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

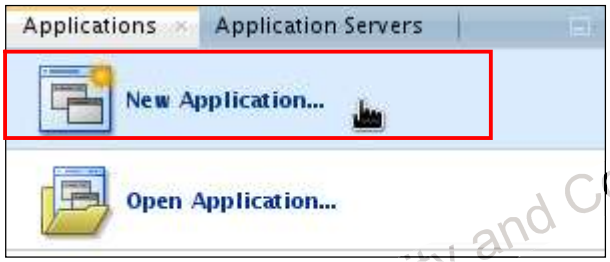
This practice assumes that you have started JDeveloper.

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.

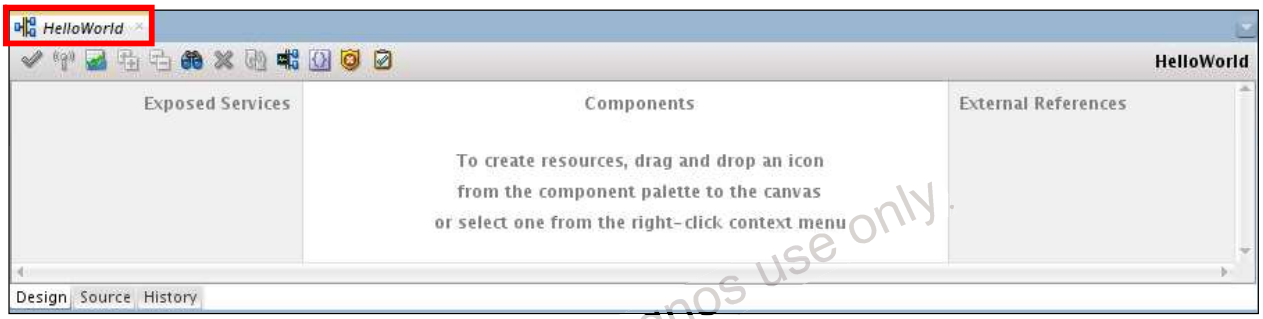
c. Supply `Basics` as the application name and click Next.

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.

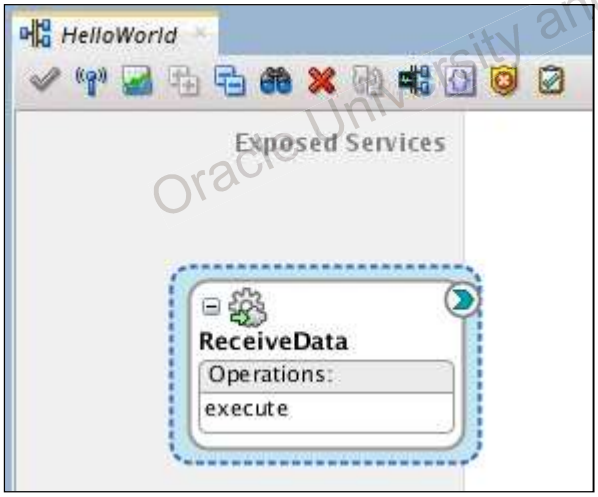
   The Create Web Service dialog box opens.

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 ( ).
b.	Create WSDL	Click the “Add a new message part” icon ( ).

Step	Window/Page Description	Choices or Values
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	Copy Options: Deselect the "Maintain original directory structure for imported files" option. 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, which receives data from the application's client, and then routes that data to a File adapter (which you build in the next step).

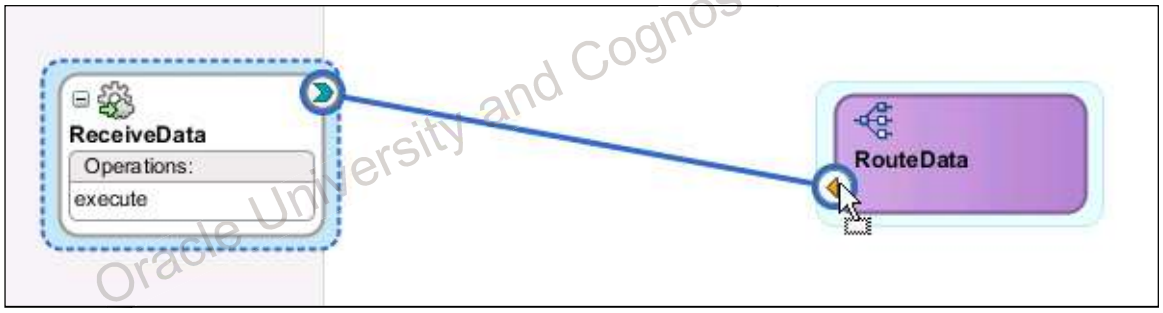
6. Create a Mediator component and wire the ReceiveData service entry point to the Mediator component, by performing the following steps:
   a. Drag a Mediator component from the Component Palette into the Components column of the overview window.

