

IBM WebSphere Adapter for JDBC

7.0.0.0

Quick Start Tutorials

The official version of this document is stored online. Any hard copy versions of this document are for REFERENCE ONLY. Users of this document are personally responsible for using the official version, and for verifying that any copies are complete and of the official version.

Note: Before using this information and the product it supports, read the information in the "Notices" section, at the end of this document.

This edition applies to version 7, release 0, modification 0 of IBM WebSphere Adapter for JDBC and to all subsequent releases and modifications until otherwise indicated in new editions.

© Copyright International Business Machines Corporation 2009. US Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Table of contents

Table of contents	i
Chapter 1. Introduction.....	1
Learning objectives	1
Audience	1
Software prerequisites.....	1
Chapter 2. Tutorial 1: Creating a record using parent child business objects with a CreateSP operation associated with the child business objects (Oracle)	2
Prepare to run through the tutorial.....	3
Extract the sample files.....	3
Configuration prerequisites	3
Configure the adapter for outbound processing	21
Set connection properties for the external service wizard	26
Select the business objects to be used with the adapter.....	27
Generating business object definitions and related artifacts	37
Deploy the module to the test environment	43
Test the assembled adapter application.....	46
Clear the sample content	51
Chapter 3. Tutorial 2: Creating a record using parent-child business objects with a CreateSP associated with the child business object (SQL Server)	52
Prepare to run through the tutorial.....	52
Extract the sample files.....	52
Configuration prerequisites	52
Configure the adapter for outbound processing	72
Set connection properties for the external service wizard	76
Select the business objects and services to be used with the adapter.....	77
Generate business object definitions and related artifacts	88
Deploy the module to the test environment	94
Test the assembled adapter application.....	97
Clear the sample content	102

Chapter 4. Tutorial 3: Creating and executing stored procedure business objects with complex data types (Oracle) 103

Prepare to run through the tutorial.....	104
Extract the sample files.....	104
Configuration prerequisites	104
Configure the adapter for outbound processing	122
Set connection properties for the external service wizard	127
Select the business objects to be used with the adapter.....	128
Generate business object definitions and related artifacts	134
Deploy the module to the test environment	141
Test the assembled adapter application.....	144
Clear the sample content	151

Chapter 5. Tutorial 4: Sending Data to Enterprise Information System using BatchSQL (Oracle)..... 152

Prepare to run through the tutorial.....	153
Extract the sample files.....	153
Configuration prerequisites	153
Configure the adapter for outbound processing	172
Set connection properties for the external service wizard	177
Select the business objects and services to be used with the adapter.....	178
Generate business object definitions and related artifacts	186
Deploy the module to the test environment	191
Test the assembled adapter application.....	193
Clear the sample content	196

Chapter 6. Tutorial 5: Receiving events from the Enterprise Information System (Oracle) 197

Prepare to run through the tutorial.....	198
Extract the sample files.....	198
Configuration prerequisites	198
Configure the adapter for inbound processing	218
Set connection properties for the external service wizard	223
Select the business objects and services to be used with the adapter.....	224
Generate business object definitions and related artifacts	227
Set up the components to be part of the Inbound environment	234
Deploy the module to the test environment	240
Test the assembled adapter application.....	241

Clear the sample content	245
Chapter 7. Tutorial 6: Executing a business object created from a stored procedure (DB2)	246
Prepare to run through the tutorial.....	246
Configuration prerequisites	246
Configure the adapter for outbound processing	252
Set connection properties for the external service wizard	259
Select the business objects and services to be used with the adapter.....	260
Generate business object definitions and related artifacts	263
Deploy the module to the test environment	267
Test the assembled adapter application.....	269
Chapter 8. Tutorial 7: Sending data to the DB2 database within XA Transaction (outbound processing)	271
Prepare to run through the tutorial.....	272
Extract the sample files.....	272
Configuration prerequisites	272
Configure the adapter for outbound processing	286
Configure the adapter for outbound processing.....	286
Set connection properties for the external service wizard	293
Select the business objects and services to be used with the adapter.....	294
Generate business object definitions and related artifacts	296
Set up the components to be part of the XA environment.....	300
Deploy the module to the test environment	308
Test the assembled adapter application.....	310
Clear the sample content	312
Chapter 9. Tutorial 8: Sending data to the Oracle database with XA transaction (outbound processing).....	313
Prepare to run through the tutorial.....	314
Extract the sample files.....	314
Configuration prerequisites	314
Configure the adapter for outbound processing	328
Set connection properties for the external service wizard	328
Set up the components to be part of the XA environment.....	342
Deploy the module to the test environment	350
Test the assembled adapter application.....	352
Clear the sample content	354

Chapter 10. Tutorial 9: Receiving events from the Oracle database using data source with prepared statement cache (inbound processing)	355
Prepare to run through the tutorial.....	355
Extract the sample files.....	355
Configuration prerequisites	356
Configure data source statement cache.....	375
Configure the adapter for inbound processing	378
Set connection properties for the external service wizard	383
Select the business objects and services to be used with the adapter.....	384
Generate business object definitions and related artifacts	387
Set up the components to be part of the inbound environment	394
Deploy the module to the test environment	400
Test the assembled adapter application.....	401
Clear the sample content	404
Chapter 11. Tutorial 10: Generate wrapper business objects (Oracle) 405	
Prepare to run through the tutorial.....	405
Extract the sample files.....	405
Configuration prerequisites	406
Configure the adapter for outbound processing	419
Set connection properties for the external service wizard	424
Select the business objects and services to be used with the adapter.....	425
Generate business object definitions and related artifacts	432
Deploy the module to the test environment	438
Test the assembled adapter application.....	440
Clear the sample content	444
Chapter 12. Tutorial 11: Creating business objects for stored procedure and executing stored procedure with Execute operation (SQL Server)	445
Prepare to run through the tutorial.....	446
Extract the sample files.....	446
Configuration prerequisites	446
Configure the adapter for outbound processing	466
Set connection properties for the external service wizard	470
Select the business objects and services to be used with the adapter.....	471
Generate business object definitions and related artifacts	476

Deploy the module to the test environment	484
Test the assembled adapter application.....	487
Clear the sample content	492
Chapter 13 . Tutorial 12: Retrieve business object from database using user defined query (DB2)	493
Prerequisites to run the scenario.....	493
Extract the sample files.....	493
Configuration prerequisites	493
Create an adapter project in WebSphere Integration Developer	498
Configure the adapter for outbound processing	500
Set connection properties for the external service wizard	504
Select the business objects and services to be used with the adapter.....	505
Generate business object definitions and related artifacts	509
Deploy the module to the test environment	516
Test the assembled adapter application.....	517
Clear the sample content	520
Chapter 14 . Tutorial 13: Checking for the existence of a business object (Oracle)	521
Prepare to run through the tutorial.....	521
Extract the sample files.....	521
Configuration prerequisites	521
Configure the adapter for outbound processing	534
Set connection properties for the external service wizard	539
Select the business objects to be used with the adapter.....	540
Generate business object definitions and related artifacts	542
Deploy the module to the test environment	548
Test the assembled adapter application.....	550
Chapter 15 . Tutorial 14: Generate Wrapper business objects for Inbound (Oracle)	555
Prepare to run through the tutorial.....	556
Extract the sample files.....	556
Configuration prerequisites	556
Configure the adapter for inbound processing	574
Set connection properties for the external service wizard	579
Select the business objects to be used with the adapter.....	580
Generate business object definitions and related artifacts	582

Set up the components to be part of the inbound environment	593
Deploy the module to the test environment	599
Test the assembled adapter application.....	602
Chapter 16. Troubleshooting	606
Chapter 17. Notices	608

Chapter 1. Introduction

WebSphere Adapter for JDBC 7.0.0.0 enables the bidirectional connectivity for integration to any database application. The exchange of data for such applications happens at the database level. Updates to the database may need to be applied to another Enterprise Information System (EIS) and changes in an EIS may need to be applied to a database. The JDBC resource adapter can integrate with any database, as long as there is a JDBC driver that supports the JDBC 2.0 or higher specification, available for the database. Examples of such databases include Oracle, Microsoft SQLServer, DB2, Sybase, and Informix.

Learning objectives

After completing the tutorial, you should be able to perform the following tasks:

- Create an adapter project in WebSphere Integration Developer.
 - Discover services and associated business objects from the enterprise information system (EIS) and make them part of the adapter project.
 - Create a deployable module that you install on WebSphere Process Server or WebSphere Enterprise Service Bus.
 - Test the module and validate the results.
-

Audience

These tutorials are for integration developers who design, assemble, test, and deploy business integration solutions.

Software prerequisites

To use these tutorials, you must have the following applications installed:

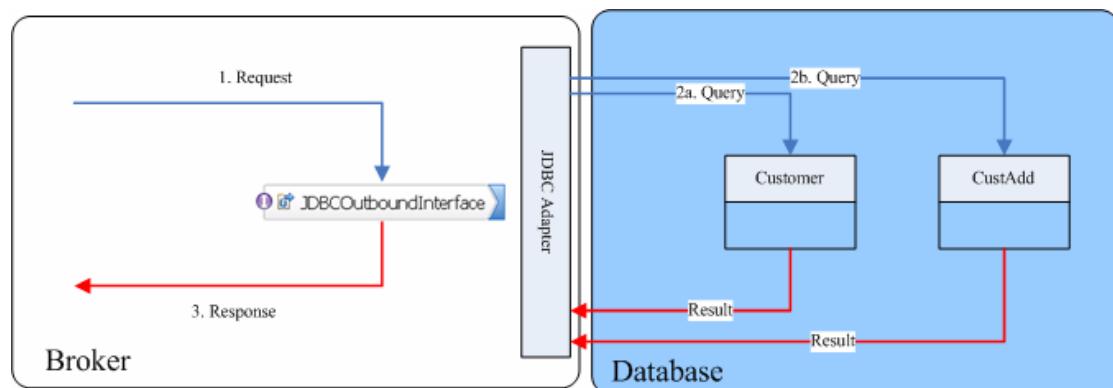
- WebSphere Integration Developer version 7.0.0.0
 - WebSphere Process Server version 7.0.0.0
 - WebSphere Adapter for JDBC version 7.0.0.0
 - JDBC Driver for Oracle
 - JDBC Driver for DB2
 - JDBC Driver for SQLServer
-

Chapter 2. Tutorial 1: Creating a record using parent child business objects with a CreateSP operation associated with the child business objects (Oracle)

This tutorial demonstrates how to create records in a table in a parent child relationship for WebSphere Adapter for JDBC 7.0.0.0. The scenario also demonstrates the use of a stored procedure attached to a business object. A stored procedure is associated with the child business object using the CreateSP verb ASI. The adapter calls the stored procedure to create the record in the child table instead of generating the insert SQL statement.

About this task

In this scenario, an application SCA component raises a create Customer business object request to the JDBC Outbound Interface. The JDBC adapter generates SQL statements to insert corresponding Customer and CustAdd records into the database. Finally, the JDBC adapter generates response according to the input business object and the execution results of the SQL statements. The following figure represents this scenario.



Prepare to run through the tutorial

Extract the sample files

Replicas of the artifacts that you create when using the external service wizard are provided as sample files for your reference. Use these files to verify if the files you create using the external service wizard are correct.

Download the sample zip file and extract it into a directory of your choice (you may want to create a new directory).

Configuration prerequisites

Before configuring the adapter, you must complete the following tasks:

- Create tables and stored procedure
- Create an authentication alias
- Create a data source

Create tables and stored procedure

You must create the following tables and stored procedures in the Oracle database before starting the scenario.

```
CREATE TABLE CUSTOMER (
    PKEY VARCHAR2(10) NOT NULL PRIMARY KEY,
    FNAME VARCHAR2(20) ,
    LNAME VARCHAR2(20) ,
    CCODE VARCHAR2(10) ) ;

CREATE TABLE CUSTADD (
    ADDRID VARCHAR2(10) NOT NULL PRIMARY KEY,
    CUSTID VARCHAR2(10) ,
    CITY VARCHAR2(20) ,
    ZIPCODE VARCHAR2(10) ) ;

CREATE or REPLACE PROCEDURE CREATEADDRESS
(addr_id IN varchar2, cust_id IN varchar2, city IN
varchar2, zipcode IN varchar2)
AS
BEGIN
INSERT into CUSTADD (ADDRID, CUSTID, CITY, ZIPCODE)
values
    (addr_id, cust_id, city, zipcode);
END;
```

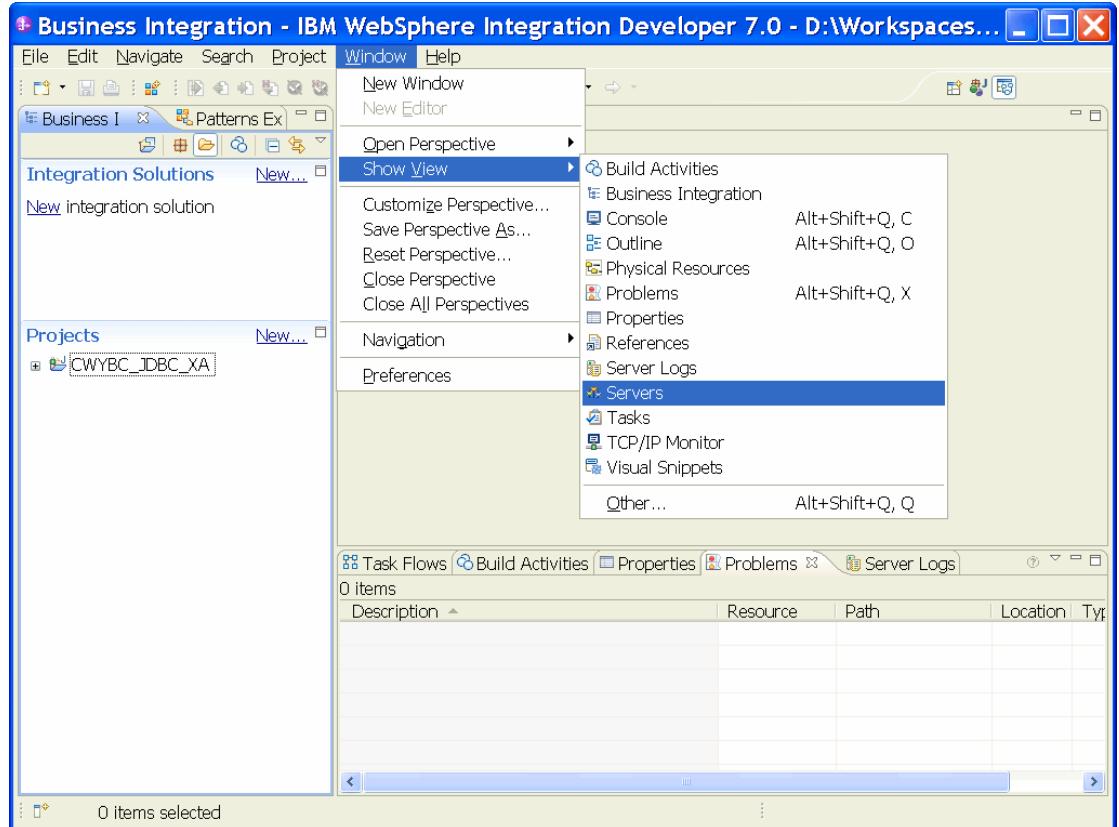
Create an authentication alias

The authentication alias needs to be set because the data source that is used to generate artifacts will use the username and password set in the authentication alias to connect to the database.

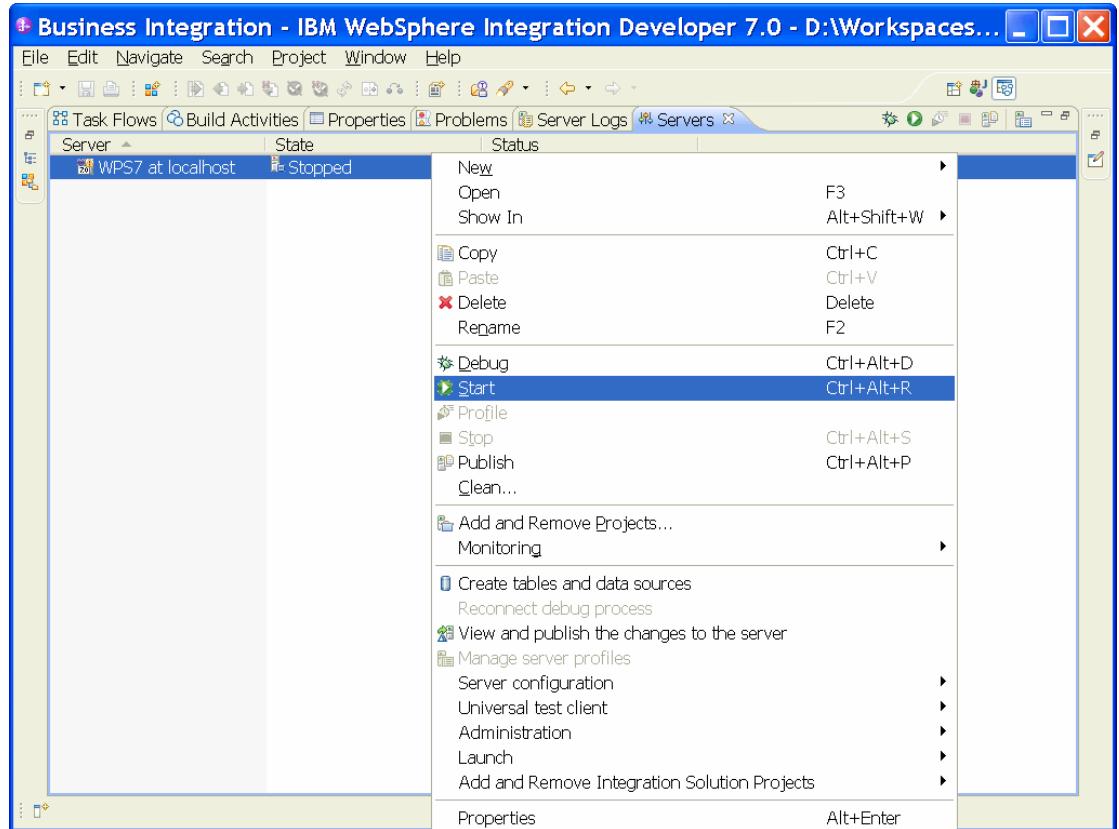
WebSphere software

Follow these steps to set the authentication alias in the WebSphere Process Server administrative console.

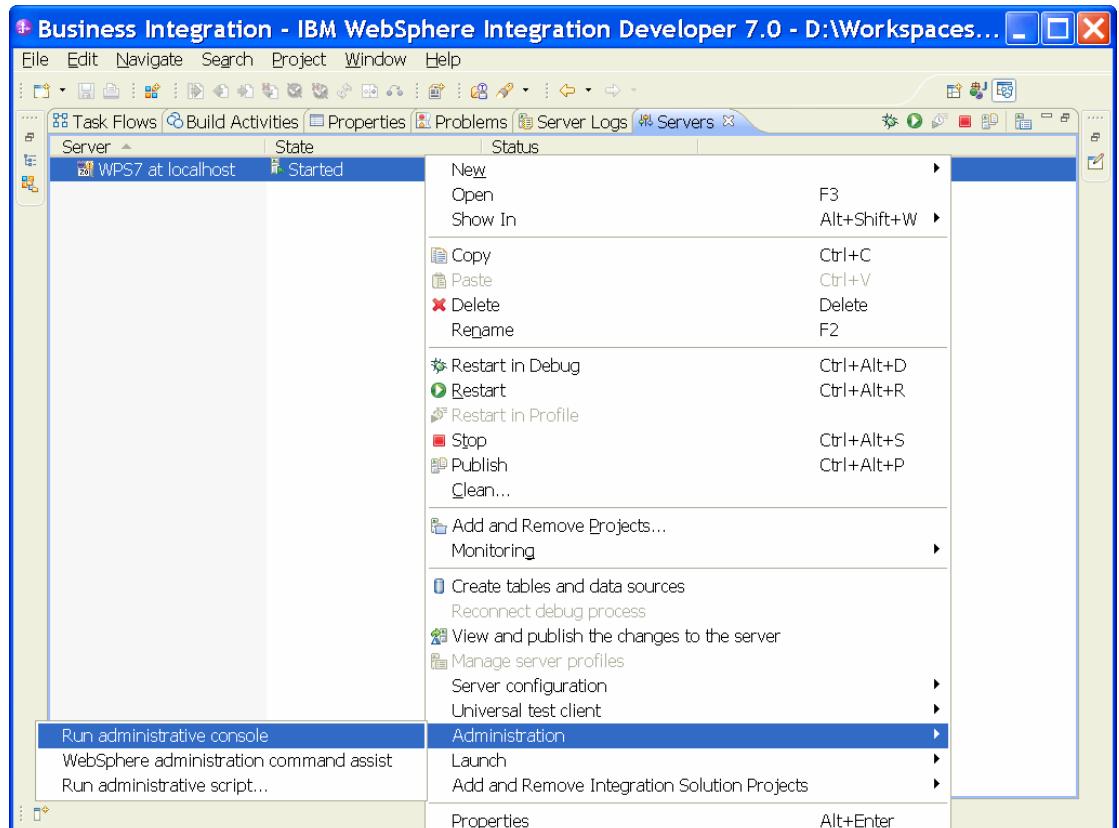
1. In WebSphere Integration Developer, switch to the **Servers** view by selecting **Windows > Show View > Servers**.



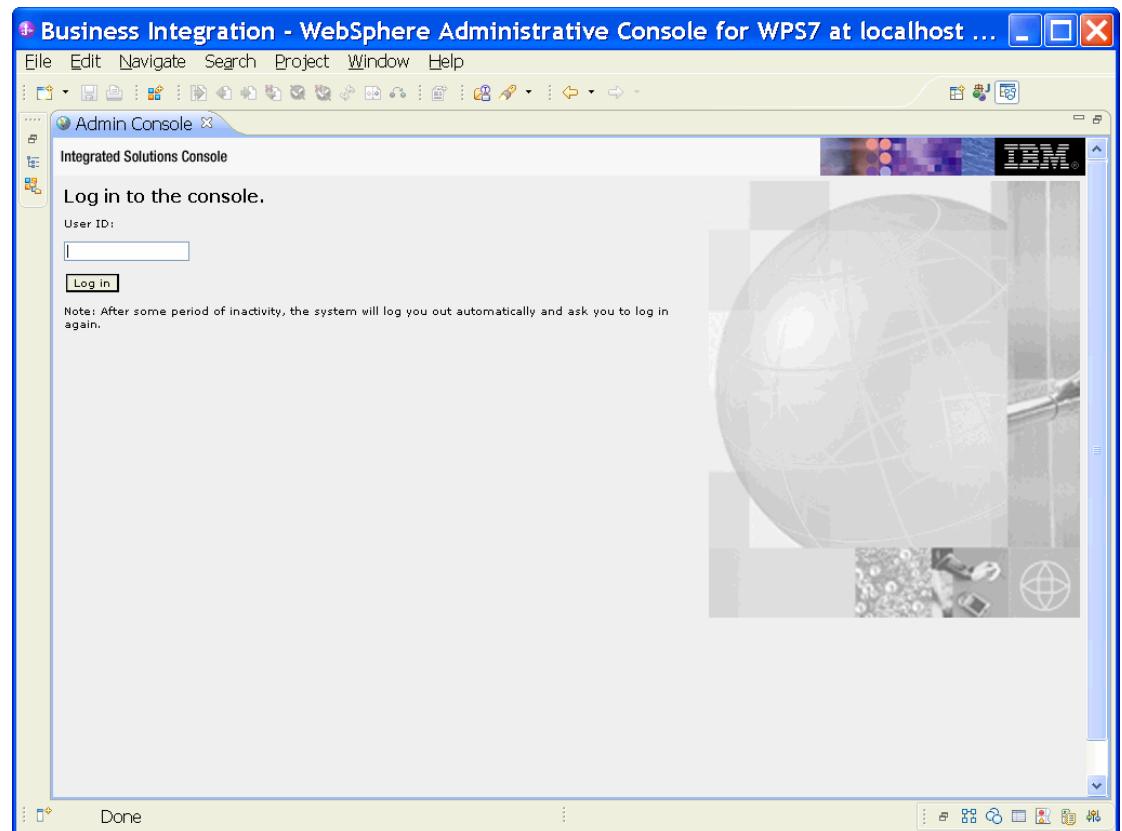
2. In the **Servers** view, right-click the server that you want to start and select **Start**.



- After the server is started, right-click the server, and select **Administration > Run administrative console**.



- Log on to the administrative console.



5. Click **Security** → **Global security**.

WebSphere software

The screenshot shows the left sidebar of the WebSphere software interface. At the top, there is a dropdown menu labeled "View: All tasks". Below it is a hierarchical navigation tree:

- Welcome
- Guided Activities
- Servers
- Applications
- Services
- Resources
- Security
 - Business Integration Security
 - Global security
 - Security domains
 - Administrative Authorization Groups
 - SSL certificate and key management
 - Security auditing
 - Bus security
- Environment
- Integration Applications
- System administration
- Users and Groups
- Monitoring and Tuning
- Troubleshooting
- Service integration
- UDDI

- On the right, click **J2C Authentication Data** under **Java Authentication and Authorization Service**.

The screenshot shows the "Global security" configuration page. At the top, it says "Cell=localhostNode01Cell, Profile=AppSrv01" and has a "Close page" button.

Global security
Use this panel to configure administration and the default application security policy. This security configuration applies to functions and is used as a default security policy for user applications. Security domains can be defined to override and control applications.

Administrative security
 Enable administrative security

- [Administrative user roles](#)
- [Administrative group roles](#)
- [Administrative authentication](#)

Application security
 Enable application security

Java 2 security
 Use Java 2 security to restrict application access to local resources

- Warn if applications are granted custom permissions
- Restrict access to resource authentication data

User account repository
Current realm definition:
Federated repositories

Available realm definitions:
Federated repositories

Authentication
Authentication mechanisms and expiration

- LTPA
- Kerberos and LTPA
[Kerberos configuration](#)
- SWAM (deprecated): No authentication
[Authentication cache settings](#)

J2C authentication data
Specifies a list of user identities and passwords for Java(TM) 2 connector security to use.

Security domains
 External authorization providers
 Custom properties

WebSphere software

A list of existing aliases is displayed.

[Global security > JAAS - J2C authentication data](#)

Specifies a list of user identities and passwords for Java(TM) 2 connector security to use.

Prefix new alias names with the node name of the cell (for compatibility with earlier releases)

[Apply](#)

[+ Preferences](#)

New	Delete			
Select	Alias	User ID	Description	
You can administer the following resources:				
<input type="checkbox"/>	BSpace JDBC Alias	wbiuser	Business Space Authorization Alias	
<input type="checkbox"/>	CommonEventInfrastructureJMSAuthAlias	wbiuser	Authentication alias for the Common Event Infrastructure JMS Topics and Queues	
<input type="checkbox"/>	SCA Auth Alias	wbiuser	This is the alias used by SCA to login to a secured SIBus	
<input type="checkbox"/>	localhostNode01Cell/nlNode01/server1/EventAuthDataAliasDerby		Derby authentication alias for the Event Server	
Total 4				

7. Click **New** to create a new authentication entry. Type the alias name, and username and password to connect to the database. Click **OK**.

WebSphere software

Cell=localhostNode01Cell, Profile=AppSrv01

Global security

[Global security](#) > [JAAS - J2C authentication data](#) > New

Specifies a list of user identities and passwords for Java(TM) 2 connector security to use.

General Properties

* Alias: Alias_Oracle
* User ID: sample
* Password:
Description:

[Apply] [OK] [Reset] [Cancel]

8. Click **Save** to save the changes.

Cell=localhostNode01Cell, Profile=AppSrv01

Global security

Messages

⚠ Changes have been made to your local configuration. You can:

- [Save](#) directly to the master configuration.
- [Review](#) changes before saving or discarding.

⚠ The server may need to be restarted for these changes to take effect.

[Global security](#) > [JAAS - J2C authentication data](#)

Specifies a list of user identities and passwords for Java(TM) 2 connector security to use.

Prefix new alias names with the node name of the cell (for compatibility with earlier releases)

[Apply]

You have created an authentication alias that will be used to configure the data source.

WebSphere software

Preferences			
		New	Delete
Select Alias		User ID	Description
You can administer the following resources:			
<input type="checkbox"/>	BSpace JDBC Alias	wbiuser	Business Space Authorization Alias
<input type="checkbox"/>	CommonEventInfrastructureJMSAuthAlias	wbiuser	Authentication alias for the Common Event Infrastructure JMS Topics and Queues
<input type="checkbox"/>	SCA Auth Alias	wbiuser	This is the alias used by SCA to login to a secured SIBus
<input type="checkbox"/>	localhostNode01Cell/nlNode01/server1/EventAuthDataAliasDerby		Derby authentication alias for the Event Server
<input type="checkbox"/>	nlNode01/AliasOracle	luweiqin	
Total 5			

Create the data source

Create a data source in WebSphere Process Server, which the adapter will use to connect to the database. This data source will be used later when generating the artifacts for the module.

Note: This tutorial will use Oracle as the database and the Oracle thin driver,ojdbc6.jar.

Here are the steps to create the data source in the WebSphere Process Server administrative console.

1. In the administrative console, select **Environment → WebSphere Variables**.

WebSphere software



2. On the right, click **ORACLE_JDBC_DRIVER_PATH** and specify the path of the ojdbc6.jar file in the **Value** field. Click **OK**.

WebSphere software

Cell=localhostNode01Cell, Profile=AppSrv01

WebSphere Variables

[WebSphere Variables](#) > ORACLE_JDBC_DRIVER_PATH

Use this page to define substitution variables. Variables specify a level of indirection for some system-defined values, such as file system root directories. Variables have a scope level, which is either server, node, cluster, or cell. Values at one scope level can differ from values at other levels. When a variable has conflicting scope values, the more granular scope value overrides values at greater scope levels. Therefore, server variables override node variables, which override cluster variables, which override cell variables.

Configuration

General Properties

* Name: ORACLE_JDBC_DRIVER_PATH

Value: D:\Lib

Description: The directory that contains the Oracle thin or oci8 JDBC Driver.

Buttons: Apply, OK, Reset, Cancel

3. Click **Save** to save the changes.

WebSphere Variables

Messages

⚠ Changes have been made to your local configuration. You can:
• [Save](#) directly to the master configuration.
• [Review](#) changes before saving or discarding.
⚠ The server may need to be restarted for these changes to take effect.

The variable is added and appears in the list.

Preferences

New Delete

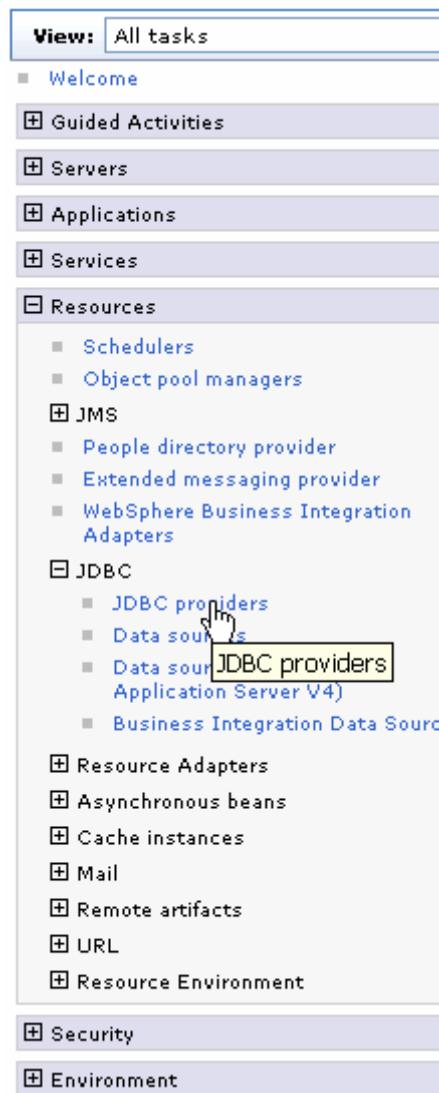
Select Name Value Scope

You can administer the following resources:

<input type="checkbox"/>	MQ_INSTALL_ROOT	`\${WAS_INSTALL_ROOT}/lib/wmq	Node=n1Node01
<input type="checkbox"/>	ORACLE_JDBC_DRIVER_PATH	D:\Lib	Node=n1Node01
<input type="checkbox"/>	OS400_NATIVE_JDBC40_DRIVER_PATH		Node=n1Node01
<input type="checkbox"/>	OS400_NATIVE_JDBC_DRIVER_PATH		Node=n1Node01
<input type="checkbox"/>	OS400_TOOLBOX_JDBC_DRIVER_PATH		Node=n1Node01
<input type="checkbox"/>	ORACLE_JDBC_DRIVER_PATH		Node=n1Node01

WebSphere software

4. Select **Resources** → **JDBC** -> **JDBC Providers**.



5. Click **New** in the JDBC providers window.

JDBC providers

JDBC providers

Use this page to edit properties of a JDBC provider. The JDBC provider object encapsulates the specific JDBC driver implementation class for access to the specific vendor database of your environment. Learn more about this task in a [guided activity](#). A guided activity provides a list of task steps and more general information about the topic.

Scope: Cell=**localhostNode01Cell**, Node=**n1Node01**

Scope specifies the level at which the resource definition is visible. For detailed information on what scope is and how it works, [see the scope settings help](#).

Node=n1Node01

Preferences

Select	Name	Scope	Description
None			
Total 0			

6. In the Create new JDBC provider page, select an Oracle database with a connection pool data source for the Oracle JDBC driver. Click **Next**.

Create a new JDBC Provider

Create a new JDBC Provider

→ Step 1: Create new JDBC provider

Step 2: Enter database class path information

Step 3: Summary

Create new JDBC provider

Set the basic configuration values of a JDBC provider, which encapsulates the specific vendor JDBC driver implementation classes that are required to access the database. The wizard fills in the name and the description fields, but you can type different values.

Scope

cells:localhostNode01Cell:nodes:n1Node01

* Database type

Oracle

* Provider type

Oracle JDBC Driver

* Implementation type

Connection pool data source

* Name

Oracle JDBC Driver

Description

Oracle JDBC Driver

Next → Cancel

7. In the Enter database classpath information page, enter the following value in the **Class path** field:
\$(ORACLE_JDBC_DRIVER_PATH)/ojdbc6.jar, where
\$(ORACLE_JDBC_DRIVER_PATH) is library path for the run time.
8. Click **Next**.

WebSphere software

Cell=localhostNode01Cell, Profile=AppSrv01

Create a new JDBC Provider

Step 1: Create new JDBC provider
→ Step 2: Enter database class path information
Step 3: Summary

Enter database class path information

Set the environment variables that represent the JDBC driver class files, which WebSphere(R) Application Server uses to define your JDBC provider. This wizard page displays the file names; you supply only the directory locations of the files. Use complete directory paths when you type the JDBC driver file locations. For example: C:\SQLLIB\java on Windows(R) or /home/db2inst1/sqllib/java on Linux(TM).

If a value is specified for you, you may click Next to accept the value.

Class path:

`${ORACLE_JDBC_DRIVER_PATH}/ojdbc6.jar`

Directory location for "ojdbc6.jar" which is saved as WebSphere variable `${ORACLE_JDBC_DRIVER_PATH}`

D:\Lib

Previous **Next** Cancel

9. Click **Finish**.

Cell=localhostNode01Cell, Profile=AppSrv01

[Close page](#)

Create a new JDBC Provider

Step 1: Create new JDBC provider
Step 2: Enter database class path information
→ Step 3: Summary

Summary

Summary of actions:

Options	Values
Scope	cells:localhostNode01Cell:nodes:n1Node01
JDBC provider name	Oracle JDBC Driver
Description	Oracle JDBC Driver
Class path	<code> \${ORACLE_JDBC_DRIVER_PATH}/ojdbc6.jar</code>
<code> \${ORACLE_JDBC_DRIVER_PATH}</code>	D:\Lib
Implementation class name	oracle.jdbc.pool.OracleConnectionPoolDataSource

Previous **Finish** Cancel

10. Click **Save** to save the changes.

WebSphere software

Cell=localhostNode01Cell, Profile=AppSrv01

JDBC providers

Messages

Changes have been made to your local configuration.
You can:

- Save directly to the master configuration.
- Review changes before saving or discarding.

The server may need to be restarted for these changes to take effect.

The JDBC provider is added and appears in the list.

Cell=localhostNode01Cell, Profile=AppSrv01

JDBC providers

Use this page to edit properties of a JDBC provider. The JDBC provider object encapsulates the specific JDBC driver implementation class for access to the specific vendor database of your environment. Learn more about this task in a [guided activity](#). A guided activity provides a list of task steps and more general information about the topic.

Scope: Cell=**localhostNode01Cell**, Node=**nINode01**

Scope specifies the level at which the resource definition is visible. For detailed information on what scope is and how it works, [see the scope settings help](#).

Node=nINode01

Preferences

New	Delete		
<input type="checkbox"/>	<input type="checkbox"/>		
<input type="checkbox"/>	<input type="checkbox"/>		
Select	Name ▲	Scope ▲	Description ▲
You can administer the following resources:			
<input type="checkbox"/>	Oracle JDBC Driver	Node=nINode01	Oracle JDBC Driver
Total 1			

11. Click the Oracle JDBC provider you just created. Under **Additional Properties**, click **Data sources**. Click **New**.

WebSphere software

Cell=localhostNode01Cell, Profile=AppSrv01

The screenshot shows the 'JDBC providers' page with the 'Oracle JDBC Driver' provider selected. Under the 'Data sources' tab, there is a table listing one data source named 'None'. The table has columns for Select, Name, JNDI name, Scope, Provider, Description, and Category. Below the table, it says 'Total 0'. At the top of the page, there are buttons for New, Delete, Test connection, and Manage state... along with some icons.

12. Type any value in the **JNDI name** field, and select the authentication alias. Click **Next**.

Cell=localhostNode01Cell, Profile=AppSrv01

The screenshot shows the 'Create a data source' wizard, Step 1: Enter basic data source information. The left sidebar lists steps: Step 1: Enter basic data source information (current), Step 2: Enter database specific properties for the data source, Step 3: Setup security aliases, and Step 4: Summary. The main panel contains fields for Scope (cells:localhostNode01Cell:nodes:n1Node01), JDBC provider name (Oracle JDBC Driver), Data source name (* Oracle JDBC Driver DataSource), and JNDI name (* OracleDS). The 'Next' button at the bottom is highlighted with a cursor.

13. Provide the appropriate URL value and select a data store helper class name from the **Data store helper class name** list as shown in the following figure. Click **Next**.

WebSphere software

Cell=localhostNode01Cell, Profile=AppSrv01

Create a data source

Step 1: Enter basic data source information

→ Step 2: Enter database specific properties for the data source

Step 3: Setup security aliases

Step 4: Summary

Enter database specific properties for the data source

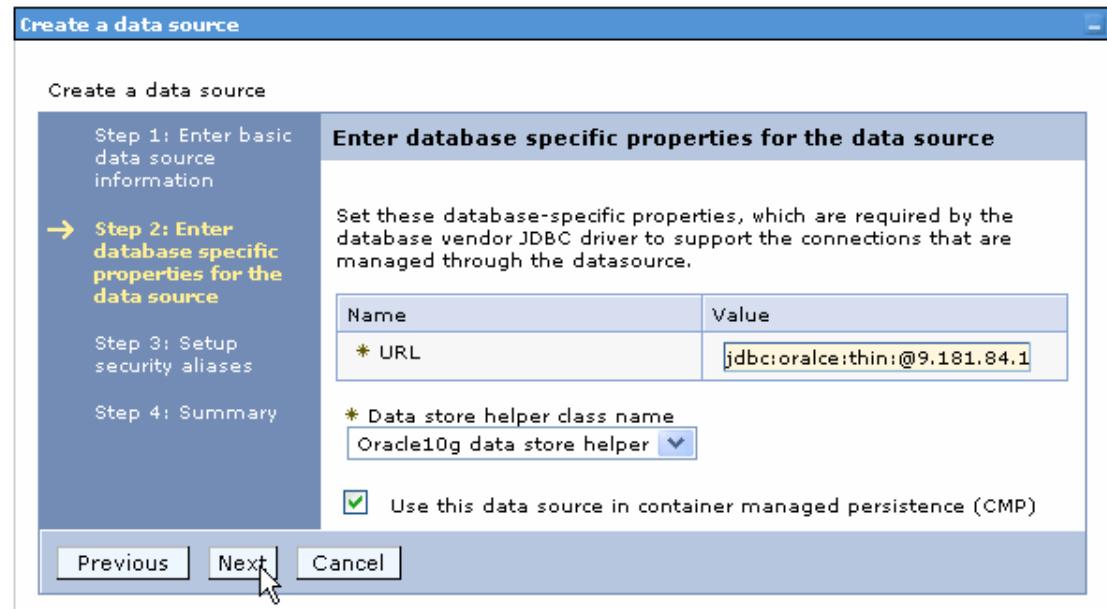
Set these database-specific properties, which are required by the database vendor JDBC driver to support the connections that are managed through the datasource.

Name	Value
* URL	jdbc:oracle:thin:@9.181.84.1

* Data store helper class name
Oracle10g data store helper

Use this data source in container managed persistence (CMP)

Previous Next Cancel



14. Select the authentication alias you just created from the **Component-managed authentication alias** list, and click **Next**.

Cell=localhostNode01Cell, Profile=AppSrv01 Close

Create a data source

Step 1: Enter basic data source information

Step 2: Enter database specific properties for the data source

→ Step 3: Setup security aliases

Step 4: Summary

Setup security aliases

Select the authentication values for this resource.

Component-managed authentication alias
nlNode01/Alias_Oracle

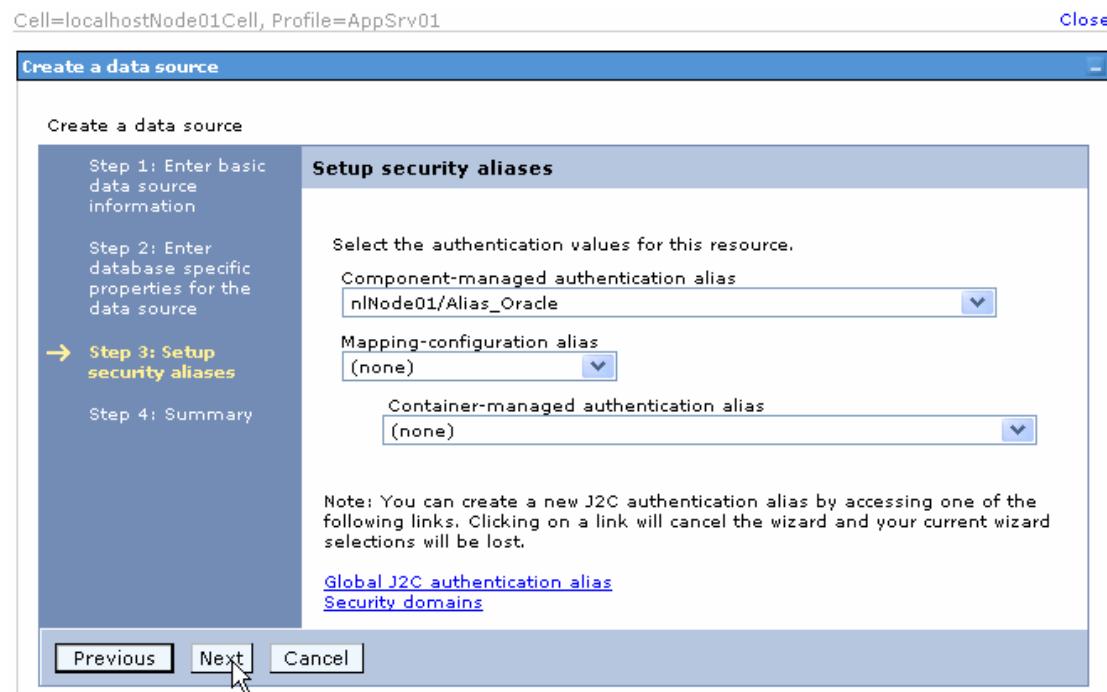
Mapping-configuration alias
(none)

Container-managed authentication alias
(none)

Note: You can create a new J2C authentication alias by accessing one of the following links. Clicking on a link will cancel the wizard and your current wizard selections will be lost.

[Global J2C authentication alias](#)
[Security domains](#)

Previous Next Cancel



The Summary of the values entered for the data source is displayed.

15. Click **Finish**.

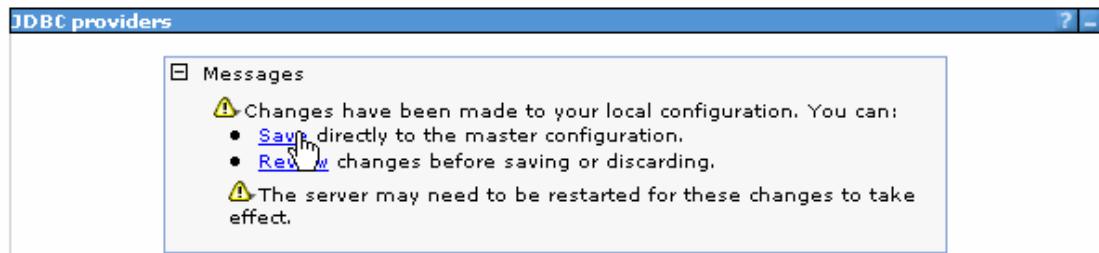
WebSphere software

Create a data source

Step 1: Enter basic data source information Step 2: Enter database specific properties for the data source Step 3: Setup security aliases → Step 4: Summary	Summary Summary of actions: <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">Options</th> <th style="text-align: left;">Values</th> </tr> </thead> <tbody> <tr> <td>Scope</td> <td>cells:localhostNode01Cell:nodes:n1Node01</td> </tr> <tr> <td>Data source name</td> <td>Oracle JDBC Driver DataSource</td> </tr> <tr> <td>JNDI name</td> <td>OracleDS</td> </tr> <tr> <td>Select an existing JDBC provider</td> <td>Oracle JDBC Driver</td> </tr> <tr> <td>Implementation class name</td> <td>oracle.jdbc.pool.OracleConnectionPoolDataSource</td> </tr> <tr> <td>URL</td> <td>jdbc:oracle:thin:@9.181.84.136:1521:ord</td> </tr> <tr> <td>Data store helper class name</td> <td>com.ibm.websphere.radapter.Oracle10gDataStoreHelper</td> </tr> <tr> <td>Use this data source in container managed persistence (CMP)</td> <td>true</td> </tr> <tr> <td>Component-managed authentication alias</td> <td>n1Node01/Alias_Oracle</td> </tr> <tr> <td>Mapping-configuration alias</td> <td>(none)</td> </tr> <tr> <td>Container-managed authentication alias</td> <td>(none)</td> </tr> </tbody> </table>	Options	Values	Scope	cells:localhostNode01Cell:nodes:n1Node01	Data source name	Oracle JDBC Driver DataSource	JNDI name	OracleDS	Select an existing JDBC provider	Oracle JDBC Driver	Implementation class name	oracle.jdbc.pool.OracleConnectionPoolDataSource	URL	jdbc:oracle:thin:@9.181.84.136:1521:ord	Data store helper class name	com.ibm.websphere.radapter.Oracle10gDataStoreHelper	Use this data source in container managed persistence (CMP)	true	Component-managed authentication alias	n1Node01/Alias_Oracle	Mapping-configuration alias	(none)	Container-managed authentication alias	(none)
Options	Values																								
Scope	cells:localhostNode01Cell:nodes:n1Node01																								
Data source name	Oracle JDBC Driver DataSource																								
JNDI name	OracleDS																								
Select an existing JDBC provider	Oracle JDBC Driver																								
Implementation class name	oracle.jdbc.pool.OracleConnectionPoolDataSource																								
URL	jdbc:oracle:thin:@9.181.84.136:1521:ord																								
Data store helper class name	com.ibm.websphere.radapter.Oracle10gDataStoreHelper																								
Use this data source in container managed persistence (CMP)	true																								
Component-managed authentication alias	n1Node01/Alias_Oracle																								
Mapping-configuration alias	(none)																								
Container-managed authentication alias	(none)																								

16. Click **Save** to save the changes.

Cell=localhostNode01Cell, Profile=AppSrv01



17. Select the check box corresponding to the data source you created in the previous step and click **Test connection**.

Cell=localhostNode01Cell, Profile=AppSrv01

JDBC providers

[JDBC providers](#) > [Oracle JDBC Driver](#) > Data sources

Use this page to edit the settings of a datasource that is associated with your selected JDBC provider. The datasource object supplies your application with connections for accessing the database. Learn more about this task in a [guided activity](#). A guided activity provides a list of task steps and more general information about the topic.

Preferences

New	Delete	Test connection	Manage state...
Select	Name ▾	JNDI name ▾	Scope ▾
Provider ▾	Description ▾	Category ▾	

You can administer the following resources:

<input checked="" type="checkbox"/>	Oracle JDBC Driver DataSource	OracleDS	Node=n1Node01	Oracle JDBC Driver	New JDBC Datasource	
-------------------------------------	---	----------	---------------	--------------------	---------------------	--

Total 1

The connection should succeed as shown in the following figure. If you experience problems while testing the connection, refer to the "Troubleshooting" section.

Cell=localhostNode01Cell, Profile=AppSrv01

JDBC providers

[JDBC providers](#) > [Oracle JDBC Driver](#) > Data sources

Use this page to edit the settings of a datasource that is associated with your selected JDBC provider. The datasource object supplies your application with connections for accessing the database. Learn more about this task in a [guided activity](#). A guided activity provides a list of task steps and more general information about the topic.

Preferences

Messages

The test connection operation for data source Oracle JDBC Driver DataSource on server server1 at node n1Node01 was successful.

[Information](#)

New	Delete	Test connection	Manage state...
Select	Name ▾	JNDI name ▾	Scope ▾
Provider ▾	Description ▾	Category ▾	

You can administer the following resources:

<input type="checkbox"/>	Oracle JDBC Driver DataSource	OracleDS	Node=n1Node01	Oracle JDBC Driver	New JDBC Datasource	
--------------------------	---	----------	---------------	--------------------	---------------------	--

Total 1

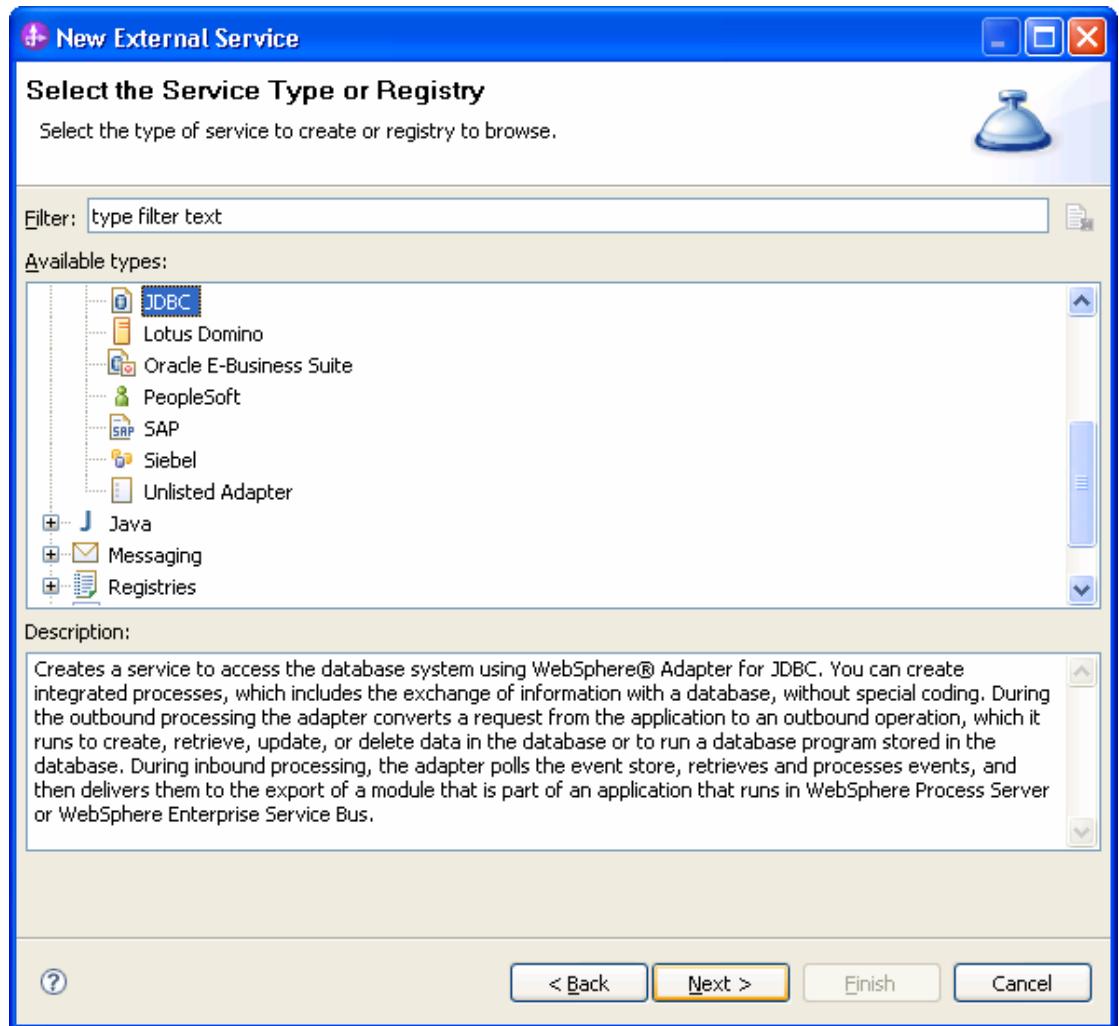
Note: The data source is created which will be used by the adapter to connect to the database.

Configure the adapter for outbound processing

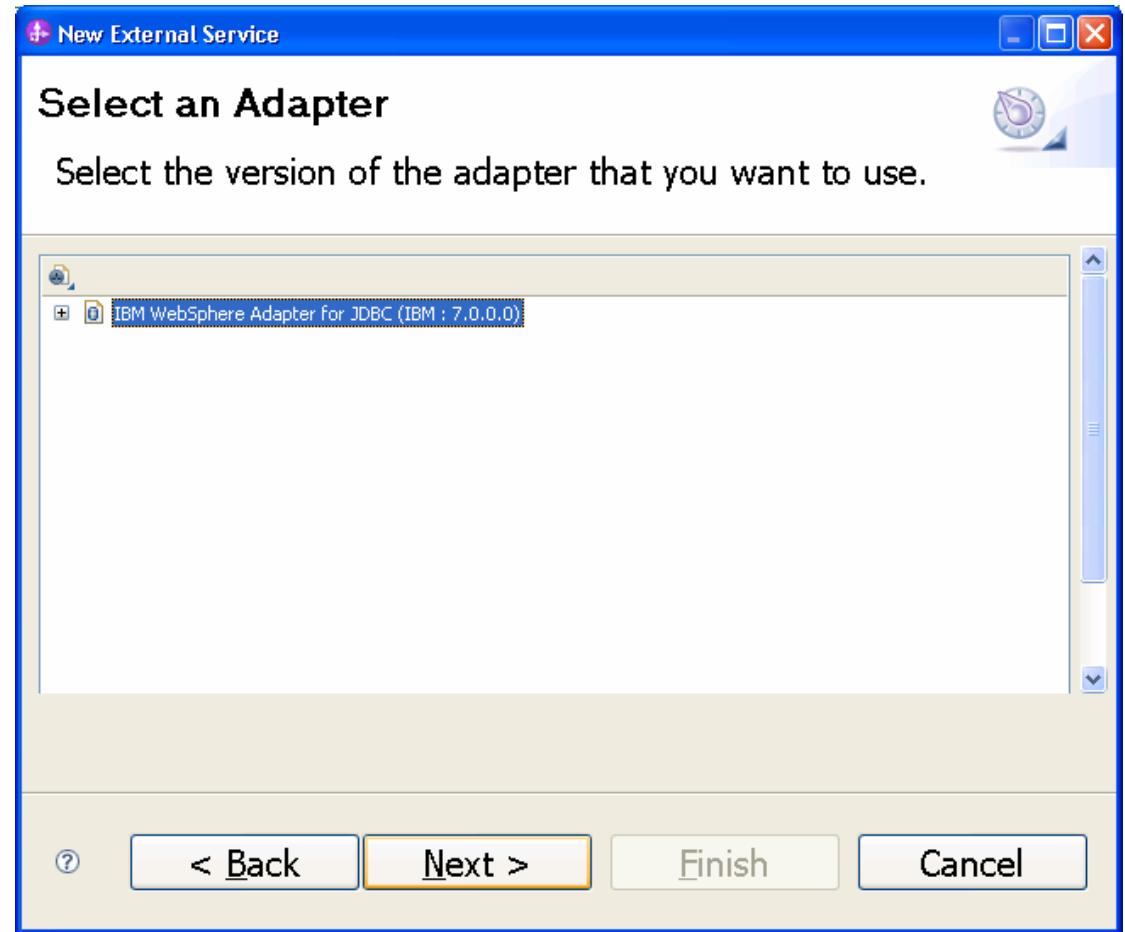
WebSphere software

Run the external service wizard to specify business objects, services, and configuration details.

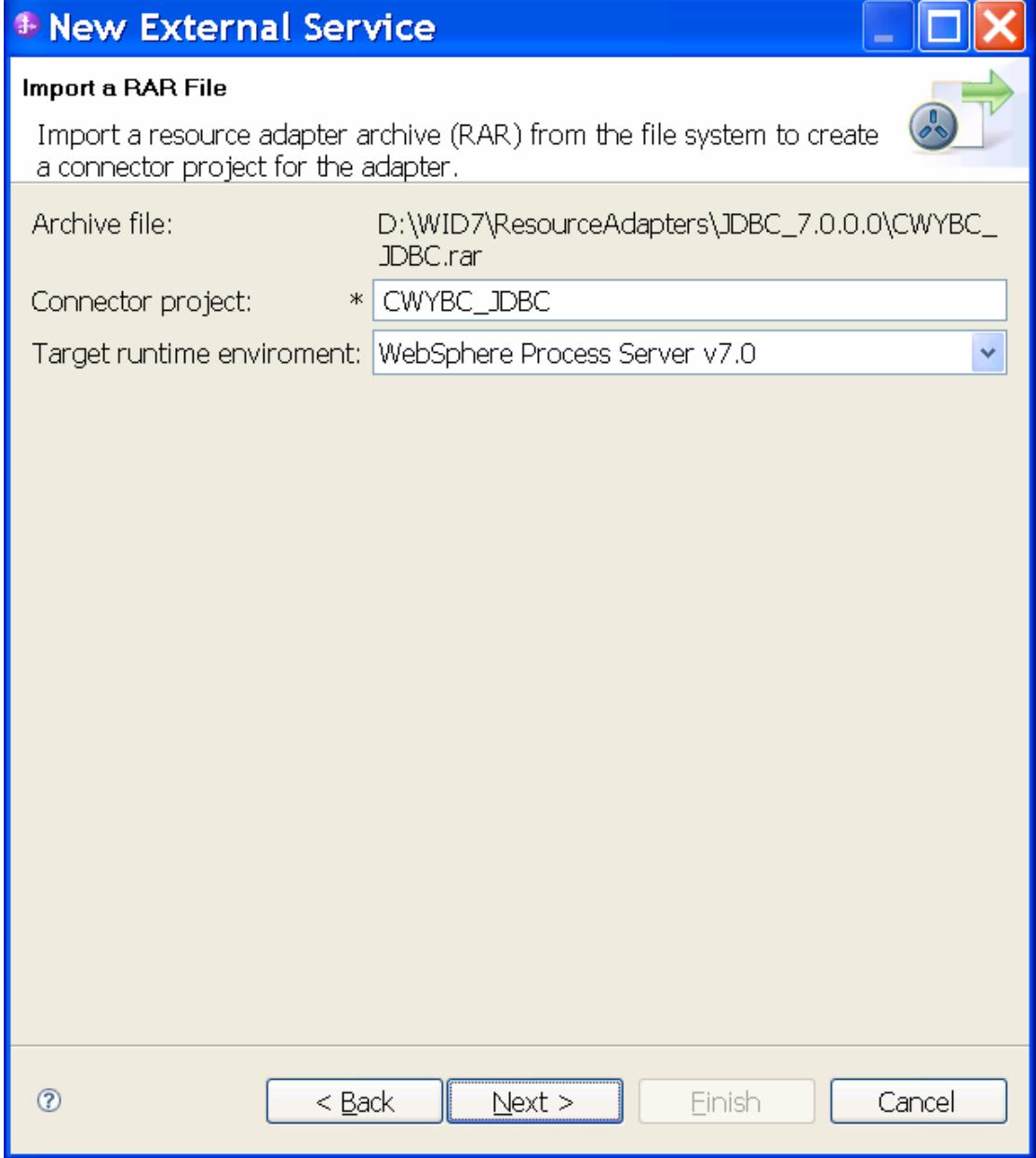
1. Switch to the Business Integration Perspective in WebSphere Integration Developer by selecting **Window -> Open Perspective Business Integration**.
2. Start the external service wizard by selecting **File-> New -> External Service**.
3. In the **Available Types** area, select **Adapters > JDBC** and then click **Next**.



4. Select the **IBM WebSphere Adapter for JDBC (IBM: 7.0.0.0)** and click **Next**.



5. In the **Connector project** field enter **CWYBC_JDBC**, and in the **Target runtime environment** field, select the appropriate runtime. Click **Next**.



New External Service

Import a RAR File

Import a resource adapter archive (RAR) from the file system to create a connector project for the adapter.



Archive file: D:\WID7\ResourceAdapters\JDBC_7.0.0.0\CWYBC_JDBC.rar

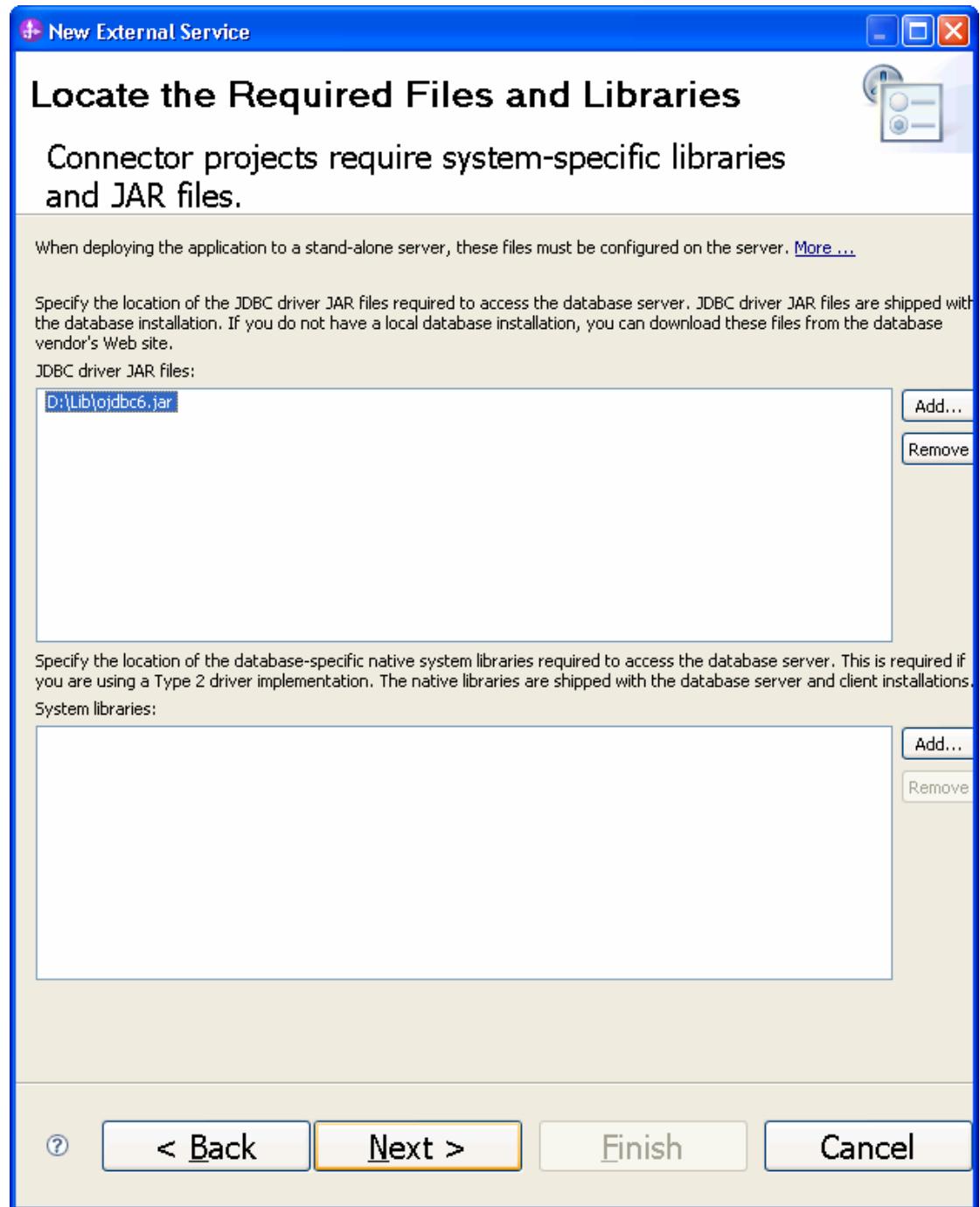
Connector project: * CWYBC_JDBC

Target runtime environment: WebSphere Process Server v7.0

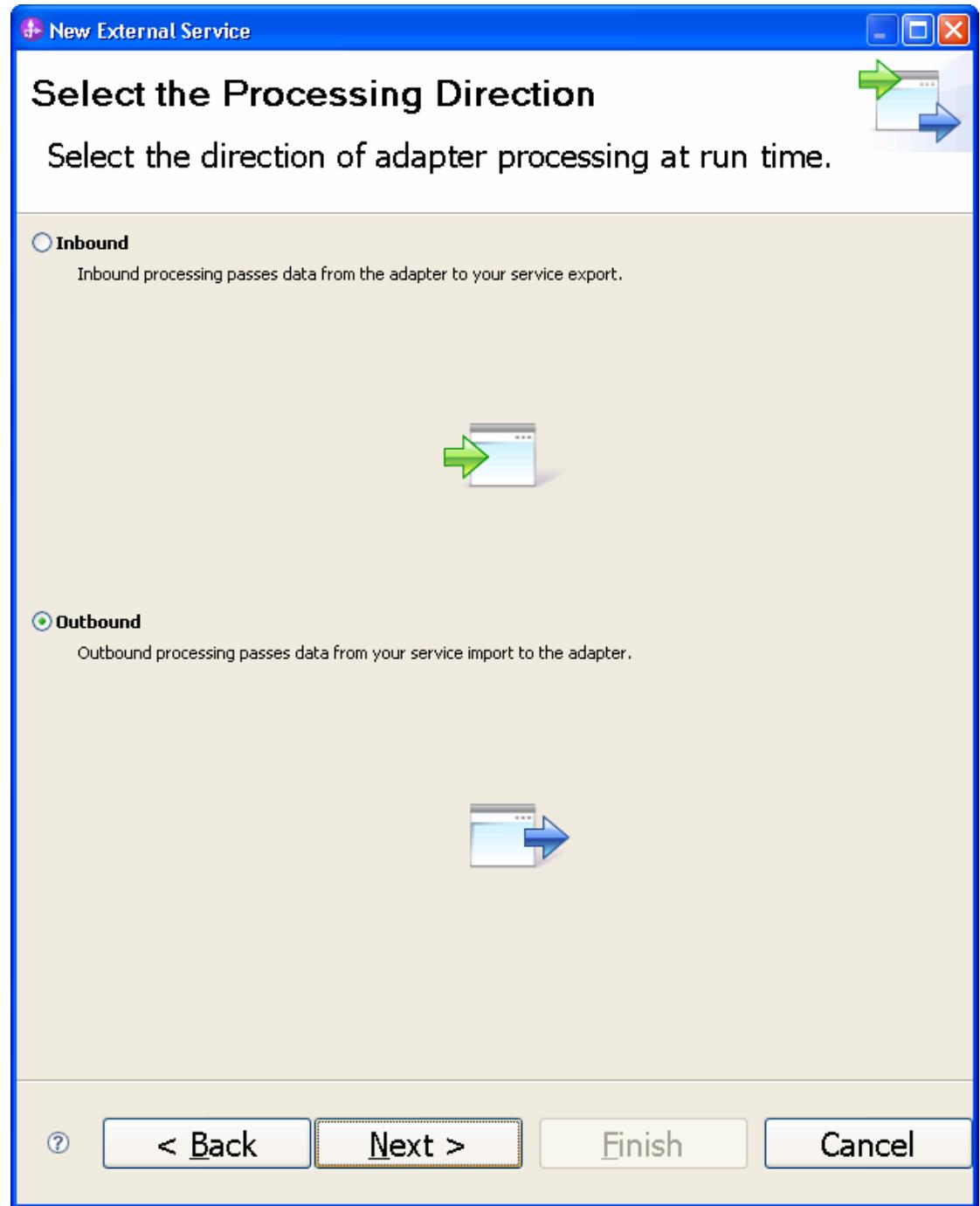
[?](#)[< Back](#)[Next >](#)[Finish](#)[Cancel](#)

WebSphere software

6. In the **JDBC driver JAR files** field, click **Add**, to add the JDBC driver class to connect to the database. Browse to select the driver JAR file and click **Next**.



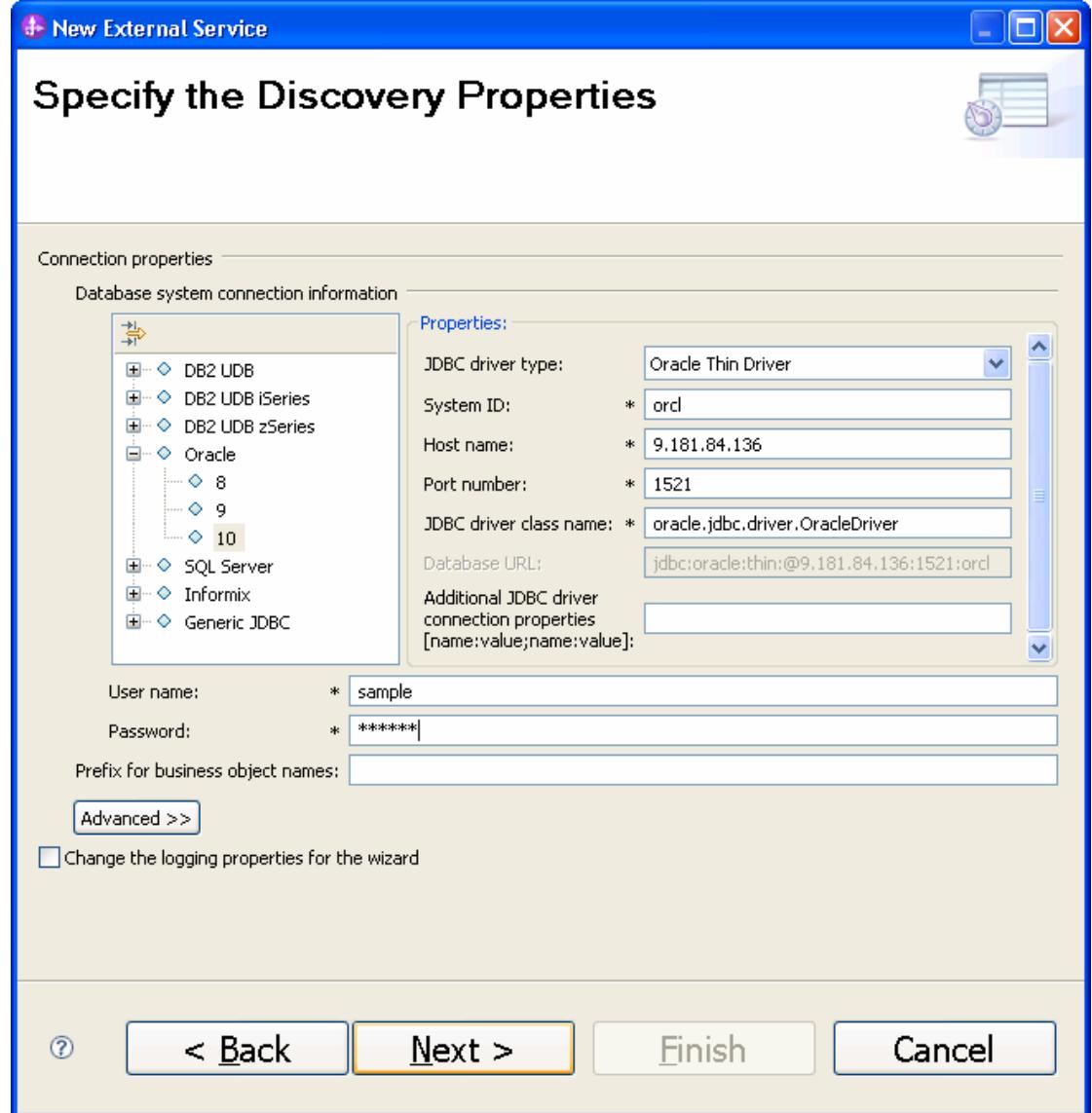
7. Select **Outbound** and click **Next**.



Set connection properties for the external service wizard

To connect to the Oracle database:

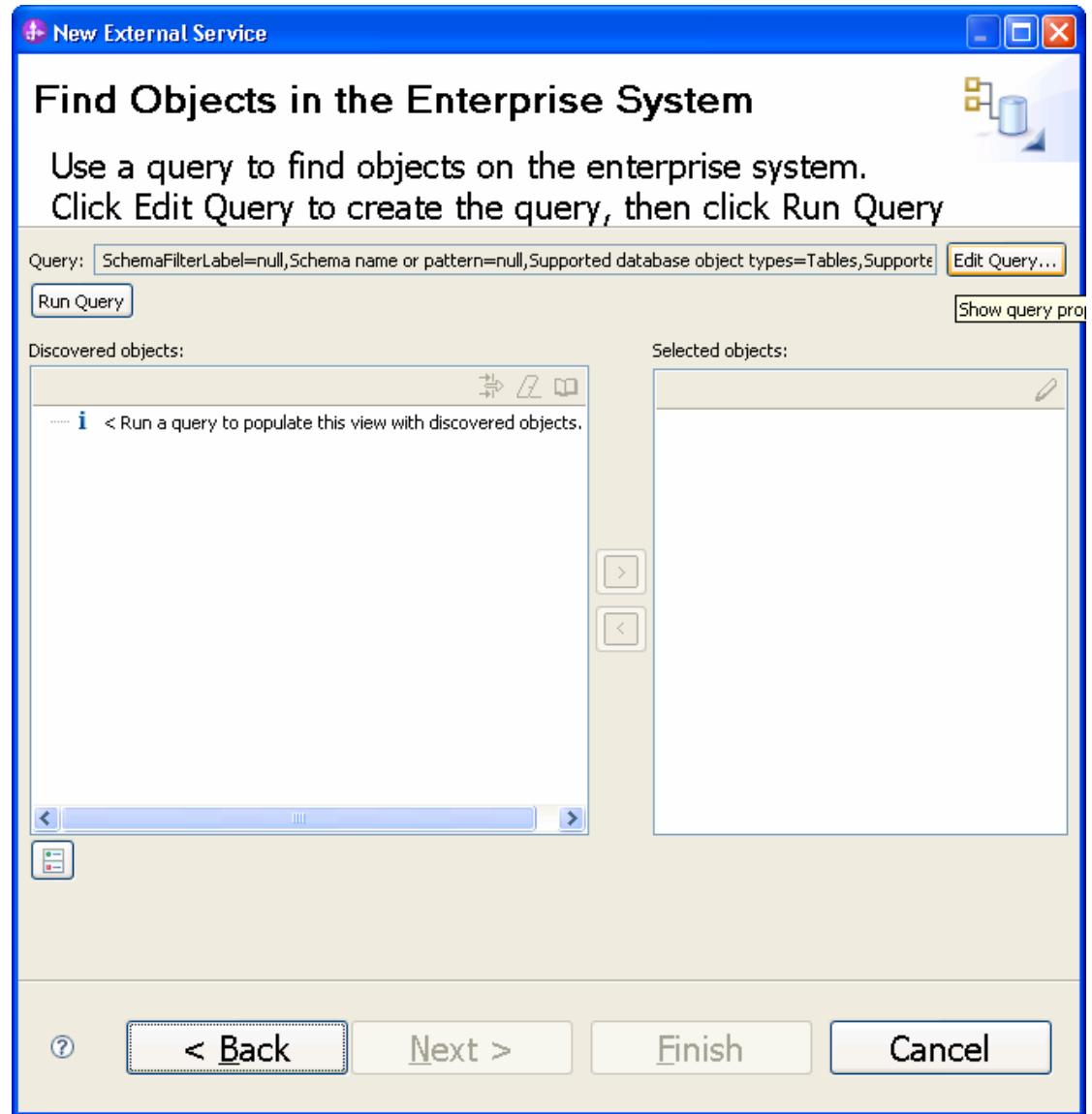
1. Expand the **Oracle** node from **Database system connection information** then select **10**.
2. Enter **System ID**, **Host name**, **Port number**, **User name** and **Password** fields, and then click **Next**.



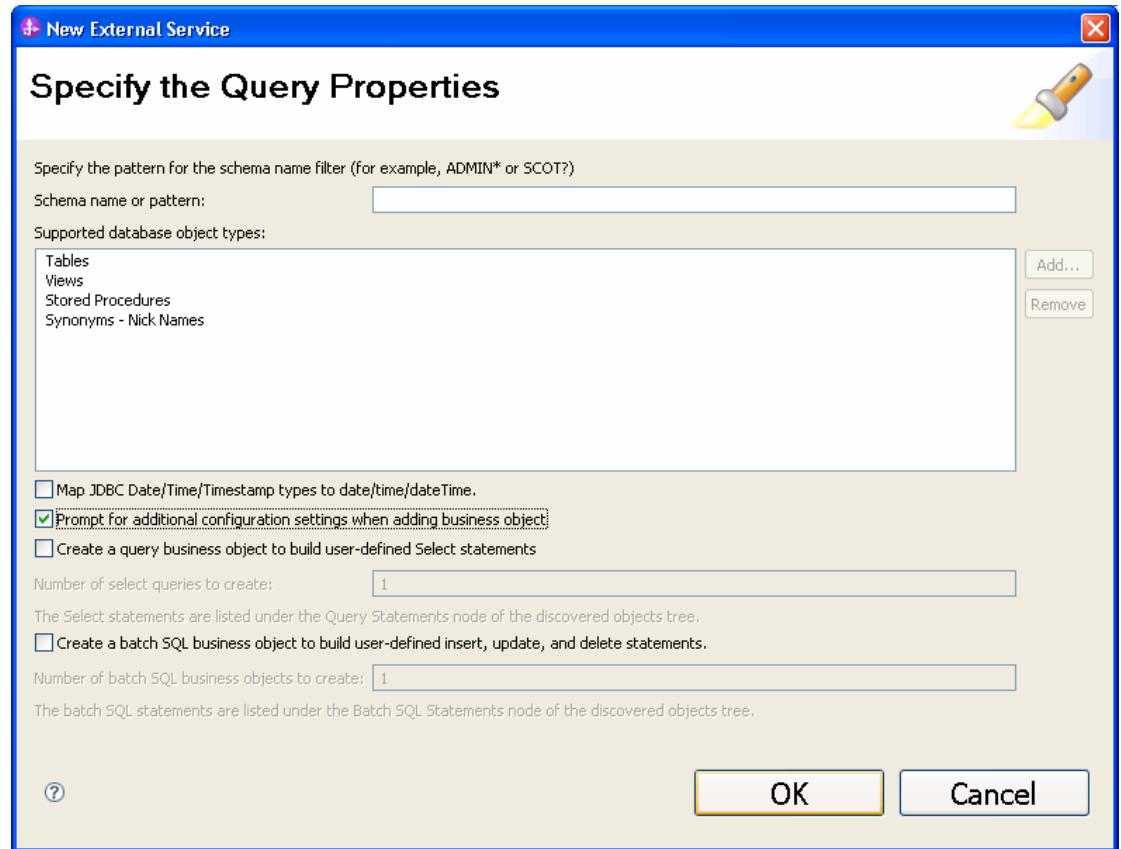
Select the business objects to be used with the adapter

Follow these steps to select the **Customer** and **CustAdd** business objects:

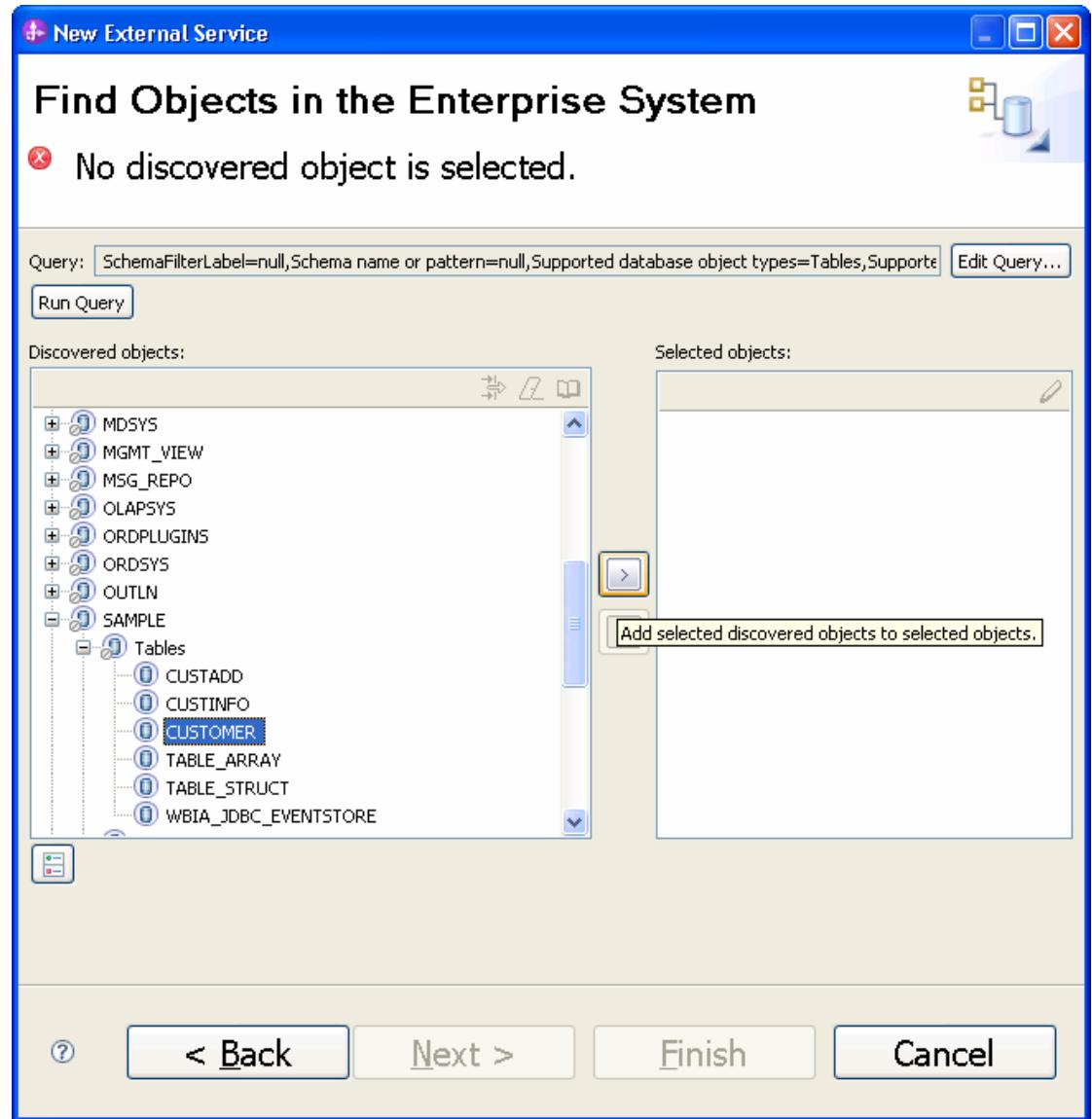
1. In the Find Objects in Enterprise System window, click **Edit Query**.



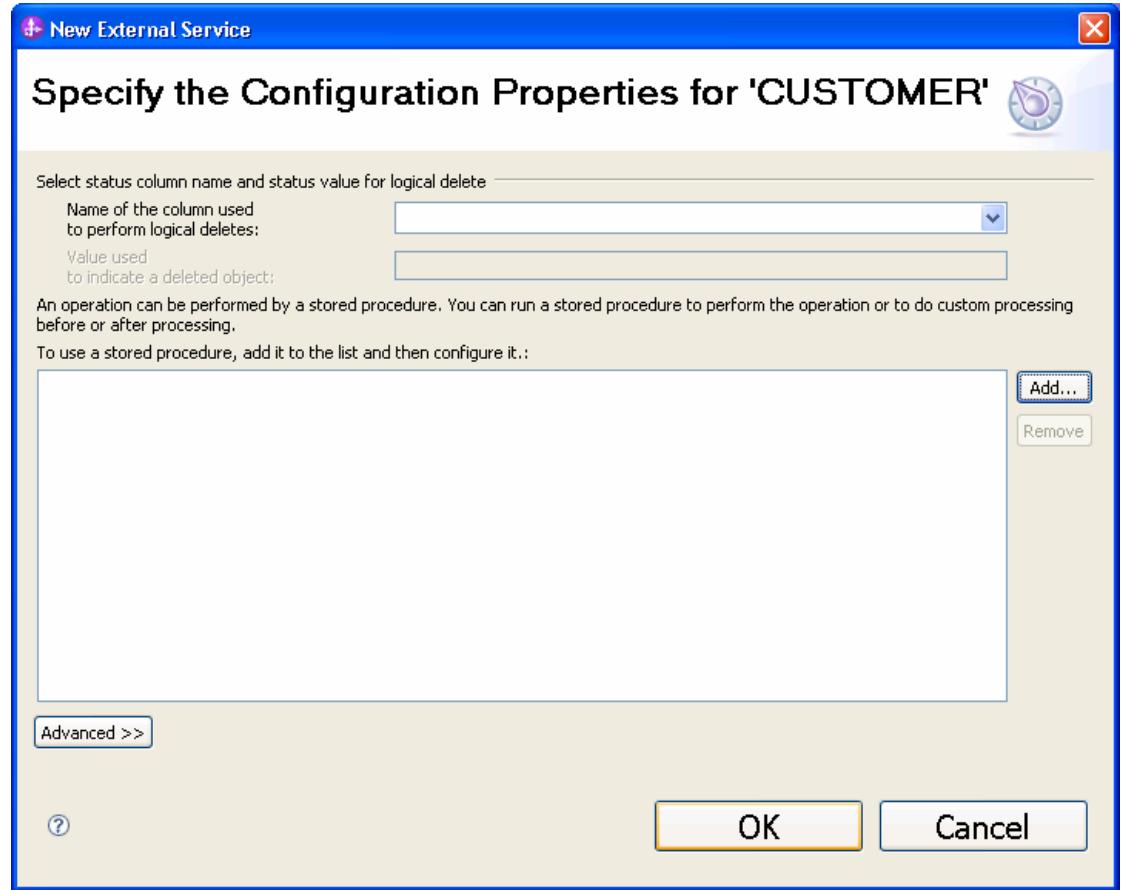
2. In the Specify the Query Properties window, select the **Prompt for additional configuration settings when adding business objects** check box and click **OK**.



3. Click **Run Query**.
4. Expand the **SAMPLE** (for this tutorial only) node, select **Tables** and expand it.



5. Select the CUSTOMER table and click . In the Specify the Configuration Properties for 'CUSTOMER' window, click **OK**.



6. Select the CUSTADD table and click .
7. In the Specify the Configuration Properties for 'CUSTADD' window, select **CUSTOMER (SAMPLE)** from the **Choose parent table** list, and then select **PKEY** for **CUSTID** in the Build a foreign key area. Select the **Parent object owns child object(cascade delete)** check box. Click **Add**.

New External Service

Specify the Configuration Properties for 'CUSTADD'

Select status column name and status value for logical delete

Name of the column used to perform logical deletes:

Value used to indicate a deleted object:

Choose parent table from the list for the selected child

Choose parent table: CUSTOMER (SAMPLE)

Single cardinality

Build a foreign key relationship by selecting a parent table column for each child column

ADDRID:	NONE
CUSTID:	PKEY
CITY:	NONE
ZIPCODE:	NONE

Parent object owns child object (cascade delete)

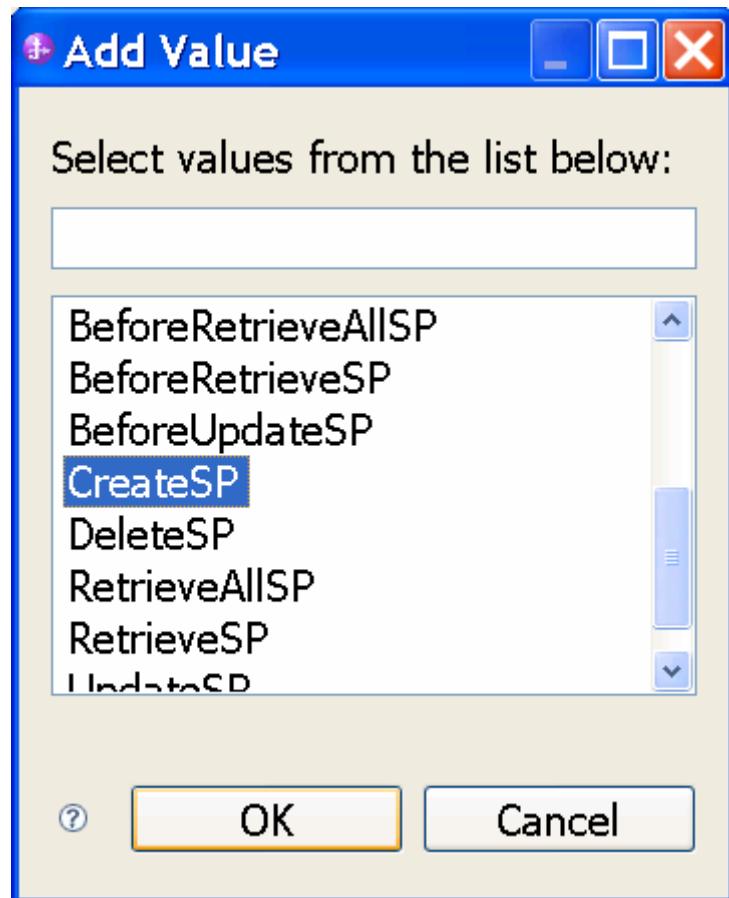
Preserves CUSTADD when the parent is updated.

CUSTADD required for operations on parent

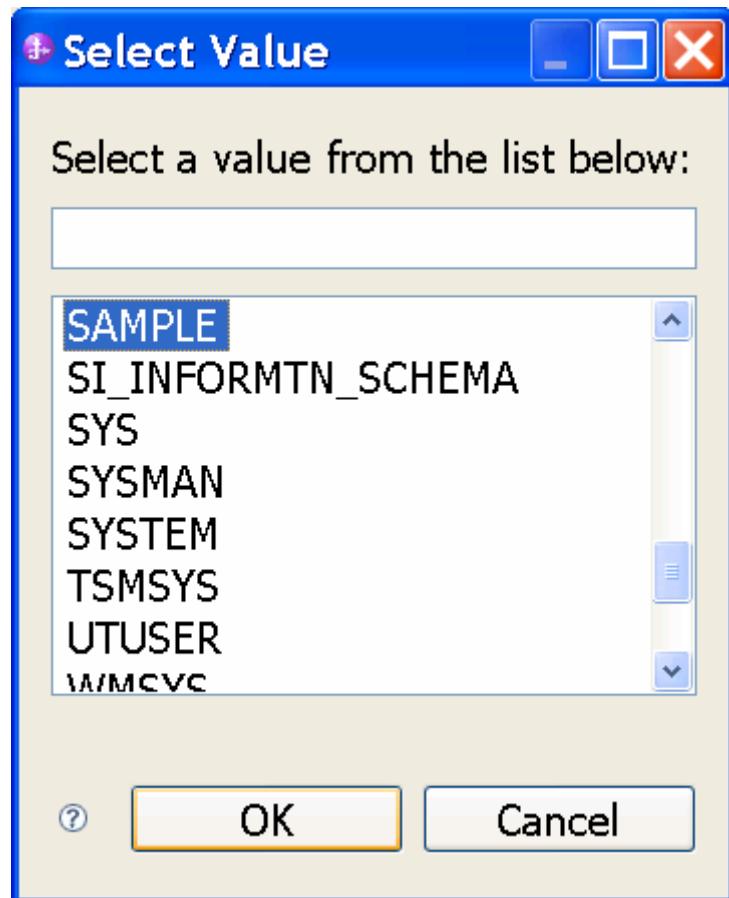
An operation can be performed by a stored procedure. You can run a stored procedure to perform the operation or to do custom processing before or after processing.

To use a stored procedure, add it to the list and then configure it.:

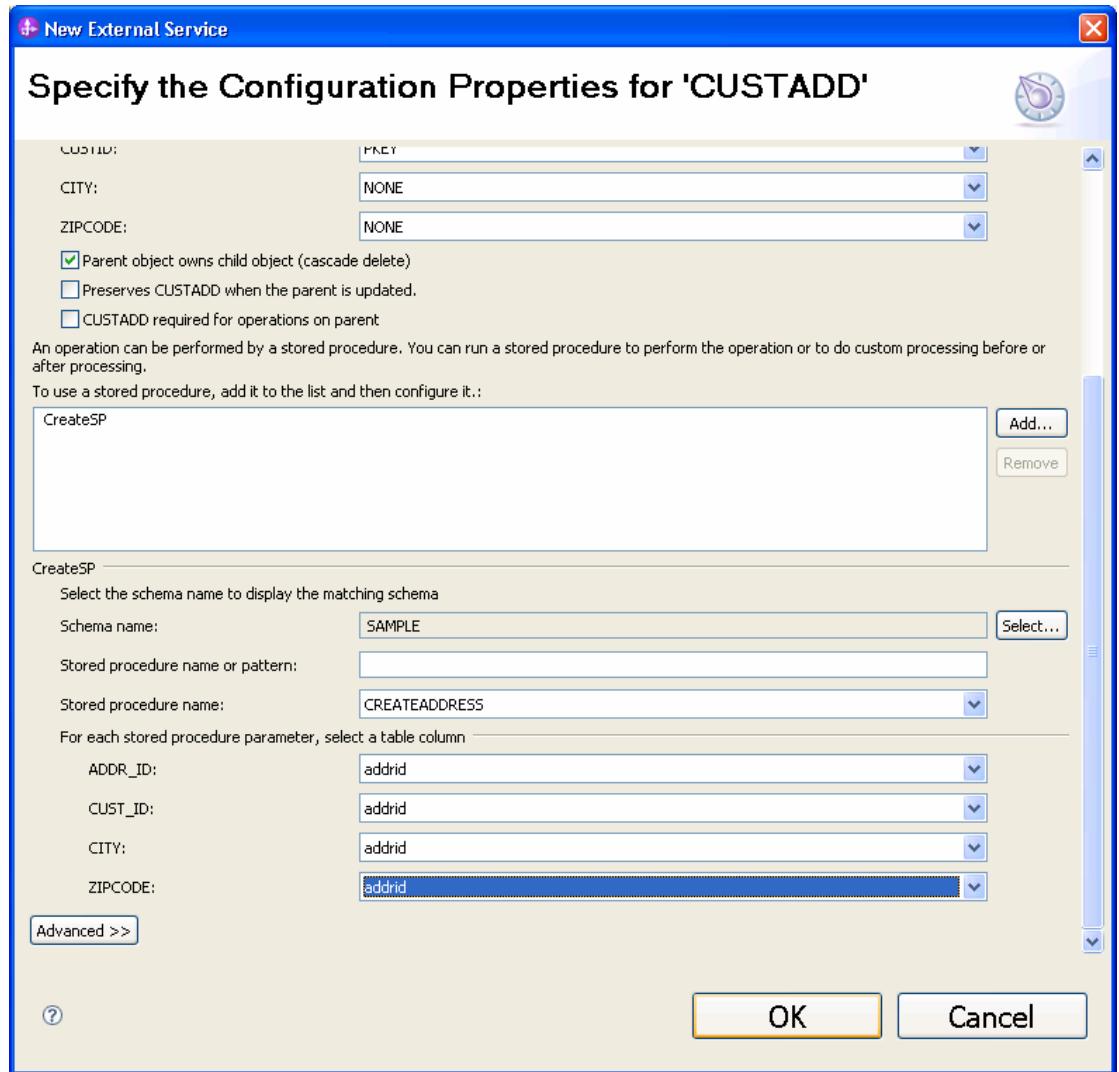
8. Select **CreateSP** and click **OK**.



9. Select **SAMPLE** for schema name.



10. Select **CREATEADDRESS** form the stored procedure name list.



11. Select stored procedure parameter for each column.

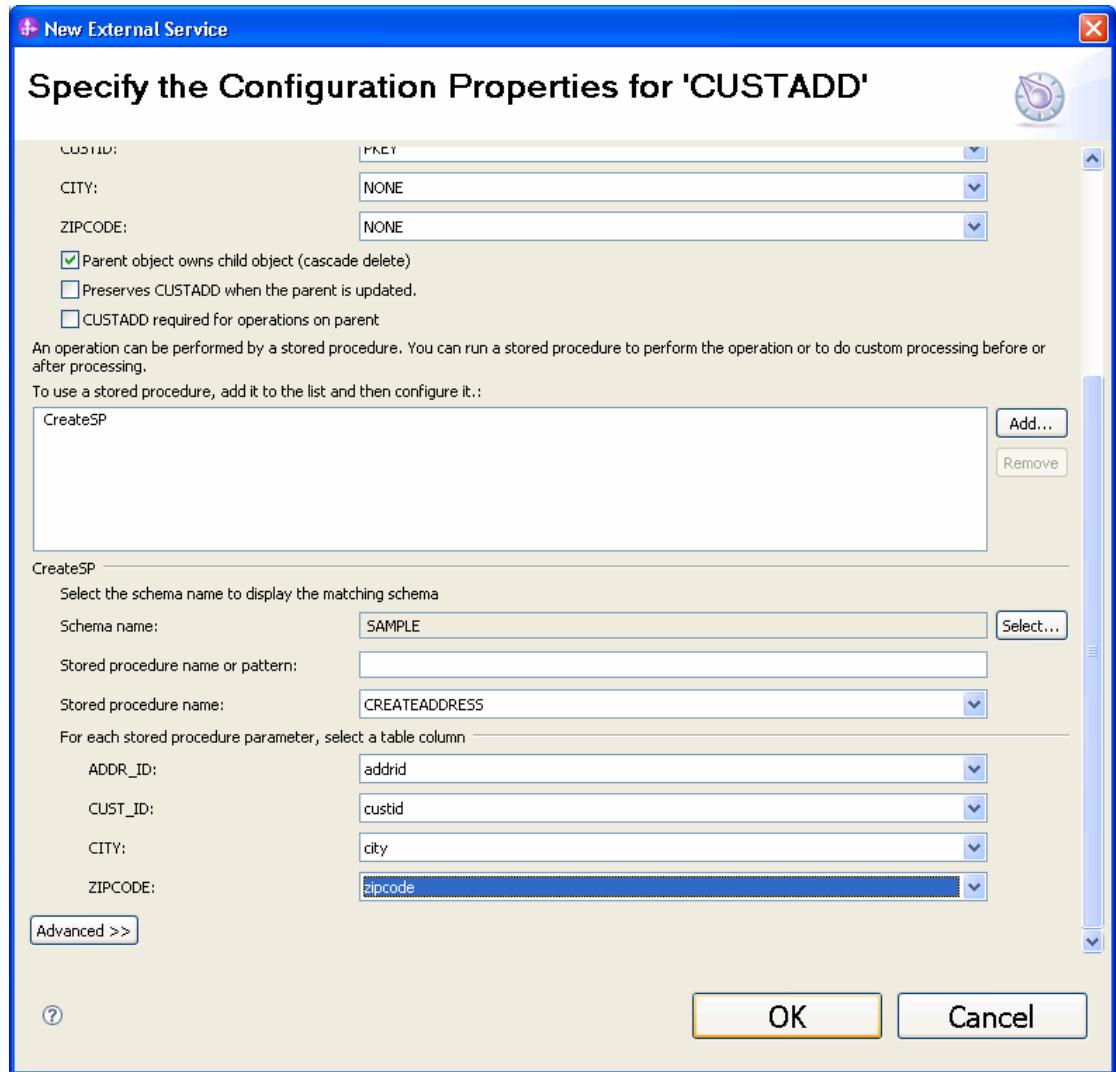
ADDR_ID: addrid

CUST_ID: custid

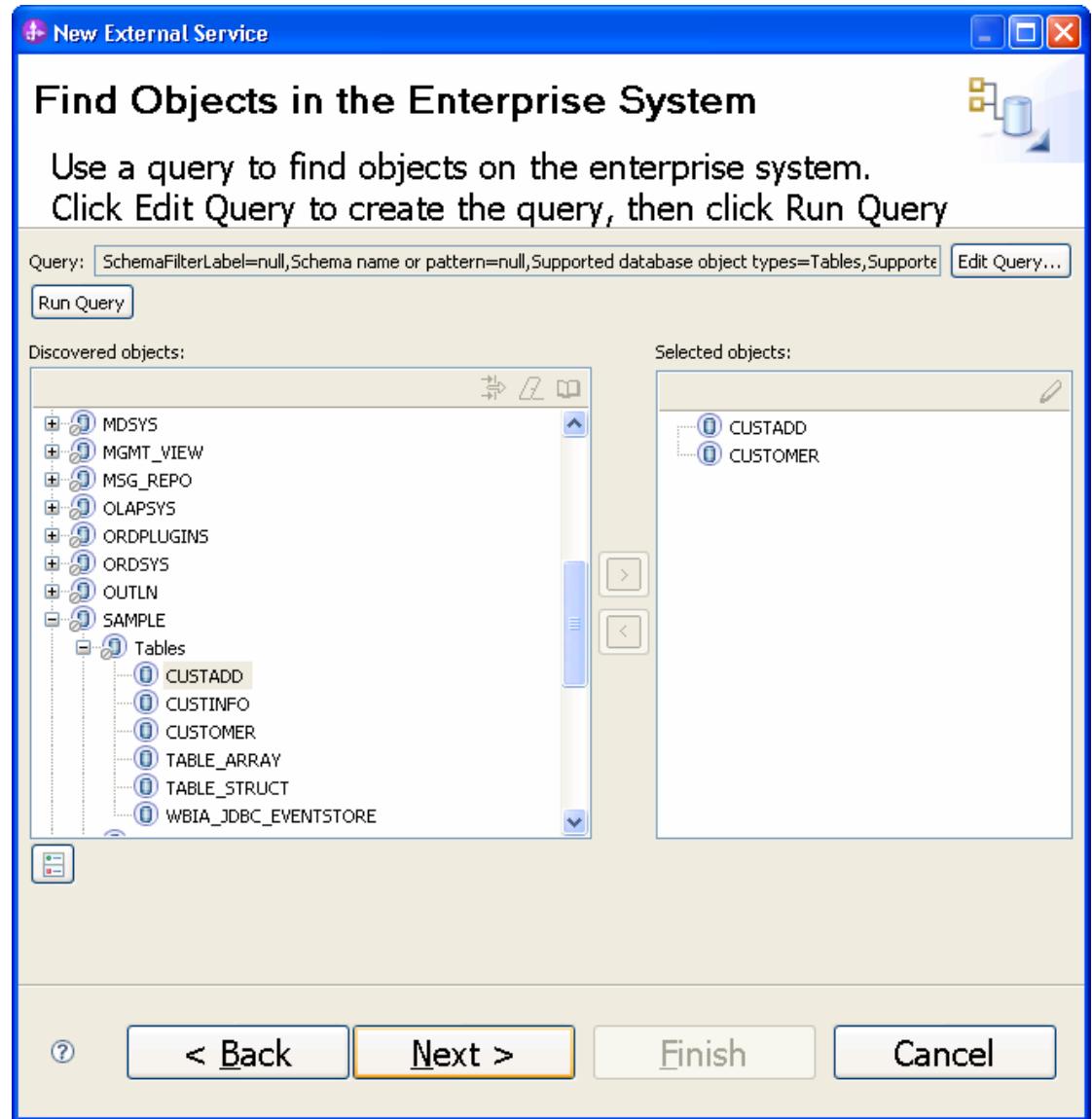
CITY: city

ZIPCODE: zipcode

12. Click **OK**.



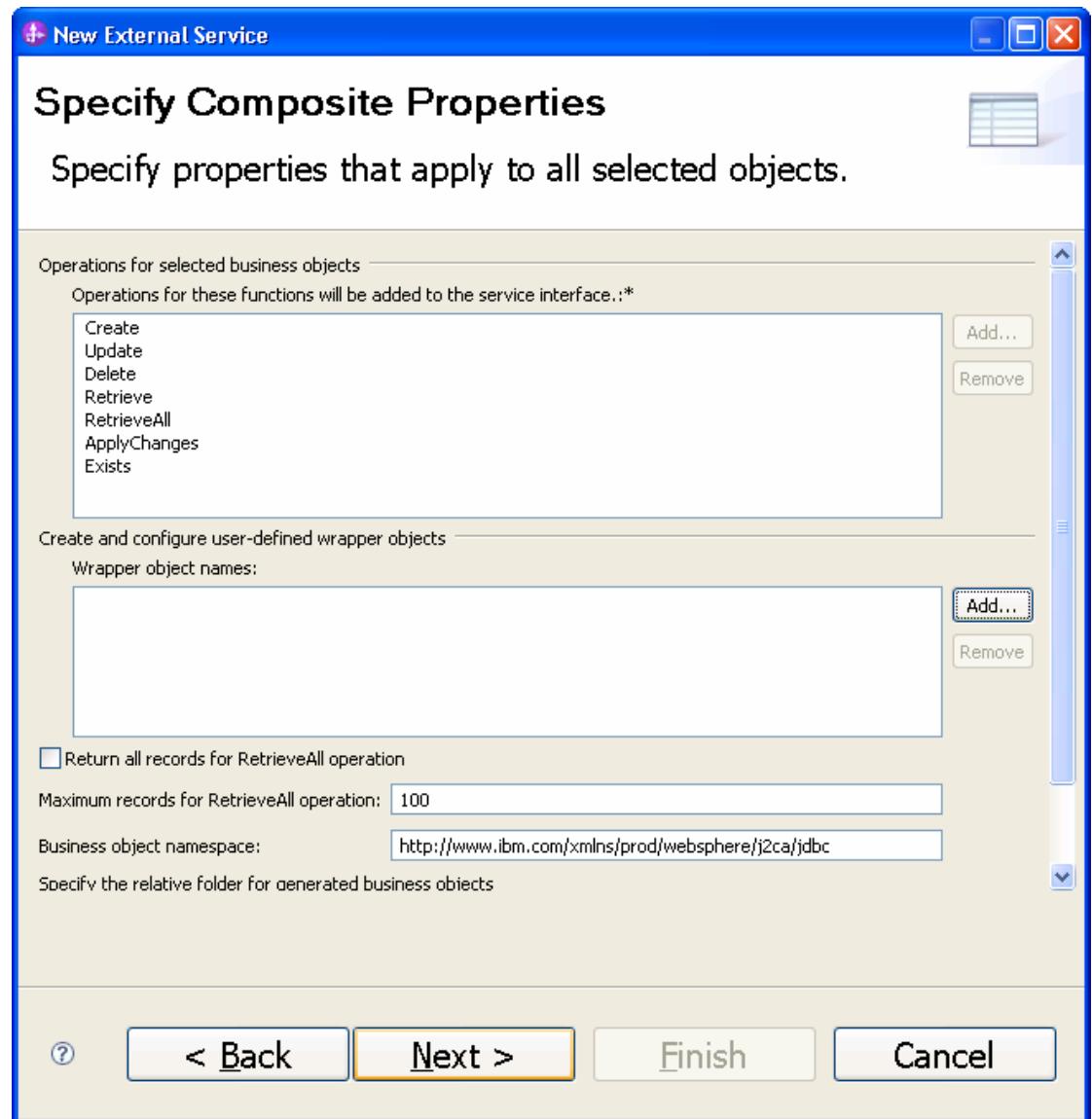
13. In the Find Objects in Enterprise System window, click **Next**.



Generating business object definitions and related artifacts

Follow these steps to generate the business object definitions.

1. In the Specify Composite Properties window, accept the default values for all fields and click **Next**.



2. In the Specify the Service Generation and Deployment window, perform the following steps:
 - a) Select **Other** for security options under **Deployment Properties**.
 - b) Clear the **Join the global transaction** check box.
 - c) Select **Specify predefined connection pool DataSource** from the **Database connection information** list.
 - d) Enter **OracleDS** in the **Connection pool DataSource JNDI Name** field, and click **Next**.

New External Service

Specify the Service Generation and Deployment



Specify properties for generating the service and running it on the server.

Service Operations

To modify the names, or add a description to the operations to be generated in the interface file, click [Edit Operations...](#)

Deployment Properties

How do you want to specify the security credentials?

Using an existing JAAS alias (recommended)
A Java Authentication and Authorization Services (JAAS) alias is the preferred method.
J2C authentication data entry:

Using security properties from the managed connection factory
The properties will be stored as plain text; no encryption is used.
User name:
Password:

Other
Use if no security is required or will be handled by the EIS system, or the RAR will be deployed on the server and security will be specified by the properties in the JNDI lookup name.

The quality of service that is used to join the transaction provides a higher degree of data integrity, especially when a failure occurs. To participate in a global transaction, a predefined XA DataSource or XA database connection information must be specified in the connection properties. [More ...](#)

Join the global transaction

Deploy connector project:

Specify the settings used to connect to JDBC at run time:

Connection settings:

Connection Properties

To join a global transaction, specify a predefined XA datasource or XA database connection information. When not joining a global transaction, either the XA connection information or the local connection information can be specified.

Database connection information:

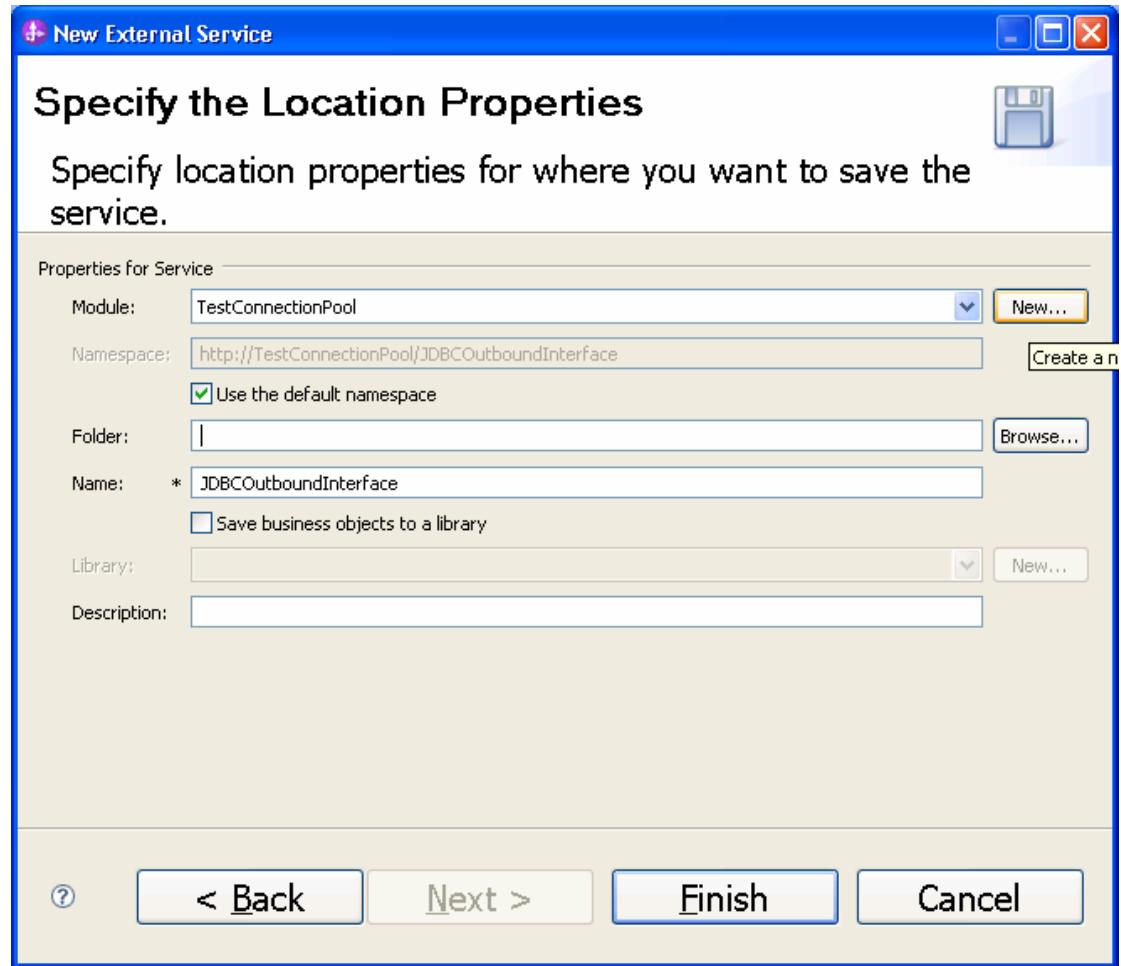
Database system connection information

Database vendor:
Connection pool DataSource JNDI name:

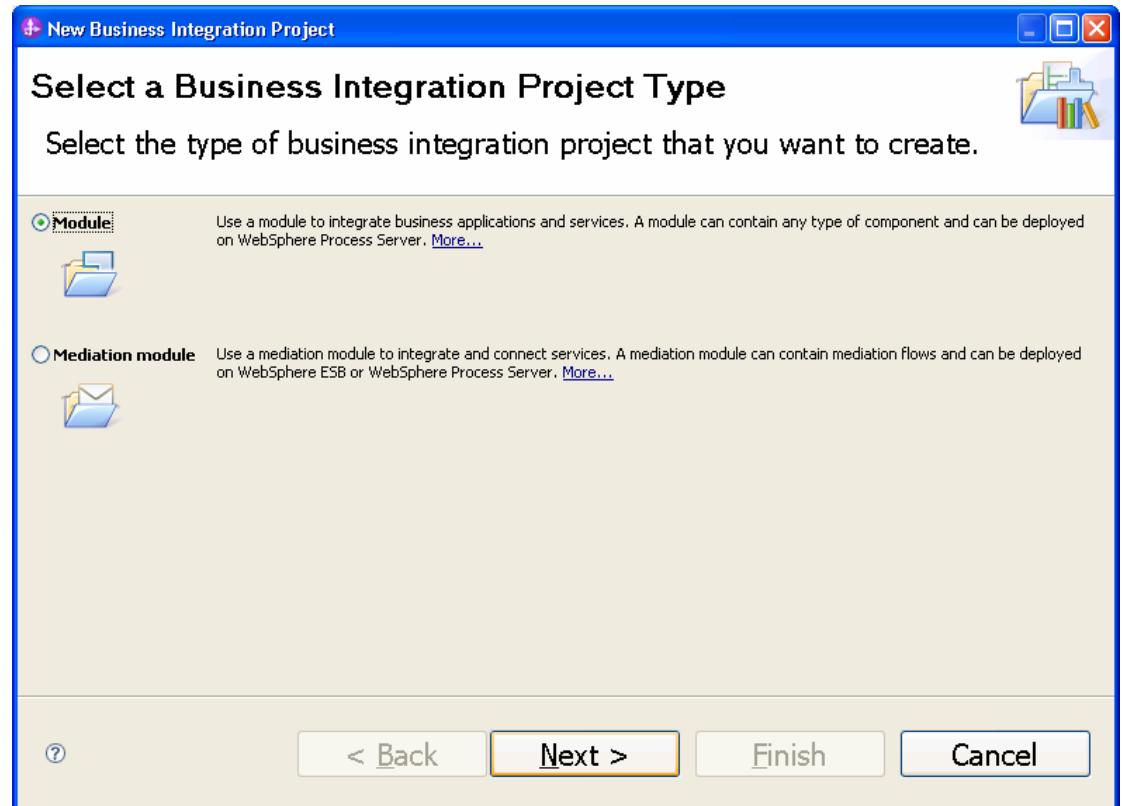
[Advanced >>](#)

[?](#) [**< Back**](#) [**Next >**](#) [Finish](#) [Cancel](#)

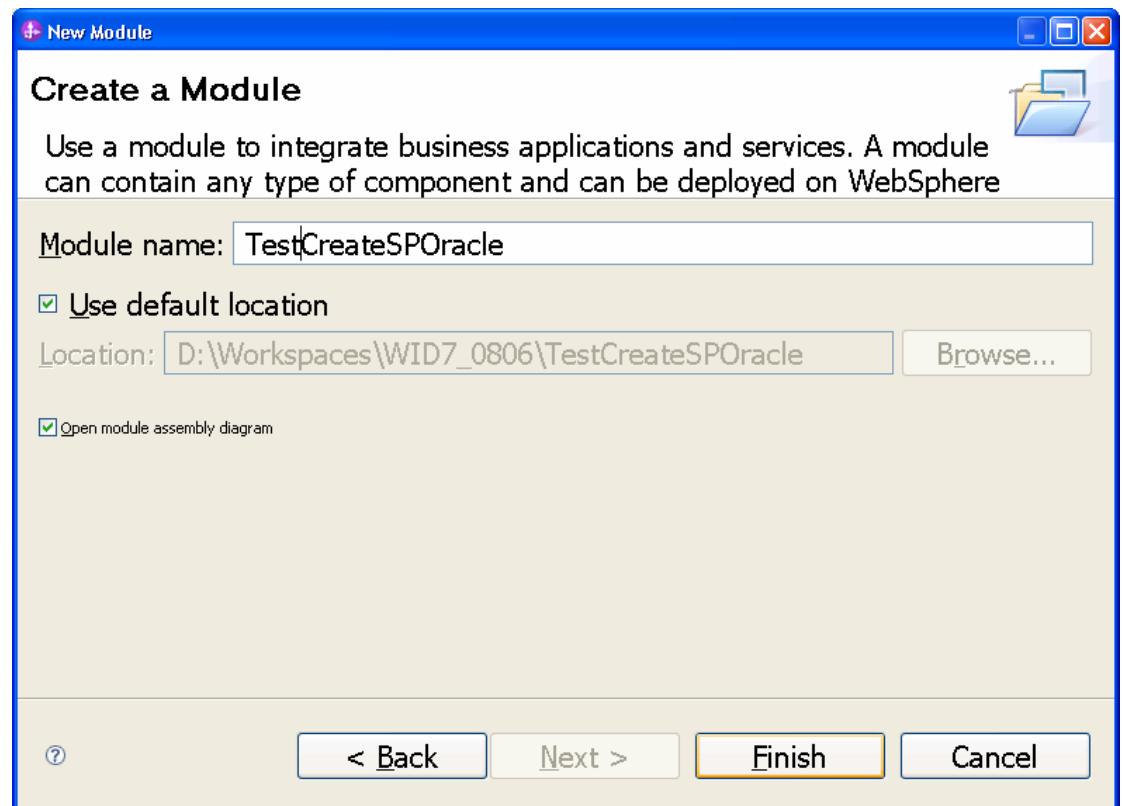
- Click **New** in the Specify the Location Properties window.



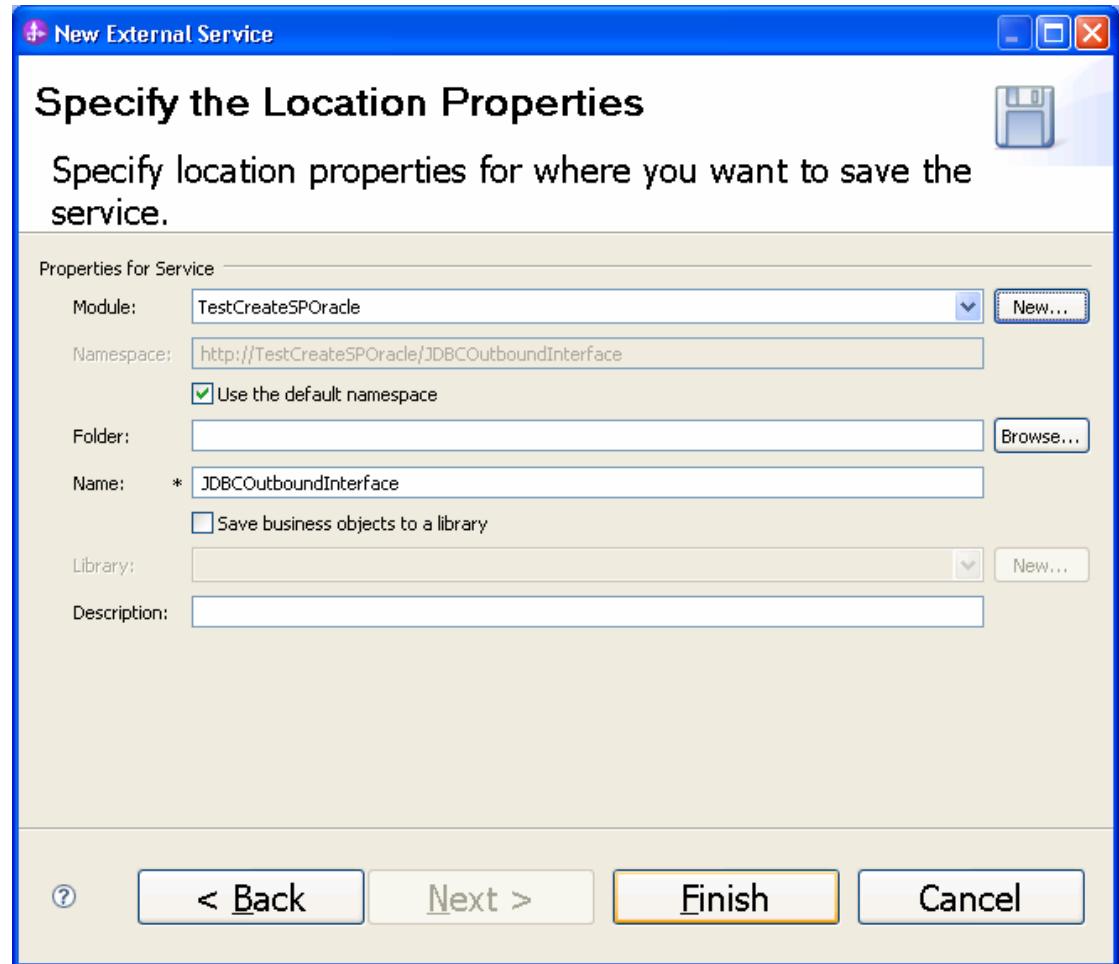
4. In the Select a Business Integration Project Type window, select **Module** and click **Next**.



5. In the Create a Module window, type **TestCreateSPOracle** in the **Module Name** field and click **Finish**.



6. Click **Finish** to complete service creation.



7. Expand the created Business Integration Project and verify whether the artifacts are generated correctly.

Projects [New...](#) □

- + CommonThirdParty
- + CWYBC_JDBC_Local
- + CWYBC_JDBC_XA
- TestCreateSPOracle
 - Assembly Diagram
 - JDBCOutboundInterface
 - Dependencies
 - Integration Logic
 - Data Types
 - ExistsResult
 - IntegrityConstraintFault
 - MatchesExceededLimitFault
 - MissingDataFault
 - MultipleMatchingRecordsFault
 - ObjectNotFoundFault
 - PrimaryKeyValuePair
 - RecordNotFoundFault
 - SampleCustadd
 - SampleCustaddBG
 - SampleCustaddContainer
 - SampleCustomer
 - SampleCustomerBG
 - SampleCustomerContainer
 - UniqueConstraintFault
 - WBIFault
 - Interfaces
 - JDBCOutboundInterface
 - Transformations
- + TestInboundWrapper

Deploy the module to the test environment

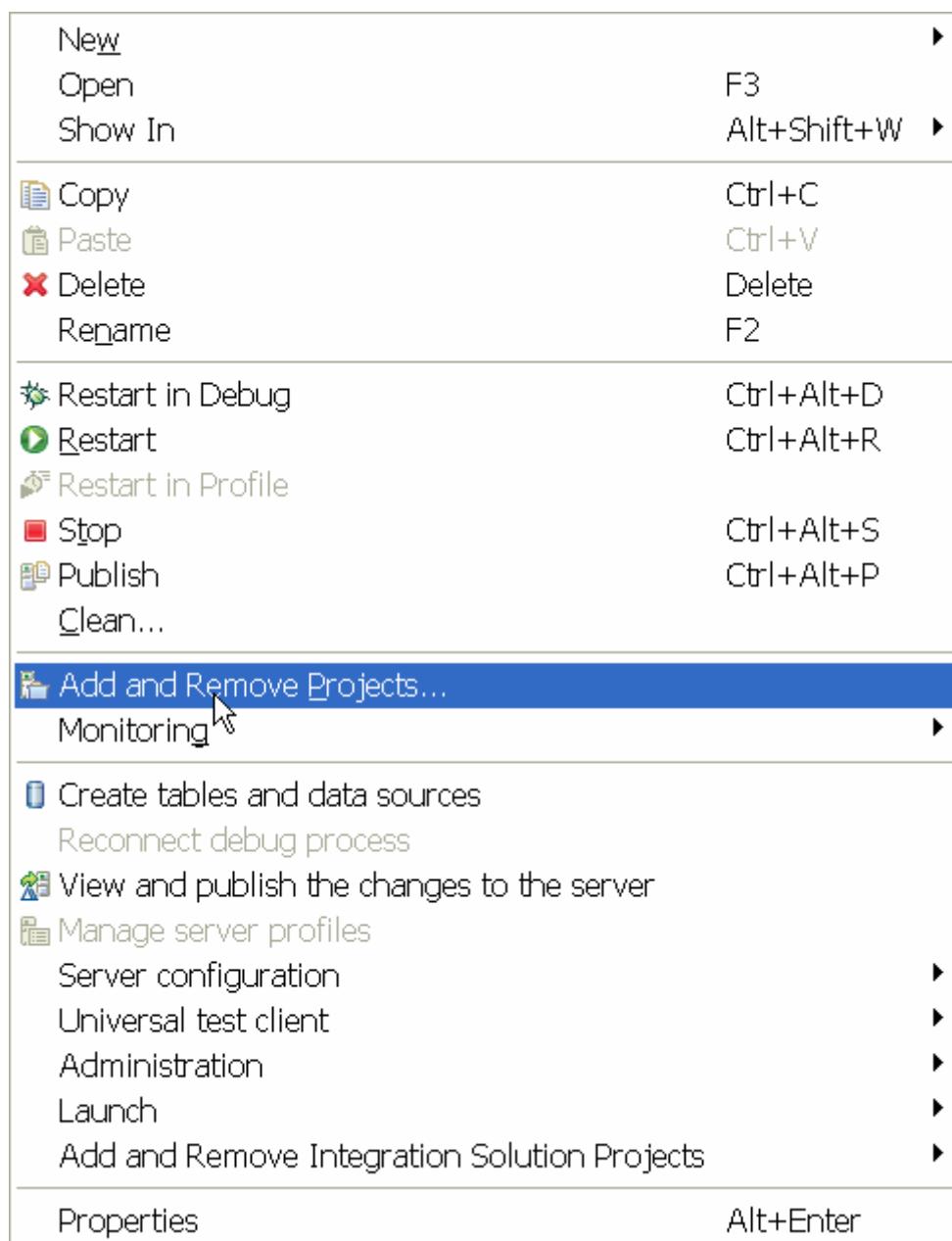
After running the external service wizard, you will have an SCA module that contains an Enterprise Information System import. You must install this SCA module in the WebSphere Integration Developer integration test client. To do this, you must add the SCA module you

WebSphere software

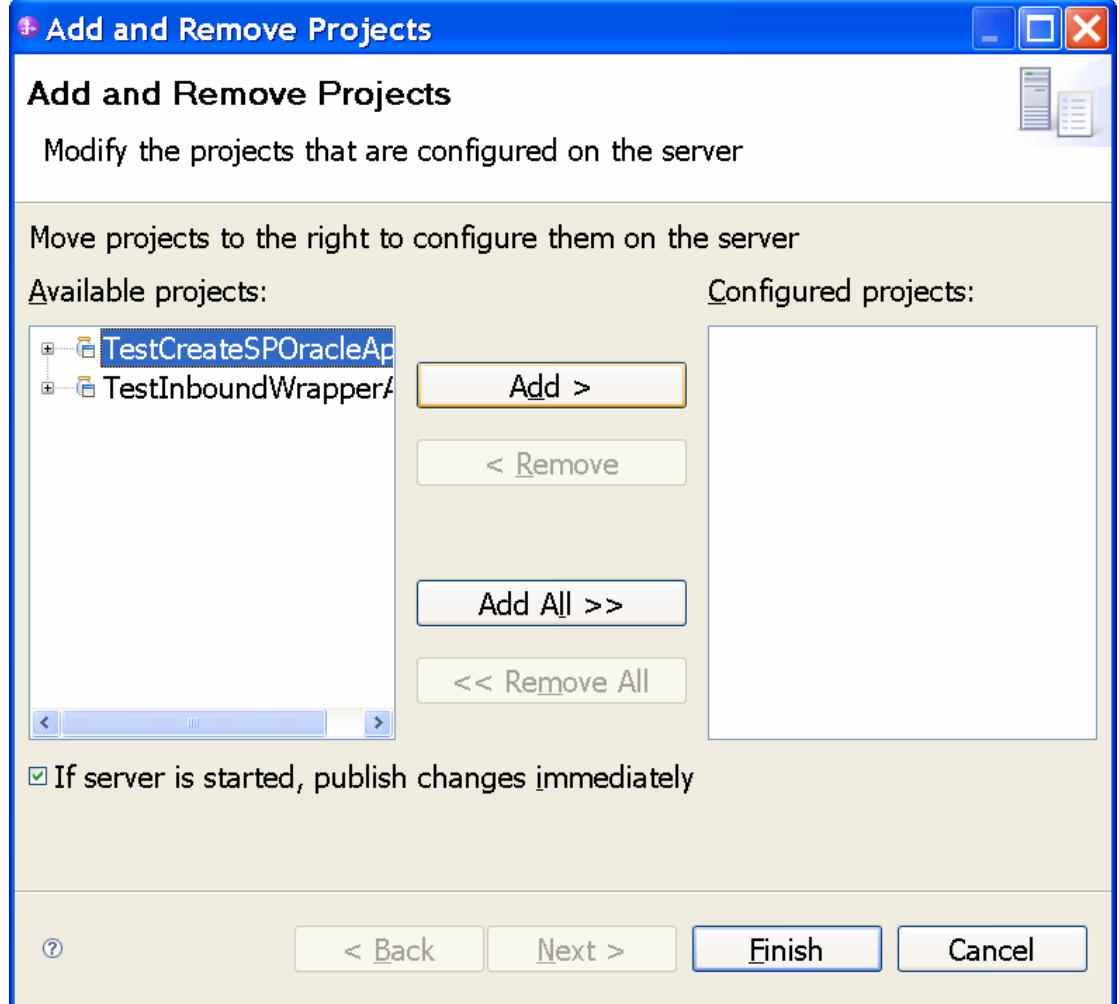
created earlier to the server using the **Servers** view in WebSphere Integration Developer.

Steps for adding the SCA module to the server:

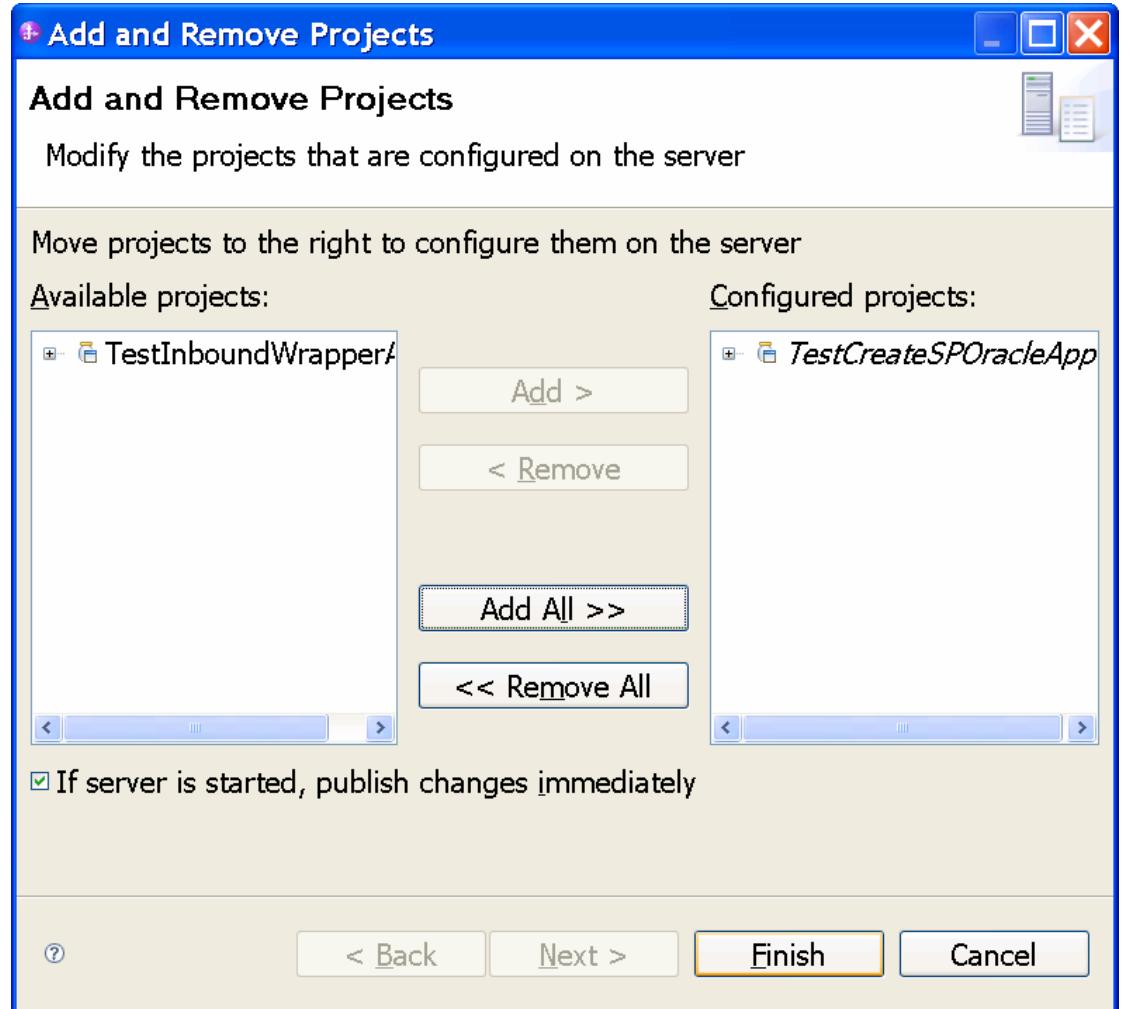
1. In WebSphere Integration Developer, switch to the **Servers** view by selecting **Windows > Show View > Servers**.
2. In the Servers tab in the lower-right pane of the WebSphere Integration Developer screen, right-click the server, and select **Start**.
3. After the server is started, right-click the server, and select **Add and Remove projects**.



The Add and Remove Projects window lists the available projects in the WebSphere Integration Developer workspace.



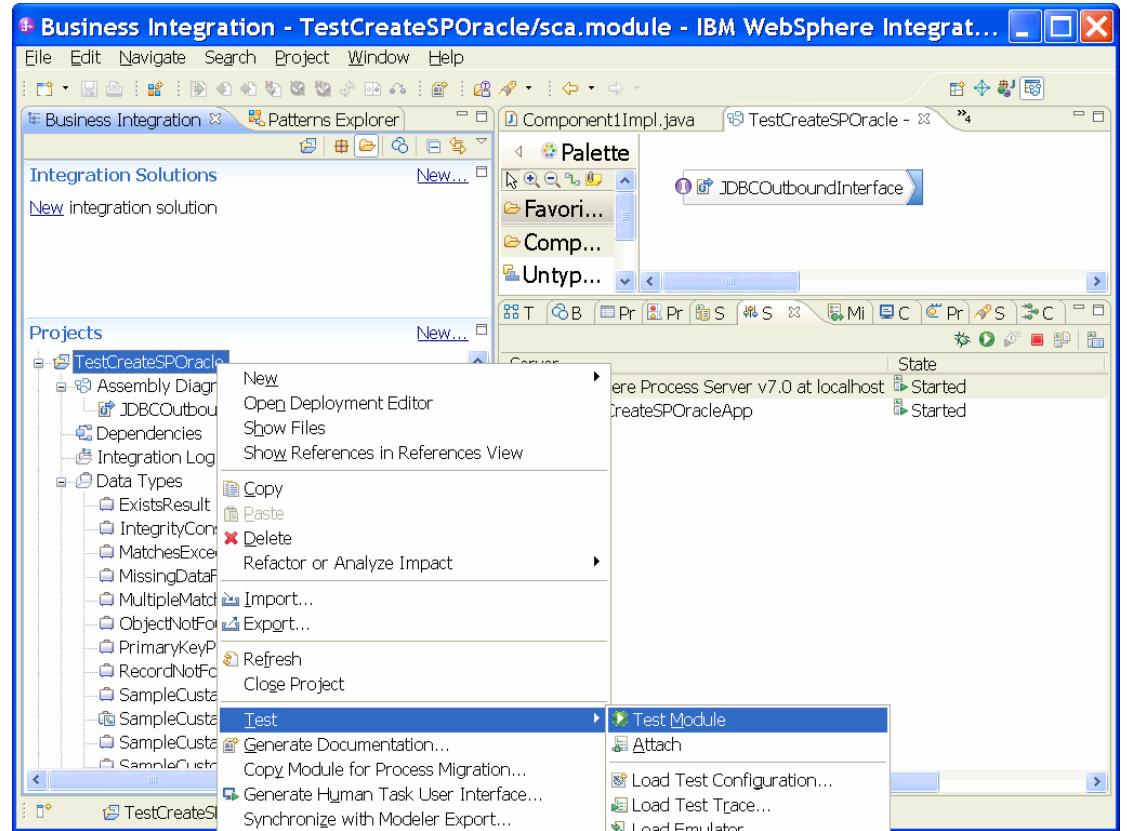
4. Select your project (**TestCreateSPOracleApp**) and click **Add** to configure the project on the server. Click **Finish**.



Test the assembled adapter application

Test the assembled adapter application using the WebSphere Integration Developer integration test client.

1. Select the **TestCreateSPOracle** module, right-click, and select **Test > Test Module**. The Test Client window is displayed.



2. Select **createSampleCustomerBG** from the **Operation** list.

WebSphere software

General Properties

Detailed Properties

Specify the component, interface, operation, and input parameter values for the Invoke event, then click the Continue icon in the Events area to run the test. [More...](#)

<u>Configuration:</u>	Default Module Test
<u>Module:</u>	TestCreateSPOracle
<u>Component:</u>	JDBCOutboundInterface
<u>Interface:</u>	JDBCOutboundInterface
<u>Operation:</u>	createSampleCustomerBG

Initial request parameters:

Value editor XML editor

The screenshot shows the 'Value editor' interface for specifying request parameters. It displays a hierarchical tree view of parameters under the operation 'createSampleCustomerBG'. The parameters listed are 'verb', 'pkey', 'fname', 'lname', 'ccode', and 'custadd'. The 'verb' parameter is set to 'Create'. The 'pkey', 'fname', 'lname', and 'ccode' parameters have their values filled in. The 'custadd' parameter is currently empty.

Name	Type	Value
createSampleCustomerBG	SampleCustomerBG	✓
verb	verb<string>	✓ Create
SampleCustomerBG	SampleCustomerBG	✓
pkey	string	✓
fname	string	✓
lname	string	✓
ccode	string	✓
custadd	SampleCustomerBG	✓

① Type: <http://www.ibm.com/xmlns/prod/websphere/j2ca/jdbc/>

3. Enter **Create** for the verb and specify values for **pkey**, **lname**, **fname** and **ccode** as shown in the figure.

WebSphere software

Initial request parameters:

Value editor XML editor

The screenshot shows the WebSphere Value editor interface. At the top, there are two radio buttons: 'Value editor' (which is selected) and 'XML editor'. Below the radio buttons is a table with three columns: 'Name', 'Type', and 'Value'. The table contains the following data:

Name	Type	Value
createSampleC	SampleCust...	✓
verb	verb<string>	✓ Create
SampleCust	SampleCust...	✓
pkey	string	✓ 100
fname	string	✓ testFname
lname	string	✓ testLname
ccode	string	✓ testcCode
custaddobj	SampleCust...	✓

At the bottom of the editor, there is a status bar with the text '(i) Type: http://www.w3.org/2001/XMLSchema#string'.

4. Right-click **custaddobj** and select **Add Elements**.

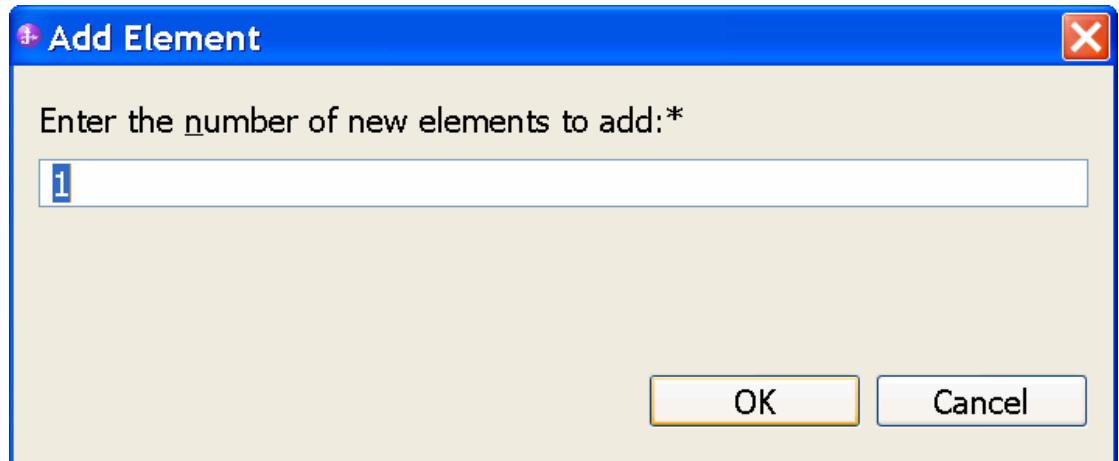
↳ Initial request parameters:
 Value editor XML editor

The screenshot shows the WebSphere Value editor interface with a context menu open over the 'custaddobj' row. The menu items are: 'Copy Value', 'Paste Value', 'Select All', and 'Add Elements...'. The 'Add Elements...' item is highlighted with a blue background.

The table data is identical to the one in the previous screenshot:

Name	Type	Value
createSampleC	SampleCust...	✓
verb	verb<string>	✓ Create
SampleCust	SampleCust...	✓
pkey	string	✓ 100
fname	string	✓ testFname
lname	string	✓ testLname
ccode	string	✓ testcCode
custaddobj	SampleCust...	✓

5. Enter 1 and click **OK**.



6. Enter values for **custaddobj[0]** as shown in the figure below.

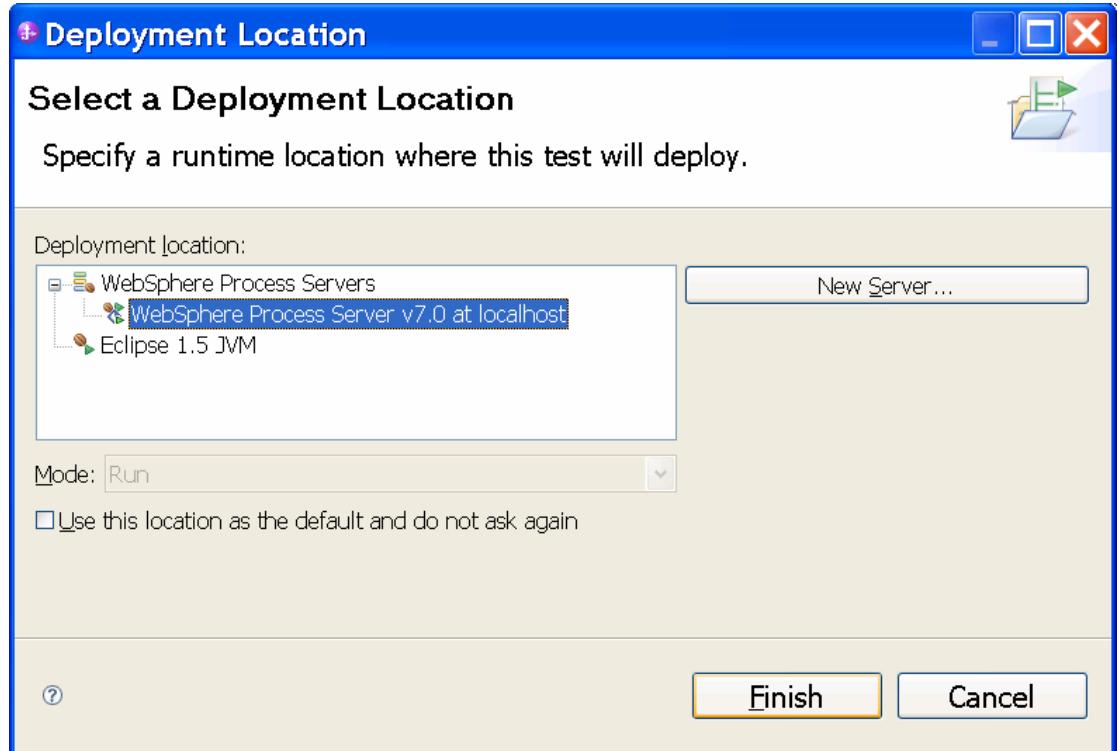
Initial request parameters:

Value editor XML editor

	Name	Type	Value
createSampleCusto	SampleCustom...	✓	
verb	verb<string>	✓	Create
SampleCustomer	SampleCustomer	✓	
pkey	string	✓	100
fname	string	✓	testFname
lname	string	✓	testLname
ccode	string	✓	testcCode
custaddobj	SampleCustadd[]	„	
custaddob	SampleCustadd	✓	
addrid	string	✓	100
custid	string	✓	101
city	string	✓	Beijing
zipcode	string	✓	100000

(i) Type: <http://www.w3.org/2001/XMLSchema#string>

7. To execute the service, click .
8. In the Select a Deployment Location window, select the server, and click **Finish**.



9. Check the data in the EIS to ensure that it is populated correctly.

Clear the sample content

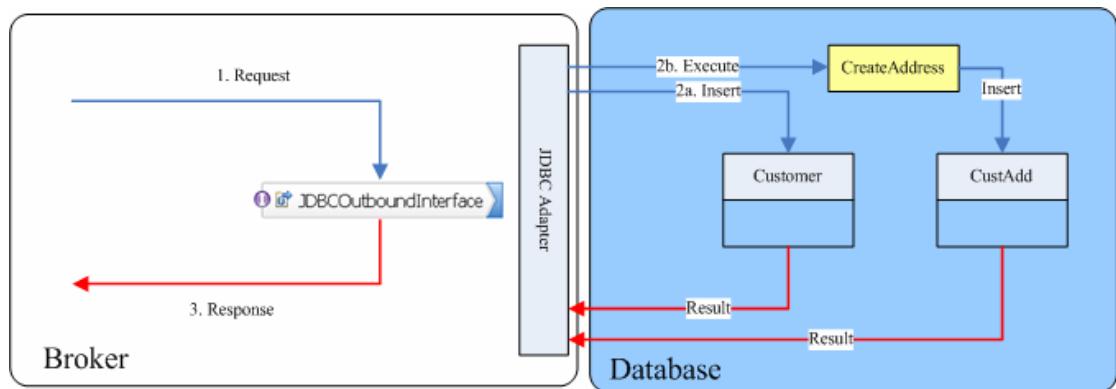
After you have tested the application, clear the sample content to return the data to its original state.

Chapter 3. Tutorial 2: Creating a record using parent-child business objects with a CreateSP associated with the child business object (SQL Server)

This tutorial demonstrates how WebSphere Adapter for JDBC 7.0.0.0 populates the Customer and Address information into the database where the CUSTOMER and ADDRESS tables have a parent-child relationship. A stored procedure is used to populate the Address (child) information.

About this task

In this scenario, an application SCA component raises a create Customer business object request to the JDBC Outbound Interface. The JDBC adapter generates SQL statements to insert corresponding CUSTOMER and ADDRESS records into database. Finally, the JDBC adapter generates response according to the input business object and the execution results of the SQL statements. The following figure represents this scenario:



Prepare to run through the tutorial

Extract the sample files

Replicas of the artifacts that you create when using the external service wizard are provided as sample files for your reference. Use these files to verify if the files you create using the external service wizard are correct.

Download the sample zip file and extract it into a directory of your choice (you may want to create a new directory).

Configuration prerequisites

Before configuring the adapter, you must complete the following tasks:

- Create tables and stored procedure

- Create an authentication alias
- Create a data source

Create tables and stored procedure

You must create the following tables and stored procedure in the MS SQLServer database before starting the scenario.

a. Script for creating CUSTOMER and ADDRESS tables

```
CREATE TABLE CUSTOMER (
    PKEY VARCHAR(10) NOT NULL PRIMARY KEY,
    FNAME VARCHAR(20) ,
    LNAME VARCHAR(20) ,
    CCODE VARCHAR(10) ) ;

CREATE TABLE ADDRESS (
    ADDRID VARCHAR(10) NOT NULL PRIMARY KEY,
    CUSTID VARCHAR(10) ,
    CITY VARCHAR(20) ,
    ZIPCODE VARCHAR(10) ) ;
```

b. Script for create CREATEADDRESS procedure

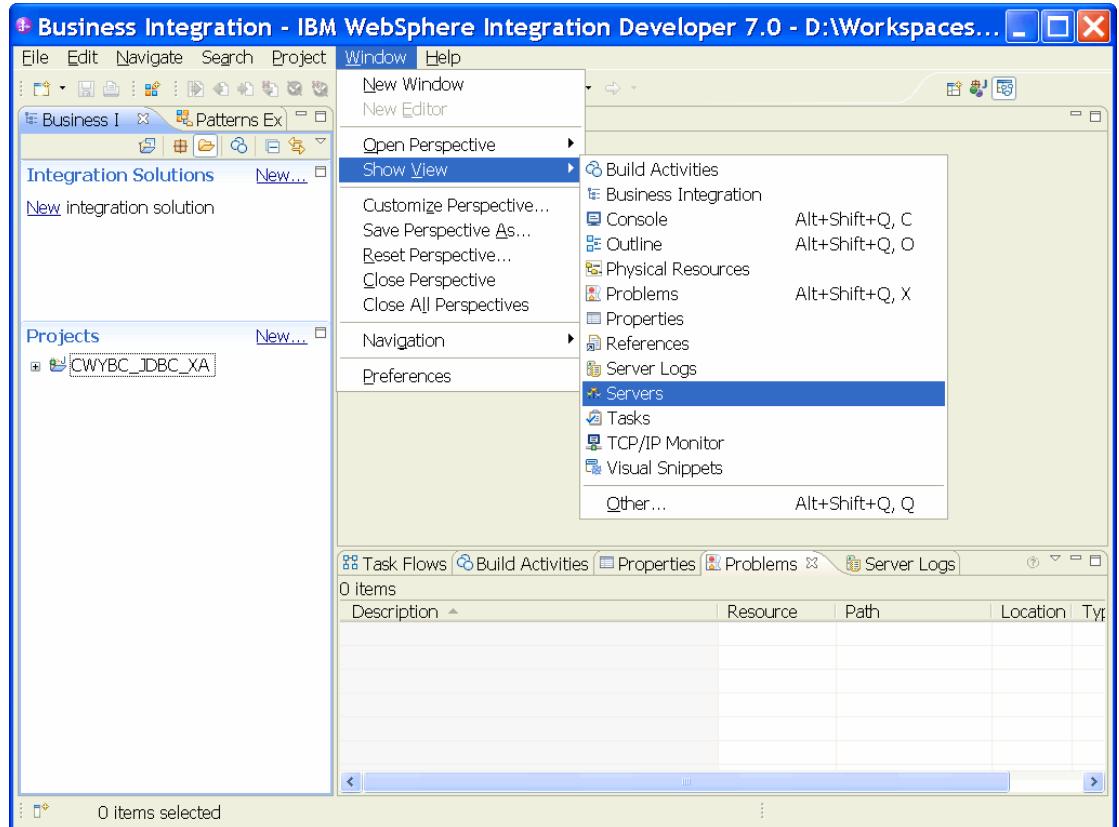
```
CREATE PROCEDURE CREATEADDRESS
(@addr_id varchar, @cust_id varchar, @city varchar,
@zipcode varchar)
AS
begin
INSERT into ADDRESS (ADDRID, CUSTID, CITY, ZIPCODE)
values
        (@addr_id, @cust_id, @city, @zipcode);
end;
```

Create an authentication alias

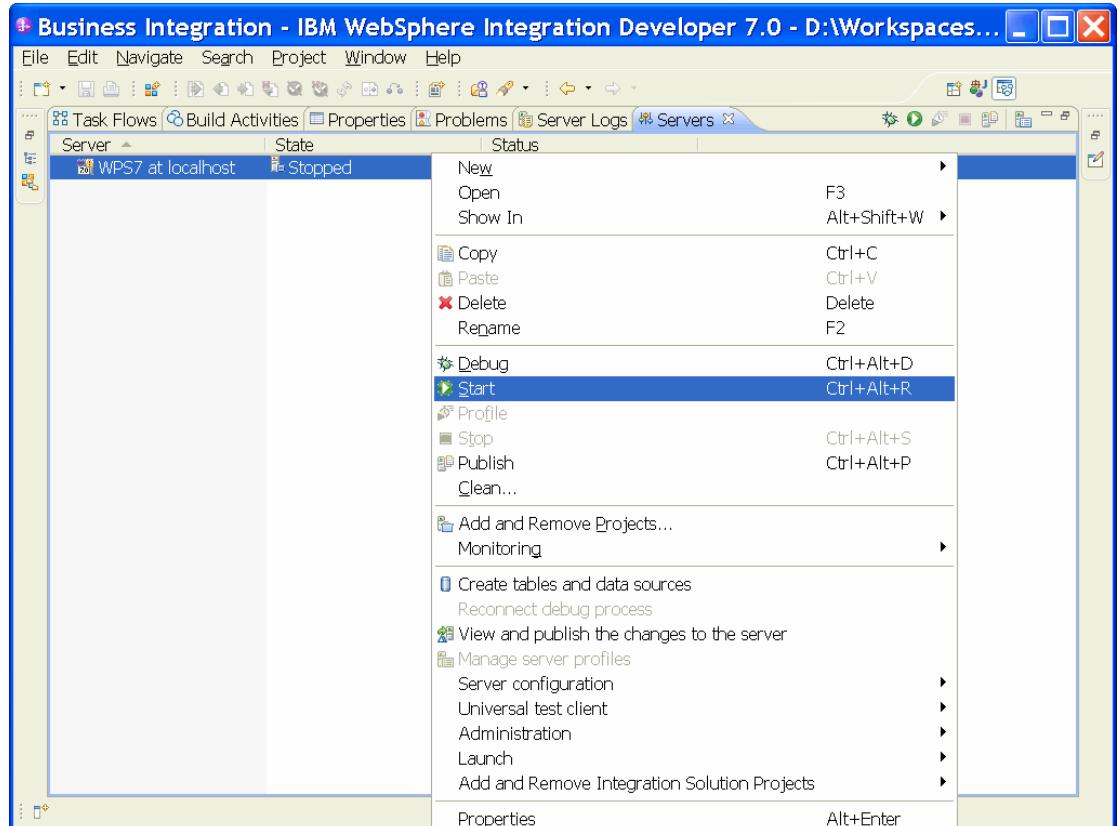
The authentication alias needs to be set because the data source that is used to generate artifacts will use the username and password set in the authentication alias to connect to the database.

Follow these steps to set the authentication alias in the WebSphere Process Server administrative console.

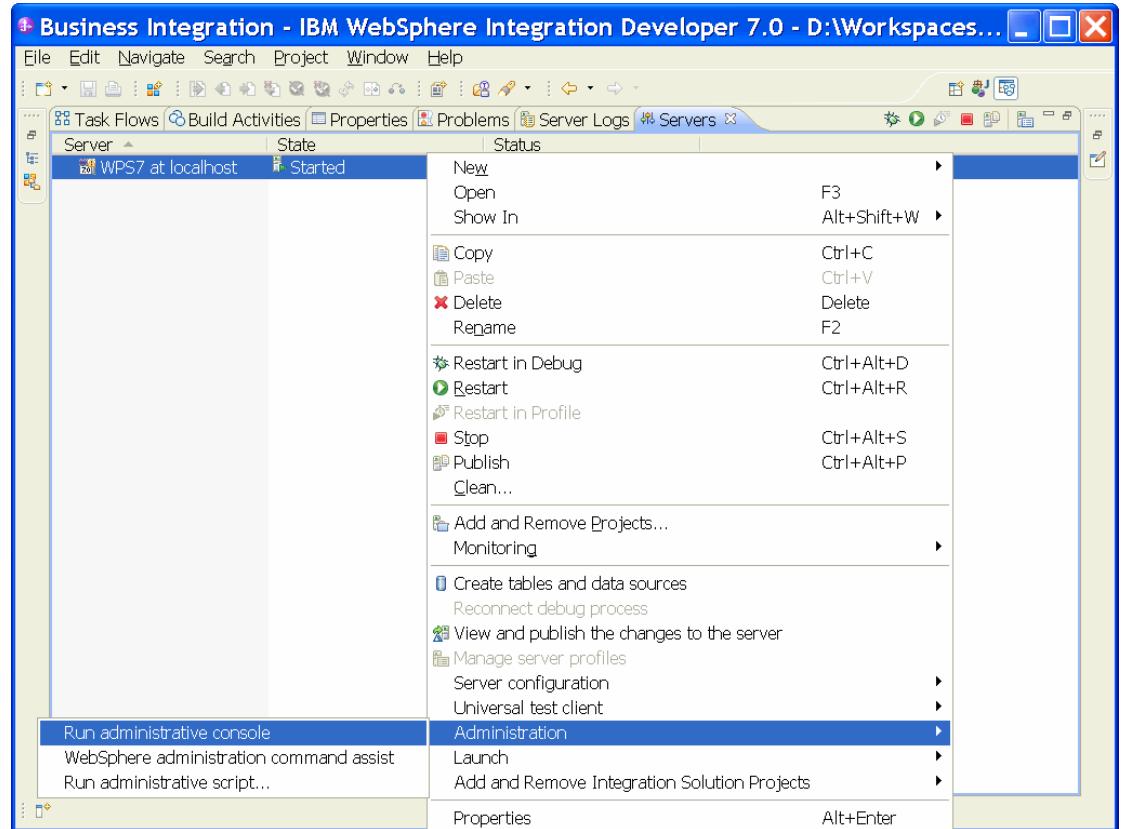
1. In WebSphere Integration Developer, switch to the **Servers** view by selecting **Windows > Show View > Servers**.



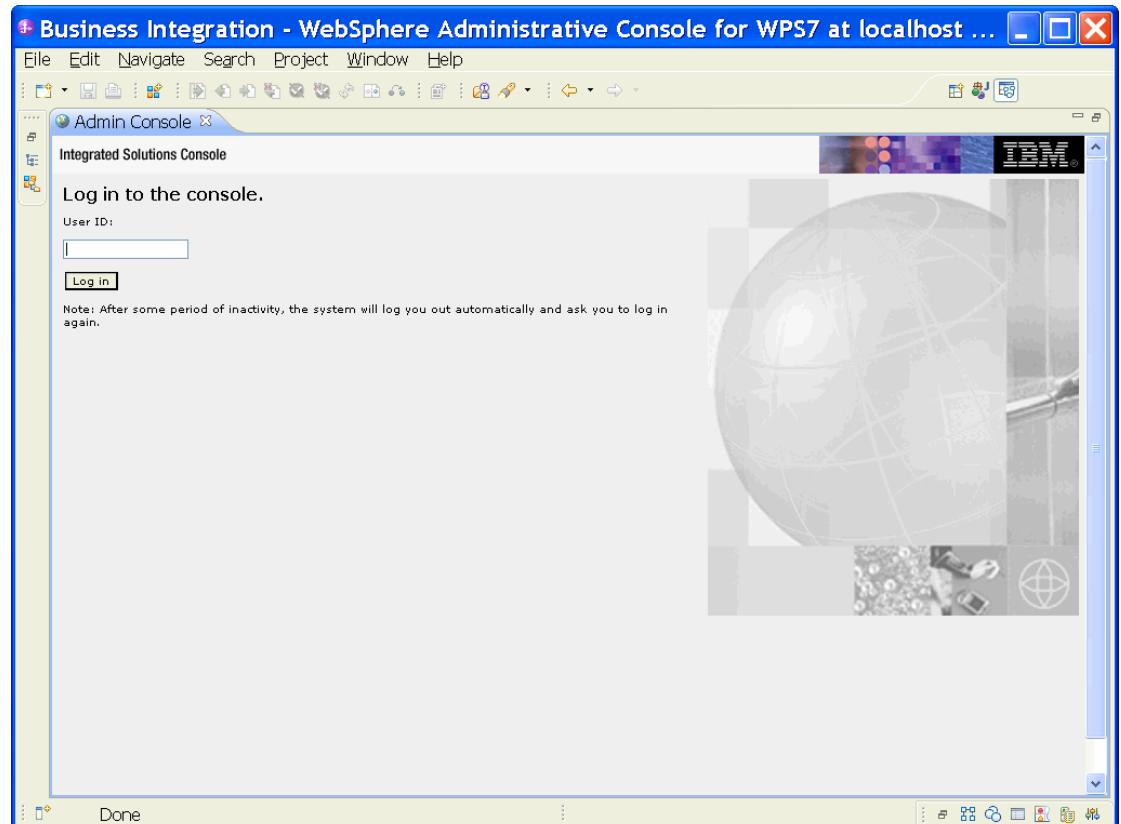
2. In the **Servers** view, right-click the server that you want to start and select **Start**.



3. After the server is started, right-click the server, and select **Administration > Run administrative console**.



4. Log on to the administrative console.



5. Click **Security → Global security**.

WebSphere software

View: All tasks

- Welcome
- [+] Guided Activities
- [+] Servers
- [+] Applications
- [+] Services
- [+] Resources
- [-] Security
 - Business Integration Security
 - Global security
 - Security domains
 - Administrative Authorization Groups
 - SSL certificate and key management
 - Security auditing
 - Bus security
- [+] Environment
- [+] Integration Applications
- [+] System administration
- [+] Users and Groups
- [+] Monitoring and Tuning
- [+] Troubleshooting
- [+] Service integration
- [+] UDDI

6. Under **Java Authentication and Authorization Service**, click **J2C authentication data**.

Cell=localhostNode01Cell, Profile=AppSrv01 Close page

Global security

Global security
Use this panel to configure administration and the default application security policy. This security configuration applies to functions and is used as a default security policy for user applications. Security domains can be defined to override and control applications.

Security Configuration Wizard	Security Configuration Report
<div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> Administrative security <p><input type="checkbox"/> Enable administrative security Administrative user roles <input type="checkbox"/> Administrative group roles <input type="checkbox"/> Administrative authentication</p> </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> Application security <p><input checked="" type="checkbox"/> Enable application security</p> </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> Java 2 security <p><input type="checkbox"/> Use Java 2 security to restrict application access to local resources <input checked="" type="checkbox"/> Warn if applications are granted custom permissions <input type="checkbox"/> Restrict access to resource authentication data</p> </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> User account repository Current realm definition: Federated repositories Available realm definitions: Federated repositories Configure... Set as current </div>	
<div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> Authentication Authentication mechanisms and expiration <input checked="" type="radio"/> LTPA <input type="radio"/> Kerberos and LTPA Kerberos configuration <input type="radio"/> SWAM (deprecated): No authentication Authentication cache settings </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> <input type="checkbox"/> Web and SIP security <input type="checkbox"/> RMI/IOP security <input type="checkbox"/> Java Authentication and Authorization <input type="checkbox"/> Application logins <input type="checkbox"/> System logins <input type="checkbox"/> J2C authentication data </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> <p>Specifies a list of user identities and passwords for Java(TM) 2 connector security to use.</p> <p>Security domains External authorization providers Custom properties</p> </div>	

WebSphere software

A list of existing aliases is displayed.

[Global security > JAAS - J2C authentication data](#)

Specifies a list of user identities and passwords for Java(TM) 2 connector security to use.

Prefix new alias names with the node name of the cell (for compatibility with earlier releases)

[Apply](#)

[+ Preferences](#)

New	Delete			
Select	Alias	User ID	Description	
You can administer the following resources:				
<input type="checkbox"/>	BSpace JDBC Alias	wbiuser	Business Space Authorization Alias	
<input type="checkbox"/>	CommonEventInfrastructureJMSAuthAlias	wbiuser	Authentication alias for the Common Event Infrastructure JMS Topics and Queues	
<input type="checkbox"/>	SCA Auth Alias	wbiuser	This is the alias used by SCA to login to a secured SIBus	
<input type="checkbox"/>	localhostNode01Cell/nlNode01/server1/EventAuthDataAliasDerby		Derby authentication alias for the Event Server	
Total 4				

7. Click **New** to create a new authentication entry. Type the alias name, and username and password to connect to the database. Click **OK**.

WebSphere software

Cell=localhostNode01Cell, Profile=AppSrv01

Global security

[Global security > JAAS - J2C authentication data > New](#)

Specifies a list of user identities and passwords for Java(TM) 2 connector security to use.

General Properties

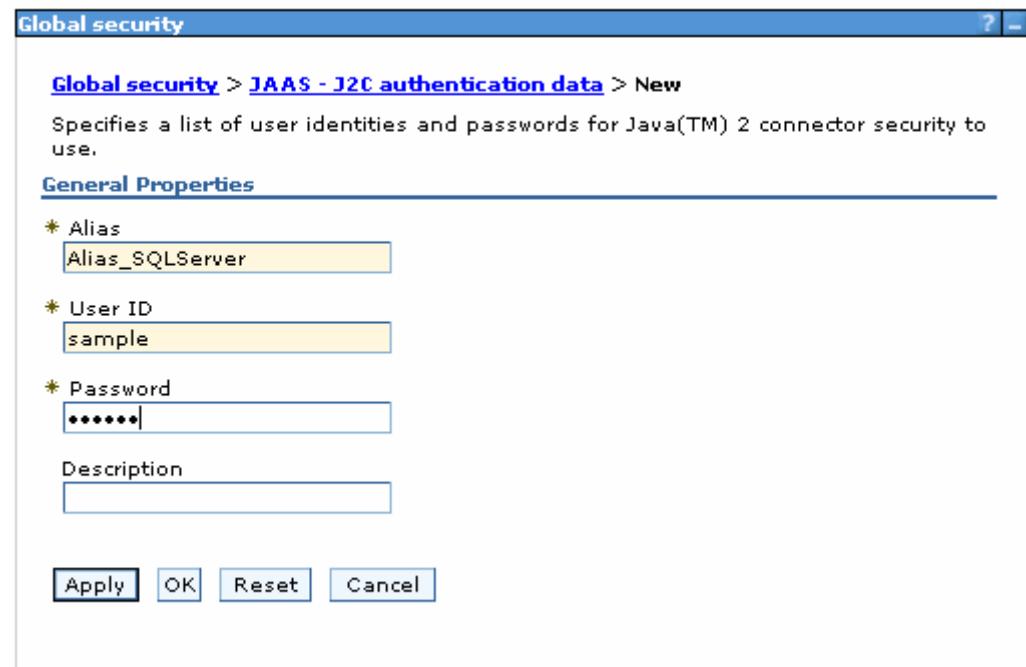
* Alias
Alias_SQLServer

* User ID
sample

* Password

Description

Apply OK Reset Cancel



8. Click **Save** to save the changes.

Cell=localhostNode01Cell, Profile=AppSrv01

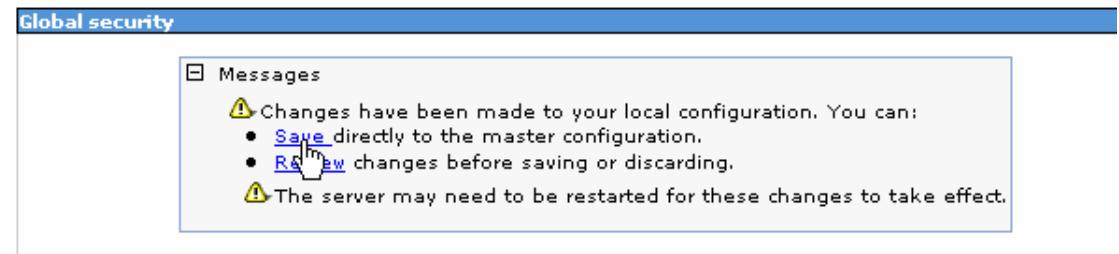
Global security

Messages

Changes have been made to your local configuration. You can:

- [Save](#) directly to the master configuration.
- [Rollback](#) changes before saving or discarding.

The server may need to be restarted for these changes to take effect.



You have created an authentication alias that will be used to configure the data source.

WebSphere software

Preferences

New Delete

Select Alias User ID Description

You can administer the following resources:

<input type="checkbox"/>	BSpace JDBC Alias	wbiuser	Business Space Authorization Alias
<input type="checkbox"/>	CommonEventInfrastructureJMSAuthAlias	wbiuser	Authentication alias for the Common Event Infrastructure JMS Topics and Queues
<input type="checkbox"/>	SCA Auth Alias	wbiuser	This is the alias used by SCA to login to a secured SIBus
<input type="checkbox"/>	localhostNode01Cell/n1Node01/server1/EventAuthDataAliasDerby		Derby authentication alias for the Event Server
<input type="checkbox"/>	n1Node01/Alias_Oracle	sample	
<input type="checkbox"/>	n1Node01/Alias_SQLServer	sample	

Total 6

Create a data source

Create a data source in WebSphere Process Server, which the adapter will use to connect to the database. This data source is used later when generating the artifacts for the module.

Note: This tutorial uses SQL Server as the database and the SQL Server JDBC driver sqljdbc.jar.

Here are the steps to create the data source in the WebSphere Process Server administrative console.

1. In the administrative console, select **Environment → WebSphere Variables**.

WebSphere software



2. On the right, select **MICROSOFT_JDBC_DRIVER_PATH** and specify the path of the sqljdbc.jar file in the **Value** field. Click **OK**.

WebSphere software

Cell=localhostNode01Cell, Profile=AppSrv01

WebSphere Variables

[WebSphere Variables](#) > **MICROSOFT_JDBC_DRIVER_PATH**

Use this page to define substitution variables. Variables specify a level of indirection for some system-defined values, such as file system root directories. Variables have a scope level, which is either server, node, cluster, or cell. Values at one scope level can differ from values at other levels. When a variable has conflicting scope values, the more granular scope value overrides values at greater scope levels. Therefore, server variables override node variables, which override cluster variables, which override cell variables.

Configuration

General Properties

* Name: **MICROSOFT_JDBC_DRIVER_F**

Value: **D:\Lib**

Description:
The directory that contains the Microsoft SQL Server JDBC Driver.

Apply OK Reset Cancel

3. Click **Save** to save the changes.

WebSphere Variables

Messages

⚠ Changes have been made to your local configuration. You can:

- [Save](#) directly to the master configuration.
- [Review](#) changes before saving or discarding.

⚠ The server may need to be restarted for these changes to take effect.

The variable is added and appears in the list.

WebSphere software

Preferences			
	New	Delete	
Select	Name	Value	Scope
You can administer the following resources:			
<input type="checkbox"/>	MICROSOFT JDBC DRIVER_NATIVEPATH		Node=nINode01
<input type="checkbox"/>	MICROSOFT JDBC DRIVER_PATH	D:\Lib	Node=nINode01
<input type="checkbox"/>	MQ_INSTALL_ROOT	\$(WAS_INSTALL_ROOT)/lib/WMQ	Node=nINode01
<input type="checkbox"/>	ORACLE JDBC DRIVER_PATH	D:\Lib	Node=nINode01
<input type="checkbox"/>	OS400_NATIVE JDBC40_DRIVER_PATH		Node=nINode01
<input type="checkbox"/>	OS400_NATIVE JDBC_DRIVER_PATH		Node=nINode01
<input type="checkbox"/>	OS400_TOOLBOX JDBC DRIVER_PATH		Node=nINode01
<input type="checkbox"/>	SCA_BUS_ID	localhostNode01Cell	Cell=localhostNode01Cell
<input type="checkbox"/>	SERVER_LOG_ROOT	\$(LOG_ROOT)/server1	Node=nINode01,Server=serve
<input type="checkbox"/>	SYBASE JDBC DRIVER_PATH		Node=nINode01
<input type="checkbox"/>	UNIVERSAL JDBC DRIVER_PATH	\$(WAS_INSTALL_ROOT)/universalDriver/lib	Node=nINode01

4. Select **Resources → JDBC -> JDBC Providers.**

WebSphere software



5. Click **New** in the JDBC providers window.

Cell=localhostNode01Cell, Profile=AppSrv01

JDBC providers

Use this page to edit properties of a JDBC provider. The JDBC provider object encapsulates the specific JDBC driver implementation class for access to the specific vendor database of your environment. Learn more about this task in a [guided activity](#). A guided activity provides a list of task steps and more general information about the topic.

Scope: Cell=**localhostNode01Cell**, Node=**nINode01**

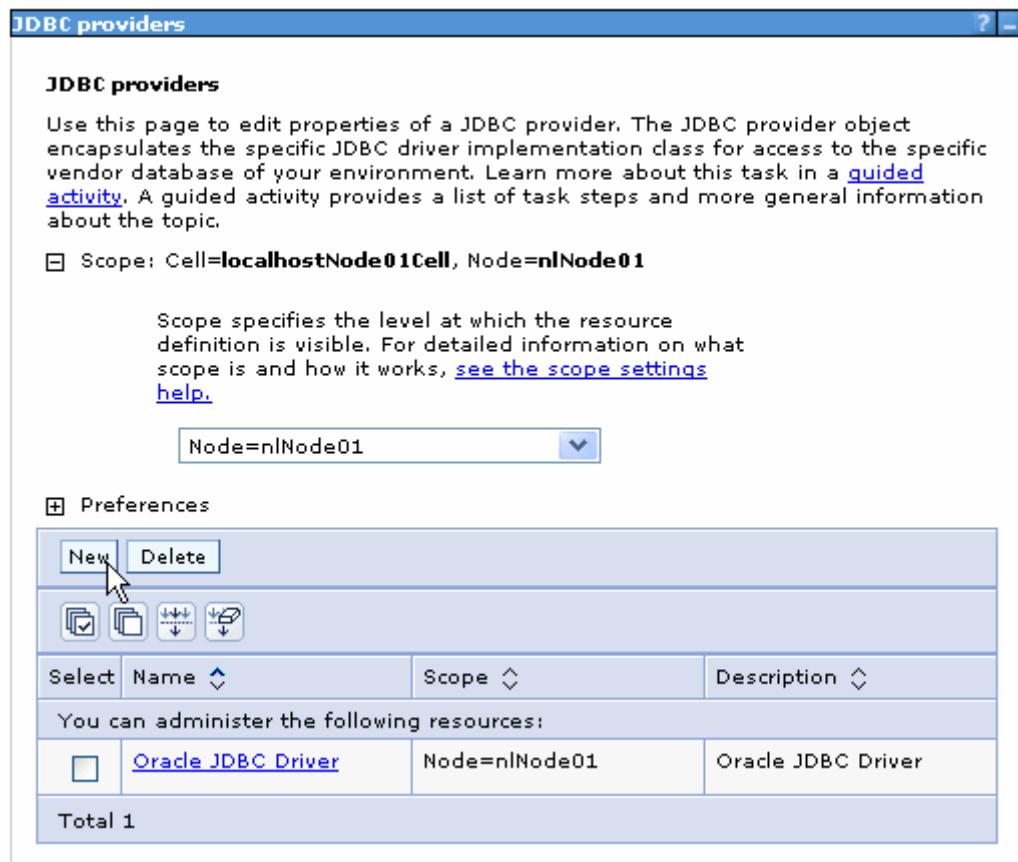
Scope specifies the level at which the resource definition is visible. For detailed information on what scope is and how it works, [see the scope settings help](#).

Node=nINode01

Preferences

Select	Name	Scope	Description
<input type="checkbox"/>	Oracle JDBC Driver	Node=nINode01	Oracle JDBC Driver

Total 1



6. In the Create new JDBC provider page, select an SQL Server database with a connection pool data source for the SQL Server JDBC driver. Click **Next**.

Create a new JDBC Provider

Step 1: Create new JDBC provider

Step 2: Enter database class path information

Step 3: Summary

Create new JDBC provider

Set the basic configuration values of a JDBC provider, which encapsulates the specific vendor JDBC driver implementation classes that are required to access the database. The wizard fills in the name and the description fields, but you can type different values.

Scope
cells:localhostNode01Cell:nodes:nNode01

* Database type
SQL Server

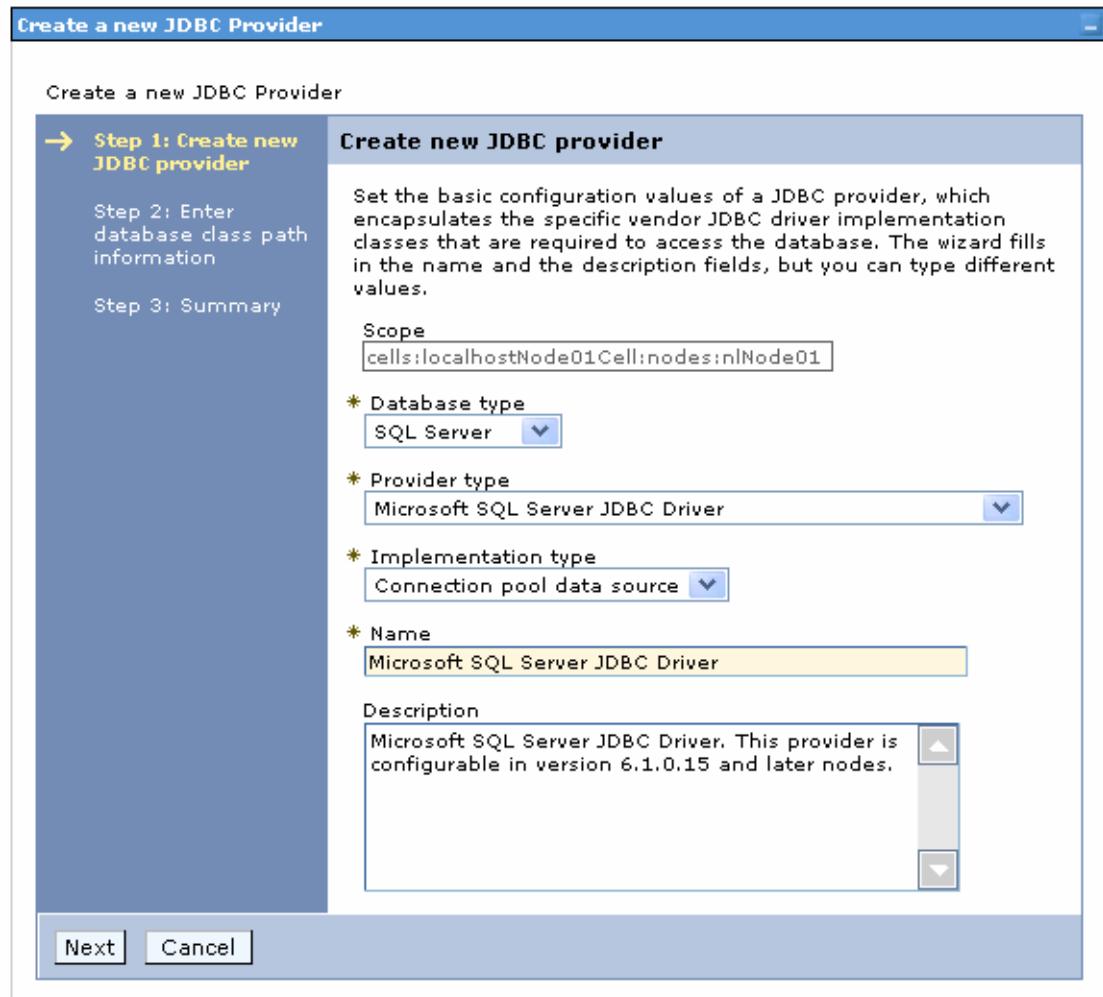
* Provider type
Microsoft SQL Server JDBC Driver

* Implementation type
Connection pool data source

* Name
Microsoft SQL Server JDBC Driver

Description
Microsoft SQL Server JDBC Driver. This provider is configurable in version 6.1.0.15 and later nodes.

Next Cancel



7. In the Enter database classpath information page, enter the following value in the **Class path** field:
\$(MICROSOFT_JDBC_DRIVER_PATH)/sqljdbc.jar, where
\$(MICROSOFT_JDBC_DRIVER_PATH) is library path for the run time.
8. Click **Next**.

WebSphere software

Cell=localhostNode01Cell, Profile=AppSrv01

Create a new JDBC Provider

Step 1: Create new JDBC provider

→ Step 2: Enter database class path information

Step 3: Summary

Enter database class path information

Set the environment variables that represent the JDBC driver class files, which WebSphere(R) Application Server uses to define your JDBC provider. This wizard page displays the file names; you supply only the directory locations of the files. Use complete directory paths when you type the JDBC driver file locations. For example: C:\SQLLIB\java on Windows(R) or /home/db2inst1/sqllib/java on Linux(TM).

If a value is specified for you, you may click Next to accept the value.

Class path:

`#{MICROSOFT_JDBC_DRIVER_PATH}/sqljdbc.jar`

Directory location for "sqljdbc.jar" which is saved as WebSphere variable `#{MICROSOFT_JDBC_DRIVER_PATH}`

`D:\Lib`

Native library path

Directory location which is saved as WebSphere variable `#{MICROSOFT_JDBC_DRIVER_NATIVEPATH}`

`#{MICROSOFT_JDBC_DRIVER_NATIVEPATH}`

Previous | Next | Cancel

9. In the Summary page, click **Finish**.

Cell=localhostNode01Cell, Profile=AppSrv01

[Close page](#)

Create a new JDBC Provider

Step 1: Create new JDBC provider

Step 2: Enter database class path information

→ Step 3: Summary

Summary

Summary of actions:

Options	Values
Scope	cells:localhostNode01Cell:nodes:n1Node01
JDBC provider name	Microsoft SQL Server JDBC Driver
Description	Microsoft SQL Server JDBC Driver. This provider is configurable in version 6.1.0.15 and later nodes.
Class path	<code>#{MICROSOFT_JDBC_DRIVER_PATH}/sqljdbc.jar</code>
<code>#{MICROSOFT_JDBC_DRIVER_PATH}</code>	<code>D:\Lib</code>
Native path	<code>#{MICROSOFT_JDBC_DRIVER_NATIVEPATH}</code>
<code>#{MICROSOFT_JDBC_DRIVER_NATIVEPATH}</code>	
Implementation class name	<code>com.microsoft.sqlserver.jdbc.SQLServerConnectionPoolDataSource</code>

Previous | Finish | Cancel

10. Click **Save** to save the changes.

WebSphere software

Cell=localhostNode01Cell, Profile=AppSrv01

JDBC providers

Messages

Changes have been made to your local configuration.
You can:

- Save directly to the master configuration.
- Review changes before saving or discarding.

The server may need to be restarted for these changes to take effect.

The JDBC provider is added and appears in the list.

Cell=localhostNode01Cell, Profile=AppSrv01

JDBC providers

Use this page to edit properties of a JDBC provider. The JDBC provider object encapsulates the specific JDBC driver implementation class for access to the specific vendor database of your environment. Learn more about this task in a [guided activity](#). A guided activity provides a list of task steps and more general information about the topic.

Scope: Cell=**localhostNode01Cell**, Node=**nINode01**

Scope specifies the level at which the resource definition is visible. For detailed information on what scope is and how it works, [see the scope settings help](#).

Node=nINode01

Preferences

New	Delete		
<input type="checkbox"/>	<input type="checkbox"/> Microsoft SQL Server JDBC Driver	Node=nINode01	Microsoft SQL Server JDBC Driver. This provider is configurable in version 6.1.0.15 and later nodes.
<input type="checkbox"/>	<input type="checkbox"/> Oracle JDBC Driver	Node=nINode01	Oracle JDBC Driver

Total 2

11. Select the SQL Server JDBC provider you created. Under **Additional Properties**, click **Data sources**. Click **New**.

WebSphere software

Cell=localhostNode01Cell, Profile=AppSrv01

JDBC providers

JDBC providers > Microsoft SQL Server JDBC Driver > Data sources

Use this page to edit the settings of a datasource that is associated with your selected JDBC provider. The datasource object supplies your application with connections for accessing the database. Learn more about this task in a [guided activity](#). A guided activity provides a list of task steps and more general information about the topic.

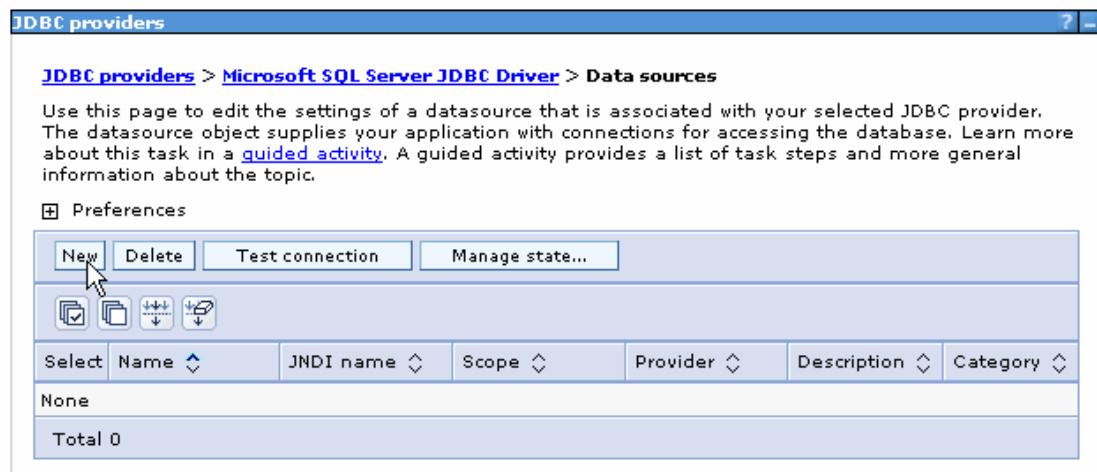
Preferences

New Delete Test connection Manage state...

Select Name JNDI name Scope Provider Description Category

None

Total 0



12. Type any value in the **JNDI name** field, and select the authentication alias. Click **Next**.

Cell=localhostNode01Cell, Profile=AppSrv01

Create a data source

Step 1: Enter basic data source information

Set the basic configuration values of a datasource for association with your JDBC provider. A datasource supplies the physical connections between the application server and the database.

Requirement: Use the Datasources (WebSphere(R) Application Server V4) console pages if your applications are based on the Enterprise JavaBeans(TM) (EJB) 1.0 specification or the Java(TM) Servlet 2.2 specification.

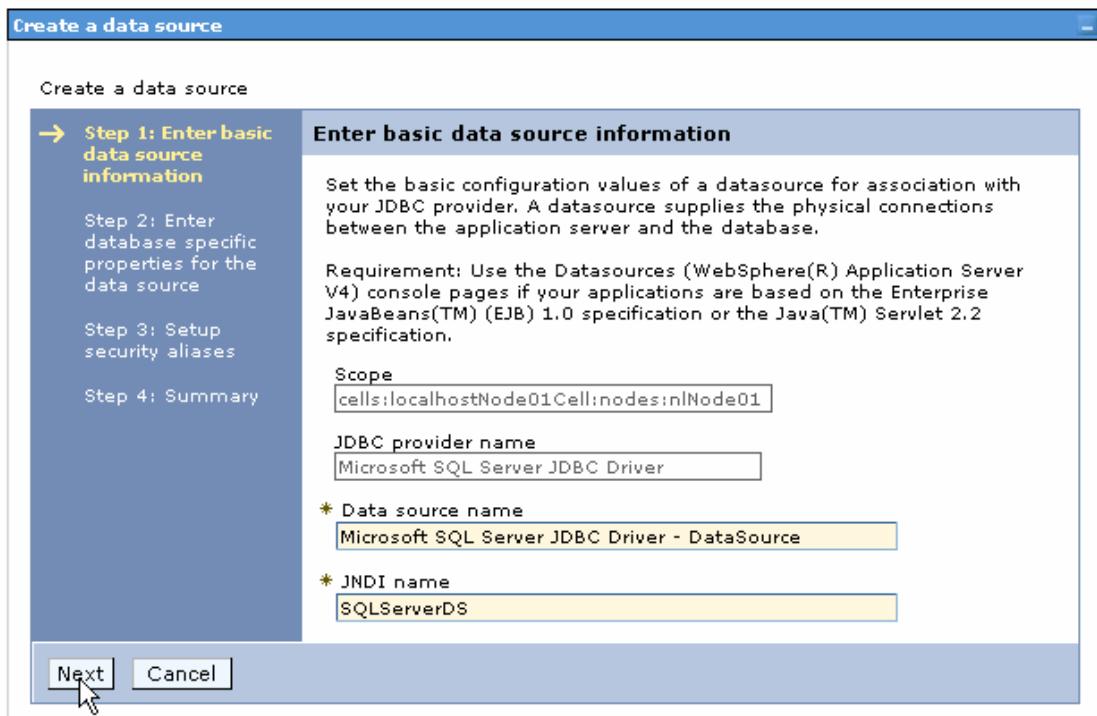
Scope
cells:localhostNode01Cell:nodes:n1Node01

JDBC provider name
Microsoft SQL Server JDBC Driver

* Data source name
Microsoft SQL Server JDBC Driver - DataSource

* JNDI name
SQLServerDS

Next Cancel



13. Enter appropriate values in the **Database name**, **Port number**, and **Server name** fields. Click **Next**.

Cell=localhostNode01Cell, Profile=AppSrv01

Create a data source

Step 1: Enter basic data source information → Step 2: Enter database specific properties for the data source Step 3: Setup security aliases Step 4: Summary	Enter database specific properties for the data source Set these database-specific properties, which are required by the database vendor JDBC driver to support the connections that are managed through the datasource.								
	<table border="1" style="width: 100%;"> <thead> <tr> <th>Name</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Database name</td> <td>sample</td> </tr> <tr> <td>Port number</td> <td>1433</td> </tr> <tr> <td>Server name</td> <td>9.181.84.136</td> </tr> </tbody> </table>	Name	Value	Database name	sample	Port number	1433	Server name	9.181.84.136
Name	Value								
Database name	sample								
Port number	1433								
Server name	9.181.84.136								
	<input checked="" type="checkbox"/> Use this data source in container managed persistence (CMP)								
Previous	Next								
Cancel									

14. Select the authentication alias you just created from the **Component-managed authentication alias** list and click **Next**.

Cell=localhostNode01Cell, Profile=AppSrv01

Create a data source

Step 1: Enter basic data source information Step 2: Enter database specific properties for the data source → Step 3: Setup security aliases Step 4: Summary	Setup security aliases Select the authentication values for this resource. Component-managed authentication alias: nlNode01/Alias_SQLServer Mapping-configuration alias: (none) Container-managed authentication alias: (none) Note: You can create a new J2C authentication alias by accessing one of the following links. Clicking on a link will cancel the wizard and your current wizard selections will be lost. Global J2C authentication alias Security domains
Previous	Next
Cancel	

15. In the Summary page, review the values entered for the data source and click **Finish**.

WebSphere software

Cell=localhostNode01Cell, Profile=AppSrv01

Create a data source

Step 1: Enter basic data source information
Step 2: Enter database specific properties for the data source
Step 3: Setup security aliases
→ Step 4: Summary

Summary	
Summary of actions:	
Options	Values
Scope	cells:localhostNode01Cell:nodes:nNode01
Data source name	Microsoft SQL Server JDBC Driver - DataSource
JNDI name	SQLServerDS
Select an existing JDBC provider	Microsoft SQL Server JDBC Driver
Implementation class name	com.microsoft.sqlserver.jdbc.SQLServerConnectionPoolDataSource
Database name	sample
Port number	1433
Server name	9.181.84.136
Use this data source in container managed persistence (CMP)	true
Component-managed authentication alias	nNode01/Alias_SQLServer
Mapping-configuration alias	(none)
Container-managed authentication alias	(none)

Previous | Finish | Cancel

16. Click **Save** to save the changes.

Cell=localhostNode01Cell, Profile=AppSrv01

DATABASE providers

Messages

⚠ Changes have been made to your local configuration. You can:
• [Save](#) directly to the master configuration.
• [Revert](#) changes before saving or discarding.
⚠ The server may need to be restarted for these changes to take effect.

17. Select the check box corresponding to the data source you created in the previous step and click **Test connection**.

WebSphere software

Cell=localhostNode01Cell, Profile=AppSrv01

JDBC providers

JDBC providers > Microsoft SQL Server JDBC Driver > Data sources

Use this page to edit the settings of a datasource that is associated with your selected JDBC provider. The datasource object supplies your application with connections for accessing the database. Learn more about this task in a [guided activity](#). A guided activity provides a list of task steps and more general information about the topic.

Preferences

New Delete Test connection Manage state...

Select Name JNDI name Scope Provider Description Category

You can administer the following resources:

<input checked="" type="checkbox"/>	Microsoft SQL Server JDBC Driver - DataSource	SQLServerDS	Node=n1Node01	Microsoft SQL Server JDBC Driver	Data source for the Microsoft SQL Server JDBC Driver. This data source type is configurable in version 6.1.0.15 and later nodes.	
-------------------------------------	---	-------------	---------------	----------------------------------	--	--

Total 1

The connection should succeed shown in the following figure. If you experience problems while testing the connection, refer to the "Troubleshooting" section.

Cell=localhostNode01Cell, Profile=AppSrv01

JDBC providers

JDBC providers > Microsoft SQL Server JDBC Driver > Data sources

Use this page to edit the settings of a datasource that is associated with your selected JDBC provider. The datasource object supplies your application with connections for accessing the database. Learn more about this task in a [guided activity](#). A guided activity provides a list of task steps and more general information about the topic.

Messages

The test connection operation for data source Microsoft SQL Server JDBC Driver - DataSource on server server1 at node n1Node01 was successful with 6 warning(s). [View JVM log](#) for further details.

Preferences

New Delete Test connection Manage state...

Select Name JNDI name Scope Provider Description Category

You can administer the following resources:

<input type="checkbox"/>	Microsoft SQL Server JDBC Driver - DataSource	SQLServerDS	Node=n1Node01	Microsoft SQL Server JDBC Driver	Data source for the Microsoft SQL Server JDBC Driver. This data source type is configurable in version 6.1.0.15 and later nodes.	
--------------------------	---	-------------	---------------	----------------------------------	--	--

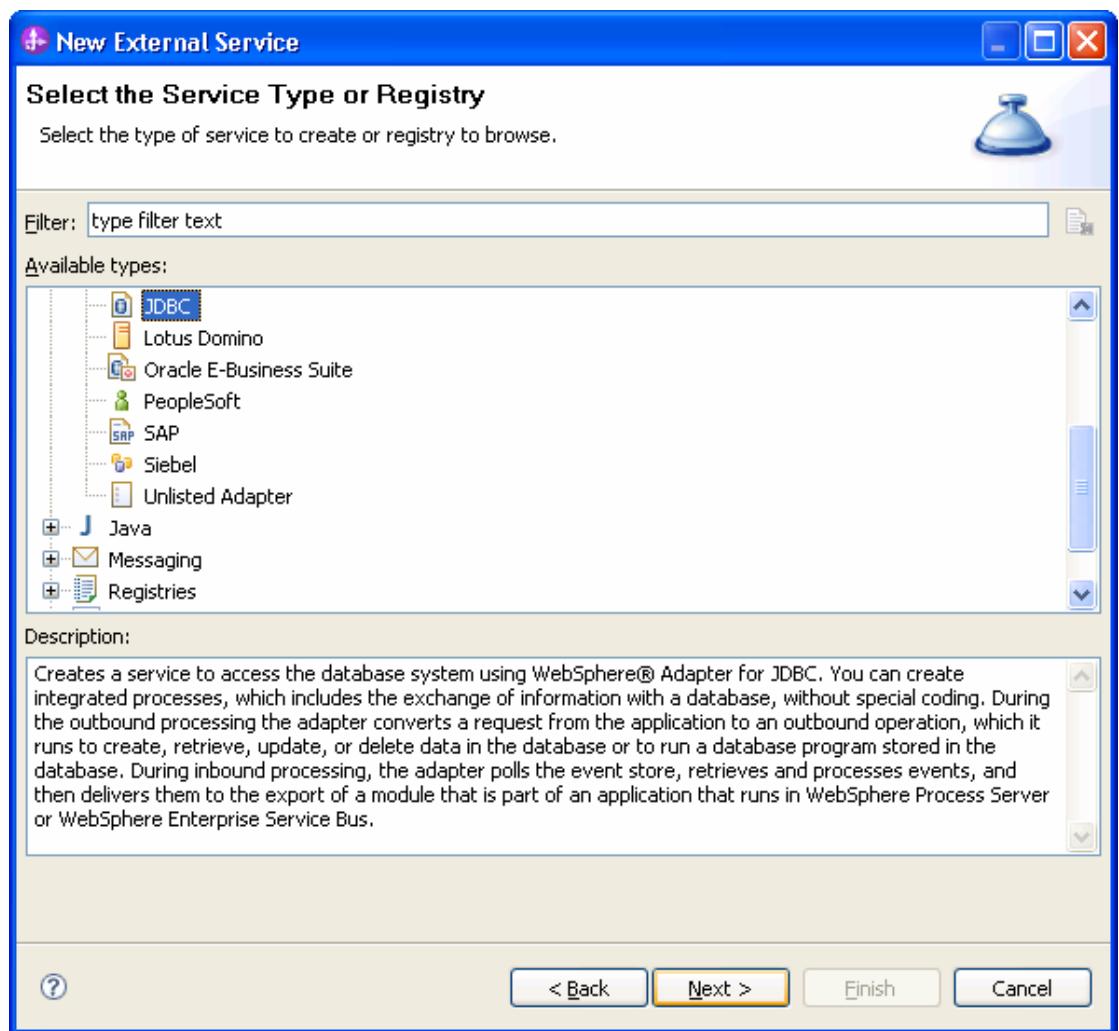
Total 1

Note: The data source is created which will be used by the adapter to connect to the database.

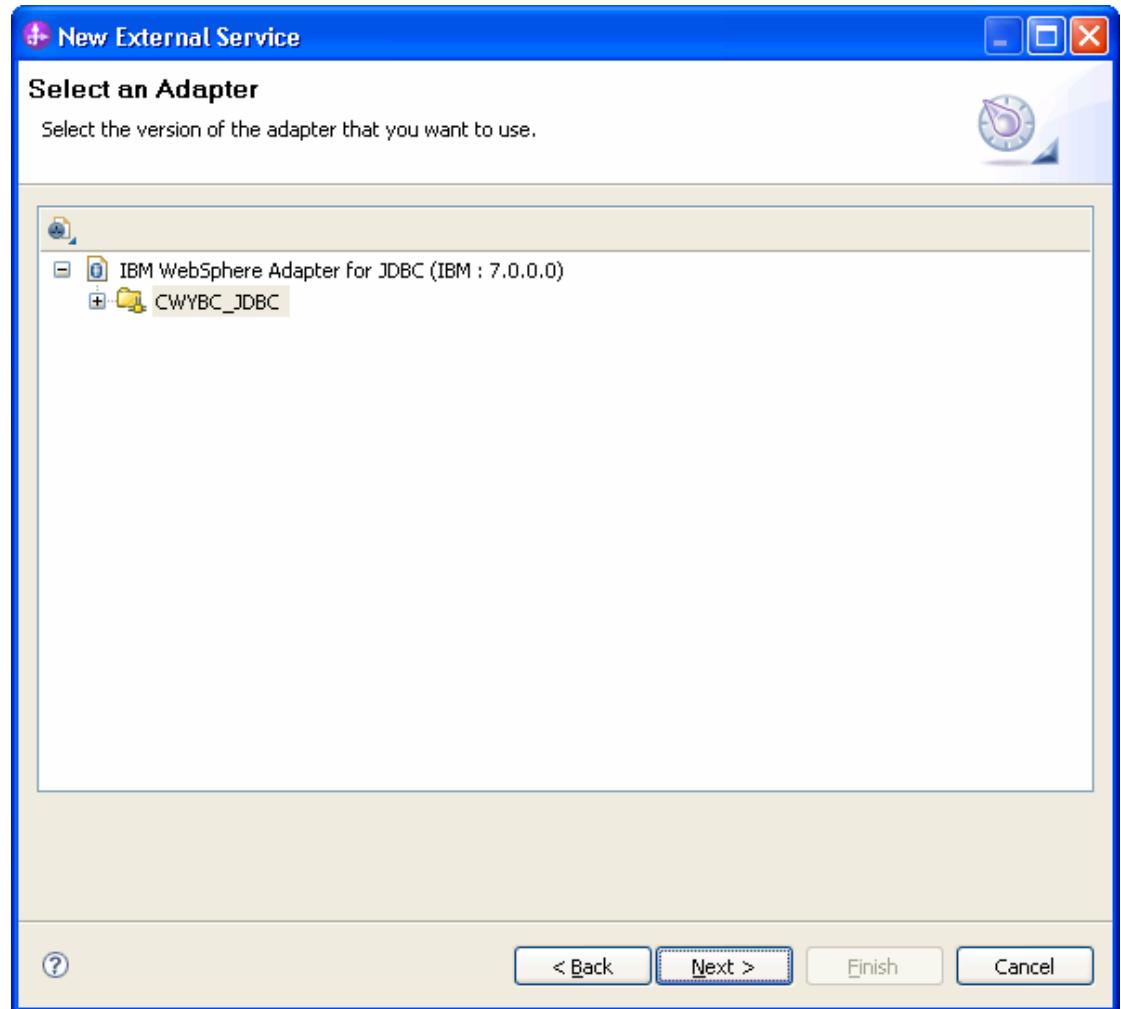
Configure the adapter for outbound processing

Run the external service wizard to specify business objects, services, and configuration details.

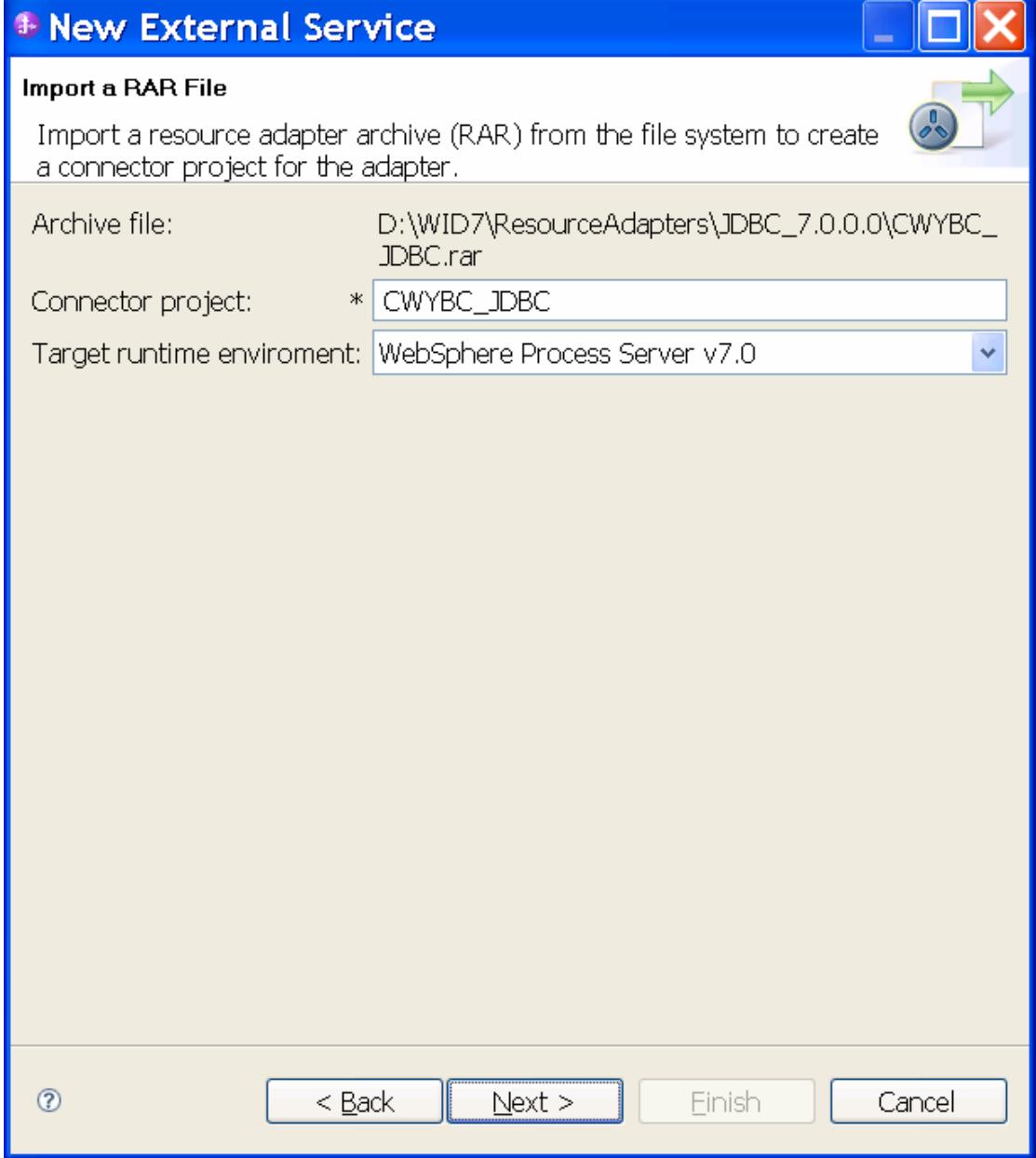
1. Switch to the Business Integration Perspective in WebSphere Integration Developer by selecting **Window -> Open Perspective Business Integration**.
2. Start the external service wizard by selecting **File-> New -> External Service**.
3. In the **Available Types** area, select **Adapters > JDBC** and then click **Next**.



4. Select the **IBM WebSphere Adapter for JDBC (IBM: 7.0.0.0)** and click **Next**.



5. In the **Connector project** field enter **CWYBC_JDBC**, and in the **Target runtime environment** field, select the appropriate runtime. Click **Next**.



New External Service

Import a RAR File

Import a resource adapter archive (RAR) from the file system to create a connector project for the adapter.



Archive file: D:\WID7\ResourceAdapters\JDBC_7.0.0.0\CWYBC_JDBC.rar

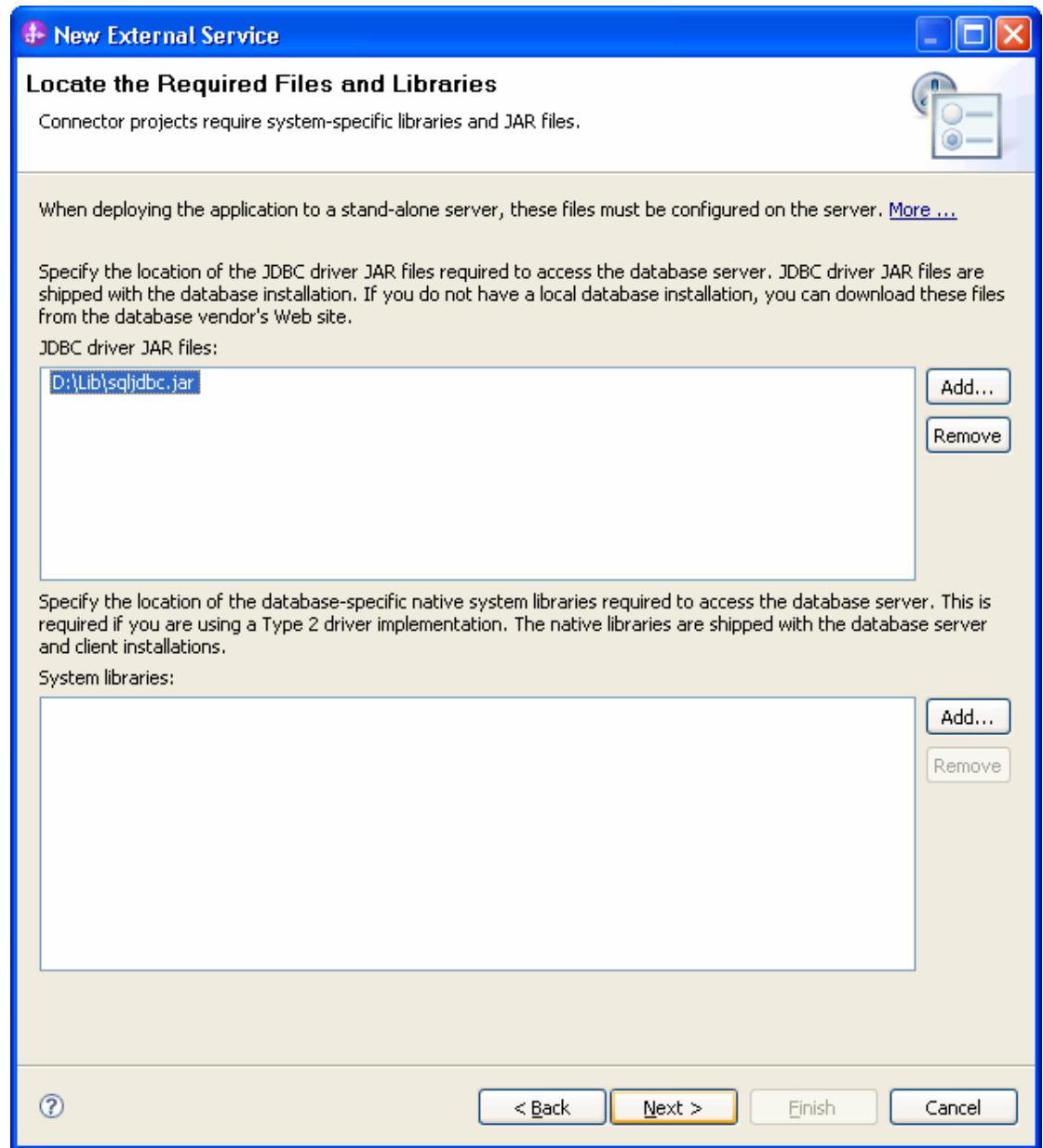
Connector project: * CWYBC_JDBC

Target runtime environment: WebSphere Process Server v7.0

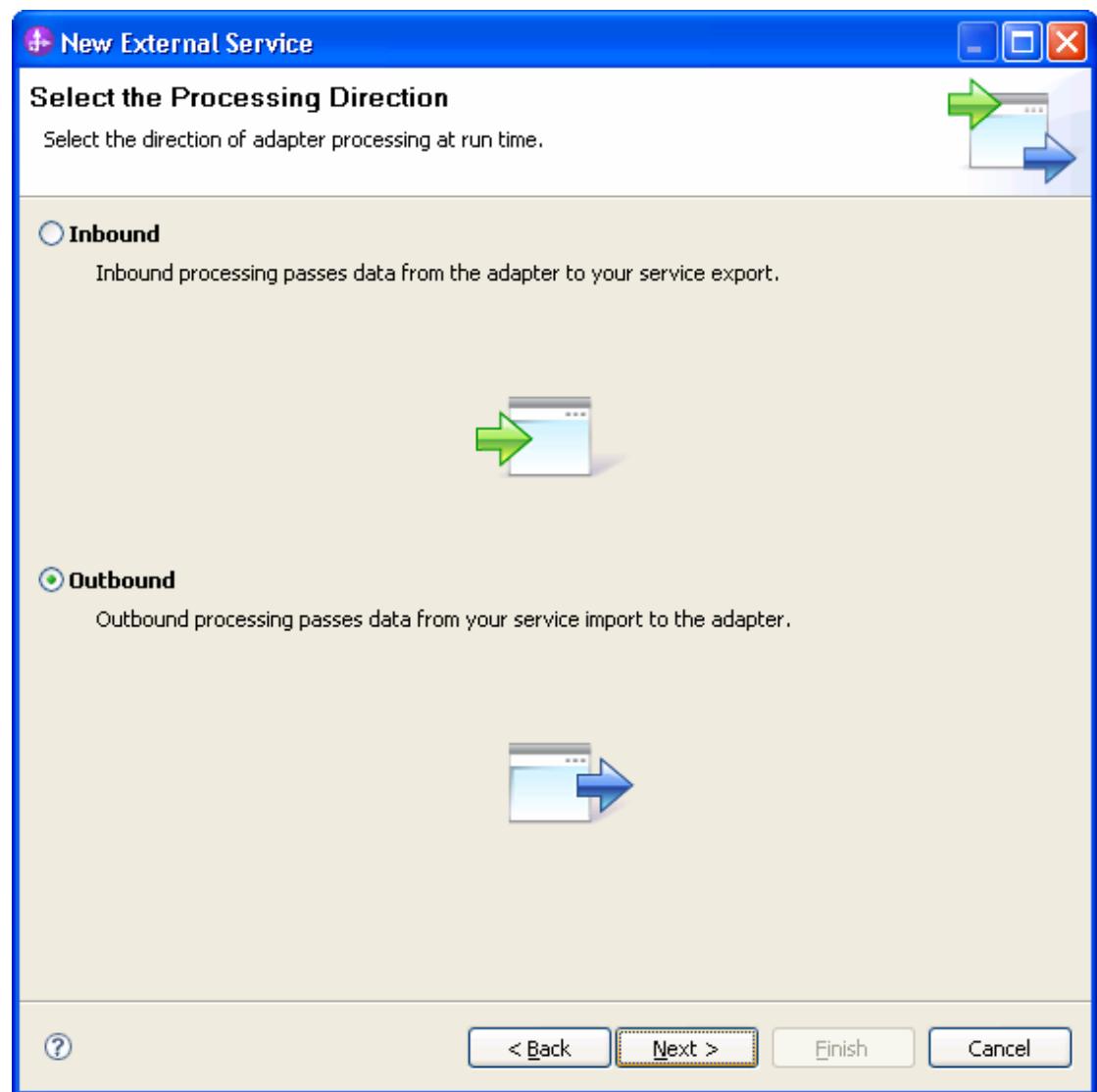
[?](#)[< Back](#)[Next >](#)[Finish](#)[Cancel](#)

WebSphere software

6. In the **JDBC driver JAR files** field, click **Add**, to add the JDBC driver class to connect to the database. Browse to select the driver JAR file and click **Next**.



7. Select **Outbound** and click **Next**.

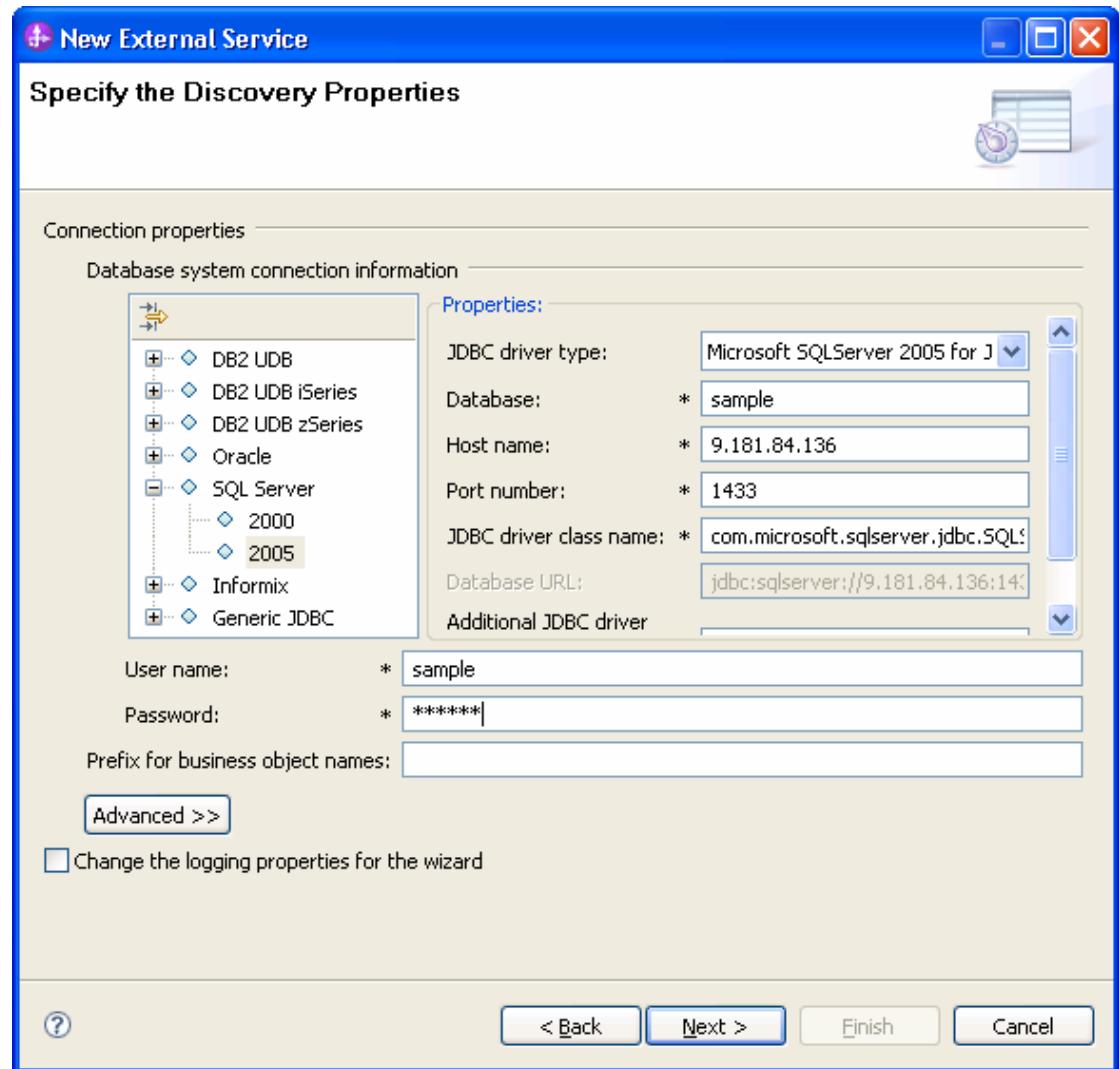


Set connection properties for the external service wizard

To connect to the SQL Server:

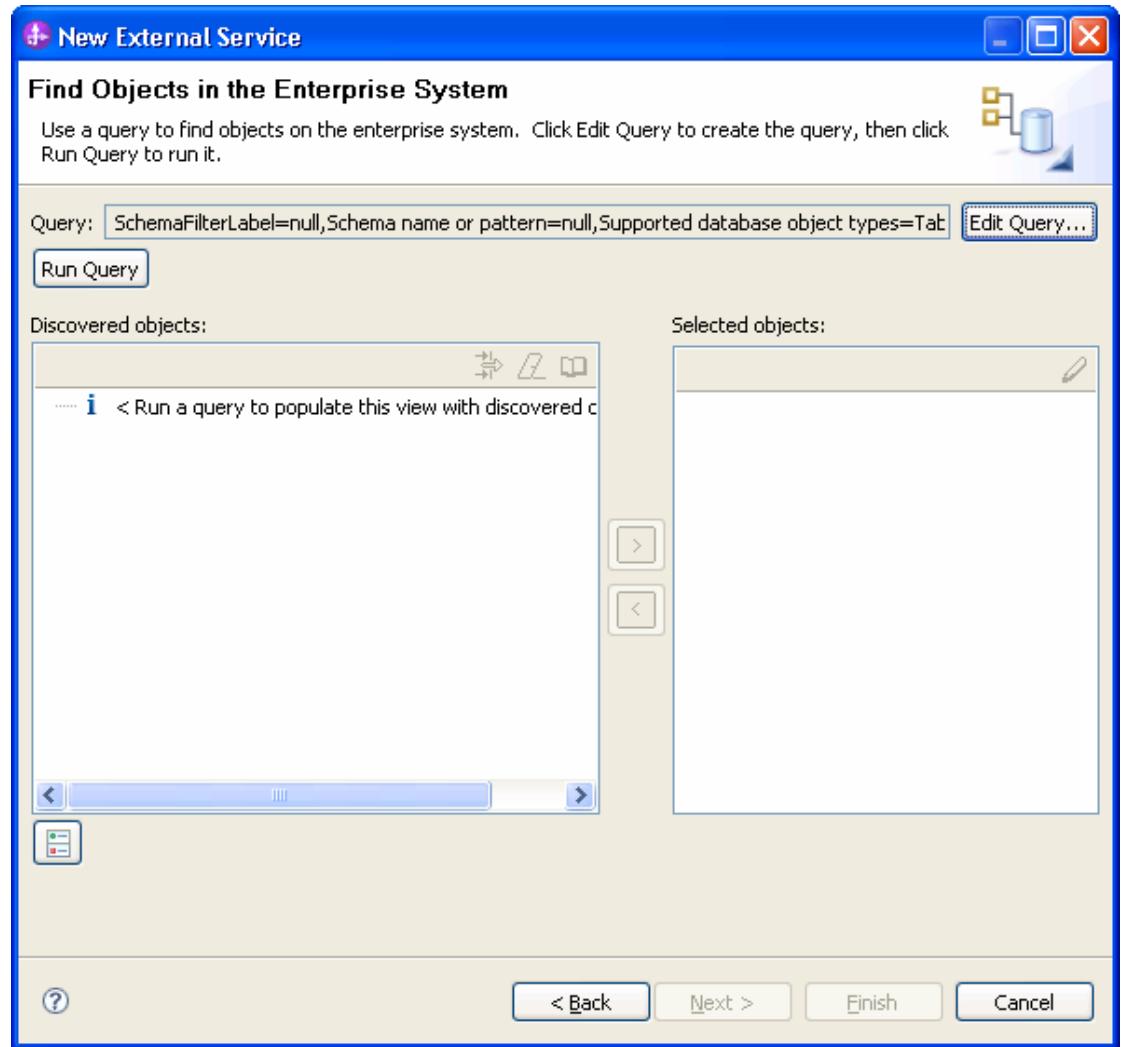
WebSphere software

1. Expand the **SQL Server** node in the **Database system connection information** area and select **2005**.
2. Enter **Database**, **Host name**, **Port number**, **User name** and **Password** fields, and click **Next**.

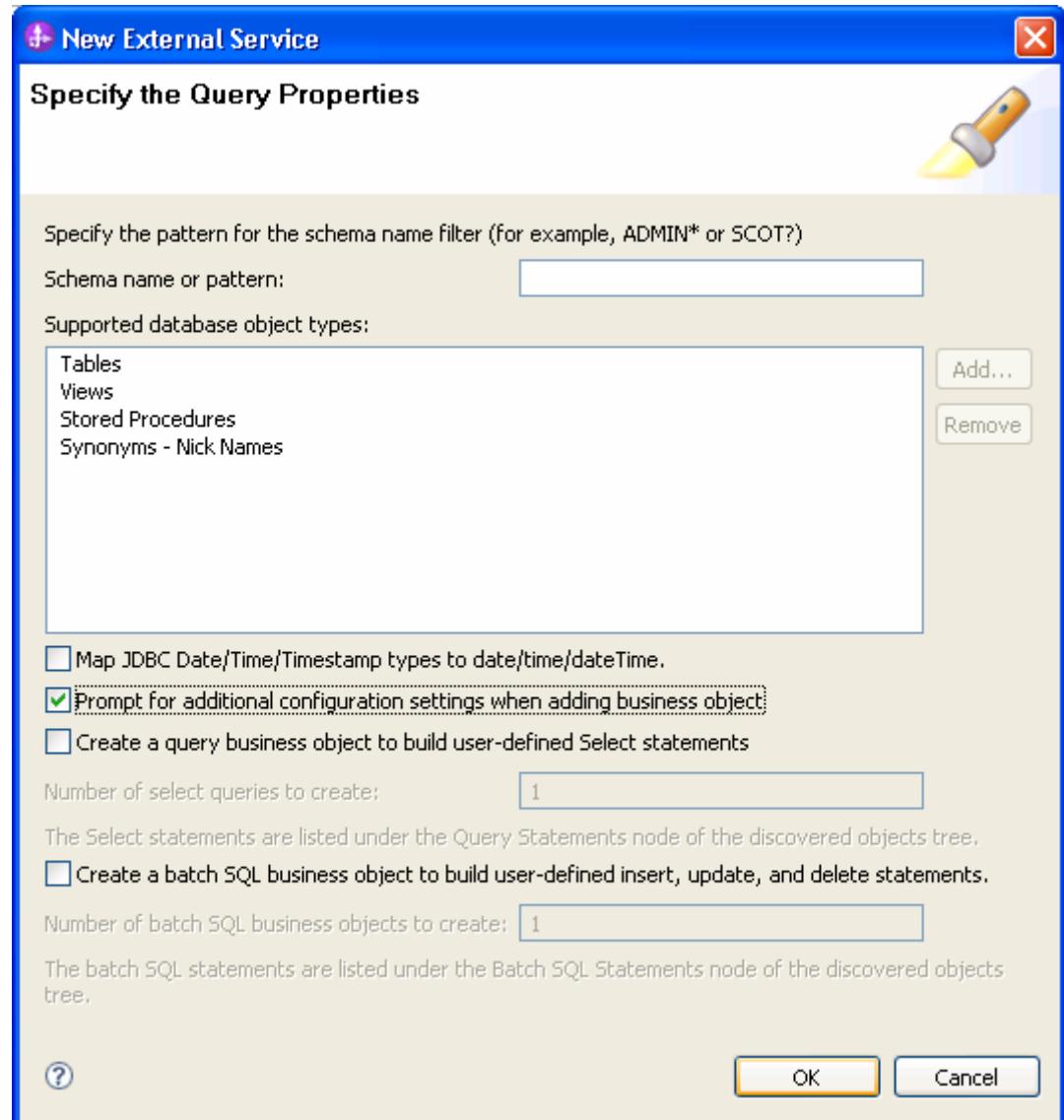


Select the business objects and services to be used with the adapter

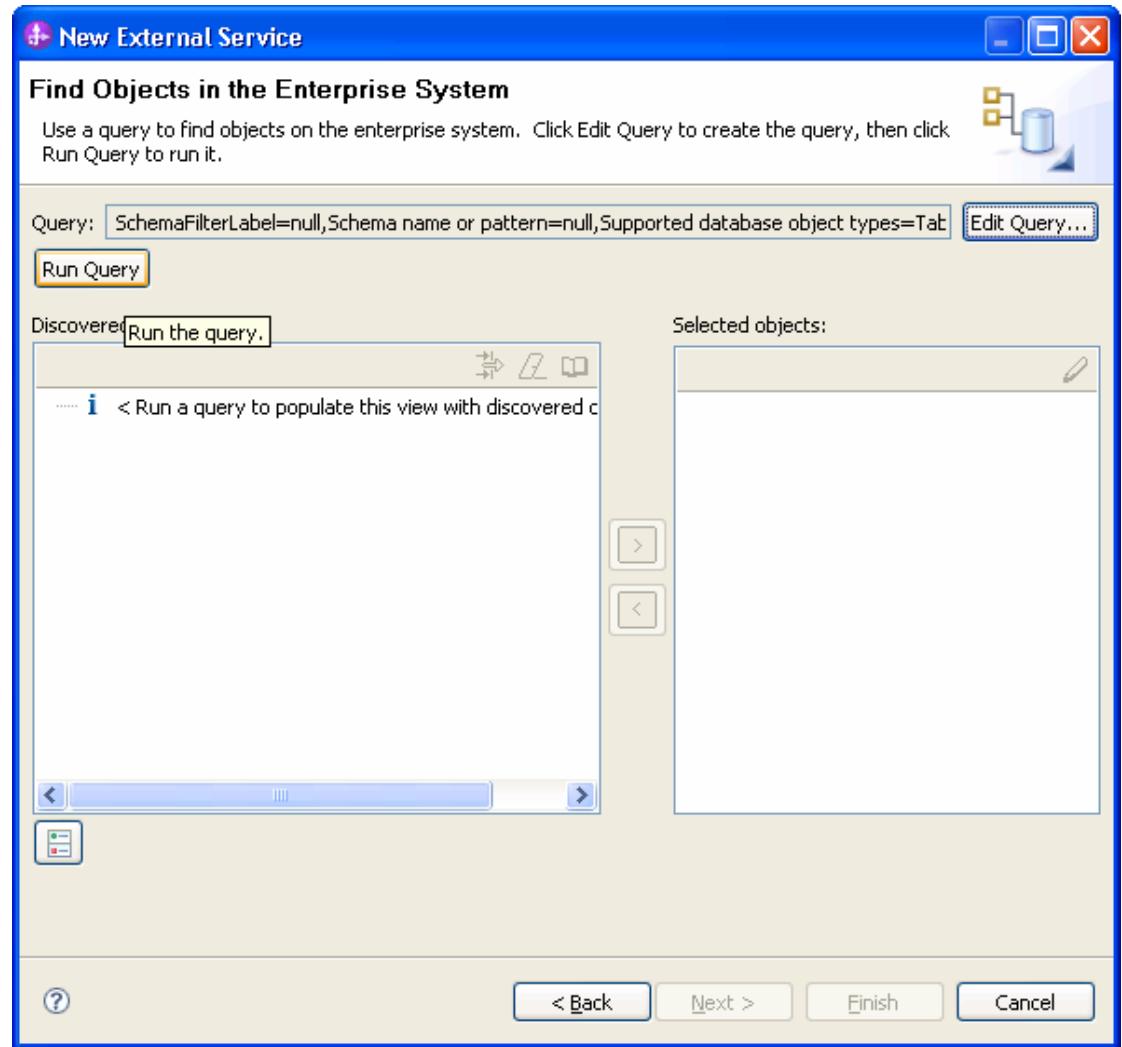
1. Find Objects in Enterprise System window, click **Edit Query**.



