Practices for Lesson 1: Introduction to Service-Oriented Architecture

Practices for Lesson 1: Overview

Practices Overview

The Oracle SOA Suite Quick Start distribution allows you to quickly install a development or an evaluation environment on a single-host computer. Quick Starts are installations for development or evaluation only. If you want a production environment, start with “Planning the Oracle SOA Suite and Business Process Management Installation” in *Installing and Configuring Oracle SOA Suite and Business Process Management* to install a full-scale development environment that can be scaled to a production environment.

In this practice, you are guided through the Developer ‘Quick Start’ installation process and should be able to use Oracle SOA Suite in about 30 minutes. The installer:

- Installs all components that are necessary for development with the core of Oracle SOA Suite (BPEL, Business Rules, Mediator, and Human Work Flow)
- Launches JDeveloper after the installation, automatically registering the IDE plug-ins for Oracle SOA Suite

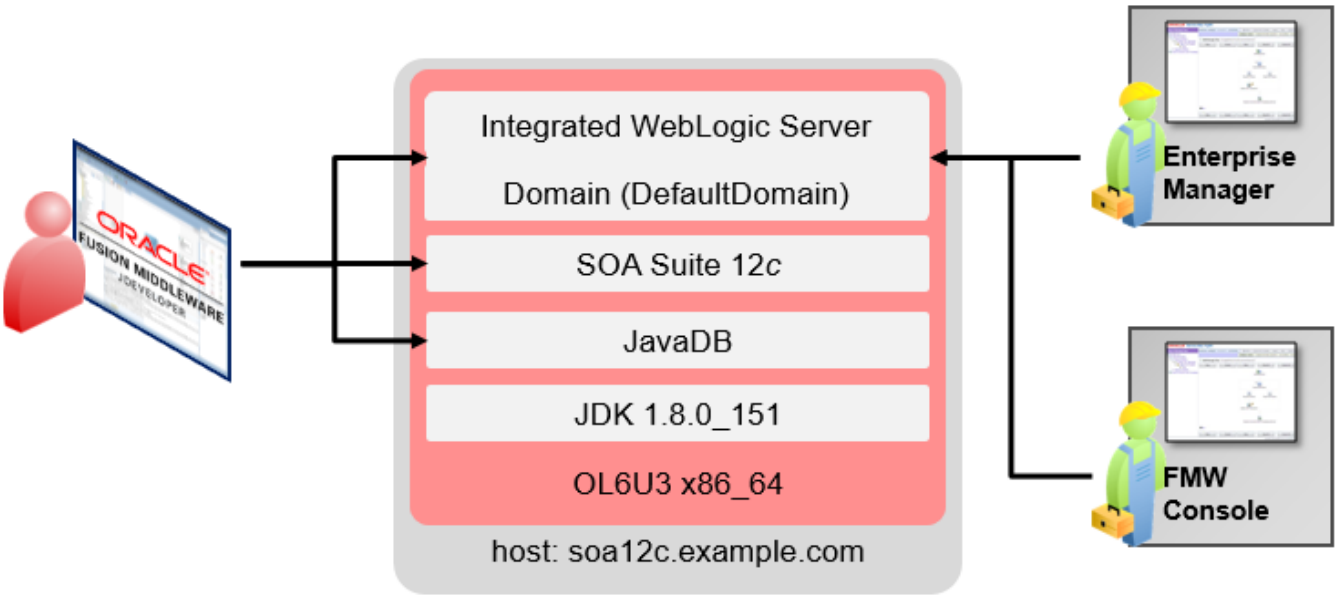
The SOA Suite 12*c* Developer installation contains the following:

- JDeveloper
- WebLogic Server
- SOA Suite runtime components
  - BPEL, Human Workflow, Rules, and Mediator
  - Service Bus
- JDeveloper IDE plug-ins for the preceding components

All software is installed in a single middleware home directory, which you specify.

In JDeveloper, the WebLogic Integrated Server is pre-configured with the preceding SOA Suite runtime components and JavaDB. JavaDB is a development database that allows you to start development with SOA Suite without the need to run the repository creation utility (RCU).

Your environment looks like the following on the hosted server when you have finished:



Practice 1-1: Installing Oracle SOA Suite

Tasks

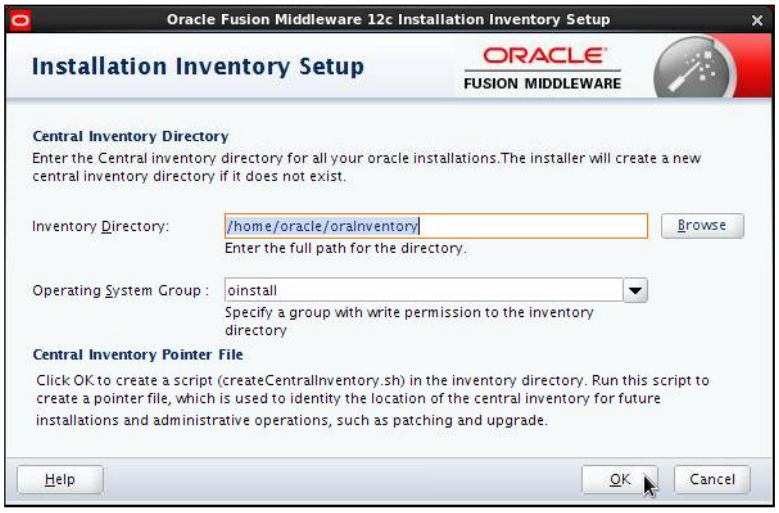
1. Right-click the Terminal icon on your desktop and select Open.

   A terminal window opens.

2. In the terminal window, initiate the installation script by issuing the following commands:

   ```
   cd /home/oracle/labs/scripts/util
   sh startInstall.sh
   ```

   The package is extracted and the Installation Inventory Setup dialog box appears.



   Note: On UNIX and Linux systems, the wizard begins by prompting for the location of the Installation Inventory file. (This screen does not appear on Windows operating systems.) The Central Inventory is a master list of all Oracle products installed on this host. Provide the location of an existing Central Inventory file or specify where you want to create your Central Inventory. Make sure that the operating system group name selected on this screen has write permissions to the Central Inventory location.

   There is more to know about Central Inventory files and directories but for the purpose of this course, we accept the default `/home/oracle/oraInventory` as the directory name for our installation inventory. For more information about the Central Inventory, see “Understanding the Oracle Central Inventory” in *Installing Software with the Oracle Universal Installer*.

3. Click OK to accept the default installation inventory directory.

   The package is extracted and the installation wizard appears.

4. Use the instructions in the following table to complete the installation wizard.

1	Welcome	Click **Next**. Skip Auto Updates. Click **Next**.
2	Installation Location	Provide the installation directory: `/u01/app/fmw12c/Oracle_Home` Click **Next**.

3	Prerequisite Checks	Verify success and click **Next**.
4	Installation Summary	Review all the components that are installed in one step. Click **Install**.
5	Installation Progress	This step installs, links, and configures all software. This step takes 6–7 minutes. When 100% is complete, click **Next**.
6	Installation Complete	Deselect the "Start JDeveloper" check box. Click **Finish**.

Note: For more information about the Oracle Fusion Middleware directory structure, see "Selecting Directories for Installation and Configuration" in *Planning an Installation of Oracle Fusion Middleware.*

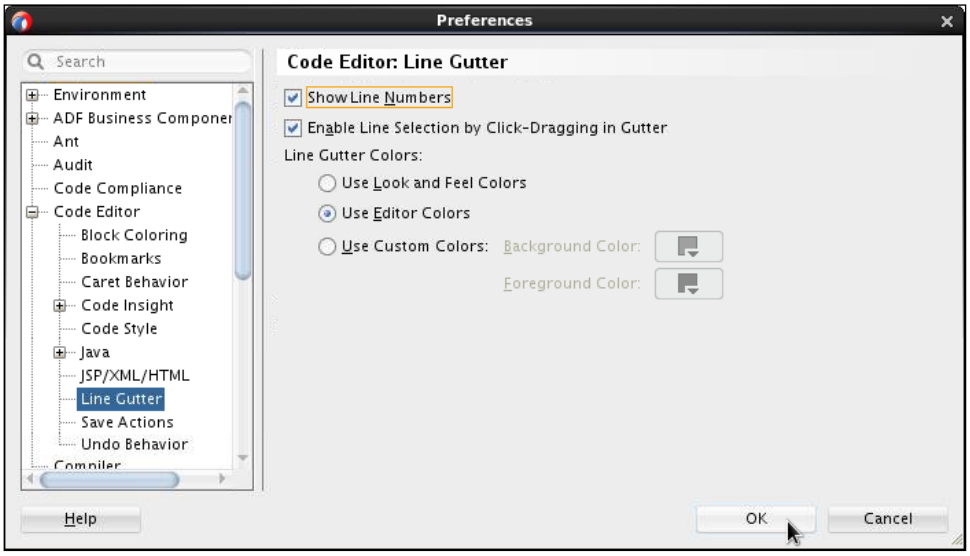
5. Start JDeveloper.
   a. In the open terminal window, issue the following commands:
      ```
      cd
      cp /home/oracle/labs/scripts/util/startJDev.sh .
      ```
      The script to start JDeveloper is copied to a more convenient location.
   b. In the terminal window, issue the following command to start JDeveloper:
      ```
      sh startJDev.sh
      ```
      JDeveloper is started.

   Tip: Each time you are asked to start JDeveloper in this course, use this command.

   The Select Role dialog box opens.
6. Complete the following steps to configure JDeveloper defaults:
   a. Deselect the check box next to "Always prompt for role selection on startup."
   b. To accept the Studio Developer (default) role, click **OK**.
      The Confirm Import Preferences dialog box is displayed.
   c. Click No.
      The Oracle Usage Tracking dialog box is displayed.
   d. Deselect the check box next to "Allow automated usage reporting to Oracle."
   e. Click OK.
      JDeveloper opens and displays the Start Page tab.
   f. Deselect the check box next to "Show on Startup" and close the Start page.

   Note: In 12*c,* JDeveloper automatically scans the middleware directory and registers all the IDE plug-ins. Unlike earlier releases, you do not have to download them separately. When JDeveloper comes up, it is ready for SOA Suite development.
7. Set JDeveloper preferences.
   a. In the JDeveloper menu, select Tools > Preferences.
      The Preferences dialog box is displayed.
   b. Navigate to Code Editor > Line Gutter.

c. Select the Show Line Numbers check box.



d. Click OK.

8. Start the Integrated Server and configure a domain.

Note: Launching the Integrated WebLogic Server from JDeveloper generates a domain, which is designated as the default domain. However, you also have the option of launching a JDeveloper-independent version of the WebLogic Server to configure a stand-alone domain. Creating a stand-alone domain is the preferred and Oracle recommended way of creating a WebLogic Quick Start SOA domain. However, for simplicity, in this course, you will use the Integrated WebLogic Domain.

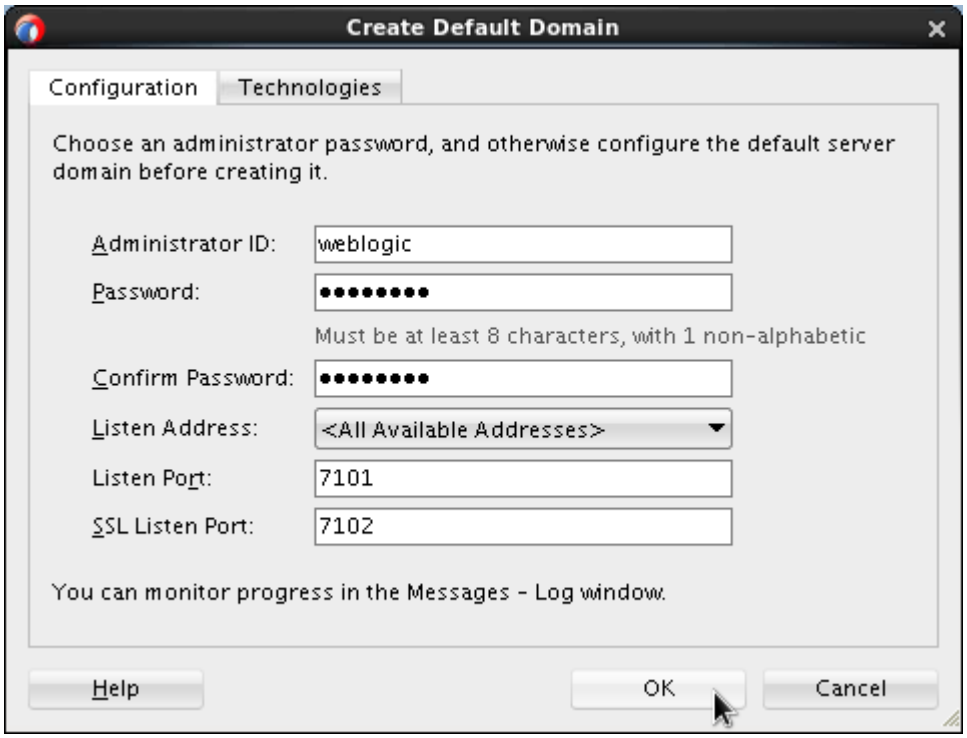
For more information, see “Configuring a Domain” in *Installing SOA Suite and Business Process Management Suite Quick Start for Developers.*

a. In the JDeveloper main menu, select Run > Start Server Instance.