      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.

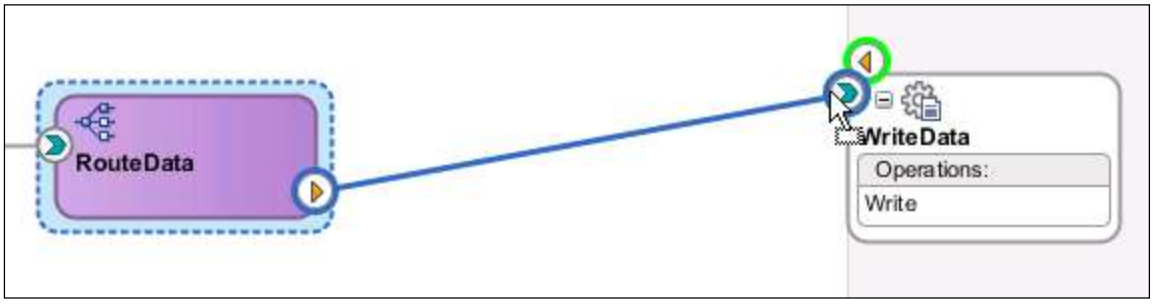
8. In the overview window, drag a File adapter component from the Technology section of the Component Palette onto the External References column.

   The FILE Adapter Configuration wizard opens.

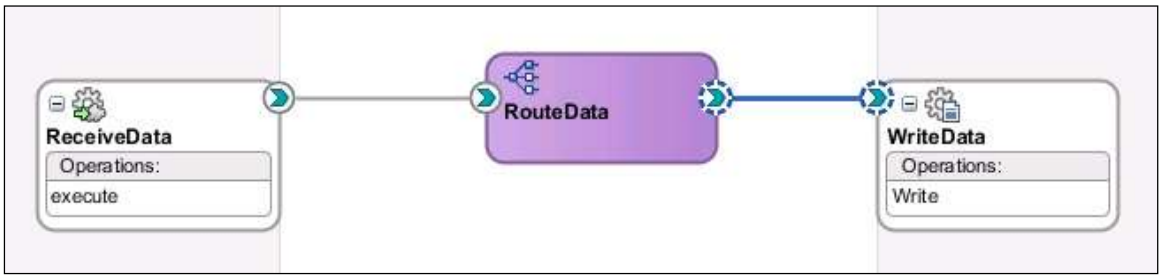
9. On the Adapter Configuration wizard pages, use the instructions in the following table to configure the File Adapter:

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.
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.



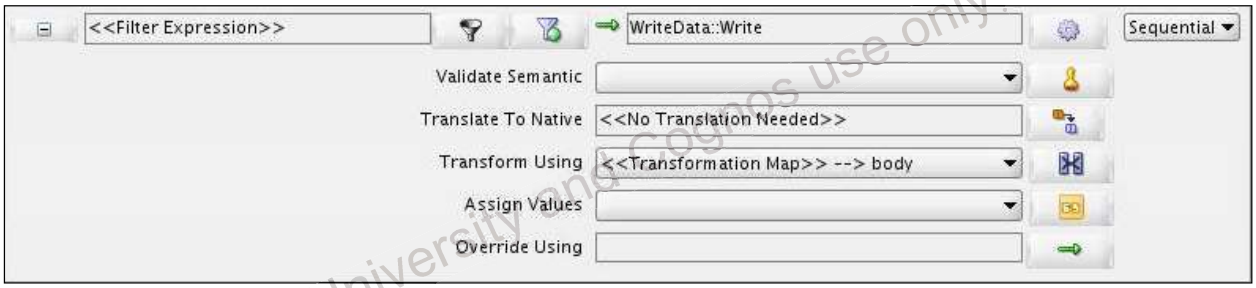
Adding a Transformation to the Mediator

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.

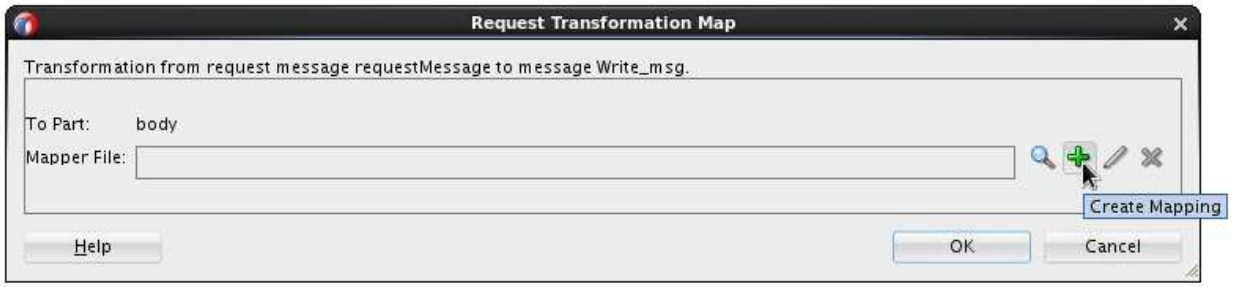
    The Mediator configuration editor (`RouteData.mplan`) opens.

