

1 End-2-End Purchase Order Processing

1	End-2-End Purchase Order Processing	1
1.1	Introduction.....	1
1.2	How to use this tutorial	1
1.3	Description of the application.....	2
1.3.1	SOA Components	4
1.4	Prerequisites	4
1.5	Project Files.....	4
1.5.1	File Conventions	5
1.6	Setting up the tutorial	5
1.6.1	Installing the database schema	5
1.6.2	Creating the JMS resources.....	5
1.6.3	Creating the database resource	11

1.1 Introduction

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

This SOA composite application contains a database adapter, file adapter, JMS adapter, Web services binding, mediator routing, BPEL process, Business Rule, and Human Task. It introduces each component with a minimum of complexity. The complete application is built in 9 steps.

There are three setup steps required before you begin: 1) install the database schema, 2) create the JMS resources, and 3) create the database resources. These setup steps are described later in this chapter.

1.2 How to use this tutorial

This tutorial was written to provide the most flexibility for all types of users. The ready-to-deploy solution at the end of each chapter is available so you may choose to start the tutorial at any chapter, beginning with the solution for the previous chapter. In addition, the instructions for each chapter are written at three levels of complexity: 1) detailed step by step instructions, 2) high level functional and design notes, and 3) quick-build list of objects.

This tutorial can be used in the following ways.

1. **As a demo** – You complete the setup described in this chapter and deploy chapter 9 to skip design time and to demo the multiple components of this application. Go to

the solutions folder for Chapter 9 to get the composites you need to deploy. If needed, see Appendix A for more details on running the servers, deploying, and testing the application.

2. **Complete tutorial for new users** – You complete each chapter sequentially using the step by step instructions. When you are finished, you will have an introduction to all of the many components of this application, both design and runtime. This complete tutorial takes 4-6 hours, depending on your level of experience.
3. **Selected tutorial for all users** – To run any single chapter to learn about a particular component, first complete the setup in this chapter and then simply open the solution from the previous chapter and begin the next chapter. The time to complete a chapter takes from about 10-60 minutes, depending on your experience.
4. **Tutorial for advanced users** – If you have experience with SOA 10g and or 11g you may choose to design the solutions using only the introduction and design notes available with each chapter. These notes describe the functional aspects of the chapter and some details on the solution design. Of course, the advanced user could design the solution as desired. The time to complete this challenge for experience users varies with their level of experience.
5. **Quick build** – Each chapter includes a final section called *Operations and naming* which lists the objects and their names used in the chapter. Experienced users can use this quick-build list of objects to build the designed solution quickly, completing the entire application in less than 2 hours.

