1 End-2-End Purchase Order Processing Quickstart

1.1 Introduction

This Quick Start tutorial for the End-2-end 105 PO Processing sample shows you how to install the pre-built sample. To build the sample from scratch, see the full step-by-step documentation for this sample.

In this sample we install a SOA composite application to process and approve purchase orders. The purchase order details can come in from any source (in our case a testing page), the credit card status for the customer is validated and if the credit card is good, the order continues. An order for a large purchase price requires a manual approval step. Finally, the order is written to a text file to be processed by the fulfillment house.

This SOA composite application contains a database adapter, web services binding, Mediator ESB routing service, BPEL process and human task.

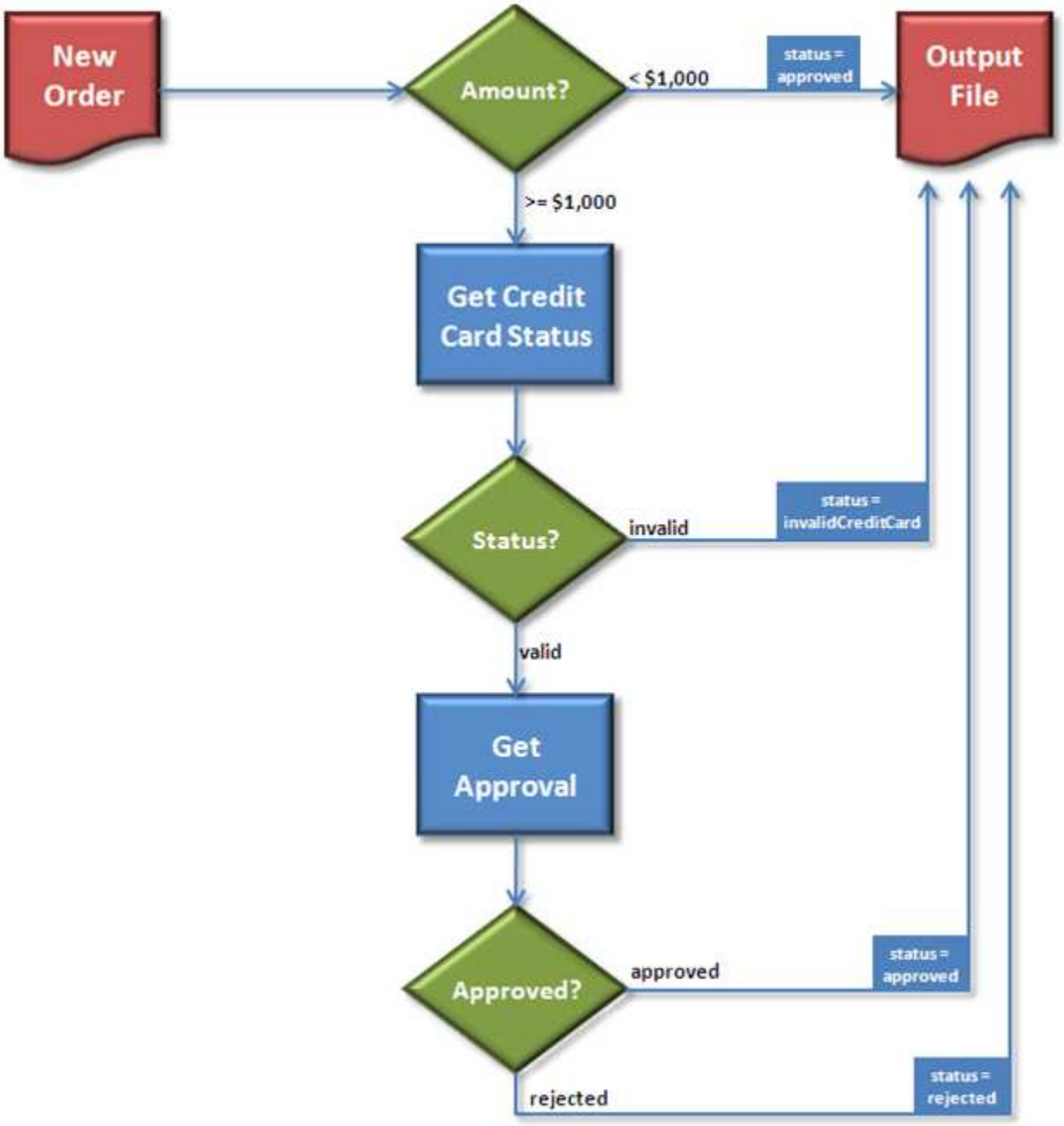
1.2 Description of the application

The application you will install is the back-end processing of a new order. The business process is as follows.

