IBM WebSphere Adapter for JDBC 7.5.0.0

Quick Start Tutorials

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.

Table of contents

Chapter 1. Introduction

WebSphere Adapter for JDBC 7.5.0.0 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 IBM Integration Designer.
- 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 IBM 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:

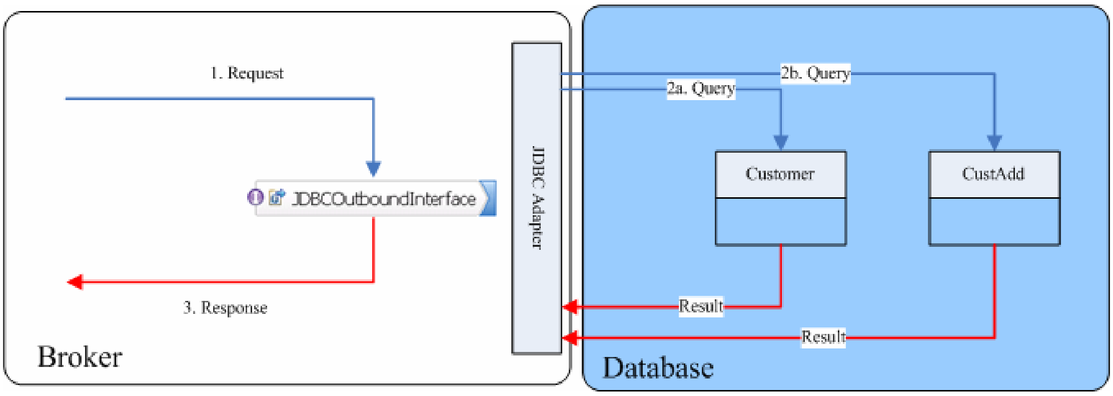
- IBM Integration Designer version 7.5.0.0
- IBM Process Server version 7.5.0.0
- WebSphere Adapter for JDBC version 7.5.0.0
- JDBC Driver for Oracle
- JDBC Driver for DB2
- JDBC Driver for SQLServer

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 7.5.0.0. 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.

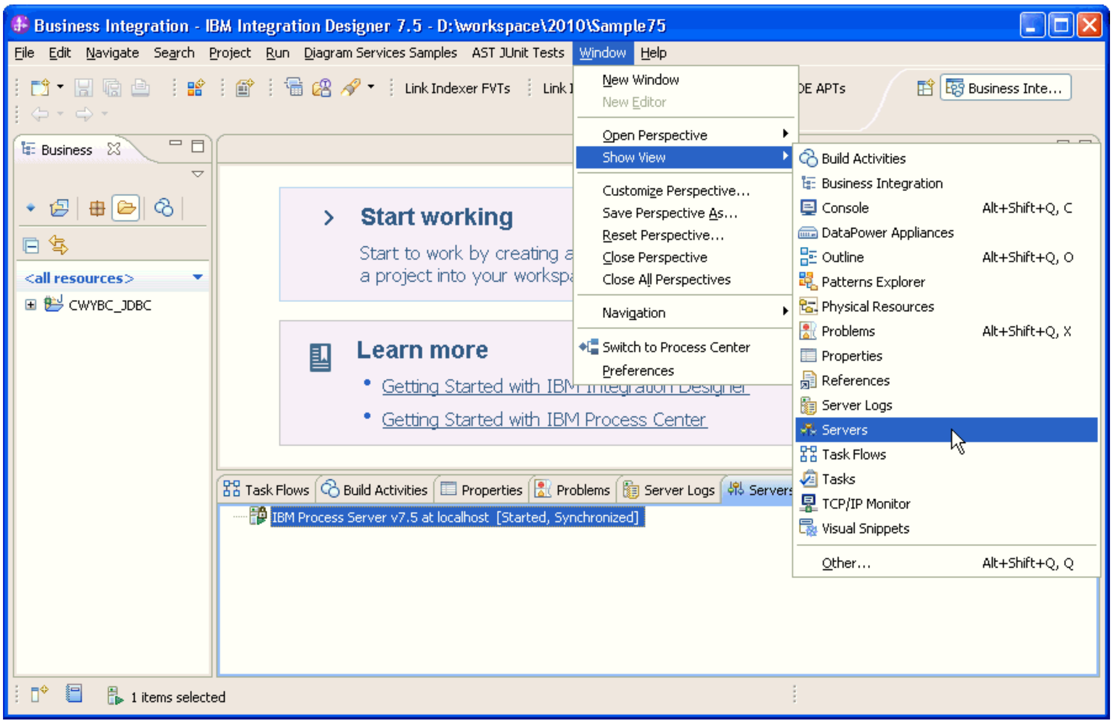
```
CREATE TABLE CUSTOMER  (
        PKEY VARCHAR2(10) NOT NULL PRIMARY KEY,
        FNAME VARCHAR2(20) ,
        LNAME VARCHAR2(20) ,
        CCODE VARCHAR2(10) ) ;

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

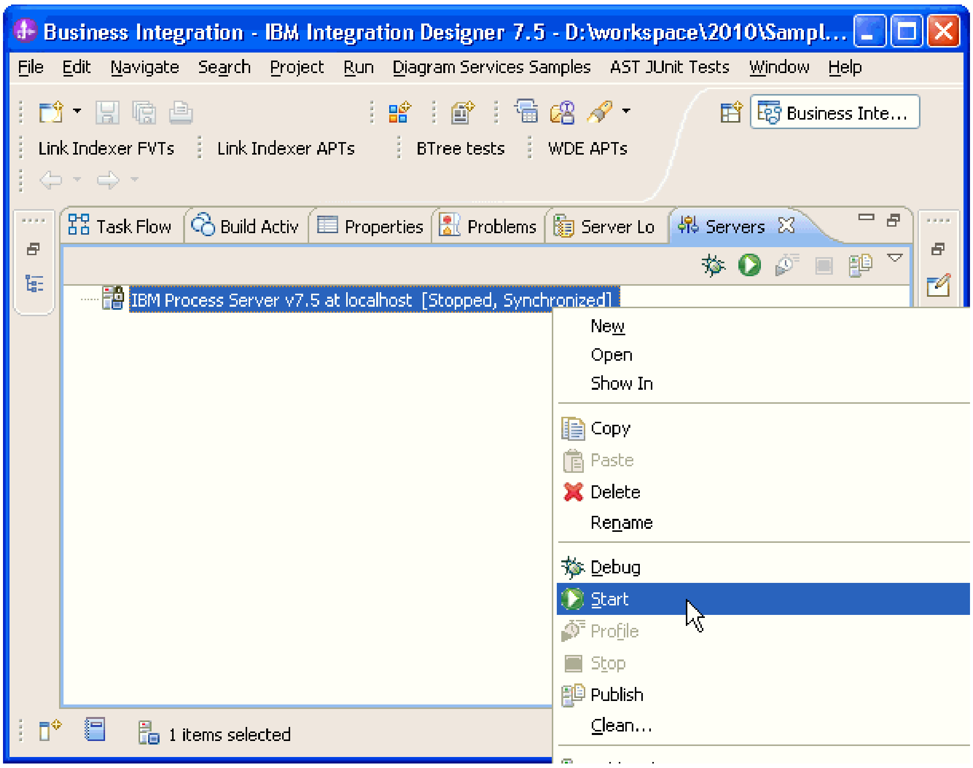
The authentication alias needs to be set because the data source created in the next section uses the username and password set in the authentication alias to connect to the database.

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

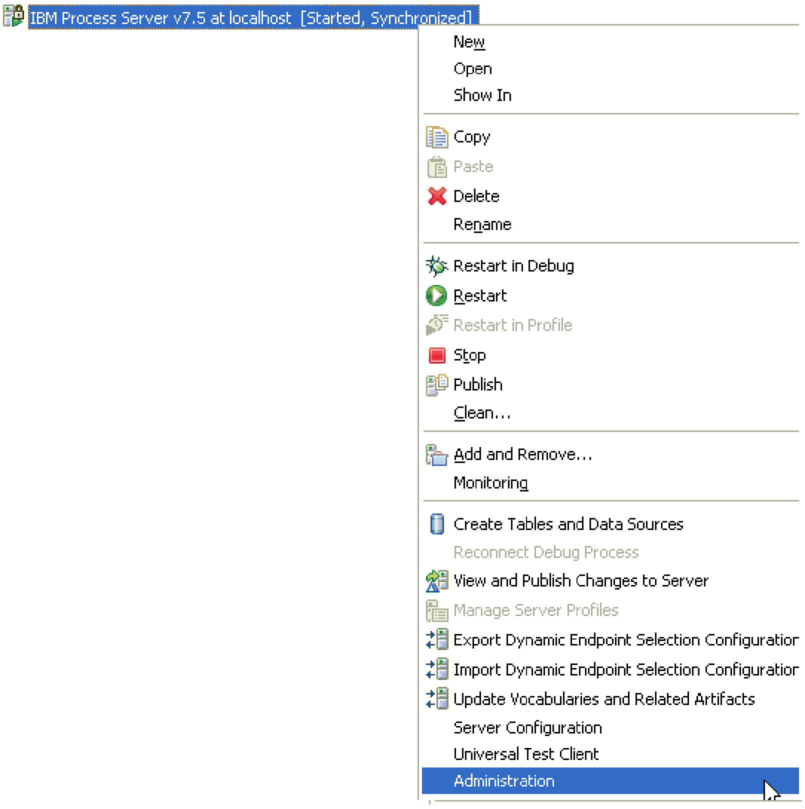
1. In IBM Integration Designer, switch to the **Servers** view by selecting **Windows > Show View > Servers**.



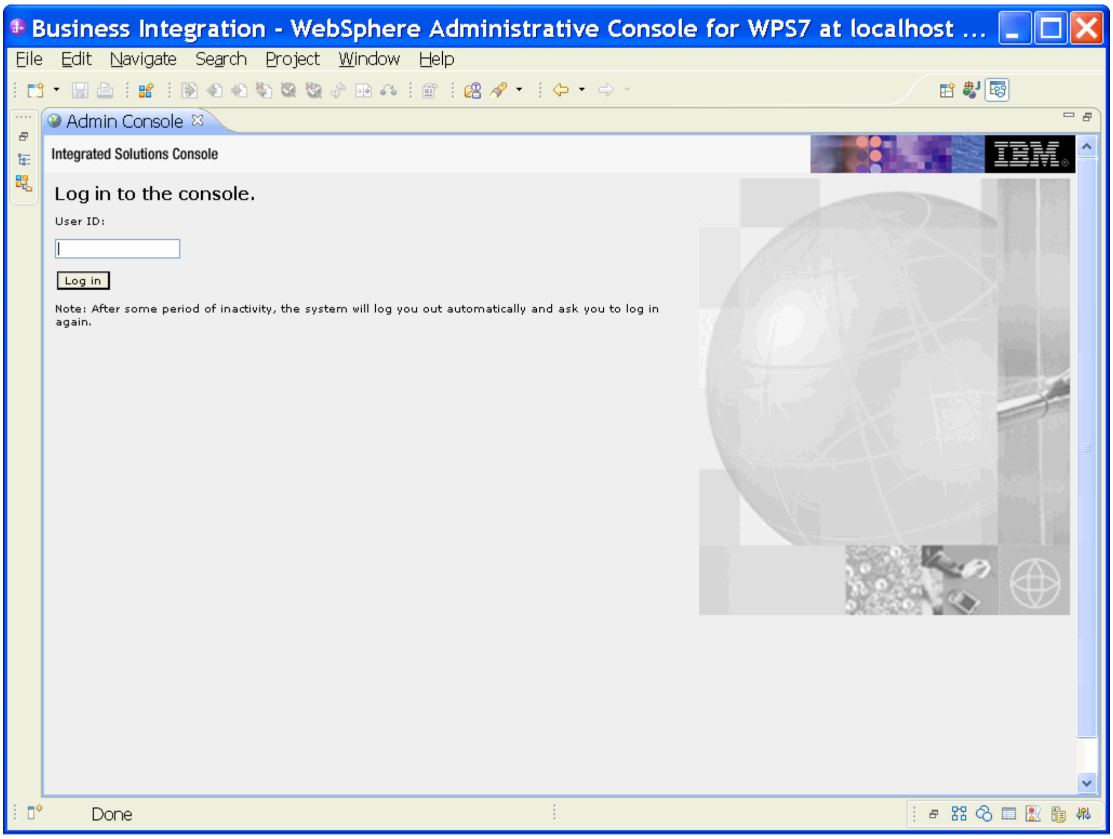
2. In the **Servers** view, right-click the server that you want to start and select **Start**.


Business Integration - IBM Integration Designer 7.5 - D:\workspace\2010\Sampl...
File Edit Navigate Search Project Run Diagram Services Samples AST JUnit Tests Window Help
Link Indexer FVTs Link Indexer APTs BTree tests WDE APTs
Business Inte...
Task Flow Build Activ Properties Problems Server Lo Servers
IBM Process Server v7.5 at localhost [Stopped, Synchronized]
New
Open
Show In
Copy
Paste
Delete
Rename
Debug
Start
Profile
Stop
Publish
Clean...
1 items selected

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


IBM Process Server v7.5 at localhost [Started, Synchronized]
New
Open
Show In
Copy
Paste
Delete
Rename
Restart in Debug
Restart
Restart in Profile
Stop
Publish
Clean...
Add and Remove...
Monitoring
Create Tables and Data Sources
Reconnect Debug Process
View and Publish Changes to Server
Manage Server Profiles
Export Dynamic Endpoint Selection Configuratior
Import Dynamic Endpoint Selection Configuratior
Update Vocabularies and Related Artifacts
Server Configuration
Universal Test Client
Administration

4. Log on to the administrative console.

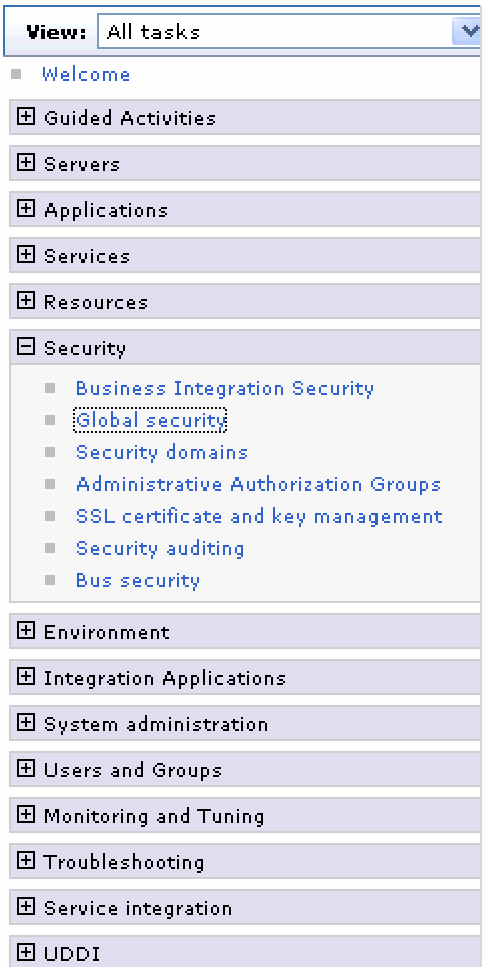

Business Integration - WebSphere Administrative Console for WPS7 at localhost ...
File Edit Navigate Search Project Window Help
Admin Console
Integrated Solutions Console
IBM
Log in
Done

Log in to the console.

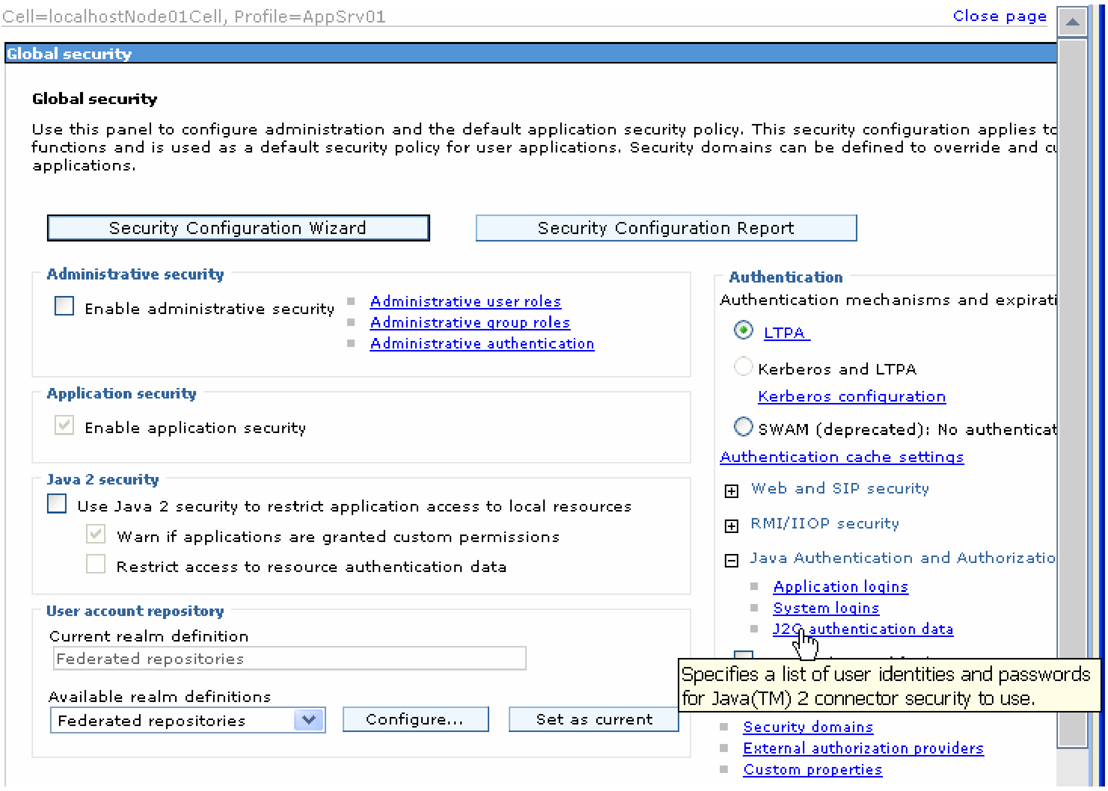
User ID:

Note: After some period of inactivity, the system will log you out automatically and ask you to log in again.

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



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



Cell=localhostNode01Cell, Profile=AppSrv01

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 cu applications.

Administrative security

□ Enable administrative security

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 expirati

● LTPA

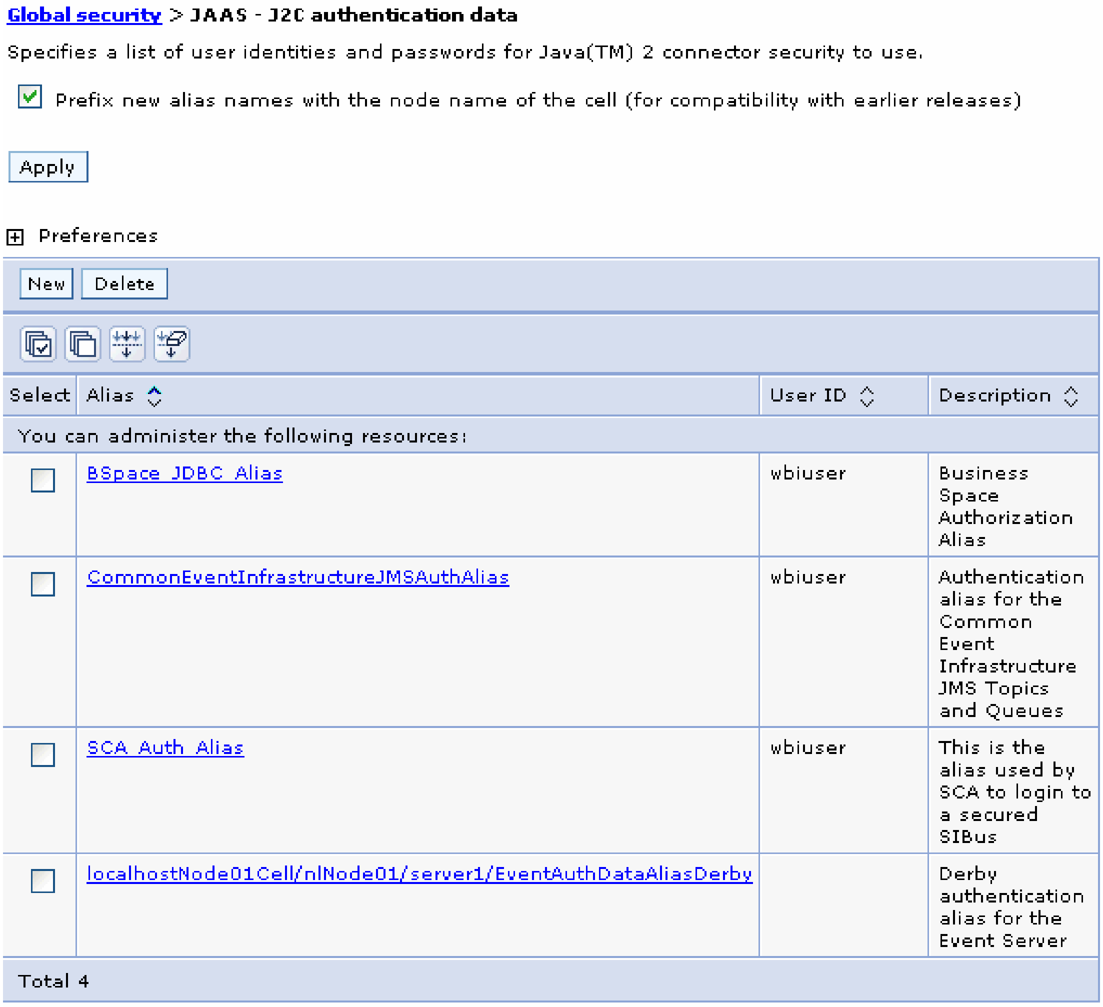
○ Kerberos and LTPA

○ SWAM (deprecated): No authenticat

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

Close page
Security Configuration Wizard
Security Configuration Report
Administrative user roles
Administrative group roles
Administrative authentication
Configure...
Set as current
Kerberos configuration
Authentication cache settings
⊞ Web and SIP security
⊞ RMI/IIOP security
⊟ Java Authentication and Authorizatio
Application logins
System logins
J2C authentication data
Security domains
External authorization providers
Custom properties

A list of existing aliases is displayed.



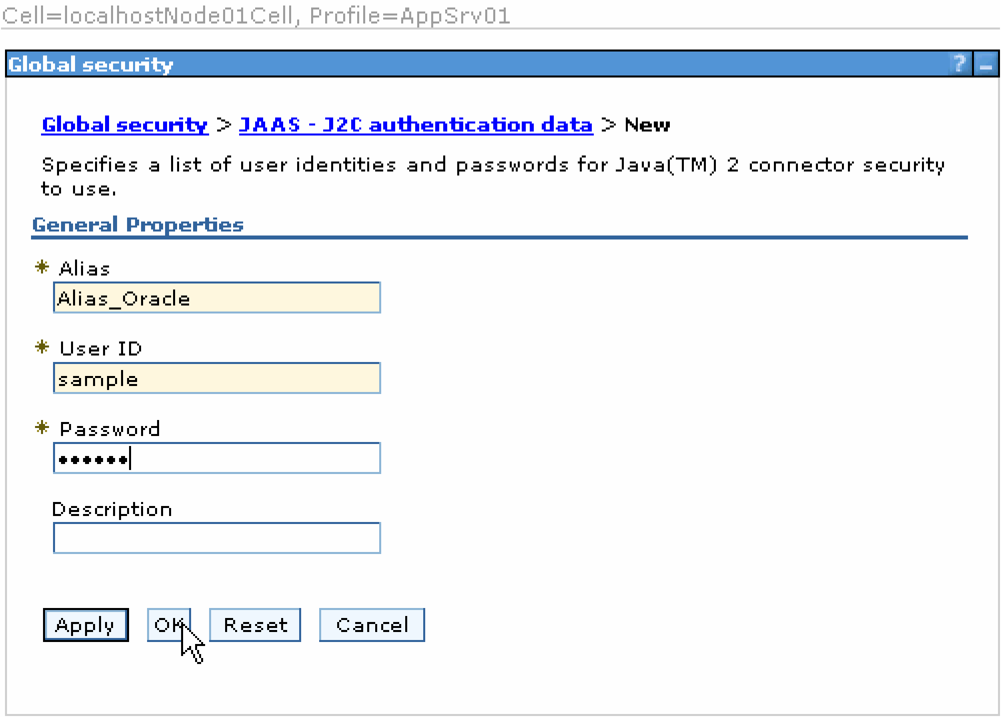
Global security > JAAS - J2C authentication data

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)

Select	Alias	User ID	Description
You can administer the following resources:			
□	BSpace_JDBC_Alias	wbiuser	Business Space Authorization Alias
□	CommonEventInfrastructureJMSAuthAlias	wbiuser	Authentication alias for the Common Event Infrastructure JMS Topics and Queues
□	SCA_Auth_Alias	wbiuser	This is the alias used by SCA to login to a secured SIBus
□	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**.



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

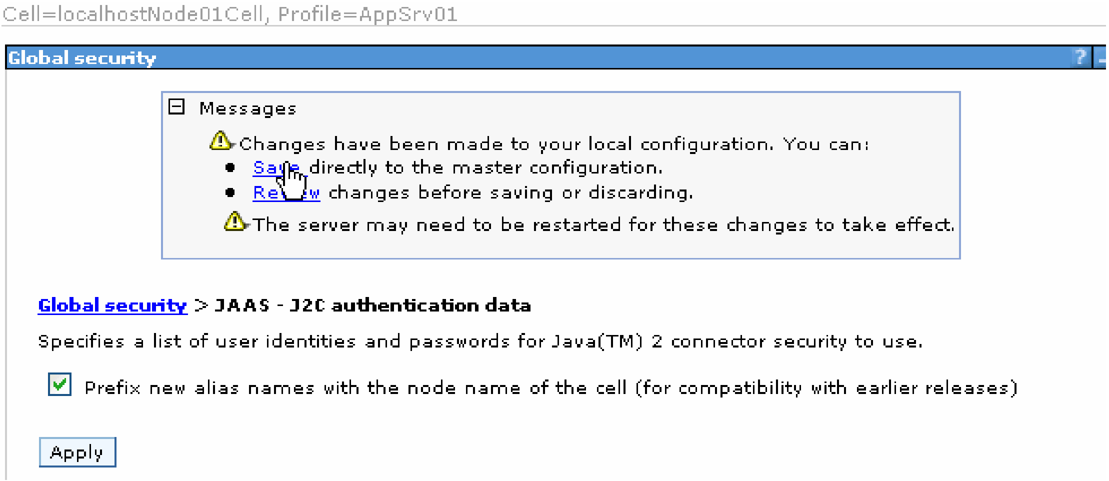
* Alias
Alias_Oracle

* User ID
sample

* Password
••••••

Description

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.

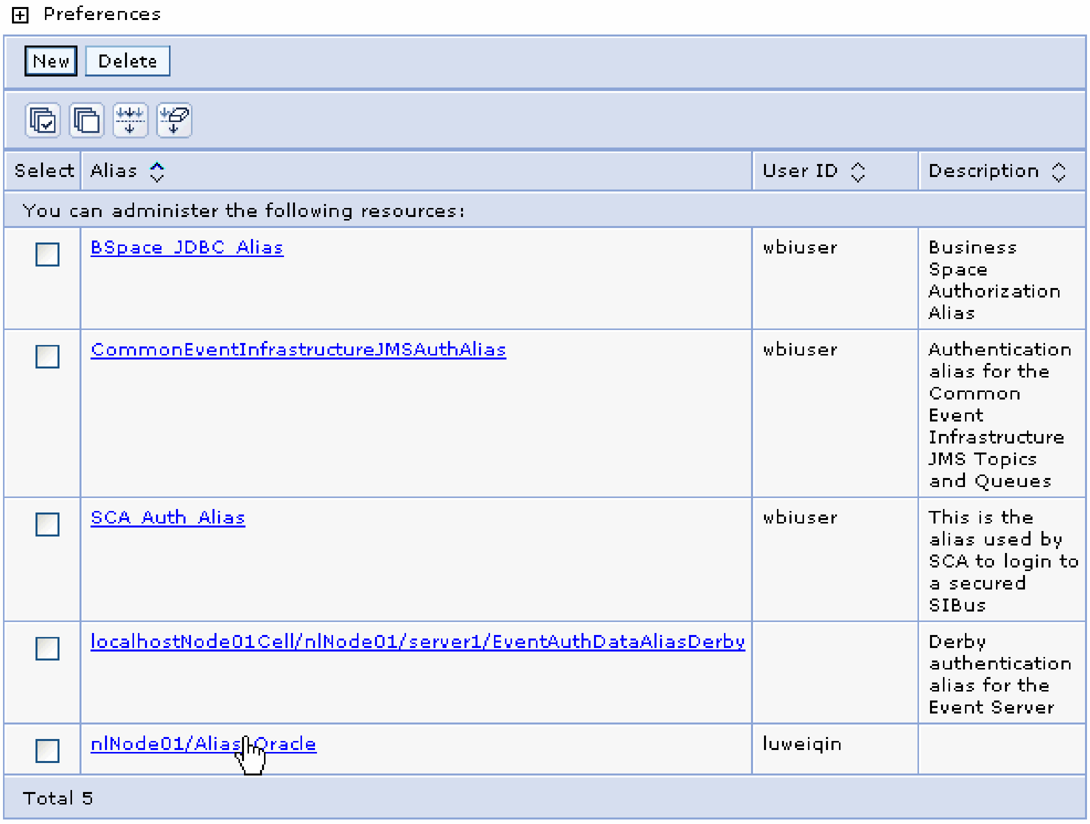
⚠ The server may need to be restarted for these changes to take effect.

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)

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



Select	Alias	User ID	Description
You can administer the following resources:			
□	BSpace_JDBC_Alias	wbiuser	Business Space Authorization Alias
□	CommonEventInfrastructureJMSAuthAlias	wbiuser	Authentication alias for the Common Event Infrastructure JMS Topics and Queues
□	SCA_Auth_Alias	wbiuser	This is the alias used by SCA to login to a secured SIBus
□	localhostNode01Cell/nlNode01/server1/EventAuthDataAliasDerby		Derby authentication alias for the Event Server
□	nlNode01/Alias_Oracle	luweiqin	
Total 5			

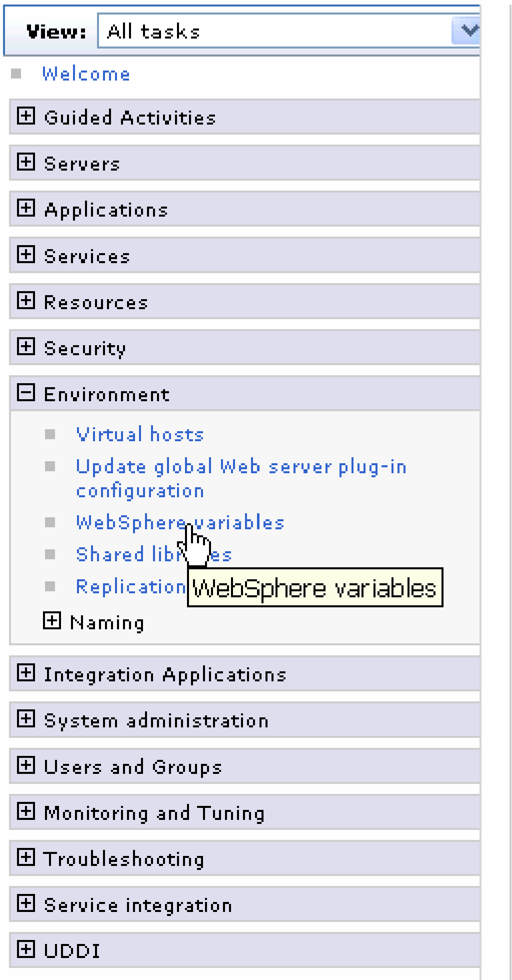
Create the data source

Create a data source in IBM 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.

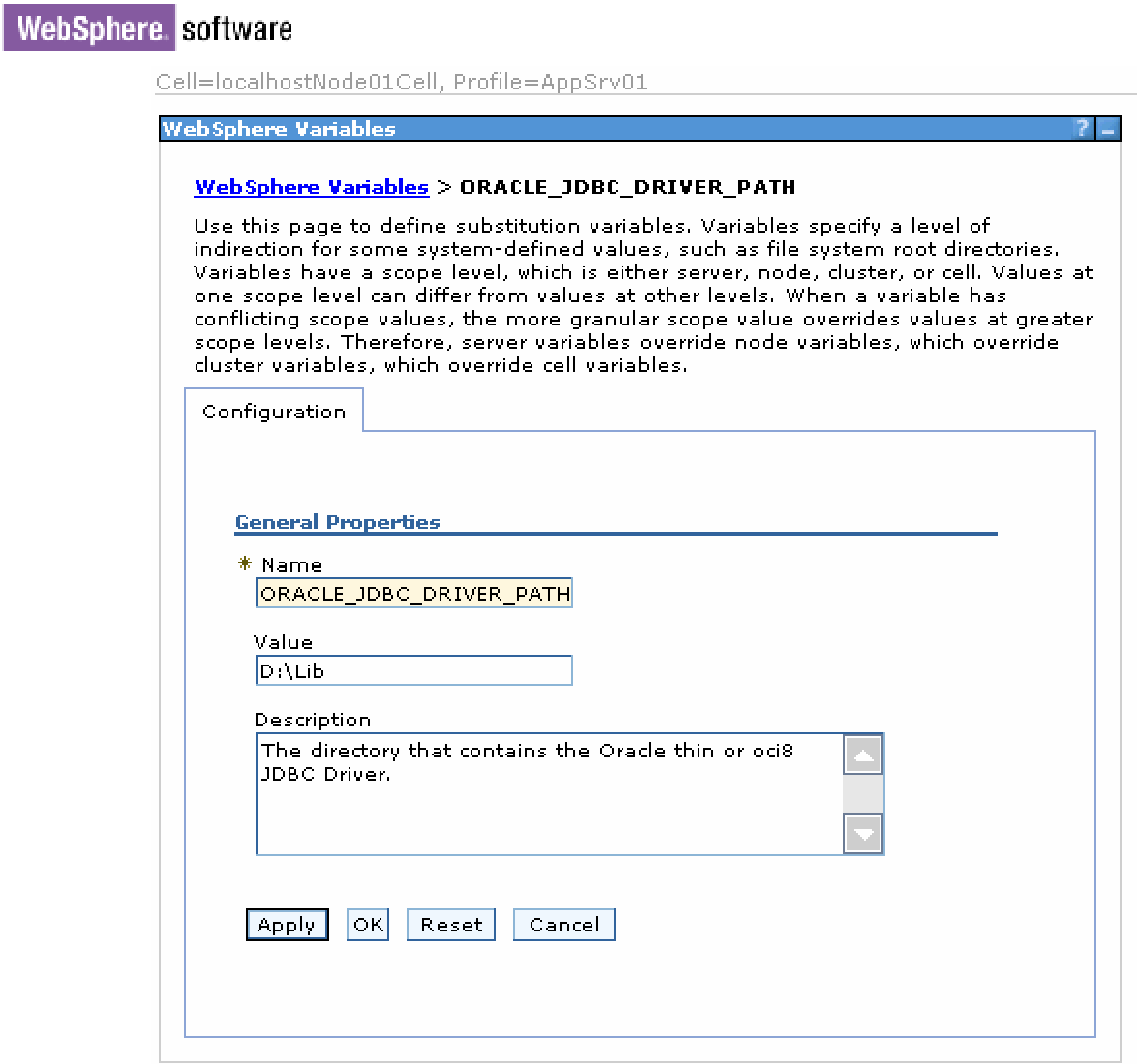
Note: This tutorial will use Oracle as the database and the Oracle thin driver, ojdbc6.jar.

Here are the steps to create the data source in the IBM 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**.



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

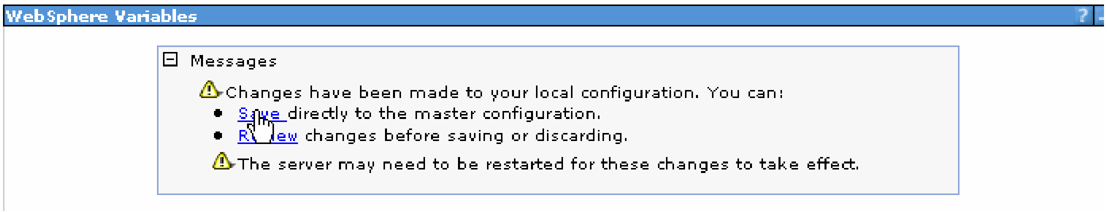
General Properties

* Name: ORACLE_JDBC_DRIVER_PATH

Value: D:\Lib

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

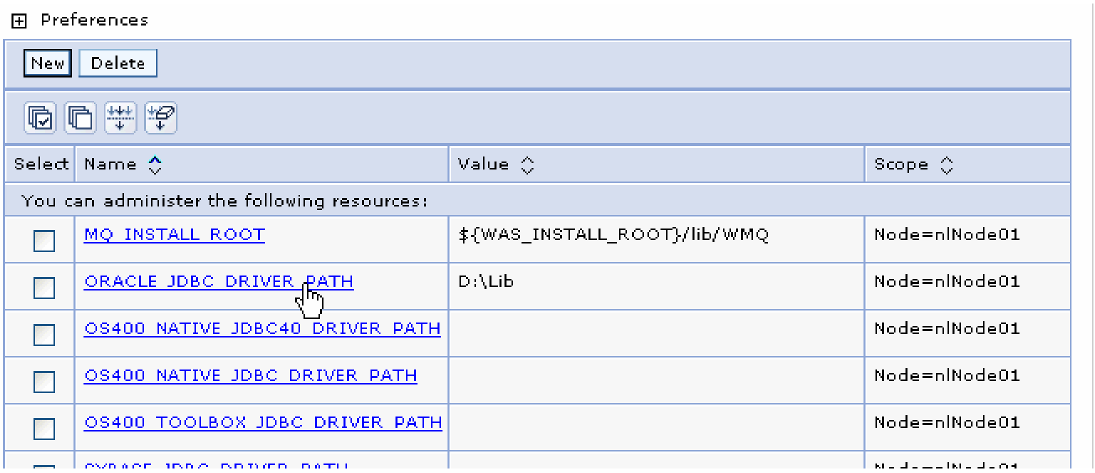
3. Click **Save** to save the changes.



⚠ Changes have been made to your local configuration. You can:
- Save directly to the master configuration.
- Review changes before saving or discarding.

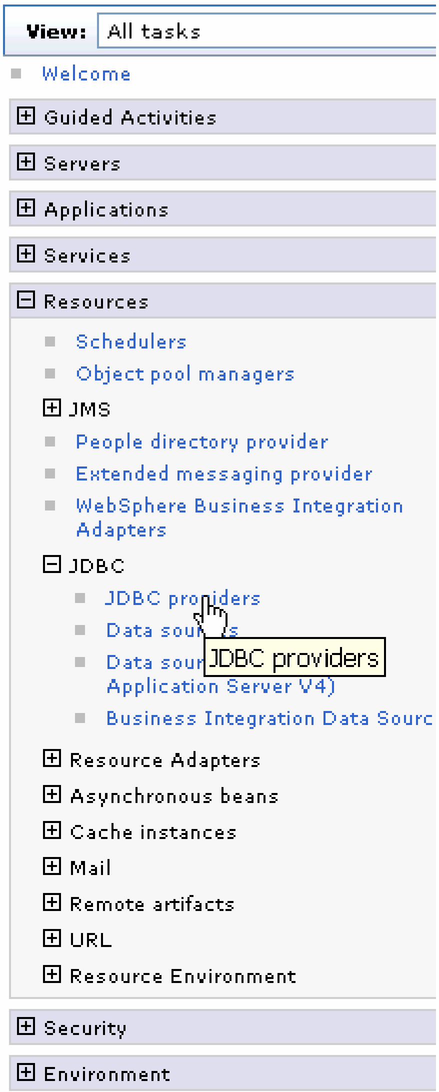
⚠ The server may need to be restarted for these changes to take effect.

The variable has been added and appears in the list.

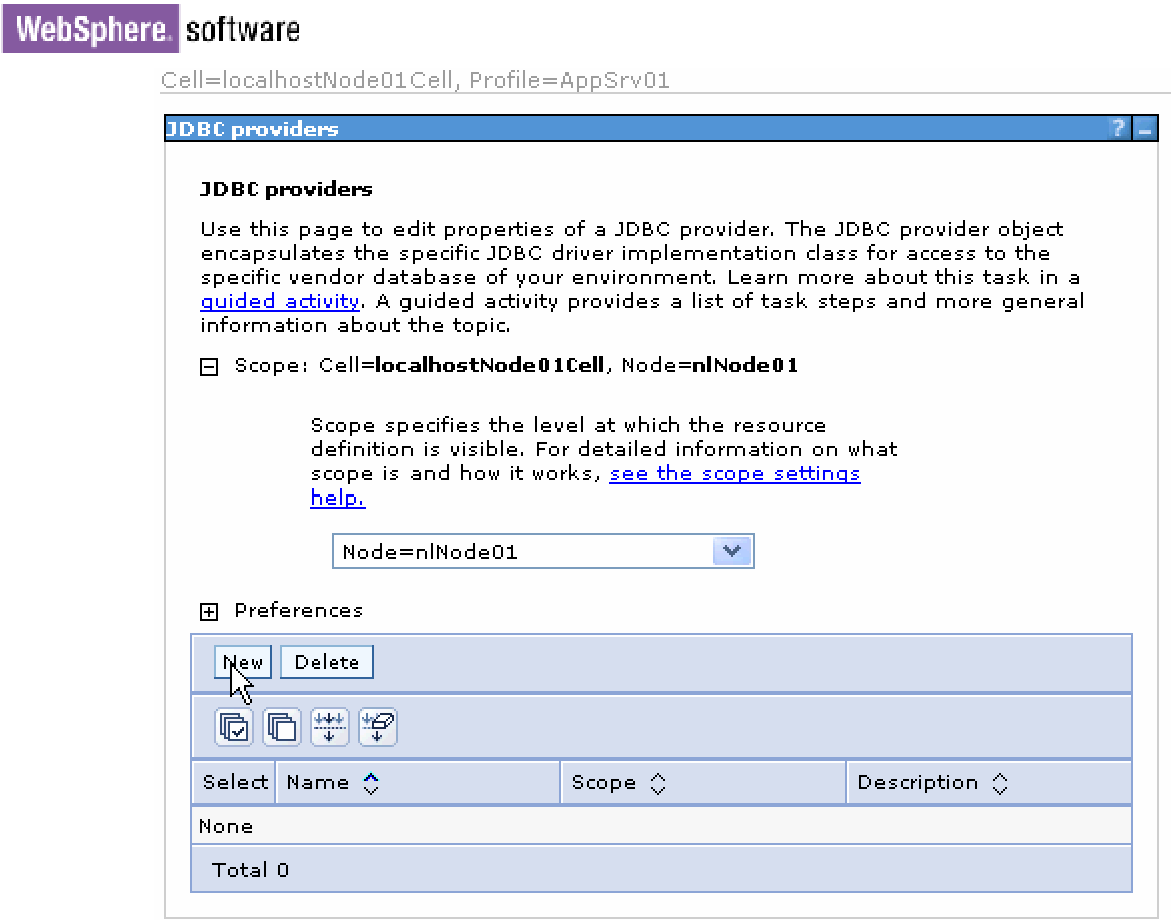


Select	Name	Value	Scope
You can administer the following resources:			
□	MQ_INSTALL_ROOT	${WAS_INSTALL_ROOT}/lib/WMQ	Node=nlNode01
□	ORACLE_JDBC_DRIVER_PATH	D:\Lib	Node=nlNode01
□	OS400_NATIVE_JDBC40_DRIVER_PATH		Node=nlNode01
□	OS400_NATIVE_JDBC_DRIVER_PATH		Node=nlNode01
□	OS400_TOOLBOX_JDBC_DRIVER_PATH		Node=nlNode01
□	SYBASE_JDBC_DRIVER_PATH		Node=nlNode01

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



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



WebSphere. software

Cell=localhostNode01Cell, Profile=AppSrv01

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=**nlNode01**

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=nlNode01

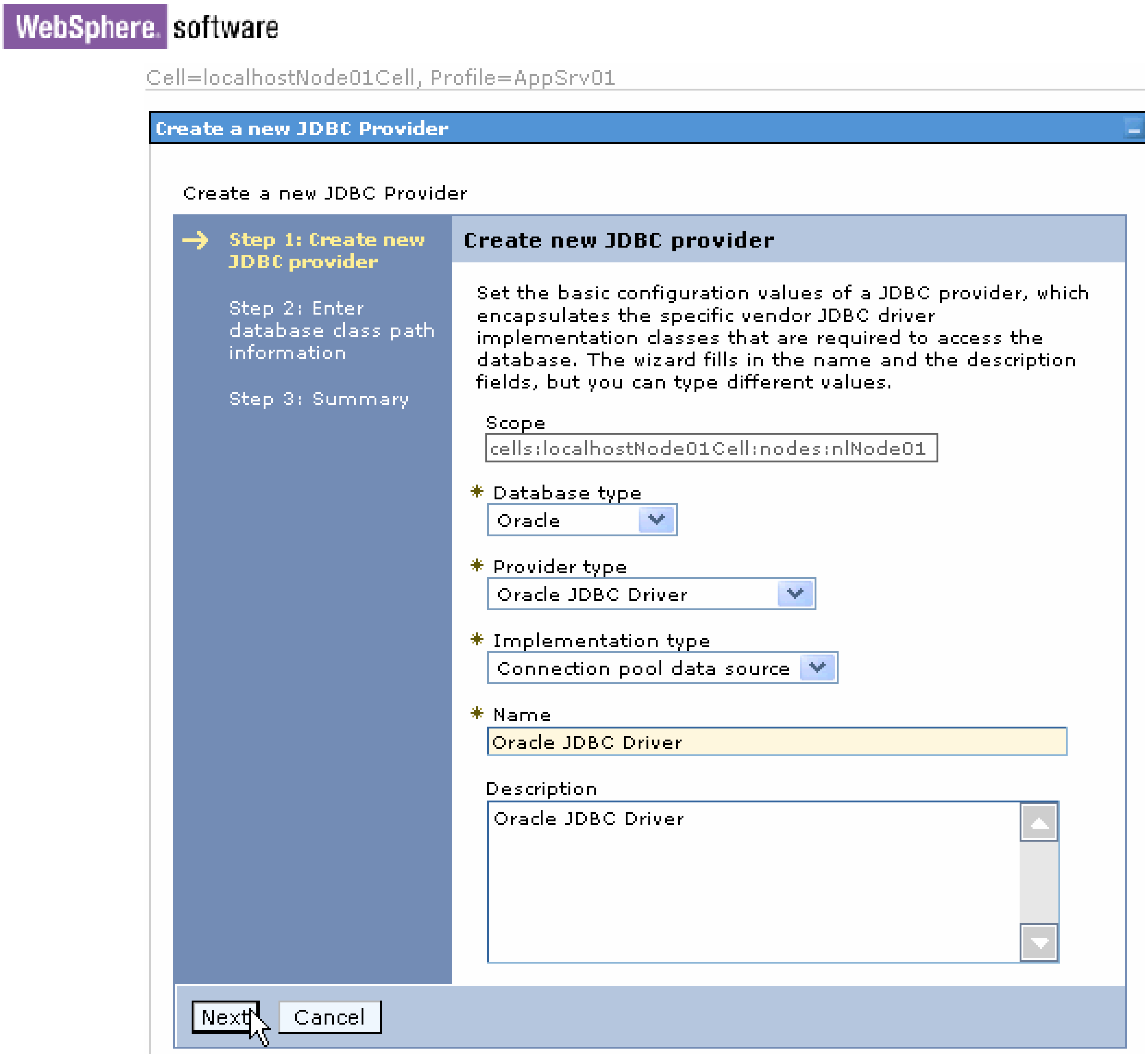
⊞ Preferences

New Delete

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**.



Cell=localhostNode01Cell, Profile=AppSrv01

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:nlNode01

* Database type
Oracle

* Provider type
Oracle JDBC Driver

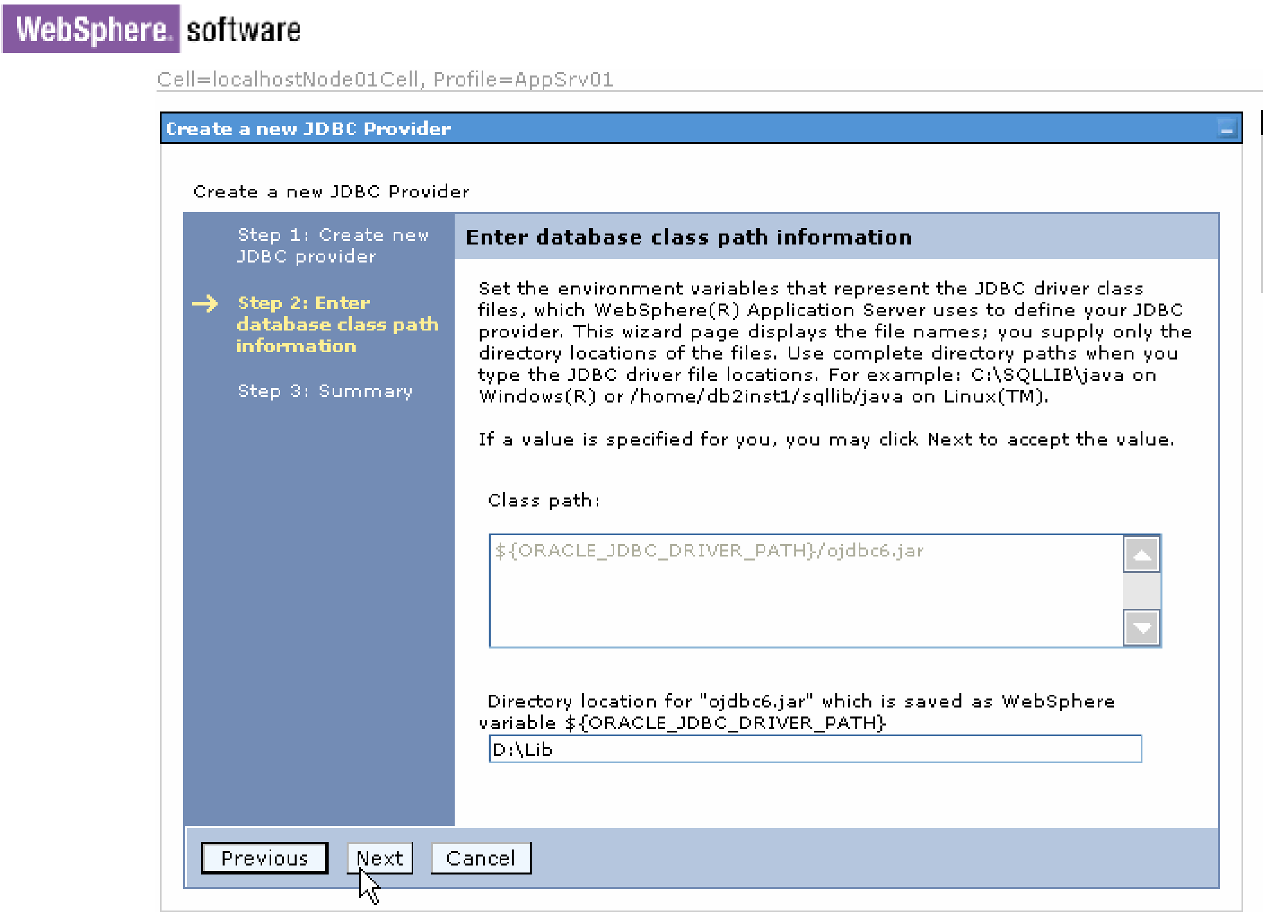
* Implementation type
Connection pool data source

* Name
Oracle JDBC Driver

Description
Oracle JDBC Driver

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**.



Cell=localhostNode01Cell, Profile=AppSrv01

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: 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.

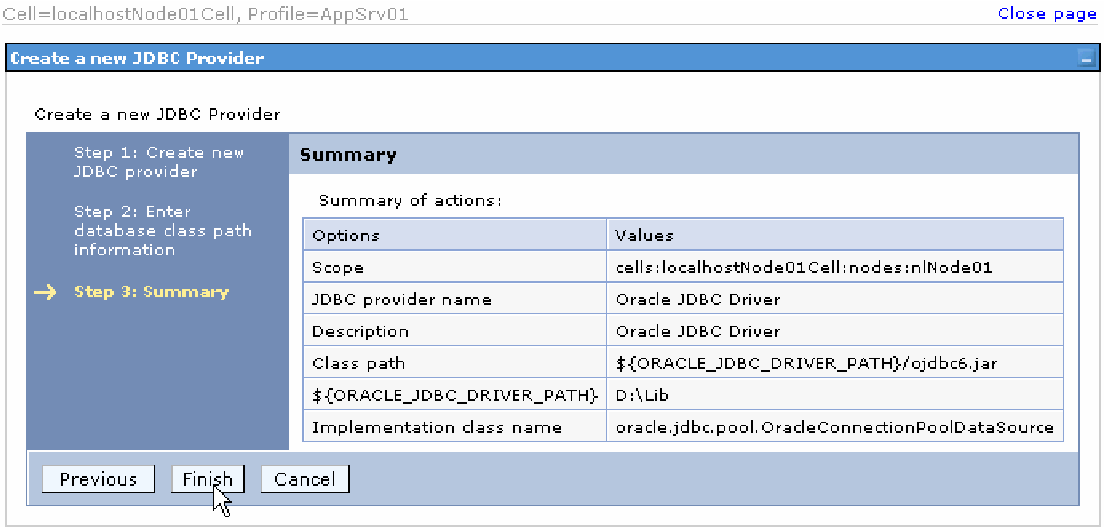
Class path:

${ORACLE_JDBC_DRIVER_PATH}/ojdbc6.jar

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

D:\Lib

9. Click **Finish**.



Cell=localhostNode01Cell, Profile=AppSrv01 Close page

Create a new JDBC Provider

Create a new JDBC Provider

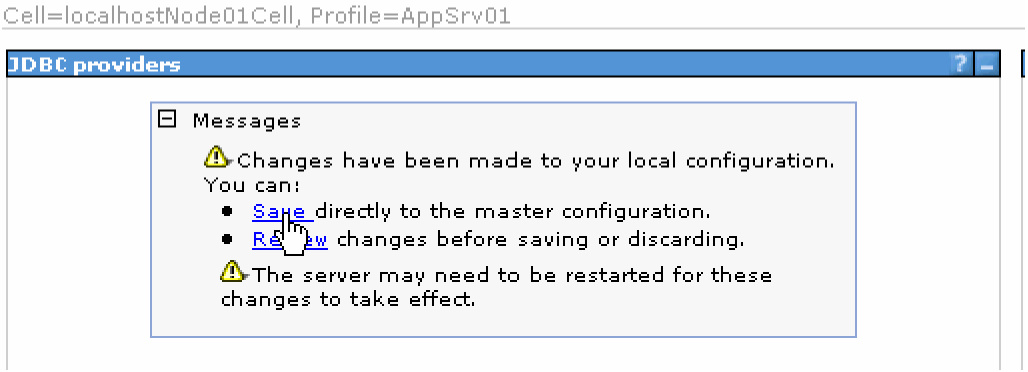
Step 1: Create new JDBC provider
Step 2: Enter database class path information
→ Step 3: Summary

Summary

Summary of actions:

Options	Values
Scope	cells:localhostNode01Cell:nodes:nlNode01
JDBC provider name	Oracle JDBC Driver
Description	Oracle JDBC Driver
Class path	${ORACLE_JDBC_DRIVER_PATH}/ojdbc6.jar
${ORACLE_JDBC_DRIVER_PATH}	D:\Lib
Implementation class name	oracle.jdbc.pool.OracleConnectionPoolDataSource

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



Cell=localhostNode01Cell, Profile=AppSrv01

JDBC providers

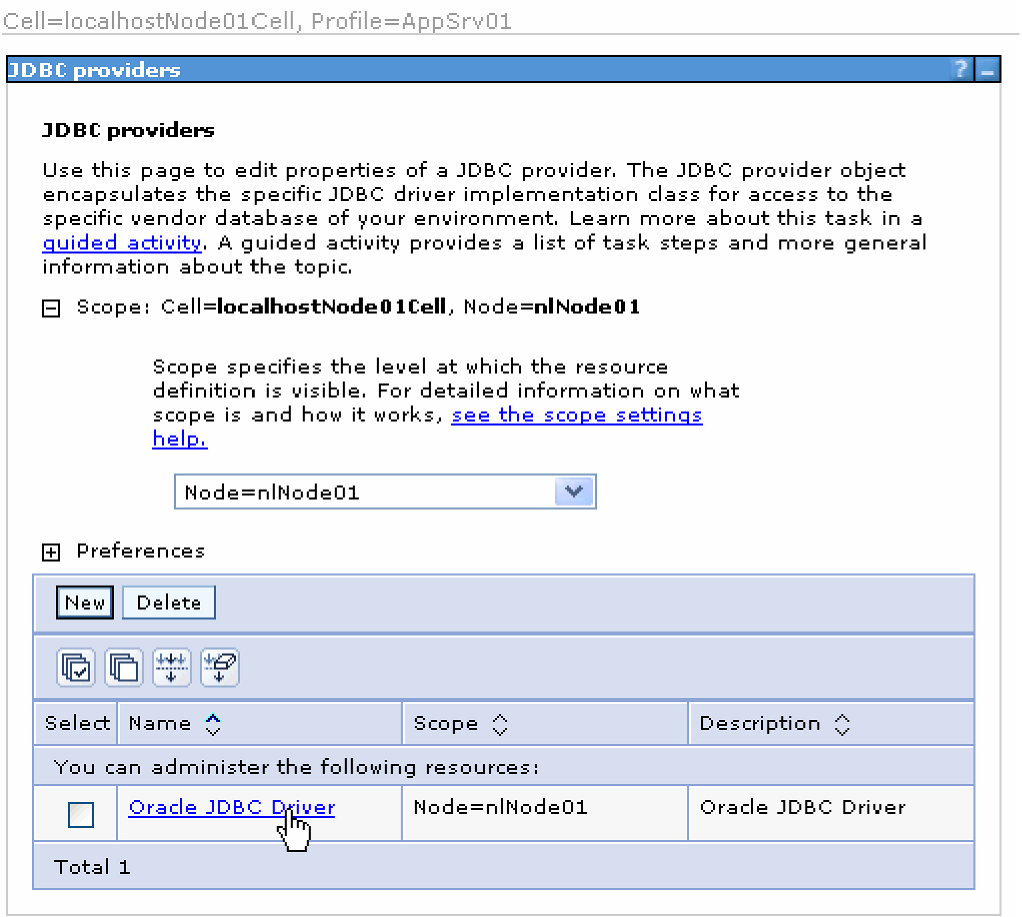
⊟ Messages

⚠ 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

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=**nlNode01**

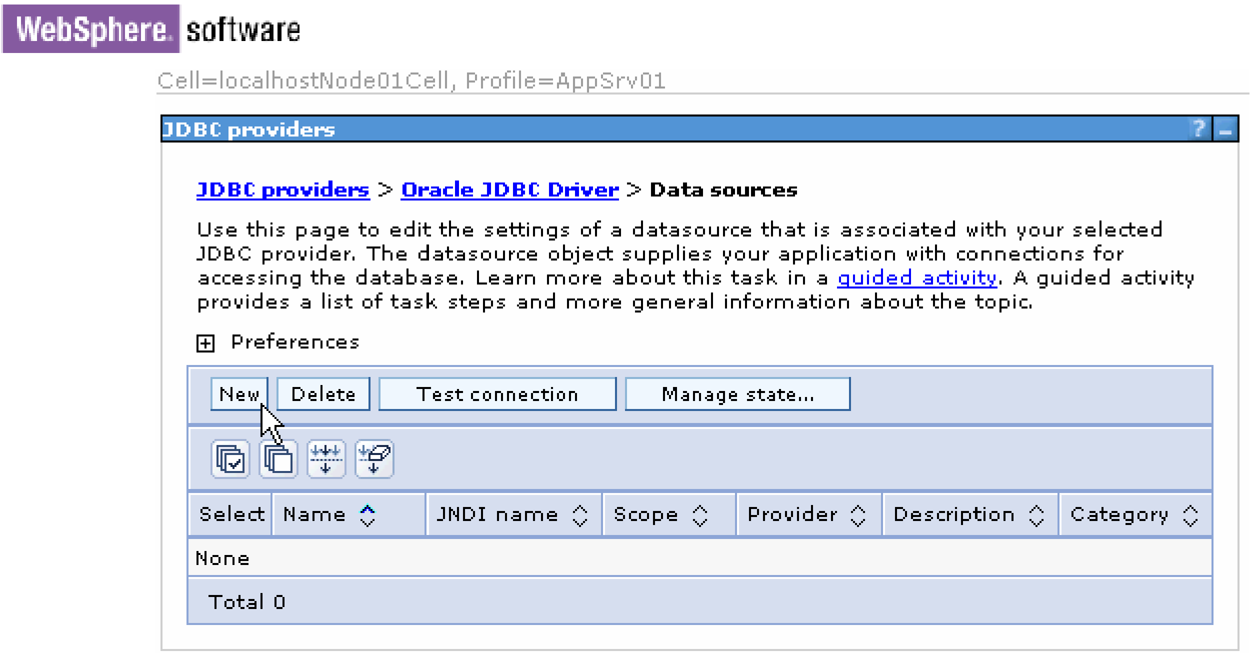
Scope specifies the level at which the resource definition is visible. For detailed information on what scope is and how it works, see the scope settings help.

Node=nlNode01

⊞ Preferences

Select	Name	Scope	Description
You can administer the following resources:			
□	Oracle JDBC Driver	Node=nlNode01	Oracle JDBC Driver
Total 1			

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

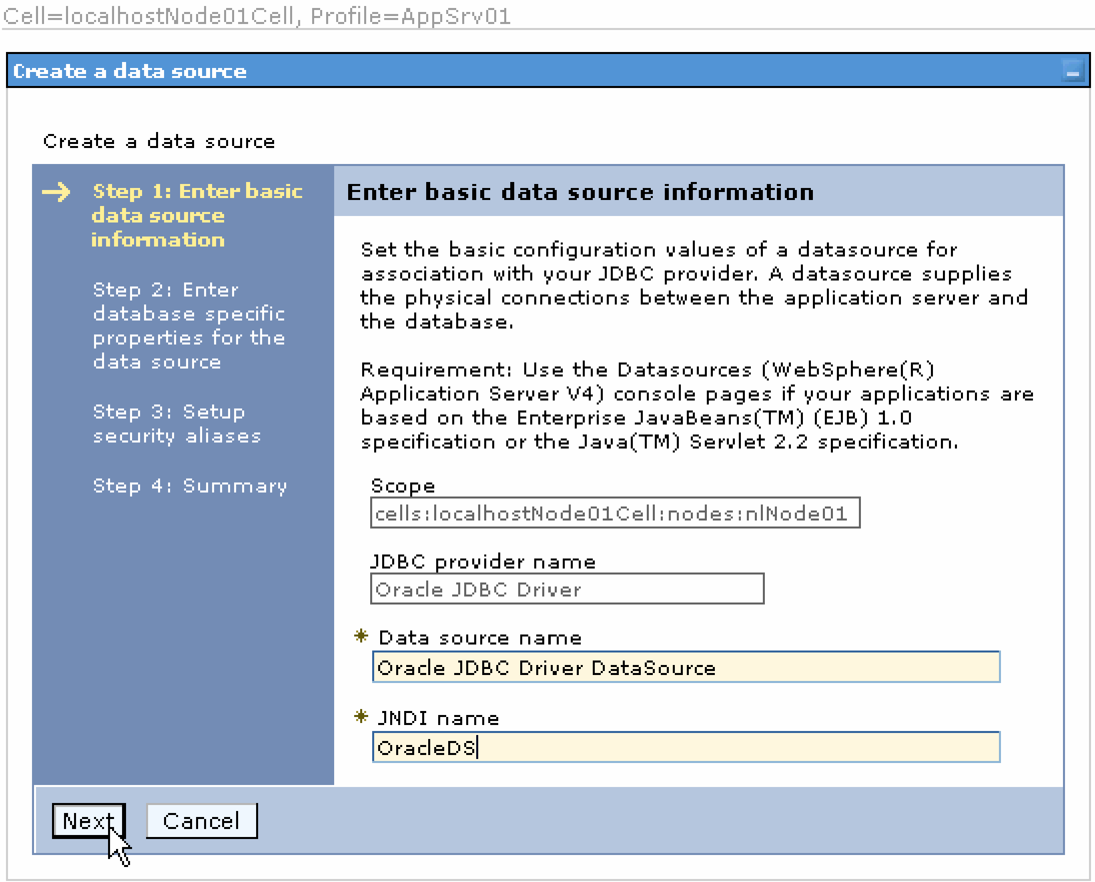


JDBC providers > Oracle JDBC Driver > Data sources

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.

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
Next
Cancel

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.

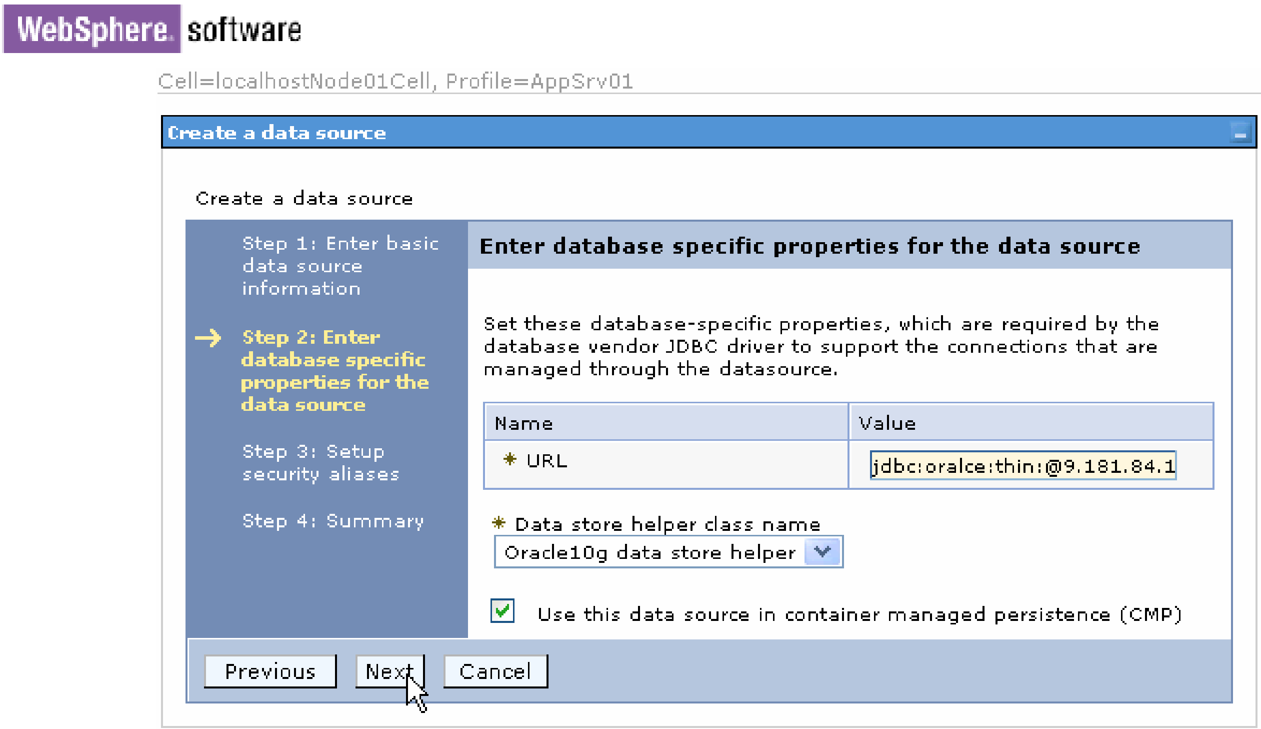
Scope: cells:localhostNode01Cell:nodes:nlNode01

JDBC provider name: Oracle JDBC Driver

* Data source name: Oracle JDBC Driver DataSource

* JNDI name: OracleDS

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

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

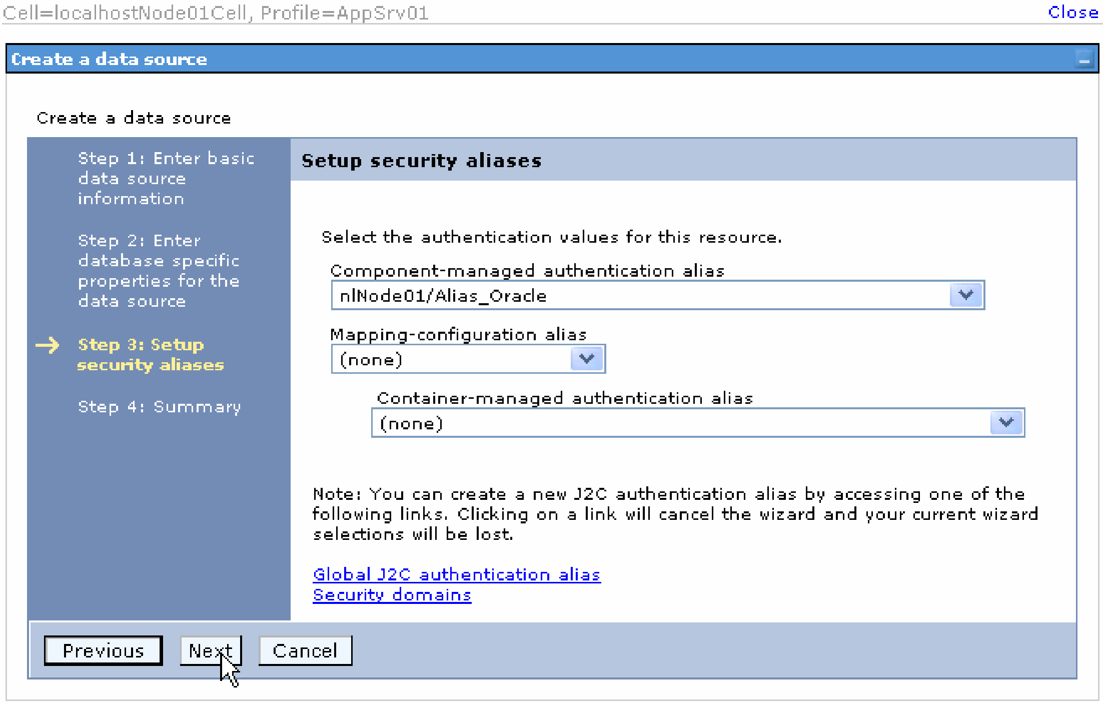
Set these database-specific properties, which are required by the database vendor JDBC driver to support the connections that are managed through the datasource.

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

* Data store helper class name
Oracle10g data store helper

☑ Use this data source in container managed persistence (CMP)

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

Setup security aliases

Select the authentication values for this resource.

Component-managed authentication alias
nlNode01/Alias_Oracle

Mapping-configuration alias
(none)

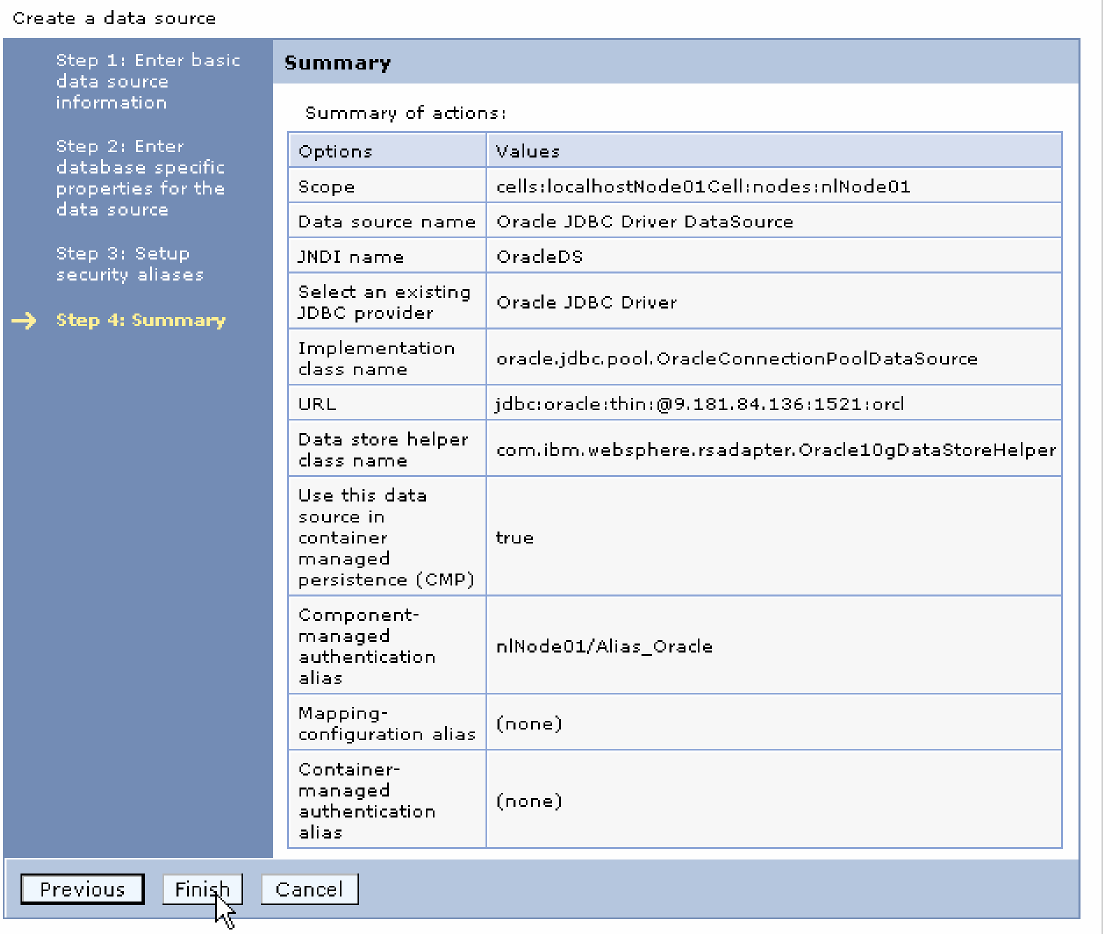
Container-managed authentication alias
(none)

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

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

15. Click **Finish**.


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
Previous
Finish
Cancel

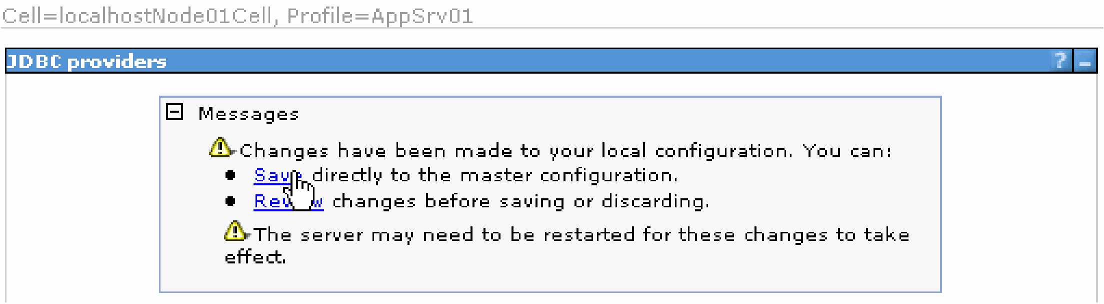
Create a data source

Summary

Summary of actions:

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:orcl
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/Alias_Oracle
Mapping-configuration alias	(none)
Container-managed authentication alias	(none)

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



Cell=localhostNode01Cell, Profile=AppSrv01

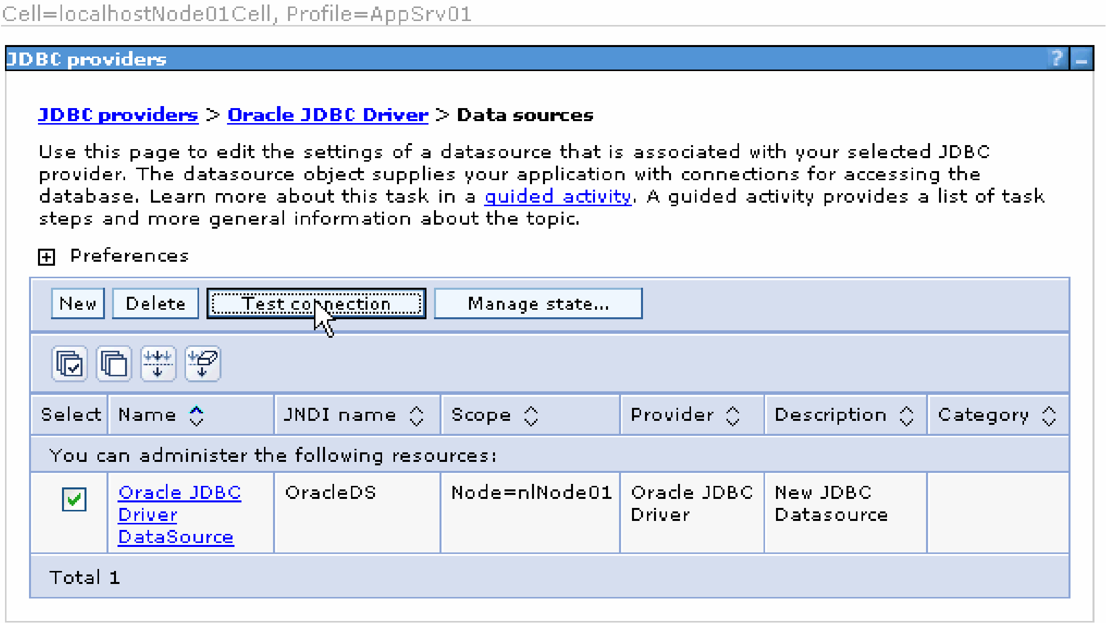
⊟ Messages

⚠ 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.

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

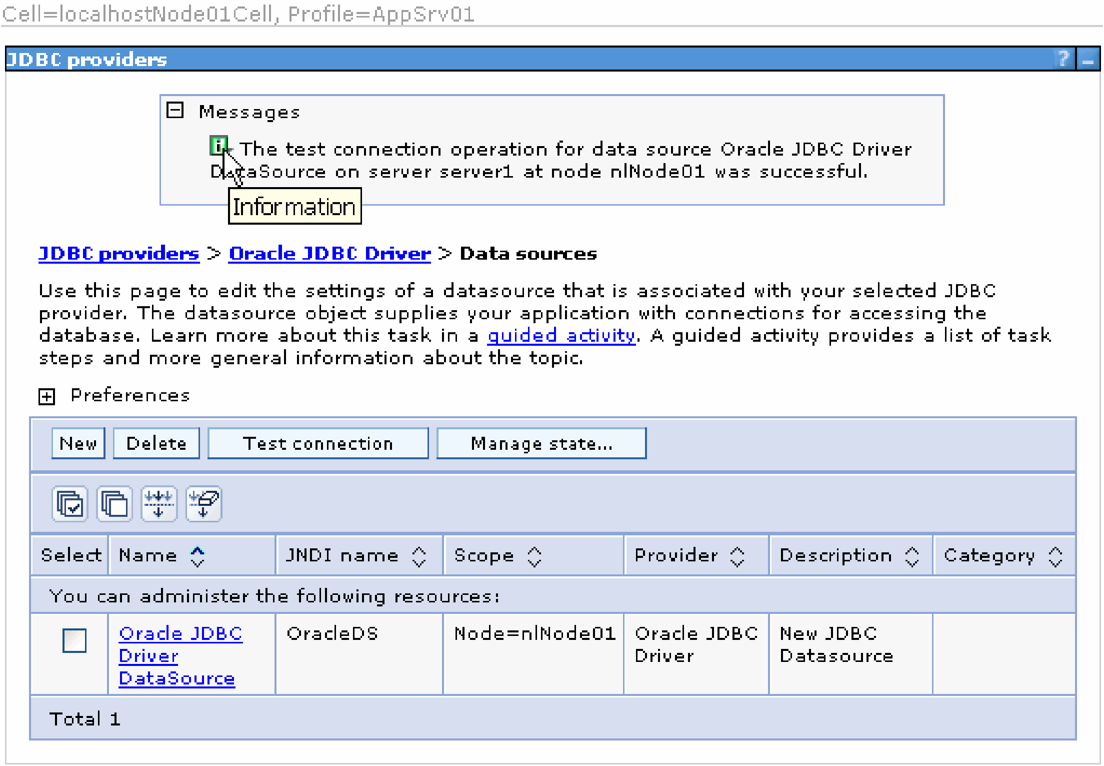
JDBC providers > Oracle JDBC Driver > Data sources

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

Select	Name	JNDI name	Scope	Provider	Description	Category
You can administer the following resources:						
☑	Oracle JDBC Driver DataSource	OracleDS	Node=nlNode01	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

⊟ Messages

The test connection operation for data source Oracle JDBC Driver DataSource on server server1 at node nlNode01 was successful.

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

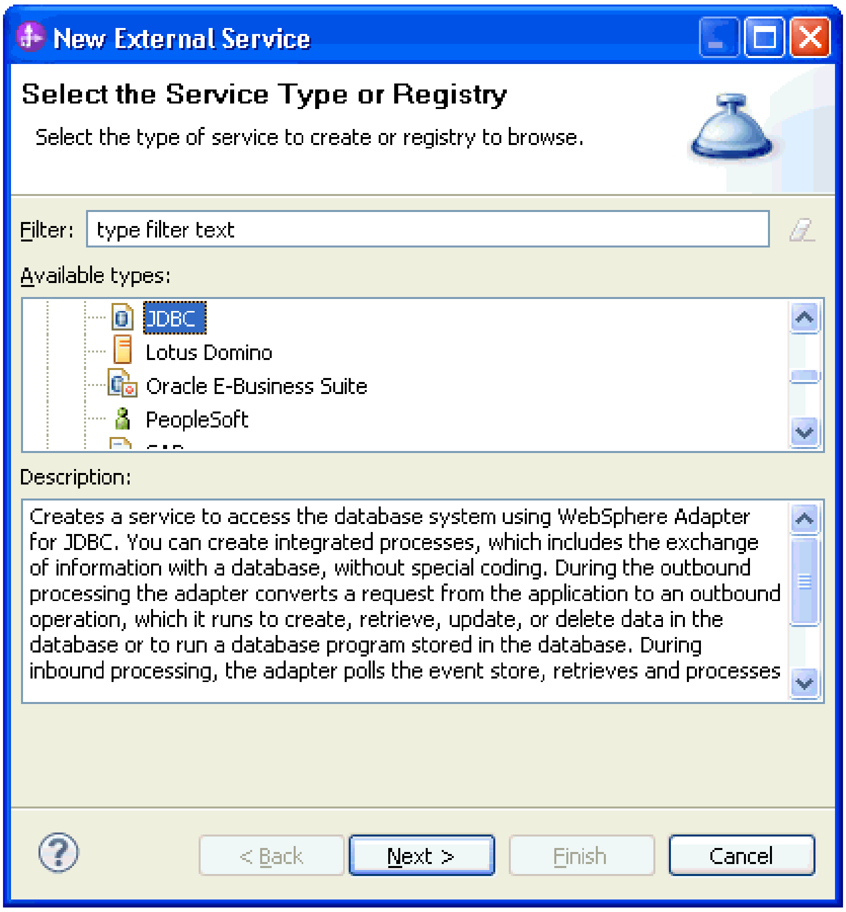
Select	Name	JNDI name	Scope	Provider	Description	Category
You can administer the following resources:						
□	Oracle JDBC Driver DataSource	OracleDS	Node=nlNode01	Oracle JDBC Driver	New JDBC Datasource	
Total 1						

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

Configure the adapter for outbound processing

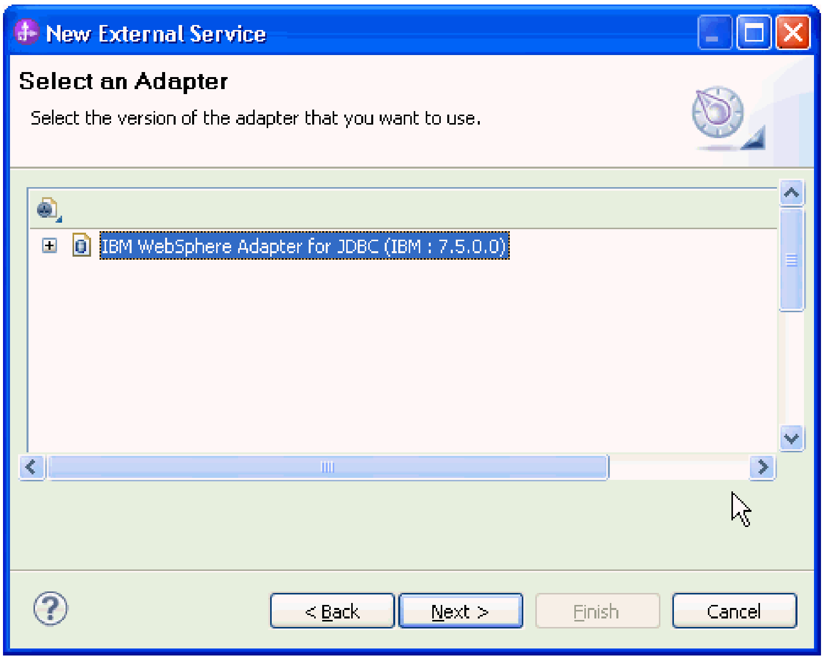
Run the external service wizard to specify business objects, services, and configuration details.

1. Switch to the Business Integration Perspective in IBM Integration Designer 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**.



Creates a service to access the database system using WebSphere Adapter for JDBC. You can create integrated processes, which includes the exchange of information with a database, without special coding. During the outbound processing the adapter converts a request from the application to an outbound operation, which it runs to create, retrieve, update, or delete data in the database or to run a database program stored in the database. During inbound processing, the adapter polls the event store, retrieves and processes

4. Select the **IBM WebSphere Adapter for JDBC (IBM: 7.5.0.0)** and click **Next**.



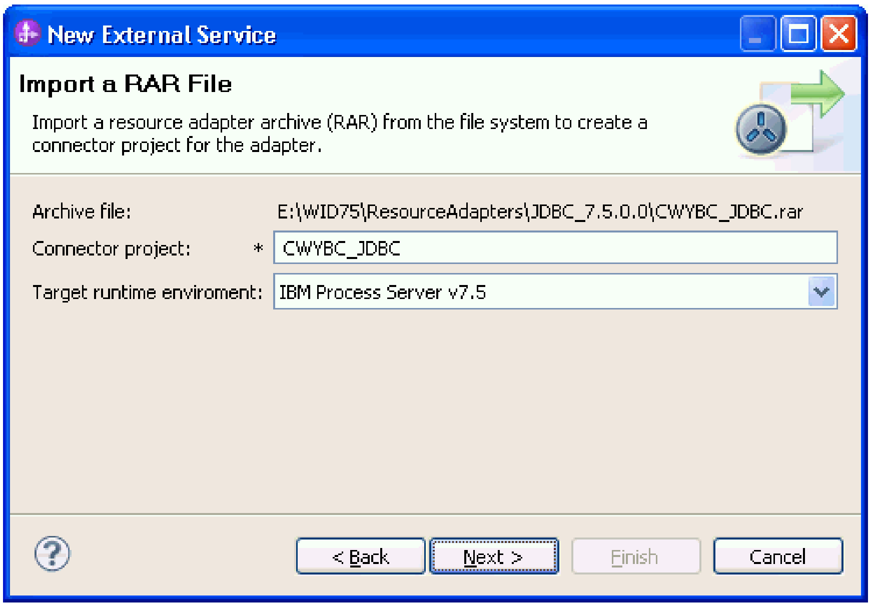
New External Service

Select an Adapter

Select the version of the adapter that you want to use.

IBM WebSphere Adapter for JDBC (IBM : 7.5.0.0)

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



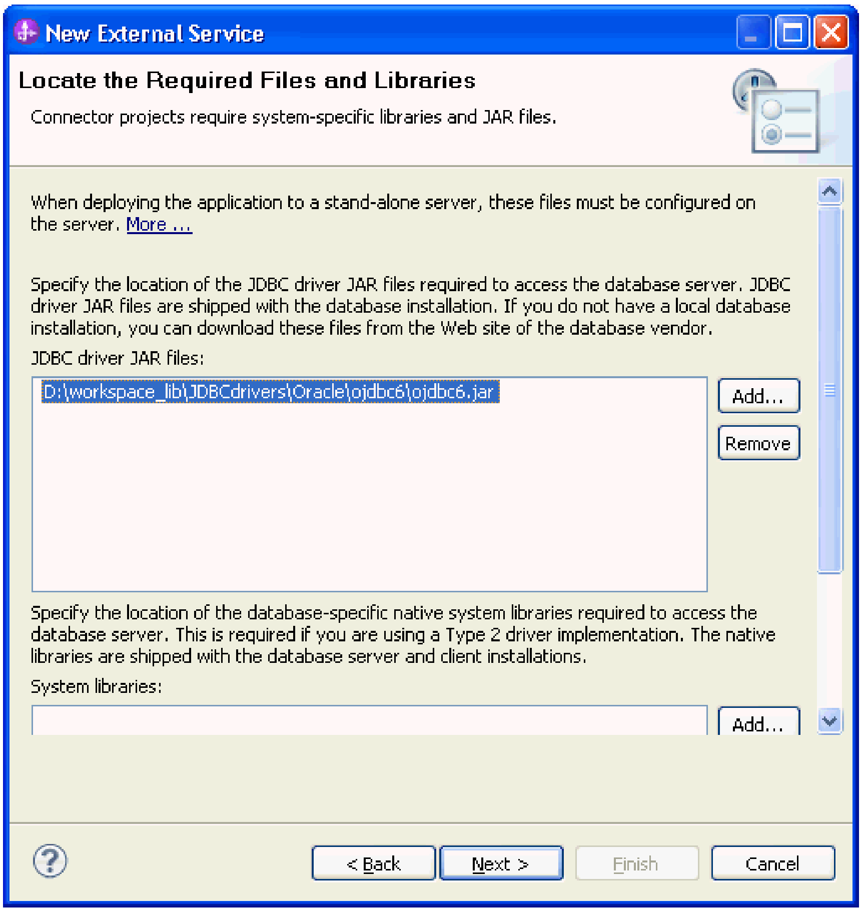
New External Service

Import a RAR File

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

Archive file:	E:\WID75\ResourceAdapters\JDBC_7.5.0.0\CWYBC_JDBC.rar
Connector project: *	CWYBC_JDBC
Target runtime enviroment:	IBM Process Server v7.5

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**.



New External Service

Locate the Required Files and Libraries

Connector projects require system-specific libraries and JAR files.

When deploying the application to a stand-alone server, these files must be configured on the server. More ...

Specify the location of the JDBC driver JAR files required to access the database server. JDBC driver JAR files are shipped with the database installation. If you do not have a local database installation, you can download these files from the Web site of the database vendor.

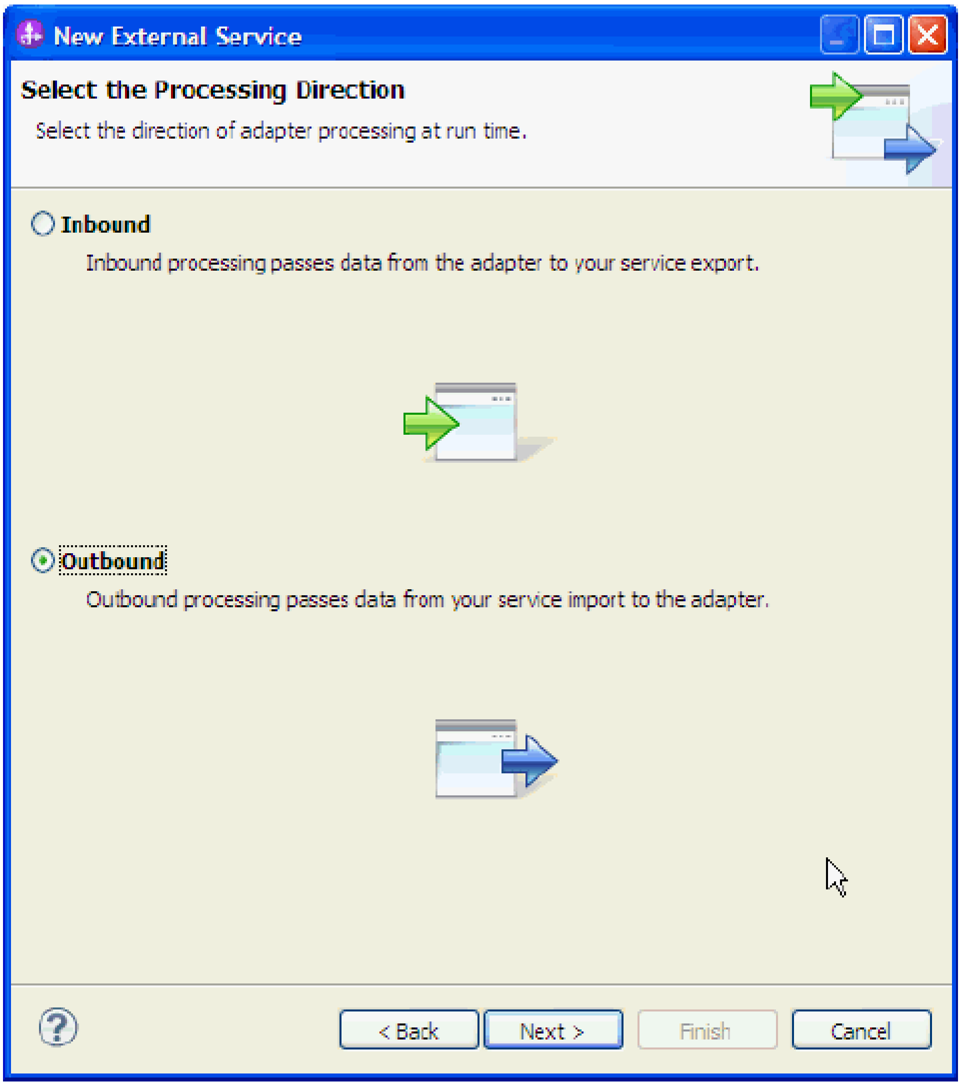
JDBC driver JAR files:

D:\workspace_lib\JDBCdrivers\Oracle\ojdbc6\ojdbc6.jar

Specify the location of the database-specific native system libraries required to access the database server. This is required if you are using a Type 2 driver implementation. The native libraries are shipped with the database server and client installations.

System libraries:

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



New External Service

Select the Processing Direction

Select the direction of adapter processing at run time.

○ Inbound

Inbound processing passes data from the adapter to your service export.

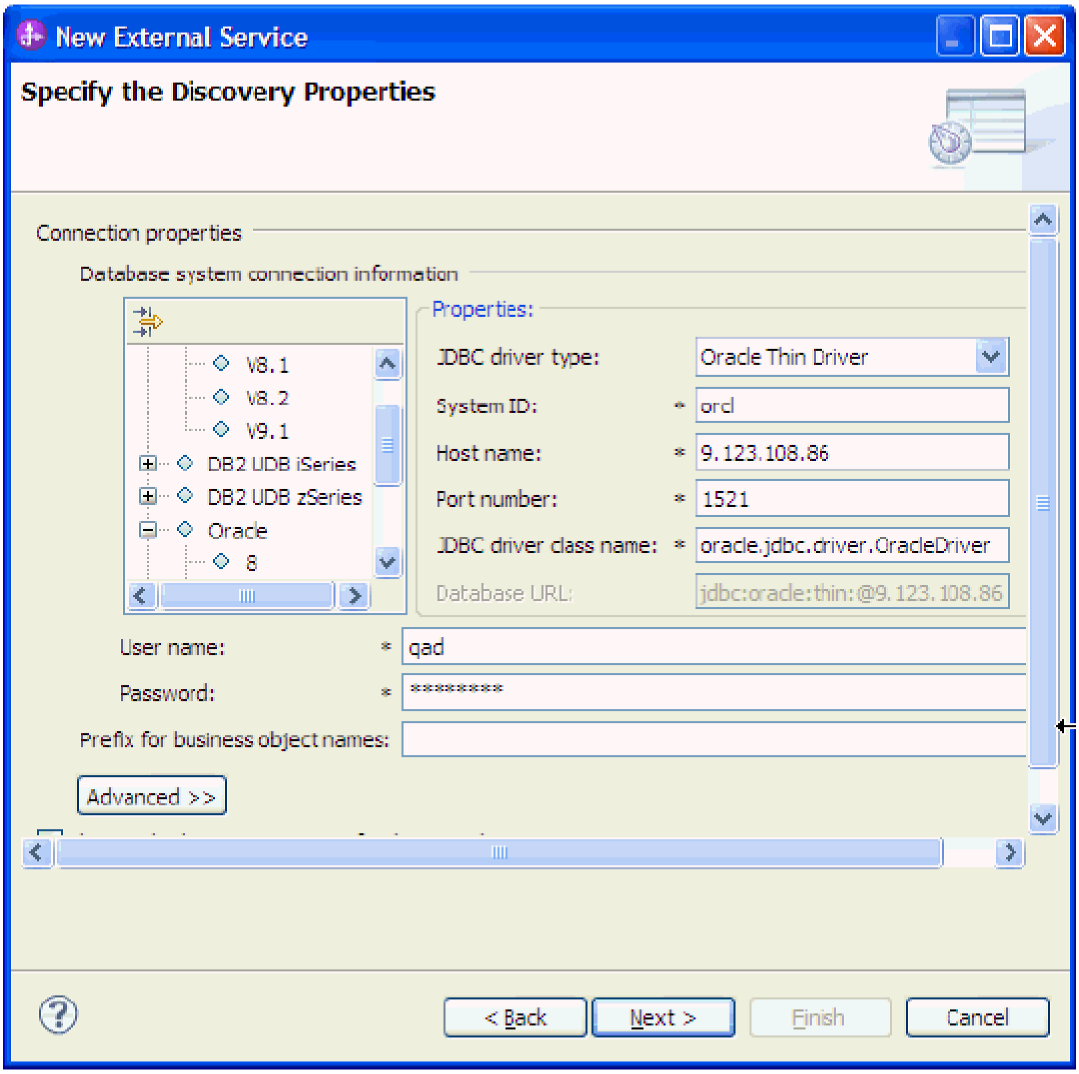
● Outbound

Outbound processing passes data from your service import to the adapter.

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**.



New External Service

Specify the Discovery Properties

Connection properties

Database system connection information

Properties:

JDBC driver type: Oracle Thin Driver

System ID: * orcl

Host name: * 9.123.108.86

Port number: * 1521

JDBC driver class name: * oracle.jdbc.driver.OracleDriver

Database URL: jdbc:oracle:thin:@9.123.108.86

User name: * qad

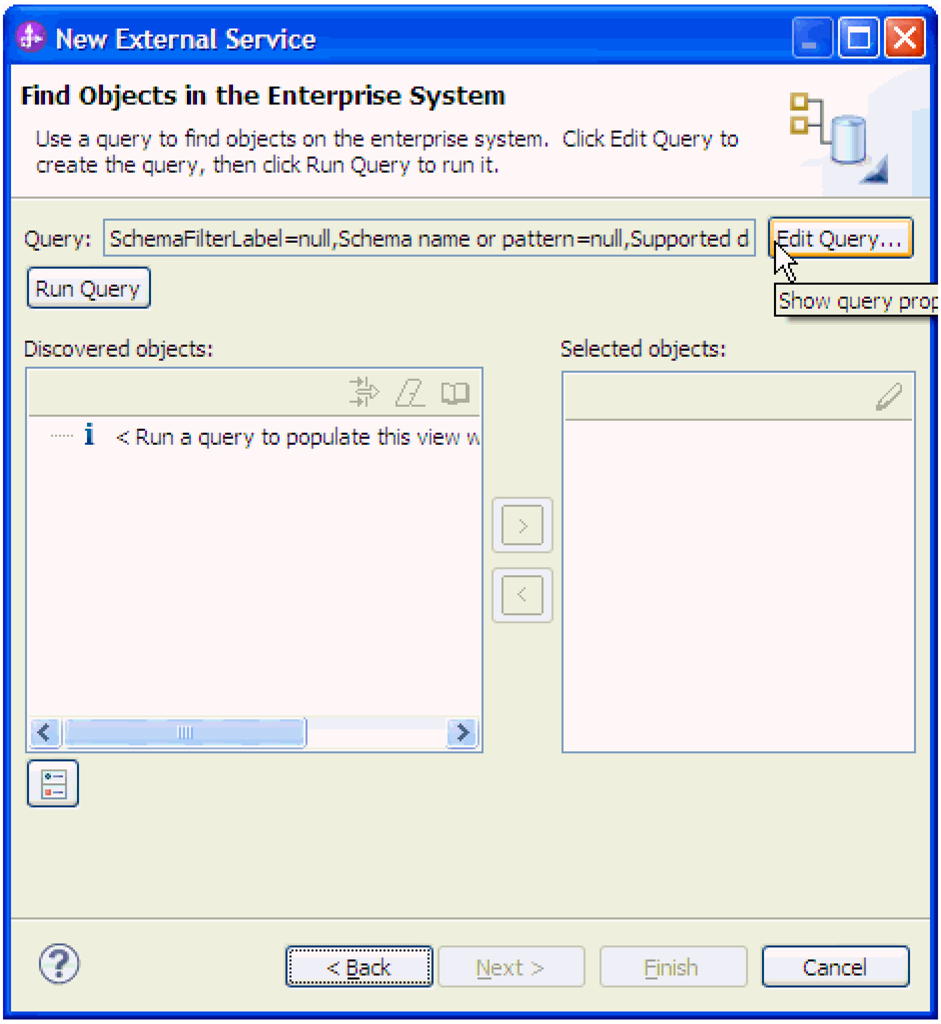
Password: * *********

Prefix for business object names:

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**.



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.

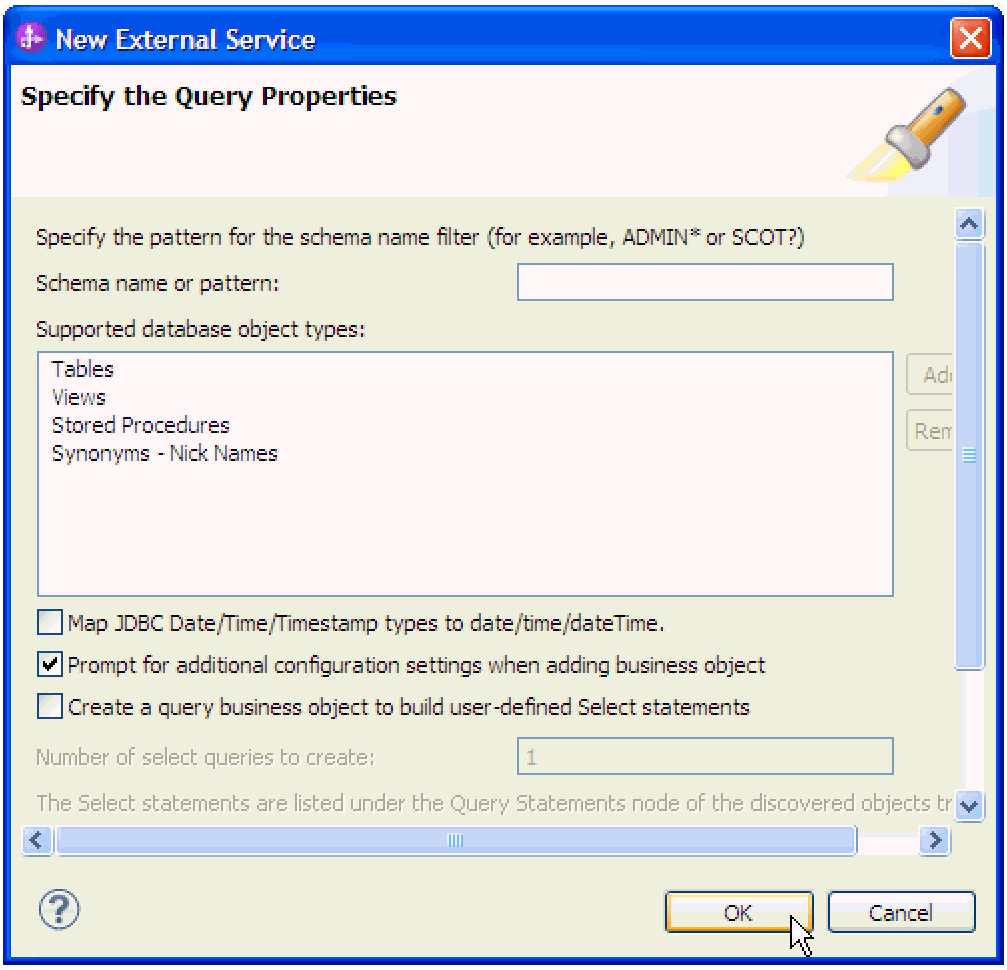
Query: SchemaFilterLabel=null,Schema name or pattern=null,Supported d

Discovered objects:

< Run a query to populate this view w

Selected objects:

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



Specify the Query Properties

Specify the pattern for the schema name filter (for example, ADMIN* or SCOT?)

Schema name or pattern:

Supported database object types:

Tables
Views
Stored Procedures
Synonyms - Nick Names

□ Map JDBC Date/Time/Timestamp types to date/time/dateTime.

☑ Prompt for additional configuration settings when adding business object

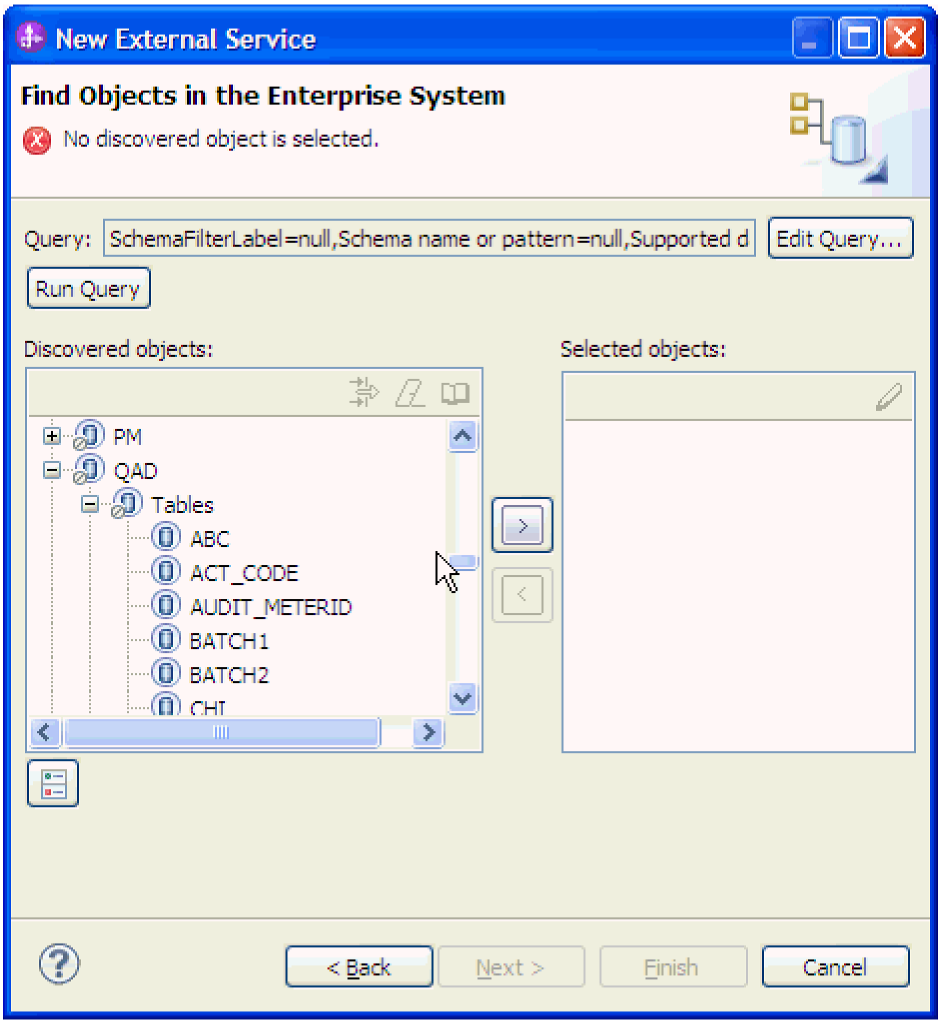
□ Create a query business object to build user-defined Select statements

Number of select queries to create: 1

The Select statements are listed under the Query Statements node of the discovered objects tr

3. Click **Run Query**.

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



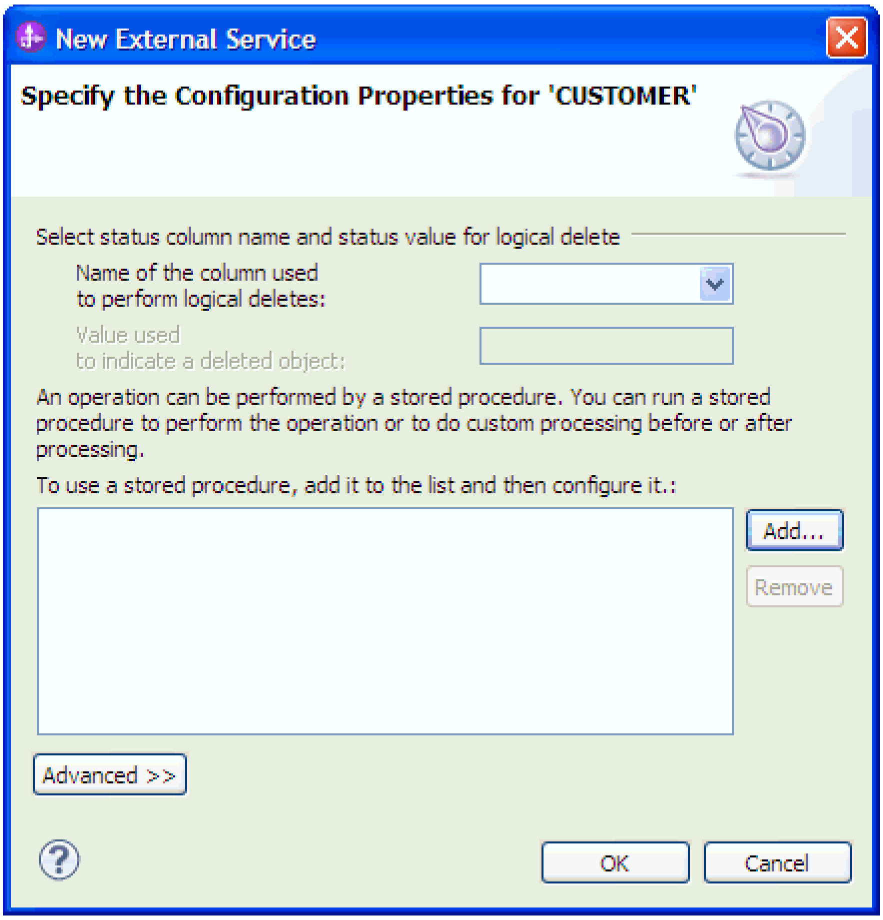
New External Service

Find Objects in the Enterprise System

No discovered object is selected.

Query: SchemaFilterLabel=null,Schema name or pattern=null,Supported d

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



New External Service

Specify the Configuration Properties for 'CUSTOMER'

Select status column name and status value for logical delete

Name of the column used to perform logical deletes:

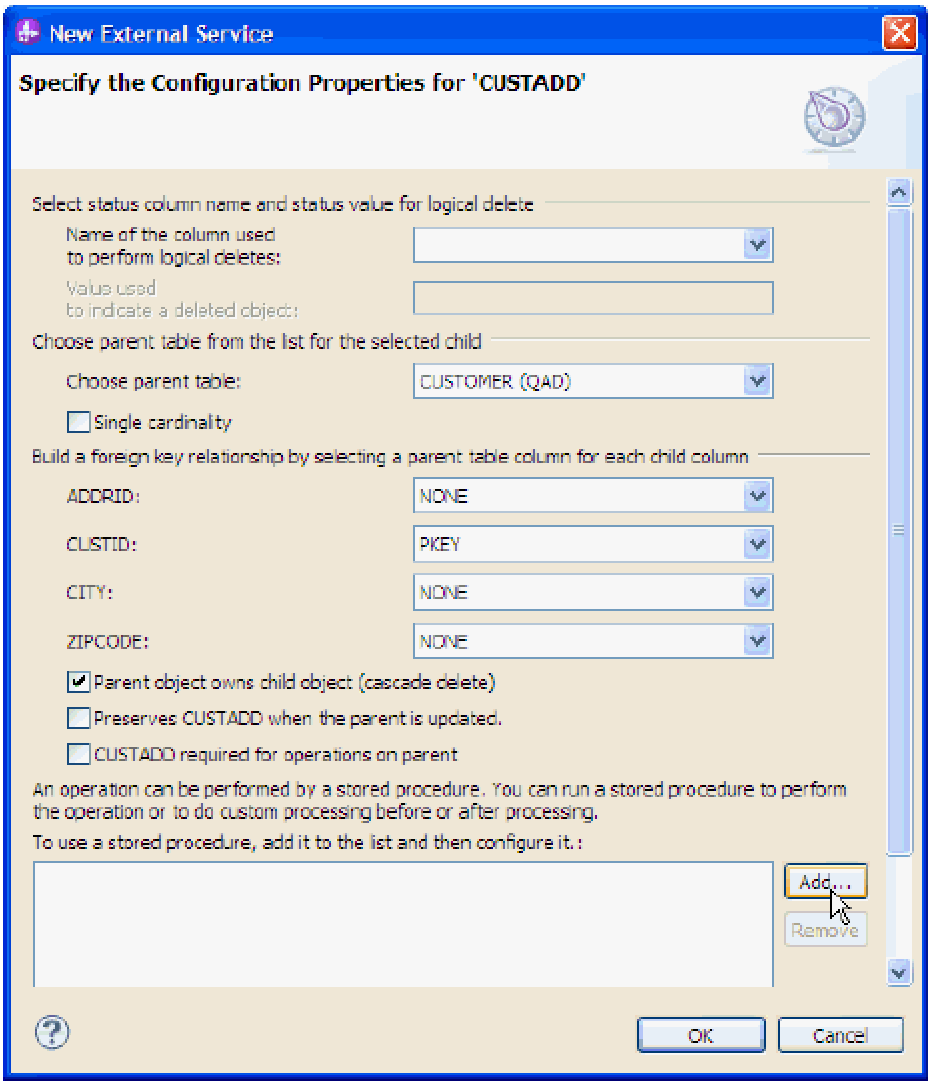
Value used to indicate a deleted object:

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.:

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 (QAD)

□ 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)

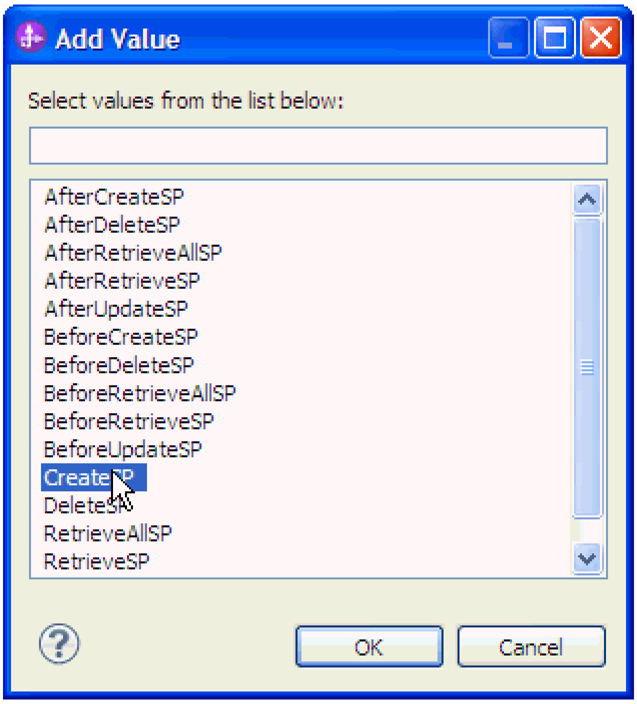
□ Preserves CUSTADD when the parent is updated.

□ 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**.

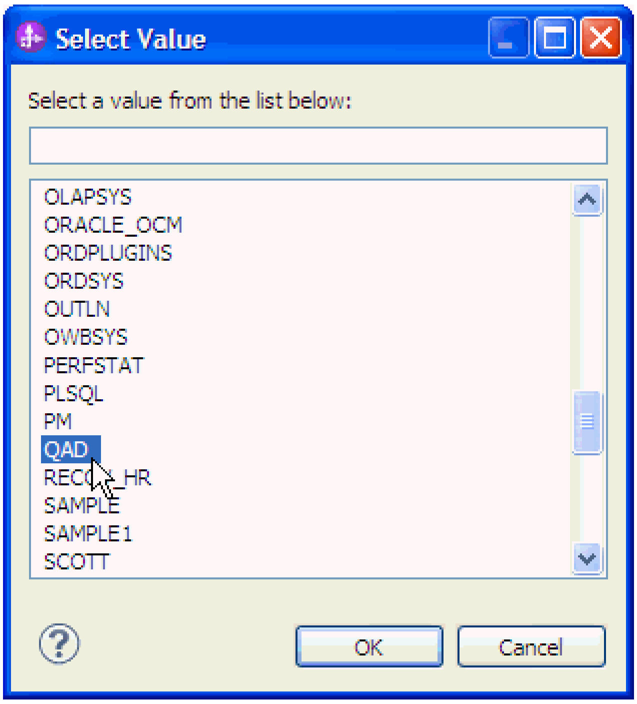


Add Value

Select values from the list below:

AfterCreateSP
AfterDeleteSP
AfterRetrieveAllSP
AfterRetrieveSP
AfterUpdateSP
BeforeCreateSP
BeforeDeleteSP
BeforeRetrieveAllSP
BeforeRetrieveSP
BeforeUpdateSP
CreateSP
DeleteSP
RetrieveAllSP
RetrieveSP

9. Select **QAD** for schema name.

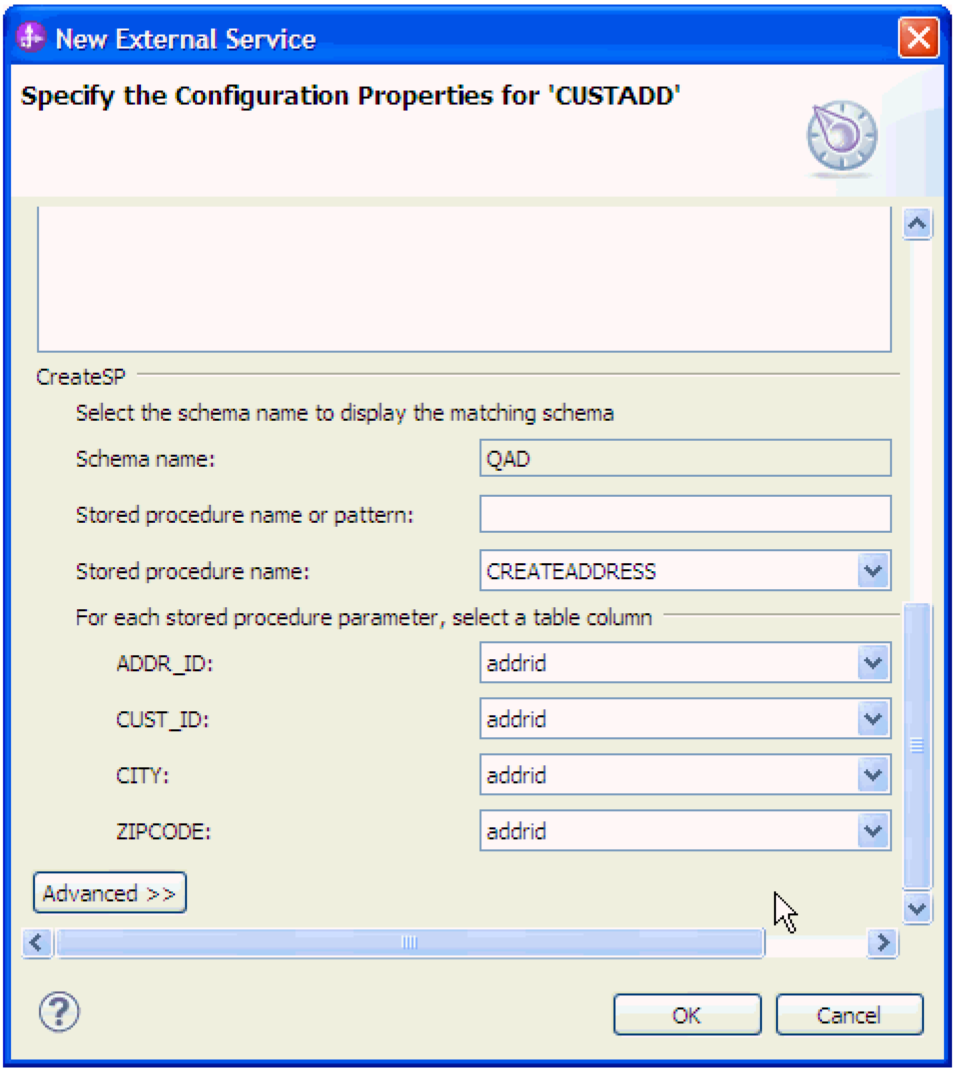


Select Value

Select a value from the list below:

- OLAPSYS
- ORACLE_OCM
- ORDPLUGINS
- ORDSYS
- OUTLN
- OWBSYS
- PERFSTAT
- PLSQL
- PM
- QAD
- RECO_HR
- SAMPLE
- SAMPLE1
- SCOTT

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



Specify the Configuration Properties for 'CUSTADD'

CreateSP

Select the schema name to display the matching schema

Schema name: QAD

Stored procedure name or pattern:

Stored procedure name: CREATEADDRESS

For each stored procedure parameter, select a table column

ADDR_ID: addrid

CUST_ID: addrid

CITY: addrid

ZIPCODE: addrid

11. Select stored procedure parameter for each column.

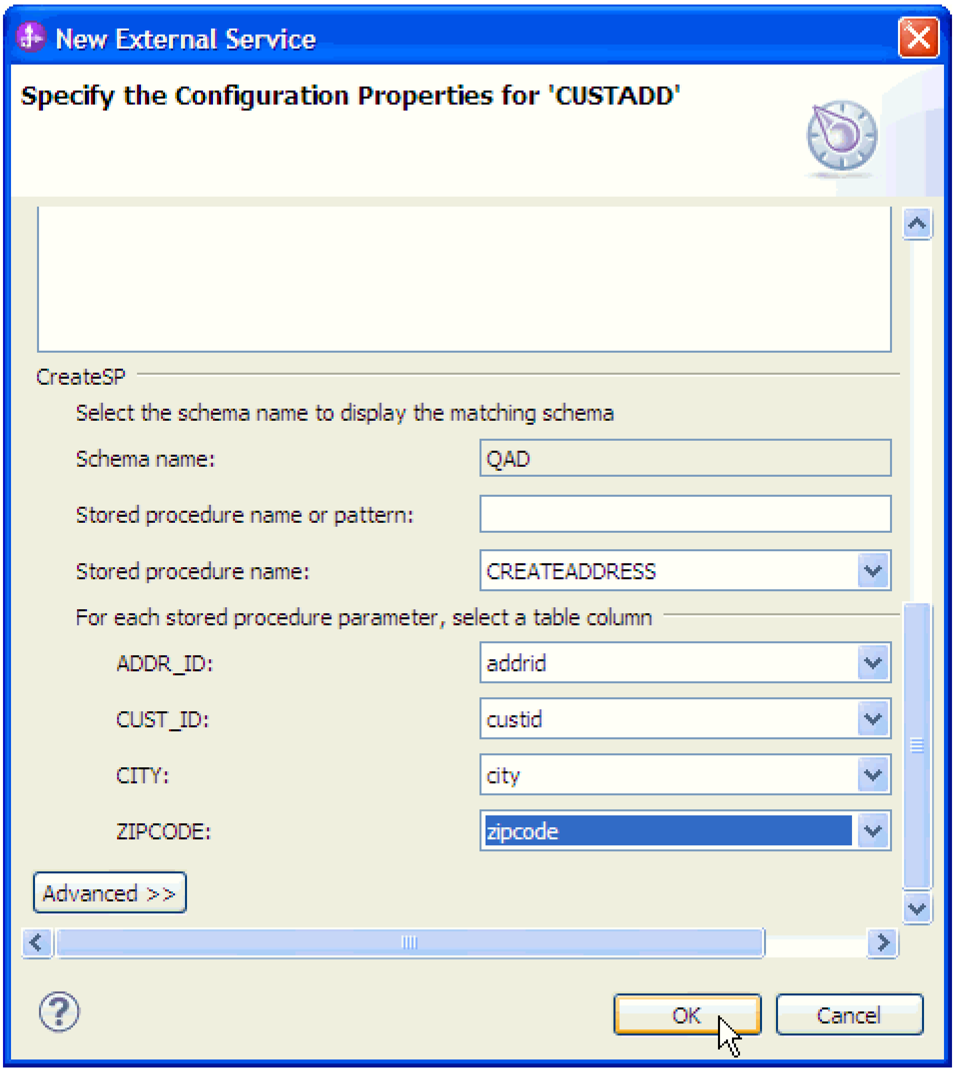
    ADDR_ID: addrid

    CUST_ID: custid

    CITY: city

    ZIPCODE: zipcode

12. Click **OK**.



New External Service

Specify the Configuration Properties for 'CUSTADD'

CreateSP

Select the schema name to display the matching schema

Schema name: QAD

Stored procedure name or pattern:

Stored procedure name: CREATEADDRESS

For each stored procedure parameter, select a table column

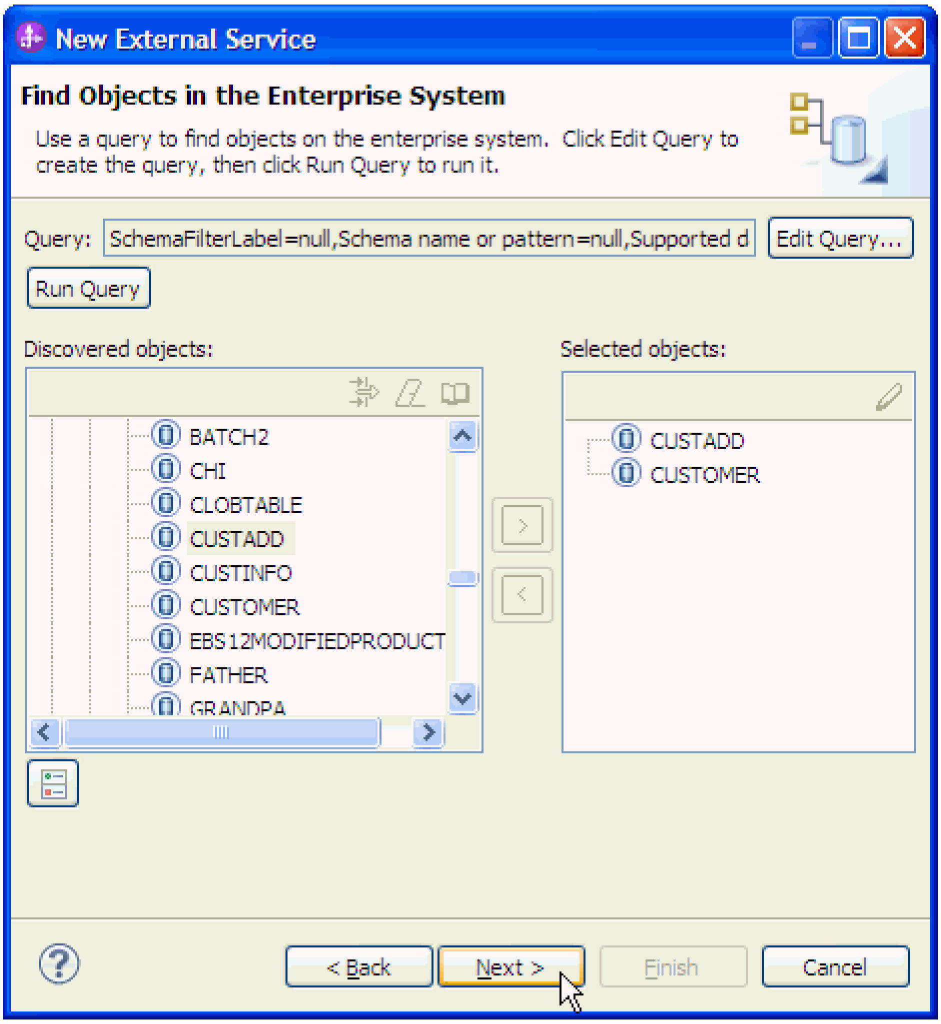
ADDR_ID: addrid

CUST_ID: custid

CITY: city

ZIPCODE: zipcode

13. In the Find Objects in Enterprise System window, 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 d

Discovered objects:

- BATCH2
- CHI
- CLOBTABLE
- CUSTADD
- CUSTINFO
- CUSTOMER
- EBS12MODIFIEDPRODUCT
- FATHER
- GRANDPA

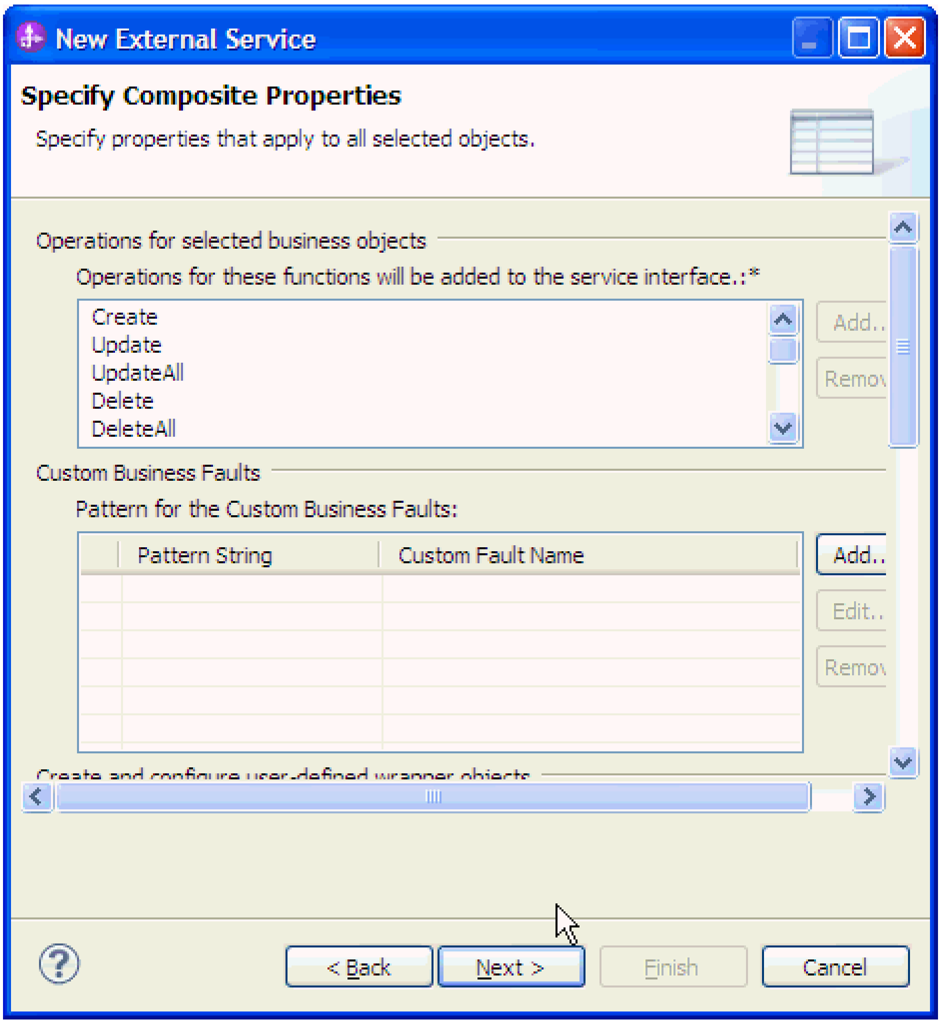
Selected objects:

- CUSTADD
- CUSTOMER

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**.



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
- UpdateAll
- Delete
- DeleteAll

Custom Business Faults

Pattern for the Custom Business Faults:

Pattern String	Custom Fault Name

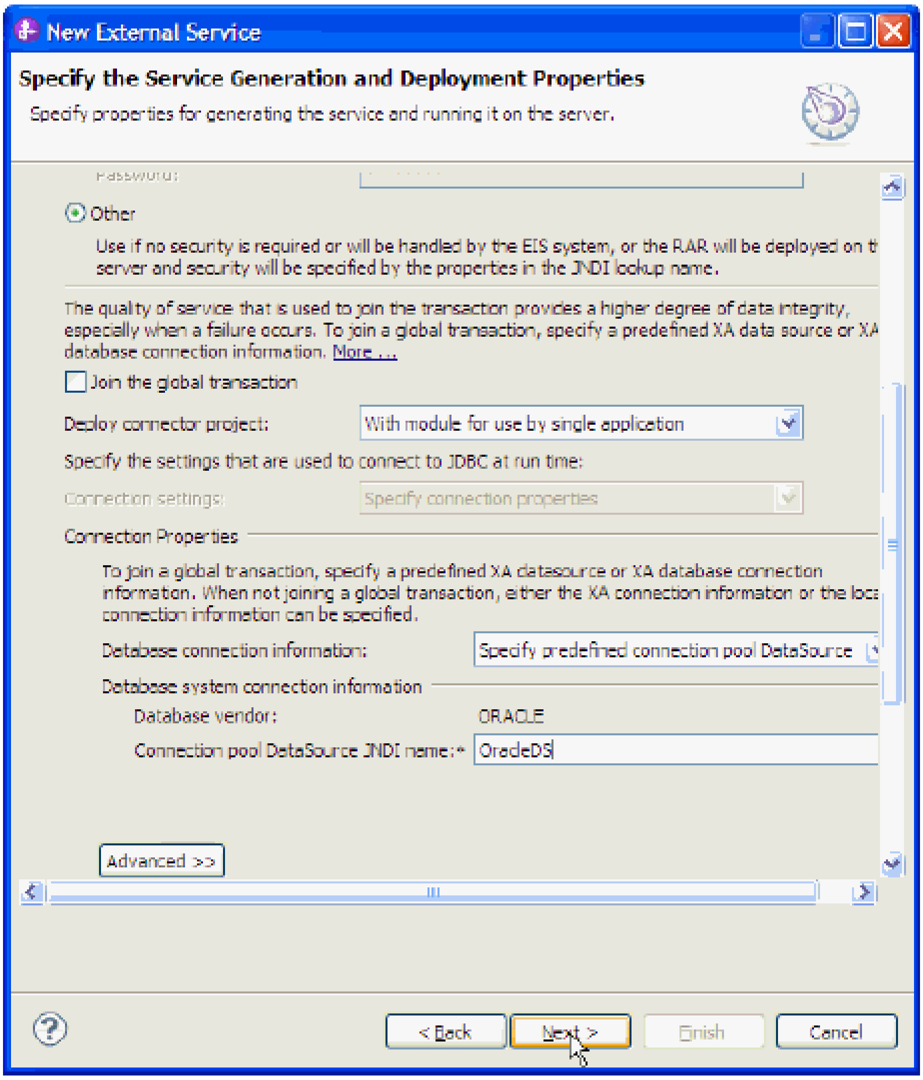
2. In the Specify the Service Generation and Deployment window, perform the following steps:

   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**.



Specify the Service Generation and Deployment Properties

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

Password:

● Other

Use if no security is required or will be handled by the EIS system, or the RAR will be deployed on th 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 join a global transaction, specify a predefined XA data source or XA database connection information. More ...

□ Join the global transaction

Deploy connector project: With module for use by single application

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

Connection settings: Specify connection properties

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 loca connection information can be specified.

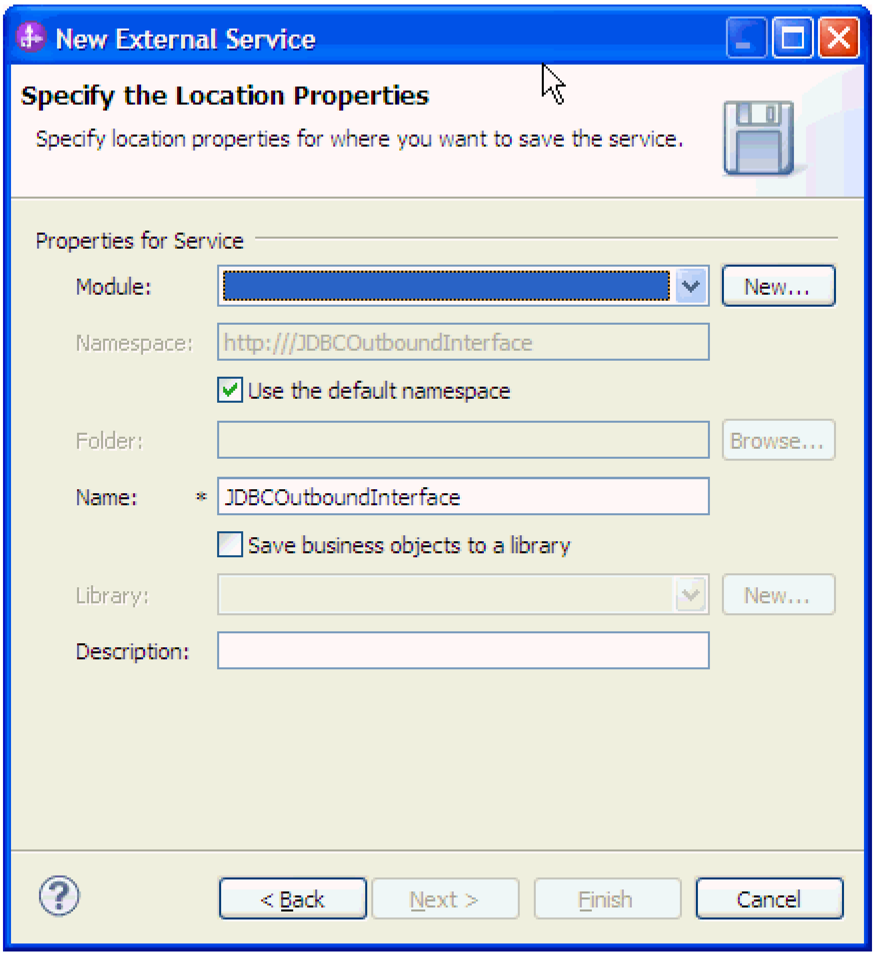
Database connection information: Specify predefined connection pool DataSource

Database system connection information

Database vendor: ORACLE

Connection pool DataSource JNDI name:* OracleDS

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



New External Service

Specify the Location Properties

Specify location properties for where you want to save the service.

Properties for Service

Module:

Namespace: http:///JDBCOutboundInterface

☑ Use the default namespace

Folder:

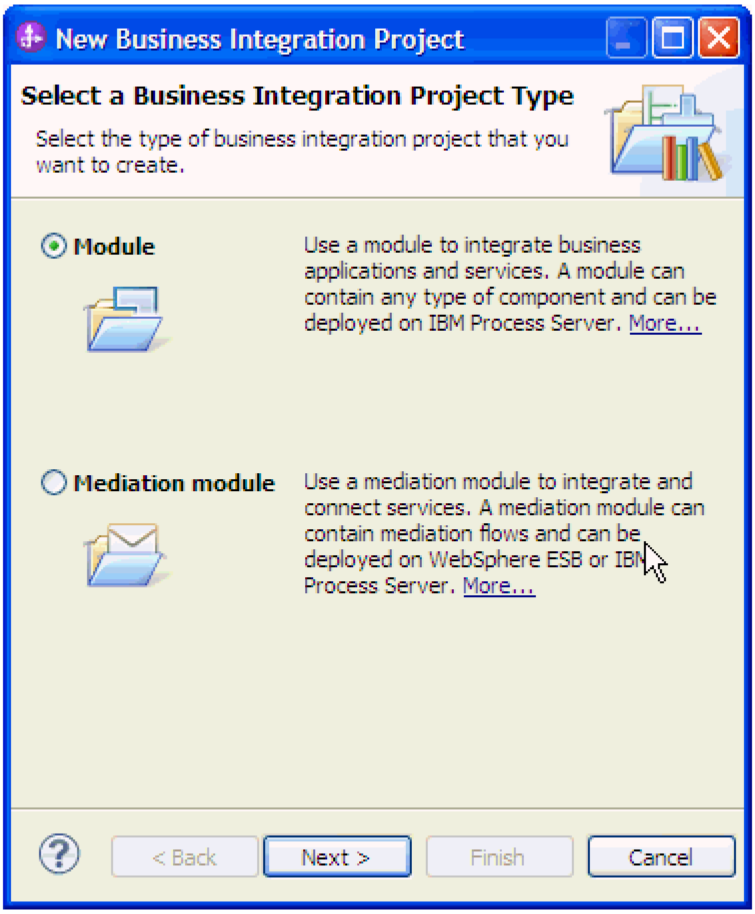
Name: * JDBCOutboundInterface

□ Save business objects to a library

Library:

Description:

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



New Business Integration Project

Select a Business Integration Project Type

Select the type of business integration project that you want to create.

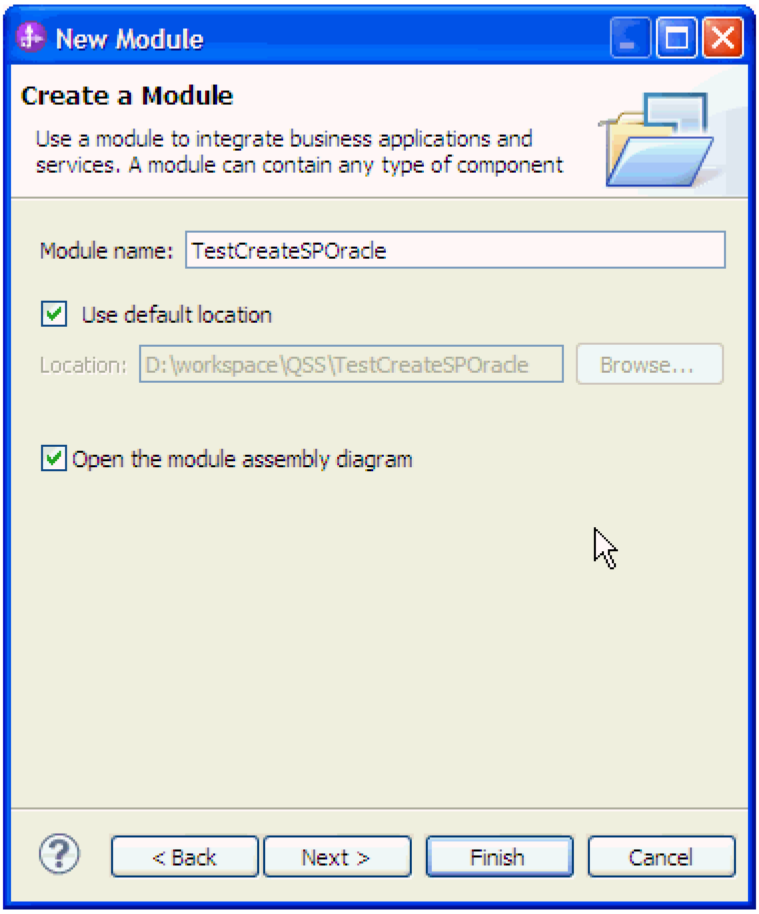
● Module

Use a module to integrate business applications and services. A module can contain any type of component and can be deployed on IBM Process Server. More...

○ Mediation module

Use a mediation module to integrate and connect services. A mediation module can contain mediation flows and can be deployed on WebSphere ESB or IBM Process Server. More...

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



Create a Module

Use a module to integrate business applications and services. A module can contain any type of component

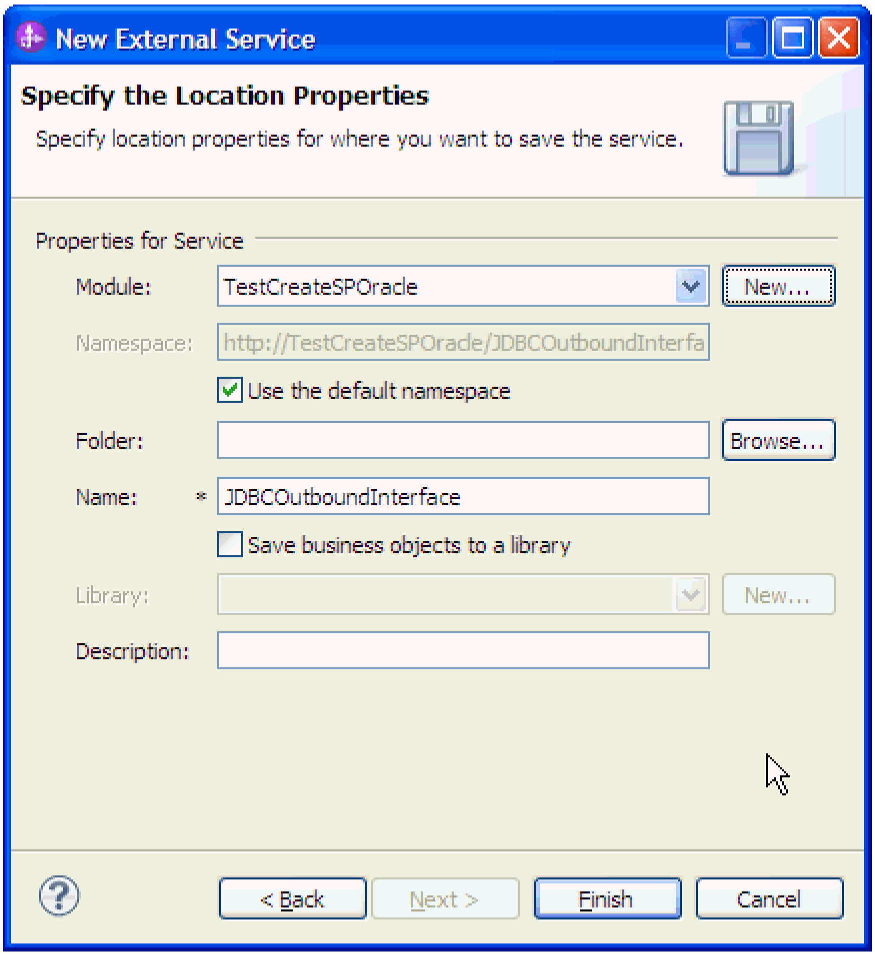
Module name: TestCreateSPOracle

☑ Use default location

Location: D:\workspace\QSS\TestCreateSPOracle

☑ Open the module assembly diagram

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



New External Service

Specify the Location Properties

Specify location properties for where you want to save the service.

Properties for Service

Module: TestCreateSPOracle

Namespace: http://TestCreateSPOracle/JDBCOutboundInterfa

☑ Use the default namespace

Folder:

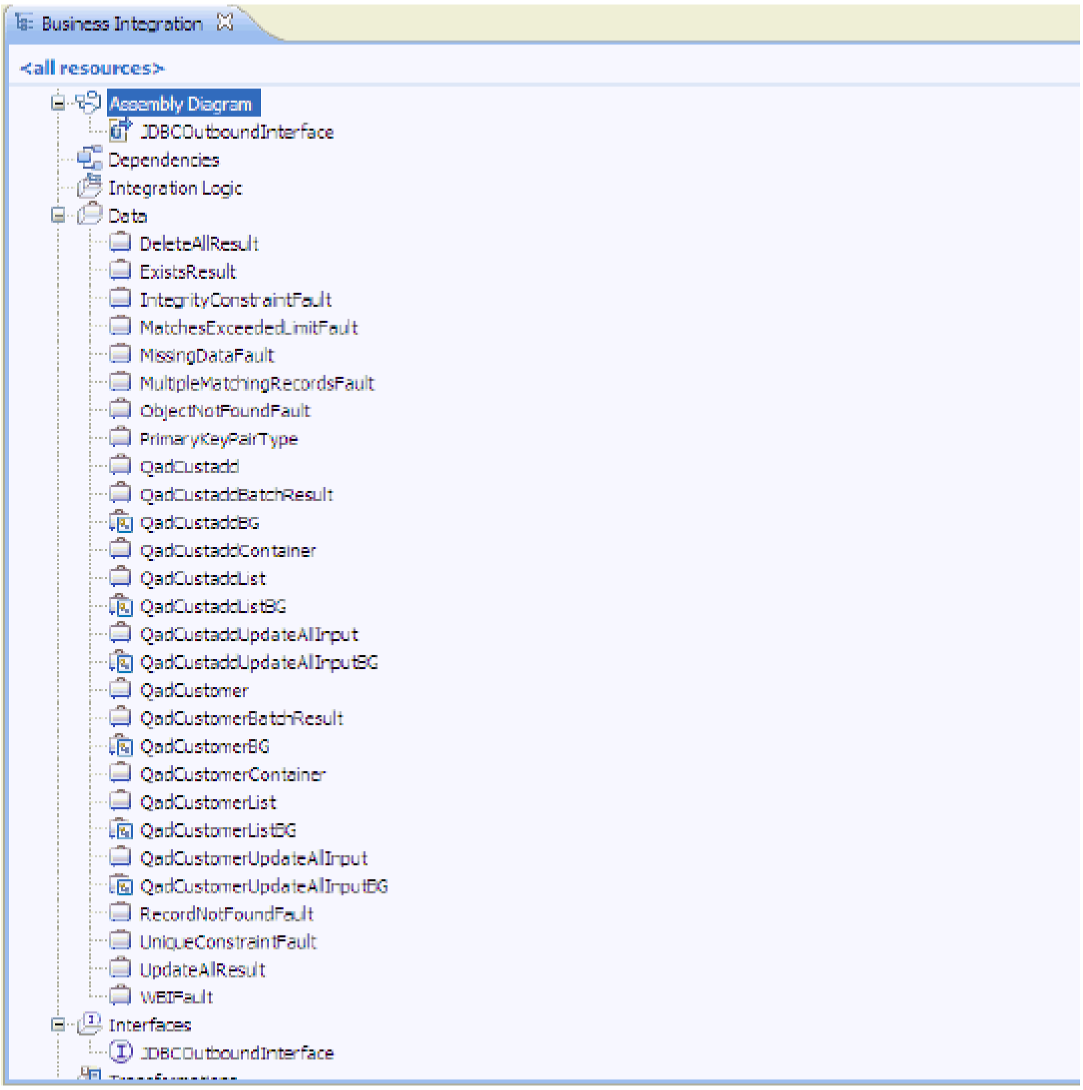
Name: * JDBCOutboundInterface

□ Save business objects to a library

Library:

Description:

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



Business Integration

<all resources>

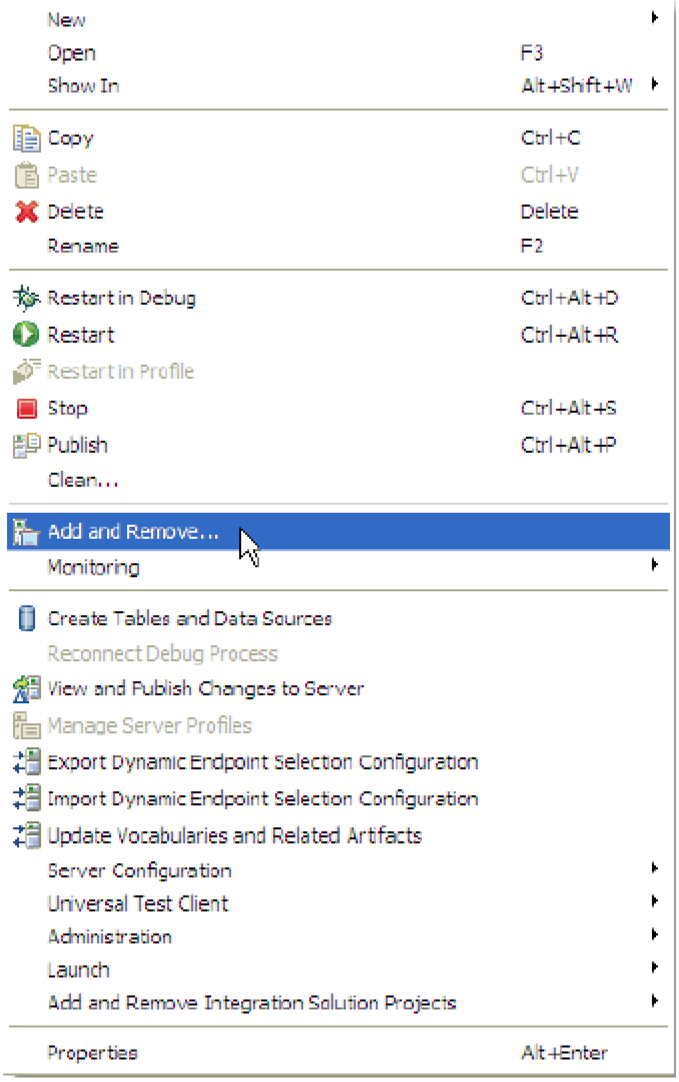
- Assembly Diagram
  - JDBCOutboundInterface
- Dependencies
- Integration Logic
- Data
  - DeleteAllResult
  - ExistsResult
  - IntegrityConstraintFault
  - MatchesExceededLimitFault
  - MissingDataFault
  - MultipleMatchingRecordsFault
  - ObjectNotFoundFault
  - PrimaryKeyPairType
  - QadCustadd
  - QadCustaddBatchResult
  - QadCustaddBG
  - QadCustaddContainer
  - QadCustaddList
  - QadCustaddListBG
  - QadCustaddUpdateAllInput
  - QadCustaddUpdateAllInputBG
  - QadCustomer
  - QadCustomerBatchResult
  - QadCustomerBG
  - QadCustomerContainer
  - QadCustomerList
  - QadCustomerListBG
  - QadCustomerUpdateAllInput
  - QadCustomerUpdateAllInputBG
  - RecordNotFoundFault
  - UniqueConstraintFault
  - UpdateAllResult
  - WBIFault
- Interfaces
  - JDBCOutboundInterface

Deploy the module to the test environment

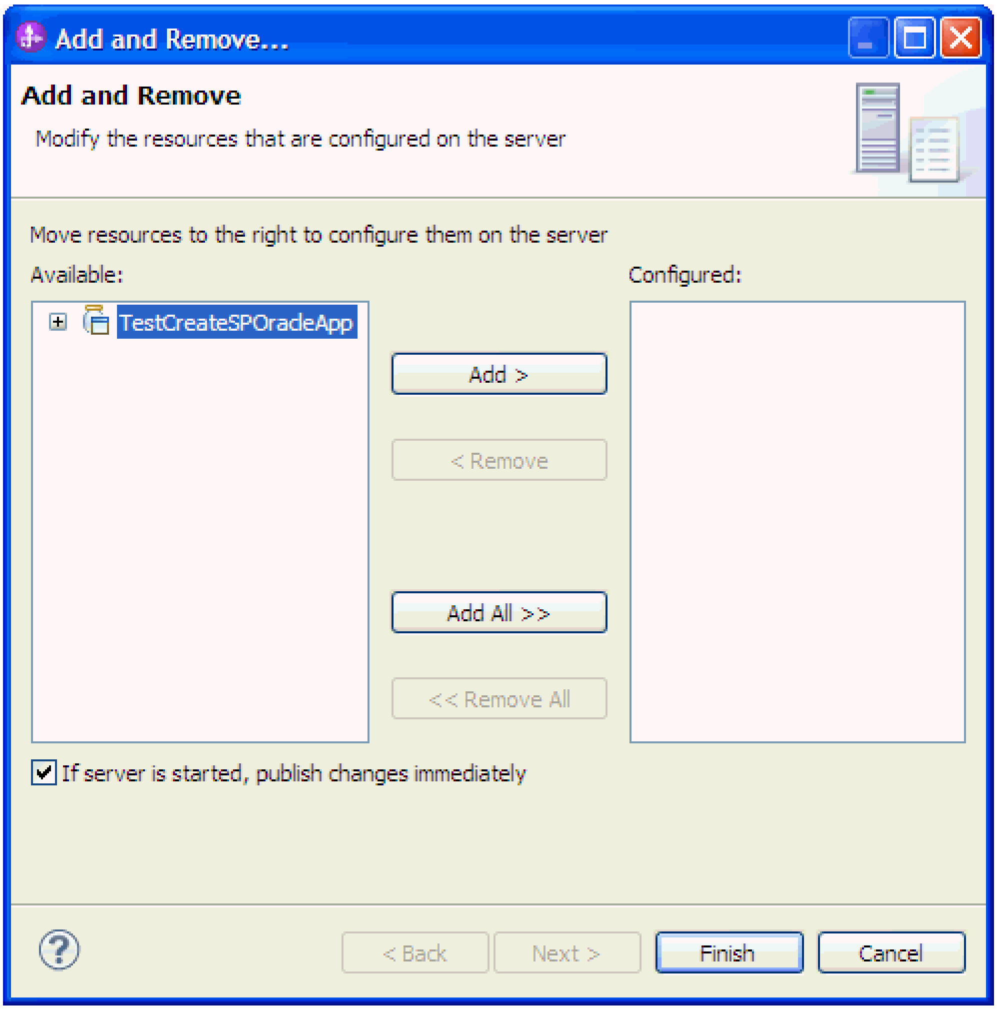
After running the external service wizard, you will have an SCA module that contains an Enterprise Information System import. You must install this SCA module in the IBM Integration Designer integration test client. To do this, you must add the SCA module you created earlier to the server using the **Servers**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

Steps for adding the SCA module to the server:

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



The Add and Remove… window lists the available projects in the IBM Integration Designer workspace.



Add and Remove...

Add and Remove

Modify the resources that are configured on the server

Move resources to the right to configure them on the server

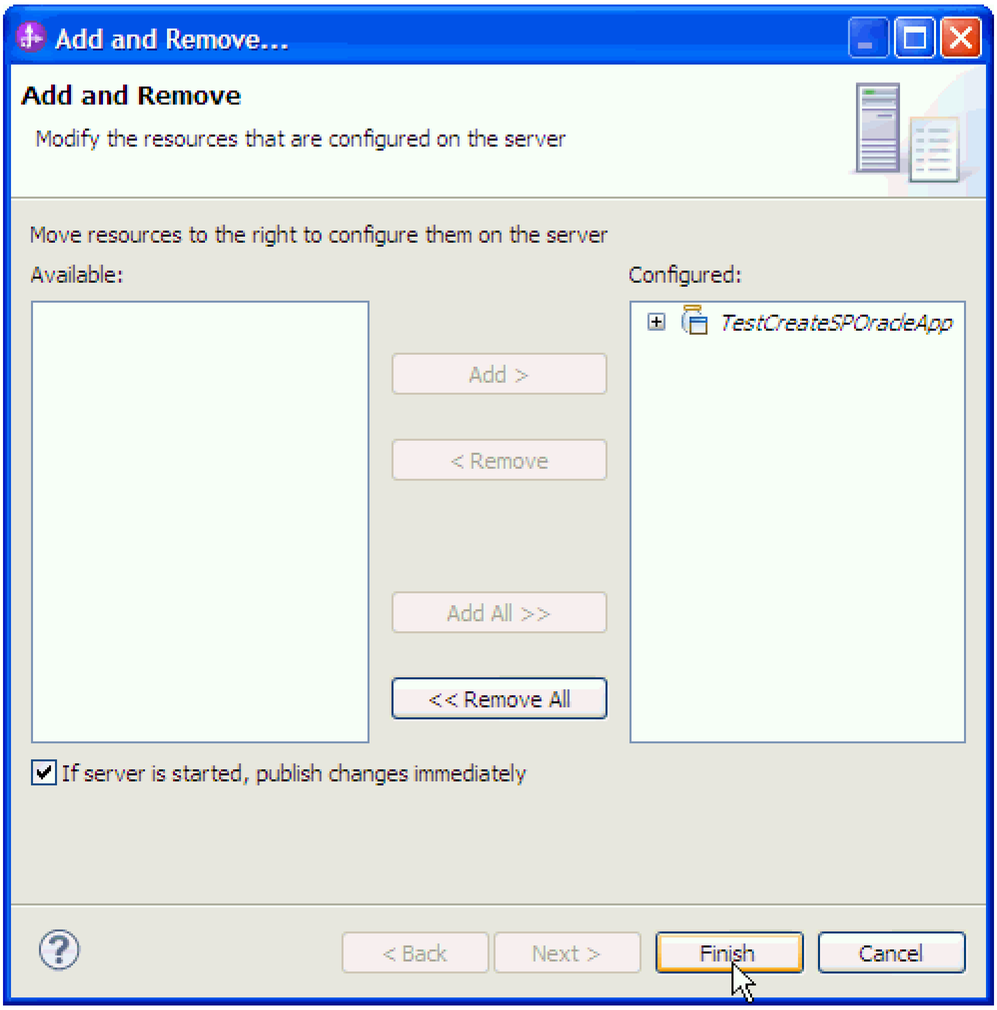
Available:

Configured:

⊞ TestCreateSPOracleApp

☑ If server is started, publish changes immediately

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



Add and Remove...

Add and Remove

Modify the resources that are configured on the server

Move resources to the right to configure them on the server

Available:

Configured:

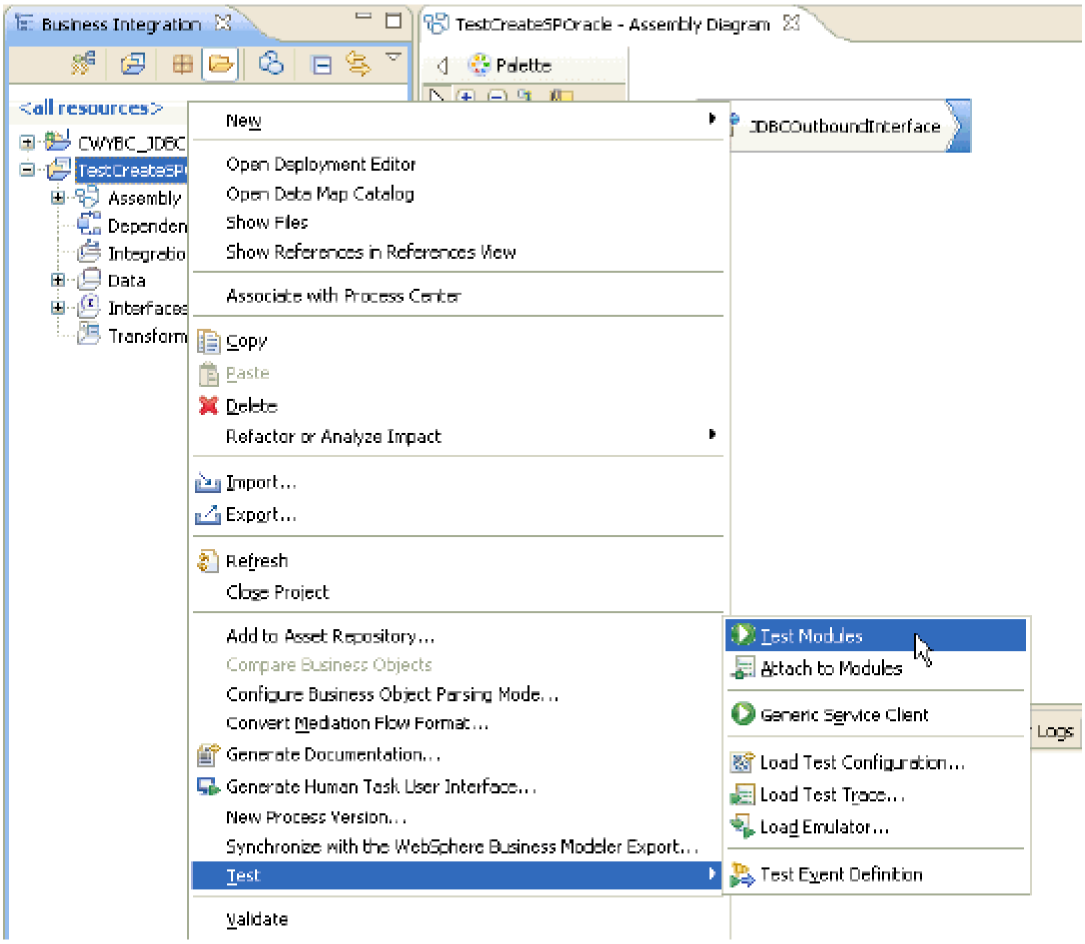
TestCreateSPOracleApp

☑ If server is started, publish changes immediately

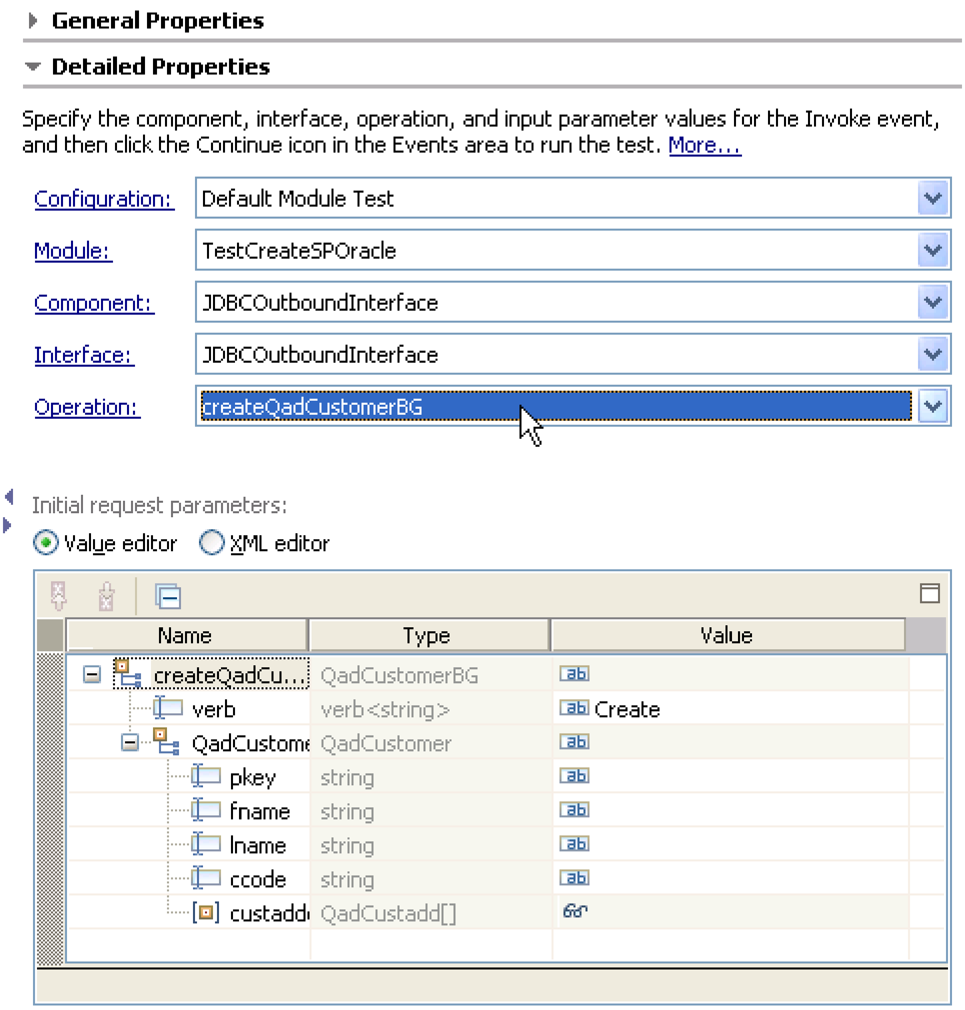
Test the assembled adapter application

Test the assembled adapter application using the IBM Integration Designer integration test client.

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



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



▸ General Properties

▾ Detailed Properties

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

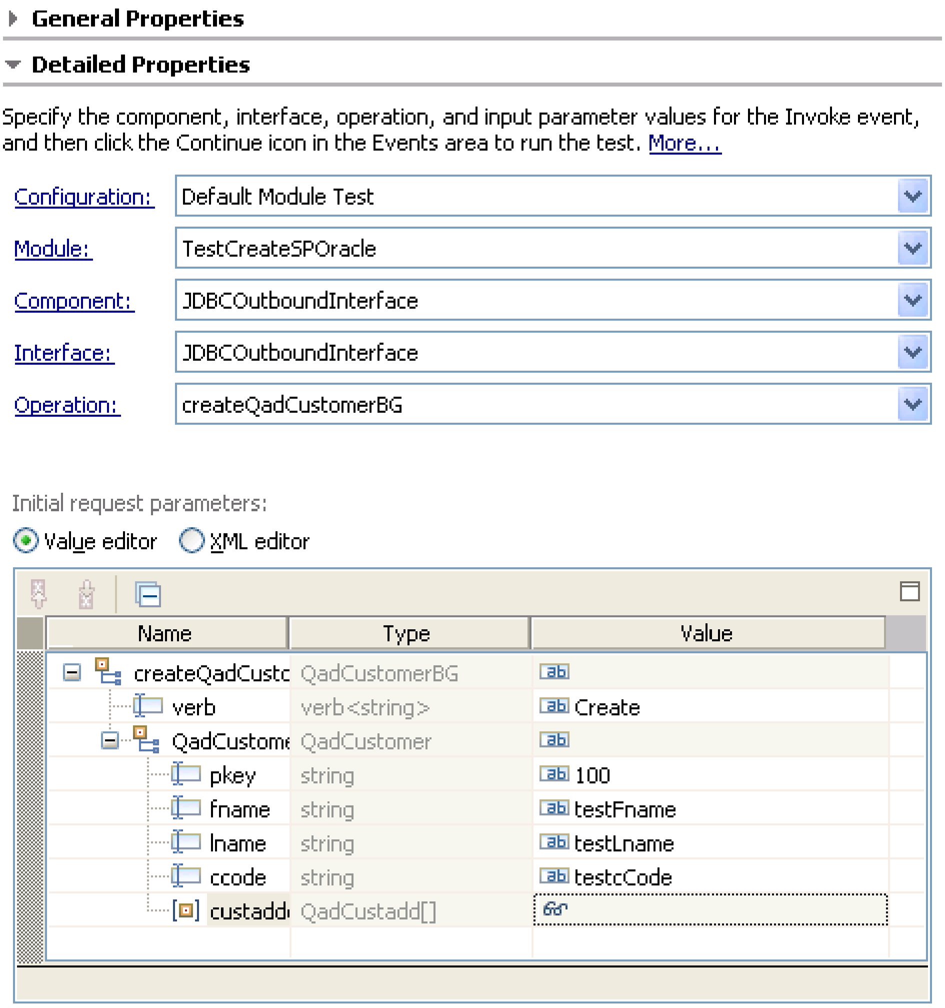
Configuration:	Default Module Test
Module:	TestCreateSPOracle
Component:	JDBCOutboundInterface
Interface:	JDBCOutboundInterface
Operation:	createQadCustomerBG

Initial request parameters:

● Value editor ○ XML editor

Name	Type	Value
createQadCu...	QadCustomerBG	
verb	verb<string>	Create
QadCustome	QadCustomer	
pkey	string	
fname	string	
lname	string	
ccode	string	
custadd	QadCustadd[]	

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



General Properties

Detailed Properties

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

Configuration: Default Module Test

Module: TestCreateSPOracle

Component: JDBCOutboundInterface

Interface: JDBCOutboundInterface

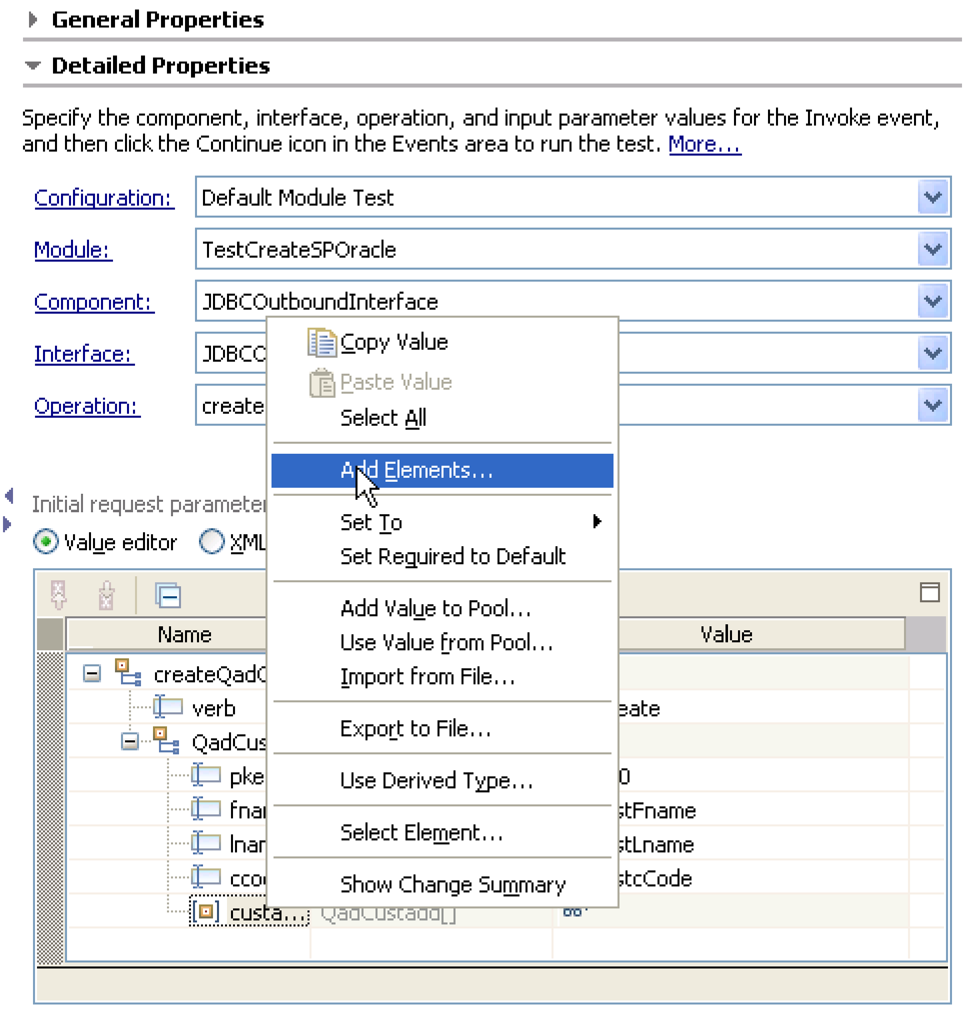