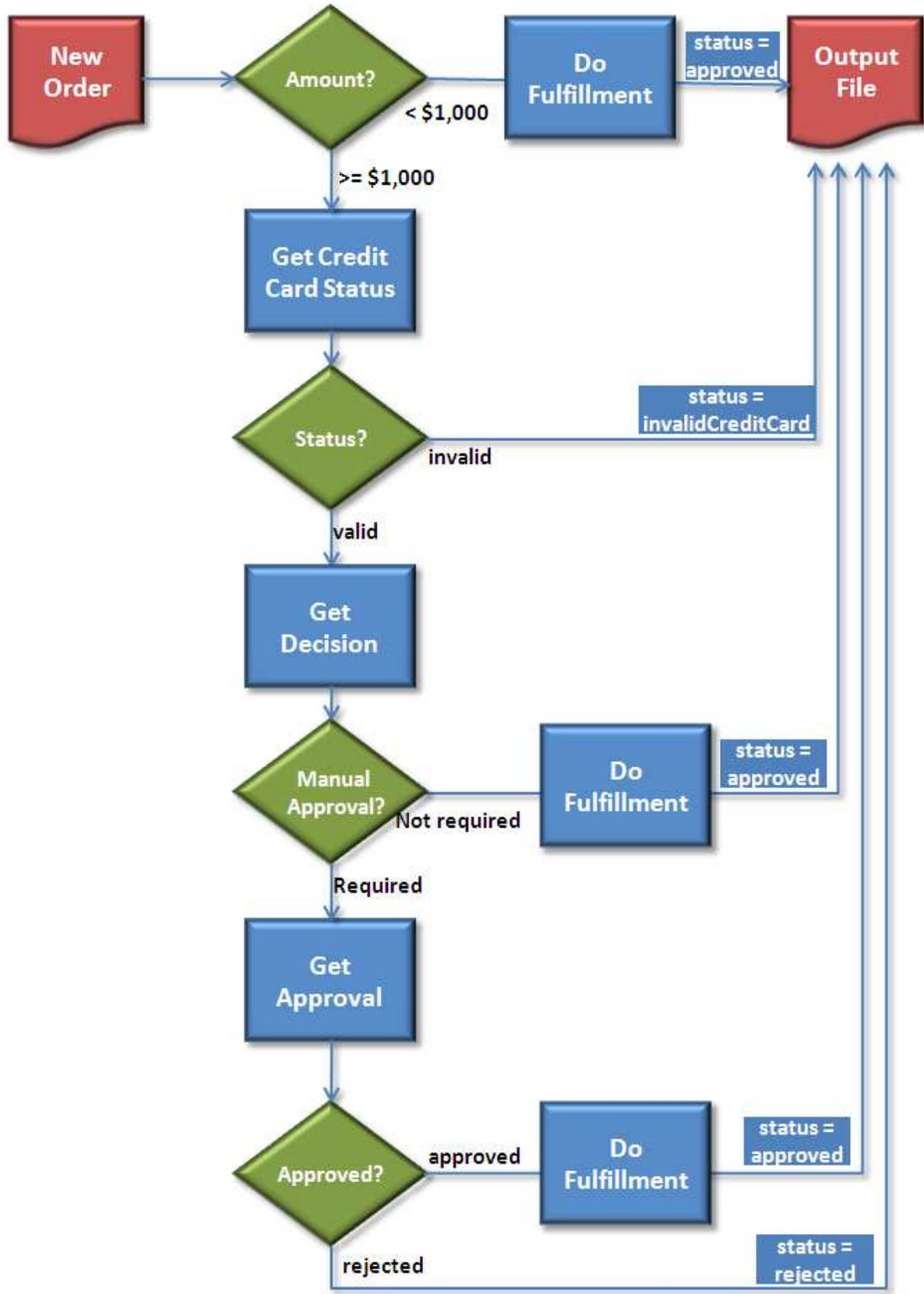
1.3 Description of the application

The application you will build is the back-end processing of a new order. The business process to be implemented is as follows.

- All orders will be written to a text file for later reference.
- Small orders (defined as those under \$1,000) are to be automatically approved.
- Large orders (those greater than or equal to \$1,000) go through a validation and approval process.
 - The customer's credit card must be validated.
 - If the order is \$5,000 or more, a customer service representative must manually approve the order.
- Approved orders shall have the status "approved".
- Large orders with an invalid credit card shall have the status "invalidCreditCard".
- Large orders rejected by the customer service representative shall have the status "rejected".
- All approved orders are sent to the fulfillment service which uses the order value to determine the fulfillment vendor: <1000 goes to USPS, >= 1000 and < 5000 goes to UPS, > 5000 goes to FedEx.

Figure 1 shows a visual representation of the business process.

Figure 1 Visual view of the business process



1.3.1 SOA Components

In terms of the SOA application you will create the following

- A service that accepts new orders in XML format.
- A File Adapter service that can write XML messages (such as orders) to a file.
- A BPEL service implementing the large order process
- The credit card validation service used by the large order BPEL service to validate a given credit card.
- A Database Adapter service used by the credit card validation service to retrieve the status of a given credit card from the database.
- A BPEL service implementing the Fulfillment process
- A JMS adapter service that produces messages for the fulfillment vendors
- A content-based Mediator service to route small orders to the file adapter and fulfillment process and large orders to the large order process.
- The large order BPEL service also uses a Business Rule component to determine if manual approval is required and seeks human approval using a Human Task component.
- The Fulfillment BPEL service also uses a Business Rule component to determine the fulfillment vendor.

1.4 Prerequisites

This tutorial requires the following:

- An Oracle database
- Oracle JDeveloper 11g
- Oracle SOA Suite 11g
- SOA Foundation materials: SOA11gFoundationTutorial.zip

1.5 Project Files

This tutorial comes with several directories and files:

- `doc` – contains the instructions for creating and running the application
- `input` – a few files with sample input data
- `sql` – two database scripts used during setup
- `schemas` – xsd files used throughout sample when defining services
- `ch2` – `ch9` – a solution project for each chapter
- Other directories and files used in the appendix chapters

1.5.1 File Conventions

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

1.6 Setting up the tutorial

This tutorial requires database and JMS resources setup.