- All orders will be written to a text file to be processed by the fulfillment house.
- Small orders (defined as those under $1,000) are to be automatically approved.
- Large orders (those greater than or equal to $1,000) go through a validation and approval process.
  - The customer's credit card must be validated.
  - A customer service representative must manually approve the order.
- Approved orders shall have the status "approved".
- Large orders with an invalid credit card shall have the status "invalidCreditCard".
- Large orders rejected by the customer service representative shall have the status "rejected".

Figure 1 shows a visual representation of the business process.

Figure 1 Visual view of the business process



1.2.1 SOA Components

In terms of the SOA application there are the following elements

- A service that accepts new orders in XML format.
- A File Adapter service that can write XML messages (such as orders) out to a file.
- A content-based Mediator service to route small orders to the file adapter and large orders to a BPEL process.

- A BPEL process that validates the credit card by calling a service, and seeks human approval using a Human Task component.
- The credit card validation service used by the BPEL process to validate a given credit card.
- A Database Adapter service to retrieve the status of a given credit card from the database.

1.3 Prerequisites

This document assumes you have already installed the following:

- An Oracle database
- Oracle JDeveloper 11
- Oracle SOA Suite 11 (may be bundled with JDeveloper)

1.4 Project Files

The ZIP file for this sample came with several directories and files:

- `doc` – contains the instructions for creating and running the demo
- `input` – a few xml files with sample input data
- `sql` – two database scripts used during setup
- `schemas` – xsd files used throughout sample when defining services
- `solutions` – a solution project for each chapter

1.4.1 File Conventions

After unzipping the file for this sample, move or copy it to `c:\po`. This document assumes that path. If you unzipped somewhere else then adjust accordingly when `c:\po` is referenced through this document.

1.4.2 Installing the database schema

This demo requires a table in the database.

1. Create the `soademo` user. It is ok to run this script even if the `soademo` user already exists. From a command line, cd to the `c:\po\sql` directory and run the following replacing `pw` with your own system user's password:

```
cd c:\po\sql
sqlplus system/pw @create_soademo_user.sql
```

2. Create the credit card info table. It is ok to run this script more than once, even if the table already exists.

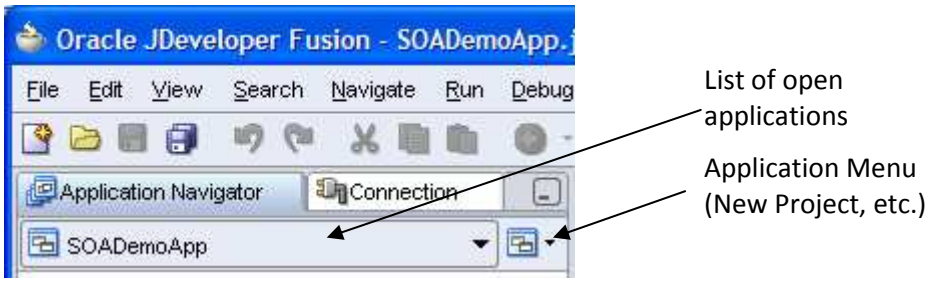
```
sqlplus soademo/soademo @create_creditrating_table.sql
```

1.5 Deploying

Open and deploy the Credit Card Validation and the PO Processing applications.

1. In JDeveloper, select `View > Application Navigator`.
   If there are no currently open applications in JDeveloper you will see two options in the Application Navigator window: `New Application` and `Open Application`. Select `Open Application`.