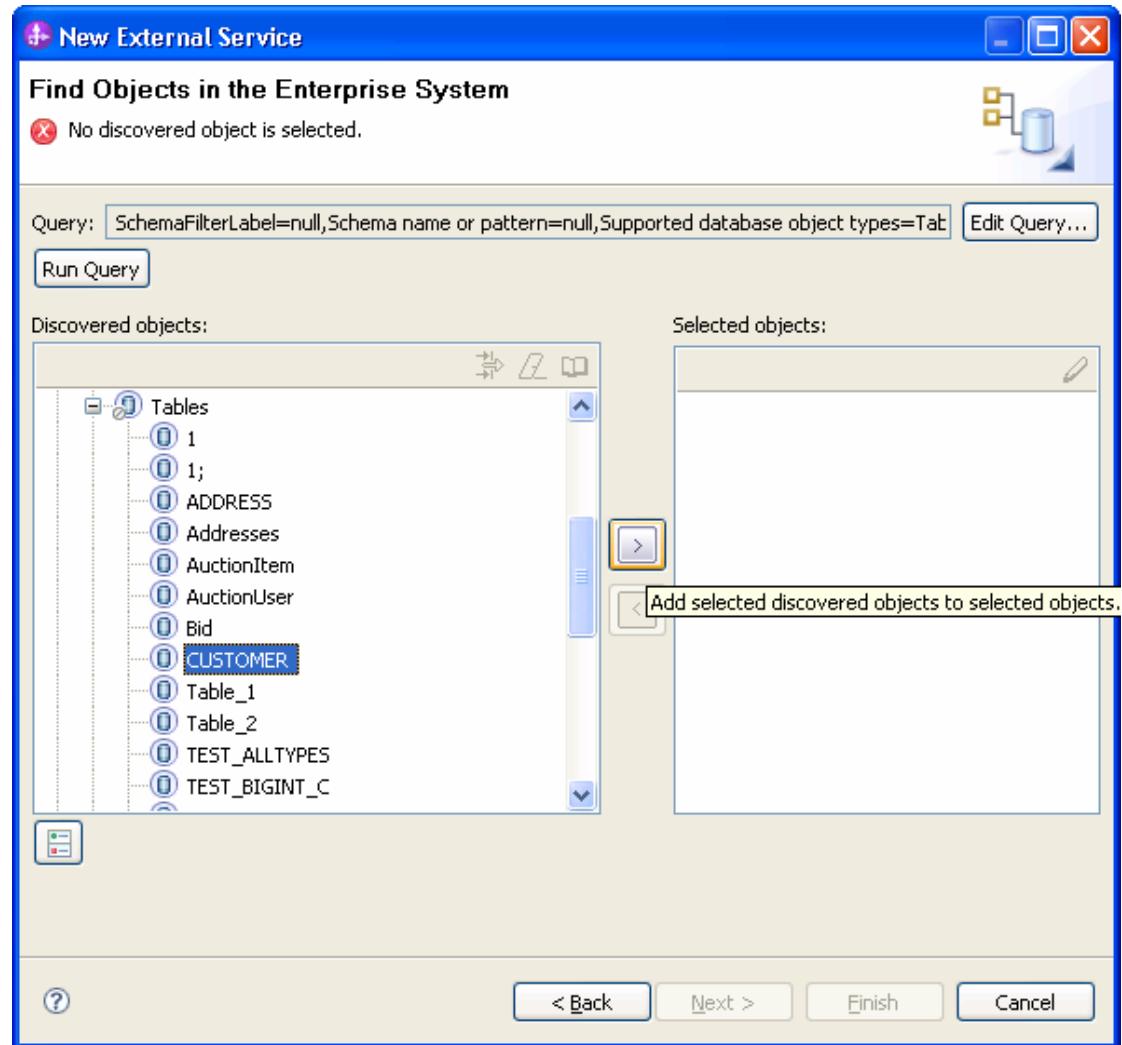
2. In the Specify the Query Properties window, select the **Prompt for additional configuration settings when adding business objects** check box and click **OK**.



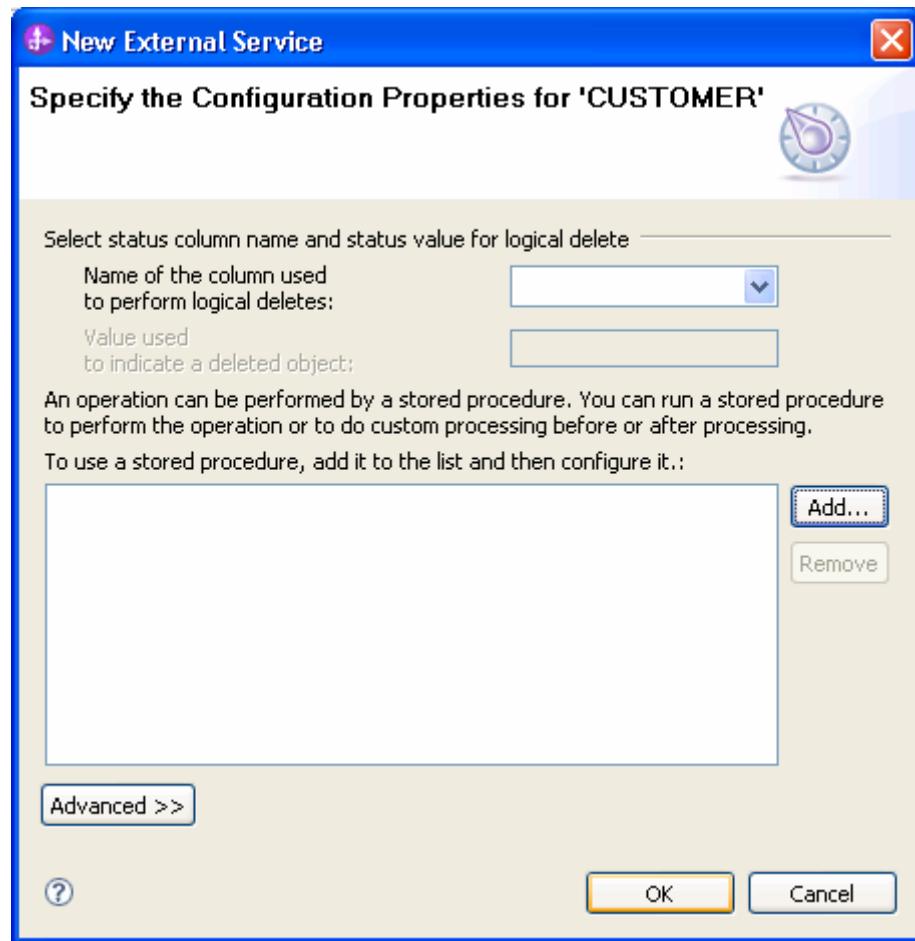
3. Click **Run Query**.



4. Expand the **dbo** (for this tutorial only) node, select **Tables** and expand it.



5. Select the **CUSTOMER** table and click . In the Specify the Configuration Properties for 'CUSTOMER' window click **OK**.



6. Select the ADDRESS table and click 
7. In the Specify the Configuration Properties for 'ADDRESS' window, select **CUSTOMER (dbo)** from the **Choose parent table** list, and then select **PKEY** for **CUSTID** in the **Build a foreign key** area. Select the **Parent object owns child object(cascade delete)** check box. Click **Add**.

New External Service

Specify the Configuration Properties for 'ADDRESS'

Select status column name and status value for logical delete

Name of the column used to perform logical deletes:

Value used to indicate a deleted object:

Choose parent table from the list for the selected child

Choose parent table: CUSTOMER (dbo)

Single cardinality

Build a foreign key relationship by selecting a parent table column for each child column

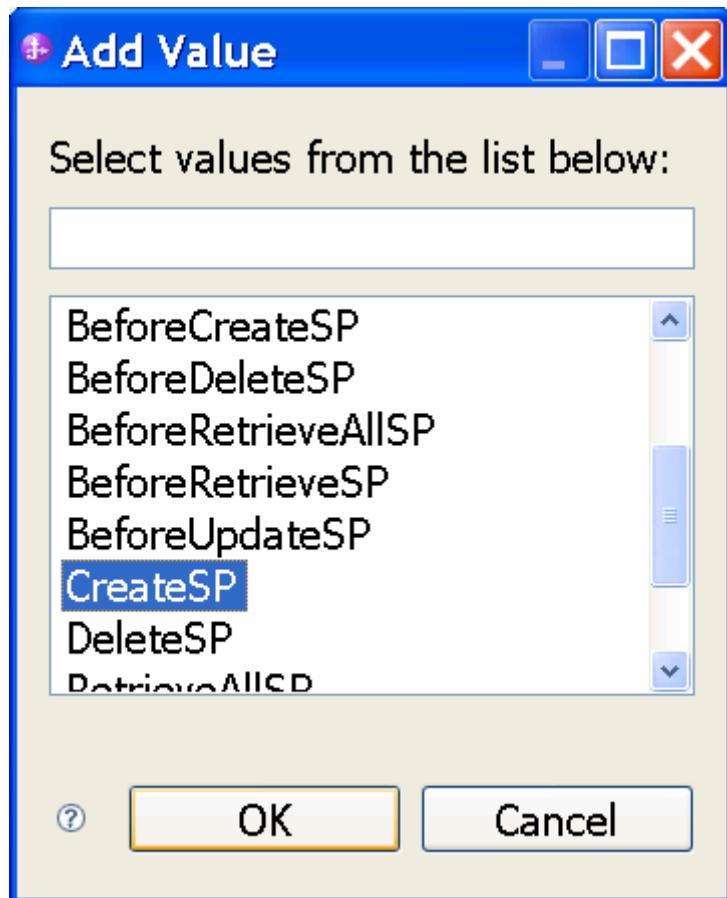
ADDRID:	<input type="text" value="NONE"/>
CUSTID:	<input type="text" value="PKEY"/>
CITY:	<input type="text" value="NONE"/>
ZIPCODE:	<input type="text" value="NONE"/>

Parent object owns child object (cascade delete)
 Preserves ADDRESS when the parent is updated.
 ADDRESS required for operations on parent

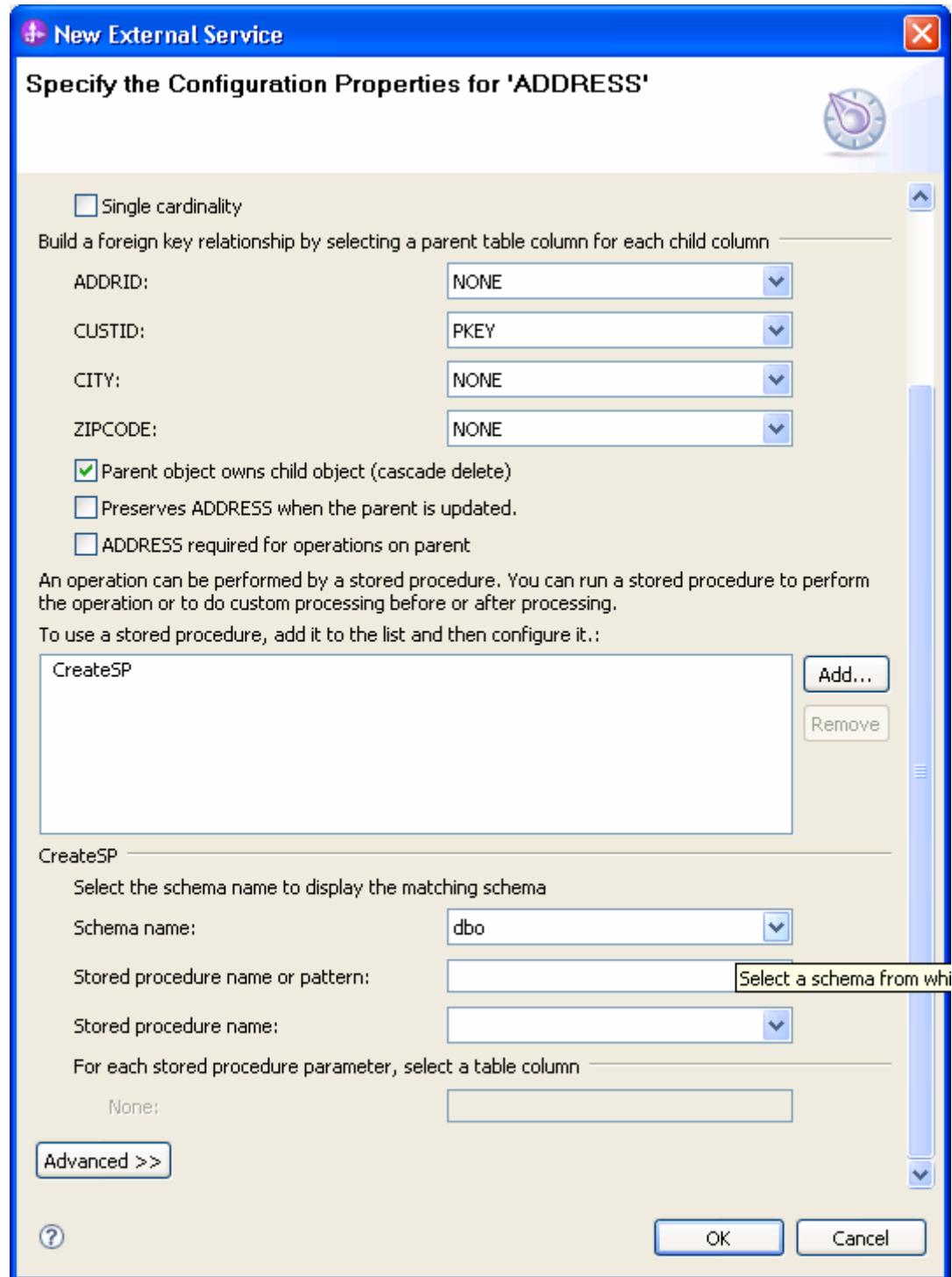
An operation can be performed by a stored procedure. You can run a stored procedure to perform the operation or to do custom processing before or after processing.

To use a stored procedure, add it to the list and then configure it.:

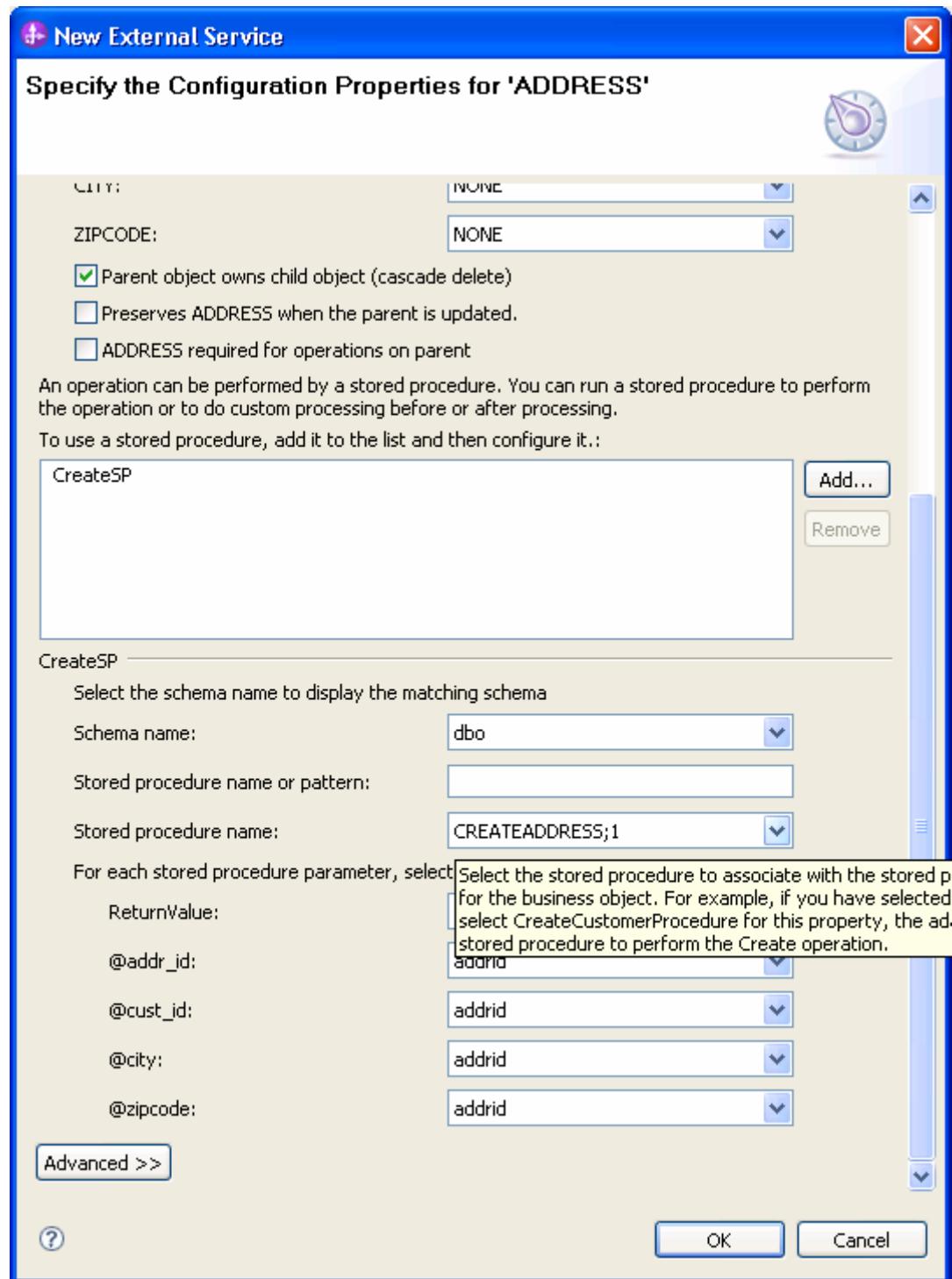
8. Select **CreateSP** and click **OK**.



9. Select **dbo** for the schema name.



10. Select **CREATEADDRESS;1** from stored procedure name list.



11. Select stored procedure parameter for each column.

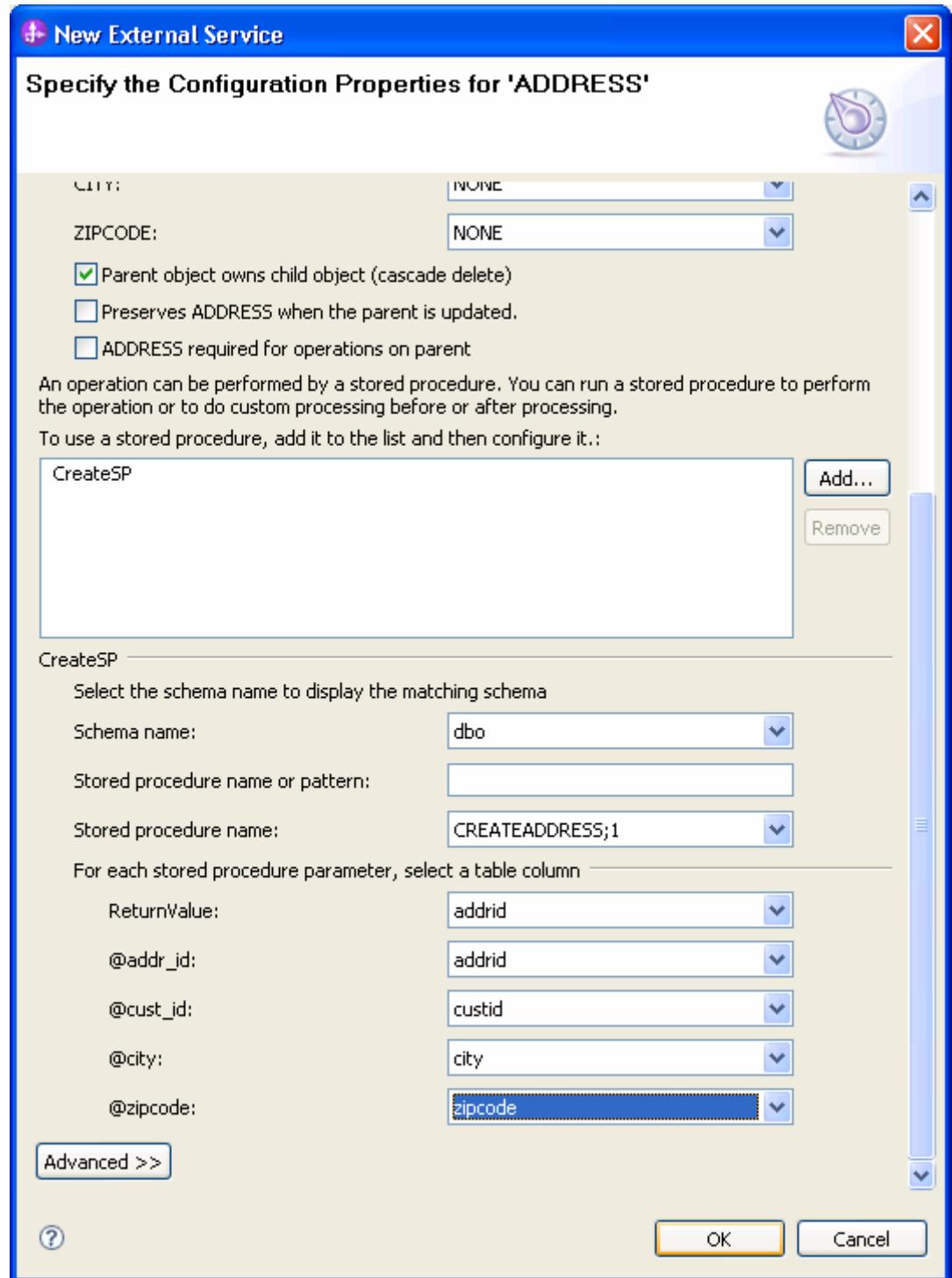
@addr_id: addr

@cust_id: custid

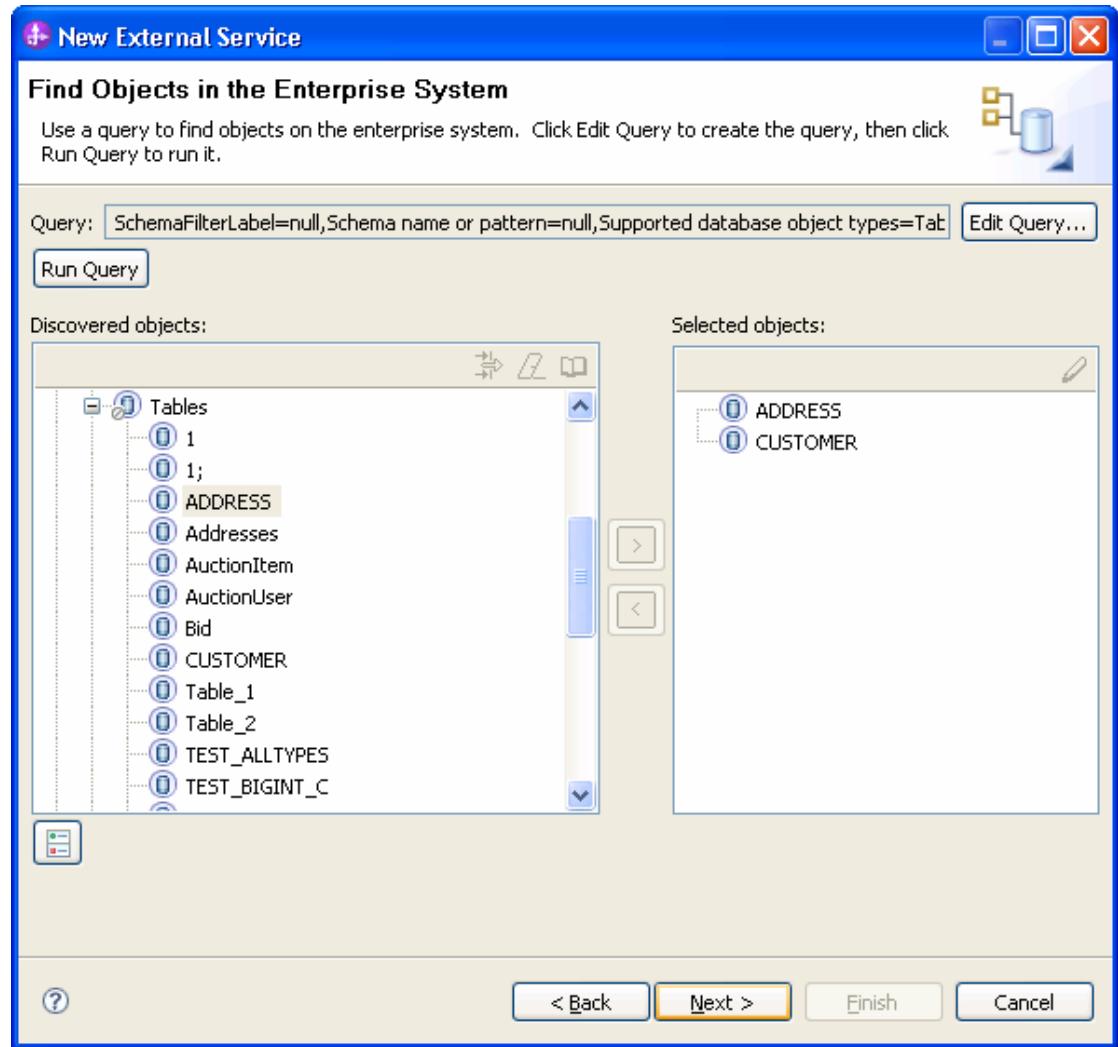
@city: city

@zipcode: zipcode

12. Click **OK**.



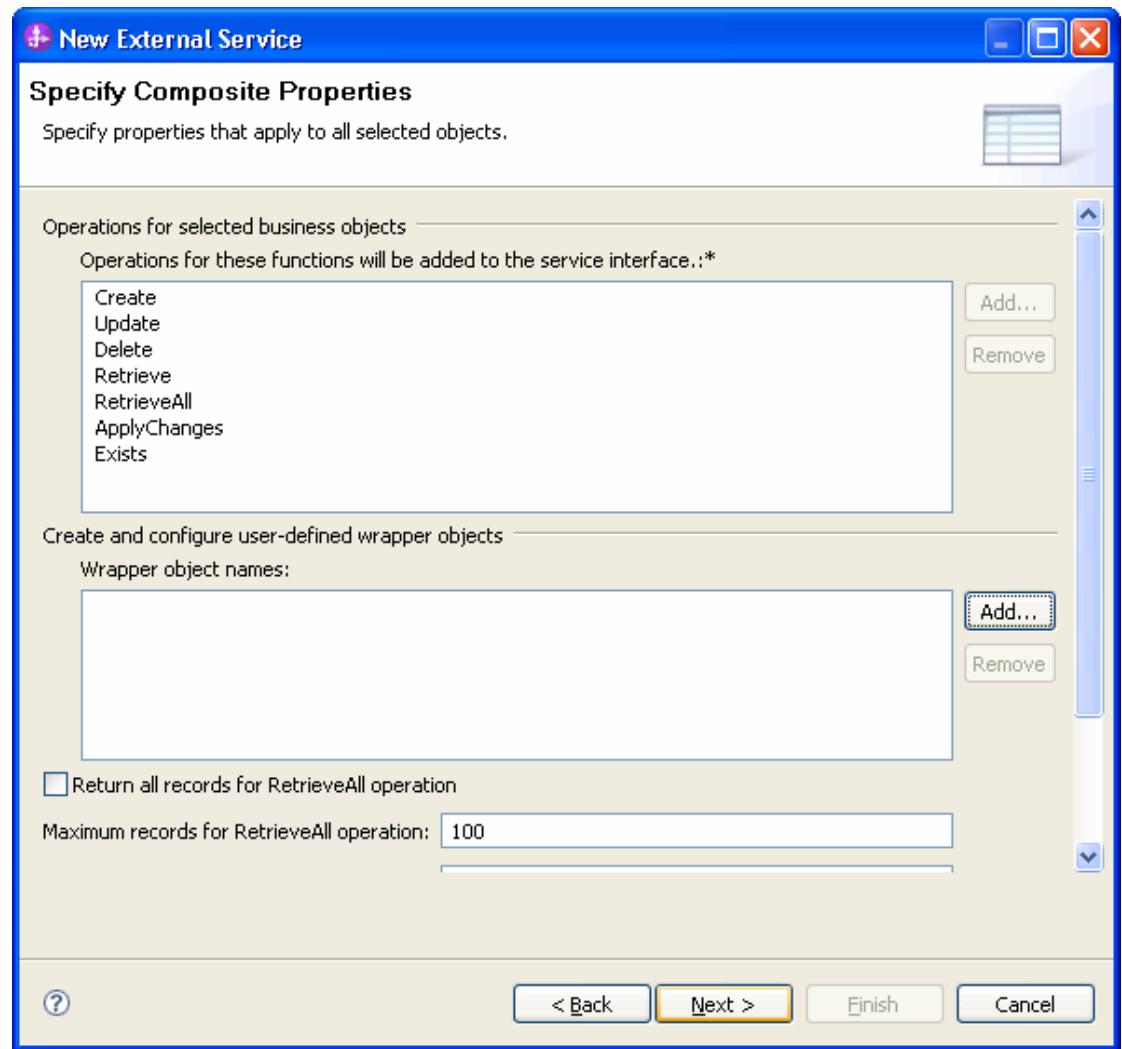
13. In the Find Objects in Enterprise System window, click **Next**.



Generate business object definitions and related artifacts

Follow these steps to generate the business object definitions.

1. In the Specify Composite Properties window, accept the default values for all fields and click **Next**.



2. In the Specify the Service Generation and Deployment Properties window, perform the following steps:
 - a) Select **Other** for security options under **Deployment Properties**.
 - b) Clear the **Join the global transaction** check box.
 - c) Select **Specify predefined connection pool DataSource** from the **Database connection information** list.
 - d) Enter **SQLServerDS** in the **Connection pool DataSource JNDI Name** field, and click **Next**.

New External Service

Specify the Service Generation and Deployment Properties

Specify properties for generating the service and running it on the server.

Service Operations

To modify the names, or add a description to the operations to be generated in the interface file, click Edit Operations.

[Edit Operations...](#)

Deployment Properties

How do you want to specify the security credentials?

Using an existing JAAS alias (recommended)
A Java Authentication and Authorization Services (JAAS) alias is the preferred method.
J2C authentication data entry:

Using security properties from the managed connection factory
The properties will be stored as plain text; no encryption is used.
User name:
Password:

Other
Use if no security is required or will be handled by the EIS system, or the RAR will be deployed on the server and security will be specified by the properties in the JNDI lookup name.

The quality of service that is used to join the transaction provides a higher degree of data integrity, especially when a failure occurs. To participate in a global transaction, a predefined XA DataSource or XA database connection information must be specified in the connection properties. [More ...](#)

Join the global transaction

Deploy connector project:

Specify the settings used to connect to JDBC at run time:

Connection settings:

Connection Properties

To join a global transaction, specify a predefined XA datasource or XA database connection information. When not joining a global transaction, either the XA connection information or the local connection information can be specified.

Database connection information:

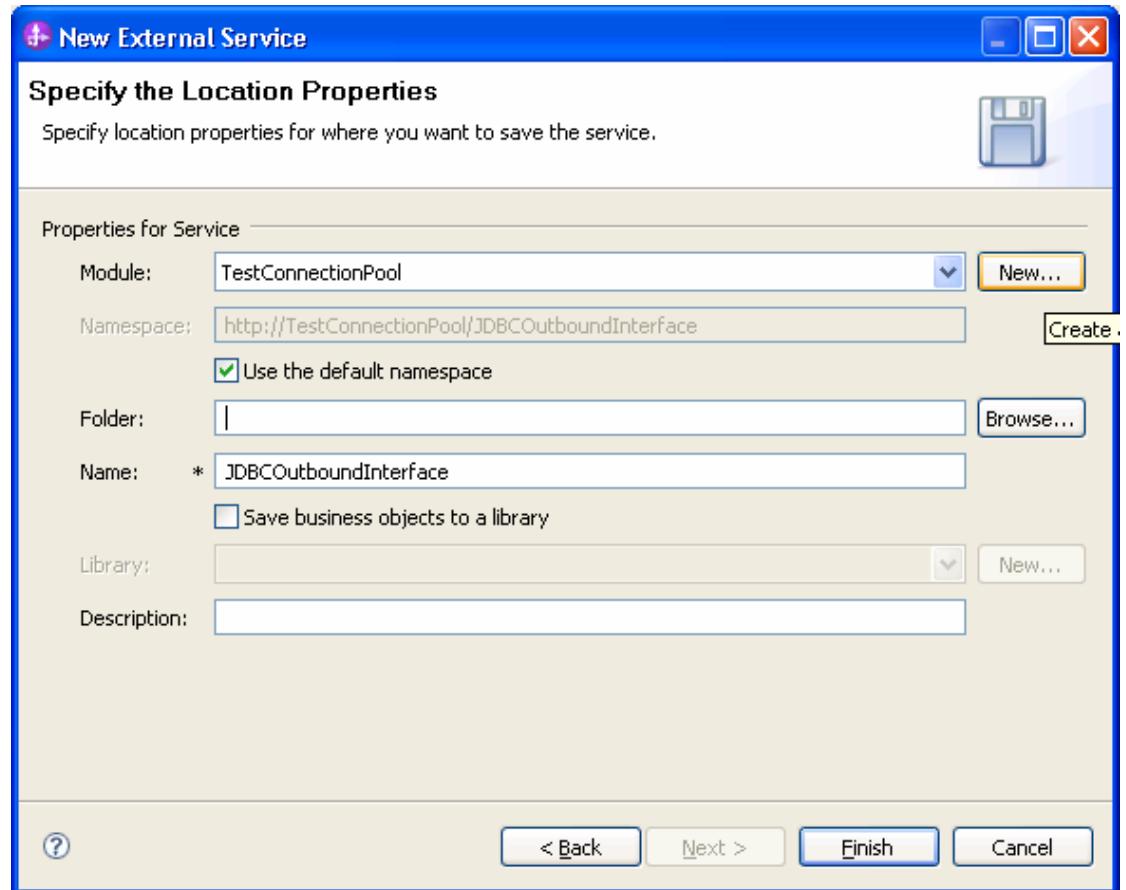
Database system connection information

Database vendor:
Connection pool DataSource JNDI name:

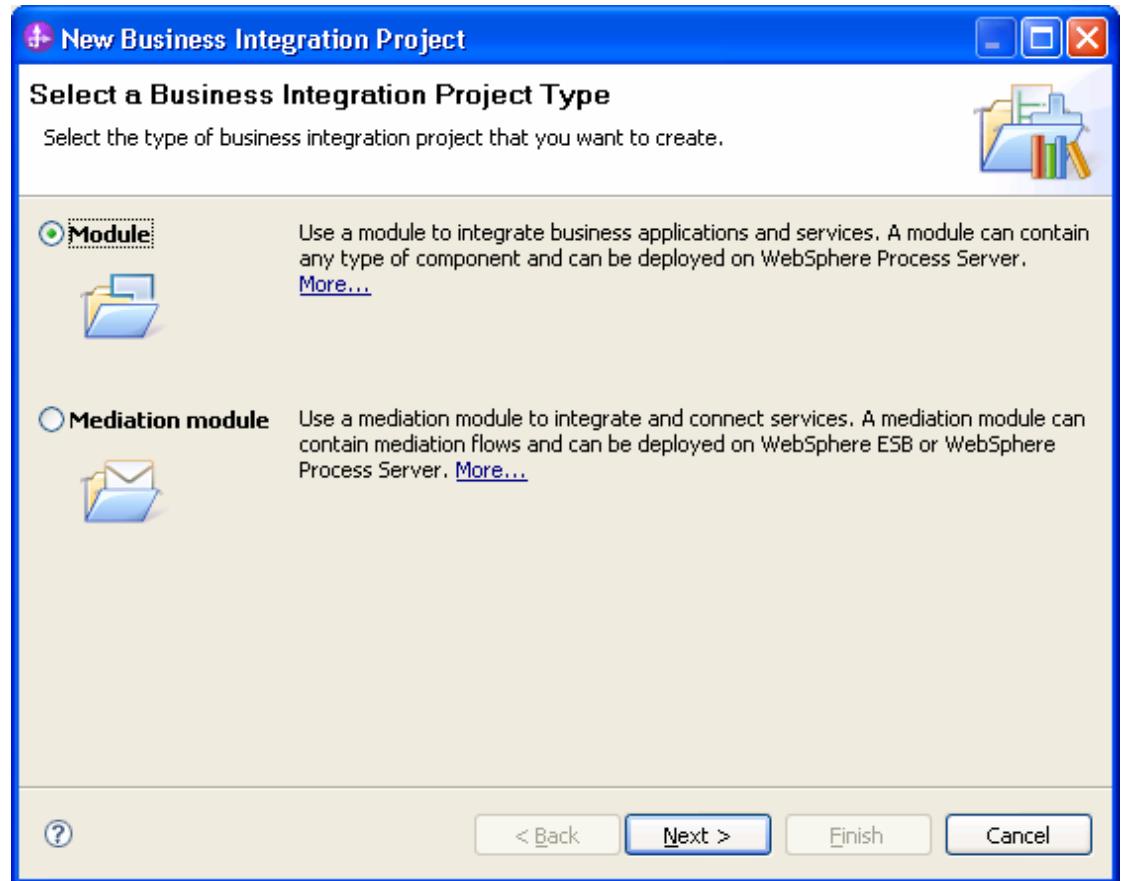
[Advanced >>](#)

[?](#) < Back **Next >** Finish Cancel

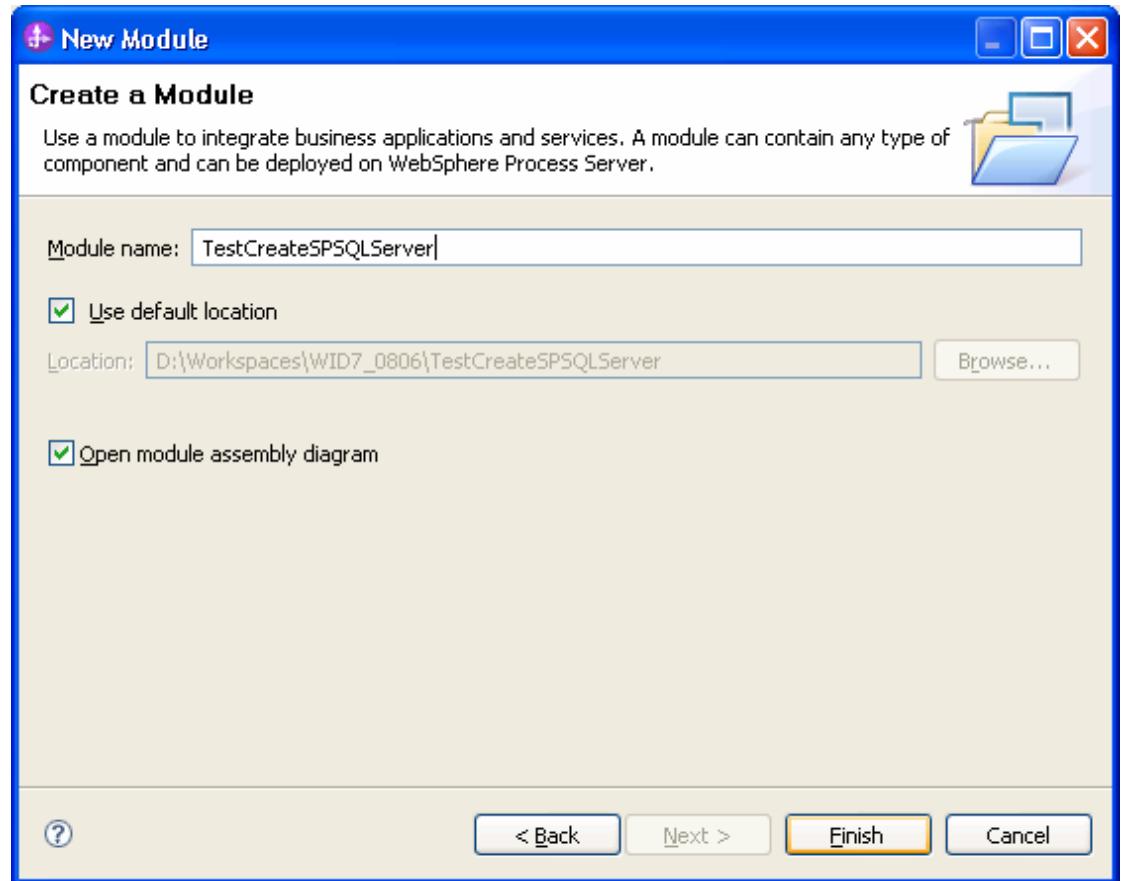
- Click **New** in the Specify the Location Properties window.



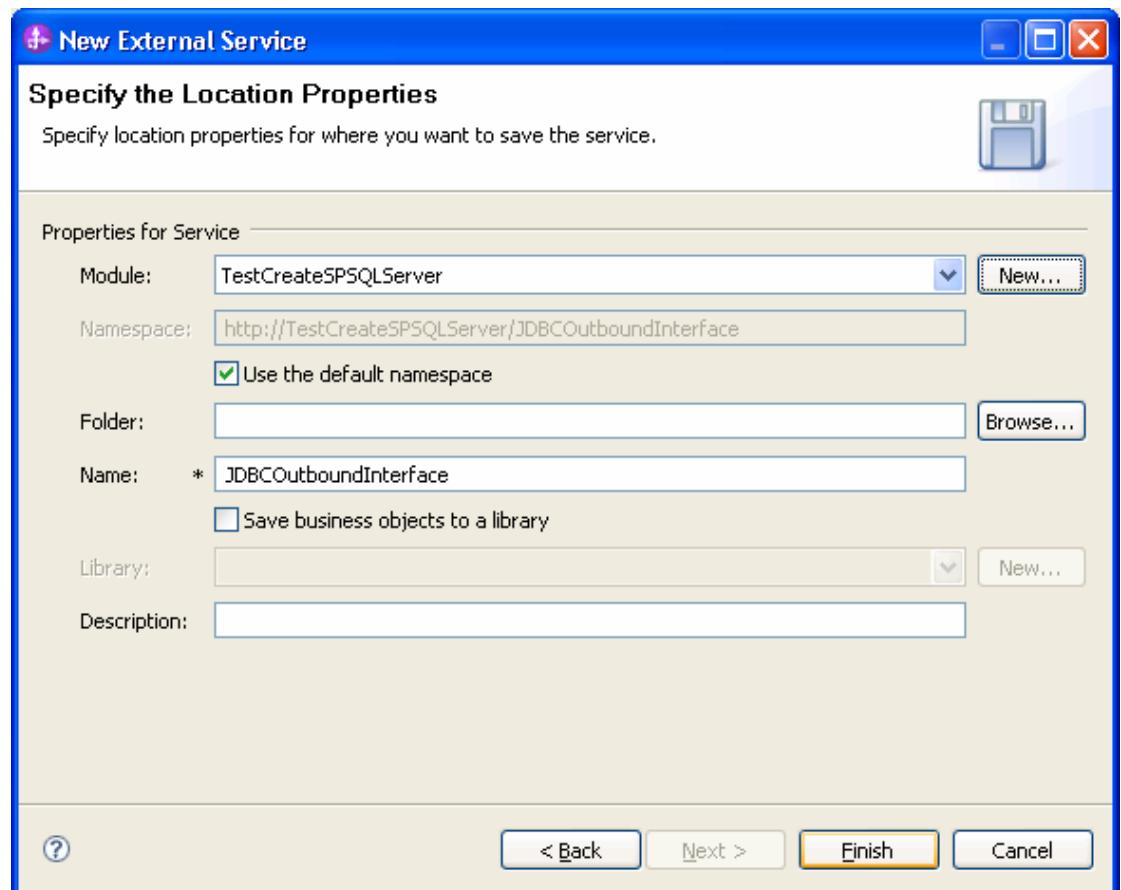
4. In the Select a Business Integration Project Type window, select **Module** and click **Next**.



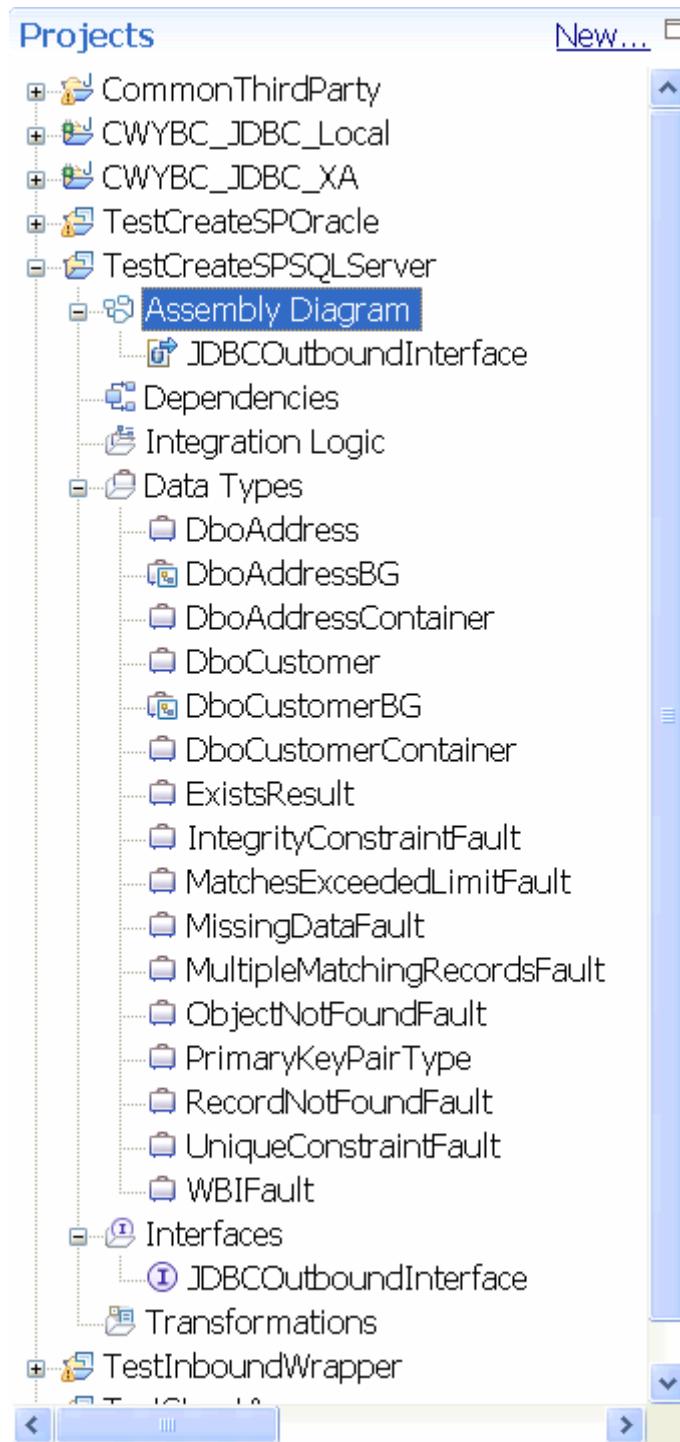
5. In the Create a Module window, type **TestCreateSPSQLServer** in the **Module Name** field and click **Finish**.



6. Click **Finish** to complete service creation.



7. Expand the created Business Integration Project and verify whether the artifacts are generated correctly.



Deploy the module to the test environment

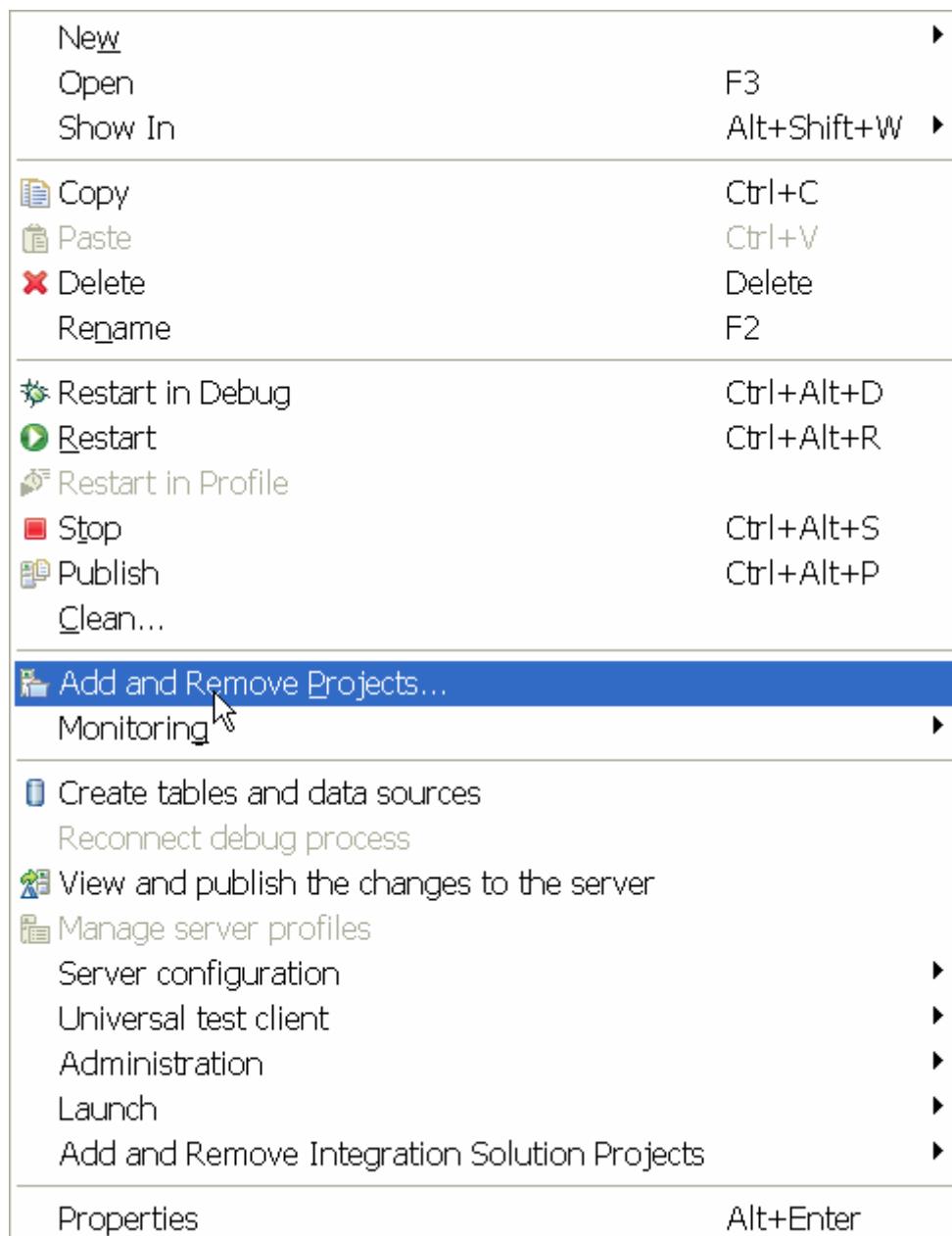
The result of running the external service wizard is a Service Component Architecture (SCA) module that contains an Enterprise Information System import. You must install this SCA module in the WebSphere Integration Developer integration test client. To do this,

WebSphere software

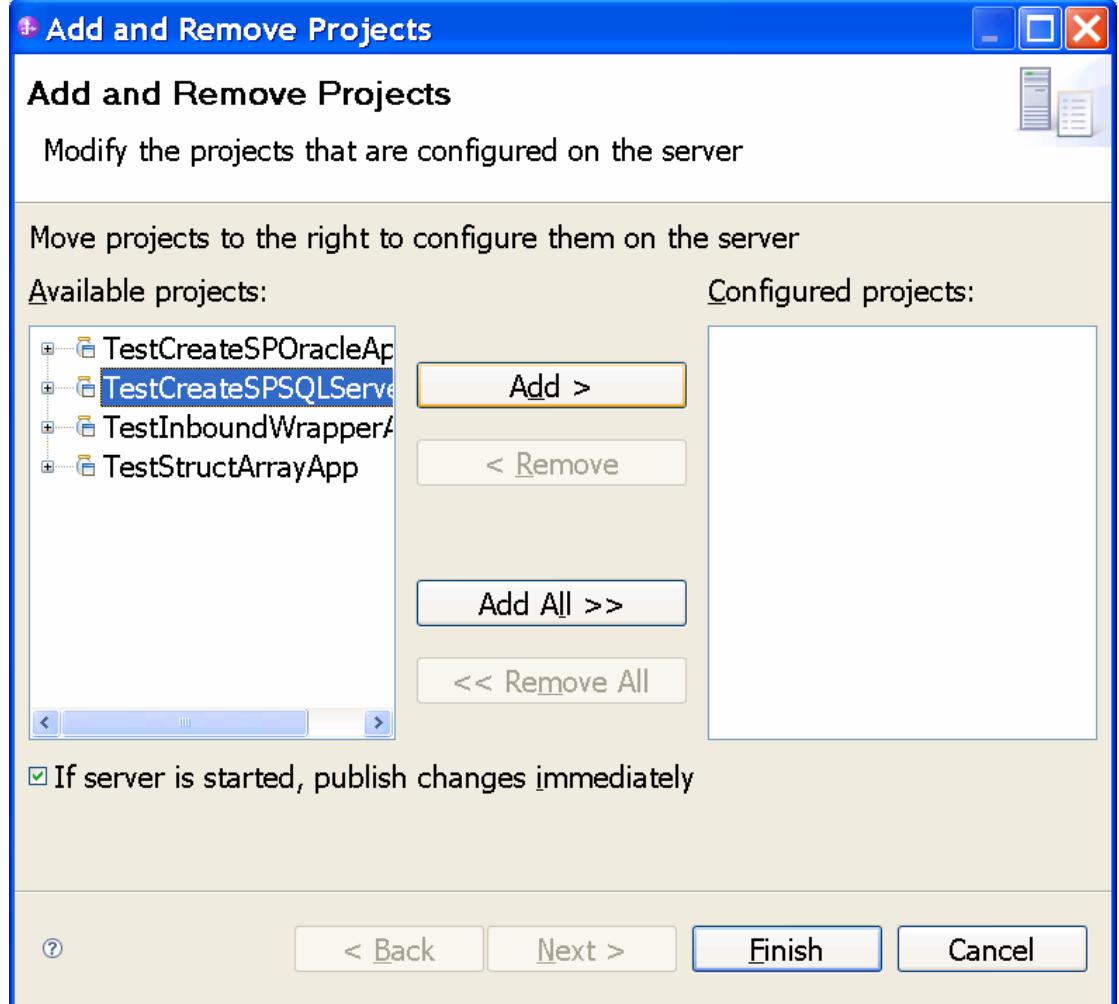
you must add the SCA module you created earlier to the server using the **Servers** view in WebSphere Integration Developer.

Steps for adding the SCA module to the server:

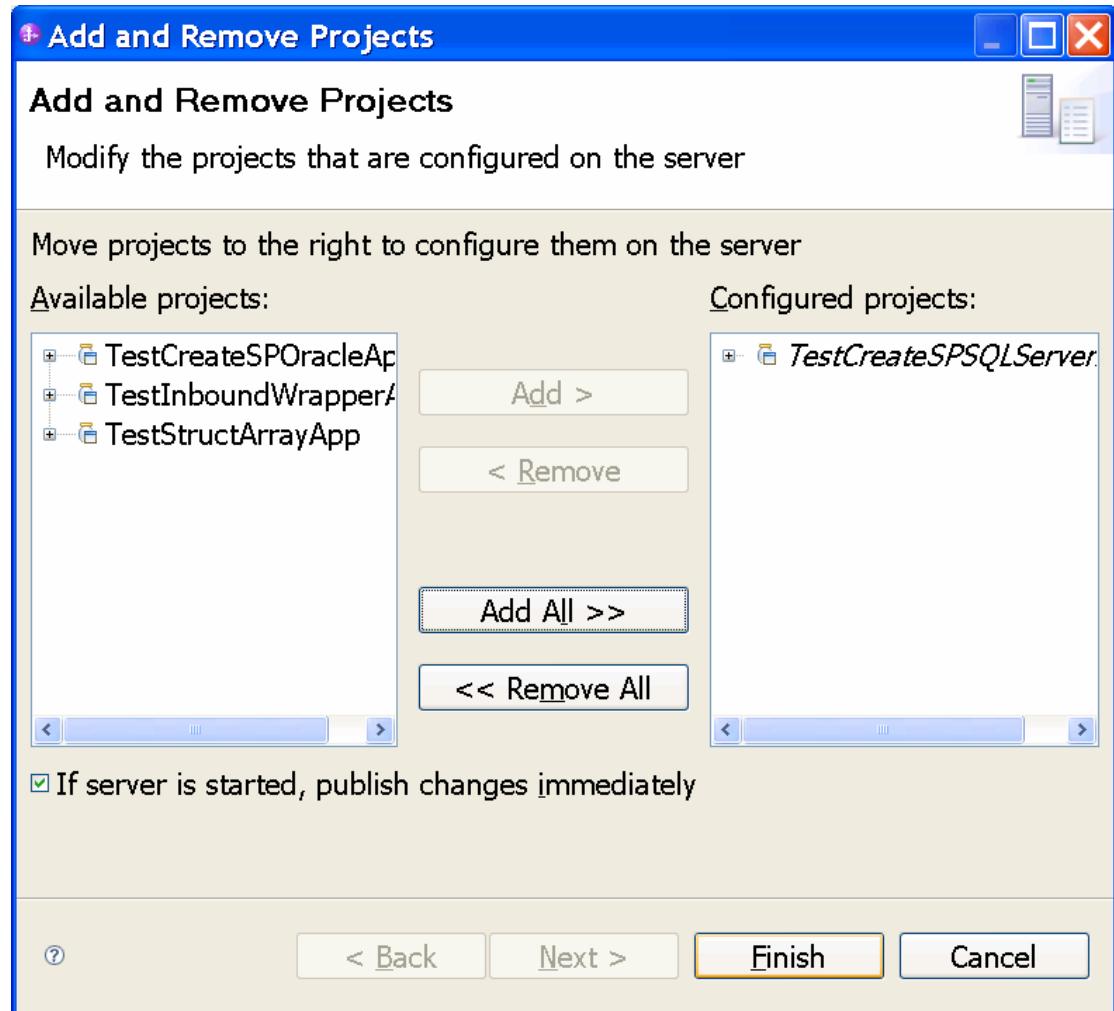
1. In WebSphere Integration Developer, switch to the **Servers** view by selecting **Windows > Show View > Servers**.
2. In the Servers tab in the lower-right pane of the WebSphere Integration Developer screen, right-click the server, and select **Start**.
3. After the server is started, right-click the server, and select **Add and Remove projects**.



The Add and Remove Projects window lists the available projects in the WebSphere Integration Developer workspace.



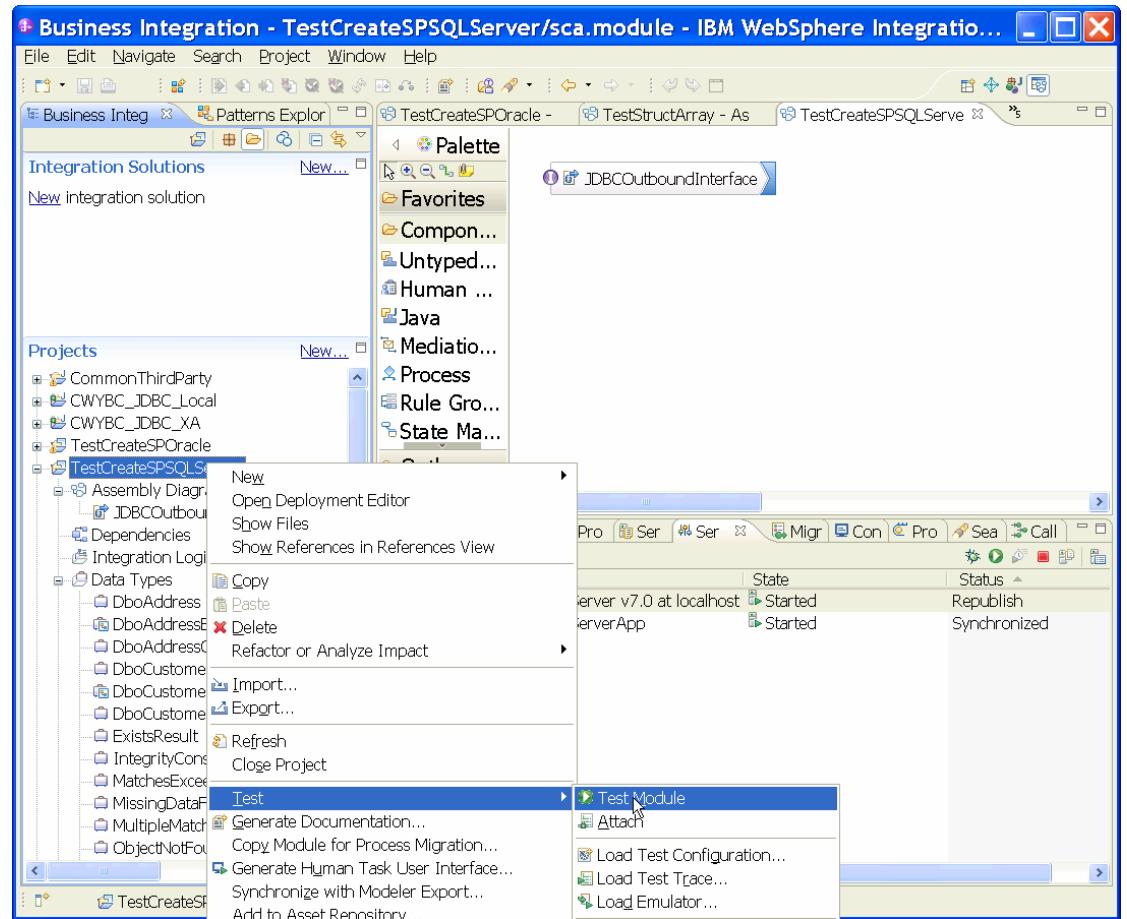
4. Select your project (**TestCreateSPSQLServerApp**) and click **Add** to configure the project on the server. Click **Finish**.



Test the assembled adapter application

Test the assembled adapter application using the WebSphere Integration Developer integration test client.

1. Select the **TestCreateSPSQLServer** module, right-click and select **Test > Test Module**. The Test Client window is displayed.



2. Select **createDboCustomerBG** from the **Operation** list.

WebSphere software

General Properties

Detailed Properties

Specify the component, interface, operation, and input parameter values for the Invoke event, then click the Continue icon in the Events area to run the test. [More...](#)

<u>Configuration:</u>	Default Module Test
<u>Module:</u>	TestCreateSPSQLServer
<u>Component:</u>	JDBCOutboundInterface
<u>Interface:</u>	JDBCOutboundInterface
<u>Operation:</u>	createDboCustomerBG

Initial request parameters:

Value editor XML editor

Name	Type	Value
createDb...	DboCustomerBG	✓
verb	verb<string>	✓ Create
DboCust...	DboCustomer	✓
pkey	string	✓
fname	string	✓
Iname	string	✓
ccode	string	✓
address	DboAddress[]	...

3. Enter **Create** for the verb and specify values for **pkey**, **Iname**, **fname** and **ccode** as shown in the figure.

WebSphere software

Initial request parameters:

Value editor XML editor

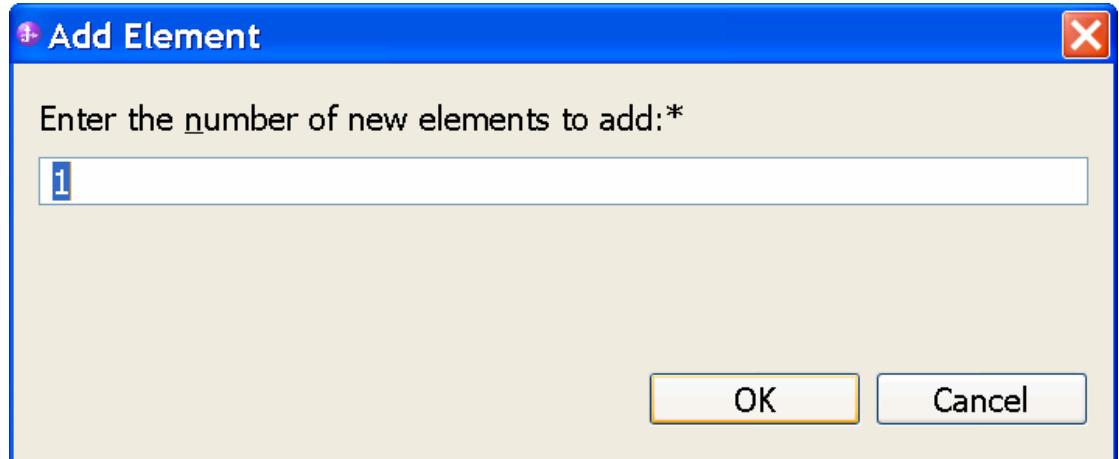
Name	Type	Value
createDboCustomer	DboCustomerBG	✓
verb	verb<string>	✓ Create
DboCustomer	DboCustomer	✓
pkey	string	✓ 100
fname	string	✓ testFname
lname	string	✓ testLname
ccode	string	✓ testCcode
address	DboAddress[]	[]

4. Right-click **addressobj** and select **Add Elements...**.

Initial request parameters:
 Value editor XML editor

Name	Type	Value
createDboCustomerE	DboCustomerBG	✓
verb	verb<string>	✓ Create
DboCustomer	DboCustomer	✓
pkey	string	✓ 100
fname	string	✓ testFname
lname	string	✓ testLname
ccode	string	✓ testCcode
address	DboAddress	[]

5. Enter 1 and click **OK**.



6. Enter the values for **addressobj[0]** as shown in the figure below.

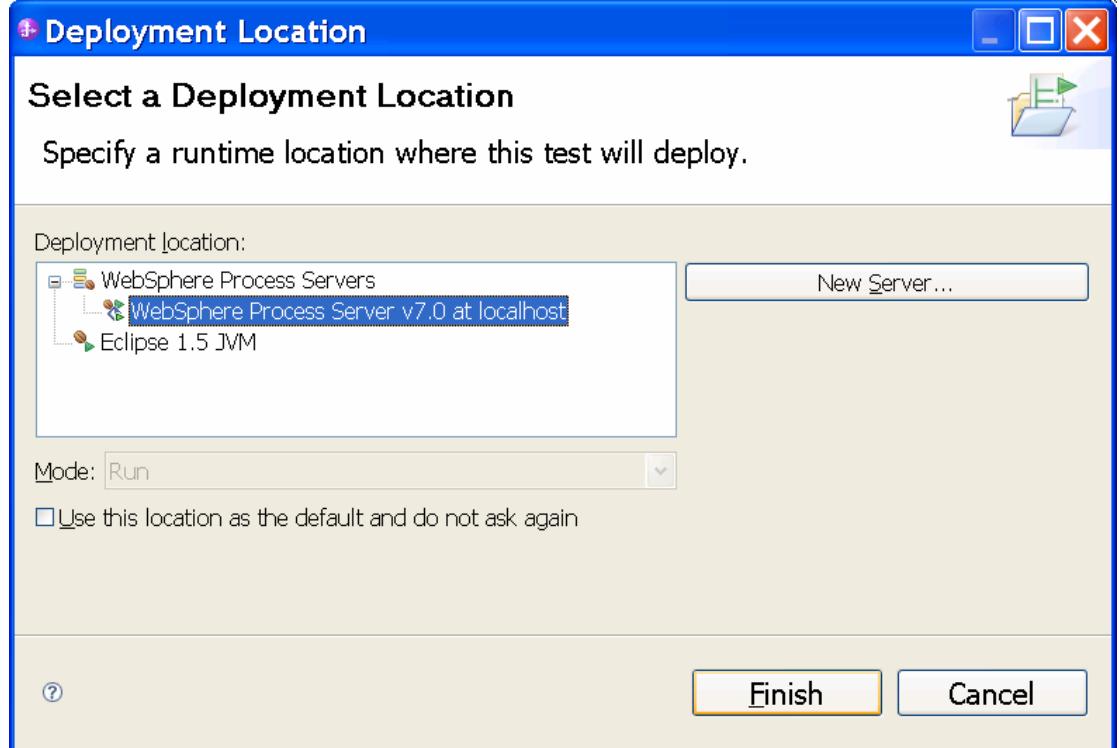
Initial request parameters:

Value editor XML editor

	Name	Type	Value
createDboCustor	DboCustomer		✓
verb	verb<string>	✓	Create
DboCustomer	DboCustomer		✓
pkey	string	✓	100
fname	string	✓	testFrame
lname	string	✓	testLname
ccode	string	✓	testCcode
addressobj	DboAddress[]	...	
address	DboAddress	✓	
addr	string	✓	100
custid	string	✓	100
city	string	✓	Beijing
zipcc	string	✓	100000

Type: <http://www.w3.org/2001/XMLSchema#string>

7. To execute the service click .
8. In the Deployment Location window, select the server, and click **Finish**.



The result of the test execution will be displayed once completed.

Clear the sample content

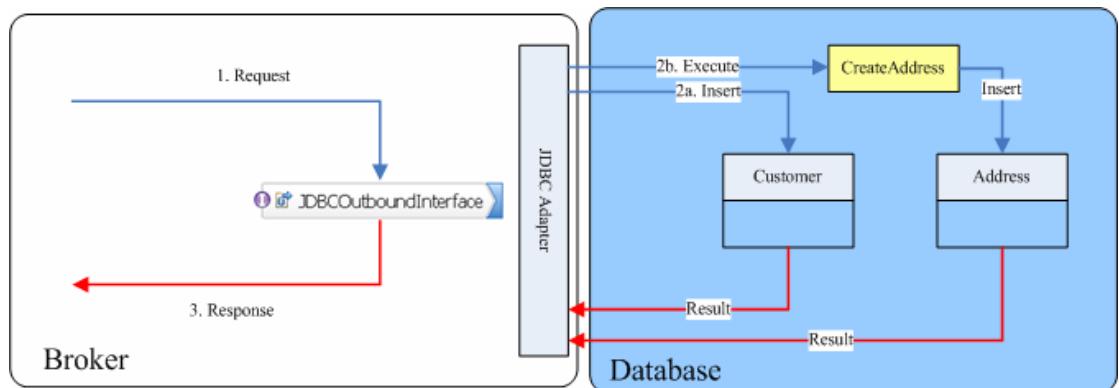
After a record created with the WebSphere Integration Developer environment, it can be removed using a Delete operation.

Chapter 4. Tutorial 3: Creating and executing stored procedure business objects with complex data types (Oracle)

This tutorial demonstrates how WebSphere Adapter for JDBC 7.0.0.0 creates business object for stored procedure and executes the stored procedure using the Execute operation. This tutorial also demonstrates the support for Array and Struct data types.

About this task

In this scenario, an application SCA component raises an execute request to the JDBC Outbound Interface. Then JDBC adapter constructs the complex SQL types according to the input business object and generates execute SQL statement to call the corresponding stored procedure. The stored procedure executes its internal business logic and generates output. Finally, the JDBC adapter generates a response according to the execution status and output of the stored procedure. The following figure represents this scenario:



Prepare to run through the tutorial

Extract the sample files

Replicas of the artifacts that you create when using the external service wizard are provided as sample files for your reference. Use these files to verify if the files you create using the external service wizard are correct.

Download the sample zip file and extract it into a directory of your choice (you may want to create a new directory).

Configuration prerequisites

Before configuring the adapter, you must complete the following tasks:

- Create types, tables and stored procedure
- Create an authentication alias
- Create a data source

Create types, tables and stored procedure

You must create the following tables, objects and stored procedures in the Oracle database before starting the scenario.

a. Script for creating the reference type

```
CREATE OR REPLACE TYPE ARRAYTYPE AS
  VARRAY(10) OF          VARCHAR2(50);
  /
```

This script creates a reference type of Array that holds up to 10 records of type VARCHAR2. It is used as an input type in our stored procedure.

```
CREATE OR REPLACE TYPE STRUCTTYPE AS OBJECT (
  EMPID VARCHAR2(10),
  NAME VARCHAR2(20),
  TITLE VARCHAR2(10)
);
/
```

This script creates a reference type of Struct that has three columns. It is used as an output type in our stored procedure.

Note: To create reference types, enter a forward slash (/) at the end and then press the Return key.

b. Script for creating tables

Create two tables that will be used in the stored procedure.

```
CREATE TABLE TABLE_ARRAY (
    ID VARCHAR2(10),
    INFO ARRAYTYPE );
```

```
CREATE TABLE TABLE_STRUCT (
    ID VARCHAR2(10) ,
    INFO STRUCTTYPE ) ;
```

Insert a record into TABLE_STRUCT by executing the following SQL statement:

```
INSERT INTO TABLE_STRUCT VALUES ('100',
STRUCTTYPE('10', 'xyz', 'SE'));
```

c. Script for creating the stored procedure

The stored procedure takes an Array type as an input parameter and returns a Struct type as an output parameter.

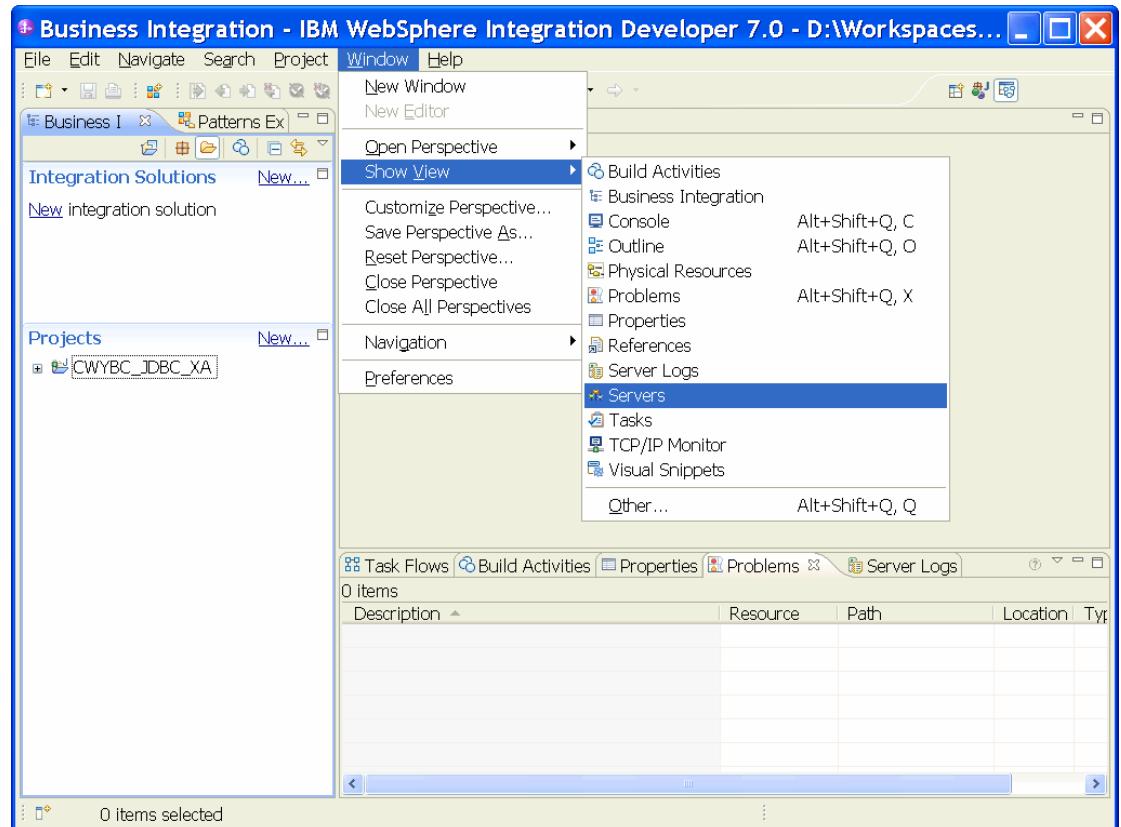
```
CREATE OR REPLACE PROCEDURE SAMPLE_ARRAY_STRUCT (
pkey IN VARCHAR, arr IN ARRAYTYPE, strt OUT STRUCTTYPE
)
IS BEGIN
INSERT INTO TABLE_ARRAY VALUES (pkey, arr);
SELECT INFO INTO strt FROM TABLE_STRUCT WHERE ID =
pkey;
END SAMPLE_ARRAY_STRUCT;
/
```

Create an authentication alias

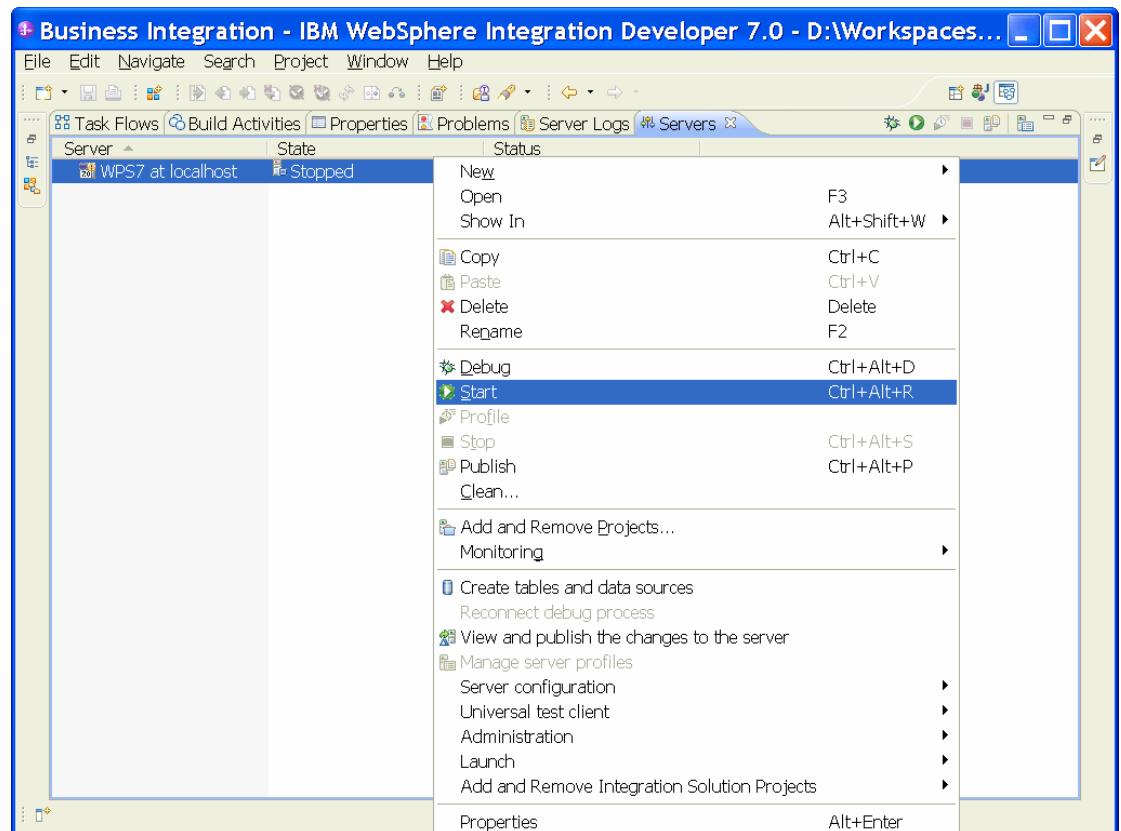
The authentication alias needs to be set because the data source created in the next section uses the username and password set in the authentication alias to connect to the database.

Follow these steps to set the authentication alias in the WebSphere Process Server administrative console.

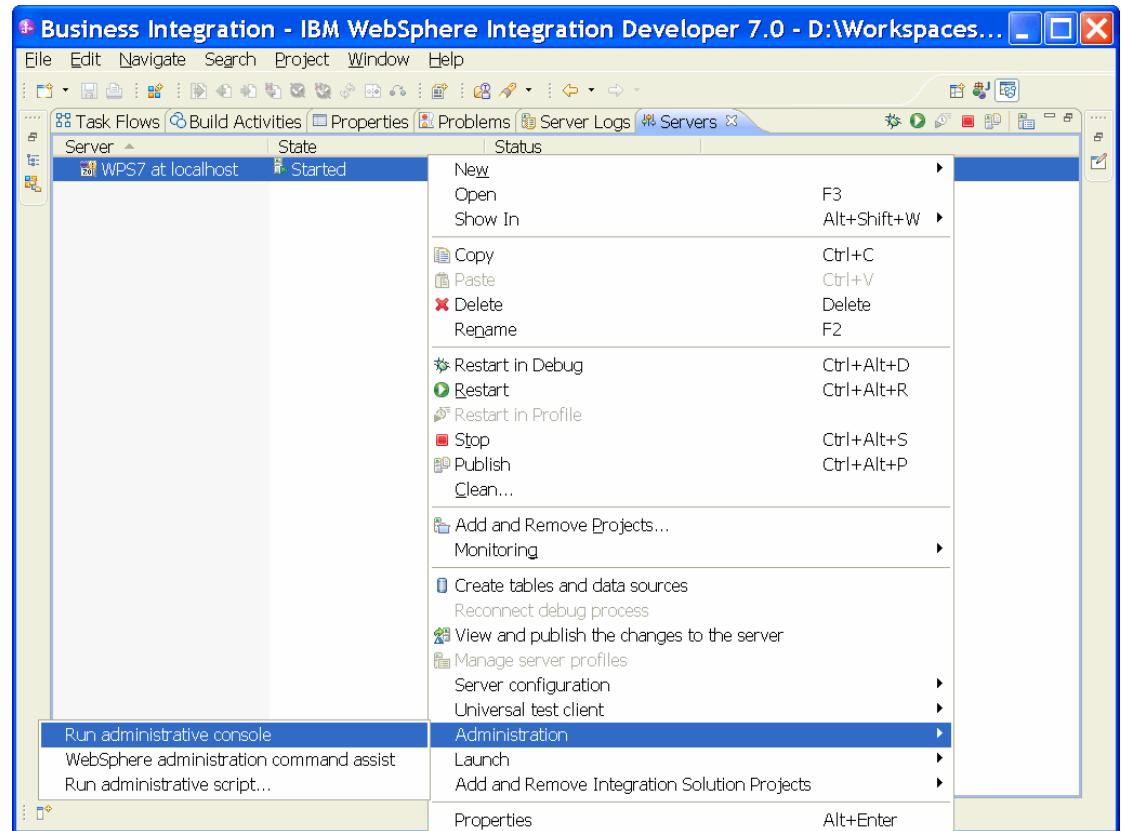
1. In WebSphere Integration Developer, switch to the **Servers** view by selecting **Windows > Show View > Servers**.



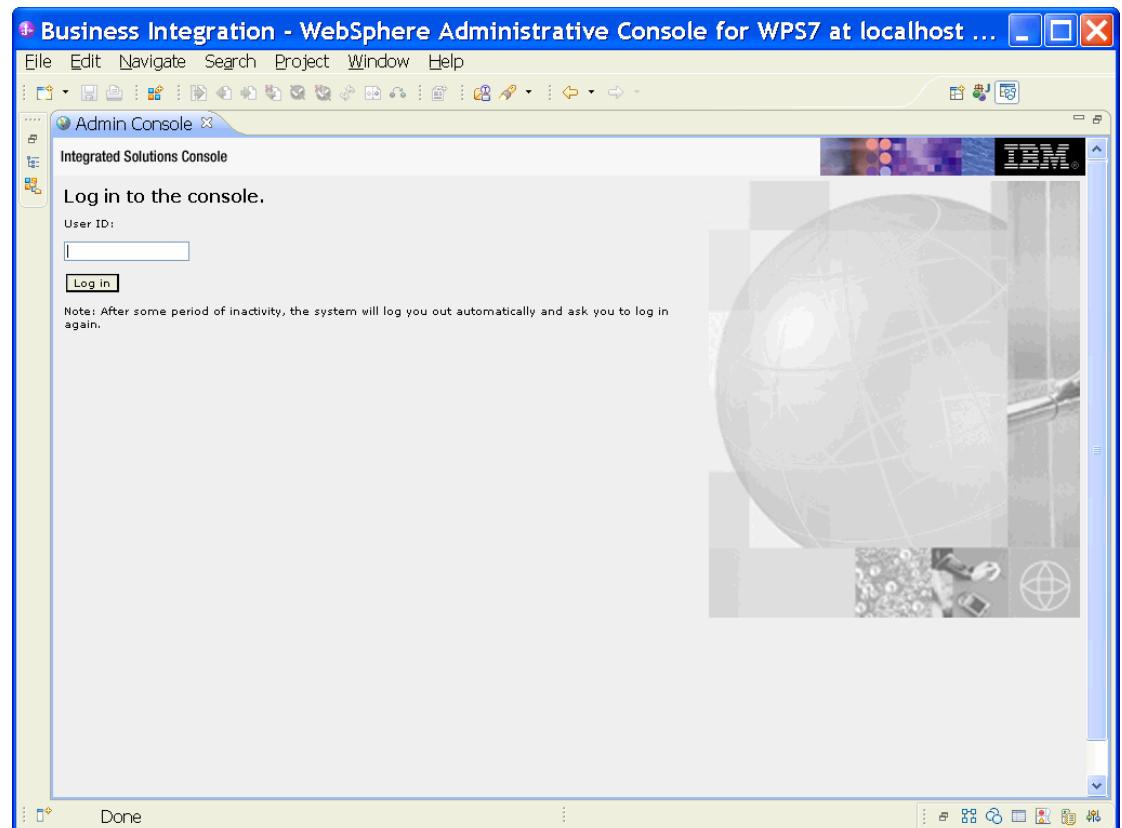
2. In the **Servers** view, right-click the server that you want to start and select **Start**.



3. After the server is started, right-click the server, and select **Administration > Run administrative console**.



4. Log on to the administrative console.



WebSphere software

5. Click **Security → Global security.**



6. On the right, click **J2C Authentication Data** under **Java Authentication and Authorization Service.**

Global security

Use this panel to configure administration and the default application security policy. This security configuration applies to functions and is used as a default security policy for user applications. Security domains can be defined to override and control applications.

[Security Configuration Wizard](#)[Security Configuration Report](#)**Administrative security**

- Enable administrative security
 - [Administrative user roles](#)
 - [Administrative group roles](#)
 - [Administrative authentication](#)

Application security

- Enable application security

Java 2 security

- Use Java 2 security to restrict application access to local resources
 - Warn if applications are granted custom permissions
 - Restrict access to resource authentication data

User account repository

Current realm definition

Federated repositories

Available realm definitions

Federated repositories

[Configure...](#)[Set as current](#)**Authentication**

Authentication mechanisms and expiration

[LTPA](#)[Kerberos and LTPA](#)[Kerberos configuration](#)[SWAM \(deprecated\): No authentication](#)[Authentication cache settings](#) Web and SIP security RMI/IOP security Java Authentication and Authorization[Application logins](#)[System logins](#)[J2C authentication data](#)

Specifies a list of user identities and passwords for Java(TM) 2 connector security to use.

 [Security domains](#) [External authorization providers](#) [Custom properties](#)

A list of existing aliases is displayed.

WebSphere software

Global security > JAAS - J2C authentication data

Specifies a list of user identities and passwords for Java(TM) 2 connector security to use.

Prefix new alias names with the node name of the cell (for compatibility with earlier releases)

Apply

Preferences

<input type="button" value="New"/> <input type="button" value="Delete"/>			
Select	Alias	User ID	Description
You can administer the following resources:			
<input type="checkbox"/>	BSpace JDBC Alias	wbiuser	Business Space Authorization Alias
<input type="checkbox"/>	CommonEventInfrastructureJMSAuthAlias	wbiuser	Authentication alias for the Common Event Infrastructure JMS Topics and Queues
<input type="checkbox"/>	SCA Auth Alias	wbiuser	This is the alias used by SCA to login to a secured SIBus
<input type="checkbox"/>	localhostNode01Cell/nlNode01/server1/EventAuthDataAliasDerby		Derby authentication alias for the Event Server
Total 4			

- Click **New** to create a new authentication entry. Type the alias name, and username and password to connect to the database. Click **OK**.

Cell=localhostNode01Cell, Profile=AppSrv01

Global security

Global security > JAAS - J2C authentication data > New

Specifies a list of user identities and passwords for Java(TM) 2 connector security to use.

General Properties

* Alias

* User ID

* Password

Description

Buttons

- Click **Save** to save the changes.

Cell=localhostNode01Cell, Profile=AppSrv01

Global security > JAAS - J2C authentication data

Specifies a list of user identities and passwords for Java(TM) 2 connector security to use.

Prefix new alias names with the node name of the cell (for compatibility with earlier releases)

Apply

You have created an authentication alias that will be used to configure the data source.

Select	Alias	User ID	Description
<input type="checkbox"/>	BSpace JDBC Alias	wbiuser	Business Space Authorization Alias
<input type="checkbox"/>	CommonEventInfrastructureJMSAuthAlias	wbiuser	Authentication alias for the Common Event Infrastructure JMS Topics and Queues
<input type="checkbox"/>	SCA Auth Alias	wbiuser	This is the alias used by SCA to login to a secured SIBus
<input type="checkbox"/>	localhostNode01Cell/nlNode01/server1/EventAuthDataAliasDerby		Derby authentication alias for the Event Server
<input type="checkbox"/>	nlNode01/Alias	luweiqin	

Total 5

Create the data source

Create a data source in WebSphere Process Server, which the adapter will use to connect to the database. This data source will be used later when generating the artifacts for the module.

Note: This tutorial will use Oracle as the database and the Oracle thin driver, ojdbc6.jar.

WebSphere software

Here are the steps to create the data source in the WebSphere Process Server administrative console.

1. In the administrative console, select **Environment → WebSphere Variables**.



2. On the right, click **ORACLE_JDBC_DRIVER_PATH** and specify the path of the ojdbc6.jar file in the **Value** field. Click **OK**.

WebSphere software

Cell=localhostNode01Cell, Profile=AppSrv01

WebSphere Variables

[WebSphere Variables](#) > ORACLE_JDBC_DRIVER_PATH

Use this page to define substitution variables. Variables specify a level of indirection for some system-defined values, such as file system root directories. Variables have a scope level, which is either server, node, cluster, or cell. Values at one scope level can differ from values at other levels. When a variable has conflicting scope values, the more granular scope value overrides values at greater scope levels. Therefore, server variables override node variables, which override cluster variables, which override cell variables.

Configuration

General Properties

* Name: ORACLE_JDBC_DRIVER_PATH

Value: D:\Lib

Description: The directory that contains the Oracle thin or oci8 JDBC Driver.

Buttons: Apply, OK, Reset, Cancel

3. Click **Save** to save the changes.

WebSphere Variables

Messages

⚠ Changes have been made to your local configuration. You can:
• [Save](#) directly to the master configuration.
• [Review](#) changes before saving or discarding.
⚠ The server may need to be restarted for these changes to take effect.

The variable has been added and appears in the list.

Preferences

New Delete

Select Name Value Scope

You can administer the following resources:

<input type="checkbox"/>	MQ_INSTALL_ROOT	`\${WAS_INSTALL_ROOT}/lib/wmq	Node=n1Node01
<input type="checkbox"/>	ORACLE_JDBC_DRIVER_PATH	D:\Lib	Node=n1Node01
<input type="checkbox"/>	OS400_NATIVE_JDBC40_DRIVER_PATH		Node=n1Node01
<input type="checkbox"/>	OS400_NATIVE_JDBC_DRIVER_PATH		Node=n1Node01
<input type="checkbox"/>	OS400_TOOLBOX_JDBC_DRIVER_PATH		Node=n1Node01
<input type="checkbox"/>	ORACLE_JDBC_DRIVER_PATH		Node=n1Node01

WebSphere software

4. Select **Resources** → **JDBC** -> **JDBC Providers**.



5. Click **New** in the JDBC providers window.

JDBC providers

JDBC providers

Use this page to edit properties of a JDBC provider. The JDBC provider object encapsulates the specific JDBC driver implementation class for access to the specific vendor database of your environment. Learn more about this task in a [guided activity](#). A guided activity provides a list of task steps and more general information about the topic.

Scope: Cell=**localhostNode01Cell**, Node=**n1Node01**

Scope specifies the level at which the resource definition is visible. For detailed information on what scope is and how it works, [see the scope settings help](#).

Node=n1Node01

Preferences

Select	Name	Scope	Description
None			
Total 0			

6. In the Create new JDBC provider page, select an Oracle database with a connection pool data source for the Oracle JDBC driver. Click **Next**.

Create a new JDBC Provider

Create a new JDBC Provider

→ Step 1: Create new JDBC provider

Step 2: Enter database class path information

Step 3: Summary

Create new JDBC provider

Set the basic configuration values of a JDBC provider, which encapsulates the specific vendor JDBC driver implementation classes that are required to access the database. The wizard fills in the name and the description fields, but you can type different values.

Scope

cells:localhostNode01Cell:nodes:n1Node01

* Database type

Oracle

* Provider type

Oracle JDBC Driver

* Implementation type

Connection pool data source

* Name

Oracle JDBC Driver

Description

Oracle JDBC Driver

Next → Cancel

7. In the Enter database classpath information page, enter the following value in the **Class path** field:
\$(ORACLE_JDBC_DRIVER_PATH)/ojdbc6.jar, where
\$(ORACLE_JDBC_DRIVER_PATH) is library path for the run time.
8. Click **Next**.

WebSphere software

Cell=localhostNode01Cell, Profile=AppSrv01

Create a new JDBC Provider

Step 1: Create new JDBC provider
→ Step 2: Enter database class path information
Step 3: Summary

Enter database class path information

Set the environment variables that represent the JDBC driver class files, which WebSphere(R) Application Server uses to define your JDBC provider. This wizard page displays the file names; you supply only the directory locations of the files. Use complete directory paths when you type the JDBC driver file locations. For example: C:\SQLLIB\java on Windows(R) or /home/db2inst1/sqllib/java on Linux(TM).

If a value is specified for you, you may click Next to accept the value.

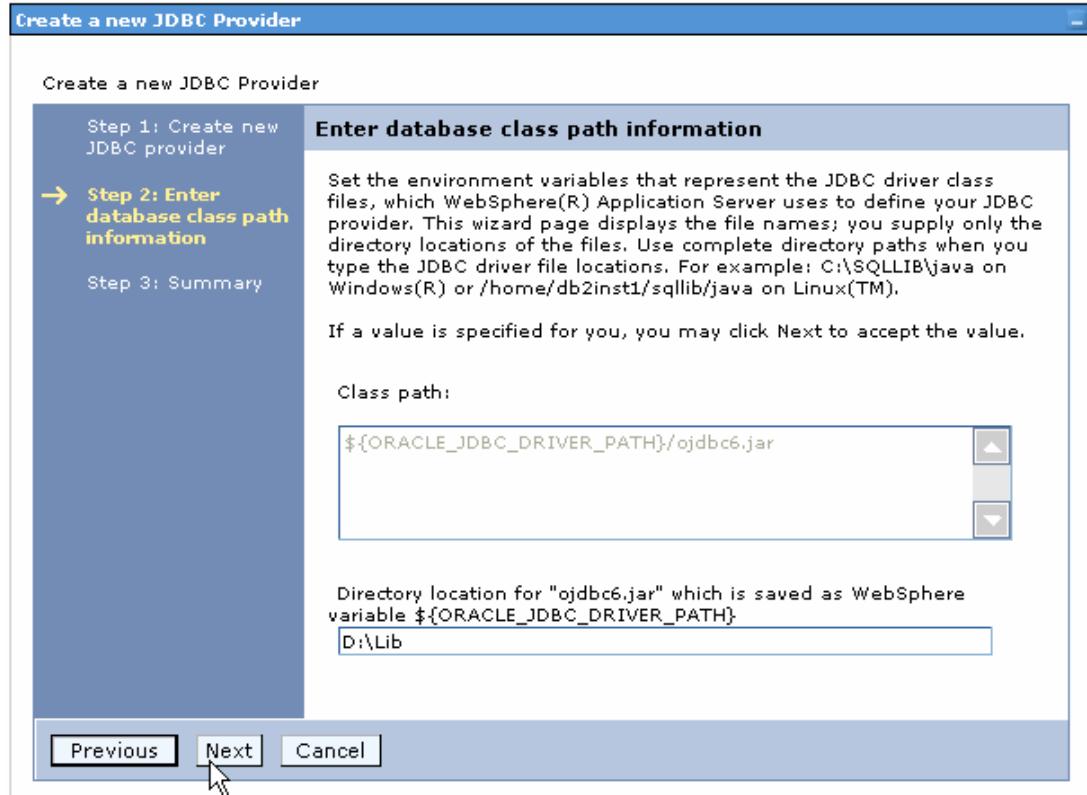
Class path:

`${ORACLE_JDBC_DRIVER_PATH}/ojdbc6.jar`

Directory location for "ojdbc6.jar" which is saved as WebSphere variable `${ORACLE_JDBC_DRIVER_PATH}`

D:\Lib

Previous **Next** Cancel



9. Click **Finish**.

Cell=localhostNode01Cell, Profile=AppSrv01

[Close page](#)

Create a new JDBC Provider

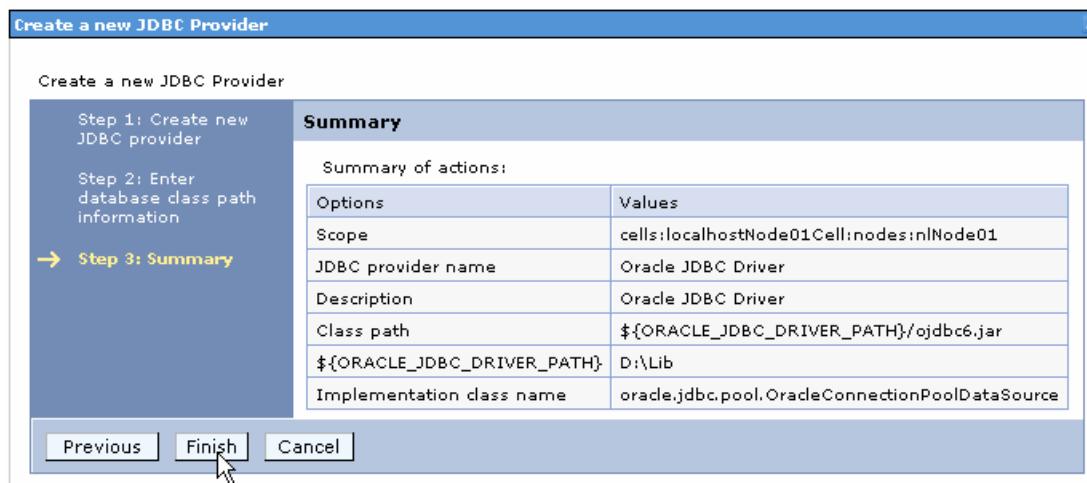
Step 1: Create new JDBC provider
Step 2: Enter database class path information
→ Step 3: Summary

Summary

Summary of actions:

Options	Values
Scope	cells:localhostNode01Cell:nodes:n1Node01
JDBC provider name	Oracle JDBC Driver
Description	Oracle JDBC Driver
Class path	<code> \${ORACLE_JDBC_DRIVER_PATH}/ojdbc6.jar</code>
<code> \${ORACLE_JDBC_DRIVER_PATH}</code>	D:\Lib
Implementation class name	oracle.jdbc.pool.OracleConnectionPoolDataSource

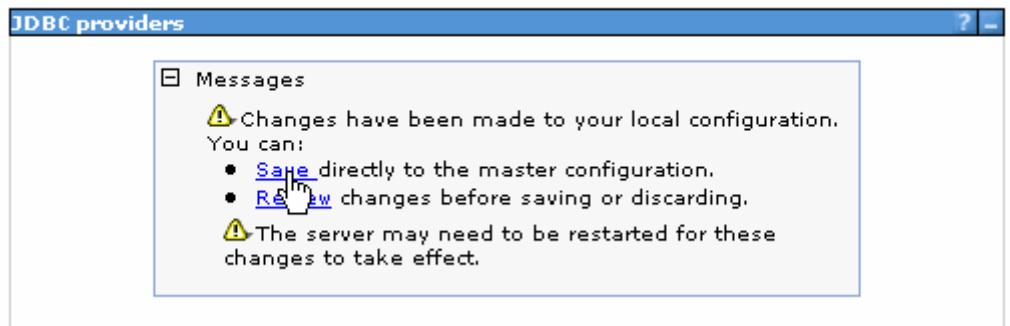
Previous **Finish** Cancel



10. Click **Save** to save the changes.

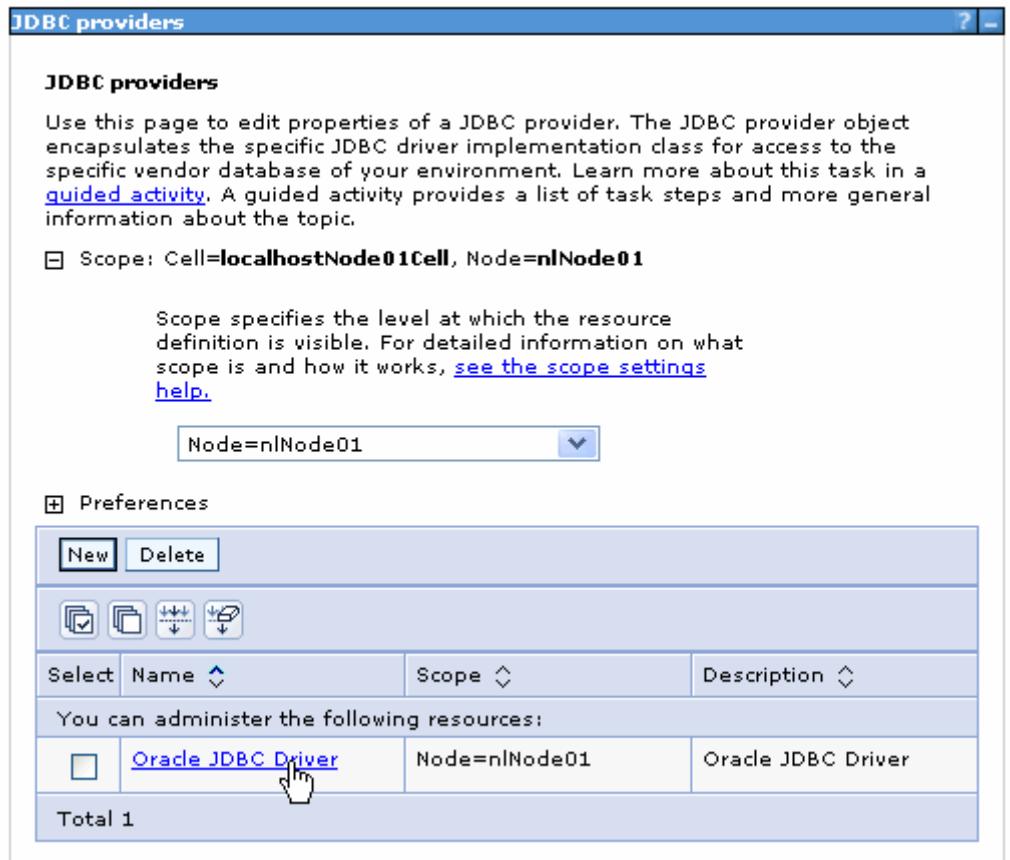
WebSphere software

Cell=localhostNode01Cell, Profile=AppSrv01



The JDBC provider is added and appears in the list.

Cell=localhostNode01Cell, Profile=AppSrv01



11. Click the Oracle JDBC provider you just created. Under **Additional Properties**, click **Data sources**. Click **New**.

WebSphere software

Cell=localhostNode01Cell, Profile=AppSrv01

The screenshot shows the 'JDBC providers' page under 'Oracle JDBC Driver > Data sources'. A tooltip for 'New' is visible over the 'New' button. The table lists one data source named 'None'.

Select	Name	JNDI name	Scope	Provider	Description	Category
	None					
Total 0						

12. Type any value in the **JNDI name** field, and select the authentication alias. Click **Next**.

Cell=localhostNode01Cell, Profile=AppSrv01

The screenshot shows the 'Create a data source' wizard at Step 1: Enter basic data source information. The 'Scope' field is set to 'cells:localhostNode01Cell:nodes:n1Node01'. The 'JDBC provider name' is 'Oracle JDBC Driver'. The 'Data source name' is 'Oracle JDBC Driver DataSource' and the 'JNDI name' is 'OracleDS'. The 'Next' button is highlighted with a cursor.

13. Provide the appropriate URL value and select a data store helper class name from the **Data store helper class name** list as shown in the following figure. Click **Next**.

WebSphere software

Cell=localhostNode01Cell, Profile=AppSrv01

Create a data source

Step 1: Enter basic data source information

→ Step 2: Enter database specific properties for the data source

Step 3: Setup security aliases

Step 4: Summary

Enter database specific properties for the data source

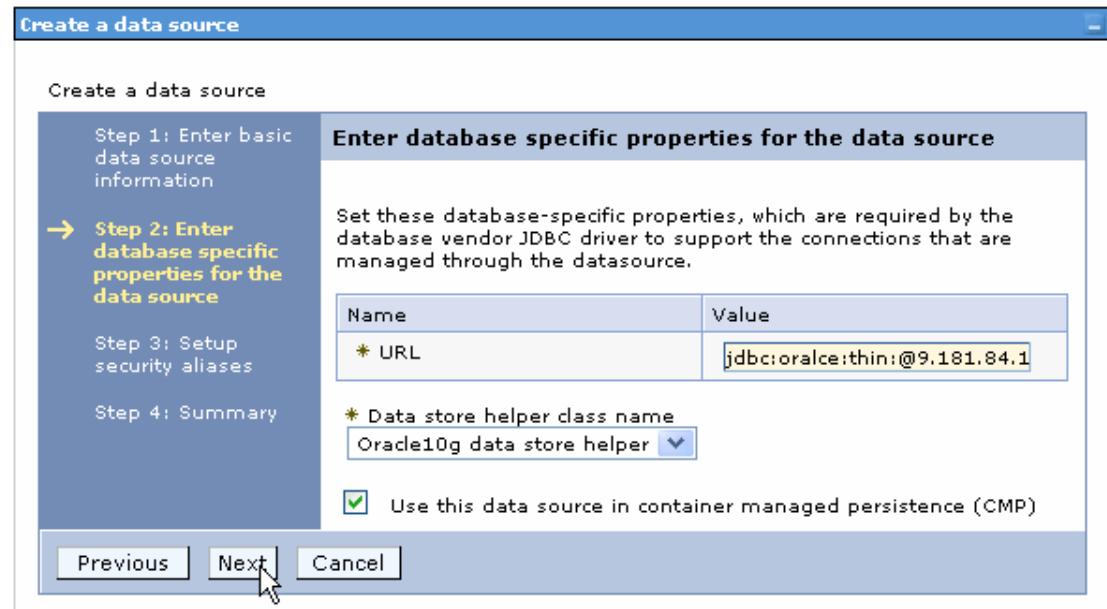
Set these database-specific properties, which are required by the database vendor JDBC driver to support the connections that are managed through the datasource.

Name	Value
* URL	jdbc:oracle:thin:@9.181.84.1

* Data store helper class name
Oracle10g data store helper

Use this data source in container managed persistence (CMP)

Previous Next Cancel



14. Select the authentication alias you just created from the **Component-managed authentication alias** list. Click **Next**.

Cell=localhostNode01Cell, Profile=AppSrv01 Close

Create a data source

Step 1: Enter basic data source information

Step 2: Enter database specific properties for the data source

→ Step 3: Setup security aliases

Step 4: Summary

Setup security aliases

Select the authentication values for this resource.

Component-managed authentication alias
nlNode01/Alias_Oracle

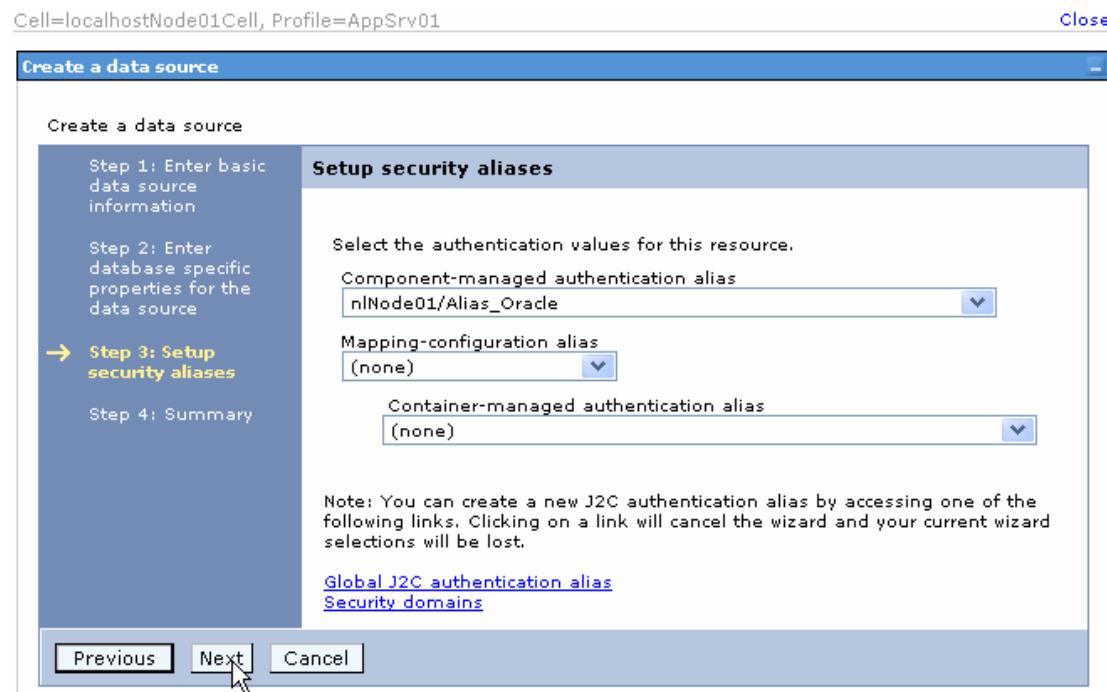
Mapping-configuration alias
(none)

Container-managed authentication alias
(none)

Note: You can create a new J2C authentication alias by accessing one of the following links. Clicking on a link will cancel the wizard and your current wizard selections will be lost.

[Global J2C authentication alias](#)
[Security domains](#)

Previous Next Cancel



The Summary of the values entered for the data source will be shown.

15. Click **Finish**.

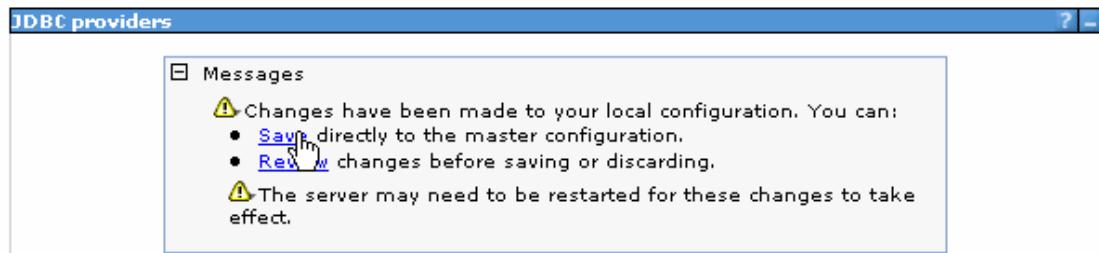
WebSphere software

Create a data source

Step 1: Enter basic data source information Step 2: Enter database specific properties for the data source Step 3: Setup security aliases → Step 4: Summary	Summary Summary of actions: <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">Options</th> <th style="text-align: left;">Values</th> </tr> </thead> <tbody> <tr> <td>Scope</td> <td>cells:localhostNode01Cell:nodes:n1Node01</td> </tr> <tr> <td>Data source name</td> <td>Oracle JDBC Driver DataSource</td> </tr> <tr> <td>JNDI name</td> <td>OracleDS</td> </tr> <tr> <td>Select an existing JDBC provider</td> <td>Oracle JDBC Driver</td> </tr> <tr> <td>Implementation class name</td> <td>oracle.jdbc.pool.OracleConnectionPoolDataSource</td> </tr> <tr> <td>URL</td> <td>jdbc:oracle:thin:@9.181.84.136:1521:ord</td> </tr> <tr> <td>Data store helper class name</td> <td>com.ibm.websphere.radapter.Oracle10gDataStoreHelper</td> </tr> <tr> <td>Use this data source in container managed persistence (CMP)</td> <td>true</td> </tr> <tr> <td>Component-managed authentication alias</td> <td>n1Node01/Alias_Oracle</td> </tr> <tr> <td>Mapping-configuration alias</td> <td>(none)</td> </tr> <tr> <td>Container-managed authentication alias</td> <td>(none)</td> </tr> </tbody> </table>	Options	Values	Scope	cells:localhostNode01Cell:nodes:n1Node01	Data source name	Oracle JDBC Driver DataSource	JNDI name	OracleDS	Select an existing JDBC provider	Oracle JDBC Driver	Implementation class name	oracle.jdbc.pool.OracleConnectionPoolDataSource	URL	jdbc:oracle:thin:@9.181.84.136:1521:ord	Data store helper class name	com.ibm.websphere.radapter.Oracle10gDataStoreHelper	Use this data source in container managed persistence (CMP)	true	Component-managed authentication alias	n1Node01/Alias_Oracle	Mapping-configuration alias	(none)	Container-managed authentication alias	(none)
Options	Values																								
Scope	cells:localhostNode01Cell:nodes:n1Node01																								
Data source name	Oracle JDBC Driver DataSource																								
JNDI name	OracleDS																								
Select an existing JDBC provider	Oracle JDBC Driver																								
Implementation class name	oracle.jdbc.pool.OracleConnectionPoolDataSource																								
URL	jdbc:oracle:thin:@9.181.84.136:1521:ord																								
Data store helper class name	com.ibm.websphere.radapter.Oracle10gDataStoreHelper																								
Use this data source in container managed persistence (CMP)	true																								
Component-managed authentication alias	n1Node01/Alias_Oracle																								
Mapping-configuration alias	(none)																								
Container-managed authentication alias	(none)																								

16. Click **Save** to save the changes.

Cell=localhostNode01Cell, Profile=AppSrv01



17. Select the check box corresponding to the data source you created in the previous step and click **Test connection**.

Cell=localhostNode01Cell, Profile=AppSrv01

JDBC providers

[JDBC providers](#) > [Oracle JDBC Driver](#) > Data sources

Use this page to edit the settings of a datasource that is associated with your selected JDBC provider. The datasource object supplies your application with connections for accessing the database. Learn more about this task in a [guided activity](#). A guided activity provides a list of task steps and more general information about the topic.

Preferences

New	Delete	Test connection	Manage state...
Select	Name ▾	JNDI name ▾	Scope ▾
Provider ▾	Description ▾	Category ▾	

You can administer the following resources:

<input checked="" type="checkbox"/> Oracle JDBC Driver DataSource	OracleDS	Node=n1Node01	Oracle JDBC Driver	New JDBC Datasource	
---	----------	---------------	--------------------	---------------------	--

Total 1

The connection should succeed as shown in the following figure. If you experience problems while testing the connection, refer to the "Troubleshooting" section.

Cell=localhostNode01Cell, Profile=AppSrv01

JDBC providers

[JDBC providers](#) > [Oracle JDBC Driver](#) > Data sources

Use this page to edit the settings of a datasource that is associated with your selected JDBC provider. The datasource object supplies your application with connections for accessing the database. Learn more about this task in a [guided activity](#). A guided activity provides a list of task steps and more general information about the topic.

Preferences

New	Delete	Test connection	Manage state...
Select	Name ▾	JNDI name ▾	Scope ▾
Provider ▾	Description ▾	Category ▾	

Messages

The test connection operation for data source Oracle JDBC Driver DataSource on server server1 at node n1Node01 was successful.

[Information](#)

You can administer the following resources:

<input type="checkbox"/> Oracle JDBC Driver DataSource	OracleDS	Node=n1Node01	Oracle JDBC Driver	New JDBC Datasource	
--	----------	---------------	--------------------	---------------------	--

Total 1

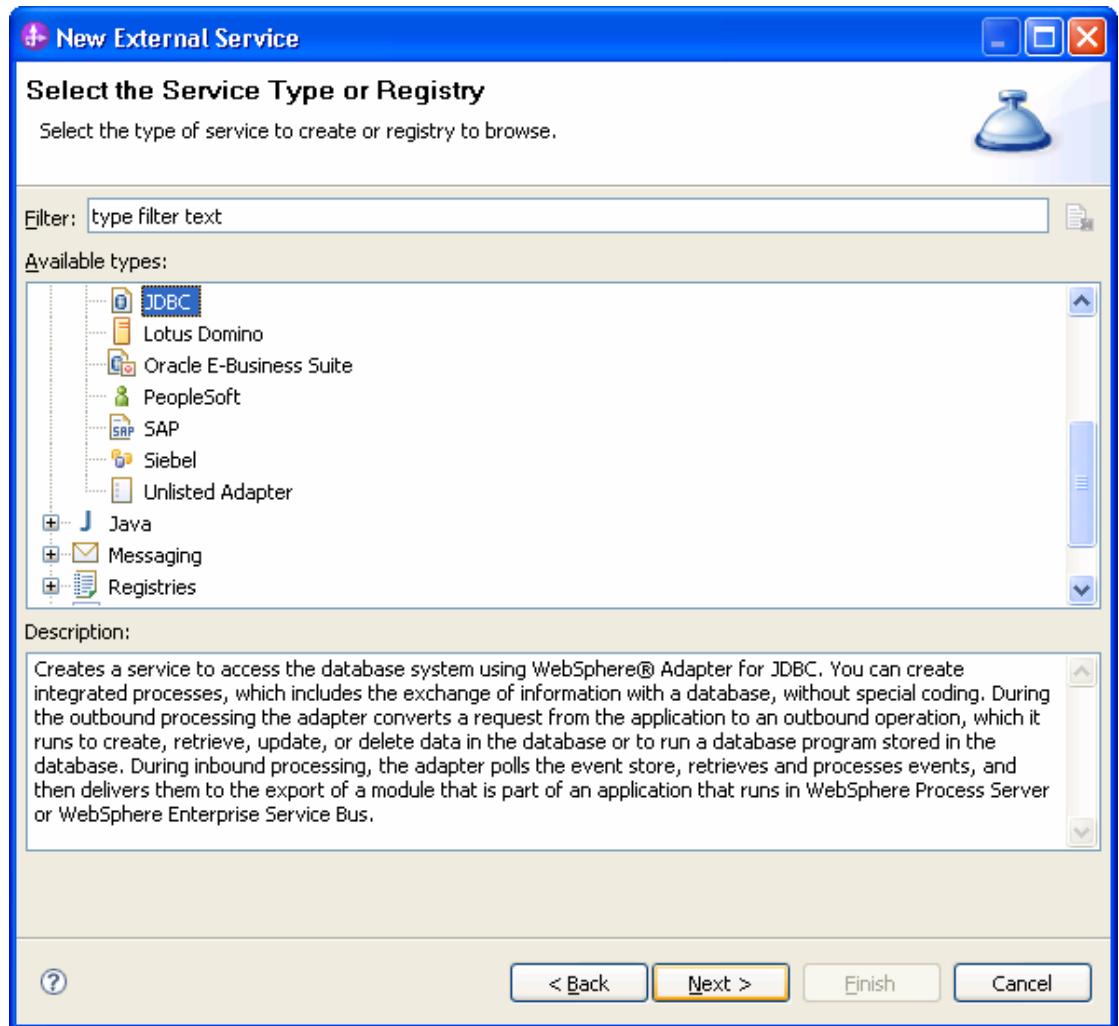
Note: The data source is created which will be used by the adapter to connect to the database.

Configure the adapter for outbound processing

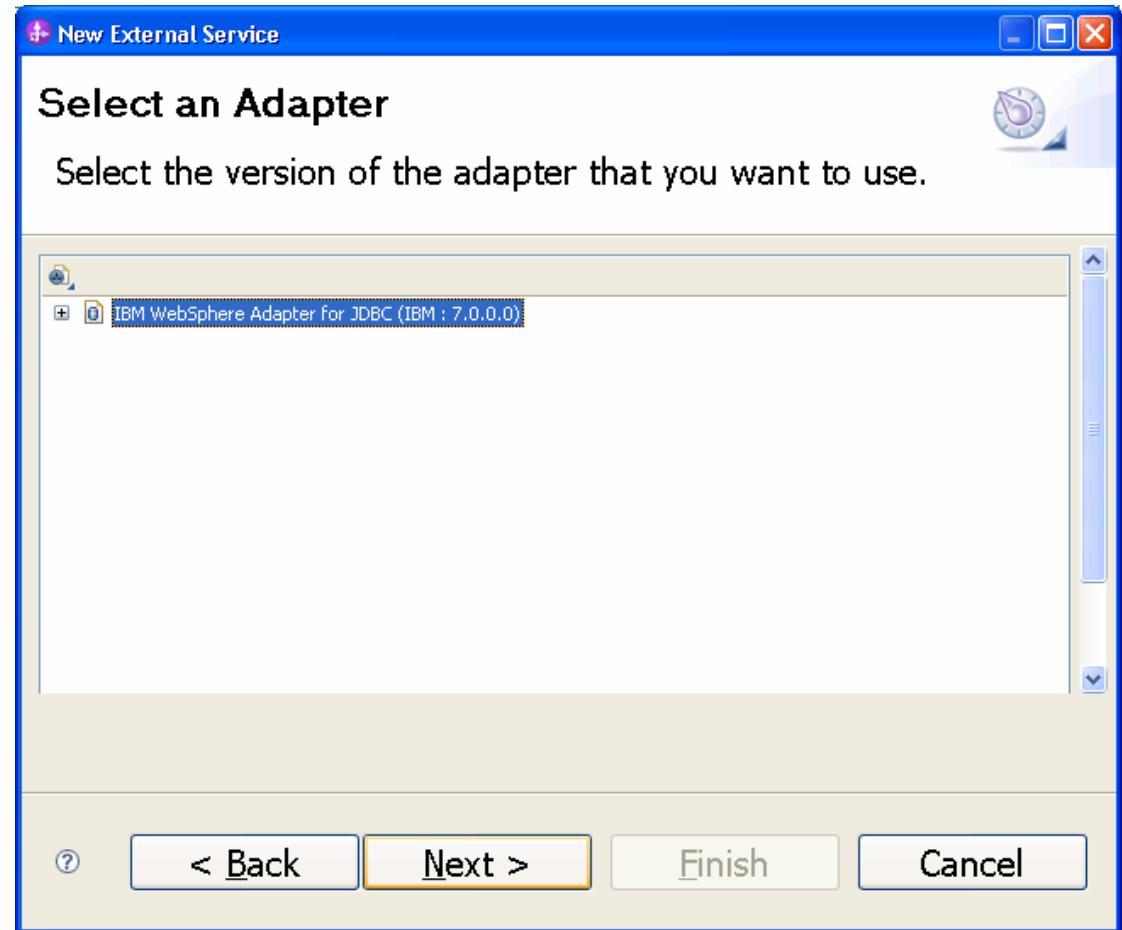
WebSphere software

Run the external service wizard to specify business objects, services, and configuration details.

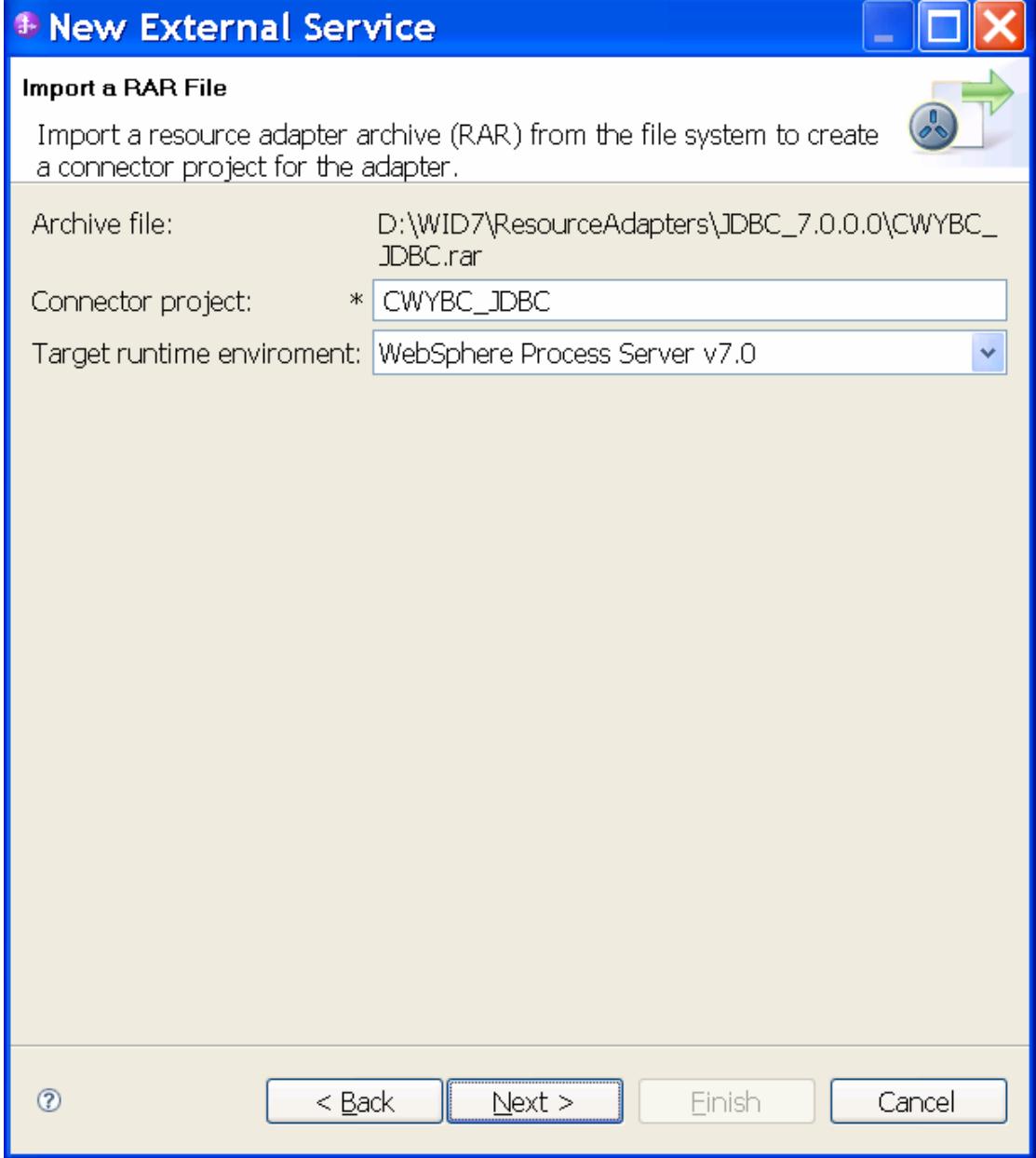
1. Switch to the Business Integration Perspective in WebSphere Integration Developer by selecting **Window -> Open Perspective Business Integration**.
2. Start the external service wizard by selecting **File-> New -> External Service**.
3. In the **Available Types** area, select **Adapters > JDBC** and then click **Next**.



4. Select the **IBM WebSphere Adapter for JDBC (IBM: 7.0.0.0)** and click **Next**.



5. In the **Connector project** field enter **CWYBC_JDBC**, and in the **Target runtime environment** field, select appropriate runtime. Click **Next**.



New External Service

Import a RAR File

Import a resource adapter archive (RAR) from the file system to create a connector project for the adapter.



Archive file: D:\WID7\ResourceAdapters\JDBC_7.0.0.0\CWYBC_JDBC.rar

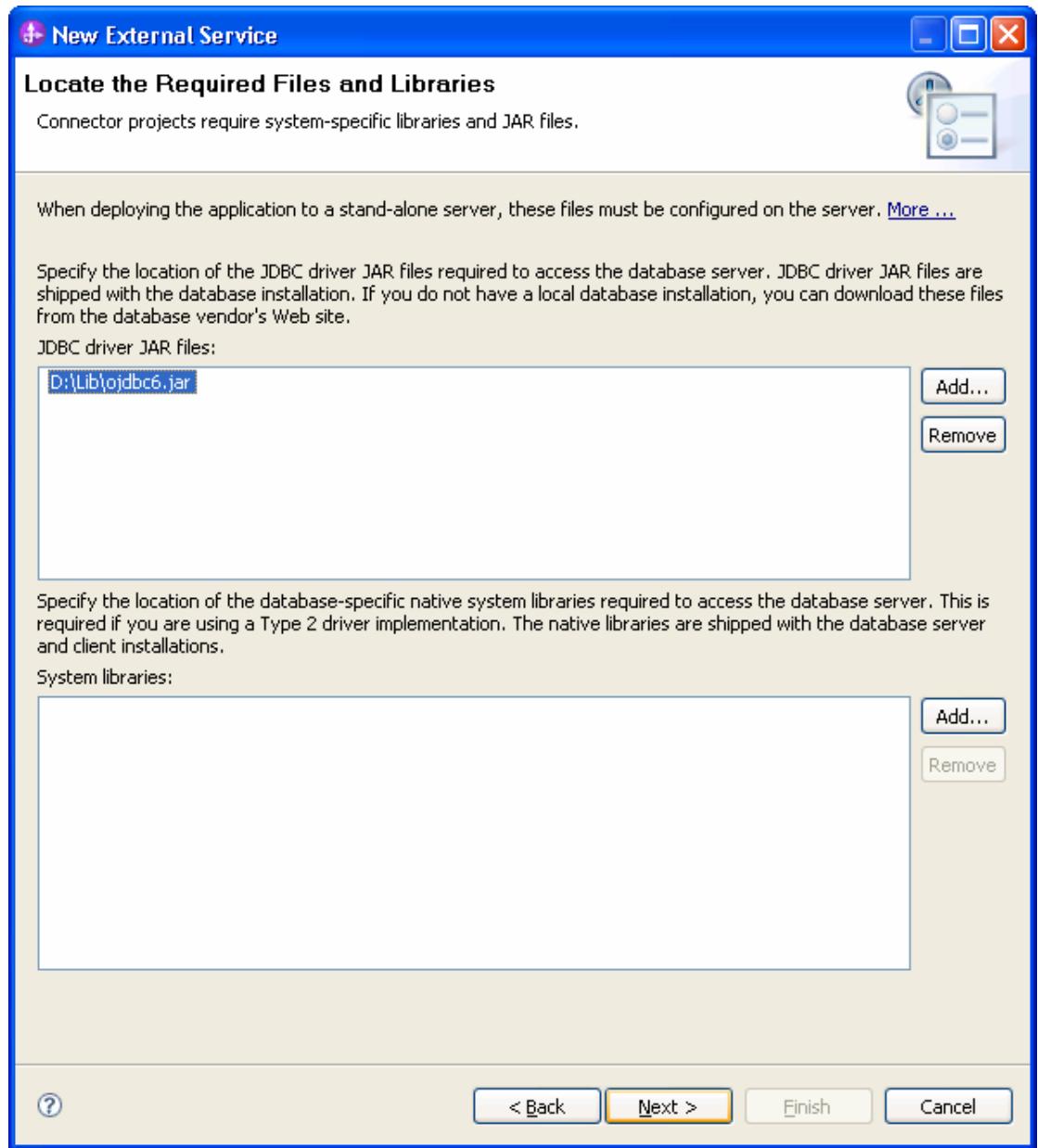
Connector project: * CWYBC_JDBC

Target runtime environment: WebSphere Process Server v7.0

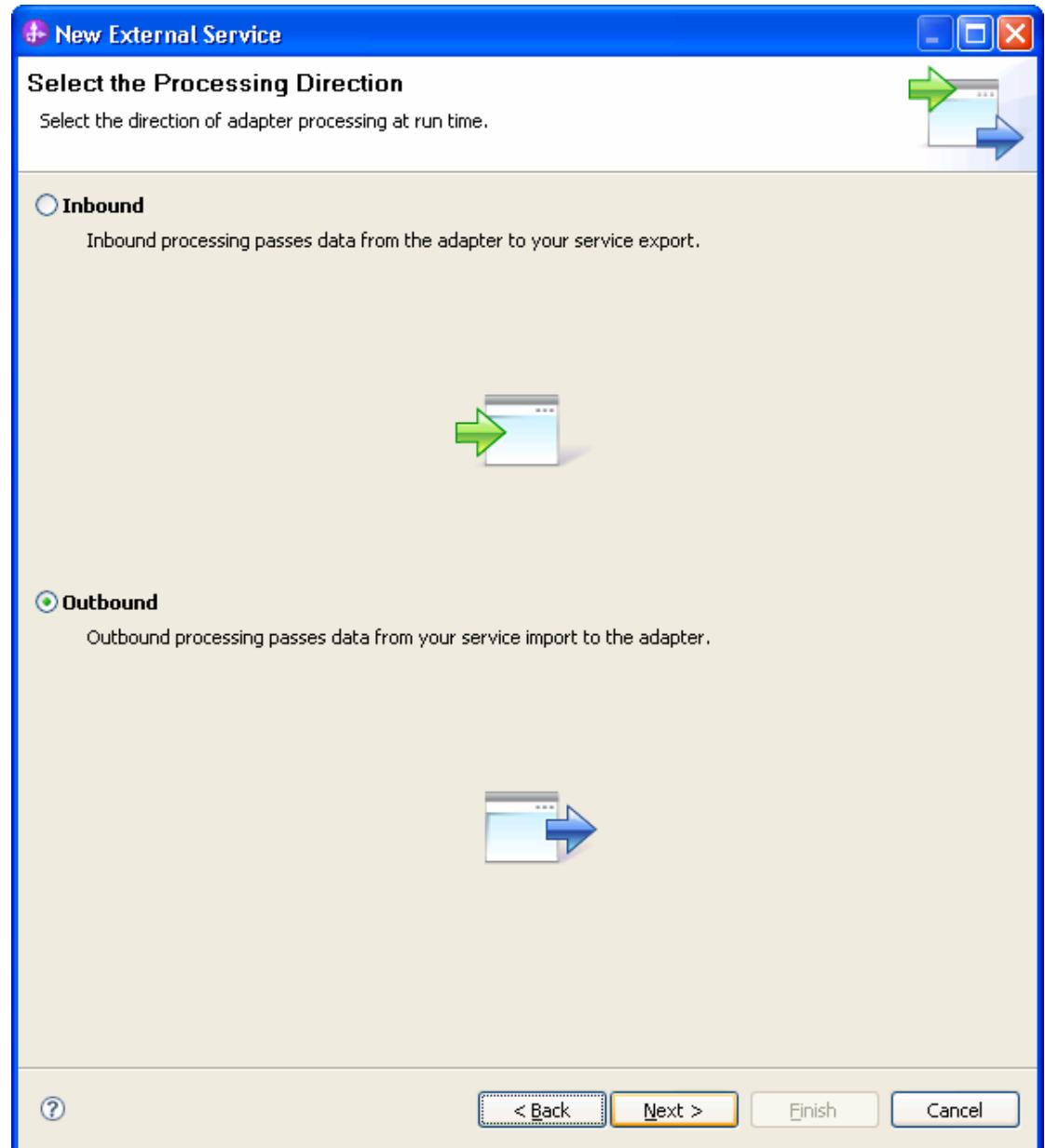
[?](#)[< Back](#)[Next >](#)[Finish](#)[Cancel](#)

WebSphere software

6. In the **JDBC driver JAR files** field, click **Add**, to add the JDBC driver class to connect to the database. Browse to select the driver JAR file and click **Next**.



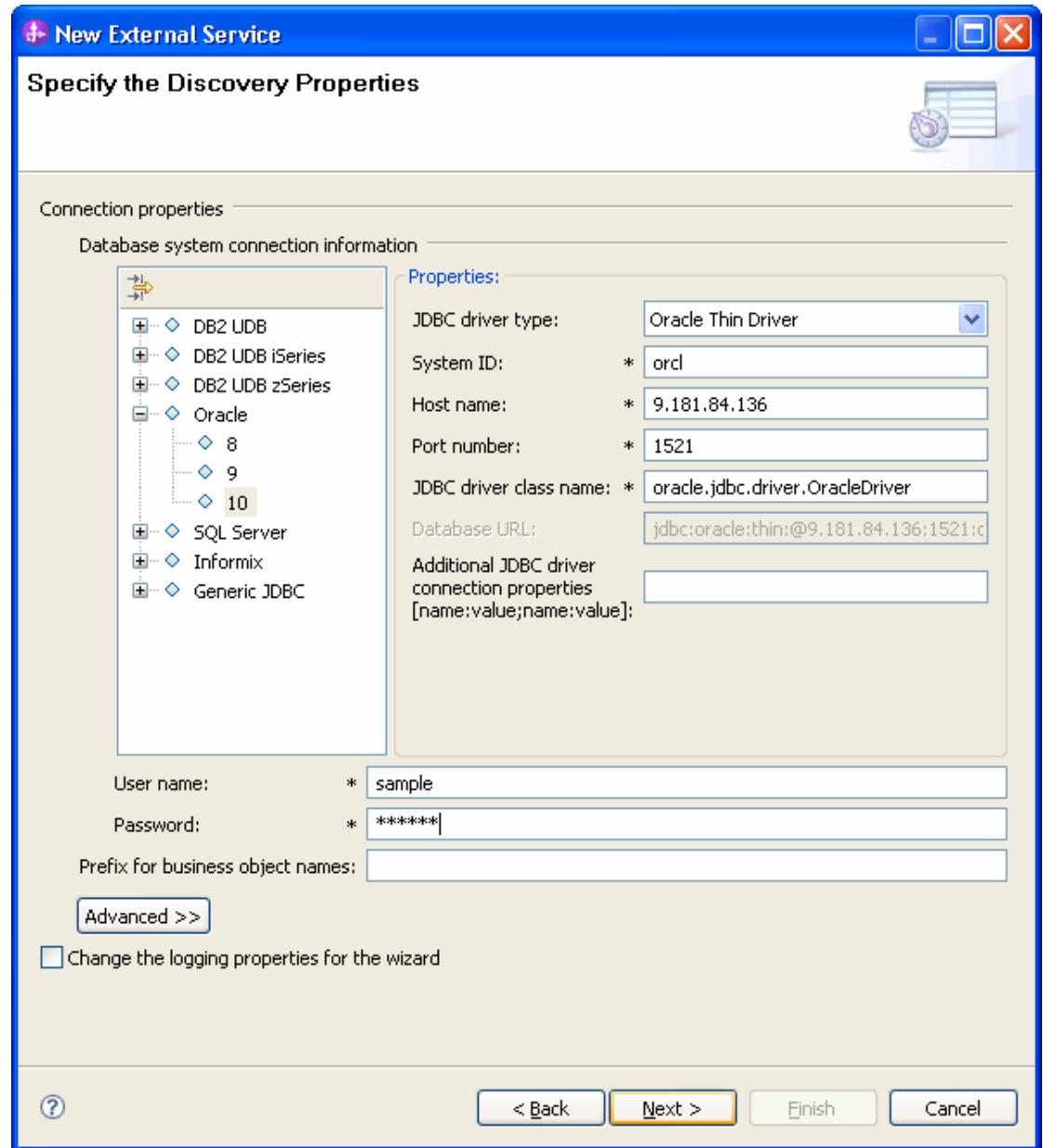
7. Select **Outbound** and click **Next**.



Set connection properties for the external service wizard

To connect to the Oracle database:

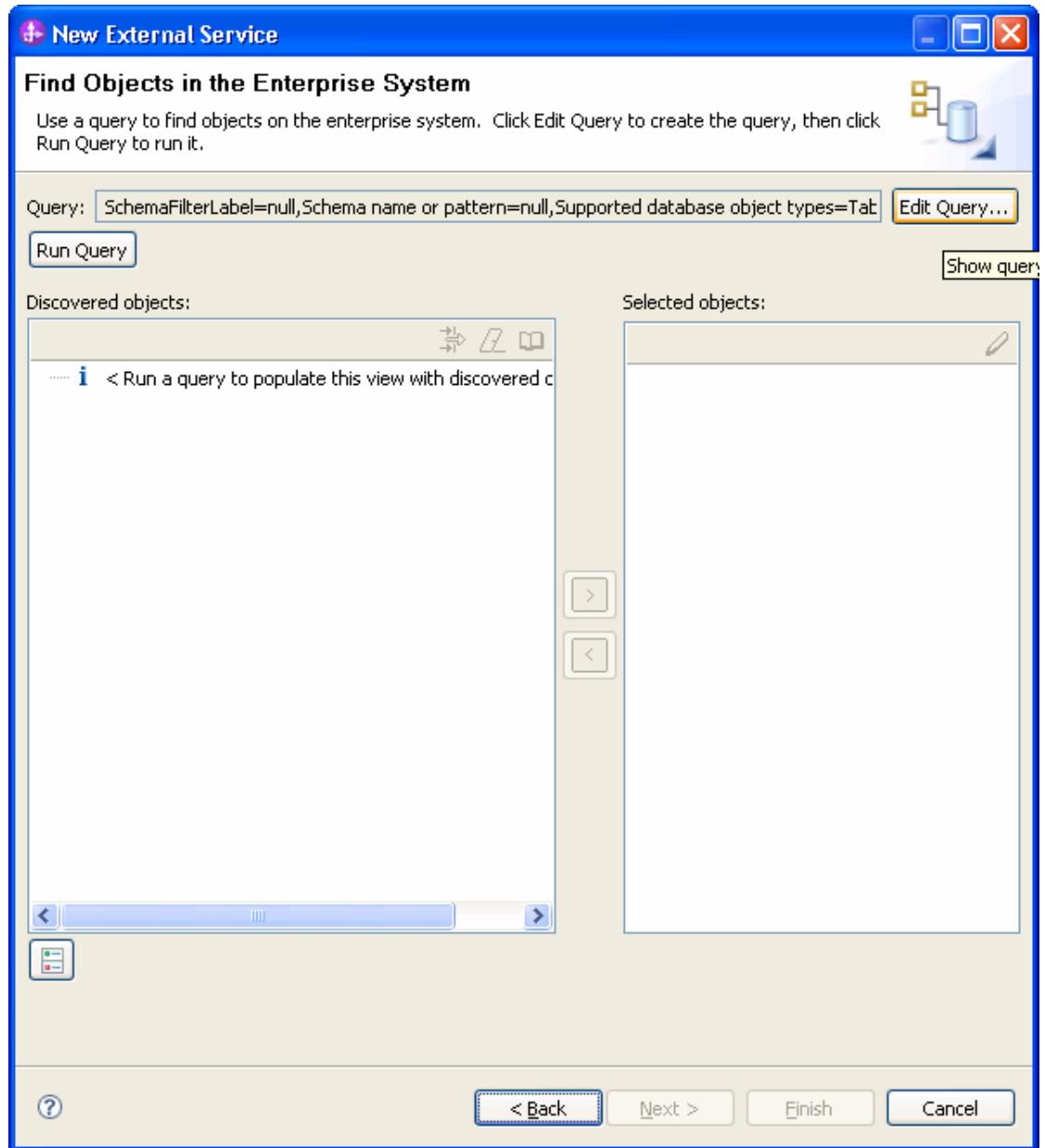
1. Expand the **Oracle** node from **Database system connection information** then select **10**.
2. Enter **System ID**, **Host name**, **Port number**, **User name** and **Password** fields, and then click **Next**.



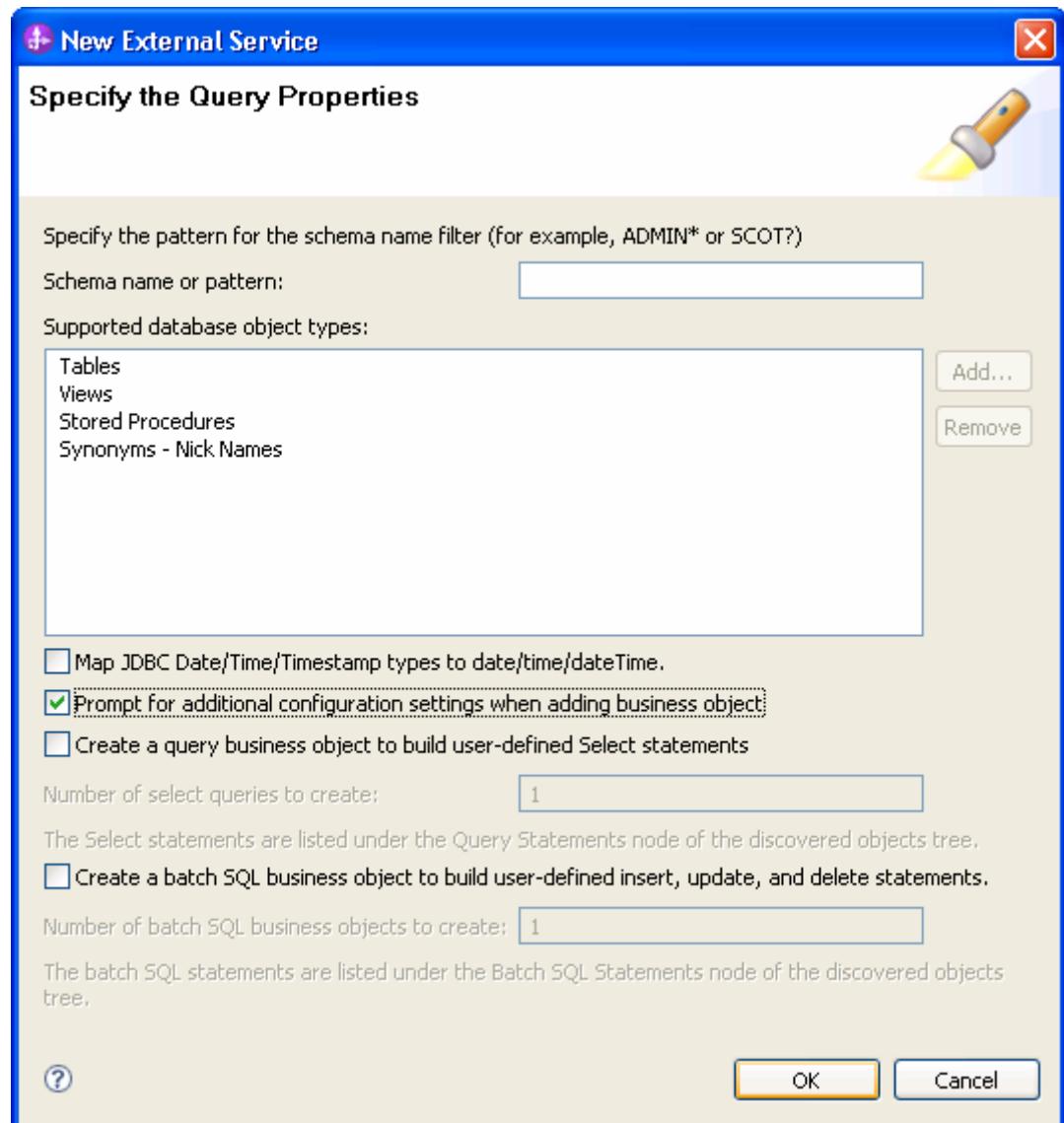
Select the business objects to be used with the adapter

Follow these steps to select the **Customer** and **Address** business object:

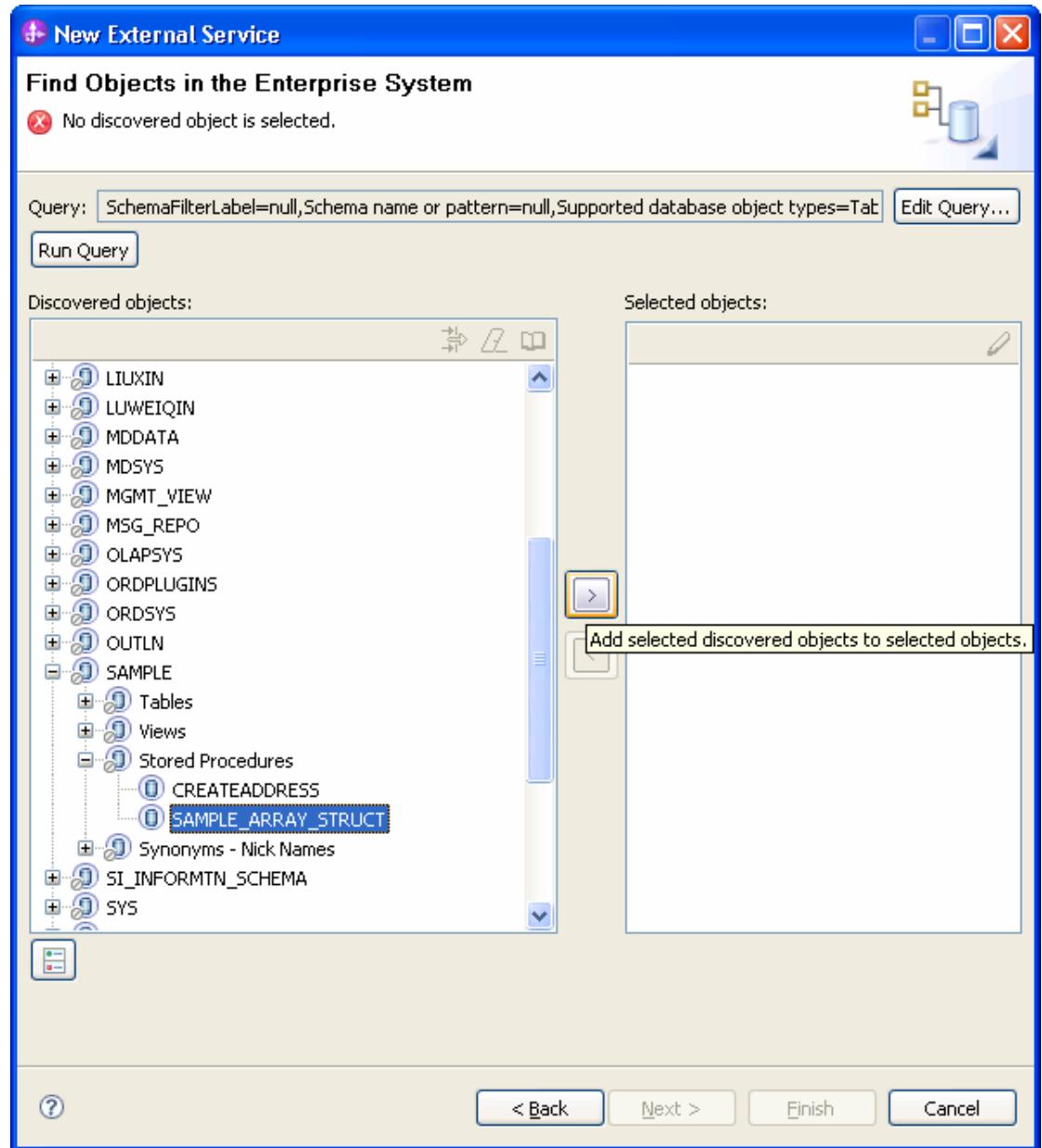
1. In the Object Discovery and Selection screen, click **Edit Query**.



2. In the Specify the Query Properties window, select the **Prompt for additional configuration settings when adding business objects** check box and click **OK**.



3. Click **Run Query**.
4. Expand the **SAMPLE** (for this tutorial only) node and select **Stored Procedures** and expand it.
5. Select **SAMPLE_ARRAY_STRUCT** from stored procedures and click .



6. In the Specify the Configuration Properties for 'SAMPLE_ARRAY_STRUCT' window, accept the default value for the **Maximum number of result sets returned** field and check whether the data types of parameters are correct.

New External Service

Specify the Configuration Properties for 'SAMPLE_ARRAY_STRUCT'

Generate a business object for the stored procedure

Business object

Stored procedure name: SAMPLE.SAMPLE_ARRAY_STRUCT

The maximum number of ResultSets returned from the stored procedure.: 0

Attributes

PKEY

Data type: string

Sample Value: |

ARR

Data type: ARRAY

Type name: * SAMPLE.ARRAYTYPE

Attributes

Attribute1

Data type: string

Sample Value: |

STRT

Data type: STRUCT

Type name: * SAMPLE.STRUCTTYPE

Attributes

EMPID

Data type: string

NAME

Data type: string

TITLE

Data type: string

Returned ResultSets

None:

Validate the stored procedure

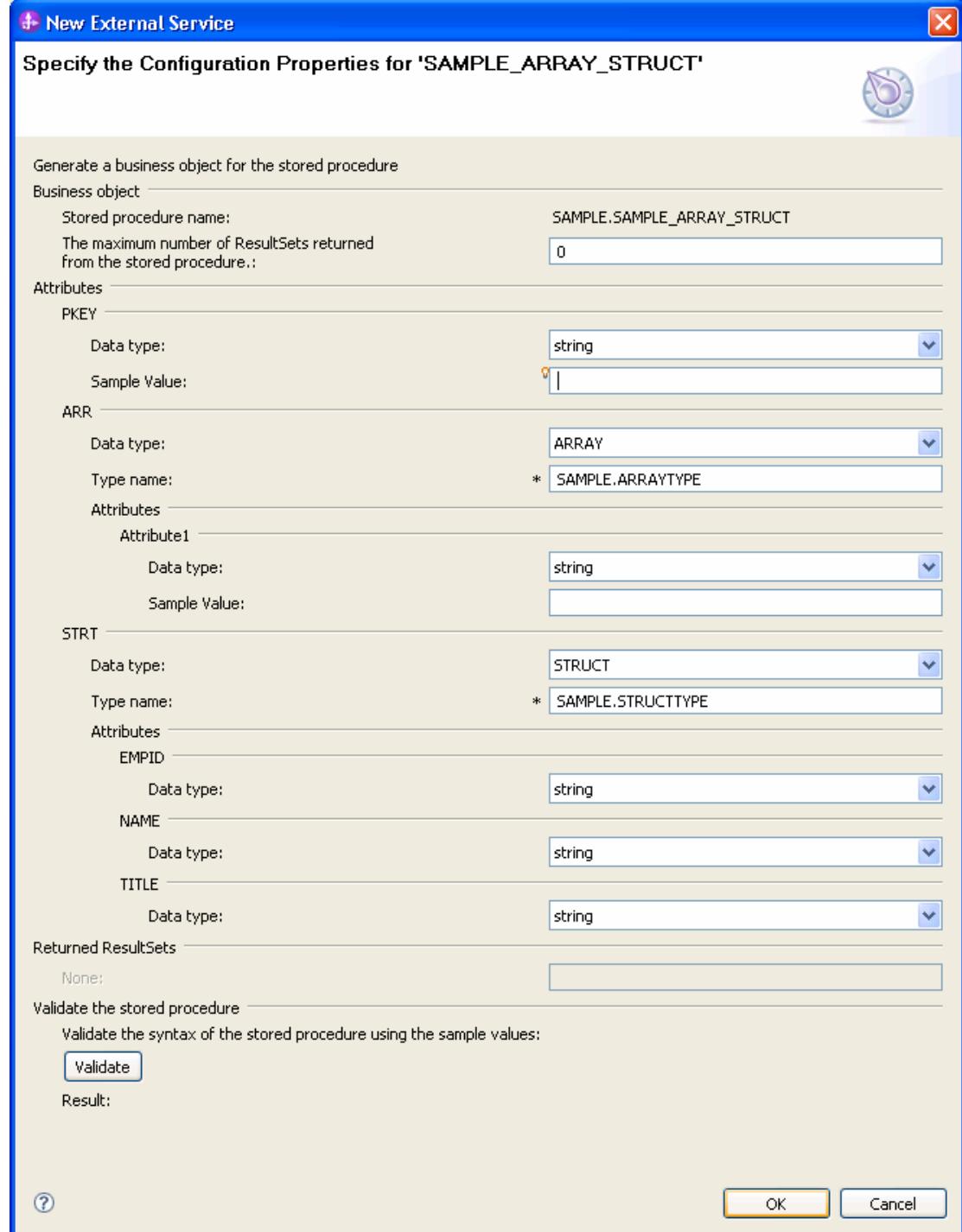
Validate the syntax of the stored procedure using the sample values:

Validate

Result:

?

OK Cancel



7. Enter sample values for the stored procedure input types, click **Validate** to verify if the stored procedure executes successfully. Check the **Result** to verify the result of validation and click **OK**.

New External Service

Specify the Configuration Properties for 'SAMPLE_ARRAY_STRUCT'

Generate a business object for the stored procedure

Business object

Stored procedure name: SAMPLE.SAMPLE_ARRAY_STRUCT

The maximum number of ResultSets returned from the stored procedure.: 0

Attributes

PKEY

Data type: string
Sample Value: 100

ARR

Data type: ARRAY
Type name: * SAMPLE.ARRAYTYPE

Attributes

Attribute1

Data type: string
Sample Value: s1

STRT

Data type: STRUCT
Type name: * SAMPLE.STRUCTTYPE

Attributes

EMPID

Data type: string

NAME

Data type: string

TITLE

Data type: string

Returned ResultSets

None:

Validate the stored procedure

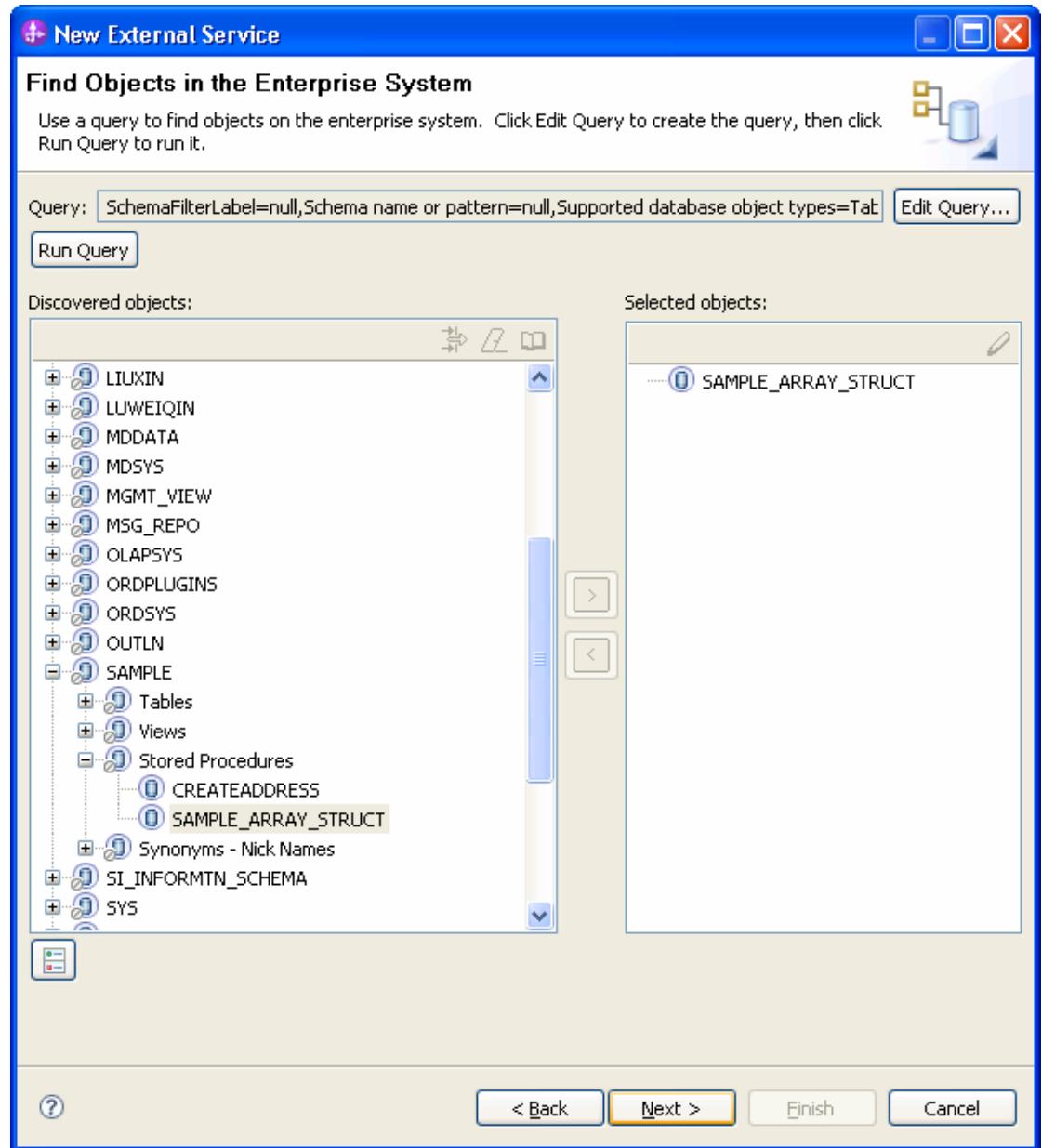
Validate the syntax of the stored procedure using the sample values:

Validate

Result: Validation was successful.

OK **Cancel**

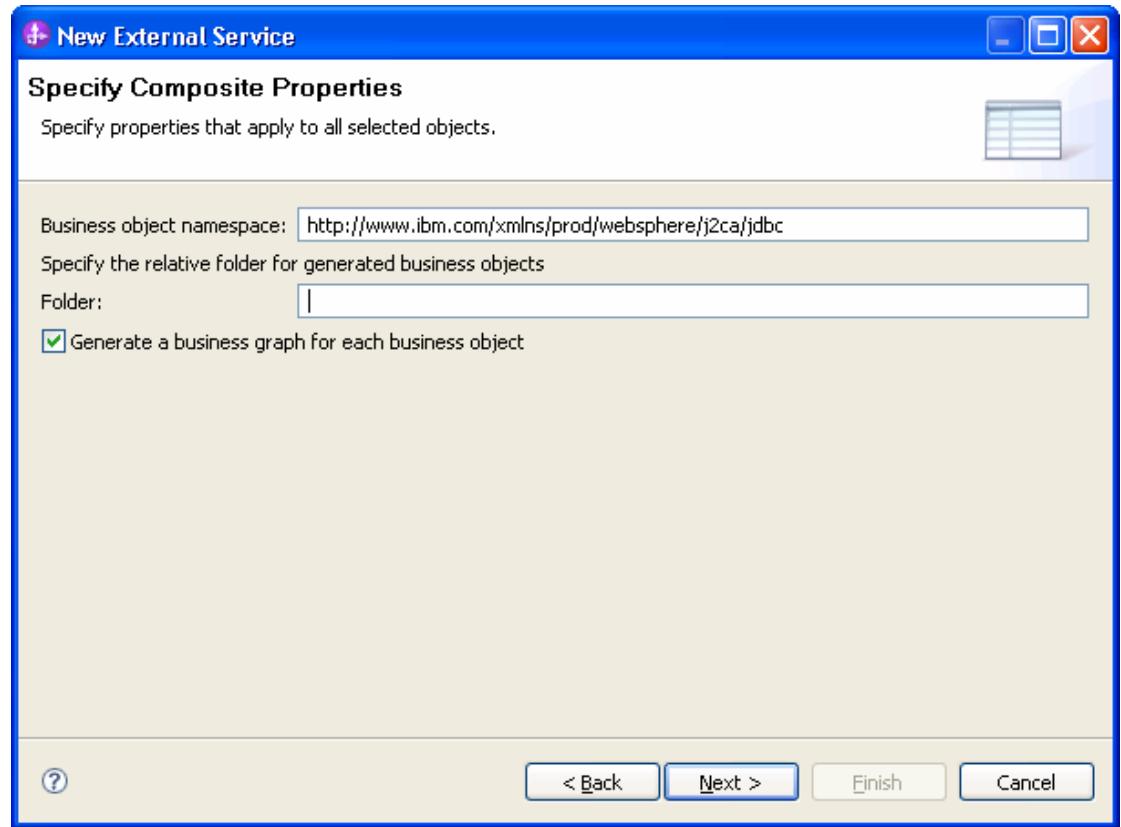
8. Click **Next**.



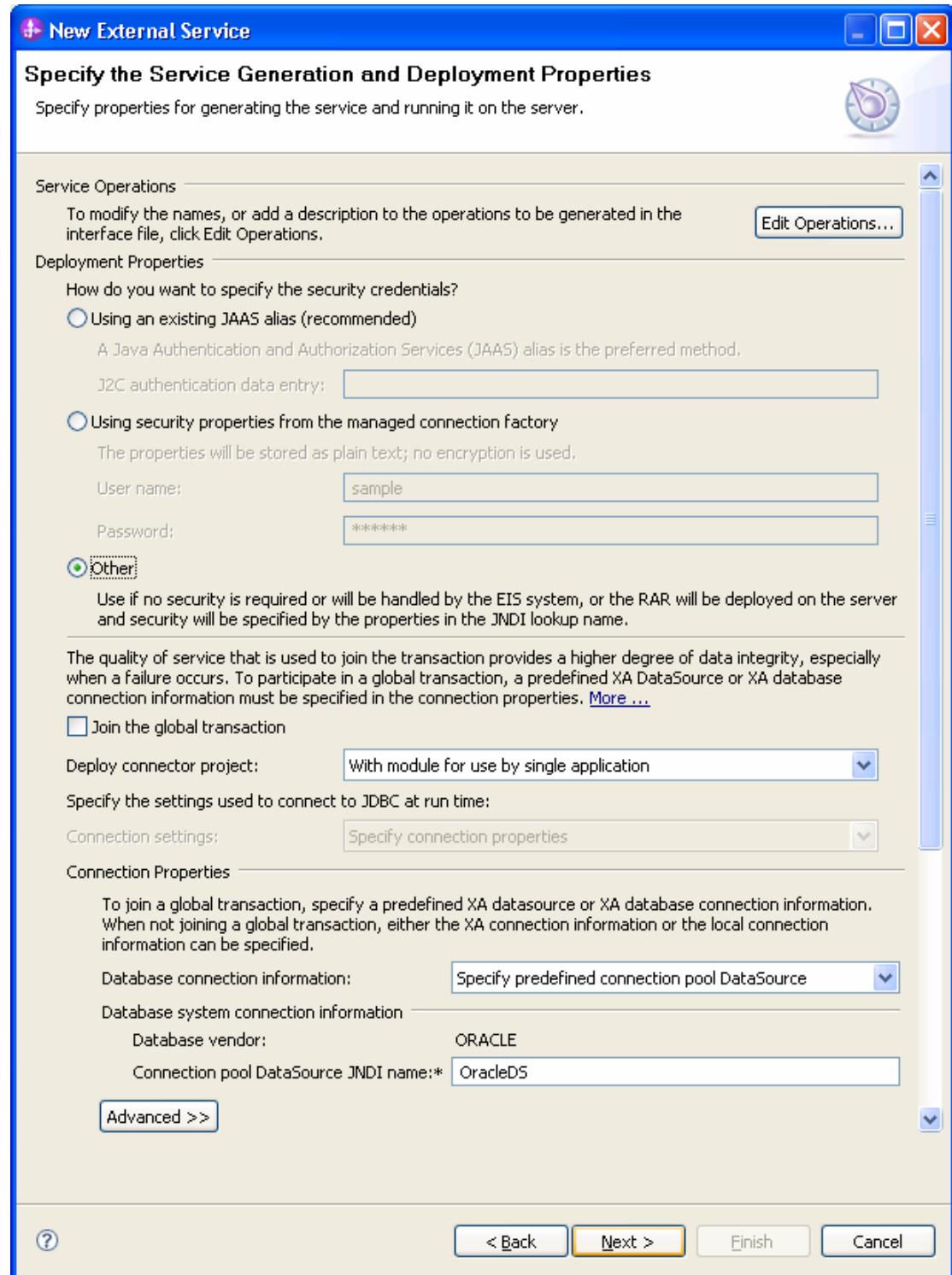
Generate business object definitions and related artifacts

Follow these steps to generate the business object definitions.

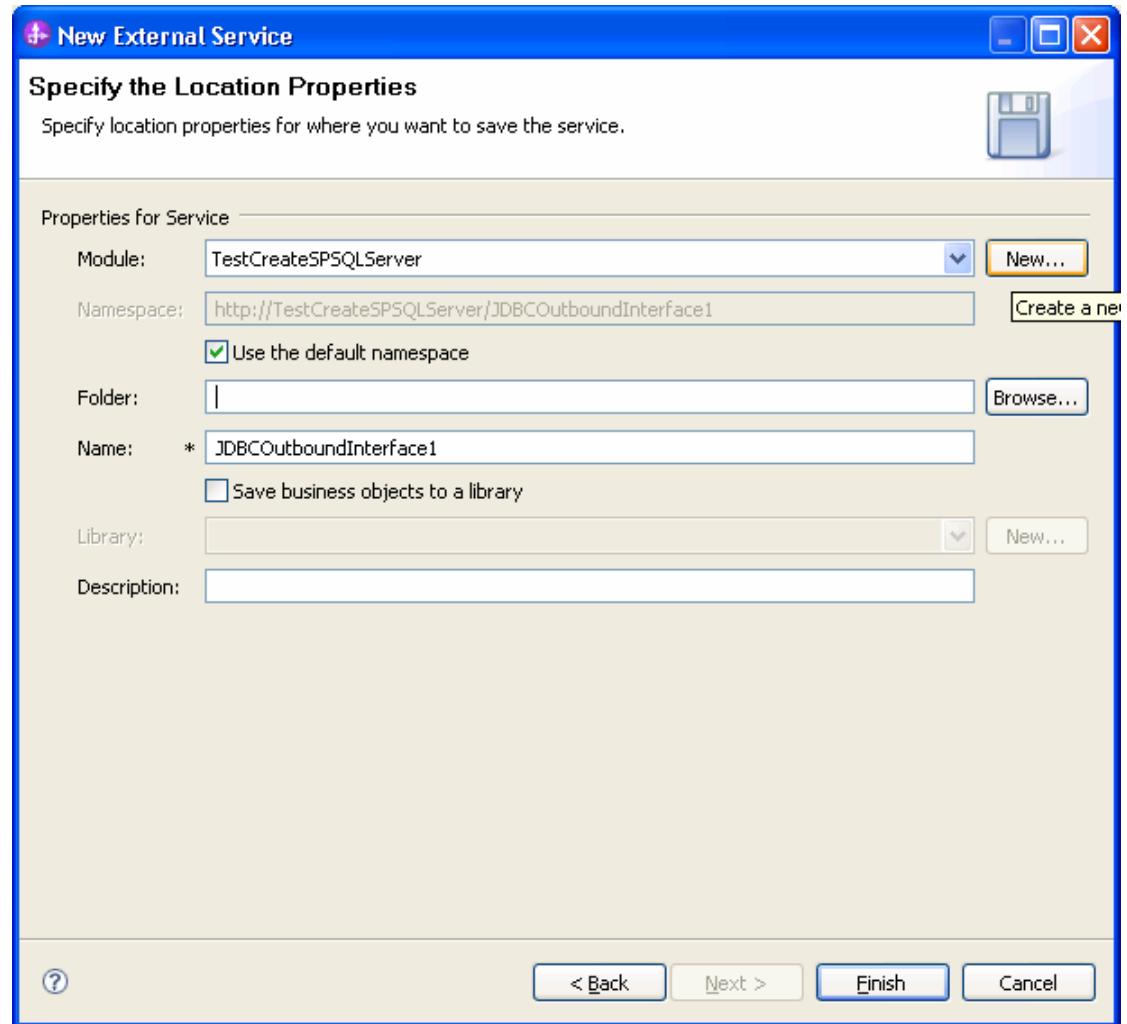
1. In the Specify Composite Properties window, accept the default values for the all fields and click **Next**.



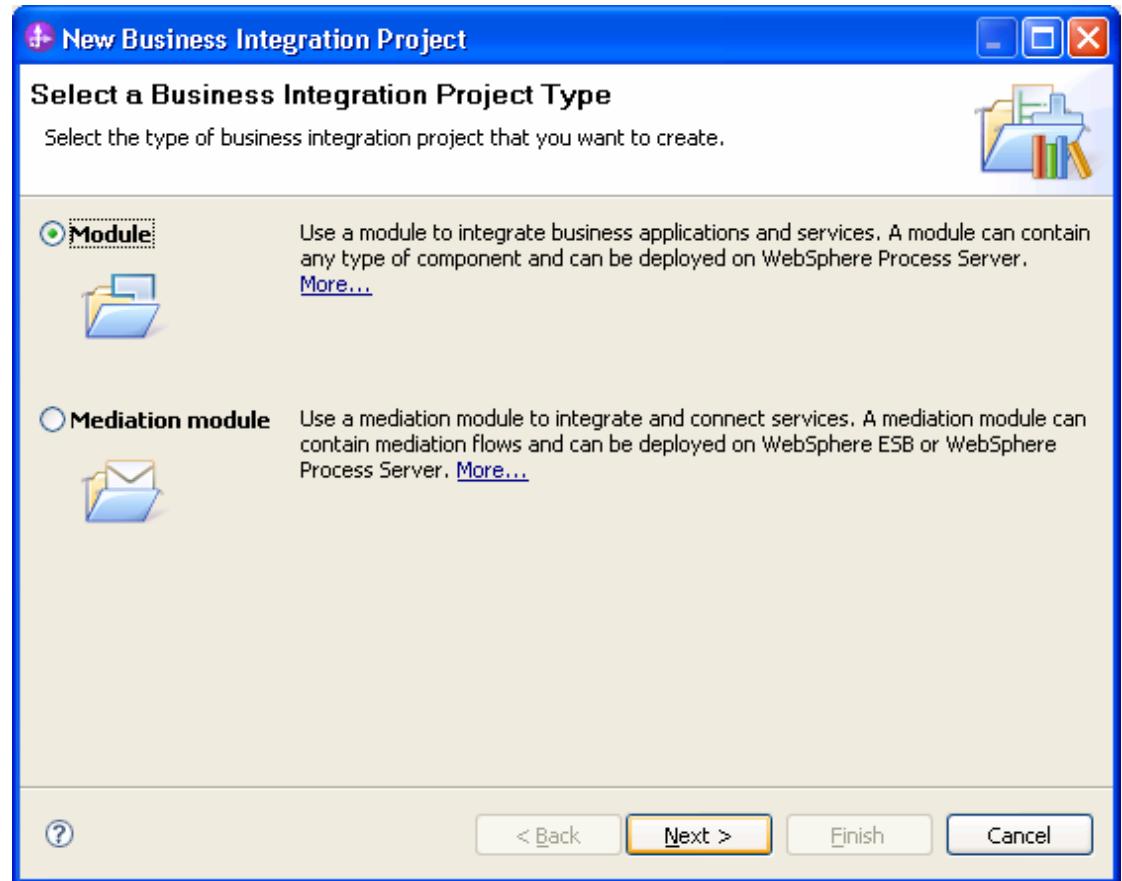
2. In the Specify the Service Generation and Deployment Properties window, perform the following steps:
 - a) Select **Other** for security options under **Deployment Properties**.
 - b) Clear the **Join the global transaction** check box.
 - c) Select **Specify predefined connection pool DataSource** from the **Database connection information** list.
 - d) Enter **OracleDS** in the **Connection pool DataSource JNDI Name** field, and click **Next**.



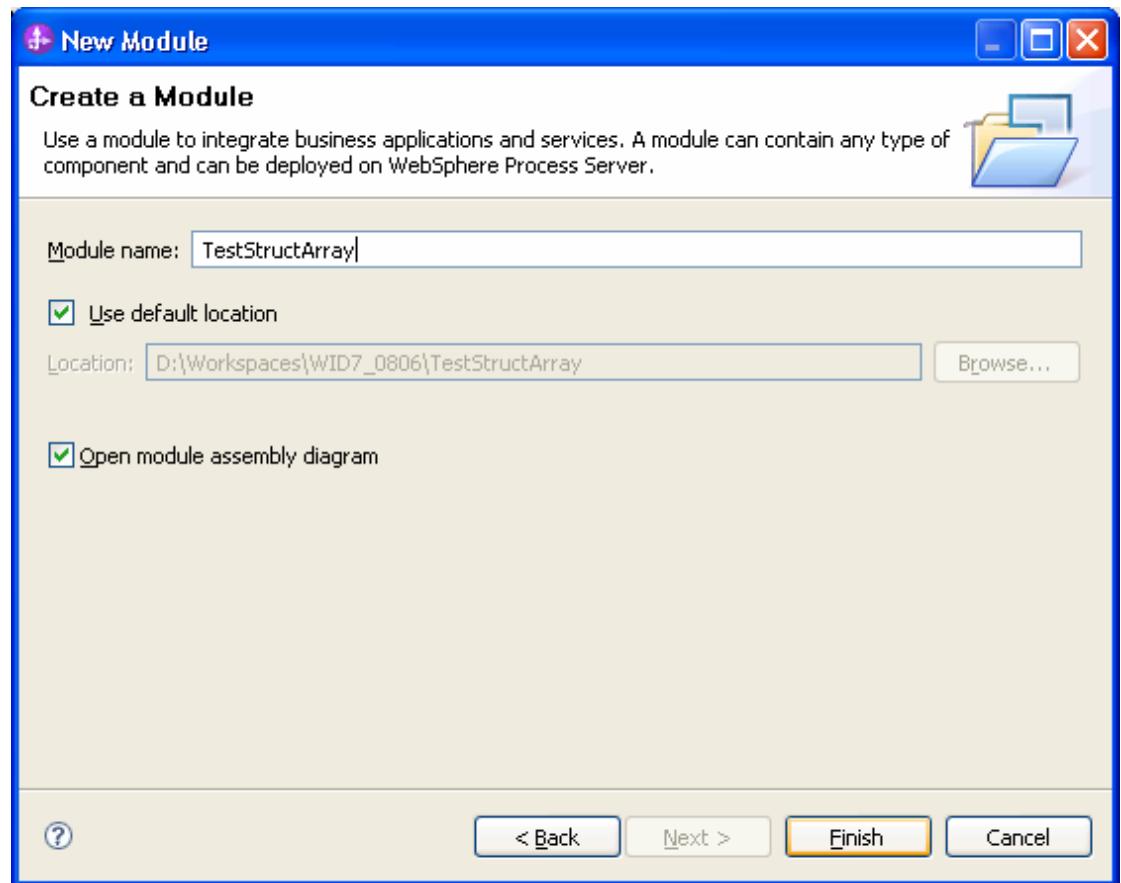
3. Click **New** in the Specify the Location Properties window.



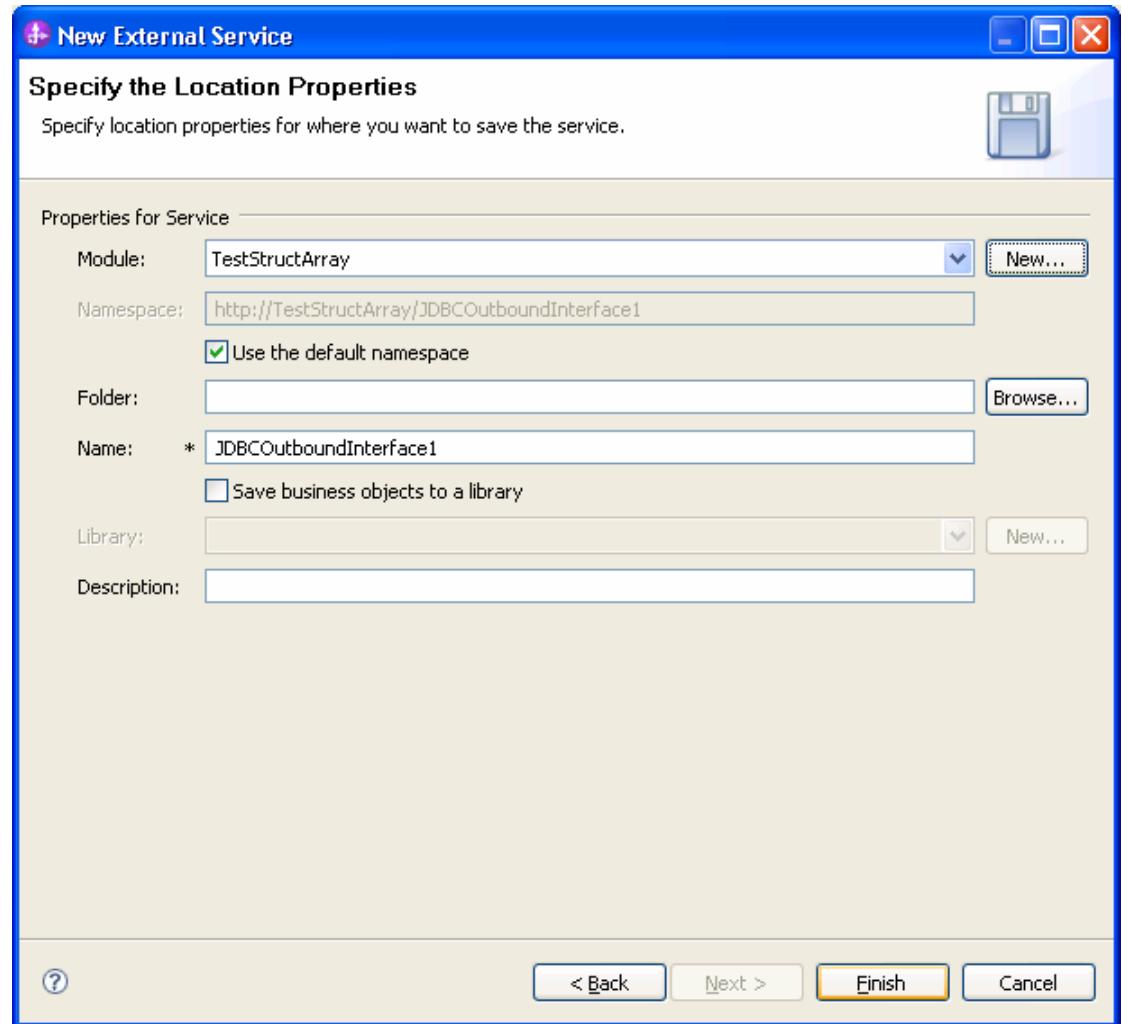
4. In the Select a Business Integration Project Type window, select **Module** and click **Next**.



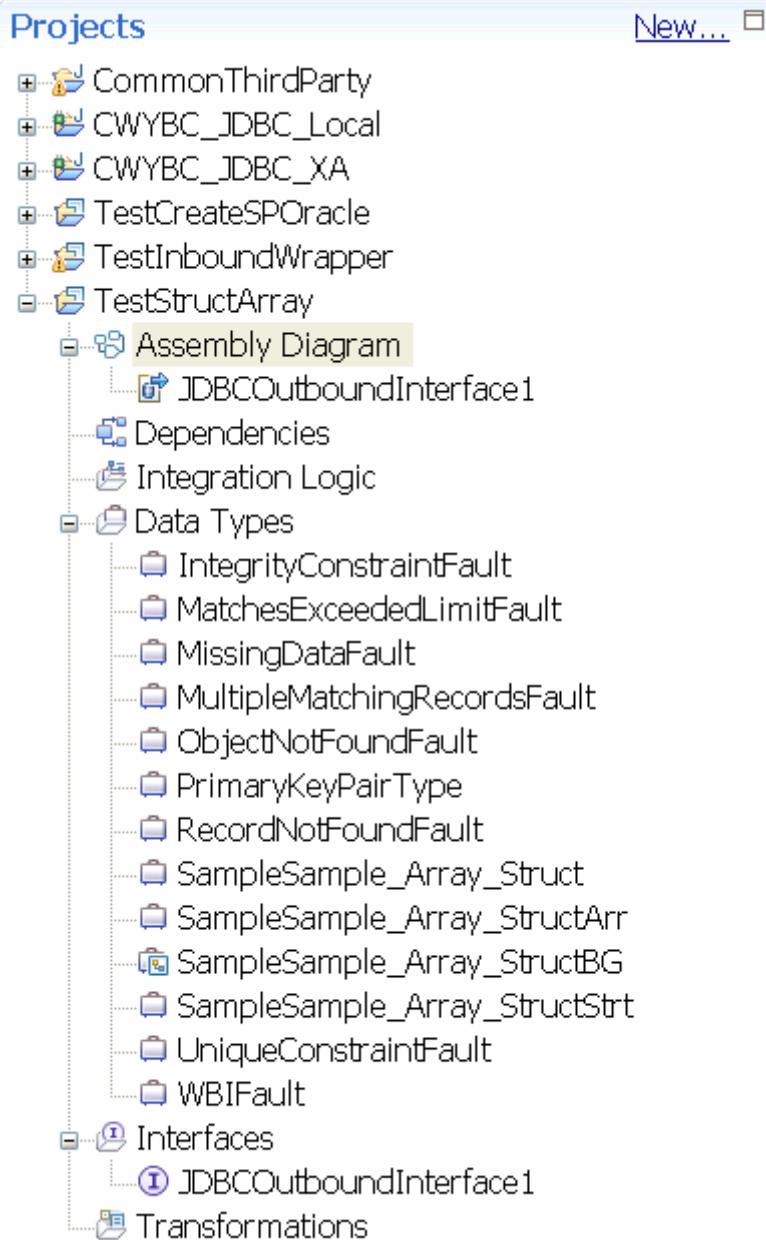
5. In the Create a Module window, type **TestStructArray** in the **Module Name** field and click **Finish**.



6. Click **Finish** to complete service creation.



7. Expand the created Business Integration Project and verify whether the artifacts are generated correctly.



Deploy the module to the test environment

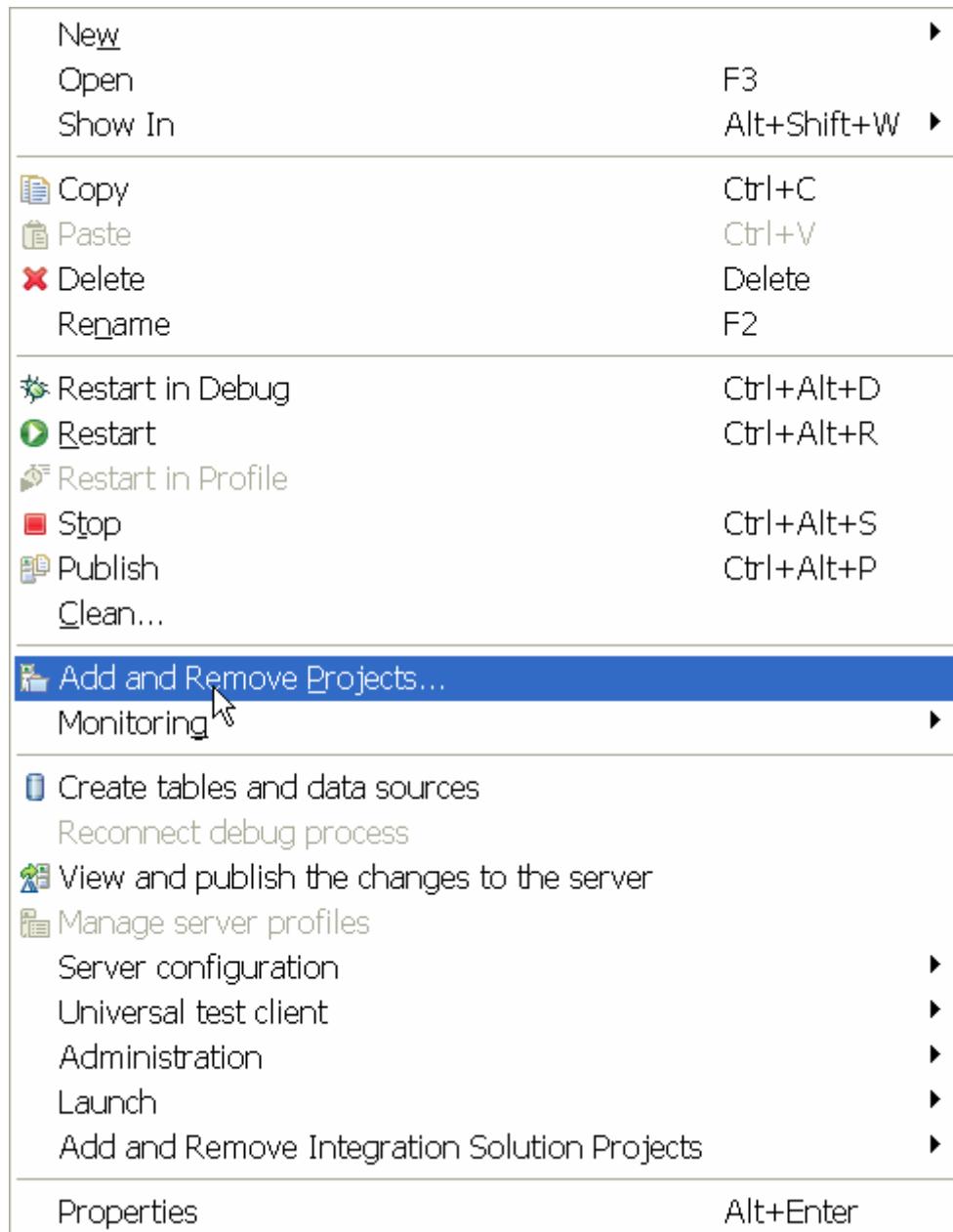
After running the external service wizard, you will have an SCA module that contains an Enterprise Information System import. You must install this SCA module in the WebSphere Integration Developer integration test client. To do this, you must add the SCA module you created earlier to the server using the **Servers** view in WebSphere Integration Developer.

Steps for adding the SCA module to the server:

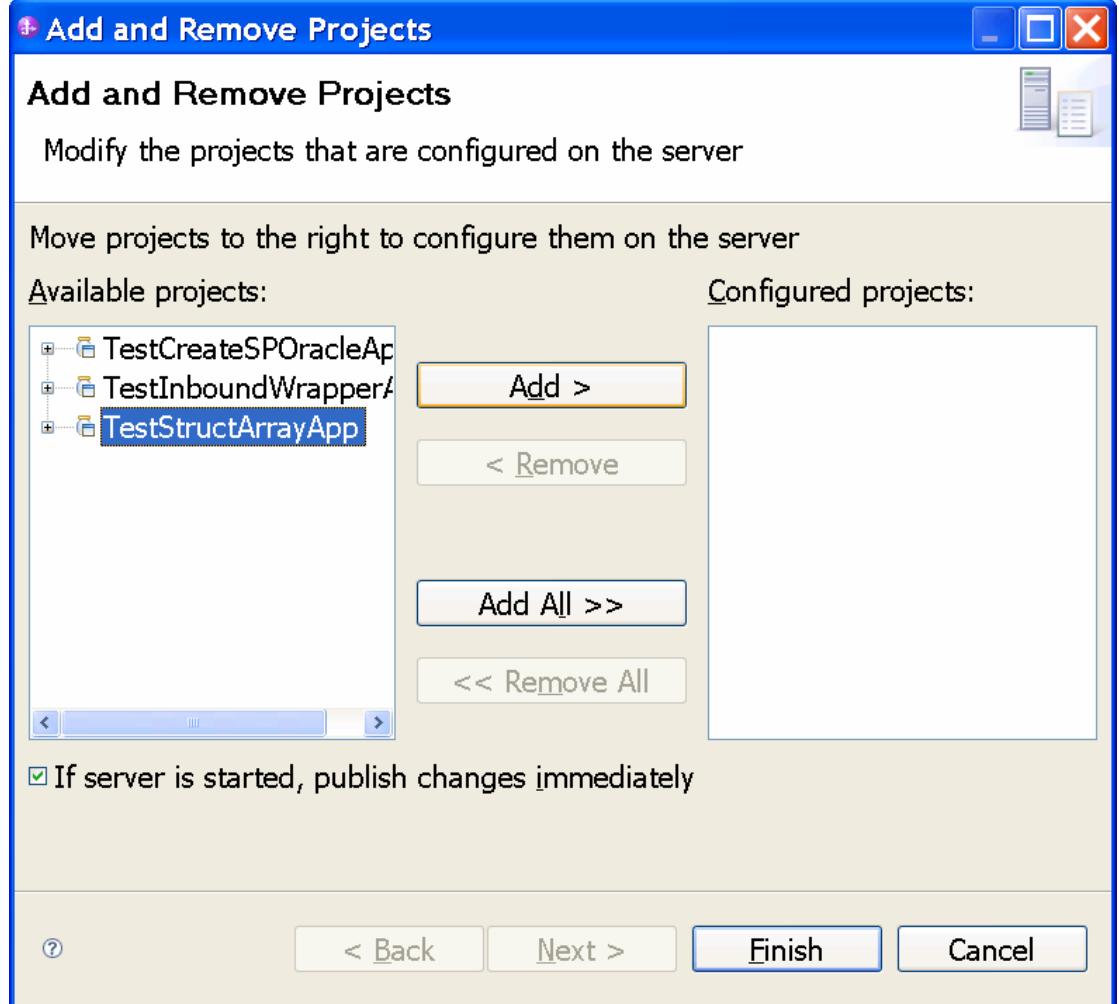
1. In WebSphere Integration Developer, switch to the **Servers** view by selecting **Windows > Show View > Servers**.

WebSphere software

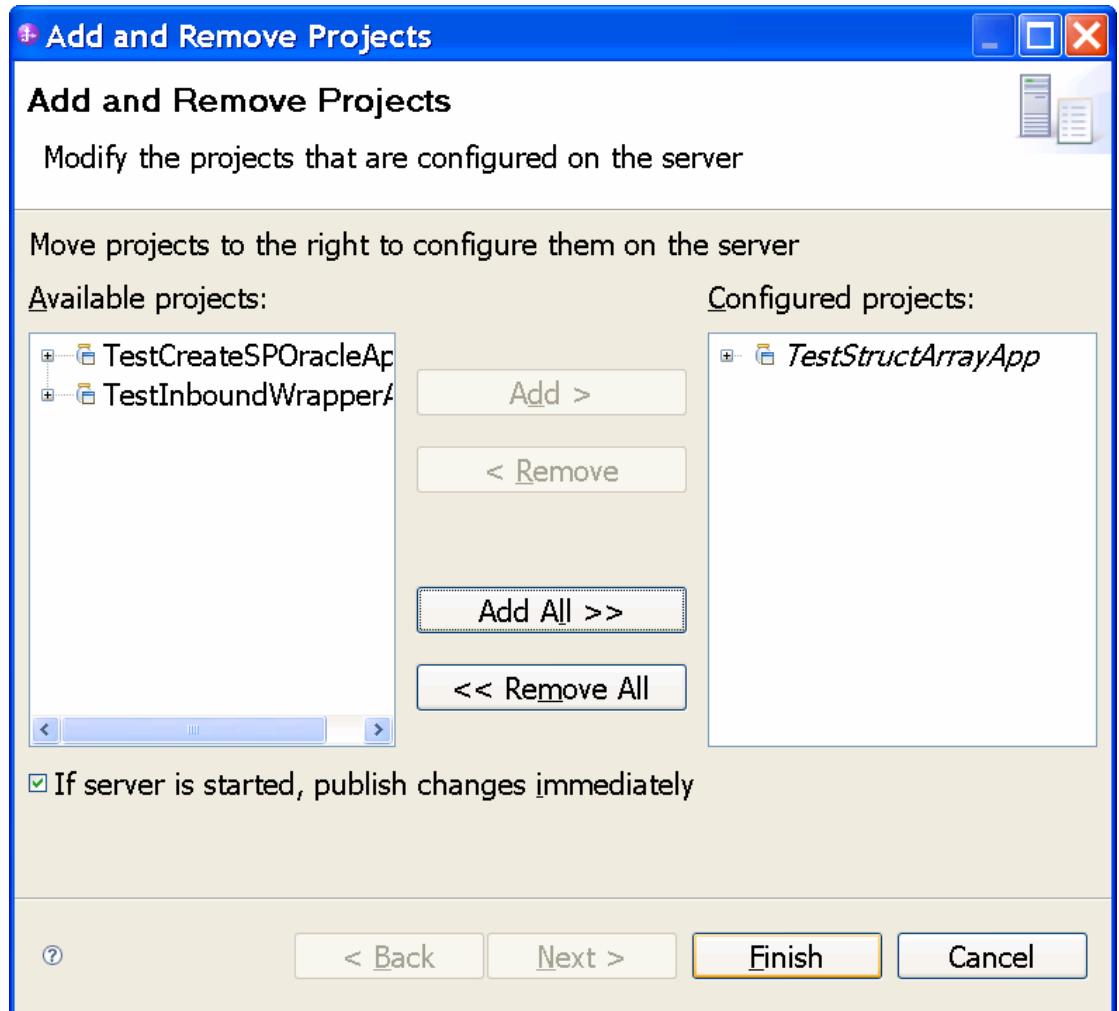
2. In the Servers tab in the lower-right pane of the WebSphere Integration Developer screen, right-click the server, and select **Start**.
3. After the server is started, right-click the server, and select **Add and Remove projects**.



The Add and Remove Projects window lists the available projects in the WebSphere Integration Developer workspace.



4. Select your project (**TestStructArrayApp**) and click **Add** to configure the project on the server. Click **Finish**.

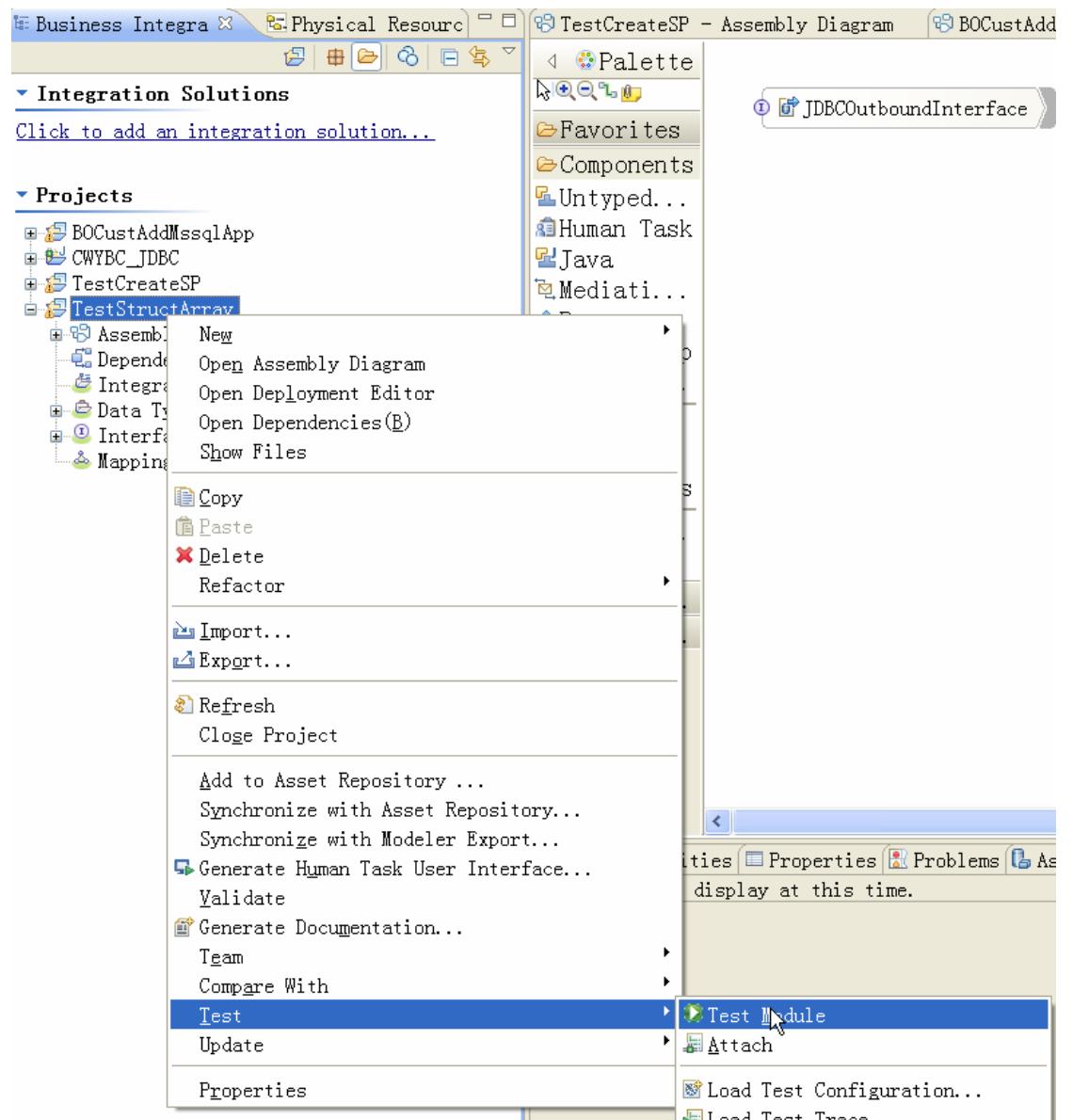


Test the assembled adapter application

Test the assembled adapter application using the WebSphere Integration Developer integration test client.

1. Select the **TestStructArray** module, right-click, and select **Test > Test Module**. The Test Client window is displayed.

WebSphere software



There is only one supported operation for business objects created for stored procedures. The **executeSampleSample_Array_StructBG** operation is selected by default.

WebSphere software

General Properties

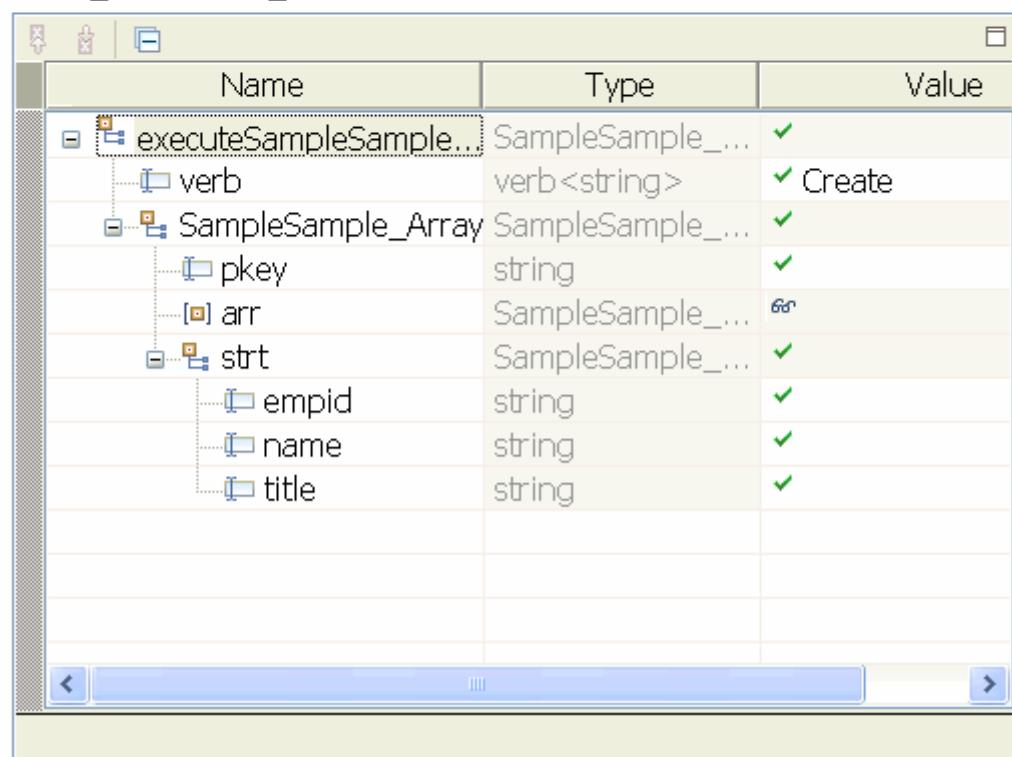
Detailed Properties

Specify the component, interface, operation, and input parameter values for the Invoke event, then click the Continue icon in the Events area to run the test. [More...](#)

<u>Configuration:</u>	Default Module Test
<u>Module:</u>	TestStructArray
<u>Component:</u>	JDBCOutboundInterface1
<u>Interface:</u>	JDBCOutboundInterface1
<u>Operation:</u>	executeSampleSample_Array_StructBG

Initial request parameters:

Value editor XML editor



Name	Type	Value
executeSampleSample...	SampleSample_...	✓
verb	verb<string>	✓ Create
SampleSample_Array	SampleSample_...	✓
pkey	string	✓
arr	SampleSample_...	✓
strt	SampleSample_...	✓
empid	string	✓
name	string	✓
title	string	✓

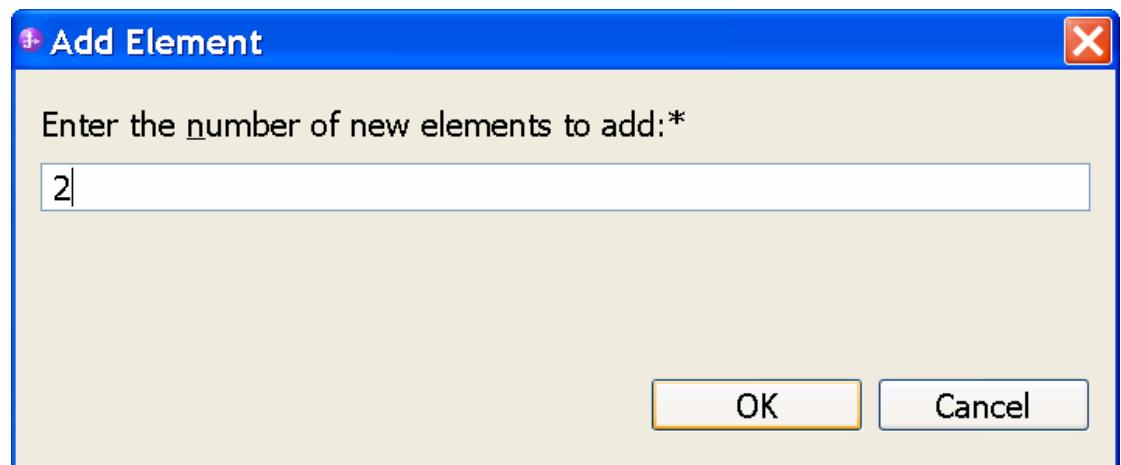
2. Right-click **arr** and click **Add Elements**.

Initial request parameters:

Value editor XML editor

Name	Type	Value
executeSampleSample_F	SampleSample_...	✓
verb	verb<string>	✓ Create
SampleSample_Array	SampleSample_...	✓
pkey	string	✓
arr		SampleSample
strt		
empid		
name		
title		

- In the Add Element window enter 2, to create two child objects and click **OK**.



- Populate data for the input pkey and the two array child attributes as shown in the figure.

Initial request parameters:

Value editor XML editor

The screenshot shows the WebSphere Value editor interface. At the top, there are three icons: a magnifying glass, a folder, and a file. Below them is a toolbar with buttons for 'New', 'Open', 'Save', 'Print', and 'Exit'. The main area is a table with three columns: 'Name', 'Type', and 'Value'. The 'Name' column contains hierarchical paths, and the 'Type' column shows the data type for each parameter. The 'Value' column contains the current value or status of each parameter. A green checkmark indicates a valid value, while a red 'X' indicates an invalid or unset value.

Name	Type	Value
executeSampleSamp	SampleSamp...	✓
verb	verb<string>	✓ Create
SampleSample_A	SampleSamp...	✓
pkey	string	✓ 100
arr	SampleSamp...	✗
arr[0]	SampleSamp...	✓
attribute	string	✓ abc
arr[1]	SampleSamp...	✓
attribute	string	✓ xyz
strt	SampleSamp...	✓
empid	string	✓
name	string	✓
title	string	✓

5. Unset the value for the **strt** attribute, which is an output type by right-clicking on **strt** and select **Set To→Unset**.

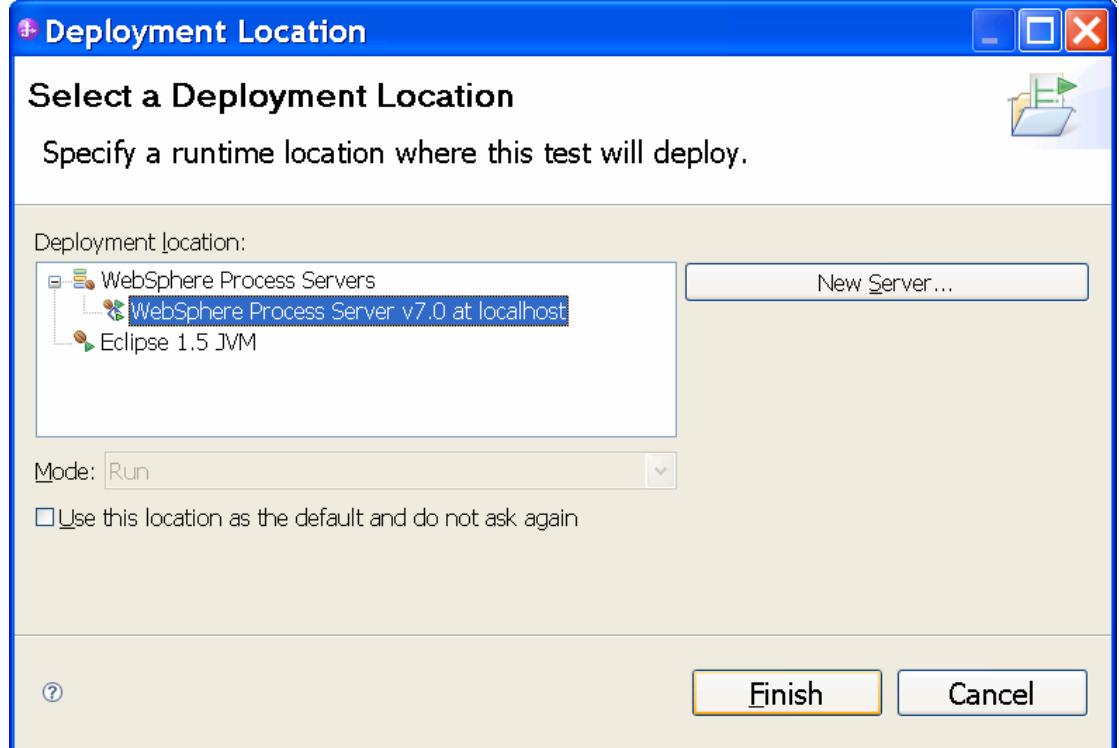
Initial request parameters:
 Value editor XML editor

Name	Type	Value
executeSampleSampl	SampleSamp...	✓
verb	verb<string>	✓ Create
SampleSample_/_	SampleSamp...	✓
pkey	string	✓ 100
arr	SampleSamp...	✓
arr[0]	SampleSamp...	✓
attribute	string	✓ abc
arr[1]	SampleSamp...	✓
attribute	string	✓ xyz
str	SampleSamp...	✓
empid		
name		
title		

Context menu options (from top to bottom):
Copy Value
Paste Value
Select All
Set To
Set Required to Default
Add Value to Pool...
Use Value from Pool...

Sub-menu for 'Set To':
Value...
Default
Unset
Null

6. Execute the service by click .
7. In the Select Deployment location window, select the server, and click **Finish**.



- Check the output of the service, and check the data in the Enterprise Information System to ensure it matches expected values.

Return parameters:

Name	Type	Value
executeJabdullaSa...	JabdullaSample_Array_S...	✓
verb	verb<string>	✓
JabdullaSample_Arr:	JabdullaSample_Array_S...	✓
pkey	string	✓ 100
arr	JabdullaSample_Array_S...	✓
arr[0]	JabdullaSample_Array_S...	✓
attribute1	string	✓ abc
arr[1]	JabdullaSample_Array_S...	✓
attribute1	string	✓ xyz
strt	JabdullaSample_Array_S...	✓
empid	string	✓ 10
name	string	✓ xyz
title	string	✓ SE

Clear the sample content

After you have tested the application, clear the sample content to return the data to its original state.

Chapter 5. Tutorial 4: Sending Data to Enterprise Information System using BatchSQL (Oracle)

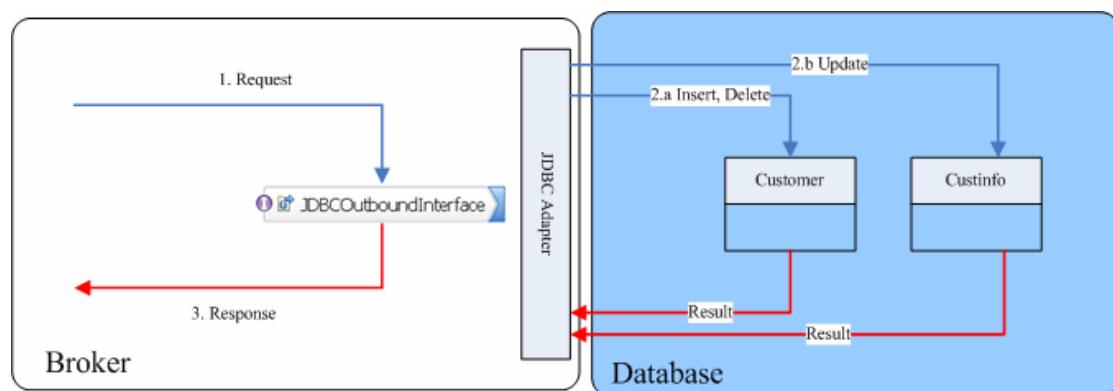
This tutorial demonstrates how to create batch SQL business object to execute multiple SQL statements using WebSphere Adapter for JDBC 7.0.0.0.

About this task

In this scenario, an application SCA component raises a batch SQL execution request to the JDBC Outbound Interface. The JDBC adapter executes a batch SQL to complete the following database operations:

- Insert a record into CUSTOMER table
- Delete a record from CUSTOMER table
- Update a record in CUSTINFO table

Finally, the JDBC adapter returns the execution result to a SDO and sends a response to the SCA component. The following figure represents the scenario:



Prepare to run through the tutorial

Extract the sample files

Replicas of the artifacts that you create when using the external service wizard are provided as sample files for your reference. Use these files to verify that the files you create with the external service wizard are correct.

Download the sample zip file and extract it into a directory of your choice (you may want to create a new directory).

Configuration prerequisites

Before configuring the adapter, you must complete the following tasks:

- Create tables and records
- Create an authentication alias
- Create a data source

Create tables and records

You must create the following tables in the Oracle database before starting the scenario.

a. Script for creating the tables

```
CREATE TABLE CUSTINFO (
    CCODE VARCHAR2(10) NOT NULL PRIMARY KEY,
    CDATA VARCHAR2(20)) ;

CREATE TABLE CUSTOMER (
    PKEY VARCHAR2(10) NOT NULL PRIMARY KEY,
    FNAME VARCHAR2(20) ,
    LNAME VARCHAR2(20) ,
    CCODE VARCHAR2(10) ) ;
```

b. Script for inserting records into tables

Insert a record in Customer table.

```
INSERT INTO CUSTOMER (pkey,ccode,fname,lname)
values('Test', 'ANITA','MEHTA','IBM');
```

Insert a record in CUSTINFO table.

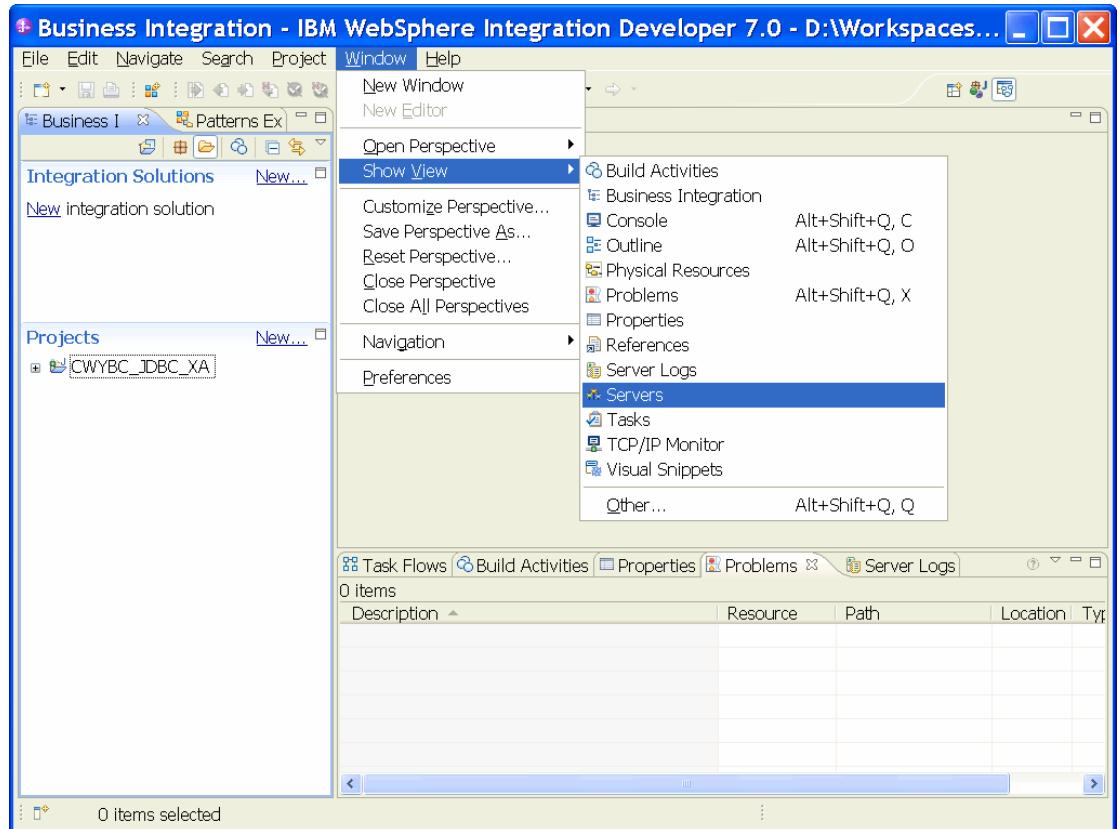
```
INSERT INTO CUSTINFO (ccode, cdata) values('Test1',
'ABC');
```

Create an authentication alias

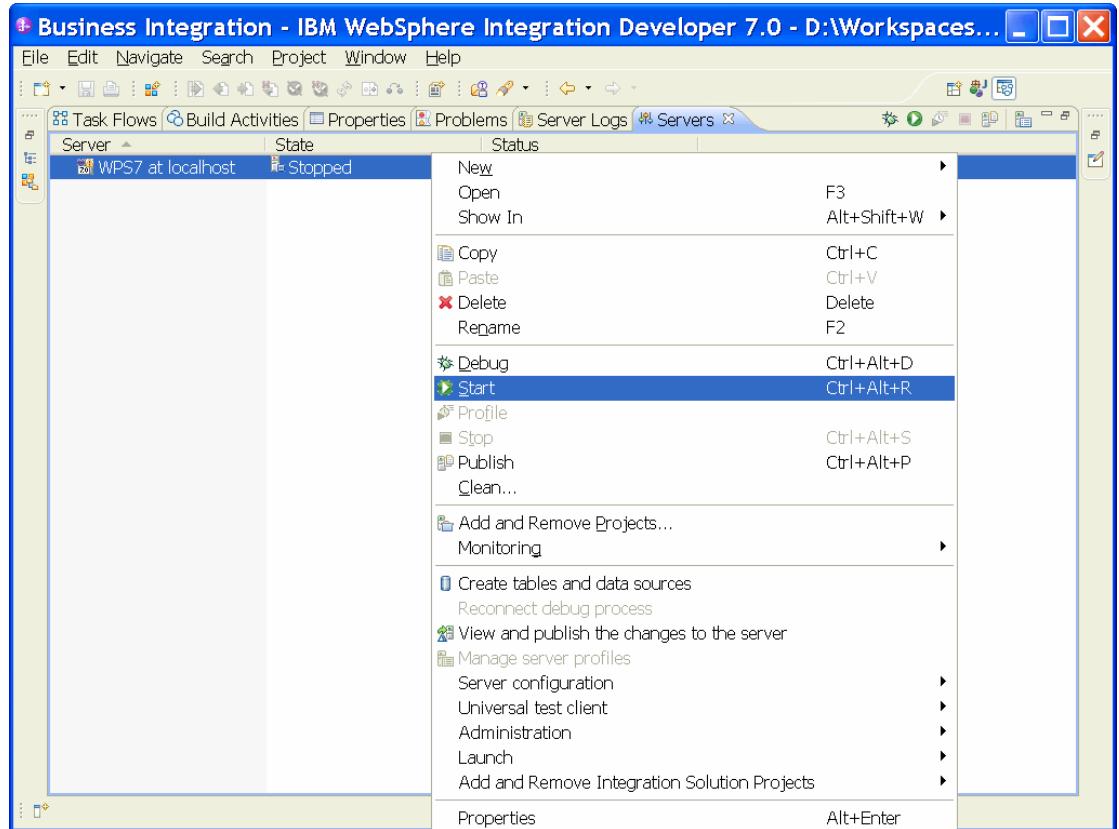
The authentication alias needs to be set because the data source created in the next section uses the username and password set in the authentication alias to connect to the database.

Follow these steps to set the authentication alias in the WebSphere Process Server administrative console.

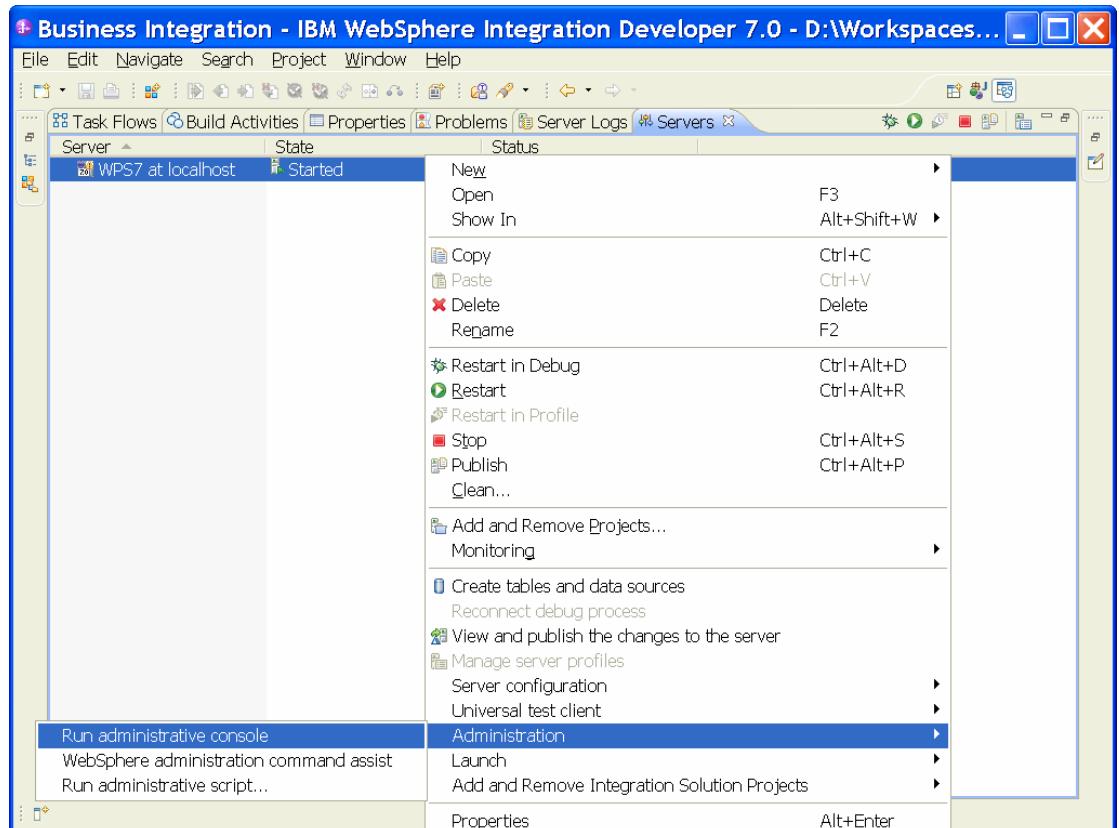
1. In WebSphere Integration Developer, switch to the **Servers** view by selecting **Windows > Show View > Servers**.



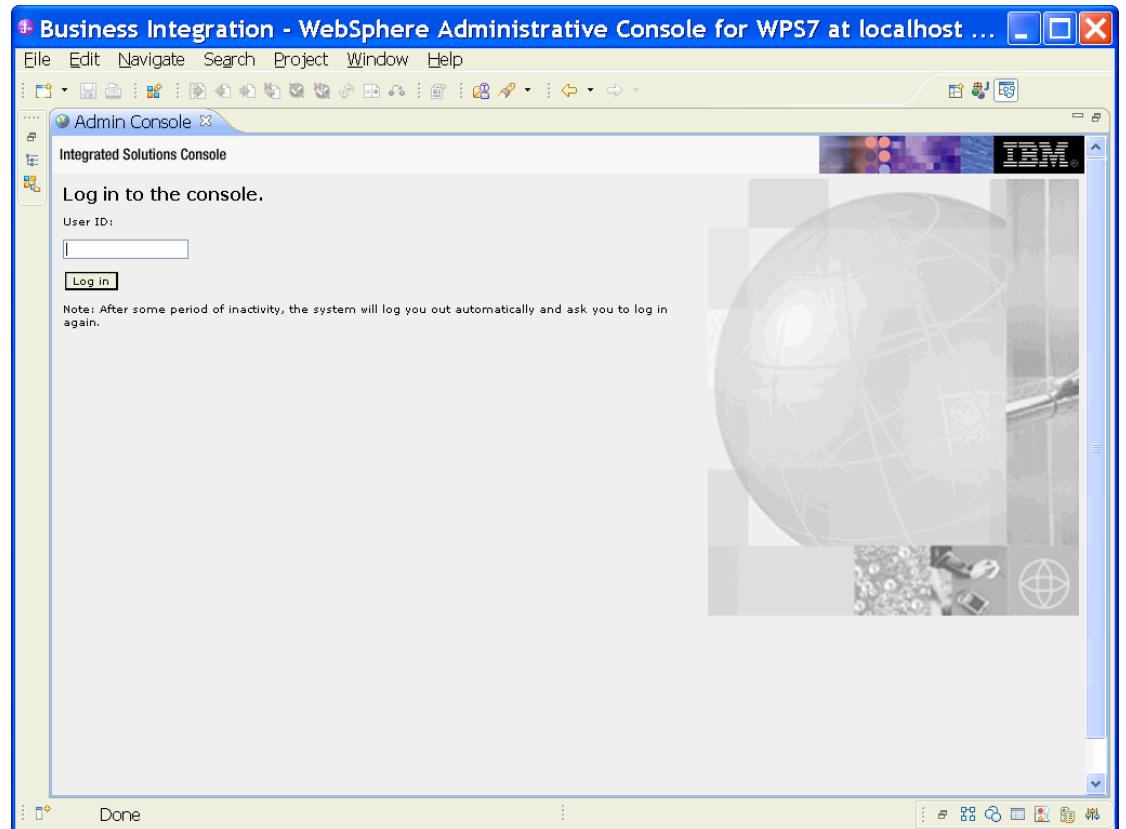
2. In the **Servers** view, right-click the server that you want to start and select **Start**.