1.6.1 Installing the database schema

This demo requires a table in the database owned by the `soademo` user.

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

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

3. Now, create the credit card info table. It is ok to run this script again even if the table already exists. Be sure to run it with the `soademo` user for proper ownership.

```
cd c:\po\sql
sqlplus soademo/soademo @create_creditrating_table.sql
```

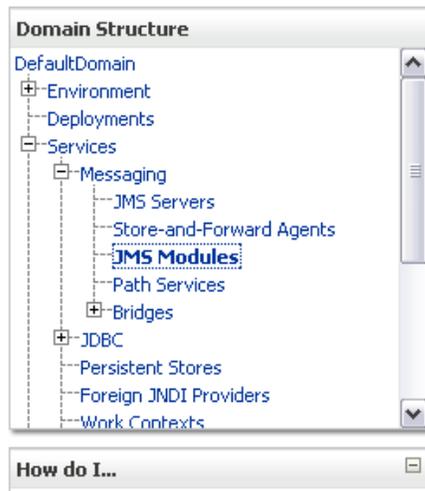
1.6.2 Creating the JMS resources

This application requires a JMS resource. You create this using the WebLogic Server console. First you create the JMS queue and connection factory and then a connection pool for that connection factory.

1. Make sure your server is started using the instructions in Appendix A. If it is not already open, open `http://localhost:7001/console` to start the Web Logic Server (WLS) console and login using `weblogic/welcome1`. Replace the host and port and username/password to match your own configuration.

First add the JMS Queue.

2. In the WLS console, on left navigation bar, expand `Services > Messaging` and click `JMS Modules`.
3. Click `SOAJMSModule` (click on the name, not the checkbox)

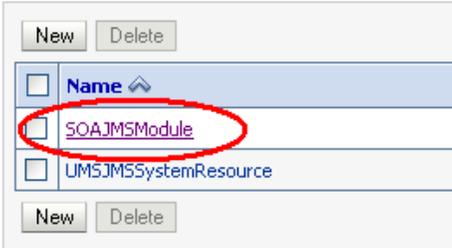


JMS system resources are configured and stored as distributed queues, distributed topics, foreign servers:

This page summarizes the JMS system modules that h

[Customize this table](#)

JMS Modules



4. In the **Summary of Resources** table, click **New**
5. Select the resource type: **Queue** and click **Next**
6. Enter name: *demoFulfillmentQueue* and JNDI name: *jms/demoFulfillmentQueue*

Create a New JMS System Module Resource

Back Next Finish Cancel

JMS Destination Properties

The following properties will be used to identify your new Queue. The current module is SOAJMSModule.

* Indicates required fields

* **Name:**

JNDI Name:

Template:

Back Next Finish Cancel

7. Click **Next**
8. Select **Subdeployment** *SOASubDeployment*
9. Select JMS Server *SOAJMSServer*

Create a New JMS System Module Resource

Back Next Finish Cancel

The following properties will be used to target your new JMS system module resource

Use this page to select a subdeployment to assign this system module resource. A subdeployment is a mechanism by which a SAF agent. If necessary, you can create a new subdeployment by clicking the **Create a New Subdeployment** button on the module's subdeployment management page.

Select the subdeployment you want to use. If you select (none), no targeting will occur.

Subdeployments: SOASubDeployment

What targets do you want to assign to this subdeployment?

Targets :

JMS Servers
<input checked="" type="radio"/> SOAJMServer
<input type="radio"/> UMSJMServer

Back Next Finish Cancel

10. Click Finish

Now add the Connection Factory.

11. In the Summary of Resources table, click New

12. Select the resource type: Connection Factory and click Next

13. Enter name: demoCF and JNDI name: jms/demoCF.

Create a New JMS System Module Resource

Back Next Finish Cancel

Connection Factory Properties

The following properties will be used to identify your new connection factory. The current module is SOAJMSModule.
* Indicates required fields

What would you like to name your new connection factory?

* Name: demoCF

What JNDI Name would you like to use to look up your new connection factory?

JNDI Name: jms/demoCF

Back Next Finish Cancel

14. Click Next

15. Click Finish

You should see both the queue and the connection factory listed in the **Summary of Resources** as shown here.