The Create Default Domain dialog box opens.

b. Provide and confirm the password **`welcome1`**.

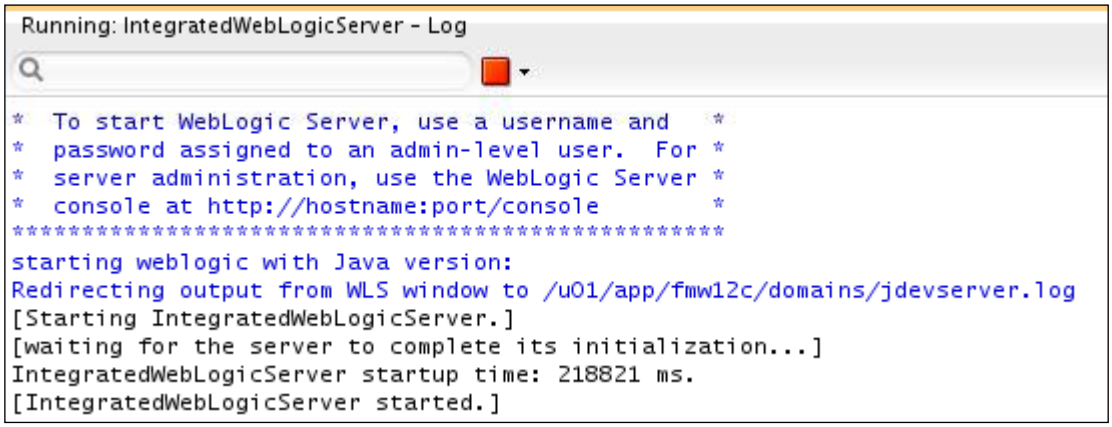
c. Click OK.



Domain creation begins. This takes several (about 10) minutes. You can monitor progress in the Running: IntegratedWebLogicServer pane at the bottom center of JDeveloper screen.

When you see the message: `[IntegratedWebLogicServer started.]` the domain configuration has been completed and the server is running.

Note: It may appear as if the server hangs at times. Please be patient, the configuration and startup will complete.

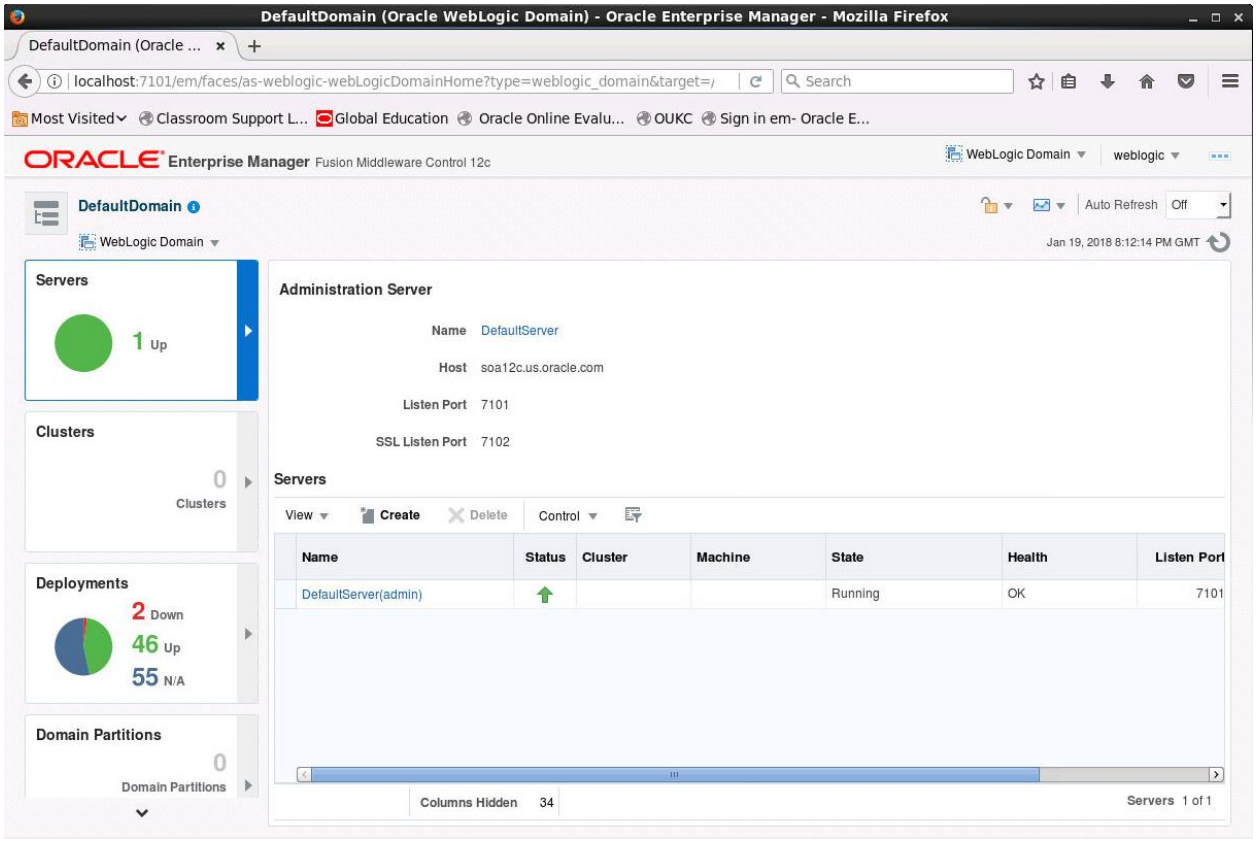


9. Verify your domain.
   a. Open a web browser and access Enterprise Manager Fusion Middleware Control, which is located at http://localhost:7101/em.

      Tip: You may wish to bookmark this URL. You will use it frequently in this course.

b. Log in by using the username `weblogic` and password `welcome1`.

You are logged in to DefaultDomain. The server status is shown as Up.



c. Log out of Enterprise Manager.

10. Disable Secure Sockets Layer (SSL).

SSL is enabled by default in the Integrated WebLogic Server. We do not have stringent design time requirements, and some later practices are simplified by not using it, so we disable SSL by completing the following steps:

a. Go to the Administrator Console at http://localhost:7101/console.

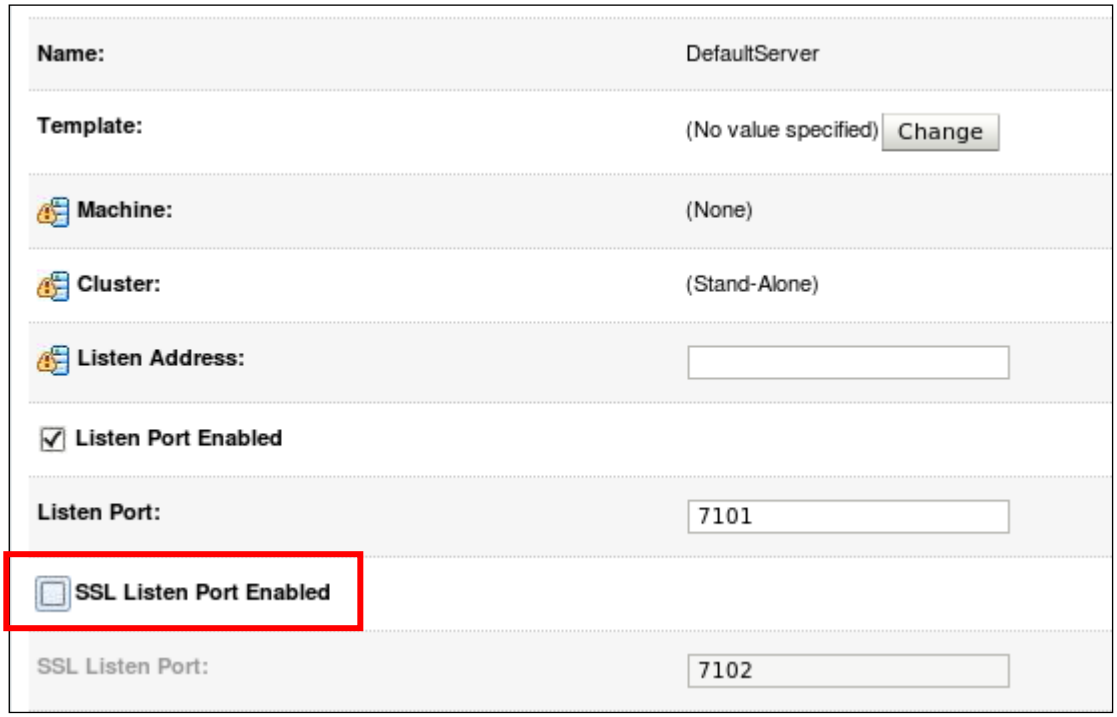
Tip: You may wish to bookmark this URL. You will use it again in this course.

b. Log in by using the username `weblogic` and password `welcome1`.

c. In the Domain Structure pane, click Environment > Servers.

d. In the main pane, click DefaultServer(admin).

e. Deselect **SSL Listen Port Enabled**.



f. Click Save.

g. Log out of Fusion Middleware Console.

h. From the JDeveloper main menu, stop and restart the Integrated WebLogic Server.

- Select Run > Terminate > IntegratedWebLogicServer.
- Select Run > Start Server Instance (IntegratedWebLogicServer).

Summary

In this practice, you have completed the one-time task of installing Oracle SOA Suite. You have also been introduced to several tasks that you execute repeatedly in this course, including:

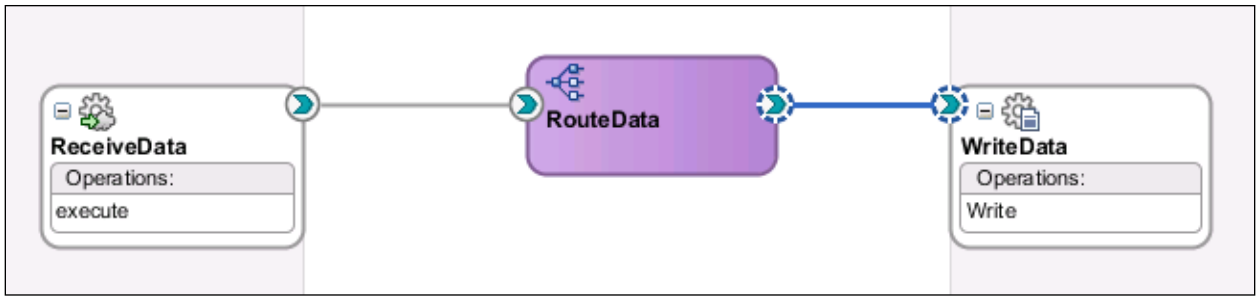
- Accessing Enterprise Manager
- Accessing Fusion Middleware Console
- Starting and stopping the Integrated WebLogic Server

Practices for Lesson 2: Getting Started with Composite Applications

Practices for Lesson 2: Overview

Practices Overview

In this practice, you create, deploy, and run a simple SOA composite application. The goal of this practice is to become acquainted with the development environment and to begin forming an understanding of composite application components.



Practice 2-1: Creating the Composite Application

Overview

In this practice, you create a composite application that includes a Mediator component that routes incoming data to a File adapter service reference. The File adapter writes the data to a file.

Assumptions

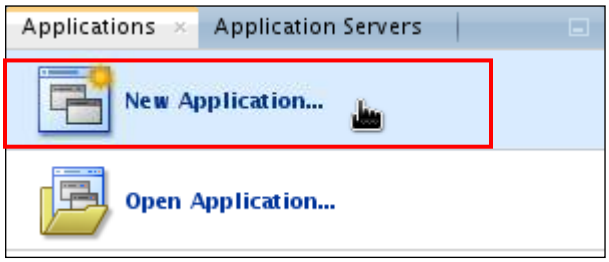
This practice assumes that you have started JDeveloper.