- After the server is started, right-click the server, and select **Administration > Run administrative console**.



- Log on to the administrative console.



5. Click **Security → Global security**.

WebSphere software

The screenshot shows the left sidebar of the WebSphere software interface. At the top, there is a dropdown menu labeled "View: All tasks". Below it is a hierarchical navigation tree:

- Welcome
- Guided Activities
- Servers
- Applications
- Services
- Resources
- Security
 - Business Integration Security
 - Global security (highlighted with a red box)
 - Security domains
 - Administrative Authorization Groups
 - SSL certificate and key management
 - Security auditing
 - Bus security
- Environment
- Integration Applications
- System administration
- Users and Groups
- Monitoring and Tuning
- Troubleshooting
- Service integration
- UDDI

6. Under **Java Authentication and Authorization Service**, click **J2C authentication data**.

Global security

Use this panel to configure administration and the default application security policy. This security configuration applies to functions and is used as a default security policy for user applications. Security domains can be defined to override and control applications.

[Security Configuration Wizard](#)[Security Configuration Report](#)**Administrative security**

- Enable administrative security
 - [Administrative user roles](#)
 - [Administrative group roles](#)
 - [Administrative authentication](#)

Application security

- Enable application security

Java 2 security

- Use Java 2 security to restrict application access to local resources
 - Warn if applications are granted custom permissions
 - Restrict access to resource authentication data

User account repository

Current realm definition

Federated repositories

Available realm definitions

Federated repositories

[Configure...](#)[Set as current](#)**Authentication**

Authentication mechanisms and expiration

[LTPA](#)[Kerberos and LTPA](#)[Kerberos configuration](#)[SWAM \(deprecated\): No authentication](#)[Authentication cache settings](#) Web and SIP security RMI/IOP security Java Authentication and Authorization[Application logins](#)[System logins](#)[J2C authentication data](#)

Specifies a list of user identities and passwords for Java(TM) 2 connector security to use.

 [Security domains](#) [External authorization providers](#) [Custom properties](#)

A list of existing aliases is displayed.

WebSphere software

Global security > JAAS - J2C authentication data

Specifies a list of user identities and passwords for Java(TM) 2 connector security to use.

Prefix new alias names with the node name of the cell (for compatibility with earlier releases)

Apply

Preferences

New Delete			
Select	Alias ▾	User ID ▾	Description ▾
You can administer the following resources:			
<input type="checkbox"/>	BSpace JDBC Alias	wbiuser	Business Space Authorization Alias
<input type="checkbox"/>	CommonEventInfrastructureJMSAuthAlias	wbiuser	Authentication alias for the Common Event Infrastructure JMS Topics and Queues
<input type="checkbox"/>	SCA Auth Alias	wbiuser	This is the alias used by SCA to login to a secured SIBus
<input type="checkbox"/>	localhostNode01Cell/nlNode01/server1/EventAuthDataAliasDerby		Derby authentication alias for the Event Server
Total 4			

7. Click **New** to create a new authentication entry. Type the alias name, and username and password to connect to the database. Click **OK**.

WebSphere software

Cell=localhostNode01Cell, Profile=AppSrv01

Global security

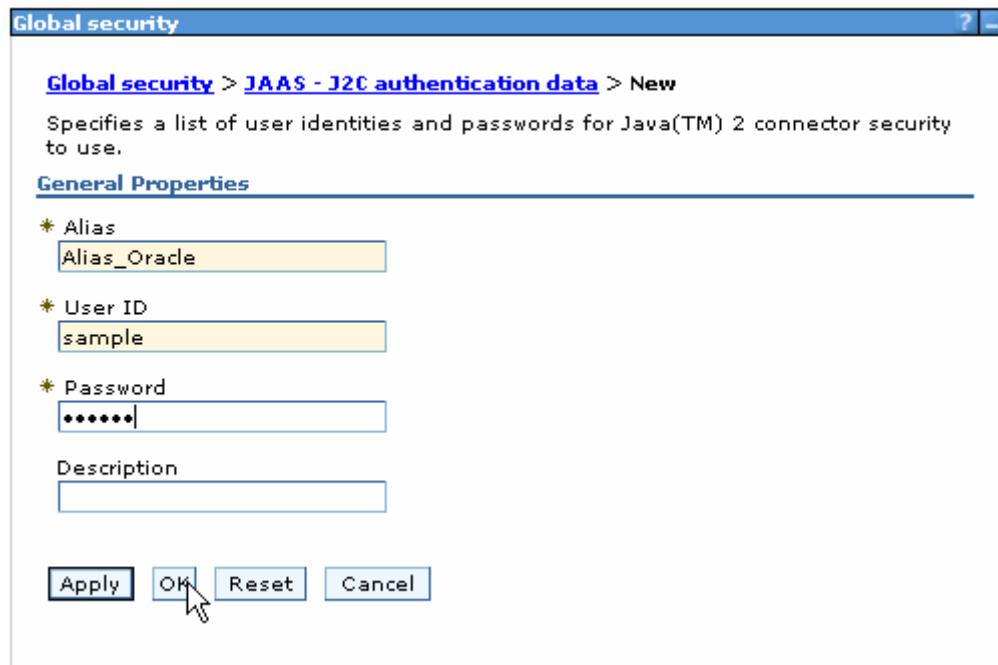
[Global security](#) > [JAAS - J2C authentication data](#) > New

Specifies a list of user identities and passwords for Java(TM) 2 connector security to use.

General Properties

* Alias: Alias_Oracle
* User ID: sample
* Password:
Description:

[Apply] [OK] [Reset] [Cancel]



8. Click **Save** to save the changes.

Cell=localhostNode01Cell, Profile=AppSrv01

Global security

[Global security](#) > [JAAS - J2C authentication data](#)

Specifies a list of user identities and passwords for Java(TM) 2 connector security to use.

Prefix new alias names with the node name of the cell (for compatibility with earlier releases)

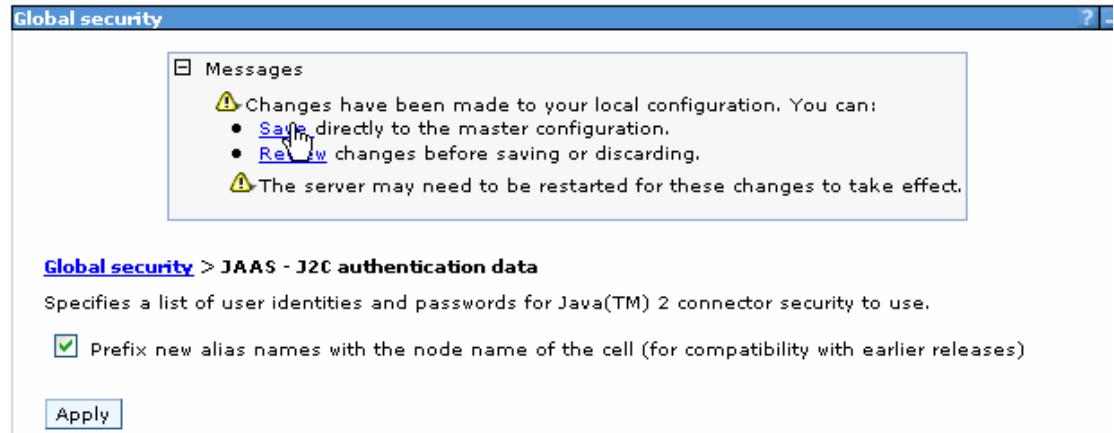
[Apply]

Messages

Changes have been made to your local configuration. You can:

- [Save](#) directly to the master configuration.
- [Review](#) changes before saving or discarding.

The server may need to be restarted for these changes to take effect.



You have created an authentication alias that will be used to configure the data source.

Preferences			
	New	Delete	
Select	Alias	User ID	Description
You can administer the following resources:			
<input type="checkbox"/>	BSpace JDBC Alias	wbiuser	Business Space Authorization Alias
<input type="checkbox"/>	CommonEventInfrastructureJMSAuthAlias	wbiuser	Authentication alias for the Common Event Infrastructure JMS Topics and Queues
<input type="checkbox"/>	SCA Auth Alias	wbiuser	This is the alias used by SCA to login to a secured SIBus
<input type="checkbox"/>	localhostNode01Cell/nlNode01/server1/EventAuthDataAliasDerby		Derby authentication alias for the Event Server
<input type="checkbox"/>	nlNode01/AliasOracle	luweiqin	
Total 5			

Create a data source

Create a data source in WebSphere Process Server, which the adapter will use to connect to the database. This data source is used later when generating the artifacts for the module.

Note: This tutorial uses Oracle as the database and the Oracle thin driver,ojdbc6.jar.

Here are the steps to create the data source in the WebSphere Process Server administrative console.

1. In the administrative console, select **Environment → WebSphere Variables**.

WebSphere software



2. On the right page, select **ORACLE_JDBC_DRIVER_PATH** and specify the path of the ojdbc6.jar file in the **Value** field. Click **OK**.

WebSphere software

Cell=localhostNode01Cell, Profile=AppSrv01

WebSphere Variables

[WebSphere Variables](#) > ORACLE_JDBC_DRIVER_PATH

Use this page to define substitution variables. Variables specify a level of indirection for some system-defined values, such as file system root directories. Variables have a scope level, which is either server, node, cluster, or cell. Values at one scope level can differ from values at other levels. When a variable has conflicting scope values, the more granular scope value overrides values at greater scope levels. Therefore, server variables override node variables, which override cluster variables, which override cell variables.

Configuration

General Properties

* Name: ORACLE_JDBC_DRIVER_PATH

Value: D:\Lib

Description: The directory that contains the Oracle thin or oci8 JDBC Driver.

Buttons: Apply, OK, Reset, Cancel

3. Click **Save** to save the changes.

WebSphere Variables

Messages

⚠ Changes have been made to your local configuration. You can:
• [Save](#) directly to the master configuration.
• [Review](#) changes before saving or discarding.
⚠ The server may need to be restarted for these changes to take effect.

The variable has been added and appears in the list.

Preferences

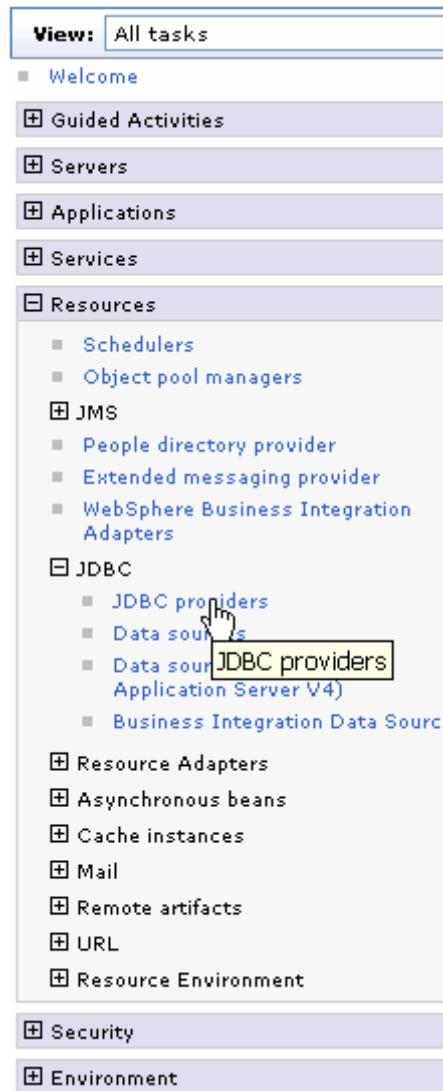
New Delete

MQ JDBC DRIVER PATH

Select	Name	Value	Scope
You can administer the following resources:			
<input type="checkbox"/>	MQ_INSTALL_ROOT	`\${WAS_INSTALL_ROOT}/lib/WMQ	Node=n1Node01
<input type="checkbox"/>	ORACLE_JDBC_DRIVER_PATH	D:\Lib	Node=n1Node01
<input type="checkbox"/>	OS400_NATIVE_JDBC40_DRIVER_PATH		Node=n1Node01
<input type="checkbox"/>	OS400_NATIVE_JDBC_DRIVER_PATH		Node=n1Node01
<input type="checkbox"/>	OS400_TOOLBOX_JDBC_DRIVER_PATH		Node=n1Node01
<input type="checkbox"/>	ORACLE_JDBC_DRIVER_PATH		Node=n1Node01

WebSphere software

4. Select **Resources** → **JDBC** -> **JDBC Providers**.



5. Click **New** in the JDBC providers window.

JDBC providers

JDBC providers

Use this page to edit properties of a JDBC provider. The JDBC provider object encapsulates the specific JDBC driver implementation class for access to the specific vendor database of your environment. Learn more about this task in a [guided activity](#). A guided activity provides a list of task steps and more general information about the topic.

Scope: Cell=**localhostNode01Cell**, Node=**n1Node01**

Scope specifies the level at which the resource definition is visible. For detailed information on what scope is and how it works, [see the scope settings help](#).

Node=n1Node01

Preferences

Select	Name	Scope	Description
None			
Total 0			

6. Select an Oracle database with a connection pool data source for the Oracle JDBC driver. Click **Next**.

Create a new JDBC Provider

Create a new JDBC Provider

→ Step 1: Create new JDBC provider

Step 2: Enter database class path information

Step 3: Summary

Create new JDBC provider

Set the basic configuration values of a JDBC provider, which encapsulates the specific vendor JDBC driver implementation classes that are required to access the database. The wizard fills in the name and the description fields, but you can type different values.

Scope

cells:localhostNode01Cell:nodes:n1Node01

* Database type

Oracle

* Provider type

Oracle JDBC Driver

* Implementation type

Connection pool data source

* Name

Oracle JDBC Driver

Description

Oracle JDBC Driver

Next **Cancel**

7. In the Enter database classpath information page, enter the following value for the **Class path** field:

`$(ORACLE_JDBC_DRIVER_PATH)/ojdbc6.jar`, where
`$(ORACLE_JDBC_DRIVER_PATH)` is library path for the run time.

8. Click **Next**.

WebSphere software

Cell=localhostNode01Cell, Profile=AppSrv01

Create a new JDBC Provider

Step 1: Create new JDBC provider
→ Step 2: Enter database class path information
Step 3: Summary

Enter database class path information

Set the environment variables that represent the JDBC driver class files, which WebSphere(R) Application Server uses to define your JDBC provider. This wizard page displays the file names; you supply only the directory locations of the files. Use complete directory paths when you type the JDBC driver file locations. For example: C:\SQLLIB\java on Windows(R) or /home/db2inst1/sqllib/java on Linux(TM).

If a value is specified for you, you may click Next to accept the value.

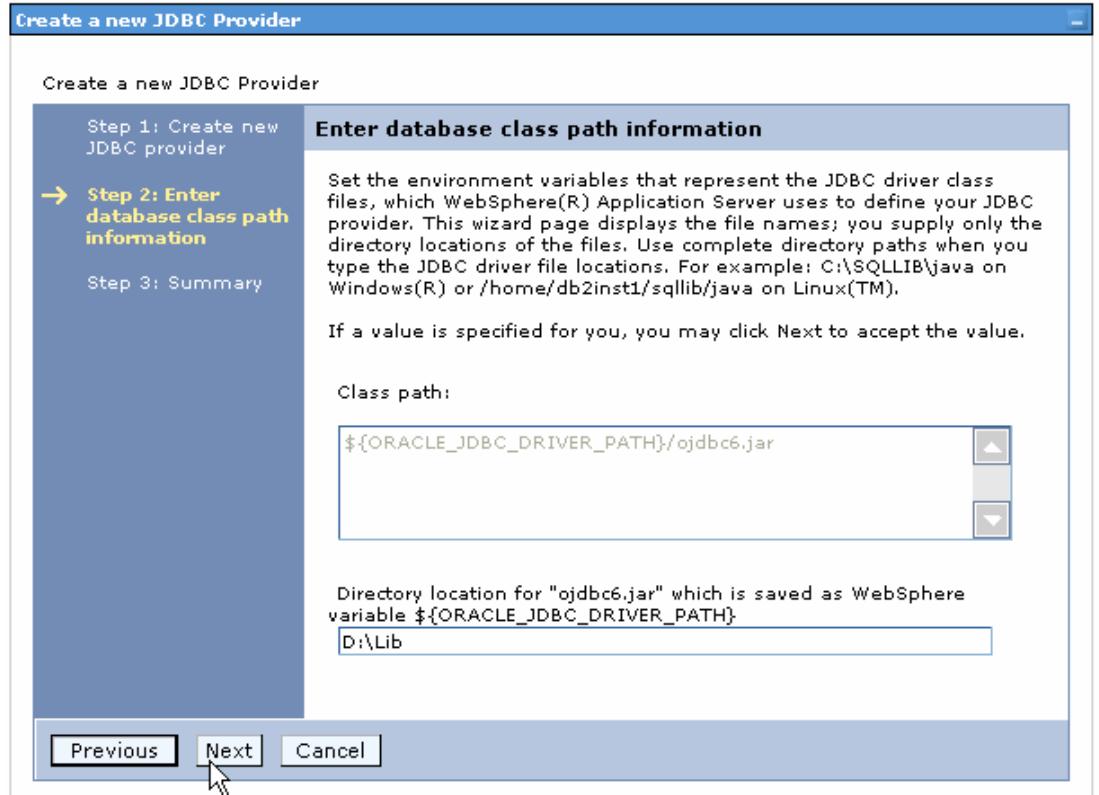
Class path:

`${ORACLE_JDBC_DRIVER_PATH}/ojdbc6.jar`

Directory location for "ojdbc6.jar" which is saved as WebSphere variable `${ORACLE_JDBC_DRIVER_PATH}`

D:\Lib

Previous **Next** Cancel



9. In the Summary page, click **Finish**.

Cell=localhostNode01Cell, Profile=AppSrv01 [Close page](#)

Create a new JDBC Provider

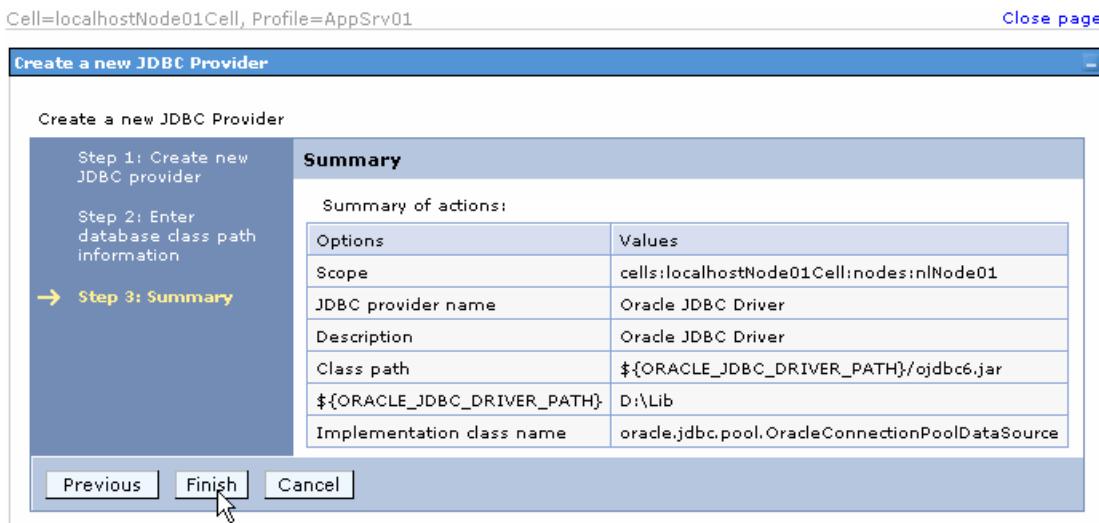
Step 1: Create new JDBC provider
Step 2: Enter database class path information
→ Step 3: Summary

Summary

Summary of actions:

Options	Values
Scope	cells:localhostNode01Cell:nodes:n1Node01
JDBC provider name	Oracle JDBC Driver
Description	Oracle JDBC Driver
Class path	<code> \${ORACLE_JDBC_DRIVER_PATH}/ojdbc6.jar</code>
<code> \${ORACLE_JDBC_DRIVER_PATH}</code>	D:\Lib
Implementation class name	oracle.jdbc.pool.OracleConnectionPoolDataSource

Previous **Finish** Cancel



10. Click **Save**.

WebSphere software

Cell=localhostNode01Cell, Profile=AppSrv01

JDBC providers

Messages

Changes have been made to your local configuration.
You can:

- Save directly to the master configuration.
- Review changes before saving or discarding.

The server may need to be restarted for these changes to take effect.

The JDBC provider is added and appears in the list.

Cell=localhostNode01Cell, Profile=AppSrv01

JDBC providers

Use this page to edit properties of a JDBC provider. The JDBC provider object encapsulates the specific JDBC driver implementation class for access to the specific vendor database of your environment. Learn more about this task in a [guided activity](#). A guided activity provides a list of task steps and more general information about the topic.

Scope: Cell=**localhostNode01Cell**, Node=**nINode01**

Scope specifies the level at which the resource definition is visible. For detailed information on what scope is and how it works, [see the scope settings help](#).

Node=nINode01

Preferences

New	Delete		
<input type="checkbox"/>	<input type="checkbox"/>		
<input type="checkbox"/>	<input type="checkbox"/>		
Select	Name ▲	Scope ▲	Description ▲
You can administer the following resources:			
<input type="checkbox"/>	Oracle JDBC Driver	Node=nINode01	Oracle JDBC Driver
Total 1			

11. Select the Oracle JDBC provider you just created. Under **Additional Properties**, click **Data sources**. Click **New**.

WebSphere software

Cell=localhostNode01Cell, Profile=AppSrv01

The screenshot shows the 'JDBC providers' page with the 'Oracle JDBC Driver' selected. The 'Data sources' table lists one entry: 'None'. The 'New' button is highlighted with a cursor.

12. Type any value in the **JNDI name** field, and select the authentication alias. Click **Next**.

Cell=localhostNode01Cell, Profile=AppSrv01

The screenshot shows the 'Create a data source' wizard at Step 1. The 'Scope' field is set to 'cells:localhostNode01Cell:nodes:n1Node01'. The 'JDBC provider name' is 'Oracle JDBC Driver'. The 'Data source name' is 'Oracle JDBC Driver DataSource' and the 'JNDI name' is 'OracleDS'. The 'Next' button is highlighted with a cursor.

13. Provide the appropriate URL value and select a data store helper class name from the **Data store helper class name** list as shown in the following figure. Click **Next**.

WebSphere software

Cell=localhostNode01Cell, Profile=AppSrv01

Create a data source

Step 1: Enter basic data source information

→ Step 2: Enter database specific properties for the data source

Step 3: Setup security aliases

Step 4: Summary

Enter database specific properties for the data source

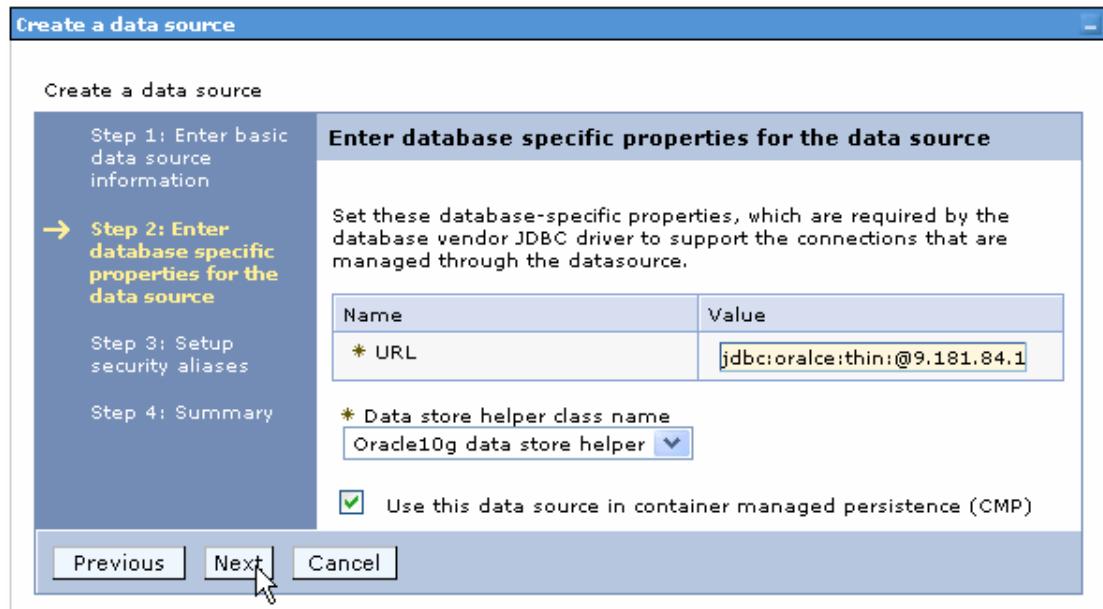
Set these database-specific properties, which are required by the database vendor JDBC driver to support the connections that are managed through the datasource.

Name	Value
* URL	jdbc:oracle:thin:@9.181.84.1

* Data store helper class name
Oracle10g data store helper

Use this data source in container managed persistence (CMP)

Previous Next Cancel



14. Select the authentication alias you just created from the **Component-managed authentication alias** field and click **Next**.

Cell=localhostNode01Cell, Profile=AppSrv01 Close

Create a data source

Step 1: Enter basic data source information

Step 2: Enter database specific properties for the data source

→ Step 3: Setup security aliases

Step 4: Summary

Setup security aliases

Select the authentication values for this resource.

Component-managed authentication alias
nlNode01/Alias_Oracle

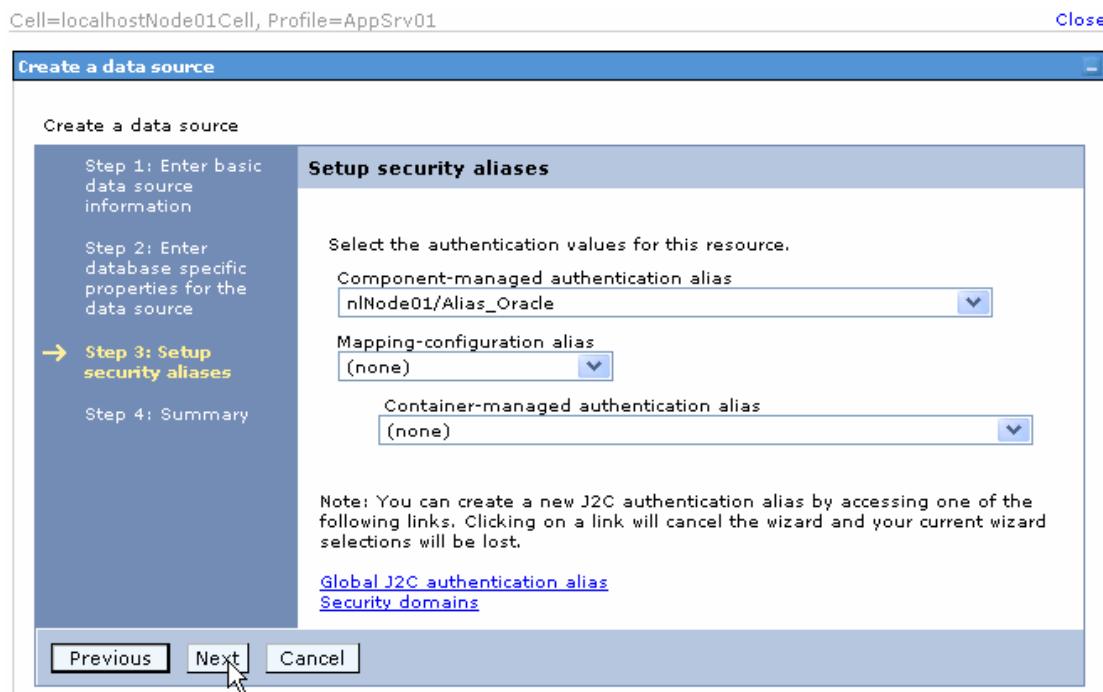
Mapping-configuration alias
(none)

Container-managed authentication alias
(none)

Note: You can create a new J2C authentication alias by accessing one of the following links. Clicking on a link will cancel the wizard and your current wizard selections will be lost.

[Global J2C authentication alias](#)
[Security domains](#)

Previous Next Cancel



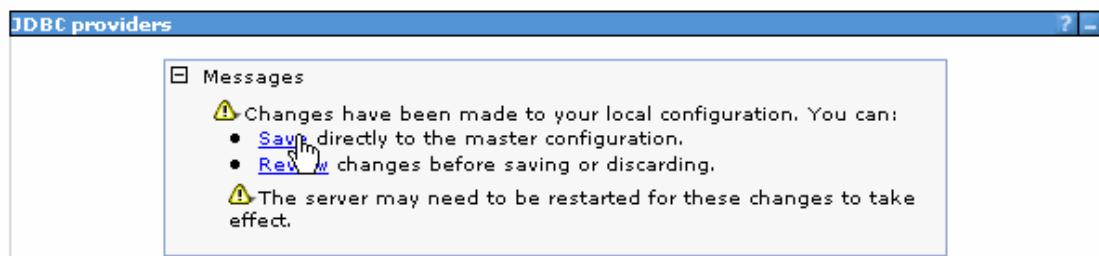
15. In the Summary page, review the values entered for the data source and click **Finish**.

Create a data source

Summary	
Summary of actions:	
Options	Values
Scope	cells:localhostNode01Cell:nodes:n1Node01
Data source name	Oracle JDBC Driver DataSource
JNDI name	OracleDS
Select an existing JDBC provider	Oracle JDBC Driver
Implementation class name	oracle.jdbc.pool.OracleConnectionPoolDataSource
URL	jdbc:oracle:thin:@9.181.84.136:1521:ord
Data store helper class name	com.ibm.websphere.radapter.Oracle10gDataStoreHelper
Use this data source in container managed persistence (CMP)	true
Component-managed authentication alias	n1Node01/Alias_Oracle
Mapping-configuration alias	(none)
Container-managed authentication alias	(none)

16. Click **Save** to save the changes.

Cell=localhostNode01Cell, Profile=AppSrv01



17. Select the data source you just created and click **Test connection**.

Cell=localhostNode01Cell, Profile=AppSrv01

JDBC providers

[JDBC providers](#) > [Oracle JDBC Driver](#) > Data sources

Use this page to edit the settings of a datasource that is associated with your selected JDBC provider. The datasource object supplies your application with connections for accessing the database. Learn more about this task in a [guided activity](#). A guided activity provides a list of task steps and more general information about the topic.

Preferences

New	Delete	Test connection	Manage state...
Select	Name ▾	JNDI name ▾	Scope ▾
Provider ▾	Description ▾	Category ▾	

You can administer the following resources:

<input checked="" type="checkbox"/>	Oracle JDBC Driver DataSource	OracleDS	Node=n1Node01	Oracle JDBC Driver	New JDBC Datasource	
-------------------------------------	---	----------	---------------	--------------------	---------------------	--

Total 1

The connection should succeed as indicated by the message shown in the following figure. If you experience problems with the test connection, refer to the “Troubleshooting” section.

Cell=localhostNode01Cell, Profile=AppSrv01

JDBC providers

[JDBC providers](#) > [Oracle JDBC Driver](#) > Data sources

Messages

The test connection operation for data source Oracle JDBC Driver DataSource on server server1 at node n1Node01 was successful.

[Information](#)

Use this page to edit the settings of a datasource that is associated with your selected JDBC provider. The datasource object supplies your application with connections for accessing the database. Learn more about this task in a [guided activity](#). A guided activity provides a list of task steps and more general information about the topic.

Preferences

New	Delete	Test connection	Manage state...
Select	Name ▾	JNDI name ▾	Scope ▾
Provider ▾	Description ▾	Category ▾	

You can administer the following resources:

<input type="checkbox"/>	Oracle JDBC Driver DataSource	OracleDS	Node=n1Node01	Oracle JDBC Driver	New JDBC Datasource	
--------------------------	---	----------	---------------	--------------------	---------------------	--

Total 1

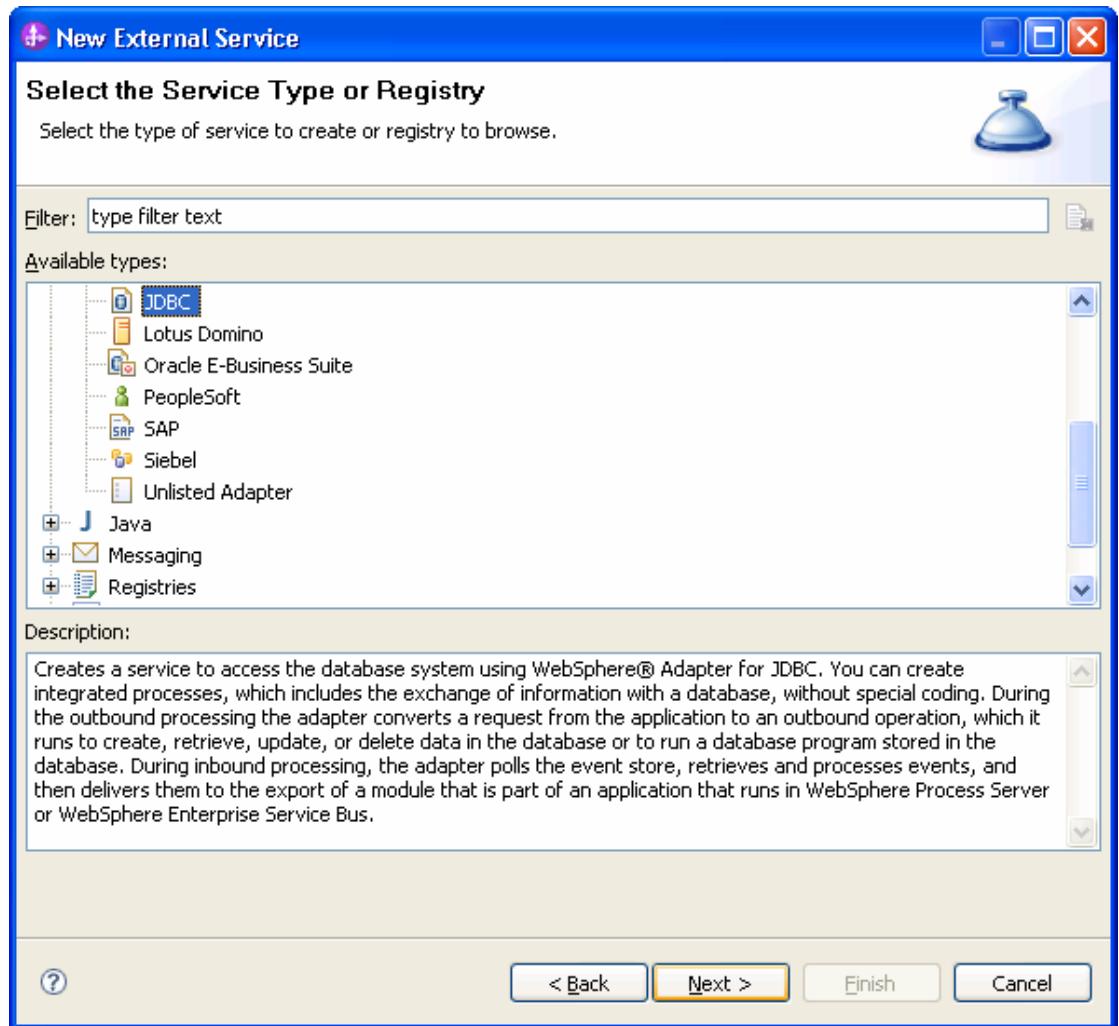
The data source is created and it will be used by the adapter to connect to the database.

Configure the adapter for outbound processing

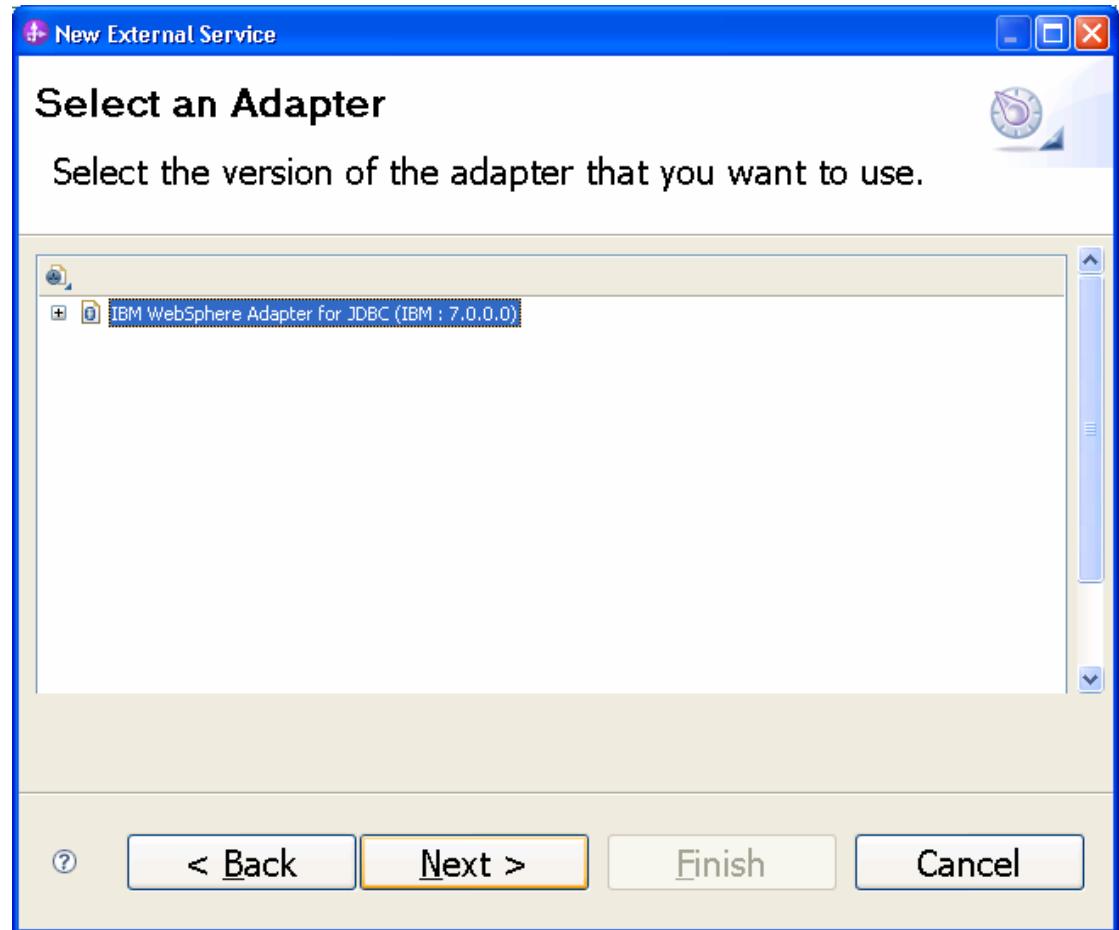
WebSphere software

Run the external service wizard to specify business objects, services, and configuration details.

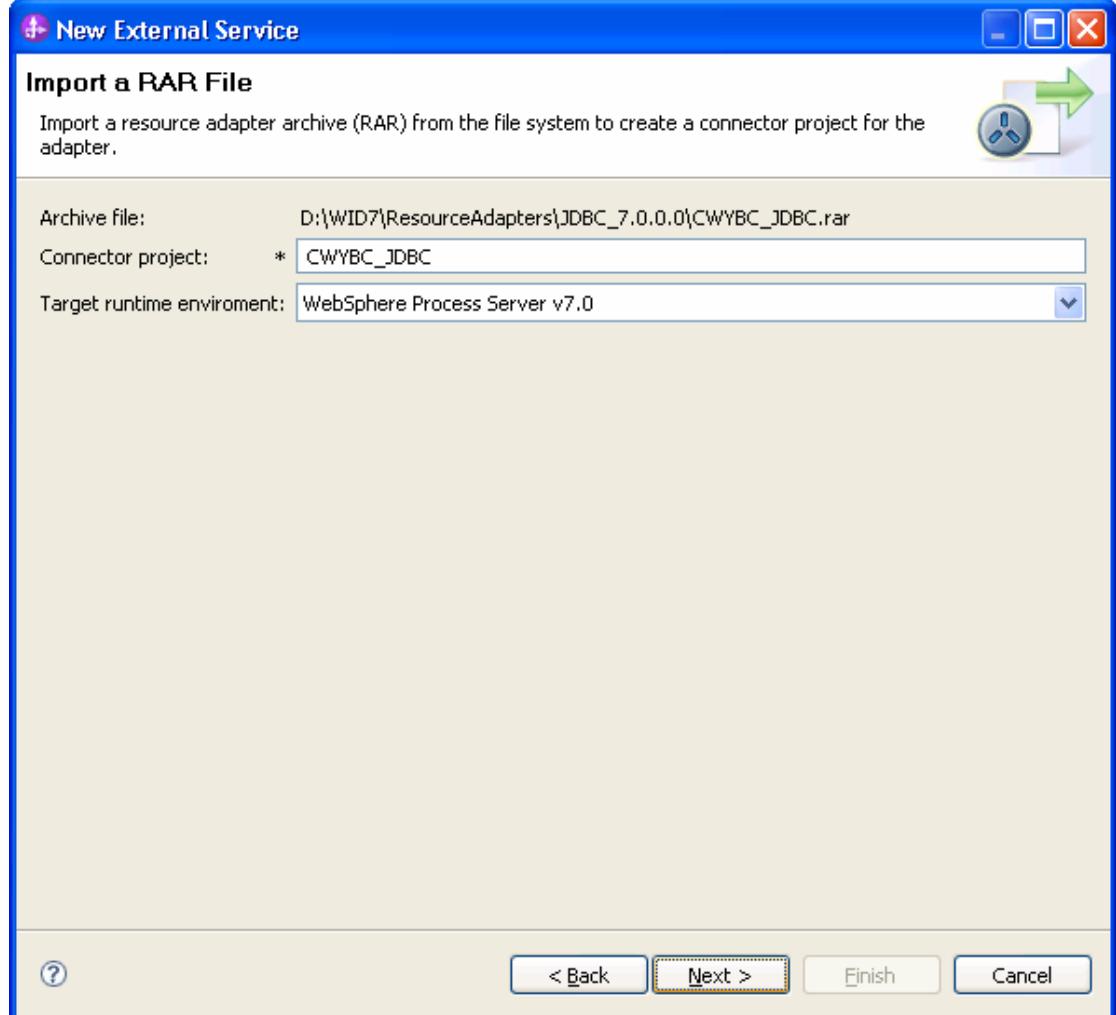
1. Switch to the Business Integration Perspective in WebSphere Integration Developer by selecting **Window -> Open Perspective Business Integration**.
2. Start the external service wizard by selecting **File-> New -> External Service**.
3. In the **Available Types** area, select **Adapters > JDBC** and click **Next**.



4. Select **IBM WebSphere Adapter for JDBC (IBM: 7.0.0.0)** and click **Next**.

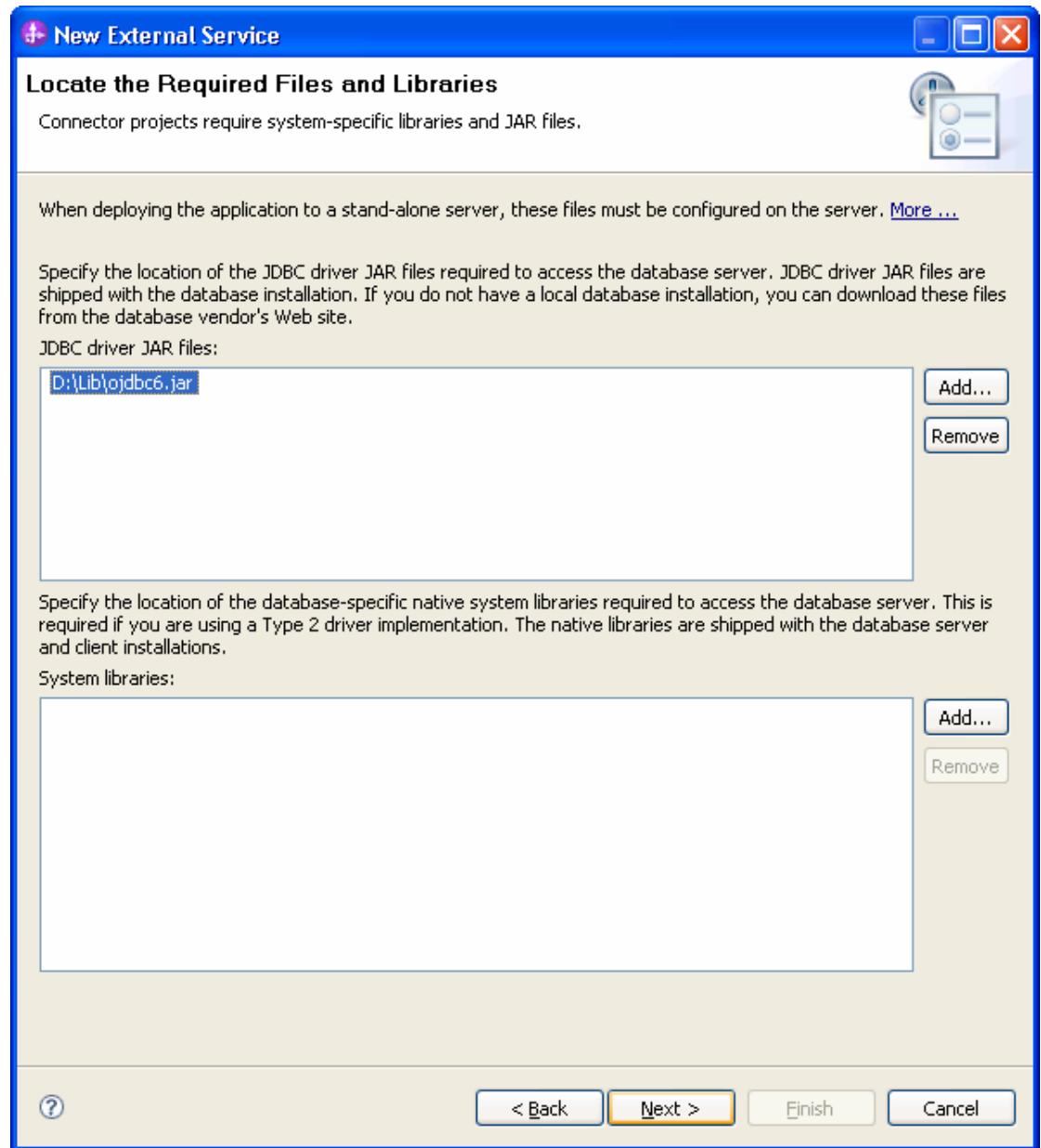


5. In the **Connector project** field enter **CWYBC_JDBC**.
6. In the **Target runtime environment** field, select the appropriate runtime and click **Next**.

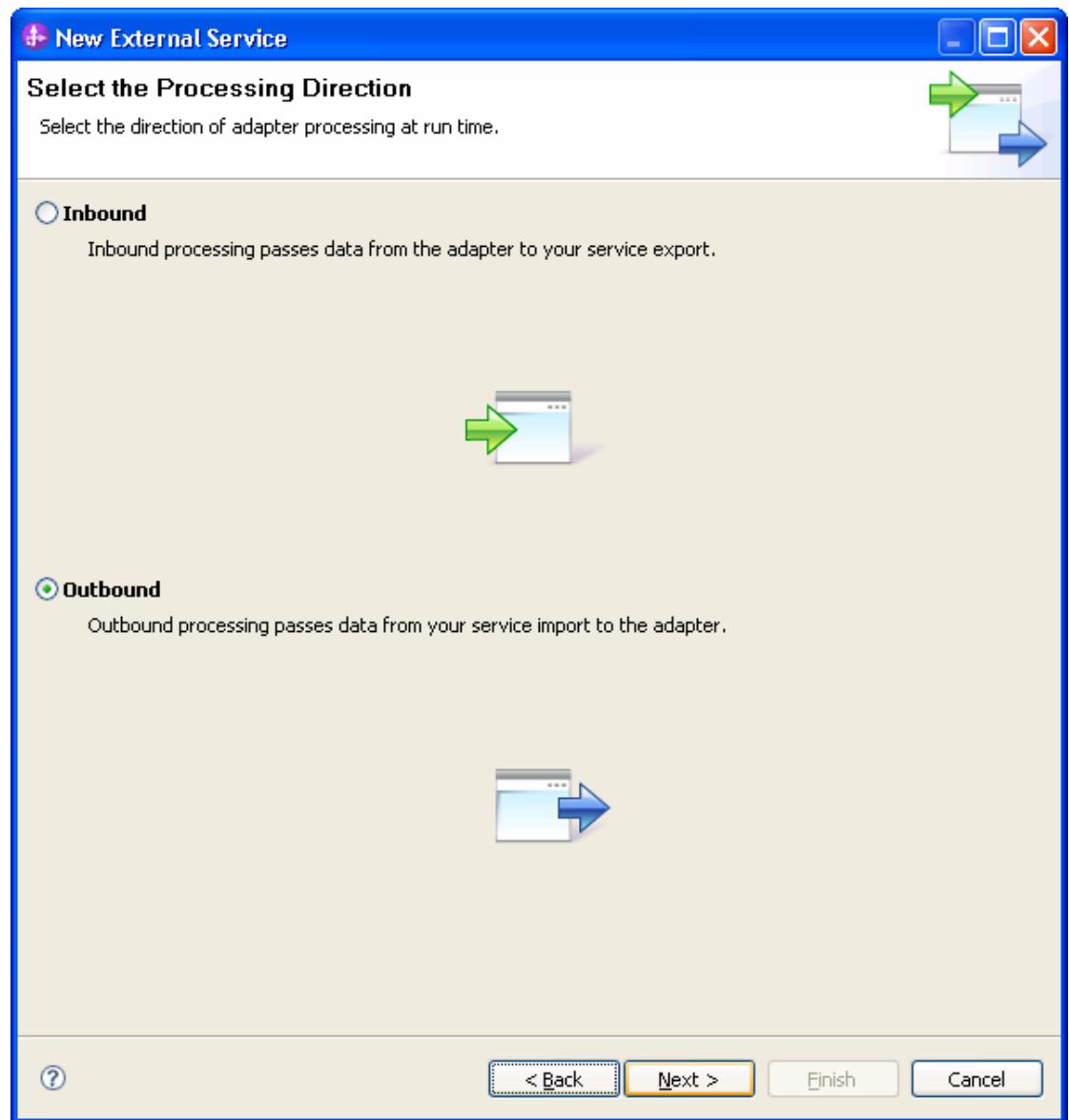


WebSphere software

7. In the **JDBC driver JAR files** field, click **Add** to add the JDBC driver class to connect to the database. Browse to select the driver JAR file and click **Next**.



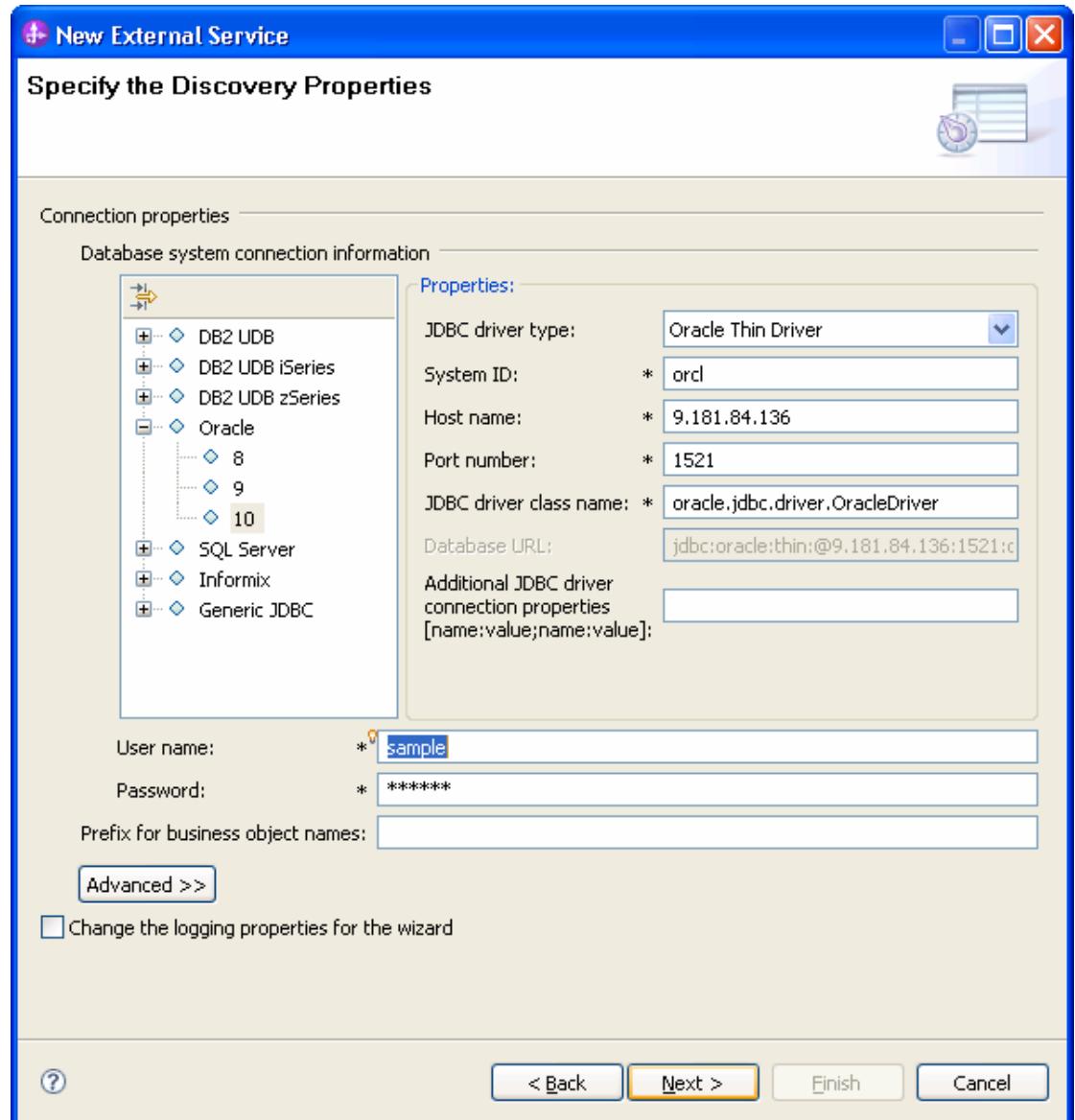
8. Select **Outbound** and click **Next**.



Set connection properties for the external service wizard

To connect to the Oracle database:

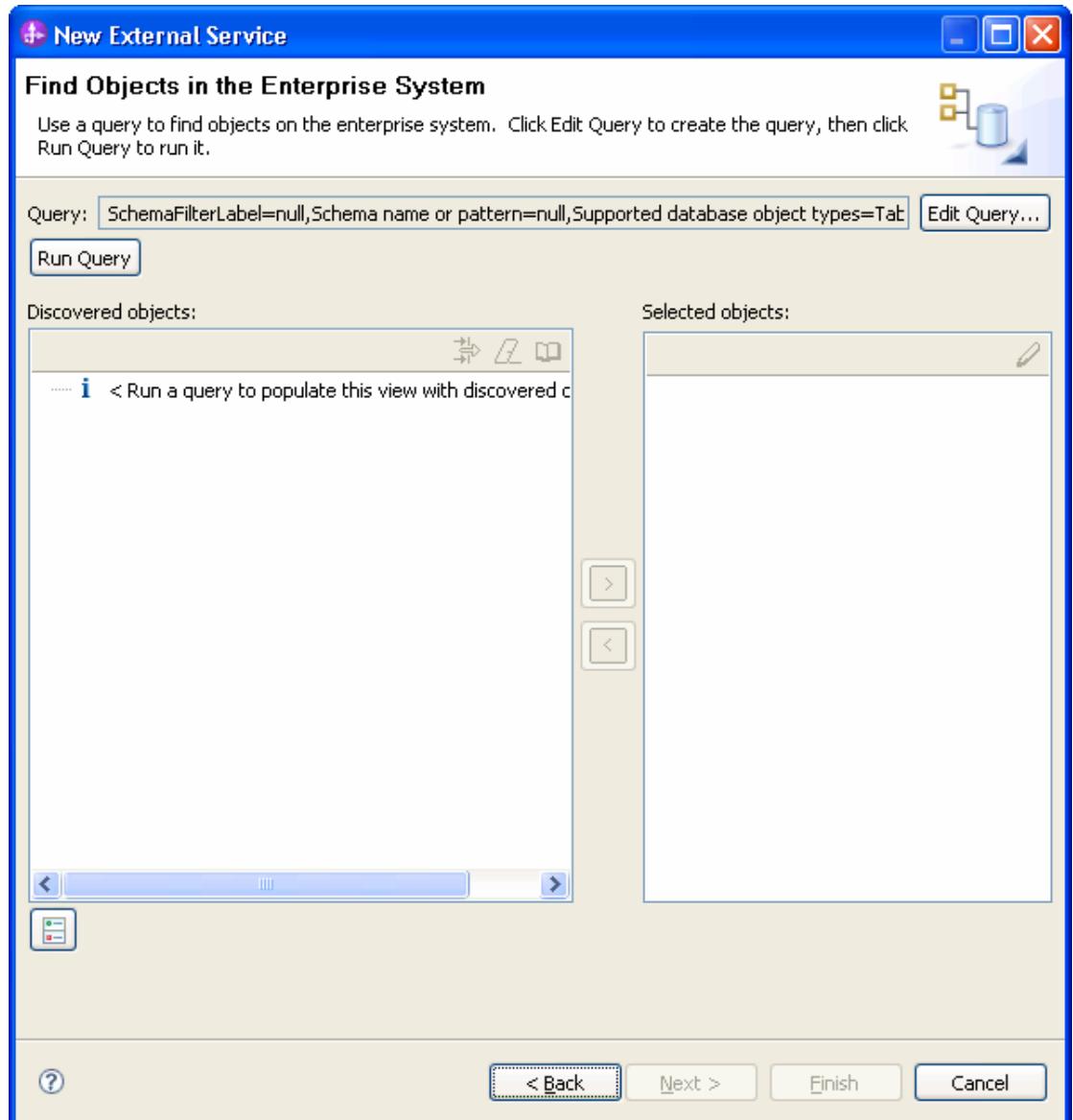
1. Expand the **Oracle** node in the **Database system connection information** area and select **10**.
2. Enter values in the **System ID**, **Host name**, **Port number**, **User name** and **Password** fields, and then click **Next**.



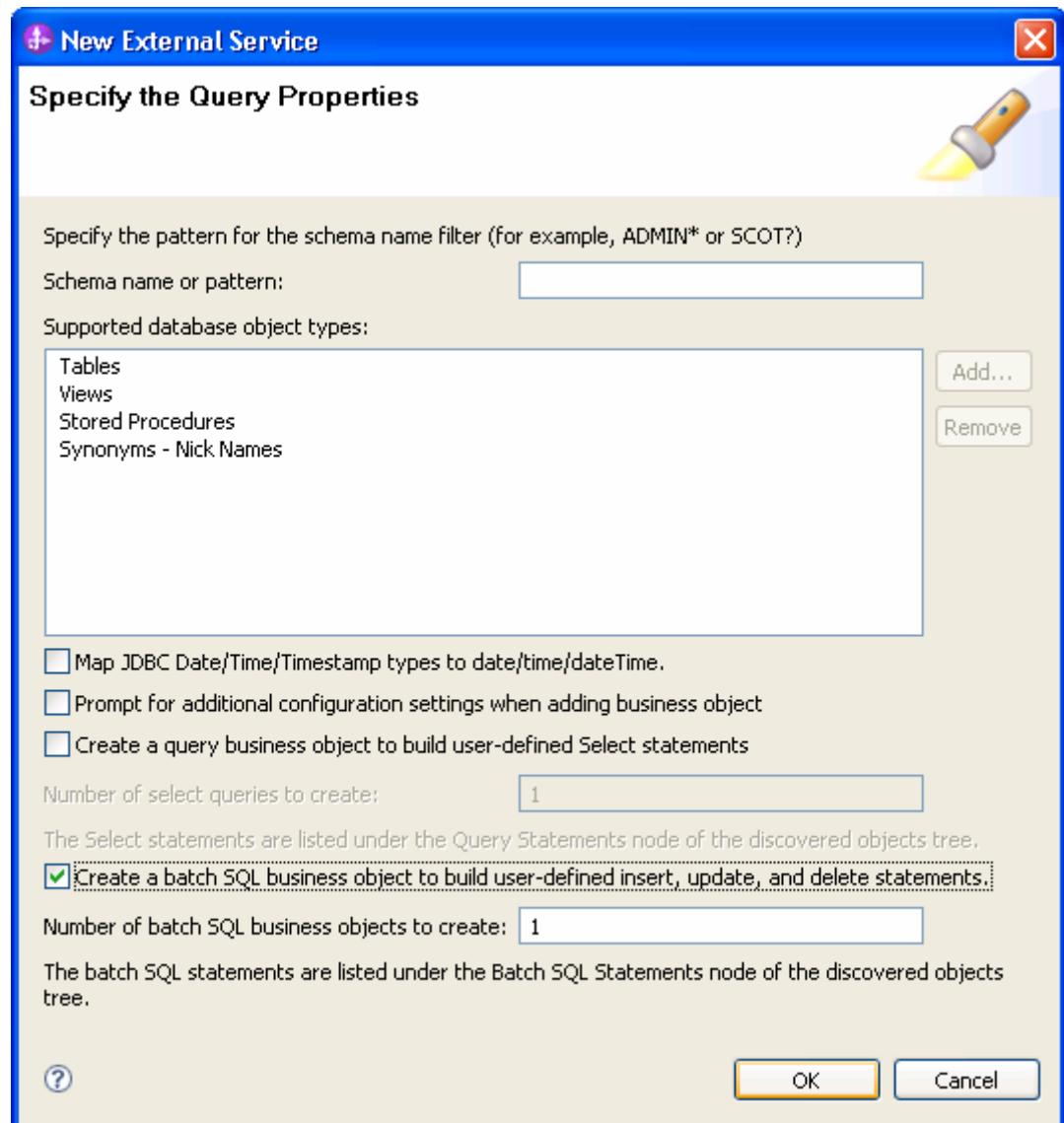
Select the business objects and services to be used with the adapter

Follow these steps to select the data for outbound Processing:

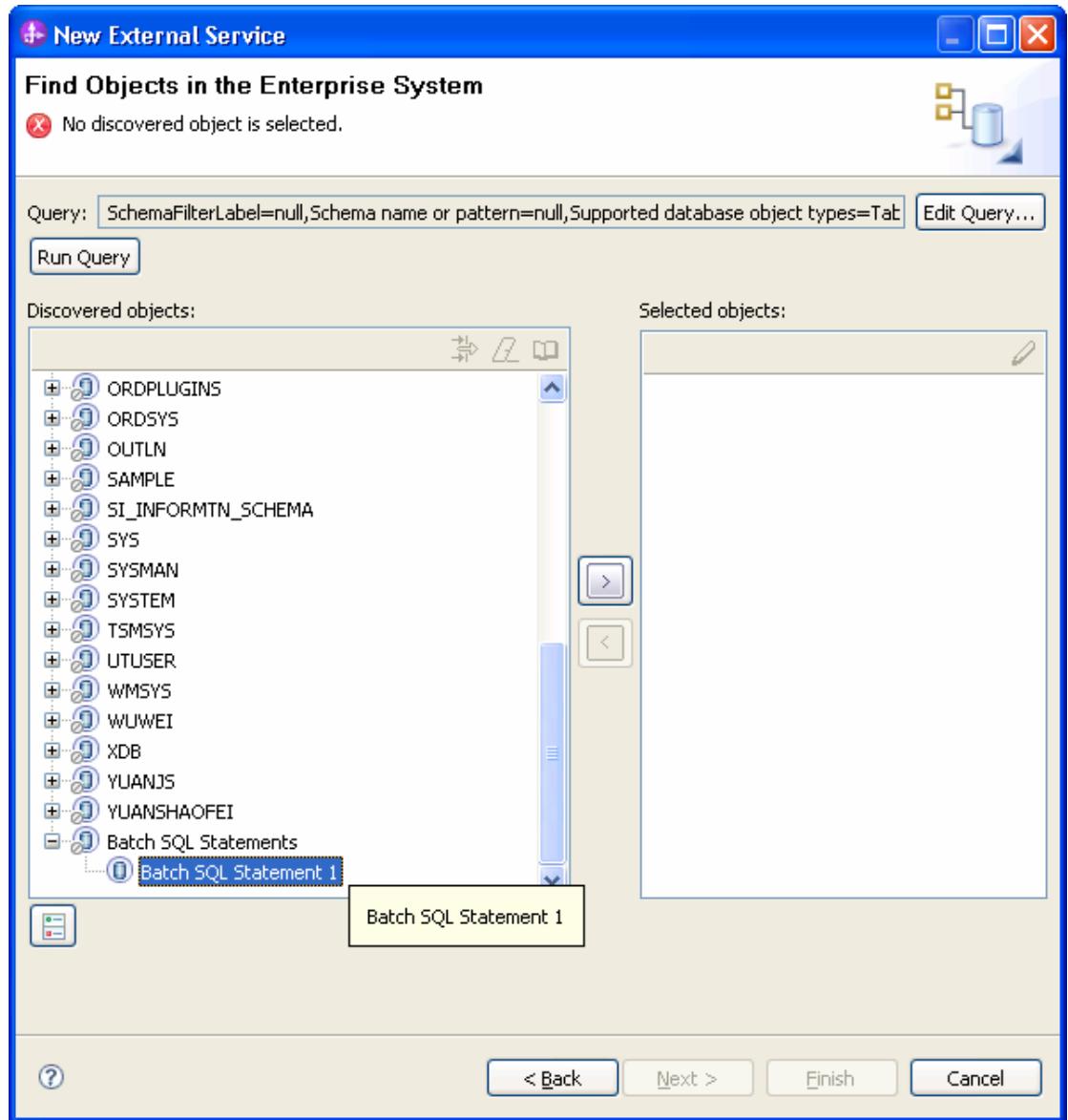
1. In the Object Discovery and Selection screen, click **Edit Query**.



2. In the Specify the Query Properties window, select the **Create batch SQL business object...** check box and accept the default value for the **Number of batch SQL business objects to be created** field. Click **OK**.

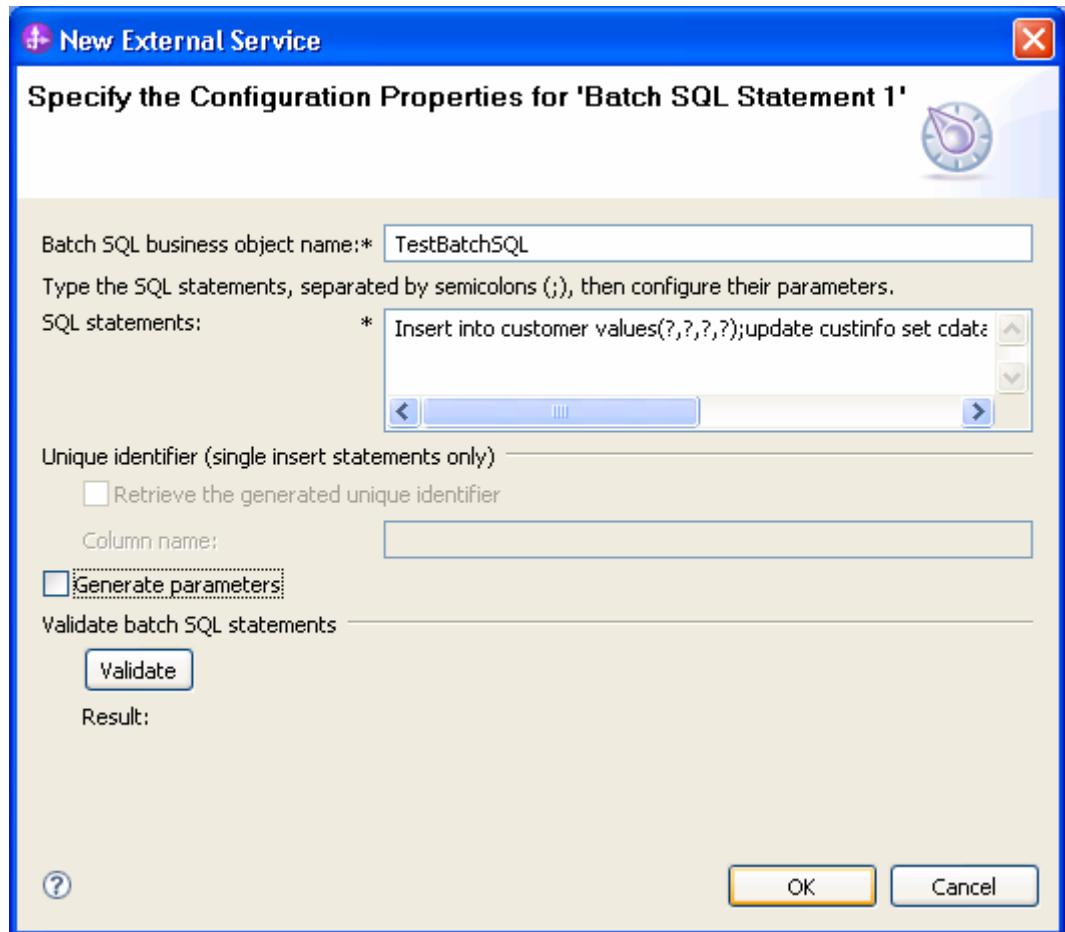


3. In the Find Objects in Enterprise System window, click **Run Query**.
4. Expand the **Batch SQL Statements** node.

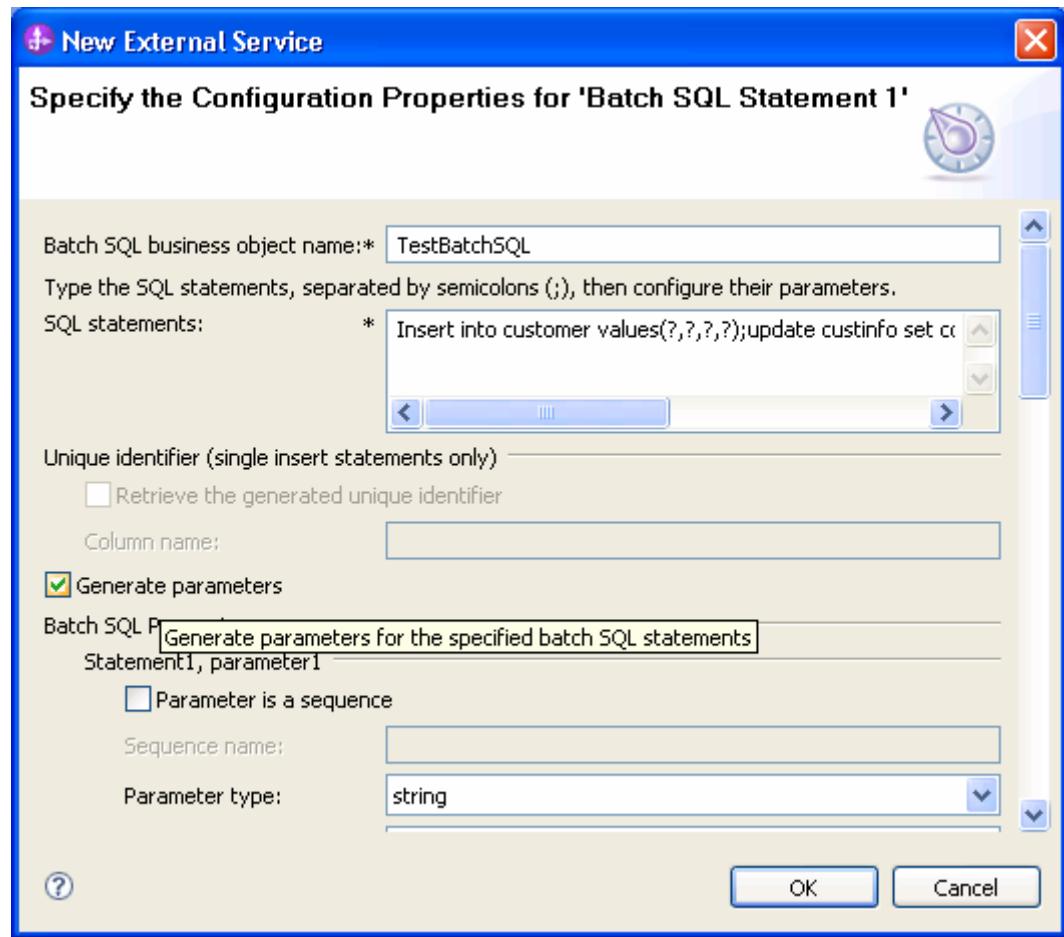


5. Select **Batch SQL Statement1** and click .
6. In the Specify the Configuration Properties for Batch SQL window, specify the following details:
 - a) In the **Batch SQL business object name** field, enter **TestBatchSQL**.
 - b) In the **SQL Statements** field, enter the following query:

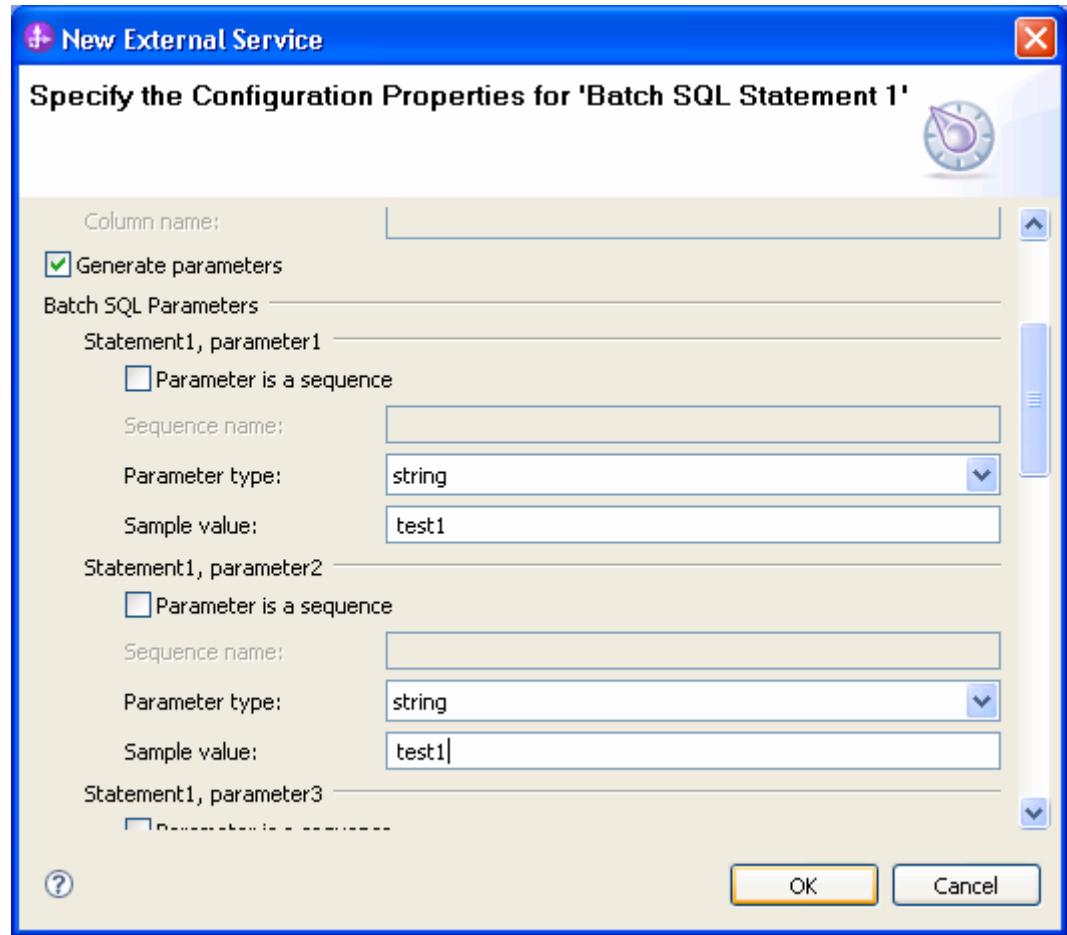
```
Insert into customer values(?, ?, ?, ?); update  
custinfo set cdata=? Where ccode=?; delete from  
customer where pkey=?
```



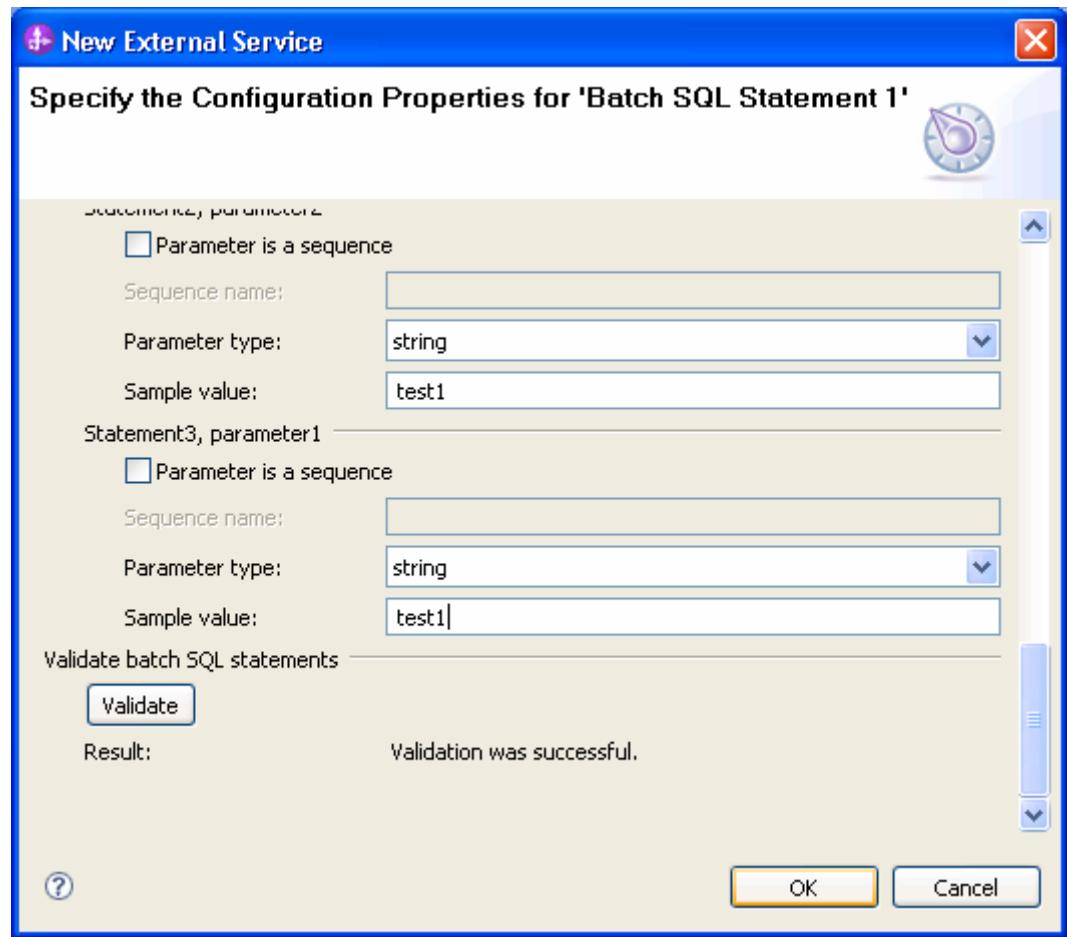
7. Select the **Generate Parameters** check box. Parameter fields corresponding to each '?' in the SQL Statements will be generated as shown in the figure below:



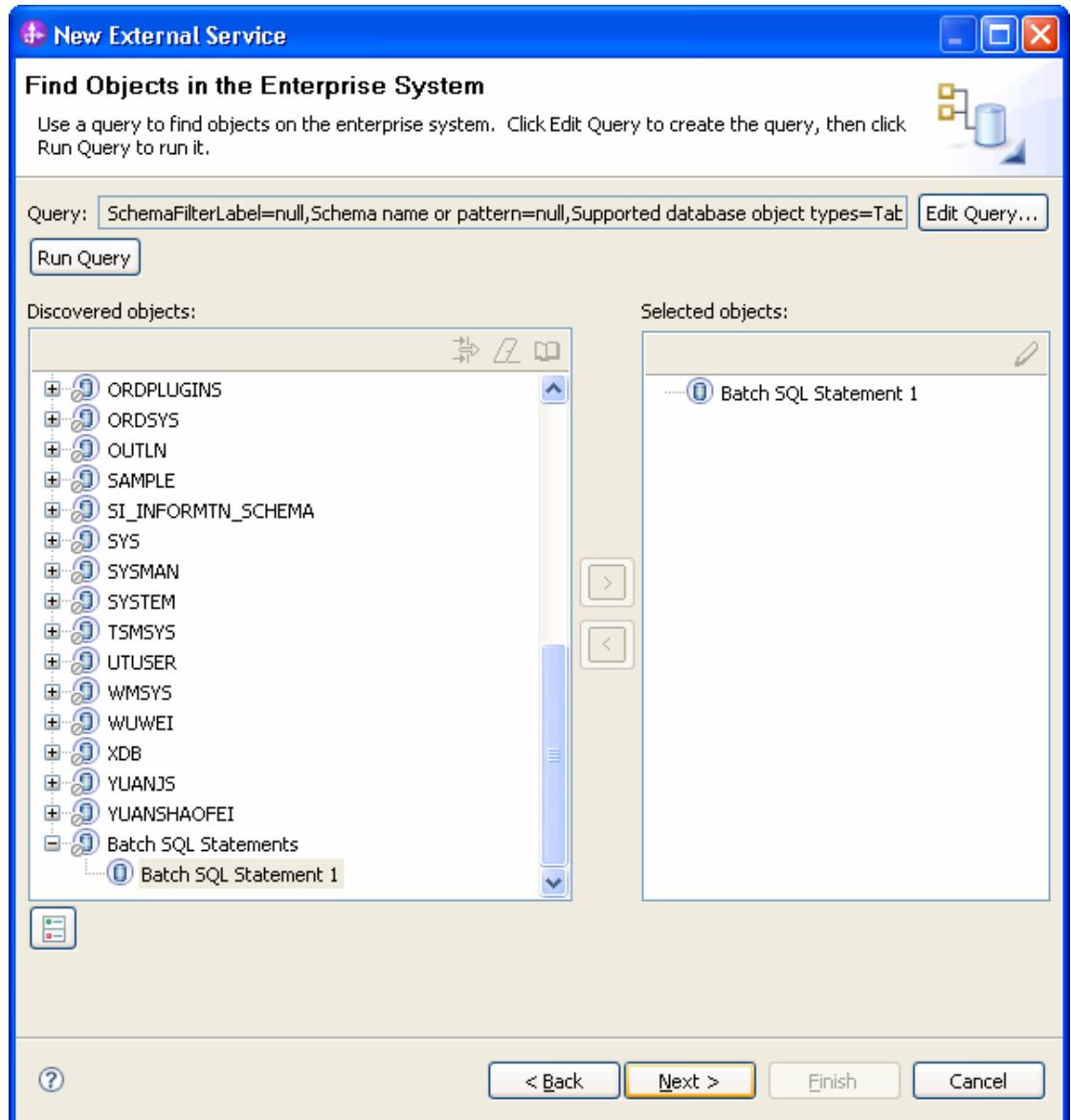
8. Select the parameter type and enter the sample value for each parameter in all the statements.



9. Click **Validate**. The validation result is displayed. Click **OK**.



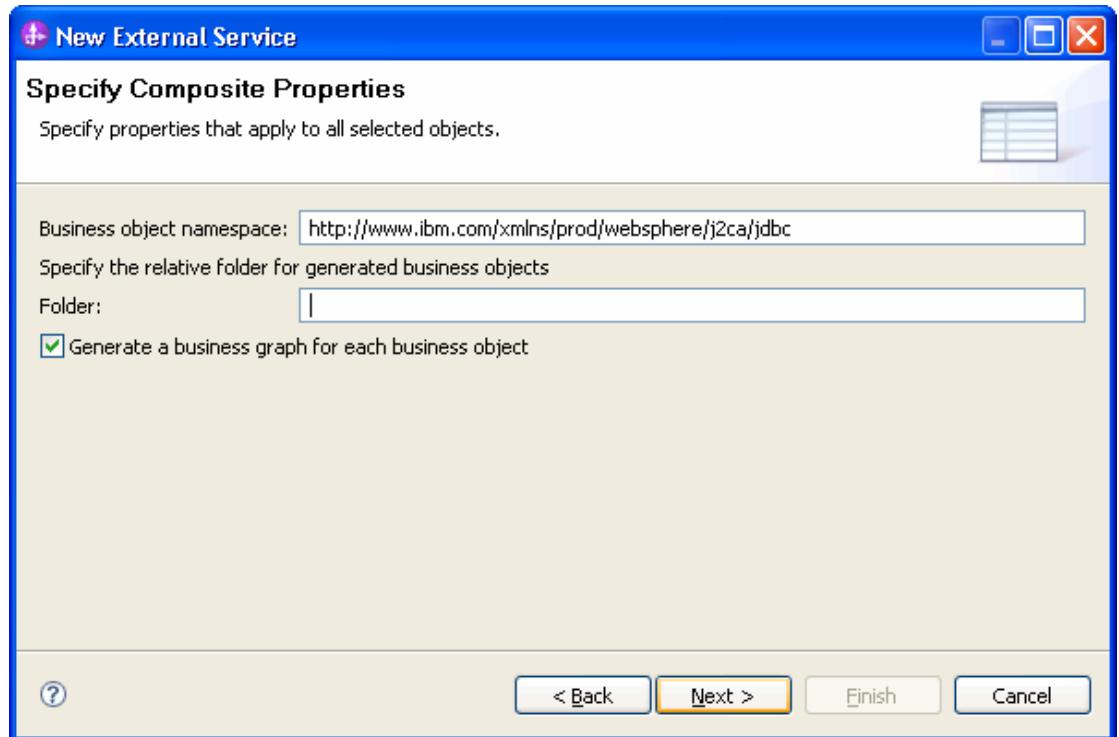
10. The Batch SQL Statement1 will be listed in the **Selected Objects**.
Click **Next**.



Generate business object definitions and related artifacts

Follow these steps to generate the business object definitions.

1. In the Specify Composite Properties window, accept the default values for the all fields and click **Next**.



2. In the Specify the Service Generation and Deployment Properties window, perform the following steps:
 - a) Select **Other** for security options under **Deployment Properties**.
 - b) Clear the **Join the global transaction** check box.
 - c) Select **Specify predefined connection pool DataSource** from the **Database connection information** list.
 - d) Enter **OracleDS** in the **Connection pool DataSource JNDI Name** field, and click **Next**.

New External Service

Specify the Service Generation and Deployment Properties

Specify properties for generating the service and running it on the server.

Service Operations

To modify the names, or add a description to the operations to be generated in the interface file, click Edit Operations.

[Edit Operations...](#)

Deployment Properties

How do you want to specify the security credentials?

Using an existing JAAS alias (recommended)
A Java Authentication and Authorization Services (JAAS) alias is the preferred method.
J2C authentication data entry:

Using security properties from the managed connection factory
The properties will be stored as plain text; no encryption is used.
User name:
Password:

Other
Use if no security is required or will be handled by the EIS system, or the RAR will be deployed on the server and security will be specified by the properties in the JNDI lookup name.

The quality of service that is used to join the transaction provides a higher degree of data integrity, especially when a failure occurs. To participate in a global transaction, a predefined XA DataSource or XA database connection information must be specified in the connection properties. [More ...](#)

Join the global transaction

Deploy connector project:

Specify the settings used to connect to JDBC at run time:

Connection settings:

Connection Properties

To join a global transaction, specify a predefined XA datasource or XA database connection information. When not joining a global transaction, either the XA connection information or the local connection information can be specified.

Database connection information:

Database system connection information

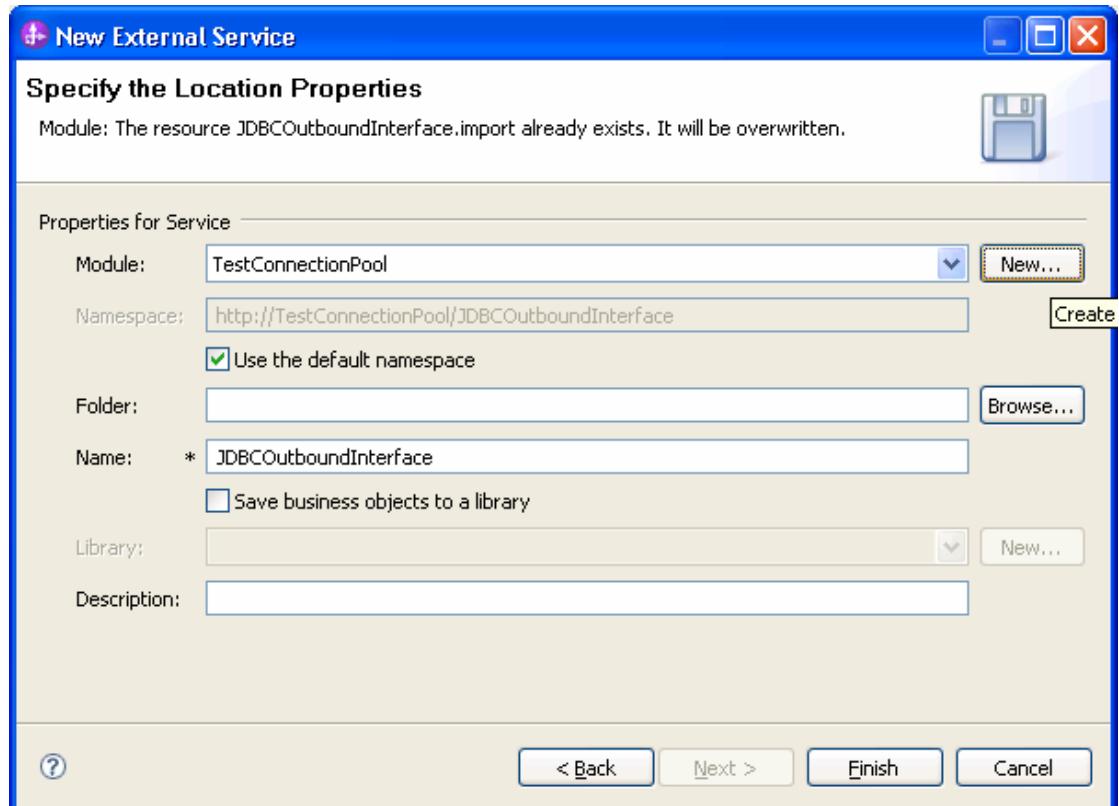
Database vendor: ORACLE

Connection pool DataSource JNDI name:*

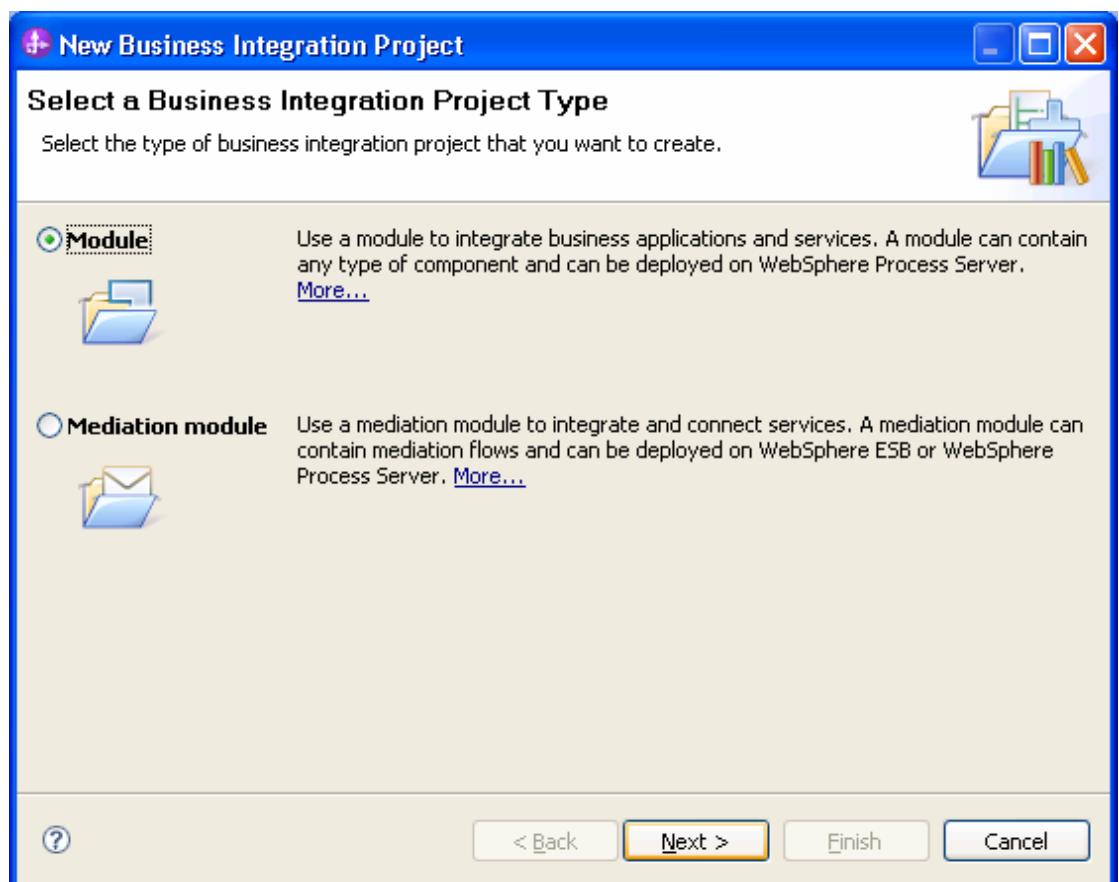
[Advanced >>](#)

[?](#) < Back **Next >** Finish Cancel

3. In the Specify the Location Properties window, click **New**.

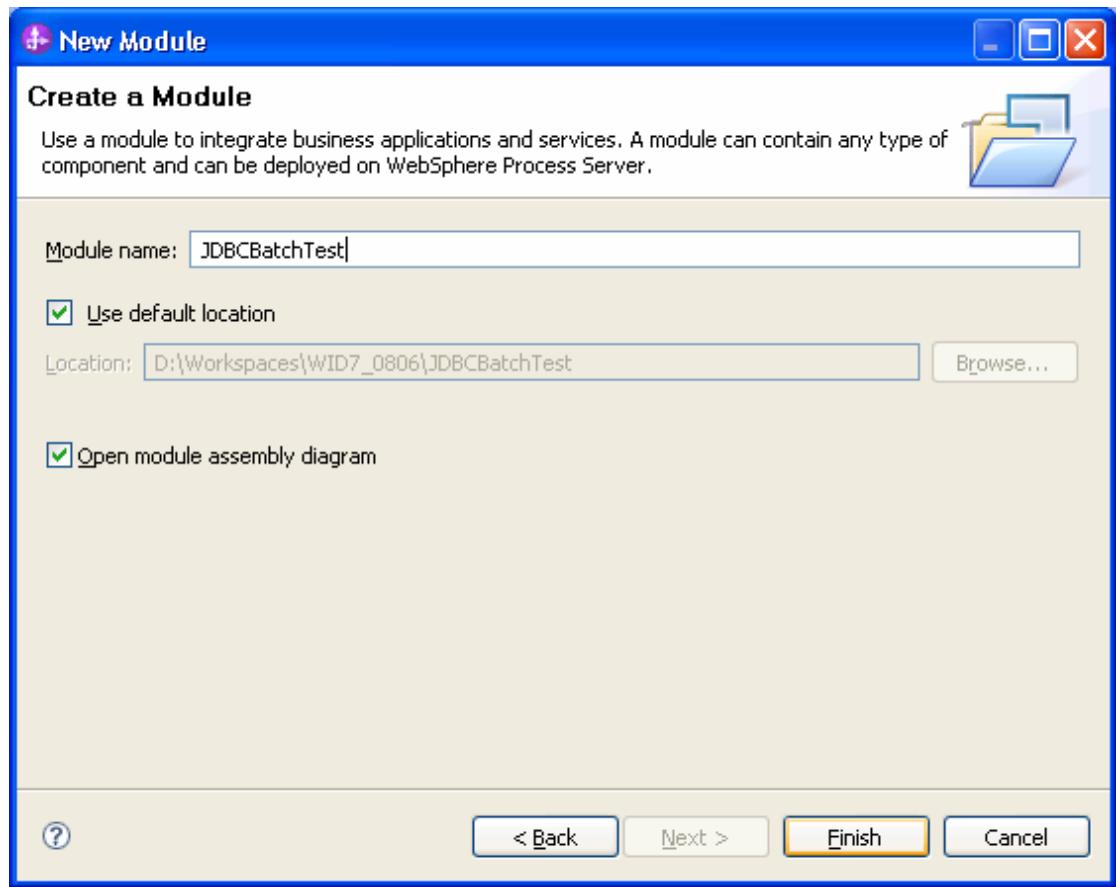


4. In the Select a Business Integration Project Type window, select **Module** and click **Next**.

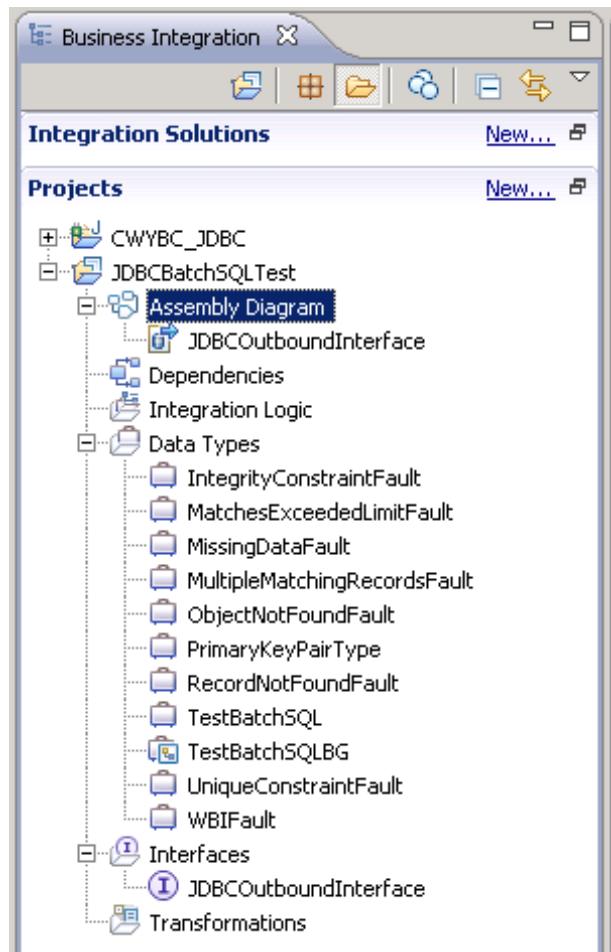


WebSphere software

5. In the Create a Module window, type **JDBCBatchSQLTest** in the **Module Name** field and click **Finish**.



6. Click **Finish** to complete service creation.
7. Verify the results.



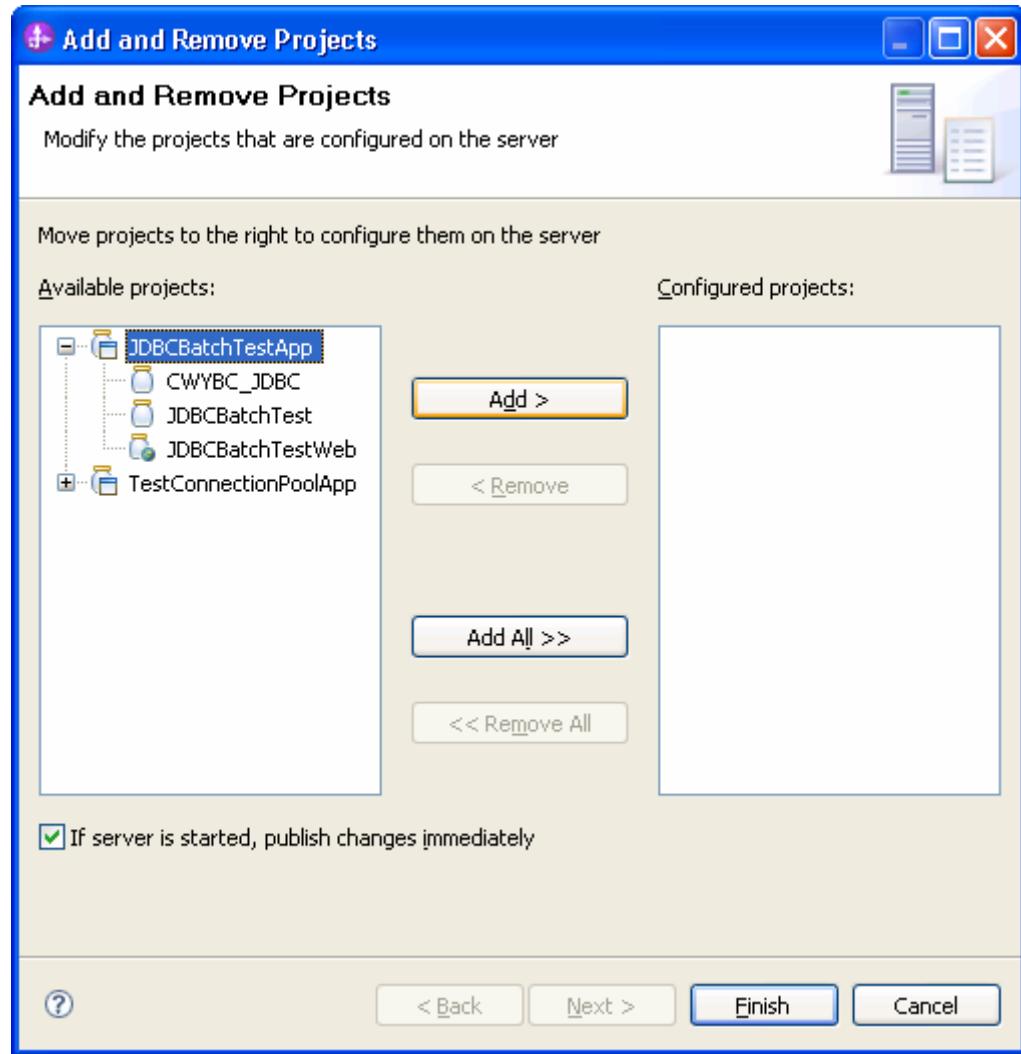
Deploy the module to the test environment

After running the external service wizard, you will have an SCA module that contains an Enterprise Information System (EIS) import. You must install this SCA module in the WebSphere Integration Developer integration test client.

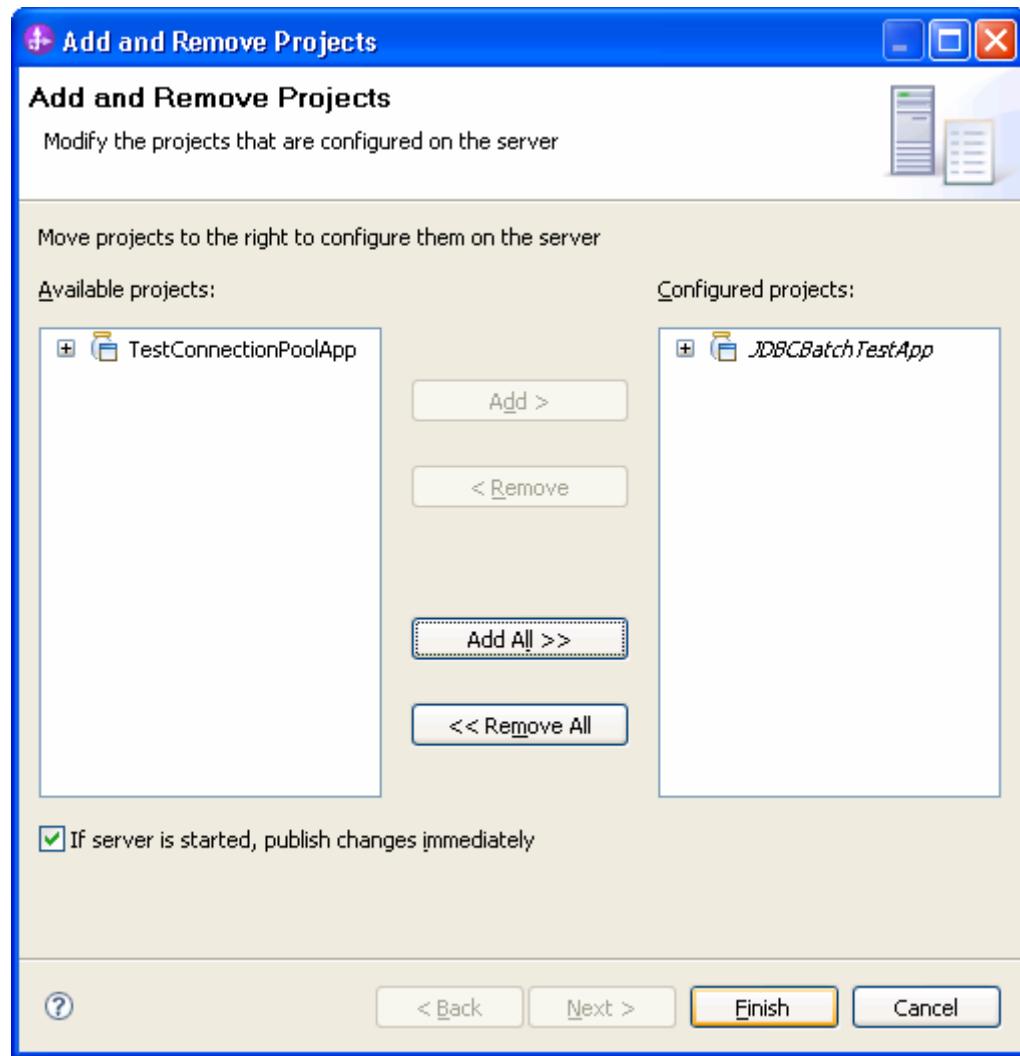
To do this, you must add the SCA module you created earlier to the server using the **Servers** view in WebSphere Integration Developer.

Steps for adding the SCA module to the server:

1. In WebSphere Integration Developer, switch to the **Servers** view by selecting **Windows > Show View > Servers**.
2. In the Servers tab in the lower-right pane of the WebSphere Integration Developer screen, right-click the server, and select **Start**.
3. After the server is started, right-click the server, and select **Add and Remove projects**.



4. Add the SCA module to the server.

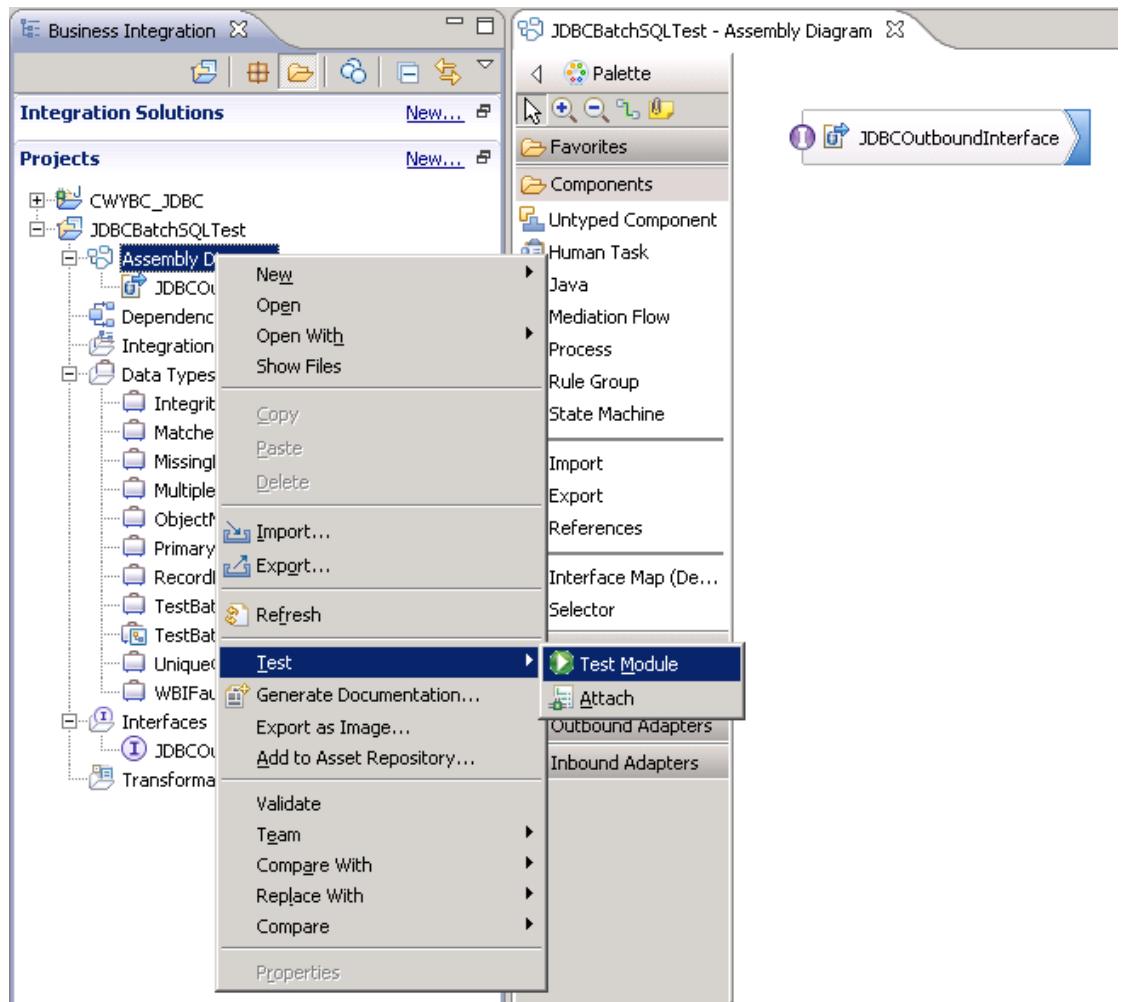


5. Click **Finish**.

Test the assembled adapter application

Test the assembled adapter application using the WebSphere Integration Developer integration test client.

1. Select the **JDBCBatchSQLTest** module, right-click, and select **Test > Test Module**.



The screenshot shows the WebSphere Integration Developer interface. On the left, there's a toolbar with icons for file operations like New, Open, Save, and Run. Below the toolbar, the title bar says "Invoke". The main area has two tabs: "General Properties" and "Detailed Properties". Under "General Properties", the configuration is set to "Default Module Test", module is "JDBCBatchSQLTest", component is "JDBCOutboundInterface", interface is "JDBCOutboundInterface", and operation is "executeTestBatchSQLBG". Below these tabs, a section titled "Initial request parameters" displays a table of parameters. The table has columns for Name, Type, and Value. The rows show nested parameter structures. For example, under "executeTestBatchSQLBG", there are "verb" (string) and "TestBatchSQL" (TestBatchSQL). "verb" has a value of "Create". "TestBatchSQL" has several "state" entries with values like "TC", "John", "McNay", "IBM", "Test1", "Test", and two "int" values both set to 1.

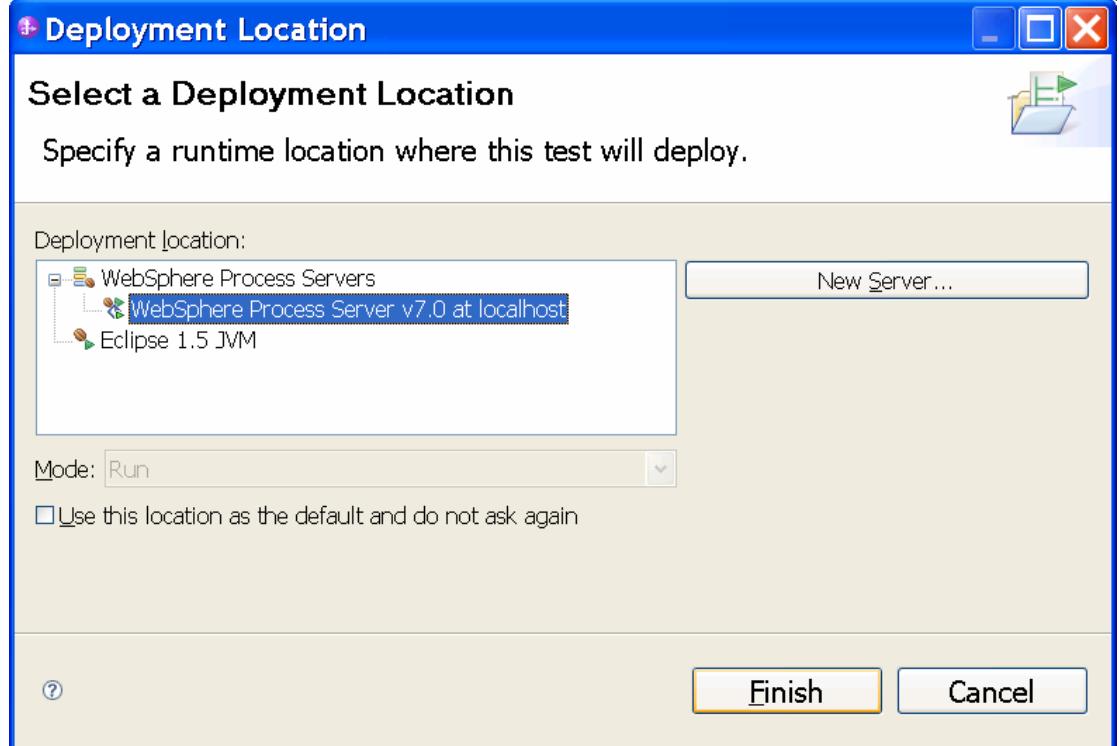
Name	Type	Value
executeTestBatchSQLBG	TestBatchSQLBG	✓
verb	verb<string>	✓ Create
TestBatchSQL	TestBatchSQL	✓
state1	string	✓ TC
state2	string	✓ John
state3	string	✓ McNay
state4	string	✓ IBM
state5	string	✓ IBM
state6	string	✓ Test1
state7	string	✓ Test
state8	int	✗
state9	int	✗
state10	int	✓

- Populate data for parameters as shown in the figure below.

This screenshot shows a detailed view of the parameter table from the previous step. The table has columns for Name, Type, and Value. It shows the same hierarchical structure of parameters as the first screenshot, with "executeTestBatchSQLBG" at the top, followed by "verb" and "TestBatchSQL". The "TestBatchSQL" row has several child parameters: "statement1paramete", "statement1paramete", "statement1paramete", "statement1paramete", "statement2paramete", "statement2paramete", "statement3paramete", "statement1status", "statement2status", and "statement3status". Each of these child parameters has its own type and value listed in the table.

Name	Type	Value
executeTestBatchSQLBG	TestBatchSQLBG	✓
verb	verb<string>	✗✓
TestBatchSQL	TestBatchSQL	✓
statement1paramete	string	✓ TC
statement1paramete	string	✓ John
statement1paramete	string	✓ McNay
statement1paramete	string	✓ IBM
statement2paramete	string	✓ IBM
statement3paramete	string	✓ Test1
statement1status	int	✓ 1
statement2status	int	✓ 1
statement3status	int	✓ 1

- To execute the service, click **Continue** .
- In the Select a Deployment location window, select the server and click **Finish**.



- Check the output of the service, and check the data in the Enterprise Information System to ensure it matches the expected values.

Name	Type	Value
executeTestBatchSQLBGInput	TestBatchSQLBG	✓
verb	verb<string>	✗✓
TestBatchSQL	TestBatchSQL	✓
statement1parameter	string	✓ TestCustomer
statement1parameter	string	✓ John
statement1parameter	string	✓ McNay
statement1parameter	string	✓ IBM
statement2parameter	string	✓ IBM
statement3parameter	string	✓ Test1
statement1status	int	✓ Test
statement2status	int	✓ 1
statement3status	int	✓ 1

Clear the sample content

After you have tested the application, clear the sample content to return the data to its original state.

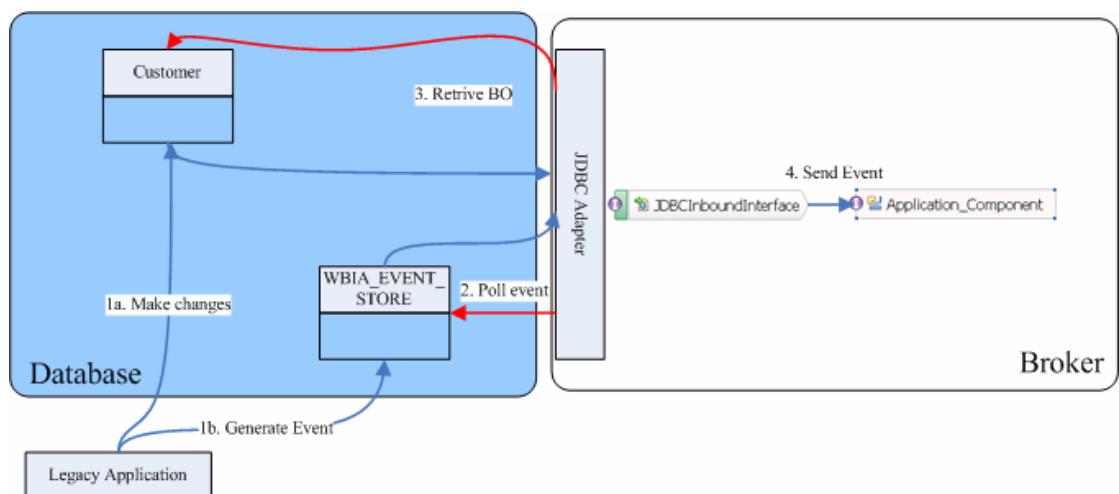
Chapter 6. Tutorial 5: Receiving events from the Enterprise Information System (Oracle)

This tutorial demonstrates how WebSphere Adapter for JDBC 7.0.0.0 polls the inbound events from the database table.

About this task

In this scenario, a legacy application makes a change to the CUSTOMER table in a single operation. Here we will insert an event record into the event table (WBIA_EVENT_TABLE). The JDBC adapter will poll the events from the database periodically. If a new event found, it will fetch the event and corresponding business objects from database. Finally, the JDBC adapter will convert the event to a SDO and send it to the destination SCA component.

The following figure represents the whole scenario:



Prepare to run through the tutorial

Extract the sample files

Replicas of the artifacts that you create when using the external service wizard are provided as sample files for your reference. Use these files to verify if the files you create using the external service wizard are correct.

Download the sample zip file and extract it into a directory of your choice (you may want to create a new directory).

Configuration prerequisites

Before configuring the adapter, you must complete the following tasks:

- Create tables and stored procedures
- Create an authentication alias
- Create a data source

Create tables and stored procedures

You must create the following tables and stored procedures in the Oracle database before starting the scenario.

a. Script for creating the tables

```
CREATE TABLE CUSTOMER (
    PKEY VARCHAR2(10) NOT NULL PRIMARY KEY,
    FNAME VARCHAR2(20) ,
    LNAME VARCHAR2(20) ,
    CCODE VARCHAR2(10) ) ;

CREATE SEQUENCE EVENT_SEQ INCREMENT BY 1 START WITH
1 MINVALUE 1 CACHE 20 ;

CREATE TABLE WBIA_JDBC_EVENTSTORE
(
    EVENT_ID INTEGER NOT NULL PRIMARY KEY,
    XID          VARCHAR2(200),
    OBJECT_KEY    VARCHAR2(80)      NOT NULL,
    OBJECT_NAME   VARCHAR2(40)      NOT NULL,
    OBJECT_FUNCTION VARCHAR2(40)      NOT
NULL,
    EVENT_PRIORITY INTEGER        NOT
NULL,
    EVENT_TIME     TIMESTAMP,
    EVENT_STATUS   INTEGER        NOT NULL,
    EVENT_COMMENT  VARCHAR2(100)
) ;
```

b. Script for creating triggers for Inbound

```
CREATE OR REPLACE TRIGGER EVENT_CREATE AFTER INSERT
ON CUSTOMER
REFERENCING OLD AS O NEW AS N
FOR EACH ROW
BEGIN
INSERT INTO wbia_jdbc_eventstore (event_id,
object_key, object_name,object_function,
event_priority, event_status)
VALUES (event_seq.nextval,:N.pkey,
'SampleCustomerBG', 'Create', 1, 0);
END;
/

CREATE OR REPLACE TRIGGER EVENT_DELETE AFTER DELETE
ON CUSTOMER
REFERENCING OLD AS O NEW AS N
FOR EACH ROW
BEGIN
INSERT INTO wbia_jdbc_eventstore (event_id,
object_key, object_name,object_function,
event_priority, event_status)
VALUES (event_seq.nextval,:O.pkey,
'SampleCustomerBG', 'Delete', 1, 0);
END;
/

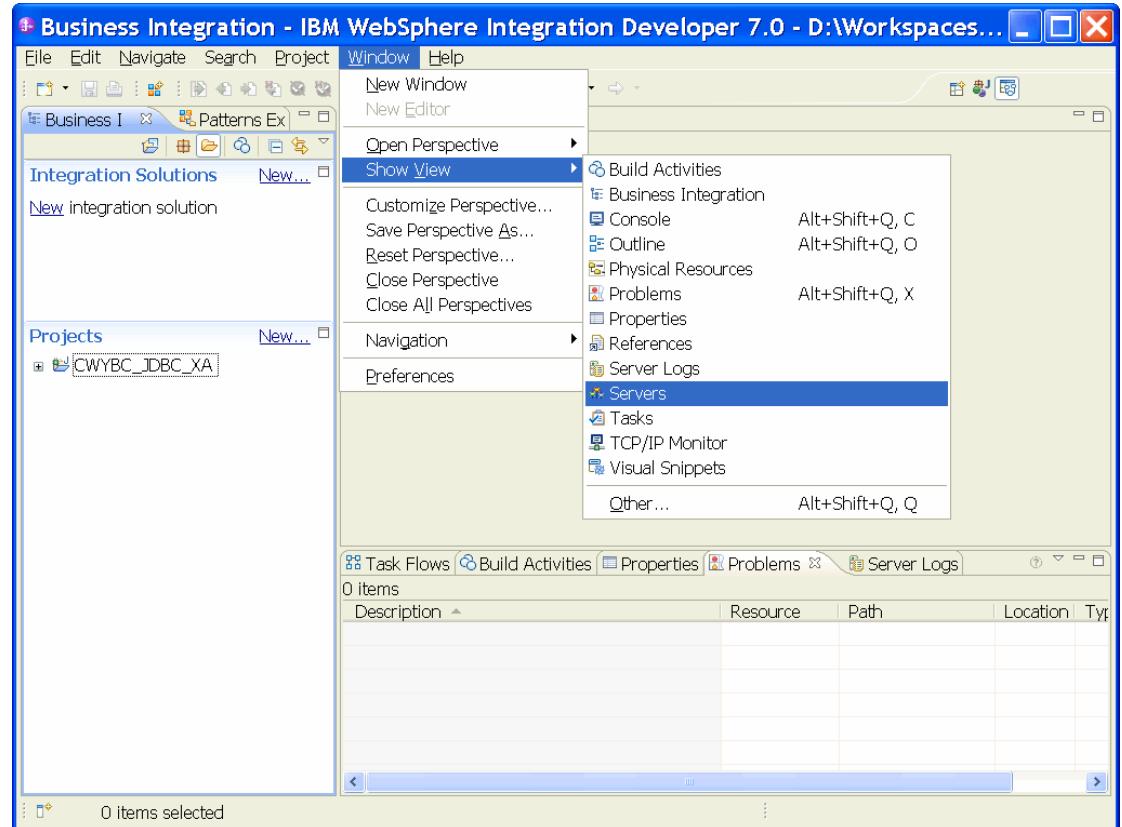
CREATE OR REPLACE TRIGGER EVENT_UPDATE AFTER UPDATE
OF PKEY, CCODE, FNAME, LNAME ON CUSTOMER
REFERENCING OLD AS O NEW AS N
FOR EACH ROW
BEGIN
INSERT INTO wbia_jdbc_eventstore (event_id,
object_key, object_name, object_function,
event_priority, event_status)
VALUES (event_seq.nextval,:N.pkey,
'SampleCustomerBG', 'Update', 1, 0);
END;
/
```

Create an authentication alias

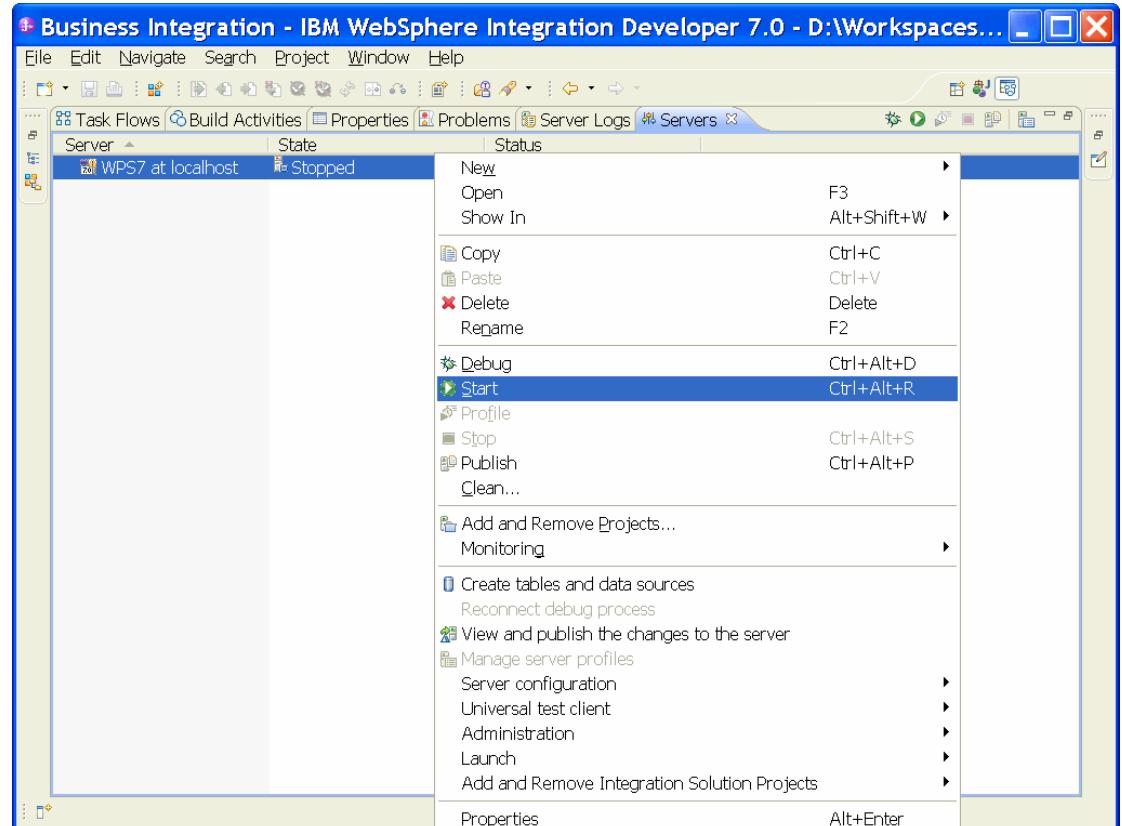
The authentication alias needs to be set because the data source created in the next section uses the username and password set in the authentication alias to connect to the database.

Follow these steps to set the authentication alias in the WebSphere Process Server administrative console.

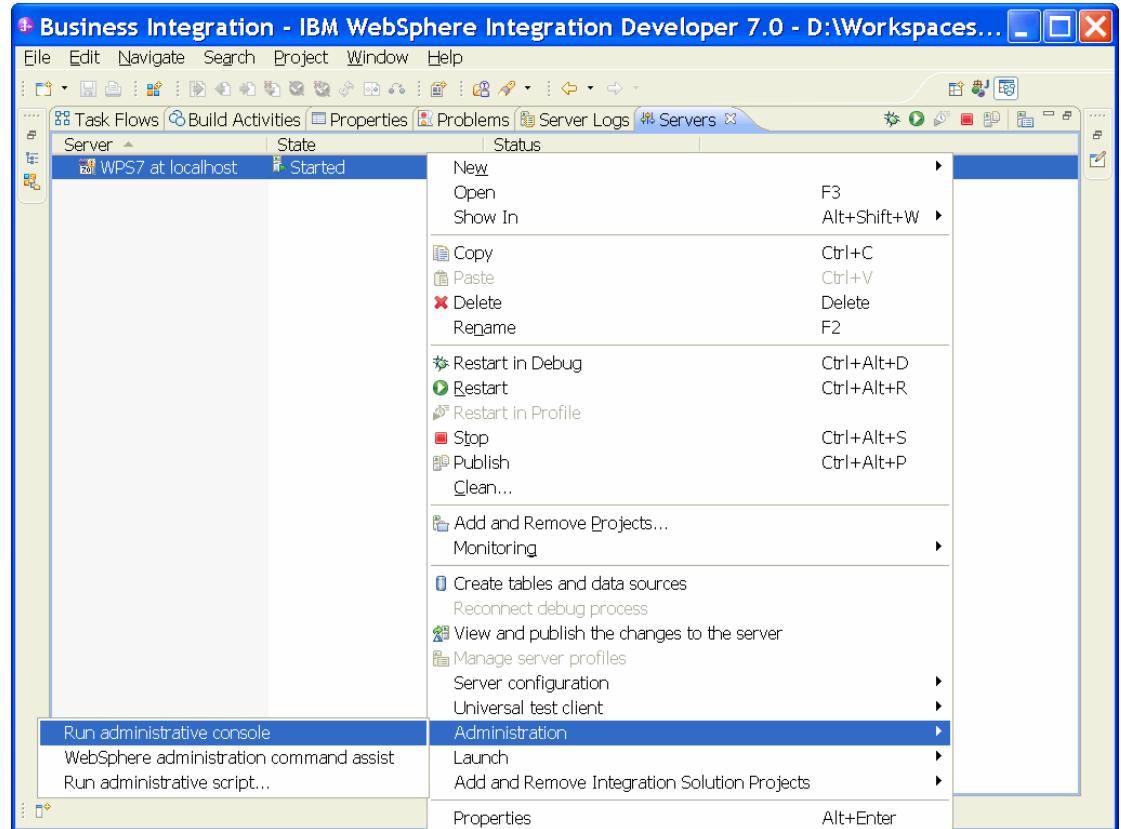
1. In WebSphere Integration Developer, switch to the **Servers** view by selecting **Windows > Show View > Servers**.



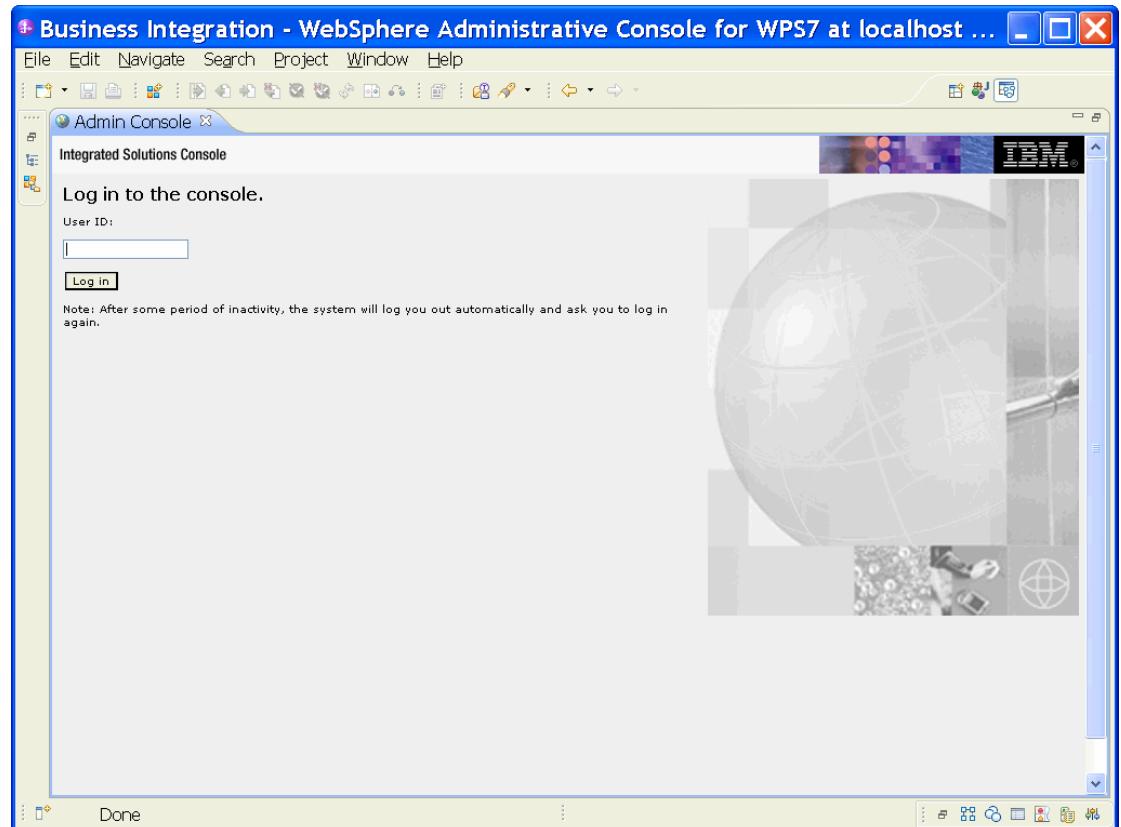
2. In the **Servers** view, right-click the server that you want to start and select **Start**.



3. After the server is started, right-click the server, and select **Administration > Run administrative console**.



4. Log on to the administrative console.



5. Click **Security → Global security**.

WebSphere software

The screenshot shows the left-hand navigation pane of the WebSphere software. At the top, there is a dropdown menu labeled "View: All tasks". Below it is a hierarchical list of management categories:

- Welcome
- Guided Activities
- Servers
- Applications
- Services
- Resources
- Security
 - Business Integration Security
 - Global security (highlighted with a red box)
 - Security domains
 - Administrative Authorization Groups
 - SSL certificate and key management
 - Security auditing
 - Bus security
- Environment
- Integration Applications
- System administration
- Users and Groups
- Monitoring and Tuning
- Troubleshooting
- Service integration
- UDDI

6. Under **Java Authentication and Authorization Service**, click **J2C authentication data**.

Global security

Use this panel to configure administration and the default application security policy. This security configuration applies to functions and is used as a default security policy for user applications. Security domains can be defined to override and control applications.

[Security Configuration Wizard](#)[Security Configuration Report](#)**Administrative security**

- Enable administrative security
 - [Administrative user roles](#)
 - [Administrative group roles](#)
 - [Administrative authentication](#)

Application security

- Enable application security

Java 2 security

- Use Java 2 security to restrict application access to local resources
 - Warn if applications are granted custom permissions
 - Restrict access to resource authentication data

User account repository

Current realm definition

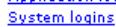
Federated repositories

Available realm definitions

Federated repositories

[Configure...](#)[Set as current](#)**Authentication**

Authentication mechanisms and expiration

[LTPA](#)[Kerberos and LTPA](#)[Kerberos configuration](#)[SWAM \(deprecated\): No authentication](#)[Authentication cache settings](#)[Web and SIP security](#)[RMI/IOP security](#)[Java Authentication and Authorization](#)[Application logins](#)[System logins](#)[J2C authentication data](#)

Specifies a list of user identities and passwords for Java(TM) 2 connector security to use.

[Security domains](#)[External authorization providers](#)[Custom properties](#)

A list of existing aliases is displayed.

WebSphere software

Global security > JAAS - J2C authentication data

Specifies a list of user identities and passwords for Java(TM) 2 connector security to use.

Prefix new alias names with the node name of the cell (for compatibility with earlier releases)

Apply

Preferences

New Delete			
Select	Alias ▾	User ID ▾	Description ▾
You can administer the following resources:			
<input type="checkbox"/>	BSpace JDBC Alias	wbiuser	Business Space Authorization Alias
<input type="checkbox"/>	CommonEventInfrastructureJMSAuthAlias	wbiuser	Authentication alias for the Common Event Infrastructure JMS Topics and Queues
<input type="checkbox"/>	SCA Auth Alias	wbiuser	This is the alias used by SCA to login to a secured SIBus
<input type="checkbox"/>	localhostNode01Cell/nlNode01/server1/EventAuthDataAliasDerby		Derby authentication alias for the Event Server
Total 4			

7. Click **New** to create a new authentication entry. Type the alias name, and username and password to connect to the database. Click **OK**.

WebSphere software

Cell=localhostNode01Cell, Profile=AppSrv01

Global security

[Global security](#) > [JAAS - J2C authentication data](#) > New

Specifies a list of user identities and passwords for Java(TM) 2 connector security to use.

General Properties

* Alias: Alias_Oracle
* User ID: sample
* Password:
Description:

[Apply] [OK] [Reset] [Cancel]

8. Click **Save** to save the changes.

Cell=localhostNode01Cell, Profile=AppSrv01

Global security

Messages

⚠ Changes have been made to your local configuration. You can:

- [Save](#) directly to the master configuration.
- [Review](#) changes before saving or discarding.

⚠ The server may need to be restarted for these changes to take effect.

[Global security](#) > [JAAS - J2C authentication data](#)

Specifies a list of user identities and passwords for Java(TM) 2 connector security to use.

Prefix new alias names with the node name of the cell (for compatibility with earlier releases)

[Apply]

You have created an authentication alias that will be used to configure the data source.

WebSphere software

Preferences			
		New	Delete
Select	Alias	User ID	Description
You can administer the following resources:			
<input type="checkbox"/>	BSpace JDBC Alias	wbiuser	Business Space Authorization Alias
<input type="checkbox"/>	CommonEventInfrastructureJMSAuthAlias	wbiuser	Authentication alias for the Common Event Infrastructure JMS Topics and Queues
<input type="checkbox"/>	SCA Auth Alias	wbiuser	This is the alias used by SCA to login to a secured SIBus
<input type="checkbox"/>	localhostNode01Cell/nlNode01/server1/EventAuthDataAliasDerby		Derby authentication alias for the Event Server
<input type="checkbox"/>	nlNode01/Alias Oracle	luweiqin	
Total 5			

Create a data source

Create a data source in WebSphere Process Server, which the adapter will use to connect to the database. This data source is used later when generating the artifacts for the module.

Note: This tutorial uses Oracle as the database and the Oracle thin driver,ojdbc6.jar.

Here are the steps to create the data source in the WebSphere Process Server administrative console.

1. In the administrative console, select **Environment → WebSphere Variables**.

WebSphere software



2. On the right page, select **ORACLE_JDBC_DRIVER_PATH** and specify the path of the ojdbc6.jar file in the **Value** field. Click **OK**.

WebSphere software

Cell=localhostNode01Cell, Profile=AppSrv01

WebSphere Variables

[WebSphere Variables](#) > ORACLE_JDBC_DRIVER_PATH

Use this page to define substitution variables. Variables specify a level of indirection for some system-defined values, such as file system root directories. Variables have a scope level, which is either server, node, cluster, or cell. Values at one scope level can differ from values at other levels. When a variable has conflicting scope values, the more granular scope value overrides values at greater scope levels. Therefore, server variables override node variables, which override cluster variables, which override cell variables.

Configuration

General Properties

* Name: ORACLE_JDBC_DRIVER_PATH

Value: D:\Lib

Description: The directory that contains the Oracle thin or oci8 JDBC Driver.

Apply OK Reset Cancel

- Click **Save** to save the changes.

WebSphere Variables

Messages

Changes have been made to your local configuration. You can:

- Save directly to the master configuration.
- Review changes before saving or discarding.

The server may need to be restarted for these changes to take effect.

The variable has been added and appears in the list.

Preferences

New Delete

MQ JDBC DRIVER PATH

Select	Name	Value	Scope
You can administer the following resources:			
<input type="checkbox"/>	MQ_INSTALL_ROOT	`\${WAS_INSTALL_ROOT}/lib/WMQ	Node=n1Node01
<input type="checkbox"/>	ORACLE_JDBC_DRIVER_PATH	D:\Lib	Node=n1Node01
<input type="checkbox"/>	OS400_NATIVE_JDBC40_DRIVER_PATH		Node=n1Node01
<input type="checkbox"/>	OS400_NATIVE_JDBC_DRIVER_PATH		Node=n1Node01
<input type="checkbox"/>	OS400_TOOLBOX_JDBC_DRIVER_PATH		Node=n1Node01
<input type="checkbox"/>	ORACLE_JDBC_DRIVER_PATH		Node=n1Node01

WebSphere software

4. Select **Resources** → **JDBC** -> **JDBC Providers**.



5. Click **New** in the JDBC providers window.

JDBC providers

JDBC providers

Use this page to edit properties of a JDBC provider. The JDBC provider object encapsulates the specific JDBC driver implementation class for access to the specific vendor database of your environment. Learn more about this task in a [guided activity](#). A guided activity provides a list of task steps and more general information about the topic.

Scope: Cell=**localhostNode01Cell**, Node=**n1Node01**

Scope specifies the level at which the resource definition is visible. For detailed information on what scope is and how it works, [see the scope settings help](#).

Node=n1Node01

Preferences

Select	Name	Scope	Description
None			
Total 0			

6. Select an Oracle database with a connection pool data source for the Oracle JDBC driver. Click **Next**.

Create a new JDBC Provider

Create a new JDBC Provider

→ Step 1: Create new JDBC provider

Step 2: Enter database class path information

Step 3: Summary

Create new JDBC provider

Set the basic configuration values of a JDBC provider, which encapsulates the specific vendor JDBC driver implementation classes that are required to access the database. The wizard fills in the name and the description fields, but you can type different values.

Scope

cells:localhostNode01Cell:nodes:n1Node01

* Database type

Oracle

* Provider type

Oracle JDBC Driver

* Implementation type

Connection pool data source

* Name

Oracle JDBC Driver

Description

Oracle JDBC Driver

Next **Cancel**

7. In the Enter database classpath information page, enter the following value for the **Class path** field:
\$(ORACLE_JDBC_DRIVER_PATH)/ojdbc6.jar, where
\$(ORACLE_JDBC_DRIVER_PATH) is library path for the run time.
8. Click **Next**.

WebSphere software

Cell=localhostNode01Cell, Profile=AppSrv01

Create a new JDBC Provider

Step 1: Create new JDBC provider
→ Step 2: Enter database class path information
Step 3: Summary

Enter database class path information

Set the environment variables that represent the JDBC driver class files, which WebSphere(R) Application Server uses to define your JDBC provider. This wizard page displays the file names; you supply only the directory locations of the files. Use complete directory paths when you type the JDBC driver file locations. For example: C:\SQLLIB\java on Windows(R) or /home/db2inst1/sqllib/java on Linux(TM).

If a value is specified for you, you may click Next to accept the value.

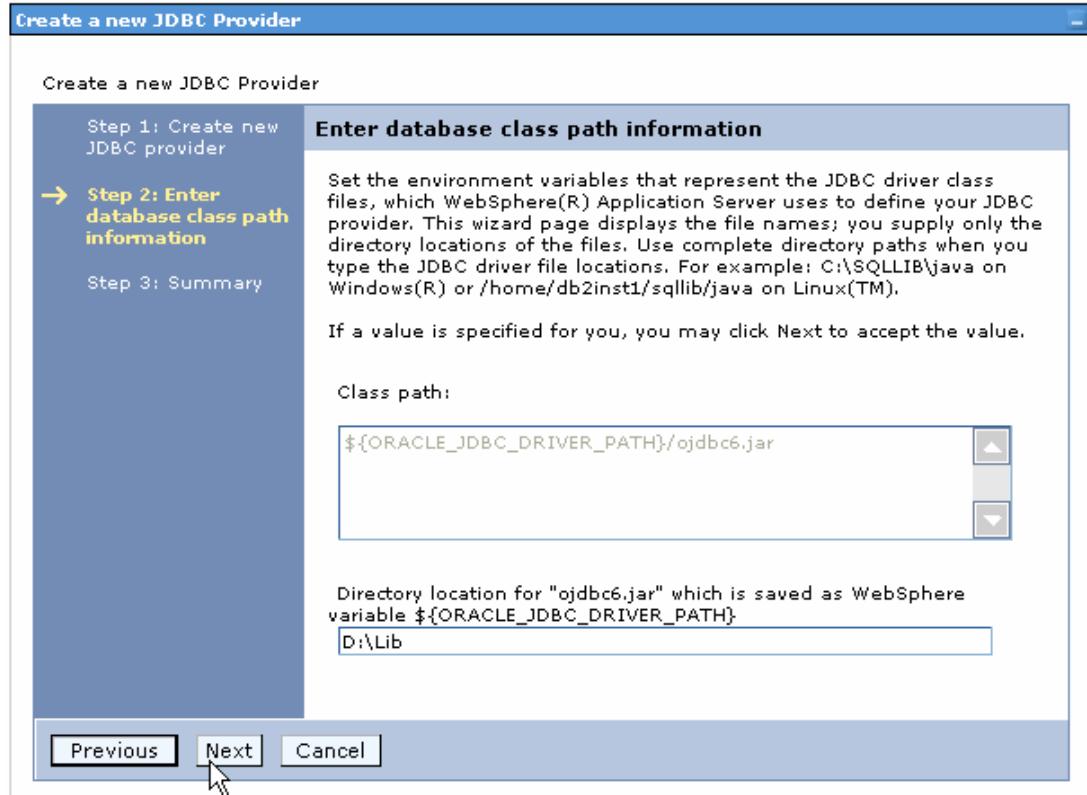
Class path:

`${ORACLE_JDBC_DRIVER_PATH}/ojdbc6.jar`

Directory location for "ojdbc6.jar" which is saved as WebSphere variable `${ORACLE_JDBC_DRIVER_PATH}`

D:\Lib

Previous **Next** Cancel



9. In the Summary page, click **Finish**.

Cell=localhostNode01Cell, Profile=AppSrv01 [Close page](#)

Create a new JDBC Provider

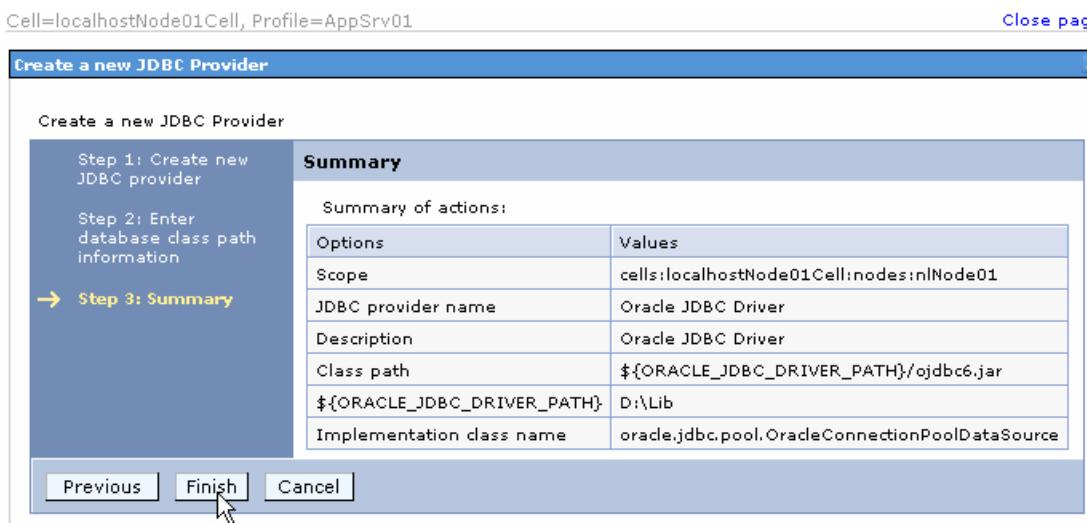
Step 1: Create new JDBC provider
Step 2: Enter database class path information
→ Step 3: Summary

Summary

Summary of actions:

Options	Values
Scope	cells:localhostNode01Cell:nodes:n1Node01
JDBC provider name	Oracle JDBC Driver
Description	Oracle JDBC Driver
Class path	<code> \${ORACLE_JDBC_DRIVER_PATH}/ojdbc6.jar</code>
<code> \${ORACLE_JDBC_DRIVER_PATH}</code>	D:\Lib
Implementation class name	oracle.jdbc.pool.OracleConnectionPoolDataSource

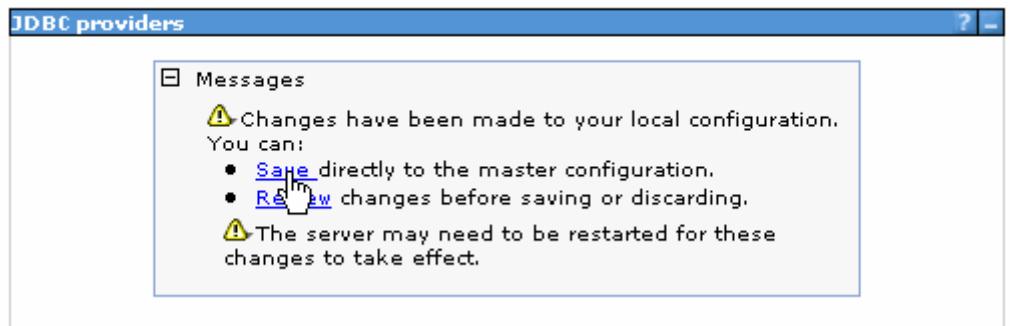
Previous **Finish** Cancel



10. Click **Save**.

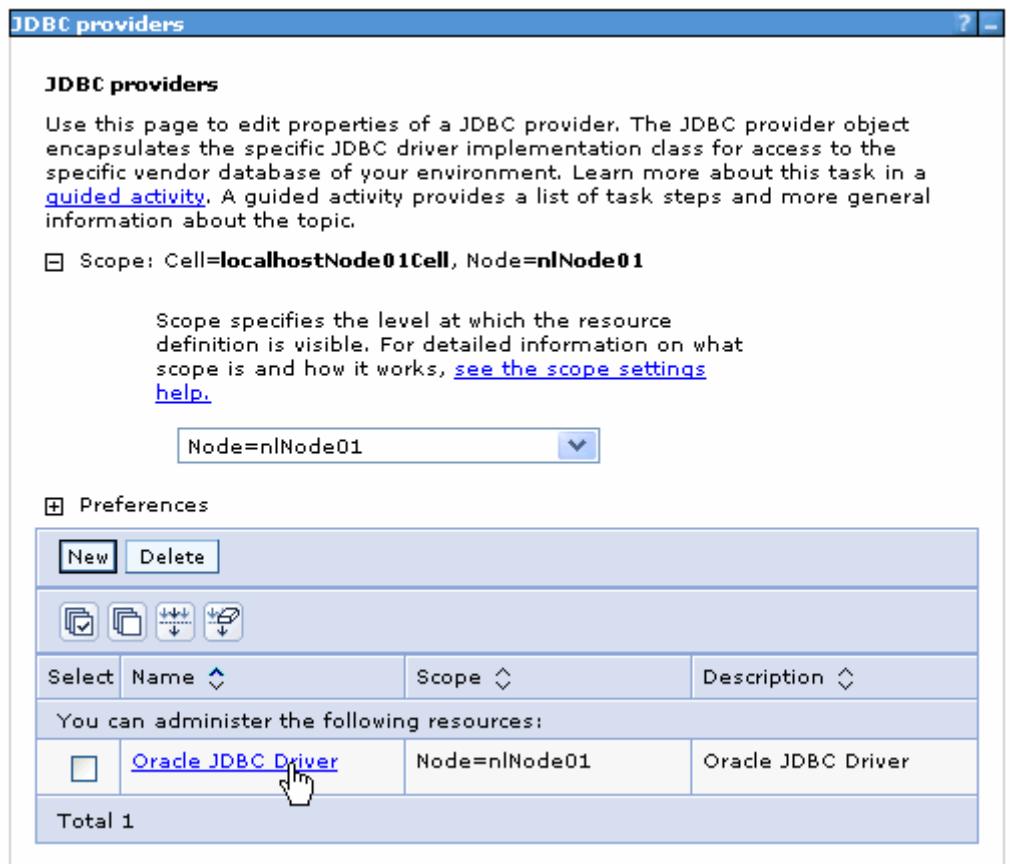
WebSphere software

Cell=localhostNode01Cell, Profile=AppSrv01



The JDBC provider is added and appears in the list.

Cell=localhostNode01Cell, Profile=AppSrv01



11. Select the Oracle JDBC provider you just created. Under **Additional Properties**, click **Data sources**. Click **New**.

WebSphere software

Cell=localhostNode01Cell, Profile=AppSrv01

JDBC providers

JDBC providers > Oracle JDBC Driver > Data sources

Use this page to edit the settings of a datasource that is associated with your selected JDBC provider. The datasource object supplies your application with connections for accessing the database. Learn more about this task in a [guided activity](#). A guided activity provides a list of task steps and more general information about the topic.

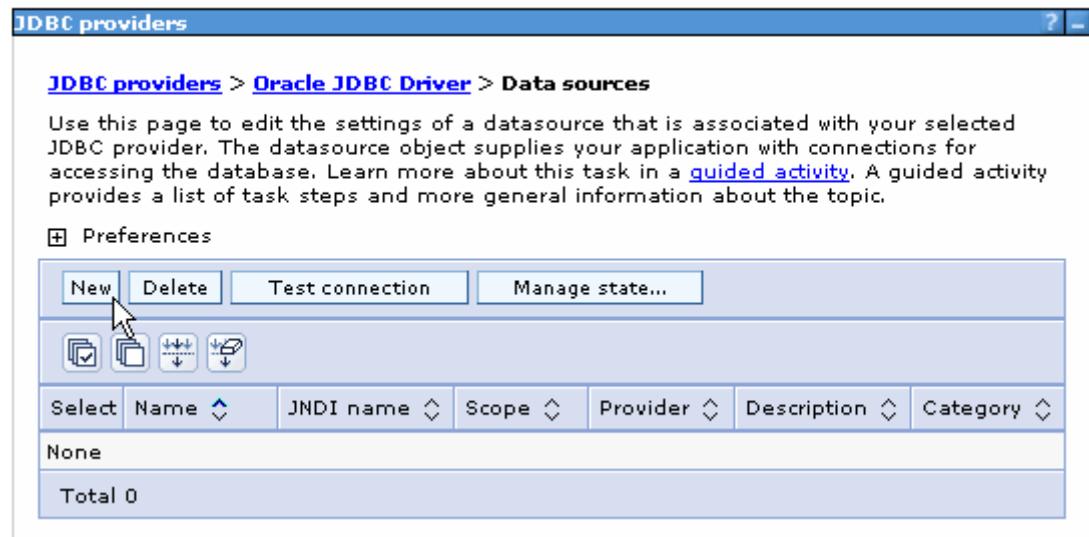
Preferences

New Delete Test connection Manage state...

Select Name JNDI name Scope Provider Description Category

None

Total 0



12. Type any value in the **JNDI name** field, and select the authentication alias. Click **Next**.

Cell=localhostNode01Cell, Profile=AppSrv01

Create a data source

Create a data source

Step 1: Enter basic data source information

Step 2: Enter database specific properties for the data source

Step 3: Setup security aliases

Step 4: Summary

Enter basic data source information

Set the basic configuration values of a datasource for association with your JDBC provider. A datasource supplies the physical connections between the application server and the database.

Requirement: Use the Datasources (WebSphere(R) Application Server V4) console pages if your applications are based on the Enterprise JavaBeans(TM) (EJB) 1.0 specification or the Java(TM) Servlet 2.2 specification.

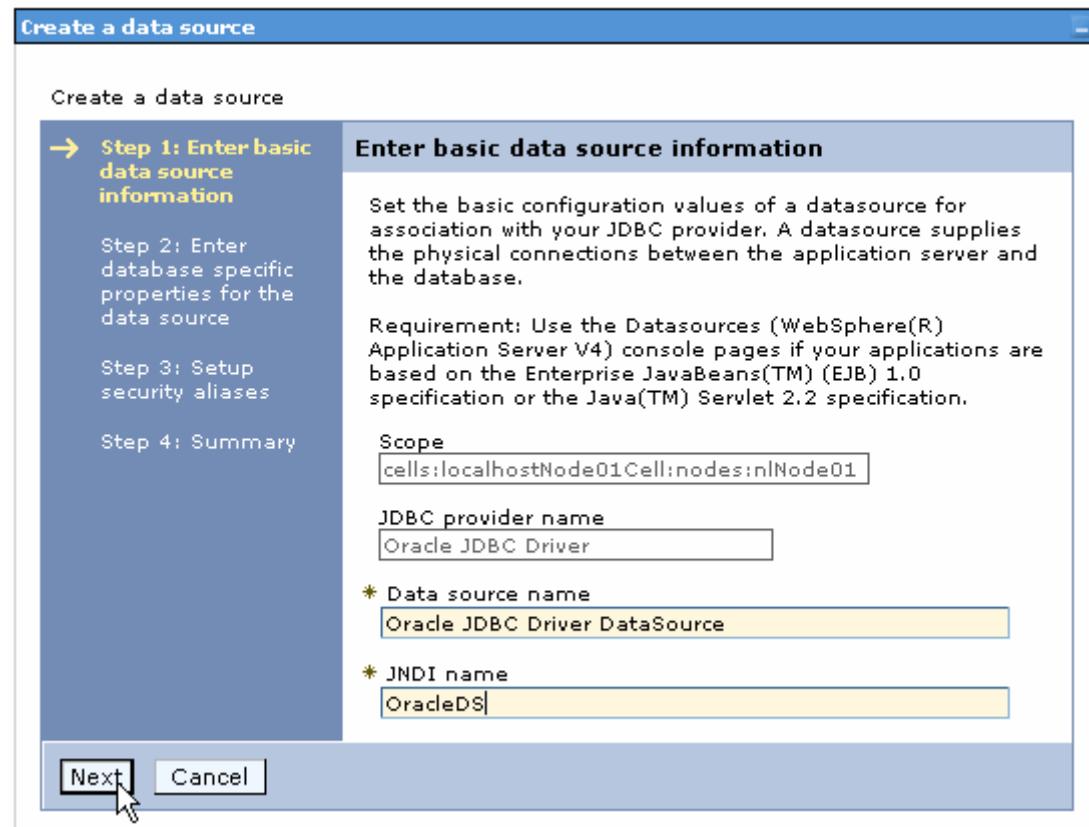
Scope: cells:localhostNode01Cell:nodes:n1Node01

JDBC provider name: Oracle JDBC Driver

* Data source name: Oracle JDBC Driver DataSource

* JNDI name: OracleDS

Next **Cancel**



13. Provide the appropriate URL value and select a data store helper class name from the **Data store helper class name** list as shown in the following figure. Click **Next**.

WebSphere software

Cell=localhostNode01Cell, Profile=AppSrv01

Create a data source

Step 1: Enter basic data source information

→ Step 2: Enter database specific properties for the data source

Step 3: Setup security aliases

Step 4: Summary

Enter database specific properties for the data source

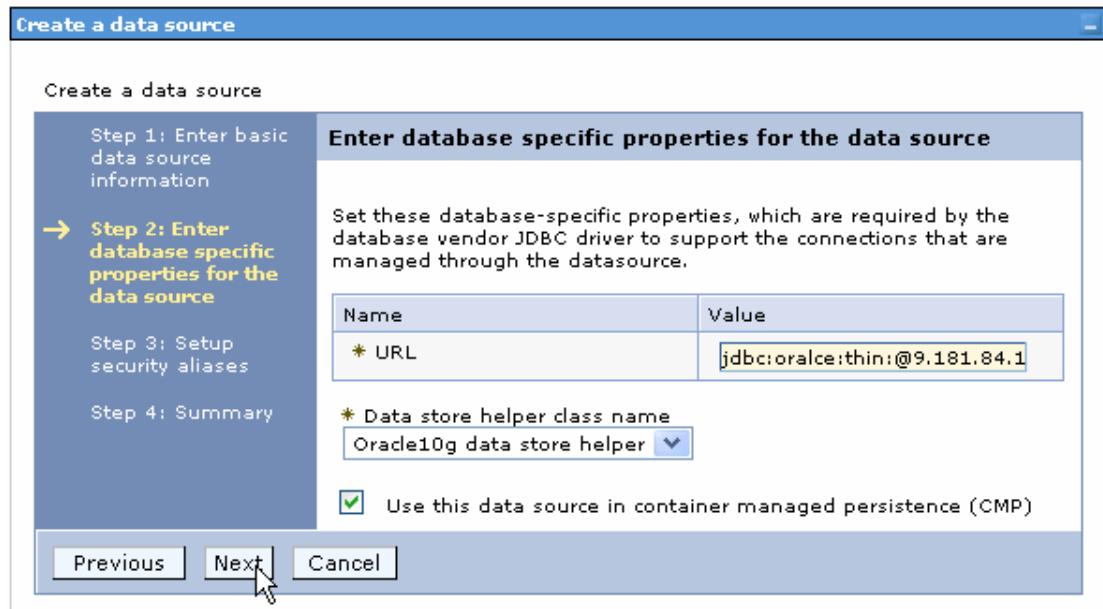
Set these database-specific properties, which are required by the database vendor JDBC driver to support the connections that are managed through the datasource.

Name	Value
* URL	jdbc:oracle:thin:@9.181.84.1

* Data store helper class name
Oracle10g data store helper

Use this data source in container managed persistence (CMP)

Previous Next Cancel



14. Select the authentication alias you just created from the **Component-managed authentication alias** field and click **Next**.

Cell=localhostNode01Cell, Profile=AppSrv01 Close

Create a data source

Step 1: Enter basic data source information

Step 2: Enter database specific properties for the data source

→ Step 3: Setup security aliases

Step 4: Summary

Setup security aliases

Select the authentication values for this resource.

Component-managed authentication alias
nlNode01/Alias_Oracle

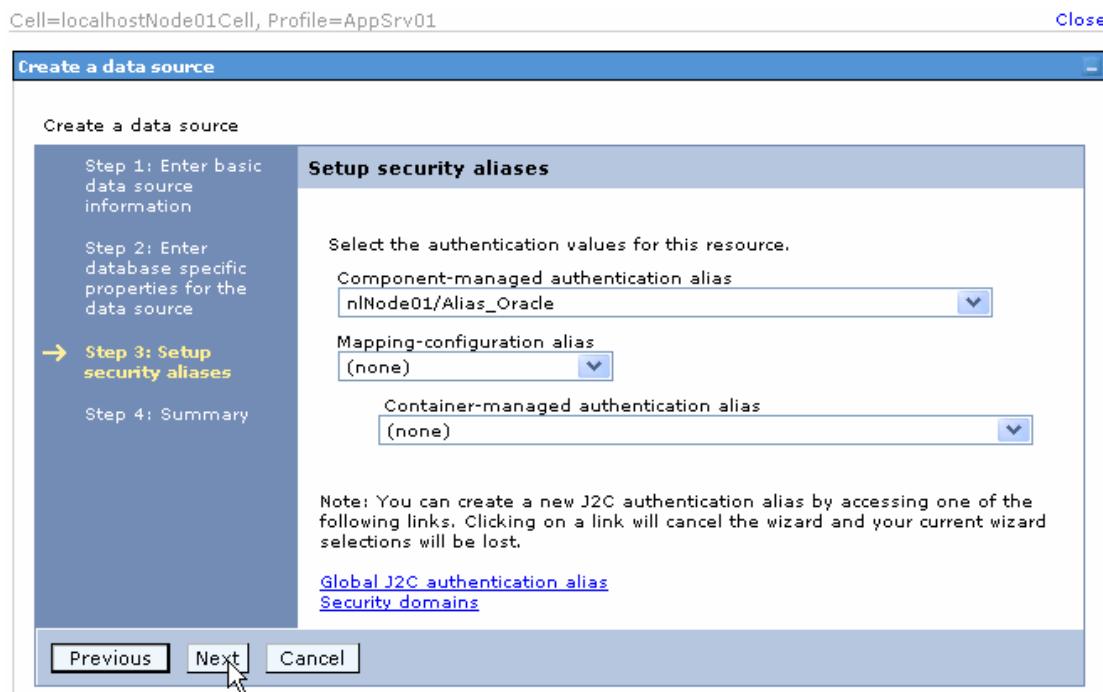
Mapping-configuration alias
(none)

Container-managed authentication alias
(none)

Note: You can create a new J2C authentication alias by accessing one of the following links. Clicking on a link will cancel the wizard and your current wizard selections will be lost.

[Global J2C authentication alias](#)
[Security domains](#)

Previous Next Cancel



15. In the Summary page, review the values entered for the data source and click **Finish**.

WebSphere software

Create a data source

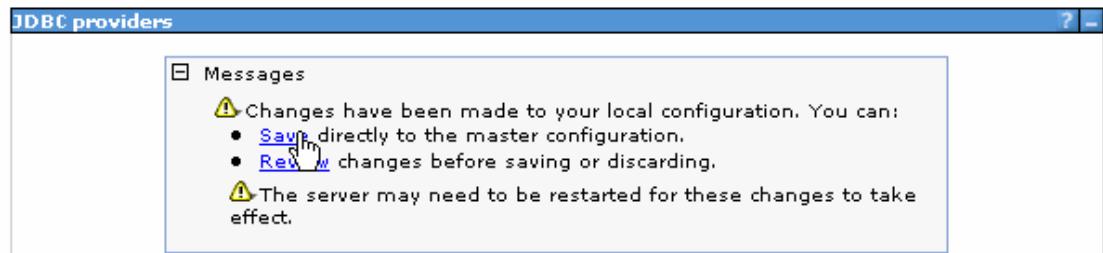
Step 1: Enter basic data source information
Step 2: Enter database specific properties for the data source
Step 3: Setup security aliases
→ Step 4: Summary

Summary	
Summary of actions:	
Options	Values
Scope	cells:localhostNode01Cell:nodes:n1Node01
Data source name	Oracle JDBC Driver DataSource
JNDI name	OracleDS
Select an existing JDBC provider	Oracle JDBC Driver
Implementation class name	oracle.jdbc.pool.OracleConnectionPoolDataSource
URL	jdbc:oracle:thin:@9.181.84.136:1521:ord
Data store helper class name	com.ibm.websphere.radapter.Oracle10gDataStoreHelper
Use this data source in container managed persistence (CMP)	true
Component-managed authentication alias	n1Node01/Alias_Oracle
Mapping-configuration alias	(none)
Container-managed authentication alias	(none)

Previous **Finish** Cancel

16. Click **Save** to save the changes.

Cell=localhostNode01Cell, Profile=AppSrv01



17. Select the data source you just created and click **Test connection**.

Cell=localhostNode01Cell, Profile=AppSrv01

JDBC providers

[JDBC providers](#) > [Oracle JDBC Driver](#) > Data sources

Use this page to edit the settings of a datasource that is associated with your selected JDBC provider. The datasource object supplies your application with connections for accessing the database. Learn more about this task in a [guided activity](#). A guided activity provides a list of task steps and more general information about the topic.

Preferences

New	Delete	Test connection	Manage state...
Select	Name ▾	JNDI name ▾	Scope ▾
Provider ▾	Description ▾	Category ▾	

You can administer the following resources:

<input checked="" type="checkbox"/>	Oracle JDBC Driver DataSource	OracleDS	Node=n1Node01	Oracle JDBC Driver	New JDBC Datasource	
-------------------------------------	---	----------	---------------	--------------------	---------------------	--

Total 1

The connection should succeed as indicated by the message shown in the following figure. If you experience problems with the test connection, refer to the “Troubleshooting” section.

Cell=localhostNode01Cell, Profile=AppSrv01

JDBC providers

[JDBC providers](#) > [Oracle JDBC Driver](#) > Data sources

Messages

The test connection operation for data source Oracle JDBC Driver DataSource on server server1 at node n1Node01 was successful.

[Information](#)

Use this page to edit the settings of a datasource that is associated with your selected JDBC provider. The datasource object supplies your application with connections for accessing the database. Learn more about this task in a [guided activity](#). A guided activity provides a list of task steps and more general information about the topic.

Preferences

New	Delete	Test connection	Manage state...
Select	Name ▾	JNDI name ▾	Scope ▾
Provider ▾	Description ▾	Category ▾	

You can administer the following resources:

<input type="checkbox"/>	Oracle JDBC Driver DataSource	OracleDS	Node=n1Node01	Oracle JDBC Driver	New JDBC Datasource	
--------------------------	---	----------	---------------	--------------------	---------------------	--

Total 1

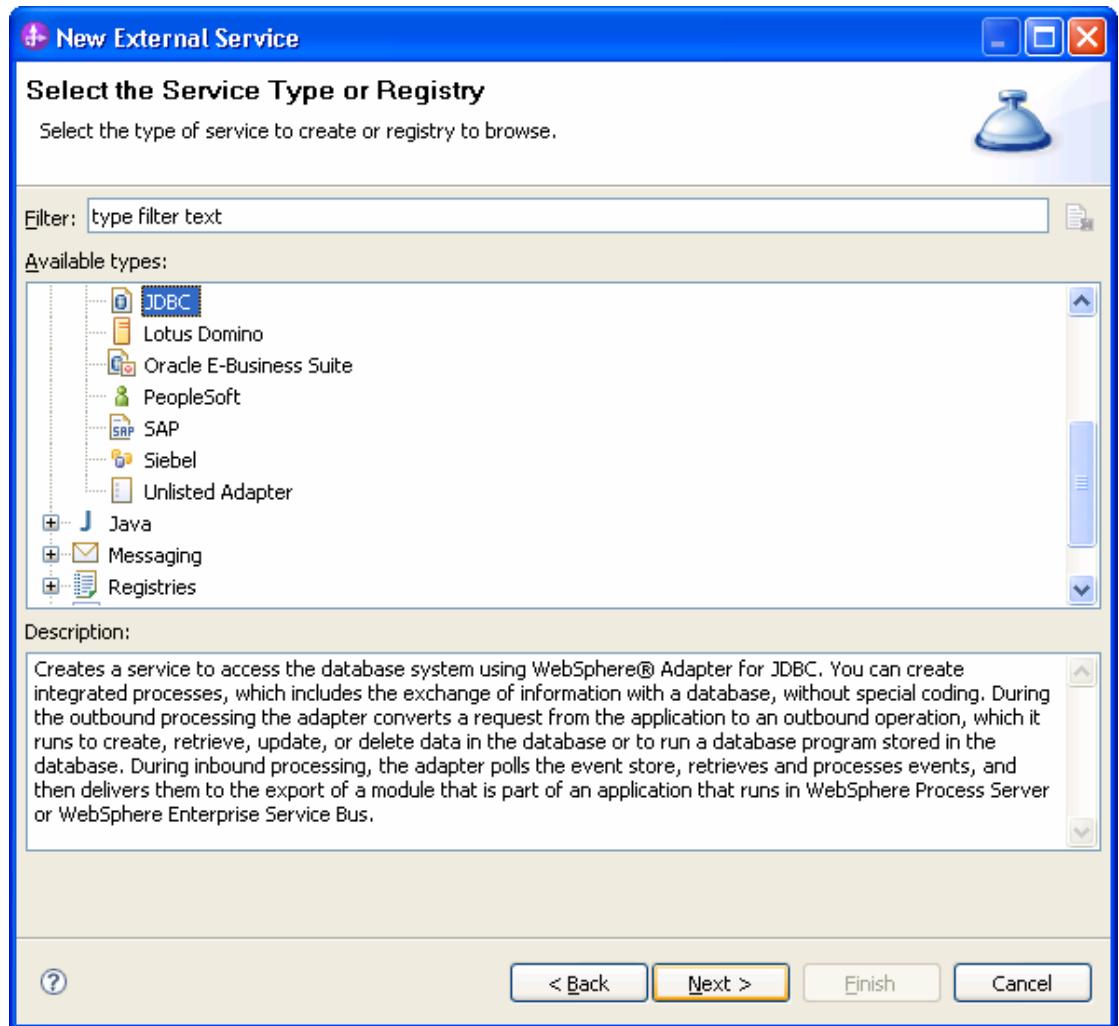
The data source is created and it will be used by the adapter to connect to the database.

Configure the adapter for inbound processing

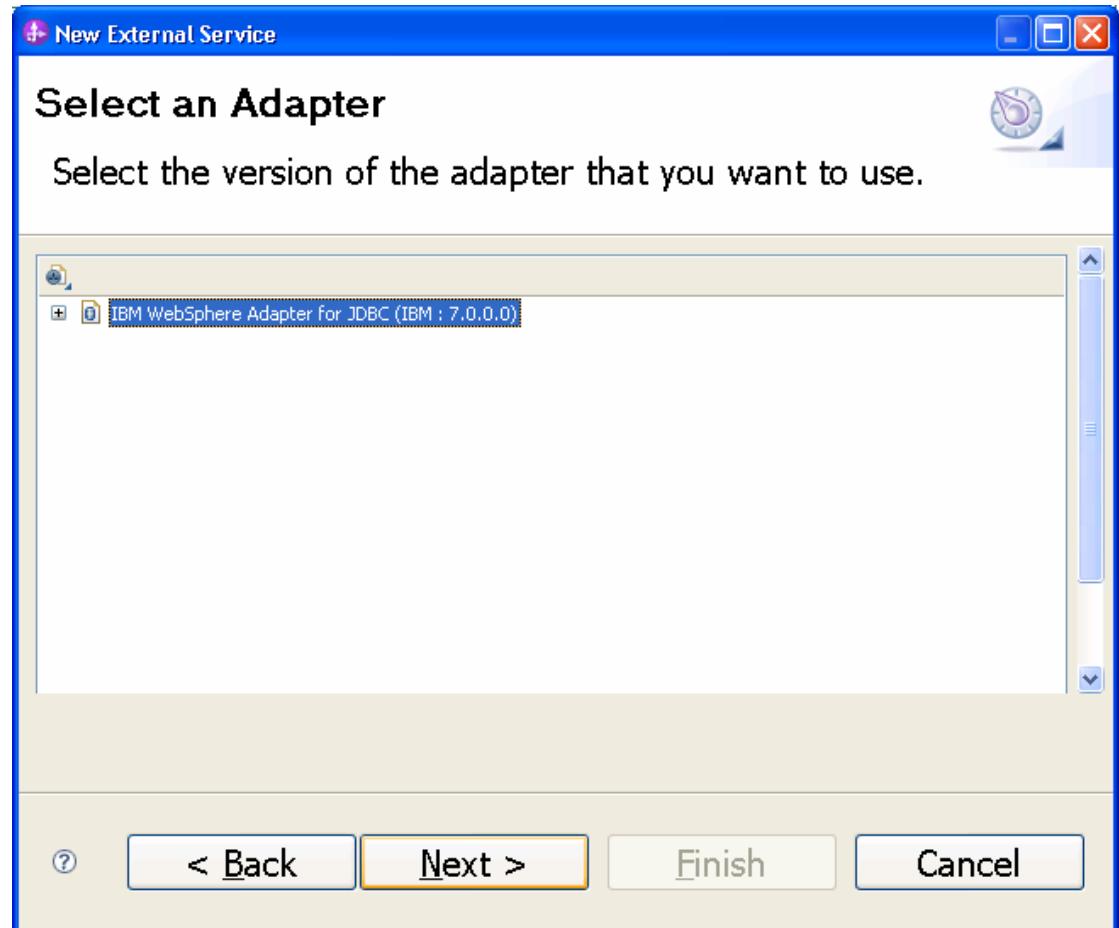
WebSphere software

Run the external service wizard to specify business objects, services, and configuration details.

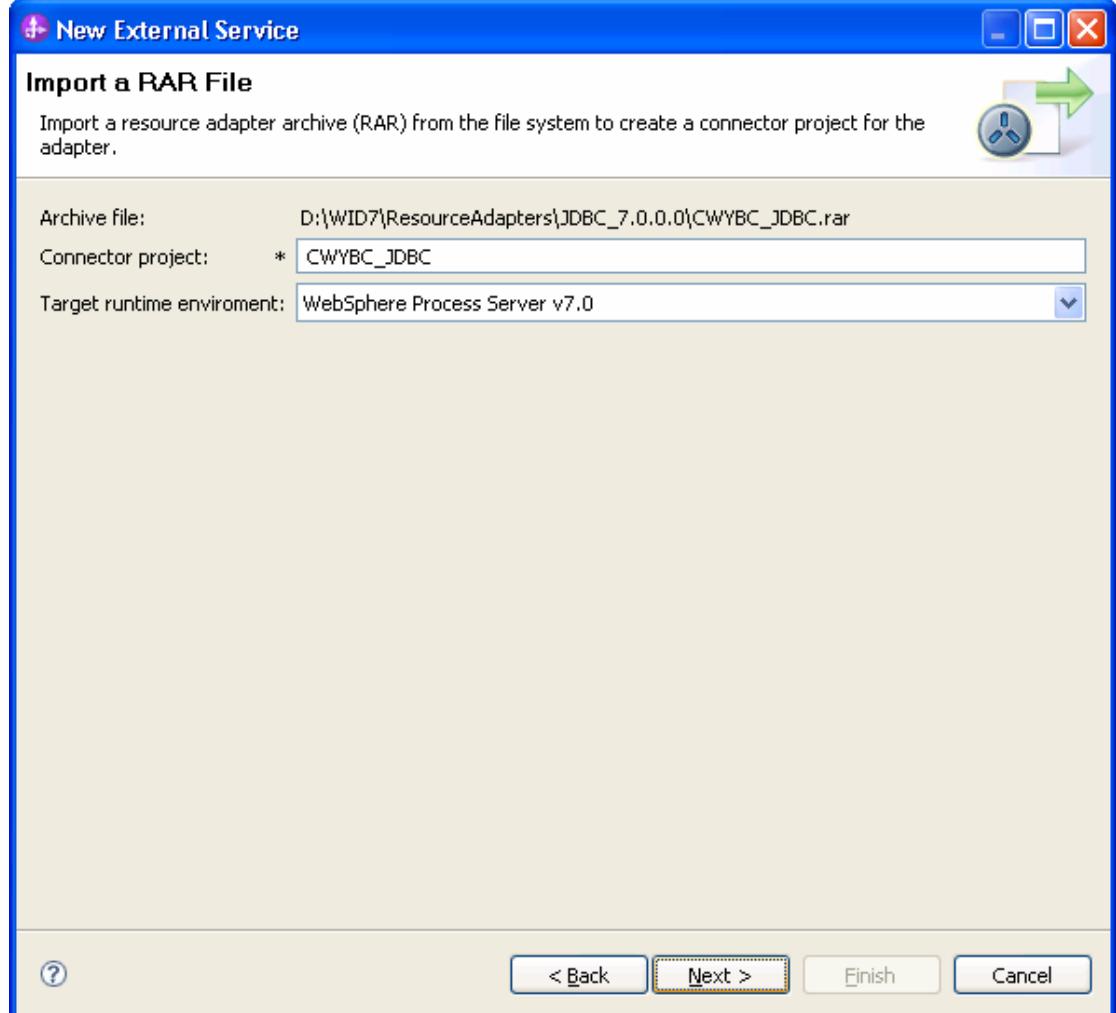
1. Switch to the Business Integration Perspective in WebSphere Integration Developer by selecting **Window -> Open Perspective Business Integration**.
2. Start the external service wizard by selecting **File-> New -> External Service**.
3. In the **Available Types** area, select **Adapters > JDBC** and click **Next**.



4. Select **IBM WebSphere Adapter for JDBC (IBM: 7.0.0.0)** and click **Next**.

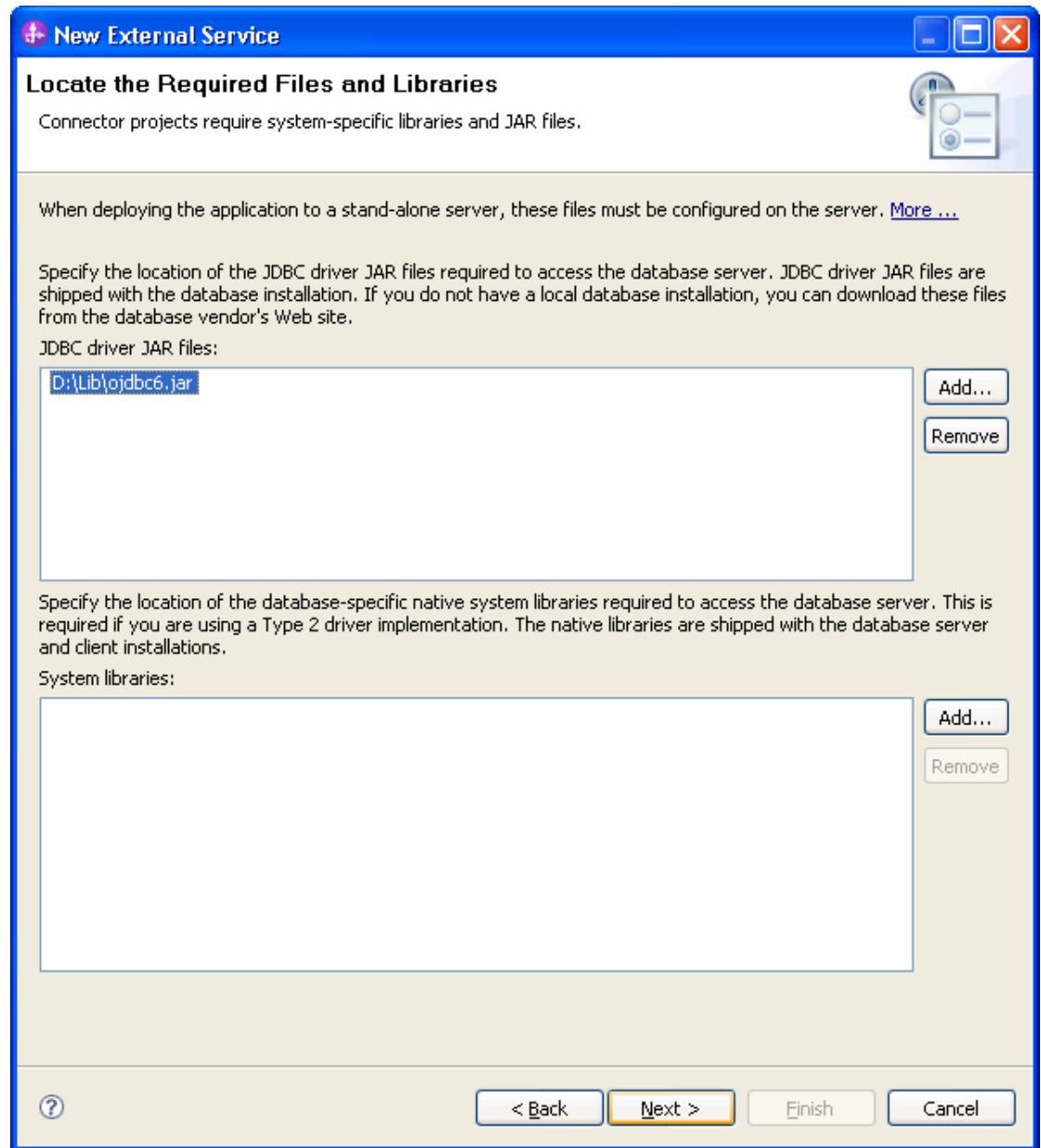


5. In the **Connector project** field enter **CWYBC_JDBC**.
6. In the **Target runtime environment** field, select the appropriate runtime and click **Next**.

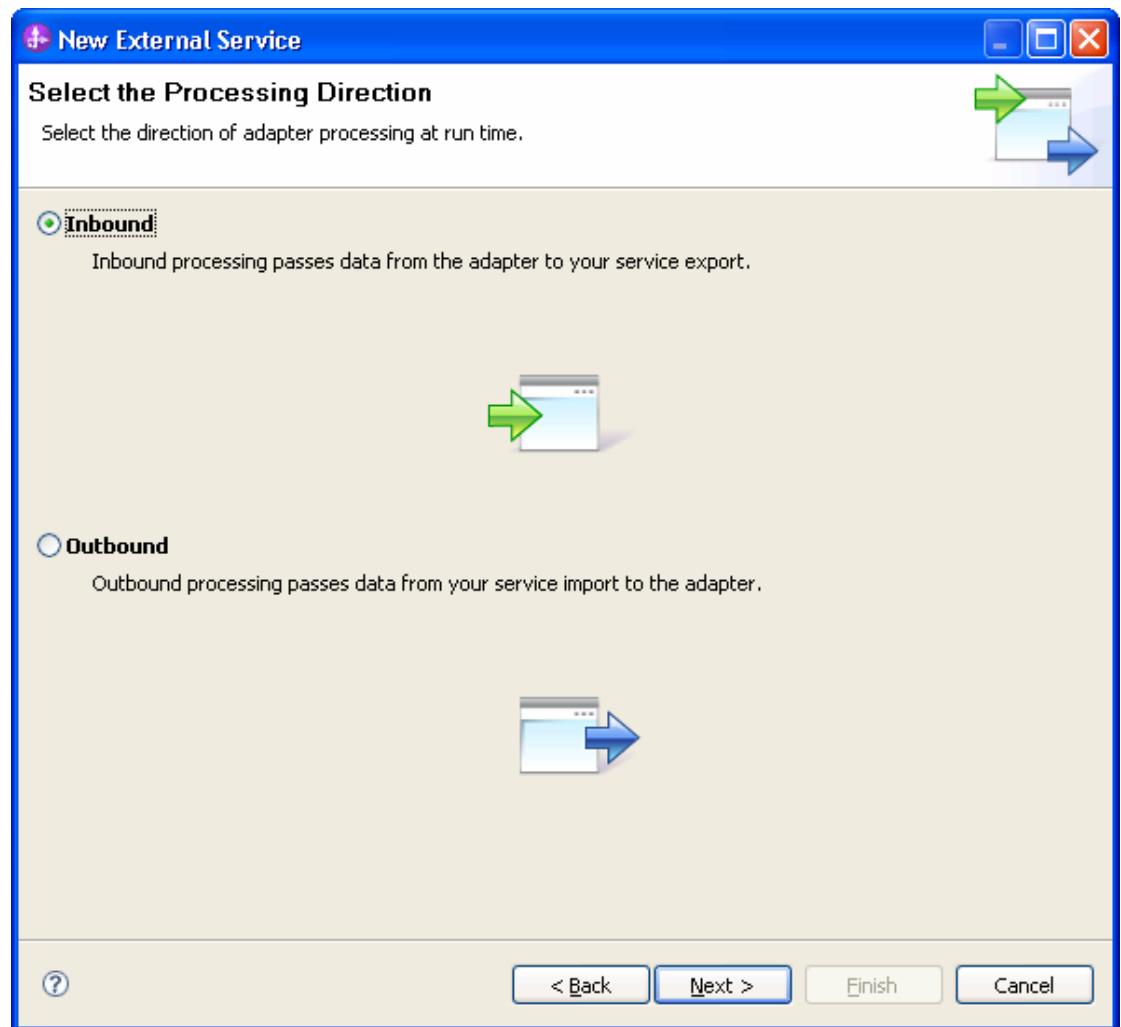


WebSphere software

7. In the **JDBC driver JAR files** field, click **Add** to add the JDBC driver class to connect to the database. Browse to select the driver JAR file and click **Next**.



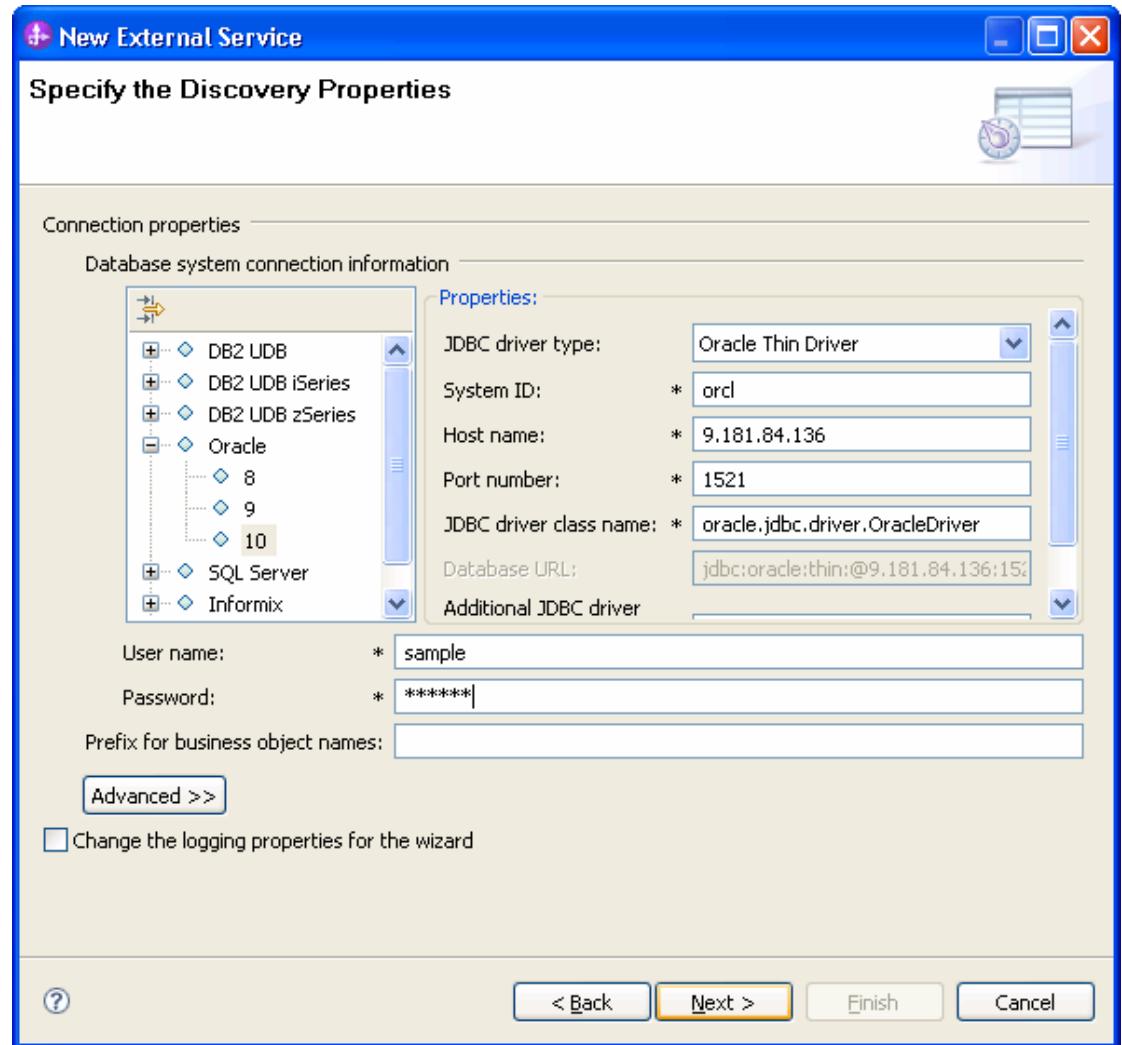
8. Select **Inbound** and click **Next**.



Set connection properties for the external service wizard

To connect to the Oracle database:

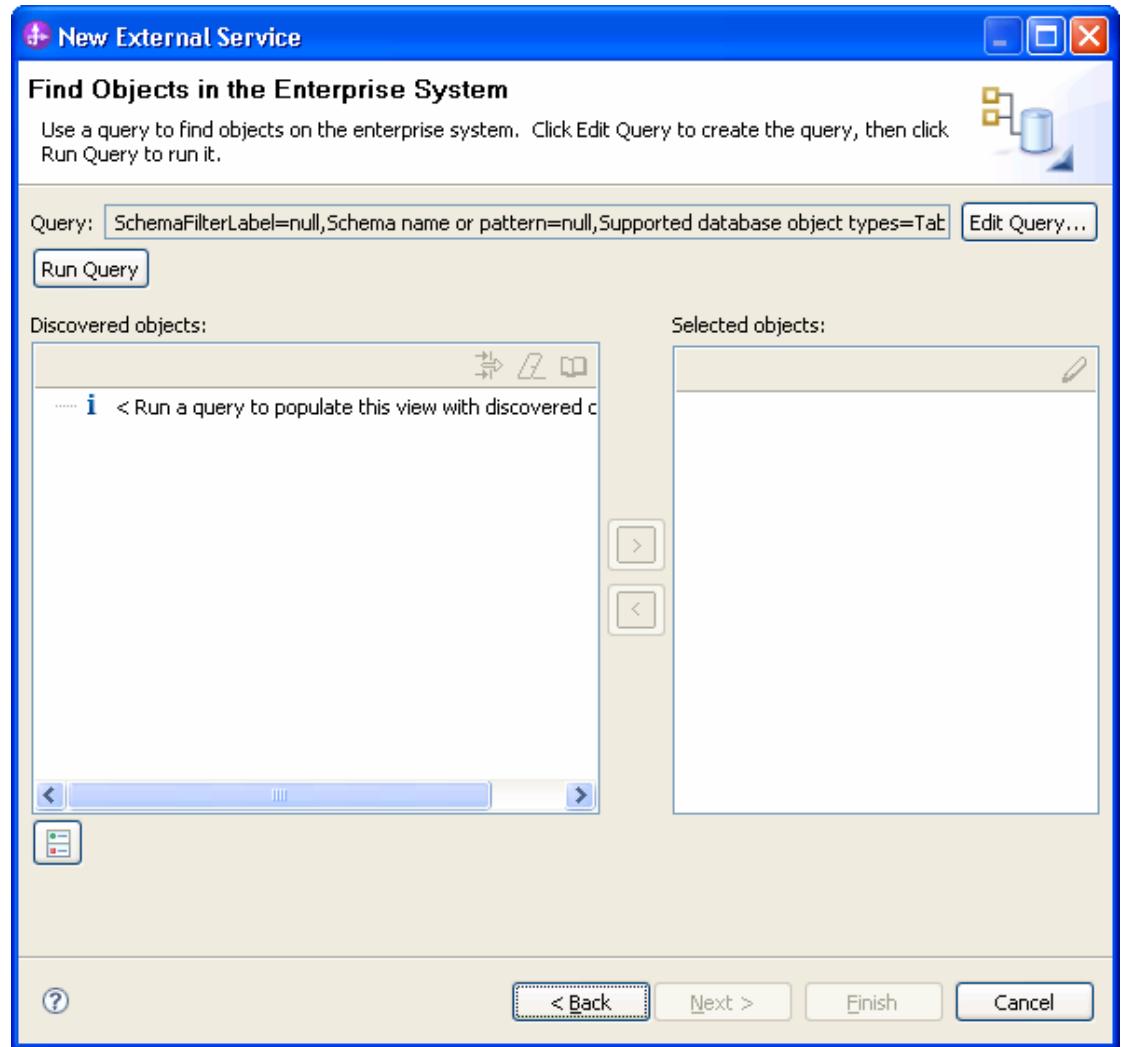
1. Expand the **Oracle** node in the **Database system connection information** area and then select **10**.
2. Enter values in the **System ID**, **Host name**, **Port number**, **User name** and **Password** fields, and click **Next**.



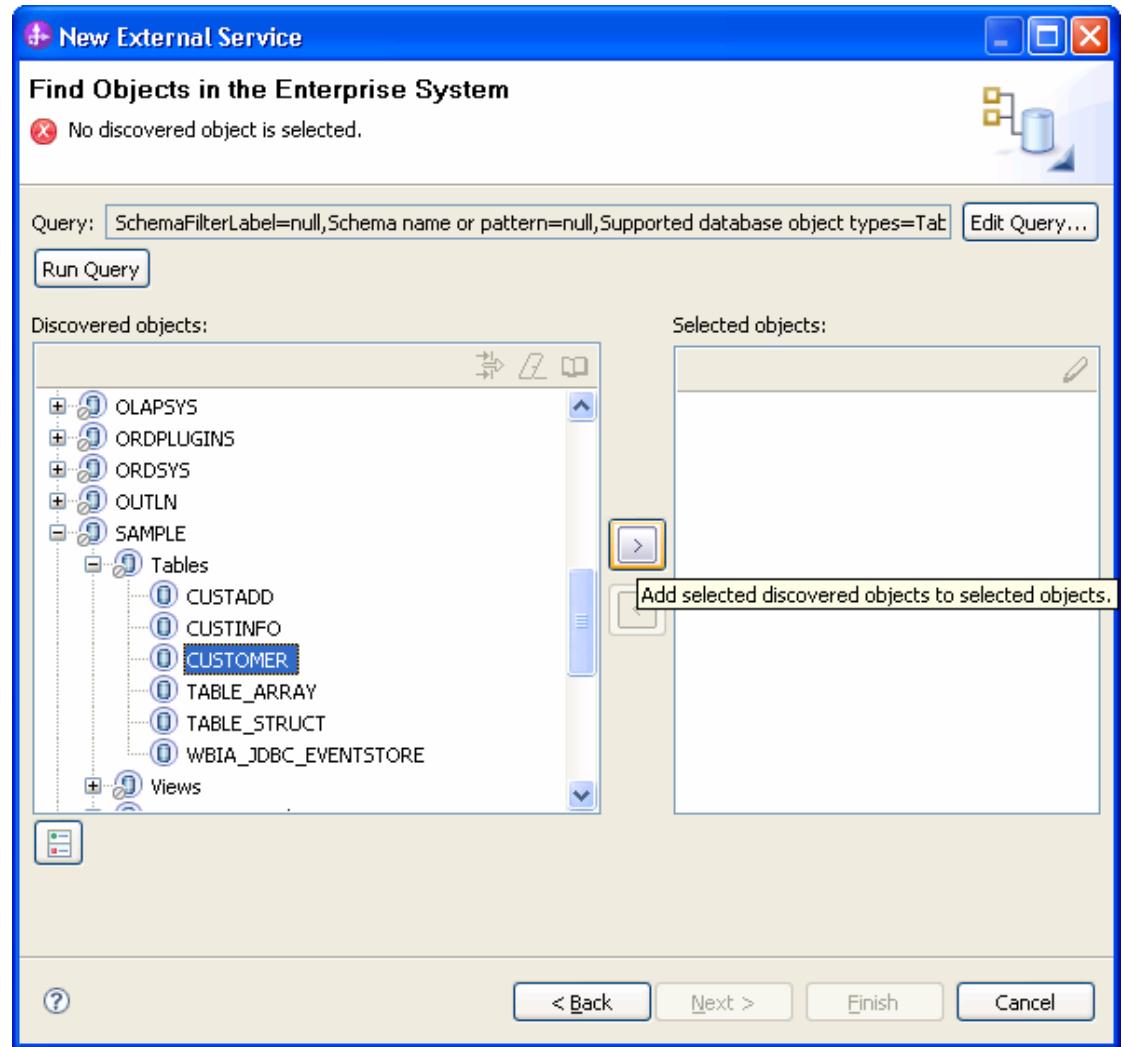
Select the business objects and services to be used with the adapter

Follow these steps to select the data for Inbound processing:

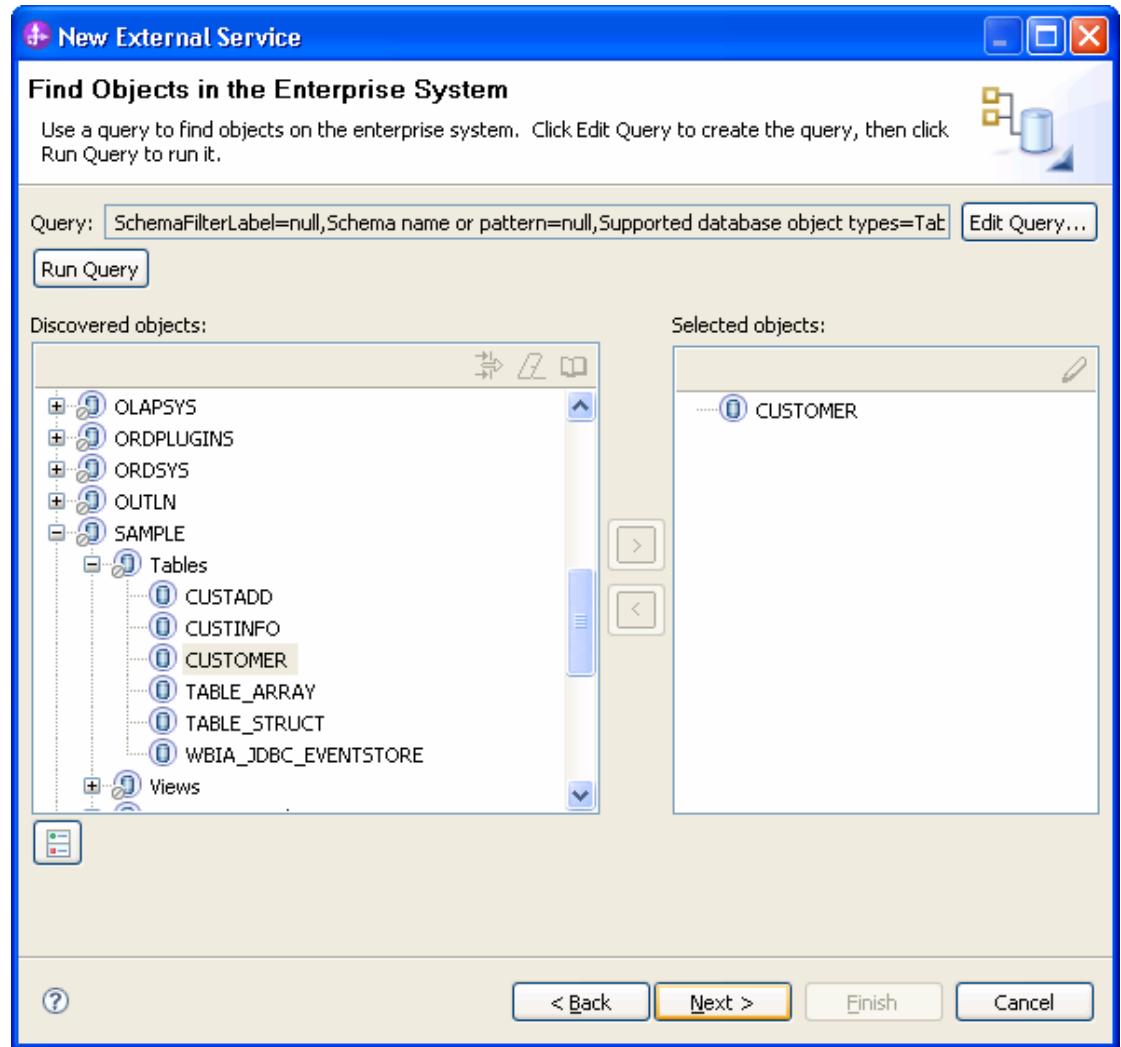
1. In the Find Objects in Enterprise System window, click **Run Query**.



2. Expand the **SAMPLE** (for this tutorial only) node, select **Tables** and expand it.



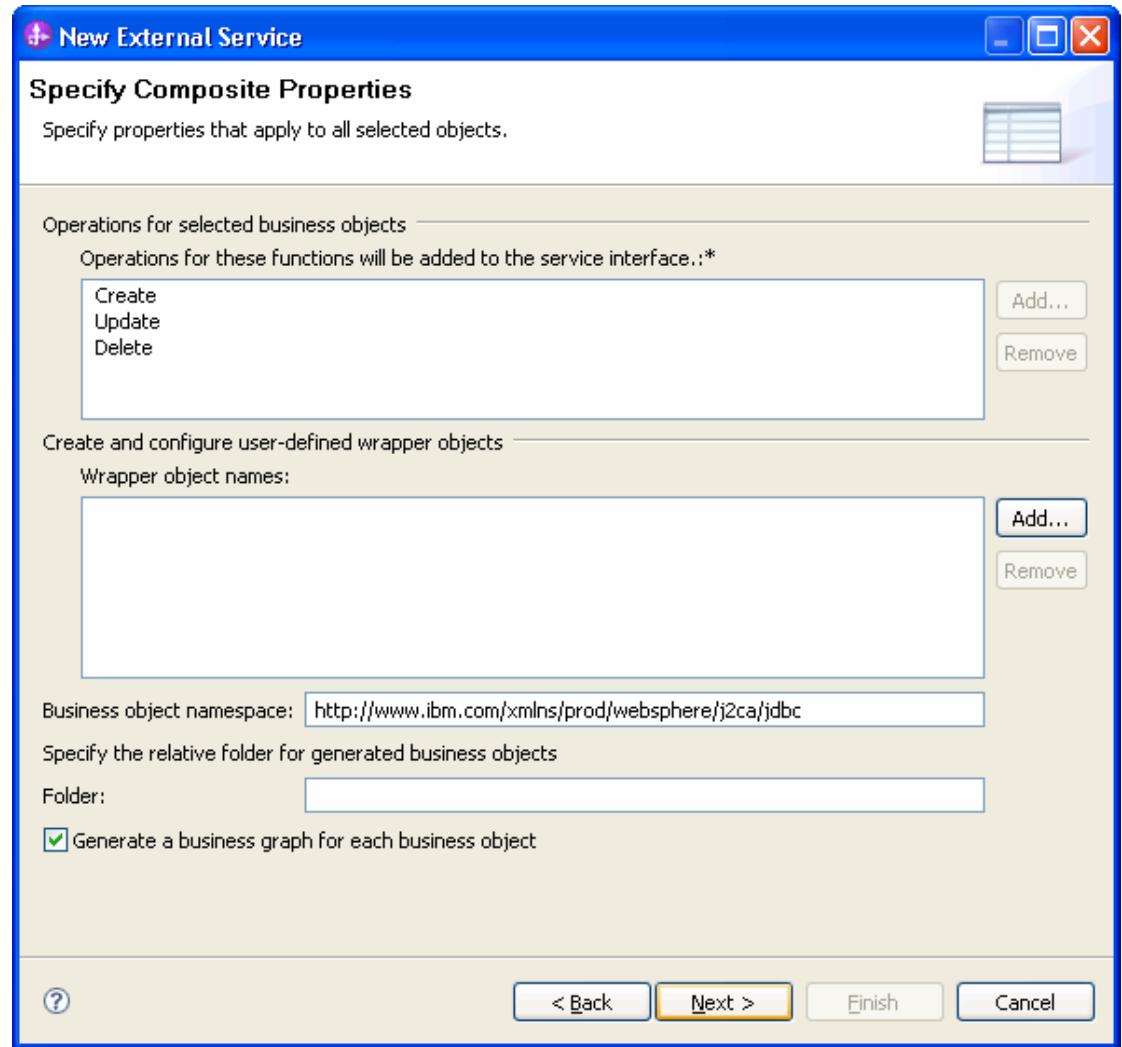
3. Select the CUSTOMER table and click .
4. Click **Next**.



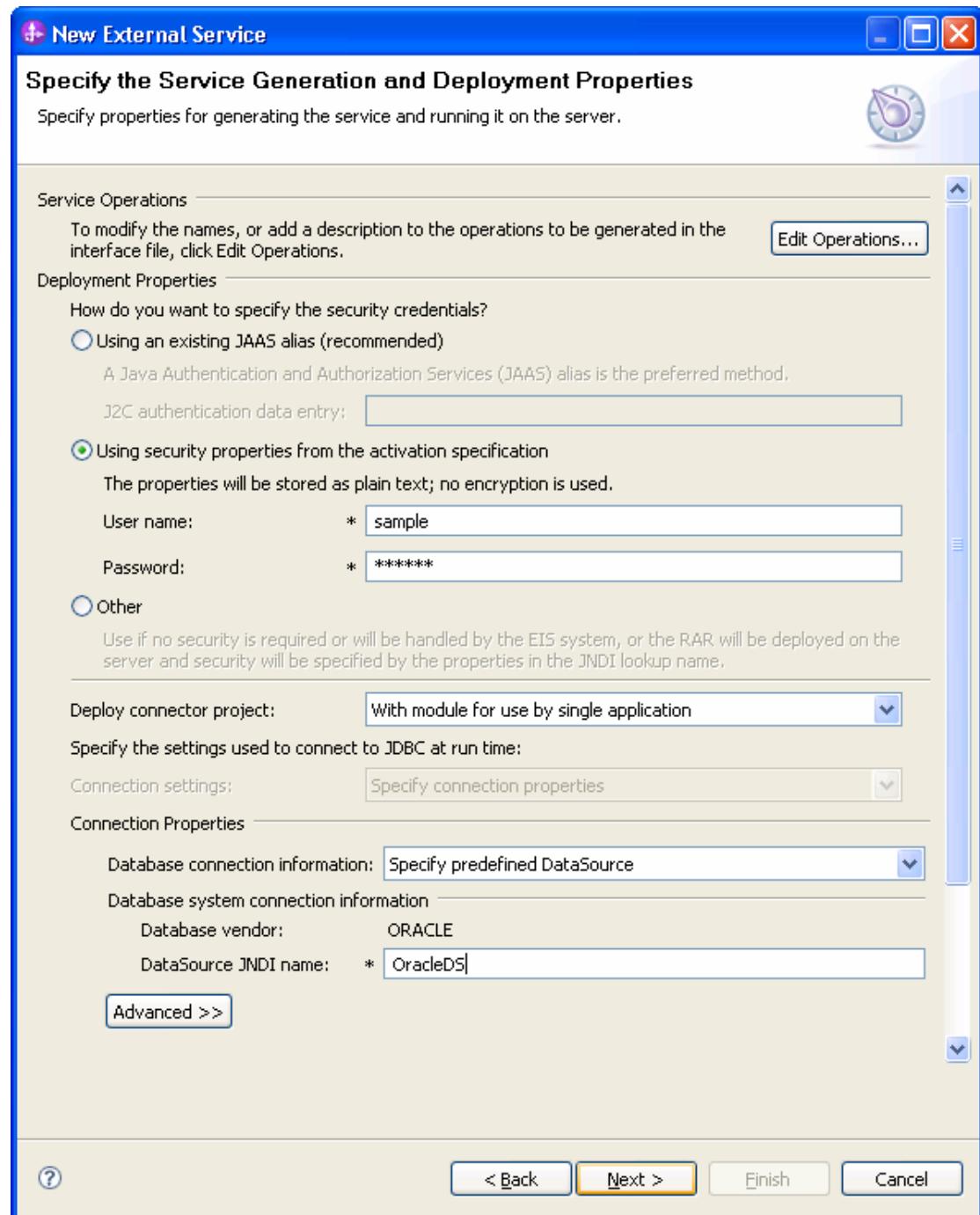
Generate business object definitions and related artifacts

Follow these steps to generate the business object definitions.

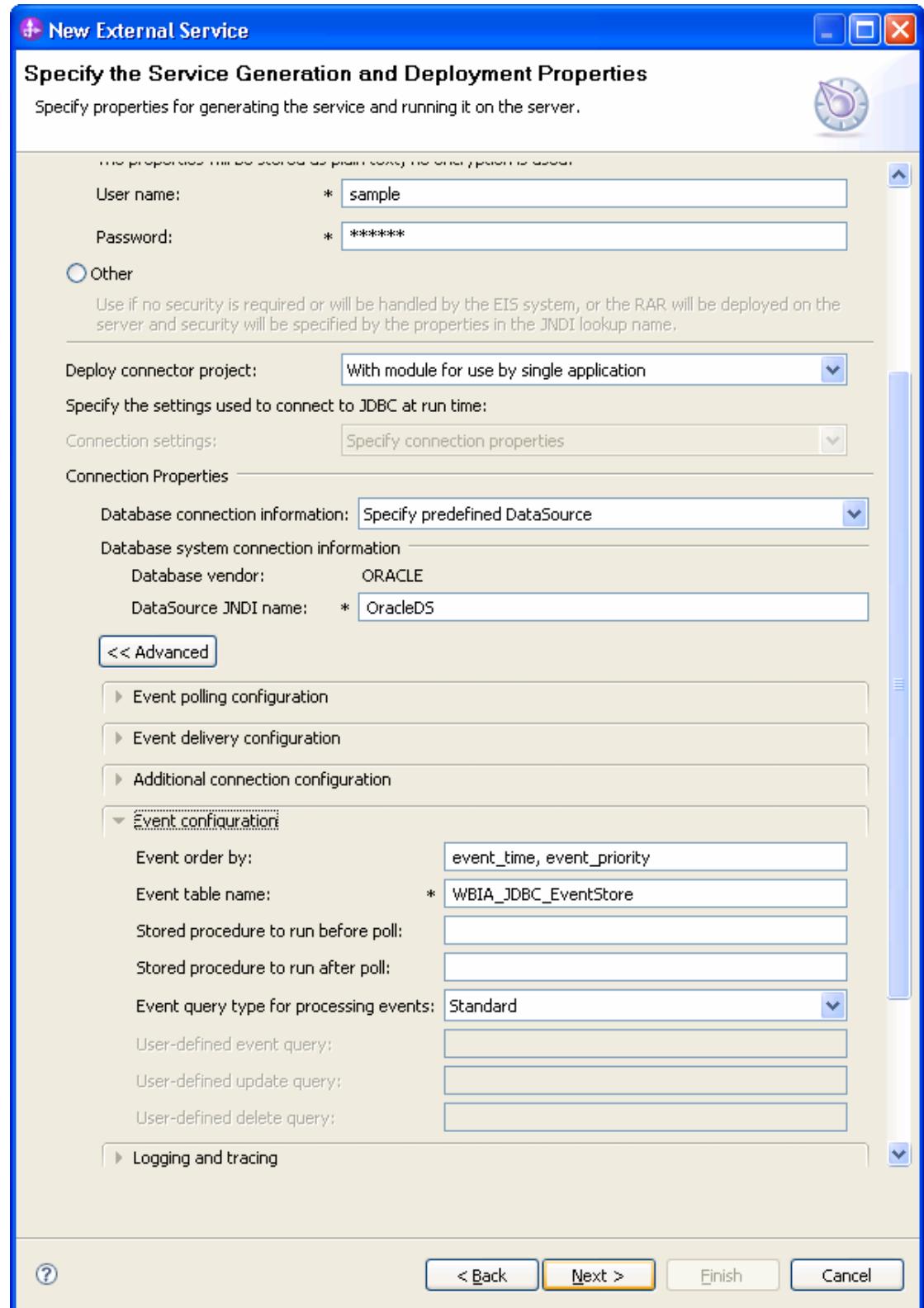
1. In the Specify Composite Properties window, accept the default values and click **Next**.



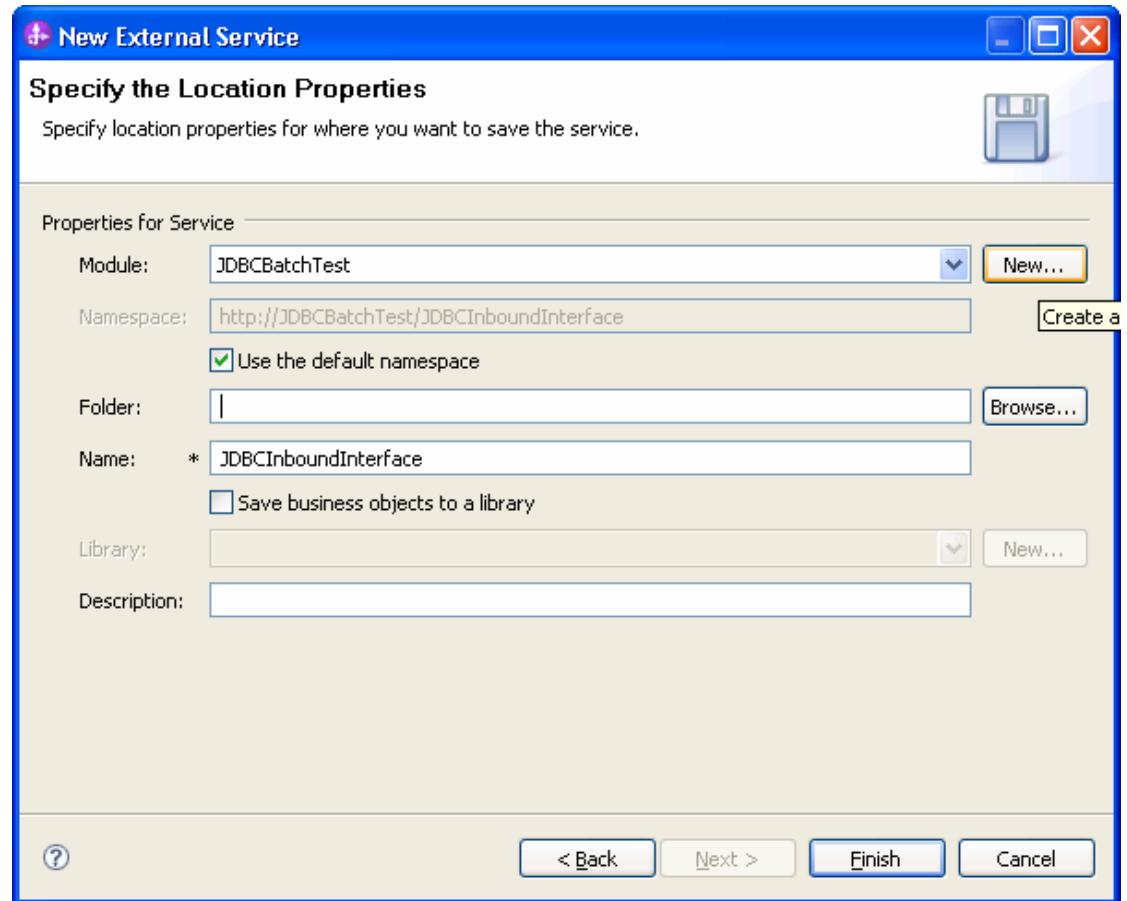
2. In the Specify the Service Generation and Deployment Properties window, perform the following steps:
 - a) Select **Using security properties from the activation specification** as the section option.
 - b) Select **Specify predefined DataSource** from the **Database connection information** list.
 - c) Click **Advanced**.



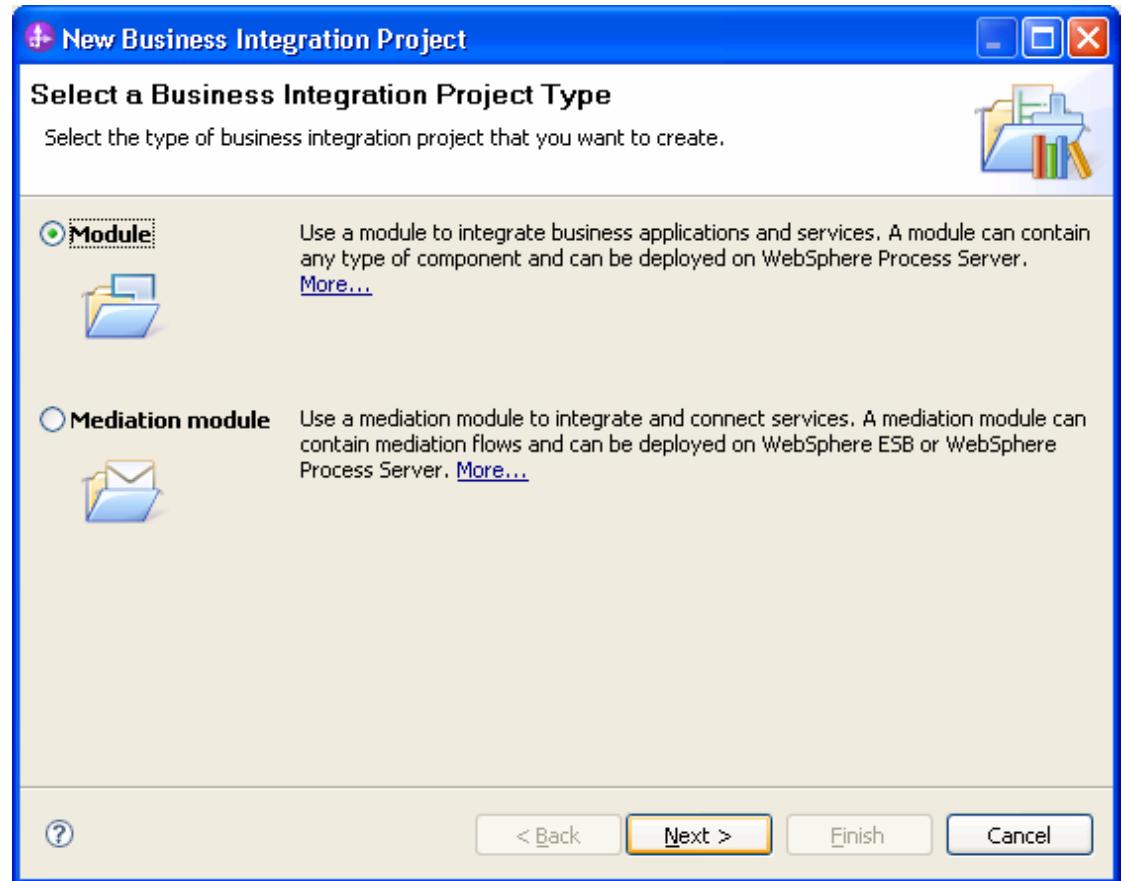
- d) In the Event Configuration area, enter the values for the **Event Order By**, **Event Table Name** fields and click **Next**.



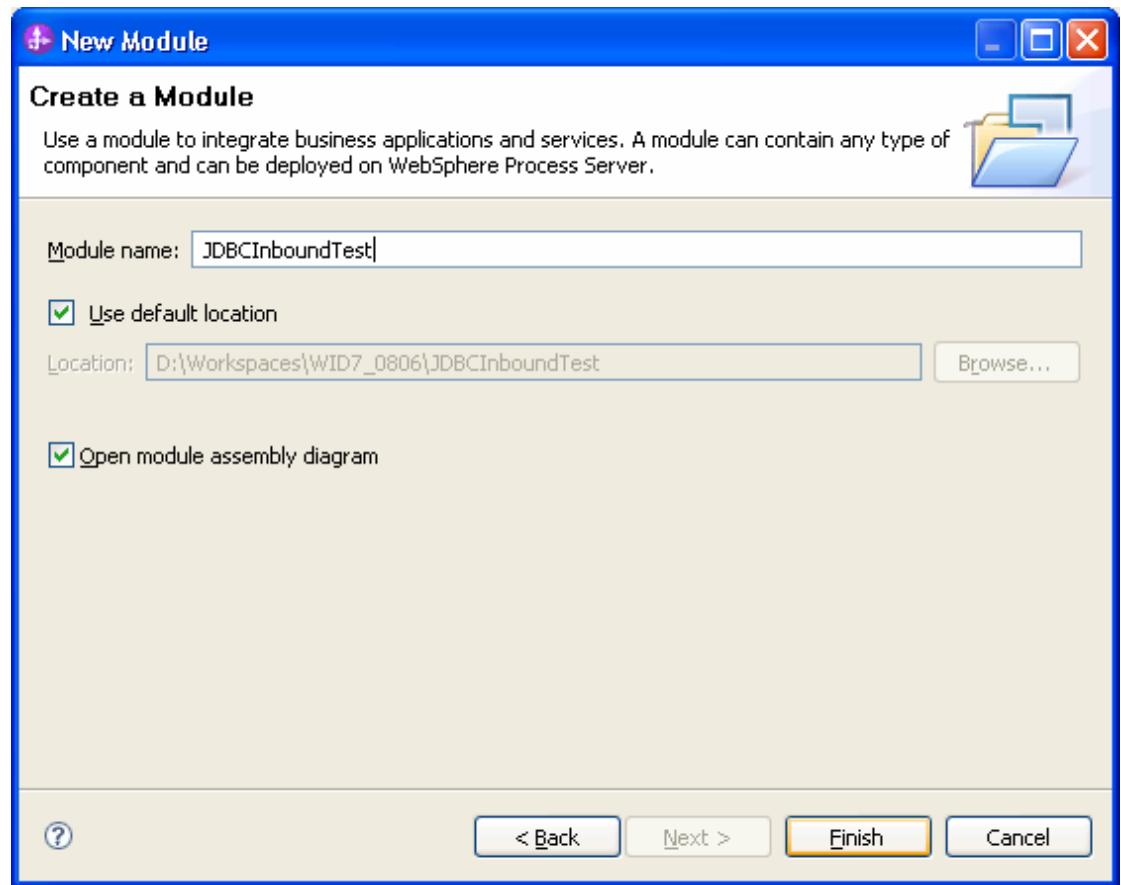
3. In the Specify the location Properties window, click **New**.



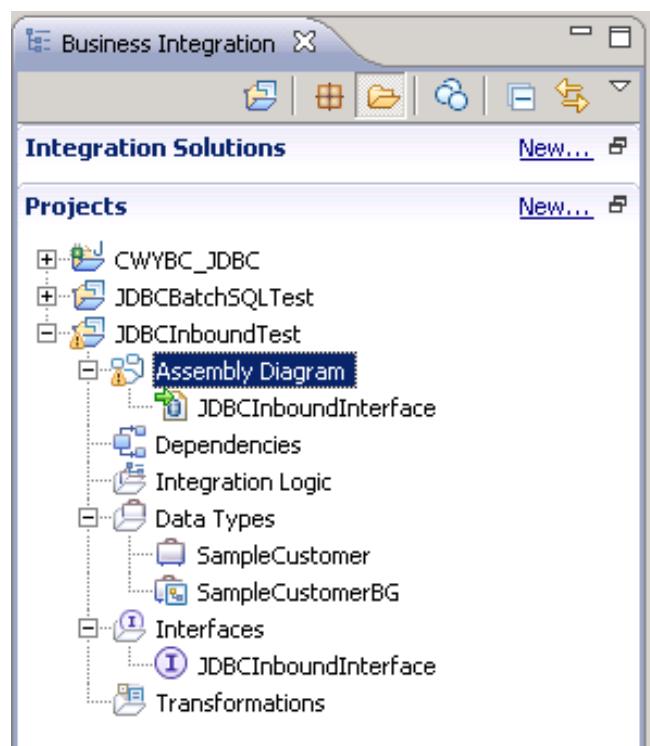
4. In the Select a Business Integration Project Type window, select **Module** and click **Next**.