<input type="checkbox"/>	demoCF	Connection Factory	jms/demoCF	Default Targetting	DefaultServer
<input type="checkbox"/>	demoFulfillmentQueue	Queue	jms/demoFulfillmentQueue	SOASubDeployment	SOAJMServer

Now add the connection pool. The connection pool is configured in the JMSAdapter application and uses a Deployment Plan. First, create a directory to contain that plan.

16. Create `C:\Oracle\Middleware\home_LOAD16F\Oracle_SOA1\soa\JMSPlan`(adjust path according to your installation)
17. In the left navigation bar of the WLS console, click **Deployments**
18. Click **JMS Adapter** (click on the name, not the checkbox)
19. Click the **Configuration** tab and then the **Outbound Connection Pools** tab

Configuration editing is enabled. Future changes will automatically be activated as you modify, add or delete items in this domain.

Domain Structure

- DefaultDomain
 - Environment
 - Deployments**
 - Services
 - Messaging
 - JMS Servers
 - Store-and-Forward Agents
 - JMS Modules
 - Path Services
 - Bridges
 - JDBC
 - Persistent Stores
 - Foreign JNDI Providers
 - Work Contexts

Settings for **JmsAdapter**

Overview | Deployment Plan | **Configuration** | Security | Targets | Control | Testing | Monitoring | Notes

General | Properties | **Outbound Connection Pools** | Admin Objects | Workload | Instrumentation

This page displays a table of Outbound Connection Pool groups and instances for this resource adapter. The top level entries are by connection factory interface and the instances are listed by their JNDI names. Expand a group to obtain configuration information for a group. Click the name of a group or instance to configure it. Automatically generated Connection Pools are not displayed.

Outbound Connection Pool Configuration Table

New Delete

<input type="checkbox"/>	Groups and Instances	Connection Factory
<input type="checkbox"/>	oracle.tip.adapter.jms.IJmsConnectionFactory	oracle.tip.adapter.jms.IJmsConnectionFactory

New Delete

20. Click **New**
21. Select the factory `oracle.tip.adapter.jms.IJmsConnectionFactory`

Outbound Connection Groups

<input type="checkbox"/>	Outbound Connection Group
<input checked="" type="checkbox"/>	oracle.tip.adapter.jms.IJmsConnectionFactory

Back | Next | Finish | Cancel

22. Click **Next**
23. Enter `eis/Queue/demo`.

Create a New Outbound Connection

Back Next Finish Cancel

JNDI name for Outbound Connection Instance

Enter the JNDI name that you want to use to obtain the new connection instance

* Indicates required fields

The Outbound Connection instance represents a connection pool. The JNDI name can be used to obtain the pool at runtime.

 * JNDI Name:

Back Next Finish Cancel

24. Click **Finish**

25. At this point, you should be asked for the deployment plan location. Click on the path to the directory you created earlier and enter *Plan.xml* for the plan name.

Save Deployment Plan

You have made configuration changes that need to be stored in a new deployment plan.

Select or enter the path of a deployment plan file. The path must end with '.xml'. It is highly recommended that this file be named 'Plan.xml'.

Each plan should be located in its own directory, otherwise applications can inadvertently share deployment plan files. The plan file will be overwritten.

Path:

Recently Used Paths: (none)

Current Location: localhost \ C: \ Oracle \ Middleware \ home_LOAD16F \ Oracle_SOA1 \ soa

-  applications
-  connectors
-  modules
-  thirdparty

OK Cancel

26. If there is a plan file selected with the radio button at the bottom of the screen, the one you enter will take precedence. Click **Next** and then click **Finish**.

27. Verify the plan name is set to *Plan.xml* in the *JMSPPlan* directory.

Settings for JmsAdapter

Overview | Deployment Plan | Configuration | Security | Targets | Control | Testing | Monitoring | Notes

Save

This page displays basic information about this resource adapter deployment.

