

A Deploying and Running a Composite Application

A.1	Introduction.....	3
A.2	Deploying the application.....	3
A.2.1	Deploying to the integrated container in JDeveloper.....	3
A.2.2	Deploying to an external container	6
A.3	Running and testing the application.....	13
A.3.1	Getting the service description (WSDL)	20

A.1 Introduction

A SOA composite application has to first be deployed to an application server container, and then run from there. Deploy your application by following the instructions in **A.2 Deploying the application**. Once it is deployed, follow the instructions in or **A.3 Running and testing the application**.

A.2 Deploying the application

You have two choices about which container to deploy to:

1. The integrated container in JDeveloper
2. An external or remote container that's started independently

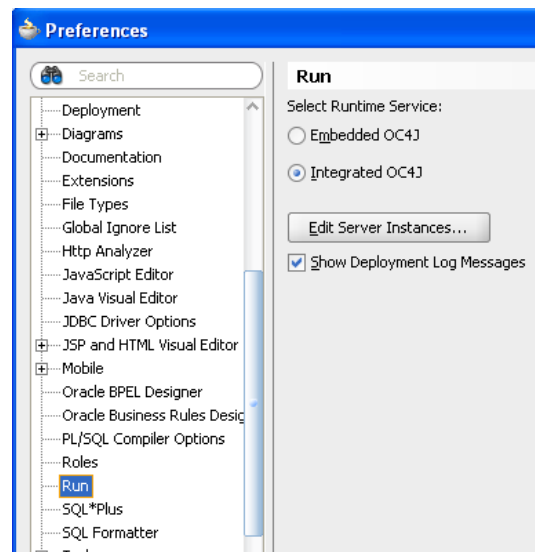
Determine which method you will be using, then follow the directions in either **A.2.1 Deploying to the integrated container in JDeveloper** or **A.2.2 Deploying to an external container**.

A.2.1 Deploying to the integrated container in JDeveloper

When you use the Integrated OC4J with JDeveloper, you start the SOA server from within JDeveloper as part of your deploy action. The SOA server should have already been configured before deploying. If you have not completed the Configure SOA step, please see the installation instructions.

There are two important things to remember when using the Integrated OC4J.

1. Be sure to set your server preference to Integrated OC4J. Use the Tools/Preference dialog, select Run in the left navigation bar and select Integrated OC4J.



2. When re-deploying your application, always undeploy first (see instructions below).

After you have created your own composite, you can deploy it to the server. There are two ways to deploy as described below.

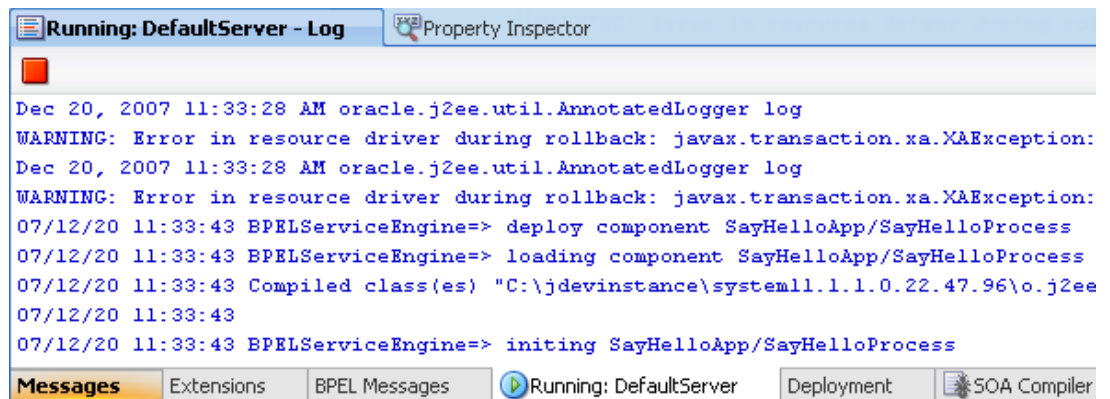
1. Deploy method A – [Run](#)

This method rebuilds the app in memory and does not write the binary to disk. It uses the existing (defaulted) deployment plan. This method is for quick turnaround during development. It does not leave the application deployed on the SOA server when the run command terminates.

- a. Select `composite.xml` in your application and then select the [Run](#) button (green arrow) in the toolbar. The [Run](#) command starts the server if it is not already started and deploys the application using the existing deployment plan.



- b. Watch the messages in the logs until you see the server log show that your application was initialized.



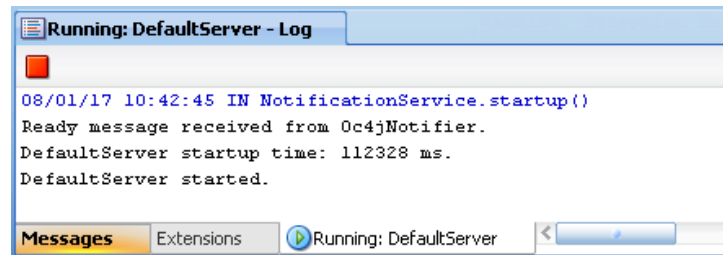
3. Deploy method B – [Deploy](#)

This method rebuilds the app and also writes the binary to disk. It also brings up the deployment plan dialog before deploying. This method is used when you want the application to stay deployed, for example when you are deploying a set of applications that work together. When the server is stopped and restarted, the applications will still be deployed.

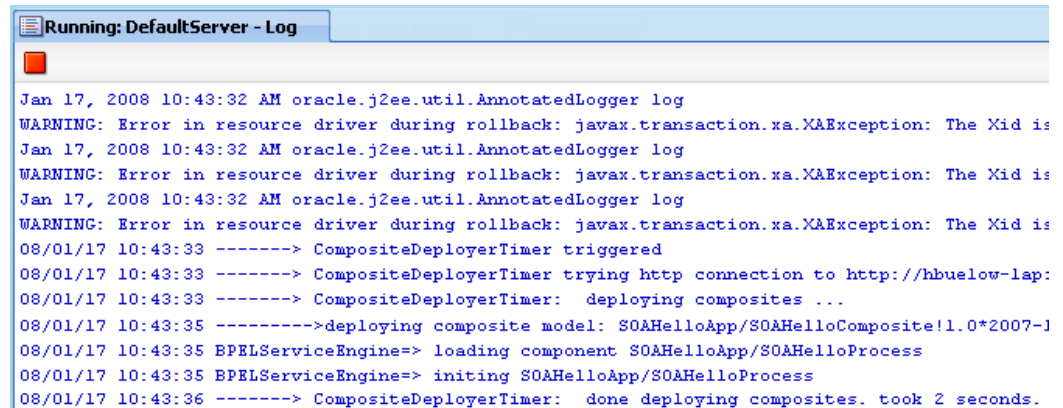
- a. If the server is not started, press the [Start Server Instance](#) button on toolbar.