Operation: createQadCustomerBG

Initial request parameters:

● Value editor ○ XML editor

Name	Type	Value
createQadCusto	QadCustomerBG	ab
verb	verb<string>	ab Create
QadCustome	QadCustomer	ab
pkey	string	ab 100
fname	string	ab testFname
lname	string	ab testLname
ccode	string	ab testcCode
custadd	QadCustadd[]	

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



General Properties

Detailed Properties

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

Configuration: Default Module Test

Module: TestCreateSPOracle

Component: JDBCOutboundInterface

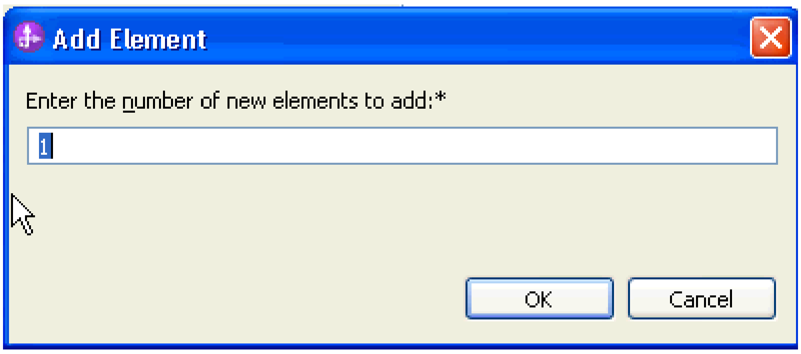
Interface: JDBCC

Operation: create

Initial request paramete

● Value editor ○ XML

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

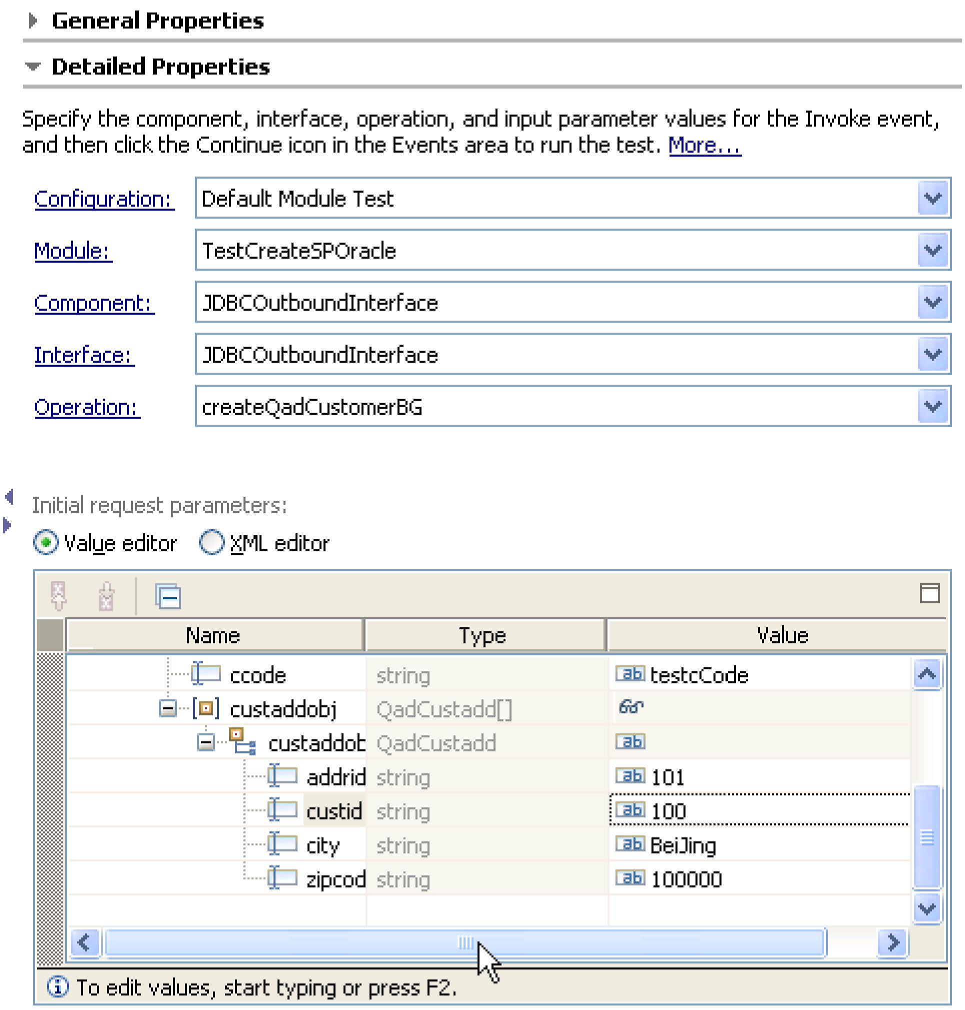


Add Element

Enter the number of new elements to add:*

1

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



Detailed Properties

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

Configuration:	Default Module Test
Module:	TestCreateSPOracle
Component:	JDBCOutboundInterface
Interface:	JDBCOutboundInterface
Operation:	createQadCustomerBG

Initial request parameters:

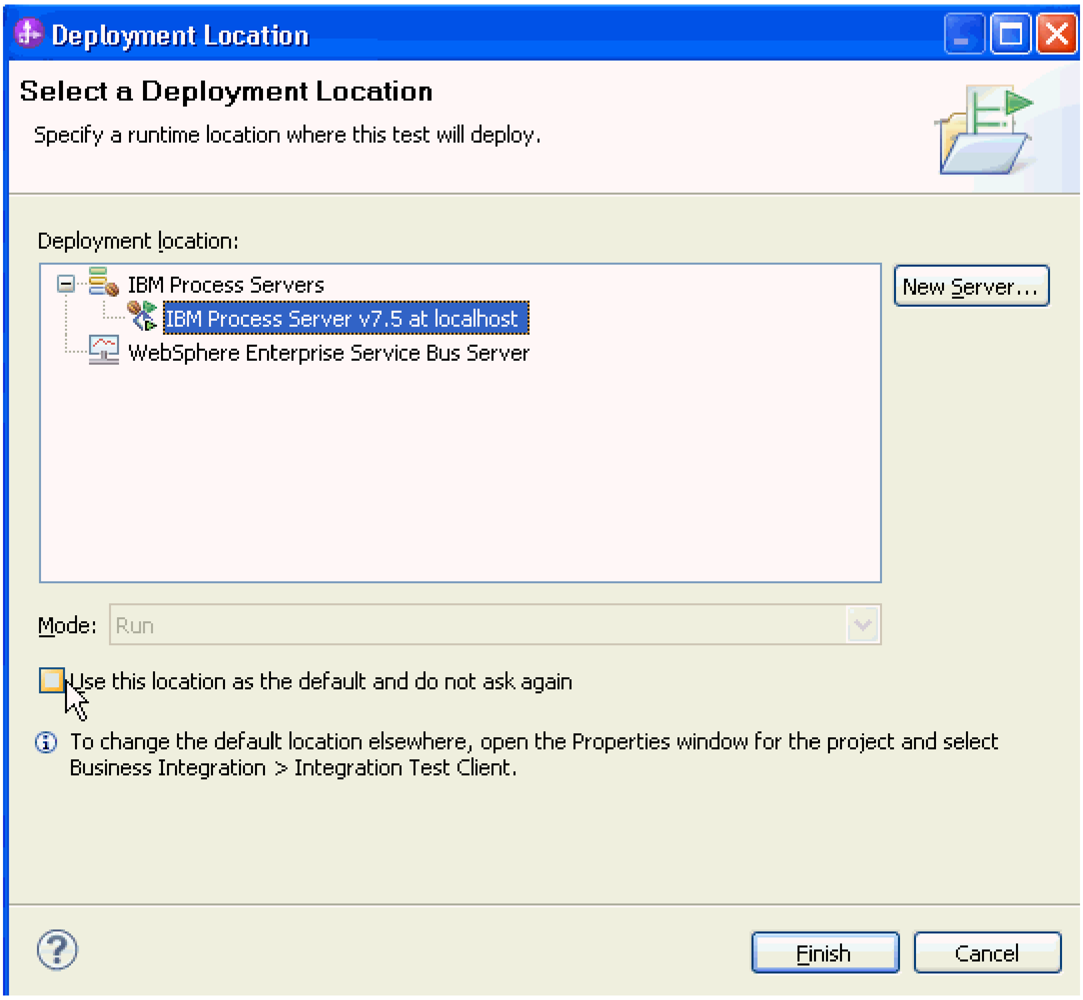
● Value editor ○ XML editor

Name	Type	Value
ccode	string	testcCode
custaddobj	QadCustadd[]	
custaddob	QadCustadd	
addrid	string	101
custid	string	100
city	string	BeiJing
zipcod	string	100000

To edit values, start typing or press F2.

7. To execute the service, click ▶.

8. In the Select a 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:

- IBM Process Servers
  - IBM Process Server v7.5 at localhost
- WebSphere Enterprise Service Bus Server

Mode: Run

☐ Use this location as the default and do not ask again

To change the default location elsewhere, open the Properties window for the project and select Business Integration > Integration Test Client.

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

Clear the sample content

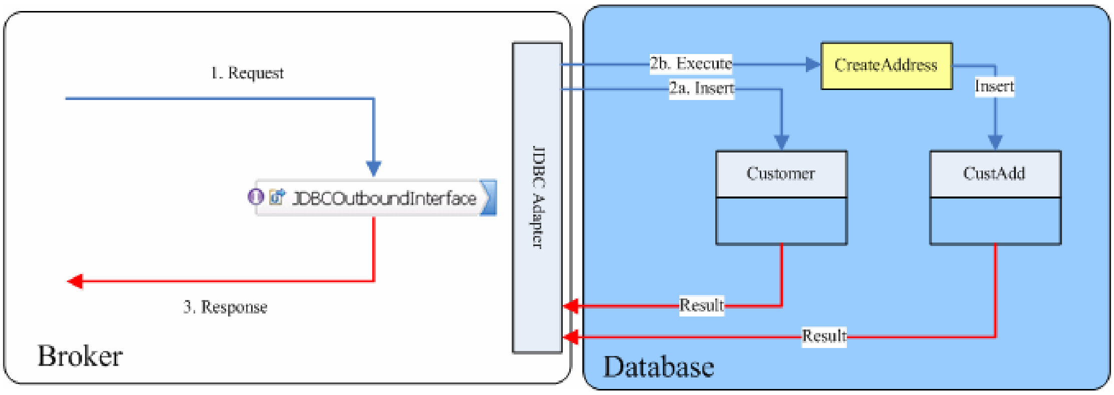
After you have tested the application, clear the sample content to return the data to its original state.

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 7.5.0.0 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

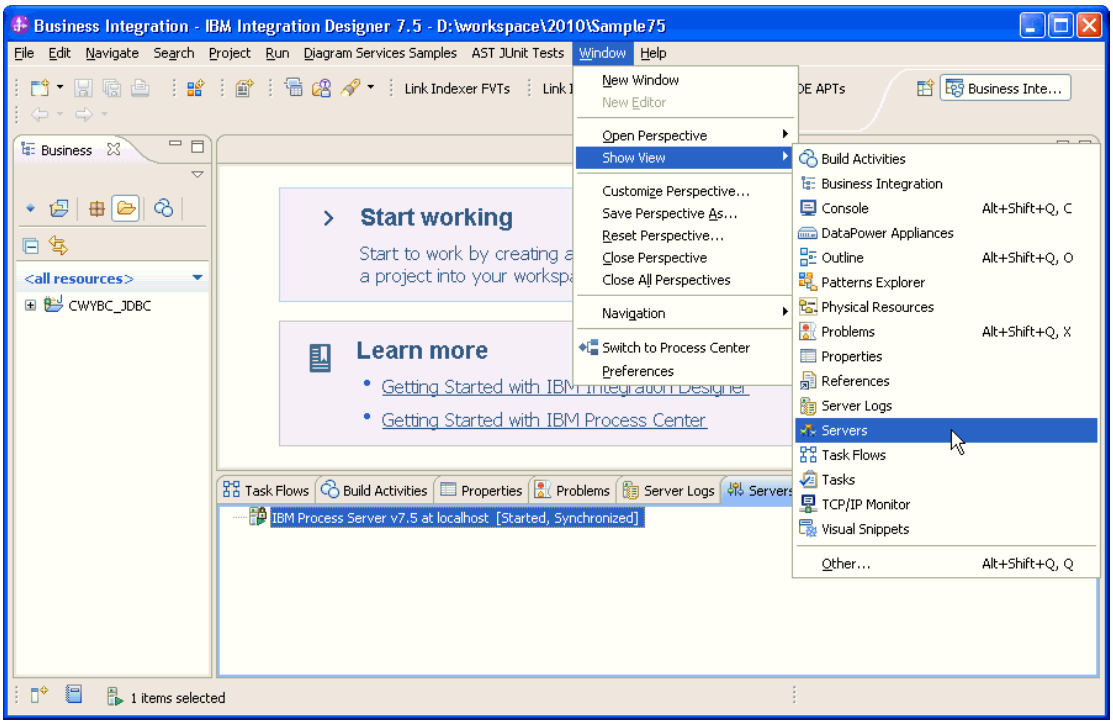
```
CREATE PROCEDURE CREATEADDRESS
(@addr_id varchar(10), @cust_id varchar(10), @city
varchar(10), @zipcode varchar(10))
AS
begin
INSERT into ADDRESS (ADDRID, CUSTID, CITY, ZIPCODE)
values
             (@addr_id, @cust_id, @city, @zipcode);
end;
```

Create an authentication alias

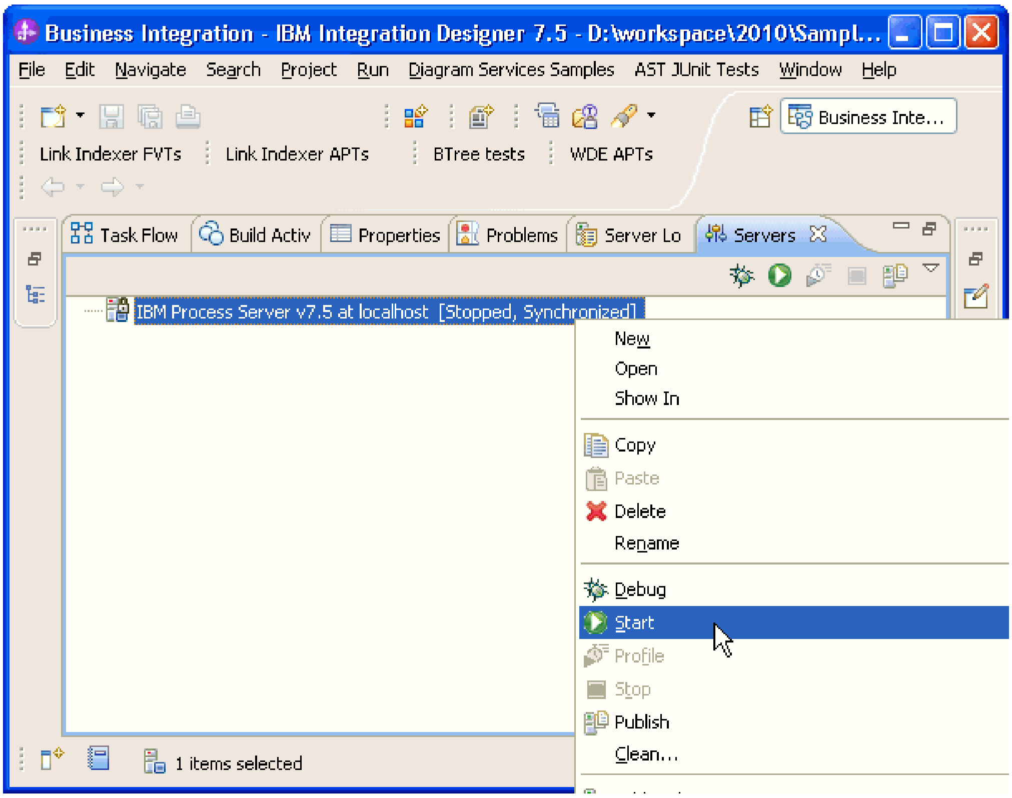
The authentication alias needs to be set because the data source that is used to generate artifacts will use the username and password set in the authentication alias to connect to the database.

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

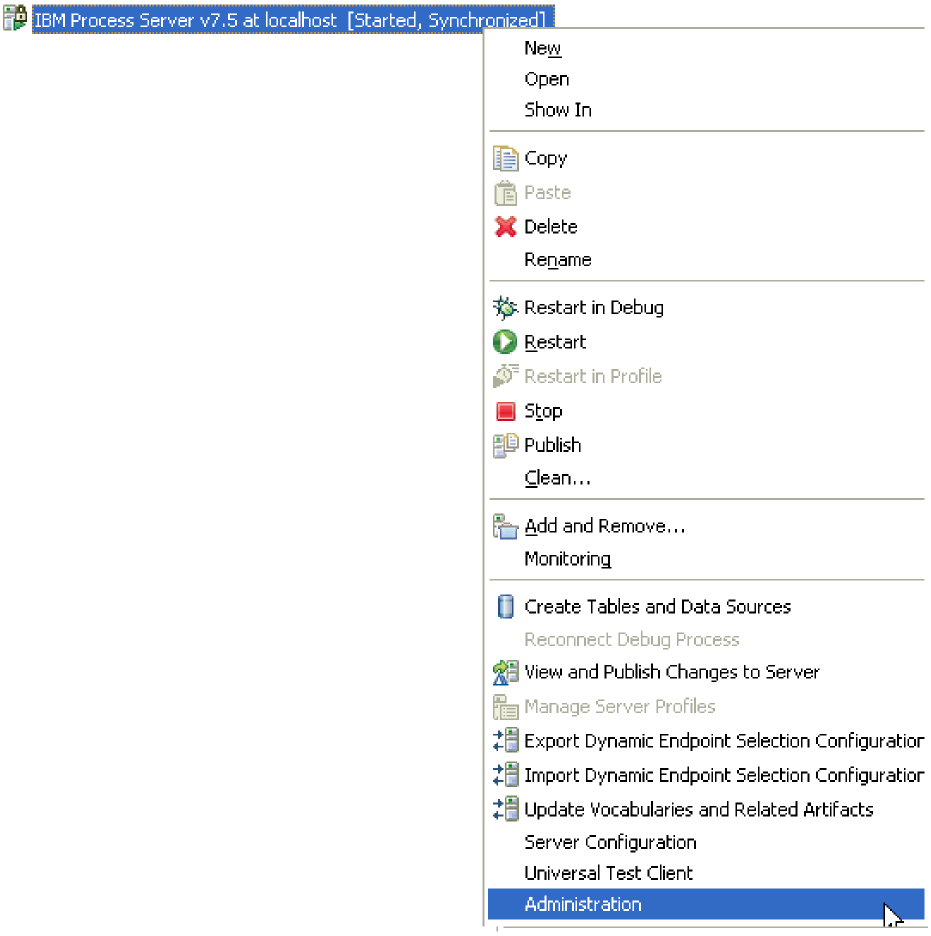
1. In IBM Integration Designer, switch to the **Servers** view by selecting **Windows > Show View > Servers**.



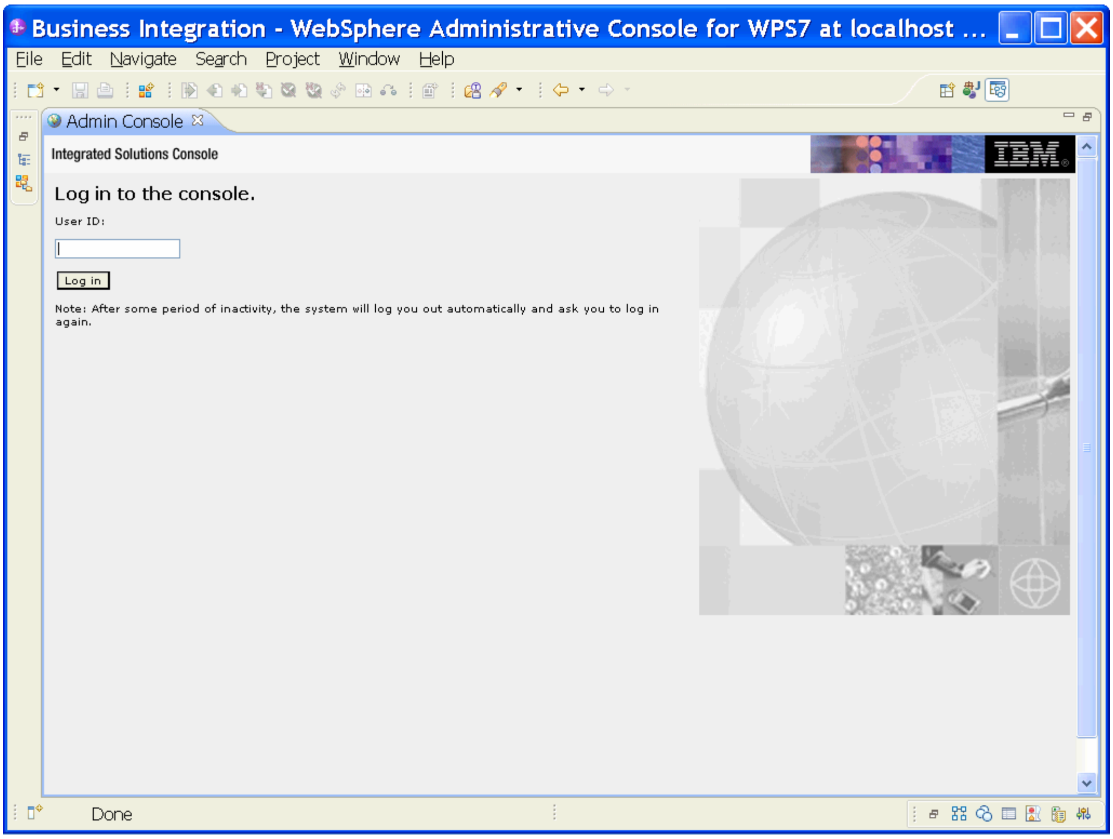
2. In the **Servers** view, right-click the server that you want to start and select **Start**.


Business Integration - IBM Integration Designer 7.5 - D:\workspace\2010\Sampl...
File Edit Navigate Search Project Run Diagram Services Samples AST JUnit Tests Window Help
Task Flow Build Activ Properties Problems Server Lo Servers
IBM Process Server v7.5 at localhost [Stopped, Synchronized]
New
Open
Show In
Copy
Paste
Delete
Rename
Debug
Start
Profile
Stop
Publish
Clean...
1 items selected

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


IBM Process Server v7.5 at localhost [Started, Synchronized]
New
Open
Show In
Copy
Paste
Delete
Rename
Restart in Debug
Restart
Restart in Profile
Stop
Publish
Clean...
Add and Remove...
Monitoring
Create Tables and Data Sources
Reconnect Debug Process
View and Publish Changes to Server
Manage Server Profiles
Export Dynamic Endpoint Selection Configuratior
Import Dynamic Endpoint Selection Configuratior
Update Vocabularies and Related Artifacts
Server Configuration
Universal Test Client
Administration

4. Log on to the administrative console.


Business Integration - WebSphere Administrative Console for WPS7 at localhost ...
File
Edit
Navigate
Search
Project
Window
Help
Admin Console
Integrated Solutions Console
IBM
Done

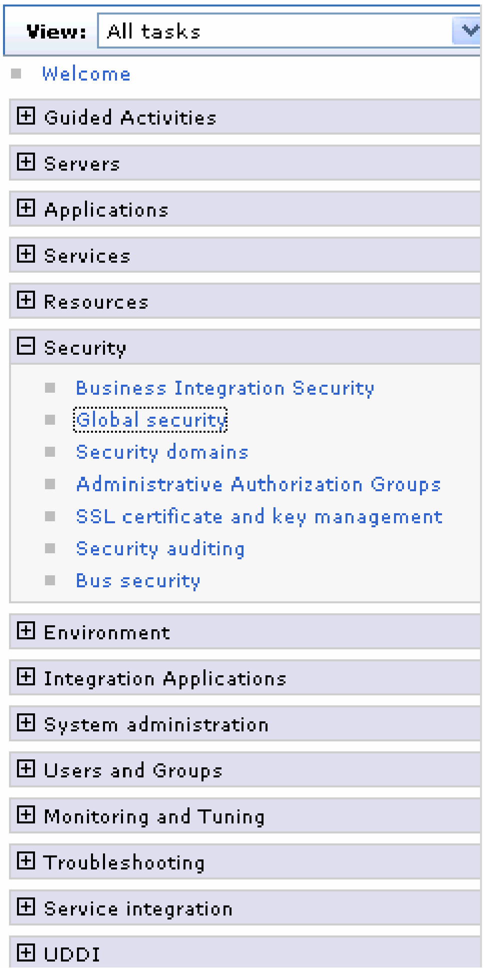
Log in to the console.

User ID:

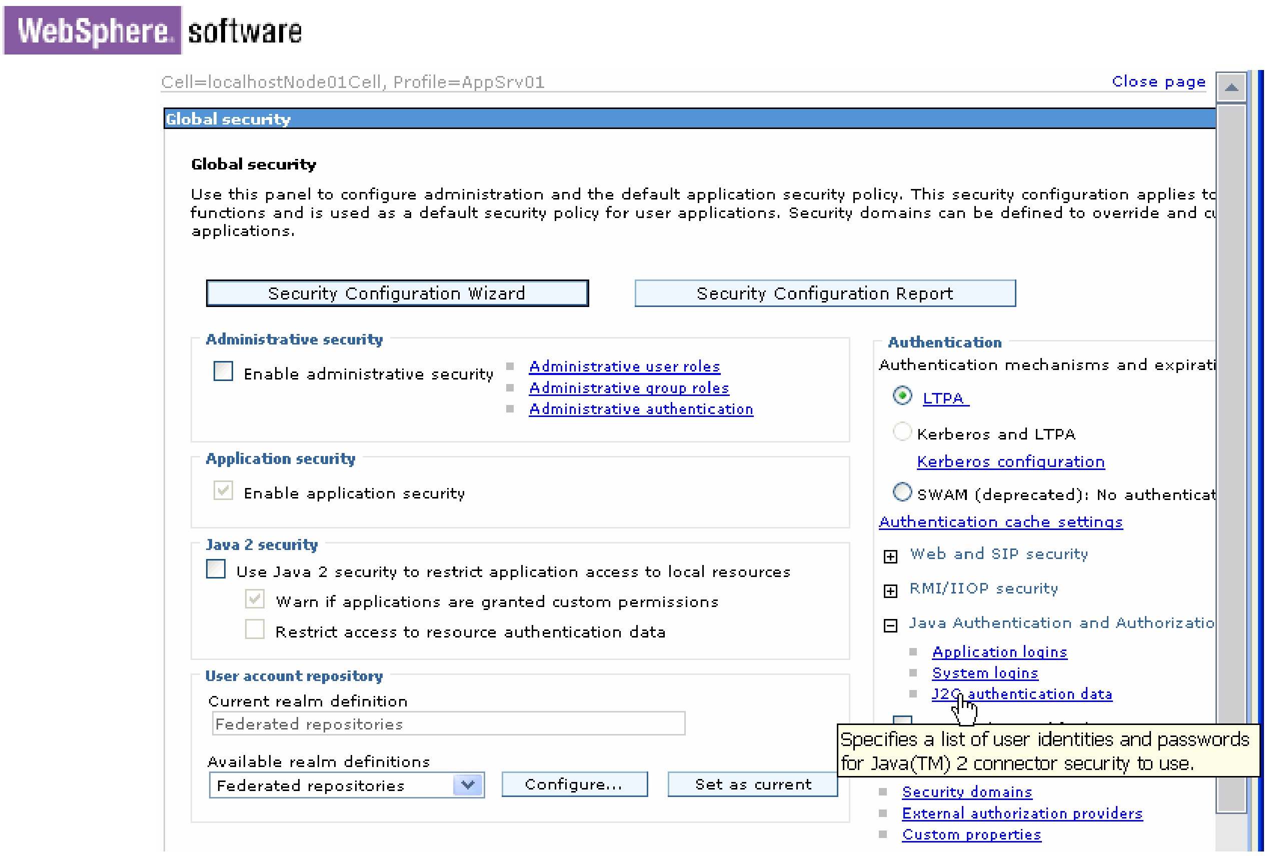
Log in

Note: After some period of inactivity, the system will log you out automatically and ask you to log in again.

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



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



Cell=localhostNode01Cell, Profile=AppSrv01

Global security

Global security

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

● LTPA

○ Kerberos and LTPA

Kerberos configuration

○ SWAM (deprecated): No authenticat

Authentication cache settings

⊞ Web and SIP security

⊞ RMI/IIOP security

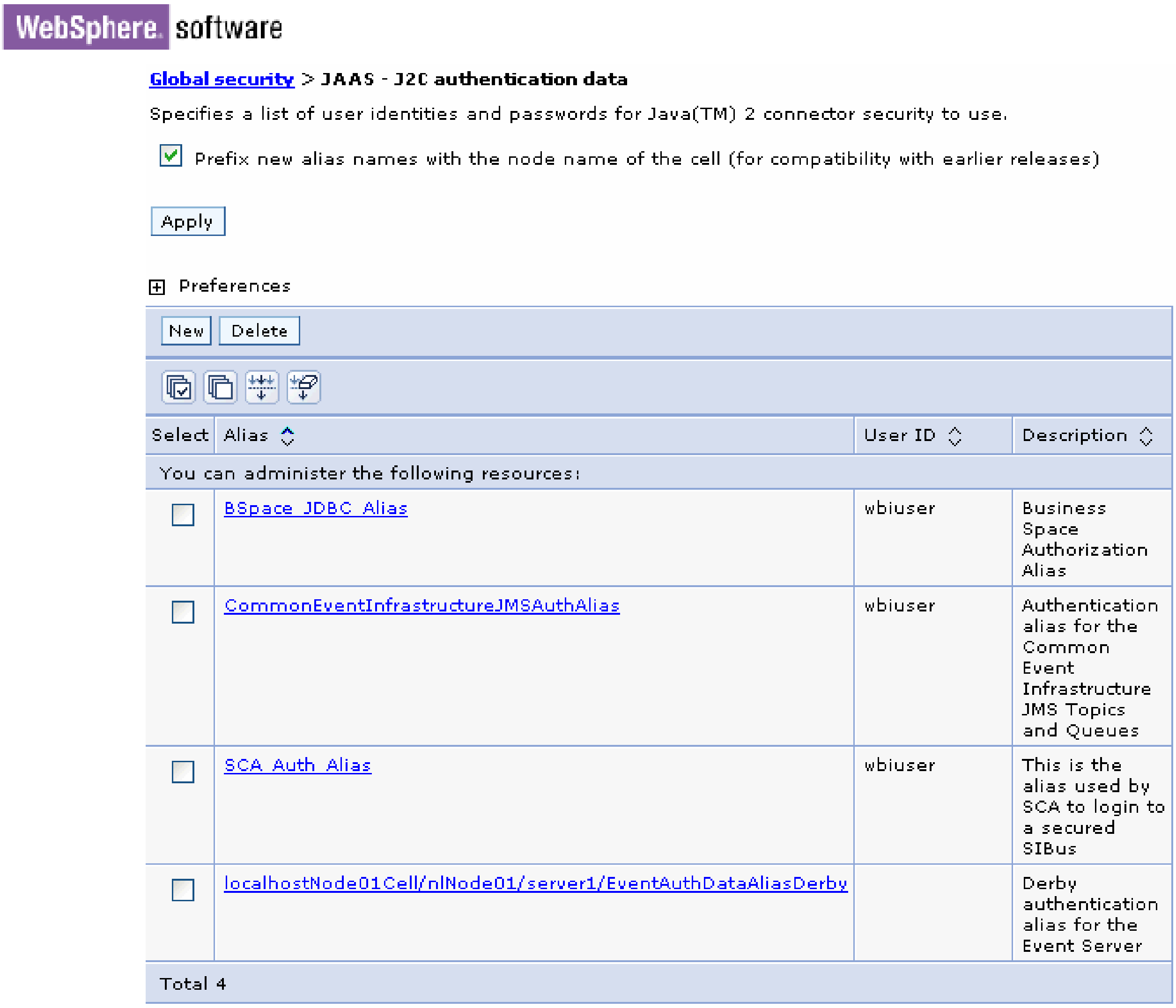
⊟ Java Authentication and Authorizatio

- 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

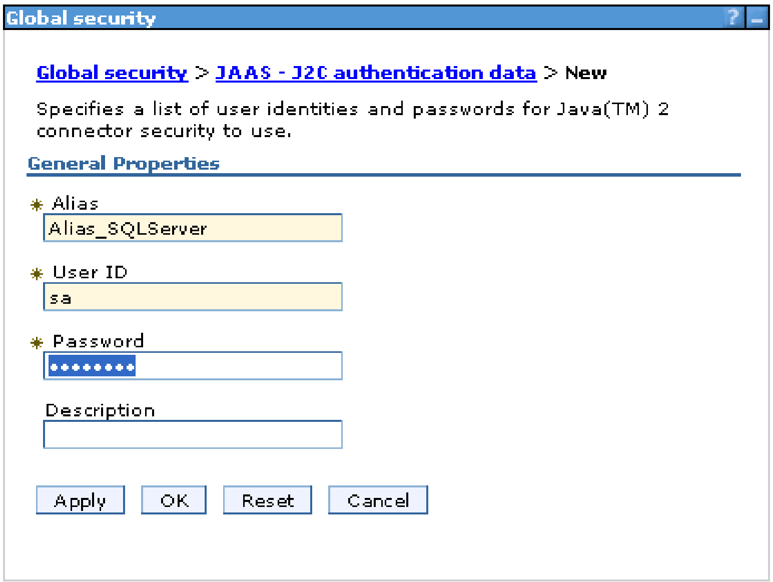
Global security > JAAS - J2C authentication data

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)

Select	Alias	User ID	Description
You can administer the following resources:			
□	BSpace_JDBC_Alias	wbiuser	Business Space Authorization Alias
□	CommonEventInfrastructureJMSAuthAlias	wbiuser	Authentication alias for the Common Event Infrastructure JMS Topics and Queues
□	SCA_Auth_Alias	wbiuser	This is the alias used by SCA to login to a secured SIBus
□	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**.



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

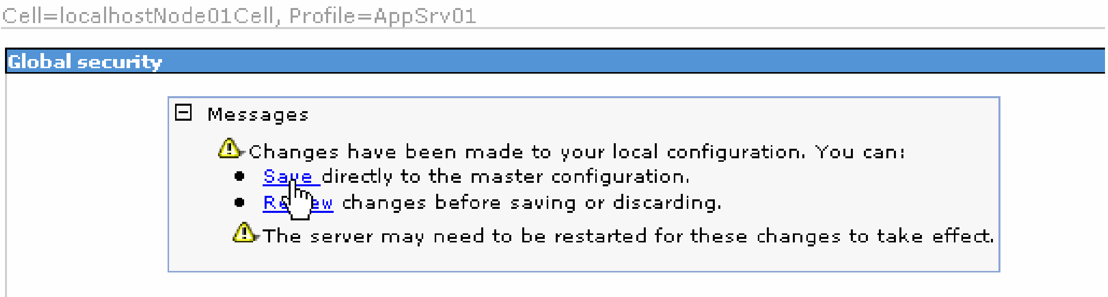
* Alias: Alias_SQLServer

* User ID: sa

* Password: ••••••••

Description:

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



Cell=localhostNode01Cell, Profile=AppSrv01

Global security

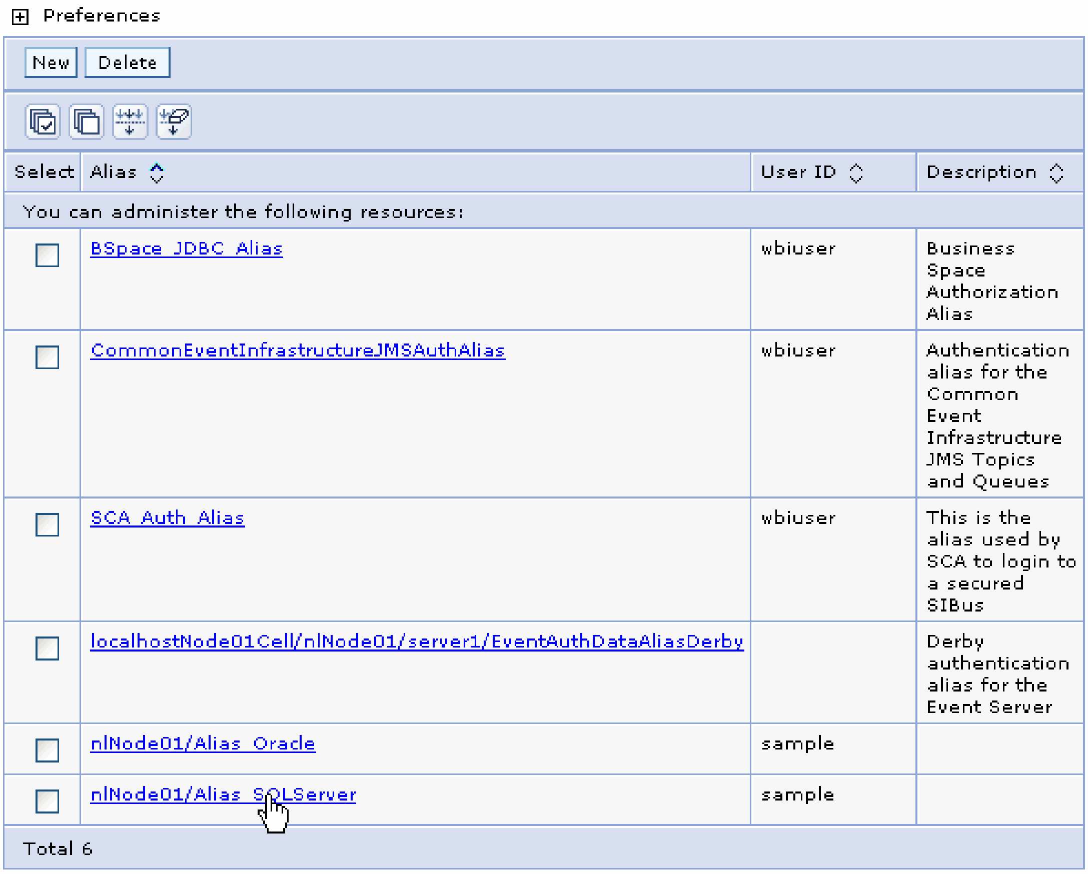
⊟ Messages

⚠ 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.

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:			
□	BSpace_JDBC_Alias	wbiuser	Business Space Authorization Alias
□	CommonEventInfrastructureJMSAuthAlias	wbiuser	Authentication alias for the Common Event Infrastructure JMS Topics and Queues
□	SCA_Auth_Alias	wbiuser	This is the alias used by SCA to login to a secured SIBus
□	localhostNode01Cell/nlNode01/server1/EventAuthDataAliasDerby		Derby authentication alias for the Event Server
□	nlNode01/Alias_Oracle	sample	
□	nlNode01/Alias_SQLServer	sample	
Total 6			

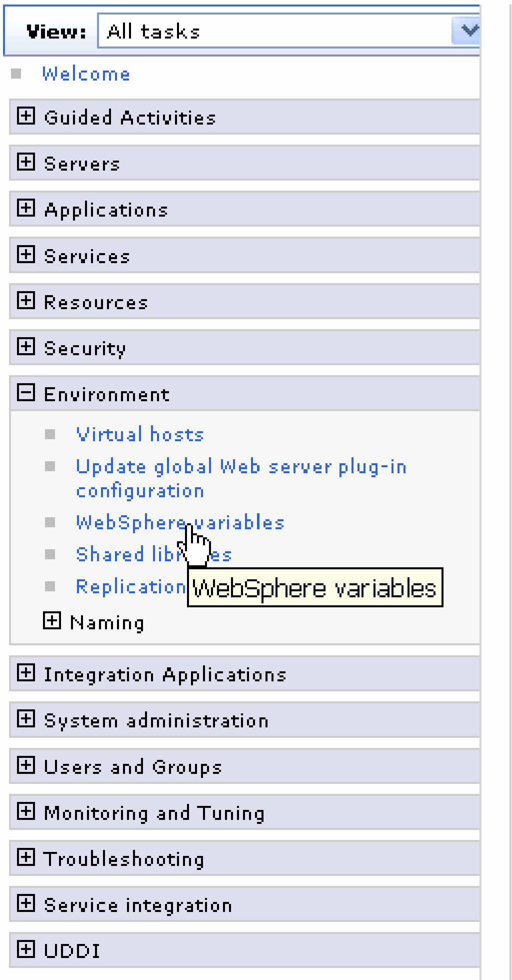
Create a data source

Create a data source in IBM 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.

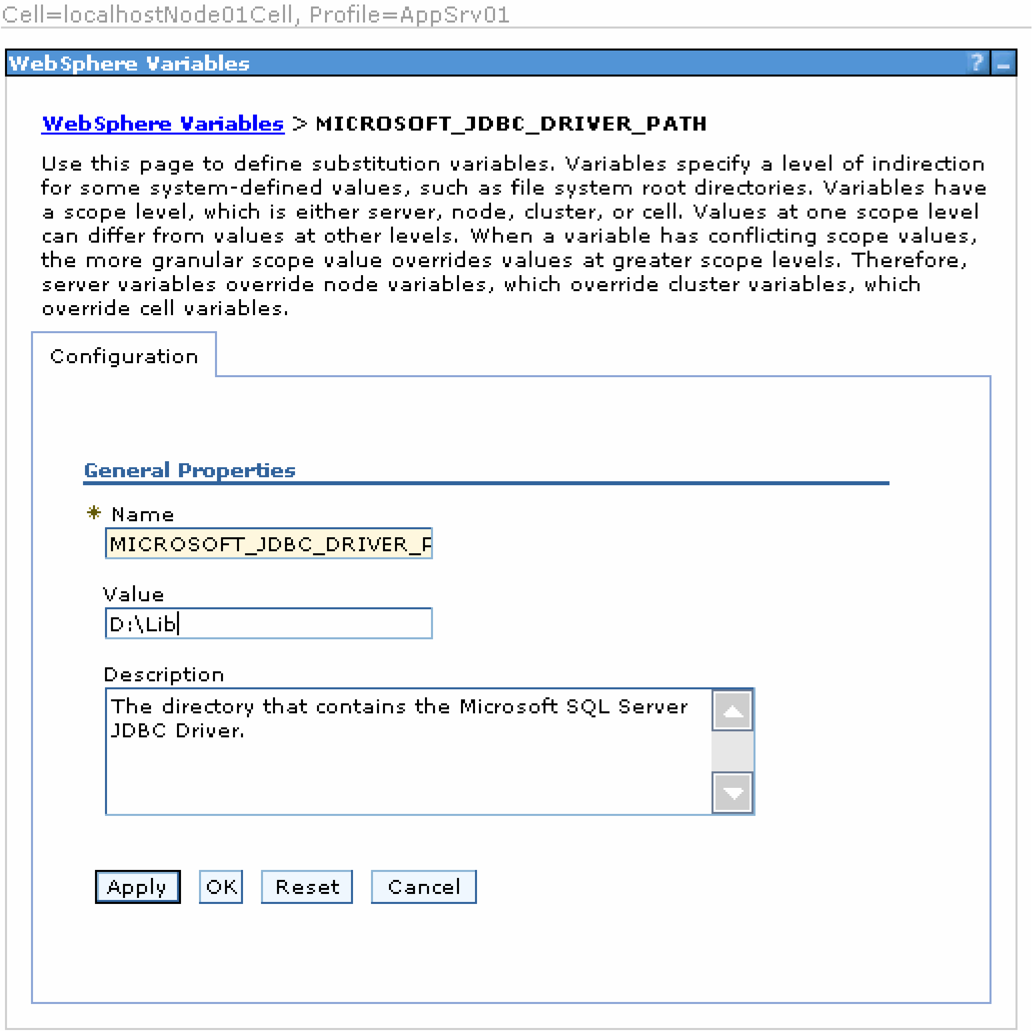
Note: This tutorial uses SQL Server as the database and the SQL Server JDBC driver sqljdbc.jar.

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

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



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



Cell=localhostNode01Cell, Profile=AppSrv01

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.

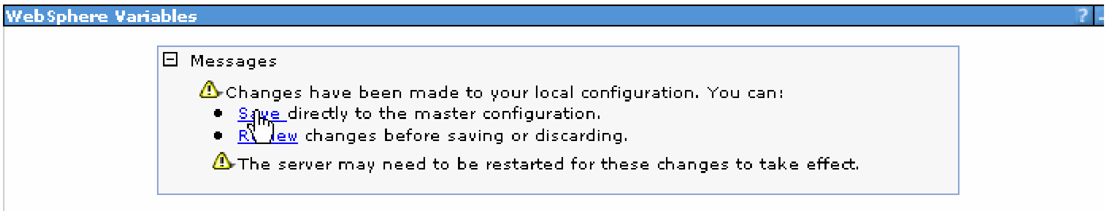
General Properties

* Name
MICROSOFT_JDBC_DRIVER_P

Value
D:\Lib

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

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



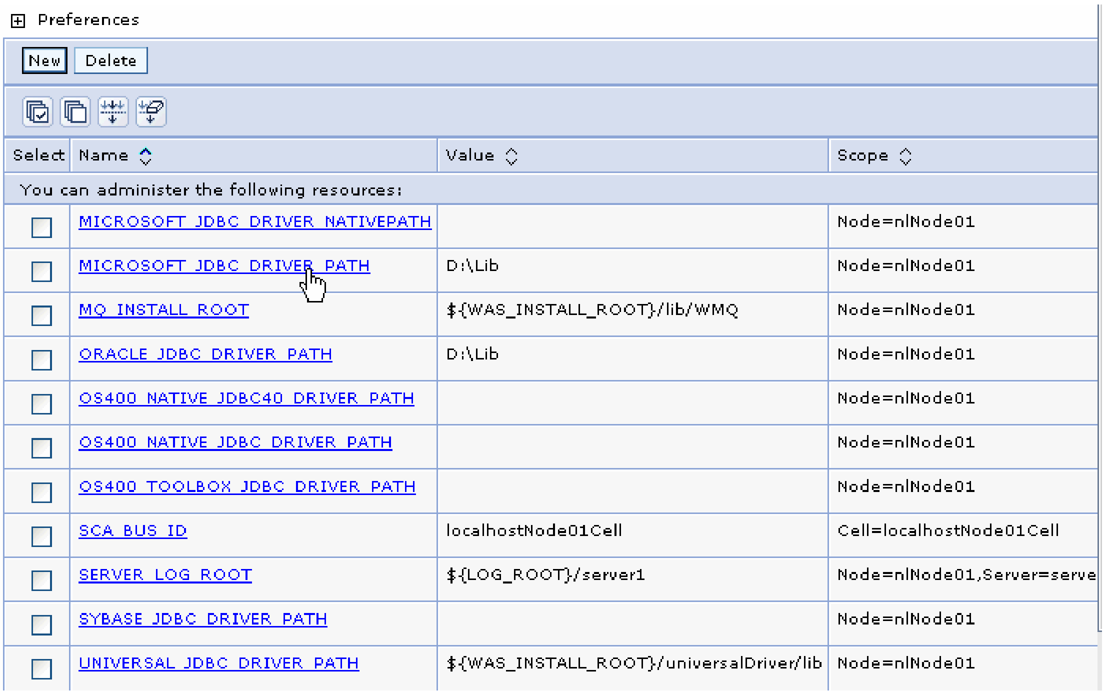
Messages

⚠ 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.

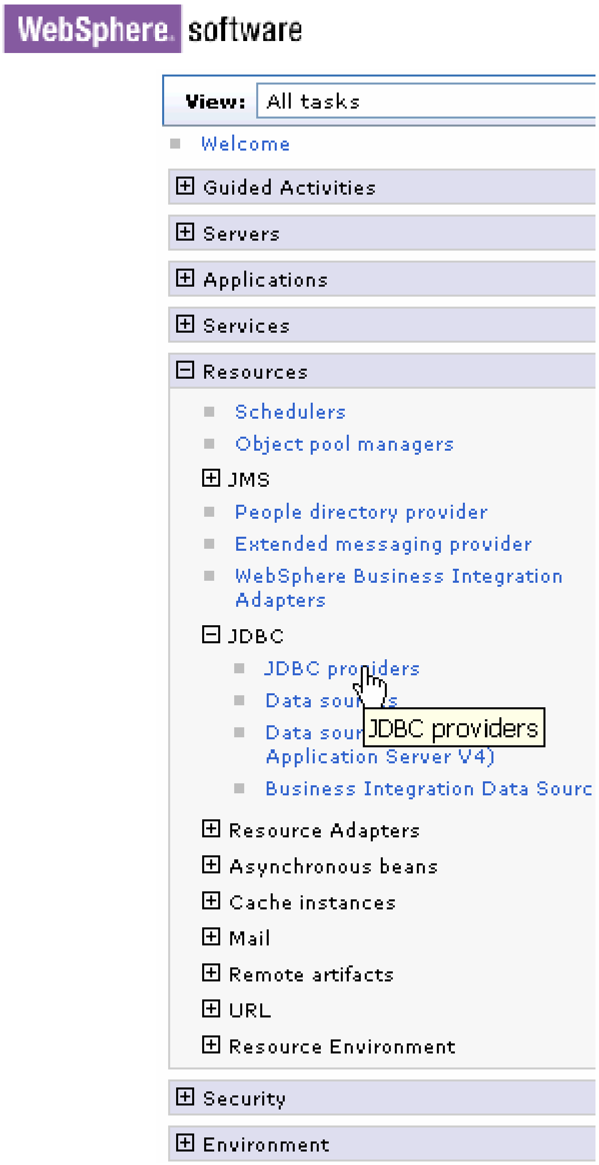


⊞ Preferences

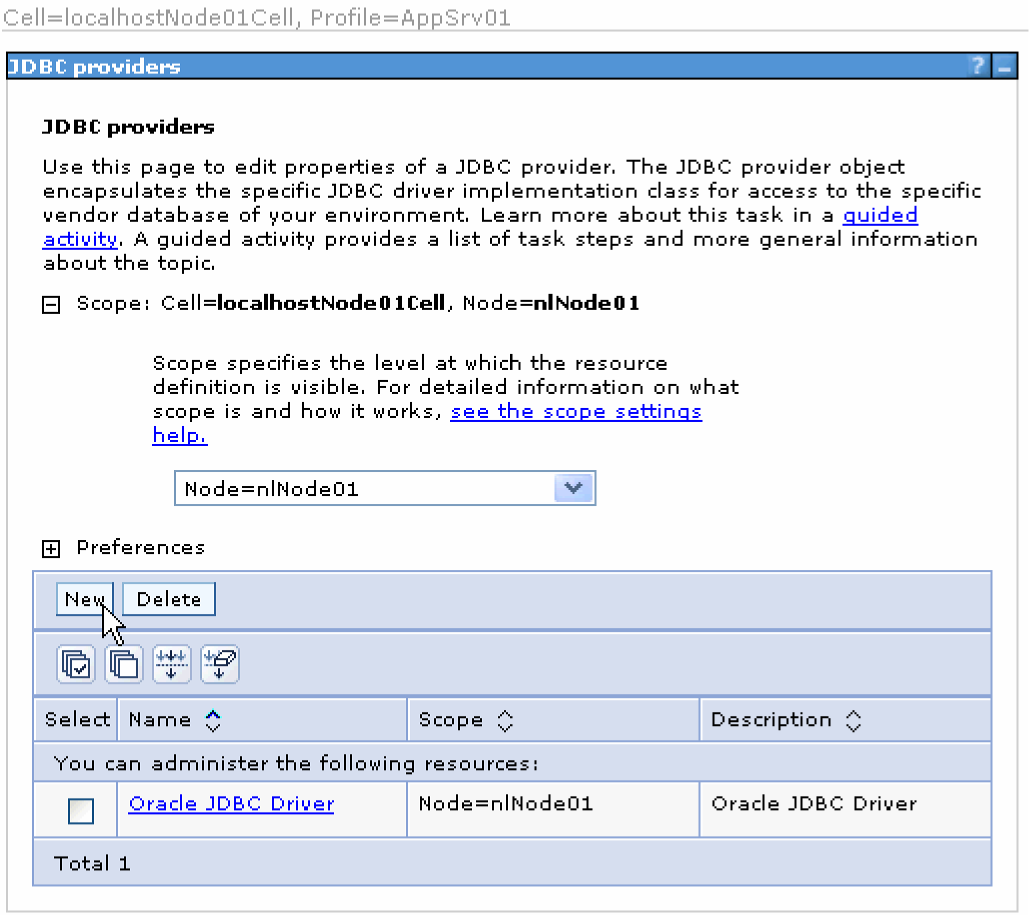
New Delete

Select	Name	Value	Scope
You can administer the following resources:			
□	MICROSOFT_JDBC_DRIVER_NATIVEPATH		Node=nlNode01
□	MICROSOFT_JDBC_DRIVER_PATH	D:\Lib	Node=nlNode01
□	MQ_INSTALL_ROOT	${WAS_INSTALL_ROOT}/lib/WMQ	Node=nlNode01
□	ORACLE_JDBC_DRIVER_PATH	D:\Lib	Node=nlNode01
□	OS400_NATIVE_JDBC40_DRIVER_PATH		Node=nlNode01
□	OS400_NATIVE_JDBC_DRIVER_PATH		Node=nlNode01
□	OS400_TOOLBOX_JDBC_DRIVER_PATH		Node=nlNode01
□	SCA_BUS_ID	localhostNode01Cell	Cell=localhostNode01Cell
□	SERVER_LOG_ROOT	${LOG_ROOT}/server1	Node=nlNode01,Server=serve
□	SYBASE_JDBC_DRIVER_PATH		Node=nlNode01
□	UNIVERSAL_JDBC_DRIVER_PATH	${WAS_INSTALL_ROOT}/universalDriver/lib	Node=nlNode01

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



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



Cell=localhostNode01Cell, Profile=AppSrv01

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=**nlNode01**

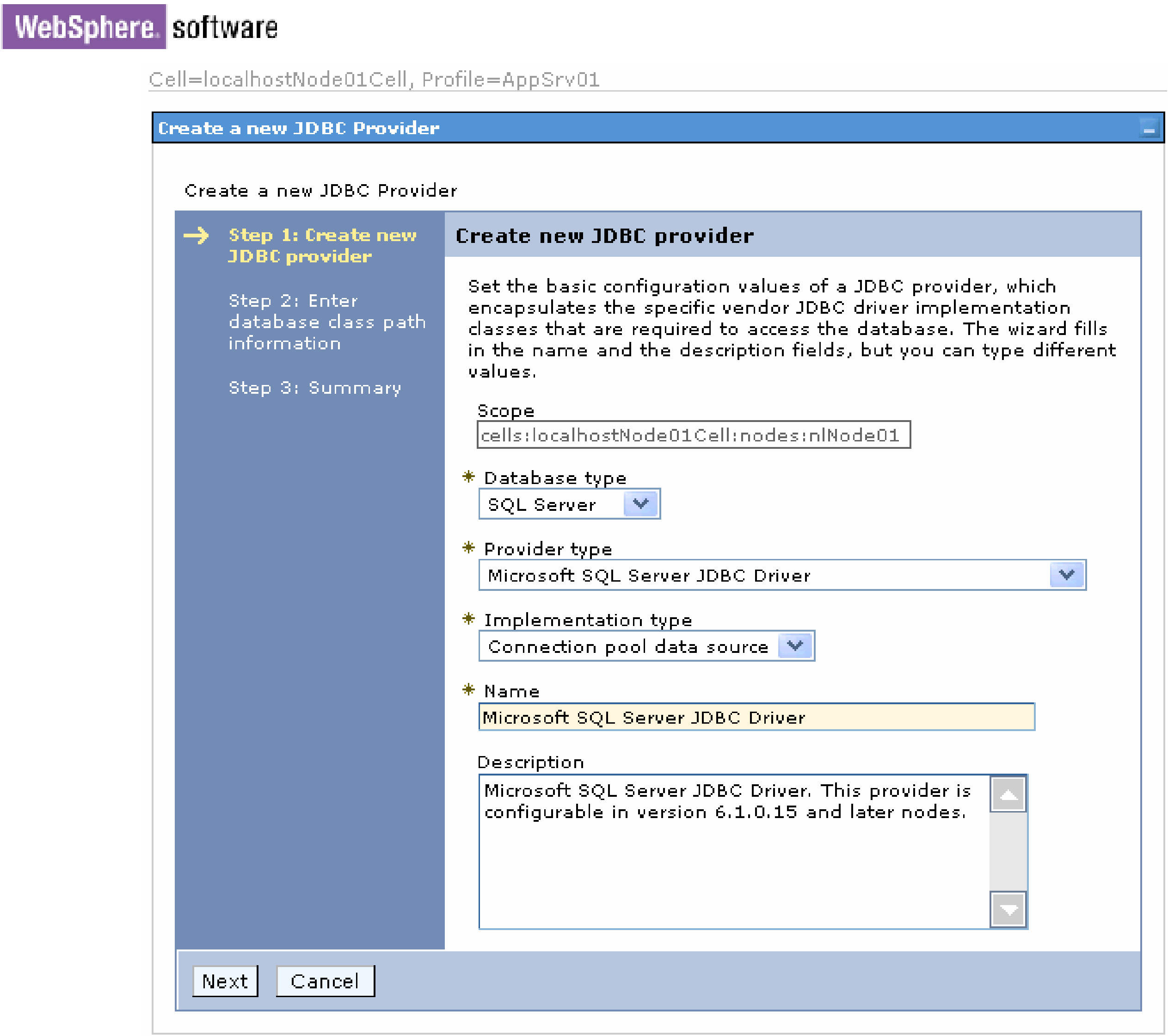
Scope specifies the level at which the resource definition is visible. For detailed information on what scope is and how it works, see the scope settings help.

Node=nlNode01

⊞ Preferences

Select	Name	Scope	Description
You can administer the following resources:			
□	Oracle JDBC Driver	Node=nlNode01	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**.



Cell=localhostNode01Cell, Profile=AppSrv01

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:nlNode01

* Database type
SQL Server

* Provider type
Microsoft SQL Server JDBC Driver

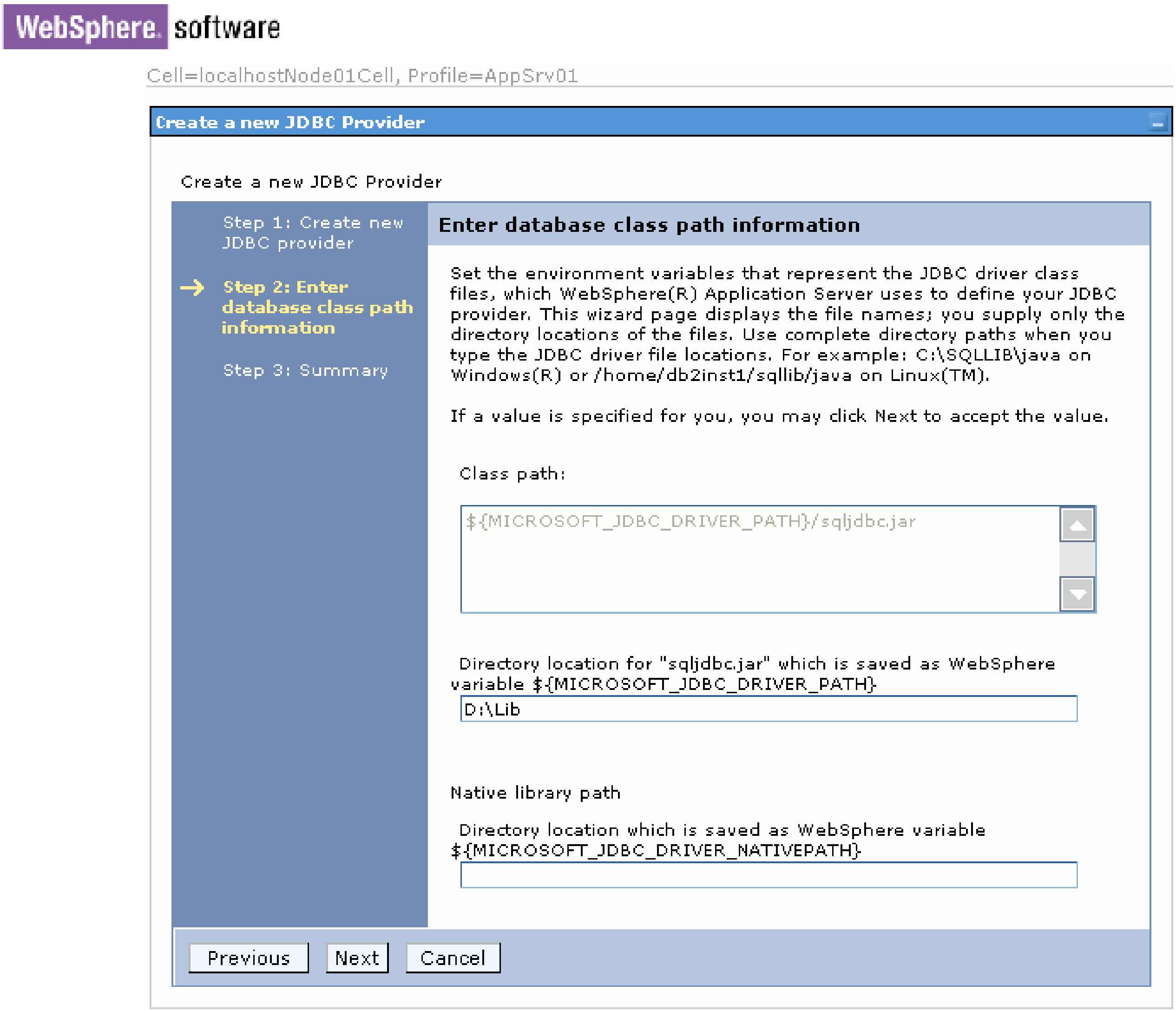
* Implementation type
Connection pool data source

* 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.

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

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

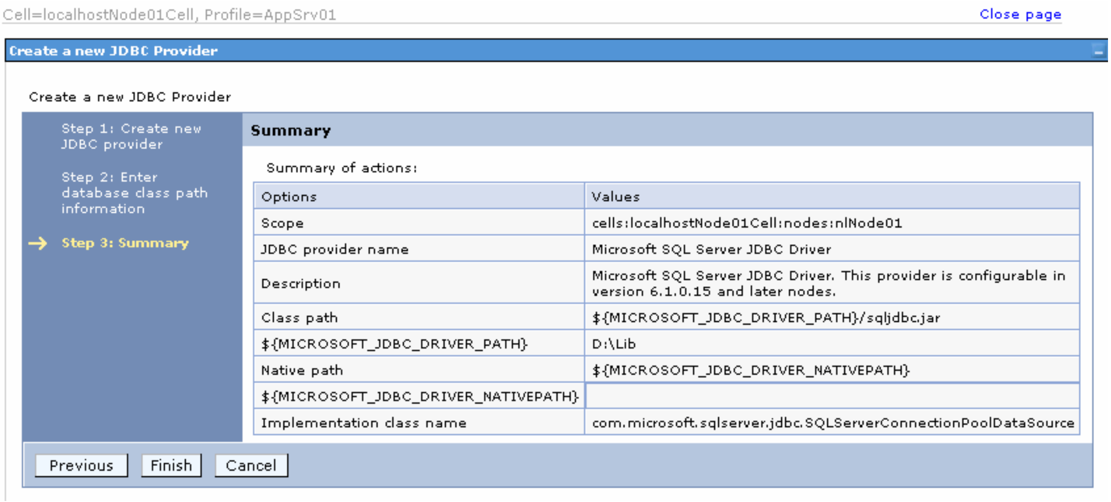
Directory location for "sqljdbc.jar" which is saved as WebSphere variable ${MICROSOFT_JDBC_DRIVER_PATH}

D:\Lib

Native library path

Directory location which is saved as WebSphere variable ${MICROSOFT_JDBC_DRIVER_NATIVEPATH}

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



Cell=localhostNode01Cell, Profile=AppSrv01
Close page
Create a new JDBC Provider

Create a new JDBC Provider

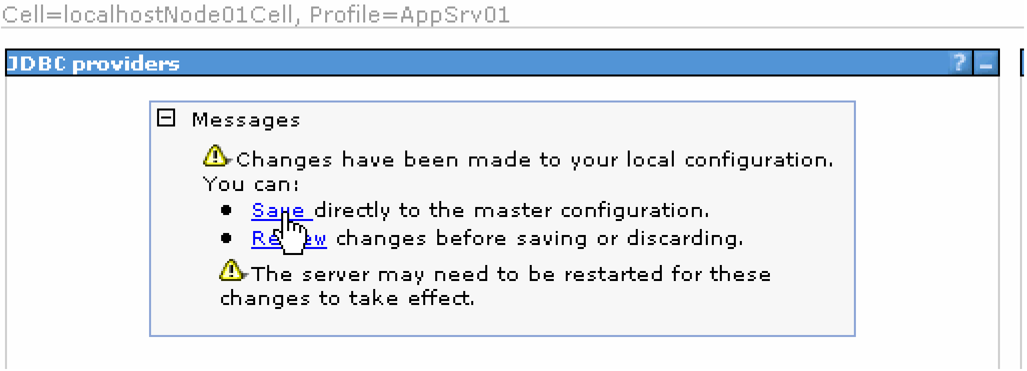
Step 1: Create new JDBC provider
Step 2: Enter database class path information
→ Step 3: Summary

Summary

Summary of actions:

Options	Values
Scope	cells:localhostNode01Cell:nodes:nlNode01
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	${MICROSOFT_JDBC_DRIVER_PATH}/sqljdbc.jar
${MICROSOFT_JDBC_DRIVER_PATH}	D:\Lib
Native path	${MICROSOFT_JDBC_DRIVER_NATIVEPATH}
${MICROSOFT_JDBC_DRIVER_NATIVEPATH}	
Implementation class name	com.microsoft.sqlserver.jdbc.SQLServerConnectionPoolDataSource

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



Cell=localhostNode01Cell, Profile=AppSrv01

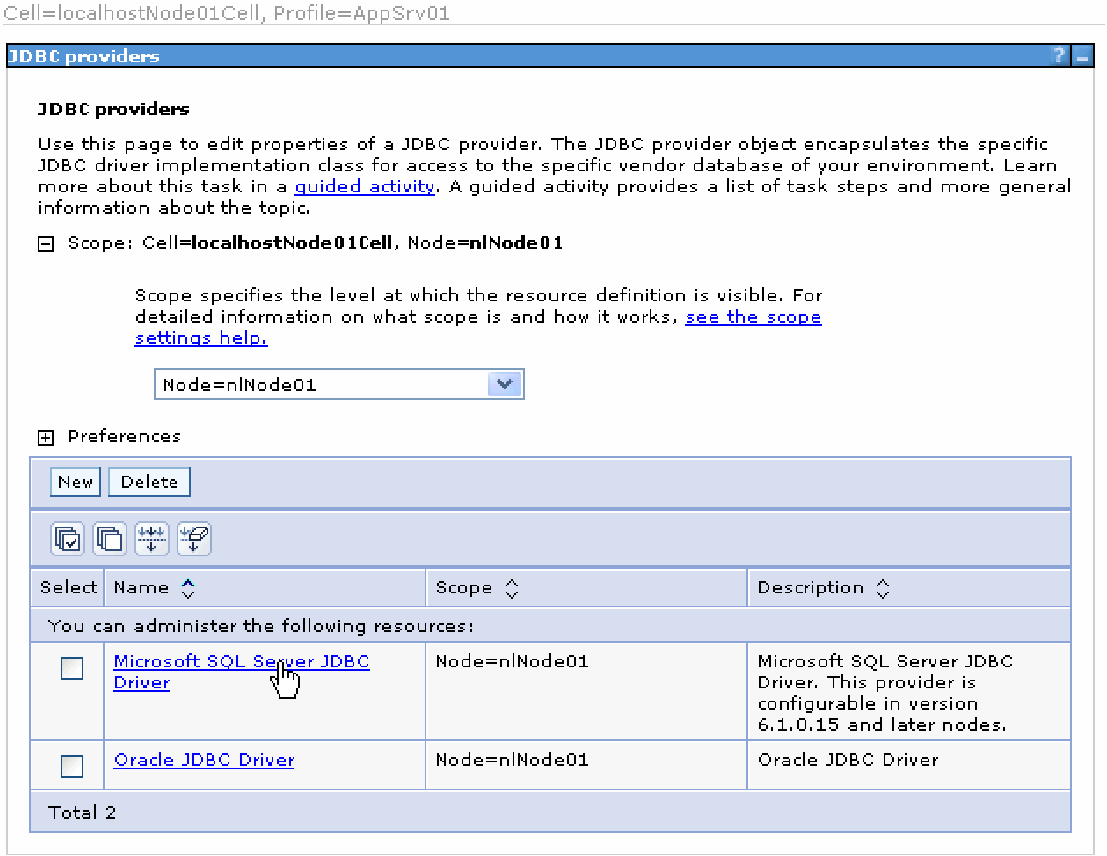
Messages

⚠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=**nlNode01**

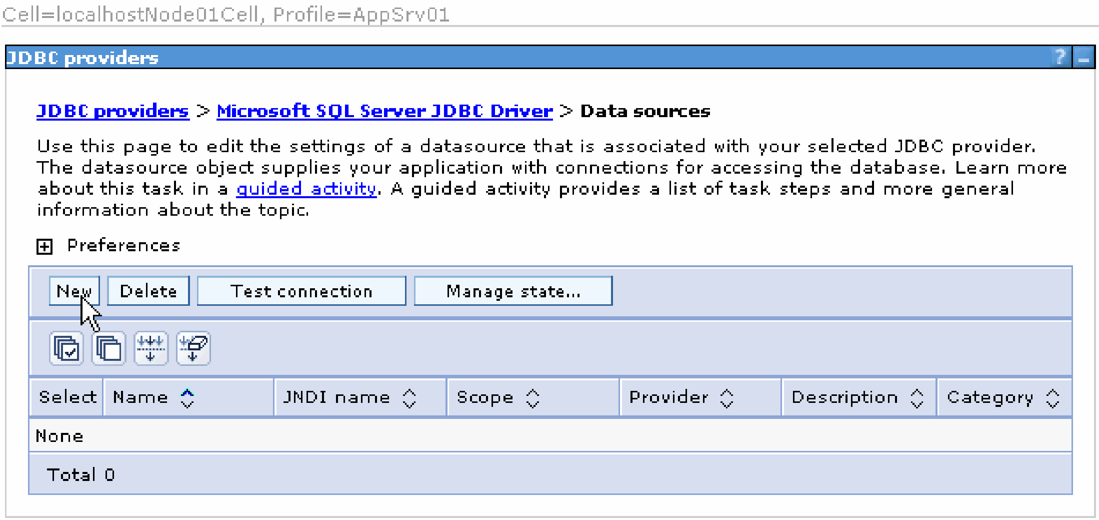
Scope specifies the level at which the resource definition is visible. For detailed information on what scope is and how it works, see the scope settings help.

Node=nlNode01

⊞ Preferences

Select	Name	Scope	Description
You can administer the following resources:			
□	Microsoft SQL Server JDBC Driver	Node=nlNode01	Microsoft SQL Server JDBC Driver. This provider is configurable in version 6.1.0.15 and later nodes.
□	Oracle JDBC Driver	Node=nlNode01	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

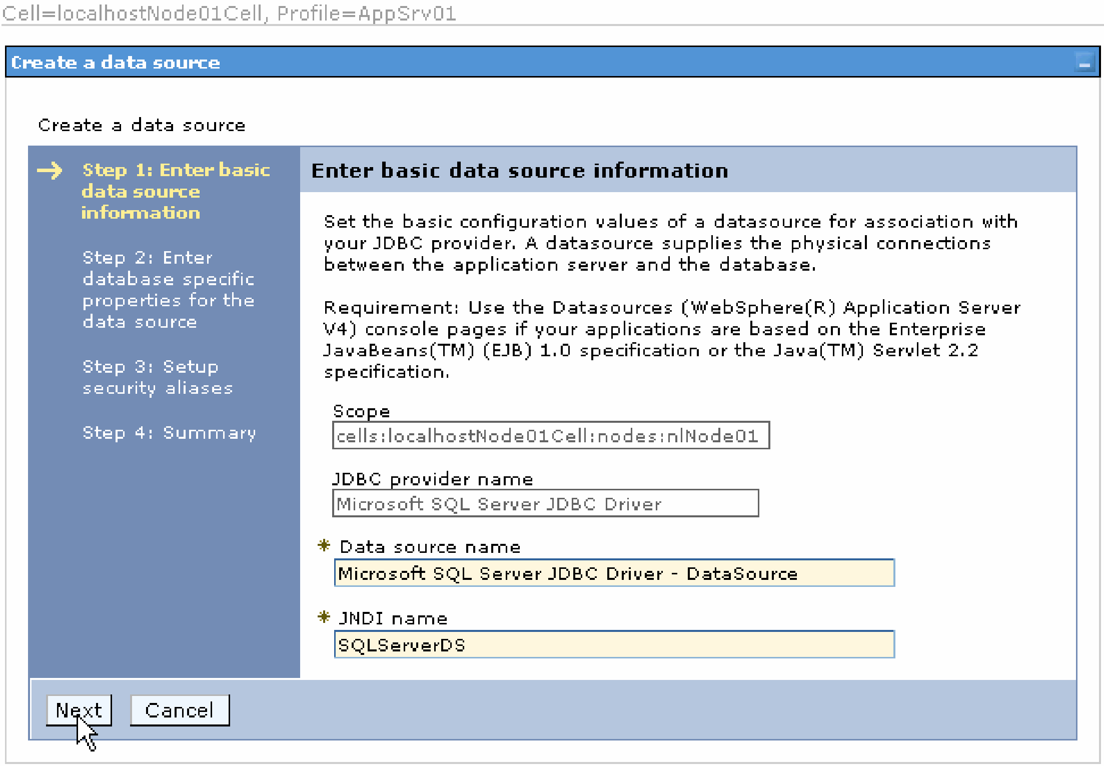
JDBC providers > Microsoft SQL Server JDBC Driver > Data sources

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

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.

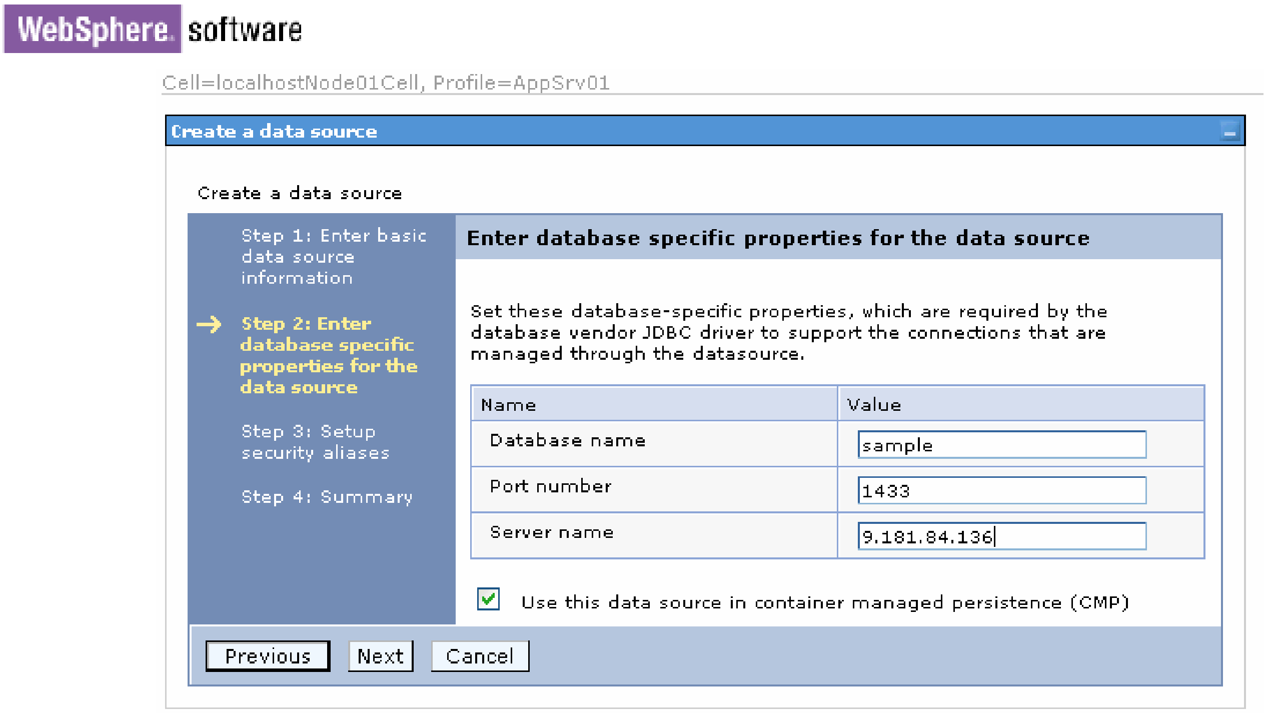
Scope
cells:localhostNode01Cell:nodes:nlNode01

JDBC provider name
Microsoft SQL Server JDBC Driver

* Data source name
Microsoft SQL Server JDBC Driver - DataSource

* JNDI name
SQLServerDS

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

Create a data source

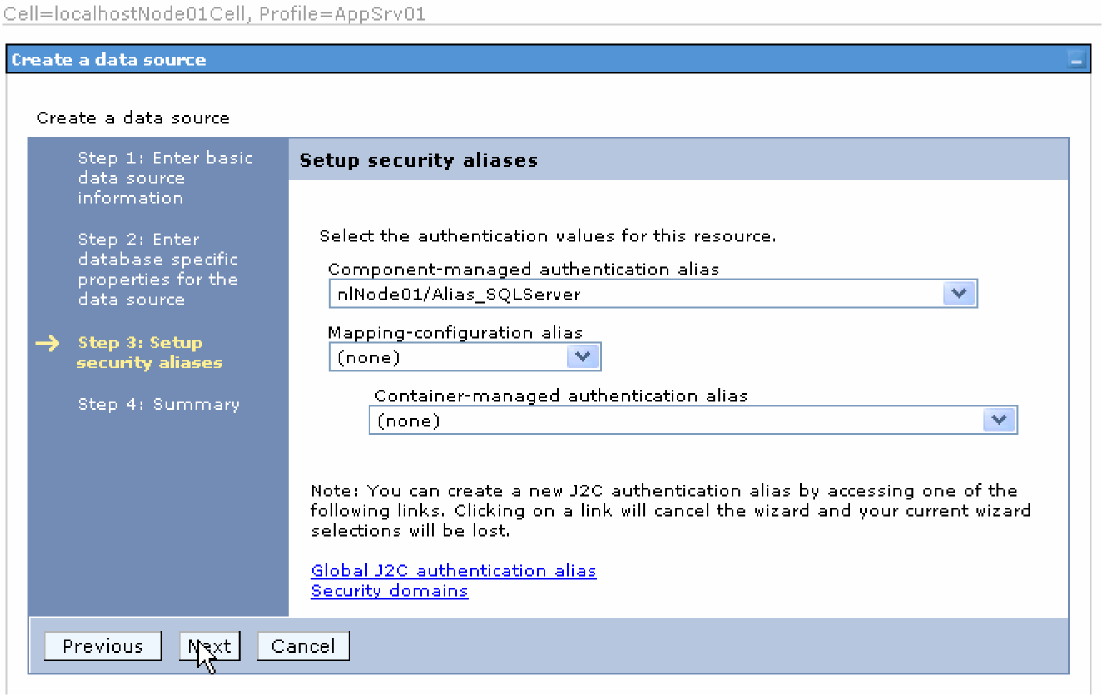
Enter database specific properties for the data source

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
Database name	sample
Port number	1433
Server name	9.181.84.136

☑ Use this data source in container managed persistence (CMP)

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
Previous
Next
Cancel

Create a data source

Setup security aliases

Select the authentication values for this resource.

Component-managed authentication alias: nlNode01/Alias_SQLServer

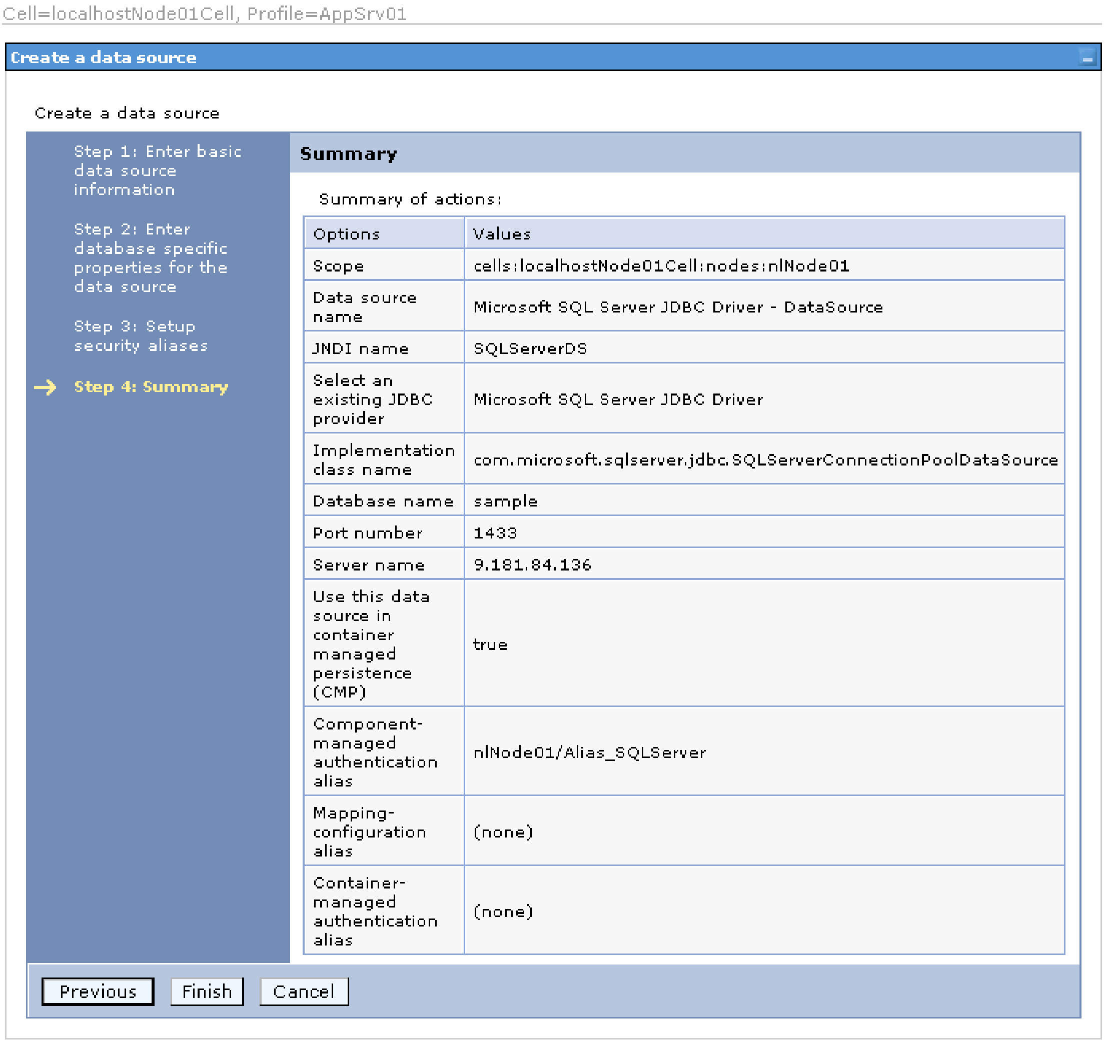
Mapping-configuration alias: (none)

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

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



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

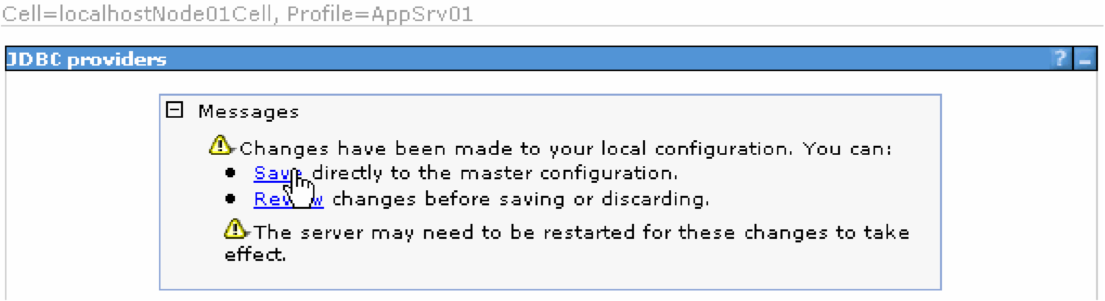
→ **Step 4: Summary**

Summary

Summary of actions:

Options	Values
Scope	cells:localhostNode01Cell:nodes:nlNode01
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	nlNode01/Alias_SQLServer
Mapping-configuration alias	(none)
Container-managed authentication alias	(none)

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



Cell=localhostNode01Cell, Profile=AppSrv01

JDBC providers

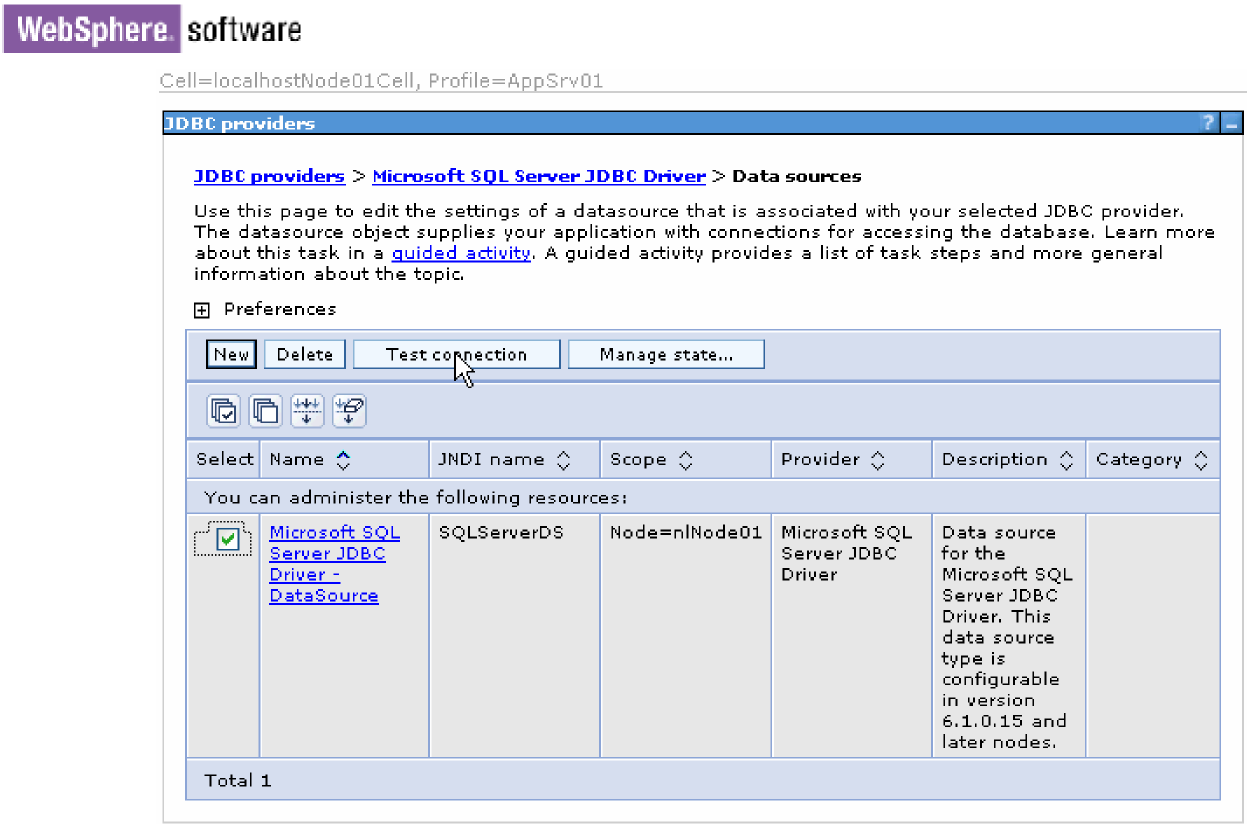
⊟ Messages

⚠ 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.

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

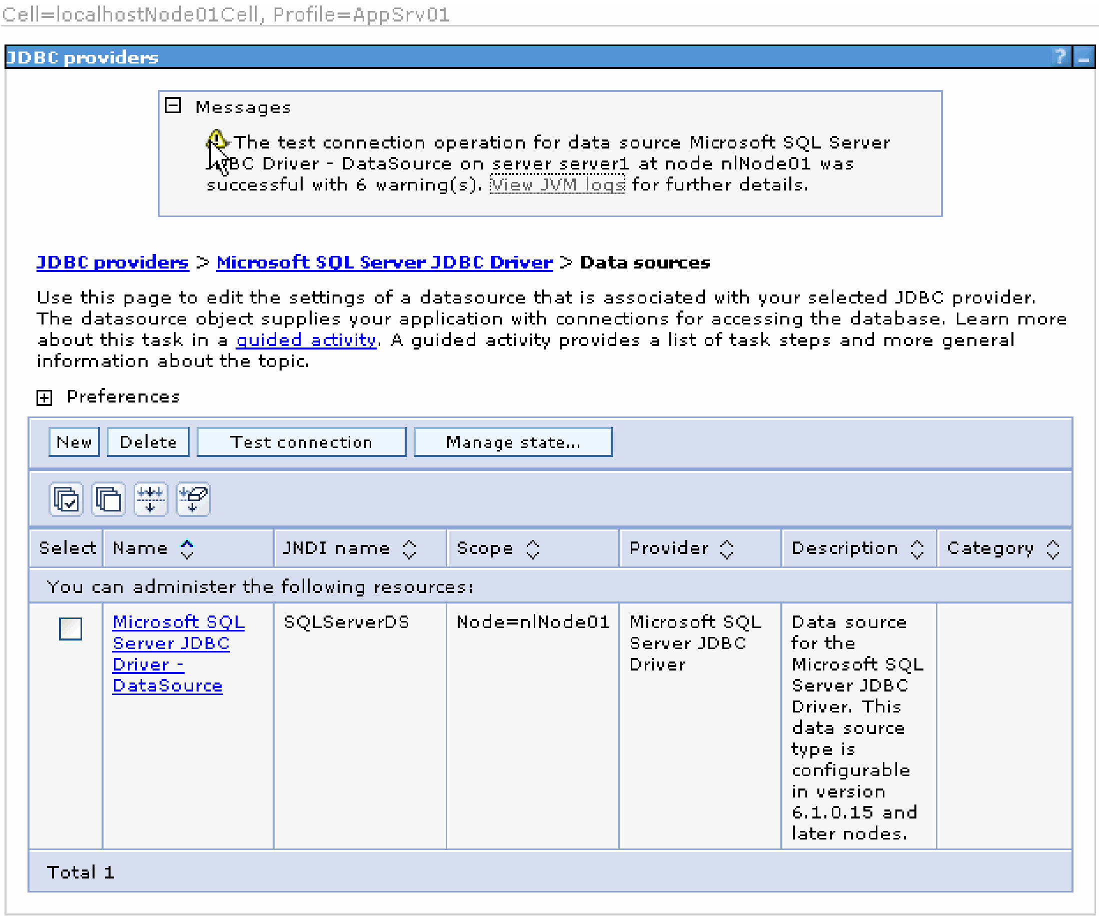
JDBC providers > Microsoft SQL Server JDBC Driver > Data sources

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

Select	Name	JNDI name	Scope	Provider	Description	Category
You can administer the following resources:						
☑	Microsoft SQL Server JDBC Driver - DataSource	SQLServerDS	Node=nlNode01	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

⊟ Messages

⚠ The test connection operation for data source Microsoft SQL Server JDBC Driver - DataSource on server server1 at node nlNode01 was successful with 6 warning(s). View JVM logs for further details.

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

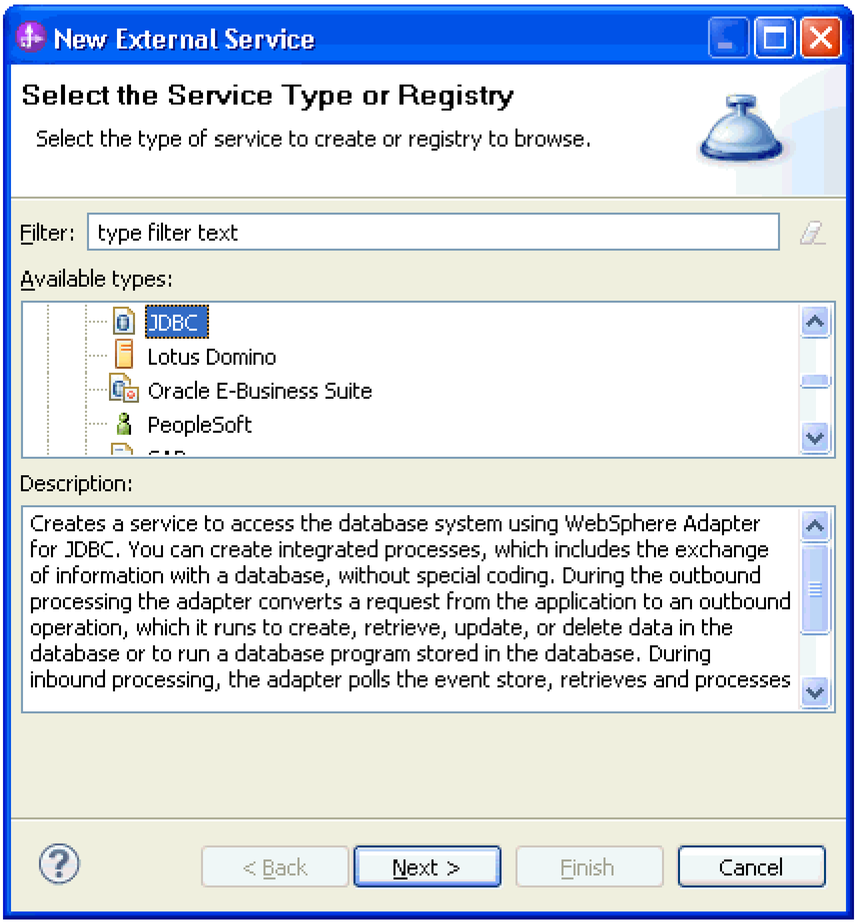
Select	Name	JNDI name	Scope	Provider	Description	Category
You can administer the following resources:						
□	Microsoft SQL Server JDBC Driver - DataSource	SQLServerDS	Node=nlNode01	Microsoft SQL Server JDBC Driver	Data source for the Microsoft SQL Server JDBC Driver. This data source type is configurable 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 IBM Integration Designer 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**.



New External Service

Select the Service Type or Registry

Select the type of service to create or registry to browse.

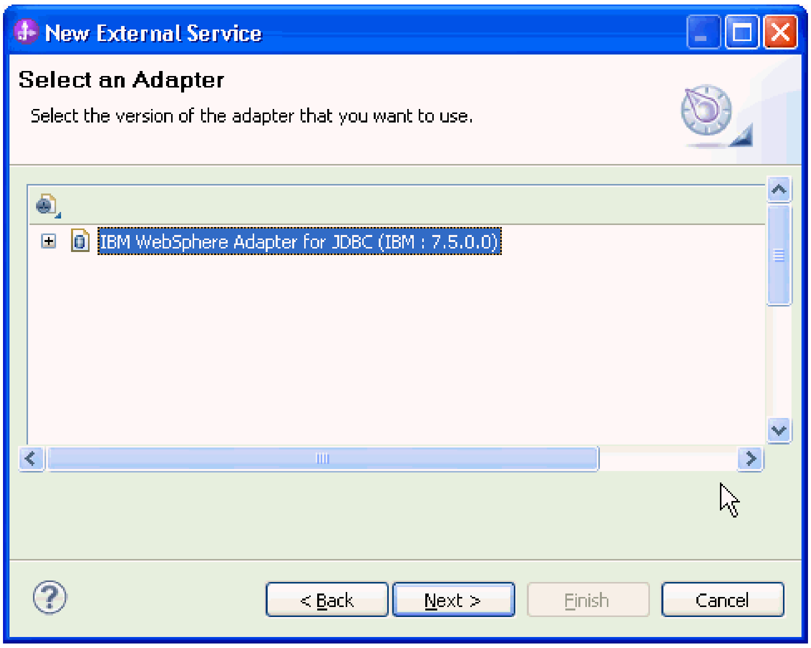
Filter: type filter text

Available types:

Description:

Creates a service to access the database system using WebSphere Adapter for JDBC. You can create integrated processes, which includes the exchange of information with a database, without special coding. During the outbound processing the adapter converts a request from the application to an outbound operation, which it runs to create, retrieve, update, or delete data in the database or to run a database program stored in the database. During inbound processing, the adapter polls the event store, retrieves and processes

4. Select the **IBM WebSphere Adapter for JDBC (IBM: 7.5.0.0)** and click **Next**.

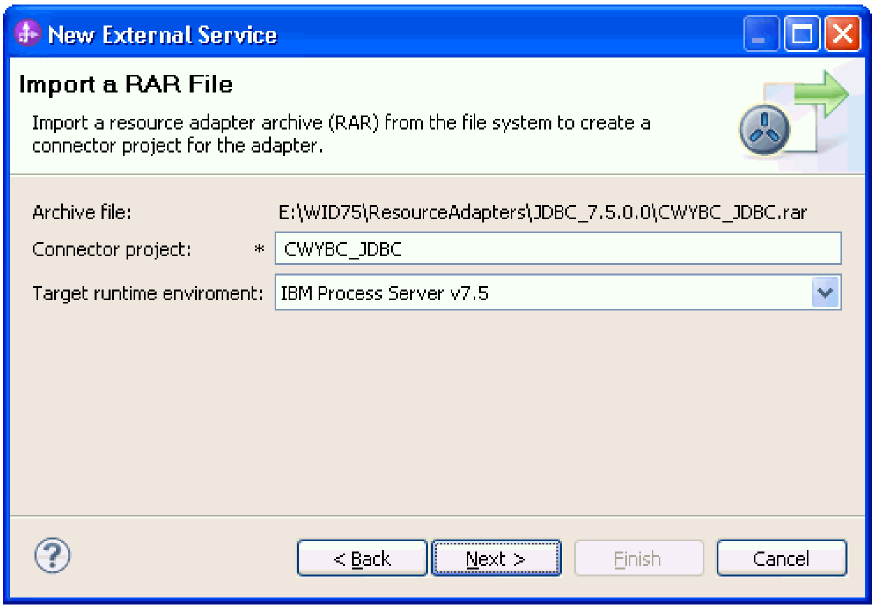


Select an Adapter

Select the version of the adapter that you want to use.

IBM WebSphere Adapter for JDBC (IBM : 7.5.0.0)

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



Import a RAR File

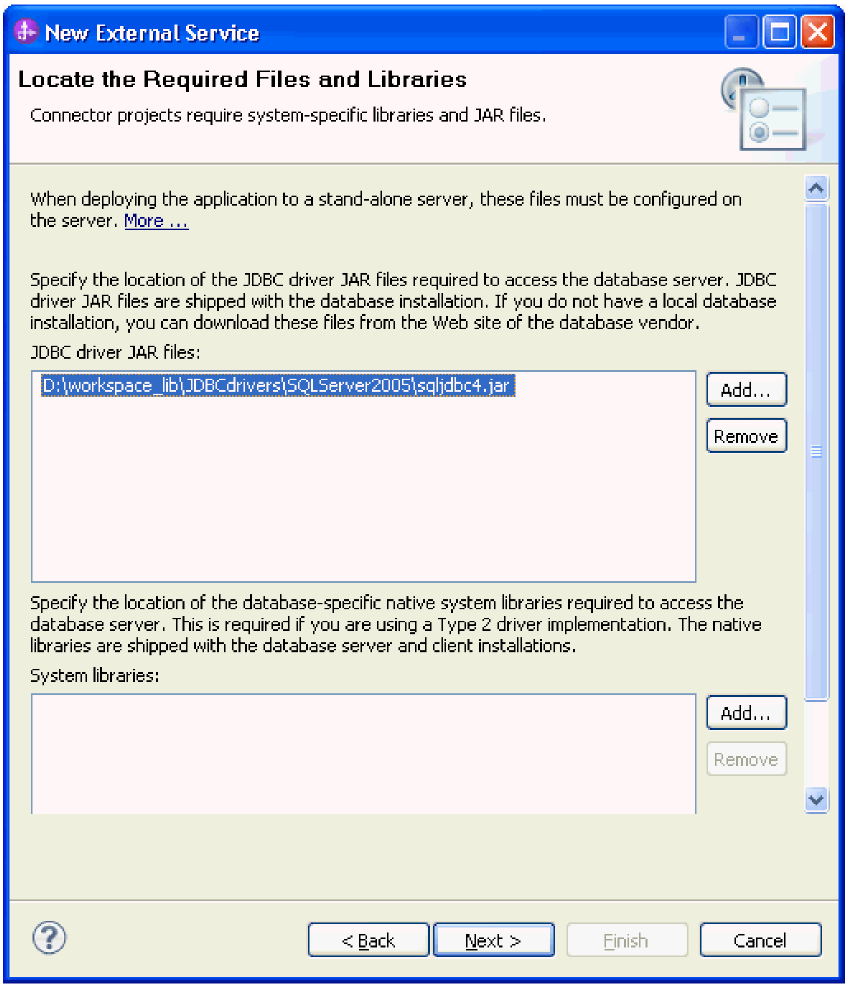
Import a resource adapter archive (RAR) from the file system to create a connector project for the adapter.

Archive file: E:\WID75\ResourceAdapters\JDBC_7.5.0.0\CWYBC_JDBC.rar

Connector project: * CWYBC_JDBC

Target runtime enviroment: IBM Process Server v7.5

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**.



Locate the Required Files and Libraries

Connector projects require system-specific libraries and JAR files.

When deploying the application to a stand-alone server, these files must be configured on the server. More ...

Specify the location of the JDBC driver JAR files required to access the database server. JDBC driver JAR files are shipped with the database installation. If you do not have a local database installation, you can download these files from the Web site of the database vendor.

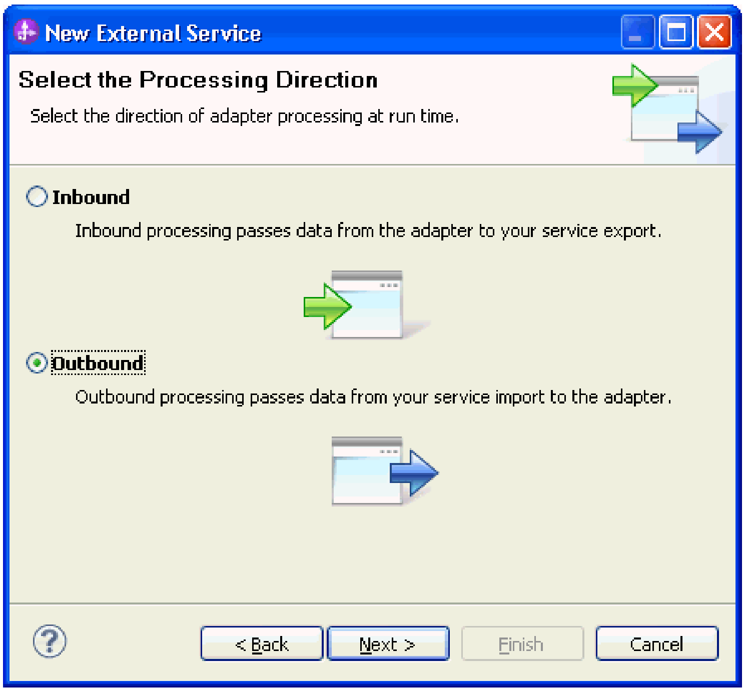
JDBC driver JAR files:

D:\workspace_lib\JDBCdrivers\SQLServer2005\sqljdbc4.jar

Specify the location of the database-specific native system libraries required to access the database server. This is required if you are using a Type 2 driver implementation. The native libraries are shipped with the database server and client installations.

System libraries:

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



Select the Processing Direction

Select the direction of adapter processing at run time.

○ **Inbound**

Inbound processing passes data from the adapter to your service export.

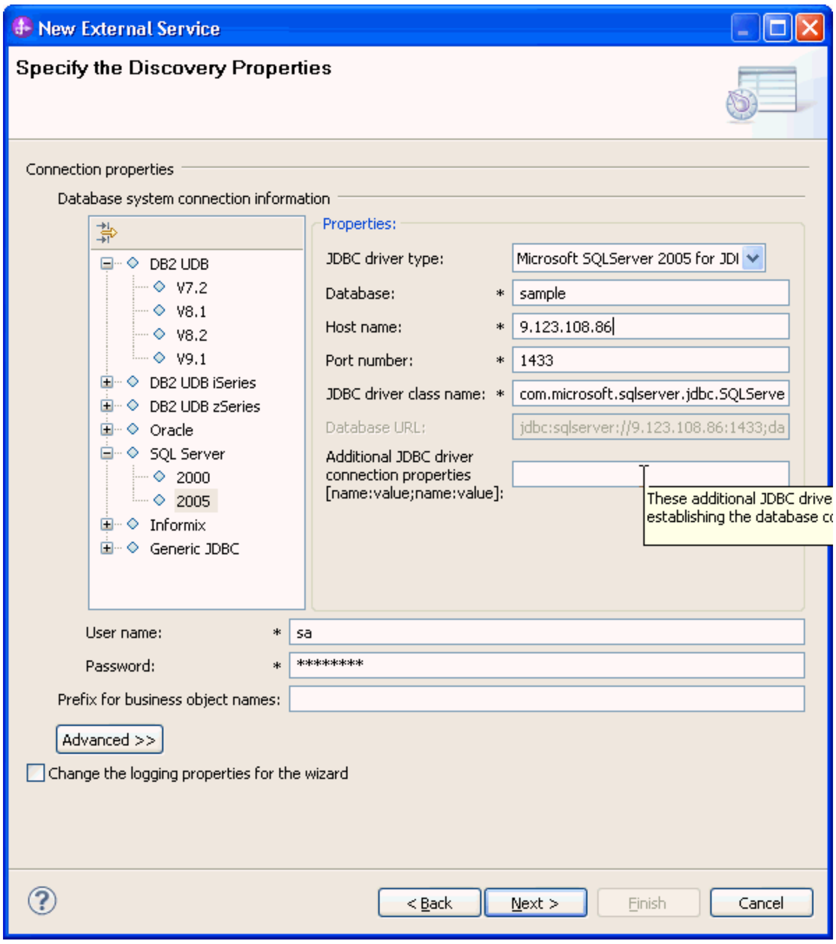
● **Outbound**

Outbound processing passes data from your service import to the adapter.

Set connection properties for the external service wizard

To connect to the SQL Server:

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**.



JDBC driver type: Microsoft SQLServer 2005 for JDI

Database: * sample

Host name: * 9.123.108.86

Port number: * 1433

JDBC driver class name: * com.microsoft.sqlserver.jdbc.SQLServe

Database URL: jdbc:sqlserver://9.123.108.86:1433;da

Additional JDBC driver connection properties [name:value;name:value]:

These additional JDBC drive
establishing the database co

User name: * sa

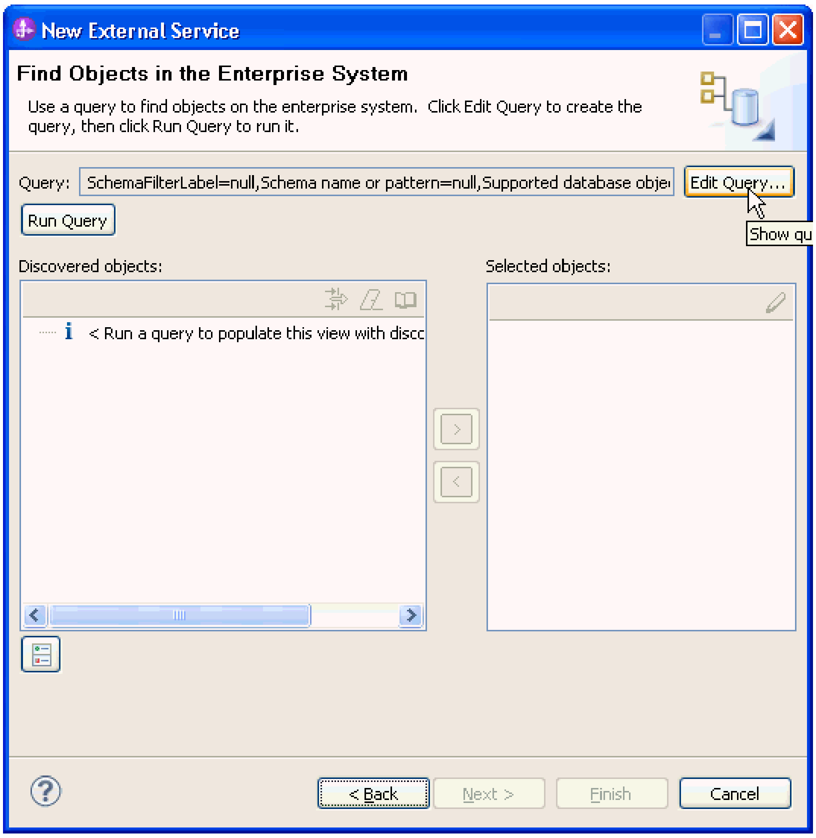
Password: * *********

Prefix for business object names:

☐ Change the logging properties for the wizard

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

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



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.

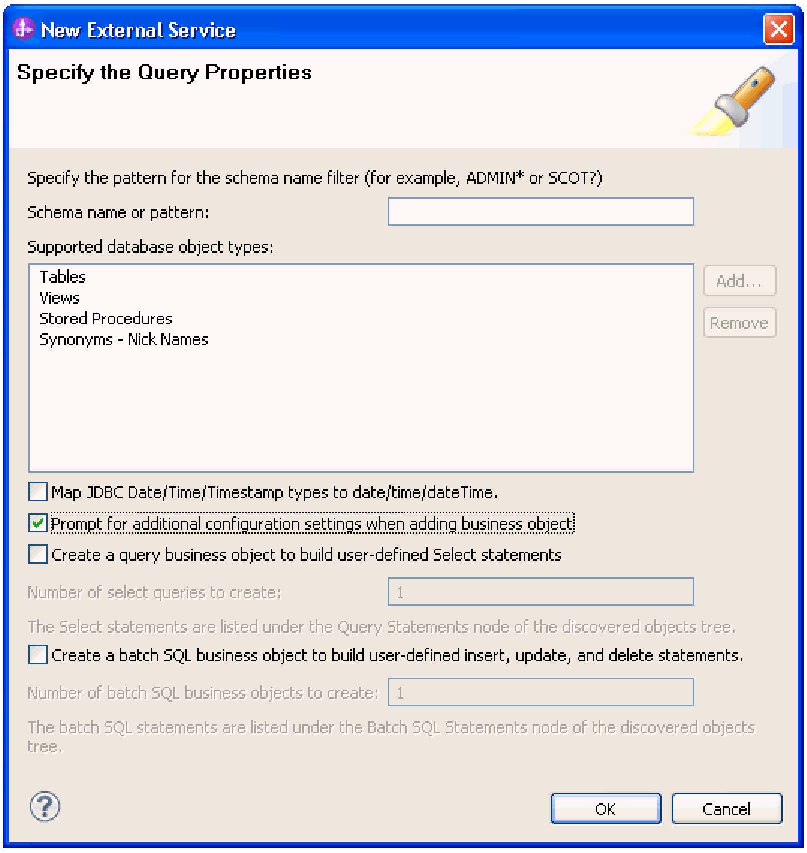
Query: SchemaFilterLabel=null,Schema name or pattern=null,Supported database obje

Discovered objects:

< Run a query to populate this view with disco

Selected objects:

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



New External Service

Specify the Query Properties

Specify the pattern for the schema name filter (for example, ADMIN* or SCOT?)

Schema name or pattern:

Supported database object types:

Tables
Views
Stored Procedures
Synonyms - Nick Names

□ Map JDBC Date/Time/Timestamp types to date/time/dateTime.

☑ Prompt for additional configuration settings when adding business object

□ Create a query business object to build user-defined Select statements

Number of select queries to create: 1

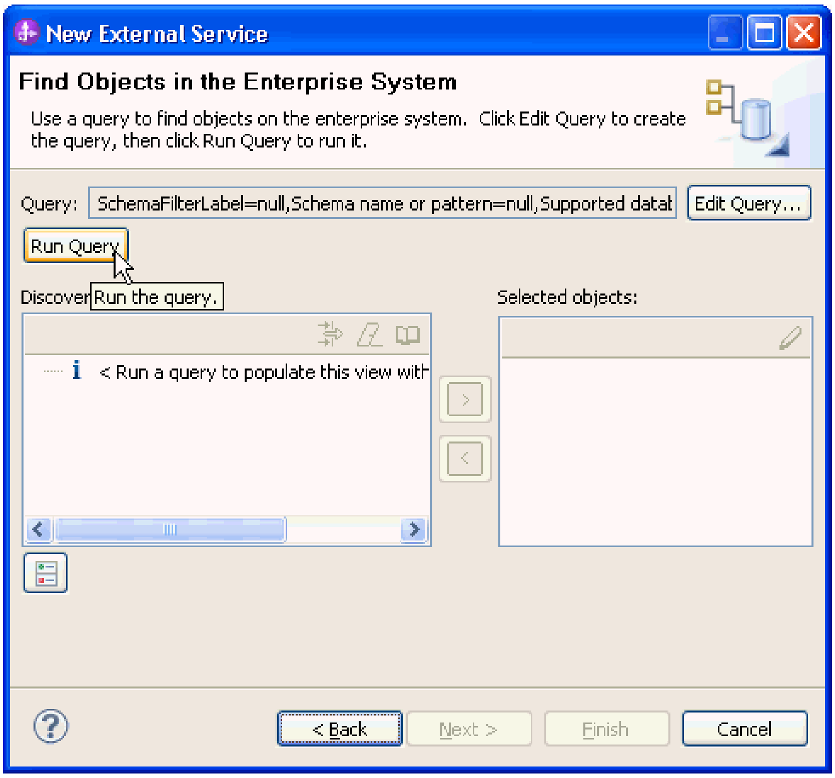
The Select statements are listed under the Query Statements node of the discovered objects tree.

□ Create a batch SQL business object to build user-defined insert, update, and delete statements.

Number of batch SQL business objects to create: 1

The batch SQL statements are listed under the Batch SQL Statements node of the discovered objects tree.

3. Click **Run Query**.



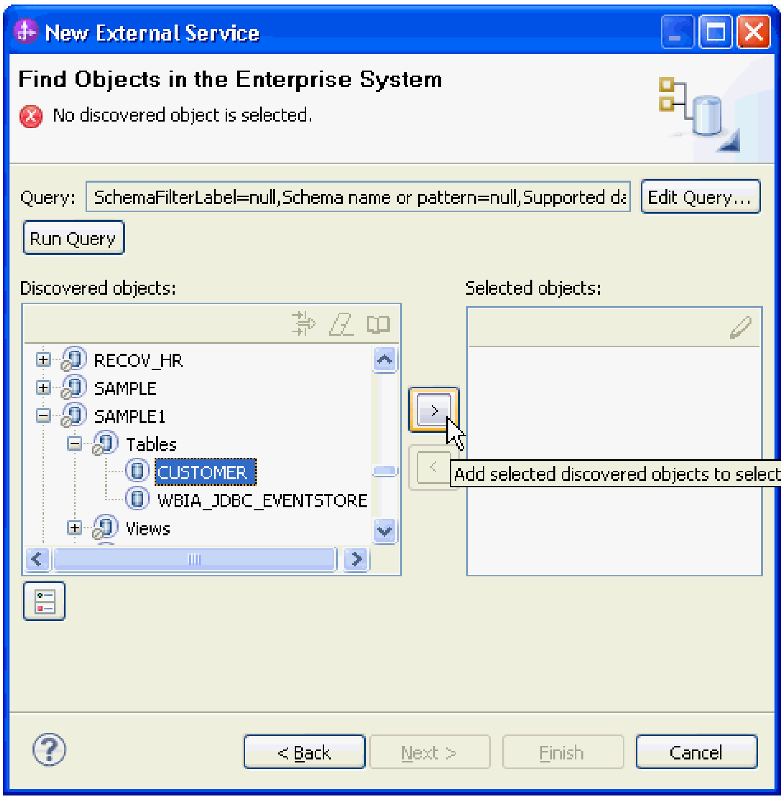
New External Service

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 datat

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



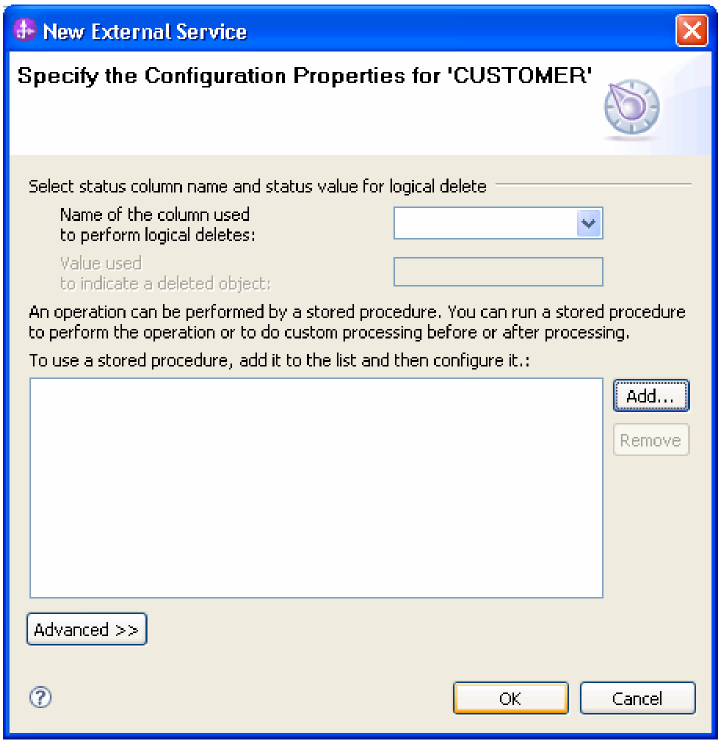
New External Service

Find Objects in the Enterprise System

No discovered object is selected.

Query: SchemaFilterLabel=null,Schema name or pattern=null,Supported da

5. Select the **CUSTOMER** table and click [>]. In the Specify the Configuration Properties for ‘CUSTOMER’ window click **OK.**



New External Service

Specify the Configuration Properties for 'CUSTOMER'

Select status column name and status value for logical delete

Name of the column used to perform logical deletes:

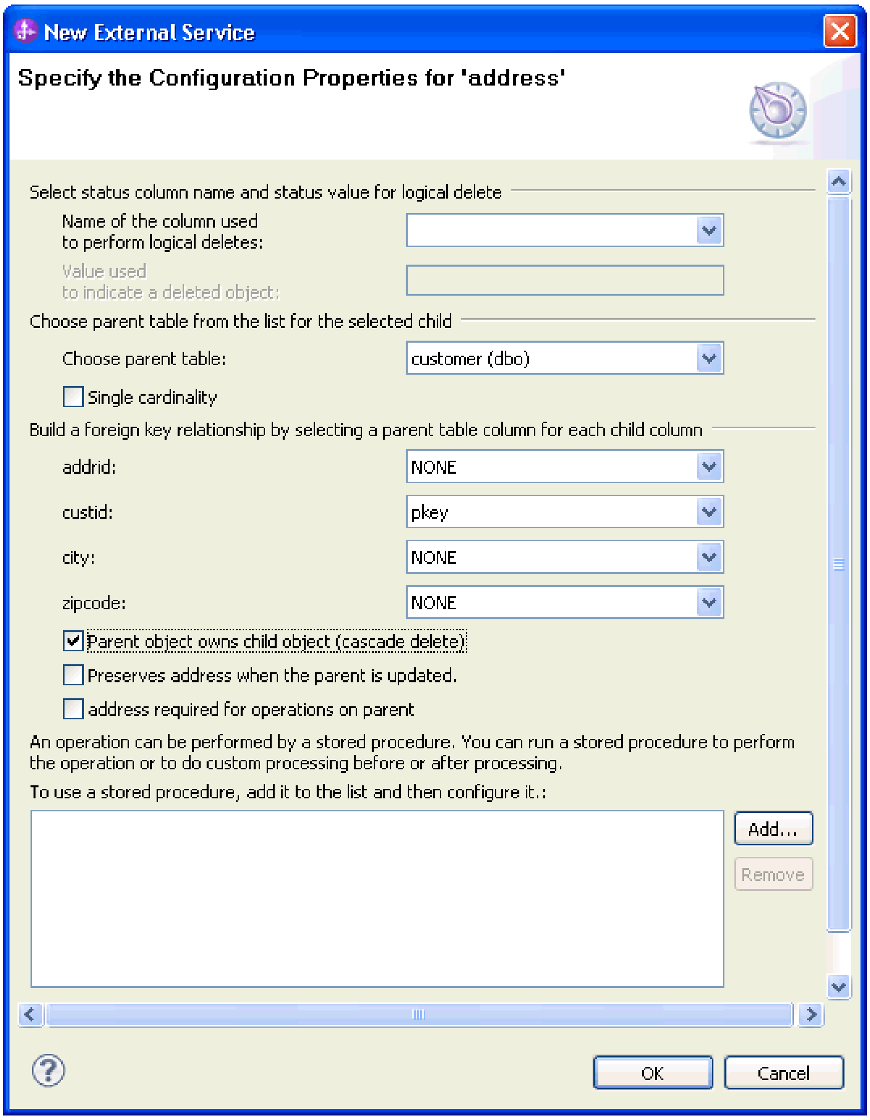
Value used to indicate a deleted object:

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.:

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: NONE

custid: pkey

city: NONE

zipcode: NONE

☑ Parent object owns child object (cascade delete)

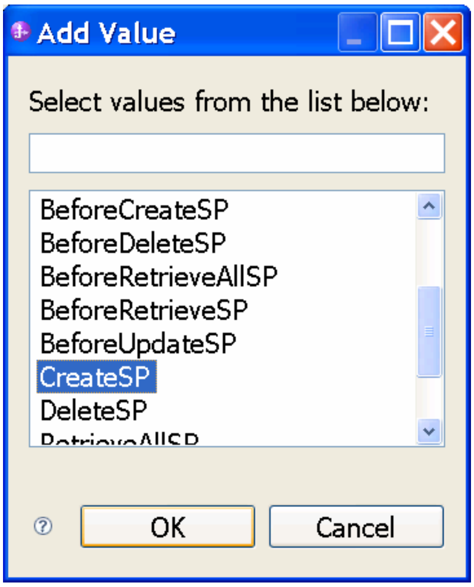
□ Preserves address when the parent is updated.

□ 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**.

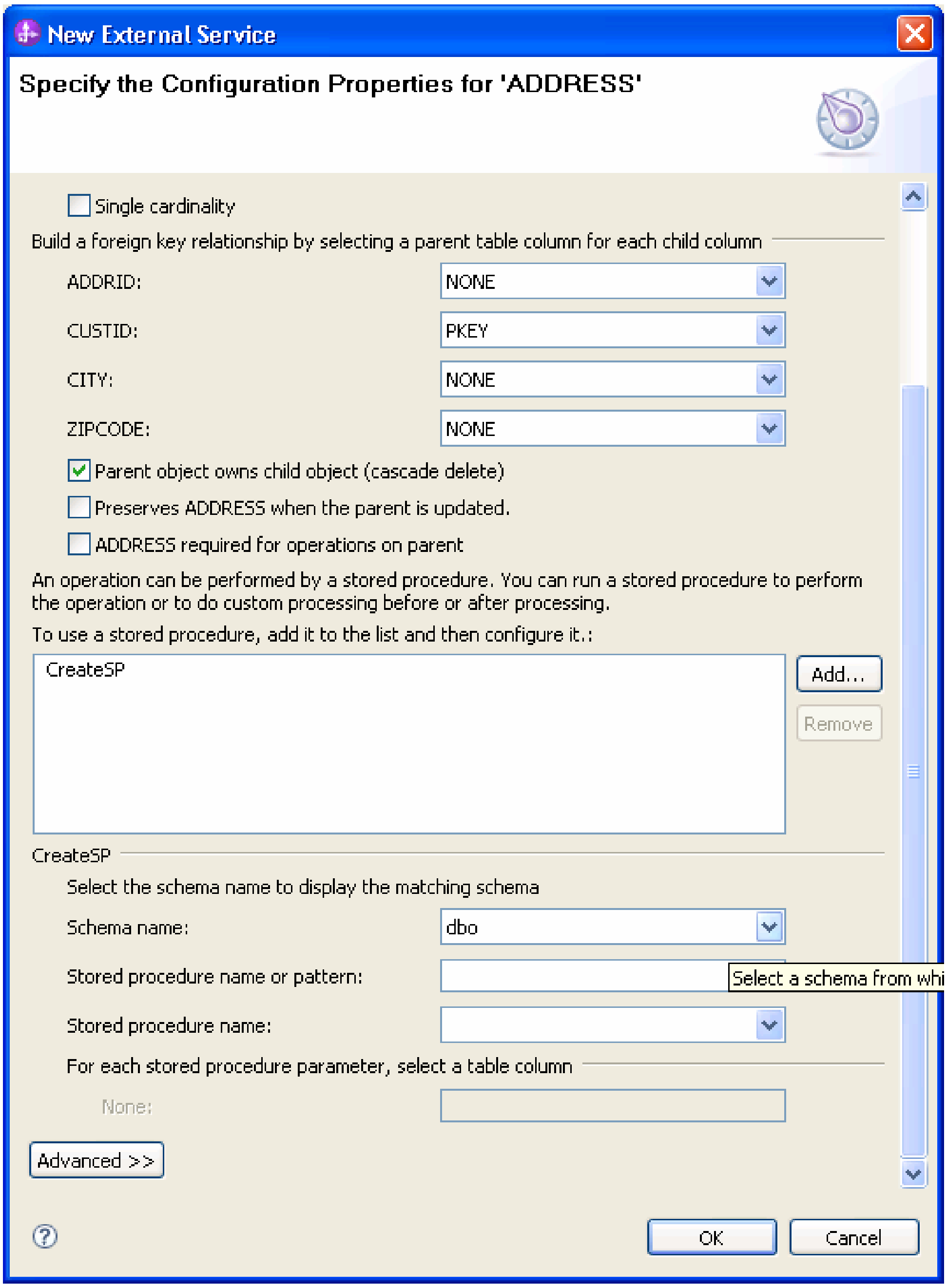


Add Value

Select values from the list below:

- BeforeCreateSP
- BeforeDeleteSP
- BeforeRetrieveAllSP
- BeforeRetrieveSP
- BeforeUpdateSP
- CreateSP
- DeleteSP
- RetrieveAllSP

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



New External Service

Specify the Configuration Properties for 'ADDRESS'

□ 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 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.:

CreateSP

CreateSP

Select the schema name to display the matching schema

Schema name: dbo

Stored procedure name or pattern:

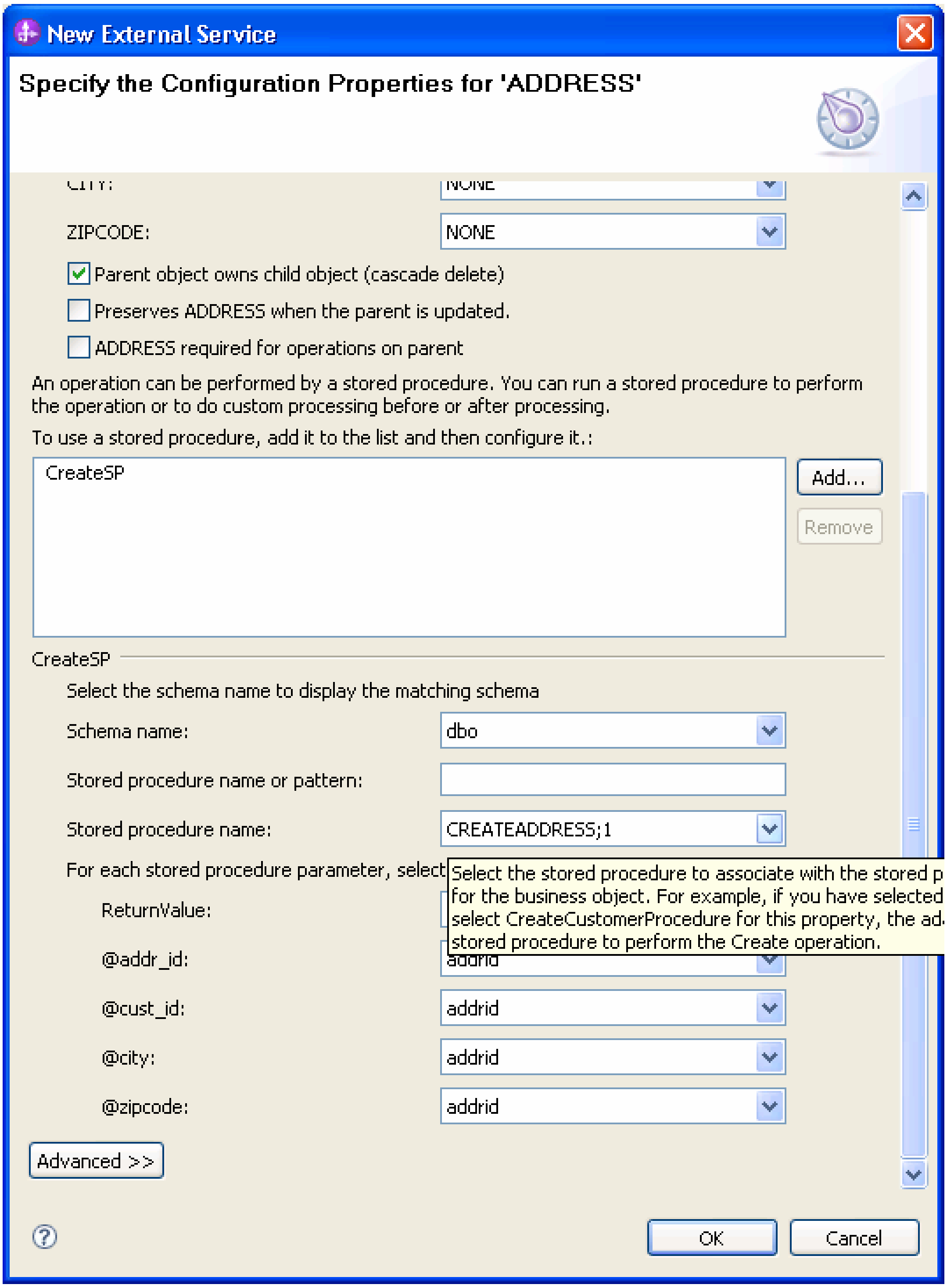
Select a schema from whi

Stored procedure name:

For each stored procedure parameter, select a table column

None:

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



New External Service

Specify the Configuration Properties for 'ADDRESS'

CITY: NONE

ZIPCODE: 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.:

CreateSP

CreateSP

Select the schema name to display the matching schema

Schema name: dbo

Stored procedure name or pattern:

Stored procedure name: CREATEADDRESS;1

For each stored procedure parameter, select

Select the stored procedure to associate with the stored p
for the business object. For example, if you have selected
select CreateCustomerProcedure for this property, the ad
stored procedure to perform the Create operation.

ReturnValue:

@addr_id: addrid

@cust_id: addrid

@city: addrid

@zipcode: addrid

11. Select stored procedure parameter for each column.

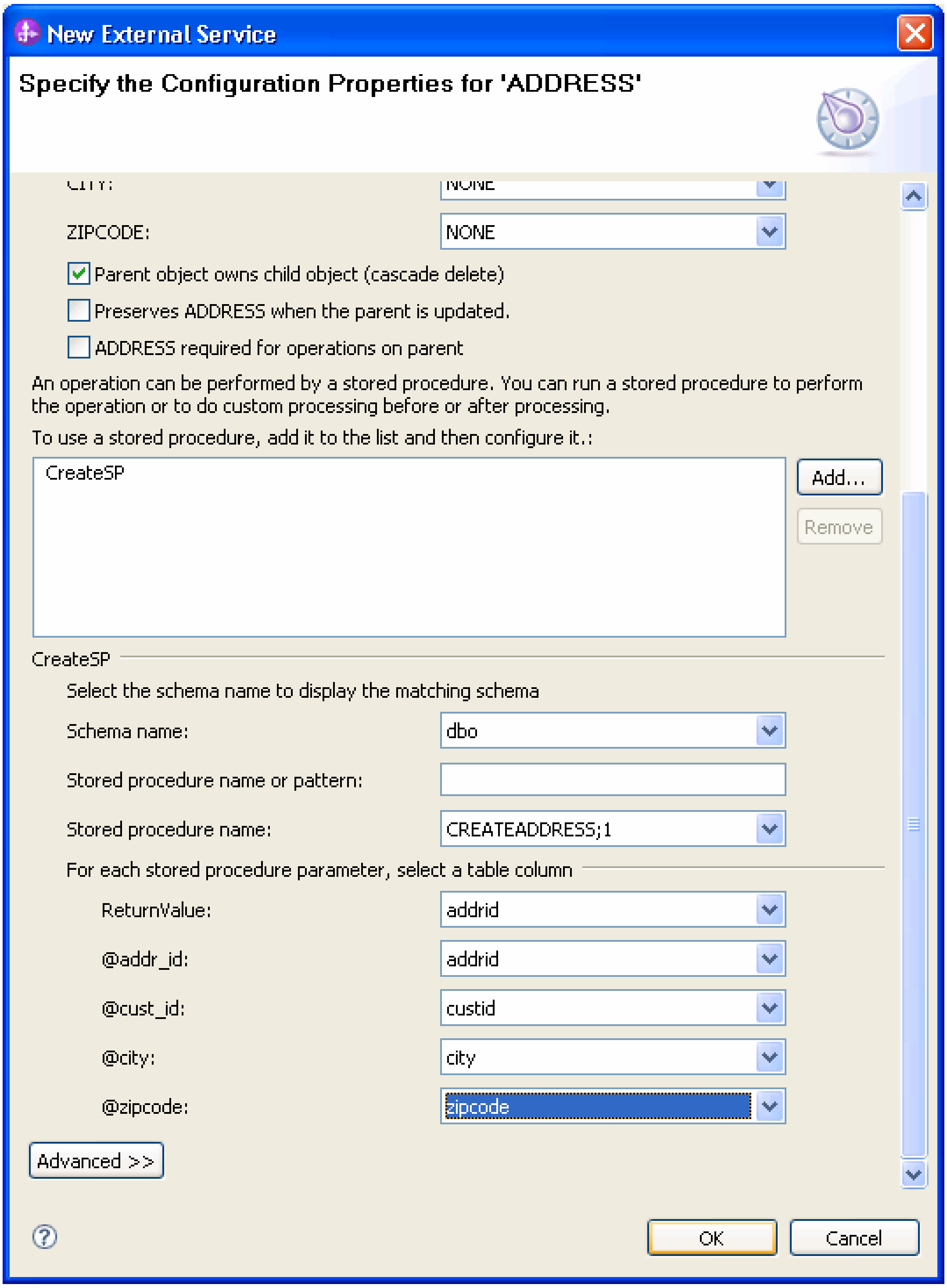
    @addr_id: addrid

    @cust_id: custid

    @city: city

    @zipcode: zipcode

12. Click **OK**.



New External Service

Specify the Configuration Properties for 'ADDRESS'

CITY: NONE

ZIPCODE: 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.:

CreateSP

CreateSP

Select the schema name to display the matching schema

Schema name: dbo

Stored procedure name or pattern:

Stored procedure name: CREATEADDRESS;1

For each stored procedure parameter, select a table column

ReturnValue: addrid

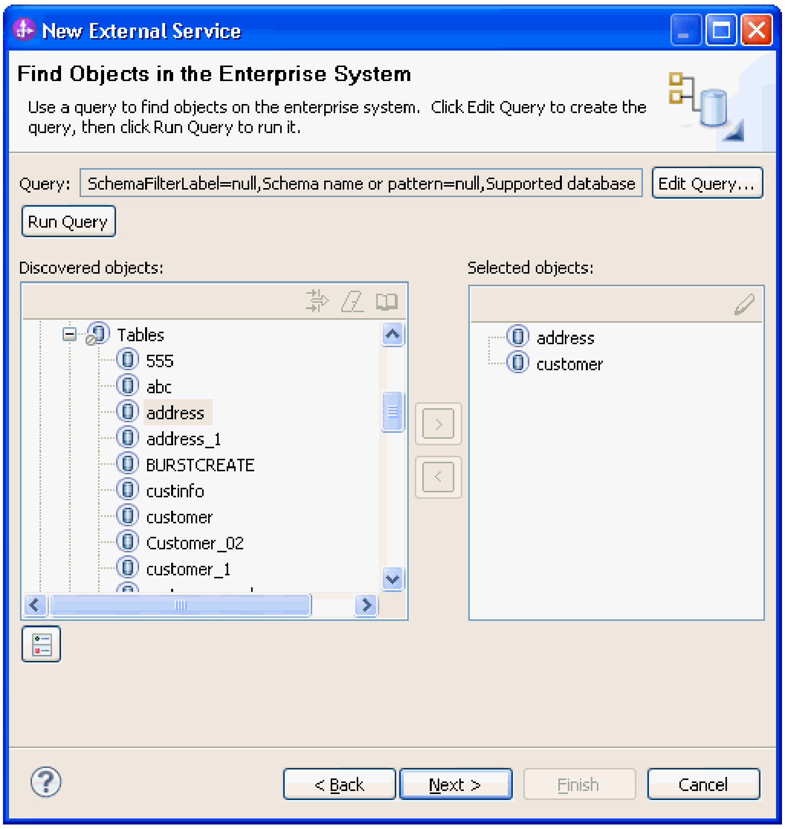
@addr_id: addrid

@cust_id: custid

@city: city

@zipcode: zipcode

13. In the Find Objects in Enterprise System window, 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

Discovered objects:

- Tables
  - 555
  - abc
  - address
  - address_1
  - BURSTCREATE
  - custinfo
  - customer
  - Customer_02
  - customer_1

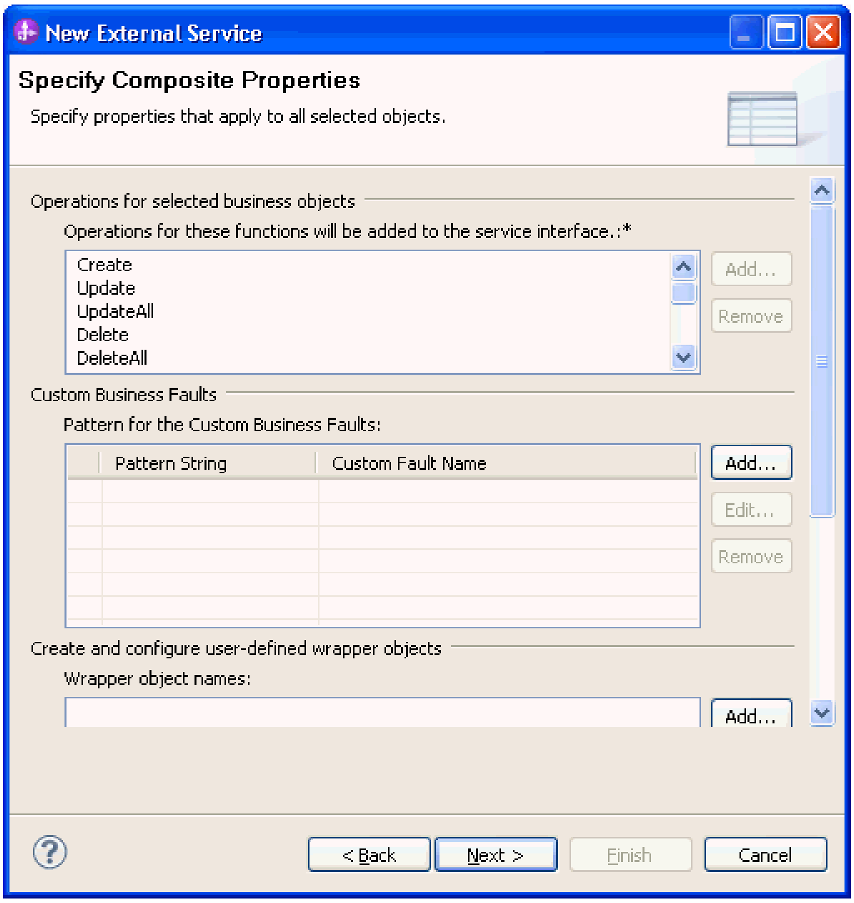
Selected objects:

- address
- customer

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**.



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
UpdateAll
Delete
DeleteAll

Custom Business Faults

Pattern for the Custom Business Faults:

Pattern String	Custom Fault Name

Create and configure user-defined wrapper objects

Wrapper object names:

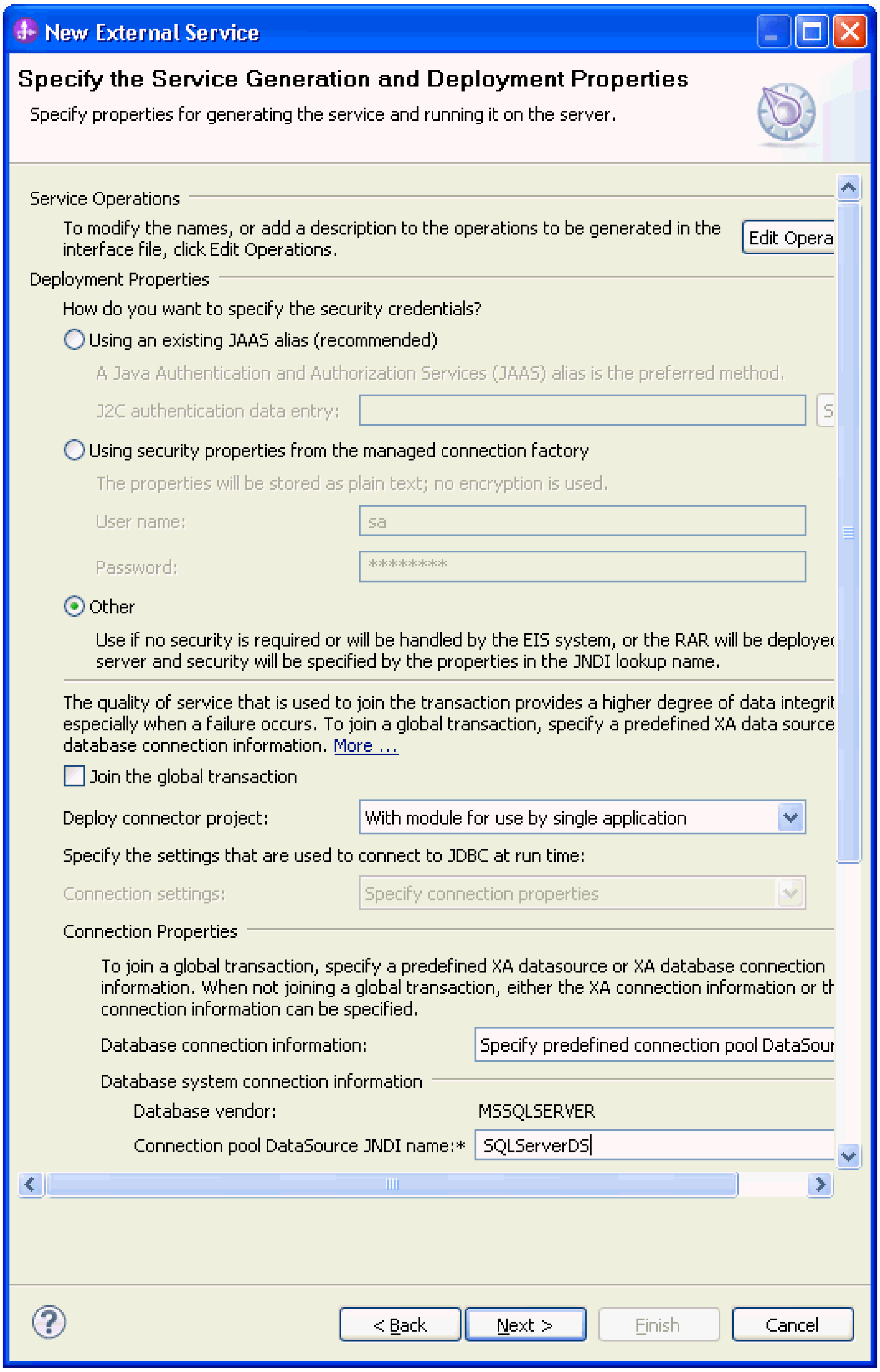
2. In the Specify the Service Generation and Deployment Properties window, perform the following steps:

   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**.



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 managed connection factory

The properties will be stored as plain text; no encryption is used.

User name: sa

Password: ********

● Other

Use if no security is required or will be handled by the EIS system, or the RAR will be deployed 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 integrit especially when a failure occurs. To join a global transaction, specify a predefined XA data source database connection information. More ...

□ Join the global transaction

Deploy connector project: With module for use by single application

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

Connection settings: Specify connection properties

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 th connection information can be specified.

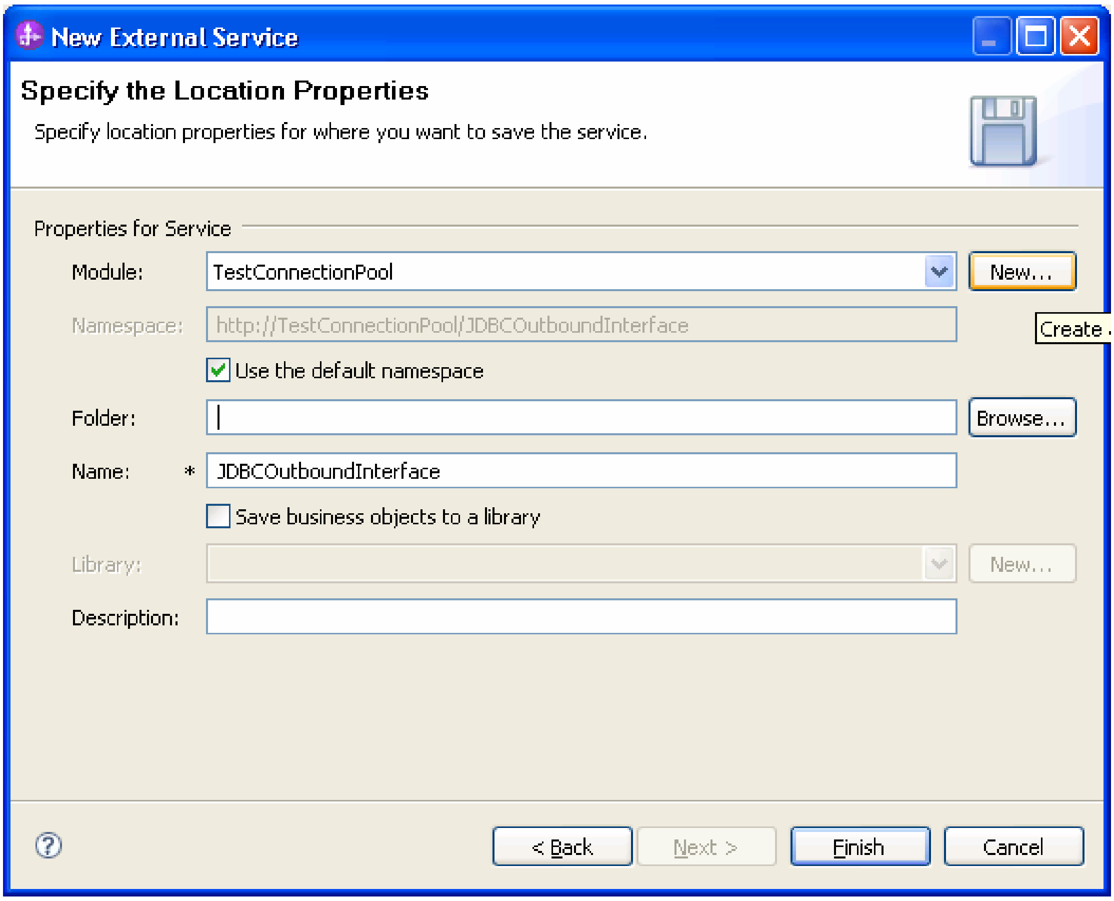
Database connection information: Specify predefined connection pool DataSour

Database system connection information

Database vendor: MSSQLSERVER

Connection pool DataSource JNDI name:* SQLServerDS

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



New External Service

Specify the Location Properties

Specify location properties for where you want to save the service.

Properties for Service

Module: TestConnectionPool

Namespace: http://TestConnectionPool/JDBCOutboundInterface

☑ Use the default namespace

Folder:

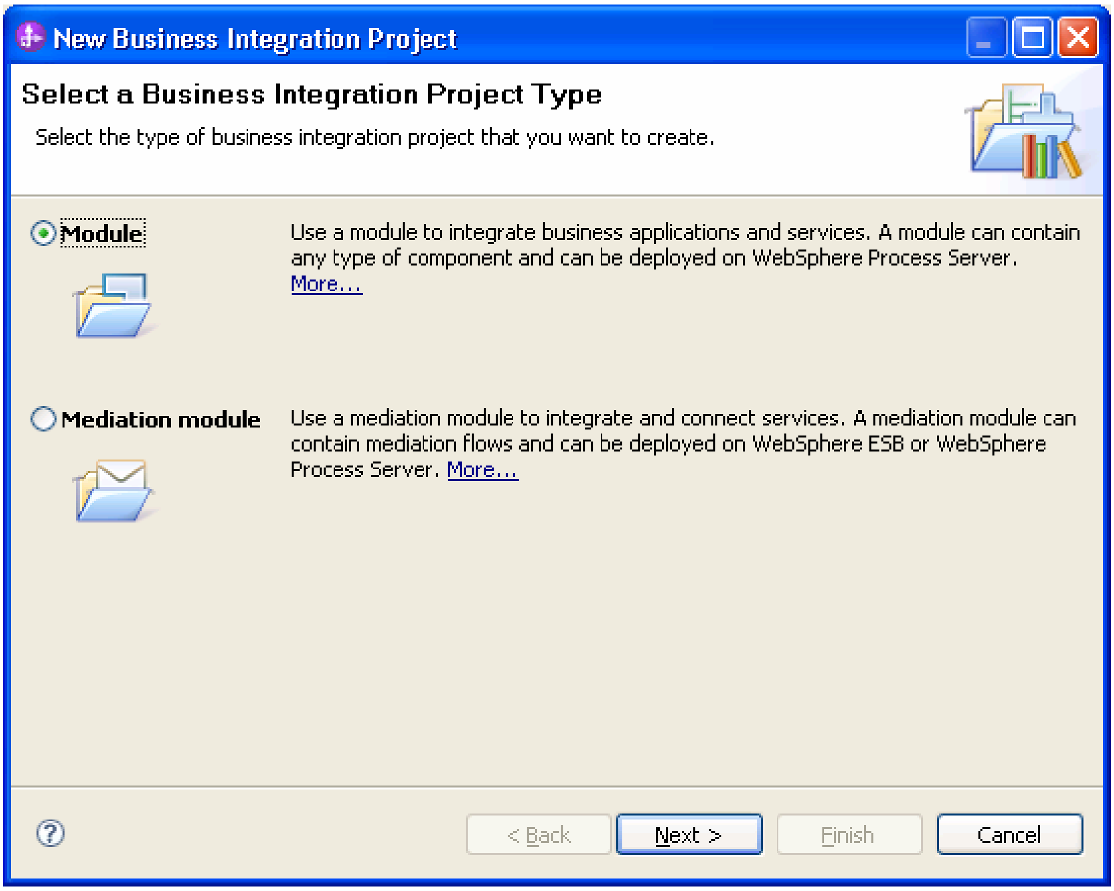
Name: * JDBCOutboundInterface

☐ Save business objects to a library

Library:

Description:

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



New Business Integration Project

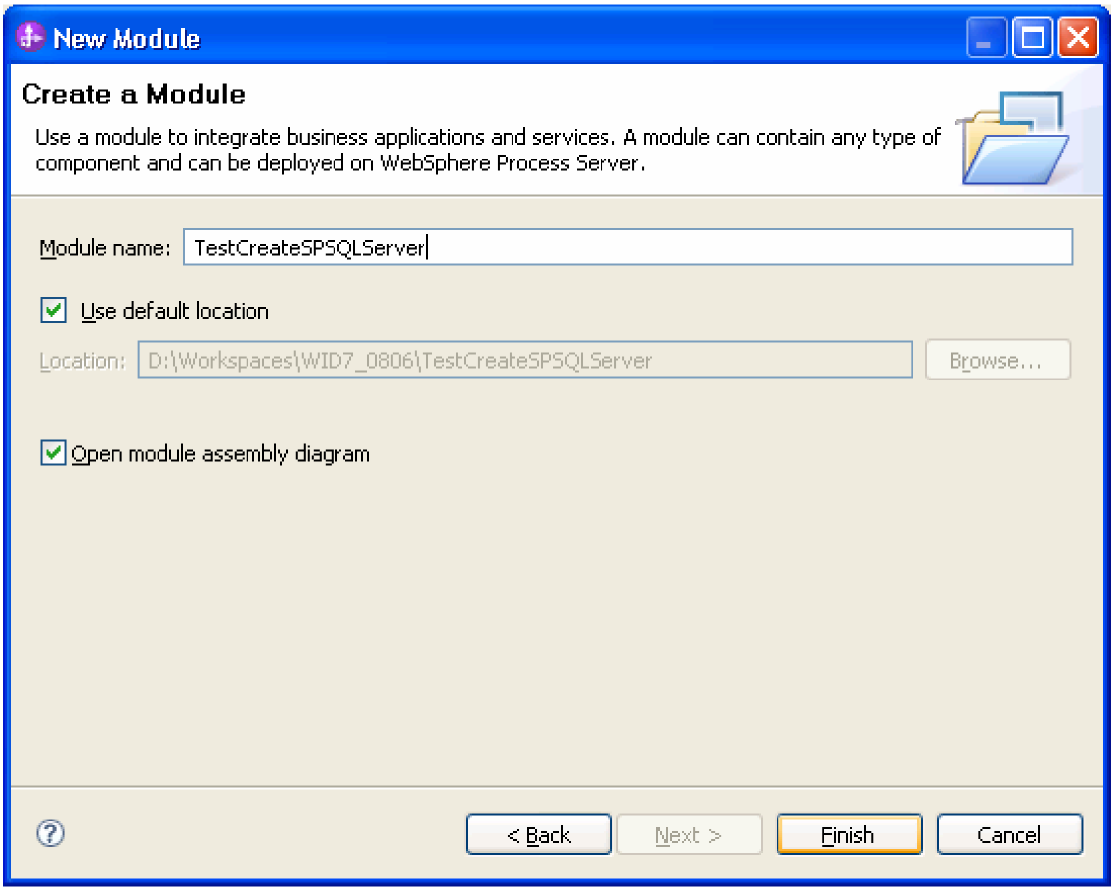
Select a Business Integration Project Type

Select the type of business integration project that you want to create.

● Module — Use a module to integrate business applications and services. A module can contain any type of component and can be deployed on WebSphere Process Server. More...

○ Mediation module — Use a mediation module to integrate and connect services. A mediation module can contain mediation flows and can be deployed on WebSphere ESB or WebSphere Process Server. More...

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



New Module

Create a Module

Use a module to integrate business applications and services. A module can contain any type of component and can be deployed on WebSphere Process Server.

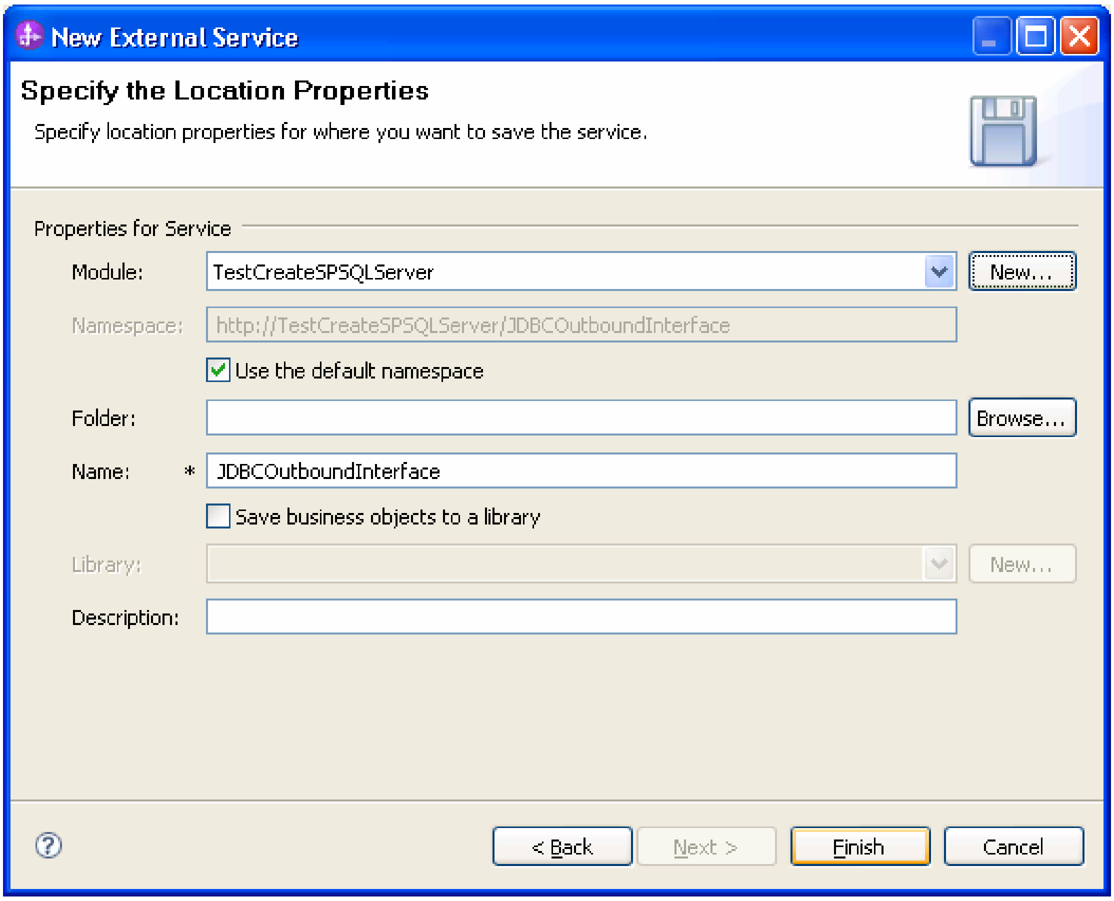
Module name: TestCreateSPSQLServer

☑ Use default location

Location: D:\Workspaces\WID7_0806\TestCreateSPSQLServer

☑ Open module assembly diagram

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



New External Service

Specify the Location Properties

Specify location properties for where you want to save the service.

Properties for Service

Module: TestCreateSPSQLServer

Namespace: http://TestCreateSPSQLServer/JDBCOutboundInterface

☑ Use the default namespace

Folder:

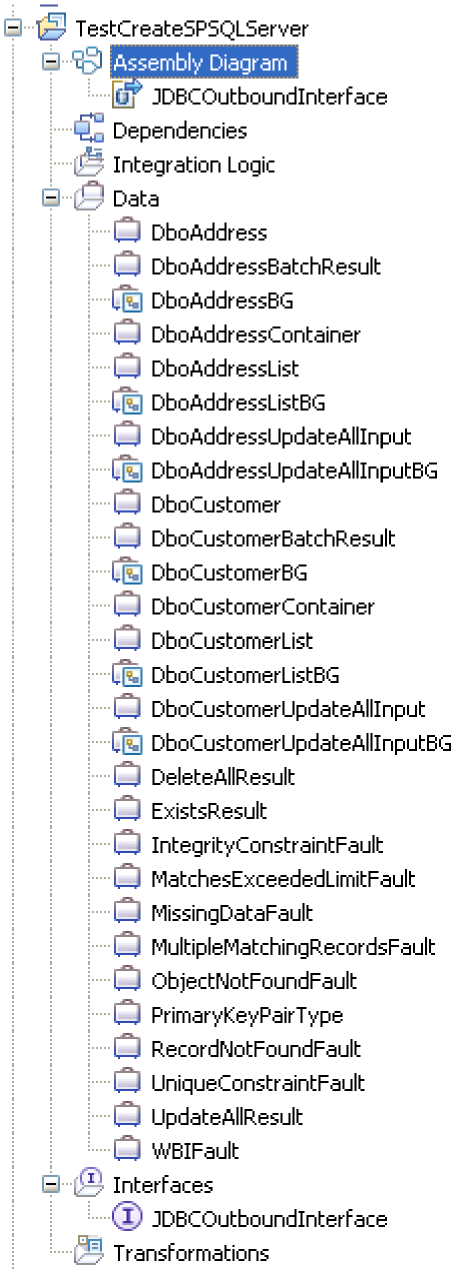
Name: * JDBCOutboundInterface

□ Save business objects to a library

Library:

Description:

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