Name:	JmsAdapter	The Ser
Source Path:	C:\Oracle\Middleware\home_0309\Oracle_SOA1\soa\connectors\JmsAdapter.rar	The Ser
Deployment Plan:	C:\Oracle\Middleware\home_0309\Oracle_SOA1\soa\JMSPlan\Plan.xml	The Infoc
Staging Mode:	nostage	The sou duri
Security Model:	DDOnly	The

Go back to the new connection pool to hook up the connection factory

28. Click the **Configuration** tab
29. Expand the factory: *oracle.tip.adapter.jms.IJmsConnectionFactory*
30. Click on *eis/Queue/demo*

Outbound Connection Pool Configuration Table

New Delete

<input type="checkbox"/>	Groups and Instances ^	Connection Factory
<input type="checkbox"/>	 oracle.tip.adapter.jms.IJmsConnectionFactory	oracle.tip.adapter.jms.
<input type="checkbox"/>	eis/activemq/Queue	oracle.tip.adapter.jms.
<input type="checkbox"/>	eis/ajms/Queue	oracle.tip.adapter.jms.
<input type="checkbox"/>	eis/ajms/Topic	oracle.tip.adapter.jms.
<input type="checkbox"/>	eis/fioranomq/Topic	oracle.tip.adapter.jms.
<input type="checkbox"/>	eis/jbossmq/Queue	oracle.tip.adapter.jms.
<input type="checkbox"/>	eis/pramati/Queue	oracle.tip.adapter.jms.
<input type="checkbox"/>	 eis/Queue/demo	oracle.tip.adapter.jms.
<input type="checkbox"/>	eis/sunmq/Queue	oracle.tip.adapter.jms.

31. To change the property value, you must use the ENTER key and then **Save**. Do not use the TAB key. Follow these instructions exactly: Select the cell on the far right for the property **ConnectionFactoryLocation**. An edit box appears. Type in the value *jms/demoCF* and press ENTER. You must use the ENTER key for the value to be entered in the field.

Outbound Connection Properties

Save Showing 1 to 7 of 7 Previous | Next

<input type="checkbox"/>	Property Name ^	Property Type	Property Value
<input type="checkbox"/>	AcknowledgeMode	java.lang.String	AUTO_ACKNOWLEDGE
<input type="checkbox"/>	ConnectionFactoryLocation	java.lang.String	jms/demoCF
<input type="checkbox"/>	FactoryProperties	java.lang.String	
<input type="checkbox"/>	IsTopic	java.lang.Boolean	false
<input type="checkbox"/>	IsTransacted	java.lang.Boolean	false
<input type="checkbox"/>	Password	java.lang.String	
<input type="checkbox"/>	Username	java.lang.String	

Save Showing 1 to 7 of 7 Previous | Next

32. Click **Save**. You must use the **Save** button to save the value.
33. Now the adapter must be redeployed. Click **Deployments** in the left nav bar.
34. Select the checkbox next to JMSAdapter
35. Click **Update** at the top of the Deployment table.
36. Verify that the correct deployment plan is selected

Update Application Assistant

Back Next Finish Cancel

Locate new deployment files

You have elected to update the JmsAdapter application.

Update this application in place with new deployment plan changes. (A deployment plan must be specified for this option)

Deployment plan path: C:\Oracle\Middleware\home_0309\Oracle_SOA1\soa\JMSPlan\Plan.xml

Redeploy this application using the following deployment files:

Source path: C:\Oracle\Middleware\home_0309\Oracle_SOA1\soa\connectors\JmsAdapter.rar

Deployment plan path: C:\Oracle\Middleware\home_0309\Oracle_SOA1\soa\JMSPlan\Plan.xml

Back Next Finish Cancel

37. Click **Next**
38. Click **Finish**

1.6.3 Creating the database resource

This application requires a database data source. You create this using the WebLogic Server console. First create the data source and then a connection pool for that data source.

39. Make sure your server is started using the instructions in Appendix A. If it is not already open, open <http://localhost:7001/console> to start the Web Logic Server (WLS) console and login using `weblogic/welcome1`. Replace the host and port and username/password to match your own configuration.
40. On the left navigation bar, Click **Services > JDBC > Data Sources**.
41. In the data source table, click **New**.

42. Enter the data source information

Name: *soademoDatabase*

JNDI Name: *jdbc/soademoDatabase*

Database Type: *Oracle*

The Database driver defaults to the correct driver: Oracle's Driver (Thin XA) for Instance connections

43. Click **Next**, click **Next** again

44. Enter the database information.

Database Name: XE (your database SID)

Host name: localhost (host where your database is running)

Port: 1521 (set according to your configuration)

Database user name: soademo (created in previous section)

Database user password: soademo

Create a New JDBC Data Source

Back Next Finish Cancel

Connection Properties
Define Connection Properties.

