2 Creating the Credit Card Validation Service

2.1 Introduction

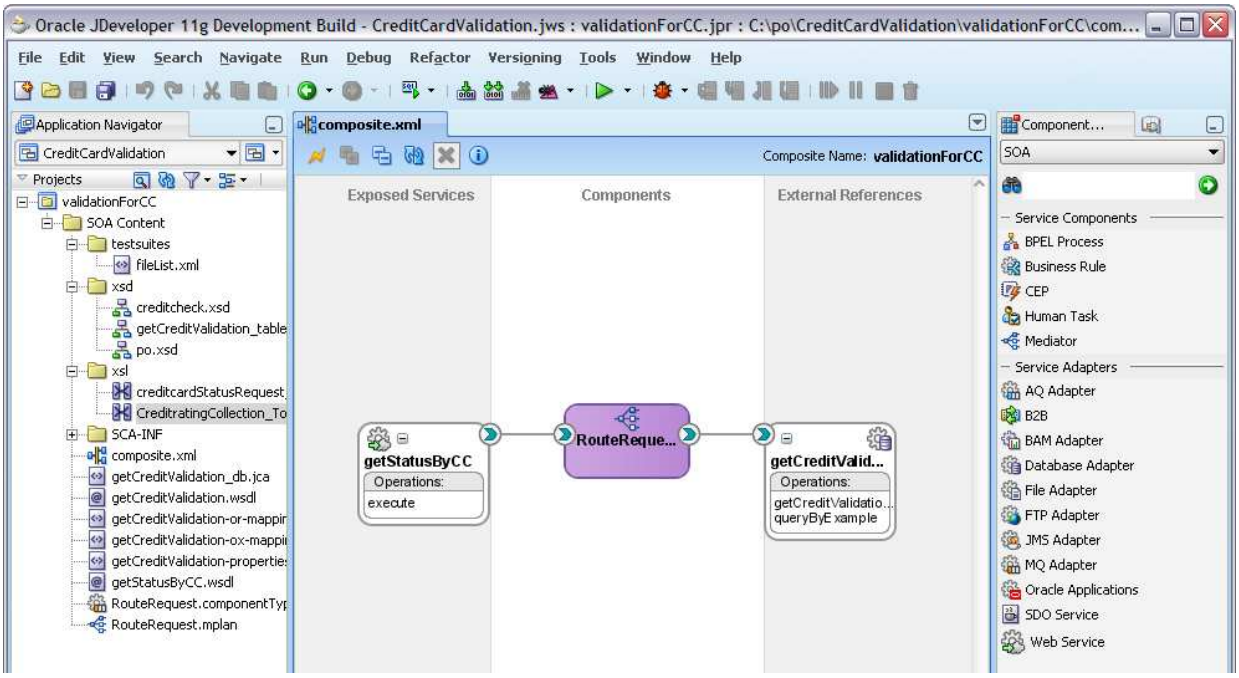
Note: The solution for this chapter can be found in `c:\po\solutions\ch2`.

In this section you will build a simple credit service. This service will look up the status (valid or not valid) of a given credit card from a database. You will learn:

- The basic steps involved in building a SOA composite project
- How to quickly expose SQL operations over SOAP

The implementation of this service is done using an ESB Mediator to route the request to the database adapter which executes the query on the database and returns a result. Once done, our ESB flow will look like Figure 1.

Figure 1 Credit Service



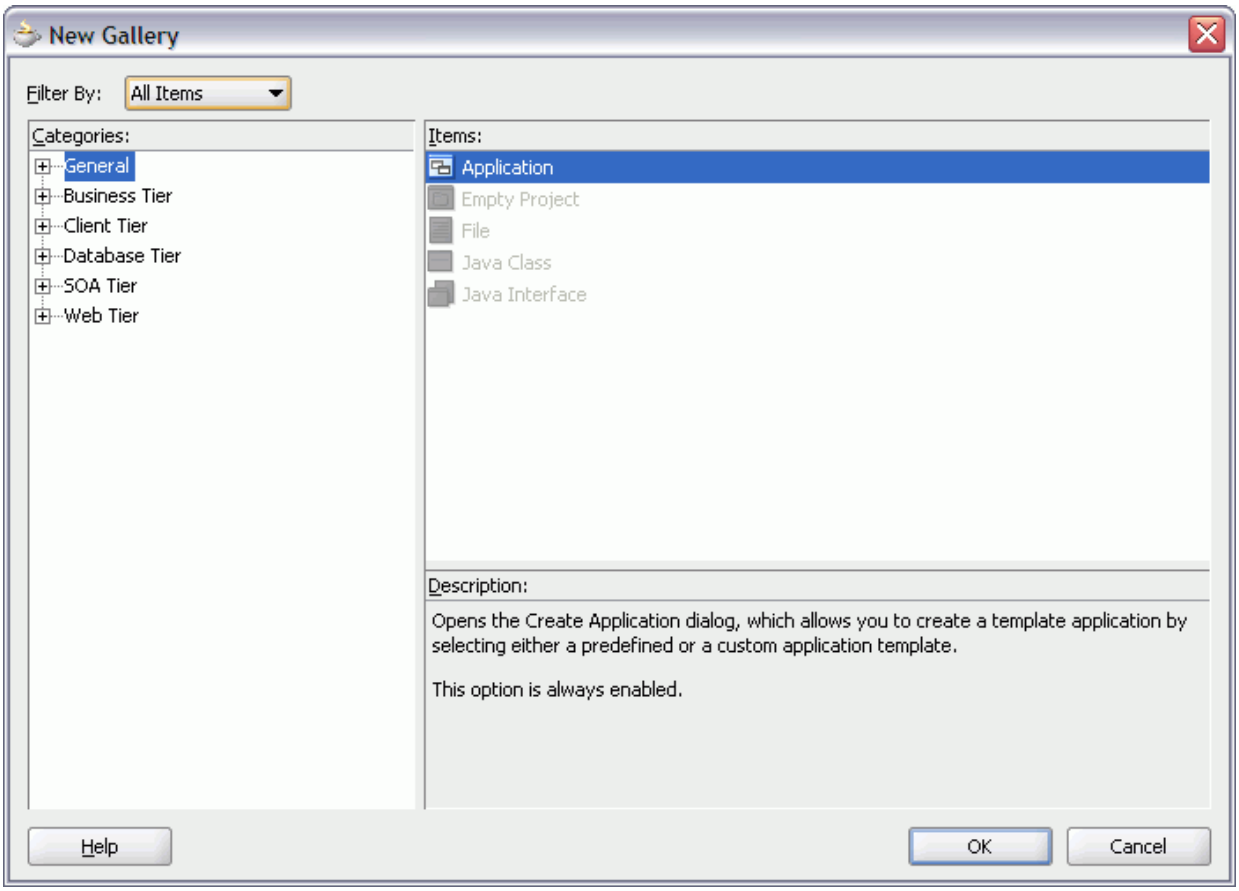
2.2 Designing the flow

First you build the composite application in JDeveloper, then it gets deployed to the server to run and test it.

2.2.1 Creating a new application

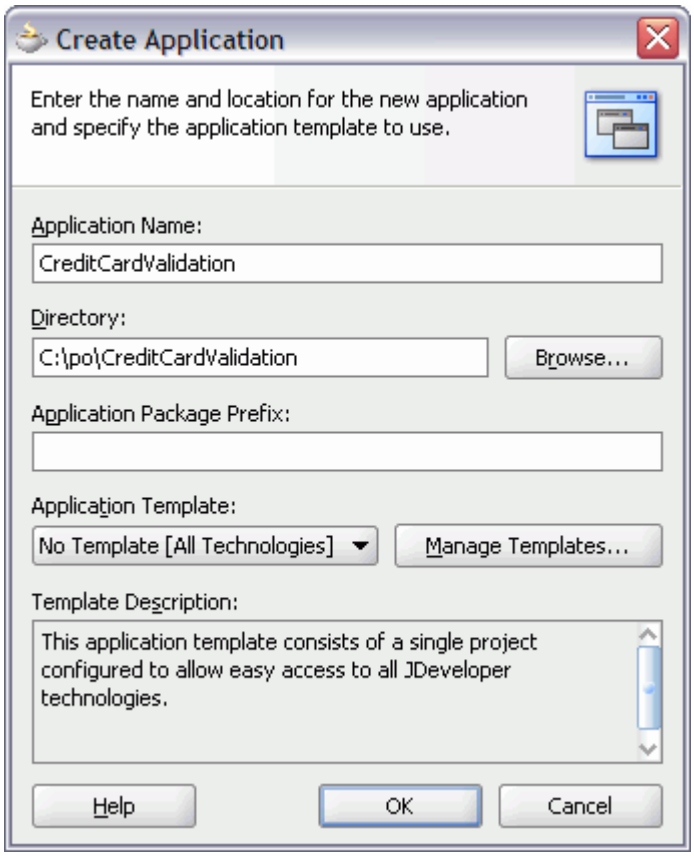
1. Start JDeveloper.
2. Create a new application. There are various ways and shortcuts to do this, and in this case choose **File > New…** from the menu.
3. Change the **Filter By** list to **All Items**.
4. From the **Categories** tree, click on **General** (not expand it).
5. Select **Application** from the **Items** field.

Figure 2 Creating a new application



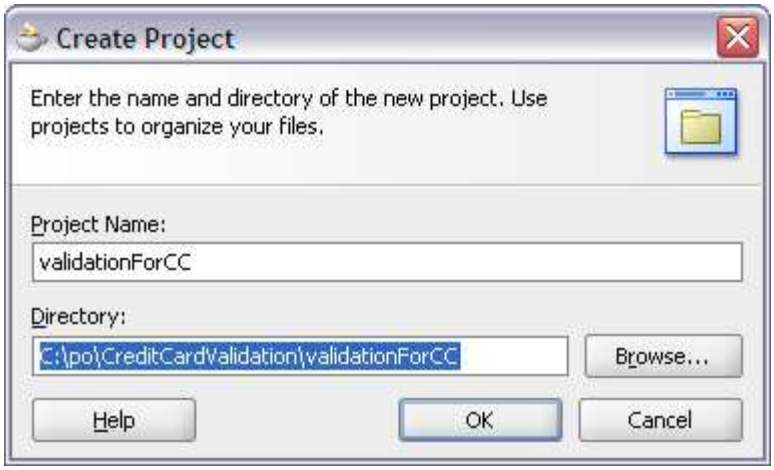
6. Click **OK**.
7. In the subsequent **Create Application** dialog set the following fields, leaving the others with their default values:
   - **Application Name:** CreditCardValidation
   - **Directory:** C:\po\CreditCardValidation

Figure 3 Create application dialog



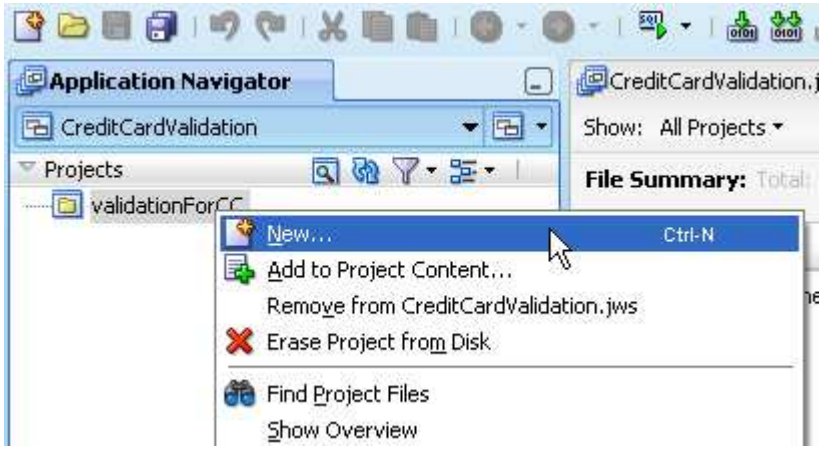
8. Click **OK**.
9. When you create a new application you are prompted to create a new project. Set the following fields:
   - **Project Name:** validationForCC
   - **Directory:** C:\po\CreditCardValidation\validationForCC

Figure 4 New project dialog



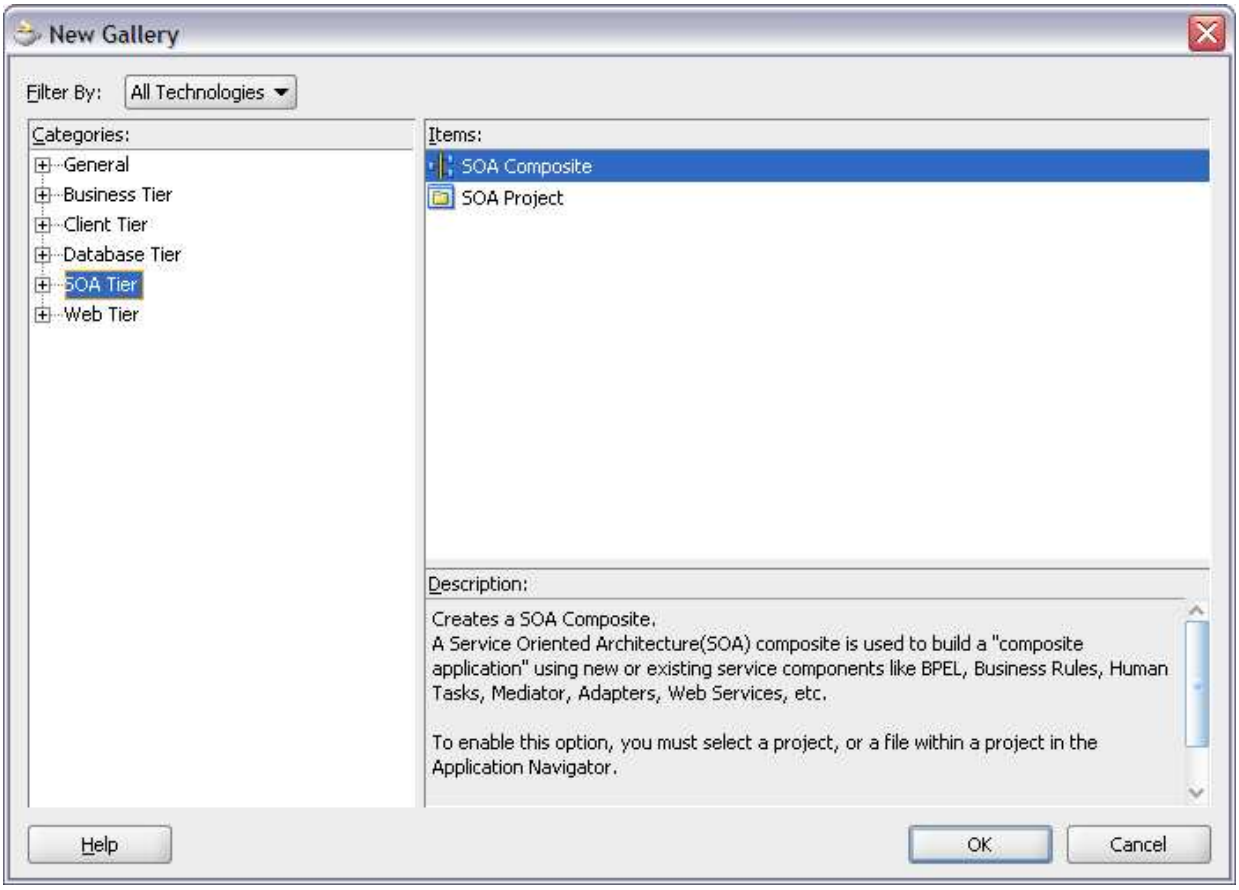
10. This will create a new empty project. Add a composite by right-click the project name **validationForCC** and select **New…**.

Figure 5 Adding new content to a project



11. In the **New Gallery** dialog, select the **SOA Tier** node and then **SOA Composite**.

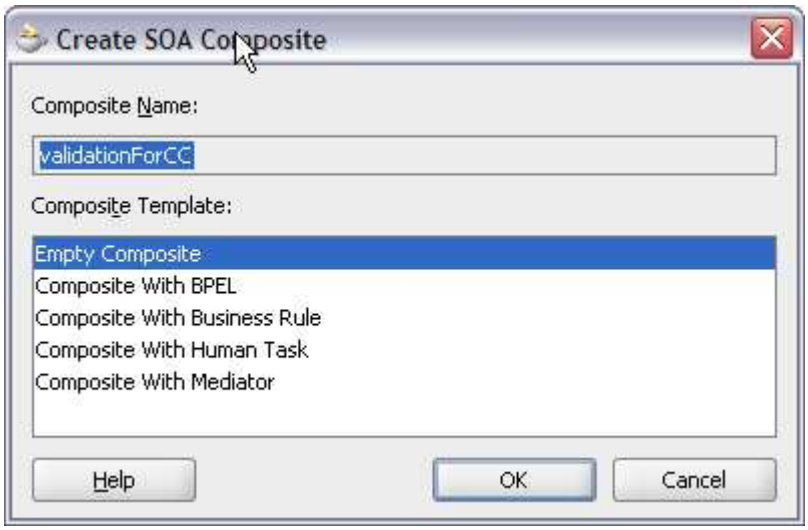
Figure 6 Adding a SOA Composite



12. Click **OK**.

13. A dialog will appear allowing you to start with an empty composite, or with a component already added. In this case, select **Empty Composite**.

