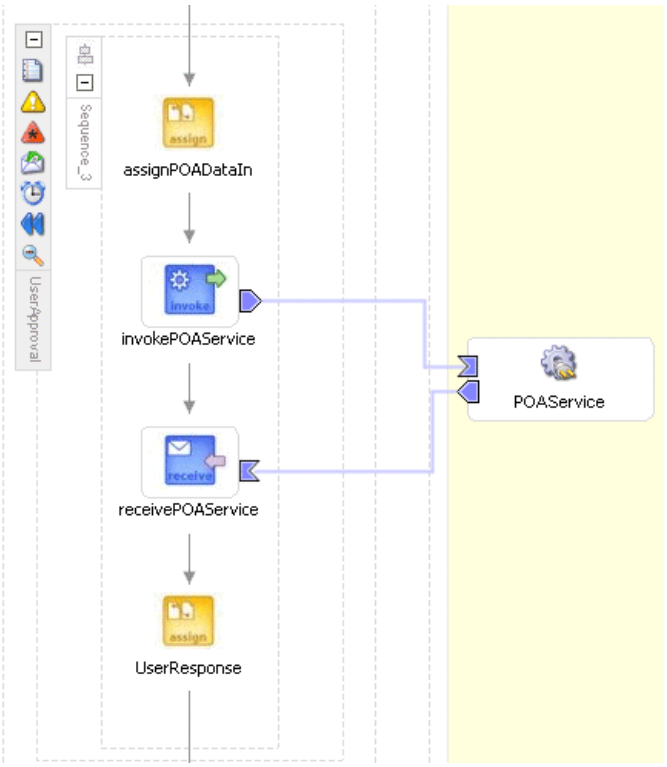
8. Click **OK**.
9. Click **Create** in the **Copy Rules** tab to again display the Create Copy Rule window.
10. Enter the following details to create a second statement:

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 Rule 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 **Applications Navigator** section.
2. Right-click **OrderBooking**.
3. Select **Deploy** > **LocalBPELServer** > **Deploy to default domain**.
4. Enter the domain password of **bpel** when prompted.
5. Increment the version number of the project when prompted (for example, enter **1.7**) and click **OK**.

   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 5 on page 2-15.

Running and Verifying the Order Process

1. Log into Oracle BPEL Console using Internet Explorer by selecting **Start** > **All Programs** > **Oracle -** *Oracle_Home* > **Oracle BPEL Process Manager 10.1.2** > **BPEL Console** (or refresh the page if it is already open).
2. Click the **BPEL Processes** tab.
3. Note which **OrderBooking** version has an asterisk next to the version number. This is the default version.
4. 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.
5. Scroll down and click **Mark as Default** to make this the default version of **OrderBooking**.
6. 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.
7. If you have installed Patch 1, go to Step 8. Otherwise, you must first perform Steps 7a through 7b on page 9-10. See Chapter 1, "Introduction" for information about where to obtain Patch 1.
8. Click the **Dashboard** tab.
9. 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**.
10. Start the correct **OrderBooking** version instance in one of the following ways:
    - Copy `Oracle_Home\integration\orabpel\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**.

– Select **XML Source** from the **Initiating a test instance** list. 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.

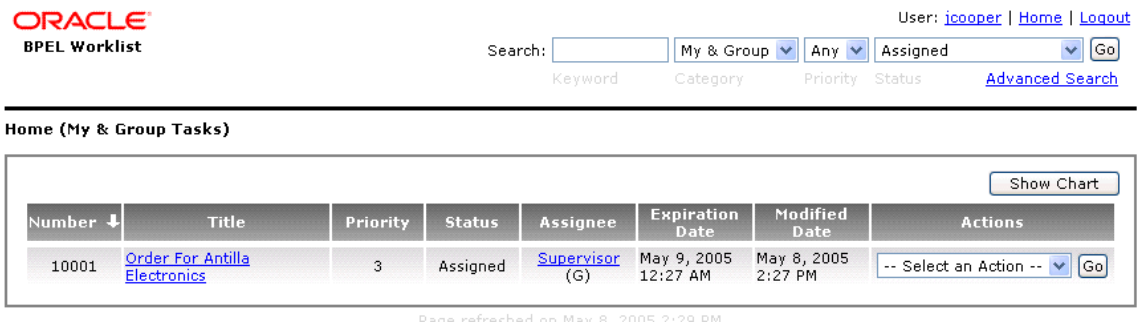
– Open the provided **CreateOrderBookingUI** application at `http://localhost:9700/CreateOrderBookingUI` and provide inputs to initiate a process.