5. In the Create a Module window, type **JDBCInboundTest** in the **Module Name** field and click **Finish**.



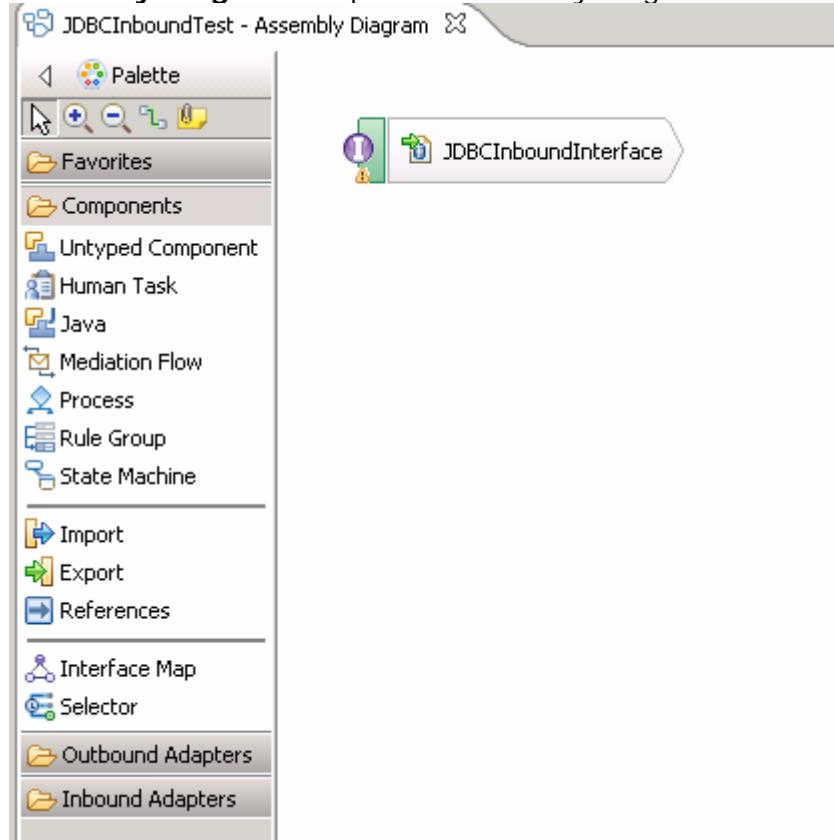
6. Click **Finish** to complete service creation.
7. Verify the results.



Set up the components to be part of the Inbound environment

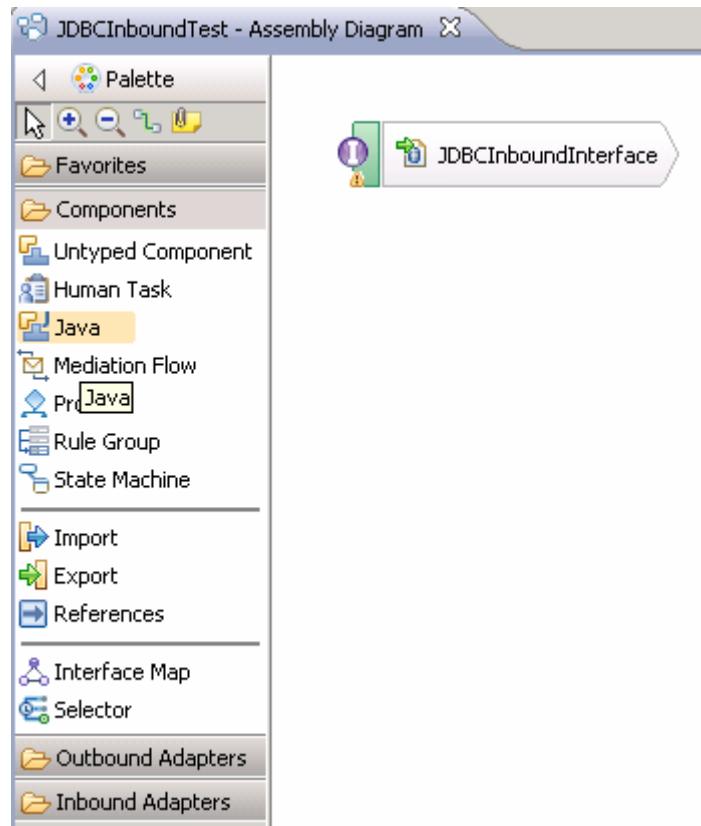
Add the components and set transaction specific properties for them so that they are part of the inbound environment.

1. In the Business Integration view, double click **JDBCInboundTest > Assembly Diagram** to open the Assembly Diagram.

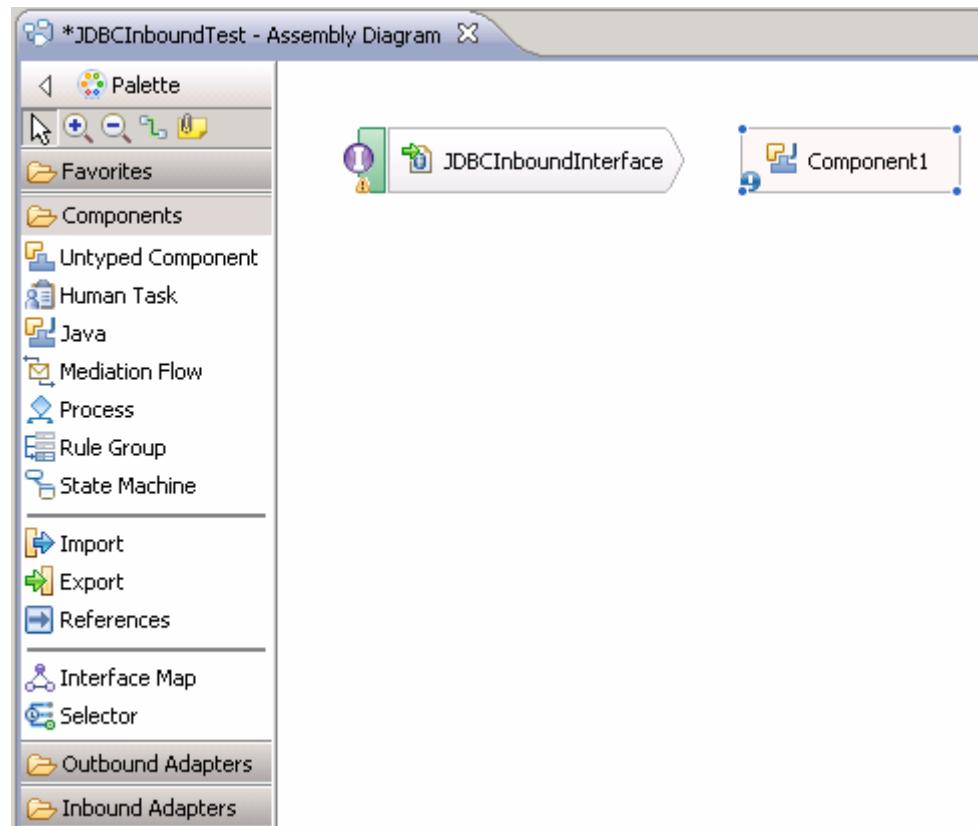


2. From the Palette, select the **Java** component and drop it on the assembly diagram.

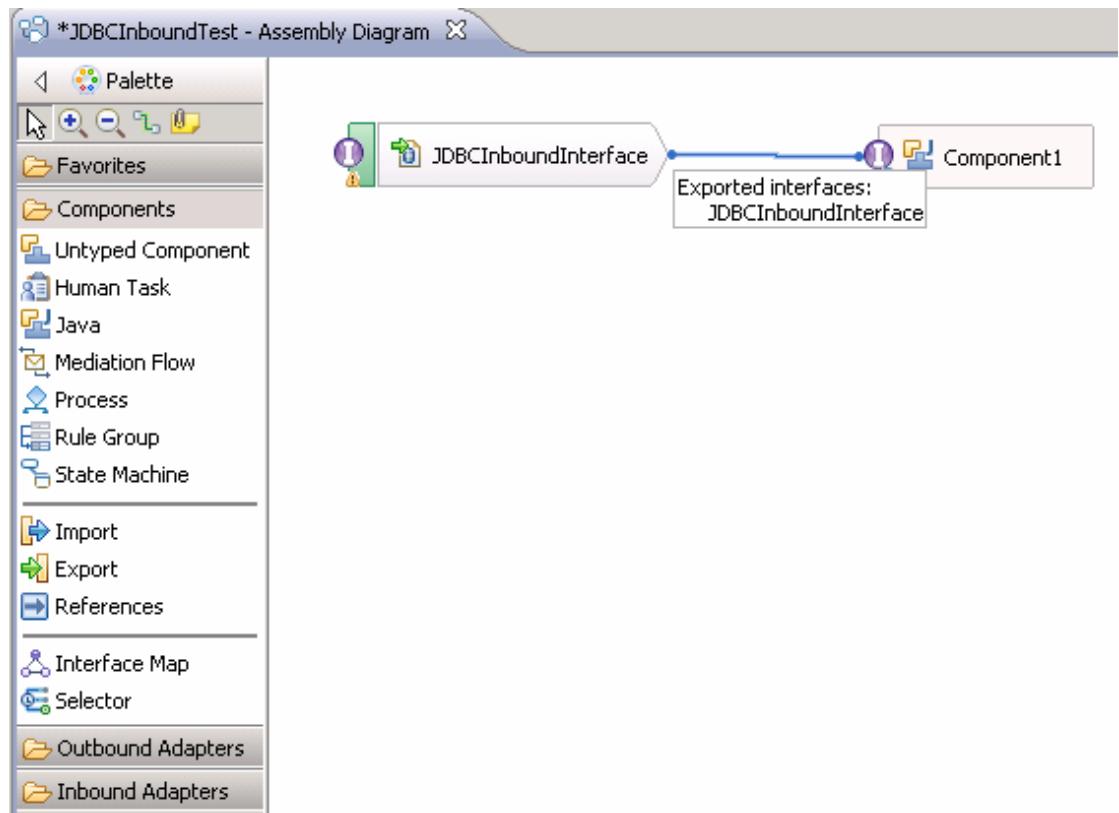
WebSphere software



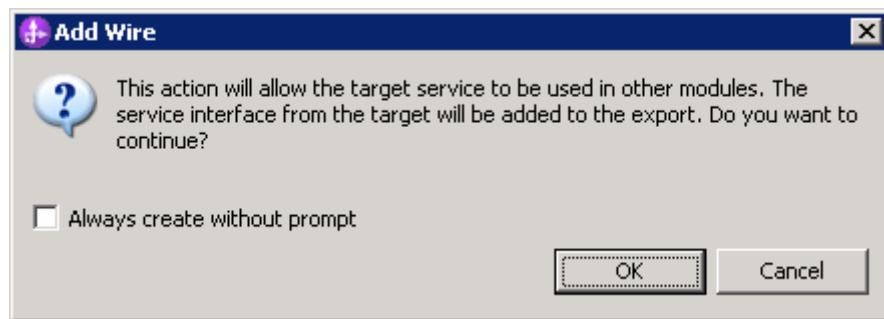
A component named **Component1** is created in the Assembly diagram.

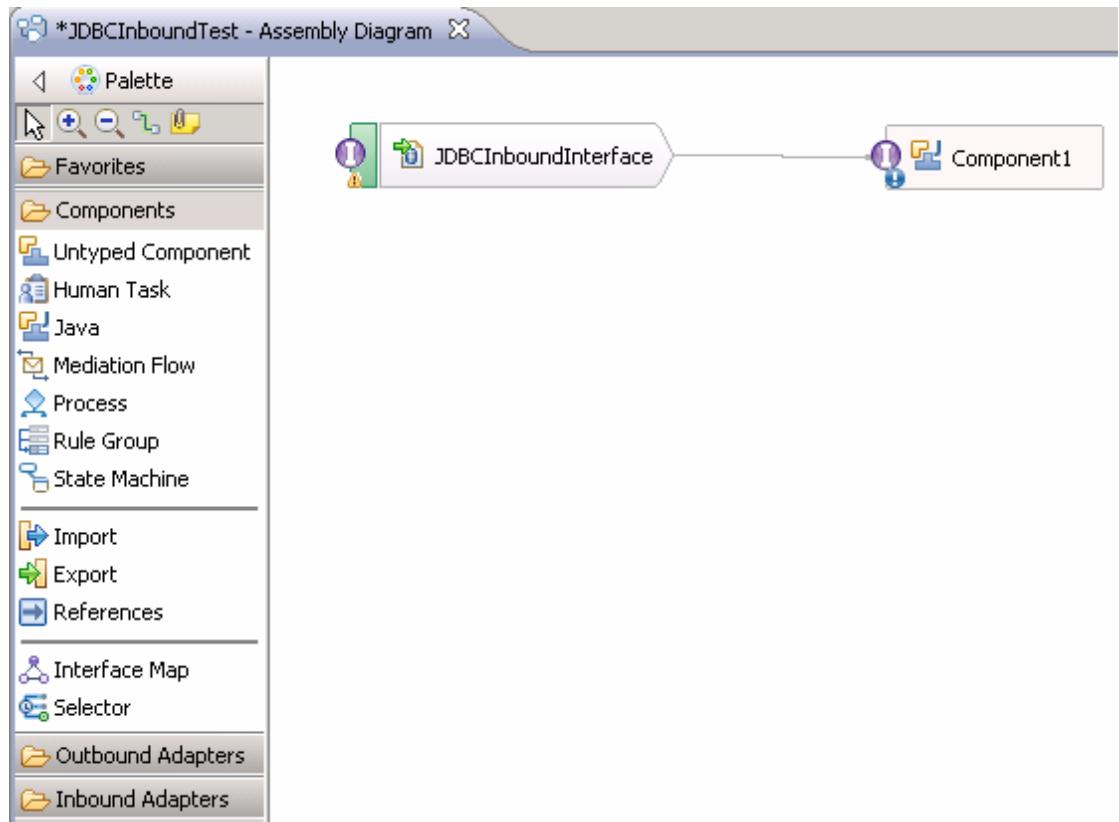


3. Wire **JDBCInboundInterface** to **Component1** by dragging the mouse pointer from the rear end of **JDBCInboundInterface** to the front end of **Component1**.

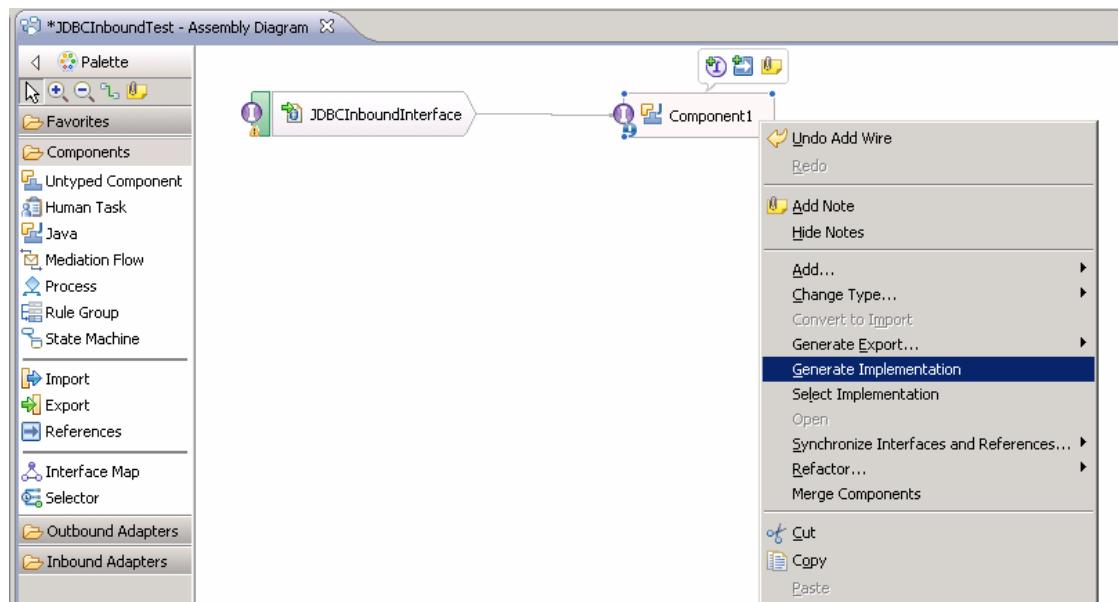


Note: Before the preceding window, i.e., before wiring you will see the following window. Click **OK**.





4. Generate the implementation for Java component. Right-click the component, and select **Generate Implementation** to complete the service creation.



5. Highlight the default package and select **OK**.

The Java Editor displays the Component1Impl.java file.

```

import commonj.sdo.DataObject;

public class ComponentImpl {
    /**
     * Default constructor.
     */
    public ComponentImpl() {
        super();
    }

    /**
     * Return a reference to the component service instance for this implementation
     * class. This method should be used when passing this service to a partner reference
     * or if you want to invoke this component service asynchronously.
     *
     * @generated (com.ibm.wbit.java)
     */
    @SuppressWarnings("unused")
    private Object getMyService() {
        return (Object) ServiceManager.INSTANCE.locateService("self");
    }

    /**
     * Method generated to support implementation of operation "createSampleCustomerBG" defined for WSDL port type
     * named "JDBCInboundInterface".
     *
     * The presence of commonj.sdo.DataObject as the return type and/or as a parameter
     * type conveys that it is a complex type. Please refer to the WSDL Definition for more information
     * on the type of input, output and fault(s).
     */
    public void createSampleCustomerBG(DataObject createSampleCustomerBGInput) {
}

```

6. Scroll down and locate the createSampleCustomer(DataObject createSampleCustomerBGInput) method that needs to be implemented. Write the code into the method so the complete method looks as follows:

```

    /**
     * Method generated to support implementation of operation "createSampleCustomerBG" defined for
     * named "JDBCInboundInterface".
     *
     * The presence of commonj.sdo.DataObject as the return type and/or as a parameter
     * type conveys that it is a complex type. Please refer to the WSDL Definition for more information
     * on the type of input, output and fault(s).
     */
    public void createSampleCustomerBG(DataObject createSampleCustomerBGInput) {
        // To get or set attributes for DataObject createSampleCustomerBGInput, use the APIs as shown
        // To set a string attribute in createSampleCustomerBGInput, use createSampleCustomerBGInput
        // To get a string attribute in createSampleCustomerBGInput, use createSampleCustomerBGInput
        // To set a dataObject attribute in createSampleCustomerBGInput, use createSampleCustomerBGInput
        // To get a dataObject attribute in createSampleCustomerBGInput, use createSampleCustomerBGInput
        System.out.println("Create customer");
        DataObject bg = createSampleCustomerBGInput;
        DataObject bo = bg.getDataObject("SampleCustomer");
        System.out.println("CUSTOMER KEY is: " + bo.getString("pkey"));
        System.out.println("CUSTOMER LAST NAME is: " + bo.getString("lname"));
        System.out.println("CUSTOMER FIRST NAME is: " + bo.getString("fname"));
        System.out.println("CUSTOMER CODE is: " + bo.getString("ccode"));
        System.out.println("CREATE end");
    }

```

7. Scroll down and locate the updateSampleCustomer(DataObject updateSampleCustomerBGInput) method that needs to be implemented. Write the code into the method so the complete method looks as follows:

```


    /**
     * Method generated to support implementation of operation "updateSampleCustomerBG" defined for
     * named "JDBCInboundInterface".
     *
     * The presence of commonj.sdo.DataObject as the return type and/or as a parameter
     * type conveys that it is a complex type. Please refer to the WSDL Definition for more information
     * on the type of input, output and fault(s).
     */
    public void updateSampleCustomerBG(DataObject updateSampleCustomerBGInput) {
        // To get or set attributes for DataObject updateSampleCustomerBGInput, use the APIs as shown
        // To set a string attribute in updateSampleCustomerBGInput, use updateSampleCustomerBGInput
        // To get a string attribute in updateSampleCustomerBGInput, use updateSampleCustomerBGInput
        // To set a dataObject attribute in updateSampleCustomerBGInput, use updateSampleCustomerBGInput
        // To get a dataObject attribute in updateSampleCustomerBGInput, use updateSampleCustomerBGInput
        System.out.println("UPDATE customer");
        DataObject bg = updateSampleCustomerBGInput;
        DataObject bo = bg.getDataObject("SampleCustomer");
        System.out.println("CUSTOMER KEY is: " + bo.getString("pkey"));
        System.out.println("CUSTOMER LAST NAME is: " + bo.getString("lname"));
        System.out.println("CUSTOMER FIRST NAME is: " + bo.getString("fname"));
        System.out.println("CUSTOMER CODE is: " + bo.getString("ccode"));
        System.out.println("UPDATE end");
    }
}


```

8. Scroll down and locate the deleteSampleCustomer(DataObject deleteSampleCustomerBGInput) method that needs to be implemented. Write the code into the method so the complete method looks as follows:

```


    /**
     * Method generated to support implementation of operation "deleteSampleCustomerBG" defined for
     * named "JDBCInboundInterface".
     *
     * The presence of commonj.sdo.DataObject as the return type and/or as a parameter
     * type conveys that it is a complex type. Please refer to the WSDL Definition for more information
     * on the type of input, output and fault(s).
     */
    public void deleteSampleCustomerBG(DataObject deleteSampleCustomerBGInput) {
        // To get or set attributes for DataObject deleteSampleCustomerBGInput, use the APIs as shown
        // To set a string attribute in deleteSampleCustomerBGInput, use deleteSampleCustomerBGInput
        // To get a string attribute in deleteSampleCustomerBGInput, use deleteSampleCustomerBGInput
        // To set a dataObject attribute in deleteSampleCustomerBGInput, use deleteSampleCustomerBGInput
        // To get a dataObject attribute in deleteSampleCustomerBGInput, use deleteSampleCustomerBGInput
        System.out.println("DELETE customer");
        DataObject bg = deleteSampleCustomerBGInput;
        DataObject bo = bg.getDataObject("SampleCustomer");
        System.out.println("CUSTOMER KEY is: " + bo.getString("pkey"));
        System.out.println("CUSTOMER LAST NAME is: " + bo.getString("lname"));
        System.out.println("CUSTOMER FIRST NAME is: " + bo.getString("fname"));
        System.out.println("CUSTOMER CODE is: " + bo.getString("ccode"));
        System.out.println("DELETE end");
    }
}


```

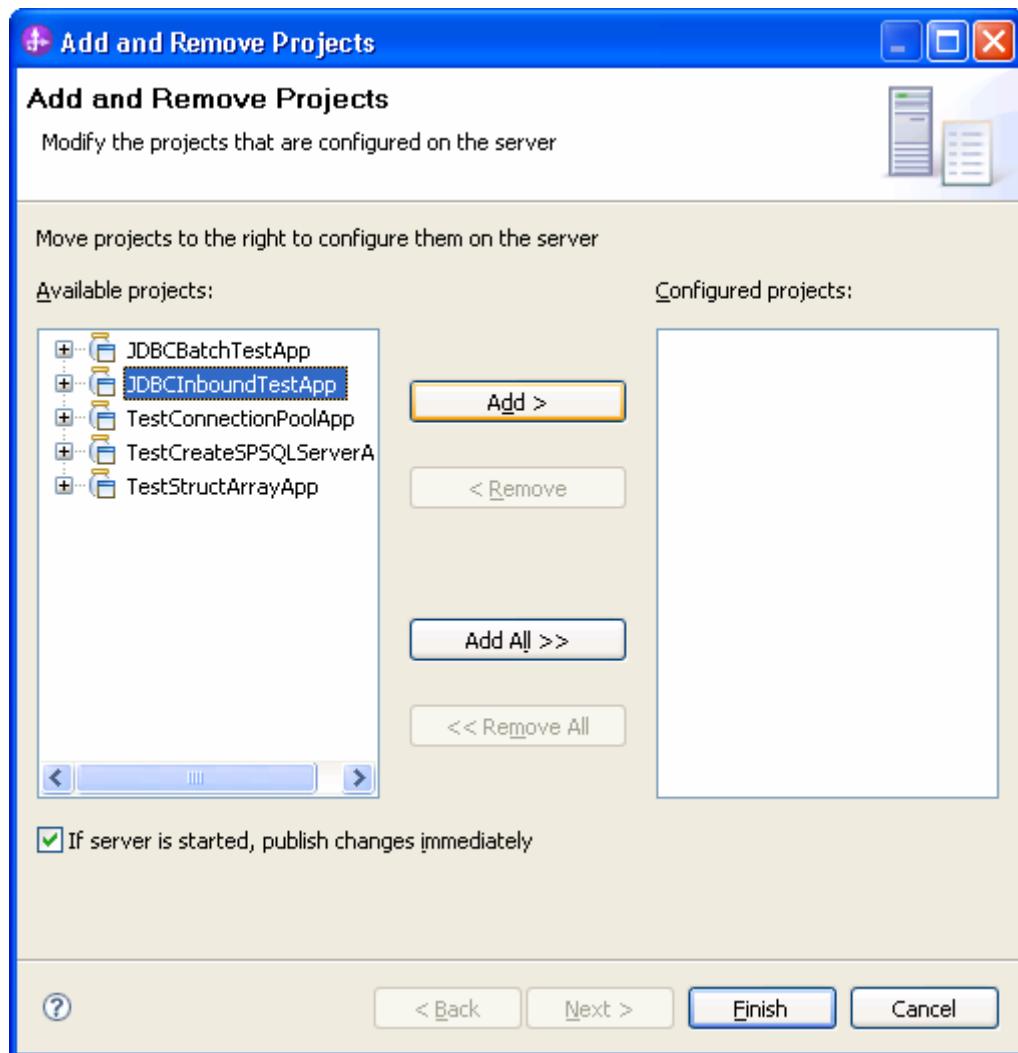
9. Select **File -> Save** to save your changes.
10. Close and save the Assembly Diagram. Wait for the workspace to complete building.

Deploy the module to the test environment

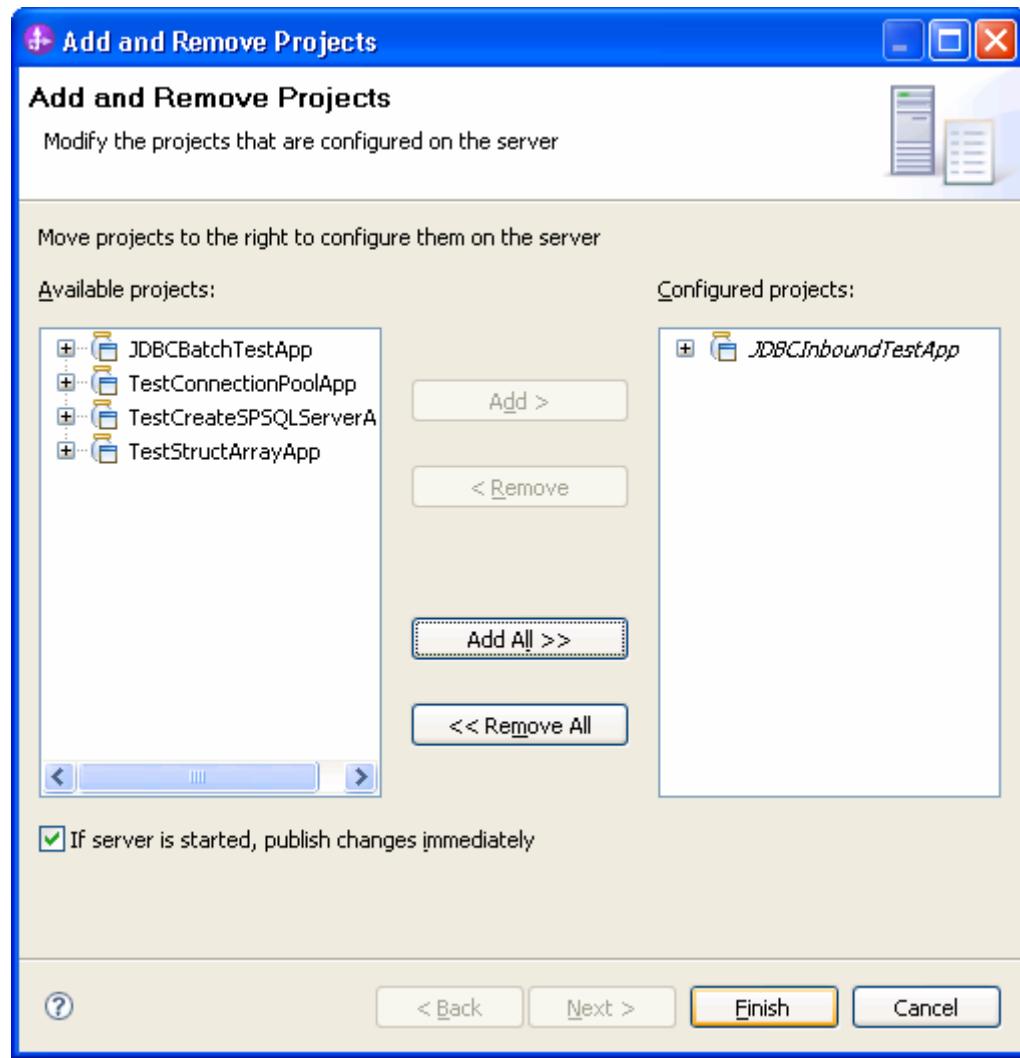
After running the external service wizard, you will have an SCA module that contains an Enterprise Information System (EIS) export. You must install this SCA module in the WebSphere Integration Developer integration test client. To do this, you must add the SCA module you created earlier to the server using the **Servers** view in WebSphere Integration Developer.

Steps for adding the SCA module to the server:

1. In WebSphere Integration Developer, switch to the **Servers** view by selecting **Windows > Show View > Servers**.
2. In the Servers tab in the lower-right pane of the WebSphere Integration Developer screen, right-click the server, and select **Start**.
3. After the server is started, right-click the server, and select **Add and Remove projects**.



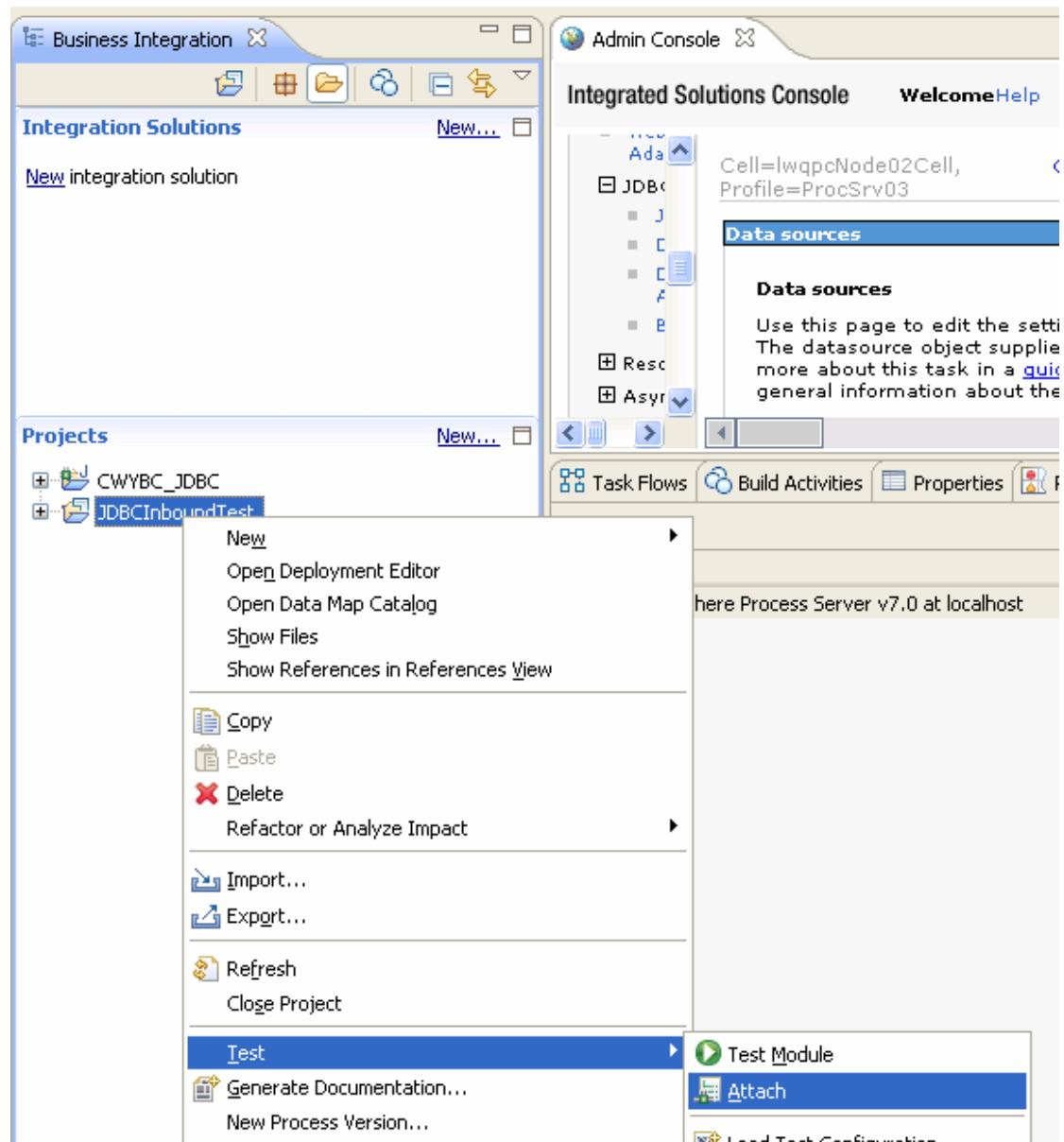
4. Add the SCA module to the server.
5. Click **Finish**.

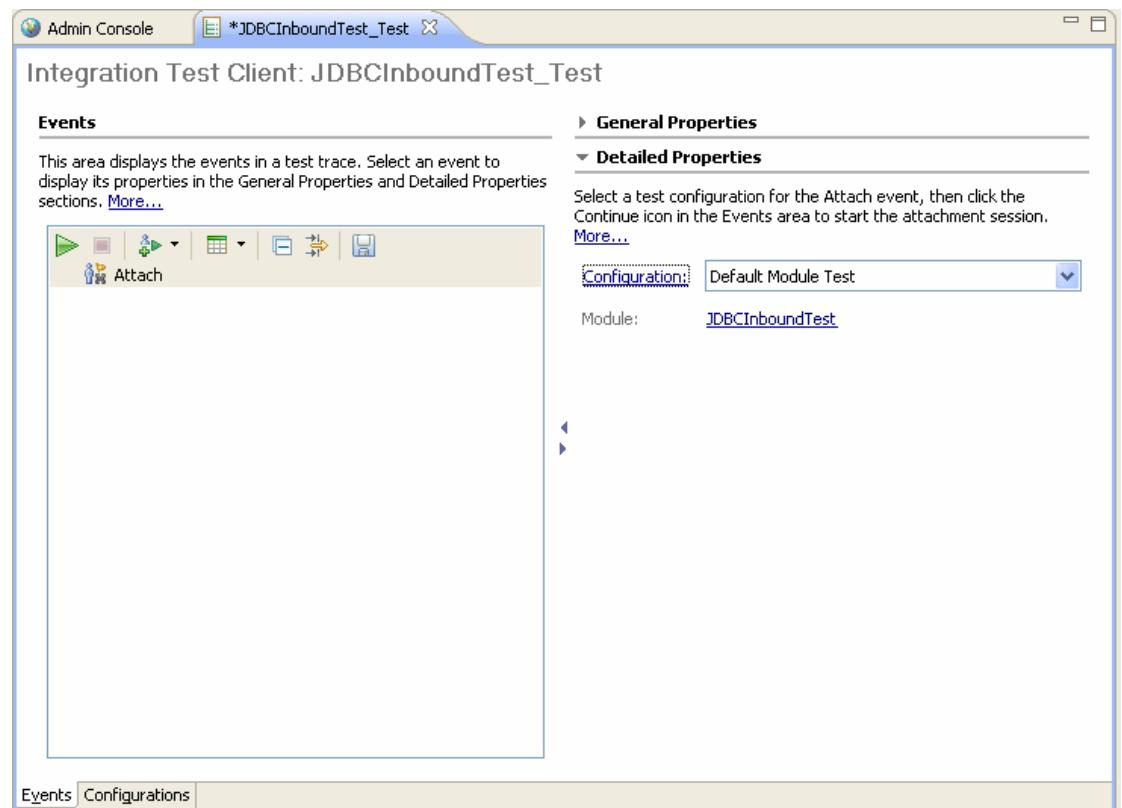


Test the assembled adapter application

Test the assembled adapter application using the WebSphere Integration Developer integration test client.

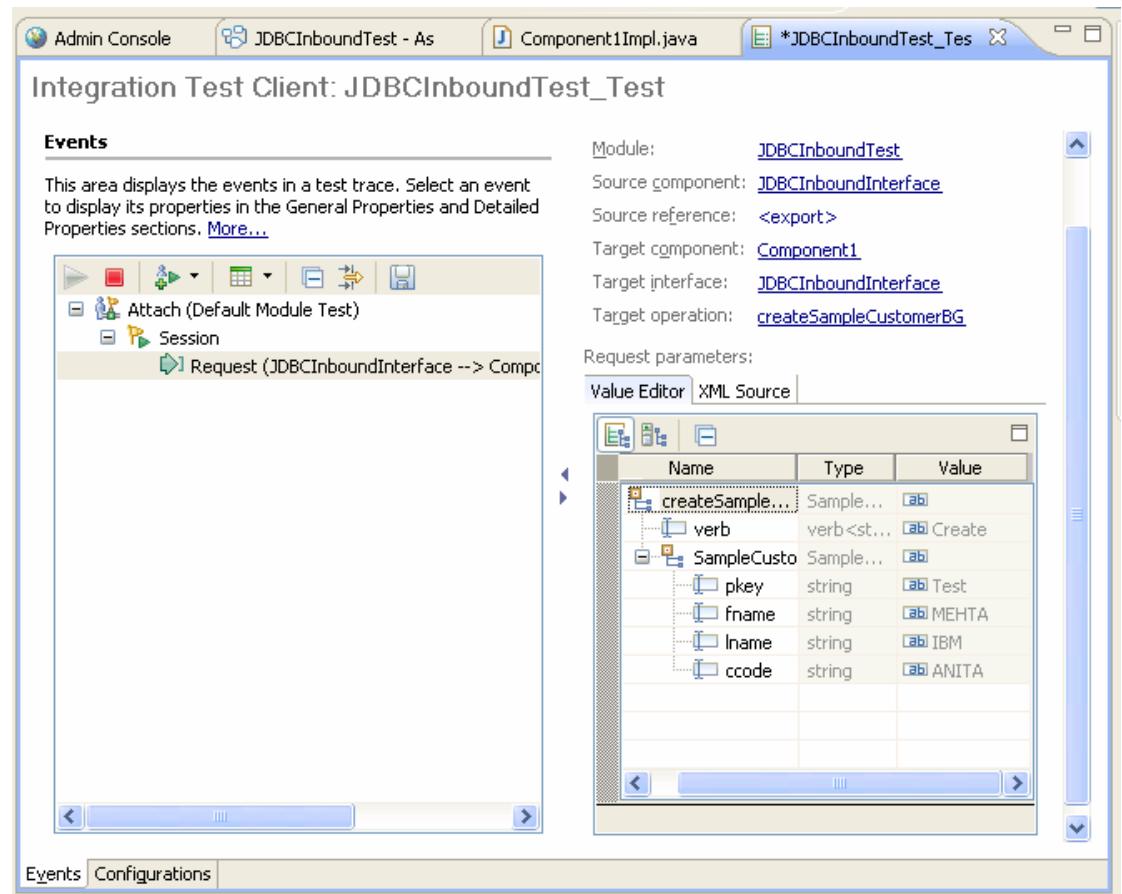
1. In the Business Integration view right-click on the JDBCInboundTest module, and select Test > Attach.





2. To execute the service, click .
3. Insert a record into the Customer table:

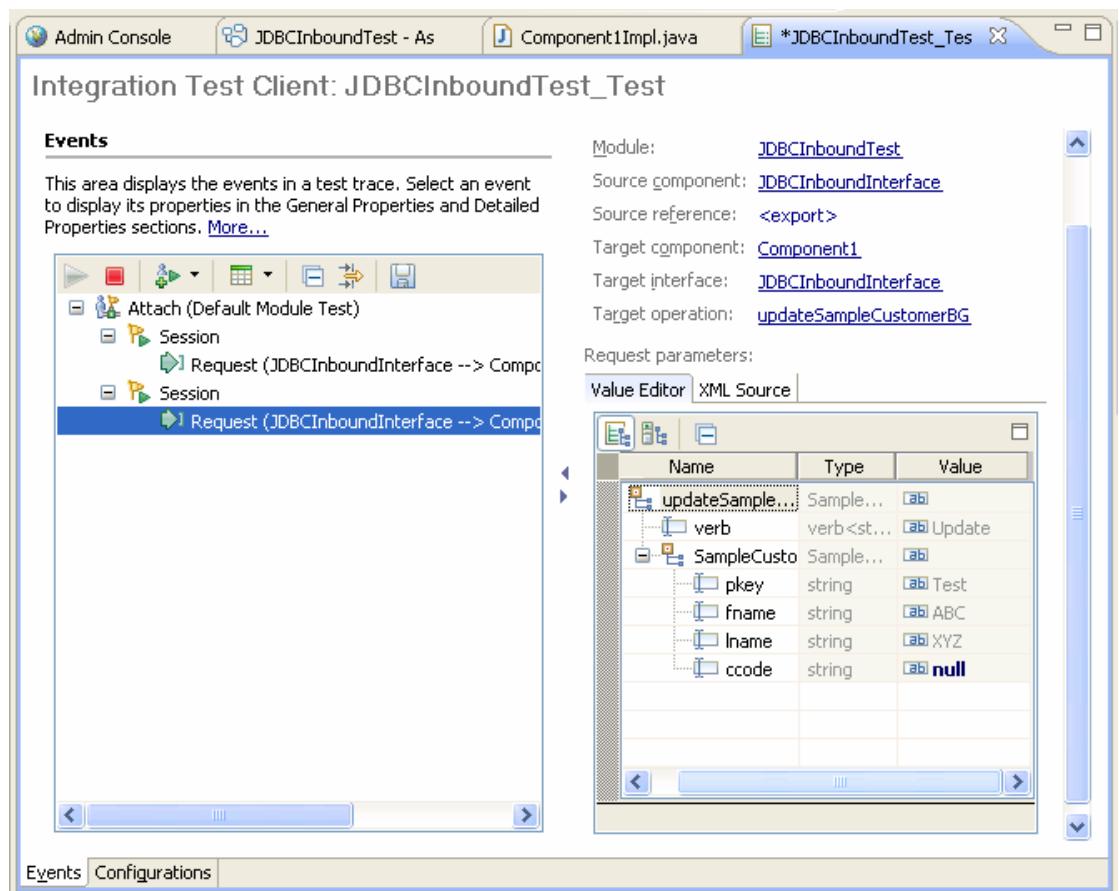
```
INSERT INTO CUSTOMER (pkey,ccode,fname, lname)
values('Test', 'ANITA', 'MEHTA', 'IBM');
```
4. Check the output of the service:



5. Update an existing record in the Customer table:

```
UPDATE CUSTOMER SET fname='ABC', lname='XYZ',  
ccode='' WHERE pkey='Test';
```

6. Check the output of the service:



Clear the sample content

After you have tested the application, clear the sample content to return the data to its original state.

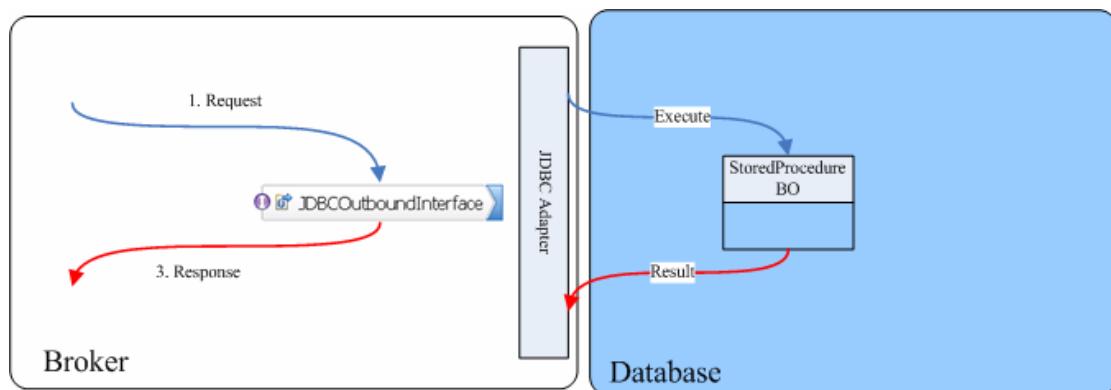
Chapter 7. Tutorial 6: Executing a business object created from a stored procedure (DB2)

This scenario demonstrates how WebSphere Adapter for JDBC 7.0.0.0 interacts with database stored procedure business object.

About this task

In this scenario, a SCA component invokes the 'Execute' operation of JDBC adapter Outbound Interface. The adapter invokes a stored procedure defined in the target database, and returns the execution result to the SCA component.

The following figure represents this scenario:



Prepare to run through the tutorial

Replicas of the artifacts that you create when using the external service wizard are provided as sample files for your reference. Use these files to verify if the files you create using the external service wizard are correct.

Download the sample zip file and extract it into a directory of your choice (you may want to create a new directory).

Configuration prerequisites

Before configuring the adapter, you must complete the following tasks:

- Create tables and stored procedure
- Create an authentication alias

Create tables and stored procedure

You must create the following tables and stored procedure in the DB2 database before starting the scenario.

a. Script for creating the Customer and Address tables

```

CREATE TABLE CUSTOMER (
    "PKEY" VARCHAR(10) NOT NULL PRIMARY KEY,
    "FNAME" VARCHAR(20) ,
    "LNAME" VARCHAR(20) ,
    "CCODE" VARCHAR(10) ) ;

CREATE TABLE ADDRESS (
    "ADDRID" VARCHAR(10) NOT NULL PRIMARY KEY,
    "CUSTID" VARCHAR(10) ,
    "CITY" VARCHAR(20) ,
    "ZIPCODE" VARCHAR(10) ) ;

```

b. Scripts for inserting records to the two tables

```

INSERT INTO CUSTOMER VALUES ('100', 'fname1',
'lname1', 'IBM');
INSERT INTO CUSTOMER VALUES ('300', 'abc', 'xyz',
'IBM');
INSERT INTO ADDRESS VALUES ('100', '100', 'cxxx',
'xxxx');
INSERT INTO ADDRESS VALUES ('120', '100', 'city1',
'zipcode1');

```

c. Scripts for creating the stored procedure

The stored procedure can be created using the DB2 Development Center or WebSphere Integration Developer.

```

CREATE PROCEDURE CustAddrSP ( )
    SPECIFIC CustAddrSP
    DYNAMIC RESULT SETS 1
-----
-----  

-- SQL Stored Procedure  

-----  

-----  

P1: BEGIN
    -- Declare cursor
    DECLARE cursor1 CURSOR WITH RETURN FOR
        SELECT CUSTOMER.FNAME, CUSTOMER.LNAME,
ADDRESS.CITY, ADDRESS.ZIPCODE
        FROM ADDRESS JOIN CUSTOMER ON ADDRESS.CUSTID
= CUSTOMER.PKEY
        ORDER BY ADDRESS.ZIPCODE ASC;

    -- Cursor left open for client application
    OPEN cursor1;
END P1

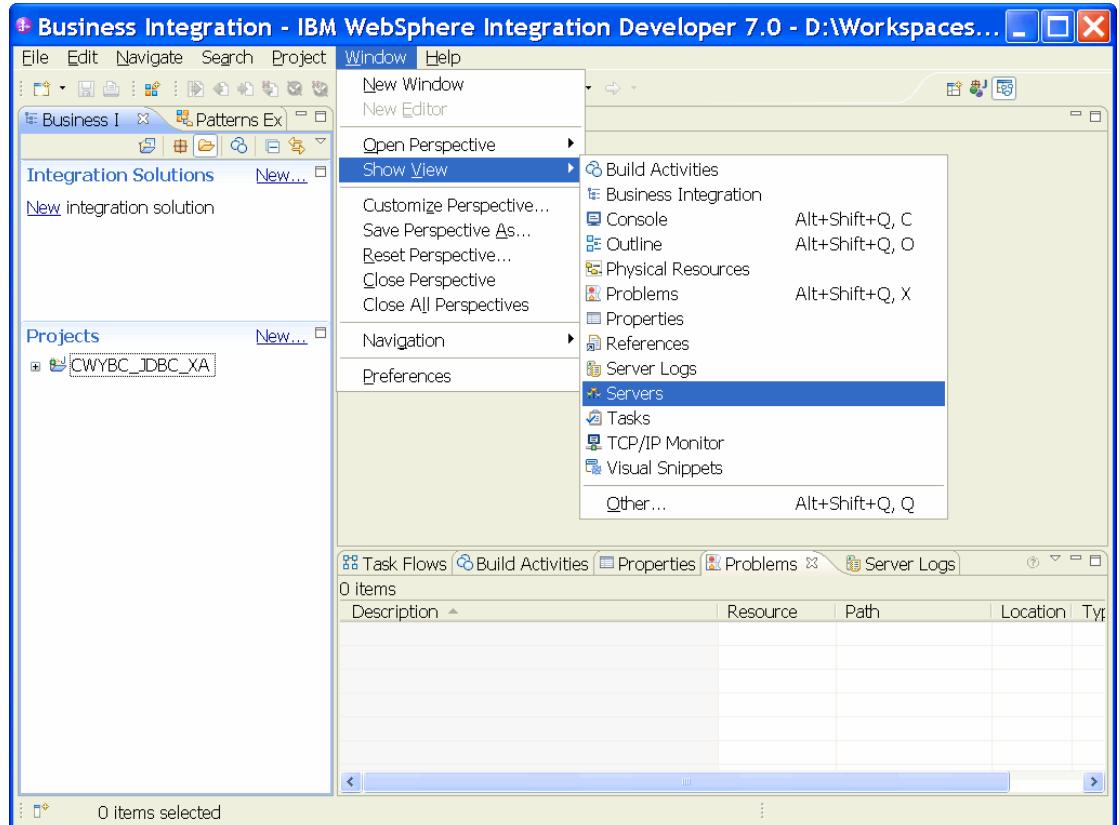
```

Create an authentication alias

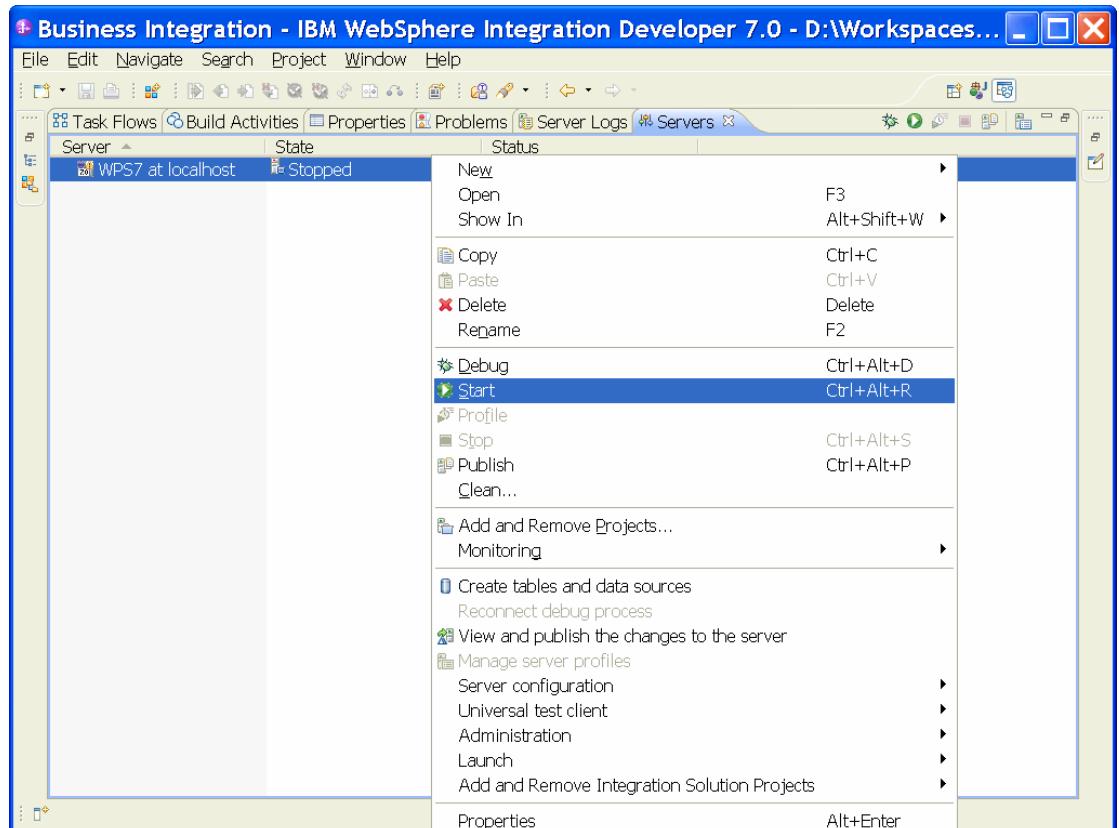
The authentication alias needs to be set because the adapter uses the username and password set in the authentication alias to connect to the database. This authentication alias will be used later when generating the artifacts for the module.

Here are the steps to set the authentication alias in WebSphere Process Server administration console.

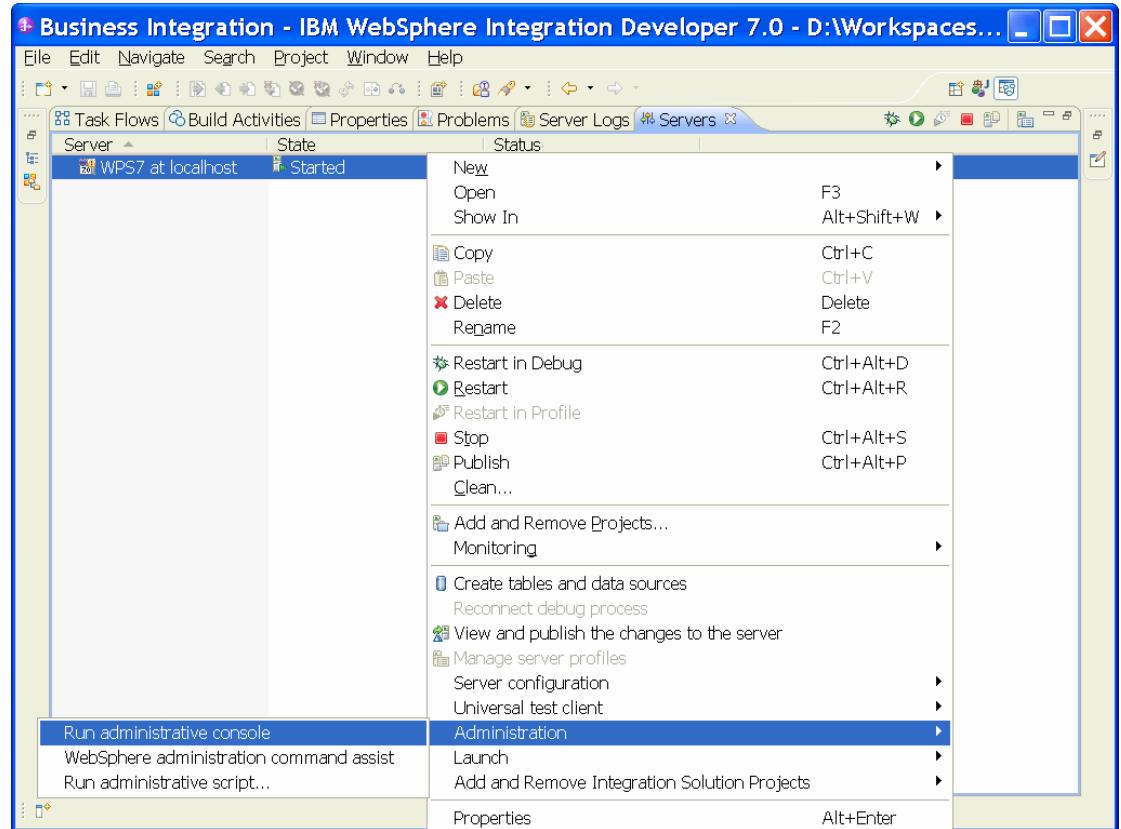
1. In WebSphere Integration Developer, switch to the **Servers** view by selecting **Windows > Show View > Servers**.



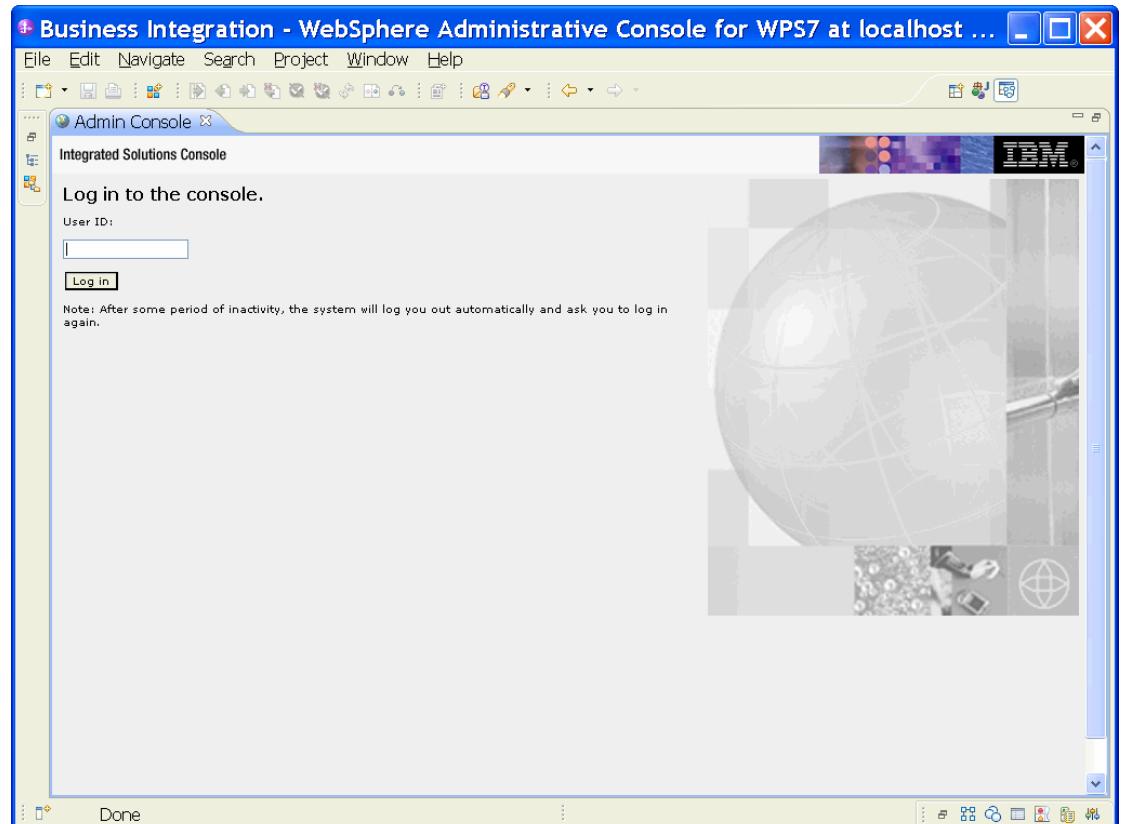
2. In the **Servers** view, right-click the server that you want to start and select **Start**.



3. After the server is started, right-click the server, and select **Administration > Run administrative console**.



4. Log on to the administrative console.



5. Click **Security → Global security**.



- Under **Java Authentication and Authorization Service**, click **J2C authentication data**.

Cell=localhostNode01Cell, Profile=AppSrv01 Close page

Global security

Global security
Use this panel to configure administration and the default application security policy. This security configuration applies to functions and is used as a default security policy for user applications. Security domains can be defined to override and control applications.

Security Configuration Wizard	Security Configuration Report
<div style="border-bottom: 1px solid #ccc; margin-bottom: 5px;"> Administrative security </div> <div style="display: flex; justify-content: space-between;"> <div style="flex: 1;"> <input type="checkbox"/> Enable administrative security <ul style="list-style-type: none"> Administrative user roles Administrative group roles Administrative authentication </div> <div style="flex: 1;"> Authentication <ul style="list-style-type: none"> <input checked="" type="radio"/> LTPA <input type="radio"/> Kerberos and LTPA <ul style="list-style-type: none"> Kerberos configuration <input type="radio"/> SWAM (deprecated): No authentication <ul style="list-style-type: none"> Authentication cache settings </div> </div> <div style="display: flex; justify-content: space-between;"> <div style="flex: 1;"> Application security <p><input checked="" type="checkbox"/> Enable application security</p> </div> <div style="flex: 1;"> Java 2 security <p><input type="checkbox"/> Use Java 2 security to restrict application access to local resources <ul style="list-style-type: none"> <input checked="" type="checkbox"/> Warn if applications are granted custom permissions <input type="checkbox"/> Restrict access to resource authentication data </p> </div> </div> <div style="display: flex; justify-content: space-between;"> <div style="flex: 1;"> User account repository <p>Current realm definition: Federated repositories</p> <p>Available realm definitions: Federated repositories Configure... Set as current</p> </div> <div style="flex: 1;"> <p>Specifies a list of user identities and passwords for Java(TM) 2 connector security to use.</p> <ul style="list-style-type: none"> Security domains External authorization providers Custom properties </div> </div>	

WebSphere software

A list of existing aliases is displayed.

[Global security > JAAS - J2C authentication data](#)

Specifies a list of user identities and passwords for Java(TM) 2 connector security to use.

Prefix new alias names with the node name of the cell (for compatibility with earlier releases)

[Apply](#)

[+ Preferences](#)

New	Delete			
Select	Alias	User ID	Description	
You can administer the following resources:				
<input type="checkbox"/>	BSpace JDBC Alias	wbiuser	Business Space Authorization Alias	
<input type="checkbox"/>	CommonEventInfrastructureJMSAuthAlias	wbiuser	Authentication alias for the Common Event Infrastructure JMS Topics and Queues	
<input type="checkbox"/>	SCA Auth Alias	wbiuser	This is the alias used by SCA to login to a secured SIBus	
<input type="checkbox"/>	localhostNode01Cell/nlNode01/server1/EventAuthDataAliasDerby		Derby authentication alias for the Event Server	
Total 4				

7. Click **New** to create a new authentication entry. Type the alias name, and a username and password that can connect to the database. Click **OK**.

Global security

[Global security](#) > [JAAS - J2C authentication data](#) > New

Specifies a list of user identities and passwords for Java(TM) 2 connector security to use.

General Properties

* Alias
Alias DB2

* User ID
db2admin

* Password

Description

Apply OK Reset Cancel

- Click **Save** to save the changes.

Cell=localhostNode01Cell, Profile=AppSrv01

Global security

[Global security](#) > [JAAS - J2C authentication data](#)

Specifies a list of user identities and passwords for Java(TM) 2 connector security to use.

Prefix new alias names with the node name of the cell (for compatibility with earlier releases)

Apply

Messages

⚠ Changes have been made to your local configuration. You can:

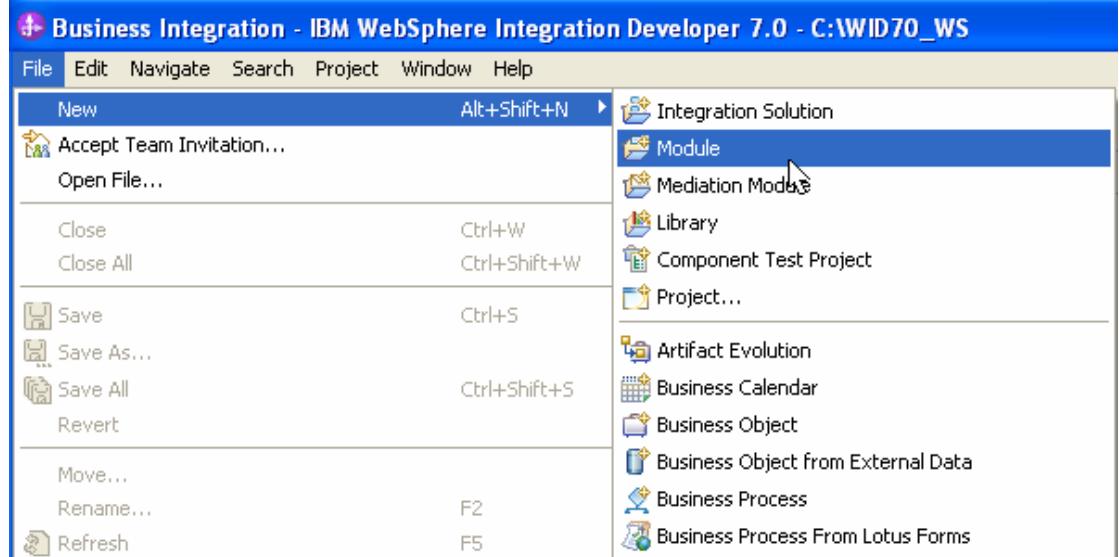
- [Save](#) directly to the master configuration.
- [Review](#) changes before saving or discarding.

⚠ The server may need to be restarted for these changes to take effect.

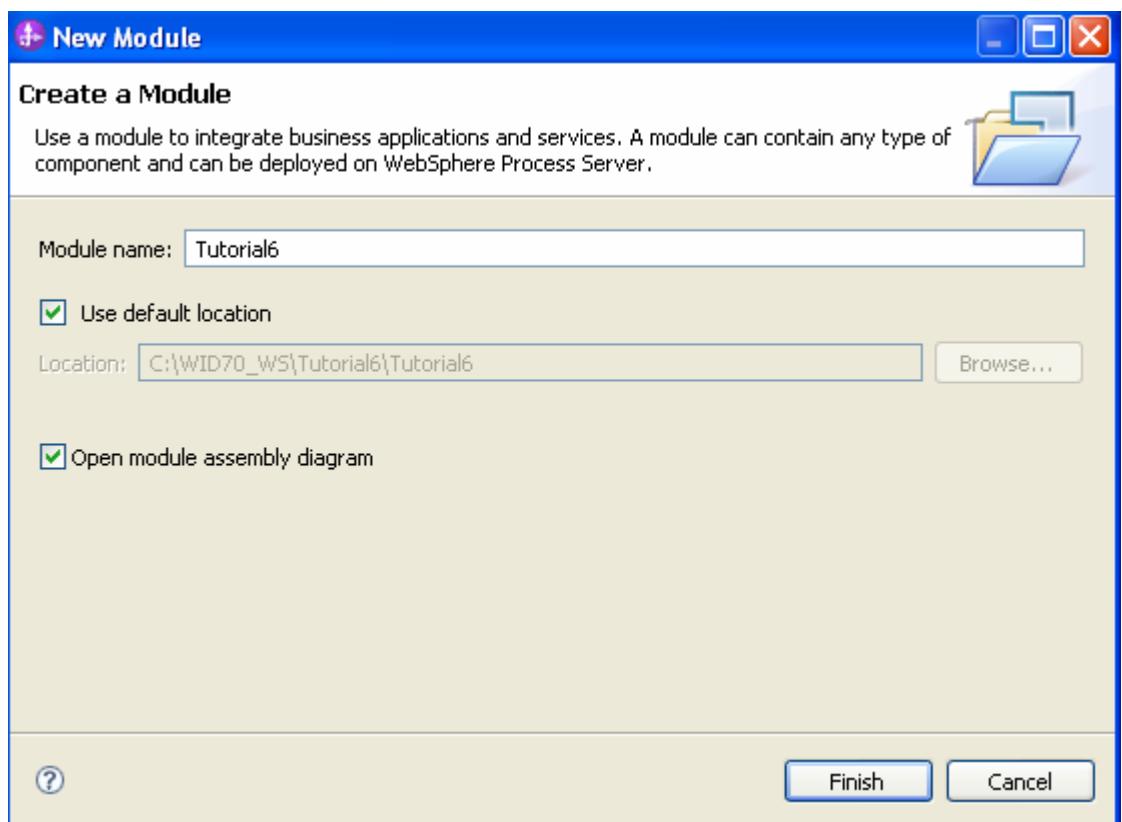
Note: You have created an authentication alias that will be used when you configure the adapter properties. Re-start the server for the changes to take effect.

Configure the adapter for outbound processing

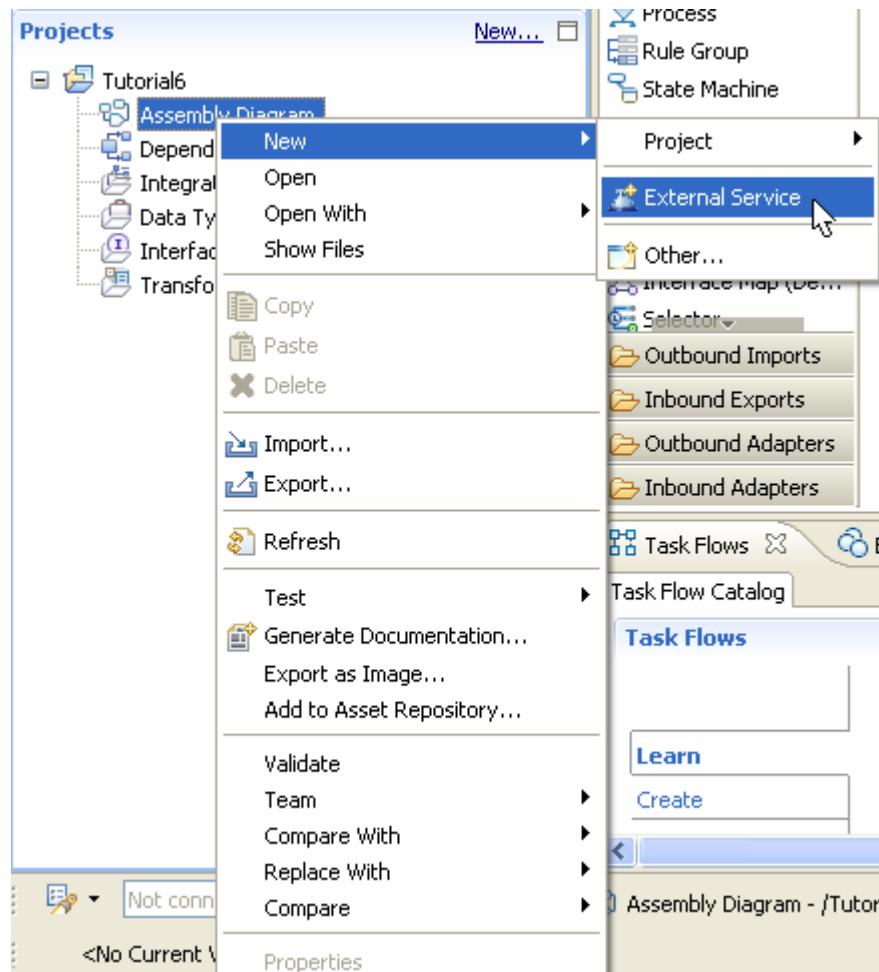
- Switch to the Business Integration perspective in WebSphere Integration Developer.
- Select **File->New->Module** to create a module project.



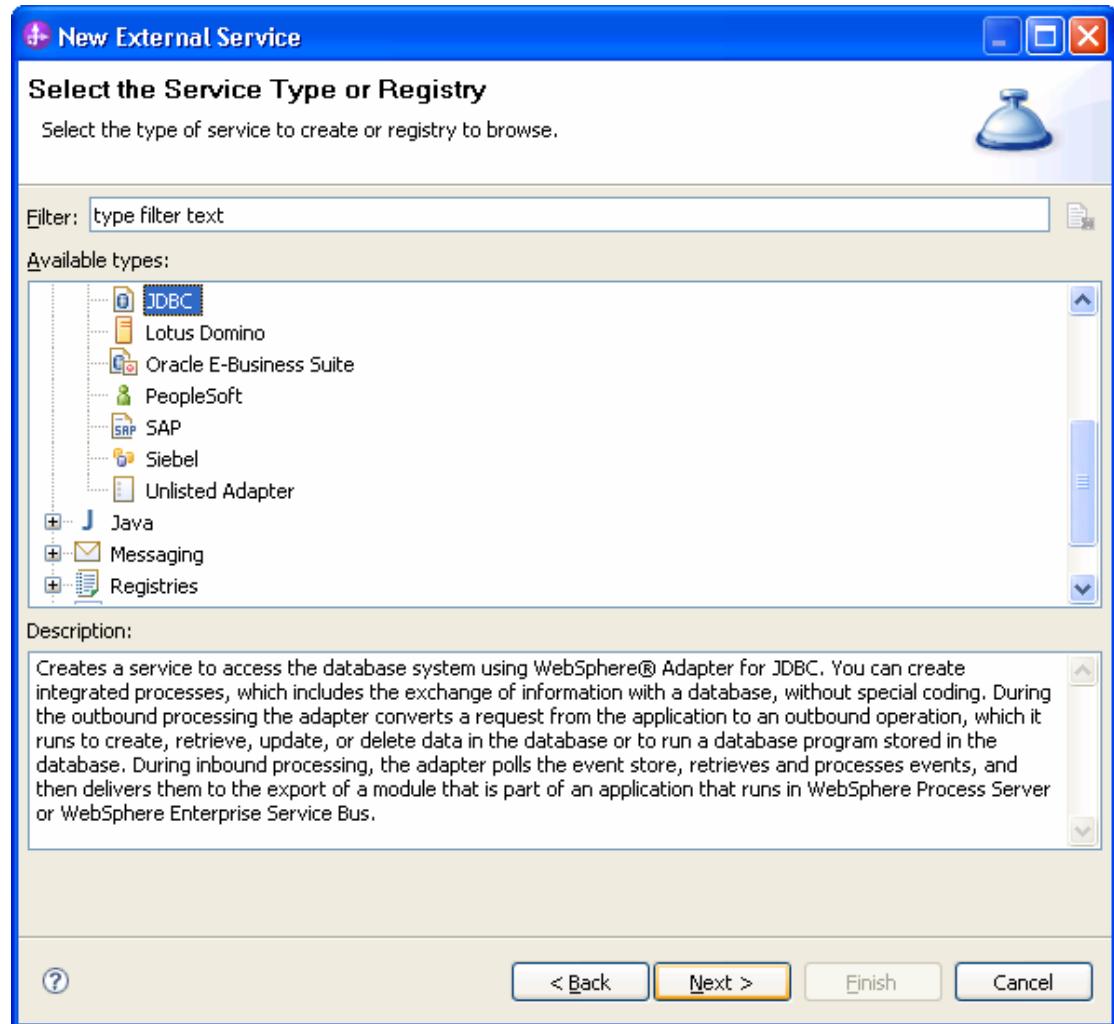
3. Specify the module name as **Tutorial6**, and click **Finish**.



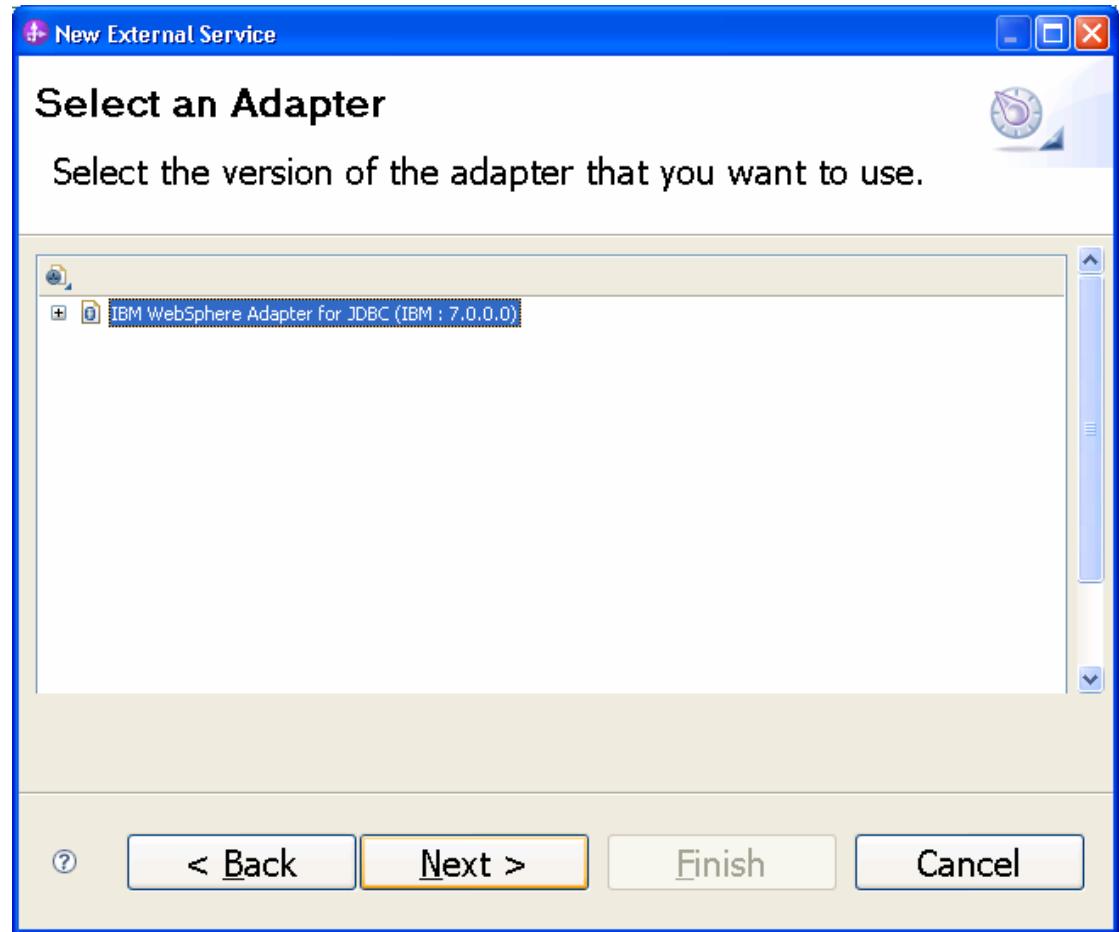
4. Expand Tutorial6 and select displayed. Right-click and select **New->External Service**.



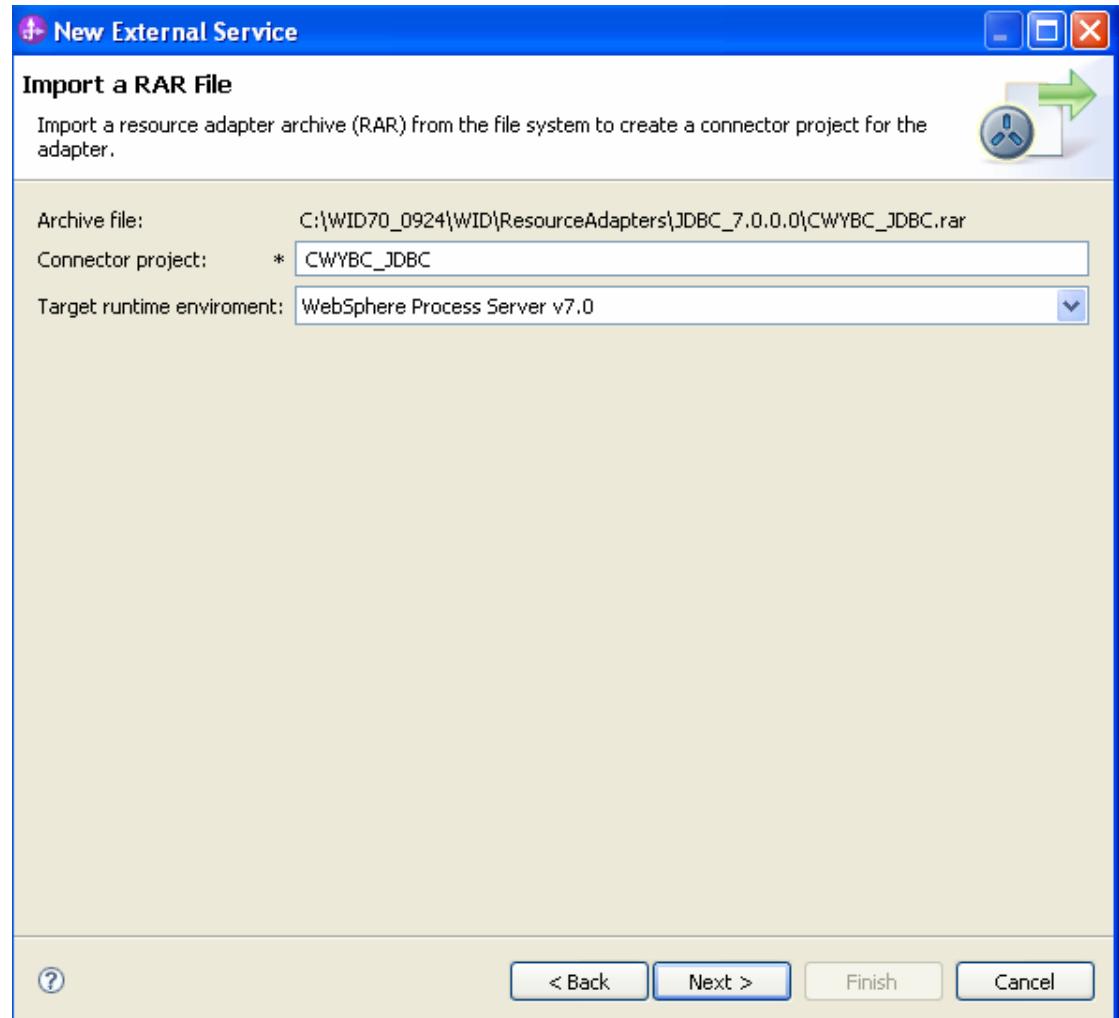
5. Select JDBC, and click **Next**.



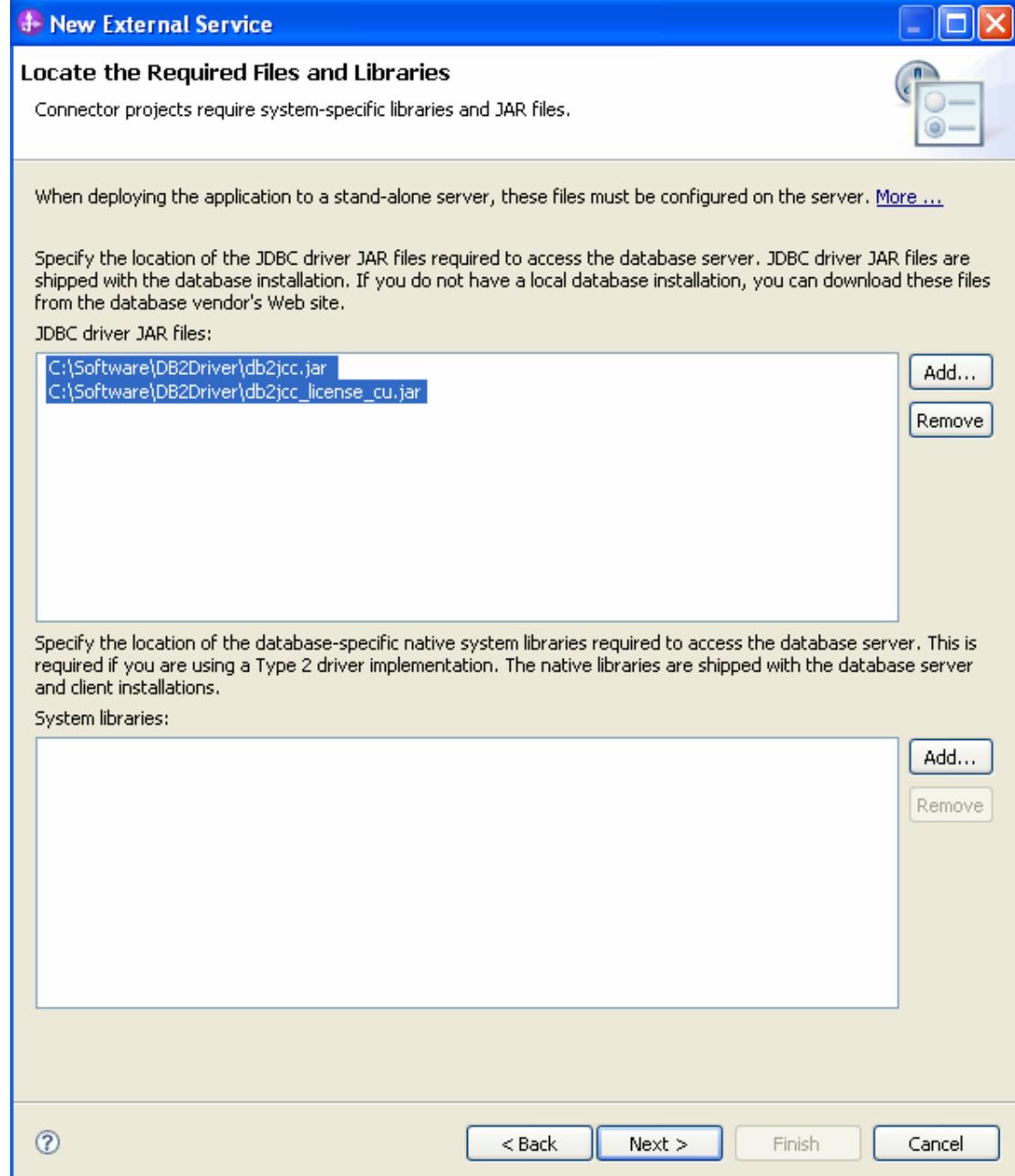
6. Select **IBM WebSphere Adapter for JDBC (IBM: 7.0.0.0)**. Click **Next**.



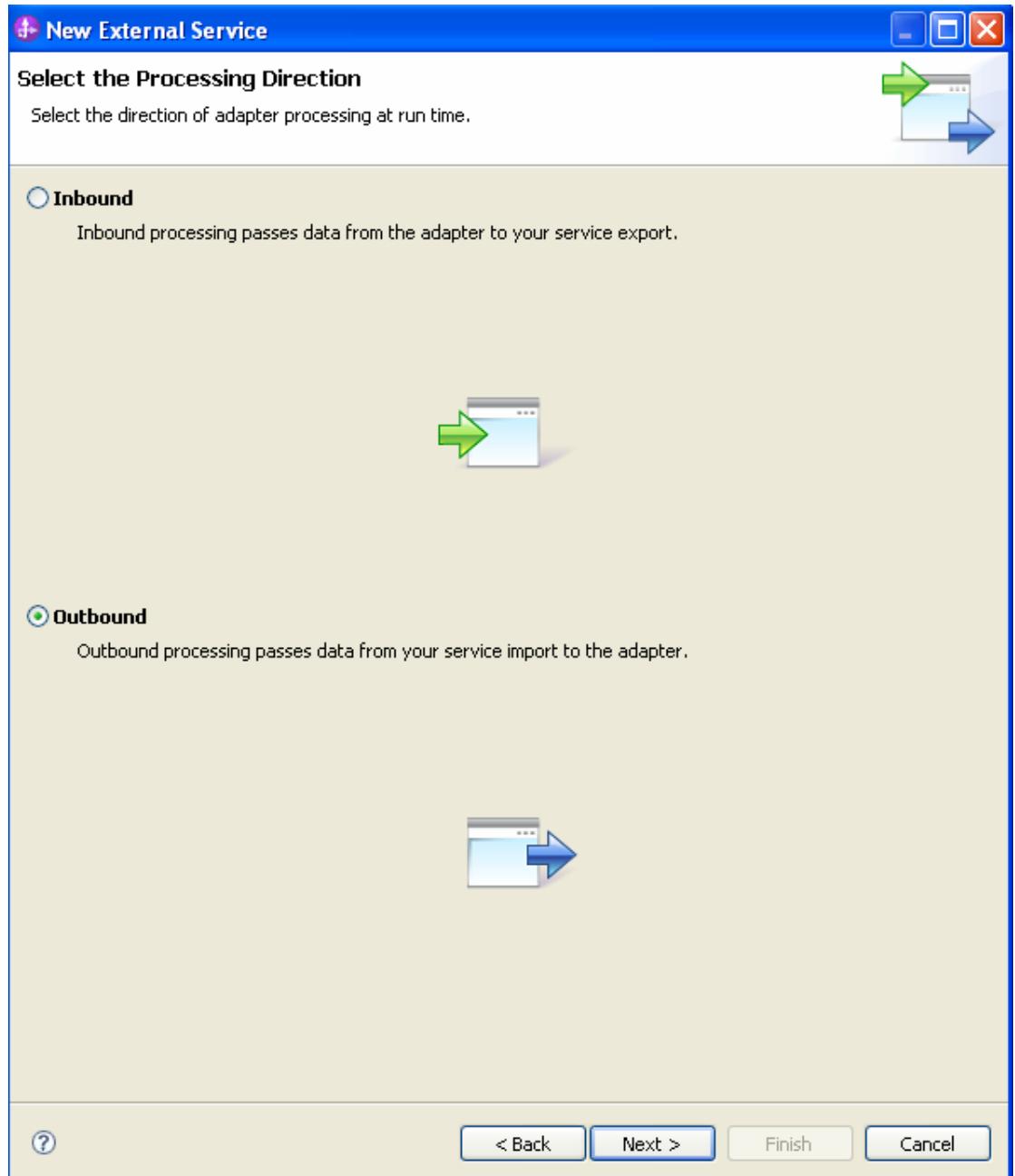
7. In the **Target Runtime environment** field, select the appropriate runtime and click **Next**.



8. In the **JDBC driver JAR files** field, click **Add** to add the JDBC driver class to connect to the database. Browse to select the driver JAR file and click **Next**.



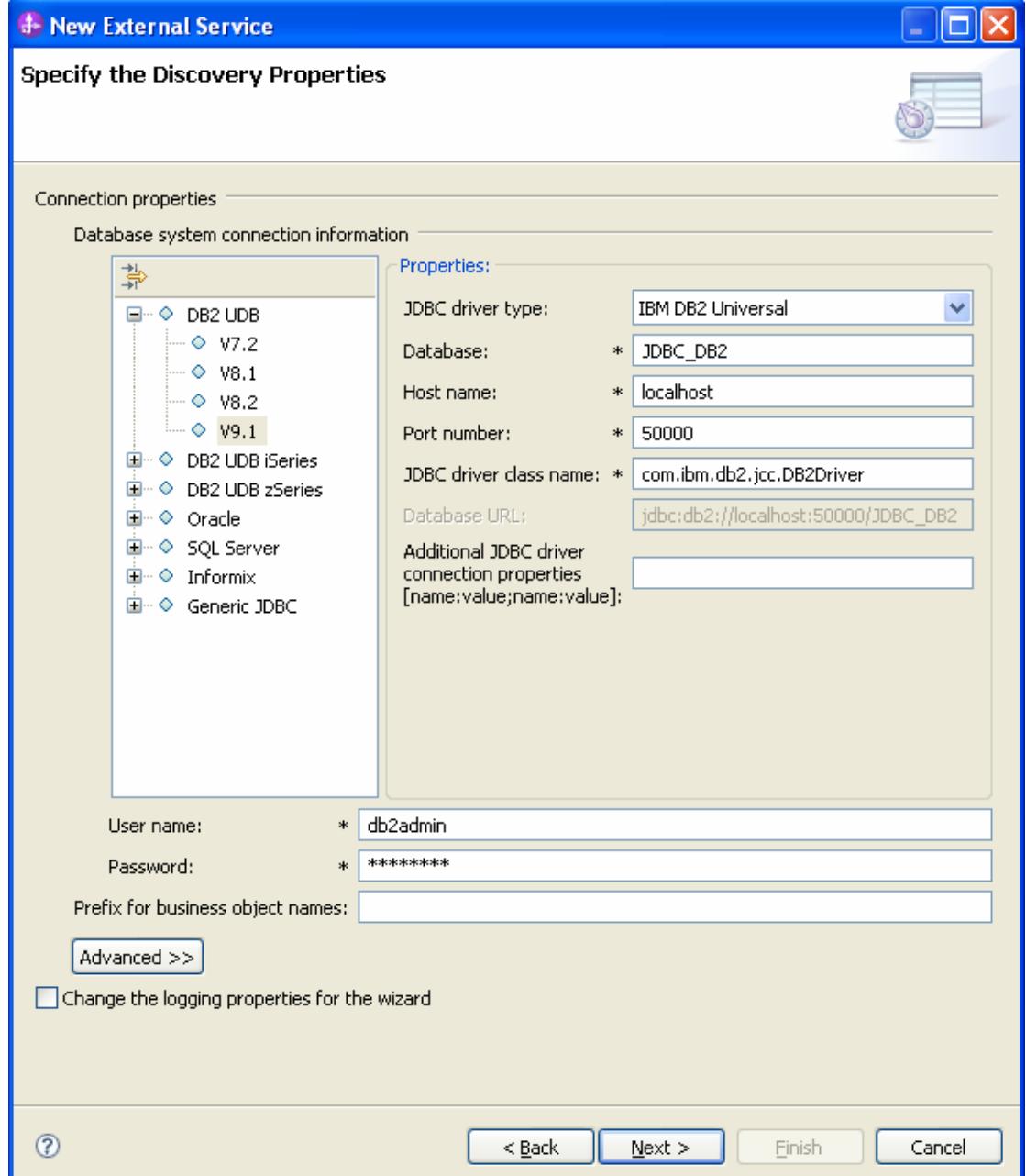
9. Select **Outbound** and click **Next**.



Set connection properties for the external service wizard

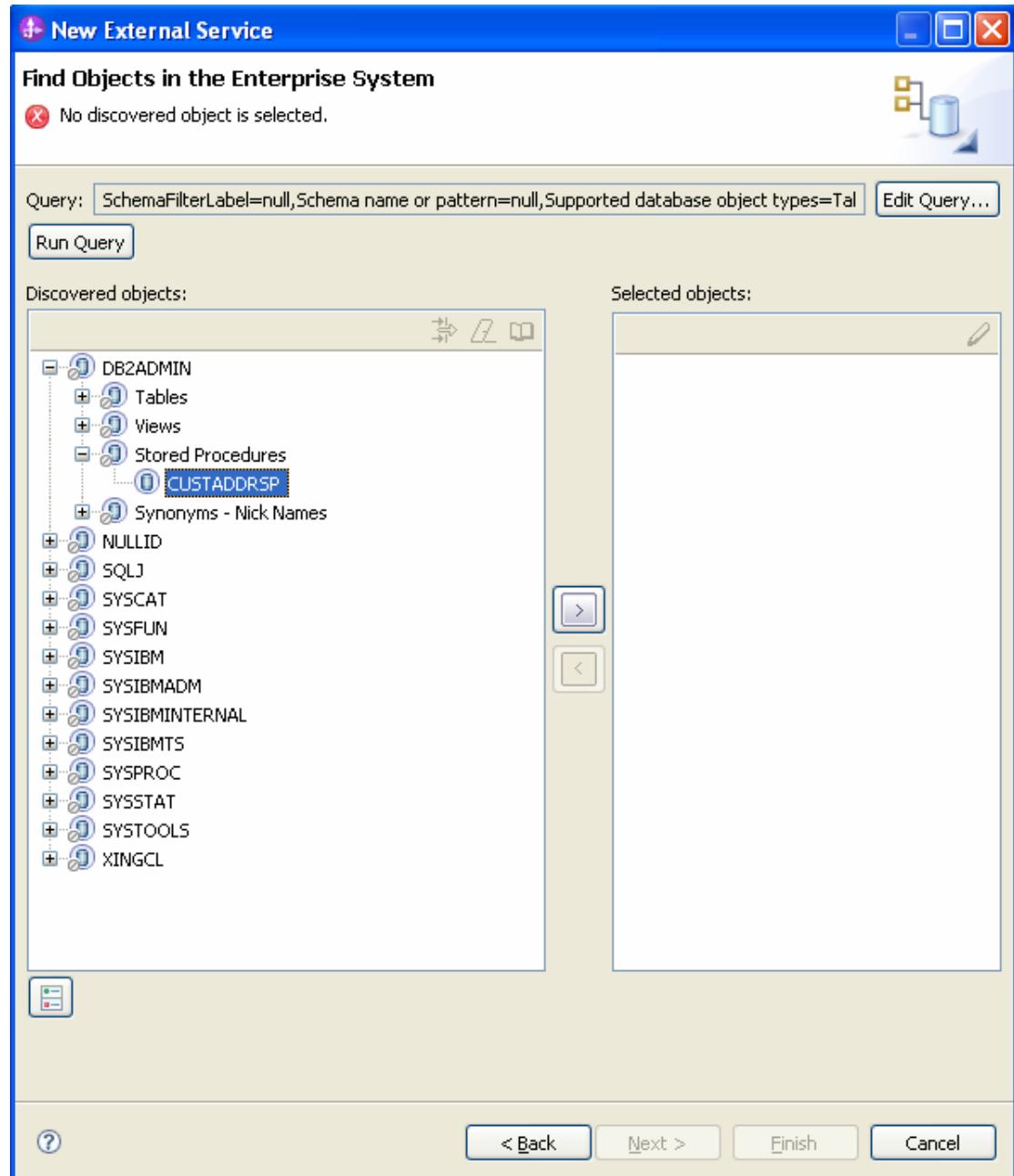
To connect to the database:

1. Expand the **DB2** node in the **Database system connection information** area and select appropriate version,
2. Enter values in the **Database**, **Host name**, **Port number**, **User name** and **Password** fields, and click **Next**.

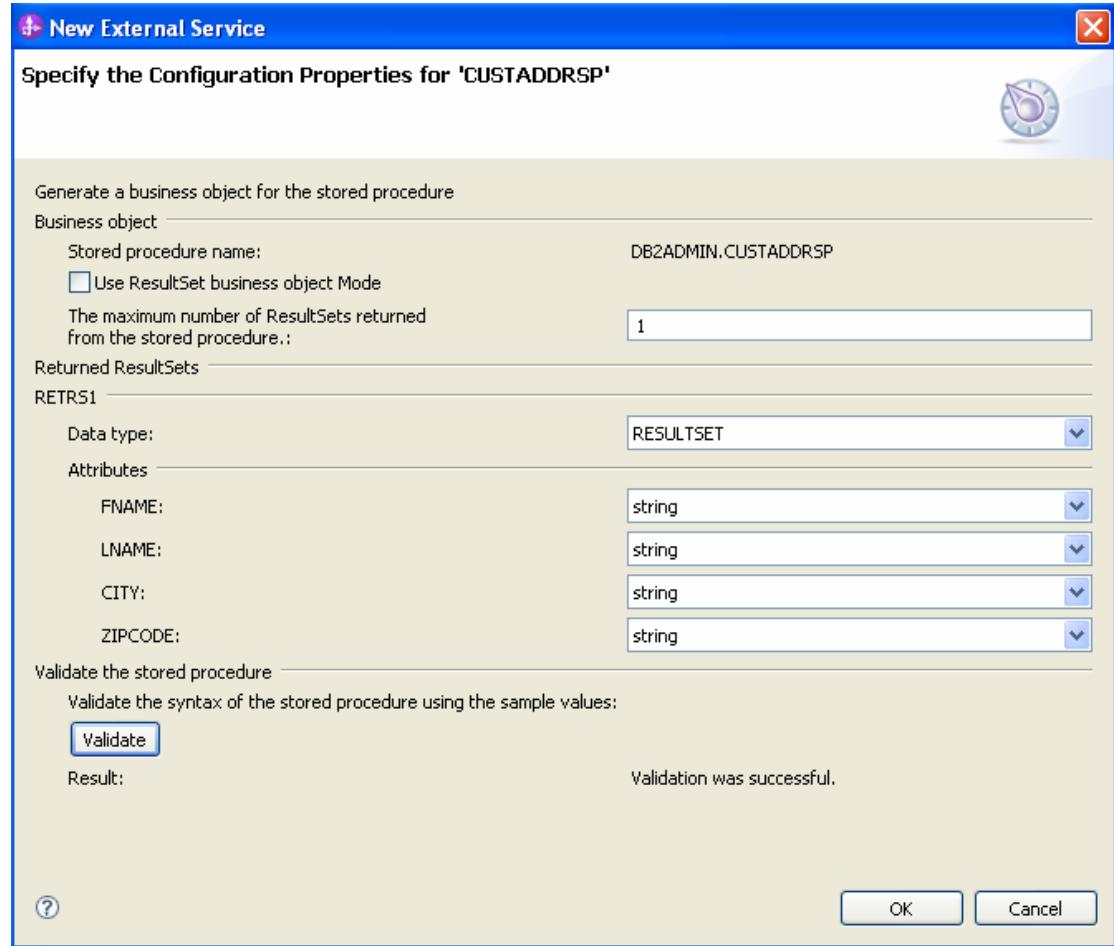


Select the business objects and services to be used with the adapter

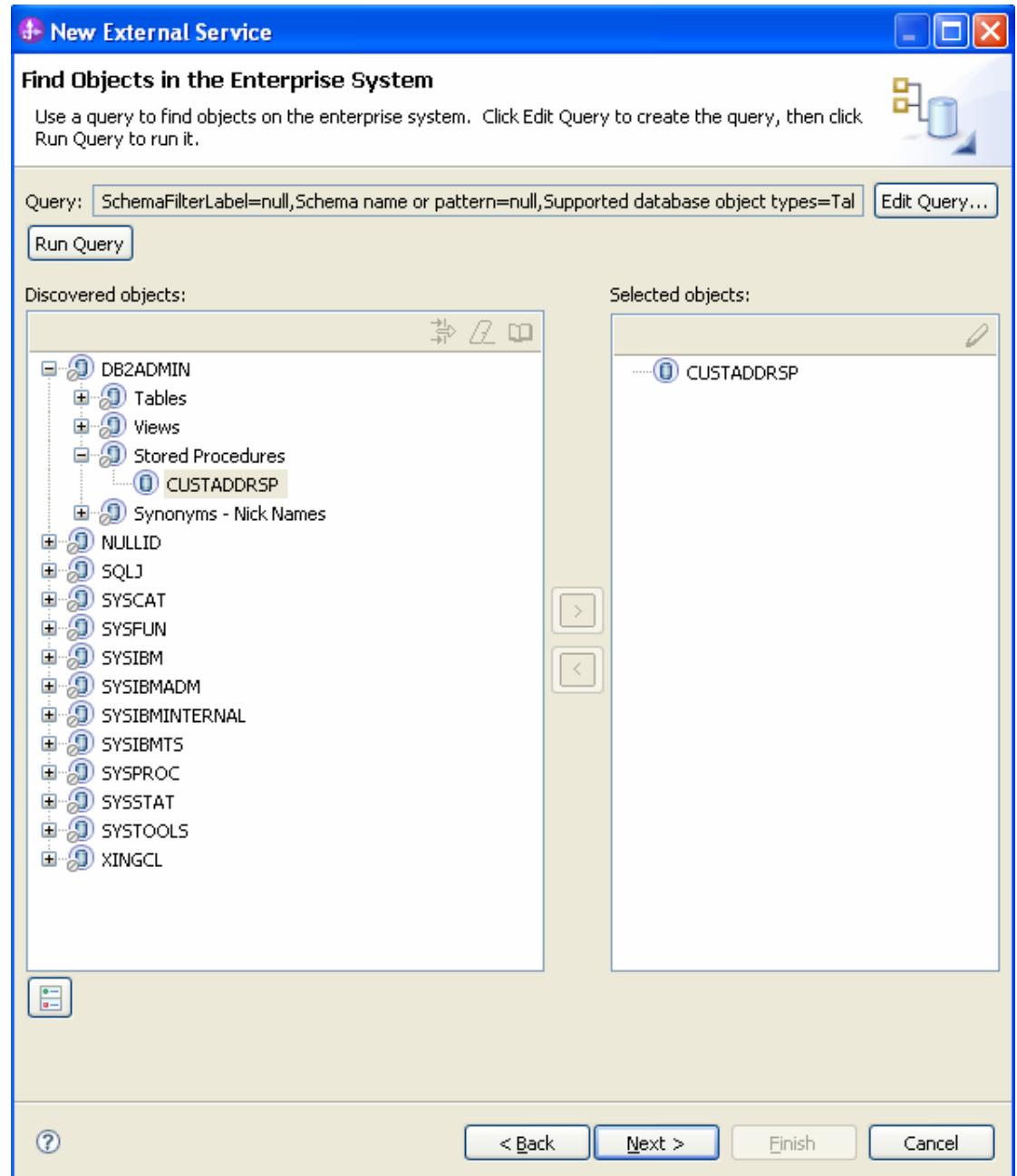
1. In the Find Object in the Enterprise System window, click **Run Query**.
2. Select **DB2ADMIN->Stored Procedures->CUSTADDRSP**, and click .



3. In the **The maximum number of ResultSets returned from the stored procedure** field, enter 1. Click **Validate**. After successful validation, click **OK**.



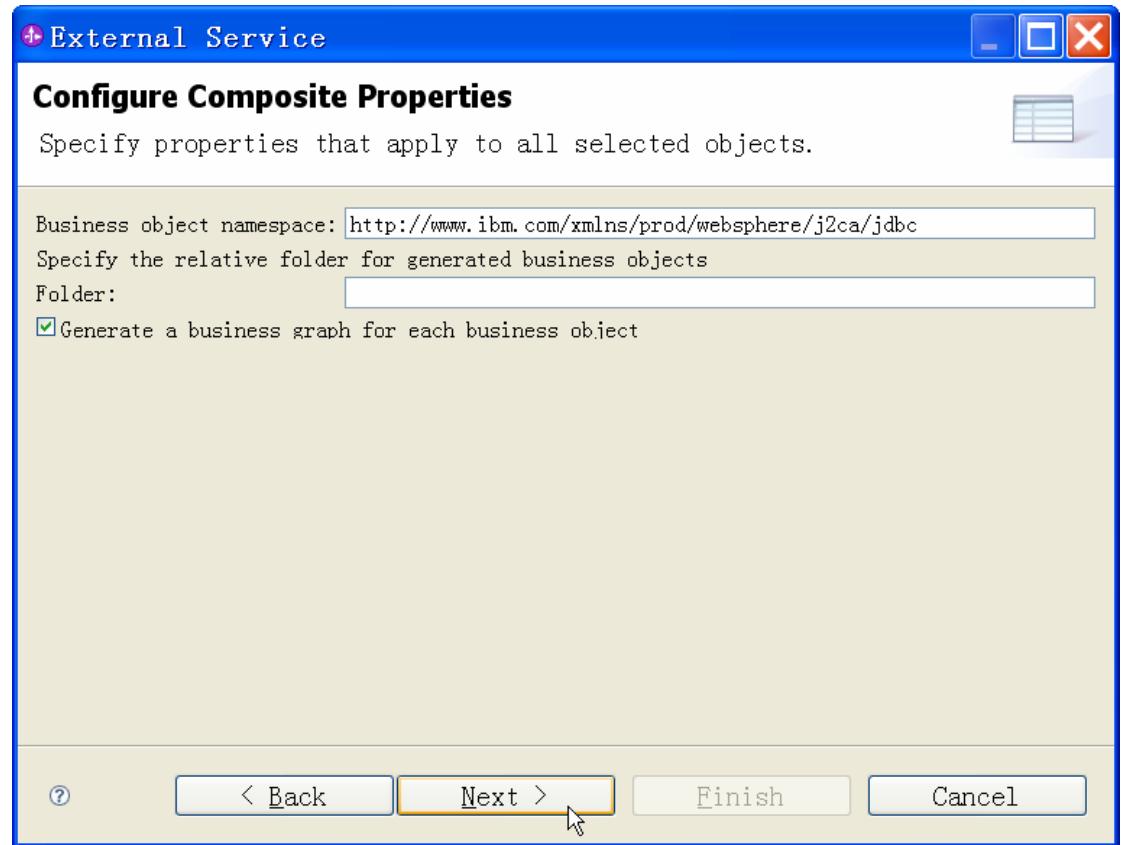
4. Click **Next**.



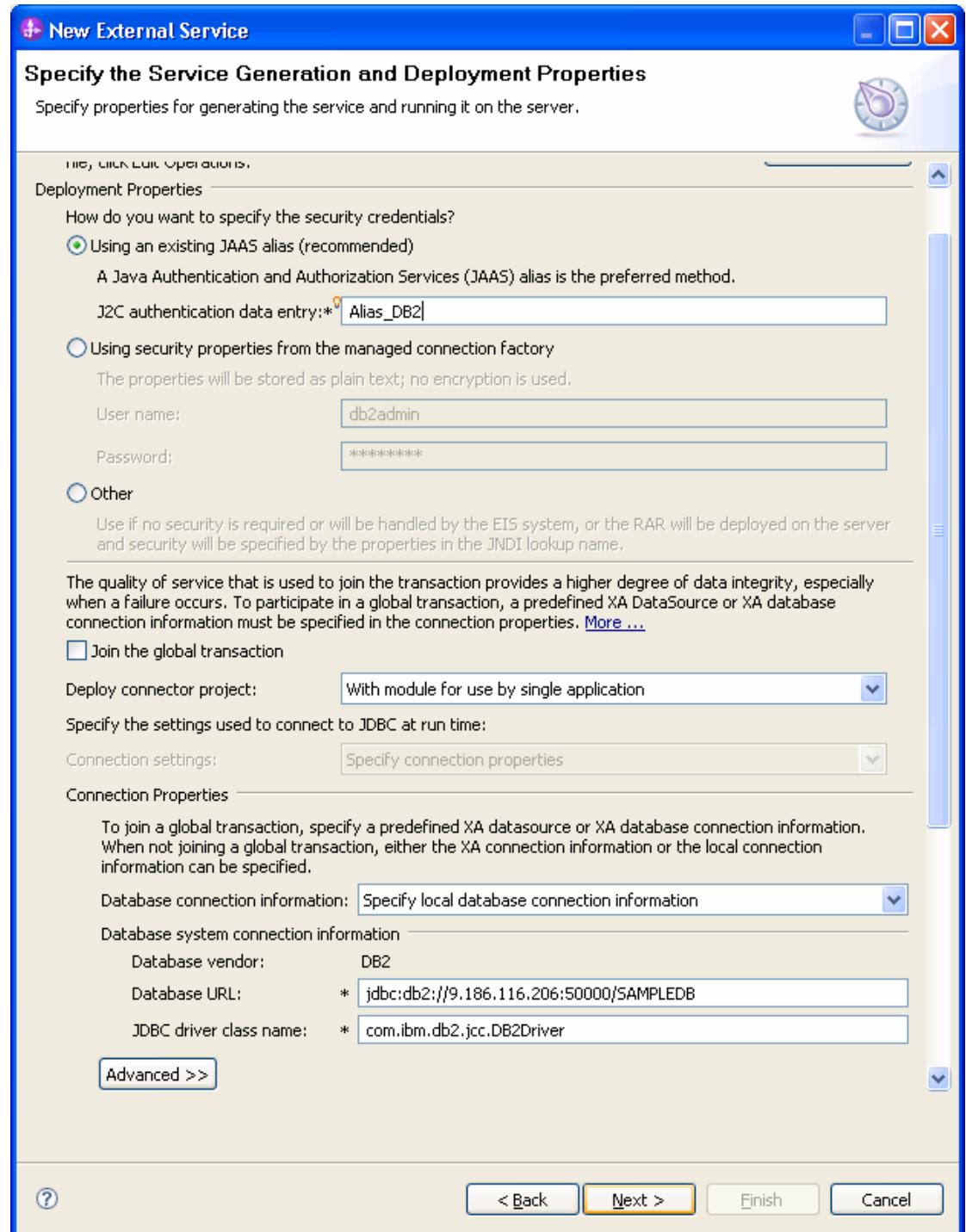
Generate business object definitions and related artifacts

Follow these steps to generate the business object definitions.

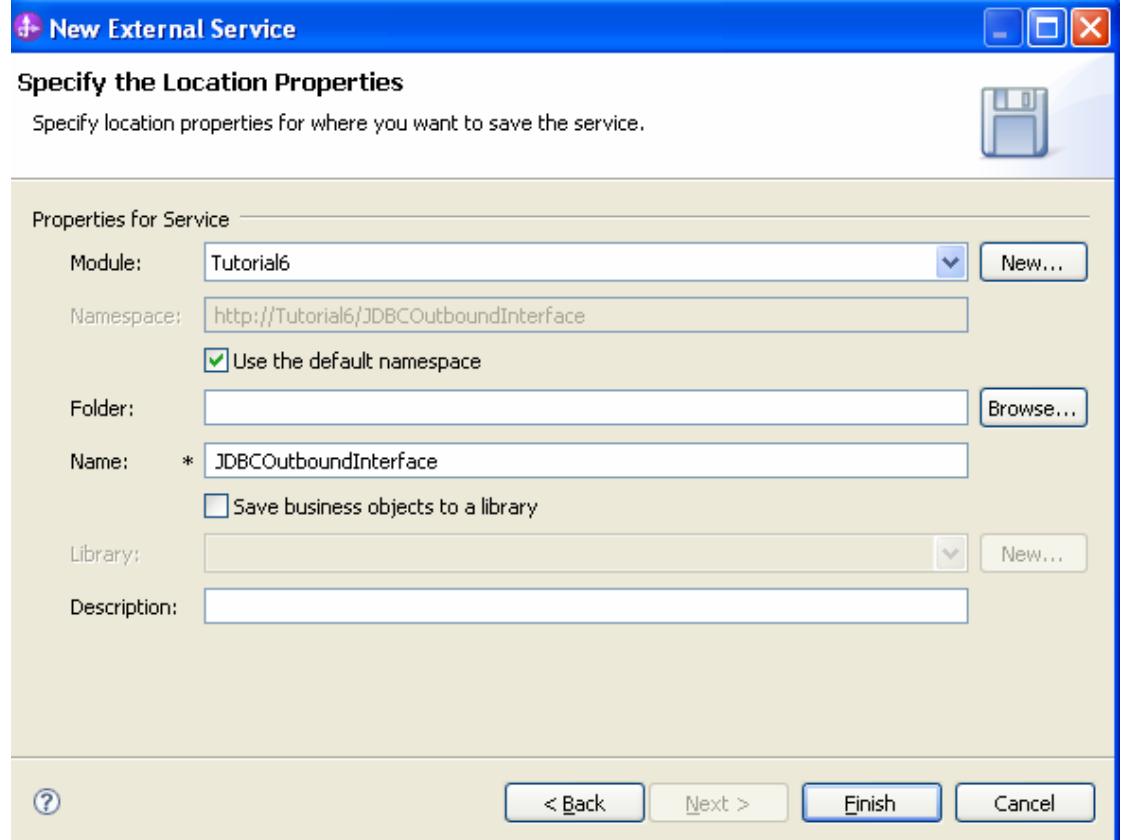
1. In the Specify Composite Properties window, accept the default values for the all fields and click **Next**.



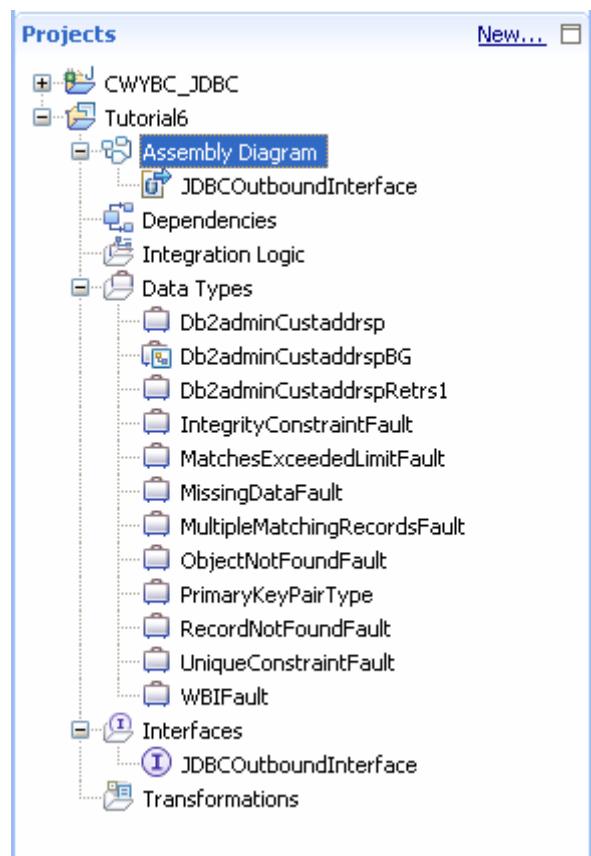
2. In the Specify the Service Generation and Deployment Properties window, perform the following steps:
 - a) In the **J2C authentication data entry** field, enter **Alias_DB2**.
 - b) Select the **Join the global transaction** check box.
 - c) Select **Specify local database connection information** from the **Database connection information** list and click **Next**.



3. In the Specify the Location Properties window, click **Finish** to complete the service creation.



4. Verify the results.

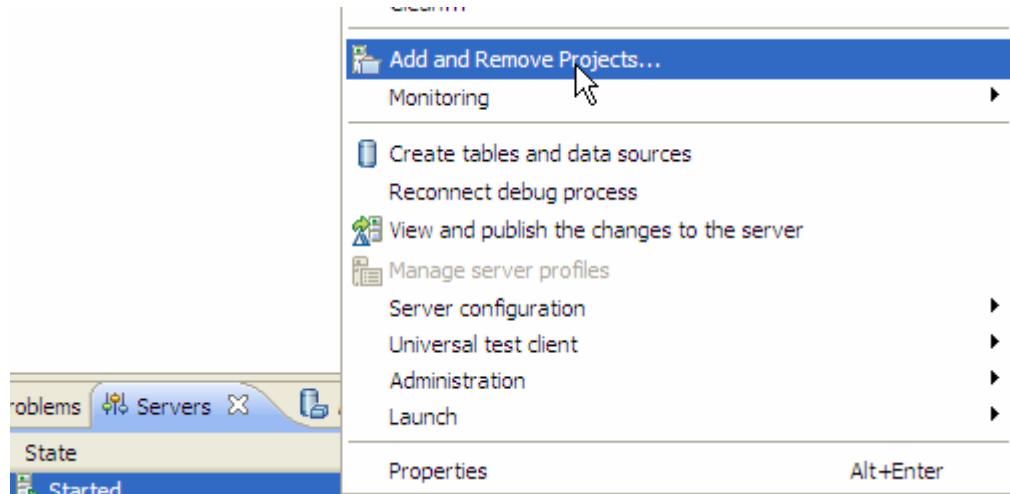


Deploy the module to the test environment

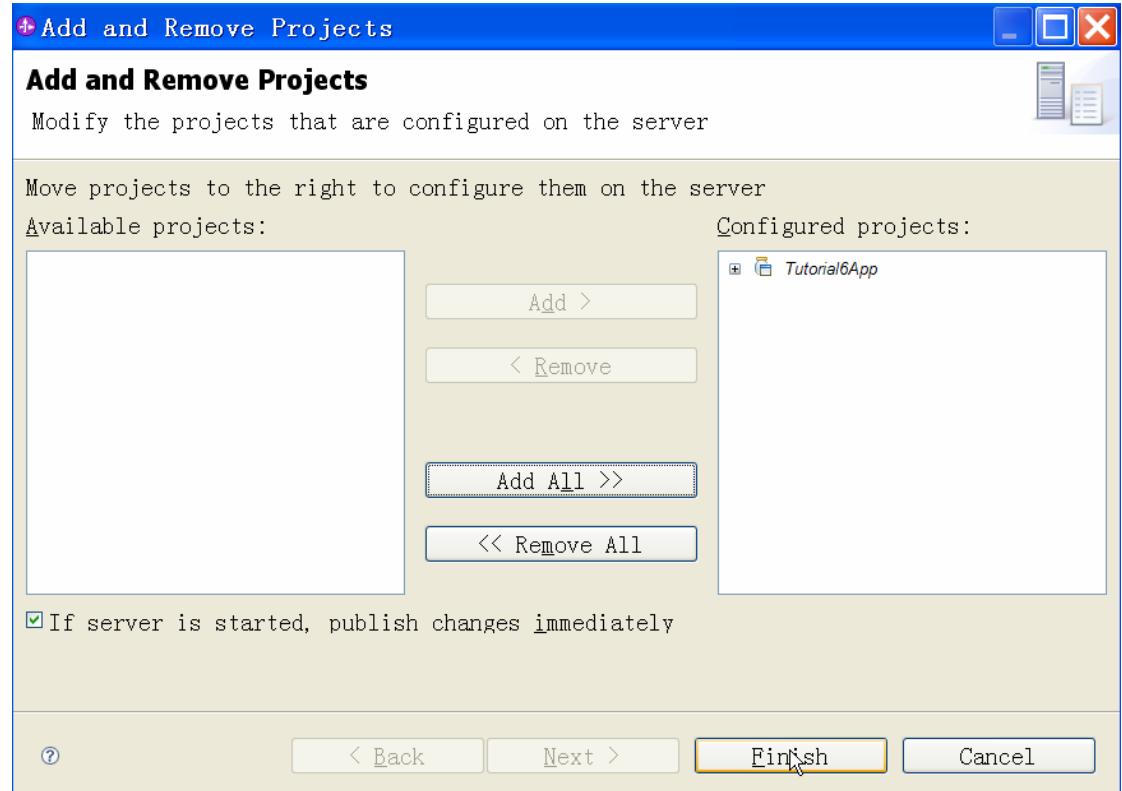
The result of running the external service wizard is an SCA module that contains an Enterprise Information System import. Install this SCA module in the WebSphere Integration Developer integration test client. To do this, you must add the SCA module you created earlier to the server using the **Servers** view in WebSphere Integration Developer.

Steps for adding the SCA module to the server:

1. In WebSphere Integration Developer, switch to the **Servers** view by selecting **Windows > Show View > Servers**.
2. In the Servers tab in the lower-right pane of the WebSphere Integration Developer screen, right-click the server, and select **Start**.
3. After the server is started, right-click the server, and select **Add and Remove projects**.



4. In the Add and Remove Projects window, select the module created earlier and click **Add**. The project moves to the **Configured Projects** list from the **Available Projects** list.

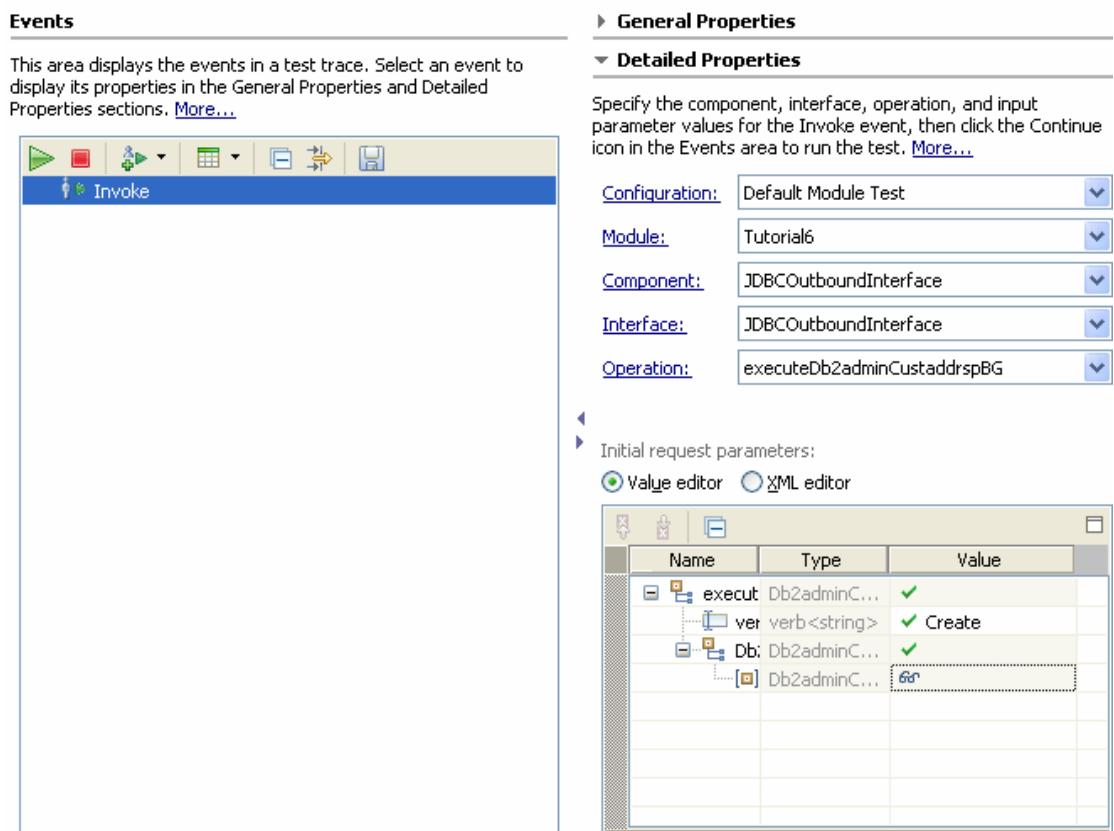


5. Click **Finish**. This deploys the project on the server. For troubleshooting issues while adding the project, see the Troubleshooting section. The Console tab in the lower-right pane displays a log while the module is being added to the server.

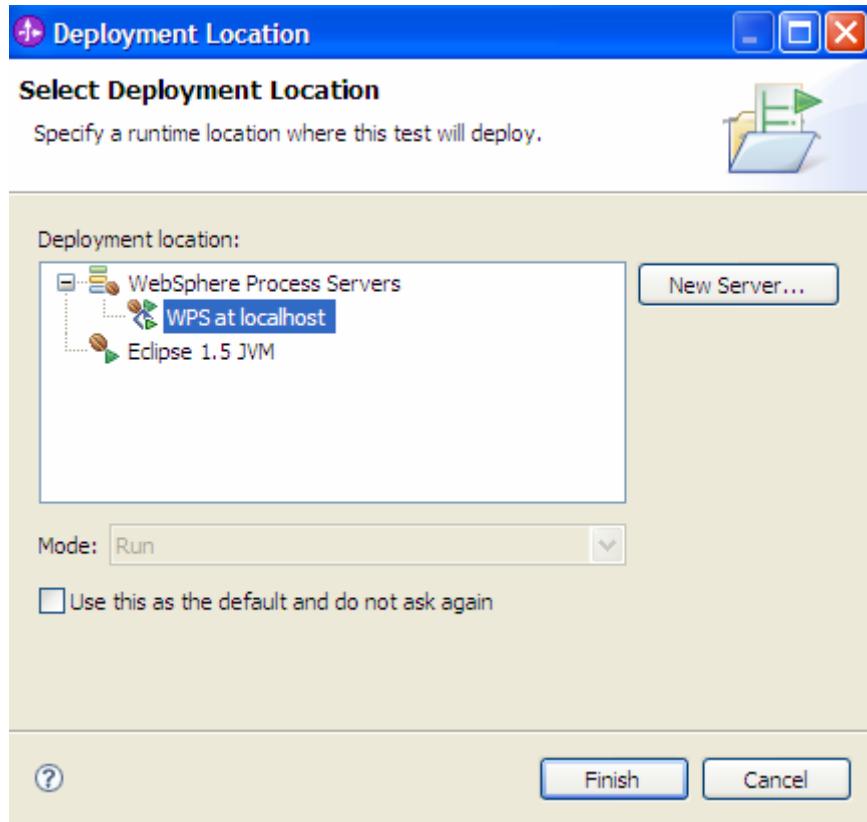
Test the assembled adapter application

Test the assembled adapter application using the WebSphere Integration Developer integration test client:

1. From the Business Integration view, right click on Tutorial6 and select Test > Test Module.
2. Leave all fields in the test client as default.



3. Execute the service by click .
4. In the Select Deployment Location window as shown below, select the server, and click **Finish**.



5. If the deployment and execution of the test module is successful, the result set should return the expected records that reflect the conditions stipulated in the stored procedure.

Detailed Properties

Module: [Tutorial6](#)
 Component: [IDBCOutboundInterface](#)
 Interface: [IDBCOutboundInterface](#)
 Operation: [executeDb2adminCustaddrspBG](#)

Return parameters:

Name	Type	Value
executeDb2adminCustaddrspBG	Db2adminCustaddrspBG	✓
verb	verb<string>	✓
Db2adminCustaddrsp	Db2adminCustaddrsp	✓
retrs1	Db2adminCustaddrspRetrs1[]	60
retrs1[0]	Db2adminCustaddrspRetrs1	✓
fname	string	fname1
lname	string	lname1
city	string	city1
zipcode	string	zipcode1
retrs1[1]	Db2adminCustaddrspRetrs1	✓
fname	string	fname1
lname	string	lname1
city	string	city1
zipcode	string	zipcode1

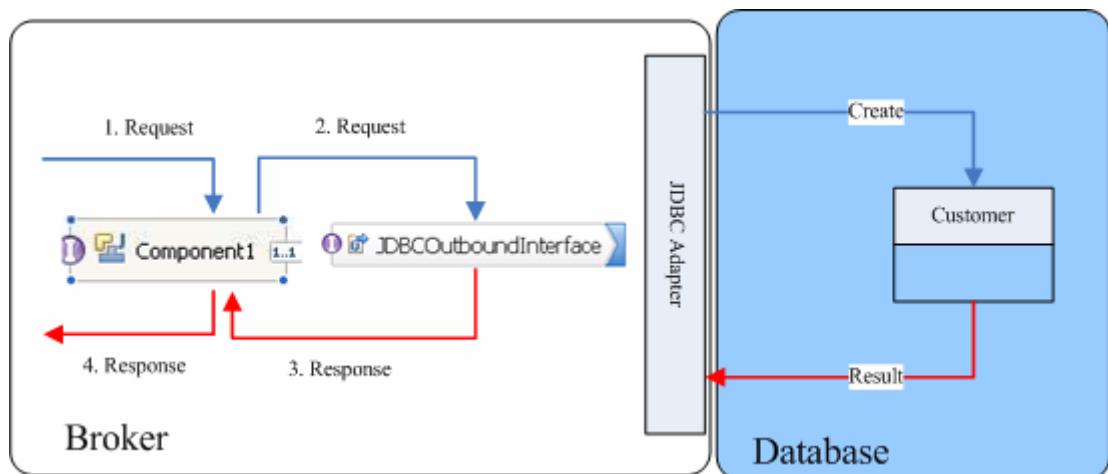
Chapter 8. Tutorial 7: Sending data to the DB2 database within XA Transaction (outbound processing)

This scenario demonstrates how WebSphere Adapter for JDBC 7.0.0.0 participates in a global transaction using XA a data source for DB2 database.

About this task

In this scenario, we will create a Java component and a JDBC adapter import component. The Java component invokes JDBC adapter to make changes to the database. Both, the java component and JDBC adapter will participate in the same global transaction.

The following figure represents this scenario:



Prepare to run through the tutorial

Extract the sample files

Replicas of the artifacts that you create when using the external service wizard are provided as sample files for your reference. Use these files to verify if the files you create using the external service wizard are correct.

Download the sample zip file and extract it into a directory of your choice (you may want to create a new directory).

Configuration prerequisites

Before configuring the adapter, you must complete the following tasks:

- Create a table
- Create an authentication alias
- Create a data source

Create a table

You must create the following table in the DB2 database before starting the scenario.

Script for creating the CUSTOMER table

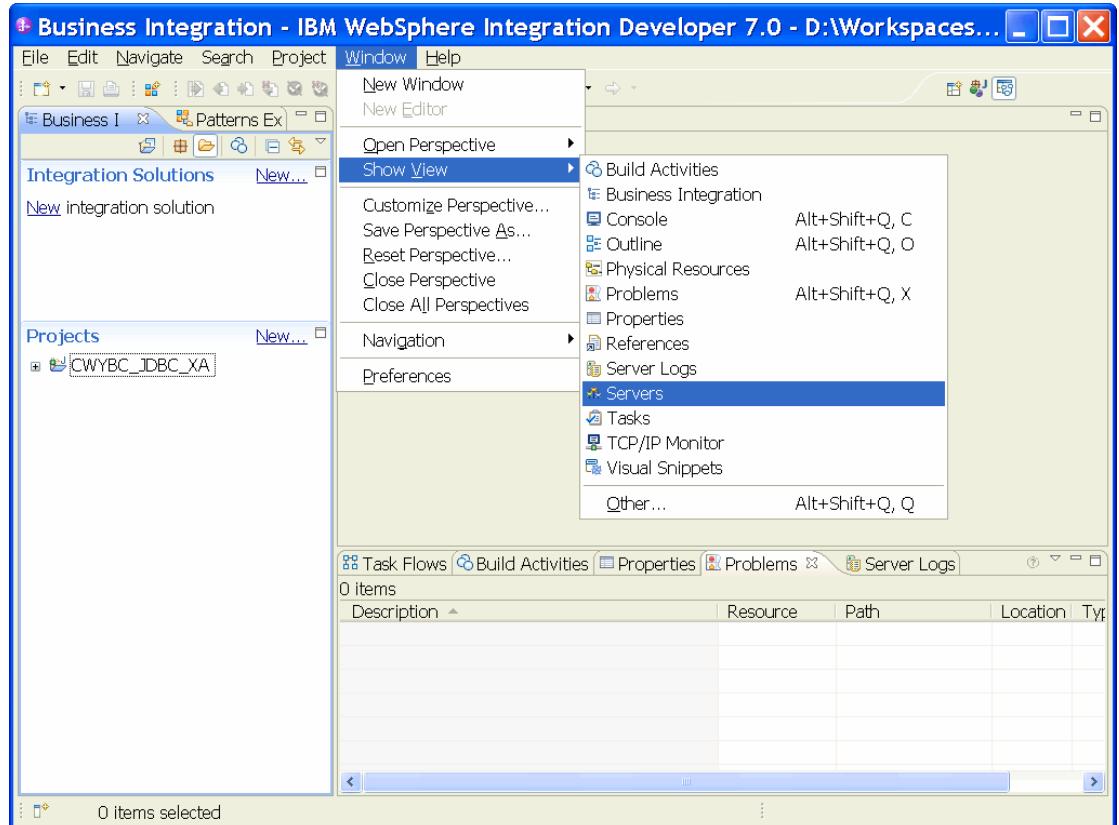
```
CREATE TABLE CUSTOMER (
    "PKEY" INTEGER NOT NULL PRIMARY KEY,
    "FNAME" VARCHAR(20) ,
    "LNAME" VARCHAR(20) ,
    "CCODE" VARCHAR(10) ) ;
```

Create an authentication alias

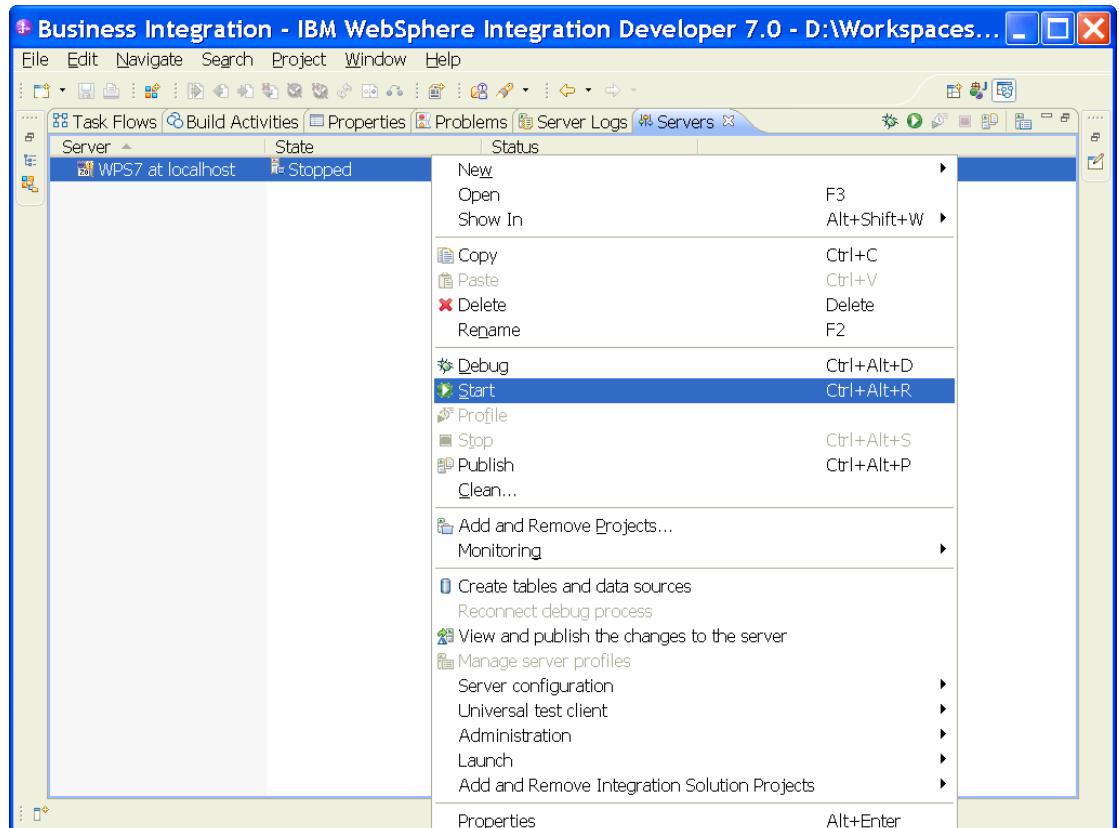
The authentication alias needs to be set because the adapter uses the username and password set in the authentication alias to connect to the database. This authentication alias will be used later when generating the artifacts for the module.

Here are the steps to set the authentication alias in WebSphere Process Server administration console.

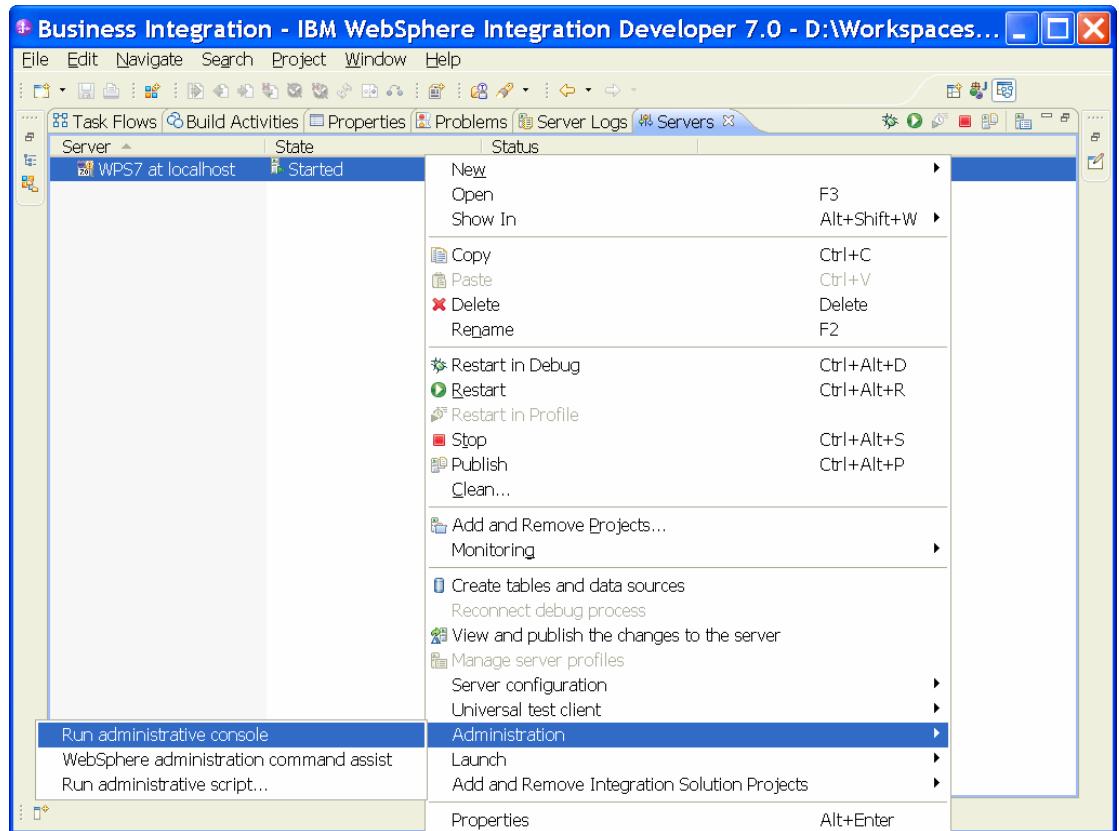
1. In WebSphere Integration Developer, switch to the **Servers** view by selecting **Windows > Show View > Servers**.



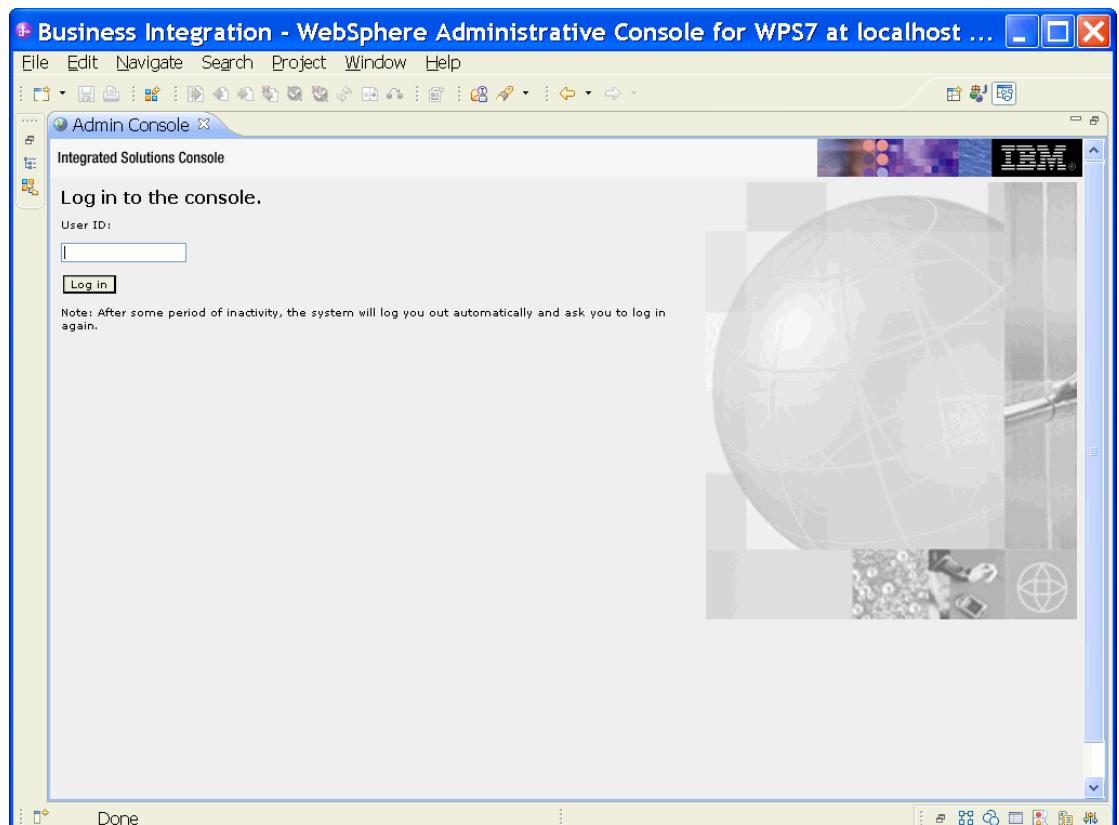
2. In the **Servers** view, right-click the server that you want to start, and select **Start**.



3. After the server is started, right-click the server, and select **Administration > Run administrative console**.



4. Log on to the administrative console by entering the username and password (if required).



5. Click **Security → Global security**.



- Under **Java Authentication and Authorization Service**, click **J2C authentication data**.

Cell=localhostNode01Cell, Profile=AppSrv01 Close page

Global security

Global security
Use this panel to configure administration and the default application security policy. This security configuration applies to functions and is used as a default security policy for user applications. Security domains can be defined to override and control applications.

Security Configuration Wizard	Security Configuration Report	
<div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> Administrative security <p><input type="checkbox"/> Enable administrative security Administrative user roles <input type="checkbox"/> Administrative group roles <input type="checkbox"/> Administrative authentication</p> </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> Application security <p><input checked="" type="checkbox"/> Enable application security</p> </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> Java 2 security <p><input type="checkbox"/> Use Java 2 security to restrict application access to local resources <input checked="" type="checkbox"/> Warn if applications are granted custom permissions <input type="checkbox"/> Restrict access to resource authentication data</p> </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> User account repository <p>Current realm definition: Federated repositories</p> <p>Available realm definitions: Federated repositories Configure... Set as current</p> </div>		<div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> Authentication <p>Authentication mechanisms and expiration</p> <p><input checked="" type="radio"/> LTPA</p> <p><input type="radio"/> Kerberos and LTPA Kerberos configuration</p> <p><input type="radio"/> SWAM (deprecated): No authentication</p> <p>Authentication cache settings</p> </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> Web and SIP security </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> RMI/IOP security </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> Java Authentication and Authorization <p><input type="checkbox"/> Application logins <input type="checkbox"/> System logins <input type="checkbox"/> J2C authentication data</p> </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> <p>Specifies a list of user identities and passwords for Java(TM) 2 connector security to use.</p> <p><input type="checkbox"/> Security domains <input type="checkbox"/> External authorization providers <input type="checkbox"/> Custom properties</p> </div>

WebSphere software

A list of existing aliases is displayed.

[Global security > JAAS - J2C authentication data](#)

Specifies a list of user identities and passwords for Java(TM) 2 connector security to use.

Prefix new alias names with the node name of the cell (for compatibility with earlier releases)

[Apply](#)

[+ Preferences](#)

New	Delete			
Select	Alias	User ID	Description	
You can administer the following resources:				
<input type="checkbox"/>	BSpace JDBC Alias	wbiuser	Business Space Authorization Alias	
<input type="checkbox"/>	CommonEventInfrastructureJMSAuthAlias	wbiuser	Authentication alias for the Common Event Infrastructure JMS Topics and Queues	
<input type="checkbox"/>	SCA Auth Alias	wbiuser	This is the alias used by SCA to login to a secured SIBus	
<input type="checkbox"/>	localhostNode01Cell/nlNode01/server1/EventAuthDataAliasDerby		Derby authentication alias for the Event Server	
Total 4				

- Click **New** to create a new authentication entry. Type the alias name and a username and password that can connect to the database, as shown in the figure. Click **OK**.

Global security

[Global security](#) > [JAAS - J2C authentication data](#) > New

Specifies a list of user identities and passwords for Java(TM) 2 connector security to use.

General Properties

* Alias
Alias DB2

* User ID
db2admin

* Password

Description

Buttons: Apply, OK, Reset, Cancel

- Click **Save** to save the changes.

Cell=localhostNode01Cell, Profile=AppSrv01

Global security

[Global security](#) > [JAAS - J2C authentication data](#)

Specifies a list of user identities and passwords for Java(TM) 2 connector security to use.

Prefix new alias names with the node name of the cell (for compatibility with earlier releases)

Buttons: Apply

Messages

⚠ Changes have been made to your local configuration. You can:

- [Save](#) directly to the master configuration.
- [Revert](#) changes before saving or discarding.

⚠ The server may need to be restarted for these changes to take effect.

Create a data source

- In administrative console, select **Resources->JDBC->JDBC Providers**.

Integrated Solutions Console Welcome admin

View: All tasks

- Welcome
- ⊕ Guided Activities
- ⊕ Servers
- ⊕ Applications
- ⊖ Resources
 - Schedulers
 - Object pool managers
 - ⊕ JMS
 - People directory provider
 - Extended messaging provider
 - WebSphere Business Integration Adapters
 - ⊖ JDBC
 - JDBC Providers
 - Data sources
 - Data sources (WebSphere Application Server V4)
 - Business Integration Data Sources
 - ⊕ Resource Adapters
 - ⊕ Asynchronous beans
 - ⊕ Cache instances
 - ⊕ Mail
 - ⊕ Remote artifacts
 - ⊕ URL
 - ⊕ Resource Environment
- ⊕ Security

2. On the right, select **Node=nINode01** from the drop-down list, and click **New**.

JDBC providers

JDBC providers

Use this page to edit properties of a JDBC provider. The JDBC provider object encapsulates the specific JDBC driver implementation class for access to the specific vendor database of your environment. Learn more about this task in a [guided activity](#). A guided activity provides a list of task steps and more general information about the topic.

Scope: Cell=localhostNode01Cell, Node=nINode01

Scope specifies the level at which the resource definition is visible. For detailed information on what scope is and how it works, [see the scope settings help](#)

Node=nINode01

Preferences

Select	Name	Scope	Description
<input type="checkbox"/>	Derby JDBC Provider (XA)	Node=nINode01	JDBC Provider for WPS/WESB

3. Select the following values for the **Database type**, **Provider type**, and **Implementation type** fields. Click **Next**.

Field	Value
Database type	DB2
Provider type	DB2 Universal JDBC Driver Provider
Implementation type	XA data source

Create a new JDBC Provider

→ Step 1: Create new JDBC provider

Step 2: Enter database class path information

Step 3: Summary

Create new JDBC provider

Set the basic configuration values of a JDBC provider, which encapsulates the specific vendor JDBC driver implementation classes that are required to access the database. The wizard fills in the name and the description fields, but you can type different values.

Scope
cells:localhostNode01Cell:nodes:n1Node01

* Database type
DB2

* Provider type
DB2 Universal JDBC Driver Provider

* Implementation type
XA data source

* Name
DB2 Universal JDBC Driver Provider (XA)

Description
XA DB2 Universal JDBC Driver-compliant Provider.
Datasources created under this provider support the use of XA to perform 2-phase commit processing.
Use of driver type 2 on WebSphere Application Server for Z/OS is not supported for datasources created under this provider.

Next **Cancel**

4. Enter the absolute path of the JDBC drivers (**db2jcc.jar**, **db2jcc_license_cu.jar**, **db2jcc_license_cisuz.jar**) directory. Click **Next**.

Create a new JDBC Provider

<p>Step 1: Create new JDBC provider</p> <p>→ Step 2: Enter database class path information</p> <p>Step 3: Summary</p>	<p>Enter database class path information</p> <p>Set the environment variables that represent the JDBC driver class files, which WebSphere(R) Application Server uses to define your JDBC provider. This wizard page displays the file names; you supply only the directory locations of the files. Use complete directory paths when you type the JDBC driver file locations. For example: /home/db2inst1/sqllib/java on Linux(TM). If a value is specified for you, you may click Next to accept the value.</p> <p>Class path:</p> <pre> \${DB2UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc.jar \${UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc_license_cu.jar \${DB2UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc_license_cisuz.jar</pre> <p>Directory location for "db2jcc.jar, db2jcc_license_cisuz.jar" which is saved as WebSphere variable \${DB2UNIVERSAL_JDBC_DRIVER_PATH} E:\lwq\DBDriver\DB2 9.5</p> <p>Native library path</p> <p>Directory location which is saved as WebSphere variable \${DB2UNIVERSAL_JDBC_DRIVER_NATIVEPATH}</p>
<input type="button" value="Previous"/> <input style="background-color: #0070C0; color: white; font-weight: bold; border: 1px solid black; border-radius: 5px; padding: 2px 10px;" type="button" value="Next"/> <input type="button" value="Cancel"/>	

5. Click **Finish**.

<p>Step 1: Create new JDBC provider</p> <p>Step 2: Enter database class path information</p> <p>→ Step 3: Summary</p>	<p>Summary</p> <p>Summary of actions:</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">Options</th> <th style="text-align: left;">Values</th> </tr> </thead> <tbody> <tr> <td>Scope</td> <td>cells:localhostNode01Cell:nodes:n1Node01</td> </tr> <tr> <td>JDBC provider name</td> <td>DB2 Universal JDBC Driver Provider (XA)</td> </tr> <tr> <td>Description</td> <td>XA DB2 Universal JDBC Driver-compliant Provider. Datasources created under this provider support the use of XA to perform 2-phase commit processing. Use of driver type 2 on WebSphere Application Server for Z/OS is not supported for datasources created under this provider.</td> </tr> <tr> <td>Class path</td> <td> <pre> \${DB2UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc.jar \${UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc_license_cu.jar \${DB2UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc_license_cisuz.jar</pre> </td> </tr> <tr> <td>Native path</td> <td>E:\lwq\DBDriver\DB2 9.5</td> </tr> <tr> <td>Implementation class name</td> <td>com.ibm.db2.jcc.DB2XADataSource</td> </tr> </tbody> </table>	Options	Values	Scope	cells:localhostNode01Cell:nodes:n1Node01	JDBC provider name	DB2 Universal JDBC Driver Provider (XA)	Description	XA DB2 Universal JDBC Driver-compliant Provider. Datasources created under this provider support the use of XA to perform 2-phase commit processing. Use of driver type 2 on WebSphere Application Server for Z/OS is not supported for datasources created under this provider.	Class path	<pre> \${DB2UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc.jar \${UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc_license_cu.jar \${DB2UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc_license_cisuz.jar</pre>	Native path	E:\lwq\DBDriver\DB2 9.5	Implementation class name	com.ibm.db2.jcc.DB2XADataSource
Options	Values														
Scope	cells:localhostNode01Cell:nodes:n1Node01														
JDBC provider name	DB2 Universal JDBC Driver Provider (XA)														
Description	XA DB2 Universal JDBC Driver-compliant Provider. Datasources created under this provider support the use of XA to perform 2-phase commit processing. Use of driver type 2 on WebSphere Application Server for Z/OS is not supported for datasources created under this provider.														
Class path	<pre> \${DB2UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc.jar \${UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc_license_cu.jar \${DB2UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc_license_cisuz.jar</pre>														
Native path	E:\lwq\DBDriver\DB2 9.5														
Implementation class name	com.ibm.db2.jcc.DB2XADataSource														
<input type="button" value="Previous"/> <input style="background-color: #0070C0; color: white; font-weight: bold; border: 1px solid black; border-radius: 5px; padding: 2px 10px;" type="button" value="Finish"/> <input type="button" value="Cancel"/>															

6. Click the JDBC Provider that you just created.

JDBC providers

JDBC providers

Messages

- ⚠ Changes have been made to your local configuration. You can:
 - [Save](#) directly to the master configuration.
 - [Review](#) changes before saving or discarding.
- ⚠ The server may need to be restarted for these changes to take effect.

JDBC providers

Use this page to edit properties of a JDBC provider. The JDBC provider object encapsulates the specific JDBC driver implementation class for access to the specific vendor database of your environment. Learn more about this task in a [guided activity](#). A guided activity provides a list of task steps and more general information about the topic.

Scope: Cell=localhostNode01Cell, Node=n1Node01

Scope specifies the level at which the resource definition is visible. For detailed information on what scope is and how it works, [see the scope settings help](#)

Preferences



Select	Name	Scope	Description
<input type="checkbox"/>	DB2 Universal JDBC Driver Provider (XA)	Node=n1Node01	XA DB2 Universal JDBC Driver-compliant Provider. Datasources created under this provider support the use of XA to perform 2-phase commit processing. Use of

7. Click **Data sources**, under **Additional Properties**.

JDBC providers

JDBC providers

Messages

- ⚠ Modifying the implementation class name will eliminate the ability to create data sources and data sources version 4 from templates.
- ⚠ Changes have been made to your local configuration. You can:
 - [Save](#) directly to the master configuration.
 - [Review](#) changes before saving or discarding.
- ⚠ The server may need to be restarted for these changes to take effect.

[JDBC providers > DB2 Universal JDBC Driver Provider \(XA\)](#)

Use this page to edit properties of a JDBC provider. The JDBC provider object encapsulates the specific JDBC driver implementation class for access to the specific vendor database of your environment.

General Properties

* Scope

cells:localhostNode01Cell:nodes:n1Node01

* Name

[DB2 Universal JDBC Driver Provider \(XA\)](#)

Description

XA DB2 Universal JDBC Driver-compliant Provider. Datasources created under this provider support the use of XA to perform 2-phase commit processing. Use of

Additional Properties

■ Data sources

■ Data sour

Use this page to edit the settings c
Application associated with your selected JDBC
source object supplies your applica
accessing the database.

8. Click **New**.

JDBC providers

Messages

- Changes have been made to your local configuration. You can:
 - [Save](#) directly to the master configuration.
 - [Review](#) changes before saving or discarding.
- The server may need to be restarted for these changes to take effect.

JDBC providers > DB2 Universal JDBC Driver Provider (XA) > Data sources

Use this page to edit the settings of a data source that is associated with your selected JDBC provider. The data source object supplies your application with connections for accessing the database. Learn more about this task in a [guided activity](#). A guided activity provides a list of task steps and more general information about the topic.

Preferences

New	Delete	Test connection	Manage state...
Select	Name ▾	JNDI name ▾	Scope ▾
None			Provider ▾
			Description ▾
			Category ▾
	Total 0		

9. Enter **jdbc/DB2XADS** for JNDI name.

10. Under **Component-managed authentication alias and XA recovery authentication alias**, select the name of the authentication alias you previously created from the drop-down list.
Click **Next**.

Create a data source

→ Step 1: Enter basic data source information Step 2: Enter database specific properties for the data source Step 3: Summary	Enter basic data source information <p>Set the basic configuration values of a data source for association with your JDBC provider. A data source supplies the physical connections between the application server and the database.</p> <p>Requirement: Use the Data sources (WebSphere(R) Application Server V4) console pages if your applications are based on the Enterprise JavaBeans (TM) (EJB) 1.0 specification or the Java(TM) Servlet 2.2 specification.</p> <p>Scope cells:localhostNode01Cell:nodes:nlNode01</p> <p>JDBC provider name DB2 Universal JDBC Driver Provider (XA)</p> <p>* Data source name DB2 Universal JDBC Driver XA DataSource</p> <p>* JNDI name jdbc/DB2XADS</p> <p>Component-managed authentication alias and XA recovery authentication alias</p> <p>Select a component-managed authentication alias. The selected authentication alias will also be set as the XA recovery authentication alias if your JDBC Provider supports XA. If you choose to create a new J2C authentication alias, the wizard will be canceled.</p> <p>Alias_DB2</p>
<input type="button" value="Next"/> <input type="button" value="Cancel"/>	

11. Enter the values below for **Database name** and **Server name**.
Click **Next**.

Integrated Solutions Console Welcome Help | Logout

JDBC providers

Create a data source

Step 1: Enter basic data source information
→ Step 2: Enter database specific properties for the data source
Step 3: Summary

Enter database specific properties for the data source

Set these database-specific properties, which are required by the database vendor JDBC driver to support the connections that are managed through this data source.

* Database name: sampledb

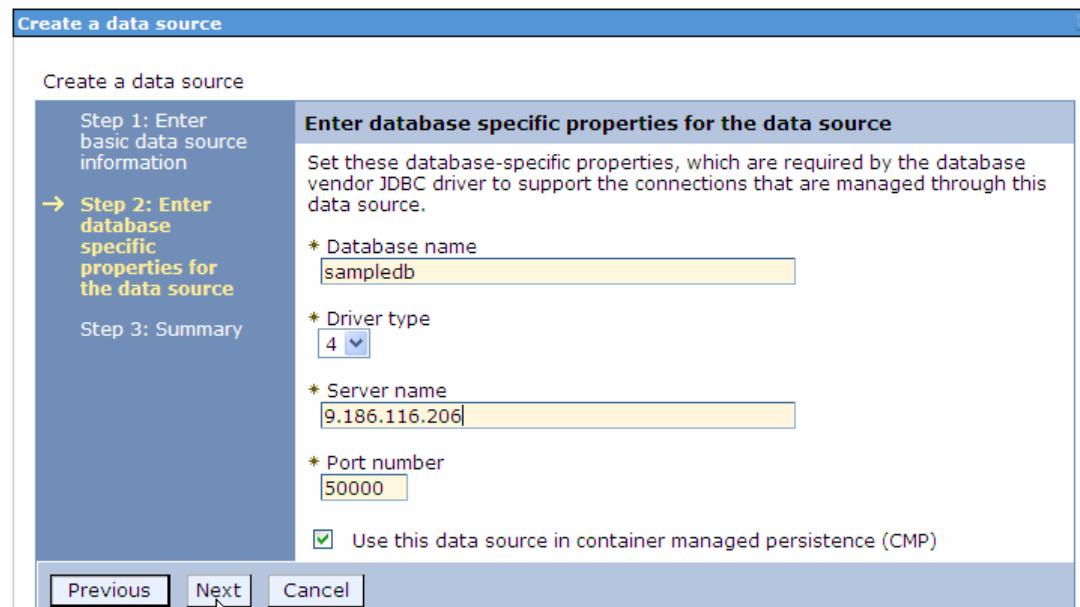
* Driver type: 4

* Server name: 9.186.116.206

* Port number: 50000

Use this data source in container managed persistence (CMP)

Previous Next Cancel



12. Click **Finish**.

Create a data source

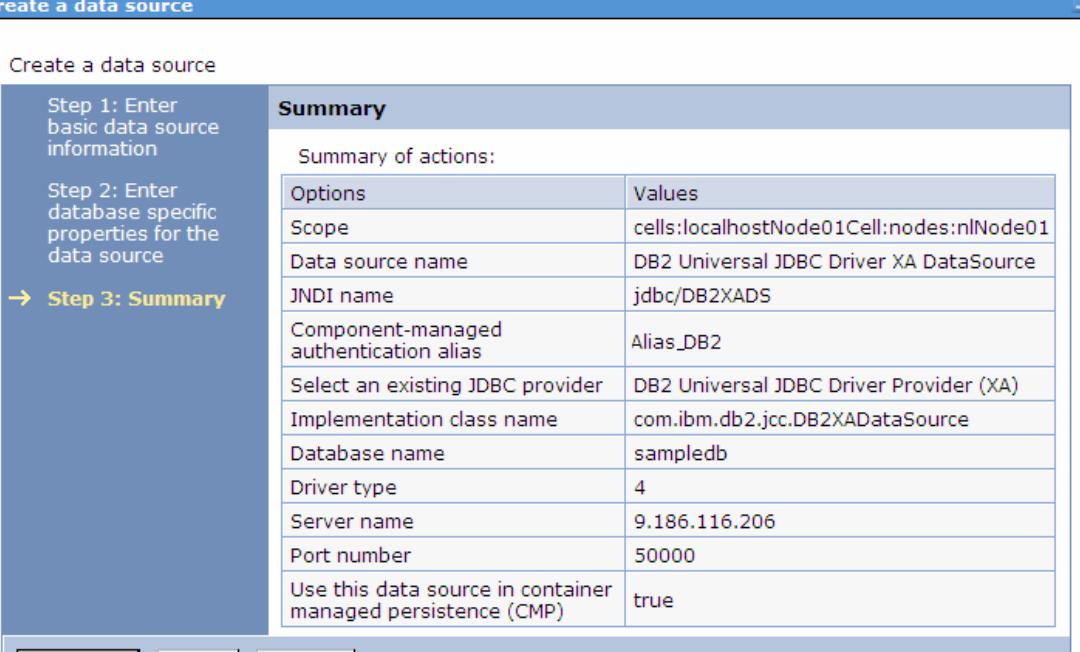
Step 1: Enter basic data source information
Step 2: Enter database specific properties for the data source
→ Step 3: Summary

Summary

Summary of actions:

Options	Values
Scope	cells:localhostNode01Cell:nodes:nlNode01
Data source name	DB2 Universal JDBC Driver XA DataSource
JNDI name	jdbc/DB2XADS
Component-managed authentication alias	Alias_DB2
Select an existing JDBC provider	DB2 Universal JDBC Driver Provider (XA)
Implementation class name	com.ibm.db2.jcc.DB2XADataSource
Database name	sampledb
Driver type	4
Server name	9.186.116.206
Port number	50000
Use this data source in container managed persistence (CMP)	true

Previous Finish Cancel



13. In the Messages area, click **Save**. This will save changes made to the local configuration onto the master configuration.

Messages

⚠ Changes have been made to your local configuration. You can:

- [Save](#) directly to the master configuration.
- [Revert](#) changes before saving or discarding.

⚠ The server may need to be restarted for these changes to take effect.

[JDBC providers > DB2 Universal JDBC Driver Provider \(XA\) > Data sources](#)

Use this page to edit the settings of a data source that is associated with your selected JDBC provider. The data source object supplies your application with connections for accessing the database. Learn more about this task in a [guided activity](#). A guided activity provides a list of task steps and more general information about the topic.

 Preferences

[New](#) [Delete](#) [Test connection](#) [Manage state...](#)



Select	Name	JNDI name	Scope	Provider	Description	Category
<input type="checkbox"/>	DB2 Universal JDBC Driver XA DataSource	jdbc/DB2XADS	Node=nlNode01	DB2 Universal JDBC Driver Provider (XA)	DB2 Universal Driver Datasource	

Total 1

14. Select the check box next to the Data source you just created. Click **Test Connection**.

[JDBC providers > DB2 Universal JDBC Driver Provider \(XA\) > Data sources](#)

Use this page to edit the settings of a data source that is associated with your selected JDBC provider. The data source object supplies your application with connections for accessing the database. Learn more about this task in a [guided activity](#). A guided activity provides a list of task steps and more general information about the topic.

 Preferences

[New](#) [Delete](#) [Test connection](#) [Manage state...](#)



Select	Name	JNDI name	Scope	Provider	Description	Category
<input checked="" type="checkbox"/>	DB2 Universal JDBC Driver XA DataSource	jdbc/DB2XADS	Node=nlNode01	DB2 Universal JDBC Driver Provider (XA)	DB2 Universal Driver Datasource	

Total 1

The connection test should succeed as indicated by the message shown in the figure below. For troubleshooting issues while testing the connection, see the Troubleshooting section.

Messages

The test connection operation for data source DB2 Universal JDBC Driver XA **DataSource** on server server1 at node nlNode01 was successful.

JDBC providers > DB2 Universal JDBC Driver Provider (XA) > Data sources

Use this page to edit the settings of a data source that is associated with your selected JDBC provider. The data source object supplies your application with connections for accessing the database. Learn more about this task in a [guided activity](#). A guided activity provides a list of task steps and more general information about the topic.

Preferences

New	Delete	Test connection	Manage state...
Select	Name ▾	JNDI name ▾	Scope ▾
<input type="checkbox"/> DB2 Universal JDBC Driver XA DataSource	jdbc/DB2XADS	Node=nlNode01	DB2 Universal JDBC Driver Provider (XA)
DB2 Universal JDBC Driver Datasource			
Total 1			

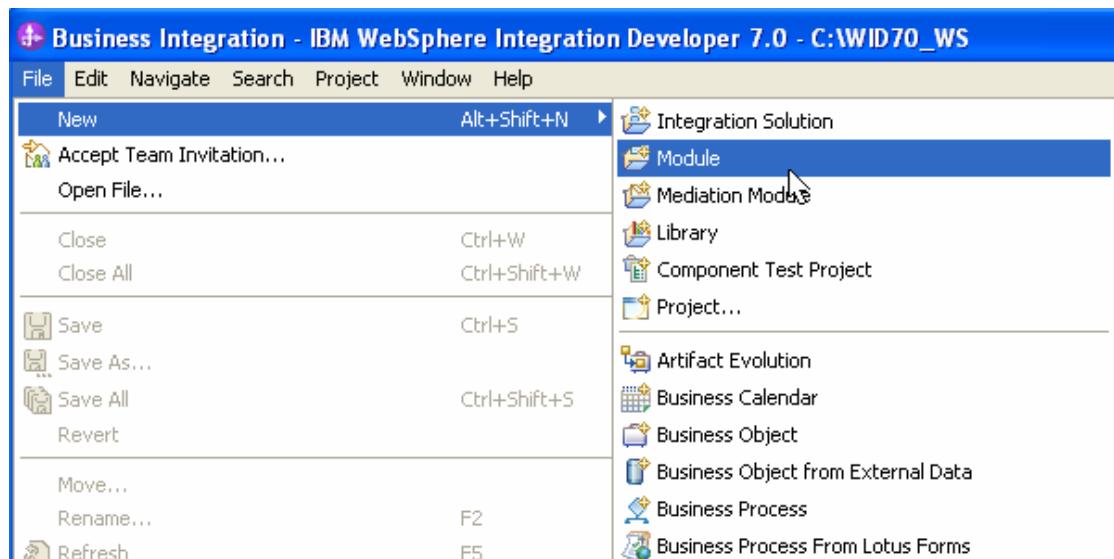
15. Close the **Administrative Console** tab.

Configure the adapter for outbound processing

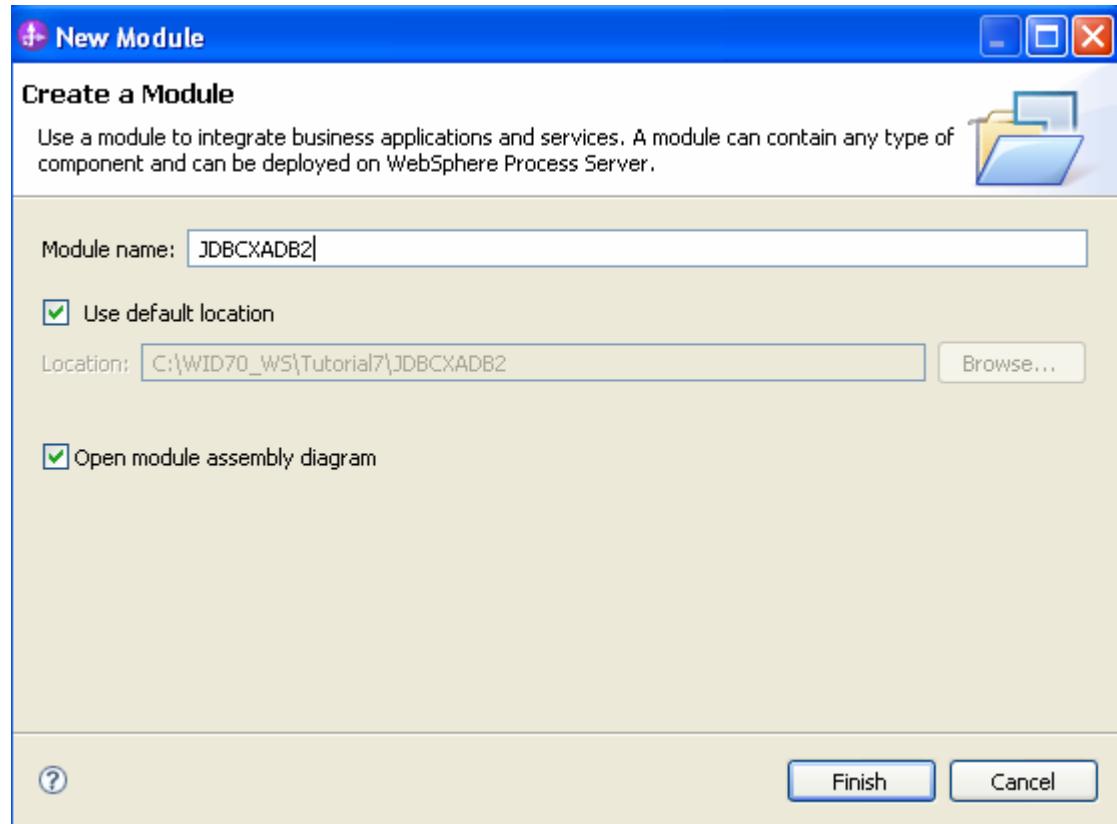
Run the external service wizard to specify business objects, services, and configuration details.

Configure the adapter for outbound processing

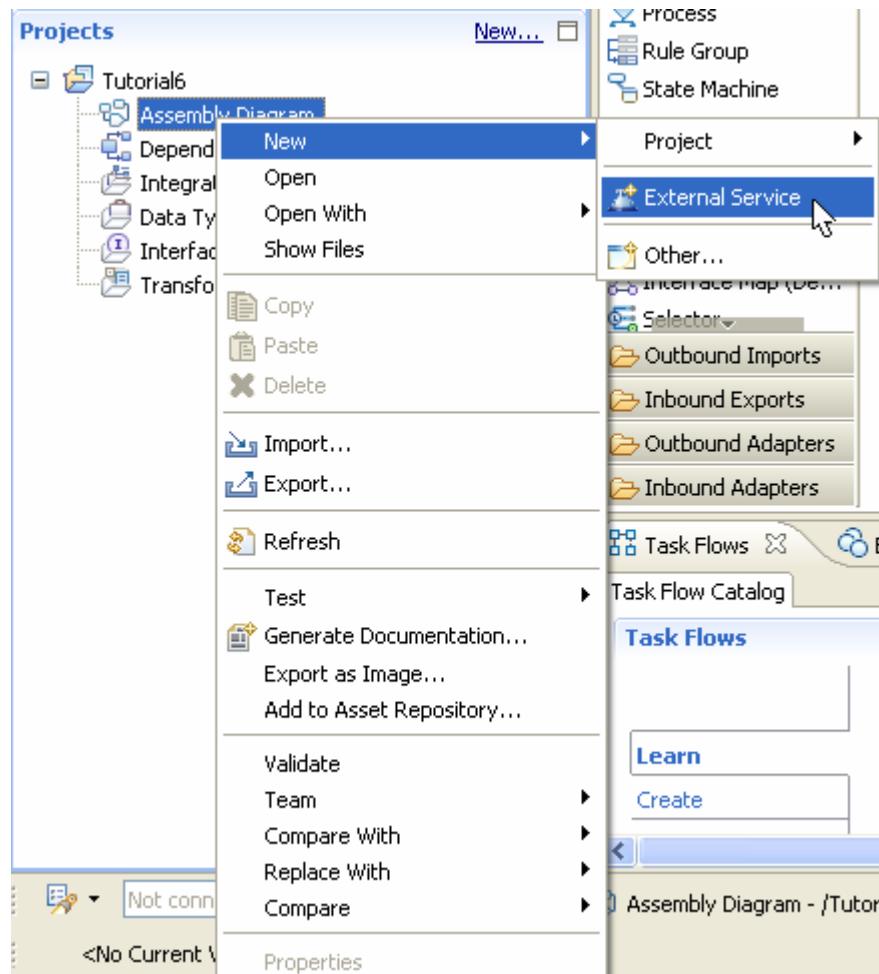
1. Switch to the Business Integration perspective in WebSphere Integration Developer.
2. Select **File->New->Module** to create a Module project.



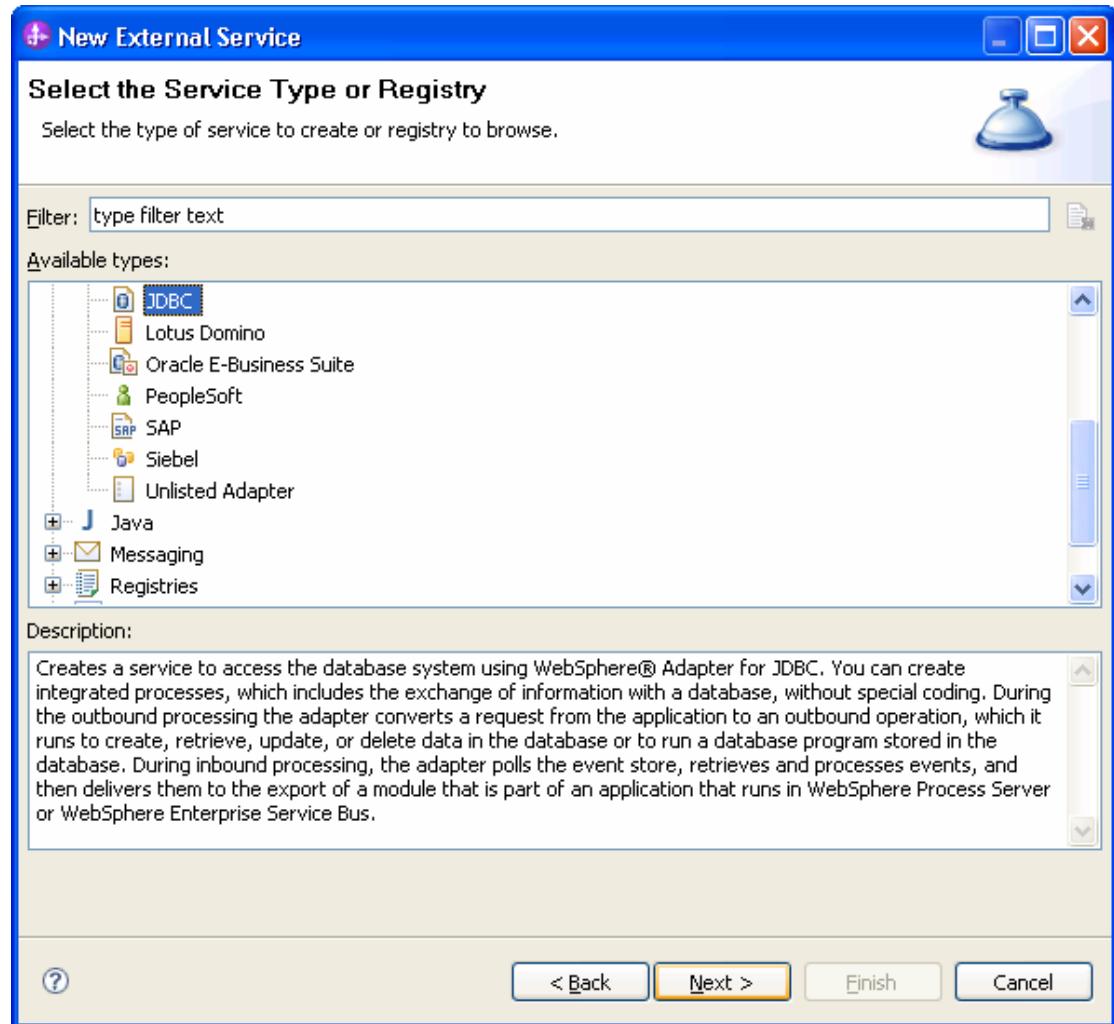
3. Specify the module name as **JDBCXADB2**, and click **Finish**.



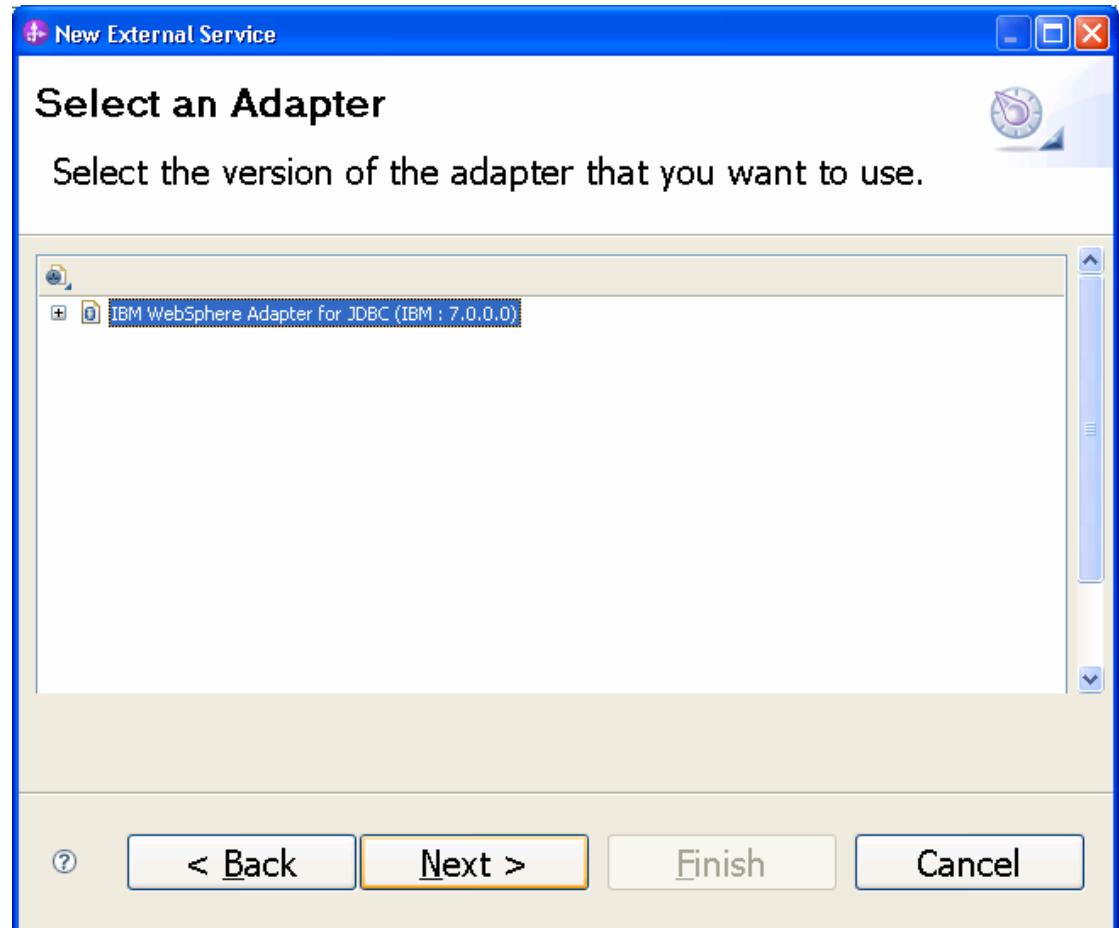
4. Right-click **JDBCXADB2->Assembly Diagram**, select **New->External Service**.



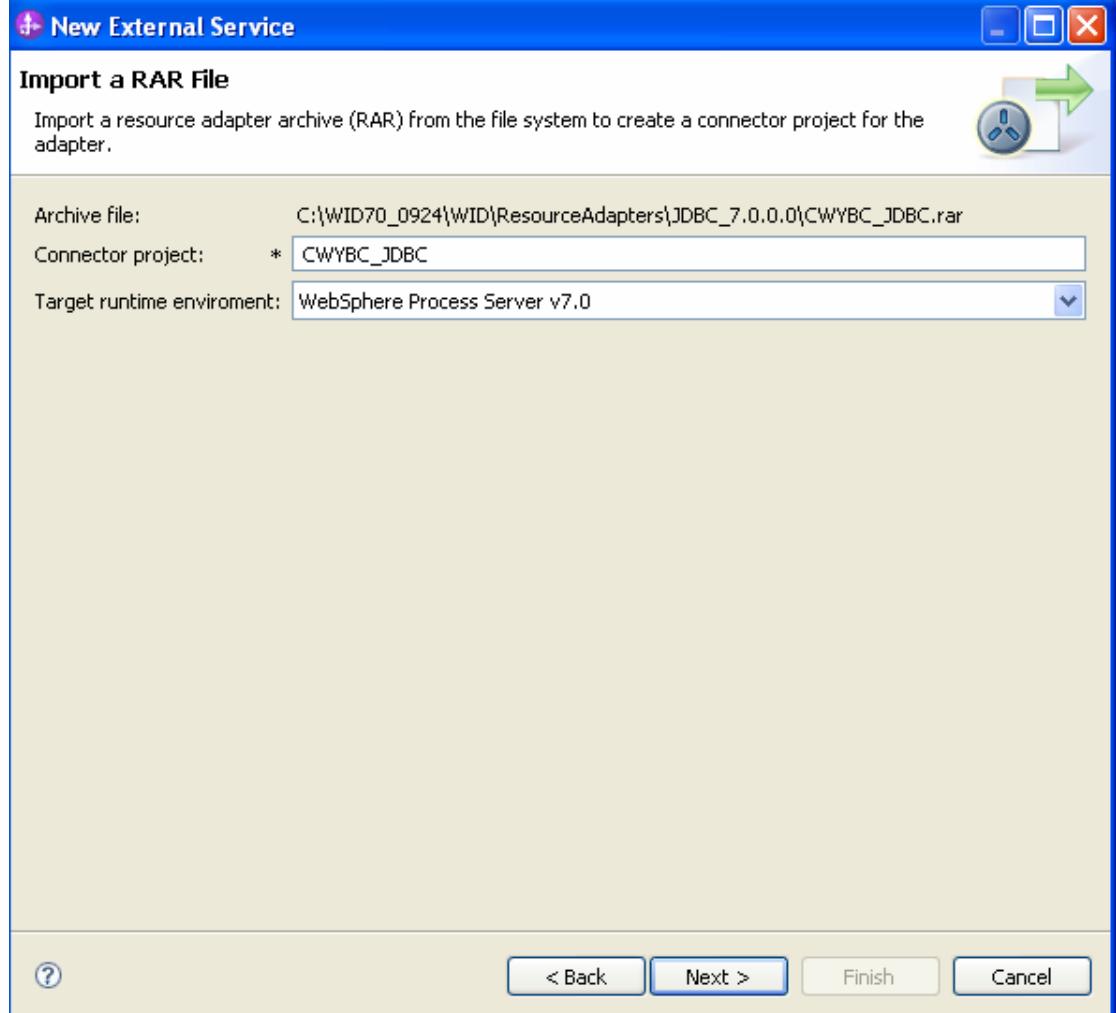
5. Select JDBC, and click **Next**.



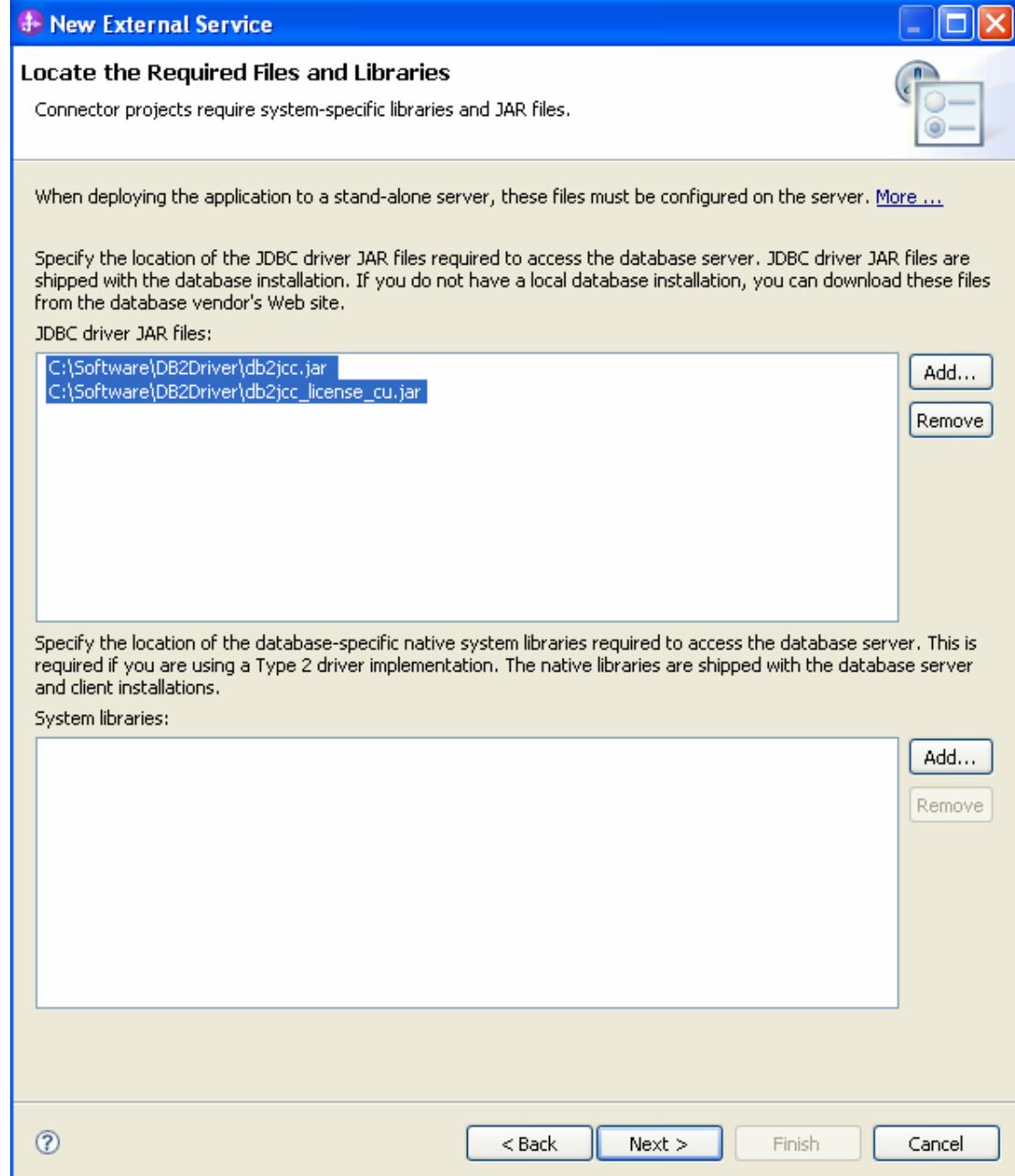
6. Select **IBM WebSphere Adapter for JDBC (IBM: 7.0.0.0)**. Click **Next**.



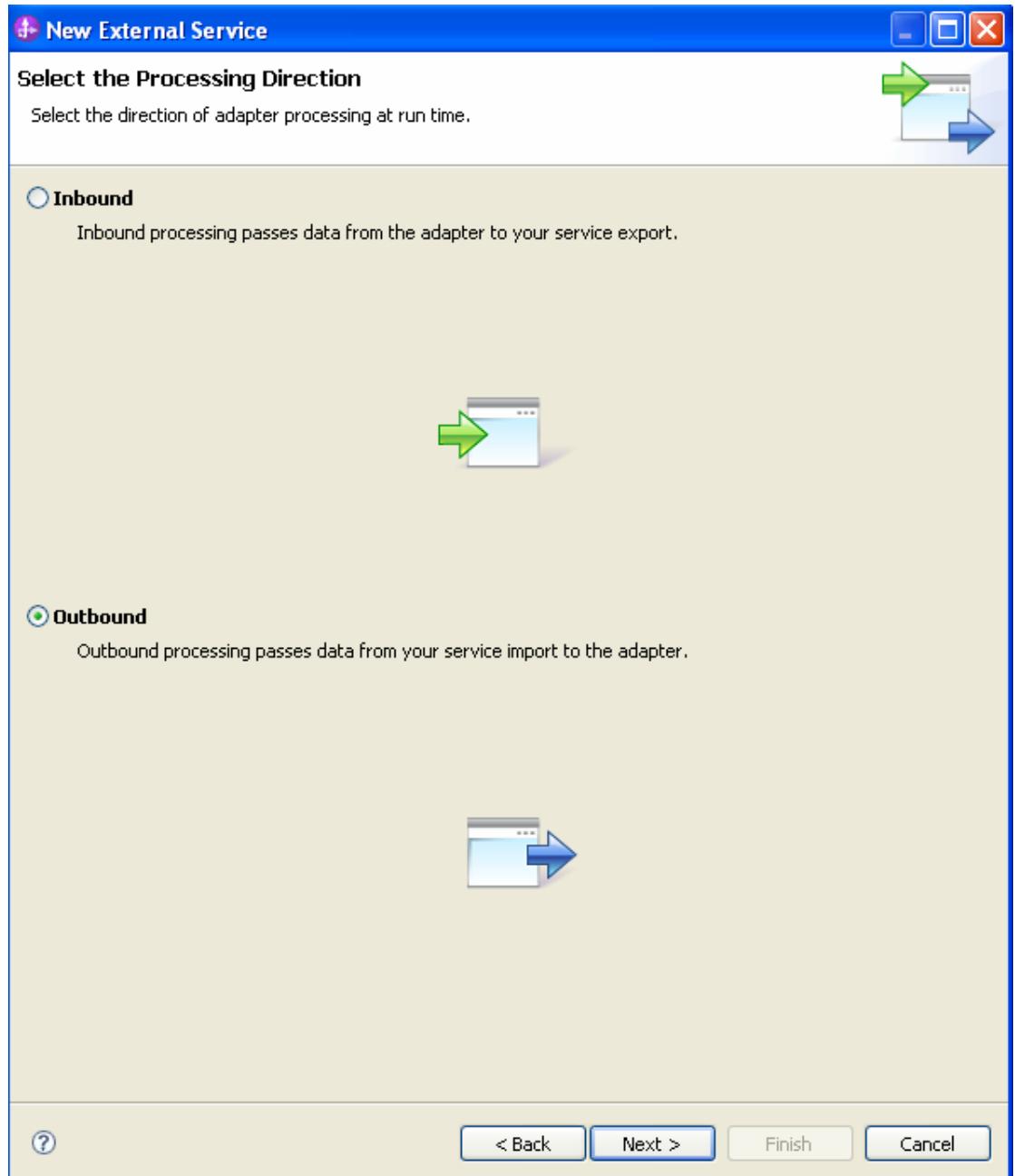
7. In the **Target Runtime environment** field, select the appropriate runtime and click **Next**.



8. Click **Add** to add the JDBC driver jar to the class path, and click **Next**.



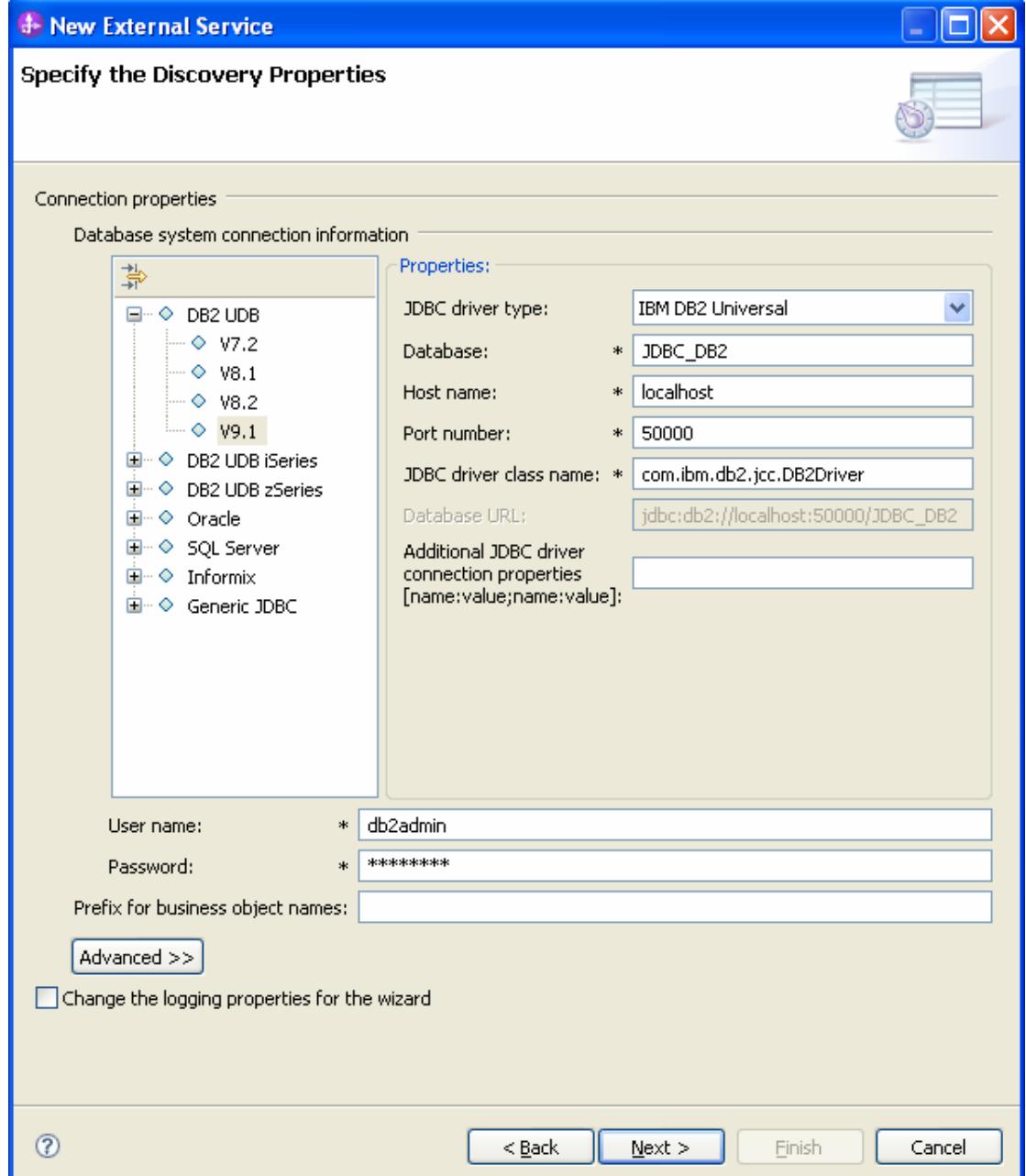
9. Select **Outbound** and click **Next**.



Set connection properties for the external service wizard

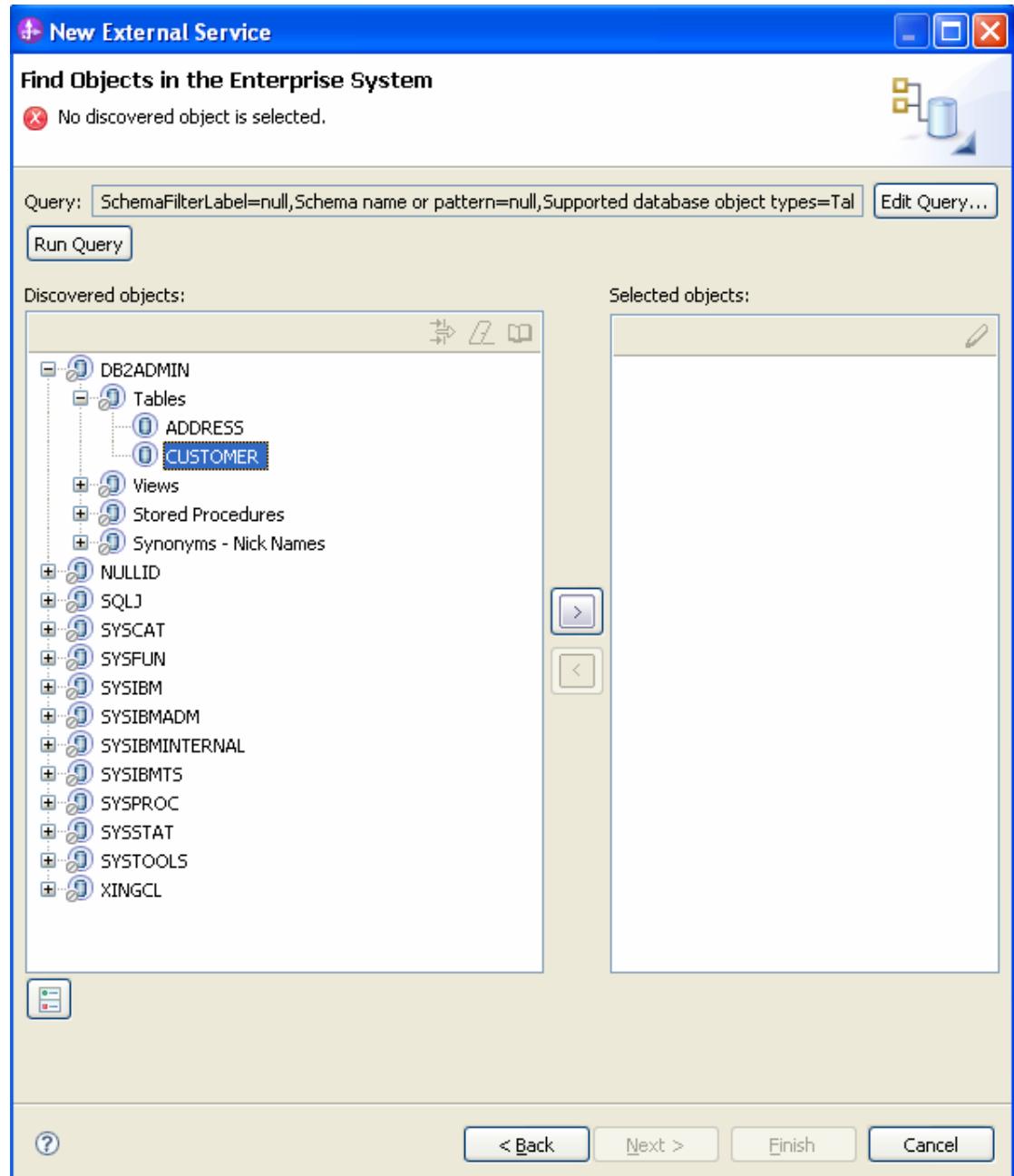
To connect to the database:

1. Expand the **DB2** node in the **Database system connection information** area and select appropriate version,
2. Enter values in the **Database**, **Host name**, **Port number**, **User name** and **Password** fields, and click **Next**.

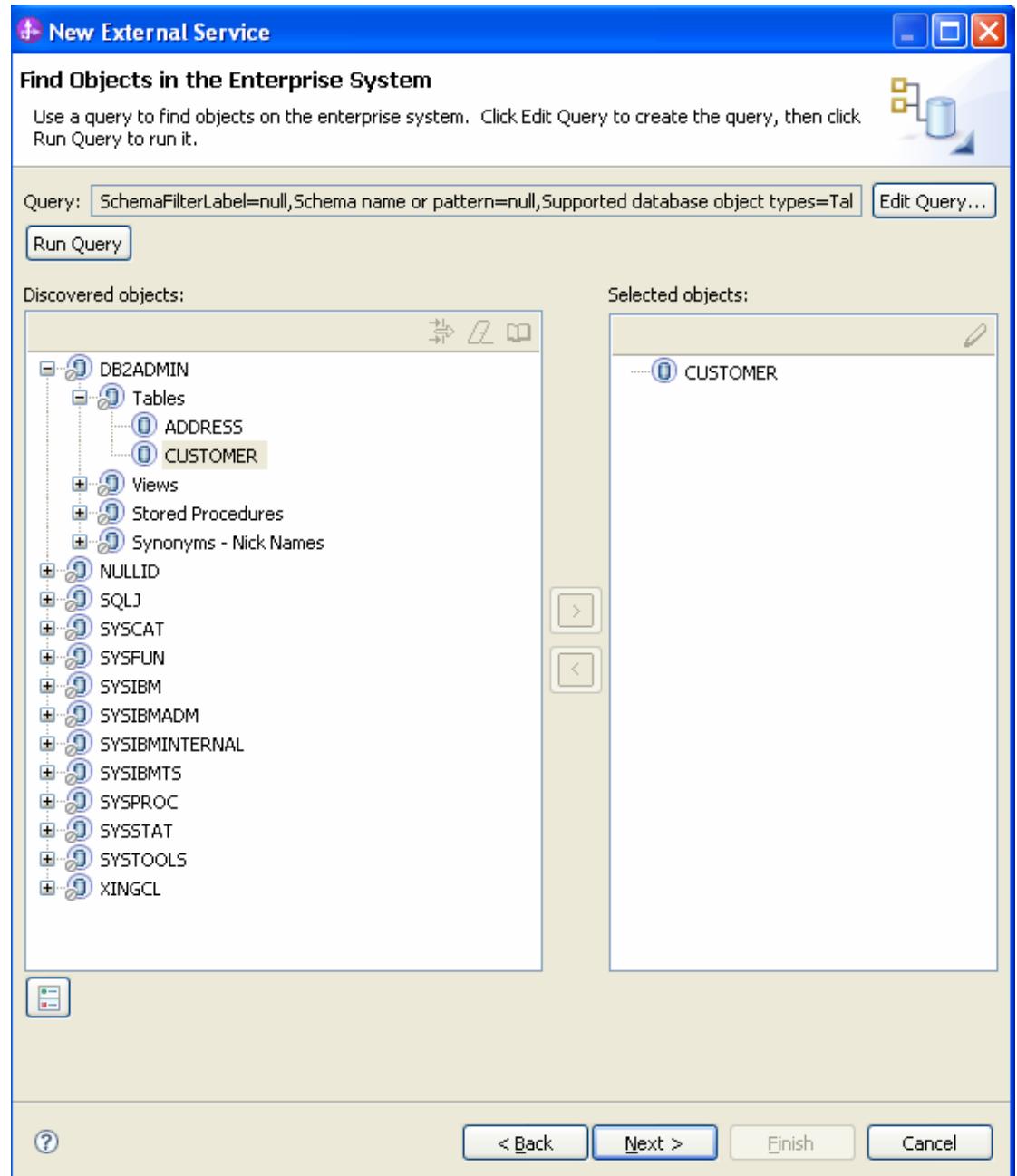


Select the business objects and services to be used with the adapter

1. Click **Run Query** to list the tables, stored procedures, views, and synonyms for each schema in the database.
2. Select **DB2ADMIN->Tables-> CUSTOMER** and click the **> (Add)**. The CUSTOMER table is added to the **Selected objects** list.



3. Click **Next**.



Generate business object definitions and related artifacts

1. In the Specify Composite Properties window, accept the default settings and click **Next**.

 New External Service

Specify Composite Properties

Specify properties that apply to all selected objects.

Operations for selected business objects
Operations for these functions will be added to the service interface.*

Create
Update
Delete
Retrieve
RetrieveAll
ApplyChanges
Exists

Add...
Remove

Create and configure user-defined wrapper objects

Wrapper object names:

Add...
Remove

Return all records for RetrieveAll operation

Maximum records for RetrieveAll operation:

Business object namespace:

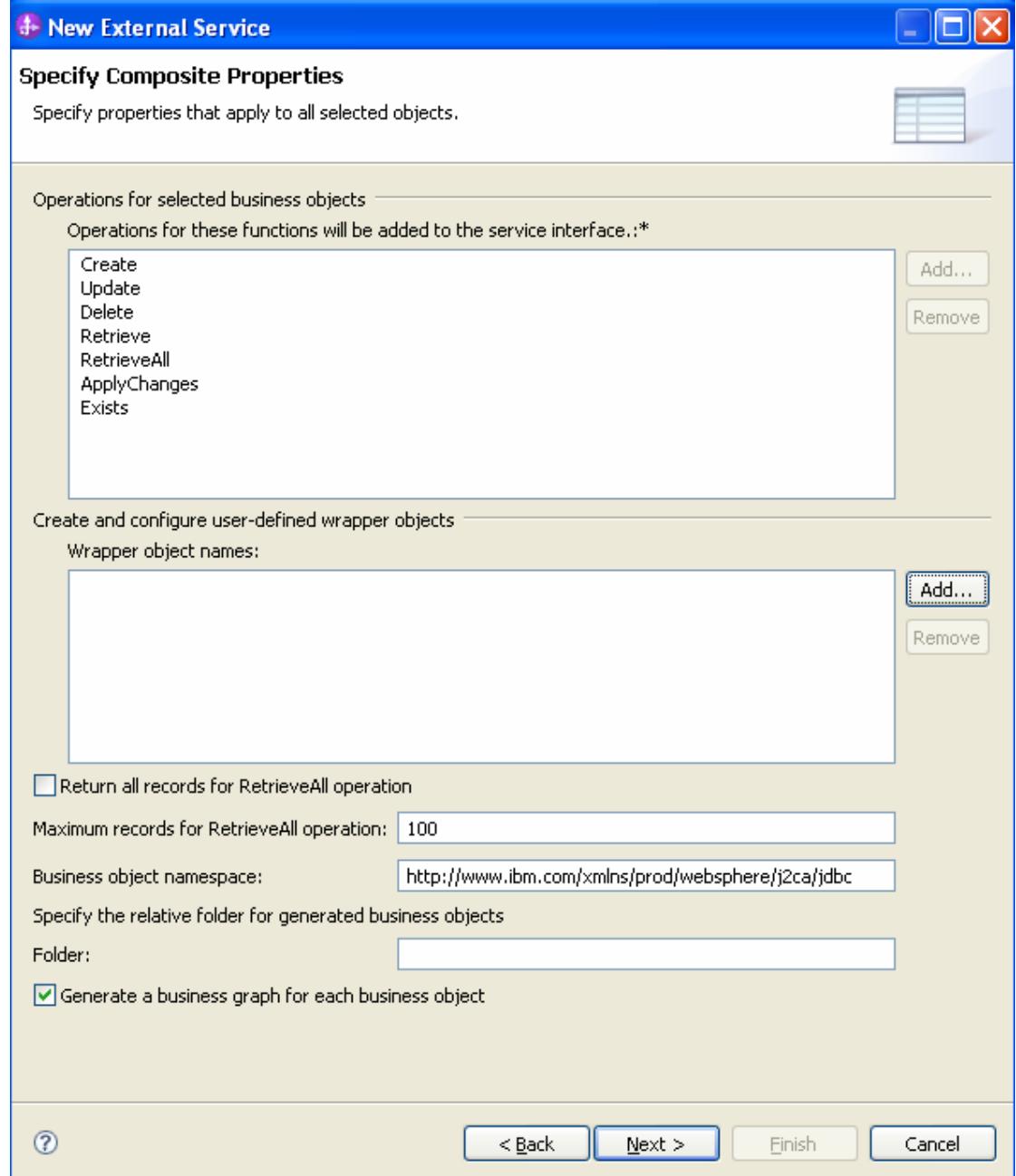
Specify the relative folder for generated business objects

Folder:

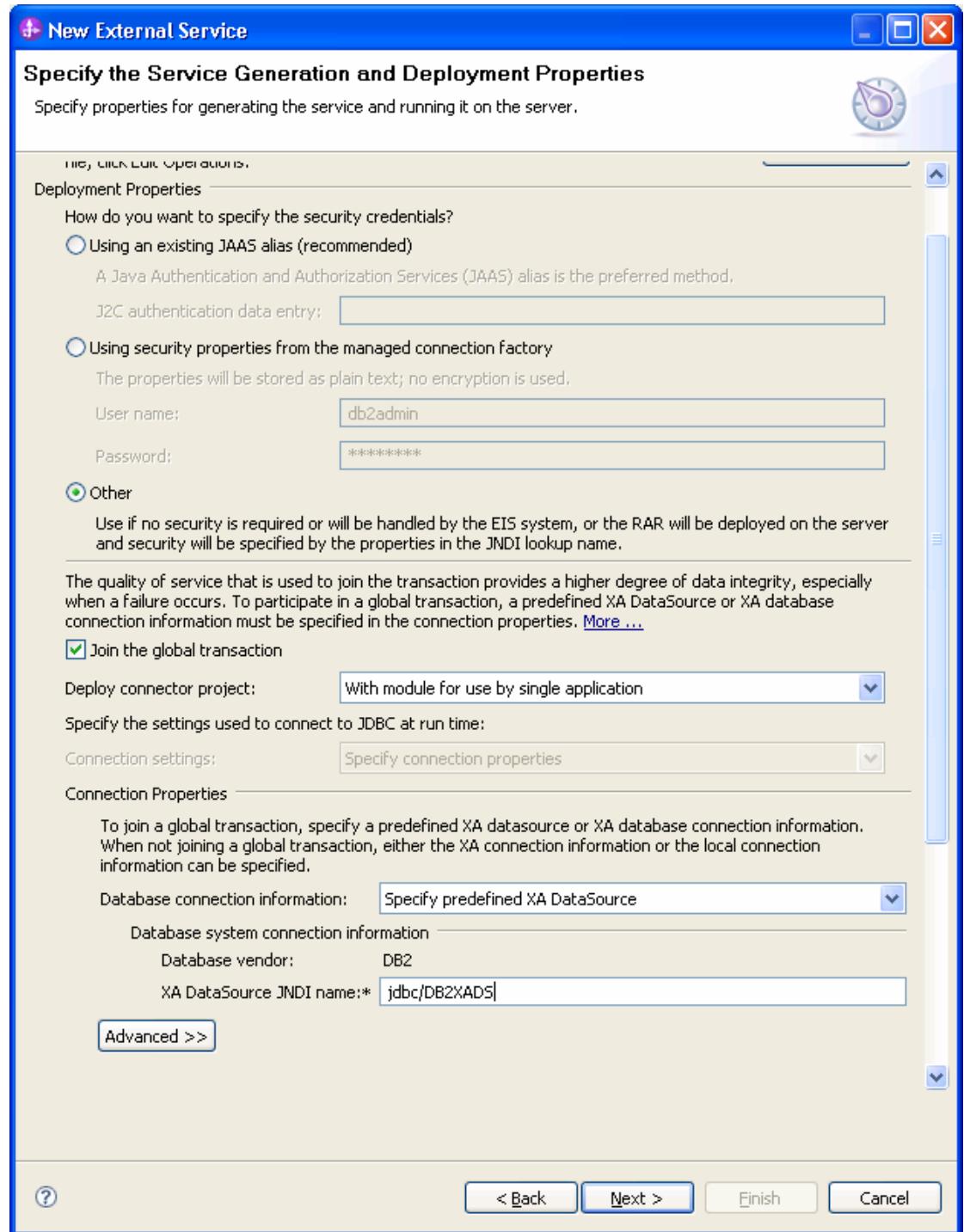
Generate a business graph for each business object

?

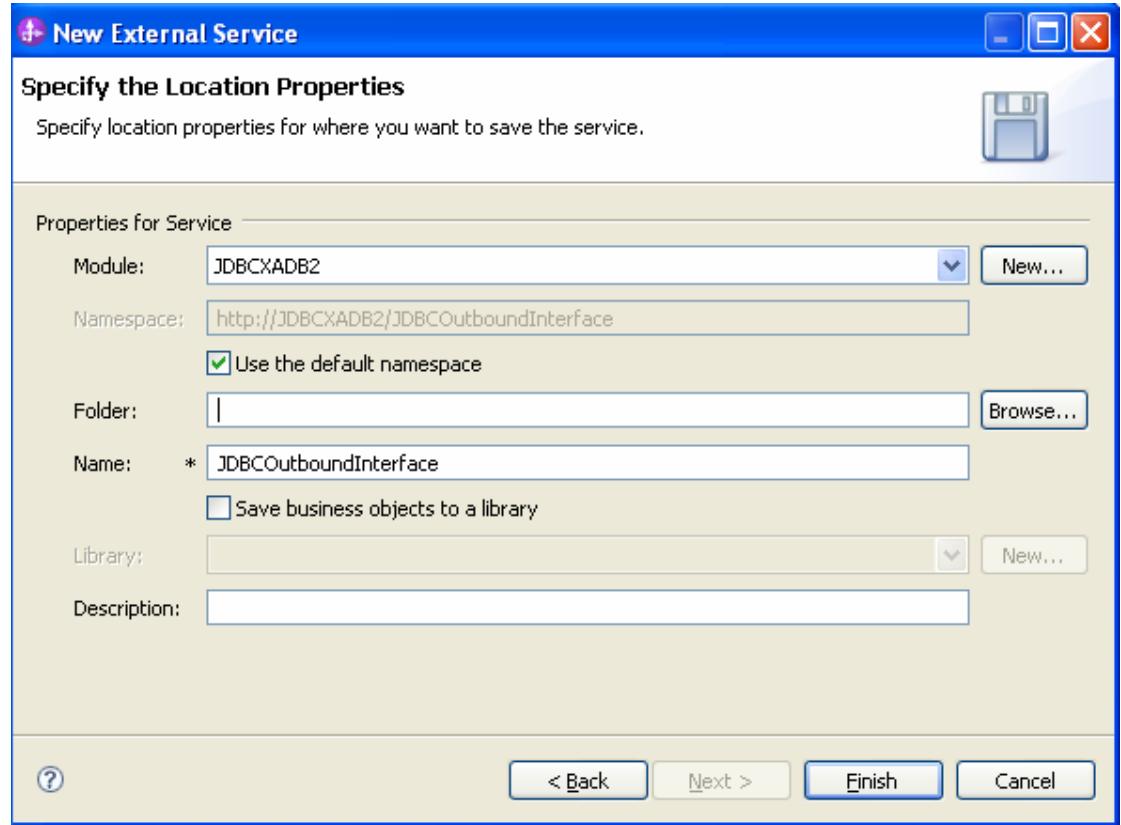
< Back Finish Cancel



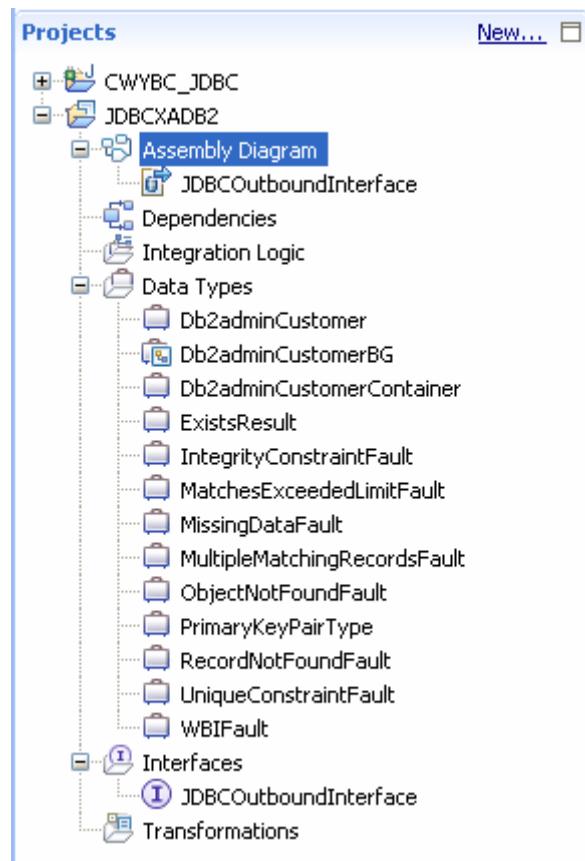
2. Select the security credential type as **Other**. Specify the **XA DataSource JNDI name** property as **jdbc/DB2XADS**. Click **Next**.



3. Click **Finish**.

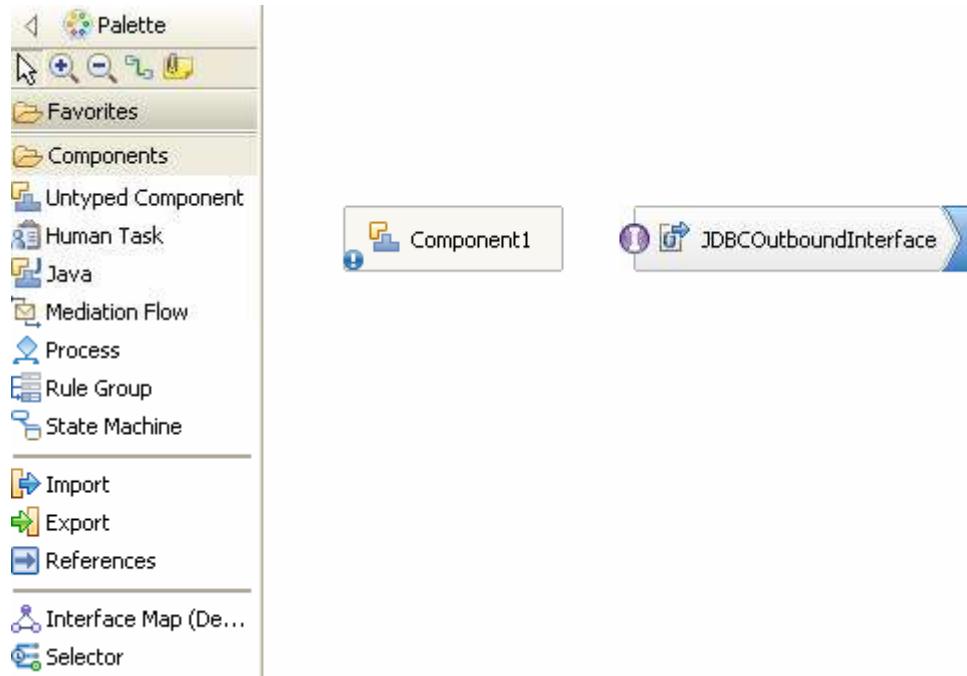


Verify the results shown below.

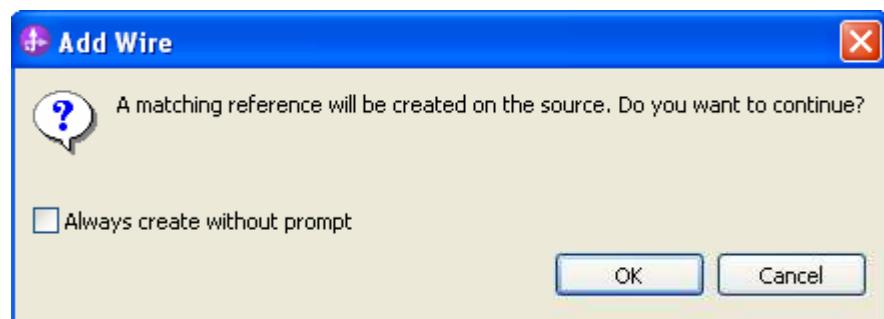


Set up the components to be part of the XA environment

1. In the **Business Integration** tab, under **JDBCXADB2** double-click **Assembly Diagram** to open it.
2. In the Palette, expand **Components** and drag **Untyped Component** to Assembly Diagram editor, and name it as **Component1**.

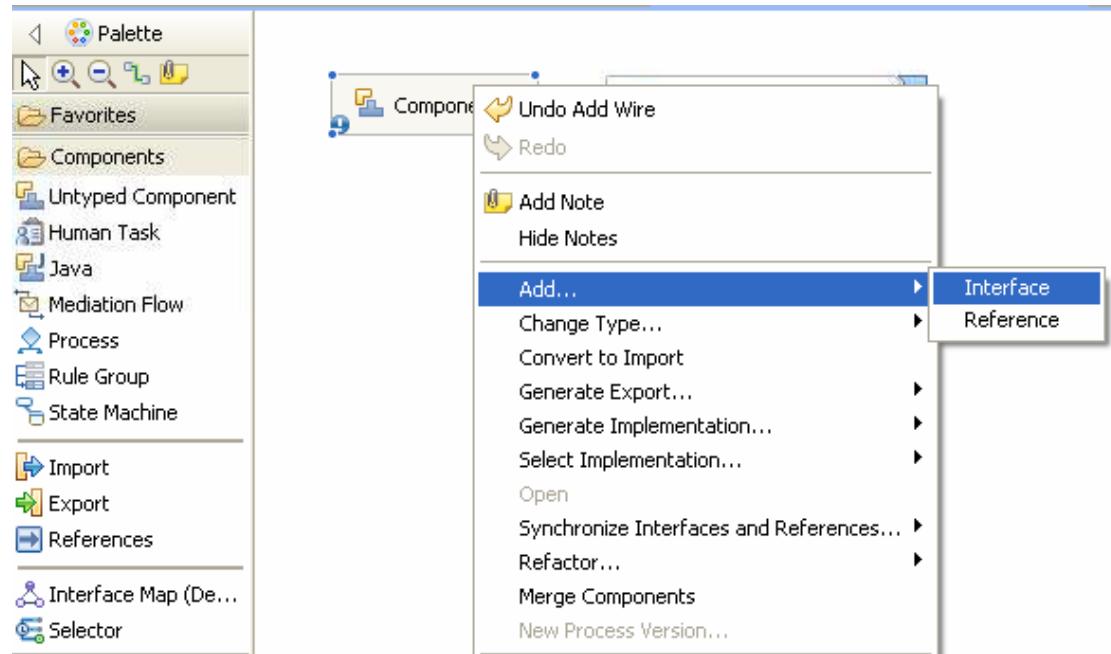


3. Wire **Component1** to **JDBCOutboundInterface**. In the Add Wire message window, click **OK**.

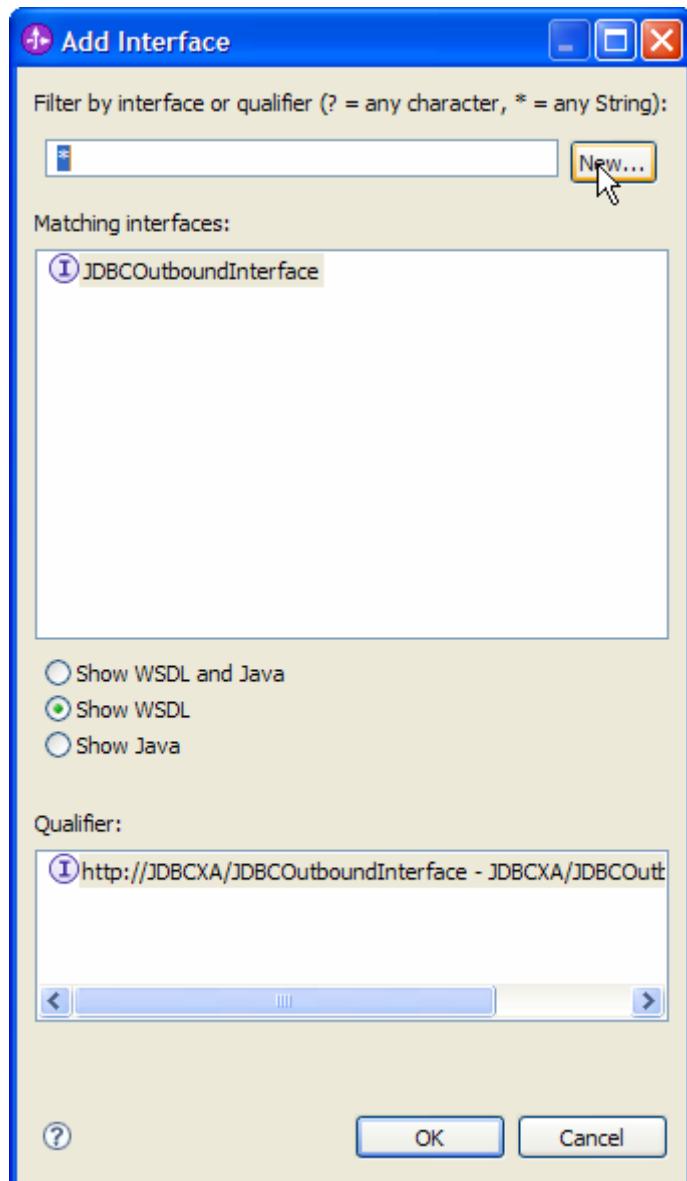


4. Right-click **Component1** and select **Add > Interface**.

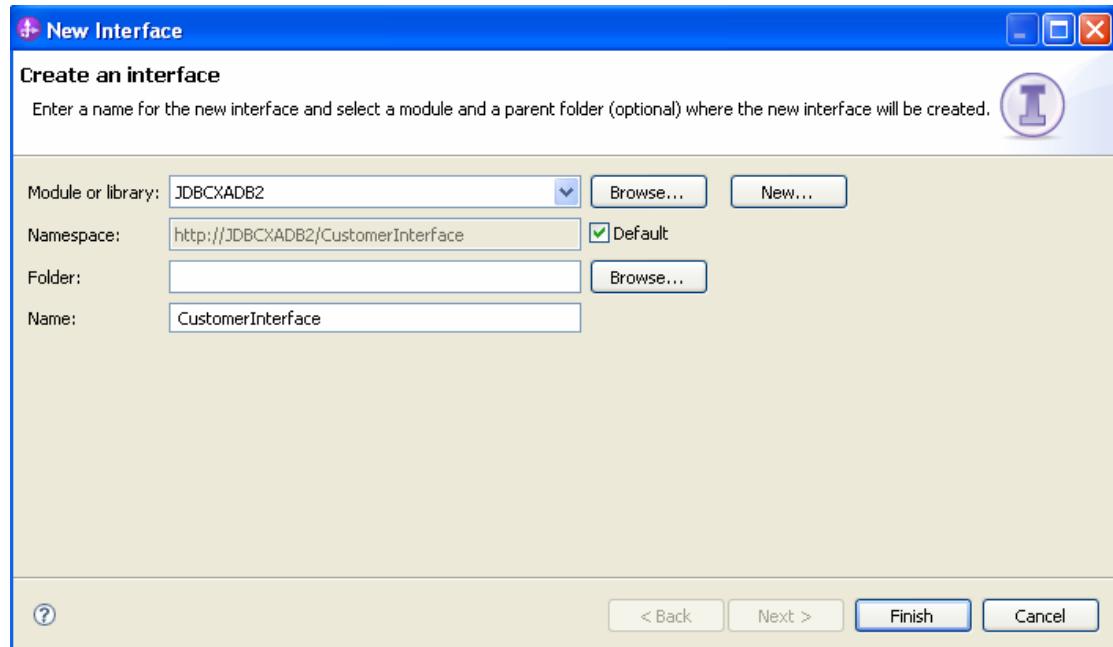
WebSphere software



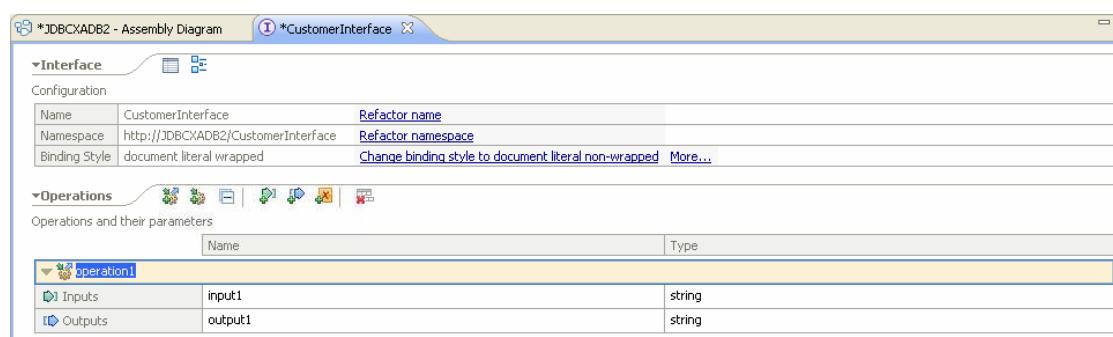
5. In the Add Interface window, click **New**.