Tasks

1. Create a new application.

   Note: When you create a SOA application, Oracle JDeveloper creates a project that contains all the source files related to your application. You can then use Oracle JDeveloper to create additional projects needed for your application.

   a. In the JDeveloper Application Navigator, select New Application.



   The New Gallery dialog box is displayed. The Applications Category is selected by default.

   b. Select SOA Application from the Items list and click OK.

The Create SOA Application wizard is displayed. The first steps in building a new application are to assign it a name and to specify the directory in which to save the source files.

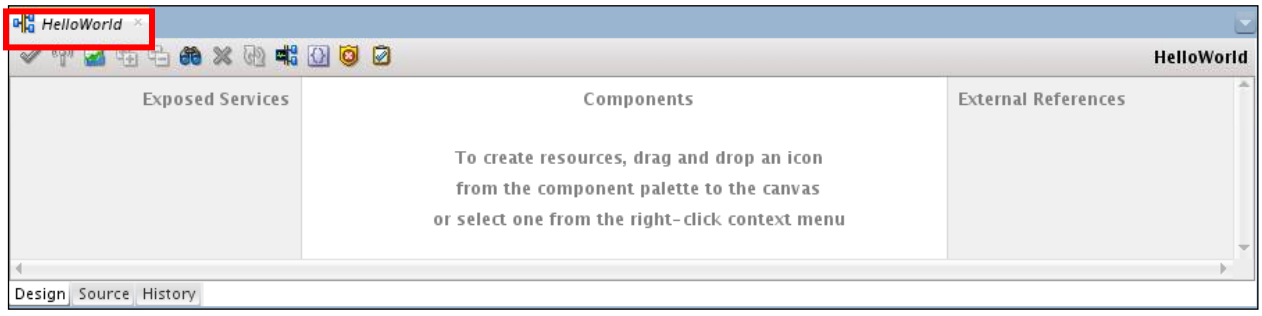
c. Supply `Basics` as the application name and click Next.

d. Supply `HelloWorld` as the project name and click Next.

The Project SOA Settings are displayed.

e. Click Finish. (A minute or more may elapse as the new project is created.)

Note: The *composite_name* file (in this case, *HelloWorld*) appears as a tab in the designer and as a file in the Applications window. This file (also known as the `composite.xml` file) is automatically created when you create a new SOA project. This file describes the entire composite assembly of services, service components, and references. There is one `composite.xml` file for each SOA project.



Note: The left swimlane is for services (such as web services, REST adapters, or JCA adapters) that provide an entry point to the SOA composite application. The right swimlane is for references that send messages to external services in the outside world, such as web services or JCA adapters. You drag and drop service components such as BPEL processes, business rules, human tasks, Oracle Mediators, and spring components onto the designer in the Components swimlane.

f. Select File > Save All (or click the Save All icon on the toolbar) to save the changes to your application and project.

Creating and Configuring the Exposed Service Interface

This interface describes how a client can call the composite application.

2. In the overview window, drag a SOAP technology component from the Technology section of the Component Palette into the Exposed Services column.

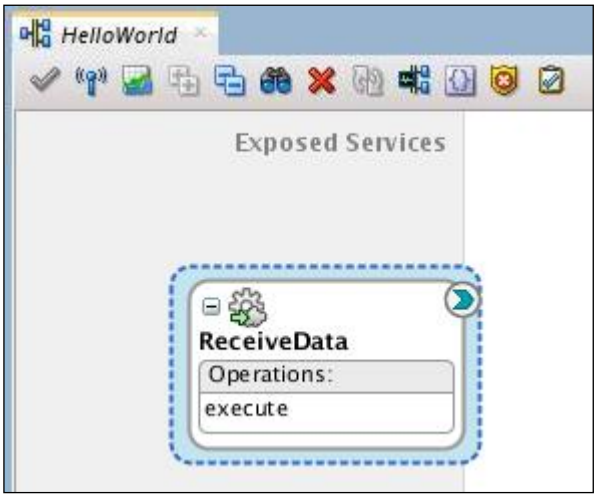
   The Create Web Service dialog box opens.

3. To configure the interface for the web service interface, perform the steps in the following table:

Step	Window/Page Description	Choices or Values
a.	Create Web Service	Name: ReceiveData Next to the WSDL URL field, click the “Generate WSDL from schema(s)” icon ().

Step	Window/Page Description	Choices or Values
b.	Create WSDL	Click the “Add a new message part” icon ( ).
c.	Add Message Part	Click the “browse for schema file” icon ( ).
d.	Type Chooser	Click the Import Schema ( ) icon.
e.	Import Schema File	With FileSystem selected, locate the `/home/oracle/labs/files/xsd` folder and select `po.xsd`. Click OK.
f.	Localize Files	Click OK.
g.	Type Chooser	Expand the Project Schema Files > `po.xsd` entry (if needed), and select the PurchaseOrder entry. Click OK.
h.	Add Message Part	Click OK.
i.	Create WSDL	Click OK.
j.	Create Web Service	Click OK.

4. In the overview window, verify that the ReceiveData icon appears in the Exposed Services column. Example:



5. Select File > Save All (or click the Save All icon on the main toolbar) to save the changes to the project.

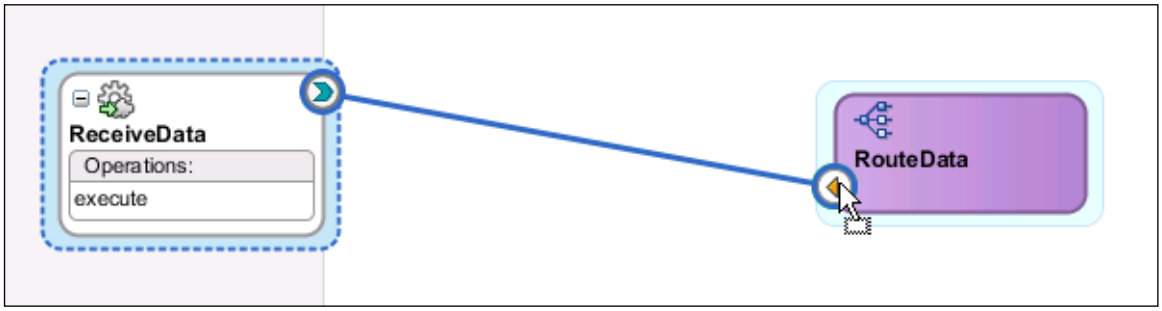
Creating and Wiring the Service Mediator Component

In this step, you add a Mediator component that receives data from the application’s client, and then routes that data to a File adapter (which you build in the next step).

6. Create a Mediator component and wire the ReceiveData service entry point to the Mediator component, by performing the following steps:
   a. Drag a Mediator component from the Component Palette into the Components column of the overview window.

      The Create Mediator dialog box opens.
   b. Set the name to RouteData.
   c. Accept the [default] Define Interface Later template.
   d. Click OK.

      Note: The Mediator interface is defined in the next step by wiring the SOAP service to the Mediator component. You can drag a defined interface to an undefined interface in either direction (reference to service or service to reference). The undefined interface then inherits the defined interface.
   e. Create a wire from the ReceiveData exposed service icon to the RouteData Mediator icon, by dragging the right-edge arrow icon from the ReceiveData exposed service to the left-edge arrow on the Mediator component. Use the following image as a guide:



7. Select File > Save, or click the Save All icon to save the changes to your application.

Creating a File Adapter as an External Reference

You now create an external reference for the File adapter to write the input message structure to a file in the file system.

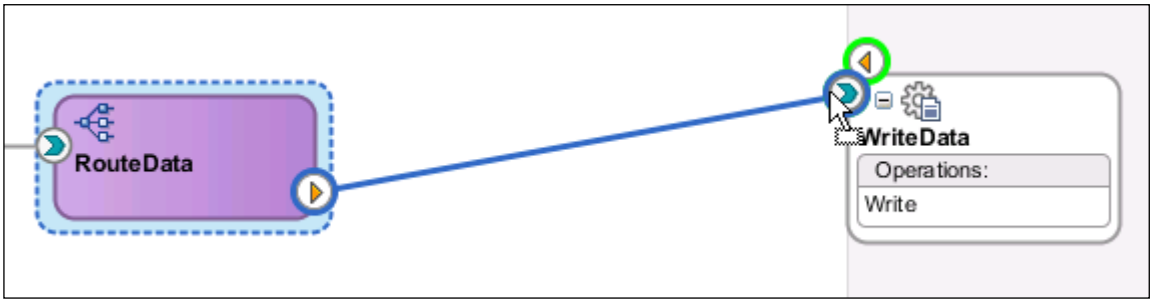
8. In the overview window, drag a File adapter component from the Technology section of the Component Palette onto the External References column.

   The FILE Adapter Configuration wizard opens.
9. On the Adapter Configuration wizard pages, use the instructions in the following table to configure the File Adapter:

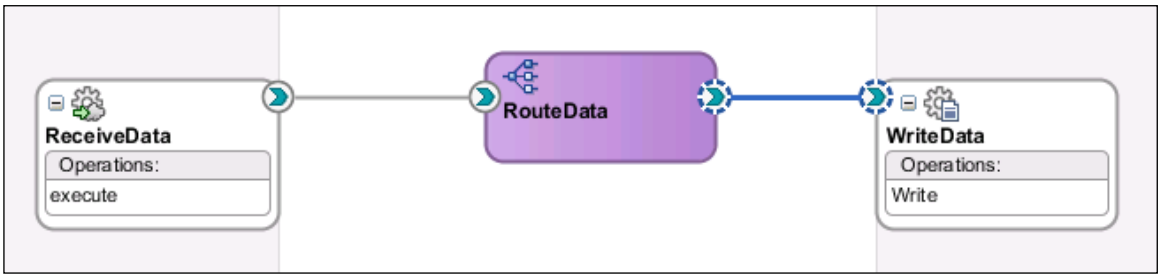
Step	Window/Page Description	Choices or Values
a.	File Adapter Reference	Service Name: WriteData Click Next.
b.	Adapter Interface	Accept the [default] “Define from operation and schema (specified later)” option. Click Next.

Step	Window/Page Description	Choices or Values
c.	File Server Connection	File Server JNDI Name: `eis/FileAdapter` Click Next.
d.	Operation	Select the Write File option. Click Next.
e.	File Configuration	Directory for Outgoing Files: `/home/oracle/labs/output/podata` File Naming Convention: `order_%SEQ%.xml` Click Next.
f.	Messages	Click the “browse for schema file” icon.
g.	Type Chooser	Expand the Project Schema Files > `po.xsd` entry and select `PurchaseOrder`. Click OK.
h.	Messages	Click Next.
i.	Finish	Click Finish.

10. Create a wire from the Mediator component to the File adapter by dragging the right-edge arrow of the Mediator component to the left-edge arrow on the File adapter icon. Use the following image as a guide:



11. Verify and save your work.



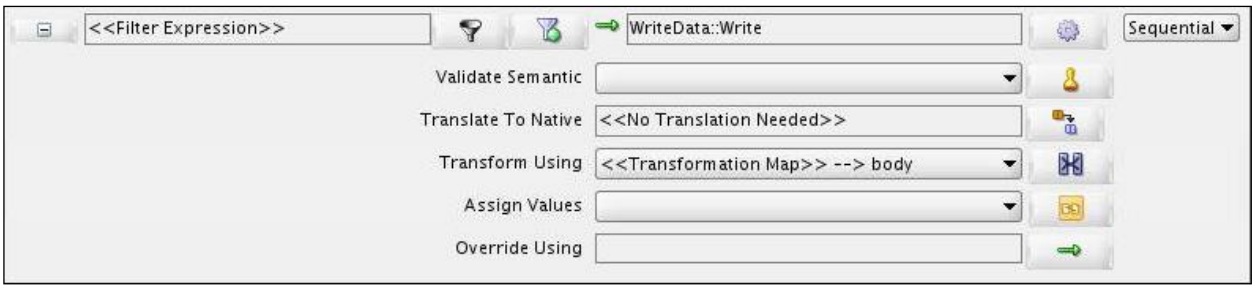
Adding a Transformation to the Mediator

This transformation reformats the incoming data before passing it to the File adapter.