```
TestCreateSPSQLServer
  Assembly Diagram
    JDBCOutboundInterface
  Dependencies
  Integration Logic
  Data
    DboAddress
    DboAddressBatchResult
    DboAddressBG
    DboAddressContainer
    DboAddressList
    DboAddressListBG
    DboAddressUpdateAllInput
    DboAddressUpdateAllInputBG
    DboCustomer
    DboCustomerBatchResult
    DboCustomerBG
    DboCustomerContainer
    DboCustomerList
    DboCustomerListBG
    DboCustomerUpdateAllInput
    DboCustomerUpdateAllInputBG
    DeleteAllResult
    ExistsResult
    IntegrityConstraintFault
    MatchesExceededLimitFault
    MissingDataFault
    MultipleMatchingRecordsFault
    ObjectNotFoundFault
    PrimaryKeyPairType
    RecordNotFoundFault
    UniqueConstraintFault
    UpdateAllResult
    WBIFault
  Interfaces
    JDBCOutboundInterface
  Transformations
```

Deploy the module to the test environment

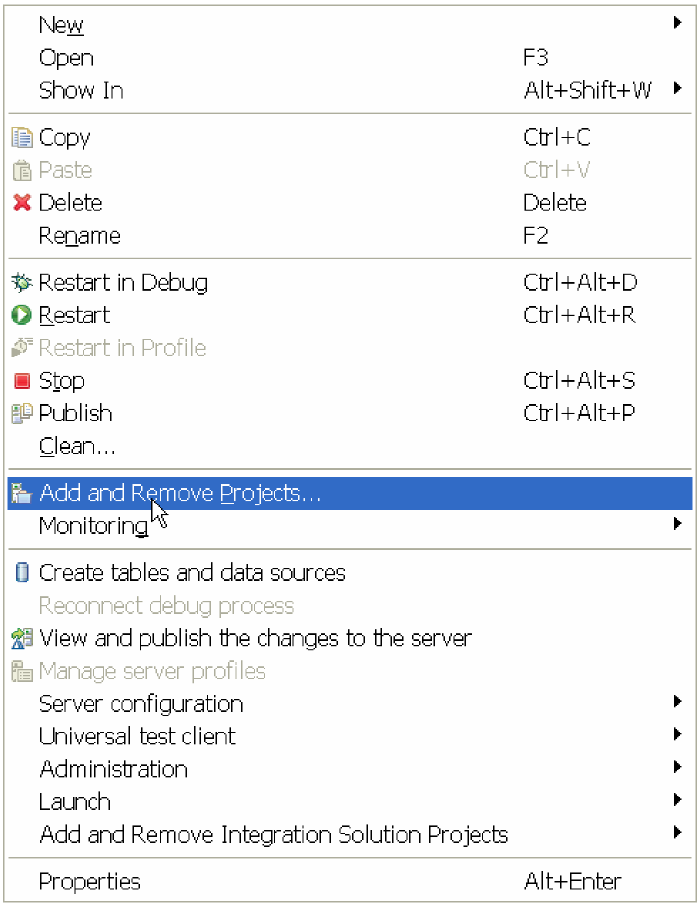
The result of running the external service wizard is a Service Component Architecture (SCA) module that contains an Enterprise Information System import. You must install this SCA module in the IBM Integration Designer integration test client. To do this, you must add the SCA module you created earlier to the server using the **Servers** view in IBM Integration Designer.

Steps for adding the SCA module to the server:

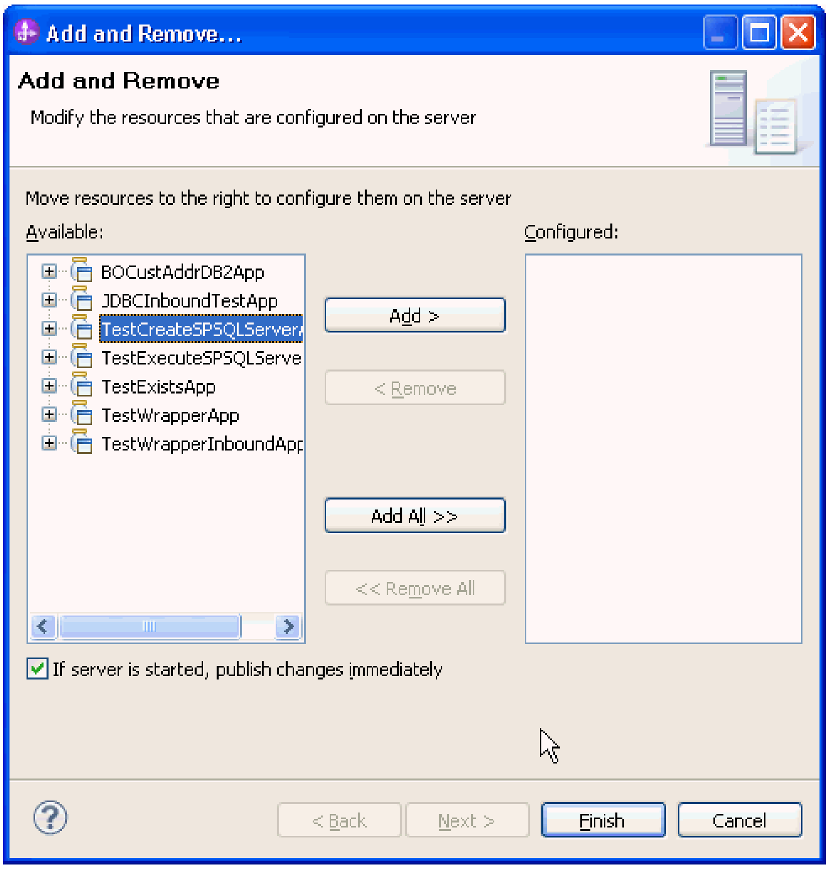
1. In IBM Integration Designer, switch to the **Servers** view by selecting **Windows** > **Show View** > **Servers**.

2. In the Servers tab in the lower-right pane of the IBM Integration Designer 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 IBM Integration Designer workspace.



Add and Remove...

Add and Remove

Modify the resources that are configured on the server

Move resources to the right to configure them on the server

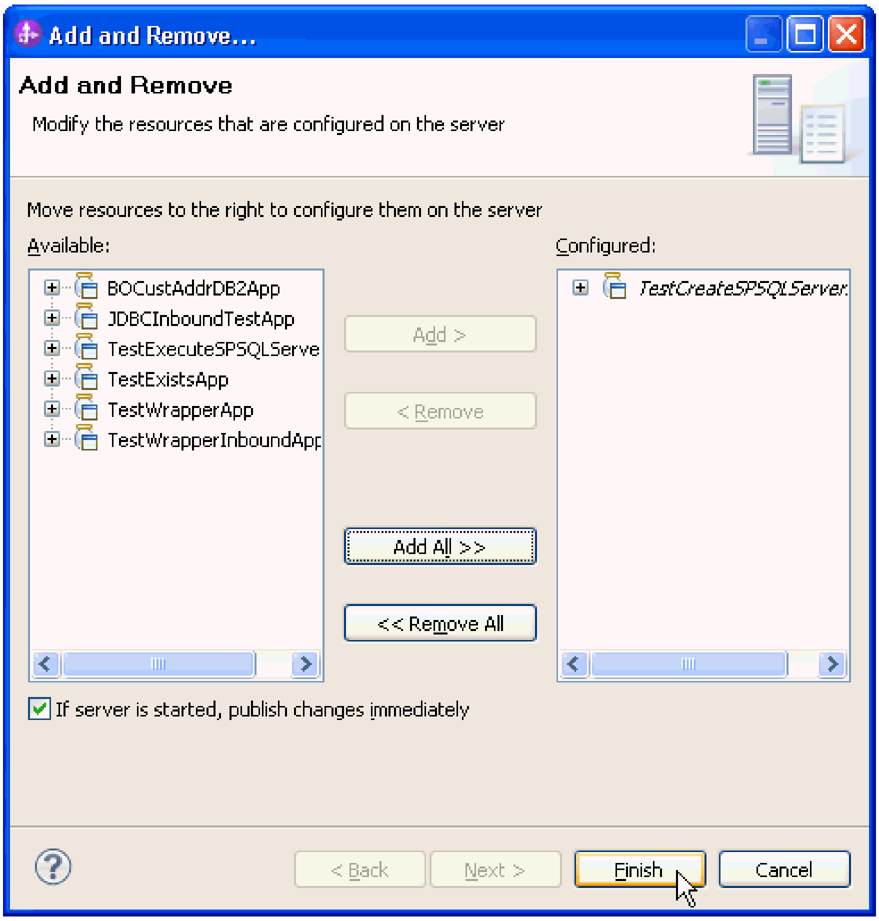
Available:

- BOCustAddrDB2App
- JDBCInboundTestApp
- TestCreateSPSQLServerA
- TestExecuteSPSQLServe
- TestExistsApp
- TestWrapperApp
- TestWrapperInboundApp

Configured:

☑ If server is started, publish changes immediately

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



Add and Remove...

Add and Remove

Modify the resources that are configured on the server

Move resources to the right to configure them on the server

Available:

- BOCustAddrDB2App
- JDBCInboundTestApp
- TestExecuteSPSQLServe
- TestExistsApp
- TestWrapperApp
- TestWrapperInboundApp

Configured:

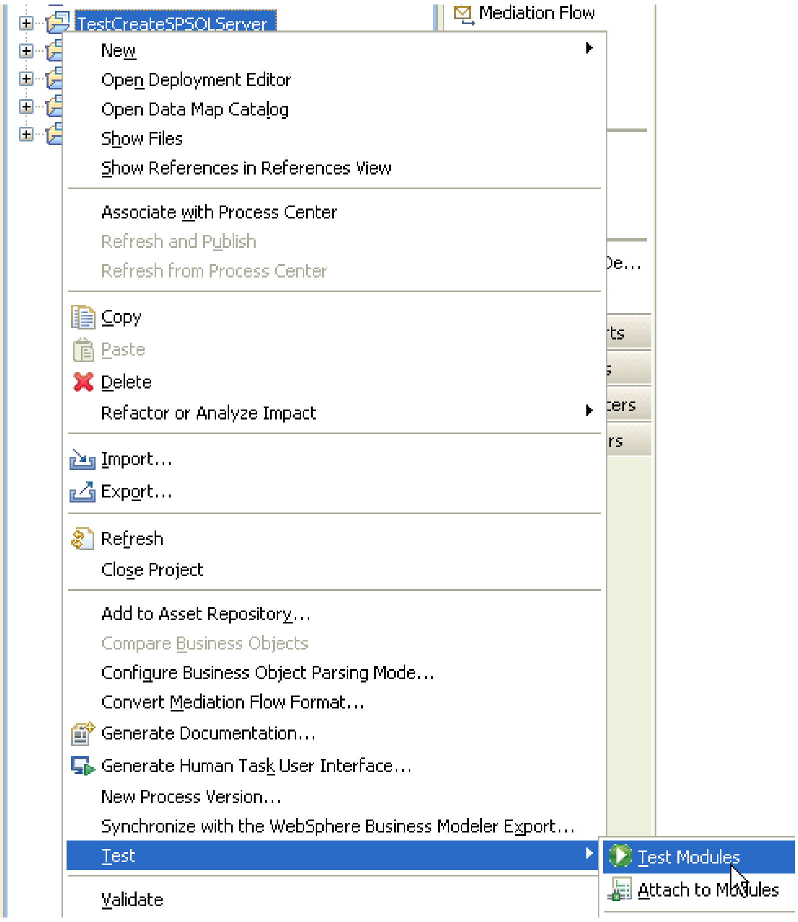
- *TestCreateSPSQLServer*

☑ If server is started, publish changes immediately

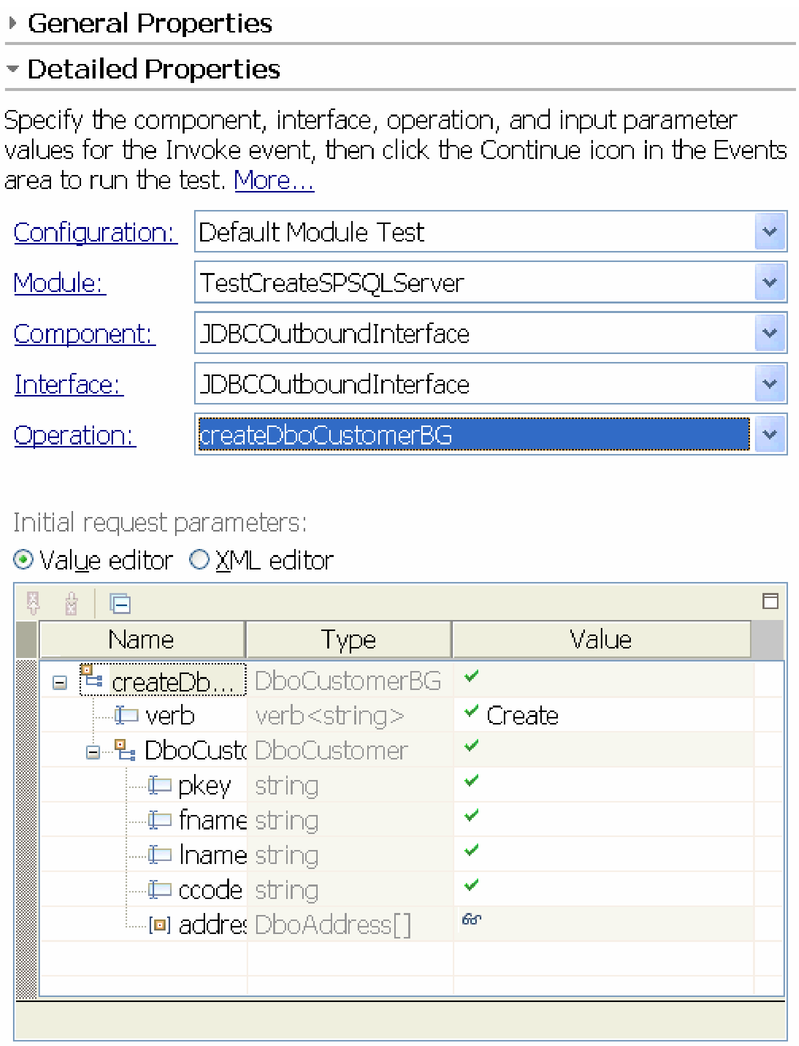
Test the assembled adapter application

Test the assembled adapter application using the IBM Integration Designer 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.



▸ 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: TestCreateSPSQLServer

Component: JDBCOutboundInterface

Interface: JDBCOutboundInterface

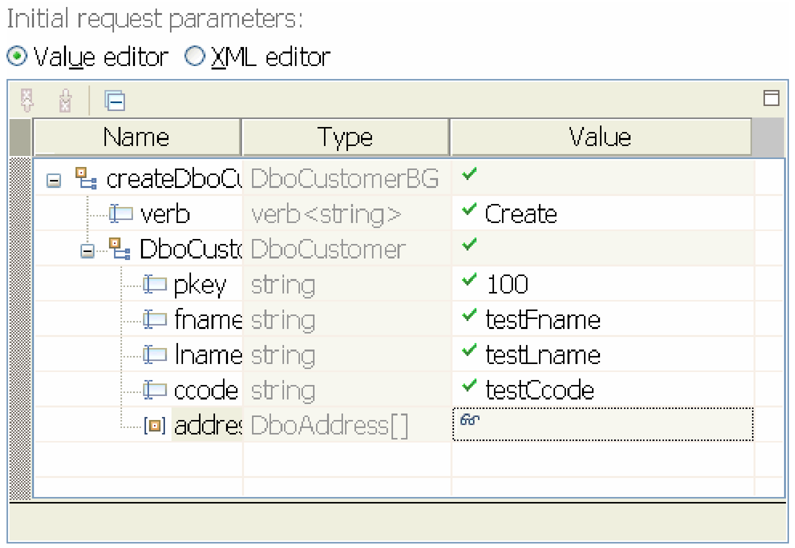
Operation: createDboCustomerBG

Initial request parameters:

● Value editor ○ XML editor

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

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

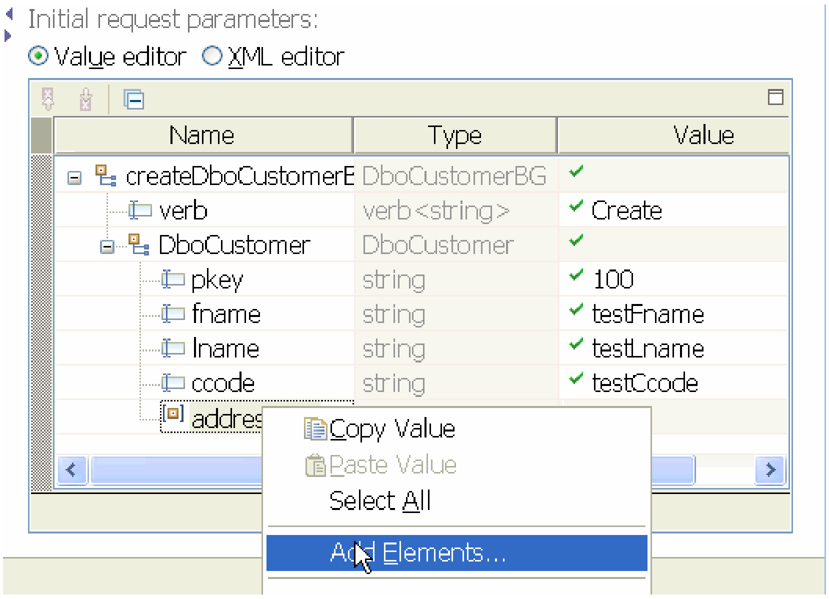


Initial request parameters:

● Value editor ○ XML editor

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

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

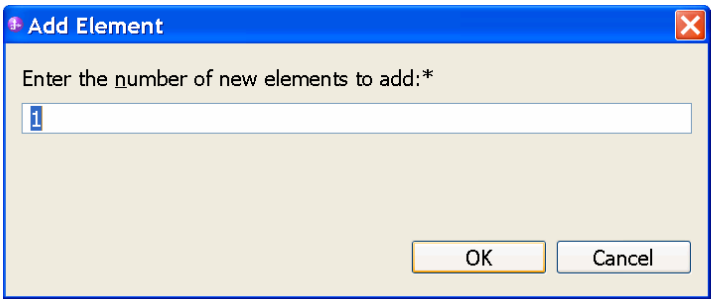


Initial request parameters:

● 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
addres		

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

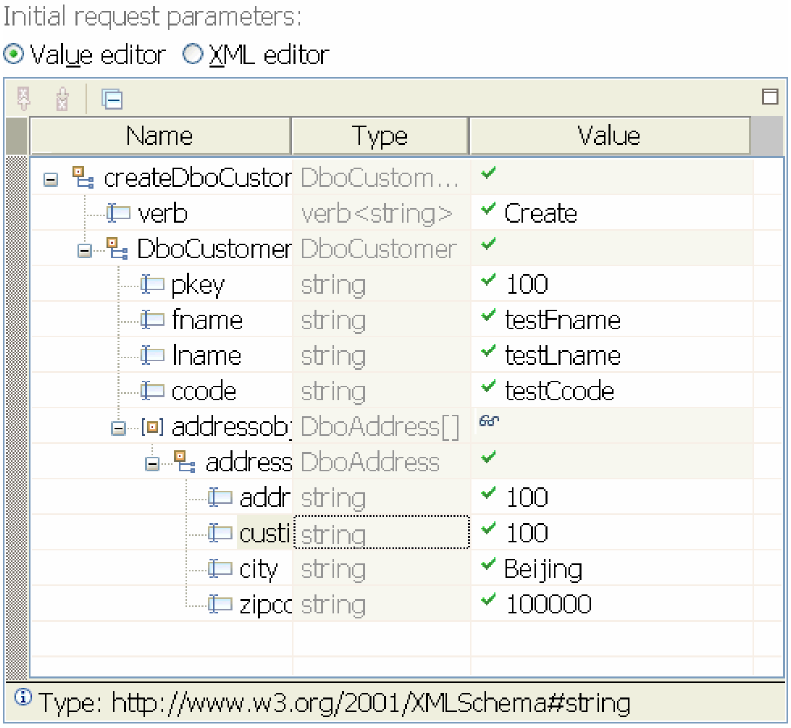


Add Element

Enter the number of new elements to add:*

1

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



Initial request parameters:

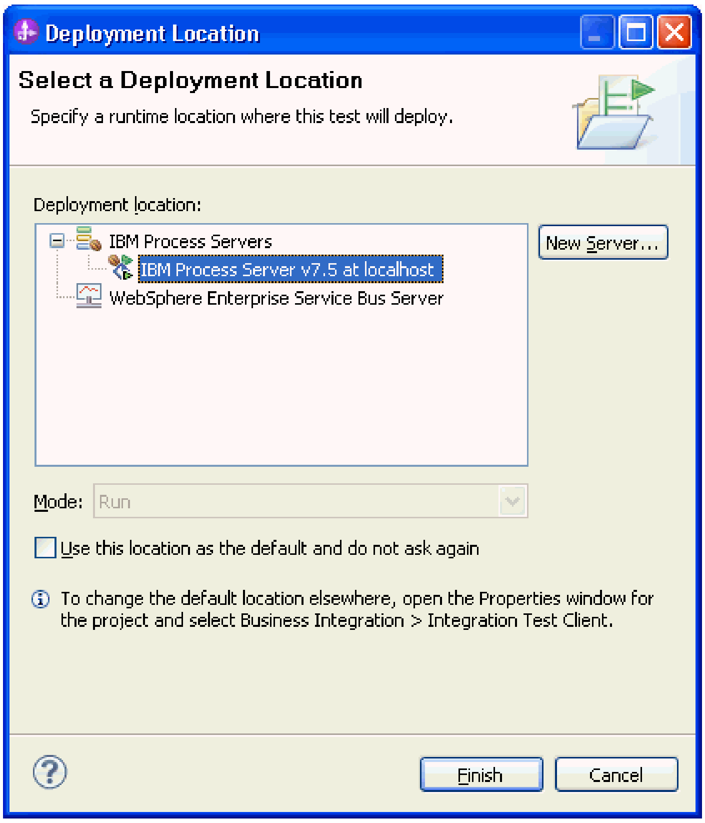
● Value editor ○ XML editor

Name	Type	Value
createDboCustor	DboCustom...	✓
verb	verb<string>	✓ Create
DboCustomer	DboCustomer	✓
pkey	string	✓ 100
fname	string	✓ testFname
lname	string	✓ testLname
ccode	string	✓ testCcode
addressob	DboAddress[]	
address	DboAddress	✓
addr	string	✓ 100
custi	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**.



Select a Deployment Location

Specify a runtime location where this test will deploy.

Deployment location:

- IBM Process Servers
  - IBM Process Server v7.5 at localhost
- WebSphere Enterprise Service Bus Server

Mode: Run

□ Use this location as the default and do not ask again

To change the default location elsewhere, open the Properties window for the project and select Business Integration > Integration Test Client.

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

Clear the sample content

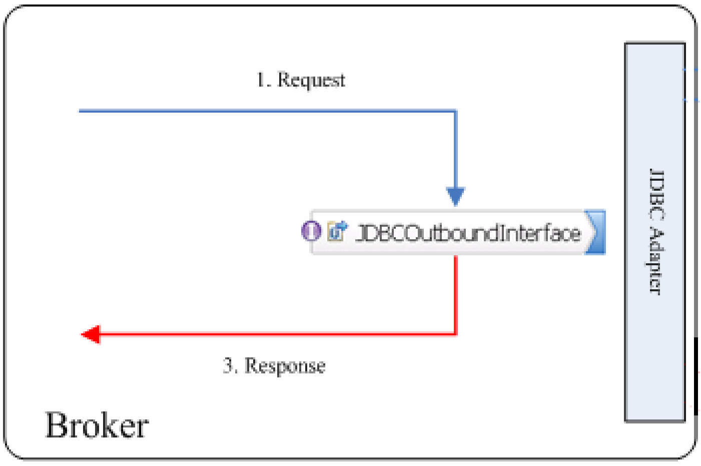
After a record created with the IBM Integration Designer environment, it can be remove using a Delete operation.

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

This tutorial demonstrates how WebSphere Adapter for JDBC 7.5.0.0 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.

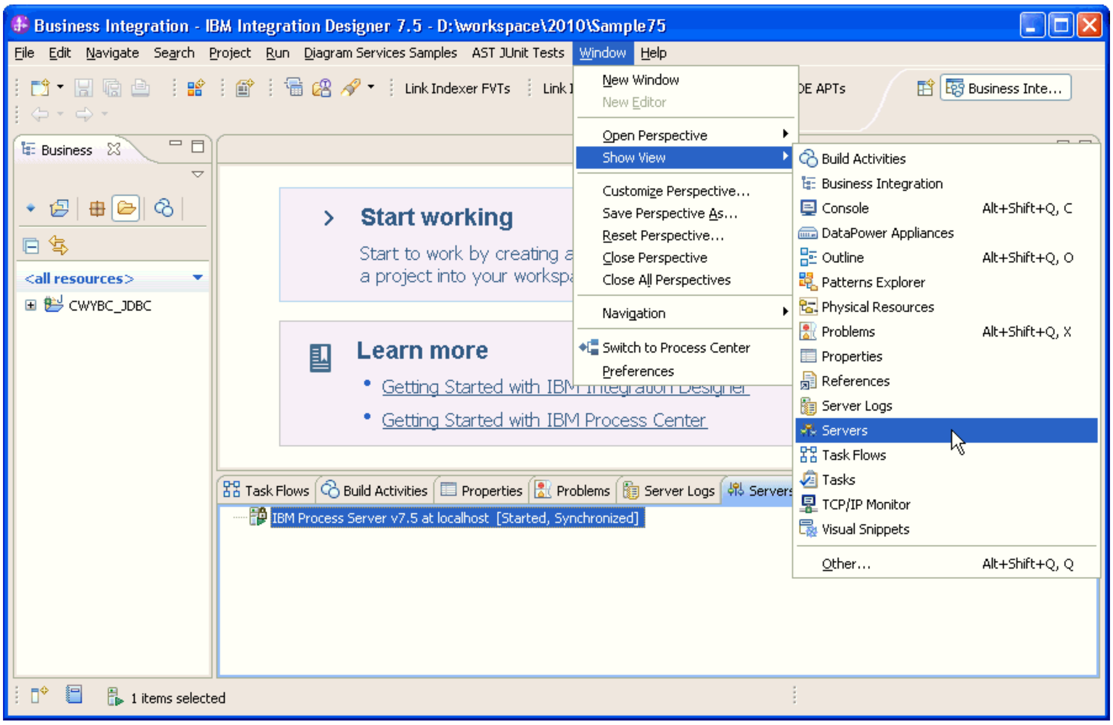
```
CREATE OR REPLACE PROCEDURE SAMPLE_ARRAY_STRUCT (
pkey IN VARCHAR, arr IN ARRAYTYPE, strt OUT STRUCTTYPE
)
IS BEGIN
INSERT INTO TABLE_ARRAY VALUES (pkey, arr);
SELECT INFO INTO strt FROM TABLE_STRUCT WHERE ID =
pkey;
END SAMPLE_ARRAY_STRUCT;
/
```

Create an authentication alias

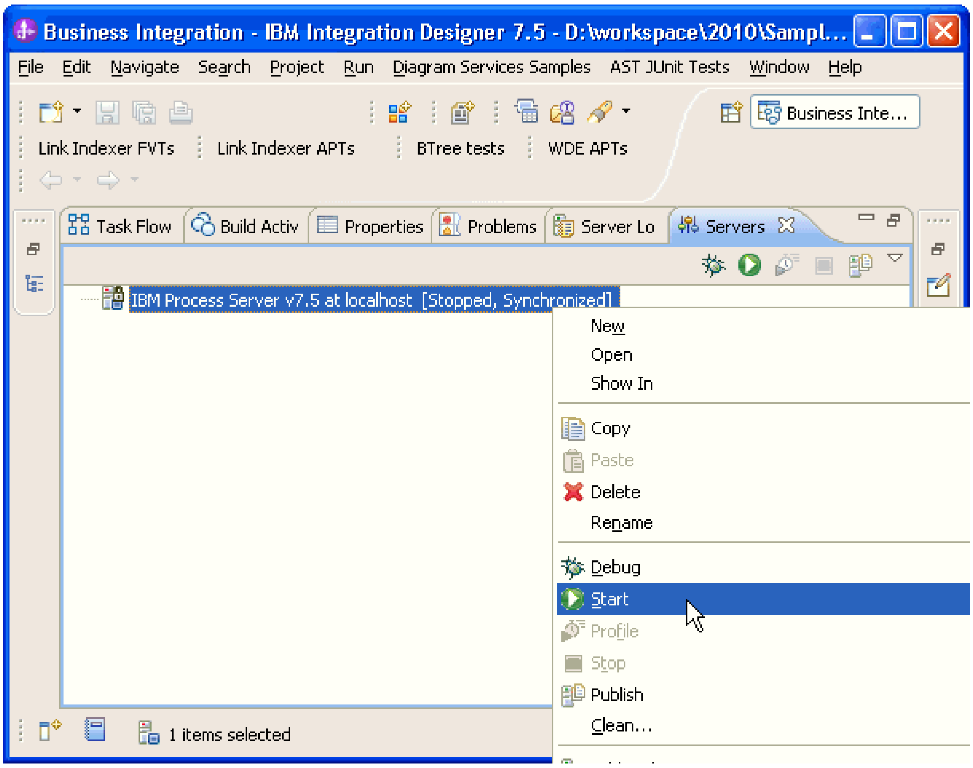
The authentication alias needs to be set because the data source created in the next section uses the username and password set in the authentication alias to connect to the database.

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

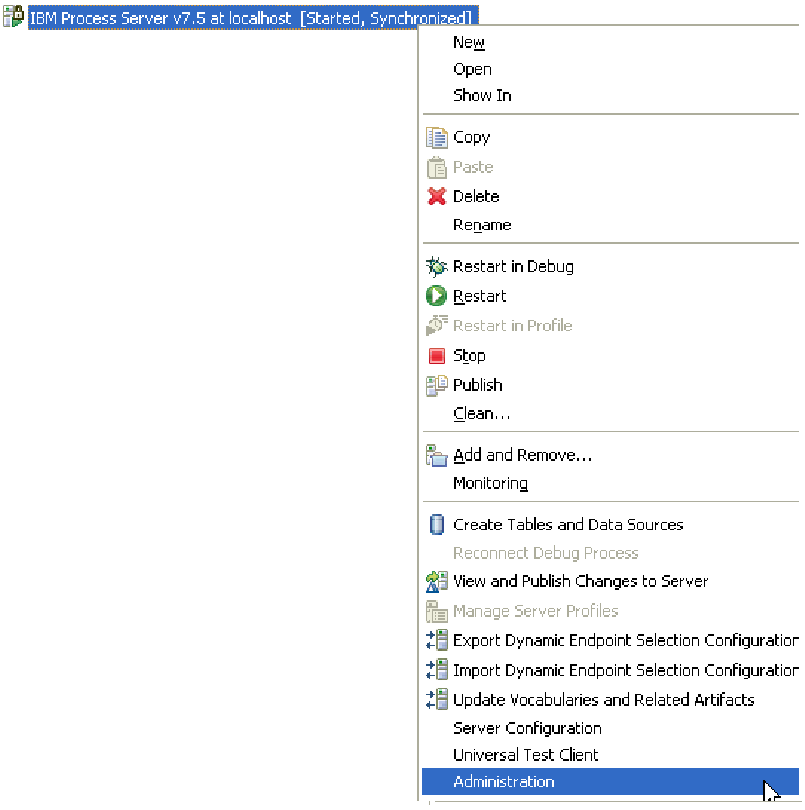
9. In IBM Integration Designer, switch to the **Servers** view by selecting **Windows > Show View > Servers**.



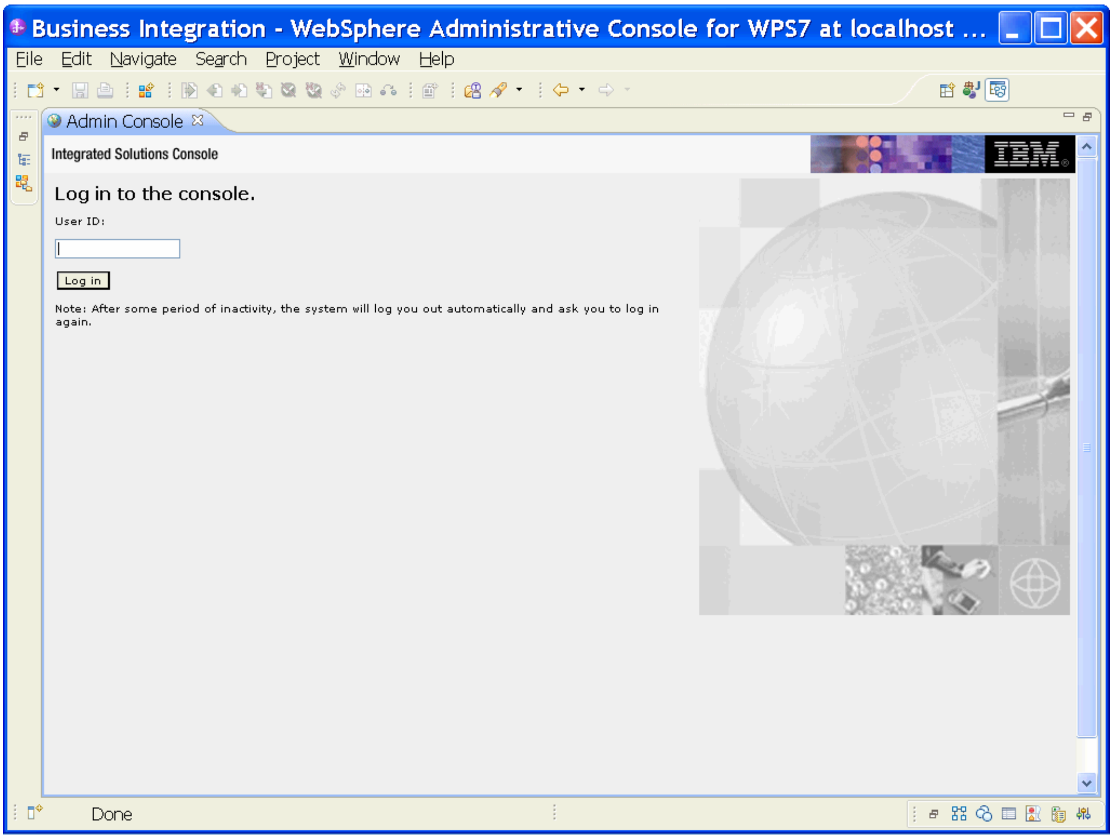
10. In the **Servers** view, right-click the server that you want to start and select **Start**.


Business Integration - IBM Integration Designer 7.5 - D:\workspace\2010\Sampl...
File Edit Navigate Search Project Run Diagram Services Samples AST JUnit Tests Window Help
Business Inte...
Link Indexer FVTs
Link Indexer APTs
BTree tests
WDE APTs
Task Flow
Build Activ
Properties
Problems
Server Lo
Servers
IBM Process Server v7.5 at localhost [Stopped, Synchronized]
New
Open
Show In
Copy
Paste
Delete
Rename
Debug
Start
Profile
Stop
Publish
Clean...
1 items selected

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


IBM Process Server v7.5 at localhost [Started, Synchronized]
New
Open
Show In
Copy
Paste
Delete
Rename
Restart in Debug
Restart
Restart in Profile
Stop
Publish
Clean...
Add and Remove...
Monitoring
Create Tables and Data Sources
Reconnect Debug Process
View and Publish Changes to Server
Manage Server Profiles
Export Dynamic Endpoint Selection Configuratior
Import Dynamic Endpoint Selection Configuratior
Update Vocabularies and Related Artifacts
Server Configuration
Universal Test Client
Administration

12. Log on to the administrative console.


Business Integration - WebSphere Administrative Console for WPS7 at localhost ...
File
Edit
Navigate
Search
Project
Window
Help
Admin Console
Integrated Solutions Console
IBM
Done

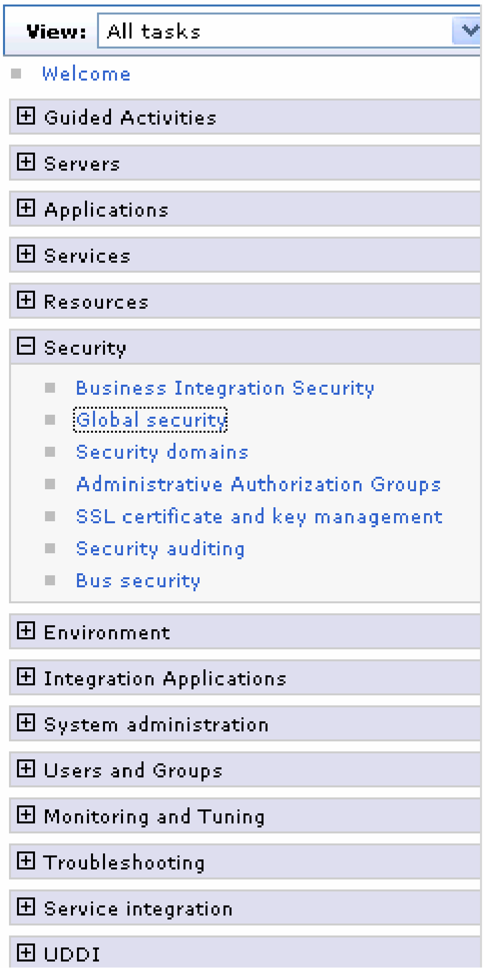
Log in to the console.

User ID:

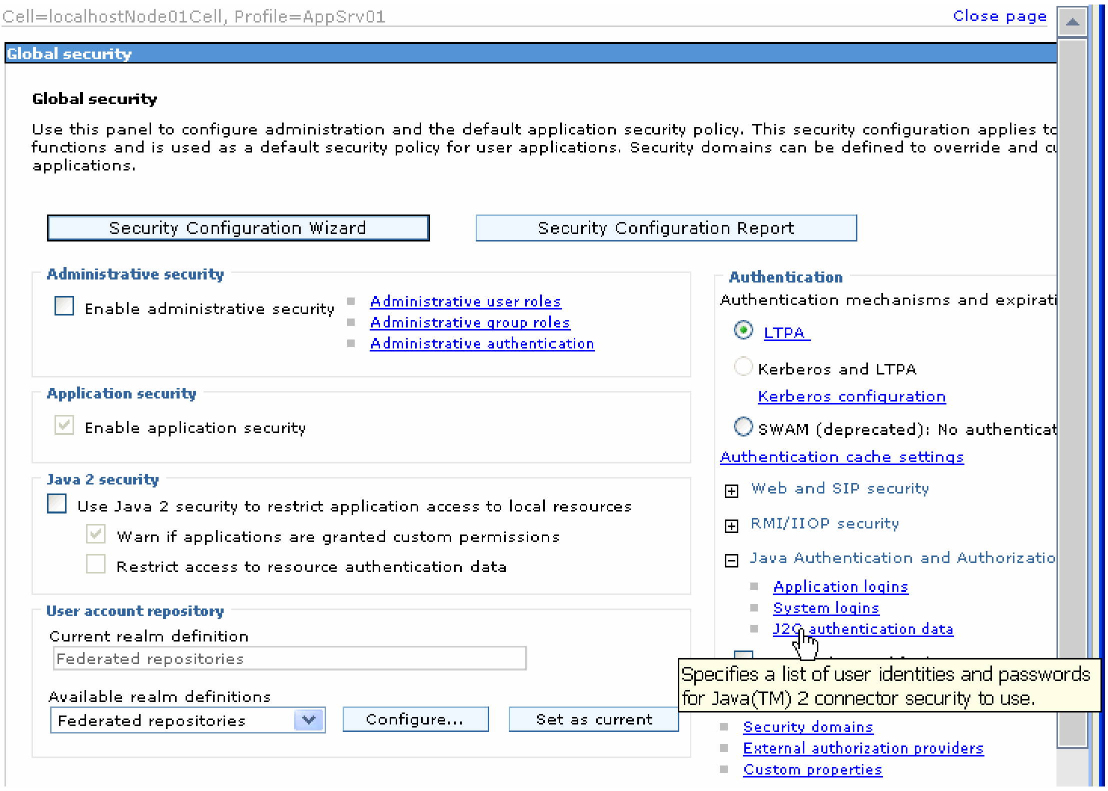
Log in

Note: After some period of inactivity, the system will log you out automatically and ask you to log in again.

13. Click **Security → Global security.**



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



Cell=localhostNode01Cell, Profile=AppSrv01

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 cu applications.

Administrative security

□ Enable administrative security

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 expirati

● LTPA

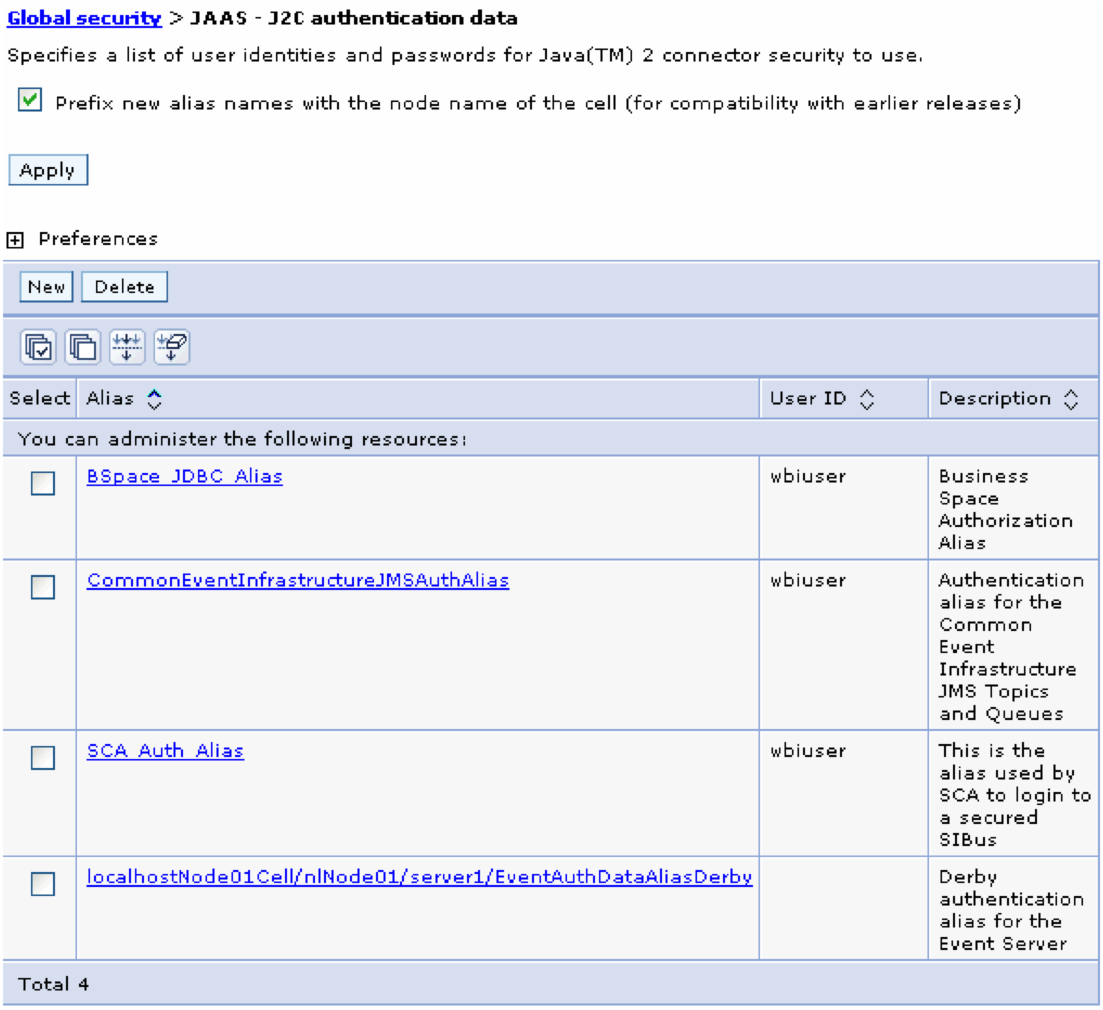
○ Kerberos and LTPA

○ SWAM (deprecated): No authenticat

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

Close page
Security Configuration Wizard
Security Configuration Report
Administrative user roles
Administrative group roles
Administrative authentication
Configure...
Set as current
Kerberos configuration
Authentication cache settings
Web and SIP security
RMI/IIOP security
Java Authentication and Authorizatio
Application logins
System logins
J2C authentication data
Security domains
External authorization providers
Custom properties

A list of existing aliases is displayed.



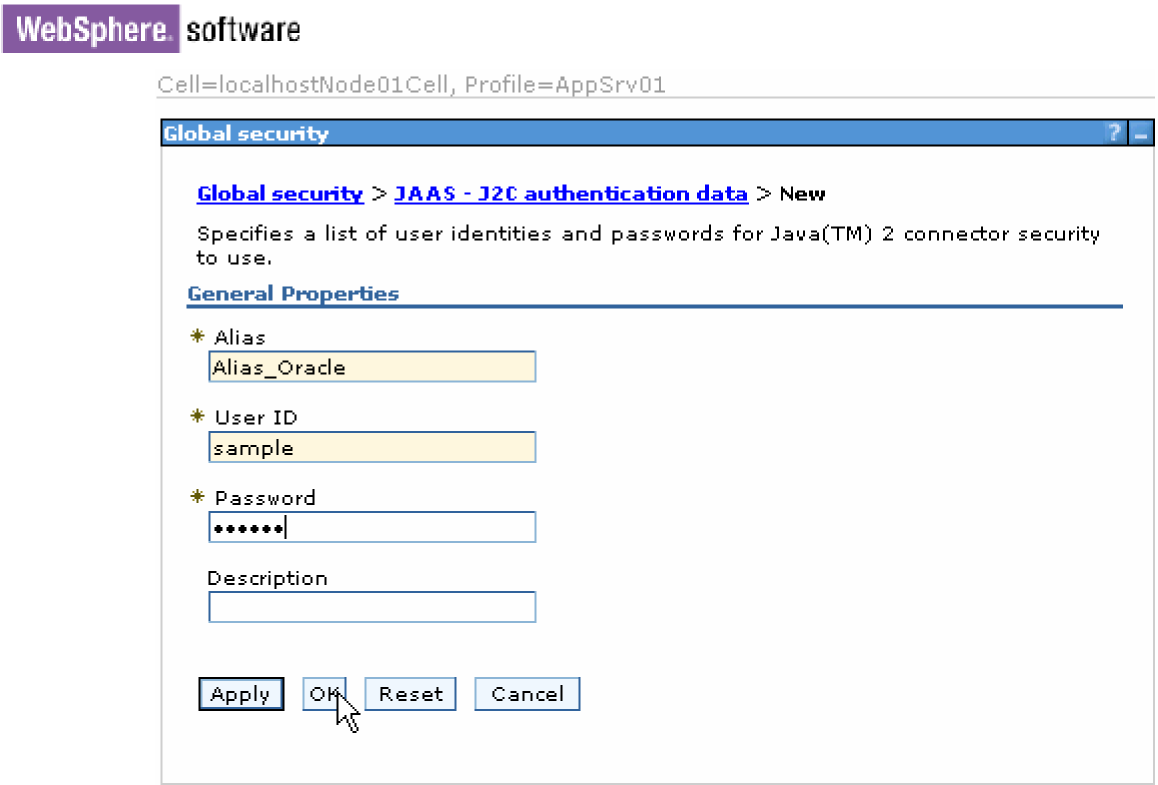
Global security > JAAS - J2C authentication data

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)

Select	Alias	User ID	Description
You can administer the following resources:			
□	BSpace_JDBC_Alias	wbiuser	Business Space Authorization Alias
□	CommonEventInfrastructureJMSAuthAlias	wbiuser	Authentication alias for the Common Event Infrastructure JMS Topics and Queues
□	SCA_Auth_Alias	wbiuser	This is the alias used by SCA to login to a secured SIBus
□	localhostNode01Cell/nlNode01/server1/EventAuthDataAliasDerby		Derby authentication alias for the Event Server
Total 4			

15. 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

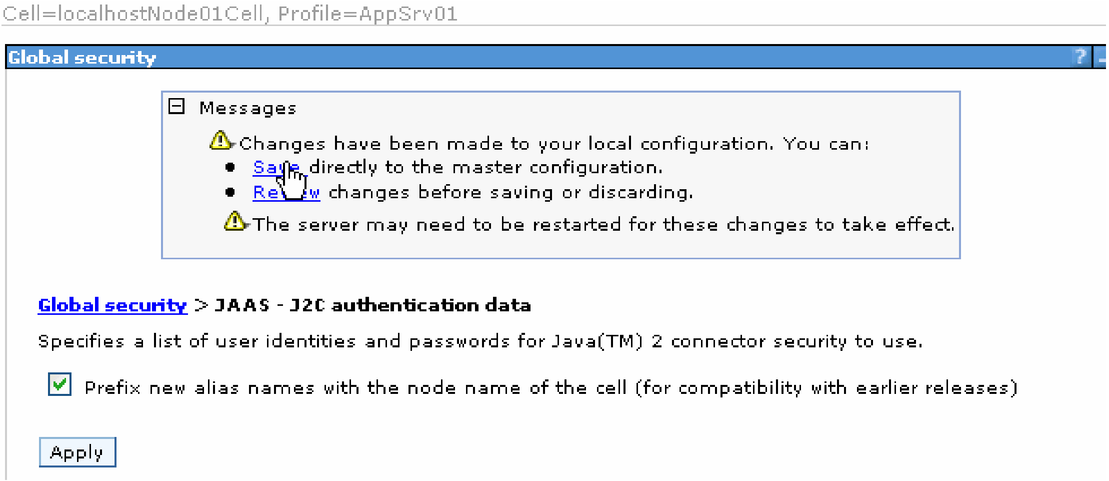
* Alias
Alias_Oracle

* User ID
sample

* Password
••••••

Description

16. 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.

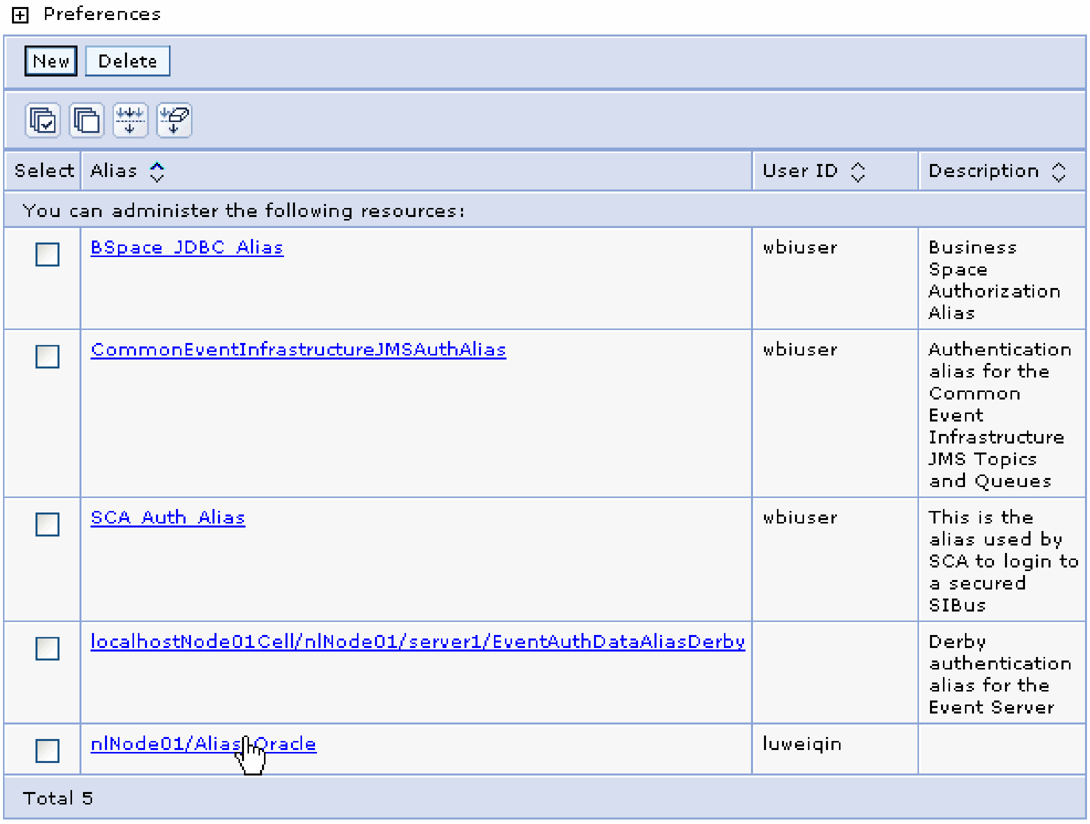
⚠ The server may need to be restarted for these changes to take effect.

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)

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



Select	Alias	User ID	Description
You can administer the following resources:			
□	BSpace_JDBC_Alias	wbiuser	Business Space Authorization Alias
□	CommonEventInfrastructureJMSAuthAlias	wbiuser	Authentication alias for the Common Event Infrastructure JMS Topics and Queues
□	SCA_Auth_Alias	wbiuser	This is the alias used by SCA to login to a secured SIBus
□	localhostNode01Cell/nlNode01/server1/EventAuthDataAliasDerby		Derby authentication alias for the Event Server
□	nlNode01/AliasIOracle	luweiqin	
Total 5			

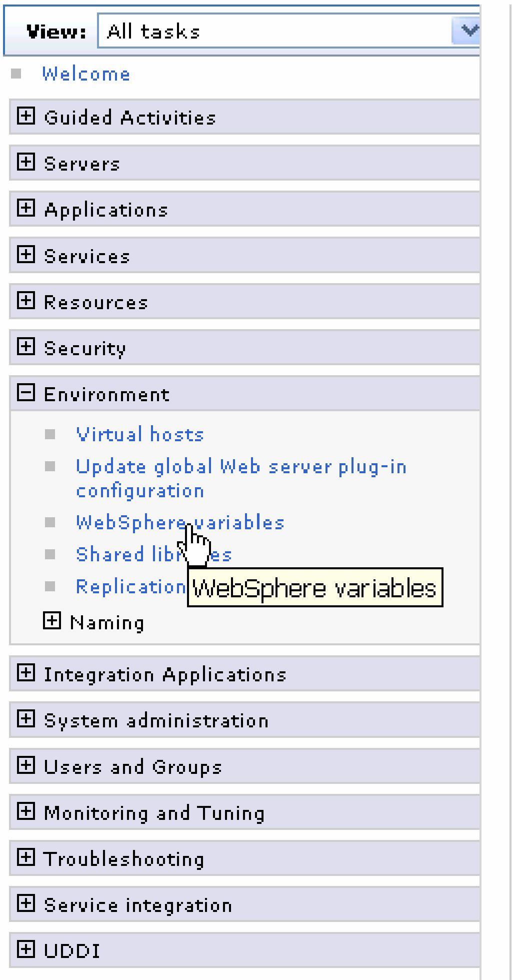
Create the data source

Create a data source in IBM 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.

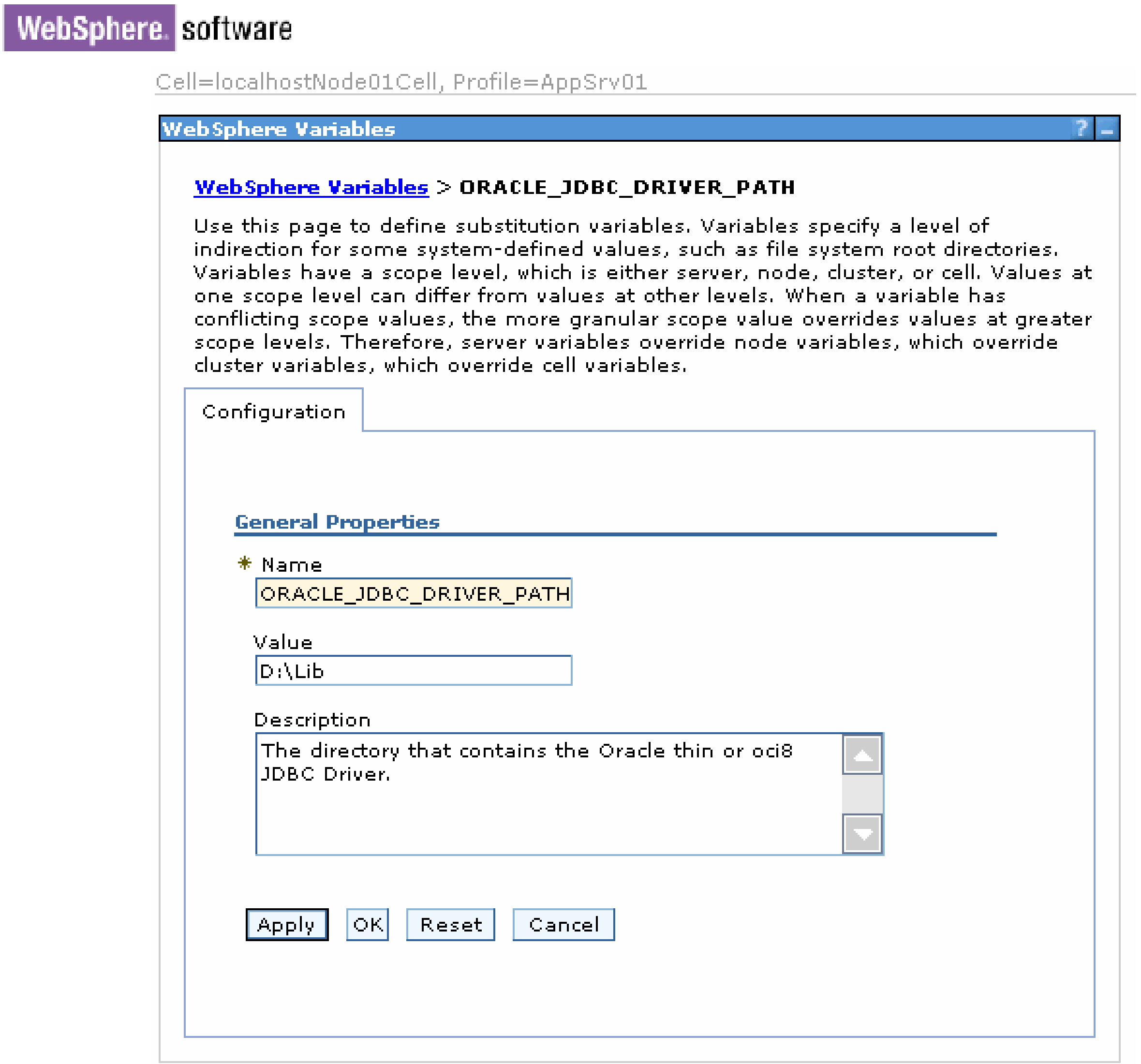
Note: This tutorial will use Oracle as the database and the Oracle thin driver, ojdbc6.jar.

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

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



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



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

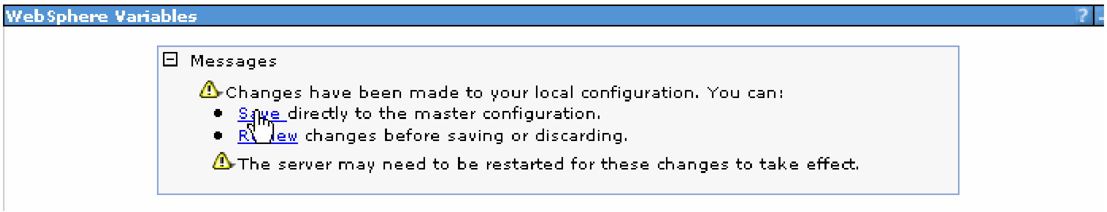
General Properties

* Name
ORACLE_JDBC_DRIVER_PATH

Value
D:\Lib

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

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

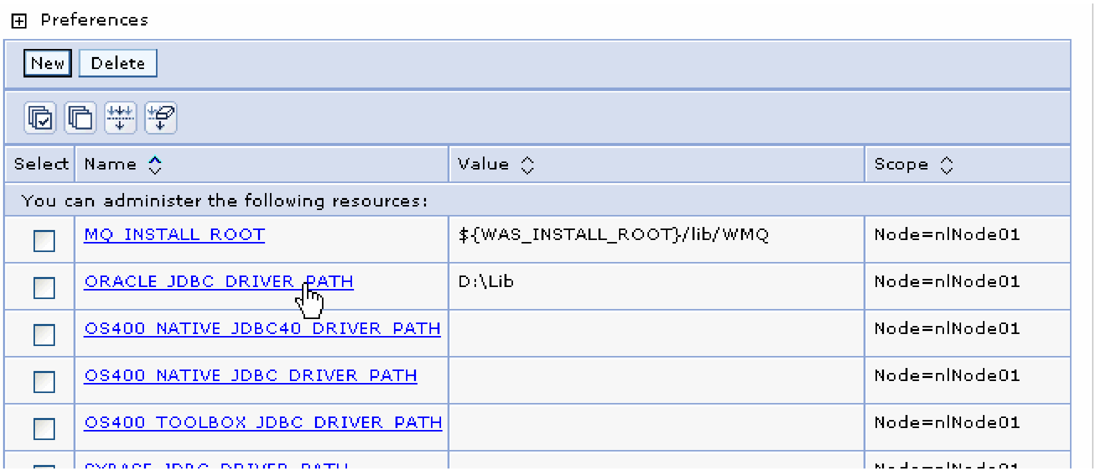

WebSphere Variables

Messages

⚠ Changes have been made to your local configuration. You can:
- Save directly to the master configuration.
- Review changes before saving or discarding.

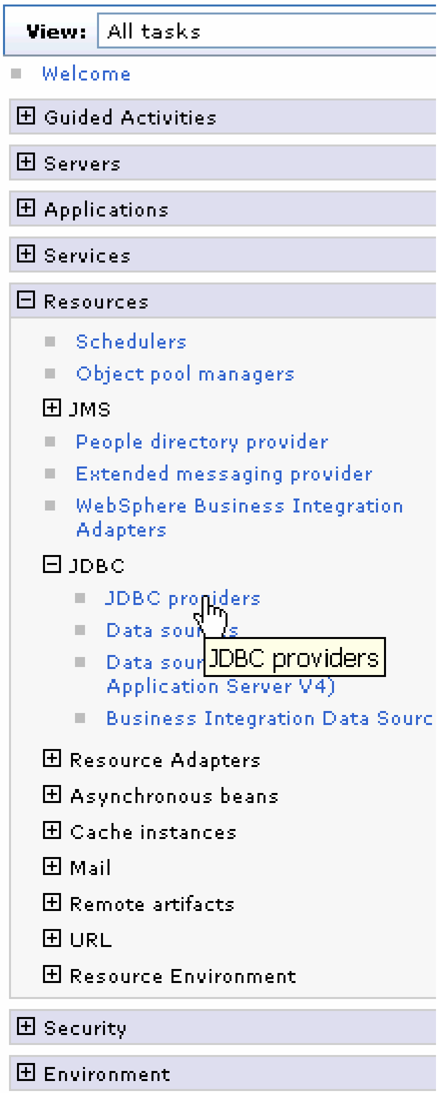
⚠ The server may need to be restarted for these changes to take effect.

The variable has been added and appears in the list.

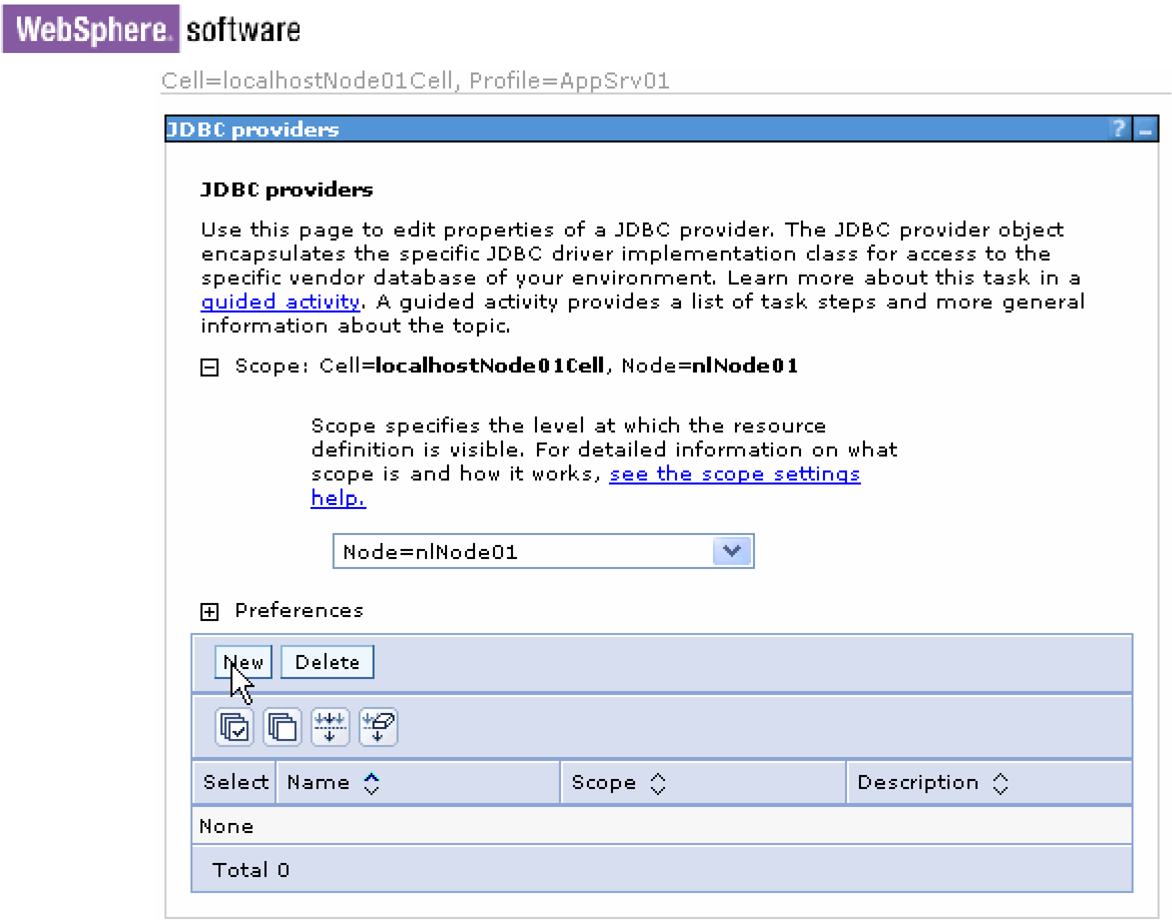


Select	Name	Value	Scope
You can administer the following resources:			
□	MQ_INSTALL_ROOT	${WAS_INSTALL_ROOT}/lib/WMQ	Node=nlNode01
□	ORACLE_JDBC_DRIVER_PATH	D:\Lib	Node=nlNode01
□	OS400_NATIVE_JDBC40_DRIVER_PATH		Node=nlNode01
□	OS400_NATIVE_JDBC_DRIVER_PATH		Node=nlNode01
□	OS400_TOOLBOX_JDBC_DRIVER_PATH		Node=nlNode01
□	SYBASE_JDBC_DRIVER_PATH		Node=nlNode01

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



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



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.

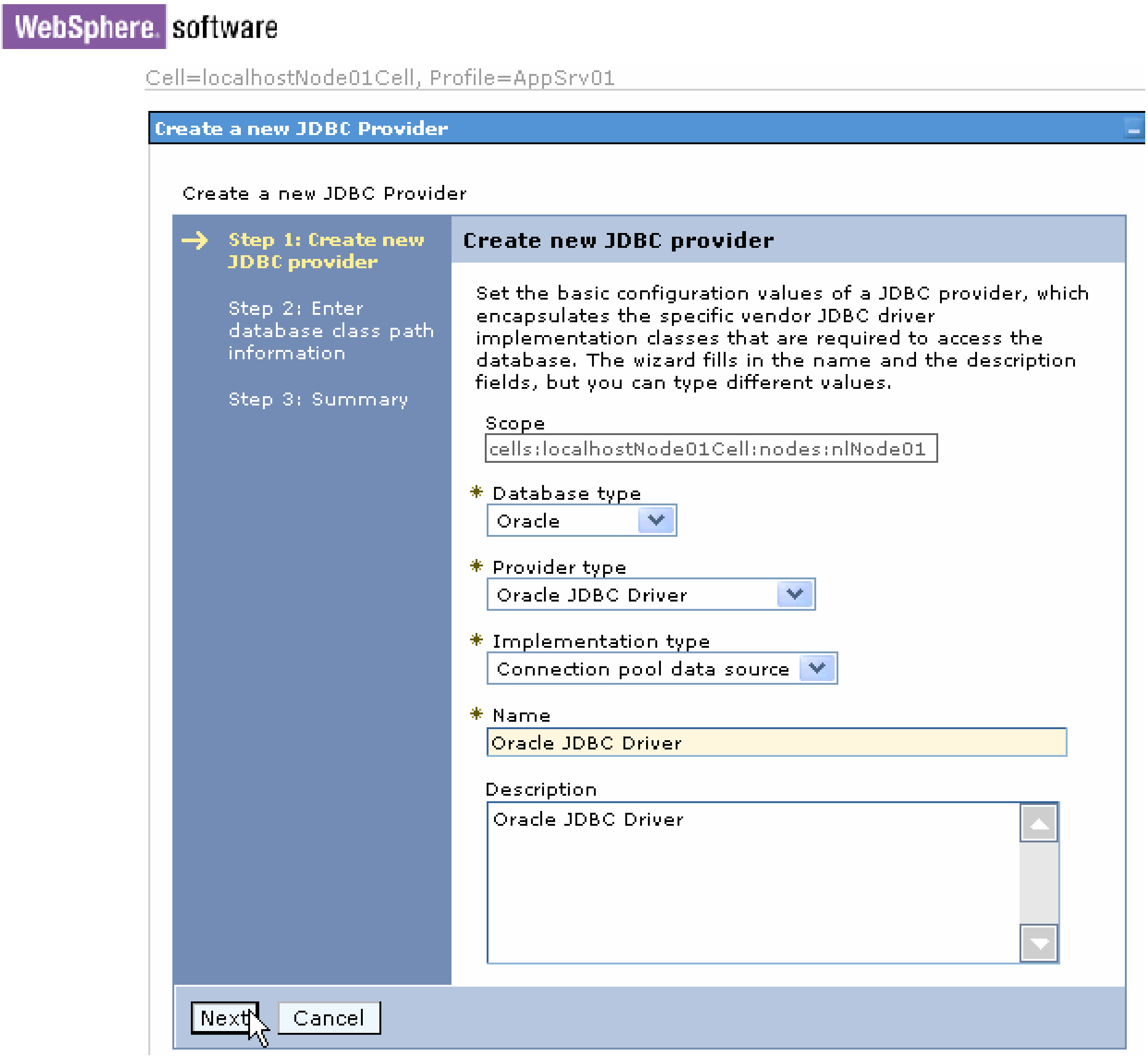
⊟ Scope: Cell=**localhostNode01Cell**, Node=**nlNode01**

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=nlNode01

Select	Name	Scope	Description
None			
Total 0			

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



Cell=localhostNode01Cell, Profile=AppSrv01

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:nlNode01

* Database type
Oracle

* Provider type
Oracle JDBC Driver

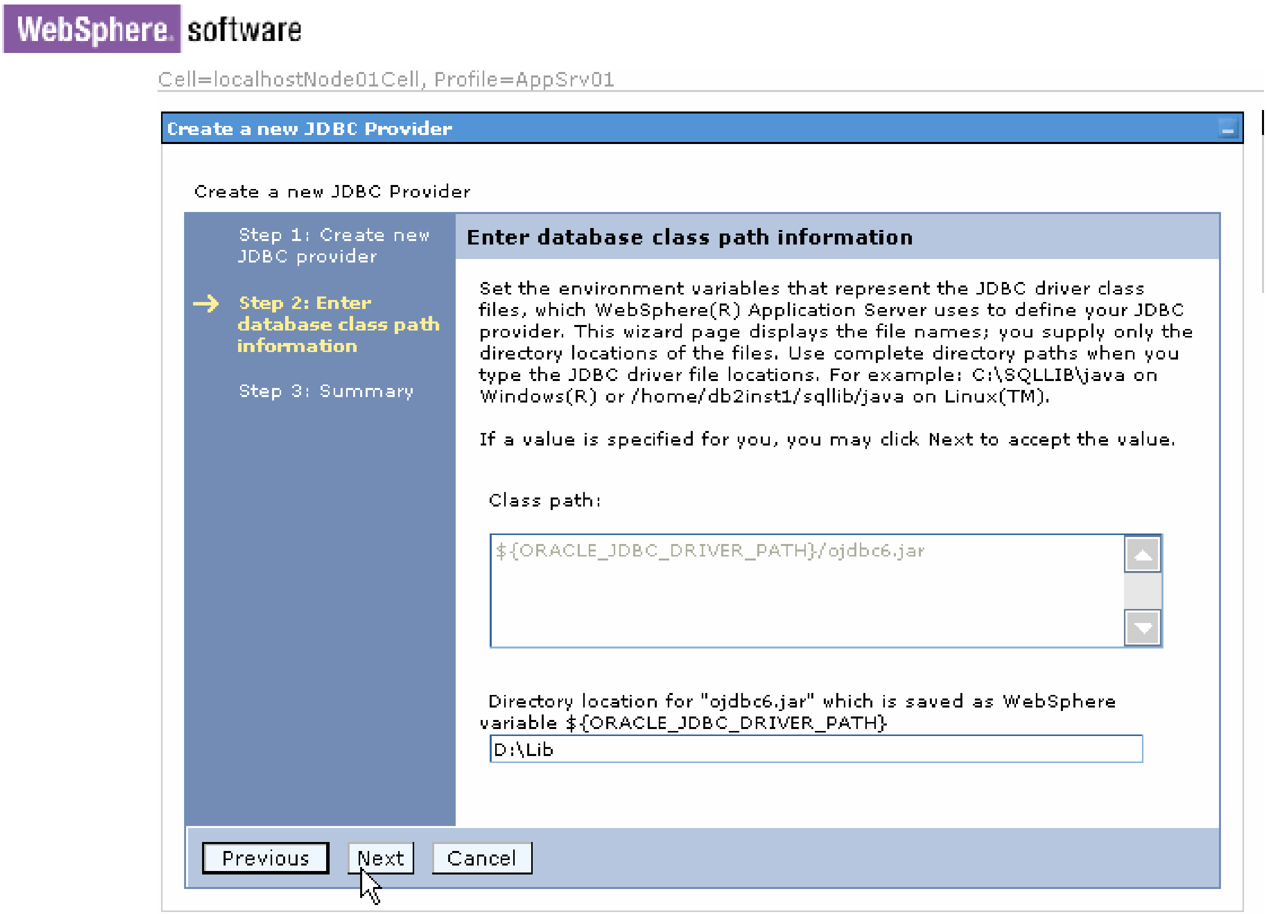
* Implementation type
Connection pool data source

* Name
Oracle JDBC Driver

Description
Oracle JDBC Driver

24. 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.

25. Click **Next**.



Cell=localhostNode01Cell, Profile=AppSrv01

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: 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.

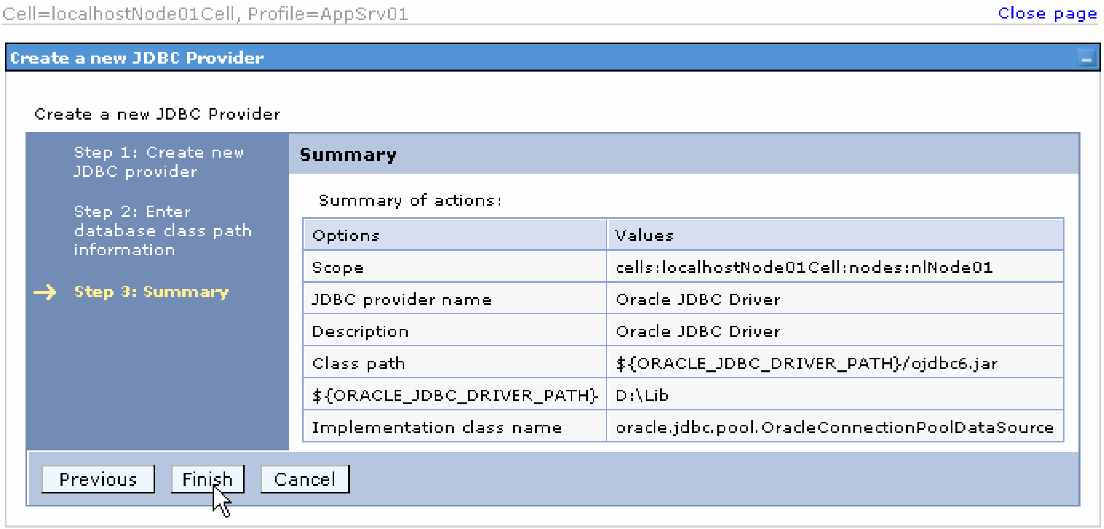
Class path:

${ORACLE_JDBC_DRIVER_PATH}/ojdbc6.jar

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

D:\Lib

26. Click **Finish**.



Cell=localhostNode01Cell, Profile=AppSrv01

Create a new JDBC Provider

Create a new JDBC Provider

Step 1: Create new JDBC provider

Step 2: Enter database class path information

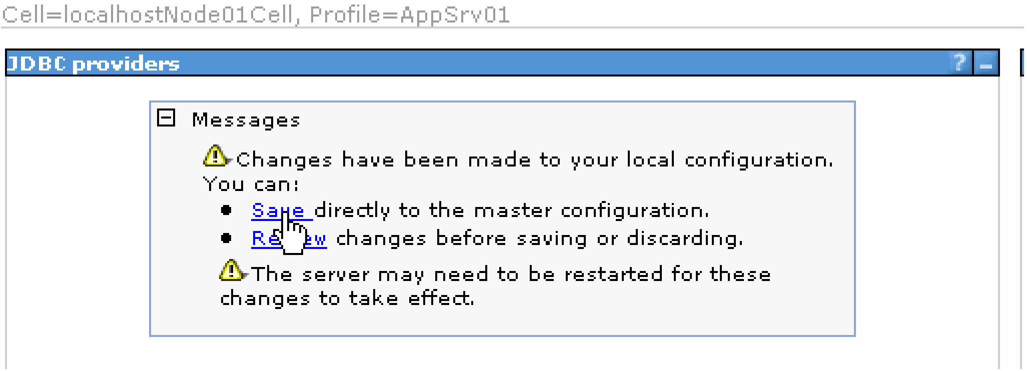
→ Step 3: Summary

Summary

Summary of actions:

Options	Values
Scope	cells:localhostNode01Cell:nodes:nlNode01
JDBC provider name	Oracle JDBC Driver
Description	Oracle JDBC Driver
Class path	${ORACLE_JDBC_DRIVER_PATH}/ojdbc6.jar
${ORACLE_JDBC_DRIVER_PATH}	D:\Lib
Implementation class name	oracle.jdbc.pool.OracleConnectionPoolDataSource

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



Cell=localhostNode01Cell, Profile=AppSrv01

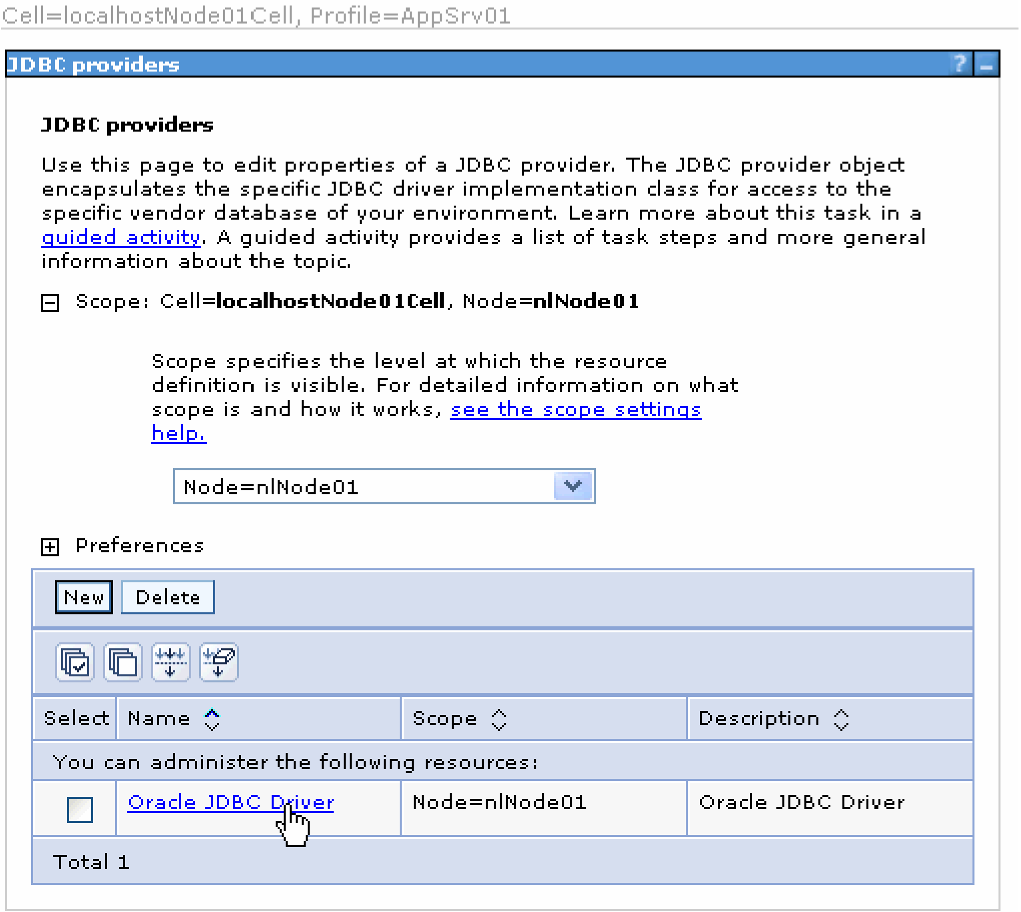
⊟ Messages

⚠ 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=**nlNode01**

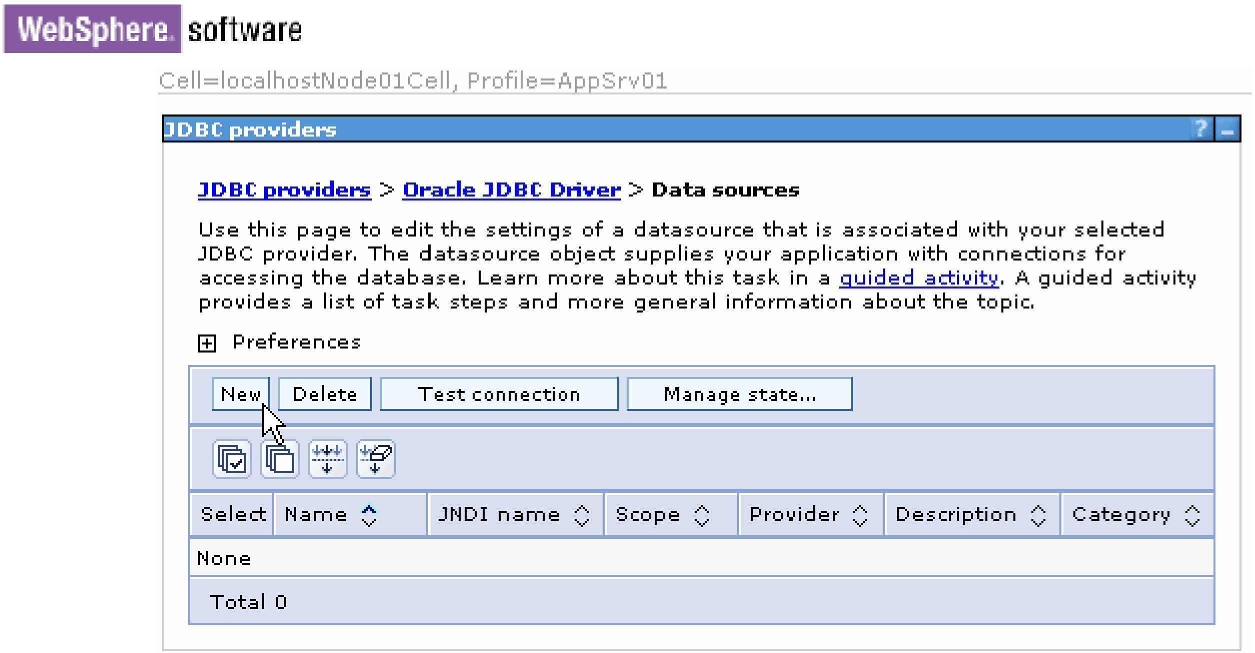
Scope specifies the level at which the resource definition is visible. For detailed information on what scope is and how it works, see the scope settings help.

Node=nlNode01

⊞ Preferences

Select	Name ⇕	Scope ⇕	Description ⇕
You can administer the following resources:			
□	Oracle JDBC Driver	Node=nlNode01	Oracle JDBC Driver
Total 1			

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



WebSphere. software

Cell=localhostNode01Cell, Profile=AppSrv01

JDBC providers

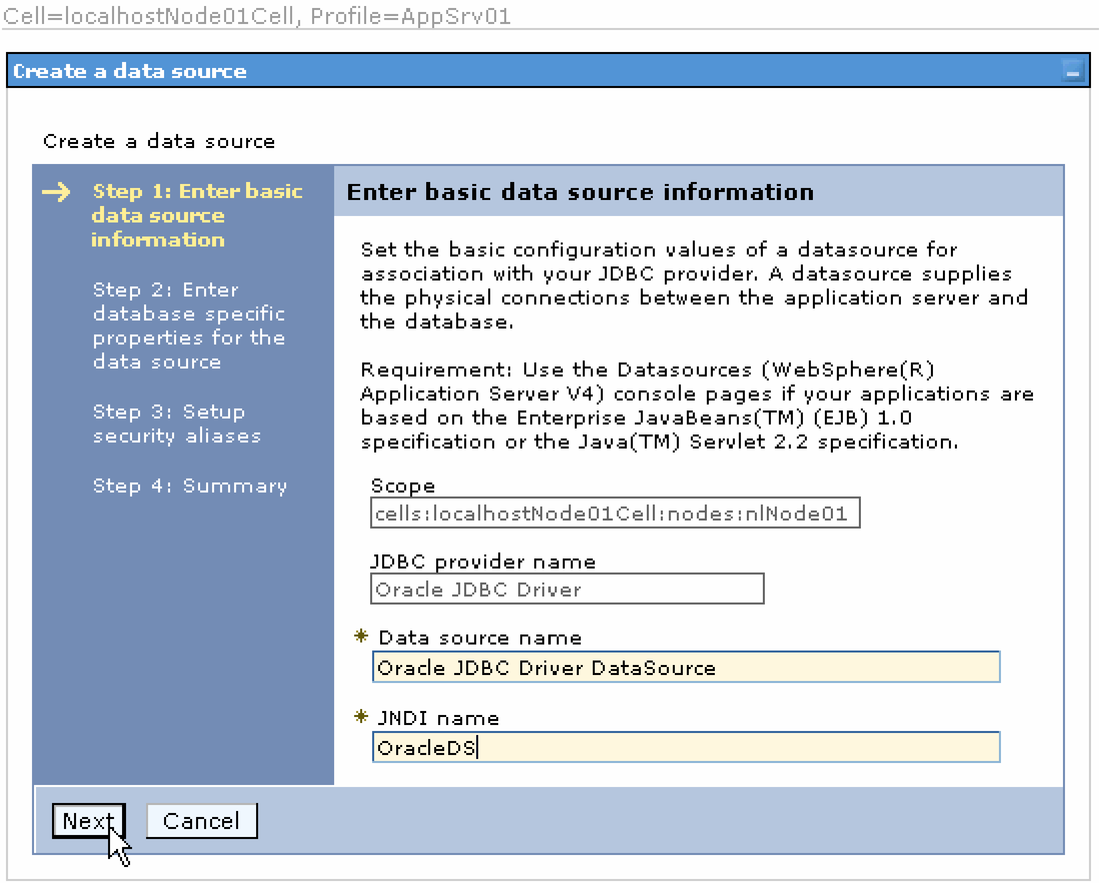
JDBC providers > Oracle JDBC Driver > Data sources

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

Select	Name	JNDI name	Scope	Provider	Description	Category
None						
Total 0						

29. 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.

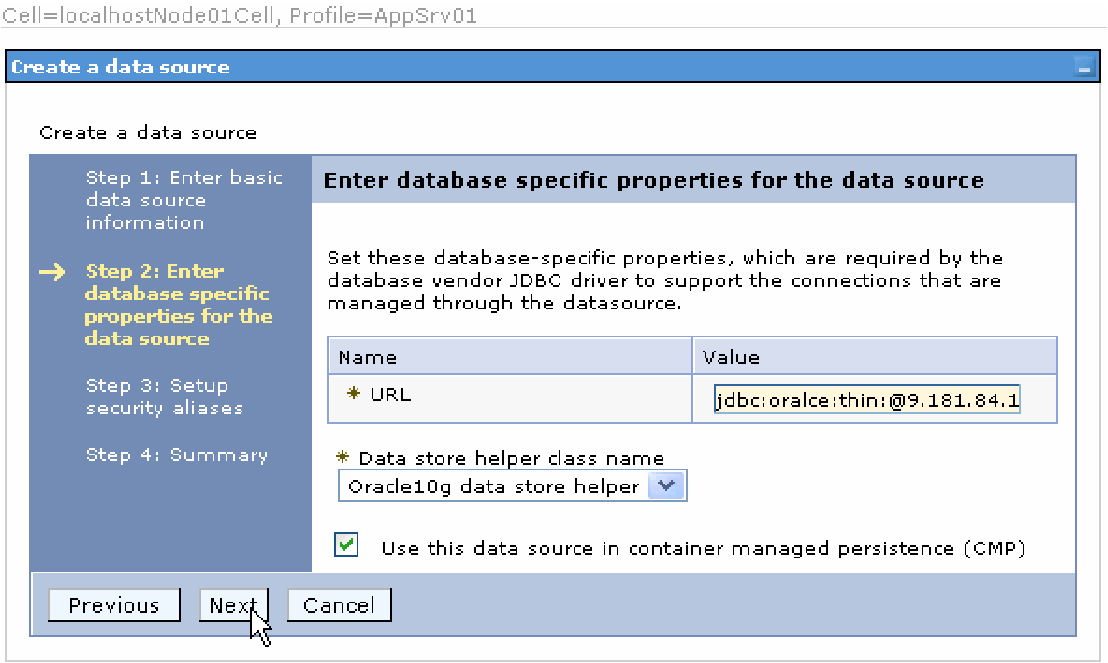
Scope
cells:localhostNode01Cell:nodes:nlNode01

JDBC provider name
Oracle JDBC Driver

* Data source name
Oracle JDBC Driver DataSource

* JNDI name
OracleDS

30. 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**.



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 database specific properties for the data source

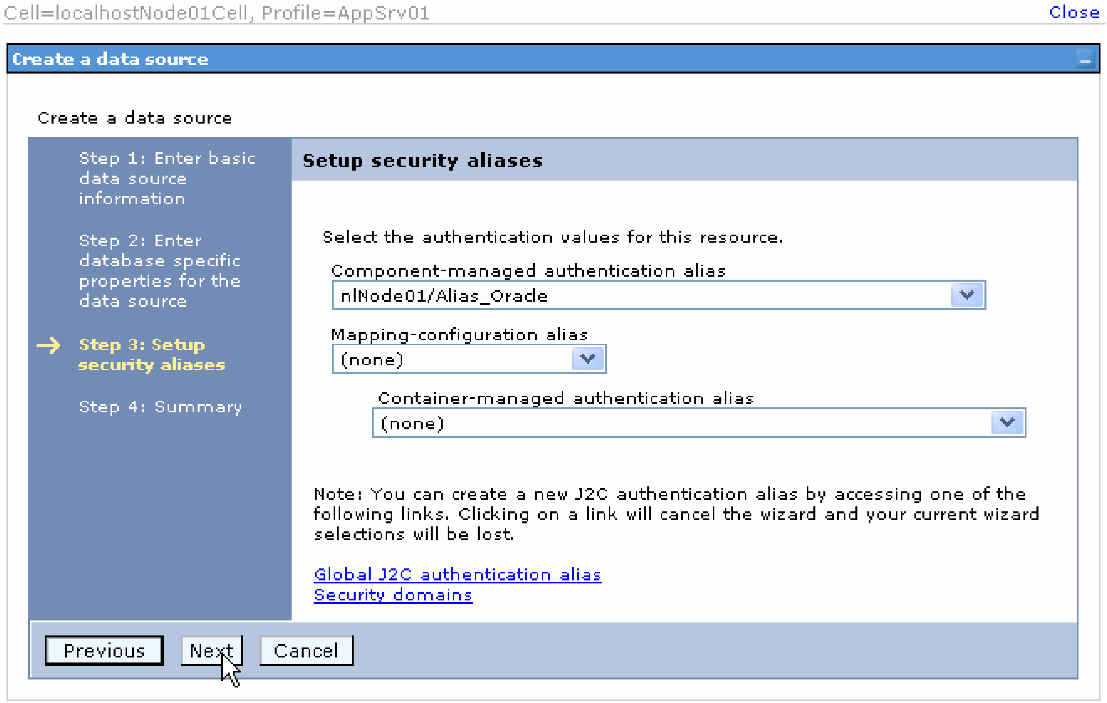
Set these database-specific properties, which are required by the database vendor JDBC driver to support the connections that are managed through the datasource.

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

* Data store helper class name
Oracle10g data store helper

☑ Use this data source in container managed persistence (CMP)

31. 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

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)

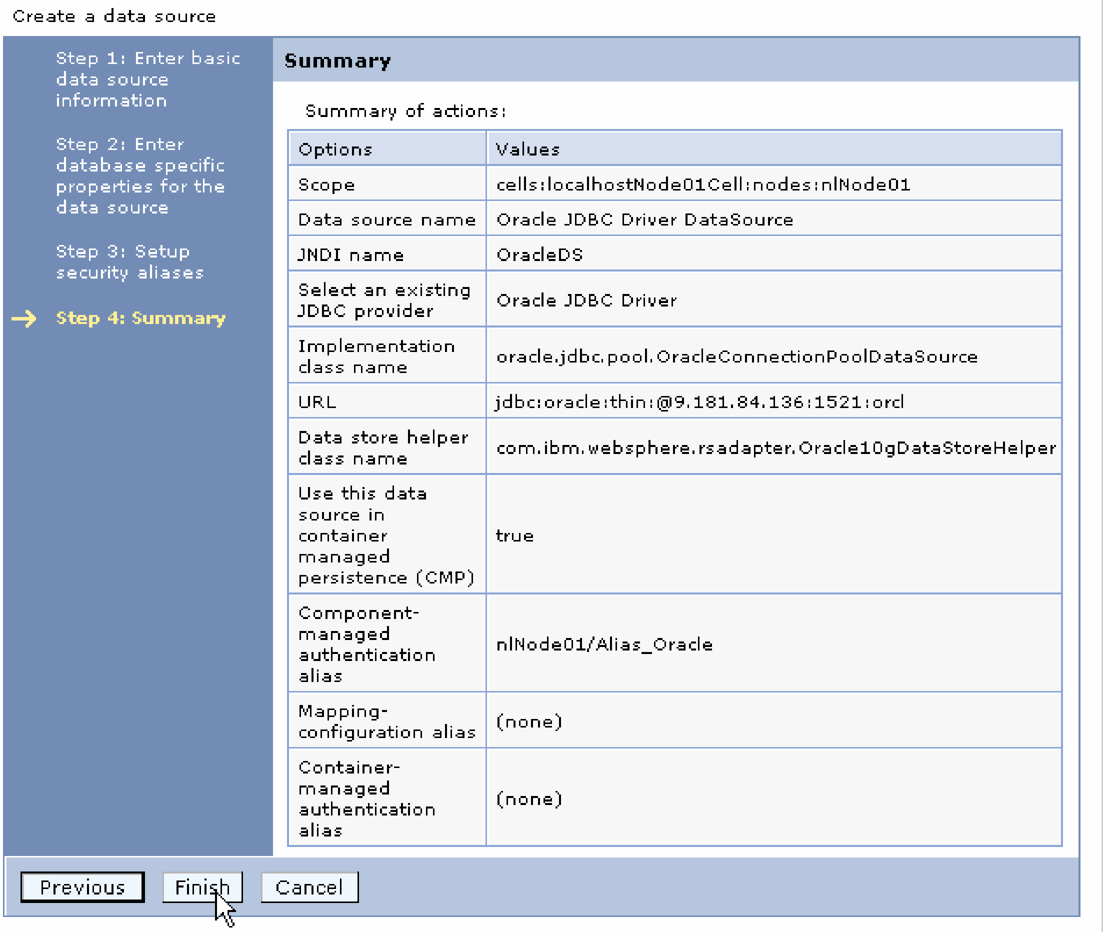
Container-managed authentication alias
(none)

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

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

32. Click **Finish**.



Create a data source

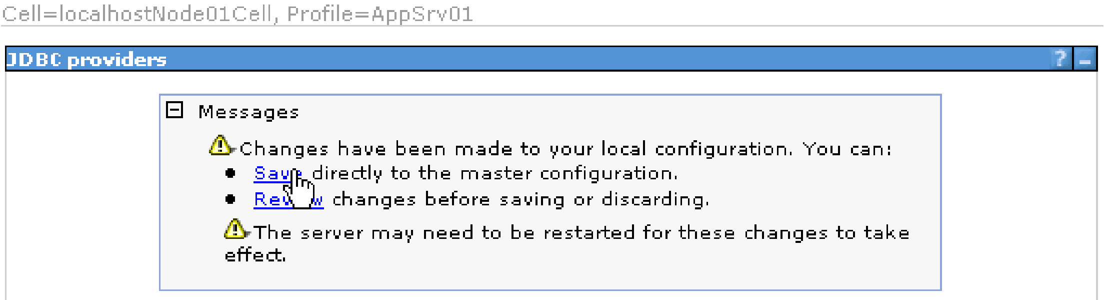
Step 1: Enter basic data source information
Step 2: Enter database specific properties for the data source
Step 3: Setup security aliases
Step 4: Summary

Summary

Summary of actions:

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:orcl
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/Alias_Oracle
Mapping-configuration alias	(none)
Container-managed authentication alias	(none)

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



Cell=localhostNode01Cell, Profile=AppSrv01

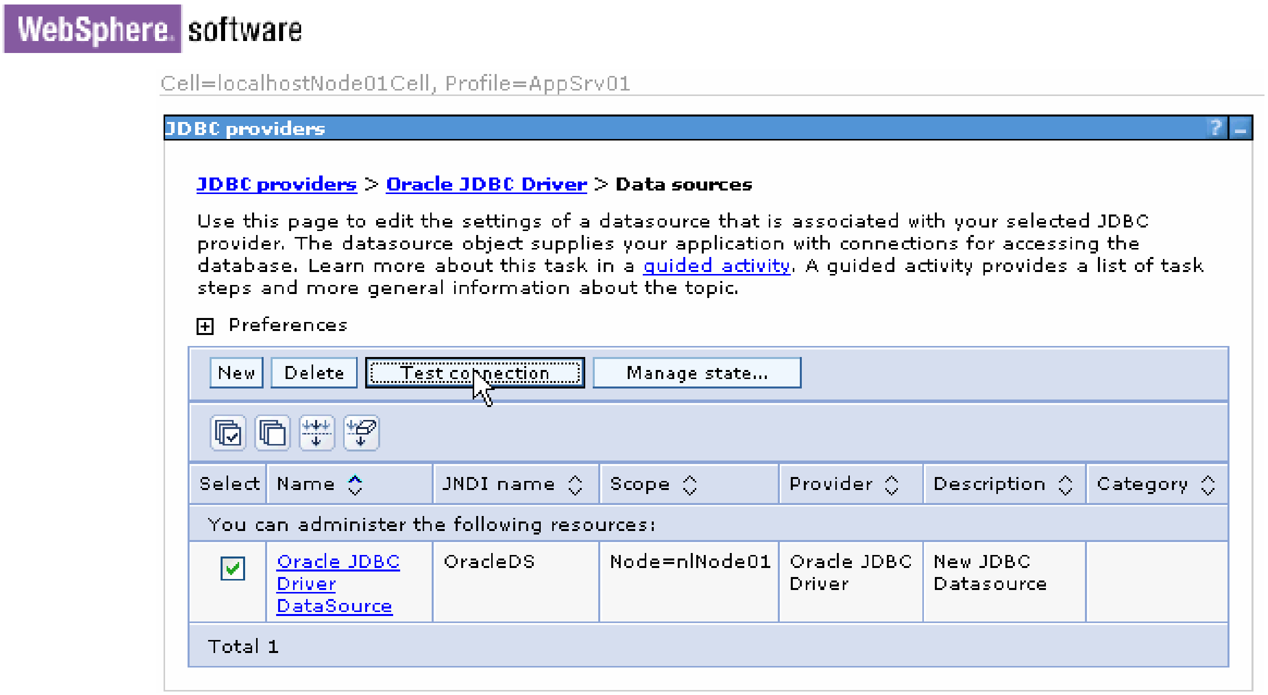
⊟ Messages

⚠ 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.

34. 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

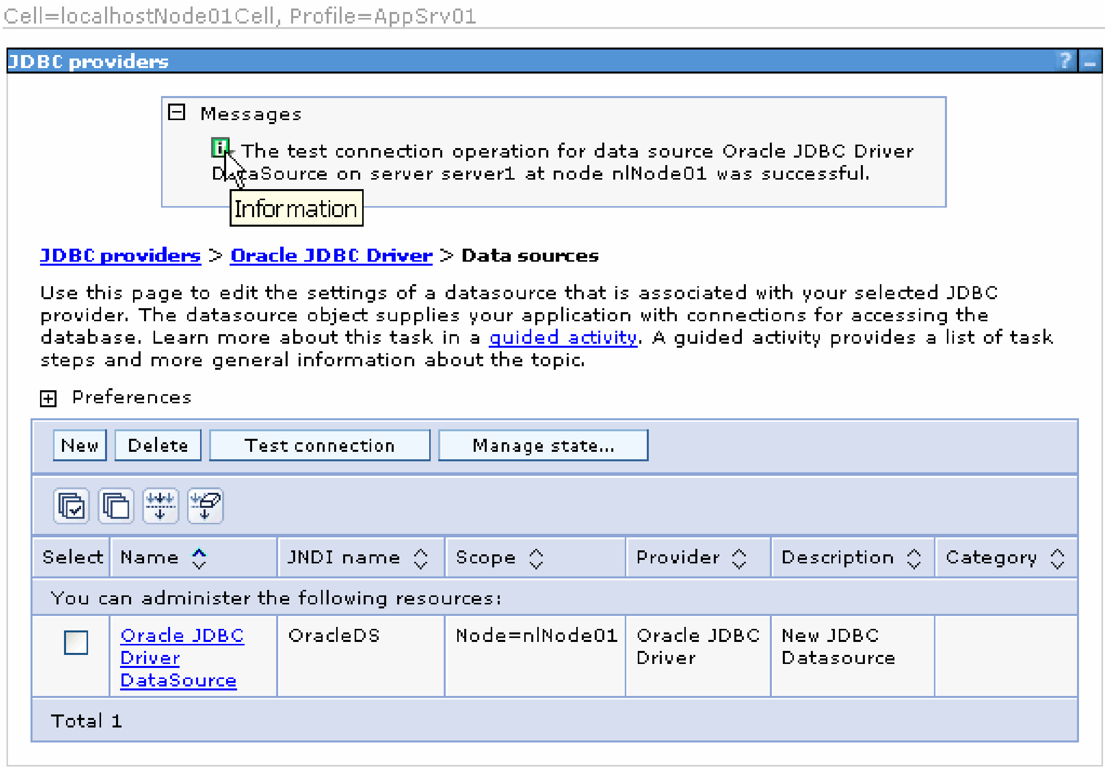
JDBC providers > Oracle JDBC Driver > Data sources

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

Select	Name	JNDI name	Scope	Provider	Description	Category
You can administer the following resources:						
☑	Oracle JDBC Driver DataSource	OracleDS	Node=nlNode01	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

⊟ Messages

The test connection operation for data source Oracle JDBC Driver DataSource on server server1 at node nlNode01 was successful.

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

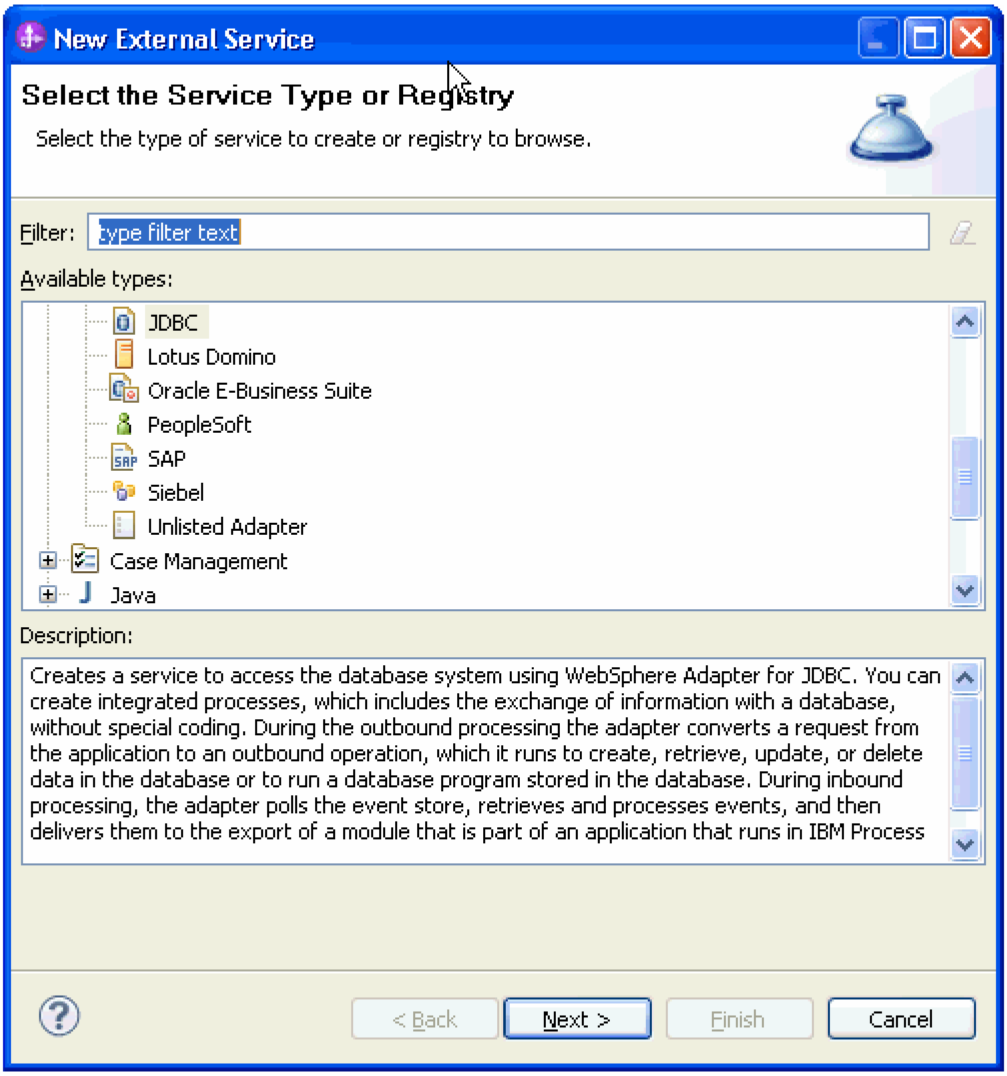
Select	Name	JNDI name	Scope	Provider	Description	Category
You can administer the following resources:						
□	Oracle JDBC Driver DataSource	OracleDS	Node=nlNode01	Oracle JDBC Driver	New JDBC Datasource	
Total 1						

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

Configure the adapter for outbound processing

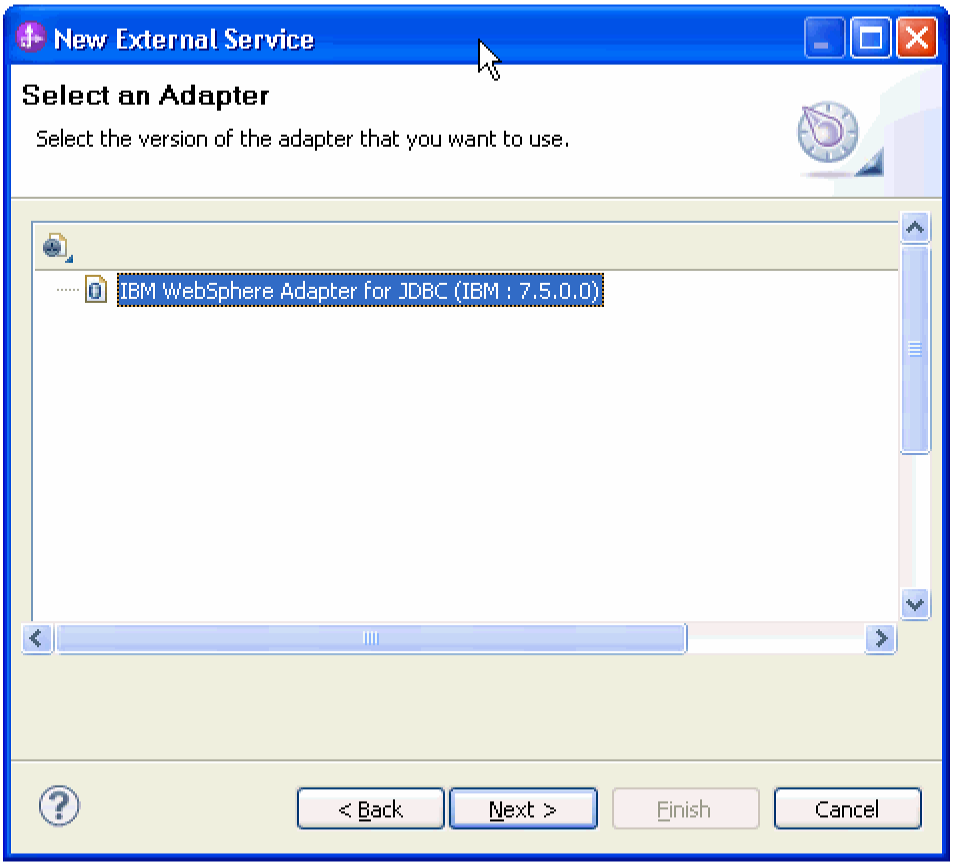
Run the external service wizard to specify business objects, services, and configuration details.

1. Switch to the Business Integration Perspective in IBM Integration Designer 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.**



Creates a service to access the database system using WebSphere Adapter for JDBC. You can create integrated processes, which includes the exchange of information with a database, without special coding. During the outbound processing the adapter converts a request from the application to an outbound operation, which it runs to create, retrieve, update, or delete data in the database or to run a database program stored in the database. During inbound processing, the adapter polls the event store, retrieves and processes events, and then delivers them to the export of a module that is part of an application that runs in IBM Process

4. Select the **IBM WebSphere Adapter for JDBC (IBM: 7.5.0.0)** and click **Next.**

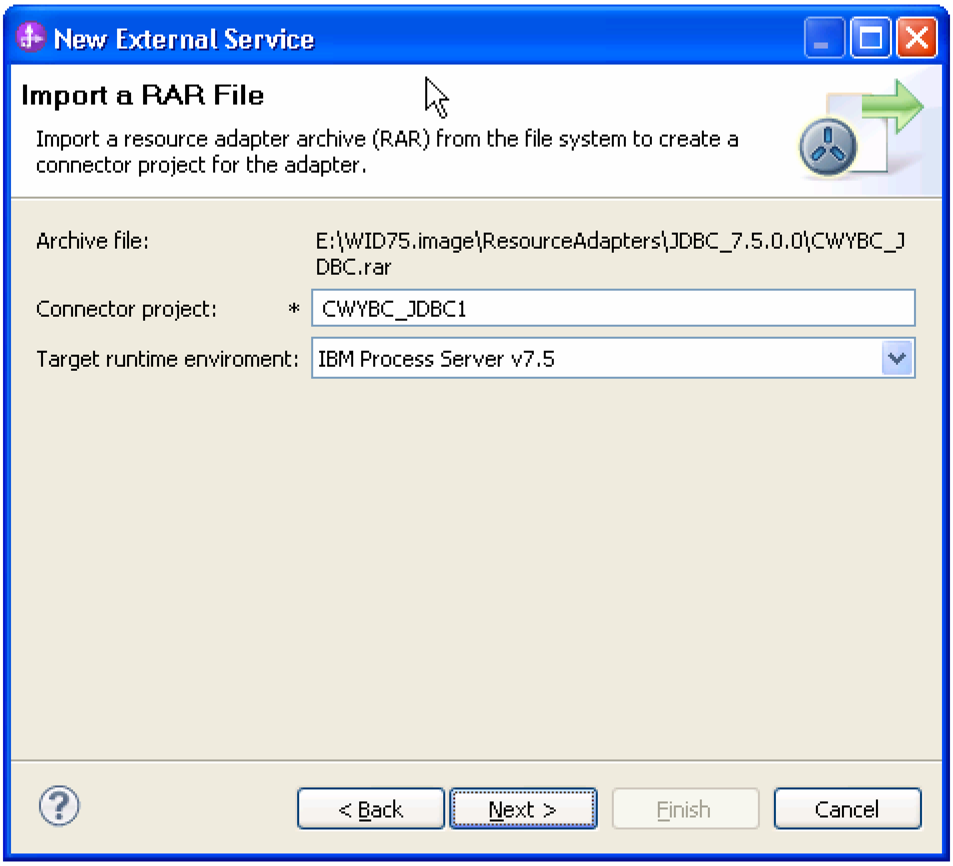


Select an Adapter

Select the version of the adapter that you want to use.

IBM WebSphere Adapter for JDBC (IBM : 7.5.0.0)

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



Import a RAR File

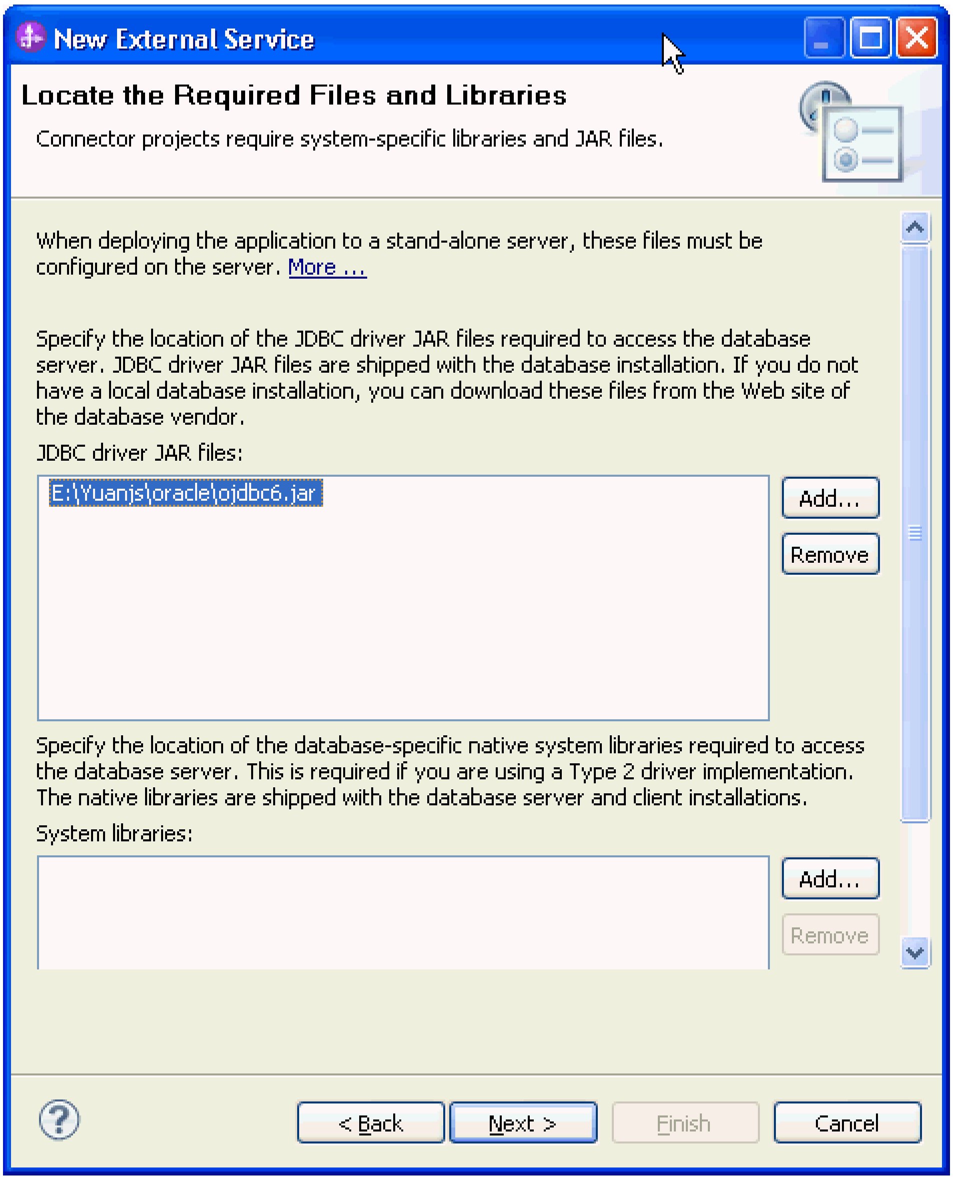
Import a resource adapter archive (RAR) from the file system to create a connector project for the adapter.

Archive file: E:\WID75.image\ResourceAdapters\JDBC_7.5.0.0\CWYBC_JDBC.rar

Connector project: * CWYBC_JDBC1

Target runtime enviroment: IBM Process Server v7.5

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**.



Locate the Required Files and Libraries

Connector projects require system-specific libraries and JAR files.

When deploying the application to a stand-alone server, these files must be configured on the server. More ...

Specify the location of the JDBC driver JAR files required to access the database server. JDBC driver JAR files are shipped with the database installation. If you do not have a local database installation, you can download these files from the Web site of the database vendor.

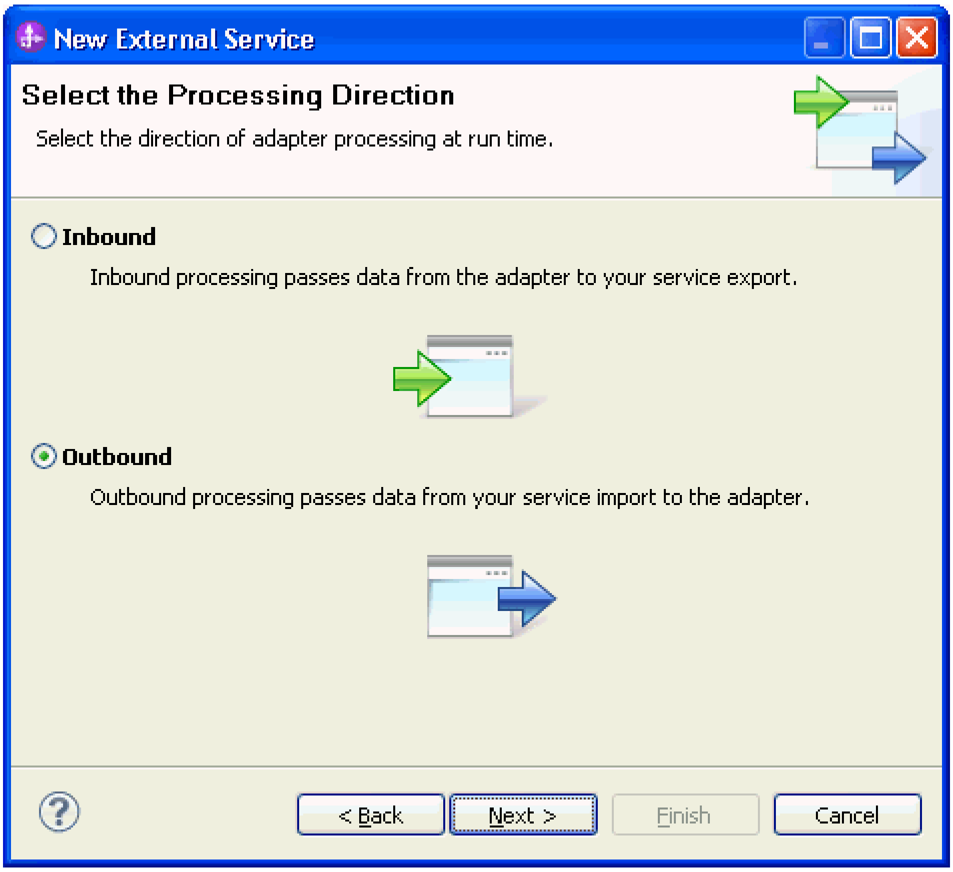
JDBC driver JAR files:

E:\Yuanjs\oracle\ojdbc6.jar

Specify the location of the database-specific native system libraries required to access the database server. This is required if you are using a Type 2 driver implementation. The native libraries are shipped with the database server and client installations.

System libraries:

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



New External Service

Select the Processing Direction

Select the direction of adapter processing at run time.

○ **Inbound**

Inbound processing passes data from the adapter to your service export.

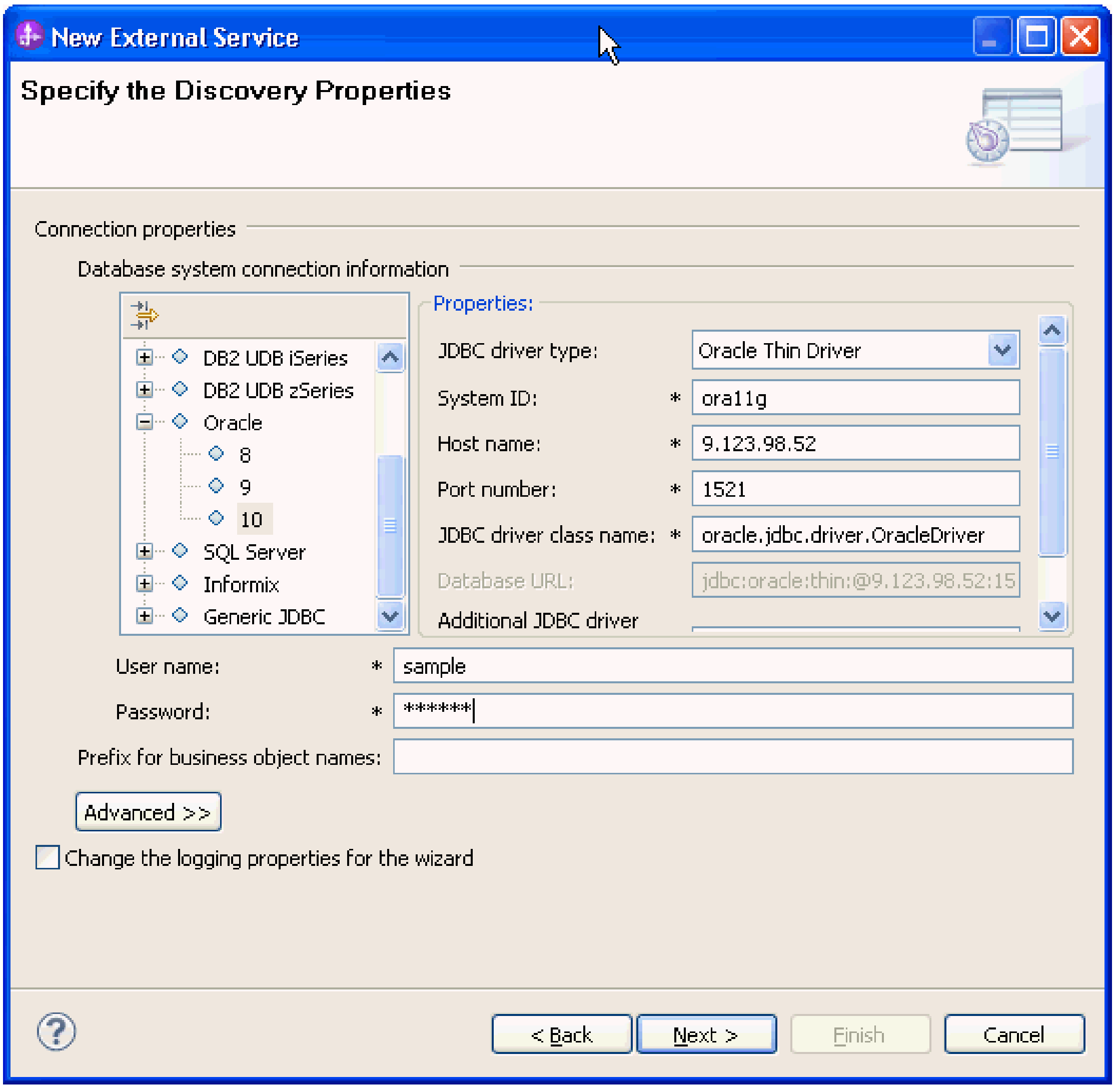
● **Outbound**

Outbound processing passes data from your service import to the adapter.

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**.



New External Service

Specify the Discovery Properties

Connection properties

Database system connection information

Properties:

JDBC driver type: Oracle Thin Driver

System ID: * ora11g

Host name: * 9.123.98.52

Port number: * 1521

JDBC driver class name: * oracle.jdbc.driver.OracleDriver

Database URL: jdbc:oracle:thin:@9.123.98.52:15

Additional JDBC driver

User name: * sample

Password: * *******

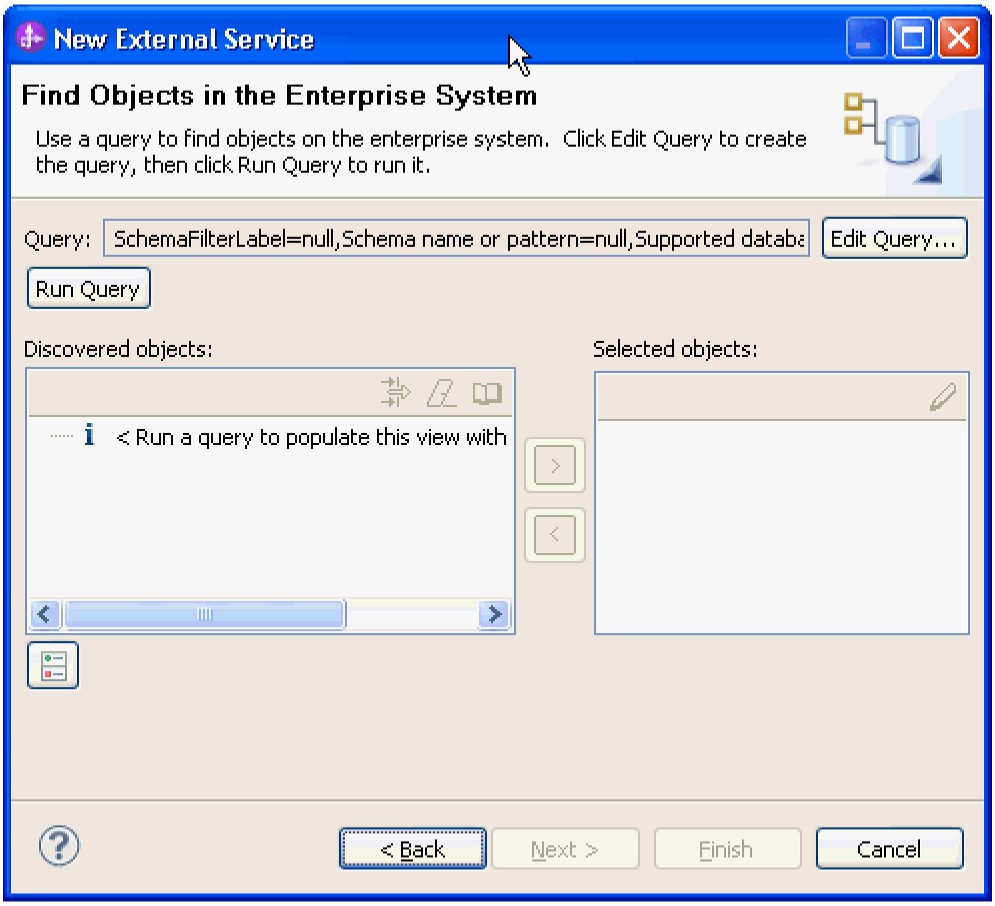
Prefix for business object names:

□ Change the logging properties for the wizard

Select the business objects to be used with the adapter

Follow these steps to select the **SAMPLE_ARRAY_STRUCT** business object:

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



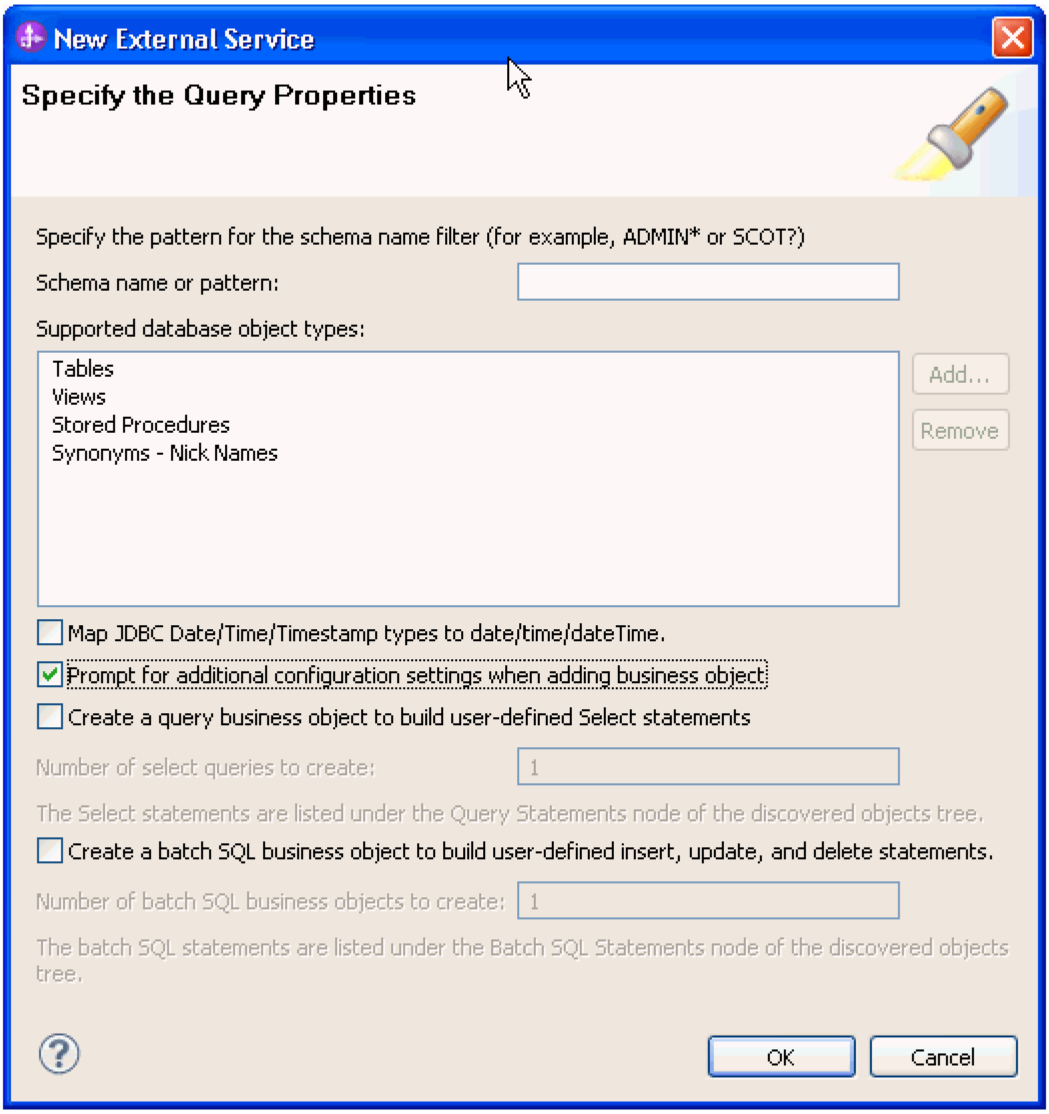
New External Service

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 databa

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



New External Service

Specify the Query Properties

Specify the pattern for the schema name filter (for example, ADMIN* or SCOT?)

Schema name or pattern:

Supported database object types:

Tables
Views
Stored Procedures
Synonyms - Nick Names

☐ Map JDBC Date/Time/Timestamp types to date/time/dateTime.

☑ Prompt for additional configuration settings when adding business object

☐ Create a query business object to build user-defined Select statements

Number of select queries to create: 1

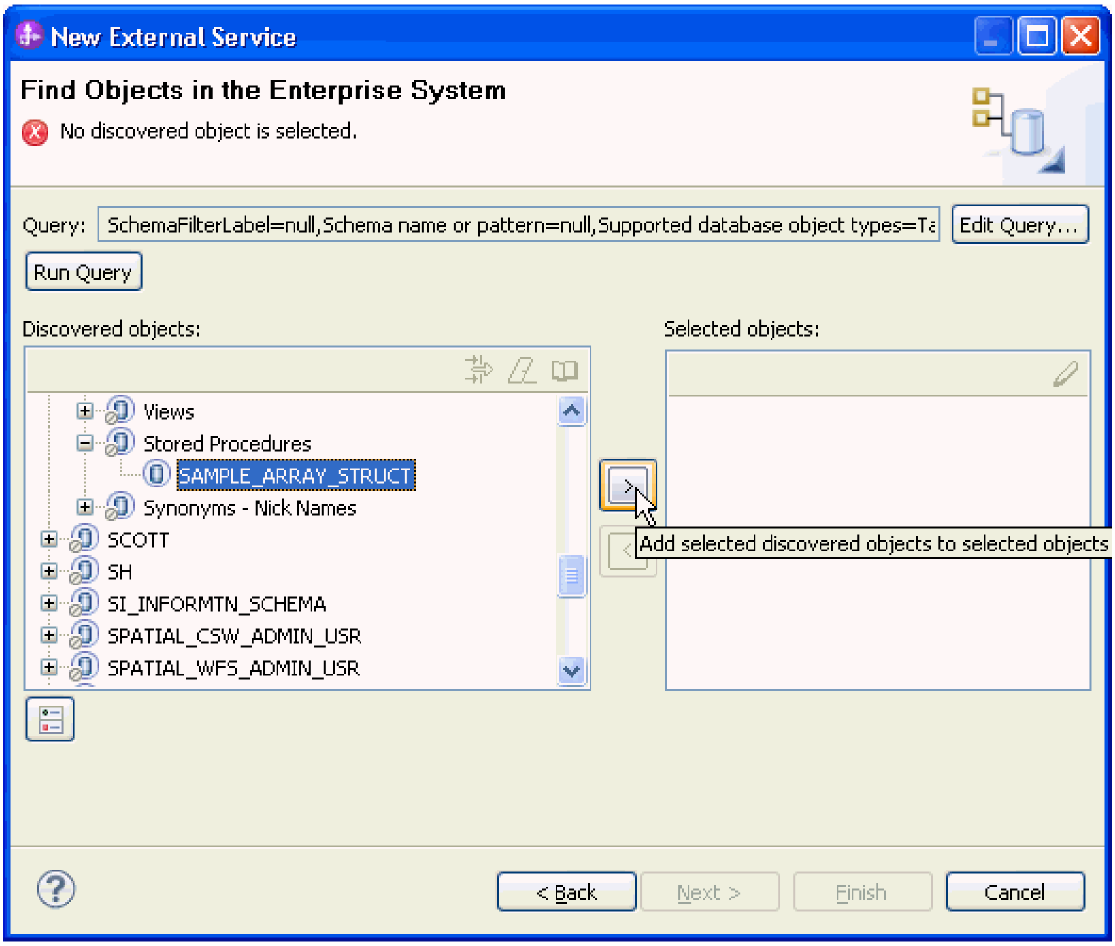
The Select statements are listed under the Query Statements node of the discovered objects tree.

☐ Create a batch SQL business object to build user-defined insert, update, and delete statements.

Number of batch SQL business objects to create: 1

The batch SQL statements are listed under the Batch SQL Statements node of the discovered objects tree.

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 >.



New External Service

Find Objects in the Enterprise System

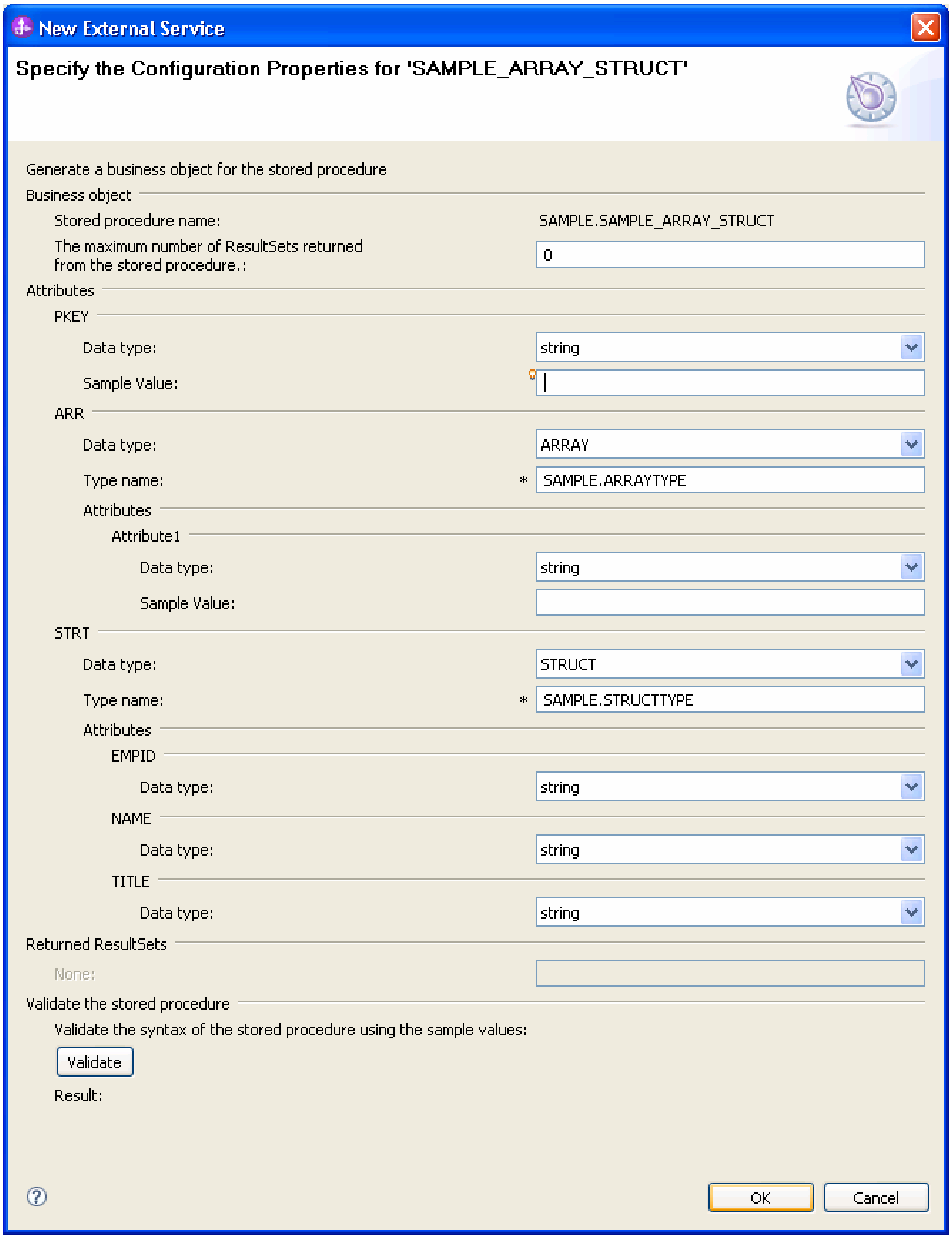
No discovered object is selected.

Query: SchemaFilterLabel=null,Schema name or pattern=null,Supported database object types=Ta

Discovered objects:

Selected objects:

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

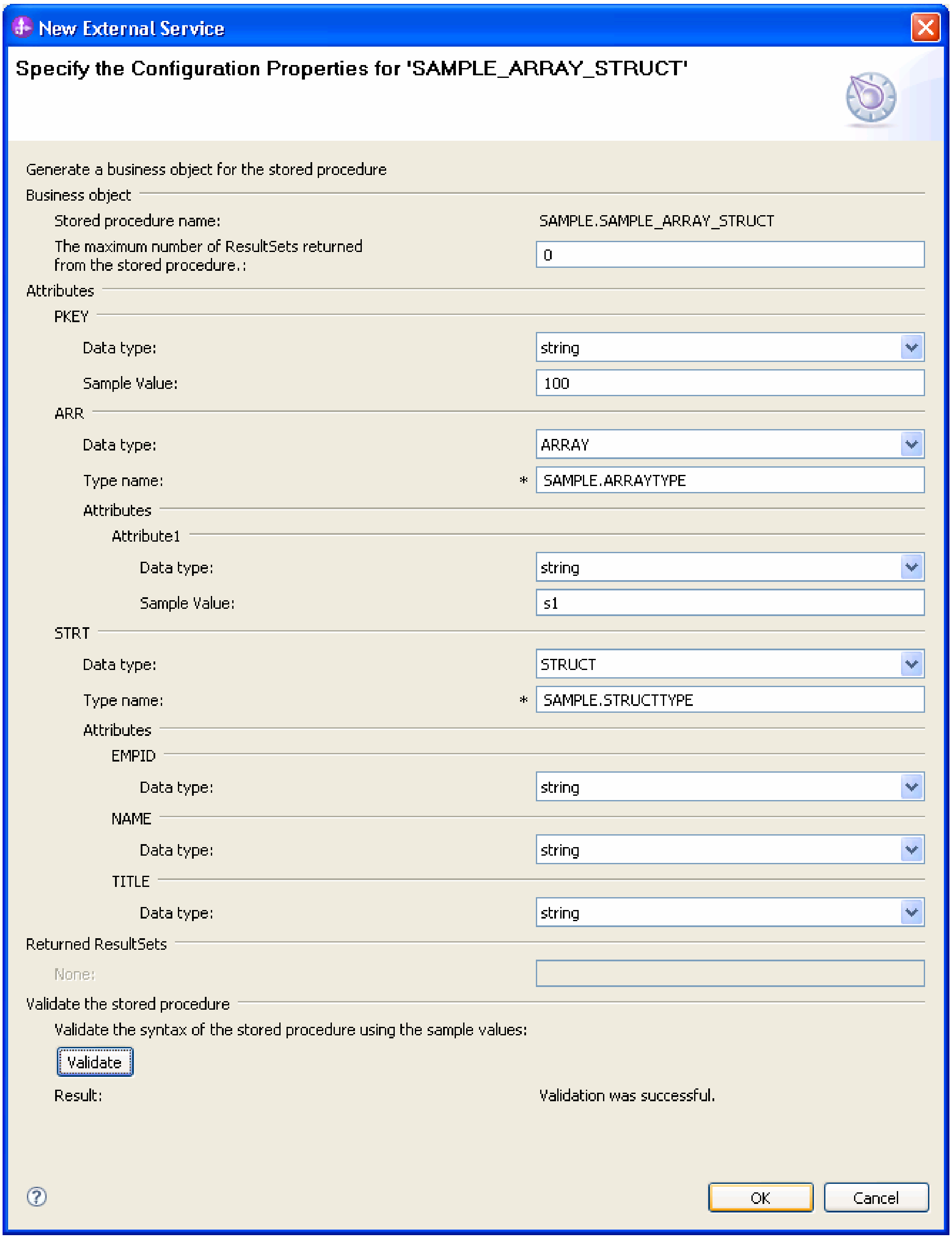
None:

Validate the stored procedure

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

Result:

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

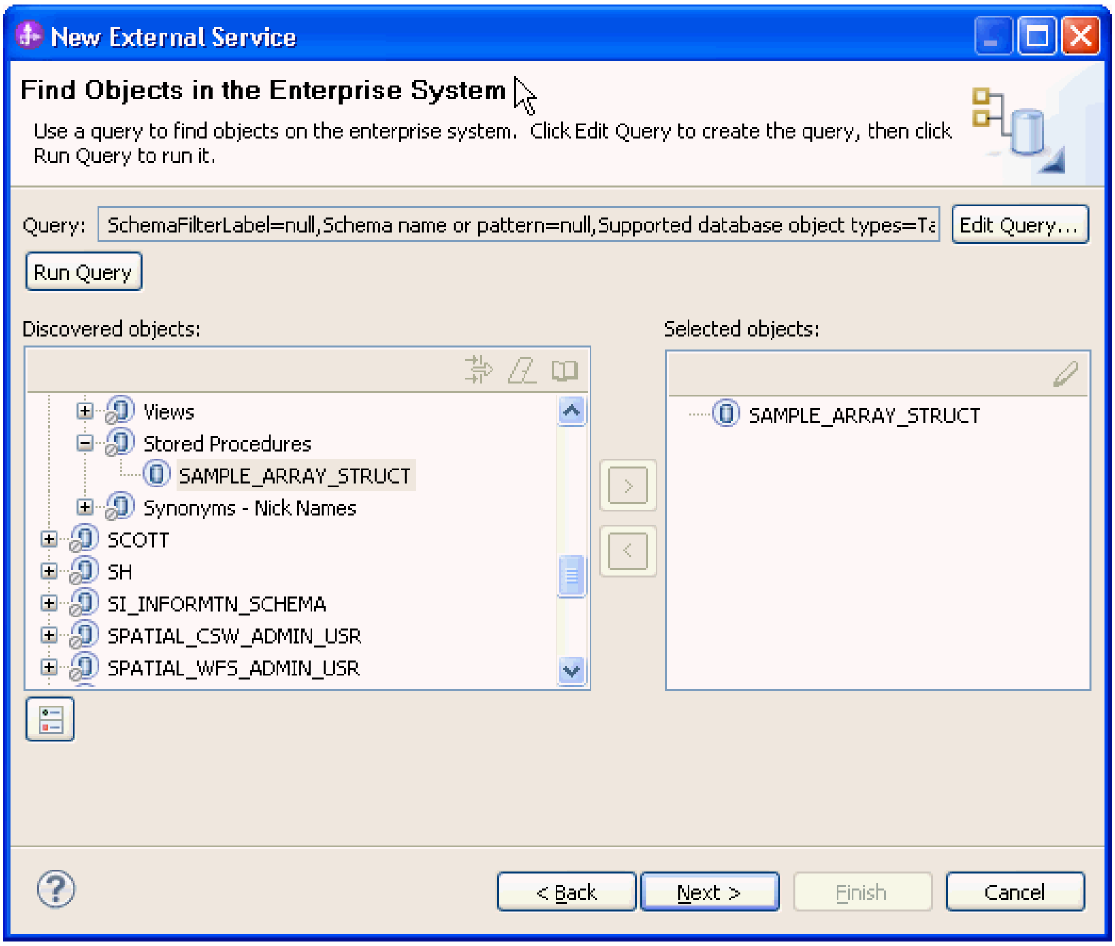
None:

Validate the stored procedure

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

Result: Validation was successful.

8. Click **Next**.



Find Objects in the Enterprise System

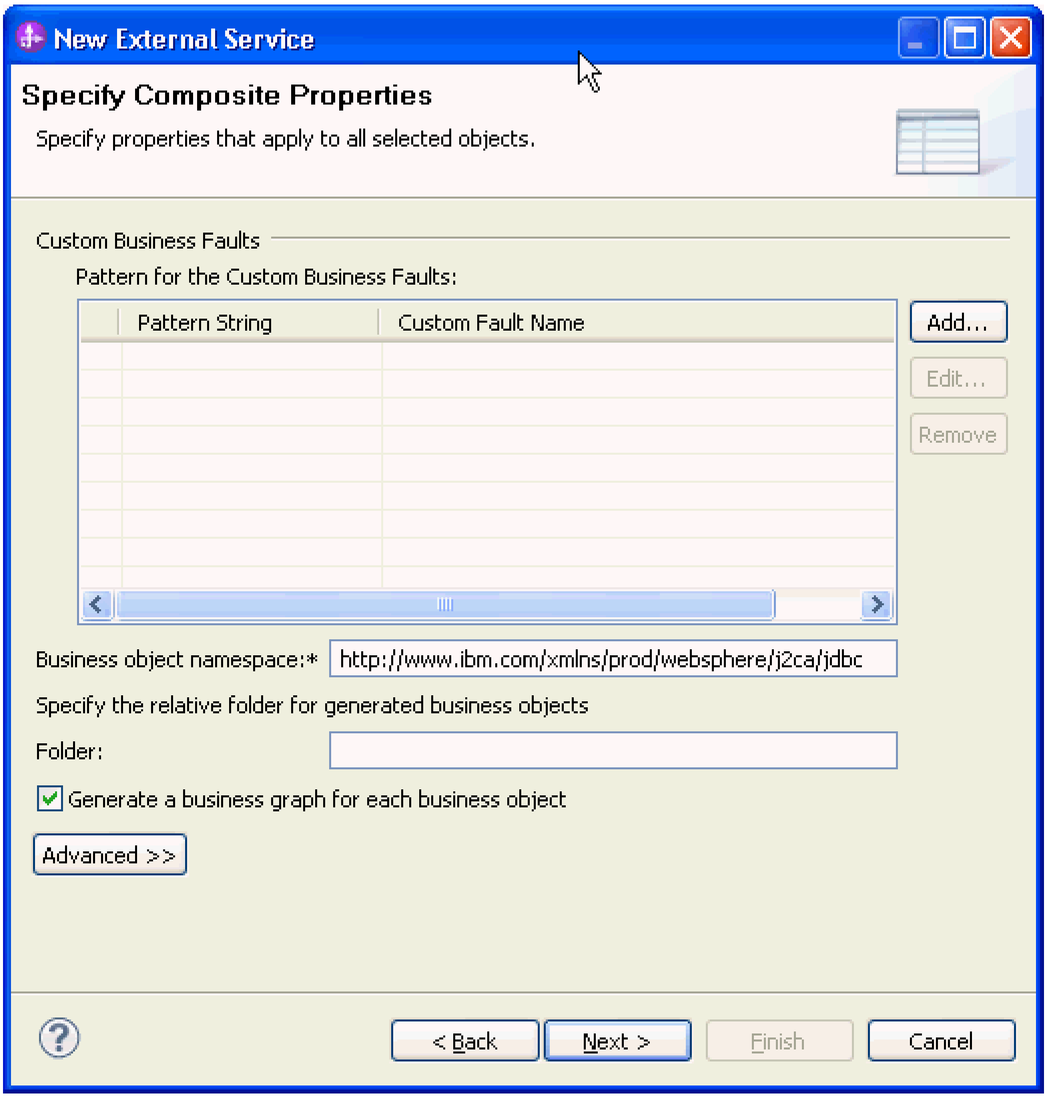
Use a query to find objects on the enterprise system. Click Edit Query to create the query, then click Run Query to run it.

Query: SchemaFilterLabel=null,Schema name or pattern=null,Supported database object types=Ta

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**.



New External Service

Specify Composite Properties

Specify properties that apply to all selected objects.

Custom Business Faults

Pattern for the Custom Business Faults:

Pattern String	Custom Fault Name

Business object namespace:* http://www.ibm.com/xmlns/prod/websphere/j2ca/jdbc

Specify the relative folder for generated business objects

Folder:

☑ Generate a business graph for each business object

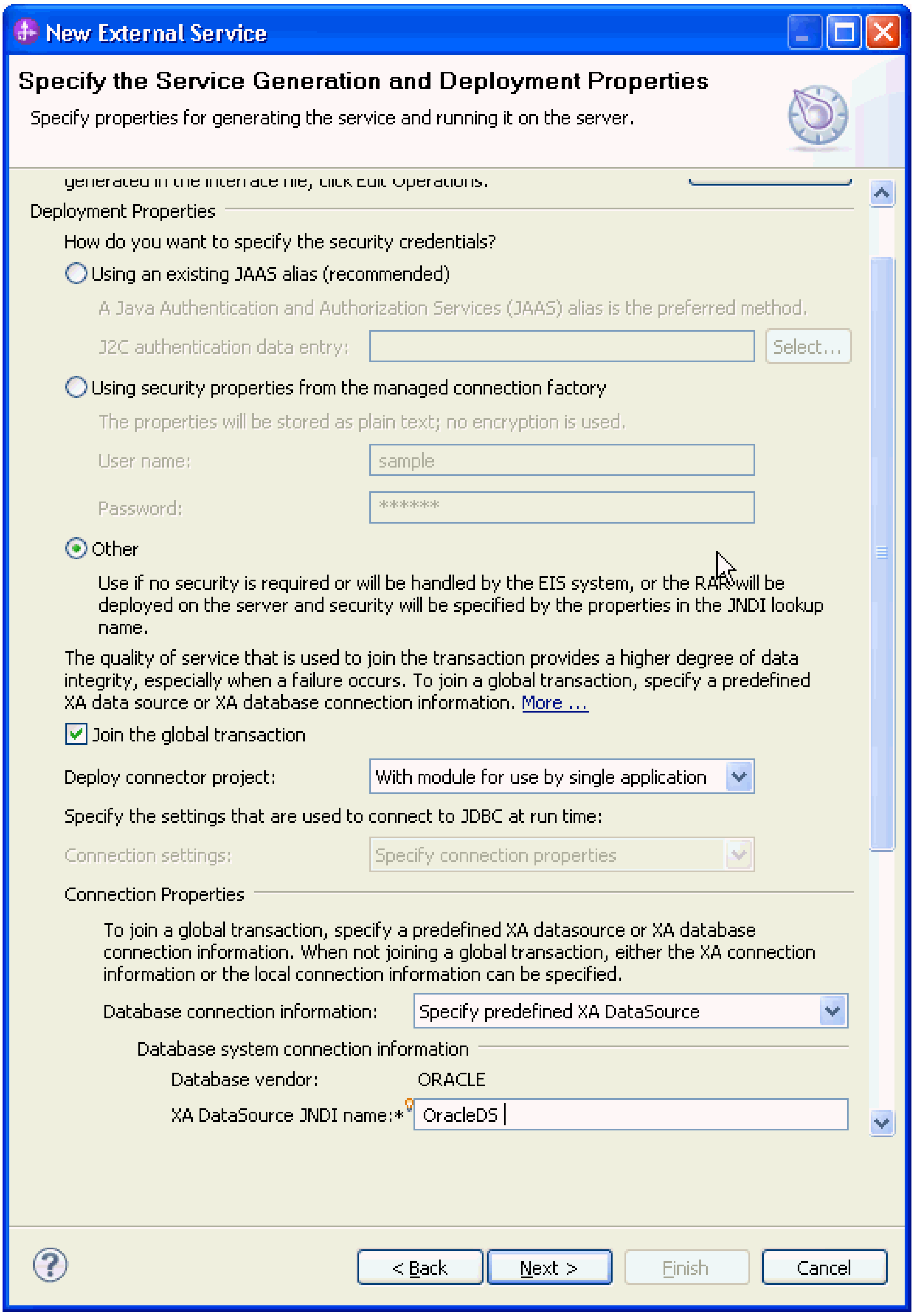
2. In the Specify the Service Generation and Deployment Properties window, perform the following steps:

   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**.



Specify the Service Generation and Deployment Properties

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

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: sample

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 join a global transaction, specify a predefined XA data source or XA database connection information. More ...

☑ Join the global transaction

Deploy connector project: With module for use by single application

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

Connection settings: Specify connection properties

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.

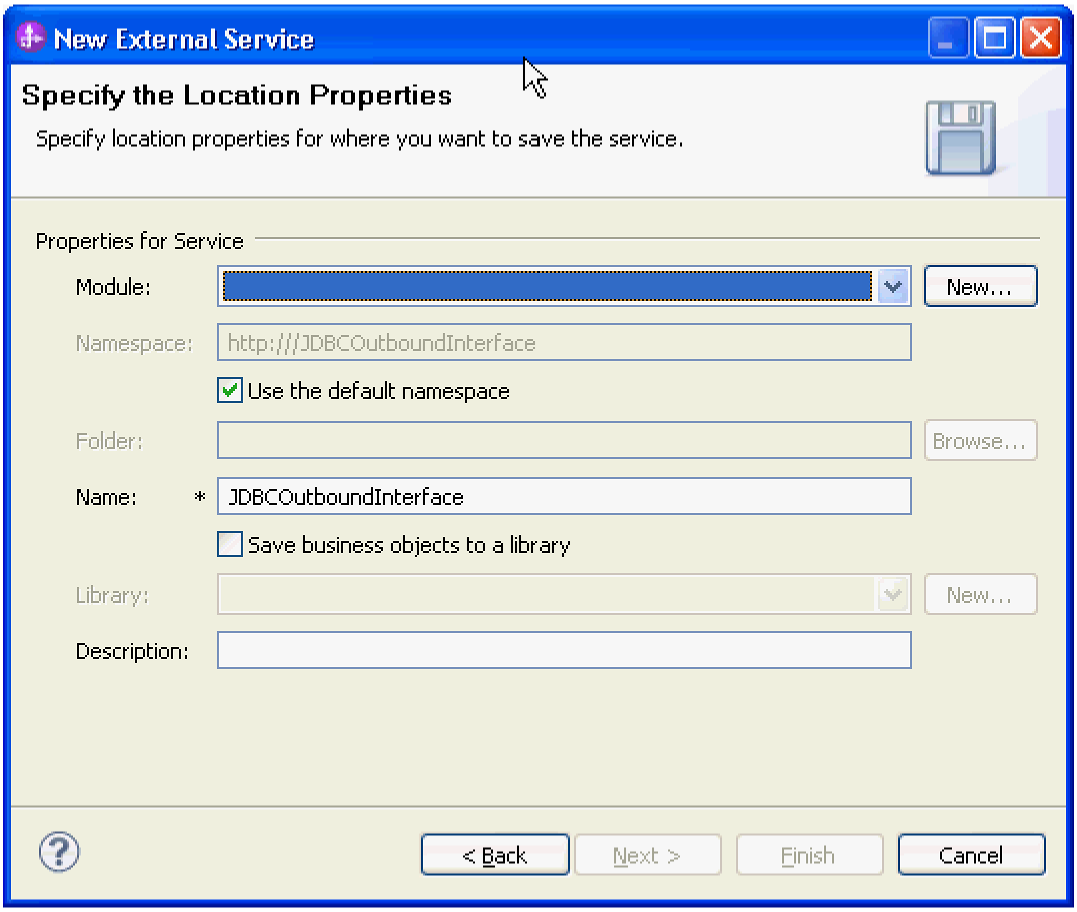
Database connection information: Specify predefined XA DataSource

Database system connection information

Database vendor: ORACLE

XA DataSource JNDI name:* OracleDS

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



New External Service

Specify the Location Properties

Specify location properties for where you want to save the service.

Properties for Service

Module:

Namespace: http:///JDBCOutboundInterface

☑ Use the default namespace

Folder:

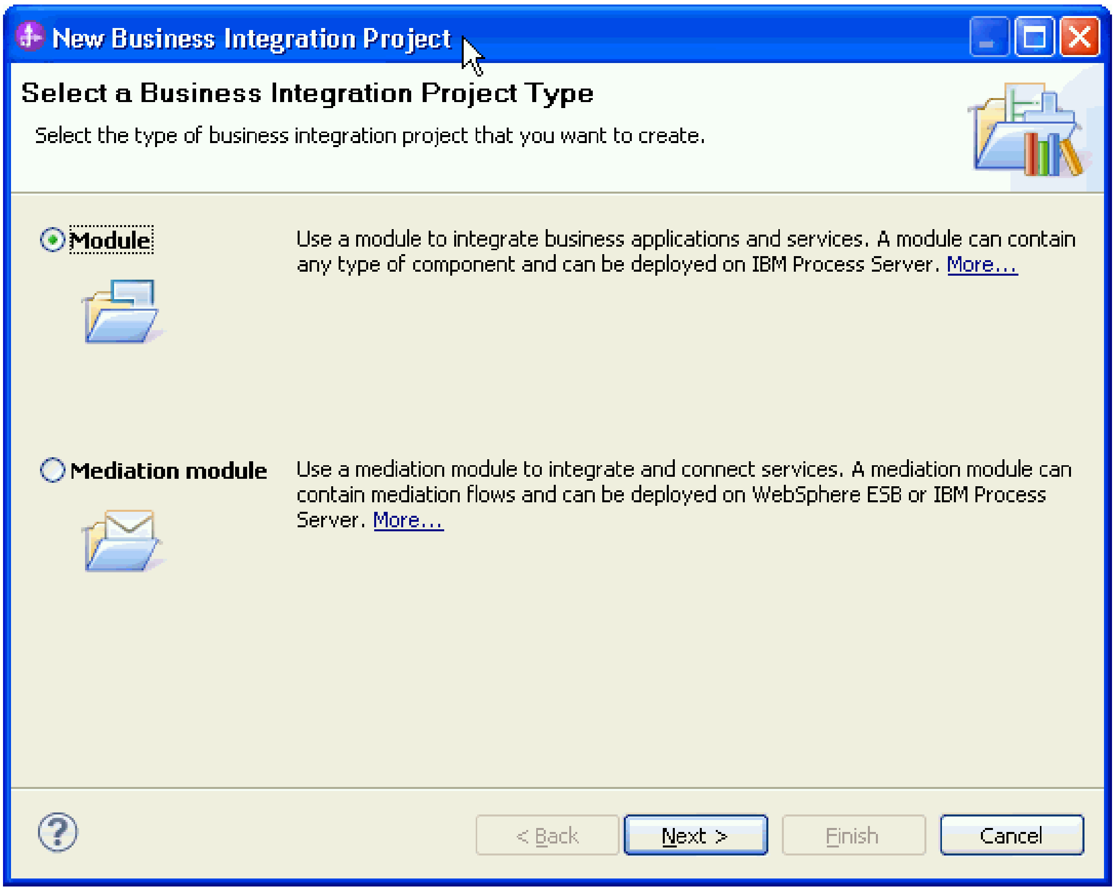
Name: * JDBCOutboundInterface

□ Save business objects to a library

Library:

Description:

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



Select a Business Integration Project Type

Select the type of business integration project that you want to create.

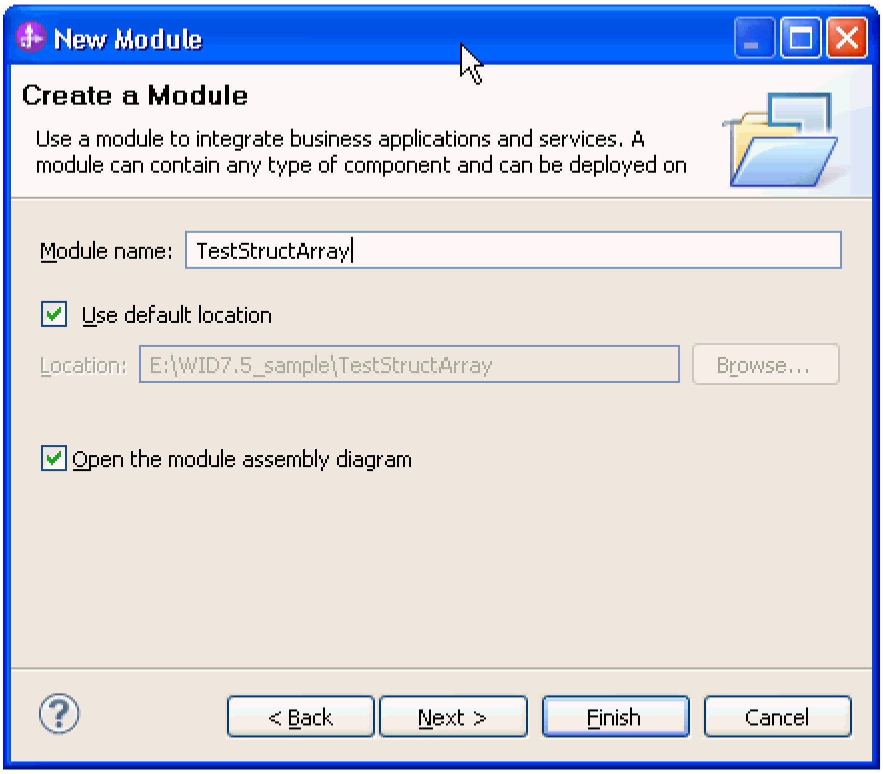
● **Module**

Use a module to integrate business applications and services. A module can contain any type of component and can be deployed on IBM Process Server. More...

○ **Mediation module**

Use a mediation module to integrate and connect services. A mediation module can contain mediation flows and can be deployed on WebSphere ESB or IBM Process Server. More...

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



Create a Module

Use a module to integrate business applications and services. A module can contain any type of component and can be deployed on

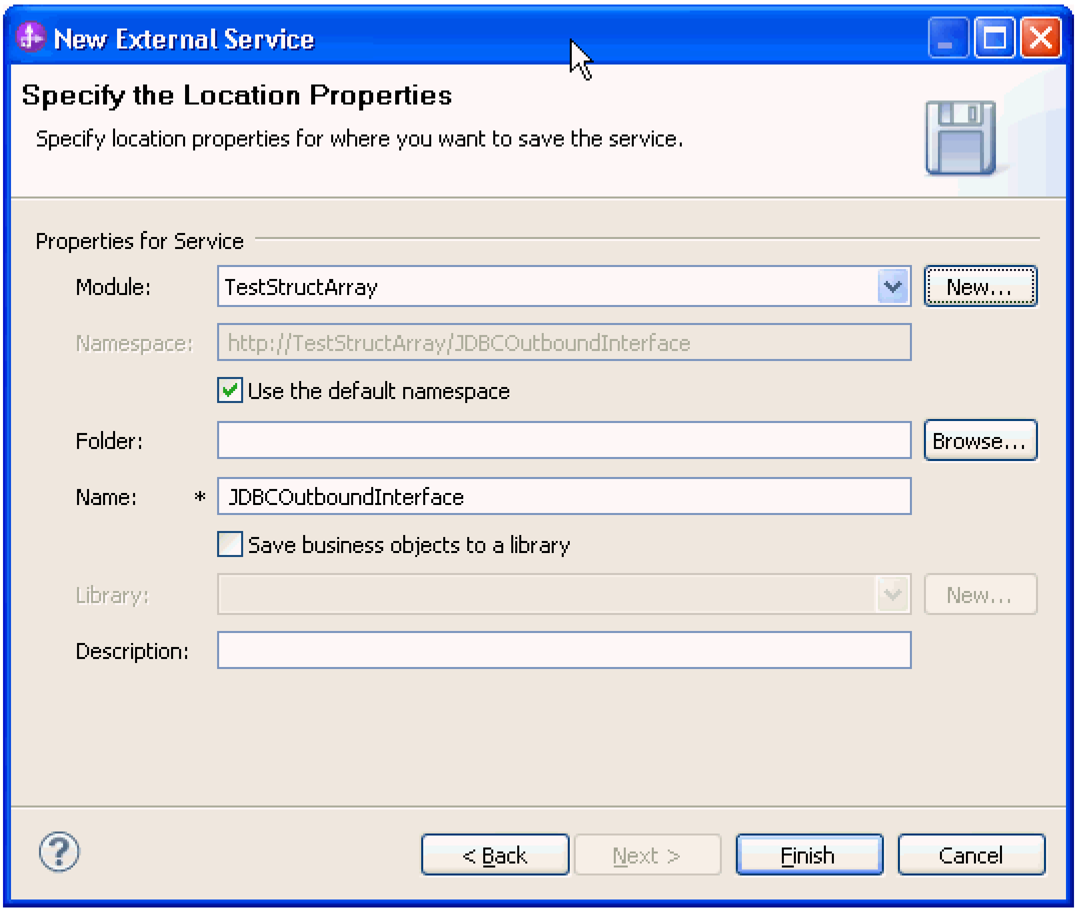
Module name: TestStructArray

☑ Use default location

Location: E:\WID7.5_sample\TestStructArray

☑ Open the module assembly diagram

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



Specify the Location Properties

Specify location properties for where you want to save the service.

Properties for Service

Module: TestStructArray

Namespace: http://TestStructArray/JDBCOutboundInterface

☑ Use the default namespace

Folder:

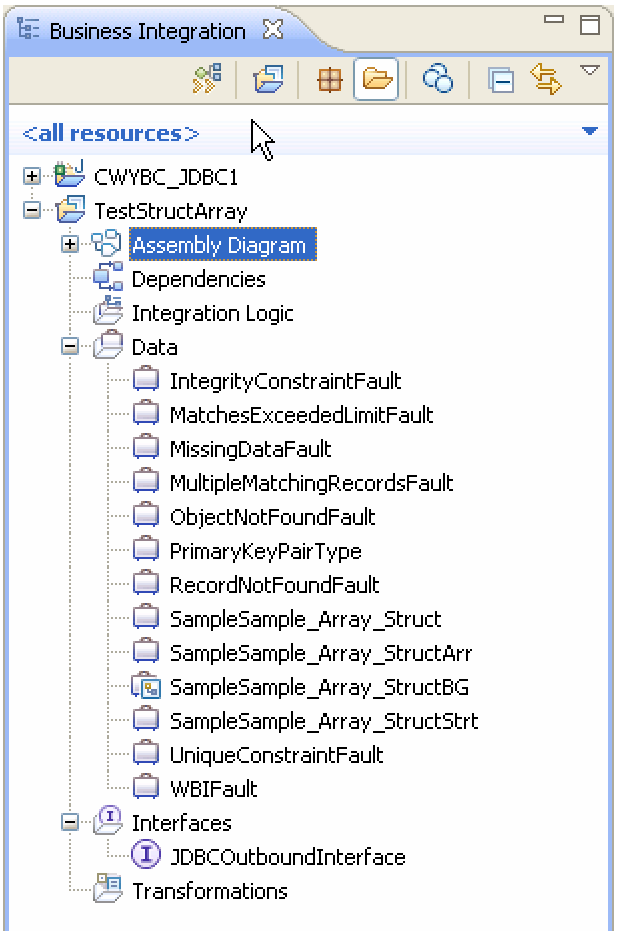
Name: * JDBCOutboundInterface

□ Save business objects to a library

Library:

Description:

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



Business Integration

<all resources>

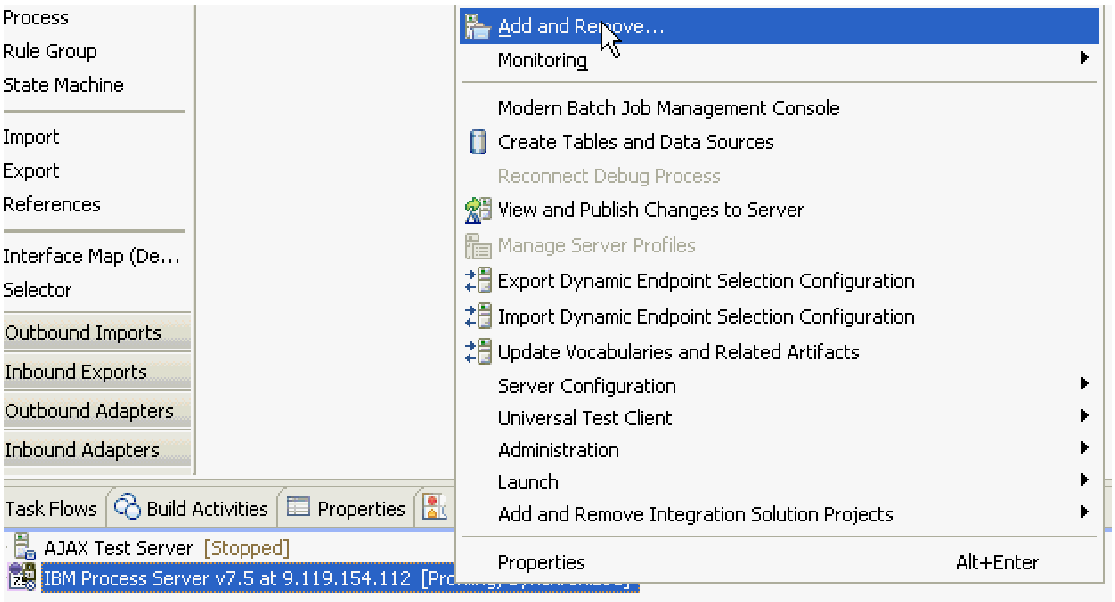
CWYBC_JDBC1
TestStructArray
Assembly Diagram
Dependencies
Integration Logic
Data
IntegrityConstraintFault
MatchesExceededLimitFault
MissingDataFault
MultipleMatchingRecordsFault
ObjectNotFoundFault
PrimaryKeyPairType
RecordNotFoundFault
SampleSample_Array_Struct
SampleSample_Array_StructArr
SampleSample_Array_StructBG
SampleSample_Array_StructStrt
UniqueConstraintFault
WBIFault
Interfaces
JDBCOutboundInterface
Transformations

Deploy the module to the test environment

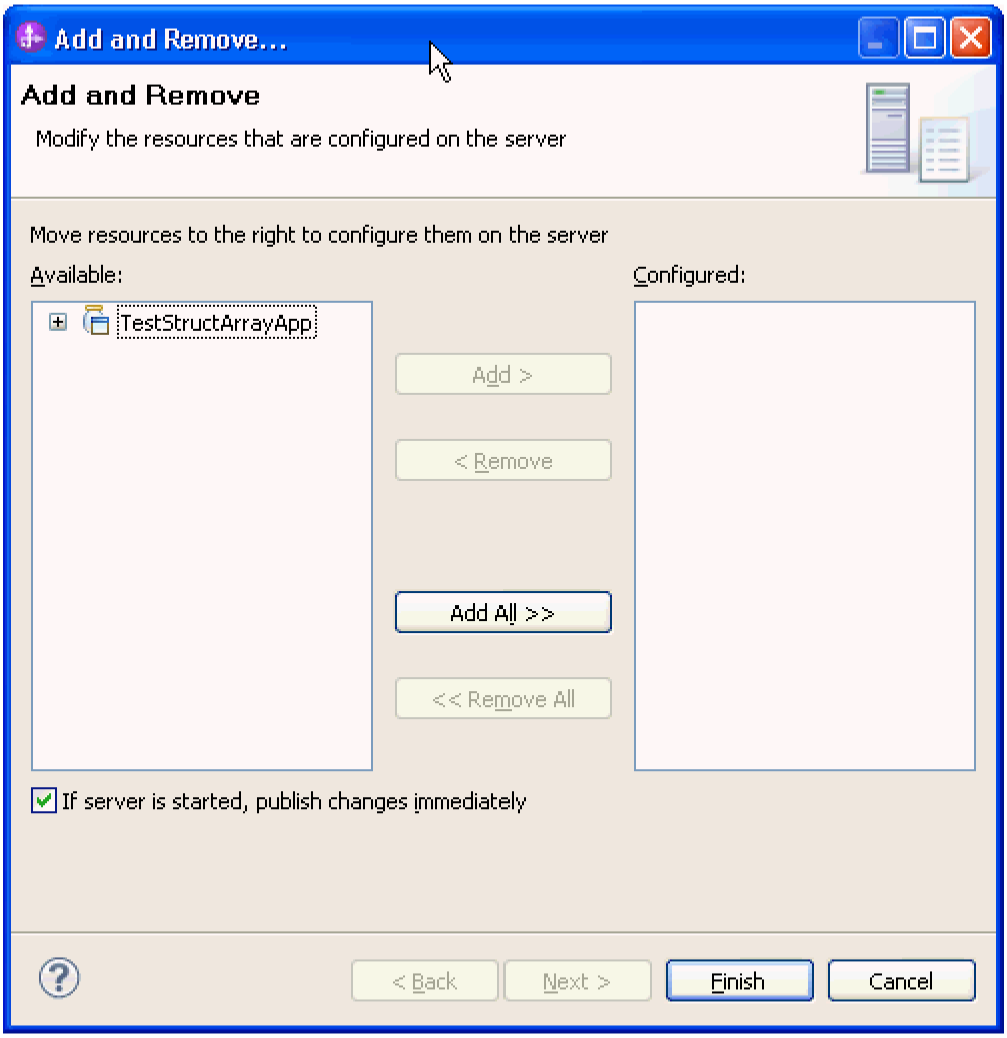
After running the external service wizard, you will have an SCA module that contains an Enterprise Information System import. You must install this SCA module in the IBM Integration Designer integration test client. To do this, you must add the SCA module you created earlier to the server using the **Servers**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

Steps for adding the SCA module to the server:

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


Process
Rule Group
State Machine
Import
Export
References
Interface Map (De...
Selector
Outbound Imports
Inbound Exports
Outbound Adapters
Inbound Adapters
Task Flows
Build Activities
Properties
AJAX Test Server [Stopped]
IBM Process Server v7.5 at 9.119.154.112 [Pro
Add and Remove...
Monitoring
Modern Batch Job Management Console
Create Tables and Data Sources
Reconnect Debug Process
View and Publish Changes to Server
Manage Server Profiles
Export Dynamic Endpoint Selection Configuration
Import Dynamic Endpoint Selection Configuration
Update Vocabularies and Related Artifacts
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 IBM Integration Designer workspace.


Add and Remove...

Add and Remove

Modify the resources that are configured on the server

Move resources to the right to configure them on the server

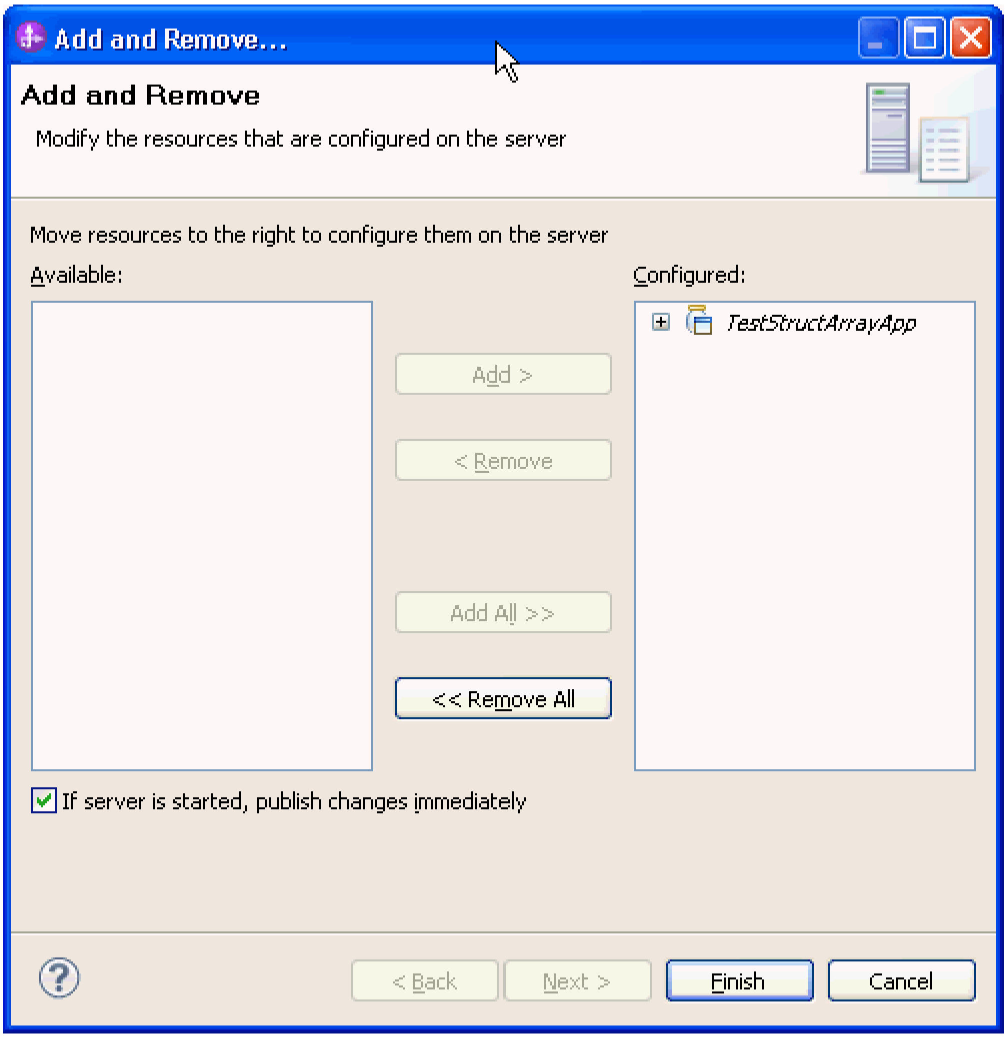
Available:

TestStructArrayApp

Configured:

☑ If server is started, publish changes immediately

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



Add and Remove...

Add and Remove

Modify the resources that are configured on the server

Move resources to the right to configure them on the server

Available:

Configured:

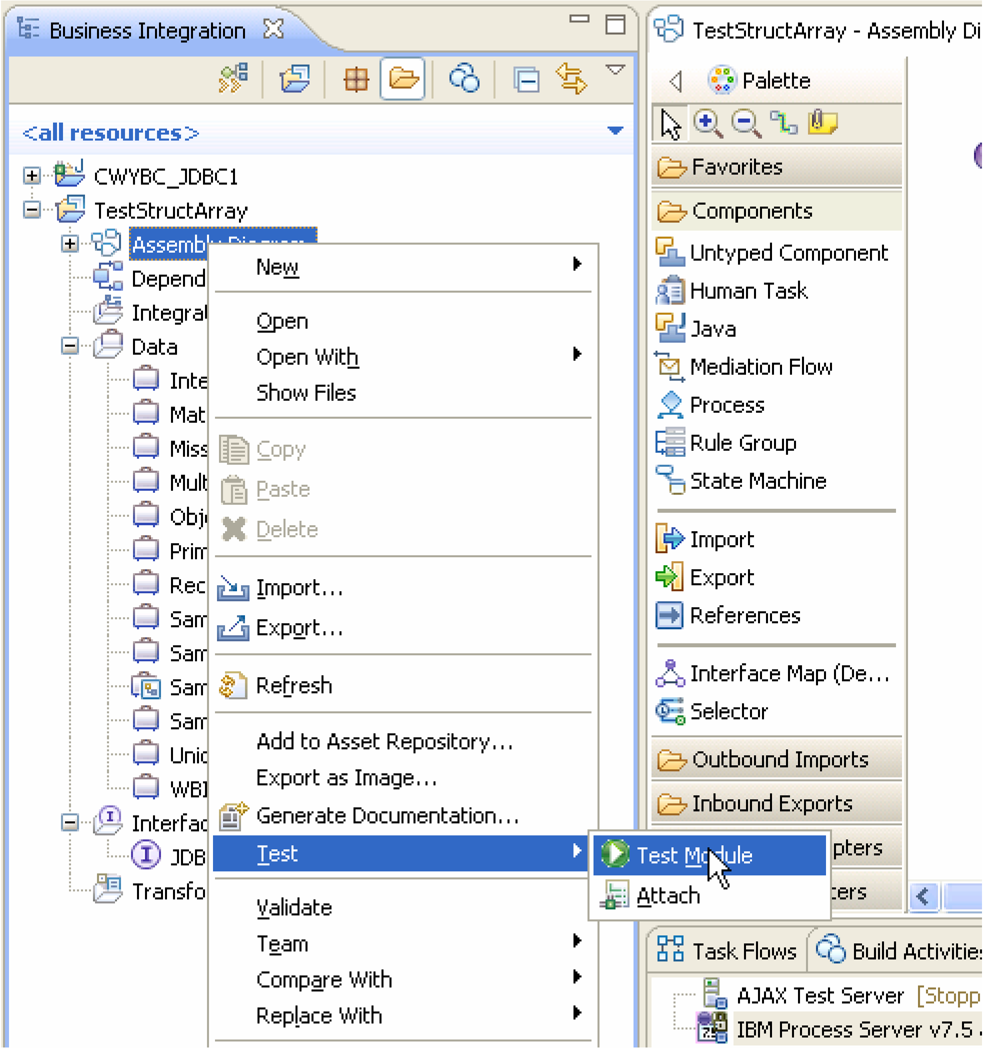
TestStructArrayApp

☑ If server is started, publish changes immediately

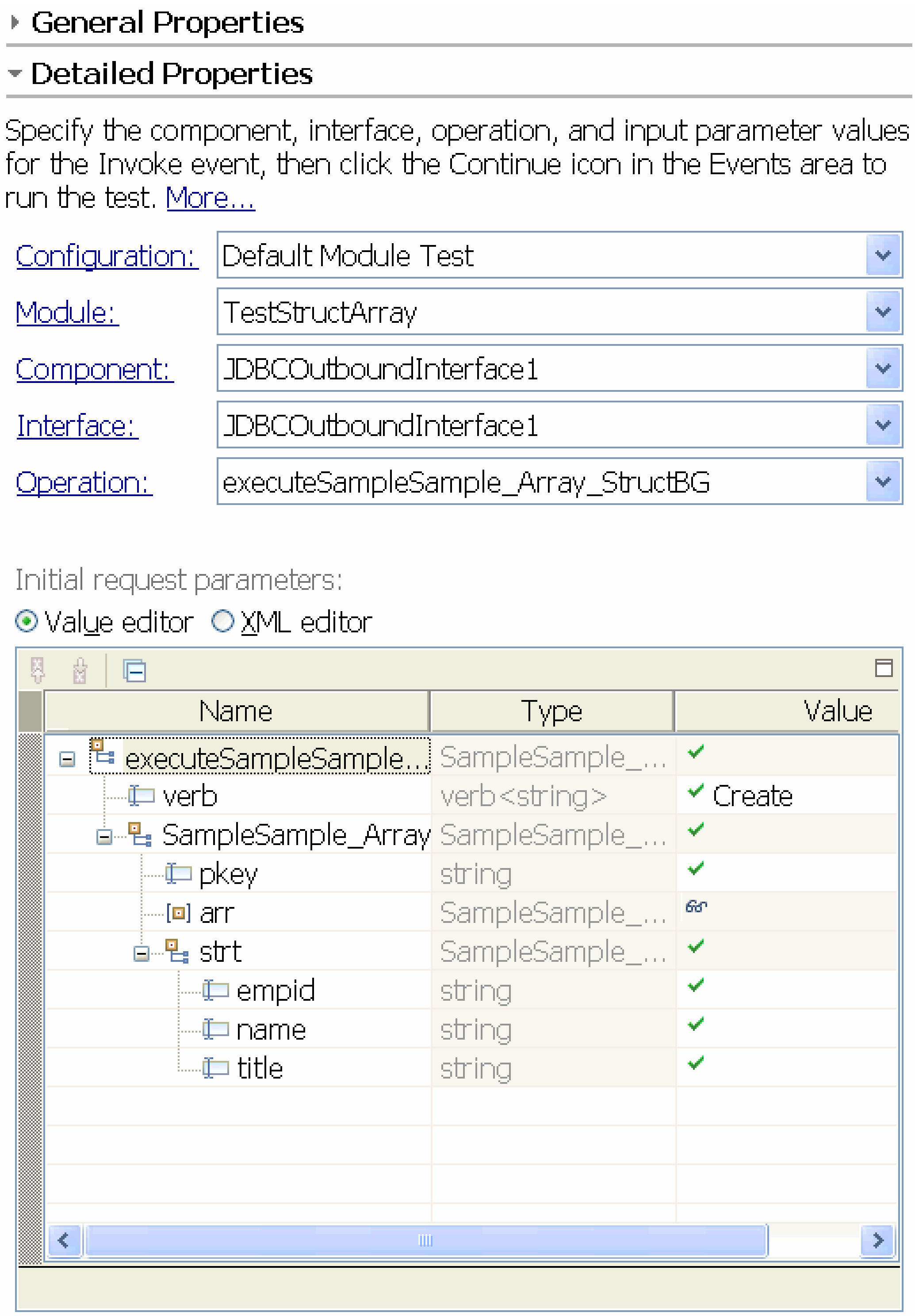
Test the assembled adapter application

Test the assembled adapter application using the IBM Integration Designer integration test client.

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



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



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: TestStructArray

Component: JDBCOutboundInterface1

Interface: JDBCOutboundInterface1

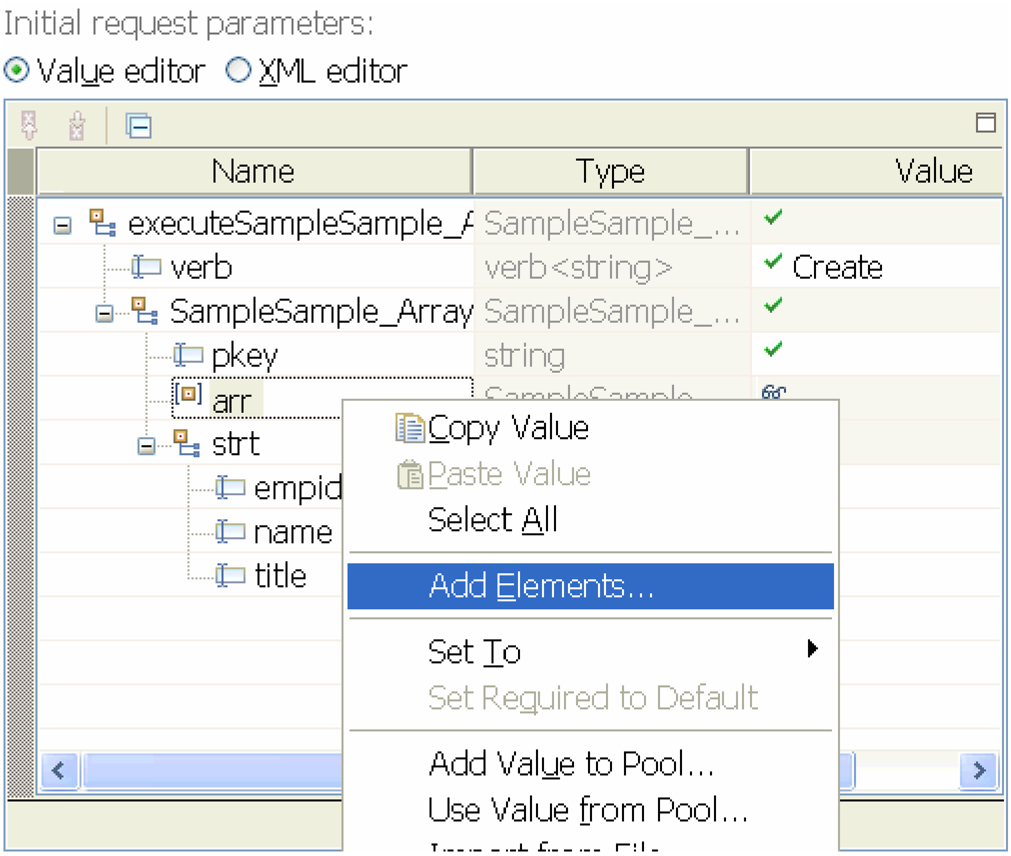
Operation: executeSampleSample_Array_StructBG

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**.

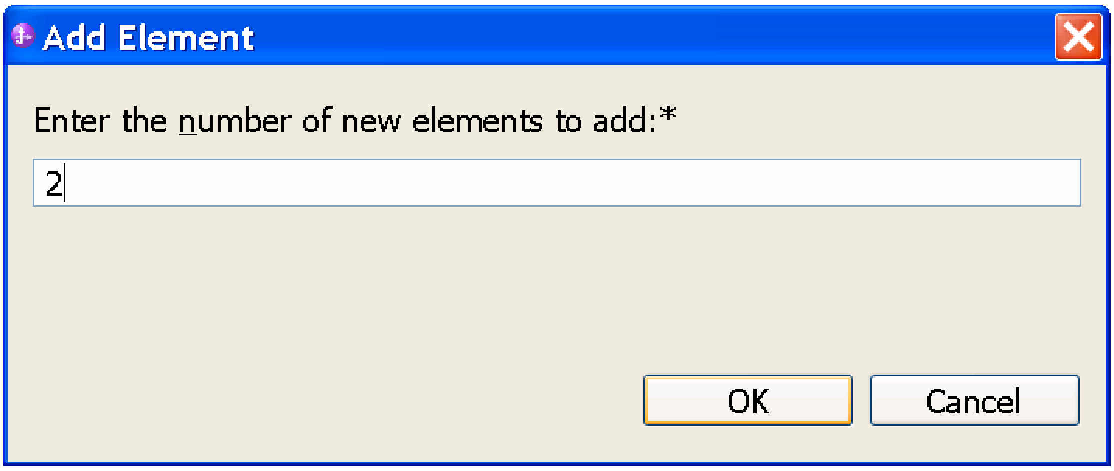


Initial request parameters:

● Value editor ○ XML editor

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

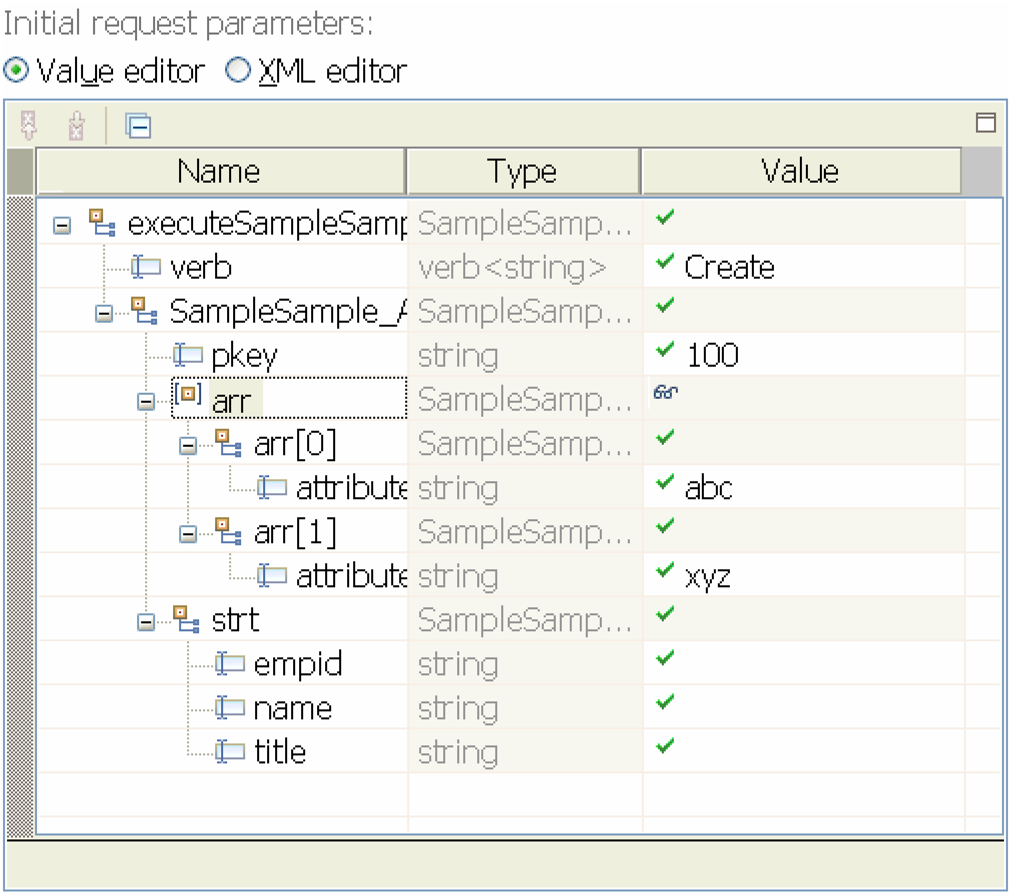
3. In the Add Element window enter 2, to create two child objects and click **OK**.



Add Element

Enter the number of new elements to add:* 2

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

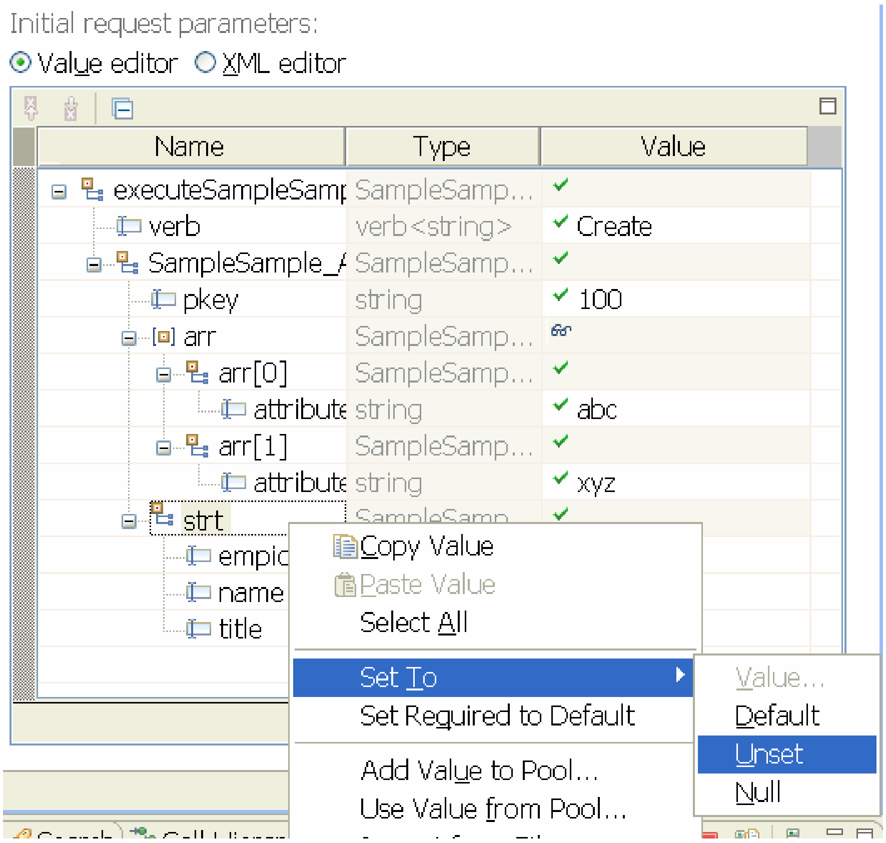


Initial request parameters:

● Value editor ○ XML editor

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**.



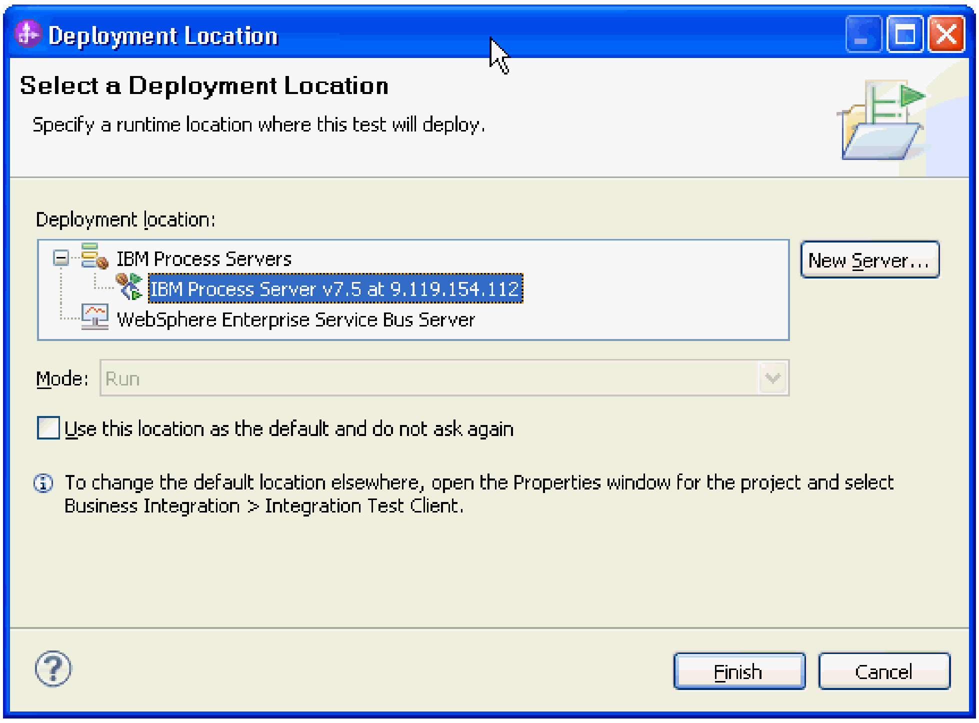
Initial request parameters:

● Value editor ○ XML editor

Name	Type	Value
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...	✓
empic		
name		
title		

6. Execute the service by click ▶.

7. 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:

- IBM Process Servers
  - IBM Process Server v7.5 at 9.119.154.112
- WebSphere Enterprise Service Bus Server

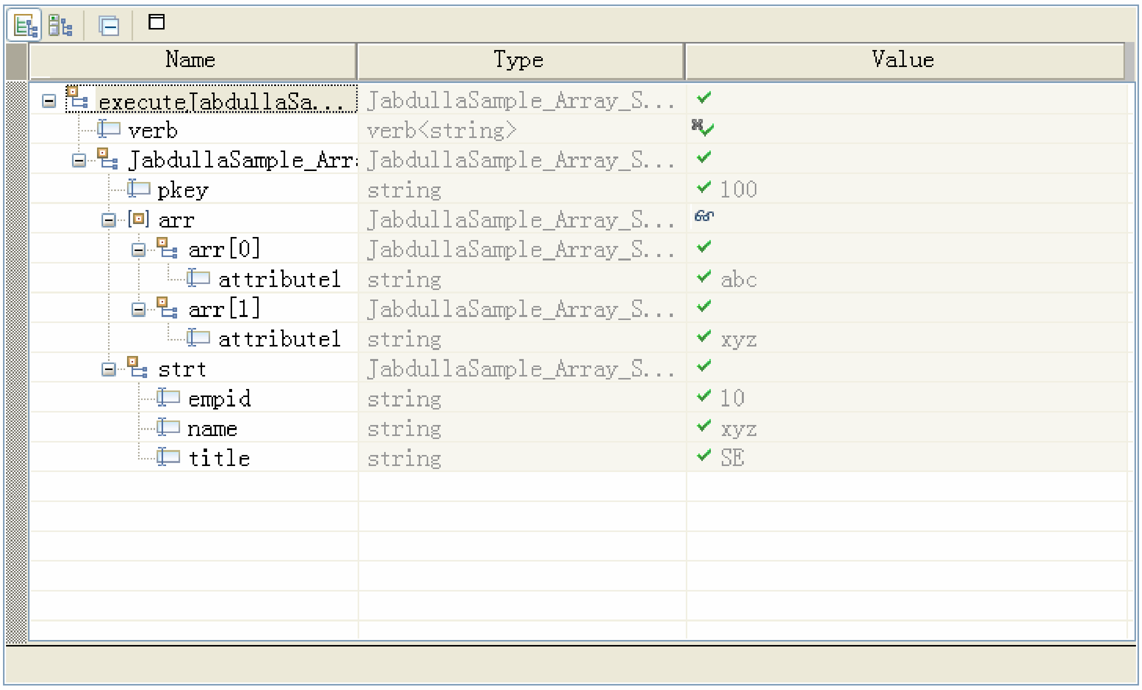
Mode: Run

□ Use this location as the default and do not ask again

To change the default location elsewhere, open the Properties window for the project and select Business Integration > Integration Test Client.

8. 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)

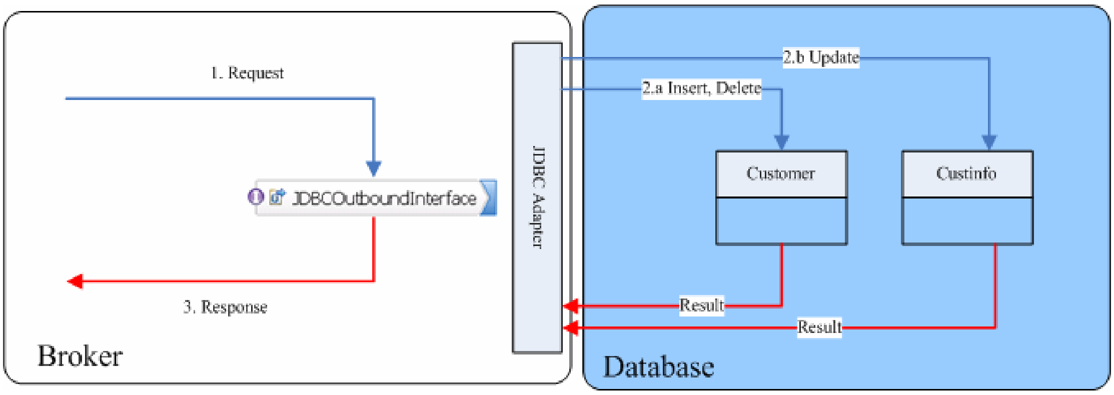
This tutorial demonstrates how to create batch SQL business object to execute multiple SQL statements using WebSphere Adapter for JDBC 7.5.0.0.

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.

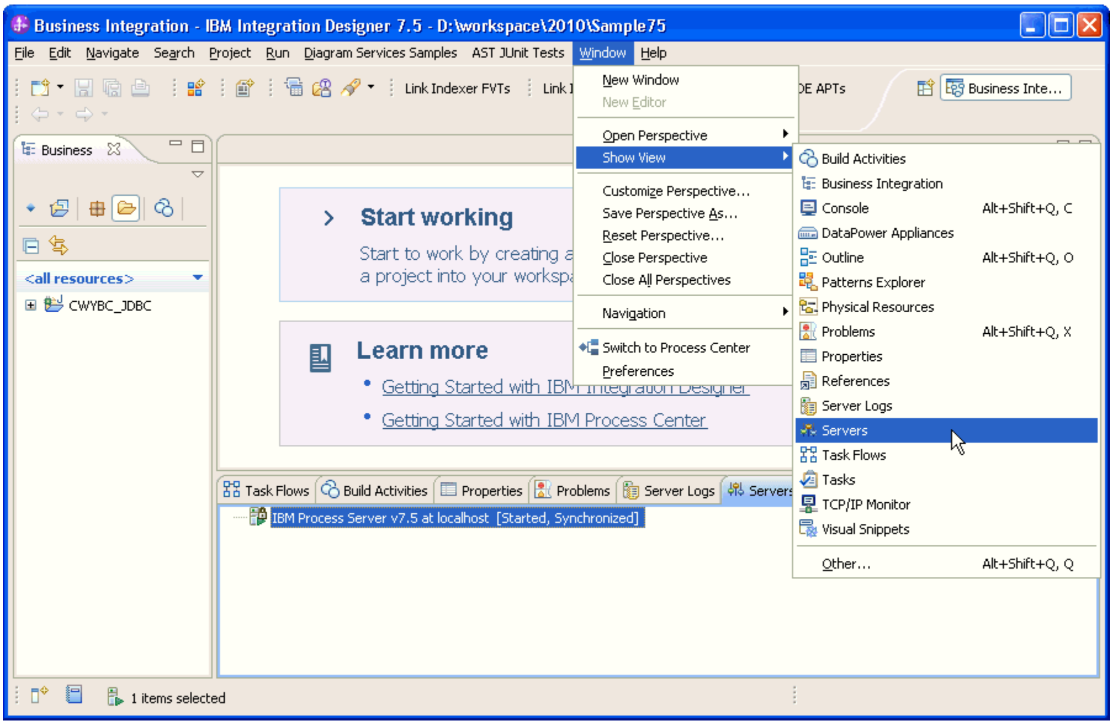
```
INSERT INTO CUSTINFO (ccode, cdata) values('Test1',
'ABC');
```

Create an authentication alias

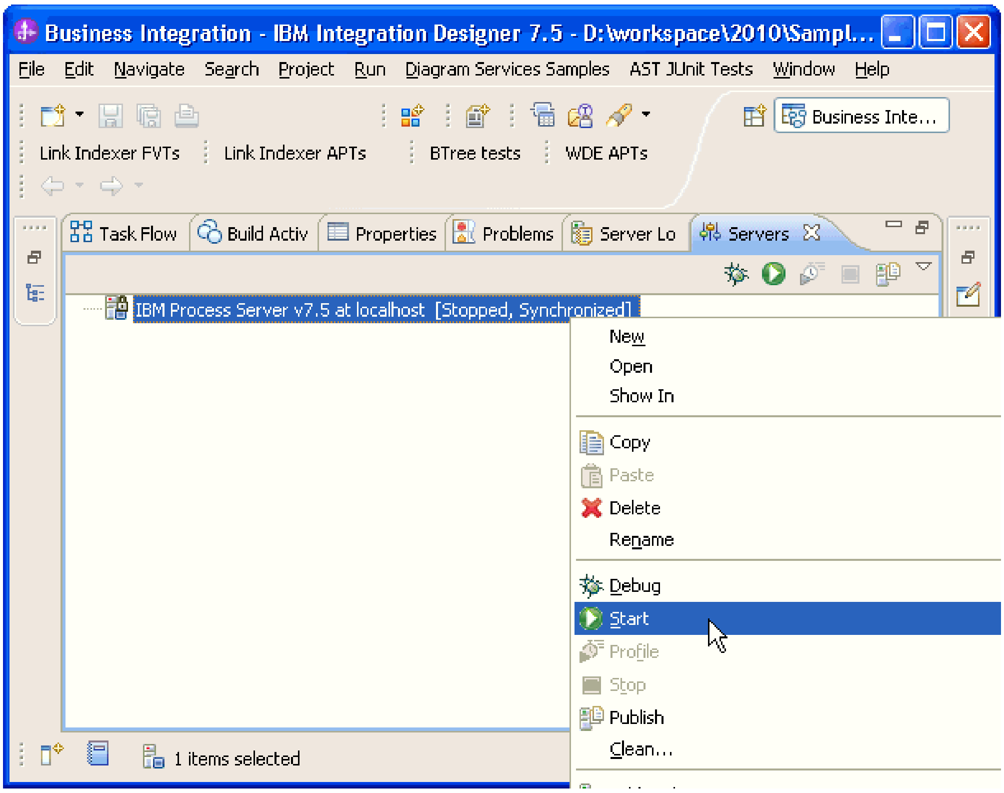
The authentication alias needs to be set because the data source created in the next section uses the username and password set in the authentication alias to connect to the database.

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

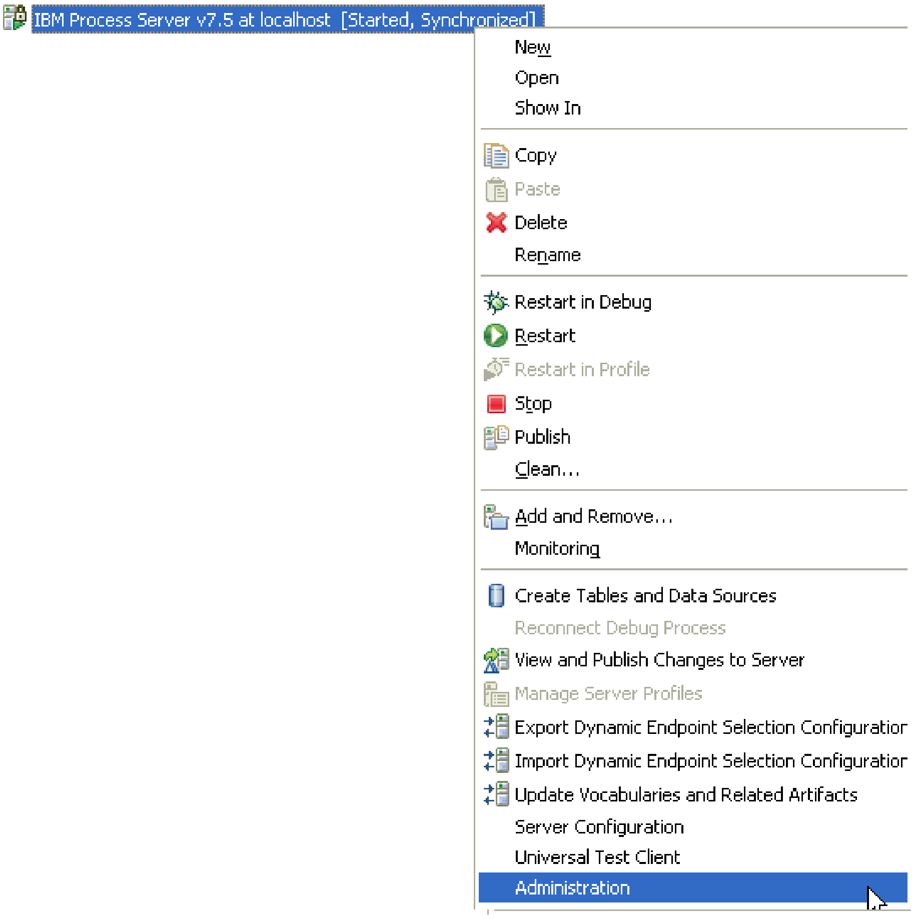
1. In IBM Integration Designer, switch to the **Servers** view by selecting **Windows > Show View > Servers**.



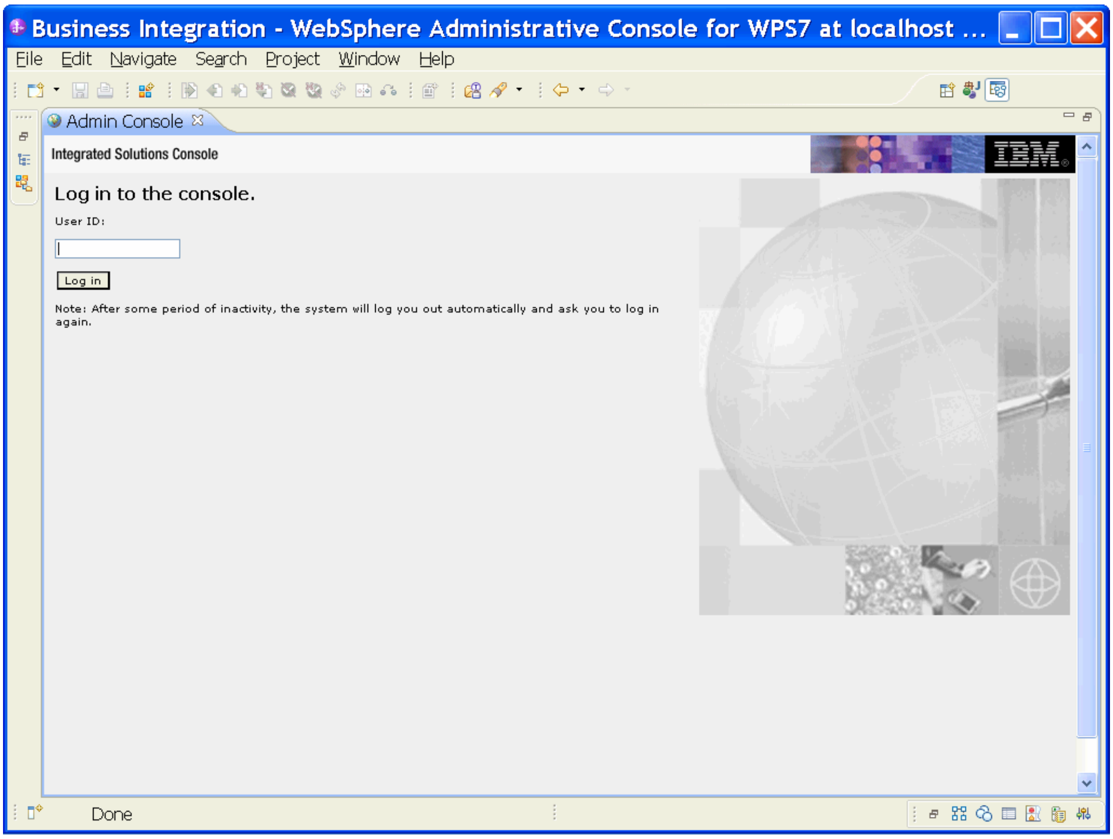
2. In the **Servers** view, right-click the server that you want to start and select **Start**.


Business Integration - IBM Integration Designer 7.5 - D:\workspace\2010\Sampl...
File Edit Navigate Search Project Run Diagram Services Samples AST JUnit Tests Window Help
Business Inte...
Link Indexer FVTs
Link Indexer APTs
BTree tests
WDE APTs
Task Flow
Build Activ
Properties
Problems
Server Lo
Servers
IBM Process Server v7.5 at localhost [Stopped, Synchronized]
New
Open
Show In
Copy
Paste
Delete
Rename
Debug
Start
Profile
Stop
Publish
Clean...
1 items selected

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


IBM Process Server v7.5 at localhost [Started, Synchronized]
New
Open
Show In
Copy
Paste
Delete
Rename
Restart in Debug
Restart
Restart in Profile
Stop
Publish
Clean...
Add and Remove...
Monitoring
Create Tables and Data Sources
Reconnect Debug Process
View and Publish Changes to Server
Manage Server Profiles
Export Dynamic Endpoint Selection Configuratior
Import Dynamic Endpoint Selection Configuratior
Update Vocabularies and Related Artifacts
Server Configuration
Universal Test Client
Administration

4. Log on to the administrative console.

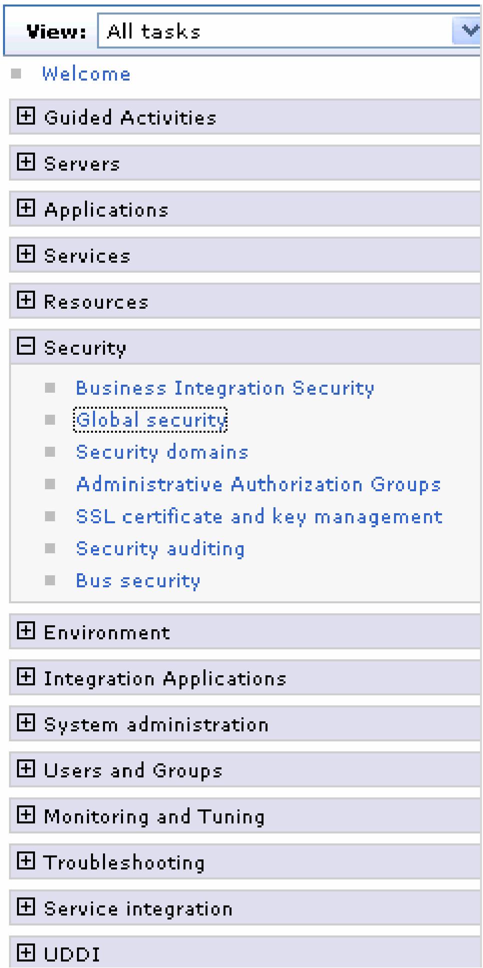

Business Integration - WebSphere Administrative Console for WPS7 at localhost ...
File
Edit
Navigate
Search
Project
Window
Help
Admin Console
Integrated Solutions Console
IBM
Log in
Done

Log in to the console.

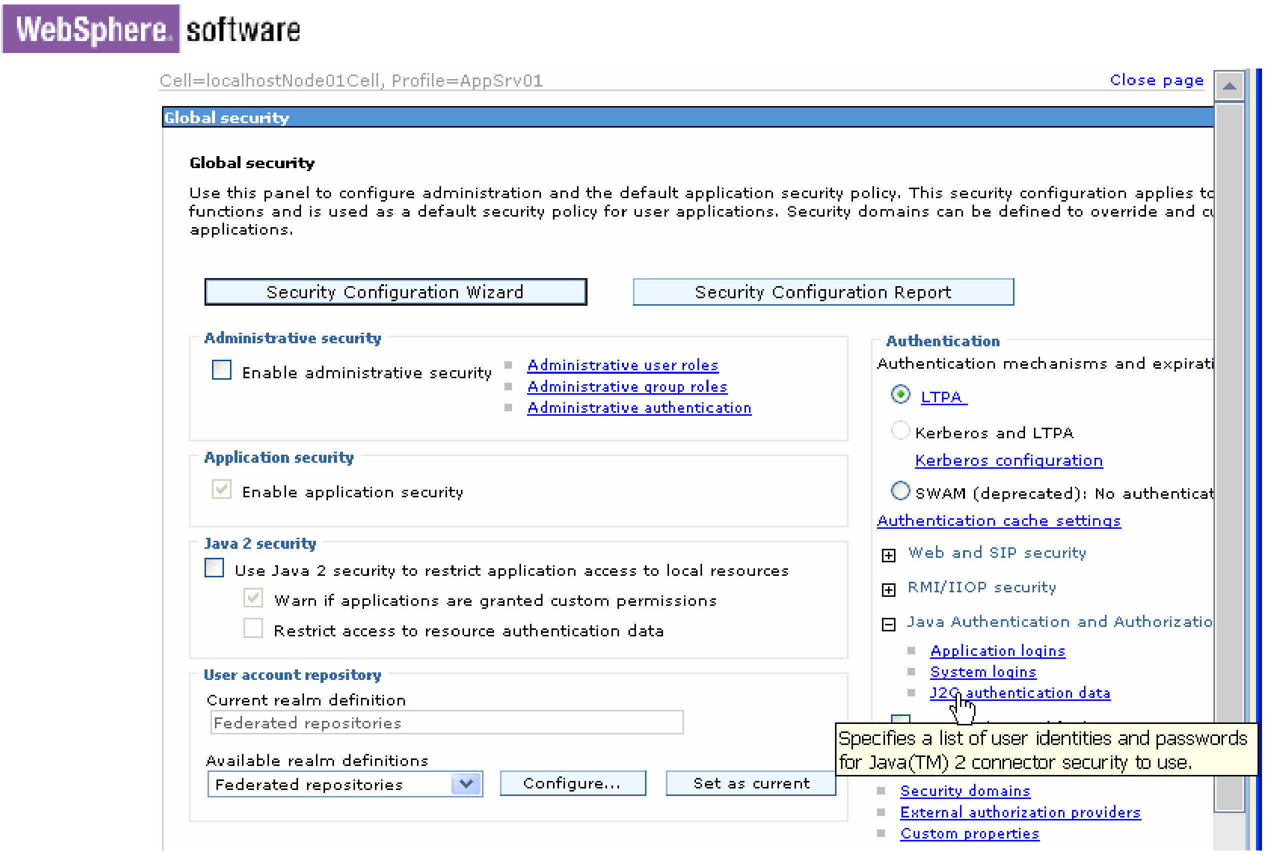
User ID:

Note: After some period of inactivity, the system will log you out automatically and ask you to log in again.

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



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


WebSphere software
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 cu 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

Authentication

Authentication mechanisms and expirati

● LTPA

○ Kerberos and LTPA

Kerberos configuration

○ SWAM (deprecated): No authenticat

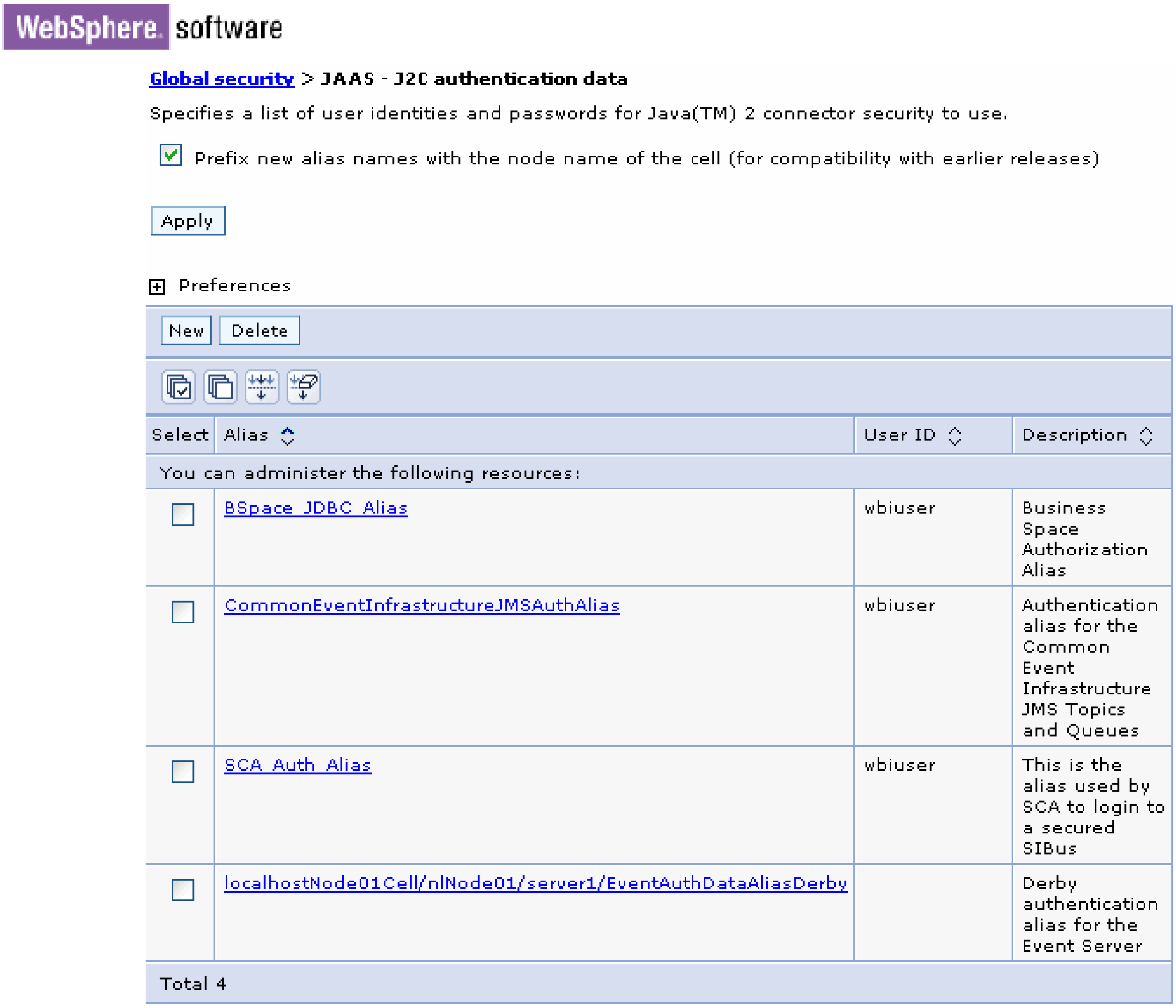
Authentication cache settings

- 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.



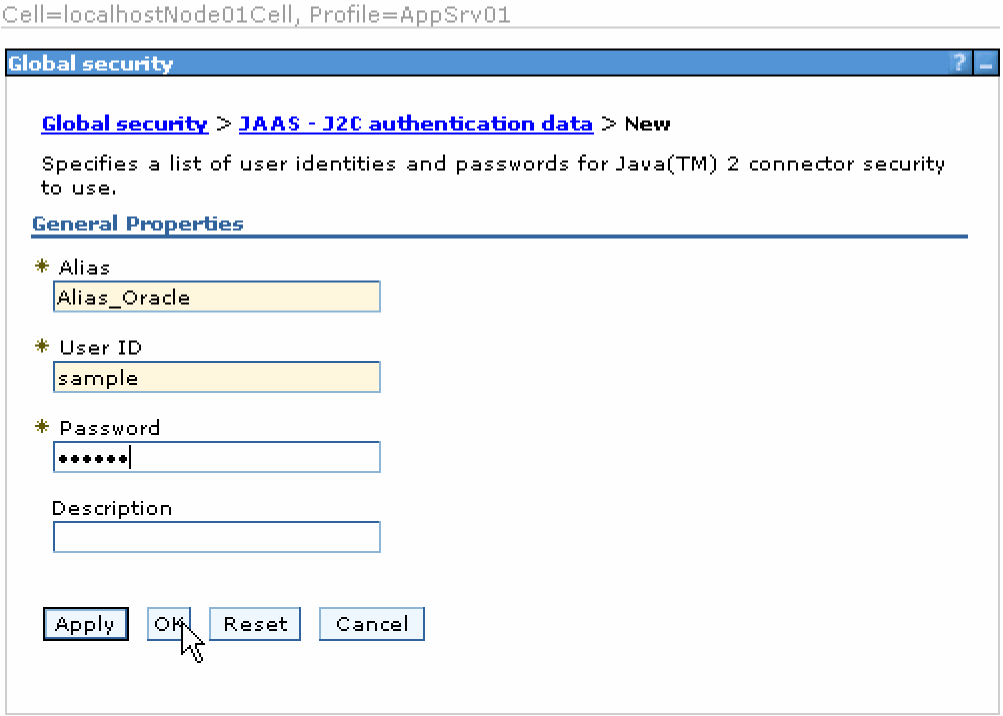
Global security > JAAS - J2C authentication data

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)

Select	Alias	User ID	Description
You can administer the following resources:			
□	BSpace_JDBC_Alias	wbiuser	Business Space Authorization Alias
□	CommonEventInfrastructureJMSAuthAlias	wbiuser	Authentication alias for the Common Event Infrastructure JMS Topics and Queues
□	SCA_Auth_Alias	wbiuser	This is the alias used by SCA to login to a secured SIBus
□	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**.



Cell=localhostNode01Cell, Profile=AppSrv01

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

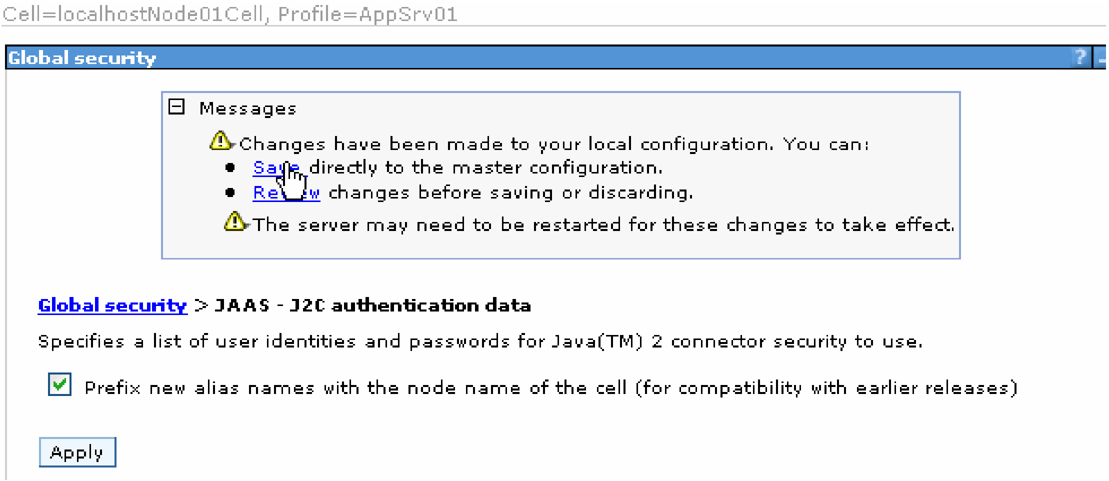
* Alias
Alias_Oracle

* User ID
sample

* Password
••••••

Description

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



Cell=localhostNode01Cell, Profile=AppSrv01

⊟ Messages

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

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

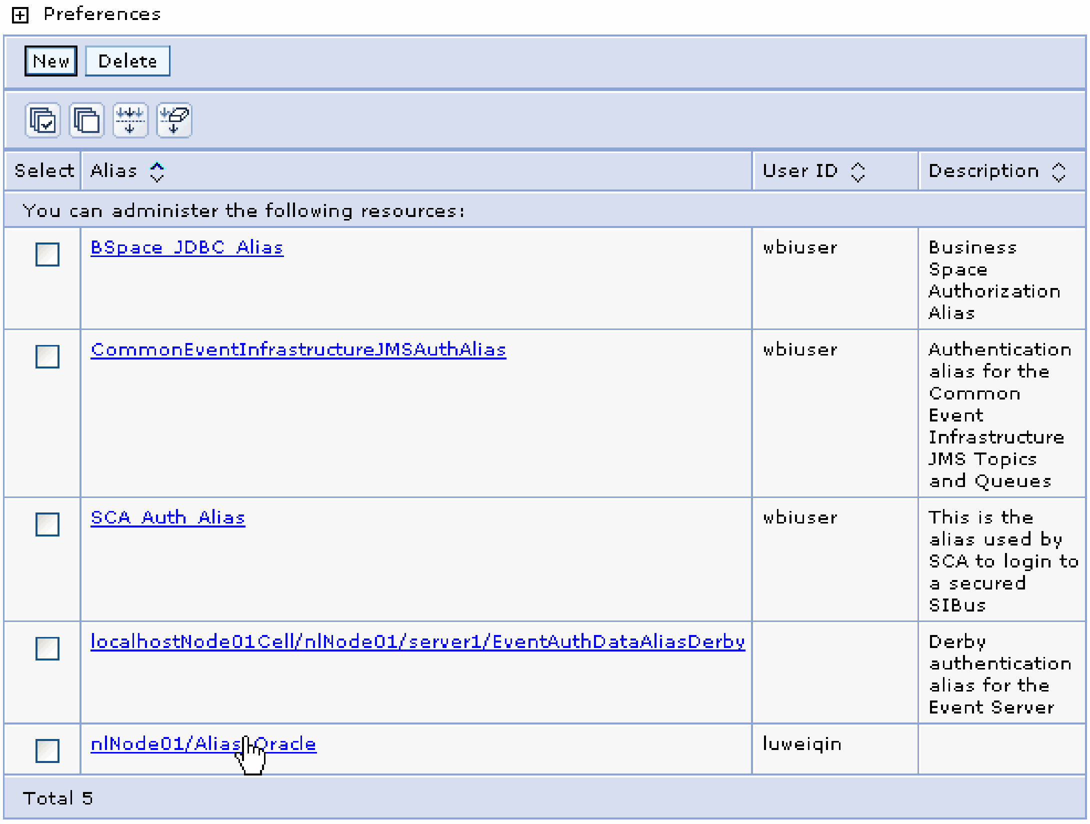
⚠ The server may need to be restarted for these changes to take effect.

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)

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



Select	Alias	User ID	Description
You can administer the following resources:			
☐	BSpace_JDBC_Alias	wbiuser	Business Space Authorization Alias
☐	CommonEventInfrastructureJMSAuthAlias	wbiuser	Authentication alias for the Common Event Infrastructure JMS Topics and Queues
☐	SCA_Auth_Alias	wbiuser	This is the alias used by SCA to login to a secured SIBus
☐	localhostNode01Cell/nlNode01/server1/EventAuthDataAliasDerby		Derby authentication alias for the Event Server
☐	nlNode01/Alias_Oracle	luweiqin	
Total 5			

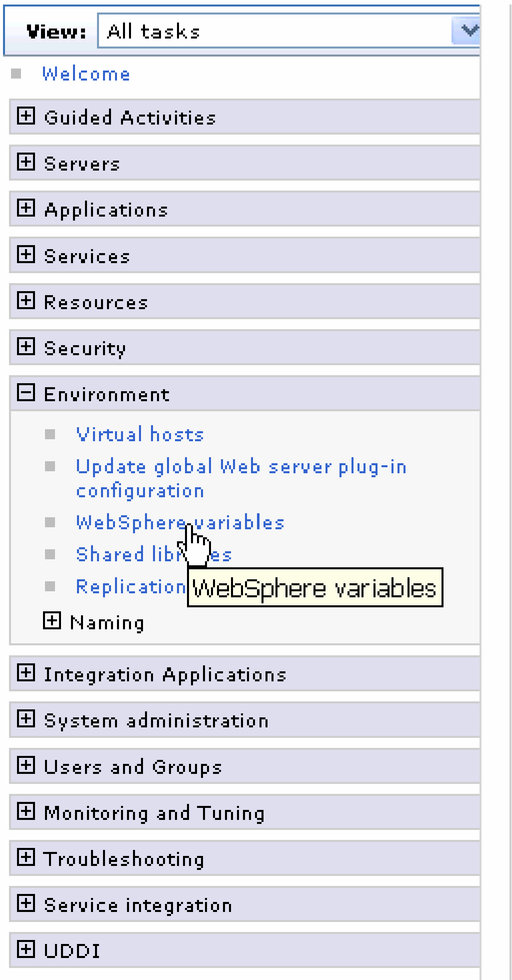
Create a data source

Create a data source in IBM 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 IBM Process Server administrative console.

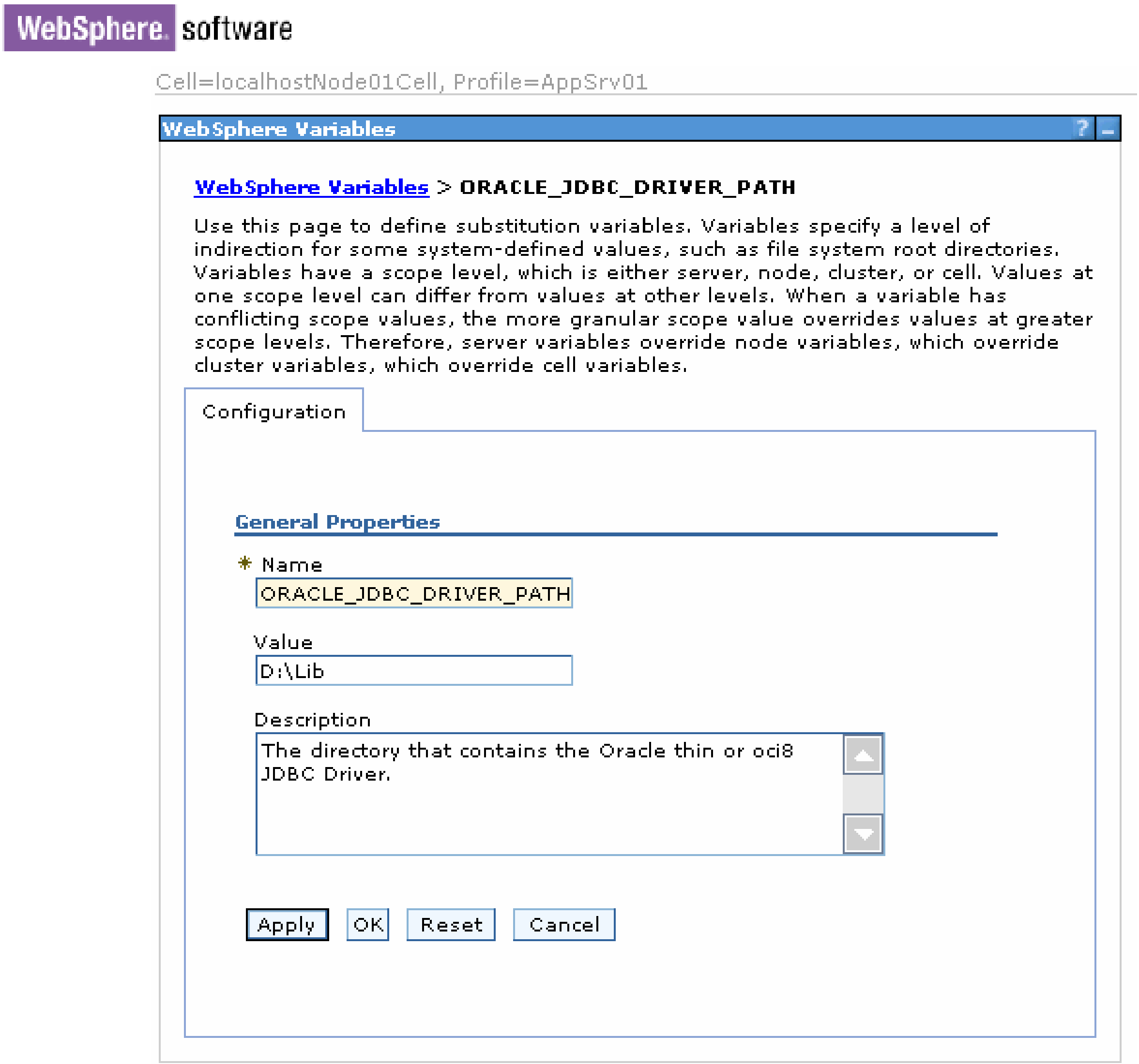
1. In the administrative console, select **Environment → WebSphere Variables**.



View: All tasks

- Welcome
- ⊞ Guided Activities
- ⊞ Servers
- ⊞ Applications
- ⊞ Services
- ⊞ Resources
- ⊞ Security
- ⊟ Environment
  - Virtual hosts
  - Update global Web server plug-in configuration
  - WebSphere variables
  - Shared libraries
  - Replication
  - ⊞ Naming
- ⊞ Integration Applications
- ⊞ System administration
- ⊞ Users and Groups
- ⊞ Monitoring and Tuning
- ⊞ Troubleshooting
- ⊞ Service integration
- ⊞ UDDI

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**.



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

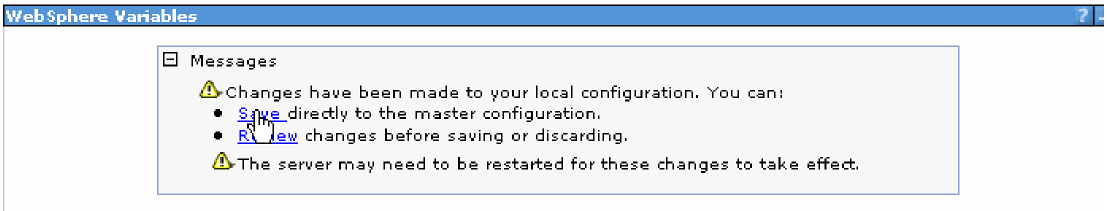
General Properties

* Name
ORACLE_JDBC_DRIVER_PATH

Value
D:\Lib

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

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


WebSphere Variables

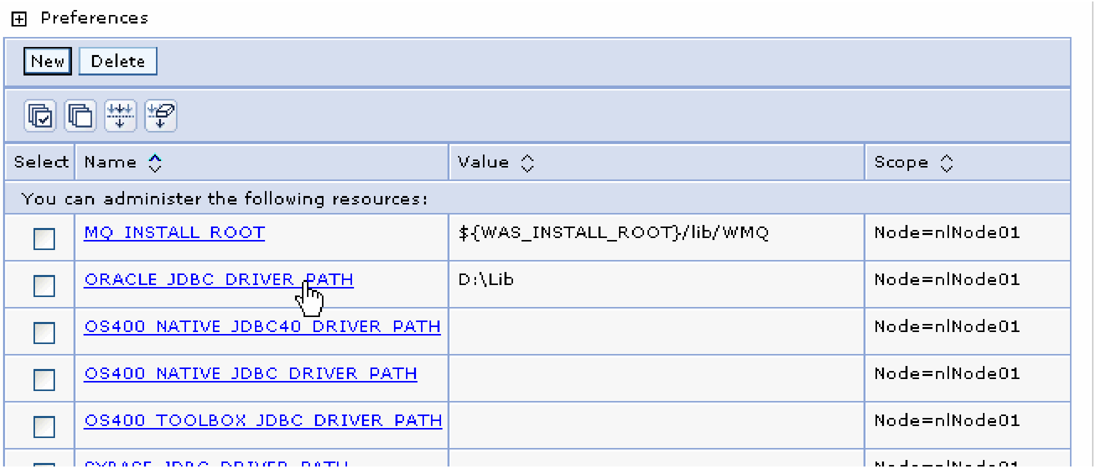
⊟ Messages

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

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

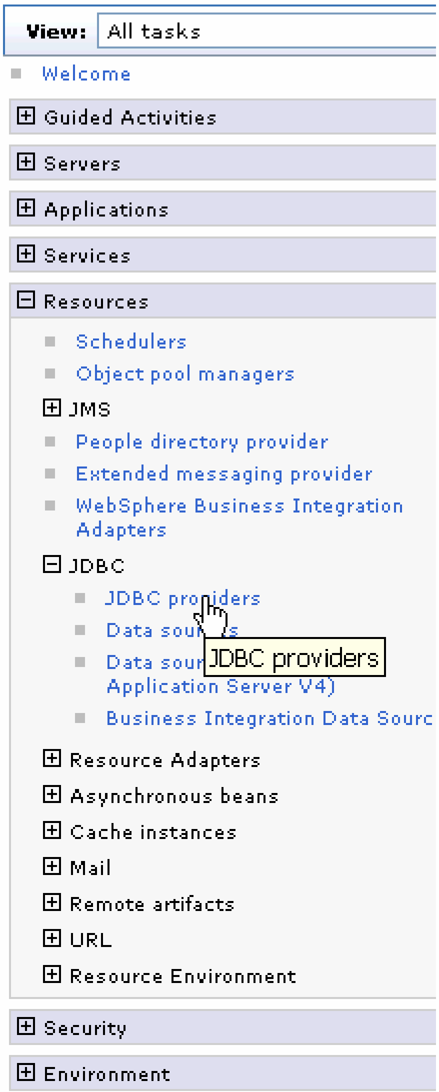
⚠ The server may need to be restarted for these changes to take effect.

The variable has been added and appears in the list.

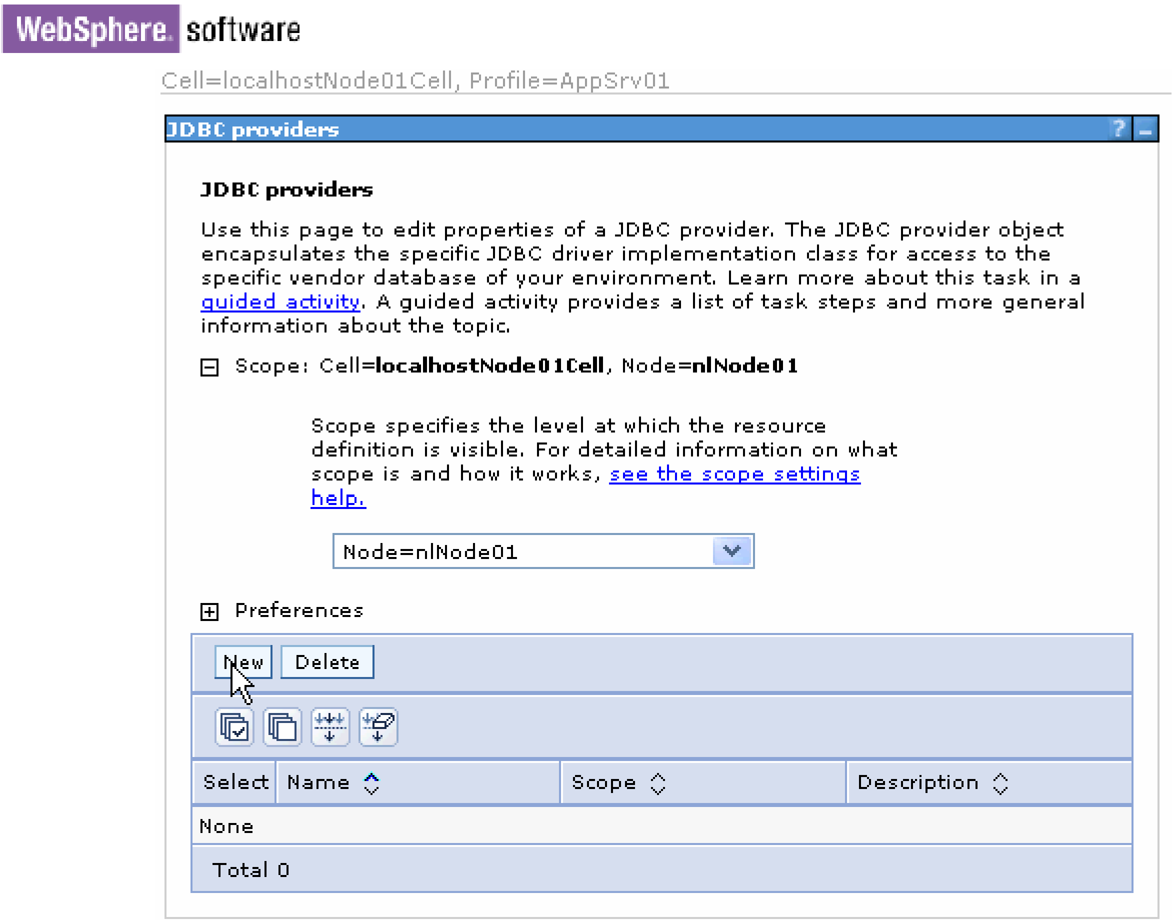

⊞ Preferences

Select	Name	Value	Scope
You can administer the following resources:			
□	MQ_INSTALL_ROOT	${WAS_INSTALL_ROOT}/lib/WMQ	Node=nlNode01
□	ORACLE_JDBC_DRIVER_PATH	D:\Lib	Node=nlNode01
□	OS400_NATIVE_JDBC40_DRIVER_PATH		Node=nlNode01
□	OS400_NATIVE_JDBC_DRIVER_PATH		Node=nlNode01
□	OS400_TOOLBOX_JDBC_DRIVER_PATH		Node=nlNode01
□	SYBASE_JDBC_DRIVER_PATH		Node=nlNode01

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



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



WebSphere software

Cell=localhostNode01Cell, Profile=AppSrv01

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=**nlNode01**

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.

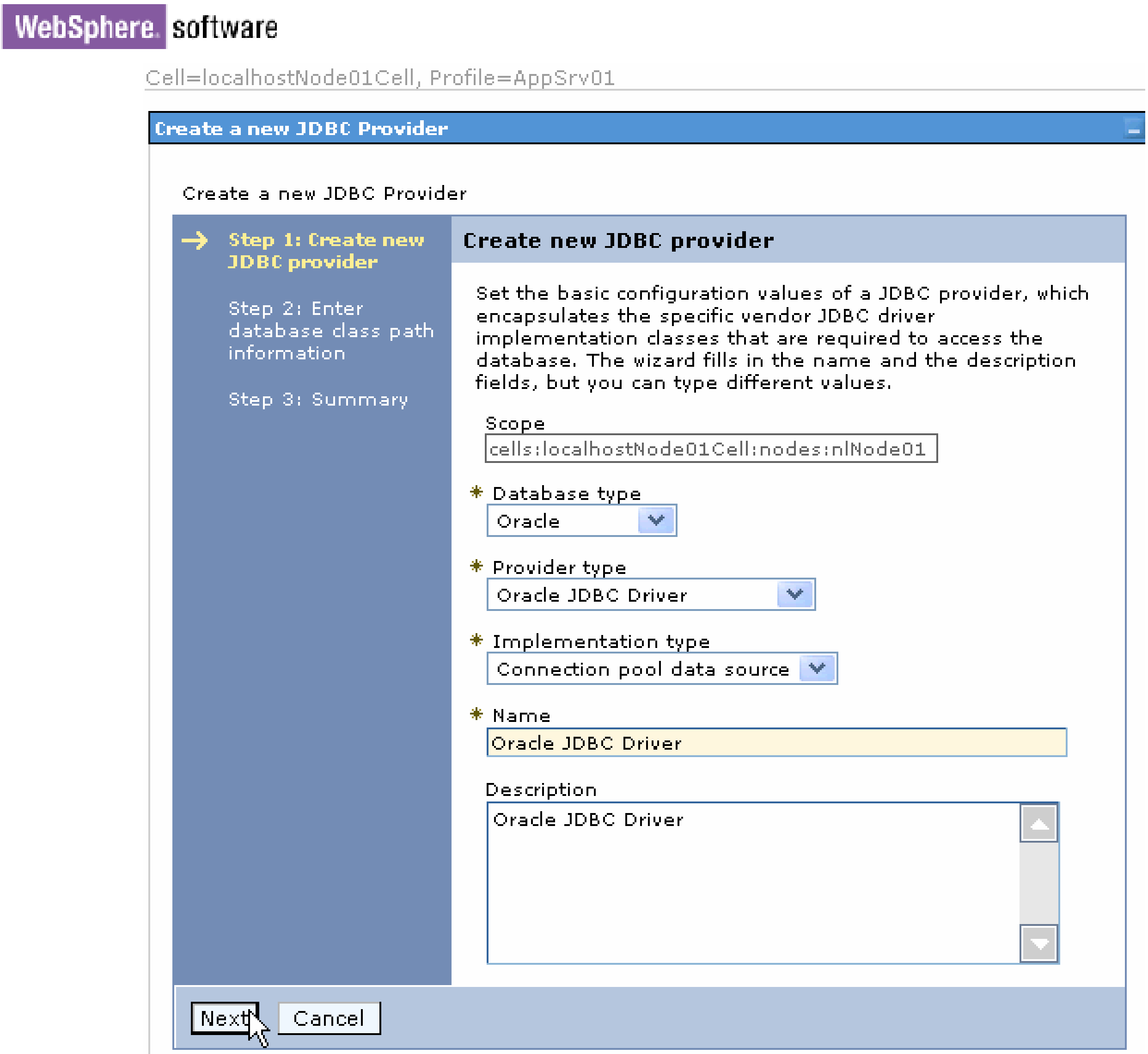
Node=nlNode01

⊞ 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**.



WebSphere. software
Cell=localhostNode01Cell, Profile=AppSrv01
Create a new JDBC Provider
Next
Cancel

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:nlNode01

* Database type
Oracle

* Provider type
Oracle JDBC Driver

* Implementation type
Connection pool data source

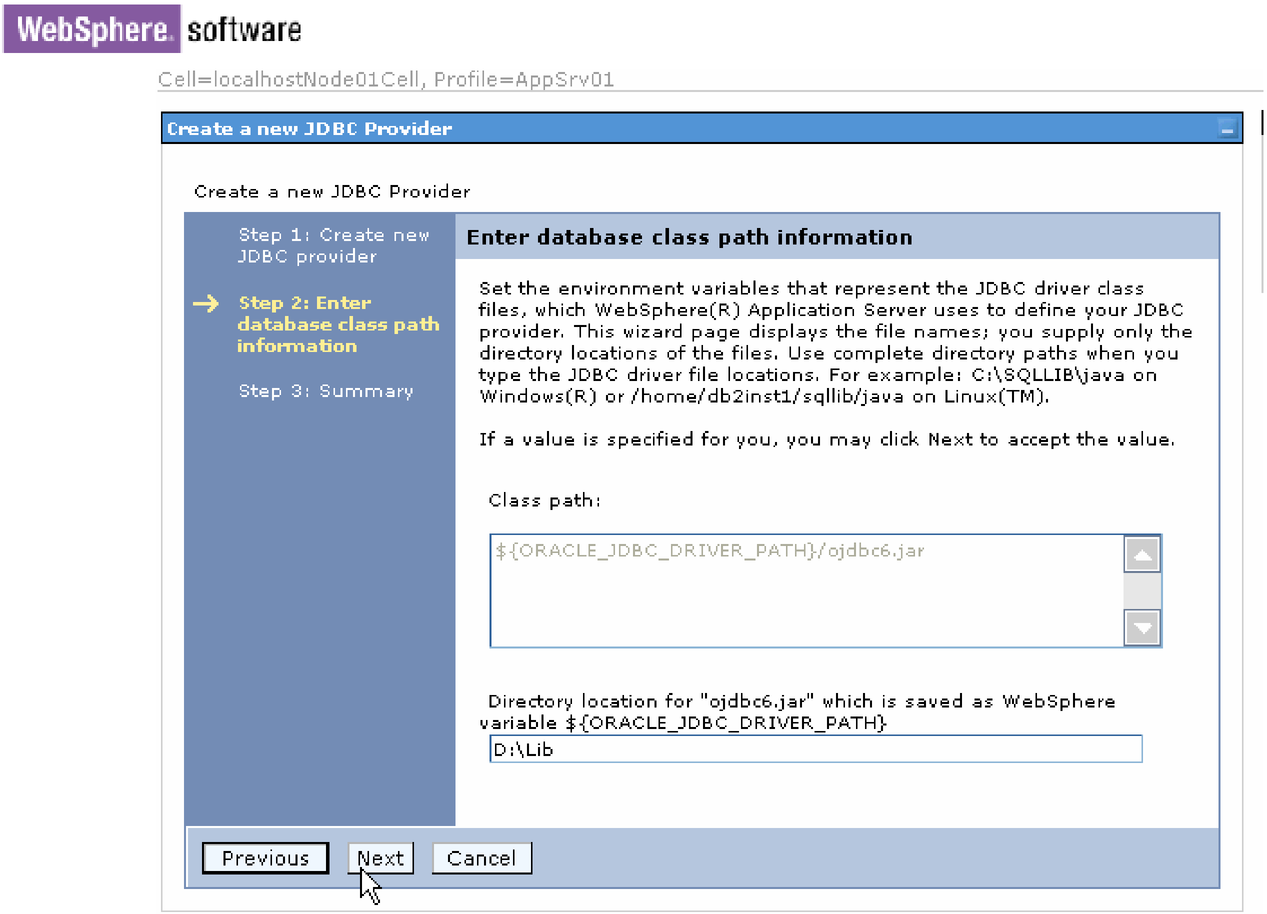
* Name
Oracle JDBC Driver

Description
Oracle JDBC Driver

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**.



Cell=localhostNode01Cell, Profile=AppSrv01

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: 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.

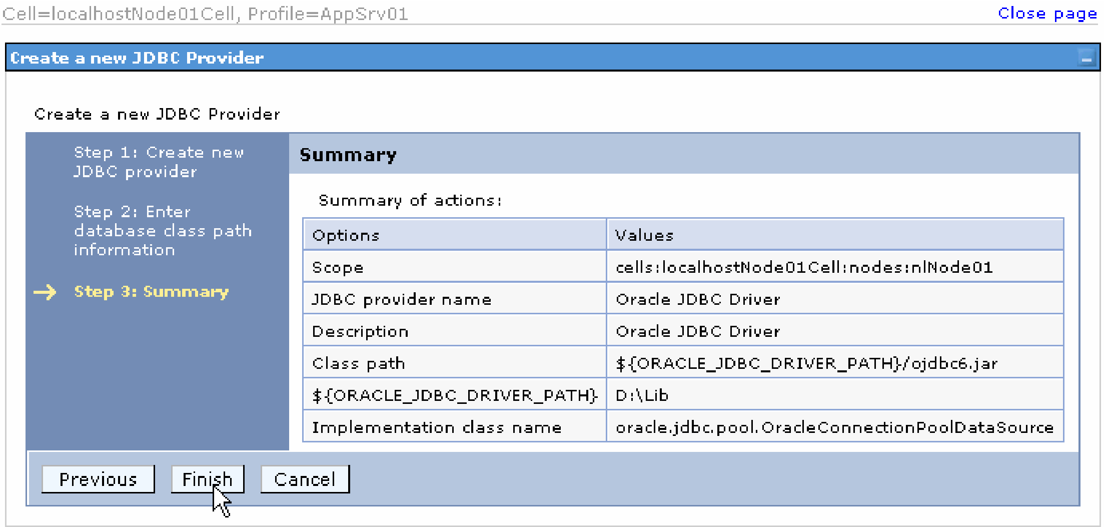
Class path:

${ORACLE_JDBC_DRIVER_PATH}/ojdbc6.jar

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

D:\Lib

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



Cell=localhostNode01Cell, Profile=AppSrv01

Create a new JDBC Provider

Create a new JDBC Provider

Step 1: Create new JDBC provider

Step 2: Enter database class path information

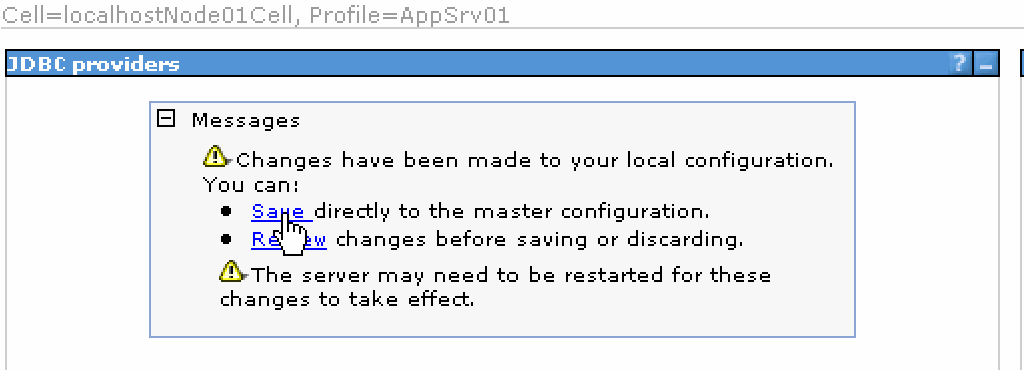
→ **Step 3: Summary**

Summary

Summary of actions:

Options	Values
Scope	cells:localhostNode01Cell:nodes:nlNode01
JDBC provider name	Oracle JDBC Driver
Description	Oracle JDBC Driver
Class path	${ORACLE_JDBC_DRIVER_PATH}/ojdbc6.jar
${ORACLE_JDBC_DRIVER_PATH}	D:\Lib
Implementation class name	oracle.jdbc.pool.OracleConnectionPoolDataSource

10. Click **Save**.



Cell=localhostNode01Cell, Profile=AppSrv01

JDBC providers

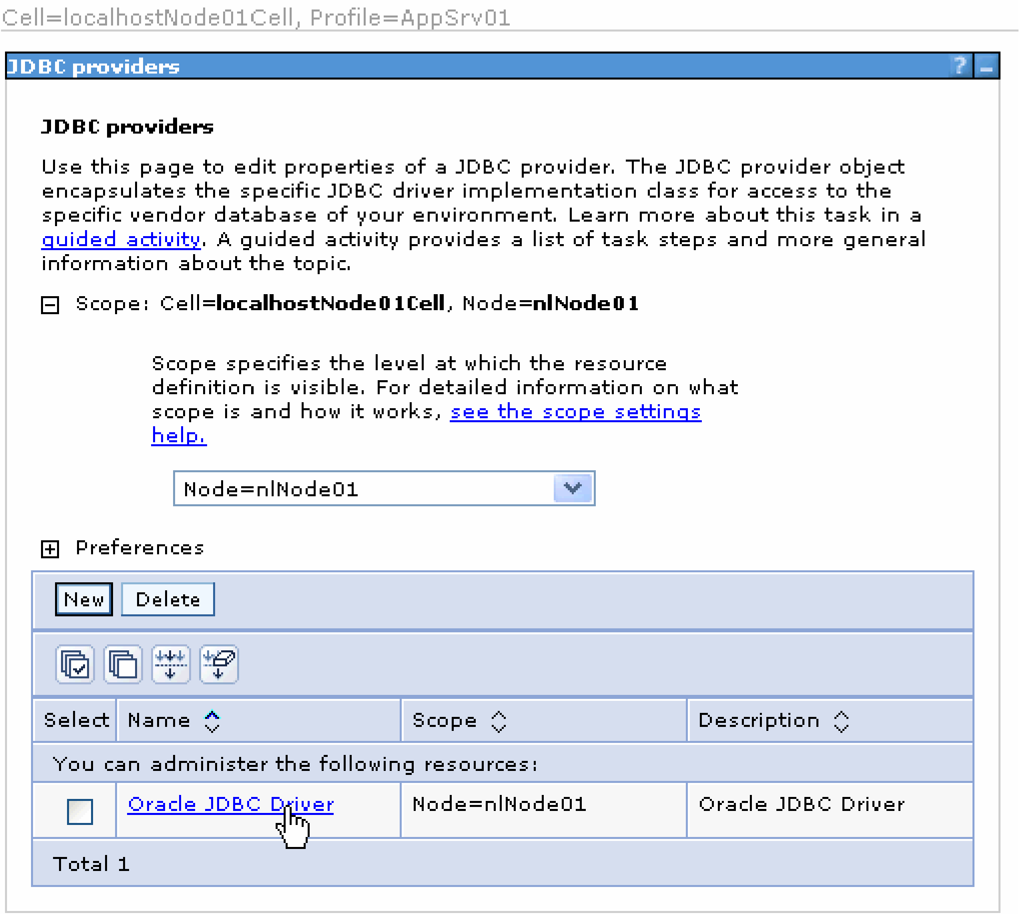
⊟ Messages

⚠ 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

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=**nlNode01**

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.

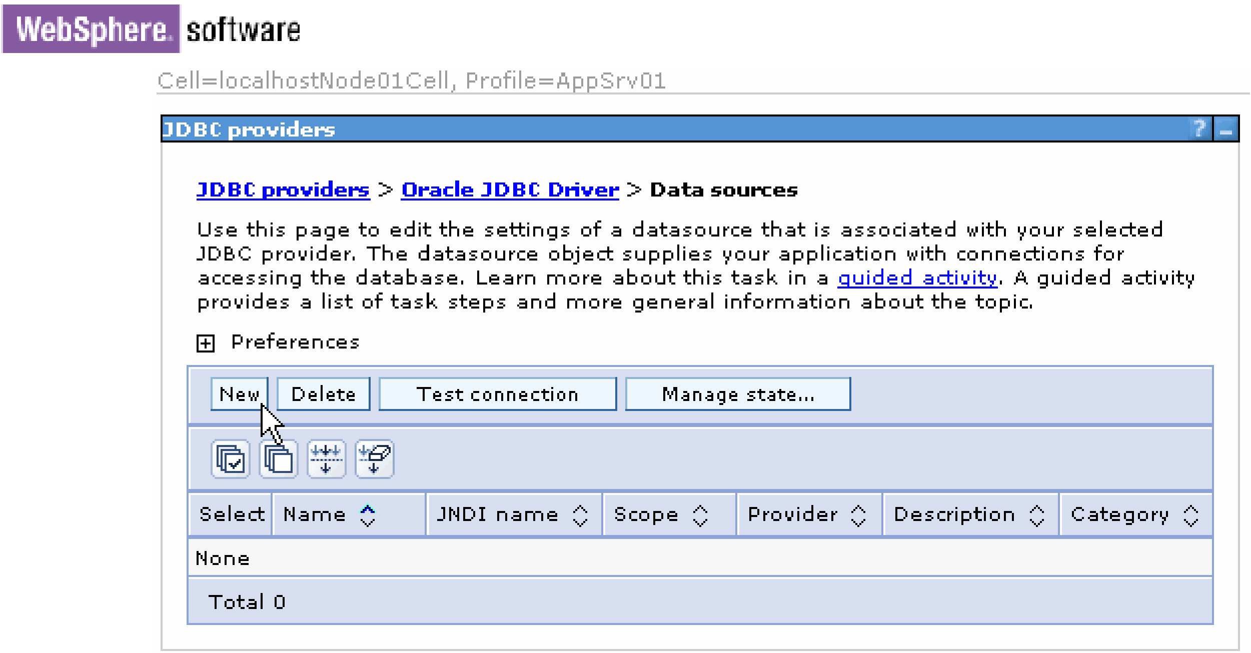
Node=nlNode01

⊞ Preferences

New Delete

Select	Name	Scope	Description
You can administer the following resources:			
□	Oracle JDBC Driver	Node=nlNode01	Oracle JDBC Driver
Total 1			

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



Cell=localhostNode01Cell, Profile=AppSrv01

JDBC providers

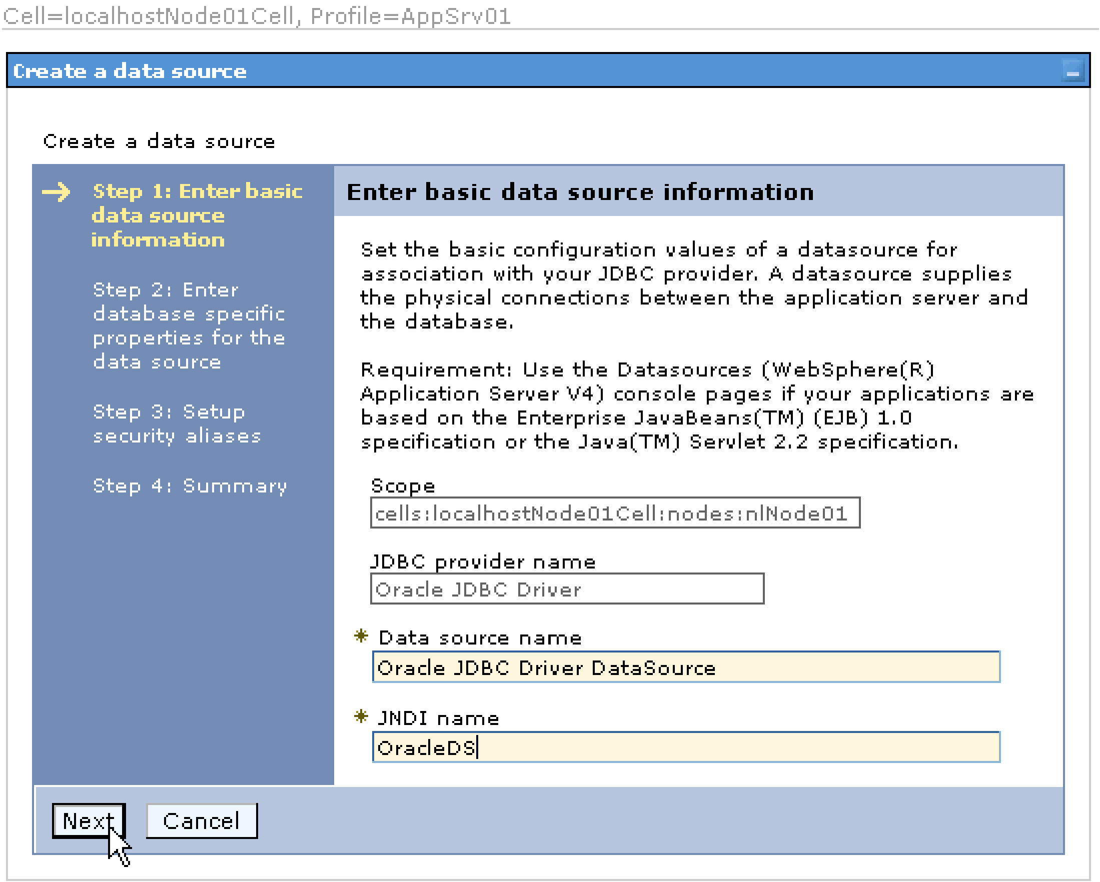
JDBC providers > Oracle JDBC Driver > **Data sources**

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

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.

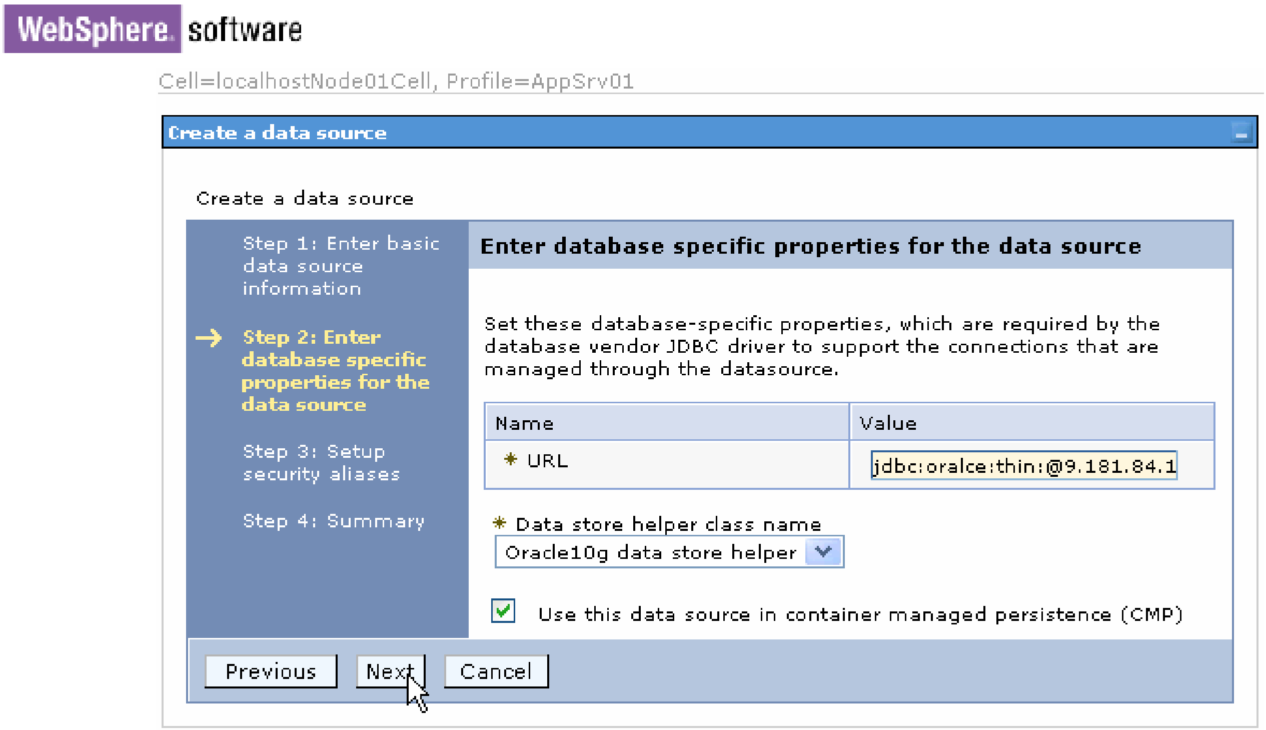
Scope
cells:localhostNode01Cell:nodes:nlNode01

JDBC provider name
Oracle JDBC Driver

* Data source name
Oracle JDBC Driver DataSource

* JNDI name
OracleDS

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

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

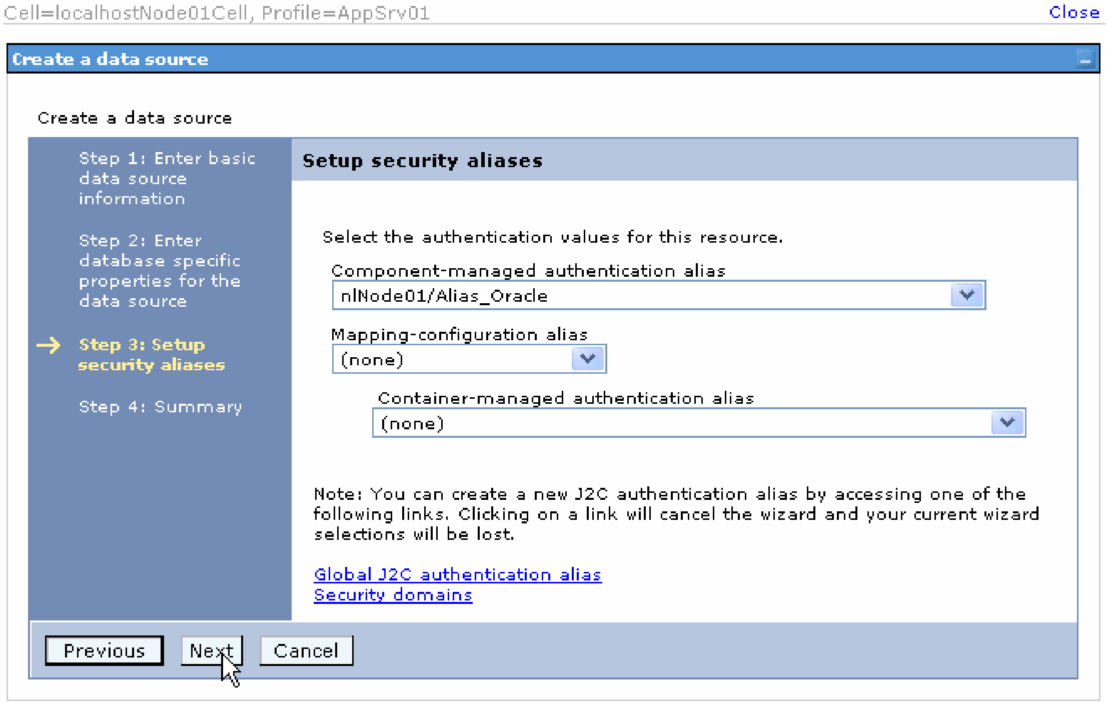
Set these database-specific properties, which are required by the database vendor JDBC driver to support the connections that are managed through the datasource.

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

* Data store helper class name
Oracle10g data store helper

☑ Use this data source in container managed persistence (CMP)

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

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

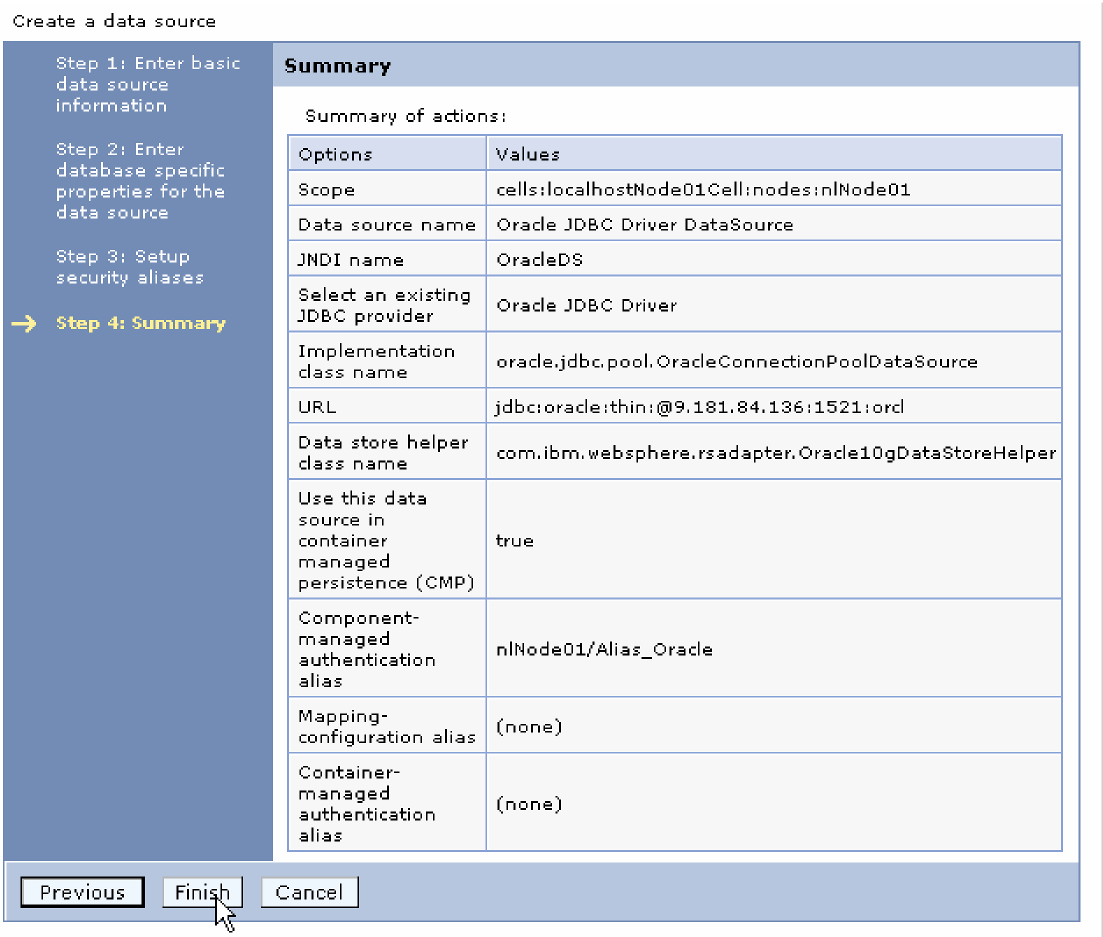
Mapping-configuration alias
(none)

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

15. 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

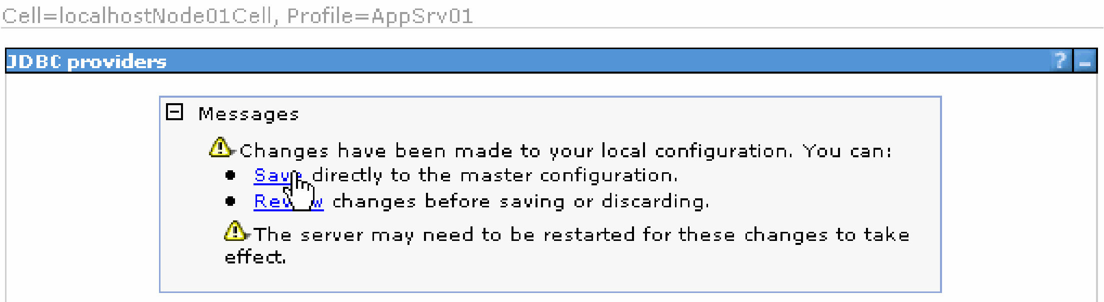
→ Step 4: Summary

Summary

Summary of actions:

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:orcl
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/Alias_Oracle
Mapping-configuration alias	(none)
Container-managed authentication alias	(none)

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



Cell=localhostNode01Cell, Profile=AppSrv01

JDBC providers

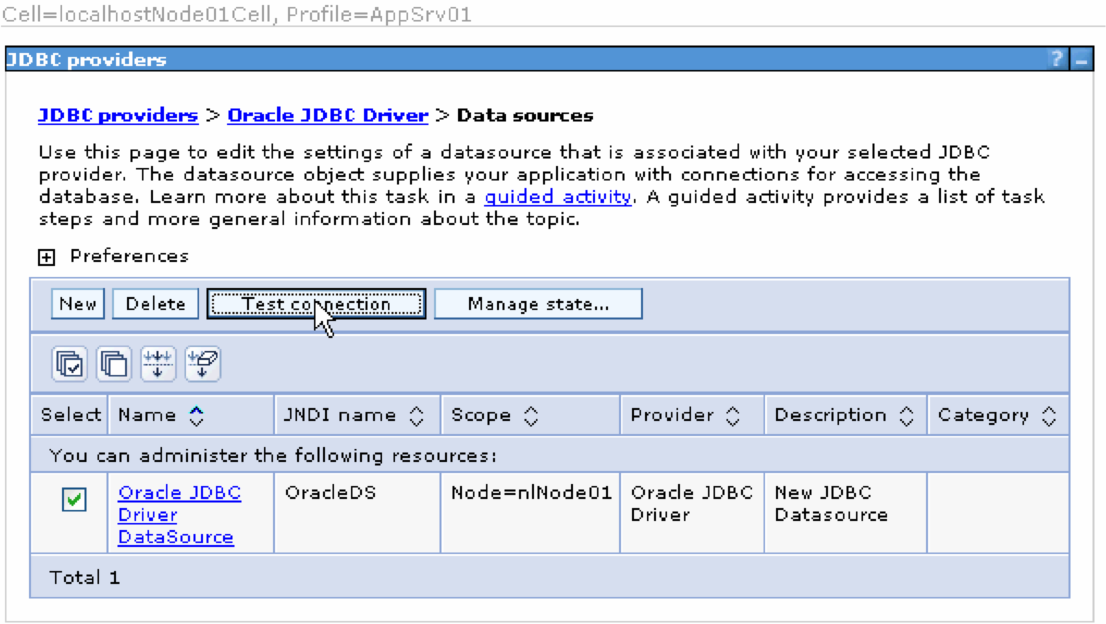
⊟ Messages

⚠ 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.

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



Cell=localhostNode01Cell, Profile=AppSrv01

JDBC providers

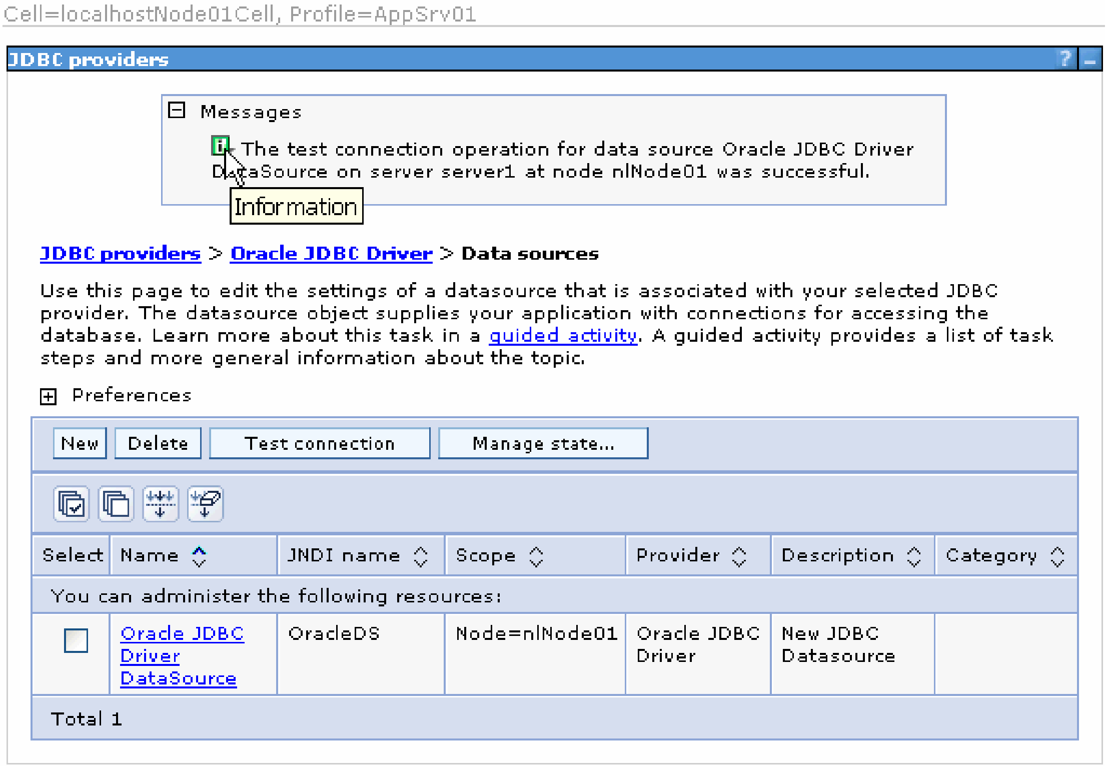
JDBC providers > Oracle JDBC Driver > Data sources

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

Select	Name	JNDI name	Scope	Provider	Description	Category
You can administer the following resources:						
☑	Oracle JDBC Driver DataSource	OracleDS	Node=nlNode01	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

⊟ Messages

The test connection operation for data source Oracle JDBC Driver DataSource on server server1 at node nlNode01 was successful.

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

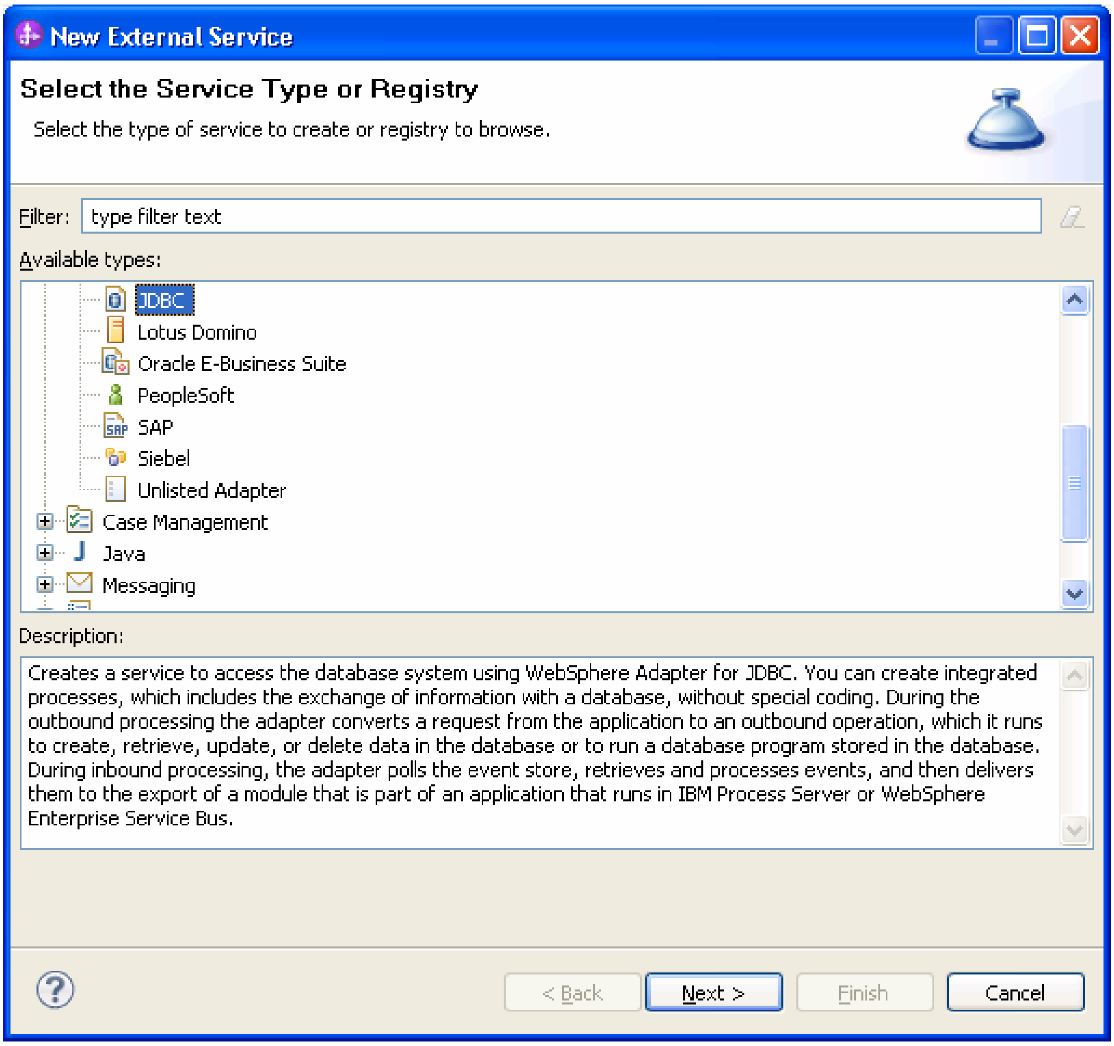
Select	Name	JNDI name	Scope	Provider	Description	Category
You can administer the following resources:						
□	Oracle JDBC Driver DataSource	OracleDS	Node=nlNode01	Oracle JDBC Driver	New JDBC Datasource	
Total 1						

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 IBM Integration Designer 7.5 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**.



Select the Service Type or Registry

Select the type of service to create or registry to browse.

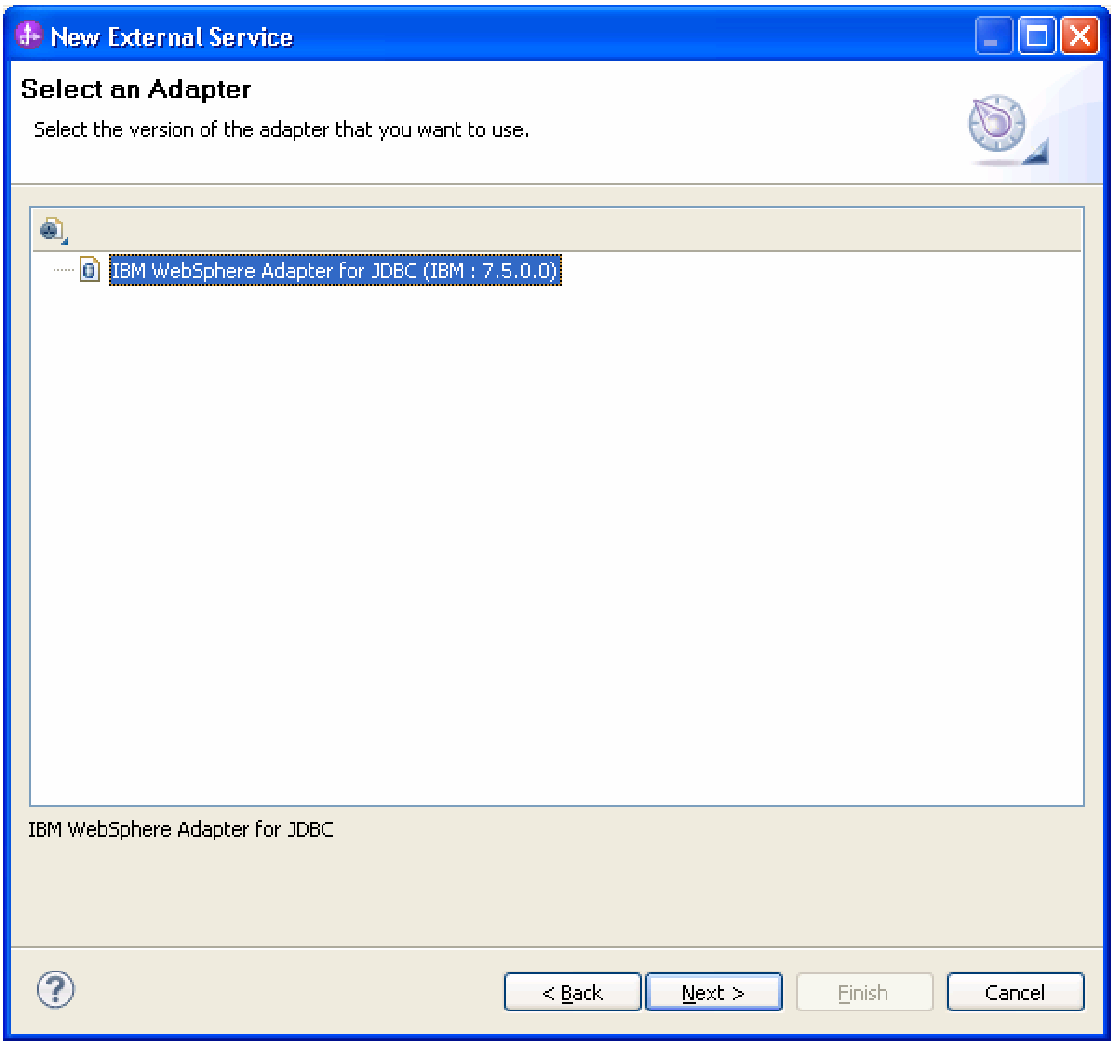
Filter: type filter text

Available types:

Description:

Creates a service to access the database system using WebSphere Adapter for JDBC. You can create integrated processes, which includes the exchange of information with a database, without special coding. During the outbound processing the adapter converts a request from the application to an outbound operation, which it runs to create, retrieve, update, or delete data in the database or to run a database program stored in the database. During inbound processing, the adapter polls the event store, retrieves and processes events, and then delivers them to the export of a module that is part of an application that runs in IBM Process Server or WebSphere Enterprise Service Bus.

4. Select **IBM WebSphere Adapter for JDBC (IBM: 7.5.0.0)** and click **Next**.



New External Service

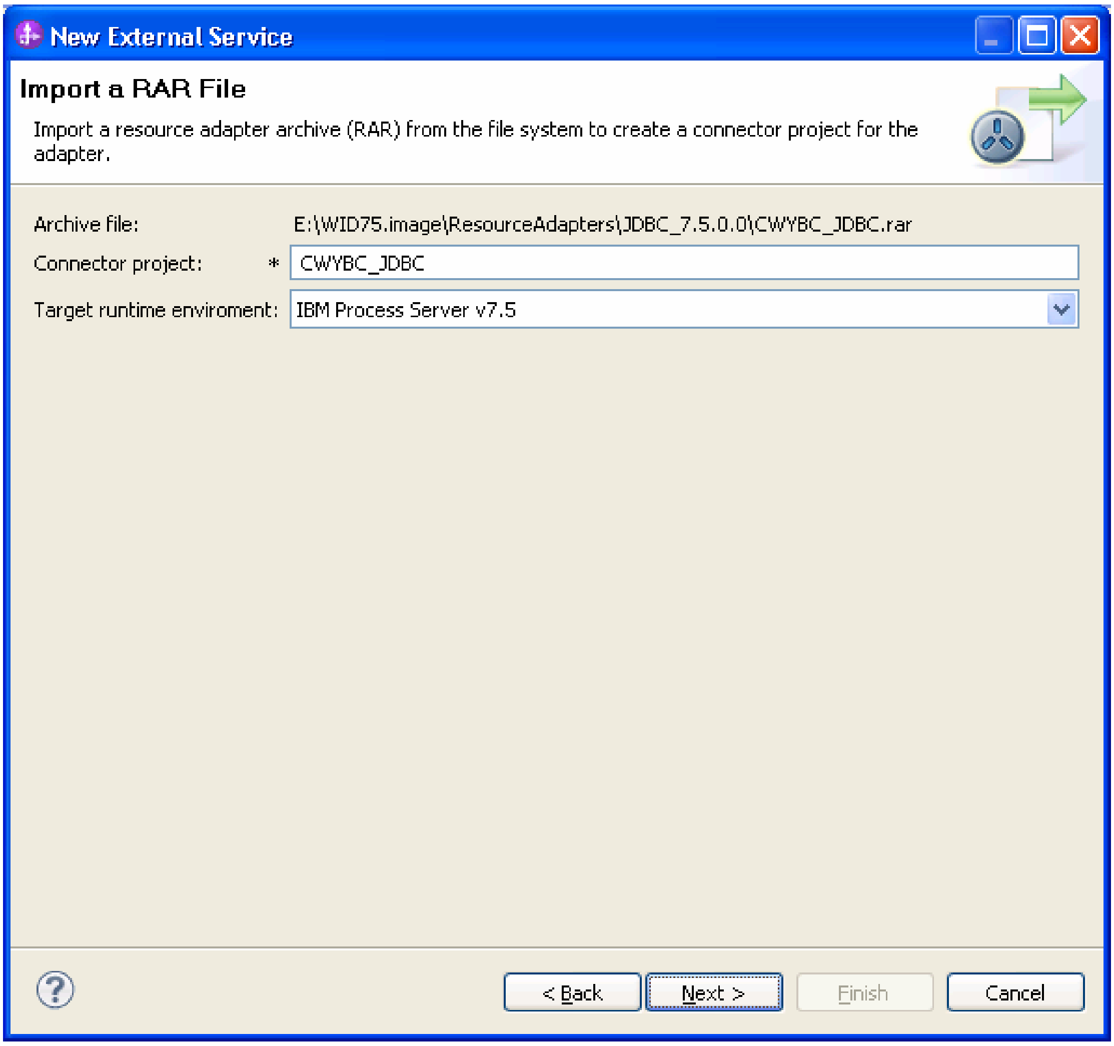
Select an Adapter

Select the version of the adapter that you want to use.

IBM WebSphere Adapter for JDBC (IBM : 7.5.0.0)

IBM WebSphere Adapter for JDBC

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



Import a RAR File

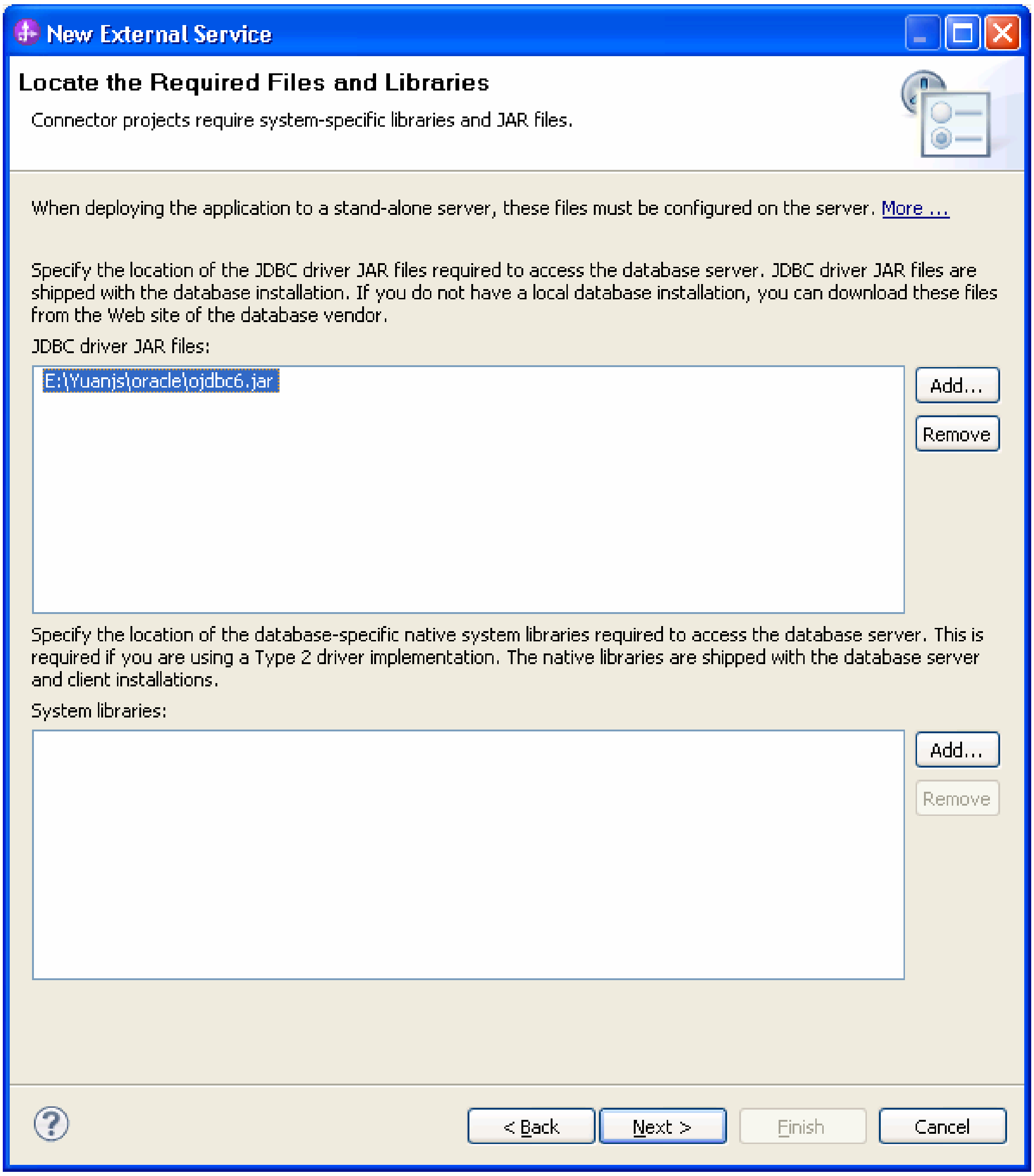
Import a resource adapter archive (RAR) from the file system to create a connector project for the adapter.

Archive file: E:\WID75.image\ResourceAdapters\JDBC_7.5.0.0\CWYBC_JDBC.rar

Connector project: * CWYBC_JDBC

Target runtime enviroment: IBM Process Server v7.5

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**.



New External Service

Locate the Required Files and Libraries

Connector projects require system-specific libraries and JAR files.

When deploying the application to a stand-alone server, these files must be configured on the server. More ...

Specify the location of the JDBC driver JAR files required to access the database server. JDBC driver JAR files are shipped with the database installation. If you do not have a local database installation, you can download these files from the Web site of the database vendor.

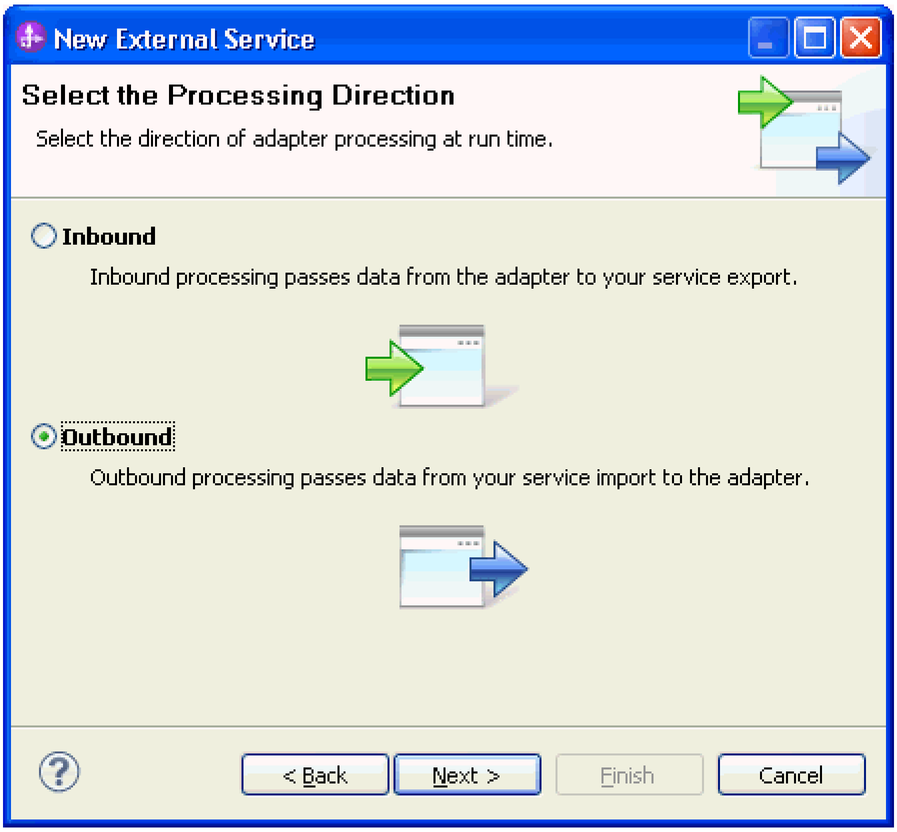
JDBC driver JAR files:

E:\Yuanjs\oracle\ojdbc6.jar

Specify the location of the database-specific native system libraries required to access the database server. This is required if you are using a Type 2 driver implementation. The native libraries are shipped with the database server and client installations.

System libraries:

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



New External Service

Select the Processing Direction

Select the direction of adapter processing at run time.

○ Inbound

Inbound processing passes data from the adapter to your service export.

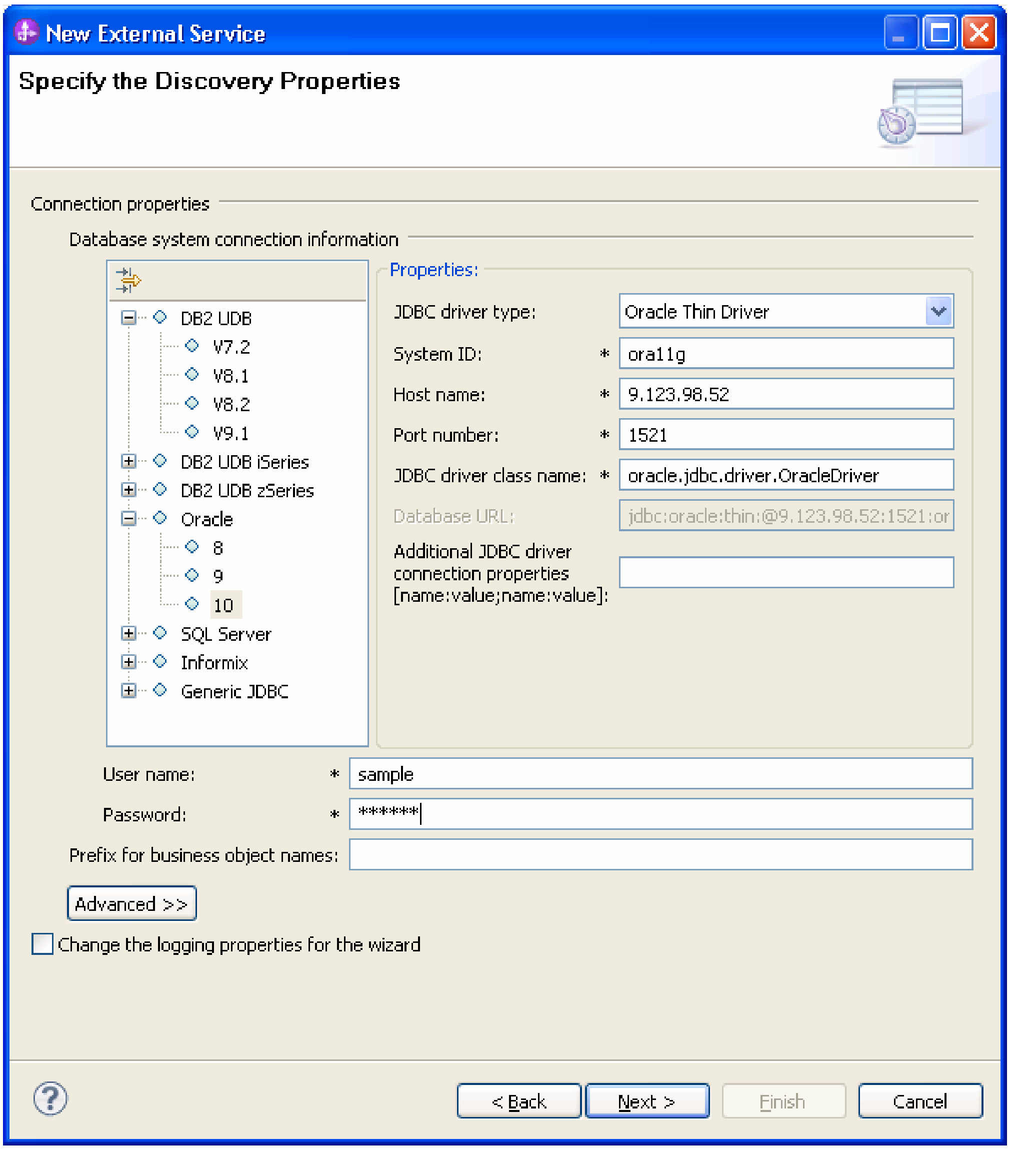
● Outbound

Outbound processing passes data from your service import to the adapter.

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**.



New External Service

Specify the Discovery Properties

Connection properties

Database system connection information

- DB2 UDB
  - V7.2
  - V8.1
  - V8.2
  - V9.1
- DB2 UDB iSeries
- DB2 UDB zSeries
- Oracle
  - 8
  - 9
  - 10
- SQL Server
- Informix
- Generic JDBC

Properties:

JDBC driver type: Oracle Thin Driver

System ID: * ora11g

Host name: * 9.123.98.52

Port number: * 1521

JDBC driver class name: * oracle.jdbc.driver.OracleDriver

Database URL: jdbc:oracle:thin:@9.123.98.52:1521:or

Additional JDBC driver connection properties [name:value;name:value]:

User name: * sample

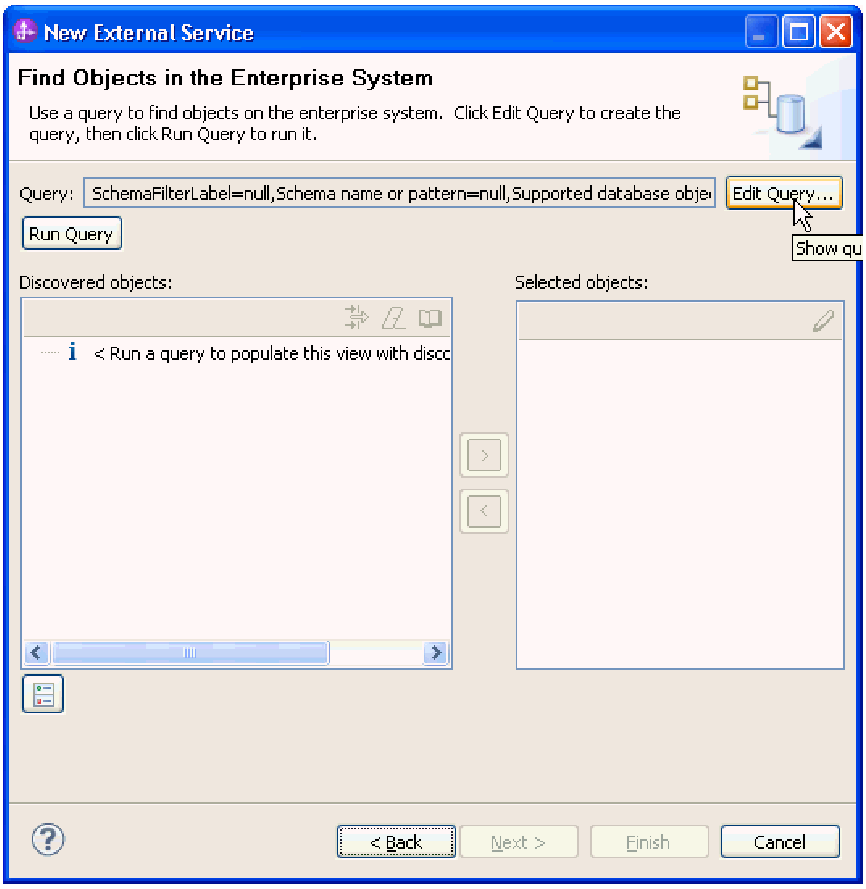
Password: * ******

Prefix for business object names:

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**.



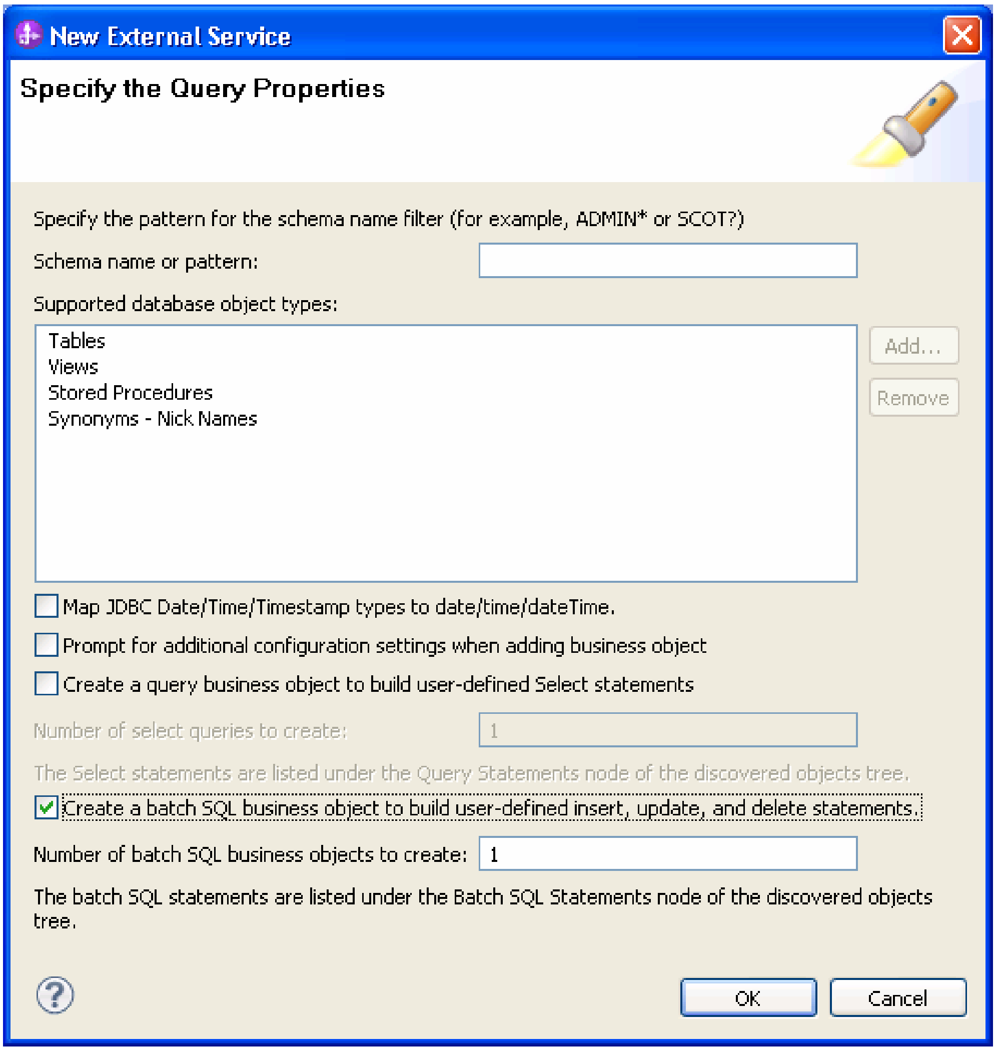
New External Service

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 obje

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**.



Specify the Query Properties

Specify the pattern for the schema name filter (for example, ADMIN* or SCOT?)

Schema name or pattern:

Supported database object types:

Tables
Views
Stored Procedures
Synonyms - Nick Names

□ Map JDBC Date/Time/Timestamp types to date/time/dateTime.

□ Prompt for additional configuration settings when adding business object

□ Create a query business object to build user-defined Select statements

Number of select queries to create: 1

The Select statements are listed under the Query Statements node of the discovered objects tree.

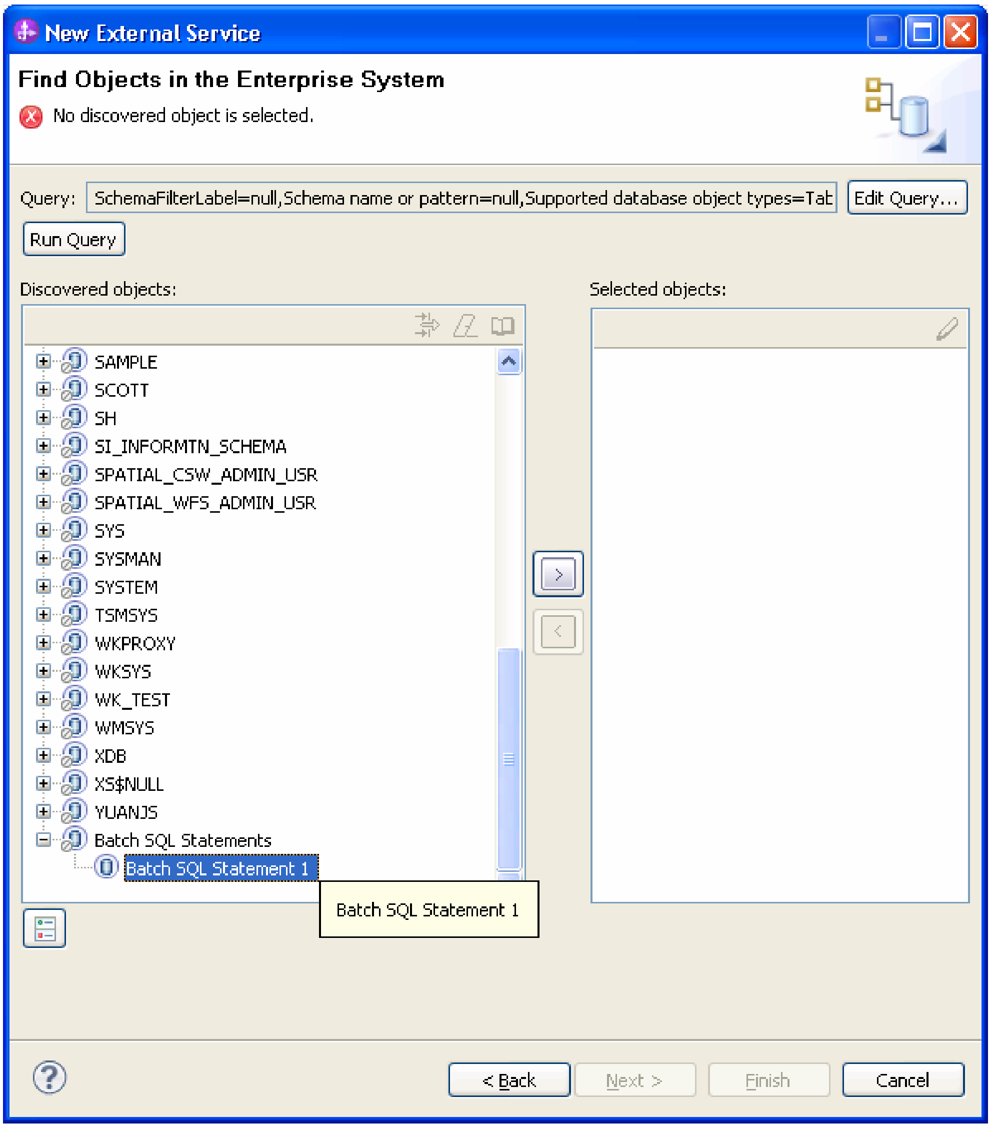
☑ Create a batch SQL business object to build user-defined insert, update, and delete statements.

Number of batch SQL business objects to create: 1

The batch SQL statements are listed under the Batch SQL Statements node of the discovered objects tree.

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

4. Expand the **Batch SQL Statements** node.



New External Service

Find Objects in the Enterprise System

No discovered object is selected.

Query: SchemaFilterLabel=null,Schema name or pattern=null,Supported database object types=Tat

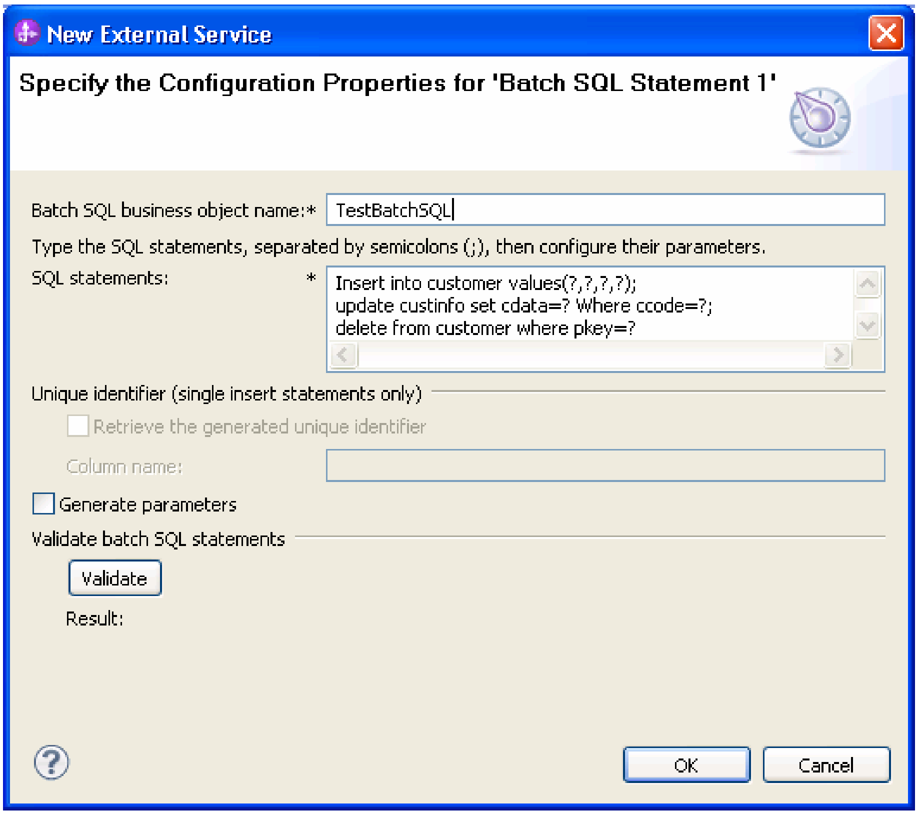
5. Select **Batch SQL Statement1** and click >.

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=?
```