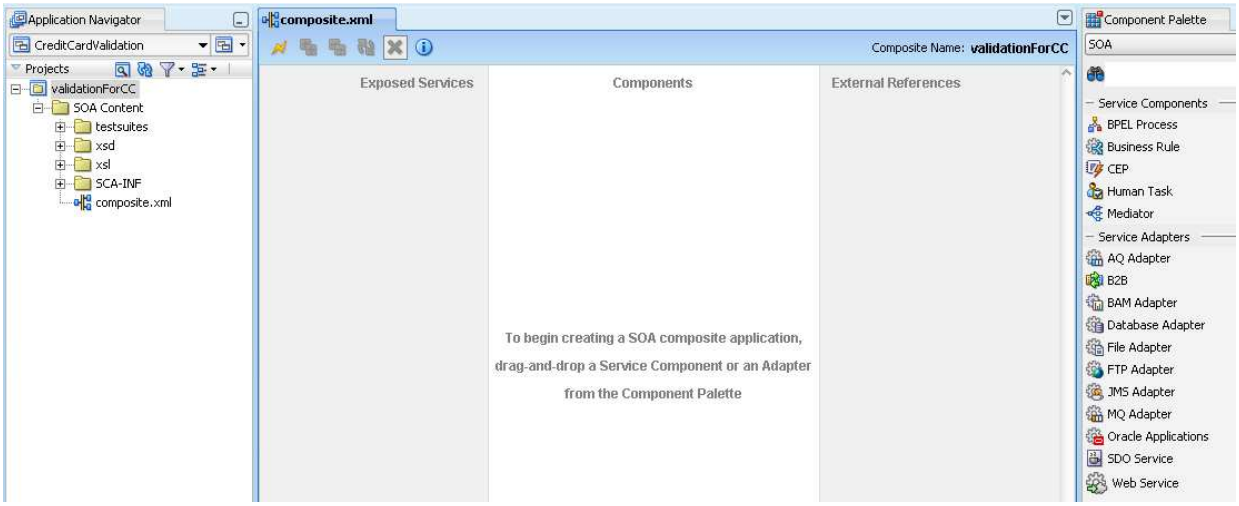
Figure 7 Create SOA Composite dialog



14. Press **OK**.

15. You will now have an empty canvas displaying three swim lanes: services, components, and references.

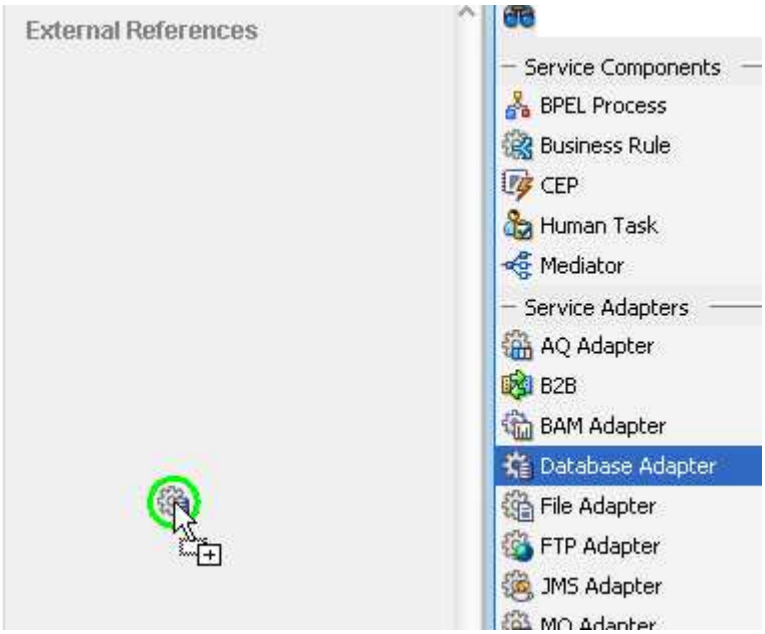
Figure 8 New composite diagram



2.2.2 Adding the database adapter

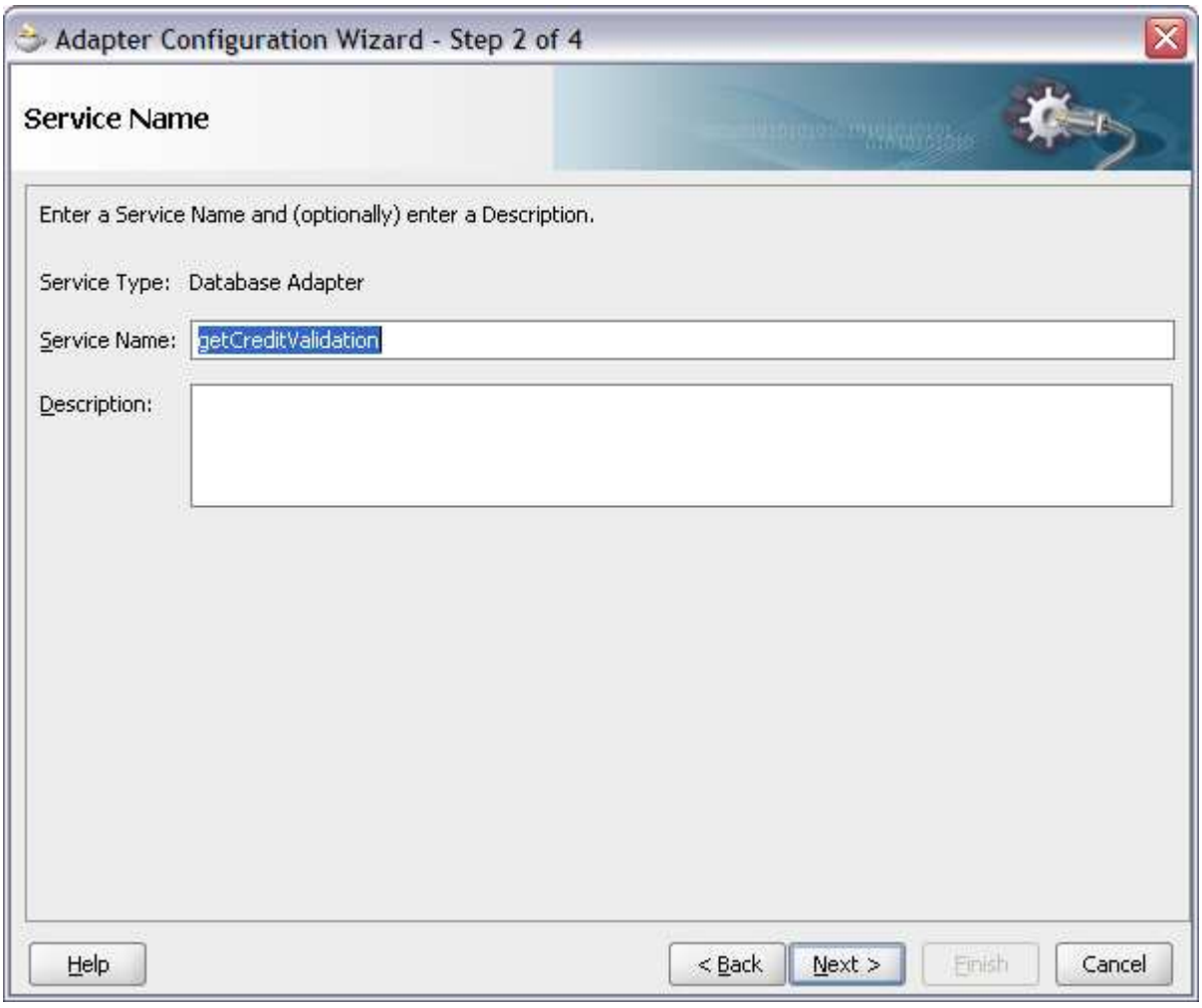
1. You can now start building the application. Drag-and-drop a database adapter onto the **References** swim lane, as shown in Figure 9. If you don't have the Component Palette open, from the menu select **View > Component Palette**.

Figure 9 Dragging a database adapter onto the composite



2. This database adapter call will return a result of valid or invalid for a given credit card from the database. A wizard takes you through the steps of configuring the database adapter. The title bar of the wizard dialog shows the step number. Click **Next** on step 1.
3. Enter the following details on step 2:
   - **Service Name:** getCreditValidation

Figure 10 Database adapter, step 2

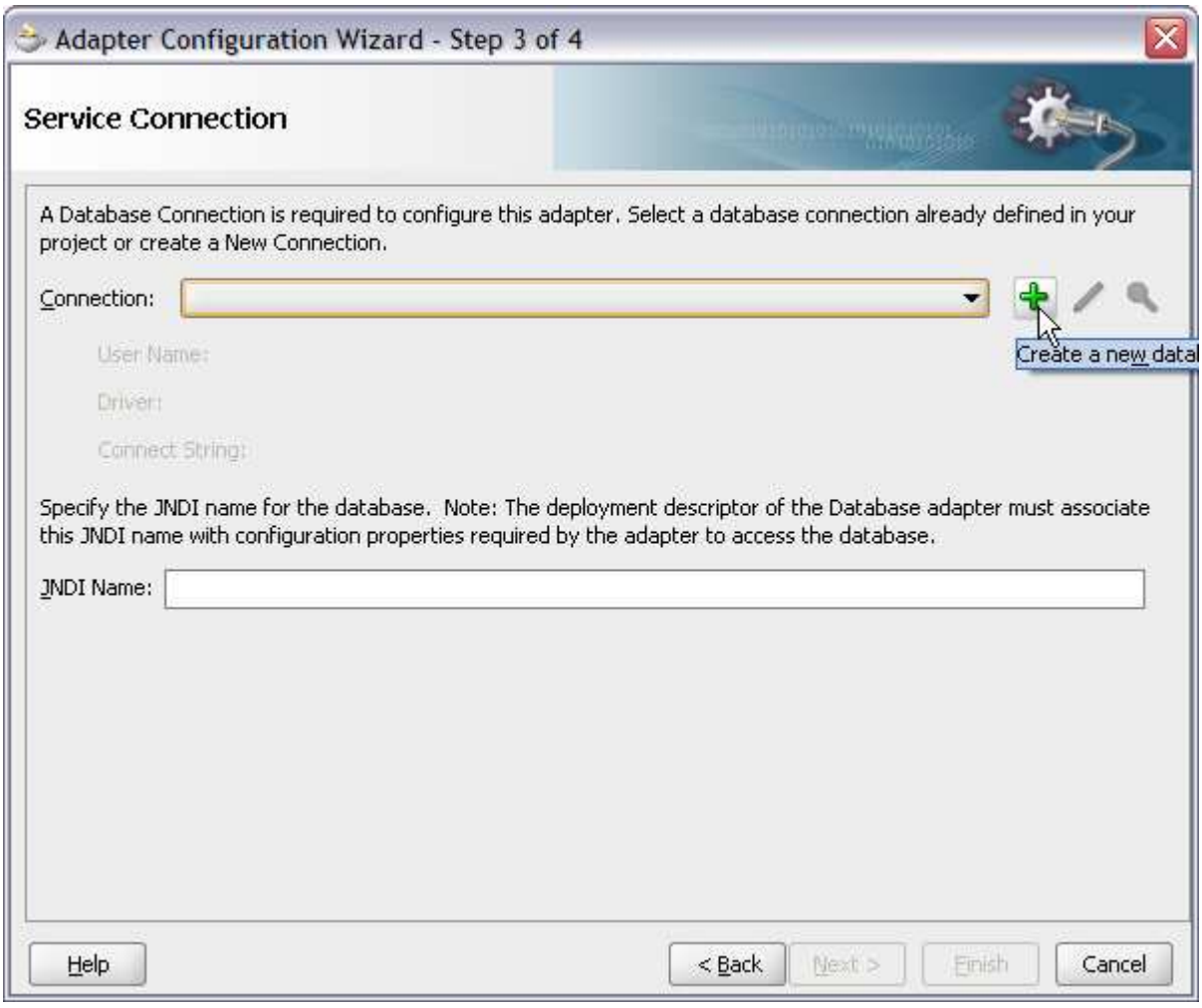


4. Click **Next** to go to step 3.

5. The database adapter will connect to the database, and in order to do that it needs a database connection which contains all the details needed to connect to the database. You can pre-create a database connection or create one on the fly. Since a database connection hasn't been created yet, you will create one from here.

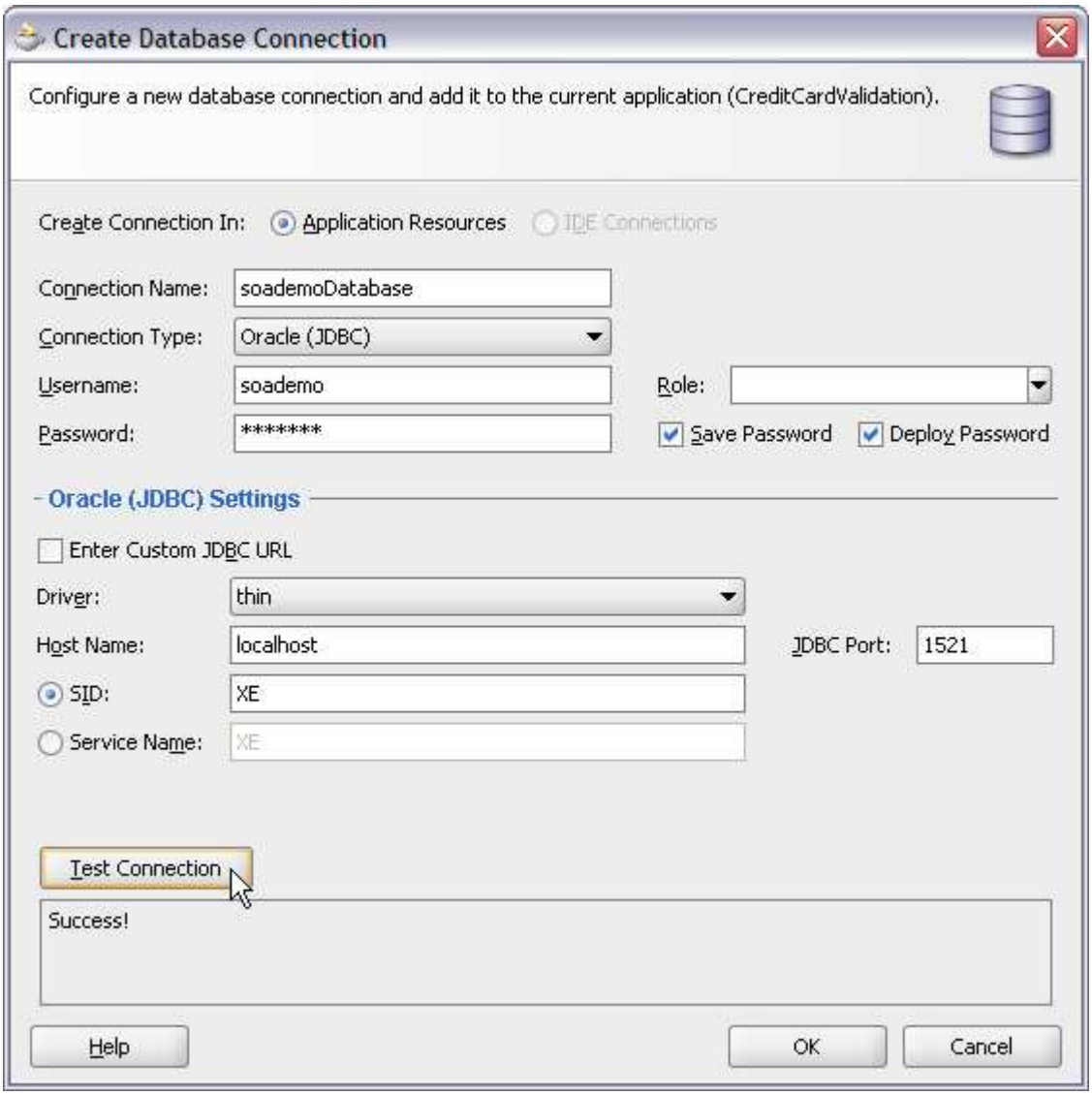
   Click the green plus icon to the right of the **Connection** poplist to create a new database connection.

Figure 11 Creating a new database connection



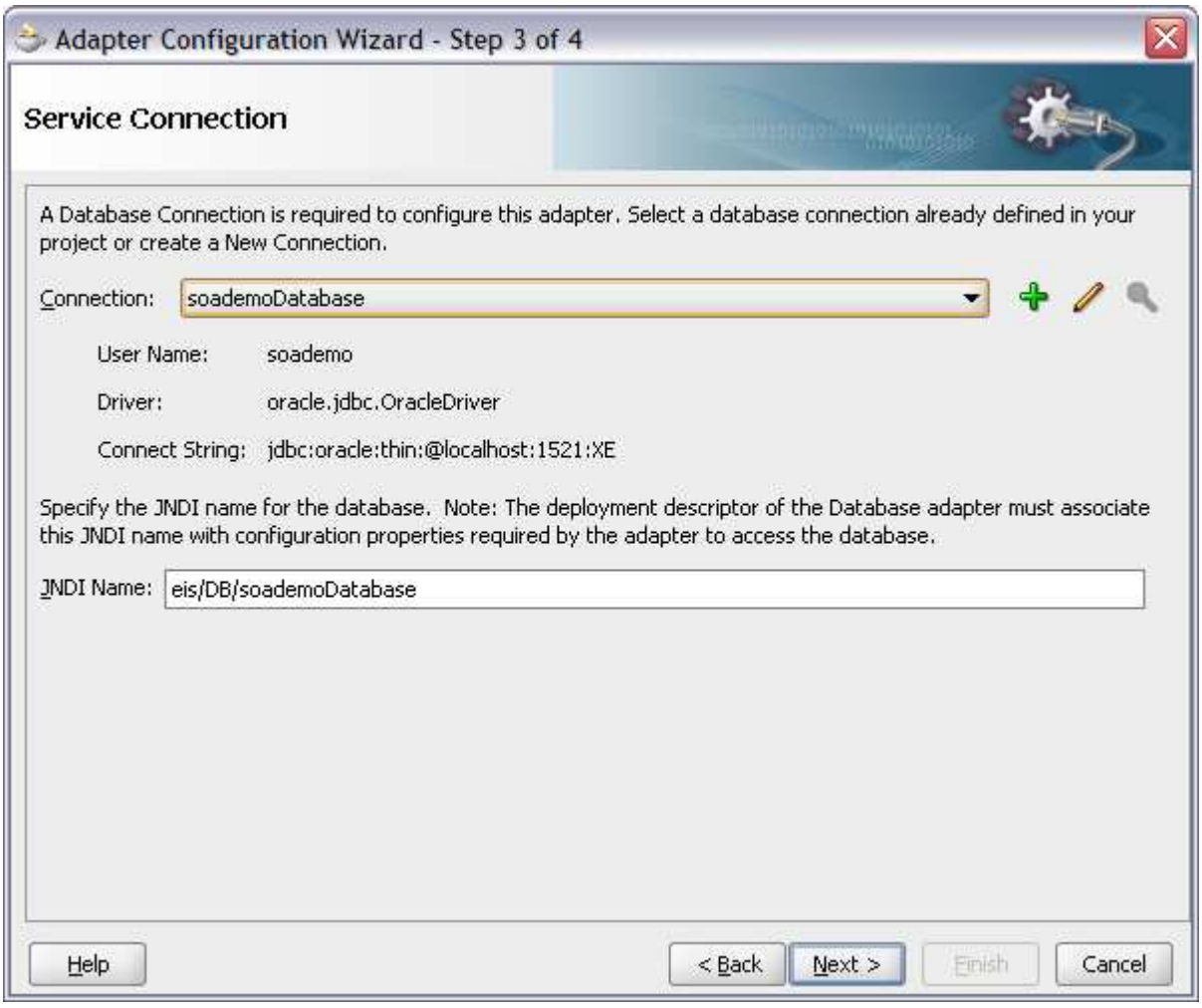
6. In the **Create Database Connection** dialog, enter the following details:
   - **Connection Name:** soademoDatabase
   - **Connection Type:** Oracle (JDBC)
   - **Username:** soademo
   - **Password:** soademo
   - **Save Password:** Checked
   - **Deploy Password:** Checked
   - **Enter Custom JDBC URL:** Unchecked
   - **Driver:** thin
   - **Host Name:** localhost
   - **JDBC Port:** 1521 (or the port number of your database)
   - **SID:** XE (or the SID of your database)

Figure 12 Create Database Connection dialog



7. Click the **Test Connection** button and verify that your connection works.
8. Click **OK** to return to step 3 of the **Database Adapter Configuration** wizard.