12. In the overview window, right-click Mediator RouteData and select Edit.

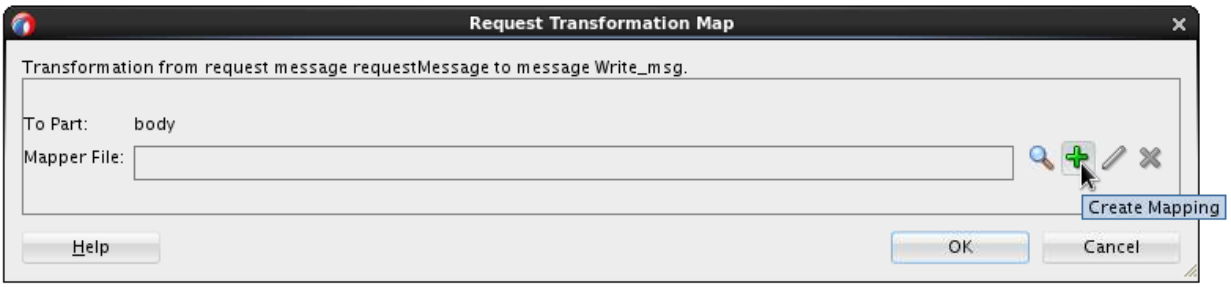
    The Mediator configuration editor (`RouteData.mplan`) opens.

13. Click the “Select an existing mapper file or create a new one” icon next to the Transform Using field.



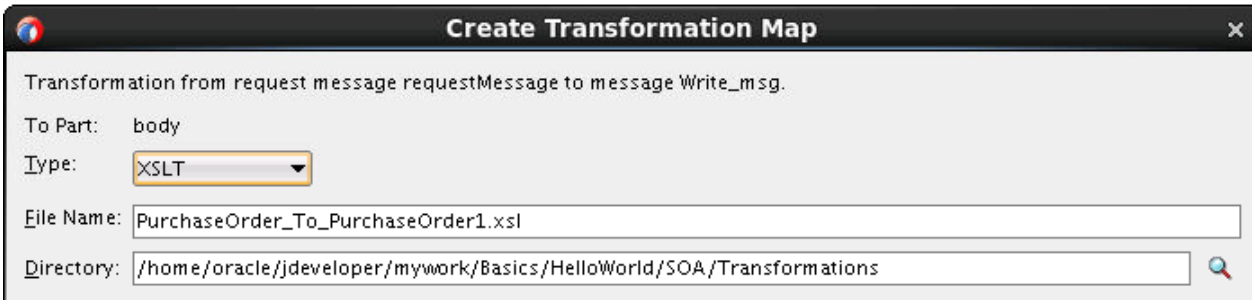
    The Request Transformation Map dialog box is displayed.

14. Click the Create Mapping icon.



    The Create Transformation Map dialog box opens.

15. To accept the default values, click OK.

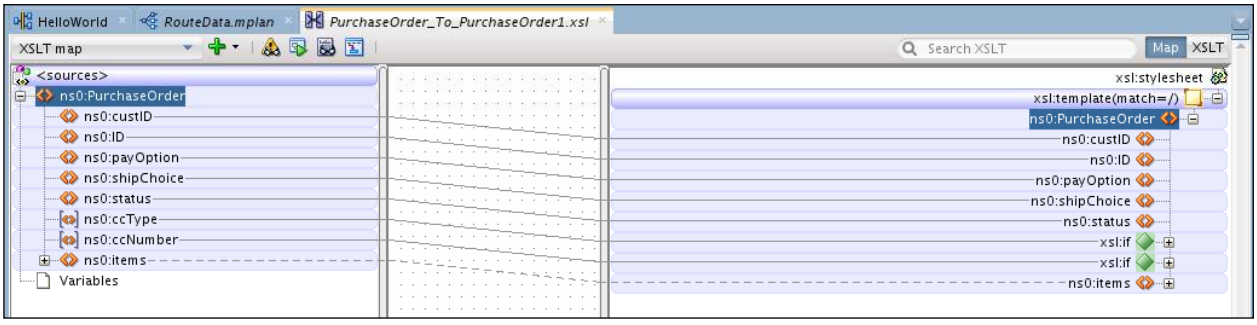


The XSLT Mapper (`PurchaseOrder_To_PurchaseOrder1.xsl`) window opens.

16. Click OK to close the Request Transformation Map dialog box.
17. Drag the PurchaseOrder element in the source column and drop it onto the PurchaseOrder element in the target column.

    The AutoMap Preferences window opens.
18. Click OK to accept the defaults.

    The AutoMap feature generates an XSLT mapping of the source nodes to the destination nodes.

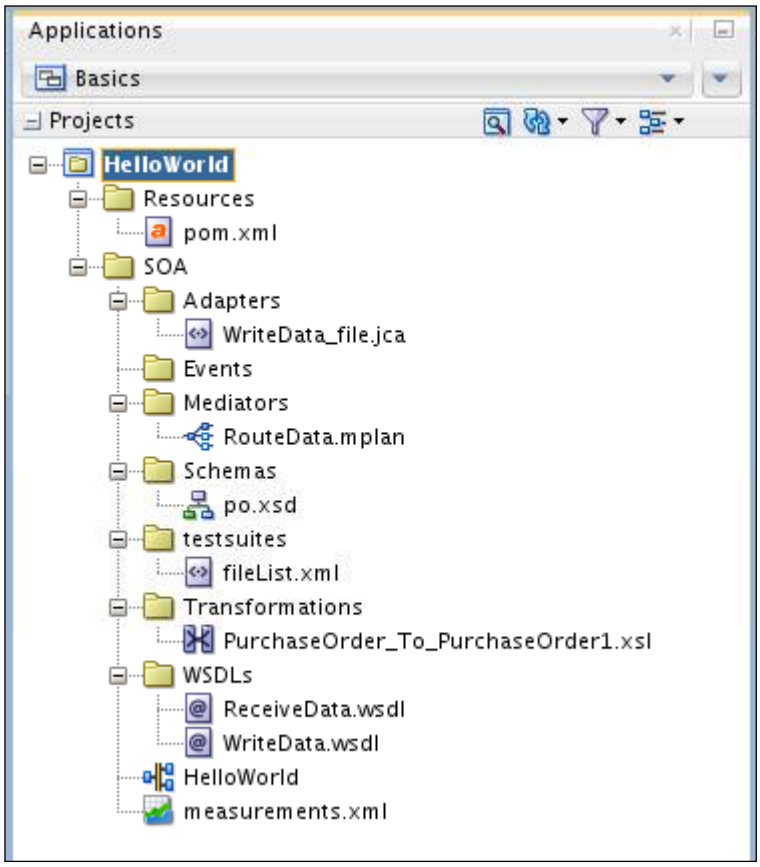


Note: Lots of interesting things are happening in this window! We will discuss data transformation in more detail a little later. For now, we have generated the minimum possible transformation to allow us to focus our attention elsewhere for this first composite application.

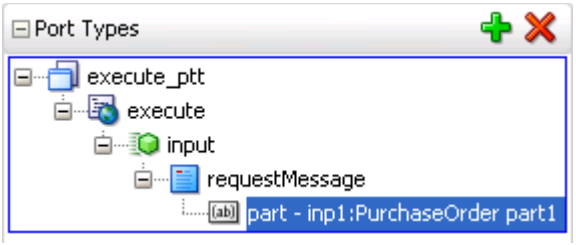
19. Save your work. Close the `.xsl` and `.mplan` file editors.

    Note: The next lesson titled “Managing and Monitoring a Composite Application” introduces the SOA Suite runtime environment, and the tools for managing, monitoring, and testing composite applications. In the practice for that lesson, you will deploy and test this application.

Examining the Generated Configuration Files

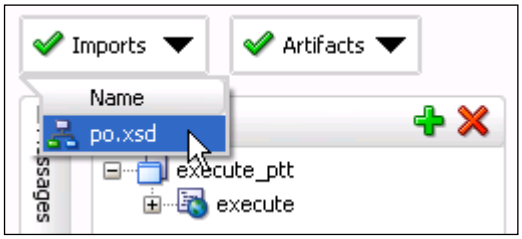


20. Open the `ReceiveData.wsdl` file in JDeveloper and consider the following questions:
    a. In the following diagram, what does `inp1:PurchaseOrder` reference?
    b. To learn more about `PurchaseOrder`, which file would you open?



    c. The reference `inp1:PurchaseOrder` names the element of the message that describes the structure of the request message.

d. Because the message formats are described in an imported XSD file, to learn more about PurchaseOrder, you would open the file `po.xsd`.



21. Close the `ReceiveData.wsdl` file and the Component—RouteData—Properties window.

Practices for Lesson 3: Managing and Monitoring

Practices for Lesson 3: Overview

Practices Overview

In this practice, you first deploy the HelloWorld composite application project to the SOA server and use the Enterprise Manager web application to test the service with a sample input. You then modify the HelloWorld File adapter to use a logical name and rerun your test. Finally, you create a configuration plan for the HelloWorld composite application to replace the destination directory for the orderfiles logical name reference property.

Practice 3-1: Deploying and Testing the HelloWorld Composite Application

Overview

In this practice, you deploy the HelloWorld composite application project to the SOA server and use the Enterprise Manager web application to test the service with a sample input. To complete this task, perform the following steps:

Assumptions

This practice assumes that you have completed Practice 2 successfully.

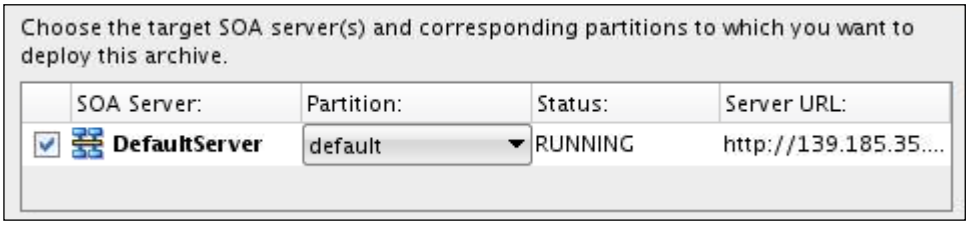
Tasks

Deploying the HelloWorld Composite Application

1. In the JDeveloper Application Navigator, right-click the HelloWorld project and select Deploy > HelloWorld.

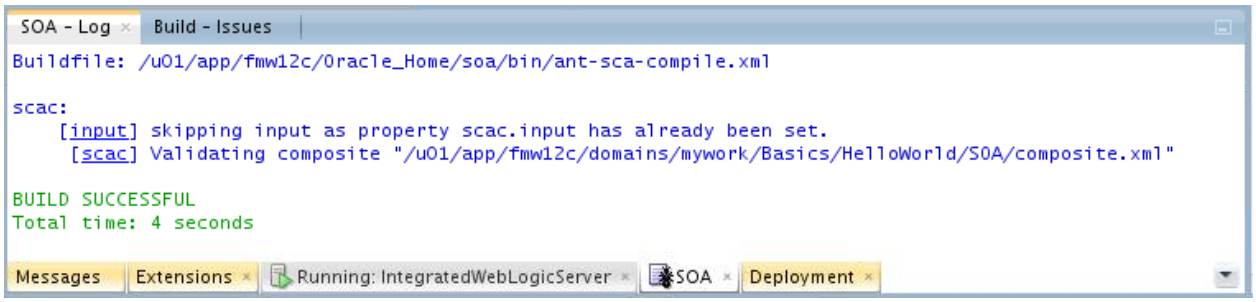
   The Deploy HelloWorld wizard opens.
2. Use the instructions in the following table to complete the deployment:

Step	Window/Page Description	Choices or Values
a.	Deployment Action	Deploy to Application Server. Click Next.
b.	Deploy Configuration	Select the “Overwrite any existing composites with the same revision ID” check box. Click Next.
c.	Select Server	IntegratedWebLogicServer Click Next.
d.	SOA Servers	Verify that the DefaultServer status is RUNNING as shown in the following screenshot. Click Next.
e.	Summary	Click Finish.

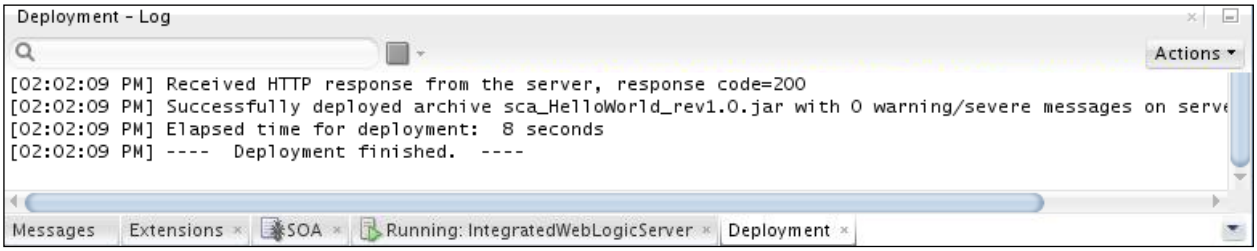


Deployment processing starts.