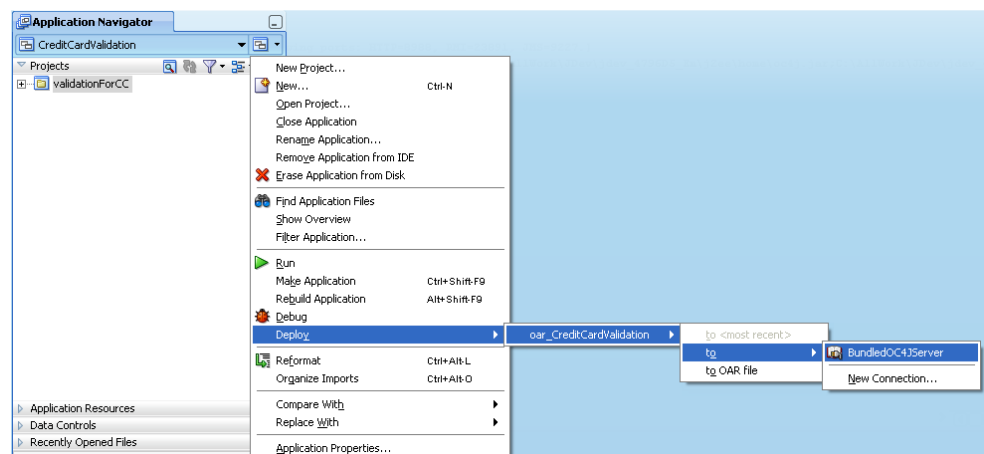
This button is disabled if you do not have an application open in JDeveloper. Wait for the server to completely startup.



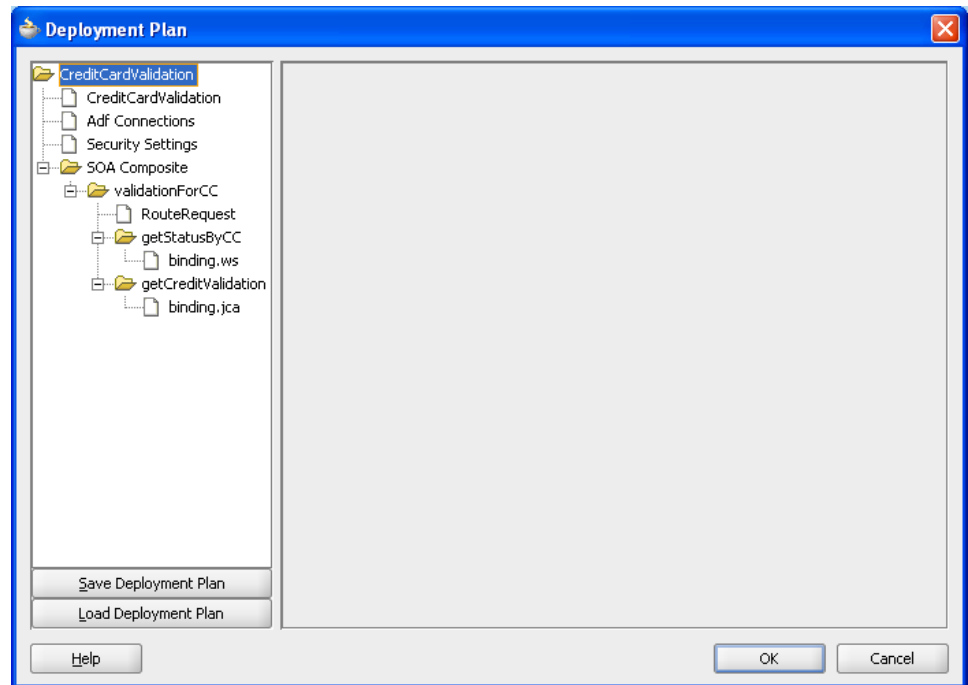
After you see, “DefaultServer started” wait a little more and you will see the final startup message “done deploying composites.” The SOA infrastructure is now initialized and you can deploy your own composite.



- b. In the application menu - the arrow to the right of the application name - select **Deploy** and follow the menu to select **BundledOC4JServer**. Make sure you have the application menu and not the project menu in order to see this option.



- c. First the application is built and then the deployment plan dialog opens. When you see the **Deployment Plan** dialog, select **OK** to continue with deployment.



After deployment finishes, you may not see messages in the server log. You verify that your application is deployed by opening the SOA Console (see section A.3).

4. Redeploy your application

If you make changes and want to redeploy your application, you must undeploy the previous deployment first using method A or B here as appropriate and then follow steps above to deploy your application again.

a. Undeploy method A - [Run](#)

- If you used the [Run](#) command to deploy your application, simply terminate the Run by selecting the red box in the top of the run application log window. This undeploys the application.

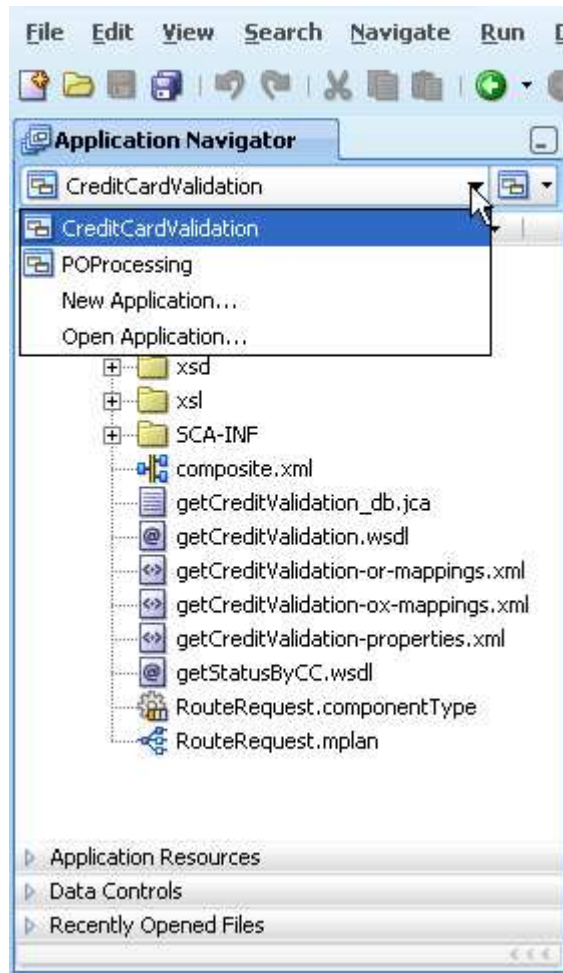
b. Undeploy method B – [Undeploy](#)

- If you used the [Deploy](#) command to deploy your application, open the [Application Server Navigator](#) from the [View](#) menu and navigate to the application.
- Right-click your application and select the [Undeploy](#) command

When your application is deployed you can run and test it. Go to section A.3 to see how to run your application.