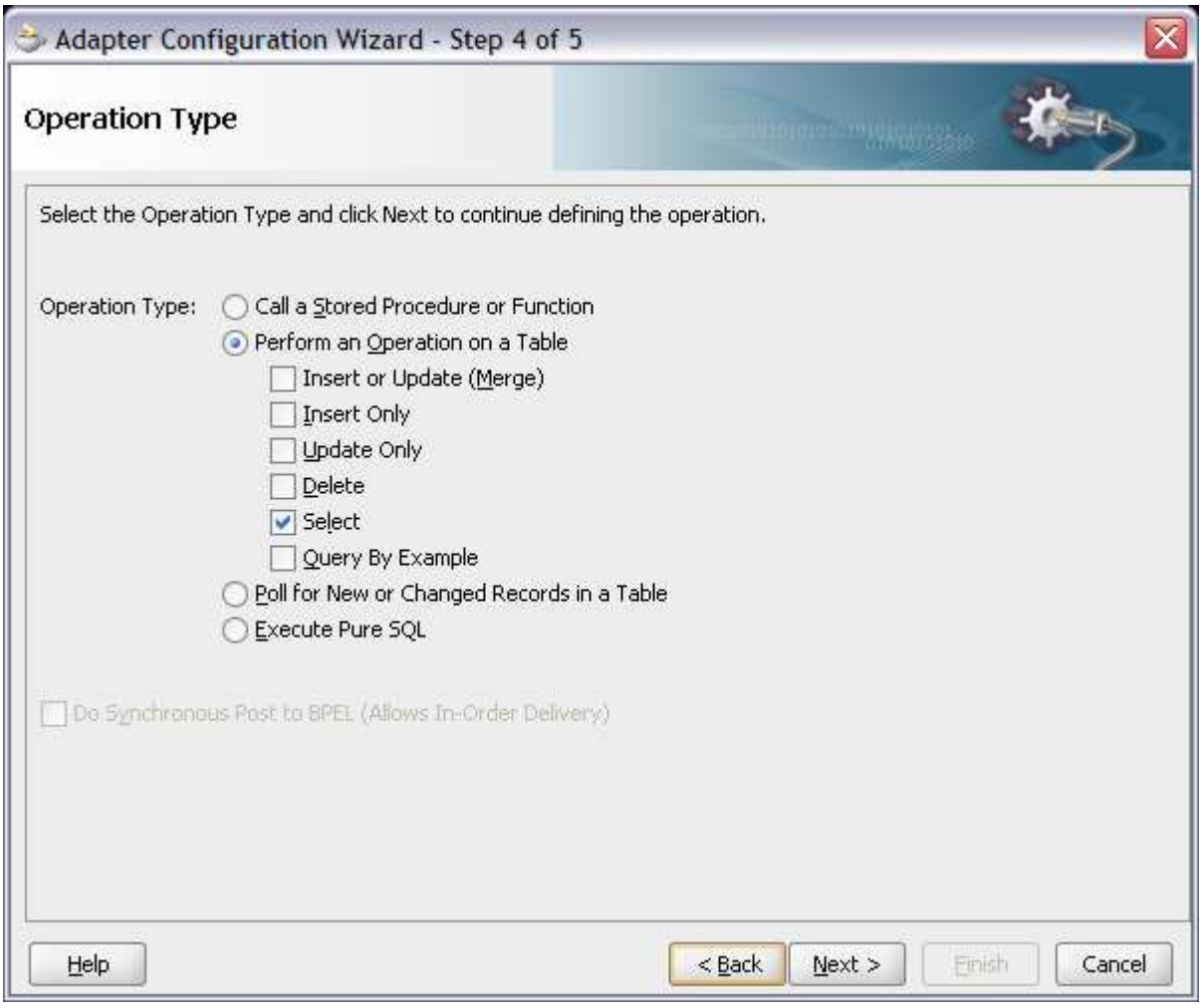
Figure 13 Adapter Configuration wizard, Step 3, after creating new connection



9. Make sure your **Connection** specifies the correct connection.
10. Click **Next**.
11. Set the following fields:
    - **Perform and Operation on a Table:** Selected
    - **Select:** Checked

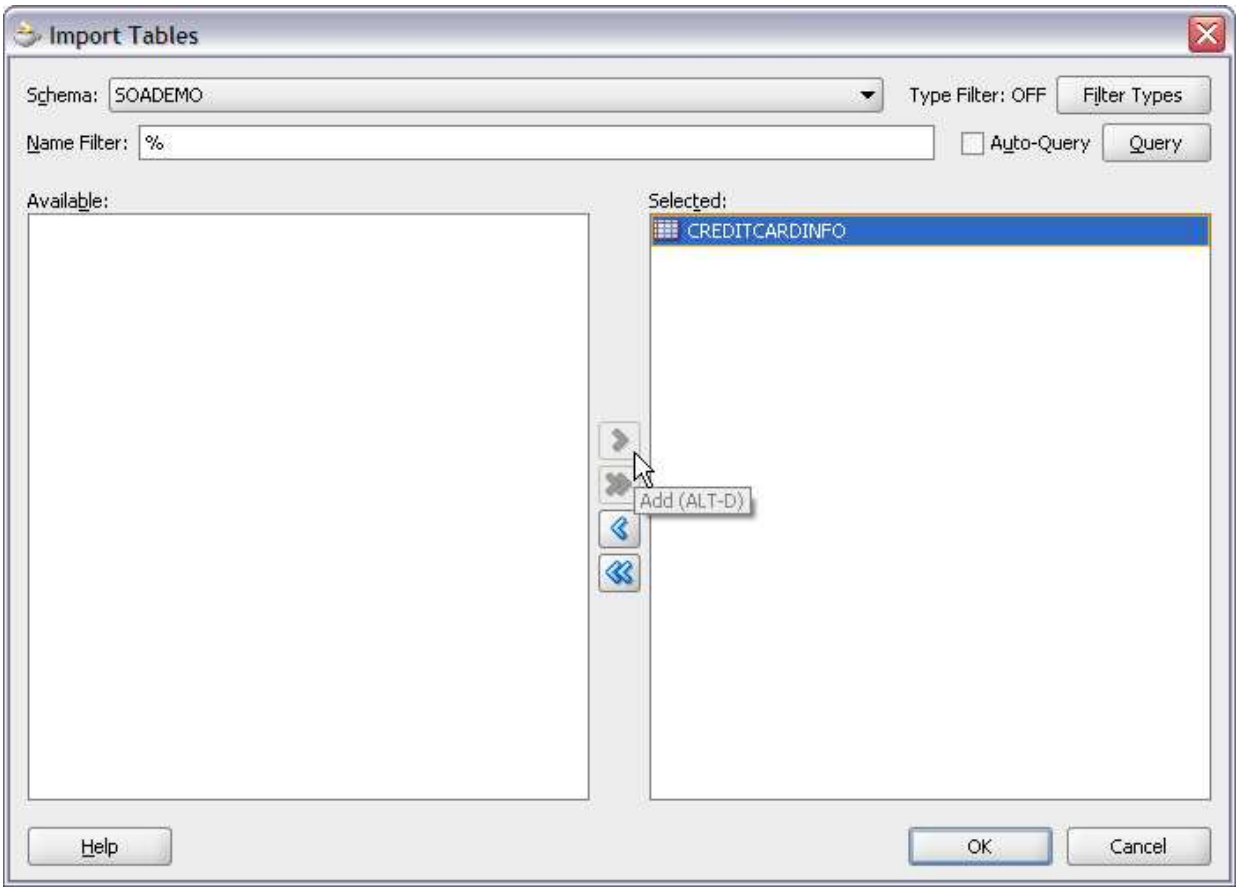
All other fields should be unchecked, see Figure 14.

Figure 14 Adapter Configuration wizard, Step 4



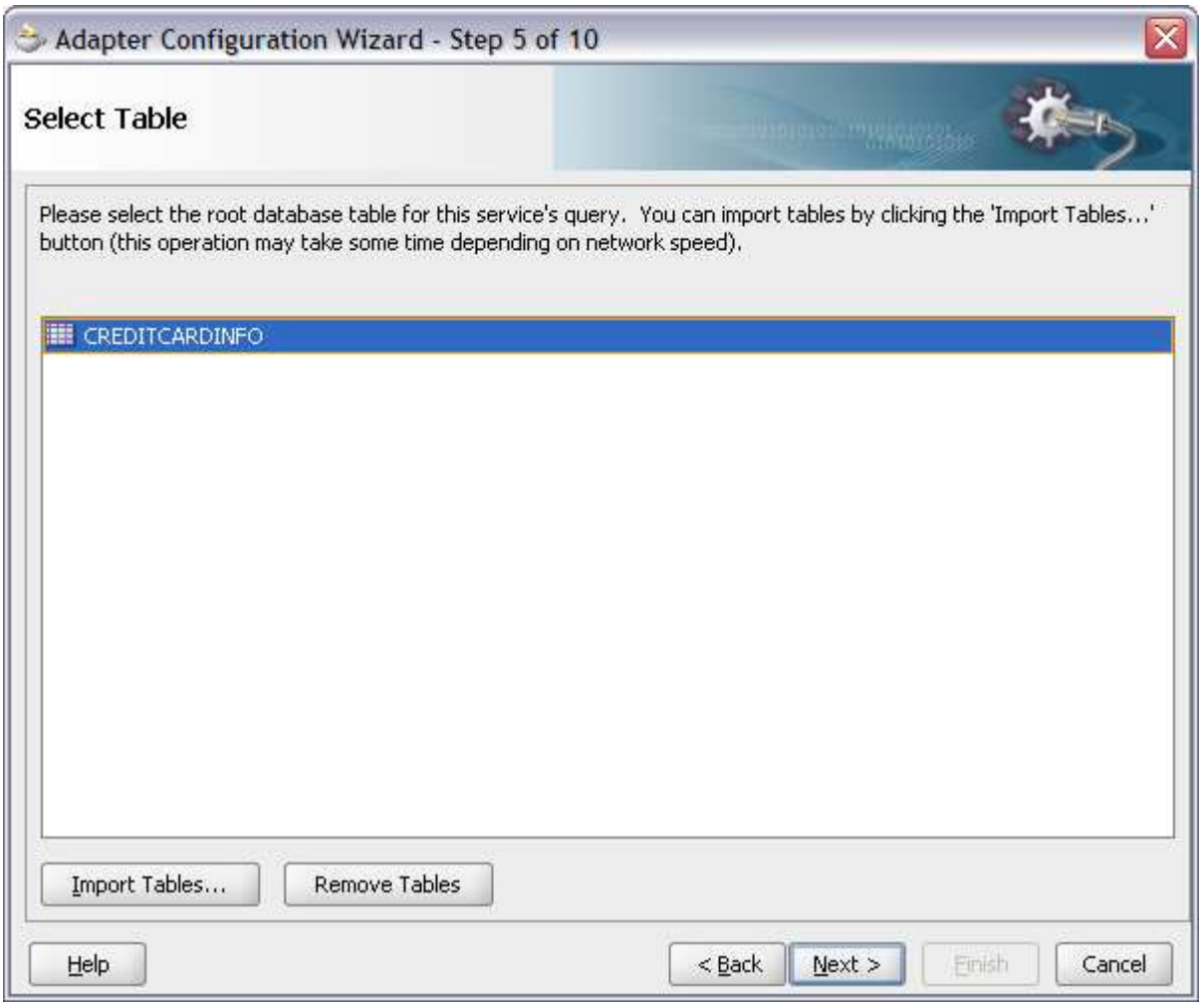
12. Click **Next**.
13. In step 5, click the **Import Tables** button to bring up the **Import Tables** dialog.
14. Click the **Query** button to retrieve the list of tables from the database.
15. Select the **CREDITCARDINFO** table and move it to the **Selected** field by pressing the **Add** or **Add All** shuttle button.

Figure 15 Adapter Configuration wizard, importing tables



16. Click **OK**.
17. Back in step 5 of the **Database Adapter Configuration** wizard, click the **CREDITCARDINFO** table to select it.

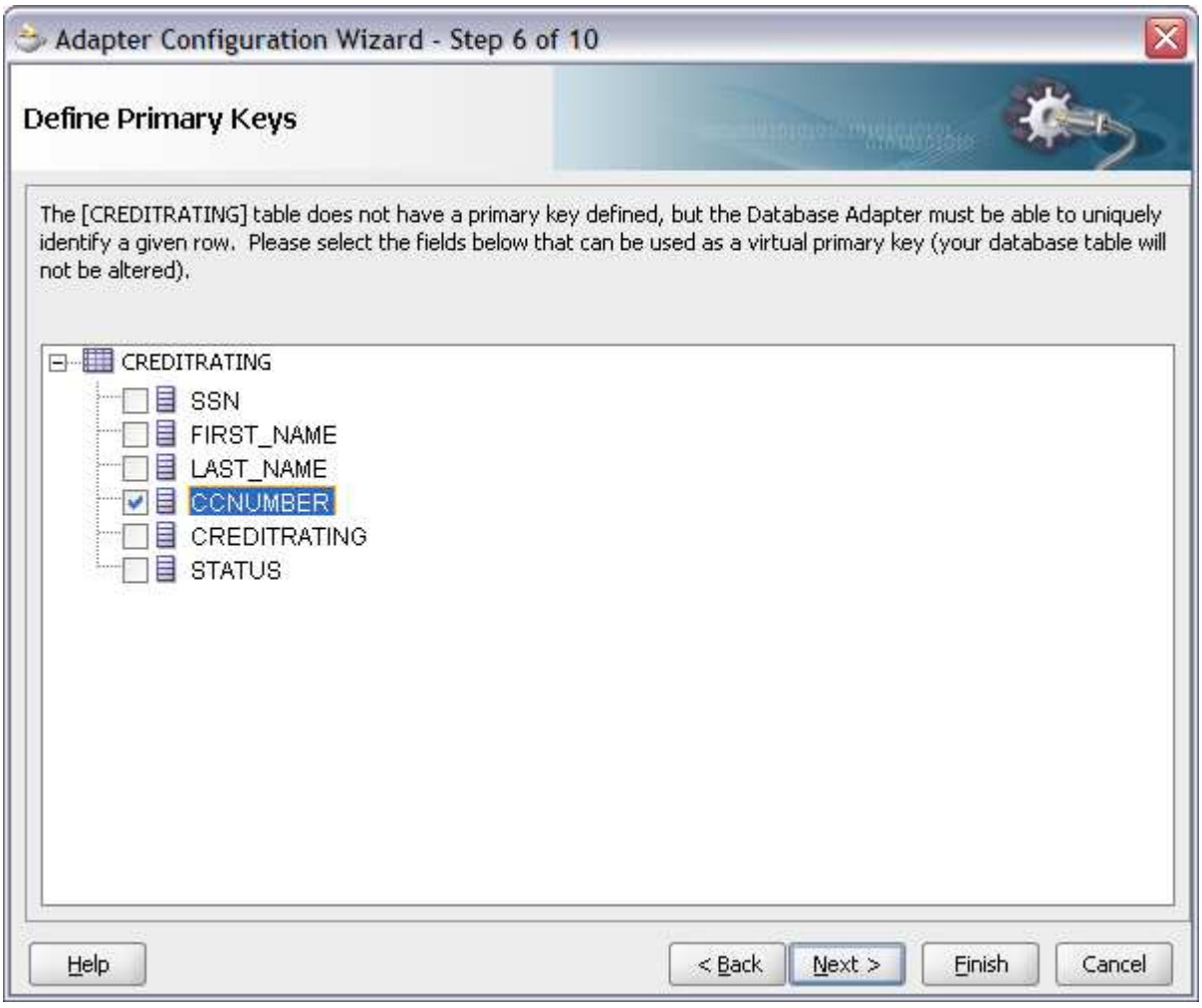
Figure 16 Database Adapter Configuration wizard, step 5



18. Click **Next**.

19. Step 6 of the wizard lets you override or define the primary key for your table. In this case no primary key is defined in the database, so you'll need to specify it. Check **CCNUMBER** and leave the rest unchecked.