3. Monitor deployment progress and check for successful compilation in the SOA – Log window.



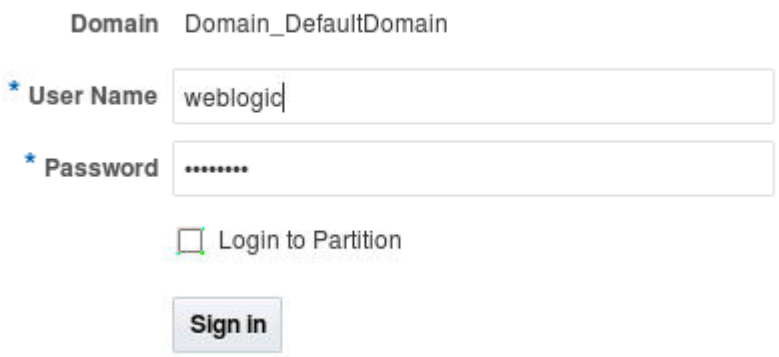
4. Verify that deployment is successful in the Deployment – Log window, similar to what is shown in the following screenshot:



Testing HelloWorld by Using Enterprise Manager Web Interface

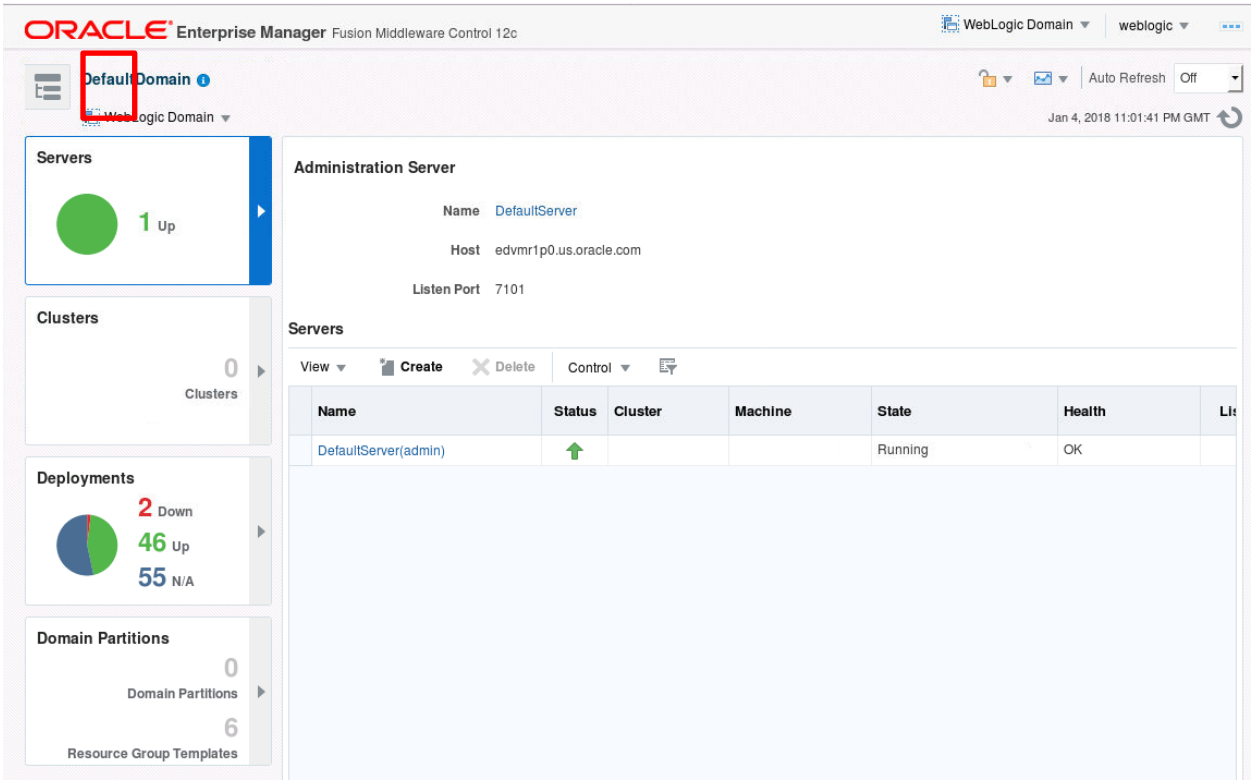
In this section, you use a web browser to access Oracle Enterprise Manager 12*c* Fusion Middleware Control and initiate a test of the deployed HelloWorld application.

5. Log in to the Oracle Enterprise Manager Fusion Middleware Control 12*c* application by performing the following steps:
   a. Open a web browser and provide the URL http://localhost:7101/em.
   b. Log in by using the username `weblogic` and the password `welcome1`.

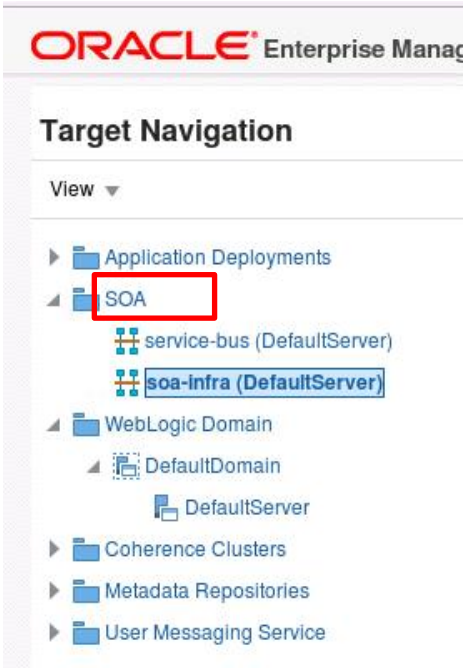


On the Enterprise Manager Fusion Middleware Control Home page,

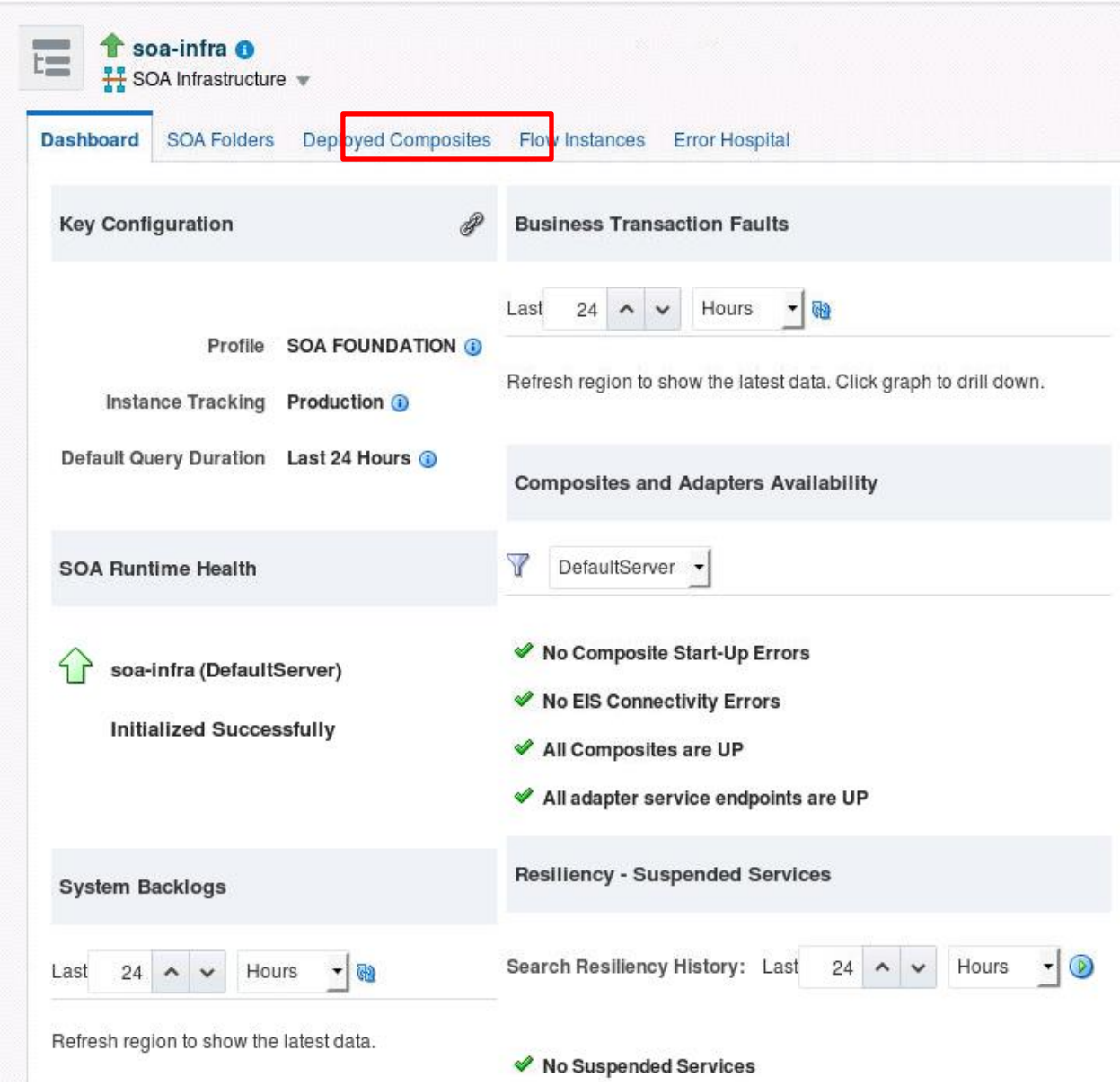
c. Click the Target Navigation link to access the SOA folder.



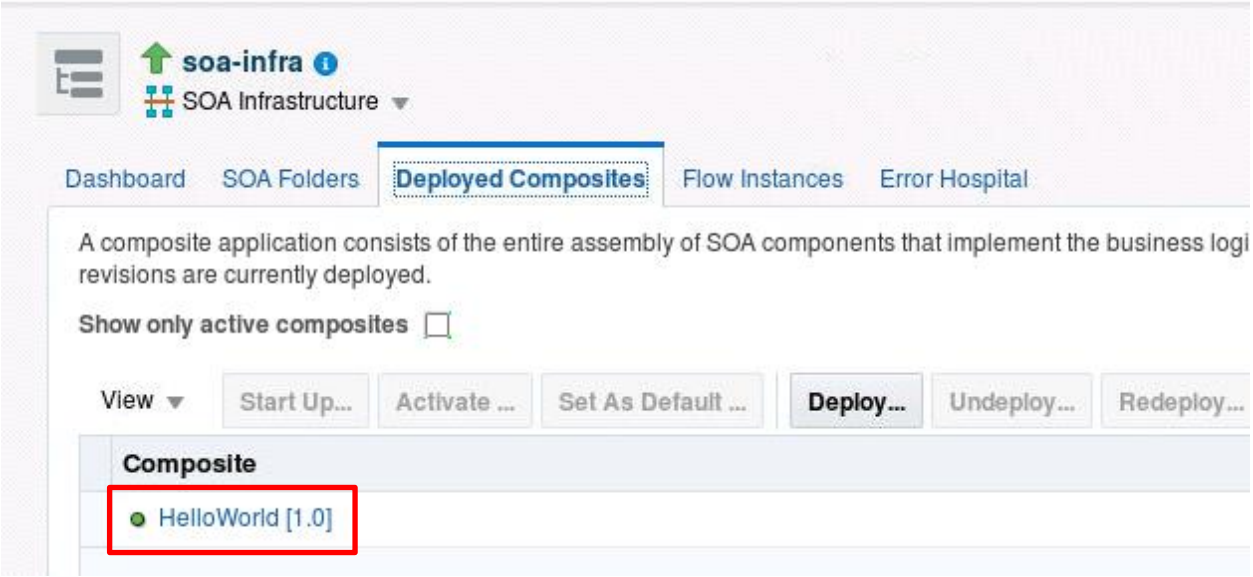
d. In the Target Navigation pane, expand the SOA folder and click the soa-infra (DefaultServer) link.



e. The soa-infra dashboard displays. If it does not, click the Target Navigation icon to cause the panel to hide). From here you can view the current status of the SOA Infrastructure and select the appropriate tabs to view the SOA folders, Deployed Composites, search for Flow Instances and see faults in the Error Hospital. Since you want to view the HellowWorld composite you just deployed, click the Deployed Composites tab.