11. Verify that the **OrderBooking** process has started.
12. Provide a price quote for **SelectManufacturing** from the following graphical user interface:

    `http://localhost:9700/SelectManufacturingUI`

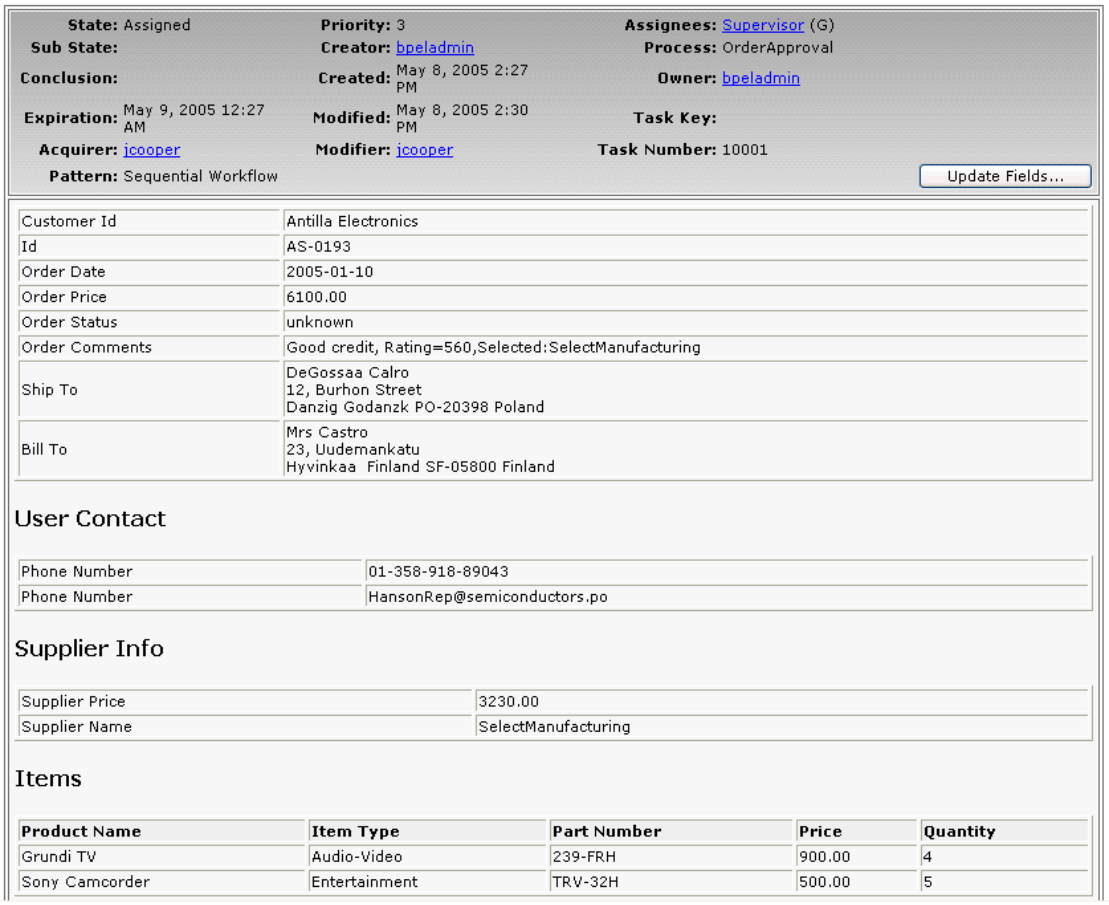
    After providing a quote, watch the process instance execute and invoke the user approval service at **OrderApproval**. Go to Oracle BPEL Console and verify that the **OrderApproval** process and **TaskActionHandler** process have both started.
13. Select **Start** > **All Programs** > **Oracle - *Oracle_Home*** > **Oracle BPEL Process Manager 10.1.2** > **Sample Worklist Application** to access the login window for Oracle BPEL Worklist Application.
14. Log in as `jcooper/welcome`. Default users are provided at install time in the following configuration file:

    Oracle_Home`\integration\orabpel\system\appserver\oc4j\j2ee\home\config\jazn-data.xml`



The list shows the task awaiting user actions.