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



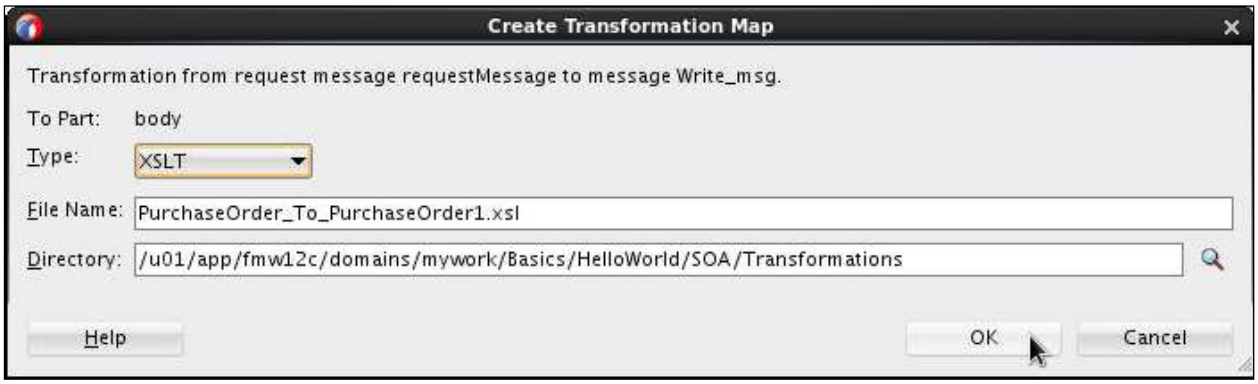
    The Request Transformation Map dialog box is displayed.

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.

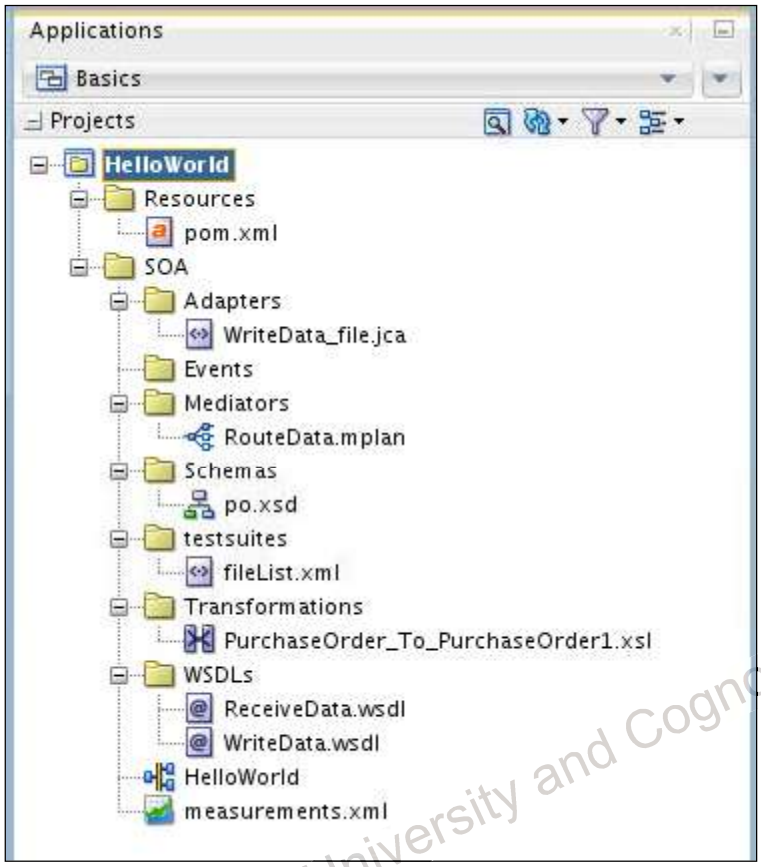
    The AutoMap feature generates an XSLT mapping of the source nodes to the destination nodes.

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.

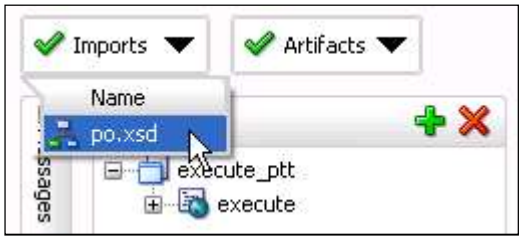
Examining the Generated Configuration Files



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.