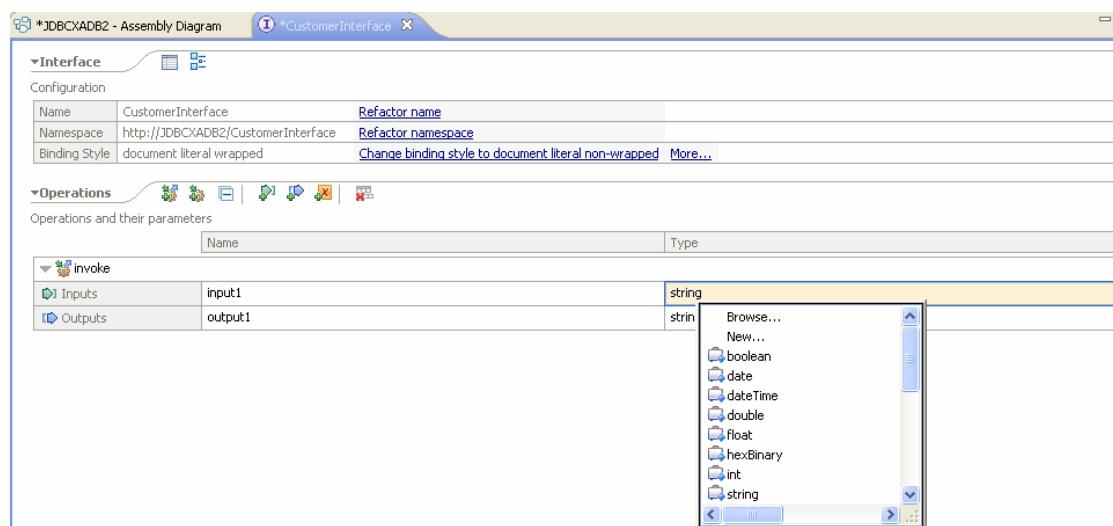
6. Enter **CustomerInterface** in the **Name** field. Click **Finish**.



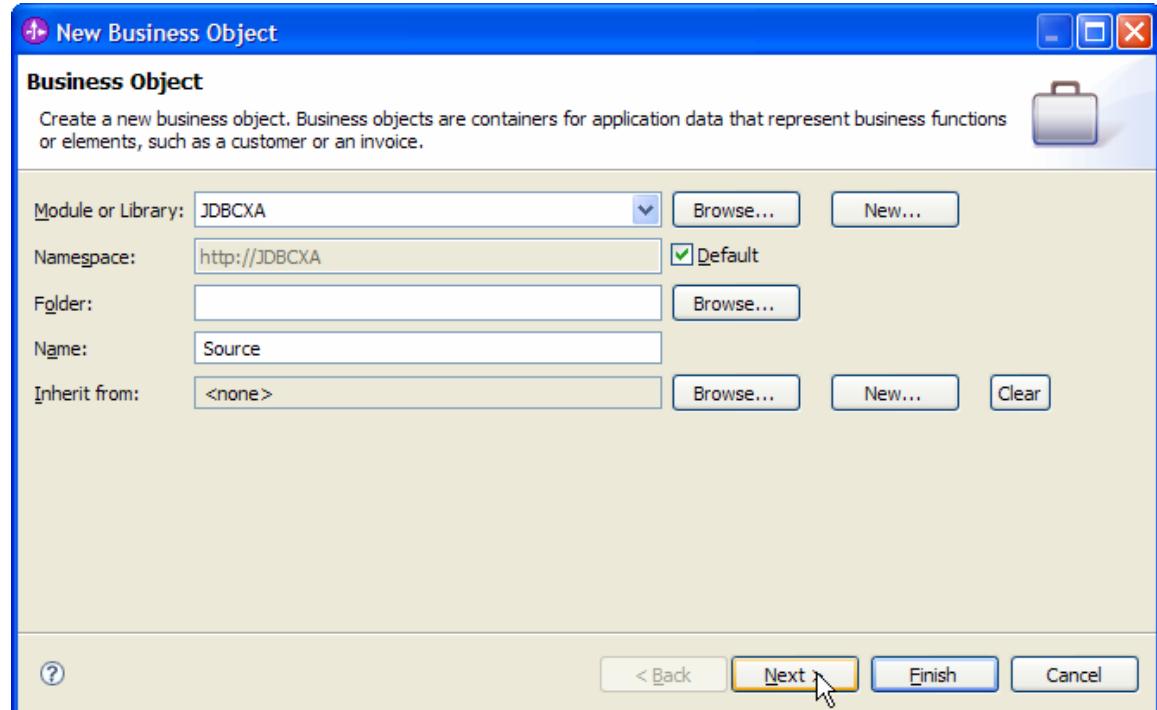
7. Click to add a new operation for **CustomerInterface** interface.



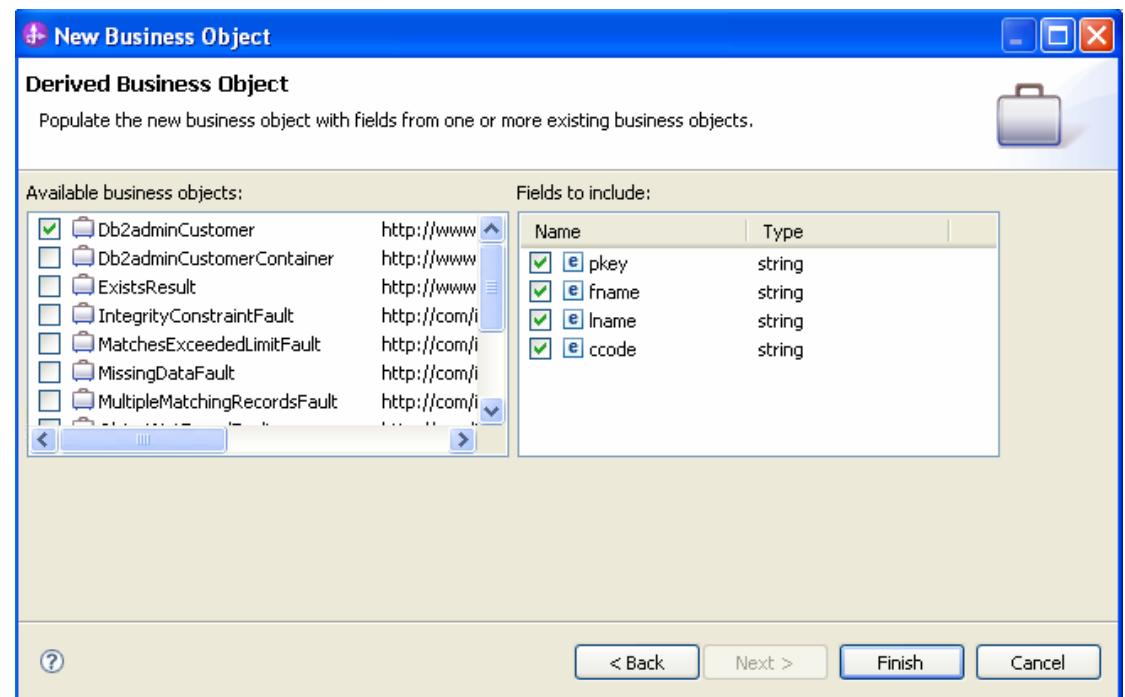
8. Rename the operation name to **invoke**. Click Type for Inputs parameter, and select **New**.



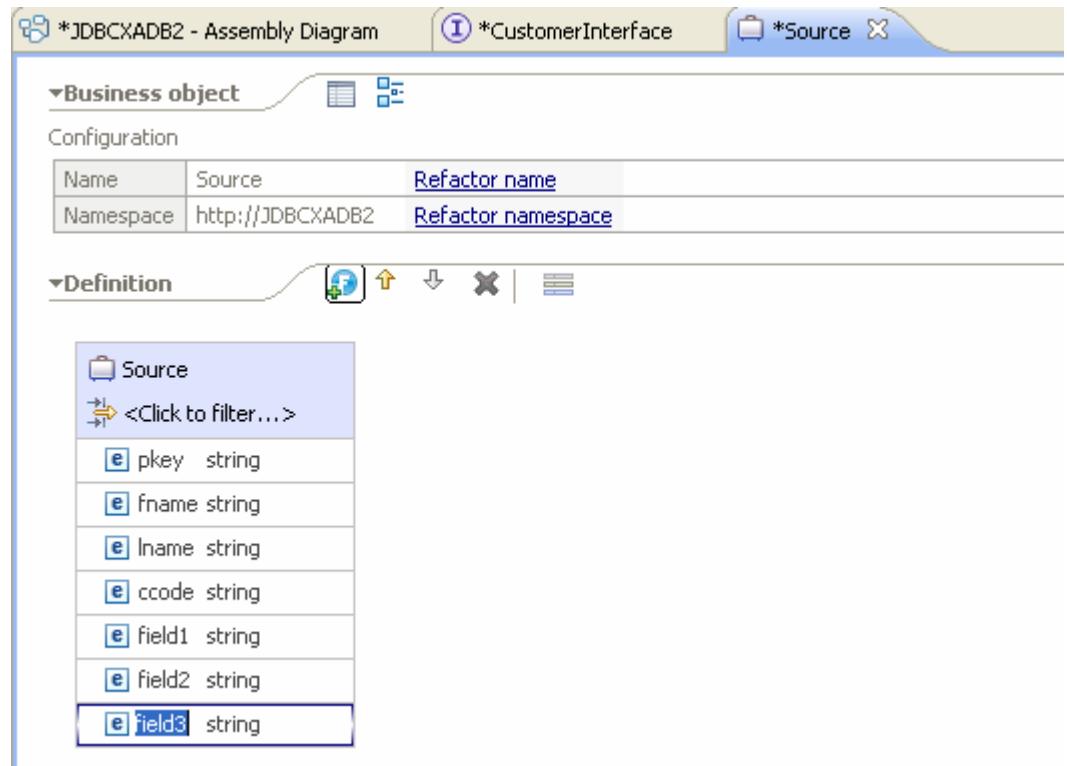
9. In the **New Business Object** window, enter **Source** in the **Name** field. Click **Next**.



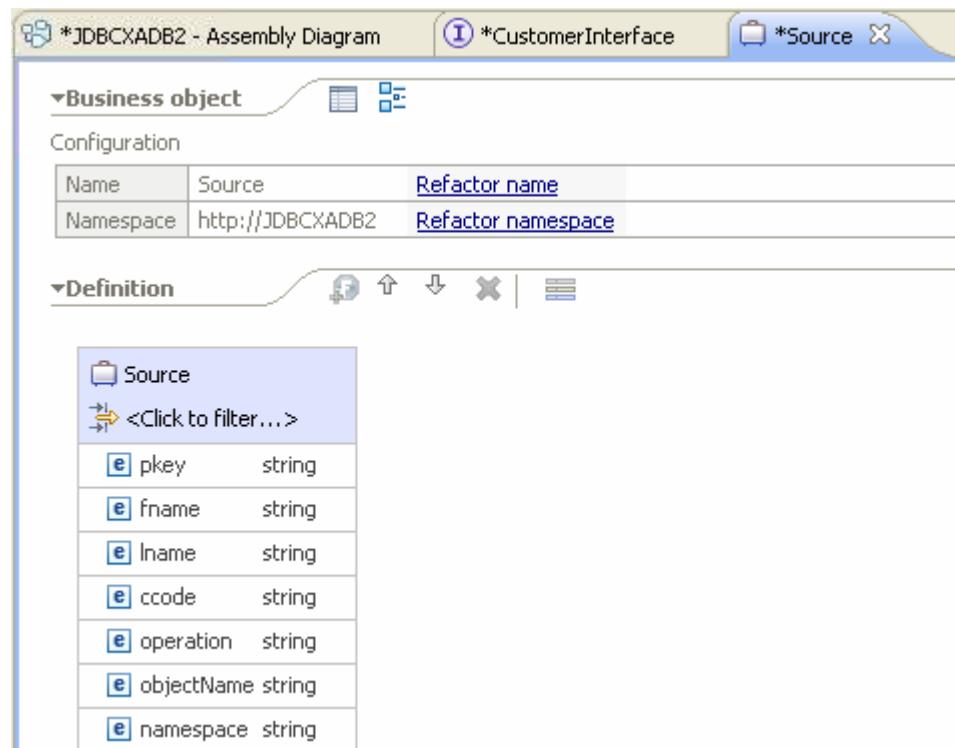
10. From the list of **Available business objects**, select the **Db2adminCustomer** to add all of the Customer business objects's attributes to the Source business object. Click **Finish**.



11. In the Business Object editor, click to add three new fields for **Source** business object.



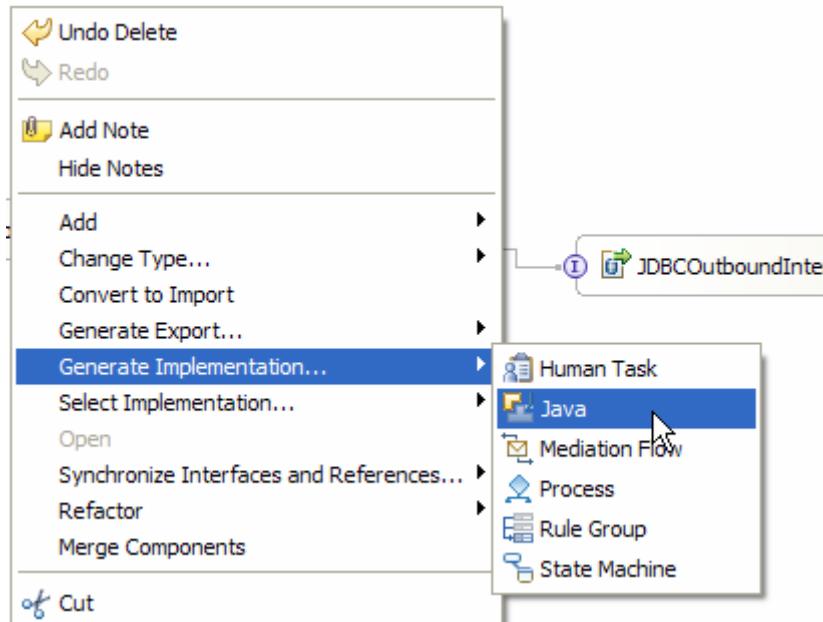
12. Then rename these three new fields as **operation**, **objectName** and **namespace**.



13. Select **File->Save All** to save all the changes.

14. Right click on **Component1** in the Assembly Diagram and select **Generate implementation... -> Java**.

WebSphere software



15. In the **Generate Implementation** window, select **default package** and click **OK**.
16. In the text editor of Component1Impl.java file, add the following **imports**.

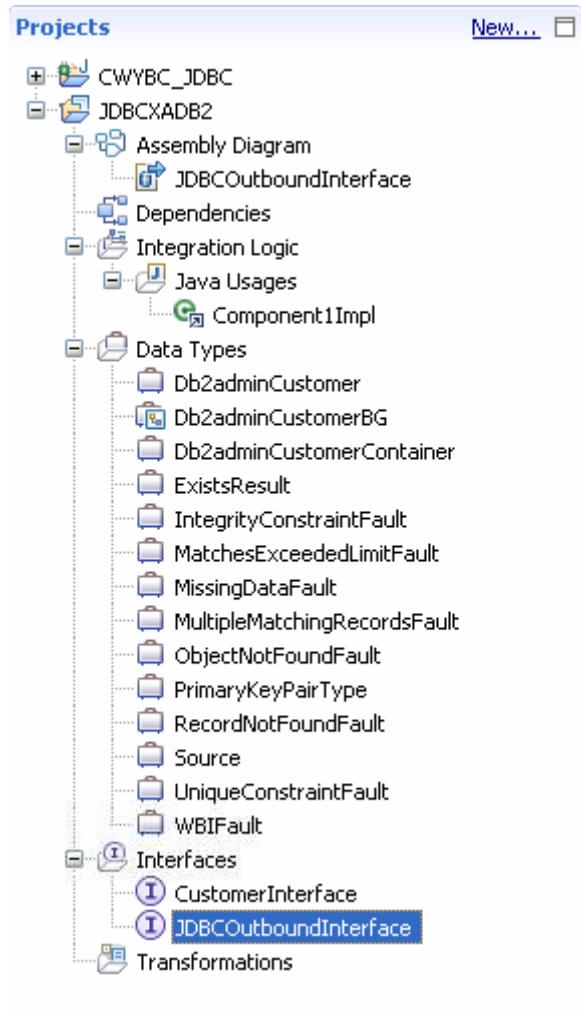
```
import com.ibm.websphere.sca.Service;
import com.ibm.websphere.sca.Ticket;
import commonj.sdo.DataObject;
import com.ibm.websphere.sca.ServiceManager;
import com.ibm.j2ca.base.SDOFactory;
import com.ibm.j2ca.base.exceptions.BusinessObjectDefinitionNotFoundException;

public class Component1Impl {
    /**
     * Default constructor.
     */
    public Component1Impl() {
        super();
    }
}
```

17. Add the following implementation for **invoke()** method.

```
public String invoke(DataObject input1) throws  
BusinessObjectDefinitionNotFoundException {  
  
    String objName =  
input1.getString("objectName");  
    String namespace =  
input1.getString("namespace");  
    DataObject customerBO =  
SDOFactory.createDataObject(namespace, objName);  
    DataObject customerBG =  
customerBO.getContainer();  
  
    customerBO.setString("pkey",  
input1.getString("pkey"));  
    customerBO.setString("fname",  
input1.getString("fname"));  
    customerBO.setString("lname",  
input1.getString("lname"));  
    customerBO.setString("ccode",  
input1.getString("ccode"));  
  
    String op = input1.getString("operation");  
  
    String operation =  
op.toLowerCase() + customerBG.getType().getName();  
  
    locateService_JDBCOutboundInterfacePartner().in  
voke(operation, customerBG);  
  
    return "Success";  
}
```

18. Select **File > Save All** to save all the changes.

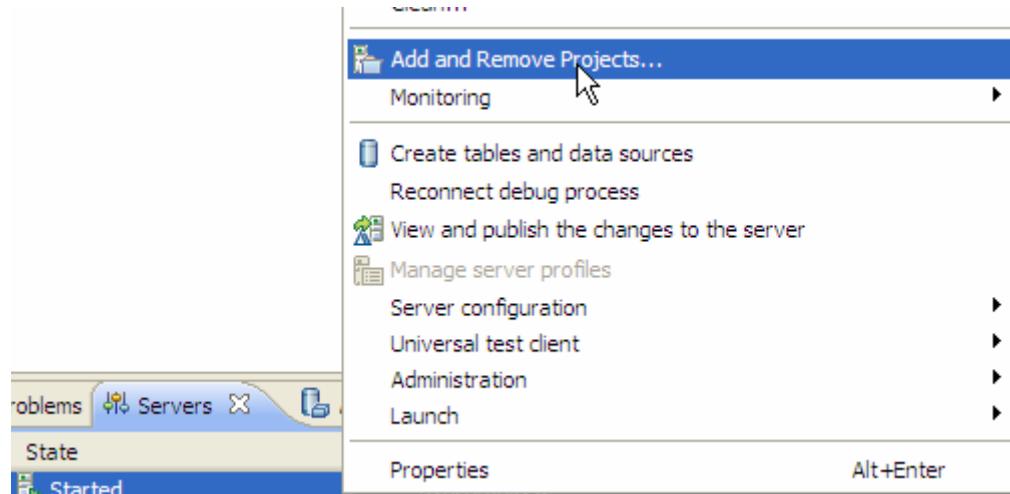


Deploy the module to the test environment

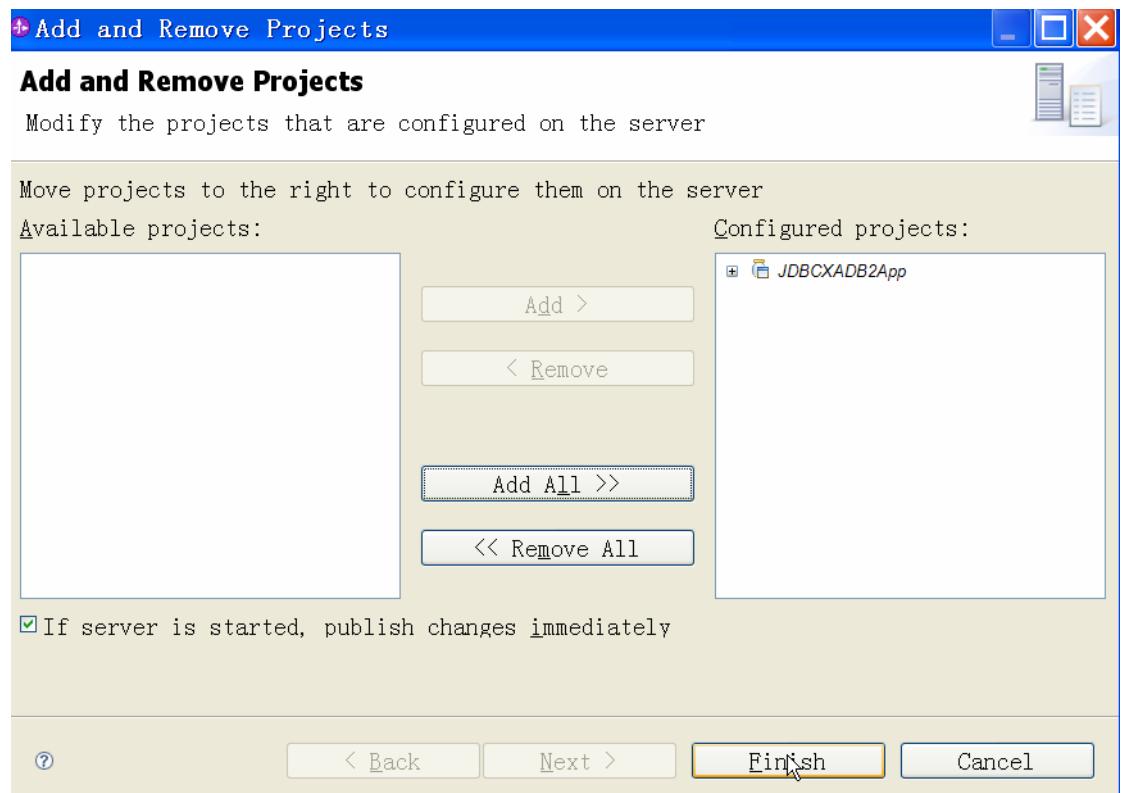
The result of running the external service wizard is an SCA module that contains an Enterprise Information System import. Install this SCA module in WebSphere Integration Developer integration test client. To do this, you must add the SCA module you created earlier to the server using the **Servers** view in WebSphere Integration Developer.

Steps for adding the SCA module to the server:

1. In WebSphere Integration Developer, switch to the **Servers** view by selecting **Windows > Show View > Servers**.
2. In the Servers tab in the lower-right pane of the WebSphere Integration Developer screen, right-click the server, and select **Start**.
3. After the server is started, right-click the server, and select **Add and Remove projects**.



- In the Add and Remove Projects window, select the module created earlier and click **Add**. The project is added to the **Configured Projects** list from the **Available Projects** list.

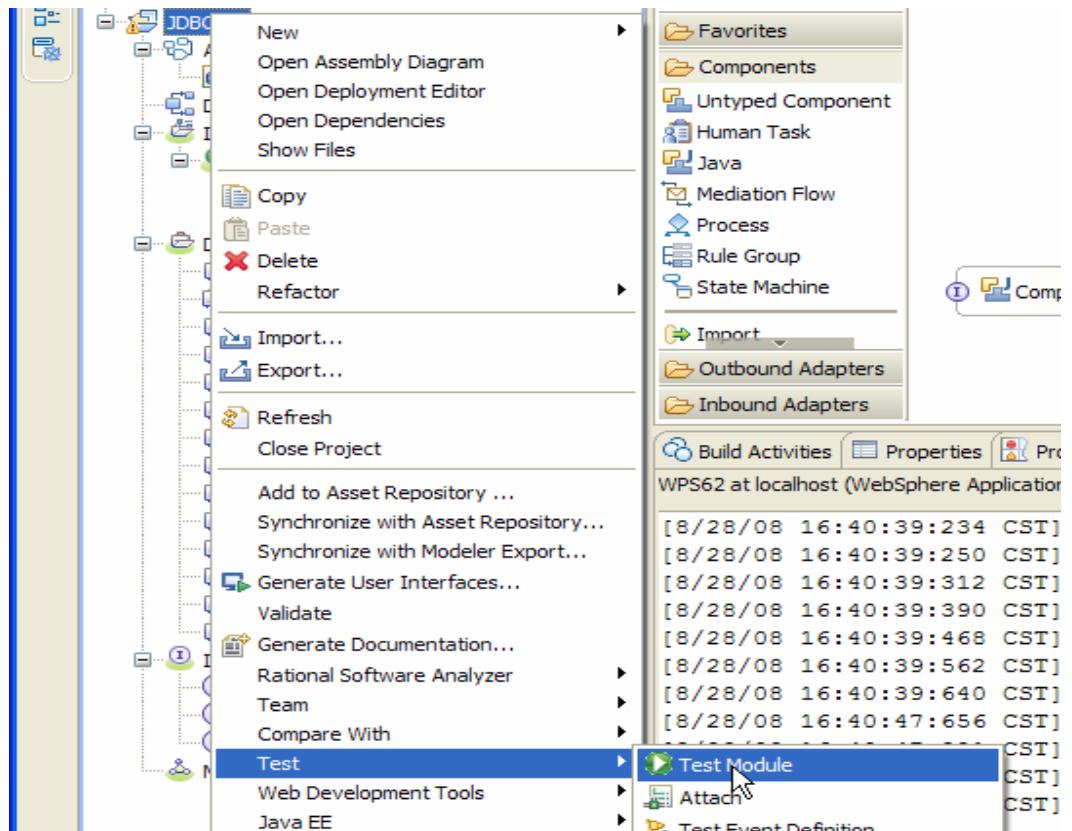


- Click **Finish**. This deploys the project on the server. For troubleshooting issues while adding the project, see the Troubleshooting section. The Console tab in the lower-right pane displays a log while the module is being added to the server.

Test the assembled adapter application

Test the assembled adapter application using the WebSphere Integration Developer integration test client:

- From the Business Integration view, right click on **JDBCXADB2** and select **Test > Test Module**.



- From the **Component** list, select **Component1**. Specify the parameters as shown in the figure below.

WebSphere software

Integration Test Client: JDBCXADB2_Test

Events

This area displays the events in a test trace. Select an event to display its properties in the General Properties and Detailed Properties sections. [More...](#)



Invoke

General Properties

Detailed Properties

Specify the component, interface, operation, and input parameter values for the Invoke event, then click the Continue icon in the Events area to run the test. [More...](#)

Configuration:	Default Module Test
Module:	JDBCXADB2
Component:	Component1
Interface:	CustomerInterface
Operation:	invoke

Initial request parameters:

Value editor XML editor

Name	Type	Value
input1	Source	✓
pkey	string	✓ 300
fname	string	✓ abc
Iname	string	✓ xyz
code	string	✓ IBM
operation	string	✓ Create
objectName	string	✓ Db2adminCustomerBG
namespace	string	✓ http://www.ibm.com/xmlns/pro...

To edit values, start typing or press F2.

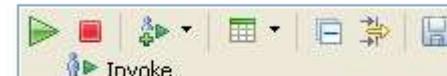
Note: Set the **operation** value to **Create**; set the **objectName** value to **Db2adminCustomerBG**; set the **namespace** value to **http://www.ibm.com/xmlns/prod/websphere/j2ca/jdbc/db2admindcustomerbg**.

3. Click to continue.

Integration Test Client: JDBCXADB2_Test

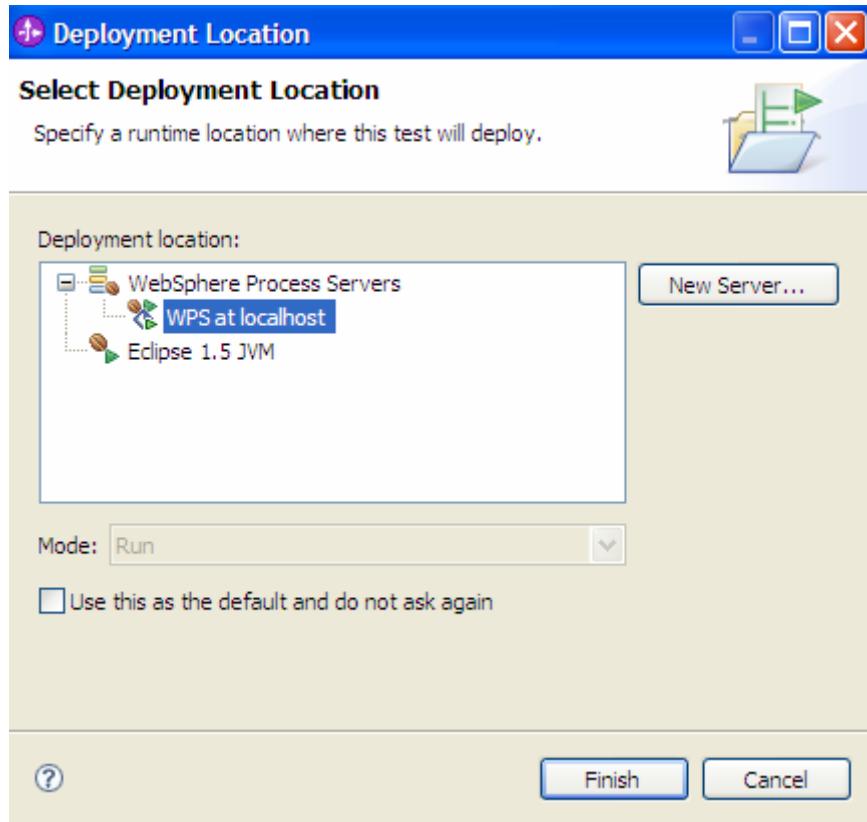
Events

This area displays the events in a test trace. Select an event to display its properties in the General Properties and Detailed Properties sections. [More...](#)



Invoke

4. In the Select Deployment Location window, select your server, and click **Finish**.



- Once the service is executed successfully, the customer record will be created in the target database. To verify the result, connect to the database and run the following SQL query:

```
SELECT * FROM CUSTOMER WHERE pkey = '300';
```

Clear the sample content

Return the data to its original state by deleting the Customer record you created in the CUSTOMER table by connecting to the database and running the SQL query:

```
DELETE FROM CUSTOMER WHERE pkey = '300';
```

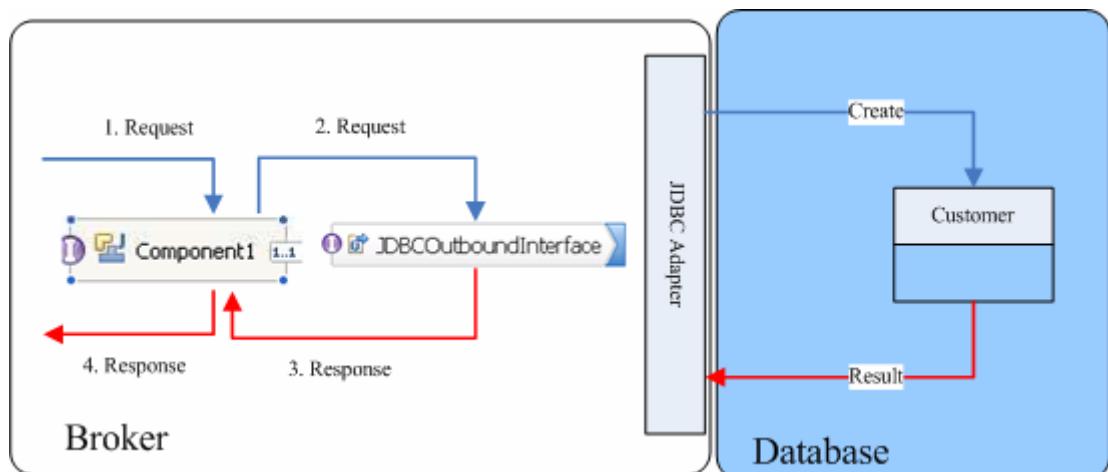
Chapter 9. Tutorial 8: Sending data to the Oracle database with XA transaction (outbound processing)

This scenario demonstrates how WebSphere Adapter for JDBC 7.0.0.0 participates in a global transaction using a XA data source for Oracle database.

About this task

In this scenario, we will create a Java component and a JDBC adapter import component. The Java component invokes JDBC adapter to make changes to the database. Both, the java component and JDBC adapter will participate in the same global transaction.

The following figure illustrates the scenario:



Prepare to run through the tutorial

Extract the sample files

Replicas of the artifacts that you create when using the external service wizard are provided as sample files for your reference. Use these files to verify if the files you create using the external service wizard are correct.

Download the sample zip file and extract it into a directory of your choice (you may want to create a new directory).

Configuration prerequisites

Before configuring the adapter, you must complete the following tasks:

- Create a table
- Create an authentication alias
- Create a data source

Create a table

You must create the following table in the Oracle database before starting the scenario.

Script for creating the CUSTOMER table:

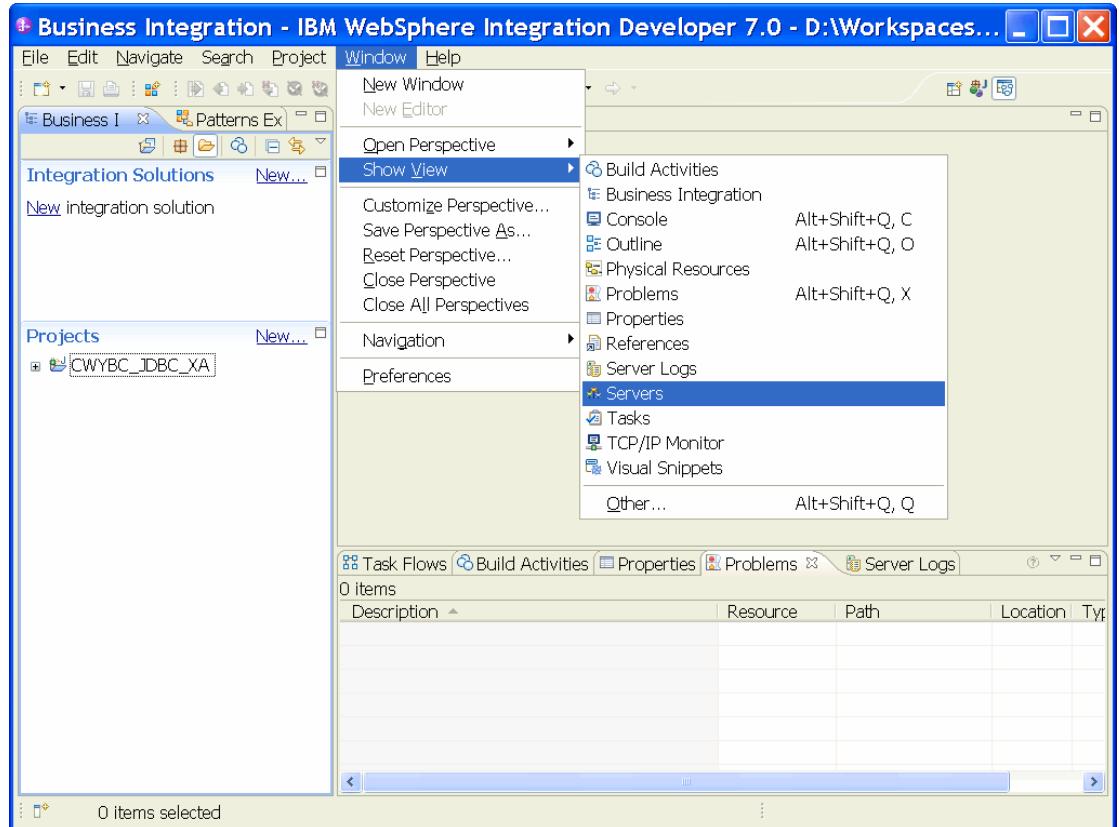
```
CREATE TABLE CUSTOMER (
    "PKEY" INTEGER NOT NULL PRIMARY KEY,
    "FNAME" VARCHAR(20) ,
    "LNAME" VARCHAR(20) ,
    "CCODE" VARCHAR(10) ) ;
```

Create an authentication alias

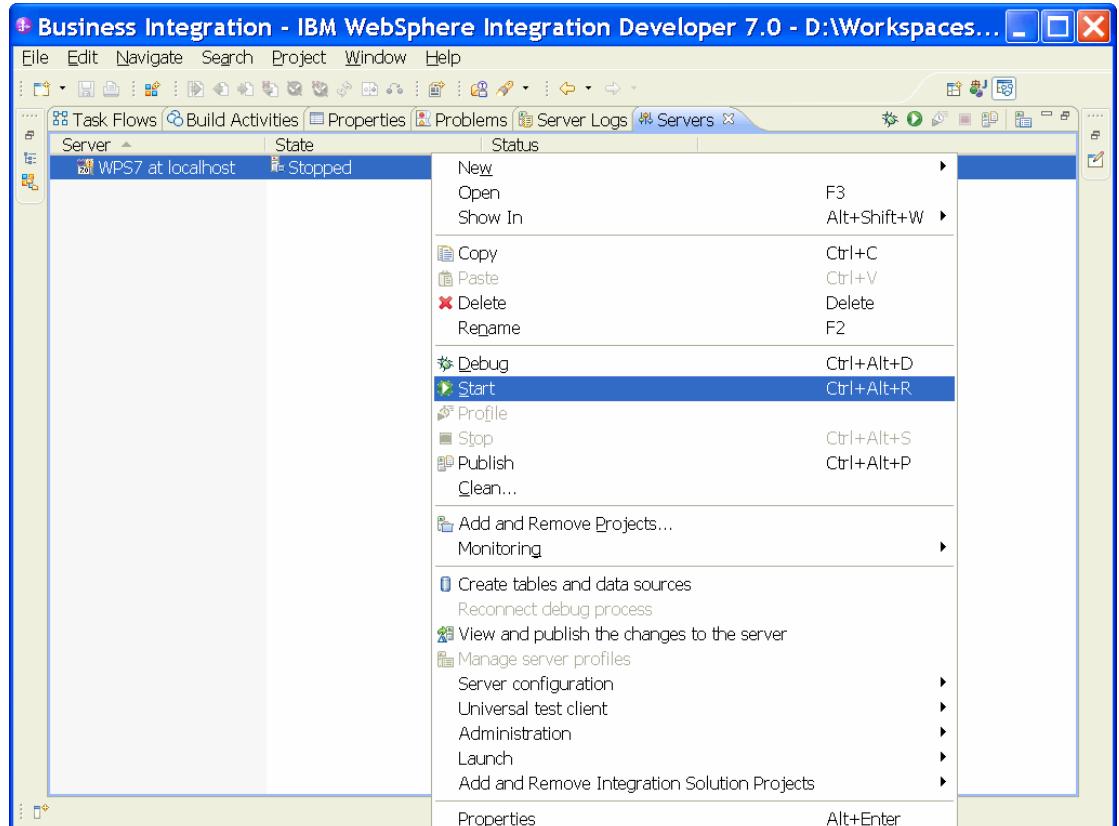
The authentication alias needs to be set because the data source created in the next section uses the username and password set in the authentication alias to connect to the database.

Follow these steps to set the authentication alias in the WebSphere Process Server administrative console.

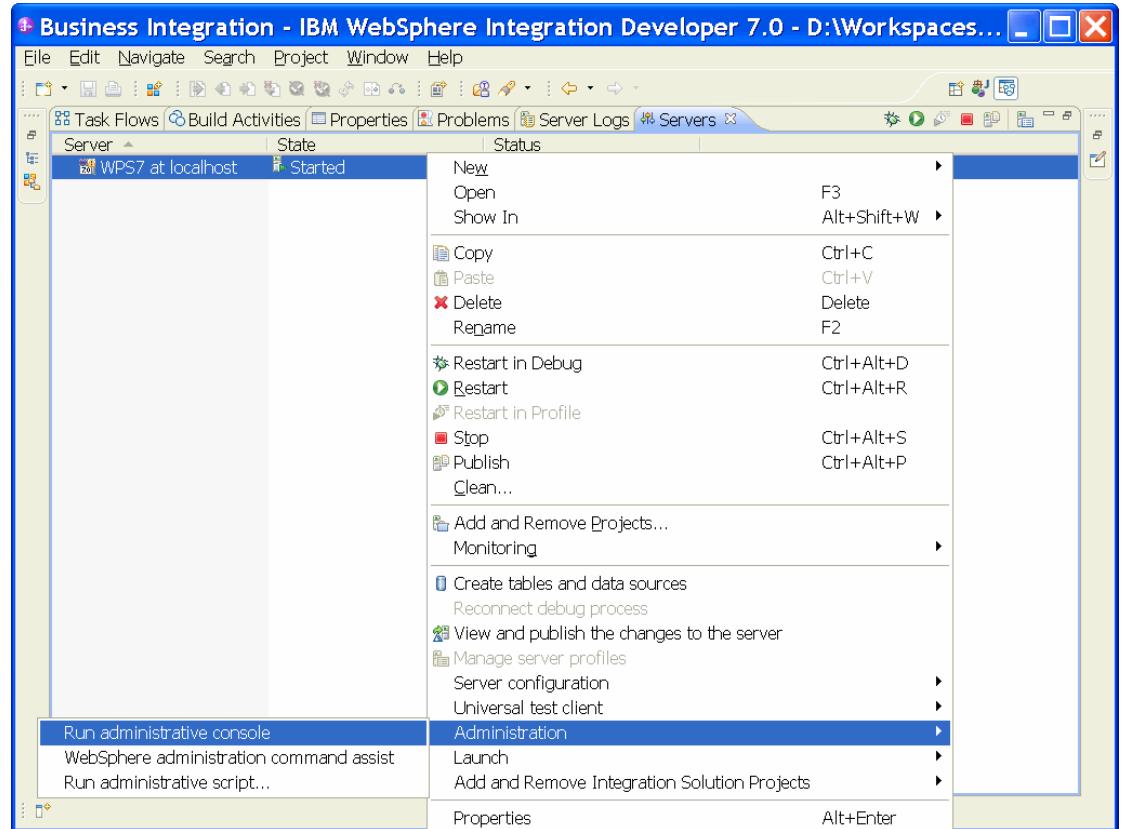
1. In WebSphere Integration Developer, switch to the **Servers** view by selecting **Windows > Show View > Servers**.



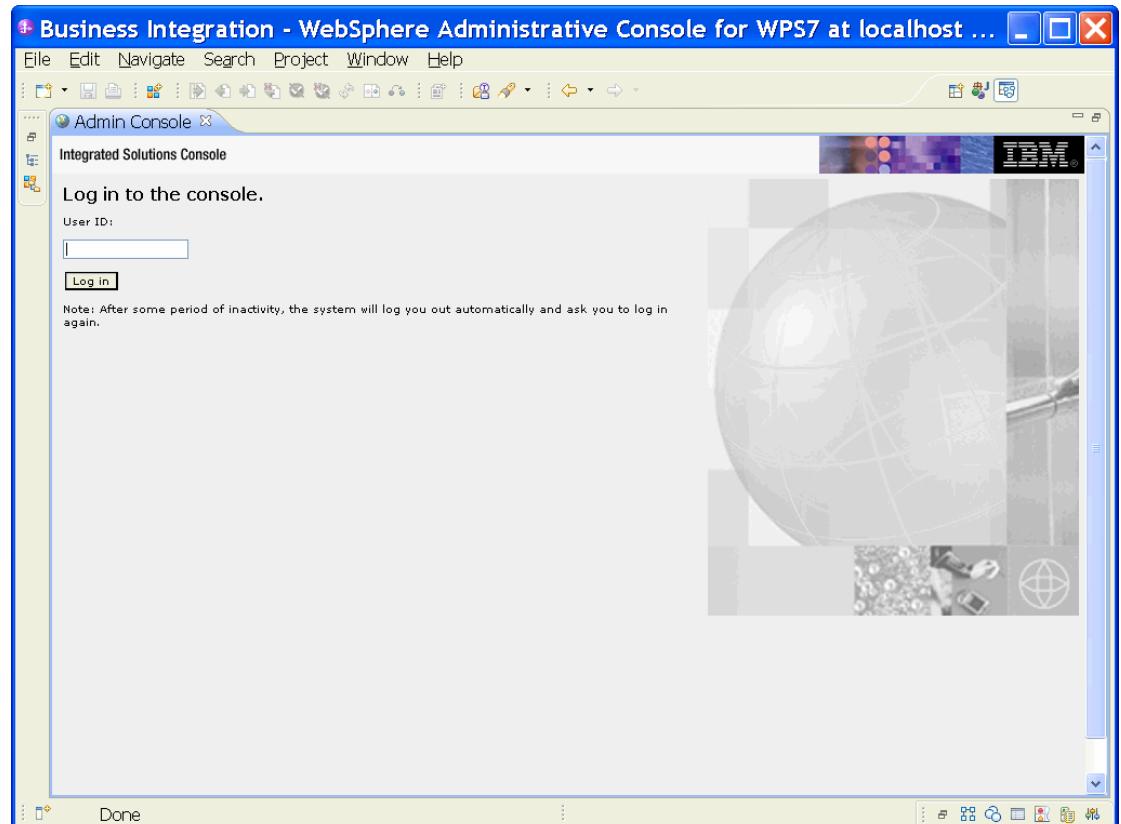
2. In the **Servers** view, right-click the server that you want to start and select **Start**.



3. After the server is started, right-click the server, and select **Administration > Run administrative console**.



4. Log on to the administrative console.



5. Click **Security → Global security**

WebSphere software

View: All tasks

- Welcome
- [+] Guided Activities
- [+] Servers
- [+] Applications
- [+] Services
- [+] Resources
- [-] Security
 - Business Integration Security
 - Global security
 - Security domains
 - Administrative Authorization Groups
 - SSL certificate and key management
 - Security auditing
 - Bus security
- [+] Environment
- [+] Integration Applications
- [+] System administration
- [+] Users and Groups
- [+] Monitoring and Tuning
- [+] Troubleshooting
- [+] Service integration
- [+] UDDI

6. Under **Java Authentication and Authorization Service**, click **J2C authentication data**.

Cell=localhostNode01Cell, Profile=AppSrv01 Close page

Global security

Global security
Use this panel to configure administration and the default application security policy. This security configuration applies to functions and is used as a default security policy for user applications. Security domains can be defined to override and control applications.

Security Configuration Wizard	Security Configuration Report	
<div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> Administrative security <p><input type="checkbox"/> Enable administrative security Administrative user roles <input type="checkbox"/> Administrative group roles <input type="checkbox"/> Administrative authentication</p> </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> Application security <p><input checked="" type="checkbox"/> Enable application security</p> </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> Java 2 security <p><input type="checkbox"/> Use Java 2 security to restrict application access to local resources <input checked="" type="checkbox"/> Warn if applications are granted custom permissions <input type="checkbox"/> Restrict access to resource authentication data</p> </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> User account repository Current realm definition: Federated repositories Available realm definitions: Federated repositories Configure... Set as current </div>		<div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> Authentication Authentication mechanisms and expiration <input checked="" type="radio"/> LTPA <input type="radio"/> Kerberos and LTPA Kerberos configuration <input type="radio"/> SWAM (deprecated): No authentication Authentication cache settings </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> <input type="checkbox"/> Web and SIP security <input type="checkbox"/> RMI/IOP security <input type="checkbox"/> Java Authentication and Authorization <input type="checkbox"/> Application logins <input type="checkbox"/> System logins <input type="checkbox"/> J2C authentication data </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> <p>Specifies a list of user identities and passwords for Java(TM) 2 connector security to use.</p> <p>Security domains External authorization providers Custom properties</p> </div>

WebSphere software

A list of existing aliases is displayed.

[Global security](#) > JAAS - J2C authentication data

Specifies a list of user identities and passwords for Java(TM) 2 connector security to use.

Prefix new alias names with the node name of the cell (for compatibility with earlier releases)

[Apply](#)

[+ Preferences](#)

New	Delete			
Select	Alias	User ID	Description	
You can administer the following resources:				
<input type="checkbox"/>	BSpace JDBC Alias	wbiuser	Business Space Authorization Alias	
<input type="checkbox"/>	CommonEventInfrastructureJMSAuthAlias	wbiuser	Authentication alias for the Common Event Infrastructure JMS Topics and Queues	
<input type="checkbox"/>	SCA Auth Alias	wbiuser	This is the alias used by SCA to login to a secured SIBus	
<input type="checkbox"/>	localhostNode01Cell/nlNode01/server1/EventAuthDataAliasDerby		Derby authentication alias for the Event Server	
Total 4				

- Click **New** to create a new authentication entry. Type the alias name and a username and password that can connect to the database, as shown in the figure. Click **OK**.

WebSphere software

Cell=localhostNode01Cell, Profile=AppSrv01

Global security

[Global security](#) > [JAAS - J2C authentication data](#) > New

Specifies a list of user identities and passwords for Java(TM) 2 connector security to use.

General Properties

* Alias: Alias_Oracle
* User ID: sample
* Password:
Description:

[Apply] [OK] [Reset] [Cancel]

8. Click **Save** to save the changes.

Cell=localhostNode01Cell, Profile=AppSrv01

Global security

Messages

⚠ Changes have been made to your local configuration. You can:

- [Save](#) directly to the master configuration.
- [Review](#) changes before saving or discarding.

⚠ The server may need to be restarted for these changes to take effect.

[Global security](#) > [JAAS - J2C authentication data](#)

Specifies a list of user identities and passwords for Java(TM) 2 connector security to use.

Prefix new alias names with the node name of the cell (for compatibility with earlier releases)

[Apply]

We have created an authentication alias that will be used to configure the data source.

WebSphere software

Preferences			
		New	Delete
Select	Alias	User ID	Description
You can administer the following resources:			
<input type="checkbox"/>	BSpace JDBC Alias	wbiuser	Business Space Authorization Alias
<input type="checkbox"/>	CommonEventInfrastructureJMSAuthAlias	wbiuser	Authentication alias for the Common Event Infrastructure JMS Topics and Queues
<input type="checkbox"/>	SCA Auth Alias	wbiuser	This is the alias used by SCA to login to a secured SIBus
<input type="checkbox"/>	localhostNode01Cell/nlNode01/server1/EventAuthDataAliasDerby		Derby authentication alias for the Event Server
<input type="checkbox"/>	nlNode01/AliasOracle	luweiqin	
Total 5			

Create a data source

1. Click **Resources > JDBC > JDBC Providers.**

Integrated Solutions Console Welcome admin

View: All tasks

- Welcome
- ⊕ Guided Activities
- ⊕ Servers
- ⊕ Applications
- ⊖ Resources
 - Schedulers
 - Object pool managers
 - ⊕ JMS
 - People directory provider
 - Extended messaging provider
 - WebSphere Business Integration Adapters
 - ⊖ JDBC
 - JDBC Providers
 - Data sources
 - Data sources (WebSphere Application Server V4)
 - Business Integration Data Sources
 - ⊕ Resource Adapters
 - ⊕ Asynchronous beans
 - ⊕ Cache instances
 - ⊕ Mail
 - ⊕ Remote artifacts
 - ⊕ URL
 - ⊕ Resource Environment
- ⊕ Security

2. On the right, select **Node=nINode01** from the drop-down list.

Integrated Solutions Console Welcome

Help | Logout

JDBC providers

JDBC providers

Use this page to edit properties of a JDBC provider. The JDBC provider object encapsulates the specific JDBC driver implementation class for access to the specific vendor database of your environment. Learn more about this task in a [guided activity](#). A guided activity provides a list of task steps and more general information about the topic.

⊖ Scope: Cell=localNode01Cell, Node=nINode01

Scope specifies the level at which the resource definition is visible. For detailed information on what scope is and how it works, [see the scope settings help](#)

Node=nINode01

⊖ Preferences

New	Delete		
<input type="checkbox"/>	<input type="checkbox"/>		
Select	Name	Scope	Description
<input type="checkbox"/>	Derby JDBC Provider (XA)	Node=nINode01	JDBC Provider for WPS/WESB

Help

Field help
For field help information, select a field label or list marker when the help cursor appears.

Page help
[More information about this page](#)

Command Assistance
[View administrative scripting command for last action](#)

- Click **New** in the JDBC providers window.

The screenshot shows the 'JDBC providers' page in the WebSphere Integrated Solutions Console. The 'Scope' dropdown is set to 'Node=nINode01'. In the 'Preferences' section, there is a 'New' button highlighted with a cursor. Below it is a table listing a single provider: 'Derby JDBC Provider (XA)' with 'Scope' 'Node=nINode01' and 'Description' 'JDBC Provider for WPS/WESB'.

Select	Name	Scope	Description
<input type="checkbox"/>	Derby JDBC Provider (XA)	Node=nINode01	JDBC Provider for WPS/WESB

- Specify the for the **Database type**, **Provider type**, and **Implementation type** fields as shown in the figure below. Click **Next**.

The screenshot shows the 'Create a new JDBC Provider' wizard, Step 1: Create new JDBC provider. The 'Scope' field is set to 'cells:localhostNode01Cell:nodes:nINode01'. The configuration details are as follows:

- * Database type: Oracle
- * Provider type: Oracle JDBC Driver
- * Implementation type: XA data source
- * Name: Oracle JDBC Driver (XA)
- Description: Oracle JDBC Driver (XA)

At the bottom, there are 'Next' and 'Cancel' buttons.

- Enter the absolute path of the JDBC driver (ojdbc6.jar) directory. Click **Next**.

JDBC providers

Create a new JDBC Provider

Create a new JDBC Provider

- Step 1: Create new JDBC provider
- Step 2: Enter database class path information
- Step 3: Summary

Enter database class path information

Set the environment variables that represent the JDBC driver class files, which WebSphere(R) Application Server uses to define your JDBC provider. This wizard page displays the file names; you supply only the directory locations of the files. Use complete directory paths when you type the JDBC driver file locations. For example: /home/db2inst1/sqllib/java on Linux(TM). If a value is specified for you, you may click Next to accept the value.

Class path:

Directory location for "ojdbc14.jar" which is saved as WebSphere variable

[Previous](#) [Next](#) [Cancel](#)

6. Click **Finish**.

JDBC providers

Create a new JDBC Provider

Create a new JDBC Provider

- Step 1: Create new JDBC provider
- Step 2: Enter database class path information
- Step 3: Summary

Summary

Summary of actions:

Options	Values
Scope	cells:localhostNode01Cell:nodes:n1Node01
JDBC provider name	Oracle JDBC Driver (XA)
Description	Oracle JDBC Driver (XA)
Class path	<input type="text" value="\${ORACLE_JDBC_DRIVER_PATH}/ojdbc14.jar"/>
<input type="text" value="\${ORACLE_JDBC_DRIVER_PATH}"/>	<input type="text" value="D:\lib\dbdrv\"/>
Native path	
Implementation class name	oracle.jdbc.xa.client.OracleXADatasource

[Previous](#) [Finish](#) [Cancel](#)

7. Click the JDBC Provider that you just created.

JDBC providers

Use this page to edit properties of a JDBC provider. The JDBC provider object encapsulates the specific JDBC driver implementation class for access to the specific vendor database of your environment. Learn more about this task in a [guided activity](#). A guided activity provides a list of task steps and more general information about the topic.

Scope: Cell=localhostNode01Cell, Node=nINode01

Scope specifies the level at which the resource definition is visible. For detailed information on what scope is and how it works, [see the scope settings help](#)

Node=nINode01

Select	Name	Scope	Description
<input type="checkbox"/>	Derby JDBC Provider (XA)	Node=nINode01	JDBC Provider for WPS/WESB
<input type="checkbox"/>	Oracle JDBC Driver	Node=nINode01	Oracle JDBC Driver
<input type="checkbox"/>	Oracle JDBC Driver (XA)	Node=nINode01	Oracle JDBC Driver (XA)

Total 3

8. Click **Data sources**, under **Additional Properties**.

General Properties

- * **Scope**: cells:localhostNode01Cell:nodes:nINode01
- * **Name**: Oracle JDBC Driver (XA)
- Description**: Oracle JDBC Driver (XA)
- Class path**: \${ORACLE_JDBC_DRIVER_PATH}/ojdbc14.jar
- Native library path**: (empty)
- * **Implementation class name**: oracle.jdbc.xa.client.OracleXADataSource

Additional Properties

- Data sources** (highlighted)
- Data sources (/WebSphere Application Server)**: Use this page to edit the settings associated with your selected JDBC source object supplies your application with the connection to accessing the database.

9. Click **New**.

Integrated Solutions Console Welcome

JDBC providers

JDBC providers

Messages

Changes have been made to your local configuration. You can:

- Save directly to the master configuration.
- Review changes before saving or discarding.

The server may need to be restarted for these changes to take effect.

[JDBC providers](#) > [Oracle JDBC Driver \(XA\)](#) > Data sources

Use this page to edit the settings of a data source that is associated with your selected JDBC provider. The data source object supplies your application with connections for accessing the database. Learn more about this task in a [guided activity](#). A guided activity provides a list of task steps and more general information about the topic.

Preferences

New	Delete	Test connection	Manage state...
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Select	Name	JNDI name	Scope
None			
Total 0			

10. Enter **jdbc/OracleXADS** for **JNDI name**.

11. Under **Component-managed authentication alias and XA recovery authentication alias**, select the name of the authentication alias you previously created from the drop-down list. Click **Next**.

Integrated Solutions Console Welcome

Create a data source

→ Step 1: Enter basic data source information

Step 2: Enter database specific properties for the data source

Step 3: Summary

Enter basic data source information

Set the basic configuration values of a data source for association with your JDBC provider. A data source supplies the physical connections between the application server and the database.

Requirement: Use the Data sources (WebSphere(R) Application Server V4) console pages if your applications are based on the Enterprise JavaBeans (TM) (EJB) 1.0 specification or the Java(TM) Servlet 2.2 specification.

Scope
cells:localhostNode01Cell:nodes:nlNode01

JDBC provider name
Oracle JDBC Driver (XA)

* Data source name
Oracle JDBC Driver XA DataSource

* JNDI name
jdbc/OracleXADS

Component-managed authentication alias and XA recovery authentication alias

Select a component-managed authentication alias. The selected authentication alias will also be set as the XA recovery authentication alias if your JDBC Provider supports XA. If you choose to [create a new J2C authentication alias](#), the wizard will be canceled.

nlNode01/Alias_Oracle

Next Cancel

12. Enter the URL to connect to the database in the **URL** field. Click **Next**.

Integrated Solutions Console Welcome

JDBC providers

Create a data source

Step 1: Enter basic data source information

→ Step 2: Enter database specific properties for the data source

Step 3: Summary

Enter database specific properties for the data source

Set these database-specific properties, which are required by the database vendor JDBC driver to support the connections that are managed through this data source.

* URL
jdbc:oracle:thin:@9.186.116.88:1521:Ora9i

* Data store helper class name
Oracle9i and prior data store helper

Use this data source in container managed persistence (CMP)

Previous Next Cancel

13. Click **Finish**.

Integrated Solutions Console Welcome

JDBC providers

Create a data source

Step 1: Enter basic data source information

Step 2: Enter database specific properties for the data source

→ Step 3: Summary

Summary	
Summary of actions:	
Options	Values
Scope	cells:localhostNode01Cell:nodes:nlNode01
Data source name	Oracle JDBC Driver XA DataSource
JNDI name	jdbc/OracleXADS
Component-managed authentication alias	nlNode01/Alias_Oracle
Select an existing JDBC provider	Oracle JDBC Driver (XA)
Implementation class name	oracle.jdbc.xa.client.OracleXADataSource
URL	jdbc:oracle:thin:@9.186.116.88:1521:Ora9i
Data store helper class name	com.ibm.websphere.radapter.OracleDataStoreHelper
Use this data source in container managed persistence (CMP)	true

Previous Finish Cancel

14. In the **Messages** area, click on **Save** link. This will save changes made to the local configuration onto the master configuration.

JDBC providers

JDBC providers

Messages

- ⚠ Changes have been made to your local configuration. You can:
 - [Save](#) directly to the master configuration.
 - [Revert](#) changes before saving or discarding.
- ⚠ The server may need to be restarted for these changes to take effect.

[JDBC providers > Oracle JDBC Driver \(XA\) > Data sources](#)

Use this page to edit the settings of a data source that is associated with your selected JDBC provider. The data source object supplies your application with connections for accessing the database. Learn more about this task in a [guided activity](#). A guided activity provides a list of task steps and more general information about the topic.

Preferences

[New](#) [Delete](#) [Test connection](#) [Manage state...](#)



Select	Name	JNDI name	Scope	Provider	Description	Category
<input type="checkbox"/>	Oracle JDBC Driver XA DataSource	jdbc/OracleXADS	Node=nlNode01	Oracle JDBC Driver (XA)	New JDBC Datasource	

Total 1

15. Select the check box next to the Data source you just created. Click **Test Connection**.

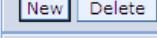
JDBC providers

[JDBC providers > Oracle JDBC Driver \(XA\) > Data sources](#)

Use this page to edit the settings of a data source that is associated with your selected JDBC provider. The data source object supplies your application with connections for accessing the database. Learn more about this task in a [guided activity](#). A guided activity provides a list of task steps and more general information about the topic.

Preferences

[New](#) [Delete](#) [Test connection](#) [Manage state...](#)



Select	Name	JNDI name	Scope	Provider	Description	Category
<input checked="" type="checkbox"/>	Oracle JDBC Driver XA DataSource	jdbc/OracleXADS	Node=nlNode01	Oracle JDBC Driver (XA)	New JDBC Datasource	

Total 1

16. The connection test should succeed as indicated by the message shown in the figure below. For troubleshooting issues while testing the connection, see the Troubleshooting section.

JDBC providers

Messages

The test connection operation for data source Oracle JDBC Driver XA Datasource on server server1 at node nlNode01 was successful.

JDBC providers > Oracle JDBC Driver (XA) > Data sources

Use this page to edit the settings of a data source that is associated with your selected JDBC provider. The data source object supplies your application with connections for accessing the database. Learn more about this task in a [guided activity](#). A guided activity provides a list of task steps and more general information about the topic.

Preferences

New	Delete	Test connection	Manage state...
Select	Name	JNDI name	Scope
<input type="checkbox"/>	Oracle JDBC Driver XA Datasource	jdbc/OracleXADS	Node=nlNode01
Provider: Oracle JDBC Driver (XA)			
Description: New JDBC Datasource			
Category:			
Total 1			

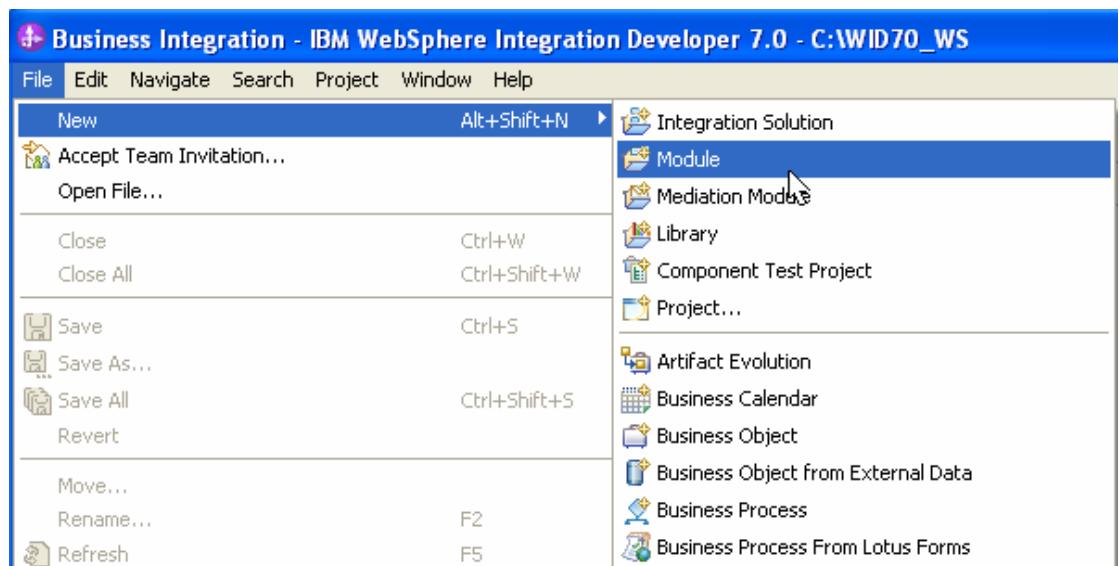
17. Close the **Administration Console** tab.

Configure the adapter for outbound processing

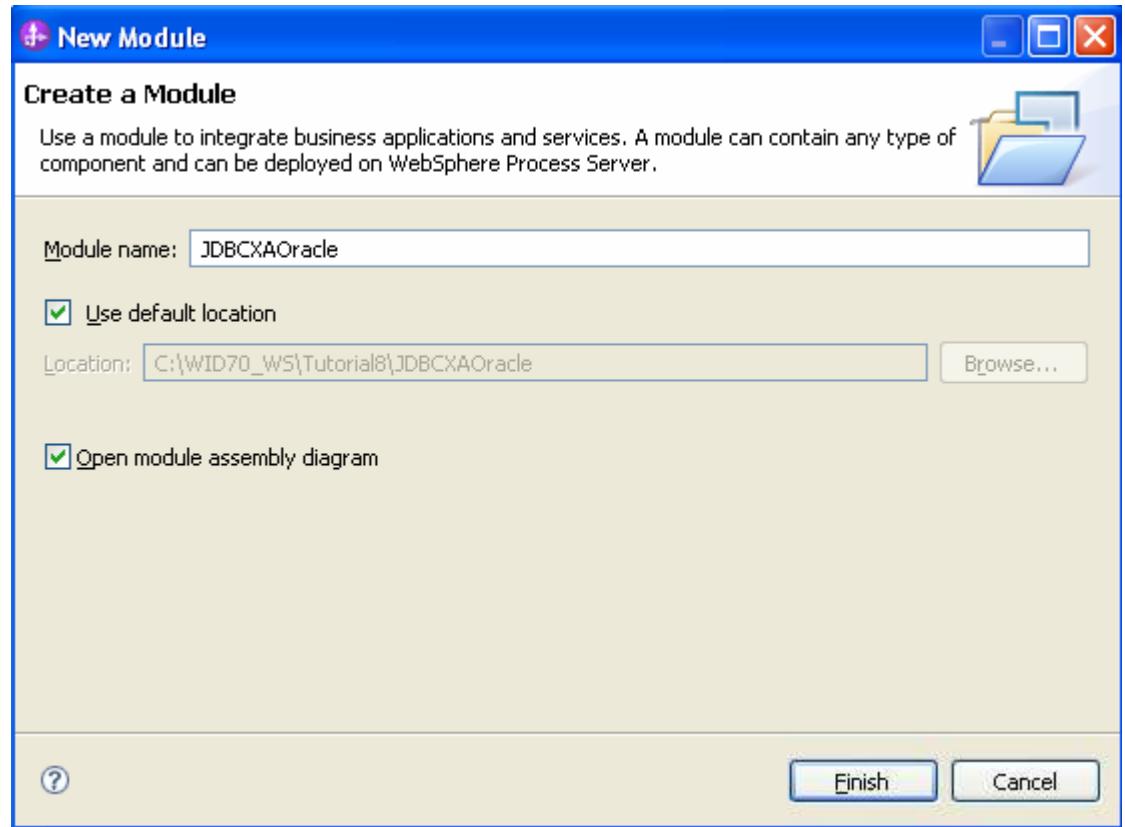
Run the external service wizard to specify business objects, services, and configuration details.

Set connection properties for the external service wizard

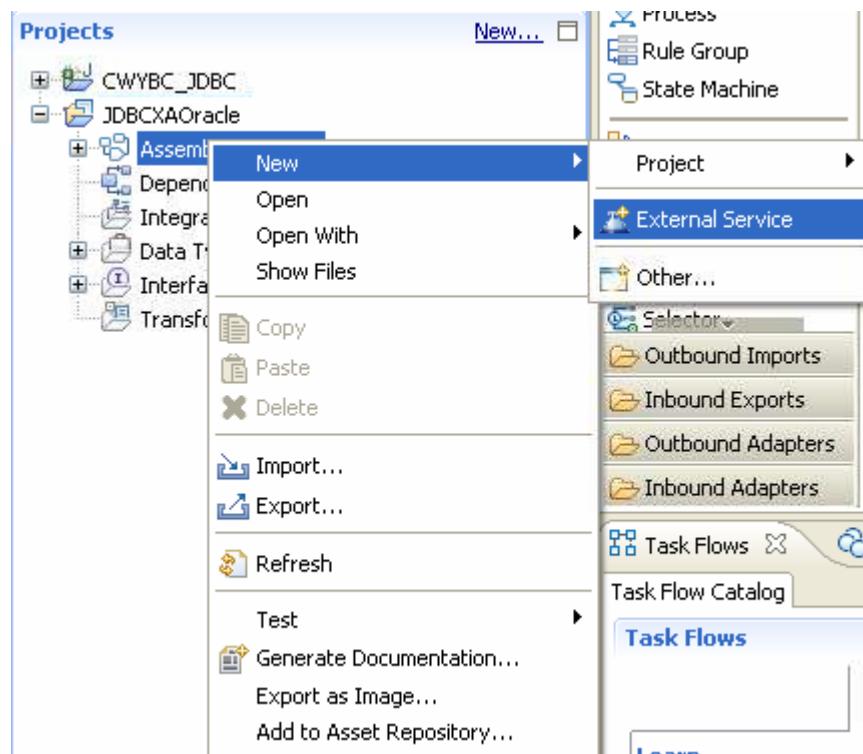
1. Switch to the Business Integration perspective in WebSphere Integration Developer.
2. Select **File->New->Module** to create a Module project.



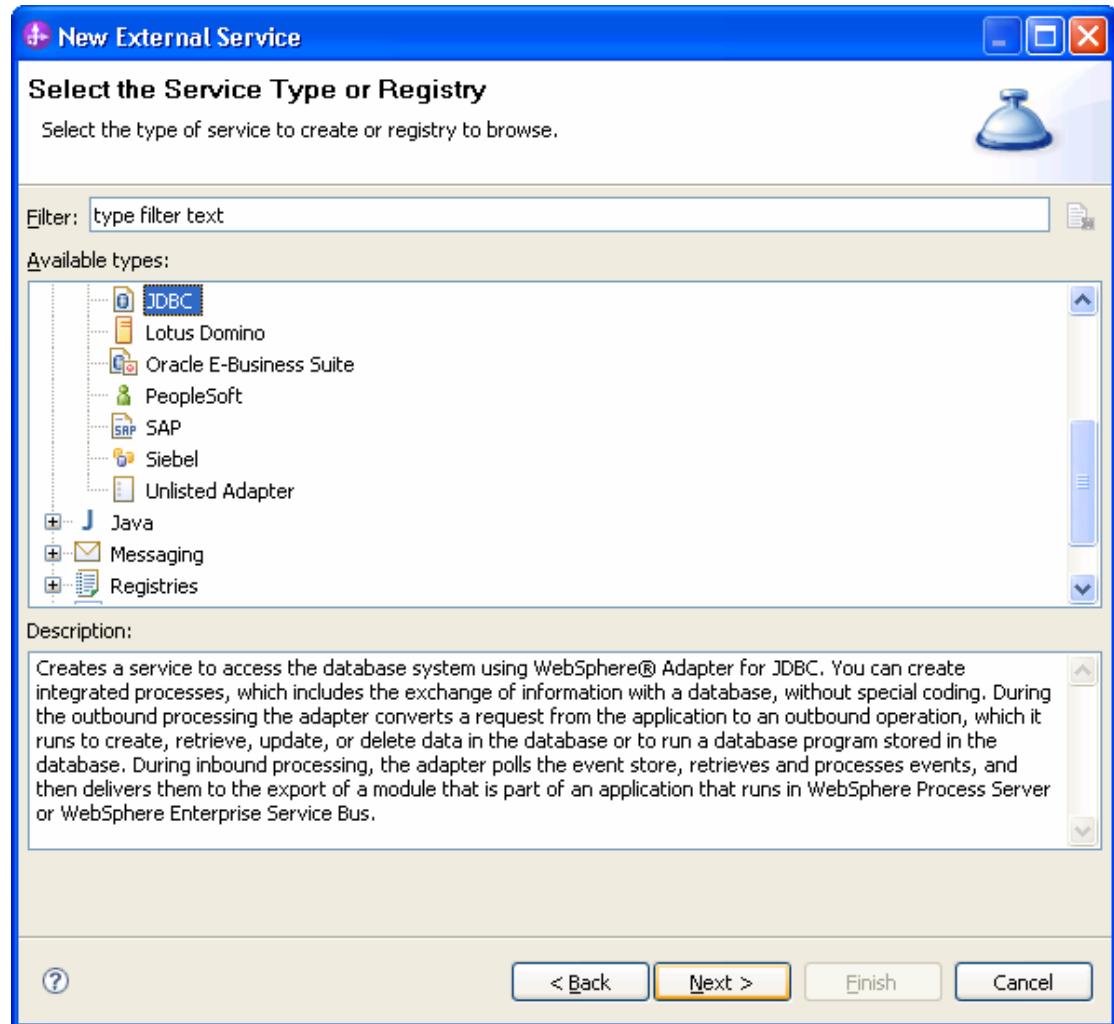
3. Specify the module name as **JDBCXAOracle**, click **Finish**.



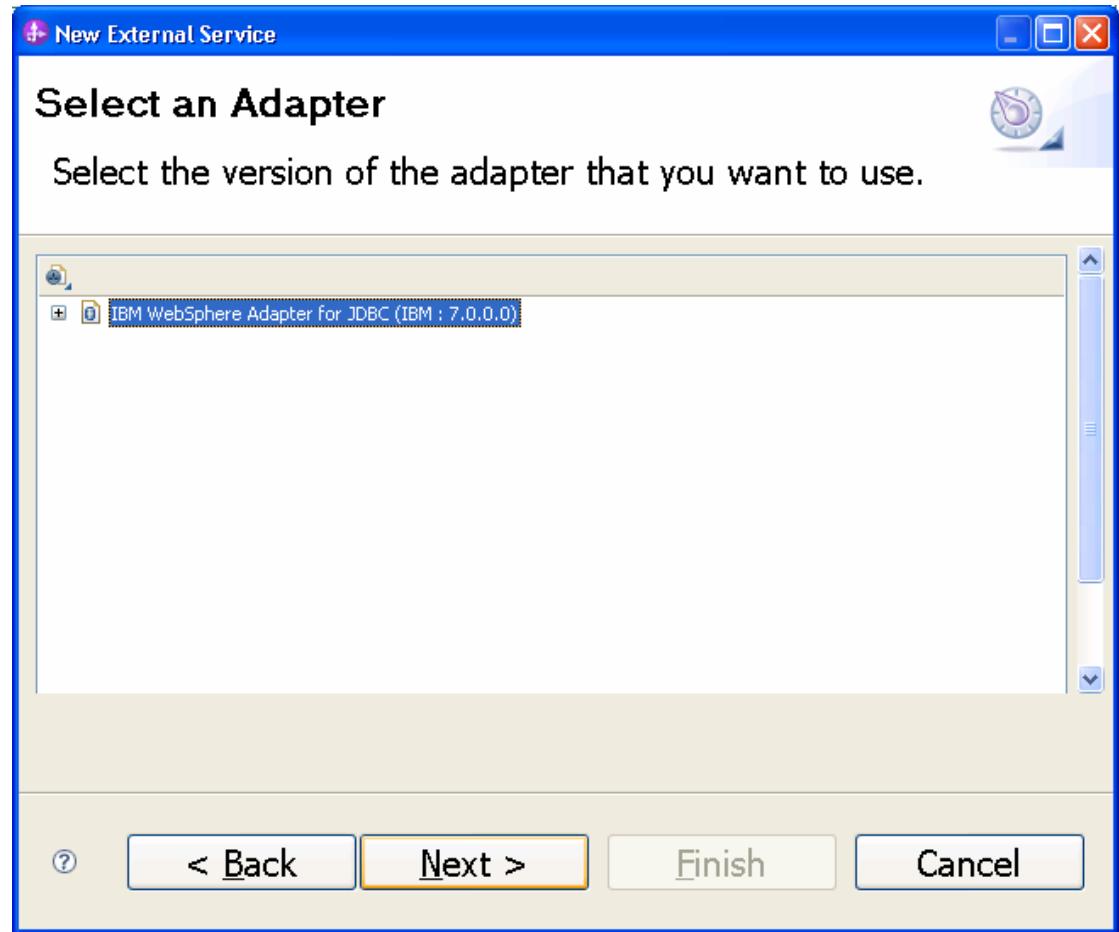
4. Right-click **Tutorial8->Assembly Diagram**, select **New->External Service**.



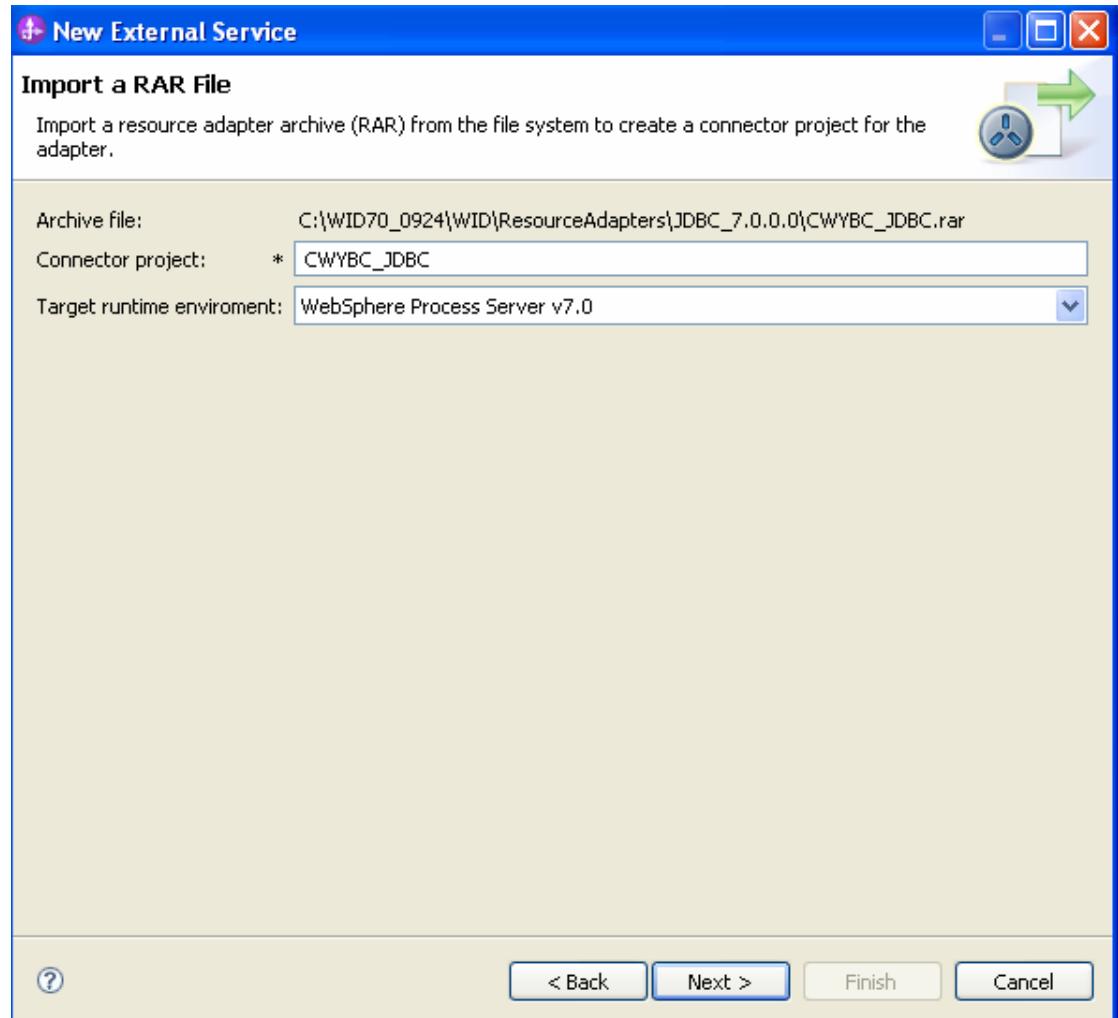
5. Select JDBC, and click **Next**.



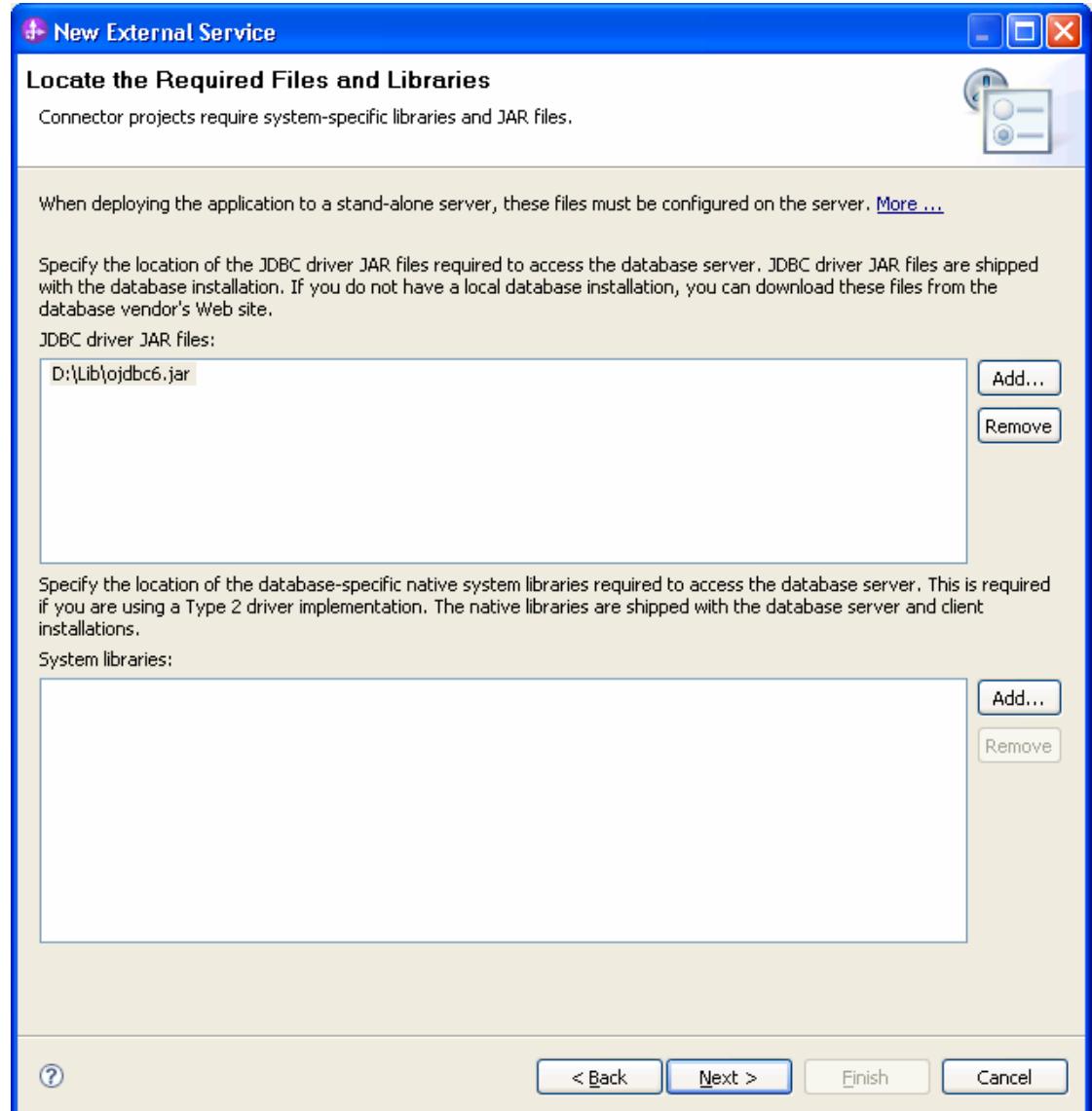
6. Select **IBM WebSphere Adapter for JDBC (IBM: 7.0.0.0)**. Click **Next**.



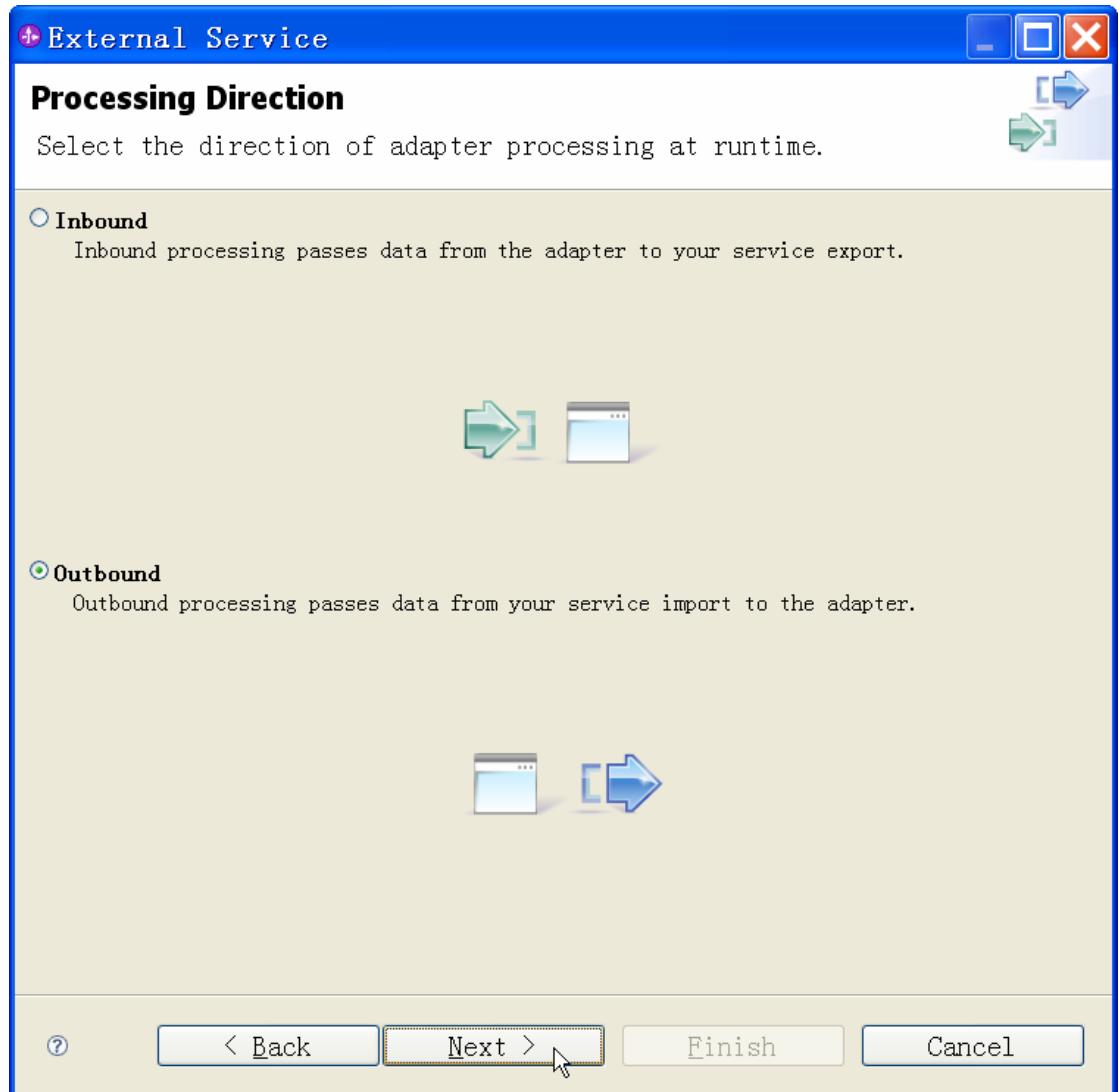
7. In the **Target Runtime environment** field, select the appropriate runtime and click **Next**.



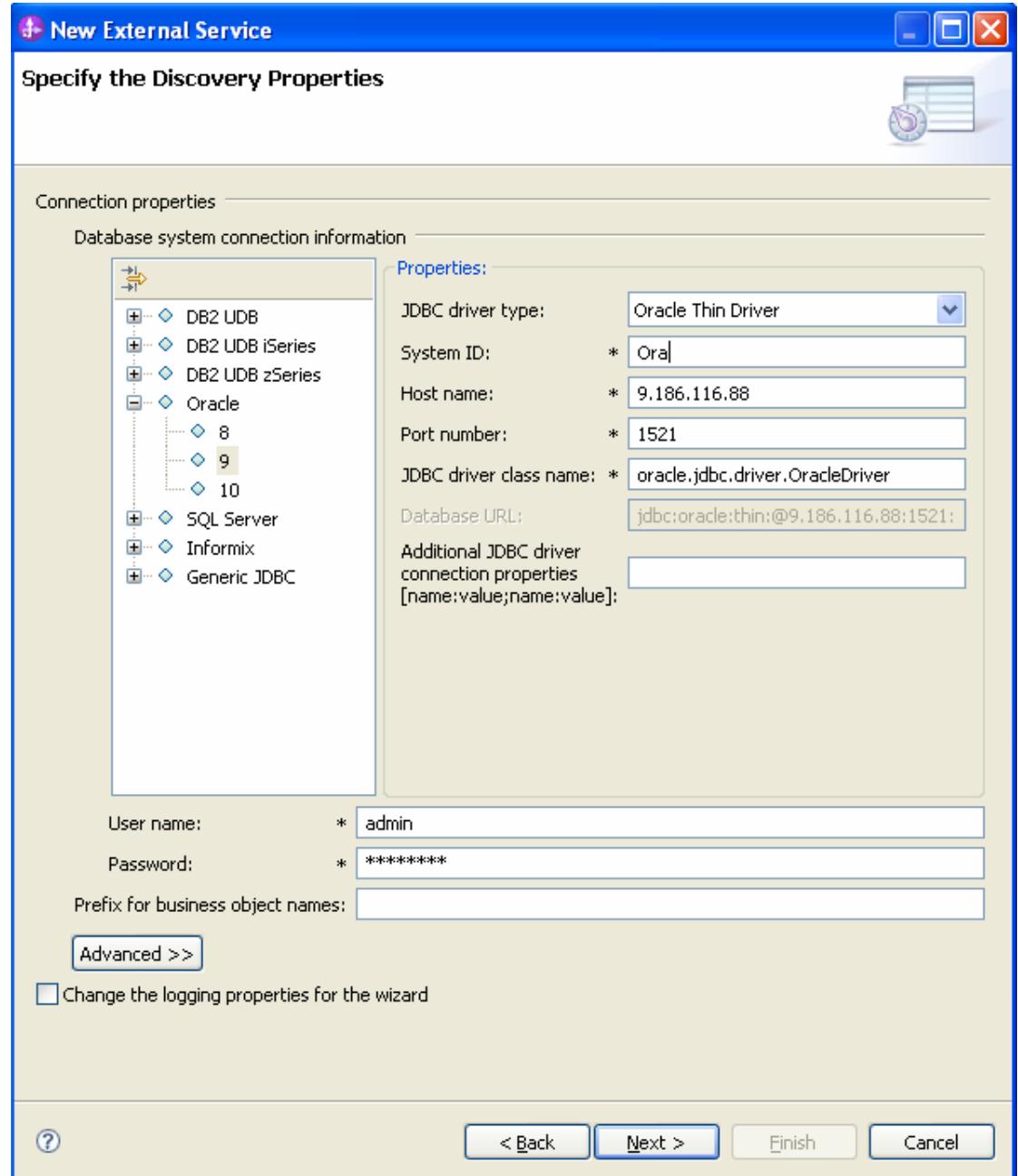
8. In the **JDBC driver JAR files** field, click **Add** to add the JDBC driver class to connect to the database. Browse to select the driver JAR file and click **Next**.



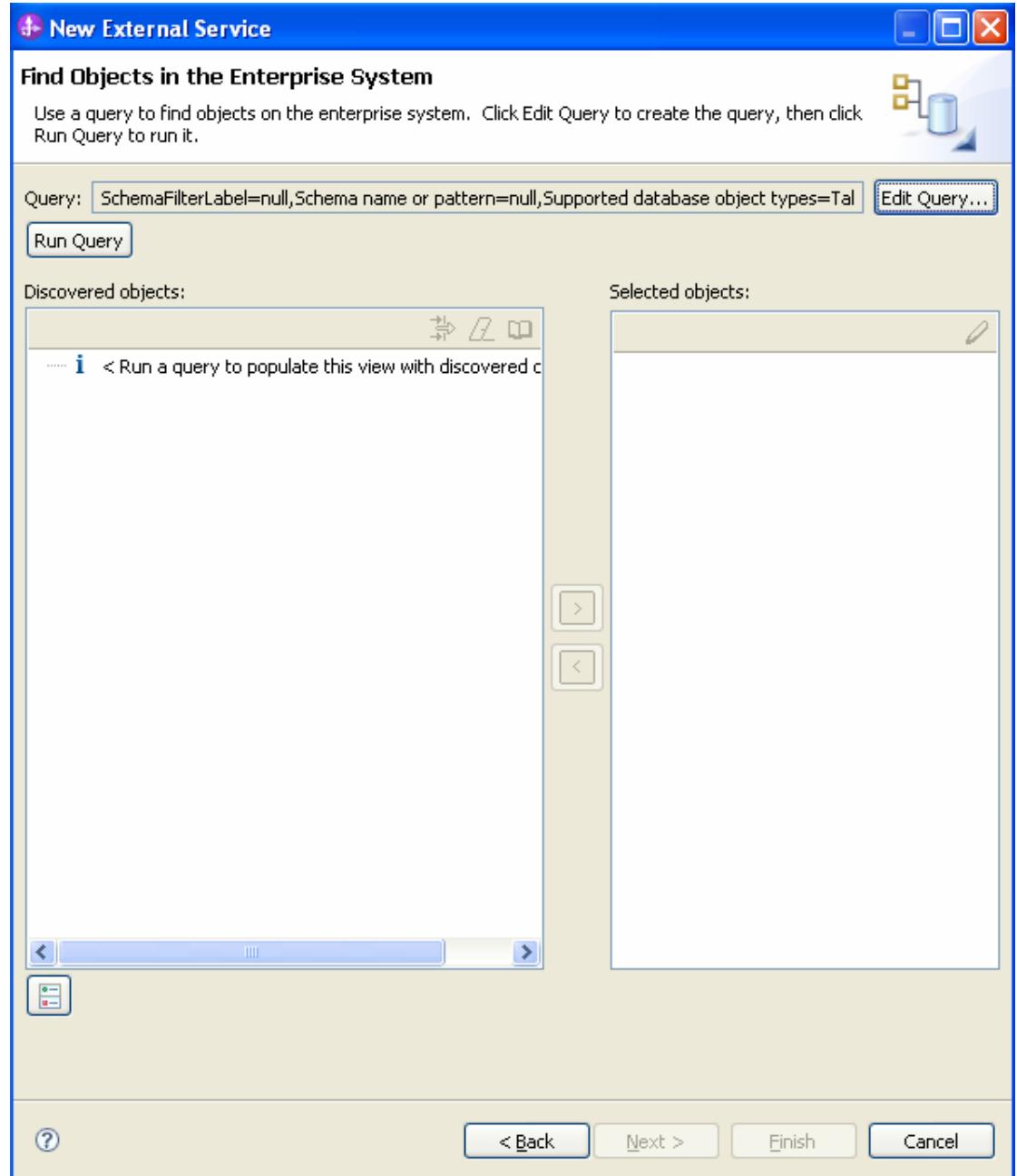
9. Click the **Outbound** radio button. Click **Next >**.



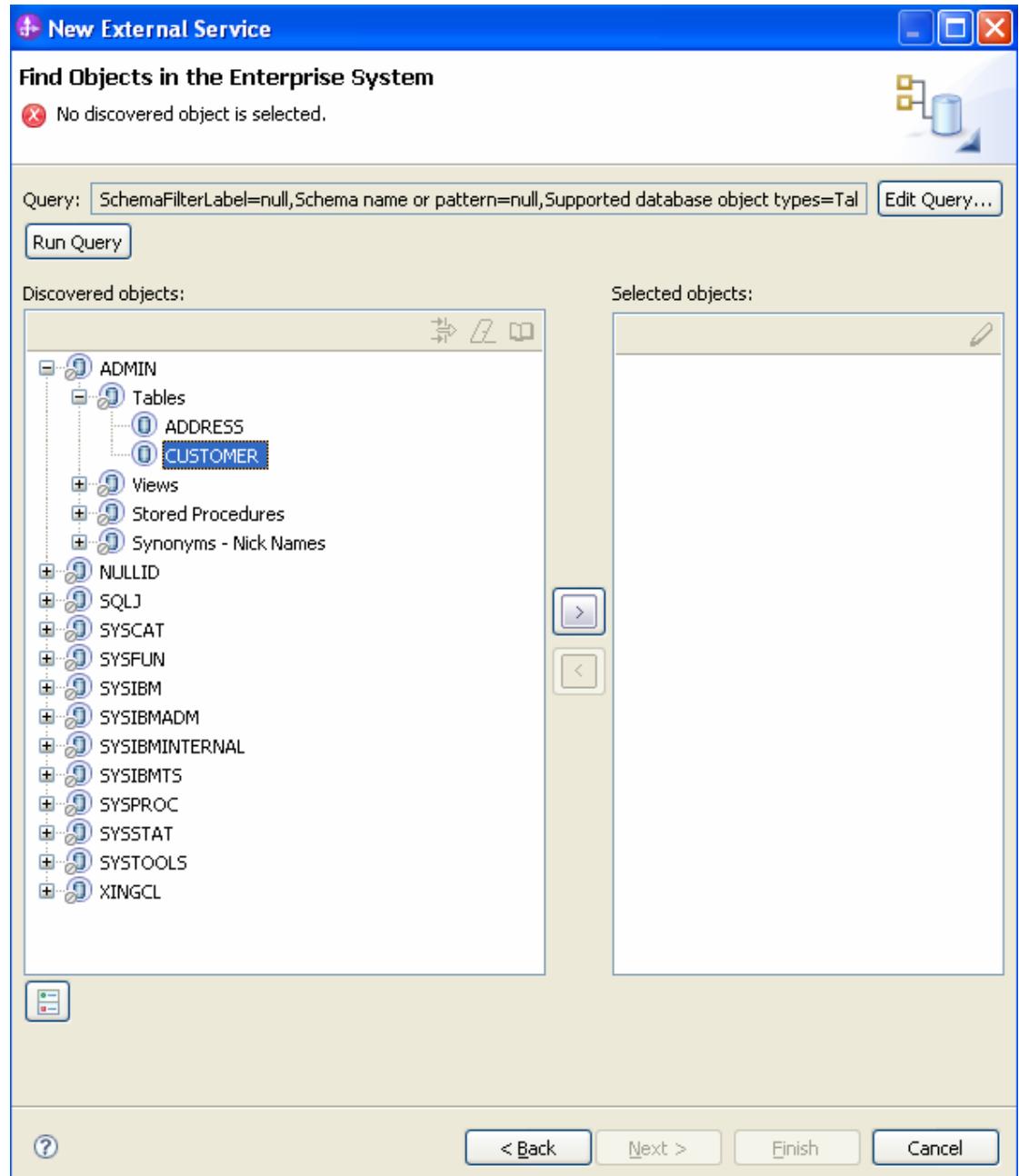
10. Expand the **Oracle** node in the **Database system connection information** area and then select **10**.
11. Enter values in the **System ID**, **Host name**, **Port number**, **User name** and **Password** fields, and click **Next**.



12. Click **Run Query** to list the tables, stored procedures, views, and synonyms for each schema in the database.



13. Expand the schema name in which you created the CUSTOMER table.
Select **Tables** > **CUSTOMER** and click the > **(Add)**. The CUSTOMER table is added to the **Selected objects** list.



14. Click **Next**.

 New External Service

Find Objects in the Enterprise System



Use a query to find objects on the enterprise system. Click Edit Query to create the query, then click Run Query to run it.

Query: SchemaFilterLabel=null,Schema name or pattern=null,Supported database object types=Tal [Edit Query...](#)

[Run Query](#)

Discovered objects:

A tree view of database objects under the ADMIN schema. The 'Tables' node is expanded, showing 'ADDRESS' and 'CUSTOMER'. Other nodes include 'Views', 'Stored Procedures', and 'Synonyms - Nick Names'. Other schemas listed are NULLID, SQLJ, SYSCAT, SYSFUN, SYSIBM, SYSIBADM, SYSIBMINTERNAL, SYSIBMTS, SYSPROC, SYSSTAT, SYSTOOLS, and XINGCL.

- ADMIN
 - Tables
 - ADDRESS
 - CUSTOMER
 - + Views
 - + Stored Procedures
 - + Synonyms - Nick Names
- + NULLID
- + SQLJ
- + SYSCAT
- + SYSFUN
- + SYSIBM
- + SYSIBADM
- + SYSIBMINTERNAL
- + SYSIBMTS
- + SYSPROC
- + SYSSTAT
- + SYSTOOLS
- + XINGCL

Selected objects:

A list view showing the 'CUSTOMER' object selected from the discovered objects.

- CUSTOMER



< Back

[Next >](#)

Finish

Cancel

15. In the Specify Composite Properties window, click **Next**.

 New External Service

Specify Composite Properties

Specify properties that apply to all selected objects.

Operations for selected business objects
Operations for these functions will be added to the service interface.*

Create
Update
Delete
Retrieve
RetrieveAll
ApplyChanges
Exists

Add...
Remove

Create and configure user-defined wrapper objects

Wrapper object names:

Add...
Remove

Return all records for RetrieveAll operation

Maximum records for RetrieveAll operation:

Business object namespace:

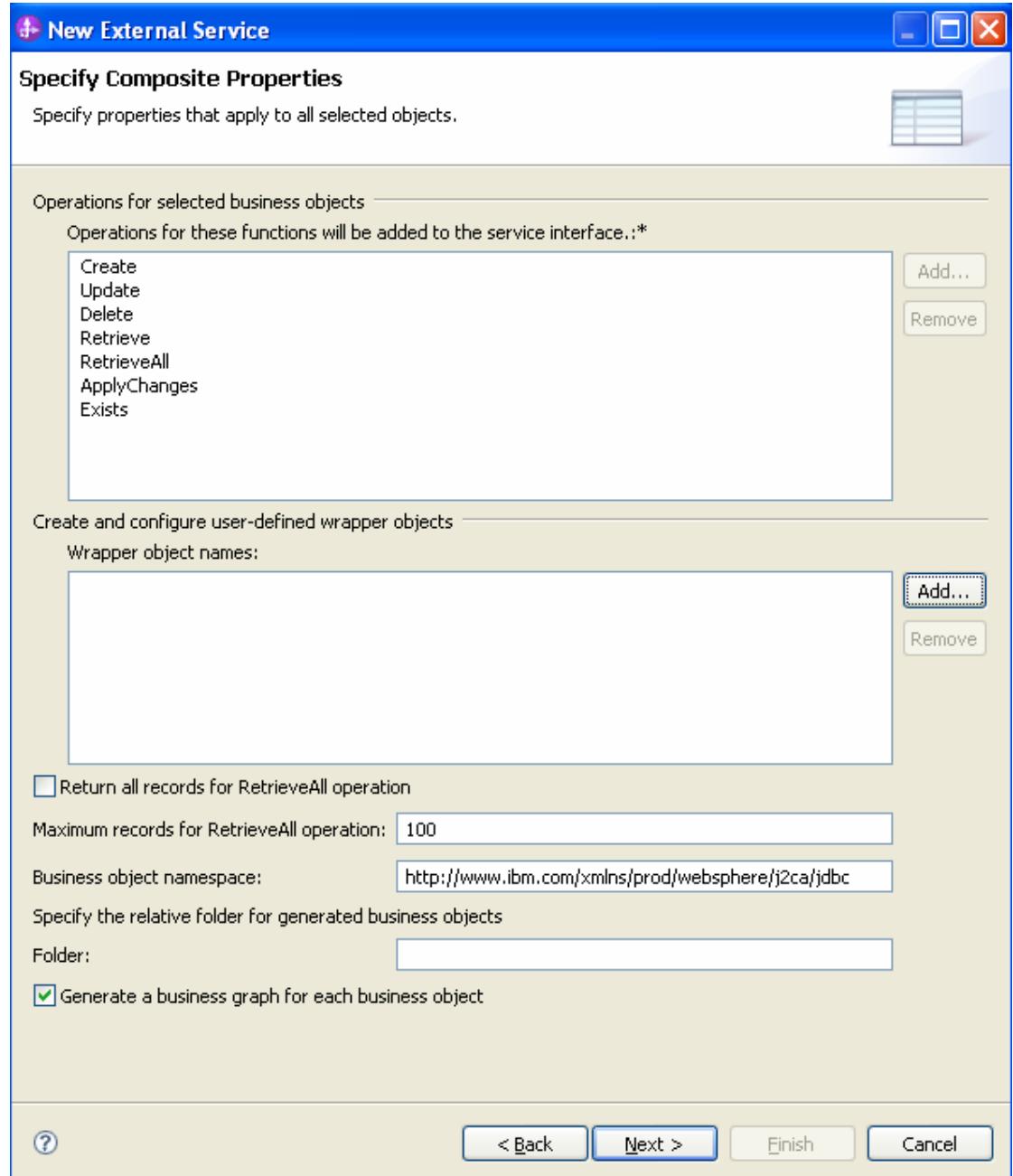
Specify the relative folder for generated business objects

Folder:

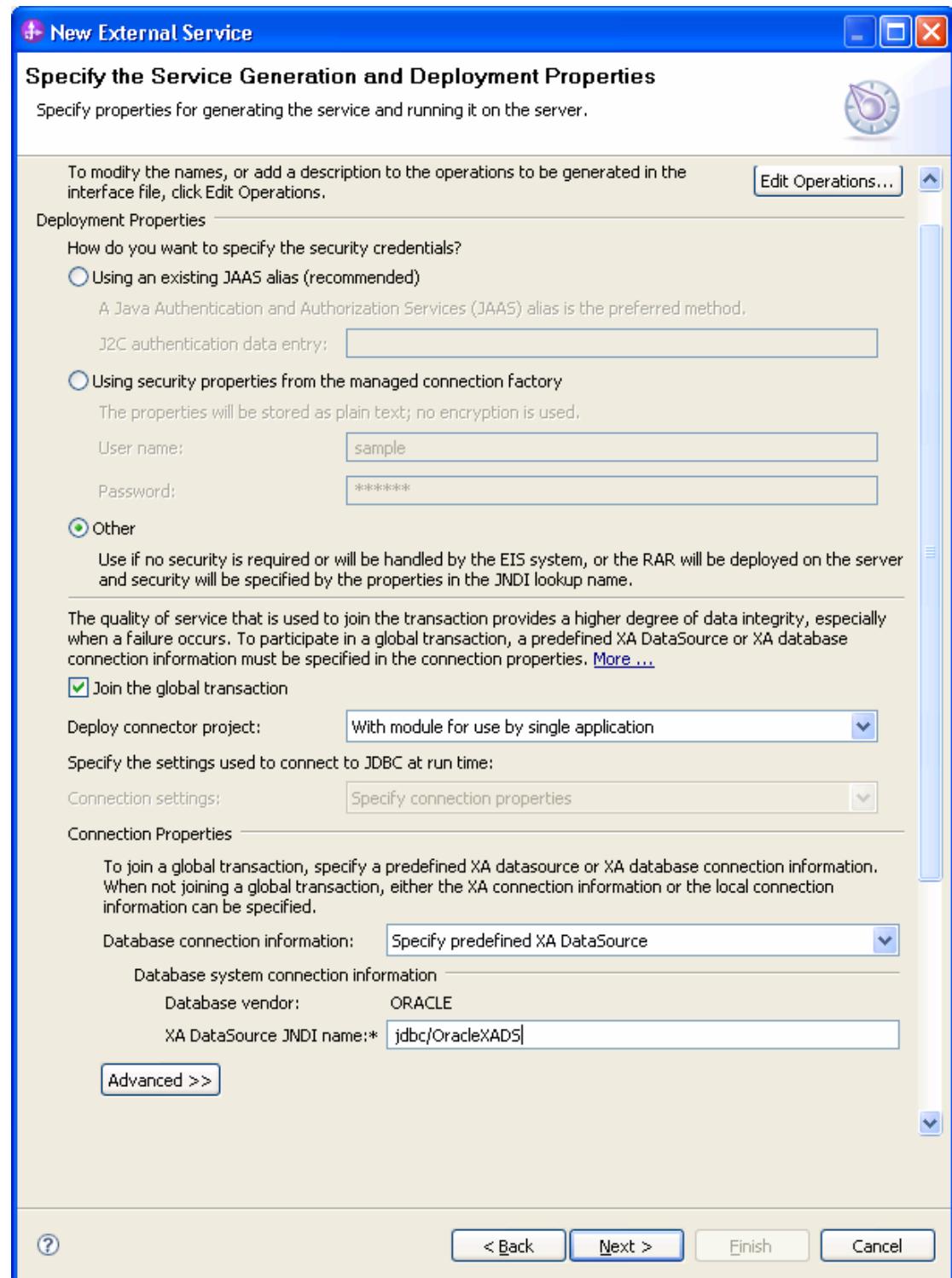
Generate a business graph for each business object

?

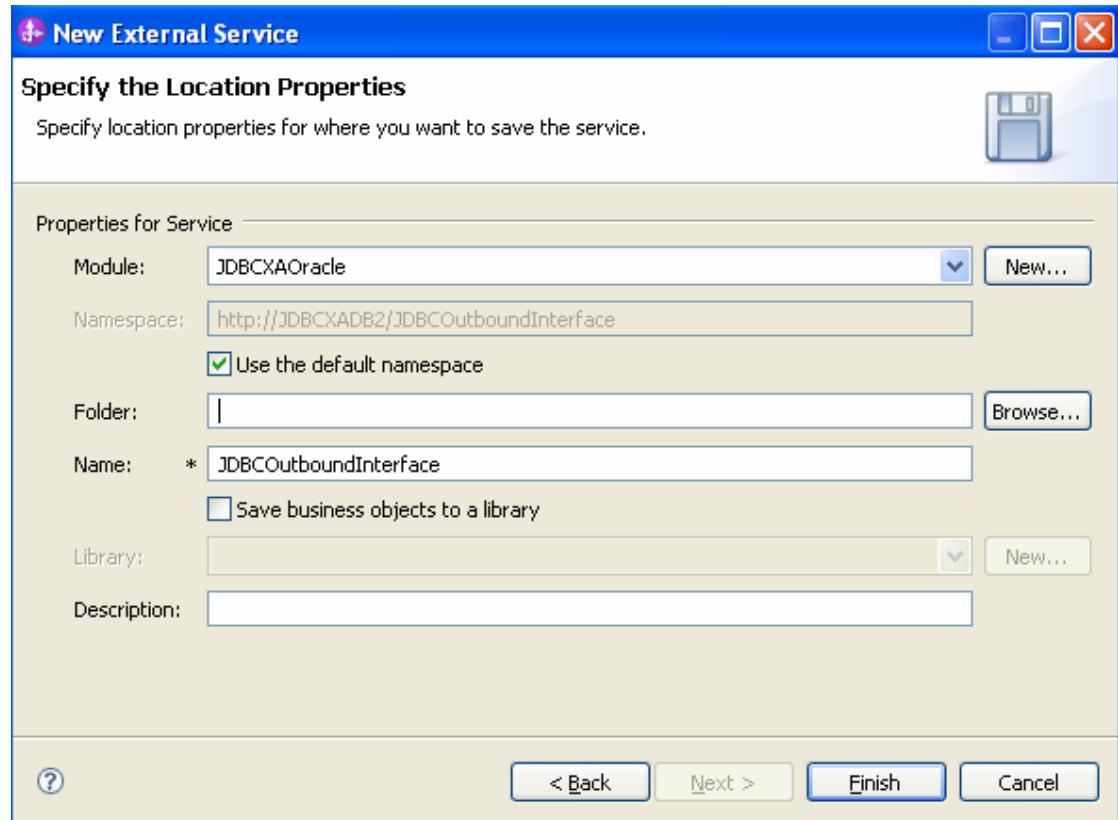
< Back Finish Cancel



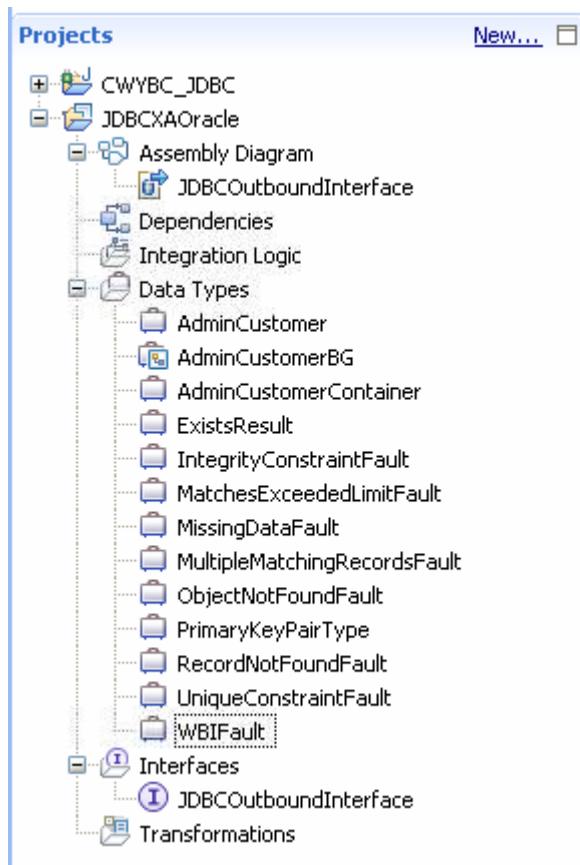
16. In **Specify the Service Generation and Deployment Properties** area, select **Other**. In the **XA DataSource JNDI name** field enter **jdbc/OracleXADS**. Click **Next**.



17. In Specify the Location Properties window, click **Finish**.

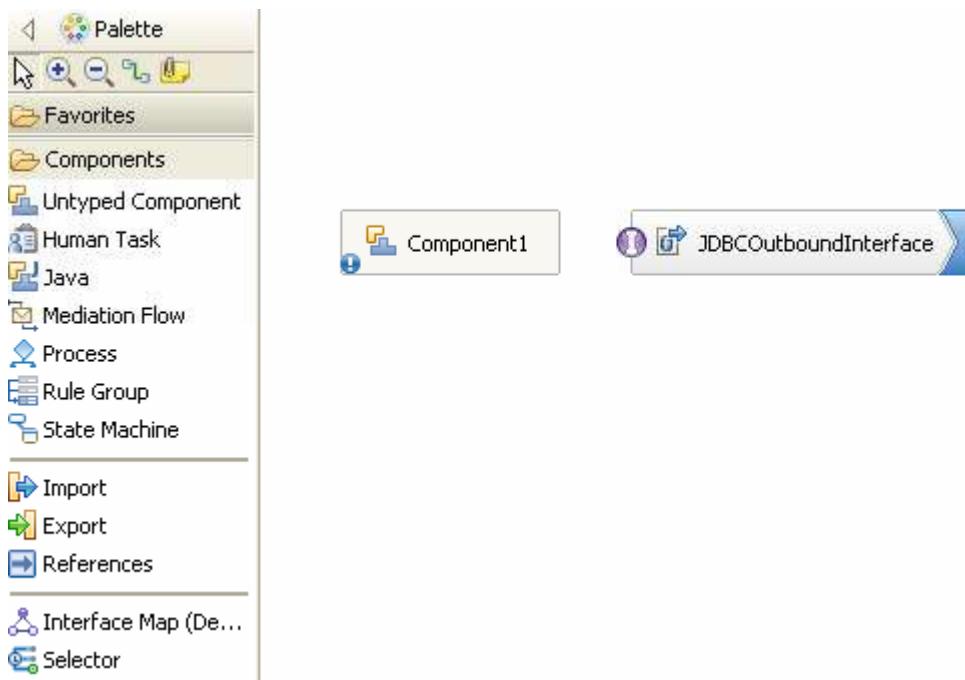


18. Verify the results shown below.

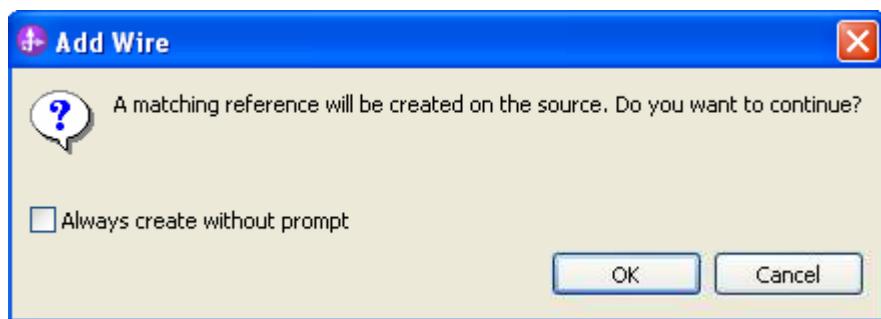


Set up the components to be part of the XA environment

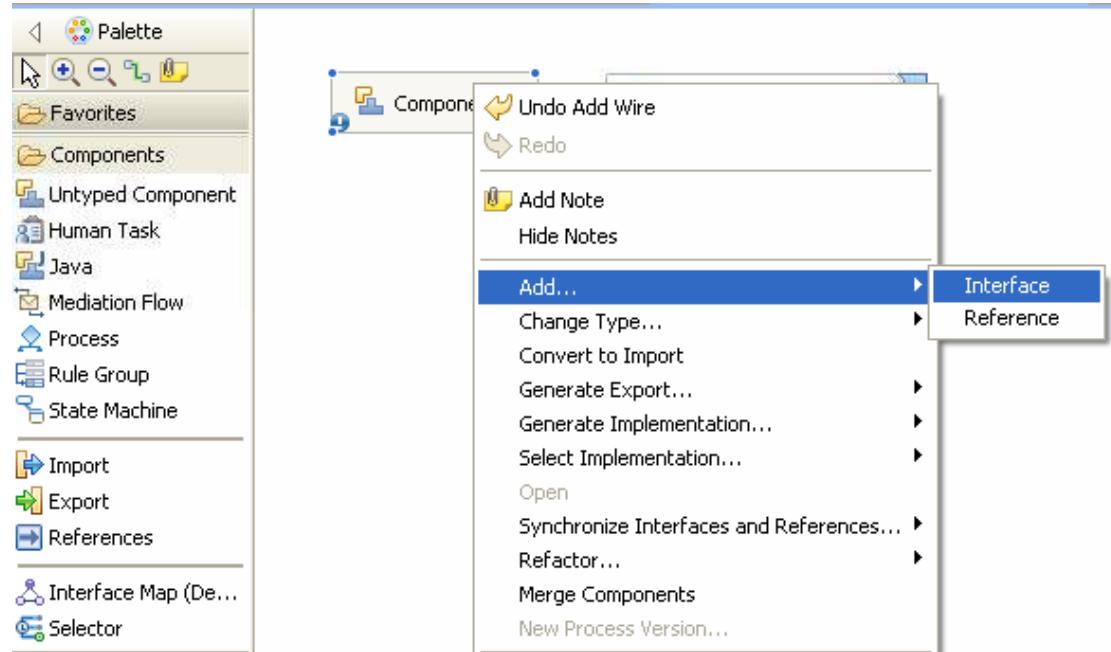
1. In the Business Integration tab, under **JDBCXAOracle** double-click **Assembly Diagram** to open it.
2. In the Palette, expand **Components** and drag **Untyped Component** to Assembly Diagram editor, and name it as **Component1**.



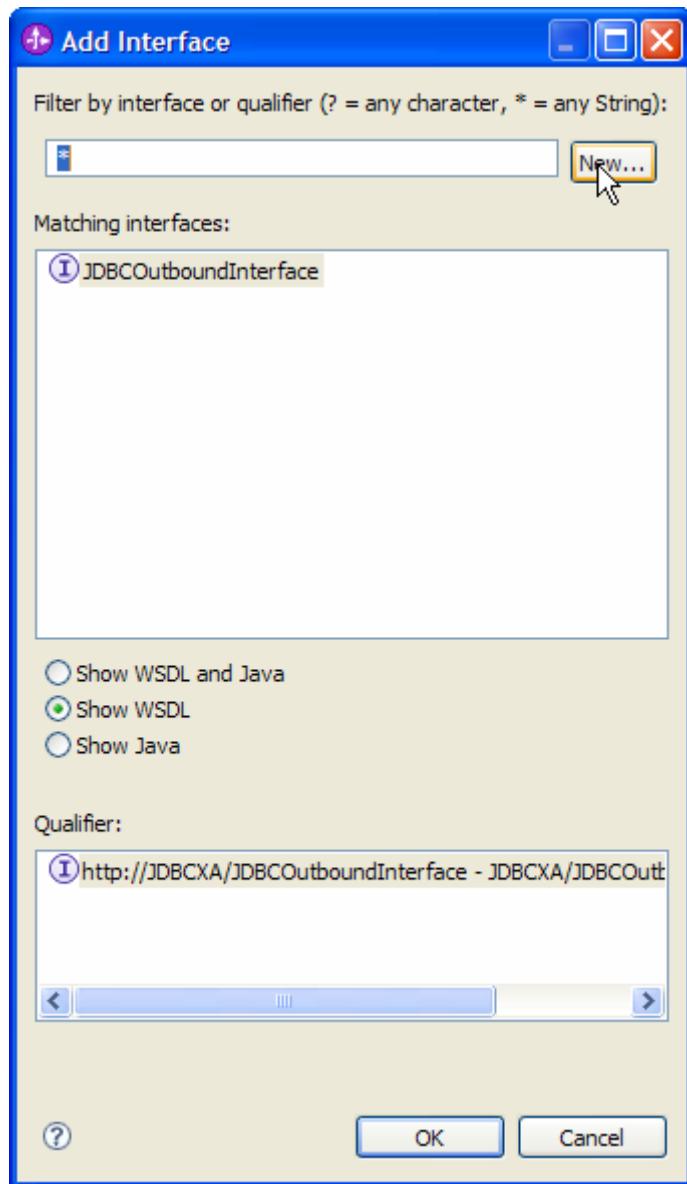
3. Wire **Component1** to **JDBCOutboundInterface**. In the Add Wire window, click **OK**.



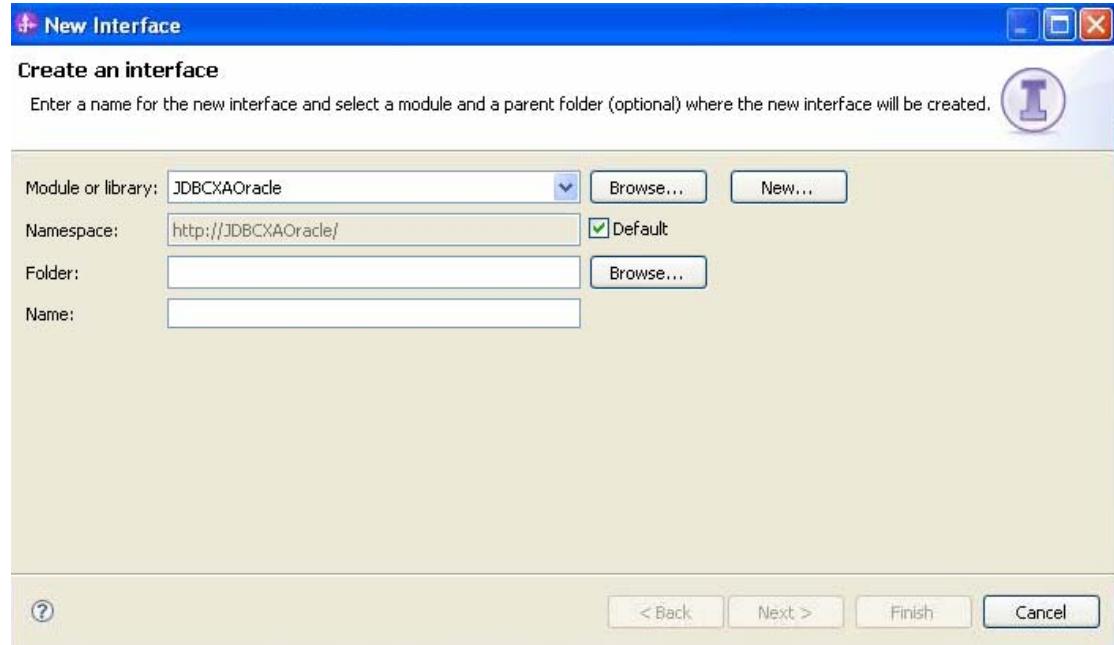
4. Right-click **Component1** and select **Add > Interface**.



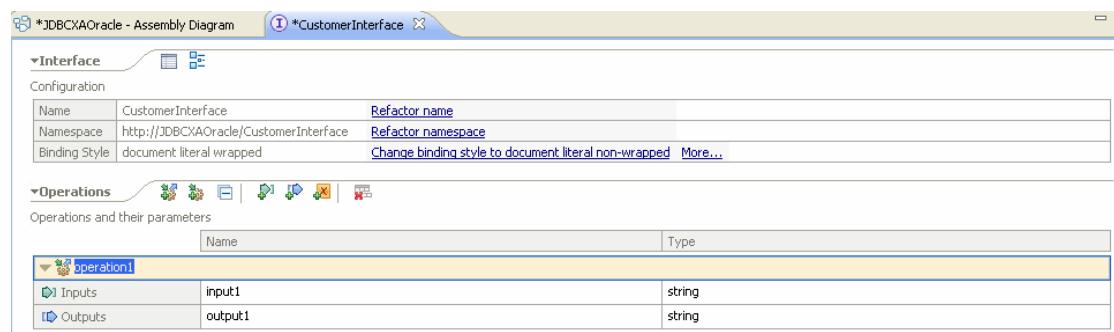
5. In the Add Interface window, click **New**.



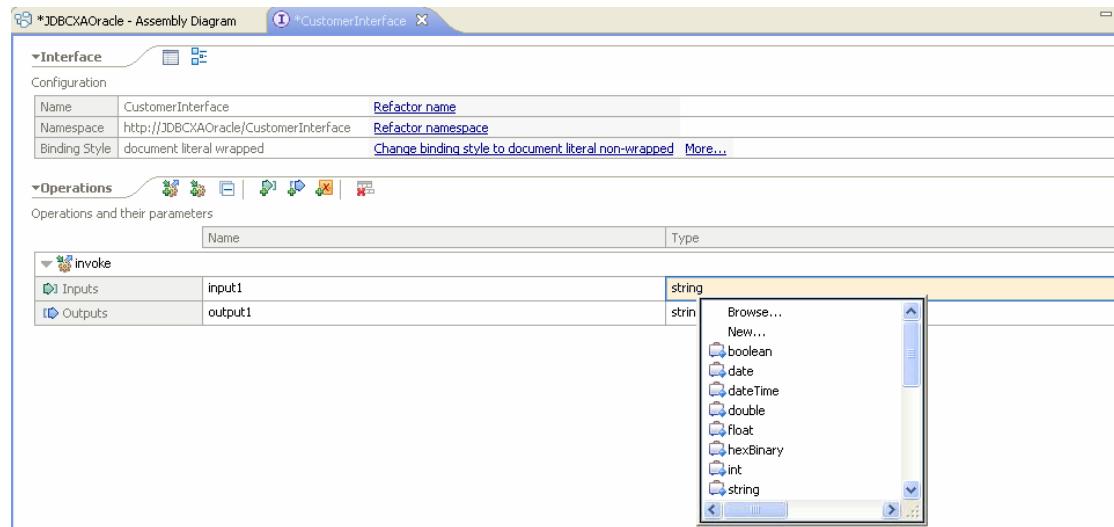
6. Enter **CustomerInterface** in the **Name** field. Click **Finish**.



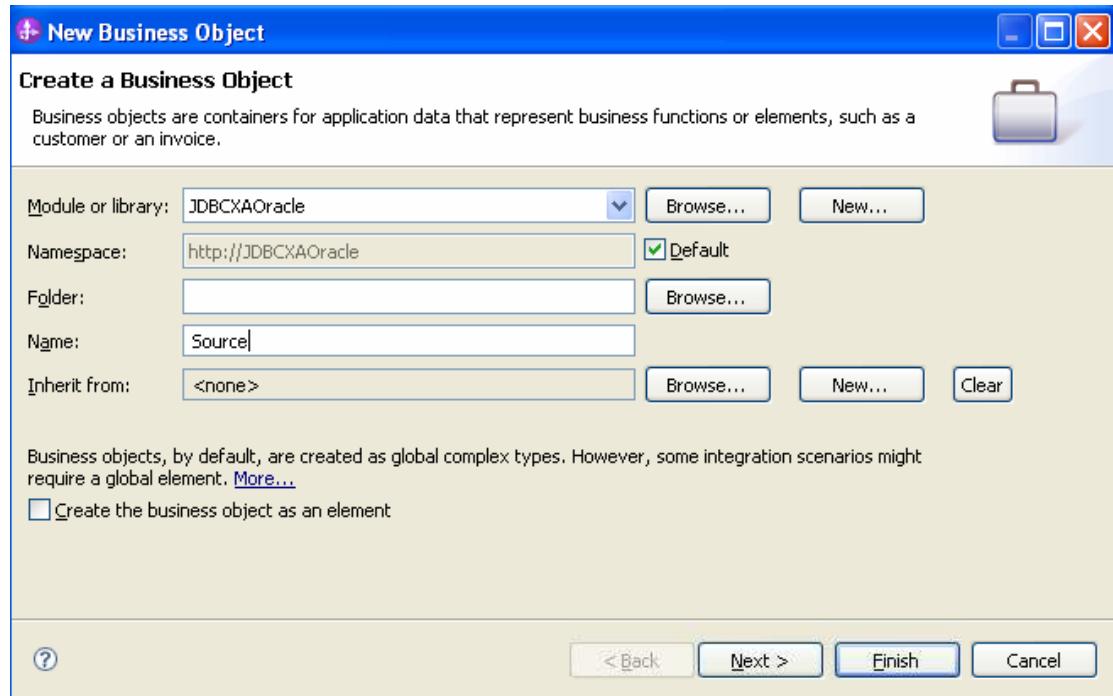
7. Click to add a new operation for **CustomerInterface** interface.



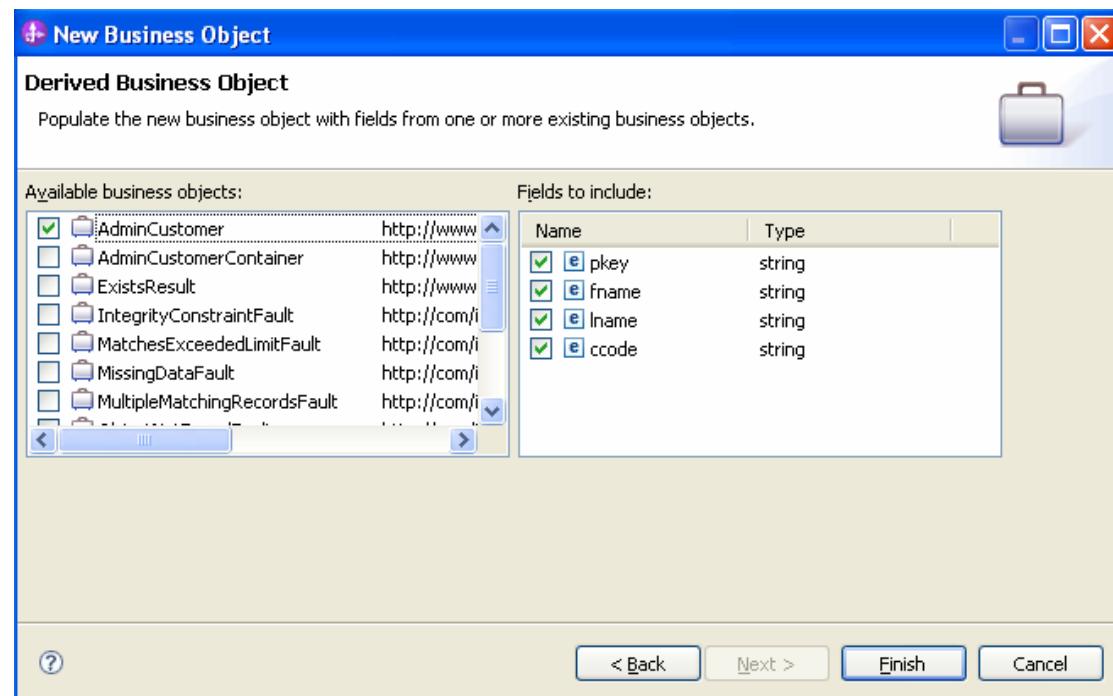
8. Rename the operation name to **invoke**. Select "Type" for the Inputs parameter, and select **New**.



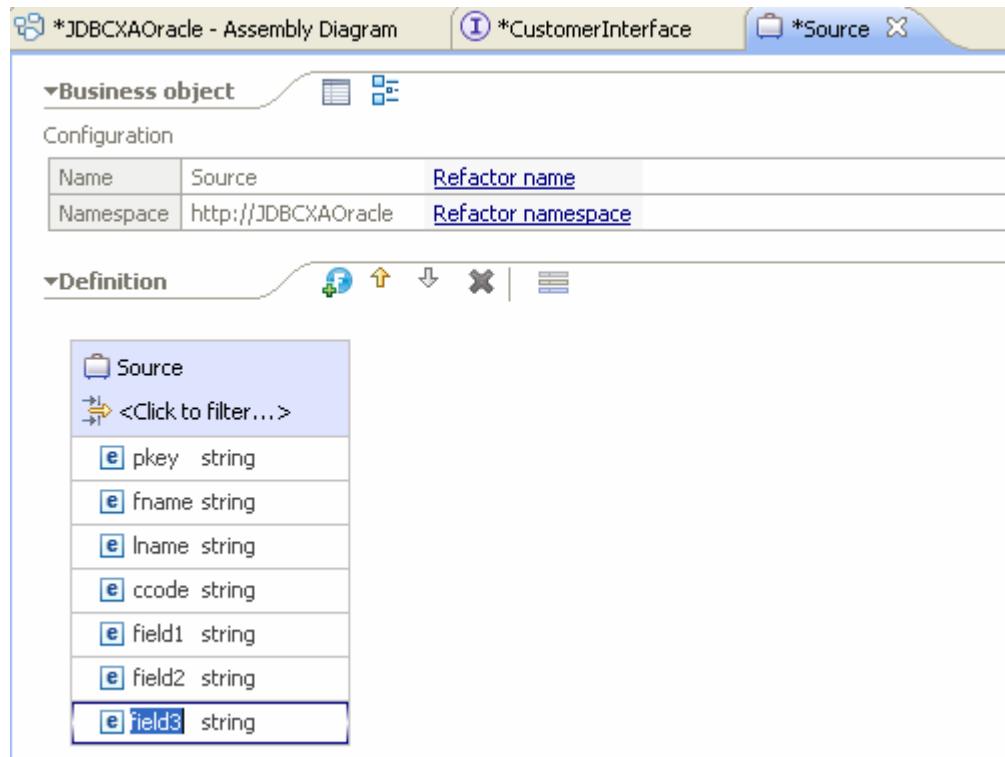
9. In the Create a Business Object window, enter **Source** in the **Name** field. Click **Next**.



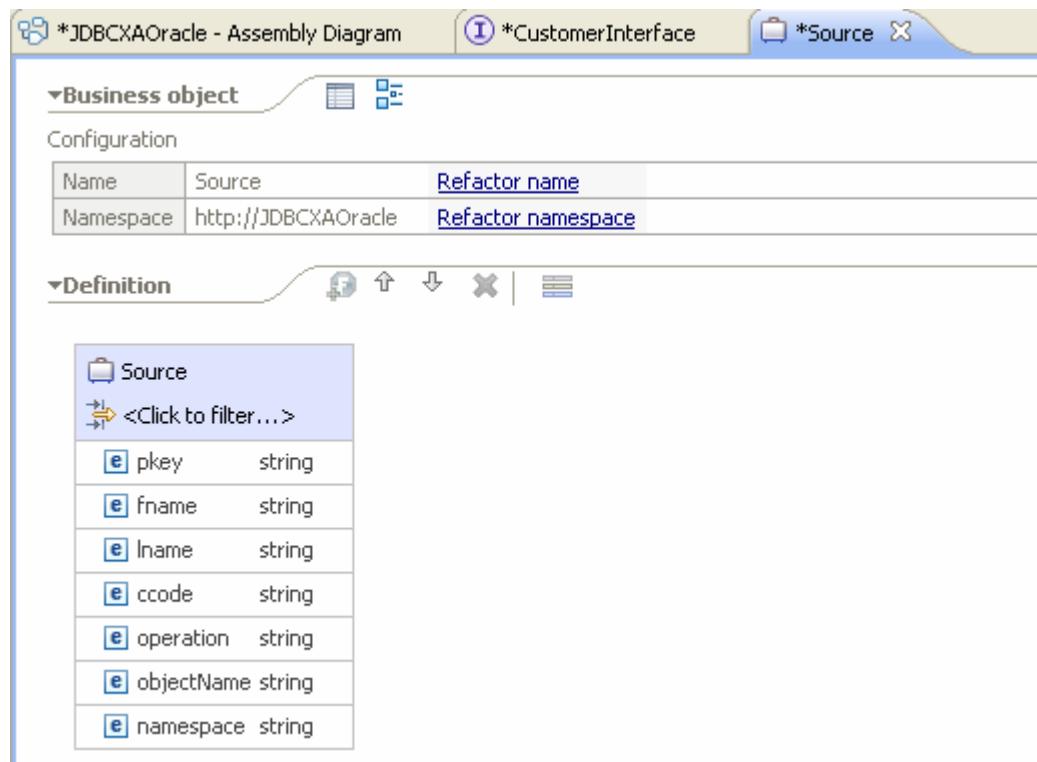
10. From the list of **Available business objects**, select **AdminCustomer** to add all Customer business objects's attributes to the Source business object. Click **Finish**.



11. In the Business Object editor, click to add three new fields for **Source** business object.



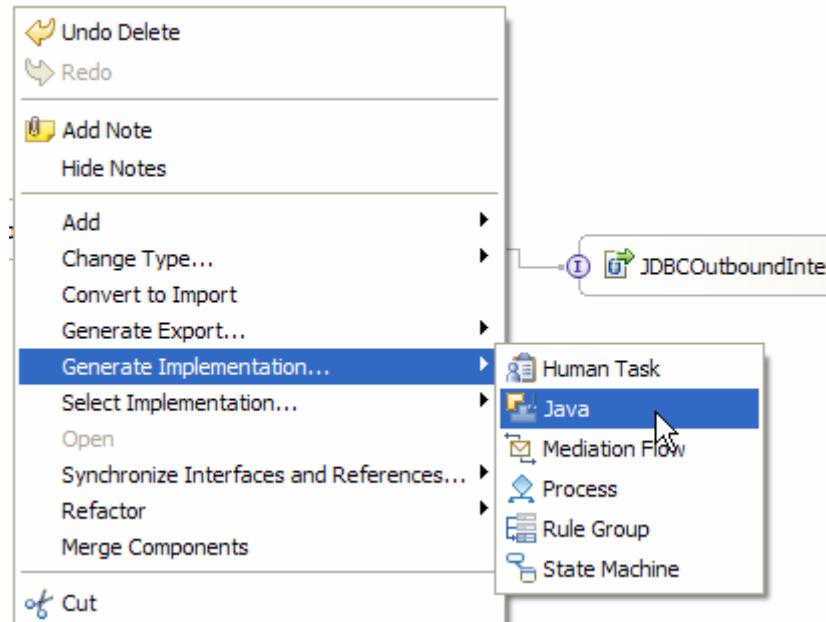
12. Rename the three new fields as **operation**, **objectName** and **namespace**.



13. Select **File->Save All** to save all the changes.

14. Right click **Component1** in the Assembly Diagram and select **Generate implementation -> Java**.

WebSphere software



15. In the Generate Implementation window, select **default package** and click **OK**.
16. Open the Component1Impl.java file in the editor and add the imports as shown in the figure below.

```
import com.ibm.websphere.sca.Service;
import com.ibm.websphere.sca.Ticket;
import commonj.sdo.DataObject;
import com.ibm.websphere.sca.ServiceManager;
import com.ibm.j2ca.base.SDOFactory;
import com.ibm.j2ca.base.exceptions.BusinessObjectDefinitionNotFoundException;

public class Component1Impl {
    /**
     * Default constructor.
     */
    public Component1Impl() {
        super();
    }
}
```

17. Add the following implementation for **invoke()** method.

```
public String invoke(DataObject input1) throws
BusinessObjectDefinitionNotFoundException {

    String objName =
input1.getString("objectName");
    String namespace =
input1.getString("namespace");
    DataObject customerBO =
SDOFactory.createDataObject(namespace, objName);
    DataObject customerBG =
customerBO.getContainer();

    customerBO.setString("pkey",
input1.getString("pkey"));
    customerBO.setString("fname",
input1.getString("fname"));
    customerBO.setString("lname",
input1.getString("lname"));
    customerBO.setString("ccode",
input1.getString("ccode"));

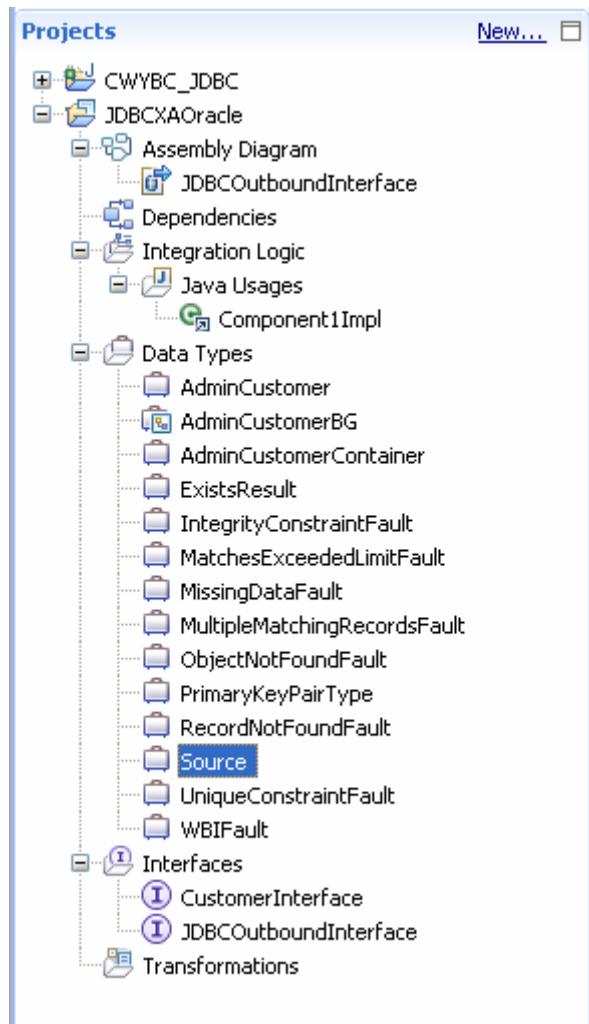
    String op = input1.getString("operation");

    String operation =
op.toLowerCase() + customerBG.getType().getName();

    locateService_JDBCOutboundInterfacePartner().in
voke(operation, customerBG);

    return "Success";
}
```

18. Select **File > Save All** to save all the changes.



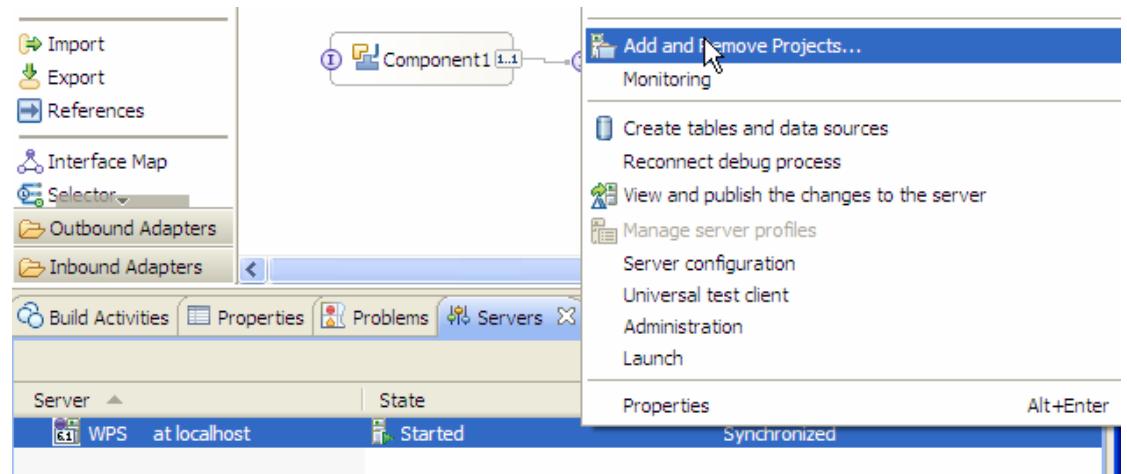
Deploy the module to the test environment

The result of running the external service wizard is an SCA module that contains an Enterprise Information System import. Install this SCA module in WebSphere Integration Developer integration test client. To do this, you must add the SCA module you created earlier to the server using the **Servers** view in WebSphere Integration Developer.

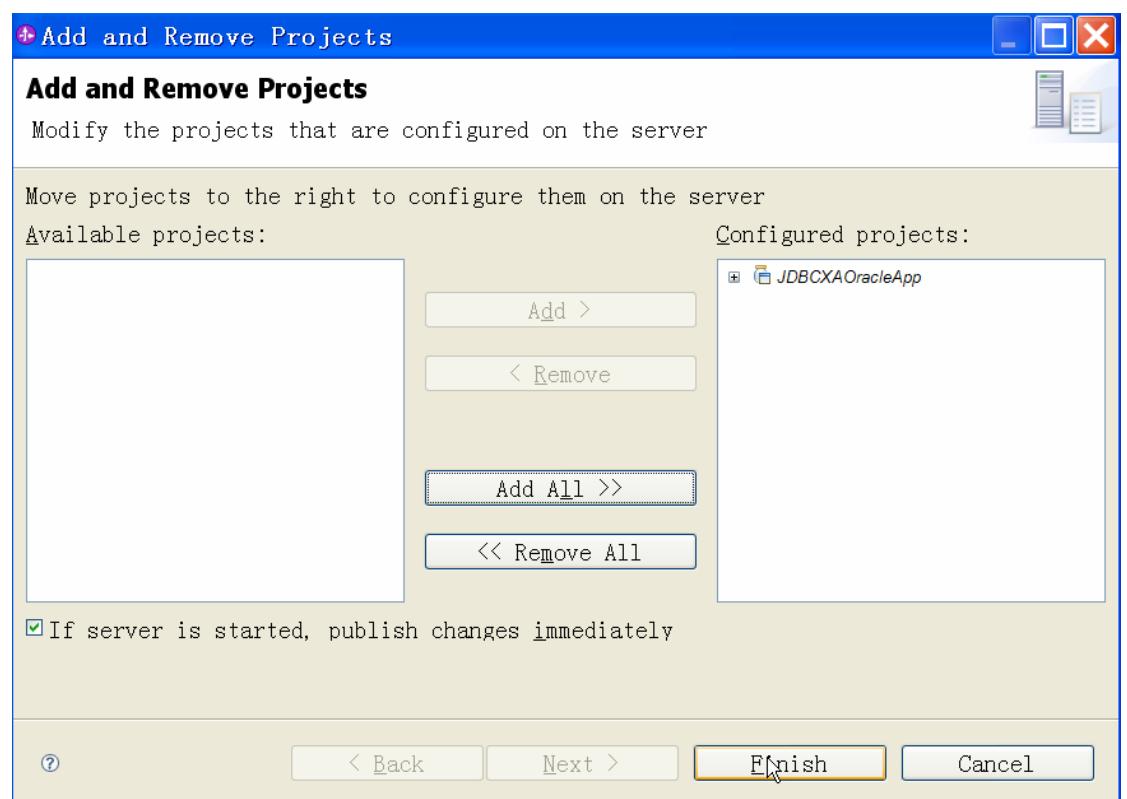
Steps for adding the SCA module to the server:

1. In WebSphere Integration Developer, switch to the **Servers** view by selecting from the toolbar **Window > Show View > Servers**.
2. In the **Servers** tab in the lower-right pane right click the server, and select **Start**.
3. After the server is started, right-click the server, and select **Add and Remove projects**.

WebSphere software



4. Add **JDBCXAOracleApp** into the **Configured projects** panel. Click **Finish**.

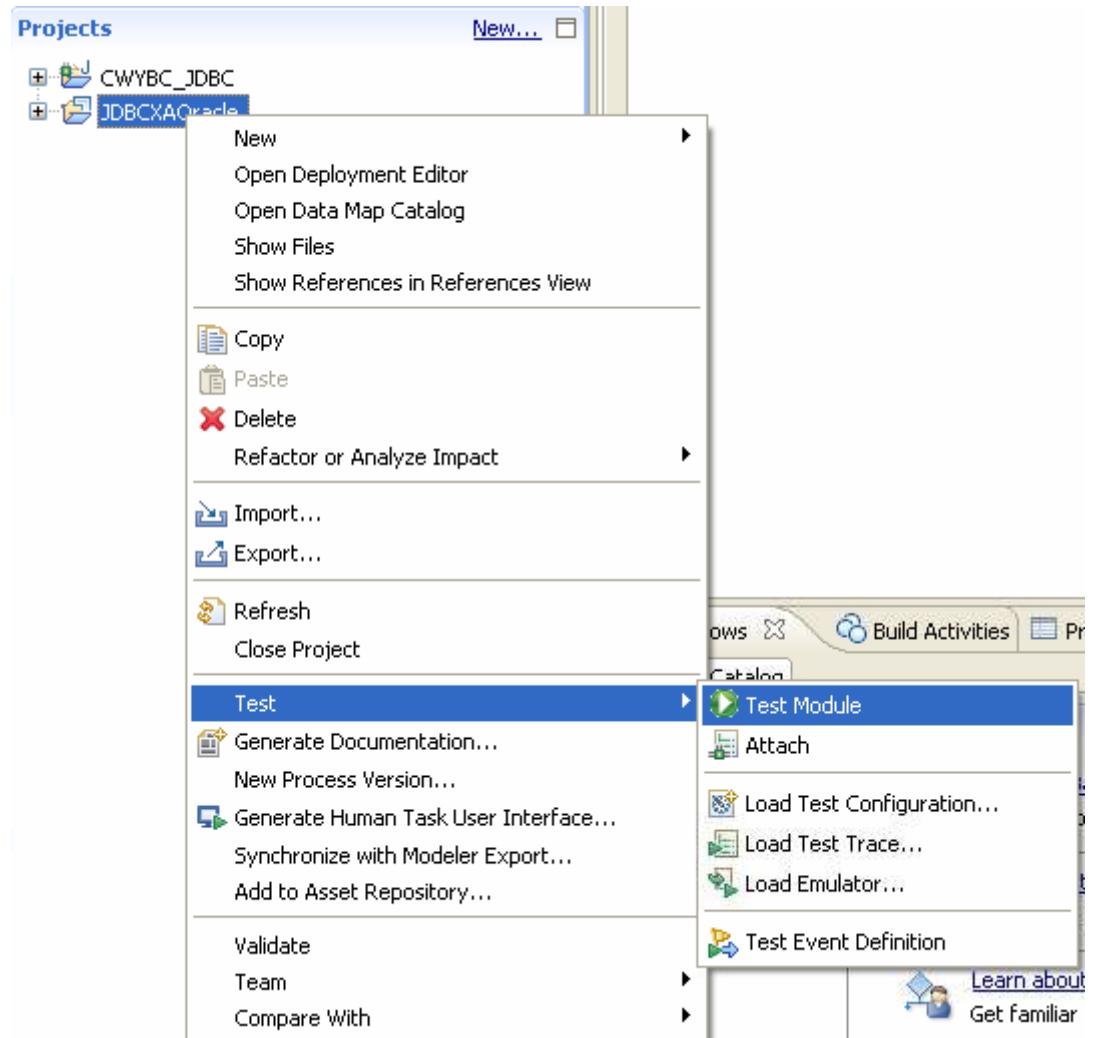


5. Click **Finish**. This deploys the project on the server. For troubleshooting issues while adding the project, see the Troubleshooting section. The Console tab in the lower-right pane displays a log while the module is being added to the server.

Test the assembled adapter application

Test the assembled adapter application using the WebSphere Integration Developer integration test client:

- From the Business Integration view, right click **JDBCXAOracle** and select **Test > Test Module**.

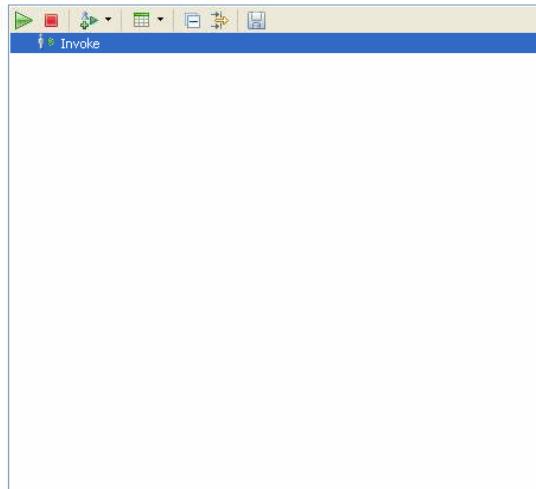


- From the **Component** list, select **Component1**. Specify the parameters as shown below.

Integration Test Client: JDBCXAOracle_Test

Events

This area displays the events in a test trace. Select an event to display its properties in the General Properties and Detailed Properties sections. [More...](#)



General Properties

Detailed Properties

Specify the component, interface, operation, and input parameter values for the Invoke event, then click the Continue icon in the Events area to run the test. [More...](#)

Configuration:	Default Module Test
Module:	JDBCXAOracle
Component:	Component1
Interface:	CustomerInterface
Operation:	invoke

Initial request parameters:

Value editor XML editor

Name	Type	Value
input1	Source	✓
pkey	string	✓ 300
fname	string	✓ abc
lname	string	✓ xyz
ccode	string	✓ IBM
operation	string	✓ create
objectName	string	✓ AdminCustomerBG
namespace	string	✓ http://www.ibm.com/xmlns/prod/websphere/j2ca/jdbc/AdminCustomerBG

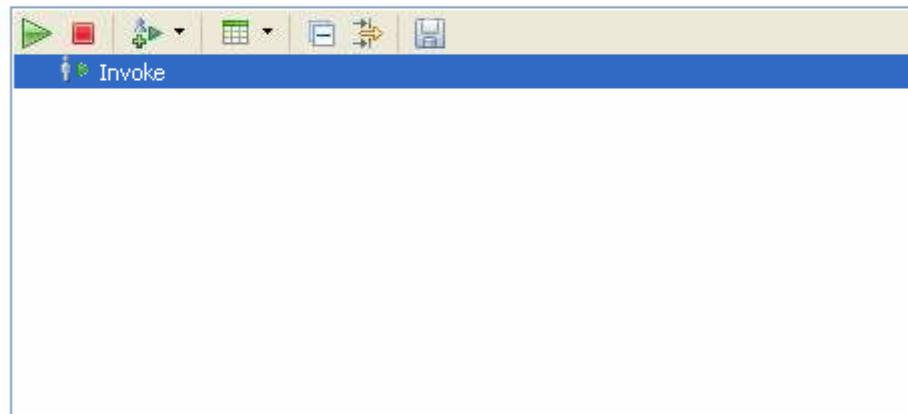
Note: Set the **operation** value to **Create**; set the **objectName** value to **AdminCustomerBG**; set the **namespace** value to **http://www.ibm.com/xmlns/prod/websphere/j2ca/jdbc/AdminCustomerBG**.

- Click to continue.

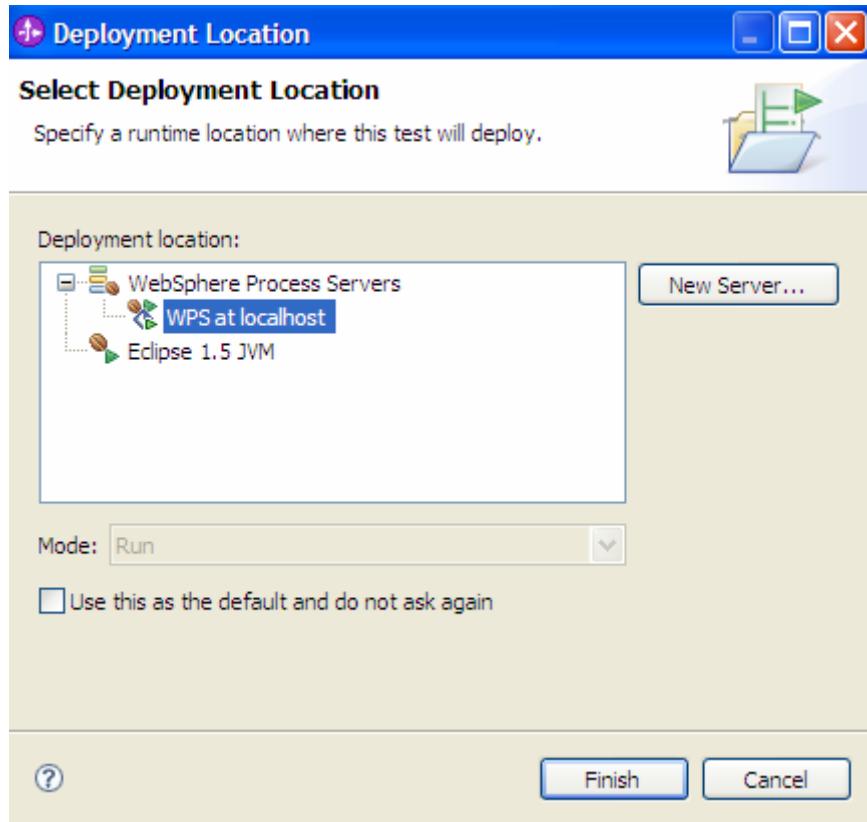
Integration Test Client: JDBCXAOracle_Test

Events

This area displays the events in a test trace. Select an event to display its properties in the General Properties and Detailed Properties sections. [More...](#)



- In the Select Deployment Location window, select the server, and click **Finish**.



- After the service is executed successfully, the customer record will be created in the target database. To verify the result, connect to the database and run the following SQL query:

```
SELECT * FROM CUSTOMER WHERE pkey = '300';
```

Clear the sample content

Return the data to its original state by deleting the Customer record you created in the CUSTOMER table by connecting to the database and running the SQL query:

```
DELETE FROM CUSTOMER WHERE pkey = '300';
```

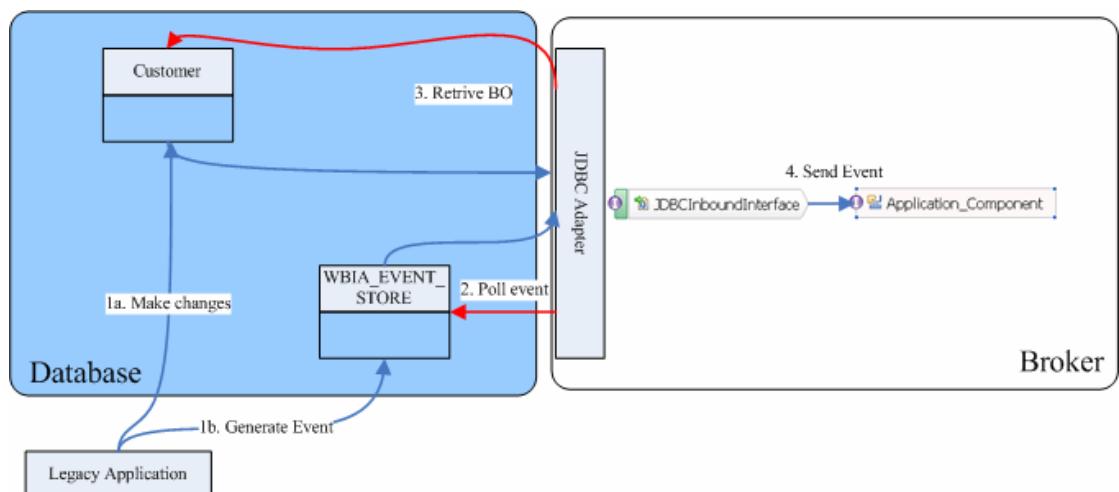
Chapter 10. Tutorial 9: Receiving events from the Oracle database using data source with prepared statement cache (inbound processing)

This tutorial demonstrates how the WebSphere Adapter for JDBC 7.0.0.0 receives events from the Oracle database using a data source with a prepared statement cache. WebSphere JDBC adapter interact with database by polling database event from an event table.

About this task

In this scenario, a legacy application makes a change to the CUSTOMER table in a single operation. Here we will insert an event record into the event table (WBIA_EVENT_TABLE). The JDBC adapter will poll the events from the database periodically. If a new event is found, it will fetch the event and corresponding business objects from database. Finally, the JDBC adapter will convert the event to a SDO and send it to the destination SCA component.

The following figure shows the whole scenario:



Prepare to run through the tutorial

Extract the sample files

Replicas of the artifacts that you create when using the external service wizard are provided as sample files for your reference. Use these files to verify if the files you create using the external service wizard are correct.

Download the sample zip file and extract it into a directory of your choice (you may want to create a new directory).

Configuration prerequisites

Before configuring the adapter, you must complete the following tasks:

- Create tables and stored procedures
- Create an authentication alias
- Create a data source

Create tables and stored procedures

You must create the following tables and stored procedures in the Oracle database before starting the scenario.

a. Script for creating the tables

```

CREATE TABLE CUSTOMER (
    PKEY VARCHAR2(10) NOT NULL PRIMARY KEY,
    FNAME VARCHAR2(20),
    LNAME VARCHAR(20),
    CCODE VARCHAR2(10) ) ;

CREATE SEQUENCE EVENT_SEQ INCREMENT BY 1 START WITH
1 MINVALUE 1 CACHE 20 ;

CREATE TABLE WBIA_JDBC_EVENTSTORE
(
    EVENT_ID INTEGER NOT NULL PRIMARY KEY,
    XID          VARCHAR2(200),
    OBJECT_KEY   VARCHAR2(80)      NOT NULL,
    OBJECT_NAME  VARCHAR2(40)      NOT NULL,
    OBJECT_FUNCTION VARCHAR2(40)      NOT
NULL,
    EVENT_PRIORITY INTEGER        NOT
NULL,
    EVENT_TIME    TIMESTAMP,
    EVENT_STATUS  INTEGER        NOT NULL,
    EVENT_COMMENT VARCHAR2(100)
) ;

```

b. Script for creating triggers for Inbound

```
CREATE OR REPLACE TRIGGER EVENT_CREATE AFTER INSERT
ON CUSTOMER
REFERENCING OLD AS O NEW AS N
FOR EACH ROW
BEGIN
INSERT INTO wbia_jdbc_eventstore (event_id,
object_key, object_name,object_function,
event_priority, event_status)
VALUES (event_seq.nextval,:N.pkey,
'SampleCustomerBG', 'Create', 1, 0);
END;
/

CREATE OR REPLACE TRIGGER EVENT_DELETE AFTER DELETE
ON CUSTOMER
REFERENCING OLD AS O NEW AS N
FOR EACH ROW
BEGIN
INSERT INTO wbia_jdbc_eventstore (event_id,
object_key, object_name,object_function,
event_priority, event_status)
VALUES (event_seq.nextval,:O.pkey,
'SampleCustomerBG', 'Delete', 1, 0);
END;
/

CREATE OR REPLACE TRIGGER EVENT_UPDATE AFTER UPDATE
OF PKEY, CCODE, FNAME, LNAME ON CUSTOMER
REFERENCING OLD AS O NEW AS N
FOR EACH ROW
BEGIN
INSERT INTO wbia_jdbc_eventstore (event_id,
object_key, object_name, object_function,
event_priority, event_status)
VALUES (event_seq.nextval,:N.pkey,
'SampleCustomerBG', 'Update', 1, 0);
END;
/
```

c. Script for inserting data into the CUSTOMER table

```
CREATE OR REPLACE PROCEDURE INSERTCUSTRECORDS AS
BEGIN
FOR cntr in 1..100 LOOP
INSERT INTO CUSTOMER (pkey,ccode,fname, lname)
values(to_char(cntr), 'ANITA','MEHTA','IBM');
End Loop;
END;
```

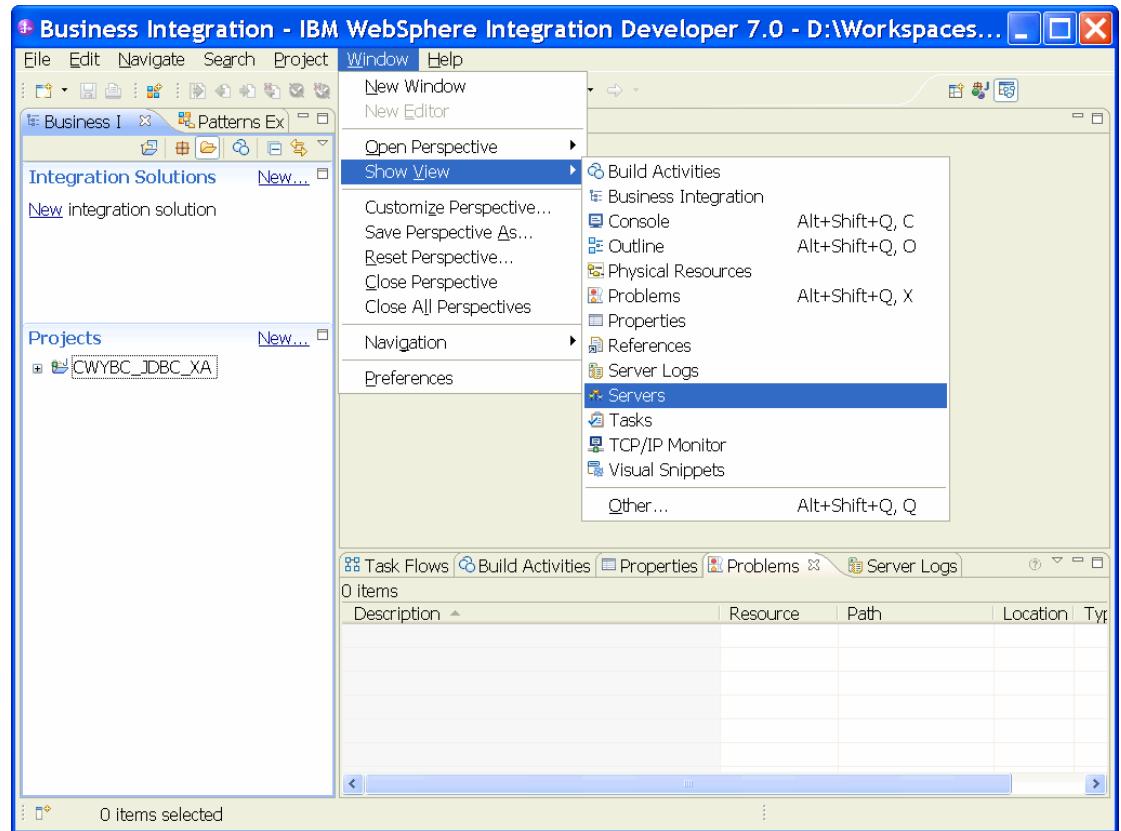
Note: After running this procedure, verify whether the records are inserted correctly into the WBIA_JDBC_EVENTSTORE table.

Create an authentication alias

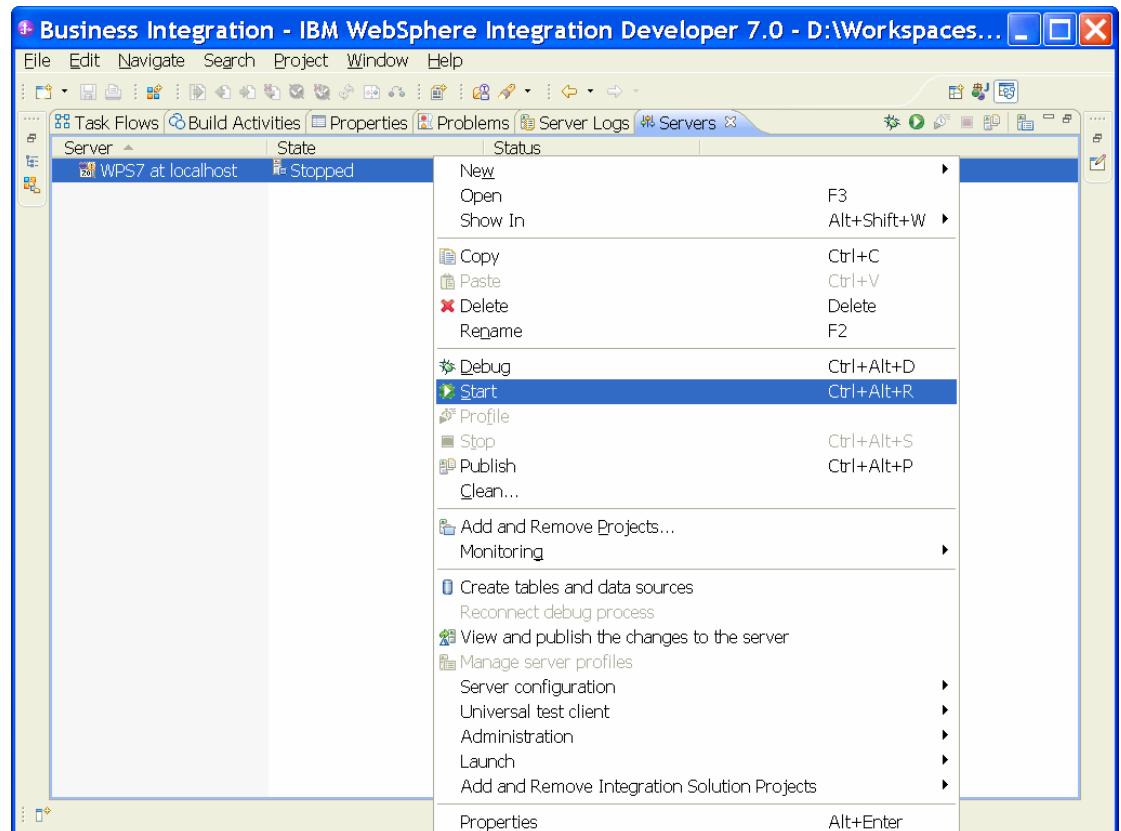
The authentication alias needs to be set because the data source created in the next section uses the username and password set in the authentication alias to connect to the database.

Follow these steps to set the authentication alias in the WebSphere Process Server administrative console.

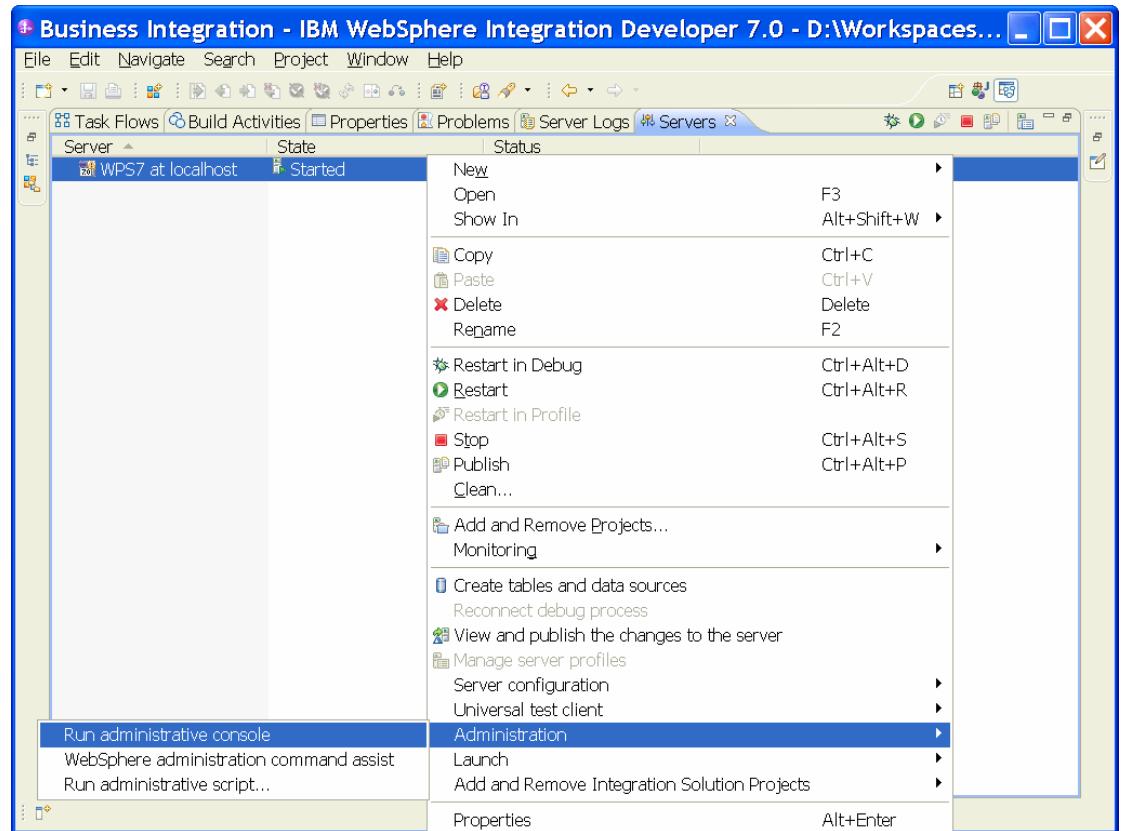
1. In WebSphere Integration Developer, switch to the **Servers** view by selecting **Windows > Show View > Servers**.



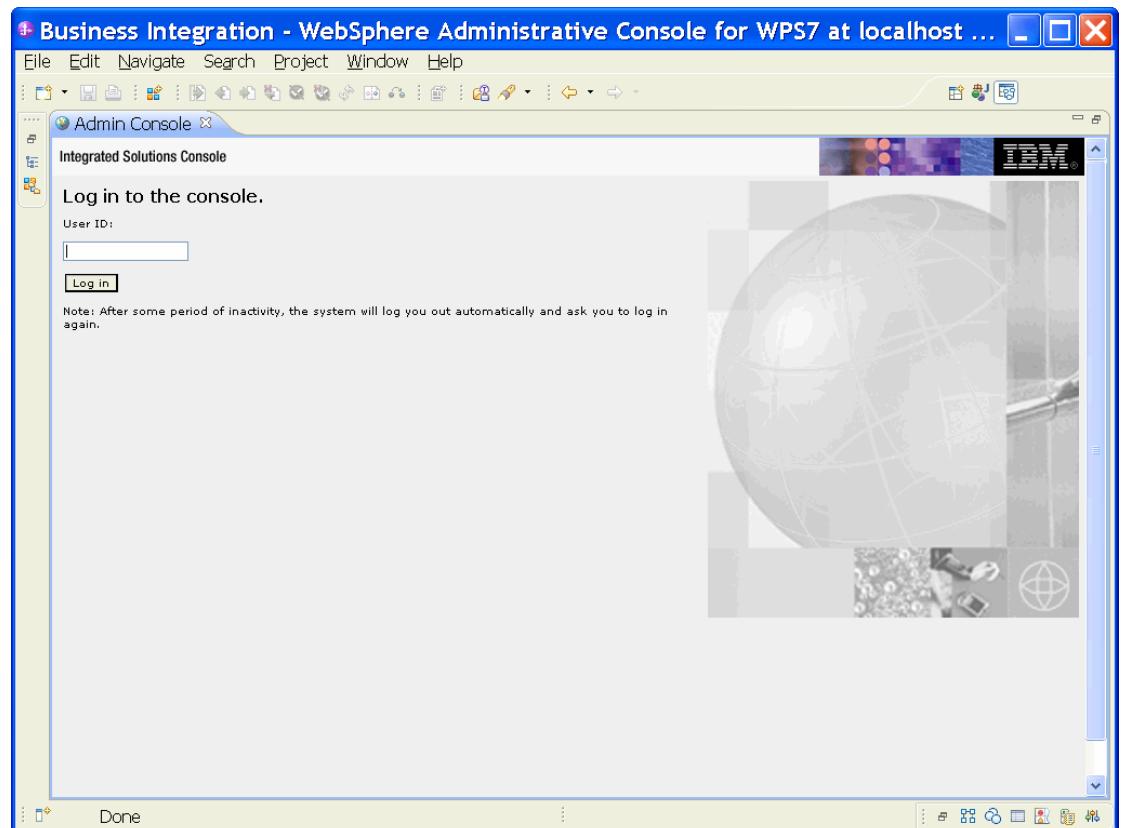
2. In the **Servers** view, right-click the server that you want to start and select **Start**.



3. After the server is started, right-click the server, and select **Administration > Run administrative console**.



4. Log on to the administrative console.



WebSphere software

5. Click **Security → Global security.**



6. Under Java Authentication and Authorization Service, click **J2C authentication data.**

Global security

Use this panel to configure administration and the default application security policy. This security configuration applies to functions and is used as a default security policy for user applications. Security domains can be defined to override and control applications.

[Security Configuration Wizard](#)[Security Configuration Report](#)**Administrative security**

- Enable administrative security
 - [Administrative user roles](#)
 - [Administrative group roles](#)
 - [Administrative authentication](#)

Application security

- Enable application security

Java 2 security

- Use Java 2 security to restrict application access to local resources
 - Warn if applications are granted custom permissions
 - Restrict access to resource authentication data

User account repository

Current realm definition

Federated repositories

Available realm definitions

Federated repositories

[Configure...](#)[Set as current](#)**Authentication**

Authentication mechanisms and expiration

[LTPA](#)[Kerberos and LTPA](#)[Kerberos configuration](#)[SWAM \(deprecated\): No authentication](#)[Authentication cache settings](#) [Web and SIP security](#) [RMI/IOP security](#) [Java Authentication and Authorization](#)[Application logins](#)[System logins](#)[J2C authentication data](#)

Specifies a list of user identities and passwords for Java(TM) 2 connector security to use.

[Security domains](#)[External authorization providers](#)[Custom properties](#)

A list of existing aliases is displayed.

WebSphere software

Global security > JAAS - J2C authentication data

Specifies a list of user identities and passwords for Java(TM) 2 connector security to use.

Prefix new alias names with the node name of the cell (for compatibility with earlier releases)

Apply

Preferences

New	Delete			
Select Alias ▾				
		User ID ▾		Description ▾
You can administer the following resources:				
<input type="checkbox"/>	BSpace JDBC Alias	wbiuser	Business Space Authorization Alias	
<input type="checkbox"/>	CommonEventInfrastructureJMSAuthAlias	wbiuser	Authentication alias for the Common Event Infrastructure JMS Topics and Queues	
<input type="checkbox"/>	SCA Auth Alias	wbiuser	This is the alias used by SCA to login to a secured SIBus	
<input type="checkbox"/>	localhostNode01Cell/nlNode01/server1/EventAuthDataAliasDerby		Derby authentication alias for the Event Server	
Total 4				

7. Click **New** to create a new authentication entry. Type the alias name, and username and password to connect to the database. Click **OK**.

WebSphere software

Cell=localhostNode01Cell, Profile=AppSrv01

Global security

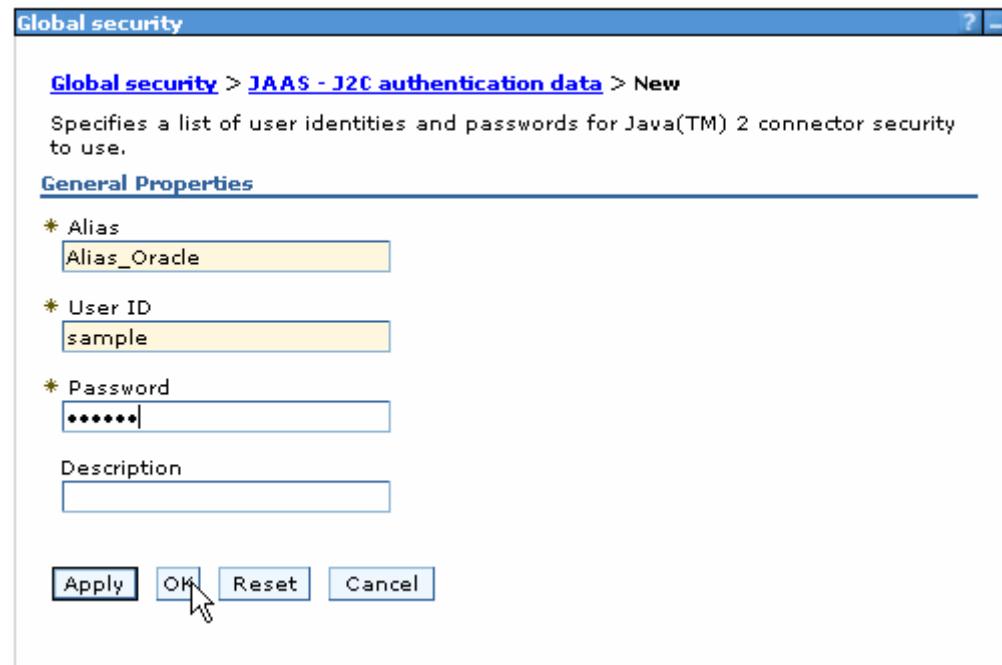
[Global security](#) > [JAAS - J2C authentication data](#) > New

Specifies a list of user identities and passwords for Java(TM) 2 connector security to use.

General Properties

* Alias: Alias_Oracle
* User ID: sample
* Password:
Description:

[Apply] [OK] [Reset] [Cancel]



8. Click **Save** to save the changes.

Cell=localhostNode01Cell, Profile=AppSrv01

Global security

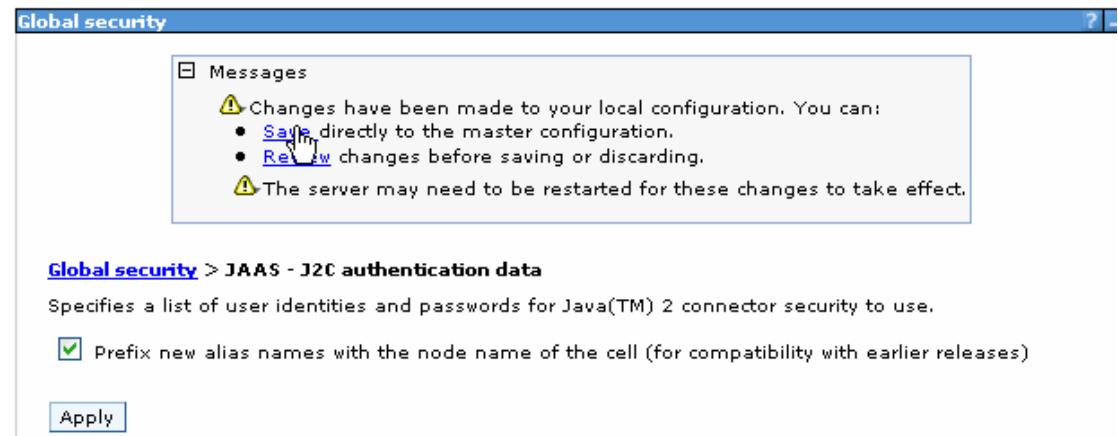
[Global security](#) > [JAAS - J2C authentication data](#)

Specifies a list of user identities and passwords for Java(TM) 2 connector security to use.

Prefix new alias names with the node name of the cell (for compatibility with earlier releases)

[Apply]

You have created an authentication alias that will be used to configure the data source.



Preferences			
	New	Delete	
Select	Alias	User ID	Description
You can administer the following resources:			
<input type="checkbox"/>	BSpace JDBC Alias	wbiuser	Business Space Authorization Alias
<input type="checkbox"/>	CommonEventInfrastructureJMSAuthAlias	wbiuser	Authentication alias for the Common Event Infrastructure JMS Topics and Queues
<input type="checkbox"/>	SCA Auth Alias	wbiuser	This is the alias used by SCA to login to a secured SIBus
<input type="checkbox"/>	localhostNode01Cell/nlNode01/server1/EventAuthDataAliasDerby		Derby authentication alias for the Event Server
<input type="checkbox"/>	nlNode01/AliasOracle	luweiqin	
Total 5			

Create a data source

Create a data source in WebSphere Process Server, which the adapter will use to connect to the database. This data source is used later when generating the artifacts for the module.

Note: This tutorial uses Oracle as the database and the Oracle thin driver,ojdbc6.jar.

Here are the steps to create the data source in the WebSphere Process Server administrative console.

1. In the administrative console, select **Environment → WebSphere Variables**.

WebSphere software



2. On the right page, select **ORACLE_JDBC_DRIVER_PATH** and specify the path of the ojdbc6.jar file in the **Value** field. Click **OK**.

WebSphere software

Cell=localhostNode01Cell, Profile=AppSrv01

WebSphere Variables

[WebSphere Variables](#) > ORACLE_JDBC_DRIVER_PATH

Use this page to define substitution variables. Variables specify a level of indirection for some system-defined values, such as file system root directories. Variables have a scope level, which is either server, node, cluster, or cell. Values at one scope level can differ from values at other levels. When a variable has conflicting scope values, the more granular scope value overrides values at greater scope levels. Therefore, server variables override node variables, which override cluster variables, which override cell variables.

Configuration

General Properties

* Name: ORACLE_JDBC_DRIVER_PATH

Value: D:\Lib

Description: The directory that contains the Oracle thin or oci8 JDBC Driver.

Buttons: Apply, OK, Reset, Cancel

3. Click **Save** to save the changes.

WebSphere Variables

Messages

⚠ Changes have been made to your local configuration. You can:
• [Save](#) directly to the master configuration.
• [Review](#) changes before saving or discarding.
⚠ The server may need to be restarted for these changes to take effect.

The variable has been added and appears in the list.

Preferences

New Delete

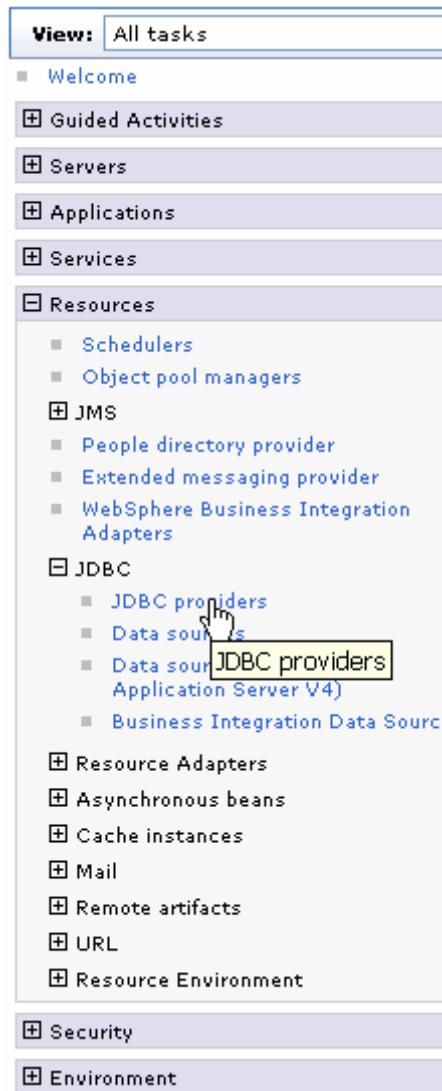
Select Name Value Scope

You can administer the following resources:

Select	Name	Value	Scope
You can administer the following resources:			
<input type="checkbox"/>	MQ_INSTALL_ROOT	`\${WAS_INSTALL_ROOT}/lib/WMQ	Node=n1Node01
<input type="checkbox"/>	ORACLE_JDBC_DRIVER_PATH	D:\Lib	Node=n1Node01
<input type="checkbox"/>	OS400_NATIVE_JDBC40_DRIVER_PATH		Node=n1Node01
<input type="checkbox"/>	OS400_NATIVE_JDBC_DRIVER_PATH		Node=n1Node01
<input type="checkbox"/>	OS400_TOOLBOX_JDBC_DRIVER_PATH		Node=n1Node01
<input type="checkbox"/>	ORACLE_JDBC_DRIVER_PATH		Node=n1Node01

WebSphere software

4. Select **Resources** → **JDBC** -> **JDBC Providers**.



5. Click **New** in the JDBC providers window.

JDBC providers

JDBC providers

Use this page to edit properties of a JDBC provider. The JDBC provider object encapsulates the specific JDBC driver implementation class for access to the specific vendor database of your environment. Learn more about this task in a [guided activity](#). A guided activity provides a list of task steps and more general information about the topic.

Scope: Cell=**localhostNode01Cell**, Node=**n1Node01**

Scope specifies the level at which the resource definition is visible. For detailed information on what scope is and how it works, [see the scope settings help](#).

Node=n1Node01

Preferences

Select	Name	Scope	Description
None			
Total 0			

6. Select an Oracle database with a connection pool data source for the Oracle JDBC driver. Click **Next**.

Create a new JDBC Provider

Create a new JDBC Provider

→ Step 1: Create new JDBC provider

Step 2: Enter database class path information

Step 3: Summary

Create new JDBC provider

Set the basic configuration values of a JDBC provider, which encapsulates the specific vendor JDBC driver implementation classes that are required to access the database. The wizard fills in the name and the description fields, but you can type different values.

Scope

cells:localhostNode01Cell:nodes:n1Node01

* Database type

Oracle

* Provider type

Oracle JDBC Driver

* Implementation type

Connection pool data source

* Name

Oracle JDBC Driver

Description

Oracle JDBC Driver

Next **Cancel**

7. In the Enter database classpath information page, enter the following value for the **Class path** field:
\$(ORACLE_JDBC_DRIVER_PATH)/ojdbc6.jar, where
\$(ORACLE_JDBC_DRIVER_PATH) is library path for the run time.
8. Click **Next**.

WebSphere software

Cell=localhostNode01Cell, Profile=AppSrv01

Create a new JDBC Provider

Step 1: Create new JDBC provider
→ Step 2: Enter database class path information
Step 3: Summary

Enter database class path information

Set the environment variables that represent the JDBC driver class files, which WebSphere(R) Application Server uses to define your JDBC provider. This wizard page displays the file names; you supply only the directory locations of the files. Use complete directory paths when you type the JDBC driver file locations. For example: C:\SQLLIB\java on Windows(R) or /home/db2inst1/sqllib/java on Linux(TM).

If a value is specified for you, you may click Next to accept the value.

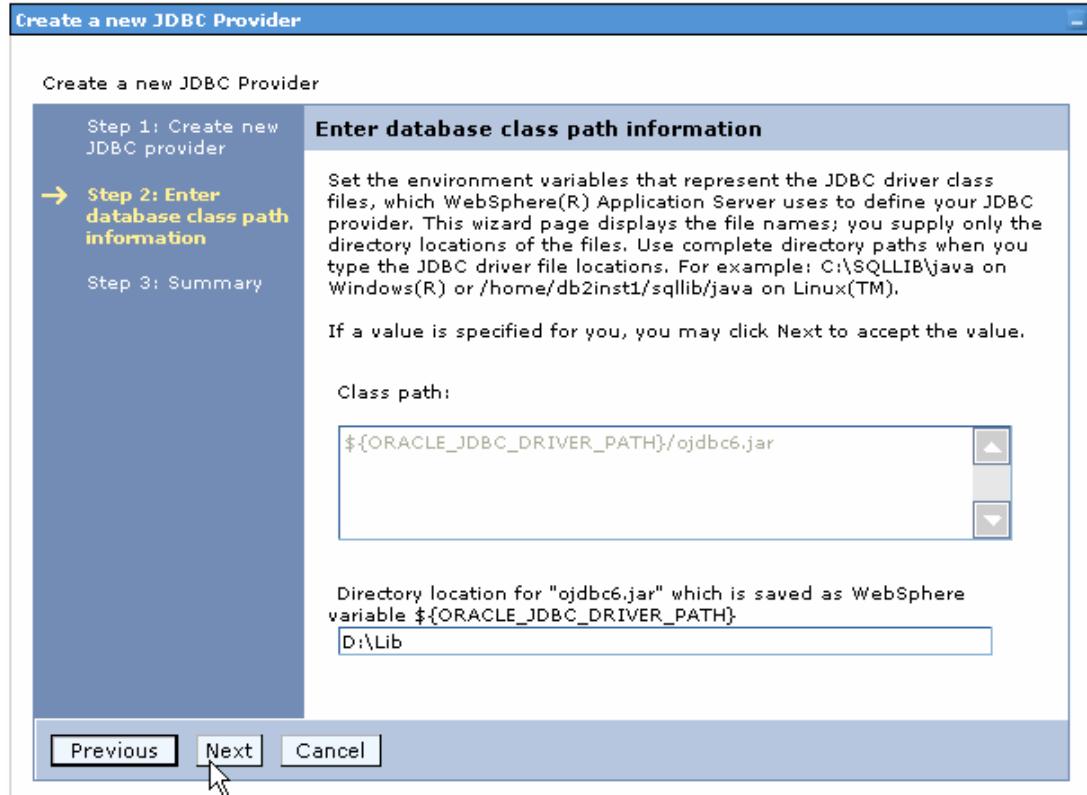
Class path:

`${ORACLE_JDBC_DRIVER_PATH}/ojdbc6.jar`

Directory location for "ojdbc6.jar" which is saved as WebSphere variable `${ORACLE_JDBC_DRIVER_PATH}`

D:\Lib

Previous **Next** Cancel



9. In the Summary page, click **Finish**.

Cell=localhostNode01Cell, Profile=AppSrv01

[Close page](#)

Create a new JDBC Provider

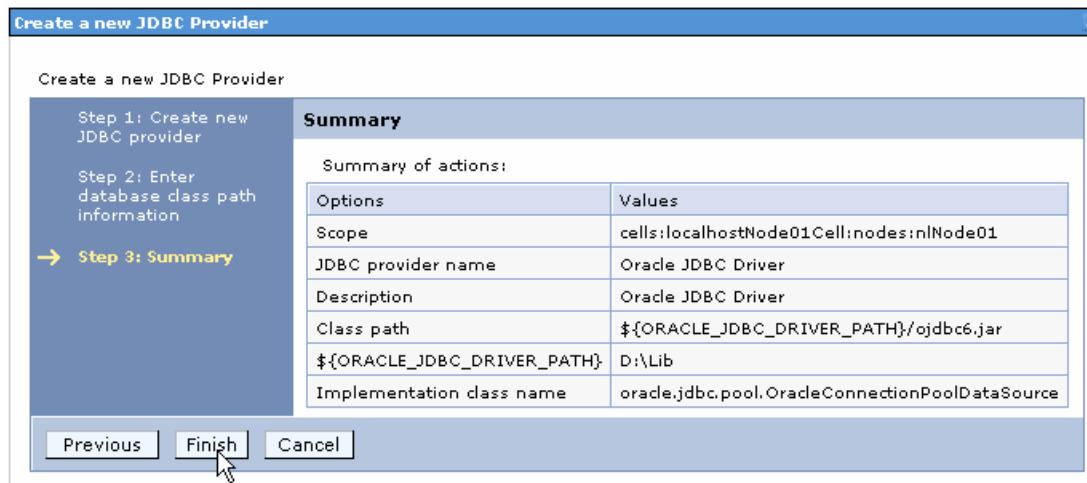
Step 1: Create new JDBC provider
Step 2: Enter database class path information
→ Step 3: Summary

Summary

Summary of actions:

Options	Values
Scope	cells:localhostNode01Cell:nodes:n1Node01
JDBC provider name	Oracle JDBC Driver
Description	Oracle JDBC Driver
Class path	<code> \${ORACLE_JDBC_DRIVER_PATH}/ojdbc6.jar</code>
<code> \${ORACLE_JDBC_DRIVER_PATH}</code>	D:\Lib
Implementation class name	oracle.jdbc.pool.OracleConnectionPoolDataSource

Previous **Finish** Cancel



10. Click **Save**.

WebSphere software

Cell=localhostNode01Cell, Profile=AppSrv01

JDBC providers

Messages

Changes have been made to your local configuration.
You can:

- Save directly to the master configuration.
- Review changes before saving or discarding.

The server may need to be restarted for these changes to take effect.

The JDBC provider is added and appears in the list.

Cell=localhostNode01Cell, Profile=AppSrv01

JDBC providers

Use this page to edit properties of a JDBC provider. The JDBC provider object encapsulates the specific JDBC driver implementation class for access to the specific vendor database of your environment. Learn more about this task in a [guided activity](#). A guided activity provides a list of task steps and more general information about the topic.

Scope: Cell=**localhostNode01Cell**, Node=**nINode01**

Scope specifies the level at which the resource definition is visible. For detailed information on what scope is and how it works, [see the scope settings help](#).

Node=nINode01

Preferences

New	Delete		
<input type="checkbox"/>	<input type="checkbox"/>		
<input type="checkbox"/>	<input type="checkbox"/>		
Select	Name ▲	Scope ▲	Description ▲
You can administer the following resources:			
<input type="checkbox"/>	Oracle JDBC Driver	Node=nINode01	Oracle JDBC Driver
Total 1			

11. Select the Oracle JDBC provider you just created. Under **Additional Properties**, click **Data sources**. Click **New**.

WebSphere software

Cell=localhostNode01Cell, Profile=AppSrv01

JDBC providers > Oracle JDBC Driver > Data sources

Use this page to edit the settings of a datasource that is associated with your selected JDBC provider. The datasource object supplies your application with connections for accessing the database. Learn more about this task in a [guided activity](#). A guided activity provides a list of task steps and more general information about the topic.

[+] Preferences

New Delete Test connection Manage state...

Select Name JNDI name Scope Provider Description Category

None Total 0

12. Type any value in the **JNDI name** field, and select the authentication alias. Click **Next**.

Cell=localhostNode01Cell, Profile=AppSrv01

Create a data source

→ Step 1: Enter basic data source information

Step 2: Enter database specific properties for the data source

Step 3: Setup security aliases

Step 4: Summary

Enter basic data source information

Set the basic configuration values of a datasource for association with your JDBC provider. A datasource supplies the physical connections between the application server and the database.

Requirement: Use the Datasources (WebSphere(R) Application Server V4) console pages if your applications are based on the Enterprise JavaBeans(TM) (EJB) 1.0 specification or the Java(TM) Servlet 2.2 specification.

Scope
cells:localhostNode01Cell:nodes:n1Node01

JDBC provider name
Oracle JDBC Driver

* Data source name
Oracle JDBC Driver DataSource

* JNDI name
OracleDS

Next Cancel

13. Provide the appropriate URL value and select a data store helper class name from the **Data store helper class name** list as shown in the following figure. Click **Next**.

WebSphere software

Cell=localhostNode01Cell, Profile=AppSrv01

Create a data source

Step 1: Enter basic data source information

→ Step 2: Enter database specific properties for the data source

Step 3: Setup security aliases

Step 4: Summary

Enter database specific properties for the data source

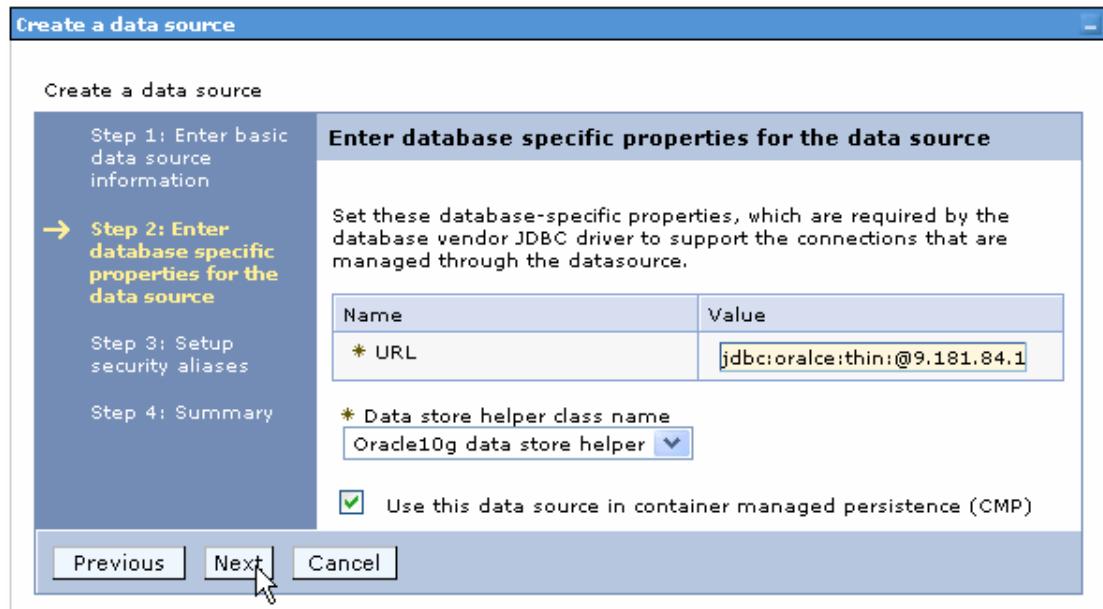
Set these database-specific properties, which are required by the database vendor JDBC driver to support the connections that are managed through the datasource.

Name	Value
* URL	jdbc:oracle:thin:@9.181.84.1

* Data store helper class name
Oracle10g data store helper

Use this data source in container managed persistence (CMP)

Previous Next Cancel



14. Select the authentication alias you just created from the **Component-managed authentication alias** field and click **Next**.

Cell=localhostNode01Cell, Profile=AppSrv01

Create a data source

Step 1: Enter basic data source information

Step 2: Enter database specific properties for the data source

→ Step 3: Setup security aliases

Step 4: Summary

Setup security aliases

Select the authentication values for this resource.

Component-managed authentication alias
nlNode01/Alias_Oracle

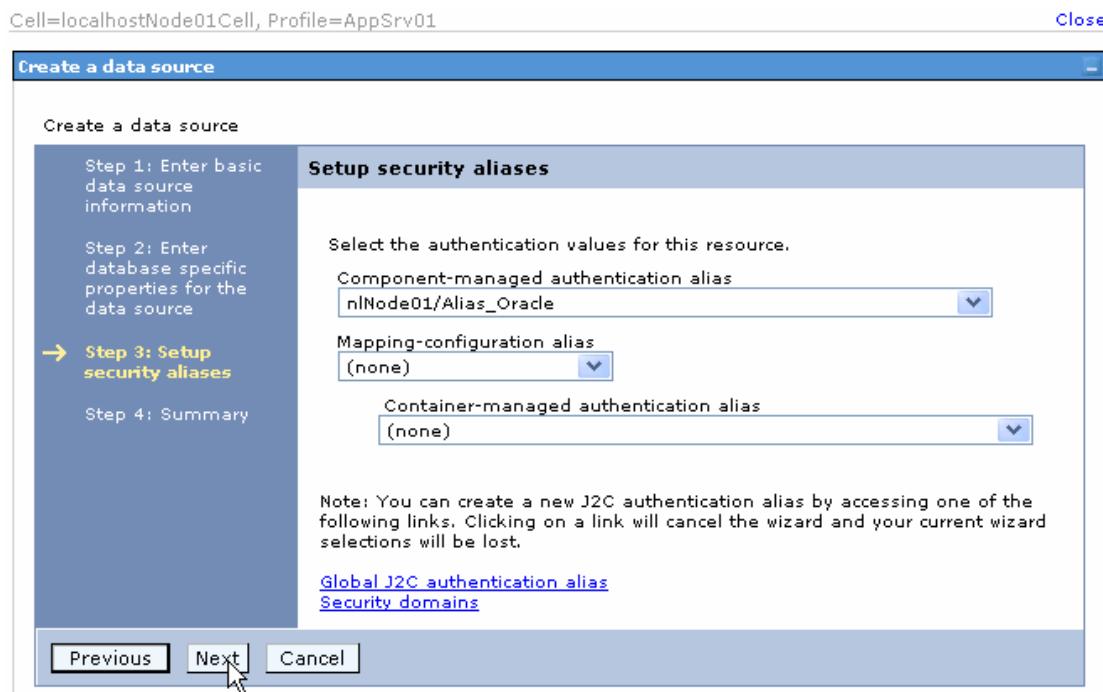
Mapping-configuration alias
(none)

Container-managed authentication alias
(none)

Note: You can create a new J2C authentication alias by accessing one of the following links. Clicking on a link will cancel the wizard and your current wizard selections will be lost.

[Global J2C authentication alias](#)
[Security domains](#)

Previous Next Cancel



15. In the Summary page, review the values entered for the data source and click **Finish**.

WebSphere software

Create a data source

Step 1: Enter basic data source information

Step 2: Enter database specific properties for the data source

Step 3: Setup security aliases

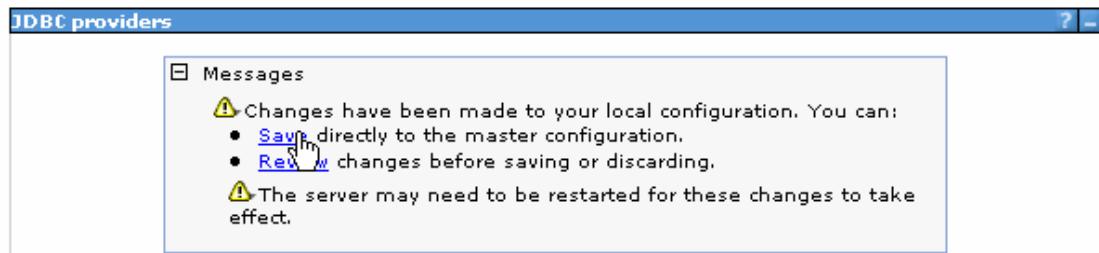
→ Step 4: Summary

Summary	
Summary of actions:	
Options	Values
Scope	cells:localhostNode01Cell:nodes:n1Node01
Data source name	Oracle JDBC Driver DataSource
JNDI name	OracleDS
Select an existing JDBC provider	Oracle JDBC Driver
Implementation class name	oracle.jdbc.pool.OracleConnectionPoolDataSource
URL	jdbc:oracle:thin:@9.181.84.136:1521:ord
Data store helper class name	com.ibm.websphere.radapter.Oracle10gDataStoreHelper
Use this data source in container managed persistence (CMP)	true
Component-managed authentication alias	n1Node01/Alias_Oracle
Mapping-configuration alias	(none)
Container-managed authentication alias	(none)

Previous **Finish** **Cancel**

16. Click **Save** to save the changes.

Cell=localhostNode01Cell, Profile=AppSrv01



17. Select the data source you just created and click **Test connection**.

WebSphere software

Cell=localhostNode01Cell, Profile=AppSrv01

JDBC providers

[JDBC providers](#) > [Oracle JDBC Driver](#) > Data sources

Use this page to edit the settings of a datasource that is associated with your selected JDBC provider. The datasource object supplies your application with connections for accessing the database. Learn more about this task in a [guided activity](#). A guided activity provides a list of task steps and more general information about the topic.

[+] Preferences

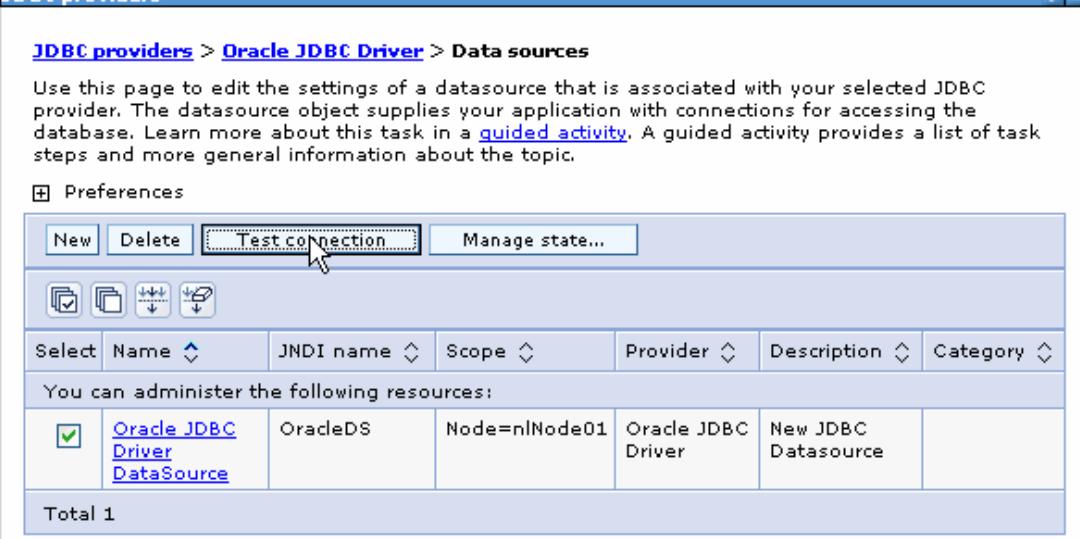
New Delete Test connection Manage state...

Select Name JNDI name Scope Provider Description Category

You can administer the following resources:

<input checked="" type="checkbox"/>	Oracle JDBC Driver DataSource	OracleDS	Node=n1Node01	Oracle JDBC Driver	New JDBC Datasource	
-------------------------------------	---	----------	---------------	--------------------	---------------------	--

Total 1



The connection should succeed as indicated by the message shown in the following figure. If you experience problems with the test connection, refer to the "Troubleshooting" section.

Cell=localhostNode01Cell, Profile=AppSrv01

JDBC providers

[JDBC providers](#) > [Oracle JDBC Driver](#) > Data sources

The test connection operation for data source Oracle JDBC Driver DataSource on server server1 at node n1Node01 was successful.

[+] Preferences

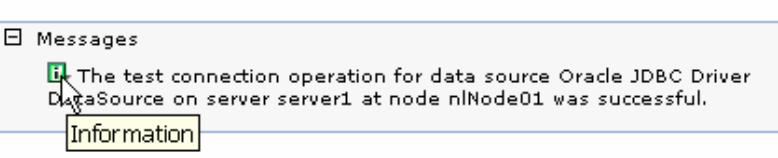
New Delete Test connection Manage state...

Select Name JNDI name Scope Provider Description Category

You can administer the following resources:

<input type="checkbox"/>	Oracle JDBC Driver DataSource	OracleDS	Node=n1Node01	Oracle JDBC Driver	New JDBC Datasource	
--------------------------	---	----------	---------------	--------------------	---------------------	--

Total 1



The data source is created and it will be used by the adapter to connect to the database.

Configure data source statement cache

1. Click the data source you just created.

JDBC providers

JDBC providers[JDBC providers](#) > [Oracle JDBC Driver](#) > Data sources

Use this page to edit the settings of a data source that is associated with your selected JDBC provider. The data source object supplies your application with connections for accessing the database. Learn more about this task in a [guided activity](#). A guided activity provides a list of task steps and more general information about the topic.

Preferences

[New](#) [Delete](#) [Test connection](#) [Manage state...](#)

Select	Name	JNDI name	Scope	Provider	Description	Category
<input type="checkbox"/>	Oracle JDBC Driver DataSource	OracleDS	Node=nlNode01	Oracle JDBC Driver	New JDBC Datasource	

Total 1

WebSphere software

2. On the right, under **Additional Properties** click **WebSphere Application Server data source properties**.



3. In the **Statement cache size** field, enter the value 20. Click **OK**.

Configuration

General Properties

Statement cache size
20 statements

Enable multithreaded access detection

Enable database reauthentication

Enable JMS one-phase optimization support

Manage cached handles

Log missing transaction context

Pretest connection properties

Pretest existing pooled connections

Retry interval
0 seconds

Pretest new connections

Number of retries
100

Retry interval
3 seconds

Pretest SQL string
SELECT 1 FROM DUAL

Apply OK Reset Cancel

4. In the Messages area, click **Save**.

Messages

⚠ Changes have been made to your local configuration. You can:

- Save directly to the master configuration.
- Review changes before saving or discarding.

⚠ The server may need to be restarted for these changes to take effect.

5. Select the data source you just created and click **Test Connection**.

JDBC providers

JDBC providers > Oracle JDBC Driver > Data sources

Use this page to edit the settings of a data source that is associated with your selected JDBC provider. The data source object supplies your application with connections for accessing the database. Learn more about this task in a [guided activity](#). A guided activity provides a list of task steps and more general information about the topic.

Preferences

New	Delete	Test connection	Manage state...														
<table border="1"> <thead> <tr> <th>Select</th> <th>Name</th> <th>JNDI name</th> <th>Scope</th> <th>Provider</th> <th>Description</th> <th>Category</th> </tr> </thead> <tbody> <tr> <td><input checked="" type="checkbox"/></td> <td>Oracle JDBC Driver DataSource</td> <td>OracleDS</td> <td>Node=nlNode01</td> <td>Oracle JDBC Driver</td> <td>New JDBC Datasource</td> <td></td> </tr> </tbody> </table>				Select	Name	JNDI name	Scope	Provider	Description	Category	<input checked="" type="checkbox"/>	Oracle JDBC Driver DataSource	OracleDS	Node=nlNode01	Oracle JDBC Driver	New JDBC Datasource	
Select	Name	JNDI name	Scope	Provider	Description	Category											
<input checked="" type="checkbox"/>	Oracle JDBC Driver DataSource	OracleDS	Node=nlNode01	Oracle JDBC Driver	New JDBC Datasource												
Total 1																	

The connection test should succeed as indicated by the message shown in the figure below. For troubleshooting issues while testing the connection, see the Troubleshooting section.

JDBC providers

JDBC providers > Oracle JDBC Driver > Data sources

Use this page to edit the settings of a data source that is associated with your selected JDBC provider. The data source object supplies your application with connections for accessing the database. Learn more about this task in a [guided activity](#). A guided activity provides a list of task steps and more general information about the topic.

Messages

The test connection operation for data source Oracle JDBC Driver DataSource on server server1 at node nlNode01 was successful.

Preferences

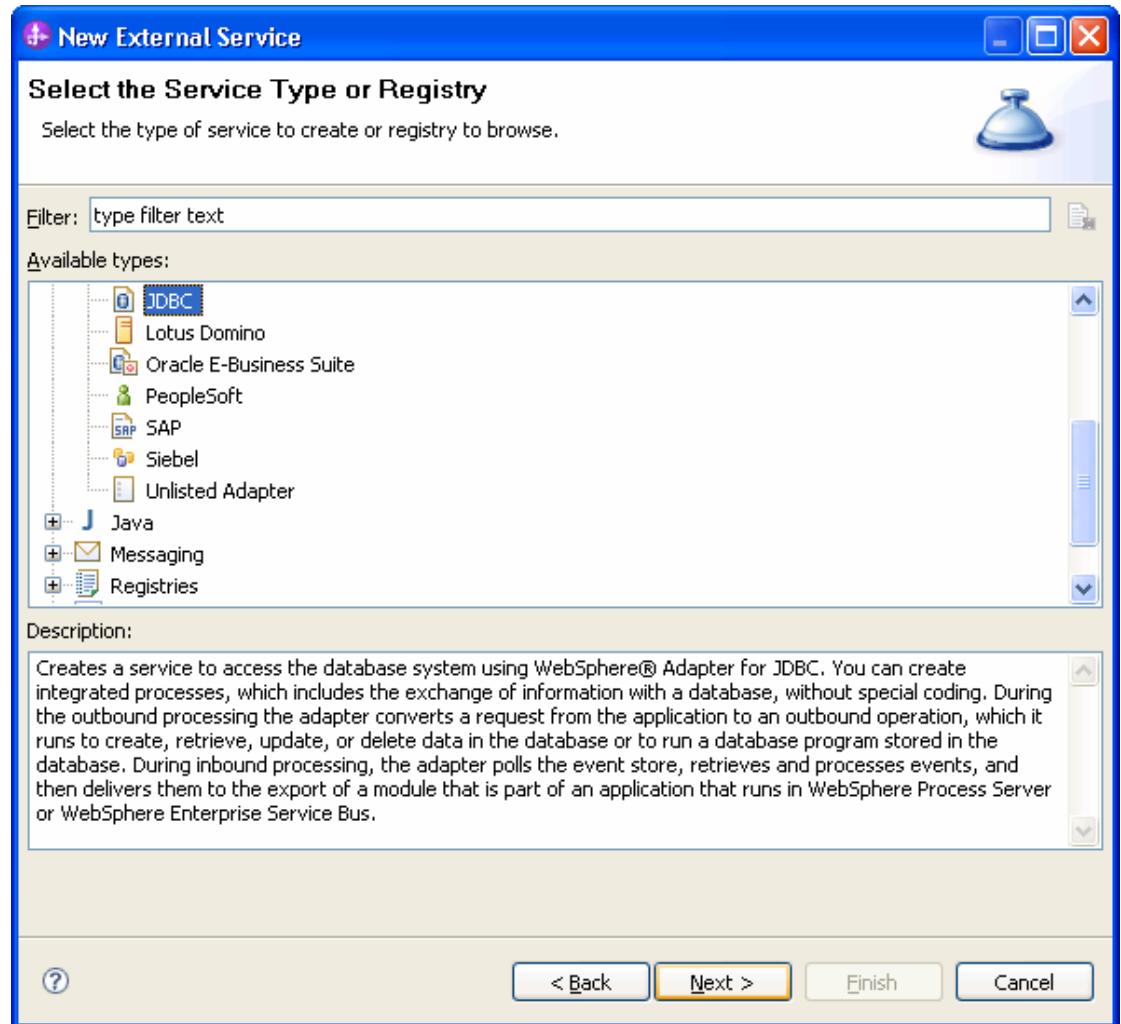
New	Delete	Test connection	Manage state...														
<table border="1"> <thead> <tr> <th>Select</th> <th>Name</th> <th>JNDI name</th> <th>Scope</th> <th>Provider</th> <th>Description</th> <th>Category</th> </tr> </thead> <tbody> <tr> <td><input type="checkbox"/></td> <td>Oracle JDBC Driver DataSource</td> <td>OracleDS</td> <td>Node=nlNode01</td> <td>Oracle JDBC Driver</td> <td>New JDBC Datasource</td> <td></td> </tr> </tbody> </table>				Select	Name	JNDI name	Scope	Provider	Description	Category	<input type="checkbox"/>	Oracle JDBC Driver DataSource	OracleDS	Node=nlNode01	Oracle JDBC Driver	New JDBC Datasource	
Select	Name	JNDI name	Scope	Provider	Description	Category											
<input type="checkbox"/>	Oracle JDBC Driver DataSource	OracleDS	Node=nlNode01	Oracle JDBC Driver	New JDBC Datasource												
Total 1																	

6. Close the Admin Console tab.

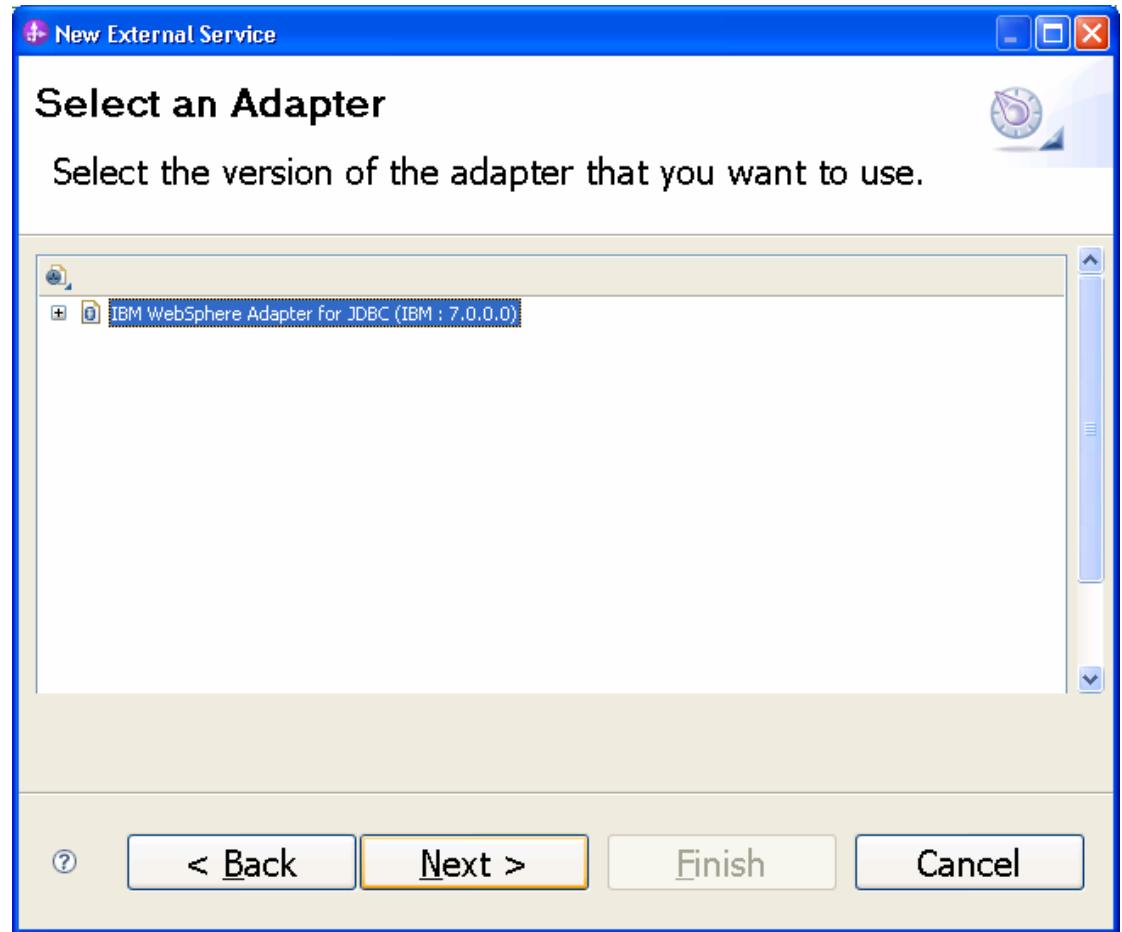
Configure the adapter for inbound processing

Run the external service wizard to specify business objects, services, and configuration details.

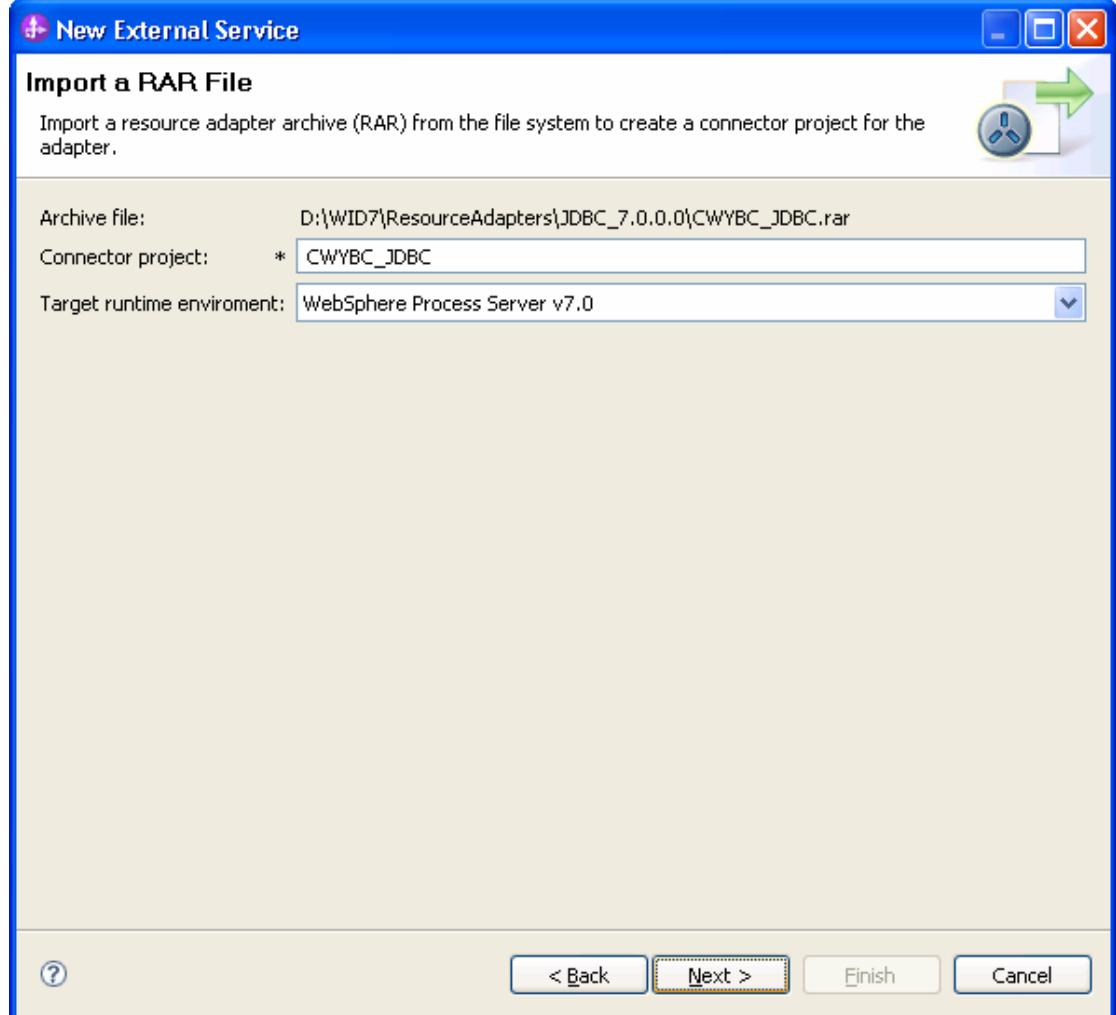
1. Switch to the Business Integration Perspective in WebSphere Integration Developer by selecting **Window -> Open Perspective Business Integration**.
2. Start the external service wizard by selecting **File-> New -> External Service**.
3. In the **Available Types** area, select **Adapters > JDBC** and click **Next**.