6. The list of the composites deployed to the default folder are displayed.

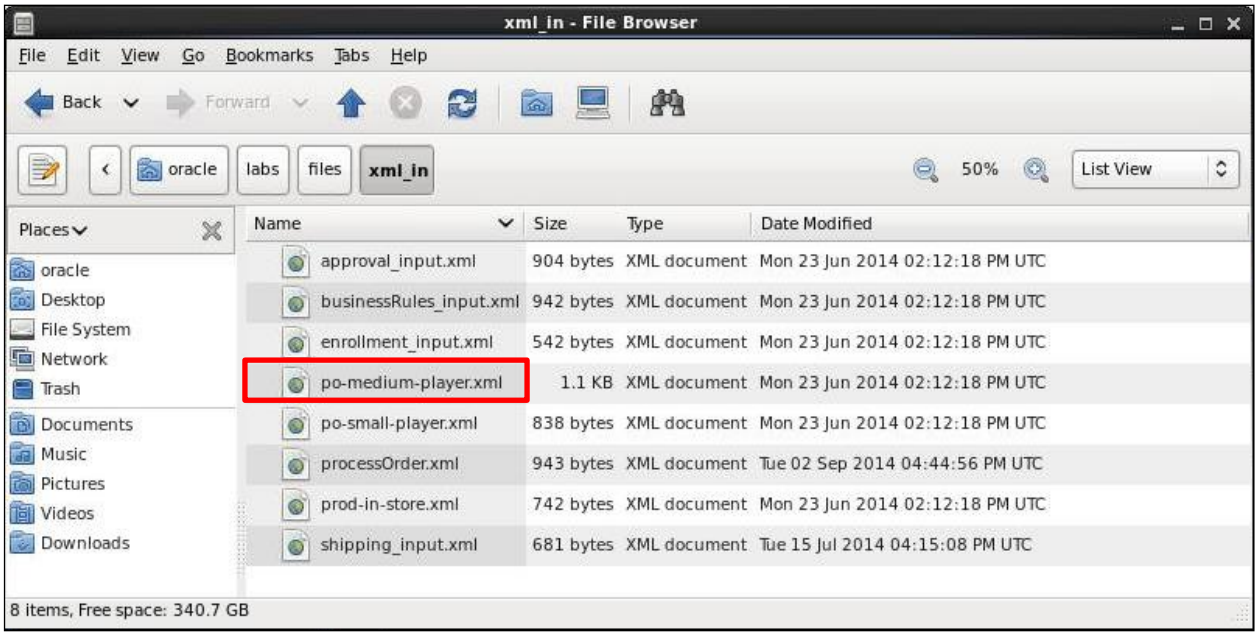


a. To initiate a test of the HelloWorld composite, perform the following steps:

b. Click the “HelloWorld [1.0]” composite to open its dashboard page and then click Test.

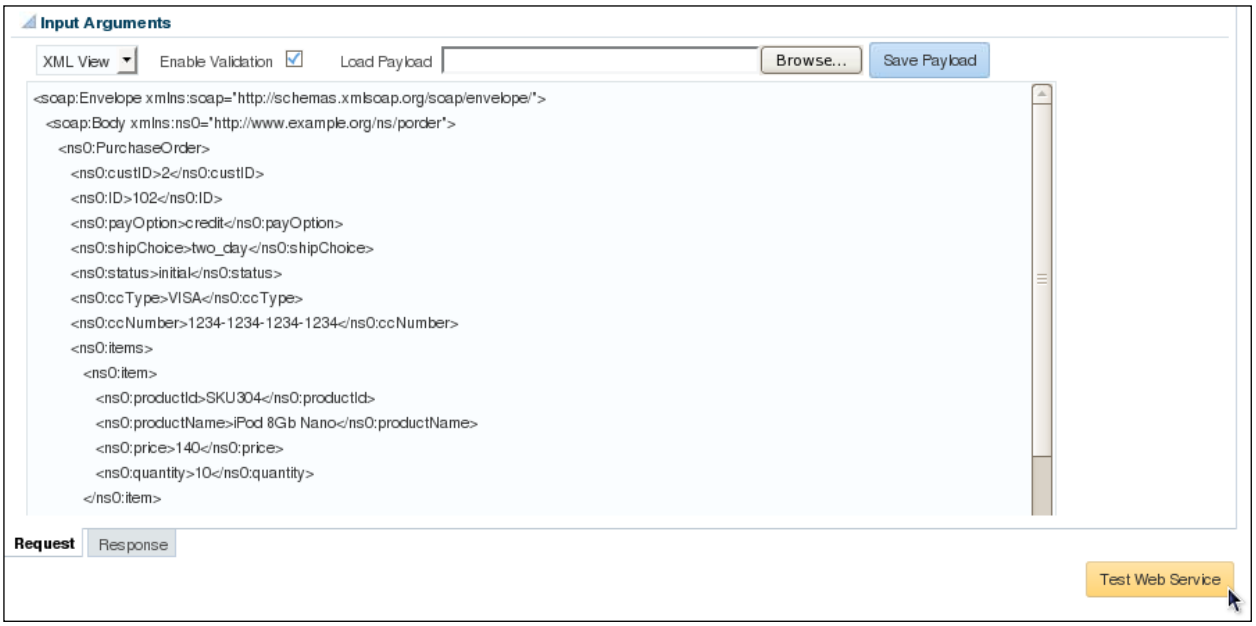
c. On the Request tab, click the Browse button and navigate to the `/home/oracle/labs/files/xml_in` folder.

d. Select `po-medium-player.xml`. Click Open.



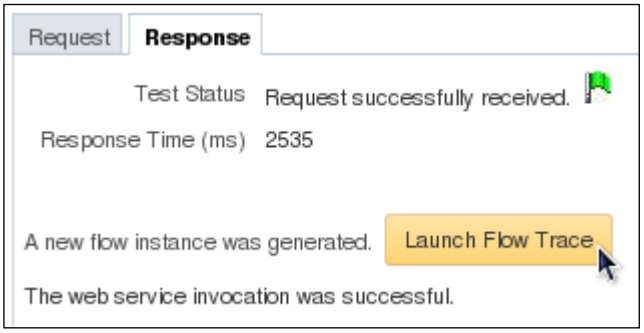
The initially supplied XML text is replaced by the contents of the file.

7. Click Test Web Service.

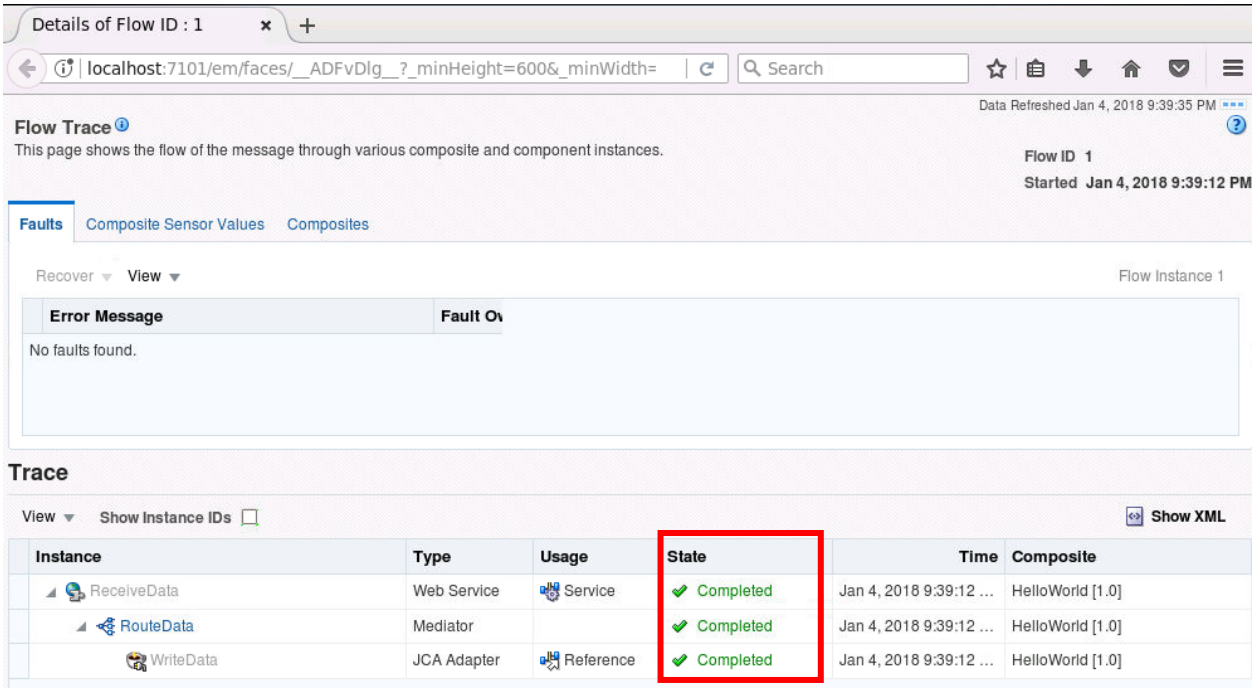


Note: This action sends the XML data as an input message to the HelloWorld composite application. It might take several seconds for the page to refresh with the response that is displayed on the Response tab. Wait until the page is refreshed.

8. On the “HelloWorld [1.0]” Response tab page, click the Launch Flow Trace button to view the results of the asynchronous composite application.

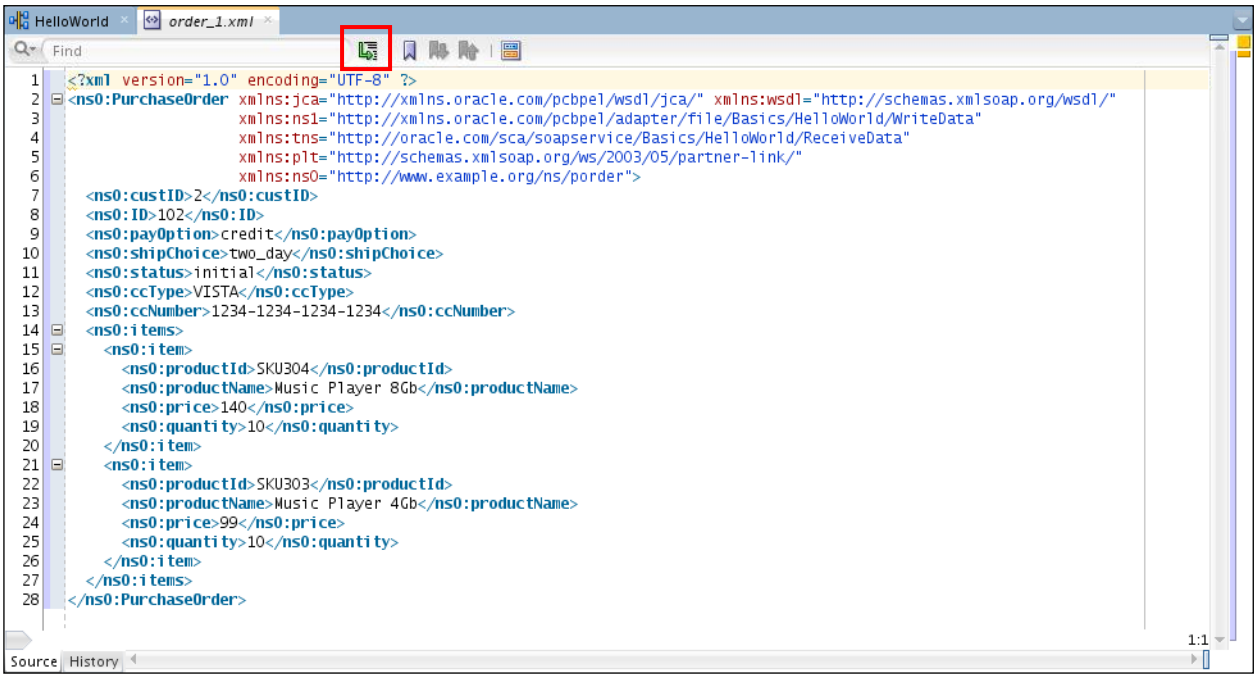


9. On the Flow Trace page, verify that the ReceiveData, RouteData, and WriteData components have a Completed state, indicating that the application executed successfully.