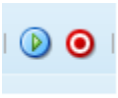
   Otherwise you will see an open application listed in the `Application Navigator` window and you will need to click on the open applications list and select `Open Application`.



2. In the dialog, browse to
   `c:\po\105-end2end-PO-Processing\Ch2\CreditCardValidation`
   and open
   `CreditCardValidation.jws`

Now, start the server if it is not already started and deploy the Credit Card Validation application using the `Deploy` command.

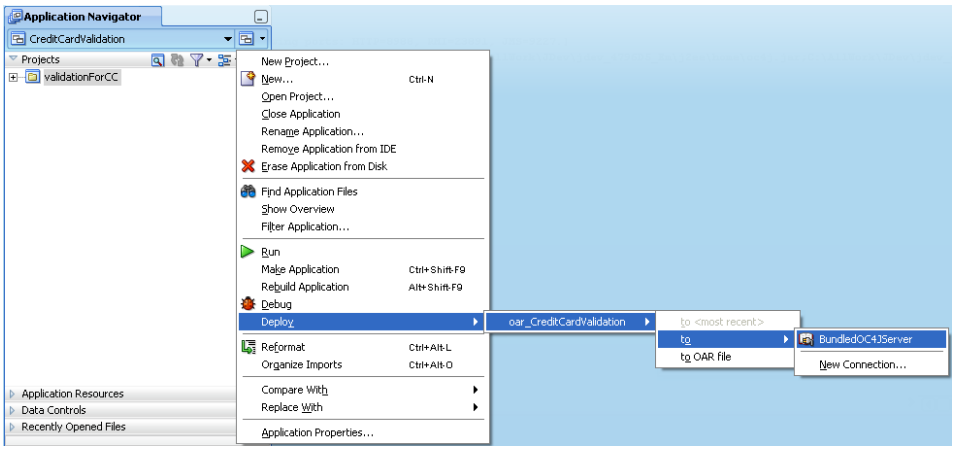
3. Select the Run button in the toolbar to start the server. This button is enabled when an application is open in JDeveloper.



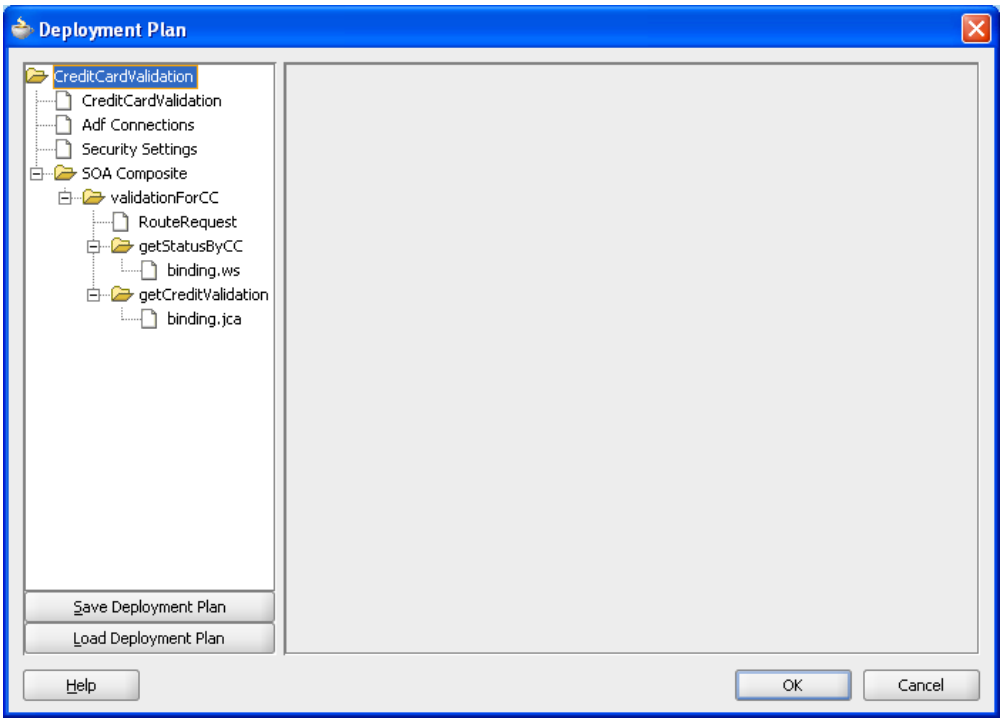
Wait for the server to finish starting before going on to the next step. The server is completely started when it posts the message, `Done deploying composites` in the log window.