4. Select **IBM WebSphere Adapter for JDBC** and click **Next**.

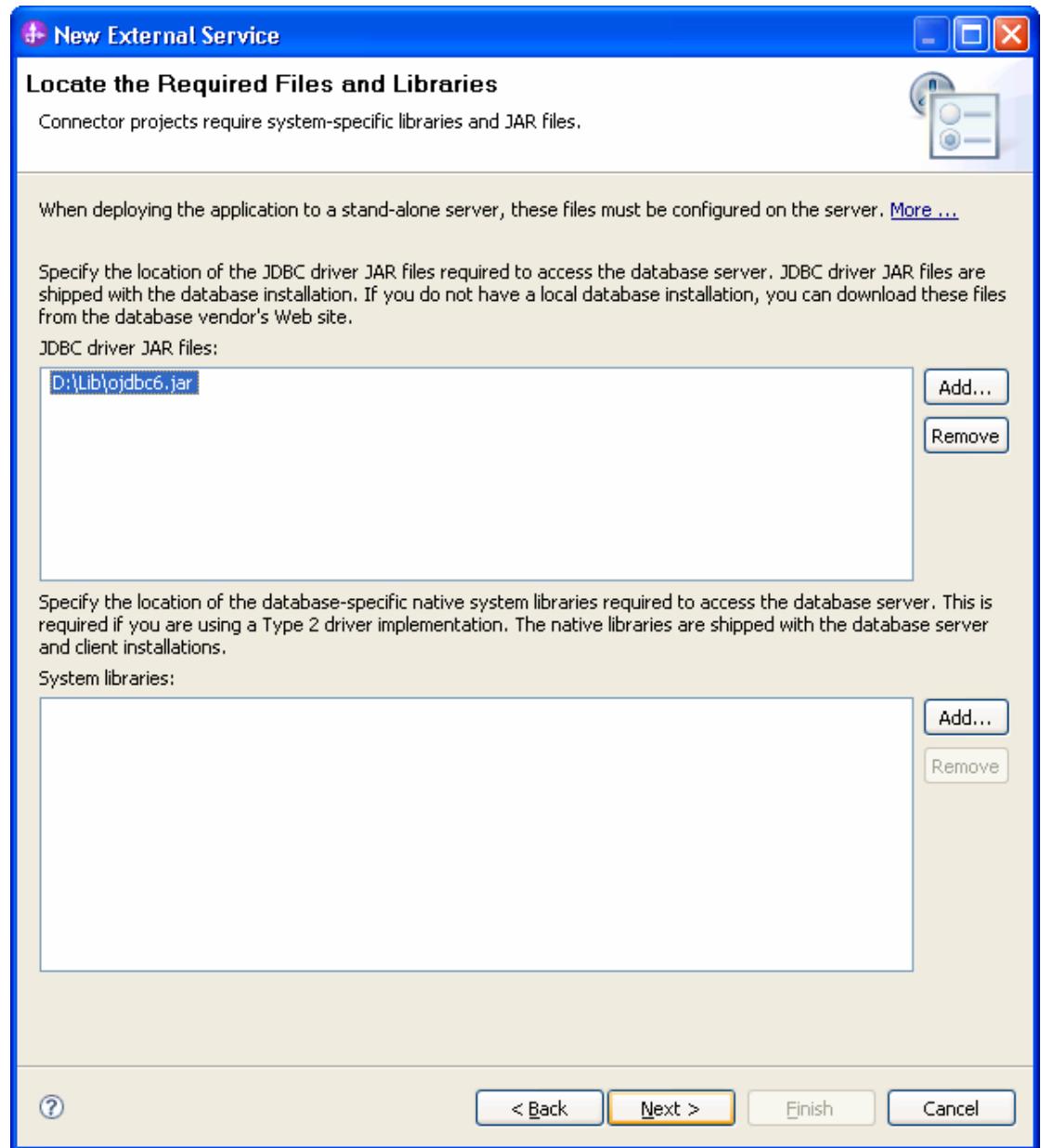


5. In the **Connector project** field, enter **CWYBC_JDBC**.
6. In the **Target runtime environment** field, select appropriate runtime and click **Next**.

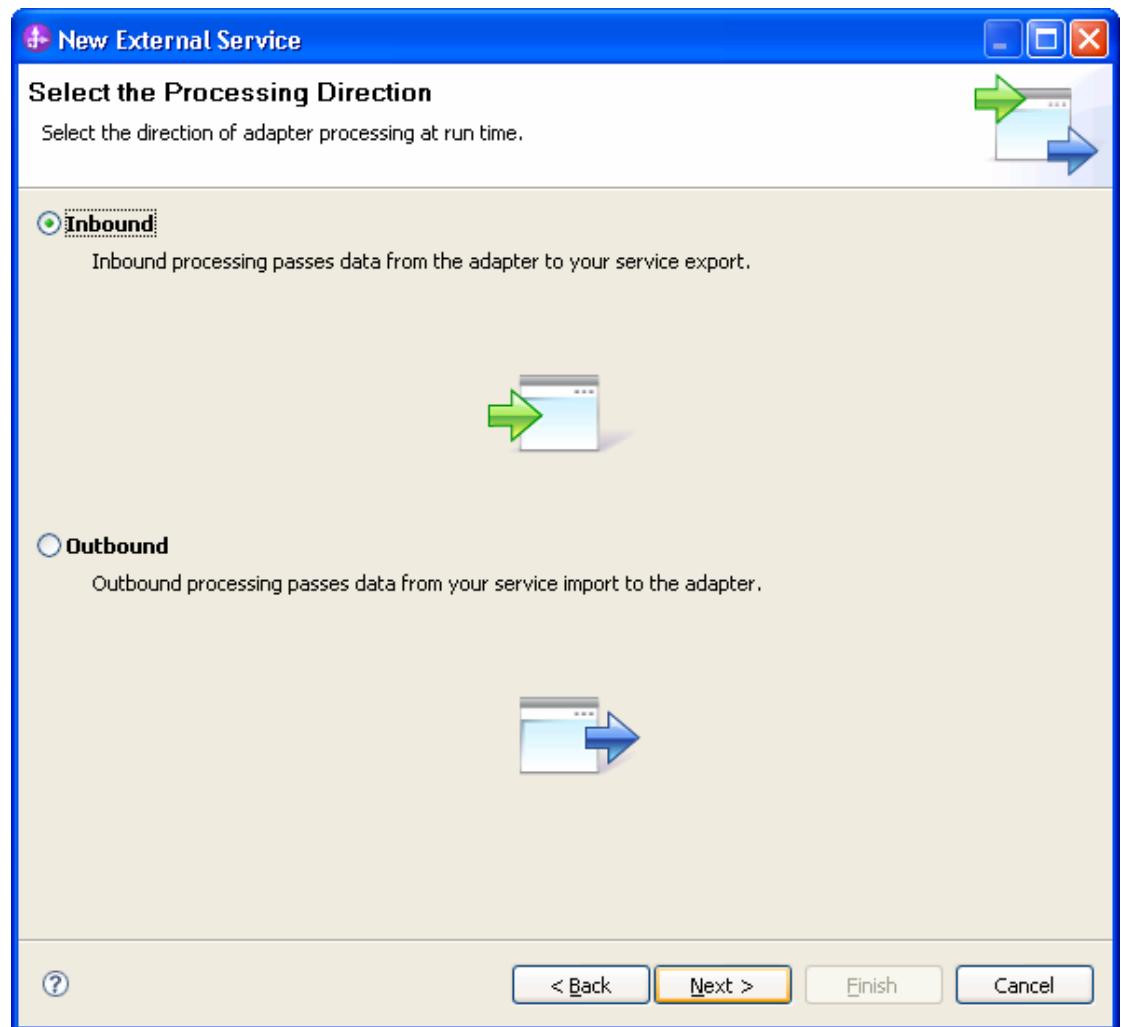


WebSphere software

7. In the **JDBC driver JAR files** field, click **Add** to add the JDBC driver class to connect to the database. Browse to select the driver JAR file and click **Next**.



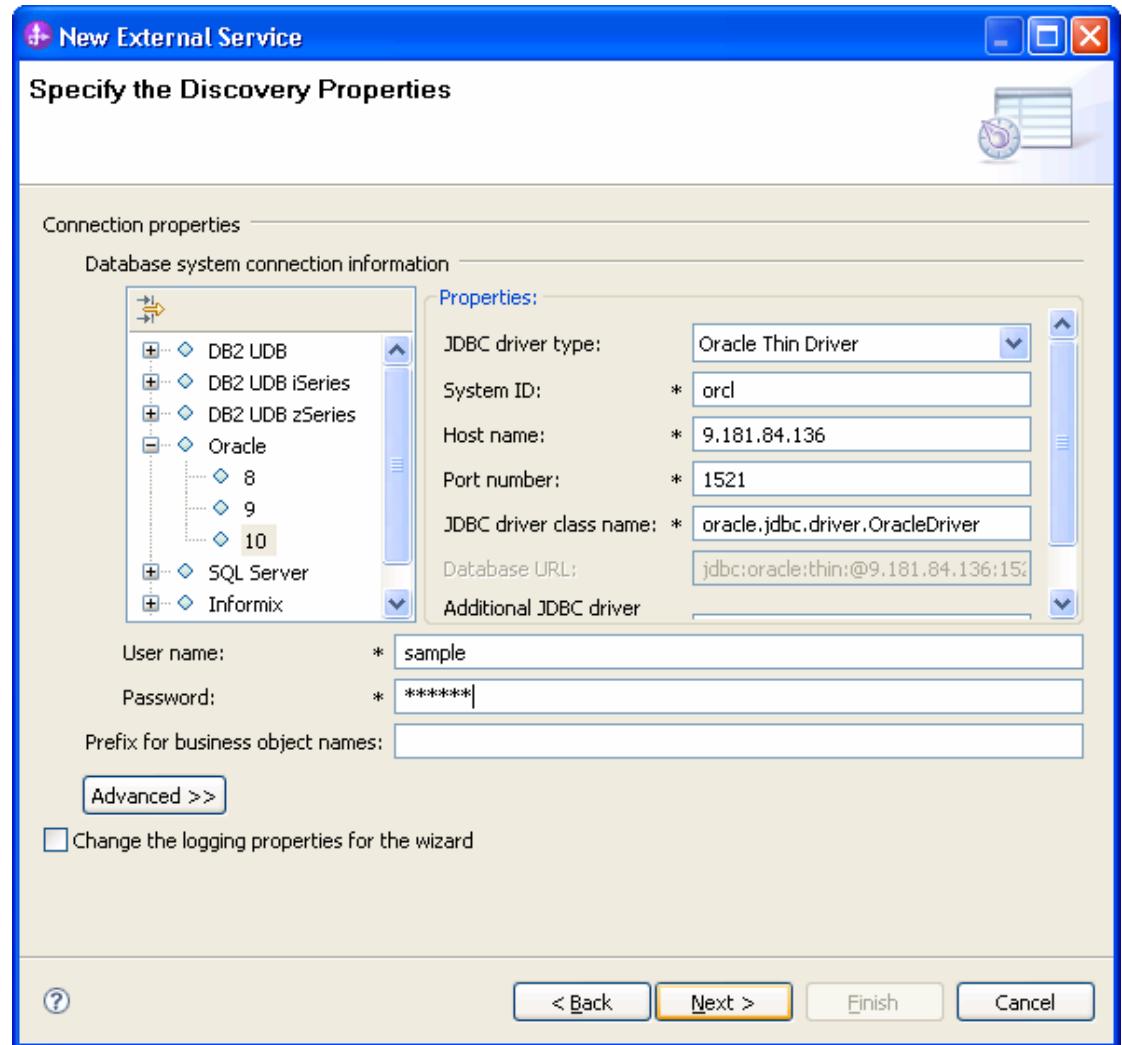
8. Select **Inbound** and click **Next**.



Set connection properties for the external service wizard

To connect to the Oracle database:

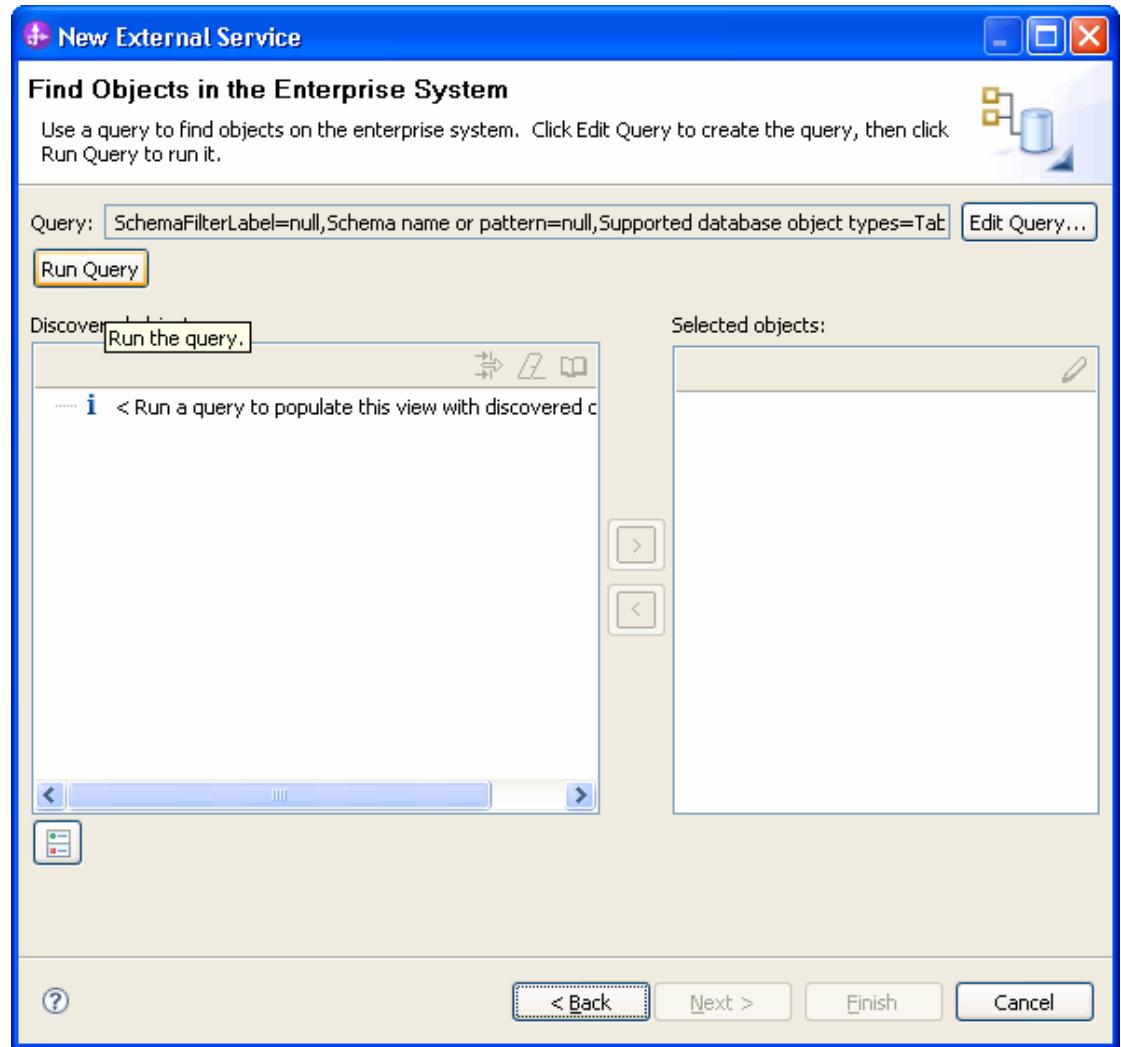
1. Expand the **Oracle** node in the **Database system connection information** area, and then select **10**.
2. Enter values in the **System ID**, **Host name**, **Port number**, **User name** and **Password** fields, and click **Next**.



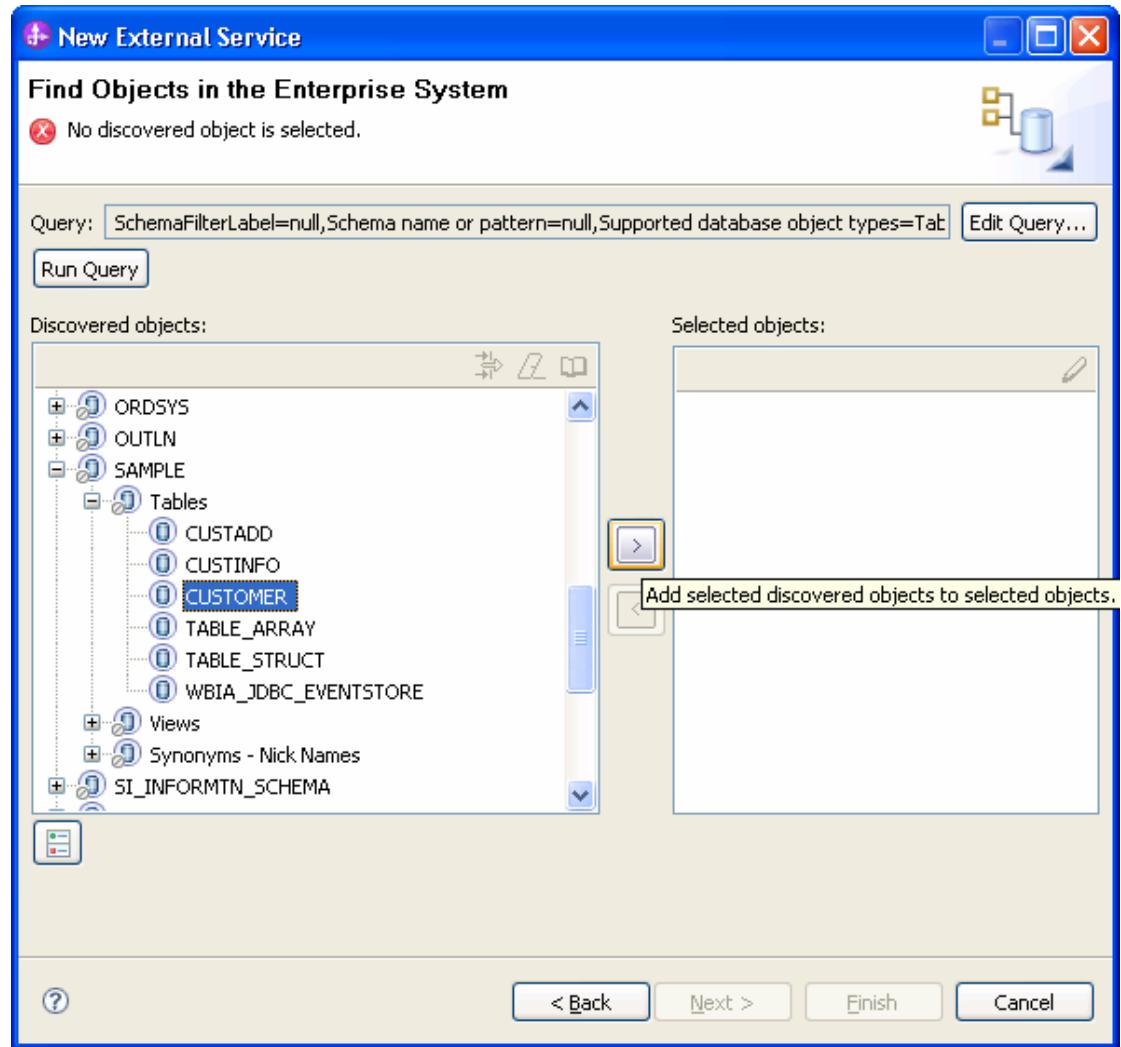
Select the business objects and services to be used with the adapter

Follow these steps to select the data for Inbound Processing:

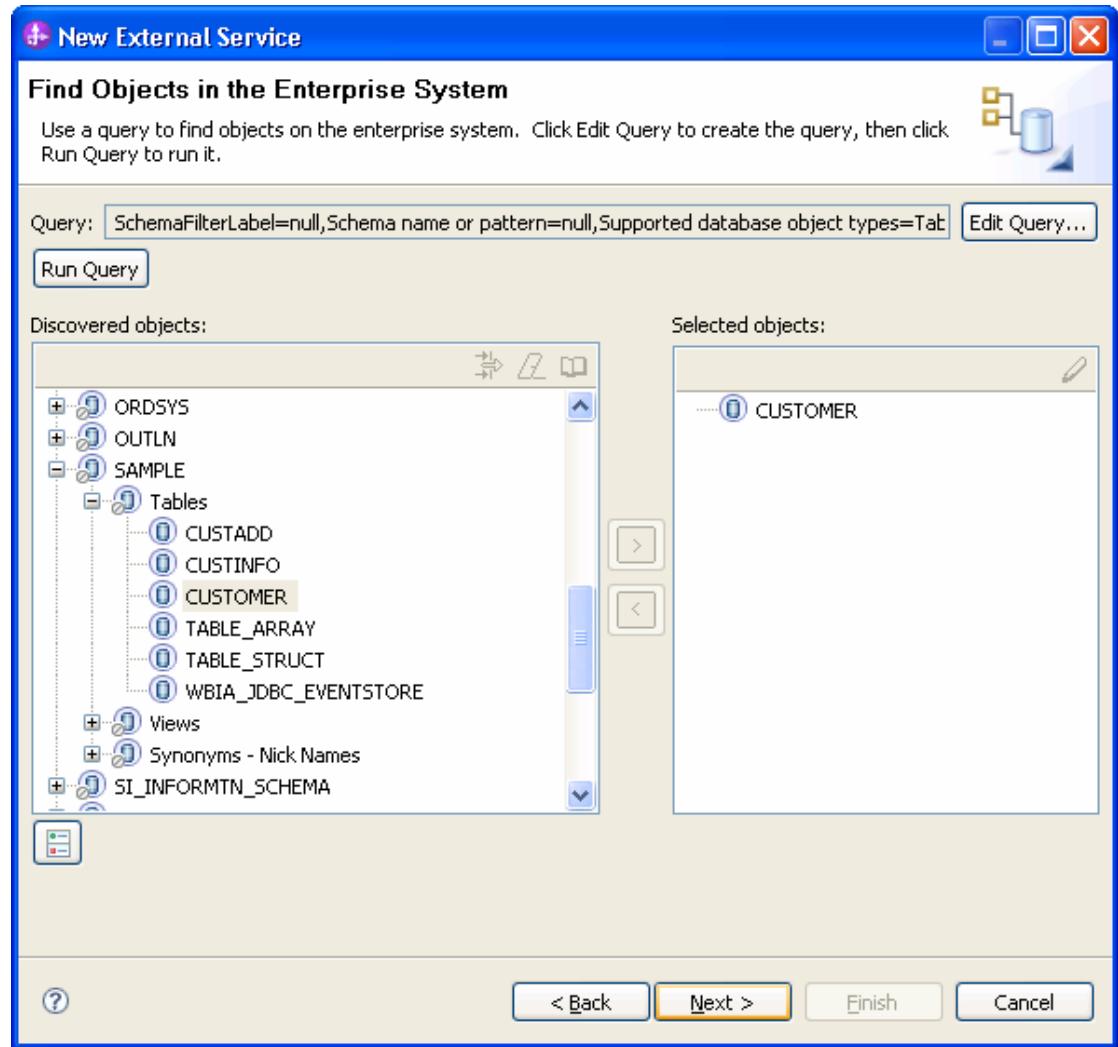
1. In the Find Objects in Enterprise System window, click **Run Query**.



2. Expand the **SAMPLE** (for this tutorial only) node, select **Tables** and expand it.



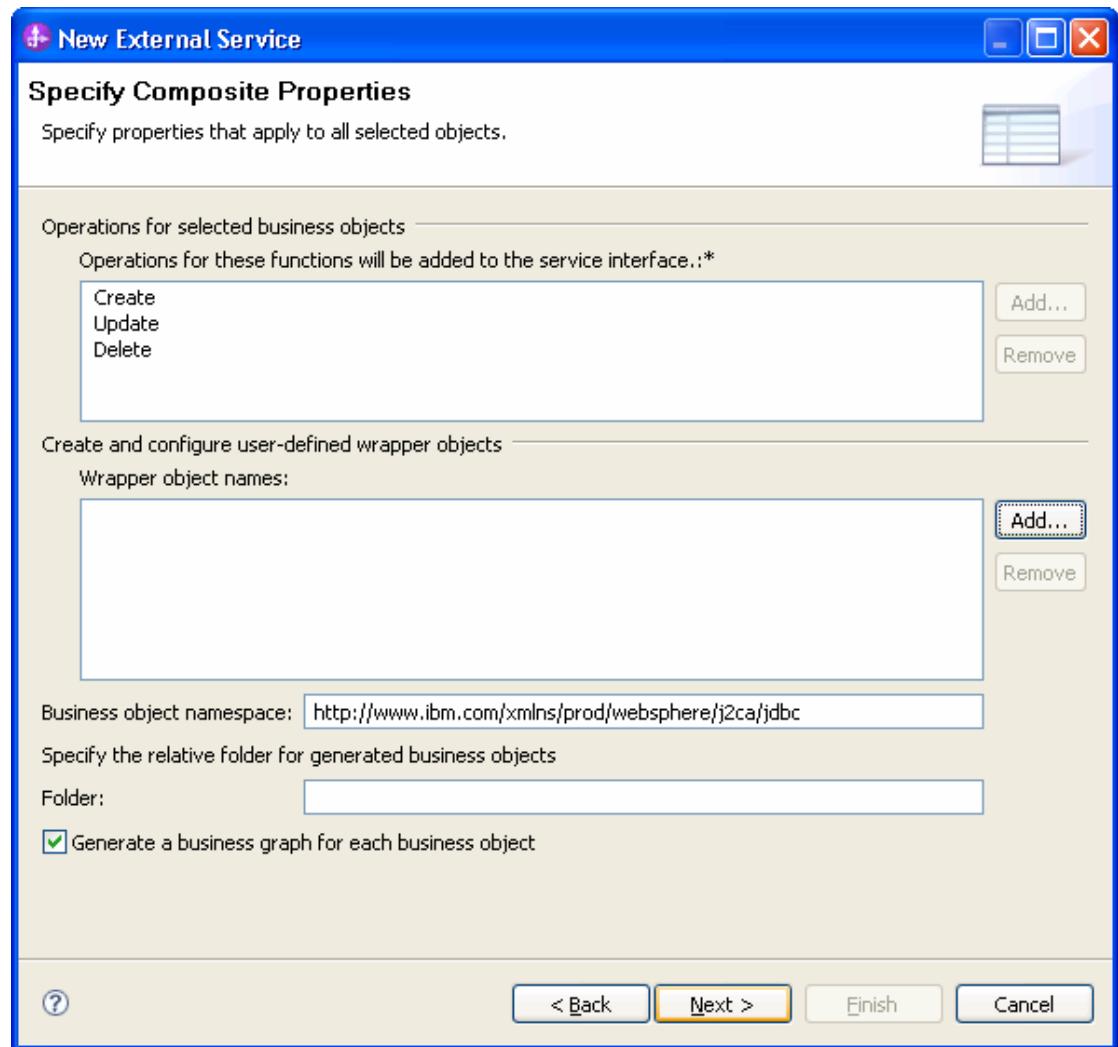
3. Select the **CUSTOMER** table and click
4. Click **Next**.



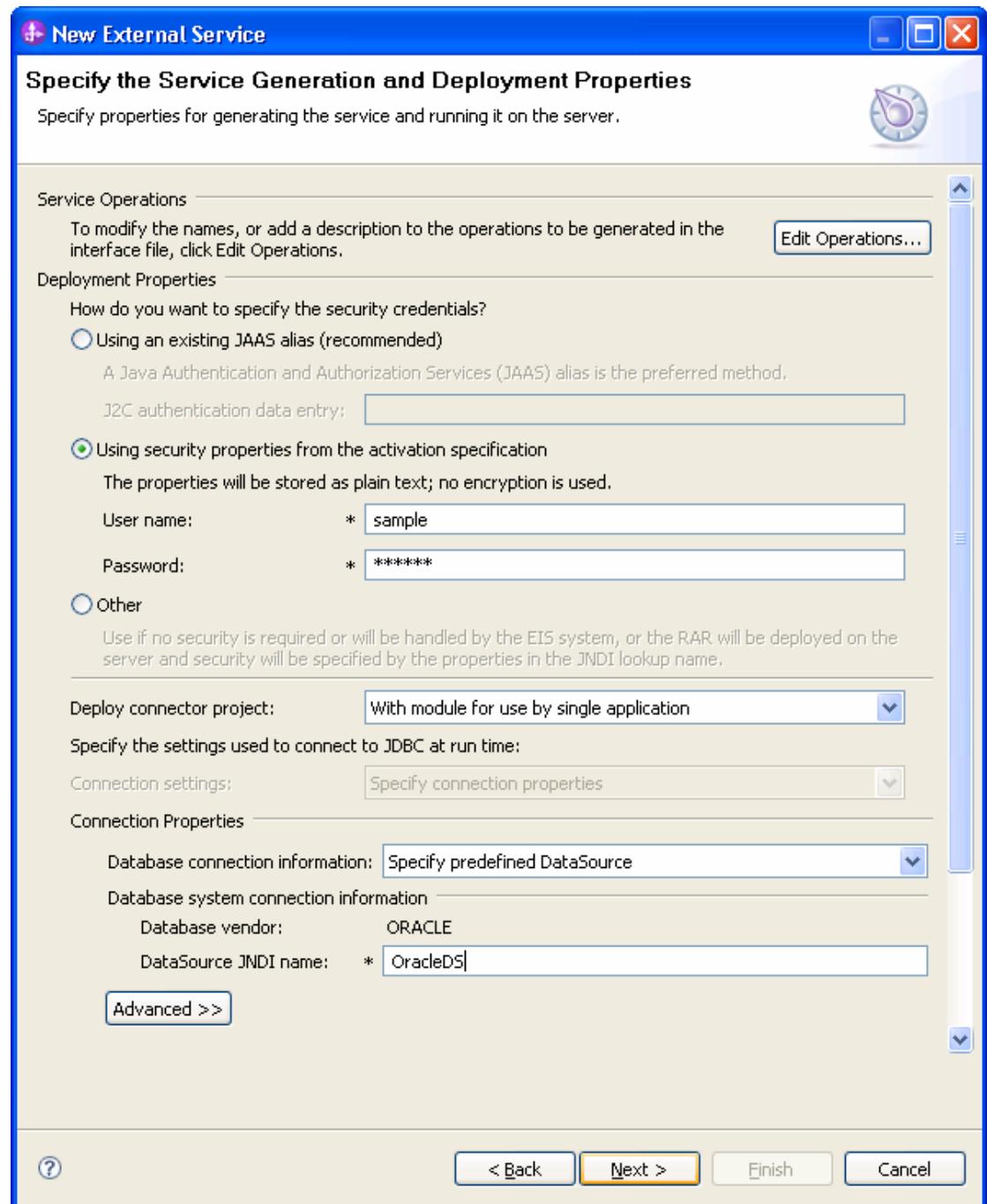
Generate business object definitions and related artifacts

Follow these steps to generate the business object definitions.

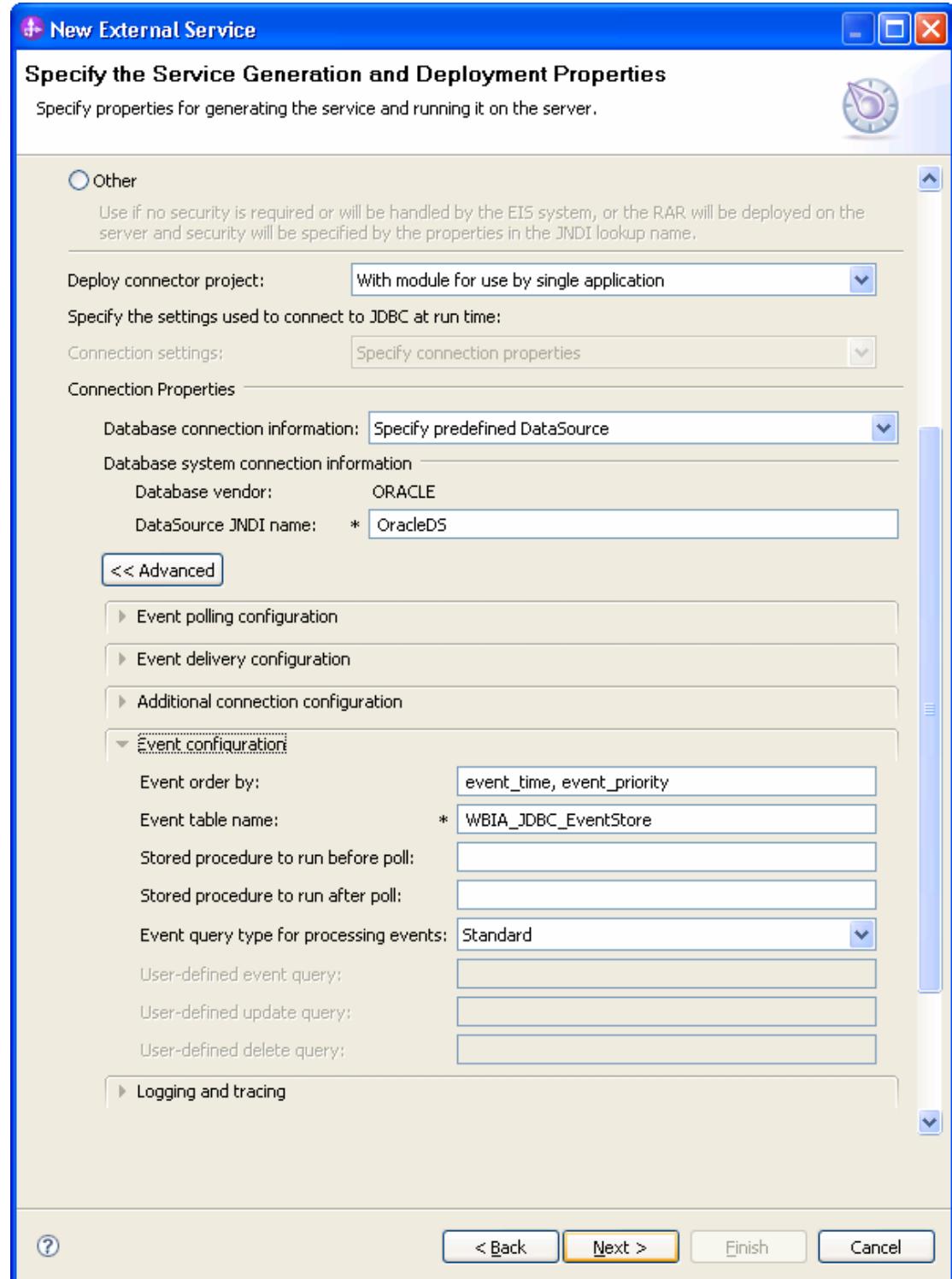
1. In the Specify Composite Properties window, accept the default values and click **Next**.



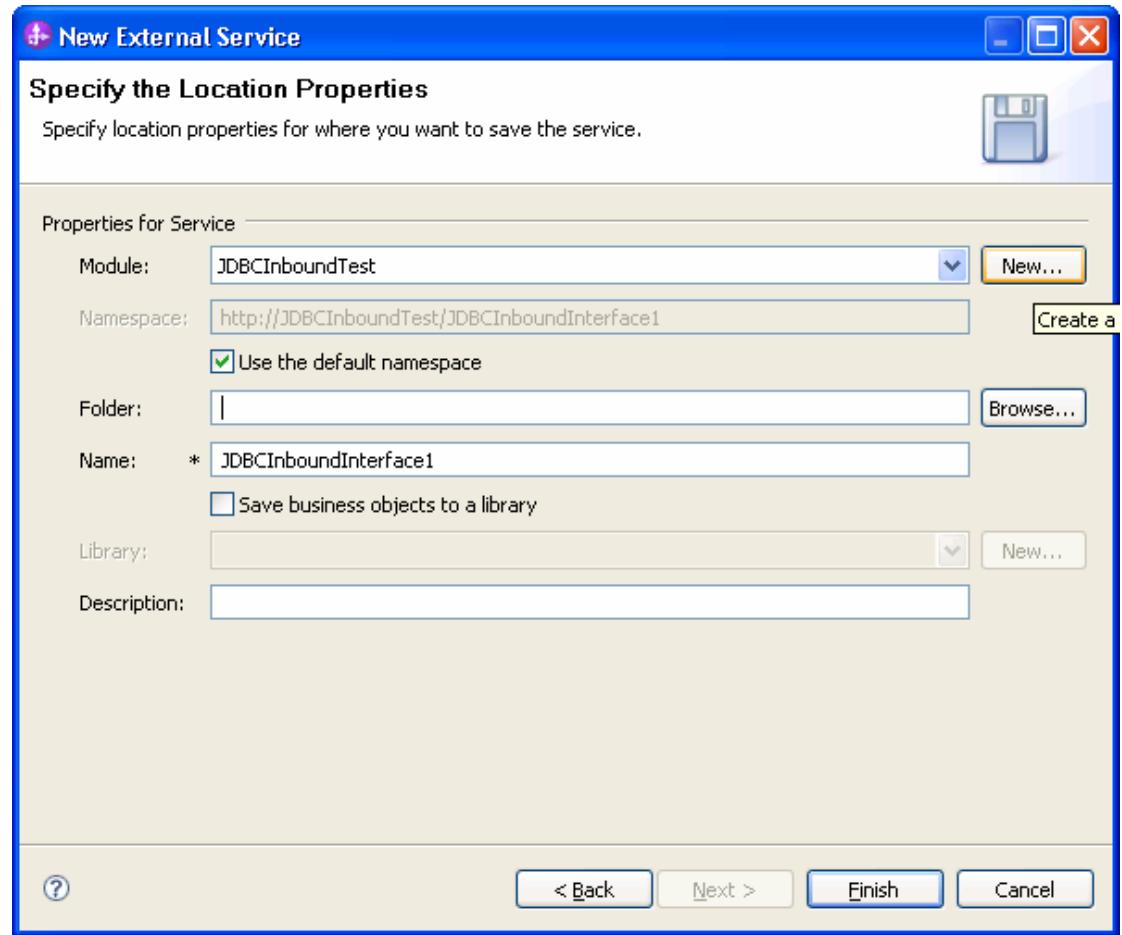
2. In the Specify the Service Generation and Deployment Properties window, perform the following steps:
 - a) Select **Using security properties from the managed connection factory**.
 - b) Select **Specify predefined DataSource** from the **Database connection information** list.
 - c) Click **Advanced**.



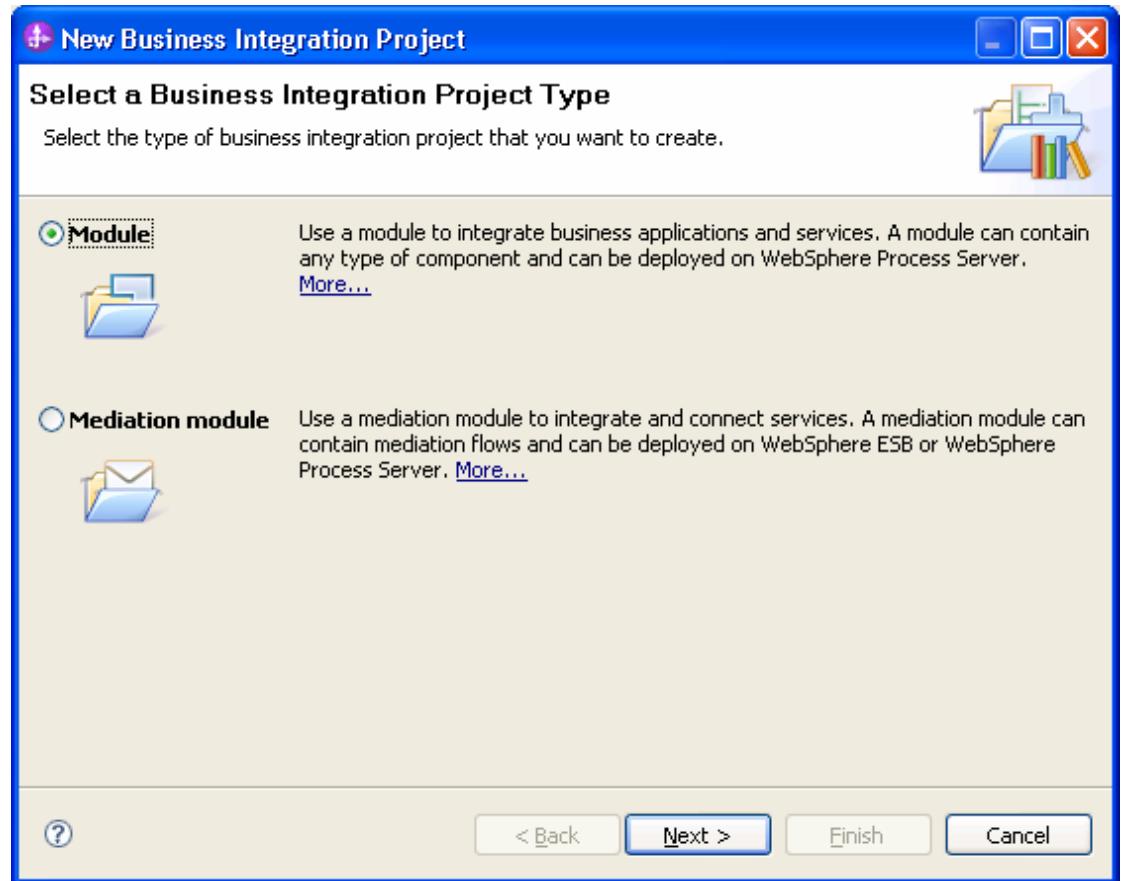
- d) In the Event Configuration area, enter the values for the **Event Order By**, **Event Table Name** fields and click **Next**.



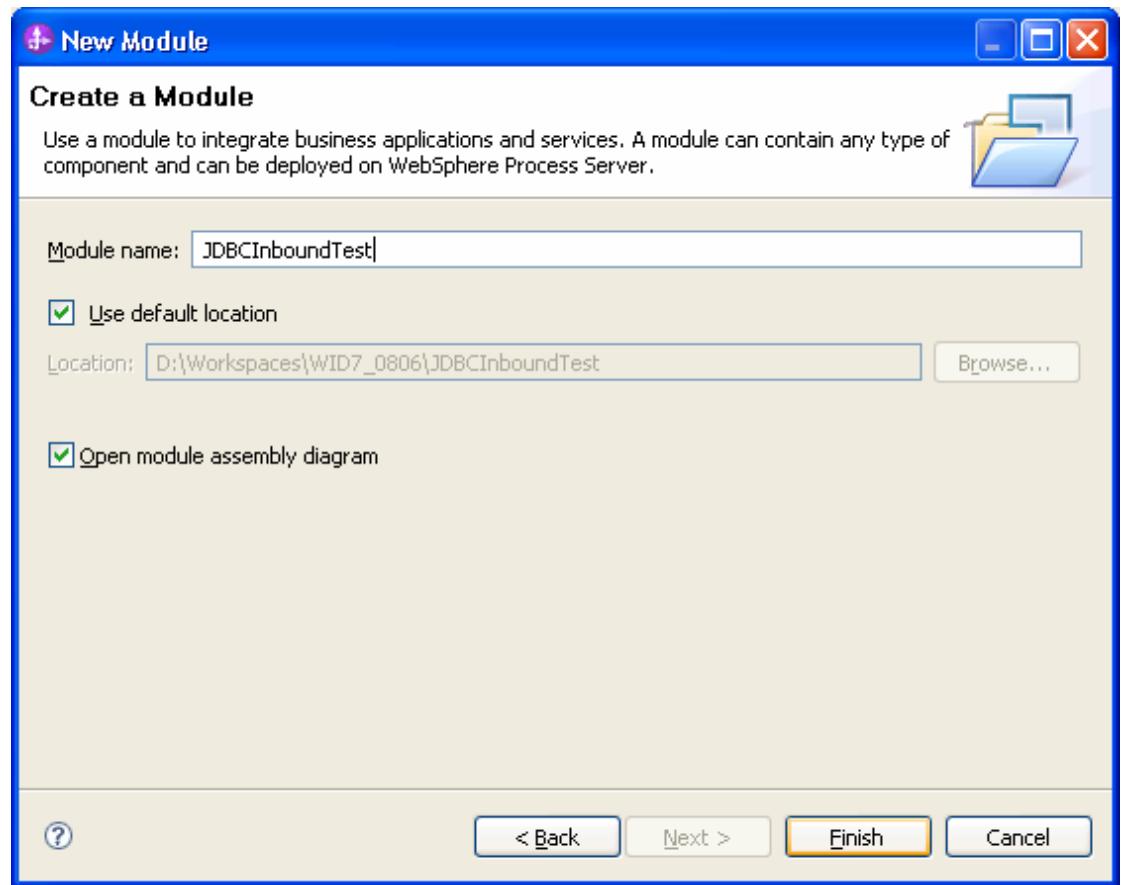
3. In the Specify the location Properties window, click **New**.



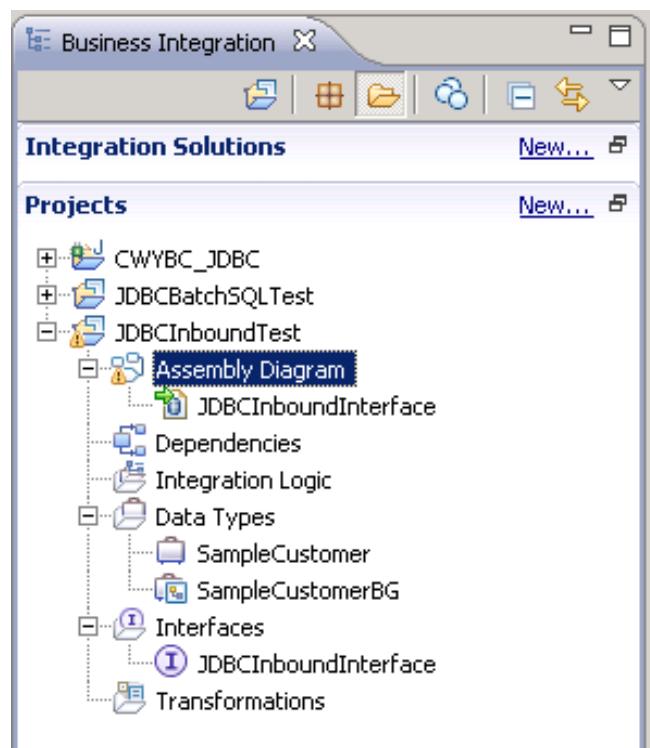
4. In the Select a Business Integration Project Type window, select **Module** and click **Next**.



5. In the Create a Module window, type **JDBCInboundTest** in the **Module Name** field and click **Finish**.



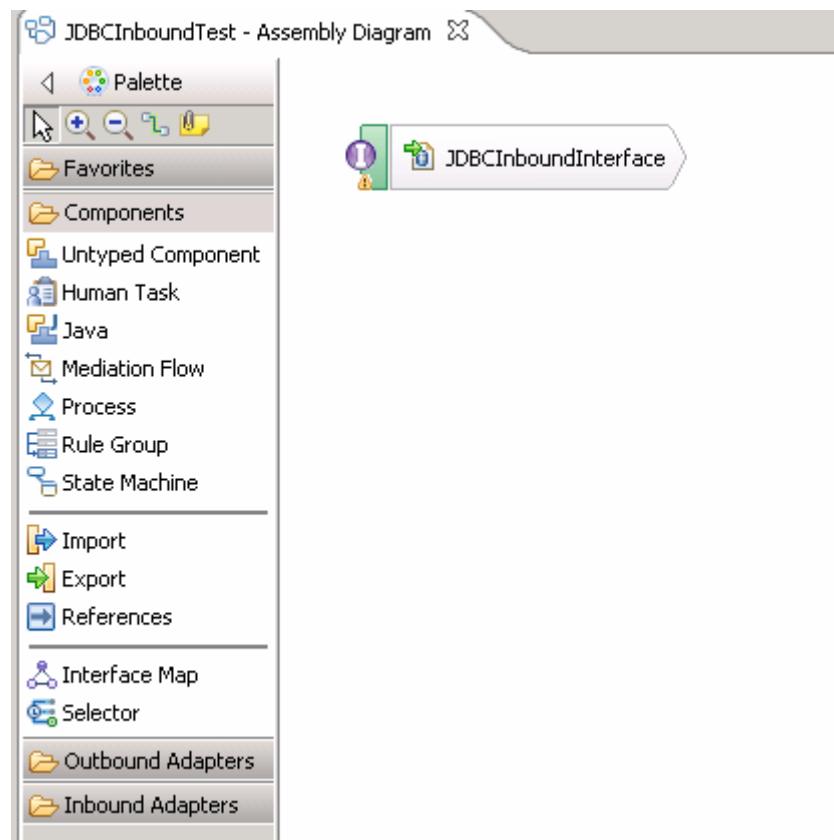
6. Click **Finish** to complete service creation.
7. Verify the results.



Set up the components to be part of the inbound environment

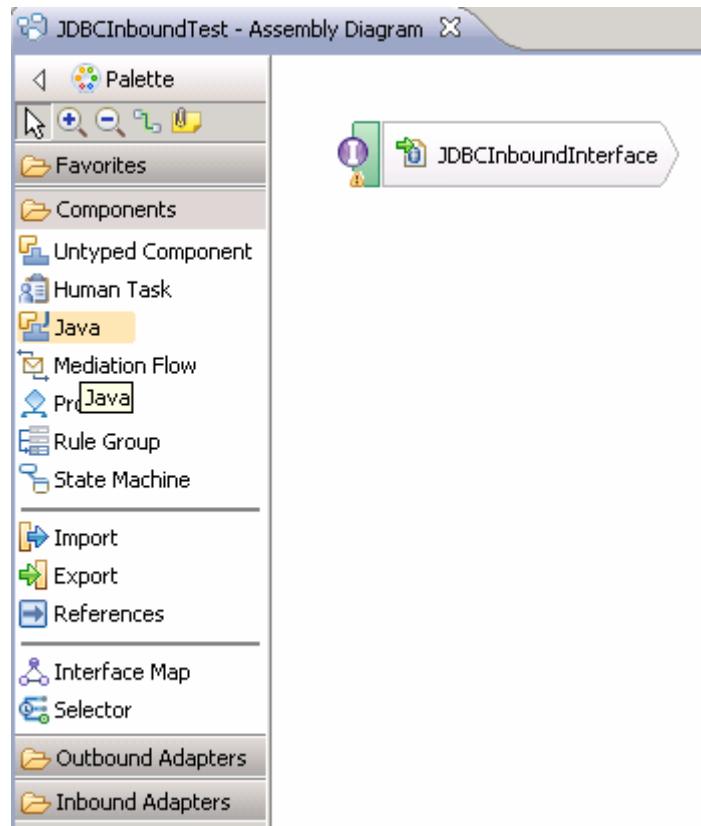
Add the components and set transaction specific properties for them so that they are part of the inbound environment.

1. Open the Assembly Diagram. It shows the **JDBCInboundInterface** that was generated when the artifacts were created.

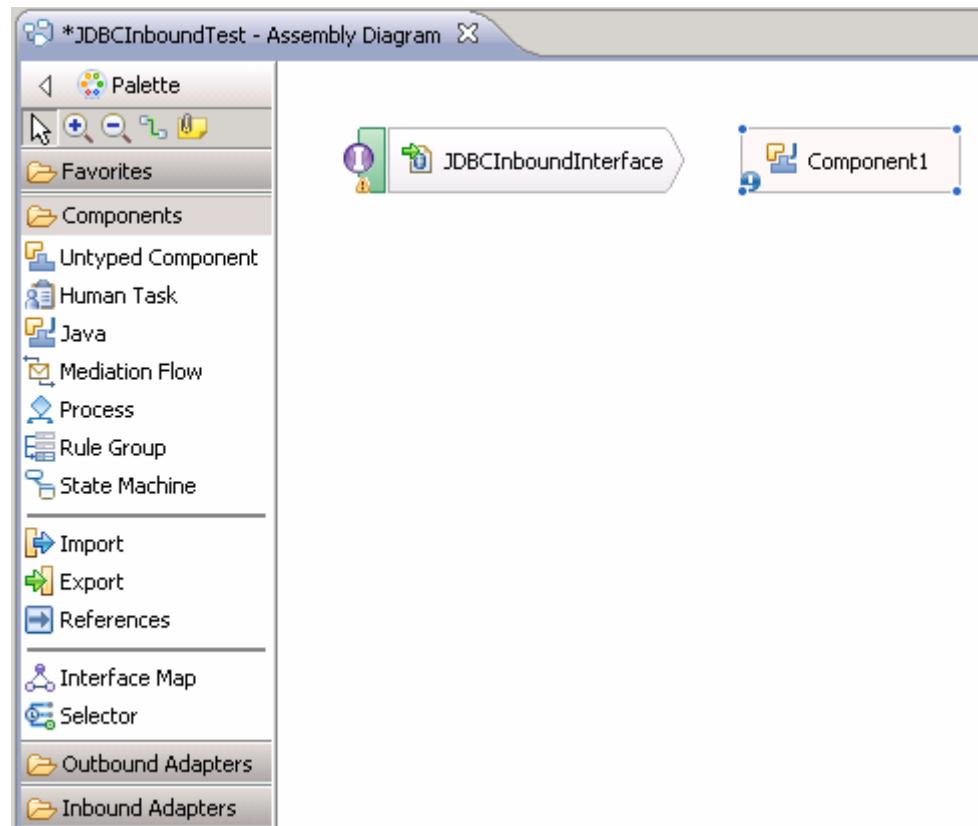


2. From the Palette, select the **Java** component and drop it on the assembly diagram.

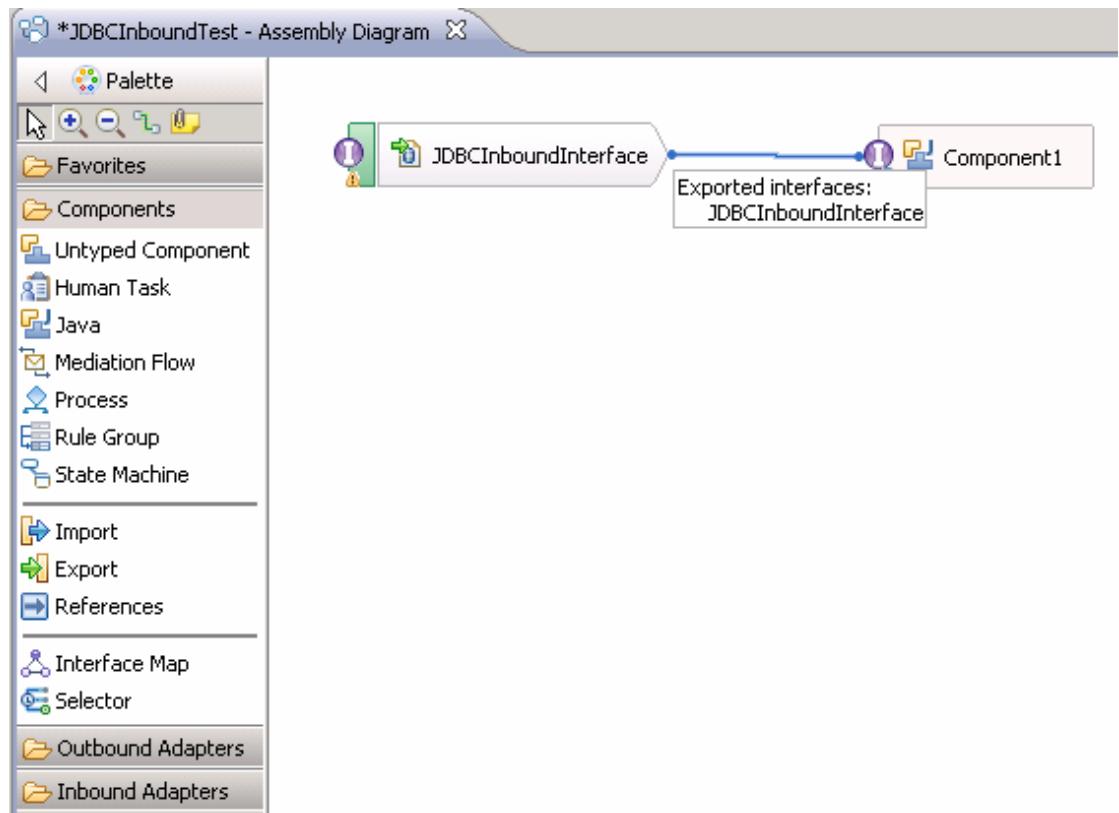
WebSphere software



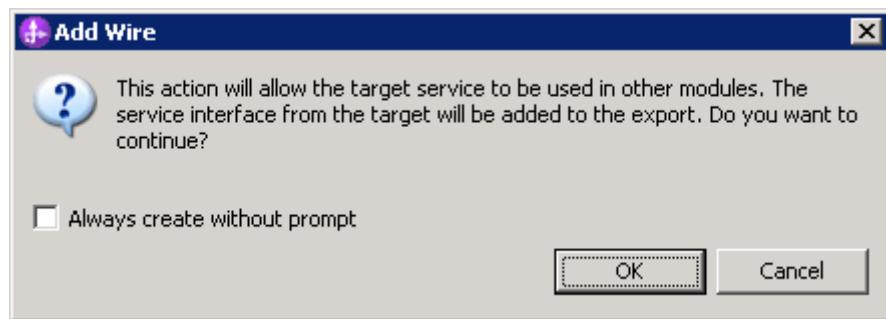
A component named **Component1** is created in the Assembly diagram.



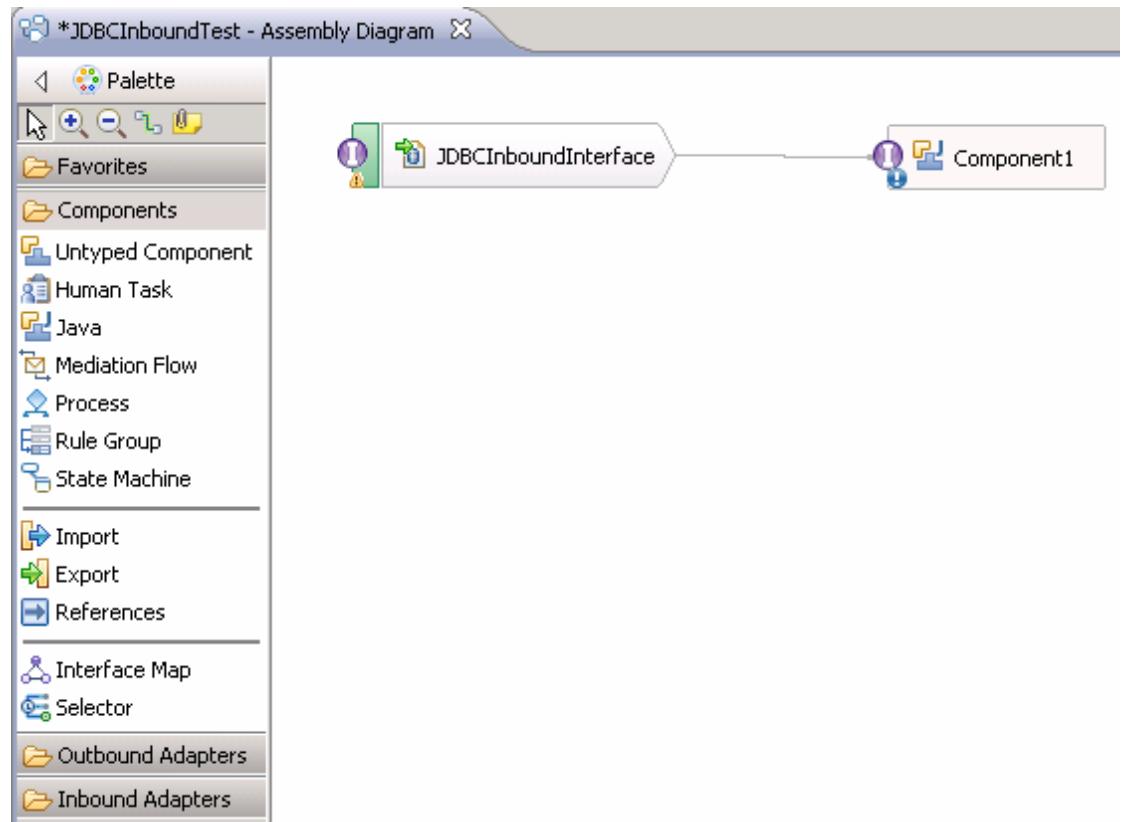
3. Wire **JDBCInboundInterface** to **Component1** by dragging the mouse pointer from the rear end of **JDBCInboundInterface** to the front end of **Component1**.



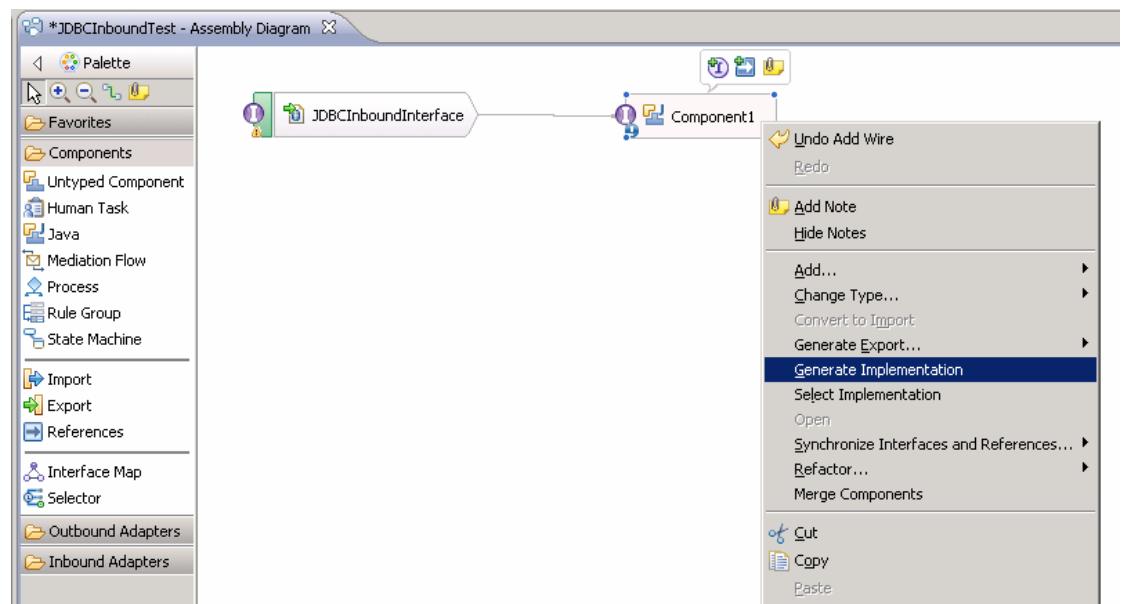
Note: Before the preceding window, i.e., before wiring you will see the following window. Click **OK**.



The Assembly diagram now looks like the figure below.



4. Generate the implementation for Java component. Right-click the component, and select **Generate Implementation** to complete service creation.



5. Highlight the default package and select **OK**.

The Java Editor displays the Component1Impl.java file.

```

import commonj.sdo.DataObject;

public class ComponentImpl {
    /**
     * Default constructor.
     */
    public ComponentImpl() {
        super();
    }

    /**
     * Return a reference to the component service instance for this implementation
     * class. This method should be used when passing this service to a partner reference
     * or if you want to invoke this component service asynchronously.
     *
     * @generated (com.ibm.wbit.java)
     */
    @SuppressWarnings("unused")
    private Object getMyService() {
        return (Object) ServiceManager.INSTANCE.locateService("self");
    }

    /**
     * Method generated to support implementation of operation "createSampleCustomerBG" defined for WSDL port type
     * named "JDBCInboundInterface".
     *
     * The presence of commonj.sdo.DataObject as the return type and/or as a parameter
     * type conveys that it is a complex type. Please refer to the WSDL Definition for more information
     * on the type of input, output and fault(s).
     */
    public void createSampleCustomerBG(DataObject createSampleCustomerBGInput) {
}

```

6. Scroll down and locate the createSampleCustomer(DataObject createSampleCustomerBGInput) method that needs to be implemented. Write the code into the method so the complete method looks as follows:

```

    /**
     * Method generated to support implementation of operation "createSampleCustomerBG" defined for
     * named "JDBCInboundInterface".
     *
     * The presence of commonj.sdo.DataObject as the return type and/or as a parameter
     * type conveys that it is a complex type. Please refer to the WSDL Definition for more information
     * on the type of input, output and fault(s).
     */
    public void createSampleCustomerBG(DataObject createSampleCustomerBGInput) {
        // To get or set attributes for DataObject createSampleCustomerBGInput, use the APIs as shown
        // To set a string attribute in createSampleCustomerBGInput, use createSampleCustomerBGInput
        // To get a string attribute in createSampleCustomerBGInput, use createSampleCustomerBGInput
        // To set a dataObject attribute in createSampleCustomerBGInput, use createSampleCustomerBGInput
        // To get a dataObject attribute in createSampleCustomerBGInput, use createSampleCustomerBGInput
        System.out.println("Create customer");
        DataObject bg = createSampleCustomerBGInput;
        DataObject bo = bg.getDataObject("SampleCustomer");
        System.out.println("CUSTOMER KEY is: " + bo.getString("pkey"));
        System.out.println("CUSTOMER LAST NAME is: " + bo.getString("lname"));
        System.out.println("CUSTOMER FIRST NAME is: " + bo.getString("fname"));
        System.out.println("CUSTOMER CODE is: " + bo.getString("ccode"));
        System.out.println("CREATE end");
    }

```

7. Scroll down and locate the updateSampleCustomer(DataObject updateSampleCustomerBGInput) method that needs to be implemented. Write the code into the method so the complete method looks as follows:

```


    /**
     * Method generated to support implementation of operation "updateSampleCustomerBG" defined for
     * named "JDBCInboundInterface".
     *
     * The presence of commonj.sdo.DataObject as the return type and/or as a parameter
     * type conveys that it is a complex type. Please refer to the WSDL Definition for more information
     * on the type of input, output and fault(s).
     */
    public void updateSampleCustomerBG(DataObject updateSampleCustomerBGInput) {
        // To get or set attributes for DataObject updateSampleCustomerBGInput, use the APIs as shown
        // To set a string attribute in updateSampleCustomerBGInput, use updateSampleCustomerBGInput
        // To get a string attribute in updateSampleCustomerBGInput, use updateSampleCustomerBGInput
        // To set a dataObject attribute in updateSampleCustomerBGInput, use updateSampleCustomerBGInput
        // To get a dataObject attribute in updateSampleCustomerBGInput, use updateSampleCustomerBGInput
        System.out.println("UPDATE customer");
        DataObject bg = updateSampleCustomerBGInput;
        DataObject bo = bg.getDataObject("SampleCustomer");
        System.out.println("CUSTOMER KEY is: " + bo.getString("pkey"));
        System.out.println("CUSTOMER LAST NAME is: " + bo.getString("lname"));
        System.out.println("CUSTOMER FIRST NAME is: " + bo.getString("fname"));
        System.out.println("CUSTOMER CODE is: " + bo.getString("ccode"));
        System.out.println("UPDATE end");
    }
}


```

8. Scroll down and locate the deleteSampleCustomer(DataObject deleteSampleCustomerBGInput) method that needs to be implemented. Write the code into the method so the complete method looks as follows:

```


    /**
     * Method generated to support implementation of operation "deleteSampleCustomerBG" defined for
     * named "JDBCInboundInterface".
     *
     * The presence of commonj.sdo.DataObject as the return type and/or as a parameter
     * type conveys that it is a complex type. Please refer to the WSDL Definition for more information
     * on the type of input, output and fault(s).
     */
    public void deleteSampleCustomerBG(DataObject deleteSampleCustomerBGInput) {
        // To get or set attributes for DataObject deleteSampleCustomerBGInput, use the APIs as shown
        // To set a string attribute in deleteSampleCustomerBGInput, use deleteSampleCustomerBGInput
        // To get a string attribute in deleteSampleCustomerBGInput, use deleteSampleCustomerBGInput
        // To set a dataObject attribute in deleteSampleCustomerBGInput, use deleteSampleCustomerBGInput
        // To get a dataObject attribute in deleteSampleCustomerBGInput, use deleteSampleCustomerBGInput
        System.out.println("DELETE customer");
        DataObject bg = deleteSampleCustomerBGInput;
        DataObject bo = bg.getDataObject("SampleCustomer");
        System.out.println("CUSTOMER KEY is: " + bo.getString("pkey"));
        System.out.println("CUSTOMER LAST NAME is: " + bo.getString("lname"));
        System.out.println("CUSTOMER FIRST NAME is: " + bo.getString("fname"));
        System.out.println("CUSTOMER CODE is: " + bo.getString("ccode"));
        System.out.println("DELETE end");
    }
}


```

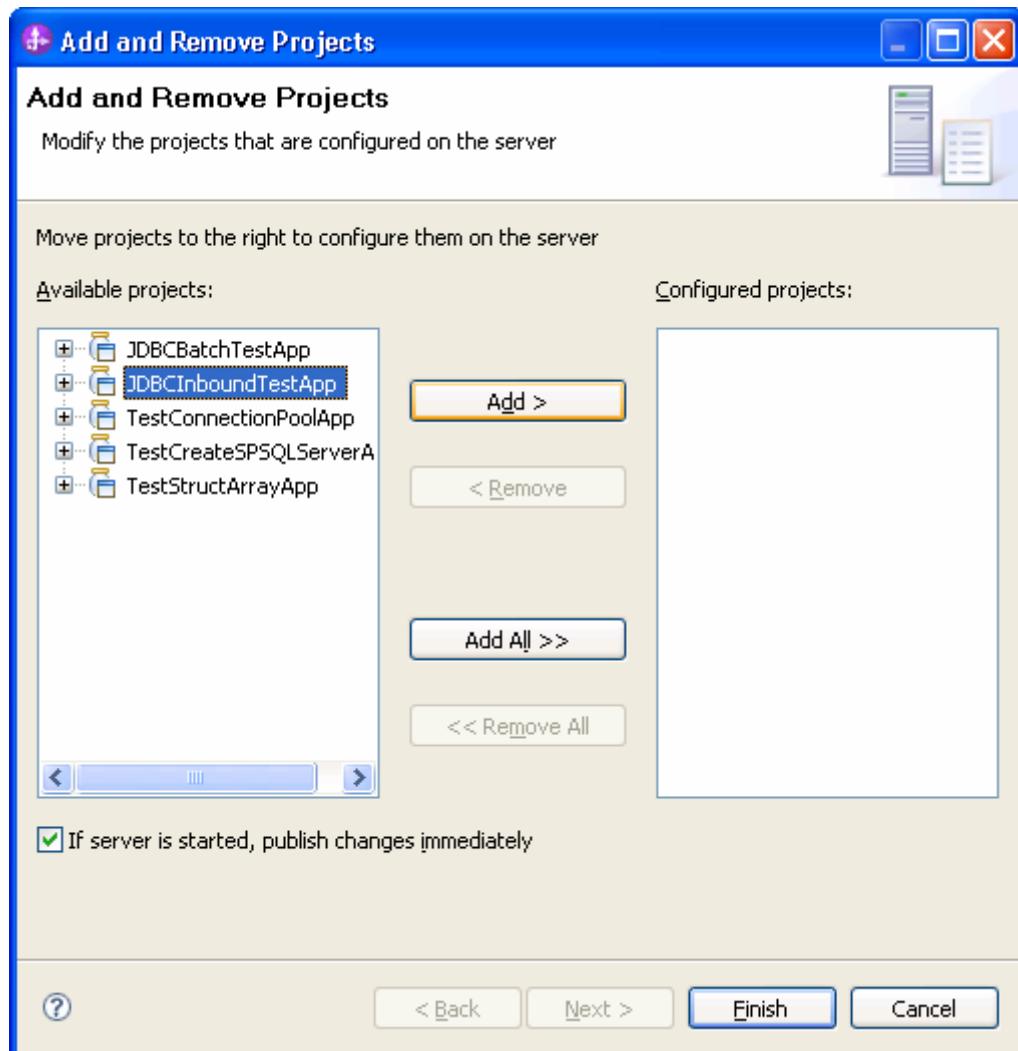
9. Select **File -> Save** to save your changes.
10. Close and save the Assembly Diagram. Wait for the workspace to complete building.

Deploy the module to the test environment

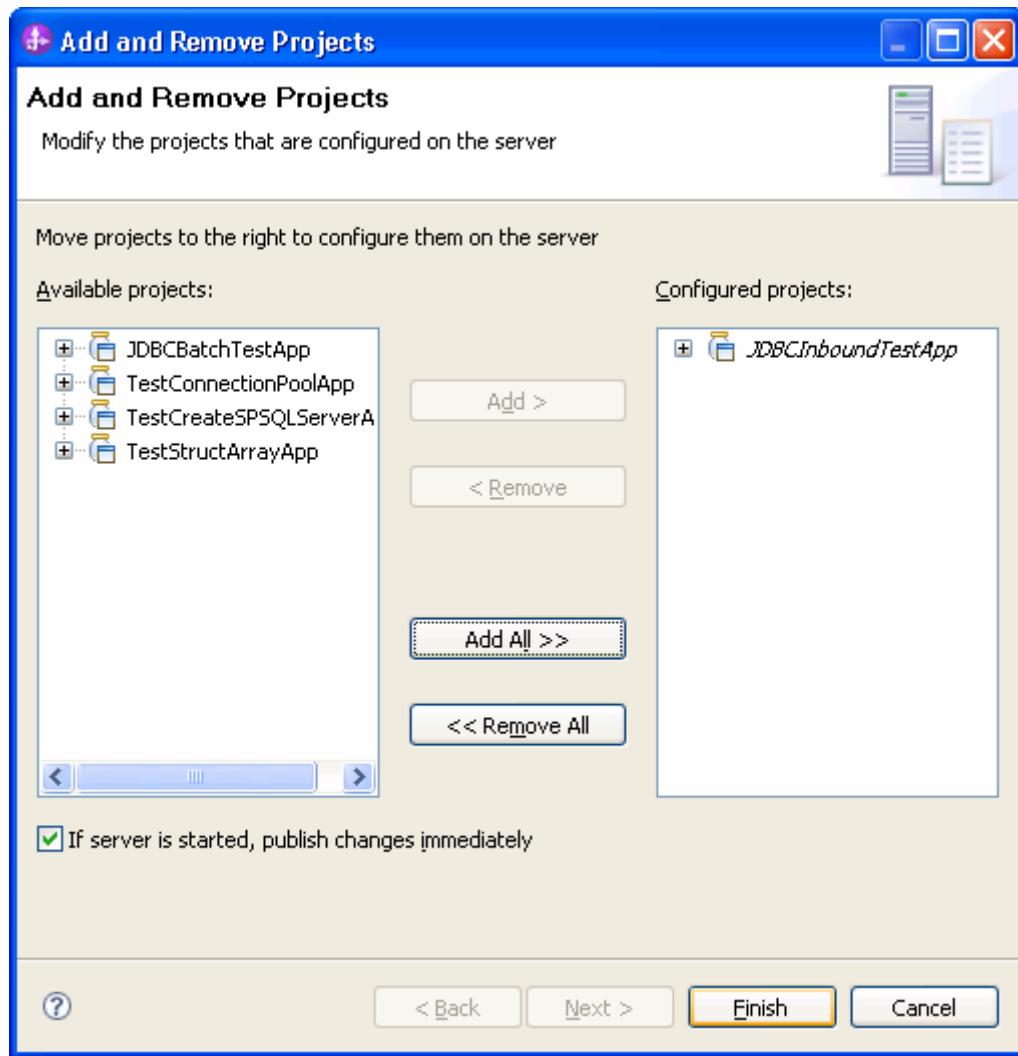
After running the external service wizard, you will have an SCA module that contains an Enterprise Information System (EIS) export. You must install this SCA module in the WebSphere Integration Developer integration test client. To do this, you must add the SCA module you created earlier to the server using the **Servers** view in WebSphere Integration Developer.

Steps for adding the SCA module to the server:

1. In WebSphere Integration Developer, switch to the **Servers** view by selecting from the toolbar **Window > Show View > Servers**.
2. In the **Servers** tab in the lower-right pane right click the server, and select **Start**.
3. After the server is started, right-click the server, and select **Add and Remove projects**.



4. Add the SCA module to the server.

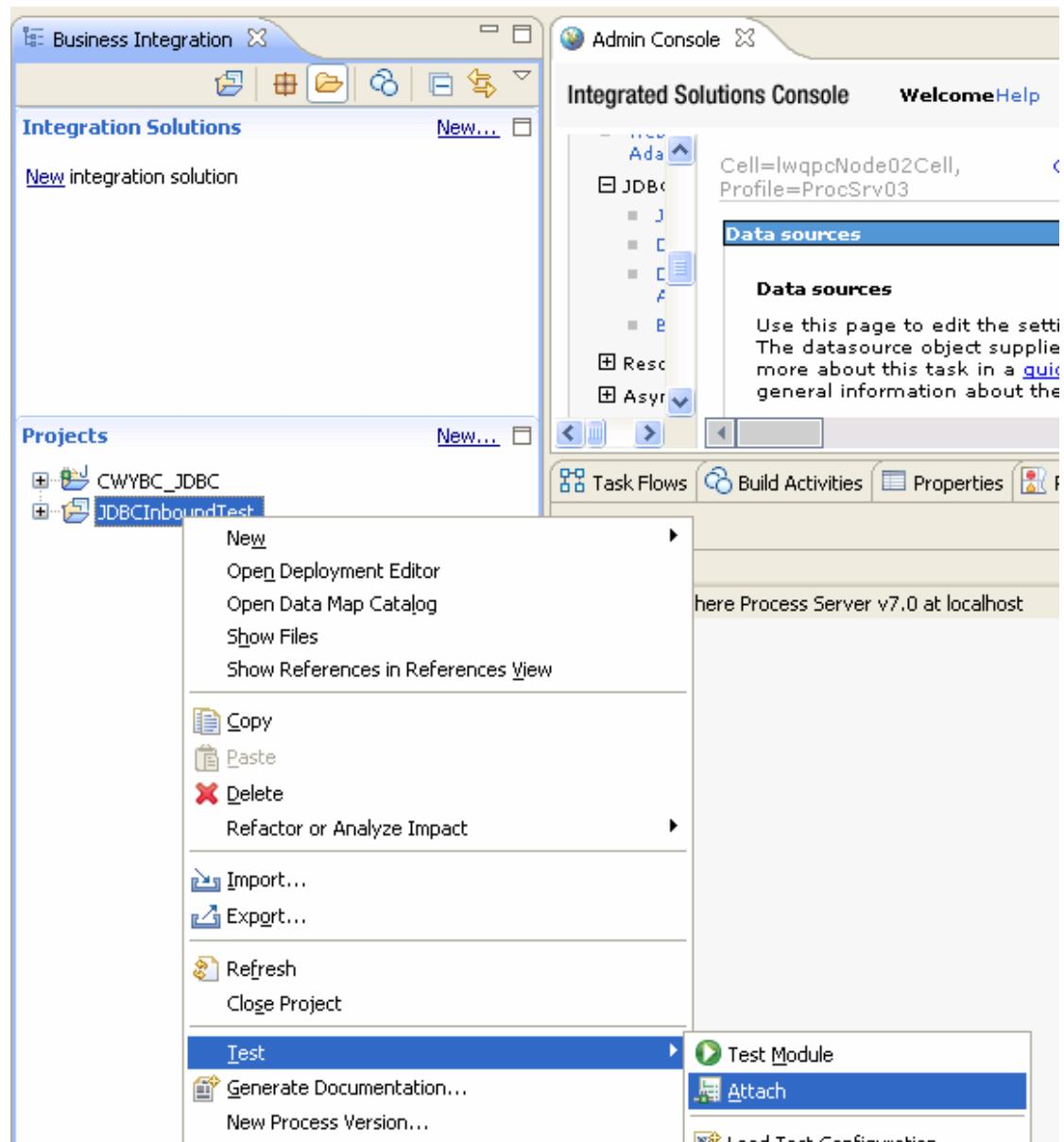


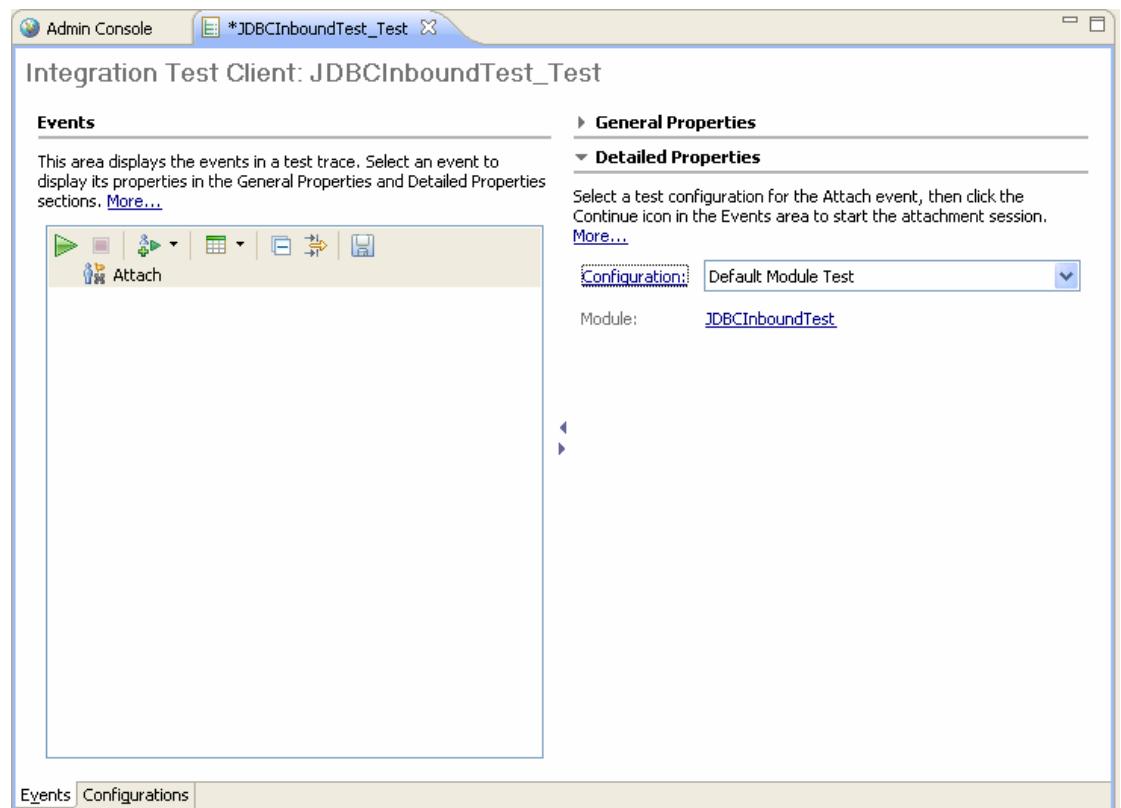
5. Click **Finish**.

Test the assembled adapter application

Test the assembled adapter application using the WebSphere Integration Developer integration test client.

1. In the Business Integration view right-click on the JDBCInboundTest module, and select Test > Attach.

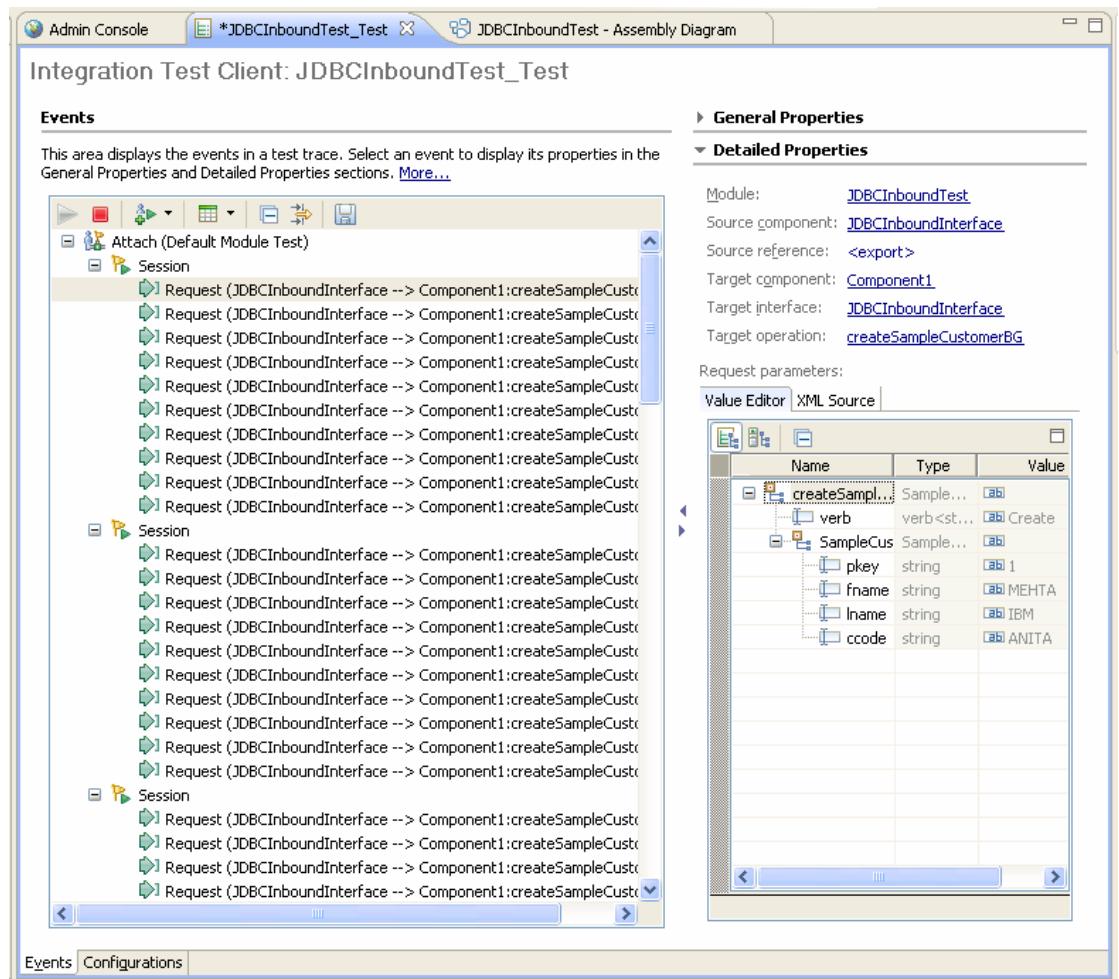




2. To execute the service, click .
3. Execute the **INSERTCUSTRECORDS** stored procedure to insert records into the Customer table:

```
BEGIN  
    INSERTCUSTRECORDS();  
END;
```

4. Check the output of the service:



Clear the sample content

Nothing is required to clean up after this tutorial.

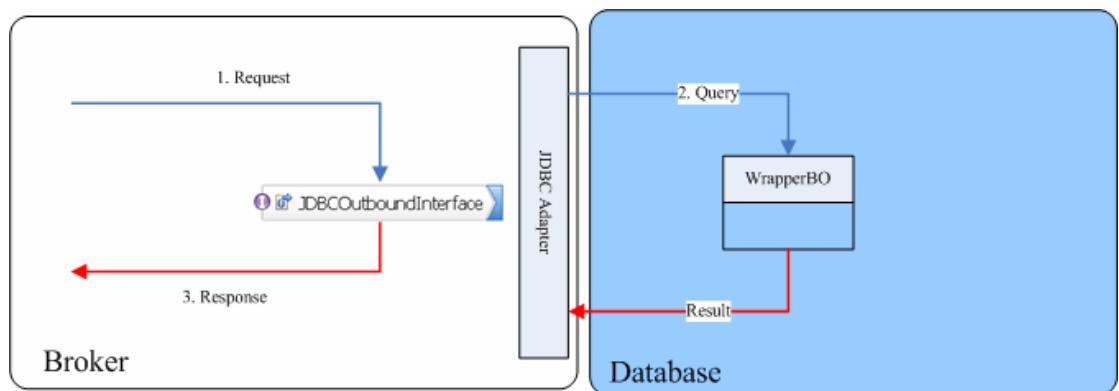
Chapter 11. Tutorial 10: Generate wrapper business objects (Oracle)

Wrapper business object is a top-level business object in business object hierarchy and it used to relate unrelated child business objects. Wrapper object needs a minimum of two table business objects to wrap them together.

This tutorial demonstrates how WebSphere Adapter for JDBC 7.0.0.0 generates wrapper business objects and creates record in tables using wrapper business objects.

About this task

In this scenario, an application SCA component raises a retrieve test request to the JDBC Outbound Interface. The JDBC adapter executes a SQL query to select all specific records back. Finally, the JDBC adapter converts the test result to a SDO and gives a response to the SCA component. The following figure represents this scenario:



Prepare to run through the tutorial

Extract the sample files

Replicas of the artifacts that you create when using the external service wizard are provided as sample files for your reference. Use these files to verify if the files you create using the external service wizard are correct.

Download the sample zip file and extract it into a directory of your choice (you may want to create a new directory).

Configuration prerequisites

Before configuring the adapter, you must complete the following tasks:

- Create a table
- Create an authentication alias
- Create a data source

Create a table

You must create the following table in the Oracle database before starting the scenario.

```
CREATE TABLE CUSTOMER (
    PKEY VARCHAR2(10) NOT NULL PRIMARY KEY,
    FNAME VARCHAR2(20) ,
    LNAME VARCHAR2(20) ,
    CCODE VARCHAR2(10) ) ;
```

Insert a record into the table you just created.

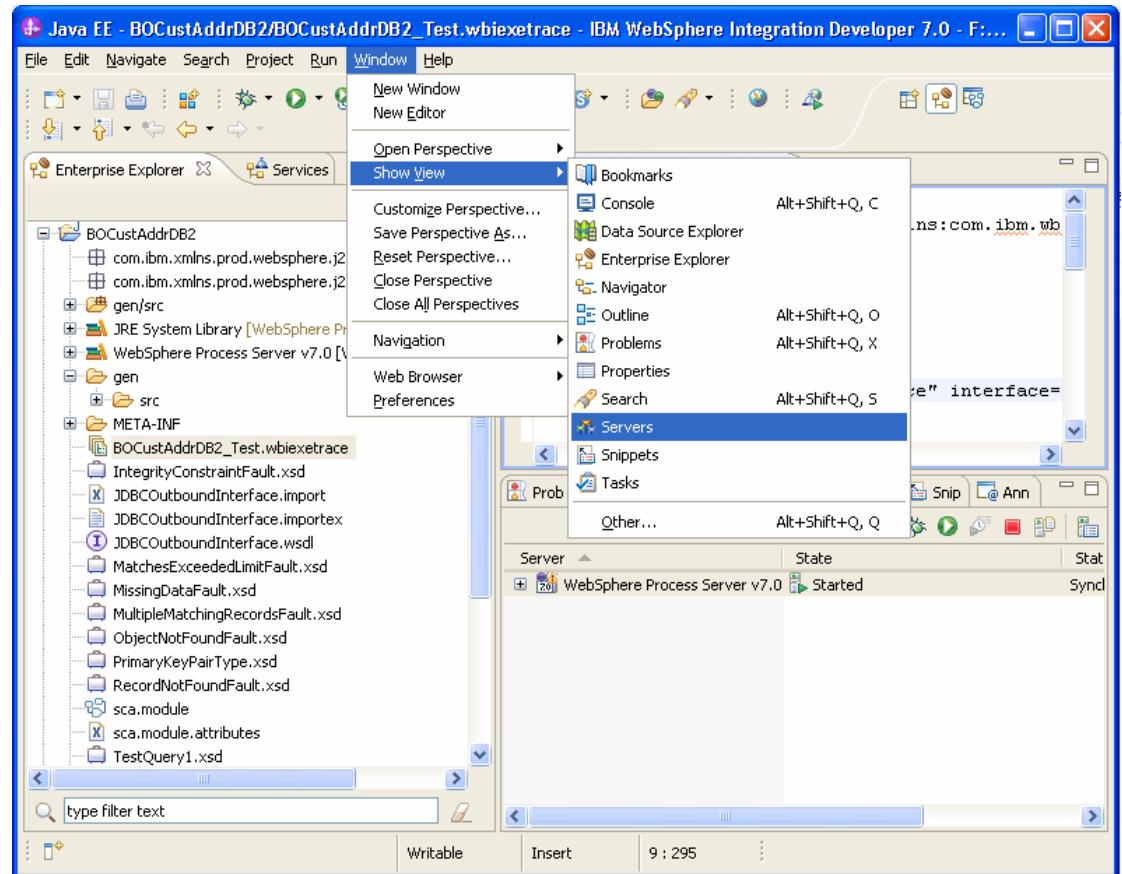
```
insert into customer values ('1000', 'testFname',
'testLname', 'testCcode')
```

Create an authentication alias

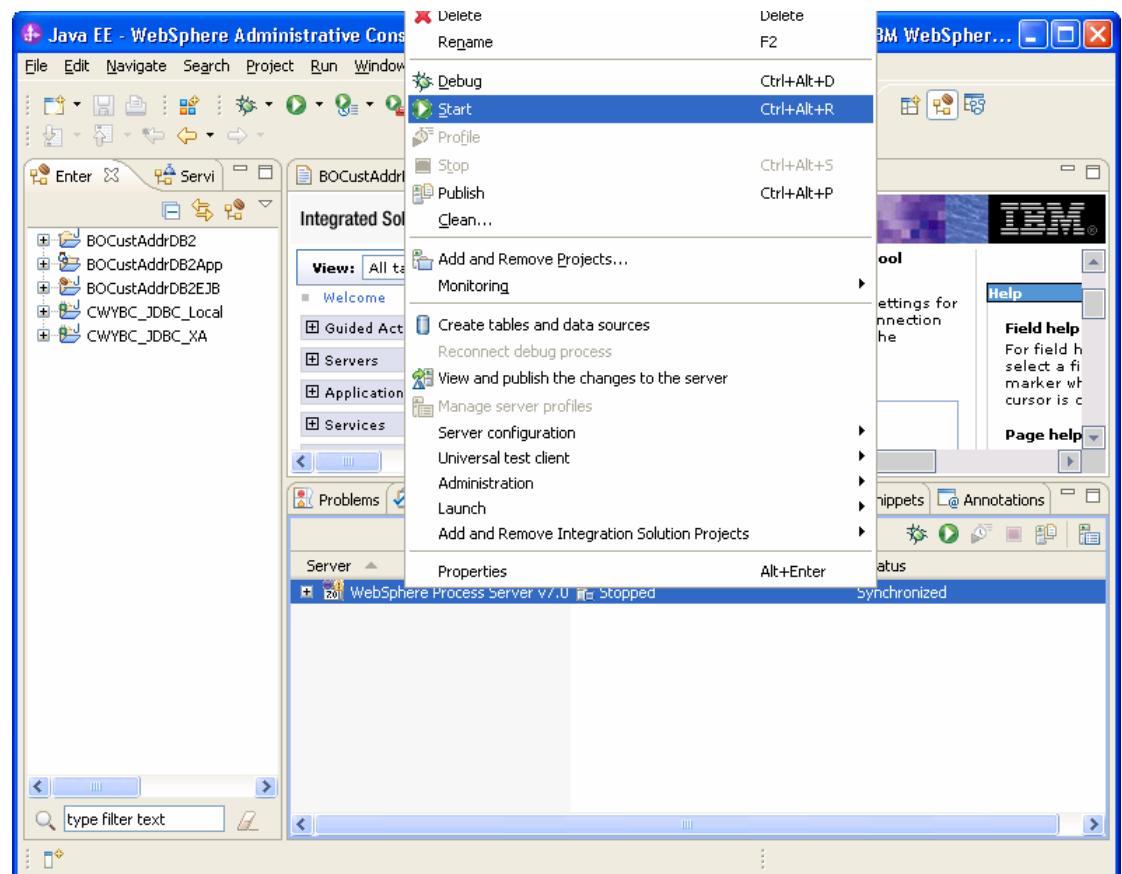
The authentication alias needs to be set because the data source created in the next section uses the username and password set in the authentication alias to connect to the database.

Follow these steps to set the authentication alias in the WebSphere Process Server administrative console.

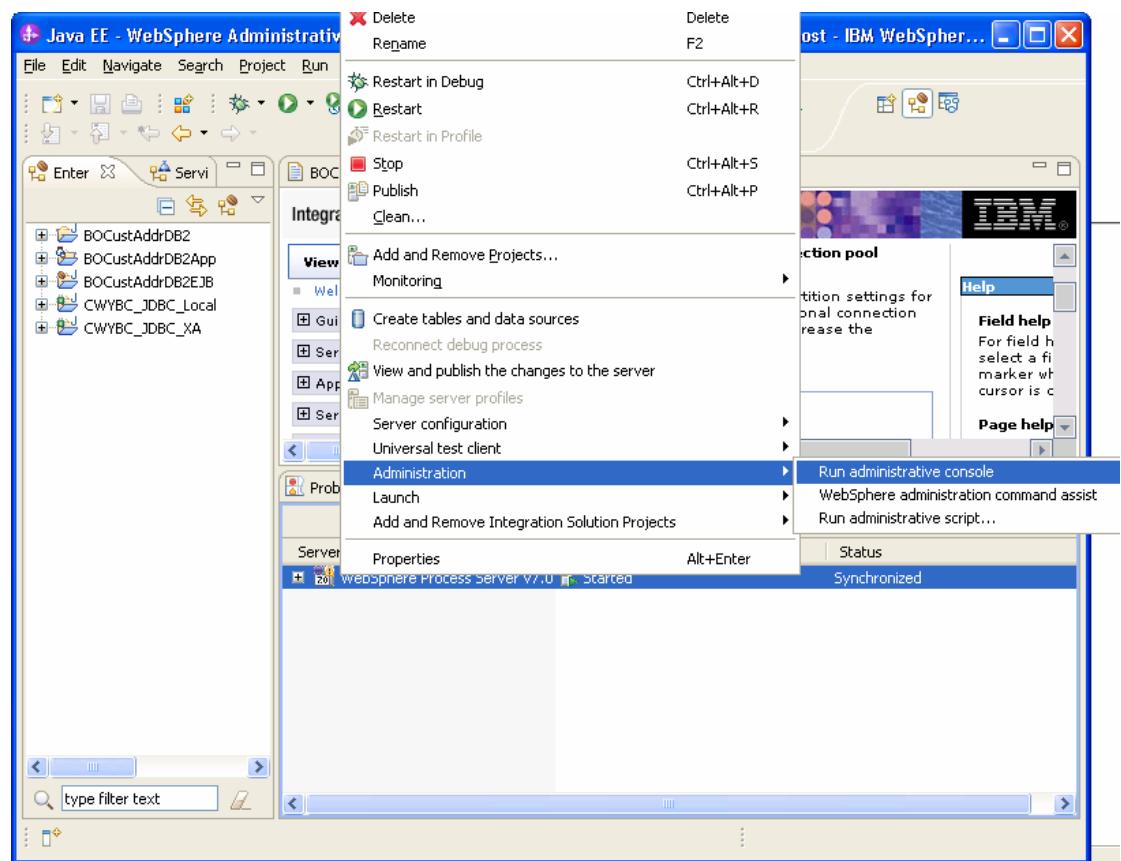
1. In WebSphere Integration Developer, switch to the **Servers** view by selecting **Window > Show View > Servers**.



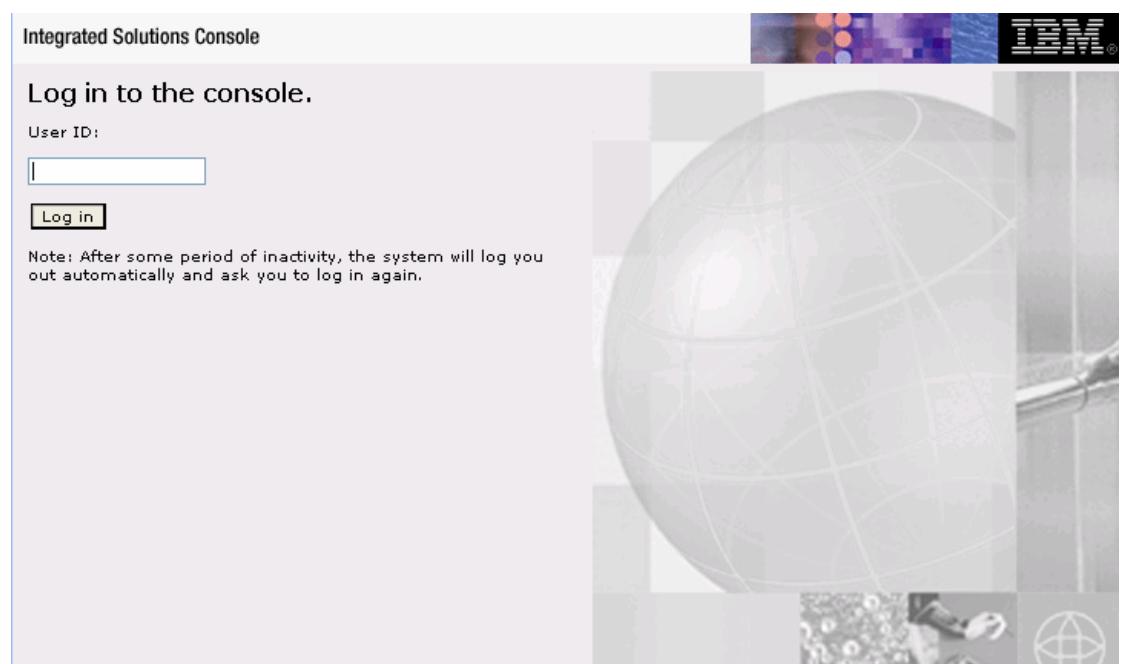
2. In the **Servers** view, right-click the server that you want to start and select **Start**.



3. After the server is started, right-click the server, and select **Administration > Run administrative console**.



4. Log on to the administrative console.



5. Click **Security → Global security**.



- Under **Java Authentication and Authorization Service**, click **J2C authentication Data**.

Integrated Solutions Console Welcome admin Help | Logout IBM

Secure administration, applications, and infrastructure Close page

Secure administration, applications, and infrastructure

The application serving environment is completely secured when administration is restricted. The applications and the infrastructure supports the administration and applications also are secured.

Configuration

Security Configuration Wizard Security Configuration Report

Administrative security

- Enable administrative security
 - [Administrative User Roles](#)
 - [Administrative Group Roles](#)

Application security

- Enable application security

Java 2 security

- Use Java 2 security to restrict application access to local resources
 - Warn if applications are granted custom permissions
 - Restrict access to resource authentication data

User account repository

- Current realm definition
- Local operating system

Authentication

- Use domain-qualified user names
- Web security
- RMI/IIOP security
- Java Authentication and Authorization Service
 - [Application logins](#)
 - [System logins](#)
 - [J2C authentication data](#)
- Authentication
 - Specifies a list of user identities and passwords for Java(TM) 2 connector security to use.
- External authorization providers
- Custom properties

A list of existing aliases is displayed.

The screenshot shows the 'Secure administration, applications, and infrastructure' section of the WebSphere console. A table lists four authentication entries:

Select	Alias	User ID	Description
<input type="checkbox"/>	BSpace JDBC Alias	TEST	Business Space Authorization Alias
<input type="checkbox"/>	CommonEventInfrastructureJMSAuthAlias	CEI	Authentication alias for the Common Event Infrastructure JMS Topics and Queues
<input type="checkbox"/>	SCA Auth Alias	SCA	This is the alias used by SCA to login to a secured SIBus
<input type="checkbox"/>	localhostNode01Cell/nlNode01/server1/EventAuthDataAliasDerby		Derby authentication alias for the Event Server

- Click **New** to create a new authentication entry. Type the alias name, and username and password to connect to the database. Click **OK**.

The screenshot shows the 'Global security > JAAS - J2C authentication data > New' dialog. The 'General Properties' section contains the following fields:

- * Alias: node1/Oracle
- * User ID: sample
- * Password: *****
- Description: (empty)

Buttons at the bottom include Apply, OK, Reset, and Cancel.

You have created an authentication alias that will be used to configure the adapter properties.

The screenshot shows the 'Integrated Solutions Console' interface with the 'Welcome' tab selected. On the left, a navigation tree includes 'View: All tasks', 'Welcome', 'Guided Activities', 'Servers', 'Applications', 'Services', 'Resources', 'Security' (with sub-options like Business Integration Security, Global security, Security domains, Administrative Authorization, SSL certificate and key management, Security auditing, Bus security), 'Environment', 'Integration Applications', and 'System administration'. The main panel displays a table of security aliases:

	Alias Name	User	Notes
<input type="checkbox"/>	BSpace JDBC Alias	wbiuser	B S A A
<input type="checkbox"/>	CommonEventInfrastructureJMSAuthAlias	wbiuser	A a C E I J a
<input type="checkbox"/>	SCA Auth Alias	wbiuser	T a S a S
<input type="checkbox"/>	localhostNode01Cell/nlNode01/server1/EventAuthDataAliasDerby		D a a E
<input type="checkbox"/>	nlNode01/luweiqin	luweiqin	
<input type="checkbox"/>	nlNode01/node1/Oracle	sample	C o

Total 6

Create a data source

Create a data source in WebSphere Process Server, which the adapter will use to connect to the database. This data source will be used in generating the artifacts for the module.

Note: This tutorial will use Oracle as the database and the Oracle thin driver, ojdbc6.jar.

Here are the steps to create the data source in the WebSphere Process Server administrative console.

1. In the administrative console, select **Environment** → **WebSphere variables**

The screenshot shows the 'Environment' section of the administrative console. The left sidebar lists 'Virtual hosts', 'Update global Web server plug-in configuration', 'WebSphere variables' (which is highlighted with a red box), 'Shared libraries', 'Replication domain', and 'Naming'. The right panel is currently empty.

2. On the right, click **ORACLE_JDBC_DRIVER_PATH** and specify the path of the ojdbc6.jar file in the **Value** field. Click **OK**.

Configuration

General Properties

* Name: ORACLE_JDBC_DRIVER_PATH

Value: F:\DBDriver\Oracle10

Description: The directory that contains the Oracle thin or oci8 JDBC Driver.

Apply OK Reset Cancel

The variable is added and appears in the list.

The screenshot shows the Integrated Solutions Console interface. On the left, there is a navigation tree with categories like Welcome, Guided Activities, Servers, Applications, Services, Resources, Security, Environment, Integration Applications, and System administration. Under the Environment category, the 'WebSphere variables' option is selected. On the right, a table lists various environment variables:

JAVA_HOME	\$(WAS_INSTALL_ROOT)\java
JVM_CACHE	
LOCALHOST_NAME	localhost
LOG_ROOT	\$(USER_INSTALL_ROOT)/logs
MICROSOFT_JDBC_DRIVER_NATIVEPATH	
MICROSOFT_JDBC_DRIVER_PATH	
MQ_INSTALL_ROOT	\$(WAS_INSTALL_ROOT)\lib\W
ORACLE_JDBC_DRIVER_PATH	F:\DBDriver\Oracle10
OS400_NATIVE_JDBC40_DRIVER_PATH	
OS400_NATIVE_JDBC_DRIVER_PATH	
OS400_TOOLBOX_JDBC_DRIVER_PATH	
SCA_BUS_ID	localhostNode01Cell
SERVER_LOG_ROOT	\$(LOG_ROOT)/server1
SYBASE_JDBC_DRIVER_PATH	

3. Select **Resources → JDBC -> JDBC Providers**.



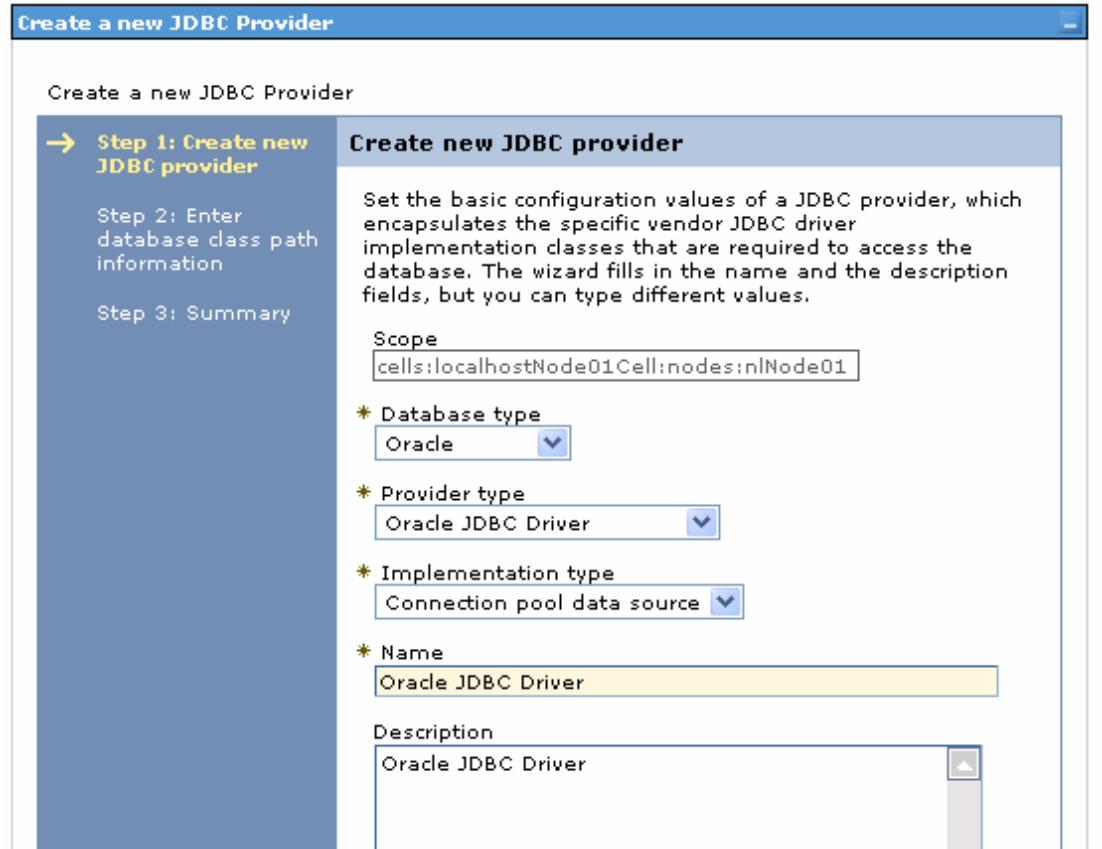
- Click **New** in the JDBC providers window.

The screenshot shows the 'JDBC providers' configuration page. The 'Scope' dropdown is set to 'Node=nINode01'. The table lists one provider:

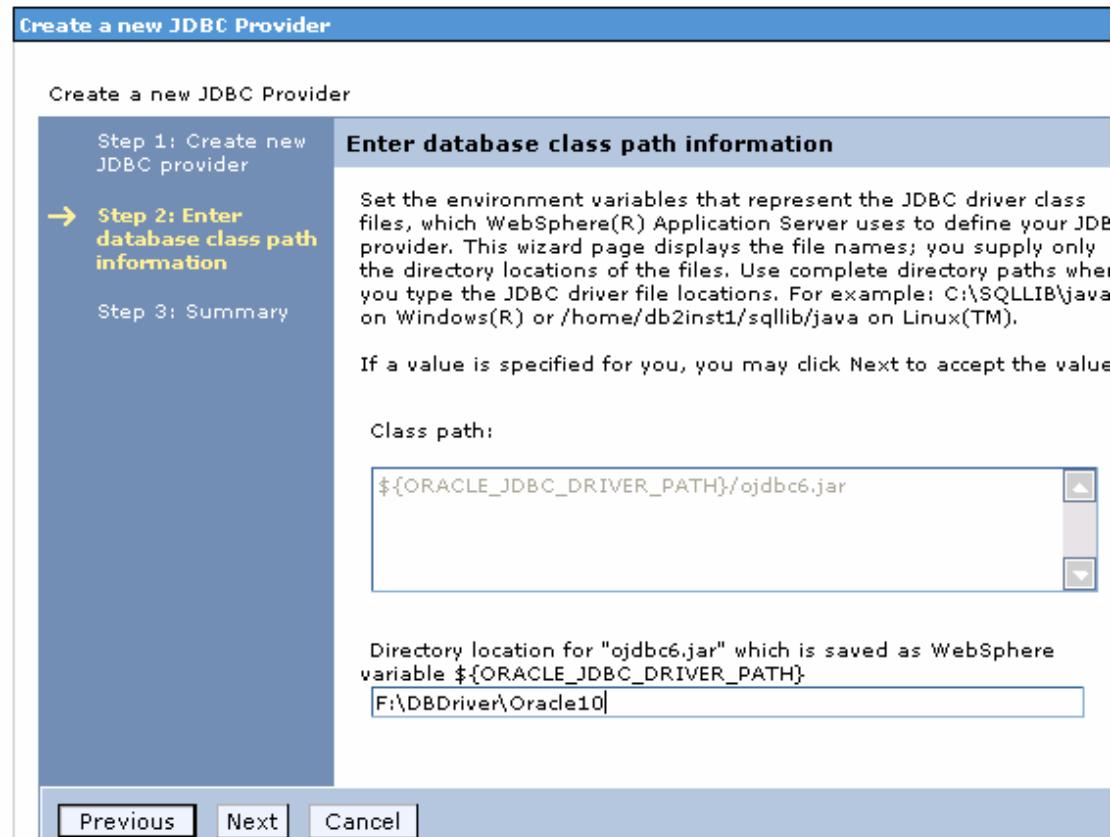
Select	Name	Scope	Description
<input type="checkbox"/>	Derby JDBC Provider (XA)	Node=nINode01	JDBC Provider for WPS/WESB

The right panel contains help sections: 'Field help' (for field information), 'Page help' (with a link to more information about the page), and 'Command Assistance' (with a link to view administrative scripting command for last action).

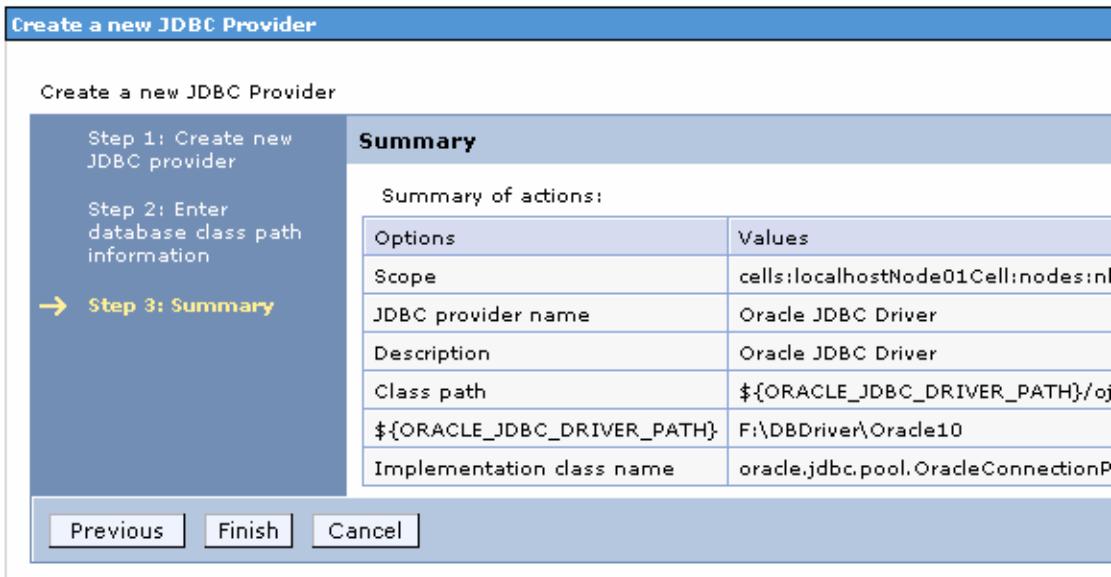
- Click **New**. Select the Oracle database with a connection pool data source for the Oracle JDBC driver. Click **Next**.



6. In the Enter database classpath information page, enter the following value for the **Class path** field:
\$(ORACLE_JDBC_DRIVER_PATH)/ojdbc6.jar, where
\$(ORACLE_JDBC_DRIVER_PATH) is library path for the run time.
Because you have added the ojdbc6.jar file to this path, you must specify that path here.
7. Click **Next**.



8. Click **Finish**.



9. Under **Additional Properties**, select **Data sources**. Click **New**.

The screenshot shows the 'JDBC providers' interface with the 'Data sources' tab selected. A message at the top explains that this page allows editing settings for a datasource associated with the selected JDBC provider. It mentions that the datasource object supplies your application with connections for accessing the database. A link to a guided activity is provided for more information.

Below the message are several buttons: 'New', 'Delete', 'Test connection', and 'Manage state...'. There are also icons for creating, deleting, and managing datasources. A search bar at the top allows filtering by 'Name', 'JNDI name', 'Scope', 'Provider', 'Description', and 'Category'. The main table displays one entry: 'None' with a count of 'Total 0'.

10. Type any value in the **JNDI name** field, and select the authentication alias “OracleDS” that you created earlier from the **Component-managed authentication alias and XA recovery authentication alias** list. Click **Next**.

The screenshot shows the 'Create a data source' wizard, specifically Step 1: Enter basic data source information. The left sidebar lists steps: Step 1 (current), Step 2, Step 3, and Step 4: Summary. The right panel contains fields for 'Scope' (set to 'cells:localhostNode01Cell:nodes:n1Node01'), 'JDBC provider name' (set to 'Oracle JDBC Driver'), and two required fields: 'Data source name' (set to 'Oracle JDBC Driver DataSource') and 'JNDI name' (set to 'OracleDS').

11. Provide the appropriate URL value and select a data store helper class name from the **Data store helper class name** list as shown in the following figure. Click **Next**.

Create a data source

Step 1: Enter basic data source information

→ Step 2: Enter database specific properties for the data source

Step 3: Setup security aliases

Step 4: Summary

Enter database specific properties for the data source

Set these database-specific properties, which are required by the database vendor JDBC driver to support the connections that are managed through the datasource.

Name	Value
* URL	jdbc:oracle:thin@9.181.84.13
* Data store helper class name	Oracle10g data store helper
<input checked="" type="checkbox"/> Use this data source in container managed persistence (CMP)	

Previous Next Cancel

12. In the Setup security aliases window, configure the aliases.

Create a data source

Step 1: Enter basic data source information

Step 2: Enter database specific properties for the data source

→ Step 3: Setup security aliases

Step 4: Summary

Setup security aliases

Select the authentication values for this resource.

Component-managed authentication alias
nlNode01/node1/Oracle

Mapping-configuration alias
(none)

Container-managed authentication alias
(none)

Note: You can create a new J2C authentication alias by accessing one of the following links. Clicking on a link will cancel the wizard and your current selections will be lost.

[Global J2C authentication alias](#)
[Security domains](#)

Previous Next Cancel

13. In the Summary page, review the values entered for the data source and click **Finish**.

Create a data source

Step 1: Enter basic data source information Step 2: Enter database specific properties for the data source Step 3: Setup security aliases → Step 4: Summary	Summary Summary of actions: <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="background-color: #d9e1f2;">Options</th> <th style="background-color: #d9e1f2;">Values</th> </tr> </thead> <tbody> <tr> <td>Scope</td> <td>cells:localhostNode01Cell:nodes:nlNode01</td> </tr> <tr> <td>Data source name</td> <td>Oracle JDBC Driver DataSource</td> </tr> <tr> <td>JNDI name</td> <td>OracleDS</td> </tr> <tr> <td>Select an existing JDBC provider</td> <td>Oracle JDBC Driver</td> </tr> <tr> <td>Implementation class name</td> <td>oracle.jdbc.pool.OracleConnectionPoolDataSource</td> </tr> <tr> <td>URL</td> <td>jdbc:oracle:thin:@9.181.84.136:1521:ord</td> </tr> <tr> <td>Data store helper class name</td> <td>com.ibm.websphere.rsadapter.Oracle10gDataStoreHelper</td> </tr> <tr> <td>Use this data source in container managed persistence (CMP)</td> <td>true</td> </tr> <tr> <td>Component-managed authentication alias</td> <td>nlNode01/node01/Oracle</td> </tr> <tr> <td>Mapping-configuration alias</td> <td>(none)</td> </tr> <tr> <td>Container-managed authentication alias</td> <td>(none)</td> </tr> </tbody> </table>	Options	Values	Scope	cells:localhostNode01Cell:nodes:nlNode01	Data source name	Oracle JDBC Driver DataSource	JNDI name	OracleDS	Select an existing JDBC provider	Oracle JDBC Driver	Implementation class name	oracle.jdbc.pool.OracleConnectionPoolDataSource	URL	jdbc:oracle:thin:@9.181.84.136:1521:ord	Data store helper class name	com.ibm.websphere.rsadapter.Oracle10gDataStoreHelper	Use this data source in container managed persistence (CMP)	true	Component-managed authentication alias	nlNode01/node01/Oracle	Mapping-configuration alias	(none)	Container-managed authentication alias	(none)
Options	Values																								
Scope	cells:localhostNode01Cell:nodes:nlNode01																								
Data source name	Oracle JDBC Driver DataSource																								
JNDI name	OracleDS																								
Select an existing JDBC provider	Oracle JDBC Driver																								
Implementation class name	oracle.jdbc.pool.OracleConnectionPoolDataSource																								
URL	jdbc:oracle:thin:@9.181.84.136:1521:ord																								
Data store helper class name	com.ibm.websphere.rsadapter.Oracle10gDataStoreHelper																								
Use this data source in container managed persistence (CMP)	true																								
Component-managed authentication alias	nlNode01/node01/Oracle																								
Mapping-configuration alias	(none)																								
Container-managed authentication alias	(none)																								

14. Click **Save** to save the changes.

JDBC providers

Messages

⚠ Changes have been made to your local configuration. You can:

- [Save](#) directly to the master configuration.
- [Review](#) changes before saving or discarding.

⚠ The server may need to be restarted for these changes to take effect.

[JDBC providers > Oracle JDBC Driver > Data sources](#)

Use this page to edit the settings of a datasource that is associated with your selected JDBC provider. The datasource object supplies your application with connections for accessing the database. Learn more about this task in a [guided activity](#). A guided activity provides a list of task steps and more general information about the topic.

Preferences

New	Delete	Test connection	Manage state...			
Select	Name ▾	JNDI name ▾	Scope ▾			
Provider ▾	Description ▾	Category ▾				
You can administer the following resources:						
<input type="checkbox"/>	Oracle JDBC Driver DataSource	OracleDS	Node=nlNode01	Oracle JDBC Driver	New JDBC Datasource	
Total 1						

15. Select the data source you just created and click **Test connection**.

The screenshot shows the 'JDBC providers' page under 'Oracle JDBC Driver'. A message at the top indicates that a test connection is in progress. Below the message, a table lists a single data source entry:

Select	Name	JNDI name	Scope	Provider	Description	Category
<input checked="" type="checkbox"/>	Oracle JDBC Driver DataSource	OracleDS	Node=nINode01	Oracle JDBC Driver	New JDBC Datasource	

Total 1

The connection should succeed as indicated by the message shown in the following figure. If you experience problems with the test connection, refer to the "Troubleshooting" section.

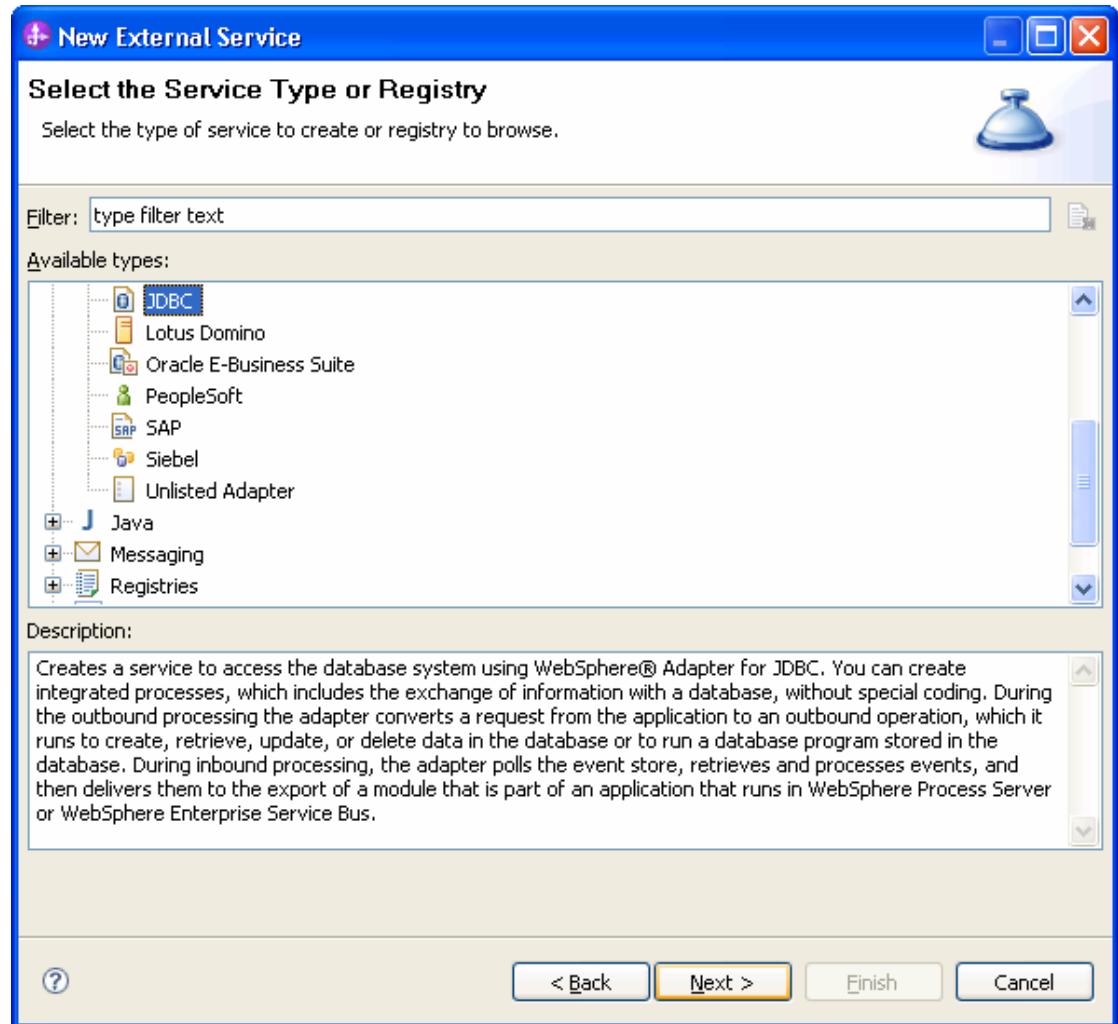
The screenshot shows the 'JDBC providers' page under 'Oracle JDBC Driver'. A message in the 'Messages' section states: 'The test connection operation for data source Oracle JDBC Driver DataSource on server server1 at node nINode01 was successful.' Below the message, the table from the previous screenshot is shown again, but the 'Select' column for the row is now empty.

The data source is created and it will be used by the adapter to connect to the database.

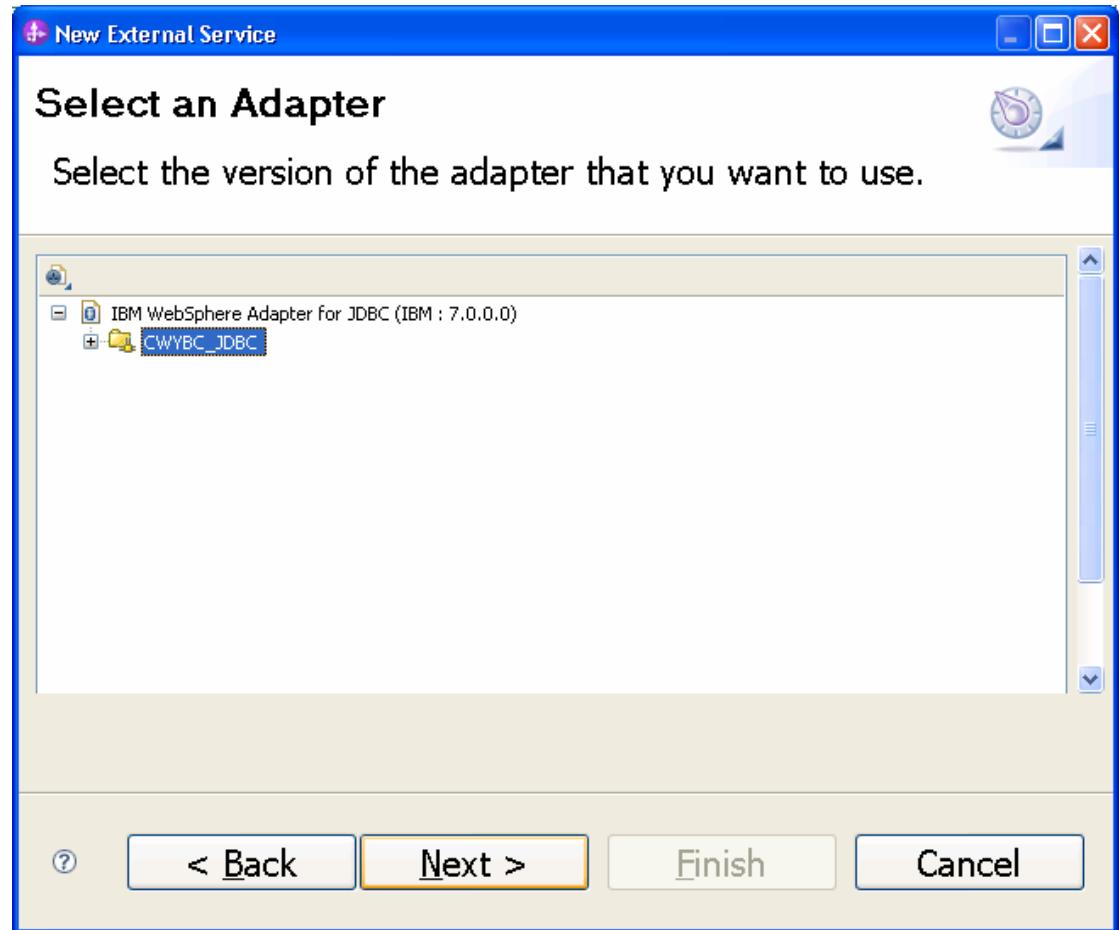
Configure the adapter for outbound processing

Run the external service wizard to specify business objects, services, and configuration details.

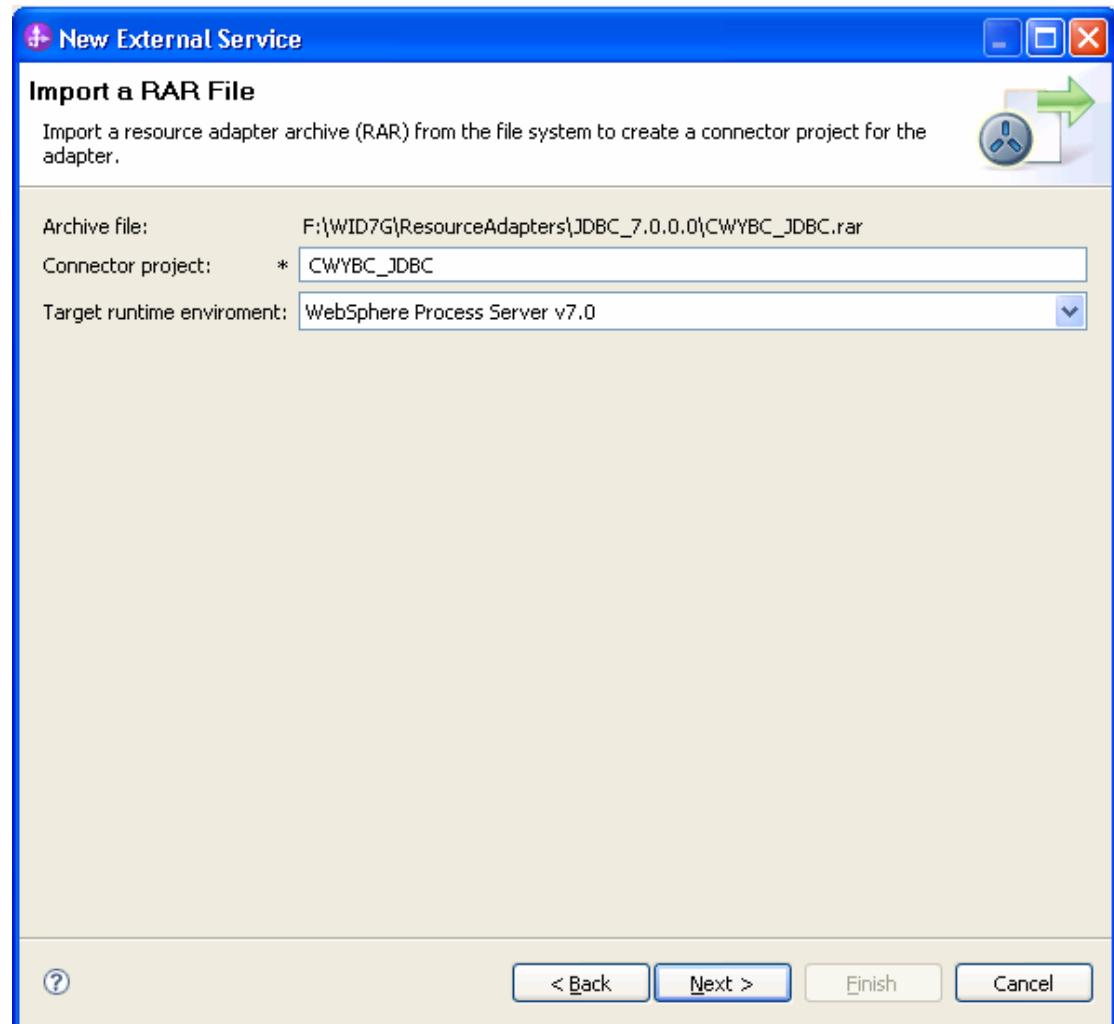
1. Switch to the Business Integration Perspective in WebSphere Integration Developer by selecting **Window -> Open Perspective Business Integration**.
2. Start the external service wizard by selecting **File-> New -> External Service**.
3. In the **Available Types** area, select **Adapters > JDBC** and click **Next**.



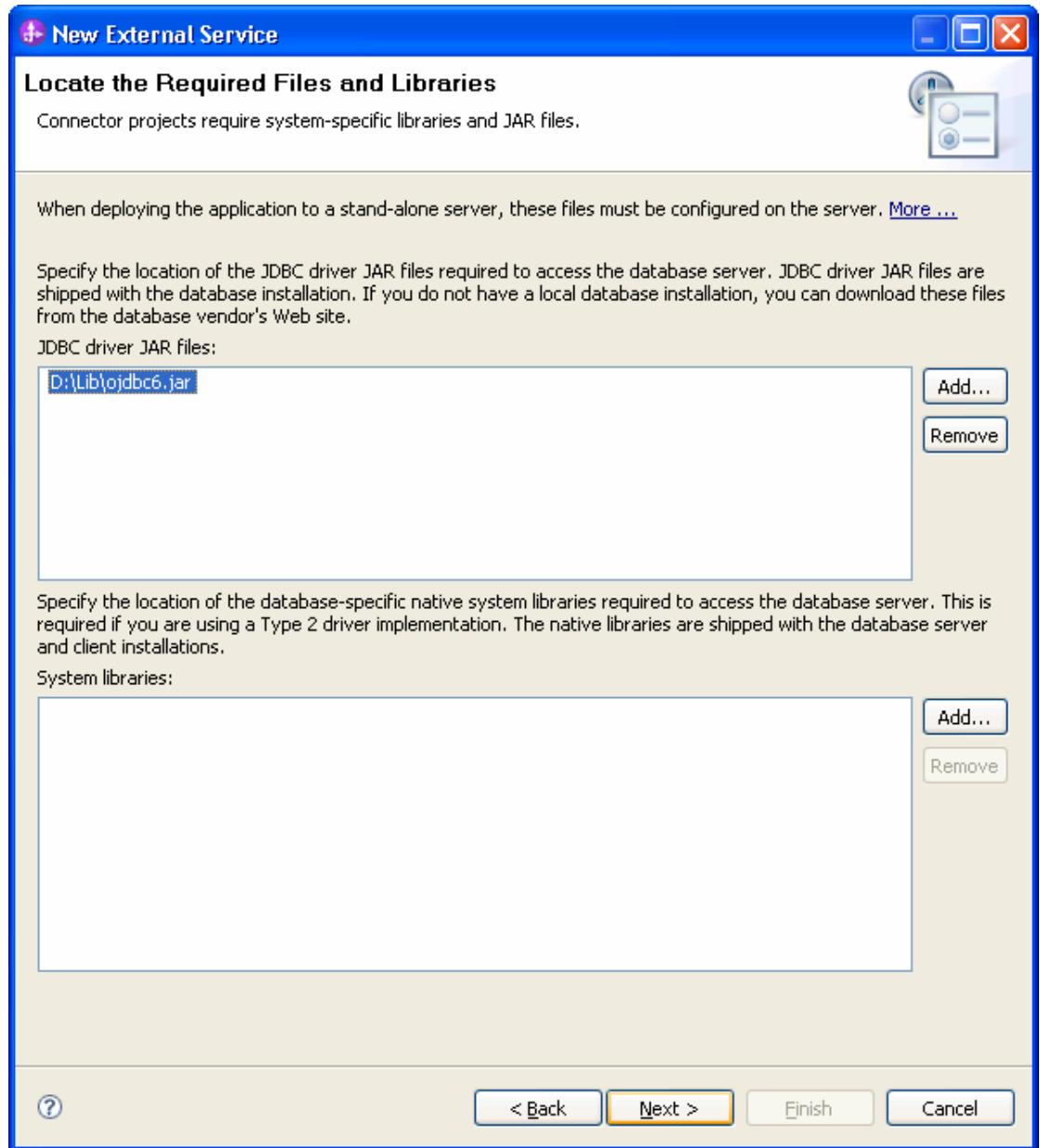
4. Select **IBM WebSphere Adapter for JDBC** and click **Next**.



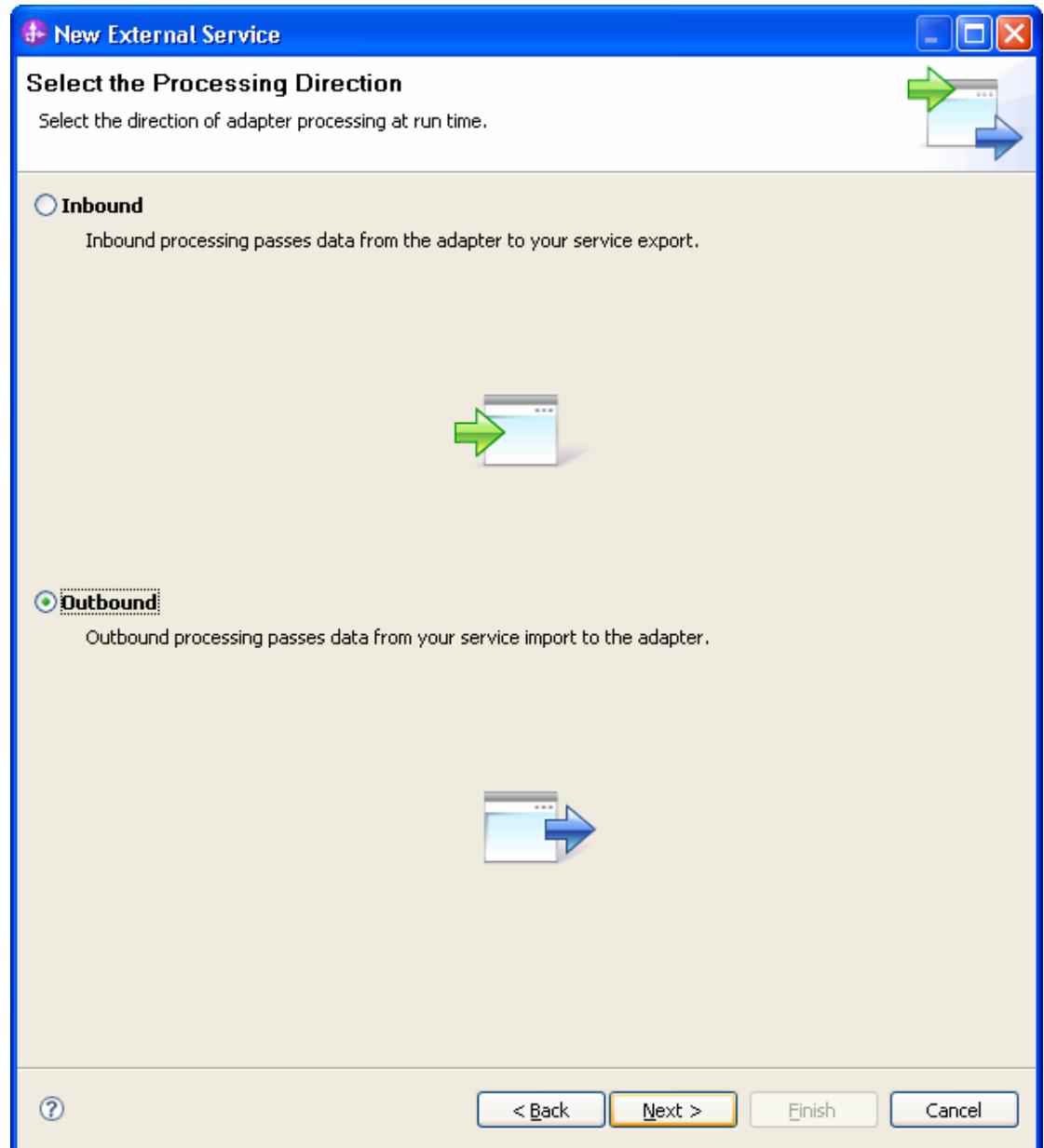
5. In the **Connector project** field enter **CWYBC_JDBC**.
6. In the **Target runtime environment** field, select the appropriate runtime and click **Next**.



7. In the **JDBC driver JAR files** field, click **Add**, to add the JDBC driver class to connect to the database. Browse to select the driver JAR file and click **Next**.



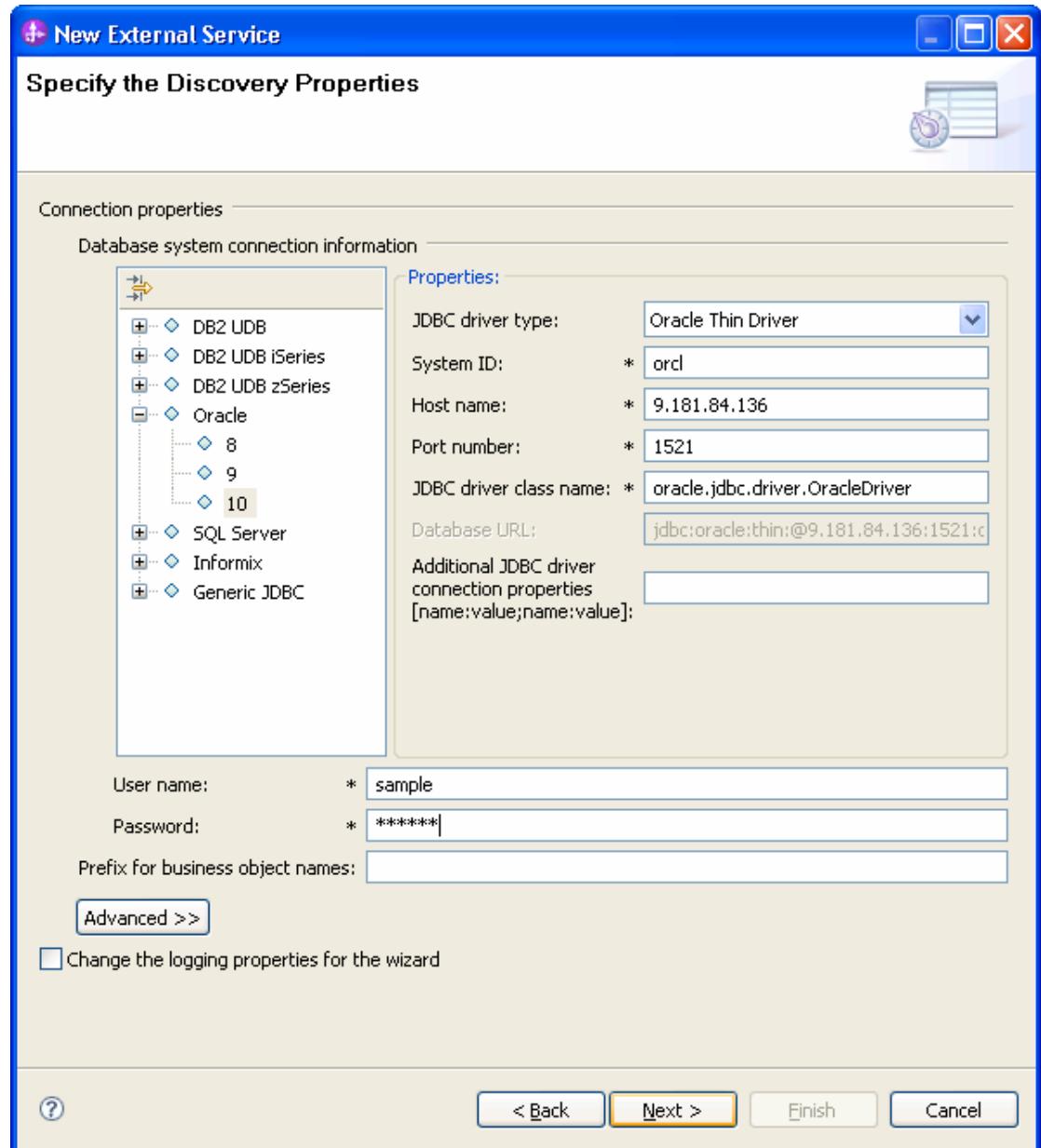
8. Select **Outbound** and click **Next**.



Set connection properties for the external service wizard

To connect to the Oracle database:

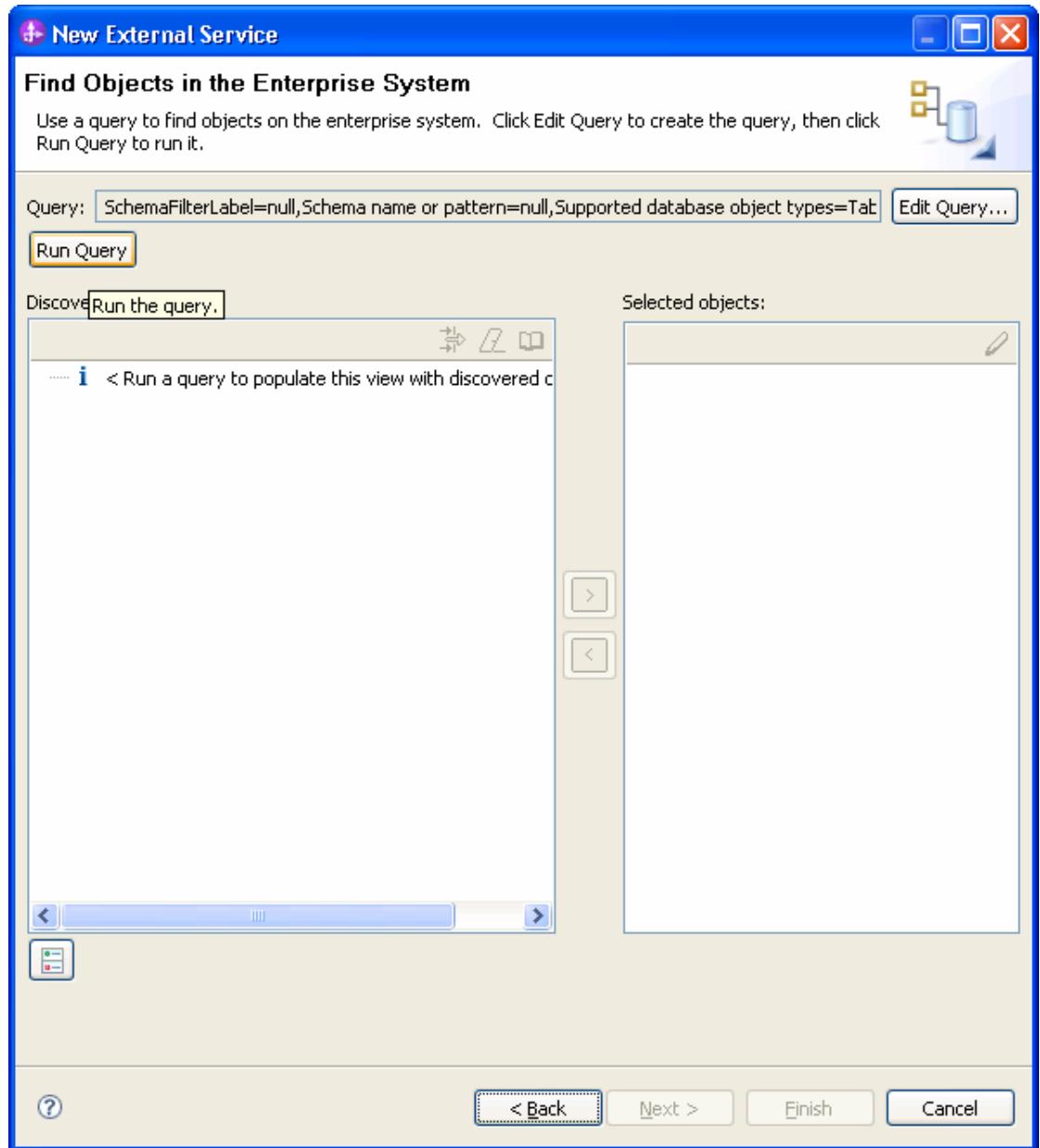
1. Expand the **Oracle** node in the **Database system connection information** area and select **10**.
2. Enter values in the **System ID**, **Host name**, **Port number**, **User name** and **Password** fields, and click **Next**.



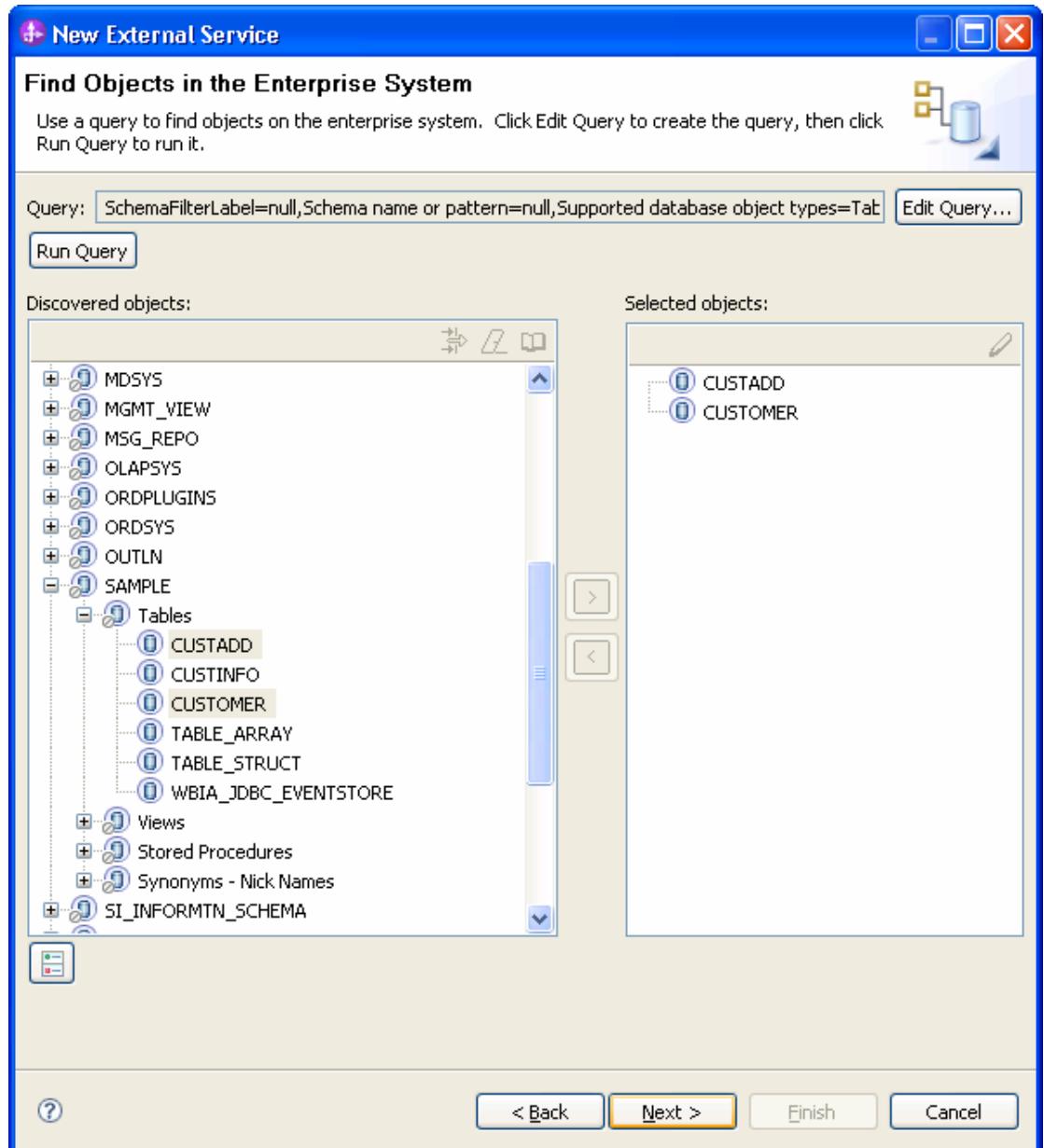
Select the business objects and services to be used with the adapter

Follow these steps to select the Customer and Address business object:

1. In the Find Objects in Enterprise System window, click **Run Query**.

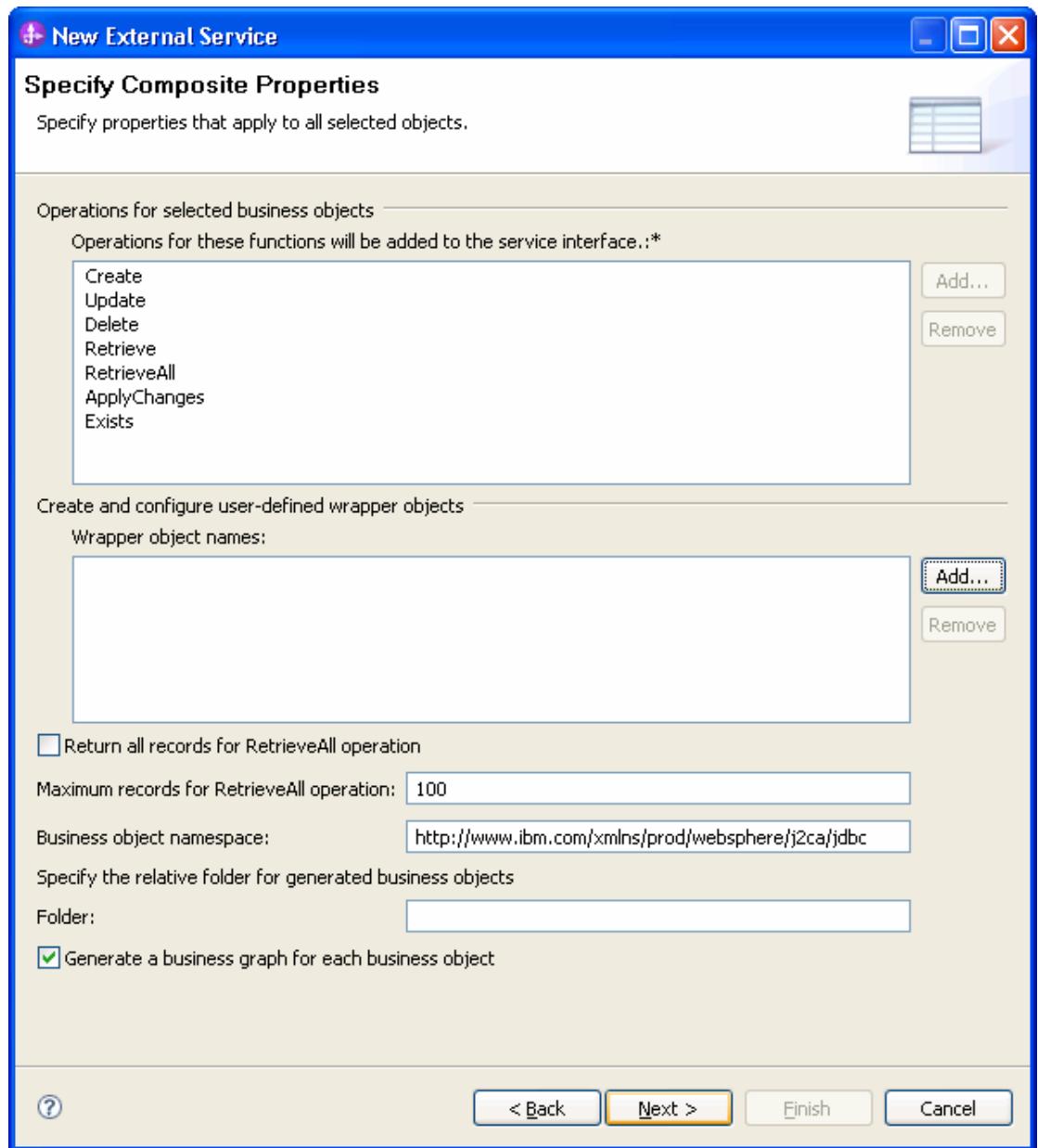


2. In the Discovered objects pane, select the **SAMPLE** (for this tutorial only) node, expand it and then select the **Tables** node and expand it.
3. Select the **CUSTOMER** and **CUSTADD** tables and click .

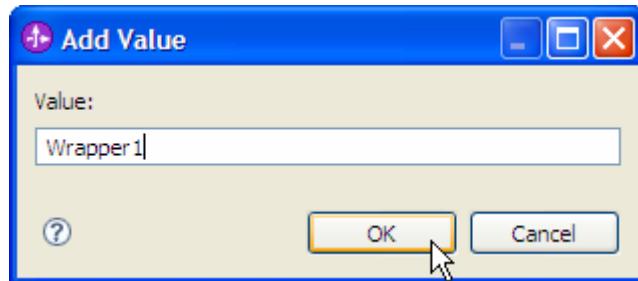


Note: Remember Wrapper business objects needs minimum two table objects.

4. Click **Next**. The Specify Composite Properties window is displayed.
5. In the **Wrapper object names** area, click **Add**.

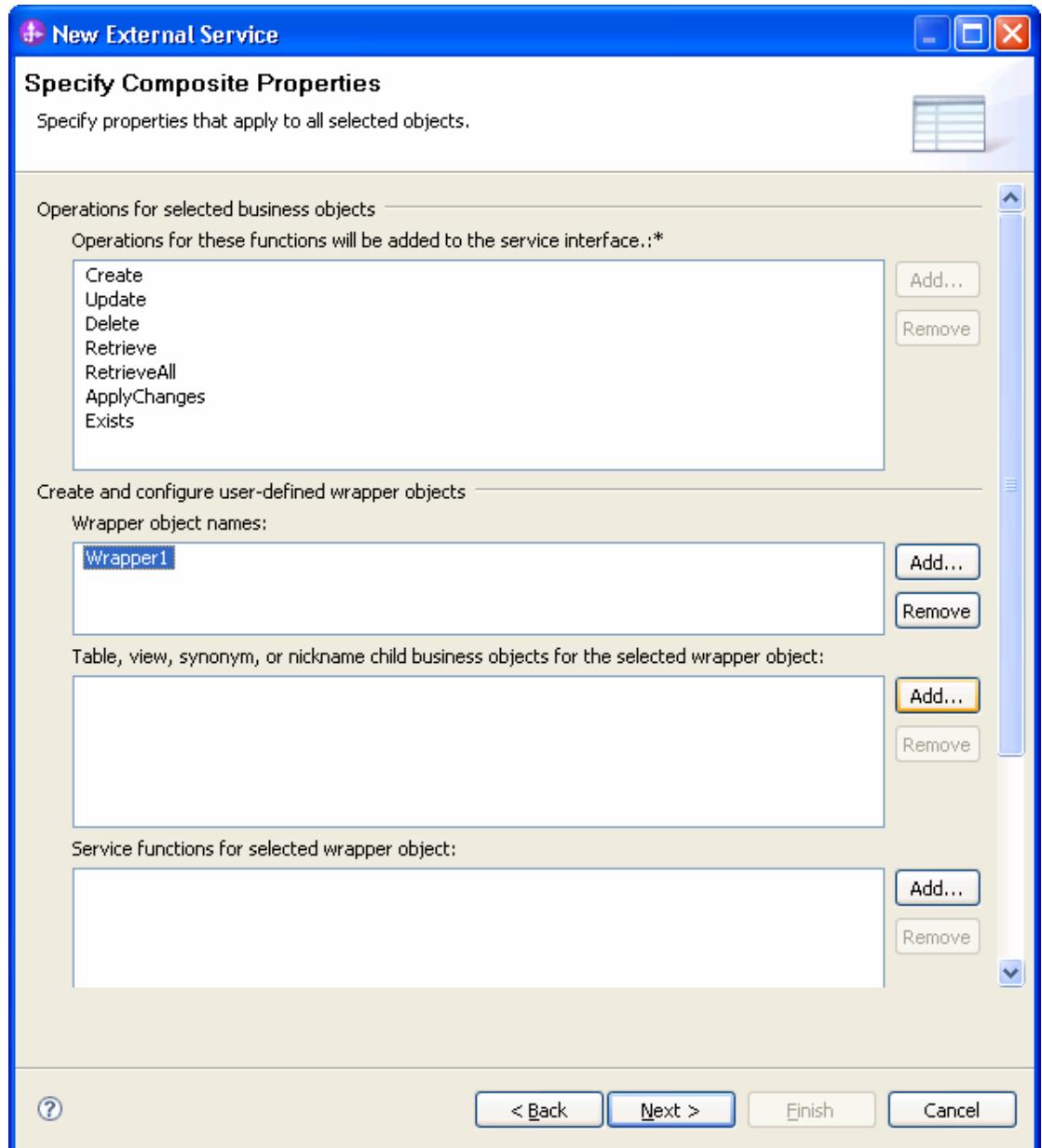


6. In the Add Value window, specify the name for the wrapper. Enter **Wrapper1** and click **OK**.

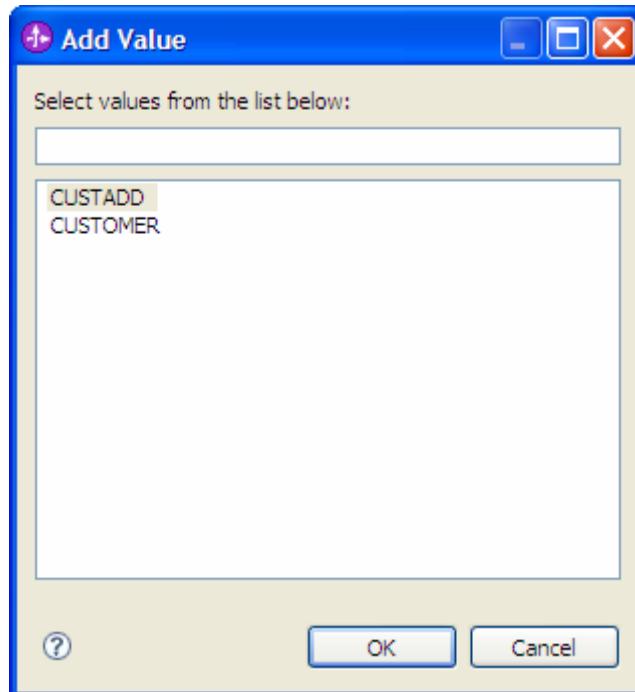


Wrapper1 is added into the **Wrapper object names** area.

7. In the **Table, View or nickname child business objects for the selected wrapper object** area, click **Add** to add child table business objects for the wrapper.

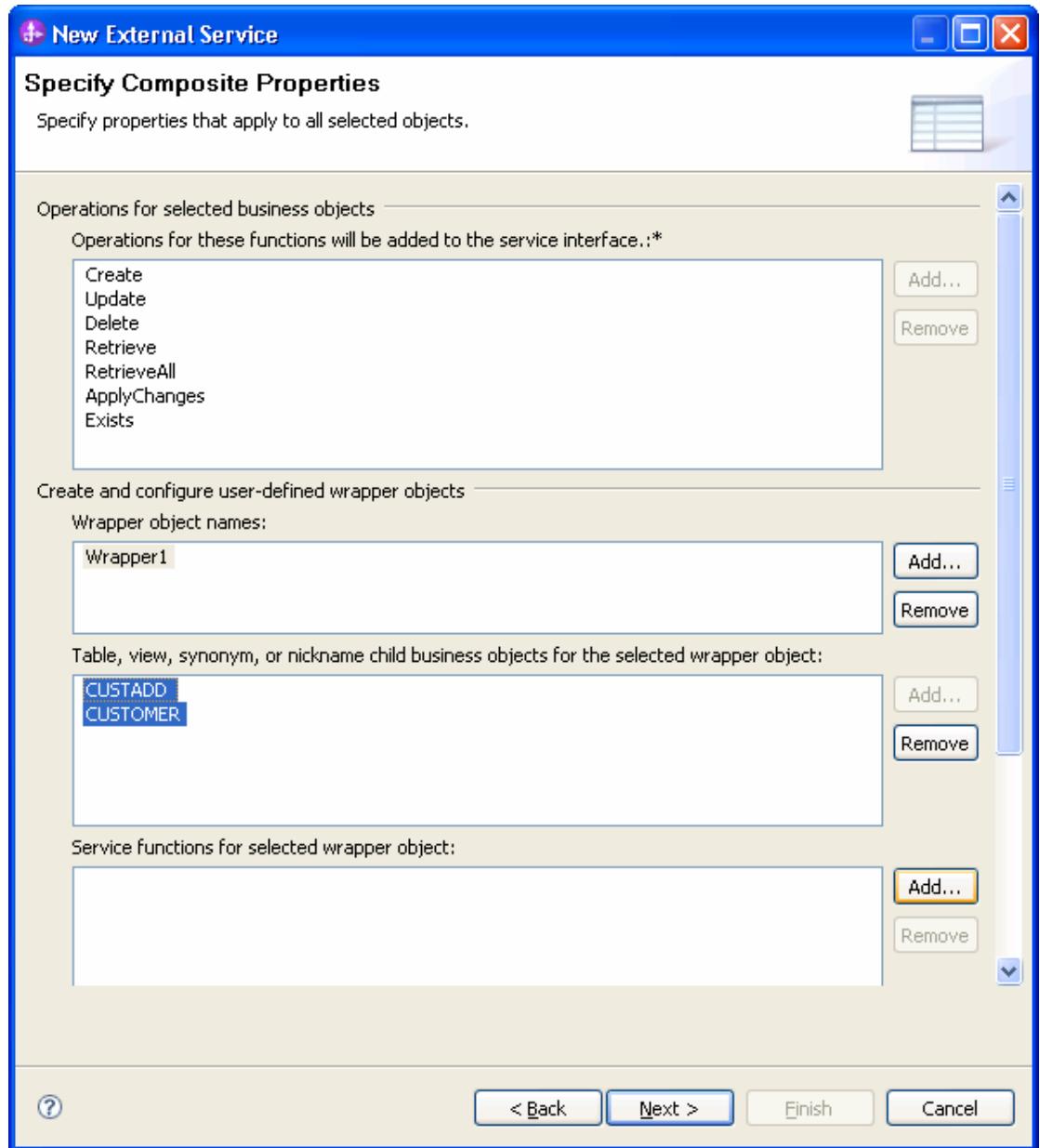


8. In the Add Value window, select CUSTADD and CUSTOMER tables and click **OK**.

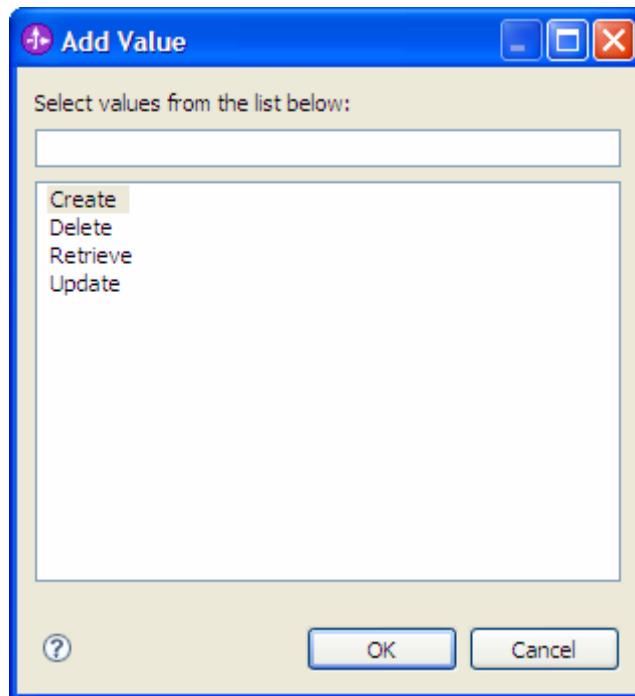


Both CUSTADD and CUSTOMER tables are added into child business objects for the selected wrapper object.

9. In the **Service functions for the selected wrapper object** area, click **Add** to add service functions to the wrapper.



10. In the Add Value window, select the Create and click **OK**.



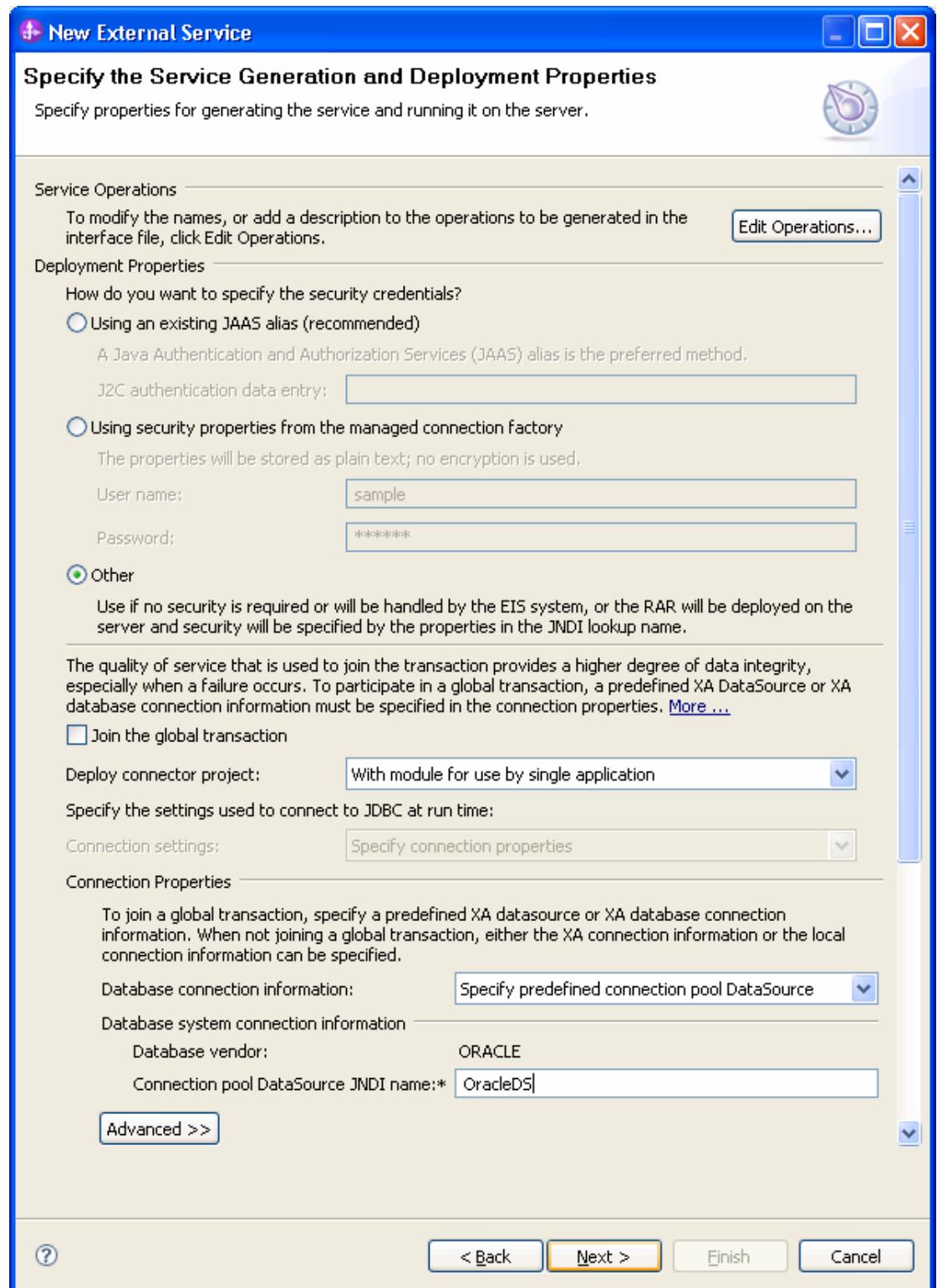
The selected service operation is added into the **Service functions for the selected wrapper object** area.

11. Accept the default values for the other fields and click **Next**.

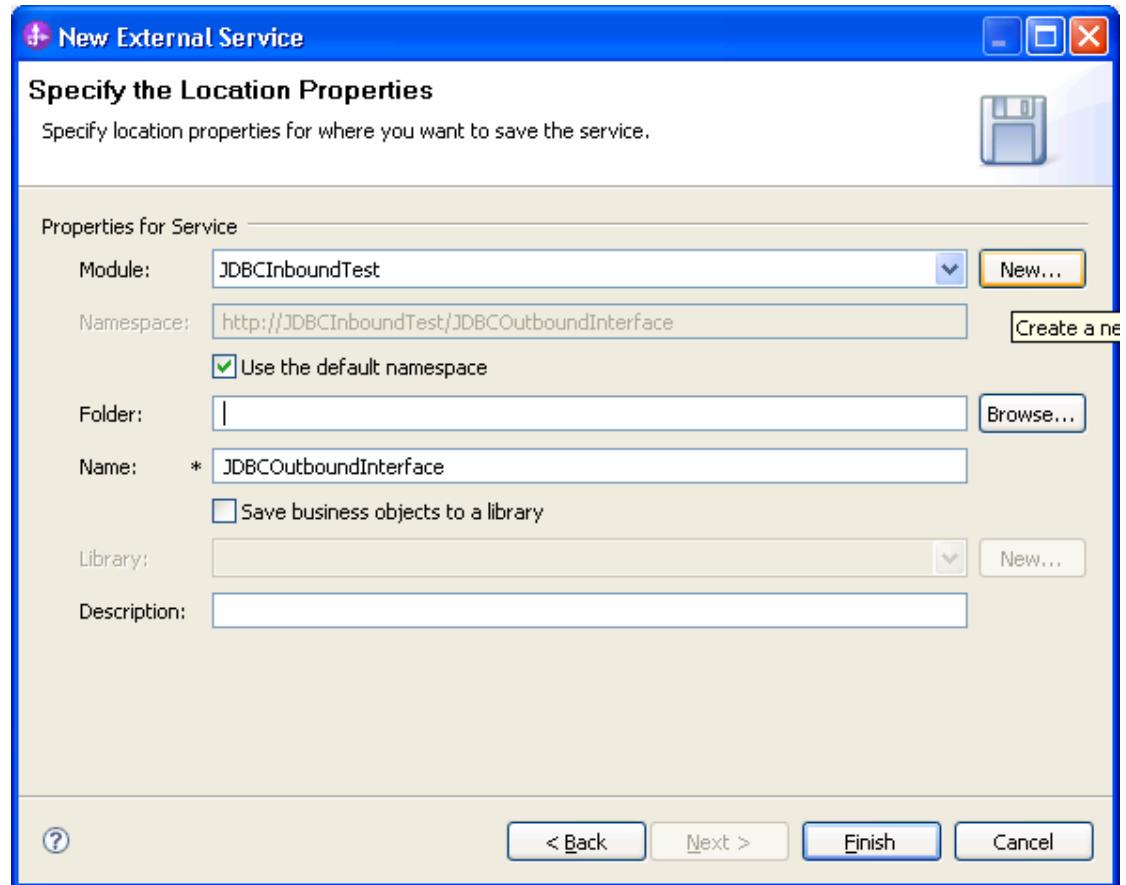
Generate business object definitions and related artifacts

Follow these steps to generate the business object definitions.

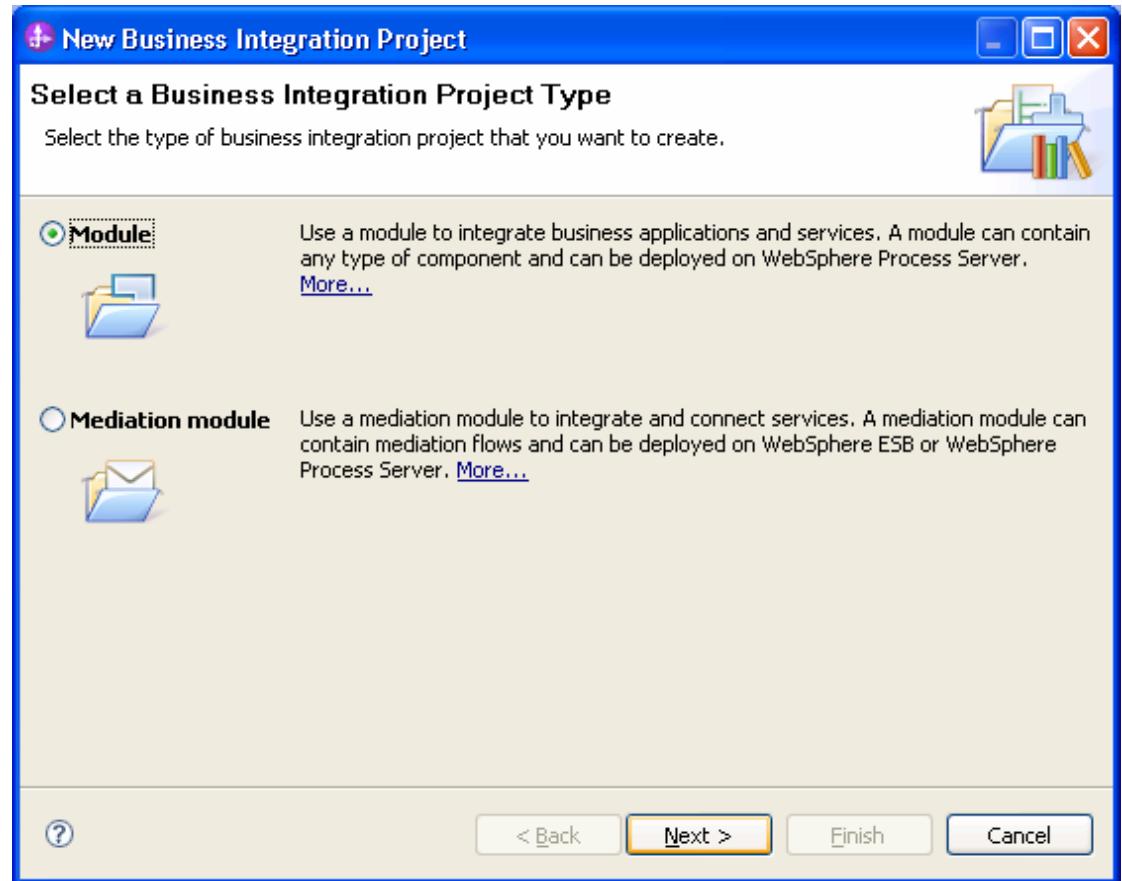
1. In the Specify the Service Generation and Deployment Properties window, perform the following steps:
 - a) Select **Other** for security options under **Deployment Properties**. Clear the **Join the global transaction** check box.
 - b) Select **Specify predefined connection pool DataSource** from the **Database connection information** list.
 - c) Enter **OracleDS** in the **Connection pool DataSource JNDI Name** field, and click **Next**.



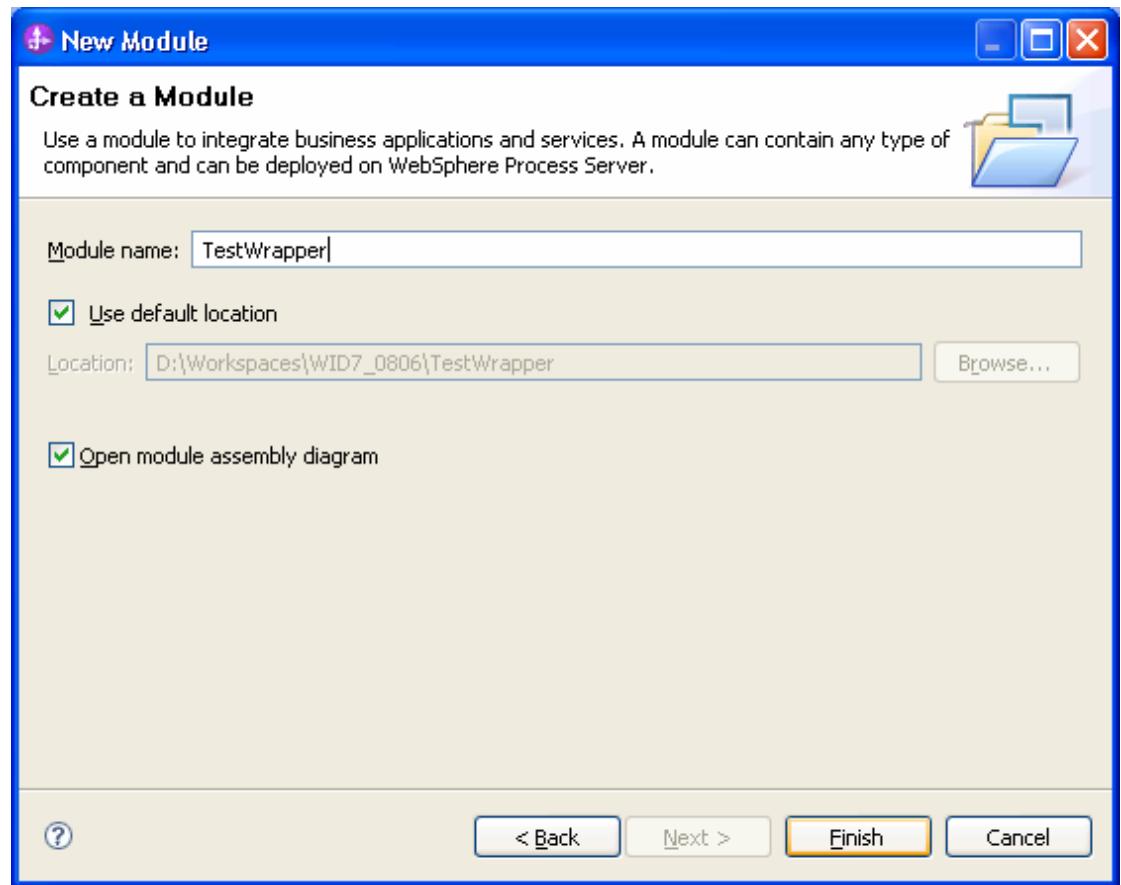
2. Click **New** in the Specify the Location Properties window.



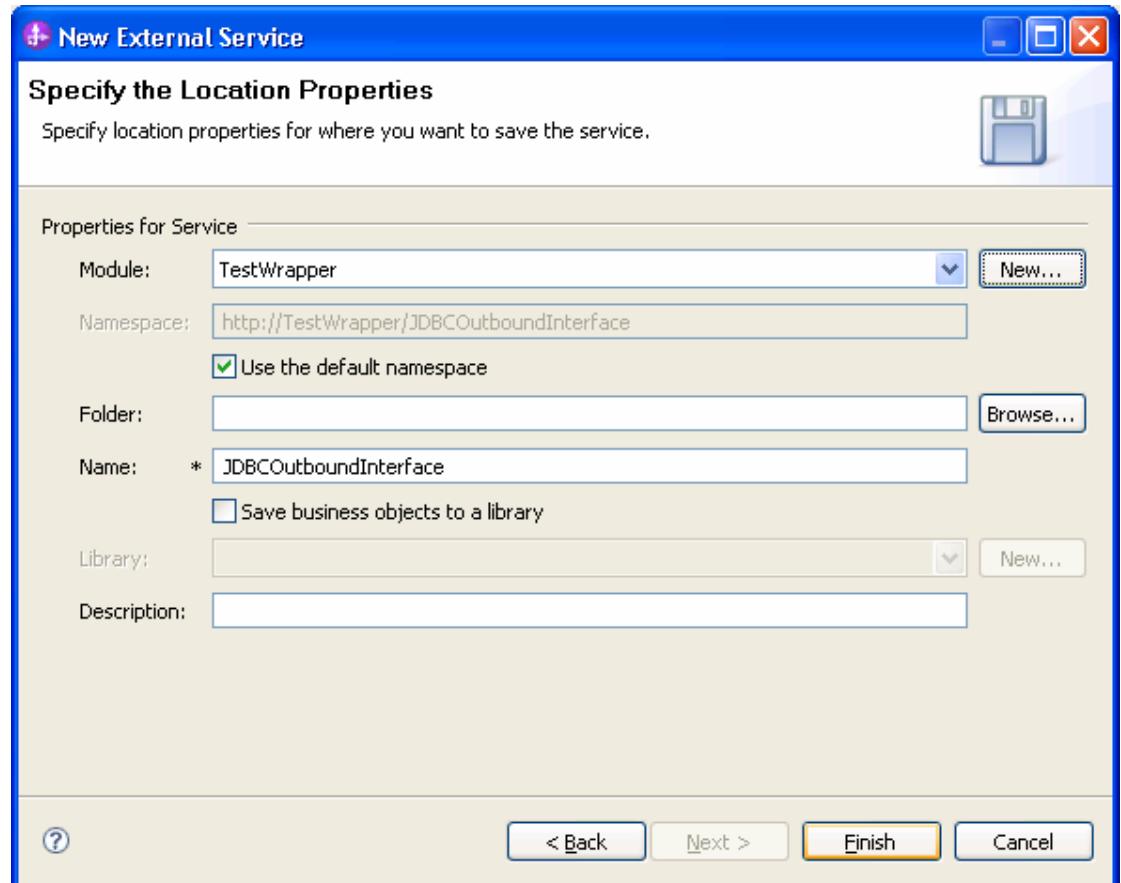
3. In the Select a Business Integration Project Type window, select **Module** and click **Next**.



4. In the Create a Module window, type **TestWrapper** in the **Module Name** field and click **Finish**.

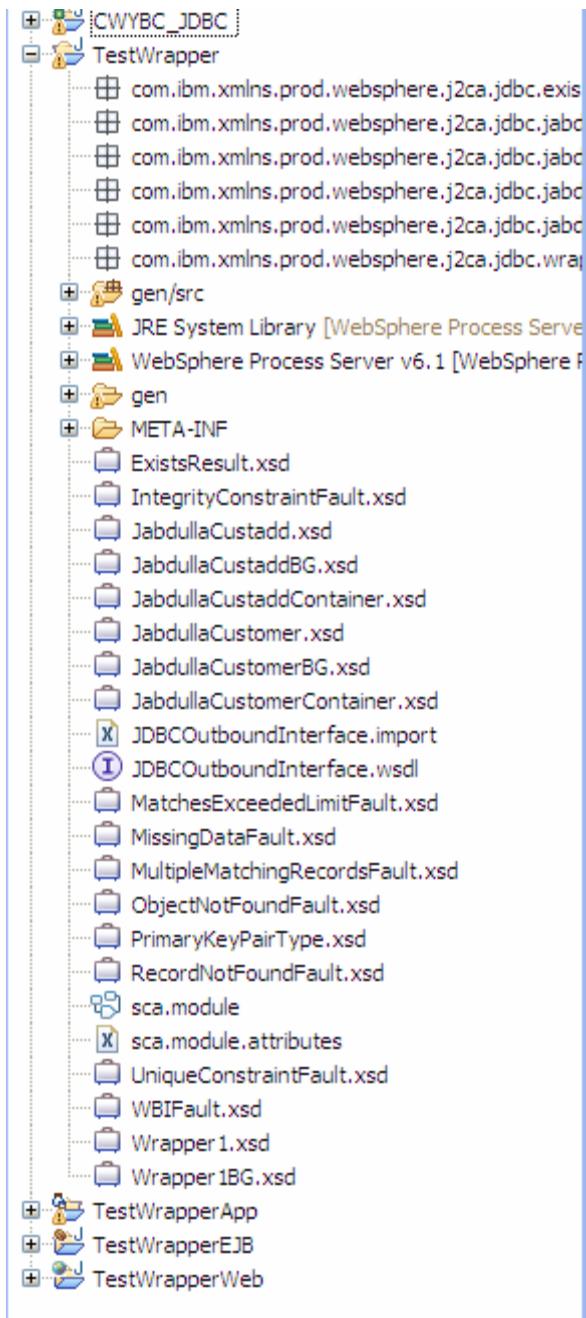


5. In the Specify the Location Properties window, accept the default values for all fields and click **Finish**.



6. Open the Project Explorer and verify business objects are created correctly.

WebSphere software



Deploy the module to the test environment

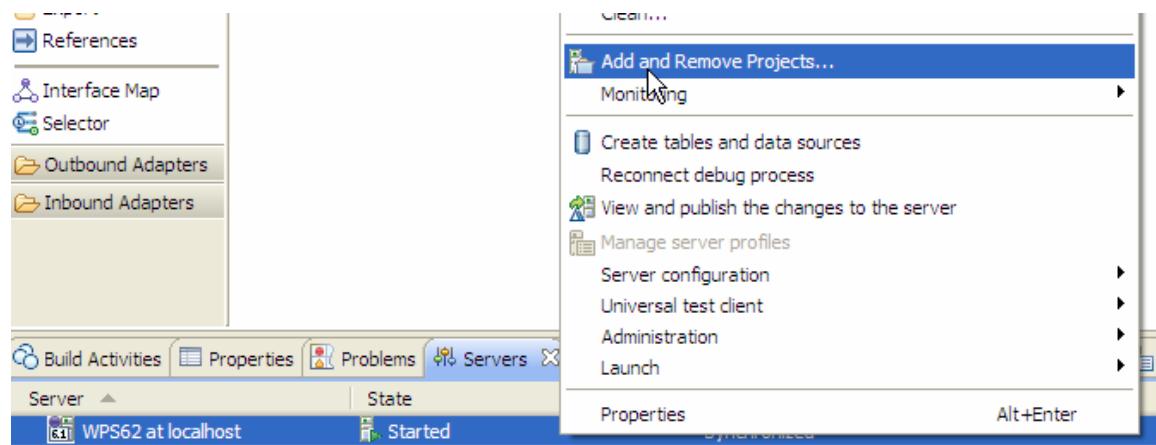
After running the external service wizard, you will have an SCA module that contains an EIS import. You must install this SCA module in the WebSphere Integration Developer integration test client. To do this, you must add the SCA module you created earlier to the server using the **Servers** view in WebSphere Integration Developer.

Steps for adding the SCA module to the server:

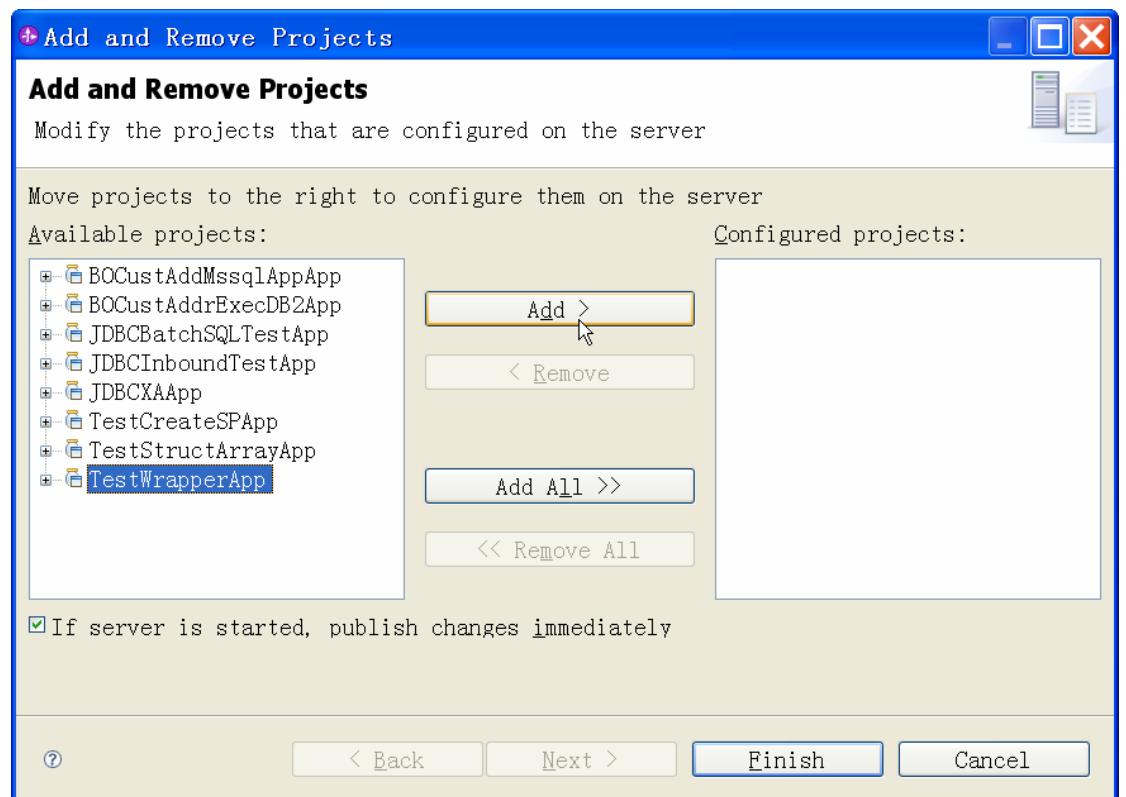
1. In WebSphere Integration Developer, switch to the **Servers** view by selecting from the toolbar **Window > Show View > Servers**.
2. In the **Servers** tab in the lower-right pane right click the server, and select **Start**.

WebSphere software

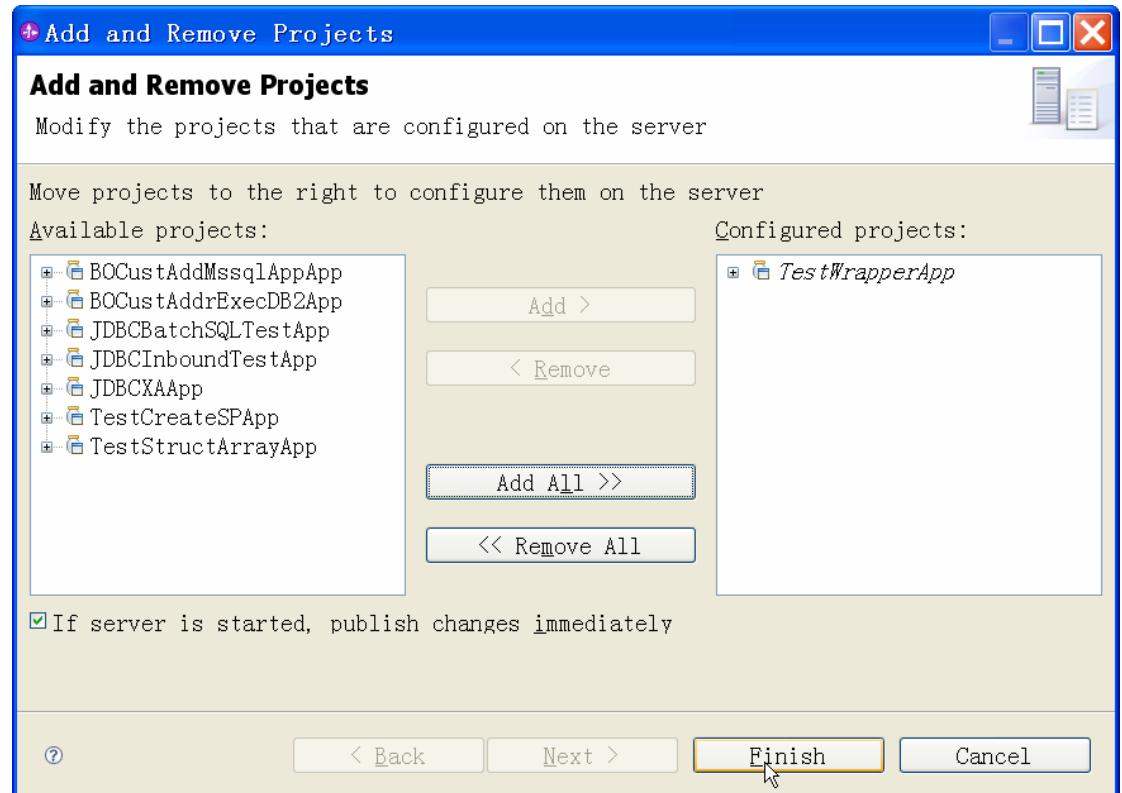
3. After the server is started, right-click the server, and select **Add and Remove projects**.



The Add and Remove Projects window lists the available projects in the WebSphere Integration Developer workspace.



4. Select your project (**TestWrapperApp**) and click **Add** to configure the project on the server and click **Finish**.

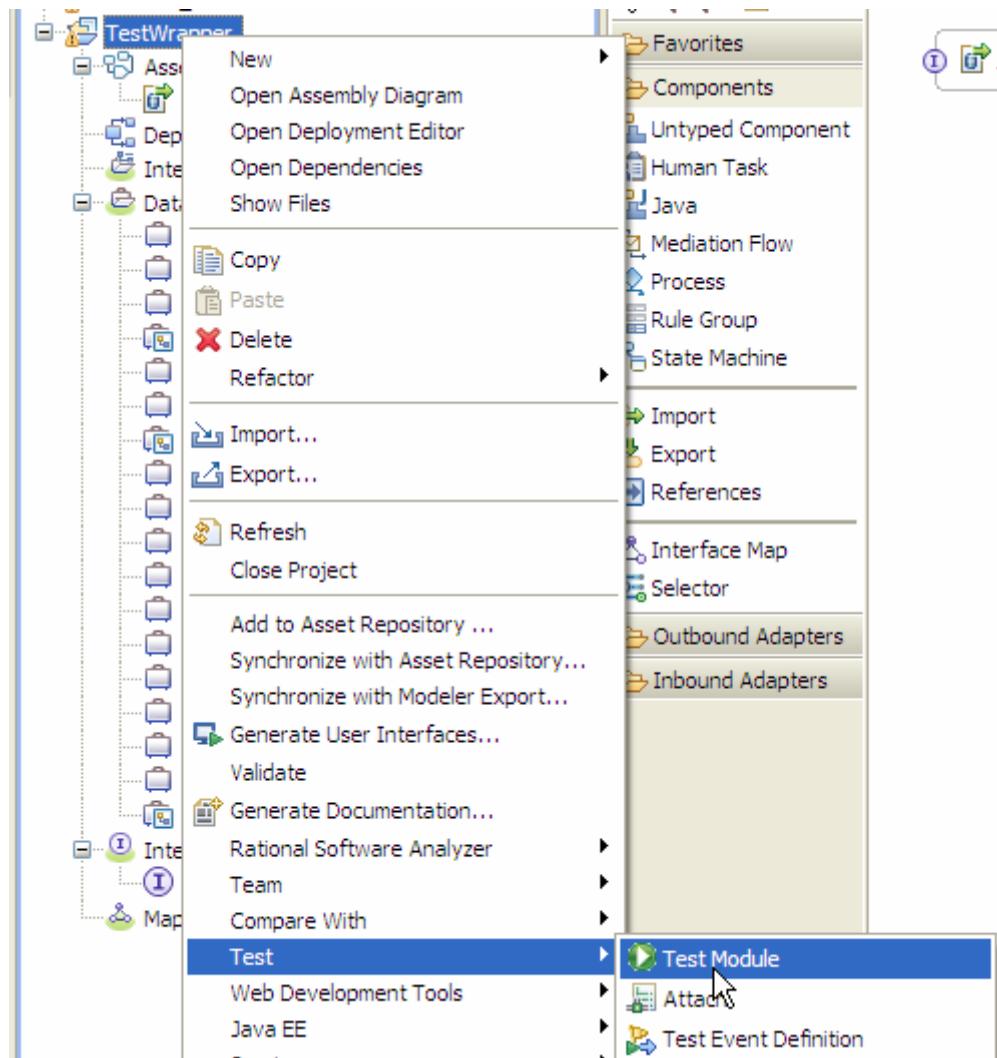


Test the assembled adapter application

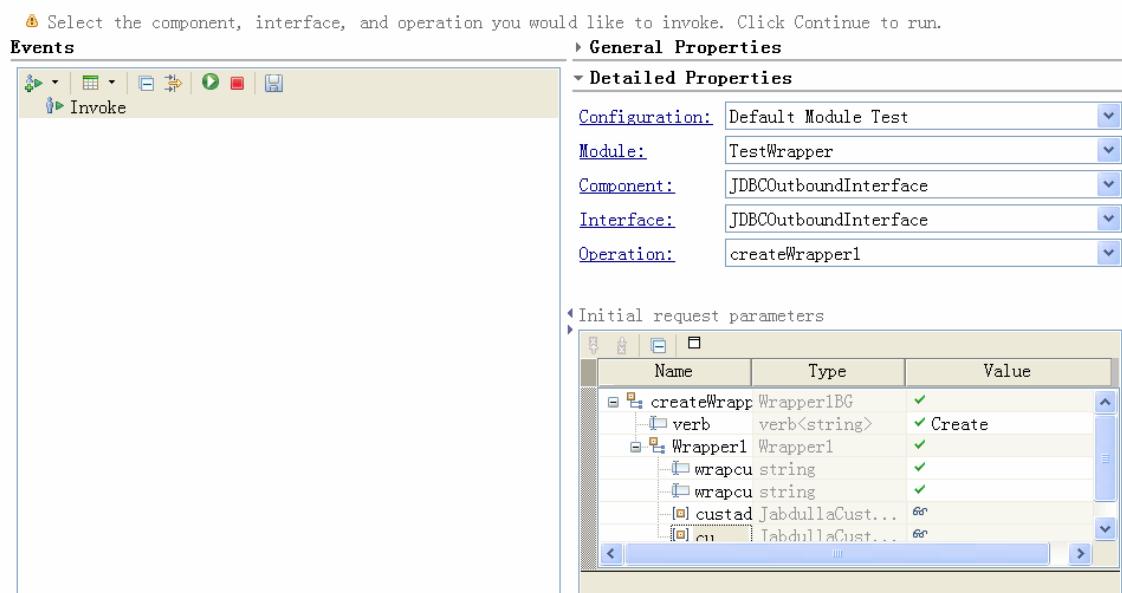
Test the assembled adapter application using the WebSphere Integration Developer integration test client.

1. Select the **TestWrapper** module, right-click, and select **Test > Test Module**. The Test Client window is displayed.

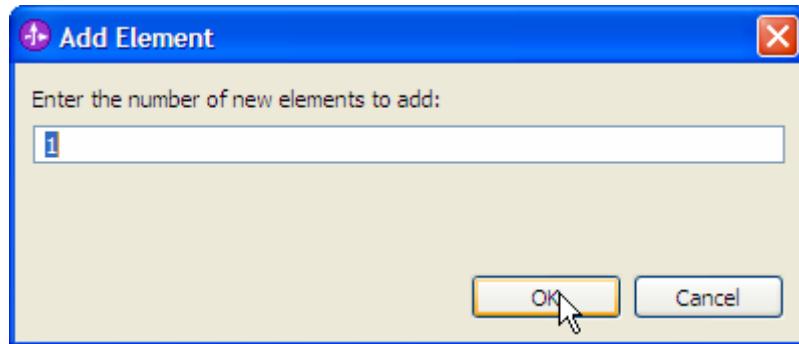
WebSphere software



2. Select **createWrapper1** from the **Operation** list and set "Create" as **verb**. Right-click **custaddobj** and select **Add Elements**.



3. Enter 1 and click **OK**.

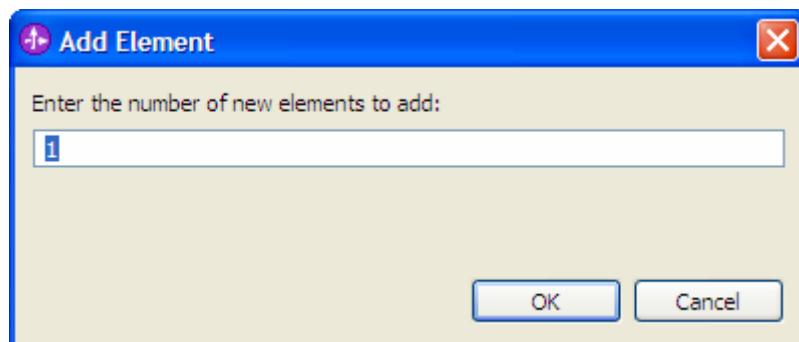


4. Enter the input values for custaddobj[0] as shown in the below figure.

Initial request parameters

	Name	Type	Value
createWrapper1Input	Wrapper1BG		✓
verb	verb<string>		✓ Create
Wrapper1	Wrapper1		✓
wrapcustaddaddrid	string		✓
wrapcustomerpkey	string		✓
custaddobj	JabdullaCustadd[]		∅
custaddobj[0]	JabdullaCustadd		✓
addrid	string		✓ 100
custid	string		✓ 100
city	string		✓ Beijing
zipcode	string		✓ 100000
customerobj	JabdullaCustomer[]		∅

5. Now, right-click over customerobj and select **Add Elements** and enter 1 and click **OK**.



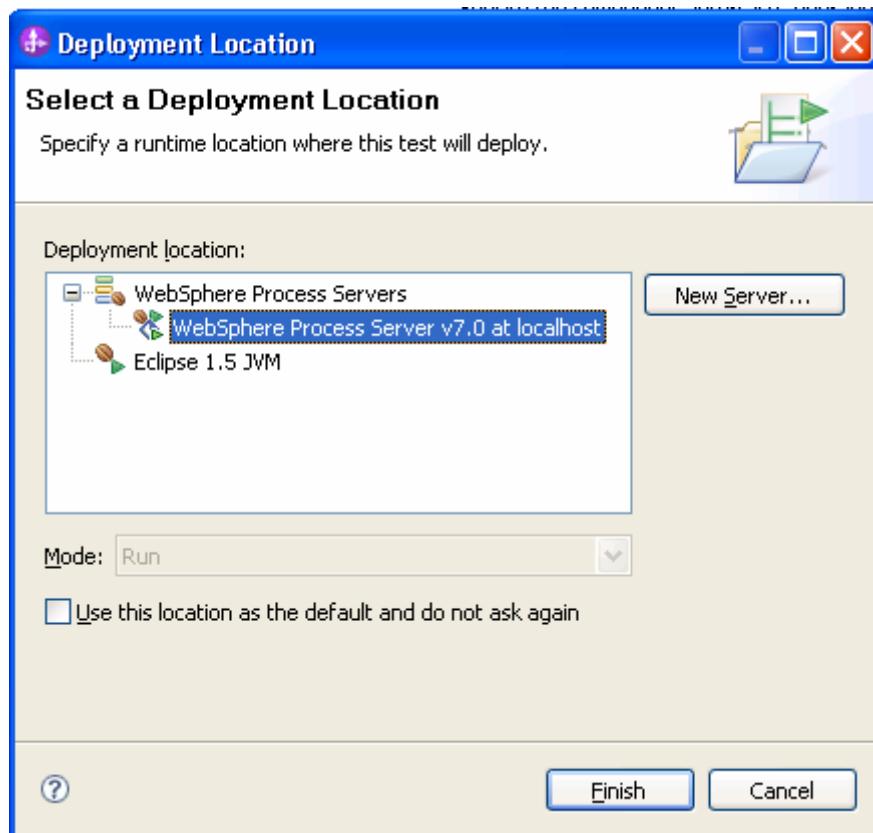
6. Enter the input values for customerobj[0] as shown in the below figure.

WebSphere software

Initial request parameters

Name	Type	Value
createWrapper1Input	Wrapper1BG	✓
verb	verb<string>	✓ Create
Wrapper1	Wrapper1	✓
wrapcustaddaddr	string	✓
wrapcustomerkey	string	✓
custaddobj	JabdullaCustadd[]	✓
custaddobj[0]	JabdullaCustadd	✓
addrid	string	✓ 100
custid	string	✓ 100
city	string	✓ Beijing
zipcode	string	✓ 100000
customerobj	JabdullaCustomer[]	✓
customerobj[0]	JabdullaCustomer	✓
pkey	string	✓ 100
fname	string	✓ IBMer
lname	string	✓ IBMer
ccode	string	✓ IBM

7. To execute the service, click Continue .
8. In the Select Deployment location window, select the server and click **Finish**.



WebSphere software

9. Check the output of the service, and check the data in the enterprise information system (EIS) to ensure it matches the expected values.

Module: [TestWrapper](#)
Component: [IDBCCOutboundInterface](#)
Interface: [IDBCCOutboundInterface](#)
Operation: [createWrapper1](#)

Return parameters:

The screenshot shows a software interface with a table titled "Return parameters". The table has columns for "Name", "Type", and "Value". The data is organized into nested structures. The first row shows a parameter named "createWrapper1(Wrapper1BG" with type "verb<string>" and value "Create". This is followed by several other parameters: "verb" (type "verb<string>", value "Create"), "Wrapper1" (type "Wrapper1", value "✓"), "wrapcustac" (type "string", value "✓"), "wrapcustom" (type "string", value "✓"), and "custaddobj" (type "JabdullaCustadd[]"). The "custaddobj" row is expanded to show two sub-rows: "custadd" (type "JabdullaCustadd", value "✓") and "customerok" (type "JabdullaCustomer[]"). The "customerok" row is also expanded to show four sub-rows: "customer" (type "JabdullaCustomer", value "✓"), "pkey" (type "string", value "100"), "fname" (type "string", value "IBMer"), "lname" (type "string", value "IBMer"), and "ccode" (type "string", value "IBM"). All values in the table are preceded by a green checkmark.

Name	Type	Value
createWrapper1(Wrapper1BG	verb<string>	✓ Create
verb	verb<string>	✓ Create
Wrapper1	Wrapper1	✓
wrapcustac	string	✓
wrapcustom	string	✓
custaddobj	JabdullaCustadd[]	...
custadd	JabdullaCustadd	✓
addr	string	✓ 100
custi	string	✓ 100
city	string	✓ Beijing
zipco	string	✓ 100000
customerok	JabdullaCustomer[]	...
customer	JabdullaCustomer	✓
pkey	string	✓ 100
fname	string	✓ IBMer
lname	string	✓ IBMer
ccode	string	✓ IBM

Clear the sample content

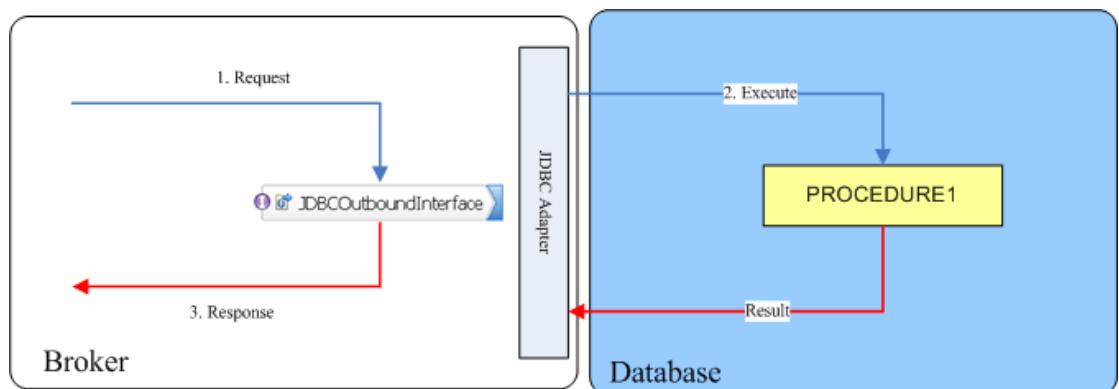
Return the data to its original state.
Nothing is required to clean up after this tutorial.

Chapter 12. Tutorial 11: Creating business objects for stored procedure and executing stored procedure with Execute operation (SQL Server)

This tutorial demonstrates how WebSphere Adapter for JDBC 7.0.0.0 creates business object for stored procedure and execute the stored procedure with execute operation. It also demonstrates the support for result sets returned by stored procedure.

About this task

In this scenario, an application SCA component raises an execute request to the JDBC Outbound Interface. The JDBC adapter generates an execute SQL statement to call the corresponding stored procedure. The stored procedure executes its internal business logic and generates output. Finally, JDBC adapter generates response according to the execution status and the output of the stored procedure. The following figure represents this scenario:



Prepare to run through the tutorial

Extract the sample files

Replicas of the artifacts that you create when using the external service wizard are provided as sample files for your reference. Use these files to verify if the files you create using the external service wizard are correct.

Download the sample zip file and extract it into a directory of your choice (you may want to create a new directory).

Configuration prerequisites

Before configuring the adapter, you must complete the following tasks:

- Create tables and stored procedure
- Create an authentication alias
- Create a data source

Create tables and stored procedure

You must create the following tables and stored procedure in the SQL Server database before starting the scenario.

a. Script for creating the reference types

Execute the below scripts to create CUSTOMER and ADDRESS tables.

```
CREATE TABLE CUSTOMER (
    PKEY VARCHAR(10) NOT NULL PRIMARY KEY,
    FNAME VARCHAR(20) ,
    LNAME VARCHAR(20) ,
    CCODE VARCHAR(10) ) ;
```

```
CREATE TABLE ADDRESS (
    ADDRID VARCHAR(10) NOT NULL PRIMARY KEY,
    CUSTID VARCHAR(10) ,
    CITY VARCHAR(20) ,
    ZIPCODE VARCHAR (10) );
```

Execute the below scripts to enter the following records in the customer table.

```
INSERT INTO CUSTOMER VALUES ('100', 'fname1',
'lname1', 'IBM');
INSERT INTO CUSTOMER VALUES ('200', 'fname2',
'lname2', 'IBM');
```

Execute the below scripts to enter the following records in the address table

WebSphere software

```
INSERT INTO ADDRESS VALUES ('100', '100', 'test1',  
'12345');  
INSERT INTO ADDRESS VALUES ('200', '200', 'test2',  
'12346');
```

b. Script for creating the stored procedure

The stored procedure can be created using the SQL Server Client.

Create a stored procedure that has one input string and one output string parameter, and returns two result sets. Stored Procedures in SQL Server Database always have return value.

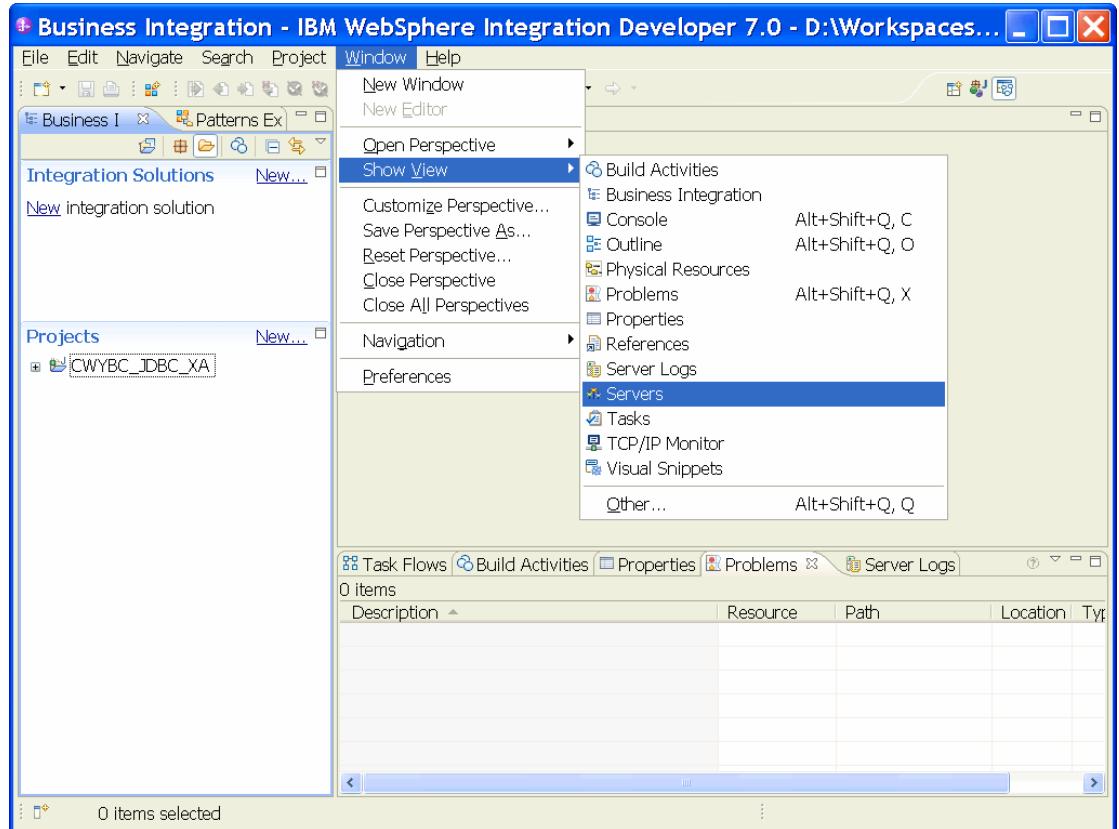
```
CREATE PROCEDURE PROCEDURE1 @var0 varchar(10),  
@var1 varchar(10) OUT  
AS  
    SELECT PKEY,LNAME,FNAME,CCODE FROM CUSTOMER;  
    SELECT ADDRID,CUSTID,CITY,ZIPCODE FROM ADDRESS;  
    Set @var1= @var0;  
    Return (@var1)  
GO
```

Create an authentication alias

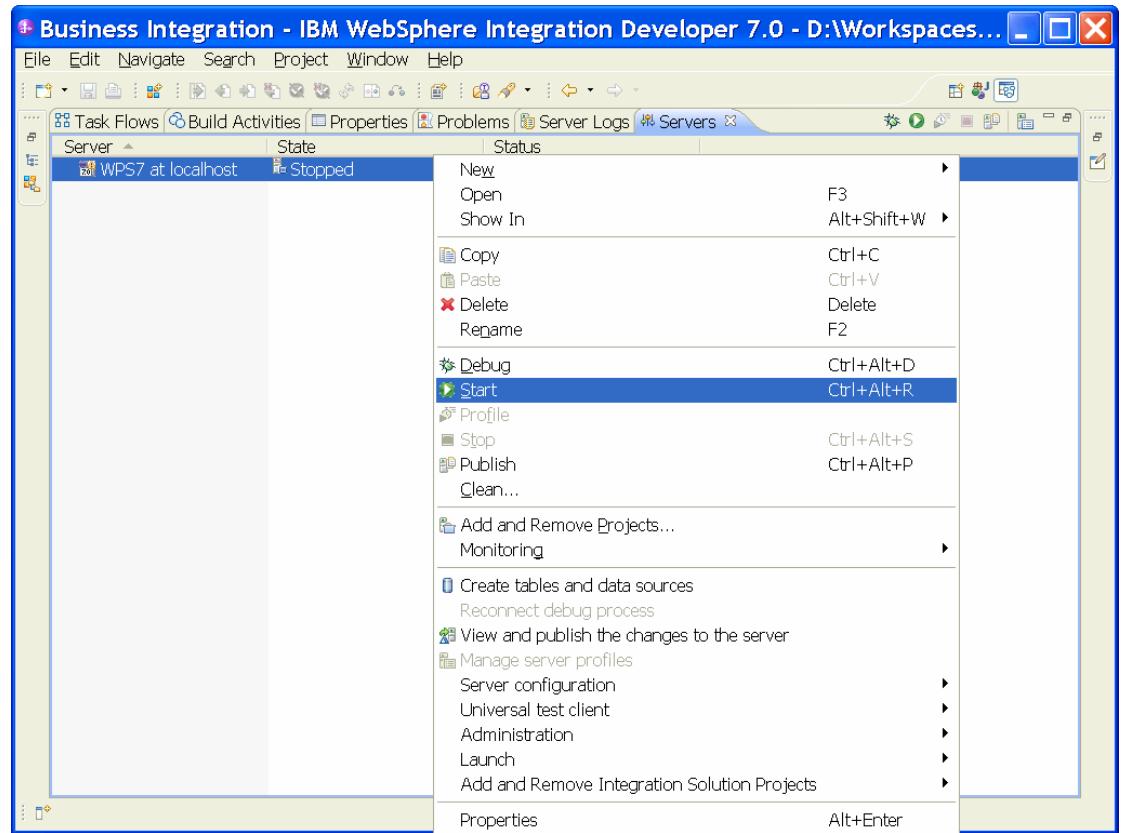
The authentication alias needs to be set because the data source created in the next section uses the username and password set in the authentication alias to connect to the database.

Follow these steps to set the authentication alias in the WebSphere Process Server administrative console.

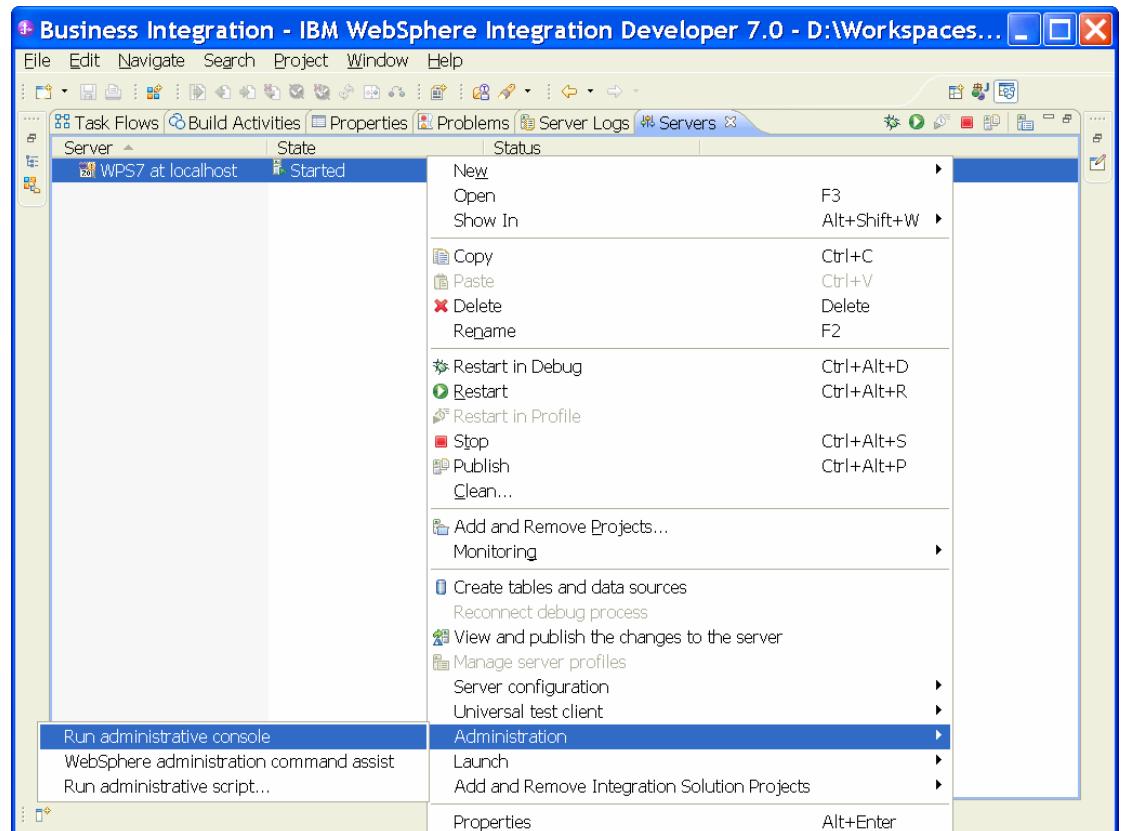
1. In WebSphere Integration Developer, switch to the **Servers** view by selecting **Windows > Show View > Servers**.



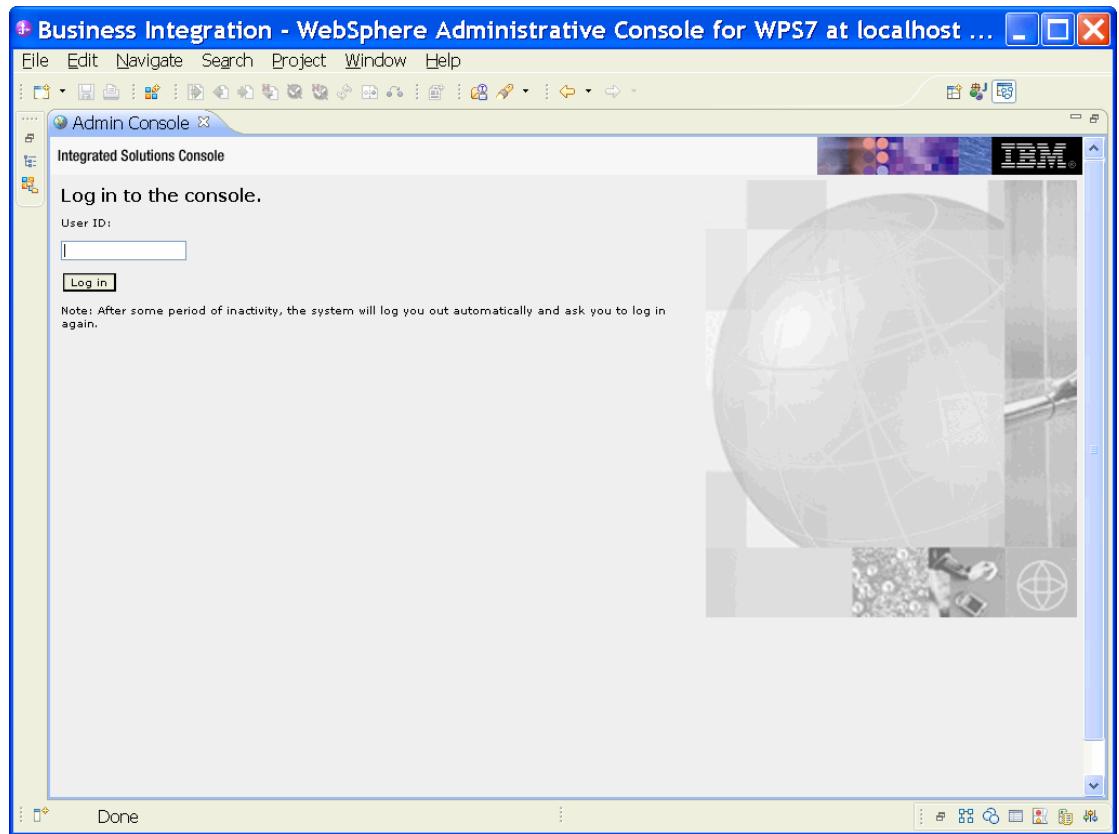
2. In the **Servers** view, right-click the server that you want to start and select **Start**.



3. After the server is started, right-click the server, and select **Administration > Run administrative console**.



4. Log on to the administrative console.



5. Click **Security → Global security**.

WebSphere software

View: All tasks

- Welcome
- [+] Guided Activities
- [+] Servers
- [+] Applications
- [+] Services
- [+] Resources
- [-] Security
 - Business Integration Security
 - Global security
 - Security domains
 - Administrative Authorization Groups
 - SSL certificate and key management
 - Security auditing
 - Bus security
- [+] Environment
- [+] Integration Applications
- [+] System administration
- [+] Users and Groups
- [+] Monitoring and Tuning
- [+] Troubleshooting
- [+] Service integration
- [+] UDDI

6. Under **Java Authentication and Authorization Service**, click **J2C authentication data**.

Cell=localhostNode01Cell, Profile=AppSrv01 Close page

Global security

Global security
Use this panel to configure administration and the default application security policy. This security configuration applies to functions and is used as a default security policy for user applications. Security domains can be defined to override and control applications.

Security Configuration Wizard	Security Configuration Report
<div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> Administrative security <p><input type="checkbox"/> Enable administrative security Administrative user roles <input type="checkbox"/> Administrative group roles <input type="checkbox"/> Administrative authentication</p> </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> Application security <p><input checked="" type="checkbox"/> Enable application security</p> </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> Java 2 security <p><input type="checkbox"/> Use Java 2 security to restrict application access to local resources <input checked="" type="checkbox"/> Warn if applications are granted custom permissions <input type="checkbox"/> Restrict access to resource authentication data</p> </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> User account repository Current realm definition: Federated repositories Available realm definitions: Federated repositories Configure... Set as current </div>	
<div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> Authentication Authentication mechanisms and expiration <input checked="" type="radio"/> LTPA <input type="radio"/> Kerberos and LTPA Kerberos configuration <input type="radio"/> SWAM (deprecated): No authentication Authentication cache settings </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> Web and SIP security </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> RMI/IOP security </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> Java Authentication and Authorization <p><input type="checkbox"/> Application logins <input type="checkbox"/> System logins <input type="checkbox"/> J2C authentication data</p> </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> Security domains </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> External authorization providers </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> Custom properties </div>	

Specifies a list of user identities and passwords for Java(TM) 2 connector security to use.

WebSphere software

A list of existing aliases is displayed.

[Global security > JAAS - J2C authentication data](#)

Specifies a list of user identities and passwords for Java(TM) 2 connector security to use.