Figure 17 Defining the primary key



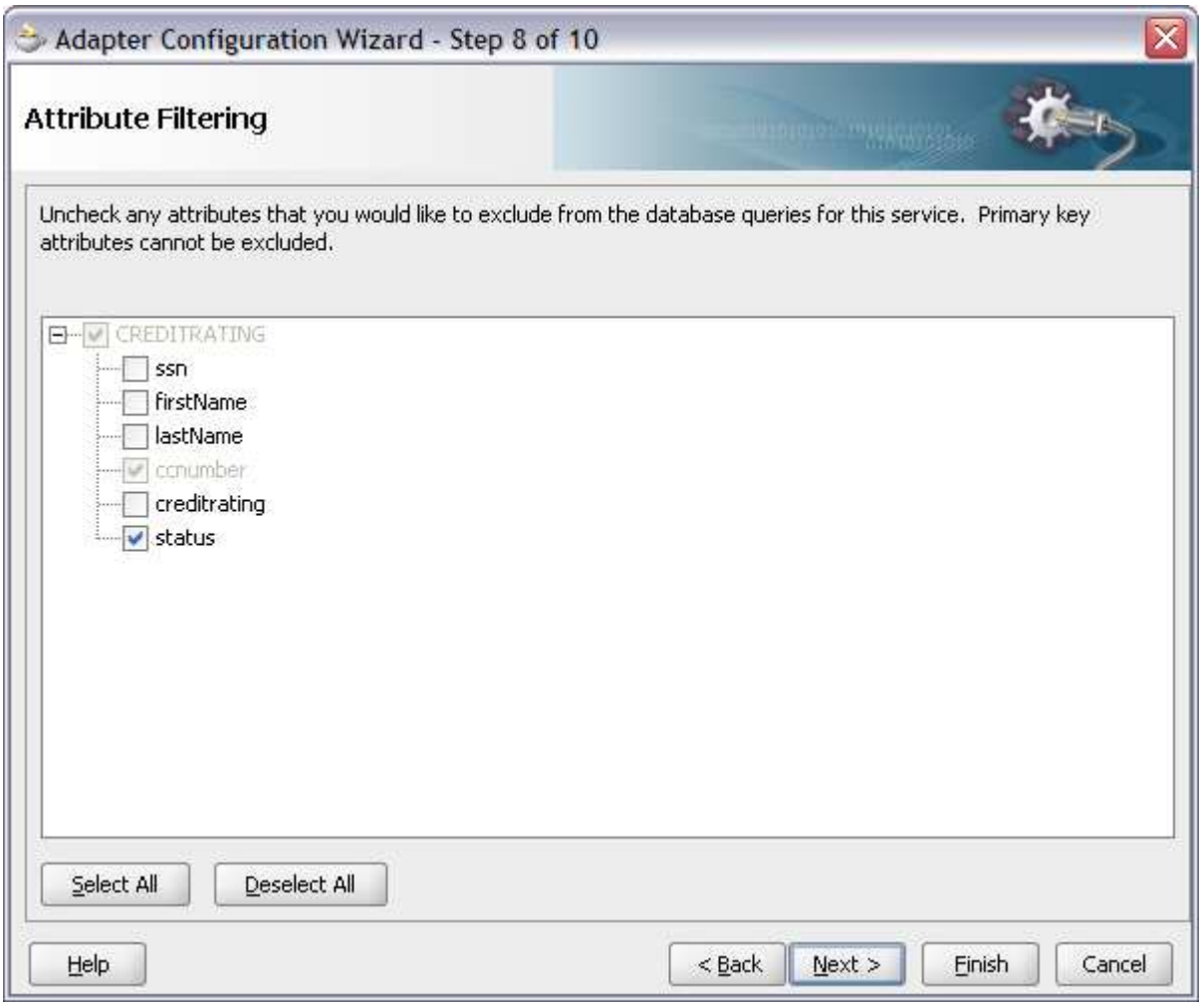
20. Click **Next**.

21. Step 7 lets you define relationships if you are selecting from multiple tables. Since there is only a single table there is no relationship to define.

    Click **Next**.

22. Uncheck all fields, in step 8, except **status**. That is the only column we want to query from the database.

Figure 18 Database Adapter Configuration wizard, step 8



23. Click **Next**.

24. Step 9 is where you can specify your selection criteria. In this case you will add a parameter to the query. The parameter will be set at runtime to select the row for the credit card you want.

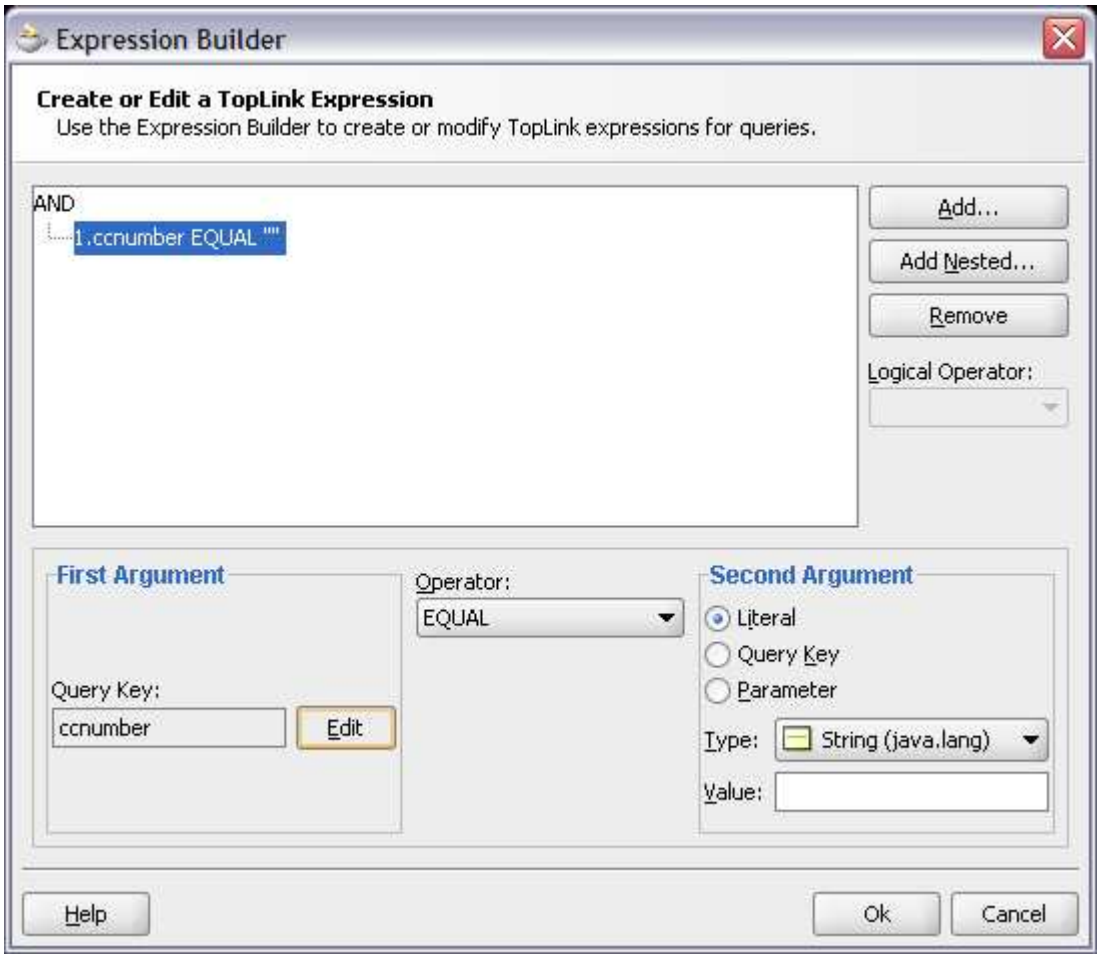
    Click the Add button to add a new parameter called **ccnb**. Click **OK**.

Figure 19 Database Adapter Configuration wizard, adding a parameter

25. Back in step 9, click the **Edit** button to bring up the **Expression Builder** dialog.

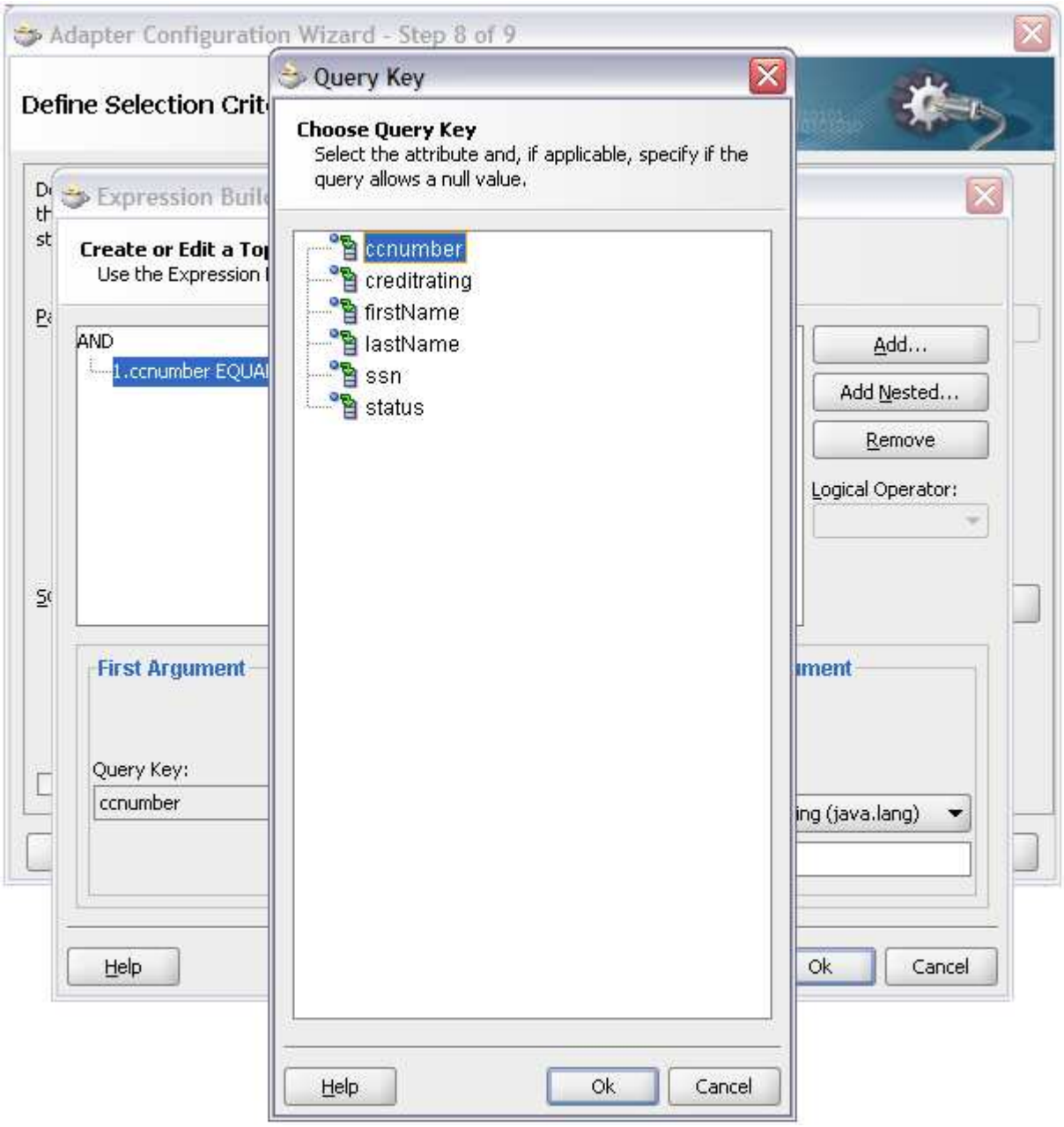
26. Press the **Add** button to add a new condition

Figure 20 Adding a new condition



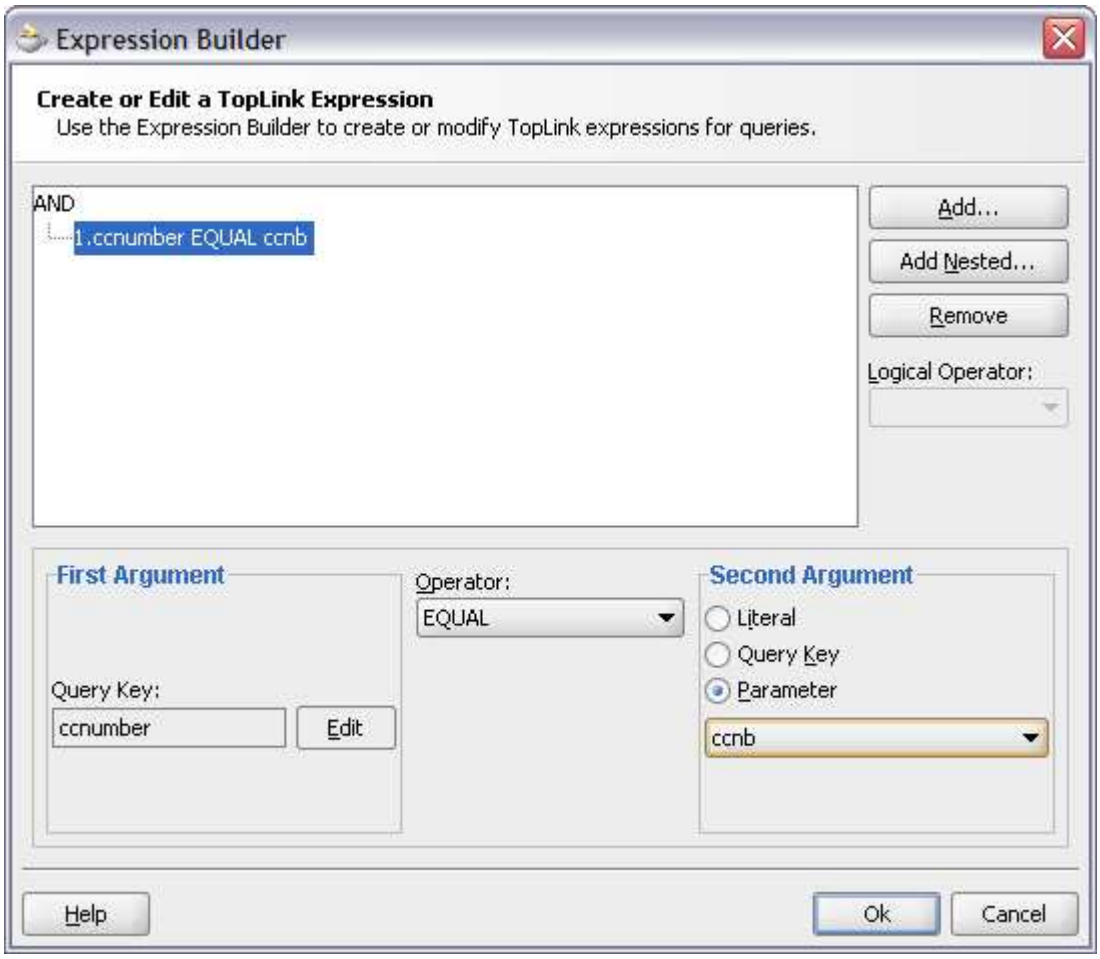
27. Press the **Edit** button in the **First Argument** section.
28. In the **Query Key** dialog select **ccnumber**.

Figure 21 Selecting the column for the condition



29. Click **OK**.
30. In the **Second Argument** section, select **Parameter**.
31. Ensure **ccnb** is selected in the poplist.

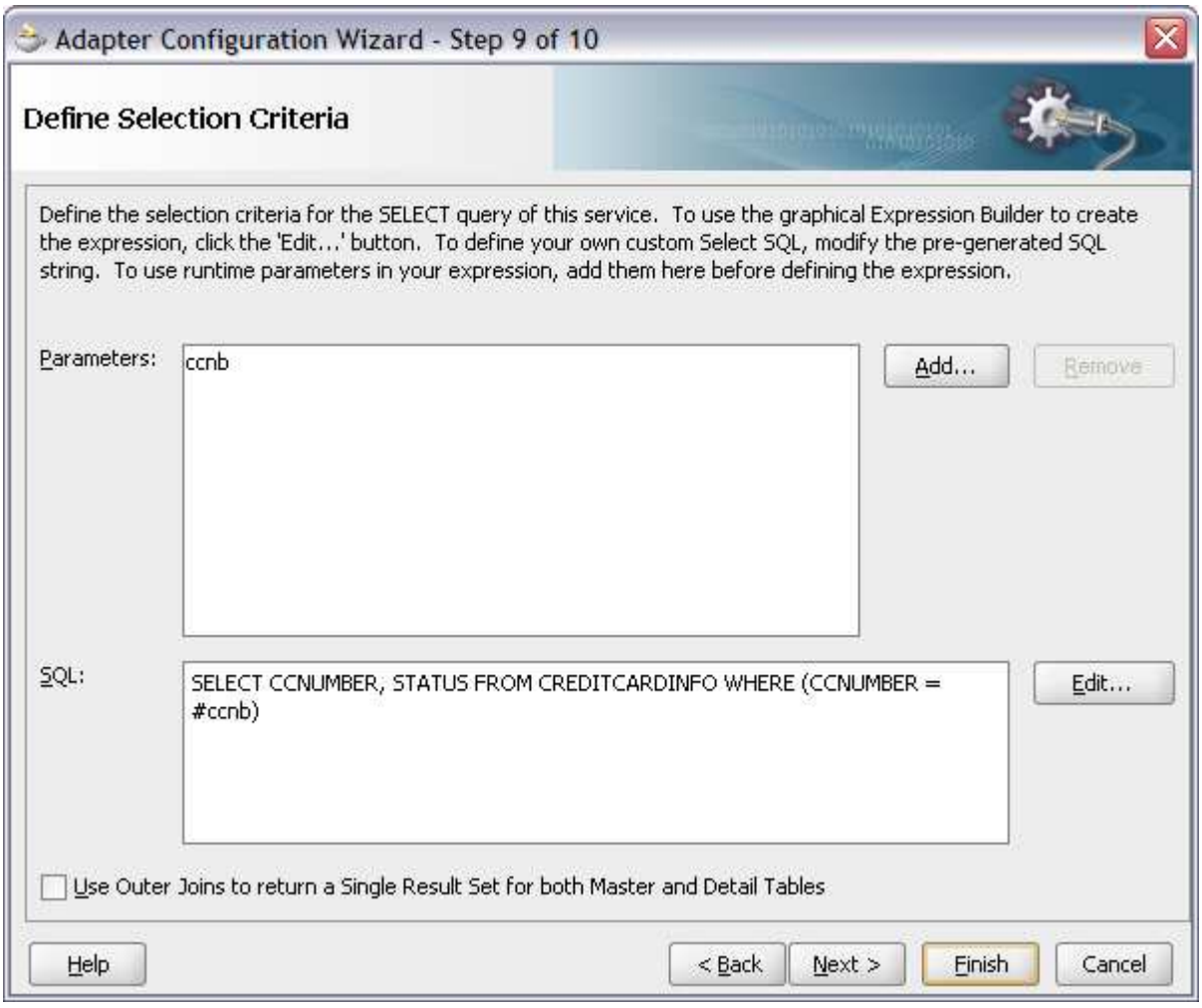
Figure 22 Specifying the parameter



32. Click **OK**.

33. Back in step 9 you see the summary of the query that you specified, as shown in Figure 23. The parameter **ccnb** will be populated at runtime and the query will select a row based on that parameter.