4. Select the application name in the application navigator.

5. Then, in the application menu (see picture above to find and select the correct menu), select `Deploy…to…BundledOC4J`. Make sure you are using the application menu and not the project menu.



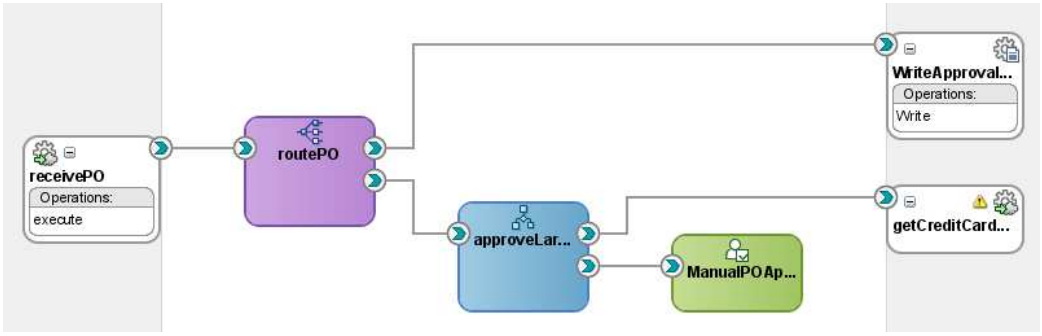
The deploy command will first compile the composite and then open the Deployment Plan.



6. Select `OK` and wait for the application to finish deploying. You will see messages in the server log when this is complete.

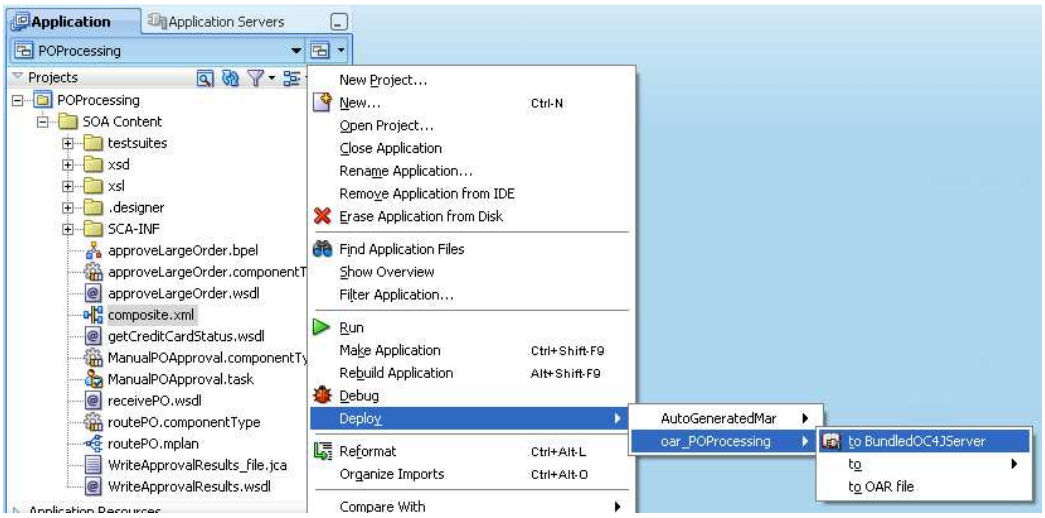
The next step is to deploy the PO Processing application. This application requires the path to the `WSDL` file of the Credit Card Validation application so the server must be running for it to find the `WSDL`.

7. Open the `POProcessing` Application just the same way you previously opened the Credit Card Validation application and expand the application to see the files within the application. Double-click on `composite.xml` to open that file.