New External Service

Specify the Configuration Properties for 'Batch SQL Statement 1'

Batch SQL business object name:* TestBatchSQL

Type the SQL statements, separated by semicolons (;), then configure their parameters.

SQL statements: *

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

Unique identifier (single insert statements only)

□ Retrieve the generated unique identifier

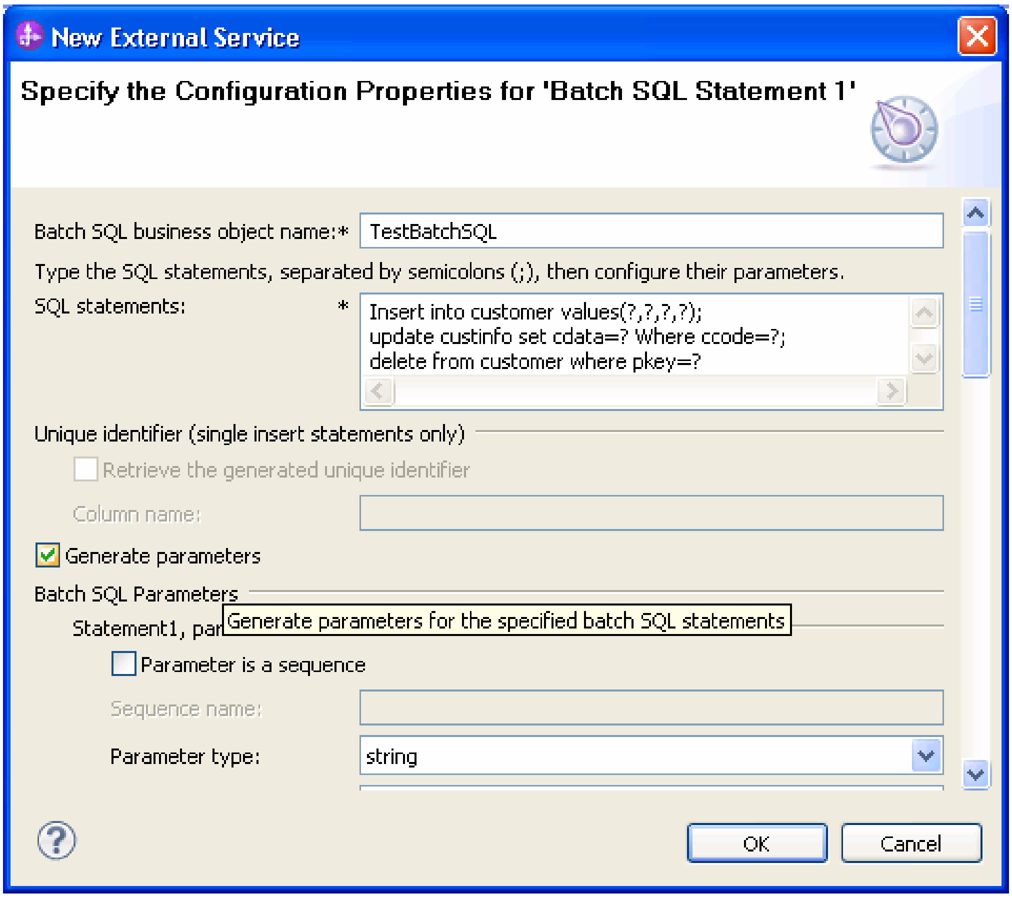
Column name:

□ Generate parameters

Validate batch SQL statements

Result:

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:



New External Service

Specify the Configuration Properties for 'Batch SQL Statement 1'

Batch SQL business object name:* TestBatchSQL

Type the SQL statements, separated by semicolons (;), then configure their parameters.

SQL statements: *

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

Unique identifier (single insert statements only)

□ Retrieve the generated unique identifier

Column name:

☑ Generate parameters

Batch SQL Parameters

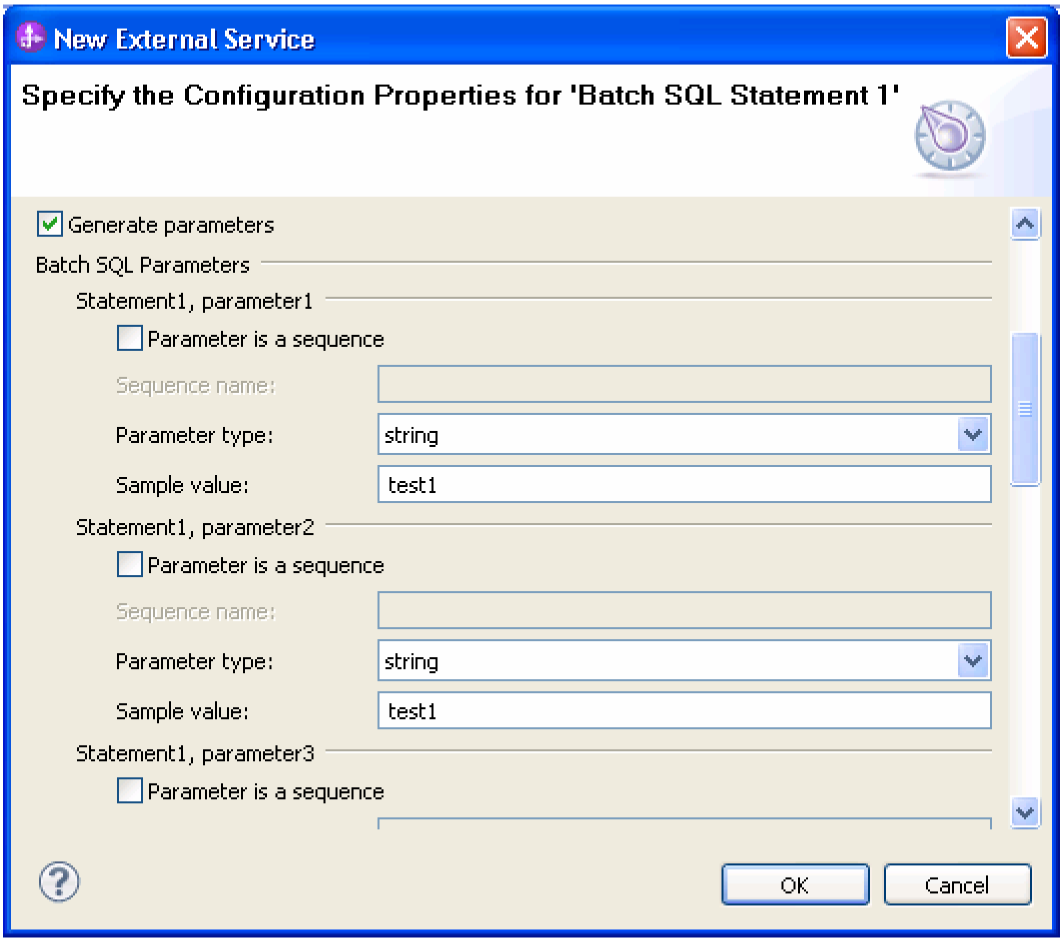
Statement1, par Generate parameters for the specified batch SQL statements

□ Parameter is a sequence

Sequence name:

Parameter type: string

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



New External Service

Specify the Configuration Properties for 'Batch SQL Statement 1'

☑ Generate parameters

Batch SQL Parameters

Statement1, parameter1

□ Parameter is a sequence

Sequence name:

Parameter type: string

Sample value: test1

Statement1, parameter2

□ Parameter is a sequence

Sequence name:

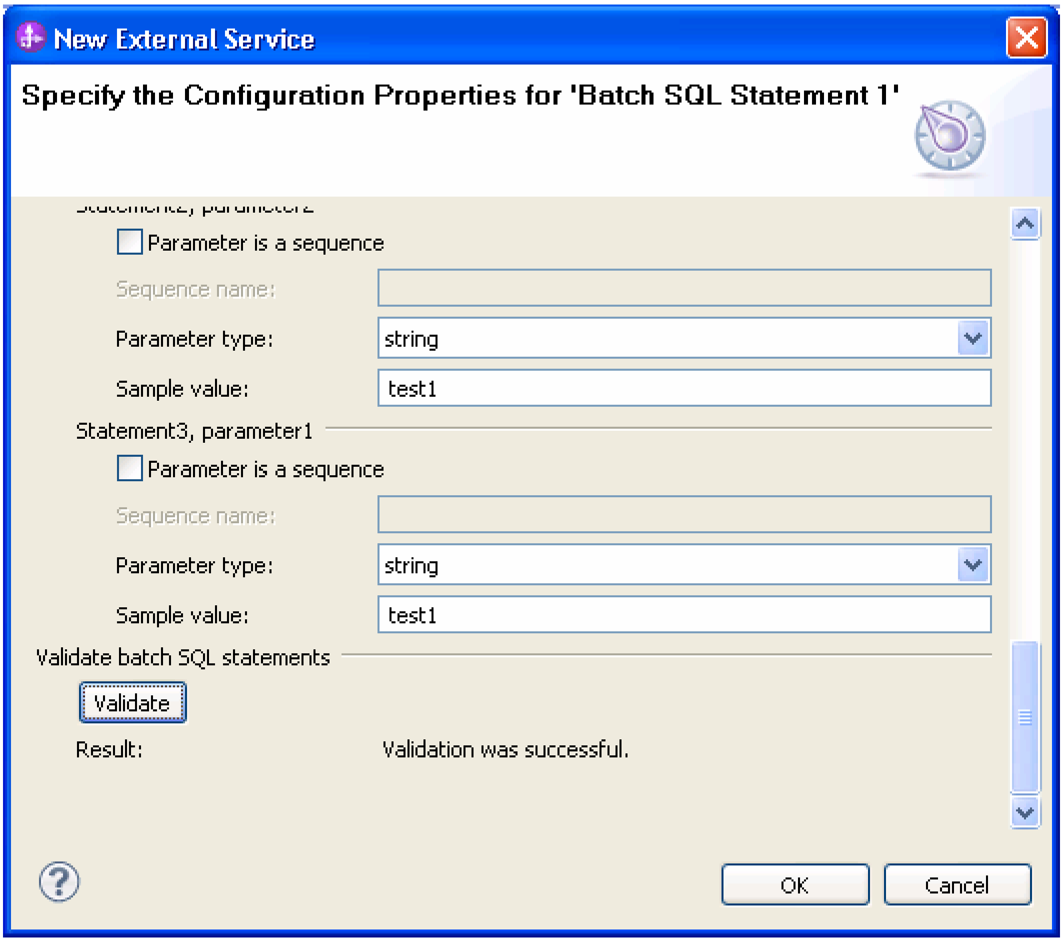
Parameter type: string

Sample value: test1

Statement1, parameter3

□ Parameter is a sequence

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



New External Service

Specify the Configuration Properties for 'Batch SQL Statement 1'

Statement2, parameter2

□ Parameter is a sequence

Sequence name:

Parameter type: string

Sample value: test1

Statement3, parameter1

□ Parameter is a sequence

Sequence name:

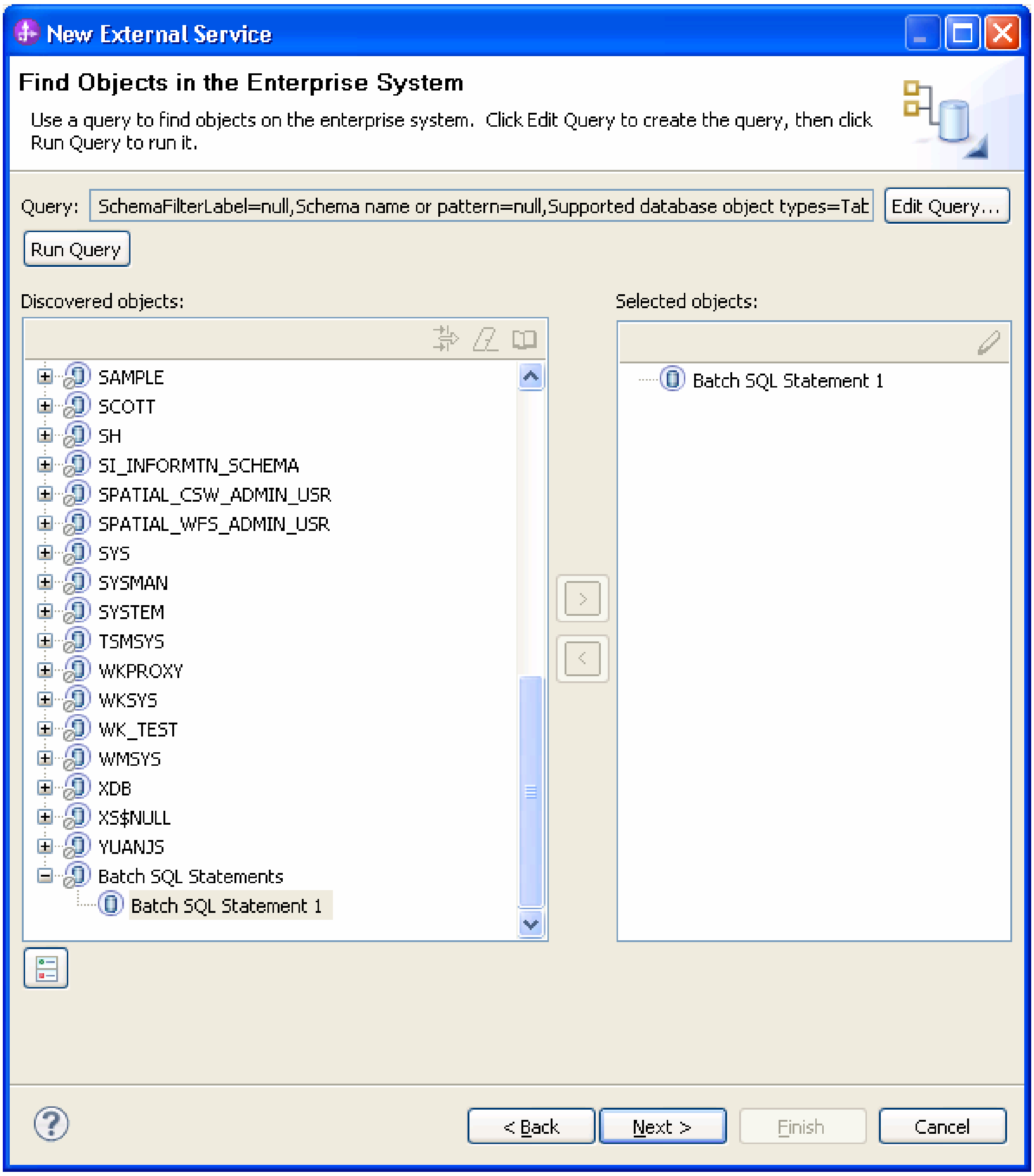
Parameter type: string

Sample value: test1

Validate batch SQL statements

Result: Validation was successful.

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



New External Service

Find Objects in the Enterprise System

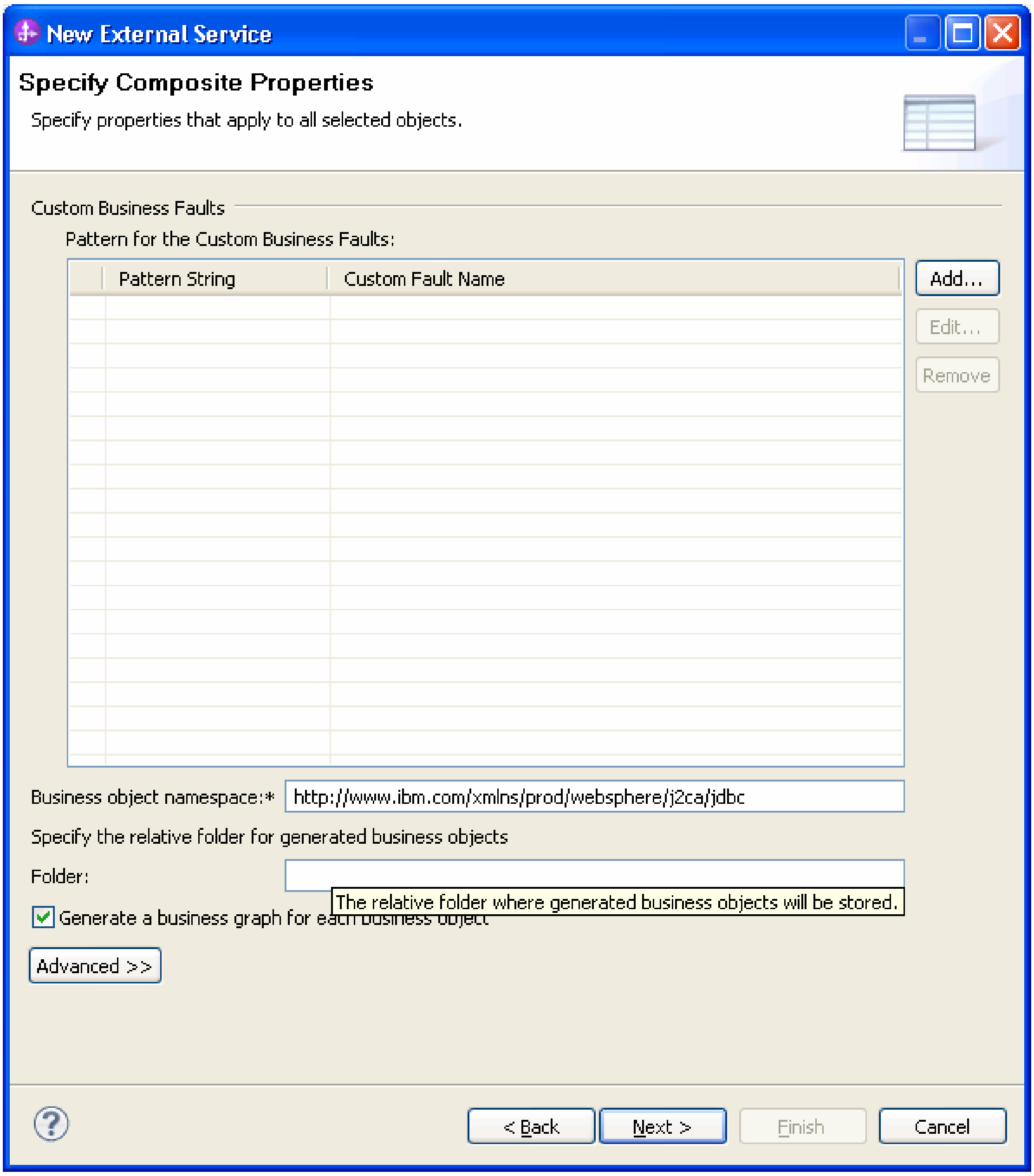
Use a query to find objects on the enterprise system. Click Edit Query to create the query, then click Run Query to run it.

Query: SchemaFilterLabel=null,Schema name or pattern=null,Supported database object types=Tat

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**.



New External Service

Specify Composite Properties

Specify properties that apply to all selected objects.

Custom Business Faults

Pattern for the Custom Business Faults:

	Pattern String	Custom Fault Name

Business object namespace:* http://www.ibm.com/xmlns/prod/websphere/j2ca/jdbc

Specify the relative folder for generated business objects

Folder:

The relative folder where generated business objects will be stored.

☑ Generate a business graph for each business object

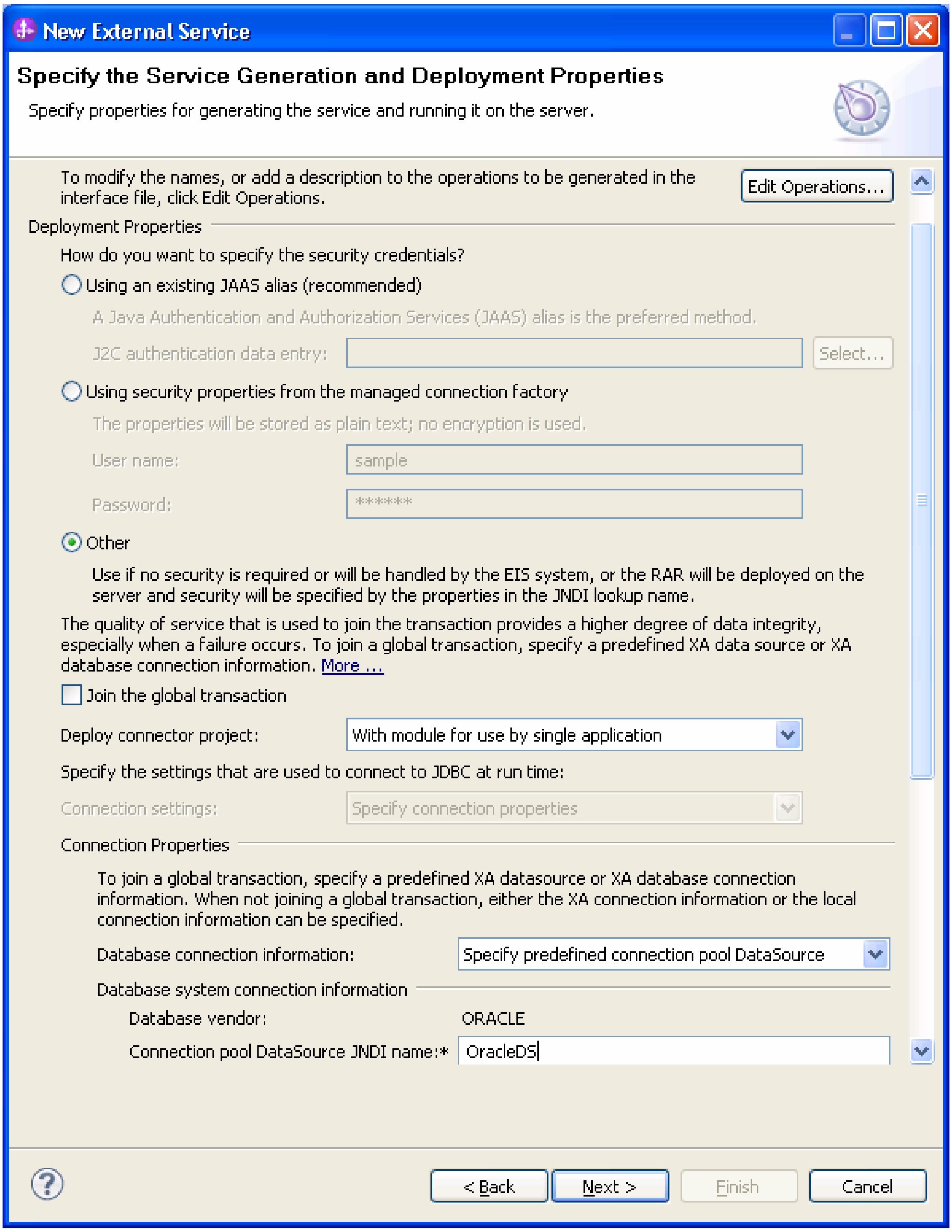
2. In the Specify the Service Generation and Deployment Properties window, perform the following steps:

   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.

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: sample

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 join a global transaction, specify a predefined XA data source or XA database connection information. More ...

□ Join the global transaction

Deploy connector project: With module for use by single application

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

Connection settings: Specify connection properties

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.

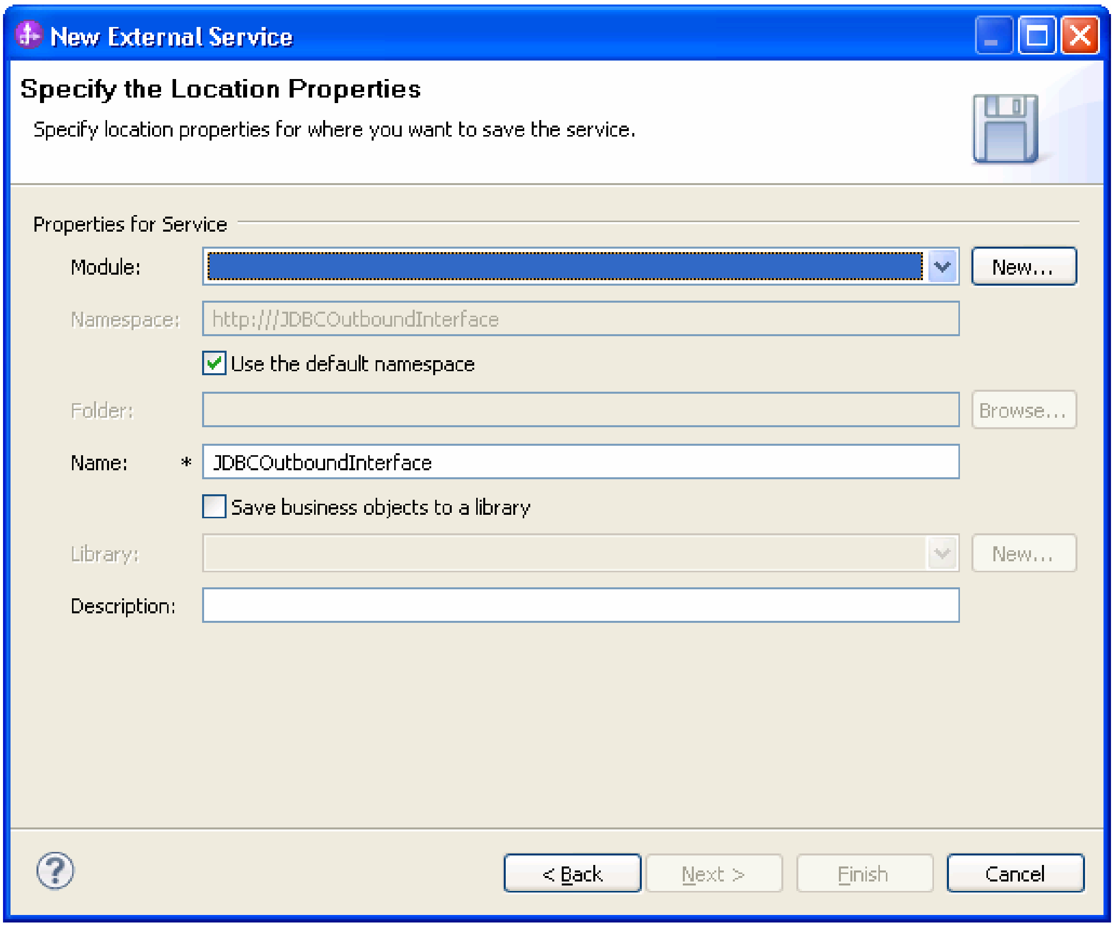
Database connection information: Specify predefined connection pool DataSource

Database system connection information

Database vendor: ORACLE

Connection pool DataSource JNDI name:* OracleDS

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



Specify the Location Properties

Specify location properties for where you want to save the service.

Properties for Service

Module: [New...]

Namespace: http:///JDBCOutboundInterface

☑ Use the default namespace

Folder: [Browse...]

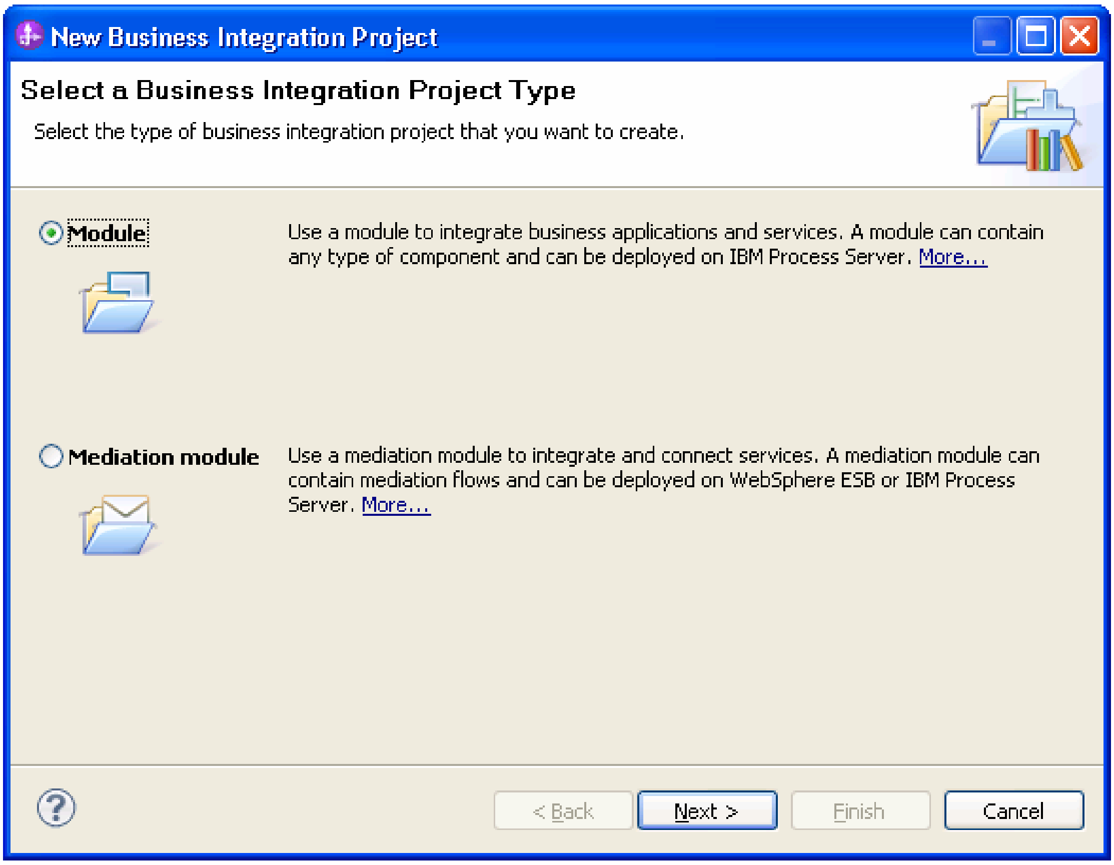
Name: * JDBCOutboundInterface

□ Save business objects to a library

Library: [New...]

Description:

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



Select a Business Integration Project Type

Select the type of business integration project that you want to create.

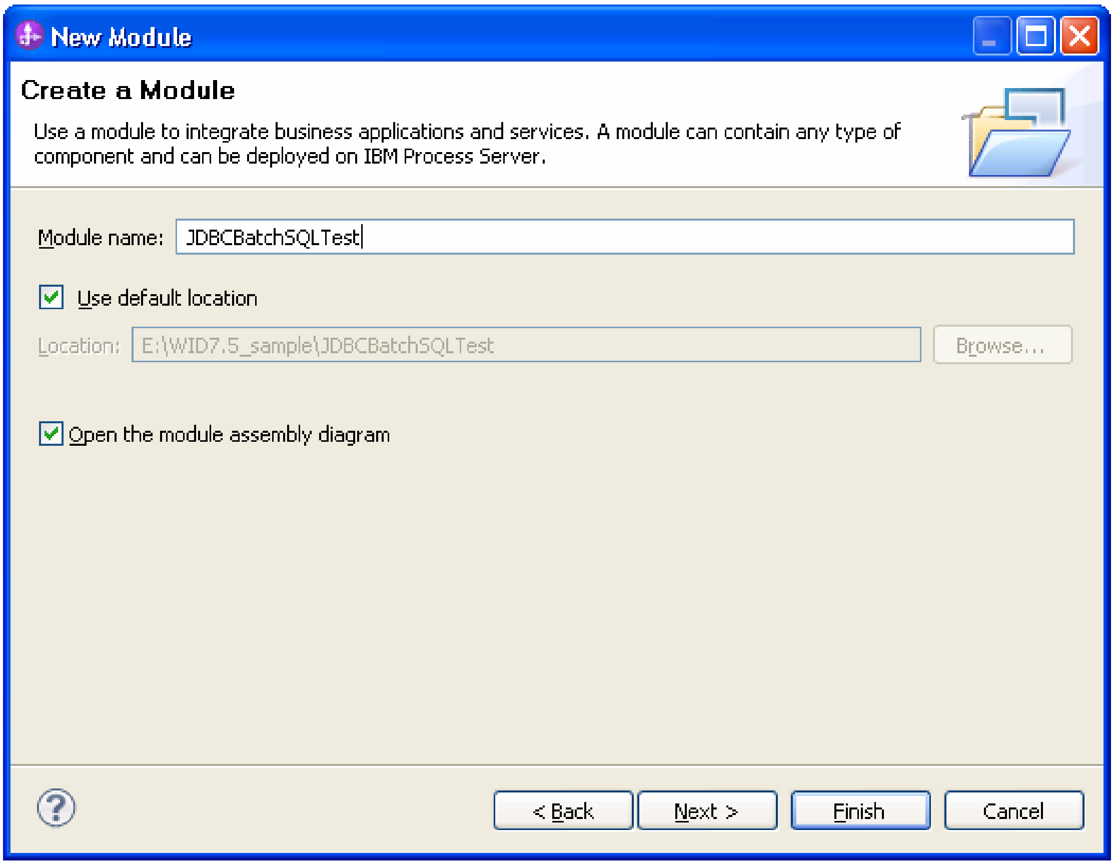
● Module

Use a module to integrate business applications and services. A module can contain any type of component and can be deployed on IBM Process Server. More...

○ Mediation module

Use a mediation module to integrate and connect services. A mediation module can contain mediation flows and can be deployed on WebSphere ESB or IBM Process Server. More...

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



Create a Module

Use a module to integrate business applications and services. A module can contain any type of component and can be deployed on IBM Process Server.

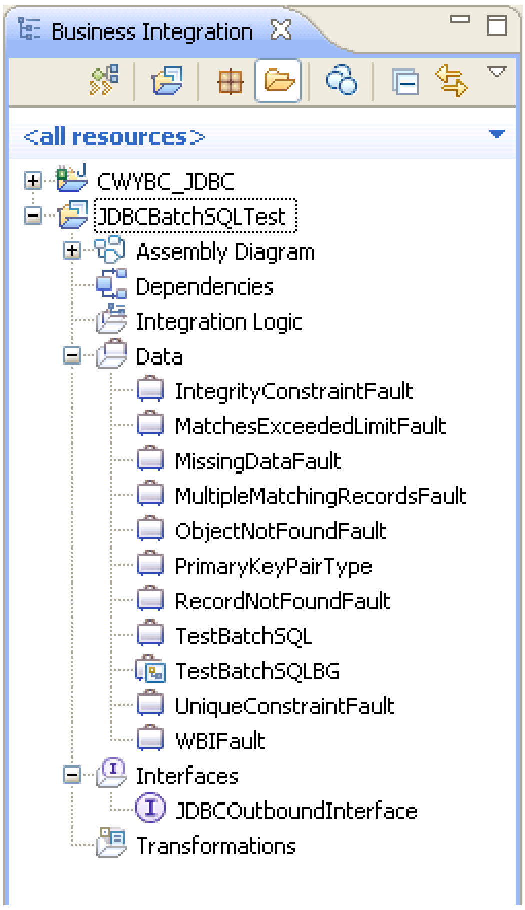
Module name: JDBCBatchSQLTest

☑ Use default location

Location: E:\WID7.5_sample\JDBCBatchSQLTest

☑ Open the module assembly diagram

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



Business Integration

<all resources>

- CWYBC_JDBC
- JDBCBatchSQLTest
  - Assembly Diagram
  - Dependencies
  - Integration Logic
  - Data
    - IntegrityConstraintFault
    - MatchesExceededLimitFault
    - MissingDataFault
    - MultipleMatchingRecordsFault
    - ObjectNotFoundFault
    - PrimaryKeyPairType
    - RecordNotFoundFault
    - TestBatchSQL
    - TestBatchSQLBG
    - UniqueConstraintFault
    - WBIFault
  - Interfaces
    - JDBCOutboundInterface
  - Transformations

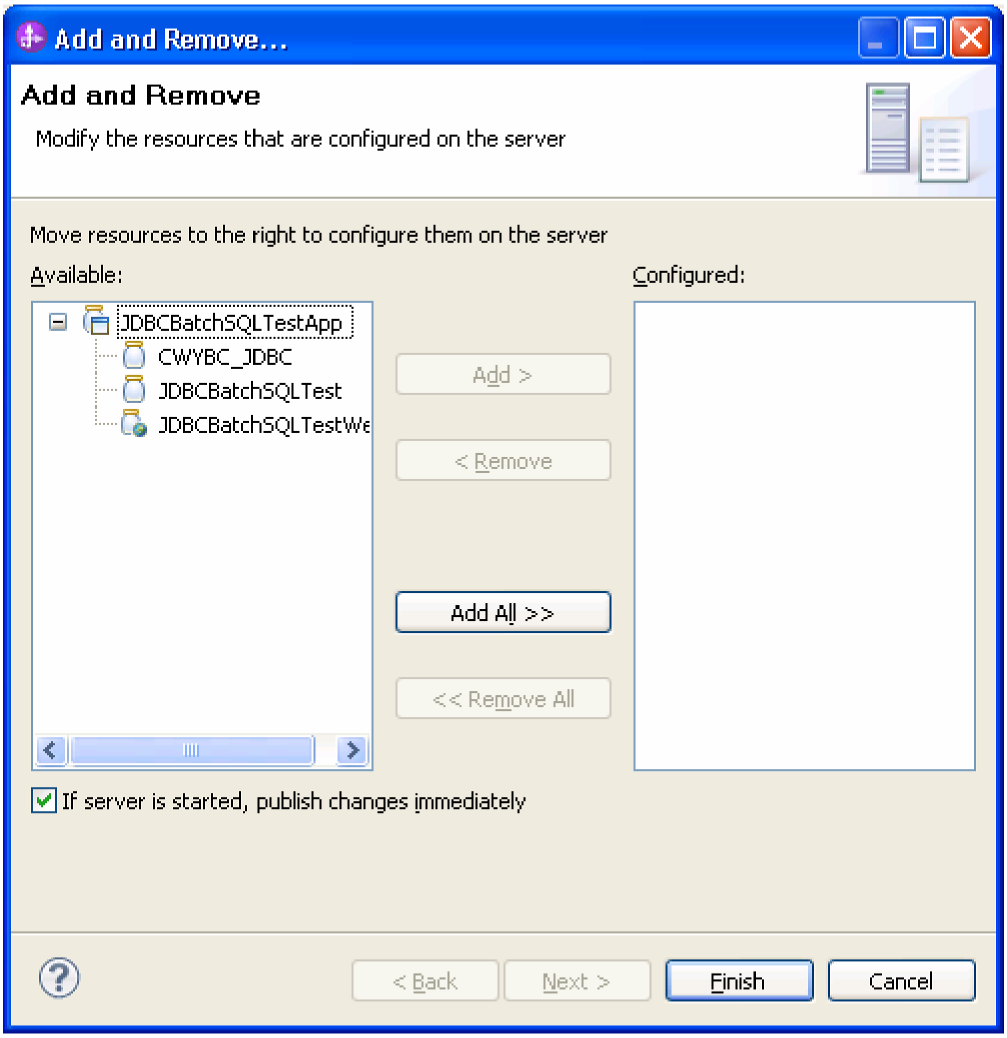
Deploy the module to the test environment

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 IBM Integration Designer integration test client.

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

Steps for adding the SCA module to the server:

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



Add and Remove...

Add and Remove

Modify the resources that are configured on the server

Move resources to the right to configure them on the server

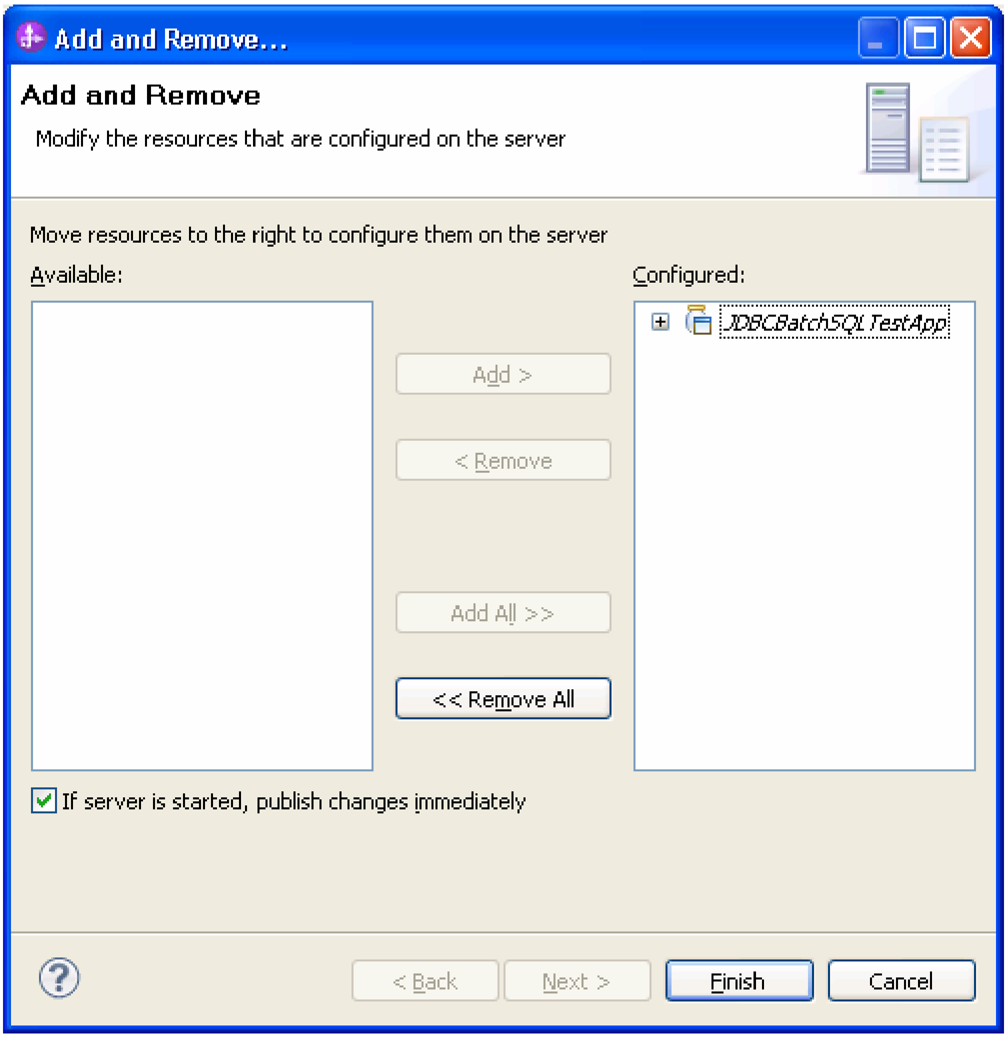
Available:

- JDBCBatchSQLTestApp
  - CWYBC_JDBC
  - JDBCBatchSQLTest
  - JDBCBatchSQLTestWe

Configured:

☑ If server is started, publish changes immediately

4. Add the SCA module to the server.



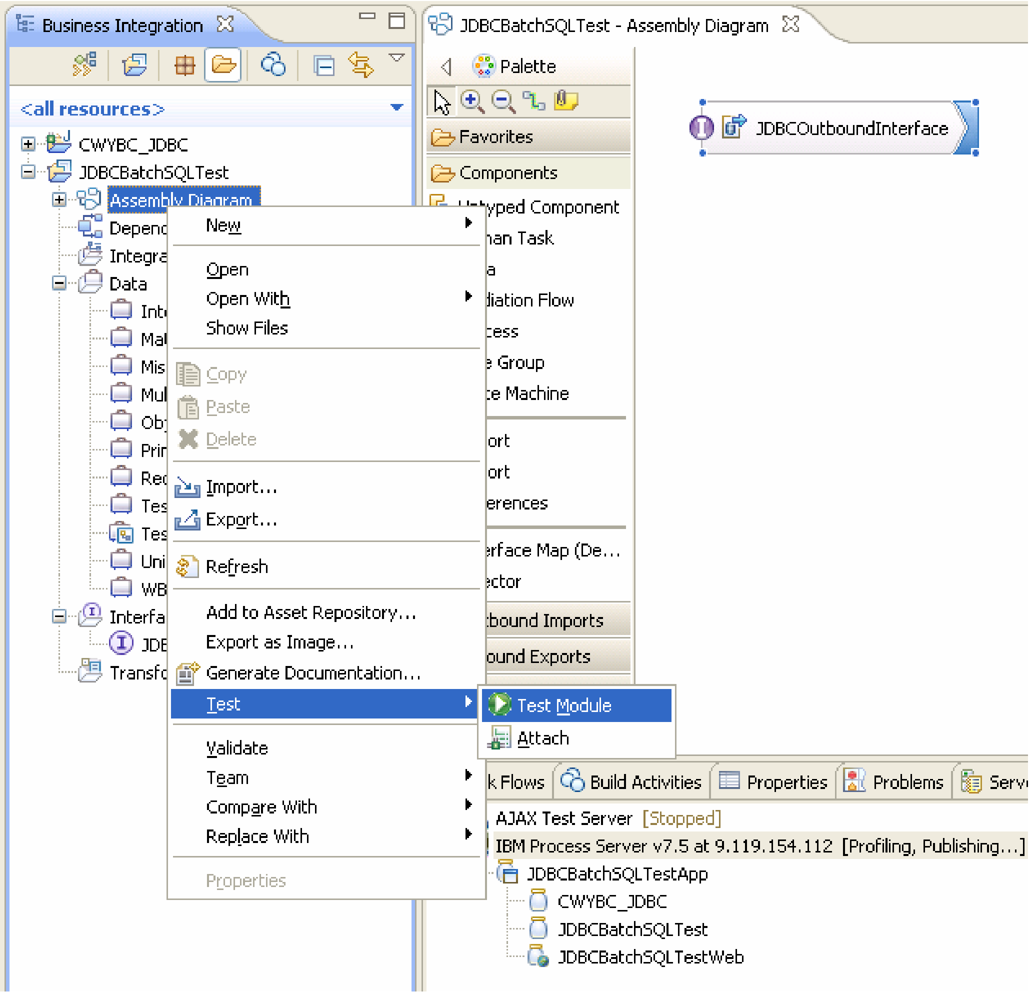
Add and Remove...
Add and Remove
Modify the resources that are configured on the server
Move resources to the right to configure them on the server
Available:
Configured:
JDBCBatchSQLTestApp
Add >
< Remove
Add All >>
<< Remove All
☑ If server is started, publish changes immediately
< Back
Next >
Finish
Cancel

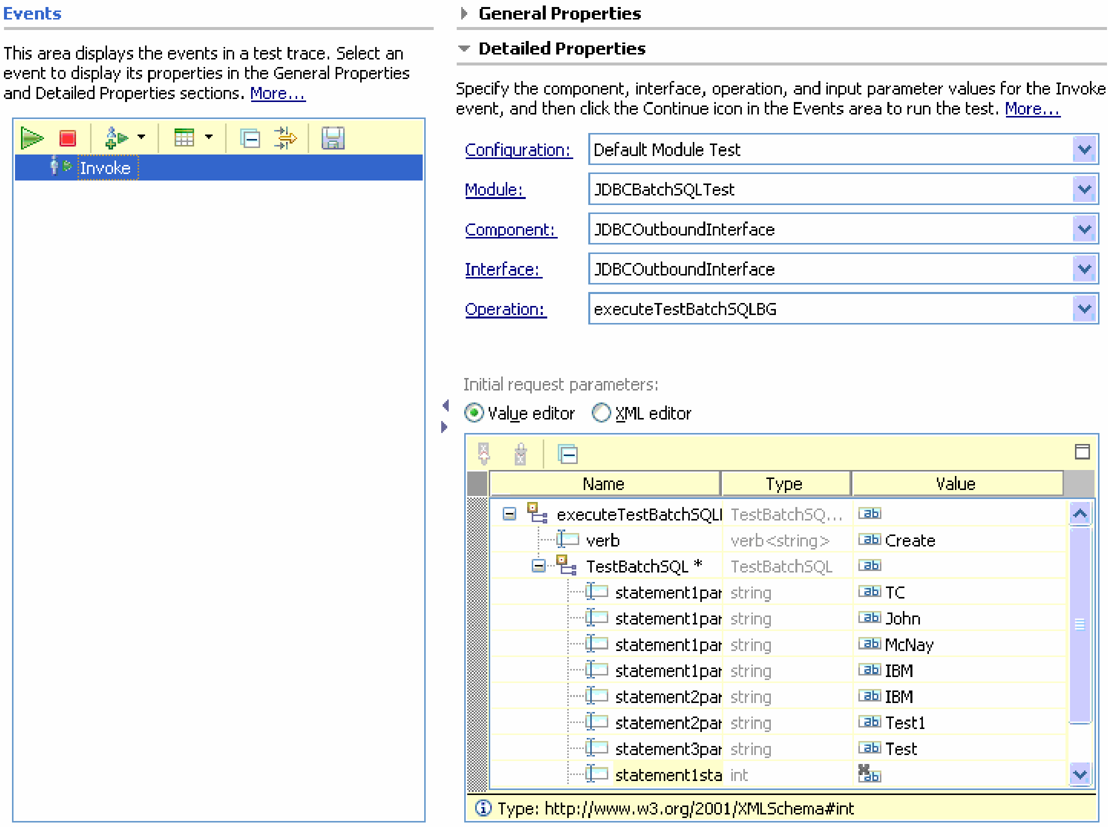
5. Click **Finish**.

Test the assembled adapter application

Test the assembled adapter application using the IBM Integration Designer integration test client.

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





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, and then click the Continue icon in the Events area to run the test. More...

Configuration: Default Module Test

Module: JDBCBatchSQLTest

Component: JDBCOutboundInterface

Interface: JDBCOutboundInterface

Operation: executeTestBatchSQLBG

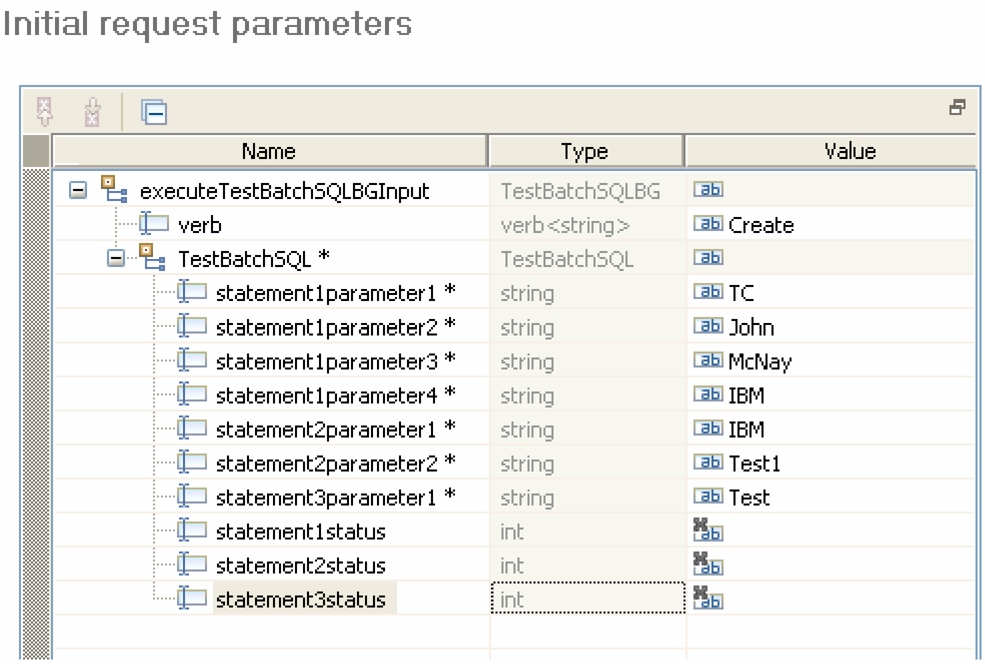
Initial request parameters:

● Value editor ○ XML editor

Name	Type	Value
executeTestBatchSQLI	TestBatchSQ...	
verb	verb<string>	Create
TestBatchSQL *	TestBatchSQL	
statement1par	string	TC
statement1par	string	John
statement1par	string	McNay
statement1par	string	IBM
statement2par	string	IBM
statement2par	string	Test1
statement3par	string	Test
statement1sta	int	

Type: http://www.w3.org/2001/XMLSchema#int

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

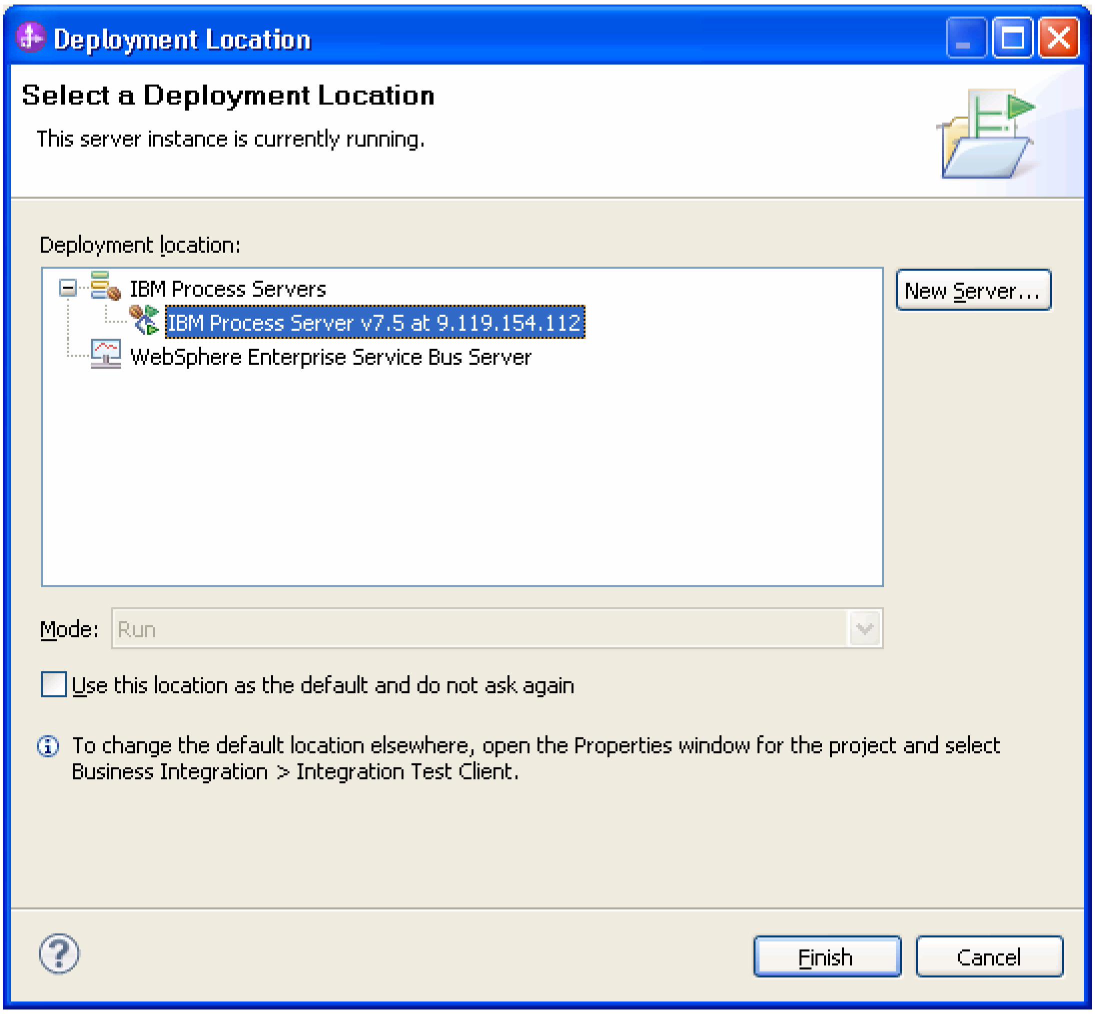


Initial request parameters

Name	Type	Value
executeTestBatchSQLBGInput	TestBatchSQLBG	
verb	verb<string>	Create
TestBatchSQL *	TestBatchSQL	
statement1parameter1 *	string	TC
statement1parameter2 *	string	John
statement1parameter3 *	string	McNay
statement1parameter4 *	string	IBM
statement2parameter1 *	string	IBM
statement2parameter2 *	string	Test1
statement3parameter1 *	string	Test
statement1status	int	
statement2status	int	
statement3status	int	

3. To execute the service, click **Continue** ▶.

4. In the Select a Deployment location window, select the server and click **Finish**.



Deployment Location

Select a Deployment Location

This server instance is currently running.

Deployment location:

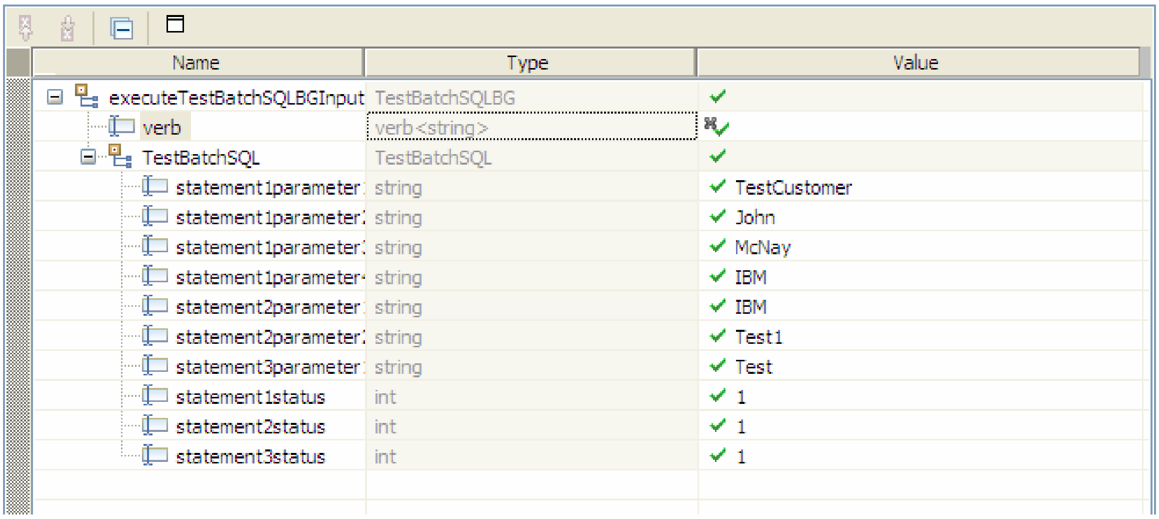
- IBM Process Servers
  - IBM Process Server v7.5 at 9.119.154.112
- WebSphere Enterprise Service Bus Server

Mode: Run

□ Use this location as the default and do not ask again

To change the default location elsewhere, open the Properties window for the project and select Business Integration > Integration Test Client.

5. 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
statement2parameter	string	✓ Test1
statement3parameter	string	✓ Test
statement1status	int	✓ 1
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.

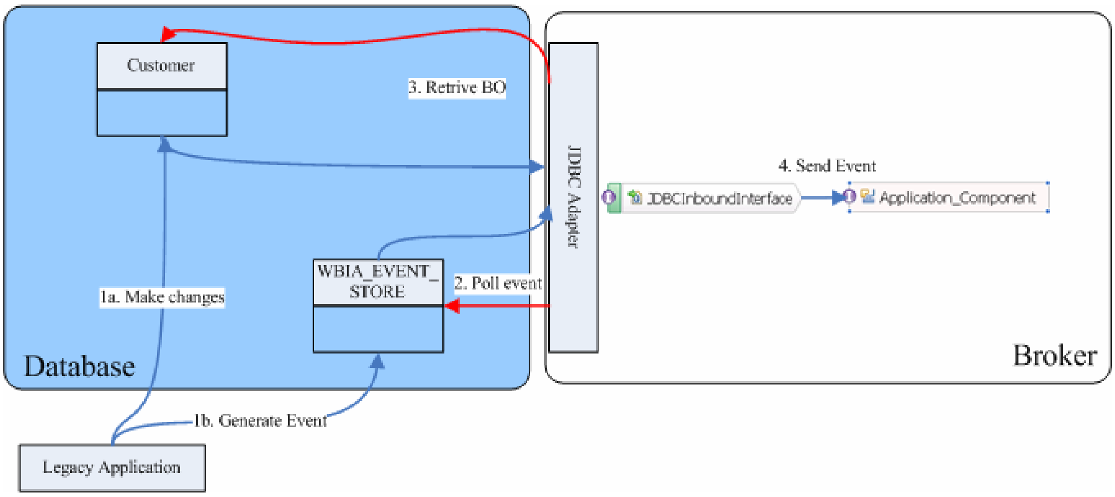
Chapter 6. Tutorial 5: Receiving events from the Enterprise Information System (Oracle)

This tutorial demonstrates how WebSphere Adapter for JDBC 7.5.0.0 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_TIMEOUT TIMESTAMP,
   CONNECTOR_ID            VARCHAR2(40),
   EVENT_COMMENT     VARCHAR2(100)
);
```

b. Script for creating triggers for Inbound

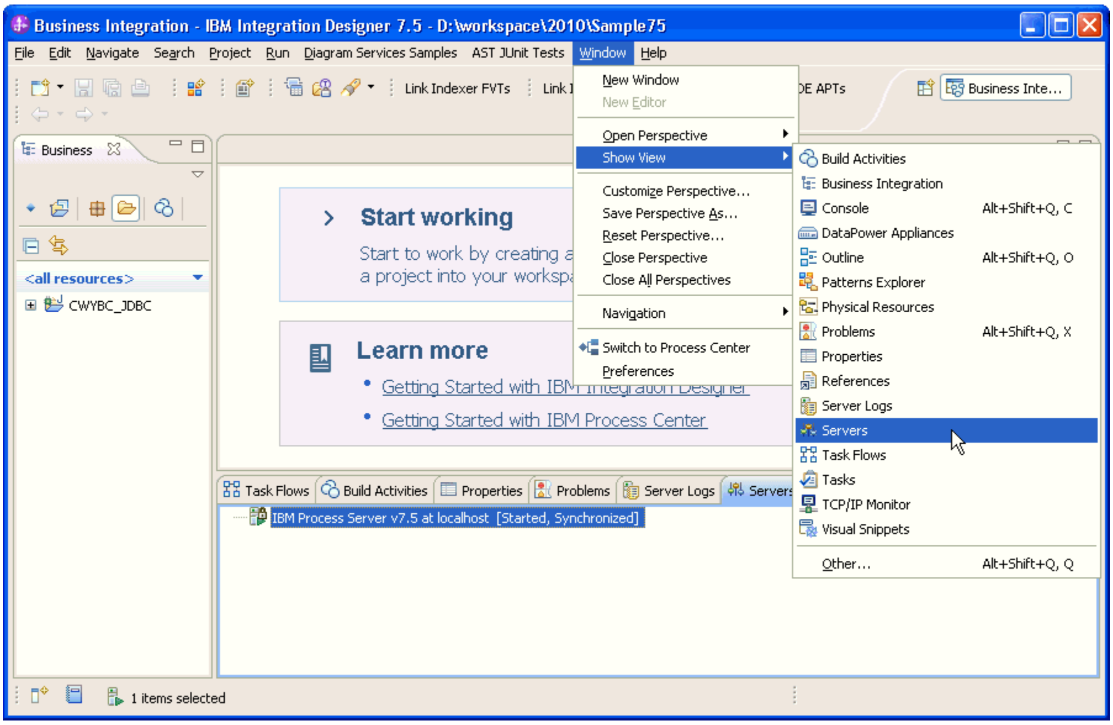
```
CREATE OR REPLACE TRIGGER EVENT_CREATE AFTER INSERT
ON CUSTOMER
REFERENCING OLD AS O NEW AS N
FOR EACH ROW
BEGIN
INSERT INTO wbia_jdbc_eventstore (event_id,
object_key, object_name,object_function,
event_priority, event_status)
VALUES (event_seq.nextval,:N.pkey,
'SampleCustomerBG', 'Create', 1, 0);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

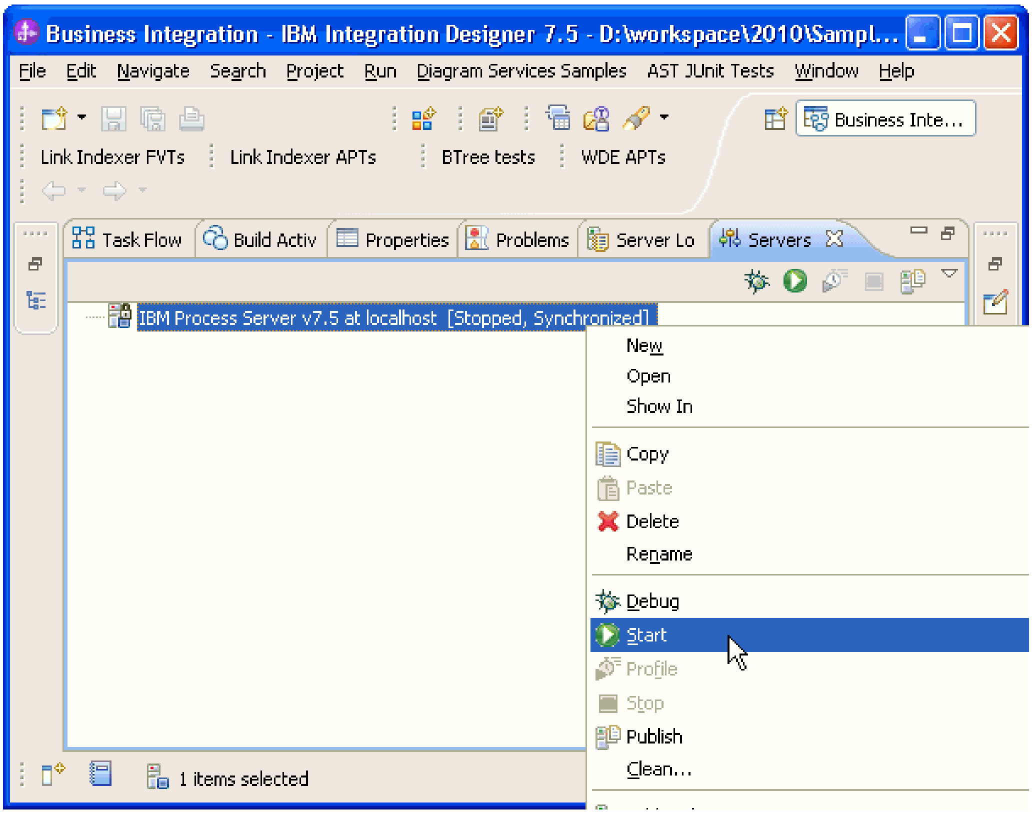
The authentication alias needs to be set because the data source created in the next section uses the username and password set in the authentication alias to connect to the database.

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

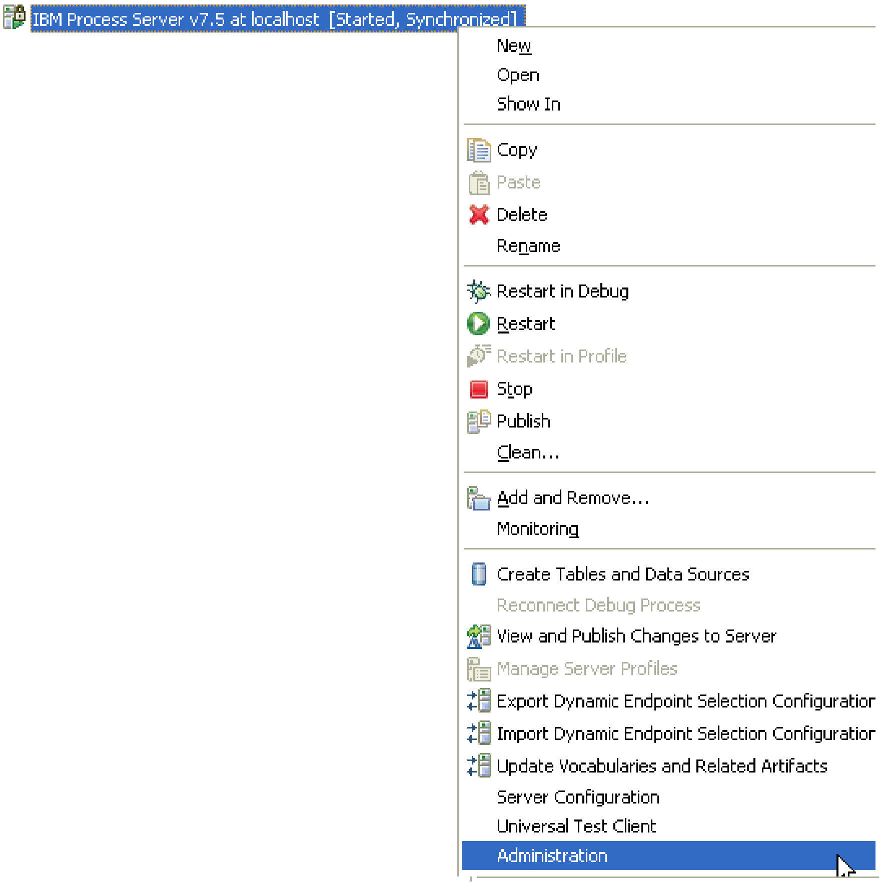
1. In IBM Integration Designer, switch to the **Servers** view by selecting **Windows > Show View > Servers**.



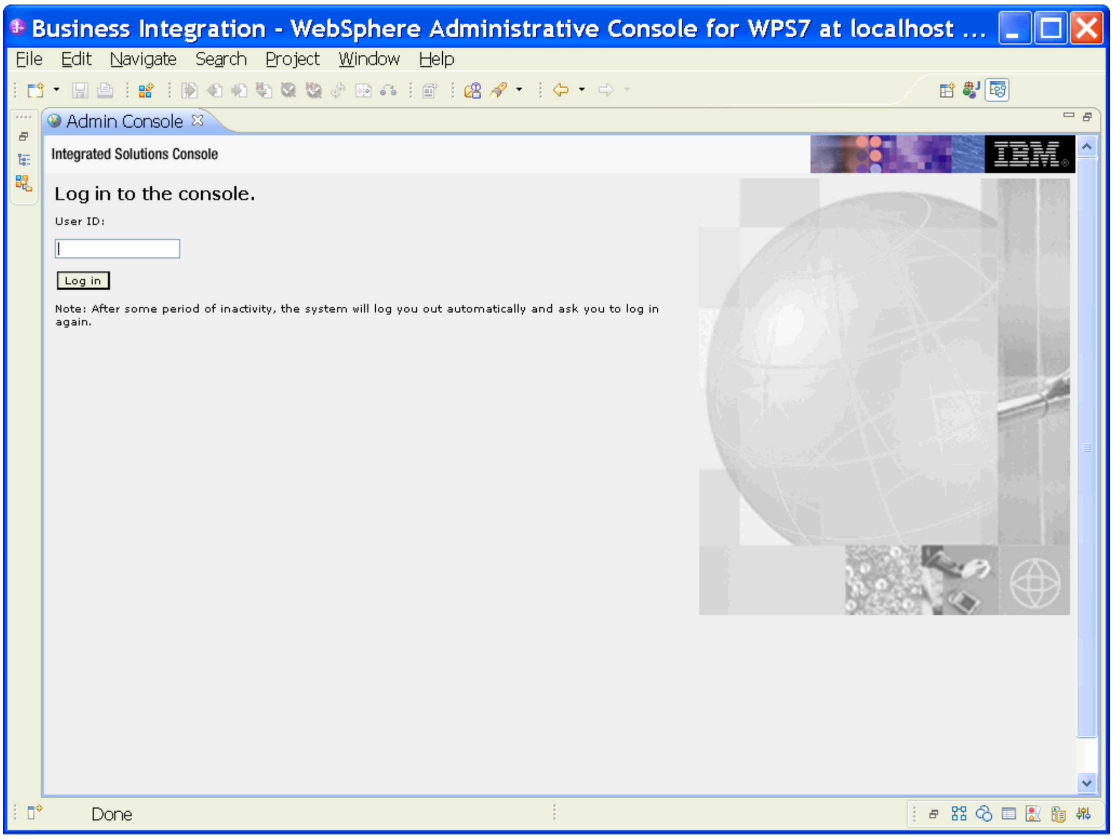
2. In the **Servers** view, right-click the server that you want to start and select **Start**.


Business Integration - IBM Integration Designer 7.5 - D:\workspace\2010\Sampl...
File Edit Navigate Search Project Run Diagram Services Samples AST JUnit Tests Window Help
Link Indexer FVTs Link Indexer APTs BTree tests WDE APTs
Business Inte...
Task Flow Build Activ Properties Problems Server Lo Servers
IBM Process Server v7.5 at localhost [Stopped, Synchronized]
New
Open
Show In
Copy
Paste
Delete
Rename
Debug
Start
Profile
Stop
Publish
Clean...
1 items selected

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


IBM Process Server v7.5 at localhost [Started, Synchronized]
New
Open
Show In
Copy
Paste
Delete
Rename
Restart in Debug
Restart
Restart in Profile
Stop
Publish
Clean...
Add and Remove...
Monitoring
Create Tables and Data Sources
Reconnect Debug Process
View and Publish Changes to Server
Manage Server Profiles
Export Dynamic Endpoint Selection Configuratior
Import Dynamic Endpoint Selection Configuratior
Update Vocabularies and Related Artifacts
Server Configuration
Universal Test Client
Administration

4. Log on to the administrative console.


Business Integration - WebSphere Administrative Console for WPS7 at localhost ...

File Edit Navigate Search Project Window Help
Admin Console
Integrated Solutions Console
IBM

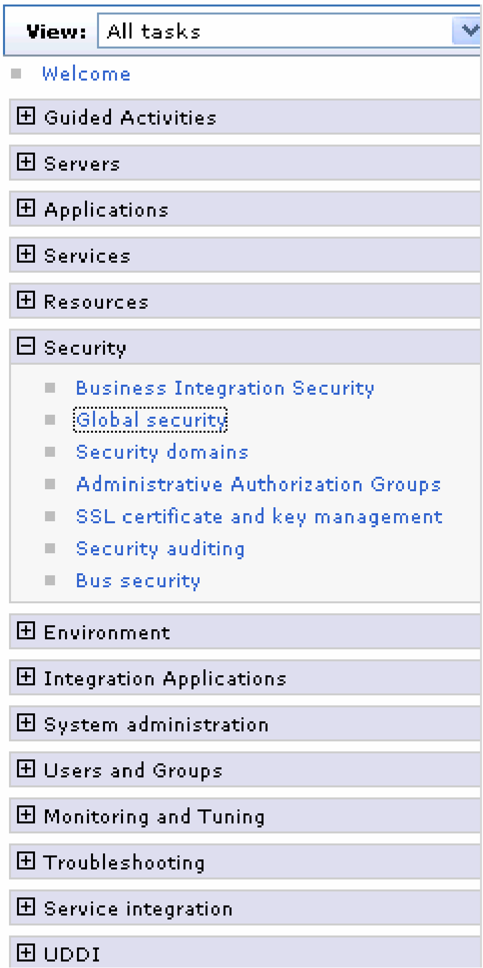
Log in to the console.

User ID:

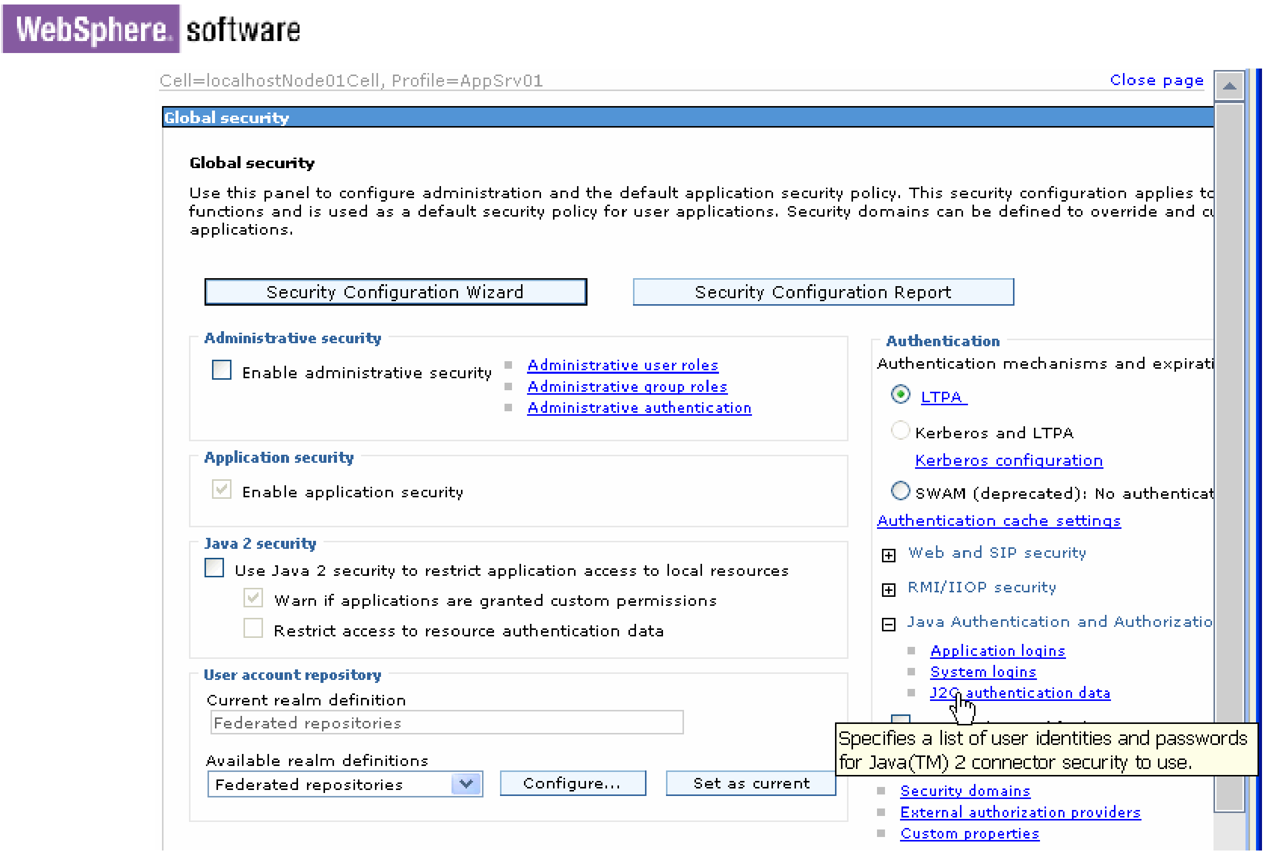
Log in

Note: After some period of inactivity, the system will log you out automatically and ask you to log in again.

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


View: All tasks

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



Cell=localhostNode01Cell, Profile=AppSrv01

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 cu 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 expirati

● LTPA

○ Kerberos and LTPA

Kerberos configuration

○ SWAM (deprecated): No authenticat

Authentication cache settings

⊞ Web and SIP security

⊞ RMI/IIOP security

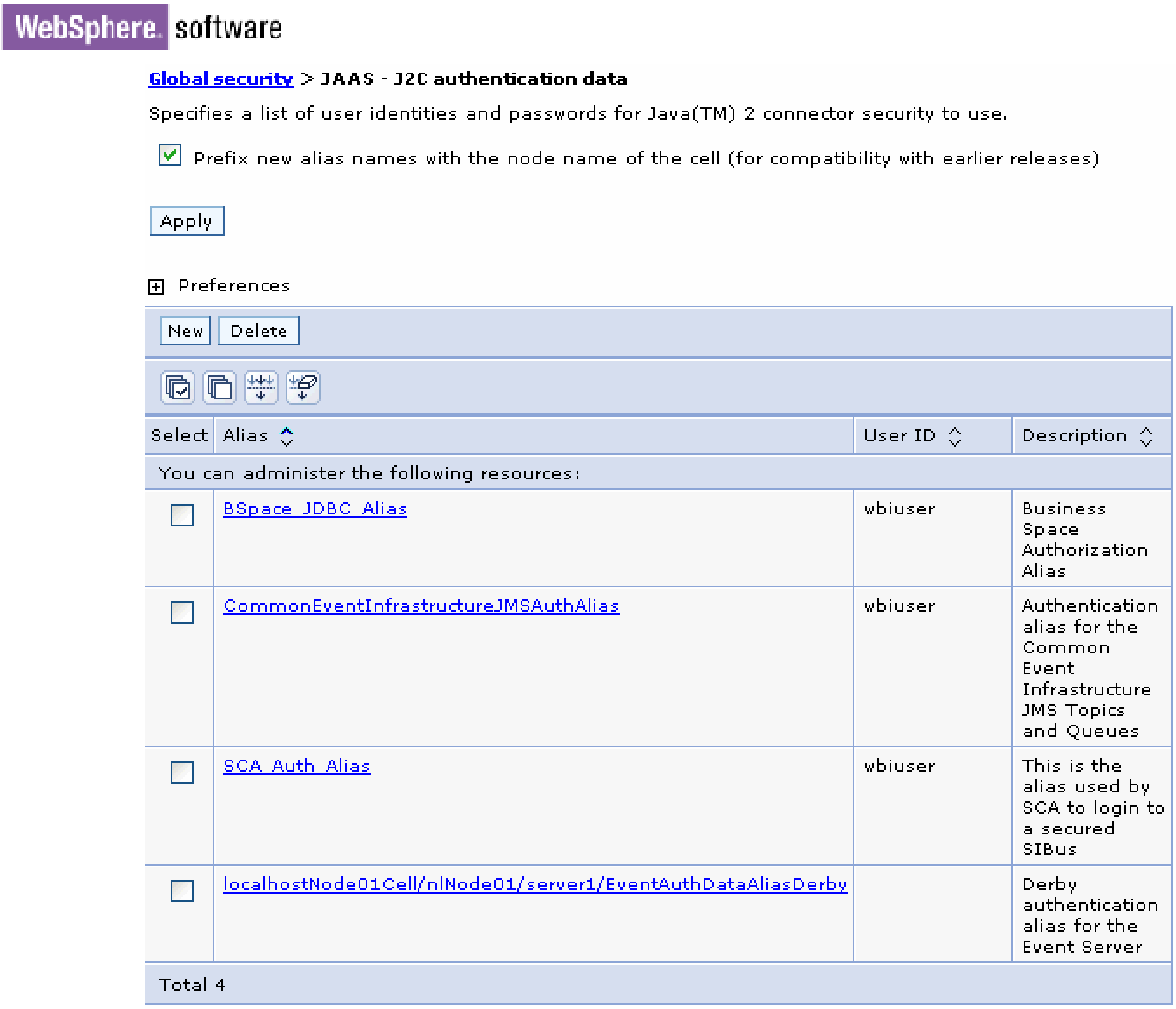
⊟ Java Authentication and Authorizatio

- 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.



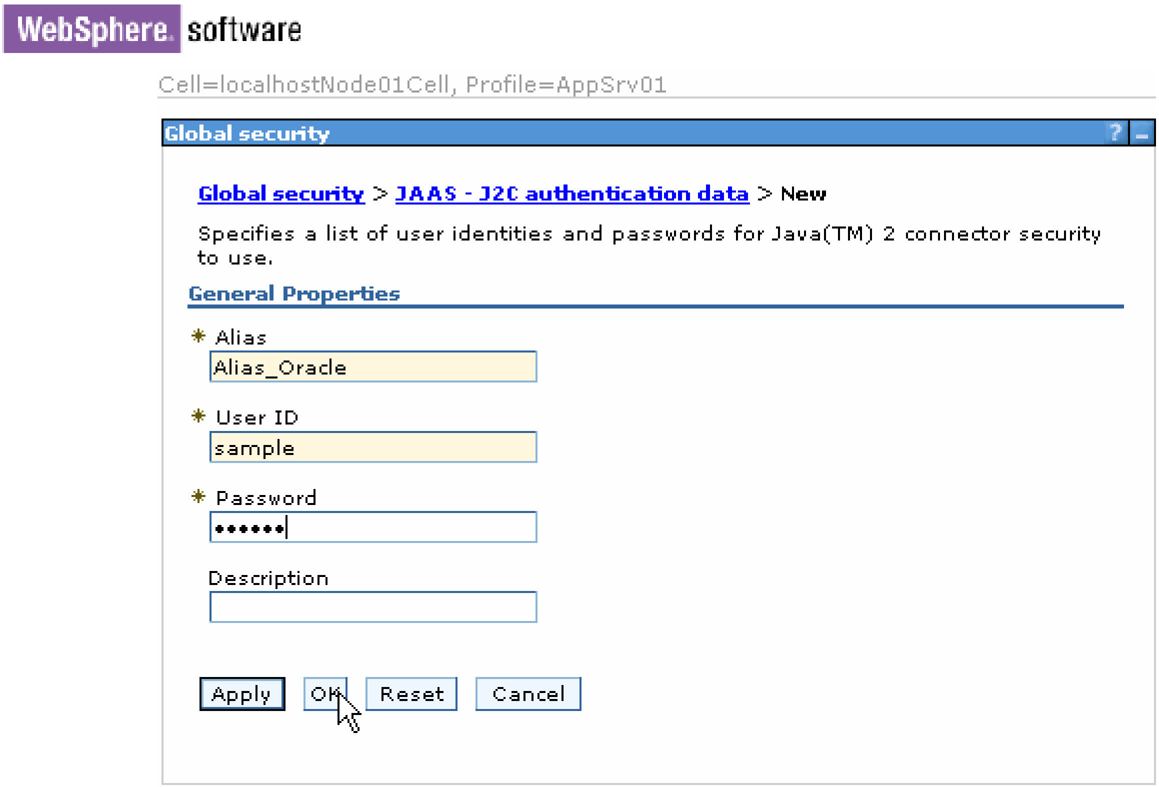
Global security > JAAS - J2C authentication data

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)

Select	Alias	User ID	Description
You can administer the following resources:			
□	BSpace_JDBC_Alias	wbiuser	Business Space Authorization Alias
□	CommonEventInfrastructureJMSAuthAlias	wbiuser	Authentication alias for the Common Event Infrastructure JMS Topics and Queues
□	SCA_Auth_Alias	wbiuser	This is the alias used by SCA to login to a secured SIBus
□	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

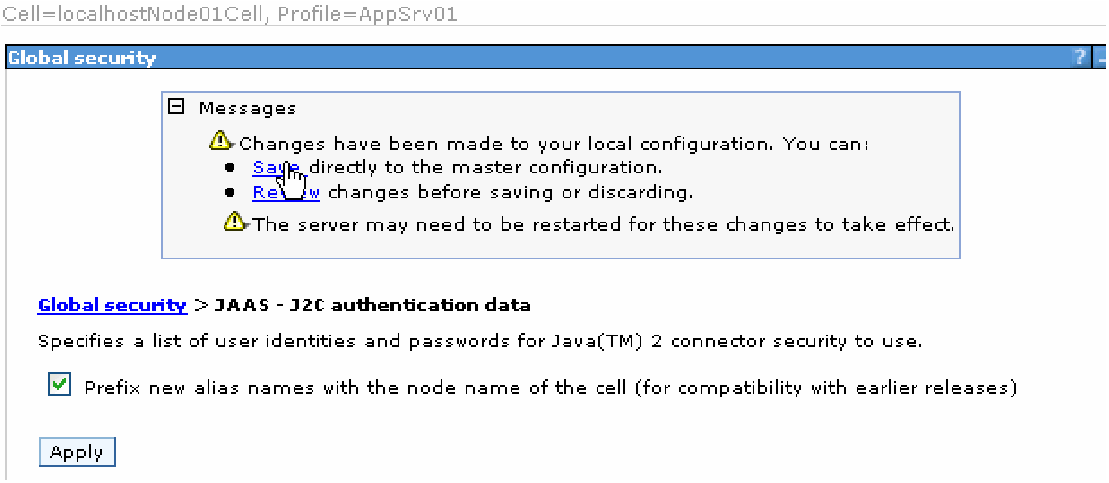
* Alias
Alias_Oracle

* User ID
sample

* Password
••••••

Description

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.

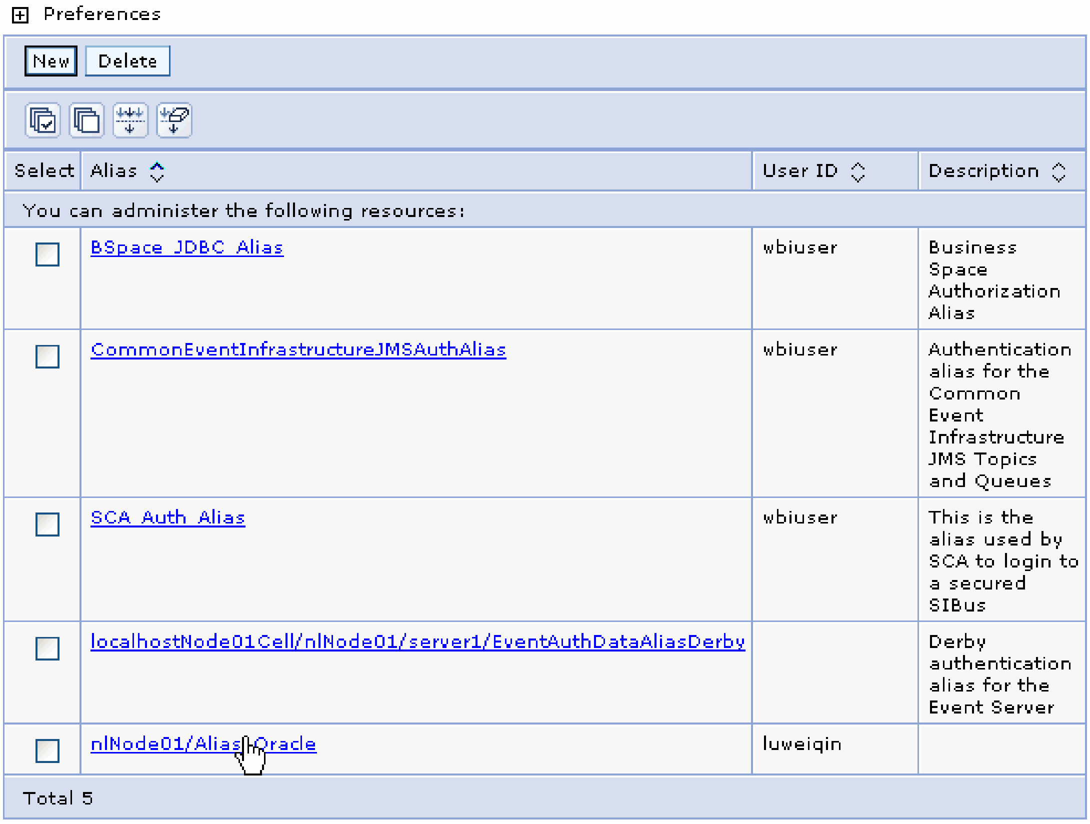
⚠ The server may need to be restarted for these changes to take effect.

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)

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



Select	Alias	User ID	Description
You can administer the following resources:			
□	BSpace_JDBC_Alias	wbiuser	Business Space Authorization Alias
□	CommonEventInfrastructureJMSAuthAlias	wbiuser	Authentication alias for the Common Event Infrastructure JMS Topics and Queues
□	SCA_Auth_Alias	wbiuser	This is the alias used by SCA to login to a secured SIBus
□	localhostNode01Cell/nlNode01/server1/EventAuthDataAliasDerby		Derby authentication alias for the Event Server
□	nlNode01/AliasOracle	luweiqin	
Total 5			

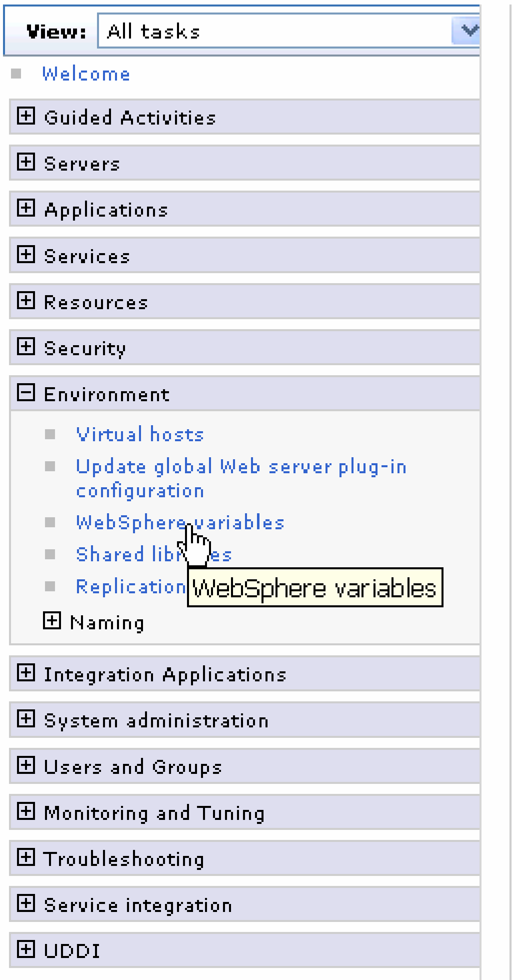
Create a data source

Create a data source in IBM 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.

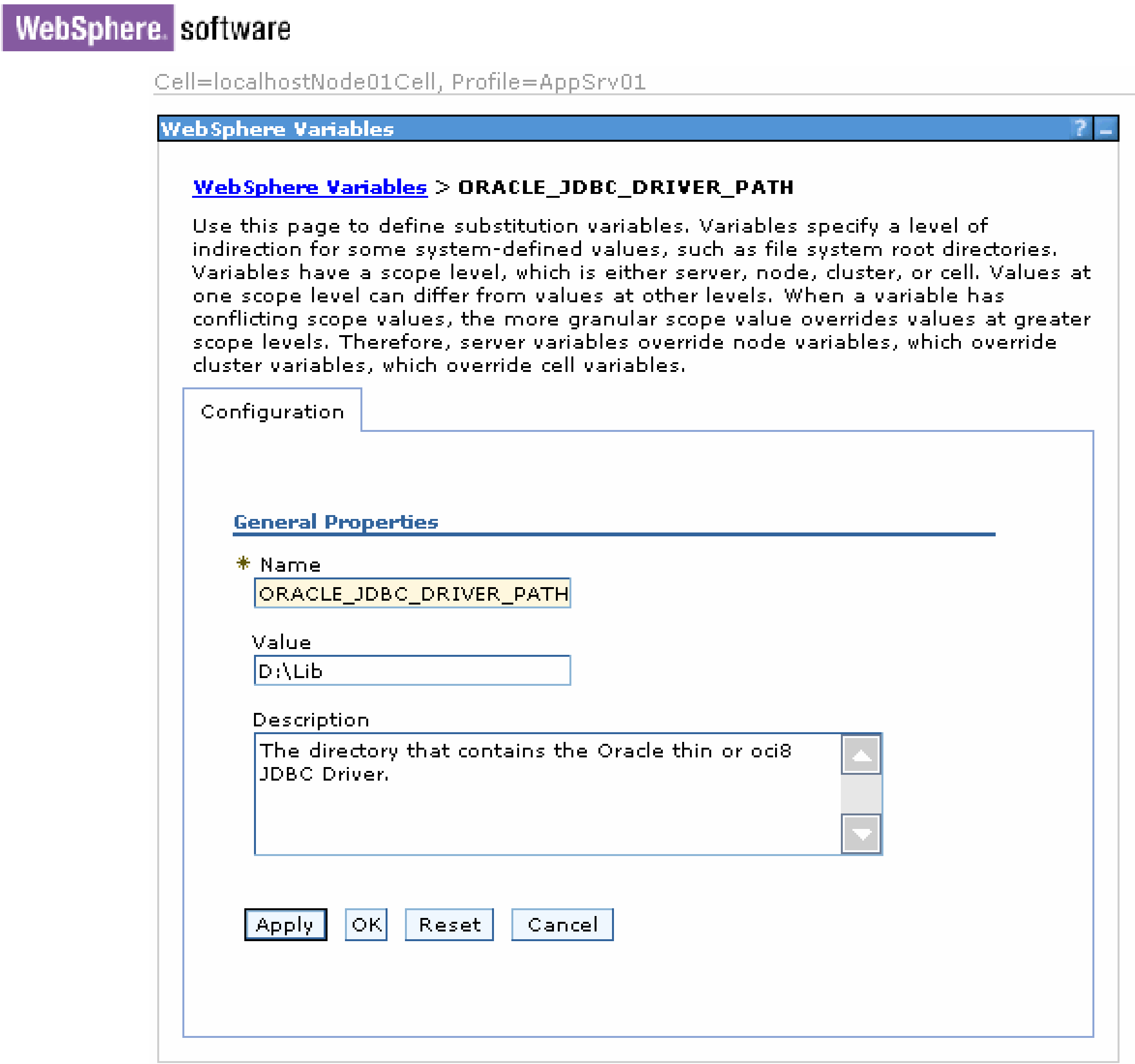
Note: This tutorial uses Oracle as the database and the Oracle thin driver, ojdbc6.jar.

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

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



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

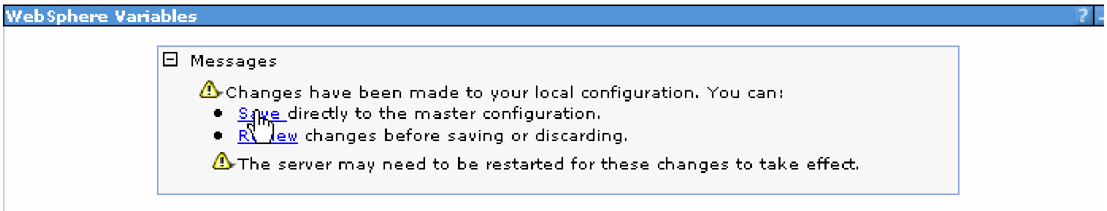
General Properties

* Name
ORACLE_JDBC_DRIVER_PATH

Value
D:\Lib

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

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



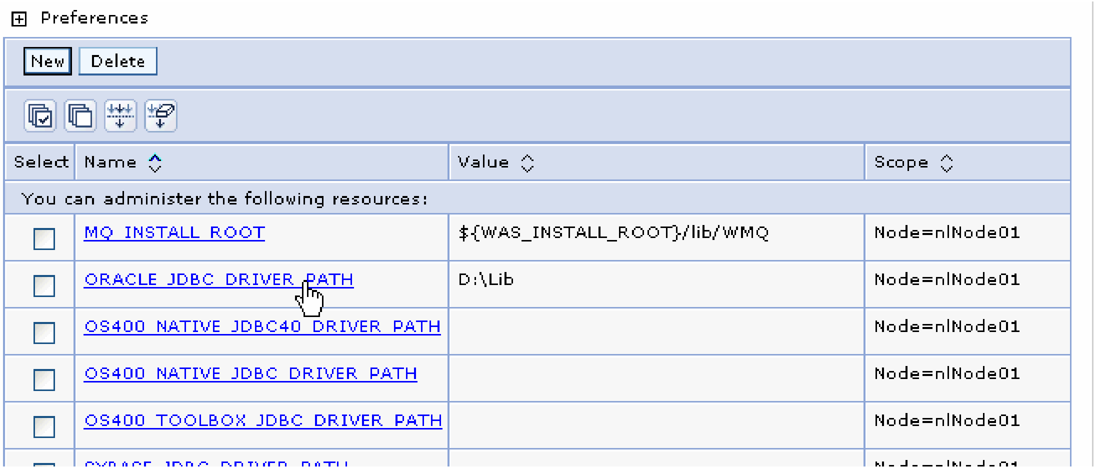
WebSphere Variables

Messages

⚠ Changes have been made to your local configuration. You can:
- Save directly to the master configuration.
- Review changes before saving or discarding.

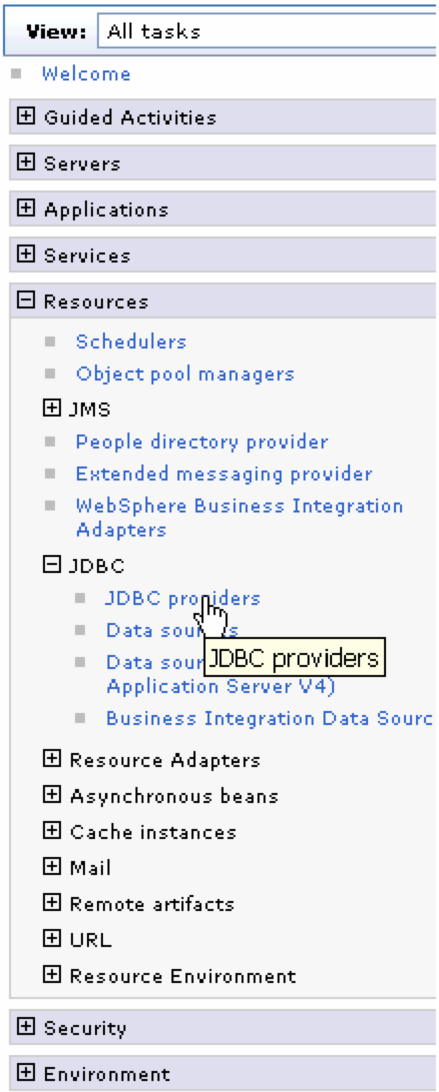
⚠ The server may need to be restarted for these changes to take effect.

The variable has been added and appears in the list.

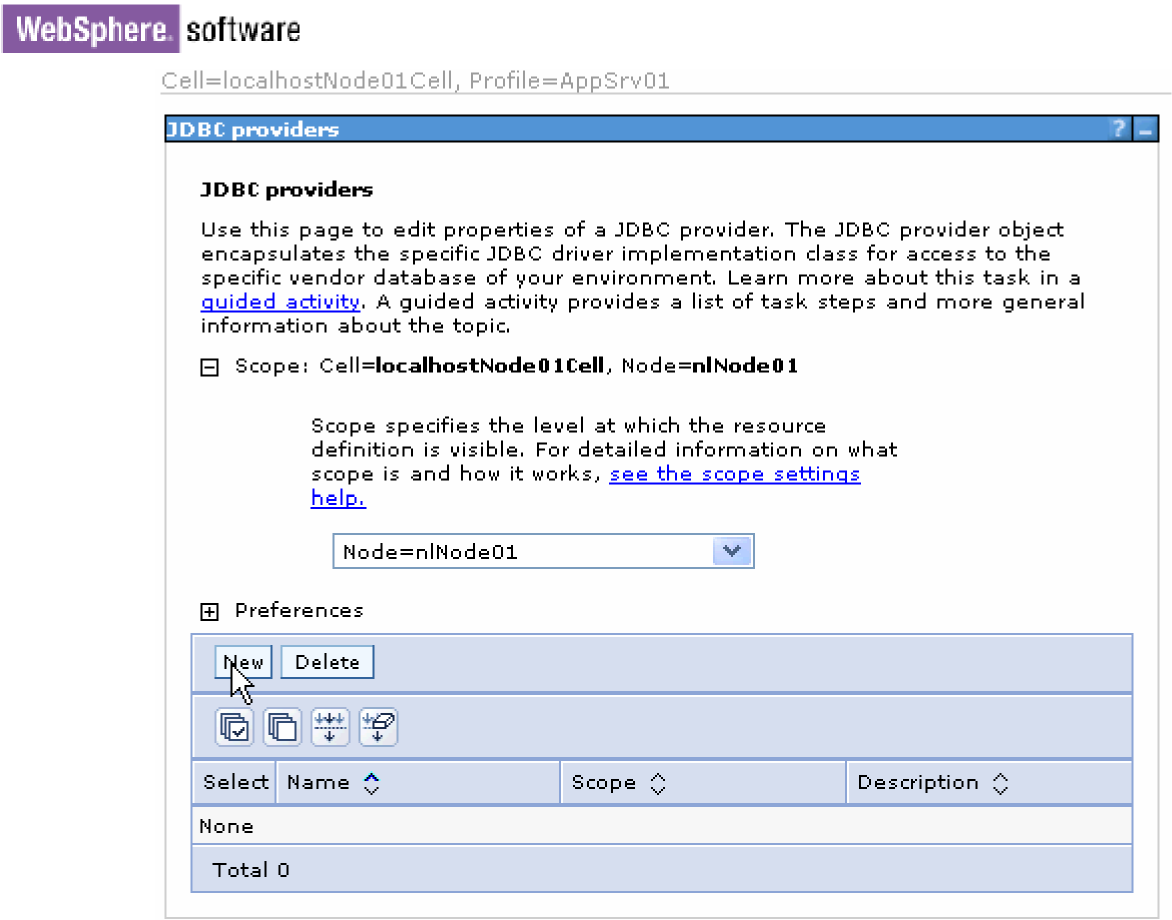


Select	Name	Value	Scope
You can administer the following resources:			
□	MQ_INSTALL_ROOT	${WAS_INSTALL_ROOT}/lib/WMQ	Node=nlNode01
□	ORACLE_JDBC_DRIVER_PATH	D:\Lib	Node=nlNode01
□	OS400_NATIVE_JDBC40_DRIVER_PATH		Node=nlNode01
□	OS400_NATIVE_JDBC_DRIVER_PATH		Node=nlNode01
□	OS400_TOOLBOX_JDBC_DRIVER_PATH		Node=nlNode01

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



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



WebSphere. software

Cell=localhostNode01Cell, Profile=AppSrv01

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=**nlNode01**

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.

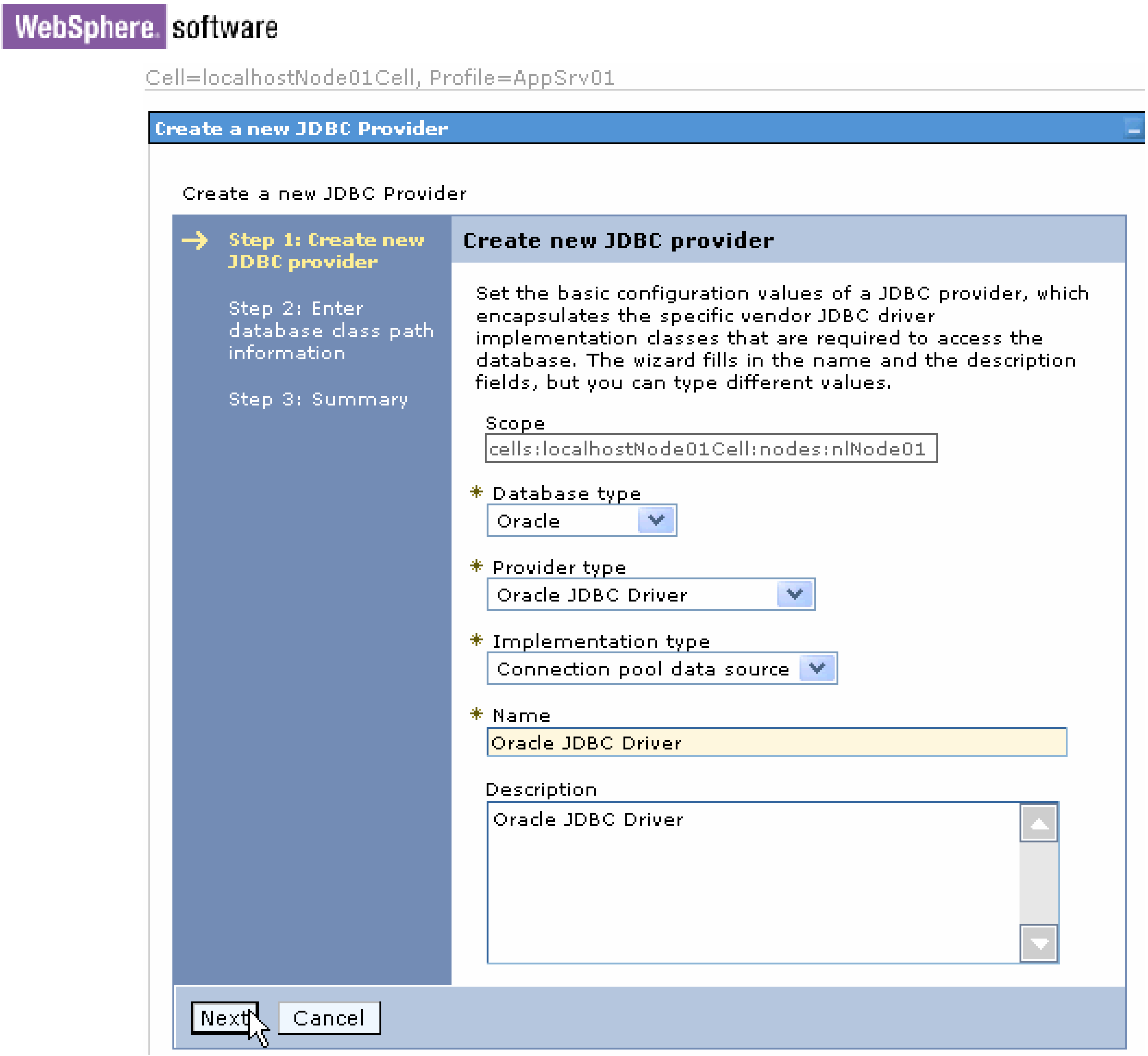
Node=nlNode01

⊞ 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**.



Cell=localhostNode01Cell, Profile=AppSrv01

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:nlNode01

* Database type
Oracle

* Provider type
Oracle JDBC Driver

* Implementation type
Connection pool data source

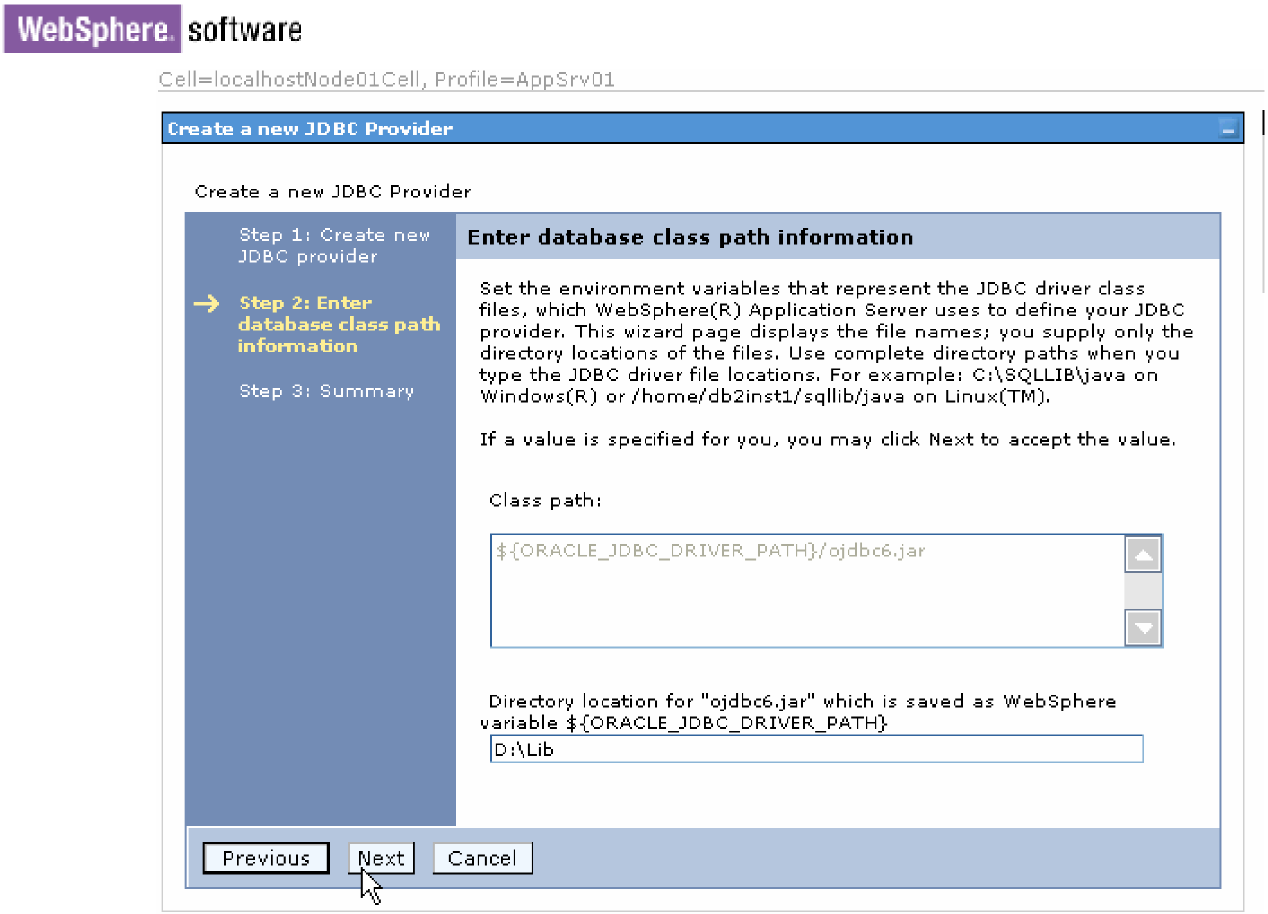
* Name
Oracle JDBC Driver

Description
Oracle JDBC Driver

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**.



Cell=localhostNode01Cell, Profile=AppSrv01

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: 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.

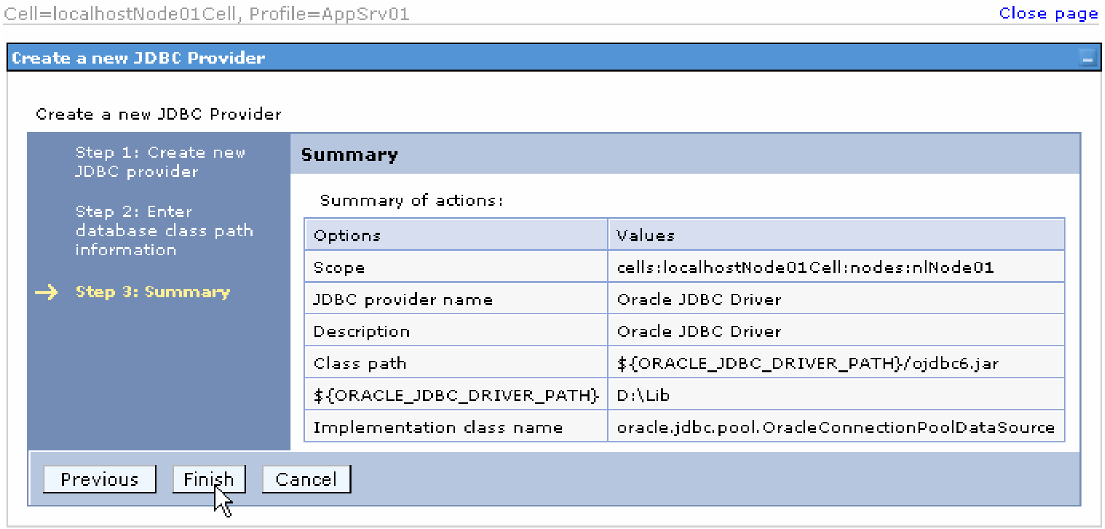
Class path:

${ORACLE_JDBC_DRIVER_PATH}/ojdbc6.jar

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

D:\Lib

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



Cell=localhostNode01Cell, Profile=AppSrv01

Create a new JDBC Provider

Create a new JDBC Provider

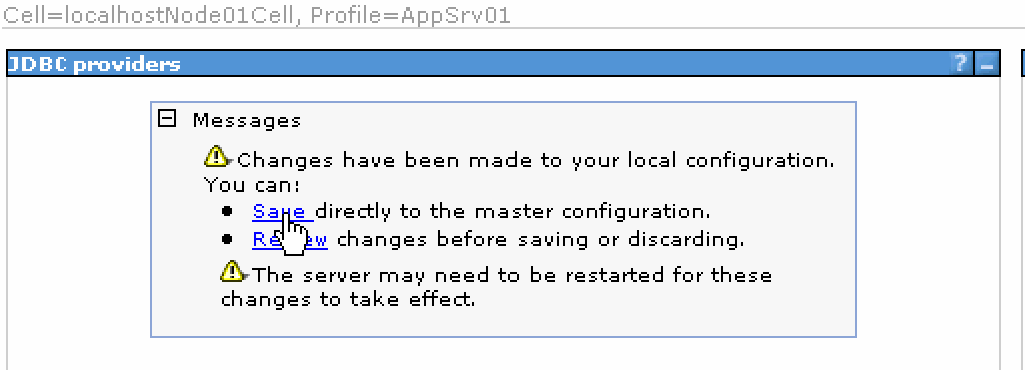
Step 1: Create new JDBC provider
Step 2: Enter database class path information
→ Step 3: Summary

Summary

Summary of actions:

Options	Values
Scope	cells:localhostNode01Cell:nodes:nlNode01
JDBC provider name	Oracle JDBC Driver
Description	Oracle JDBC Driver
Class path	${ORACLE_JDBC_DRIVER_PATH}/ojdbc6.jar
${ORACLE_JDBC_DRIVER_PATH}	D:\Lib
Implementation class name	oracle.jdbc.pool.OracleConnectionPoolDataSource

10. Click **Save**.



Cell=localhostNode01Cell, Profile=AppSrv01

JDBC providers

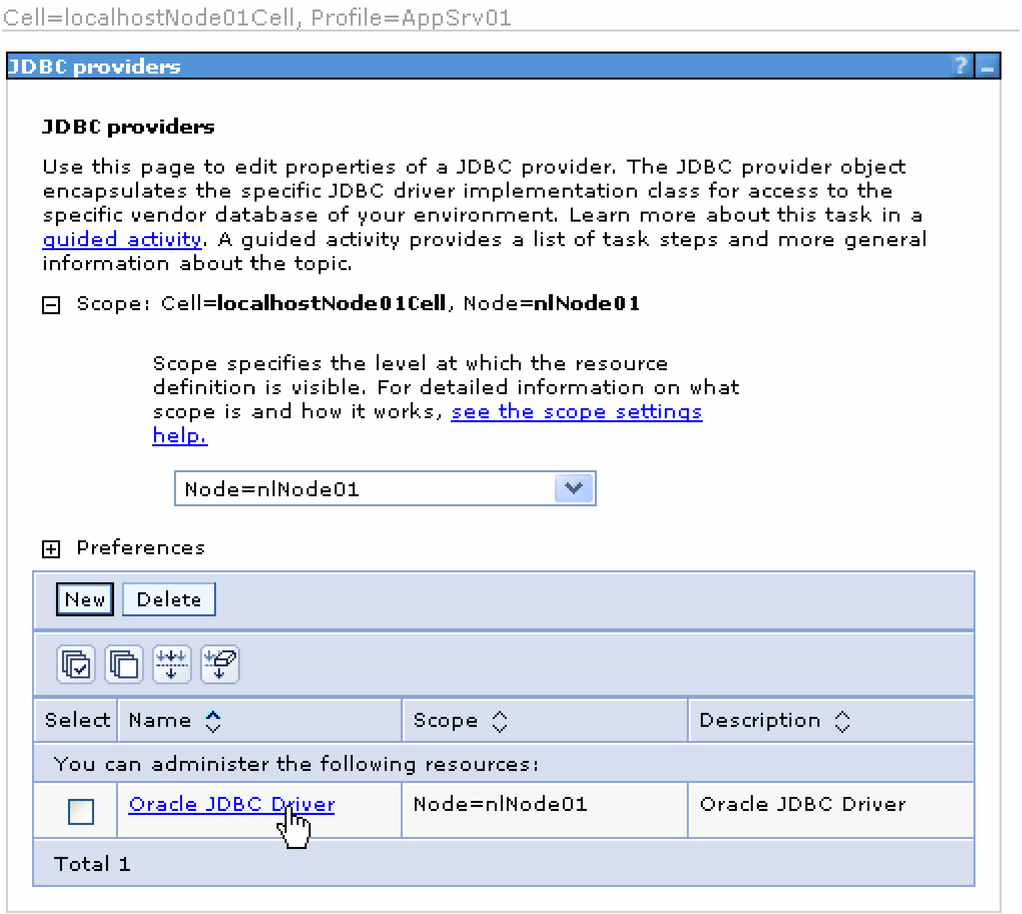
⊟ Messages

⚠ 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

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=**nlNode01**

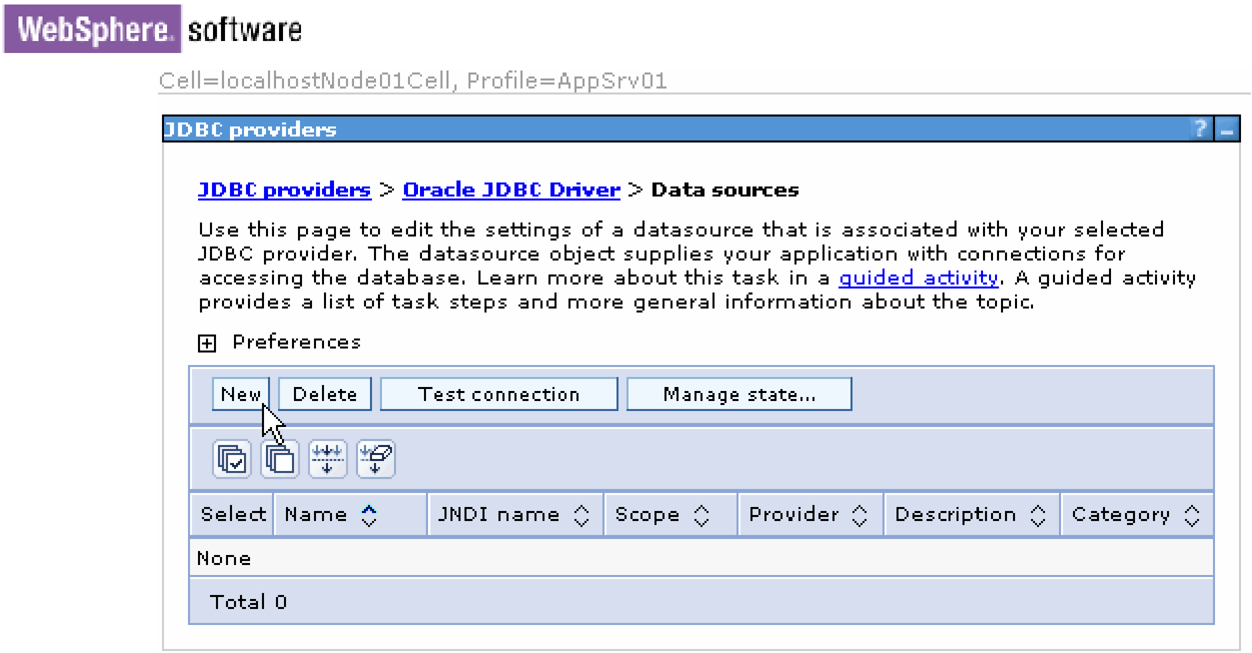
Scope specifies the level at which the resource definition is visible. For detailed information on what scope is and how it works, see the scope settings help.

Node=nlNode01

⊞ Preferences

Select	Name	Scope	Description
You can administer the following resources:			
☐	Oracle JDBC Driver	Node=nlNode01	Oracle JDBC Driver
Total 1			

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



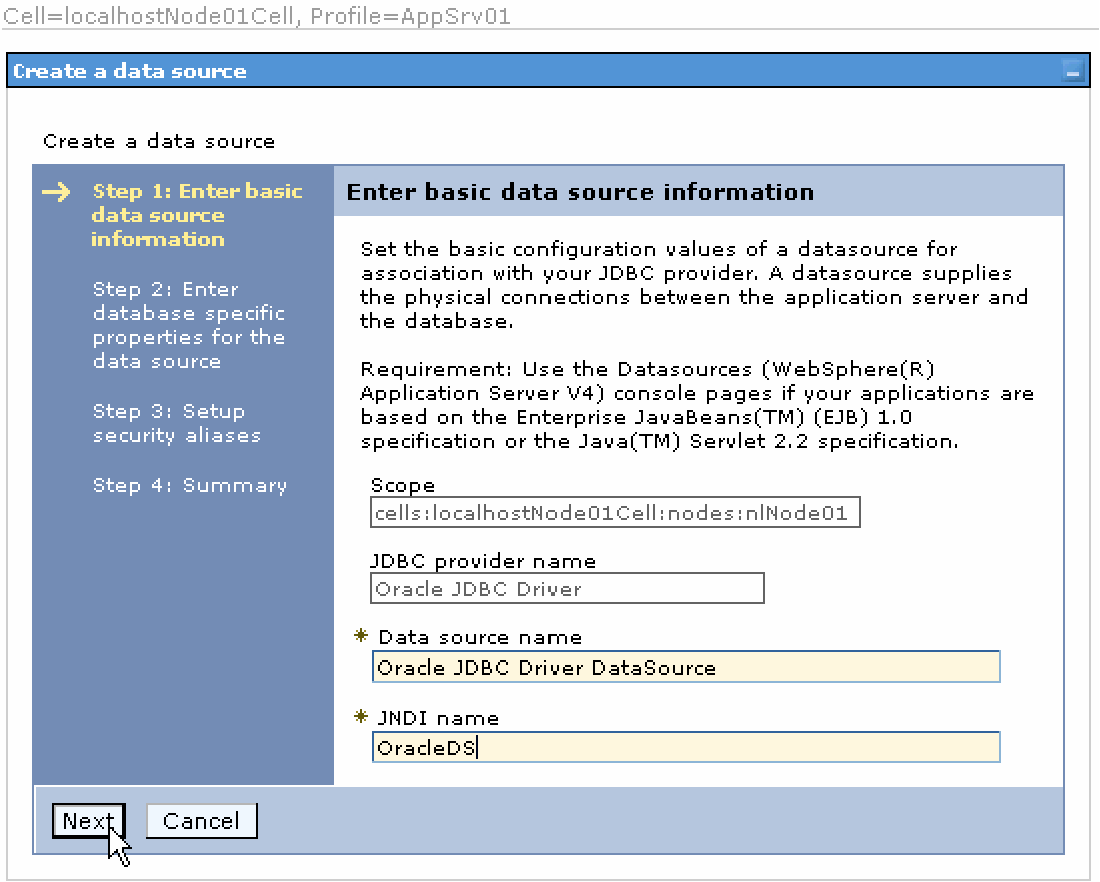
JDBC providers > Oracle JDBC Driver > Data sources

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

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
Next
Cancel

Create a data source

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.