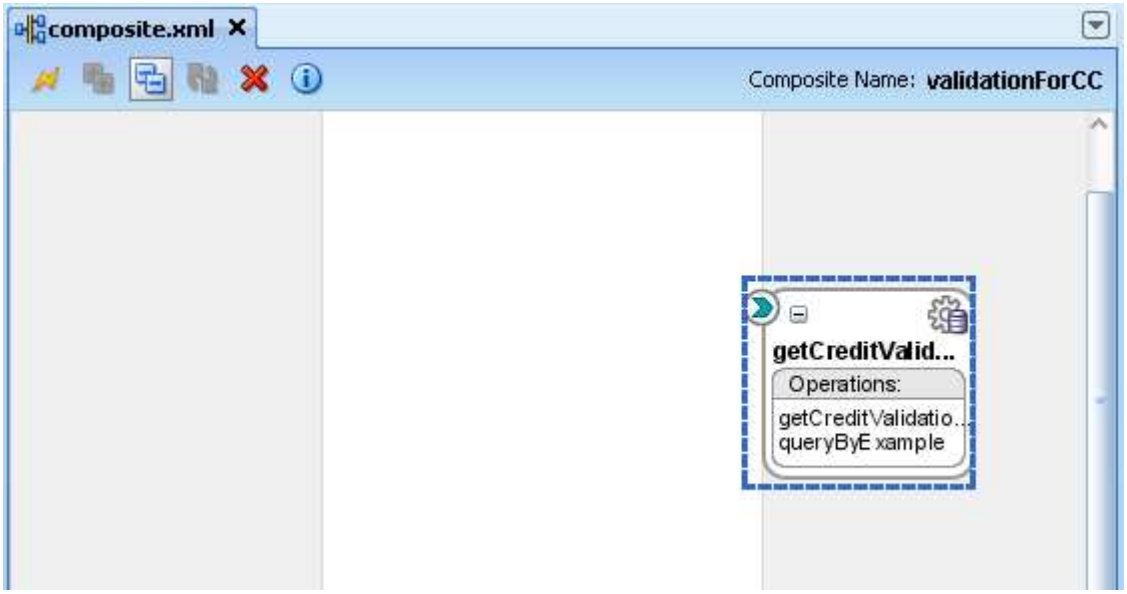
Figure 23 Summary of selection criteria



34. Click **Finish**.

The database adapter will process your choices and generate a service that implements the operation you specified, which means your project will now contain a new WSDL file to represent that service: `getCreditValidation.wsdl`. The composite diagram has been updated with this service as shown in Figure 24.

Figure 24 Composite updated with database adapter service

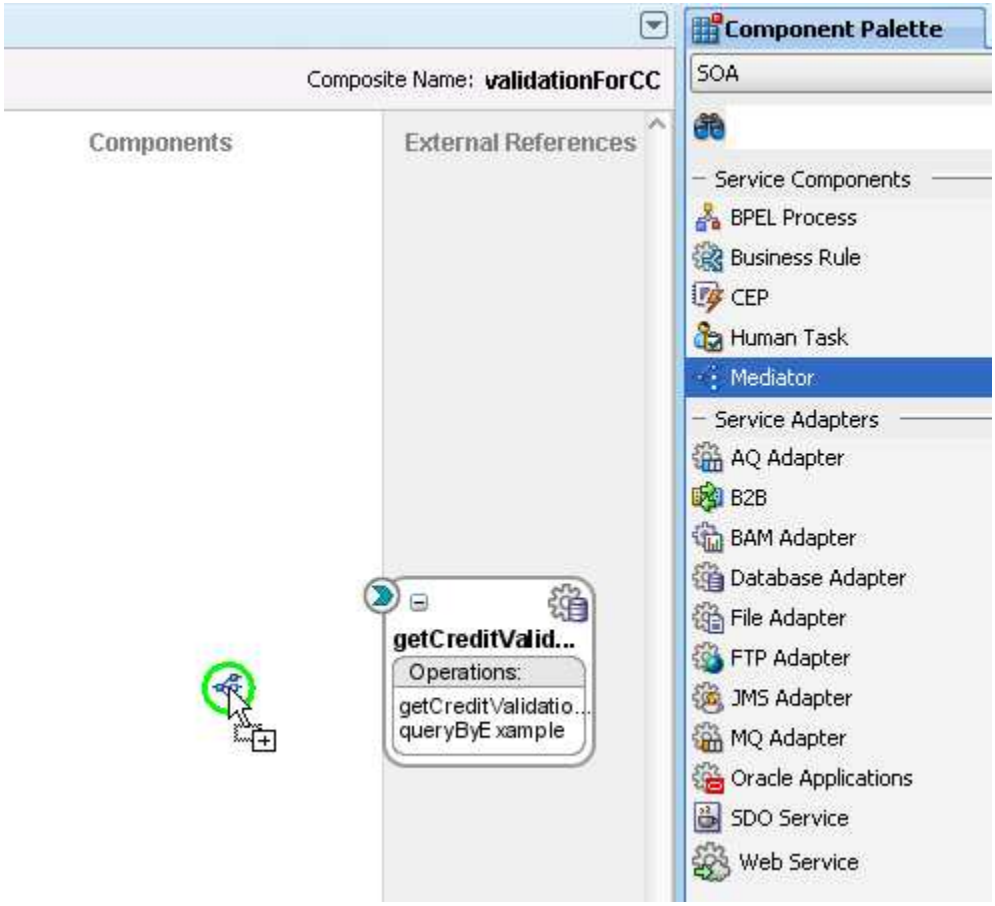


2.2.3 Adding the Mediator Component

The Mediator is the component in charge of routing in the SOA Suite. You will add a Mediator component to route request to the service you just created with the database adapter.

1. Drag a Mediator component onto the composite diagram in the **Components** swim lane.

Figure 25 Adding a Mediator component

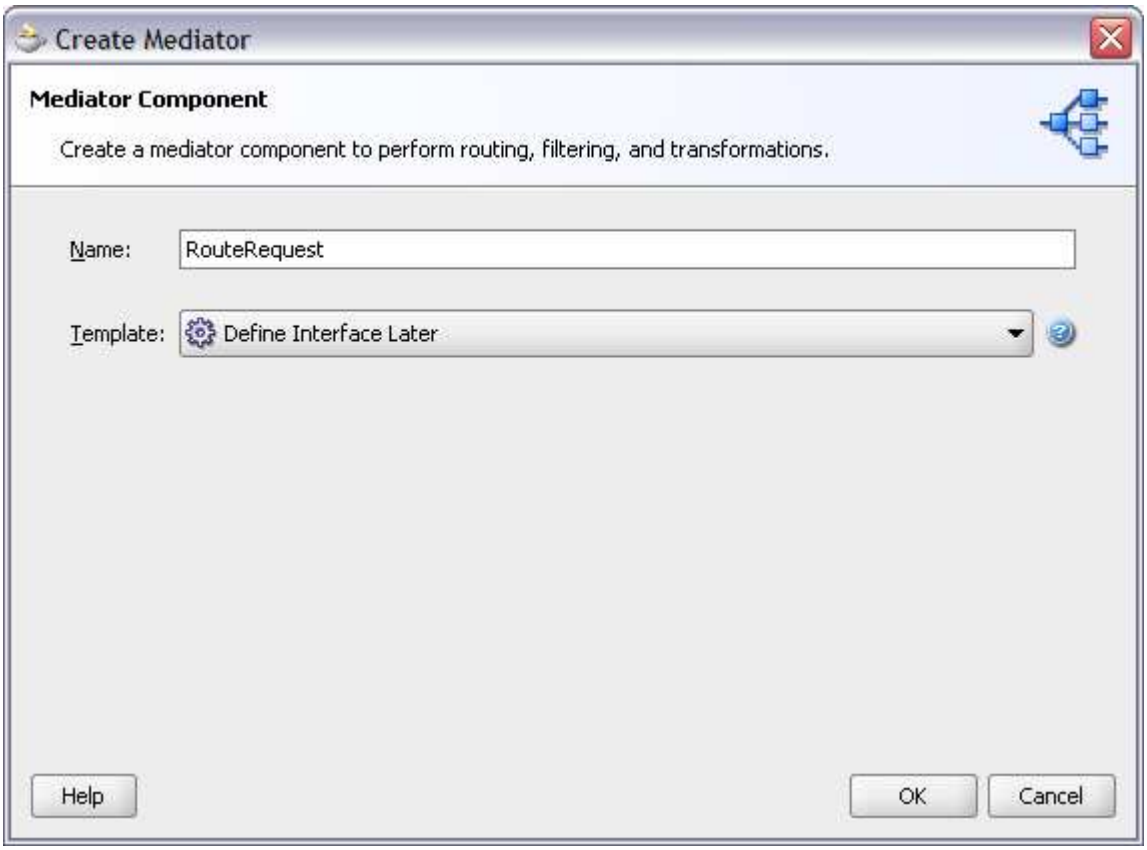


2. In the **Create Mediator** dialog, specify these settings:

   - **Name:** RouteRequest
   - **Template:** Define Interface Later

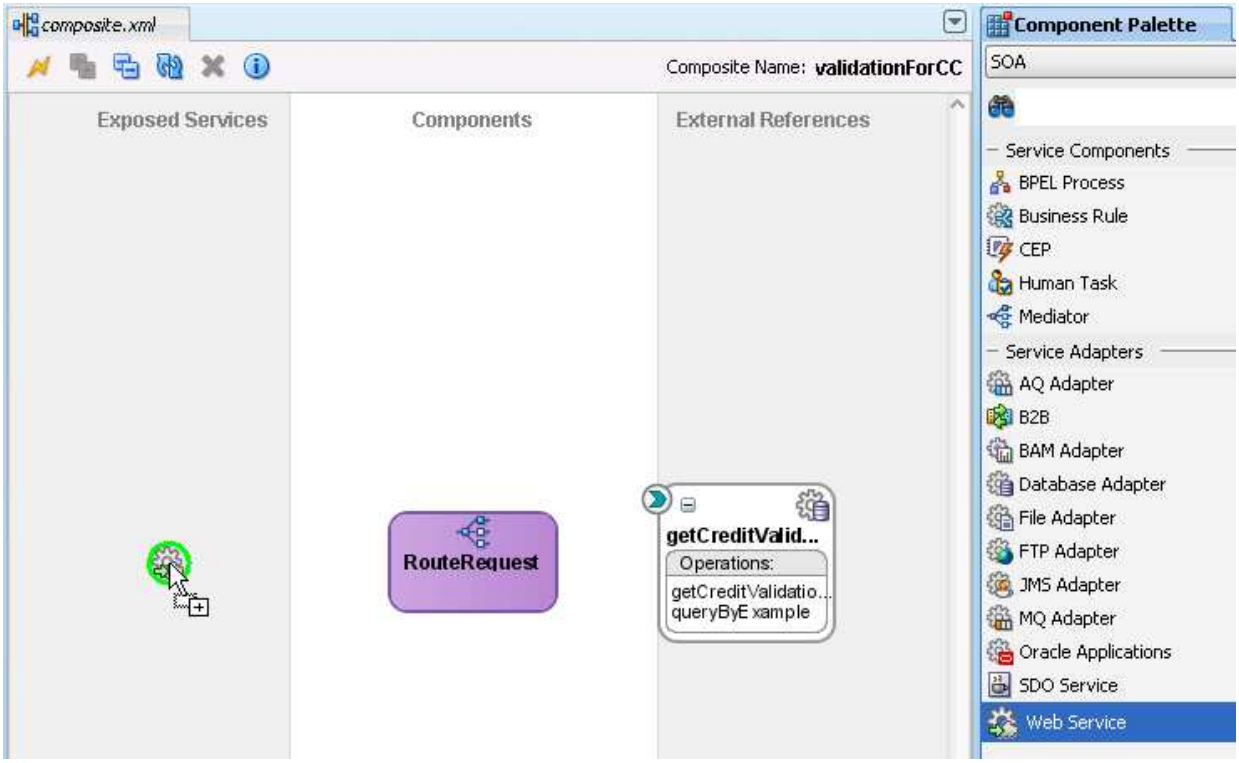
   A typical composite application will have many components, but only some of them will have a publically exposed web services interface. The composite diagrammer in JDeveloper gives you the flexibility to define the interface now, choose an existing interface, or to define the interface later.

Figure 26 New Mediator component settings



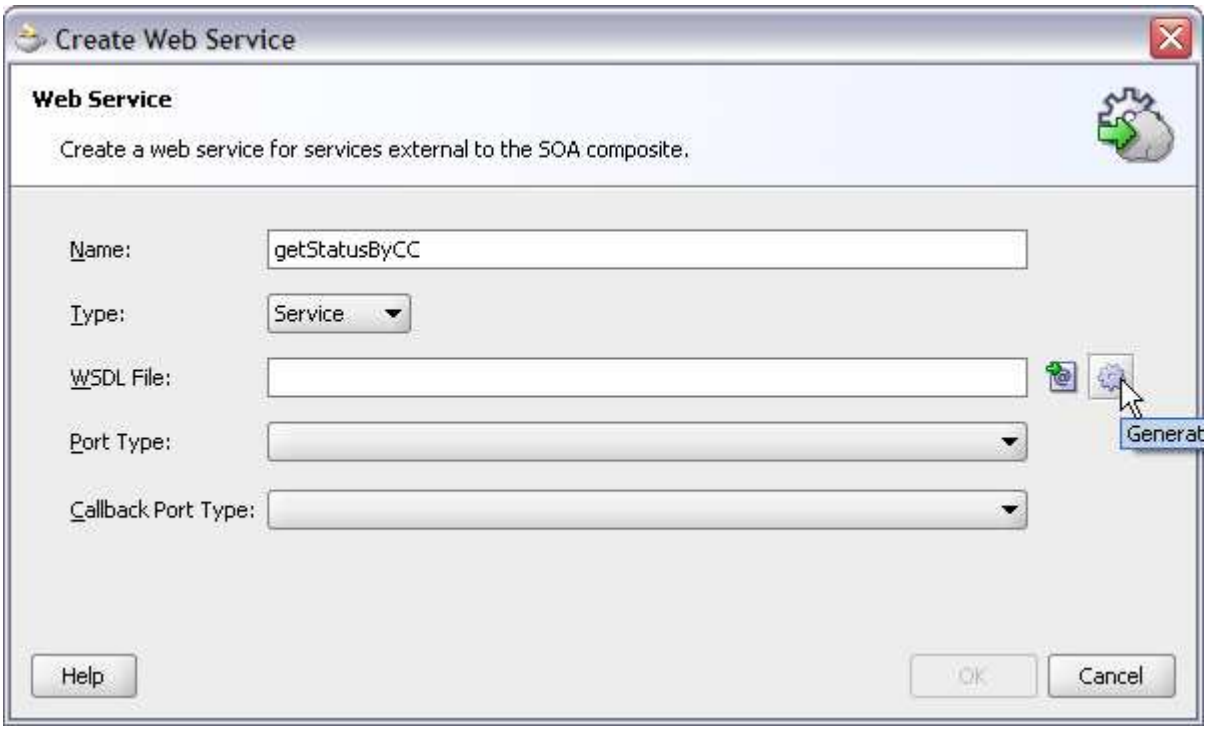
3. Click **OK**.
4. We will create a Web Service interface to expose this service using SOAP bindings. Drag a Web Service adapter to the **Exposed Services** swim lane.

Figure 27 Adding a Web Services interface



5. In the **Create Web Service** dialog that appears, set the following fields:
   - **Name:** getStatusByCC
   - **Type:** Service
6. Click the cog icon next to the **WSDL File** field to define the interface.