15. Select **Acquire** in the **Actions** list and click **Go** to acquire a task.
16. View the task details, the comments section, and so on.



State: Assigned | **Priority:** 3 | **Assignees:** Supervisor (G)
Sub State: | **Creator:** bpeladmin | **Process:** OrderApproval
Conclusion: | **Created:** May 8, 2005 2:27 PM | **Owner:** bpeladmin
Expiration: May 9, 2005 12:27 AM | **Modified:** May 8, 2005 2:30 PM | **Task Key:**
Acquirer: jcooper | **Modifier:** jcooper | **Task Number:** 10001
Pattern: Sequential Workflow | Update Fields...

Customer Id	Antilla Electronics
Id	AS-0193
Order Date	2005-01-10
Order Price	6100.00
Order Status	unknown
Order Comments	Good credit, Rating=560,Selected:SelectManufacturing
Ship To	DeGossaa Calro 12, Burhon Street Danzig Godanzk PO-20398 Poland
Bill To	Mrs Castro 23, Uudemankatu Hyvinkaa Finland SF-05800 Finland

User Contact

Phone Number	01-358-918-89043
Phone Number	HansonRep@semiconductors.po

Supplier Info

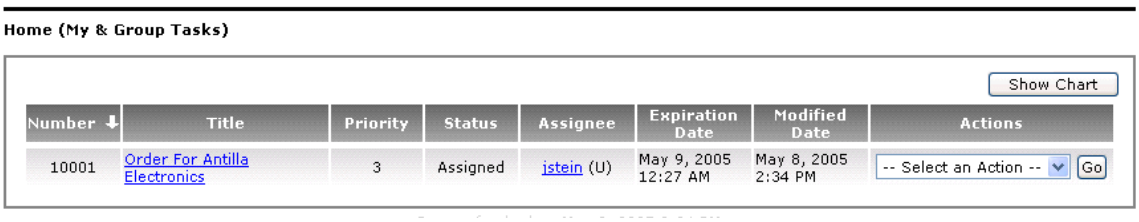
Supplier Price	3230.00
Supplier Name	SelectManufacturing

Items

Product Name	Item Type	Part Number	Price	Quantity
Grundi TV	Audio-Video	239-FRH	900.00	4
Sony Camcorder	Entertainment	TRV-32H	500.00	5

17. Approve or reject the task by selecting **Accept** or **Reject** from the **Task Action** list, and click **Go**.
18. Log out of user `jcooper`. Rejects go directly back to the process, while approved tasks require the next level of approval.
19. Log in as `jstein/welcome` (the default manager of `jcooper` in the hierarchy defined in the `jazn-data.xml` configuration file listed in Step 14).

    The list shows the task awaiting user actions.



Home (My & Group Tasks)

Show Chart

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

20. View the task details, the comments section, and so on.
21. 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.
22. View Oracle BPEL Console, 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 Console.
- 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 Console.
- Accessed the Oracle BPEL Worklist Application URL to approve the order.

12

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

This phase of the tutorial adds to the process you designed in Chapter 11, "Designing the Human 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) queues. 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 Console.
- 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 **Applications 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:

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 **Create**.

    The Create Sensor Action window appears.
14. Enter **ReportsPublisher** in the **Name** field.
15. Select **Database** from the **Publish Type** list.
16. Click **OK**.

    The Sensor Action Chooser window appears.
17. Select **ReportsPublisher** and click **OK** to return to the Create Activity Sensor window.
18. Click **Add** in the **Sensor Actions** section.

    The Sensor Action Chooser window appears.
19. Select **Sensor Actions** and click **Create**.