Prefix new alias names with the node name of the cell (for compatibility with earlier releases)

Apply

[+] Preferences

New	Delete			
Select	Alias	User ID	Description	
You can administer the following resources:				
<input type="checkbox"/>	BSpace JDBC Alias	wbiuser	Business Space Authorization Alias	
<input type="checkbox"/>	CommonEventInfrastructureJMSAuthAlias	wbiuser	Authentication alias for the Common Event Infrastructure JMS Topics and Queues	
<input type="checkbox"/>	SCA Auth Alias	wbiuser	This is the alias used by SCA to login to a secured SIBus	
<input type="checkbox"/>	localhostNode01Cell/nlNode01/server1/EventAuthDataAliasDerby		Derby authentication alias for the Event Server	
Total 4				

7. Click **New** to create a new authentication entry. Type the alias name, and username and password to connect to the database. Click **OK**.

WebSphere software

Cell=localhostNode01Cell, Profile=AppSrv01

Global security

[Global security > JAAS - J2C authentication data > New](#)

Specifies a list of user identities and passwords for Java(TM) 2 connector security to use.

General Properties

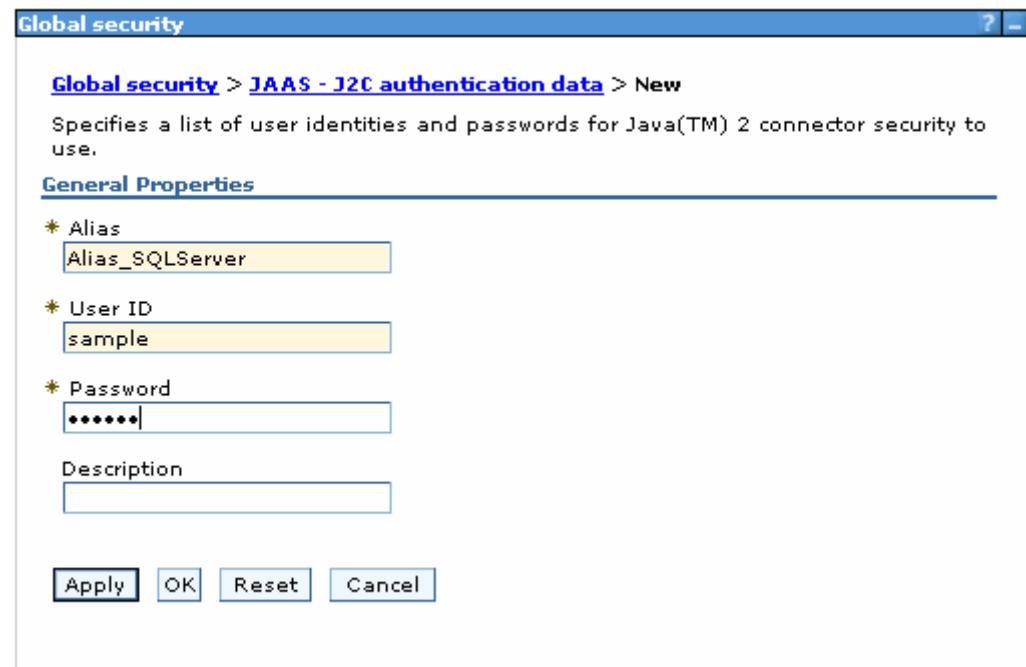
* Alias
Alias_SQLServer

* User ID
sample

* Password

Description

Apply **OK** **Reset** **Cancel**



8. Click **Save** to save the changes.

Cell=localhostNode01Cell, Profile=AppSrv01

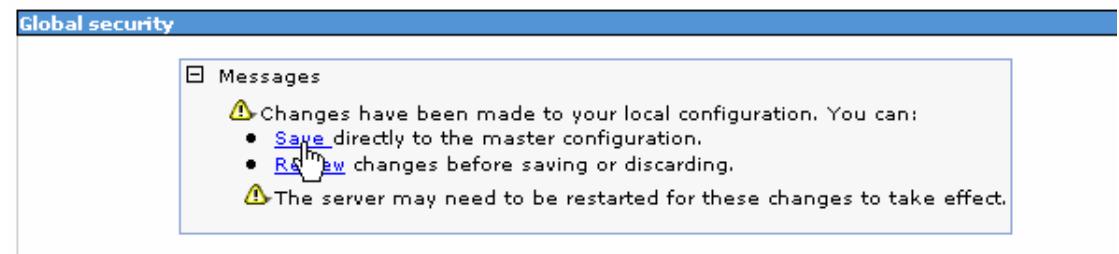
Global security

Messages

Changes have been made to your local configuration. You can:

- [Save](#) directly to the master configuration.
- [Rollback](#) changes before saving or discarding.

The server may need to be restarted for these changes to take effect.



You have created an authentication alias that will be used to configure the data source.

WebSphere software

Preferences

New Delete

Select Alias User ID Description

You can administer the following resources:

<input type="checkbox"/>	BSpace JDBC Alias	wbiuser	Business Space Authorization Alias
<input type="checkbox"/>	CommonEventInfrastructureJMSAuthAlias	wbiuser	Authentication alias for the Common Event Infrastructure JMS Topics and Queues
<input type="checkbox"/>	SCA Auth Alias	wbiuser	This is the alias used by SCA to login to a secured SIBus
<input type="checkbox"/>	localhostNode01Cell/n1Node01/server1/EventAuthDataAliasDerby		Derby authentication alias for the Event Server
<input type="checkbox"/>	n1Node01/Alias_Oracle	sample	
<input type="checkbox"/>	n1Node01/Alias_SQLServer	sample	

Total 6

Create a data source

Create a data source in WebSphere Process Server, which the adapter will use to connect to the database. This data source is used later when generating the artifacts for the module.

Note: This tutorial uses SQL Server as the database and the SQL Server JDBC driver sqljdbc.jar.

Here are the steps to create the data source in the WebSphere Process Server administrative console.

1. In the administrative console, select **Environment → WebSphere Variables**.

WebSphere software



2. On the right, select **MICROSOFT_JDBC_DRIVER_PATH** and specify the path of the sqljdbc.jar file in the **Value** field. Click **OK**.

WebSphere software

Cell=localhostNode01Cell, Profile=AppSrv01

WebSphere Variables

[WebSphere Variables](#) > **MICROSOFT_JDBC_DRIVER_PATH**

Use this page to define substitution variables. Variables specify a level of indirection for some system-defined values, such as file system root directories. Variables have a scope level, which is either server, node, cluster, or cell. Values at one scope level can differ from values at other levels. When a variable has conflicting scope values, the more granular scope value overrides values at greater scope levels. Therefore, server variables override node variables, which override cluster variables, which override cell variables.

Configuration

General Properties

* Name: **MICROSOFT_JDBC_DRIVER_F**

Value: **D:\Lib**

Description:
The directory that contains the Microsoft SQL Server JDBC Driver.

Buttons: Apply, OK, Reset, Cancel

3. Click **Save** to save the changes.

WebSphere Variables

Messages

⚠ Changes have been made to your local configuration. You can:
• [Save](#) directly to the master configuration.
• [Review](#) changes before saving or discarding.
⚠ The server may need to be restarted for these changes to take effect.

The variable is added and appears in the list.

WebSphere software

Preferences			
	New	Delete	
Select	Name	Value	Scope
You can administer the following resources:			
<input type="checkbox"/>	MICROSOFT JDBC DRIVER_NATIVEPATH		Node=nINode01
<input type="checkbox"/>	MICROSOFT JDBC DRIVER_PATH	D:\Lib	Node=nINode01
<input type="checkbox"/>	MQ_INSTALL_ROOT	\$(WAS_INSTALL_ROOT)/lib/WMQ	Node=nINode01
<input type="checkbox"/>	ORACLE JDBC DRIVER_PATH	D:\Lib	Node=nINode01
<input type="checkbox"/>	OS400_NATIVE JDBC40_DRIVER_PATH		Node=nINode01
<input type="checkbox"/>	OS400_NATIVE JDBC_DRIVER_PATH		Node=nINode01
<input type="checkbox"/>	OS400_TOOLBOX JDBC DRIVER_PATH		Node=nINode01
<input type="checkbox"/>	SCA_BUS_ID	localhostNode01Cell	Cell=localhostNode01Cell
<input type="checkbox"/>	SERVER_LOG_ROOT	\$(LOG_ROOT)/server1	Node=nINode01,Server=serve
<input type="checkbox"/>	SYBASE JDBC DRIVER_PATH		Node=nINode01
<input type="checkbox"/>	UNIVERSAL JDBC DRIVER_PATH	\$(WAS_INSTALL_ROOT)/universalDriver/lib	Node=nINode01

4. Select **Resources → JDBC -> JDBC Providers.**

WebSphere software



5. Click **New** in the JDBC providers window.

Cell=localhostNode01Cell, Profile=AppSrv01

JDBC providers

Use this page to edit properties of a JDBC provider. The JDBC provider object encapsulates the specific JDBC driver implementation class for access to the specific vendor database of your environment. Learn more about this task in a [guided activity](#). A guided activity provides a list of task steps and more general information about the topic.

Scope: Cell=**localhostNode01Cell**, Node=**nINode01**

Scope specifies the level at which the resource definition is visible. For detailed information on what scope is and how it works, [see the scope settings help](#).

Node=nINode01

Preferences

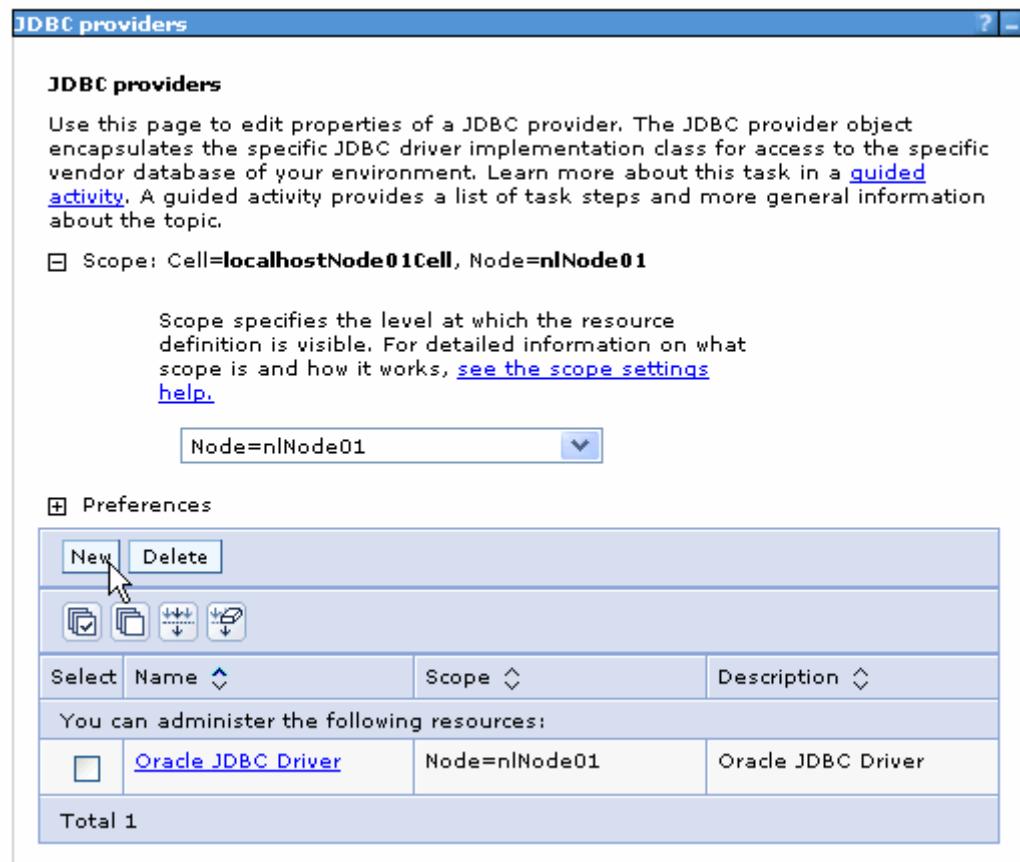
New Delete

Select Name Scope Description

You can administer the following resources:

Select	Name	Scope	Description
<input type="checkbox"/>	Oracle JDBC Driver	Node=nINode01	Oracle JDBC Driver

Total 1



6. Select **SQL Server** database with a connection pool data source for the SQL Server JDBC driver. Click **Next**.

Create a new JDBC Provider

→ Step 1: Create new JDBC provider

Step 2: Enter database class path information

Step 3: Summary

Create new JDBC provider

Set the basic configuration values of a JDBC provider, which encapsulates the specific vendor JDBC driver implementation classes that are required to access the database. The wizard fills in the name and the description fields, but you can type different values.

Scope
cells:localhostNode01Cell:nodes:nNode01

*** Database type**
SQL Server

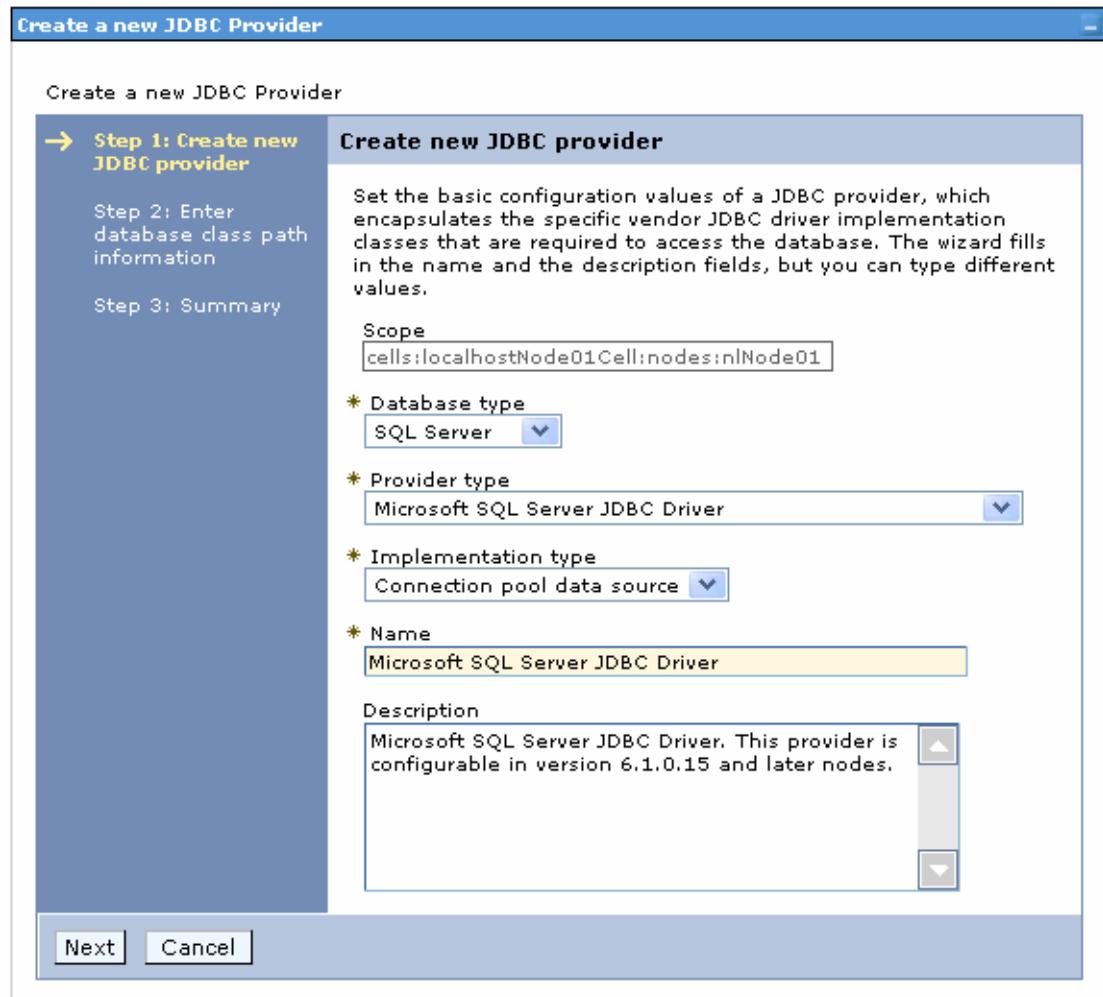
*** Provider type**
Microsoft SQL Server JDBC Driver

*** Implementation type**
Connection pool data source

*** Name**
Microsoft SQL Server JDBC Driver

Description
Microsoft SQL Server JDBC Driver. This provider is configurable in version 6.1.0.15 and later nodes.

Next **Cancel**



7. In the Enter database classpath information page, enter the following value for the **Class path** field:
\$(MICROSOFT_JDBC_DRIVER_PATH)/sqljdbc.jar, where
\$(MICROSOFT_JDBC_DRIVER_PATH) is library path for the run time.
8. Click **Next**.

WebSphere software

Cell=localhostNode01Cell, Profile=AppSrv01

Create a new JDBC Provider

Step 1: Create new JDBC provider

→ Step 2: Enter database class path information

Step 3: Summary

Enter database class path information

Set the environment variables that represent the JDBC driver class files, which WebSphere(R) Application Server uses to define your JDBC provider. This wizard page displays the file names; you supply only the directory locations of the files. Use complete directory paths when you type the JDBC driver file locations. For example: C:\SQLLIB\java on Windows(R) or /home/db2inst1/sqllib/java on Linux(TM).

If a value is specified for you, you may click Next to accept the value.

Class path:

`${MICROSOFT_JDBC_DRIVER_PATH}/sqljdbc.jar`

Directory location for "sqljdbc.jar" which is saved as WebSphere variable `${MICROSOFT_JDBC_DRIVER_PATH}`

`D:\Lib`

Native library path

Directory location which is saved as WebSphere variable `${MICROSOFT_JDBC_DRIVER_NATIVEPATH}`

`${MICROSOFT_JDBC_DRIVER_NATIVEPATH}`

Previous | Next | Cancel

9. In the Summary page, click **Finish**.

Cell=localhostNode01Cell, Profile=AppSrv01

[Close page](#)

Create a new JDBC Provider

Step 1: Create new JDBC provider

Step 2: Enter database class path information

→ Step 3: Summary

Summary

Summary of actions:

Options	Values
Scope	cells:localhostNode01Cell:nodes:n1Node01
JDBC provider name	Microsoft SQL Server JDBC Driver
Description	Microsoft SQL Server JDBC Driver. This provider is configurable in version 6.1.0.15 and later nodes.
Class path	<code> \${MICROSOFT_JDBC_DRIVER_PATH}/sqljdbc.jar</code>
<code> \${MICROSOFT_JDBC_DRIVER_PATH}</code>	<code>D:\Lib</code>
Native path	<code> \${MICROSOFT_JDBC_DRIVER_NATIVEPATH}</code>
<code> \${MICROSOFT_JDBC_DRIVER_NATIVEPATH}</code>	
Implementation class name	<code>com.microsoft.sqlserver.jdbc.SQLServerConnectionPoolDataSource</code>

Previous | Finish | Cancel

10. Click **Save** to save the changes.

WebSphere software

Cell=localhostNode01Cell, Profile=AppSrv01

JDBC providers

Messages

Changes have been made to your local configuration.
You can:

- Save directly to the master configuration.
- Review changes before saving or discarding.

The server may need to be restarted for these changes to take effect.

The JDBC provider is added and appears in the list.

Cell=localhostNode01Cell, Profile=AppSrv01

JDBC providers

Use this page to edit properties of a JDBC provider. The JDBC provider object encapsulates the specific JDBC driver implementation class for access to the specific vendor database of your environment. Learn more about this task in a [guided activity](#). A guided activity provides a list of task steps and more general information about the topic.

Scope: Cell=**localhostNode01Cell**, Node=**nINode01**

Scope specifies the level at which the resource definition is visible. For detailed information on what scope is and how it works, [see the scope settings help](#).

Node=nINode01

Preferences

New	Delete
<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>
Select	Name ▾
	Scope ▾
	Description ▾

You can administer the following resources:

<input type="checkbox"/>	Microsoft SQL Server JDBC Driver	Node=nINode01	Microsoft SQL Server JDBC Driver. This provider is configurable in version 6.1.0.15 and later nodes.
<input type="checkbox"/>	Oracle JDBC Driver	Node=nINode01	Oracle JDBC Driver

Total 2

11. Select the SQL Server JDBC provider you created. Under **Additional Properties**, click **Data sources**. Click **New**.

Cell=localhostNode01Cell, Profile=AppSrv01

JDBC providers

JDBC providers > Microsoft SQL Server JDBC Driver > Data sources

Use this page to edit the settings of a datasource that is associated with your selected JDBC provider. The datasource object supplies your application with connections for accessing the database. Learn more about this task in a [guided activity](#). A guided activity provides a list of task steps and more general information about the topic.

Preferences

New	Delete	Test connection	Manage state...
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Select	Name ▾	JNDI name ▾	Scope ▾
None			Provider ▾
Total 0	Description ▾	Category ▾	

12. Type any value in the **JNDI name** field, and select the authentication alias. Click **Next**.

Cell=localhostNode01Cell, Profile=AppSrv01

Create a data source

Step 1: Enter basic data source information

Step 2: Enter database specific properties for the data source

Step 3: Setup security aliases

Step 4: Summary

Enter basic data source information

Set the basic configuration values of a datasource for association with your JDBC provider. A datasource supplies the physical connections between the application server and the database.

Requirement: Use the Datasources (WebSphere(R) Application Server V4) console pages if your applications are based on the Enterprise JavaBeans(TM) (EJB) 1.0 specification or the Java(TM) Servlet 2.2 specification.

Scope
cells:localhostNode01Cell:nodes:n1Node01

JDBC provider name
Microsoft SQL Server JDBC Driver

* Data source name
Microsoft SQL Server JDBC Driver - DataSource

* JNDI name
SQLServerDS

Next **Cancel**

13. Provide the appropriate **Database name**, **Port number**, **Server name** value. Click **Next**.

Cell=localhostNode01Cell, Profile=AppSrv01

Create a data source

Step 1: Enter basic data source information → Step 2: Enter database specific properties for the data source Step 3: Setup security aliases Step 4: Summary	Enter database specific properties for the data source Set these database-specific properties, which are required by the database vendor JDBC driver to support the connections that are managed through the datasource. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">Name</th> <th style="text-align: left;">Value</th> </tr> </thead> <tbody> <tr> <td>Database name</td> <td>sample</td> </tr> <tr> <td>Port number</td> <td>1433</td> </tr> <tr> <td>Server name</td> <td>9.181.84.136</td> </tr> </tbody> </table> <p><input checked="" type="checkbox"/> Use this data source in container managed persistence (CMP)</p>	Name	Value	Database name	sample	Port number	1433	Server name	9.181.84.136
Name	Value								
Database name	sample								
Port number	1433								
Server name	9.181.84.136								

Previous **Next** **Cancel**

14. Select the authentication alias you just created from the **Component-managed authentication alias** field and click **Next**.

Cell=localhostNode01Cell, Profile=AppSrv01

Create a data source

Step 1: Enter basic data source information Step 2: Enter database specific properties for the data source → Step 3: Setup security aliases Step 4: Summary	Setup security aliases Select the authentication values for this resource. Component-managed authentication alias: nlNode01/Alias_SQLServer Mapping-configuration alias: (none) Container-managed authentication alias: (none) Note: You can create a new J2C authentication alias by accessing one of the following links. Clicking on a link will cancel the wizard and your current wizard selections will be lost. Global J2C authentication alias Security domains
---	---

Previous **Next** **Cancel**

15. In the Summary page, review the values entered for the data source and click **Finish**.

WebSphere software

Cell=localhostNode01Cell, Profile=AppSrv01

Create a data source

Step 1: Enter basic data source information
Step 2: Enter database specific properties for the data source
Step 3: Setup security aliases
→ Step 4: Summary

Summary	
Summary of actions:	
Options	Values
Scope	cells:localhostNode01Cell:nodes:nNode01
Data source name	Microsoft SQL Server JDBC Driver - DataSource
JNDI name	SQLServerDS
Select an existing JDBC provider	Microsoft SQL Server JDBC Driver
Implementation class name	com.microsoft.sqlserver.jdbc.SQLServerConnectionPoolDataSource
Database name	sample
Port number	1433
Server name	9.181.84.136
Use this data source in container managed persistence (CMP)	true
Component-managed authentication alias	nNode01/Alias_SQLServer
Mapping-configuration alias	(none)
Container-managed authentication alias	(none)

Previous | Finish | Cancel

16. Click **Save** to save the changes.

Cell=localhostNode01Cell, Profile=AppSrv01

DATABASE providers

Messages

⚠ Changes have been made to your local configuration. You can:
• [Save](#) directly to the master configuration.
• [Revert](#) changes before saving or discarding.
⚠ The server may need to be restarted for these changes to take effect.

17. Select the check box for the newly created data source and click **Test connection**.

WebSphere software

Cell=localhostNode01Cell, Profile=AppSrv01

JDBC providers

JDBC providers > Microsoft SQL Server JDBC Driver > Data sources

Use this page to edit the settings of a datasource that is associated with your selected JDBC provider. The datasource object supplies your application with connections for accessing the database. Learn more about this task in a [guided activity](#). A guided activity provides a list of task steps and more general information about the topic.

Preferences

New Delete Test connection Manage state...

Select Name JNDI name Scope Provider Description Category

You can administer the following resources:

<input checked="" type="checkbox"/>	Microsoft SQL Server JDBC Driver - DataSource	SQLServerDS	Node=n1Node01	Microsoft SQL Server JDBC Driver	Data source for the Microsoft SQL Server JDBC Driver. This data source type is configurable in version 6.1.0.15 and later nodes.	
-------------------------------------	---	-------------	---------------	----------------------------------	--	--

Total 1

The connection should succeed as indicated by the message shown in the following figure. If you experience problems with the test connection, refer to the "Troubleshooting" section.

Cell=localhostNode01Cell, Profile=AppSrv01

JDBC providers

JDBC providers > Microsoft SQL Server JDBC Driver > Data sources

The test connection operation for data source Microsoft SQL Server JDBC Driver - DataSource on server server1 at node n1Node01 was successful with 6 warning(s). [View JVM log](#) for further details.

Messages

Preferences

New Delete Test connection Manage state...

Select Name JNDI name Scope Provider Description Category

You can administer the following resources:

<input type="checkbox"/>	Microsoft SQL Server JDBC Driver - DataSource	SQLServerDS	Node=n1Node01	Microsoft SQL Server JDBC Driver	Data source for the Microsoft SQL Server JDBC Driver. This data source type is configurable in version 6.1.0.15 and later nodes.	
--------------------------	---	-------------	---------------	----------------------------------	--	--

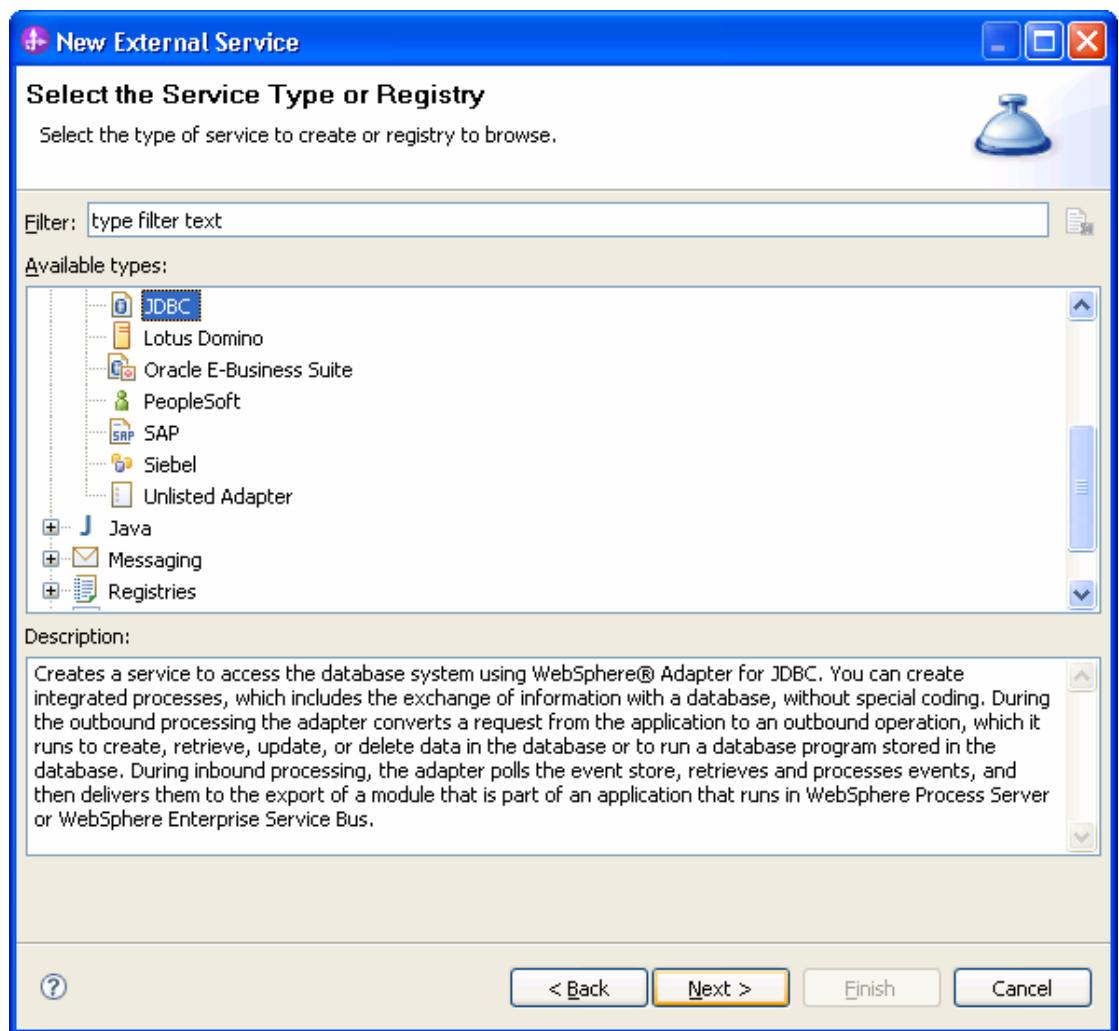
Total 1

Note: The data source is created which will be used by the adapter to connect to the database.

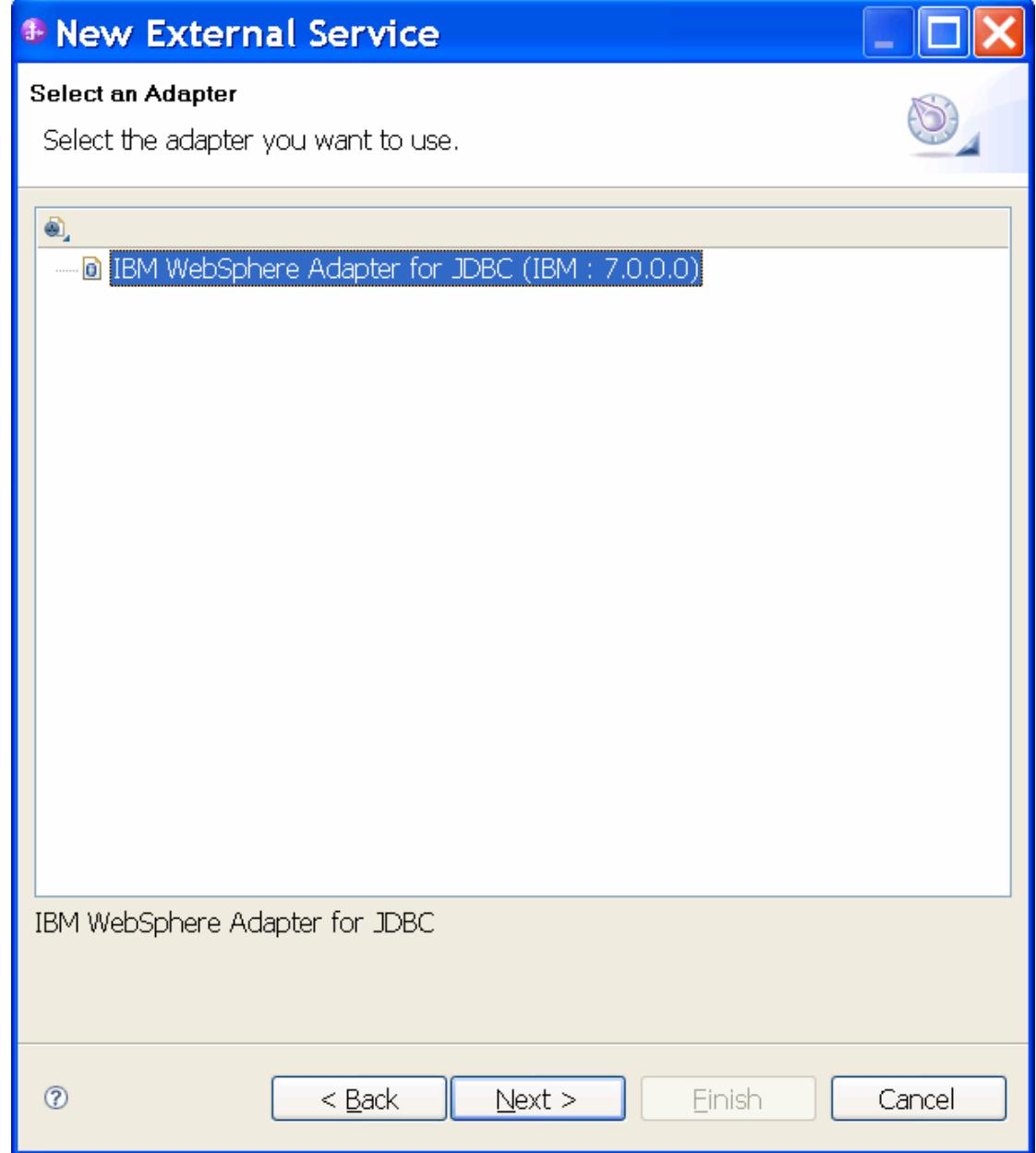
Configure the adapter for outbound processing

Run the external service wizard to specify business objects, services, and configuration details.

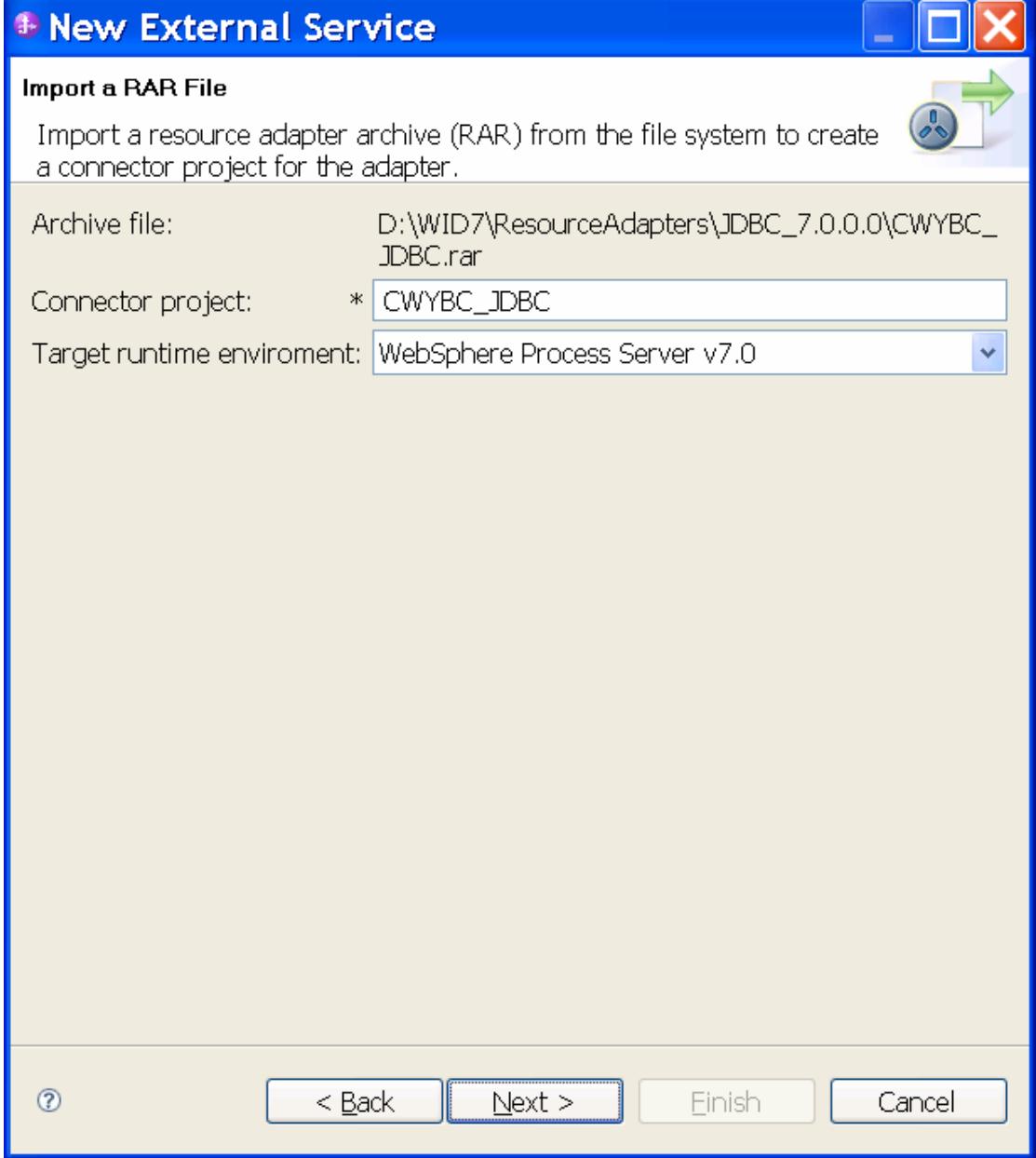
1. Switch to the Business Integration Perspective in WebSphere Integration Developer by selecting **Window -> Open Perspective Business Integration**.
2. Start the external service wizard by selecting **File-> New -> External Service**.
3. In the **Available Types** area, select **Adapters > JDBC** and then click **Next**.



4. Select the **IBM WebSphere Adapter for JDBC (IBM: 7.0.0.0)** and click **Next**.



5. In the **Connector project** field enter **CWYBC_JDBC**, and in the **Target runtime environment** field, select the appropriate runtime. Click **Next**.



New External Service

Import a RAR File

Import a resource adapter archive (RAR) from the file system to create a connector project for the adapter.



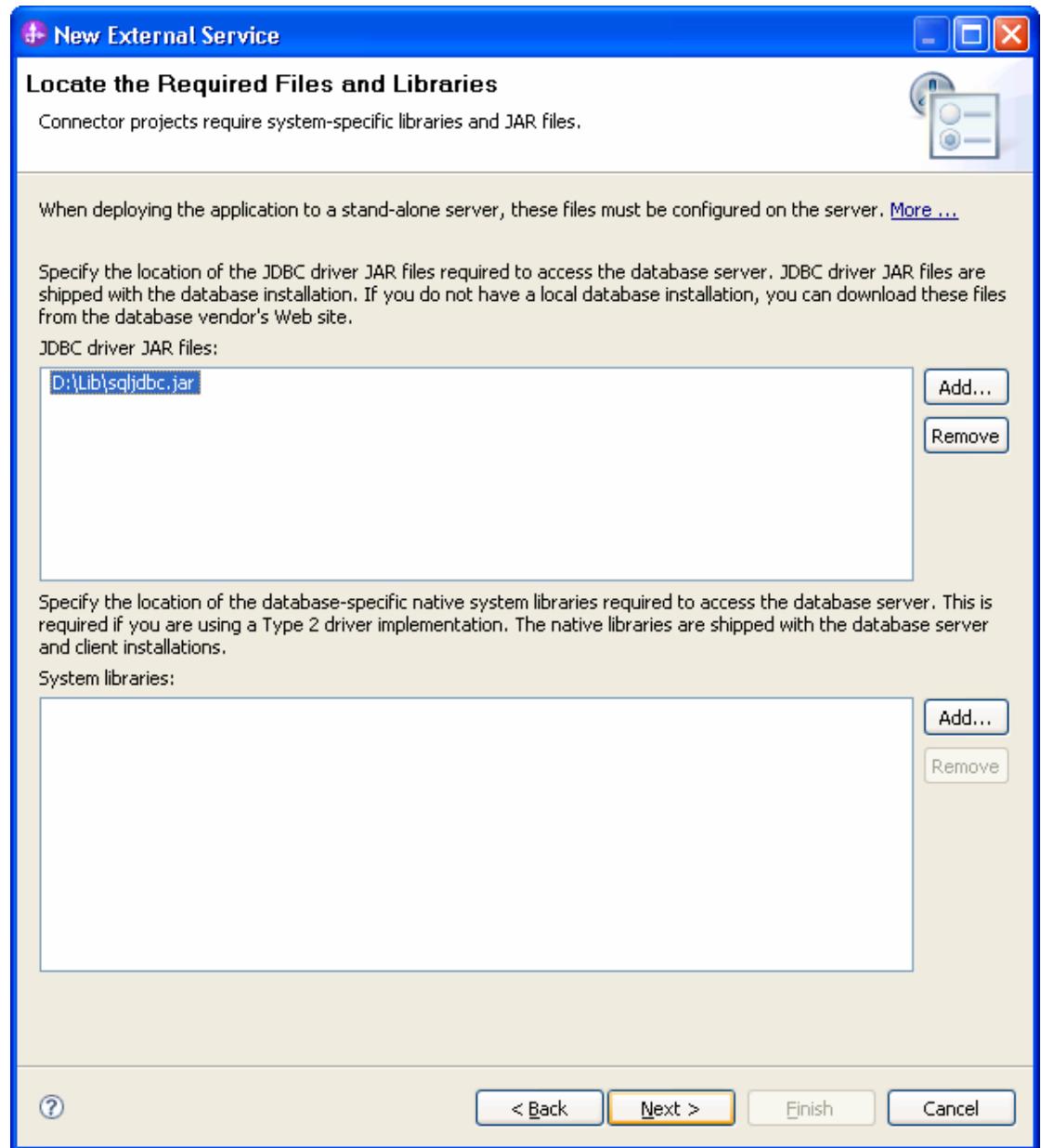
Archive file: D:\WID7\ResourceAdapters\JDBC_7.0.0.0\CWYBC_JDBC.rar

Connector project: * CWYBC_JDBC

Target runtime environment: WebSphere Process Server v7.0

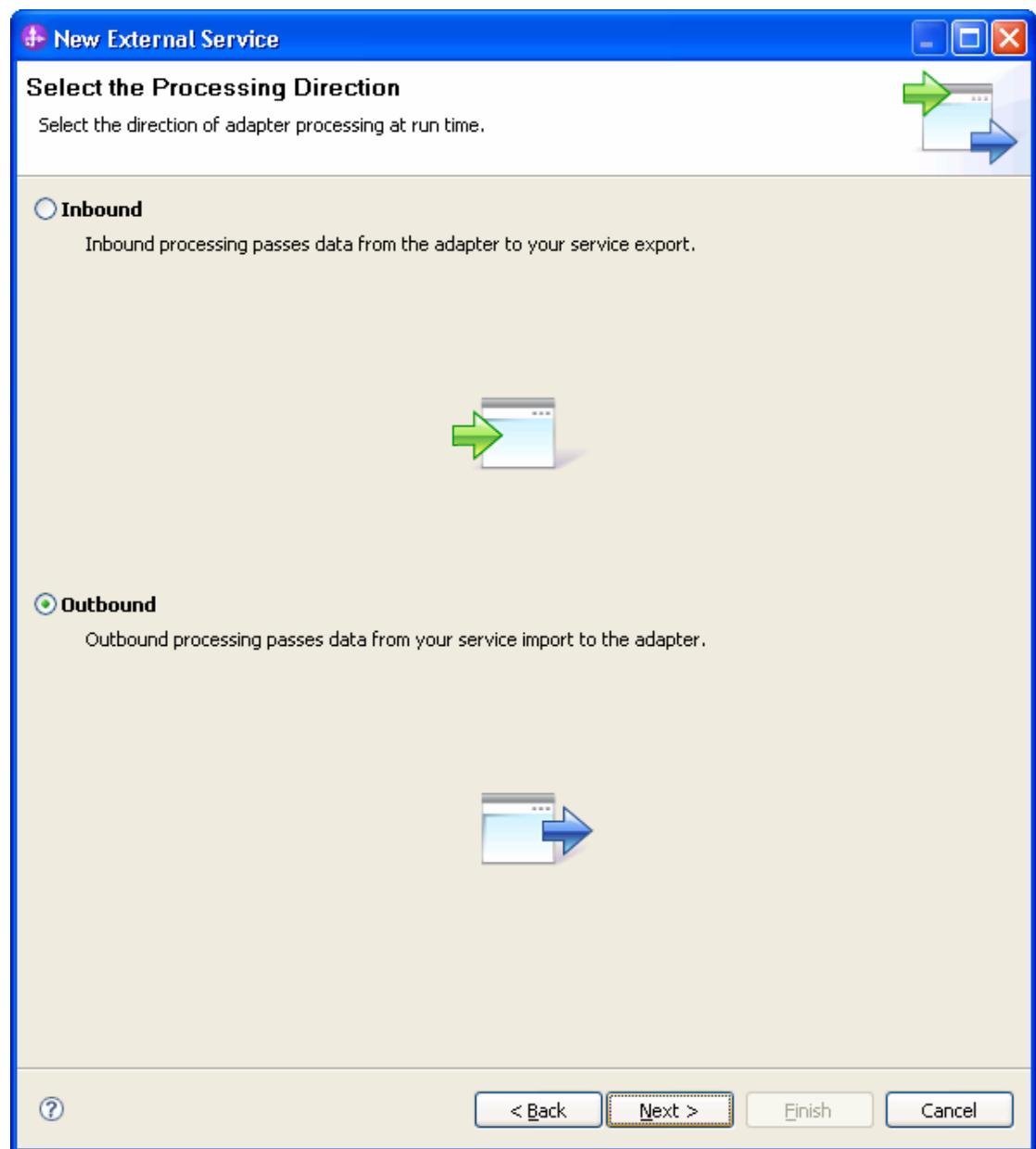
[?](#)[< Back](#)[Next >](#)[Finish](#)[Cancel](#)

6. In the **JDBC driver JAR files** field, click **Add**, to add the JDBC driver class to connect to the database. Browse to select the driver JAR file and click **Next**.



WebSphere software

7. Select **Outbound** and click **Next**.

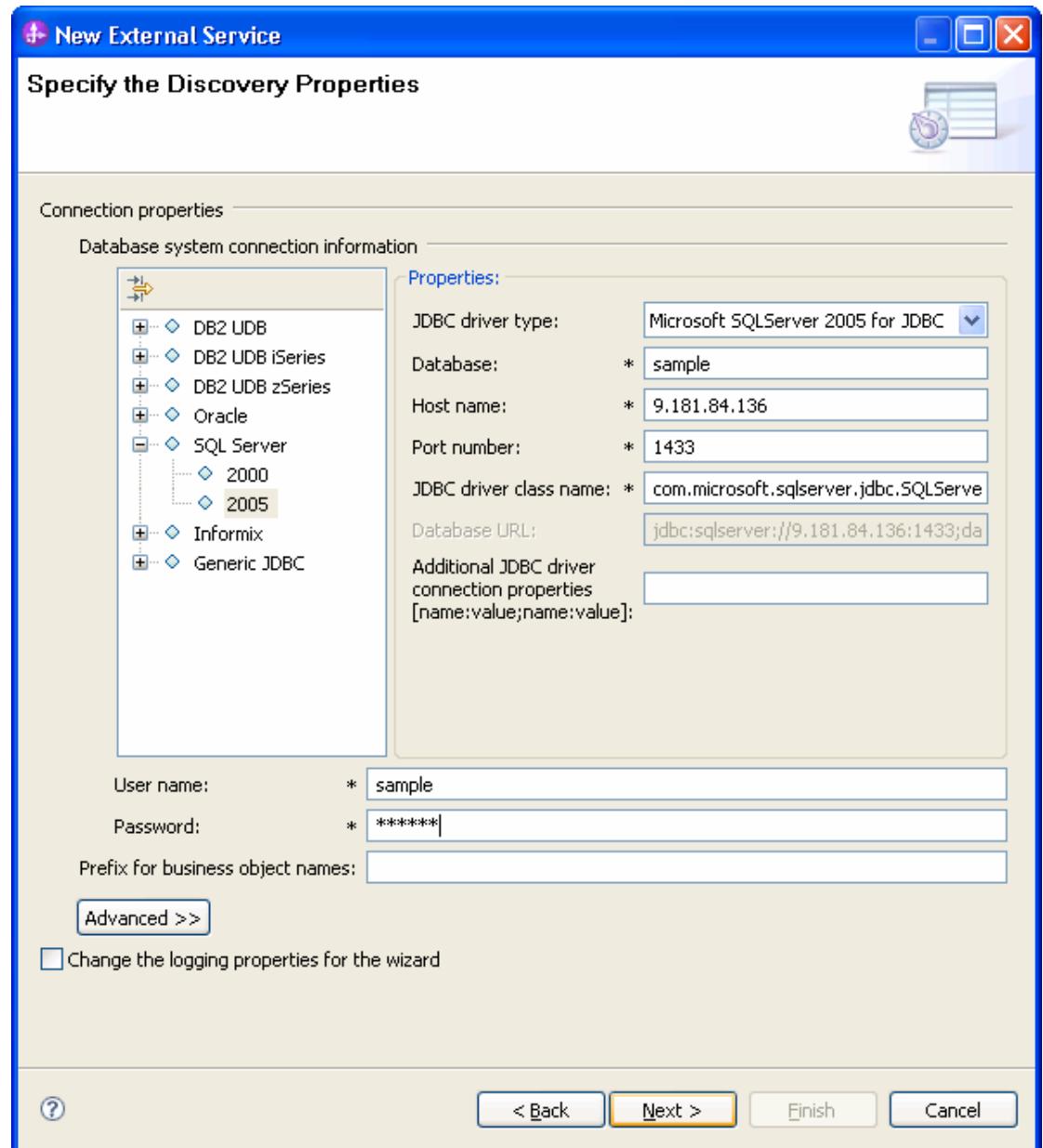


Set connection properties for the external service wizard

To connect to the SQL Server:

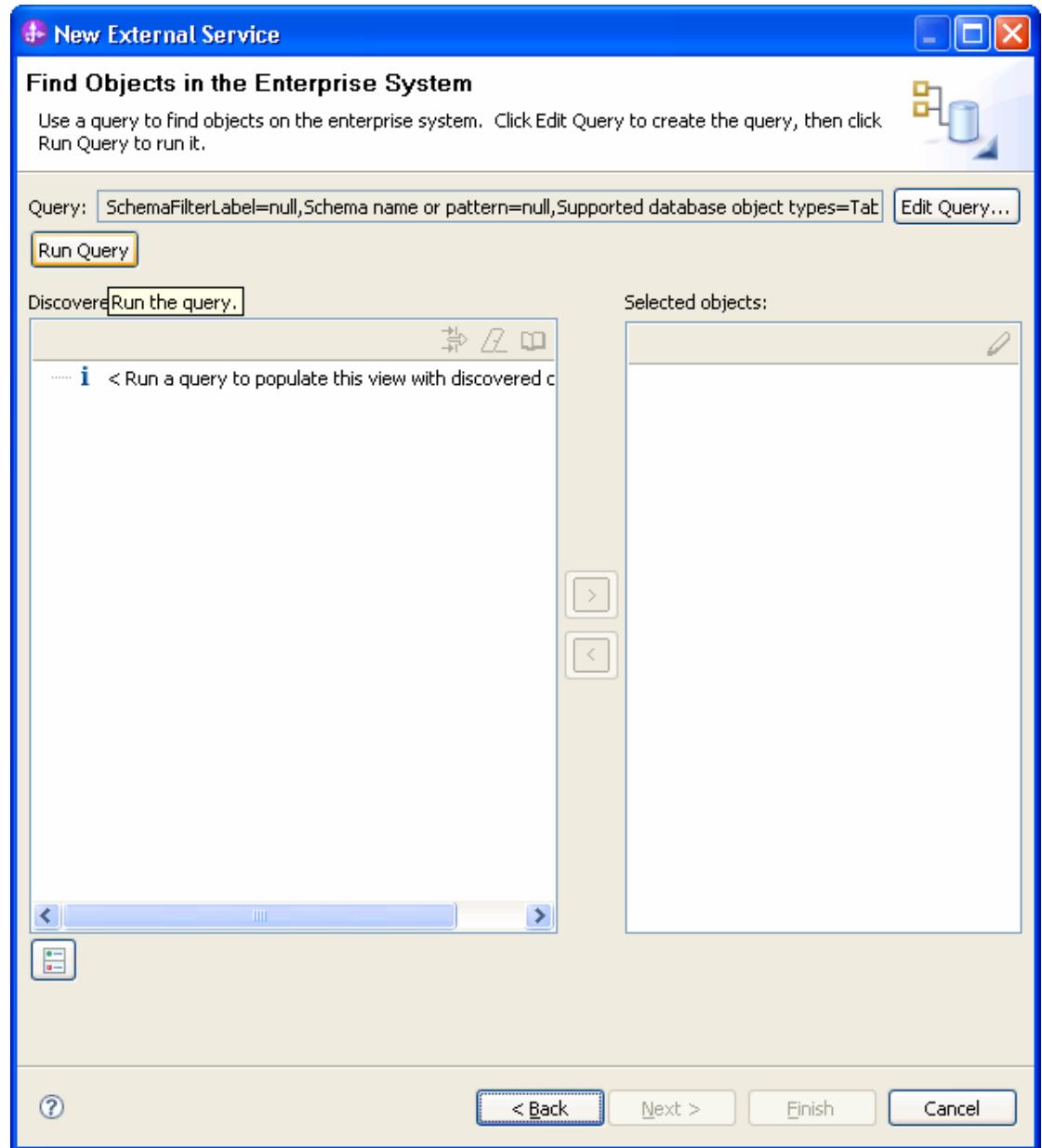
WebSphere software

1. Expand the **SQL Server** node in the Database system connection information area and select 2005.
2. Enter **Database**, **Host name**, **Port number**, **User name** and **Password** fields, and click **Next**.



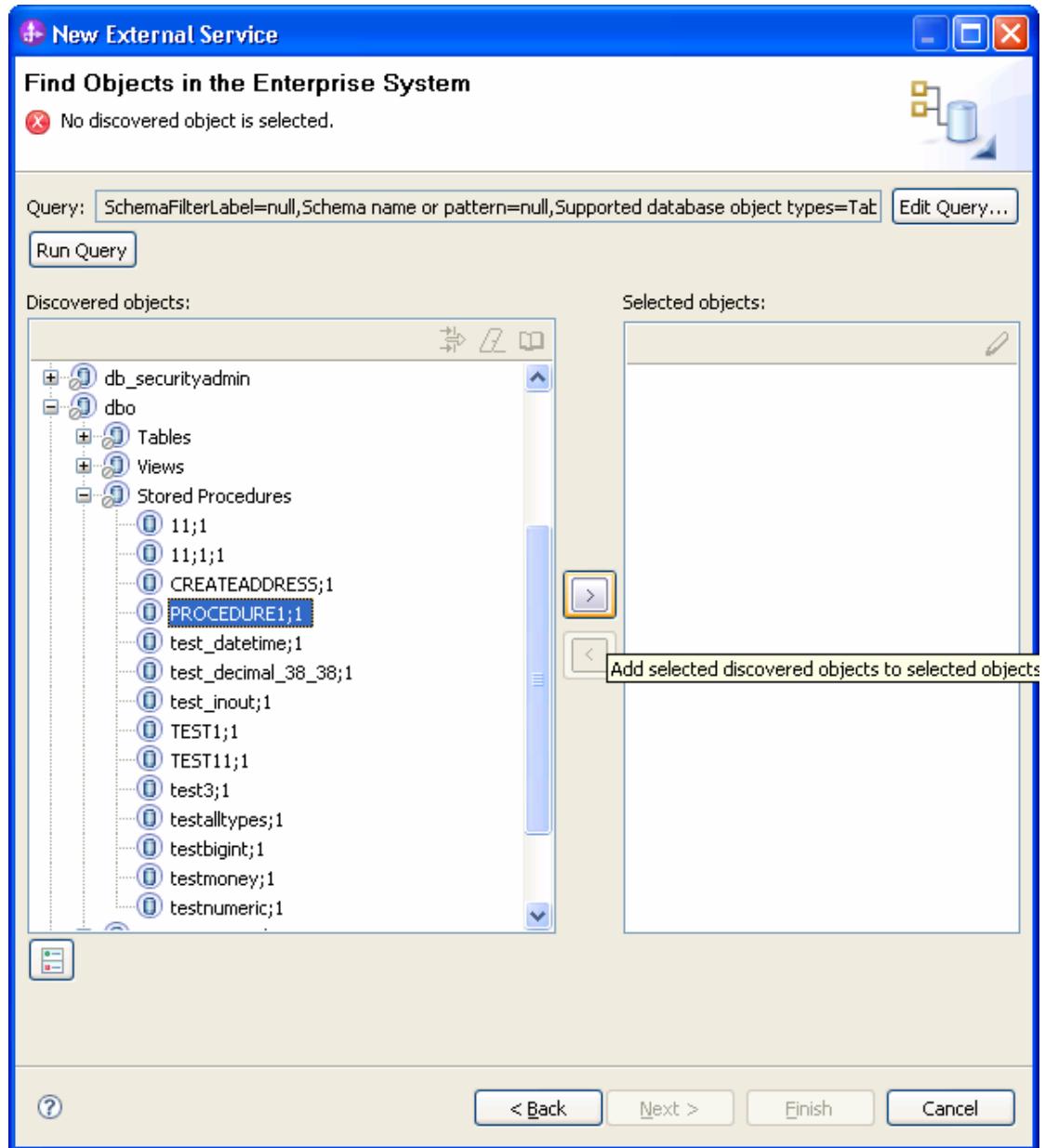
Select the business objects and services to be used with the adapter

1. In the Find Objects in Enterprise System window, click **Run Query**.



2. Expand the **dbo** (for this tutorial only) node and select **Stored Procedures** and expand it.

3. Select **PROCEDURE1;1** and click .



4. Change the maximum number of resultsets to 2 and select String as data type for **@RETURN_VALUE**, **@var0** and **@var1**.

New External Service

Specify the Configuration Properties for 'PROCEDURE1;1'

Generate a business object for the stored procedure

Business object

Stored procedure name: dbo.PROCEDURE1;1

Return value: @RETURN_VALUE

Use ResultSet business object Mode

The maximum number of ResultSets returned from the stored procedure.: 2

Attributes

@RETURN_VALUE

Data type: int

@var0

Data type: string

Sample Value:

@var1

Data type: string

Sample Value:

Returned ResultSets

None

Validate the stored procedure

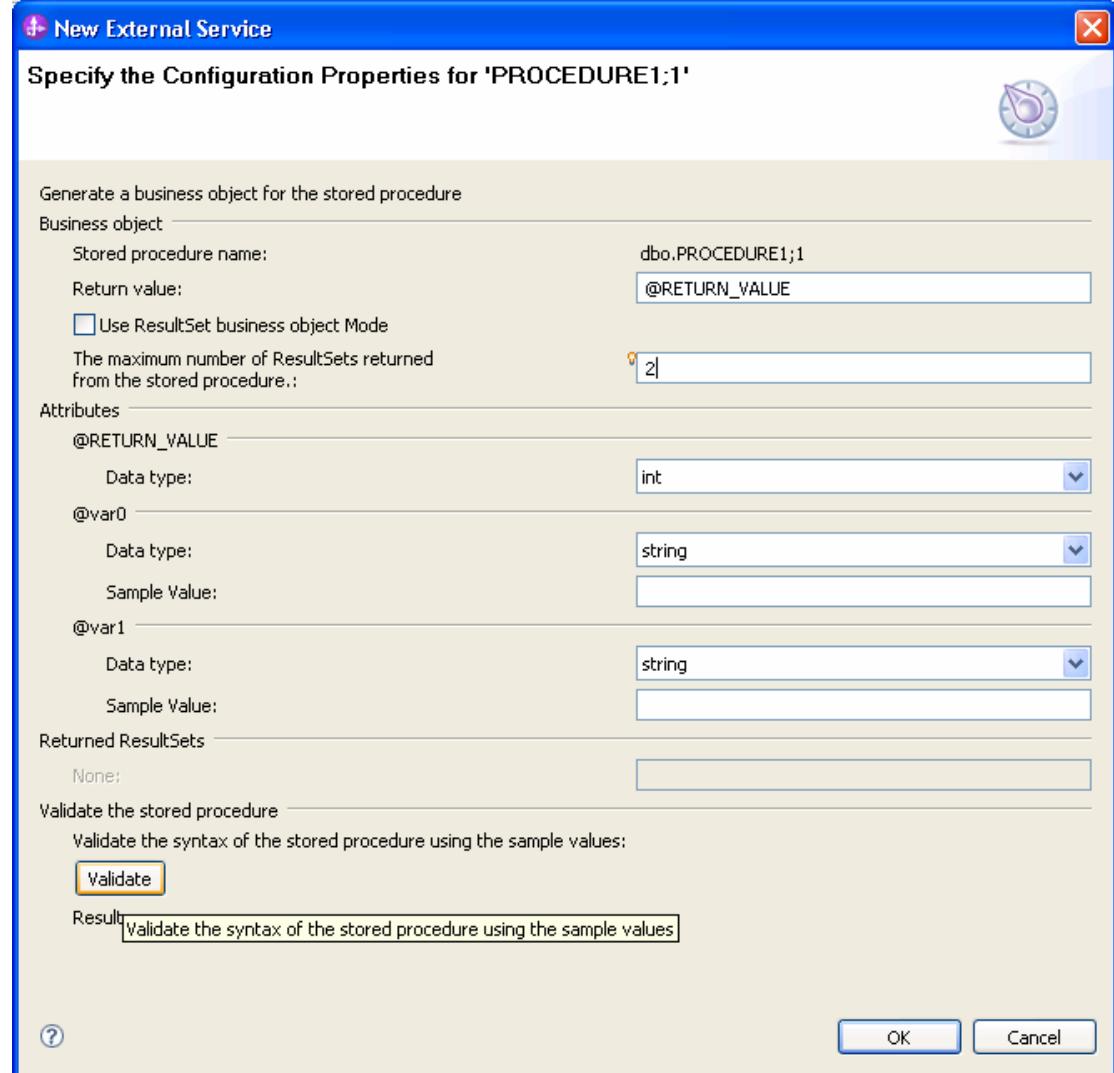
Validate the syntax of the stored procedure using the sample values:

Validate

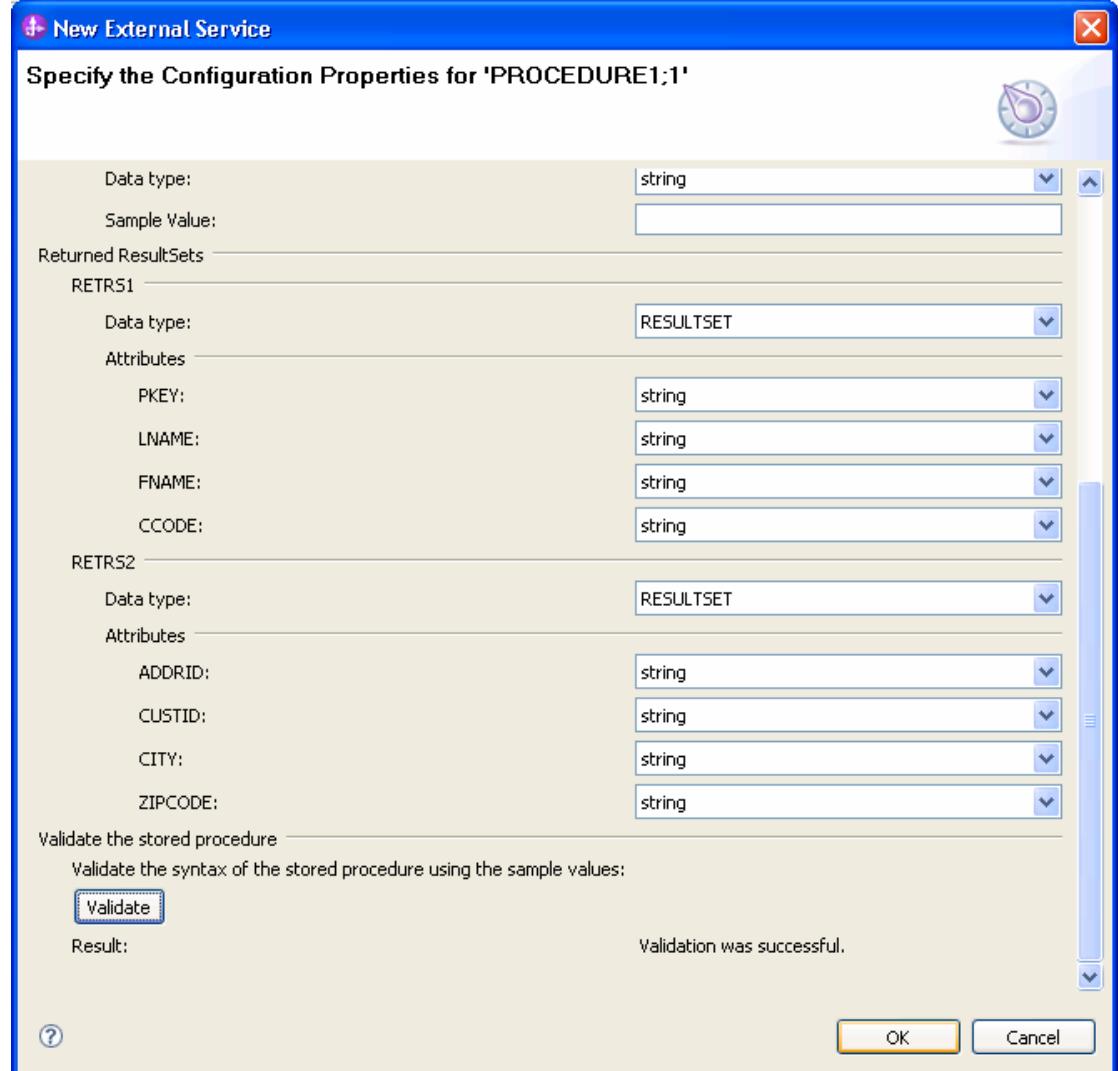
Result: Validate the syntax of the stored procedure using the sample values

?

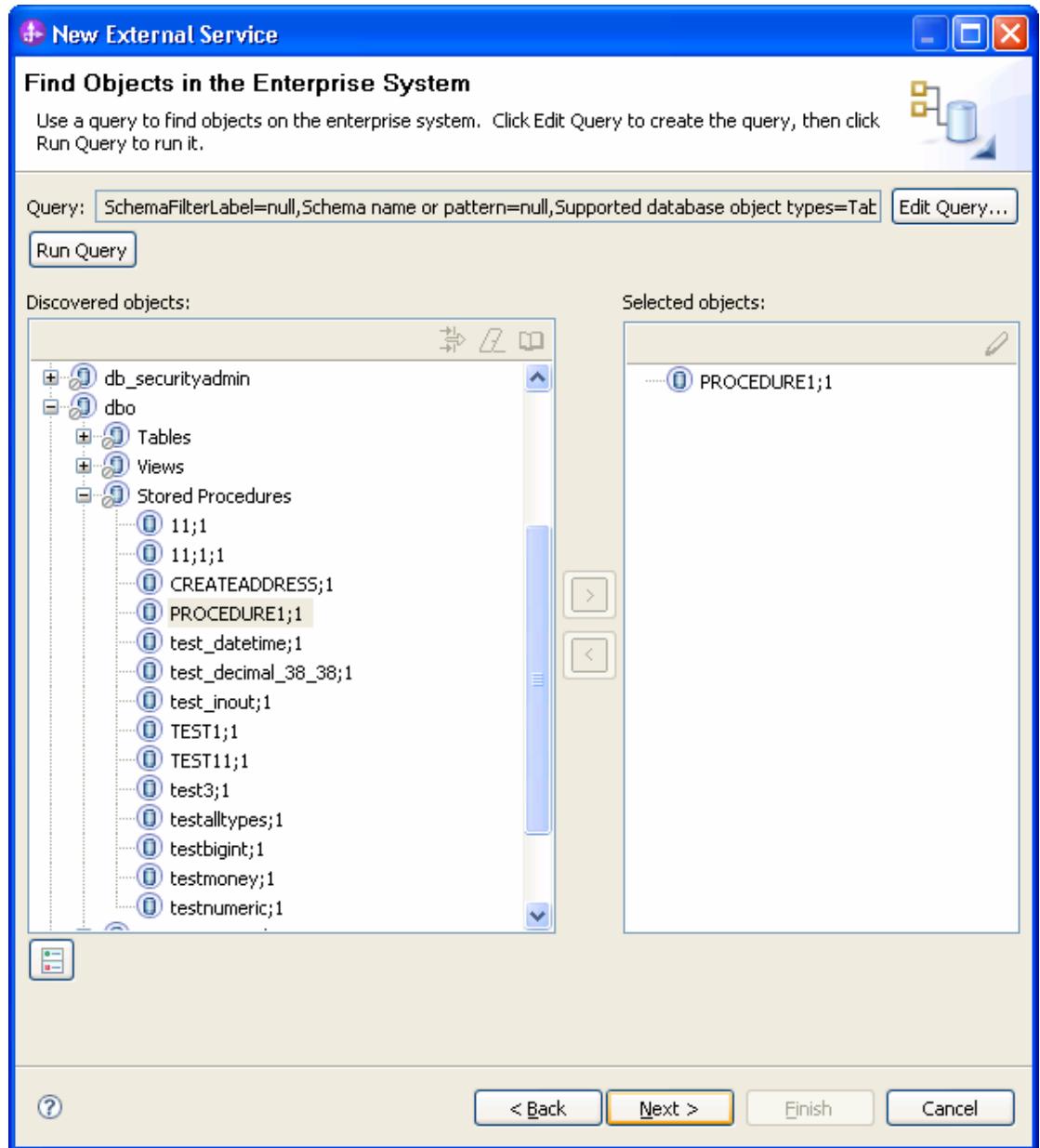
OK Cancel



5. Enter sample values for the stored procedure input types, which are **@var0** and **@var1** and click **Validate** to verify if the stored procedure executes successfully. Verify the result from the validation and click **OK**.



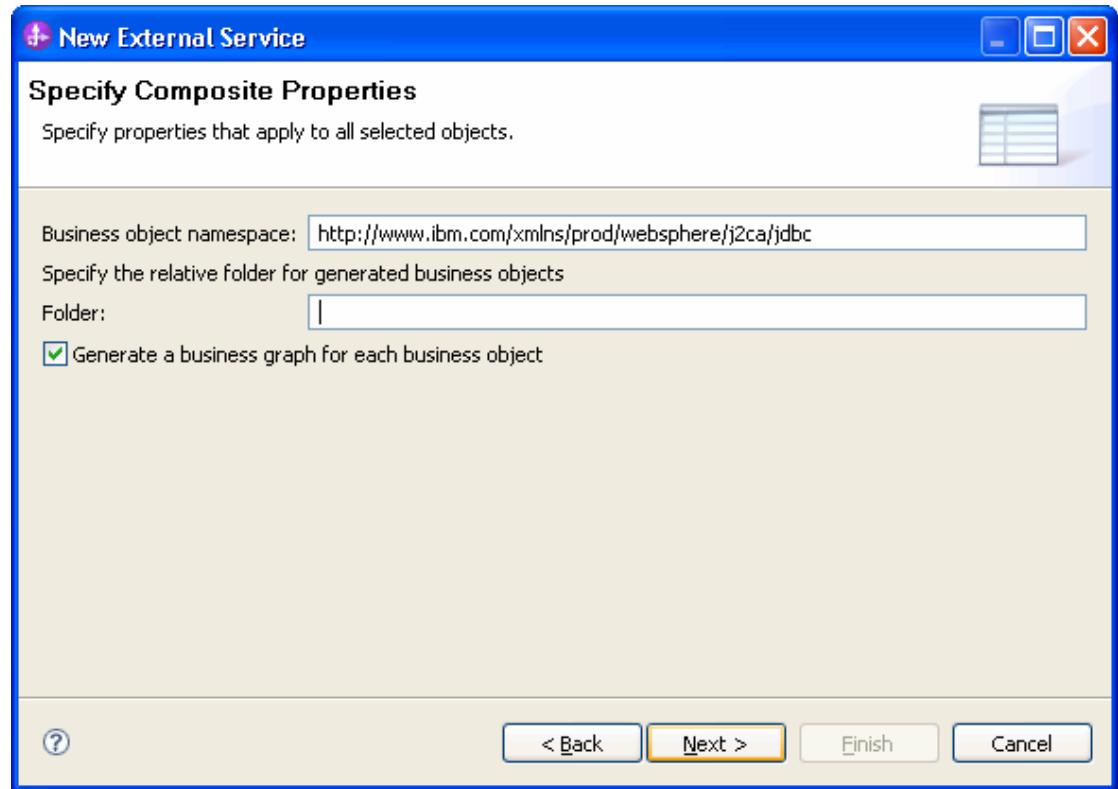
6. Make sure procedure1 is added to the selected objects list and click **Next**.



Generate business object definitions and related artifacts

Follow these steps to generate the business object definitions.

1. In the Specify Composite Properties window, accept the default values for all fields and click **Next**.



2. In the Specify the Service Generation and Deployment Properties window, select **Other** for security options under **Deployment Properties**.
 - a) Clear the **Join the global transaction** check box.
 - b) Select **Specify predefined connection pool DataSource** from the **Database connection information** list.
 - c) Enter **SQLServerDS** in the **Connection pool DataSource JNDI Name** field, and click **Next**.

New External Service

Specify the Service Generation and Deployment Properties

Specify properties for generating the service and running it on the server.

Service Operations
To modify the names, or add a description to the operations to be generated in the interface file, click Edit Operations...
[Edit Operations...](#)

Deployment Properties

How do you want to specify the security credentials?

Using an existing JAAS alias (recommended)
A Java Authentication and Authorization Services (JAAS) alias is the preferred method.
J2C authentication data entry:

Using security properties from the managed connection factory
The properties will be stored as plain text; no encryption is used.
User name:
Password:

Other
Use if no security is required or will be handled by the EIS system, or the RAR will be deployed on the server and security will be specified by the properties in the JNDI lookup name.
The quality of service that is used to join the transaction provides a higher degree of data integrity, especially when a failure occurs. To participate in a global transaction, a predefined XA DataSource or XA database connection information must be specified in the connection properties. [More ...](#)
 Join the global transaction

Deploy connector project:

Specify the settings used to connect to JDBC at run time:

Connection settings:

Connection Properties

To join a global transaction, specify a predefined XA datasource or XA database connection information. When not joining a global transaction, either the XA connection information or the local connection information can be specified.

Database connection information:

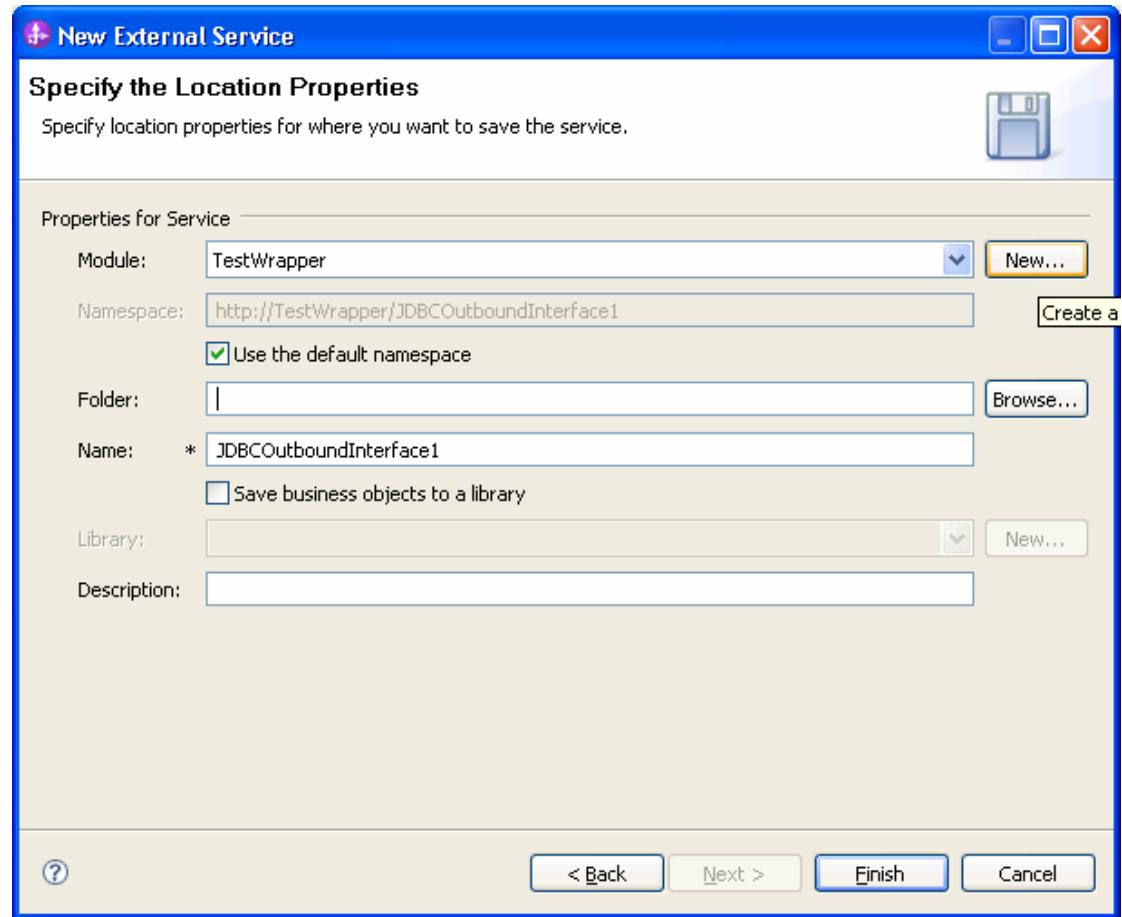
Database system connection information

Database vendor: MSSQLSERVER
Connection pool DataSource JNDI name: *

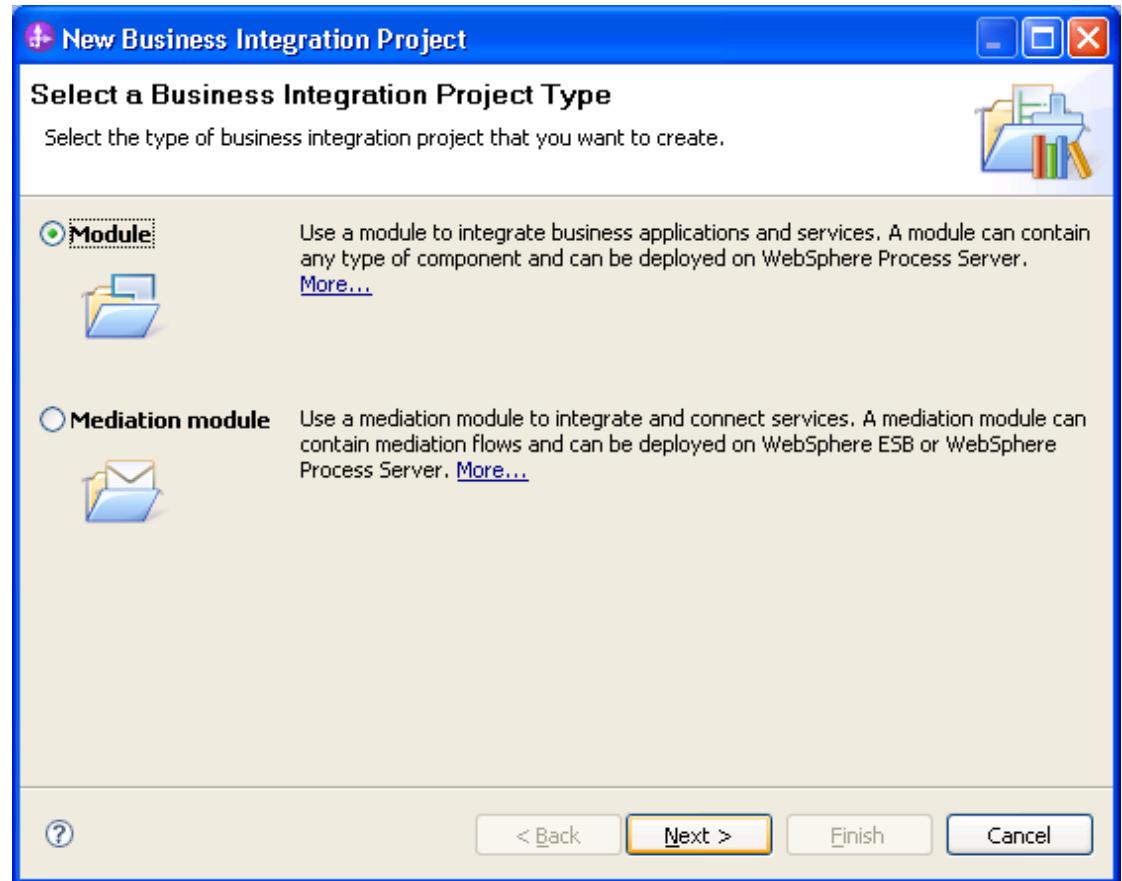
[Advanced >>](#)

[?](#) < Back **Next >** Finish Cancel

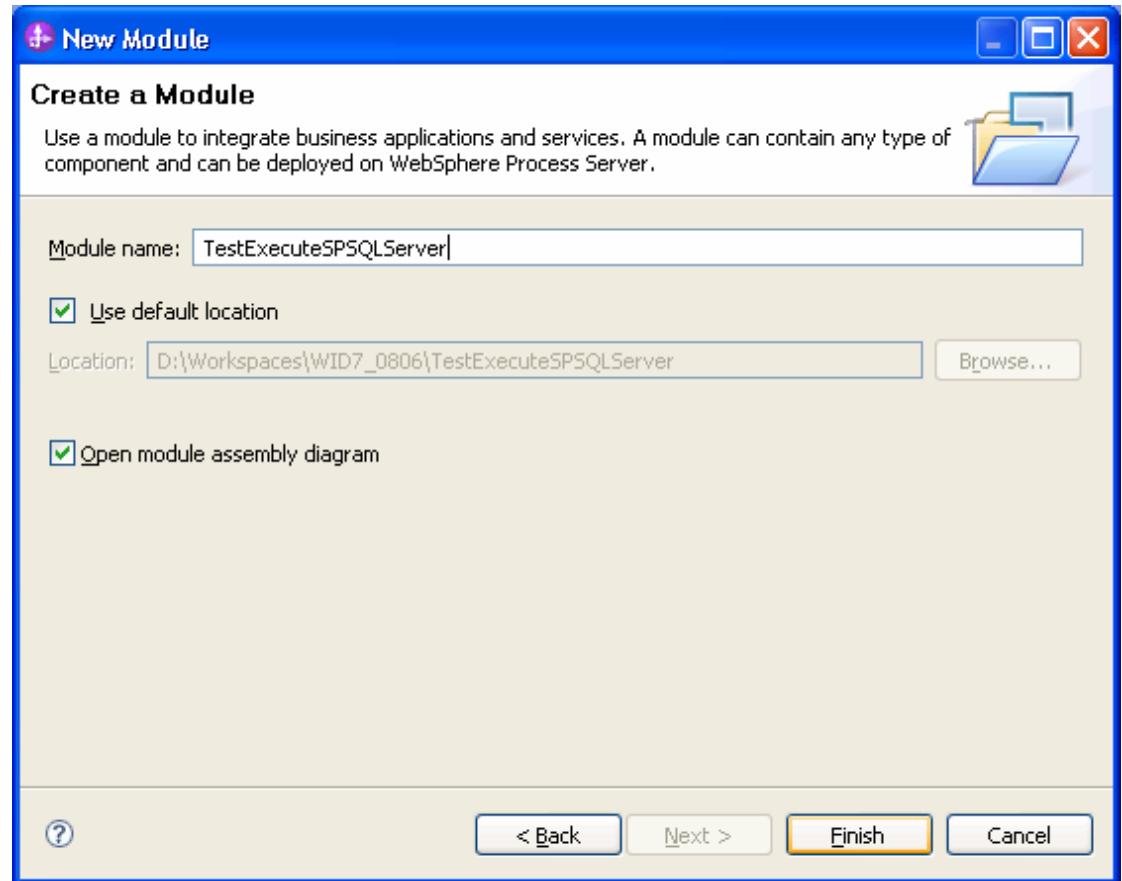
3. Click **New** in the Specify the Location Properties window.



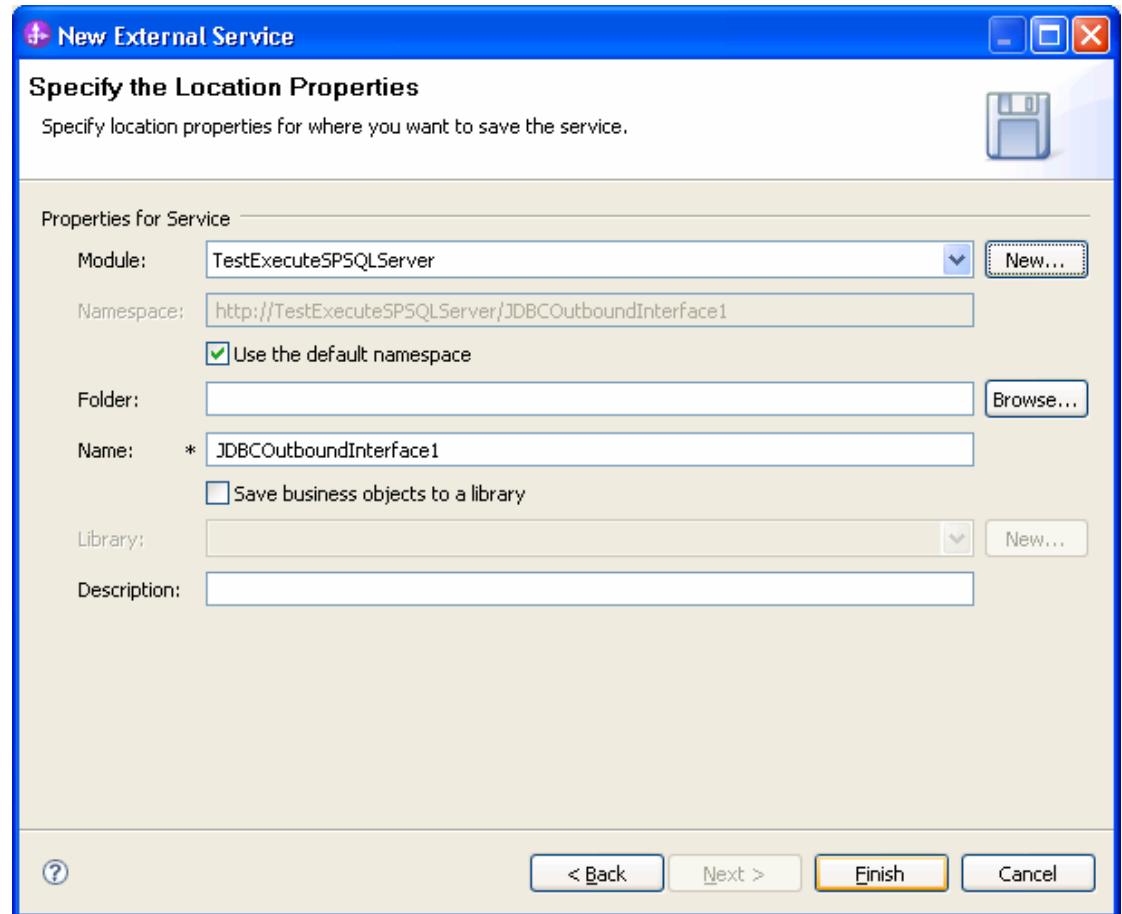
4. In the Select a Business Integration Project Type window, select **Module** and click **Next**.



5. In the Create a Module window, type **TestExecuteSPSQLServer** in the **Module Name** field and click **Finish**.



6. Accept the default values and click **Finish**.



7. Expand the created Business Integration Project and verify whether the artifacts are generated correctly.

Projects [New...](#)

- + CommonThirdParty
- + CWYBC_JDBC_Local
- + CWYBC_JDBC_XA
- + TestCreateSPOracle
- + TestCreateSPSQLServer
- **TestExecuteSPSQLServer**
 - Assembly Diagram
 - JDBCOutboundInterface
 - Dependencies
 - Integration Logic
 - Data Types
 - DboProcedure1U591
 - DboProcedure1U591BG
 - DboProcedure1U591Retrs1
 - DboProcedure1U591Retrs2
 - IntegrityConstraintFault
 - MatchesExceededLimitFault
 - MissingDataFault
 - MultipleMatchingRecordsFault
 - ObjectNotFoundFault
 - PrimaryKeyValuePair
 - RecordNotFoundFault
 - UniqueConstraintFault
 - WBIFault
 - Interfaces
 - JDBCOutboundInterface
 - Transformations
- + TestInboundWrapper
- + TestStructArray

Deploy the module to the test environment

After running the external service wizard, you will have an SCA module that contains an Enterprise Information System (EIS) import. You must install this SCA module in the WebSphere Integration Developer integration test client. To do this, you must add the SCA module you created earlier to the server using the **Servers** view in WebSphere Integration Developer.

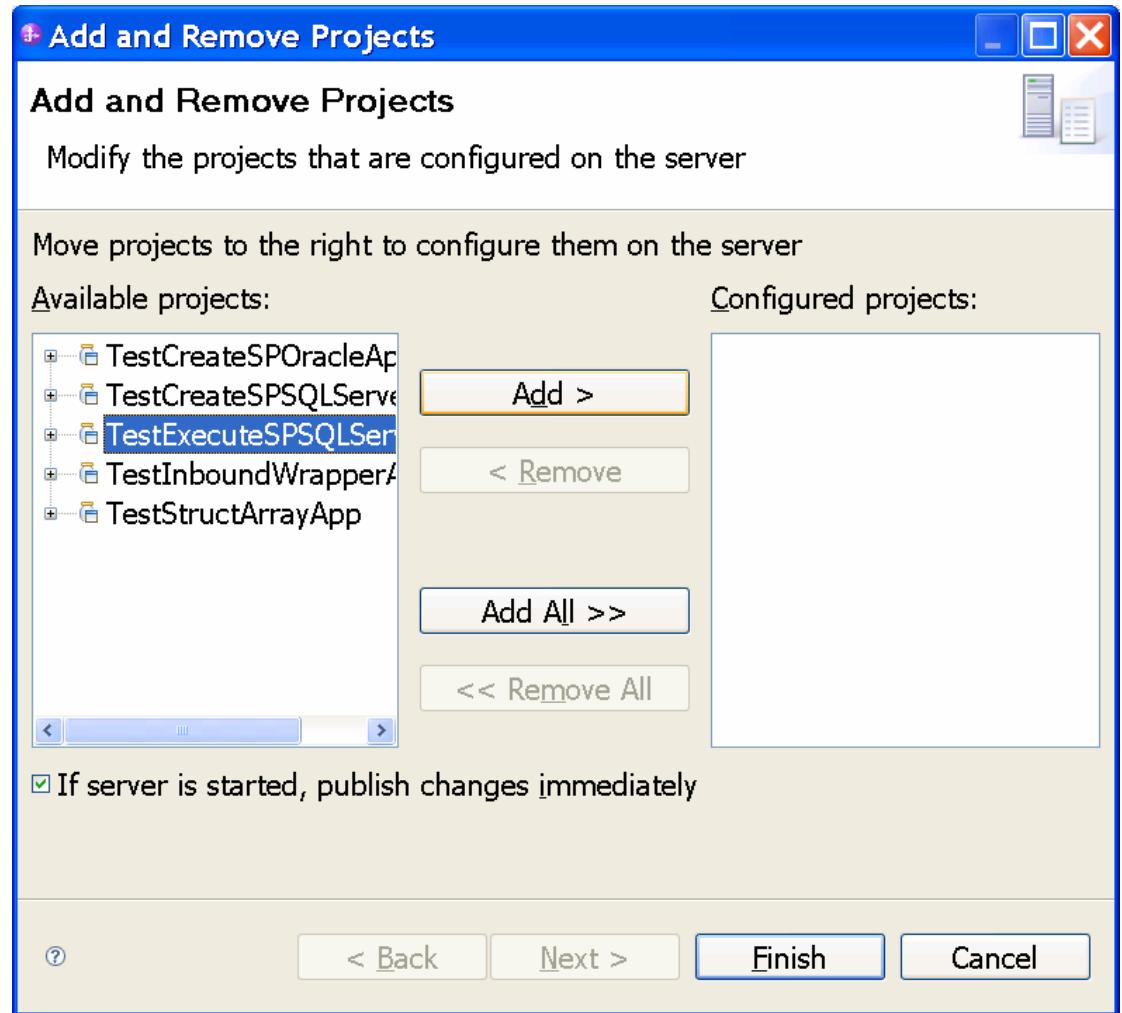
Steps for adding the SCA module to the server:

1. In WebSphere Integration Developer, switch to the **Servers** view by selecting from the toolbar **Window > Show View > Servers**.
2. In the **Servers** tab in the lower-right pane right click the server, and select **Start**.
3. After the server is started, right-click the server, and select **Add and Remove projects**.

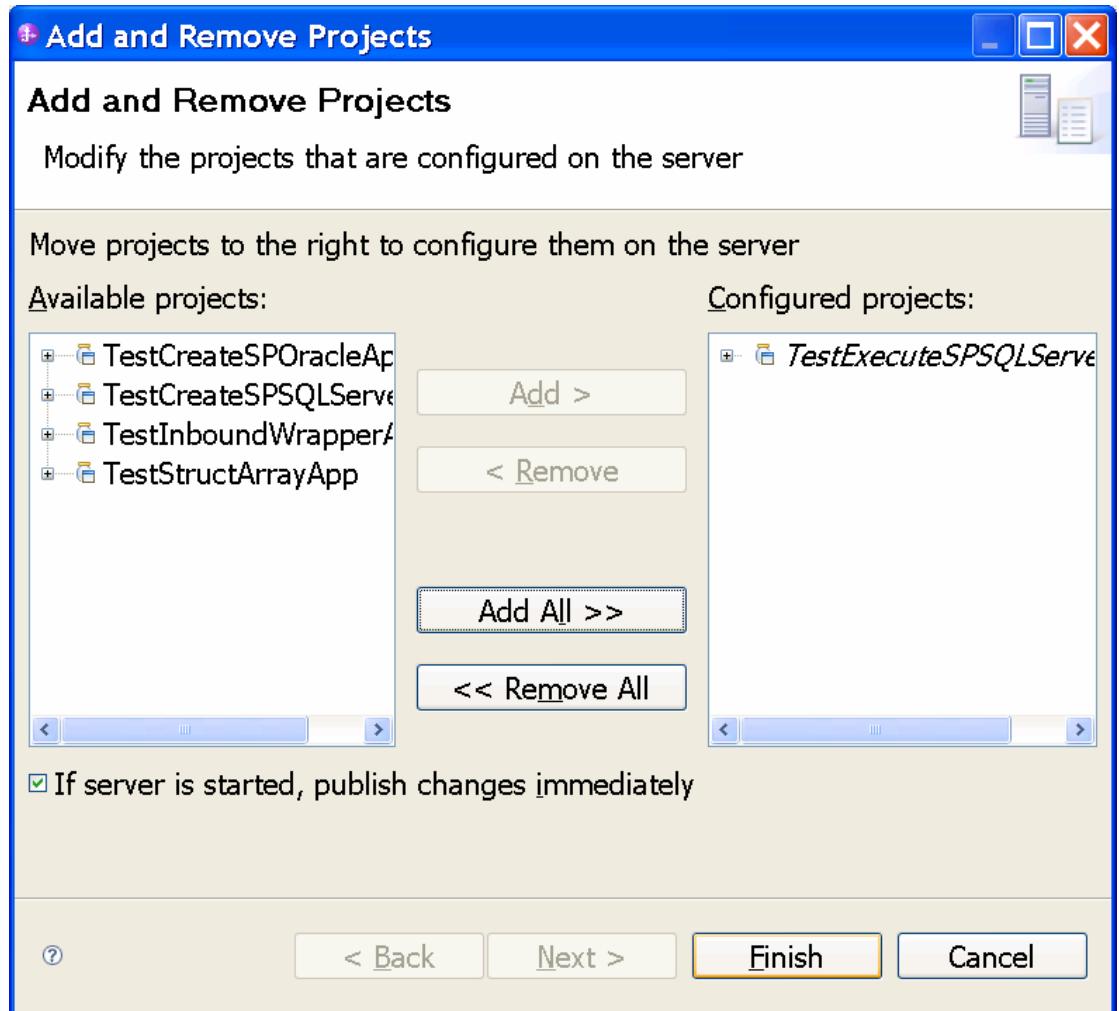
WebSphere software

New	▶
Open	F3
Show In	Alt+Shift+W ▶
Copy	Ctrl+C
Paste	Ctrl+V
✖ Delete	Delete
Rename	F2
✳ Restart in Debug	Ctrl+Alt+D
▶ Restart	Ctrl+Alt+R
⌚ Restart in Profile	
▣ Stop	Ctrl+Alt+S
Publish	Ctrl+Alt+P
Clean...	
Add and Remove Projects...	
Monitoring	▶
>Create tables and data sources	
Reconnect debug process	
View and publish the changes to the server	
Manage server profiles	
Server configuration	▶
Universal test client	▶
Administration	▶
Launch	▶
Add and Remove Integration Solution Projects	▶
Properties	Alt+Enter

The Add and Remove Projects window lists the available projects in the WebSphere Integration Developer workspace.



4. Select your project (**TestExecuteSPSQLServerApp**) and click **Add** to configure the project on the server.

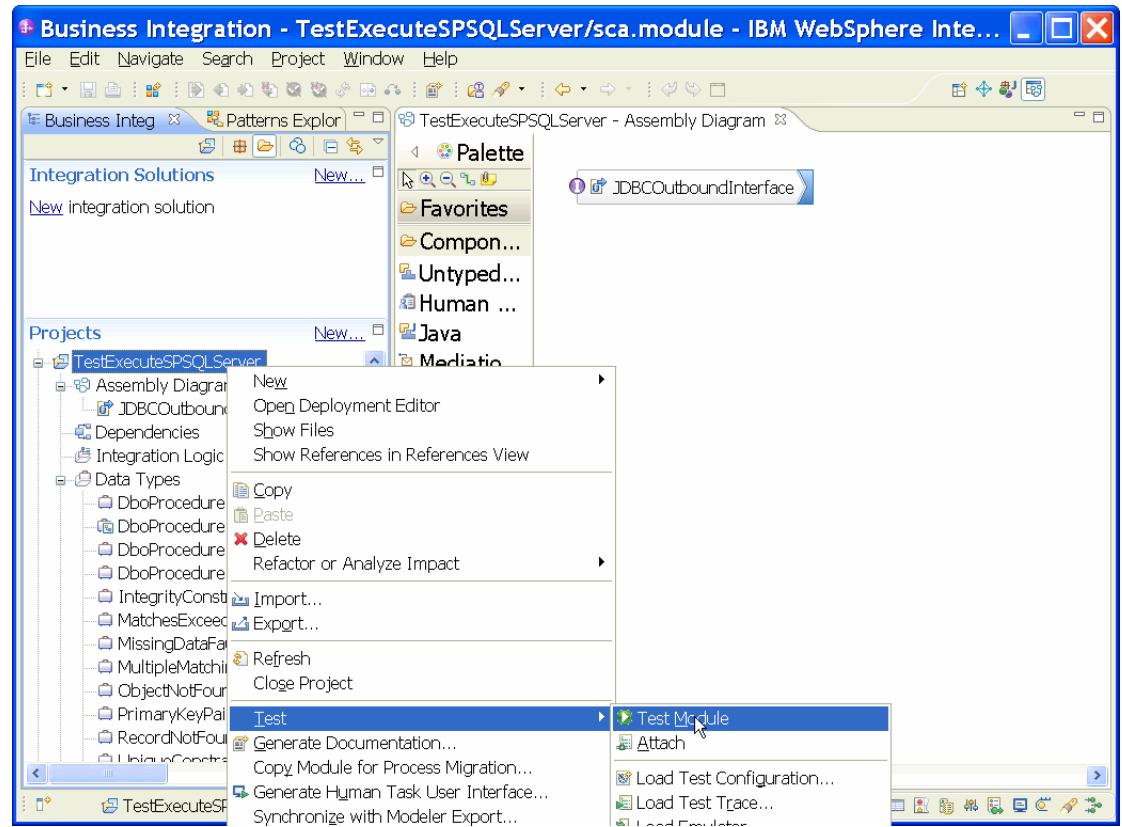


5. Click **Finish**.

Test the assembled adapter application

Test the assembled adapter application using the WebSphere Integration Developer integration test client.

1. Select the **TestExecuteSPSQLServer** Module, right-click it, and select **Test > Test Module**. The Test Client window is displayed.



2. Select **executeDboProcedure1U591BG** operation.

WebSphere software

General Properties

Detailed Properties

Specify the component, interface, operation, and input parameter values for the Invoke event, then click the Continue icon in the Events area to run the test. [More...](#)

<u>Configuration:</u>	Default Module Test	
<u>Module:</u>	TestExecuteSPSQLServer	
<u>Component:</u>	JDBCOutboundInterface	
<u>Interface:</u>	JDBCOutboundInterface	
<u>Operation:</u>	executeDboProcedure1U591BG	

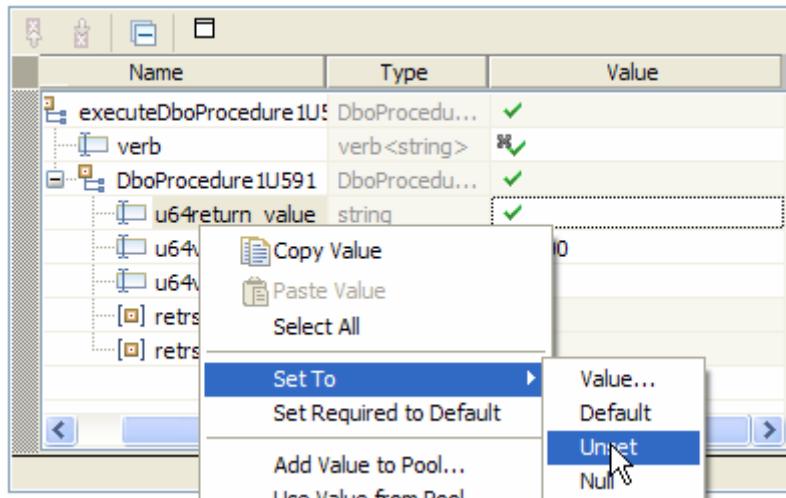
Initial request parameters:

Value editor XML editor

The screenshot shows a 'Value editor' dialog with a tree view on the left and a table on the right. The tree view shows nodes for 'executeDboProcedure1U591BG' (verb: Create), 'DboProcedure1U591BG', 'u64return_val' (int: 0), 'u64var0' (string), 'u64var1' (string), 'retrs1' (DboProcedure1U591BG), and 'retrs2' (DboProcedure1U591BG). The table has columns for Name, Type, and Value. The rows correspond to the tree nodes, with 'retrs1' having a red error icon next to it.

Name	Type	Value
executeDboProcedure1U591BG	DboProcedure1U591BG	✓
verb	verb<string>	✓ Create
DboProcedure1U591BG	DboProcedure1U591BG	✓
u64return_val	int	✓ 0
u64var0	string	✓
u64var1	string	✓
retrs1	DboProcedure1U591BG	✗
retrs2	DboProcedure1U591BG	✗

3. Enter value for the input type **var0**.
4. Unset the value for the output type **var1** and **return_value**. Right-click **u64return_value**, and select **Set To > Unset**.



An unset field is indicated by a 'X' mark.

Name	Type	Value
executeDboProcedure1U5	DboProcedu...	✓
verb	verb<string>	✓
DboProcedure1U591	DboProcedu...	✓
u64return_value	string	✓
u64var0	string	✓ 100
u64var1	string	X
[] retrs1	DboProcedu...	60
[] retrs2	DboProcedu...	60

5. To execute the service, click .
6. In the Select Deployment location window, select the server, and click **Finish**.

 Deployment Location



Select a Deployment Location



Specify a runtime location where this test will deploy.

Deployment location:

- [-] WebSphere Process Servers
 - [+] WebSphere Process Server v7.0 at localhost
 - Eclipse 1.5 JVM

New Server...

Mode:

Use this location as the default and do not ask again



Finish

Cancel

7. Check the output of the service, and check the data in the Enterprise Information System to ensure it matches the expected values.

Name	Type	Value
executeDboPro...	DboProcedure1U591BG	✓
verb	verb<string>	✗✓
DboProcedure1U5	DboProcedure1U591	✓
u64return_val	string	✓ 100
u64var0	string	✓ 100
u64var1	string	✓ 100
retrs1	DboProcedure1U591Retrs1[]	✗✓
retrs1[0]	DboProcedure1U591Retrs1	✓
pkey	string	✓ 100
lname	string	✓ lname1
fname	string	✓ fname1
ccode	string	✓ IBM
retrs1[1]	DboProcedure1U591Retrs1	✓
pkey	string	✓ 200
lname	string	✓ lname2
fname	string	✓ fname2
ccode	string	✓ IBM
retrs2	DboProcedure1U591Retrs2[]	✗✓
retrs2[0]	DboProcedure1U591Retrs2	✓
addrid	string	✓ 100
custid	string	✓ 100
city	string	✓ test1
zipcode	string	✓ 12345
retrs2[1]	DboProcedure1U591Retrs2	✓
addrid	string	✓ 200
custid	string	✓ 200
city	string	✓ test2
zipcode	string	✓ 12346

Clear the sample content

Return the data to its original state.

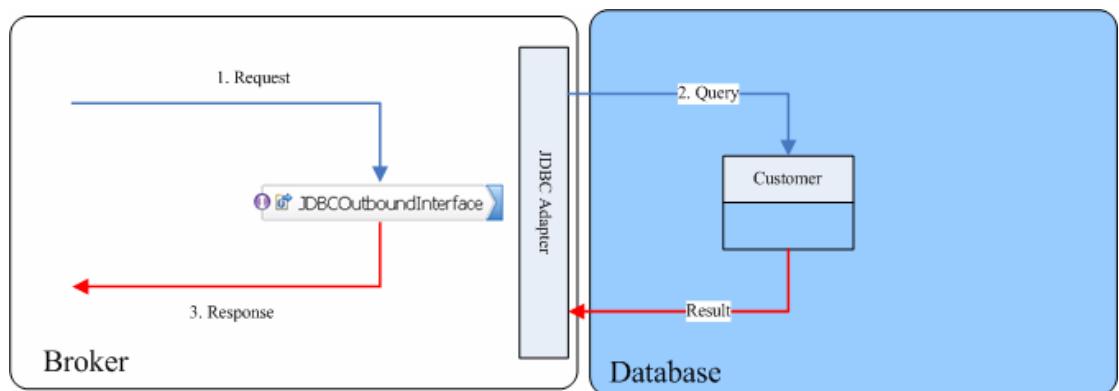
Nothing is required to clean up after this tutorial.

Chapter 13. Tutorial 12: Retrieve business object from database using user defined query (DB2)

This tutorial demonstrates how WebSphere Adapter for JDBC 7.0.0.0 populates customer information into an application's database using user defined query where CUSTOMER and ADDRESS tables have a parent-child relationship.

About this task

In this scenario, an application SCA component raises a retrieveAll test request to the JDBC Outbound Interface. The JDBC adapter executes a SQL query to select all specific records back. Finally, JDBC adapter convert the test result to a SDO and give a response to the SCA component. The following figure represents this scenario:



Prerequisites to run the scenario

Extract the sample files

Replicas of the artifacts that you create when using the external service wizard are provided as sample files for your reference. Use these files to verify if the files you create using the external service wizard are correct.

Download the sample zip file and extract it into a directory of your choice (you may want to create a new directory).

Configuration prerequisites

Before configuring the adapter, you must complete the following tasks:

- Create tables
- Create an authentication alias

Create tables

You must create the following tables in the DB2 database before starting the scenario.

Script for creating the CUSTOMER and ADDRESS tables

```
CREATE TABLE CUSTOMER (
    PKEY VARCHAR(10) NOT NULL,
    FNAME VARCHAR(20) ,
    LNAME VARCHAR(20) ,
    CCODE VARCHAR(10) ,
    PRIMARY KEY(PKEY)) ;

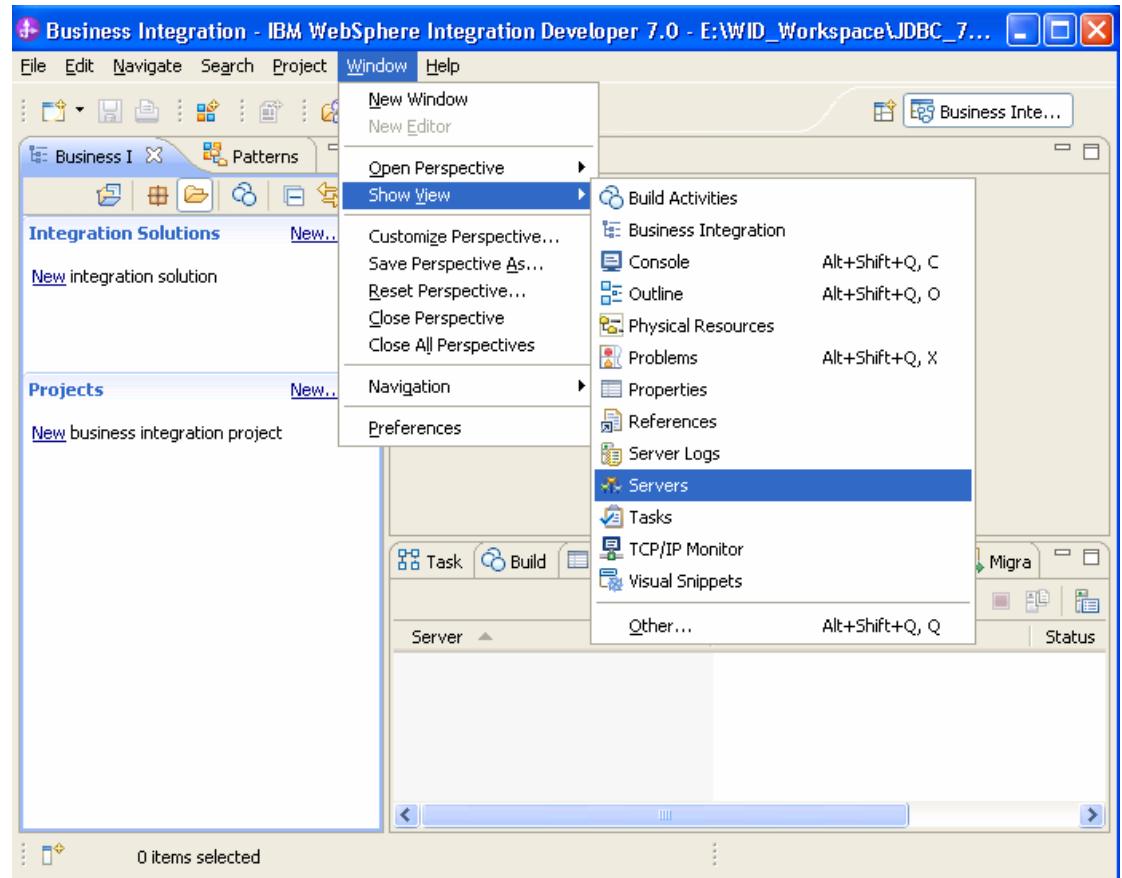
CREATE TABLE ADDRESS (
    ADDRID VARCHAR(10) NOT NULL,
    CUSTID VARCHAR(10) ,
    CITY VARCHAR(20) ,
    ZIPCODE VARCHAR(10) ,
    PRIMARY KEY(ADDRID) ) ;
```

Create an authentication alias

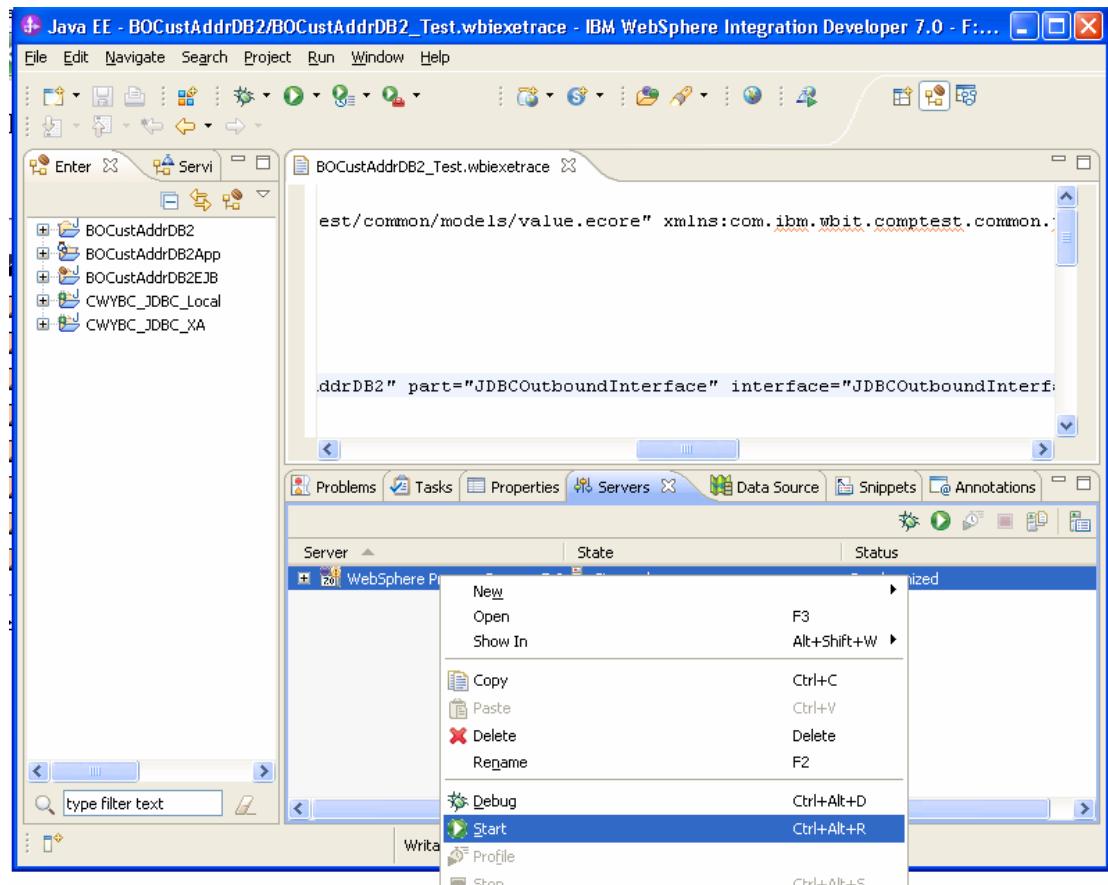
The authentication alias needs to be set because the adapter uses the username and password to connect to the database. This authentication alias will be used later when generating the artifacts for the module.

Here are the steps to set the authentication alias in WebSphere Process Server administration console.

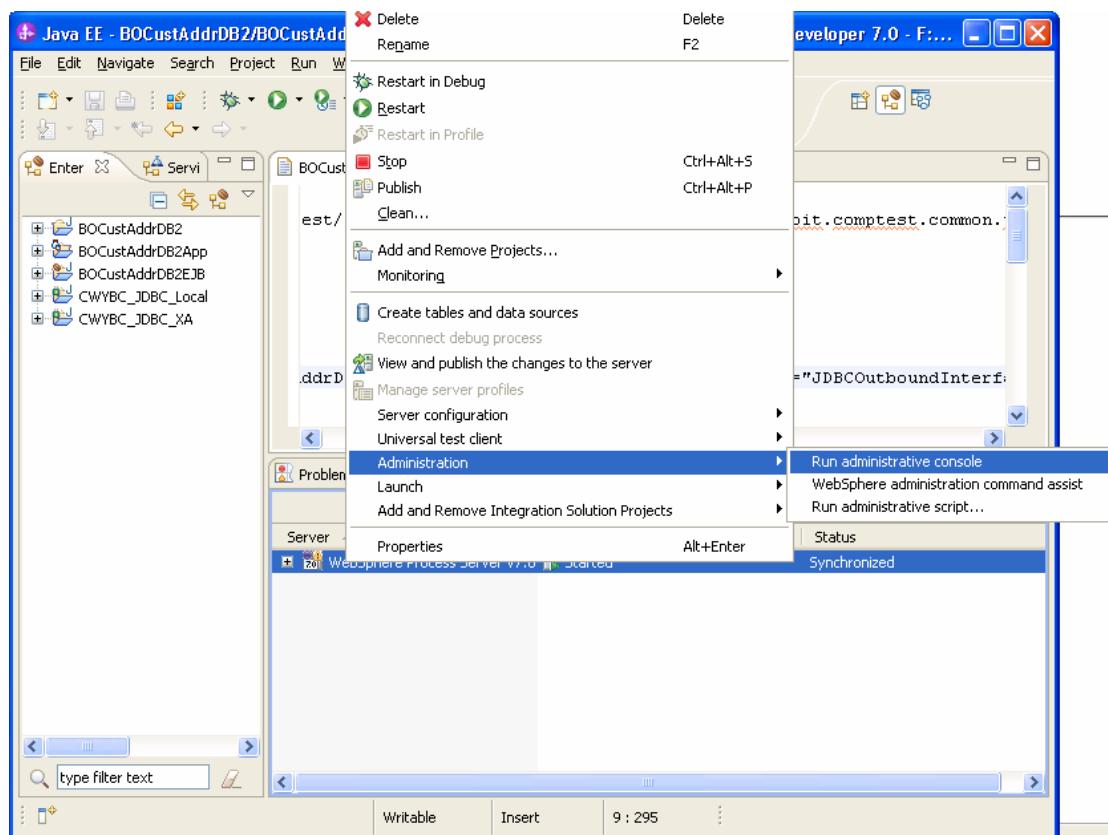
1. In WebSphere Integration Developer, switch to the **Servers** view by selecting **Window > Show View > Servers**.



2. In the **Servers** view, right-click the server that you want to start and select **Start**.



- After the server is started, right-click the server, and select **Administration > Run administrative console**.



4. Log on to the administrative console.

Integrated Solutions Console

Log in to the console.

User ID:

Log in

Note: After some period of inactivity, the system will log you out automatically and ask you to log in again.

5. In the administrative console, click **Security -> Global security**.
6. Under **Java Authentication and Authorization Service**, click **J2C authentication Data**.

The screenshot shows the 'Secure administration, applications, and infrastructure' configuration page. The 'J2C authentication Data' section is selected. It contains fields for 'Application logins', 'System logins', and 'J2C authentication data'. A mouse cursor is hovering over the 'J2C authentication data' link.

A list of existing aliases is displayed.

7. Click **New** to create a new authentication entry. Type the alias name, username and password to connect to the database. Click **OK**.

[Secure administration, applications, and infrastructure](#) > [JAAS - J2C authentication data](#) > [SCA_Auth_Alias](#)

Specifies a list of user identities and passwords for Java(TM) 2 connector security to use.

Configuration

General Properties

* Alias

SCA_Auth_Alias

* User ID

db2admin

* Password

Description

This is the alias used by SCA to

[Apply](#) [OK](#) [Reset](#) [Cancel](#)

- Click **Save** to save the changes.

Integrated Solutions Console Welcome Help | Logout

Secure administration, applications, and infrastructure

[Secure administration, applications, and infrastructure](#)

Messages

- Changes have been made to your local configuration. You can:
 - [Save](#) directly to the master configuration.
 - [Review](#) changes before saving or discarding.
- The server may need to be restarted for these changes to take effect.

New	Delete
<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>
Select	Alias
<input type="checkbox"/>	BSpace JDBC Alias
<input type="checkbox"/>	CommonEventInfrastructureJMSAuthAlias
<input type="checkbox"/>	SCA_Auth_Alias
	User ID
	TEST
	CEI
	db2admin
	Description
	Business Space Authorization Alias
	Authentication alias for the Common Event Infrastructure JMS Topics and Queues
	This is the alias used by SCA to login to a ...

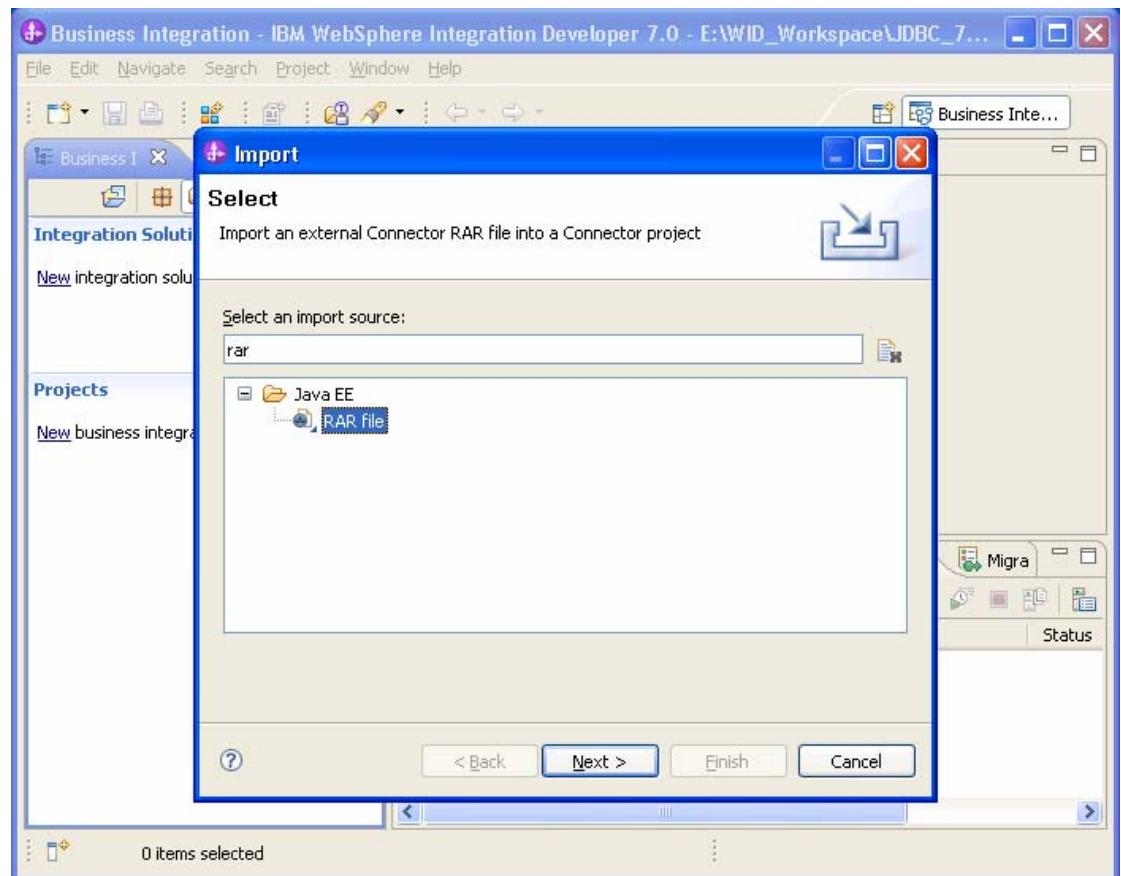
You have created an authentication alias that will be used to configure the adapter properties. Restart the server for the changes to take effect.

Create an adapter project in WebSphere Integration Developer

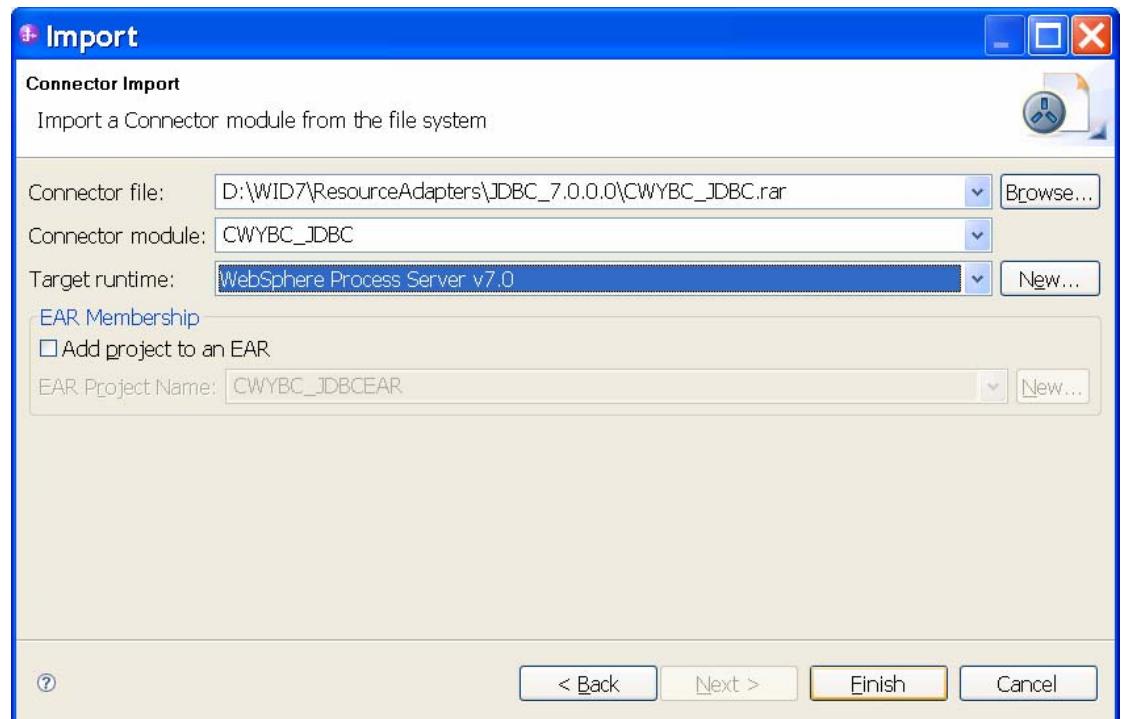
Create an adapter project by importing the resource adapter archive (.RAR) file into the Connector Projects folder in WebSphere Integration Developer. The following steps describe how to do this.

WebSphere software

1. Launch WebSphere Integration Developer by going through the menu on **Windows, Start > Programs > IBM WebSphere > Integration Developer V7.0.0.0**. Alternatively, run **wid.exe** within the installed folder.
2. In WebSphere Integration Developer, switch to the J2EE perspective. To do this, click **Window > Open perspective > Other**, from the menu if not immediately available. In the Select perspective window, select **Show all**, then select **Java EE** from the list and click **OK**.
3. In the Java EE view, import the JDBC adapter's RAR file by right-clicking, and select **Import > RAR file**.

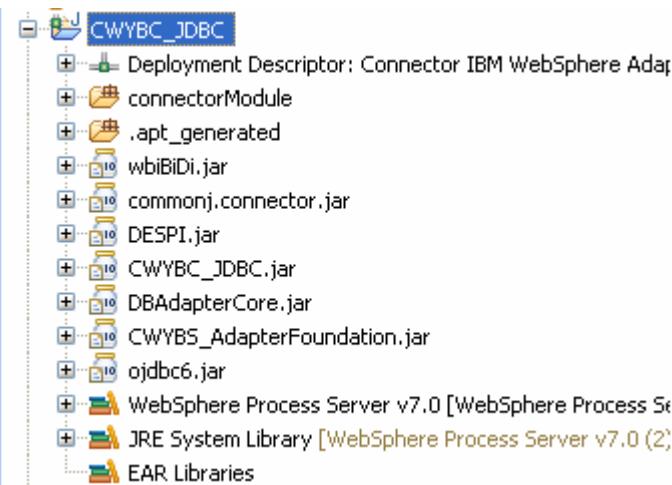


4. In the Connector Import window, click **Browse** in the **Connector file** field, then navigate to the adapter .RAR file.



Based on the imported file, the connector module's name is defined and will determine the project's name, herewith CWYBD_JDBC. The targeted runtime is WebSphere Process Server v7.0.0.0, one of the software requirements

- Accept all other defaults and click **Finish**. A new connector project named **CWYBC_JDBC** appears under the **Connector Projects** folder.



Configure the adapter for outbound processing

Run the external service wizard to specify business objects, services, and configuration details.

The wizard will guide you to do the following steps:

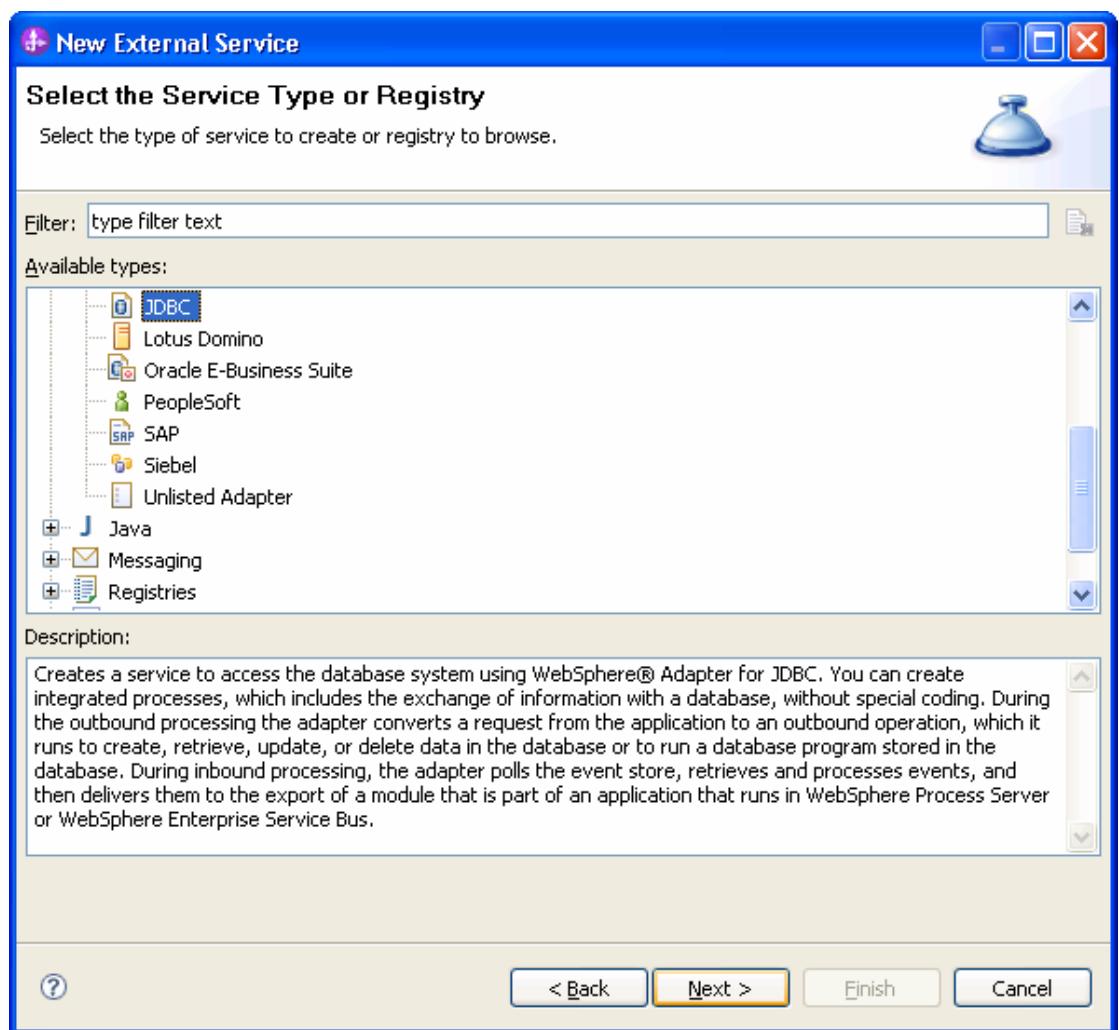
- Setting connection properties for the enterprise service discovery wizard

WebSphere software

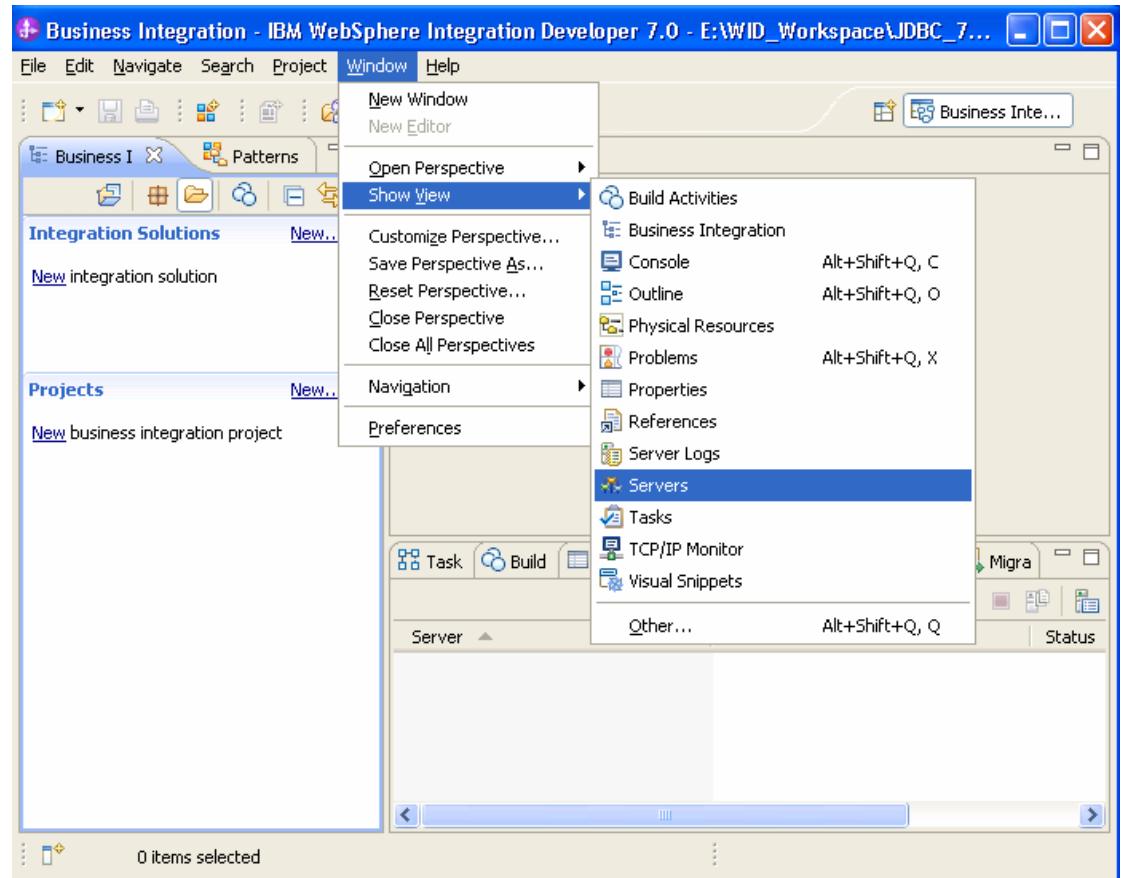
- Selecting the business objects and services to be used with the adapter
- Generating business object definitions and related artifacts

Follow these instructions to launch the Enterprise Service Discovery (ESD) wizard.

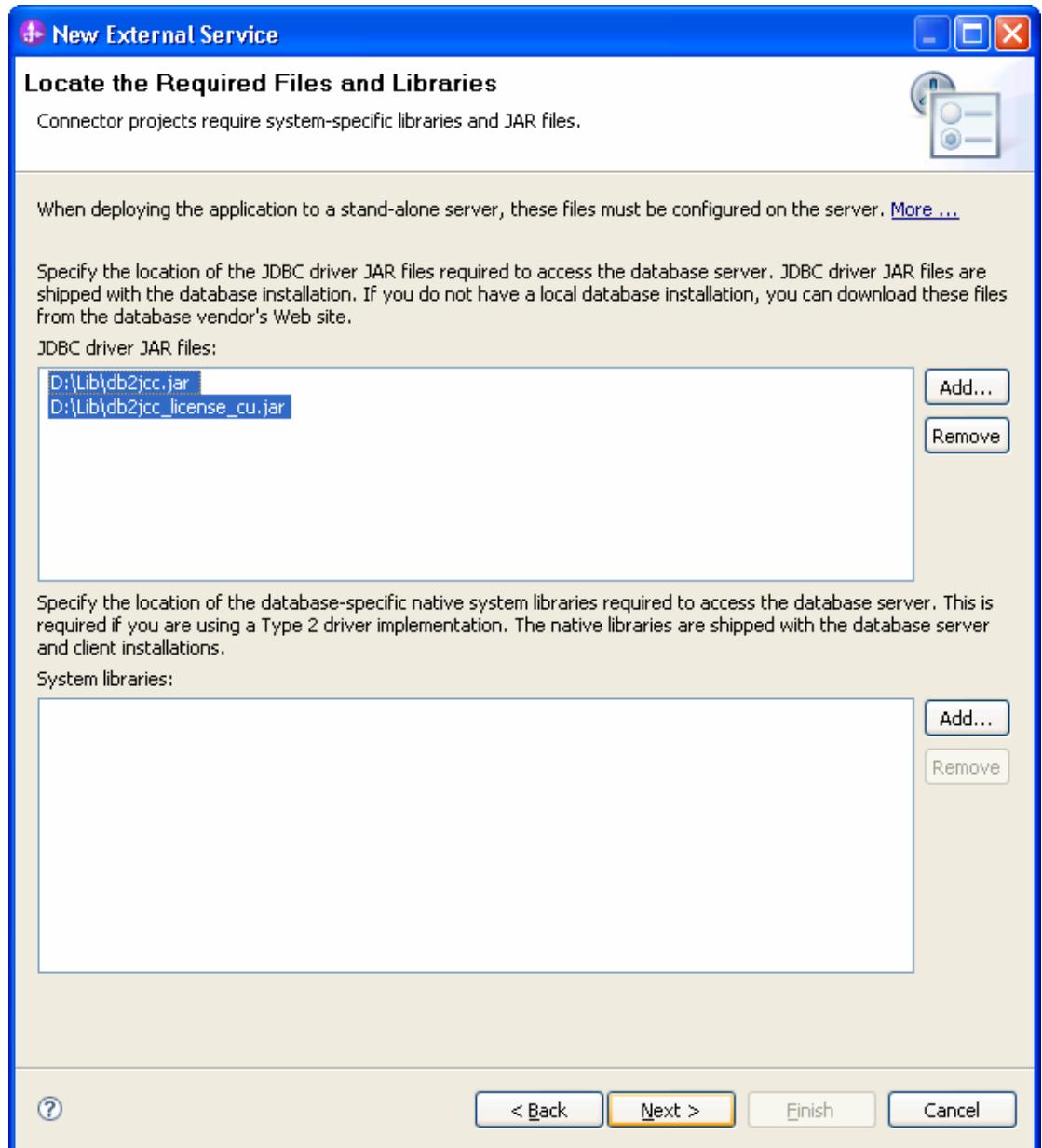
1. Switch to the Business Integration Perspective in WebSphere Integration Developer by selecting **Window -> Open Perspective Business Integration (default)**.
2. Start the external service wizard by selecting **File-> New -> External Service**.
3. In the **Available Types** area, select **Adapters > JDBC** and click **Next**.



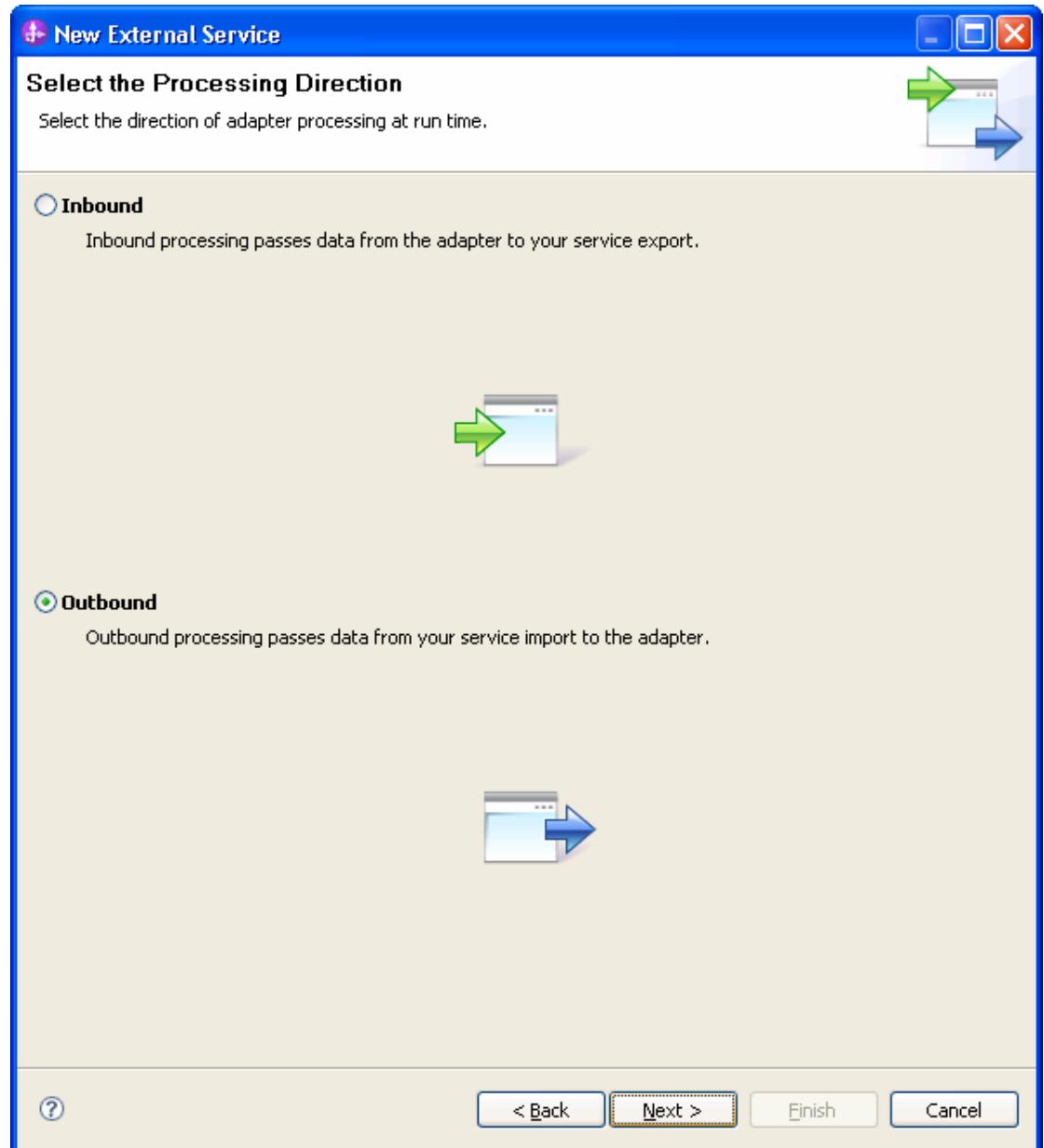
4. Select the **IBM WebSphere Adapter for JDBC (IBM: 7.0.0.0)** and click **Next**.



5. In the **Connector project** field enter **CWYBC_JDBC**.
6. If the JDBC Driver JAR files are not found on the project's classpath, a message to have the necessary files to be added is displayed. Each type of database server requires its own JDBC driver jar files to carry out its operations. The drivers are shipped with the database server. Locate the files and add them to the project. Java-based Type 4 database connectivity is recommended and works with these jar files. If Type 2 connectivity is required, locate the native system libraries and add them to the project.



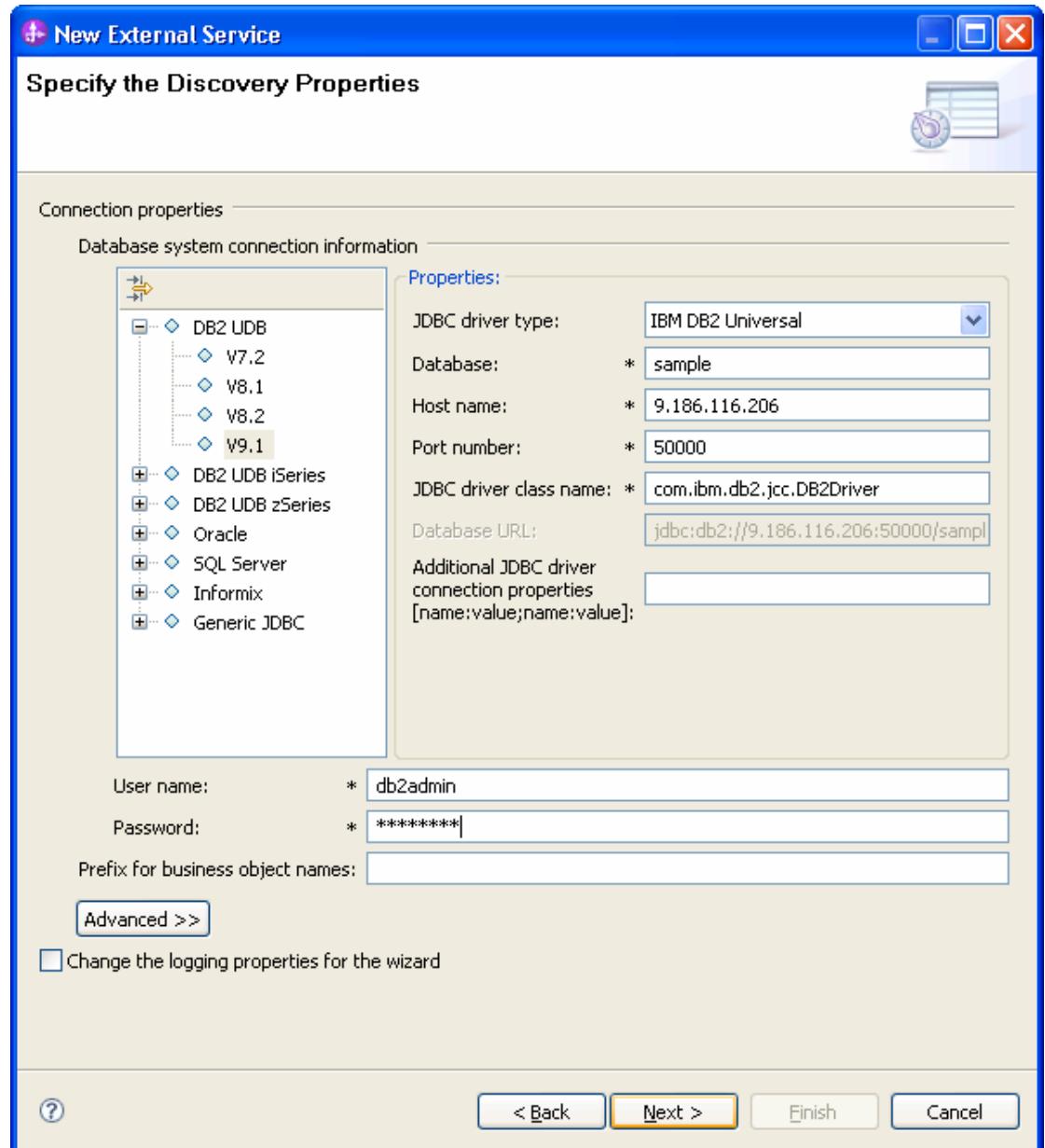
7. Select **Outbound** and click **Next**.



Set connection properties for the external service wizard

To connect to the preferred database:

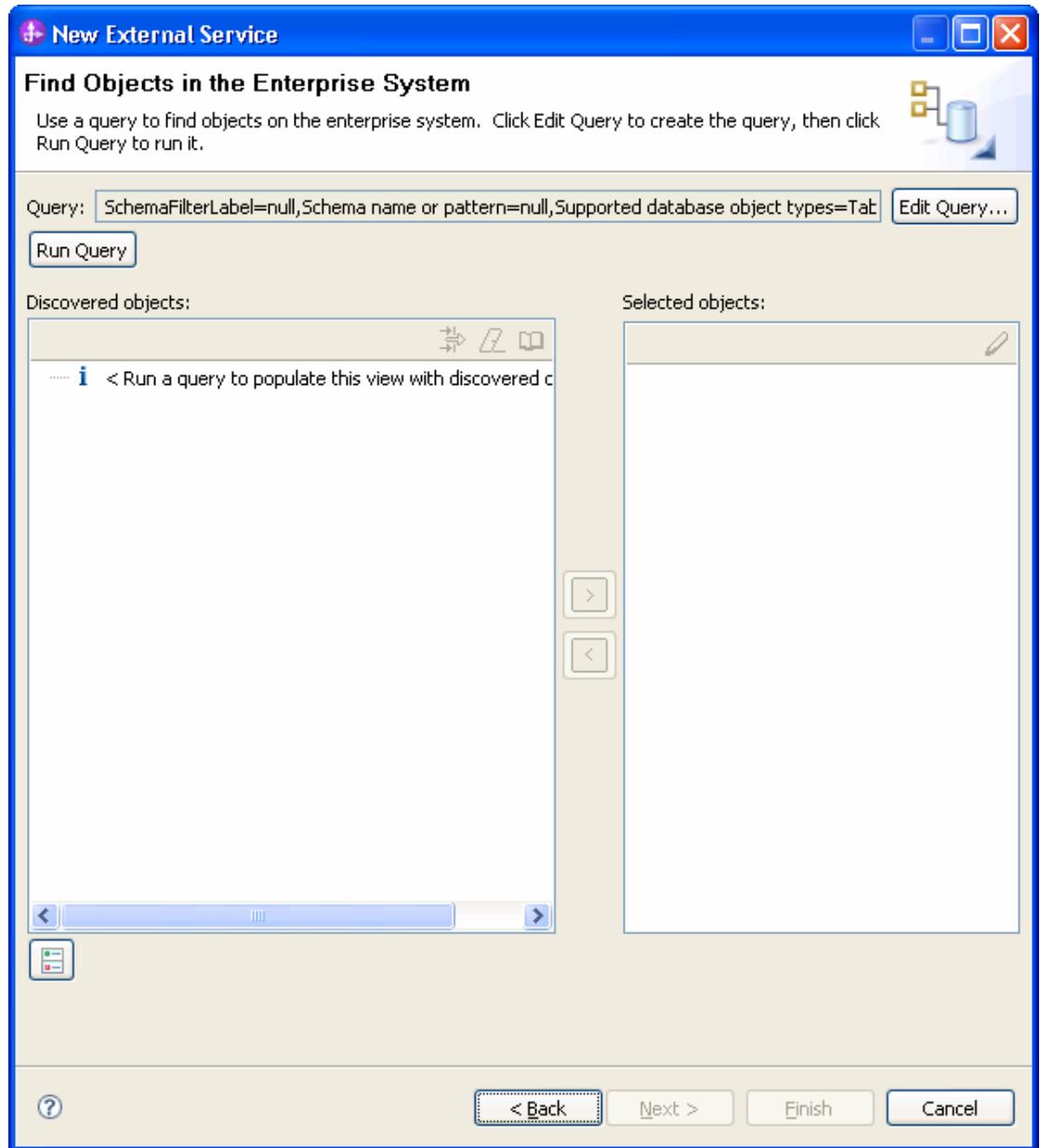
1. Select the appropriate database server in the **Database system connection information** area.
2. Enter values in the **System ID**, **Host name**, **Port number**, **User name** and **Password** fields, and click **Next**.



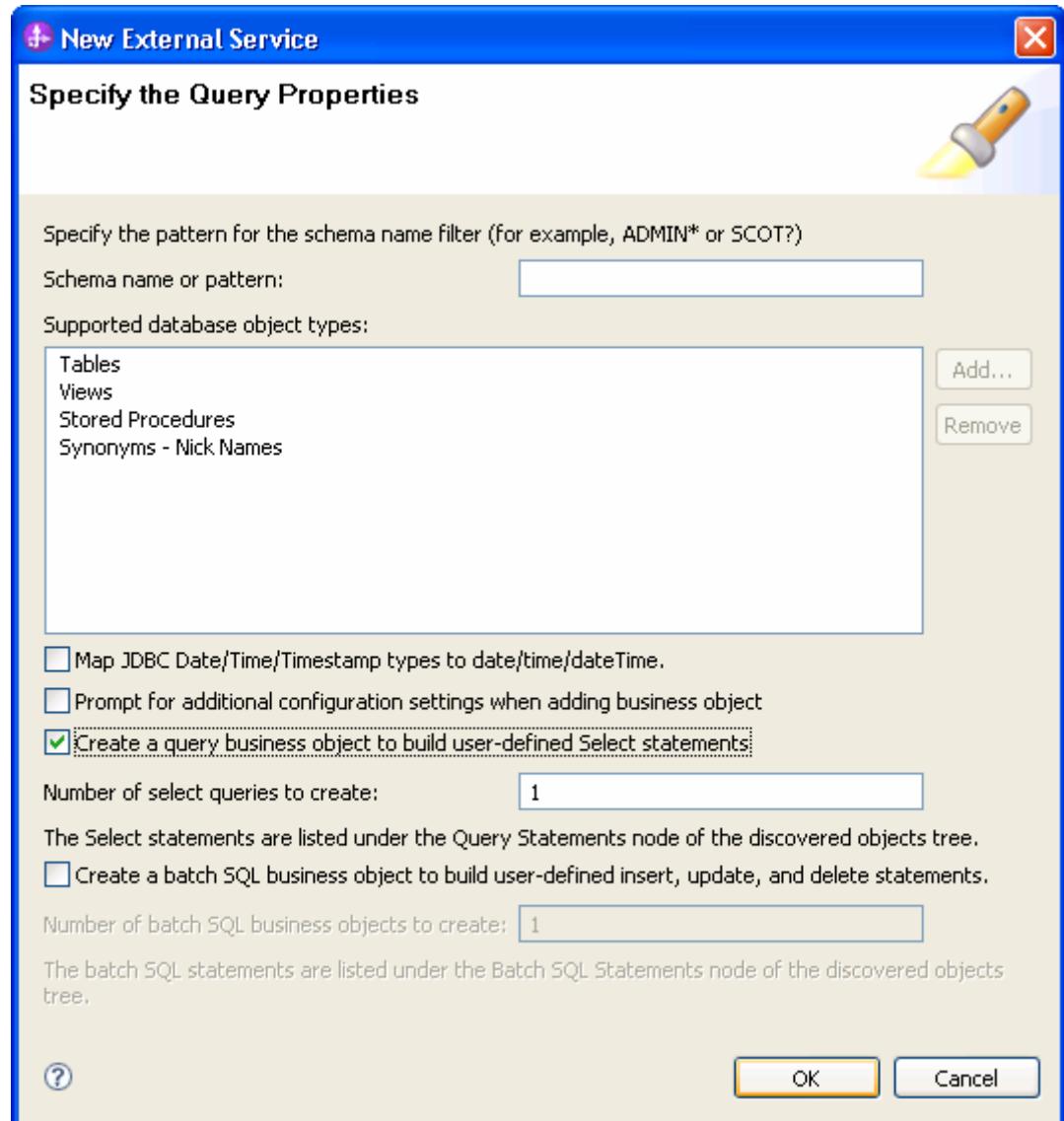
Database connection is established to retrieve the database schema.

Select the business objects and services to be used with the adapter

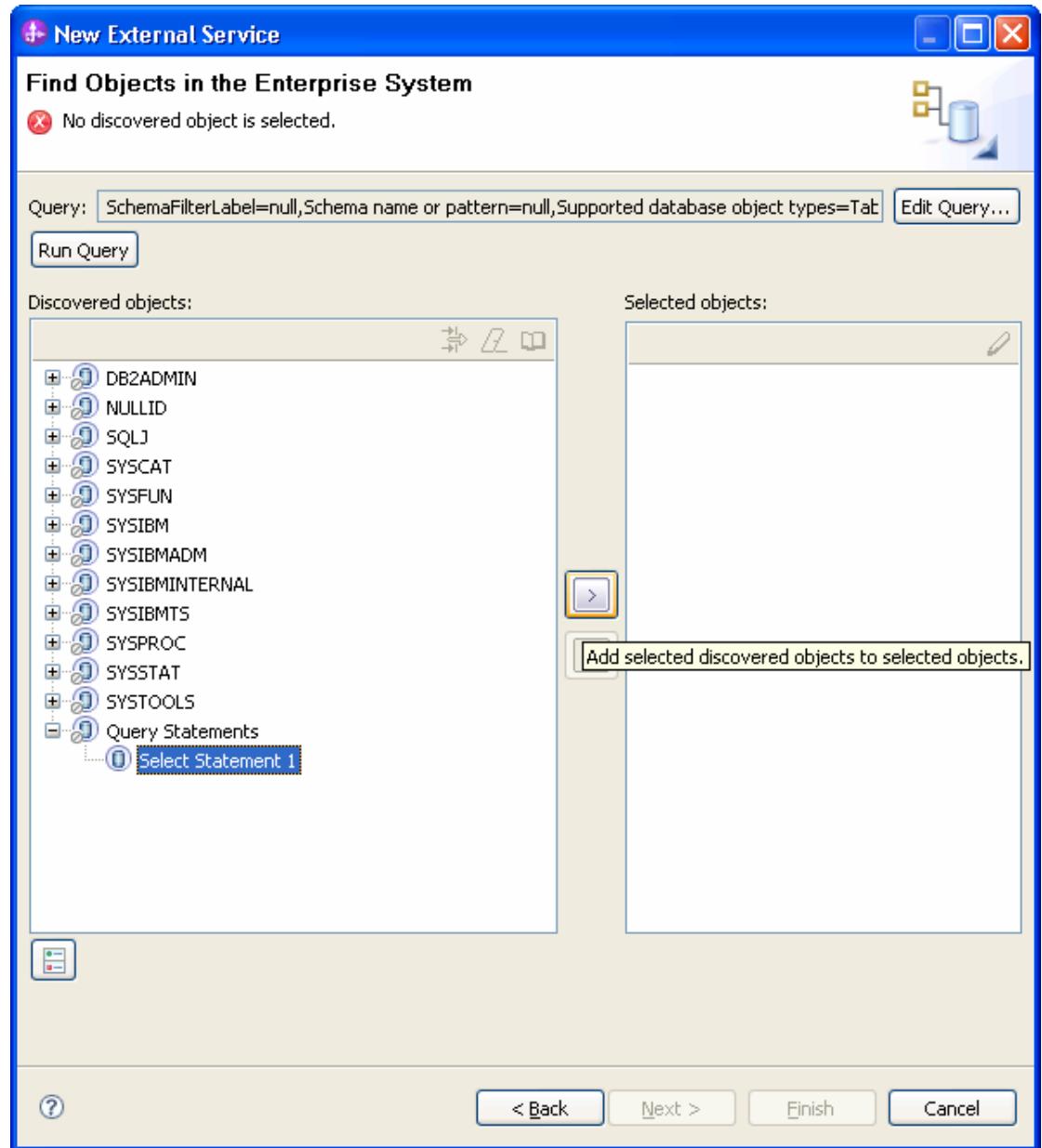
1. In the Find Objects in Enterprise System window, click **Run Query**.



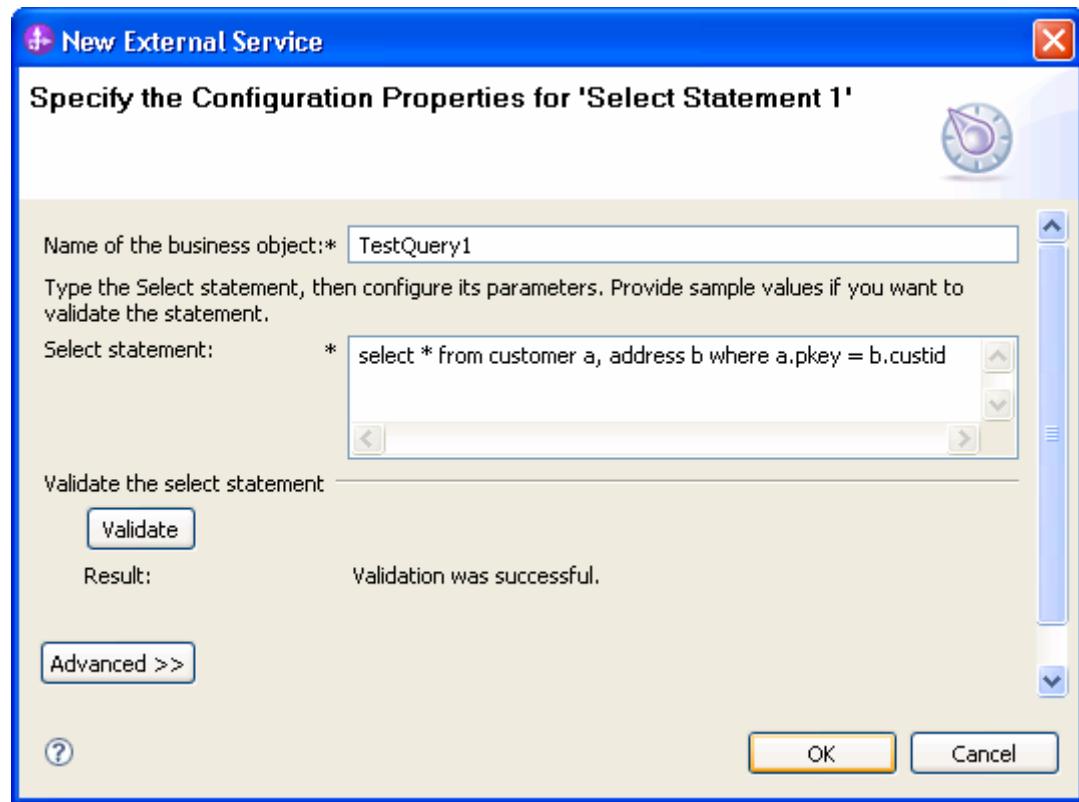
2. In the Specify the Query Properties window, select **Create a query business object to build user-defined select statements** check box and enter the number of query business objects you want to create. Click **OK**.



3. Click **Run Query**.



4. Span Expand the **Query Statements** node, select the **Select Statement 1** and click . The window to configure the query object is displayed.

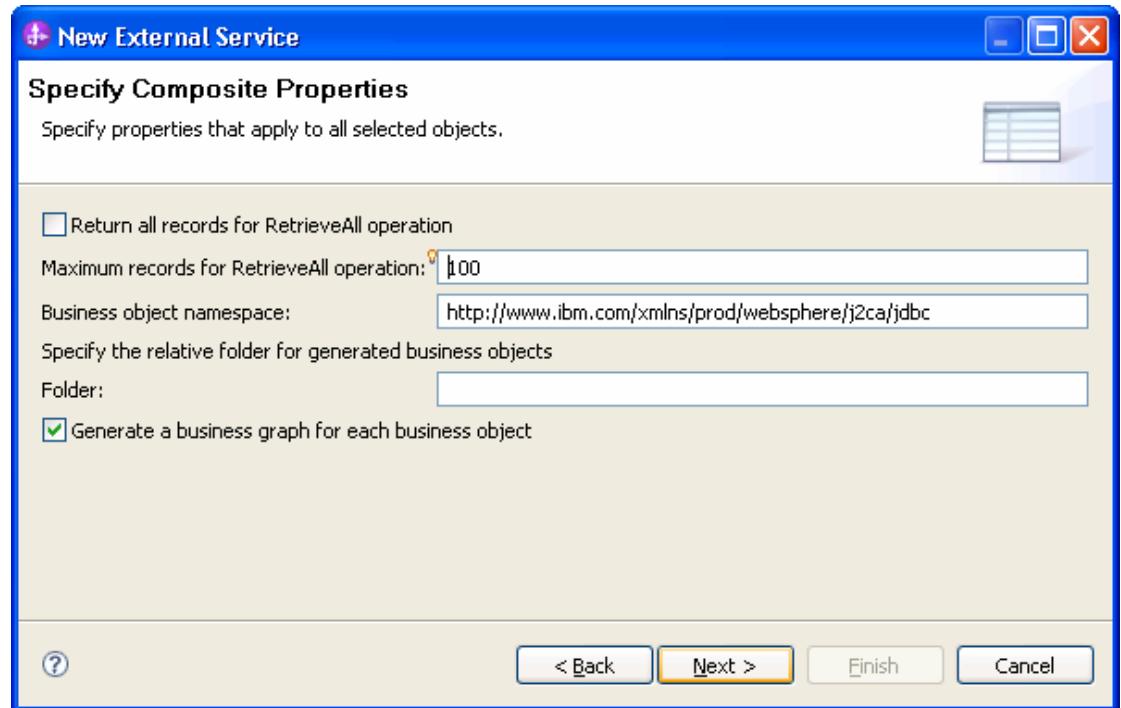


5. In the Name of the business object field, type a name for the business object. The name can contain spaces and national language characters.
6. In the Select statement field, type the SELECT statement you want to run. Indicate each parameter with a question mark (?).
7. Click **Validate**. The Result area displays the result of the validation. Click **OK**.
8. Click **Next**.

Generate business object definitions and related artifacts

Follow these steps to generate the business object definitions.

1. In the Specify Composite Properties window, accept the default values for all fields and click **Next**.



2. In the Specify the Service Generation and Deployment Properties window, perform the following steps:
 - a) Select **Using an existing JAAS alias** for security options under **Deployment Properties**.
 - b) Enter the authentication alias that you created in previous section into the **J2C Authentication data entry** field.
 - c) Clear the **Join the global transaction** check box.
 - d) Select **Specify local database connection information** from the **Database connection information** list, and click **Next**.

New External Service

Specify the Service Generation and Deployment Properties

Specify properties for generating the service and running it on the server.

Service Operations
To modify the names, or add a description to the operations to be generated in the interface file, click Edit Operations... [Edit Operations...](#)

Deployment Properties
How do you want to specify the security credentials?
 Using an existing JAAS alias (recommended)
A Java Authentication and Authorization Services (JAAS) alias is the preferred method.
J2C authentication data entry: *

Using security properties from the managed connection factory
The properties will be stored as plain text; no encryption is used.
User name:
Password:

Other
Use if no security is required or will be handled by the EIS system, or the RAR will be deployed on the server and security will be specified by the properties in the JNDI lookup name.

The quality of service that is used to join the transaction provides a higher degree of data integrity, especially when a failure occurs. To participate in a global transaction, a predefined XA DataSource or XA database connection information must be specified in the connection properties. [More ...](#)

Join the global transaction

Deploy connector project:

Specify the settings used to connect to JDBC at run time:

Connection settings:

Connection Properties
To join a global transaction, specify a predefined XA datasource or XA database connection information. When not joining a global transaction, either the XA connection information or the local connection information can be specified.

Database connection information:

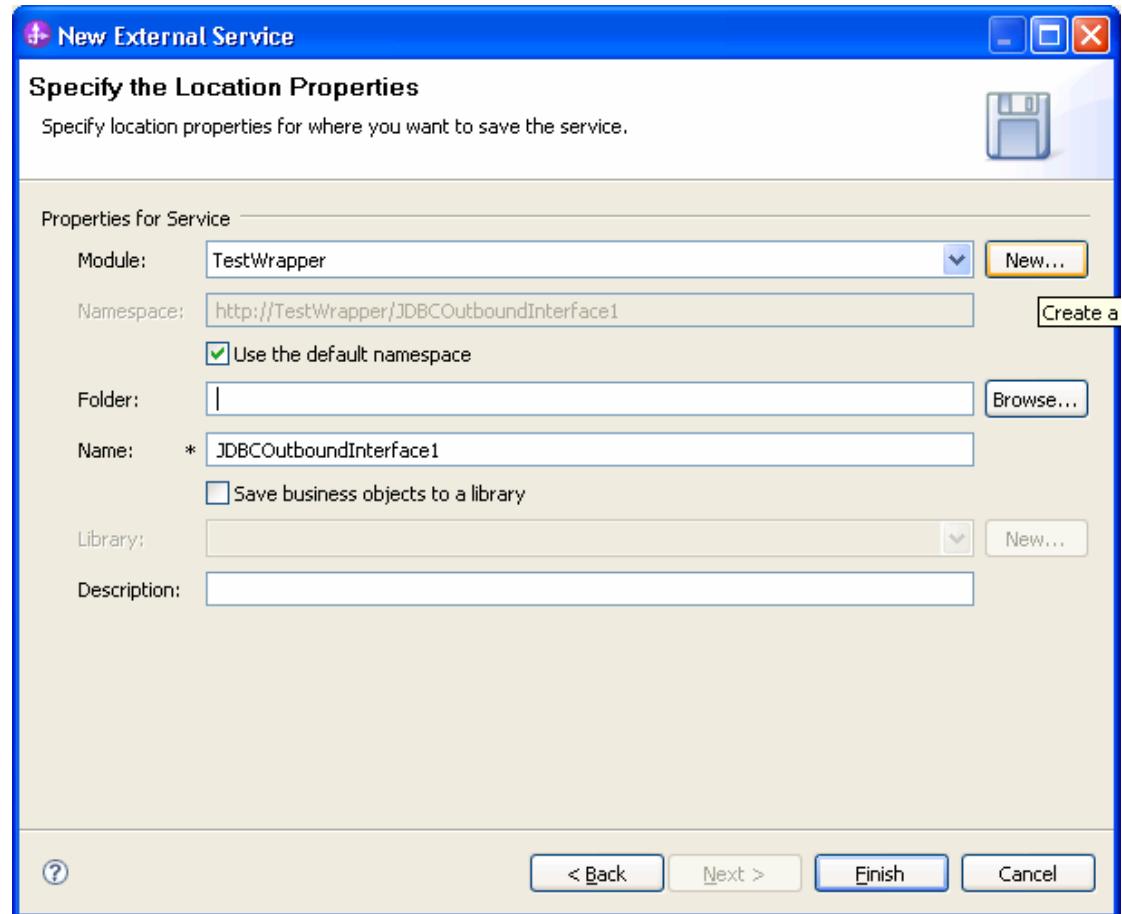
Database system connection information

Database vendor: DB2
Database URL: *
JDBC driver class name: *

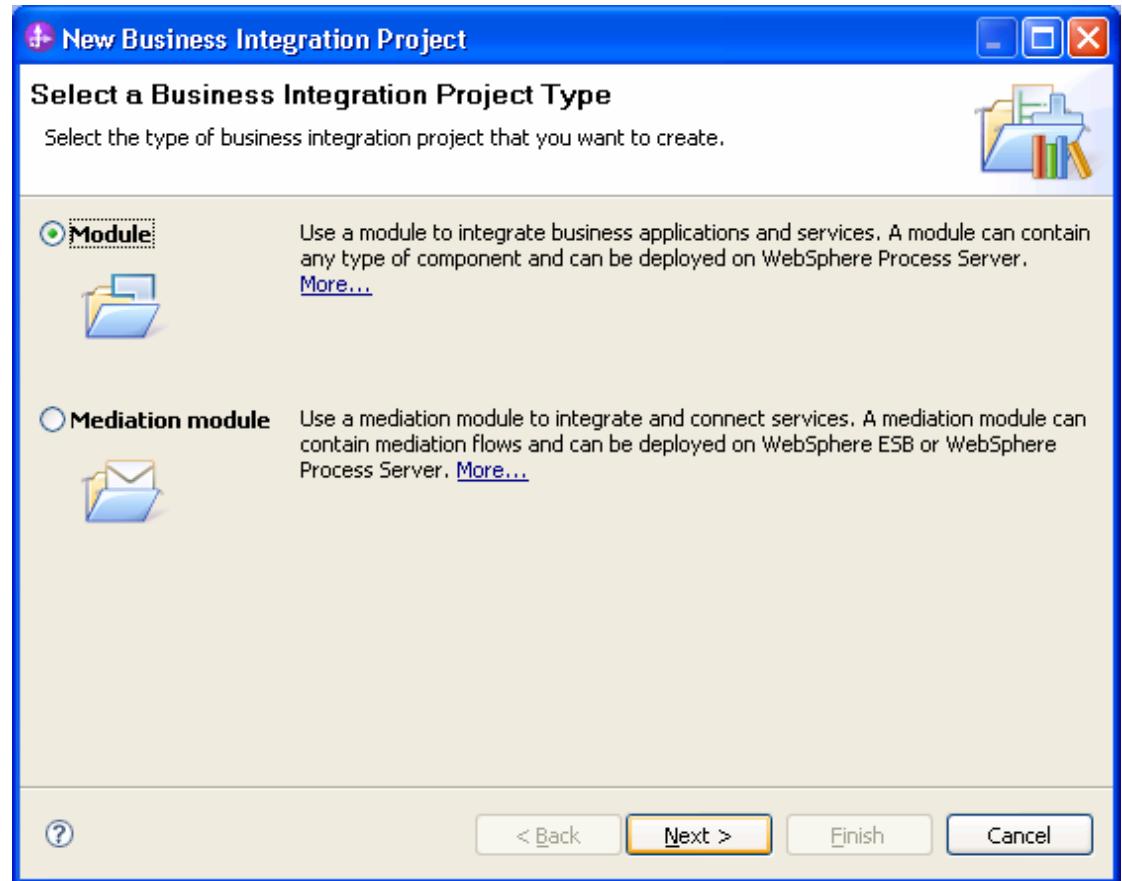
[Advanced >>](#)

[?](#) [Back](#) [Next >](#) [Finish](#) [Cancel](#)

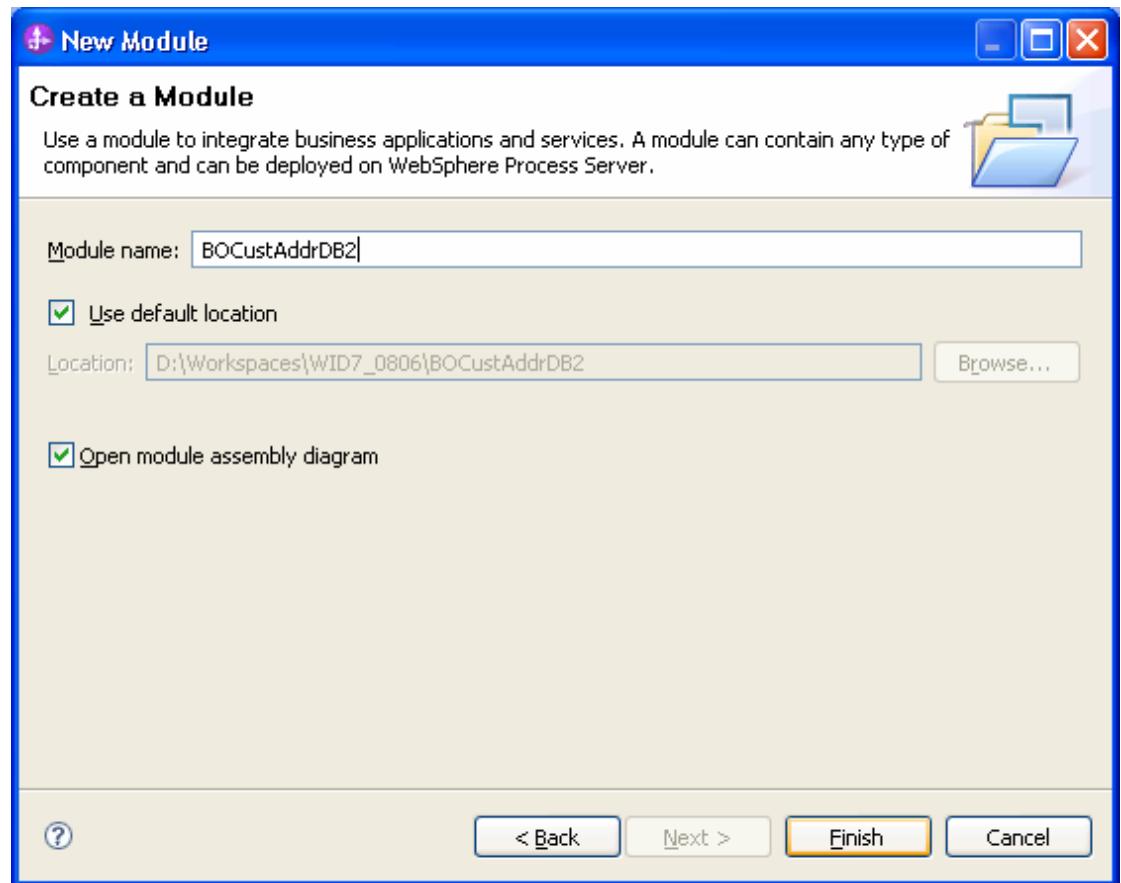
- Click **New** in the Specify the Location Properties window.



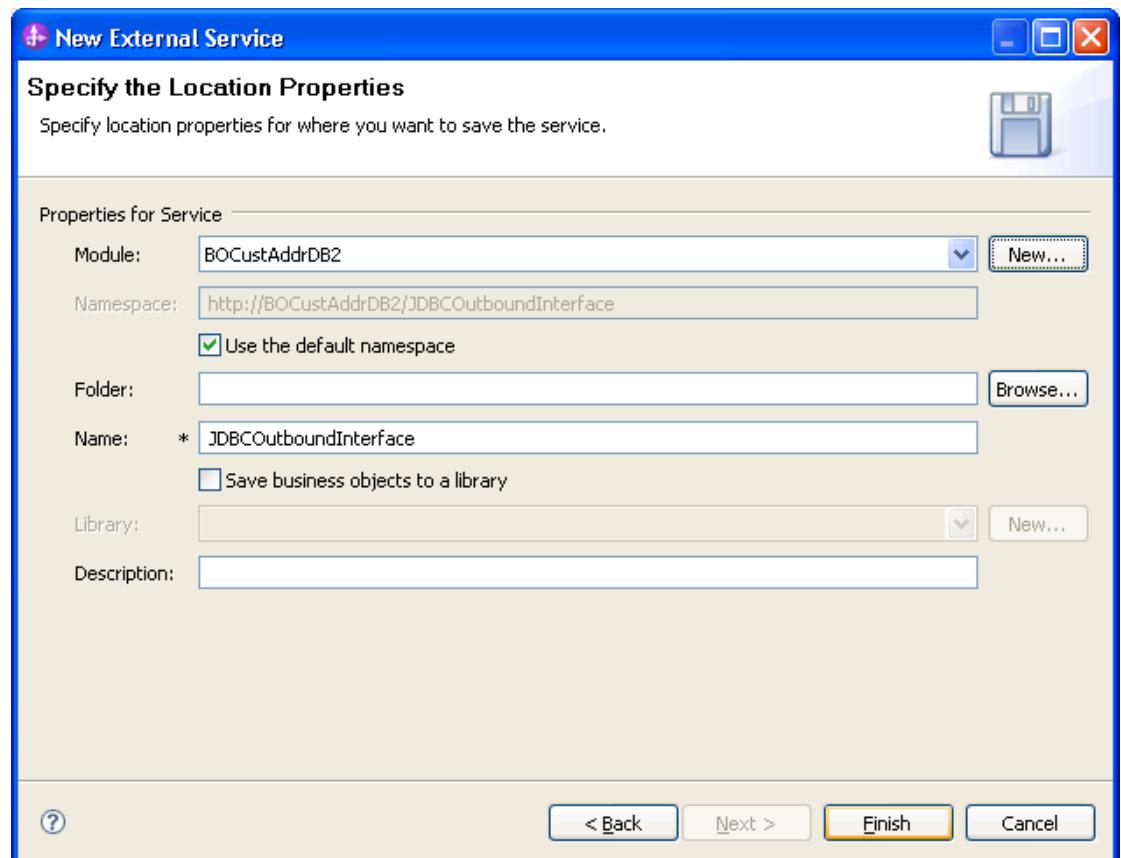
4. In the Select a Business Integration Project Type window, select **Module** and click **Next**.



5. In the Create a Module window, type **BOCustAddrDB2** in the **Module Name** field, and click **Finish**.



6. Accept the default values and click **Finish**.



WebSphere® software

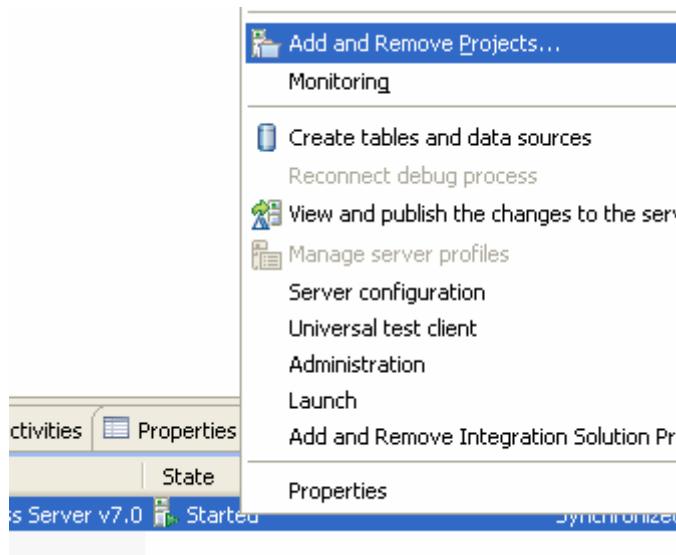
Expand the created Business Integration Project and verify whether the artifacts are generated correctly.

Deploy the module to the test environment

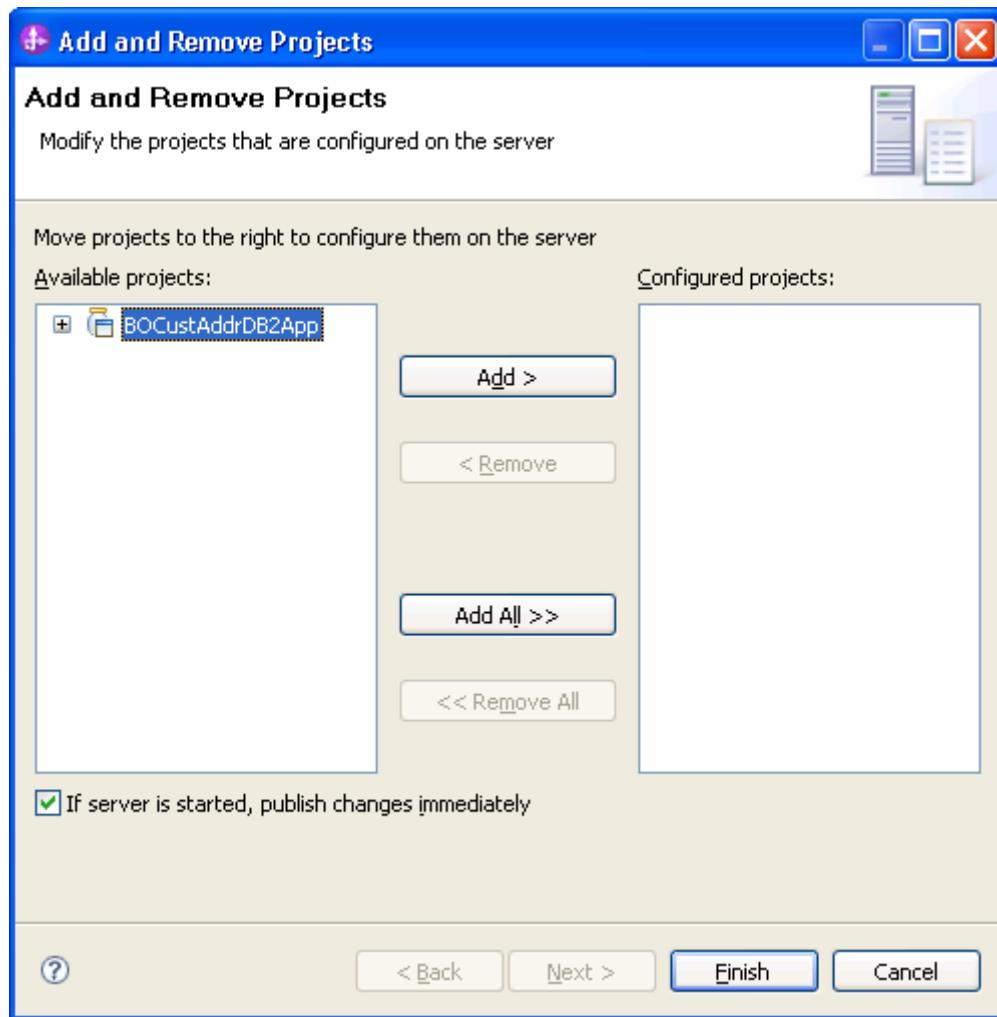
After running the external service wizard, you will have an SCA module that contains an Enterprise Information System import. You must install this SCA module in the WebSphere Integration Developer integration test client. To do this, you must add the SCA module you created earlier to the server using the **Servers** view in WebSphere Integration Developer.

Steps for adding the SCA module to the server:

1. In WebSphere Integration Developer, switch to the **Servers** view by selecting from the toolbar **Window > Show View > Servers**.
2. In the **Servers** tab in the lower-right pane right click the server, and select **Start**.
3. After the server is started, right-click the server, and select **Add and Remove projects**.



4. Add the SCA module to the server.

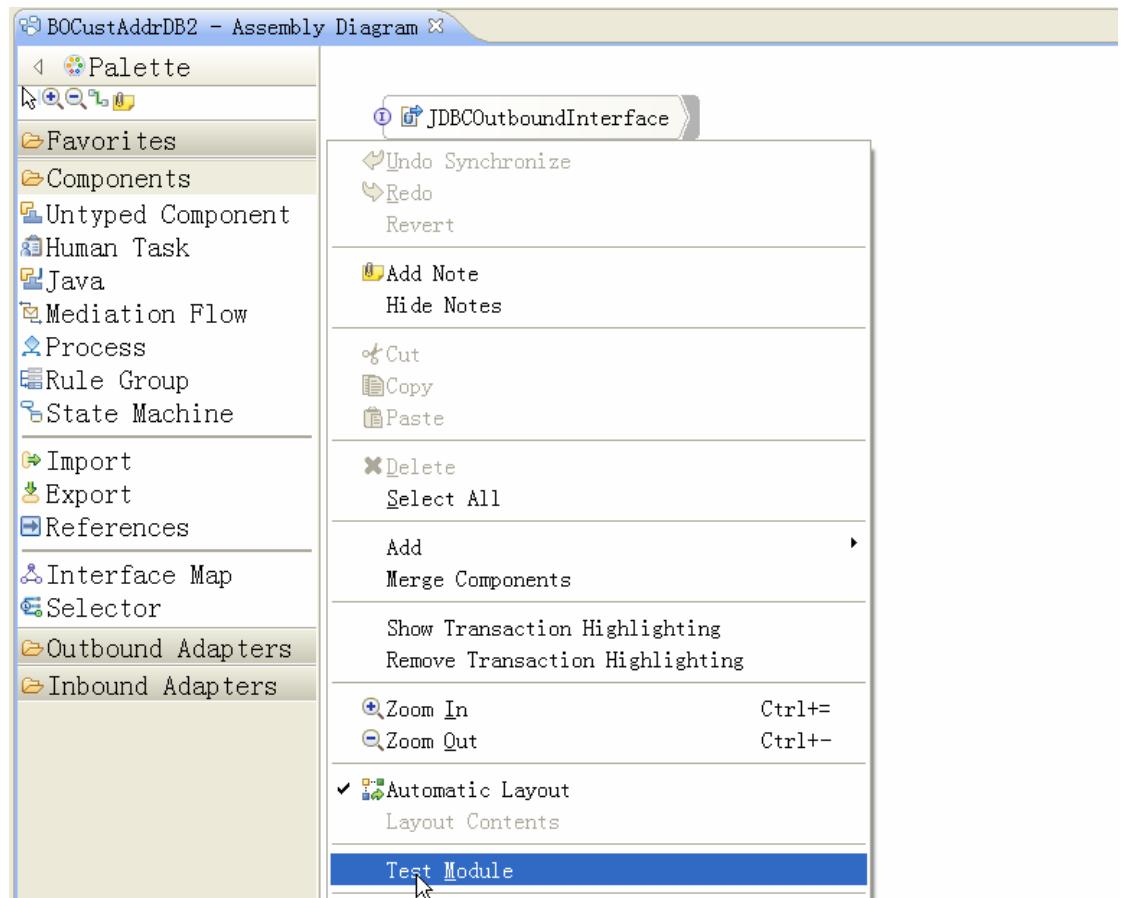


5. Click **Finish..**

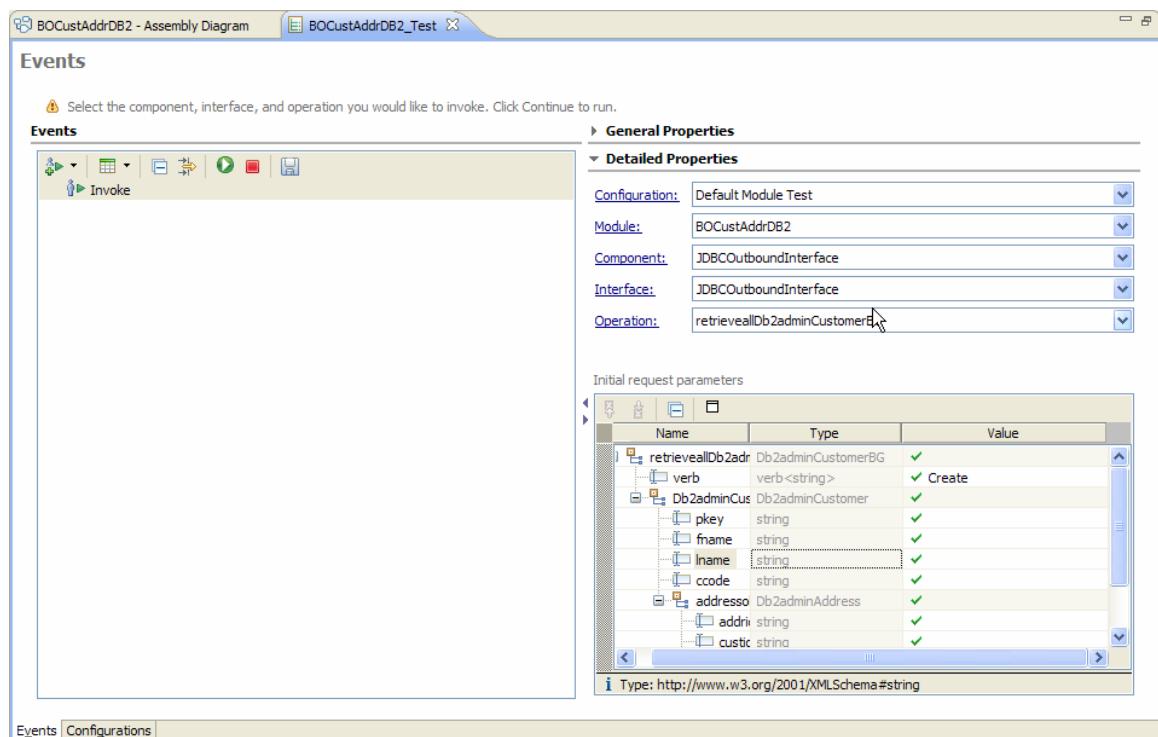
Test the assembled adapter application

Test the assembled adapter application using the WebSphere Integration Developer integration test client by following these steps:

1. Based on the SCA module file, **sca.module**, double-click to display it as an assembly diagram and right-click anywhere within the diagram to bring up the context menu for creating a test module, select **Test Module**.



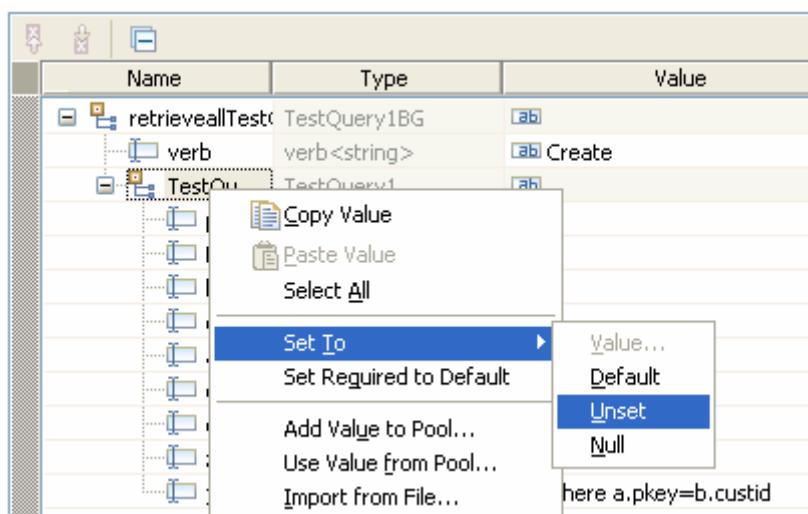
2. A test client is created with input fields.



A test client page is automatically created and displayed and it is ready to accept values before the test execution. The first invocation is

created by default. Subsequent invocation can be created by clicking .

3. Determine the type of operation. The verb prefix indicates the type of operation. For example, one of the operation names for selection, **retrieveallTestQuery1BG**, is a named combination of the verb prefix (retrieveall), business object name (TestQuery1BG), and an abbreviation for Business Graph (BG).
4. Based on the type of operation, double-click the field under **Value** column for the corresponding field under **Name** column to enter an appropriate value. Simply, unset all existing **Value** fields of the relevant business object.



5. To run the test client, click **Continue**  on the top of the page. The result of the test execution is displayed.

Module: [BOCustAddrDB2](#)
 Component: [JDBCOutboundInterface](#)
 Interface: [JDBCOutboundInterface](#)
 Operation: [retrieveallTestQuery1BG](#)

Return parameters:

[Value Editor](#) [XML Source](#)

Name	Type	Value
retrieveallTestQuery1BGOutput	TestQuery1Container	[ab]
TestQuery1 *	TestQuery1[]	[ab]
TestQuery1[0]	TestQuery1	[ab]
pkey	string	[ab] C1
fname	string	[ab] FNAME1
lname	string	[ab] LNAME1
ccode	string	[ab] 001
addrid	string	[ab] A1
custid	string	[ab] C1
city	string	[ab] BEIJING
zipcode	string	[ab] 010
jdbcwhereclause	string	[ab] X
TestQuery1[1]	TestQuery1	[ab]
pkey	string	[ab] C2
fname	string	[ab] FNAME2
lname	string	[ab] LNAME2
ccode	string	[ab] 002
addrid	string	[ab] A2
custid	string	[ab] C2
city	string	[ab] GUANGZHOU
zipcode	string	[ab] 020
jdbcwhereclause	string	[ab] X

Type: <http://www.w3.org/2001/XMLSchema#string>

Clear the sample content

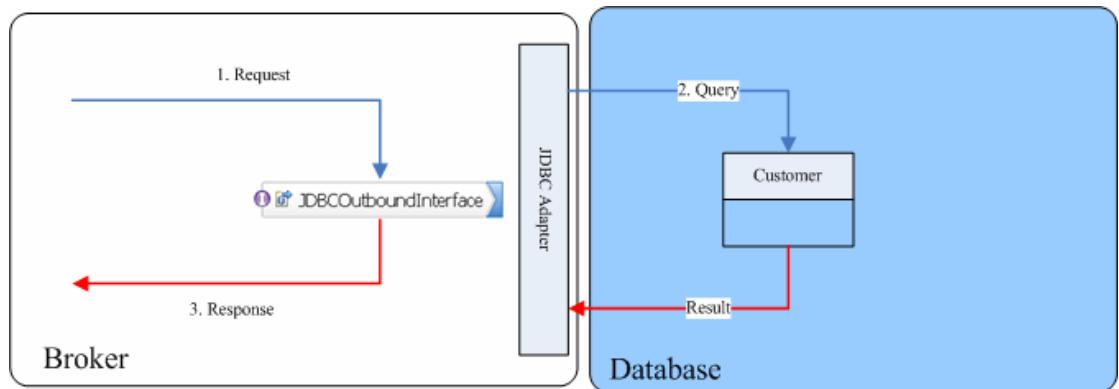
After a record has been created with the WebSphere Integration Developer environment, it can be removed with the Delete operation, determine the key field that uniquely represent the record just created and enter its value.

Chapter 14. Tutorial 13: Checking for the existence of a business object (Oracle)

This tutorial demonstrates how WebSphere Adapter for JDBC 7.0.0.0 checks for the existence of a business object.

About this task

In this scenario, an application SCA component raises an existence test request to the JDBC Outbound Interface. The JDBC adapter executes a SQL query to determine whether specific records exist or not. Finally, JDBC adapter convert the test result to a SDO and give a response to the SCA component. The following figure represents this scenario:



Prepare to run through the tutorial

Extract the sample files

Replicas of the artifacts that you create when using the external service wizard are provided as sample files for your reference. Use these files to verify if the files you create using the external service wizard are correct.

Download the sample zip file and extract it into a directory of your choice (you may want to create a new directory).

Configuration prerequisites

Before configuring the adapter, you must complete the following tasks:

- Create a table
- Create an authentication alias
- Create a data source

Create a table

You must create the following table in the Oracle database before starting the scenario.

```
CREATE TABLE CUSTOMER (
    PKEY VARCHAR2(10) NOT NULL PRIMARY KEY,
    FNAME VARCHAR2( 20 ) ,
    LNAME VARCHAR2( 20 ) ,
    CCODE VARCHAR2(10) );
```

Insert a record into the table you just created.

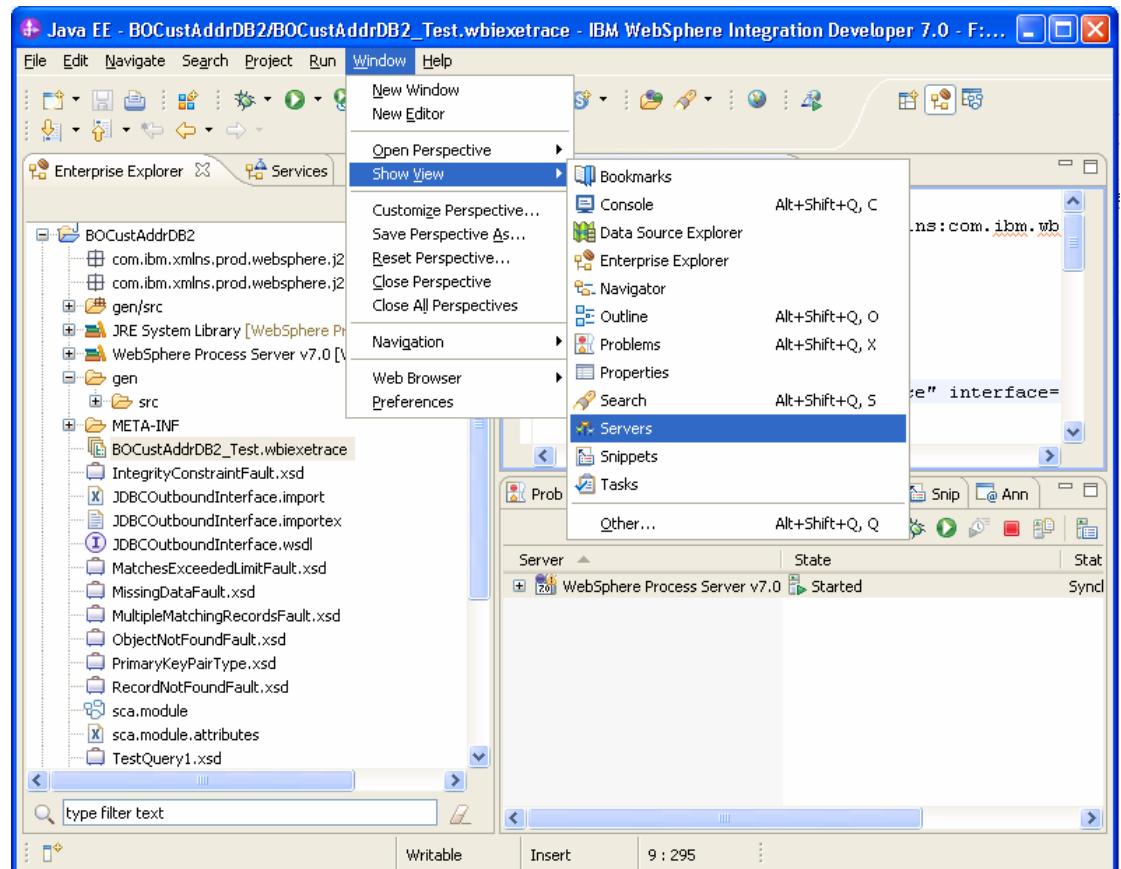
```
insert into customer values ('1000', 'testFname',
'testLname', 'testCcode')
```

Create an authentication alias

The authentication alias needs to be set because the data source created in the next section uses the username and password set in the authentication alias to connect to the database.

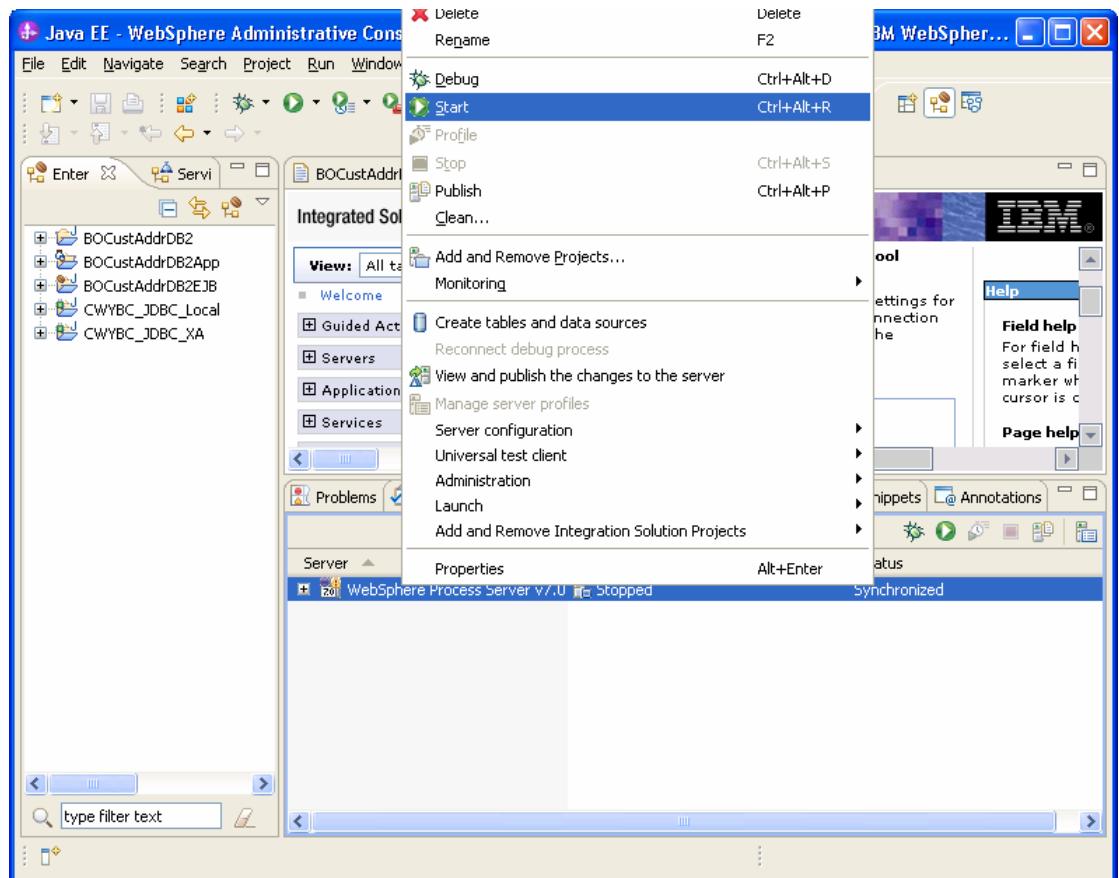
Follow these steps to set the authentication alias in the WebSphere Process Server administrative console.

1. In WebSphere Integration Developer, switch to the **Servers** view by selecting **Window > Show View > Servers**.

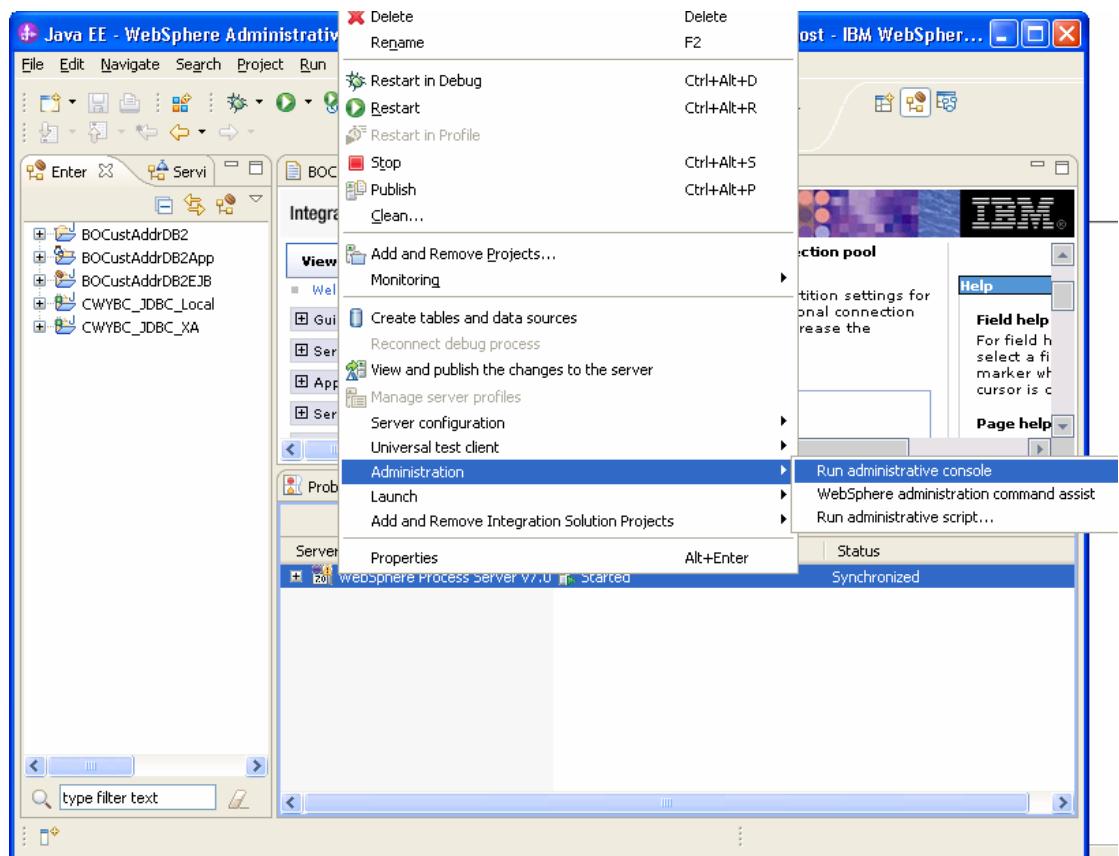


2. In the **Servers** view, right-click the server that you want to start and select **Start**.

WebSphere software

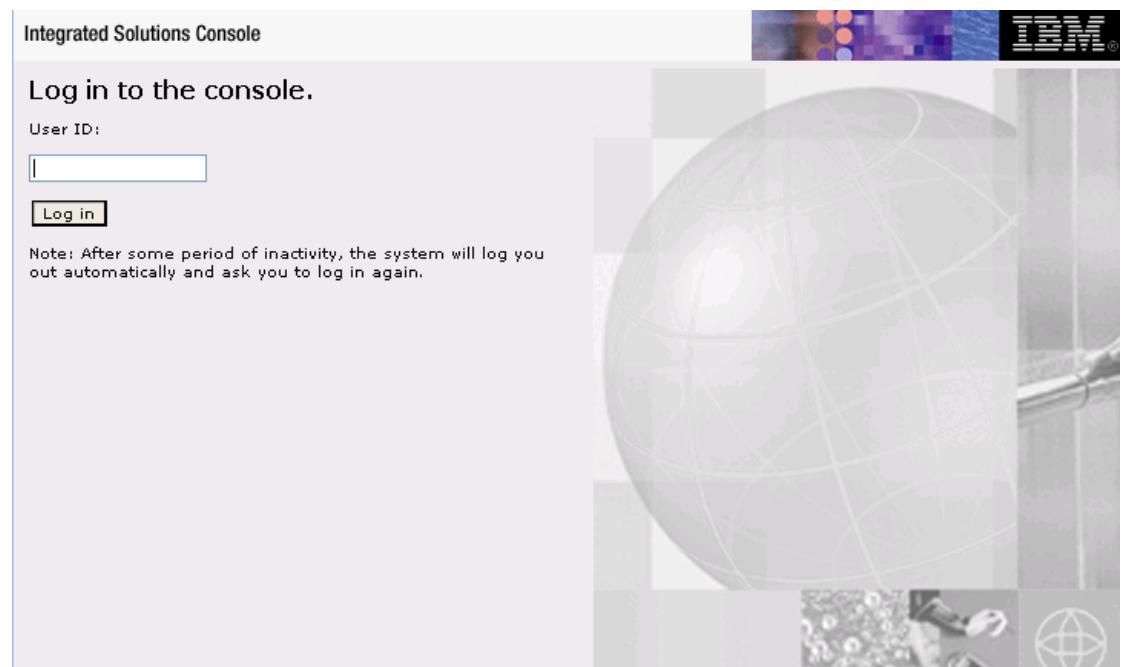


- After the server is started, right-click the server, and select **Administration > Run administrative console**.



WebSphere software

4. Log on to the administrative console.



5. Click **Security → Global security**.



6. Under **Java Authentication and Authorization Service**, click **J2C authentication Data**.

Integrated Solutions Console Welcome admin Help | Logout IBM.

Secure administration, applications, and infrastructure Close page

Secure administration, applications, and infrastructure

The application serving environment is completely secured when administration is restricted. The applications and the infrastructure supports the administration and applications also are secured.

Configuration

Security Configuration Wizard Security Configuration Report

Administrative security

Enable administrative security [Administrative User Roles](#) [Administrative Group Roles](#)

Application security

Enable application security

Java 2 security

Use Java 2 security to restrict application access to local resources Warn if applications are granted custom permissions Restrict access to resource authentication data

User account repository

Current realm definition Local operating system

Authentication

Use domain-qualified user names
 Web security
 RMI/IOP security
 Java Authentication and Authorization Service
 ■ Application logins
 ■ System logins
 ■ J2C authentication data
 Authentication *Specifies a list of user identities and passwords for Java(TM) 2 connector security to use.*

External authorization providers
 Custom properties

A list of existing aliases is displayed.

Integrated Solutions Console Welcome admin Help | Logout IBM.

Secure administration, applications, and infrastructure Close page

Secure administration, applications, and infrastructure > JAAS - J2C authentication data

Specifies a list of user identities and passwords for Java(TM) 2 connector security to use.

Help

Field help
For field help information, select field label or list item when the help cursor appears.

Page help
More information about this page

Preferences

New	Delete	Select	Alias	User ID	Description
		<input type="checkbox"/>	BSpace JDBC Alias	TEST	Business Space Authorization Alias
		<input type="checkbox"/>	CommonEventInfrastructureJMSAuthAlias	CEI	Authentication alias for the Common Event Infrastructure JMS Topics and Queues
		<input type="checkbox"/>	SCA Auth Alias	SCA	This is the alias used by SCA to login to a secured SIBus
		<input type="checkbox"/>	localhostNode01Cell/nlNode01/server1/EventAuthDataAliasDerby		Derby authentication alias for the Event Server

- Click **New** to create a new authentication entry. Type the alias name, and username and password to connect to the database. Click **OK**.

The screenshot shows the 'Global security' configuration page in the WebSphere Integrated Solutions Console. The left sidebar lists various administrative categories like 'Welcome', 'Guided Activities', 'Servers', etc. Under 'Security', 'Global security' is selected. The main panel displays a form for creating a new authentication alias. The 'Alias' field is set to 'node1/Oracle'. Other fields include 'User ID' (sample) and 'Password' (*****). A 'Description' field is also present. The right side of the screen includes a 'Help' section with links for 'Field help' and 'Page help'.

You have created an authentication alias that will be used to configure the adapter properties.

The screenshot shows a table listing six authentication aliases. The columns are 'Name' (checkbox), 'Name', 'User ID', and 'Description'. The aliases listed are: 'BSpace JDBC Alias' (wbiuser), 'CommonEventInfrastructureJMSAuthAlias' (wbiuser), 'SCA Auth Alias' (wbiuser), 'localhostNode01Cell/nNode01/server1/EventAuthDataAliasDerby' (empty), 'nNode01/luweiqin' (luweiqin), and 'nNode01/node1/Oracle' (sample). The total count is shown as 6 at the bottom of the table.

	Name	User ID	Description
<input type="checkbox"/>	BSpace JDBC Alias	wbiuser	BS A A
<input type="checkbox"/>	CommonEventInfrastructureJMSAuthAlias	wbiuser	A a C E I Ji a
<input type="checkbox"/>	SCA Auth Alias	wbiuser	T a S a S
<input type="checkbox"/>	localhostNode01Cell/nNode01/server1/EventAuthDataAliasDerby		D a a E
<input type="checkbox"/>	nNode01/luweiqin	luweiqin	
<input type="checkbox"/>	nNode01/node1/Oracle	sample	C o o
Total 6			

Create a data source

Create a data source in WebSphere Process Server, which the adapter will use to connect to the database. This data source will be used in generating the artifacts for the module.

Note: This tutorial will use Oracle as the database and the Oracle thin driver, ojdbc6.jar.

Here are the steps to create the data source in the WebSphere Process Server administrative console.

1. In the administrative console, select **Environment → WebSphere variables**

WebSphere software



2. On the right, click **ORACLE_JDBC_DRIVER_PATH** and specify the path of the ojdbc6.jar file in the **Value** field. Click **OK**.

The dialog box shows the following settings:

- * Name: **ORACLE_JDBC_DRIVER_PATH**
- Value: **F:\DBDriver\Oracle10**
- Description: **The directory that contains the Oracle thin or oci8 JDBC Driver.**

Buttons at the bottom: **Apply**, **OK**, **Reset**, **Cancel**.

The variable is added and appears in the list.

The table lists various environment variables and their values:

JAVA_HOME	<code>#{WAS_INSTALL_ROOT}/java</code>
JVM_CACHE	
LOCALHOST_NAME	localhost
LOG_ROOT	<code>#{USER_INSTALL_ROOT}/logs</code>
MICROSOFT_JDBC_DRIVER_NATIVEPATH	
MICROSOFT_JDBC_DRIVER_PATH	
MQ_INSTALL_ROOT	<code>#{WAS_INSTALL_ROOT}/lib/W</code>
ORACLE_JDBC_DRIVER_PATH	<code>F:\DBDriver\Oracle10</code>
OS400_NATIVE_JDBC40_DRIVER_PATH	
OS400_NATIVE_JDBC_DRIVER_PATH	
OS400_TOOLBOX_JDBC_DRIVER_PATH	
SCA_BUS_ID	localhostNode01Cell
SERVER_LOG_ROOT	<code>#{LOG_ROOT}/server1</code>
SYBASE_JDBC_DRIVER_PATH	

3. Select **Resources → JDBC -> JDBC Providers**.

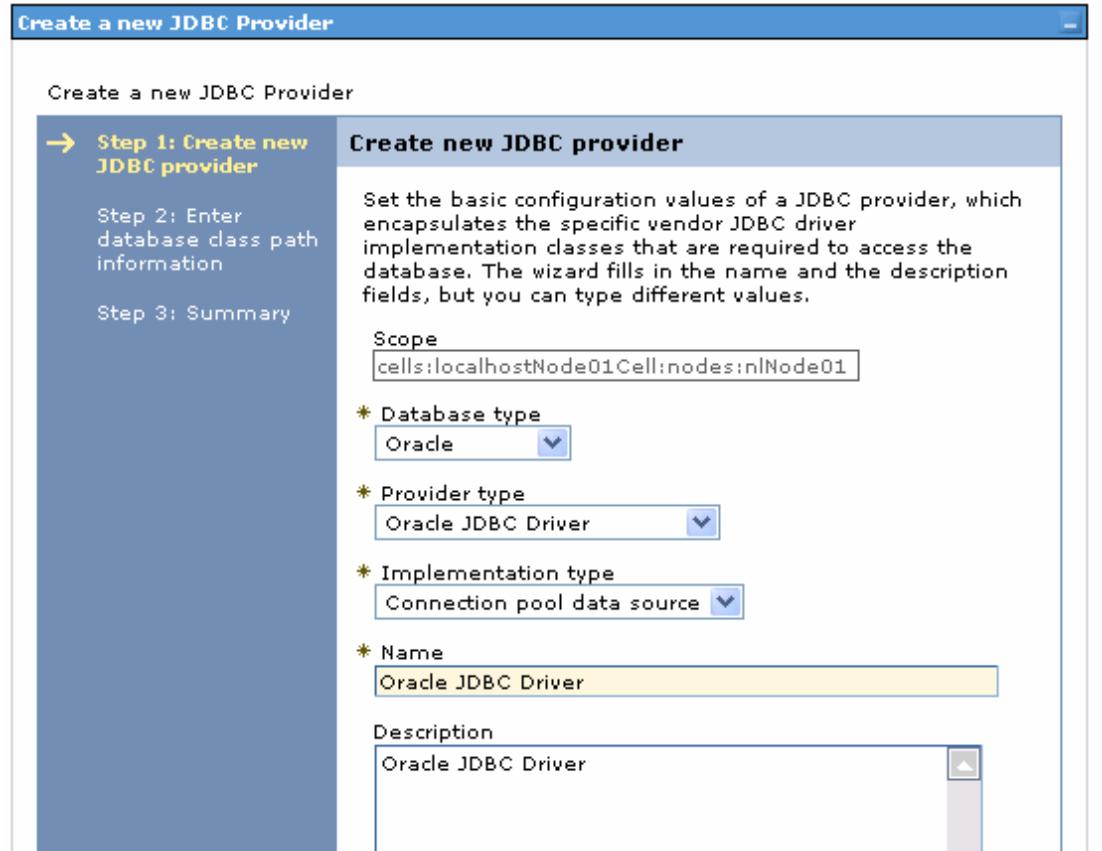


- Click **New** in the JDBC providers window.

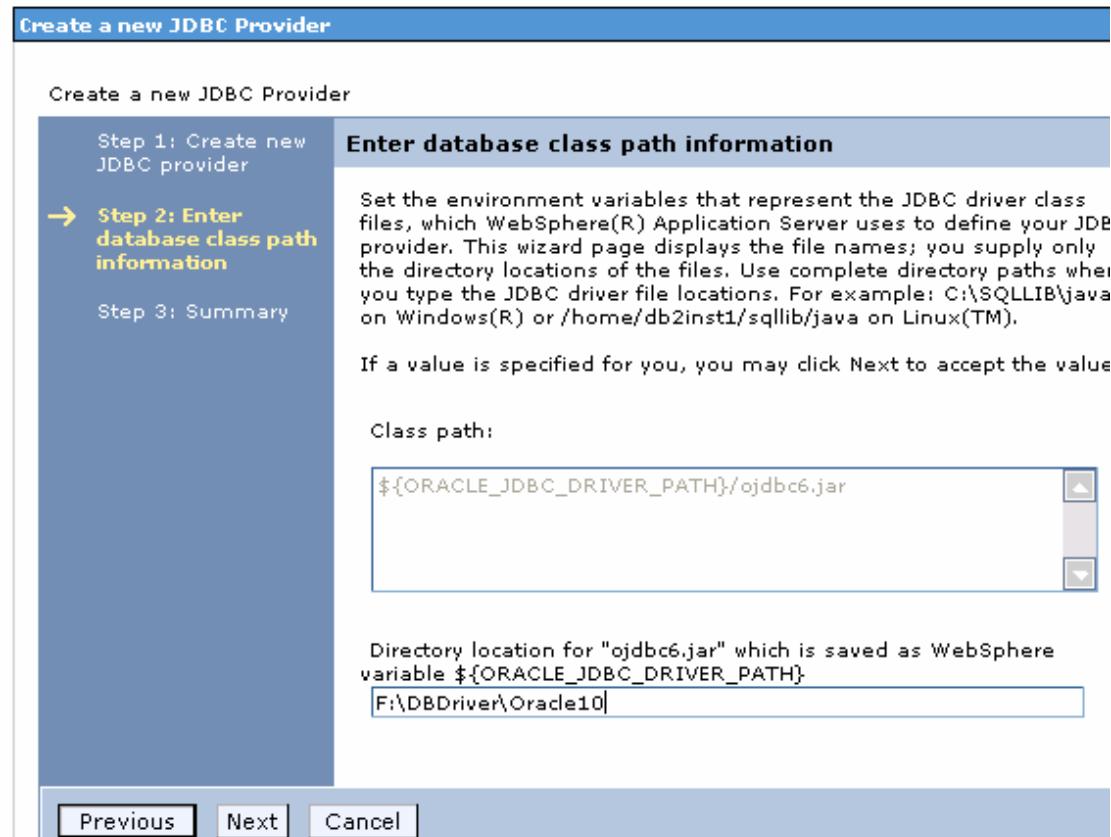
The screenshot shows the 'JDBC providers' page in the WebSphere Application Server V4 Integrated Solutions Console. The page title is 'JDBC providers'. A help panel on the right provides links for 'Field help', 'Page help', and 'Command Assistance'. The main content area shows a table with the following data:

Select	Name	Scope	Description
<input type="checkbox"/>	Derby JDBC Provider (XA)	Node=nINode01	JDBC Provider for WPS/WESB

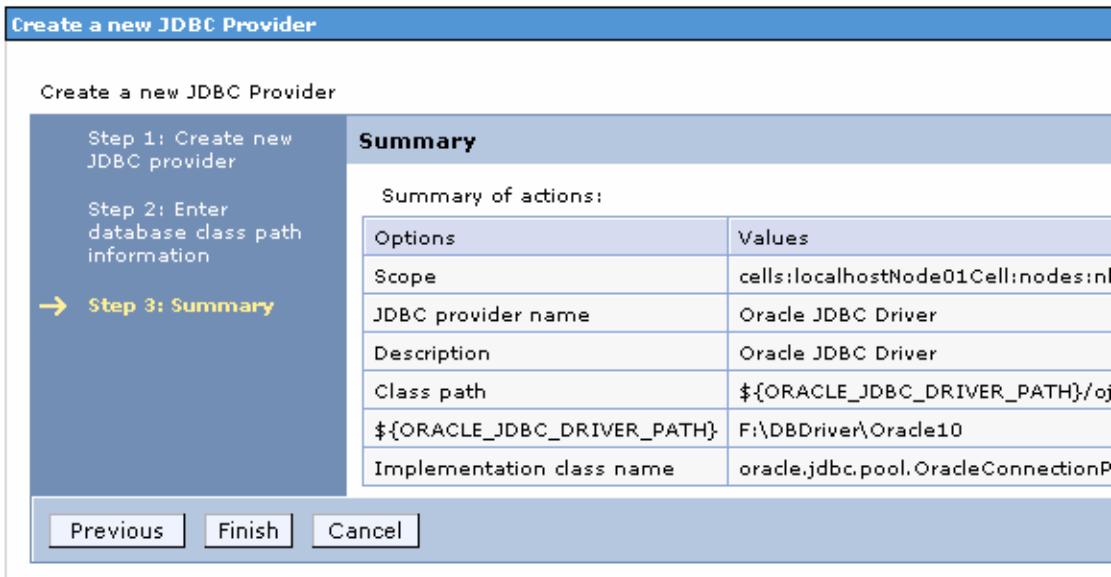
- Click **New**. Select the Oracle database with a connection pool data source for the Oracle JDBC driver. Click **Next**.



6. In the Enter database classpath information page, enter the following value for the **Class path** field:
\$(ORACLE_JDBC_DRIVER_PATH)/ojdbc6.jar, where
\$(ORACLE_JDBC_DRIVER_PATH) is library path for the run time.
Because you have added the ojdbc6.jar file to this path, you must specify that path here.
7. Click **Next**.



8. Click **Finish**.



9. Under **Additional Properties**, select **Data sources**. Click **New**.

The screenshot shows the 'JDBC providers' interface with the 'Oracle JDBC Driver' selected. The 'Data sources' tab is active. A message at the top explains that this page allows editing datasource settings for the selected provider. Below the message is a toolbar with 'New', 'Delete', 'Test connection', and 'Manage state...'. Underneath is a toolbar with icons for creating, deleting, and managing datasources. A table header row includes columns for Select, Name, JNDI name, Scope, Provider, Description, and Category. The table body shows one entry: 'None' under 'Name' and 'Total 0' below it.

10. Type any value in the **JNDI name** field, and select the authentication alias "OracleDS" that you created earlier from the **Component-managed authentication alias and XA recovery authentication alias** list. Click **Next**.

The screenshot shows the 'Create a data source' wizard. The left sidebar lists steps: Step 1: Enter basic data source information (highlighted with a yellow arrow), Step 2: Enter database specific properties for the data source, Step 3: Setup security aliases, and Step 4: Summary. The main panel is titled 'Enter basic data source information' and contains instructions: 'Set the basic configuration values of a datasource for association with your JDBC provider. A datasource supplies the physical connections between the application server and the database.' It also states a requirement: 'Requirement: Use the Datasources (WebSphere(R) Application Server V4) console pages if your applications are based on the Enterprise JavaBeans(TM) (EJB) 1.0 specification or the Java(TM) Servlet 2.2 specification.' The configuration fields include 'Scope' (set to 'cells:localhostNode01Cell:nodes:n1Node01'), 'JDBC provider name' (set to 'Oracle JDBC Driver'), and two required fields: 'Data source name' (set to 'Oracle JDBC Driver DataSource') and 'JNDI name' (set to 'OracleDS'). At the bottom are 'Next' and 'Cancel' buttons.

11. Provide the appropriate URL value and select a data store helper class name from the **Data store helper class name** list as shown in the following figure. Click **Next**.

Create a data source

Step 1: Enter basic data source information

→ Step 2: Enter database specific properties for the data source

Step 3: Setup security aliases

Step 4: Summary

Enter database specific properties for the data source

Set these database-specific properties, which are required by the database vendor JDBC driver to support the connections that are managed through the datasource.

Name	Value
* URL	jdbc:oracle:thin@9.181.84.13
* Data store helper class name	Oracle10g data store helper
<input checked="" type="checkbox"/> Use this data source in container managed persistence (CMP)	

Previous Next Cancel

12. In the Setup security aliases window, configure the aliases.

Create a data source

Step 1: Enter basic data source information

Step 2: Enter database specific properties for the data source

→ Step 3: Setup security aliases

Step 4: Summary

Setup security aliases

Select the authentication values for this resource.

Component-managed authentication alias
nlNode01/node1/Oracle

Mapping-configuration alias
(none)

Container-managed authentication alias
(none)

Note: You can create a new J2C authentication alias by accessing one of the following links. Clicking on a link will cancel the wizard and your current selections will be lost.