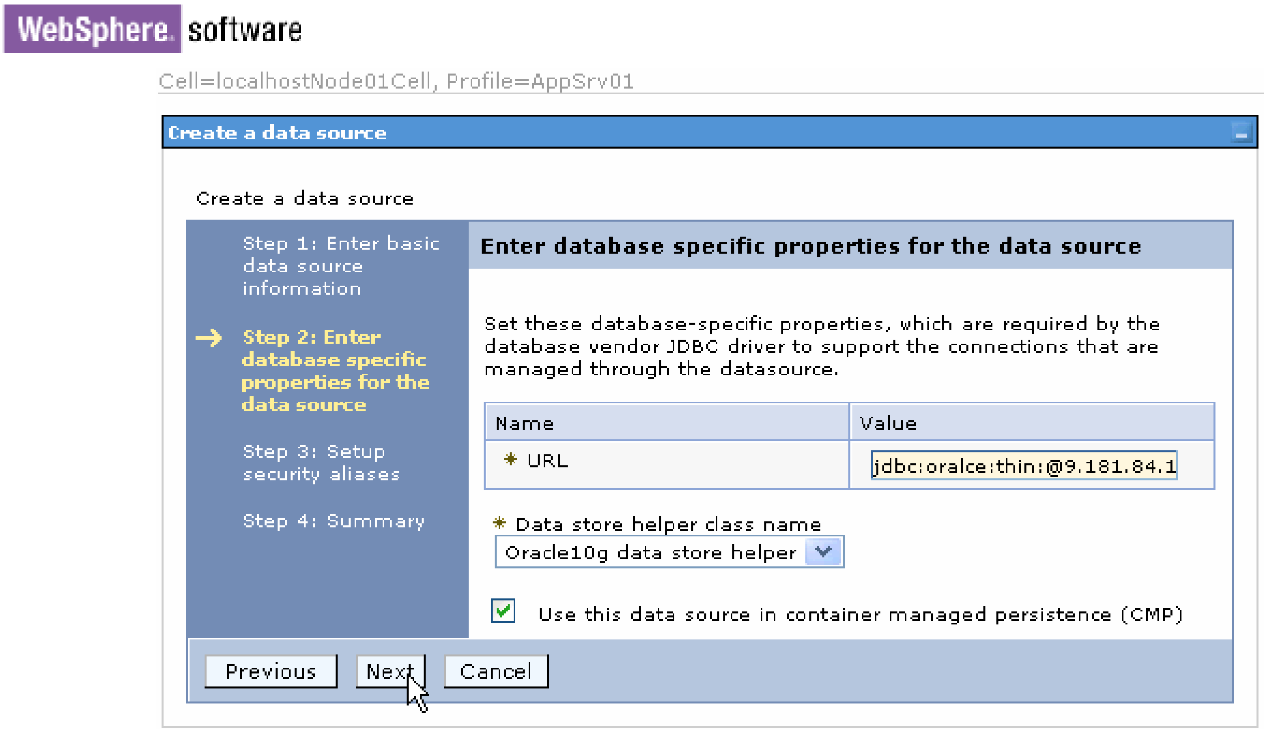
Scope: cells:localhostNode01Cell:nodes:nlNode01

JDBC provider name: Oracle JDBC Driver

* Data source name: Oracle JDBC Driver DataSource

* JNDI name: OracleDS

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
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
Previous
Next
Cancel

Cell=localhostNode01Cell, Profile=AppSrv01

Create a data source

Create a data source

Enter database specific properties for the data source

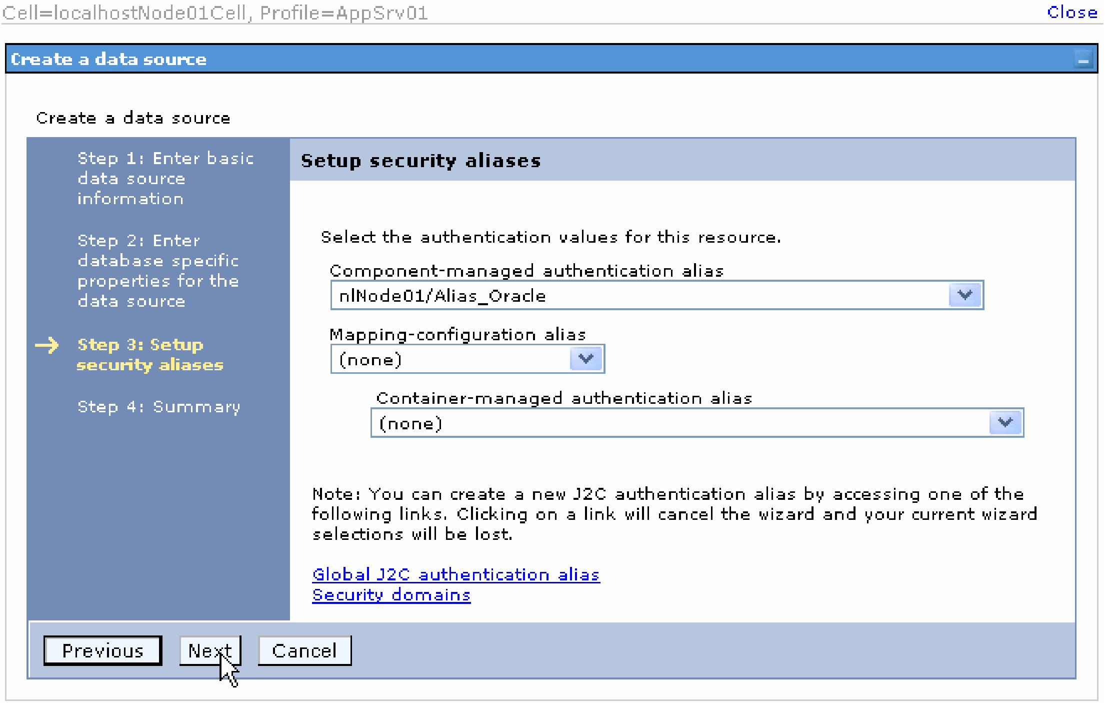
Set these database-specific properties, which are required by the database vendor JDBC driver to support the connections that are managed through the datasource.

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

* Data store helper class name
Oracle10g data store helper

☑ Use this data source in container managed persistence (CMP)

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


Close
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
Previous
Next
Cancel

Cell=localhostNode01Cell, Profile=AppSrv01

Create a data source

Create a data source

Setup security aliases

Select the authentication values for this resource.

Component-managed authentication alias
nlNode01/Alias_Oracle

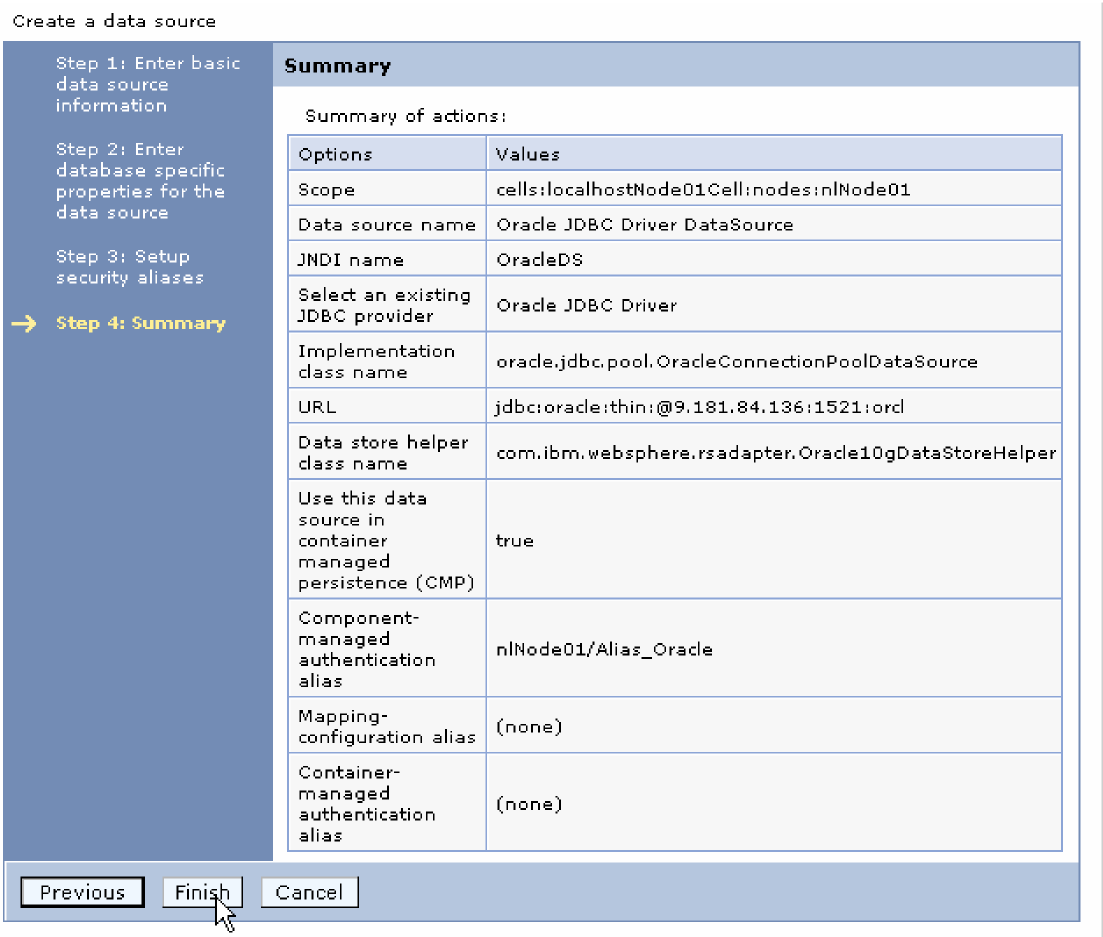
Mapping-configuration alias
(none)

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

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



Create a data source

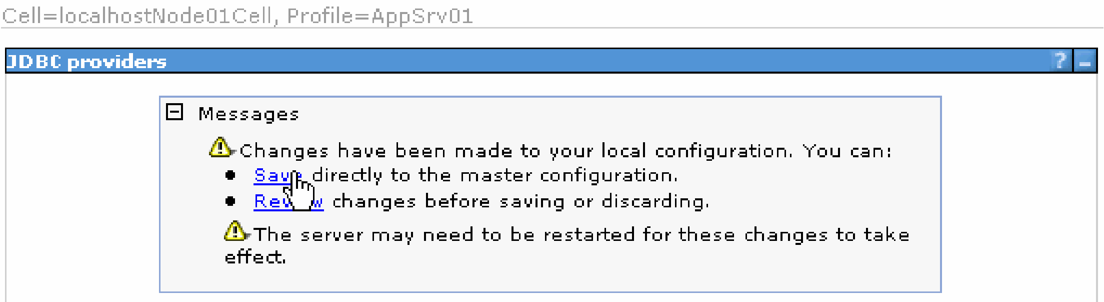
- Step 1: Enter basic data source information
- Step 2: Enter database specific properties for the data source
- Step 3: Setup security aliases
- → Step 4: Summary

Summary

Summary of actions:

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:orcl
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/Alias_Oracle
Mapping-configuration alias	(none)
Container-managed authentication alias	(none)

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



Cell=localhostNode01Cell, Profile=AppSrv01

JDBC providers

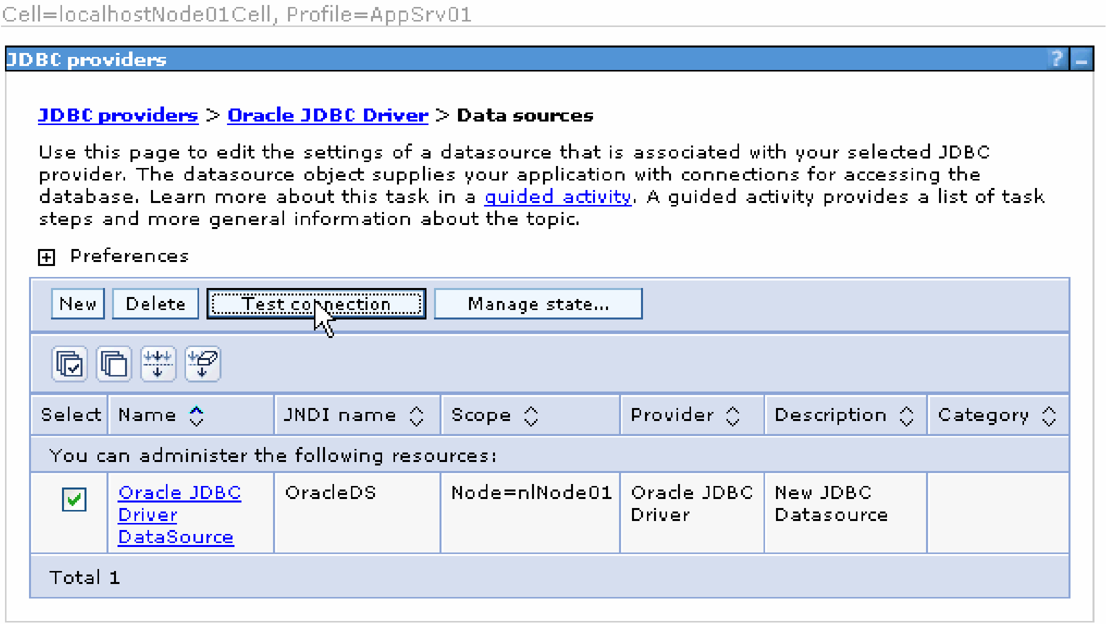
⊟ Messages

⚠ 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.

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



Cell=localhostNode01Cell, Profile=AppSrv01

JDBC providers

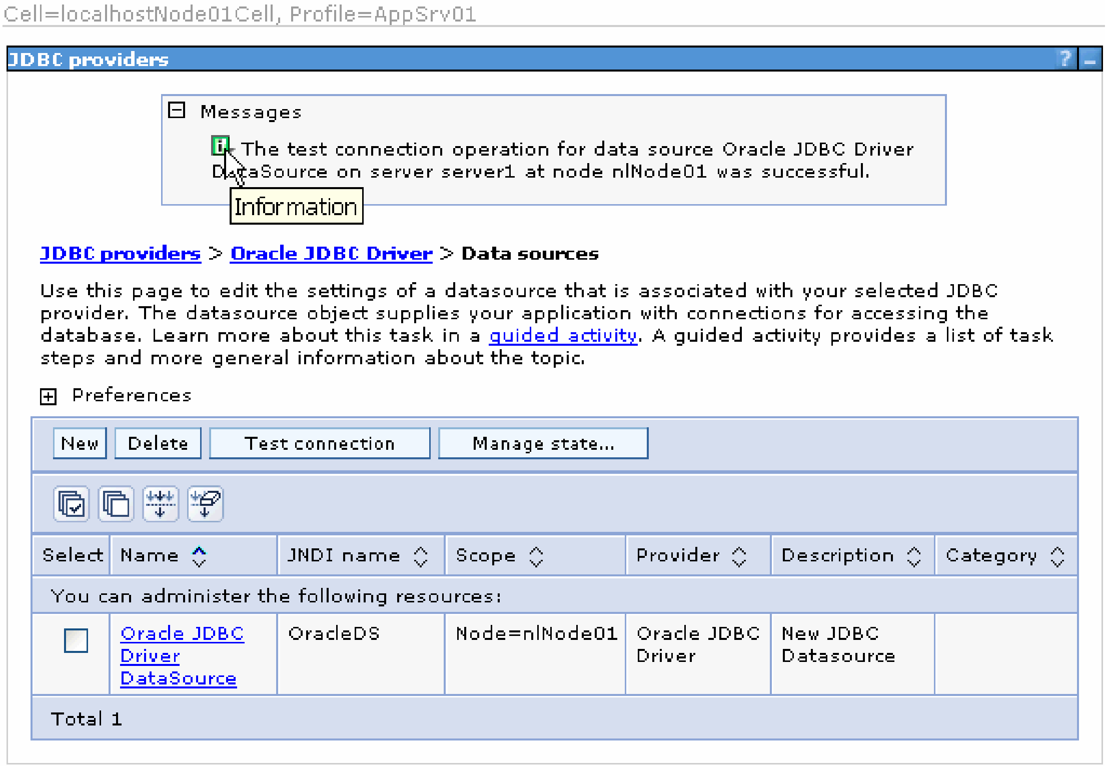
JDBC providers > Oracle JDBC Driver > Data sources

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

Select	Name	JNDI name	Scope	Provider	Description	Category
You can administer the following resources:						
☑	Oracle JDBC Driver DataSource	OracleDS	Node=nlNode01	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

⊟ Messages

The test connection operation for data source Oracle JDBC Driver DataSource on server server1 at node nlNode01 was successful.

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

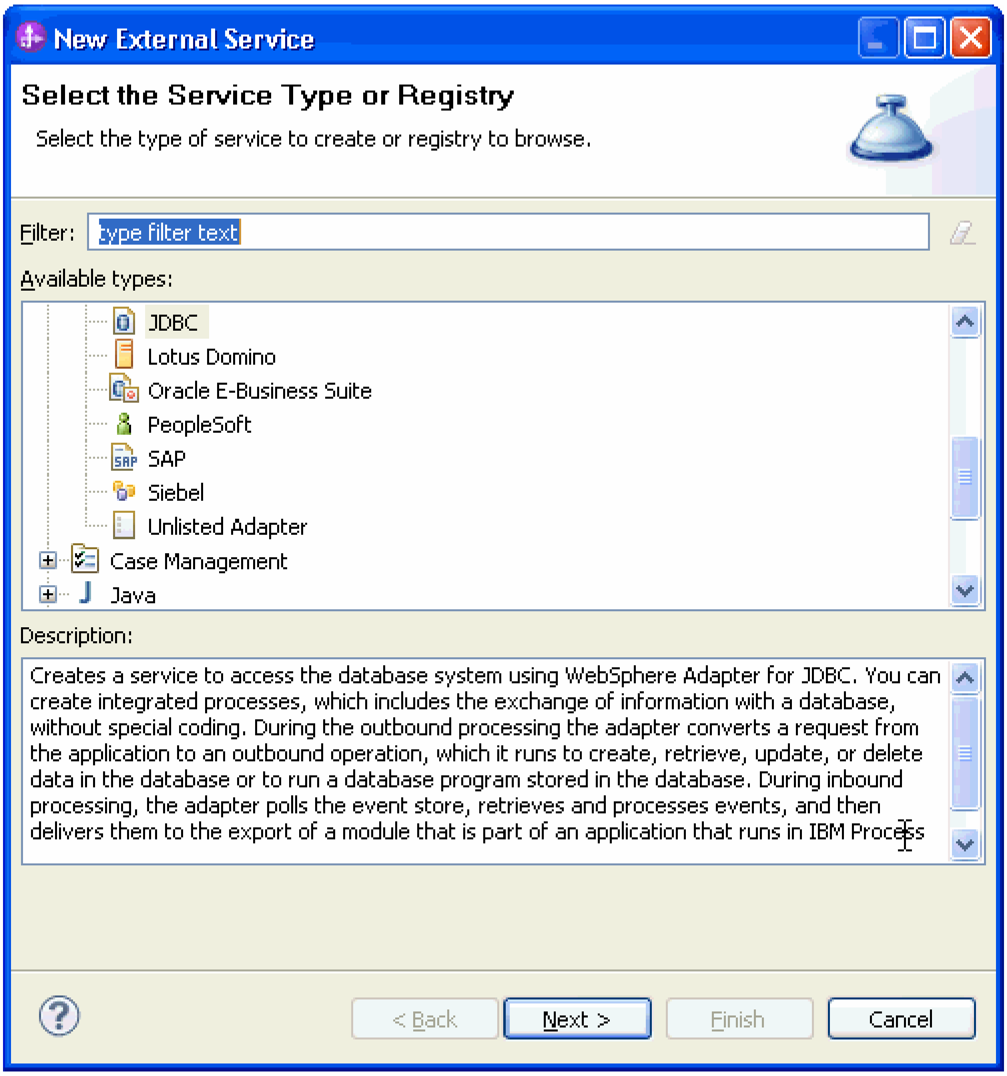
Select	Name	JNDI name	Scope	Provider	Description	Category
You can administer the following resources:						
□	Oracle JDBC Driver DataSource	OracleDS	Node=nlNode01	Oracle JDBC Driver	New JDBC Datasource	
Total 1						

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

Configure the adapter for inbound processing

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

1. Switch to the Business Integration Perspective in IBM Integration Designer 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**.



New External Service

Select the Service Type or Registry

Select the type of service to create or registry to browse.

Filter: type filter text

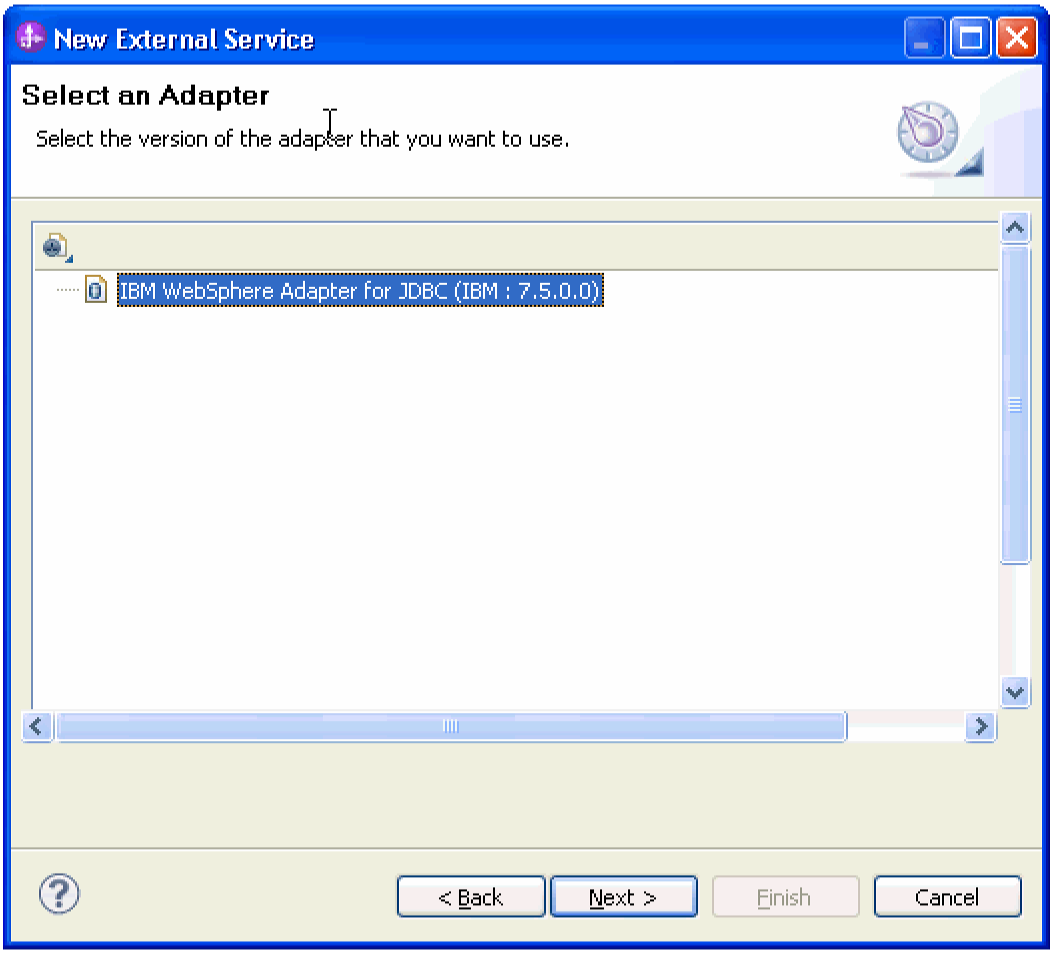
Available types:

- JDBC
- Lotus Domino
- Oracle E-Business Suite
- PeopleSoft
- SAP
- Siebel
- Unlisted Adapter
- Case Management
- Java

Description:

Creates a service to access the database system using WebSphere Adapter for JDBC. You can create integrated processes, which includes the exchange of information with a database, without special coding. During the outbound processing the adapter converts a request from the application to an outbound operation, which it runs to create, retrieve, update, or delete data in the database or to run a database program stored in the database. During inbound processing, the adapter polls the event store, retrieves and processes events, and then delivers them to the export of a module that is part of an application that runs in IBM Process

4. Select **IBM WebSphere Adapter for JDBC (IBM: 7.5.0.0)** and click **Next.**



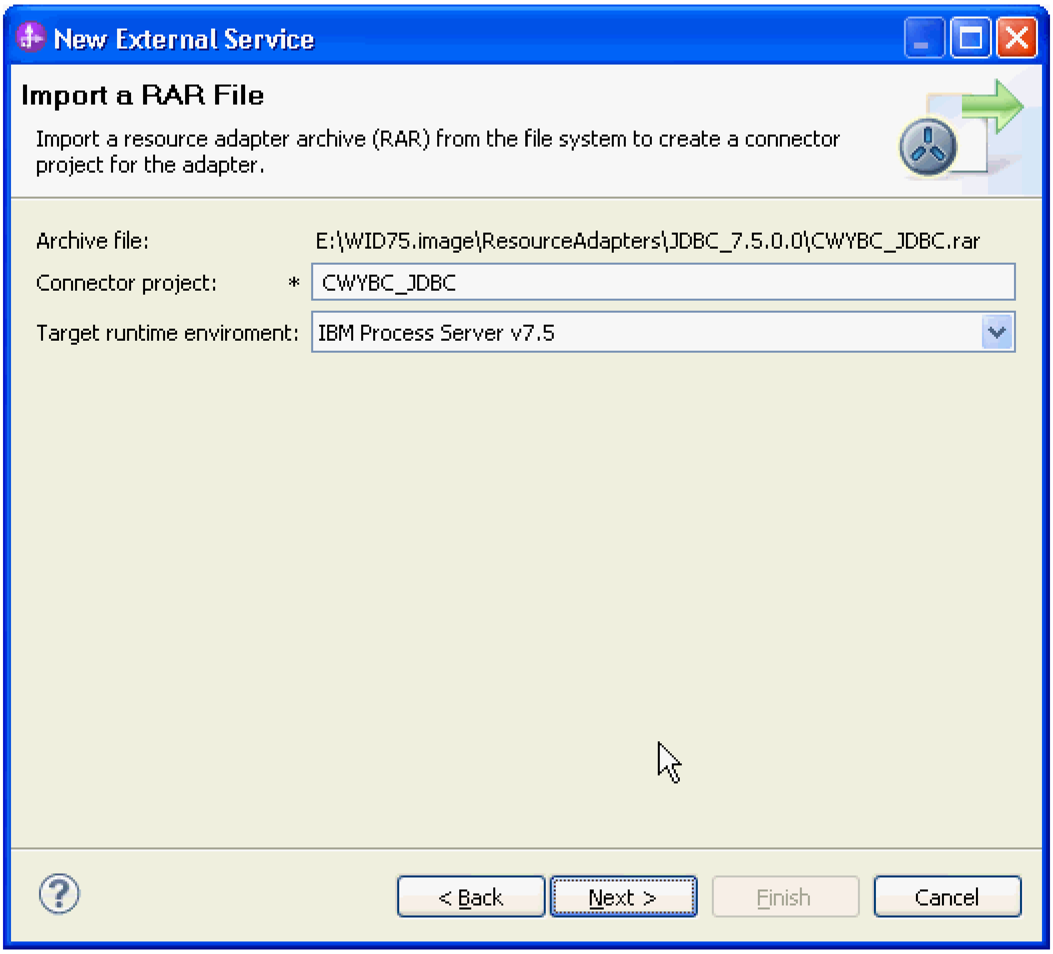
New External Service

Select an Adapter

Select the version of the adapter that you want to use.

IBM WebSphere Adapter for JDBC (IBM : 7.5.0.0)

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



Import a RAR File

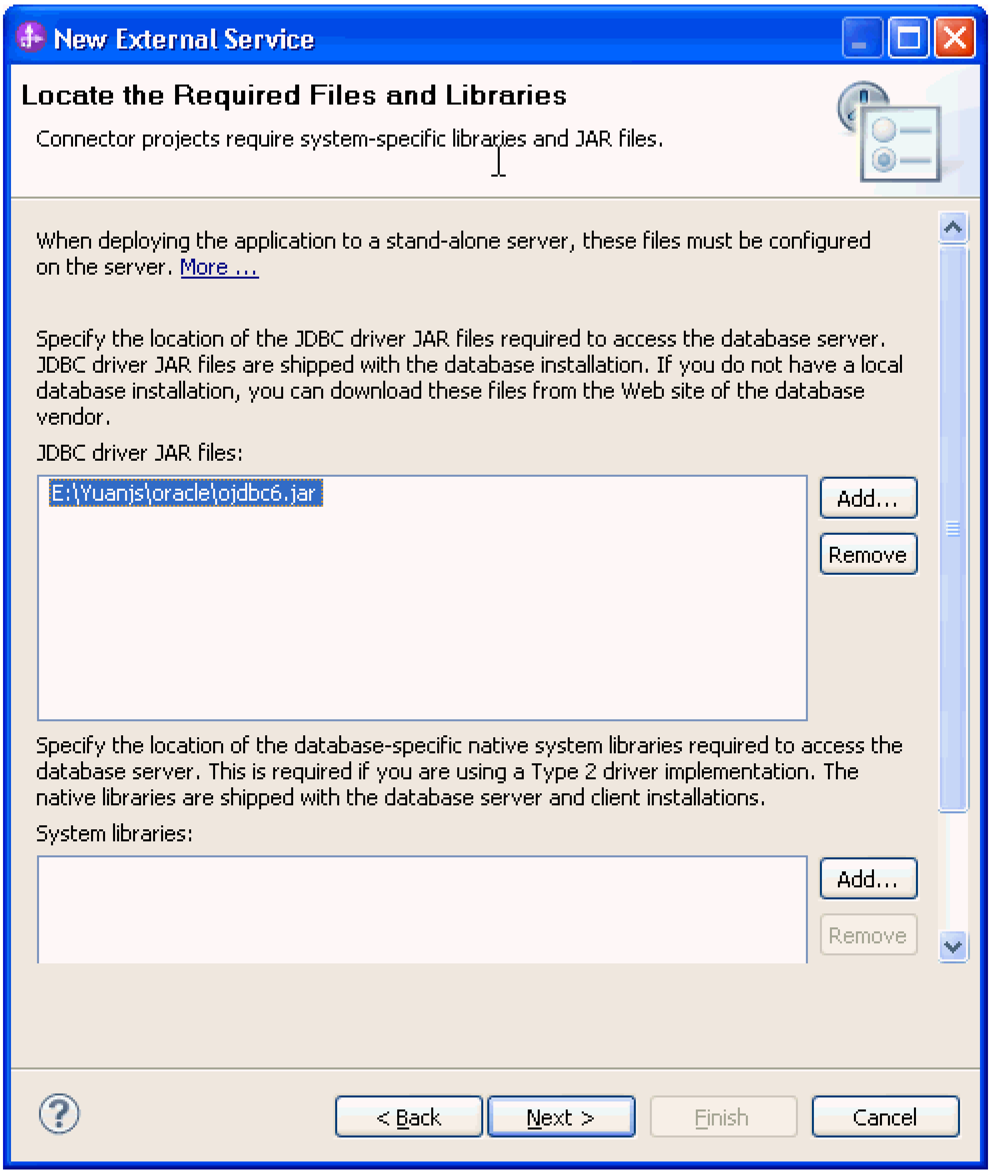
Import a resource adapter archive (RAR) from the file system to create a connector project for the adapter.

Archive file: E:\WID75.image\ResourceAdapters\JDBC_7.5.0.0\CWYBC_JDBC.rar

Connector project: * CWYBC_JDBC

Target runtime enviroment: IBM Process Server v7.5

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**.



Locate the Required Files and Libraries

Connector projects require system-specific libraries and JAR files.

When deploying the application to a stand-alone server, these files must be configured on the server. More ...

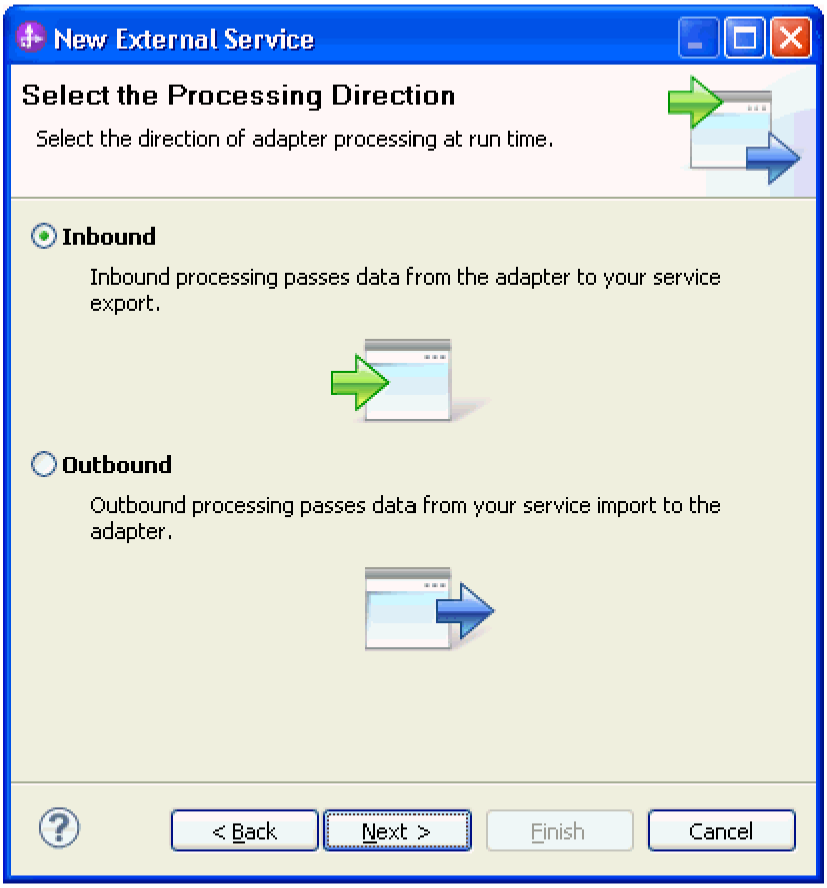
Specify the location of the JDBC driver JAR files required to access the database server. JDBC driver JAR files are shipped with the database installation. If you do not have a local database installation, you can download these files from the Web site of the database vendor.

JDBC driver JAR files: E:\Yuanjs\oracle\ojdbc6.jar

Specify the location of the database-specific native system libraries required to access the database server. This is required if you are using a Type 2 driver implementation. The native libraries are shipped with the database server and client installations.

System libraries:

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



New External Service

Select the Processing Direction

Select the direction of adapter processing at run time.

● **Inbound**

Inbound processing passes data from the adapter to your service export.

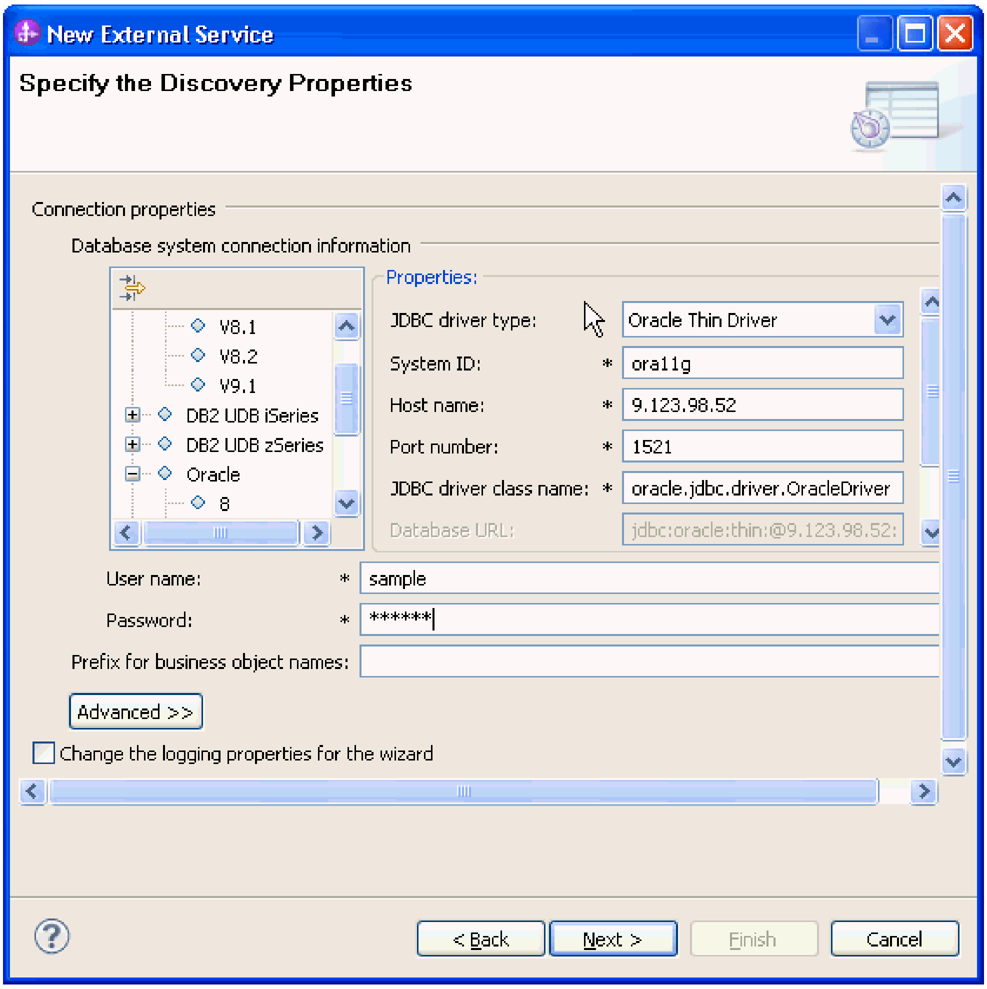
○ **Outbound**

Outbound processing passes data from your service import to the adapter.

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**.



Specify the Discovery Properties

Connection properties

Database system connection information

Properties:

JDBC driver type: Oracle Thin Driver

System ID: * ora11g

Host name: * 9.123.98.52

Port number: * 1521

JDBC driver class name: * oracle.jdbc.driver.OracleDriver

Database URL: jdbc:oracle:thin:@9.123.98.52:

User name: * sample

Password: * ******

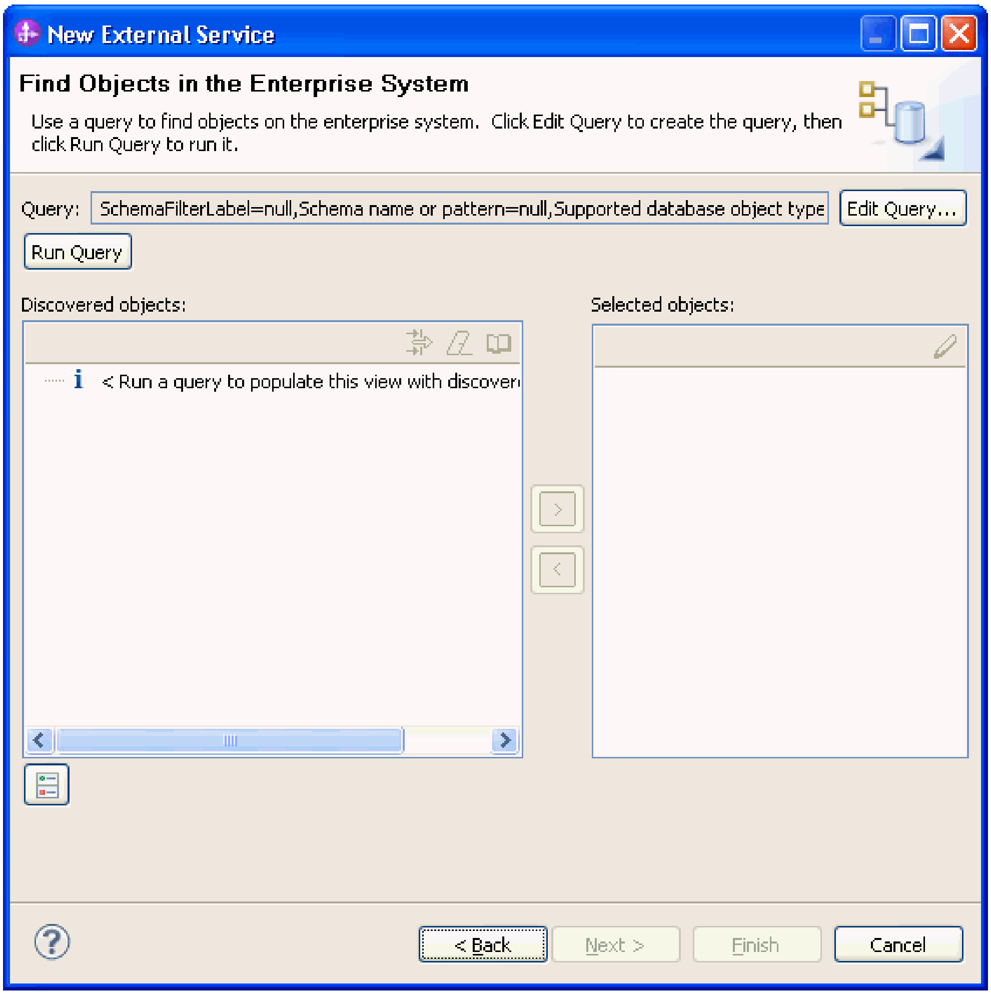
Prefix for business object names:

□ Change the logging properties for the wizard

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**.

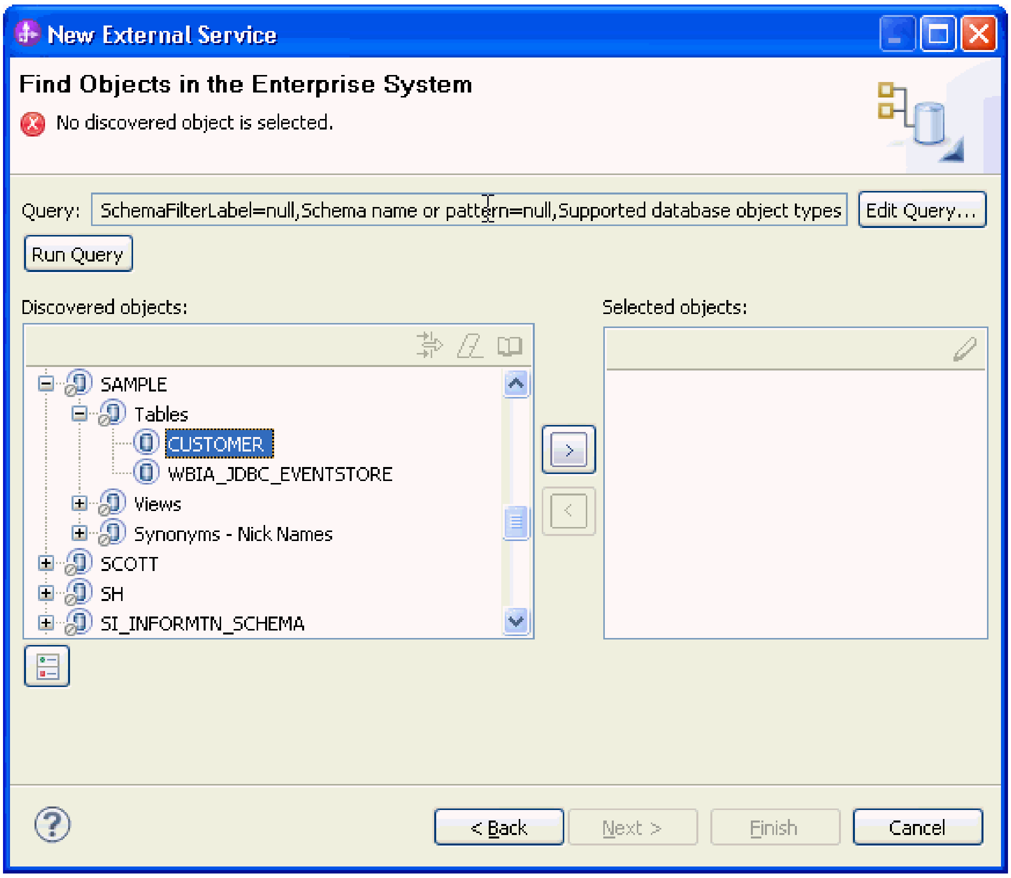


Find Objects in the Enterprise System

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 type

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



New External Service

Find Objects in the Enterprise System

No discovered object is selected.

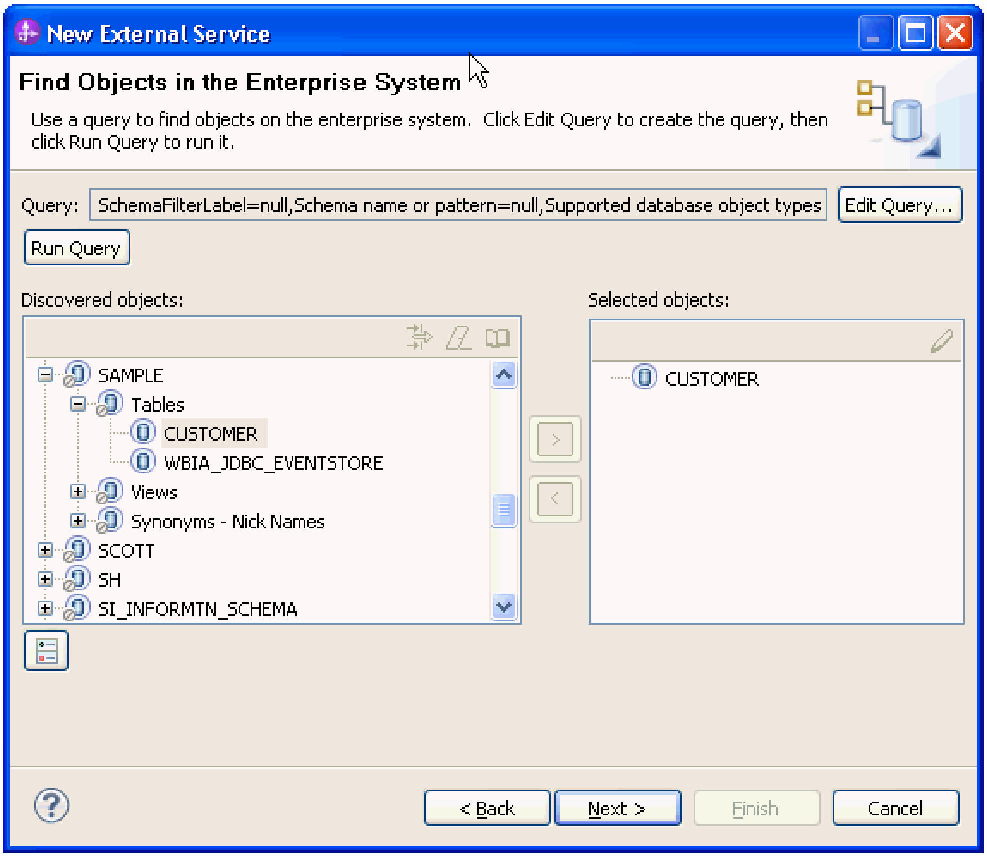
Query: SchemaFilterLabel=null,Schema name or pattern=null,Supported database object types

Discovered objects:

```
SAMPLE
  Tables
    CUSTOMER
    WBIA_JDBC_EVENTSTORE
  Views
  Synonyms - Nick Names
SCOTT
SH
SI_INFORMTN_SCHEMA
```

Selected objects:

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

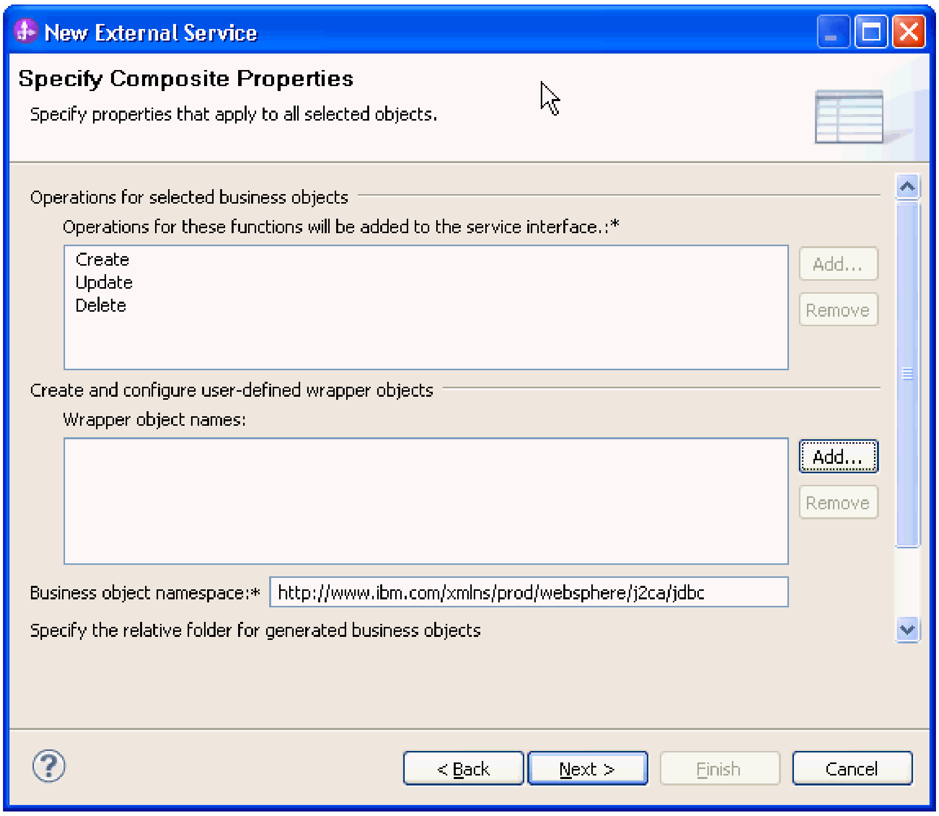


New External Service
Find Objects in the Enterprise System
Use a query to find objects on the enterprise system. Click Edit Query to create the query, then click Run Query to run it.
Query:
SchemaFilterLabel=null,Schema name or pattern=null,Supported database object types
Edit Query...
Run Query
Discovered objects:
SAMPLE
Tables
CUSTOMER
WBIA_JDBC_EVENTSTORE
Views
Synonyms - Nick Names
SCOTT
SH
SI_INFORMTN_SCHEMA
Selected objects:
CUSTOMER
< Back
Next >
Finish
Cancel

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**.



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

Create and configure user-defined wrapper objects

Wrapper object names:

Business object namespace:* http://www.ibm.com/xmlns/prod/websphere/j2ca/jdbc

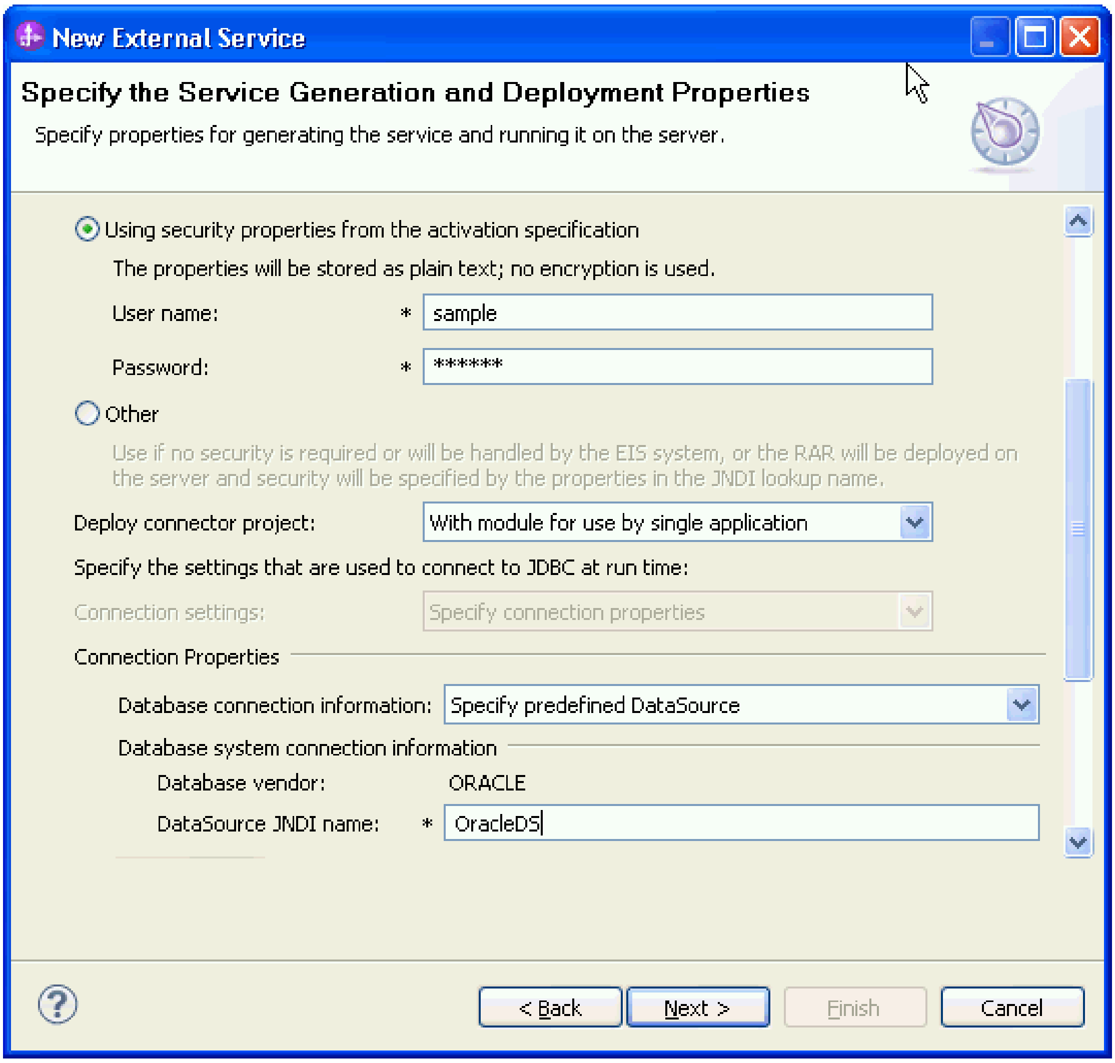
Specify the relative folder for generated business objects

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**.



Specify the Service Generation and Deployment Properties

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

● Using security properties from the activation specification

The properties will be stored as plain text; no encryption is used.

User name: * sample

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: With module for use by single application

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

Connection settings: Specify connection properties

Connection Properties

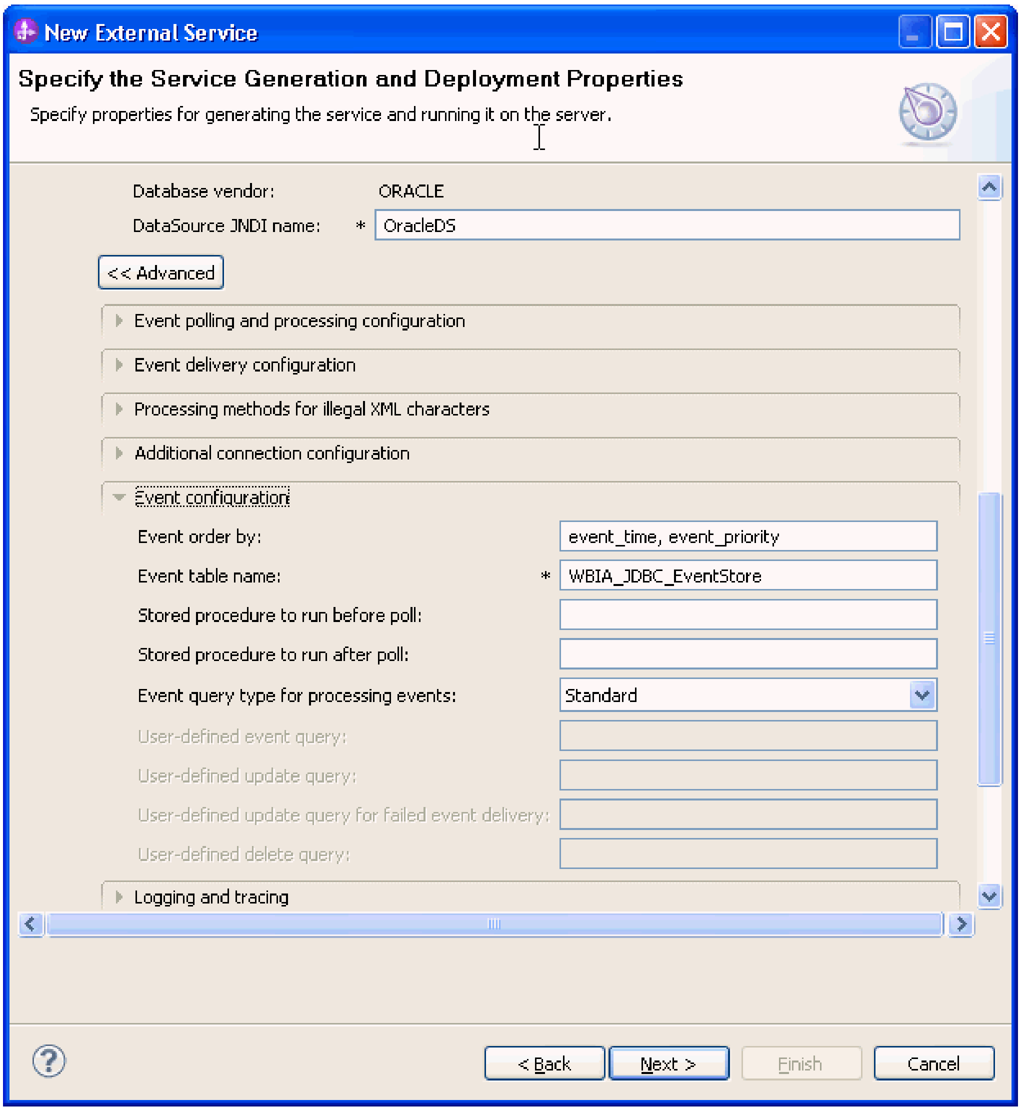
Database connection information: Specify predefined DataSource

Database system connection information

Database vendor: ORACLE

DataSource JNDI name: * OracleDS

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



Specify the Service Generation and Deployment Properties

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

Database vendor: ORACLE

DataSource JNDI name: * OracleDS

<< Advanced
Event polling and processing configuration
Event delivery configuration
Processing methods for illegal XML characters
Additional connection configuration
Event configuration

Event order by: event_time, event_priority

Event table name: * WBIA_JDBC_EventStore

Stored procedure to run before poll:

Stored procedure to run after poll:

Event query type for processing events: Standard

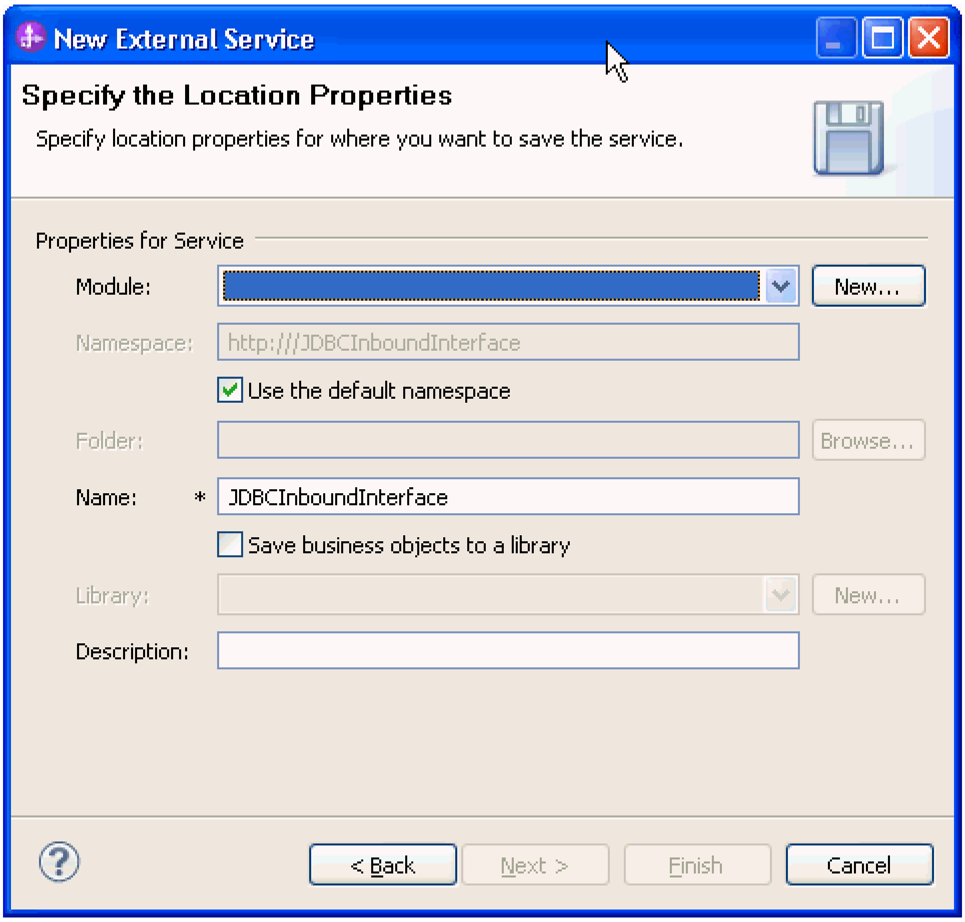
User-defined event query:

User-defined update query:

User-defined update query for failed event delivery:

User-defined delete query:

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



Specify the Location Properties

Specify location properties for where you want to save the service.

Properties for Service

Module:

Namespace: http:///JDBCInboundInterface

☑ Use the default namespace

Folder:

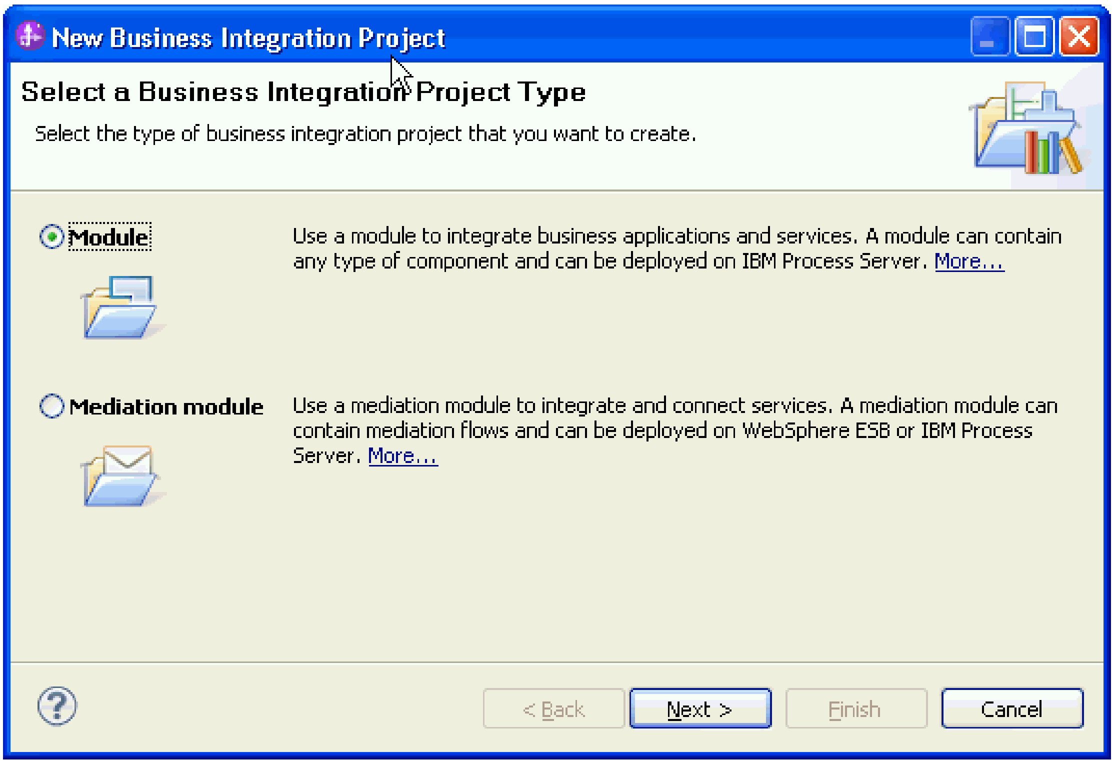
Name: * JDBCInboundInterface

□ Save business objects to a library

Library:

Description:

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



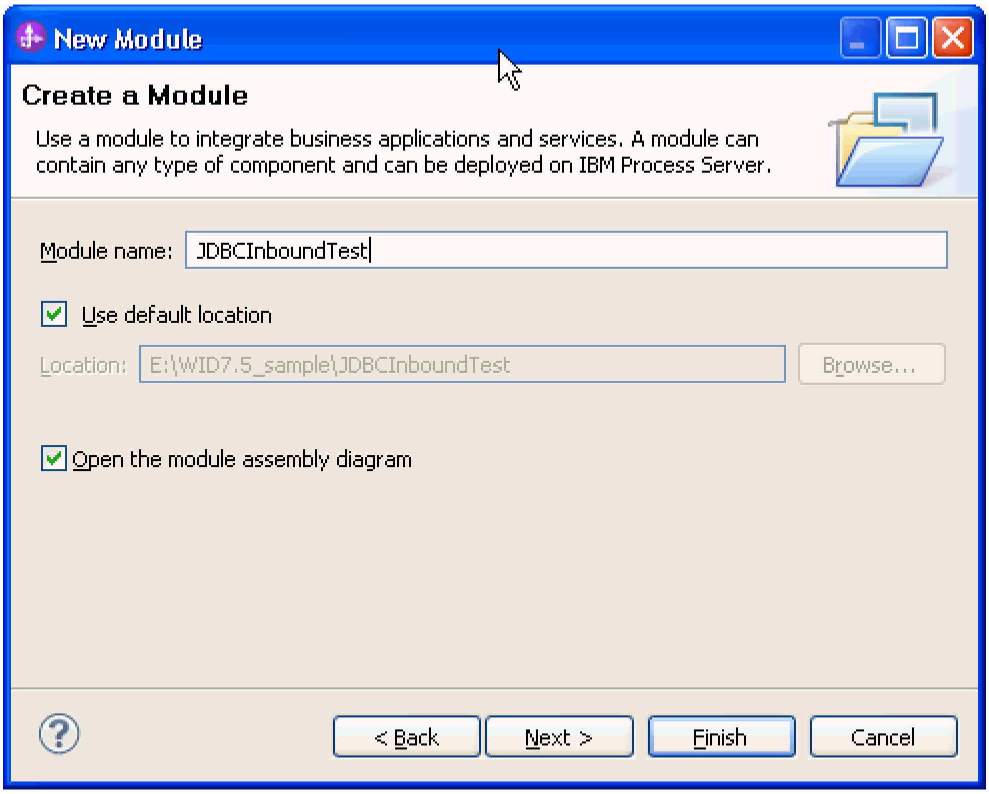
Select a Business Integration Project Type

Select the type of business integration project that you want to create.

● Module — Use a module to integrate business applications and services. A module can contain any type of component and can be deployed on IBM Process Server. More...

○ Mediation module — Use a mediation module to integrate and connect services. A mediation module can contain mediation flows and can be deployed on WebSphere ESB or IBM Process Server. More...

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



Create a Module

Use a module to integrate business applications and services. A module can contain any type of component and can be deployed on IBM Process Server.

Module name: JDBCInboundTest

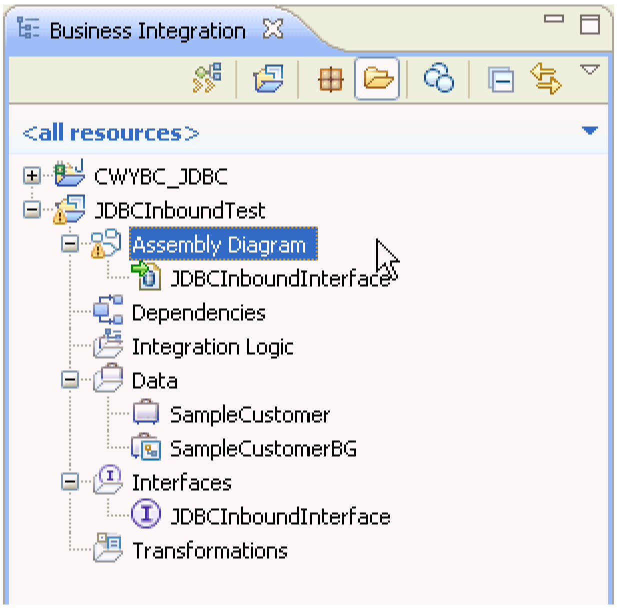
☑ Use default location

Location: E:\WID7.5_sample\JDBCInboundTest

☑ Open the module assembly diagram

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

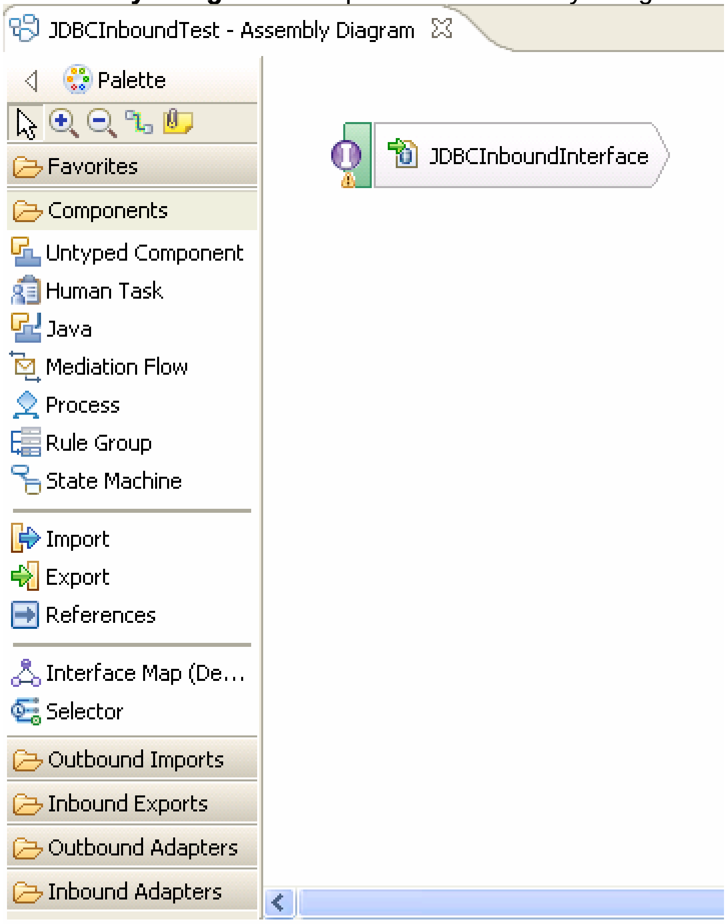
7. Verify the results.



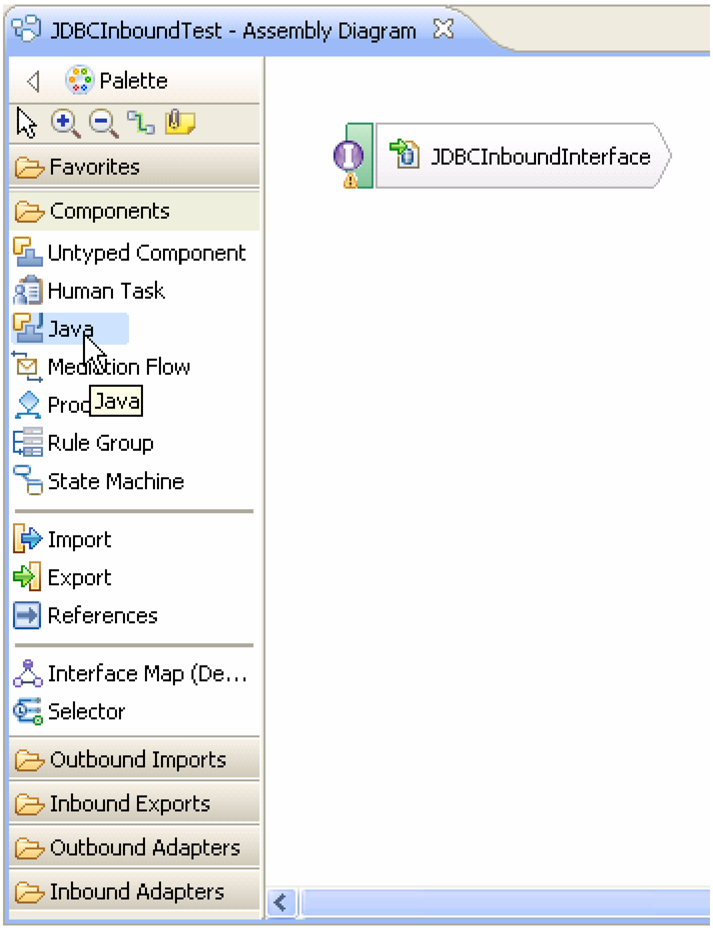
Set up the components to be part of the Inbound environment

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

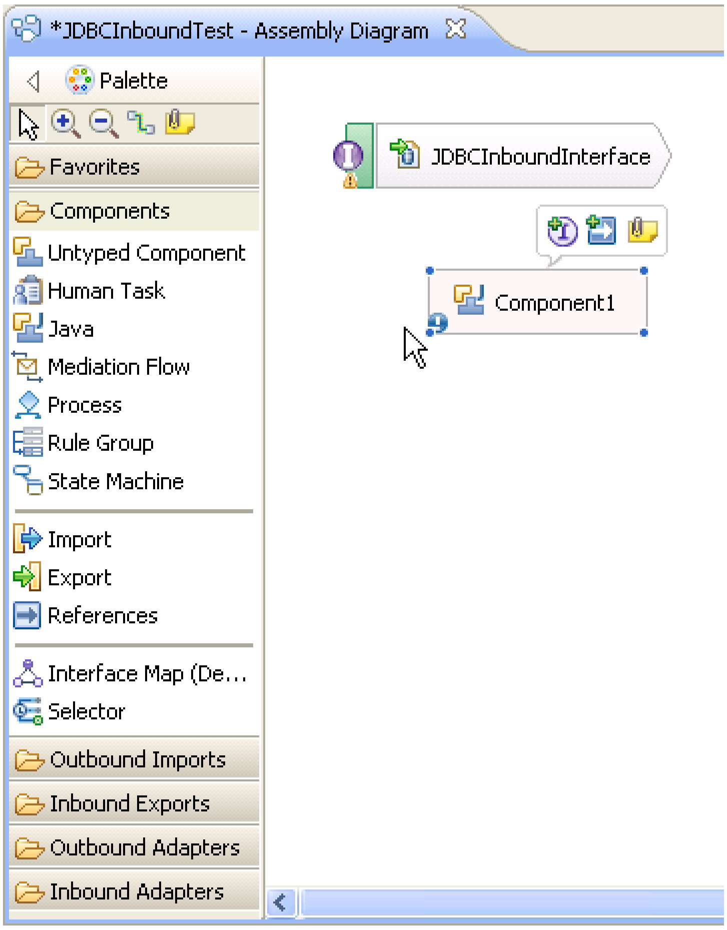
1. In the Business Integration view, double click **JDBCInboundTest > Assembly Diagram** to open the Assembly Diagram.



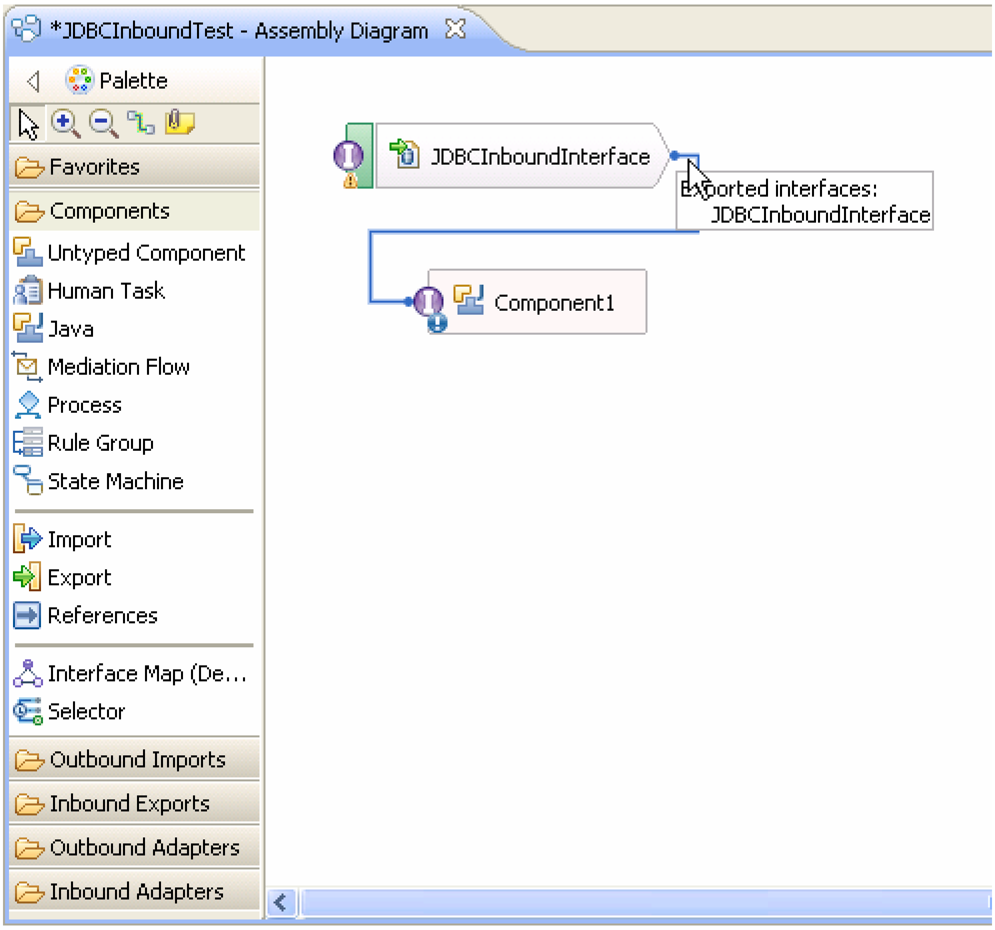
2. From the Palette, select the **Java** component and drop it on the assembly diagram.



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

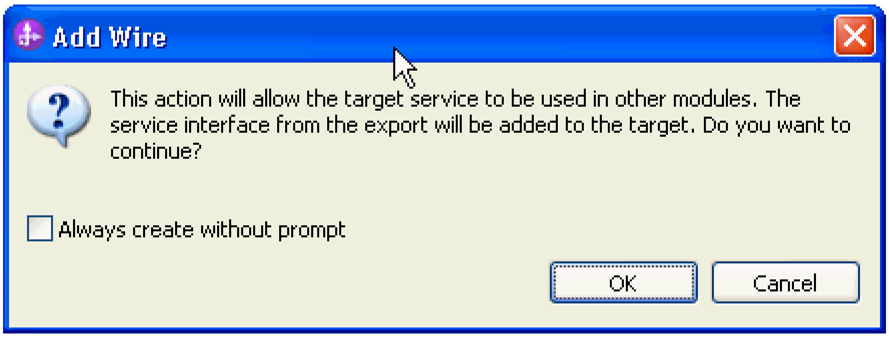


*JDBCInboundTest - Assembly Diagram
Palette
Favorites
Components
Untyped Component
Human Task
Java
Mediation Flow
Process
Rule Group
State Machine
Import
Export
References
Interface Map (De...
Selector
Outbound Imports
Inbound Exports
Outbound Adapters
Inbound Adapters
JDBCInboundInterface
Component1



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

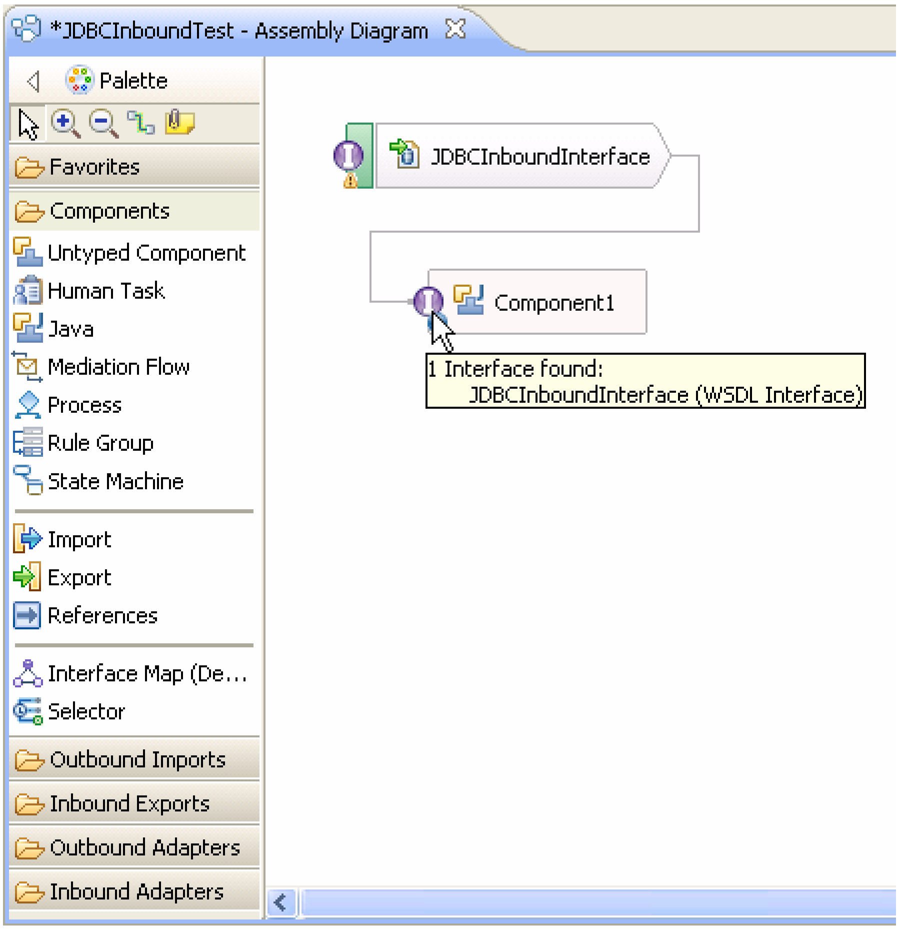
Note: Before the preceding window, i.e., before wiring you will see the following window. Click **OK**.



Add Wire

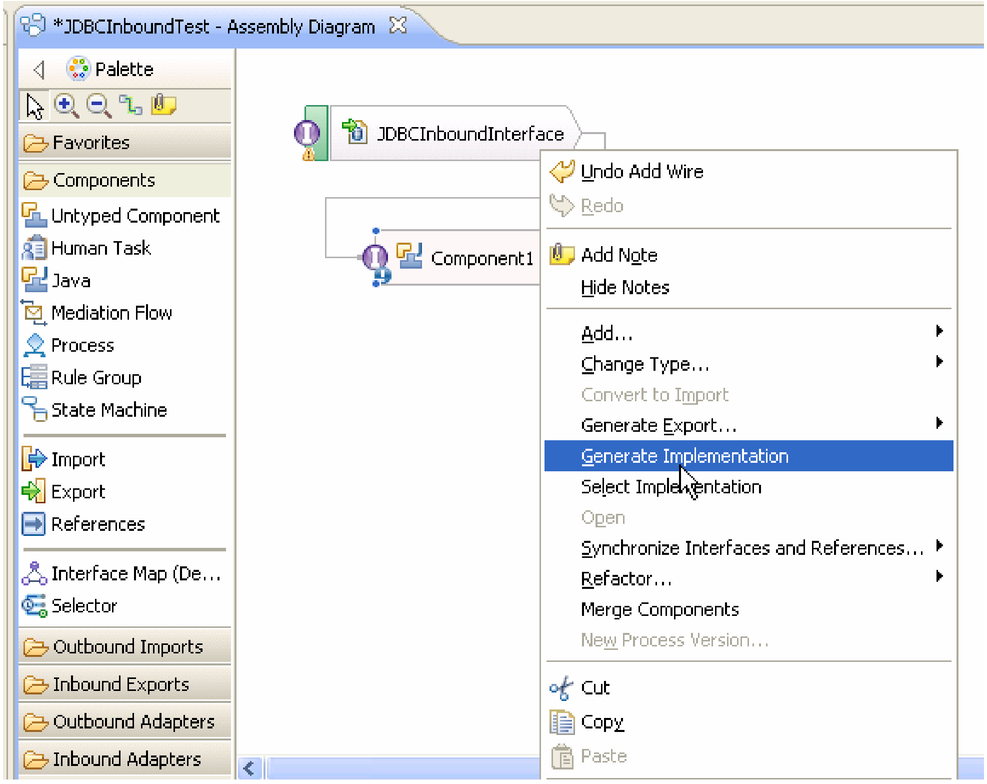
This action will allow the target service to be used in other modules. The service interface from the export will be added to the target. Do you want to continue?

□ Always create without prompt



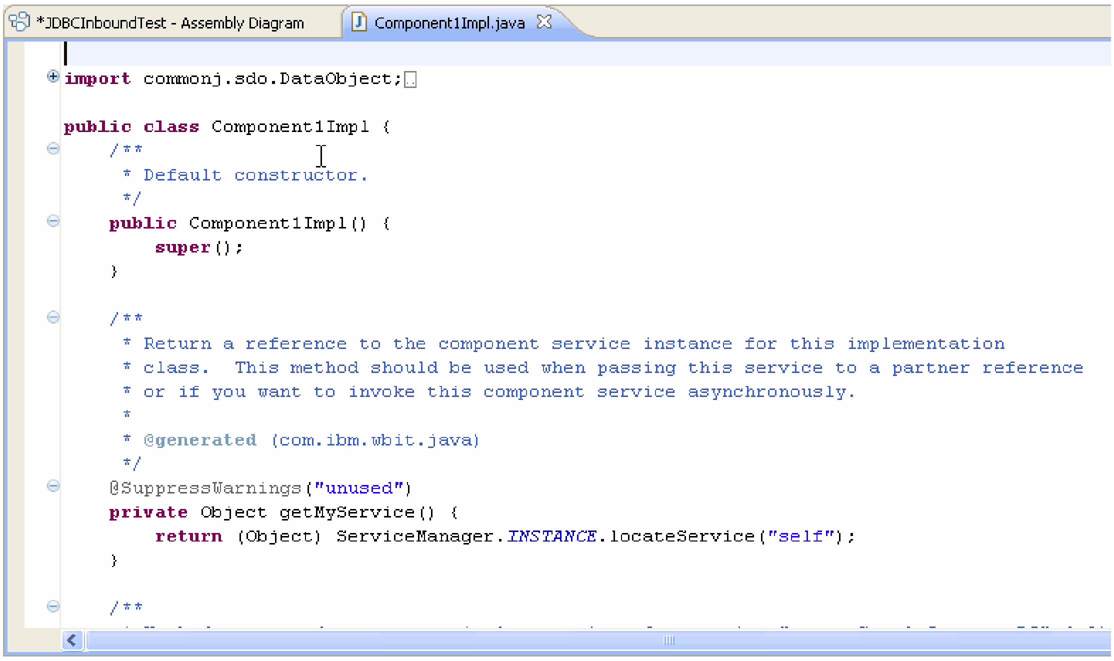
1 Interface found:
JDBCInboundInterface (WSDL Interface)

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



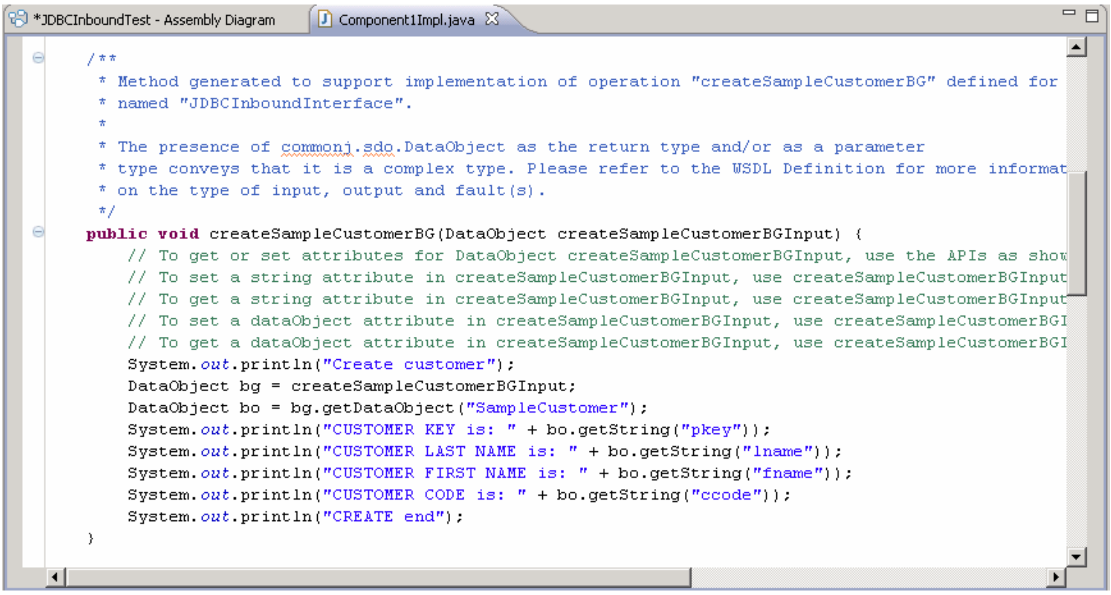
5. Highlight the default package and select **OK**.

   The Java Editor displays the Component1Impl.java file.



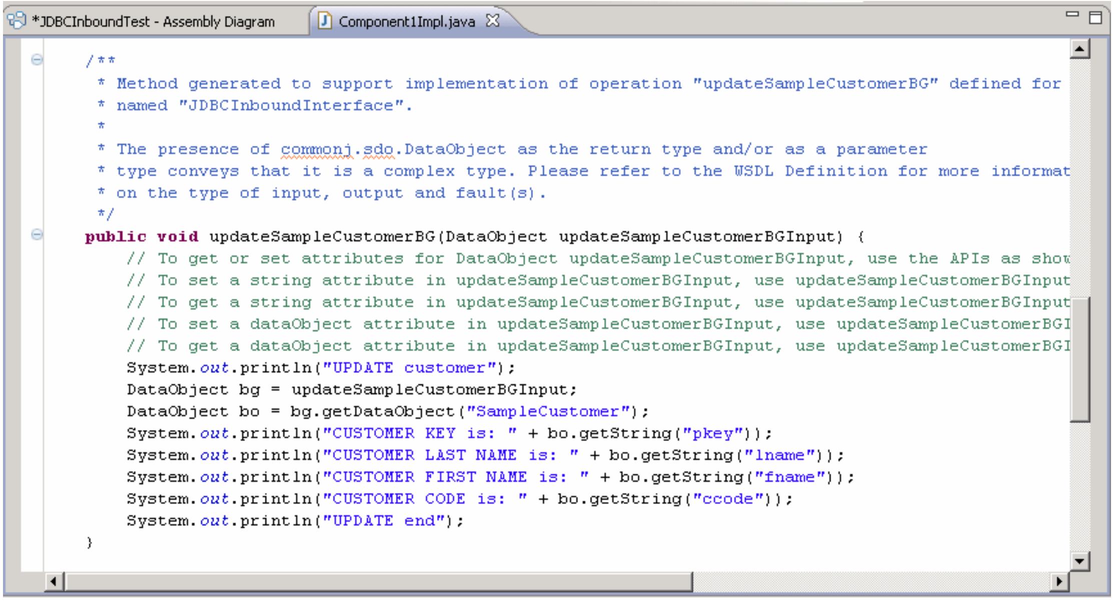
```
import commonj.sdo.DataObject;

public class 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 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");
    }

    /**
```

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:



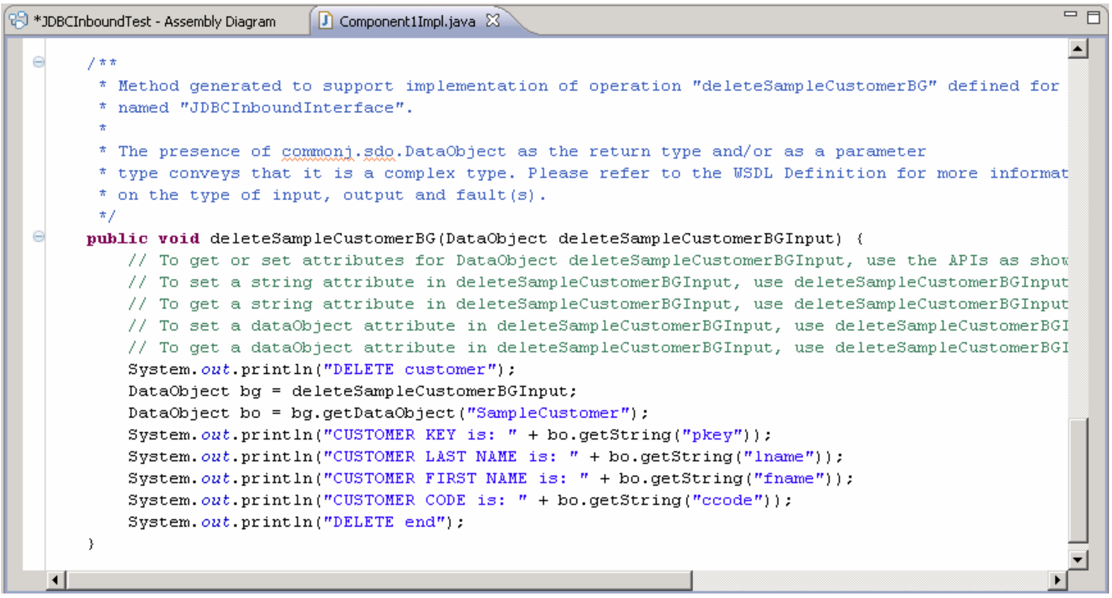
```
    /**
     * Method generated to support implementation of operation "createSampleCustomerBG" defined for
     * named "JDBCInboundInterface".
     *
     * The presence of commonj.sdo.DataObject as the return type and/or as a parameter
     * type conveys that it is a complex type. Please refer to the WSDL Definition for more informat
     * 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 show
        // 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 createSampleCustomerBGI
        // To get a dataObject attribute in createSampleCustomerBGInput, use createSampleCustomerBGI
        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 informat
 * 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 show
    // 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 updateSampleCustomerBGI
    // To get a dataObject attribute in updateSampleCustomerBGInput, use updateSampleCustomerBGI
    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 informat
 * 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 show
    // 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 deleteSampleCustomerBGI
    // To get a dataObject attribute in deleteSampleCustomerBGInput, use deleteSampleCustomerBGI
    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.

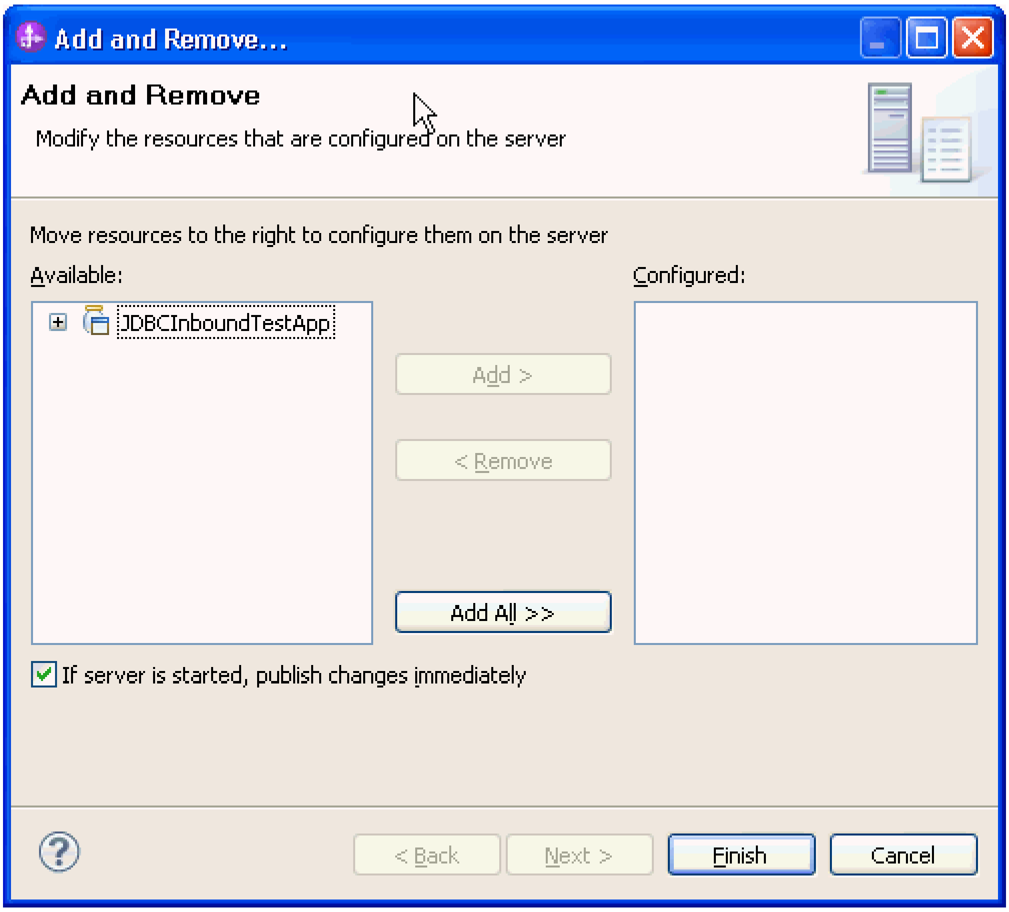
10. Close and save the Assembly Diagram. Wait for the workspace to complete building.

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 IBM Integration Designer integration test client. To do this, you must add the SCA module you created earlier to the server using the **Servers** view in IBM Integration Designer.

Steps for adding the SCA module to the server:

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



Add and Remove...

Add and Remove

Modify the resources that are configured on the server

Move resources to the right to configure them on the server

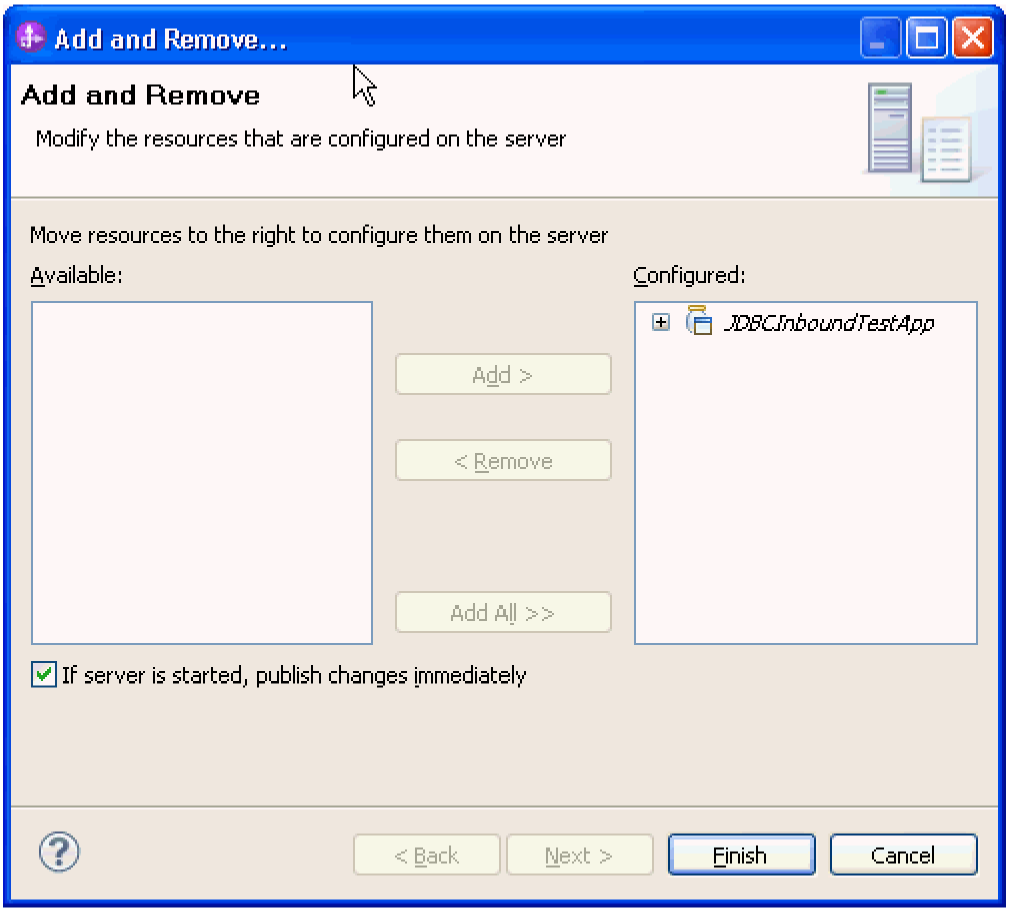
Available:

JDBCInboundTestApp

Configured:

☑ If server is started, publish changes immediately

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



Add and Remove

Modify the resources that are configured on the server

Move resources to the right to configure them on the server

Available:

Configured:

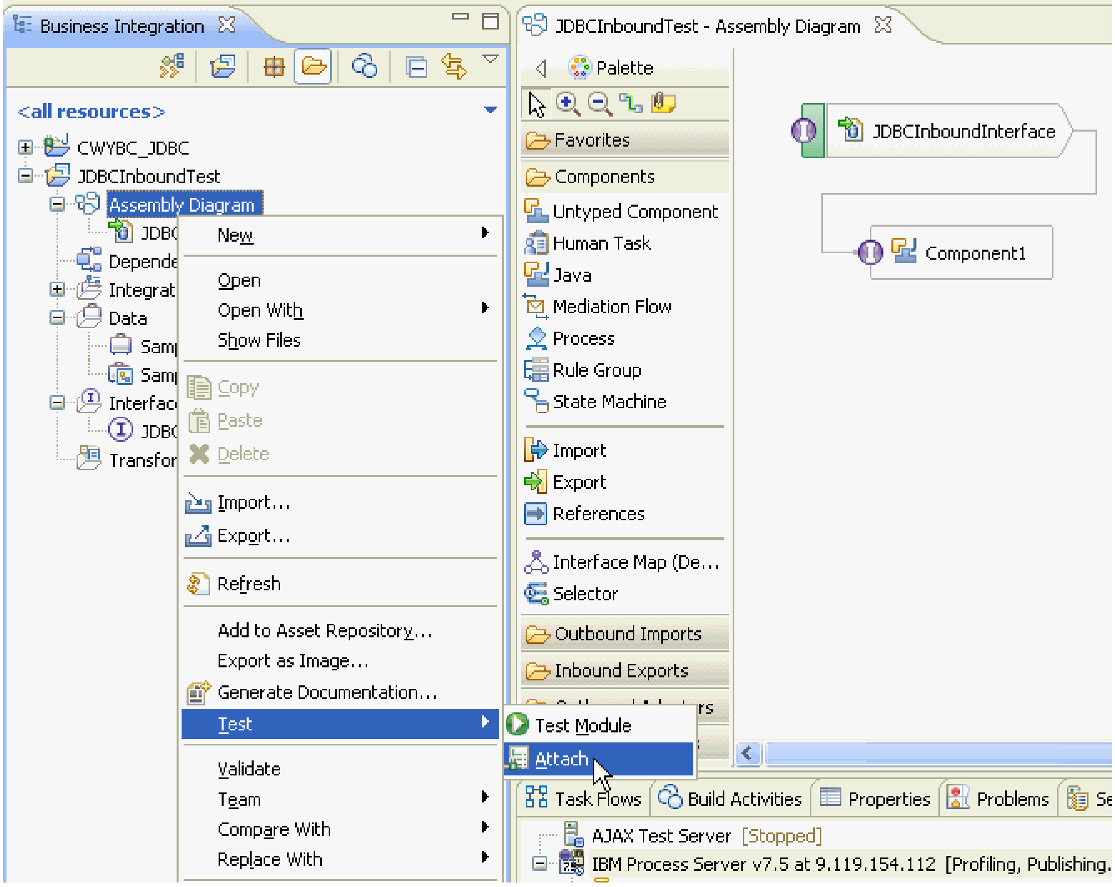
JDBCInboundTestApp

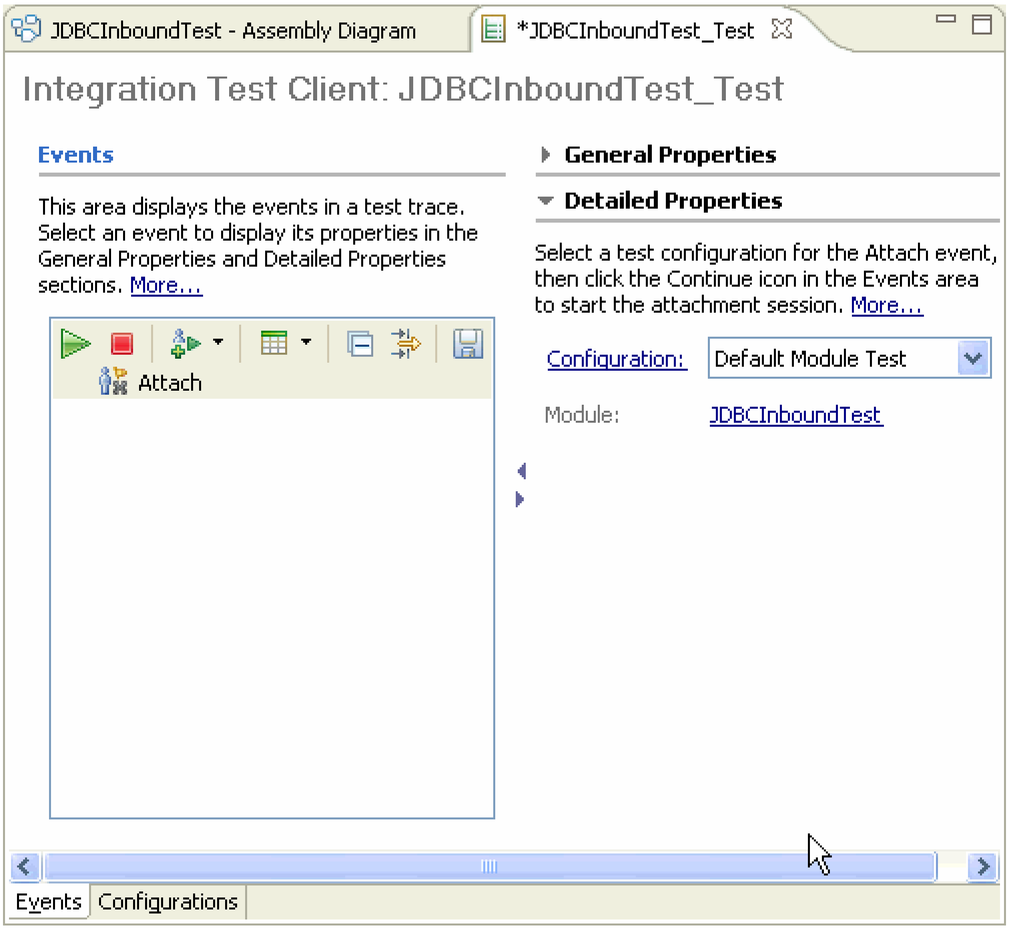
☑ If server is started, publish changes immediately

Test the assembled adapter application

Test the assembled adapter application using the IBM Integration Designer integration test client.

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


Business Integration
<all resources>
CWYBC_JDBC
JDBCInboundTest
Assembly Diagram
JDBC
Depende
Integrat
Data
Samp
Samp
Interfac
JDBC
Transfor
New
Open
Open With
Show Files
Copy
Paste
Delete
Import...
Export...
Refresh
Add to Asset Repository...
Export as Image...
Generate Documentation...
Test
Validate
Team
Compare With
Replace With
Test Module
Attach
JDBCInboundTest - Assembly Diagram
Palette
Favorites
Components
Untyped Component
Human Task
Java
Mediation Flow
Process
Rule Group
State Machine
Import
Export
References
Interface Map (De...
Selector
Outbound Imports
Inbound Exports
JDBCInboundInterface
Component1
Task Flows
Build Activities
Properties
Problems
Se
AJAX Test Server [Stopped]
IBM Process Server v7.5 at 9.119.154.112 [Profiling, Publishing.



Integration Test Client: JDBCInboundTest_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

Select a test configuration for the Attach event, then click the Continue icon in the Events area to start the attachment session. More...

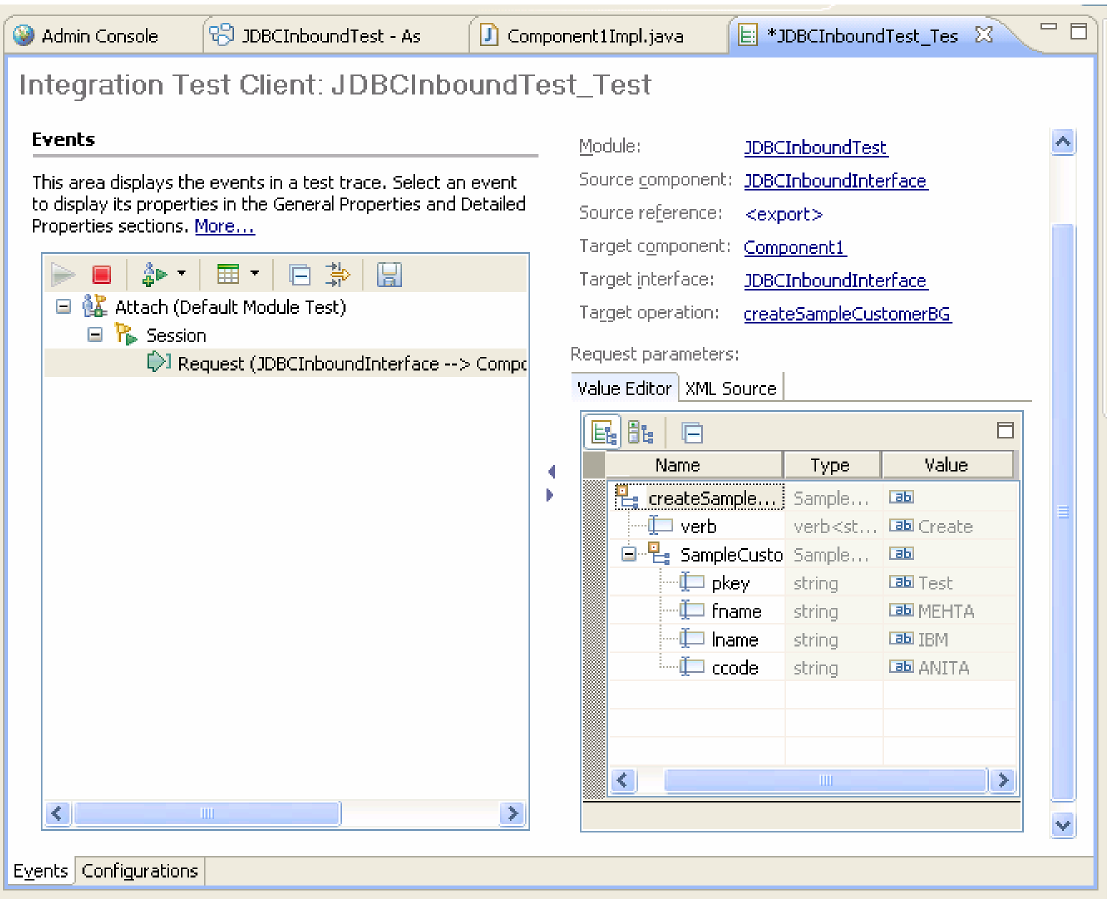
Configuration: Default Module Test

Module: JDBCInboundTest

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:



Integration Test Client: JDBCInboundTest_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...

- Attach (Default Module Test)
  - Session
    - Request (JDBCInboundInterface --> Compo

Module: JDBCInboundTest
Source component: JDBCInboundInterface
Source reference: <export>
Target component: Component1
Target interface: JDBCInboundInterface
Target operation: createSampleCustomerBG

Request parameters:

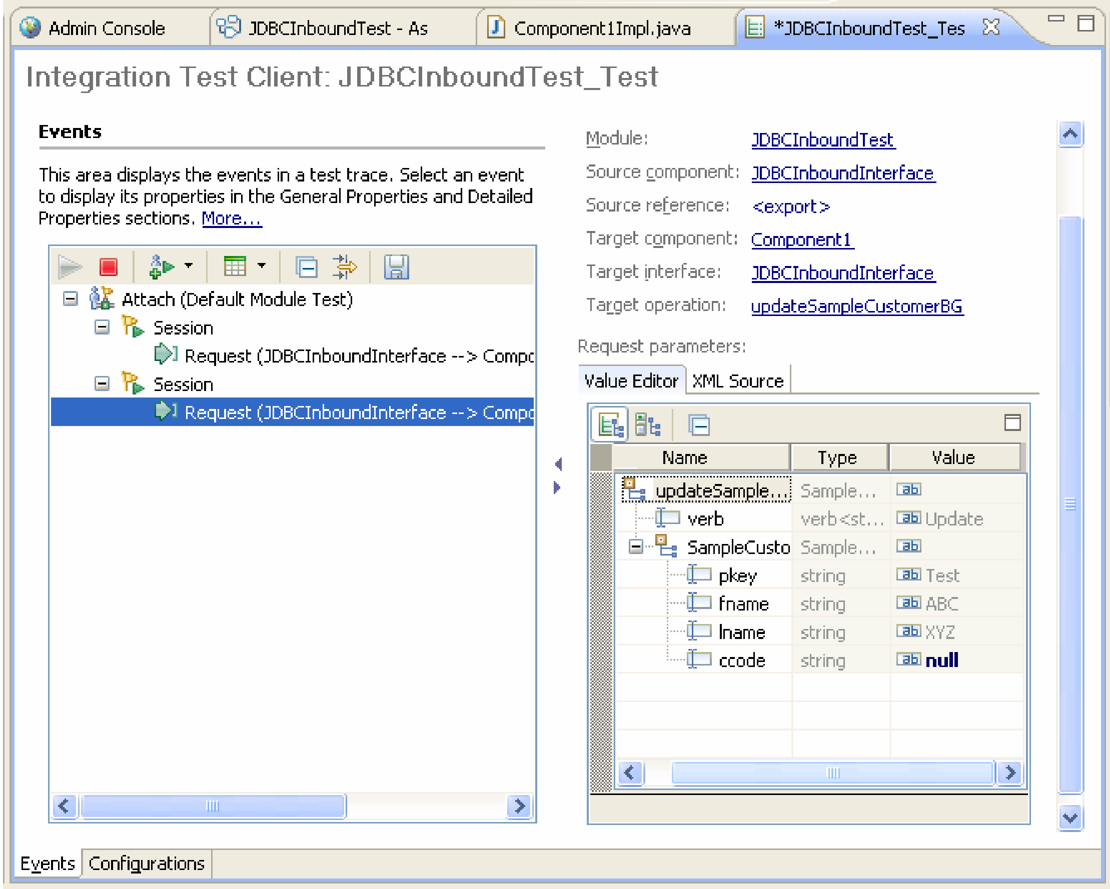
Value Editor | XML Source

Name	Type	Value
createSample...	Sample...	
verb	verb<st...	Create
SampleCusto	Sample...	
pkey	string	Test
fname	string	MEHTA
lname	string	IBM
ccode	string	ANITA

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:



Integration Test Client: JDBCInboundTest_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...

- Attach (Default Module Test)
  - Session
    - Request (JDBCInboundInterface --> Compo
  - Session
    - Request (JDBCInboundInterface --> Compo

Module: JDBCInboundTest
Source component: JDBCInboundInterface
Source reference: <export>
Target component: Component1
Target interface: JDBCInboundInterface
Target operation: updateSampleCustomerBG

Request parameters:

Name	Type	Value
updateSample...	Sample...	
verb	verb<st...	Update
SampleCusto	Sample...	
pkey	string	Test
fname	string	ABC
lname	string	XYZ
ccode	string	**null**

Clear the sample content

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

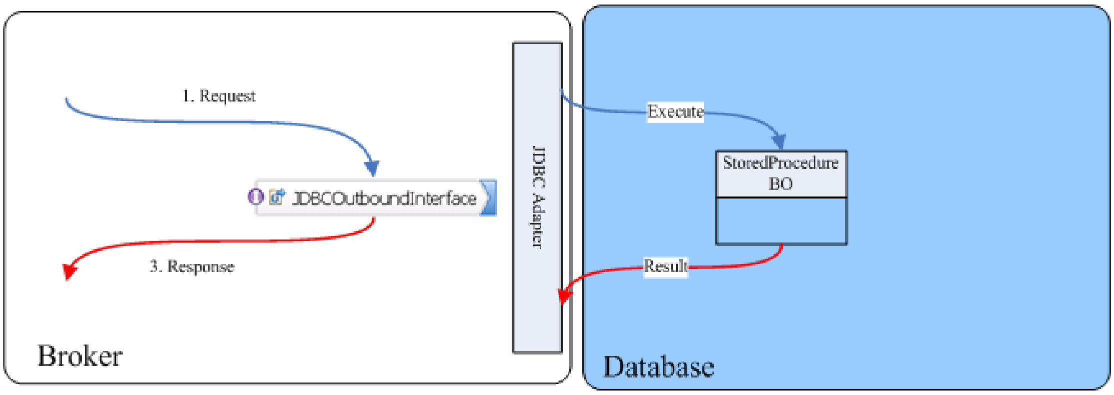
Chapter 7. Tutorial 6: Executing a business object created from a stored procedure (DB2)

This scenario demonstrates how WebSphere Adapter for JDBC 7.5.0.0 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 IBM Integration Designer.

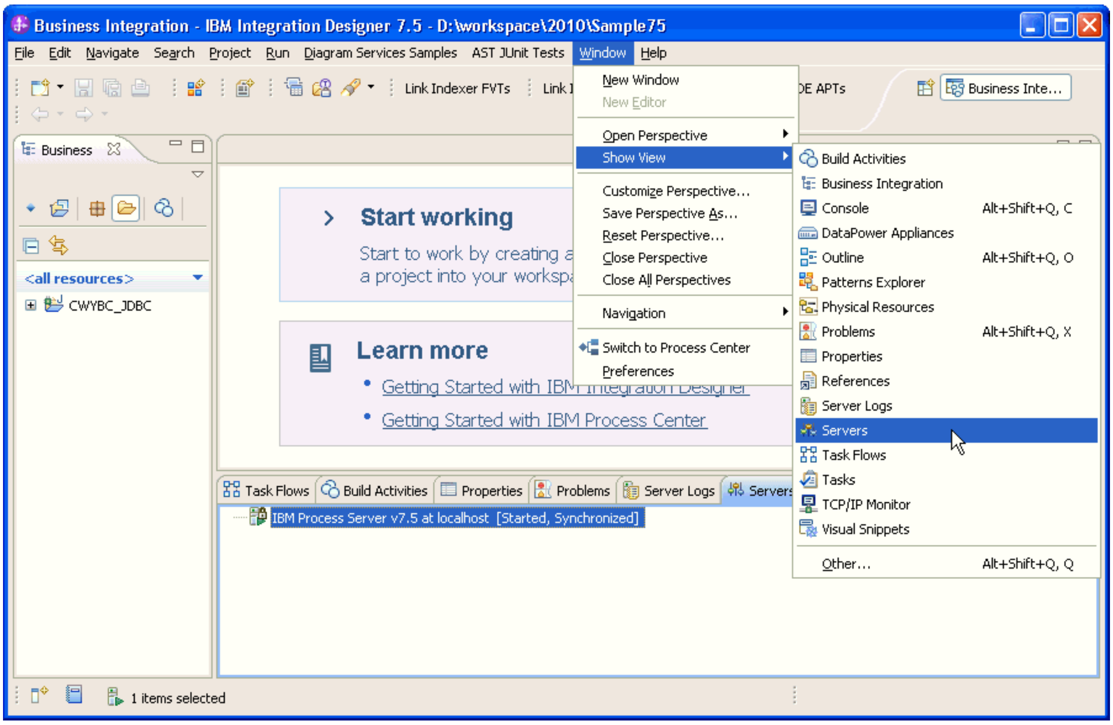
```
CREATE PROCEDURE CustAddrSP (  )
   SPECIFIC CustAddrSP
   DYNAMIC RESULT SETS 1
--------------------------------------------------
----------------------
-- SQL Stored Procedure
--------------------------------------------------
----------------------
P1: BEGIN
   -- Declare cursor
   DECLARE cursor1 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

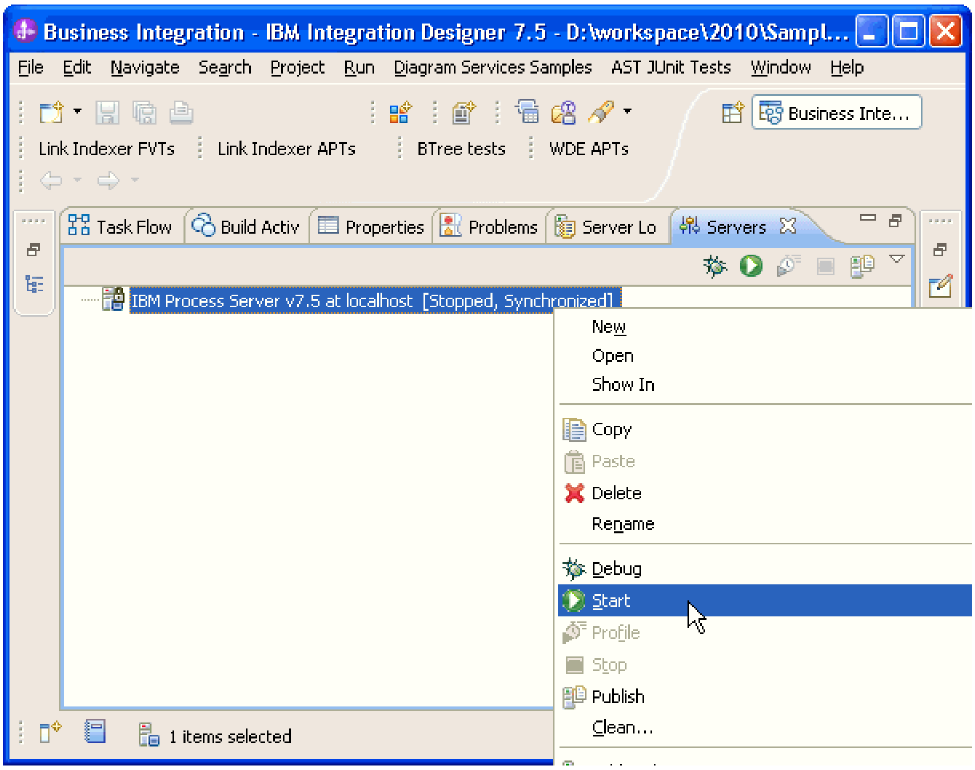
The authentication alias needs to be set because the adapter uses the username and password set in the authentication alias to connect to the database. This authentication alias will be used later when generating the artifacts for the module.

Here are the steps to set the authentication alias in IBM Process Server administration console.

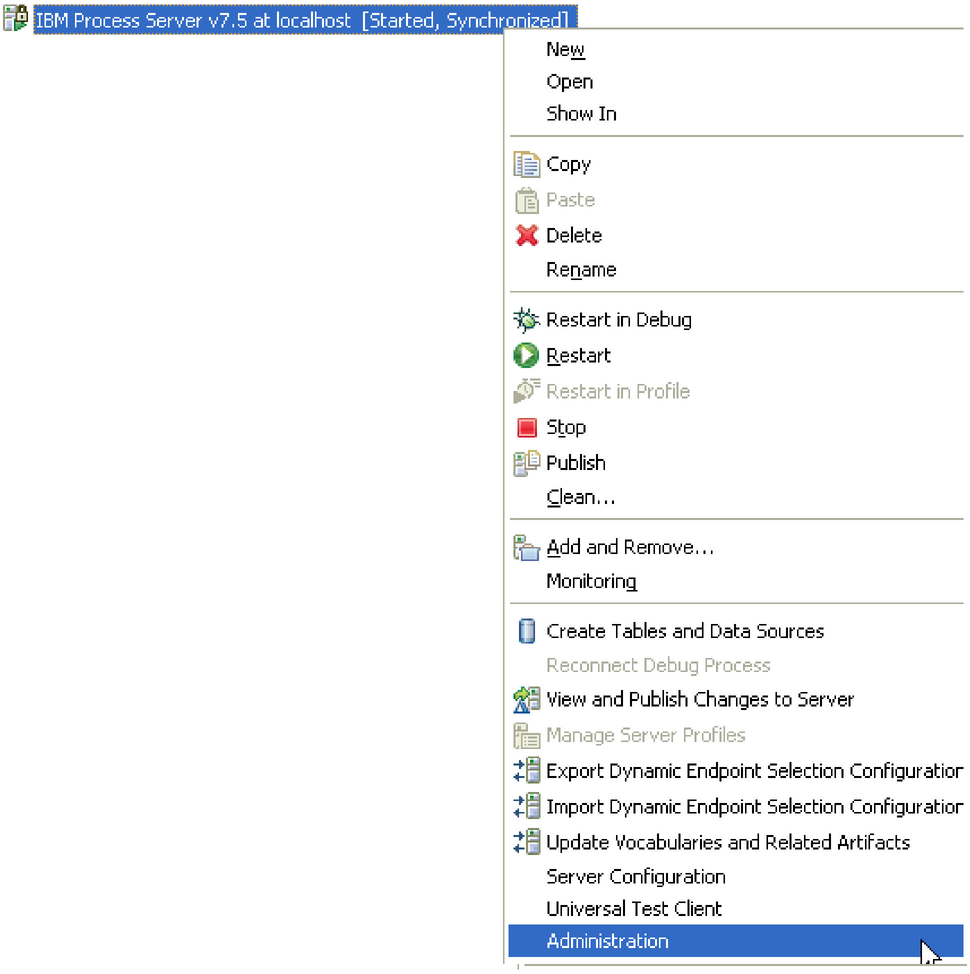
1. In IBM Integration Designer, switch to the **Servers** view by selecting **Windows > Show View > Servers**.



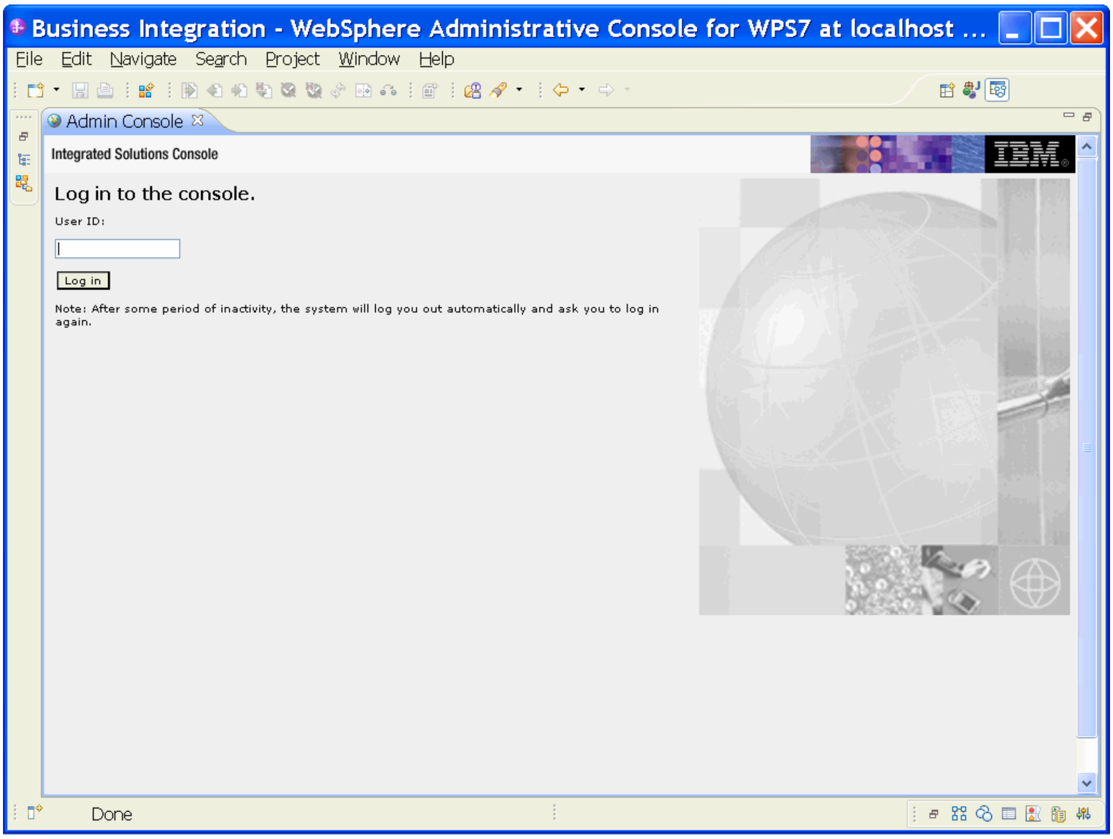
2. In the **Servers** view, right-click the server that you want to start and select **Start**.


Business Integration - IBM Integration Designer 7.5 - D:\workspace\2010\Sampl...
File Edit Navigate Search Project Run Diagram Services Samples AST JUnit Tests Window Help
Business Inte...
Link Indexer FVTs Link Indexer APTs BTree tests WDE APTs
Task Flow Build Activ Properties Problems Server Lo Servers
IBM Process Server v7.5 at localhost [Stopped, Synchronized]
New
Open
Show In
Copy
Paste
Delete
Rename
Debug
Start
Profile
Stop
Publish
Clean...
1 items selected

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


IBM Process Server v7.5 at localhost [Started, Synchronized]
New
Open
Show In
Copy
Paste
Delete
Rename
Restart in Debug
Restart
Restart in Profile
Stop
Publish
Clean...
Add and Remove...
Monitoring
Create Tables and Data Sources
Reconnect Debug Process
View and Publish Changes to Server
Manage Server Profiles
Export Dynamic Endpoint Selection Configuratior
Import Dynamic Endpoint Selection Configuratior
Update Vocabularies and Related Artifacts
Server Configuration
Universal Test Client
Administration

4. Log on to the administrative console.



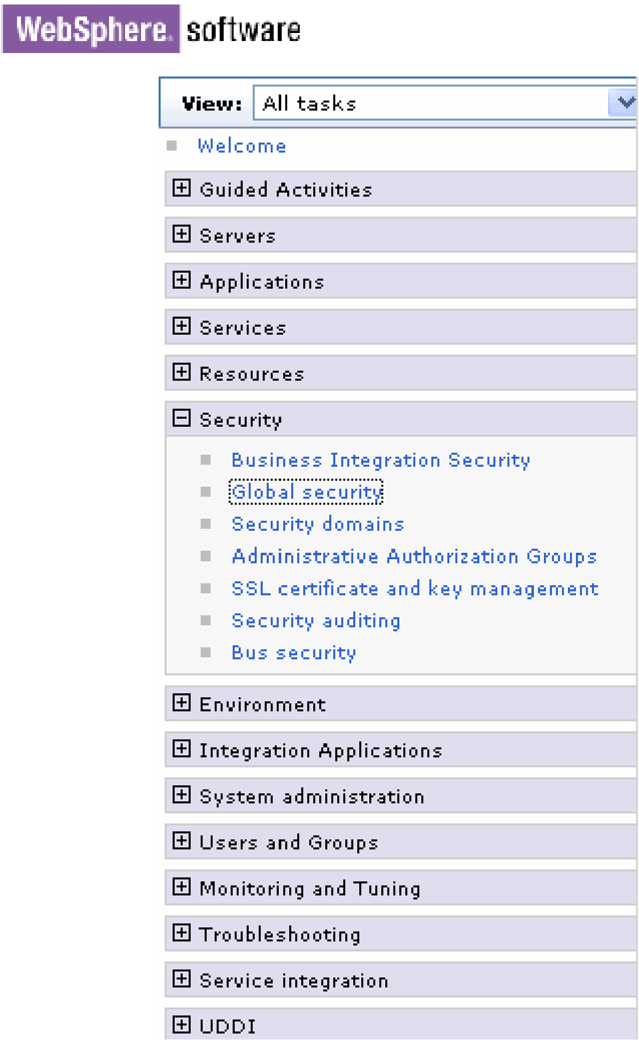
Business Integration - WebSphere Administrative Console for WPS7 at localhost ...
File Edit Navigate Search Project Window Help
Admin Console
Integrated Solutions Console
IBM
Log in
Done

Log in to the console.

User ID:

Note: After some period of inactivity, the system will log you out automatically and ask you to log in again.

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

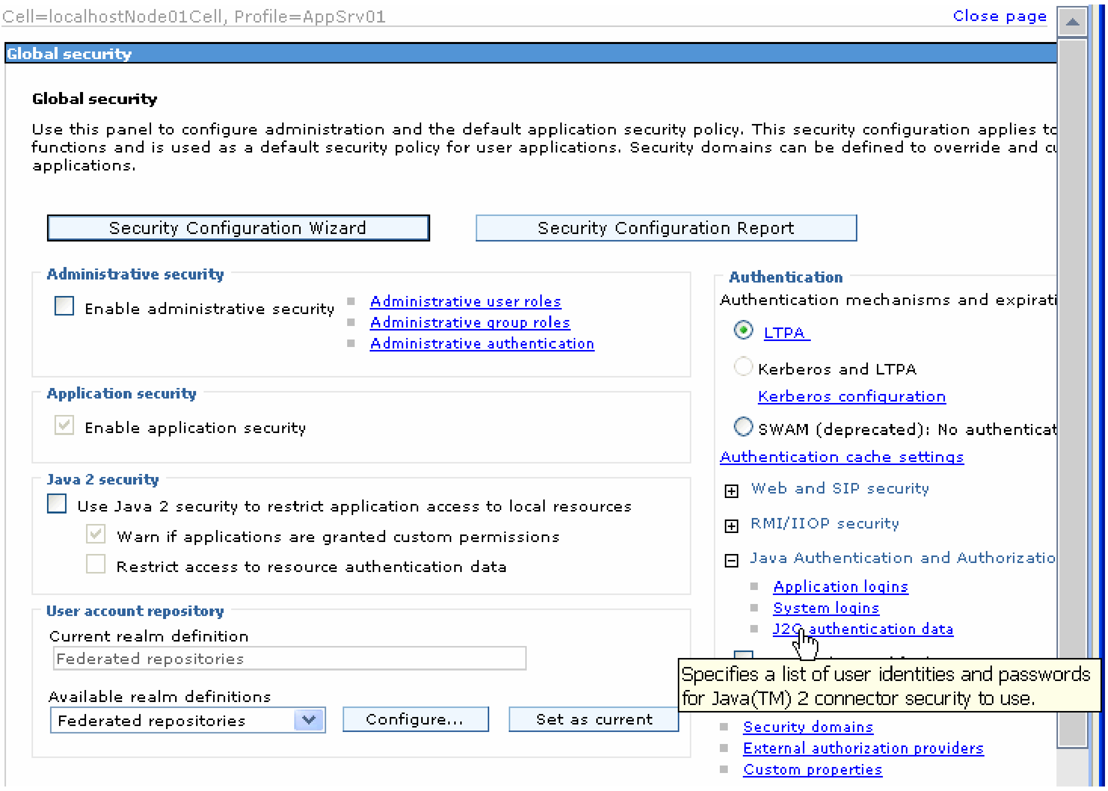


WebSphere software

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 cu 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 expirati

● LTPA

○ Kerberos and LTPA

Kerberos configuration

○ SWAM (deprecated): No authenticat

Authentication cache settings

⊞ Web and SIP security

⊞ RMI/IIOP security

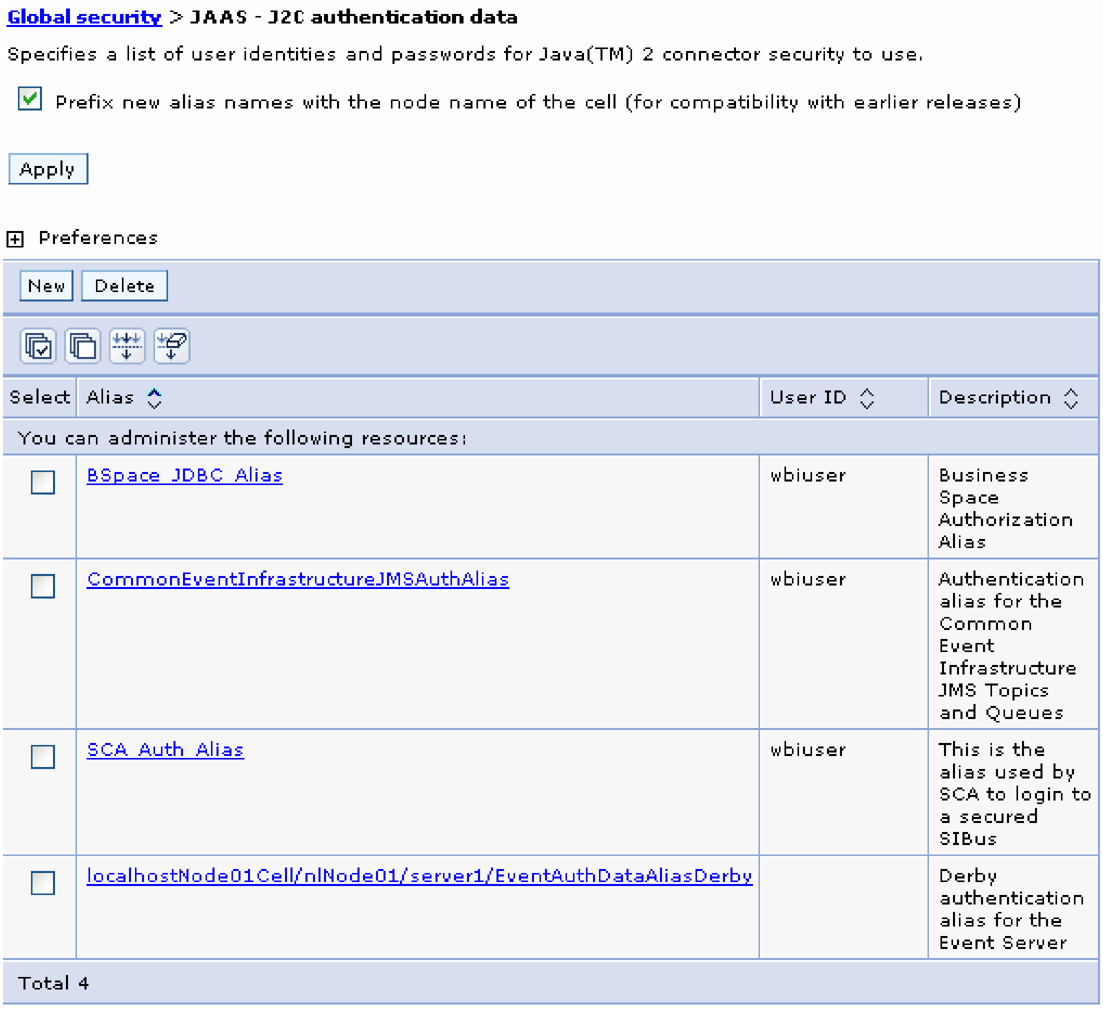
⊟ Java Authentication and Authorizatio

- 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.



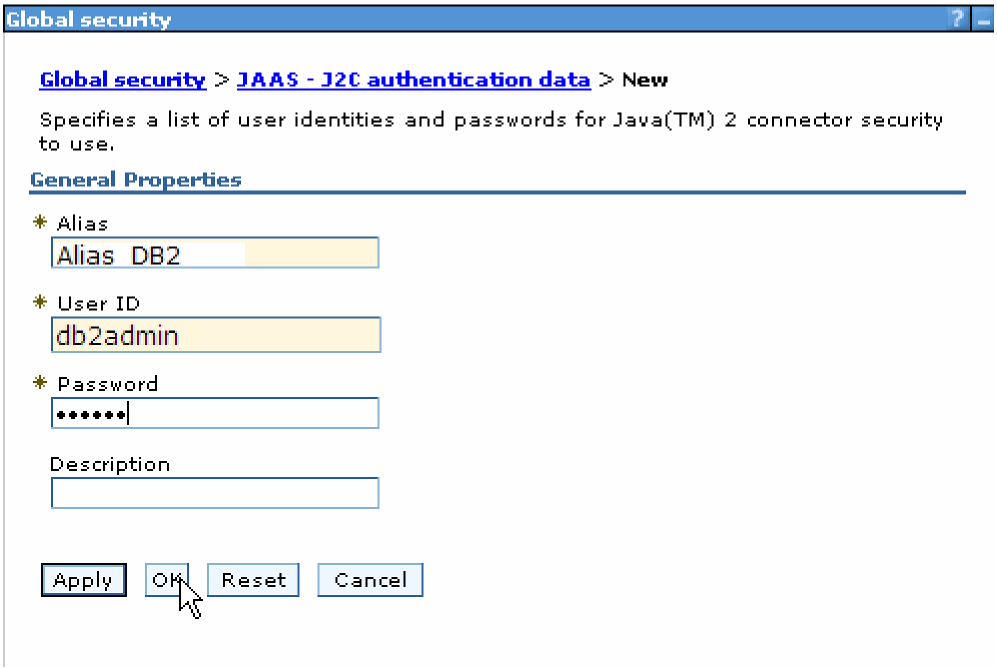
Global security > JAAS - J2C authentication data

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)

Select	Alias	User ID	Description
You can administer the following resources:			
□	BSpace_JDBC_Alias	wbiuser	Business Space Authorization Alias
□	CommonEventInfrastructureJMSAuthAlias	wbiuser	Authentication alias for the Common Event Infrastructure JMS Topics and Queues
□	SCA_Auth_Alias	wbiuser	This is the alias used by SCA to login to a secured SIBus
□	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

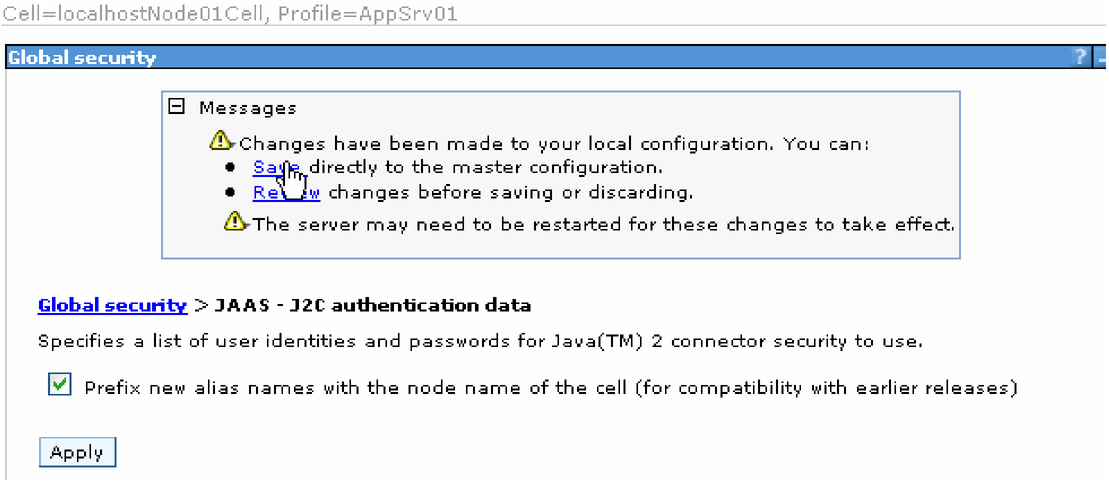
* Alias
Alias DB2

* User ID
db2admin

* Password
••••••

Description

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

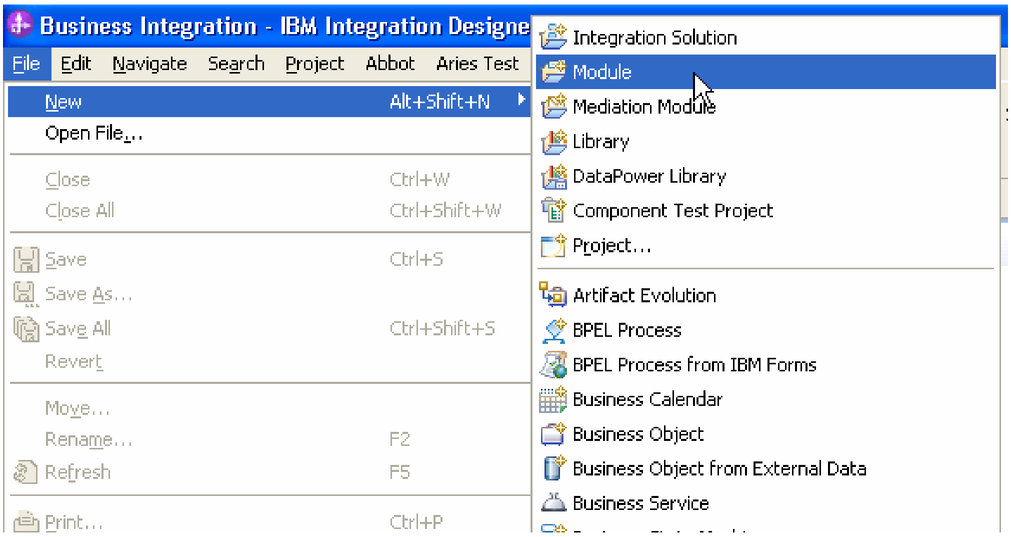
Specifies a list of user identities and passwords for Java(TM) 2 connector security to use.

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

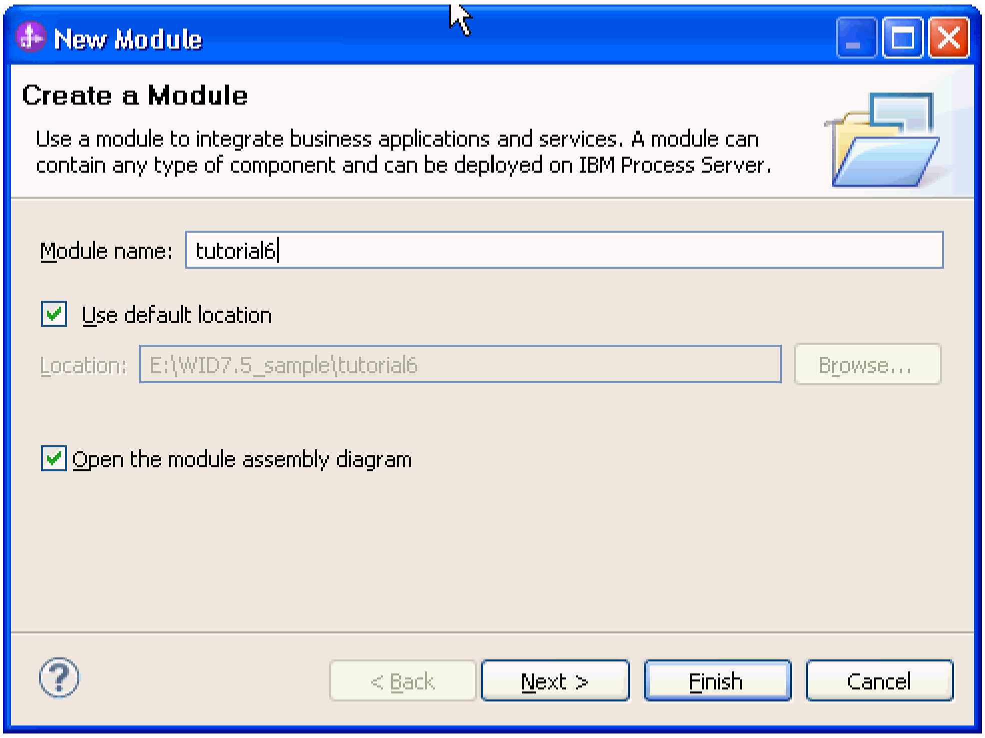
Note: You have created an authentication alias that will be used when you configure the adapter properties. Re-start the server for the changes to take effect.

Configure the adapter for outbound processing

1. Switch to the Business Integration perspective in IBM Integration Designer.
2. Select **File->New->Module** to create a module project.



3. Specify the module name as **Tutorial6**, and click **Finish**.



Create a Module

Use a module to integrate business applications and services. A module can contain any type of component and can be deployed on IBM Process Server.

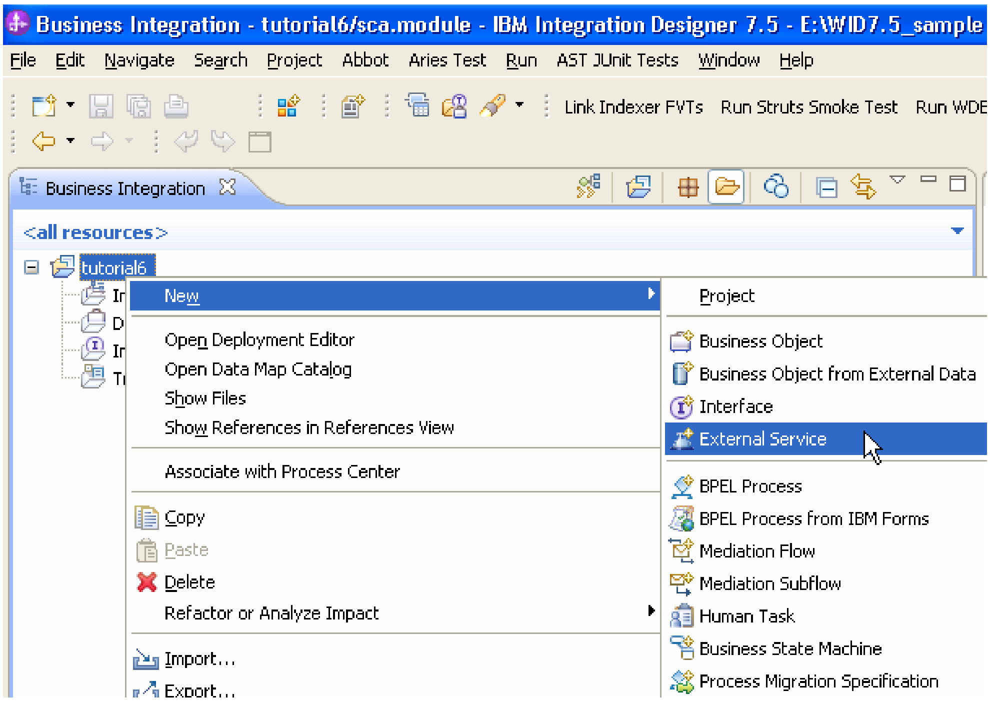
Module name: tutorial6

☑ Use default location

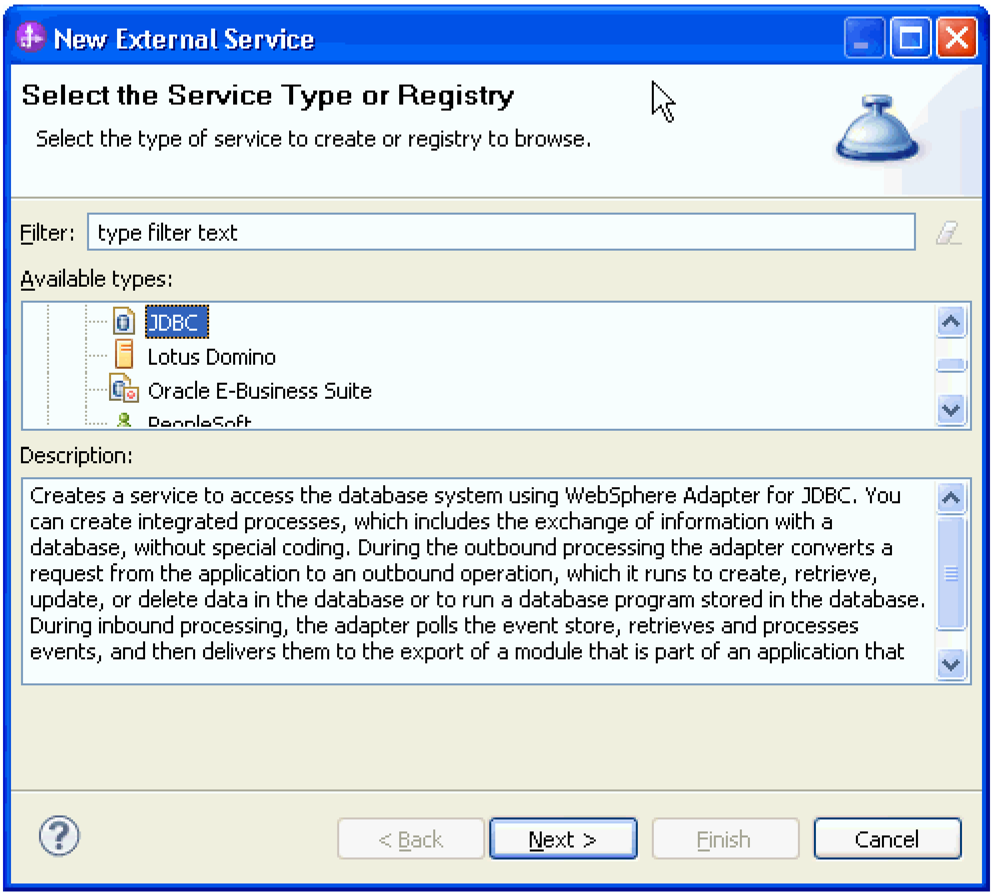
Location: E:\WID7.5_sample\tutorial6

☑ Open the module assembly diagram

4. Expand Tutorial6 and select displayed. Right-click and select **New->External Service.**

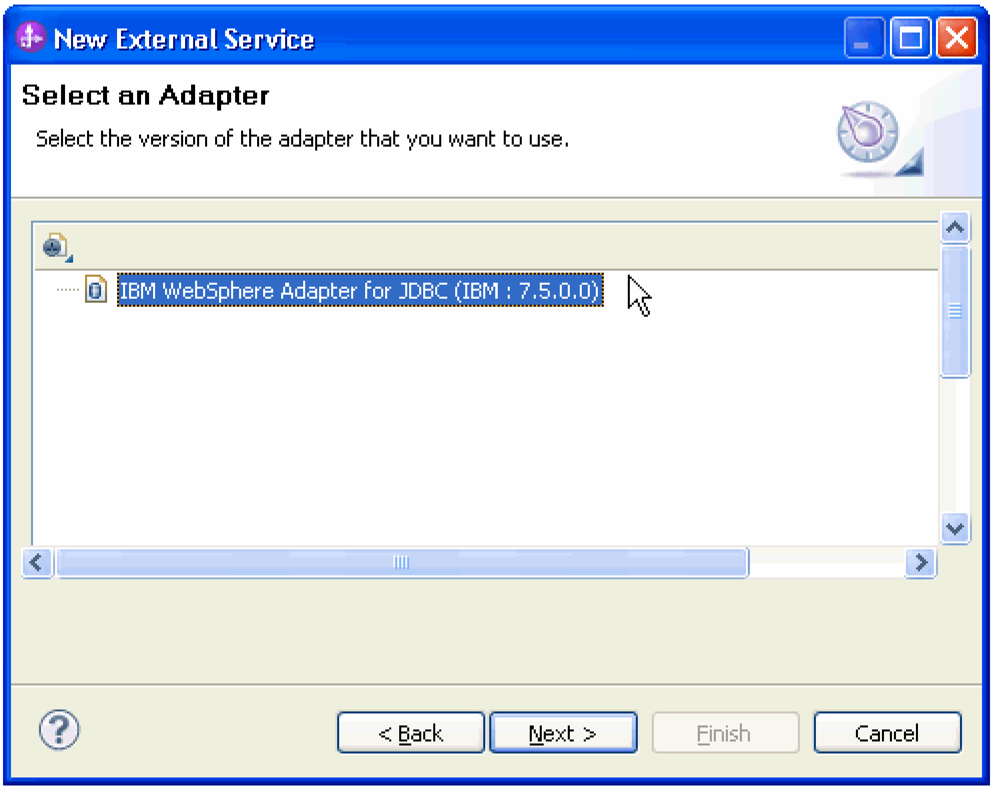

Business Integration - tutorial6/sca.module - IBM Integration Designer 7.5 - E:\WID7.5_sample
File Edit Navigate Search Project Abbot Aries Test Run AST JUnit Tests Window Help
Link Indexer FVTs Run Struts Smoke Test Run WDE
Business Integration
<all resources>
tutorial6
New
Open Deployment Editor
Open Data Map Catalog
Show Files
Show References in References View
Associate with Process Center
Copy
Paste
Delete
Refactor or Analyze Impact
Import...
Export...
Project
Business Object
Business Object from External Data
Interface
External Service
BPEL Process
BPEL Process from IBM Forms
Mediation Flow
Mediation Subflow
Human Task
Business State Machine
Process Migration Specification

5. Select JDBC, and click **Next**.



Creates a service to access the database system using WebSphere Adapter for JDBC. You can create integrated processes, which includes the exchange of information with a database, without special coding. During the outbound processing the adapter converts a request from the application to an outbound operation, which it runs to create, retrieve, update, or delete data in the database or to run a database program stored in the database. During inbound processing, the adapter polls the event store, retrieves and processes events, and then delivers them to the export of a module that is part of an application that

6. Select **IBM WebSphere Adapter for JDBC (IBM: 7.5.0.0).** Click **Next**.



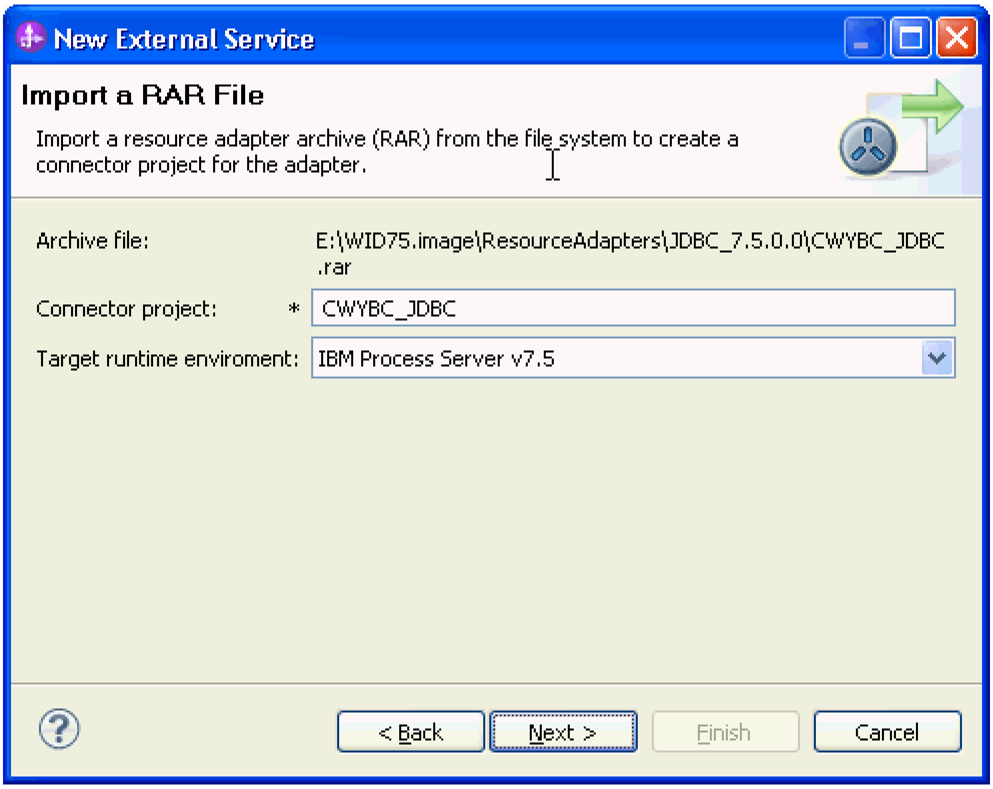
New External Service

Select an Adapter

Select the version of the adapter that you want to use.

IBM WebSphere Adapter for JDBC (IBM : 7.5.0.0)

7. In the **Target Runtime environment** field, select the appropriate runtime and click **Next**.



New External Service

Import a RAR File

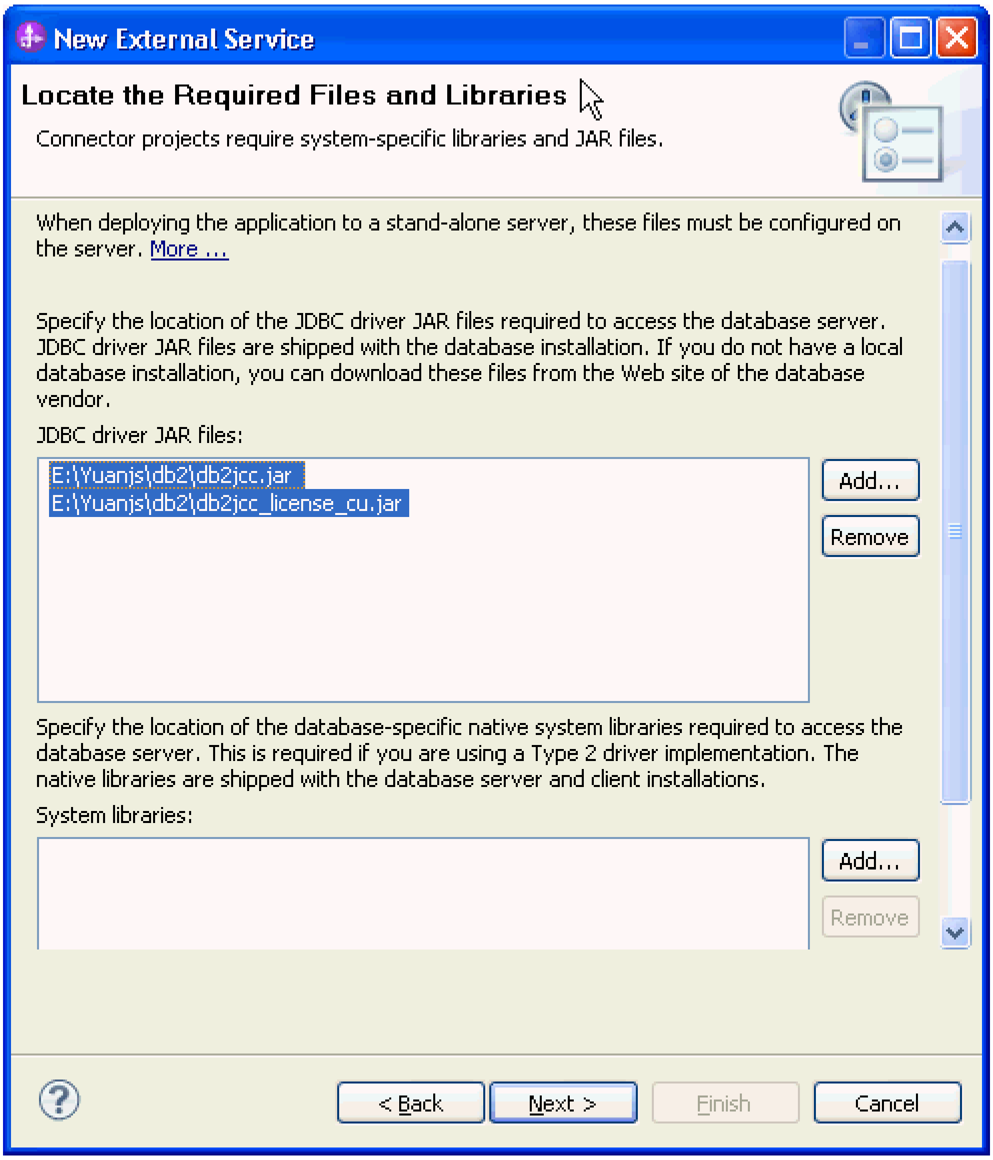
Import a resource adapter archive (RAR) from the file system to create a connector project for the adapter.

Archive file: E:\WID75.image\ResourceAdapters\JDBC_7.5.0.0\CWYBC_JDBC.rar

Connector project: * CWYBC_JDBC

Target runtime enviroment: IBM Process Server v7.5

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**.



Locate the Required Files and Libraries

Connector projects require system-specific libraries and JAR files.

When deploying the application to a stand-alone server, these files must be configured on the server. More ...

Specify the location of the JDBC driver JAR files required to access the database server. JDBC driver JAR files are shipped with the database installation. If you do not have a local database installation, you can download these files from the Web site of the database vendor.

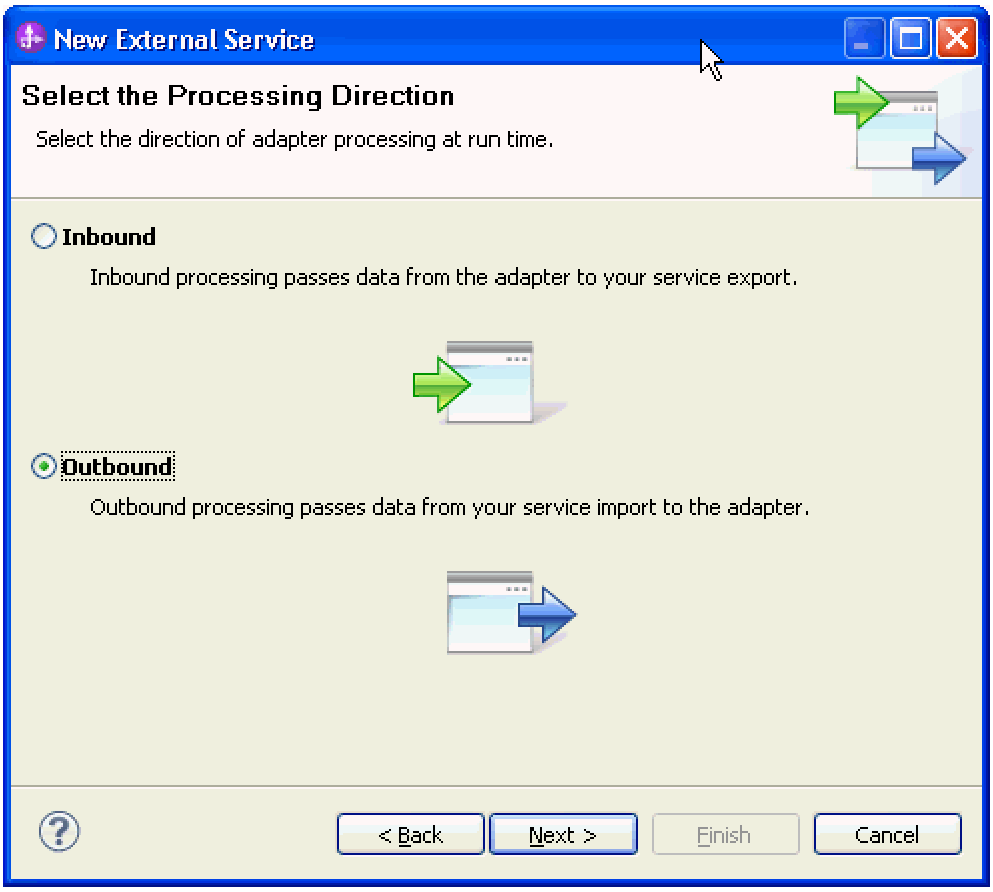
JDBC driver JAR files:

```
E:\Yuanjs\db2\db2jcc.jar
E:\Yuanjs\db2\db2jcc_license_cu.jar
```

Specify the location of the database-specific native system libraries required to access the database server. This is required if you are using a Type 2 driver implementation. The native libraries are shipped with the database server and client installations.

System libraries:

9. Select **Outbound** and click **Next**.



Select the Processing Direction

Select the direction of adapter processing at run time.

○ Inbound

Inbound processing passes data from the adapter to your service export.

● Outbound

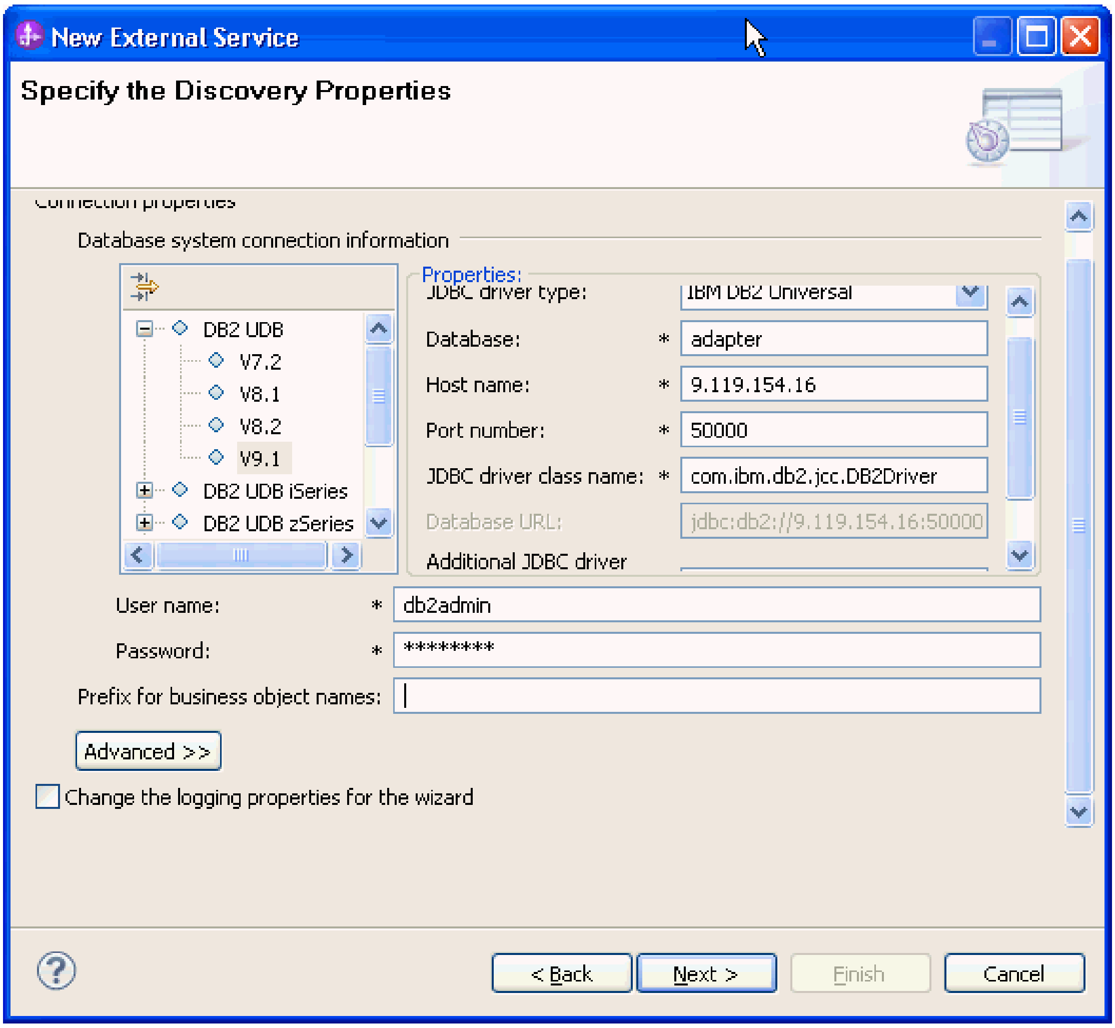
Outbound processing passes data from your service import to the adapter.

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**.



New External Service

Specify the Discovery Properties

Connection properties

Database system connection information

Properties:

JDBC driver type: IBM DB2 Universal

Database: * adapter

Host name: * 9.119.154.16

Port number: * 50000

JDBC driver class name: * com.ibm.db2.jcc.DB2Driver

Database URL: jdbc:db2://9.119.154.16:50000

Additional JDBC driver

User name: * db2admin

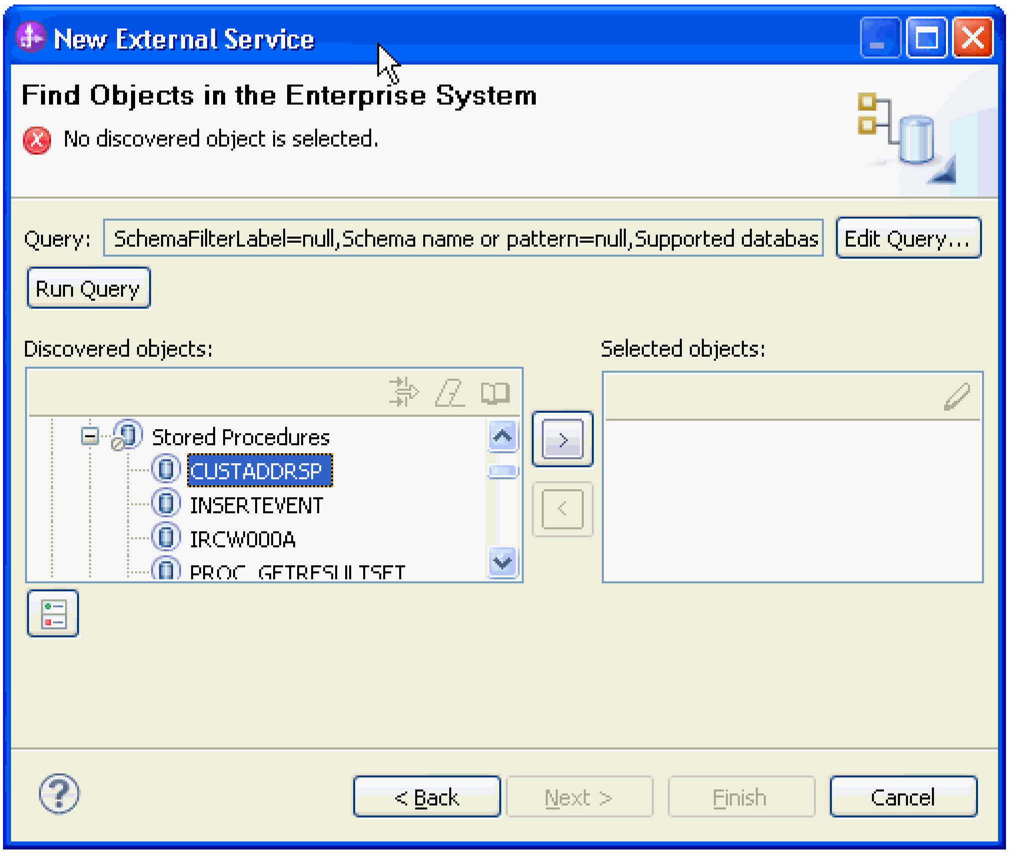
Password: * ********

Prefix for business object names:

□ Change the logging properties for the wizard

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 >.

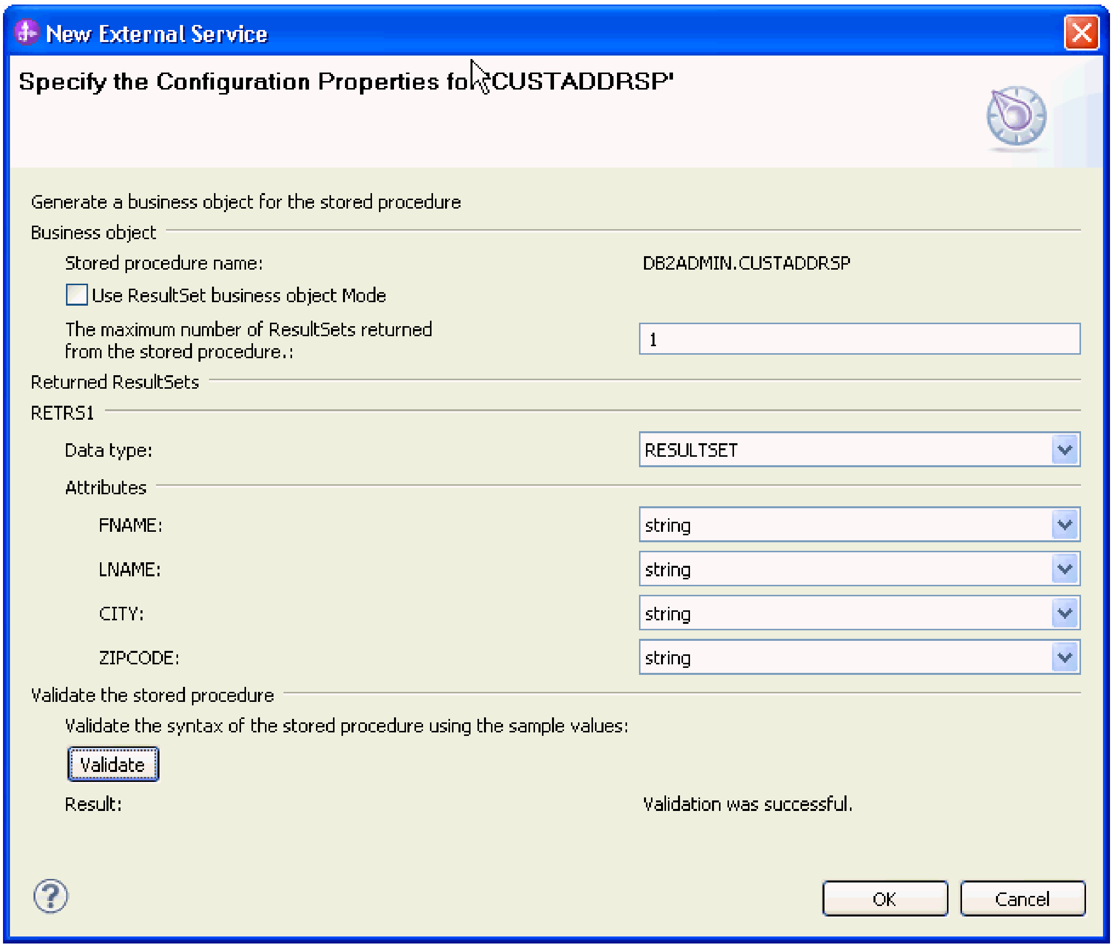


Find Objects in the Enterprise System

No discovered object is selected.

Query: SchemaFilterLabel=null,Schema name or pattern=null,Supported databas

3. In the **The maximum number of ResultSets returned from the stored procedure** field, enter 1. Click **Validate**. After successful validation, click **OK**.



New External Service

Specify the Configuration Properties for 'CUSTADDRSP'

Generate a business object for the stored procedure

Business object

Stored procedure name: DB2ADMIN.CUSTADDRSP

□ Use ResultSet business object Mode

The maximum number of ResultSets returned from the stored procedure.: 1

Returned ResultSets

RETRS1

Data type: RESULTSET

Attributes

FNAME: string

LNAME: string

CITY: string

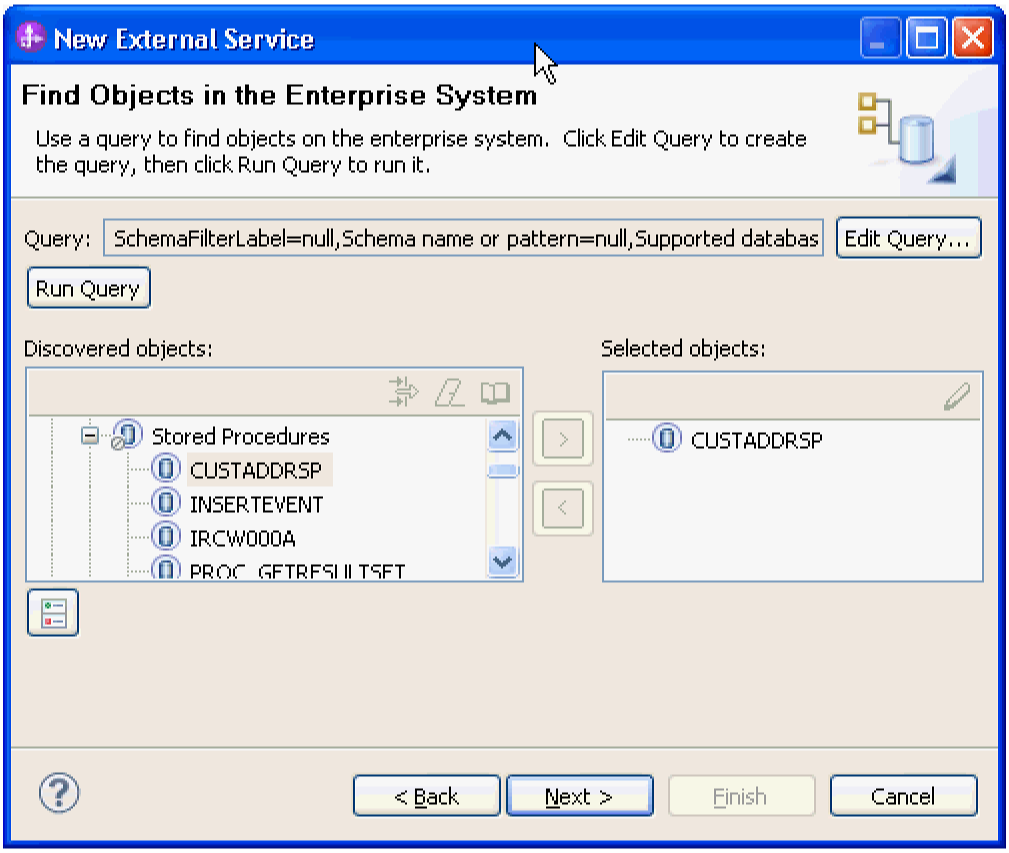
ZIPCODE: string

Validate the stored procedure

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

Result: Validation was successful.

4. Click **Next**.



New External Service

Find Objects in the Enterprise System

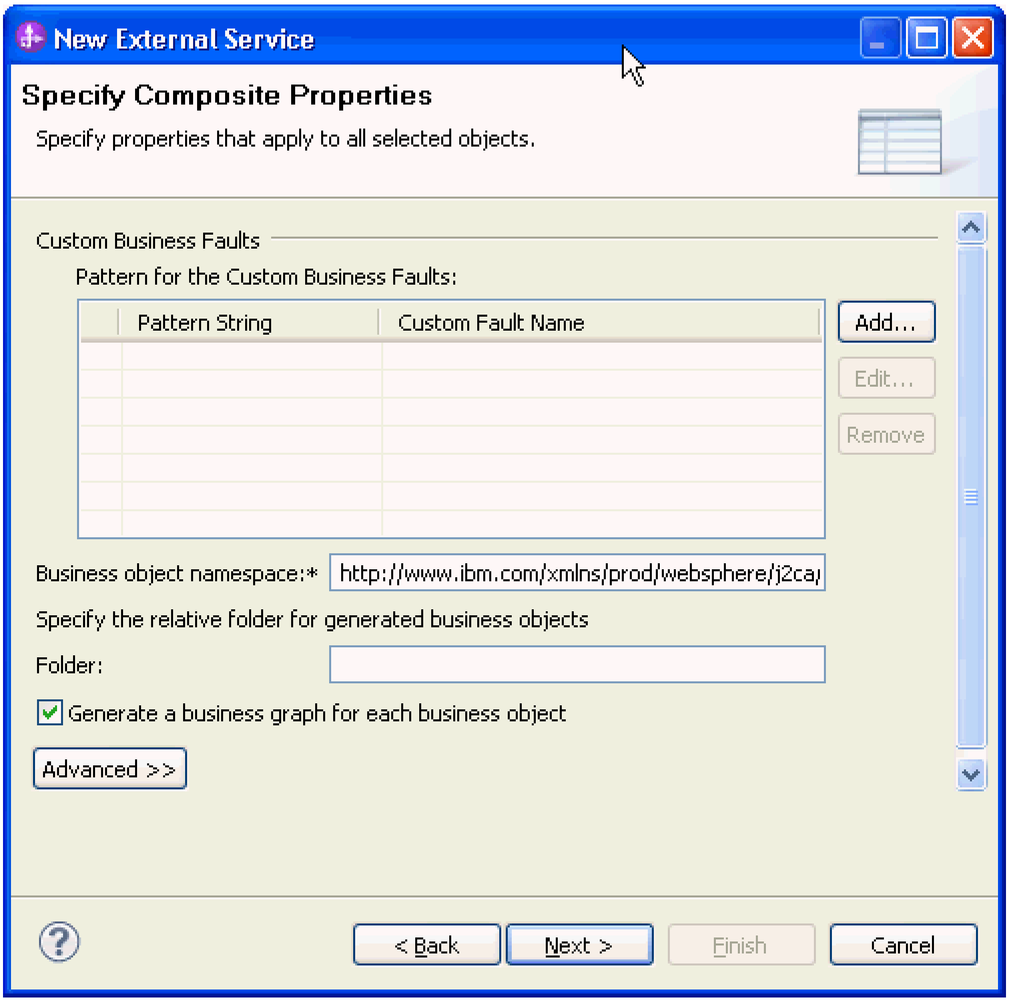
Use a query to find objects on the enterprise system. Click Edit Query to create the query, then click Run Query to run it.

Query: SchemaFilterLabel=null,Schema name or pattern=null,Supported databas

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**.



Business object namespace:* http://www.ibm.com/xmlns/prod/websphere/j2ca

Specify the relative folder for generated business objects

Folder:

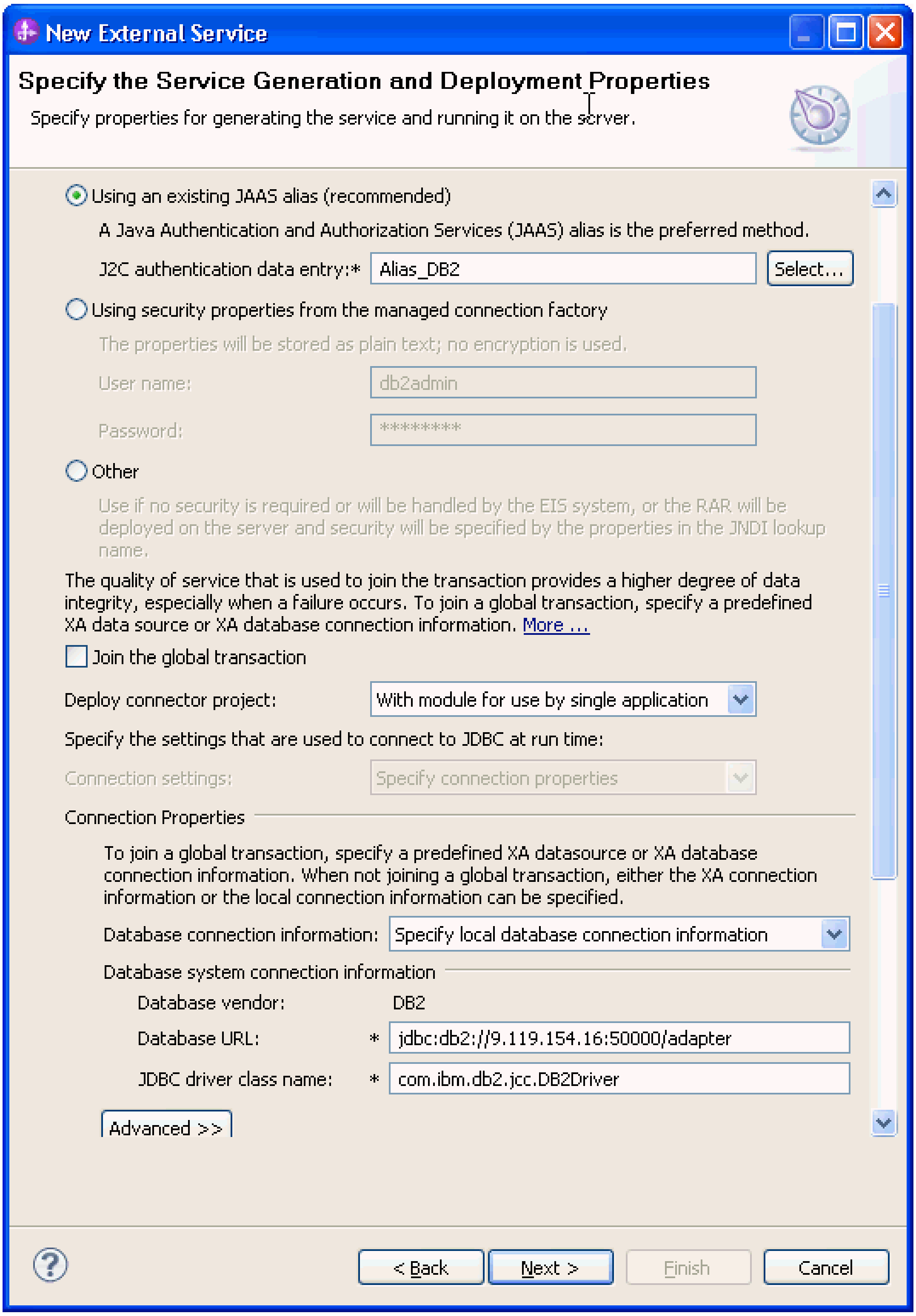
☑ Generate a business graph for each business object

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) disable the **Join the global transaction** check box.

   c) Select **Specify local database connection information** from the **Database connection information** list and click **Next**.



Specify the Service Generation and Deployment Properties

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

● Using an existing JAAS alias (recommended)

A Java Authentication and Authorization Services (JAAS) alias is the preferred method.

J2C authentication data entry:* Alias_DB2 Select...

○ Using security properties from the managed connection factory

The properties will be stored as plain text; no encryption is used.

User name: db2admin

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 join a global transaction, specify a predefined XA data source or XA database connection information. More ...

□ Join the global transaction

Deploy connector project: With module for use by single application

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

Connection settings: Specify connection properties

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: Specify local database connection information

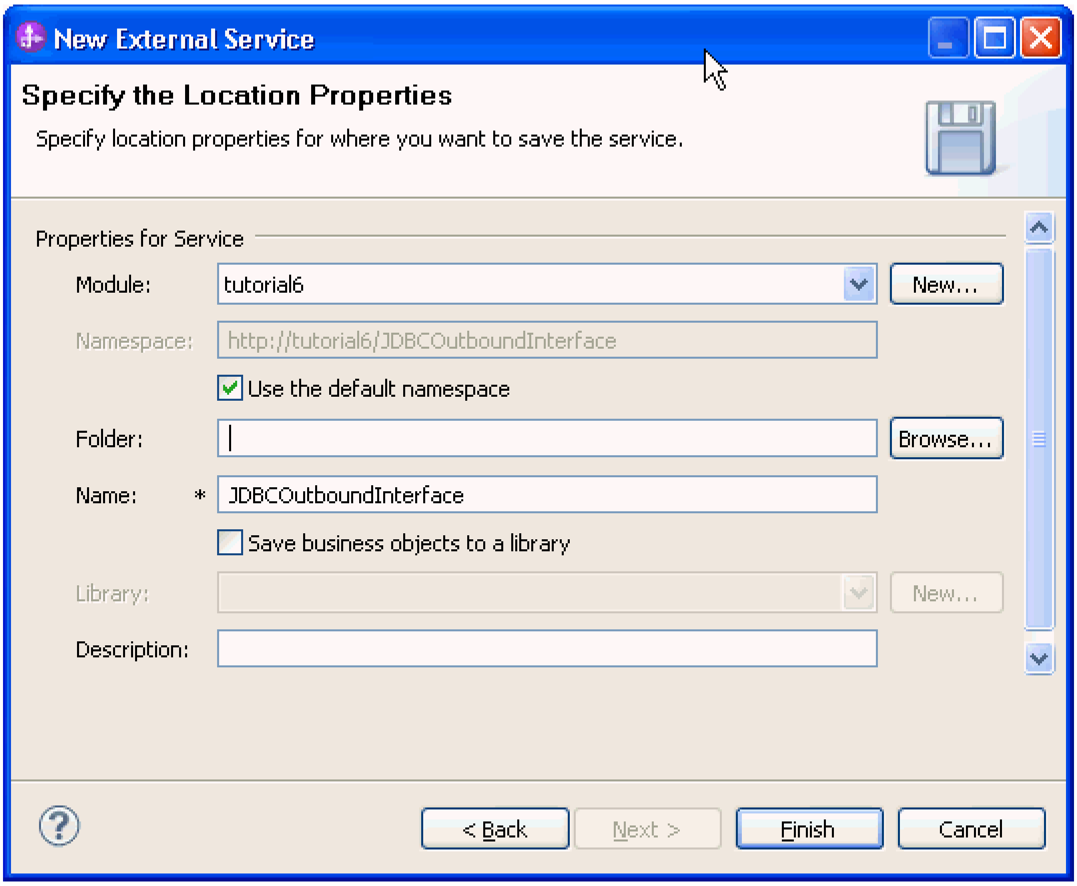
Database system connection information

Database vendor: DB2

Database URL: * jdbc:db2://9.119.154.16:50000/adapter

JDBC driver class name: * com.ibm.db2.jcc.DB2Driver

3. In the Specify the Location Properties window, click **Finish** to complete the service creation.



New External Service

Specify the Location Properties

Specify location properties for where you want to save the service.

Properties for Service

Module: tutorial6

Namespace: http://tutorial6/JDBCOutboundInterface

☑ Use the default namespace

Folder:

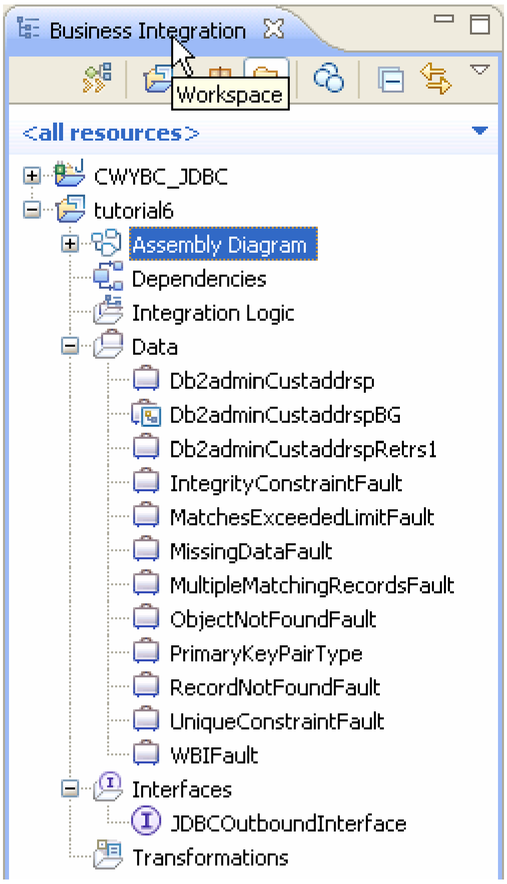
Name: * JDBCOutboundInterface

☐ Save business objects to a library

Library:

Description:

4. Verify the results.



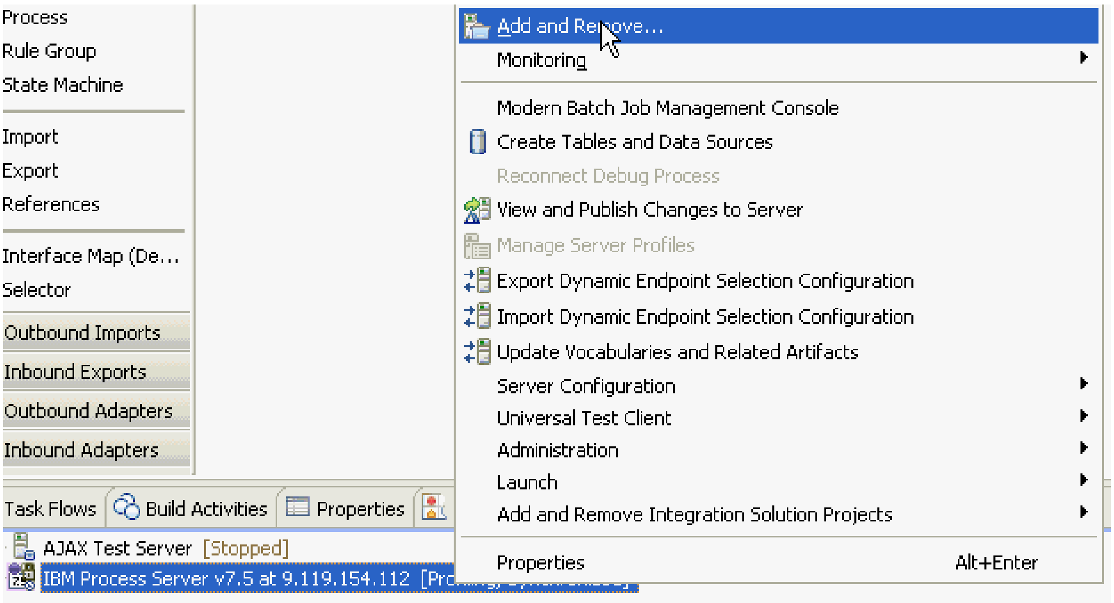
- CWYBC_JDBC
- tutorial6
  - Assembly Diagram
  - Dependencies
  - Integration Logic
  - Data
    - Db2adminCustaddrsp
    - Db2adminCustaddrspBG
    - Db2adminCustaddrspRetrs1
    - IntegrityConstraintFault
    - MatchesExceededLimitFault
    - MissingDataFault
    - MultipleMatchingRecordsFault
    - ObjectNotFoundFault
    - PrimaryKeyPairType
    - RecordNotFoundFault
    - UniqueConstraintFault
    - WBIFault
  - Interfaces
    - JDBCOutboundInterface
  - Transformations

Deploy the module to the test environment

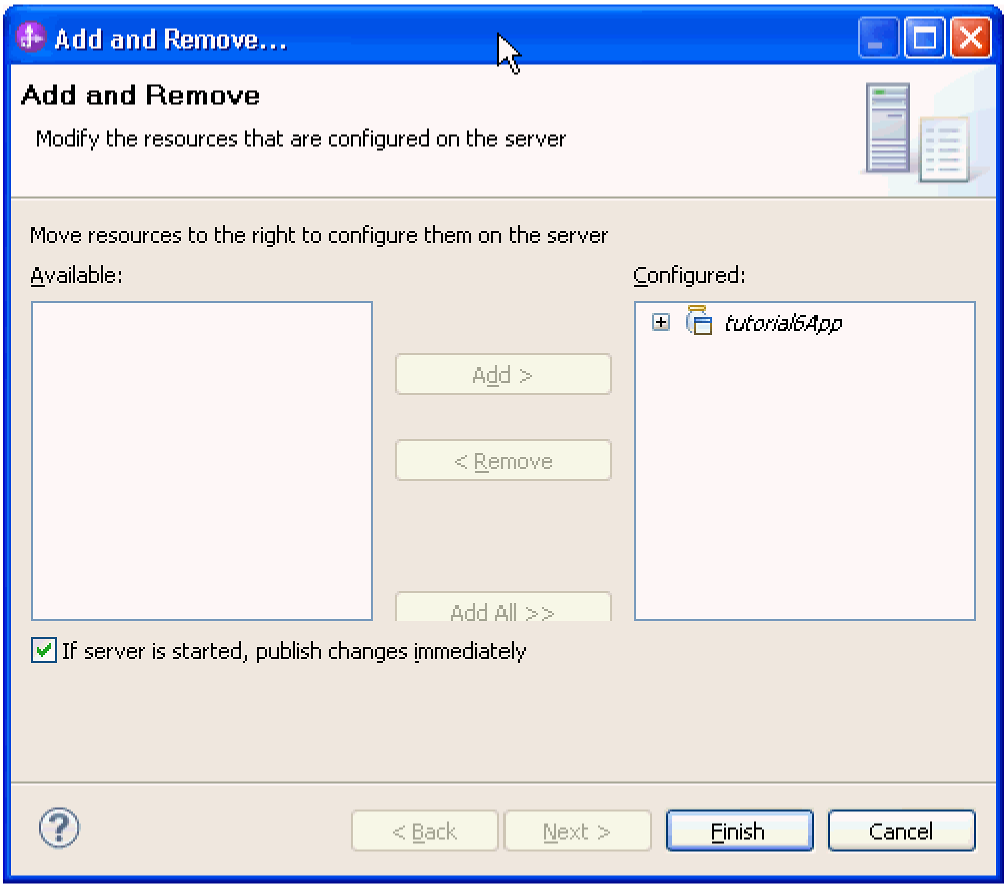
The result of running the external service wizard is an SCA module that contains an Enterprise Information System import. Install this SCA module in the IBM Integration Designer integration test client. To do this, you must add the SCA module you created earlier to the server using the **Servers** view in IBM Integration Designer.