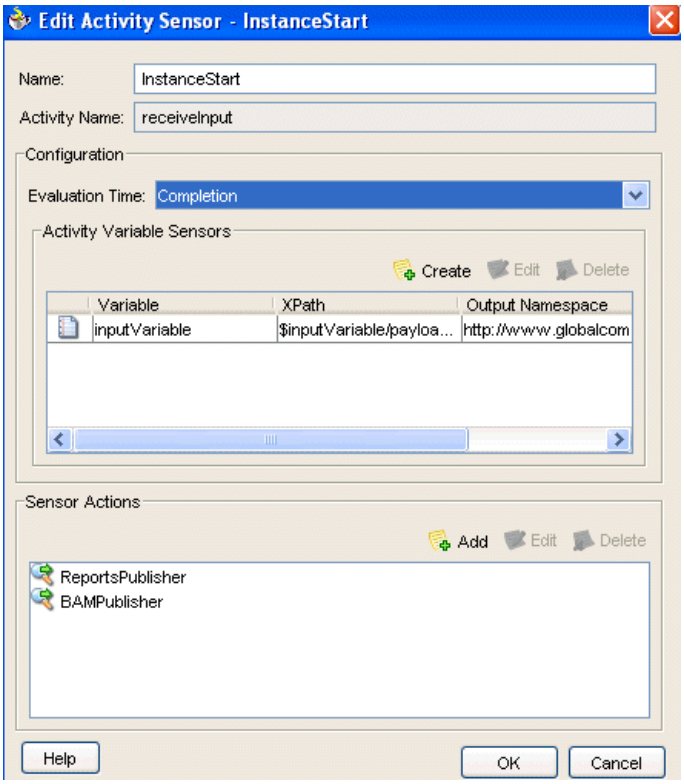
    The Create Sensor Action window appears.
20. Enter the following details.

Field	Description
Name	**BAMPublisher**
Publish Type	**JMS Queue**
JMS Connection Factory	**jms/QueueConnectionFactory**
Publish Target	**jms/demoQueue**
Filter	Leave unselected.
Enable	Leave selected.

21. Click **OK**.

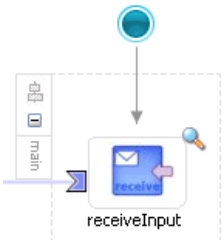
    The Sensor Action Chooser window appears.
22. Select **BAMPublisher** and click **OK** to return to the Create Activity Sensor window.

    The Create Activity Sensor window appears as follows:



23. Click **OK** in the Create Activity Sensor window and the Receive window.
24. Select **Save** from the **File** main menu.

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



25. Repeat the steps in this section to create additional sensors on the following **OrderBooking** process 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**

Create Sensor on Activity...	Name to Give Sensor...	Evaluation Time To Select...	Variable XPath To Select...
callbackClient	InstanceCompletion	Completion	$inputVariable/payload

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

Running, Validating, and Deploying the OrderBooking Process

1. Log into Oracle BPEL Console using Internet Explorer by selecting **Start** > **All Programs** > **Oracle - *Oracle_Home*** > **Oracle BPEL Process Manager 10.1.2** > **BPEL Console** (or refresh the page if it is already open).
2. Click the latest **OrderBooking** instance process to undeploy in the **Dashboard** tab.
3. Click the **Manage** link.
4. Click **Undeploy** at the bottom of the window.
5. Click **OK** when prompted to mark all open instances for this process as stale.
6. Click **Done** when prompted.
7. Go to the **Applications Navigator** section.
8. Right-click **OrderBooking**.
9. Select **Deploy** > **LocalBPELServer** > **Deploy to default domain**.
10. Increment the version number of the project when prompted and click **OK**.

    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 5 on page 2-15.

Running the Order Booking Process

1. Log into Oracle BPEL Console using Internet Explorer by selecting **Start** > **All Programs** > **Oracle - *Oracle_Home*** > **Oracle BPEL Process Manager 10.1.2** > **BPEL Console** (or refresh the page if it is already open).
2. Click the **BPEL Processes** tab.
3. Note which **OrderBooking** version has an asterisk next to the version number. This is the default version.
4. 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.
5. Scroll down and click **Mark as Default** to make this the default version of **OrderBooking**.
6. 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.
7. If you have installed Patch 1, go to Step 8. Otherwise, you must first perform Steps 7a through 7b on page 9-10. See Chapter 1, "Introduction" for information about where to obtain Patch 1.
8. Click the **Dashboard** tab.
9. Verify that the **OrderBooking** instance appears under the **Dashboard** tab.

10. Click the **OrderBooking** instance.
11. Select **XML Source** from the **Initiating a test instance** list.
12. 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. You must use a text editor to perform this task.

13. Click **Post XML Message**.
14. Click **Audit Instance**.

    Note that the **SelectManufacturing** process is pending and waiting for a price quote.
15. Provide a price quote for **SelectManufacturing** from the following graphical user interface:

    `http://localhost:9700/SelectManufacturingUI`
16. Refresh Oracle BPEL Console and note that the process is pending on **POAService**.
17. Select **Start** > **All Programs** > **Oracle -** *Oracle_Home* > **Oracle BPEL Process Manager 10.1.2** > **Sample Worklist Application** to access the login window for Oracle BPEL Worklist Application.
18. Log in as `jcooper/welcome`.