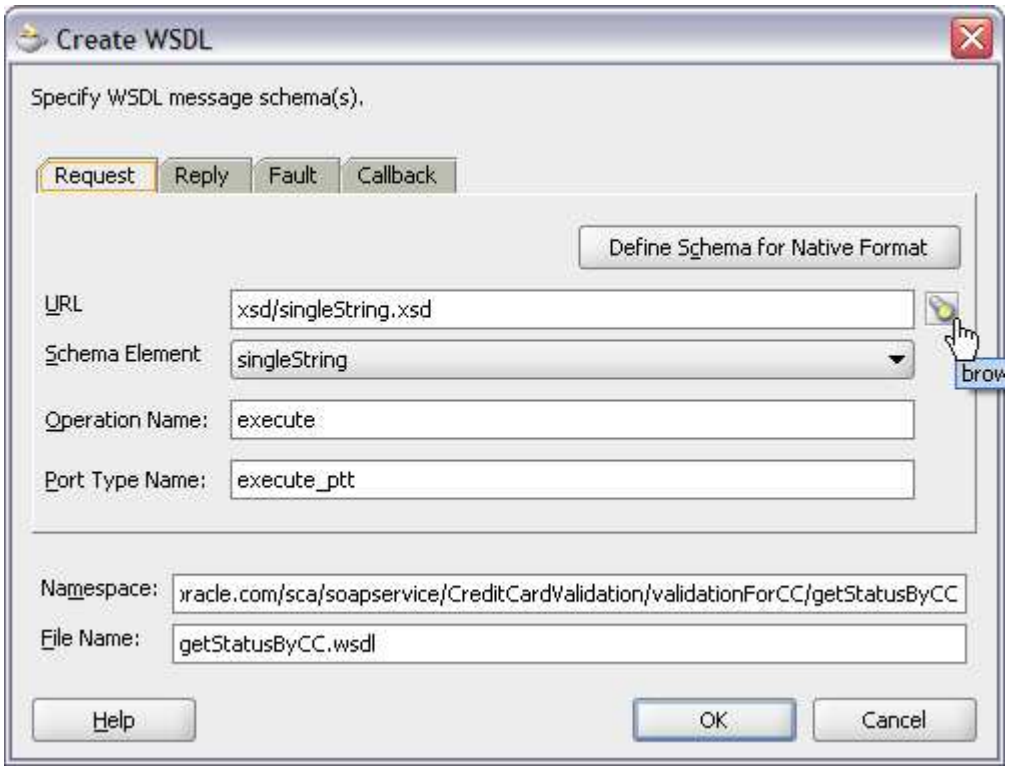
Figure 28 Creating a new service interface



7. The **Create WSDL** dialog lets you specify the message invocation types for the service. You have been given an existing XML schema definition (XSD) that specifies the types of the input and output for the service which you will reuse for this application.

   Click the flashlight icon to the right of the **URL** field to browser for a schema file.

Figure 29 Defining the message invocation types

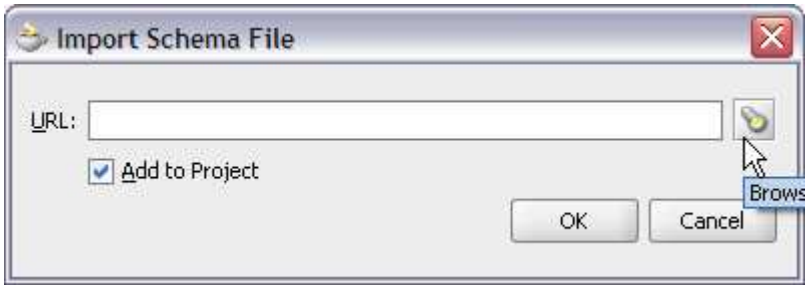


8. In the **Type Chooser** dialog, click the **Import Schema File** button.

Figure 30 Importing a schema file

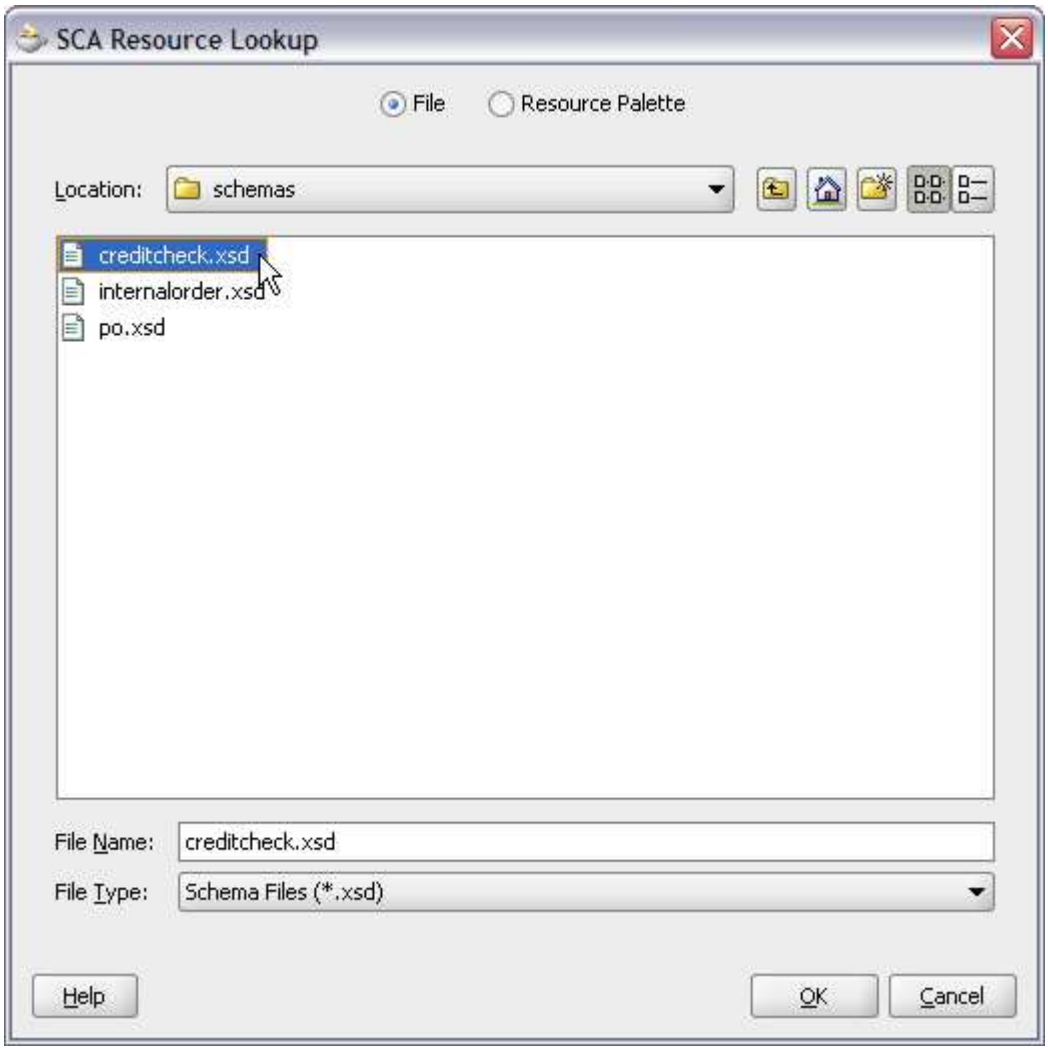
9. In the **Import Schema File** dialog, click the flashlight to browse for the schema file.

Figure 31 Importing a schema file



10. In the **SCA Resource Lookup**, make sure **File** is selected.
11. Navigate to and select `c:\po\schemas\creditcheck.xsd`.

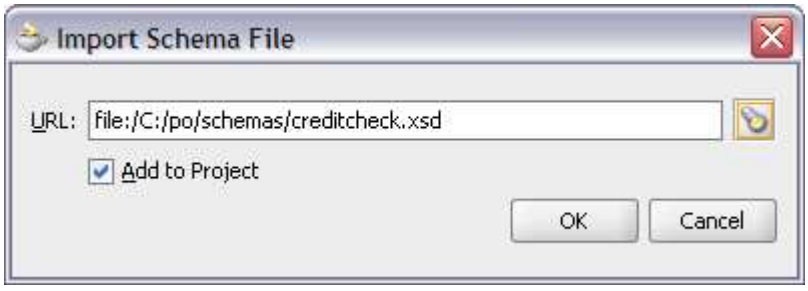
Figure 32 Selecting the schema file



12. Click **Ok**.

13. Back in the **Import Schema File** dialog, make sure **Add to Project** is selected.

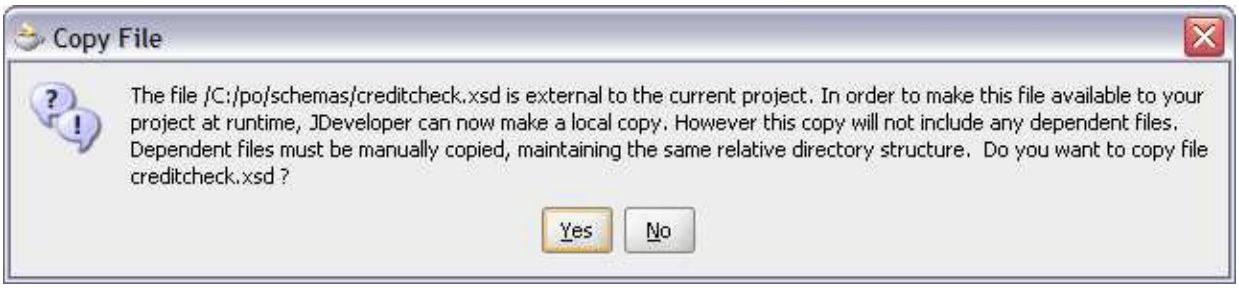
Figure 33 Import Schema File dialog



14. Click **OK**.

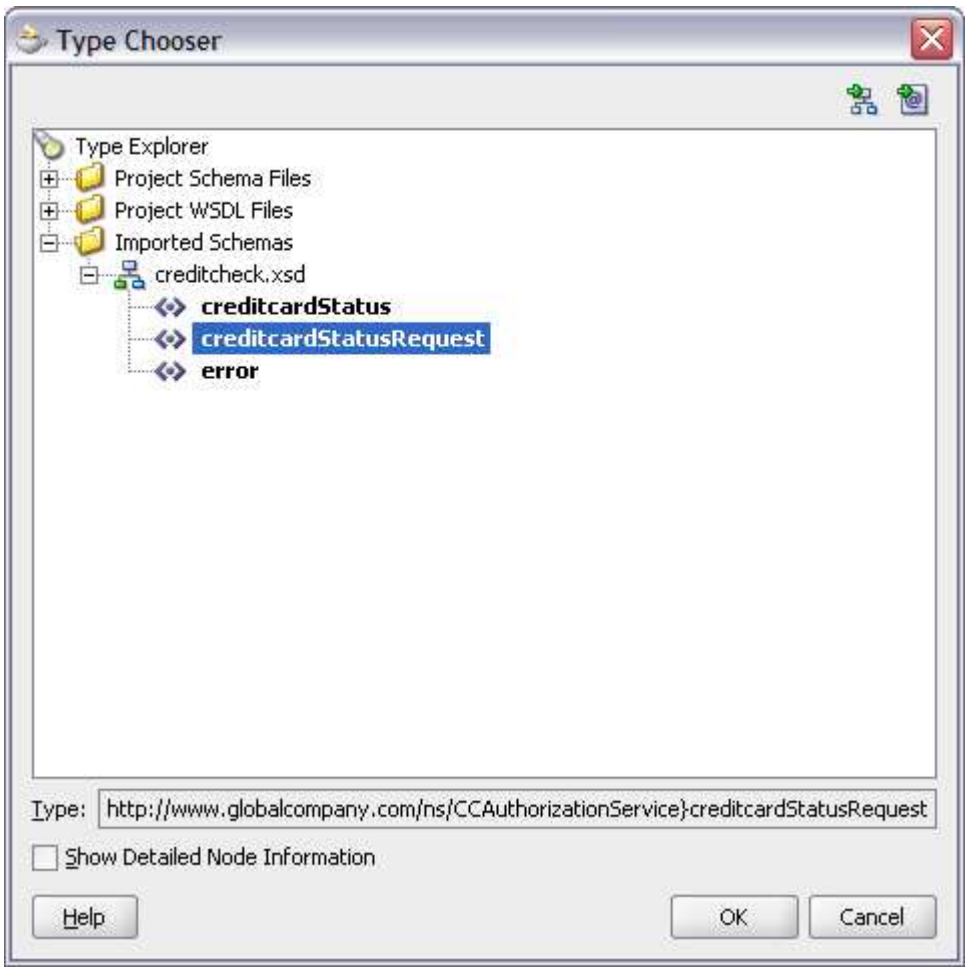
15. You may get a **Copy File** dialog asking if you want to copy the file to your project. If so, say **Yes**.

Figure 34 Copy file confirmation



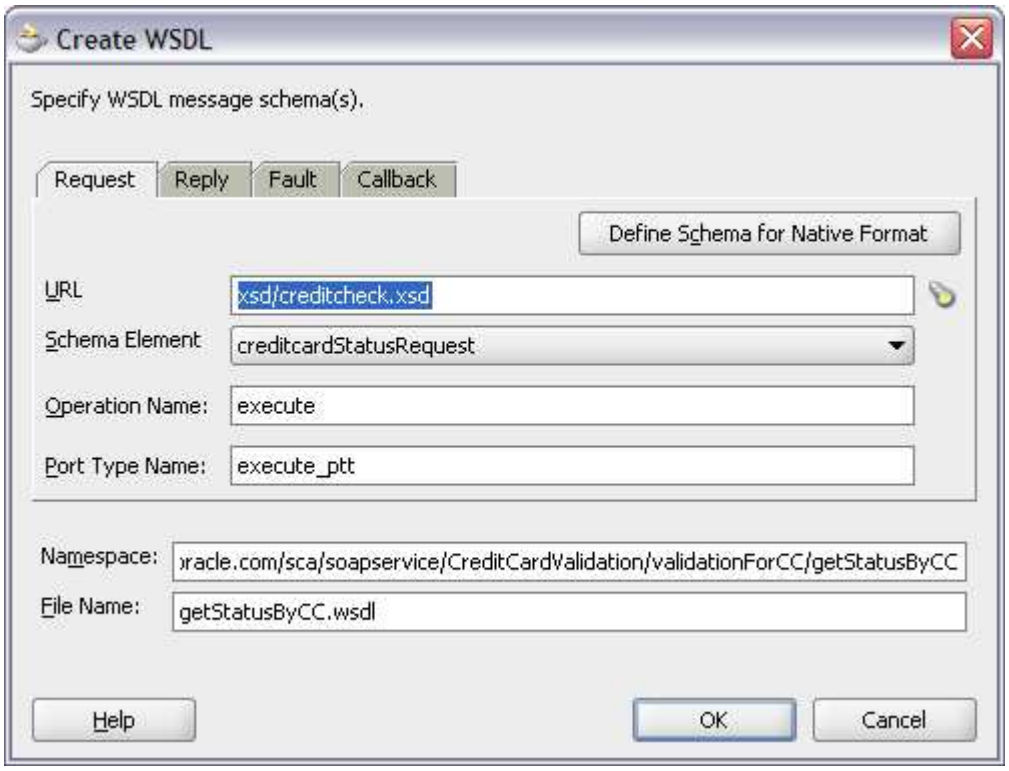
16. Back in the **Type Chooser** dialog, expand the **Imported Schemas** > **creditcheck.xsd** nodes.
17. Select **creditcardStatusRequest**.

Figure 35 Selecting the schema type



18. Click **OK**.
19. The **Create WSDL** dialog should now look like Figure 36.

Figure 36 Setting the Request invocation type



20. In a similar way, you will set the message types for the reply and fault to **creditcardStatus** and **error**. The only difference is that you don't need to import the schema definition again because it's now part of your project.

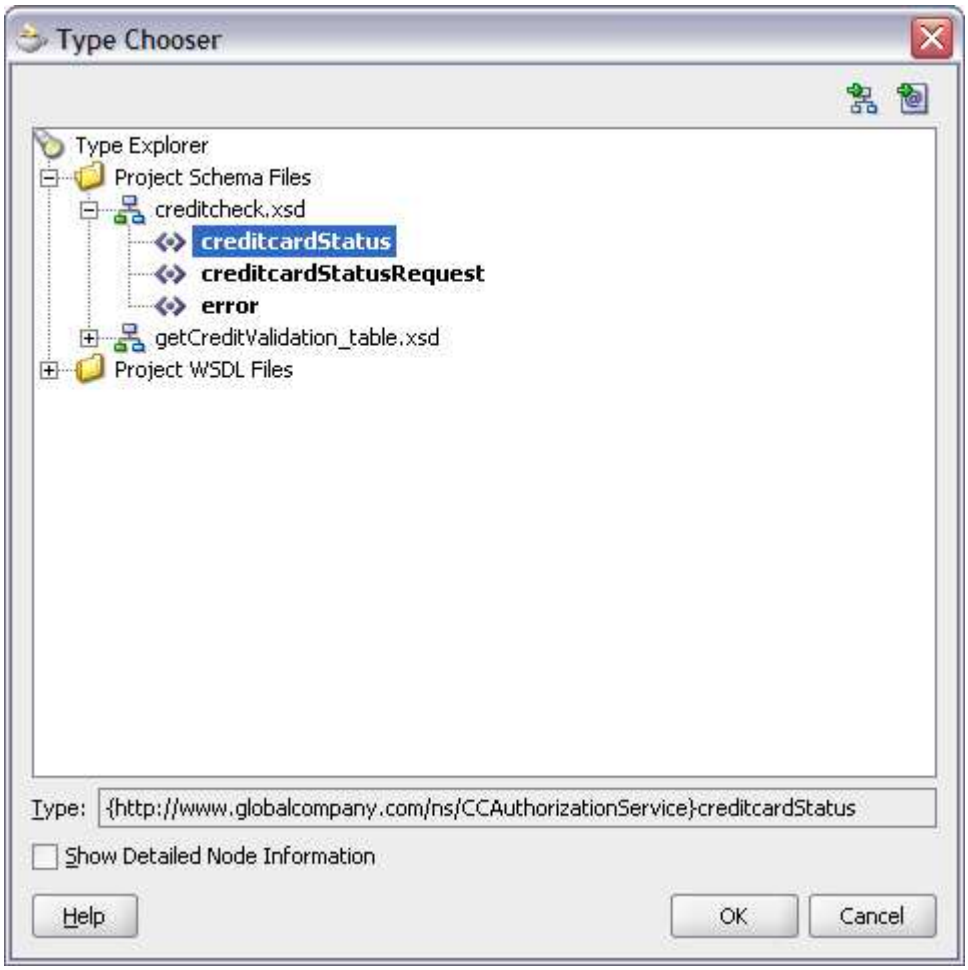
    Click the **Reply** tab.