A.2.2 Deploying to an external container

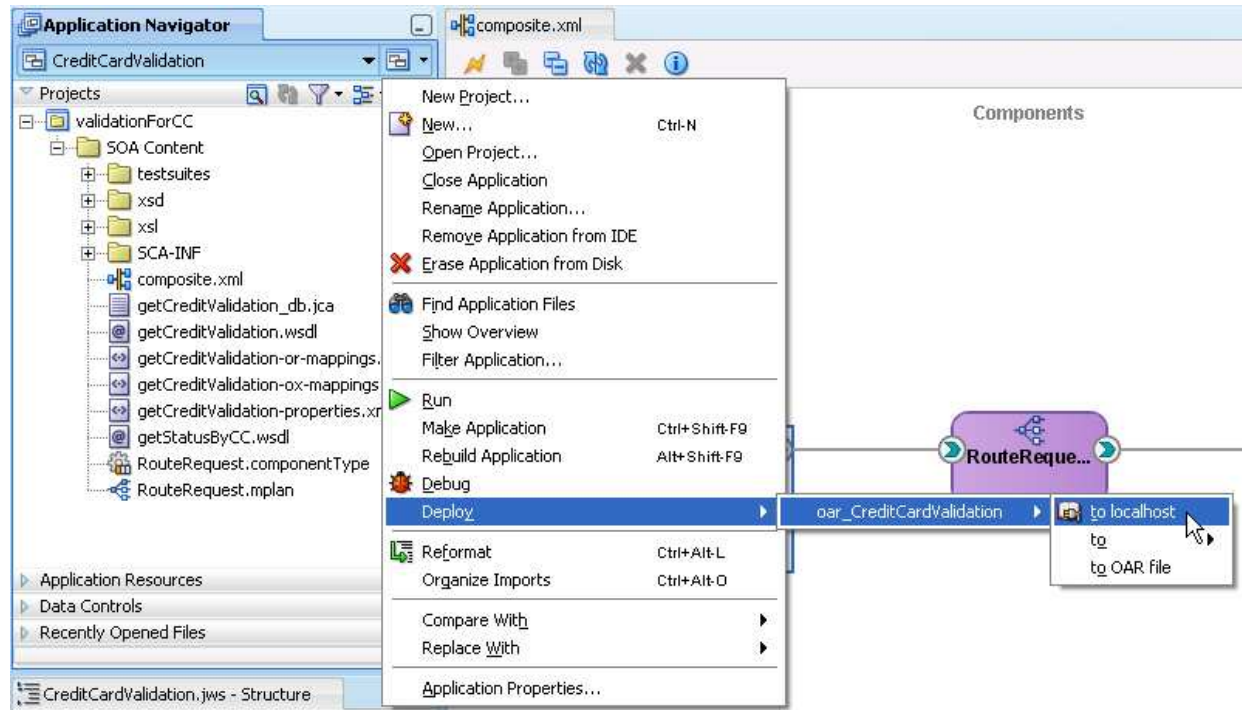
1. Before deploying, ensure the application server is up and running.
2. In JDeveloper, select the application you want to deploy from the application menu.



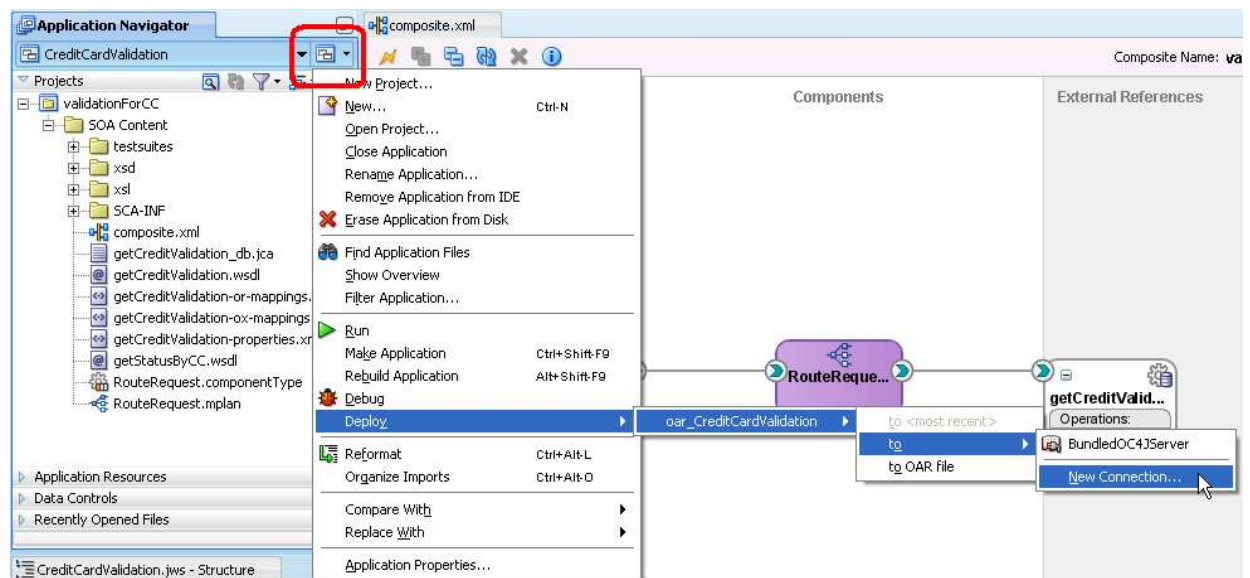
3. If this is the very first time you are deploying this application, continue from step 5 to create a new connection to the target application server. Once it's done the first time you can reuse that connection when deploying in the future.

If you have already created a connection previously, click the down arrow button on the application menu and deploy to your connection (as shown in Figure 1), and then to skip to step 13 on page A-11.

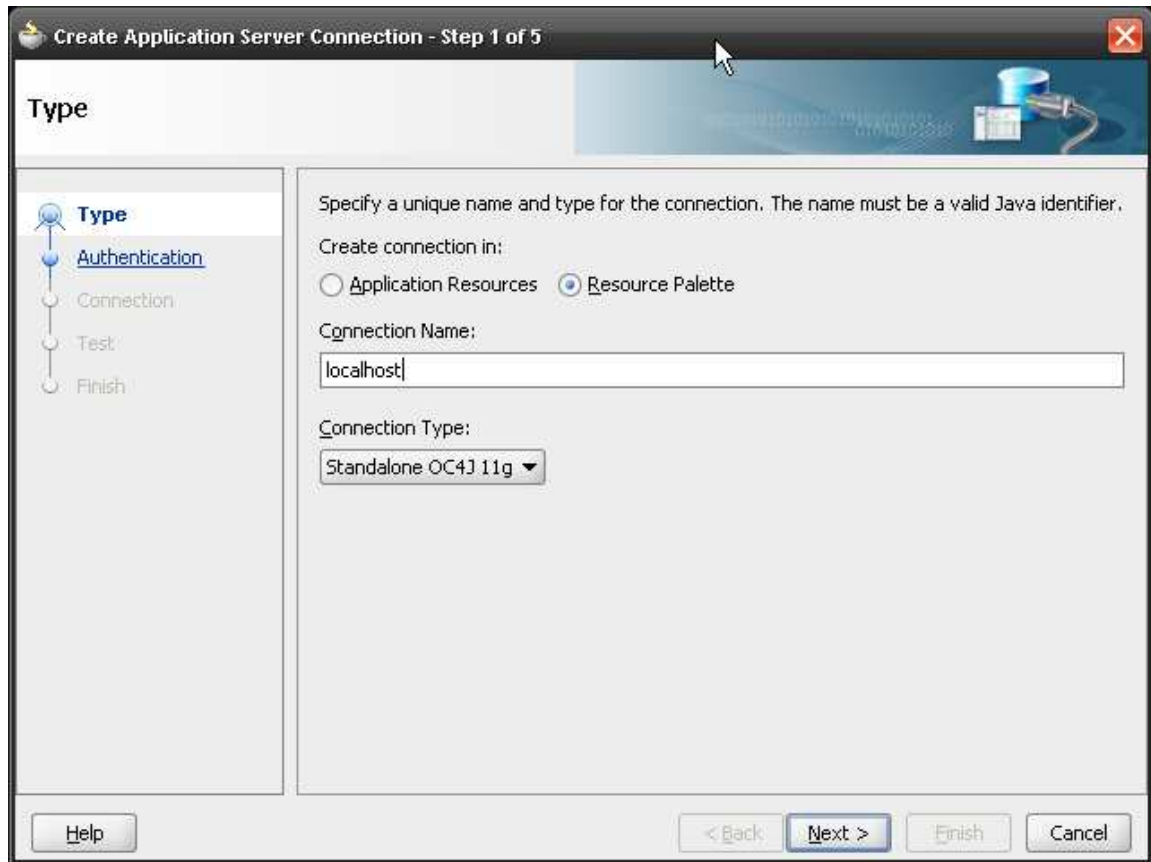
Figure 1 Deploying to a pre-defined connection