What is the name of database you would like to connect to?

Database Name: XE

What is the name or IP address of the database server?

Host Name: localhost

What is the port on the database server used to connect to the database?

Port: 1521

What database account user name do you want to use to create database connections?

Database User Name: soademo

What is the database account password to use to create database connections?

Password:

Confirm Password:

Back Next Finish Cancel

45. Click **Next**

46. Click **Test Configuration**. Confirm success message at top of page.

47. Click **Next**

48. Select the **Target server** where your SOA component is running: *soa_server1*.

49. Click **Finish**

Now create the connection pool. You have to edit the database adapter application and it uses a Deployment Plan. First, create a directory to contain that plan.

50. Create *C:\Oracle\Middleware\home_LOAD16F\Oracle_SOA1\soa\DBPlan*(adjust path according to your installation)

51. In the left navigation bar, click **Deployments**.

52. Click the DbAdapter application (click the name, not the checkbox)

53. Click the **Configuration** tab, and then click the **Outbound Connection Pools** tab.

54. Click **New**
55. Select the radio button for `javax.resource.cci.ConnectionFactory` and click **Next**
56. Enter the **JNDI Name** as follows: `eis/DB/soademoDatabase`
 Note that this is not the same value as in above step. It must match the value you enter in your database connection you create when building your application later using JDeveloper.
57. Click **Finish**
58. At this point, you should be asked for the deployment plan location. Click on the path to the directory you created earlier and enter the deployment plan name `Plan.xml`.

Update Application Assistant

Back Next Finish Cancel

Select a deployment plan.

Select or enter a deployment plan for this app. The file must exist and have a .xml extension.

Path: C:\Oracle\Middleware\home_0309\Oracle_SOA1\soa\DBPlan\Plan.xml

Recently Used Paths:
 C:\Oracle\Middleware\home_0309\Oracle_SOA1\soa\JMSPlan
 C:\Oracle\Middleware\home_0309\Oracle_SOA1\soa\connectors
 C:\Oracle\Middleware\home_0309\Oracle_SOA1\soa\DBPlan

Current Location: localhost \ C: \ Oracle \ Middleware \ home_0309 \ Oracle_SOA1 \ soa \ DBPlan

Back Next Finish Cancel

59. Click **Next** and then click **Finish**
60. Confirm the name of the deployment plan

Settings for DbAdapter

Overview Deployment Plan Configuration Security Targets Control Testing Monitoring Notes

Save

This page displays basic information about this resource adapter deployment.

Name:	DbAdapter	The
Source Path:	C:\Oracle\Middleware\home_0309\Oracle_SOA1\soa\connectors\DbAdapter.rar	The Serv
Deployment Plan:	C:\Oracle\Middleware\home_0309\Oracle_SOA1\soa\DBPlan\Plan.xml	The Info
Staging Mode:	nostage	The sour durir
Security Model:	DDOnly	The Info

Deployment Order: []

61. Now, edit the connection pool to reference the data source. Click the **Configuration** tab, expand the connection factory and click your new connection pool (click on the name, not the checkbox)
62. To change the property value, you must use the ENTER key and then **Save**. Do not use the TAB key. Follow these instructions exactly: In the **Properties** table, select the box to the far right of **xADataSource**. The edit box appears. Type in your data source name that you created above: *jdbc/soademoDatabase*. Press ENTER key to apply the value. Select **Save**. You must use the ENTER key for the value to be entered in the field. You must use the **Save** button to save the value.
63. Go back to the main **Deployments** page to **Update** the DbAdapter. Click **Deployments** in the left navigation bar.
64. Select the checkbox next to DbAdapter.
65. Click **Update**.
66. Select **Redeploy this application** and confirm the deployment plan location.

67. Click **Finish**
68. Confirm that the connection pool is added by going back to the **DbAdapter > Configuration > Outbound Connection Pools** and expand the connection factory listed there.
69. Confirm the value of the **xADataSource** property that you entered previously. Look closely! This is the most common place where the configuration is in error.
70. Go back to the JMSAdapter **Outbound Connection Pools** tab and review the connection factory property value for the *eis/Queue/demo* connection pool that you entered previously. Look closely! This is the most common place where the configuration is in error.

When the two resources are added and confirmed, you are ready to start building the SOA composites. Continue with Chapter 2.