Steps for adding the SCA module to the server:

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


Process
Rule Group
State Machine
Import
Export
References
Interface Map (De...
Selector
Outbound Imports
Inbound Exports
Outbound Adapters
Inbound Adapters
Task Flows
Build Activities
Properties
AJAX Test Server [Stopped]
IBM Process Server v7.5 at 9.119.154.112 [Pro...
Add and Remove...
Monitoring
Modern Batch Job Management Console
Create Tables and Data Sources
Reconnect Debug Process
View and Publish Changes to Server
Manage Server Profiles
Export Dynamic Endpoint Selection Configuration
Import Dynamic Endpoint Selection Configuration
Update Vocabularies and Related Artifacts
Server Configuration
Universal Test Client
Administration
Launch
Add and Remove Integration Solution Projects
Properties
Alt+Enter

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.



Add and Remove...

Add and Remove

Modify the resources that are configured on the server

Move resources to the right to configure them on the server

Available:

Configured:

tutorial6App

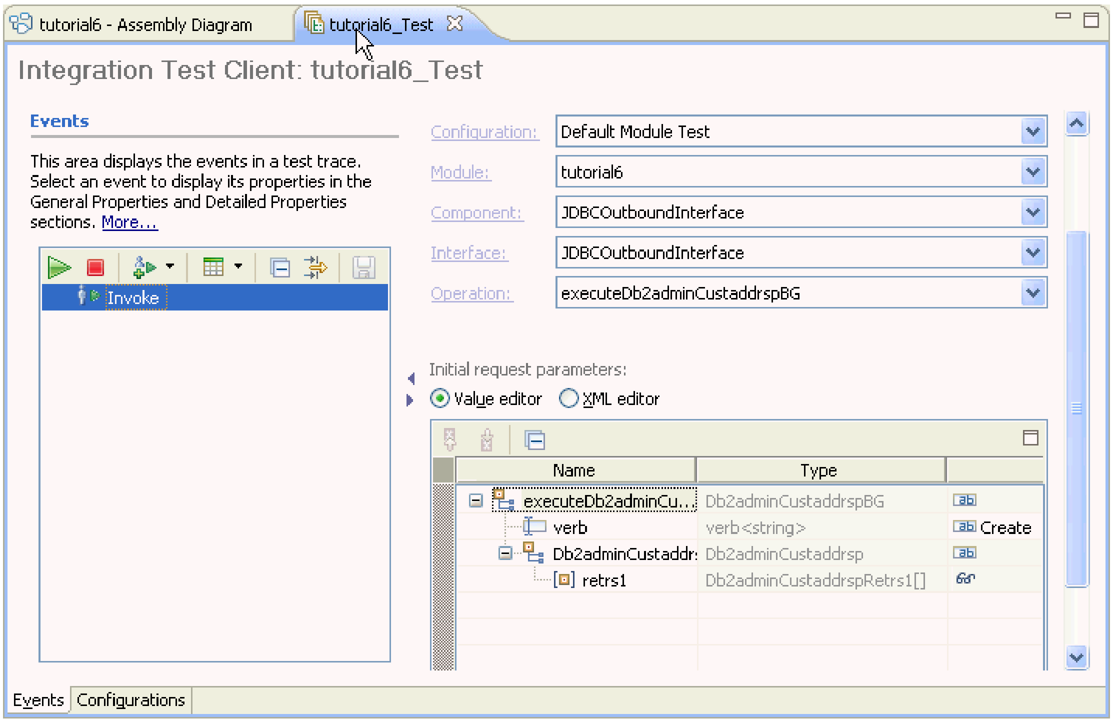
☑ If server is started, publish changes immediately

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 IBM Integration Designer 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.



Integration Test Client: tutorial6_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...

Configuration: Default Module Test

Module: tutorial6

Component: JDBCOutboundInterface

Interface: JDBCOutboundInterface

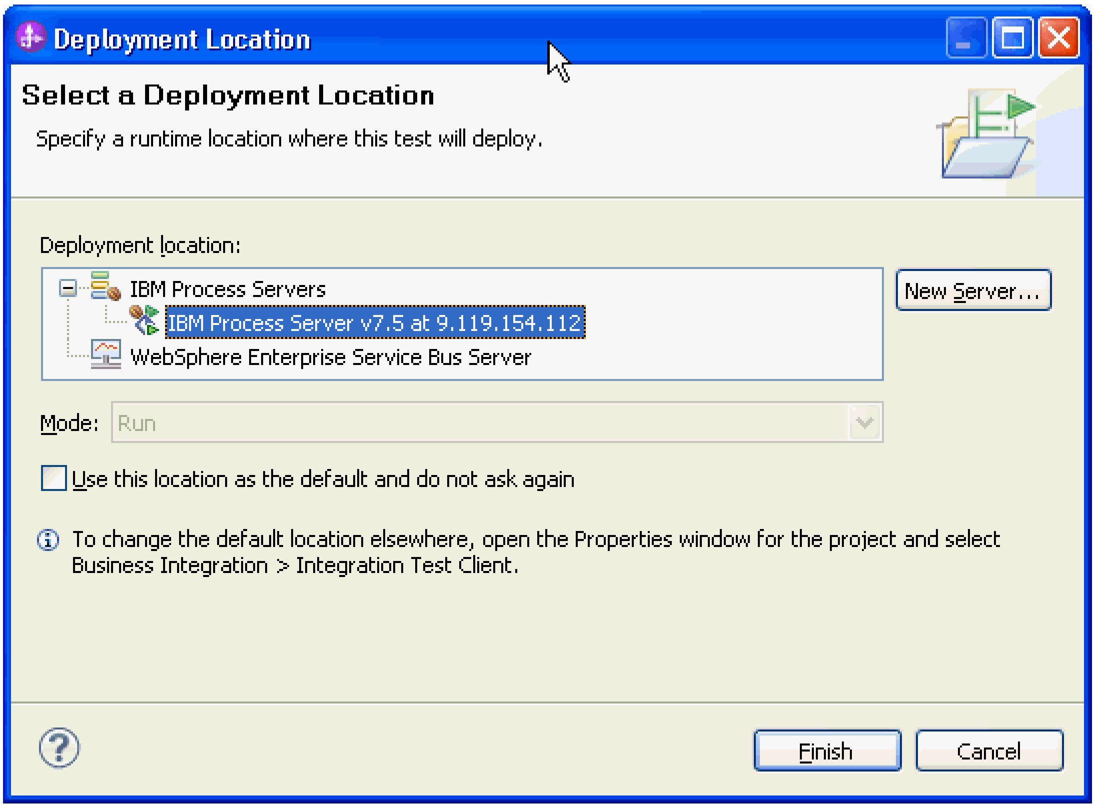
Operation: executeDb2adminCustaddrspBG

Initial request parameters:

● Value editor ○ XML editor

Name	Type	
executeDb2adminCu...	Db2adminCustaddrspBG	
verb	verb<string>	Create
Db2adminCustaddr:	Db2adminCustaddrsp	
retrs1	Db2adminCustaddrspRetrs1[]	

3. Execute the service by click ▶.
4. In the Select Deployment Location window as shown below, select the server, and click **Finish**.



Deployment Location

Select a Deployment Location

Specify a runtime location where this test will deploy.

Deployment location:

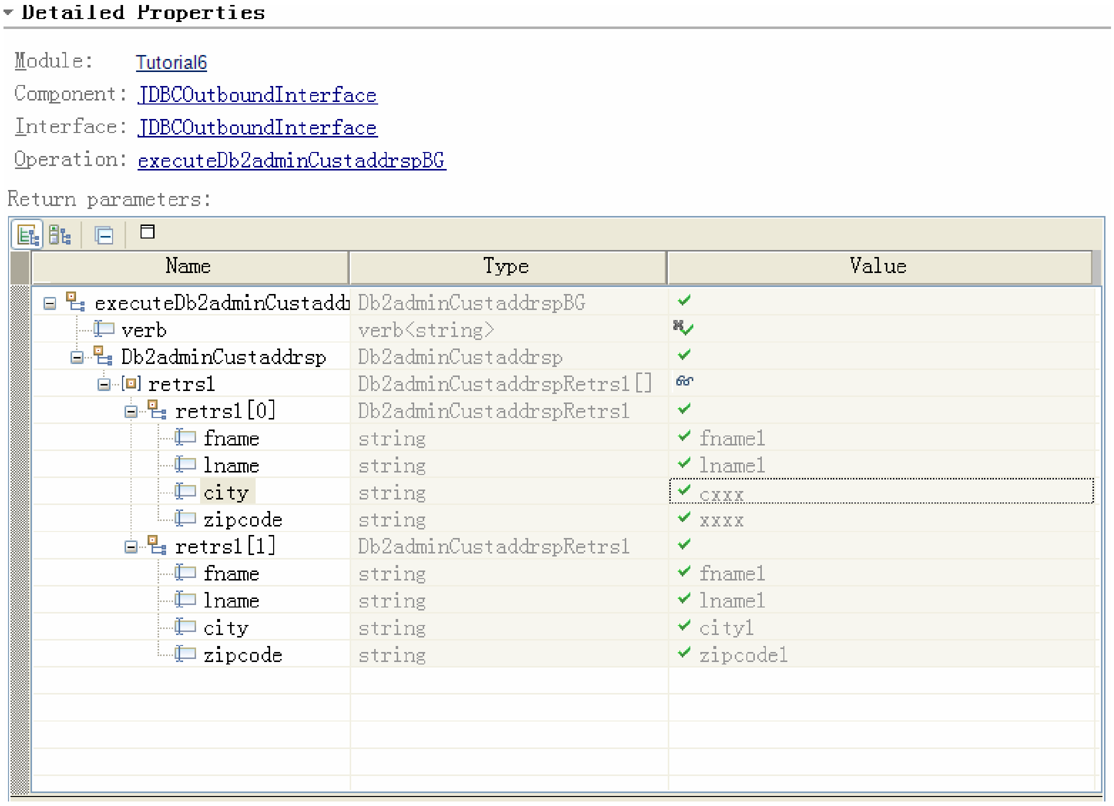
- IBM Process Servers
  - IBM Process Server v7.5 at 9.119.154.112
- WebSphere Enterprise Service Bus Server

Mode: Run

□ Use this location as the default and do not ask again

To change the default location elsewhere, open the Properties window for the project and select Business Integration > Integration Test Client.

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: JDBCOutboundInterface
Interface: JDBCOutboundInterface
Operation: executeDb2adminCustaddrspBG

Return parameters:

Name	Type	Value
executeDb2adminCustadd	Db2adminCustaddrspBG	✓
verb	verb<string>	✓
Db2adminCustaddrsp	Db2adminCustaddrsp	✓
retrs1	Db2adminCustaddrspRetrs1[]	
retrs1[0]	Db2adminCustaddrspRetrs1	✓
fname	string	✓ fname1
lname	string	✓ lname1
city	string	✓ cxxx
zipcode	string	✓ xxxx
retrs1[1]	Db2adminCustaddrspRetrs1	✓
fname	string	✓ fname1
lname	string	✓ lname1
city	string	✓ city1
zipcode	string	✓ zipcode1

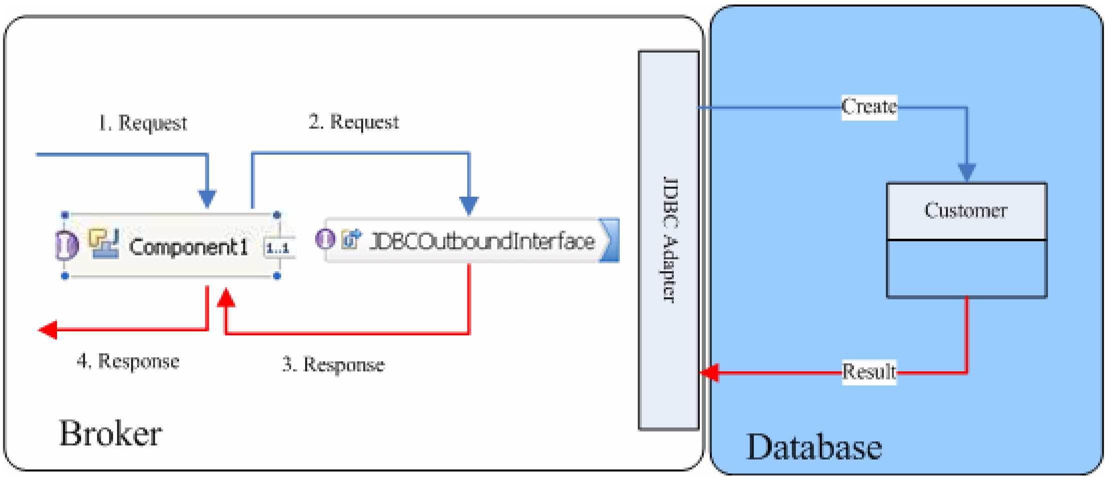
Chapter 8. Tutorial 7: Sending data to the DB2 database within XA Transaction (outbound processing)

This scenario demonstrates how WebSphere Adapter for JDBC 7.5.0.0 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

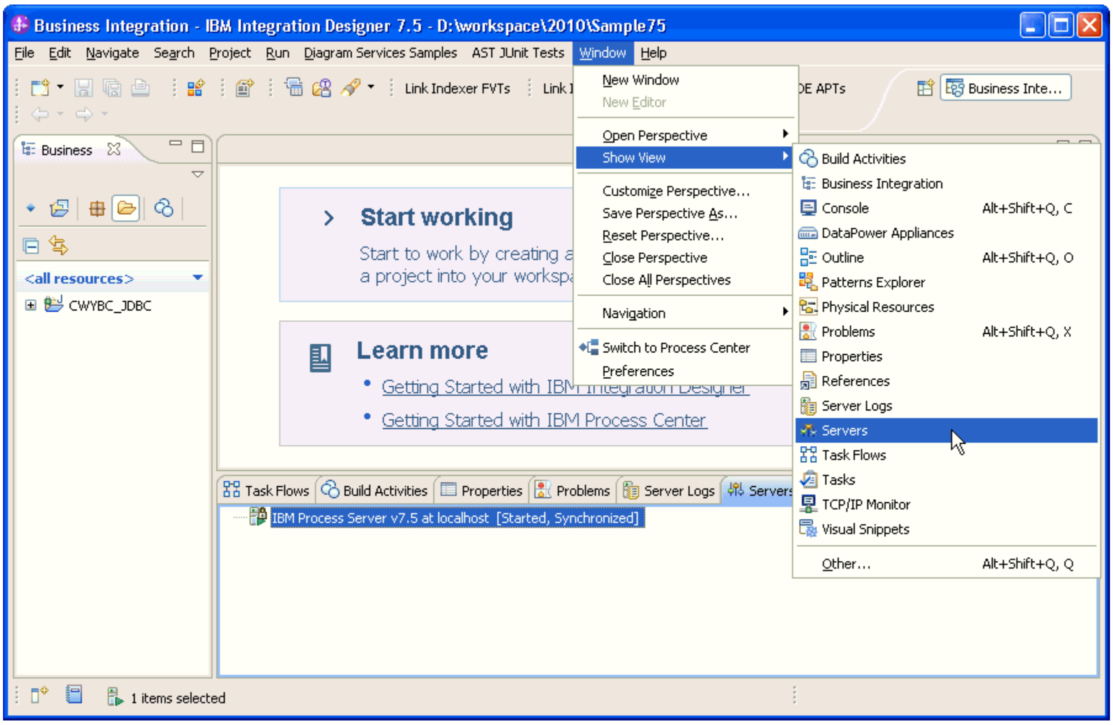
```
CREATE TABLE CUSTOMER  (
     "PKEY" INTEGER NOT NULL PRIMARY KEY,
     "FNAME" VARCHAR(20) ,
     "LNAME" VARCHAR(20) ,
     "CCODE" VARCHAR(10) ) ;
```

Create an authentication alias

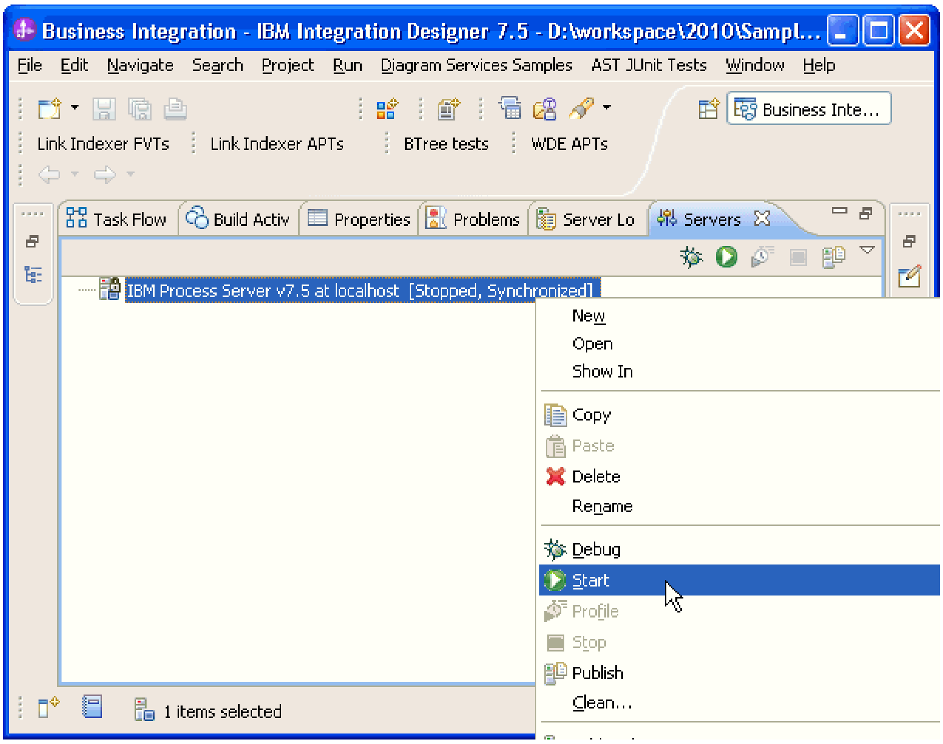
The authentication alias needs to be set because the adapter uses the username and password set in the authentication alias to connect to the database. This authentication alias will be used later when generating the artifacts for the module.

Here are the steps to set the authentication alias in IBM Process Server administration console.

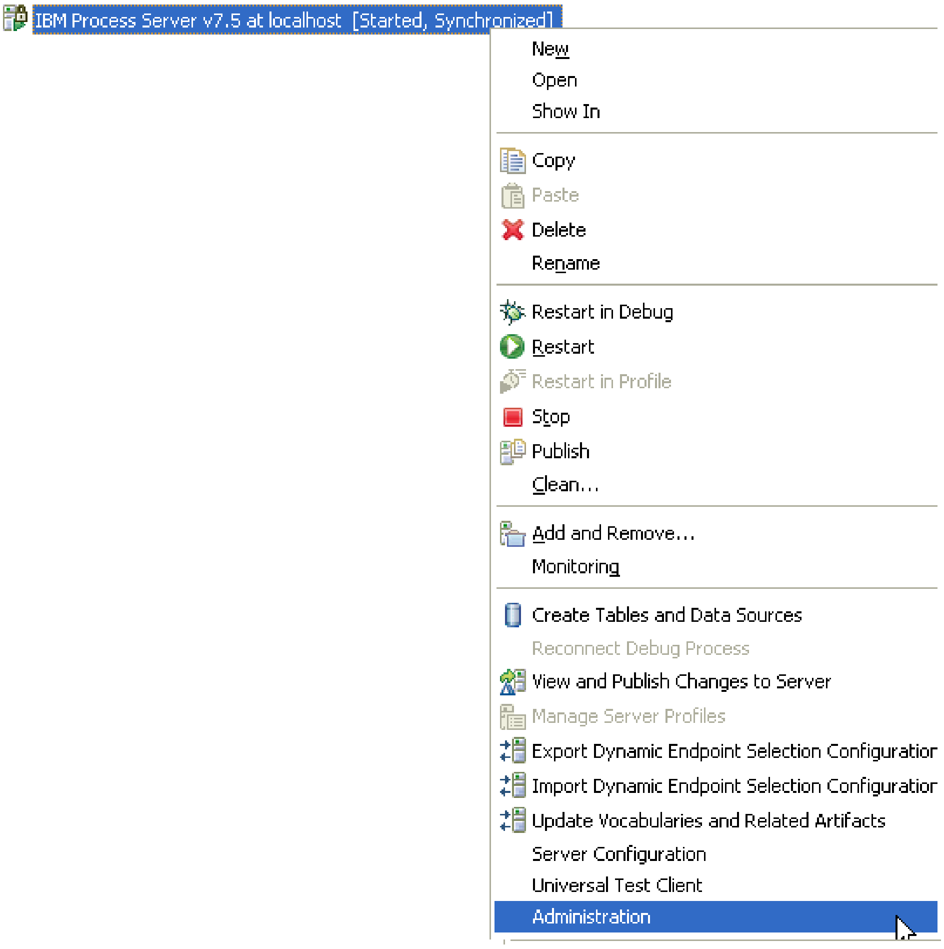
1. In IBM Integration Designer, switch to the **Servers** view by selecting **Windows > Show View > Servers**.



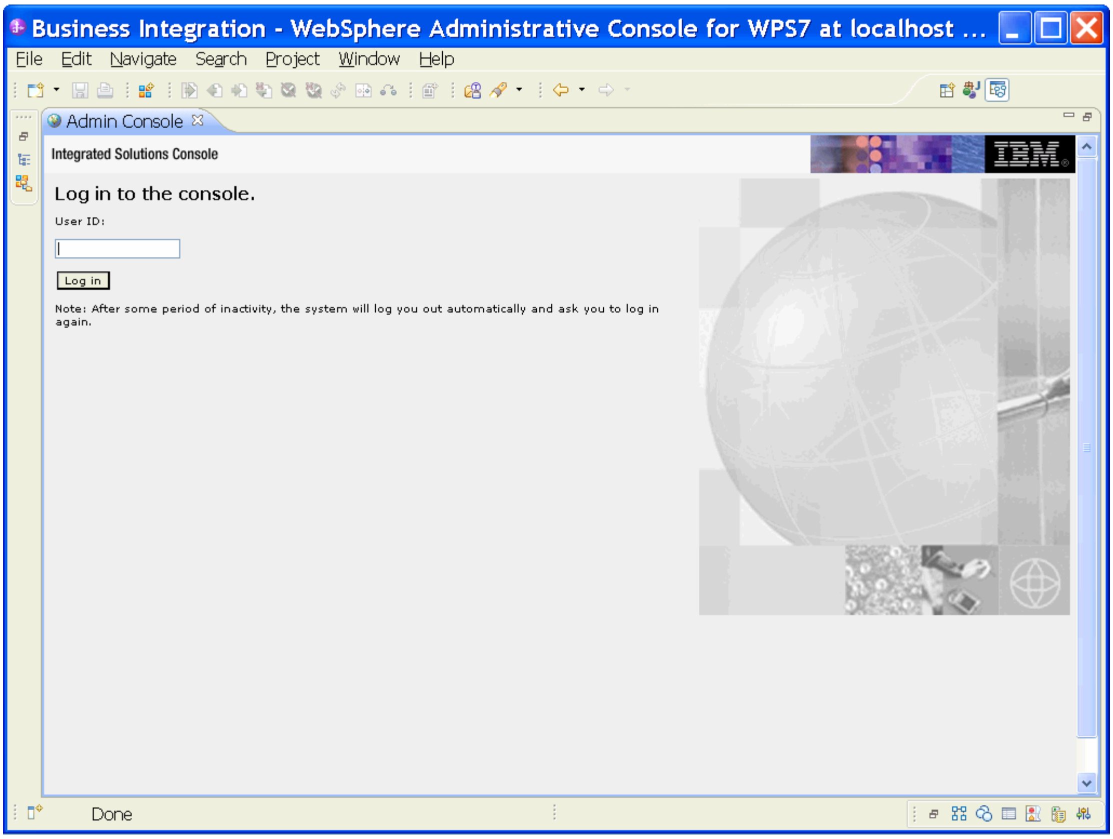
2. In the **Servers** view, right-click the server that you want to start, and select **Start**.


Business Integration - IBM Integration Designer 7.5 - D:\workspace\2010\Sampl...
File Edit Navigate Search Project Run Diagram Services Samples AST JUnit Tests Window Help
Link Indexer FVTs
Link Indexer APTs
BTree tests
WDE APTs
Business Inte...
Task Flow
Build Activ
Properties
Problems
Server Lo
Servers
IBM Process Server v7.5 at localhost [Stopped, Synchronized]
New
Open
Show In
Copy
Paste
Delete
Rename
Debug
Start
Profile
Stop
Publish
Clean...
1 items selected

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


IBM Process Server v7.5 at localhost [Started, Synchronized]
New
Open
Show In
Copy
Paste
Delete
Rename
Restart in Debug
Restart
Restart in Profile
Stop
Publish
Clean...
Add and Remove...
Monitoring
Create Tables and Data Sources
Reconnect Debug Process
View and Publish Changes to Server
Manage Server Profiles
Export Dynamic Endpoint Selection Configuratior
Import Dynamic Endpoint Selection Configuratior
Update Vocabularies and Related Artifacts
Server Configuration
Universal Test Client
Administration

4. Log on to the administrative console by entering the username and password (if required).



Business Integration - WebSphere Administrative Console for WPS7 at localhost ...

File Edit Navigate Search Project Window Help
Admin Console
Integrated Solutions Console
IBM
Done

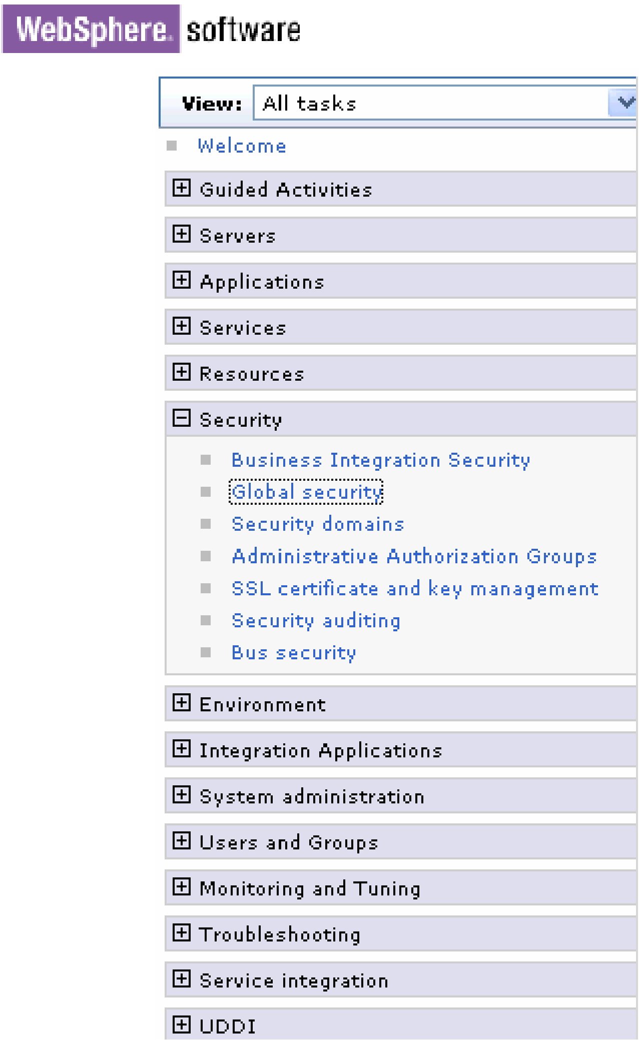
Log in to the console.

User ID:

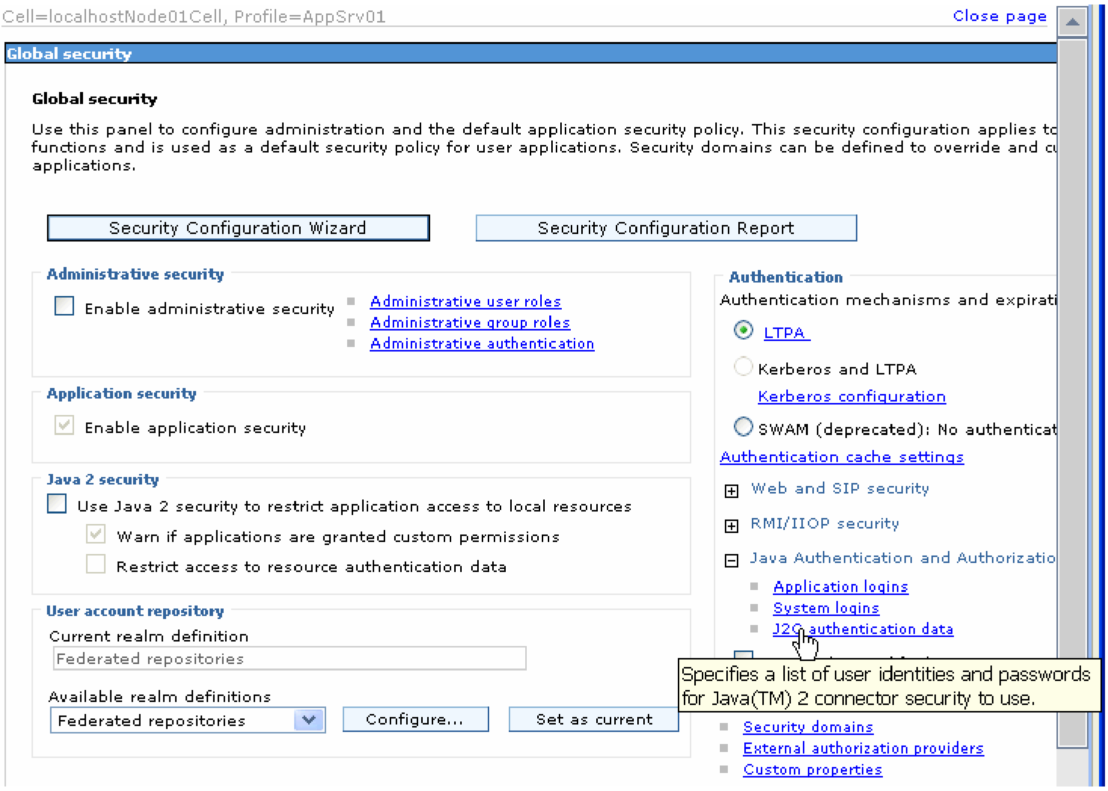
Log in

Note: After some period of inactivity, the system will log you out automatically and ask you to log in again.

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



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 cu 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

Authentication

Authentication mechanisms and expirati

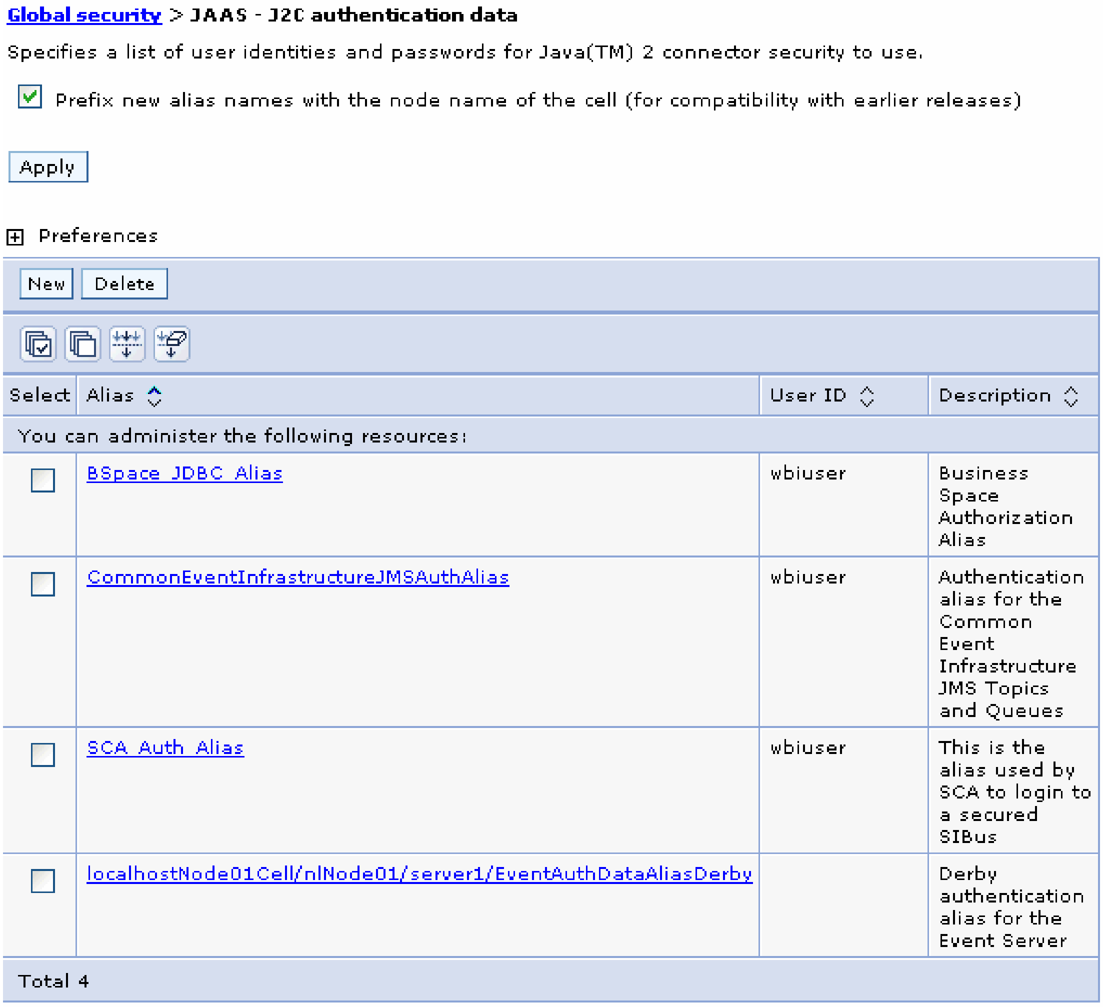
● LTPA

○ Kerberos and LTPA

○ SWAM (deprecated): No authenticat

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

A list of existing aliases is displayed.



Global security > JAAS - J2C authentication data

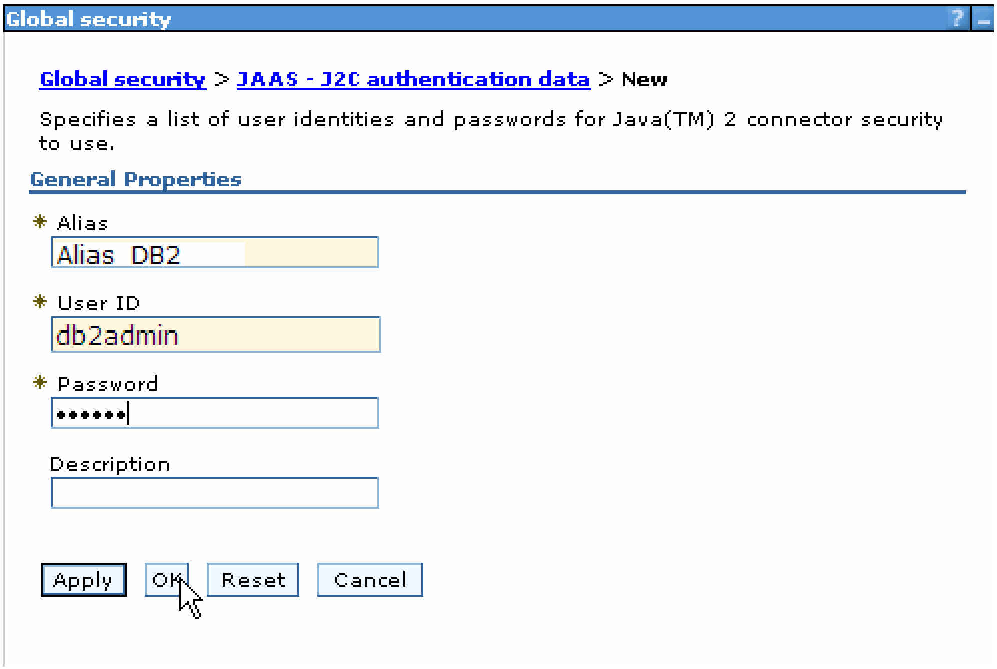
Specifies a list of user identities and passwords for Java(TM) 2 connector security to use.

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

Select	Alias	User ID	Description
You can administer the following resources:			
□	BSpace_JDBC_Alias	wbiuser	Business Space Authorization Alias
□	CommonEventInfrastructureJMSAuthAlias	wbiuser	Authentication alias for the Common Event Infrastructure JMS Topics and Queues
□	SCA_Auth_Alias	wbiuser	This is the alias used by SCA to login to a secured SIBus
□	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, 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

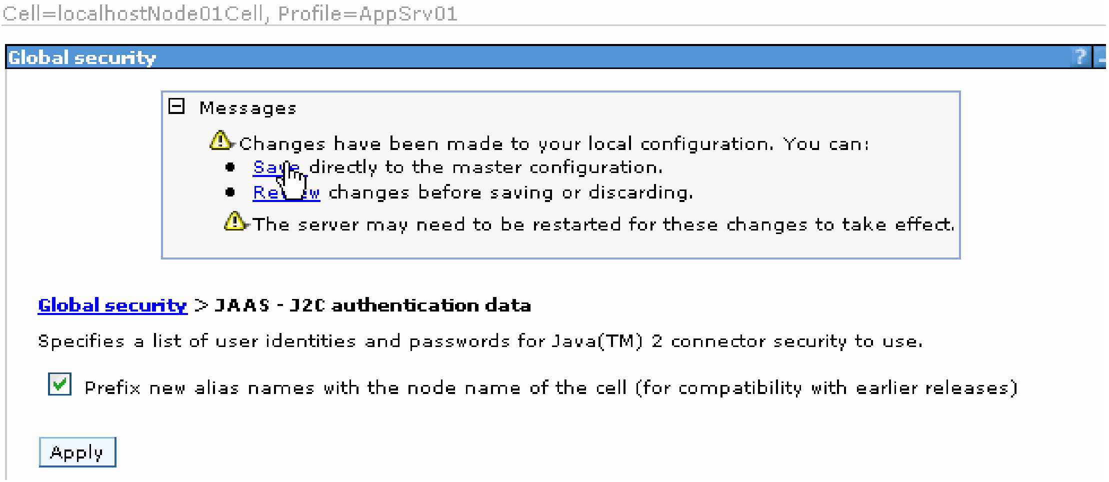
* Alias
Alias DB2

* User ID
db2admin

* Password
••••••

Description

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.

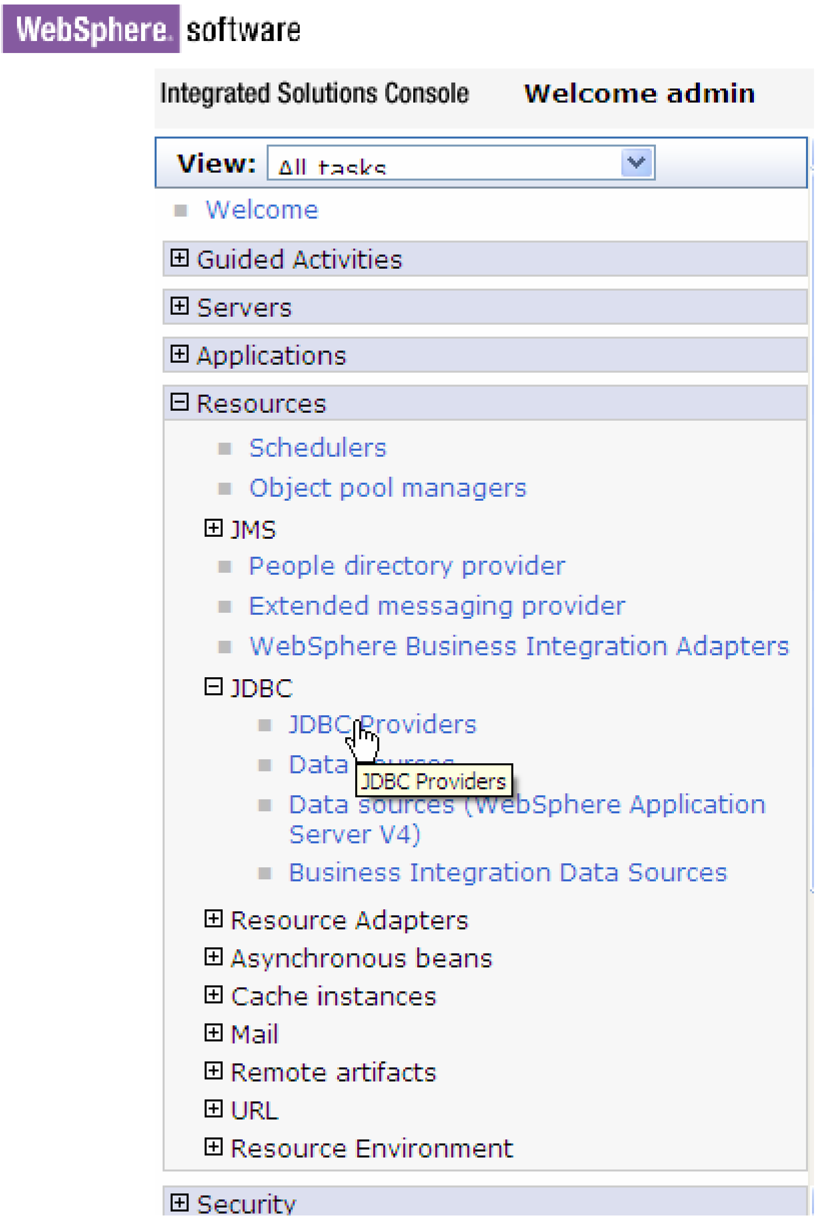
Global security > JAAS - J2C authentication data

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

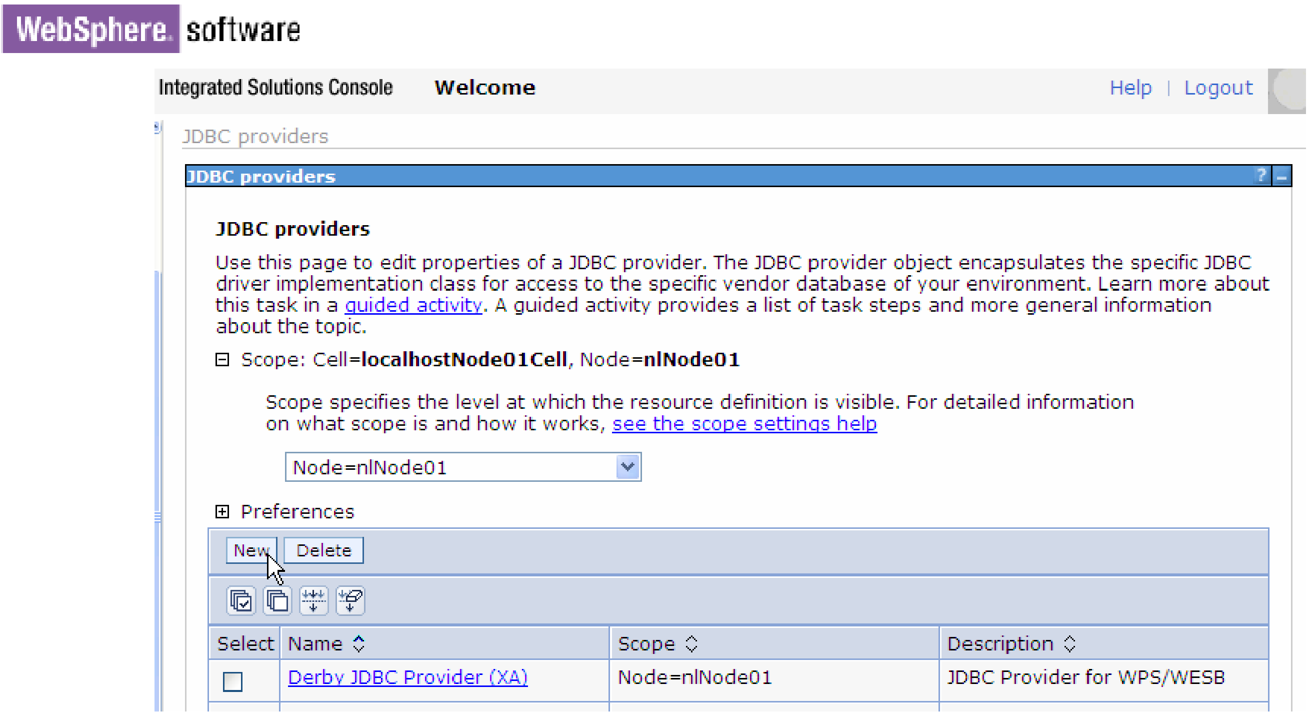
☑ Prefix new alias names with the node name of the cell (for compatibility with earlier releases)

Create a data source

1. In administrative console, select **Resources->JDBC->JDBC Providers**.



2. On the right, select **Node=nlNode01** from the drop-down list, and click **New**.



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=**nlNode01**

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

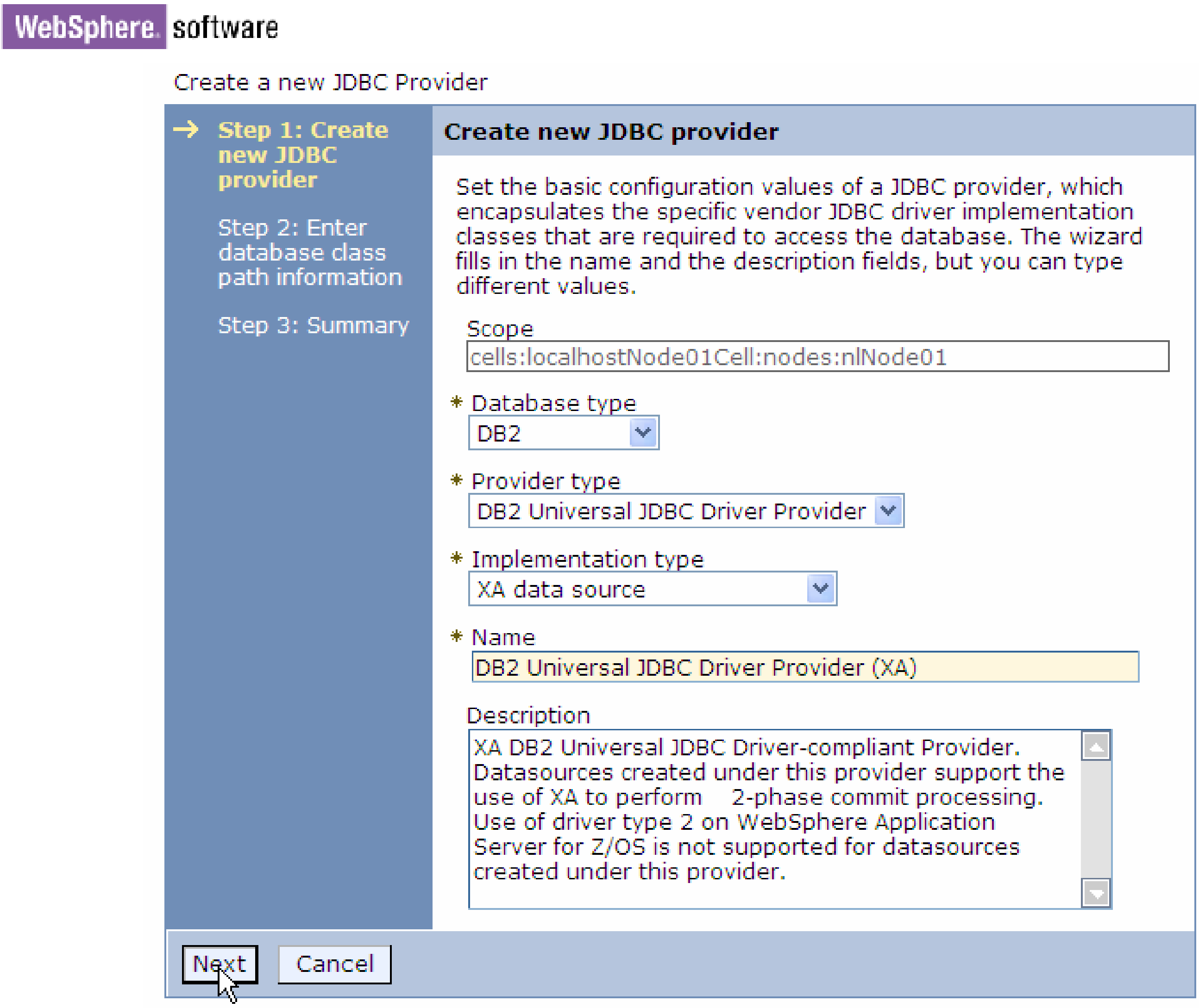
Node=nlNode01

⊞ Preferences

Select	Name	Scope	Description
□	Derby JDBC Provider (XA)	Node=nlNode01	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:nlNode01

* Database type
DB2

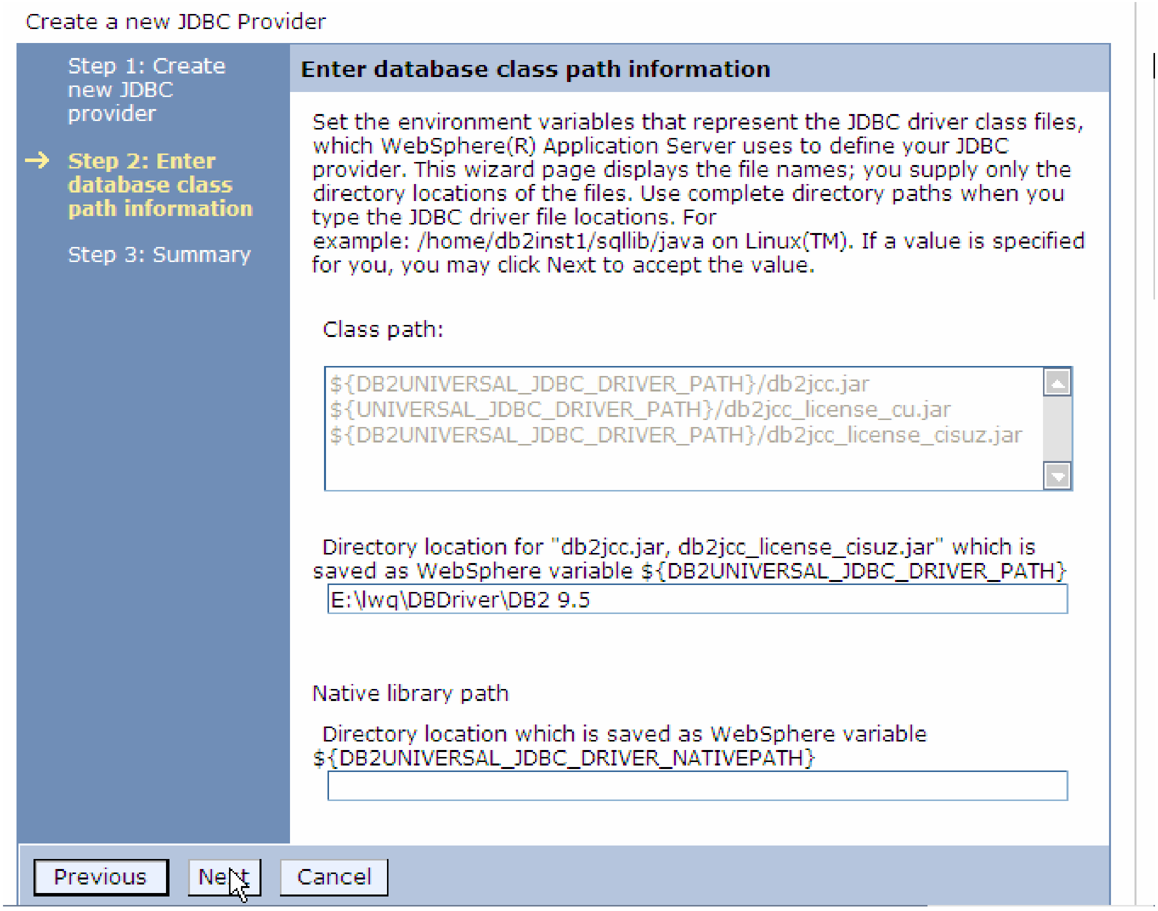
* Provider type
DB2 Universal JDBC Driver Provider

* 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.

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

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:

```
${DB2UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc.jar
${UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc_license_cu.jar
${DB2UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc_license_cisuz.jar
```

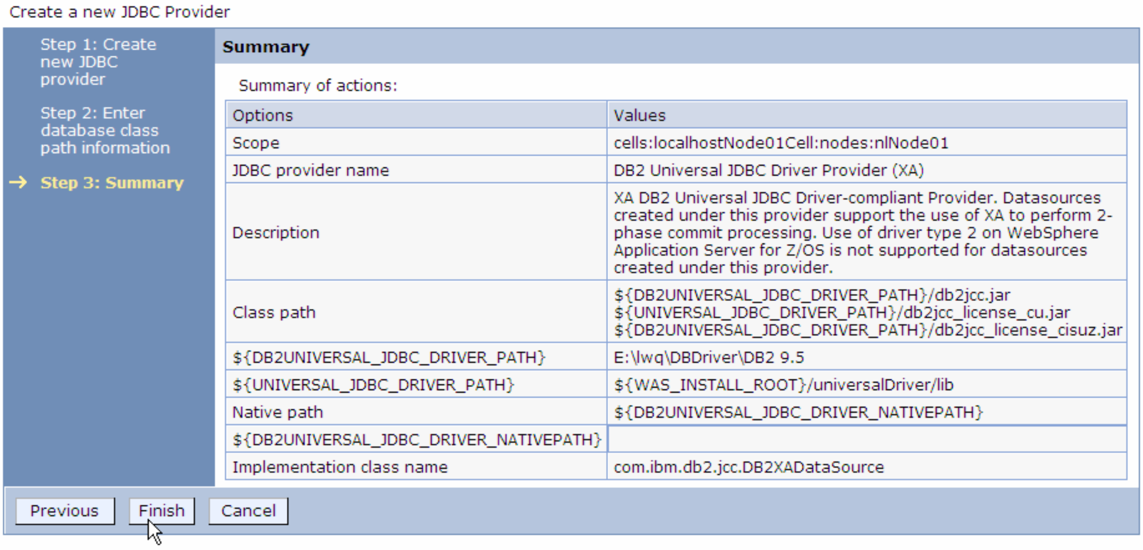
Directory location for "db2jcc.jar, db2jcc_license_cisuz.jar" which is saved as WebSphere variable ${DB2UNIVERSAL_JDBC_DRIVER_PATH}

E:\lwq\DBDriver\DB2 9.5

Native library path

Directory location which is saved as WebSphere variable ${DB2UNIVERSAL_JDBC_DRIVER_NATIVEPATH}

5. Click **Finish**.



Create a new JDBC Provider

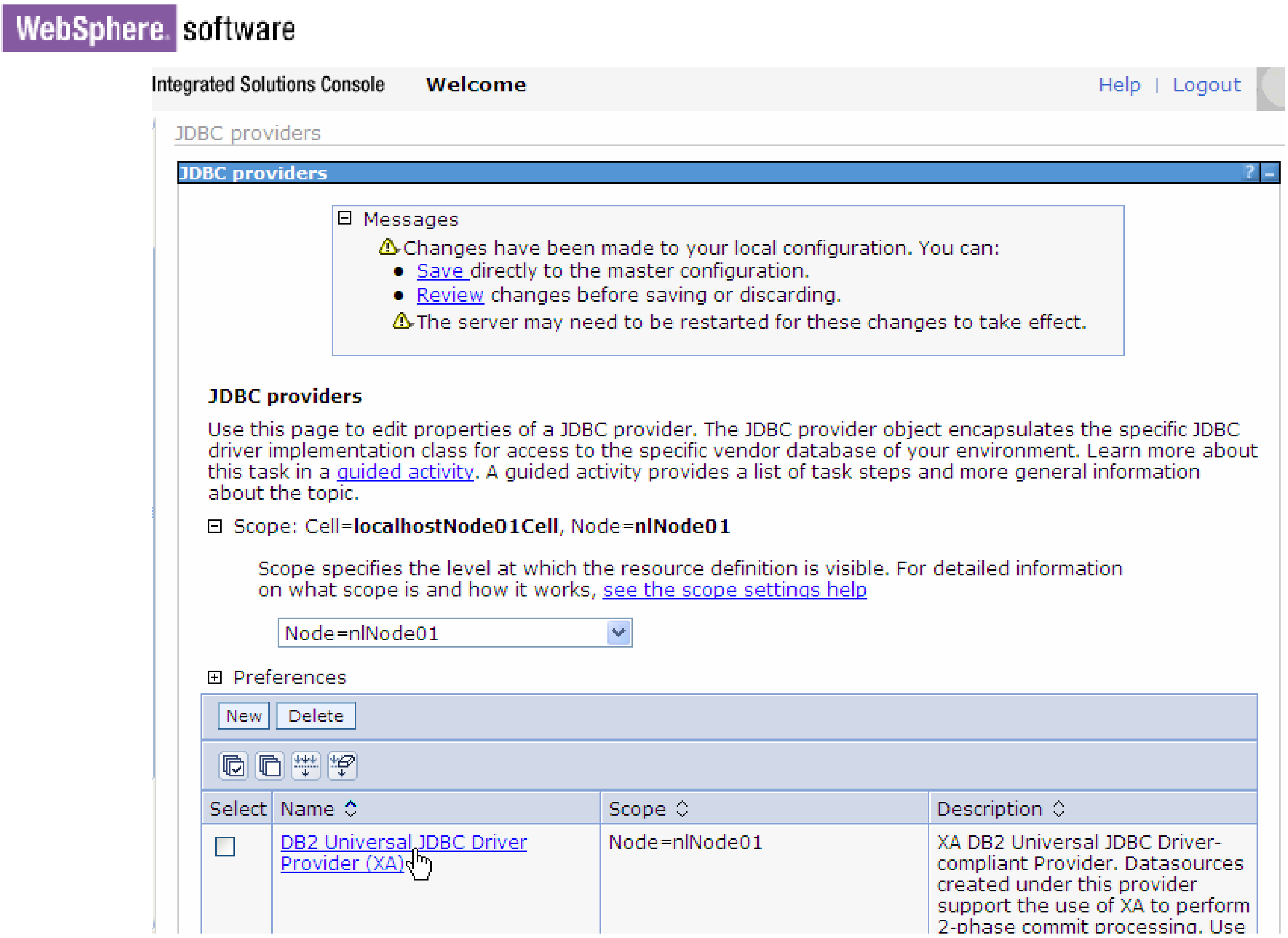
Step 1: Create new JDBC provider
Step 2: Enter database class path information
→ Step 3: Summary

Summary

Summary of actions:

Options	Values
Scope	cells:localhostNode01Cell:nodes:nlNode01
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	${DB2UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc.jar ${UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc_license_cu.jar ${DB2UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc_license_cisuz.jar
${DB2UNIVERSAL_JDBC_DRIVER_PATH}	E:\lwq\DBDriver\DB2 9.5
${UNIVERSAL_JDBC_DRIVER_PATH}	${WAS_INSTALL_ROOT}/universalDriver/lib
Native path	${DB2UNIVERSAL_JDBC_DRIVER_NATIVEPATH}
${DB2UNIVERSAL_JDBC_DRIVER_NATIVEPATH}	
Implementation class name	com.ibm.db2.jcc.DB2XADataSource

6. Click the JDBC Provider that you just created.



WebSphere software

Integrated Solutions Console Welcome Help | Logout

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=**nlNode01**

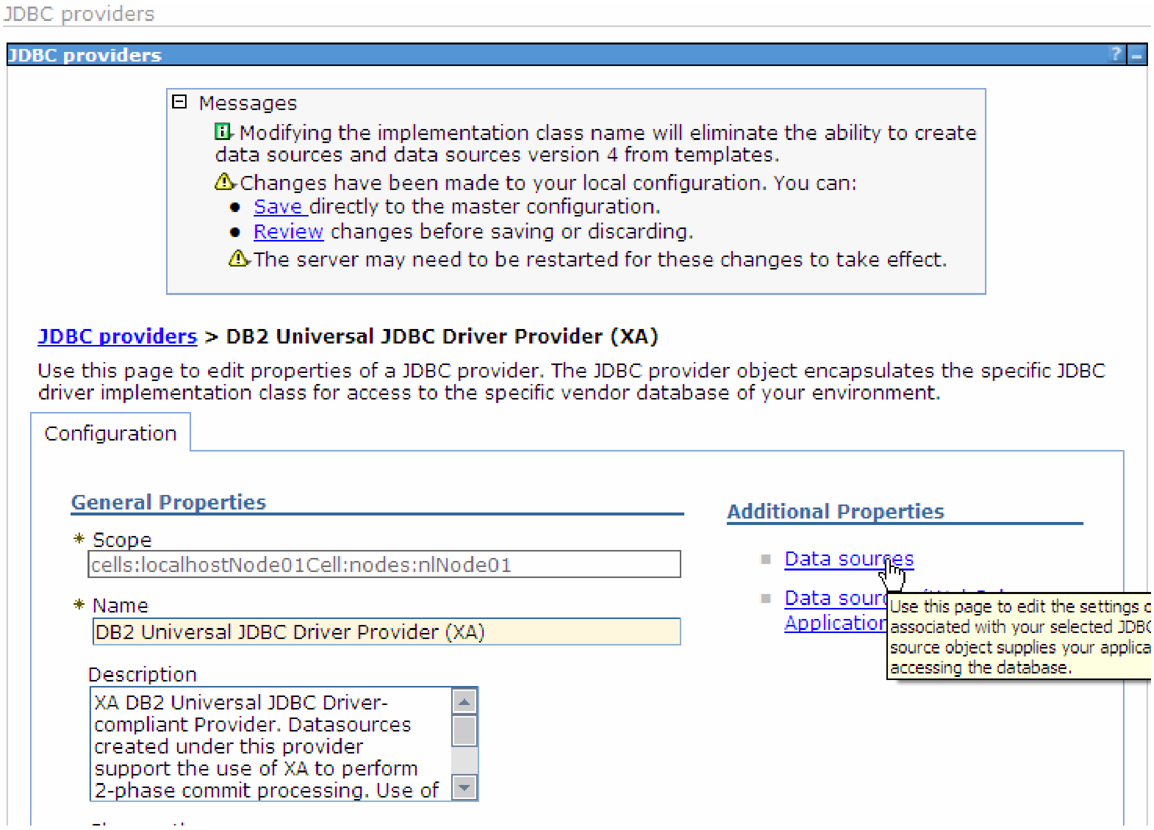
Scope specifies the level at which the resource definition is visible. For detailed information on what scope is and how it works, see the scope settings help

Node=nlNode01

⊞ Preferences

Select	Name ◇	Scope ◇	Description ◇
□	DB2 Universal JDBC Driver Provider (XA)	Node=nlNode01	XA DB2 Universal JDBC Driver-compliant Provider. Datasources created under this provider support the use of XA to perform 2-phase commit processing. Use

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.

Configuration

General Properties

* Scope
cells:localhostNode01Cell:nodes:nlNode01

* Name
DB2 Universal JDBC Driver Provider (XA)

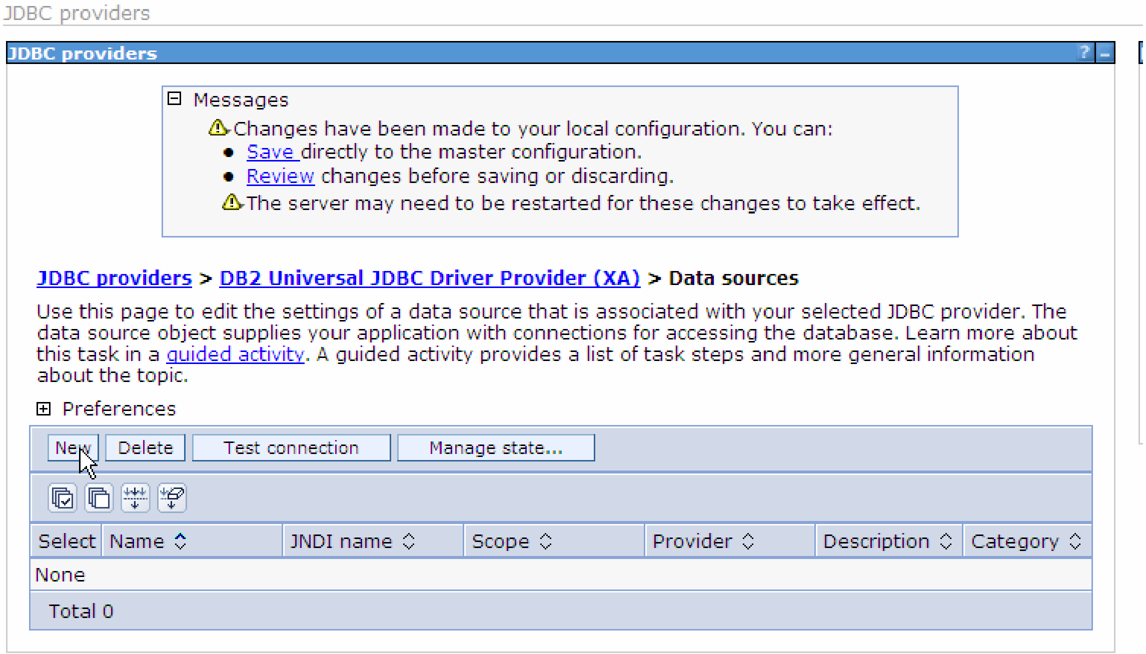
Description
XA DB2 Universal JDBC Driver-compliant Provider. Datasources created under this provider support the use of XA to perform 2-phase commit processing. Use of

Additional Properties

- Data sources
- Data sourc
- Application

Use this page to edit the settings c associated with your selected JDBC source object supplies your applica accessing the database.

8. Click **New**.



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 > DB2 Universal JDBC Driver Provider (XA) > Data sources

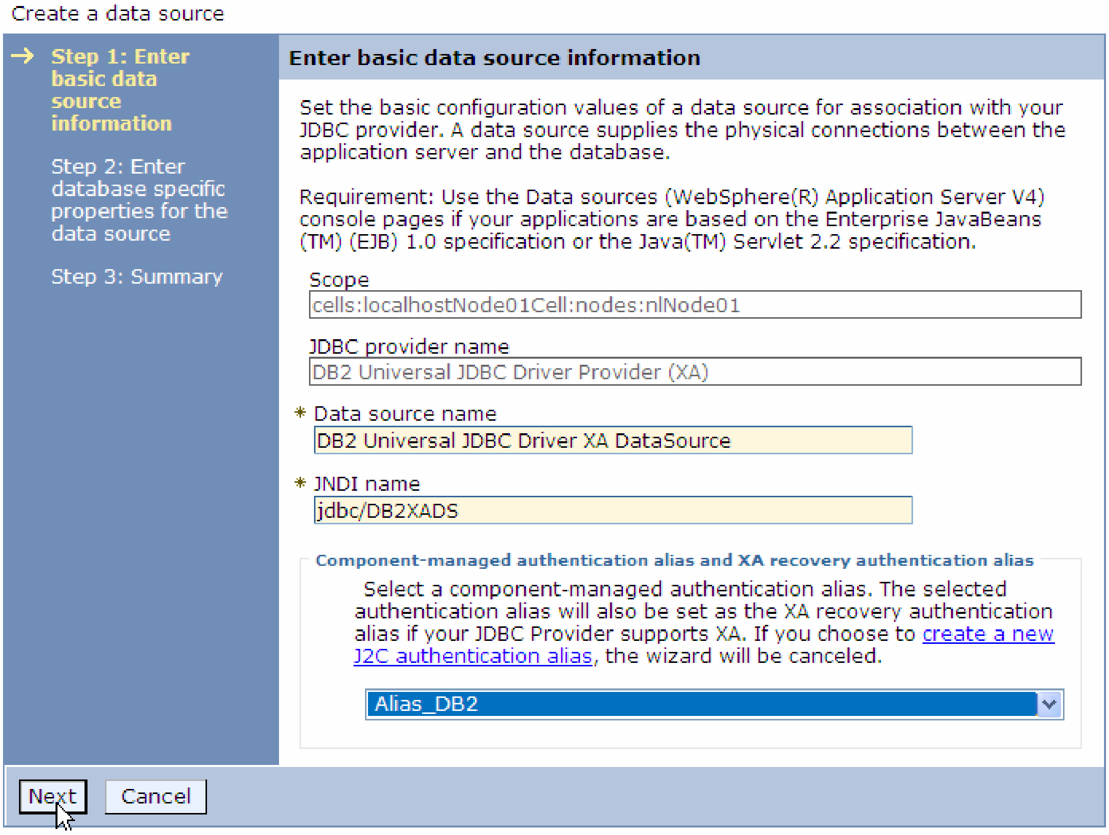
Use this page to edit the settings of a data source that is associated with your selected JDBC provider. The data source object supplies your application with connections for accessing the database. Learn more about this task in a guided activity. A guided activity provides a list of task steps and more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

⊞ Preferences

Select	Name ◇	JNDI name ◇	Scope ◇	Provider ◇	Description ◇	Category ◇
None						
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

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: DB2 Universal JDBC Driver Provider (XA)

* Data source name: DB2 Universal JDBC Driver XA DataSource

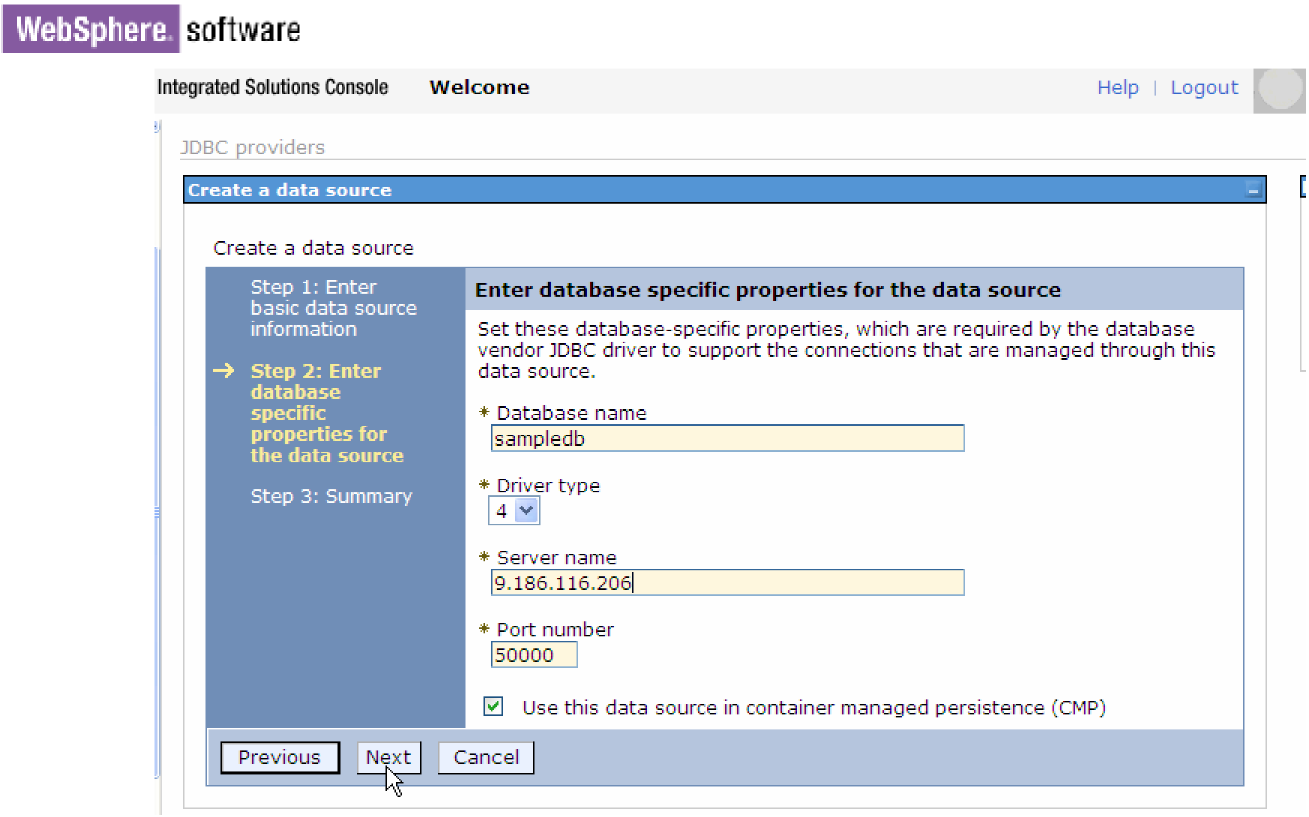
* JNDI name: jdbc/DB2XADS

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.

Alias_DB2

11. Enter the values below for **Database name** and **Server name**. Click **Next**.



WebSphere. software

Integrated Solutions Console Welcome

Help | Logout

JDBC providers

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: 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

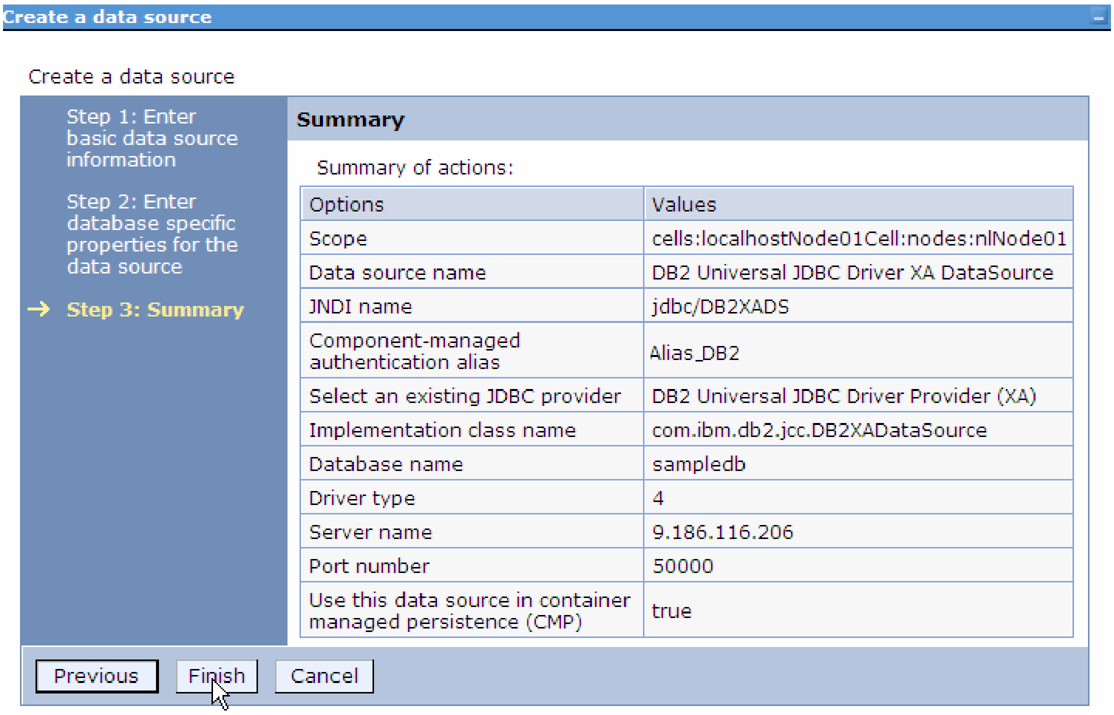
* Driver type: 4

* Server name: 9.186.116.206

* Port number: 50000

☑ Use this data source in container managed persistence (CMP)

12. Click **Finish**.



Create a data source

Create a data source

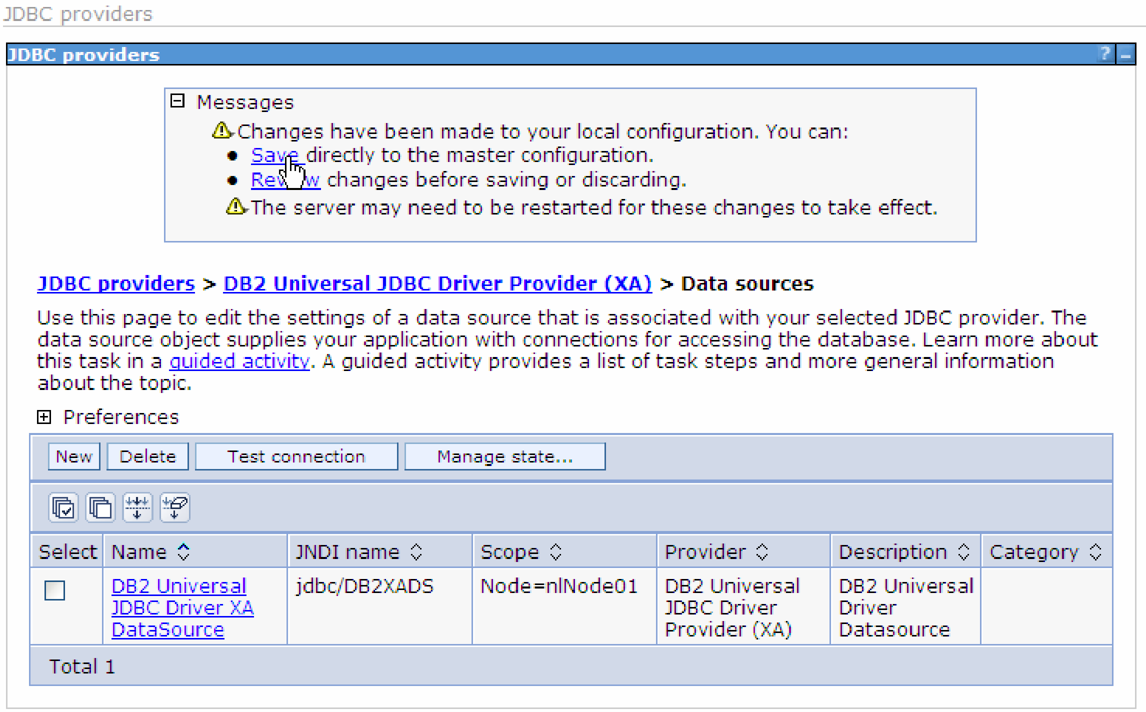
Step 1: Enter basic data source information
Step 2: Enter database specific properties for the data source
→ Step 3: Summary

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

13. In the Messages area, click **Save**. 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.
- Review changes before saving or discarding.

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

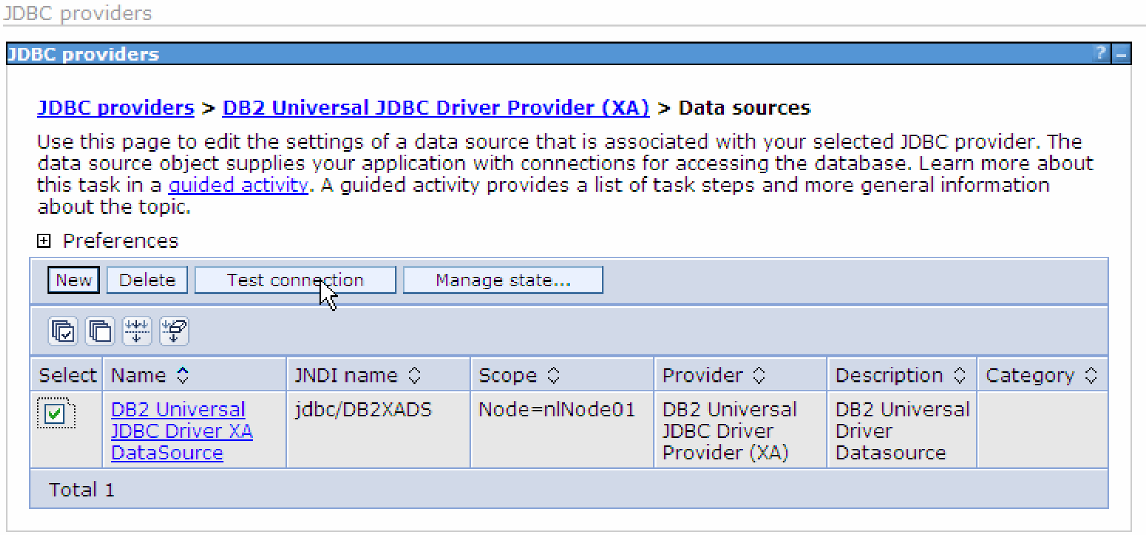
JDBC providers > DB2 Universal JDBC Driver Provider (XA) > **Data sources**

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

Select	Name	JNDI name	Scope	Provider	Description	Category
□	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

JDBC providers

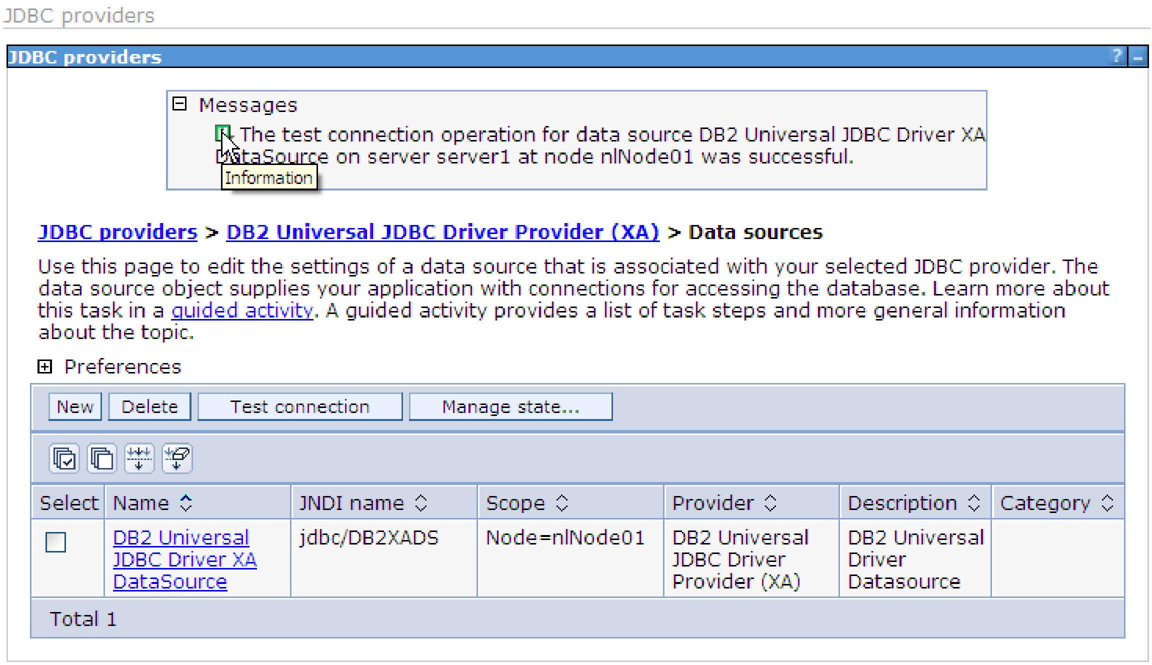
JDBC providers > DB2 Universal JDBC Driver Provider (XA) > **Data sources**

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

Select	Name	JNDI name	Scope	Provider	Description	Category
☑	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.



JDBC providers

JDBC providers

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

Select	Name	JNDI name	Scope	Provider	Description	Category
□	DB2 Universal JDBC Driver XA DataSource	jdbc/DB2XADS	Node=nlNode01	DB2 Universal JDBC Driver Provider (XA)	DB2 Universal Driver Datasource	

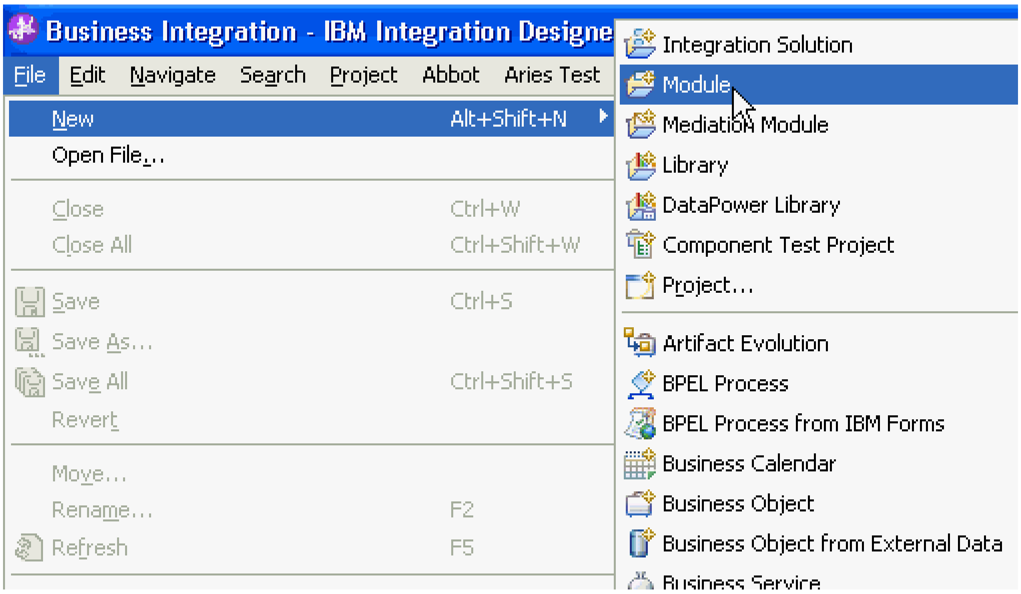
Total 1

15. Close the **Administrative Console** tab.

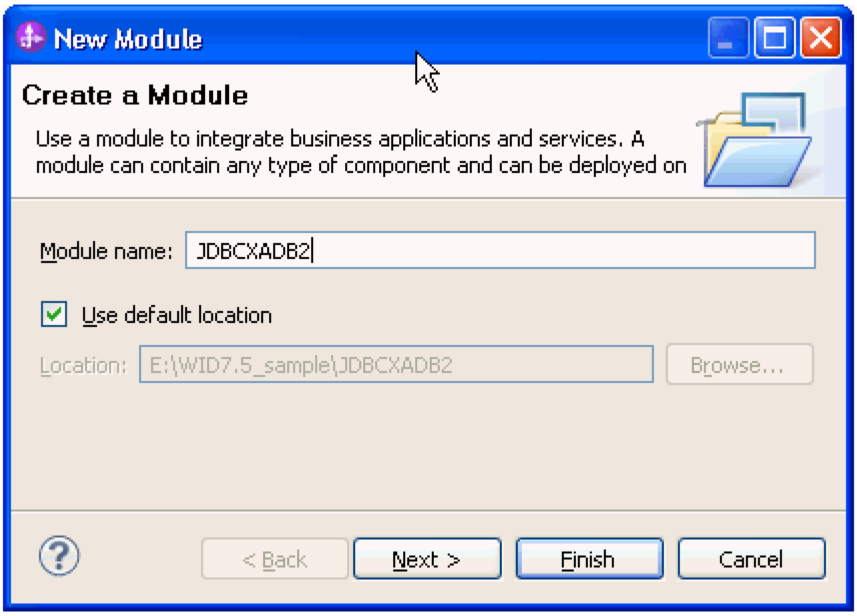
Configure the adapter for outbound processing

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

1. Switch to the Business Integration perspective in IBM Integration Designer.
2. Select **File->New->Module** to create a Module project.



3. Specify the module name as **JDBCXADB2**, and click **Finish**.



New Module

Create a Module

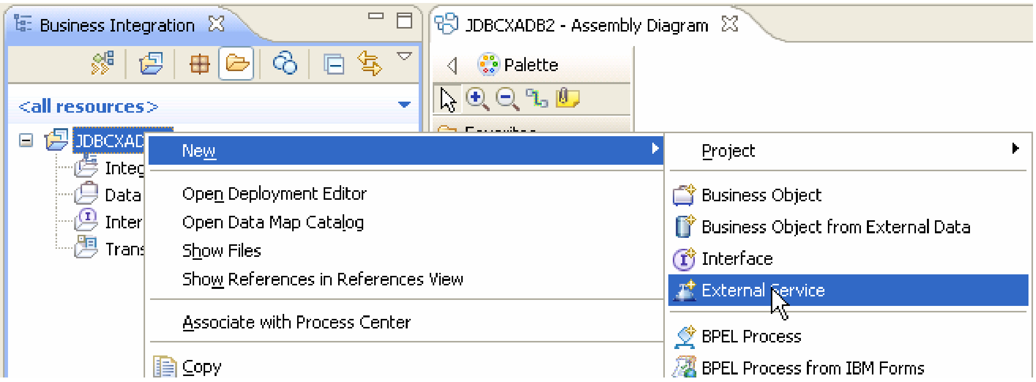
Use a module to integrate business applications and services. A module can contain any type of component and can be deployed on

Module name: JDBCXADB2

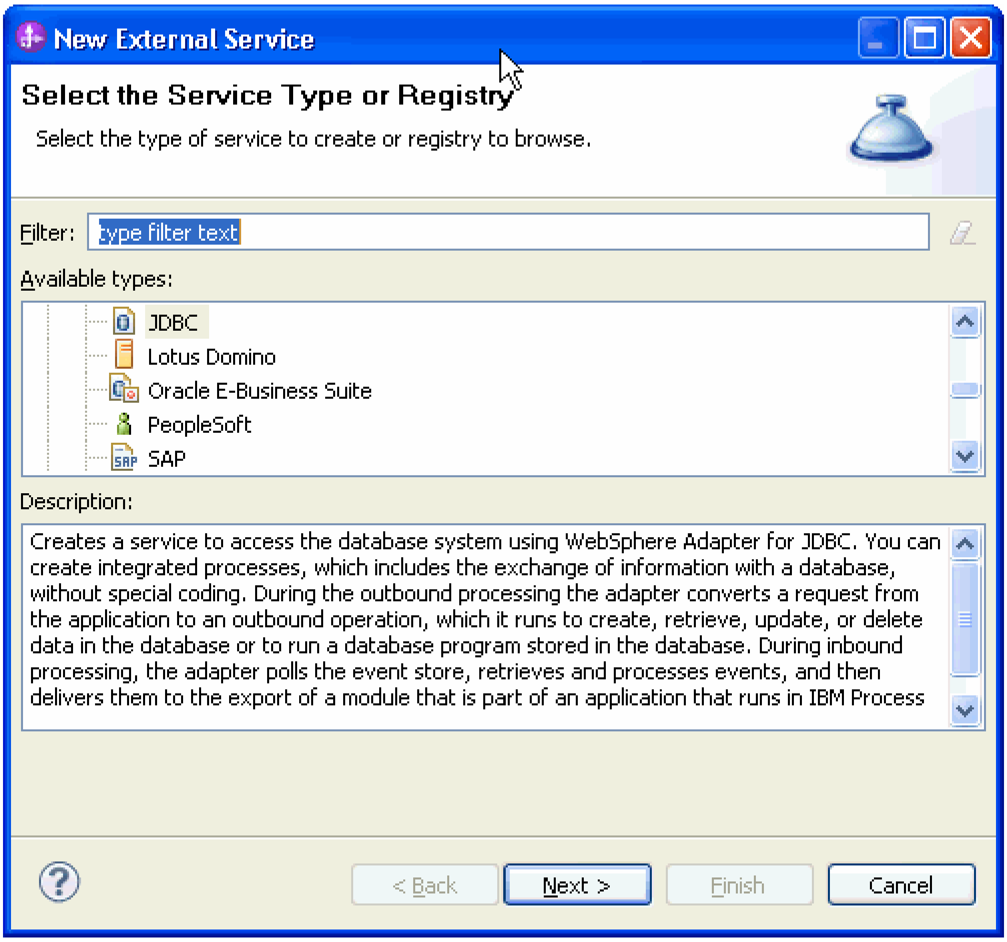
☑ Use default location

Location: E:\WID7.5_sample\JDBCXADB2

4. Right-click **JDBCXADB2**, select **New->External Service.**



5. Select JDBC, and click **Next**.



New External Service

Select the Service Type or Registry

Select the type of service to create or registry to browse.

Filter: type filter text

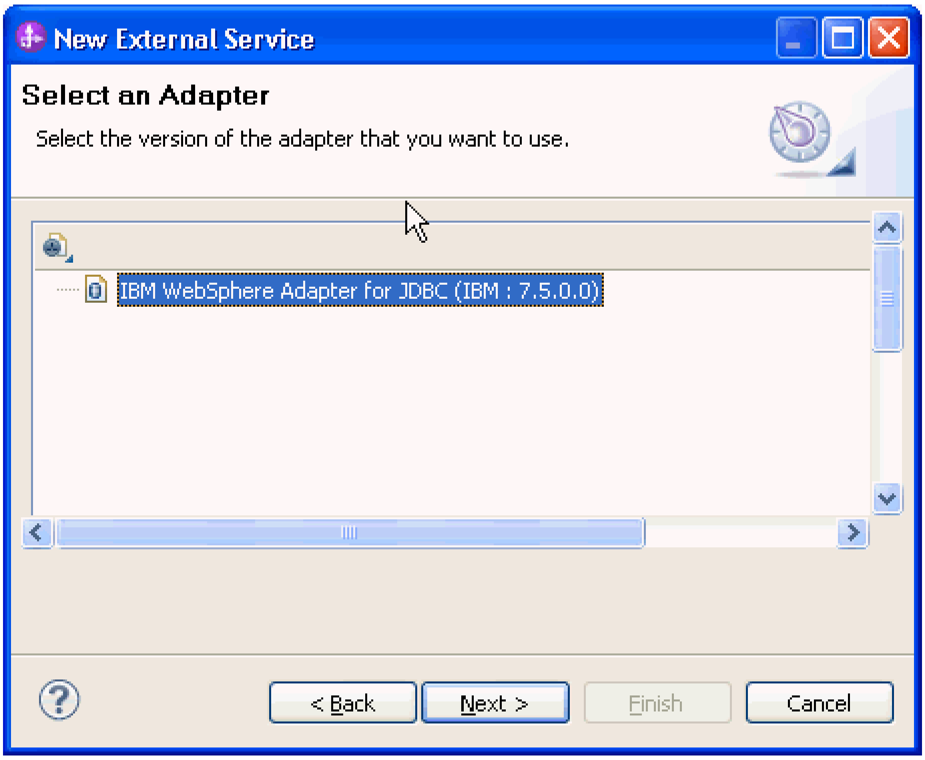
Available types:

- JDBC
- Lotus Domino
- Oracle E-Business Suite
- PeopleSoft
- SAP

Description:

Creates a service to access the database system using WebSphere Adapter for JDBC. You can create integrated processes, which includes the exchange of information with a database, without special coding. During the outbound processing the adapter converts a request from the application to an outbound operation, which it runs to create, retrieve, update, or delete data in the database or to run a database program stored in the database. During inbound processing, the adapter polls the event store, retrieves and processes events, and then delivers them to the export of a module that is part of an application that runs in IBM Process

6. Select **IBM WebSphere Adapter for JDBC (IBM: 7.5.0.0).** Click **Next**.



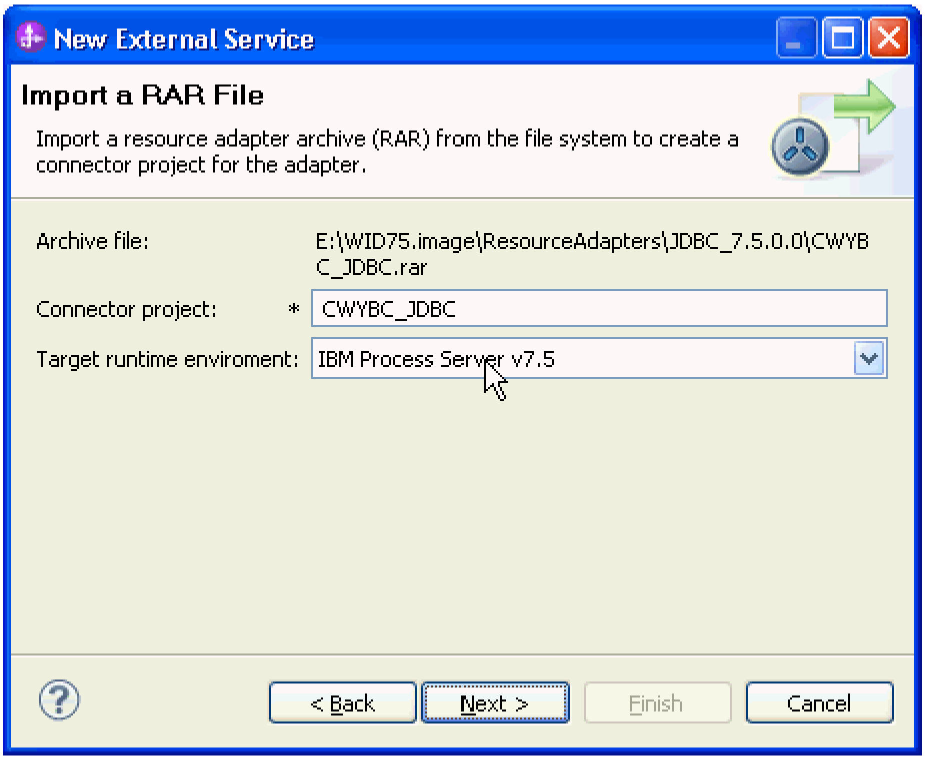
New External Service

Select an Adapter

Select the version of the adapter that you want to use.

IBM WebSphere Adapter for JDBC (IBM : 7.5.0.0)

7. In the **Target Runtime environment** field, select the appropriate runtime and click **Next.**



Import a RAR File

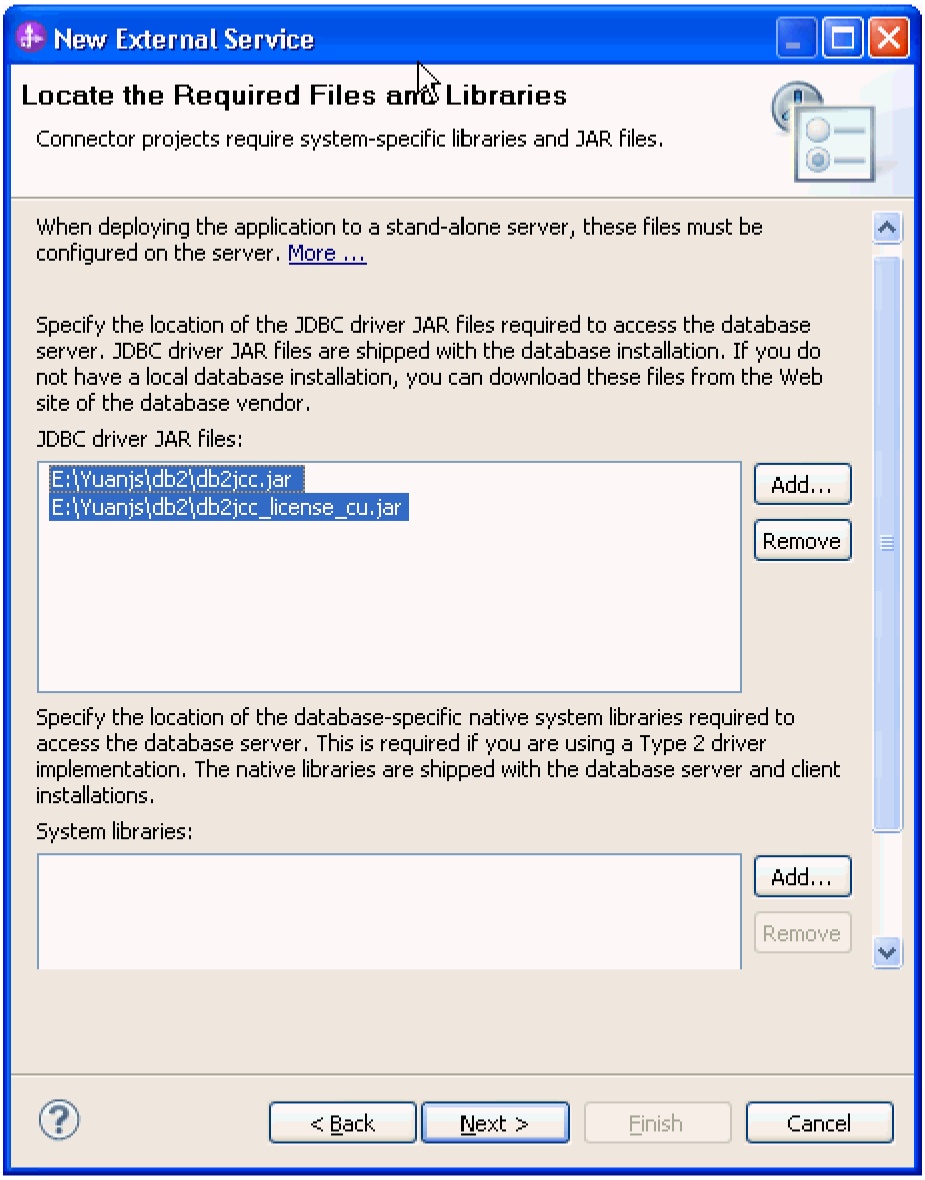
Import a resource adapter archive (RAR) from the file system to create a connector project for the adapter.

Archive file: E:\WID75.image\ResourceAdapters\JDBC_7.5.0.0\CWYBC_JDBC.rar

Connector project: * CWYBC_JDBC

Target runtime enviroment: IBM Process Server v7.5

8. Click **Add** to add the JDBC driver jar to the class path, and click **Next**.



Locate the Required Files and Libraries

Connector projects require system-specific libraries and JAR files.

When deploying the application to a stand-alone server, these files must be configured on the server. More ...

Specify the location of the JDBC driver JAR files required to access the database server. JDBC driver JAR files are shipped with the database installation. If you do not have a local database installation, you can download these files from the Web site of the database vendor.

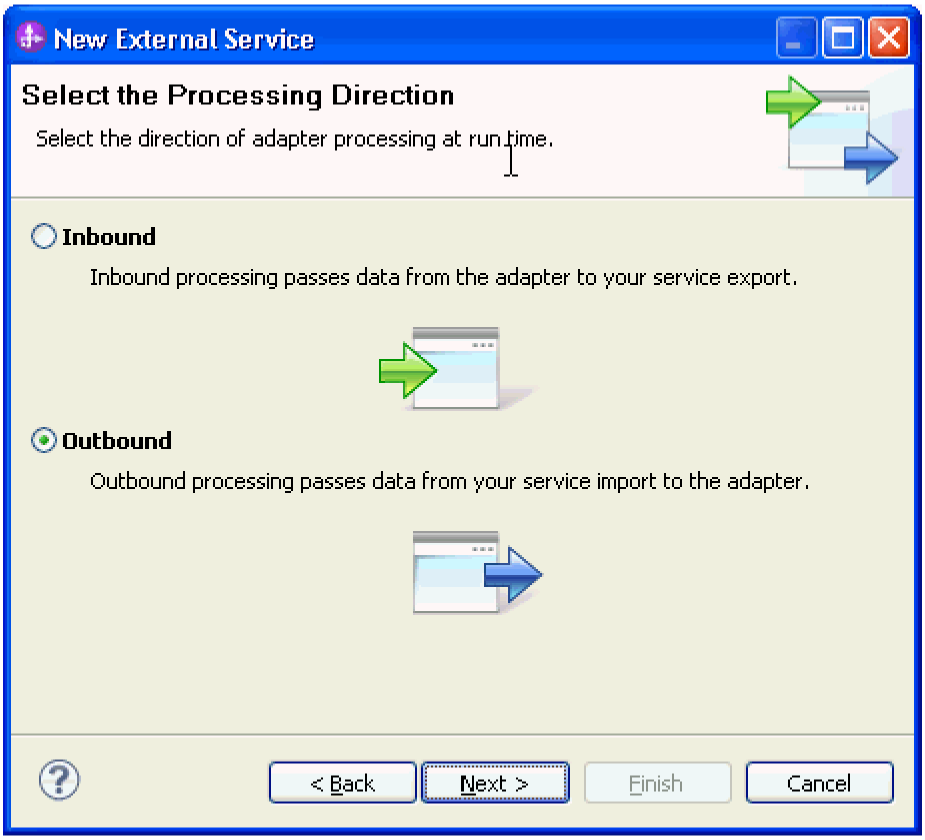
JDBC driver JAR files:

E:\Yuanjs\db2\db2jcc.jar
E:\Yuanjs\db2\db2jcc_license_cu.jar

Specify the location of the database-specific native system libraries required to access the database server. This is required if you are using a Type 2 driver implementation. The native libraries are shipped with the database server and client installations.

System libraries:

9. Select **Outbound** and click **Next**.



New External Service

Select the Processing Direction

Select the direction of adapter processing at run time.

○ **Inbound**

Inbound processing passes data from the adapter to your service export.

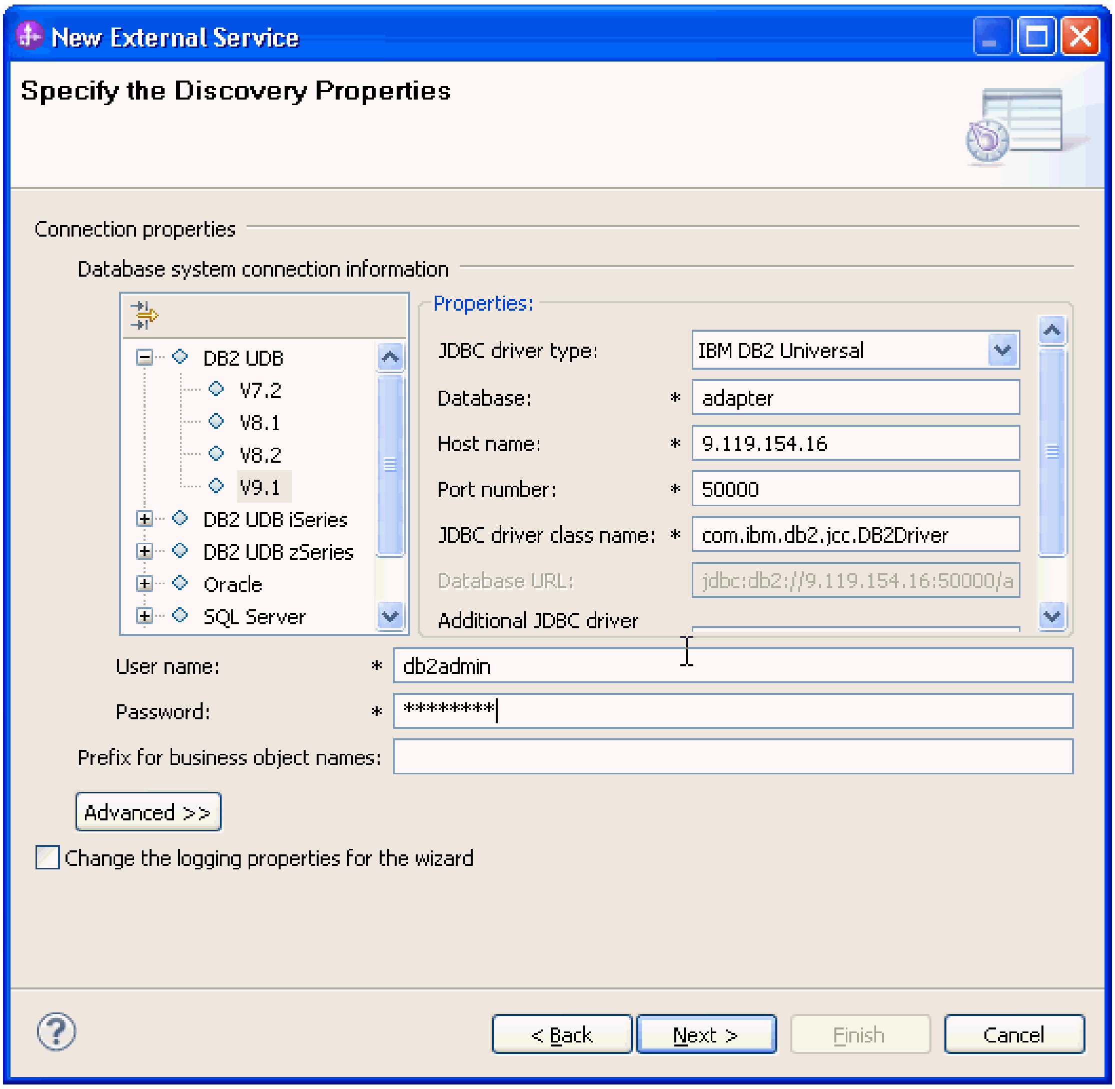
● **Outbound**

Outbound processing passes data from your service import to the adapter.

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**.



New External Service

Specify the Discovery Properties

Connection properties

Database system connection information

Properties:

JDBC driver type: IBM DB2 Universal

Database: * adapter

Host name: * 9.119.154.16

Port number: * 50000

JDBC driver class name: * com.ibm.db2.jcc.DB2Driver

Database URL: jdbc:db2://9.119.154.16:50000/a

Additional JDBC driver

User name: * db2admin

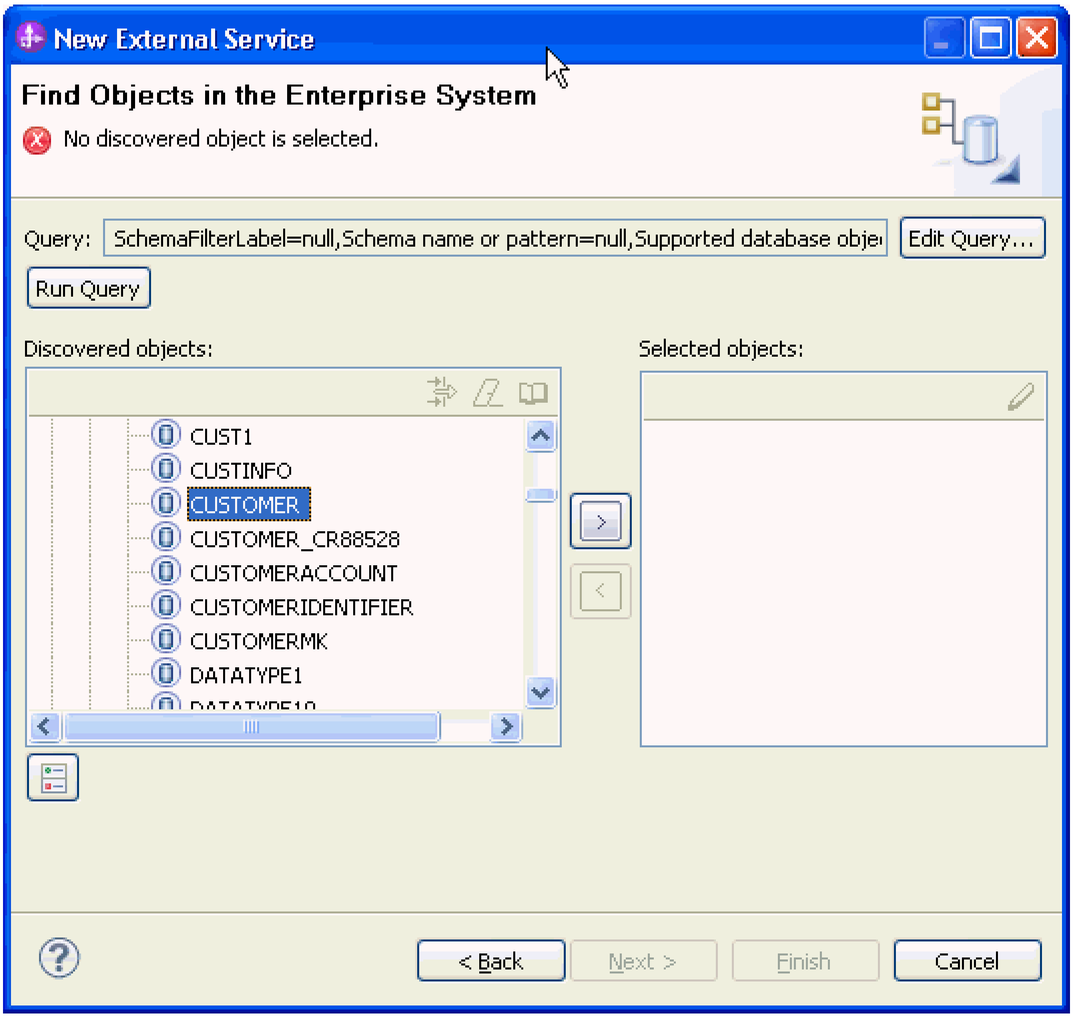
Password: * *********

Prefix for business object names:

☐ Change the logging properties for the wizard

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.



New External Service

Find Objects in the Enterprise System

No discovered object is selected.

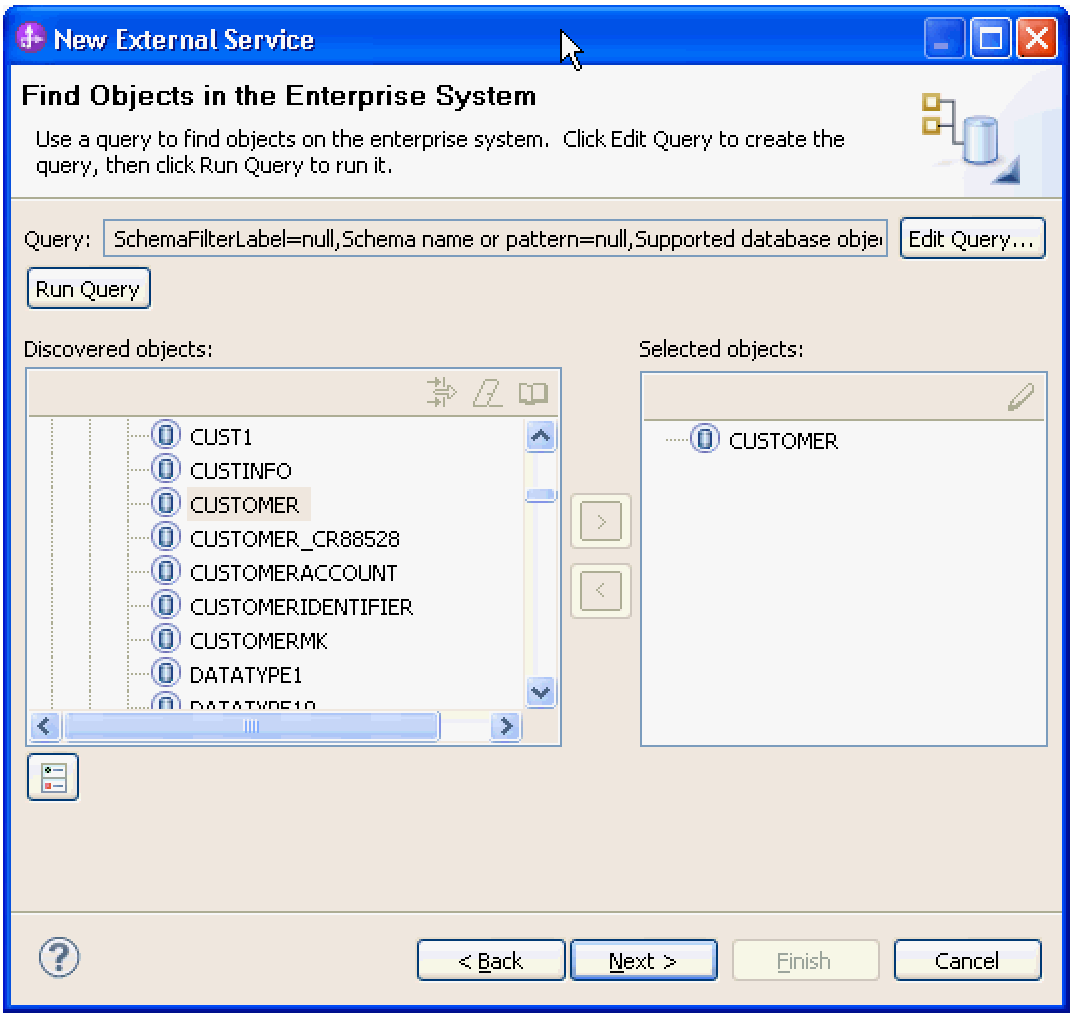
Query: SchemaFilterLabel=null,Schema name or pattern=null,Supported database obje

Discovered objects:

- CUST1
- CUSTINFO
- CUSTOMER
- CUSTOMER_CR88528
- CUSTOMERACCOUNT
- CUSTOMERIDENTIFIER
- CUSTOMERMK
- DATATYPE1
- DATATYPE10

Selected objects:

3. 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 obje

Discovered objects:

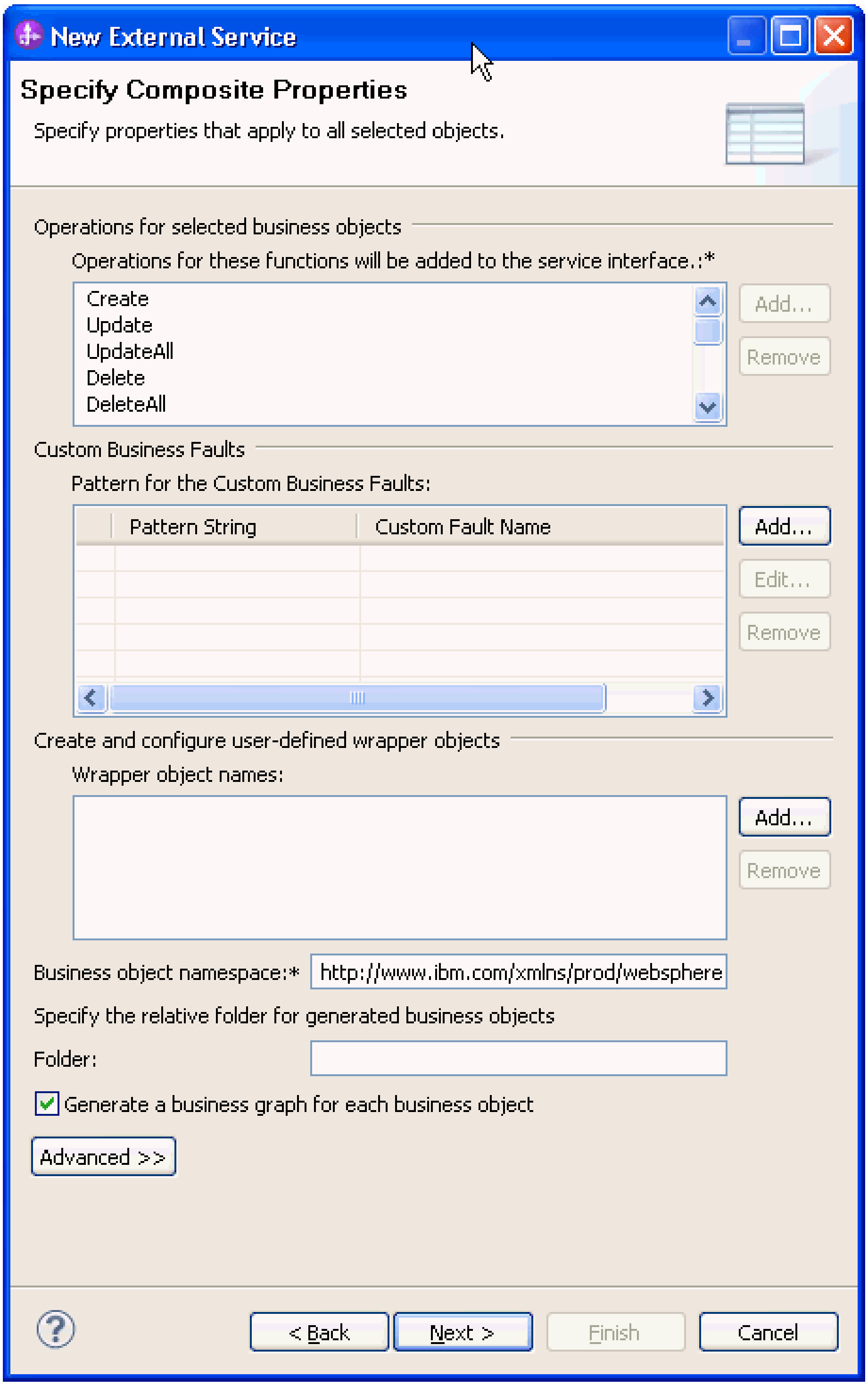
- CUST1
- CUSTINFO
- CUSTOMER
- CUSTOMER_CR88528
- CUSTOMERACCOUNT
- CUSTOMERIDENTIFIER
- CUSTOMERMK
- DATATYPE1
- DATATYPE10

Selected objects:

- CUSTOMER

Generate business object definitions and related artifacts

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



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
UpdateAll
Delete
DeleteAll

Custom Business Faults

Pattern for the Custom Business Faults:

Pattern String	Custom Fault Name

Create and configure user-defined wrapper objects

Wrapper object names:

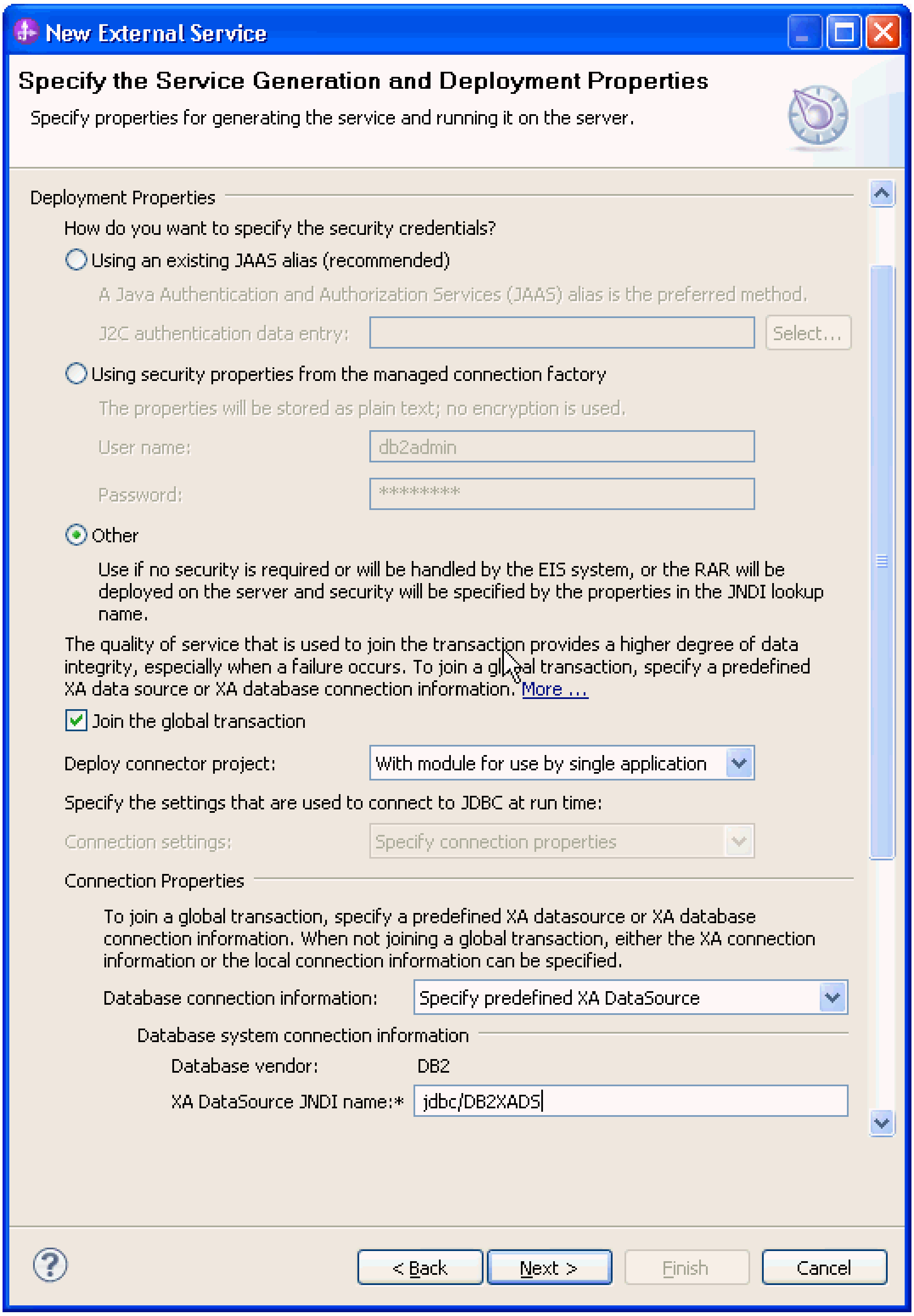
Business object namespace:* http://www.ibm.com/xmlns/prod/websphere

Specify the relative folder for generated business objects

Folder:

☑ Generate a business graph for each business object

2. Select the security credential type as **Other**. Specify the **XA DataSource JNDI name** property as **jdbc/DB2XADS**. Click **Next**.



Specify the Service Generation and Deployment Properties

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

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: Select...

○ Using security properties from the managed connection factory

The properties will be stored as plain text; no encryption is used.

User name: db2admin

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 join a global transaction, specify a predefined XA data source or XA database connection information. More ...

☑ Join the global transaction

Deploy connector project: With module for use by single application

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

Connection settings: Specify connection properties

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.

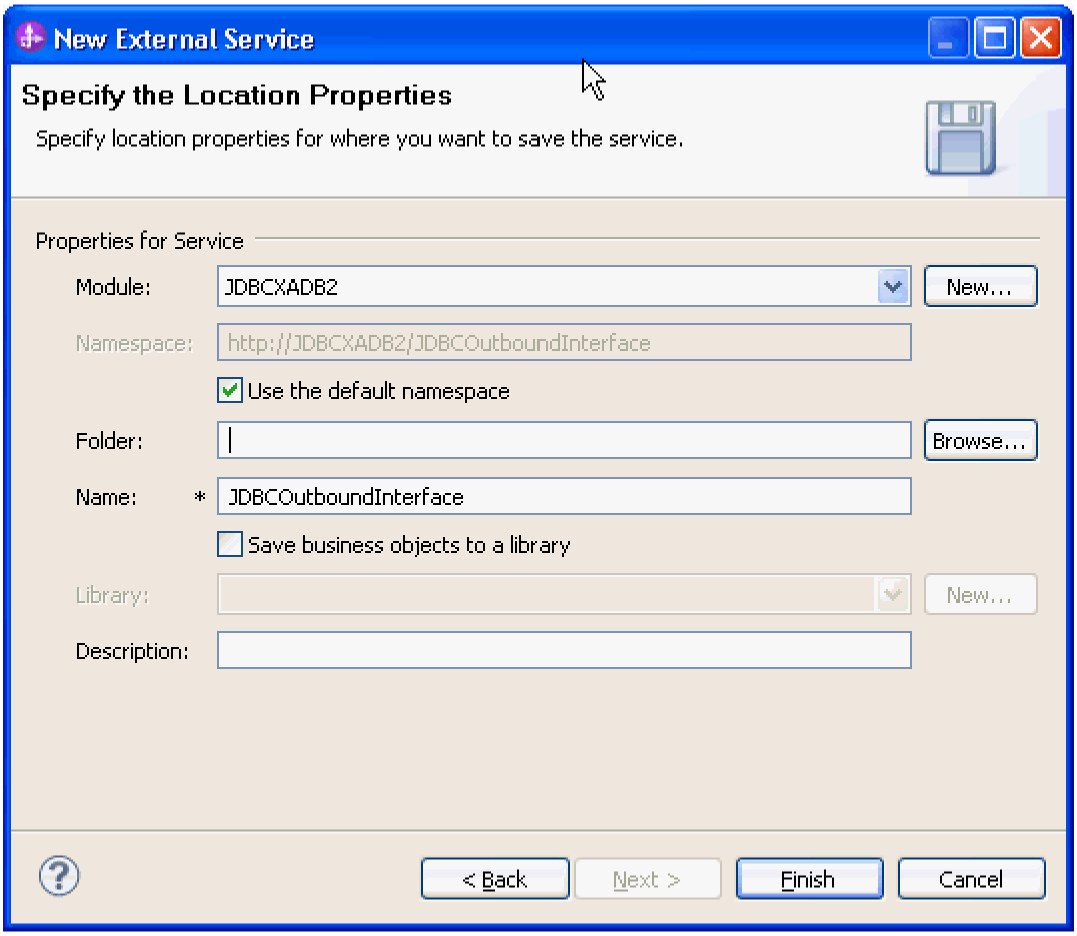
Database connection information: Specify predefined XA DataSource

Database system connection information

Database vendor: DB2

XA DataSource JNDI name:* jdbc/DB2XADS

3. Click **Finish**.



New External Service

Specify the Location Properties

Specify location properties for where you want to save the service.

Properties for Service

Module: JDBCXADB2

Namespace: http://JDBCXADB2/JDBCOutboundInterface

☑ Use the default namespace

Folder:

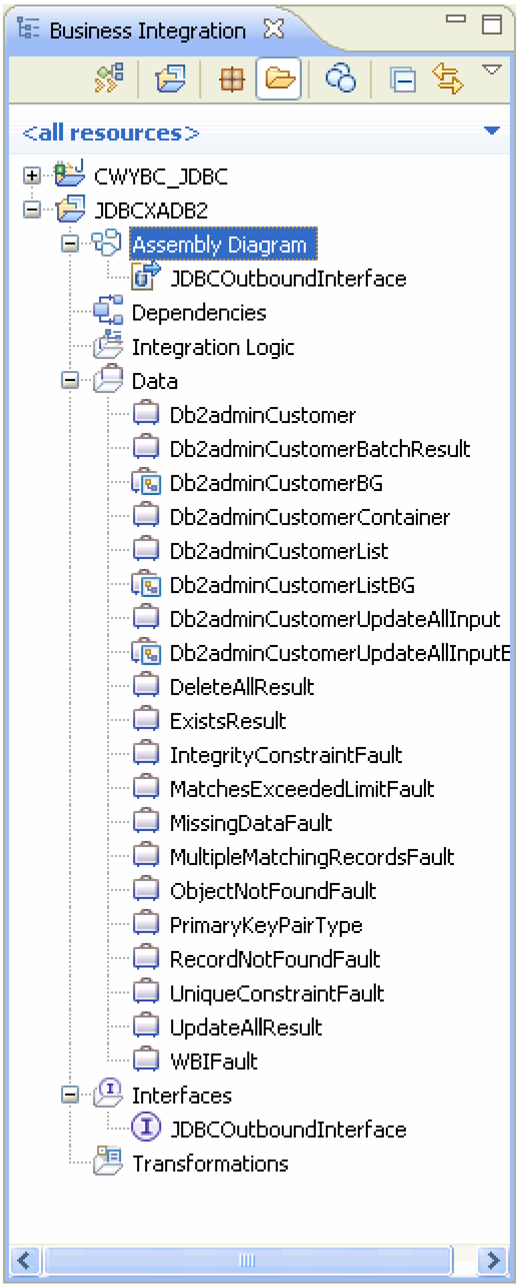
Name: * JDBCOutboundInterface

□ Save business objects to a library

Library:

Description:

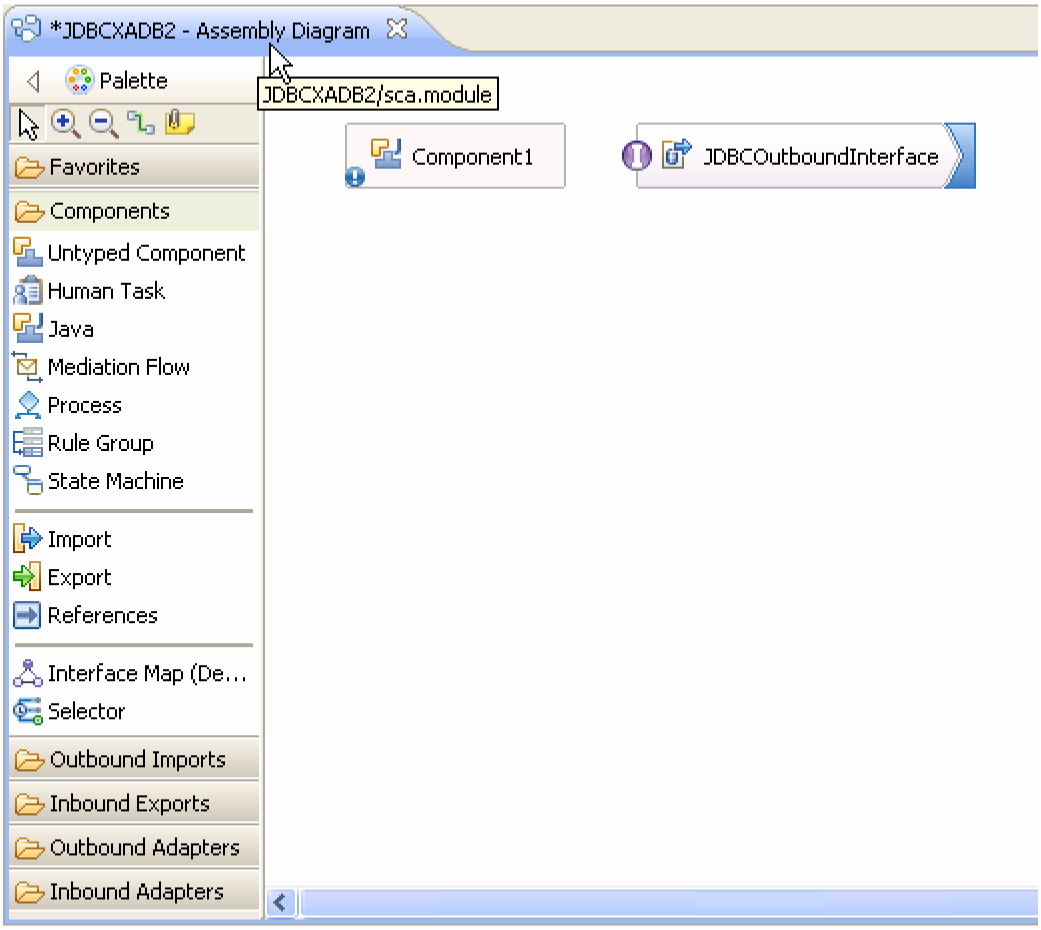
Verify the results shown below.



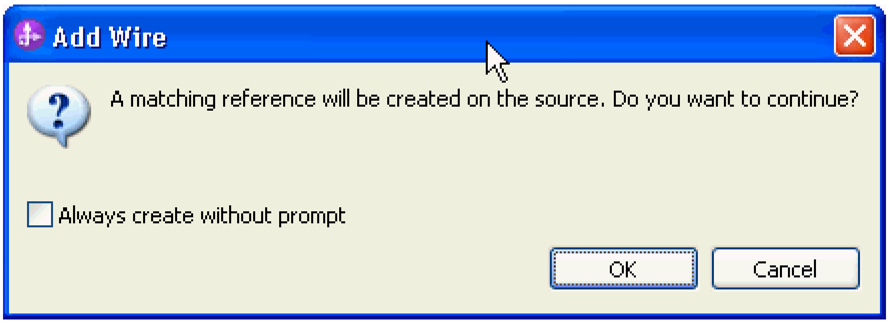
Set up the components to be part of the XA environment

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**.



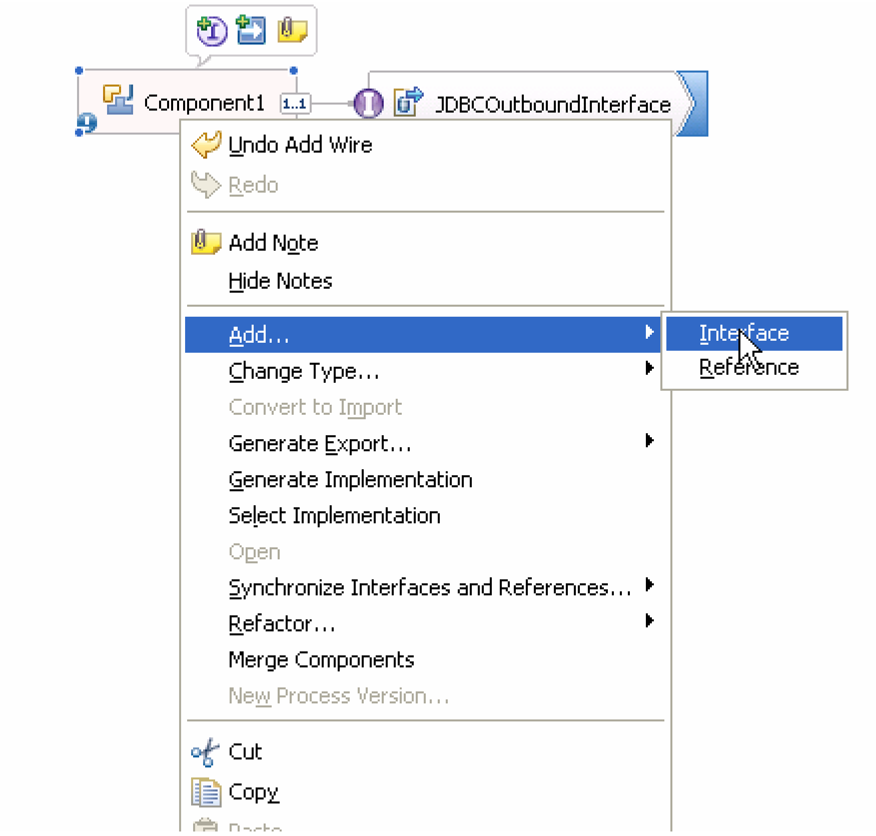
3. Wire **Component1** to **JDBCOutboundInterface**. In the Add Wire message window, click **OK**.



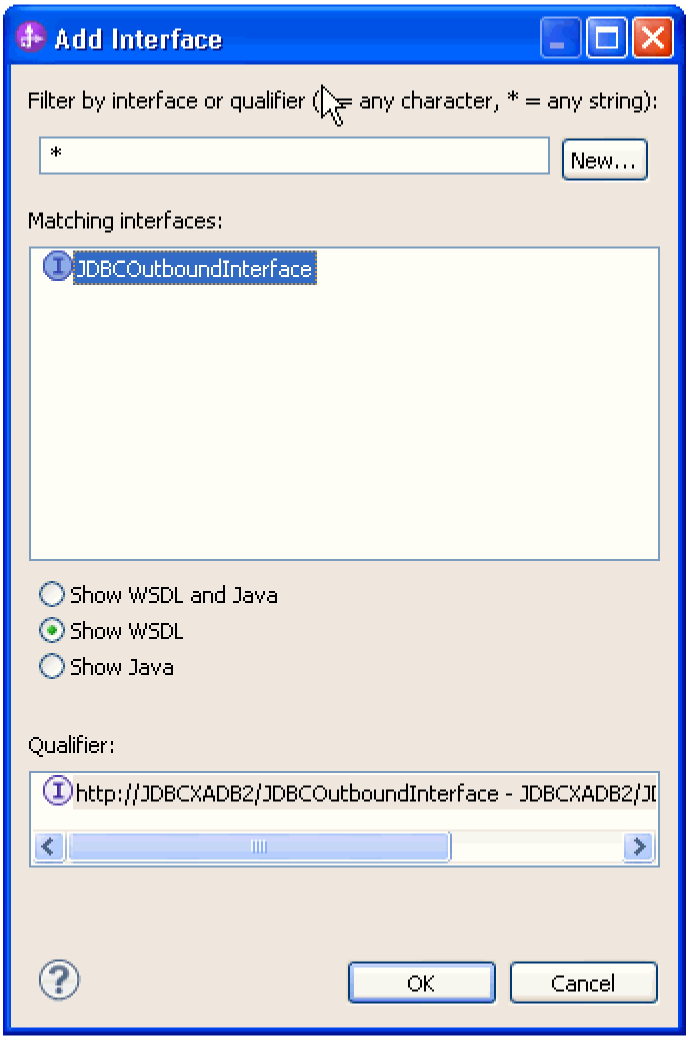
A matching reference will be created on the source. Do you want to continue?

□ Always create without prompt

4. Right-click **Component1** and select **Add > Interface**.



5. In the Add Interface window, click **New**.



Add Interface

Filter by interface or qualifier (? = any character, * = any string):

*

Matching interfaces:

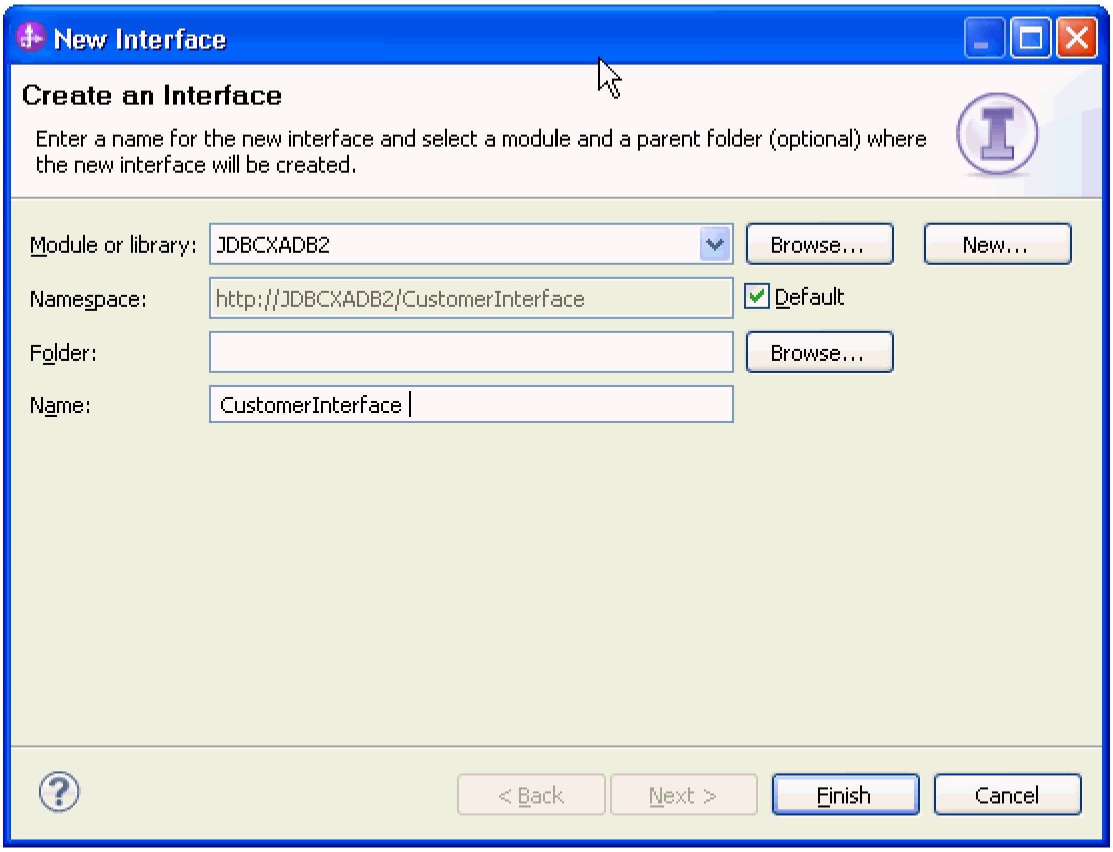
JDBCOutboundInterface

○ Show WSDL and Java
● Show WSDL
○ Show Java

Qualifier:

http://JDBCXADB2/JDBCOutboundInterface - JDBCXADB2/JI

6. Enter **CustomerInterface** in the **Name** field. Click **Finish**.



Create an Interface

Enter a name for the new interface and select a module and a parent folder (optional) where the new interface will be created.

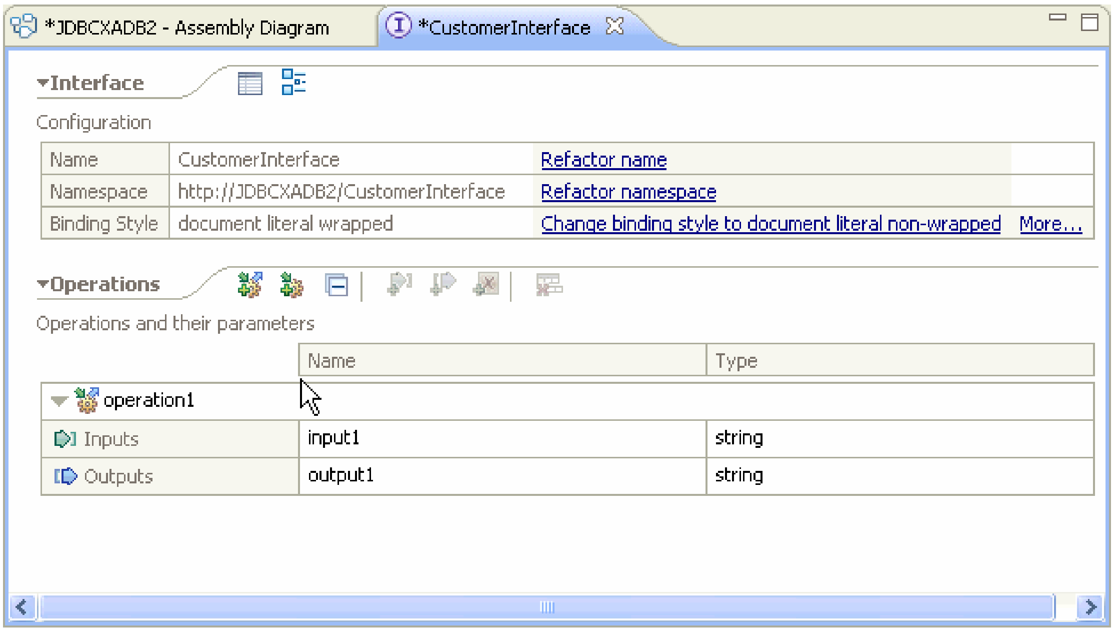
Module or library: JDBCXADB2

Namespace: http://JDBCXADB2/CustomerInterface ☑ Default

Folder:

Name: CustomerInterface

7. Click ✓ to add a new operation for **CustomerInterface** interface.



*JDBCXADB2 - Assembly Diagram | *CustomerInterface

Interface

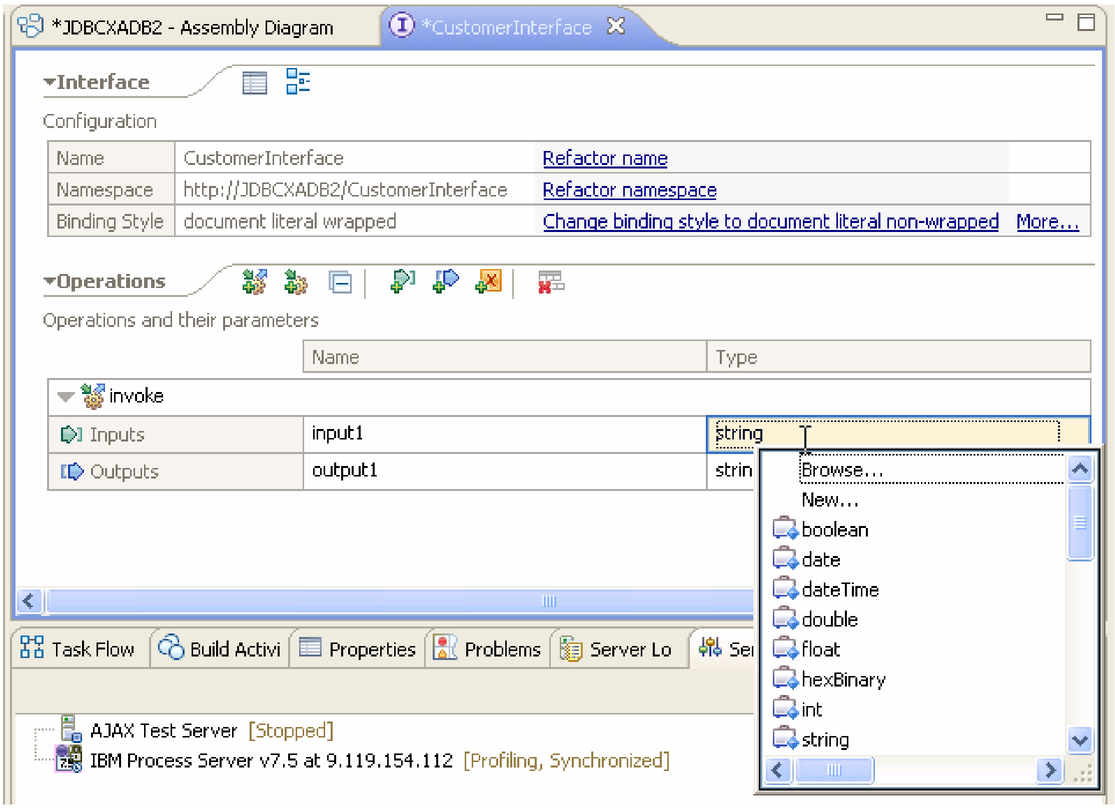
Configuration

Name	CustomerInterface	Refactor name
Namespace	http://JDBCXADB2/CustomerInterface	Refactor namespace
Binding Style	document literal wrapped	Change binding style to document literal non-wrapped More...

Operations

Operations and their parameters

	Name	Type
operation1		
Inputs	input1	string
Outputs	output1	string



*JDBCXADB2 - Assembly Diagram | *CustomerInterface

Interface

Configuration

Name	CustomerInterface	Refactor name
Namespace	http://JDBCXADB2/CustomerInterface	Refactor namespace
Binding Style	document literal wrapped	Change binding style to document literal non-wrapped More...

Operations

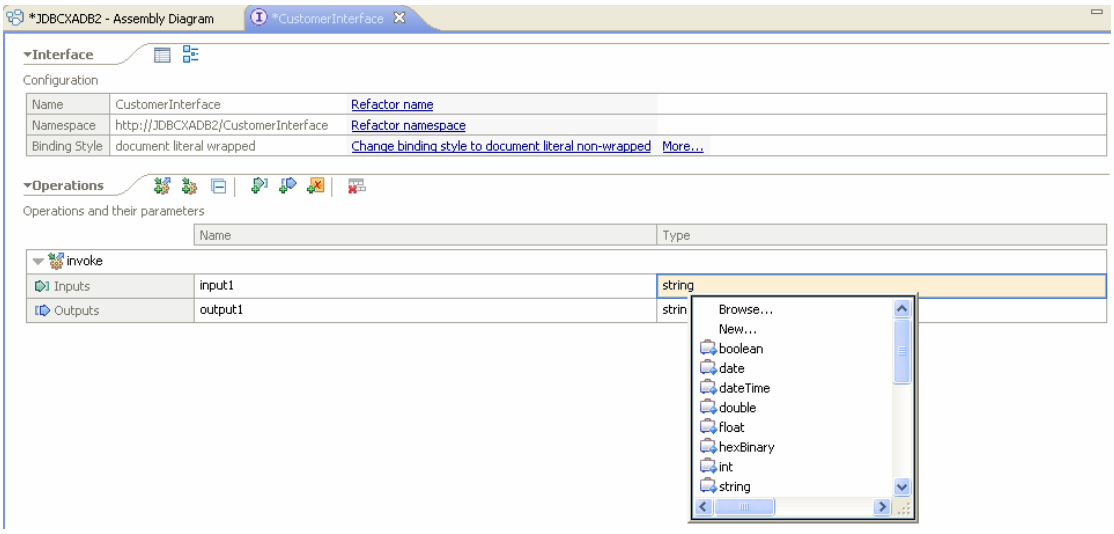
Operations and their parameters

	Name	Type
invoke		
Inputs	input1	string
Outputs	output1	strin

AJAX Test Server [Stopped]

IBM Process Server v7.5 at 9.119.154.112 [Profiling, Synchronized]

8. Rename the operation name to **invoke**. Click Type for Inputs parameter, and select **New.**


*JDBCXADB2 - Assembly Diagram
*CustomerInterface

Interface

Configuration

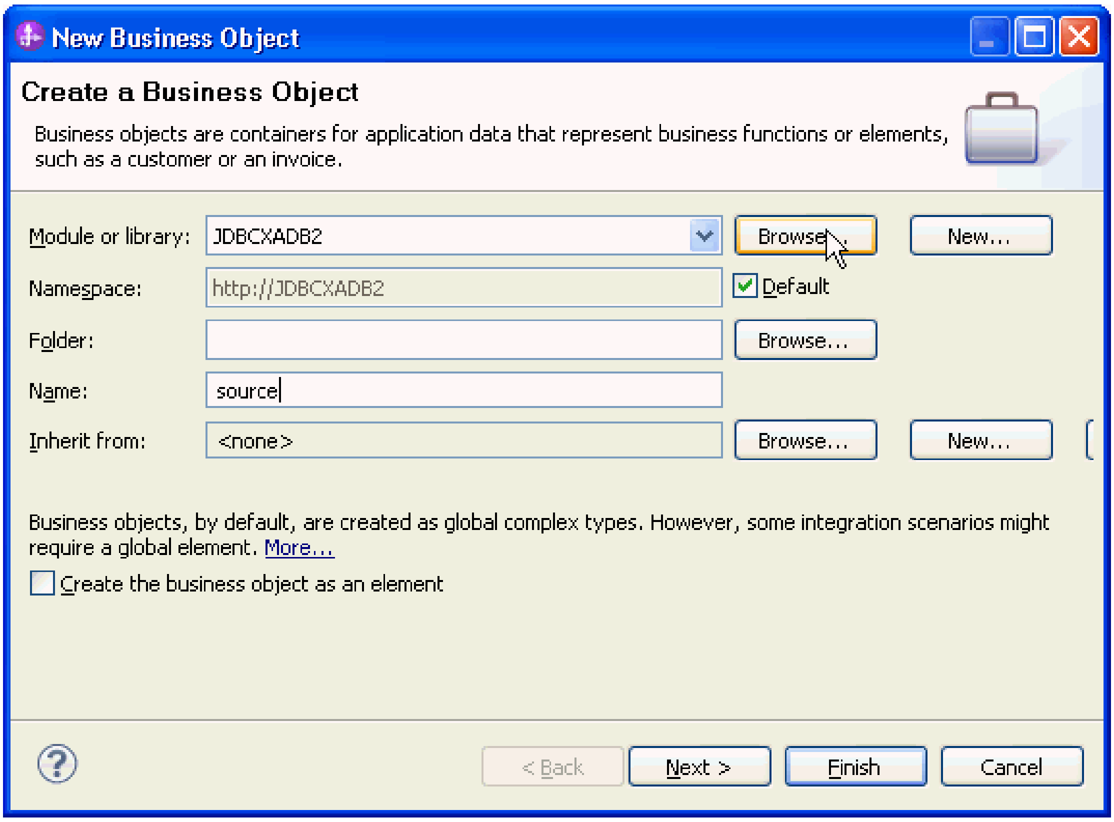
Name	CustomerInterface	Refactor name
Namespace	http://JDBCXADB2/CustomerInterface	Refactor namespace
Binding Style	document literal wrapped	Change binding style to document literal non-wrapped More...

Operations

Operations and their parameters

	Name	Type
invoke		
Inputs	input1	string
Outputs	output1	strin

9. In the **New Business Object** window, enter **Source** in the **Name** field. Click **Next**.


New Business Object

Create a Business Object

Business objects are containers for application data that represent business functions or elements, such as a customer or an invoice.

Module or library: JDBCXADB2

Namespace: http://JDBCXADB2 ☑ Default

Folder:

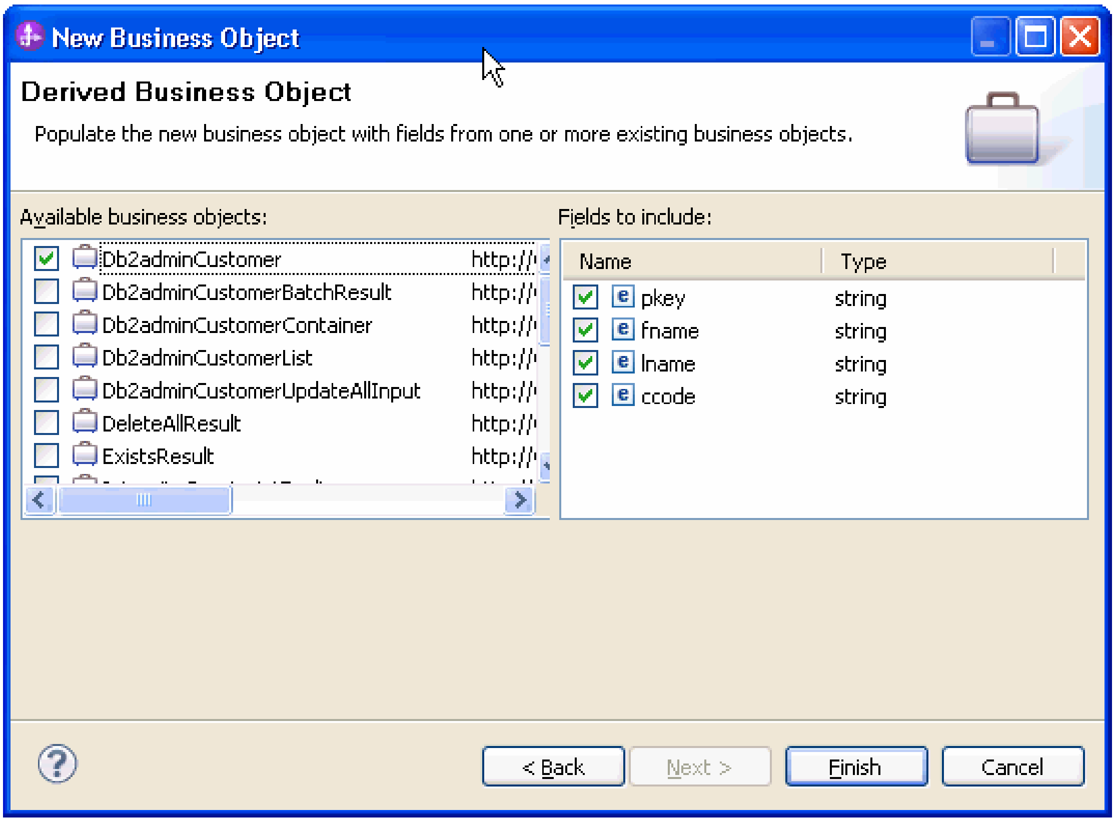
Name: source

Inherit from: <none>

Business objects, by default, are created as global complex types. However, some integration scenarios might require a global element. More...

□ Create the business object as an element

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**.



New Business Object

Derived Business Object

Populate the new business object with fields from one or more existing business objects.

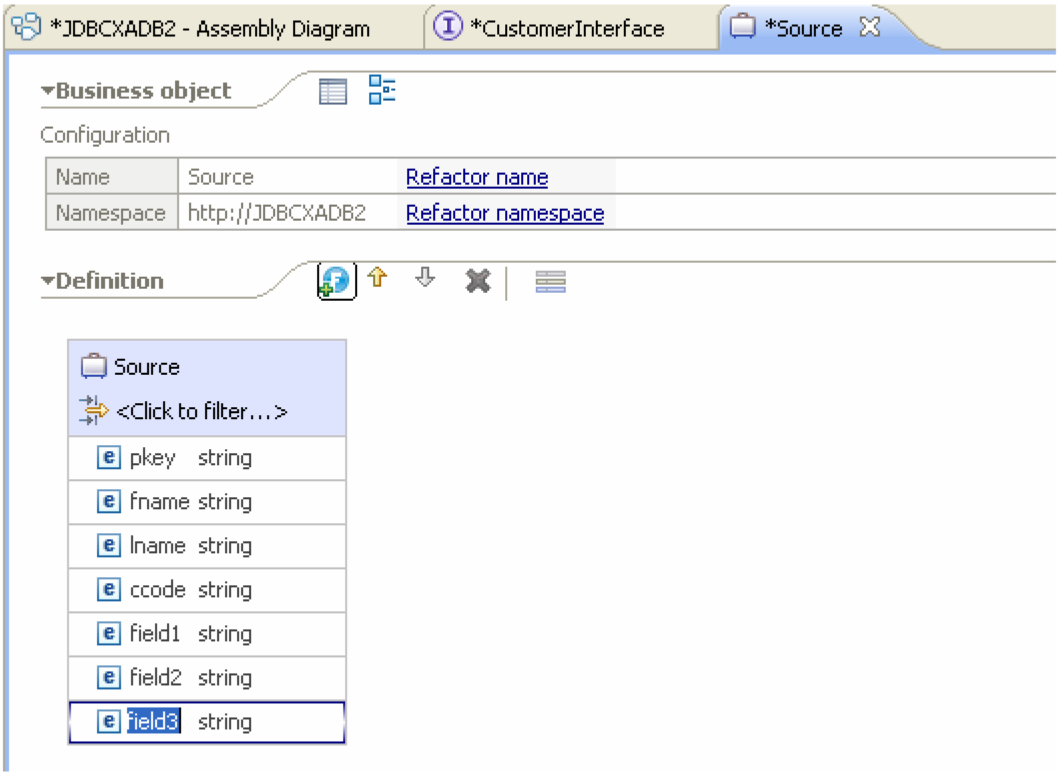
Available business objects:

- ☑ Db2adminCustomer http://
- □ Db2adminCustomerBatchResult http://
- □ Db2adminCustomerContainer http://
- □ Db2adminCustomerList http://
- □ Db2adminCustomerUpdateAllInput http://
- □ DeleteAllResult http://
- □ ExistsResult http://

Fields to include:

	Name	Type
☑	pkey	string
☑	fname	string
☑	lname	string
☑	ccode	string

11. In the Business Object editor, click to add three new fields for **Source** business object.



Business object

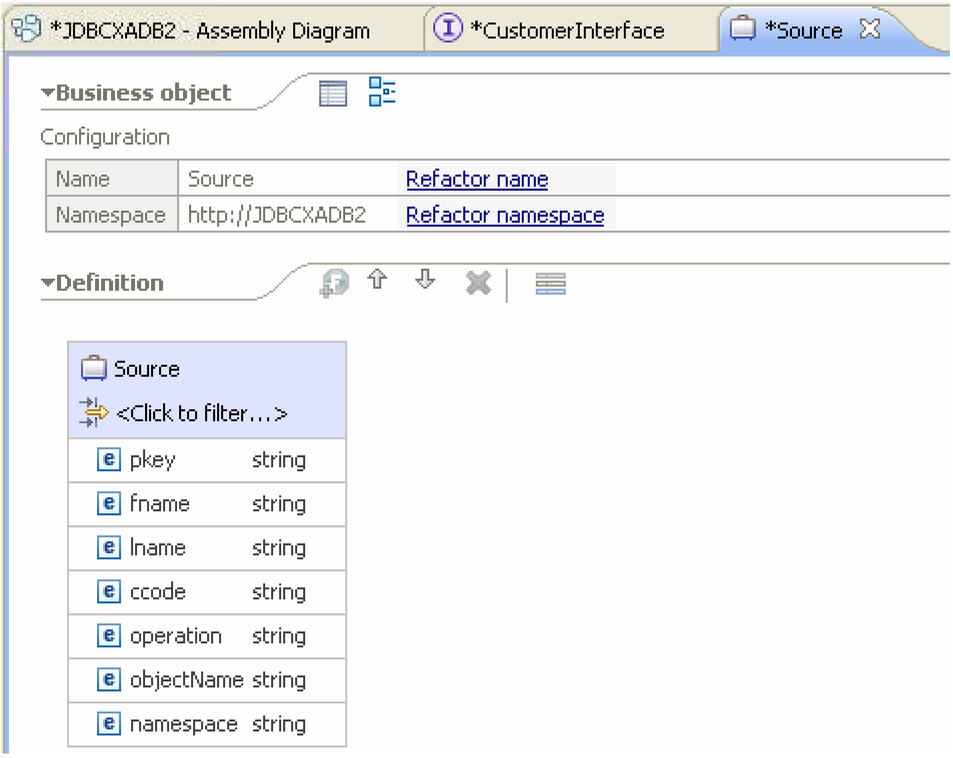
Configuration

Name	Source	Refactor name
Namespace	http://JDBCXADB2	Refactor namespace

Definition

Source	
<Click to filter…>	
pkey	string
fname	string
lname	string
ccode	string
field1	string
field2	string
field3	string

12. Then rename these three new fields as **operation**, **objectName** and **namespace**.



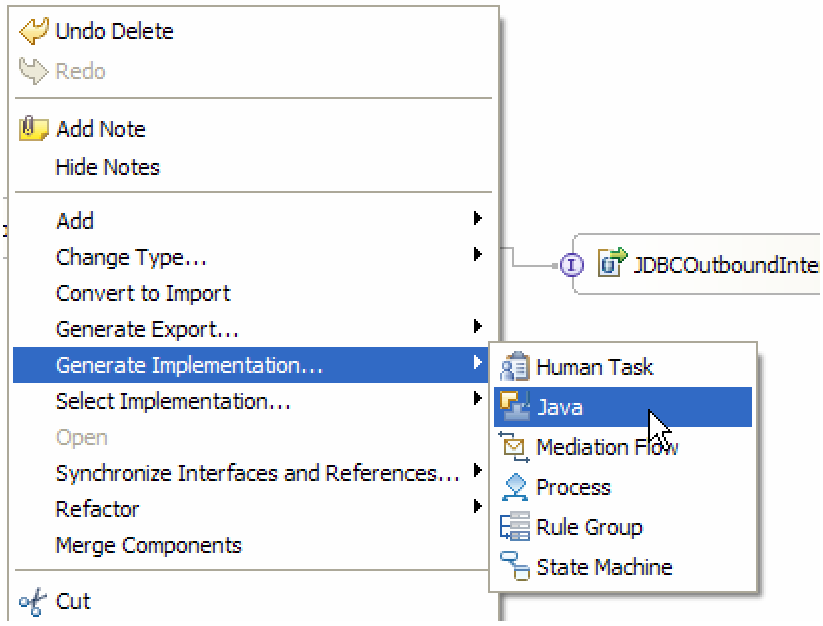
Name	Source	Refactor name
Namespace	http://JDBCXADB2	Refactor namespace

Source
<Click to filter...>

pkey	string
fname	string
lname	string
ccode	string
operation	string
objectName	string
namespace	string

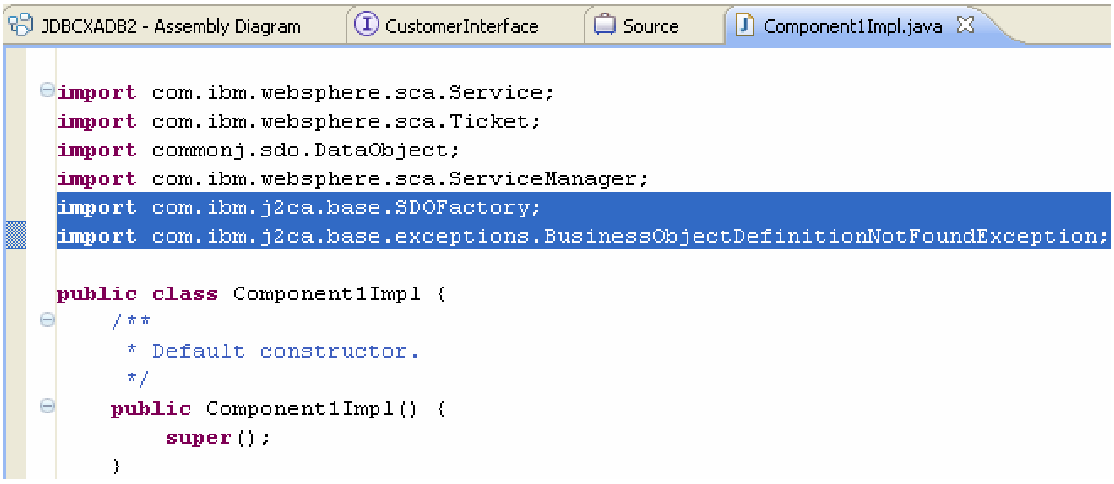
13. Select **File->Save All** to save all the changes.

14. Right click on **Component1** in the Assembly Diagram and select **Generate implementation... -> Java.**



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.**

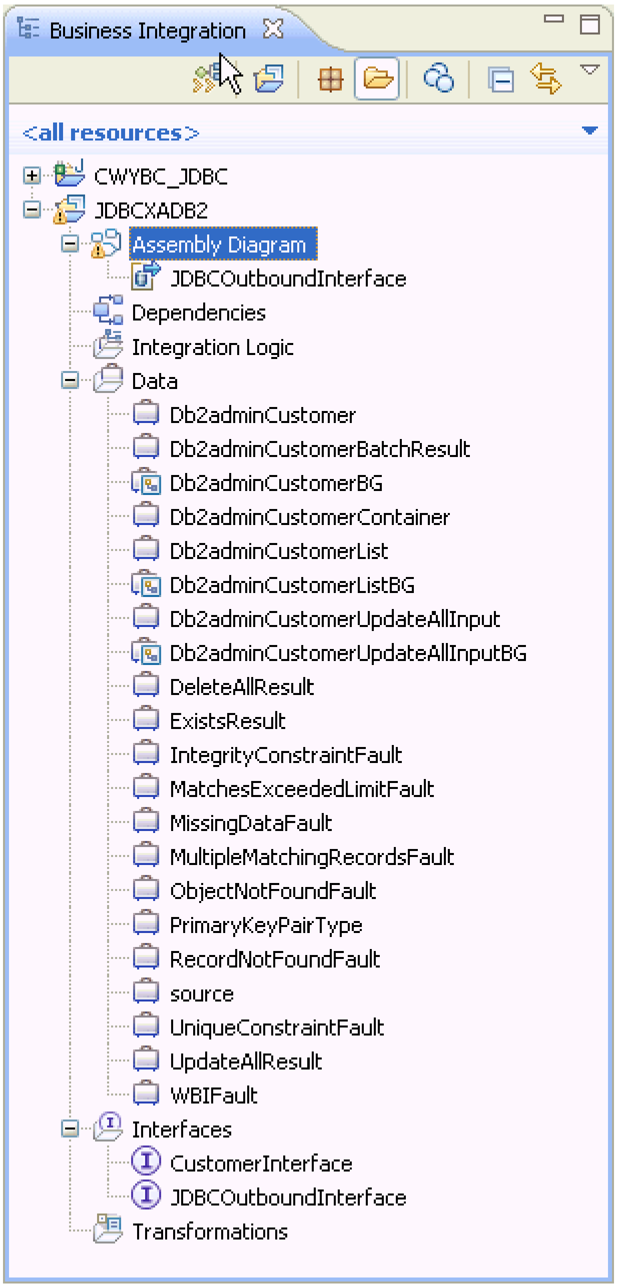


```
import com.ibm.websphere.sca.Service;
import com.ibm.websphere.sca.Ticket;
import commonj.sdo.DataObject;
import com.ibm.websphere.sca.ServiceManager;
import com.ibm.j2ca.base.SDOFactory;
import com.ibm.j2ca.base.exceptions.BusinessObjectDefinitionNotFoundException;

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.



Business Integration

<all resources>

- CWYBC_JDBC
- JDBCXADB2
  - Assembly Diagram
    - JDBCOutboundInterface
  - Dependencies
  - Integration Logic
  - Data
    - Db2adminCustomer
    - Db2adminCustomerBatchResult
    - Db2adminCustomerBG
    - Db2adminCustomerContainer
    - Db2adminCustomerList
    - Db2adminCustomerListBG
    - Db2adminCustomerUpdateAllInput
    - Db2adminCustomerUpdateAllInputBG
    - DeleteAllResult
    - ExistsResult
    - IntegrityConstraintFault
    - MatchesExceededLimitFault
    - MissingDataFault
    - MultipleMatchingRecordsFault
    - ObjectNotFoundFault
    - PrimaryKeyPairType
    - RecordNotFoundFault
    - source
    - UniqueConstraintFault
    - UpdateAllResult
    - WBIFault
  - Interfaces
    - CustomerInterface
    - JDBCOutboundInterface
  - Transformations

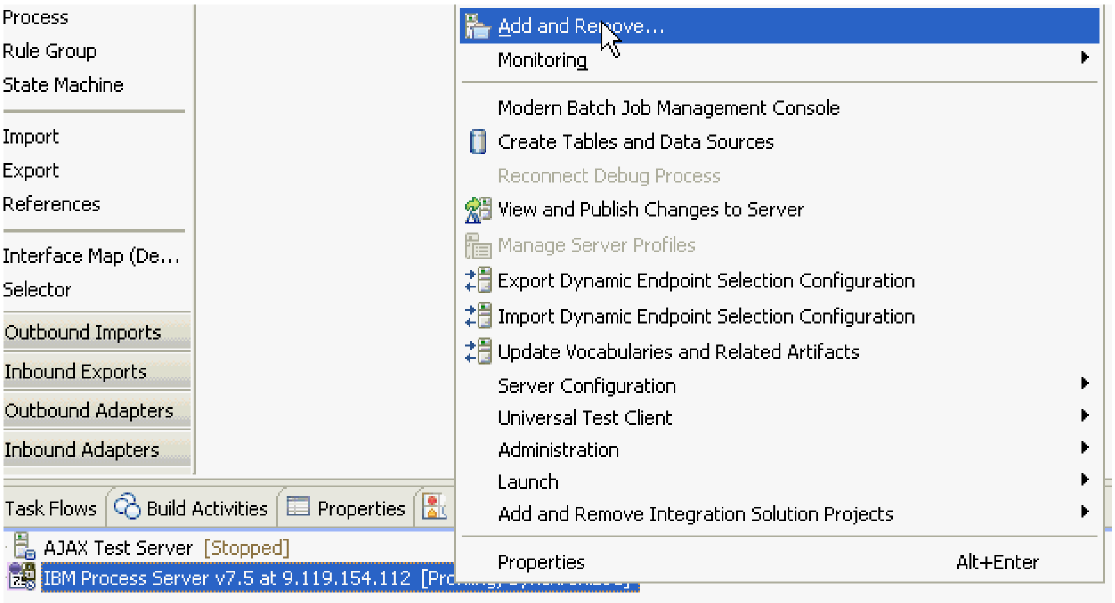
Deploy the module to the test environment

The result of running the external service wizard is an SCA module that contains an Enterprise Information System import. Install this SCA module in IBM Integration Designer integration test client. To do this, you must add the SCA module you created earlier to the server using the **Servers** view in IBM Integration Designer.

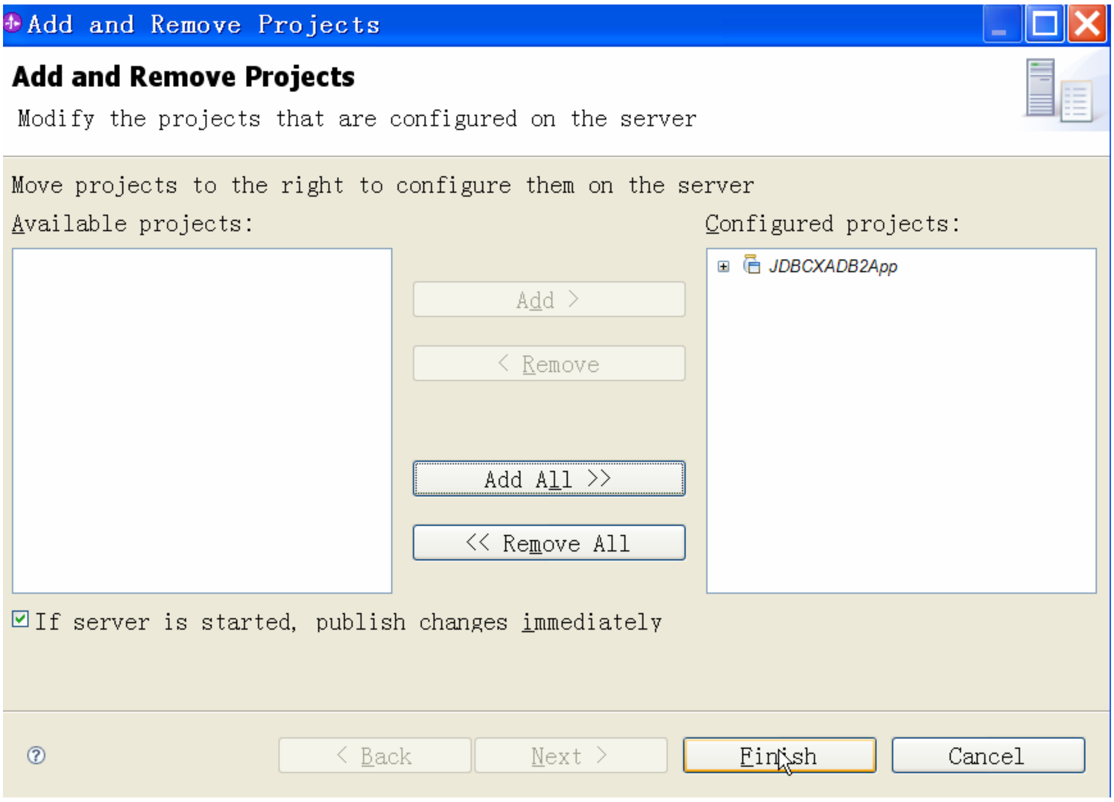
Steps for adding the SCA module to the server:

1. In IBM Integration Designer, switch to the **Servers** view by selecting **Windows > Show View > Servers**.
2. In the Servers tab in the lower-right pane of the IBM Integration Designer screen, right-click the server, and select **Start.**

3. After the server is started, right-click the server, and select **Add and Remove projects**.


Process
Rule Group
State Machine
Import
Export
References
Interface Map (De...
Selector
Outbound Imports
Inbound Exports
Outbound Adapters
Inbound Adapters
Task Flows
Build Activities
Properties
AJAX Test Server [Stopped]
IBM Process Server v7.5 at 9.119.154.112 [Pro
Add and Remove...
Monitoring
Modern Batch Job Management Console
Create Tables and Data Sources
Reconnect Debug Process
View and Publish Changes to Server
Manage Server Profiles
Export Dynamic Endpoint Selection Configuration
Import Dynamic Endpoint Selection Configuration
Update Vocabularies and Related Artifacts
Server Configuration
Universal Test Client
Administration
Launch
Add and Remove Integration Solution Projects
Properties
Alt+Enter

4. 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.


Add and Remove Projects

Add and Remove Projects

Modify the projects that are configured on the server

Move projects to the right to configure them on the server

Available projects:

Configured projects:

JDBCXADB2App

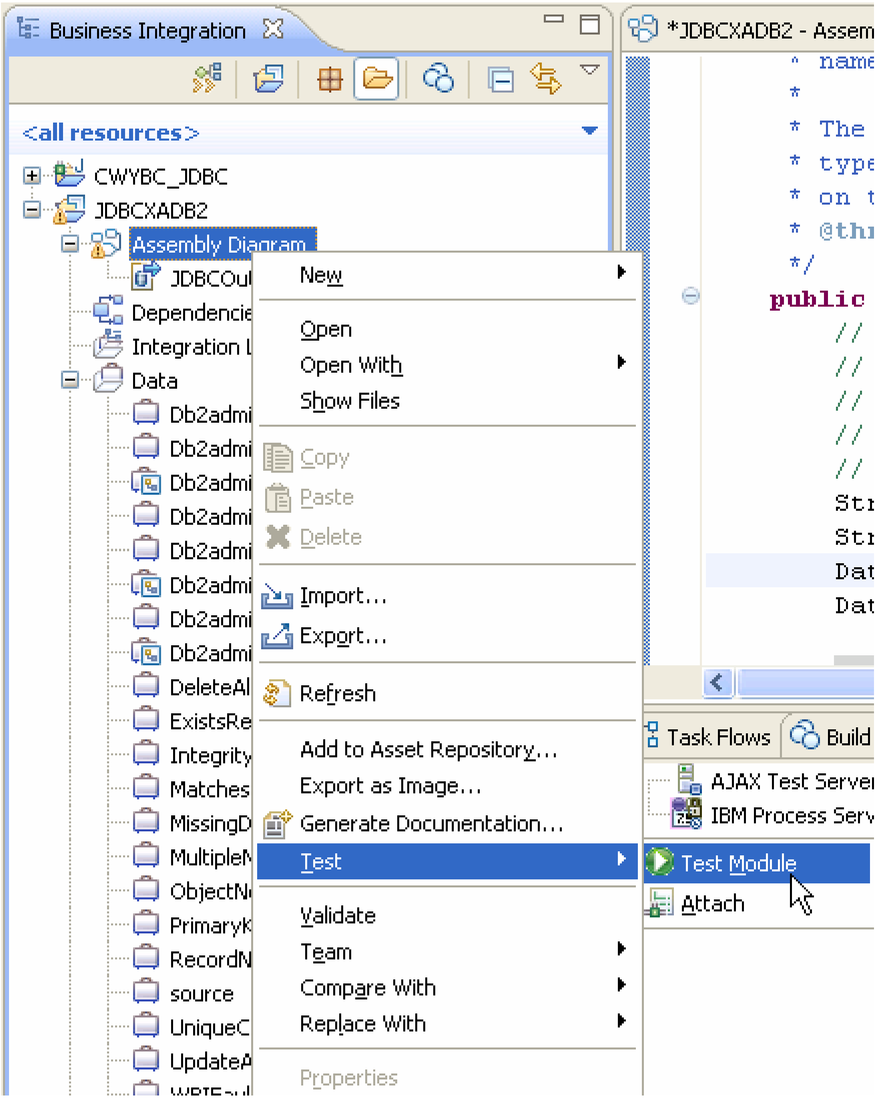
☑ If server is started, publish changes immediately

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.

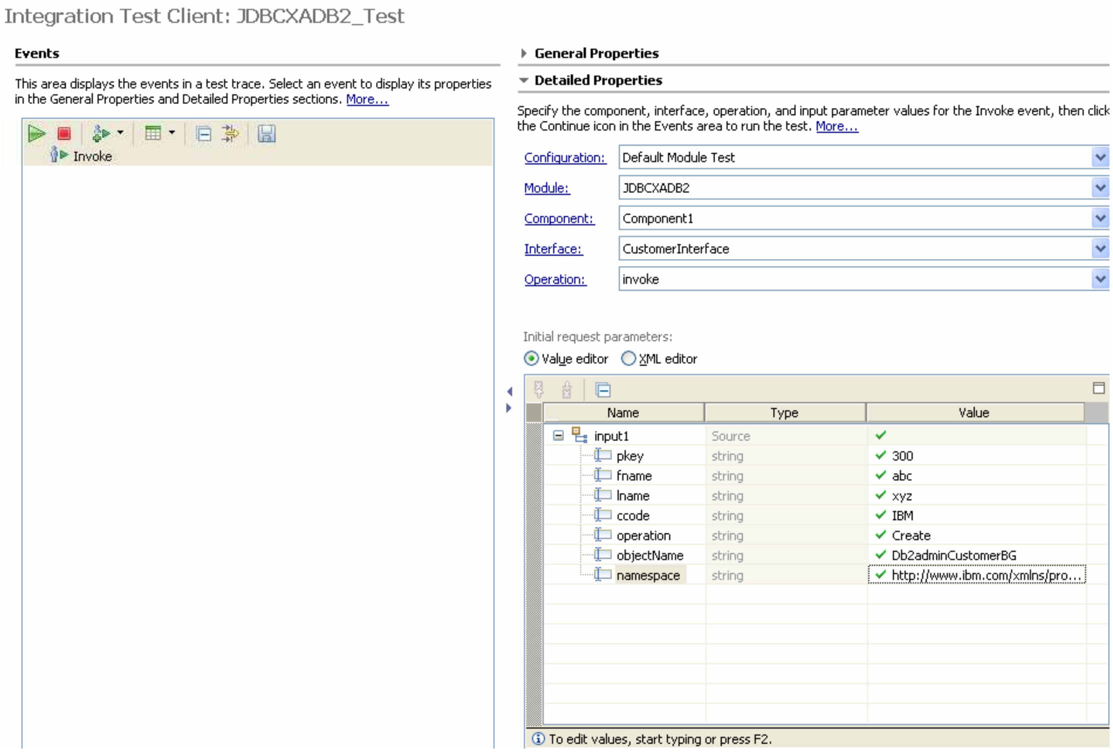
Test the assembled adapter application

Test the assembled adapter application using the IBM Integration Designer integration test client:

1. From the Business Integration view, right click on **JDBCXADB2** and select **Test > Test Module.**



2. From the **Component** list, select **Component1**. Specify the parameters as shown in the figure below.



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:

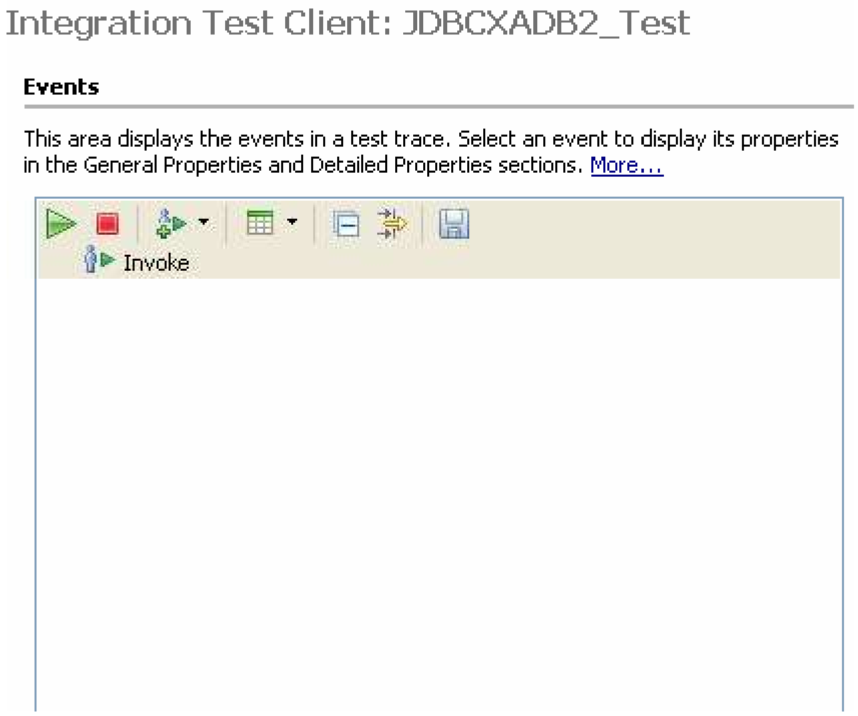
● Value editor ○ XML editor

Name	Type	Value
input1	Source	✓
pkey	string	✓ 300
fname	string	✓ abc
lname	string	✓ xyz
ccode	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/db2admincustomerbg**.

3. Click ▶ to continue.



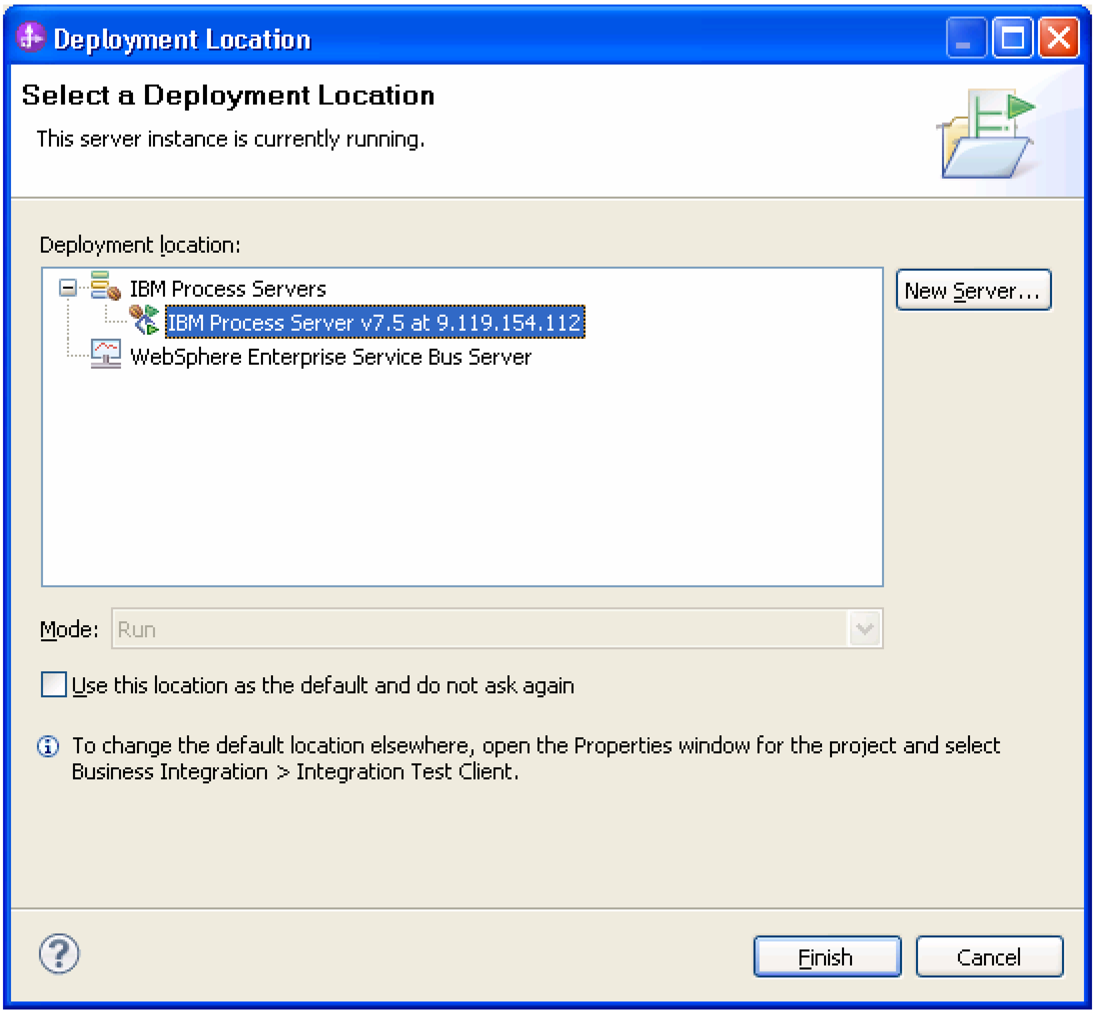
Integration Test Client: JDBCXADB2_Test

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**.



Deployment Location

Select a Deployment Location

This server instance is currently running.

Deployment location:

- IBM Process Servers
  - IBM Process Server v7.5 at 9.119.154.112
- WebSphere Enterprise Service Bus Server

Mode: Run

□ Use this location as the default and do not ask again

To change the default location elsewhere, open the Properties window for the project and select Business Integration > Integration Test Client.

5. 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';
```

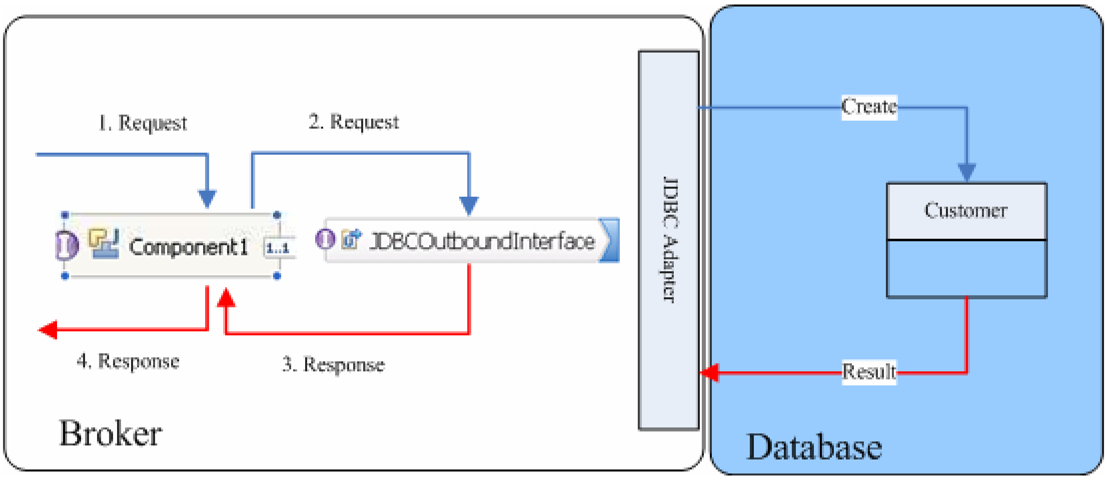
Chapter 9. Tutorial 8: Sending data to the Oracle database with XA transaction (outbound processing)

This scenario demonstrates how WebSphere Adapter for JDBC 7.5.0.0 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:

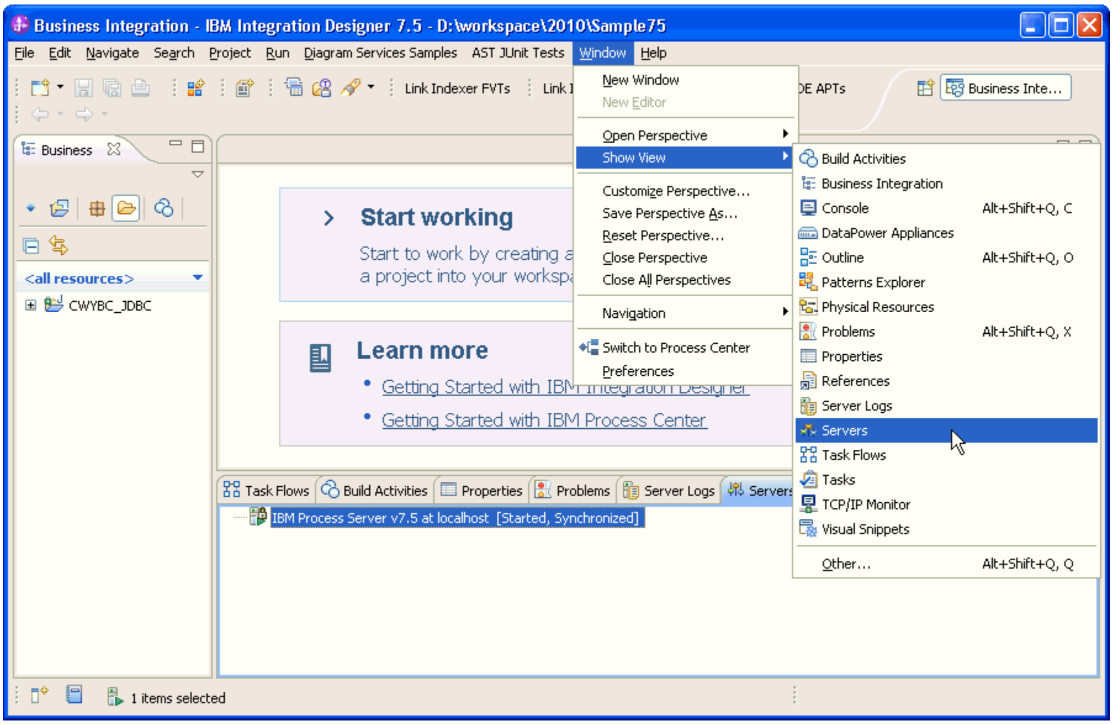
```
CREATE TABLE CUSTOMER  (
     PKEY VARCHAR2(10) NOT NULL PRIMARY KEY,
     FNAME VARCHAR2(20) ,
     LNAME VARCHAR2(20) ,
     CCODE VARCHAR2(10) ) ;
```

Create an authentication alias

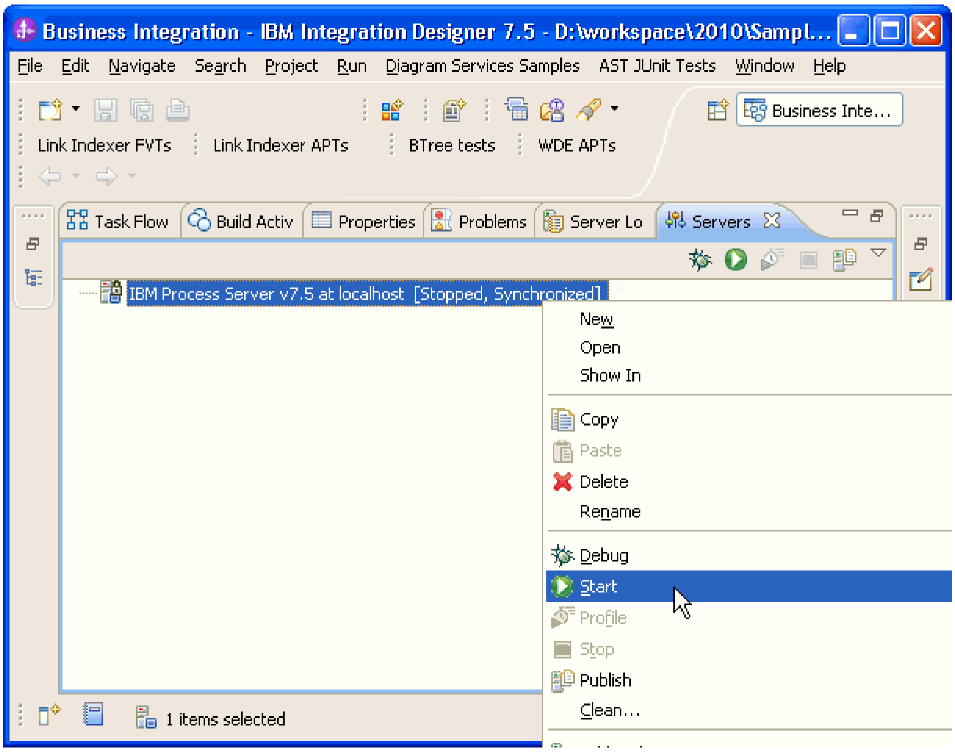
The authentication alias needs to be set because the data source created in the next section uses the username and password set in the authentication alias to connect to the database.