10. Close the Flow Trace window and minimize your browser.
11. Using JDeveloper, open the output file `/home/oracle/labs/output/podata/order_1.xml` to view its contents.
12. Click the Reformat icon.

13. Verify that the XML file contains data similar (or identical) to the following screenshot:



14. In the JDeveloper window, close the `order_1.xml` window.

Practice 3-2: Modifying the HelloWorld File Adapter to Use a Logical Name

Overview

In this practice, you modify the HelloWorld composite application and alter the File adapter to use a logical name instead of an explicit directory for orders written to the file system. You then create a binding property that is to set the directory to the `/home/oracle/labs` directory, deploy the application by using JDeveloper, and test that the order is written to a file in the `/home/oracle/labs` directory.

Assumptions

This practice assumes that you have completed Practice 3-1 successfully.

Tasks

Modifying the File Adapter to Use a Logical Name

1. If the HelloWorld overview window is not already open, in the JDeveloper Application Navigator, expand the HelloWorld project and open the `HelloWorld` file.
2. Right-click the WriteData external reference icon and select Edit.
3. Edit the File adapter external reference to use a logical name instead of a physical path. Use the following table of instructions as a guide:

Step	Window Description	Choices or Values
a.	File Adapter Reference	Click Next.
b.	Adapter Interface	Click Next.
c.	File Server Connection	Click Next.
d.	Operation	Click Next.
e.	File Configuration	Directory Specified as: Logical Name Directory for outgoing files (logical name): `orderfiles` Click Next.
f.	Messages	Click Next.
g.	Finish	Click Finish.

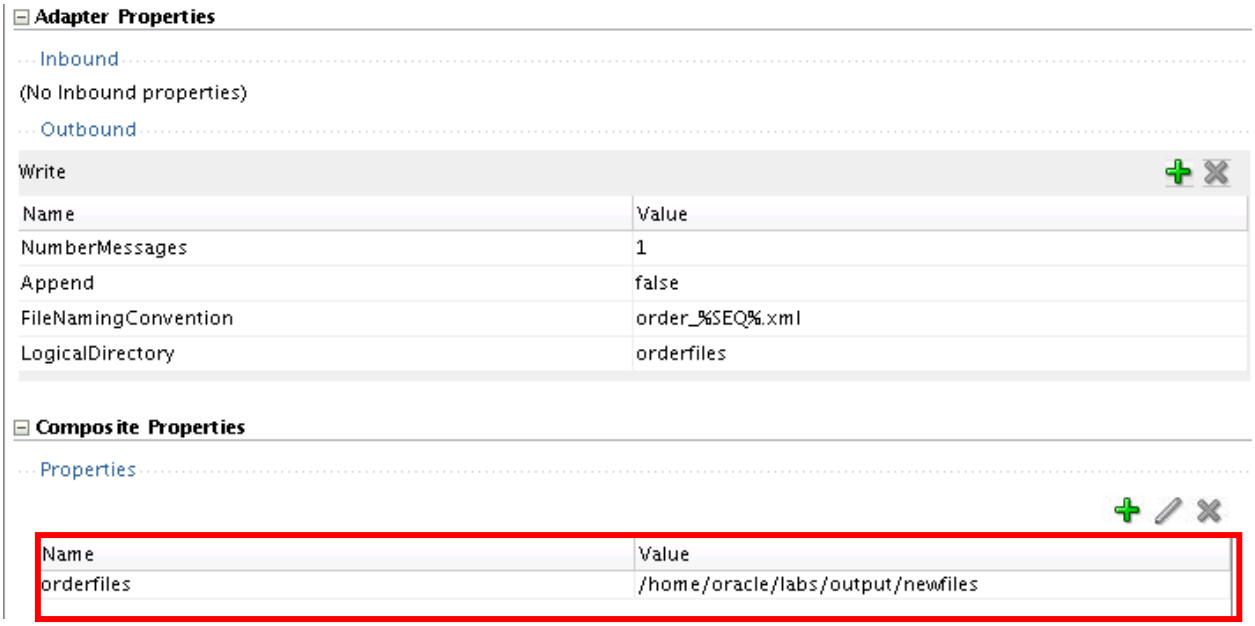
4. To set the path value for the `orderfiles` logical name, perform the following steps:
   a. To display the Properties pane, in the JDeveloper main menu, select Window > Properties.

      The Property Inspector for the Reference, WriteData, is displayed in a tabbed pane.
   b. Expand the Adapter and Composite properties.

      The adapter includes a reference property called LogicalDirectory with the value `orderfiles`.

c. Set the `orderfiles` reference property. In the Composite properties, click in the Value cell and enter the string value `/home/oracle/labs/output/newfiles`.

d. Press Enter.



5. Select File > Save All to save the changes to the application.

Deploying the Modified HelloWorld Composite Application

6. In the JDeveloper window, redeploy the modified HelloWorld project by performing the following steps:

   a. In the Application Navigator, right-click the HelloWorld project name and select Deploy > HelloWorld to IntegratedWebLogicServer.

   b. In the Deployment – Log window, observe the log messages to ensure that deployment was successful.

Testing the Modified HelloWorld Composite Application

7. Restore or re-open your browser and access Enterprise Manager. (The URL is http://localhost:7101/em. Log in as `weblogic` with the password `welcome1`.)

   Enterprise Manager opens.

   In the Target Navigation pane, expand the SOA folder and click the soa-infra (DefaultServer) link.

8. Click the Deployed Composites tab. Click “HelloWorld.”

9. On the “HelloWorld [1.0]” page, click Test.

   a. In the Input Arguments section, in Tree View mode, enter the following values:

custID:	`1`
ID:	`2`
payOption:	`credit`

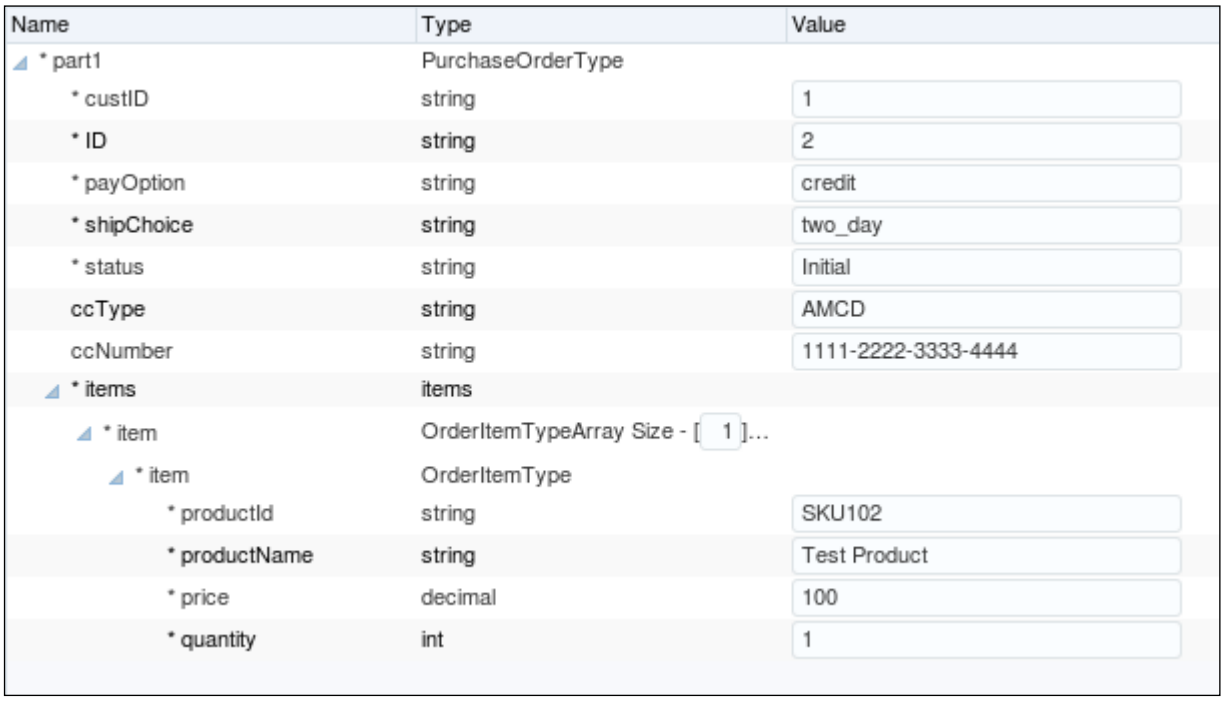
shipChoice:	`two_day`
status:	`Initial`
ccType:	`AMCD`
ccNumber:	`1111-2222-3333-4444`

b. While still in Tree View mode, expand the “items” field, and in the item field, enter the value `1` in OrderItemTypeArray Size. Click the Refresh icon . Collapse the Items array, then re-expand it. Now you are ready to enter the details for a single item.

c. Expand the OrderItemTypeArray item and the OrderItemType item, and enter the following field values:

productId:	`SKU102`
productName:	`Test Product`
price:	`100`
quantity:	`1`

d. Verify your input.

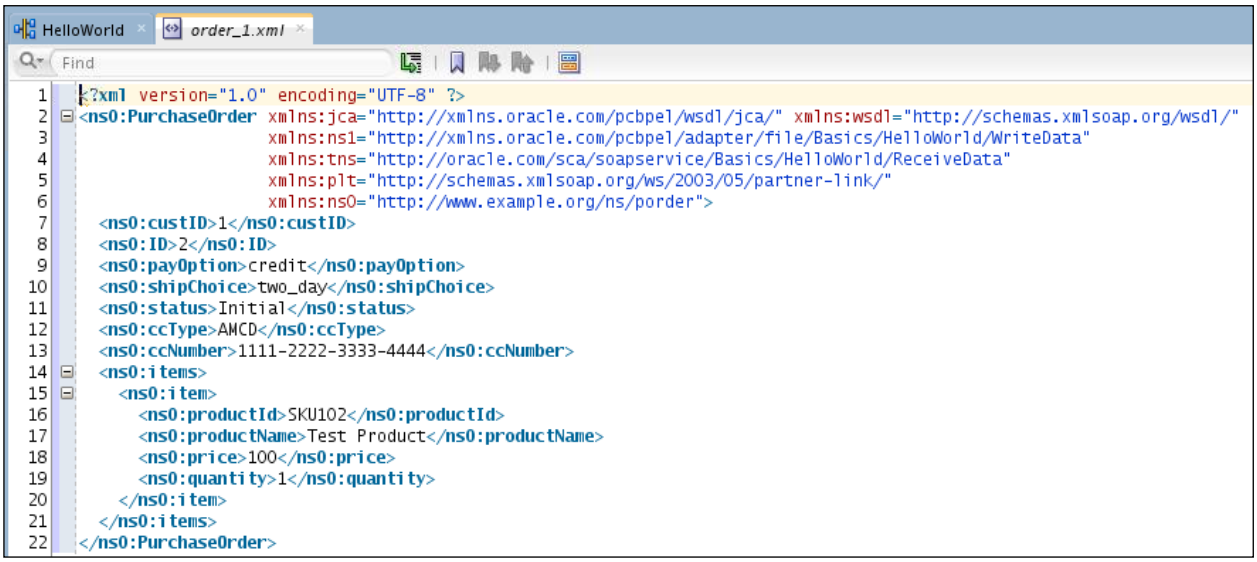


10. Click Test Web Service.

    The web service is tested. The “HelloWorld [1.0]” Response tab page is displayed. The message “Request successfully received.” is displayed.

11. In JDeveloper, open the file `/home/oracle/labs/output/newfiles/order_1.xml`.

12. Verify that the file contains the values entered in the test page form fields.



13. In JDeveloper, close the `order_1.xml` file.
14. Close or minimize the open browser windows.

Practice 3-3: Generating and Validating a Configuration Plan

Overview

As you move projects from one environment to another (for example, from testing to production), you typically must modify several environment-specific values, such as JDBC connection strings, host names of various servers, and so on. Configuration plans enable you to modify these values by using a single text (XML) file. The configuration plan is created in either Oracle JDeveloper or from the command line. During process deployment, the configuration plan is used to search the SOA project for values that must be replaced to adapt the project to the next target environment.