8. You may or may not see the warning icon on the `getCreditCard` web service reference. If you see this warning, make sure your server is currently started as in step 3 above. The warning icon means that JDeveloper can't find the `WSDL` file referenced in the web service reference. When you have resolved the warning, you can deploy the PO Processing application.

9. Using the application menu, select `Deploy….BundledOC4J`.

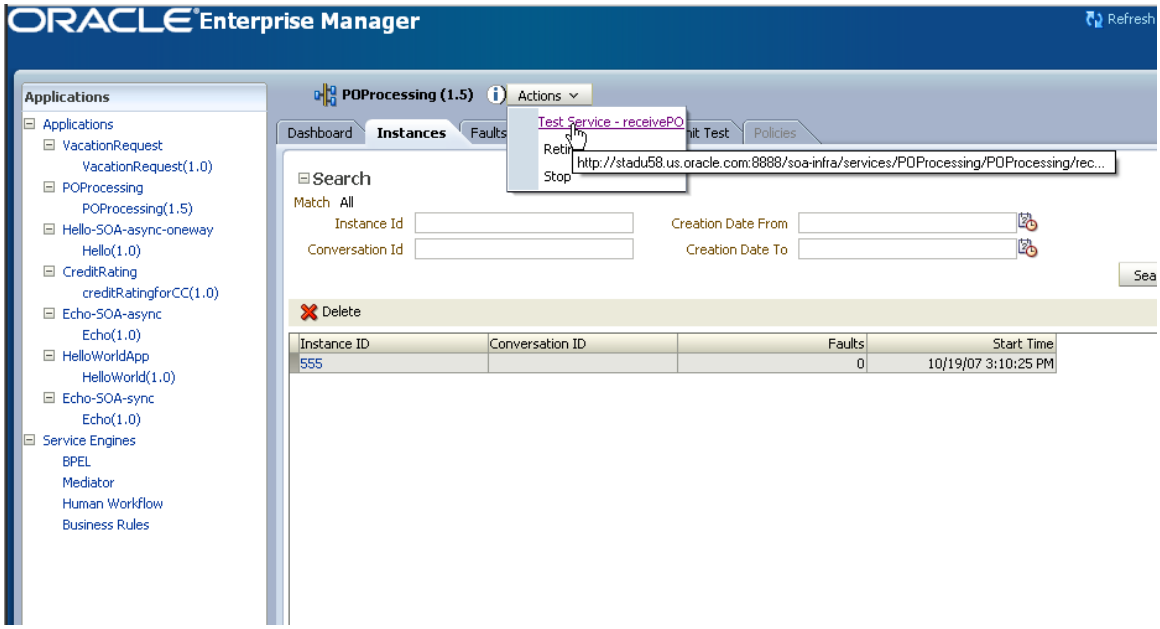


10. When you see the Deployment Plan dialog, select OK and wait for the application to finish deploying.

1.6 Running the application

11. Once the applications are deployed, you are ready to try running them. Open the SOA Console at `http://localhost:8988/SOAConsole`