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

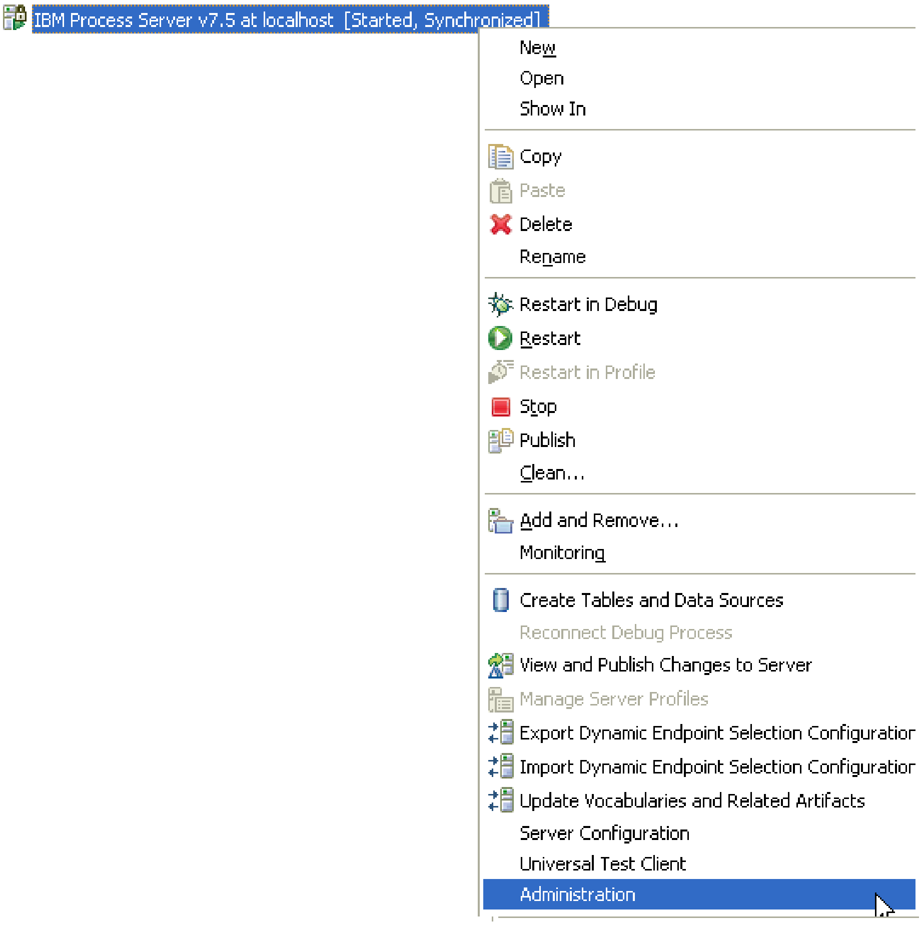
1. In IBM Integration Designer, switch to the **Servers** view by selecting **Windows > Show View > Servers**.



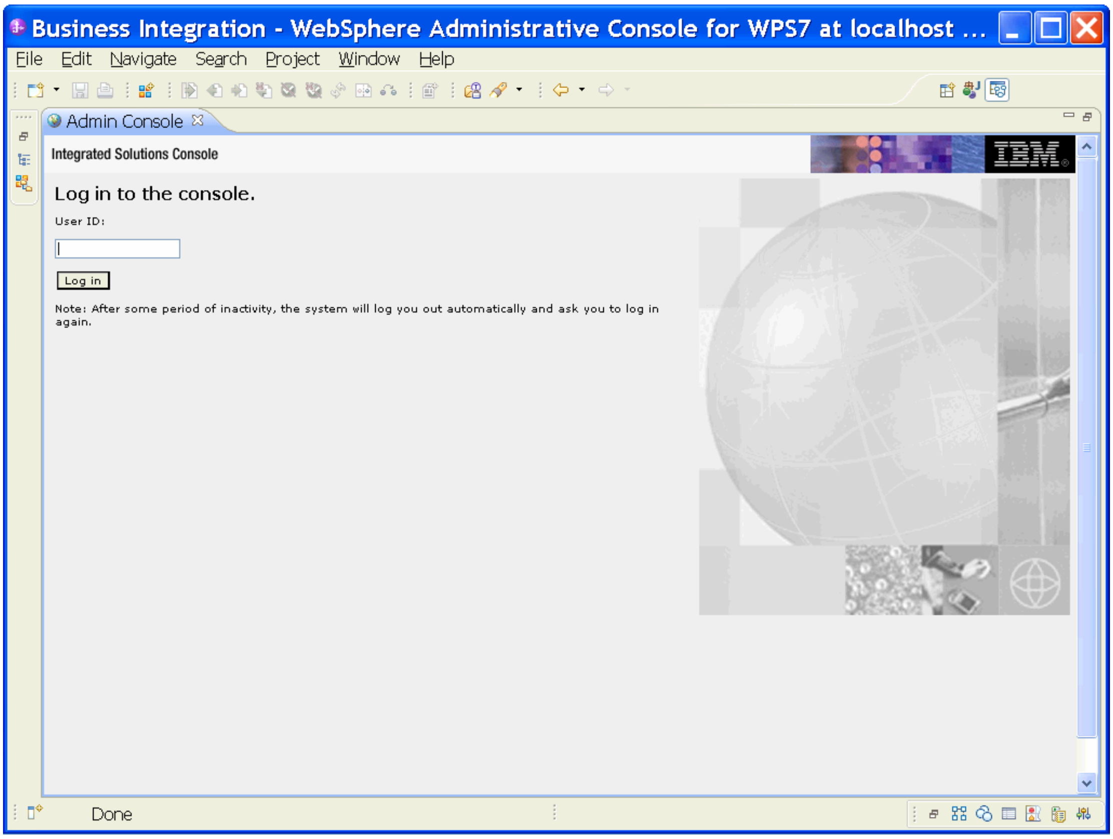
2. In the **Servers** view, right-click the server that you want to start and select **Start**.


Business Integration - IBM Integration Designer 7.5 - D:\workspace\2010\Sampl...
File Edit Navigate Search Project Run Diagram Services Samples AST JUnit Tests Window Help
Business Inte...
Link Indexer FVTs Link Indexer APTs BTree tests WDE APTs
Task Flow Build Activ Properties Problems Server Lo Servers
IBM Process Server v7.5 at localhost [Stopped, Synchronized]
New
Open
Show In
Copy
Paste
Delete
Rename
Debug
Start
Profile
Stop
Publish
Clean...
1 items selected

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


IBM Process Server v7.5 at localhost [Started, Synchronized]
New
Open
Show In
Copy
Paste
Delete
Rename
Restart in Debug
Restart
Restart in Profile
Stop
Publish
Clean...
Add and Remove...
Monitoring
Create Tables and Data Sources
Reconnect Debug Process
View and Publish Changes to Server
Manage Server Profiles
Export Dynamic Endpoint Selection Configuratior
Import Dynamic Endpoint Selection Configuratior
Update Vocabularies and Related Artifacts
Server Configuration
Universal Test Client
Administration

4. Log on to the administrative console.

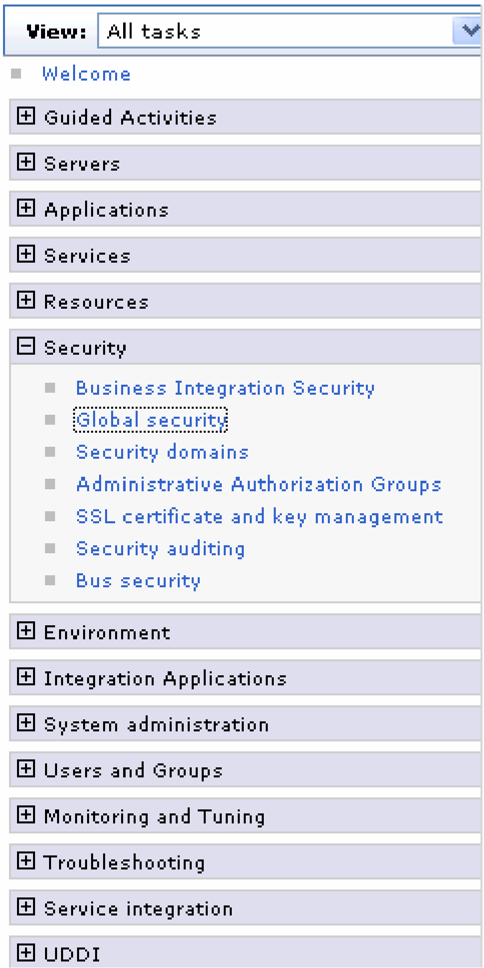

Business Integration - WebSphere Administrative Console for WPS7 at localhost ...
File Edit Navigate Search Project Window Help
Admin Console
Integrated Solutions Console
IBM
Log in
Done

Log in to the console.

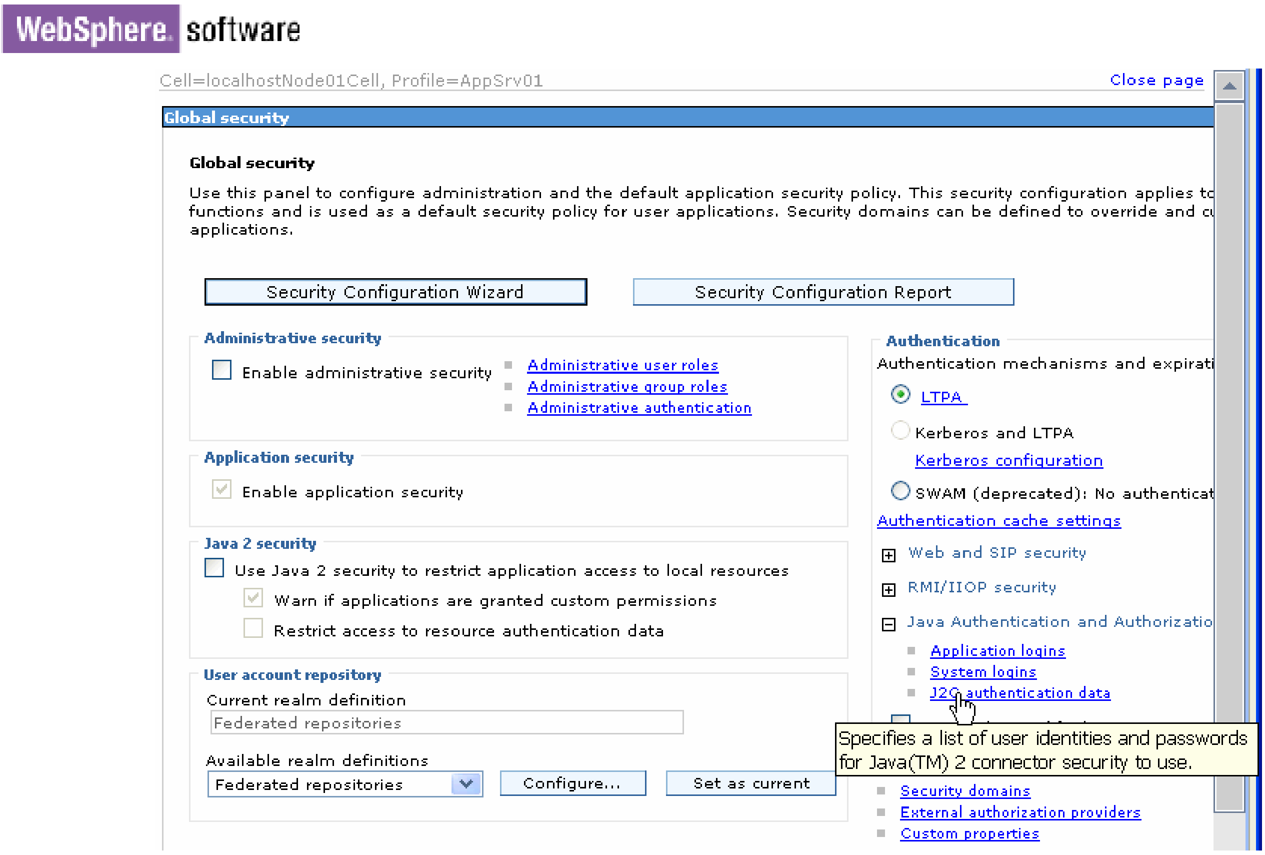
User ID:

Note: After some period of inactivity, the system will log you out automatically and ask you to log in again.

5. Click **Security → Global security**



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



WebSphere software

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 cu 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 expirati

● LTPA

○ Kerberos and LTPA

Kerberos configuration

○ SWAM (deprecated): No authenticat

Authentication cache settings

⊞ Web and SIP security

⊞ RMI/IIOP security

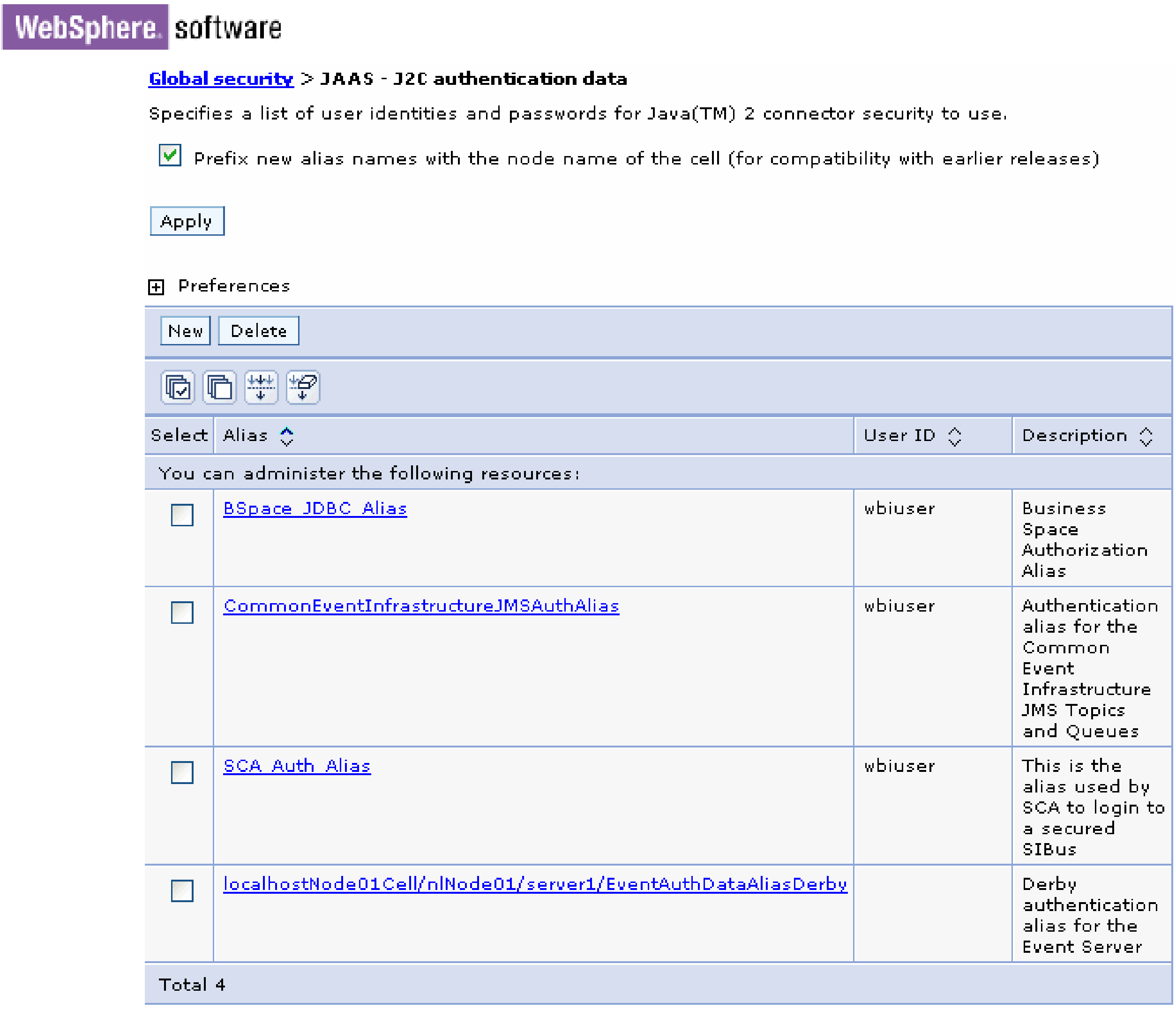
⊟ Java Authentication and Authorizatio

- 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

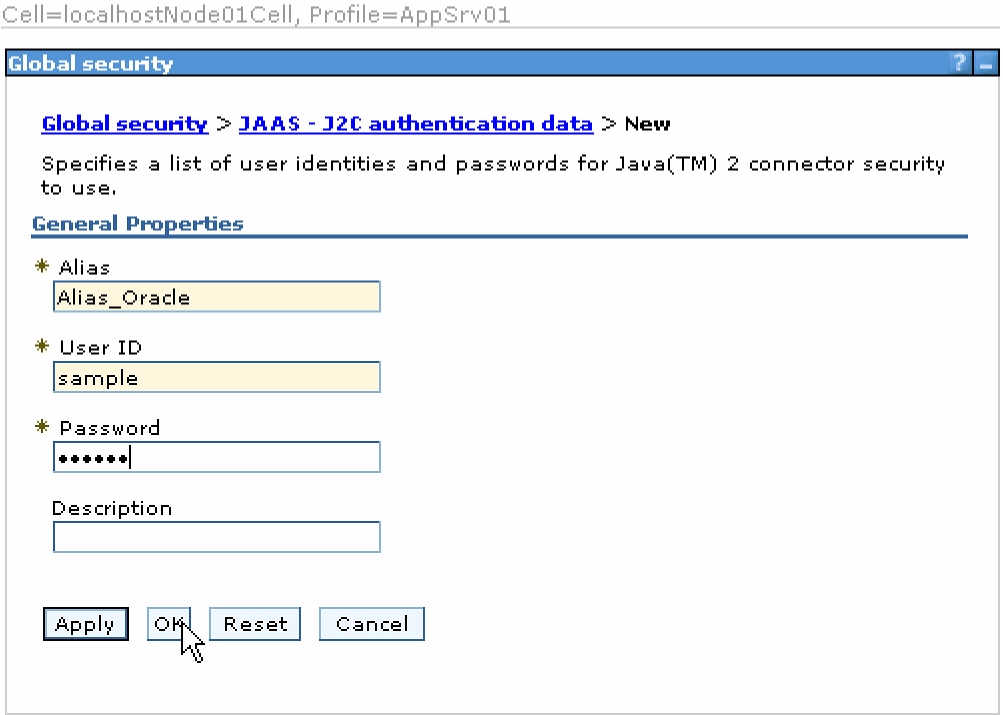
Global security > JAAS - J2C authentication data

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)

Select	Alias	User ID	Description
You can administer the following resources:			
□	BSpace_JDBC_Alias	wbiuser	Business Space Authorization Alias
□	CommonEventInfrastructureJMSAuthAlias	wbiuser	Authentication alias for the Common Event Infrastructure JMS Topics and Queues
□	SCA_Auth_Alias	wbiuser	This is the alias used by SCA to login to a secured SIBus
□	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, as shown in the figure. 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

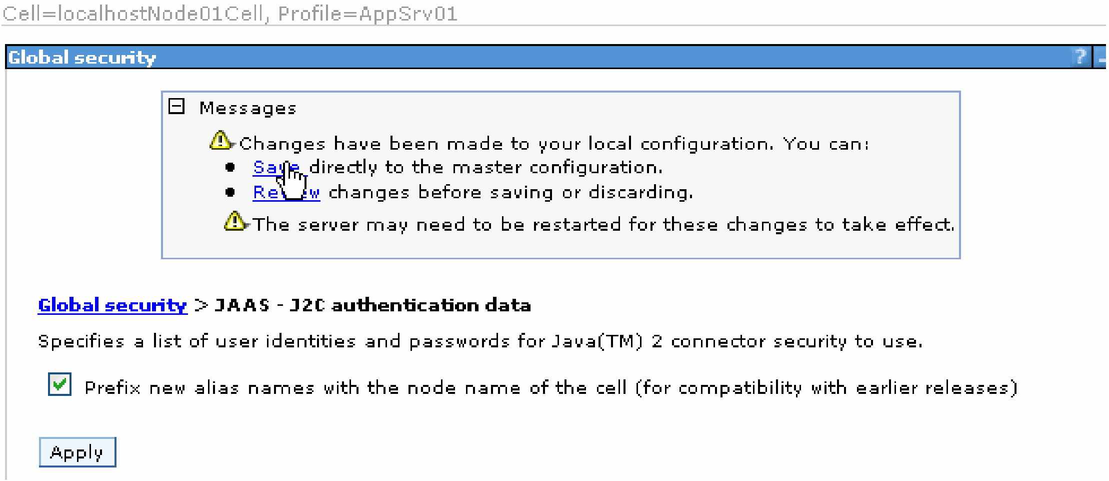
* Alias
Alias_Oracle

* User ID
sample

* Password
••••••

Description

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.

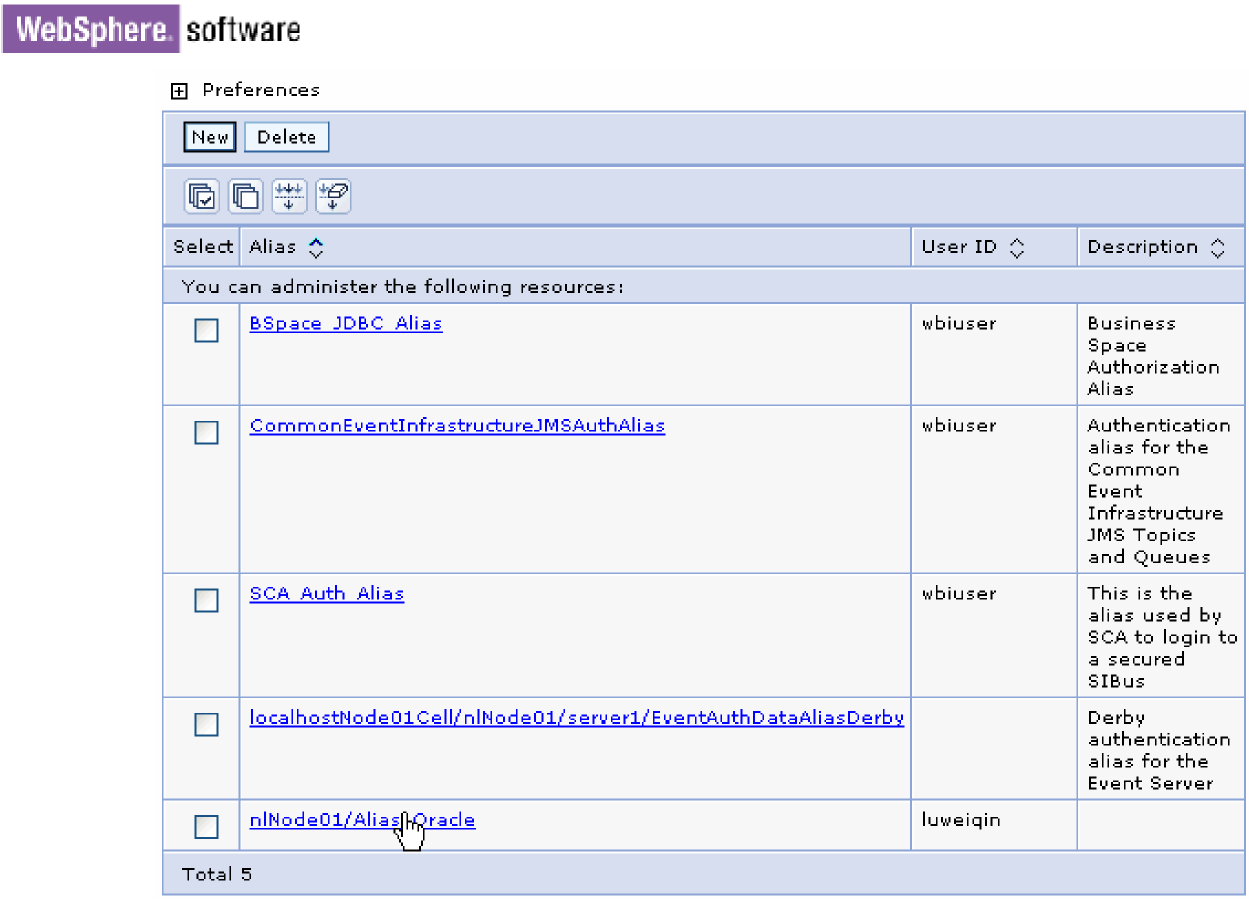
⚠ The server may need to be restarted for these changes to take effect.

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)

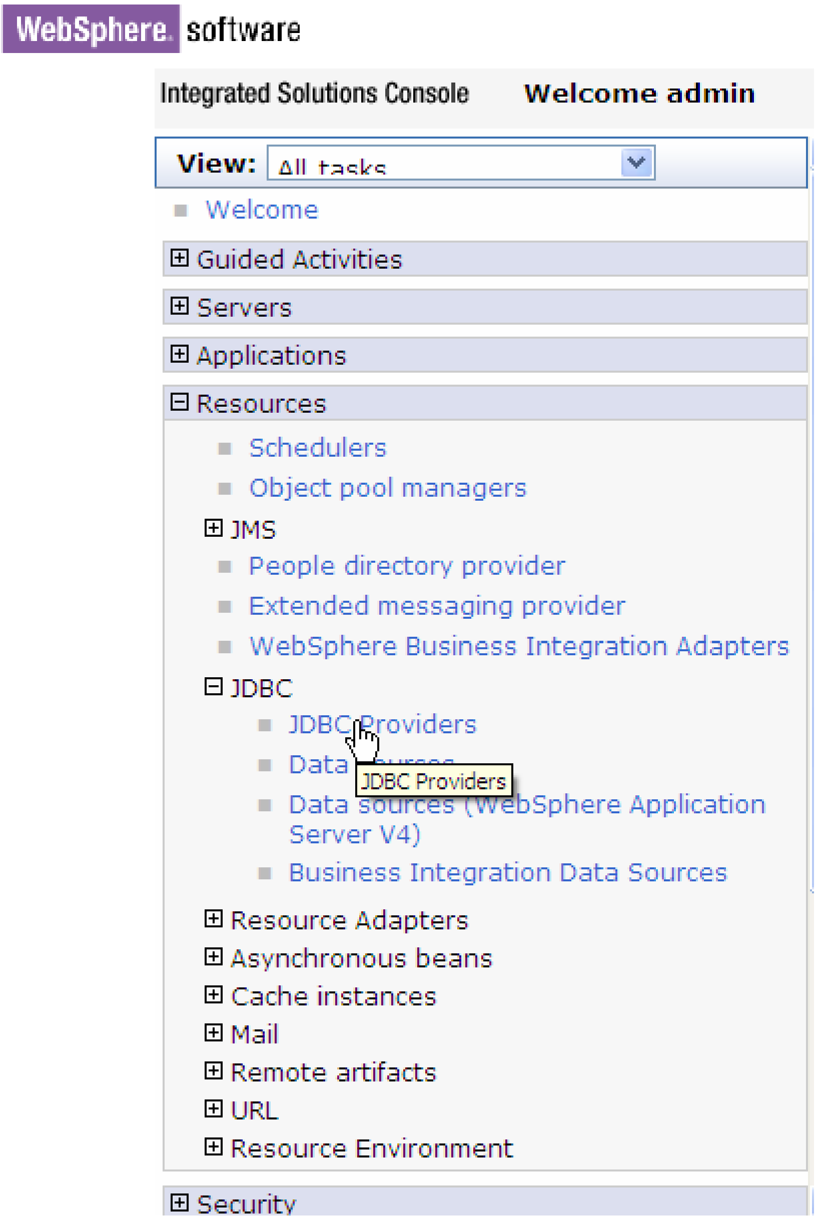
We have created an authentication alias that will be used to configure the data source.



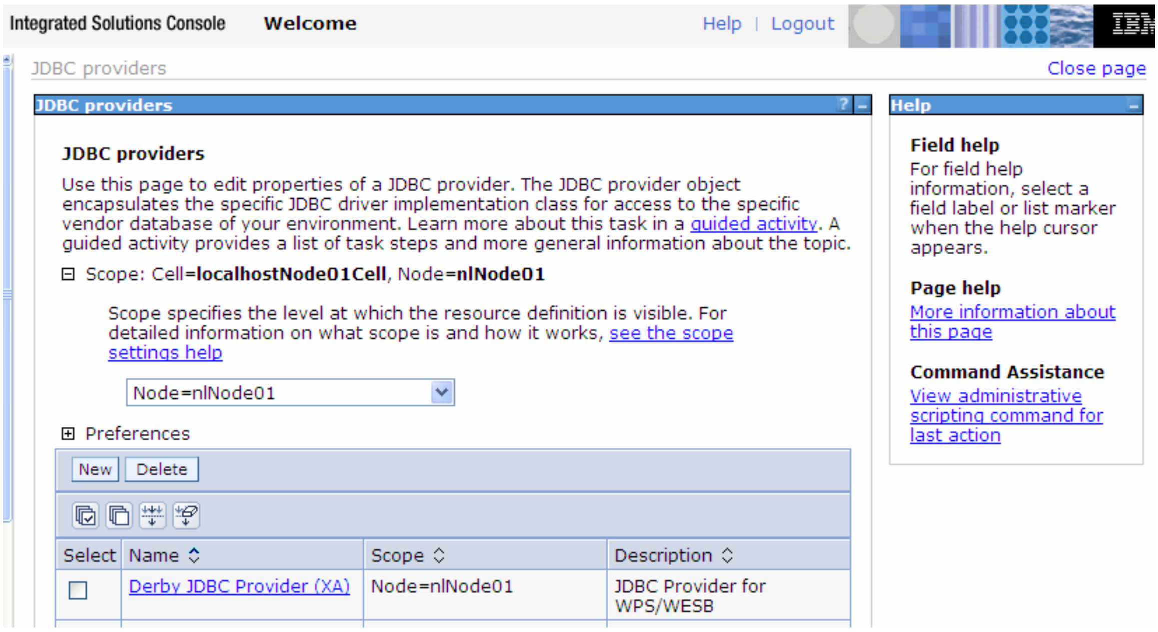
Select	Alias	User ID	Description
You can administer the following resources:			
□	BSpace_JDBC_Alias	wbiuser	Business Space Authorization Alias
□	CommonEventInfrastructureJMSAuthAlias	wbiuser	Authentication alias for the Common Event Infrastructure JMS Topics and Queues
□	SCA_Auth_Alias	wbiuser	This is the alias used by SCA to login to a secured SIBus
□	localhostNode01Cell/nlNode01/server1/EventAuthDataAliasDerby		Derby authentication alias for the Event Server
□	nlNode01/Alias_Oracle	luweiqin	
Total 5			

Create a data source

1. Click **Resources > JDBC > JDBC Providers.**



2. On the right, select **Node=nlNode01** from the drop-down list.



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=**nlNode01**

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=nlNode01

Preferences

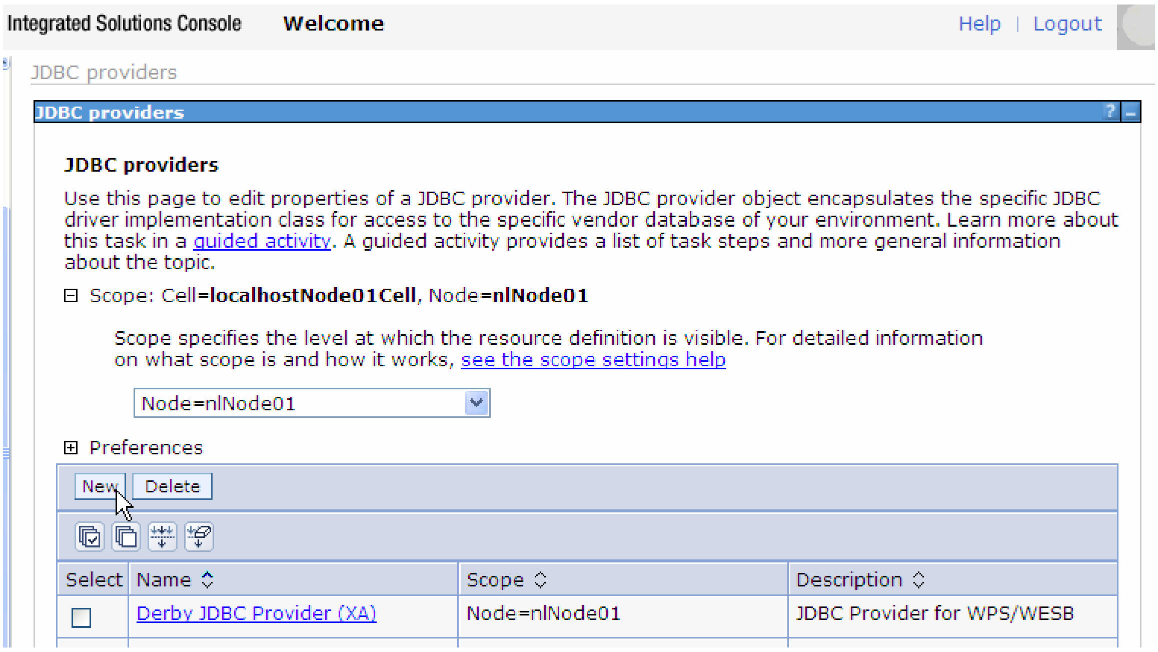
Select	Name	Scope	Description
□	Derby JDBC Provider (XA)	Node=nlNode01	JDBC Provider for WPS/WESB

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

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



Integrated Solutions Console **Welcome** Help | Logout

JDBC providers

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=**nlNode01**

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

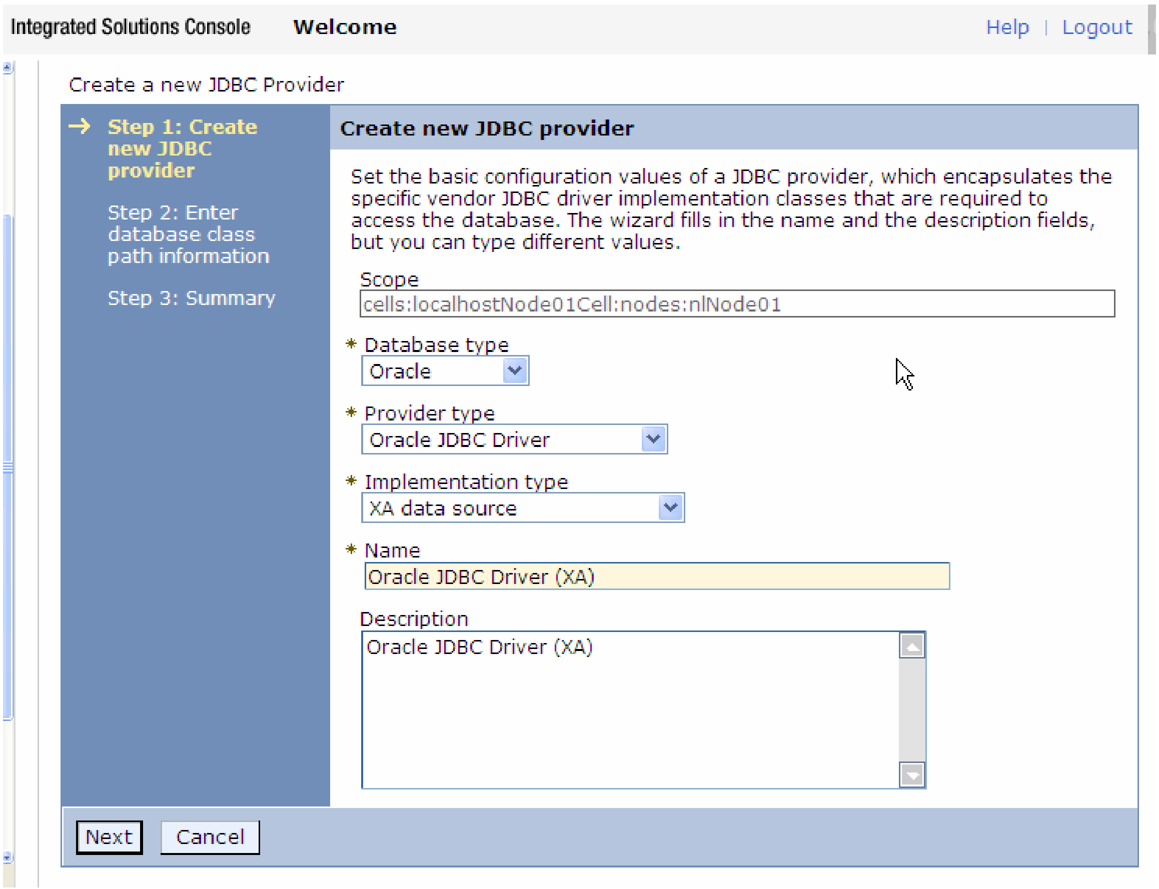
Node=nlNode01

⊞ Preferences

New Delete

Select	Name	Scope	Description
□	Derby JDBC Provider (XA)	Node=nlNode01	JDBC Provider for WPS/WESB

4. Specify the for the **Database type**, **Provider type**, and **Implementation type** fields as shown in the figure below. Click **Next**.



Integrated Solutions Console **Welcome** Help | Logout

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:nlNode01

* Database type: Oracle

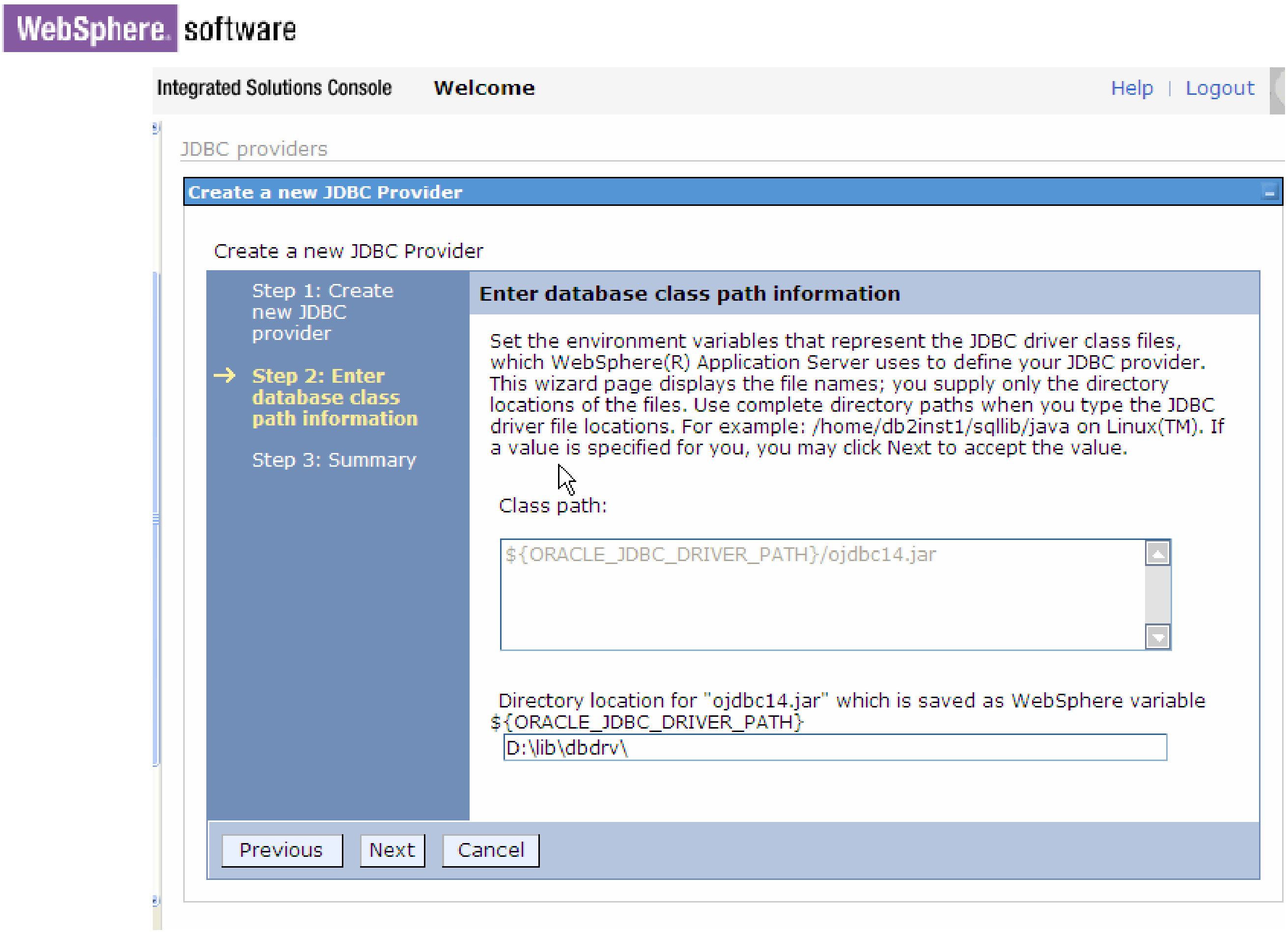
* Provider type: Oracle JDBC Driver

* Implementation type: XA data source

* Name: Oracle JDBC Driver (XA)

Description: Oracle JDBC Driver (XA)

5. Enter the absolute path of the JDBC driver (ojdbc6.jar) directory. Click **Next**.


WebSphere. software
Integrated Solutions Console
Welcome
Help | Logout
JDBC providers
Create a new JDBC Provider
Step 1: Create new JDBC provider
Step 2: Enter database class path information
Step 3: Summary
Previous
Next
Cancel

Create a new JDBC Provider

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.

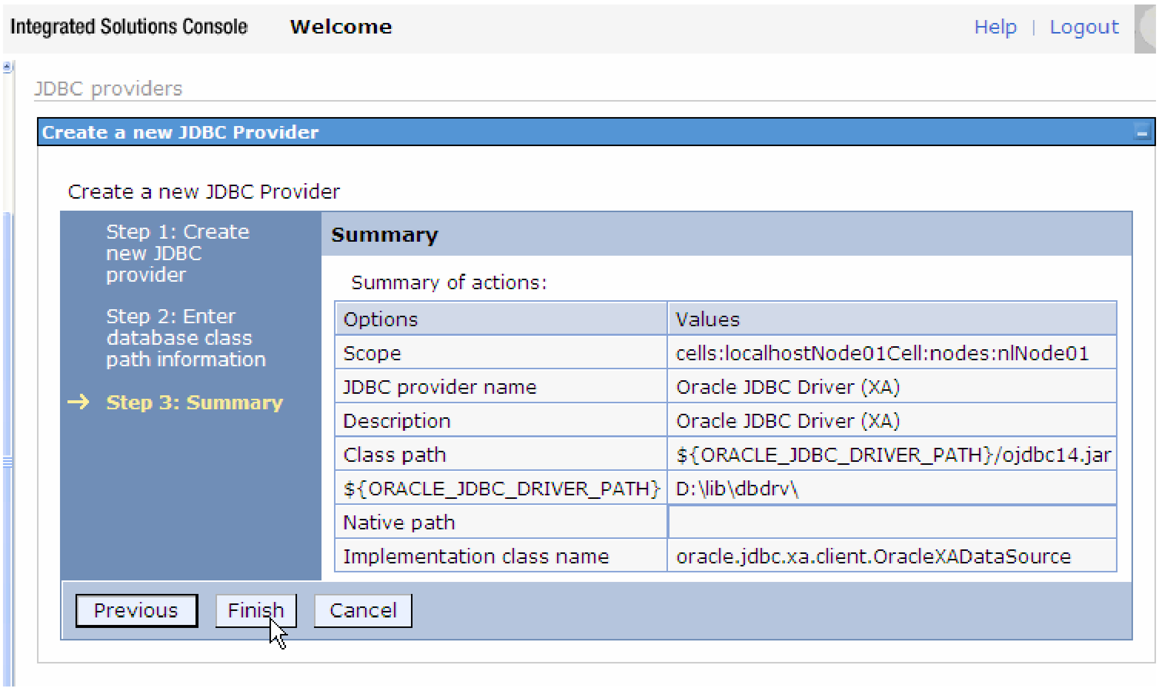
Class path:

${ORACLE_JDBC_DRIVER_PATH}/ojdbc14.jar

Directory location for "ojdbc14.jar" which is saved as WebSphere variable ${ORACLE_JDBC_DRIVER_PATH}

D:\lib\dbdrv\

6. Click **Finish**.


Integrated Solutions Console
Welcome
Help | Logout
JDBC providers
Create a new JDBC Provider
Step 1: Create new JDBC provider
Step 2: Enter database class path information
Step 3: Summary
Previous
Finish
Cancel

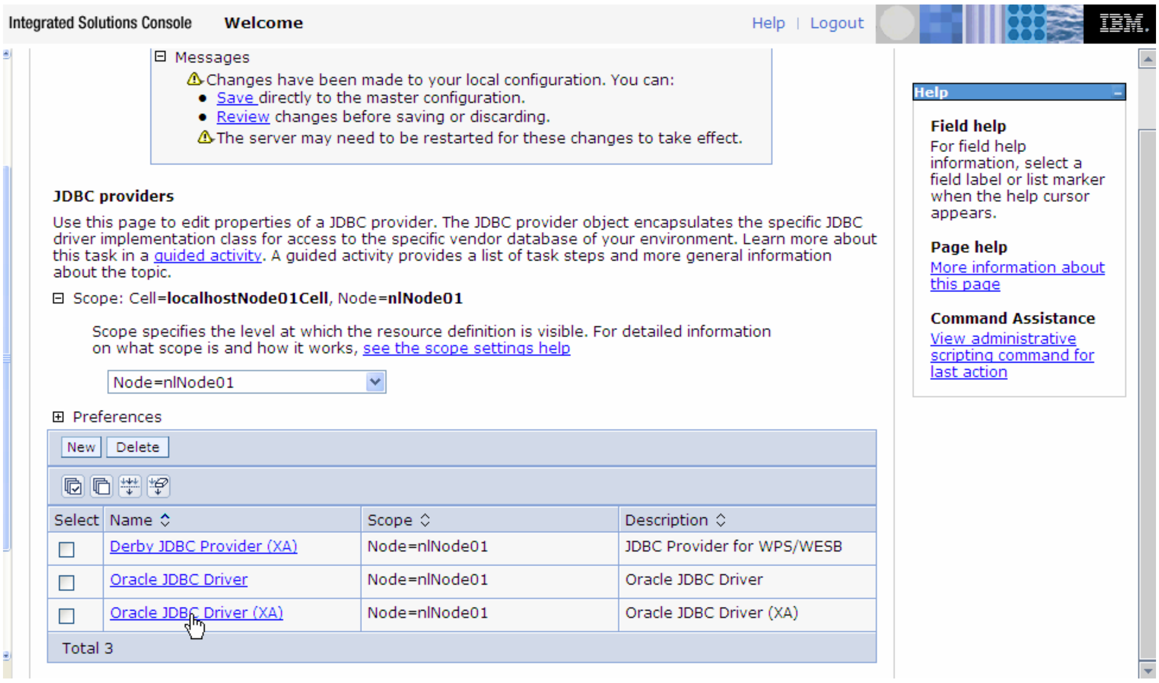
Create a new JDBC Provider

Summary

Summary of actions:

Options	Values
Scope	cells:localhostNode01Cell:nodes:nlNode01
JDBC provider name	Oracle JDBC Driver (XA)
Description	Oracle JDBC Driver (XA)
Class path	${ORACLE_JDBC_DRIVER_PATH}/ojdbc14.jar
${ORACLE_JDBC_DRIVER_PATH}	D:\lib\dbdrv\
Native path	
Implementation class name	oracle.jdbc.xa.client.OracleXADataSource

7. Click the JDBC Provider that you just created.



⊟ 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=**nlNode01**

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=nlNode01

⊞ Preferences

New | Delete

Select	Name	Scope	Description
□	Derby JDBC Provider (XA)	Node=nlNode01	JDBC Provider for WPS/WESB
□	Oracle JDBC Driver	Node=nlNode01	Oracle JDBC Driver
□	Oracle JDBC Driver (XA)	Node=nlNode01	Oracle JDBC Driver (XA)

Total 3

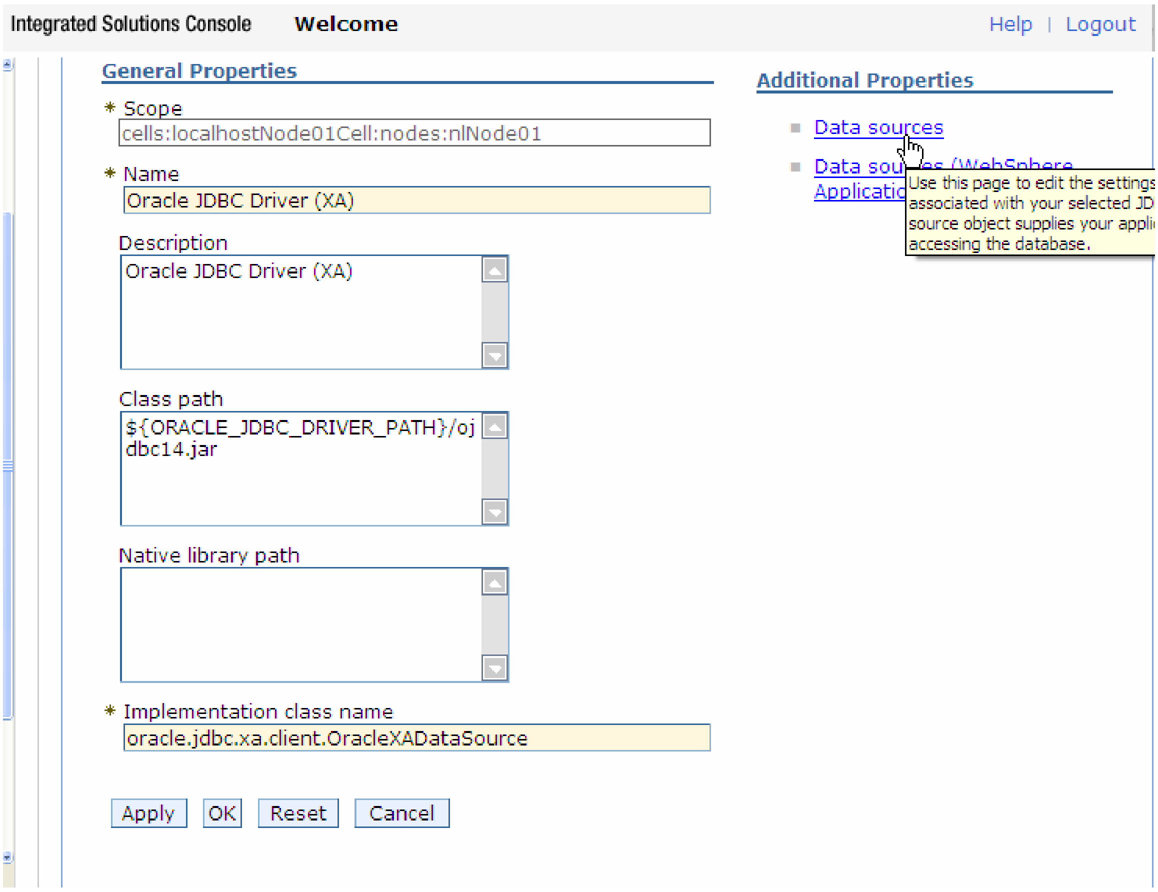
Help

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

8. Click **Data sources**, under **Additional Properties**.



General Properties

* Scope
cells:localhostNode01Cell:nodes:nlNode01

* Name
Oracle JDBC Driver (XA)

Description
Oracle JDBC Driver (XA)

Class path
${ORACLE_JDBC_DRIVER_PATH}/ojdbc14.jar

Native library path

* Implementation class name
oracle.jdbc.xa.client.OracleXADataSource

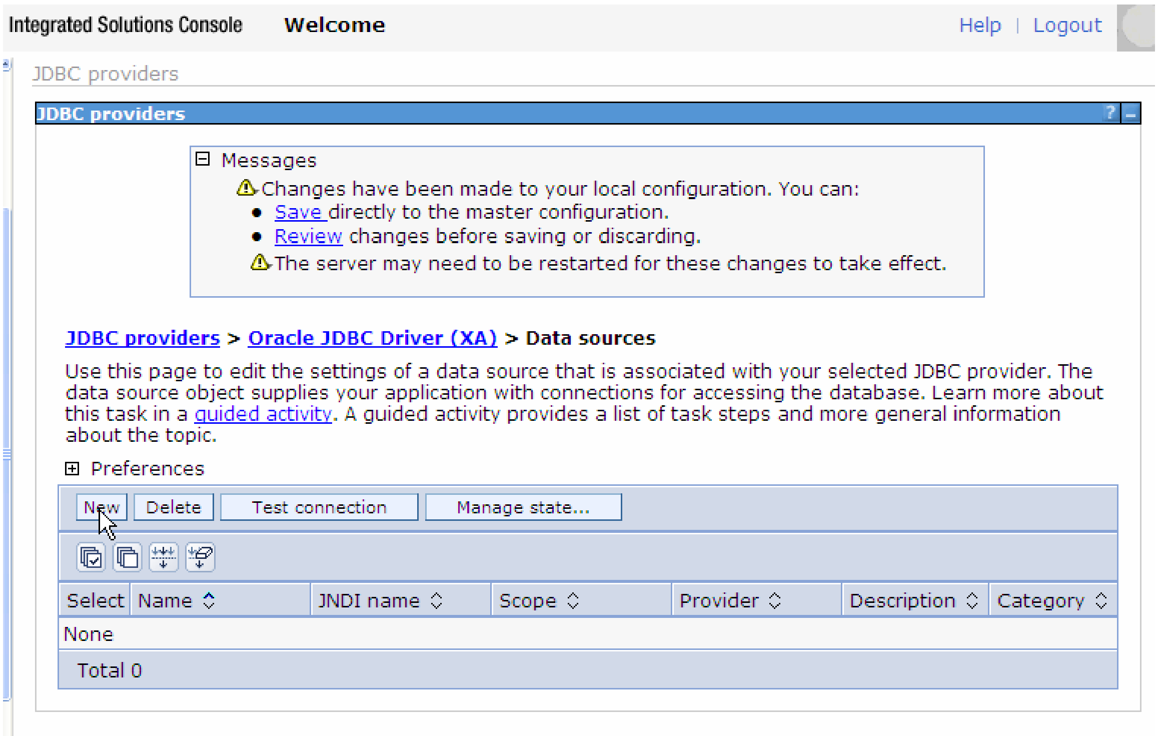
Apply | OK | Reset | Cancel

Additional Properties

- Data sources
- Data sources (WebSphere Applicatio

Use this page to edit the settings associated with your selected JD source object supplies your appli accessing the database.

9. 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 > 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.

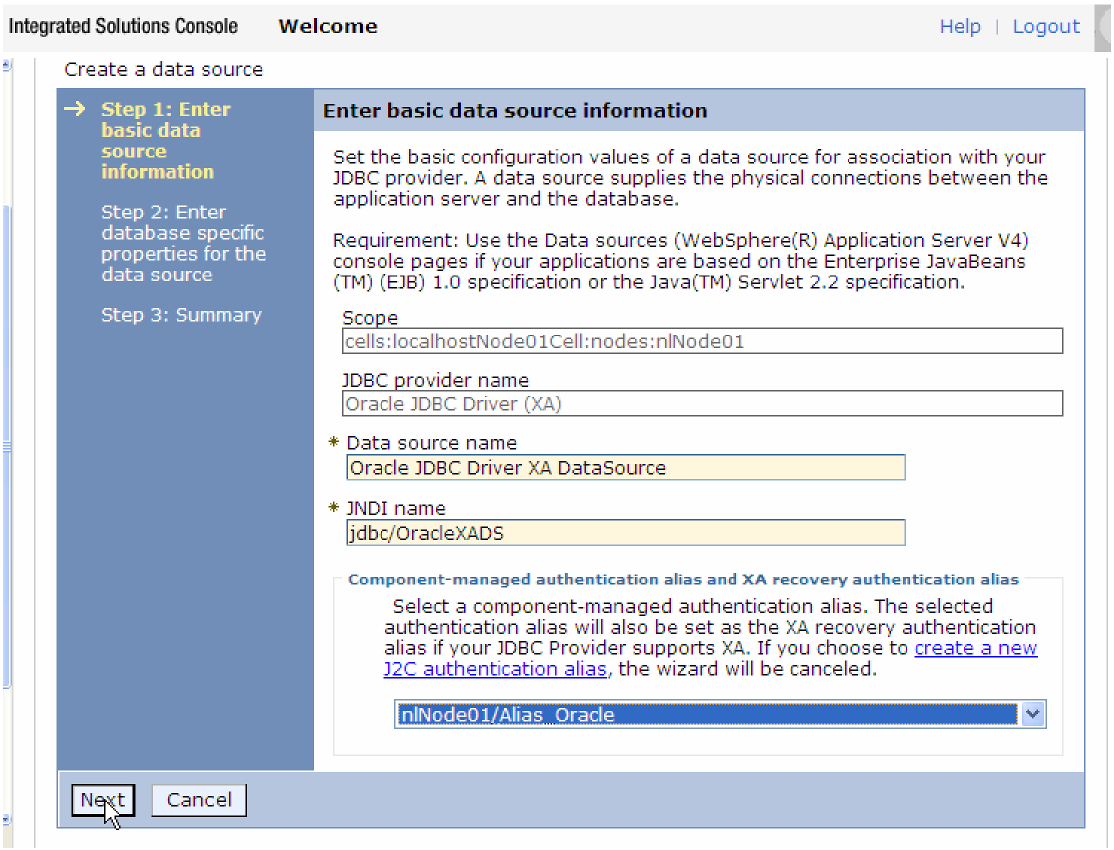
⊞ Preferences

Select	Name ◇	JNDI name ◇	Scope ◇	Provider ◇	Description ◇	Category ◇
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**.



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

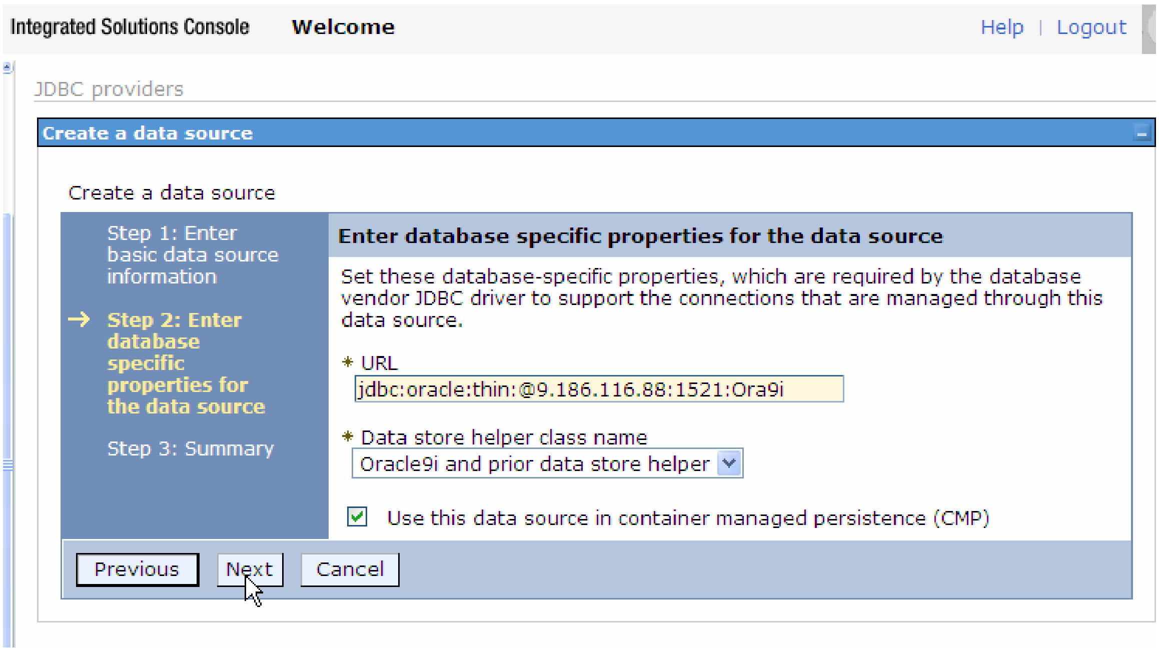
* JNDI name
jdbc/OracleXADS

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

12. Enter the URL to connect to the database in the **URL** field. Click **Next**.



Integrated Solutions Console
Welcome
Help | Logout
JDBC providers
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: Summary

Enter database specific properties for the data source

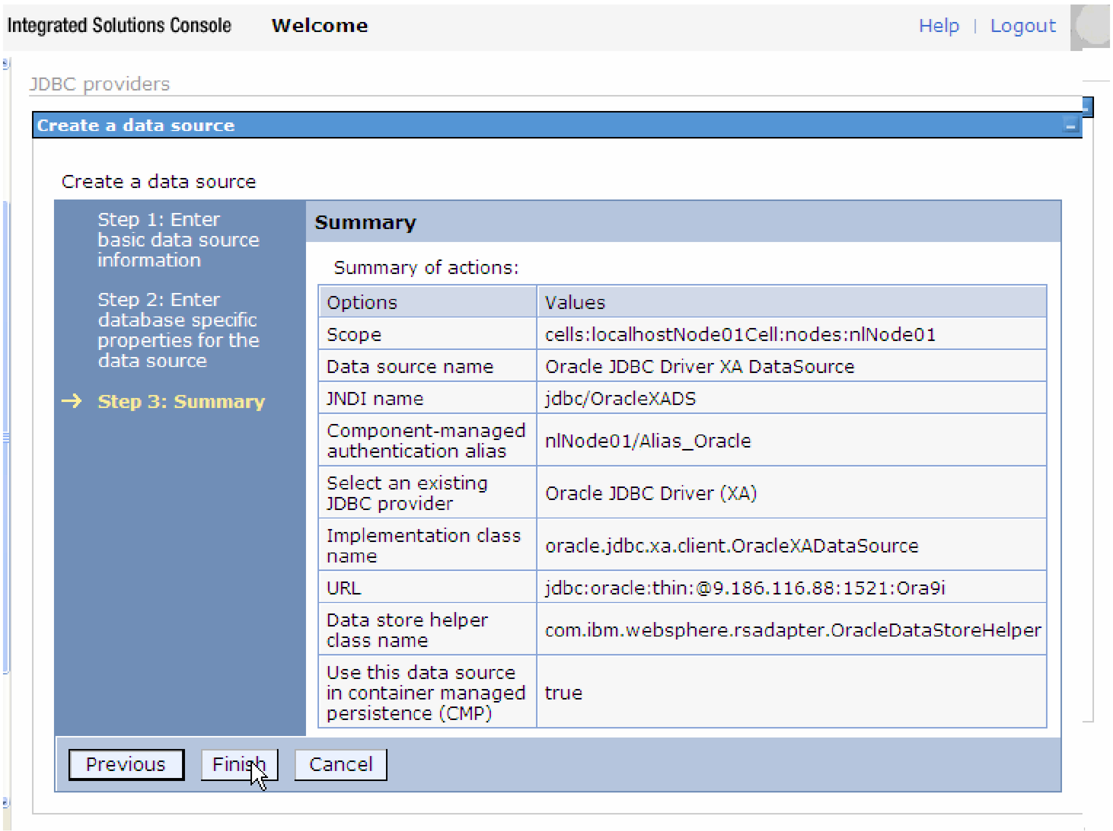
Set these database-specific properties, which are required by the database vendor JDBC driver to support the connections that are managed through this data source.

* 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)

13. Click **Finish**.



Integrated Solutions Console
Welcome
Help | Logout
JDBC providers
Create a data source

Create a data source

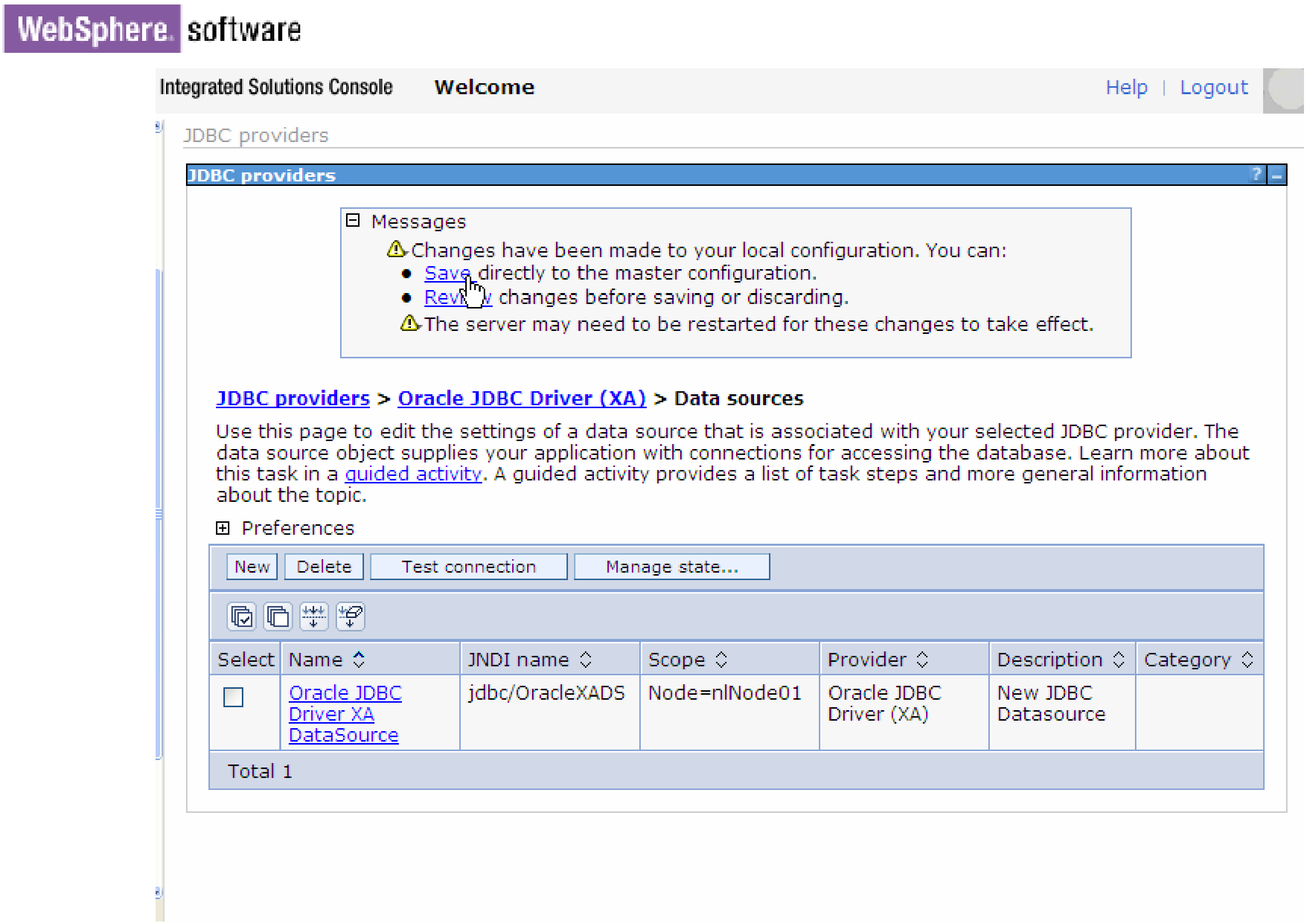
Step 1: Enter basic data source information
Step 2: Enter database specific properties for the data source
→ Step 3: Summary

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.rsadapter.OracleDataStoreHelper
Use this data source in container managed persistence (CMP)	true

14. In the **Messages** area, click on **Save** link. This will save changes made to the local configuration onto the master configuration.



WebSphere software

Integrated Solutions Console Welcome Help | Logout

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.

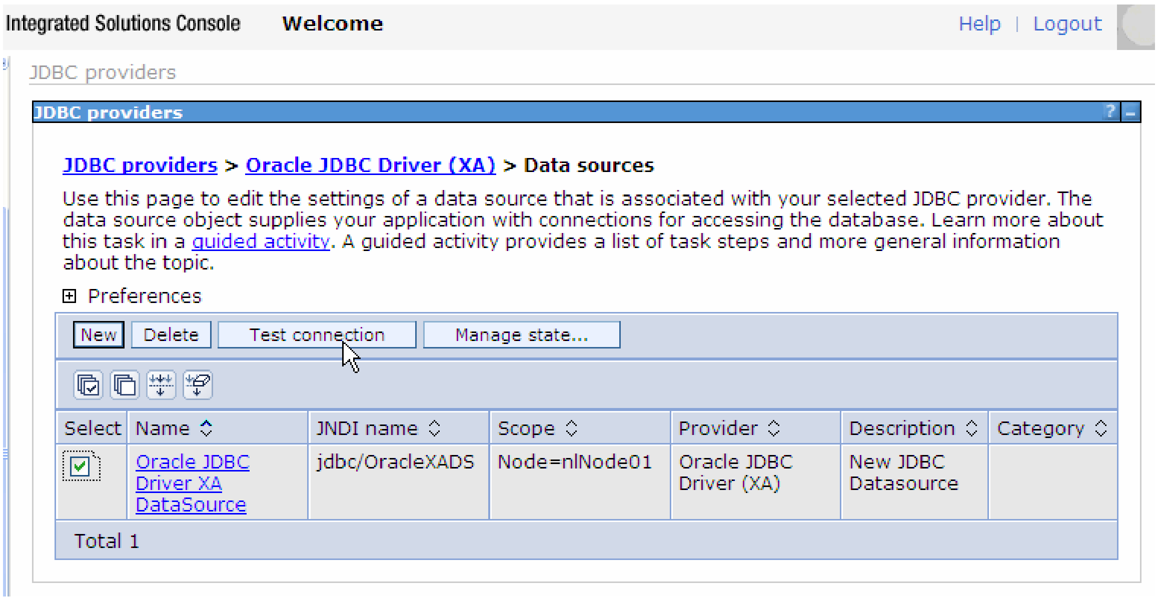
JDBC providers > Oracle JDBC Driver (XA) > **Data sources**

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

Select	Name	JNDI name	Scope	Provider	Description	Category
□	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**.



Integrated Solutions Console Welcome Help | Logout

JDBC providers

JDBC providers

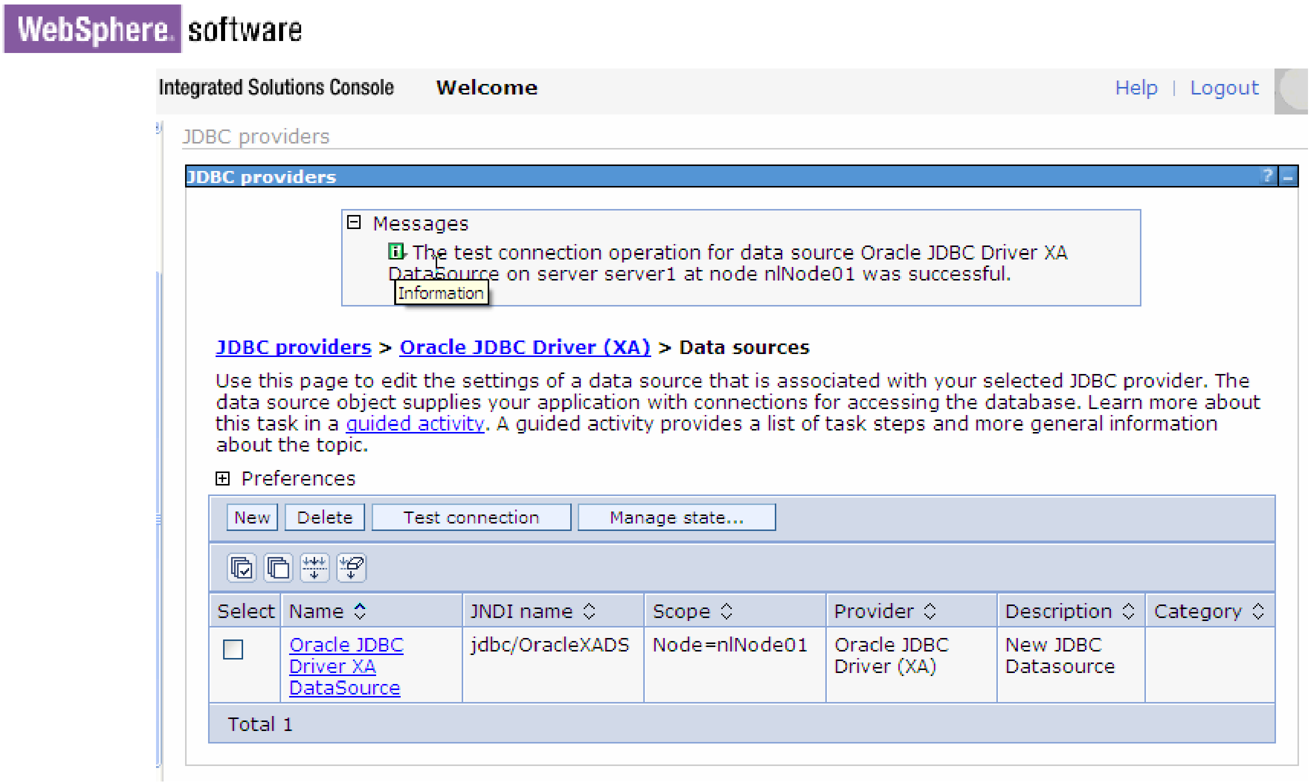
JDBC providers > Oracle JDBC Driver (XA) > **Data sources**

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

Select	Name	JNDI name	Scope	Provider	Description	Category
☑	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.



WebSphere. software

Integrated Solutions Console Welcome Help | Logout

JDBC providers

JDBC providers

⊟ Messages

The test connection operation for data source Oracle JDBC Driver XA DataSource on server server1 at node nlNode01 was successful.

Information

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 ◇
□	Oracle JDBC Driver XA DataSource	jdbc/OracleXADS	Node=nlNode01	Oracle JDBC Driver (XA)	New JDBC Datasource	

Total 1

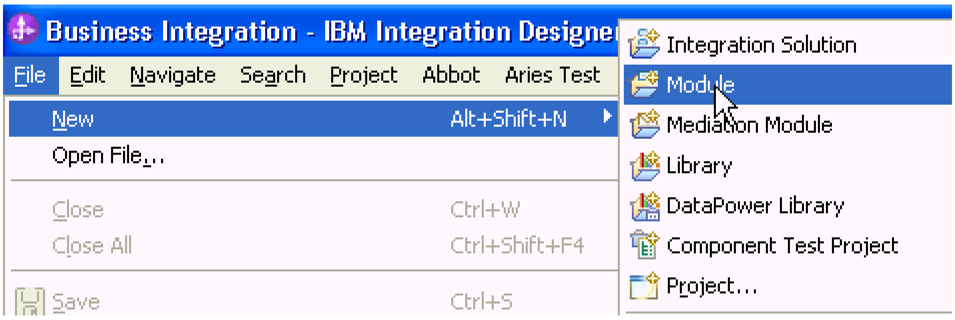
17. Close the **Administration Console** tab.

Configure the adapter for outbound processing

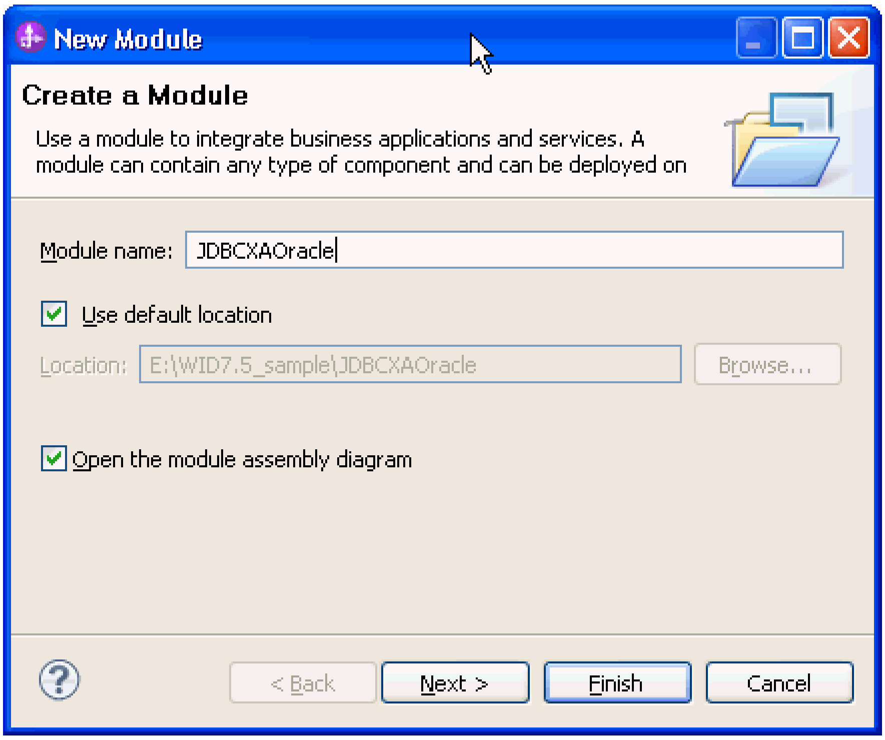
Run the external service wizard to specify business objects, services, and configuration details.

Set connection properties for the external service wizard

1. Switch to the Business Integration perspective in IBM Integration Designer.
2. Select **File->New->Module** to create a Module project.



3. Specify the module name as **JDBCXAOracle**, click **Finish**.



New Module

Create a Module

Use a module to integrate business applications and services. A module can contain any type of component and can be deployed on

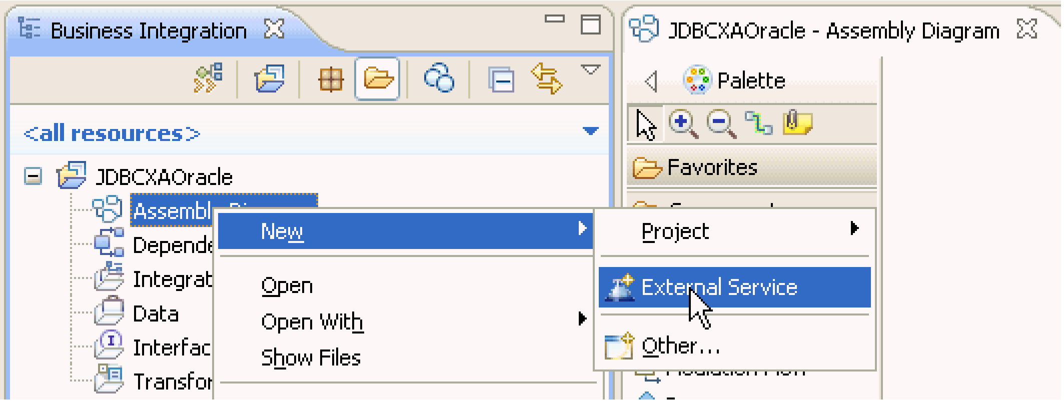
Module name: JDBCXAOracle

☑ Use default location

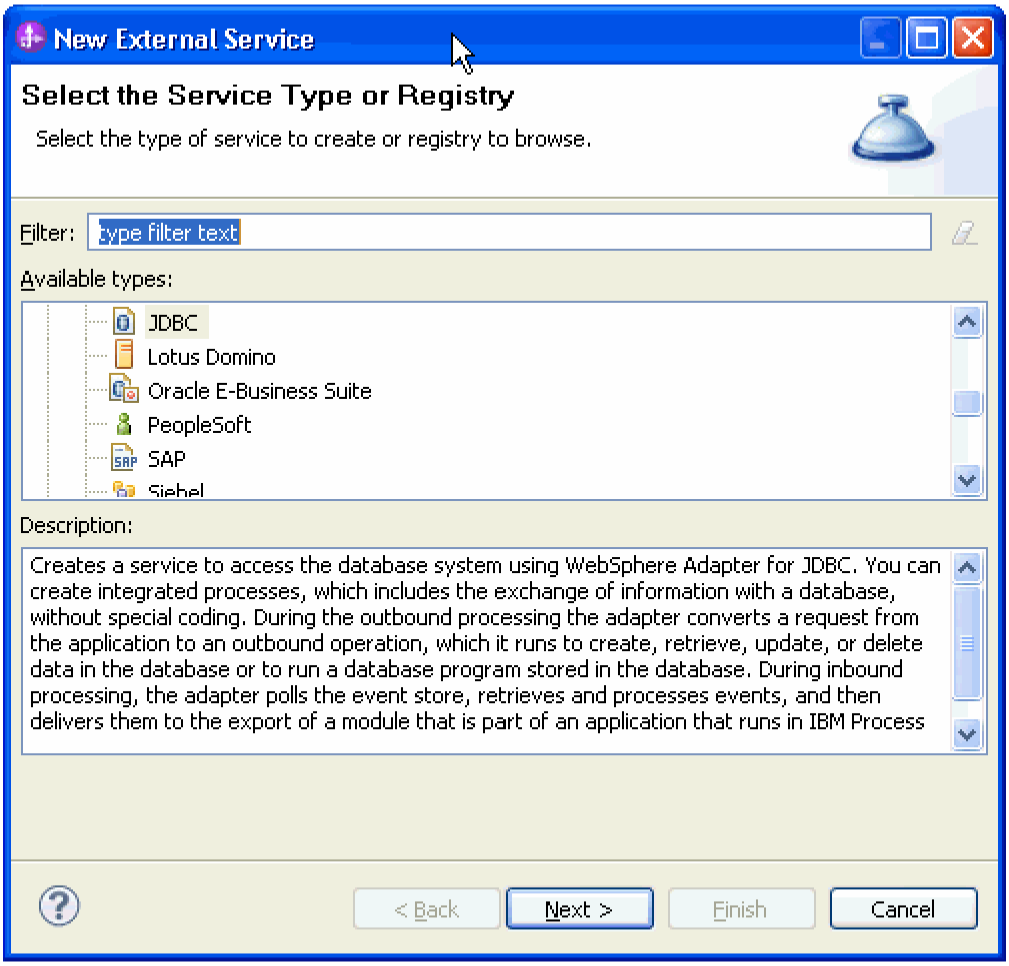
Location: E:\WID7.5_sample\JDBCXAOracle

☑ Open the module assembly diagram

4. Right-click **JDBCXAOracle->Assembly Diagram**, select **New->External Service.**



5. Select JDBC, and click **Next**.



New External Service

Select the Service Type or Registry

Select the type of service to create or registry to browse.

Filter: type filter text

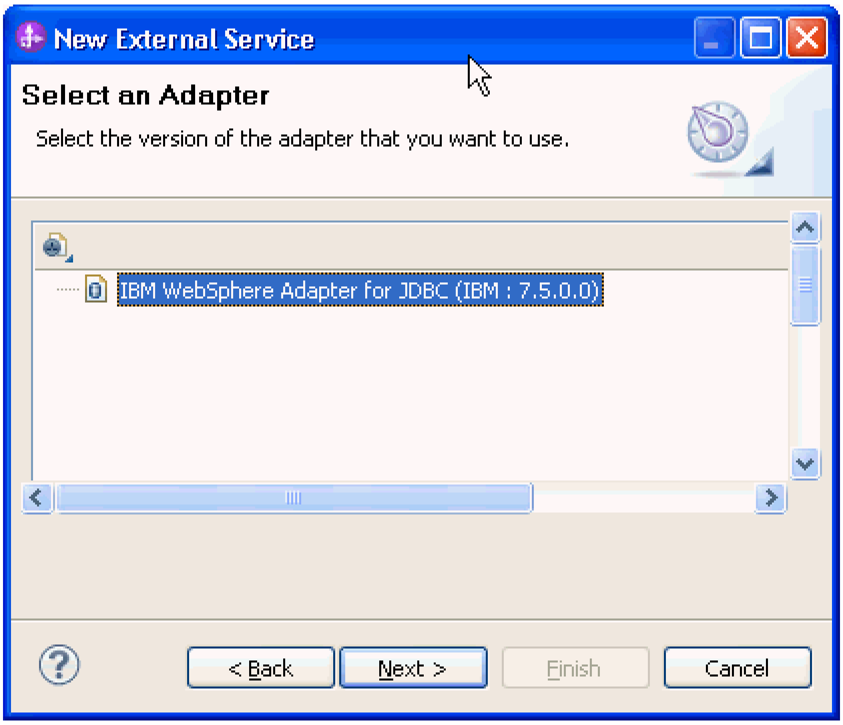
Available types:

- JDBC
- Lotus Domino
- Oracle E-Business Suite
- PeopleSoft
- SAP
- Siebel

Description:

Creates a service to access the database system using WebSphere Adapter for JDBC. You can create integrated processes, which includes the exchange of information with a database, without special coding. During the outbound processing the adapter converts a request from the application to an outbound operation, which it runs to create, retrieve, update, or delete data in the database or to run a database program stored in the database. During inbound processing, the adapter polls the event store, retrieves and processes events, and then delivers them to the export of a module that is part of an application that runs in IBM Process

6. Select **IBM WebSphere Adapter for JDBC (IBM: 7.5.0.0).** Click **Next**.



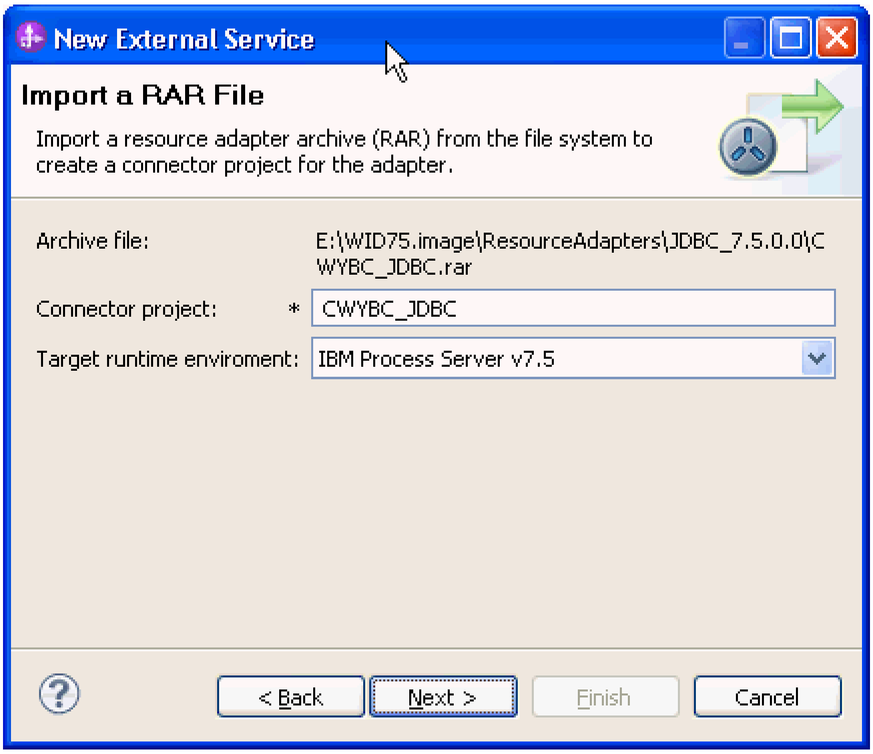
New External Service

Select an Adapter

Select the version of the adapter that you want to use.

- IBM WebSphere Adapter for JDBC (IBM : 7.5.0.0)

7. In the **Target Runtime environment** field, select the appropriate runtime and click **Next.**



Import a RAR File

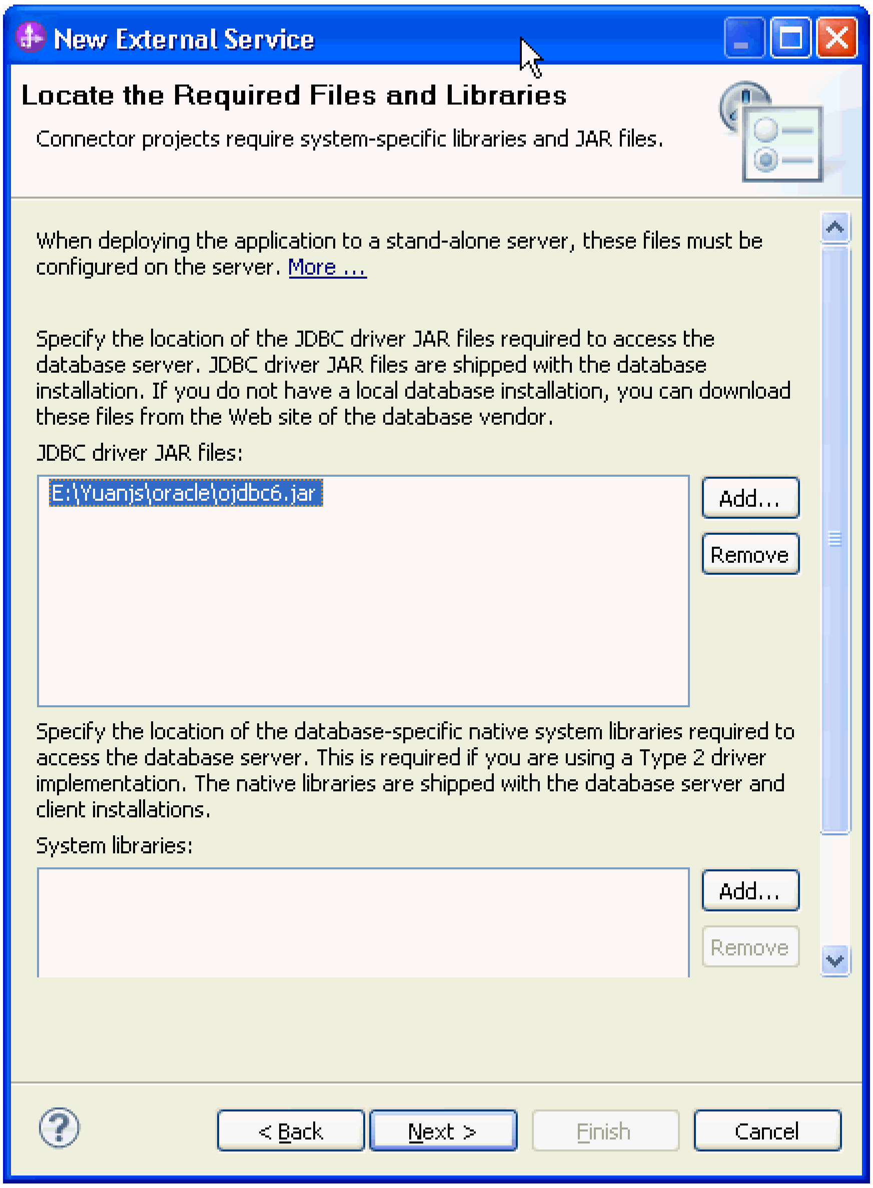
Import a resource adapter archive (RAR) from the file system to create a connector project for the adapter.

Archive file: E:\WID75.image\ResourceAdapters\JDBC_7.5.0.0\CWYBC_JDBC.rar

Connector project: * CWYBC_JDBC

Target runtime enviroment: IBM Process Server v7.5

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**.



Locate the Required Files and Libraries

Connector projects require system-specific libraries and JAR files.

When deploying the application to a stand-alone server, these files must be configured on the server. More ...

Specify the location of the JDBC driver JAR files required to access the database server. JDBC driver JAR files are shipped with the database installation. If you do not have a local database installation, you can download these files from the Web site of the database vendor.

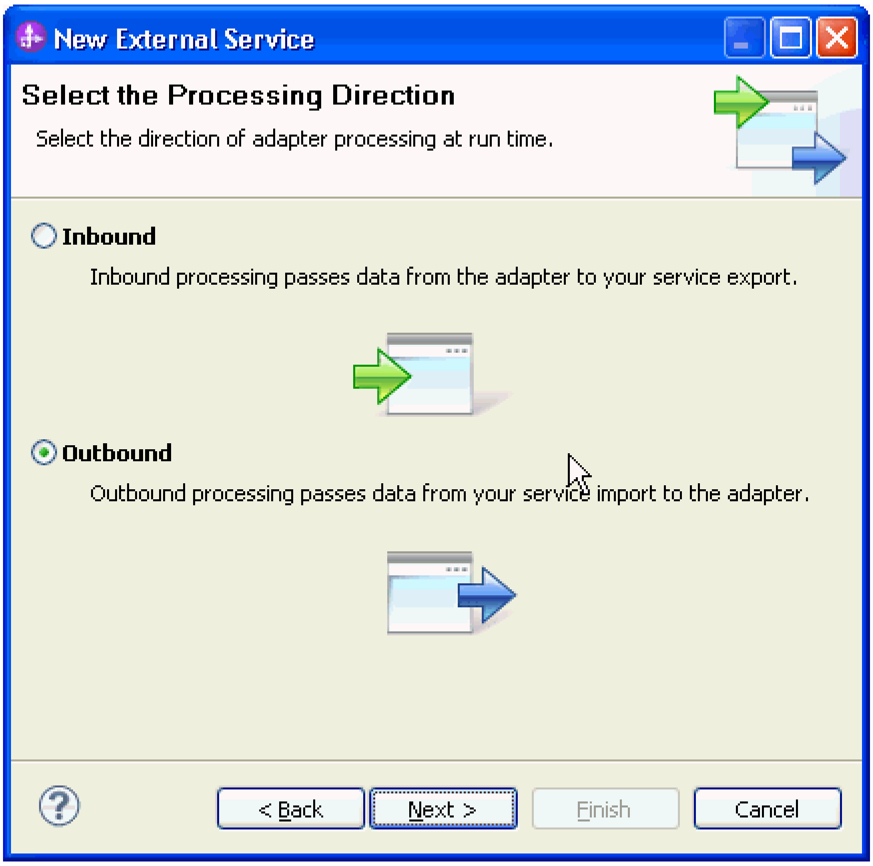
JDBC driver JAR files:

E:\Yuanjs\oracle\ojdbc6.jar

Specify the location of the database-specific native system libraries required to access the database server. This is required if you are using a Type 2 driver implementation. The native libraries are shipped with the database server and client installations.

System libraries:

9. Click the **Outbound** radio button. Click **Next**.



New External Service

Select the Processing Direction

Select the direction of adapter processing at run time.

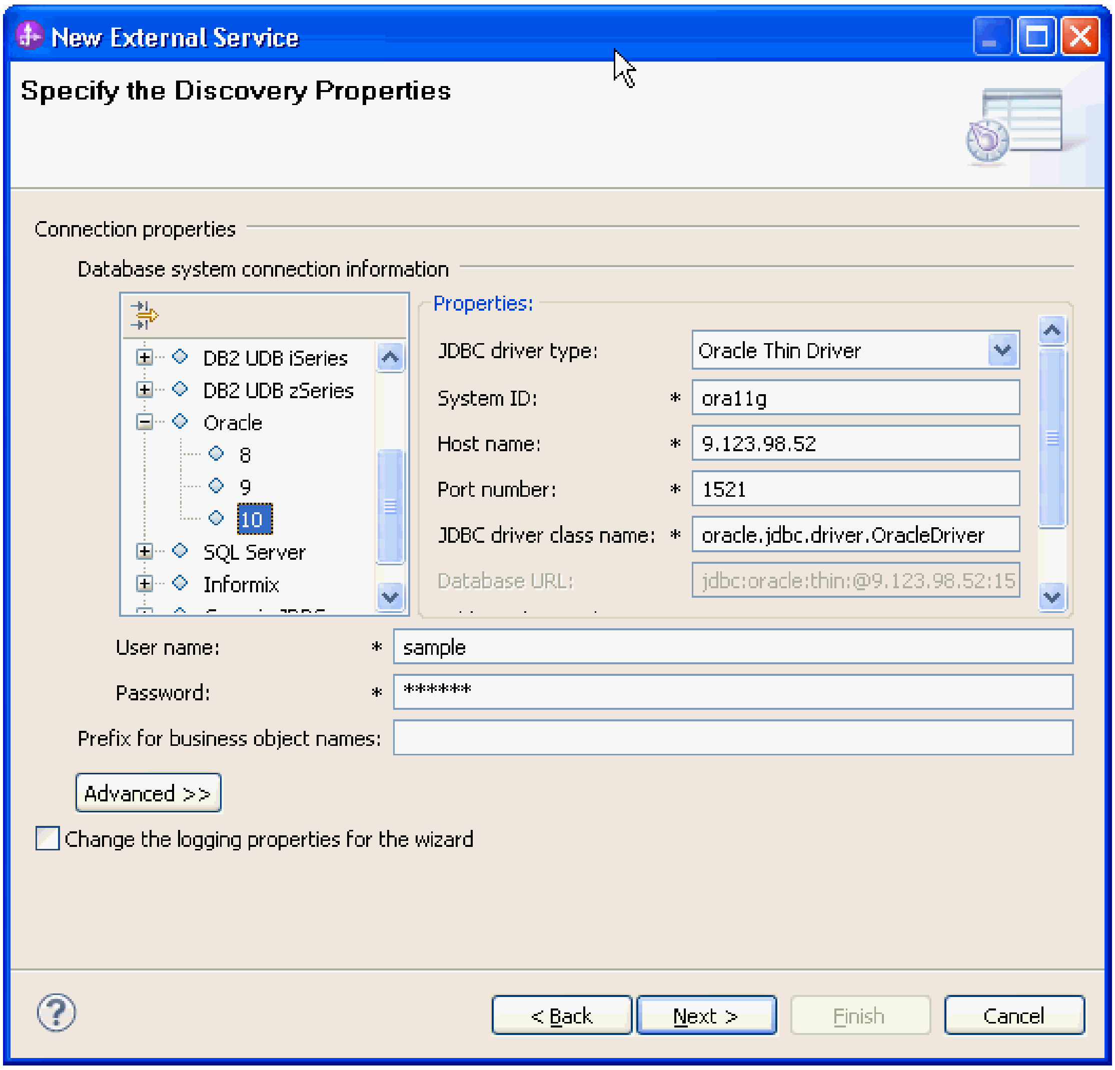
○ **Inbound**

Inbound processing passes data from the adapter to your service export.

● **Outbound**

Outbound processing passes data from your service import to the adapter.

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**.



New External Service

Specify the Discovery Properties

Connection properties

Database system connection information

Properties:

JDBC driver type: Oracle Thin Driver

System ID: * ora11g

Host name: * 9.123.98.52

Port number: * 1521

JDBC driver class name: * oracle.jdbc.driver.OracleDriver

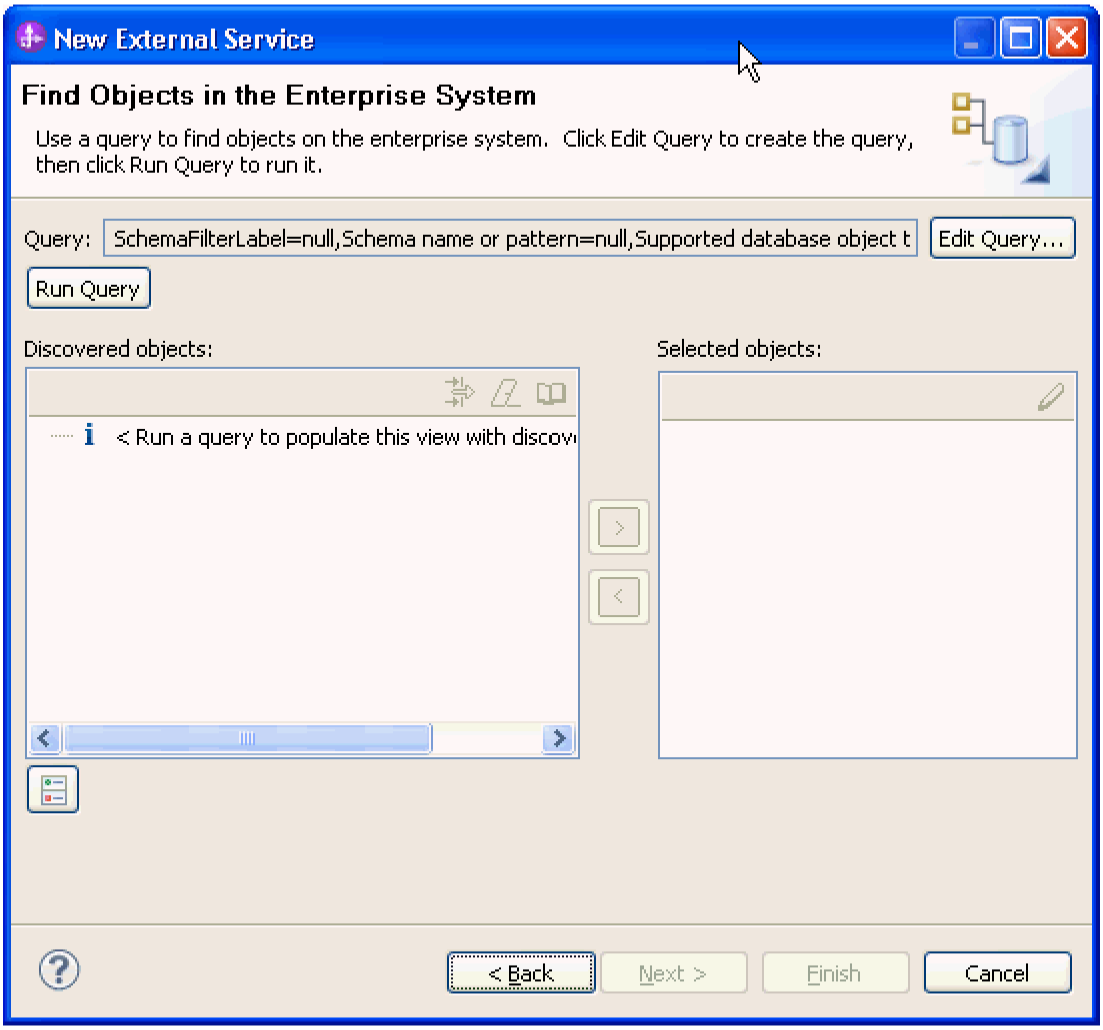
Database URL: jdbc:oracle:thin:@9.123.98.52:15

User name: * sample

Password: * *******

Prefix for business object names:

12. Click **Run Query** to list the tables, stored procedures, views, and synonyms for each schema in the database.



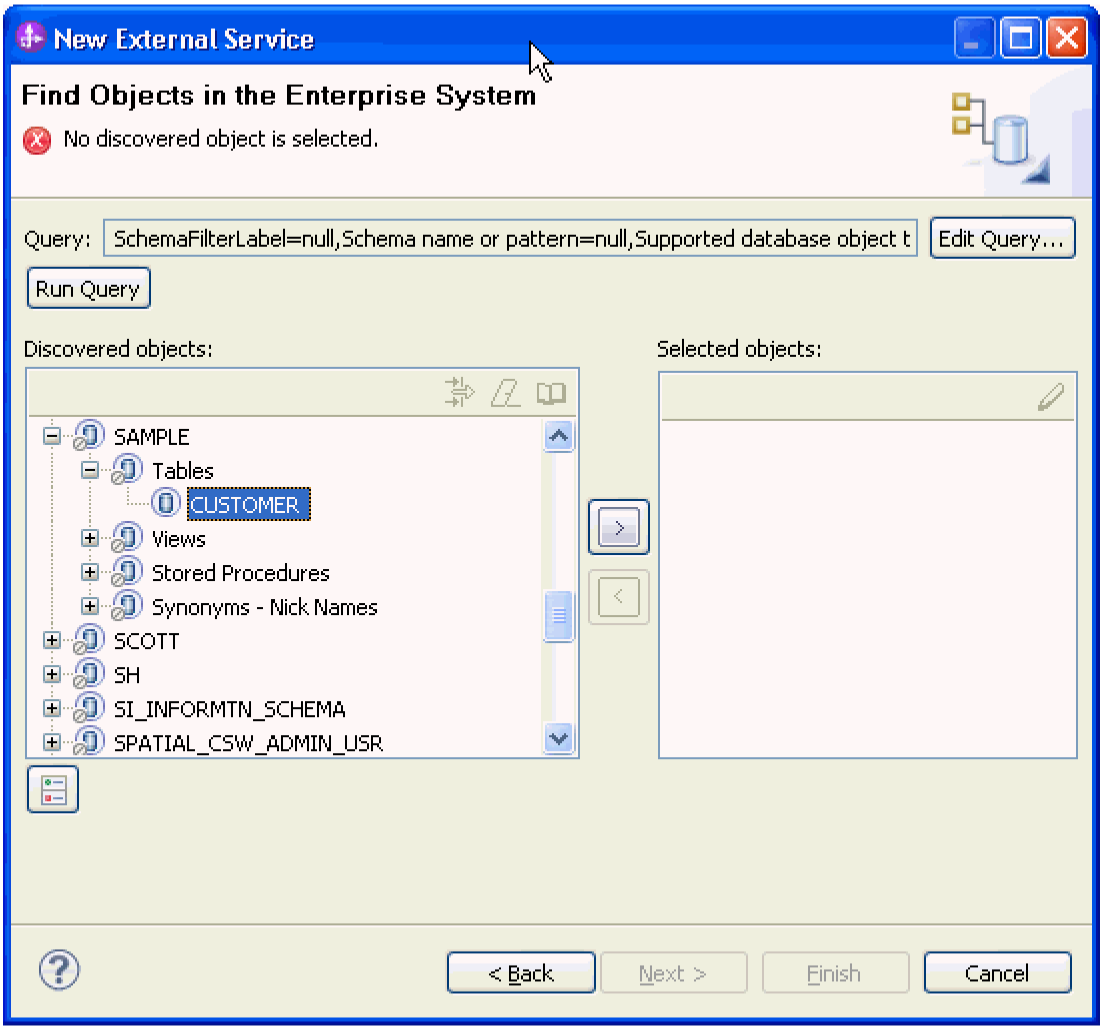
New External Service

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 t

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.



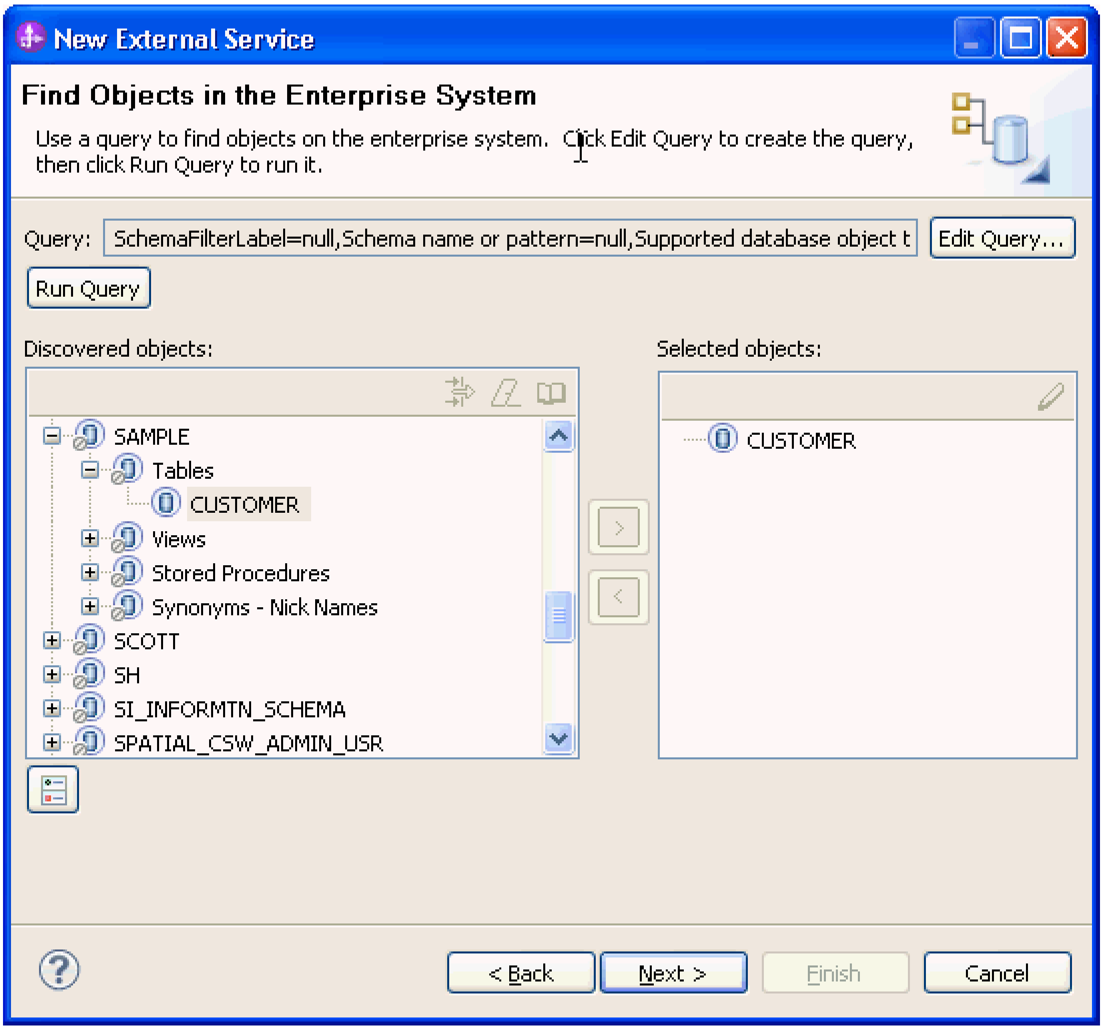
New External Service

Find Objects in the Enterprise System

No discovered object is selected.

Query: SchemaFilterLabel=null,Schema name or pattern=null,Supported database object t

14. Click **Next**.



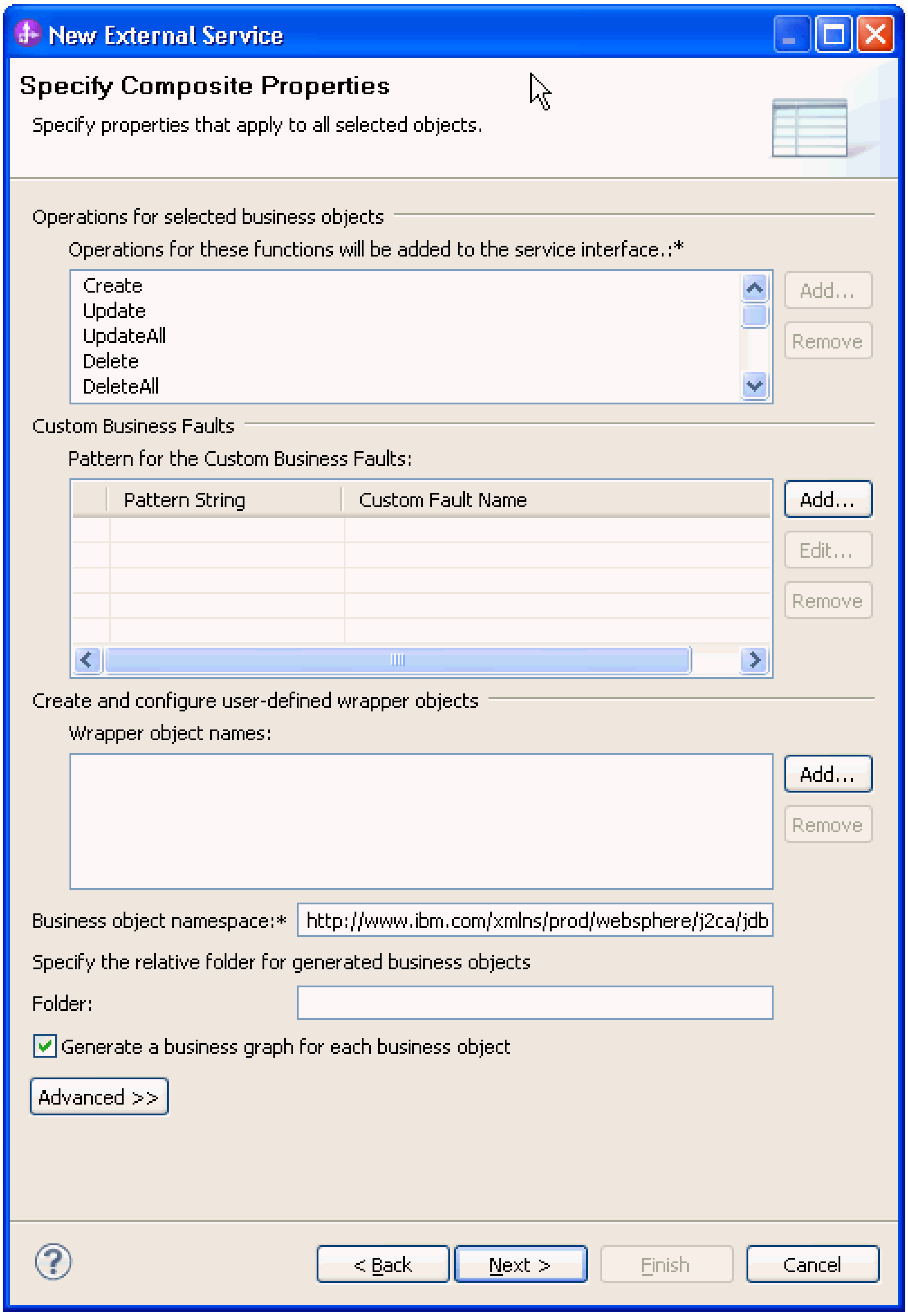
New External Service

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 t

15. In the Specify Composite Properties window, click **Next**.



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
- UpdateAll
- Delete
- DeleteAll

Custom Business Faults

Pattern for the Custom Business Faults:

	Pattern String	Custom Fault Name

Create and configure user-defined wrapper objects

Wrapper object names:

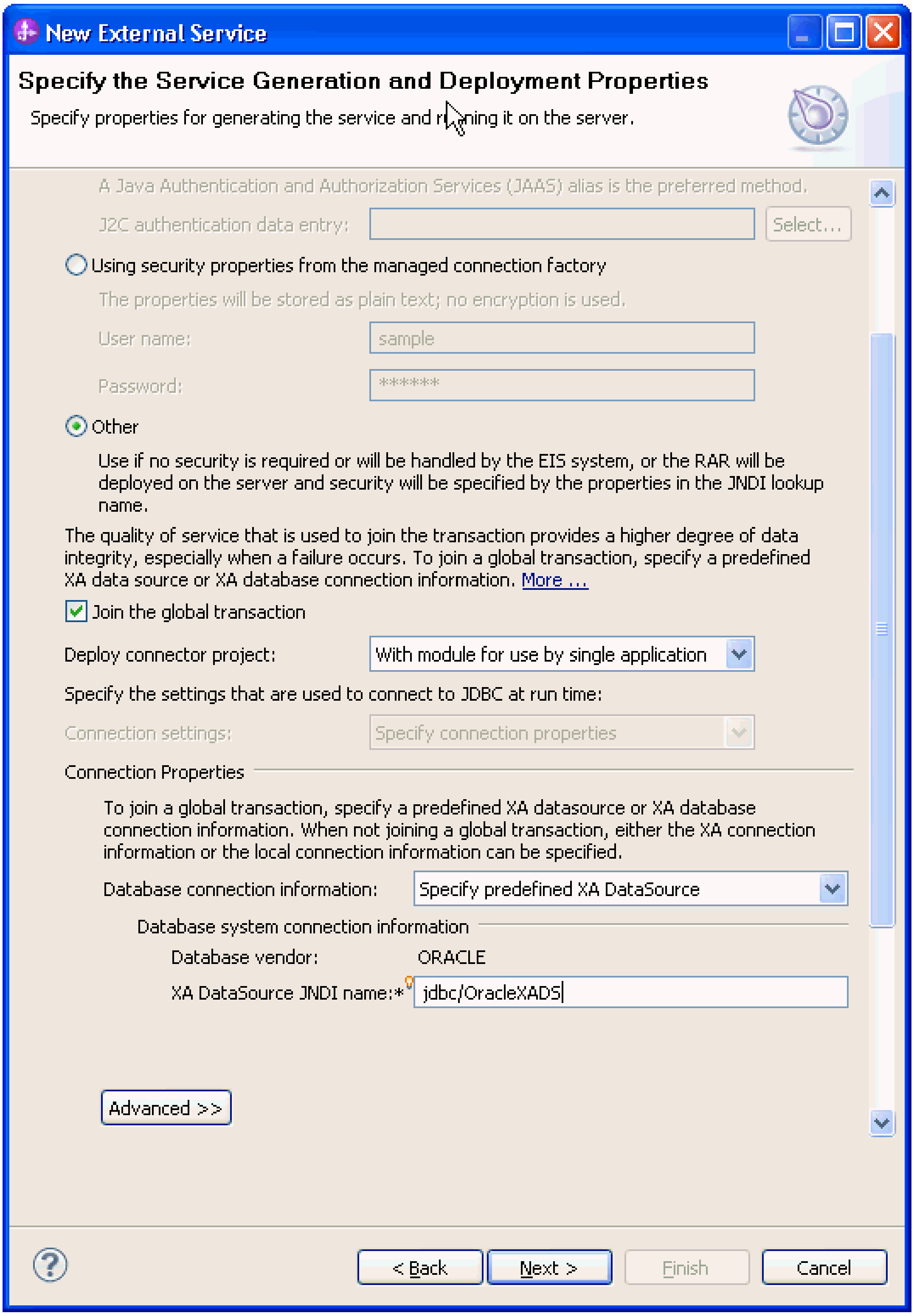
Business object namespace:* http://www.ibm.com/xmlns/prod/websphere/j2ca/jdb

Specify the relative folder for generated business objects

Folder:

☑ Generate a business graph for each business object

16. In **Specify the Service Generation and Deployment Properties** area, select **Other.** In the **XA DataSource JNDI name** field enter **jdbc/OracleXADS.** Click **Next.**



Specify the Service Generation and Deployment Properties

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

A Java Authentication and Authorization Services (JAAS) alias is the preferred method.

J2C authentication data entry: Select...

○ Using security properties from the managed connection factory

The properties will be stored as plain text; no encryption is used.

User name: sample

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 join a global transaction, specify a predefined XA data source or XA database connection information. More ...

☑ Join the global transaction

Deploy connector project: With module for use by single application

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

Connection settings: Specify connection properties

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.

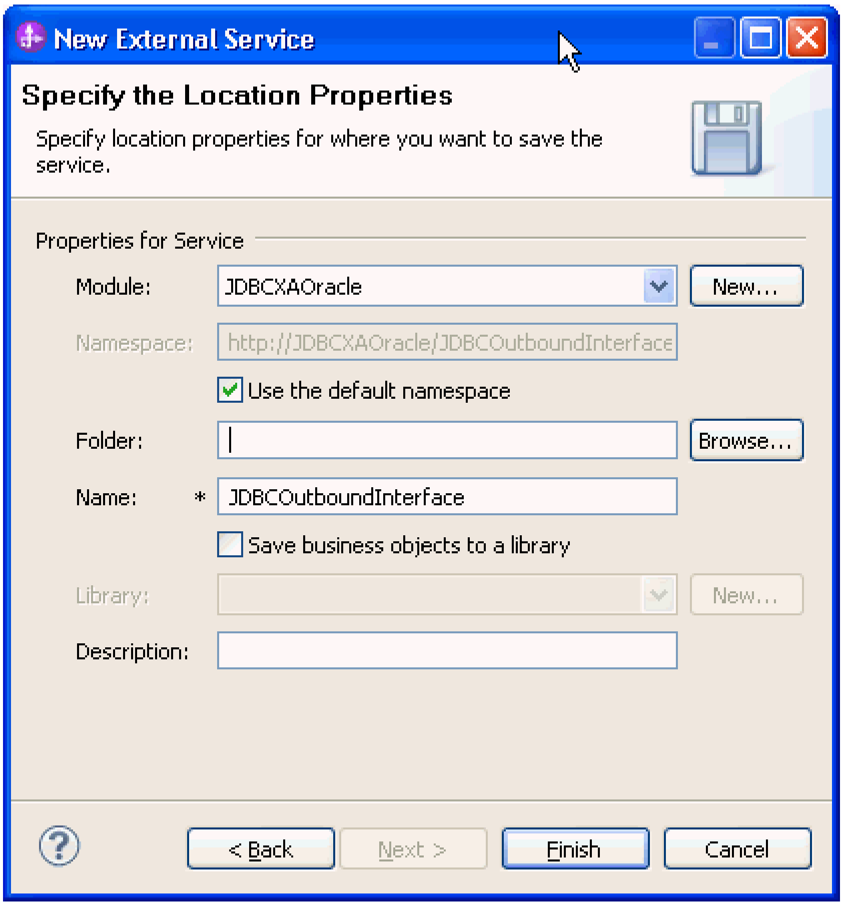
Database connection information: Specify predefined XA DataSource

Database system connection information

Database vendor: ORACLE

XA DataSource JNDI name:* jdbc/OracleXADS

17. In Specify the Location Properties window, click **Finish**.



New External Service

Specify the Location Properties

Specify location properties for where you want to save the service.

Properties for Service

Module: JDBCXAOracle

Namespace: http://JDBCXAOracle/JDBCOutboundInterface

☑ Use the default namespace

Folder:

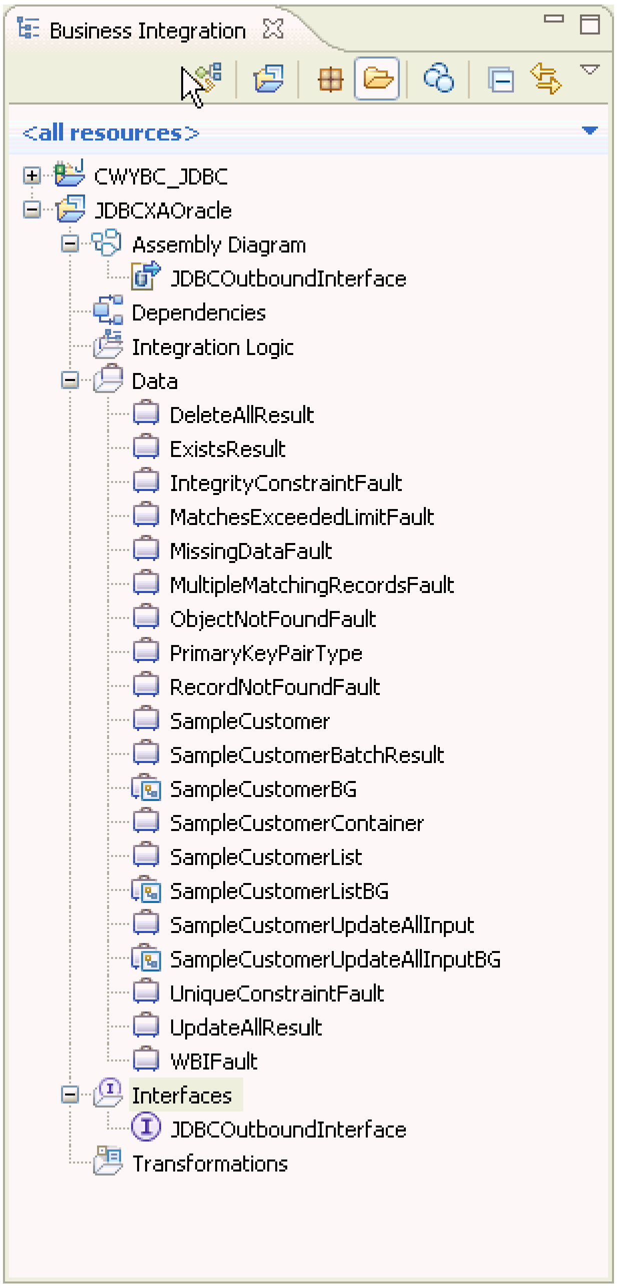
Name: * JDBCOutboundInterface

☐ Save business objects to a library

Library:

Description:

18. Verify the results shown below.



Business Integration

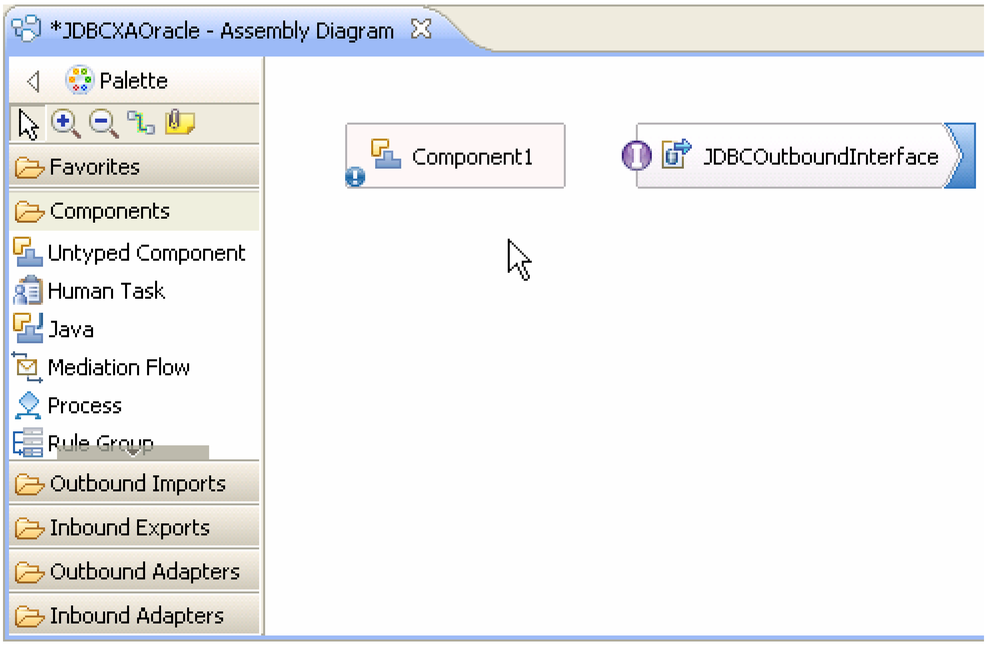
<all resources>

- CWYBC_JDBC
- JDBCXAOracle
  - Assembly Diagram
    - JDBCOutboundInterface
  - Dependencies
  - Integration Logic
  - Data
    - DeleteAllResult
    - ExistsResult
    - IntegrityConstraintFault
    - MatchesExceededLimitFault
    - MissingDataFault
    - MultipleMatchingRecordsFault
    - ObjectNotFoundFault
    - PrimaryKeyPairType
    - RecordNotFoundFault
    - SampleCustomer
    - SampleCustomerBatchResult
    - SampleCustomerBG
    - SampleCustomerContainer
    - SampleCustomerList
    - SampleCustomerListBG
    - SampleCustomerUpdateAllInput
    - SampleCustomerUpdateAllInputBG
    - UniqueConstraintFault
    - UpdateAllResult
    - WBIFault
  - Interfaces
    - JDBCOutboundInterface
  - Transformations

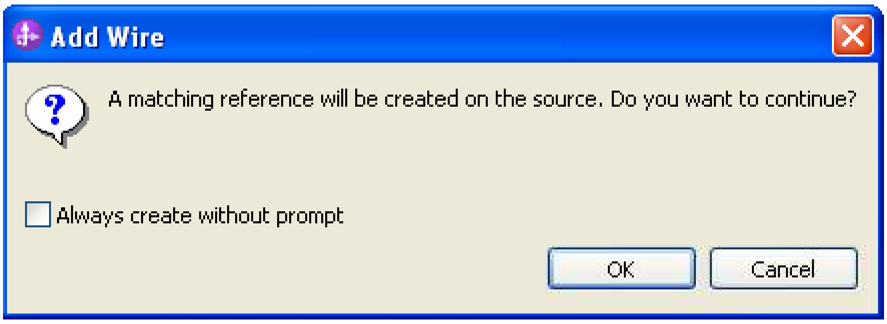
Set up the components to be part of the XA environment

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**.



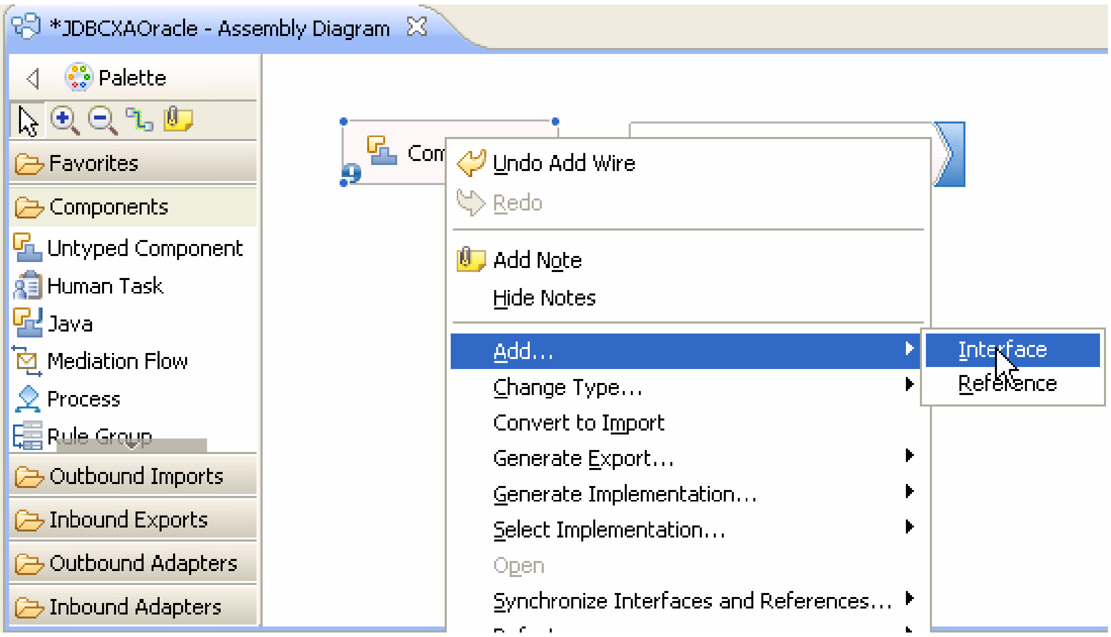
3. Wire **Component1** to **JDBCOutboundInterface**. In the Add Wire window, click **OK**.


Add Wire

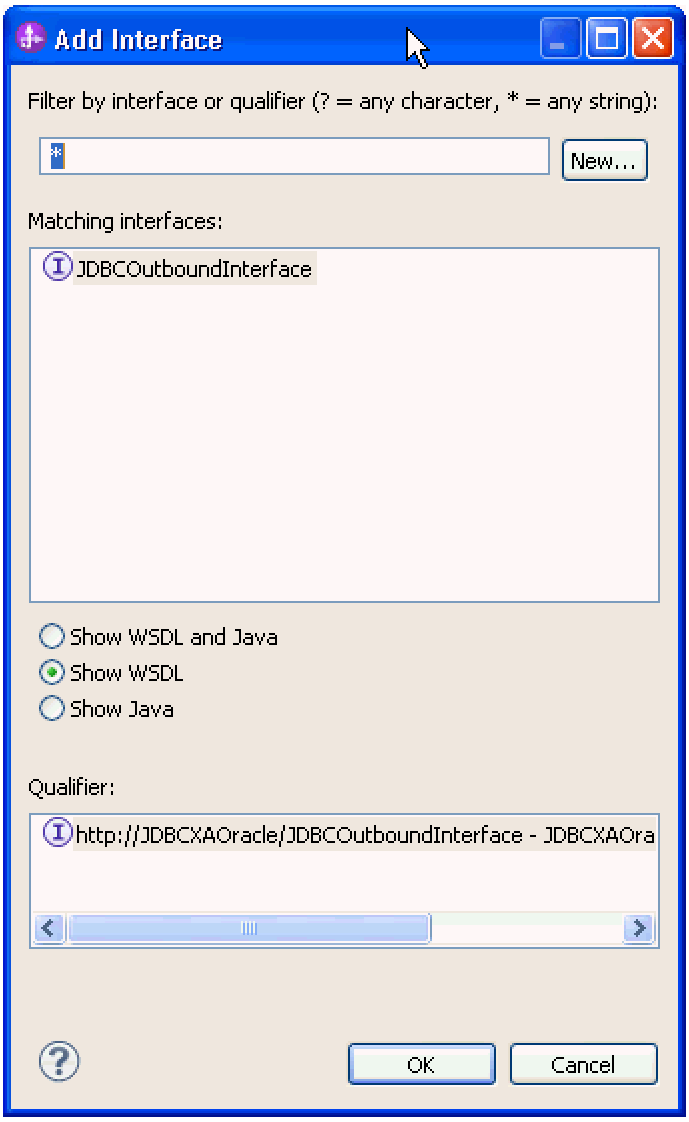
A matching reference will be created on the source. Do you want to continue?

□ Always create without prompt

4. Right-click **Component1** and select **Add > Interface**.



5. In the Add Interface window, click **New**.



Add Interface

Filter by interface or qualifier (? = any character, * = any string):

*

Matching interfaces:

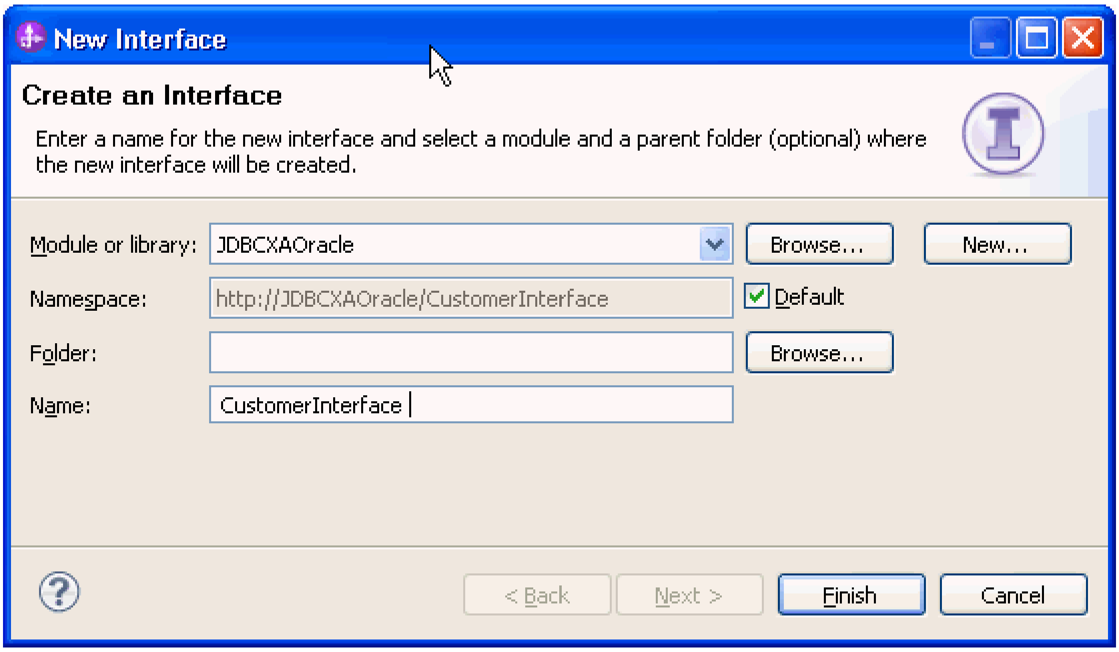
JDBCOutboundInterface

○ Show WSDL and Java
● Show WSDL
○ Show Java

Qualifier:

http://JDBCXAOracle/JDBCOutboundInterface - JDBCXAOra

6. Enter **CustomerInterface** in the **Name** field. Click **Finish**.



New Interface

Create an Interface

Enter a name for the new interface and select a module and a parent folder (optional) where the new interface will be created.

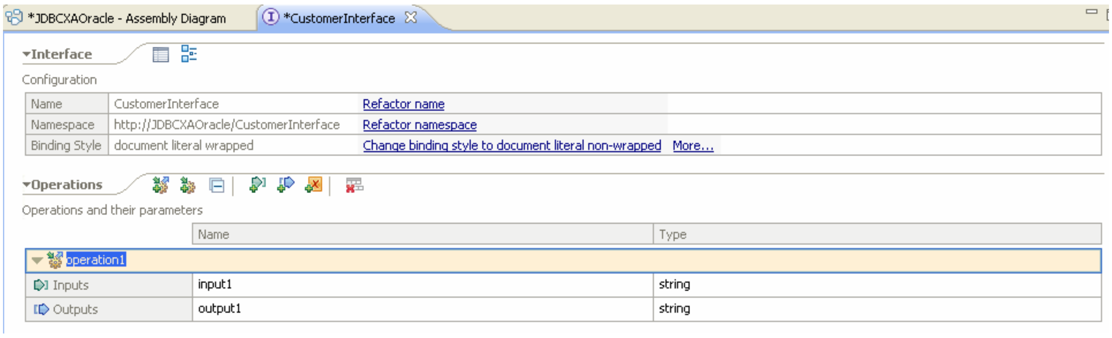
Module or library: JDBCXAOracle

Namespace: http://JDBCXAOracle/CustomerInterface ☑ Default

Folder:

Name: CustomerInterface

7. Click to add a new operation for **CustomerInterface** interface.



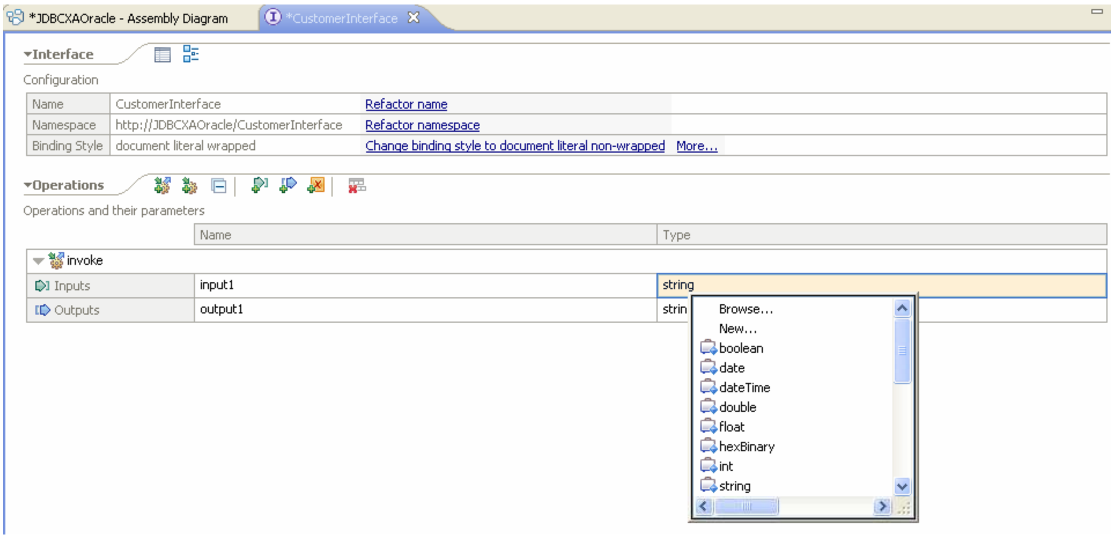
Configuration

Name	CustomerInterface	Refactor name
Namespace	http://JDBCXAOracle/CustomerInterface	Refactor namespace
Binding Style	document literal wrapped	Change binding style to document literal non-wrapped More...

Operations and their parameters

	Name	Type
operation1		
Inputs	input1	string
Outputs	output1	string

8. Rename the operation name to **invoke**. Select "Type" for the Inputs parameter, and select **New**.



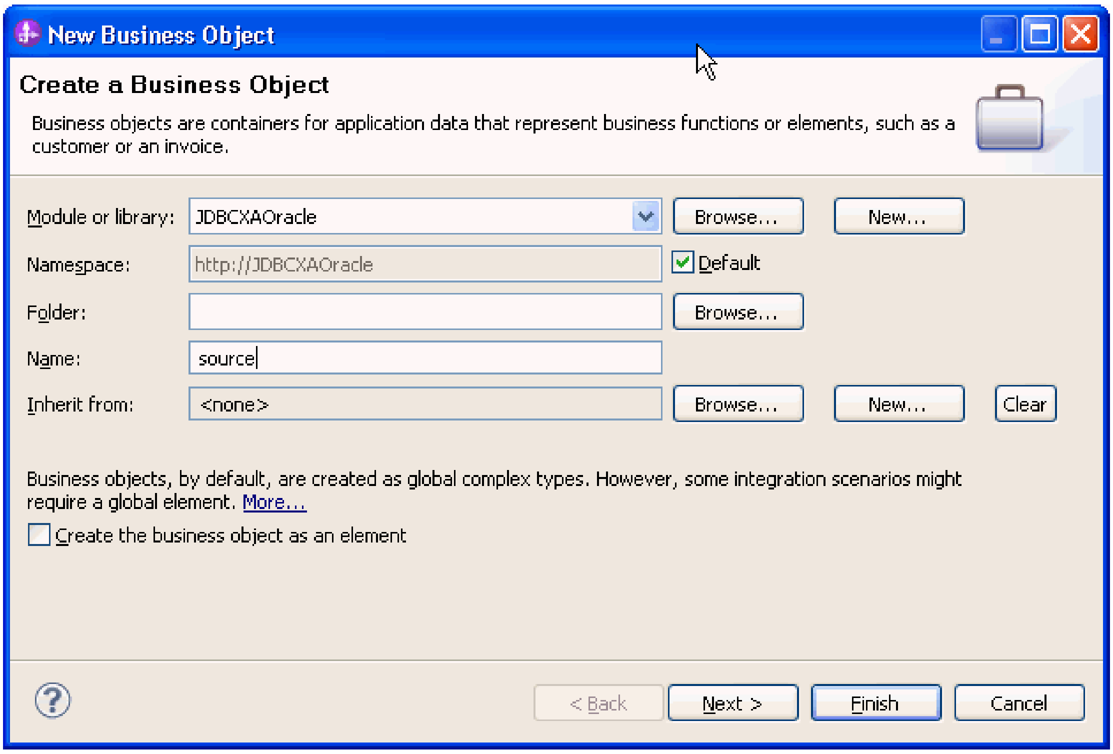
Configuration

Name	CustomerInterface	Refactor name
Namespace	http://JDBCXAOracle/CustomerInterface	Refactor namespace
Binding Style	document literal wrapped	Change binding style to document literal non-wrapped More...

Operations and their parameters

	Name	Type
invoke		
Inputs	input1	string
Outputs	output1	strin

9. In the Create a Business Object window, enter **Source** in the **Name** field. Click **Next**.



New Business Object

Create a Business Object

Business objects are containers for application data that represent business functions or elements, such as a customer or an invoice.

Module or library: JDBCXAOracle

Namespace: http://JDBCXAOracle ☑ Default

Folder:

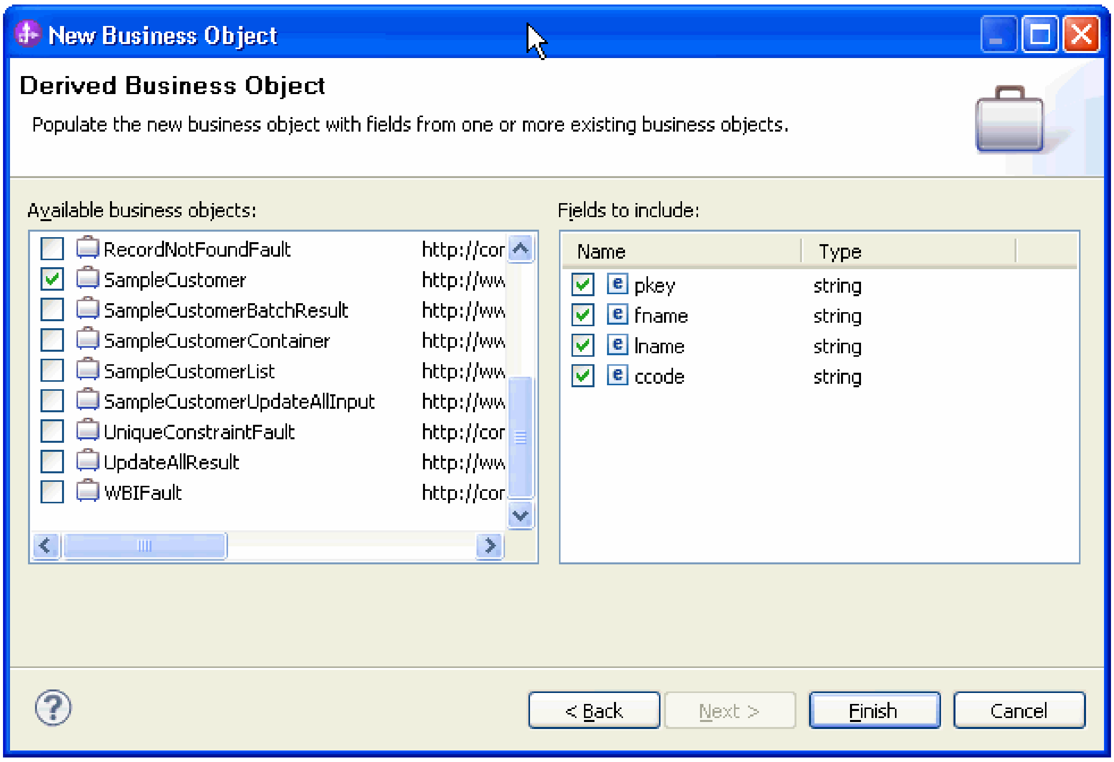
Name: source

Inherit from: <none>

Business objects, by default, are created as global complex types. However, some integration scenarios might require a global element. More...

□ Create the business object as an element

10. From the list of **Available business objects**, select **SampleCustomer** to add all Customer business objects's attributes to the Source business object. Click **Finish**.



New Business Object

Derived Business Object

Populate the new business object with fields from one or more existing business objects.

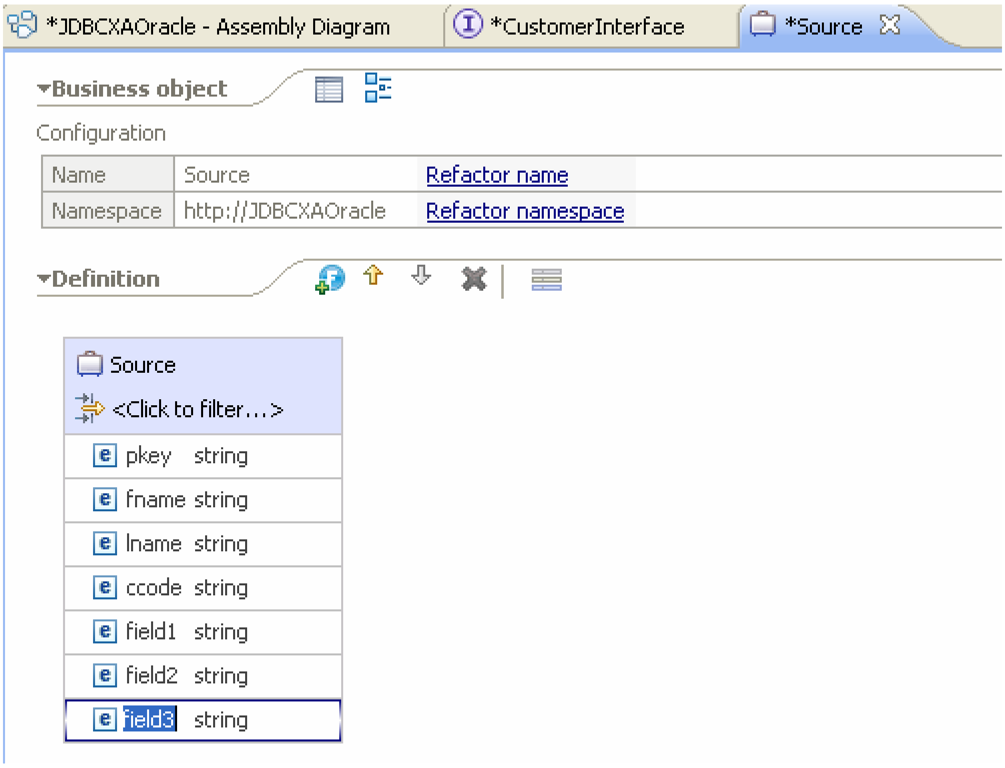
Available business objects:

□ RecordNotFoundFault http://cor
☑ SampleCustomer http://ww
□ SampleCustomerBatchResult http://ww
□ SampleCustomerContainer http://ww
□ SampleCustomerList http://ww
□ SampleCustomerUpdateAllInput http://ww
□ UniqueConstraintFault http://cor
□ UpdateAllResult http://ww
□ WBIFault http://cor

Fields to include:

Name	Type
☑ pkey	string
☑ fname	string
☑ lname	string
☑ ccode	string

11. In the Business Object editor, click to add three new fields for **Source** business object.

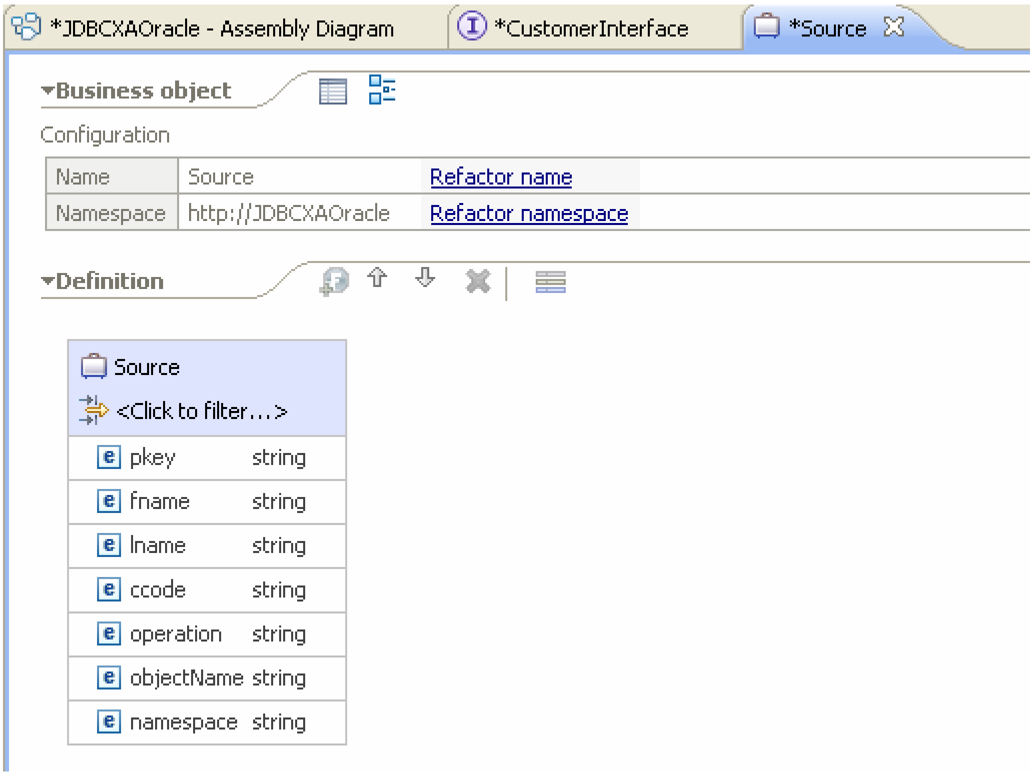


Configuration

Name	Source	Refactor name
Namespace	http://JDBCXAOracle	Refactor namespace

Source	
pkey	string
fname	string
lname	string
ccode	string
field1	string
field2	string
field3	string

12. Rename the three new fields as **operation**, **objectName** and **namespace**.



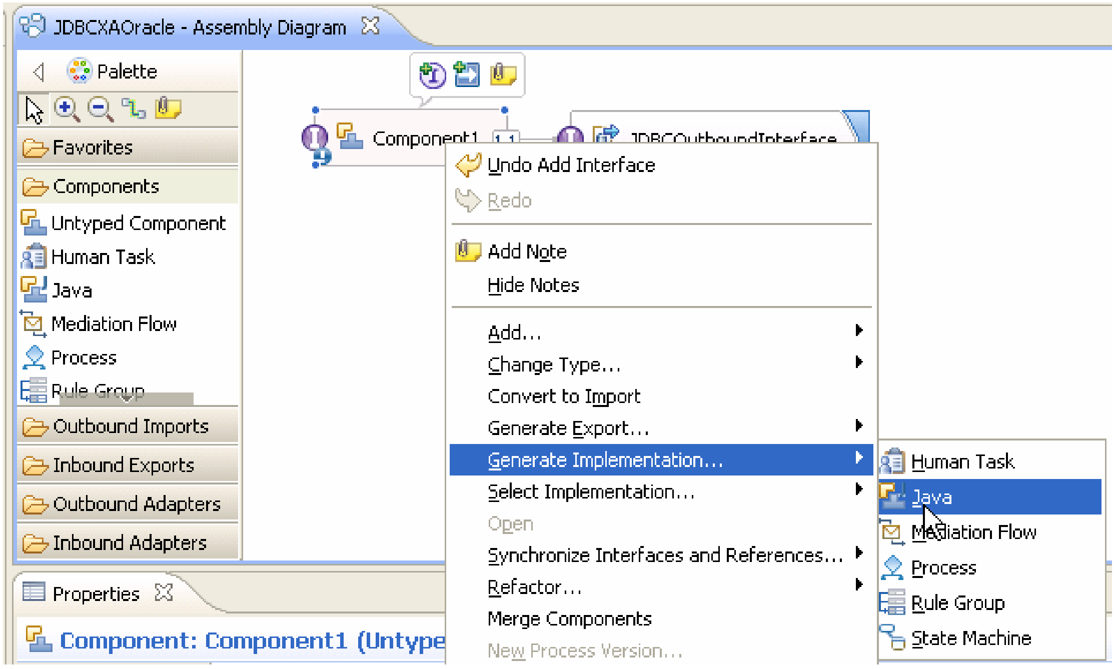
Configuration

Name	Source	Refactor name
Namespace	http://JDBCXAOracle	Refactor namespace

Source	
pkey	string
fname	string
lname	string
ccode	string
operation	string
objectName	string
namespace	string

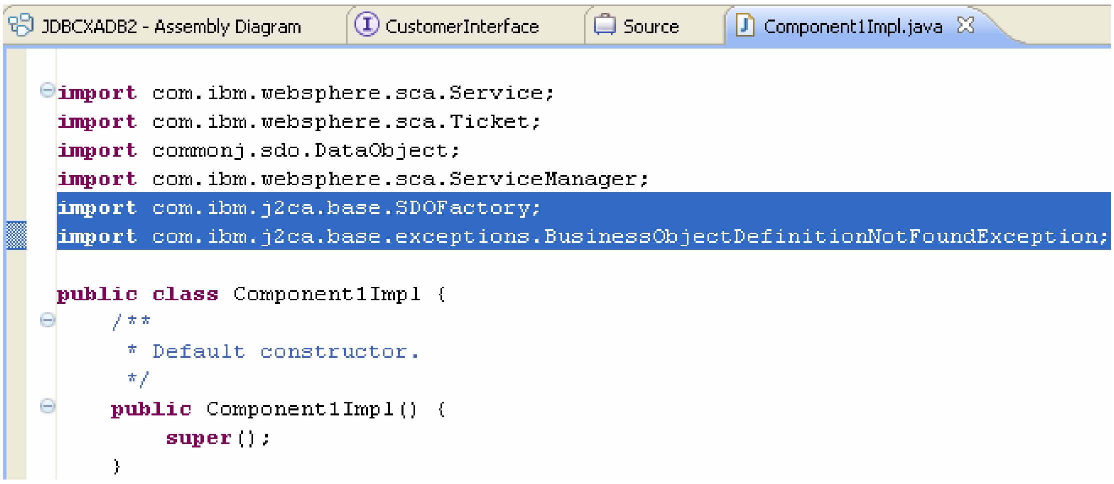
13. Select **File->Save All** to save all the changes.

14. Right click **Component1** in the Assembly Diagram and select **Generate implementation -> Java.**



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**.**

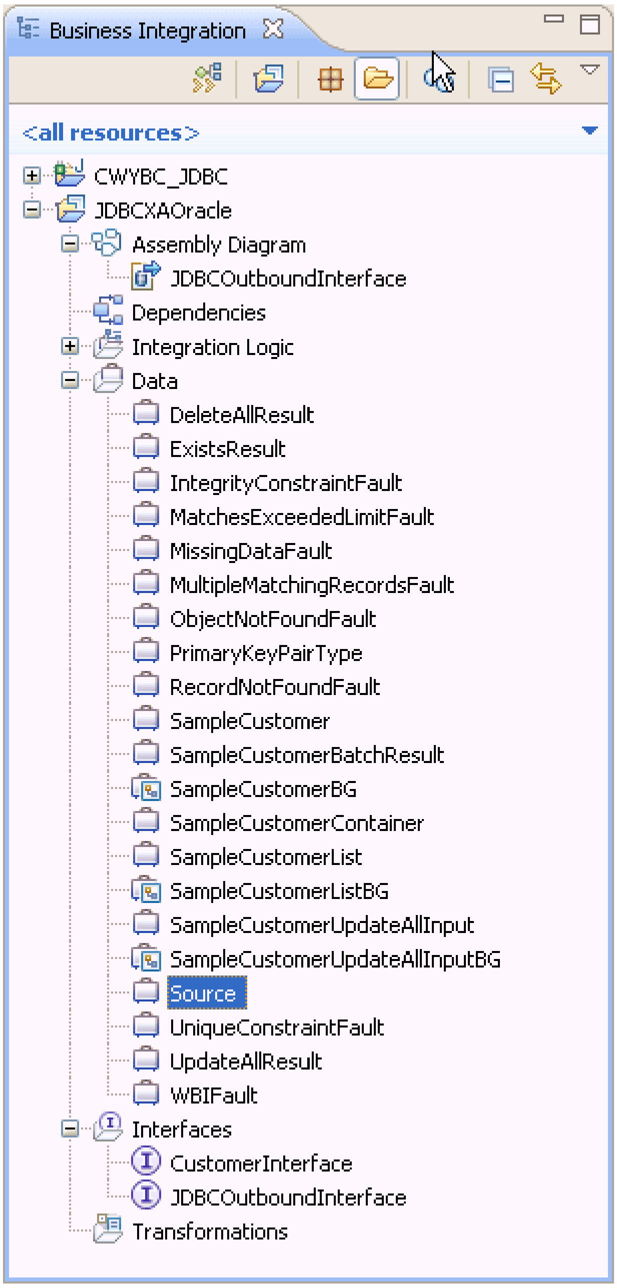


```
import com.ibm.websphere.sca.Service;
import com.ibm.websphere.sca.Ticket;
import commonj.sdo.DataObject;
import com.ibm.websphere.sca.ServiceManager;
import com.ibm.j2ca.base.SDOFactory;
import com.ibm.j2ca.base.exceptions.BusinessObjectDefinitionNotFoundException;

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.



Business Integration

<all resources>

- CWYBC_JDBC
- JDBCXAOracle
  - Assembly Diagram
    - JDBCOutboundInterface
  - Dependencies
  - Integration Logic
  - Data
    - DeleteAllResult
    - ExistsResult
    - IntegrityConstraintFault
    - MatchesExceededLimitFault
    - MissingDataFault
    - MultipleMatchingRecordsFault
    - ObjectNotFoundFault
    - PrimaryKeyPairType
    - RecordNotFoundFault
    - SampleCustomer
    - SampleCustomerBatchResult
    - SampleCustomerBG
    - SampleCustomerContainer
    - SampleCustomerList
    - SampleCustomerListBG
    - SampleCustomerUpdateAllInput
    - SampleCustomerUpdateAllInputBG
    - Source
    - UniqueConstraintFault
    - UpdateAllResult
    - WBIFault
  - Interfaces
    - CustomerInterface
    - JDBCOutboundInterface
  - Transformations

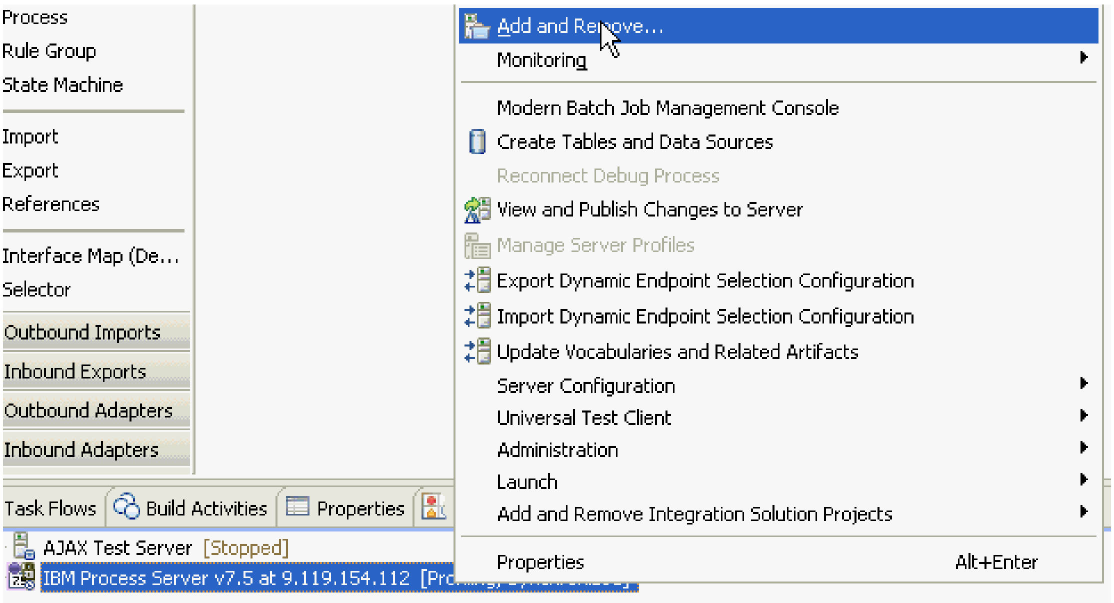
Deploy the module to the test environment

The result of running the external service wizard is an SCA module that contains an Enterprise Information System import. Install this SCA module in IBM Integration Designer integration test client. To do this, you must add the SCA module you created earlier to the server using the **Servers** view in IBM Integration Designer.

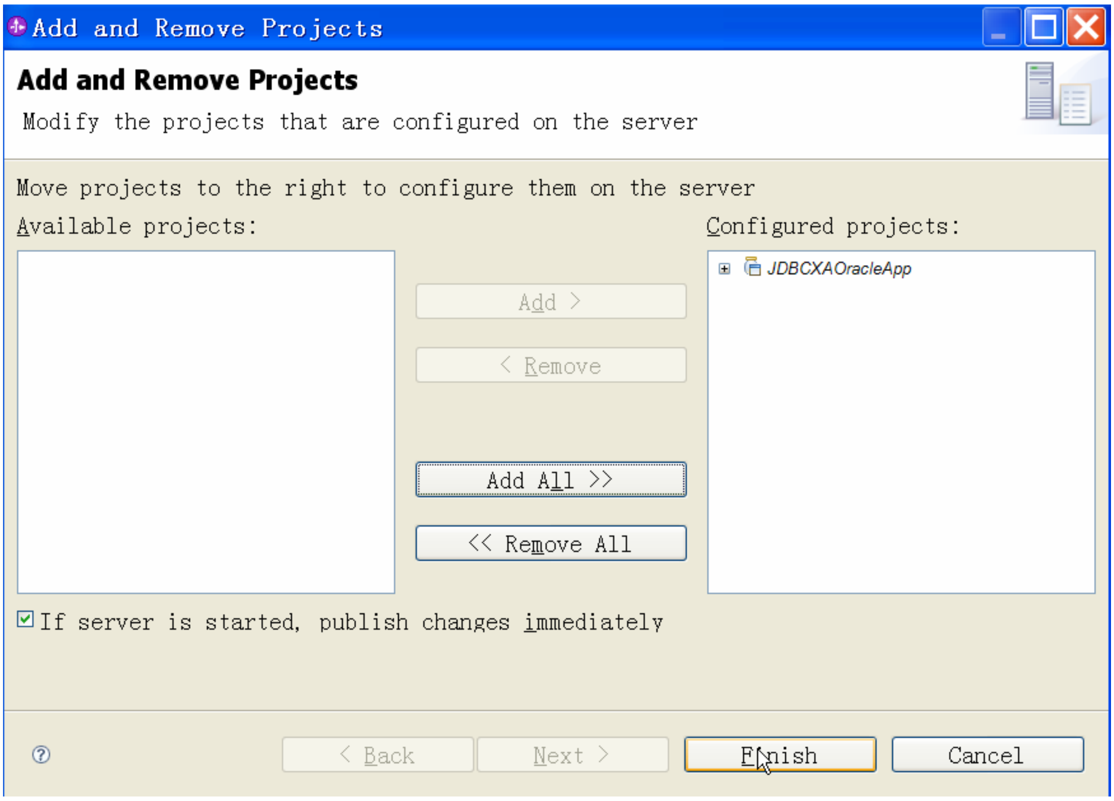
Steps for adding the SCA module to the server:

1. In IBM Integration Designer, 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**.


Process
Rule Group
State Machine
Import
Export
References
Interface Map (De...
Selector
Outbound Imports
Inbound Exports
Outbound Adapters
Inbound Adapters
Task Flows
Build Activities
Properties
AJAX Test Server [Stopped]
IBM Process Server v7.5 at 9.119.154.112 [Pro
Add and Remove...
Monitoring
Modern Batch Job Management Console
Create Tables and Data Sources
Reconnect Debug Process
View and Publish Changes to Server
Manage Server Profiles
Export Dynamic Endpoint Selection Configuration
Import Dynamic Endpoint Selection Configuration
Update Vocabularies and Related Artifacts
Server Configuration
Universal Test Client
Administration
Launch
Add and Remove Integration Solution Projects
Properties Alt+Enter

4. Add **JDBCXAOracleApp** into the **Configured projects** panel. Click **Finish**.


Add and Remove Projects

Add and Remove Projects

Modify the projects that are configured on the server

Move projects to the right to configure them on the server

Available projects:

Configured projects:

JDBCXAOracleApp

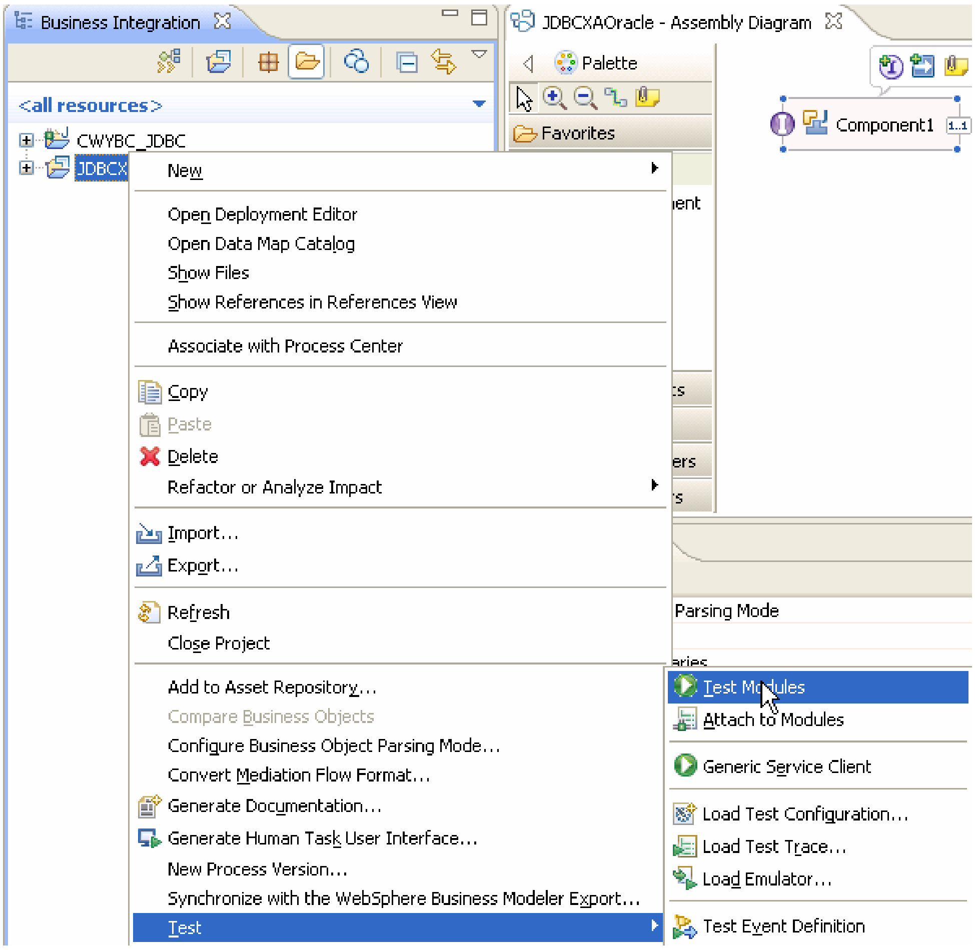
☑ If server is started, publish changes immediately

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

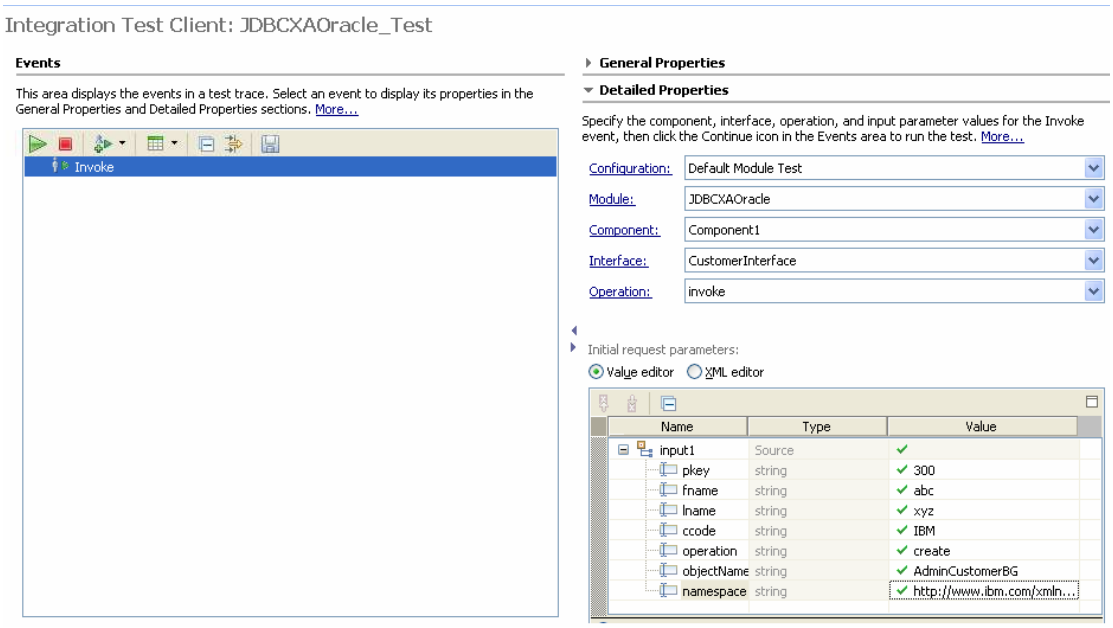
Test the assembled adapter application using the IBM Integration Designer integration test client:

1. From the Business Integration view, right click **JDBCXAOracle** and select **Test > Test Module.**



Business Integration
<all resources>
CWYBC_JDBC
JDBCX
JDBCXAOracle - Assembly Diagram
Palette
Favorites
Component1
New
Open Deployment Editor
Open Data Map Catalog
Show Files
Show References in References View
Associate with Process Center
Copy
Paste
Delete
Refactor or Analyze Impact
Import...
Export...
Refresh
Close Project
Add to Asset Repository...
Compare Business Objects
Configure Business Object Parsing Mode...
Convert Mediation Flow Format...
Generate Documentation...
Generate Human Task User Interface...
New Process Version...
Synchronize with the WebSphere Business Modeler Export...
Test
Test Modules
Attach to Modules
Generic Service Client
Load Test Configuration...
Load Test Trace...
Load Emulator...
Test Event Definition
Parsing Mode

2. 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...

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: JDBCXAOracle

Component: Component1

Interface: CustomerInterface

Operation: invoke

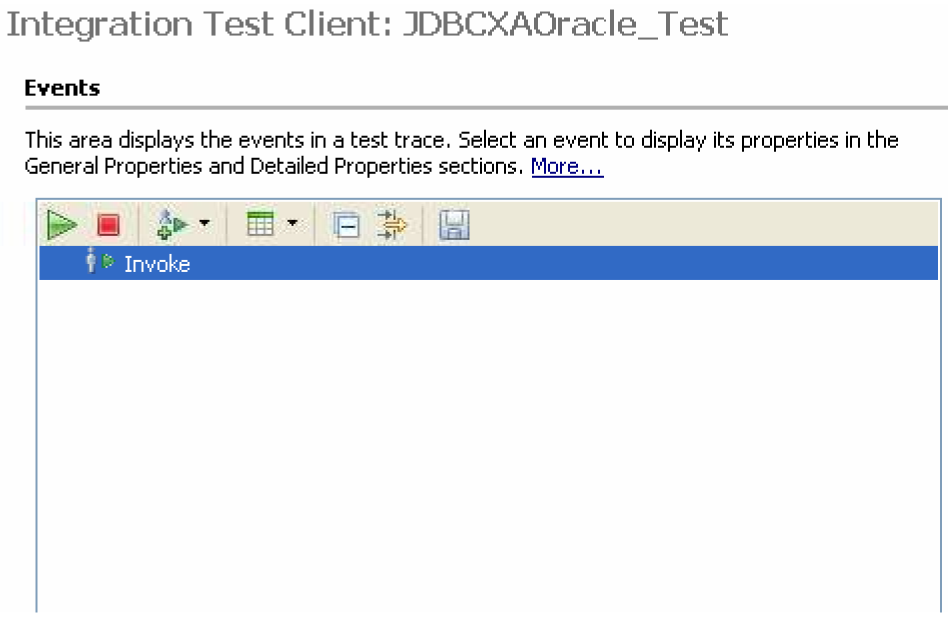
Initial request parameters:

● 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/xmln...

Note: Set the **operation** value to **Create**; set the **objectName** value to **SampleCustomerBG**; set the **namespace** value to **http://www.ibm.com/xmlns/prod/websphere/j2ca/jdbc/Samplecustomerbg**.

3. Click ▶ to continue.



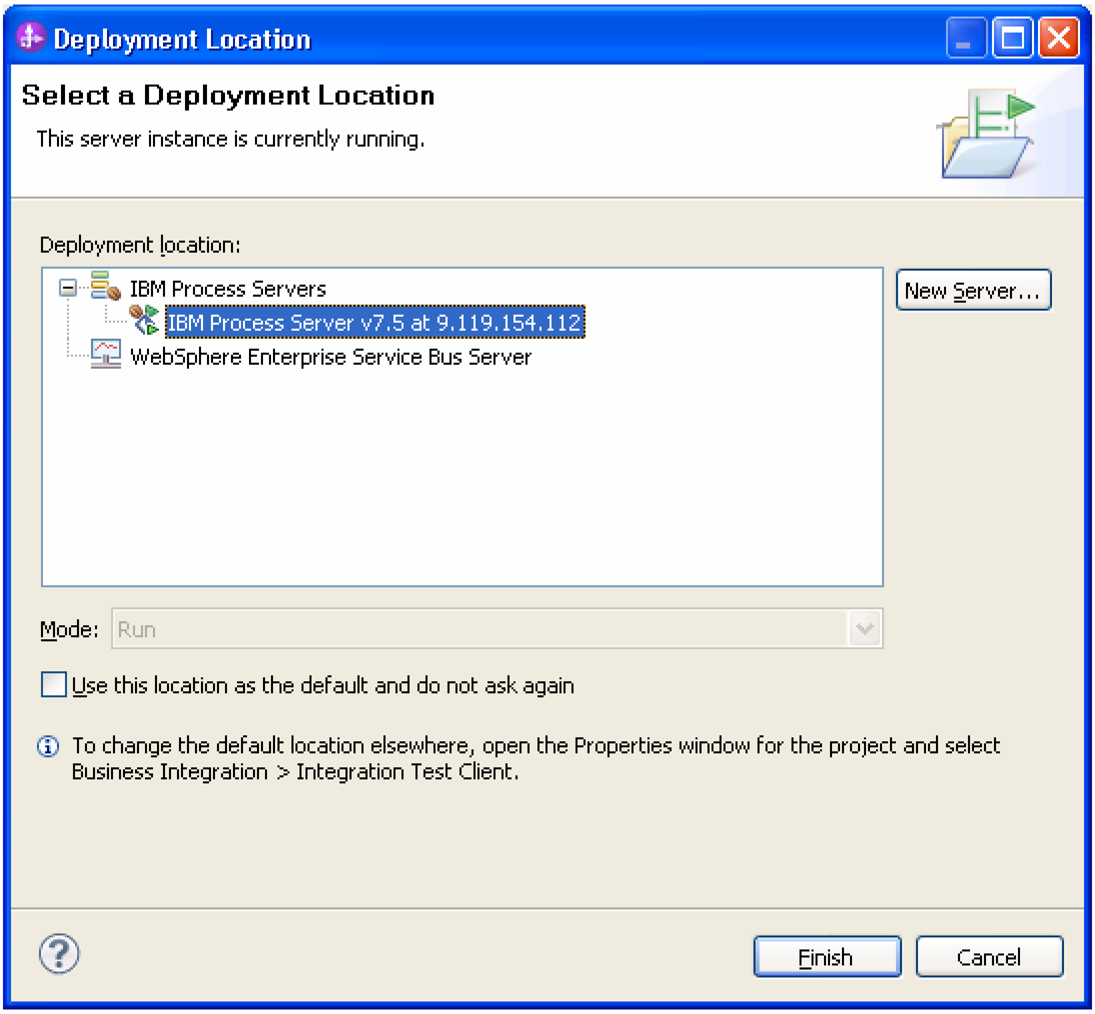
Integration Test Client: JDBCXAOracle_Test

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...

Invoke

4. In the Select Deployment Location window, select the server, and click **Finish**.



Deployment Location

Select a Deployment Location

This server instance is currently running.

Deployment location:

- IBM Process Servers
  - IBM Process Server v7.5 at 9.119.154.112
- WebSphere Enterprise Service Bus Server

Mode: Run

□ Use this location as the default and do not ask again

To change the default location elsewhere, open the Properties window for the project and select Business Integration > Integration Test Client.

6. 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';
```

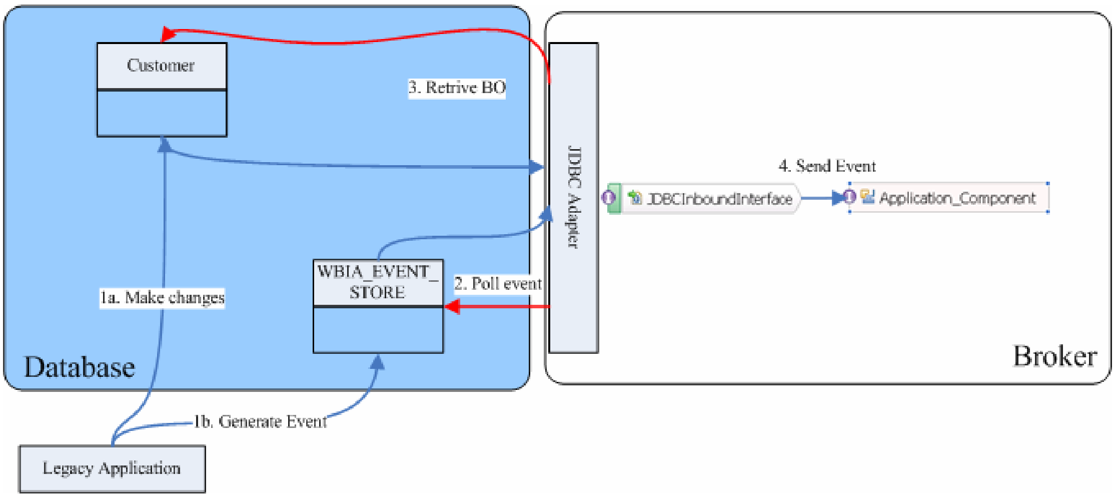
Chapter 10. Tutorial 9: Receiving events from the Oracle database using data source with prepared statement cache (inbound processing)