    The list shows the task awaiting user actions.
19. Select **Acquire** in the **Actions** list and click **Go** to acquire a task.
20. Select **Accept** from the **Task Action** list, and click **Go**.
21. Log out as user `jcooper`.
22. Log in as `jstein/welcome` (the default manager of `jcooper` in the hierarchy).
23. Select **Accept** in the **Task Action** list, and click **Go**.
24. Refresh Oracle BPEL Console and confirm that the instance has completed.
25. 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 Console.
- Observed the sensor results.

13

Using Notifications

This chapter of the tutorial describes how to add notifications.

This chapter contains the following topics:

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 Console 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 a Notification Activity
- Validating, Compiling, and Deploying the POAcknowledge Process
- Running the POAcknowledge Process

Configuring Your Company E-Mail Environment

1. Go to the *Oracle_Home*\integration\orabpel\system\services\config directory.
2. Make a backup copy of the ns_emails.xml file.
3. Open a text editor.
4. Open ns_emails.xml and replace the values in bold with ones appropriate to your company environment.

```
<EmailAccounts xmlns="http://xmlns.oracle.com/ias/pcbpel/NotificationService">
   <EmailAccount>
      <Name>Default</Name>
      <GeneralSettings>
         <FromName>Oracle BPM</FromName>
         <FromAddress>bpm1@sunbox4254.us.mycompany.com</FromAddress>
      </GeneralSettings>
      <OutgoingServerSettings>
         <SMTPHost>sunbox4254.us.mycompany.com</SMTPHost>
         <SMTPPort>225</SMTPPort>
      </OutgoingServerSettings>
      <IncomingServerSettings>
         <Server>sunbox4254.us.mycompany.com</Server>
         <Port>2110</Port>
         <Protocol>pop3</Protocol>
         <UserName>bpm1</UserName>
         <Password ns0:encrypted="false"
xmlns:ns0="http://xmlns.oracle.com/ias/pcbpel/NotificationService">welcome</Pas
sword>
         <UseSSL>false</UseSSL>
         <Folder>Inbox</Folder>
```

```
        <PollingFrequency>1</PollingFrequency>
        <PostReadOperation>
           <MarkAsRead/>
        </PostReadOperation>
     </IncomingServerSettings>
  </EmailAccount>
```

5. Save your changes and exit the file.

Creating a Notification Activity

1. Double-click **POAcknowledge.bpel** in the **Applications Navigator**.
2. Drag and drop a **Notification** activity from the **Component Palette** section to below the **InvokeFileOut Invoke** activity.

   This starts the Notification Wizard.
3. Click **Next** on the Welcome page.
4. Select **Email** on the Select a Notification channel window.
5. Enter the following details:

> **Note:** Instead of manually entering an expression, you can press **Ctrl** and then the space bar in the **Body** field. Scroll through the list of values that appear 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	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:	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. **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.'))**

6. Click **Next**.
7. Click **Finish**.

Validating, Compiling, and Deploying the POAcknowledge Process

1. Go to the **Applications Navigator** section.
2. Right-click **POAcknowledge**.
3. Select **Deploy** > **LocalBPELServer** > **Deploy to default domain**.
4. Enter the domain password of **bpel** when prompted.
5. Increment the version number again of the project when prompted and click **OK** (for example, enter **1.1**).

   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 5 on page 2-15.

Running the POAcknowledge Process

1. Log into Oracle BPEL Console using Internet Explorer by selecting **Start** > **All Programs** > **Oracle -** *Oracle_Home* > **Oracle BPEL Process Manager 10.1.2** > **BPEL Console** (or refresh the page if it is already open).

   The **Dashboard** tab of Oracle BPEL Console appears. Note that your BPEL process, **POAcknowledge**, now appears in the **Deployed BPEL Processes** list.
2. Click the **POAcknowledge** version you increased in Step 5 on page 13-4 in the **Deployed BPEL Processes** list.
3. Enter details in all fields of the HTML Form and click **Post XML Message**.
4. Refresh Oracle BPEL Console.
5. Click the **Instances** tab.
6. Click the **POAcknowledge** instance.
7. Click the **Audit** link to view progress.
8. 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 acknowledgement through e-mail notification.

Index

A

B

C

D

E

F

H

I

J

K

L

M

N

O

P

R

S

T

U

V

W

X