12. Click on **POProcessing** and in the **Actions** list choose to test your service.

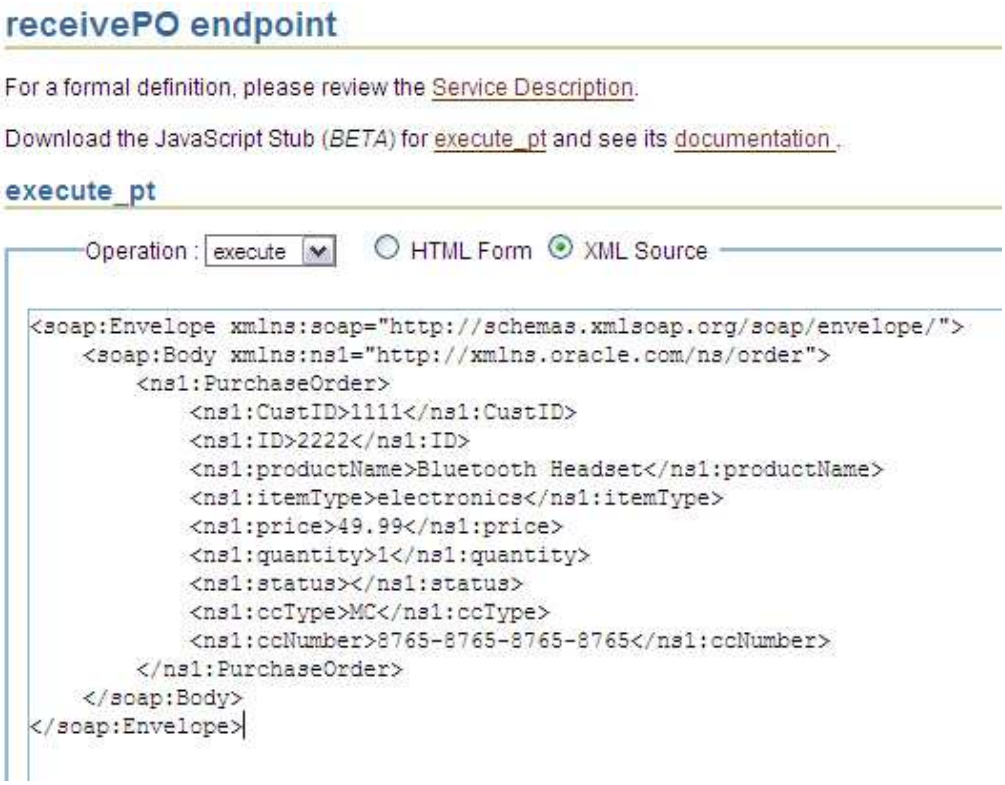


13. Enter a small order by doing one of the following:

- Type the values into in to the HTML form, or
- Click **XML Source** so you can paste in the XML payload. This is the recommended way. Open the following file in a text editor:

  `c:\po\input\po-small-Headsetx1.xml`

  Copy the entire contents and paste them into the large text field in your browser:



14. Click **Invoke**.

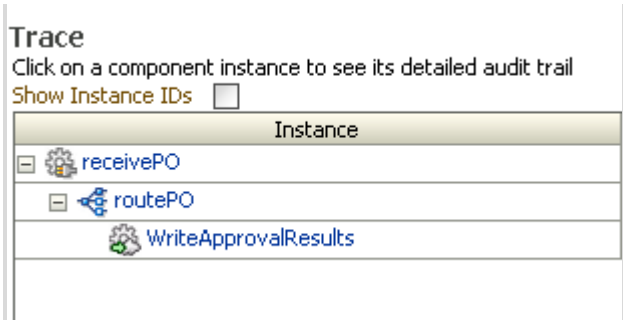
15. The **Test Result** screen won't have any response because this is a one-way invocation with no reply or callback. However the call to the File Adapter service will have resulted in a new file in `c:\temp`, called `order_n.txt`, where *n* is a sequence number like 1, 2, 3, etc.

    You can open that file with a text editor and examine it (as shown in Figure 2). Notice how field names have been translated by the mapping and are different from the input.

Figure 2 Output from order_n.txt file

```
<?xml version="1.0" ?><inp1:Order xmlns:inp1="http://xmlns.oracle.com/ns/order">
   <inp1:customerId>1111</inp1:customerId>
   <inp1:orderId>2222</inp1:orderId>
   <inp1:prodName>Bluetooth Headset</inp1:prodName>
   <inp1:itemType>electronics</inp1:itemType>
   <inp1:price>49.99</inp1:price>
   <inp1:qty>1</inp1:qty>
   <inp1:status>approved</inp1:status>
   <inp1:creditCardInfo>
      <inp1:cardNumber>8765-8765-8765-8765</inp1:cardNumber>
      <inp1:cardType>MC</inp1:cardType>
   </inp1:creditCardInfo>
</inp1:Order>
```

16. You can also return to the SOA Console, click on the application or click the **Refresh** link. In the **Last 5 Instances** section click the most recent instance id to see the execution flow.



That's it! You have now installed and run the PO Processing application.