21. Click the flashlight icon to the right of the **URL** field.

22. Expand the **Project Schema Files > creditcheck.xsd** nodes.

23. Select **creditcardStatus**.

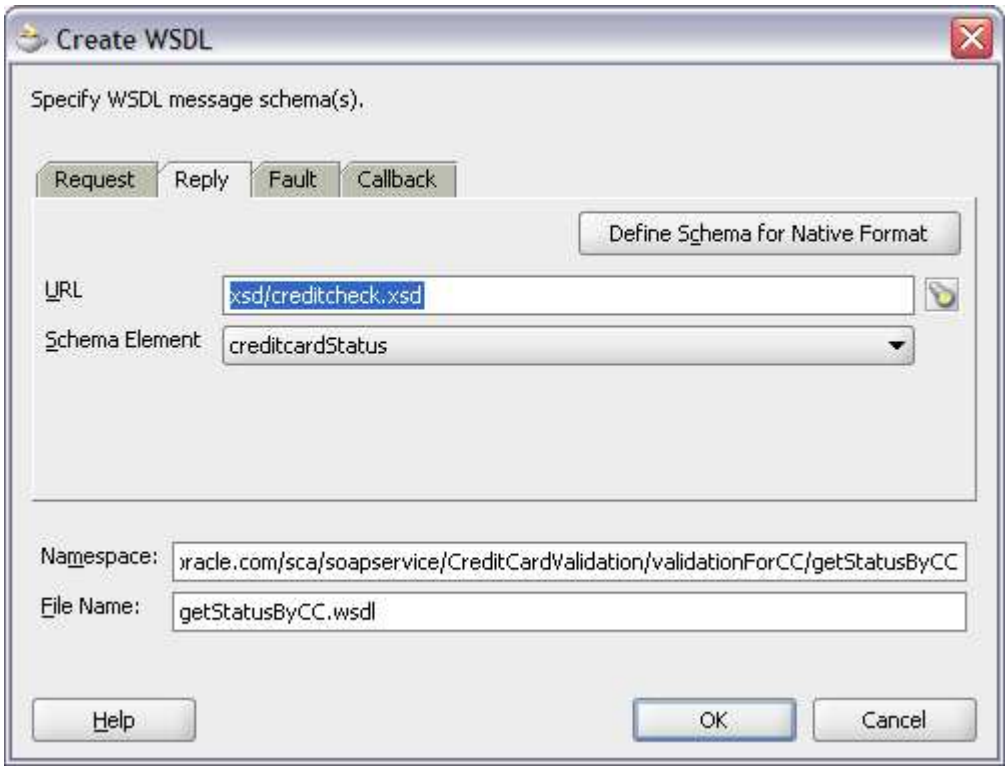
Figure 37 Choosing the reply schema type



24. Click **OK**.

25. The reply message type should look like Figure 38.

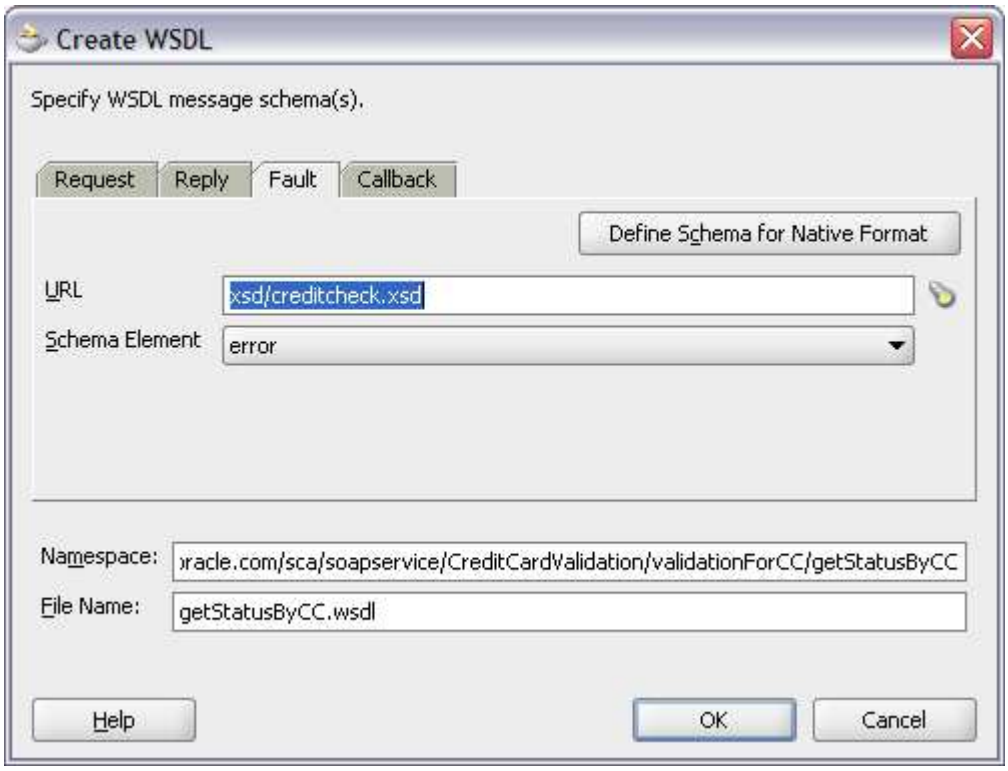
Figure 38 The reply invocation type



26. Click the **Fault** tab.

27. Using the same method, select the **error** type. The **Fault** tab should look like Figure 39.

Figure 39 The fault invocation type

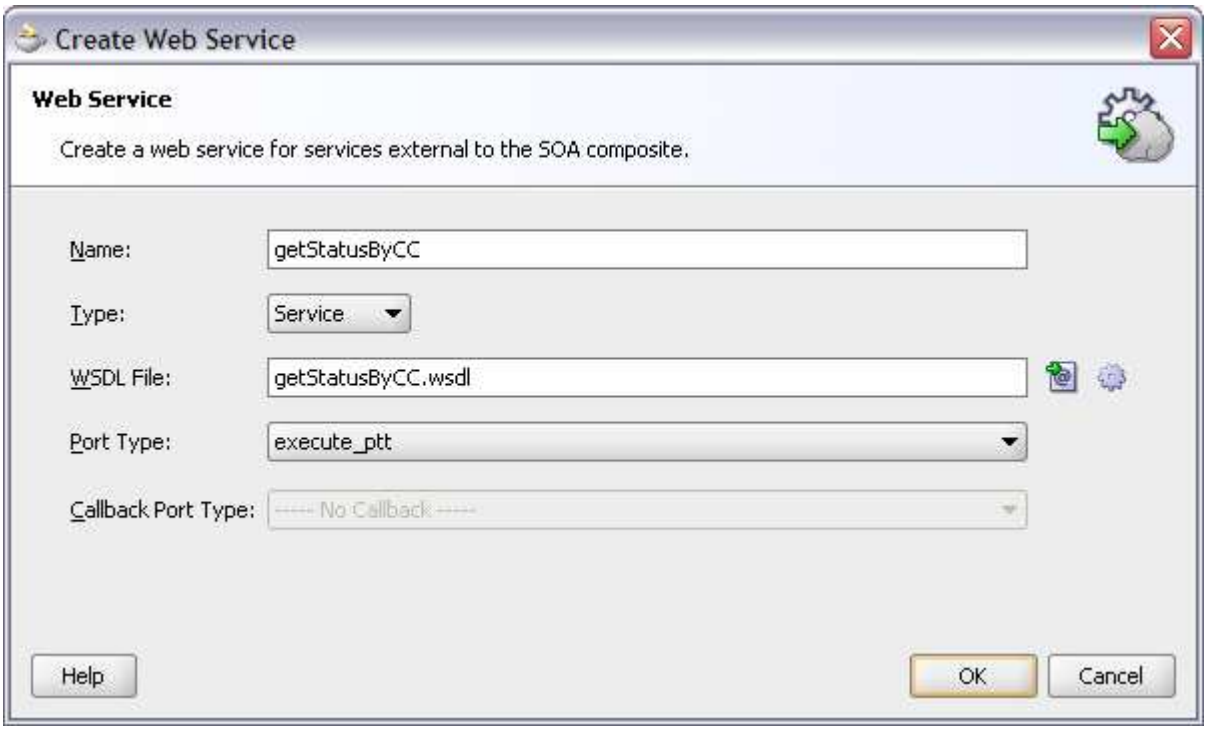


28. Click **OK** to close the **Create WSDL** dialog.

29. The **Create Web Service** dialog should now look Figure 40.

Be careful here -- If you click the cog icon again you will have to set all three invocation types again (request, reply, and fault).

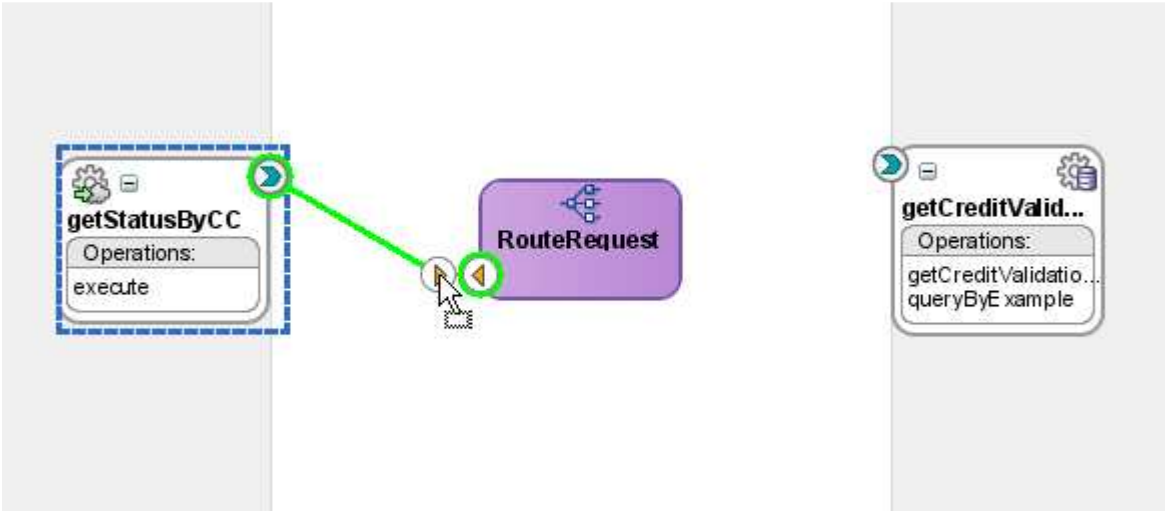
Figure 40 The web service interface



30. Click **OK**.

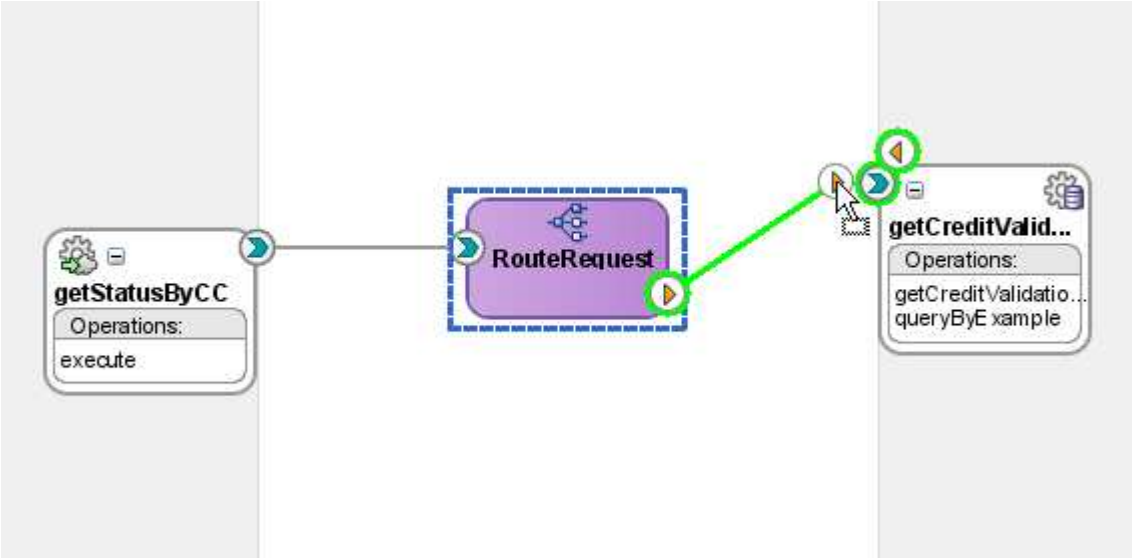
31. Now the components can be connected -- or "wired" -- together. Wire the inbound web service binding to the Mediator component:

Figure 41 Wiring the web service interface to the Mediator component



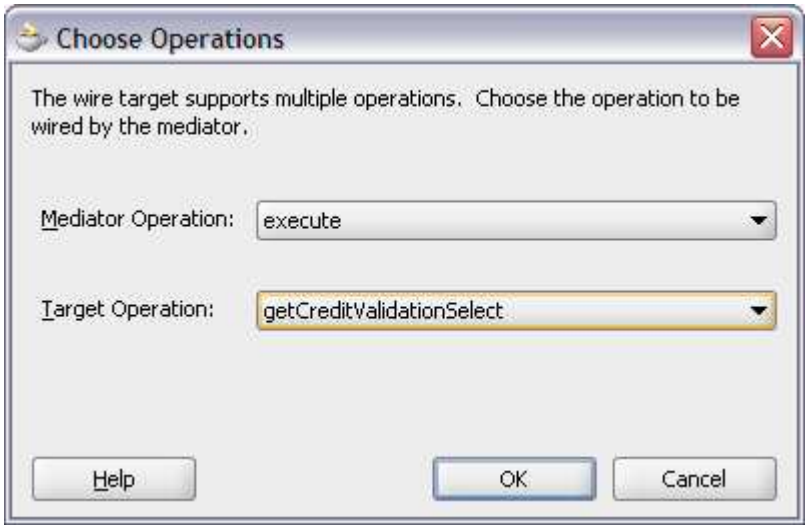
32. Wire the Mediator component to the database adapter service:

Figure 42 Wiring the Mediator component to the database adapter service



33. You may be prompted to chose the operations. If so, use these settings:
    - **Mediator Operation:** execute
    - **Target Operation:** getCreditValidationSelect

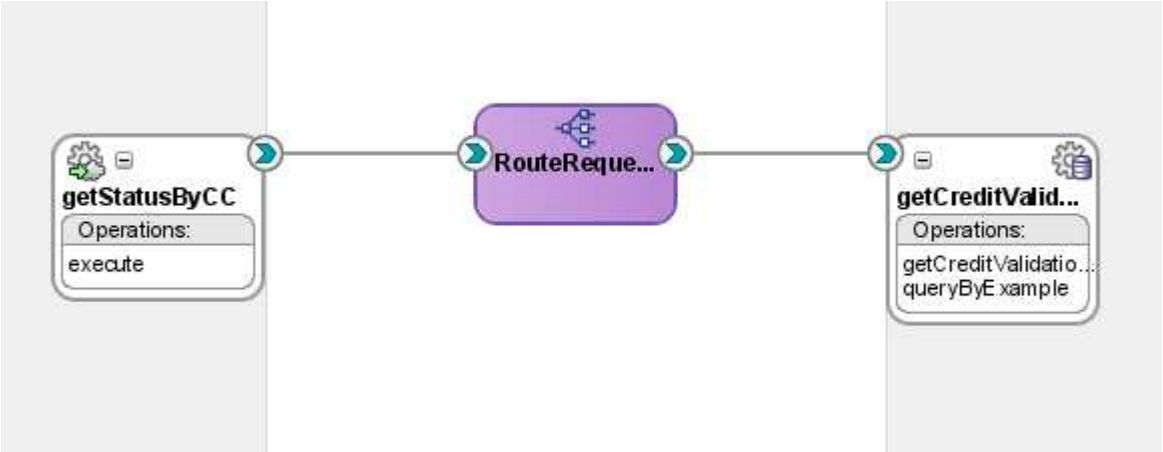
Figure 43 Choosing the operations



34. Click **OK**.

35. The composite diagram now looks like Figure 44 and gives an overview of your application.

Figure 44 The composite view



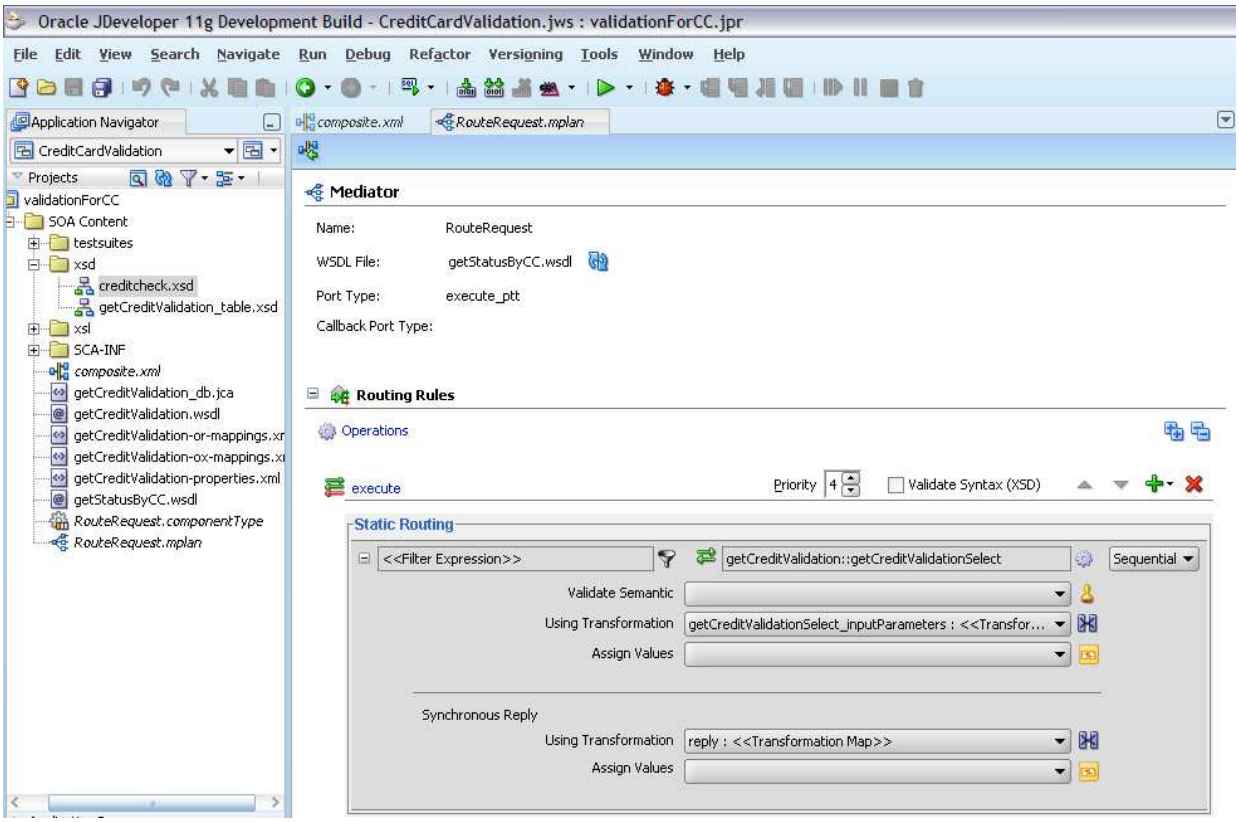
2.2.4 Adding a transformation to the Mediator component

In this section you will modify the **RouteRequest** Mediator component to do a an XSLT transformation on the message payload. That's because the incoming message to our publicly exposed service (**getStatusByCC**) is in a different format than the service created by the database adapter (**getCreditValidation**).