5. Click the down arrow button on the application menu and choose Deploy > OAR_CreditCardValidation > To > New Connection....



6. Enter a connection name, such as **localhost**.



7. Click **Next**.
8. Set the following fields:
 - **Username:** fmwadmin
 - **Password:** welcome1
 - **Deploy Password:** Checked

Create Application Server Connection - Step 2 of 5

Authentication

Specify a username and password to authenticate the connection. To bypass authentication at runtime, select Deploy Password.

Username:

Password:

☒ Deploy Password

< Back Next > Finish Cancel

9. Click **Next**.

10. Set the following fields:

- **Host Name:** localhost (or the location of your server if it's on another machine)

Leave all other fields with their default values.

Create Application Server Connection - Step 3 of 5

Connection

Please provide the host name and RMI port for the OC4J instance. This information is used to assemble an URL used to create a JMX connection to the server. An additional path may be specified as part of the URL if required; by default, it is not necessary.

Host Name: RMI Port:

URL Path:

< Back Next > Finish Cancel

11. Click the **Test Connection** button and verify the connection test succeeds.

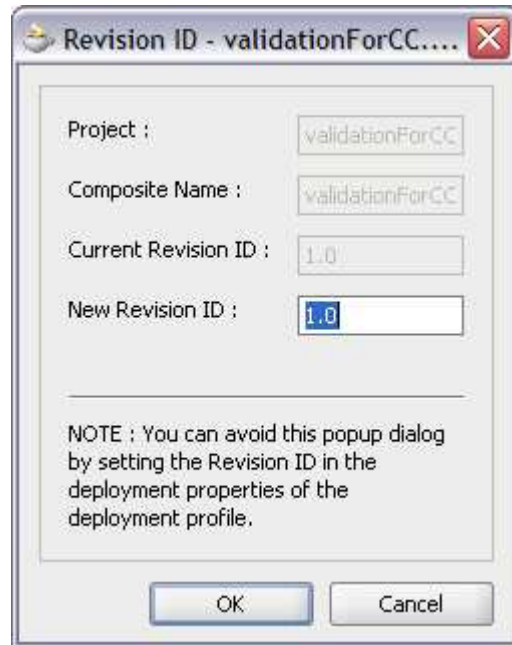
12. Click **Finish**.

That will complete the creation of the new connection, and the deployment will continue with that connection automatically.

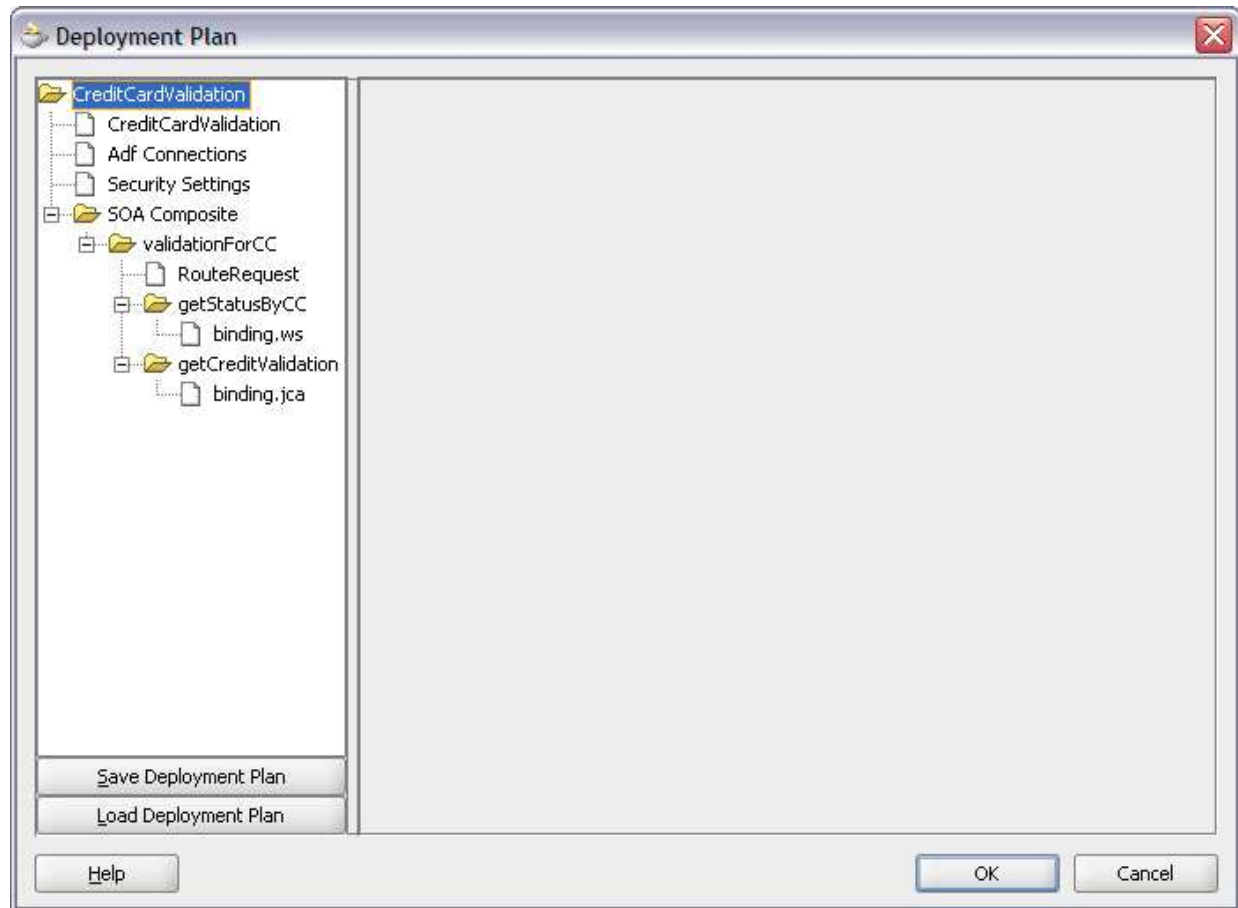
13. You may be prompted by the **Revision ID** dialog, as shown in Figure 2. The first time you deploy you won't get this dialog as it defaults to revision 1.0. But for subsequent deployments you will be prompted to specify a revision or version number.

Enter a new version, or keep the existing one. Click **OK**.