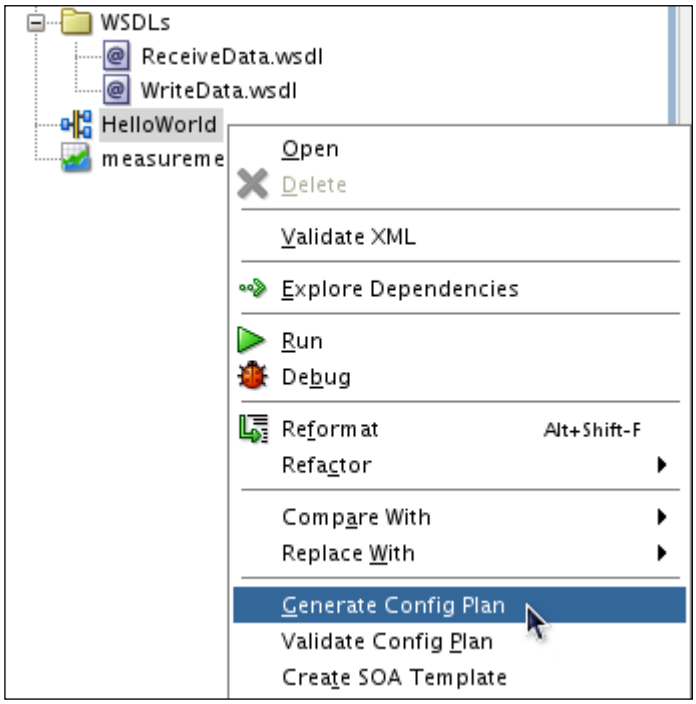
In this practice, you create a configuration plan for the HelloWorld composite application to replace the destination directory for the `orderfiles` logical name reference property. After you verify the configuration plan, you redeploy the composite application by using the Enterprise Manager web interface and applying the configuration plan, and then verify and test the changes.

Assumptions

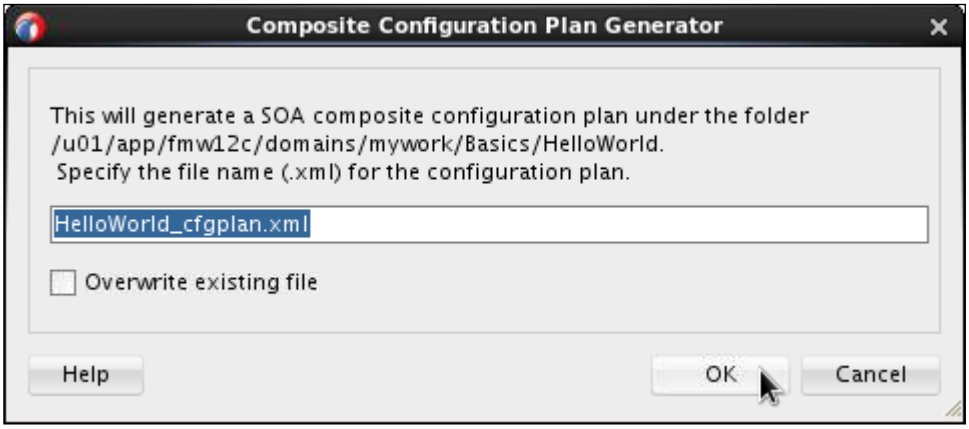
This practice assumes that you have completed Practice 3-2 successfully.

Tasks

1. Generate a configuration plan.
   a. In the Application Navigator, expand the HelloWorld project.
   b. Right-click HelloWorld (the `composite.xml` file) and select Generate Config Plan.



c. In the Composite Configuration Plan Generator dialog box, accept the default plan file name. Click OK.



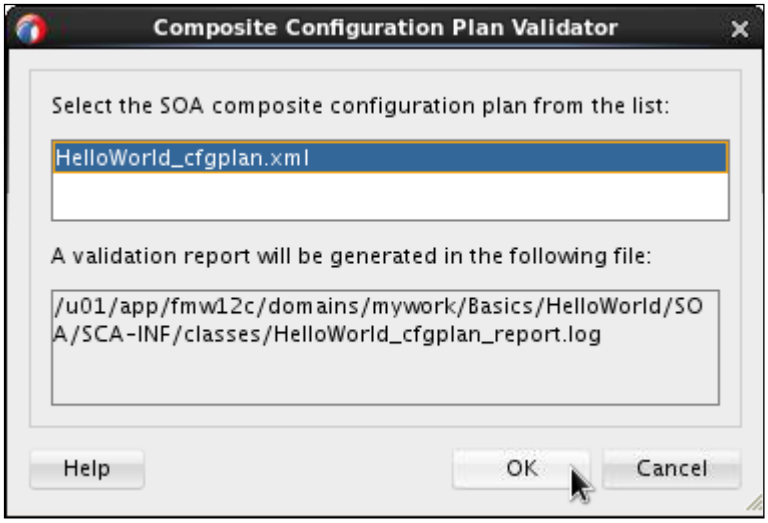
JDeveloper opens the `HelloWorld_cfgplan.xml` file in its own tabbed window.

2. Modify the configuration plan.
   a. In the `HelloWorld_cfgplan.xml` window, scroll down until you locate the `<reference name="WriteData">` element.
   b. Change the value `/home/oracle/labs/output/newfiles` to `/home/oracle/labs/output/podata`.
   c. Compare your file to the following screenshot:

```
45      <reference name="WriteData">
46        <property name="orderfiles">
47          <replace>/home/oracle/labs/output/podata</replace>
48        </property>
```

   d. Save your work.
3. Validate the configuration plan.
   a. In the Application Navigator, right-click HelloWorld (the `composite.xml` file) and select Validate Config Plan.

b. In the Composite Configuration Plan Validator dialog box, accept the default selection and click OK.



c. In the `HelloWorld_cfgplan_report.log` window, verify that the `orderfiles` Reference Property is replaced with the new value of `/home/oracle/labs/output/podata` that is specified in the configuration plan that is being validated.

```
19      Reference  [ WriteData ]
20        Property [ orderfiles ]
21            Old [ /home/oracle/labs/output/newfiles ]
22            New [ /home/oracle/labs/output/podata ]
```

In JDeveloper, close the configuration plan `.xml` and `.log` files.

Practice 3-4: Testing a Composite That Is Deployed with a Configuration Plan

Overview

In this practice, you redeploy the HelloWorld composite application by using the configuration plan file that was generated and validated in Practice 3-3. You also test the deployed application to confirm whether the configuration plan changes are applied.

Assumptions

This practice assumes that you have completed Practice 3-3 successfully.

Tasks

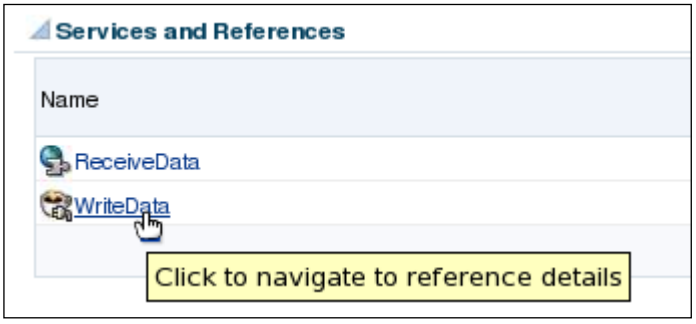
1. Return to your web browser with the Enterprise Manager page and redeploy the HelloWorld composite application by using the deployment plan file that was modified in the previous practice. To complete this task, perform the instructions in the following table:

Step	Screen/Page Description	Choices or Values
a.	Target Navigation Pane	Expand SOA and click soa-infra. From the displayed dashboard, select Deployed Composites Select the line containing “HelloWorld [1.0]”, but do not click the link itself. Select Redeploy.
b.	“HelloWorld [1.0]” Redeploy SOA Composite	In the Archive or Exploded Directory section, accept the default “Archive is on the machine where this Web browser is running” option. Click Browse.
c.	File Upload	Navigate to the `/u01/app/fmw12c/domains/mywork/Basics/HelloWorld/deploy` folder. Select the `sca_HelloWorld.jar` file. Click Open.
d.	“HelloWorld [1.0]” Redeploy SOA Composite	In the Configuration Plan section, select the “Configuration plan is on the machine where this Web browser is running” option. Click Browse.
e.	File Upload	Navigate to the `/u01/apps/fmw12c/domains/mywork/Basics/HelloWorld` folder. Select the `HelloWorld_cfgplan.xml` file.

Step	Screen/Page Description	Choices or Values
		Click Open.
f.	"HelloWorld [1.0]" Redeploy SOA Composite	Scroll back to the top (if needed) and click Next.
g.	"HelloWorld [1.0]" Redeploy SOA Composite	Click Redeploy.

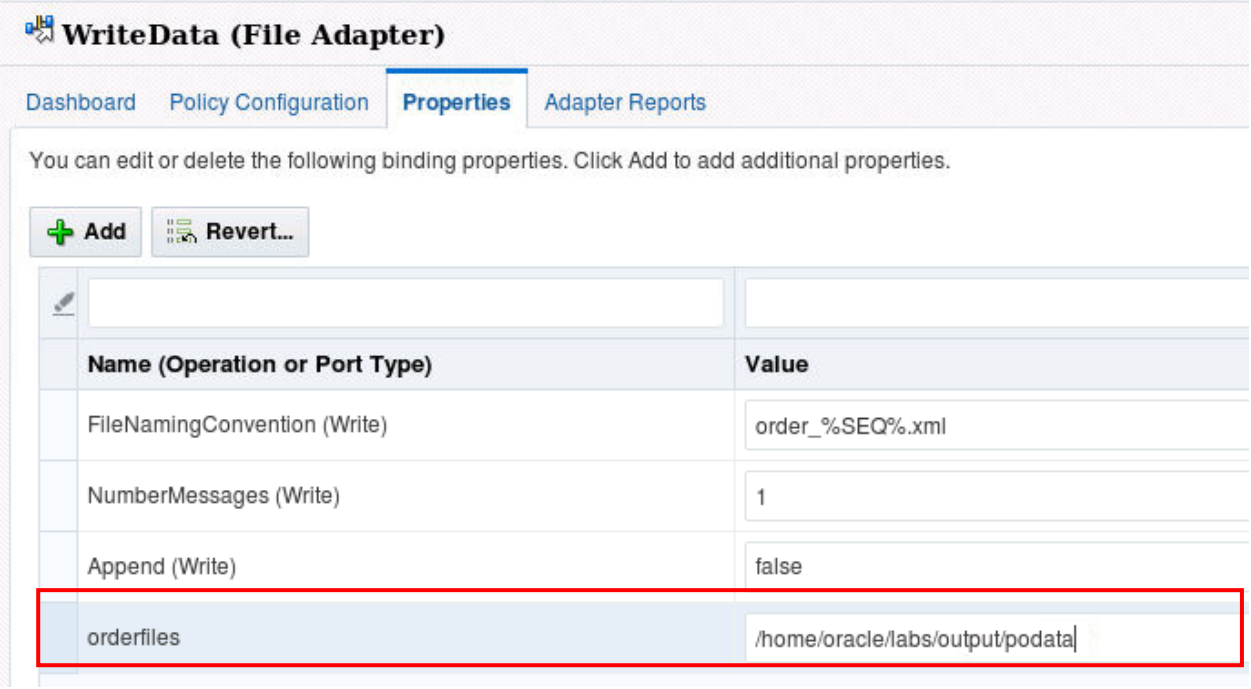
2. The Redeployment Succeeded dialog box is displayed when the deployment successfully completes.

3. Click Close.
4. Click the "HelloWorld [1.0]" link in the Target Navigation pane to view the home page (if needed).
5. In the Services and References pane, click WriteData.



6. Click the Properties tab. Note: If you see an error message you can ignore it. Click Close to dismiss the dialog box.

7. Verify that the value is `/home/oracle/labs/output/podata`, as expected due to redeployment with the configuration plan.



Note: We have not modified the HelloWorld composite application source code and have not even re-created the SOA Archive file during the redeployment process.

8. Test that the HelloWorld composite application now writes order files in the `/home/oracle/labs/output/podata` folder, as defined by the configuration plan.
   a. In Enterprise Manager, on the HelloWorld home page, click Test.
   b. On the Input Arguments Request tab, select the XML View option.
   c. Use the Browse button to replace the supplied XML data with the contents of the `/home/oracle/labs/files/xml_in/po-small-player.xml` file.
   d. Click Test Web Service.
   e. Verify that the new HelloWorld composite application instance completed successfully on the Flow Trace page.
   f. Verify that the `order_2.xml` file has been created with the supplied data in the `/home/oracle/labs/output/podata` folder.

      Note: If more than two XML files exist in the `podata` subfolder, the file with the highest sequence number contains the latest order information.