[Global J2C authentication alias](#)
[Security domains](#)

Previous Next Cancel

13. In the Summary page, review the values entered for the data source and click **Finish**.

Create a data source

Step 1: Enter basic data source information Step 2: Enter database specific properties for the data source Step 3: Setup security aliases → Step 4: Summary	Summary Summary of actions: <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="background-color: #d9e1f2;">Options</th> <th style="background-color: #d9e1f2;">Values</th> </tr> </thead> <tbody> <tr> <td>Scope</td> <td>cells:localhostNode01Cell:nodes:nlNode01</td> </tr> <tr> <td>Data source name</td> <td>Oracle JDBC Driver DataSource</td> </tr> <tr> <td>JNDI name</td> <td>OracleDS</td> </tr> <tr> <td>Select an existing JDBC provider</td> <td>Oracle JDBC Driver</td> </tr> <tr> <td>Implementation class name</td> <td>oracle.jdbc.pool.OracleConnectionPoolDataSource</td> </tr> <tr> <td>URL</td> <td>jdbc:oracle:thin:@9.181.84.136:1521:ord</td> </tr> <tr> <td>Data store helper class name</td> <td>com.ibm.websphere.rsadapter.Oracle10gDataStoreHelper</td> </tr> <tr> <td>Use this data source in container managed persistence (CMP)</td> <td>true</td> </tr> <tr> <td>Component-managed authentication alias</td> <td>nlNode01/node01/Oracle</td> </tr> <tr> <td>Mapping-configuration alias</td> <td>(none)</td> </tr> <tr> <td>Container-managed authentication alias</td> <td>(none)</td> </tr> </tbody> </table>	Options	Values	Scope	cells:localhostNode01Cell:nodes:nlNode01	Data source name	Oracle JDBC Driver DataSource	JNDI name	OracleDS	Select an existing JDBC provider	Oracle JDBC Driver	Implementation class name	oracle.jdbc.pool.OracleConnectionPoolDataSource	URL	jdbc:oracle:thin:@9.181.84.136:1521:ord	Data store helper class name	com.ibm.websphere.rsadapter.Oracle10gDataStoreHelper	Use this data source in container managed persistence (CMP)	true	Component-managed authentication alias	nlNode01/node01/Oracle	Mapping-configuration alias	(none)	Container-managed authentication alias	(none)
Options	Values																								
Scope	cells:localhostNode01Cell:nodes:nlNode01																								
Data source name	Oracle JDBC Driver DataSource																								
JNDI name	OracleDS																								
Select an existing JDBC provider	Oracle JDBC Driver																								
Implementation class name	oracle.jdbc.pool.OracleConnectionPoolDataSource																								
URL	jdbc:oracle:thin:@9.181.84.136:1521:ord																								
Data store helper class name	com.ibm.websphere.rsadapter.Oracle10gDataStoreHelper																								
Use this data source in container managed persistence (CMP)	true																								
Component-managed authentication alias	nlNode01/node01/Oracle																								
Mapping-configuration alias	(none)																								
Container-managed authentication alias	(none)																								

14. Click **Save** to save the changes.

JDBC providers

Messages

⚠ Changes have been made to your local configuration. You can:

- [Save](#) directly to the master configuration.
- [Review](#) changes before saving or discarding.

⚠ The server may need to be restarted for these changes to take effect.

[JDBC providers > Oracle JDBC Driver > Data sources](#)

Use this page to edit the settings of a datasource that is associated with your selected JDBC provider. The datasource object supplies your application with connections for accessing the database. Learn more about this task in a [guided activity](#). A guided activity provides a list of task steps and more general information about the topic.

Preferences

New	Delete	Test connection	Manage state...				
Select	Name ▾	JNDI name ▾	Scope ▾	Provider ▾	Description ▾	Category ▾	
You can administer the following resources:							
<input type="checkbox"/>	Oracle JDBC Driver DataSource	OracleDS	Node=nlNode01	Oracle JDBC Driver	New JDBC Datasource		
Total 1							

15. Select the data source you just created and click **Test connection**.

JDBC providers > Oracle JDBC Driver > Data sources

Select	Name	JNDI name	Scope	Provider	Description	Category
<input checked="" type="checkbox"/>	Oracle JDBC Driver DataSource	OracleDS	Node=nINode01	Oracle JDBC Driver	New JDBC Datasource	

Total 1

The connection should succeed as indicated by the message shown in the following figure. If you experience problems with the test connection, refer to the "Troubleshooting" section.

Messages

The test connection operation for data source Oracle JDBC Driver DataSource on server server1 at node nINode01 was successful.

JDBC providers > Oracle JDBC Driver > Data sources

Select	Name	JNDI name	Scope	Provider	Description	Category
<input type="checkbox"/>	Oracle JDBC Driver DataSource	OracleDS	Node=nINode01	Oracle JDBC Driver	New JDBC Datasource	

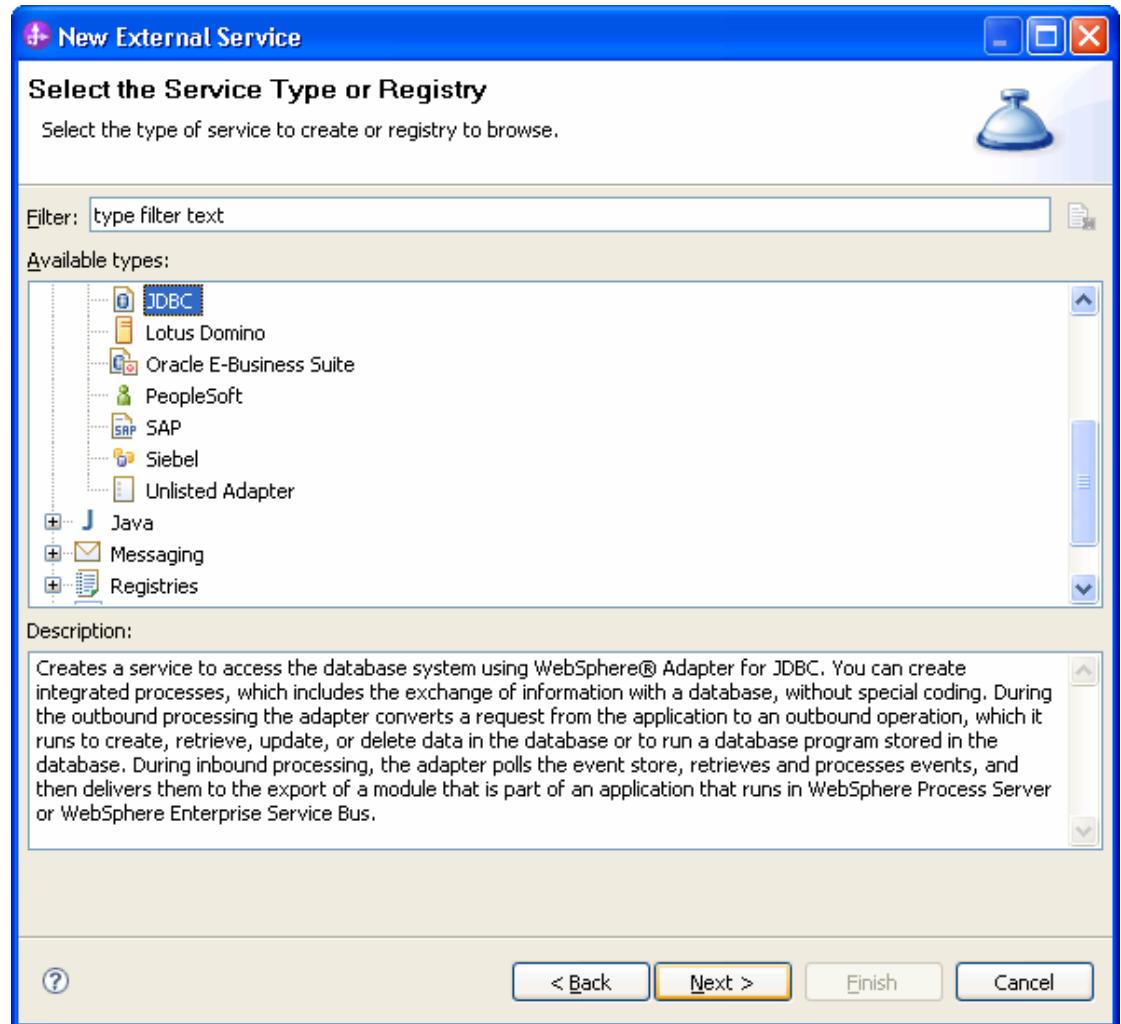
Total 1

The data source is created and it will be used by the adapter to connect to the database.

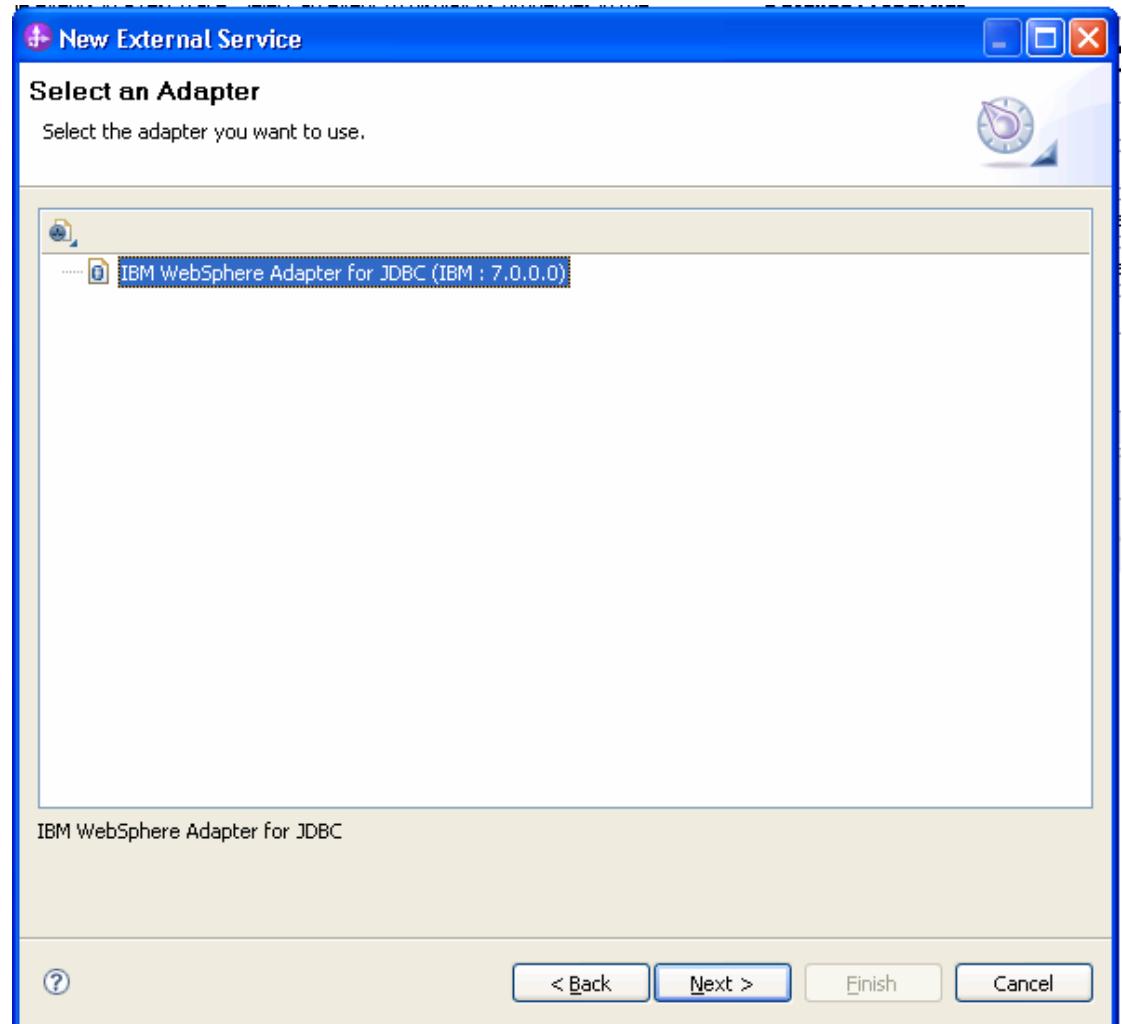
Configure the adapter for outbound processing

Run the external service wizard to specify business objects, services, and configuration details.

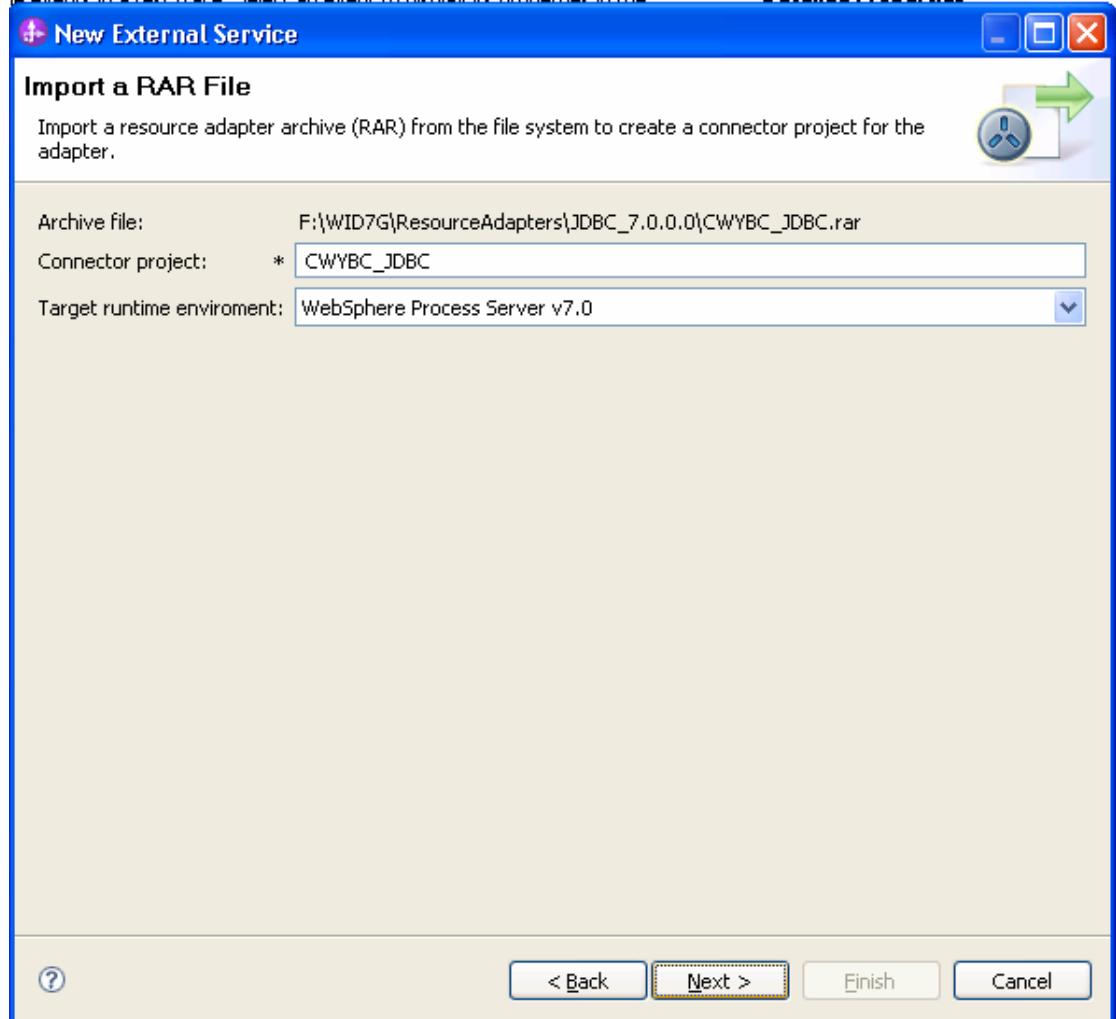
1. Switch to the Business Integration Perspective in WebSphere Integration Developer by selecting **Window -> Open Perspective Business Integration**.
2. Start the external service wizard by selecting **File-> New -> External Service**.
3. In the **Available Types** area, select **Adapters > JDBC** and click **Next**.



4. Select the **IBM WebSphere Adapter for JDBC (IBM: 7.0.0.0)** and click **Next**.

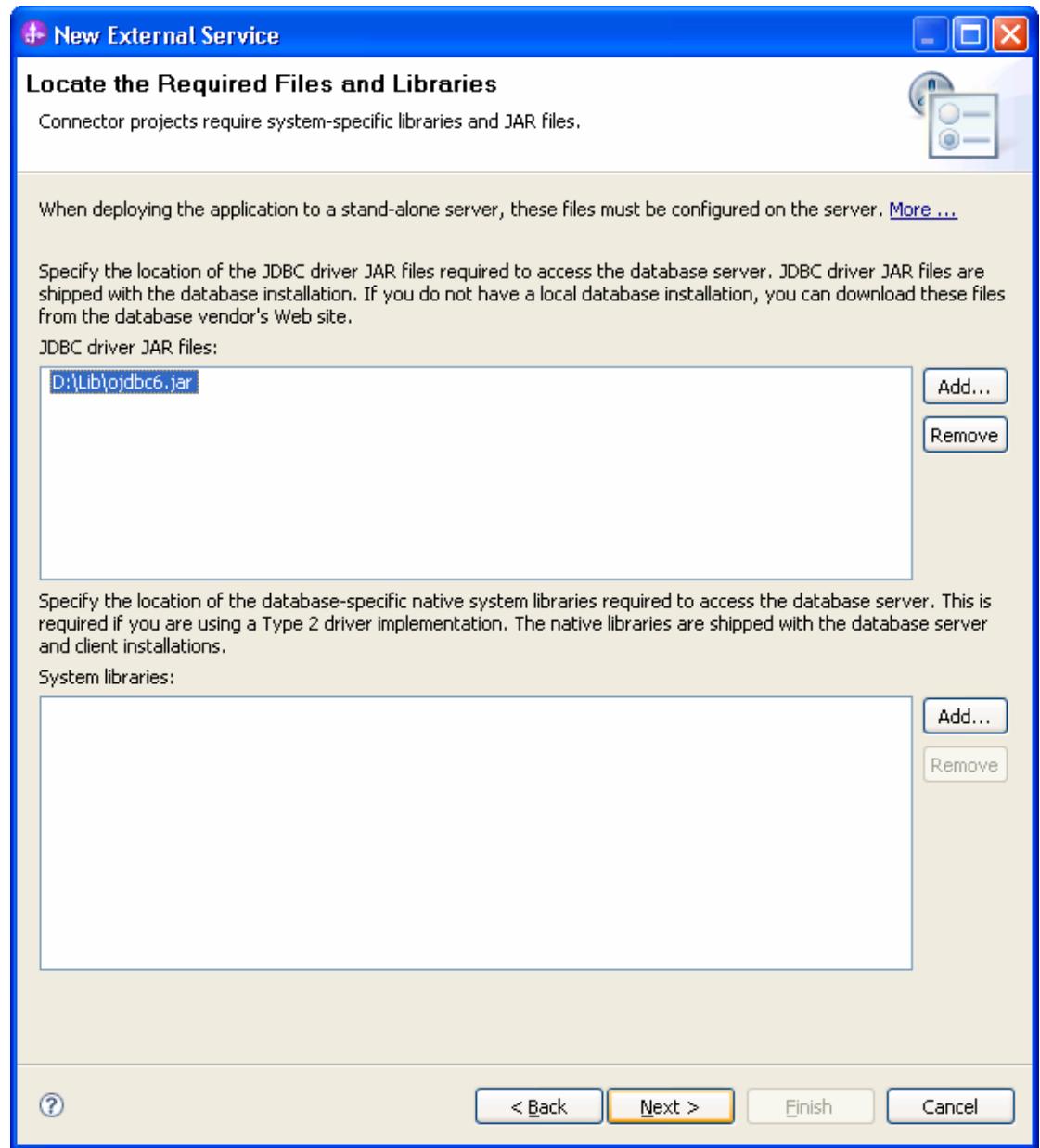


5. In the **Connector project** field enter **CWYBC_JDBC**.
6. In the **Target runtime environment** field, select appropriate runtime and click **Next**.

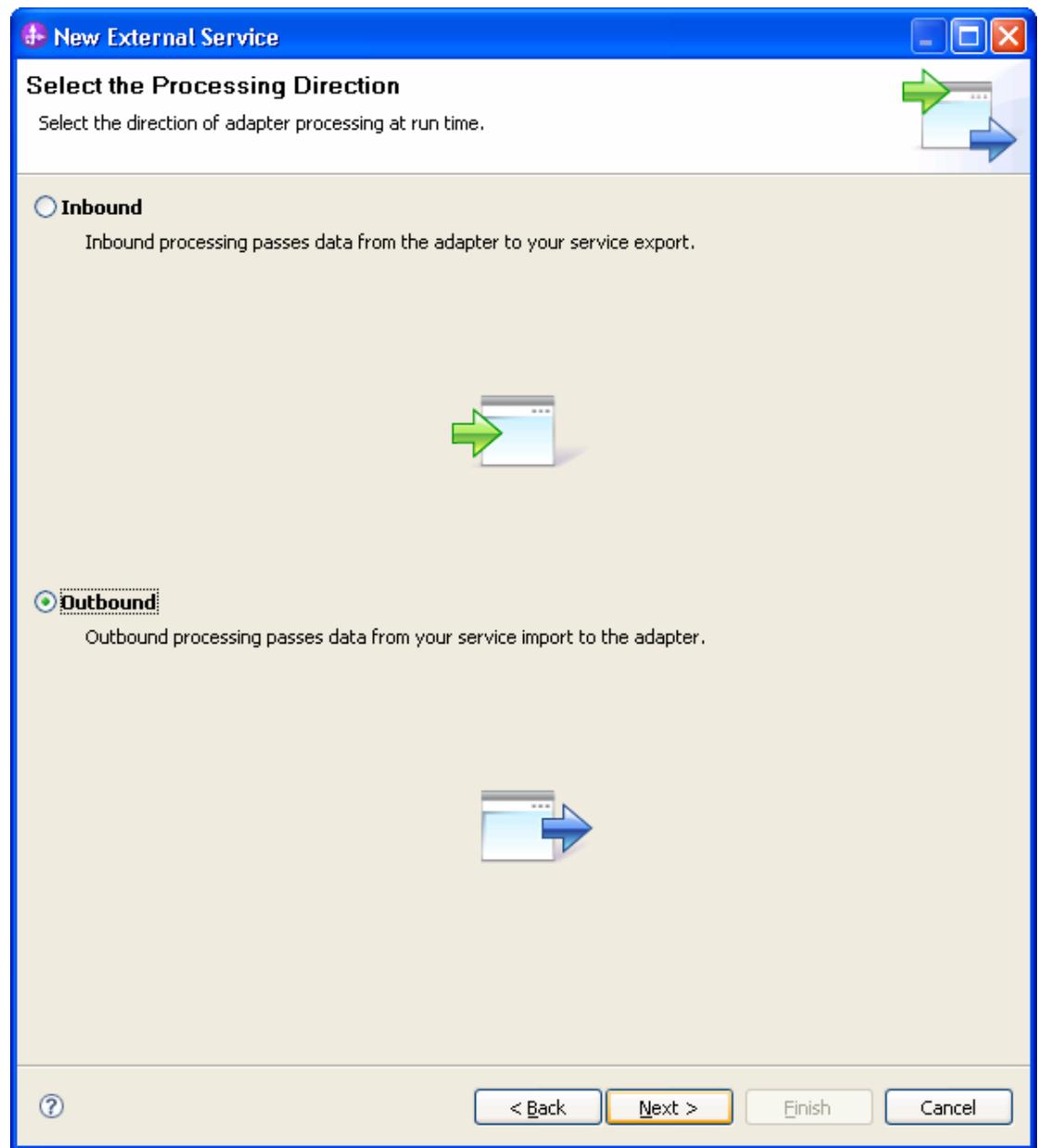


WebSphere software

7. In the **JDBC driver JAR files** field, click **Add** to add the JDBC driver class to connect to the database. Browse to select the driver JAR file and click **Next**.



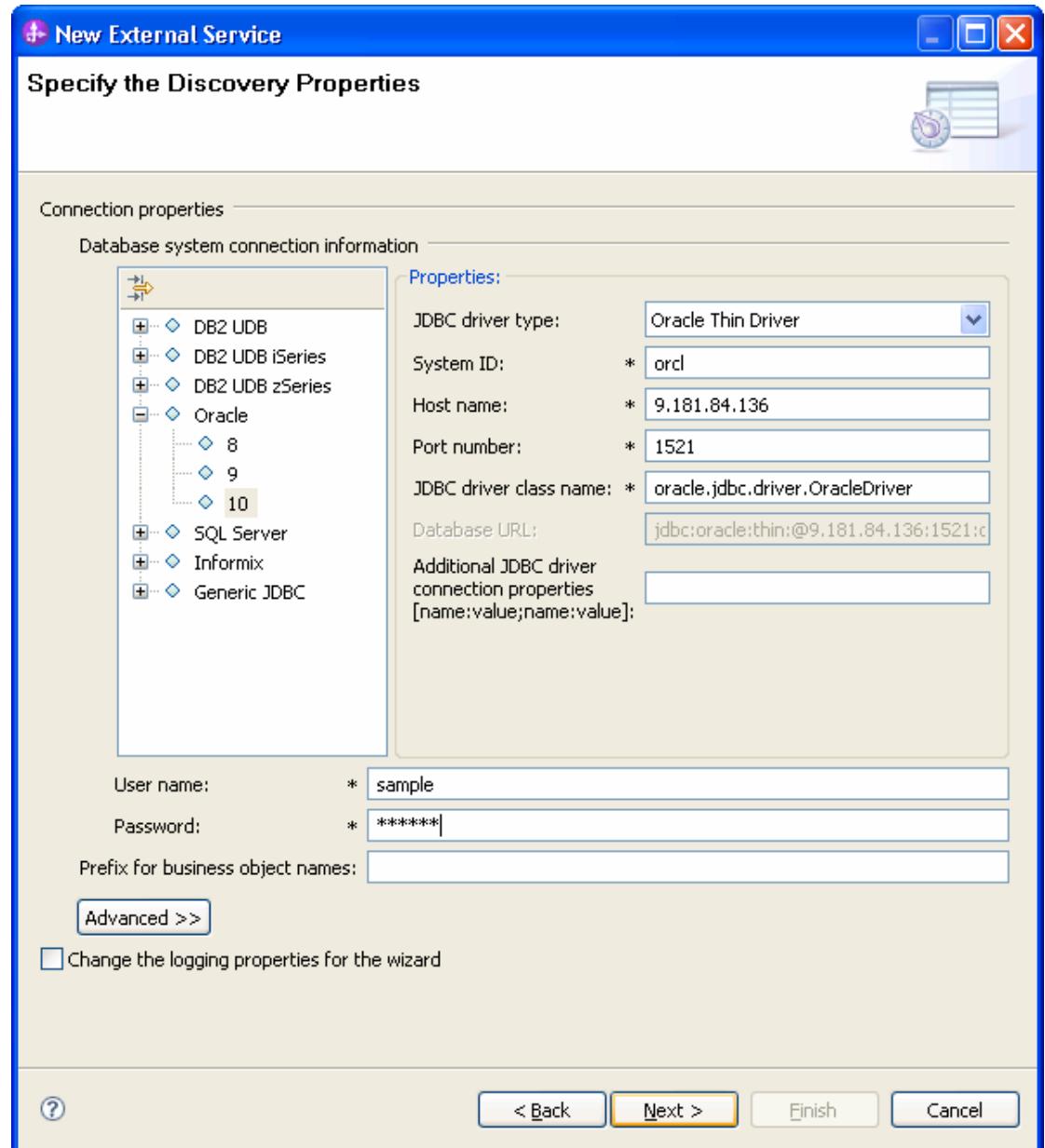
8. Select **Outbound** and click **Next**.



Set connection properties for the external service wizard

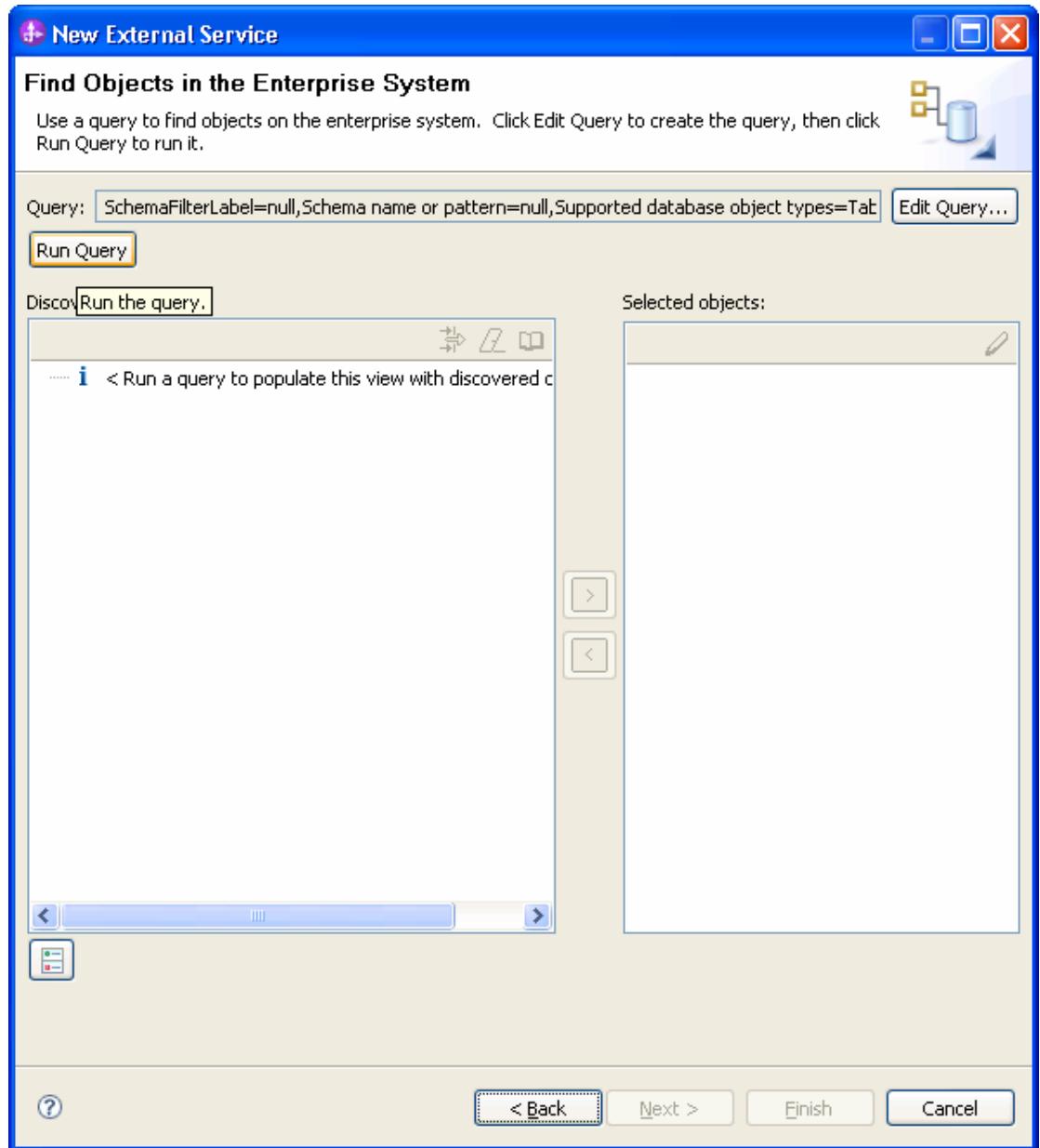
To connect to the Oracle database:

1. Expand **Oracle** from **Database system connection information** then select **10**.
2. Enter values in the **System ID**, **Host name**, **User name** and **Password** fields, and click **Next**.

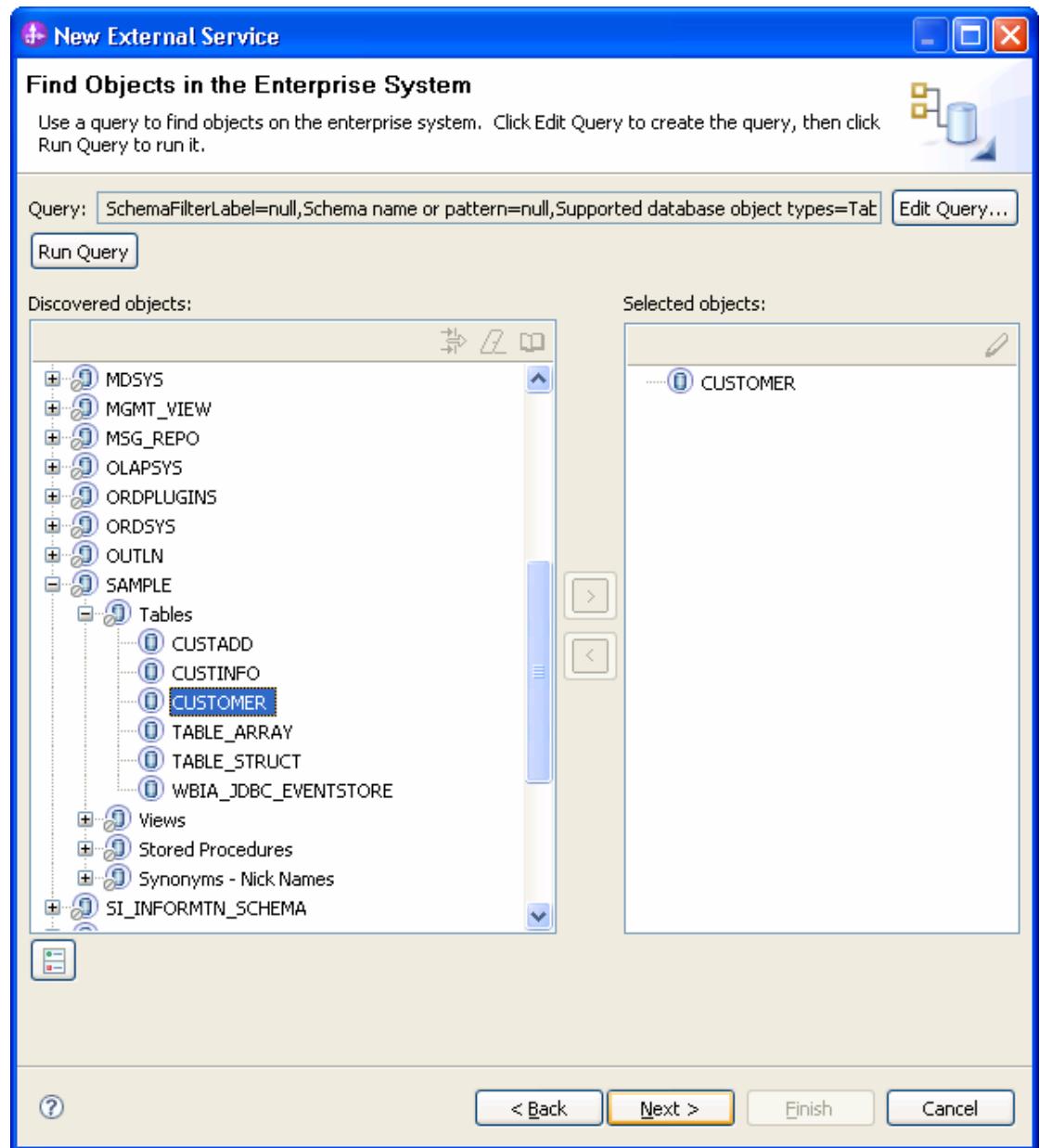


Select the business objects to be used with the adapter

1. In the Find Objects in Enterprise System window, click **Run Query**.



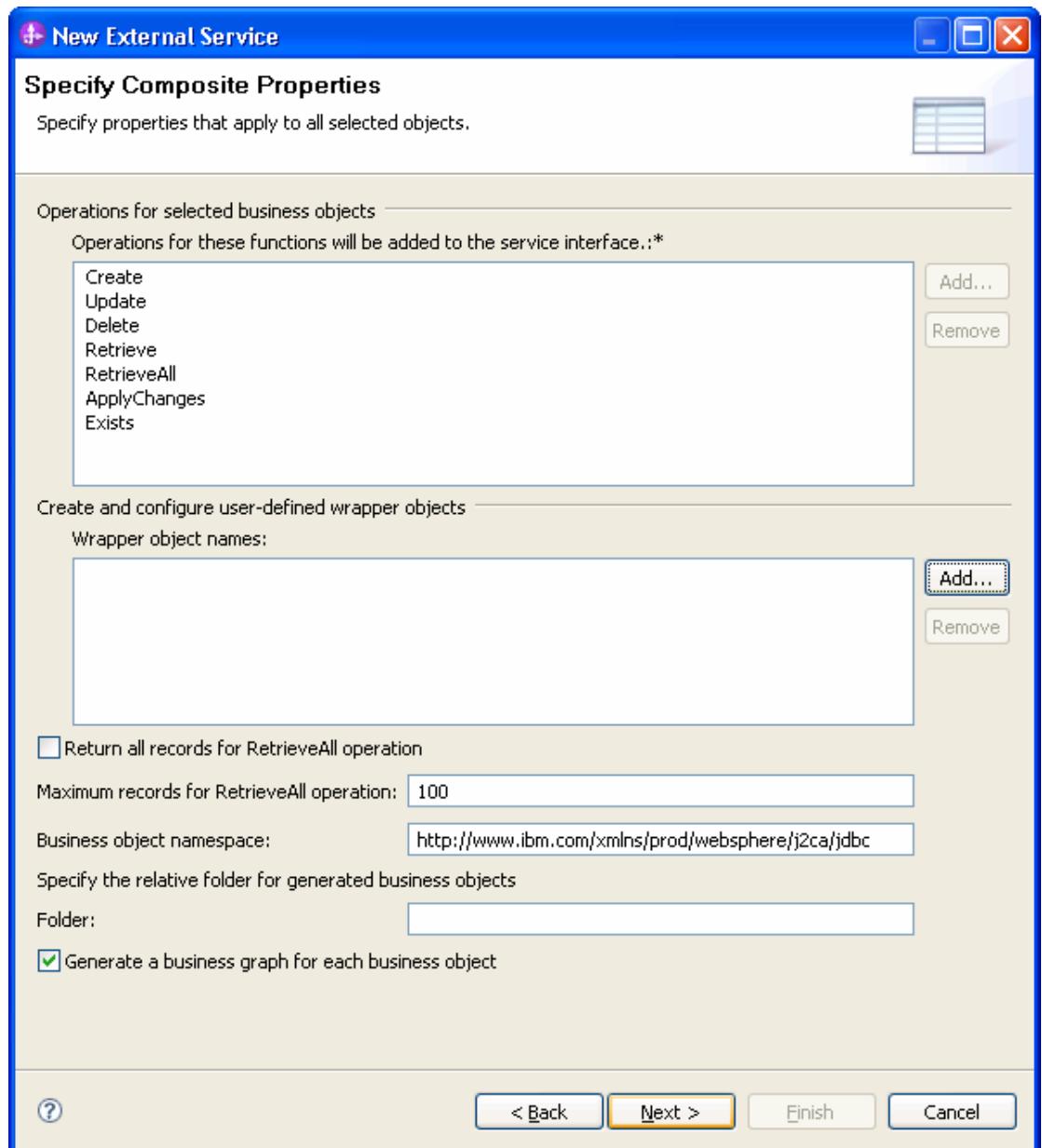
2. In the Discovered objects pane, select the JABDULLA (for this tutorial only) node and expand it. Expand **Tables**, select the CUSTOMER table and click . Click **Next**.



Generate business object definitions and related artifacts

Follow these steps to generate the business object definitions.

1. In the Specify Composite Properties window, accept the default values for the **Maximum records for RetrieveAll operation** and **Business object namespace** fields. Leave the **Generate business graph for each business object** check box selected and click **Next**.



2. In the Specify the Service Generation and Deployment Properties window, perform the following steps:
 - a) Select **Other** for security options under **Deployment properties**.
 - b) Clear the **Join the global transaction** check box.
 - c) Select **Specify predefined connection pool DataSource** from the **Database connection information** list.
 - d) Enter **OracleDS** in the **Connection pool DataSource JNDI Name** field, and click **Next**.

New External Service

Specify the Service Generation and Deployment Properties

Specify properties for generating the service and running it on the server.

Service Operations
To modify the names, or add a description to the operations to be generated in the interface file, click Edit Operations... [Edit Operations...](#)

Deployment Properties
How do you want to specify the security credentials?
 Using an existing JAAS alias (recommended)
A Java Authentication and Authorization Services (JAAS) alias is the preferred method.
J2C authentication data entry:
 Using security properties from the managed connection factory
The properties will be stored as plain text; no encryption is used.
User name:
Password:
 Other
Use if no security is required or will be handled by the EIS system, or the RAR will be deployed on the server and security will be specified by the properties in the JNDI lookup name.

The quality of service that is used to join the transaction provides a higher degree of data integrity, especially when a failure occurs. To participate in a global transaction, a predefined XA DataSource or XA database connection information must be specified in the connection properties. [More ...](#)

Join the global transaction

Deploy connector project:

Specify the settings used to connect to JDBC at run time:

Connection settings:

Connection Properties
To join a global transaction, specify a predefined XA datasource or XA database connection information. When not joining a global transaction, either the XA connection information or the local connection information can be specified.

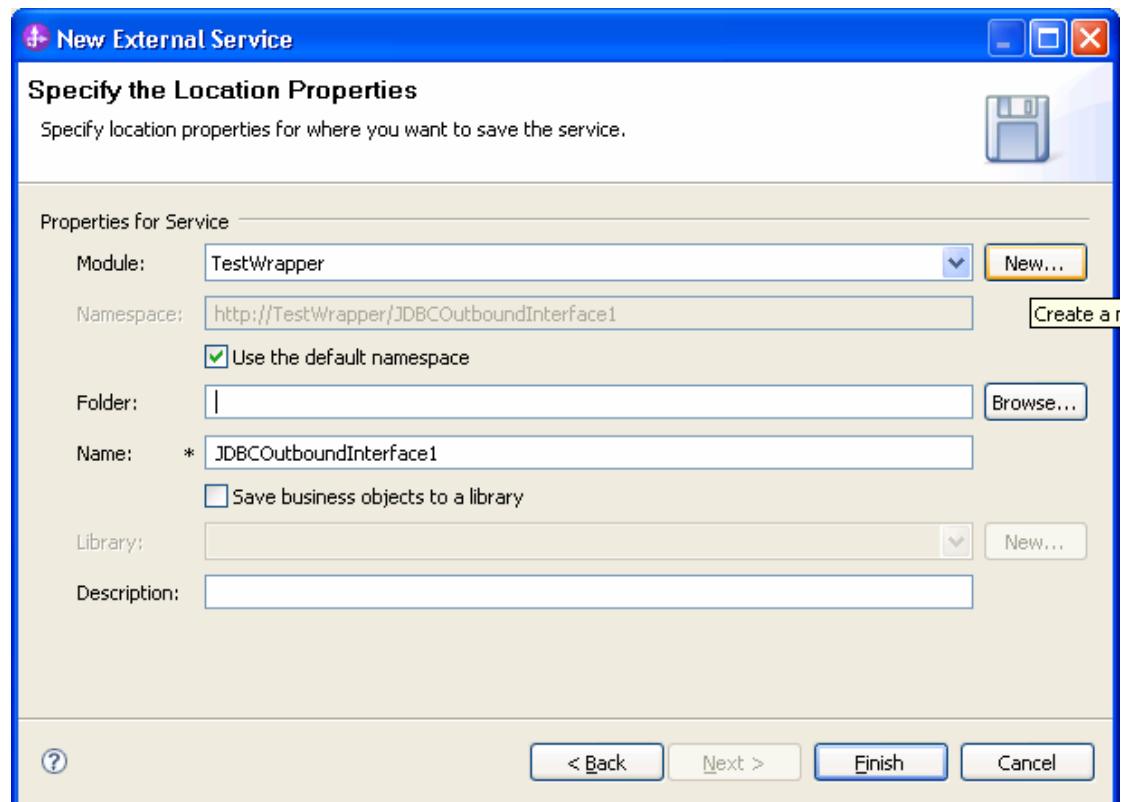
Database connection information:

Database system connection information
Database vendor: ORACLE
Connection pool DataSource JNDI name: *

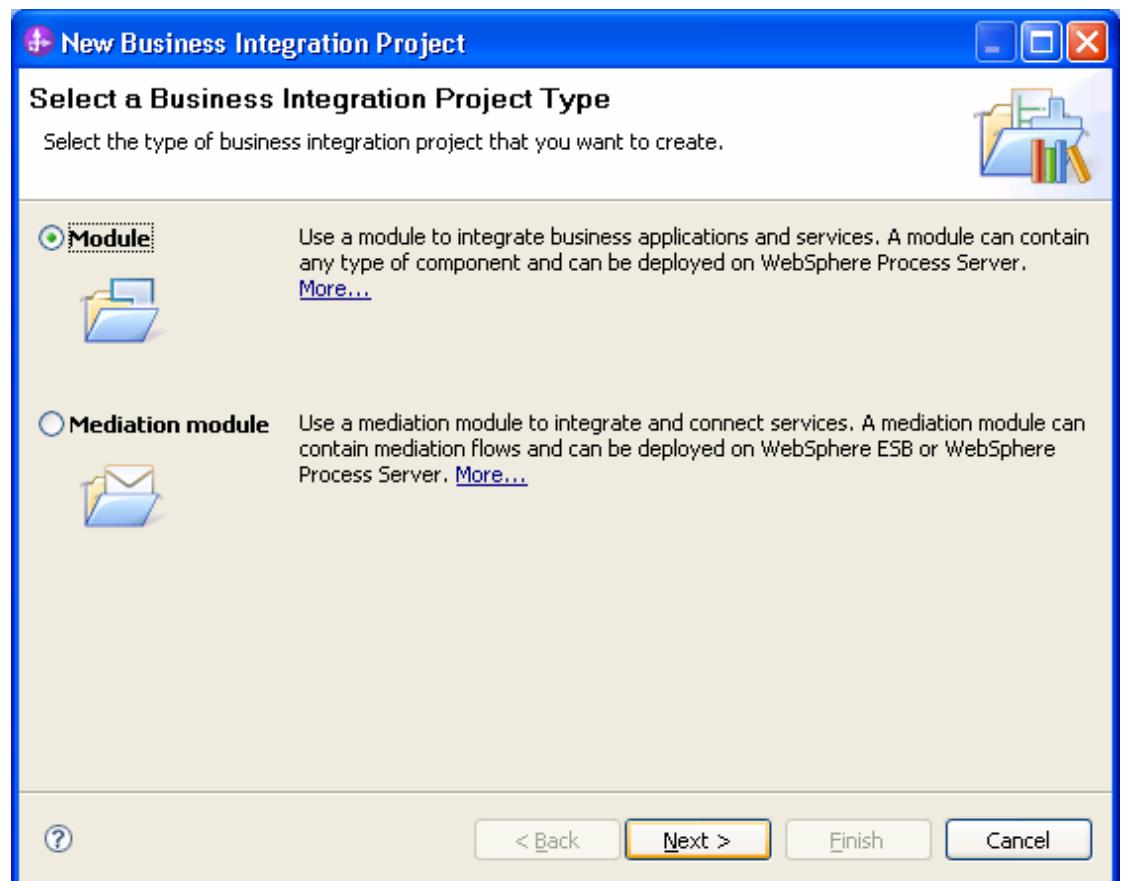
[Advanced >>](#)

[?](#) [< Back](#) [Next >](#) [Finish](#) [Cancel](#)

3. Click **New** in the Specify the Location Properties window.

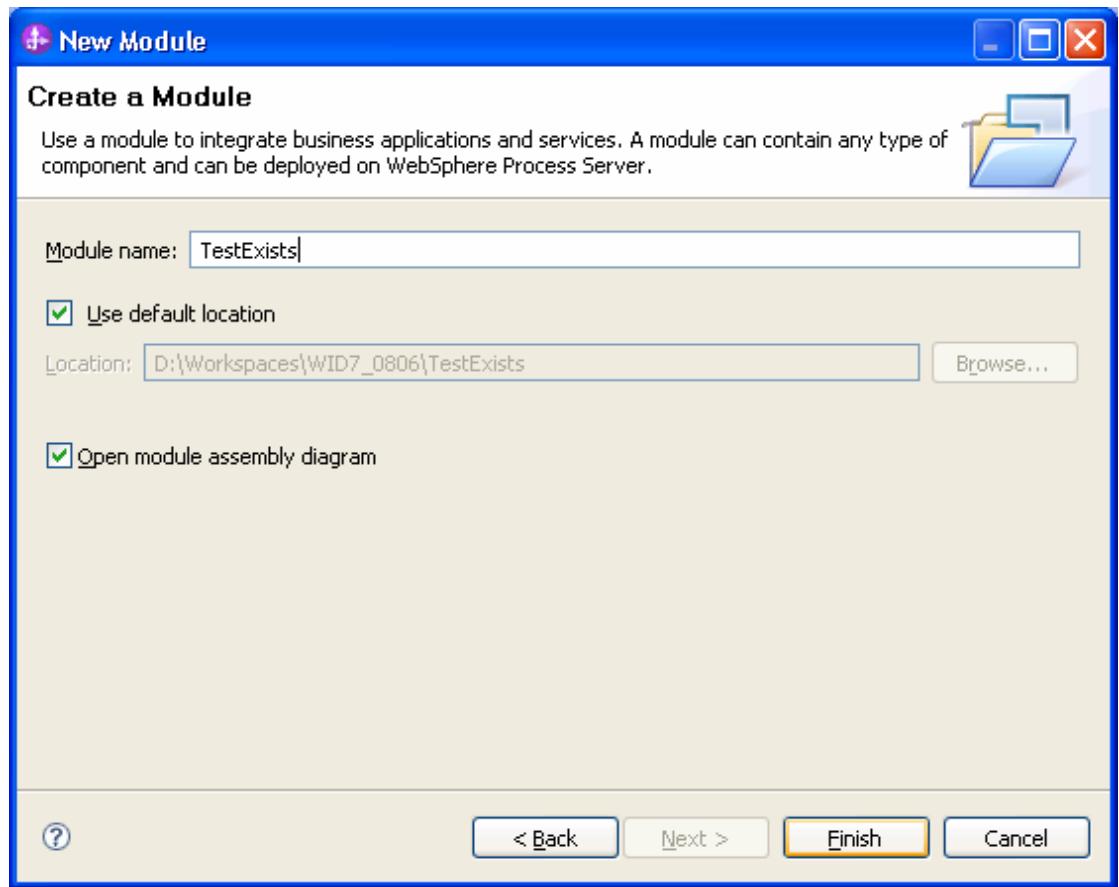


4. In the Select a Business Integration Project Type window, select **Module** and click **Next**.

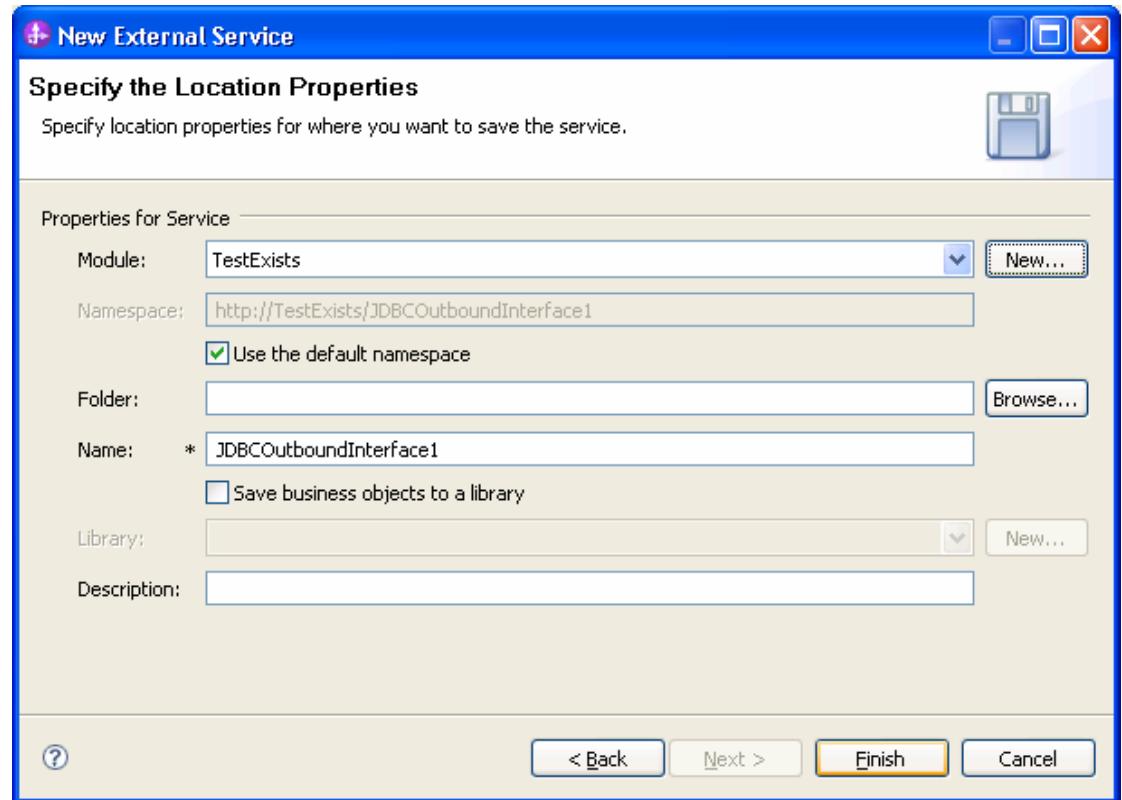


WebSphere software

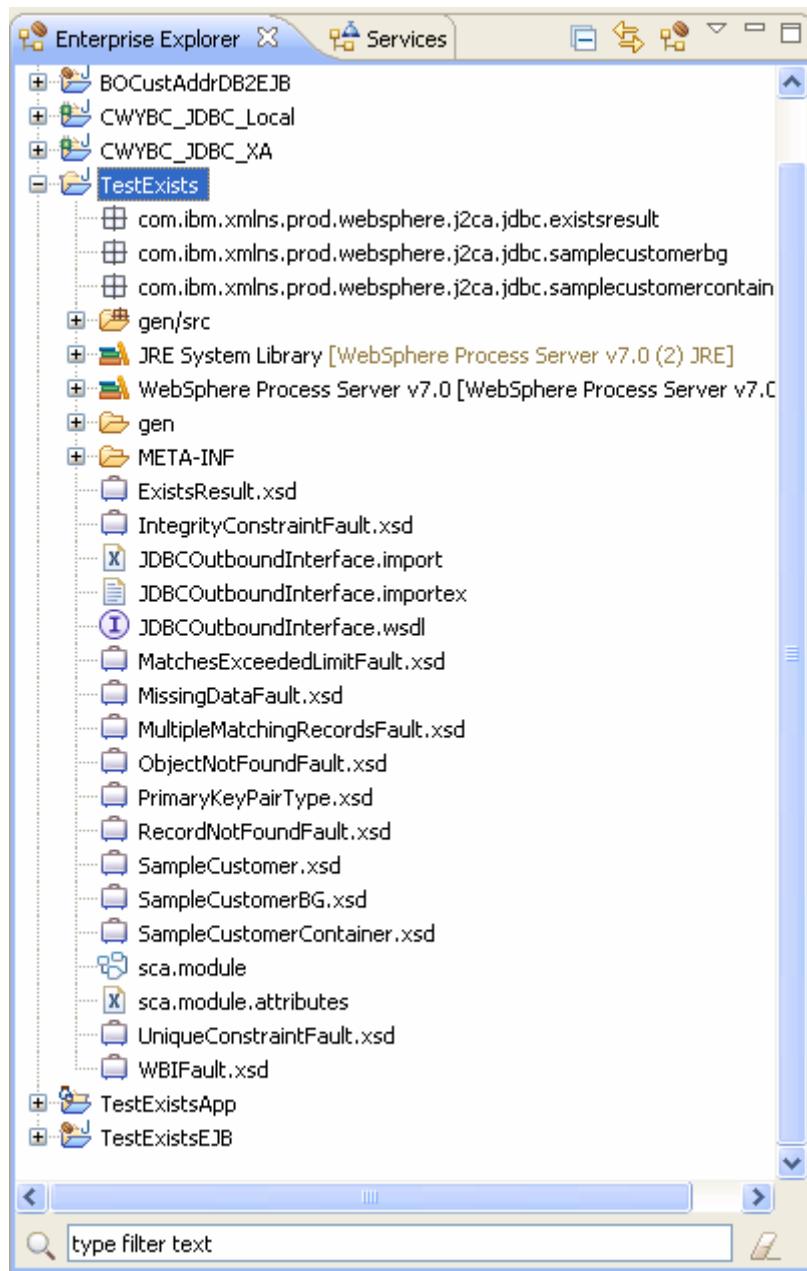
5. In the Create a Module window, type **TestExists** in the **Module Name** field and click **Finish**.



6. In the Specify the Location Properties window, click **Finish** to finish the service creation.



7. Open the Project Explorer and verify that the business objects are created correctly.



Deploy the module to the test environment

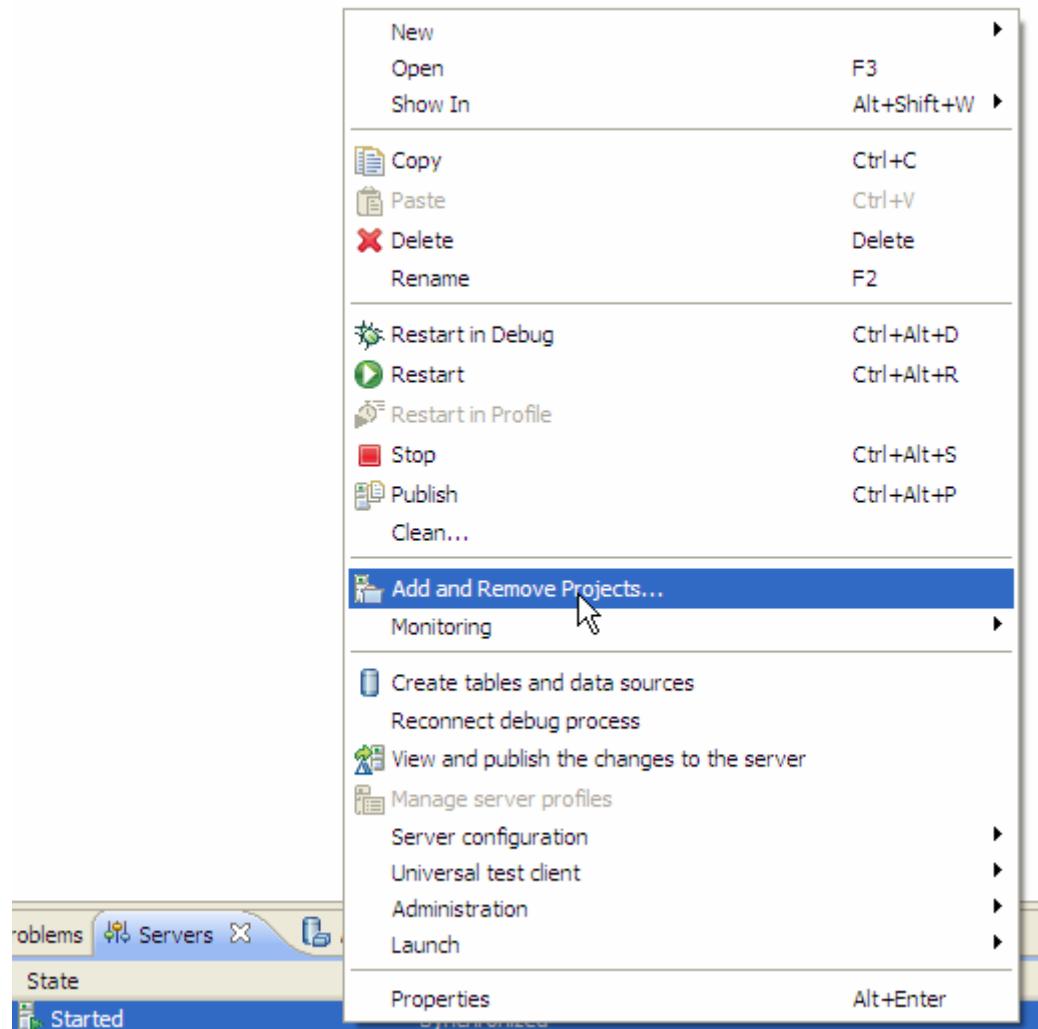
After running the external service wizard, you will have an SCA module that contains an Enterprise Information System (EIS) import. You must install this SCA module in the WebSphere Integration Developer integration test client to deploy it. To do this, you must add the SCA module you created earlier to the server using the **Servers** view in WebSphere Integration Developer.

Steps for adding the SCA module to the server:

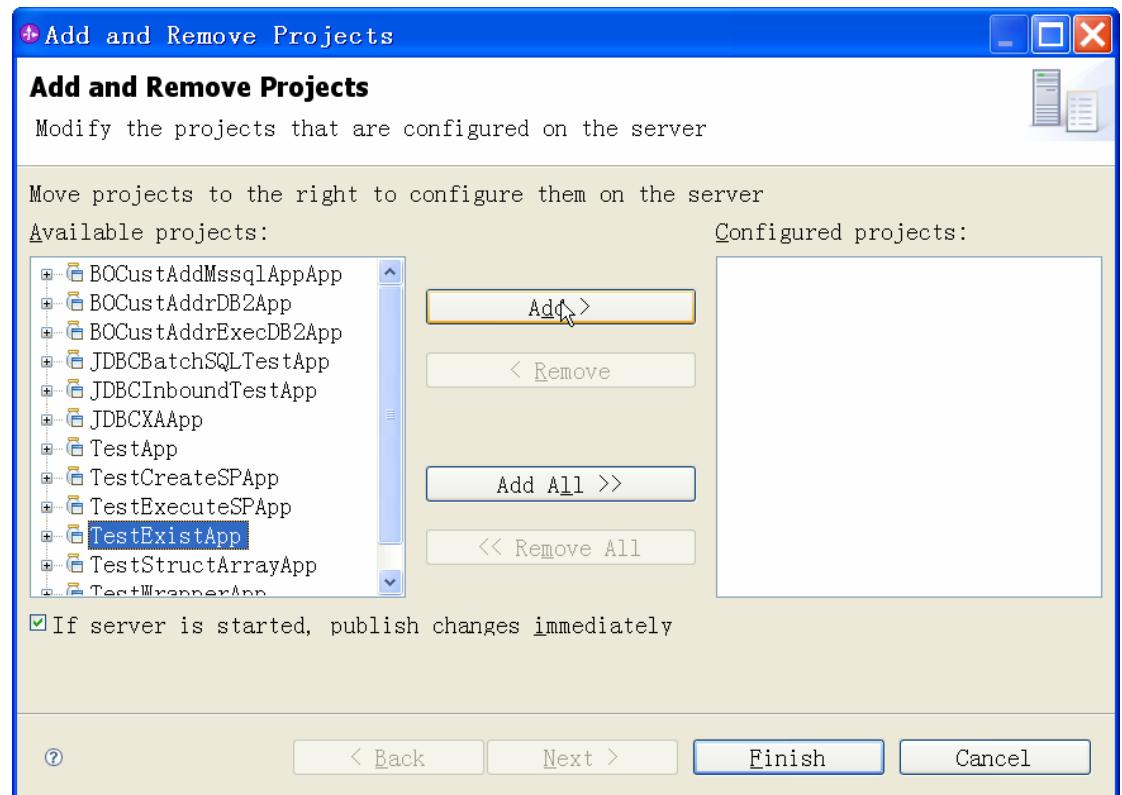
1. In WebSphere Integration Developer, switch to the **Servers** view by selecting from the toolbar **Window > Show View > Servers**.
2. In the **Servers** tab in the lower-right pane right click the server, and select **Start**.

WebSphere software

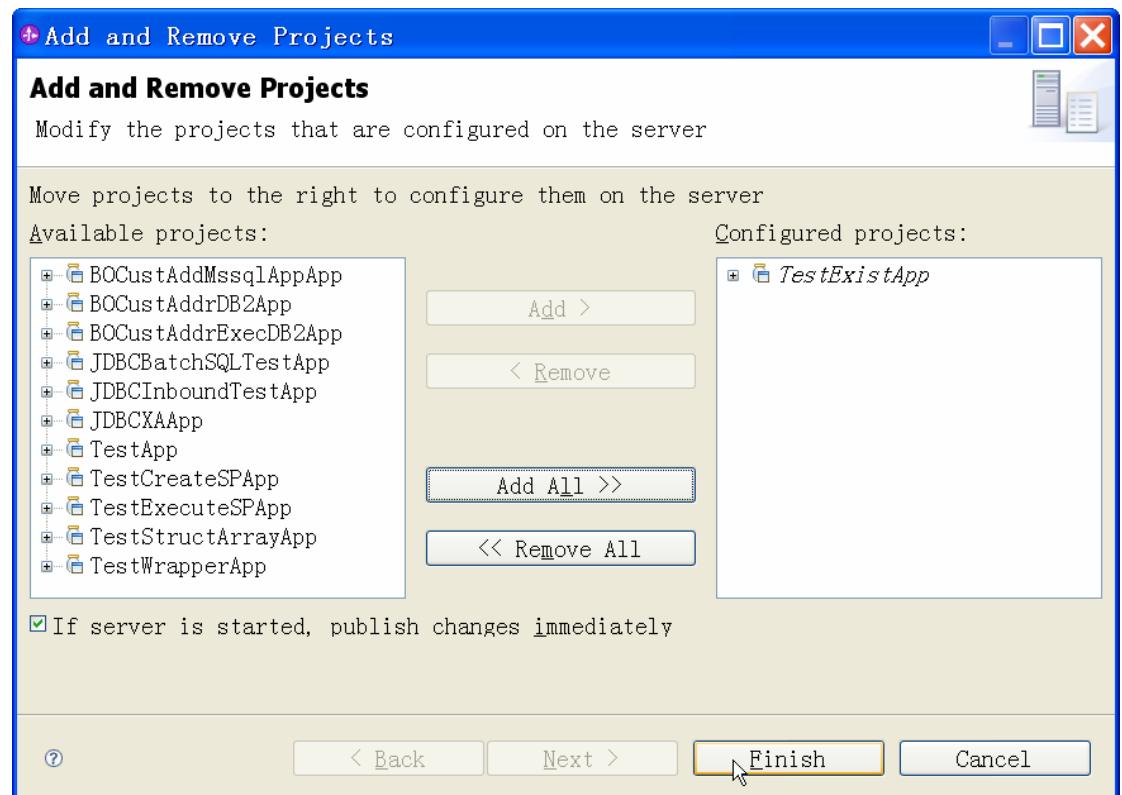
3. After the server is started, right-click the server, and select **Add and Remove projects**.



The Add and Remove Projects window lists the available projects in the WebSphere Integration Developer workspace.



4. In the Add and Remove Projects window, select your project (TestExistApp) and click **Add** to configure the project on the server. Click **Finish**.

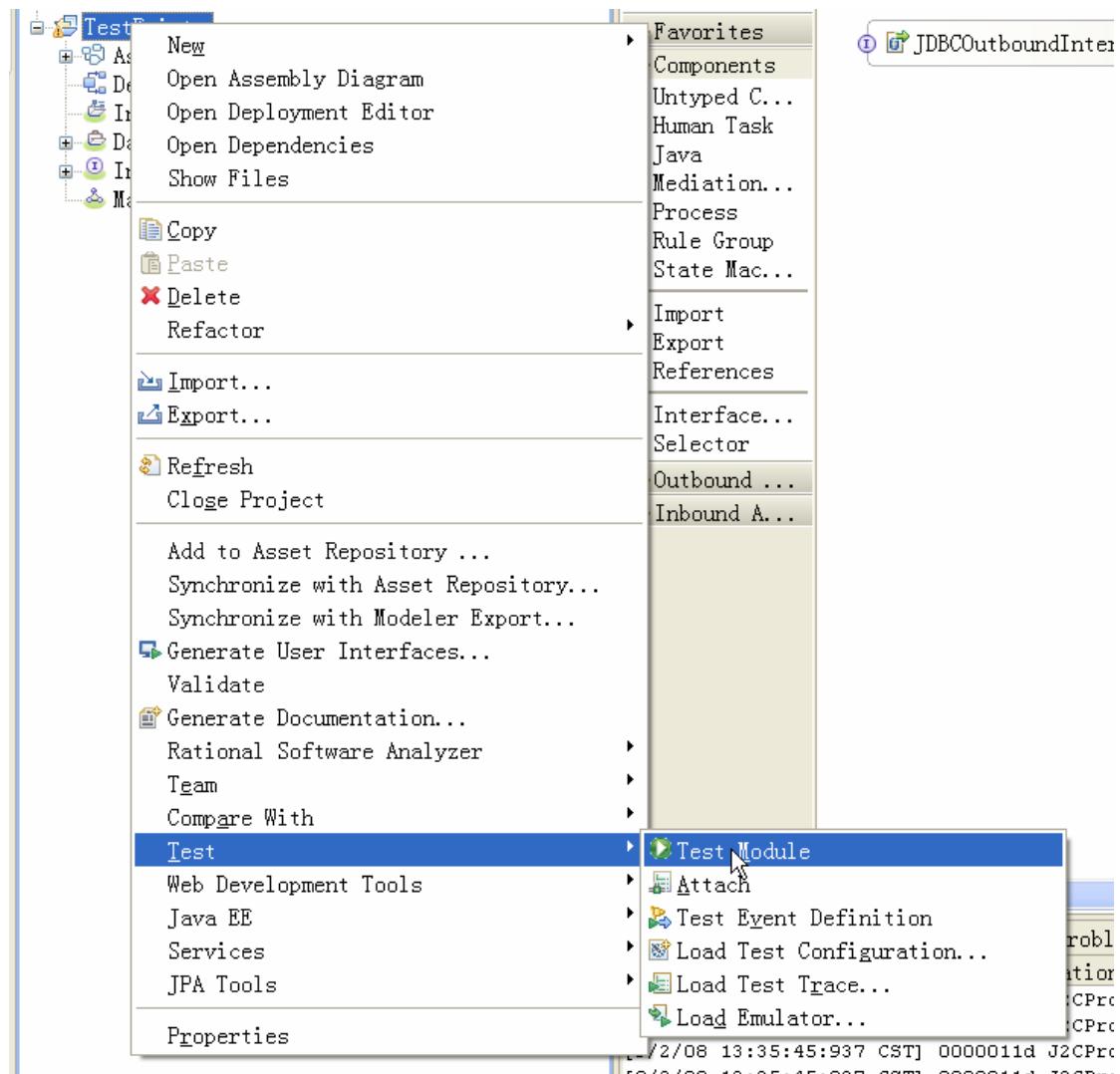


Test the assembled adapter application

WebSphere software

Test the assembled adapter application using the WebSphere Integration Developer integration test client.

1. Select the **TestExists** module, right-click, and select **Test > Test Module**. The Test Client window is displayed.



2. Select **existsJabdullaCustomerBG** from the **Operation** list.

WebSphere software

► General Properties

▼ Detailed Properties

<u>Configuration:</u>	Default Module Test	
<u>Module:</u>	TestExists	
<u>Component:</u>	JDBCOutboundInterface	
<u>Interface:</u>	JDBCOutboundInterface	
<u>Operation:</u>	existsJabdullaCustomerBG	

Initial request parameters

Name	Type	Value
existsJabdullaCustomerBG	JabdullaCustomerBG	
verb	verb<string>	 Create
JabdullaCustomer	JabdullaCustomer	
pkey	string	
fname	string	
lname	string	
ccode	string	

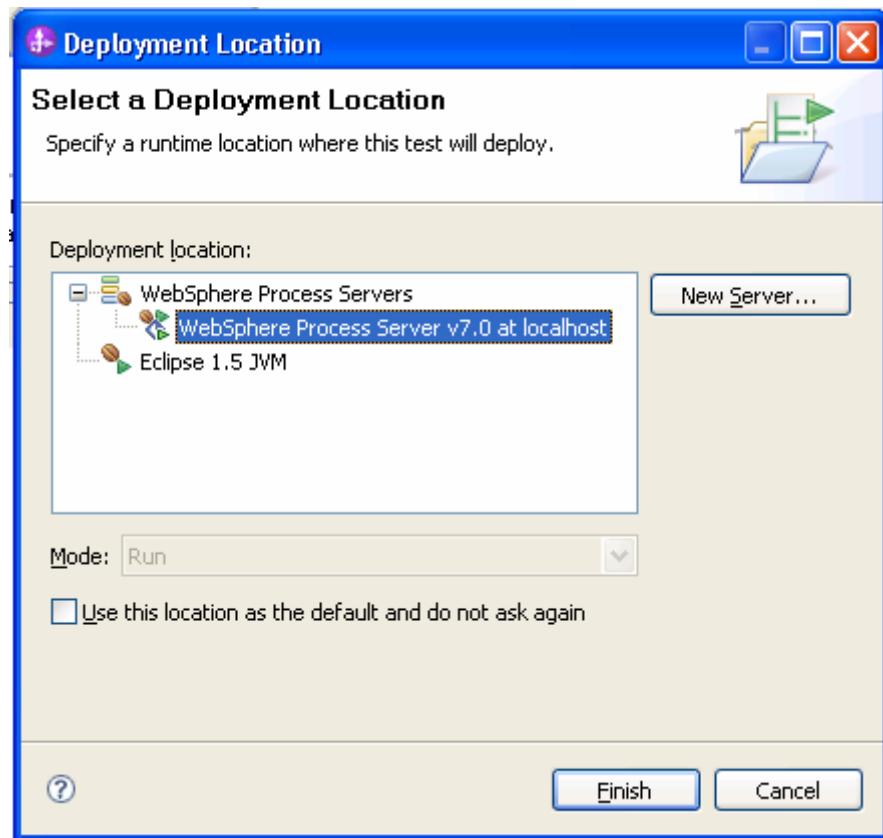
3. Right-click **verb**, select **Set To > Unset**. Enter 1000 for pkey, and unset lname, fname and ccode.

Initial request parameters

Name	Type	Value
existsJabdullaCust	JabdullaCusto...	✓
verb	verb<string>	✗✓
JabdullaCustomer	JabdullaCustomer	✓
pkey	string	✓ 1000
fname	string	✗✓
lname	string	✗✓
ccode	string	✓

A context menu is open over the 'ccode' row, showing options: Copy Value, Paste Value, Select All, Set To, Set Required to Default, Value..., Default, and Unset. The 'Unset' option is highlighted.

4. To execute the service, click Continue .
5. In the Select Deployment location window, select the server and click **Finish**.



6. Check the return value to ensure it matches expected values.

WebSphere software

Events

- Invoke (JDBCOutboundInterface:existsJabdullaCustomerBG)
 - Invoke started
 - Invoke (JDBCOutboundInterface:existsJabdullaCustomerBG)
 - Return (JDBCOutboundInterface:existsJabdullaCustomerBG)
 - Invoke returned

General Properties

Detailed Properties

Module: TestExists
Component: JDBCOutboundInterface
Interface: JDBCOutboundInterface
Operation: existsJabdullaCustomerBG

Return parameters:

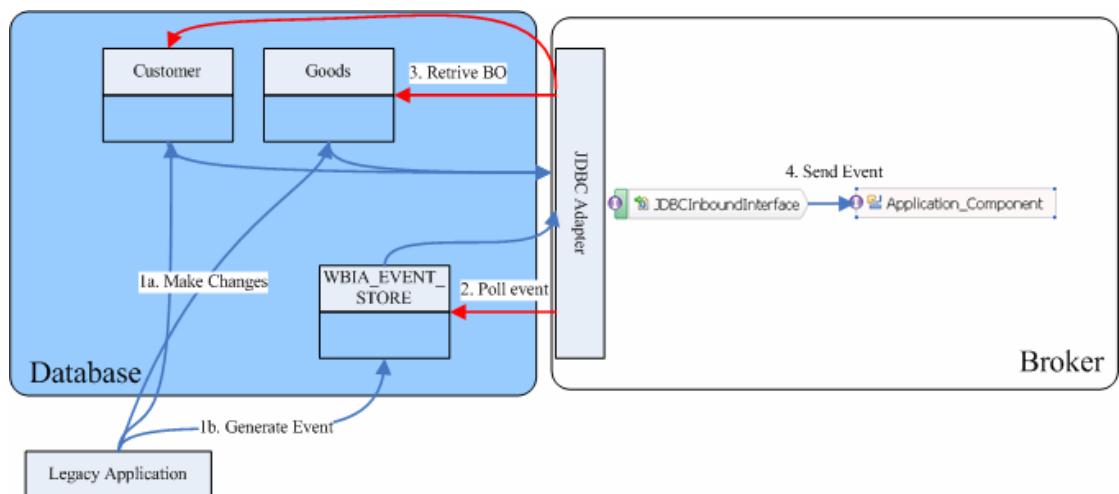
Name	Type	Value
existsJabdullaCustomerBG	ExistsResult	✓
status	boolean	✓ true
recordcount	int	✓ 1

Chapter 15. Tutorial 14: Generate Wrapper business objects for Inbound (Oracle)

This tutorial demonstrates how WebSphere Adapter for JDBC 7.0.0.0 retrieves customer information from an application's database. A wrapper business object is used to retrieve records from multiple tables with one event entry.

About this task

This scenario illustrates the ability of WebSphere JDBC adapter to interact with database by polling database event from an event table. In this scenario, a legacy application makes some change of the CUSTOMER table and the GOODS table in a single operation. Then, insert an event entry record into the event table (WBIA_EVENT_TABLE). Then, the event will be polled by JDBC adapter and send it to one SCA component. JDBC adapter screen all the database operation details, event quality assuring details and provide a simple event interface for the application component. The following figure shows the whole scenario:



This case has three steps:

1. The legacy application will make the changes and then generate an event record. For simplify reason, we will insert records using SQL statement directly.
 2. JDBC adapter will poll the event from database periodically. Thus, it will find the new events and fetch the event and corresponding business objects from database.
 3. At last, JDBC adapter will convert the event to a SDO and send it to the destination SCA component.
-

Prepare to run through the tutorial

Extract the sample files

Replicas of the artifacts that you create when using the external service wizard are provided as sample files for your reference. Use these files to verify if the files you create using the external service wizard are correct.

Download the sample zip file and extract it into a directory of your choice (you may want to create a new directory).

Configuration prerequisites

Before configuring the adapter, you must complete the following tasks:

- Create tables
- Create an authentication alias
- Create a data source

Create tables

You must create the following tables in the Oracle database before starting the scenario.

```
CREATE TABLE CUSTOMER (
    PKEY VARCHAR2(10) NOT NULL PRIMARY KEY,
    FNAME VARCHAR2(20) ,
    LNAME VARCHAR2(20) ,
    CCODE VARCHAR2(10) ) ;

CREATE TABLE CUSTADD (
    ADDRID VARCHAR2(10) NOT NULL PRIMARY KEY,
    CUSTID VARCHAR2(10) ,
    CITY VARCHAR2(20) ,
    ZIPCODE VARCHAR2(10) ) ;

CREATE TABLE WBIA_JDBC_EVENTSTORE
(
    EVENT_ID INTEGER NOT NULL PRIMARY KEY,
    XID          VARCHAR2(200),
    OBJECT_KEY   VARCHAR2(80)      NOT NULL,
    OBJECT_NAME  VARCHAR2(40)      NOT NULL,
    OBJECT_FUNCTION  VARCHAR2(40)      NOT
NULL,
    EVENT_PRIORITY  INTEGER      NOT
NULL,
    EVENT_TIME     TIMESTAMP,
    EVENT_STATUS   INTEGER      NOT NULL,
    EVENT_COMMENT  VARCHAR2(100)
);
```

Insert records into the tables we just created.

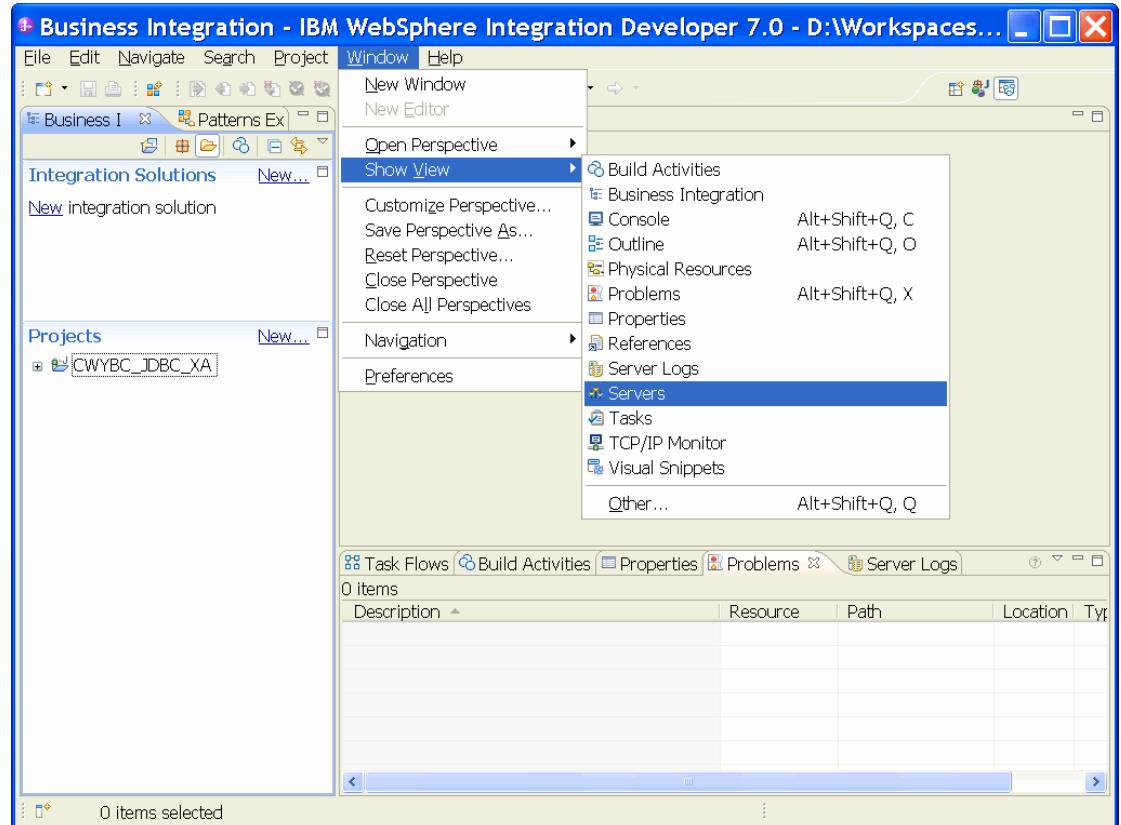
```
INSERT INTO CUSTOMER (PKEY, FNAME, LNAME, CCODE)
VALUES ('C1', 'JONE', 'TIGER', '1');
INSERT INTO CUSTOMER (PKEY, FNAME, LNAME, CCODE)
VALUES ('C2', 'ROTH', 'GREEN', '1');
INSERT INTO CUSTADD (ADDRID, CUSTID, CITY, ZIPCODE)
VALUES ('A1', 'C1', 'BEIJING', '100000');
INSERT INTO CUSTADD (ADDRID, CUSTID, CITY, ZIPCODE)
VALUES ('A2', 'C2', 'SHANGHAI', '200000');
```

Create an authentication alias

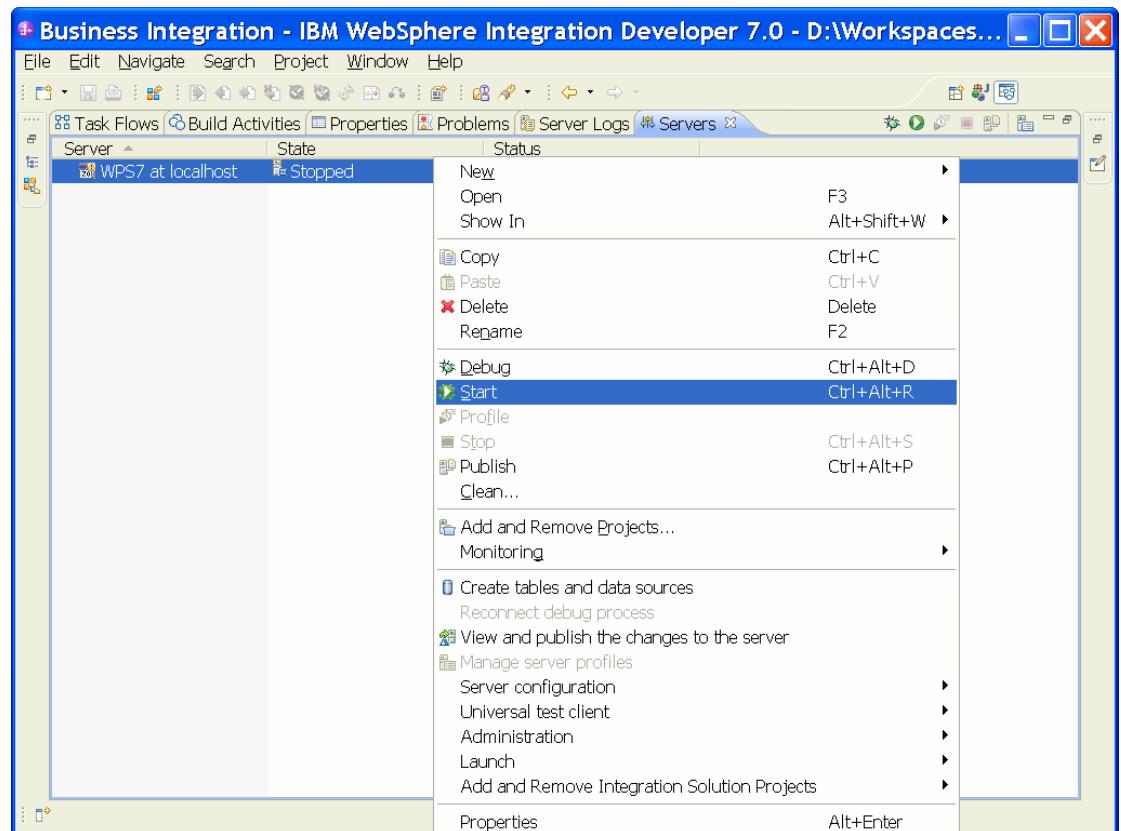
The authentication alias needs to be set because the data source created in the next section uses the username and password set in the authentication alias to connect to the database.

Follow these steps to set the authentication alias in the WebSphere Process Server administrative console.

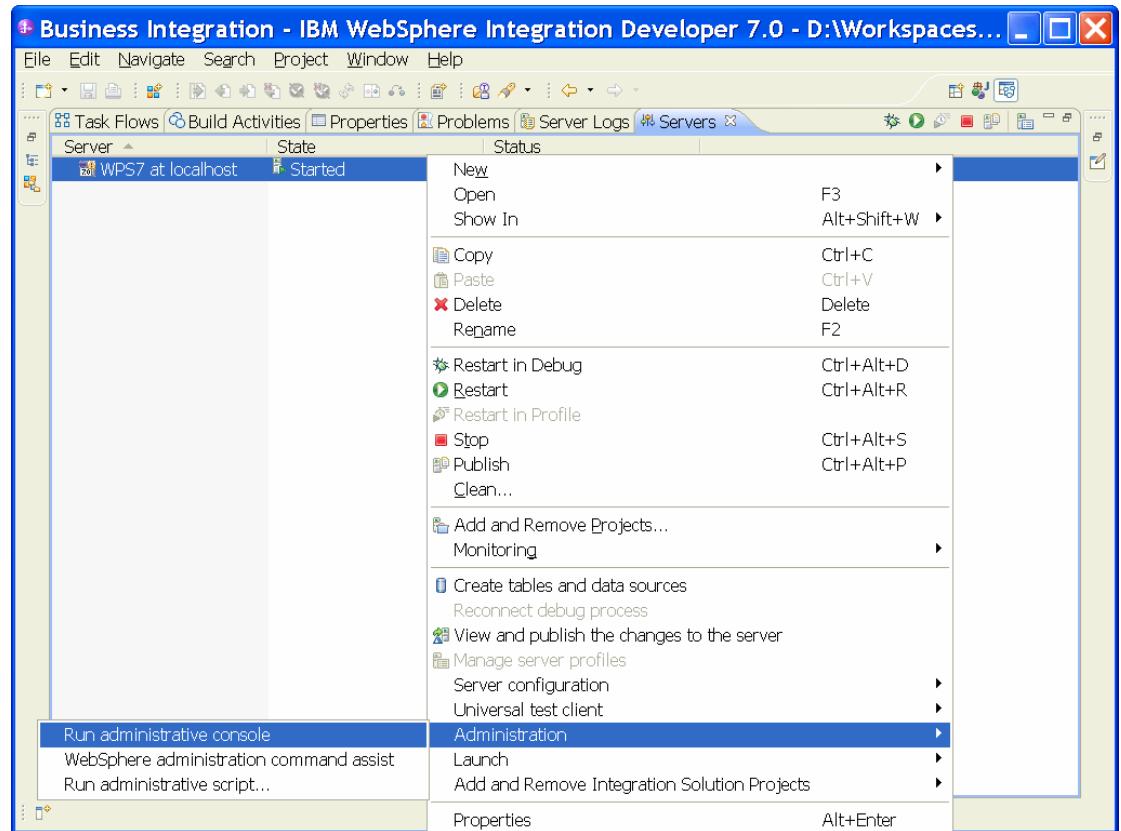
1. In WebSphere Integration Developer, switch to the **Servers** view by selecting **Windows > Show View > Servers**.



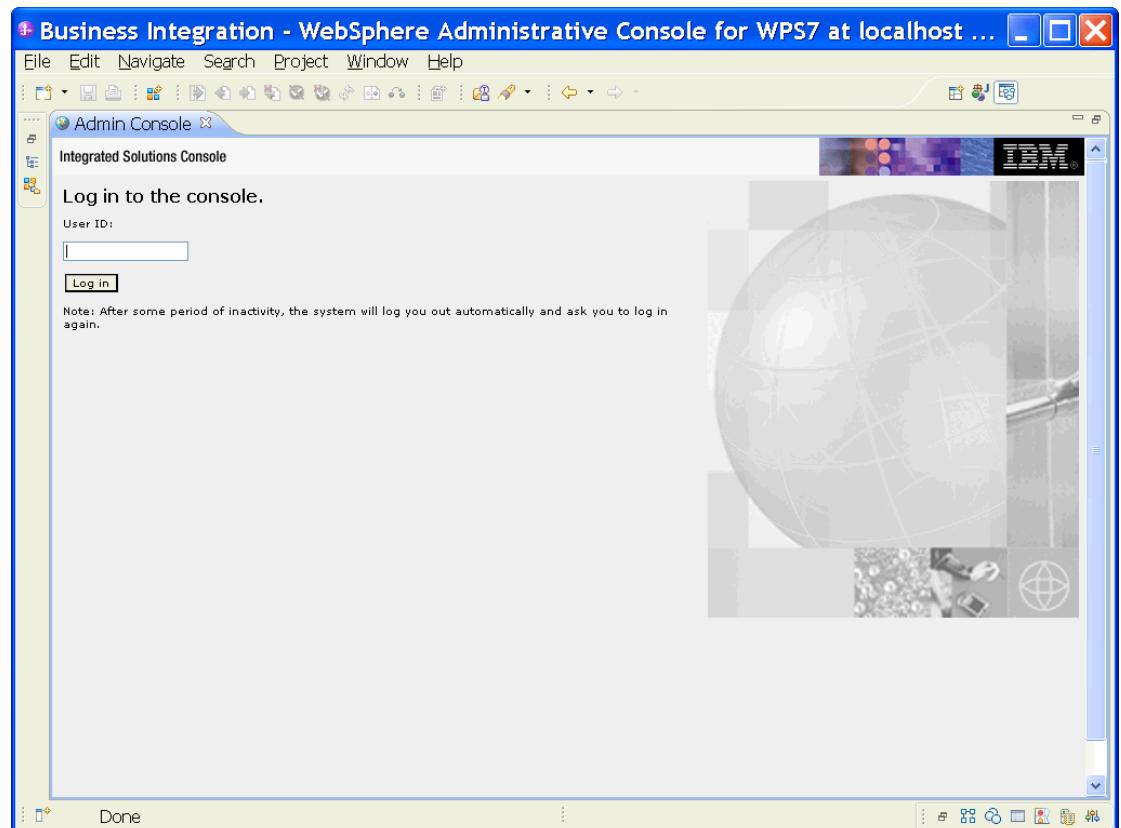
2. In the **Servers** view, right-click the server that you want to start and select **Start**.



3. After the server is started, right-click the server, and select **Administration > Run administrative console**.



4. Log on to the administrative console.



WebSphere software

5. Click **Security → Global security**.



6. On the right, click **J2C Authentication Data** under **Java Authentication and Authorization Service**.

Global security

Use this panel to configure administration and the default application security policy. This security configuration applies to functions and is used as a default security policy for user applications. Security domains can be defined to override and control applications.

[Security Configuration Wizard](#)[Security Configuration Report](#)**Administrative security**

- Enable administrative security
 - [Administrative user roles](#)
 - [Administrative group roles](#)
 - [Administrative authentication](#)

Application security

- Enable application security

Java 2 security

- Use Java 2 security to restrict application access to local resources
 - Warn if applications are granted custom permissions
 - Restrict access to resource authentication data

User account repository

Current realm definition

Federated repositories

Available realm definitions

Federated repositories

[Configure...](#)[Set as current](#)**Authentication**

Authentication mechanisms and expiration

[LTPA](#)[Kerberos and LTPA](#)[Kerberos configuration](#)[SWAM \(deprecated\): No authentication](#)[Authentication cache settings](#) Web and SIP security RMI/IOP security Java Authentication and Authorization[Application logins](#)[System logins](#)[J2C authentication data](#)

Specifies a list of user identities and passwords for Java(TM) 2 connector security to use.

 [Security domains](#) [External authorization providers](#) [Custom properties](#)

A list of existing aliases is displayed.

WebSphere software

Global security > JAAS - J2C authentication data

Specifies a list of user identities and passwords for Java(TM) 2 connector security to use.

Prefix new alias names with the node name of the cell (for compatibility with earlier releases)

Apply

Preferences

<input type="button" value="New"/> <input type="button" value="Delete"/>			
Select	Alias	User ID	Description
You can administer the following resources:			
<input type="checkbox"/>	BSpace JDBC Alias	wbiuser	Business Space Authorization Alias
<input type="checkbox"/>	CommonEventInfrastructureJMSAuthAlias	wbiuser	Authentication alias for the Common Event Infrastructure JMS Topics and Queues
<input type="checkbox"/>	SCA Auth Alias	wbiuser	This is the alias used by SCA to login to a secured SIBus
<input type="checkbox"/>	localhostNode01Cell/nlNode01/server1/EventAuthDataAliasDerby		Derby authentication alias for the Event Server
Total 4			

- Click **New** to create a new authentication entry. Type the alias name, and username and password to connect to the database. Click **OK**.

Cell=localhostNode01Cell, Profile=AppSrv01

Global security

Global security > JAAS - J2C authentication data > New

Specifies a list of user identities and passwords for Java(TM) 2 connector security to use.

General Properties

* Alias

* User ID

* Password

Description

Buttons

- Click **Save** to save the changes.

Cell=localhostNode01Cell, Profile=AppSrv01

Global security > JAAS - J2C authentication data

Specifies a list of user identities and passwords for Java(TM) 2 connector security to use.

Prefix new alias names with the node name of the cell (for compatibility with earlier releases)

Apply

You have created an authentication alias that will be used to configure the data source.

Preferences

New	Delete	Select	Alias	User ID	Description
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	BSpace JDBC Alias	wbiuser	Business Space Authorization Alias
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	CommonEventInfrastructureJMSAuthAlias	wbiuser	Authentication alias for the Common Event Infrastructure JMS Topics and Queues
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	SCA Auth Alias	wbiuser	This is the alias used by SCA to login to a secured SIBus
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	localhostNode01Cell/nlNode01/server1/EventAuthDataAliasDerby		Derby authentication alias for the Event Server
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	nlNode01/Alias	luweiqin	

Total 5

Create a data source

Create a data source in WebSphere Process Server, which the adapter will use to connect to the database. Here are the steps to create the data source in the WebSphere Process Server administrative console. This data source will be used later when generating the artifacts for the module.

Note: This tutorial will use Oracle as the database and the Oracle thin driver, ojdbc6.jar.

WebSphere software

1. In the administrative console, select **Environment → WebSphere Variables**.



2. On the right, select **ORACLE_JDBC_DRIVER_PATH** and specify the path of the ojdbc6.jar file in the **Value** field. Click **OK**.

WebSphere software

Cell=localhostNode01Cell, Profile=AppSrv01

WebSphere Variables

[WebSphere Variables](#) > ORACLE_JDBC_DRIVER_PATH

Use this page to define substitution variables. Variables specify a level of indirection for some system-defined values, such as file system root directories. Variables have a scope level, which is either server, node, cluster, or cell. Values at one scope level can differ from values at other levels. When a variable has conflicting scope values, the more granular scope value overrides values at greater scope levels. Therefore, server variables override node variables, which override cluster variables, which override cell variables.

Configuration

General Properties

* Name: ORACLE_JDBC_DRIVER_PATH

Value: D:\Lib

Description: The directory that contains the Oracle thin or oci8 JDBC Driver.

Buttons: Apply, OK, Reset, Cancel

3. Click **Save** to save the changes.

WebSphere Variables

Messages

⚠ Changes have been made to your local configuration. You can:
• [Save](#) directly to the master configuration.
• [Review](#) changes before saving or discarding.
⚠ The server may need to be restarted for these changes to take effect.

The variable is added and appears in the list.

Preferences

New Delete

MQ JDBC DRIVER PATH

Select	Name	Value	Scope
You can administer the following resources:			
<input type="checkbox"/>	MQ_INSTALL_ROOT	`\${WAS_INSTALL_ROOT}/lib/WMQ	Node=n1Node01
<input type="checkbox"/>	ORACLE_JDBC_DRIVER_PATH	D:\Lib	Node=n1Node01
<input type="checkbox"/>	OS400_NATIVE_JDBC40_DRIVER_PATH		Node=n1Node01
<input type="checkbox"/>	OS400_NATIVE_JDBC_DRIVER_PATH		Node=n1Node01
<input type="checkbox"/>	OS400_TOOLBOX_JDBC_DRIVER_PATH		Node=n1Node01
<input type="checkbox"/>	ORACLE_JDBC_DRIVER_PATH		Node=n1Node01

WebSphere software

4. Select **Resources** → **JDBC** -> **JDBC Providers**.



5. Click **New** in the JDBC providers window.

JDBC providers

JDBC providers

Use this page to edit properties of a JDBC provider. The JDBC provider object encapsulates the specific JDBC driver implementation class for access to the specific vendor database of your environment. Learn more about this task in a [guided activity](#). A guided activity provides a list of task steps and more general information about the topic.

Scope: Cell=**localhostNode01Cell**, Node=**n1Node01**

Scope specifies the level at which the resource definition is visible. For detailed information on what scope is and how it works, [see the scope settings help](#).

Node=n1Node01

Preferences

Select	Name	Scope	Description
None			
Total 0			

6. Select Oracle database with a connection pool data source for the Oracle JDBC driver. Click **Next**.

Create a new JDBC Provider**Create a new JDBC Provider****→ Step 1: Create new JDBC provider**

Step 2: Enter database class path information

Step 3: Summary

Create new JDBC provider

Set the basic configuration values of a JDBC provider, which encapsulates the specific vendor JDBC driver implementation classes that are required to access the database. The wizard fills in the name and the description fields, but you can type different values.

Scope

cells:localhostNode01Cell:nodes:n1Node01

*** Database type**

Oracle

*** Provider type**

Oracle JDBC Driver

*** Implementation type**

Connection pool data source

*** Name**

Oracle JDBC Driver

Description

Oracle JDBC Driver

Next**Cancel**

7. In the **Enter database classpath information** page, enter the following value for the **Class path** field:
\$(ORACLE_JDBC_DRIVER_PATH)/ojdbc6.jar, where
\$(ORACLE_JDBC_DRIVER_PATH) is library path for the run time.
8. Click **Next**.

WebSphere software

Cell=localhostNode01Cell, Profile=AppSrv01

Create a new JDBC Provider

Step 1: Create new JDBC provider
→ Step 2: Enter database class path information
Step 3: Summary

Enter database class path information

Set the environment variables that represent the JDBC driver class files, which WebSphere(R) Application Server uses to define your JDBC provider. This wizard page displays the file names; you supply only the directory locations of the files. Use complete directory paths when you type the JDBC driver file locations. For example: C:\SQLLIB\java on Windows(R) or /home/db2inst1/sqllib/java on Linux(TM).

If a value is specified for you, you may click Next to accept the value.

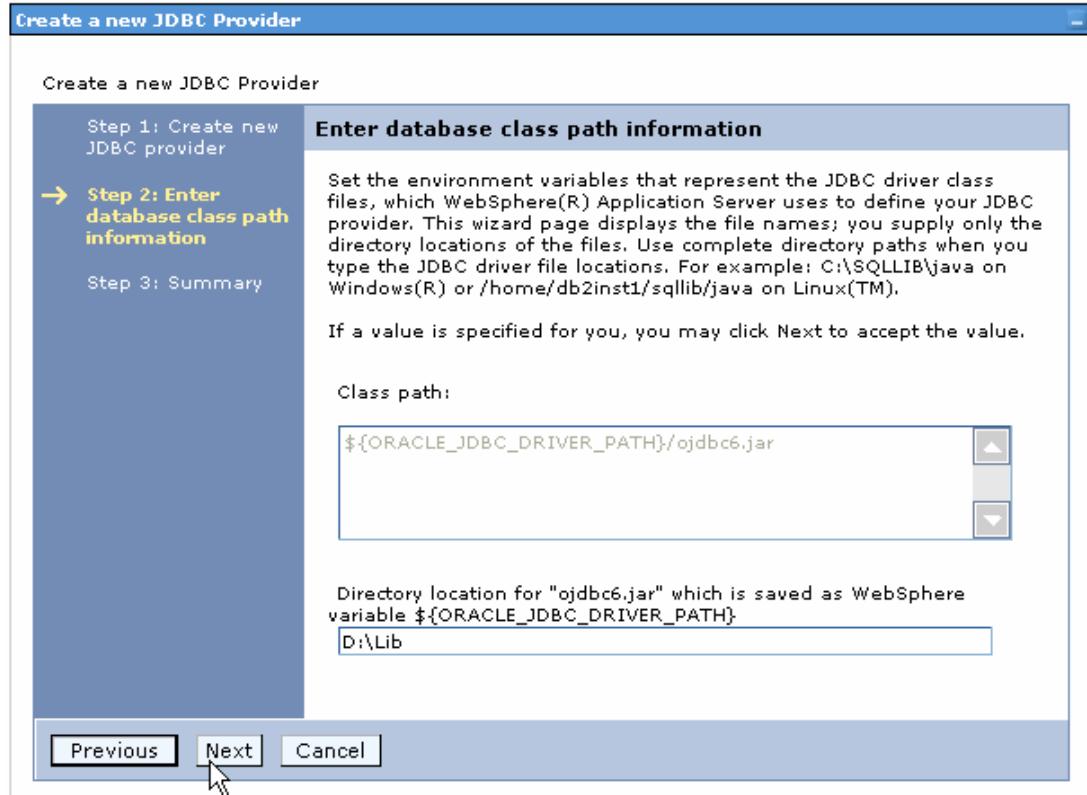
Class path:

`${ORACLE_JDBC_DRIVER_PATH}/ojdbc6.jar`

Directory location for "ojdbc6.jar" which is saved as WebSphere variable `${ORACLE_JDBC_DRIVER_PATH}`

D:\Lib

Previous **Next** Cancel



9. In the Summary page, click **Finish**.

Cell=localhostNode01Cell, Profile=AppSrv01 [Close page](#)

Create a new JDBC Provider

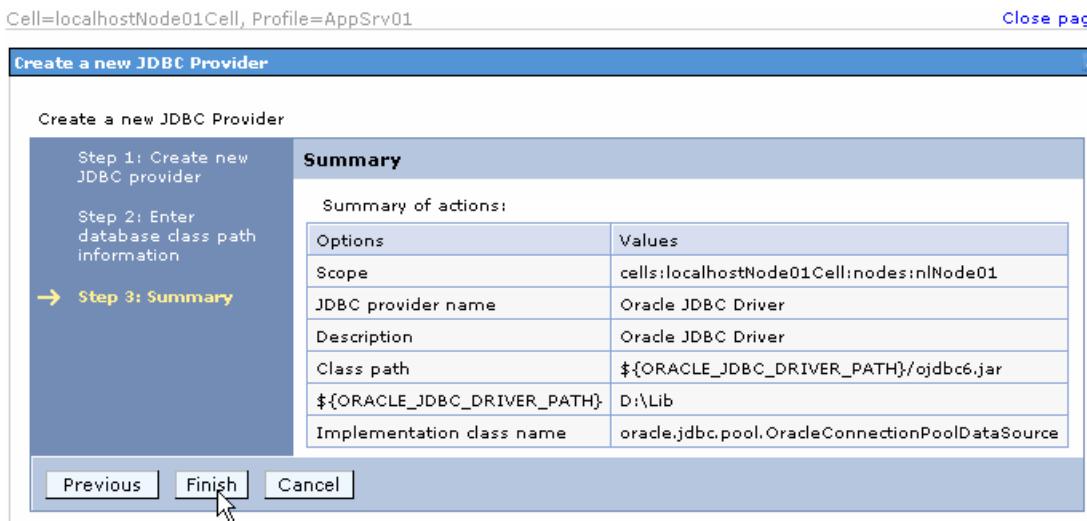
Step 1: Create new JDBC provider
Step 2: Enter database class path information
→ Step 3: Summary

Summary

Summary of actions:

Options	Values
Scope	cells:localhostNode01Cell:nodes:n1Node01
JDBC provider name	Oracle JDBC Driver
Description	Oracle JDBC Driver
Class path	<code> \${ORACLE_JDBC_DRIVER_PATH}/ojdbc6.jar</code>
<code> \${ORACLE_JDBC_DRIVER_PATH}</code>	D:\Lib
Implementation class name	oracle.jdbc.pool.OracleConnectionPoolDataSource

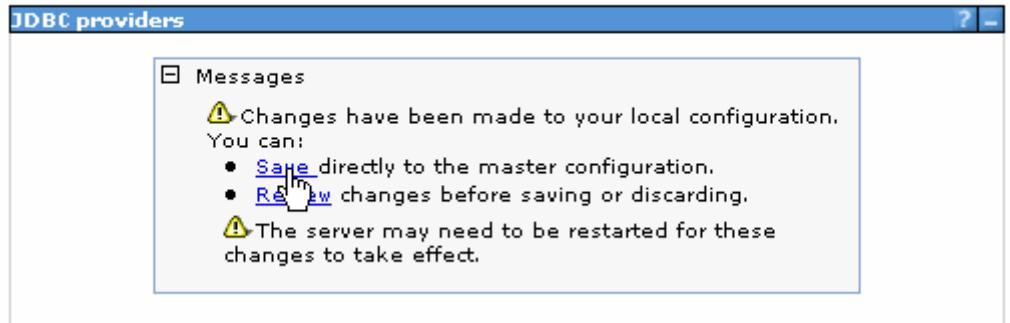
Previous **Finish** Cancel



10. Click **Save** to save the changes.

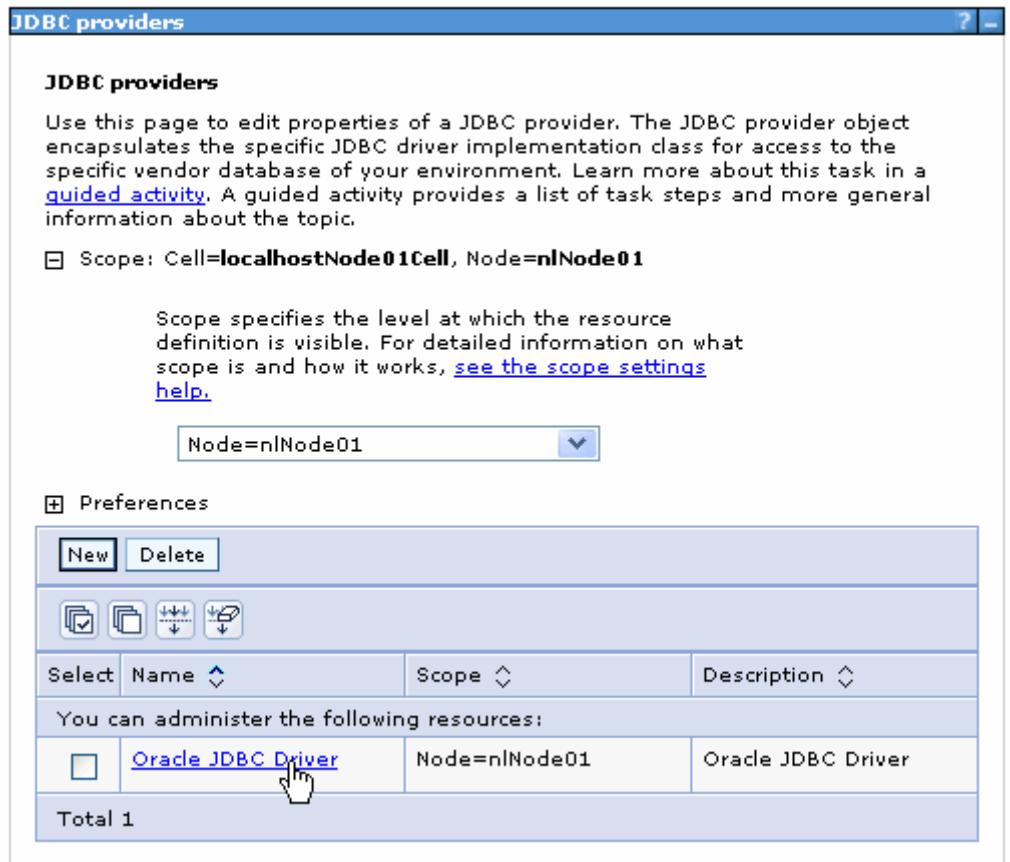
WebSphere software

Cell=localhostNode01Cell, Profile=AppSrv01



The JDBC provider is added and appears in the list.

Cell=localhostNode01Cell, Profile=AppSrv01



11. Select the Oracle JDBC provider you created. Under **Additional Properties**, select **Data sources**. Click **New**.

WebSphere software

Cell=localhostNode01Cell, Profile=AppSrv01

The screenshot shows the 'JDBC providers' page with the 'Oracle JDBC Driver' selected. Under 'Data sources', there is a table with one entry: 'None'. Below the table, it says 'Total 0'. At the top of the table, there are buttons for 'New', 'Delete', 'Test connection', and 'Manage state...'. There are also icons for creating, deleting, and managing connections.

12. Type any value in the **JNDI name** field, and select the authentication alias. Click **Next**.

Cell=localhostNode01Cell, Profile=AppSrv01

The screenshot shows the 'Create a data source' wizard at Step 1: Enter basic data source information. It has a sidebar with steps: Step 1: Enter basic data source information (highlighted), Step 2: Enter database specific properties for the data source, Step 3: Setup security aliases, and Step 4: Summary. The main panel contains fields for Scope (cells:localhostNode01Cell:nodes:n1Node01), JDBC provider name (Oracle JDBC Driver), Data source name (* Oracle JDBC Driver DataSource), and JNDI name (* OracleDS). The 'Next' button is highlighted with a cursor.

13. Provide the appropriate URL value and select a data store helper class name from the **Data store helper class name** list as shown in the following figure. Click **Next**.

WebSphere software

Cell=localhostNode01Cell, Profile=AppSrv01

Create a data source

Step 1: Enter basic data source information

→ Step 2: Enter database specific properties for the data source

Step 3: Setup security aliases

Step 4: Summary

Enter database specific properties for the data source

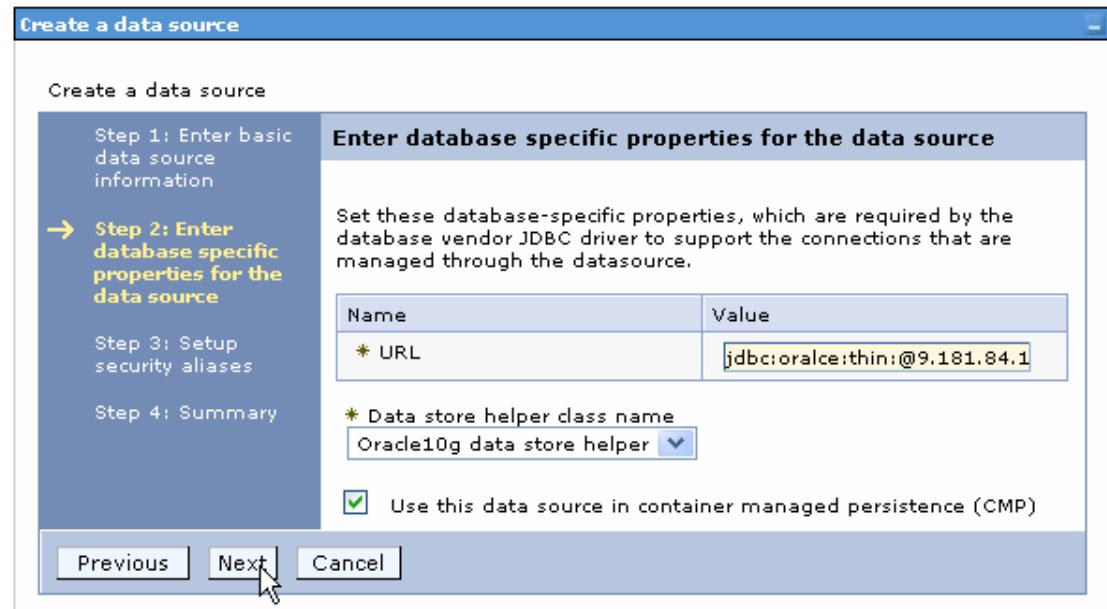
Set these database-specific properties, which are required by the database vendor JDBC driver to support the connections that are managed through the datasource.

Name	Value
* URL	jdbc:oracle:thin:@9.181.84.1

* Data store helper class name
Oracle10g data store helper

Use this data source in container managed persistence (CMP)

Previous Next Cancel



14. Select the authentication alias you just created from the **Component-managed authentication alias** field and click **Next**.

Cell=localhostNode01Cell, Profile=AppSrv01

Create a data source

Step 1: Enter basic data source information

Step 2: Enter database specific properties for the data source

→ Step 3: Setup security aliases

Step 4: Summary

Setup security aliases

Select the authentication values for this resource.

Component-managed authentication alias
nlNode01/Alias_Oracle

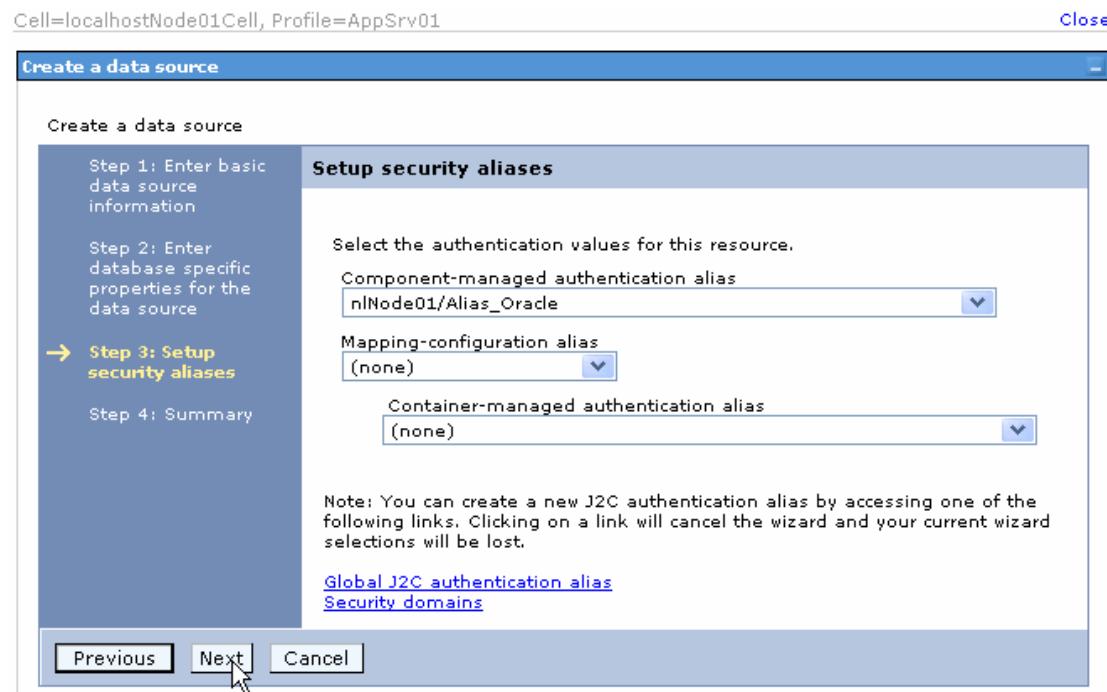
Mapping-configuration alias
(none)

Container-managed authentication alias
(none)

Note: You can create a new J2C authentication alias by accessing one of the following links. Clicking on a link will cancel the wizard and your current wizard selections will be lost.

[Global J2C authentication alias](#)
[Security domains](#)

Previous Next Cancel



15. In the Summary page, review the values entered for the data source and click **Finish**.

WebSphere software

Create a data source

Step 1: Enter basic data source information

Step 2: Enter database specific properties for the data source

Step 3: Setup security aliases

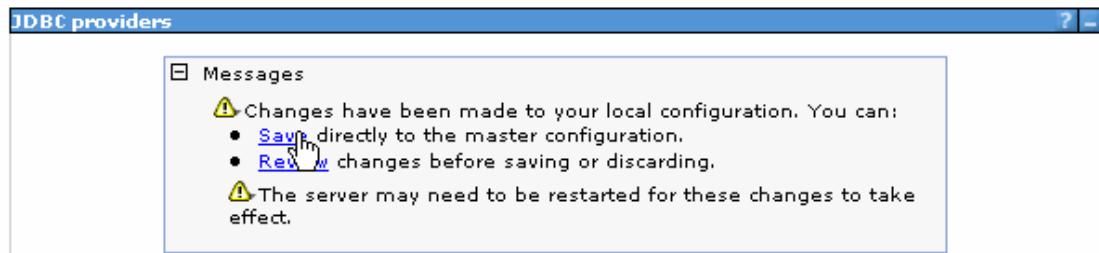
→ Step 4: Summary

Summary	
Summary of actions:	
Options	Values
Scope	cells:localhostNode01Cell:nodes:n1Node01
Data source name	Oracle JDBC Driver DataSource
JNDI name	OracleDS
Select an existing JDBC provider	Oracle JDBC Driver
Implementation class name	oracle.jdbc.pool.OracleConnectionPoolDataSource
URL	jdbc:oracle:thin:@9.181.84.136:1521:ord
Data store helper class name	com.ibm.websphere.radapter.Oracle10gDataStoreHelper
Use this data source in container managed persistence (CMP)	true
Component-managed authentication alias	n1Node01/Alias_Oracle
Mapping-configuration alias	(none)
Container-managed authentication alias	(none)

Previous **Finish** **Cancel**

16. Click **Save** to save the changes.

Cell=localhostNode01Cell, Profile=AppSrv01



17. Select the newly created data source and click **Test connection**.

Cell=localhostNode01Cell, Profile=AppSrv01

JDBC providers

[JDBC providers](#) > [Oracle JDBC Driver](#) > Data sources

Use this page to edit the settings of a datasource that is associated with your selected JDBC provider. The datasource object supplies your application with connections for accessing the database. Learn more about this task in a [guided activity](#). A guided activity provides a list of task steps and more general information about the topic.

Preferences

New	Delete	Test connection	Manage state...
Select	Name ▾	JNDI name ▾	Scope ▾
Provider ▾	Description ▾	Category ▾	

You can administer the following resources:

<input checked="" type="checkbox"/>	Oracle JDBC Driver DataSource	OracleDS	Node=n1Node01	Oracle JDBC Driver	New JDBC Datasource	
-------------------------------------	---	----------	---------------	--------------------	---------------------	--

Total 1

The connection should succeed as indicated by the message shown in the following figure. If you experience problems with the test connection, refer to the “Troubleshooting” section.

Cell=localhostNode01Cell, Profile=AppSrv01

JDBC providers

[JDBC providers](#) > [Oracle JDBC Driver](#) > Data sources

Messages

The test connection operation for data source Oracle JDBC Driver DataSource on server server1 at node n1Node01 was successful.

[Information](#)

Use this page to edit the settings of a datasource that is associated with your selected JDBC provider. The datasource object supplies your application with connections for accessing the database. Learn more about this task in a [guided activity](#). A guided activity provides a list of task steps and more general information about the topic.

Preferences

New	Delete	Test connection	Manage state...
Select	Name ▾	JNDI name ▾	Scope ▾
Provider ▾	Description ▾	Category ▾	

You can administer the following resources:

<input type="checkbox"/>	Oracle JDBC Driver DataSource	OracleDS	Node=n1Node01	Oracle JDBC Driver	New JDBC Datasource	
--------------------------	---	----------	---------------	--------------------	---------------------	--

Total 1

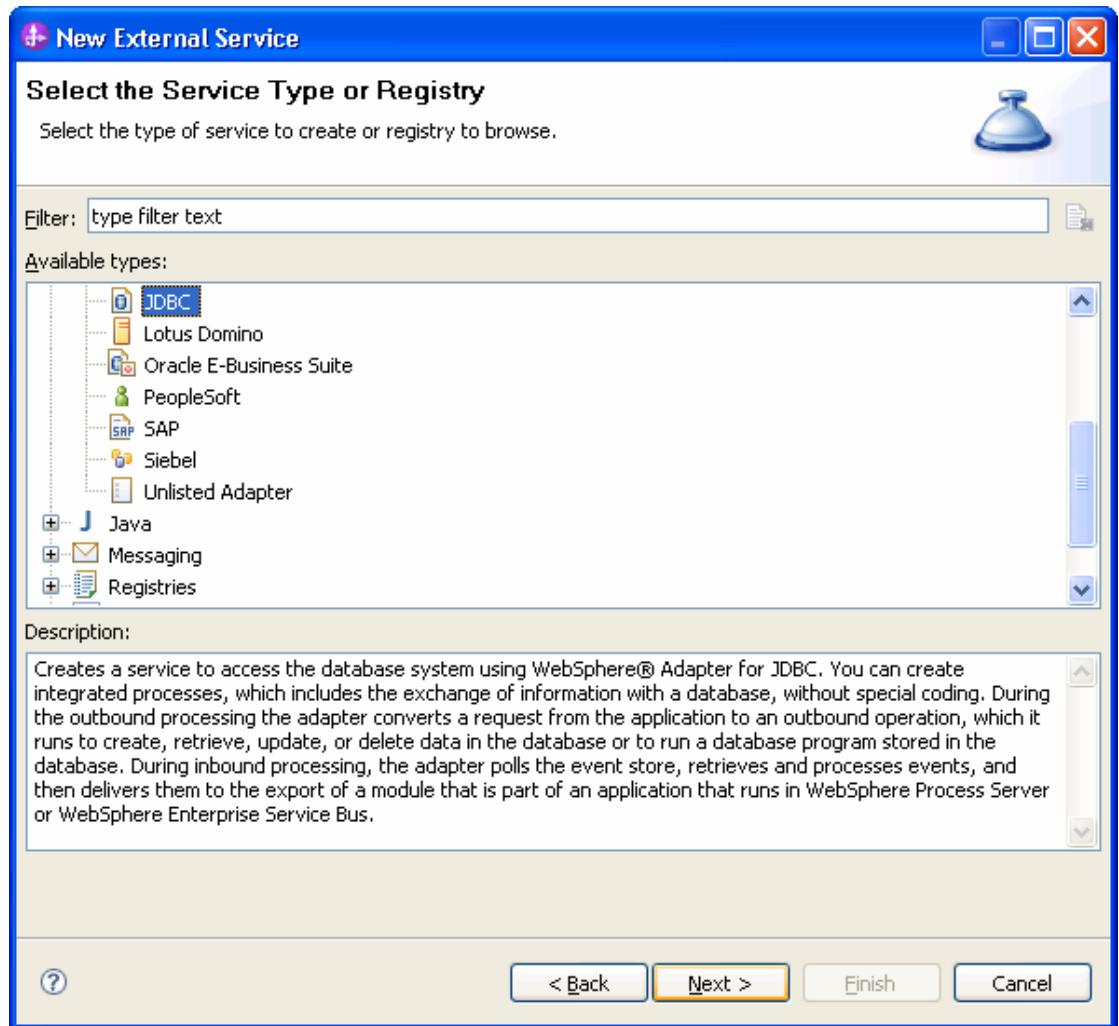
The data source is created and it will be used by the adapter to connect to the database.

Configure the adapter for inbound processing

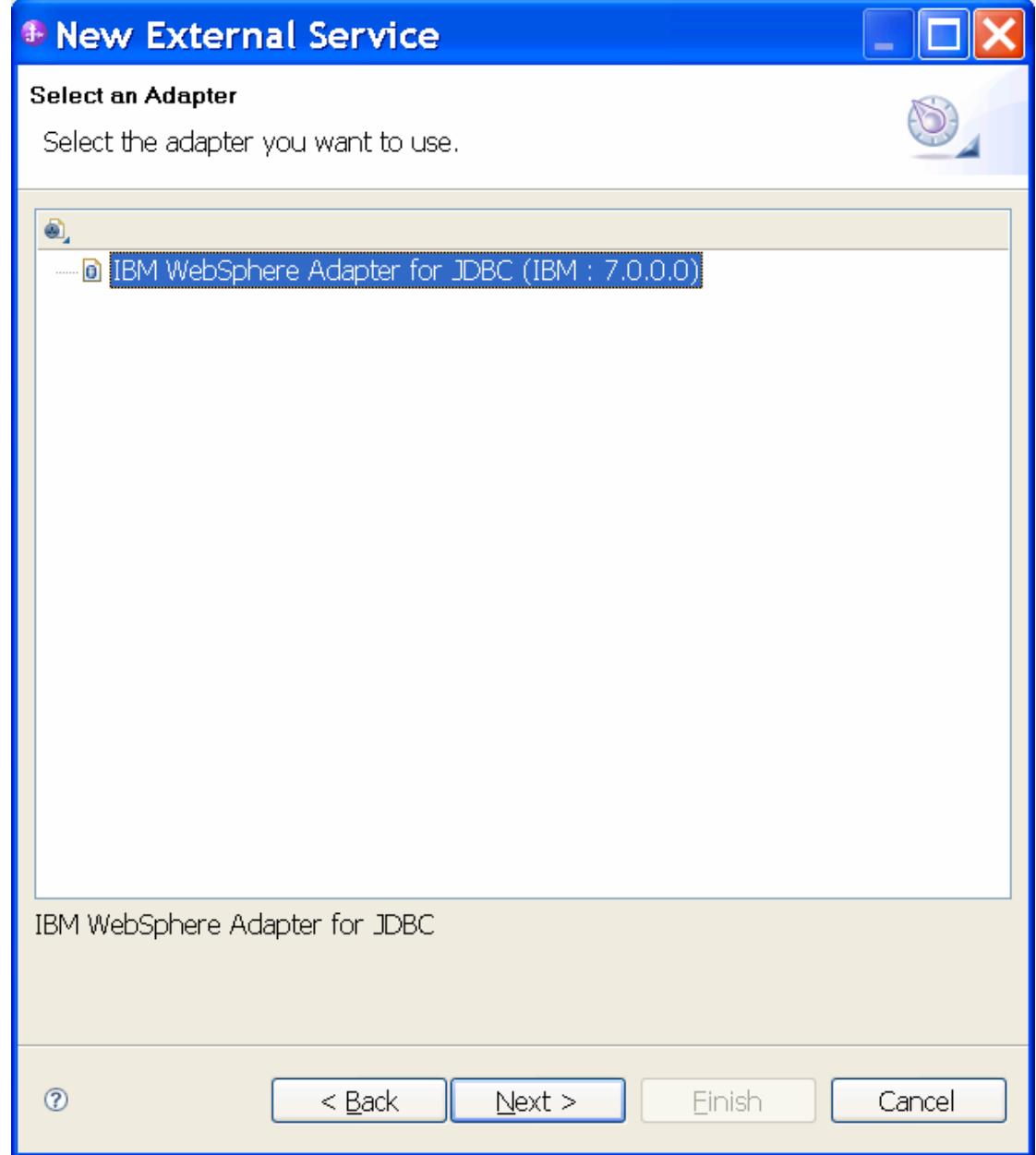
WebSphere software

Run the external service wizard to specify business objects, services, and configuration details.

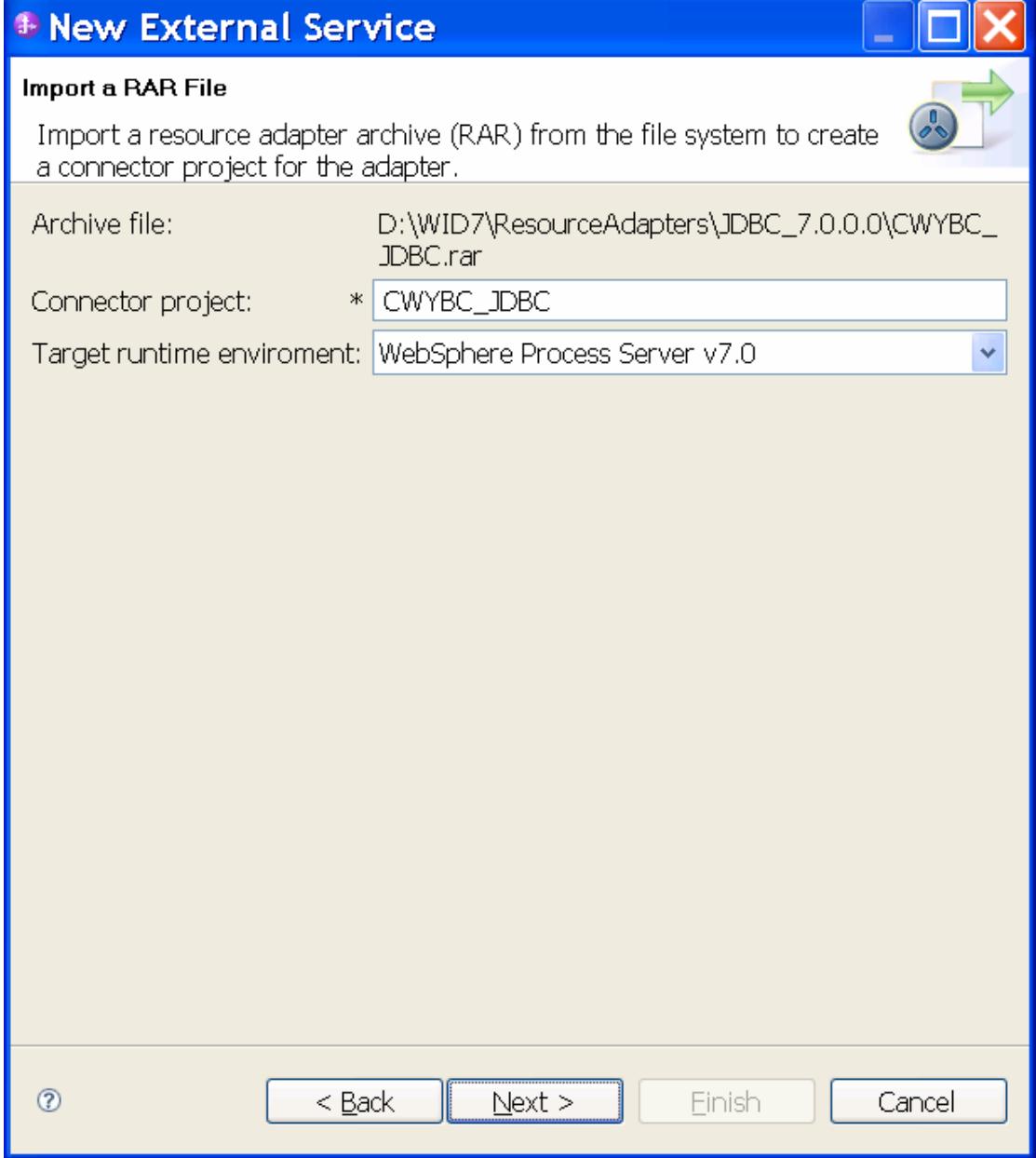
1. Switch to the Business Integration Perspective in WebSphere Integration Developer by selecting **Window -> Open Perspective Business Integration**.
2. Start the external service wizard by selecting **File-> New -> External Service**.
3. In the **Available Types** area, select **Adapters > JDBC** and click **Next**.



4. Select the **IBM WebSphere Adapter for JDBC (IBM: 7.0.0.0)** and click **Next**.



5. In the **Connector project** field, enter **CWYBC_JDBC**.
6. In the **Target runtime environment** field, select the appropriate runtime and click **Next**.



New External Service

Import a RAR File

Import a resource adapter archive (RAR) from the file system to create a connector project for the adapter.



Archive file: D:\WID7\ResourceAdapters\JDBC_7.0.0.0\CWYBC_JDBC.rar

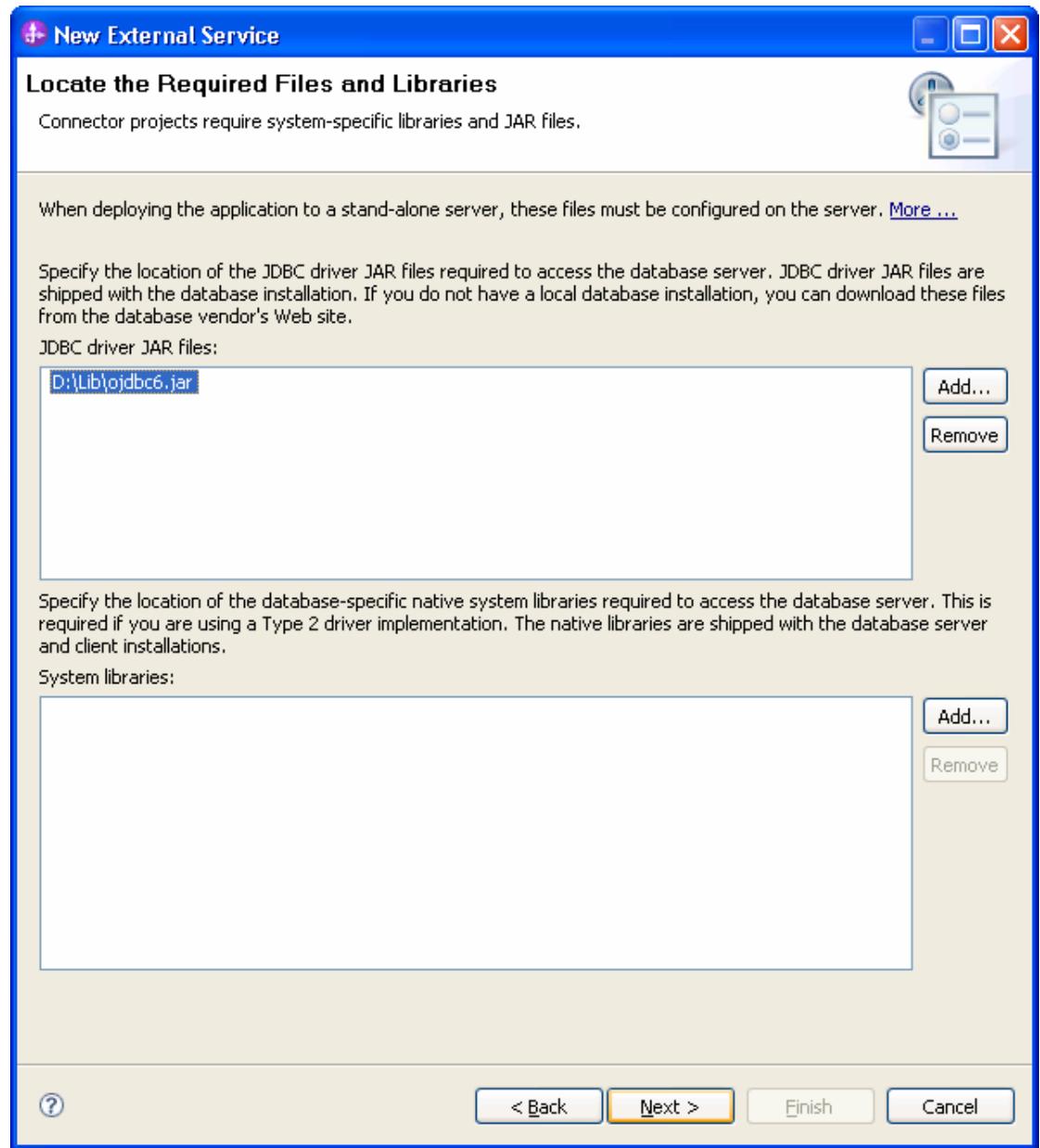
Connector project: * CWYBC_JDBC

Target runtime environment: WebSphere Process Server v7.0

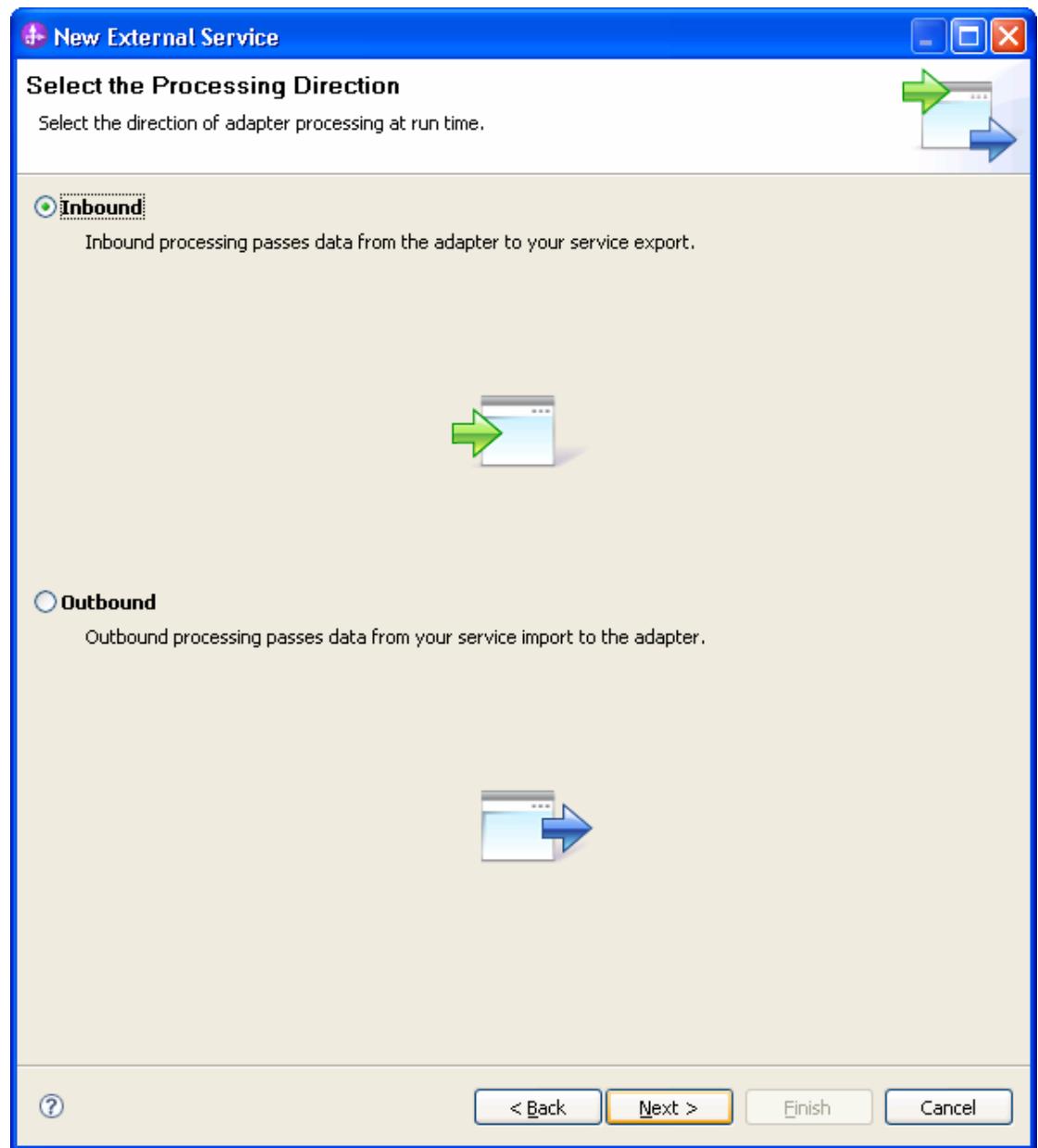
[?](#)[< Back](#)[Next >](#)[Finish](#)[Cancel](#)

WebSphere software

7. In the **JDBC driver JAR files** field, click **Add**, to add the JDBC driver class to connect to the database. Browse to select the driver JAR file and click **Next**.



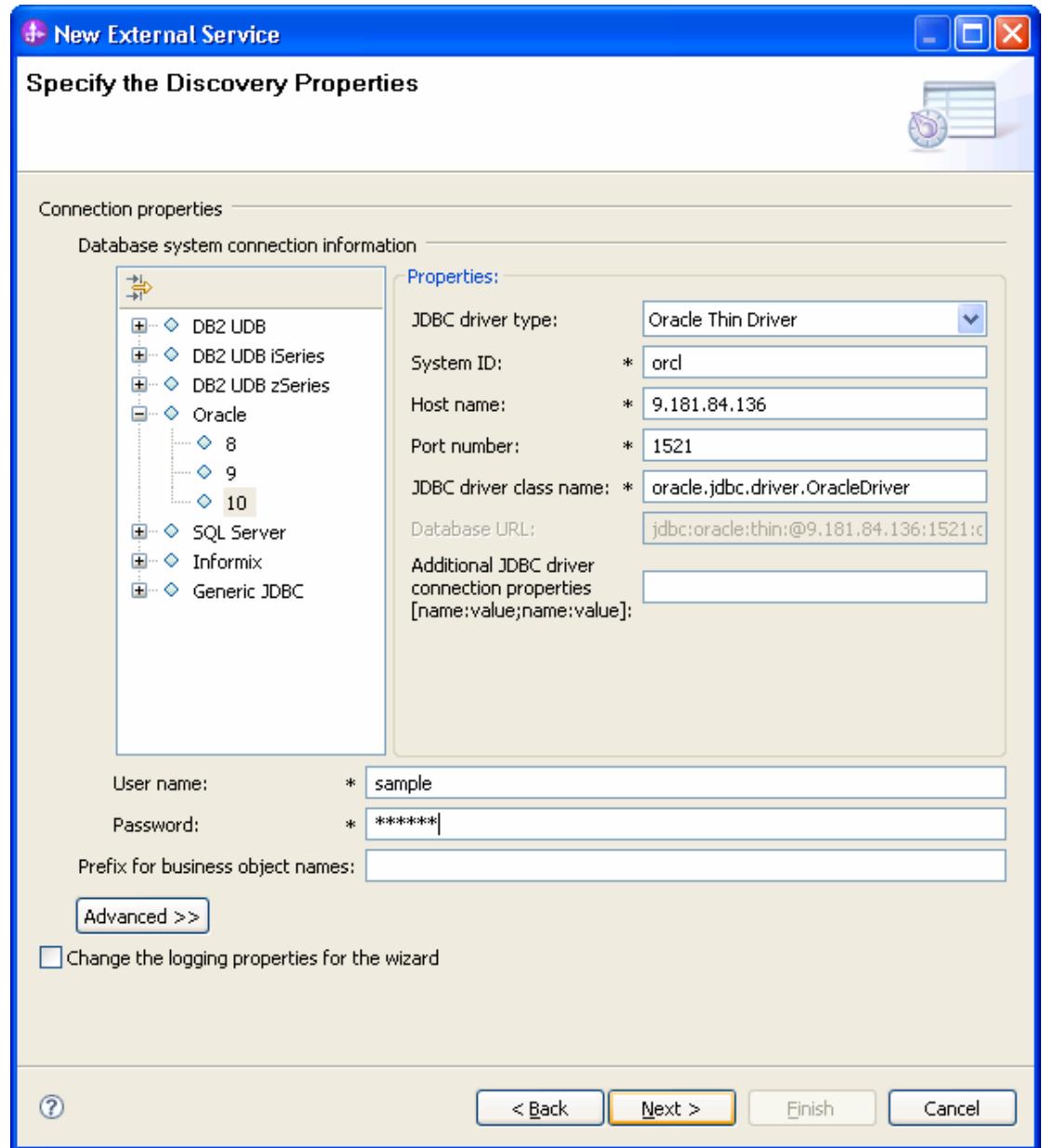
8. Select **Inbound** and click **Next**.



Set connection properties for the external service wizard

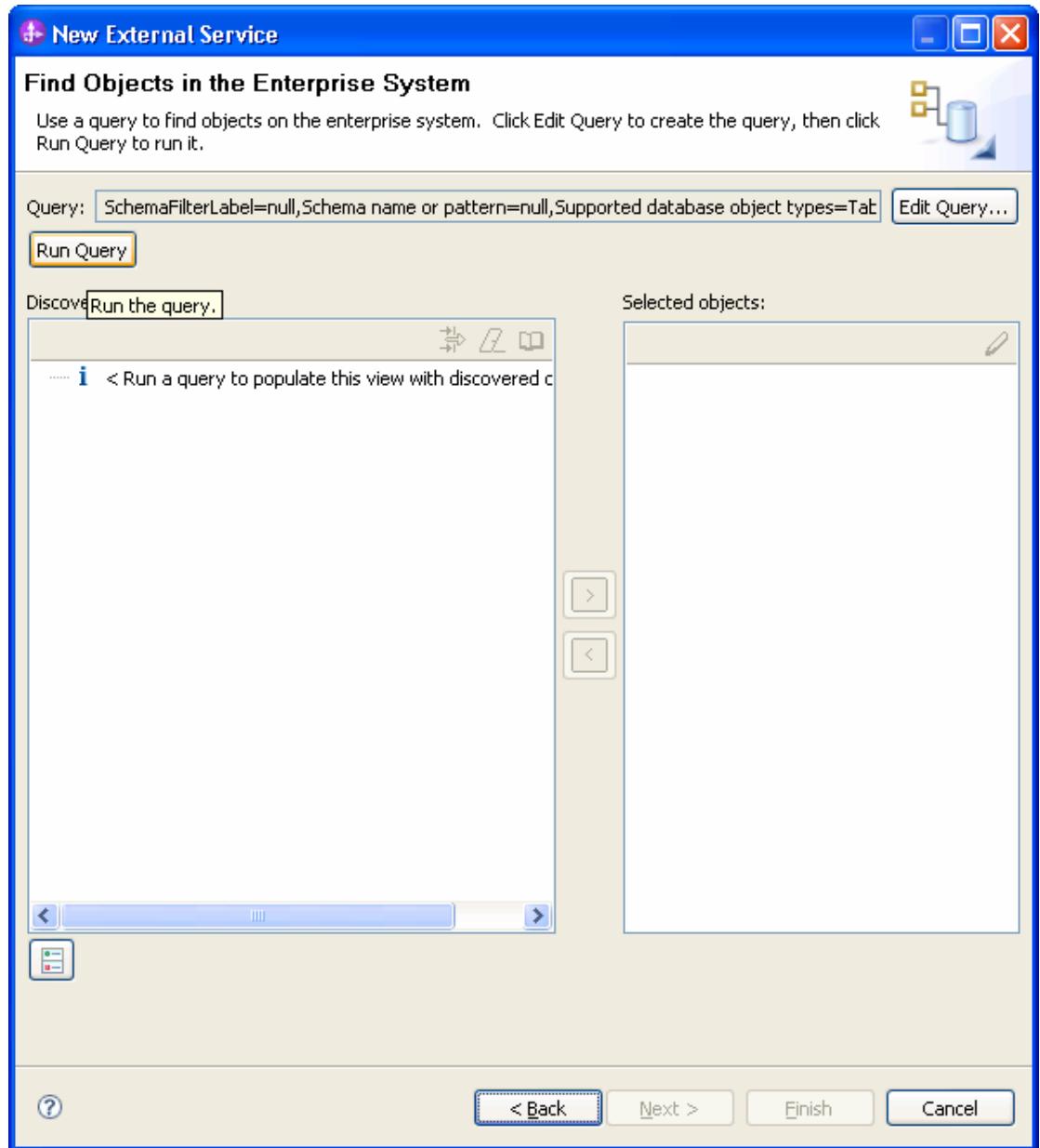
To connect to the Oracle database:

1. Expand the **Oracle** node in the **Database system connection information area** and select **10**.
2. Enter values in the **System ID**, **Host name**, **Port number**, **User name** and **Password** fields, and click **Next**.

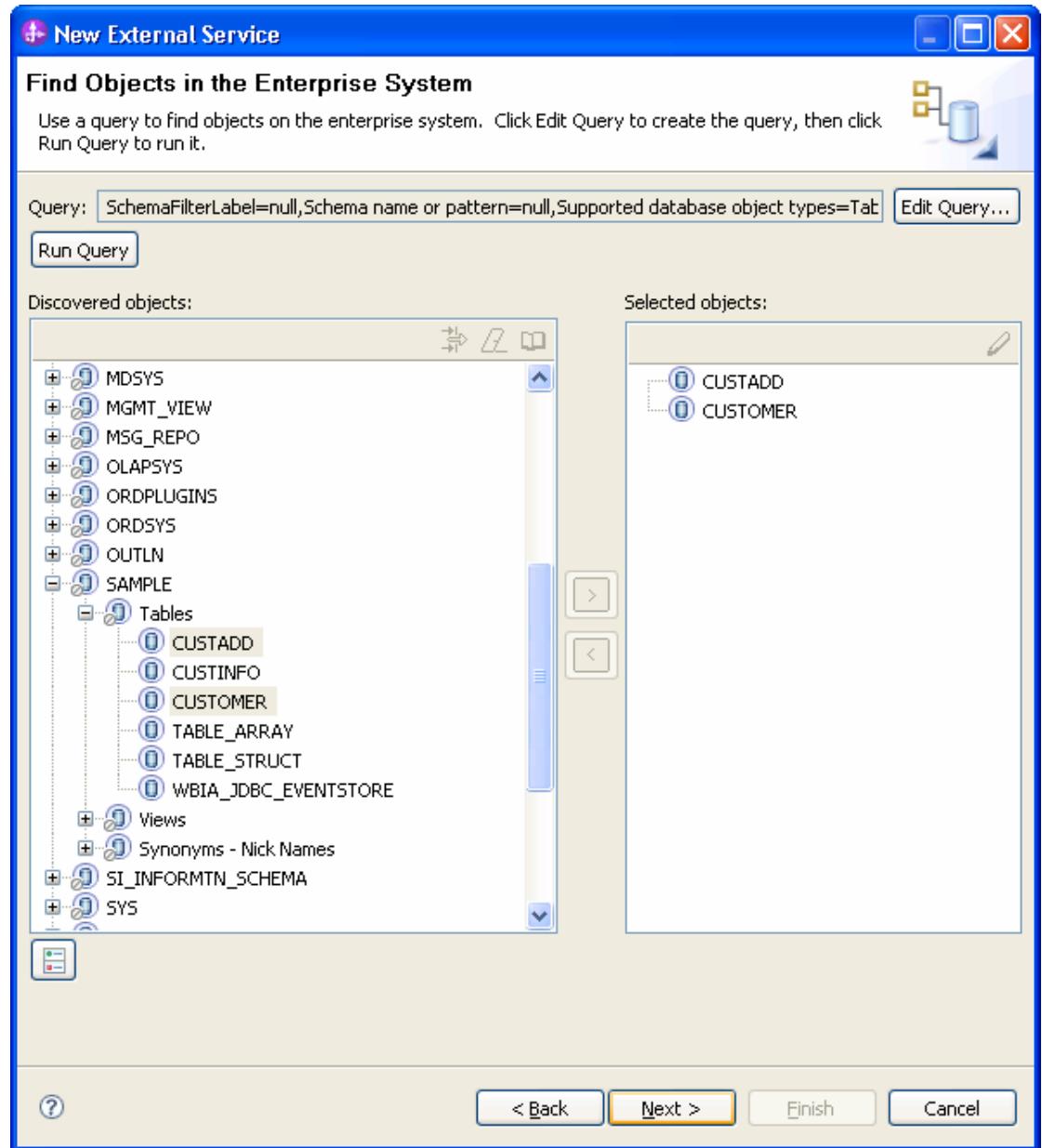


Select the business objects to be used with the adapter

1. In the Find Objects in Enterprise System window, click **Run Query**.



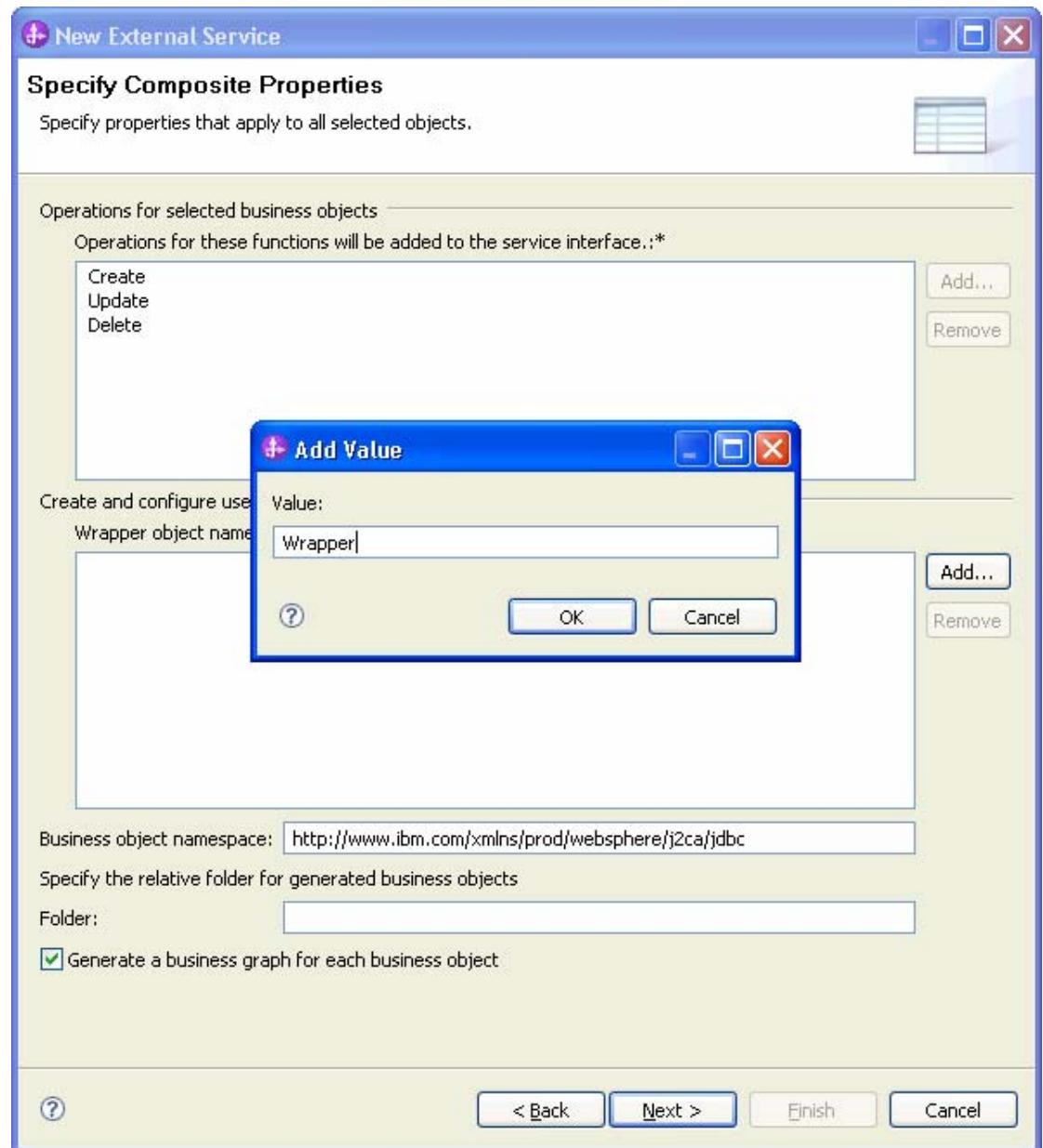
2. In the Discovered objects pane, expand the **SAMPLE** (for this tutorial only) node, select **Tables** and expand it.
 3. Select the CUSTOMER and CUSTADD tables and click .
 4. Click **Next**.



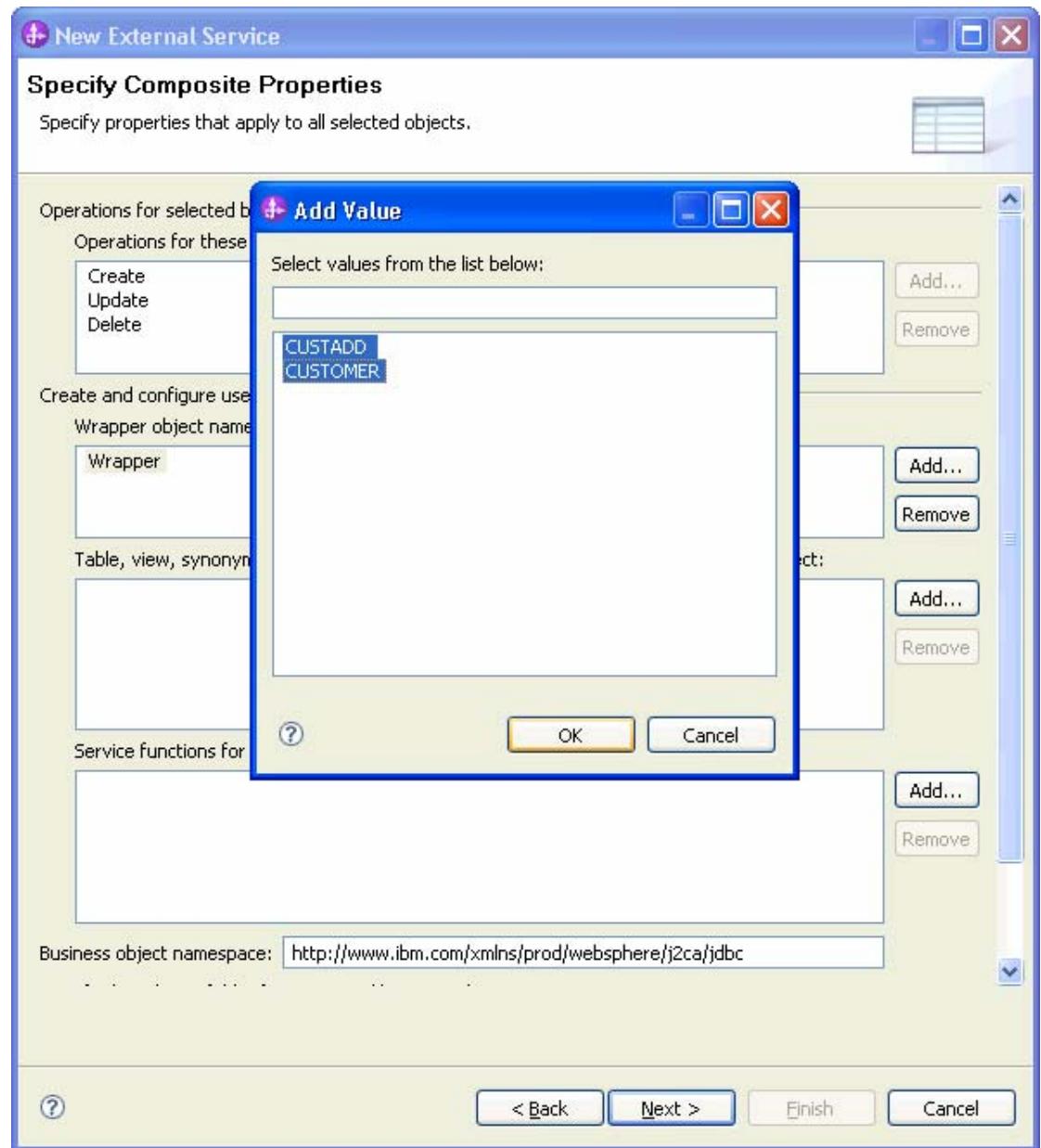
Generate business object definitions and related artifacts

Follow these steps to generate the business object definitions.

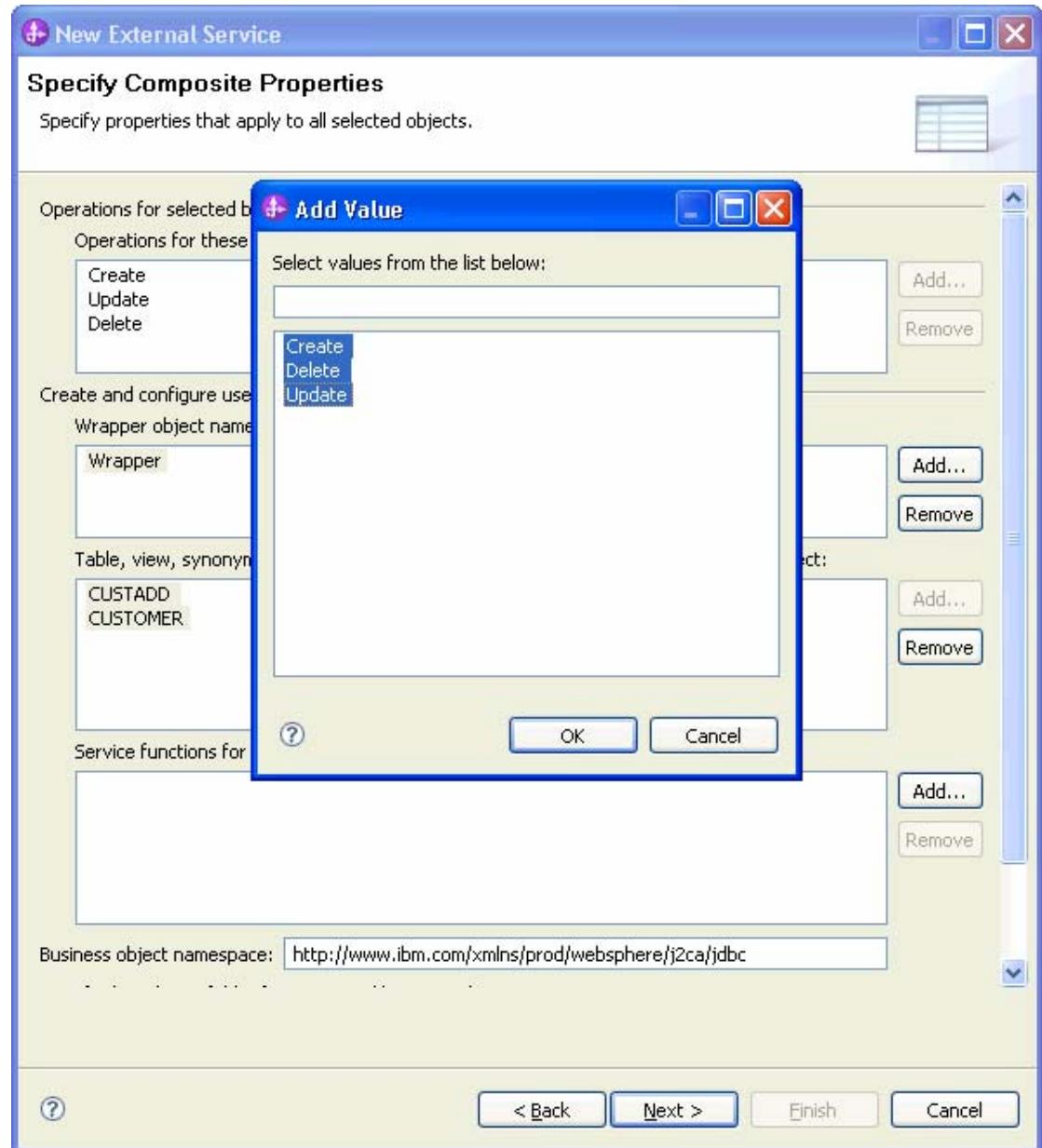
1. In the Specify Composite Properties window, click **Add** and enter the name for the new wrapper business object. Click **OK**.



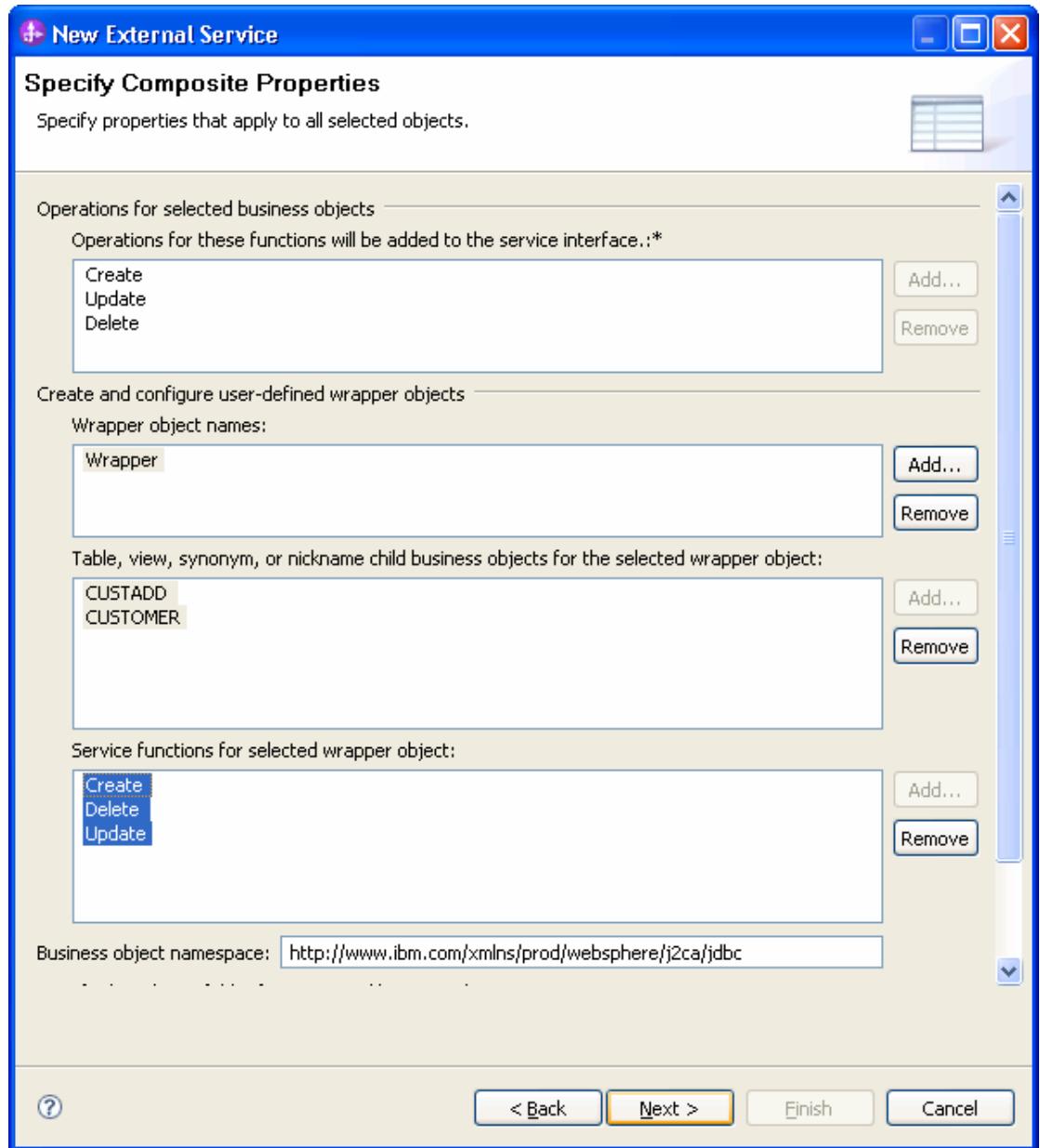
2. In the **Table, view, synonym, or nickname child objects for the selected wrapper** area, click **Add** to add CUSTOMER and CUSTADD table business objects for the wrapper.



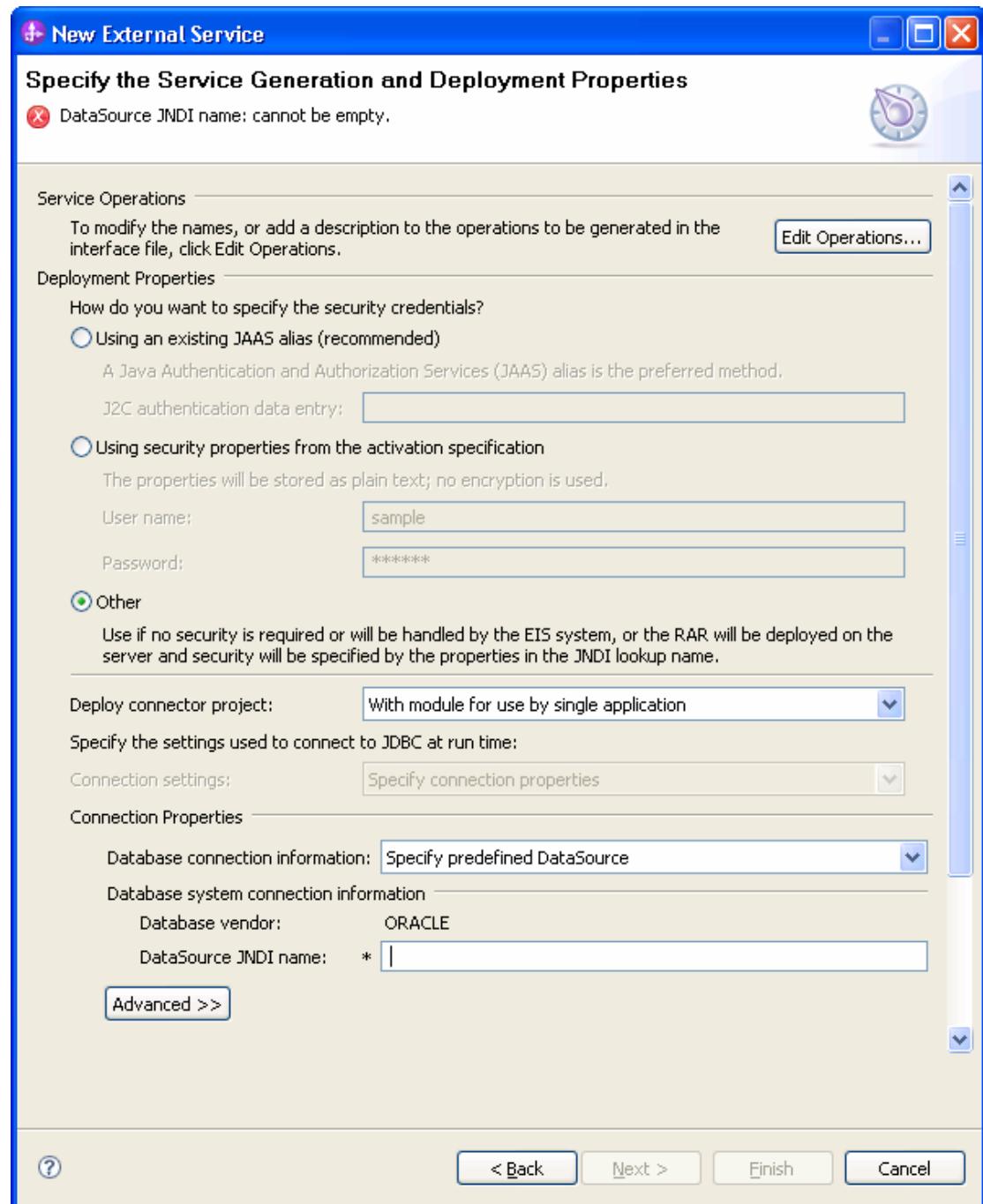
3. In the Service **functions for selected wrapper object** area, click **Add** to add functions for the wrapper.



4. Accept the default for the other fields, and click **Next**.



5. In the Specify the Service Generation and Deployment Properties window, select the security credential as **Other**. Select **Specify predefined DataSource** from the **Database connection information** list.



6. In the **DataSource JNDI name** field, enter the JNDI name of the data source, which you created in the previous section. Click **Next**.

New External Service

Specify the Service Generation and Deployment Properties

Specify properties for generating the service and running it on the server.



Service Operations

To modify the names, or add a description to the operations to be generated in the interface file, click Edit Operations...

Deployment Properties

How do you want to specify the security credentials?

Using an existing JAAS alias (recommended)
A Java Authentication and Authorization Services (JAAS) alias is the preferred method.
J2C authentication data entry:

Using security properties from the activation specification
The properties will be stored as plain text; no encryption is used.
User name:
Password:

Other
Use if no security is required or will be handled by the EIS system, or the RAR will be deployed on the server and security will be specified by the properties in the JNDI lookup name.

Deploy connector project:

Specify the settings used to connect to JDBC at run time:

Connection settings:

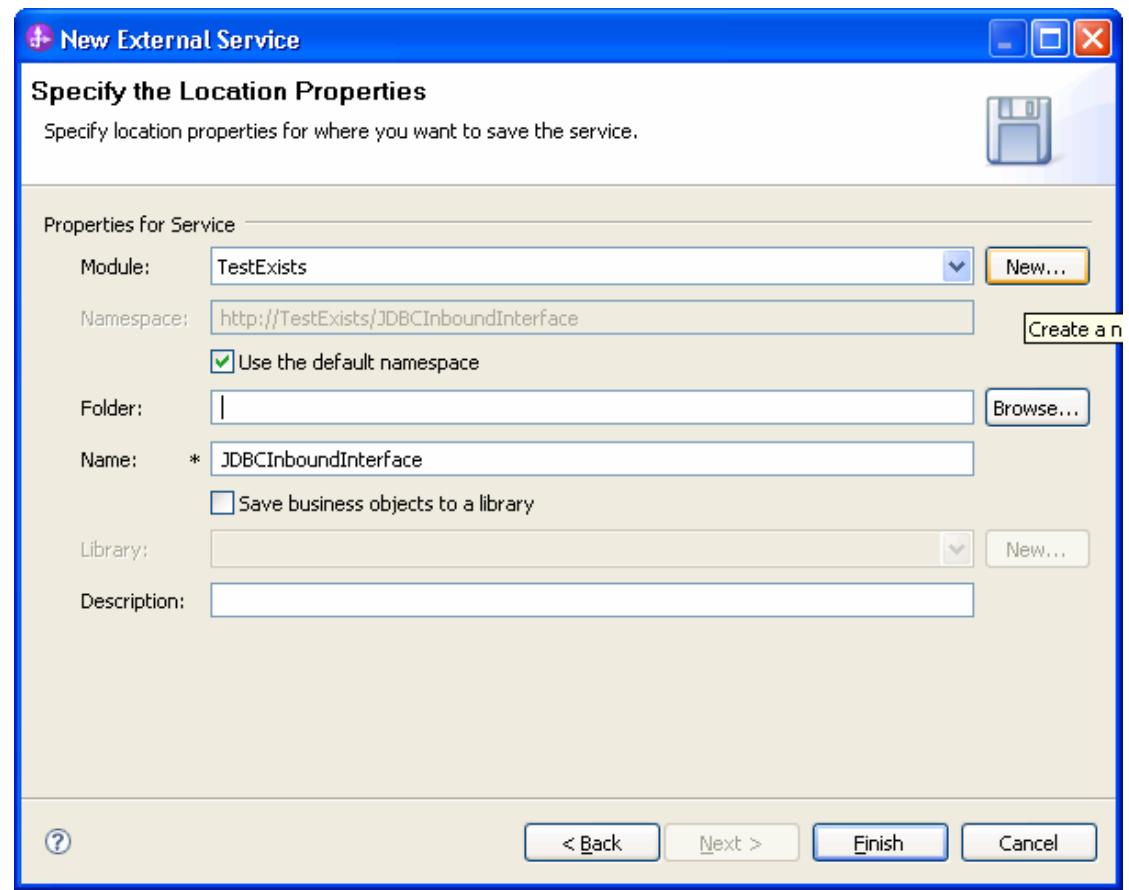
Connection Properties

Database connection information:

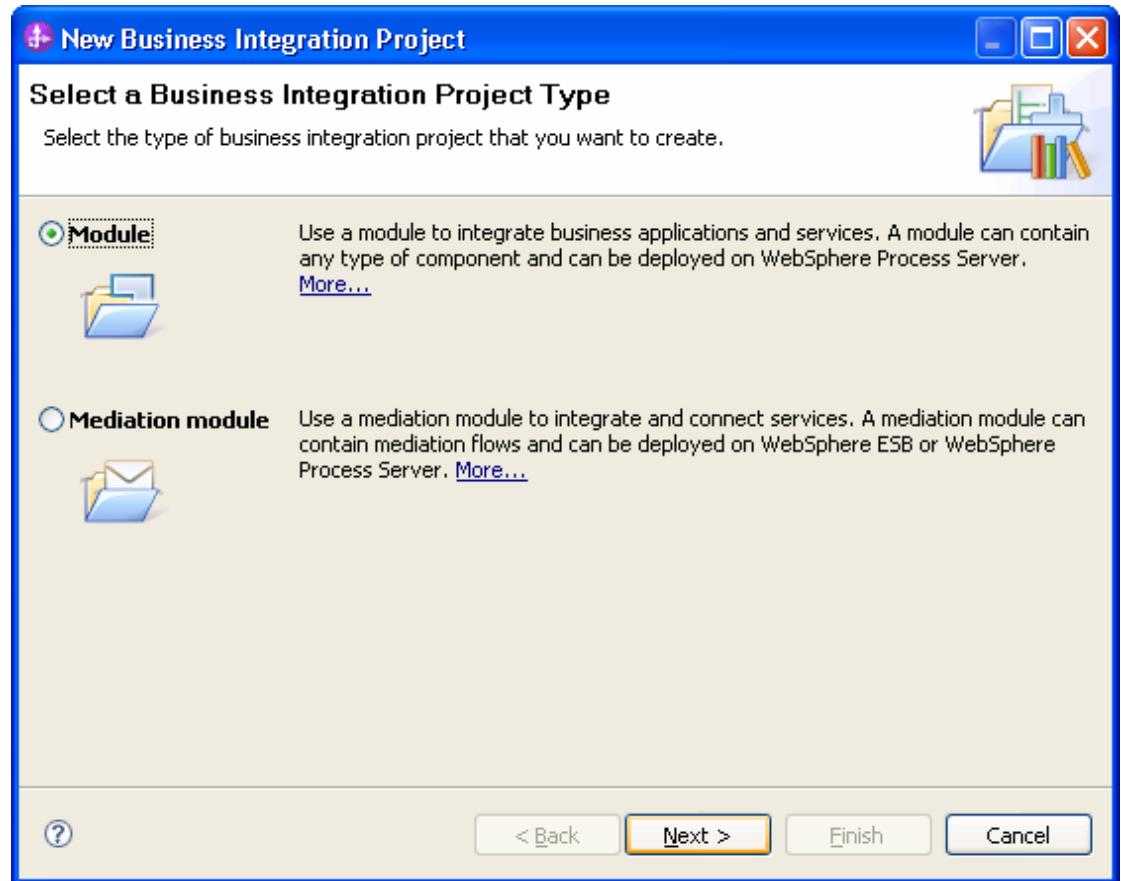
Database system connection information

Database vendor: ORACLE
DataSource JNDI name: *

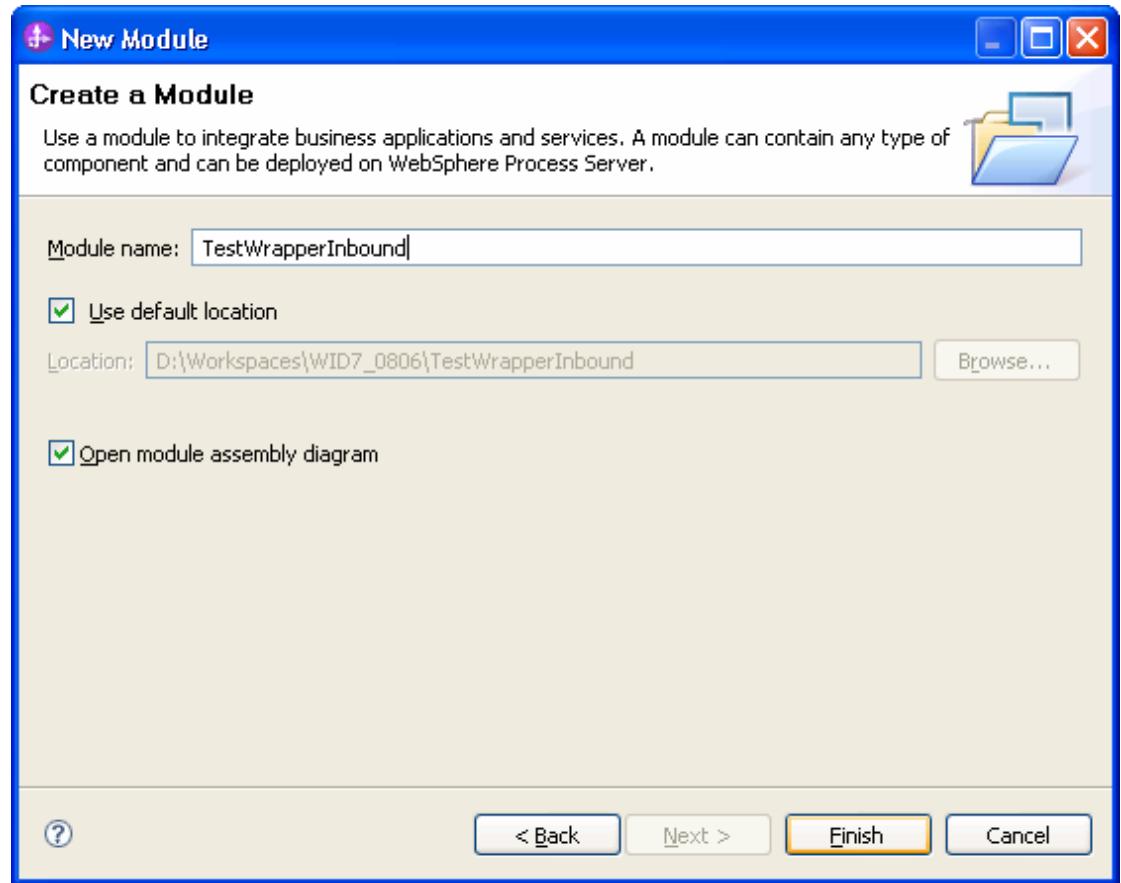
7. Click **New** in the Specify the Location Properties window.



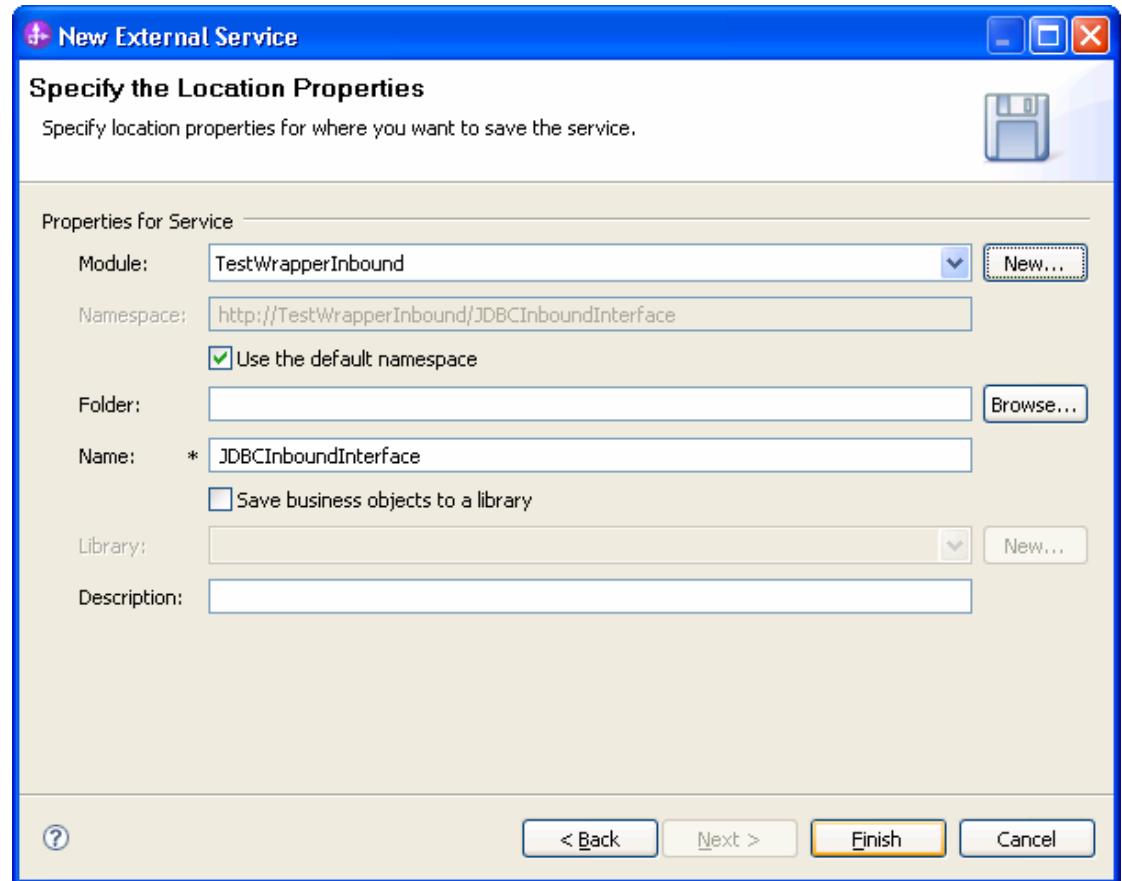
8. In the Select a Business Integration Project Type window, select **Module** and click **Next**.



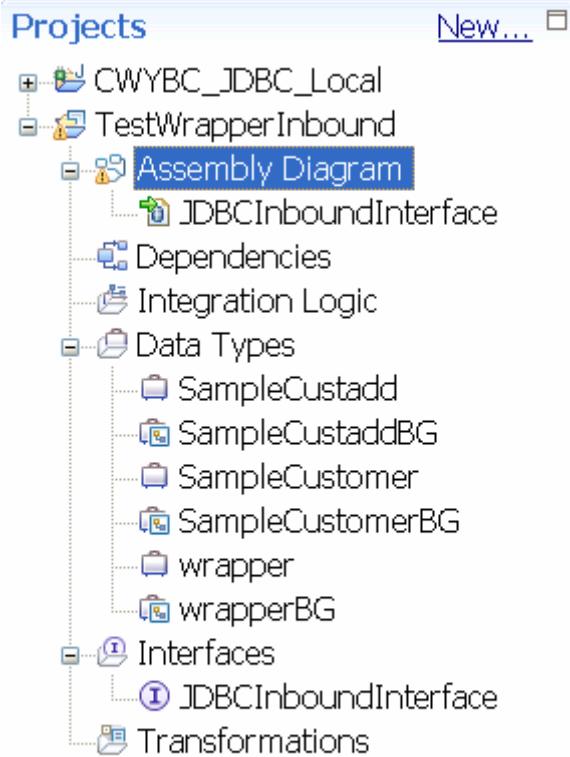
9. In the Create a Module window, type **TestWrapperInbound** in the **Module Name** field, click **Finish**.



10. In the Specify the Location Properties window, click **Finish** to finish the wizard.



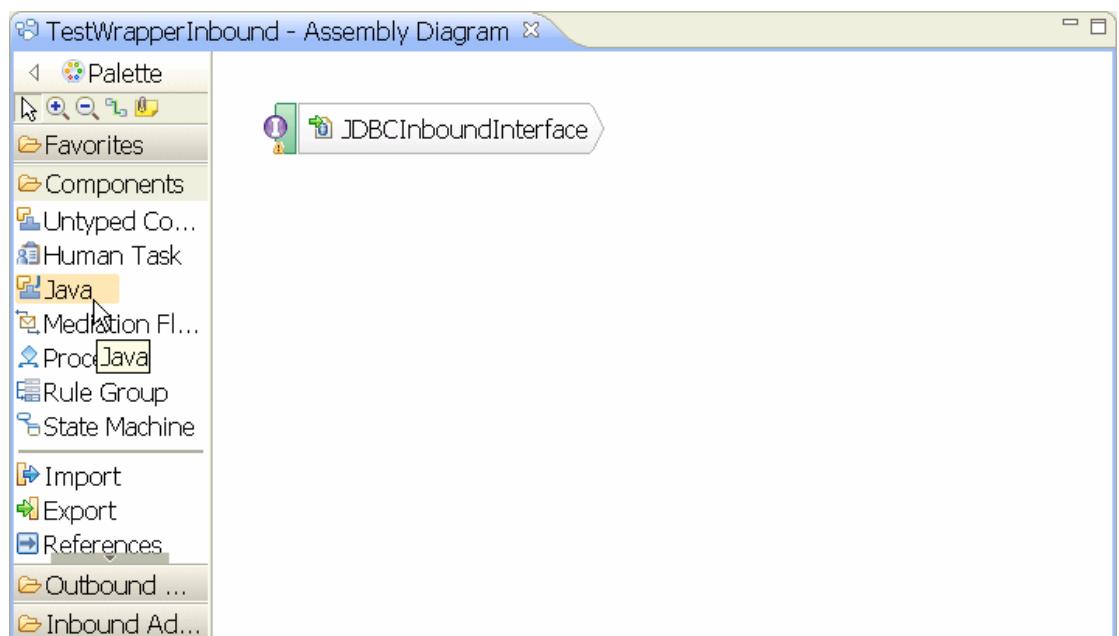
11. Expand the created Business Integration Project and verify whether the artifacts are generated correctly.



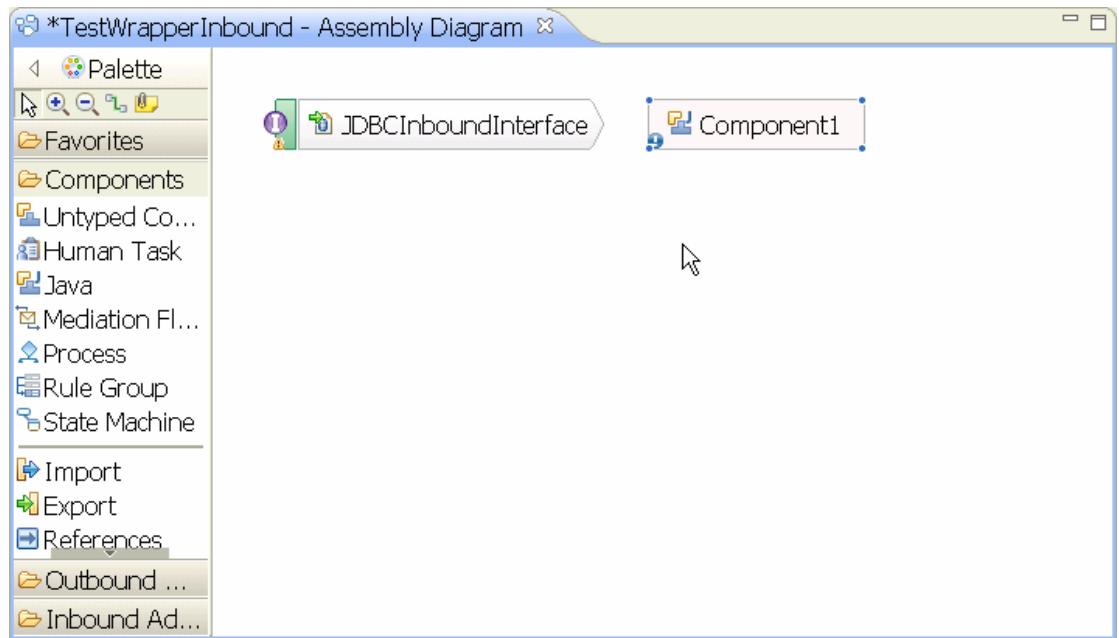
Set up the components to be part of the inbound environment

Next, we add and set up components that to be part of the inbound environment.

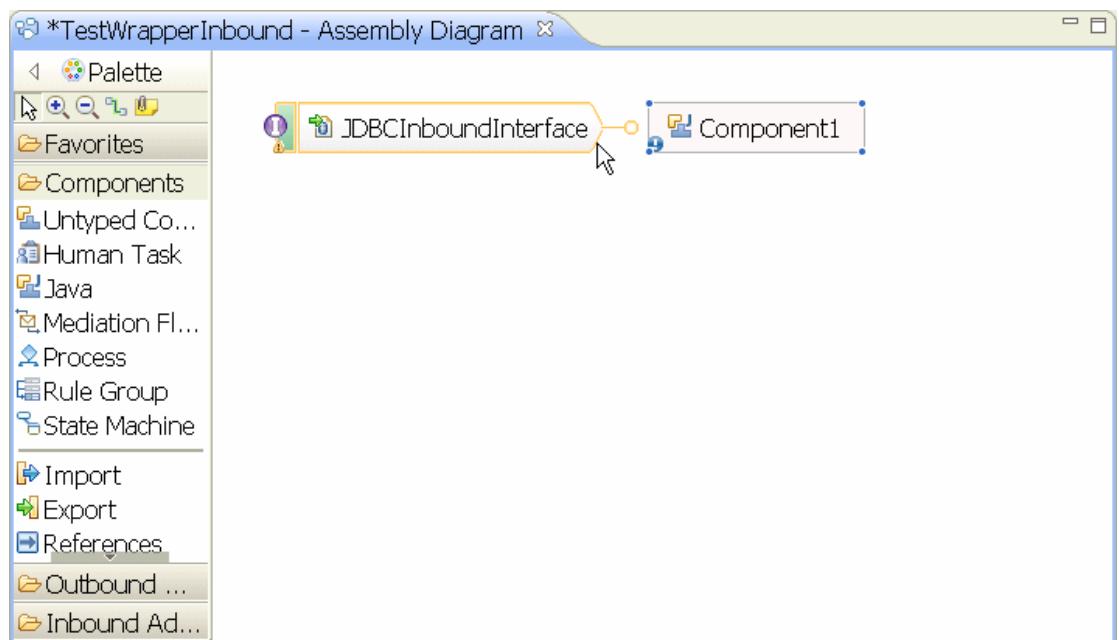
1. In the **Assembly Diagram**, in the Palette, expand **Components** and click **Java** component.



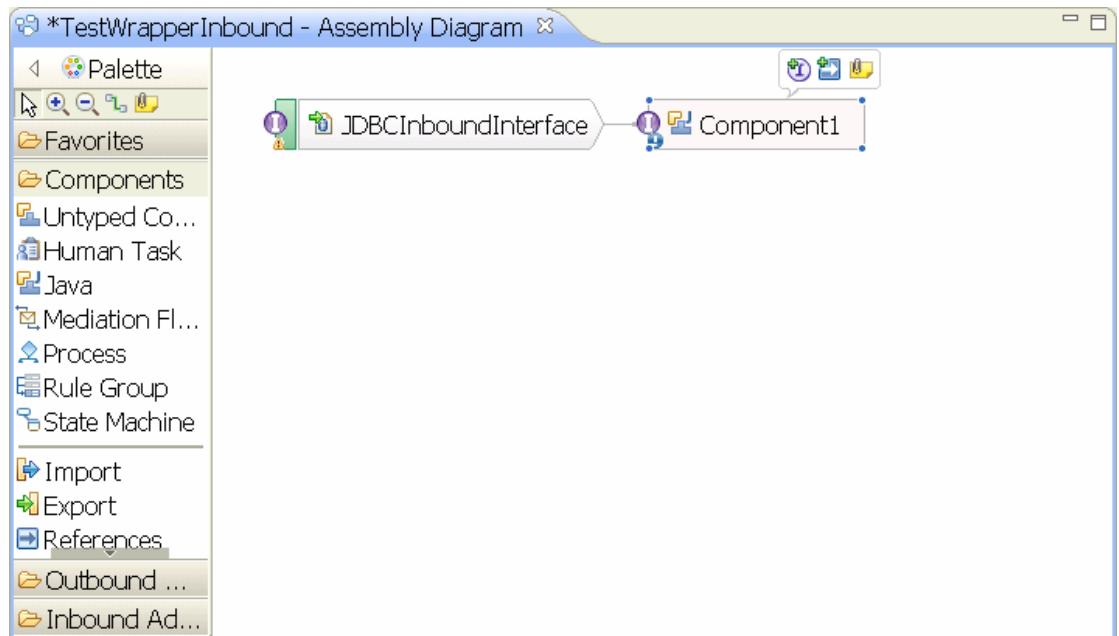
2. Click anywhere in the **TestWrapperInbound -Assembly Diagram** window (white part) to create the **Java** component, **Component1**.



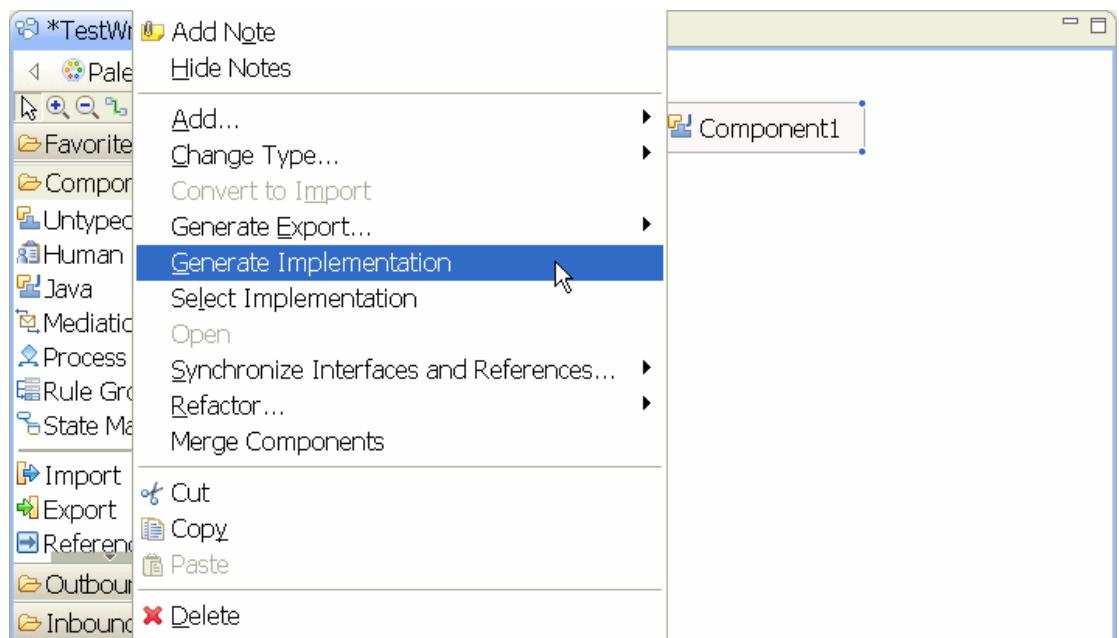
- To wire **JDBCInboundInterface** to **Component1** hover the cursor over the right end of **JDBCInboundInterface** until a yellow wire appears.



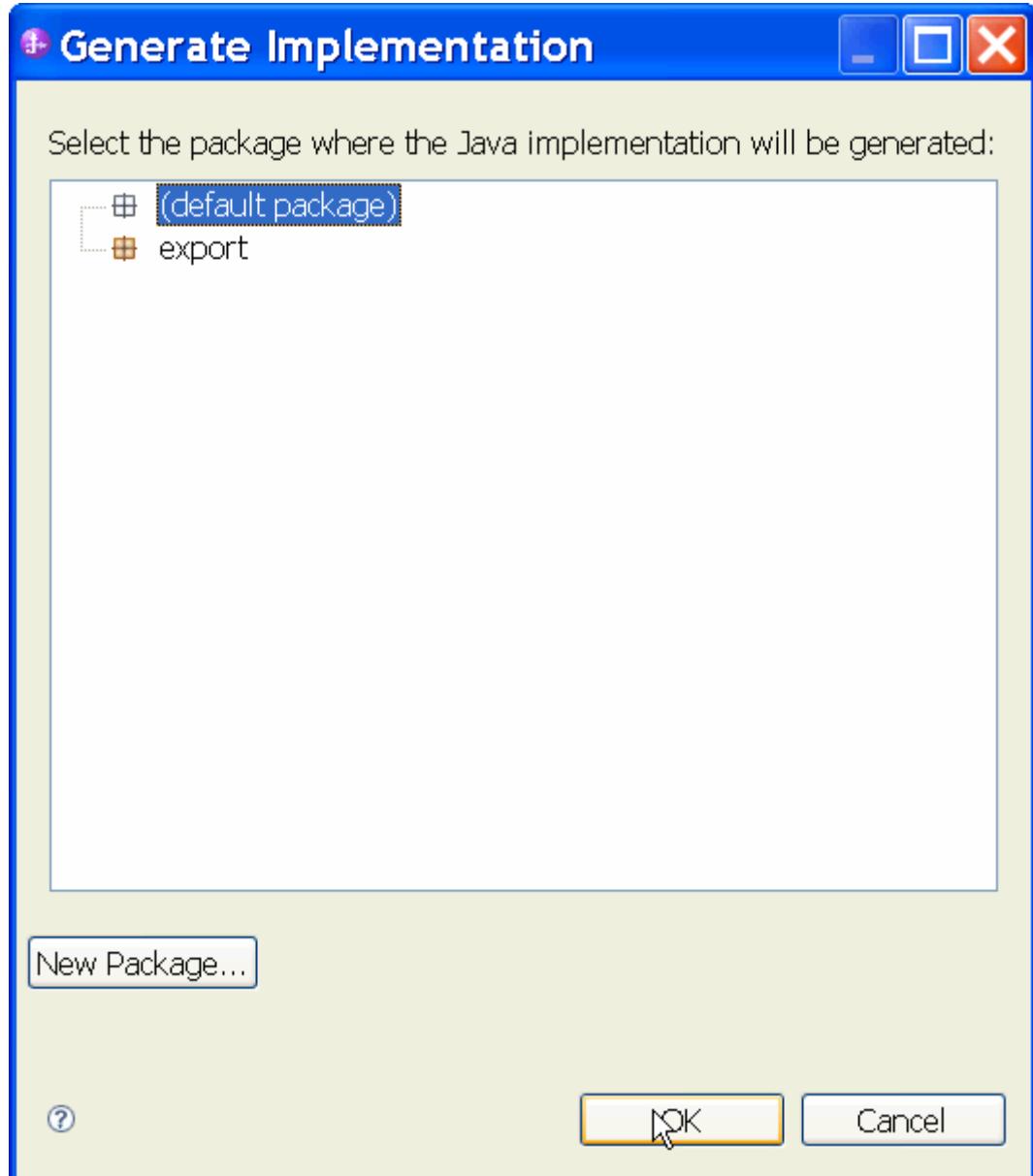
- Click on the yellow wire and drag it to the left end of **Component1**. When the Add Wire pop-up window displays click **OK**.



5. Click **File > Save** from the toolbar to save changes made.
6. Right-click on **Component1** in the **Assembly Diagram** and select **Generate implementation**.



7. In the Generate Implementation window, select **default package** and click **OK**.



8. The Java Editor will open showing the **Component1Impl.java** file.

The screenshot shows the WebSphere Studio IDE interface. The title bar reads "WebSphere software" and "Component1Impl.java". The main window displays the Java code for Component1Impl.java. The code includes imports, a constructor, a service reference method, and a commented-out wrapper creation method.

```
*TestWrapperInbound - Assembly Diagram Component1Impl.java

import commonj.sdo.DataObject;

public class Component1Impl {
    /**
     * Default constructor.
     */
    public Component1Impl() {
        super();
    }

    /**
     * Return a reference to the component service instance for this implementation class. This method should be used when passing this service to a partner or if you want to invoke this component service asynchronously.
     *
     * @generated (com.ibm.websphere.java)
     */
    @SuppressWarnings("unused")
    private Object getMyService() {
        return (Object) ServiceManager.INSTANCE.locateService("self");
    }

    /**
     * Method generated to support implementation of operation "createrwrapperBG" named "JDBCInboundInterface".
     *
     * The presence of commonj.sdo.DataObject as the return type and/or as a parameter type conveys that it is a complex type. Please refer to the WSDL Definition on the type of input, output and fault(s).
     */
    public void createrwrapperBG(DataObject createrwrapperBGInput) {
        // To get or set attributes for DataObject createrwrapperBGInput, use the
        // To set a string attribute in createrwrapperBGInput, use createrwrapperBGInput
        // To get a string attribute in createrwrapperBGInput use createrwrapperBGInput
    }
}
```

9. Scroll down and locate the **createrwrapperBG** method.

10. Replace the entire method so that it looks like the one shown below:

```

public void createwrapperBG(
    DataObject wrapper =
createwrapperBGInput.getDataObject( "wrapper" );

System.out.println( "-----" );
System.out.println( "Wrapper was created." );
DataObject customer = (DataObject)
wrapper.getList( "customerobj" ).get( 0 );
DataObject addr = (DataObject)
wrapper.getList( "custaddobj" ).get( 0 );
System.out.println( "CUSTOMER info as below:" );
System.out.println( "PKEY is: " +
    customer.getString( "pkey" ) );
System.out.println( "FNAME is: " +
    customer.getString( "fname" ) );
System.out.println( "LNAME is: " +
    customer.getString( "lname" ) );
System.out.println( "CCODE is: " +
    customer.getString( "ccode" ) );
System.out.println();
System.out.println( "CUSTADD info as below:" );
System.out.println( "ADDRID is: " +
    addr.getString( "addrid" ) );
System.out.println( "CUSTID is: " +
    addr.getString( "custid" ) );
System.out.println( "CITY is: " +
    addr.getString( "city" ) );
System.out.println( "ZIPCODE is: " +
    addr.getString( "zipcode" ) );

System.out.println( "-----" );
System.out.println();
}

```

11. Scroll down and locate the **deletewrapperBG** method.

12. Replace the entire method so that it looks like the one shown below:

```

public void deletewrapperBG(
DataObject deletewrapperBGInput) {
    DataObject wrapper =
deletewrapperBGInput.getDataObject( "wrapper" );
    System.out.println( "-----" );
    System.out.println( "Wrapper was deleted." );
    System.out.println( "PKEY of customer is: " +
        wrapper.getString( "wrapcustomerpkey" ) );
    System.out.println( "ADDRID custadd is: " +
        wrapper.getString( "wrapcustaddaddrid" ) );
    System.out.println( "-----" );
    System.out.println();
}

```

13. Scroll down and locate the **updatewrapperBG** method.

14. Replace the entire method so that it looks like the one shown below:

```
public void updatewrapperBG(
DataObject updatewrapperBGInput) {
    DataObject wrapper =
updatewrapperBGInput.getDataObject( "wrapper" );
    System.out.println( "-----" );
    System.out.println( "Wrapper was updated." );
    DataObject customer = (DataObject)
wrapper.getList( "customerobj" ).get( 0 );
    DataObject addr = (DataObject)
wrapper.getList( "custaddobj" ).get( 0 );
    System.out.println( "CUSTOMER info as below:" );
    System.out.println( "PKEY is: " +
customer.getString( "pkey" ) );
    System.out.println( "FNAME is: " +
customer.getString( "fname" ) );
    System.out.println( "LNAME is: " +
customer.getString( "lname" ) );
    System.out.println( "CCODE is: " +
customer.getString( "ccode" ) );
    System.out.println();
    System.out.println( "CUSTADD info as below:" );
    System.out.println( "ADDRID is: " +
addr.getString( "addrid" ) );
    System.out.println( "CUSTID is: " +
addr.getString( "custid" ) );
    System.out.println( "CITY is: " +
addr.getString( "city" ) );
    System.out.println( "ZIPCODE is: " +
addr.getString( "zipcode" ) );
    System.out.println( "-----" );
    System.out.println();
}
```

15. Click on **File > Save** from the toolbar to save the changes made.

Deploy the module to the test environment

After running the external service wizard, you will have an SCA module that contains an Enterprise Information System (EIS) export. You must install this SCA module in the WebSphere Integration Developer integration test client to deploy it. To do this, you must add the SCA module you created earlier to the server using the **Servers** view in WebSphere Integration Developer.

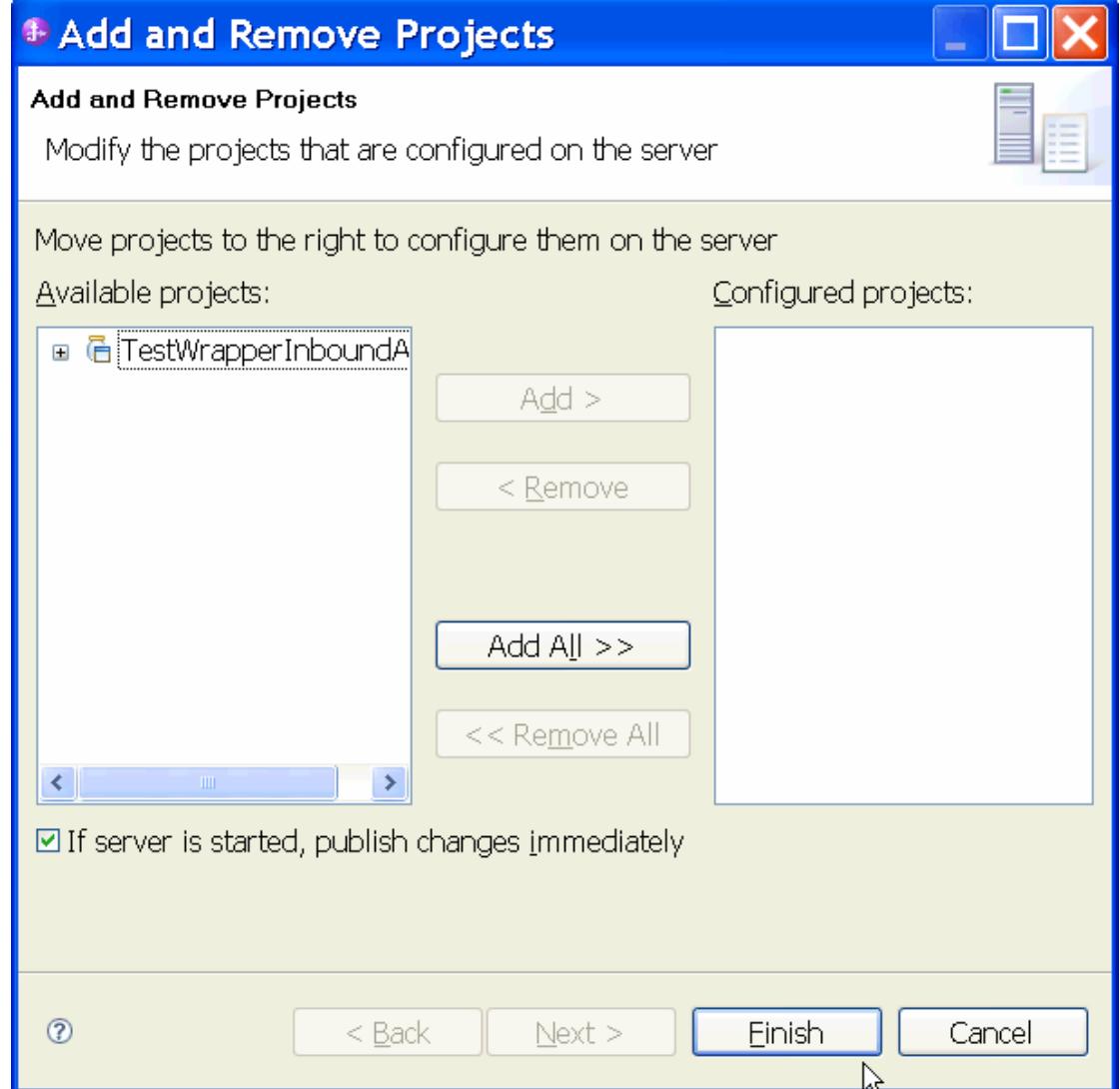
Steps for adding the SCA module to the server:

1. In WebSphere Integration Developer, switch to the **Servers** view by selecting from the toolbar **Window > Show View > Servers**.
2. In the **Servers** tab in the lower-right pane right click the server, and select **Start**.
3. After the server is started, right-click the server, and select **Add and Remove projects**.

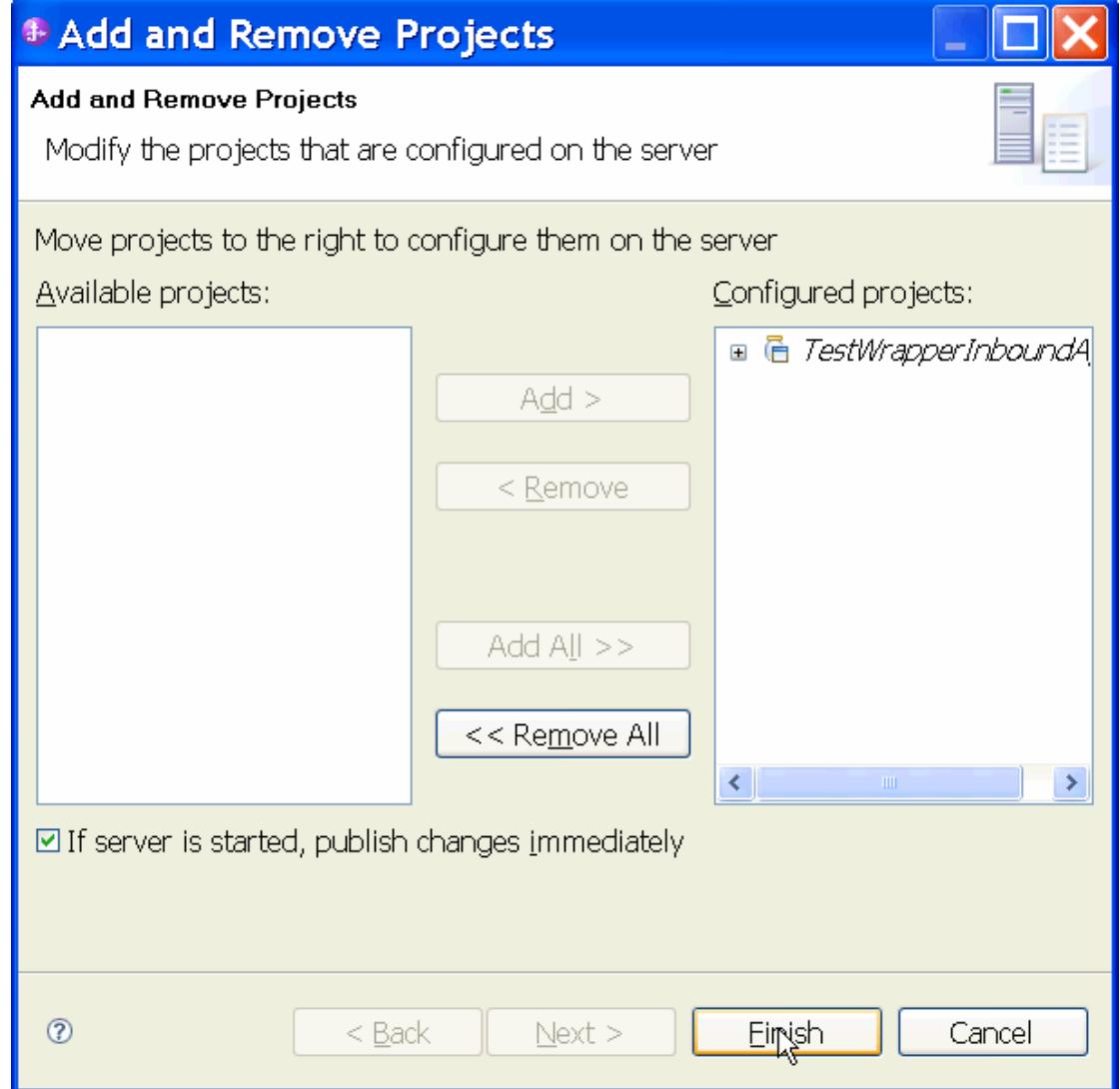
WebSphere software

New	▶
Open	F3
Show In	Alt+Shift+W ▶
Copy	Ctrl+C
Paste	Ctrl+V
✖ Delete	Delete
Rename	F2
✿ Restart in Debug	Ctrl+Alt+D
▶ Restart	Ctrl+Alt+R
⌚ Restart in Profile	
▣ Stop	Ctrl+Alt+S
Publish	Ctrl+Alt+P
Clean...	
Add and Remove Projects...	▶
Monitoring	▶
>Create tables and data sources	
Reconnect debug process	
View and publish the changes to the server	
Manage server profiles	
Server configuration	▶
Universal test client	▶
Administration	▶
Launch	▶
Add and Remove Integration Solution Projects	▶
Properties	Alt+Enter

A window is displayed that shows the available projects on the WebSphere Integration Developer workspace.



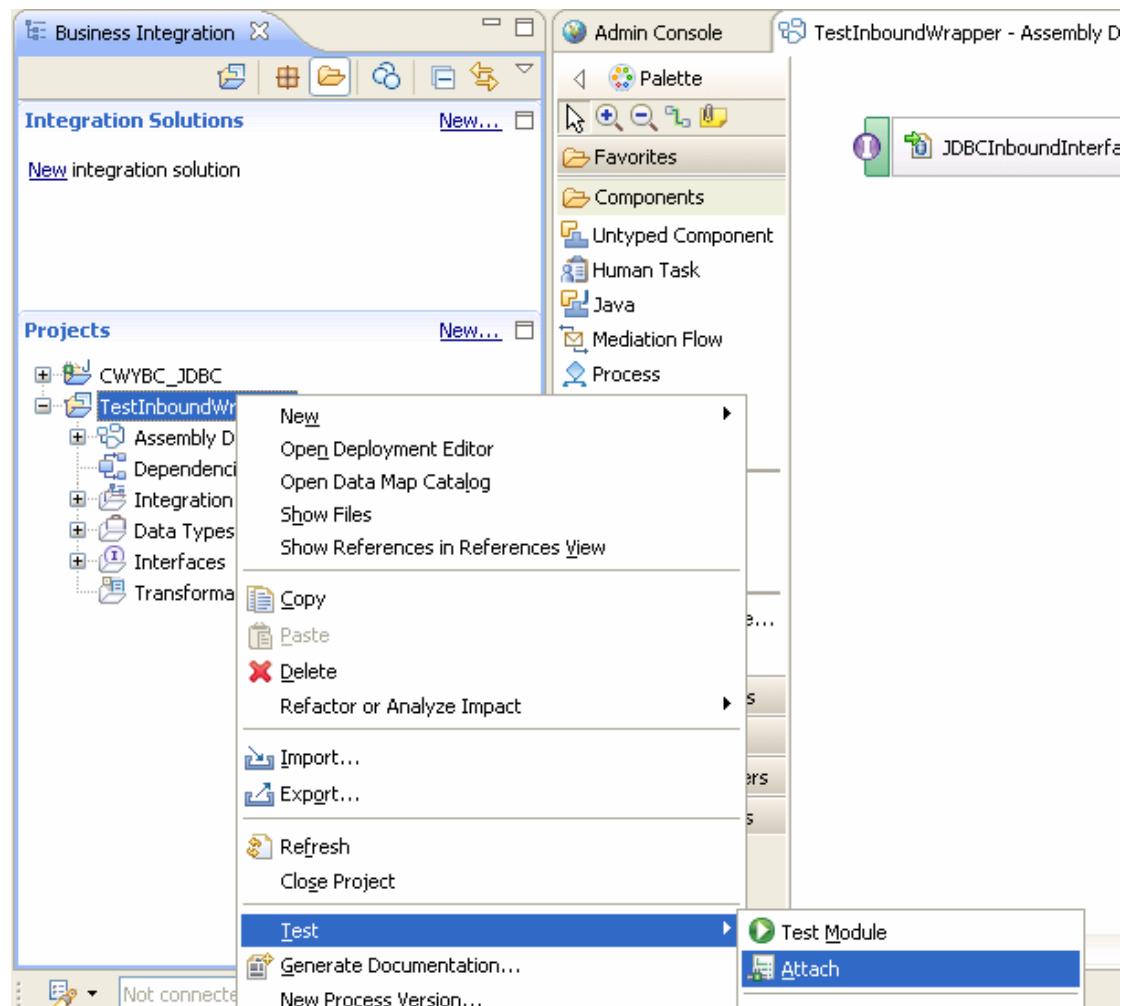
4. In the Add and Remove Projects window, select your project (TestWrapperInbound) and click **Add** to configure the project on the server. Click **Finish**.

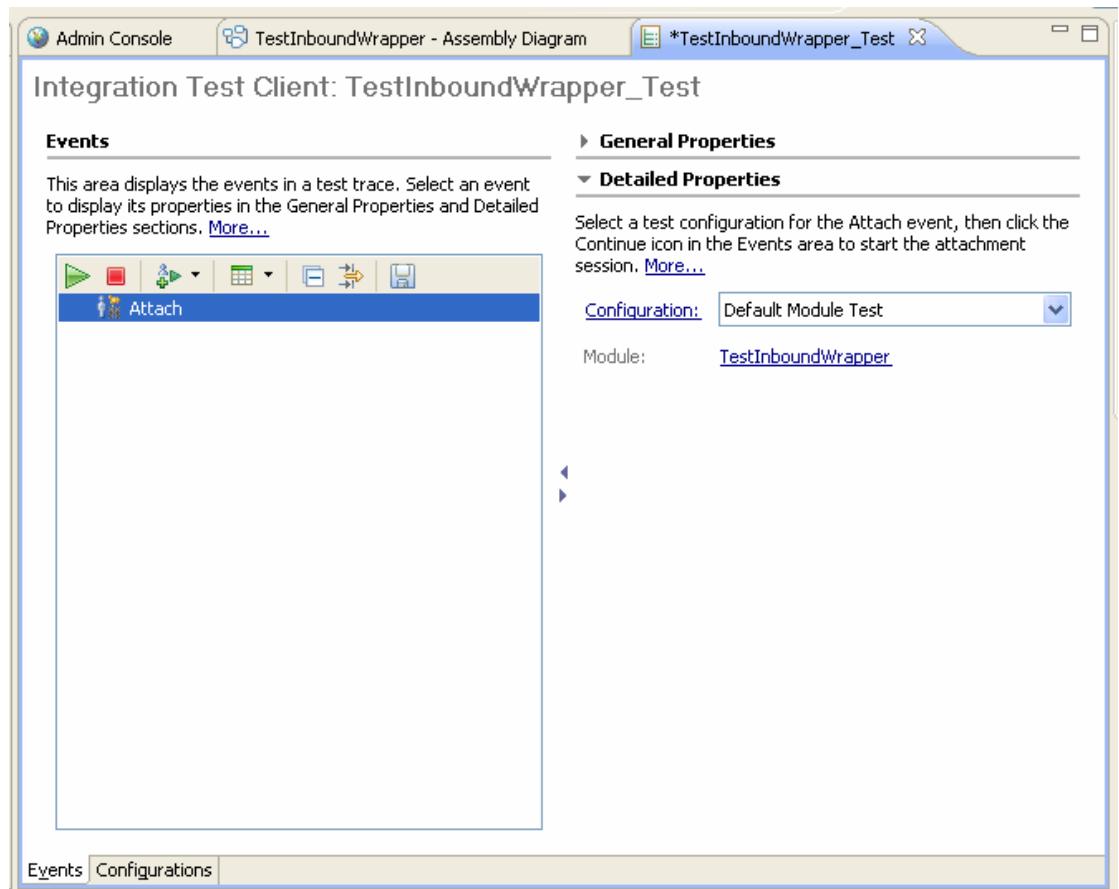


Test the assembled adapter application

Test the assembled adapter application using the WebSphere Integration Developer integration test client.

1. In the Business Integration view right-click on the TestInboundWrapper module, and select Test > Attach.





2. To execute the service, click .

3. Execute below SQL scripts to generate events:

```

INSERT INTO WBIA_JDBC_EVENTSTORE
  (EVENT_ID,OBJECT_KEY,OBJECT_NAME,OBJECT_FUNCTION,
  EVENT_PRIORITY,EVENT_TIME,EVENT_STATUS) VALUES
  (1,'A1;C1','wrapperBG','Create',1,
  SYSTIMESTAMP,0);
INSERT INTO WBIA_JDBC_EVENTSTORE
  (EVENT_ID,OBJECT_KEY,OBJECT_NAME,OBJECT_FUNCTION,
  EVENT_PRIORITY,EVENT_TIME,EVENT_STATUS) VALUES
  (2,'A2;C2','wrapperBG','Update',1,
  SYSTIMESTAMP,0);
INSERT INTO WBIA_JDBC_EVENTSTORE
  (EVENT_ID,OBJECT_KEY,OBJECT_NAME,OBJECT_FUNCTION,
  EVENT_PRIORITY,EVENT_TIME,EVENT_STATUS) VALUES
  (3,'A3;C3','wrapperBG','Delete',1,
  SYSTIMESTAMP,0);

```

Note: in a real environment, a trigger or another application, which can access the database, may insert the event record.

4. Check the output of the service:

Admin Console TestInboundWrapper - Assembly Diagram *TestInboundWrapper_Test

Integration Test Client: TestInboundWrapper_Test

Events

This area displays the events in a test trace. Select an event to display its properties in the General Properties and Detailed Properties sections. [More...](#)

Module: [TestInboundWrapper](#)
Source component: [JDBCInboundInterface](#)
Source reference: <export>
Target component: [Component1](#)
Target interface: [JDBCInboundInterface](#)
Target operation: [createwrapperBG](#)

Request parameters:

Value Editor XML Source

Name	Type	Value
wrapperBG	verb<string>	Create
verb	wrapper	A1
wrapper	string	C1
wrapcustaddaddr	string	SampleCus...
wrapcustomerkey	string	A1
custaddobj	SampleCus...	BEIJING
addrid	string	100000
custid	string	TIGER
city	string	JONE
zipcode	string	1
customerobj	SampleCus...	
pkey	string	
fname	string	
lname	string	
ccode	string	

Chapter 16. Troubleshooting

1. **Symptom:** Error while attempting to connect to Oracle database with Enterprise Metadata Discovery tool.

Resolution:

Verify whether the connection parameters have been entered correctly.

2. **Symptom:** There are no tables listed in the tree view the **Discovered objects** area.

Resolution:

Verify whether the tables are added to the **Supported database object types** property in the Query Properties window.

3. **Symptom:** A ClassNotFoundException exception is generated from the external service wizard or at runtime:

Cause: This is usually caused by configuration issues with the Oracle JDBC driver path.

Resolution:

Verify that the WebSphere variable such as ORACLE_JDBC_DRIVER_PATH is the path contains the JDBC DB driver.

Verify that the oracle driver jar is exactly the jar required.

4. **Symptom:** A primary key does not exist exception is generated from the external service wizard or at runtime.

Cause: The table does not have a primary key defined. Hence, the PrimaryKey ASI on the business object is not set to true.

Resolution:

Define a primary key in the table.

Set a PrimaryKey column on the business object.

5. **Symptom:** A record already exists exception is generated at runtime.

Cause: A record with the primary key already exists in the database.

Resolution:

Insert a record with a primary key that does not exist in the database.

6. **Symptom:** Test failed with following Exception message:
javax.resource.ResourceException: LoginException getting Subject, and with following Exception trace.

```
Caused by:  
javax.security.auth.login.LoginException:  
Incorrect authDataEntry and alias is: <AliasName>  
    at  
com.ibm.ws.security.auth.j2c.WSDefaultPrincipalMap  
ping.getMappedSubject(WSDefaultPrincipalMapping.ja  
va:505)  
    at  
com.ibm.ejs.j2c.PrivExAction.run(PrivExAction.java  
:145)  
... 53 more
```

Cause: This is usually caused by configuration issues such as incorrect J2C Authentication Data Entry value entered in the Service Generation and Deployment Configuration window.

Resolution:

Verify that the authentication alias is a full authentication alias which could be exactly found in the JAAS - J2C authentication data view under administrative console.

Chapter 17. Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing

IBM Corporation

North Castle Drive

Armonk, NY 10504-1785

U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in

writing, to:

IBM World Trade Asia Corporation Licensing

2-31 Roppongi 3-chome, Minato-ku

Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication.

IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites.

The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation Department
2Z4A/SOM1 294 Route 100
Somers, NY 10589-0100 U.S.A.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include

the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Programming interface information

Programming interface information, if provided, is intended to help you create application software using this program.

General-use programming interfaces allow you to write application software that obtain the services of this program's tools.

WebSphere® software

However, this information may also contain diagnosis, modification, and tuning information. Diagnosis, modification and tuning information is provided to help you debug your application software.

Warning:

Do not use this diagnosis, modification, and tuning information as a programming interface because it is subject to change.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A complete and current list of IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.

This product includes software developed by the Eclipse Project (<http://www.eclipse.org>).