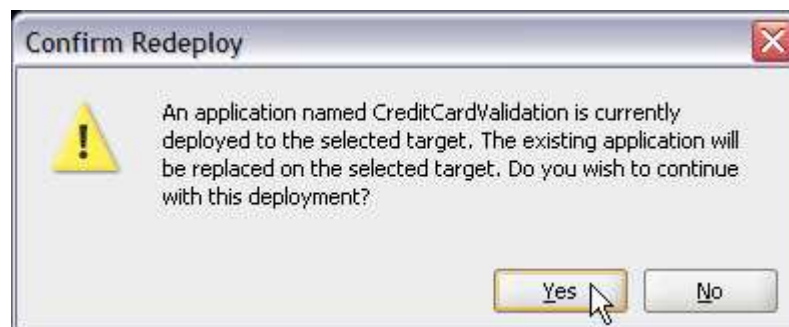
Figure 2 Revision Id dialog



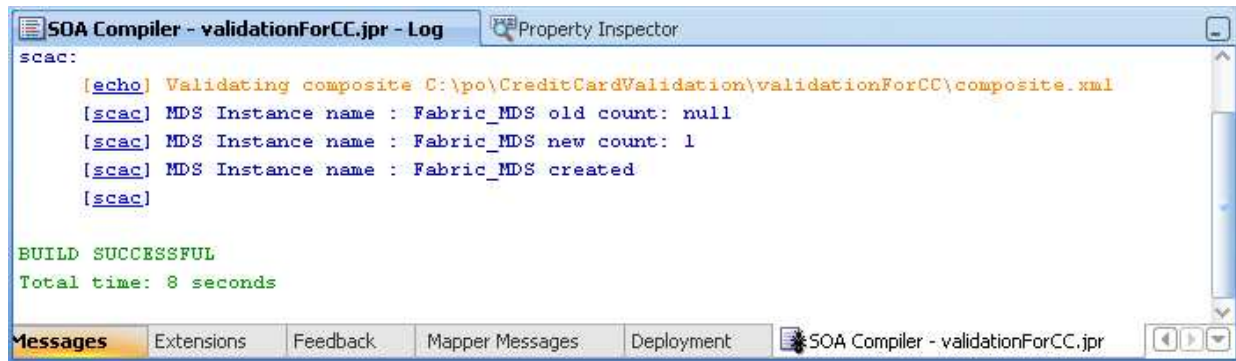
14. The **Deployment Plan** dialog will appear. You don't need to change anything, just click **OK**.



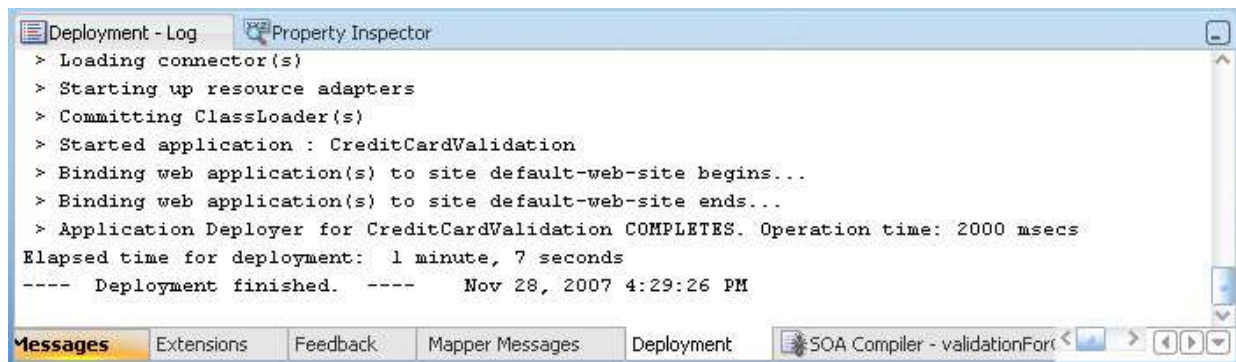
15. If you are redeploying your application you will be asked if the previous can be undeployed first. Click **Yes**.



16. First the SOA composite is compiled. Watch the **SOA Compiler** message window for any errors. It will have a green **BUILD SUCCESSFUL** message when complete (or will display the errors if there are any).