Not only does the Mediator route requests between endpoints (which you did by wiring them together) but it can also transform the data as it passes through.

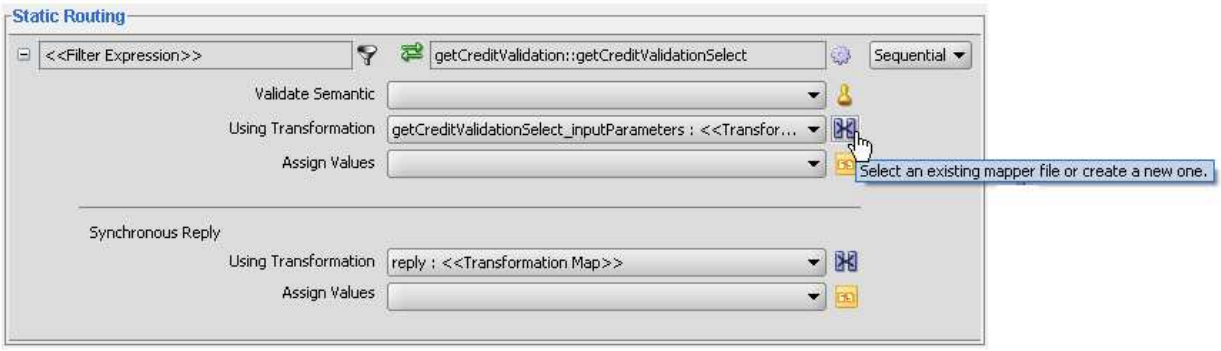
1. One way to drill down into a specific component is to double-click it from the composite diagram. Double-click the **RouteRequest** Mediator component to open the Mediator editor.

Figure 45 The Mediator editor



2. Click the transformation icon to the right of the **Using Transformation** field in the request section (that's the first **Using Transformation** field).

Figure 46 Adding a transformation for the request



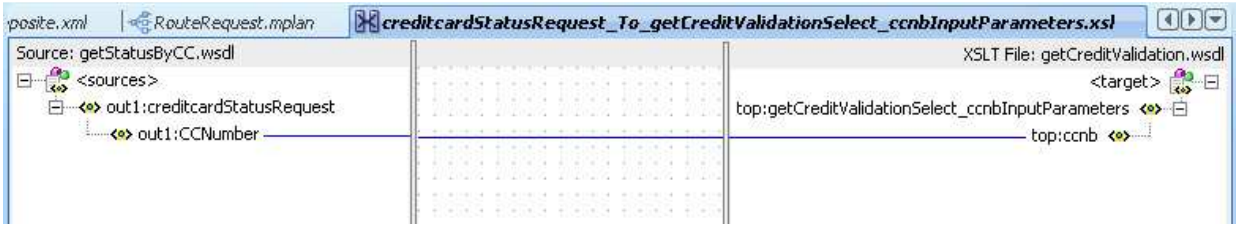
3. Select **Create New Mapper File** and accept the default name.

Figure 47 Creating a new mapper file

4. Click **OK** to open the mapping editor.
5. Expand all the nodes on both sides. You can do it manually, or right-click **<sources>** and select **Expand All**. Do the same for **<target>** on the right-hand side.
6. Map **CCNumber** from the source side to **ccnb** on the target side by dragging and dropping it. Figure 48 shows dragging the element across, and Figure 49 shows what it looks like when it's mapped.

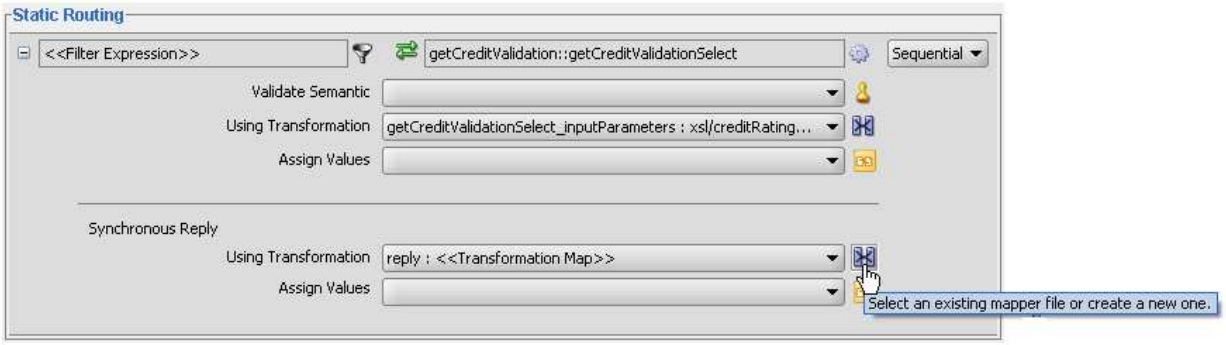
Figure 48 Dragging an dropping an element to map it

Figure 49 The completed mapping



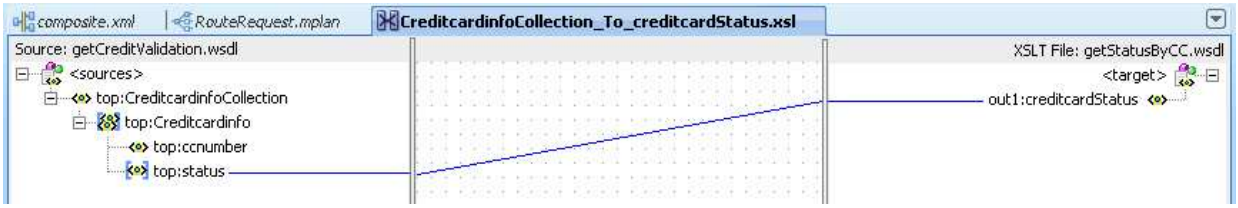
7. Save and close the mapper. You may need to use the tab bar scroll buttons to navigate right to get to the close window icon. Alternatively you can press Ctrl-S to save the mapping, then Ctrl-W to close it.
8. Back in the Mediator editor, click the transformation icon to the right of the **Using Transformation** field in the reply section (that's the second **Using Transformation** field).

Figure 50 Adding a transformation for the request



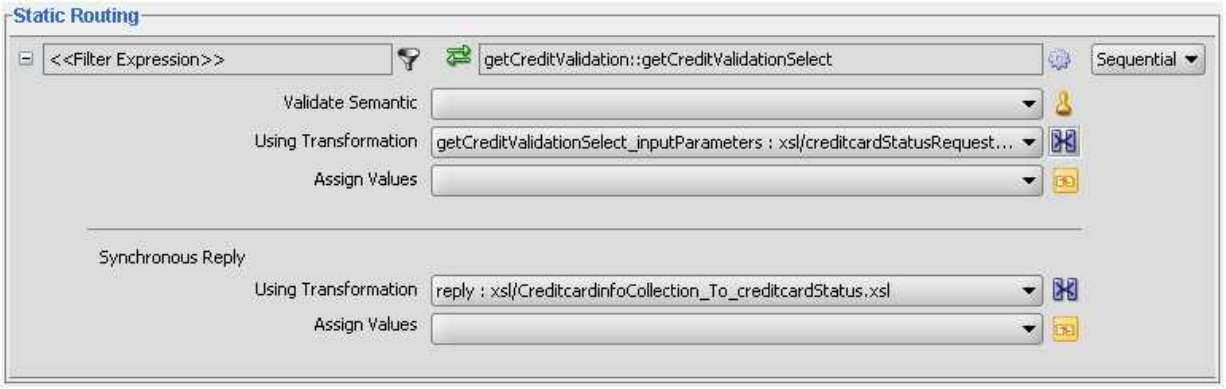
9. In the **Reply Transformation Map** dialog select **Create New Mapper File** and accept the default name.
10. Click **OK** to open the mapper.
11. Expand all the source and target nodes.
12. Map **status** from the source to **creditcardstatus** from the target.

Figure 51 Mapping the reply types



13. Save and close the mapper to return to the Mediator editor.

Figure 52 The Mediator service with mappings defined



14. Save and close the Mediator editor to return to the composite diagram. You can use the toolbar buttons or the menu, or simply press Ctrl-S to save and Ctrl-W to close this tab.

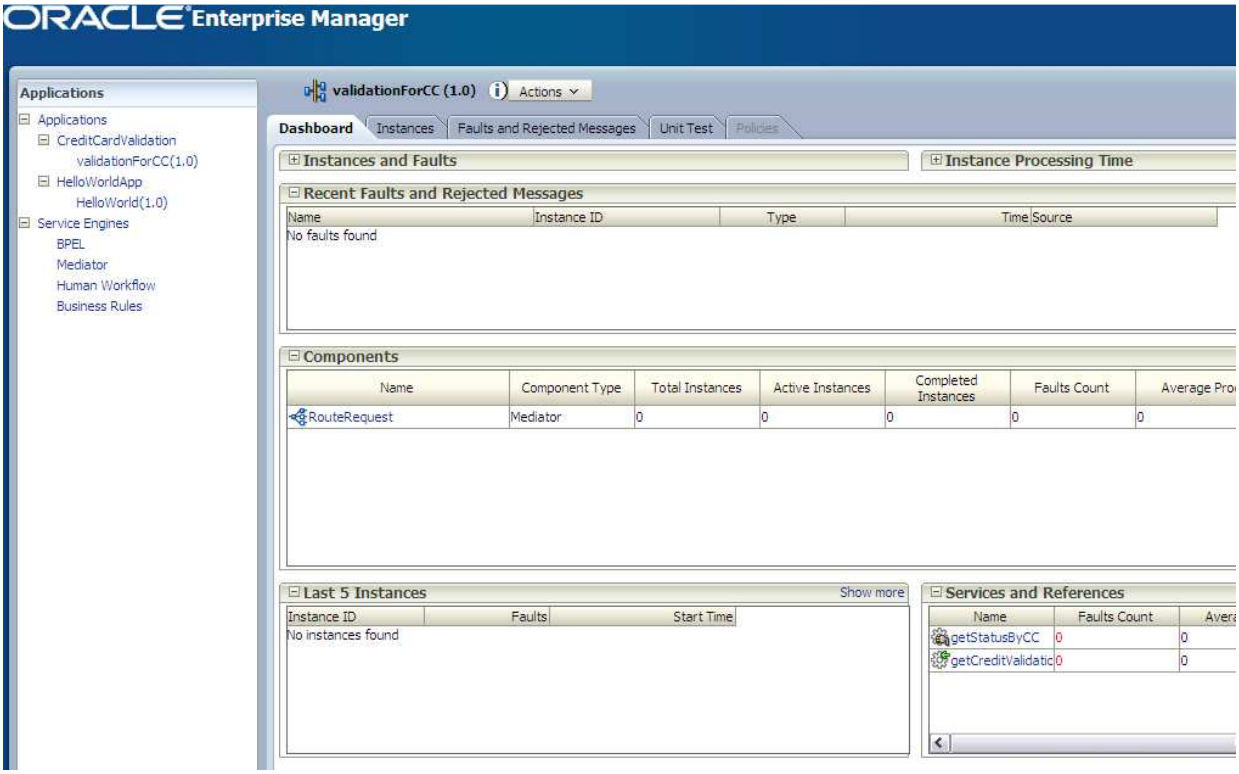
2.3 Deploying the application

The first design iteration is complete and you are now ready to deploy the composite. If you are not sure how to deploy, read the steps in **Appendix A Deploying and Running a Composite Application**.

2.4 Testing the application

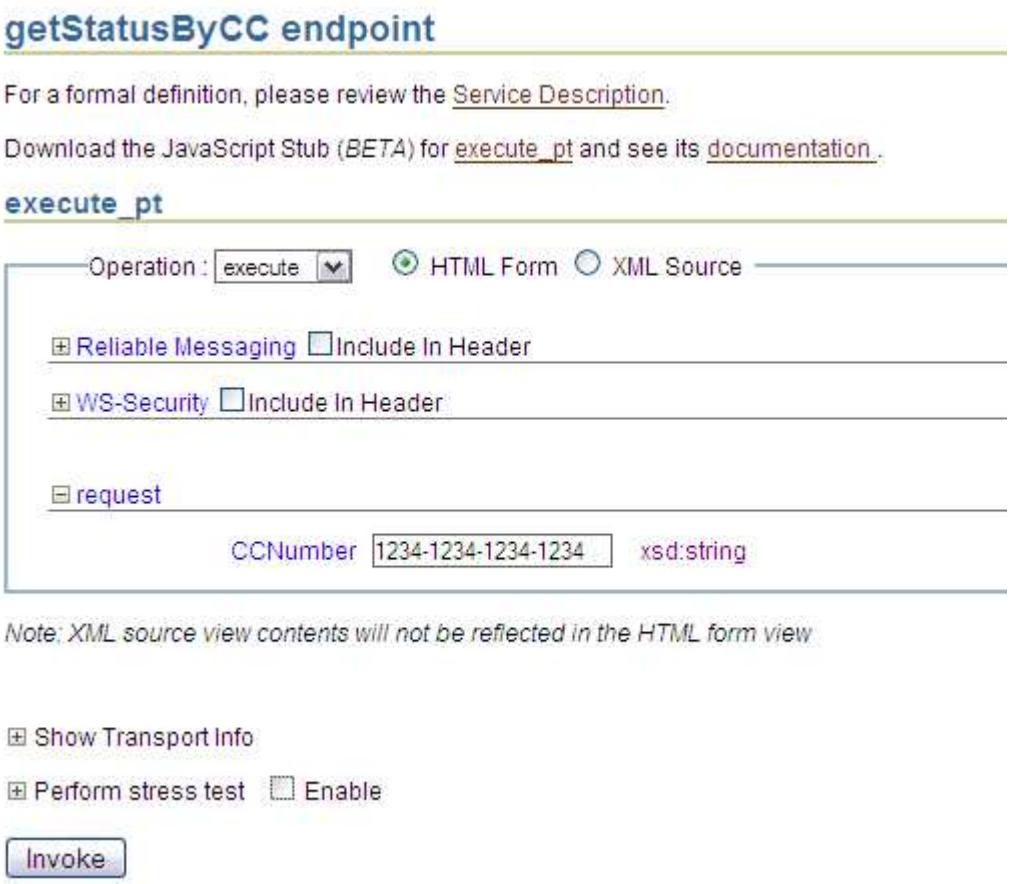
The steps to run and test a composite application are explained in **Appendix A Deploying and Running a Composite Application** in the section titled **Running and testing the application**. Read that section now if this is the first time you are running and testing a composite application.

1. Open the SOA Console.
2. Click the **validationForCC** link in the **Applications** section.

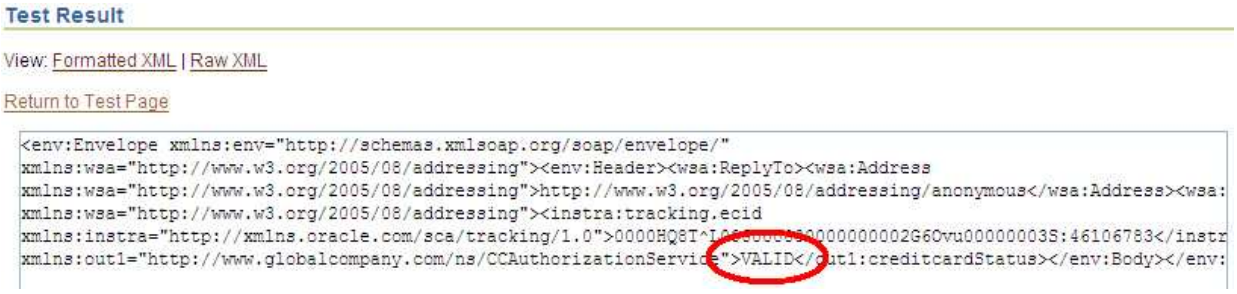


3. Open the test from the Actions menu and click **Test Service - getStatusByCC**.

4. In the **CCNumber** field, enter a credit number. In this case, enter **1234-1234-1234-1234** in the **CCNumber** field which will return a valid credit card,



5. Press the **Invoke** button.
6. You will get a response from the service which will indicate if the credit card is valid or not. Here is the output for a valid credit card:



7. Press the **Back** button of your browser, and this time enter **4321-4321-4321-4321** in the **CCNumber** field which will return an invalid credit card.
8. Click **Invoke** to see the output.

9. In an upcoming chapter, you will create a new application that calls this service, so you'll need to know the WSDL location for it. **Appendix A Deploying and Running a Composite Application** contains instructions on how to get the WSDL location in the section titled **Getting the service description (WSDL)**.

   In this case it will look something like:

http://localhost:8888/soa-infra/services/CreditCardValidation/validationForCC/getStatusByCC?WSDL