17. Once compiled, the **Deployment** tab will say **Deployment finished**.



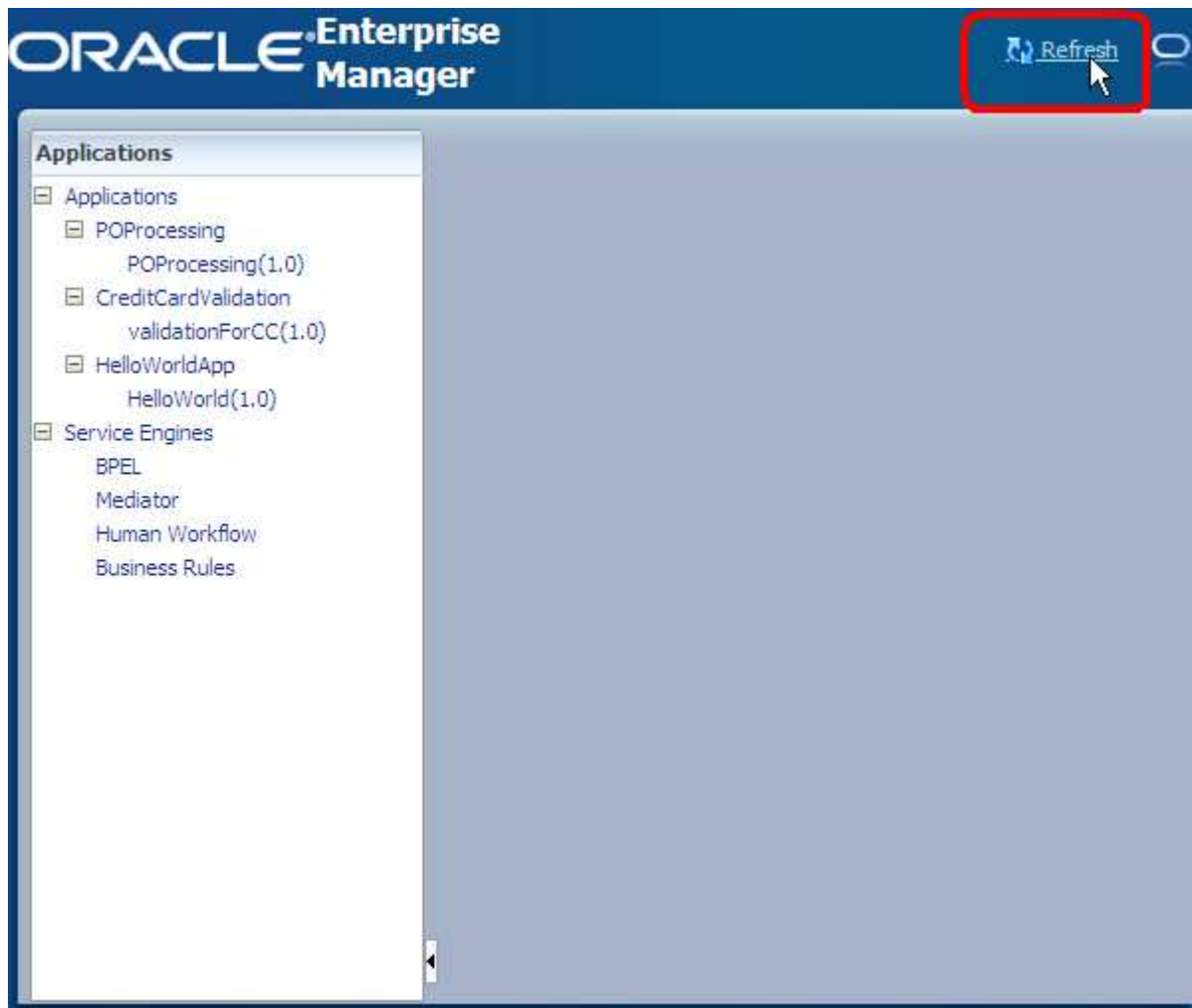
A.3 Running and testing the application

Often you'll have an application or some GUI-based front end to invoke your service so you can test it. But also frequently you will start developing your services first and won't have an application to test them with. The Oracle SOA Suite console screens allow you to run your service with any input so you can test it anytime.

1. Open your browser and navigate to the SOA Console.
 - For the Integrated OC4J use: <http://localhost:8898/SOAConsole>.
 - For the external OC4J use: <http://localhost:8888/SOAConsole>.

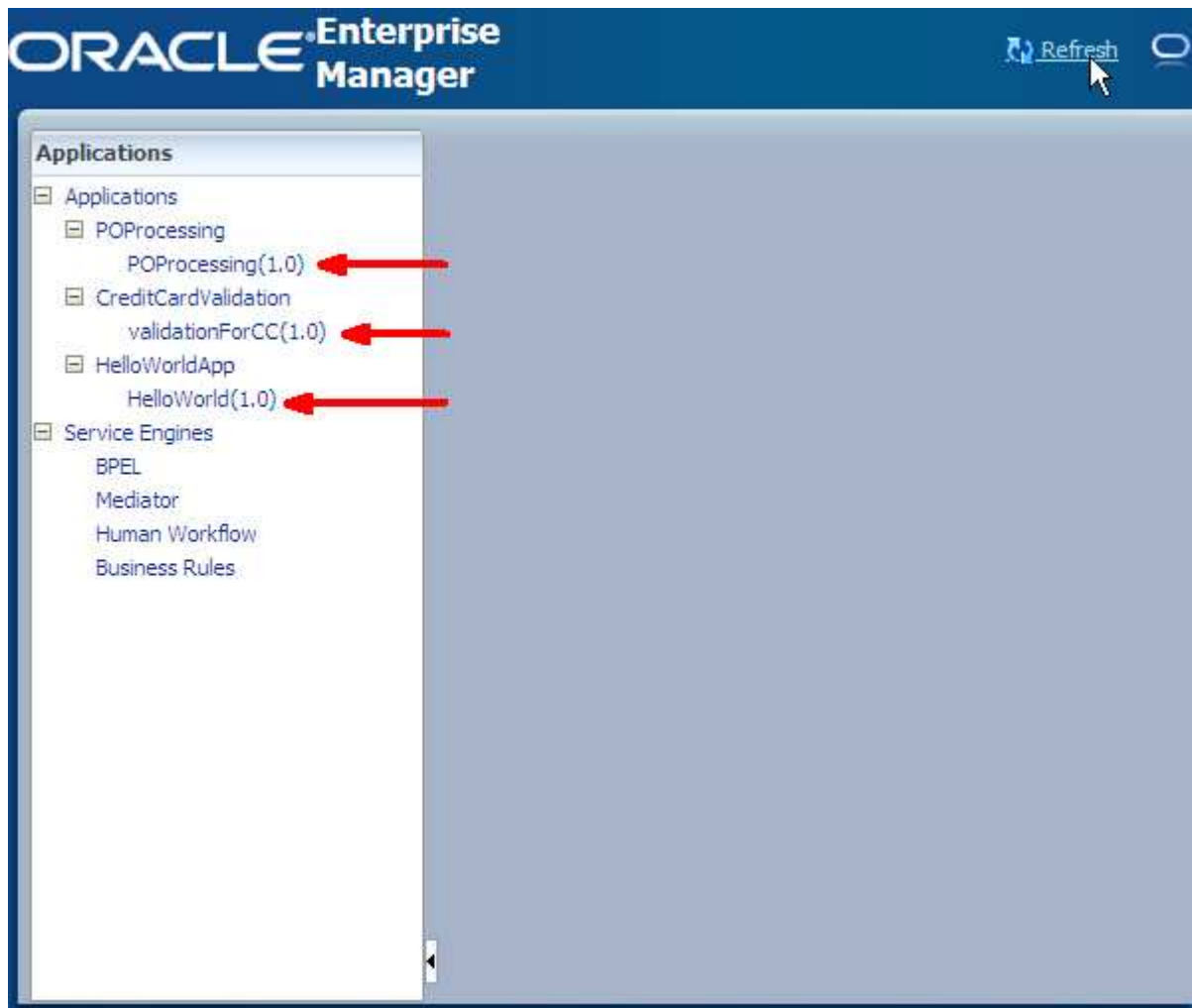
(The first time these screens are run after starting the server they will be slow as they are unpacked, compiled, and loaded into memory.)

2. In the **Applications** section on the left-hand side you should see your composite application listed. If not, press the **Refresh** link (at the top-right) occasionally until it appears.



3. Find your application, and click on the project and version you want to test. An application may have multiple versions and multiple projects, so be sure to click the correct one.

The following diagram shows what you click in order to test an application.



18. Open the test from the Actions menu and click **Test Service - <client>**, where <client> will whatever name was used for the invocation operation for your service.



19. In the input screen, specify your values for the payload. You can use the HTML form, which is default, or XML source view. For smaller amounts of the data, using the HTML form view is probably easier. But for large payloads, it's easier to copy-and-paste your data into the XML source view.

In Figure 3, the value "Jason" was entered as the input in the HTML form. In Figure 4, the same value as added in the XML source view.

Figure 3 HTML Form view

client endpoint

For a formal definition, please review the [Service Description](#).

Download the JavaScript Stub (BETA) for [HelloWorld_pt](#) and see its [documentation](#).

HelloWorld_pt

Operation : process ☒ HTML Form ☐ XML Source

☐ Reliable Messaging ☐ Include In Header

☐ WS-Security ☐ Include In Header

☐ payload

input xsd:string

Note: XML source view contents will not be reflected in the HTML form view

☐ Show Transport Info

☐ Perform stress test ☐ Enable

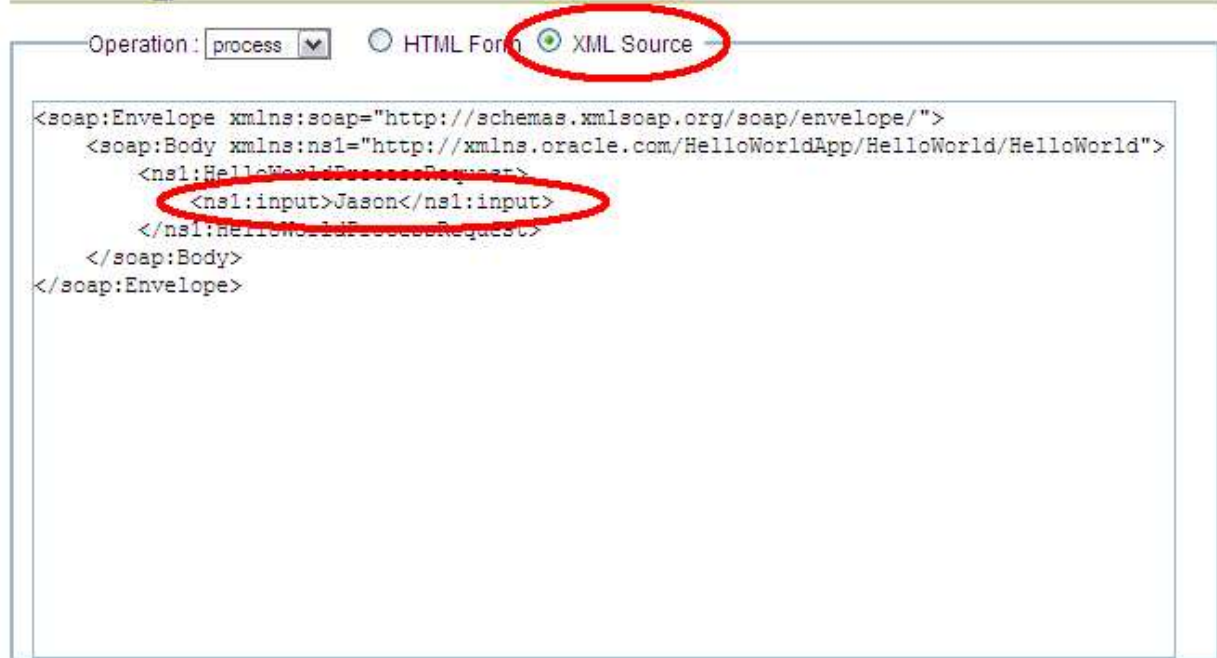
Figure 4 XML Source view

client endpoint

For a formal definition, please review the [Service Description](#).

Download the JavaScript Stub (BETA) for [HelloWorld_pt](#) and see its [documentation](#).

HelloWorld_pt



20. Whichever way you choose to specify your input, once it's done press the **Invoke** button.
21. The results screen will then appear. In this case, the HelloWorld process returns the input with "Hello " pre-pended on the front, as shown in Figure 5.

Note that not every application or process will show some results. A one-way (fire-and-forget) process, for example, won't return any results to display. Whether or not there are any results to display, you can always look at the instances ran, as shown in the next step.

Figure 5 Results from HelloWorld application**Test Result**View: [Formatted XML](#) | [Raw XML](#)[Return to Test Page](#)

```

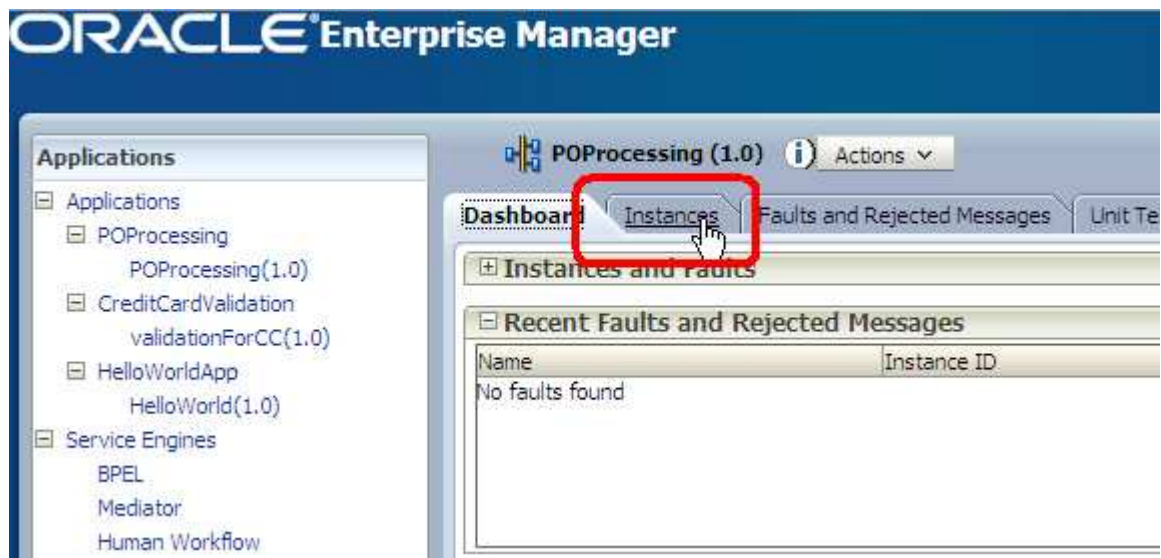
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:wsa="http://www.w3.org/2005/08/addressing"><env:Header><wsa:ReplyTo><wsa:Address
  xmlns:wsa="http://www.w3.org/2005/08/addressing">http://www.w3.org/2005/08/addressing/an
  xmlns="http://xmlns.oracle.com/HelloWorldApp/HelloWorld/HelloWorld">
    <result>Hello Jason</result>
  </helloWorldProcessResponse></env:Body></env:Envelope>

```

22. Return to the SOA Console, click on your application, and look at the **Last 5 Instances** section (as shown in Figure 6). You may need to press the **Refresh** link to update the screen first. Alternatively, you can click the **Instances** tab and see all past instances (Figure 7).

Figure 6 Viewing recent instances

Last 5 Instances Show more		
Instance ID	Faults	Start Time
302	0	12/12/07 9:55:31 AM
301	0	12/12/07 9:54:59 AM

Figure 7 Viewing all instances

23. Click the instance id of the instance you are interested in, probably the most recent one.



24. The **Trace** section will show an audit trail of the component instances from your application. Click on the component whose audit trail you would like to see, in this case for the HelloWorld BPEL process. (Your browser may prevent pop-ups so make sure that is enabled.)

25.

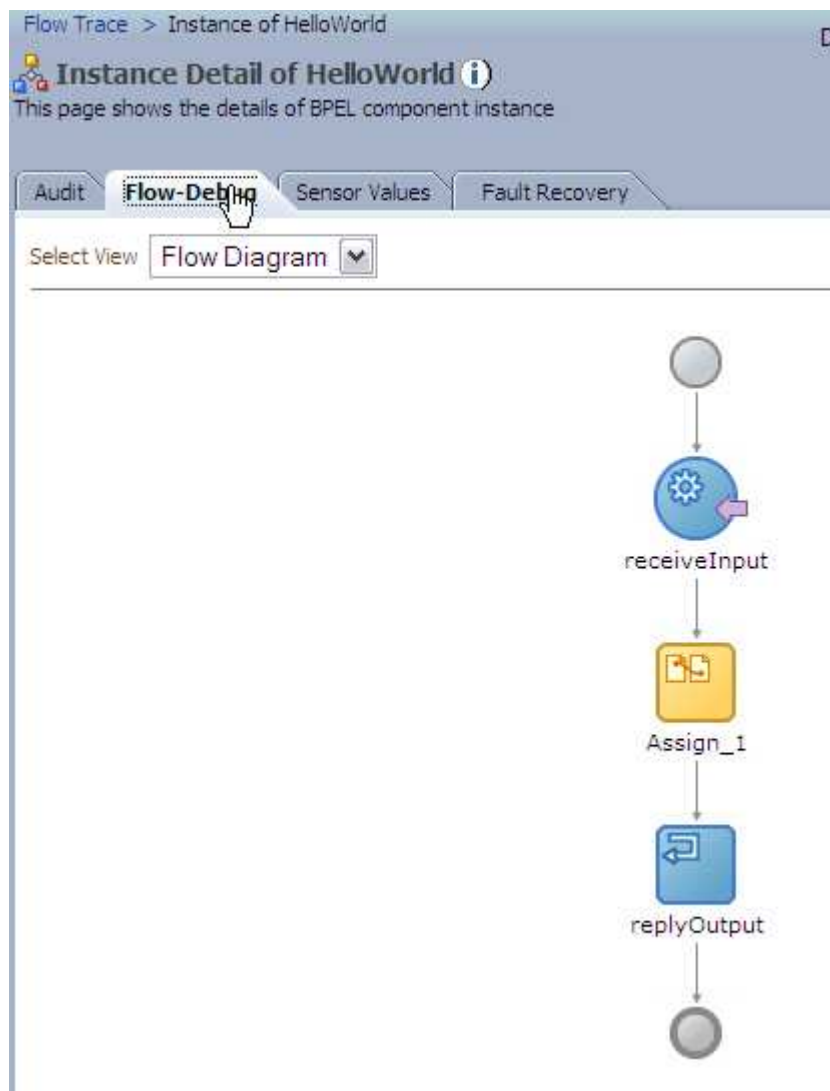
Trace

Click on a component instance to see its detailed audit trail

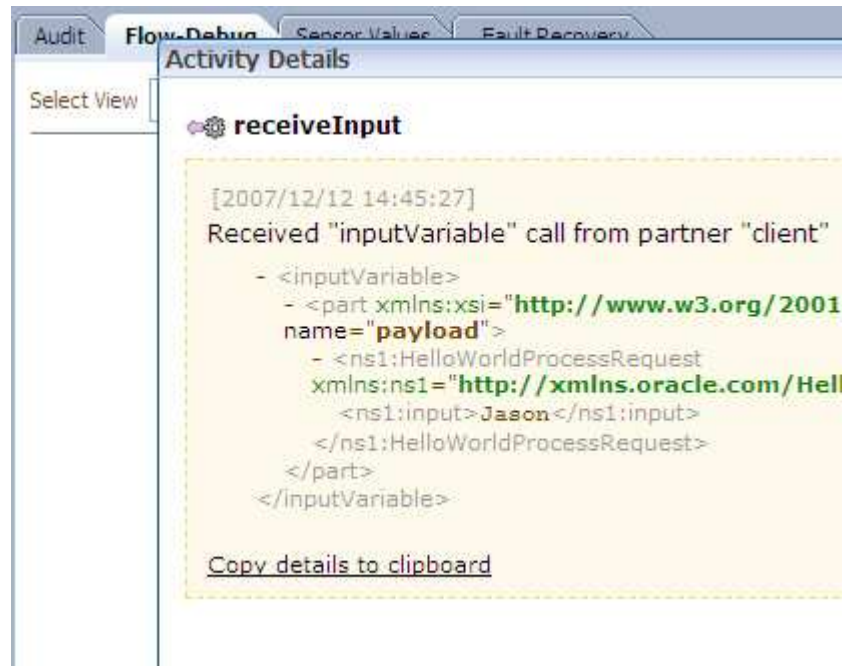
Show Instance IDs ☐

Instance	Type
 client	Service
 HelloWorld	BPEL Component

26. The audit trail will appear. For BPEL processes, click the **Flow-Debug** tab to see a visual representation of the BPEL process.



27. Click on the <receive> activity (called **receiveInput** here) to see the XML data related to that activity.



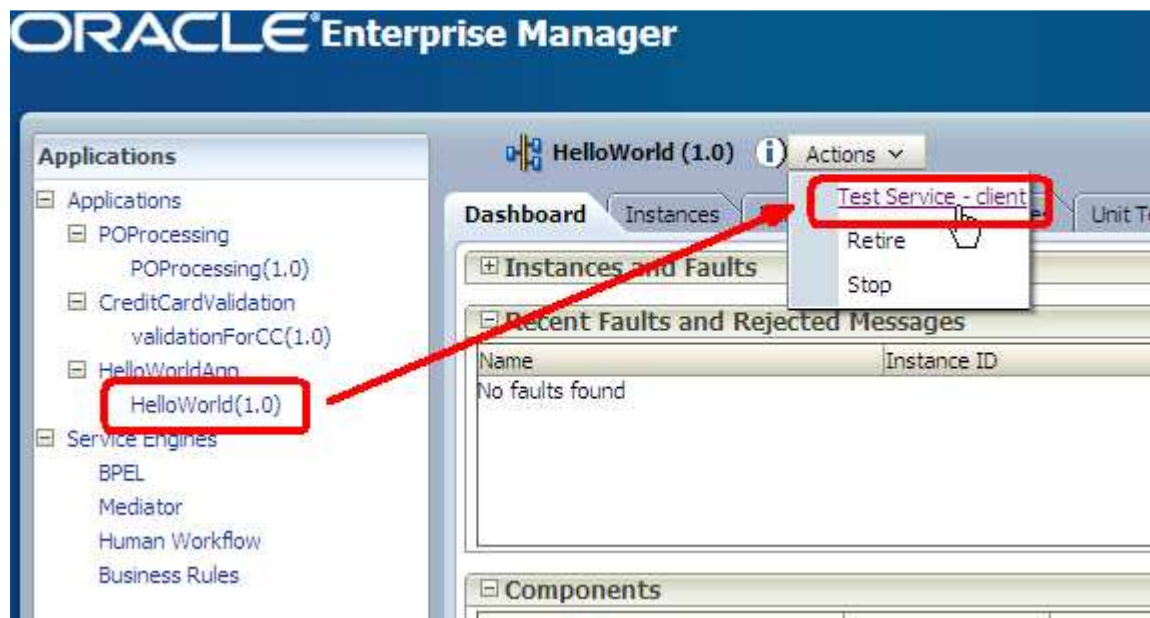
28. Click on any of the other activities to see their results.

29. Close the **Instance Detail** window when done.

A.3.1 Getting the service description (WSDL)

You can also get the service description, or WSDL, of your application from the tester. This is useful because you may need it for applications to access your service, or to expose your service to others.

1. Open the tester for the application you are interested in.



2. In the tester, the **Service Description** link is the address for your WSDL. You can copy the link location to use elsewhere, or click on it if you want to view the WSDL.

client endpoint

For a formal definition, please review the [Service Description](#).

Download the JavaScript Stub (BETA) for [HelloWorld_pt](#) and see its [documentation](#).

HelloWorld_pt

Operation : ☒ HTML Form ☐ XML Source

☐ Reliable Messaging ☐ Include In Header

☐ WS-Security ☐ Include In Header

☐ payload

input xsd:string

Note: XML source view contents will not be reflected in the HTML form view

- ☐ Show Transport Info
- ☐ Perform stress test ☐ Enable