This tutorial demonstrates how the WebSphere Adapter for JDBC 7.5.0.0 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             NUMBER(20)  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     NUMBER(5)  NOT NULL,
   event_time        DATE  DEFAULT SYSDATE NOT NULL,

   event_status         NUMBER(5)   NOT NULL,
   event_timeout TIMESTAMP,
   connector_ID            VARCHAR2(40),
   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;
```

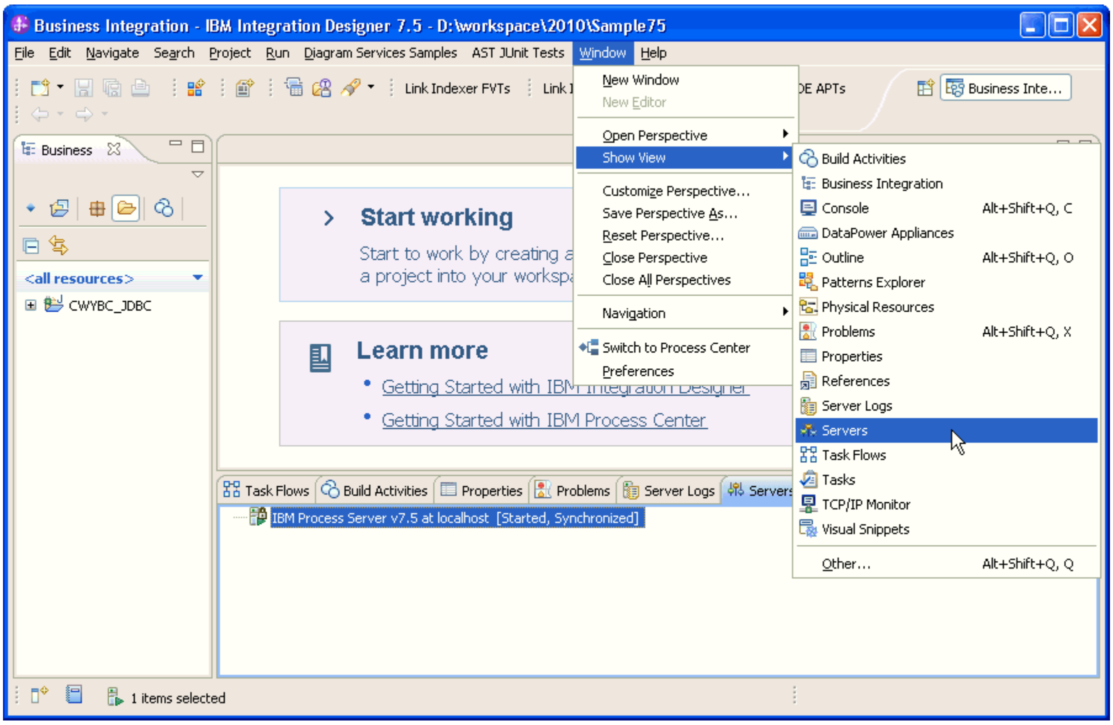
Note: After running this procedure, verify whether the records are inserted correctly into the WBIA_JDBC_EVENTSTORE table.

Create an authentication alias

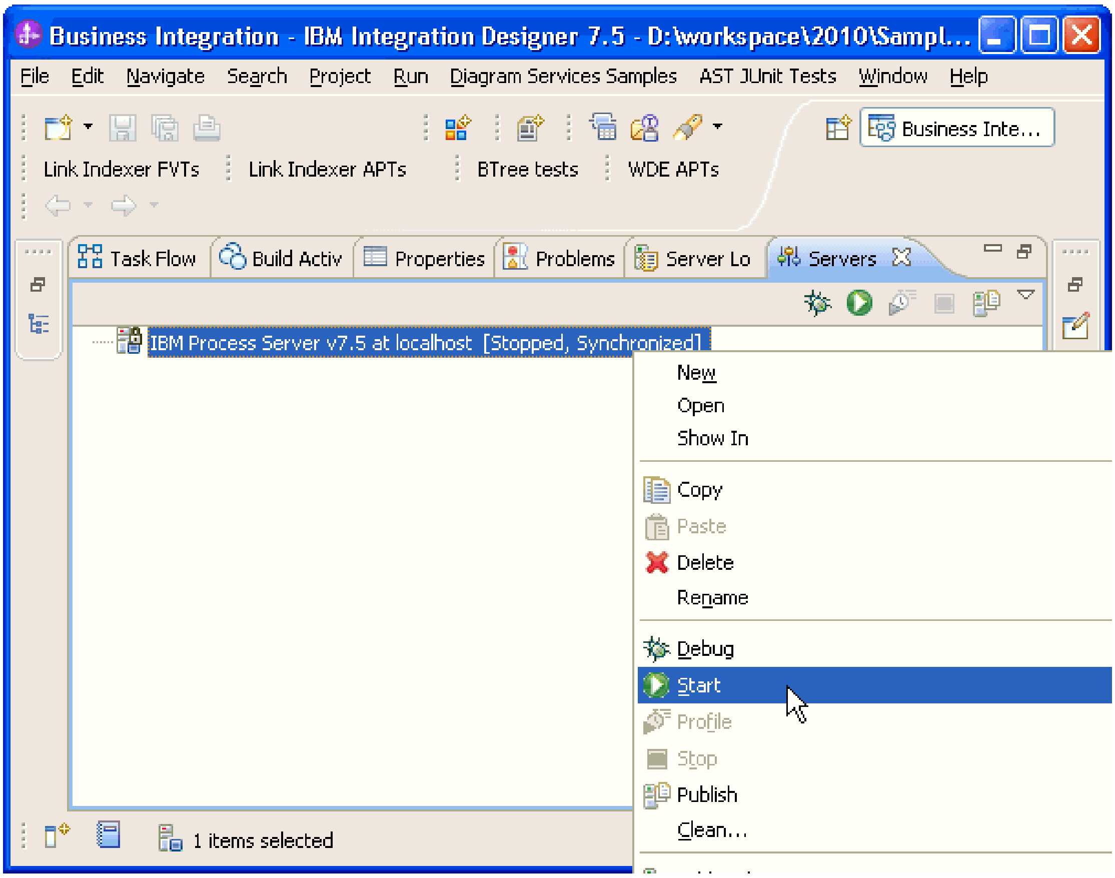
The authentication alias needs to be set because the data source created in the next section uses the username and password set in the authentication alias to connect to the database.

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

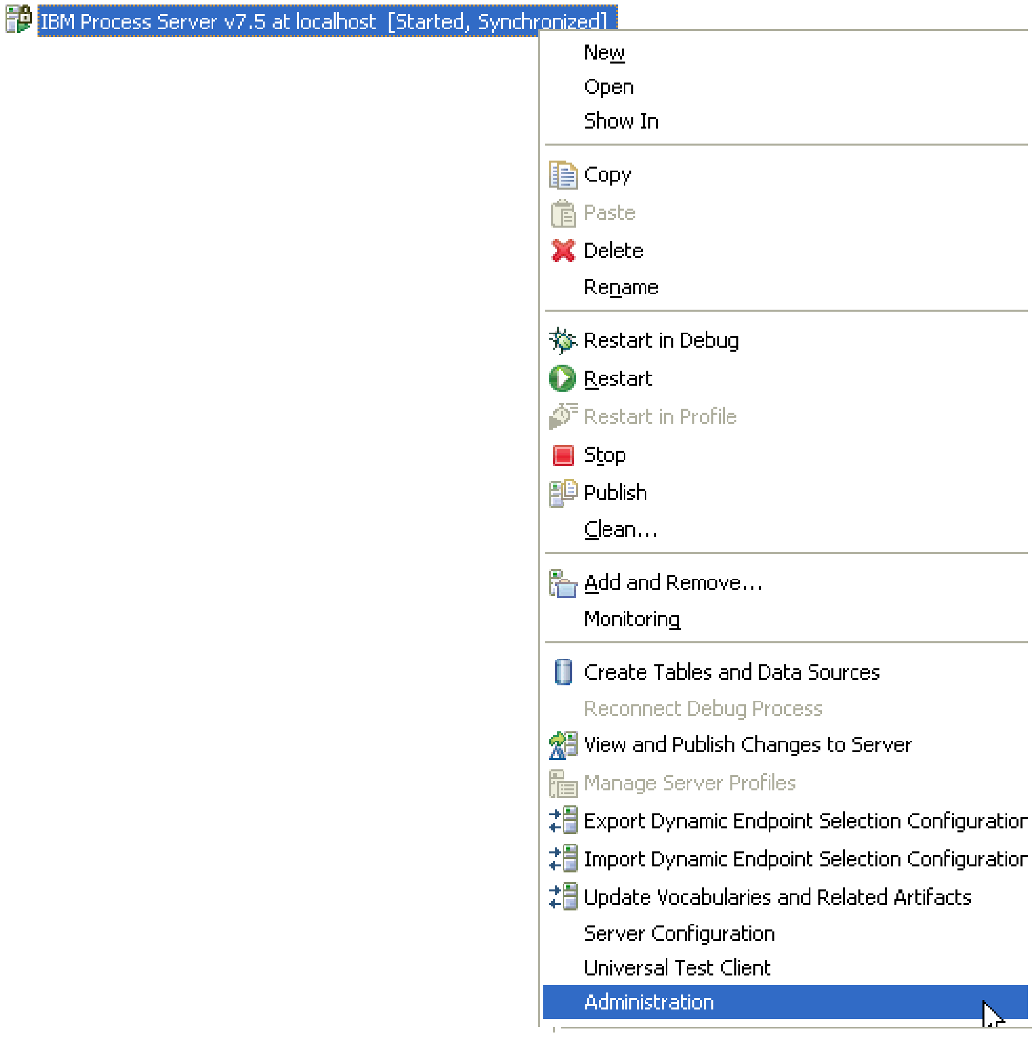
1. In IBM Integration Designer, switch to the **Servers** view by selecting **Windows > Show View > Servers**.



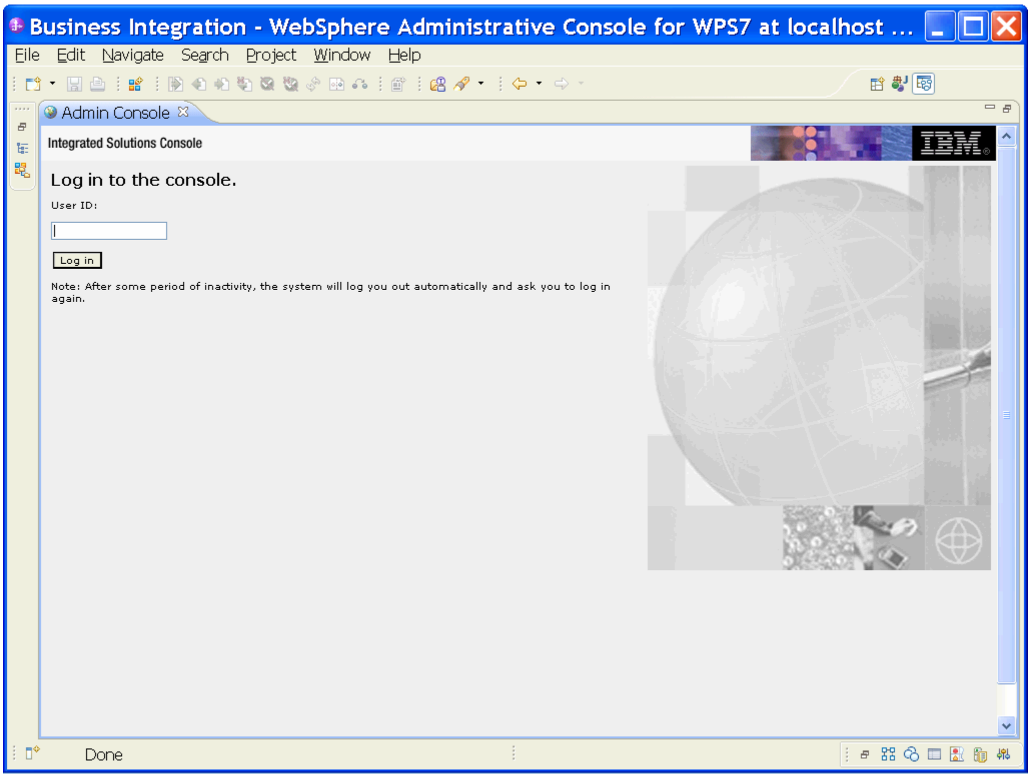
2. In the **Servers** view, right-click the server that you want to start and select **Start**.


Business Integration - IBM Integration Designer 7.5 - D:\workspace\2010\Sampl...
File Edit Navigate Search Project Run Diagram Services Samples AST JUnit Tests Window Help
Business Inte...
Link Indexer FVTs
Link Indexer APTs
BTree tests
WDE APTs
Task Flow
Build Activ
Properties
Problems
Server Lo
Servers
IBM Process Server v7.5 at localhost [Stopped, Synchronized]
New
Open
Show In
Copy
Paste
Delete
Rename
Debug
Start
Profile
Stop
Publish
Clean...
1 items selected

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


IBM Process Server v7.5 at localhost [Started, Synchronized]
New
Open
Show In
Copy
Paste
Delete
Rename
Restart in Debug
Restart
Restart in Profile
Stop
Publish
Clean...
Add and Remove...
Monitoring
Create Tables and Data Sources
Reconnect Debug Process
View and Publish Changes to Server
Manage Server Profiles
Export Dynamic Endpoint Selection Configuratior
Import Dynamic Endpoint Selection Configuratior
Update Vocabularies and Related Artifacts
Server Configuration
Universal Test Client
Administration

4. Log on to the administrative console.

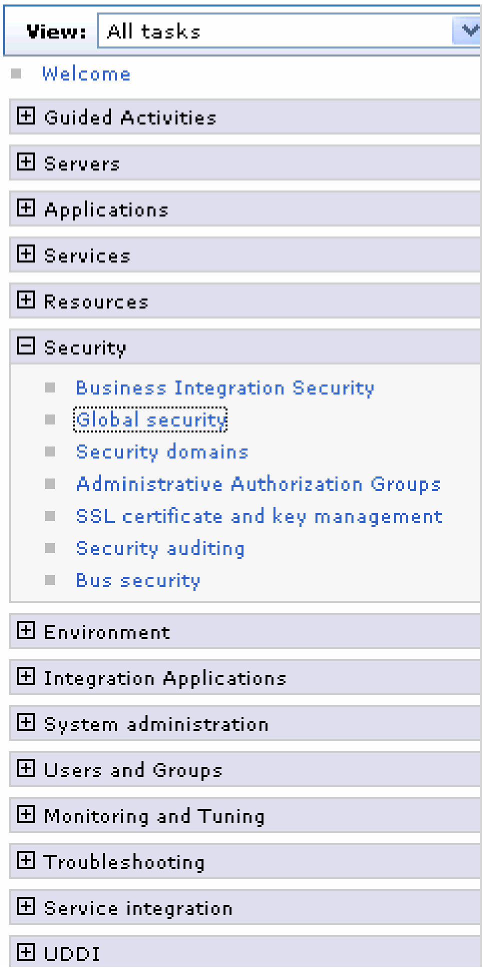

Business Integration - WebSphere Administrative Console for WPS7 at localhost ...
File Edit Navigate Search Project Window Help
Admin Console
Integrated Solutions Console
IBM
Log in
Done

Log in to the console.

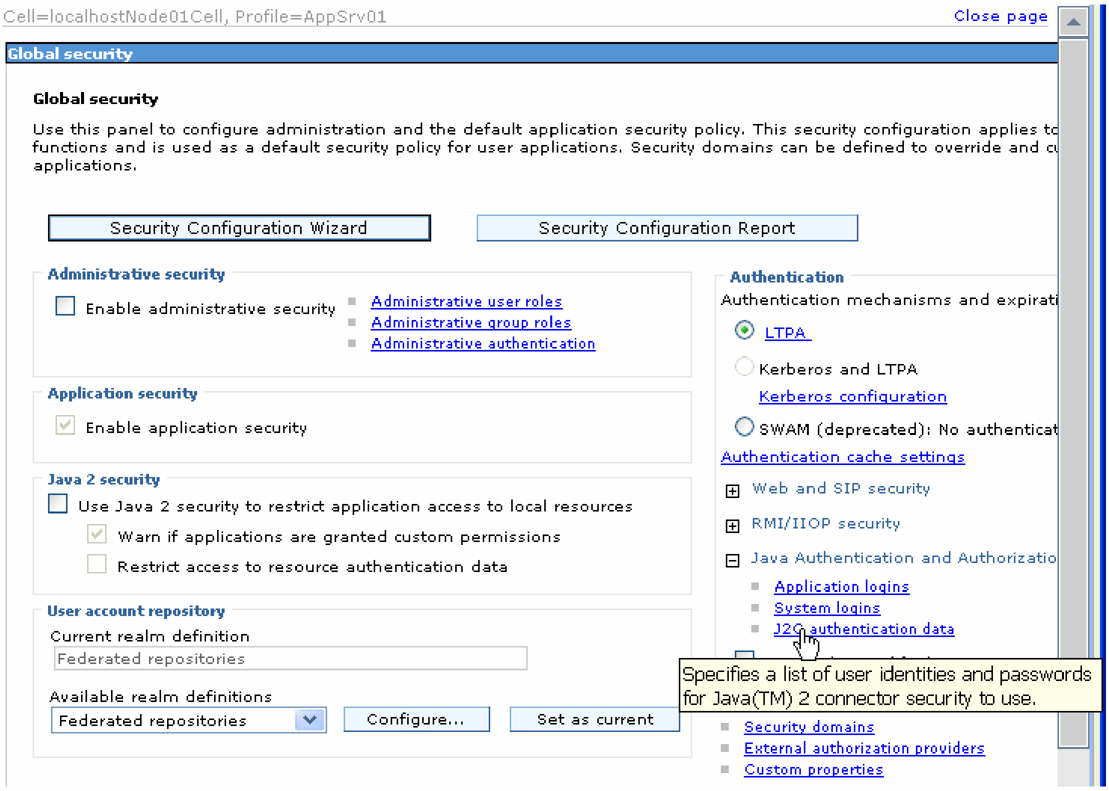
User ID:

Note: After some period of inactivity, the system will log you out automatically and ask you to log in again.

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



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 cu 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 expirati

● LTPA

○ Kerberos and LTPA

Kerberos configuration

○ SWAM (deprecated): No authenticat

Authentication cache settings

⊞ Web and SIP security

⊞ RMI/IIOP security

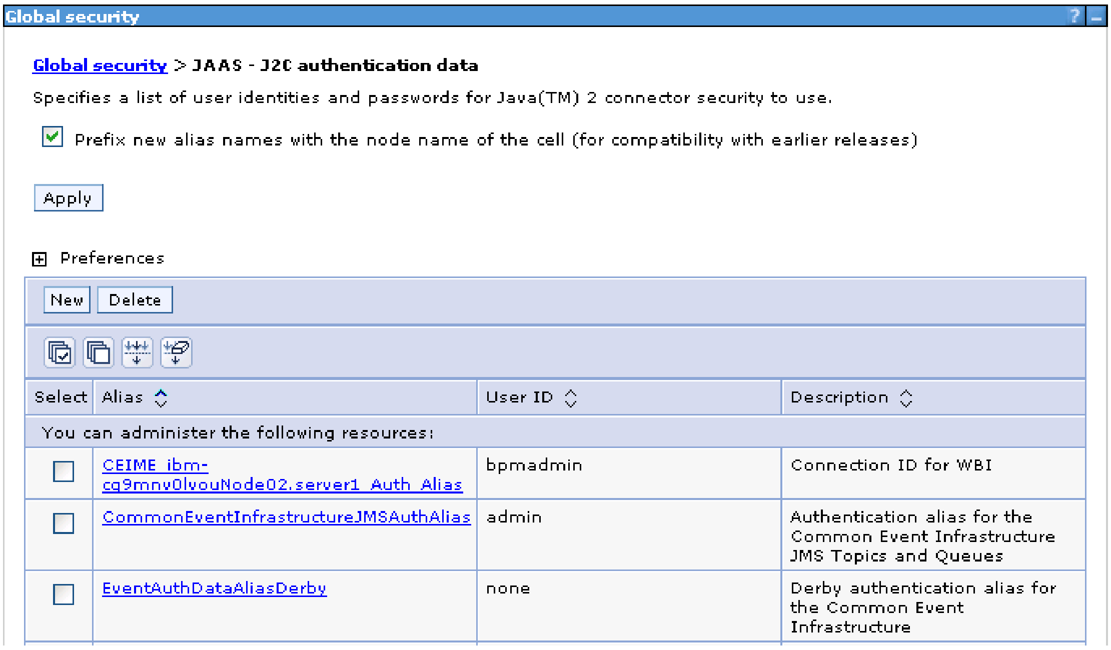
⊟ Java Authentication and Authorizatio

- 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.



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)

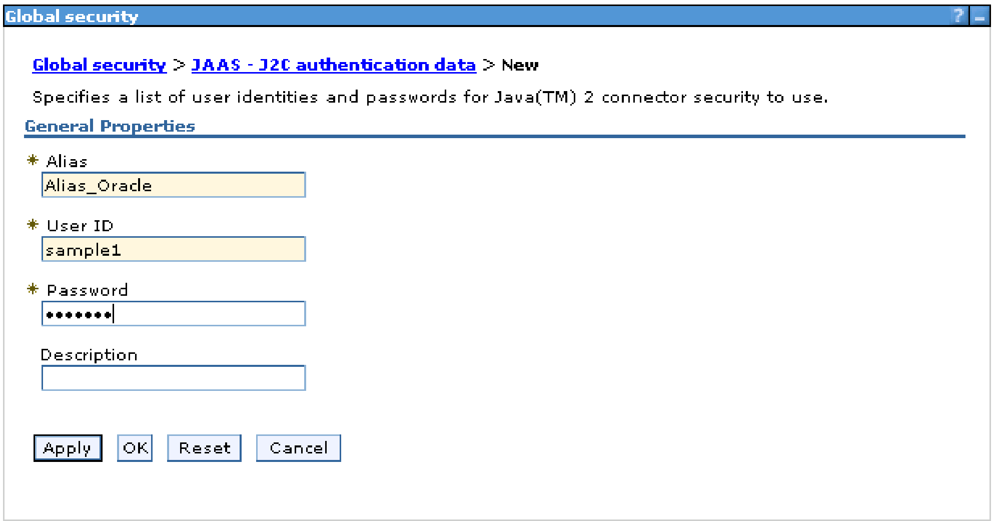
Apply

⊞ Preferences

New | Delete

Select	Alias	User ID	Description
You can administer the following resources:			
□	CEIME_ibm-cq9mnv0lvouNode02.server1_Auth_Alias	bpmadmin	Connection ID for WBI
□	CommonEventInfrastructureJMSAuthAlias	admin	Authentication alias for the Common Event Infrastructure JMS Topics and Queues
□	EventAuthDataAliasDerby	none	Derby authentication alias for the Common Event Infrastructure

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

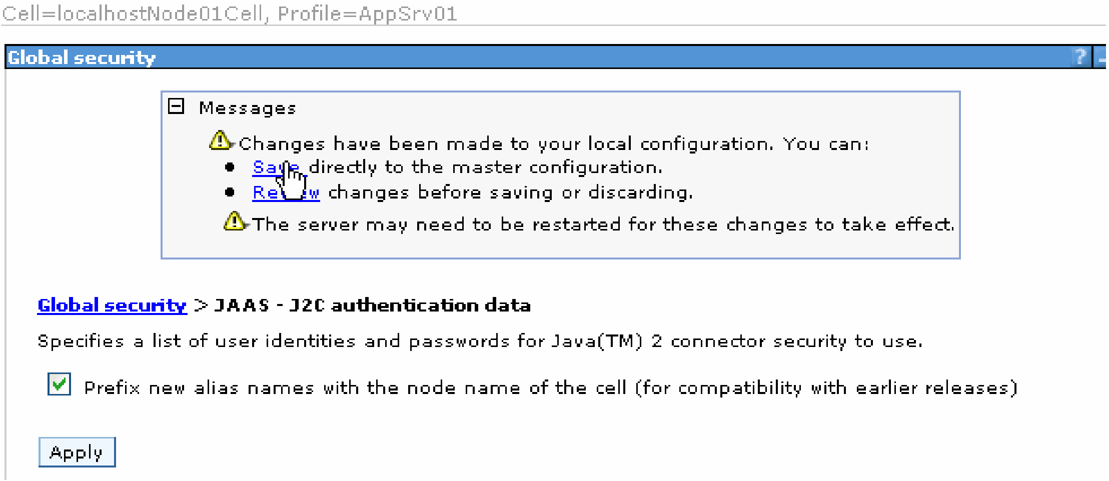
* Alias: Alias_Oracle

* User ID: sample1

* Password: •••••••

Description:

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.

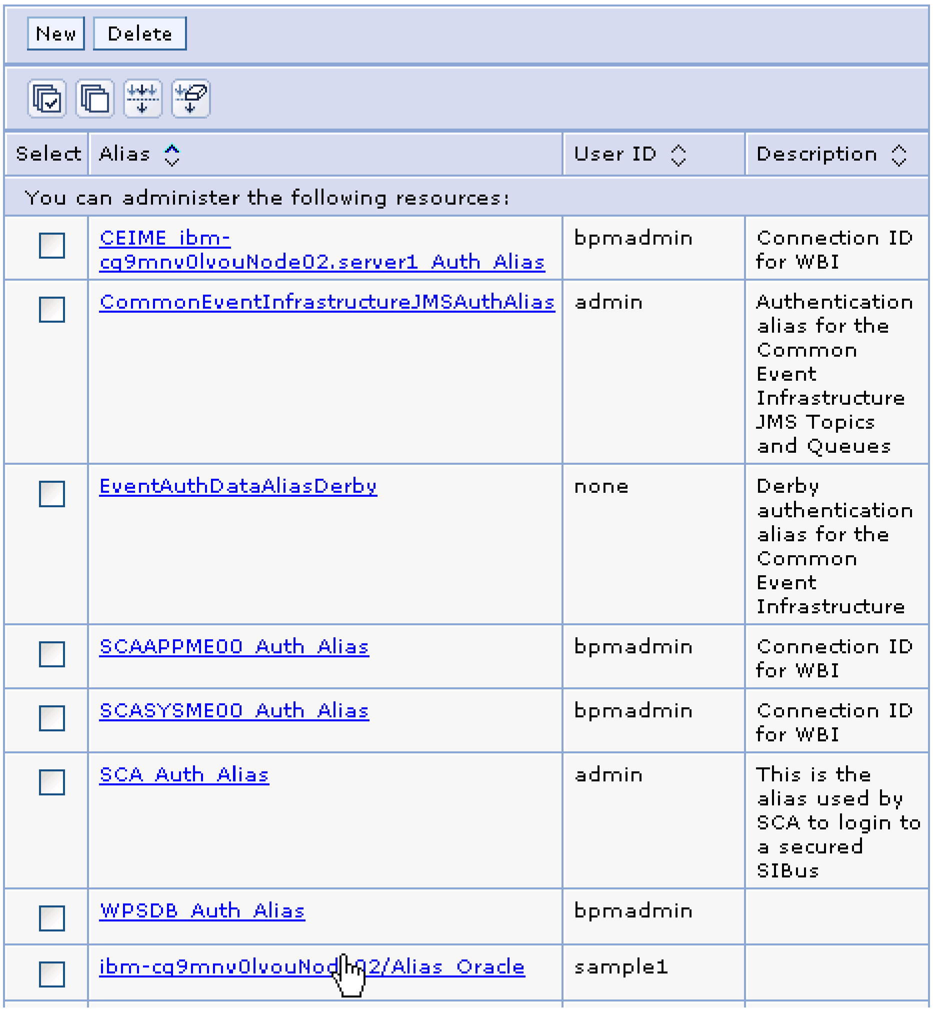
⚠ The server may need to be restarted for these changes to take effect.

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)

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



New | Delete

Select	Alias	User ID	Description
You can administer the following resources:			
□	CEIME_ibm-cq9mnv0lvouNode02.server1_Auth_Alias	bpmadmin	Connection ID for WBI
□	CommonEventInfrastructureJMSAuthAlias	admin	Authentication alias for the Common Event Infrastructure JMS Topics and Queues
□	EventAuthDataAliasDerby	none	Derby authentication alias for the Common Event Infrastructure
□	SCAAPPME00_Auth_Alias	bpmadmin	Connection ID for WBI
□	SCASYSME00_Auth_Alias	bpmadmin	Connection ID for WBI
□	SCA_Auth_Alias	admin	This is the alias used by SCA to login to a secured SIBus
□	WPSDB_Auth_Alias	bpmadmin	
□	ibm-cq9mnv0lvouNode02/Alias_Oracle	sample1	

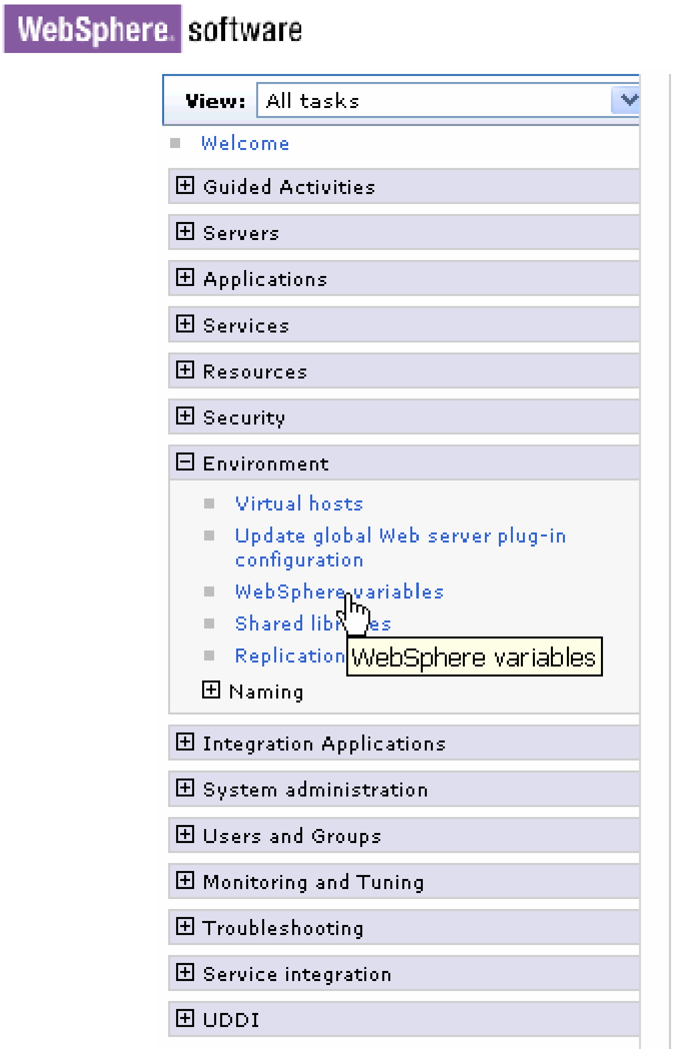
Create a data source

Create a data source in IBM 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.

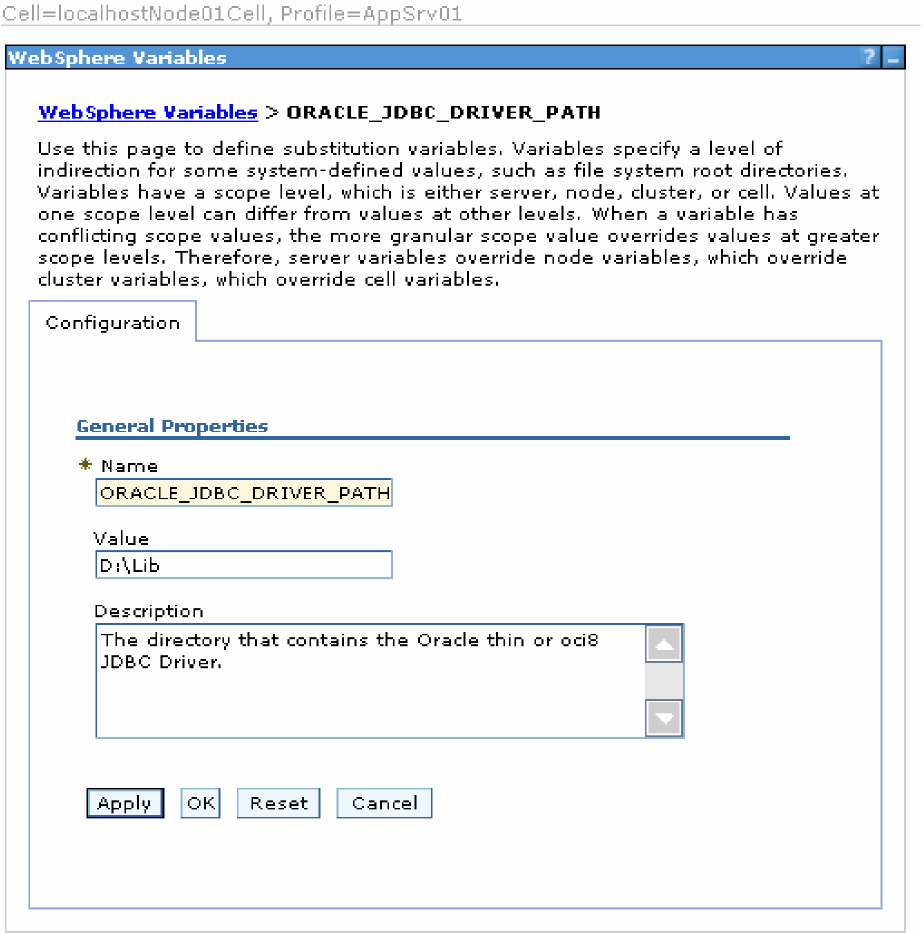
Note: This tutorial uses Oracle as the database and the Oracle thin driver, ojdbc6.jar.

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

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



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**.



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

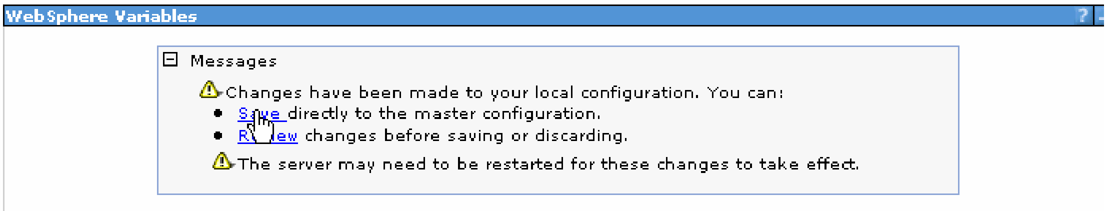
General Properties

* Name
ORACLE_JDBC_DRIVER_PATH

Value
D:\Lib

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

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



WebSphere Variables

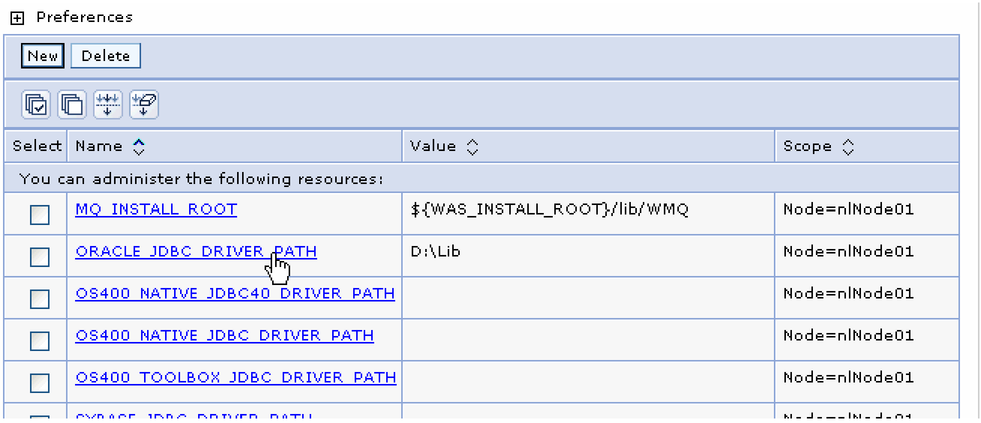
Messages

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

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

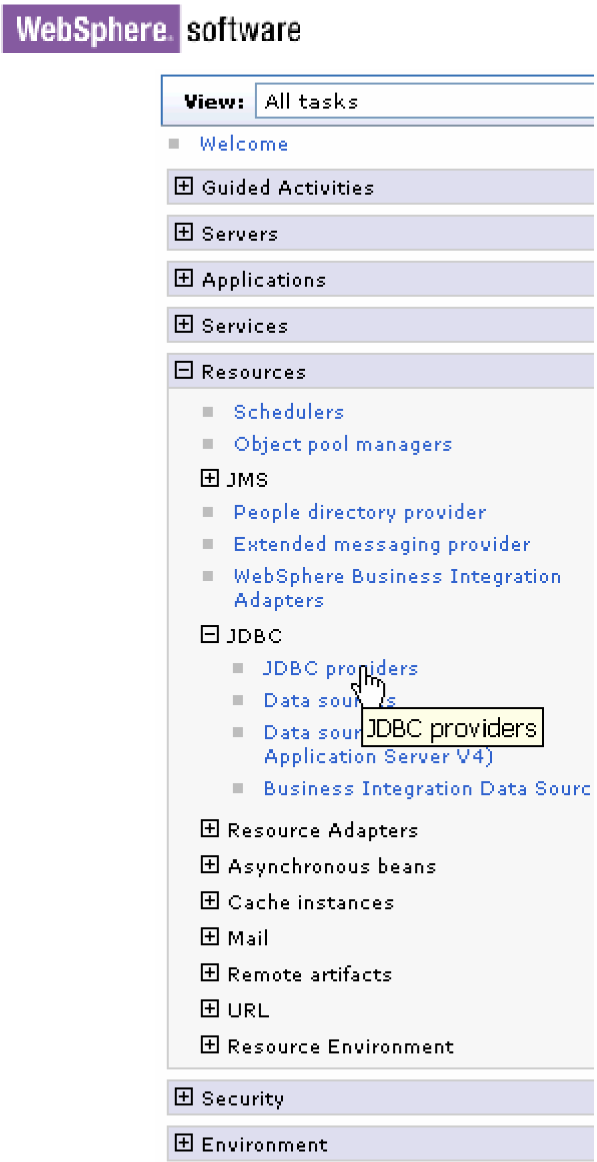
⚠ The server may need to be restarted for these changes to take effect.

The variable has been added and appears in the list.

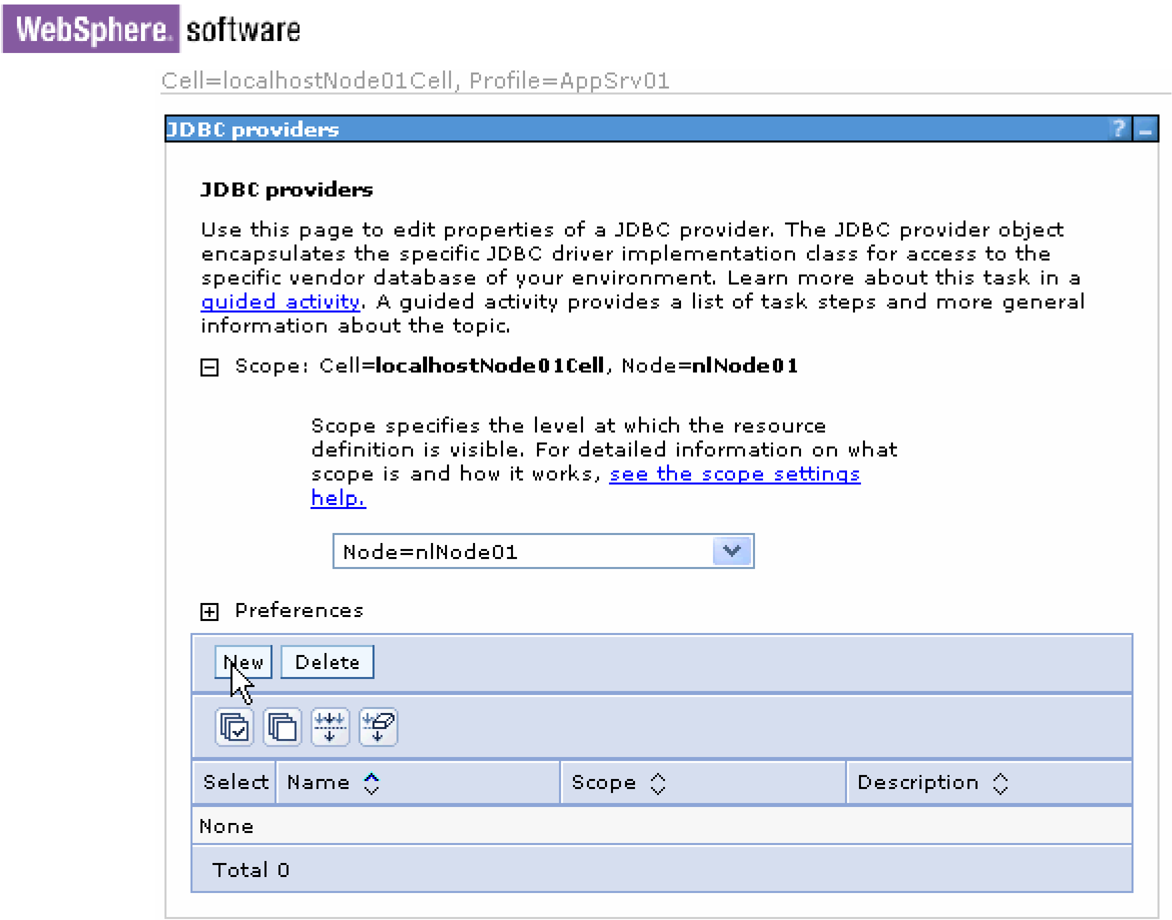


Select	Name	Value	Scope
You can administer the following resources:			
□	MQ_INSTALL_ROOT	${WAS_INSTALL_ROOT}/lib/WMQ	Node=nlNode01
□	ORACLE_JDBC_DRIVER_PATH	D:\Lib	Node=nlNode01
□	OS400_NATIVE_JDBC40_DRIVER_PATH		Node=nlNode01
□	OS400_NATIVE_JDBC_DRIVER_PATH		Node=nlNode01
□	OS400_TOOLBOX_JDBC_DRIVER_PATH		Node=nlNode01
□	SYBASE_JDBC_DRIVER_PATH		Node=nlNode01

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



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



Cell=localhostNode01Cell, Profile=AppSrv01

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=**nlNode01**

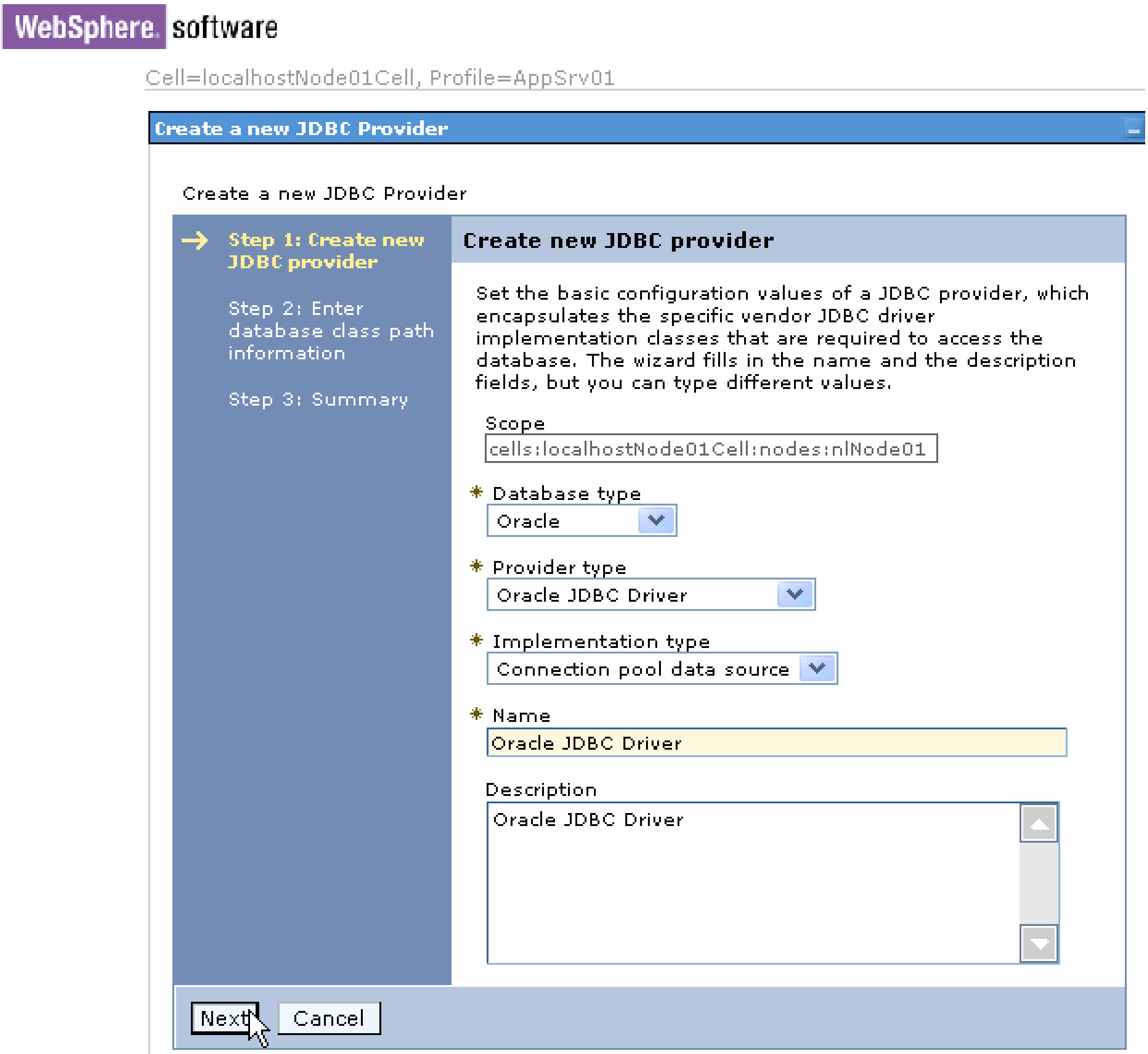
Scope specifies the level at which the resource definition is visible. For detailed information on what scope is and how it works, see the scope settings help.

Node=nlNode01

⊞ 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**.



Cell=localhostNode01Cell, Profile=AppSrv01

Create a new JDBC Provider

Create a new JDBC Provider

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

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:nlNode01

* Database type: Oracle

* Provider type: Oracle JDBC Driver

* Implementation type: Connection pool data source

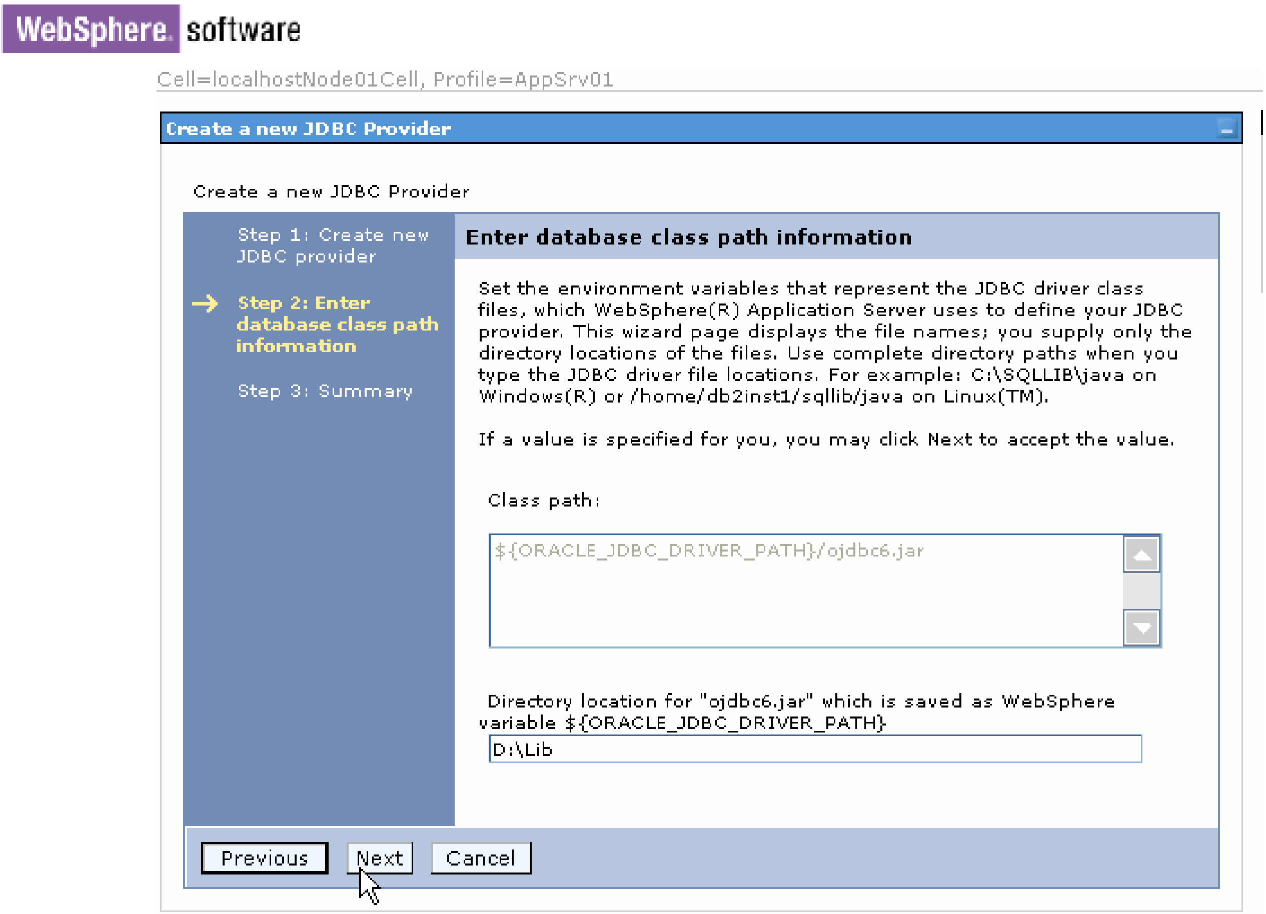
* Name: Oracle JDBC Driver

Description: Oracle JDBC Driver

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**.



Cell=localhostNode01Cell, Profile=AppSrv01

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: 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.

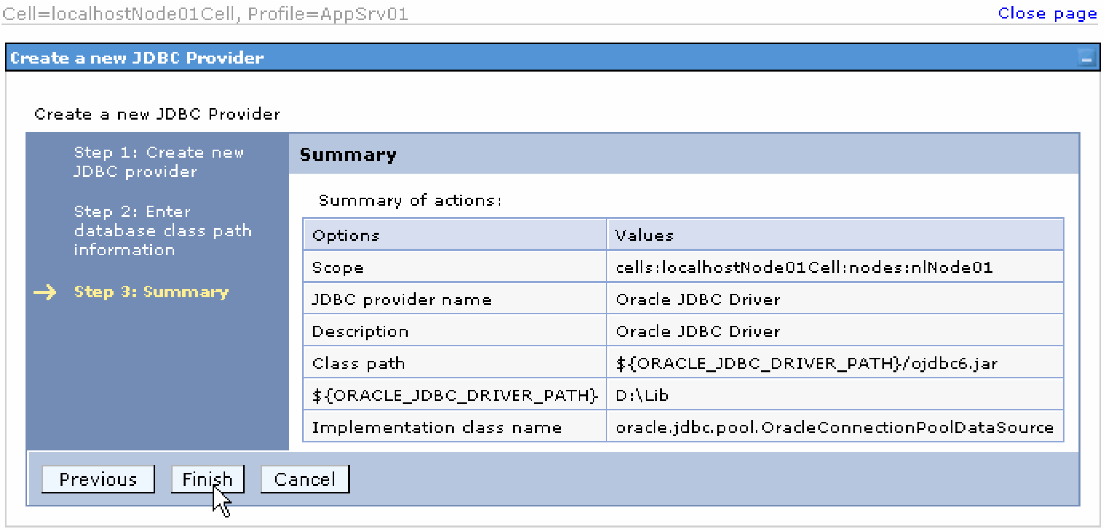
Class path:

${ORACLE_JDBC_DRIVER_PATH}/ojdbc6.jar

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

D:\Lib

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



Cell=localhostNode01Cell, Profile=AppSrv01

Create a new JDBC Provider

Create a new JDBC Provider

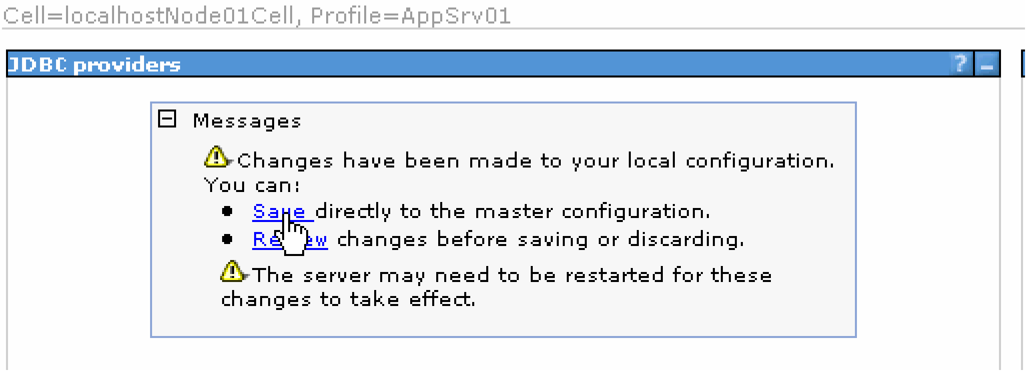
Step 1: Create new JDBC provider
Step 2: Enter database class path information
→ Step 3: Summary

Summary

Summary of actions:

Options	Values
Scope	cells:localhostNode01Cell:nodes:nlNode01
JDBC provider name	Oracle JDBC Driver
Description	Oracle JDBC Driver
Class path	${ORACLE_JDBC_DRIVER_PATH}/ojdbc6.jar
${ORACLE_JDBC_DRIVER_PATH}	D:\Lib
Implementation class name	oracle.jdbc.pool.OracleConnectionPoolDataSource

10. Click **Save**.



Cell=localhostNode01Cell, Profile=AppSrv01

JDBC providers

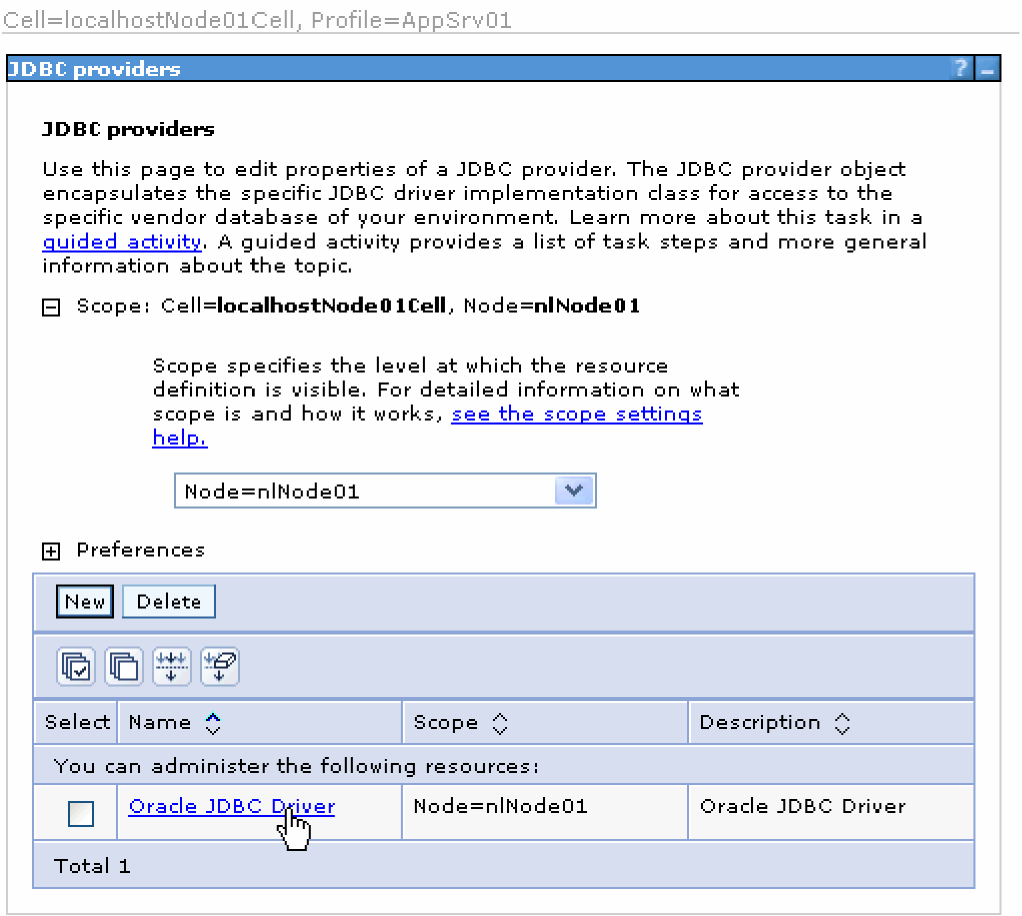
⊟ Messages

⚠ 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

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=**nlNode01**

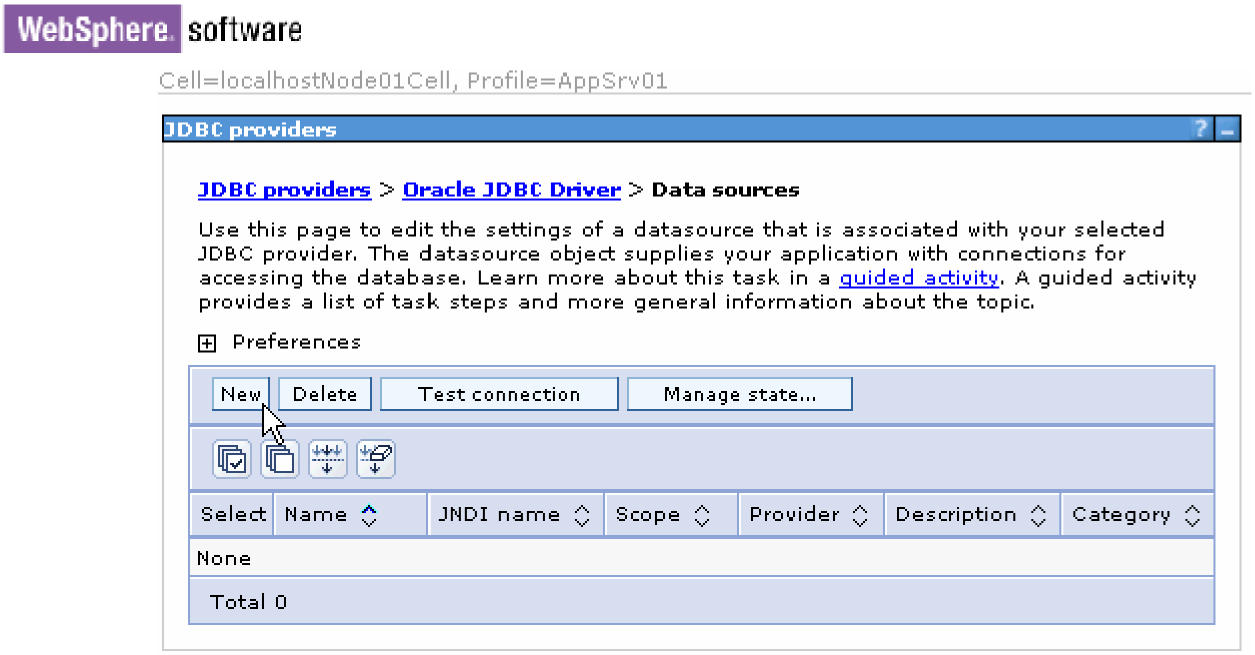
Scope specifies the level at which the resource definition is visible. For detailed information on what scope is and how it works, see the scope settings help.

Node=nlNode01

⊞ Preferences

Select	Name ⇕	Scope ⇕	Description ⇕
You can administer the following resources:			
□	Oracle JDBC Driver	Node=nlNode01	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

JDBC providers > Oracle JDBC Driver > **Data sources**

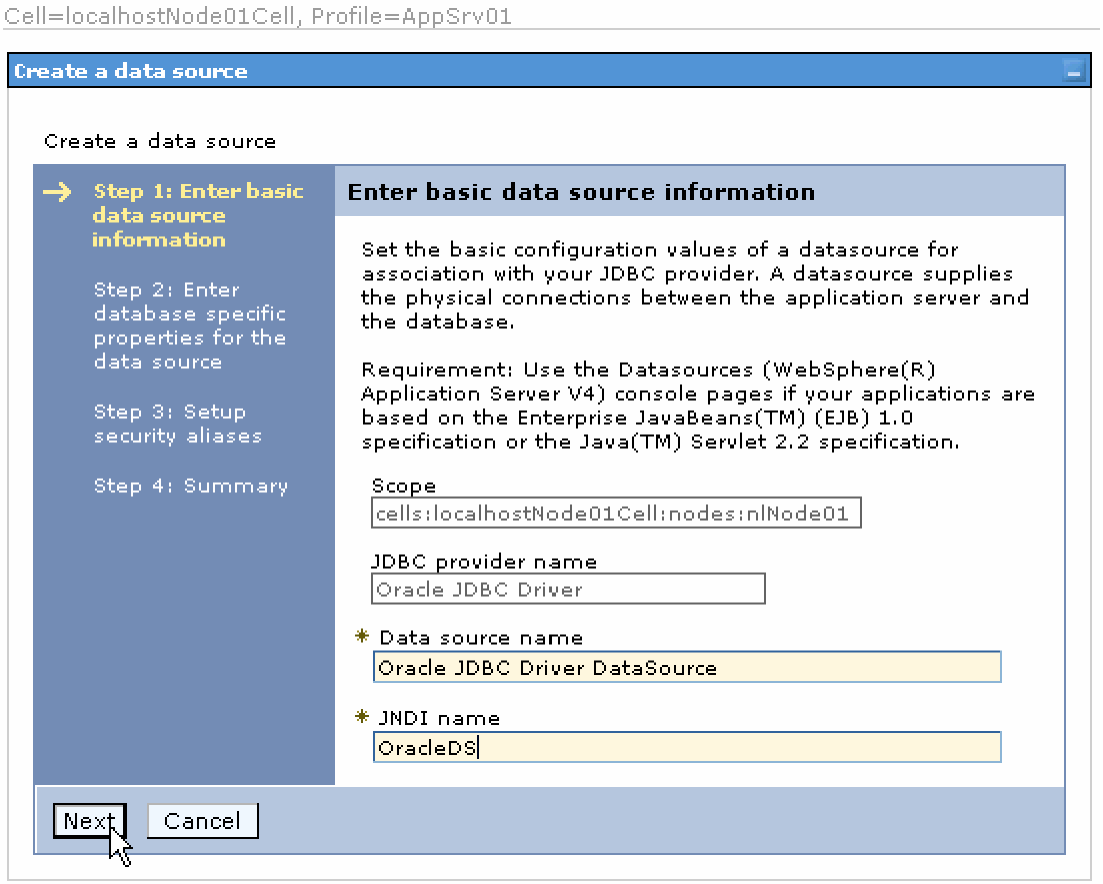
Use this page to edit the settings of a datasource that is associated with your selected JDBC provider. The datasource object supplies your application with connections for accessing the database. Learn more about this task in a guided activity. A guided activity provides a list of task steps and more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

⊞ Preferences

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.

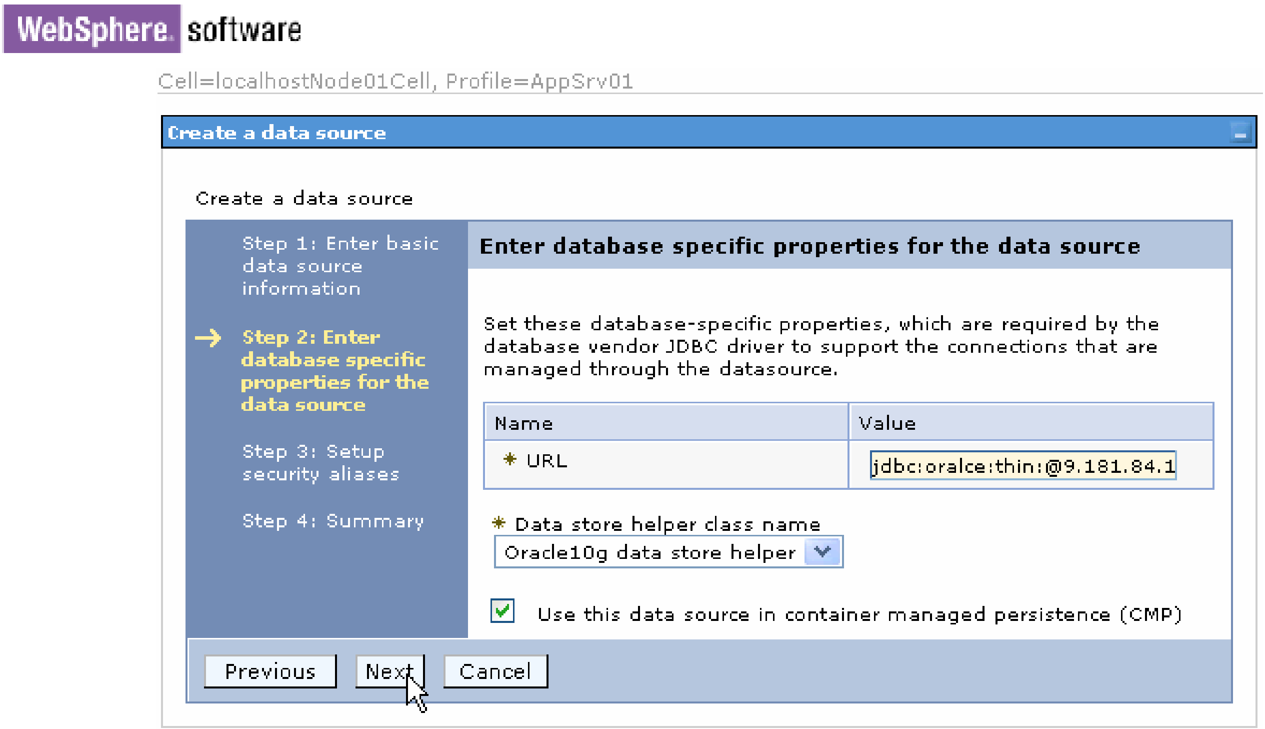
Scope: cells:localhostNode01Cell:nodes:nlNode01

JDBC provider name: Oracle JDBC Driver

* Data source name: Oracle JDBC Driver DataSource

* JNDI name: OracleDS

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

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

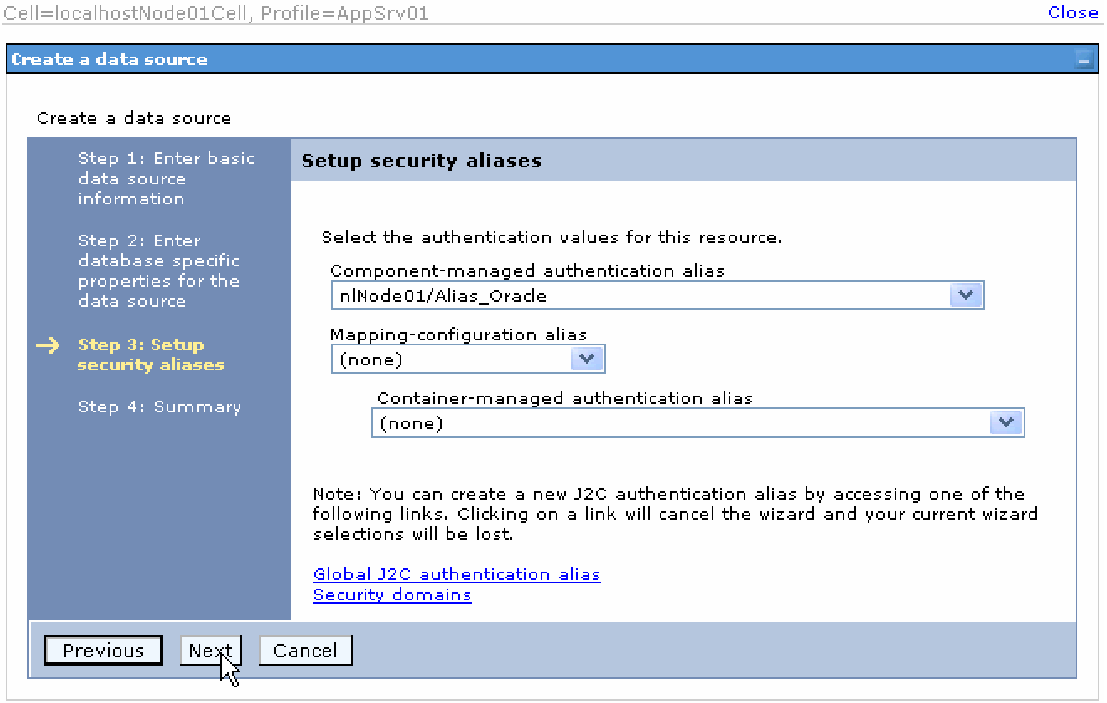
Set these database-specific properties, which are required by the database vendor JDBC driver to support the connections that are managed through the datasource.

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

* Data store helper class name
Oracle10g data store helper

☑ Use this data source in container managed persistence (CMP)

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

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

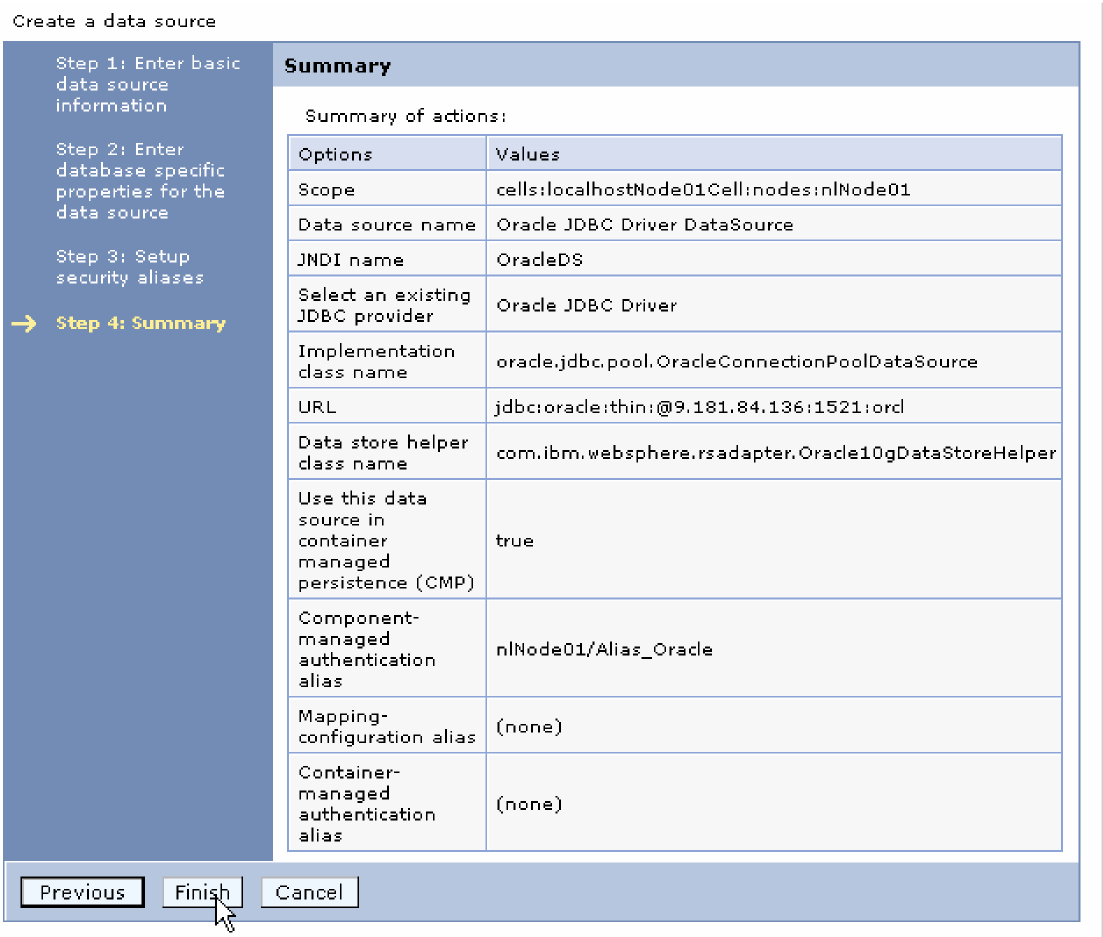
Mapping-configuration alias
(none)

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

15. 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

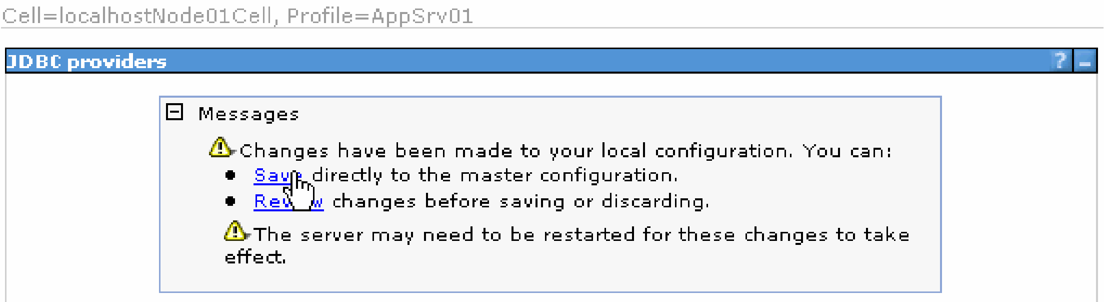
→ Step 4: Summary

Summary

Summary of actions:

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:orcl
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/Alias_Oracle
Mapping-configuration alias	(none)
Container-managed authentication alias	(none)

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



Cell=localhostNode01Cell, Profile=AppSrv01

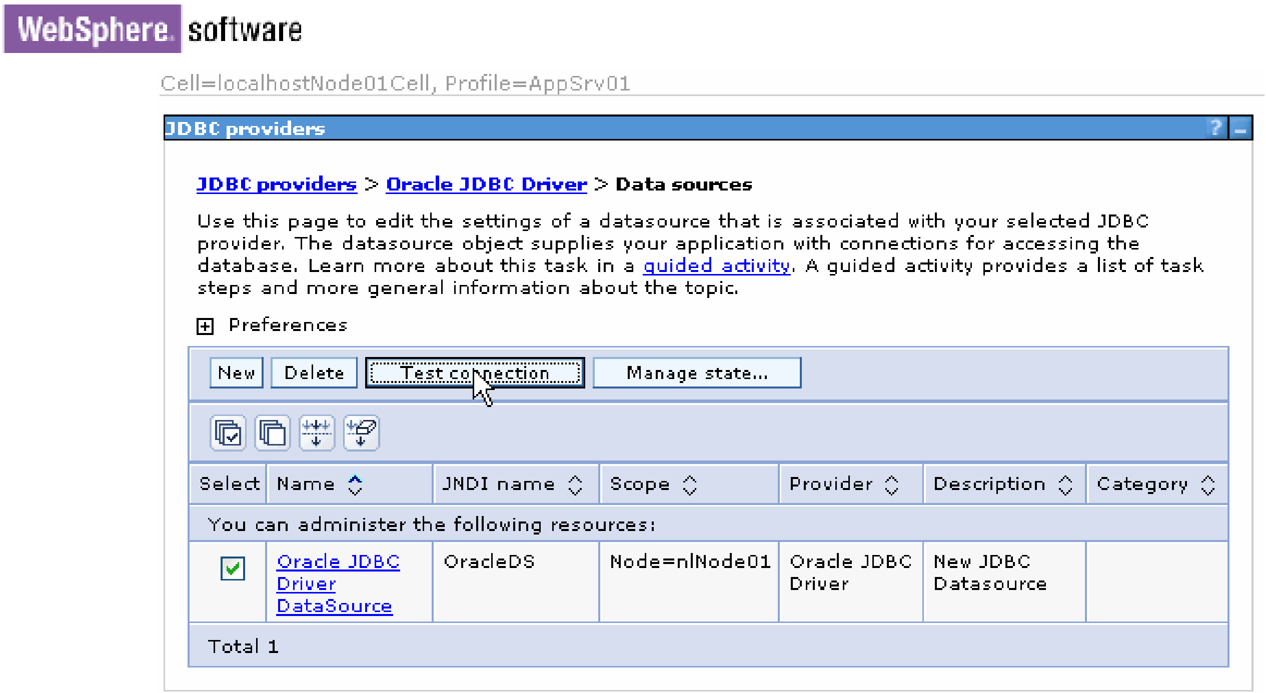
⊟ Messages

⚠ 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.

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



WebSphere. software

Cell=localhostNode01Cell, Profile=AppSrv01

JDBC providers

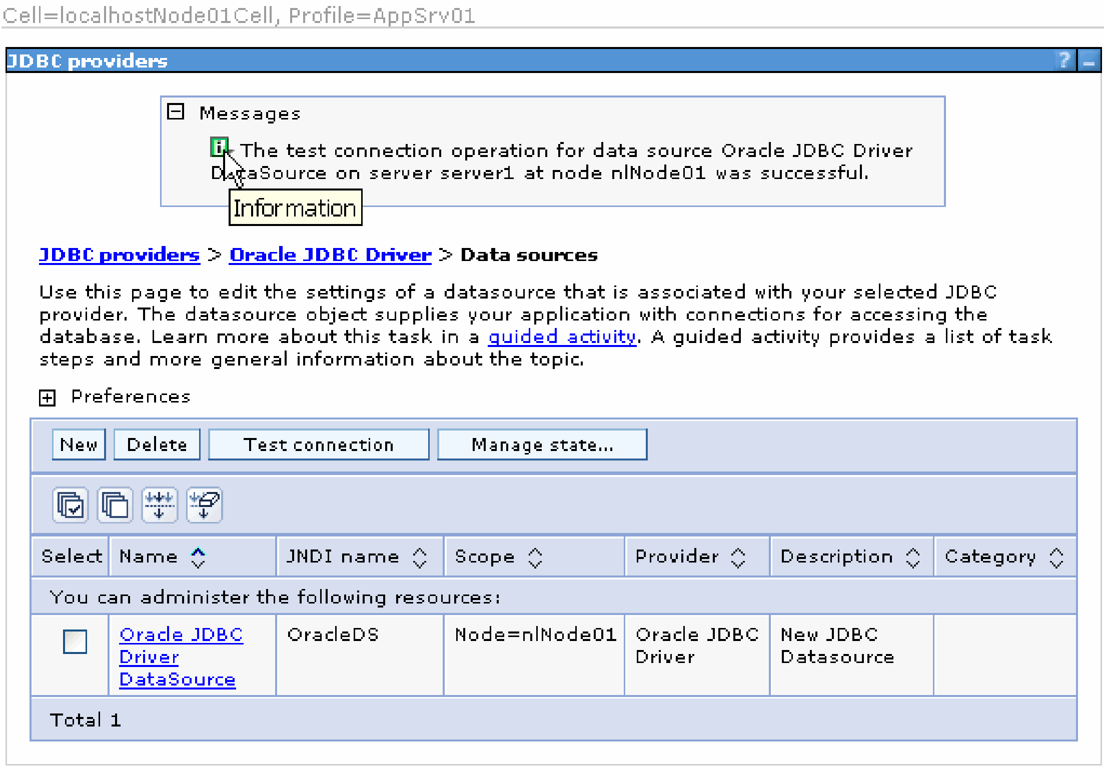
JDBC providers > Oracle JDBC Driver > Data sources

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

Select	Name	JNDI name	Scope	Provider	Description	Category
You can administer the following resources:						
☑	Oracle JDBC Driver DataSource	OracleDS	Node=nlNode01	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

⊟ Messages

The test connection operation for data source Oracle JDBC Driver DataSource on server server1 at node nlNode01 was successful.

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.

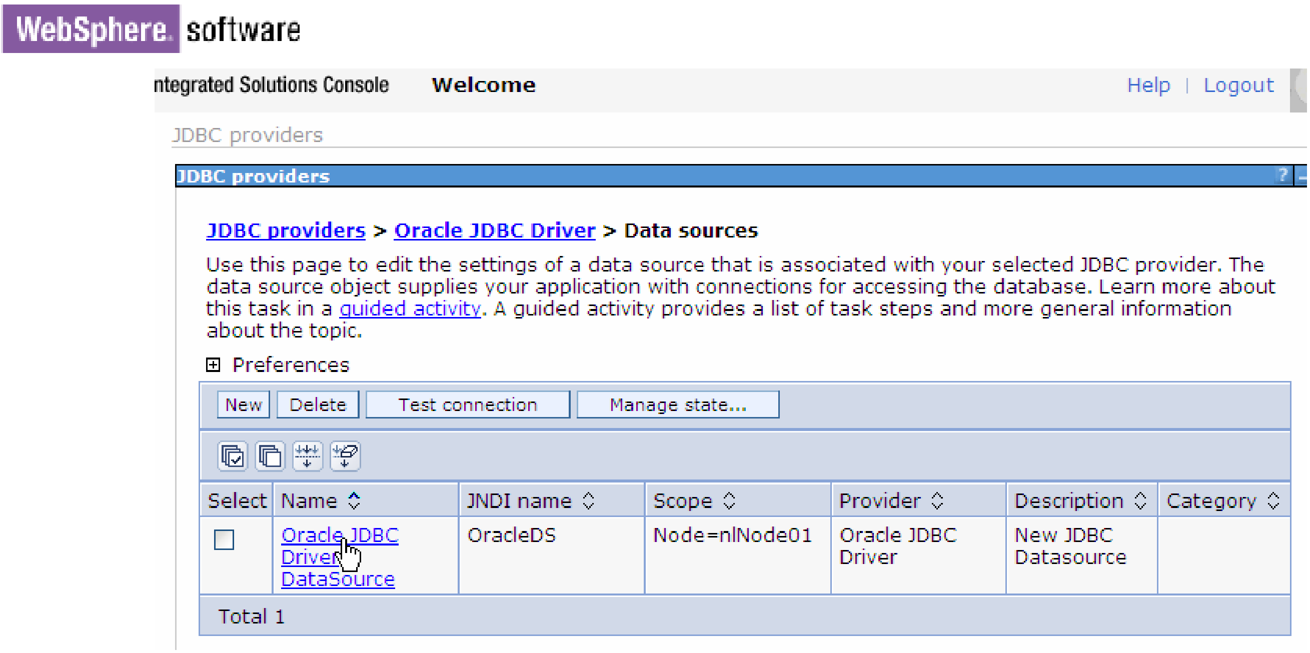
⊞ Preferences

Select	Name	JNDI name	Scope	Provider	Description	Category
You can administer the following resources:						
□	Oracle JDBC Driver DataSource	OracleDS	Node=nlNode01	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.



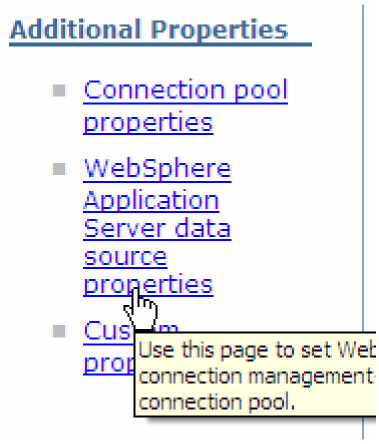
JDBC providers > Oracle JDBC Driver > Data sources

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.

Select	Name	JNDI name	Scope	Provider	Description	Category
□	Oracle JDBC Driver DataSource	OracleDS	Node=nlNode01	Oracle JDBC Driver	New JDBC Datasource	

Total 1

2. On the right, under **Additional Properties** click **WebSphere Application Server data source properties.**

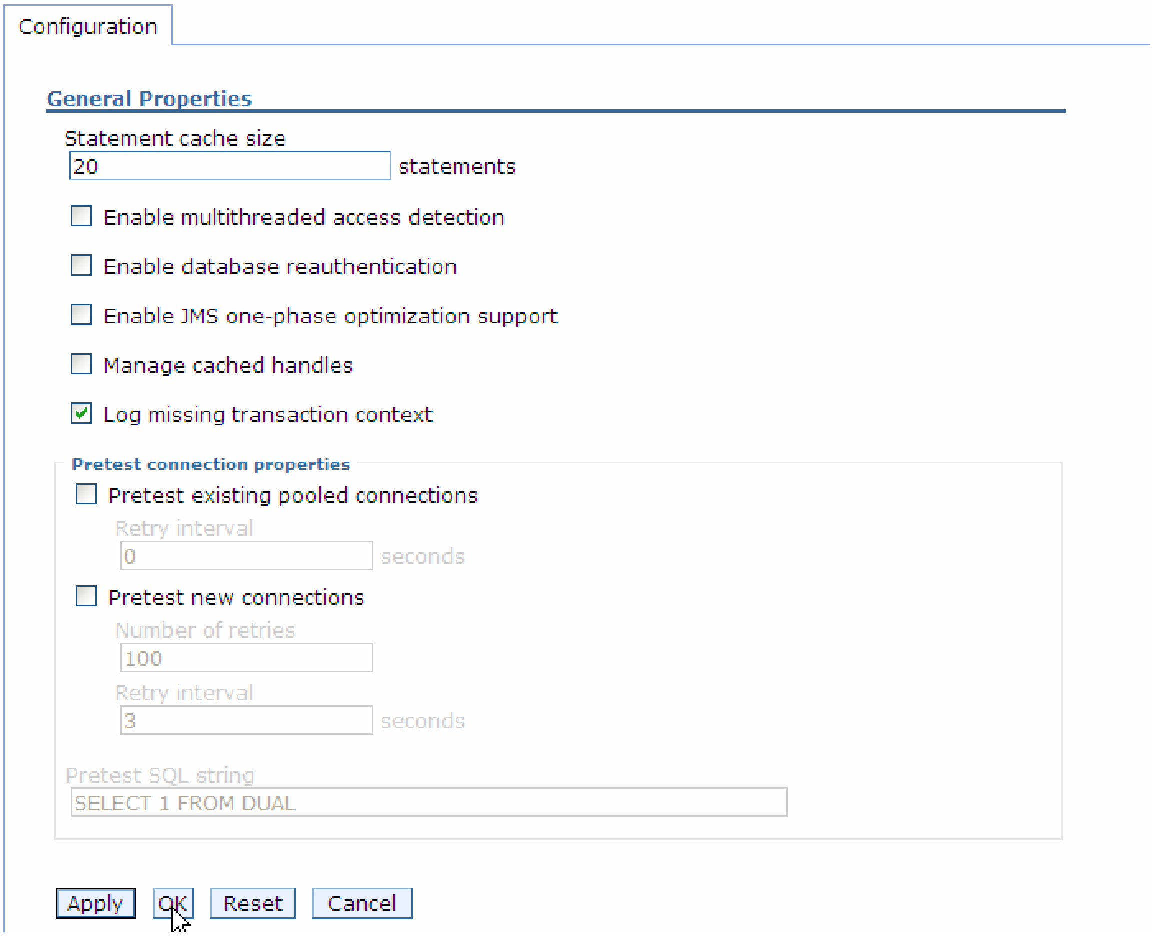


Additional Properties

- Connection pool properties
- WebSphere Application Server data source properties
- Cus m prop

Use this page to set Web connection management connection pool.

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

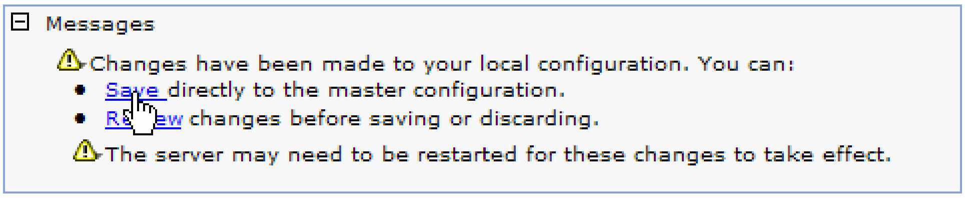
□ Pretest new connections

Number of retries
100

Retry interval
3 seconds

Pretest SQL string
SELECT 1 FROM DUAL

4. In the Messages area, click **Save**.



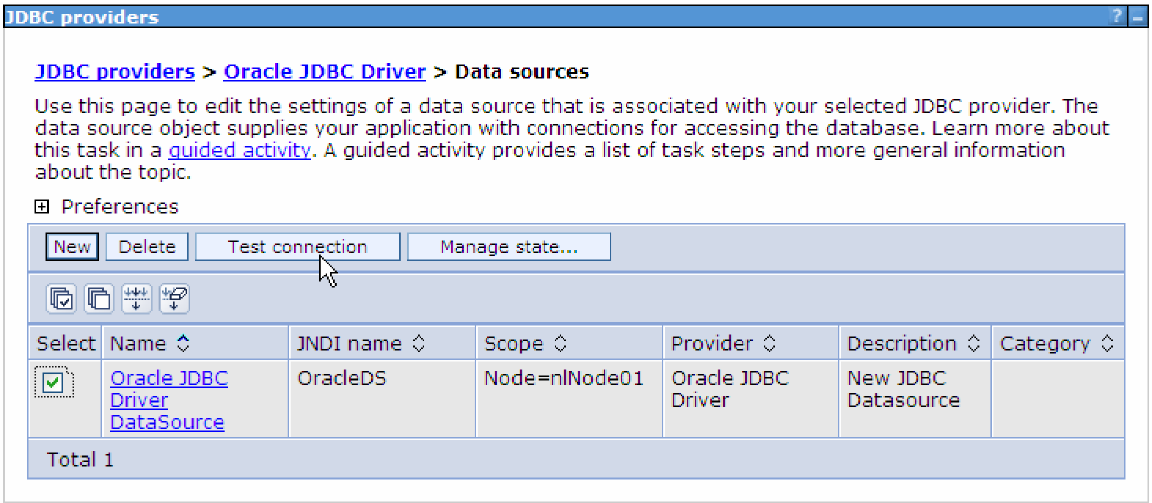
⊟ Messages

⚠ 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

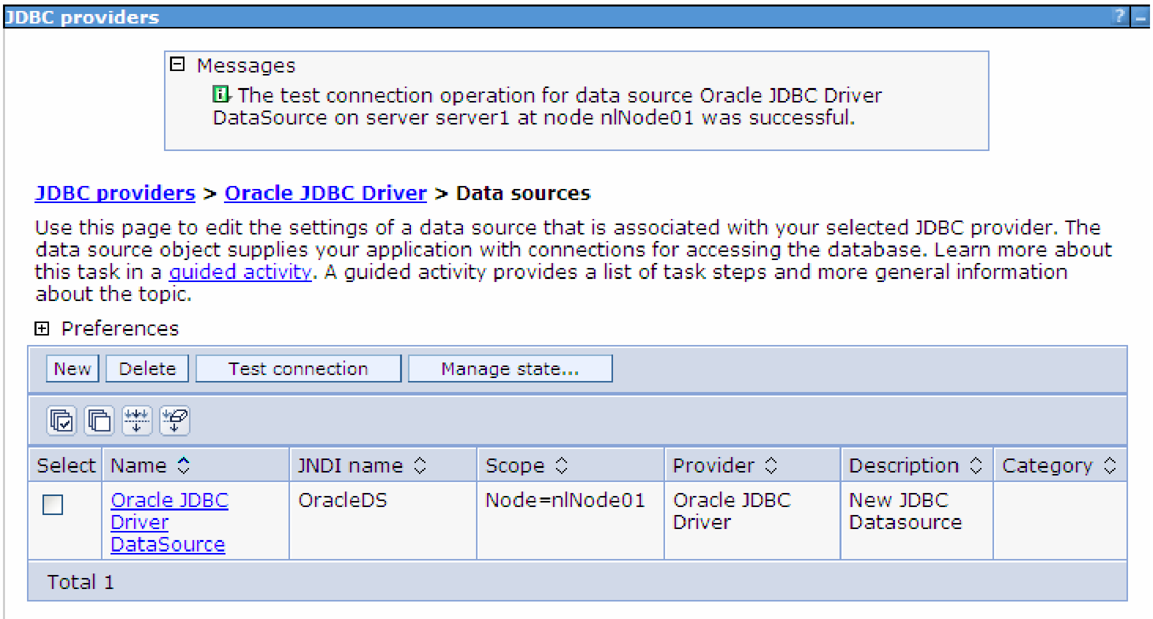
Use this page to edit the settings of a data source that is associated with your selected JDBC provider. The data source object supplies your application with connections for accessing the database. Learn more about this task in a guided activity. A guided activity provides a list of task steps and more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

⊞ Preferences

Select	Name	JNDI name	Scope	Provider	Description	Category
☑	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

⊟ Messages

The test connection operation for data source Oracle JDBC Driver DataSource on server server1 at node nlNode01 was successful.

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

Select	Name	JNDI name	Scope	Provider	Description	Category
☐	Oracle JDBC Driver DataSource	OracleDS	Node=nlNode01	Oracle JDBC Driver	New JDBC Datasource	

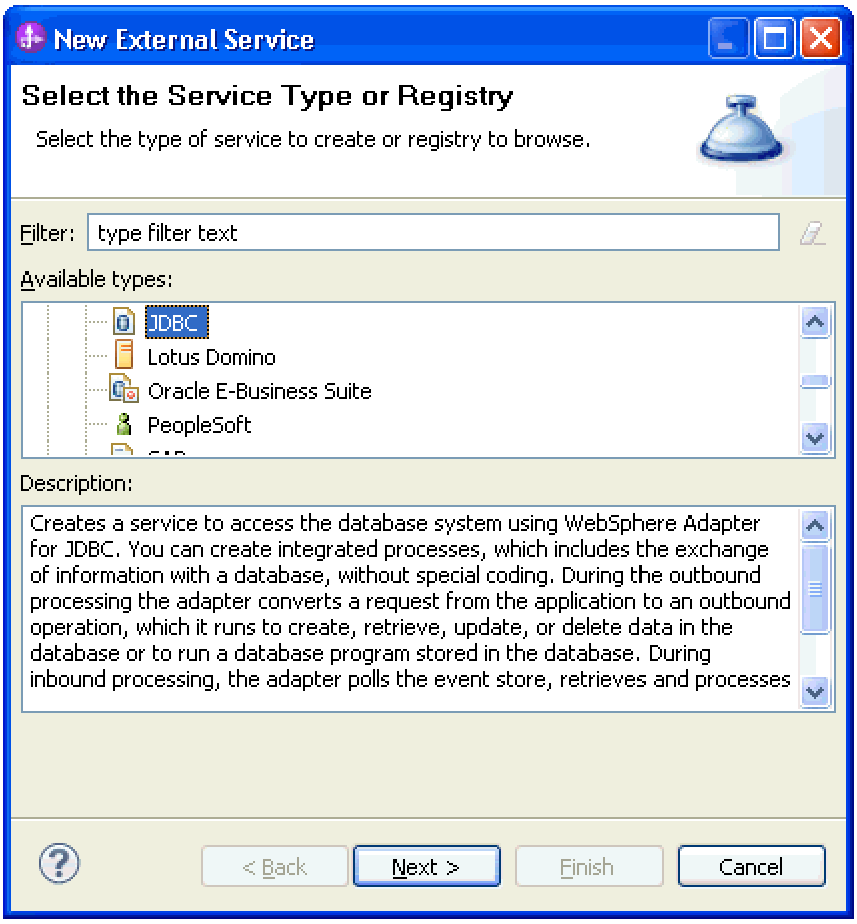
Total 1

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 IBM Integration Designer 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**.



Select the Service Type or Registry

Select the type of service to create or registry to browse.

Filter: type filter text

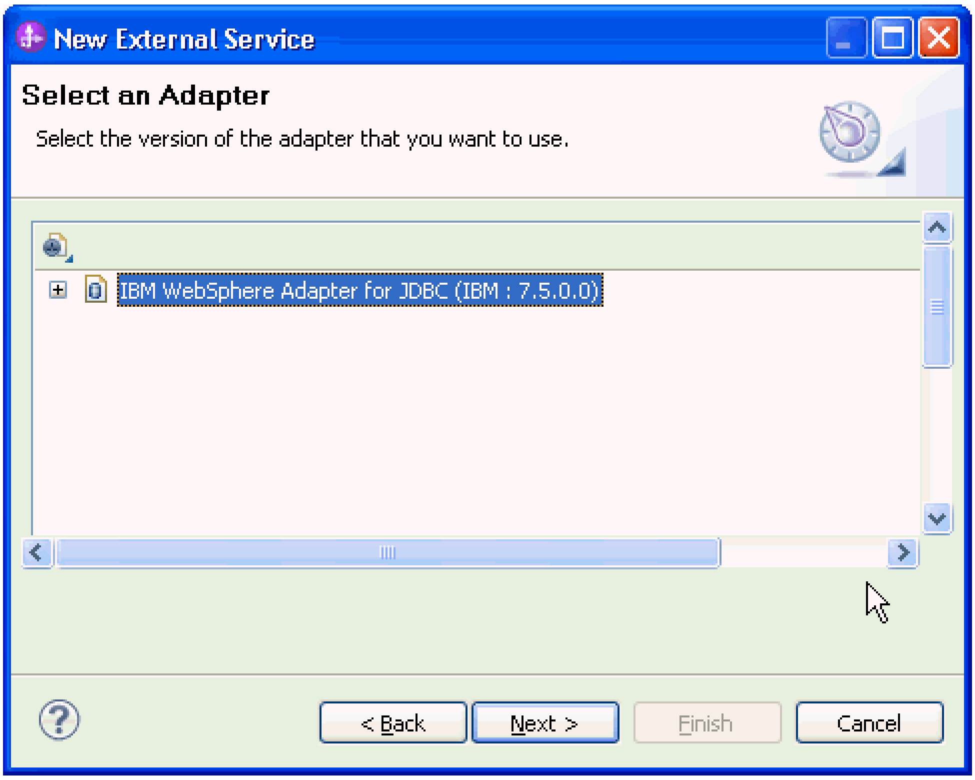
Available types:

- JDBC
- Lotus Domino
- Oracle E-Business Suite
- PeopleSoft

Description:

Creates a service to access the database system using WebSphere Adapter for JDBC. You can create integrated processes, which includes the exchange of information with a database, without special coding. During the outbound processing the adapter converts a request from the application to an outbound operation, which it runs to create, retrieve, update, or delete data in the database or to run a database program stored in the database. During inbound processing, the adapter polls the event store, retrieves and processes

4. Select **IBM WebSphere Adapter for JDBC** and click **Next.**



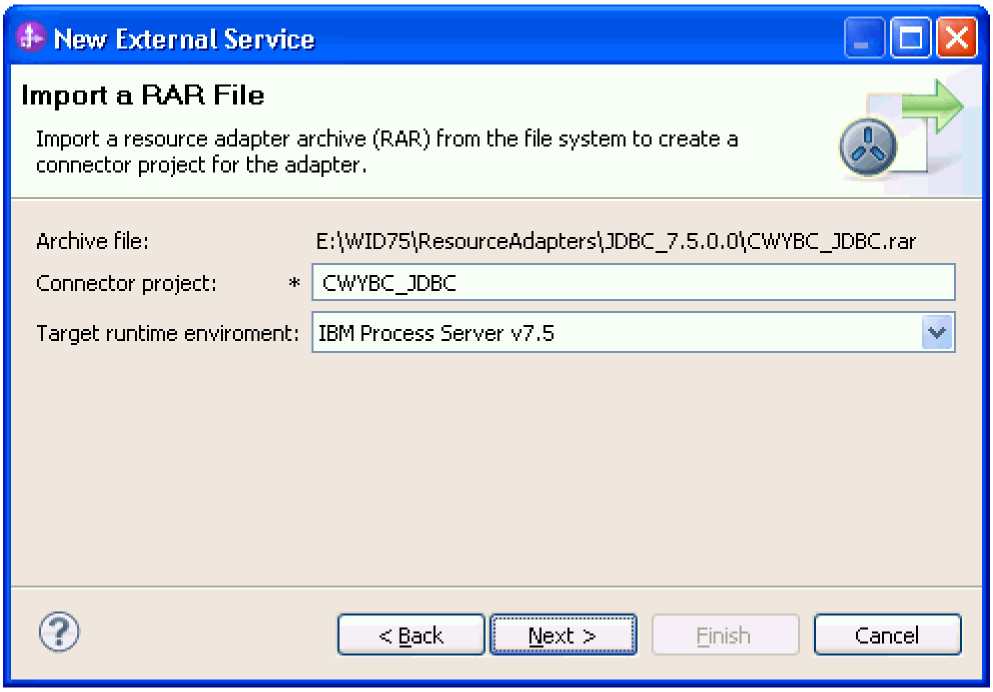
Select an Adapter

Select the version of the adapter that you want to use.

IBM WebSphere Adapter for JDBC (IBM : 7.5.0.0)

5. In the **Connector project** field, enter **CWYBC_JDBC.**

6. In the **Target runtime environment** field, select appropriate runtime and click **Next**.



New External Service

Import a RAR File

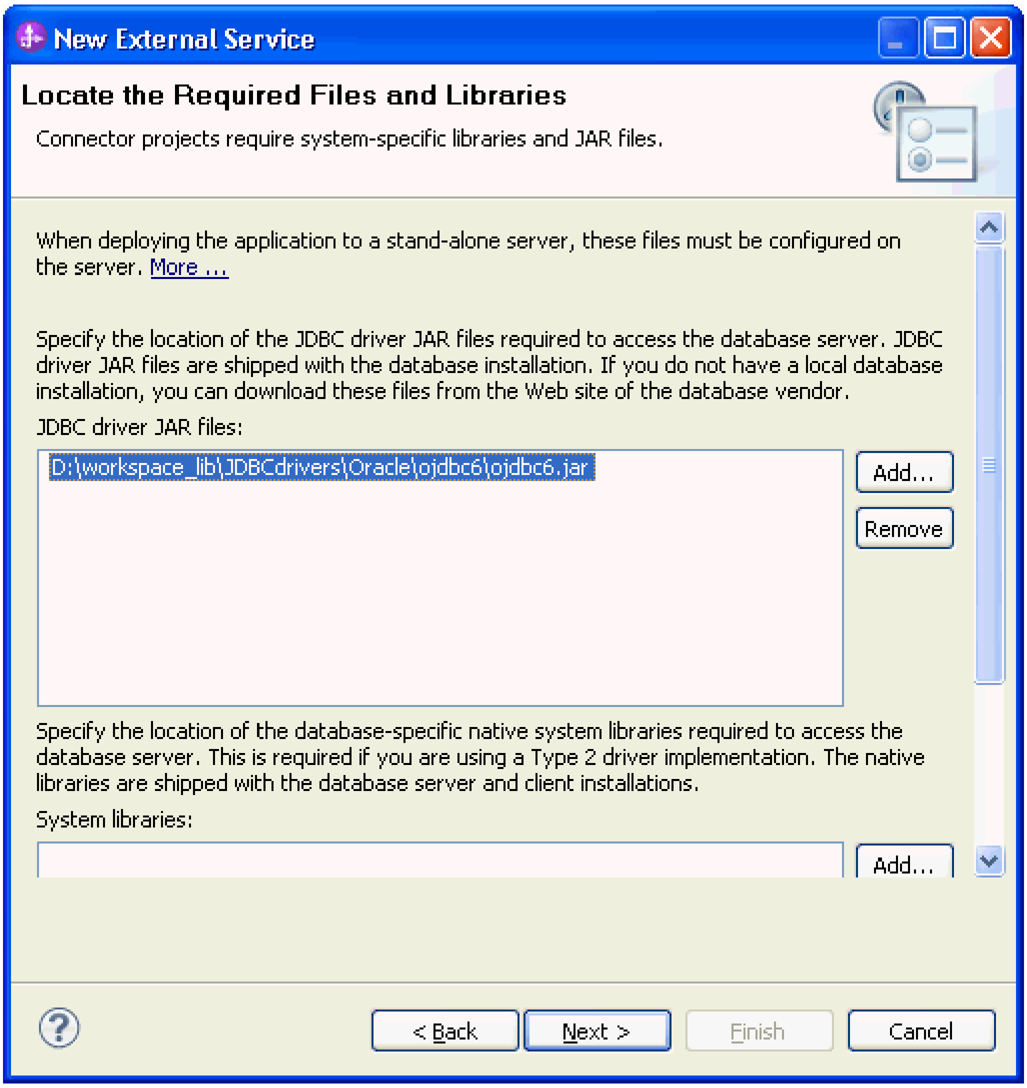
Import a resource adapter archive (RAR) from the file system to create a connector project for the adapter.

Archive file: E:\WID75\ResourceAdapters\JDBC_7.5.0.0\CWYBC_JDBC.rar

Connector project: * CWYBC_JDBC

Target runtime enviroment: IBM Process Server v7.5

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**.



New External Service

Locate the Required Files and Libraries

Connector projects require system-specific libraries and JAR files.

When deploying the application to a stand-alone server, these files must be configured on the server. More ...

Specify the location of the JDBC driver JAR files required to access the database server. JDBC driver JAR files are shipped with the database installation. If you do not have a local database installation, you can download these files from the Web site of the database vendor.

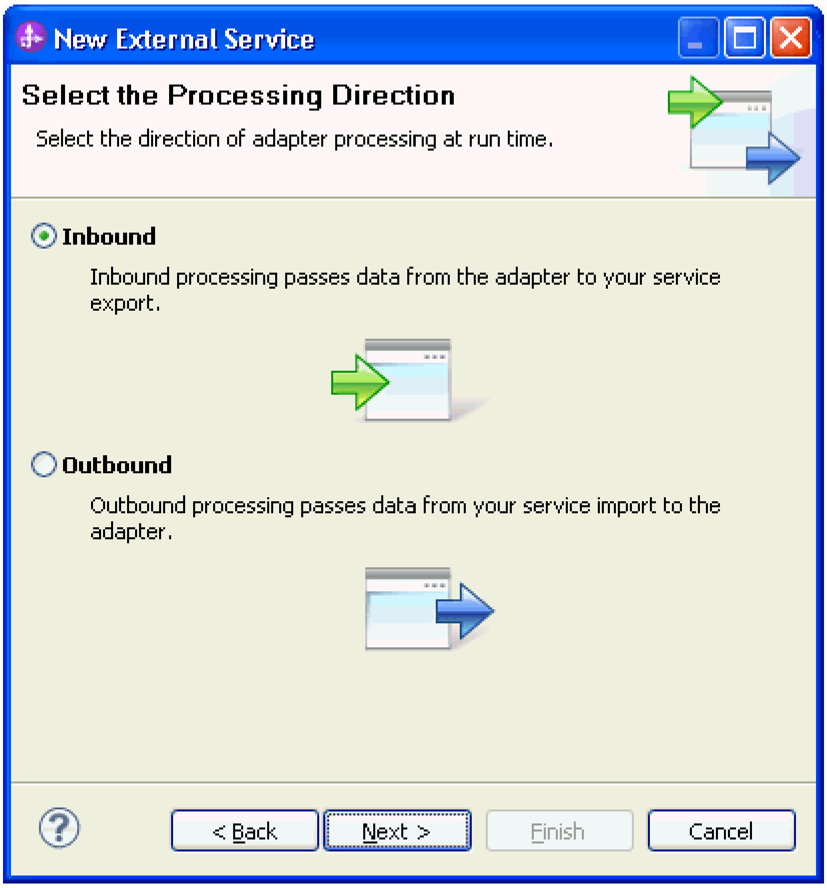
JDBC driver JAR files:

D:\workspace_lib\JDBCdrivers\Oracle\ojdbc6\ojdbc6.jar

Specify the location of the database-specific native system libraries required to access the database server. This is required if you are using a Type 2 driver implementation. The native libraries are shipped with the database server and client installations.

System libraries:

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



New External Service

Select the Processing Direction

Select the direction of adapter processing at run time.

● **Inbound**

Inbound processing passes data from the adapter to your service export.

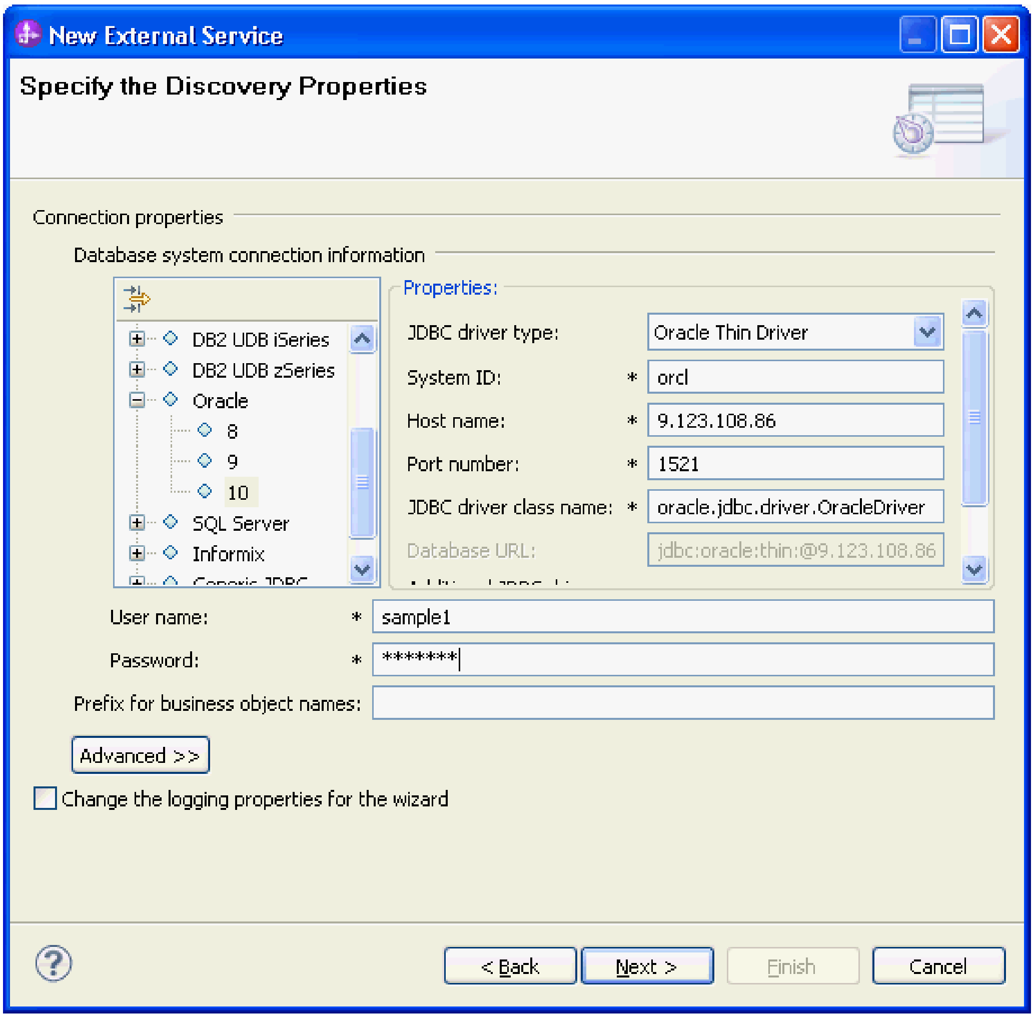
○ **Outbound**

Outbound processing passes data from your service import to the adapter.

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**.



Specify the Discovery Properties

Connection properties

Database system connection information

Properties:

JDBC driver type: Oracle Thin Driver

System ID: * orcl

Host name: * 9.123.108.86

Port number: * 1521

JDBC driver class name: * oracle.jdbc.driver.OracleDriver

Database URL: jdbc:oracle:thin:@9.123.108.86

User name: * sample1

Password: * *******

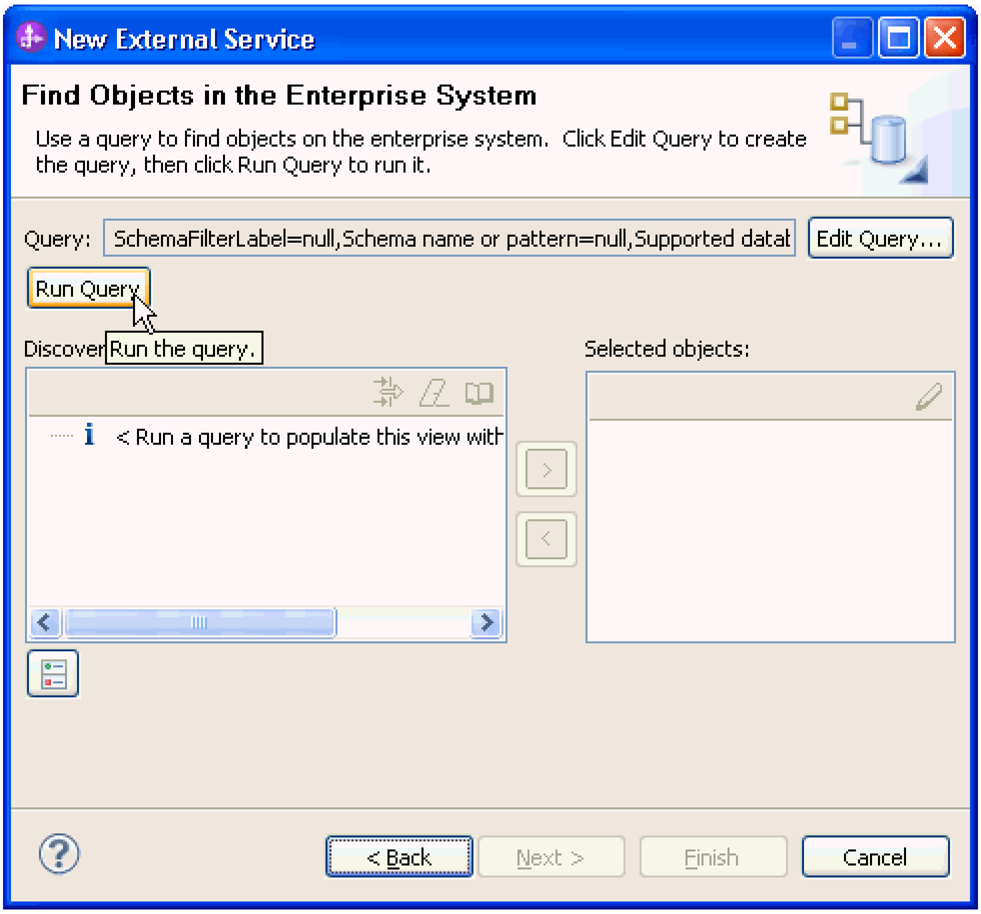
Prefix for business object names:

□ Change the logging properties for the wizard

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**.



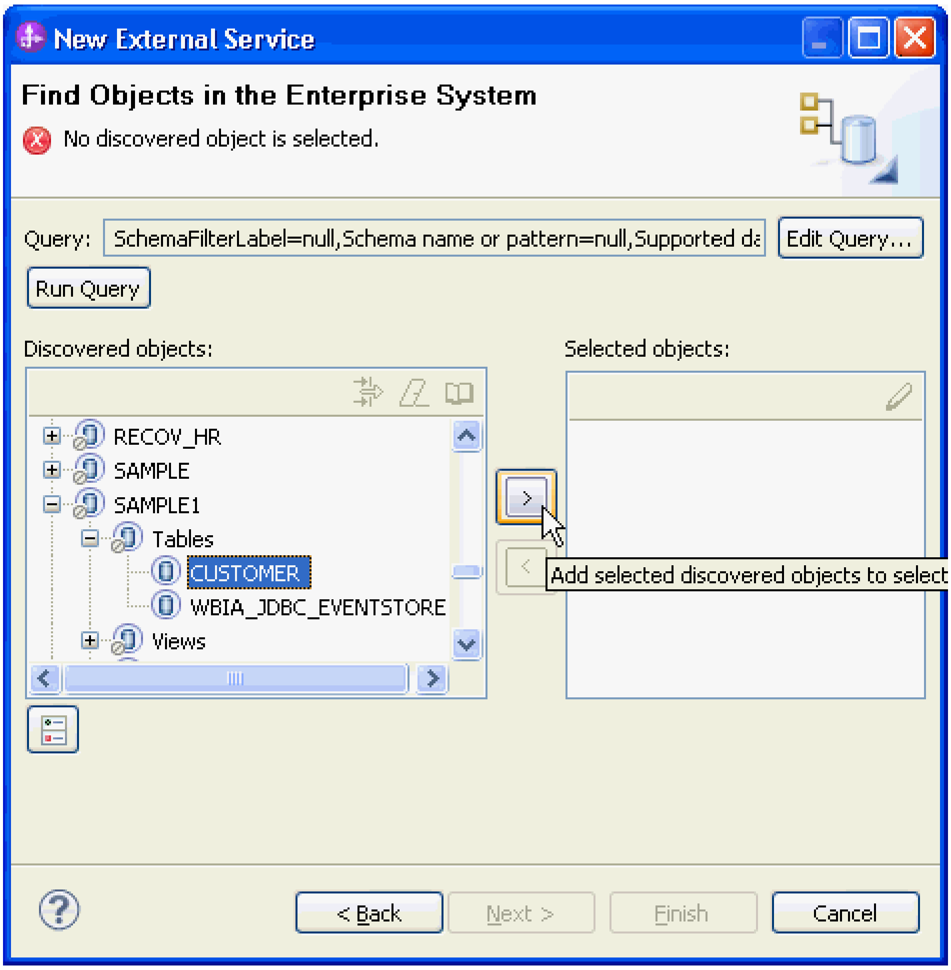
New External Service

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 datat

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



New External Service

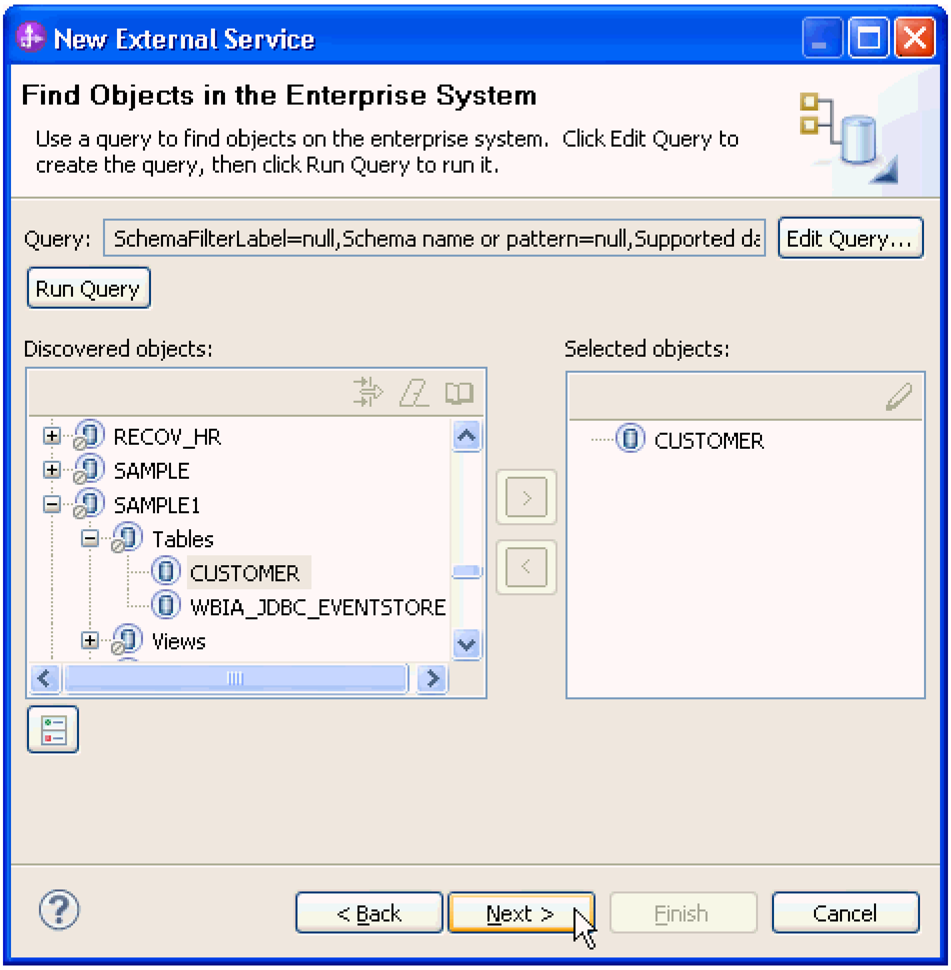
Find Objects in the Enterprise System

No discovered object is selected.

Query: SchemaFilterLabel=null,Schema name or pattern=null,Supported da

3. Select the **CUSTOMER** table and click .

4. Click **Next**.



New External Service

Find Objects in the Enterprise System

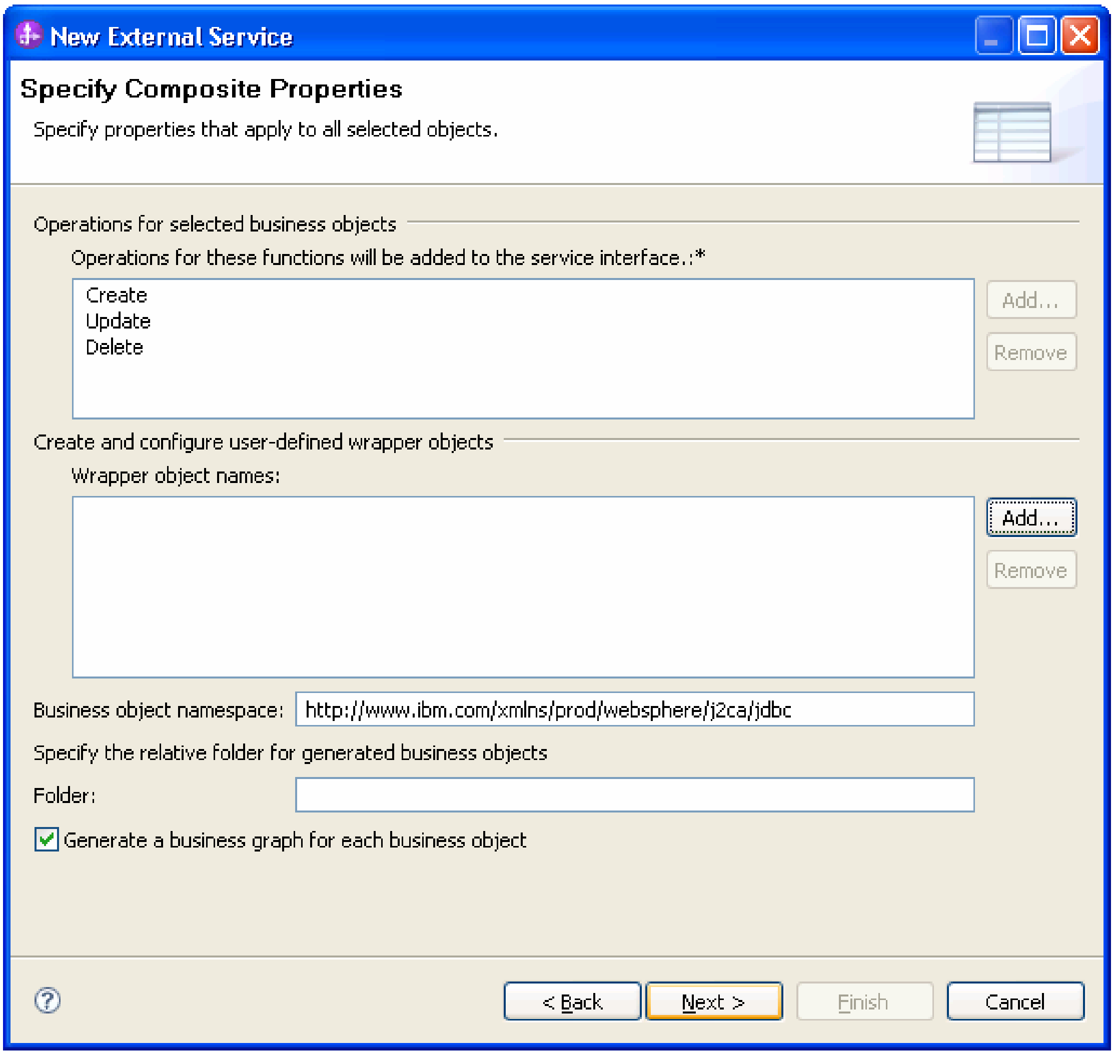
Use a query to find objects on the enterprise system. Click Edit Query to create the query, then click Run Query to run it.

Query: SchemaFilterLabel=null,Schema name or pattern=null,Supported da

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**.



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

Create and configure user-defined wrapper objects

Wrapper object names:

Business object namespace: http://www.ibm.com/xmlns/prod/websphere/j2ca/jdbc

Specify the relative folder for generated business objects

Folder:

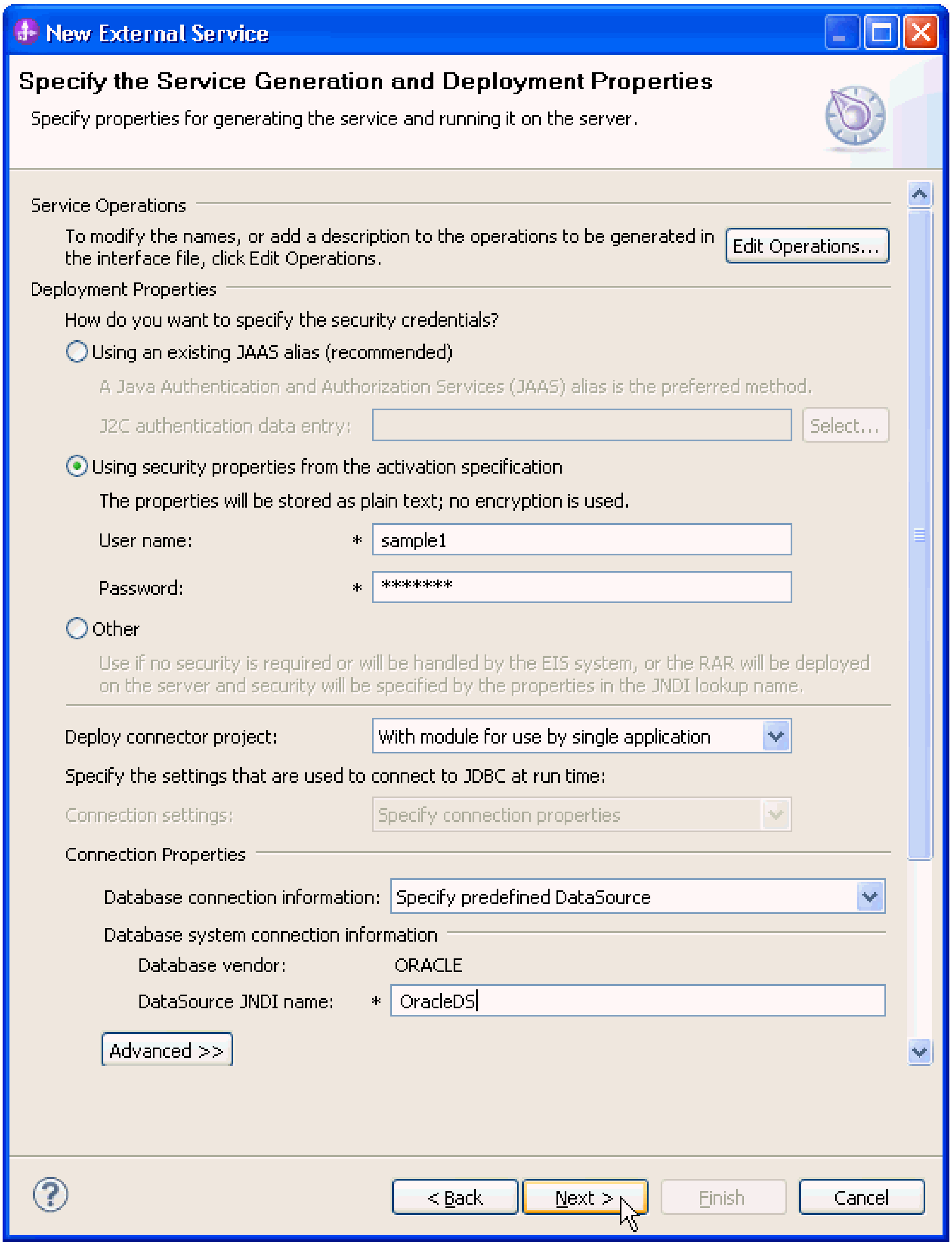
☑ Generate a business graph for each business object

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 and input the preconfigured JNDI name of oracle data source in **DataSource JNDI name**.

   c) Click **Advanced**.



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: * sample1

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: With module for use by single application

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

Connection settings: Specify connection properties

Connection Properties

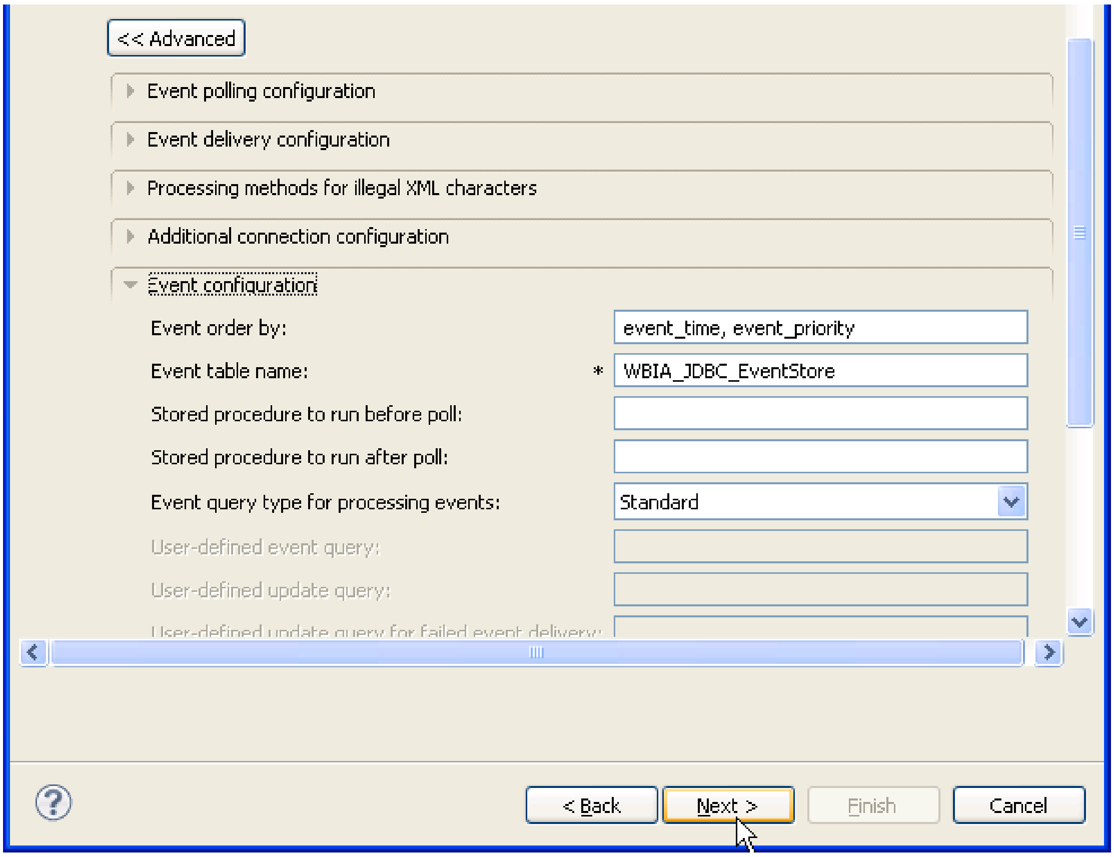
Database connection information: Specify predefined DataSource

Database system connection information

Database vendor: ORACLE

DataSource JNDI name: * OracleDS

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



<< Advanced

Event polling configuration

Event delivery configuration

Processing methods for illegal XML characters

Additional connection configuration

Event configuration

Event order by: event_time, event_priority

Event table name: * WBIA_JDBC_EventStore

Stored procedure to run before poll:

Stored procedure to run after poll:

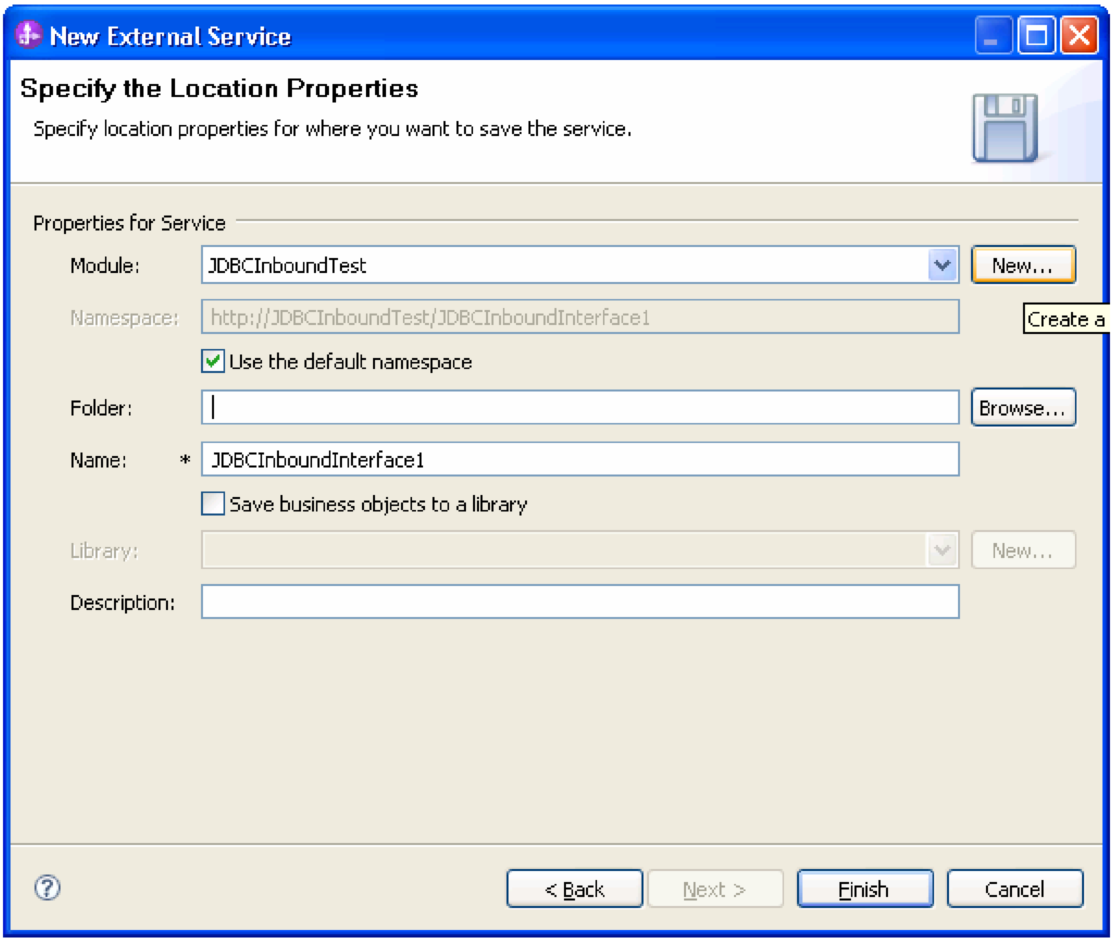
Event query type for processing events: Standard

User-defined event query:

User-defined update query:

User-defined update query for failed event delivery:

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



New External Service

Specify the Location Properties

Specify location properties for where you want to save the service.

Properties for Service

Module: JDBCInboundTest

Namespace: http://JDBCInboundTest/JDBCInboundInterface1

☑ Use the default namespace

Folder:

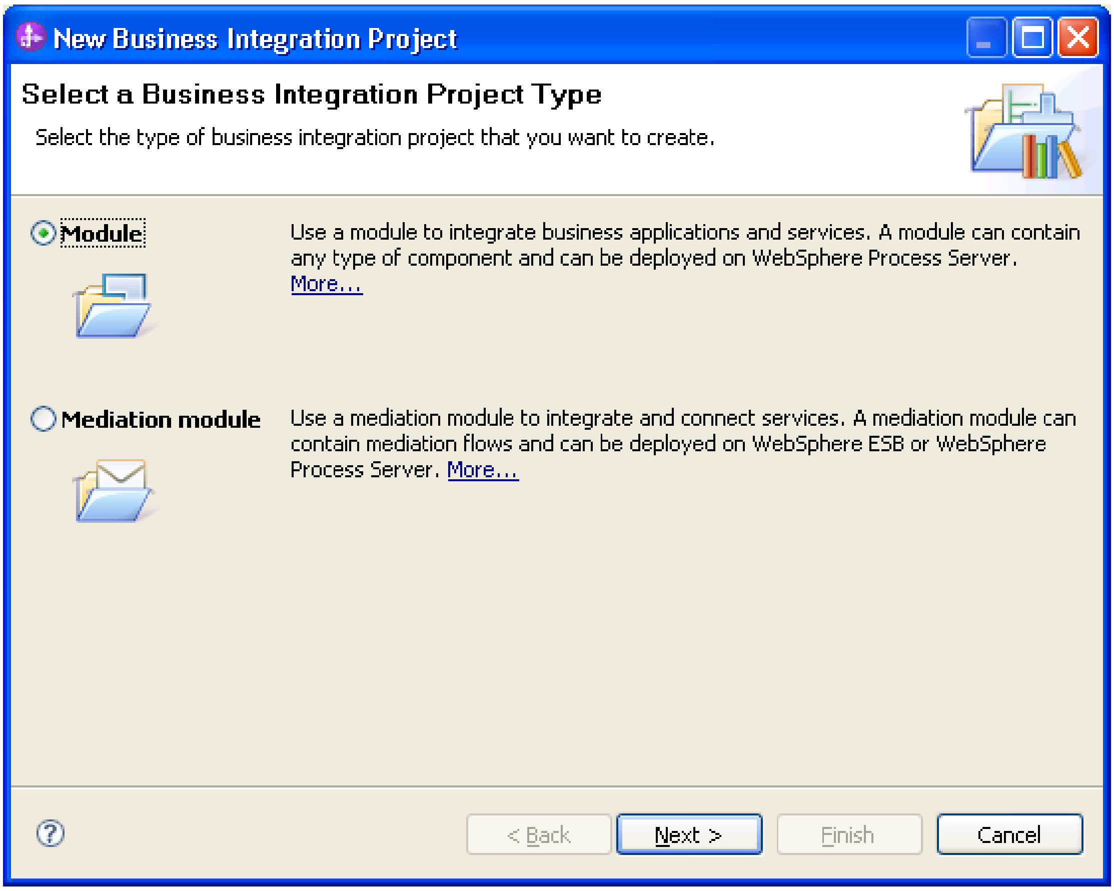
Name: * JDBCInboundInterface1

☐ Save business objects to a library

Library:

Description:

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



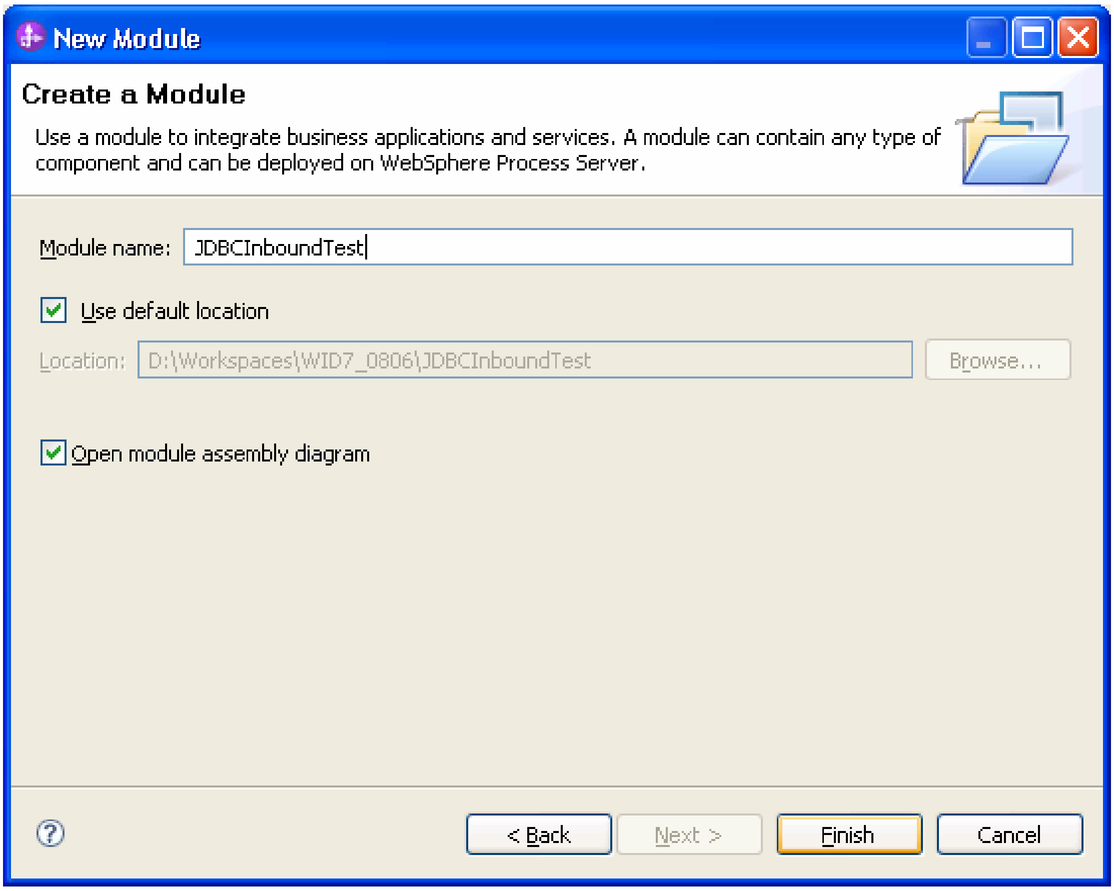
Select a Business Integration Project Type

Select the type of business integration project that you want to create.

● Module — Use a module to integrate business applications and services. A module can contain any type of component and can be deployed on WebSphere Process Server. More...

○ Mediation module — Use a mediation module to integrate and connect services. A mediation module can contain mediation flows and can be deployed on WebSphere ESB or WebSphere Process Server. More...

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



New Module

Create a Module

Use a module to integrate business applications and services. A module can contain any type of component and can be deployed on WebSphere Process Server.

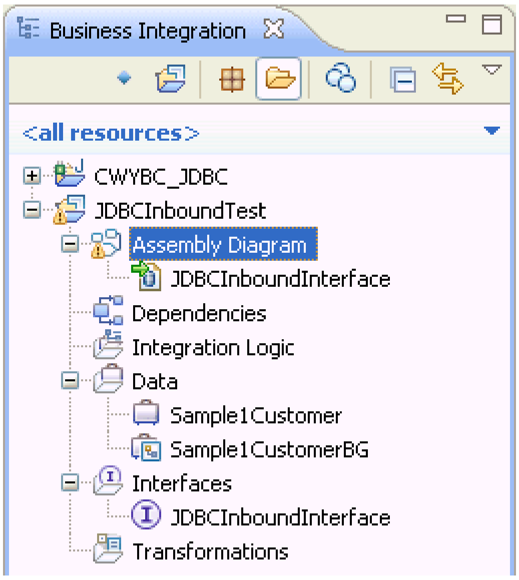
Module name: JDBCInboundTest

☑ Use default location

Location: D:\Workspaces\WID7_0806\JDBCInboundTest

☑ Open module assembly diagram

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



Business Integration

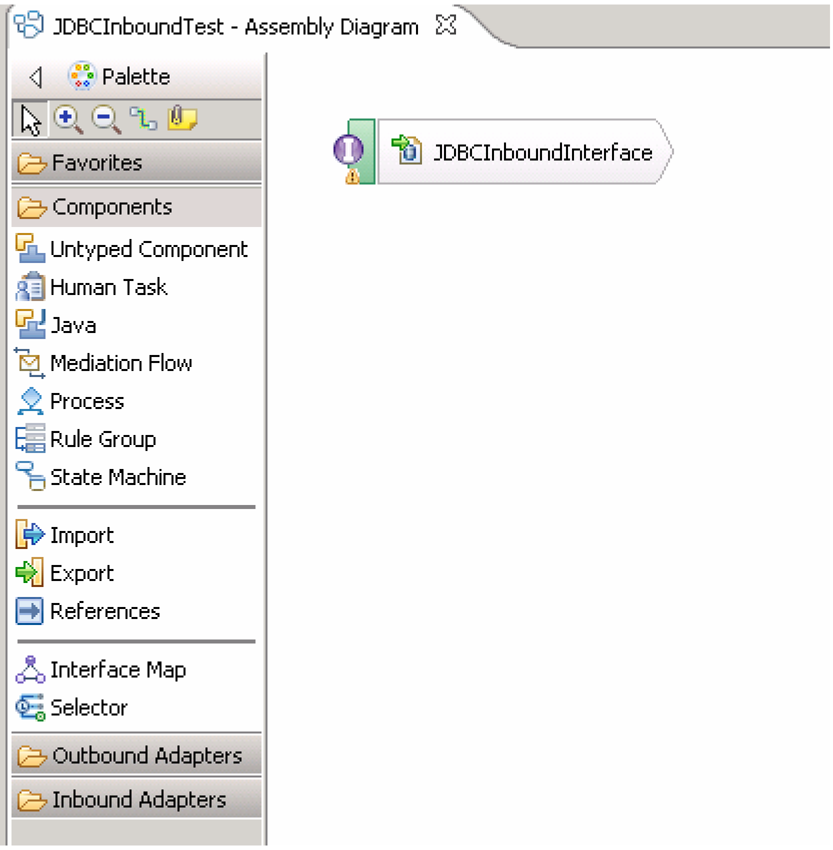
<all resources>

- CWYBC_JDBC
- JDBCInboundTest
  - Assembly Diagram
    - JDBCInboundInterface
  - Dependencies
  - Integration Logic
  - Data
    - Sample1Customer
    - Sample1CustomerBG
  - Interfaces
    - JDBCInboundInterface
  - Transformations

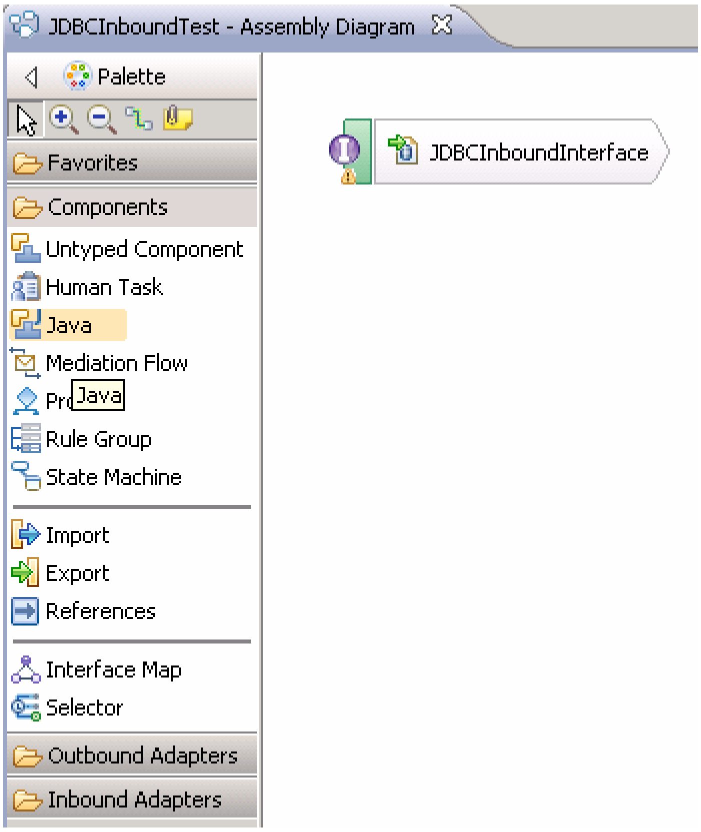
Set up the components to be part of the inbound environment

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

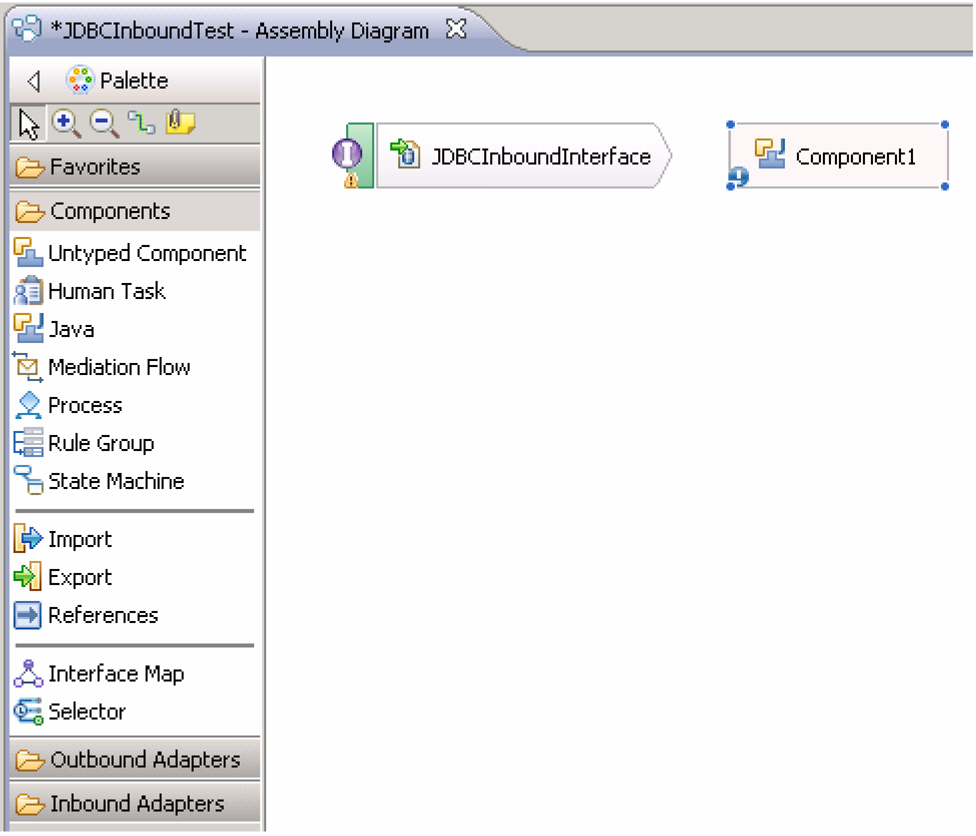
1. Open the Assembly Diagram. It shows the **JDBCInboundInterface** that was generated when the artifacts were created.



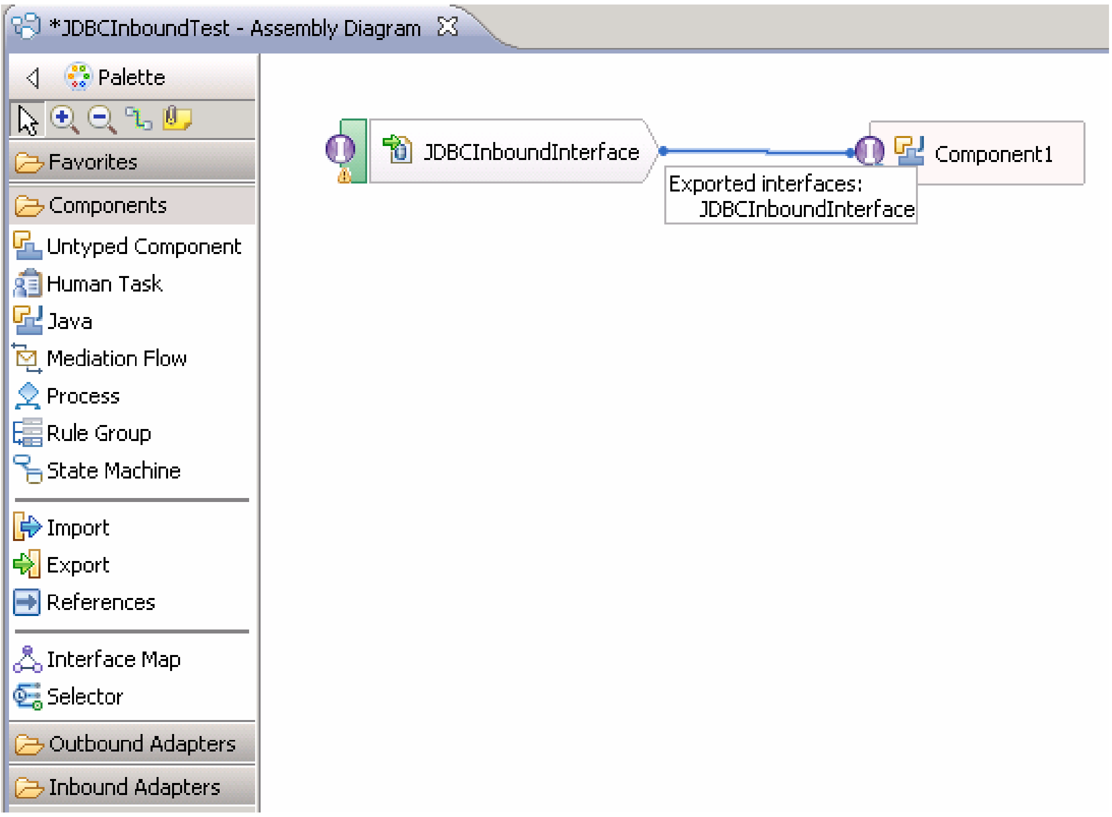
2. From the Palette, select the **Java** component and drop it on the assembly diagram.



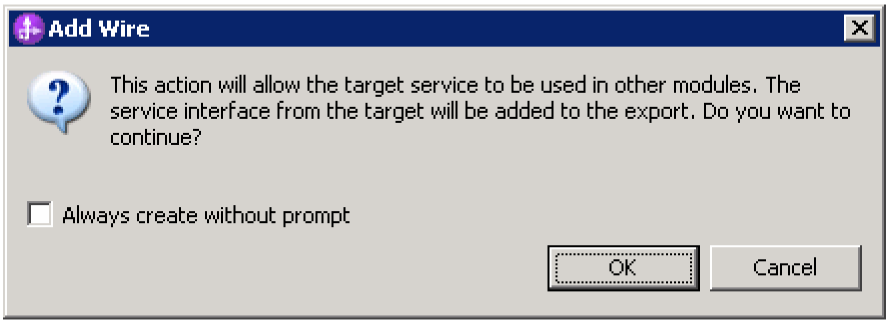
A component named **Component1** is created in the Assembly diagram.



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



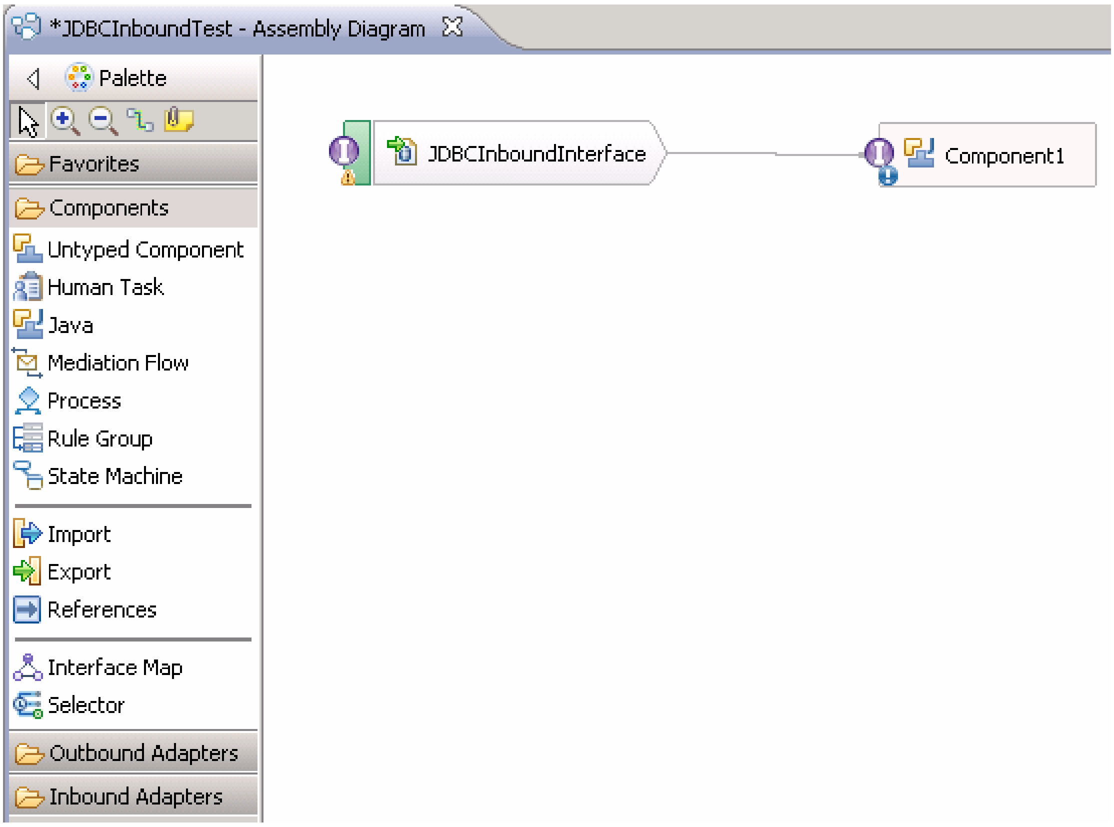
Note: Before the preceding window, i.e., before wiring you will see the following window. Click **OK**.



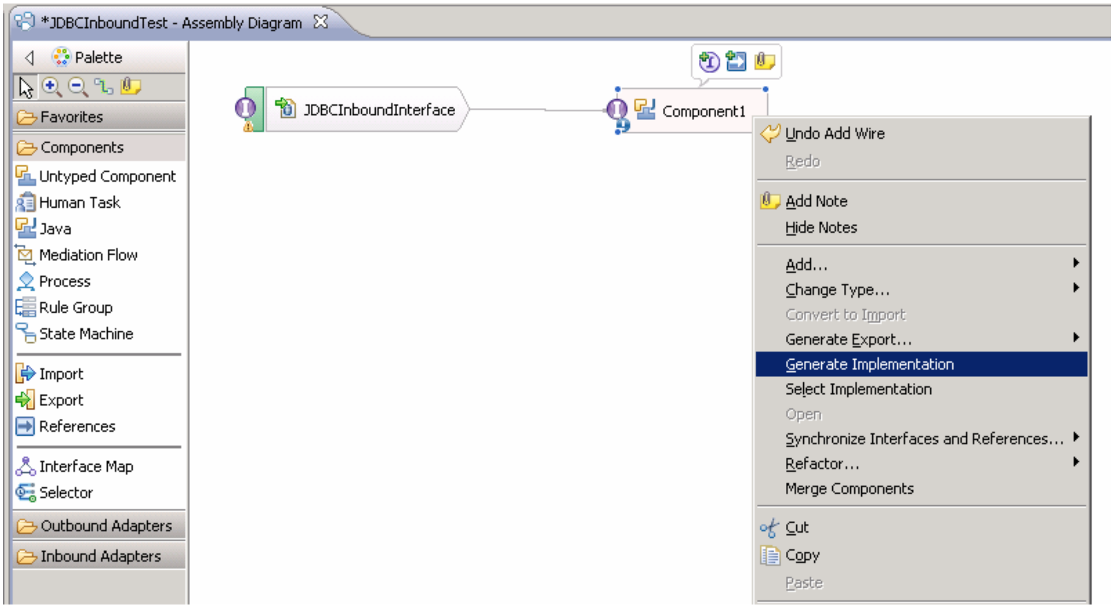
This action will allow the target service to be used in other modules. The service interface from the target will be added to the export. Do you want to continue?

□ Always create without prompt

The Assembly diagram now looks like the figure below.

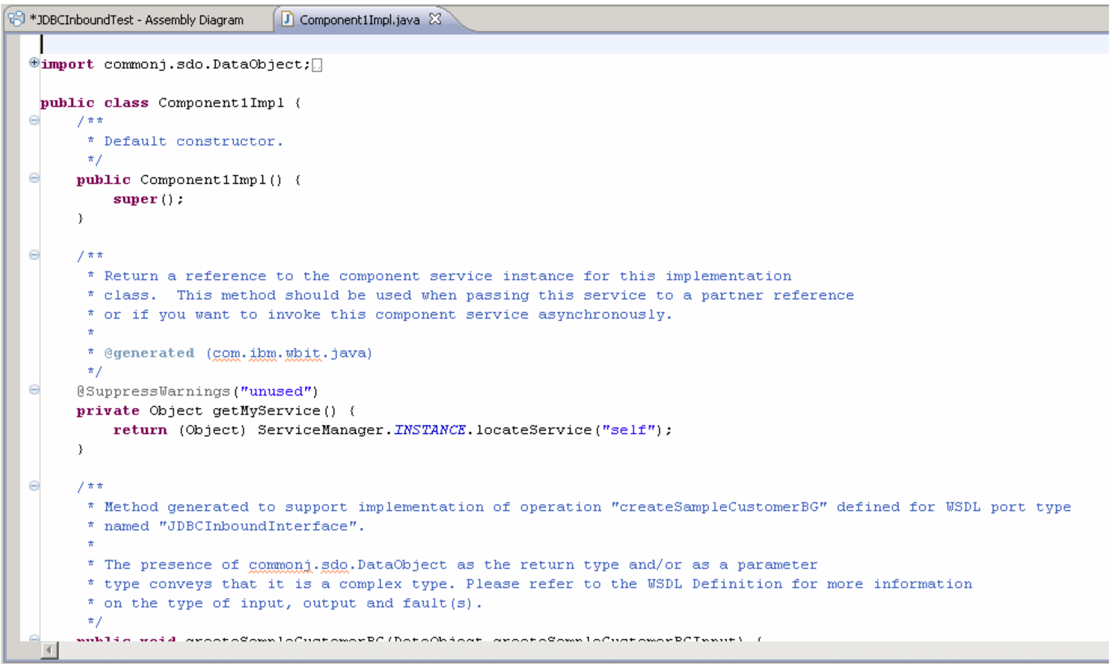


4. Generate the implementation for Java component. Right-click the component, and select **Generate Implementation** to complete service creation.



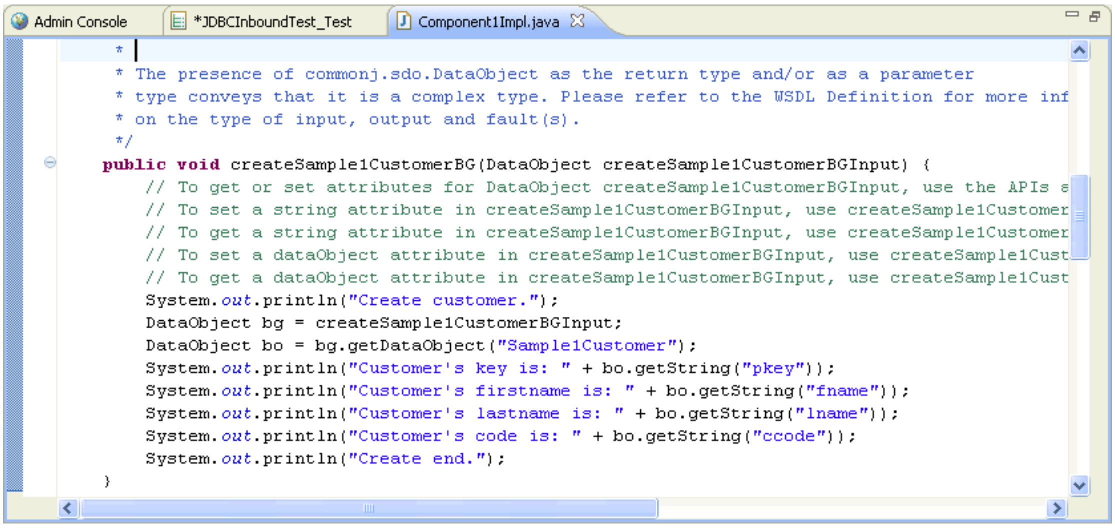
5. Highlight the default package and select **OK**.

The Java Editor displays the Component1Impl.java file.



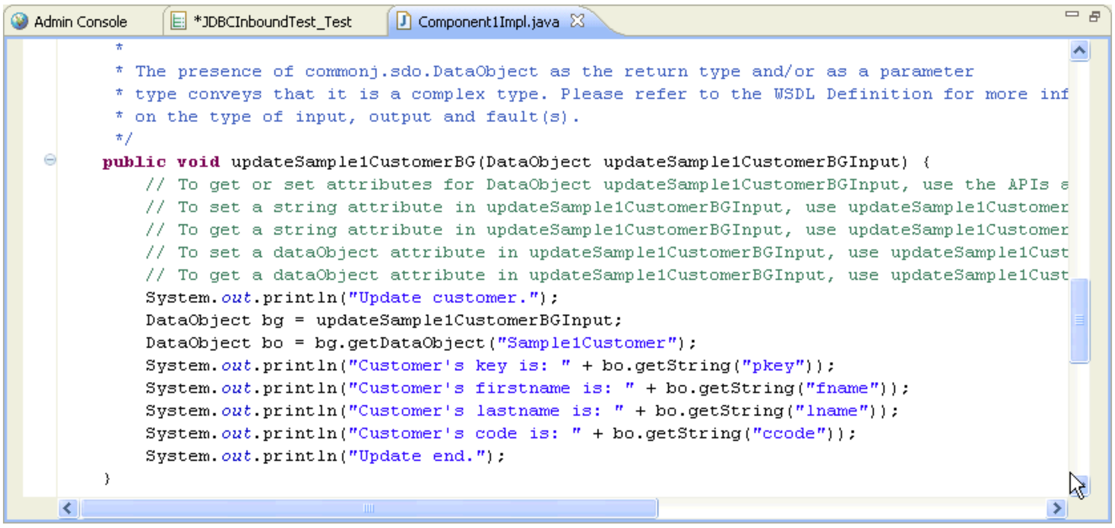
```
import commonj.sdo.DataObject;

public class 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 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 createSample1Customer(DataObject createSample1CustomerBGInput) method that needs to be implemented. Write the code into the method so the complete method looks as follows:



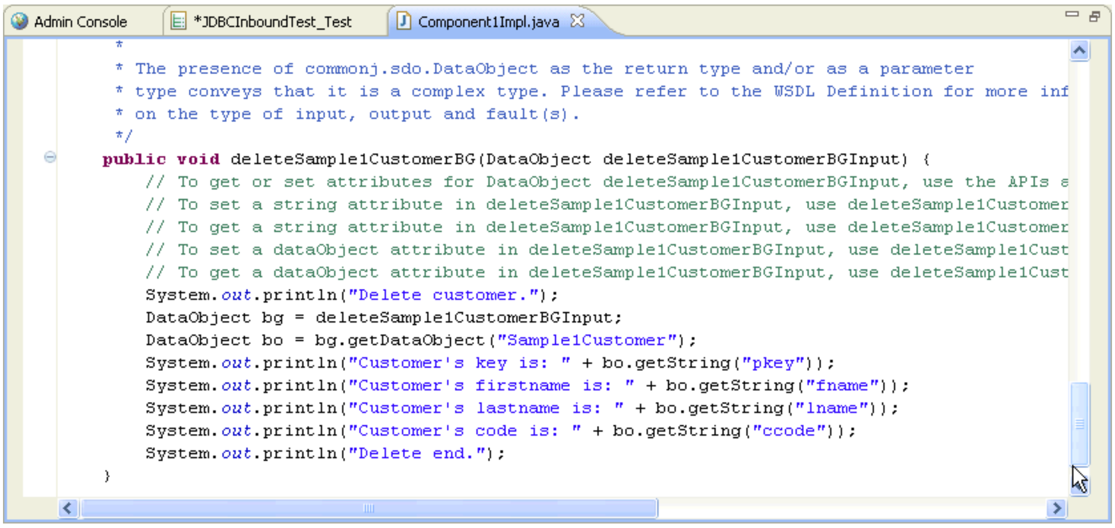
```
     *
     * The presence of commonj.sdo.DataObject as the return type and/or as a parameter
     * type conveys that it is a complex type. Please refer to the WSDL Definition for more inf
     * on the type of input, output and fault(s).
     */
    public void createSample1CustomerBG(DataObject createSample1CustomerBGInput) {
        // To get or set attributes for DataObject createSample1CustomerBGInput, use the APIs a
        // To set a string attribute in createSample1CustomerBGInput, use createSample1Customer
        // To get a string attribute in createSample1CustomerBGInput, use createSample1Customer
        // To set a dataObject attribute in createSample1CustomerBGInput, use createSample1Cust
        // To get a dataObject attribute in createSample1CustomerBGInput, use createSample1Cust
        System.out.println("Create customer.");
        DataObject bg = createSample1CustomerBGInput;
        DataObject bo = bg.getDataObject("Sample1Customer");
        System.out.println("Customer's key is: " + bo.getString("pkey"));
        System.out.println("Customer's firstname is: " + bo.getString("fname"));
        System.out.println("Customer's lastname is: " + bo.getString("lname"));
        System.out.println("Customer's code is: " + bo.getString("ccode"));
        System.out.println("Create end.");
    }
```

7. Scroll down and locate the updateSample1Customer(DataObject updateSample1CustomerBGInput) method that needs to be implemented. Write the code into the method so the complete method looks as follows:



```
     *
     * 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 inf
     * on the type of input, output and fault(s).
     */
    public void updateSample1CustomerBG(DataObject updateSample1CustomerBGInput) {
        // To get or set attributes for DataObject updateSample1CustomerBGInput, use the APIs a
        // To set a string attribute in updateSample1CustomerBGInput, use updateSample1Customer
        // To get a string attribute in updateSample1CustomerBGInput, use updateSample1Customer
        // To set a dataObject attribute in updateSample1CustomerBGInput, use updateSample1Cust
        // To get a dataObject attribute in updateSample1CustomerBGInput, use updateSample1Cust
        System.out.println("Update customer.");
        DataObject bg = updateSample1CustomerBGInput;
        DataObject bo = bg.getDataObject("Sample1Customer");
        System.out.println("Customer's key is: " + bo.getString("pkey"));
        System.out.println("Customer's firstname is: " + bo.getString("fname"));
        System.out.println("Customer's lastname is: " + bo.getString("lname"));
        System.out.println("Customer's code is: " + bo.getString("ccode"));
        System.out.println("Update end.");
    }
```

8. Scroll down and locate the deleteSample1Customer(DataObject deleteSample1CustomerBGInput) method that needs to be implemented. Write the code into the method so the complete method looks as follows:



```
     *
     * 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 inf
     * on the type of input, output and fault(s).
     */
    public void deleteSample1CustomerBG(DataObject deleteSample1CustomerBGInput) {
        // To get or set attributes for DataObject deleteSample1CustomerBGInput, use the APIs a
        // To set a string attribute in deleteSample1CustomerBGInput, use deleteSample1Customer
        // To get a string attribute in deleteSample1CustomerBGInput, use deleteSample1Customer
        // To set a dataObject attribute in deleteSample1CustomerBGInput, use deleteSample1Cust
        // To get a dataObject attribute in deleteSample1CustomerBGInput, use deleteSample1Cust
        System.out.println("Delete customer.");
        DataObject bg = deleteSample1CustomerBGInput;
        DataObject bo = bg.getDataObject("Sample1Customer");
        System.out.println("Customer's key is: " + bo.getString("pkey"));
        System.out.println("Customer's firstname is: " + bo.getString("fname"));
        System.out.println("Customer's lastname is: " + bo.getString("lname"));
        System.out.println("Customer's code is: " + bo.getString("ccode"));
        System.out.println("Delete end.");
    }
```

9. Select **File -> Save** to save your changes.

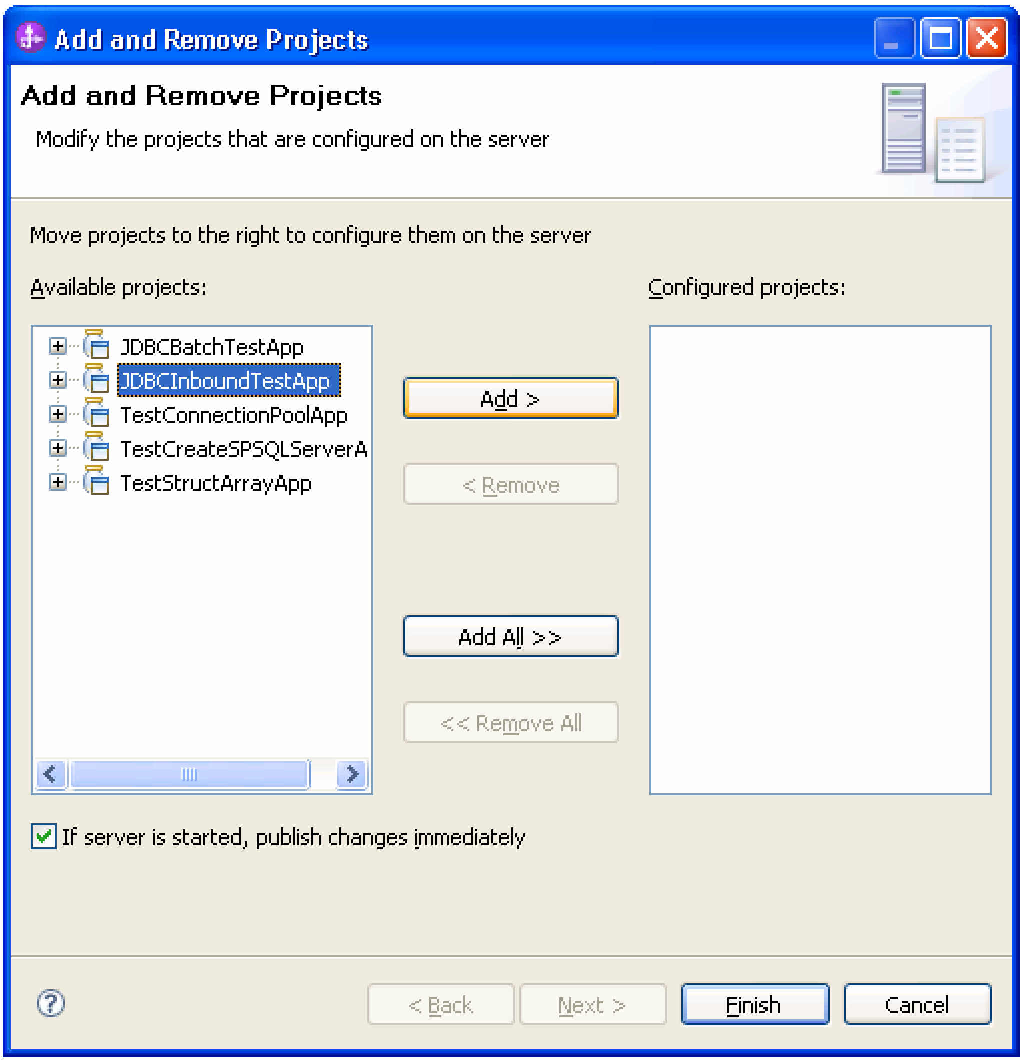
10. Close and save the Assembly Diagram. Wait for the workspace to complete building.

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 IBM Integration Designer integration test client. To do this, you must add the SCA module you created earlier to the server using the **Servers** view in IBM Integration Designer.

Steps for adding the SCA module to the server:

1. In IBM Integration Designer, 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**.



Add and Remove Projects

Add and Remove Projects

Modify the projects that are configured on the server

Move projects to the right to configure them on the server

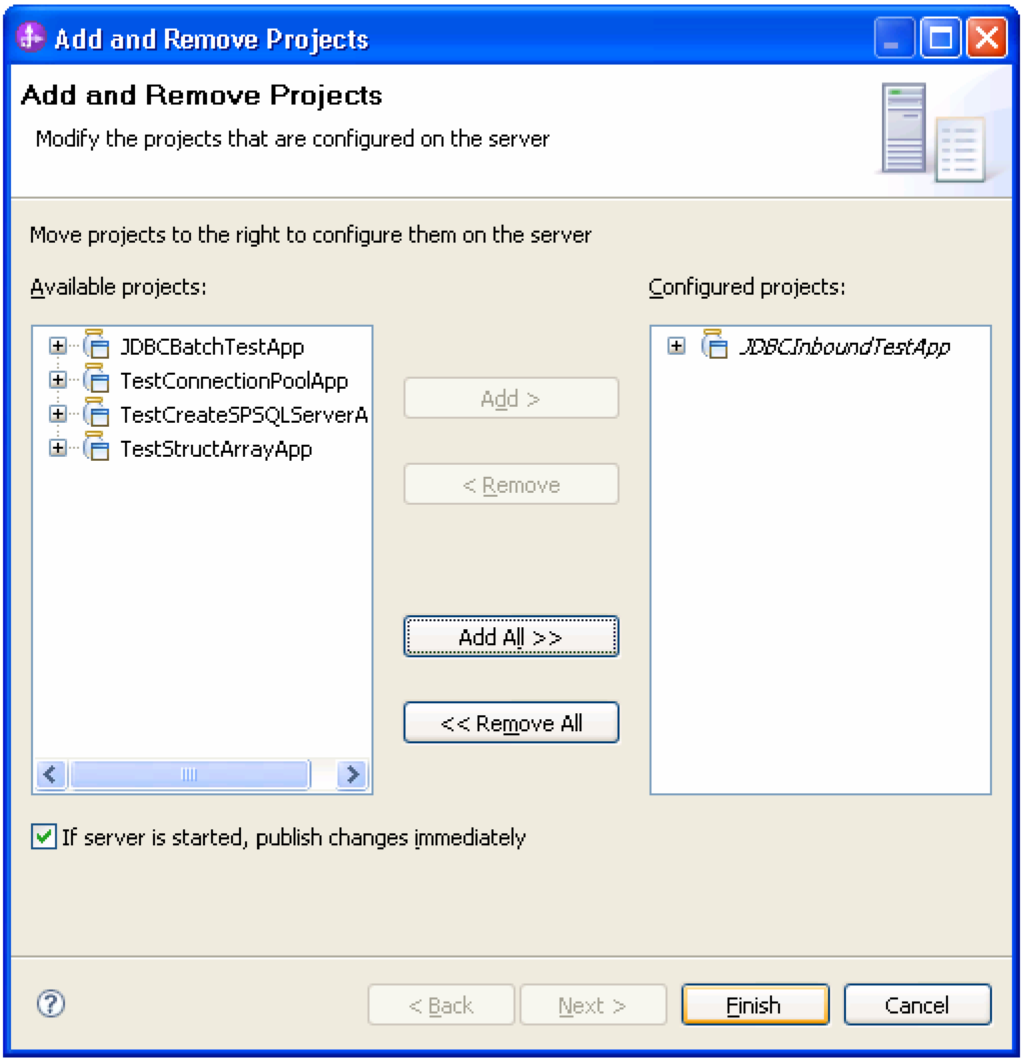
Available projects:

- JDBCBatchTestApp
- JDBCInboundTestApp
- TestConnectionPoolApp
- TestCreateSPSQLServerA
- TestStructArrayApp

Configured projects:

☑ If server is started, publish changes immediately

4. Add the SCA module to the server.



Add and Remove Projects

Modify the projects that are configured on the server

Move projects to the right to configure them on the server

Available projects:

- JDBCBatchTestApp
- TestConnectionPoolApp
- TestCreateSPSQLServerA
- TestStructArrayApp

Configured projects:

- JDBCInboundTestApp

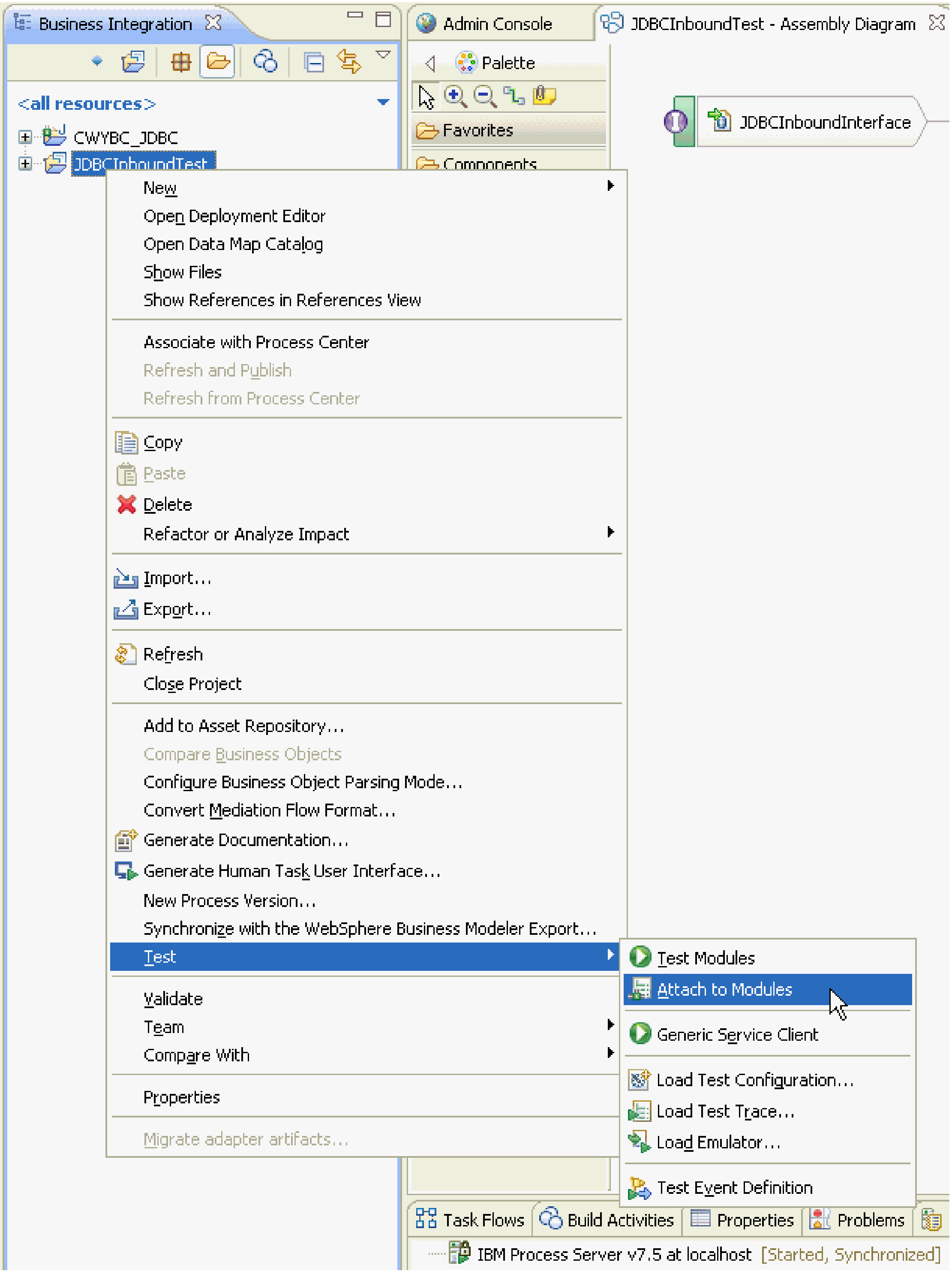
☑ If server is started, publish changes immediately

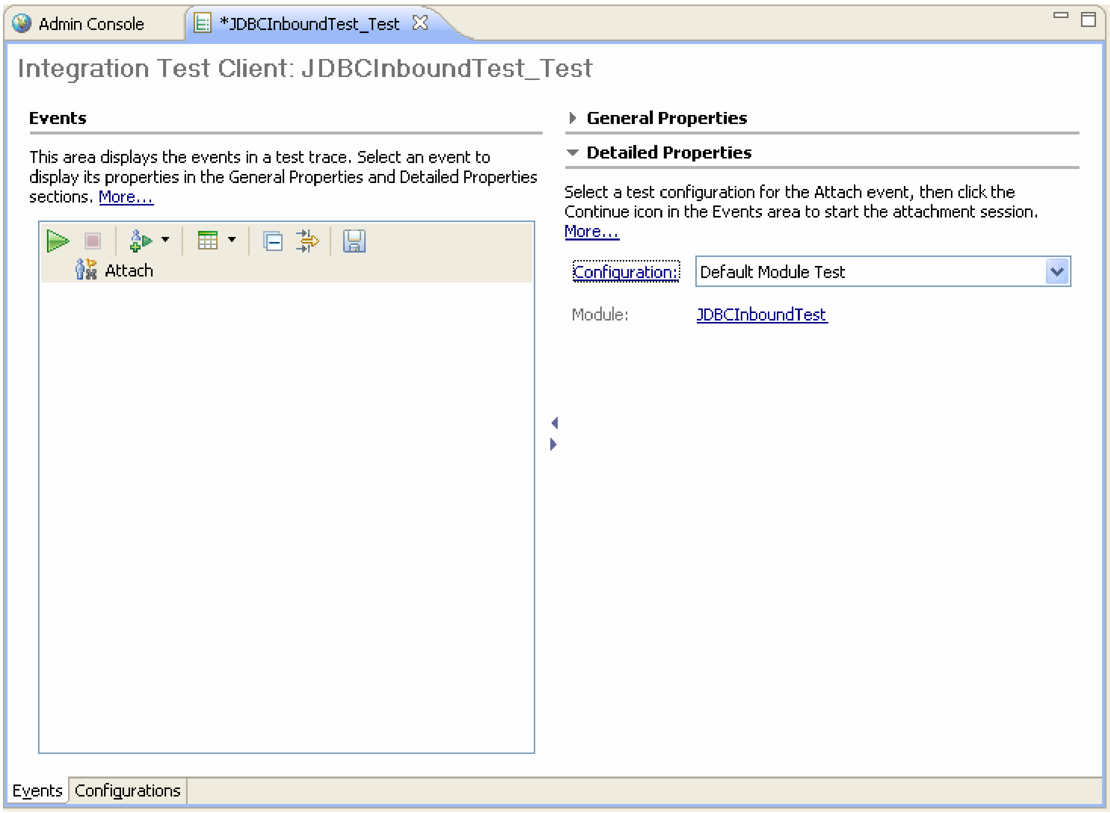
5. Click **Finish**.

Test the assembled adapter application

Test the assembled adapter application using the IBM Integration Designer integration test client.

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


Business Integration
<all resources>
CWYBC_JDBC
JDBCInboundTest
Admin Console
JDBCInboundTest - Assembly Diagram
Palette
Favorites
Components
JDBCInboundInterface
New
Open Deployment Editor
Open Data Map Catalog
Show Files
Show References in References View
Associate with Process Center
Refresh and Publish
Refresh from Process Center
Copy
Paste
Delete
Refactor or Analyze Impact
Import...
Export...
Refresh
Close Project
Add to Asset Repository...
Compare Business Objects
Configure Business Object Parsing Mode...
Convert Mediation Flow Format...
Generate Documentation...
Generate Human Task User Interface...
New Process Version...
Synchronize with the WebSphere Business Modeler Export...
Test
Validate
Team
Compare With
Properties
Migrate adapter artifacts...
Test Modules
Attach to Modules
Generic Service Client
Load Test Configuration...
Load Test Trace...
Load Emulator...
Test Event Definition
Task Flows
Build Activities
Properties
Problems
IBM Process Server v7.5 at localhost [Started, Synchronized]



Integration Test Client: JDBCInboundTest_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

Select a test configuration for the Attach event, then click the Continue icon in the Events area to start the attachment session. More...

Configuration: Default Module Test

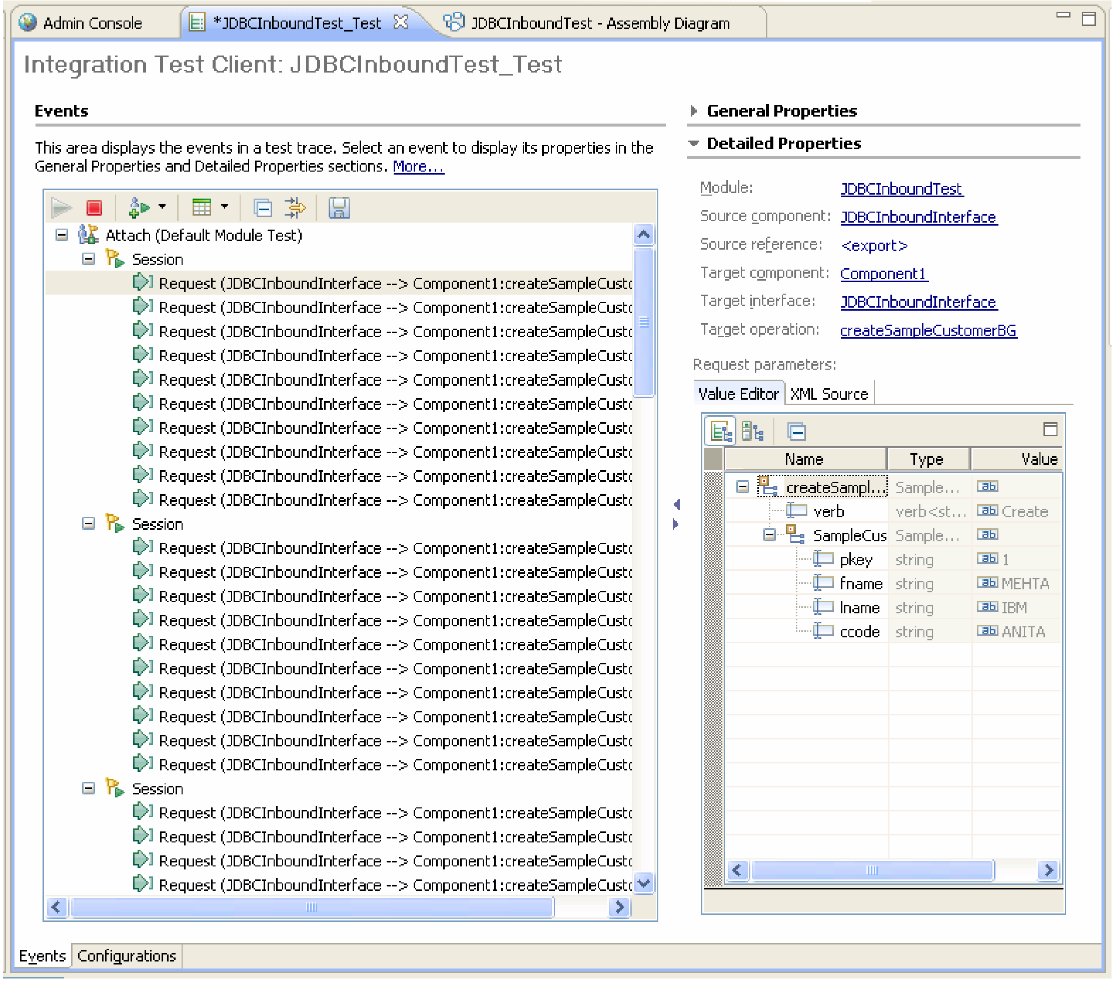
Module: JDBCInboundTest

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:



Integration Test Client: JDBCInboundTest_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...

- Attach (Default Module Test)
  - Session
    - Request (JDBCInboundInterface --> Component1:createSampleCust
    - Request (JDBCInboundInterface --> Component1:createSampleCust
    - Request (JDBCInboundInterface --> Component1:createSampleCust
    - Request (JDBCInboundInterface --> Component1:createSampleCust
    - Request (JDBCInboundInterface --> Component1:createSampleCust
    - Request (JDBCInboundInterface --> Component1:createSampleCust
    - Request (JDBCInboundInterface --> Component1:createSampleCust
    - Request (JDBCInboundInterface --> Component1:createSampleCust
    - Request (JDBCInboundInterface --> Component1:createSampleCust
    - Request (JDBCInboundInterface --> Component1:createSampleCust
  - Session
    - Request (JDBCInboundInterface --> Component1:createSampleCust
    - Request (JDBCInboundInterface --> Component1:createSampleCust
    - Request (JDBCInboundInterface --> Component1:createSampleCust
    - Request (JDBCInboundInterface --> Component1:createSampleCust
    - Request (JDBCInboundInterface --> Component1:createSampleCust
    - Request (JDBCInboundInterface --> Component1:createSampleCust
    - Request (JDBCInboundInterface --> Component1:createSampleCust
    - Request (JDBCInboundInterface --> Component1:createSampleCust
    - Request (JDBCInboundInterface --> Component1:createSampleCust
    - Request (JDBCInboundInterface --> Component1:createSampleCust
  - Session
    - Request (JDBCInboundInterface --> Component1:createSampleCust
    - Request (JDBCInboundInterface --> Component1:createSampleCust
    - Request (JDBCInboundInterface --> Component1:createSampleCust
    - Request (JDBCInboundInterface --> Component1:createSampleCust

General Properties

Detailed Properties

Module: JDBCInboundTest
Source component: JDBCInboundInterface
Source reference: <export>
Target component: Component1
Target interface: JDBCInboundInterface
Target operation: createSampleCustomerBG

Request parameters:

Value Editor | XML Source

Name	Type	Value
createSampl...	Sample...	
verb	verb<st...	Create
SampleCus	Sample...	
pkey	string	1
fname	string	MEHTA
lname	string	IBM
ccode	string	ANITA

Clear the sample content

Nothing is required to clean up after this tutorial.

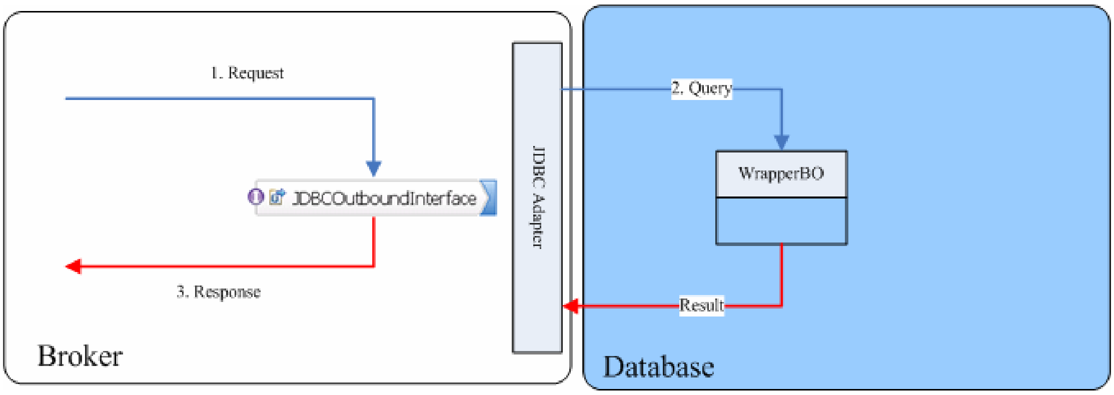
Chapter 11. Tutorial 10: Generate wrapper business objects (Oracle)

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 7.5.0.0 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.

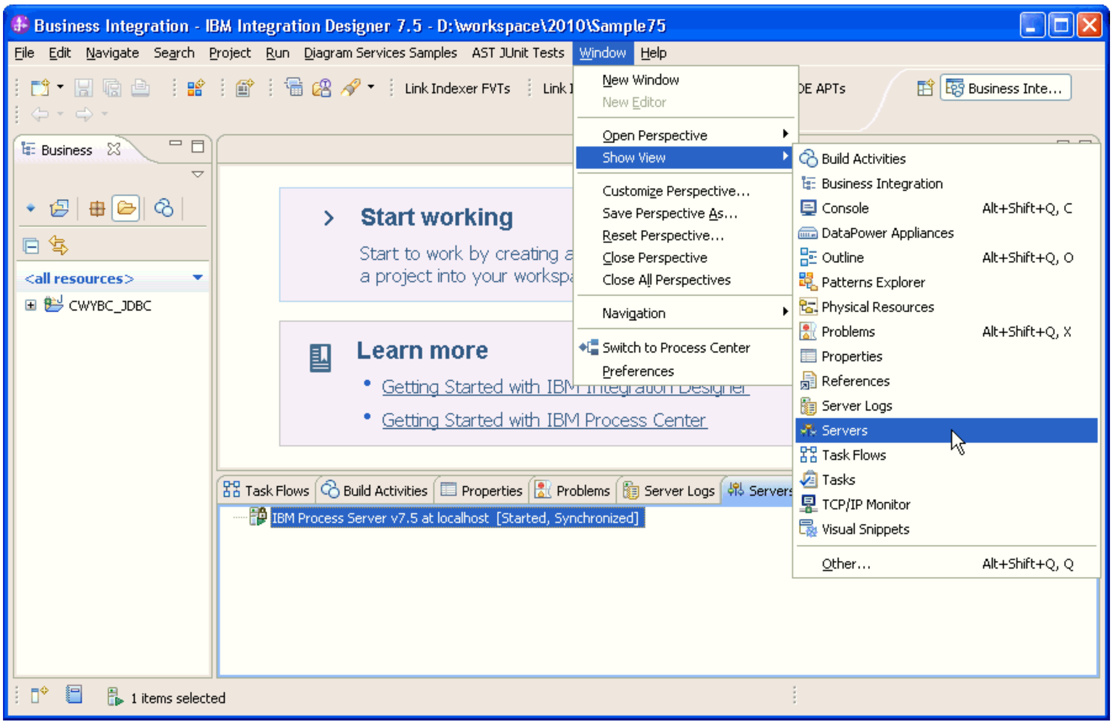
```
insert into customer values ('1000', 'testFname',
'testLname', 'testCcode')
```

Create an authentication alias

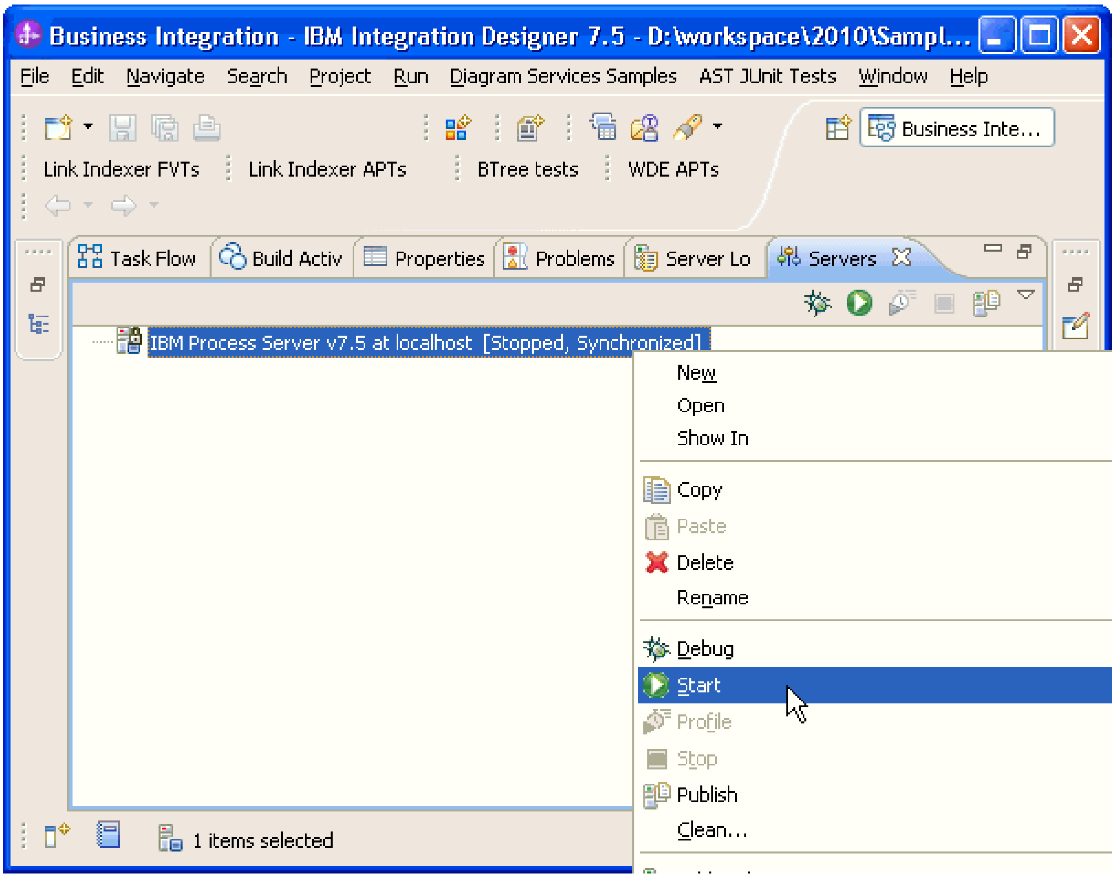
The authentication alias needs to be set because the data source created in the next section uses the username and password set in the authentication alias to connect to the database.

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

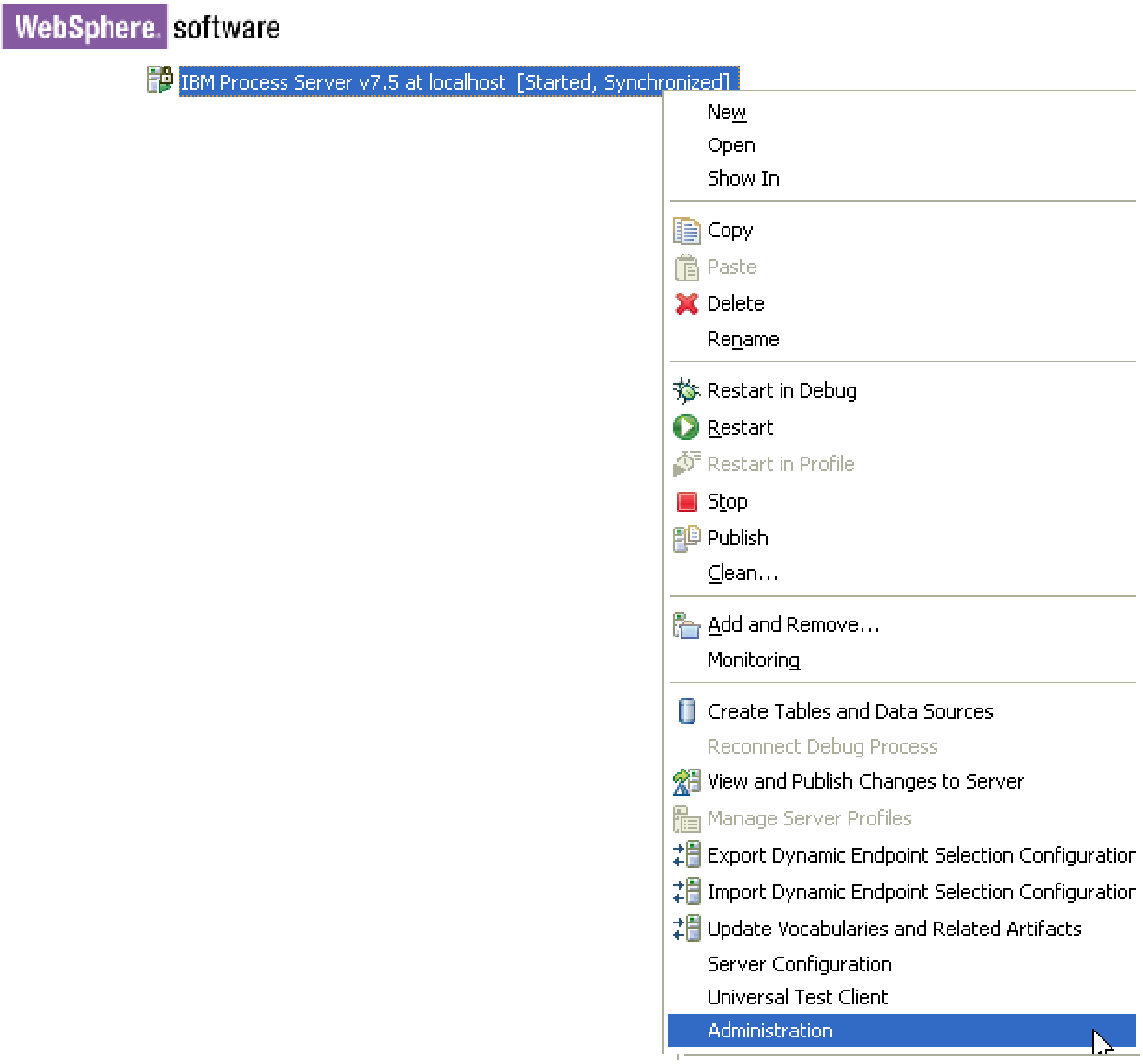
1. In IBM Integration Designer, switch to the **Servers** view by selecting **Window > Show View > Servers**.



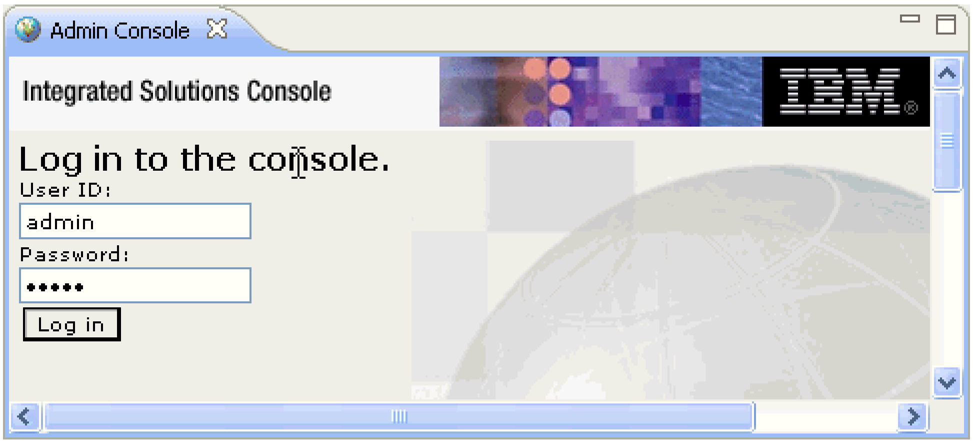
2. In the **Servers** view, right-click the server that you want to start and select **Start**.


Business Integration - IBM Integration Designer 7.5 - D:\workspace\2010\Sampl...
File Edit Navigate Search Project Run Diagram Services Samples AST JUnit Tests Window Help
Business Inte...
Link Indexer FVTs
Link Indexer APTs
BTree tests
WDE APTs
Task Flow
Build Activ
Properties
Problems
Server Lo
Servers
IBM Process Server v7.5 at localhost [Stopped, Synchronized]
New
Open
Show In
Copy
Paste
Delete
Rename
Debug
Start
Profile
Stop
Publish
Clean...
1 items selected

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


WebSphere. software
IBM Process Server v7.5 at localhost [Started, Synchronized]
New
Open
Show In
Copy
Paste
Delete
Rename
Restart in Debug
Restart
Restart in Profile
Stop
Publish
Clean...
Add and Remove...
Monitoring
Create Tables and Data Sources
Reconnect Debug Process
View and Publish Changes to Server
Manage Server Profiles
Export Dynamic Endpoint Selection Configuratior
Import Dynamic Endpoint Selection Configuratior
Update Vocabularies and Related Artifacts
Server Configuration
Universal Test Client
Administration

4. Log on to the administrative console.

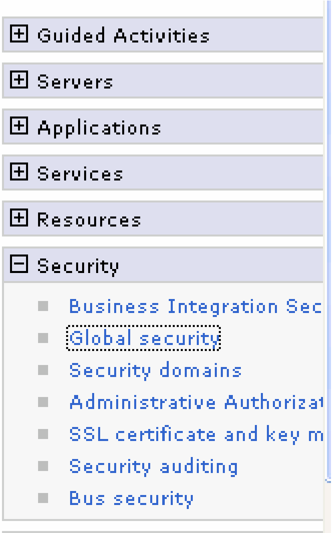


Log in to the console.

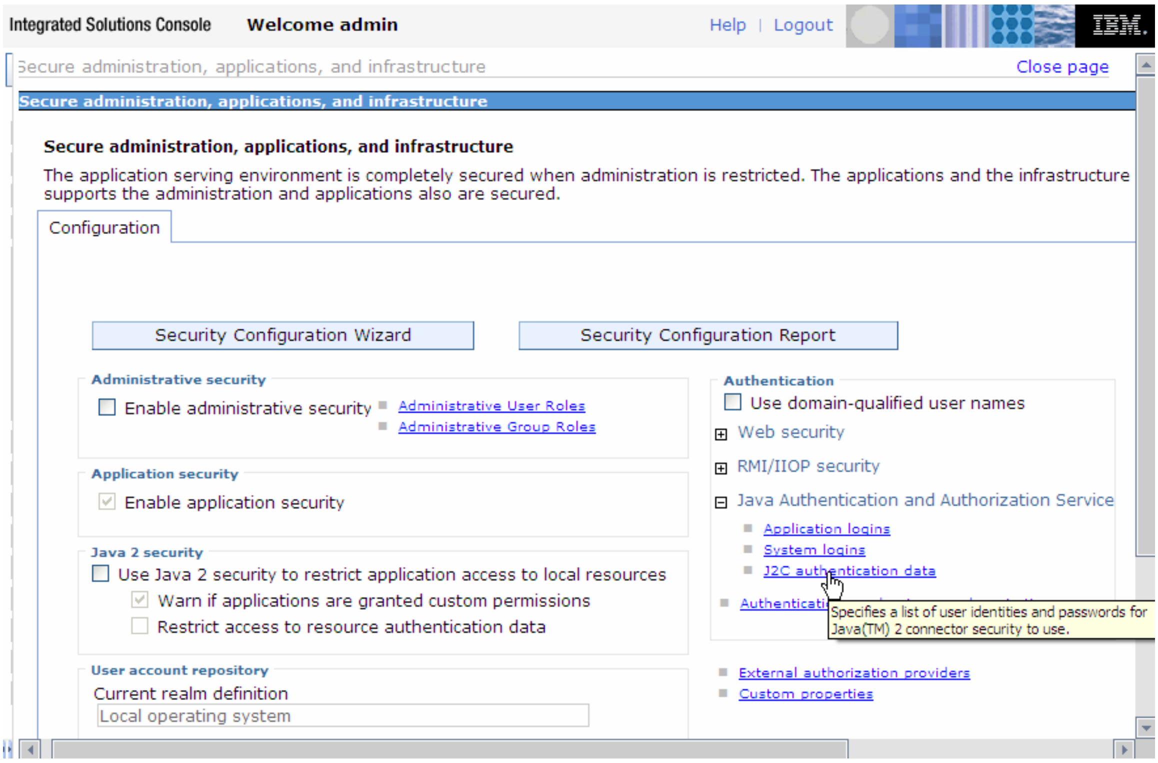
User ID: admin

Password: •••••

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

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

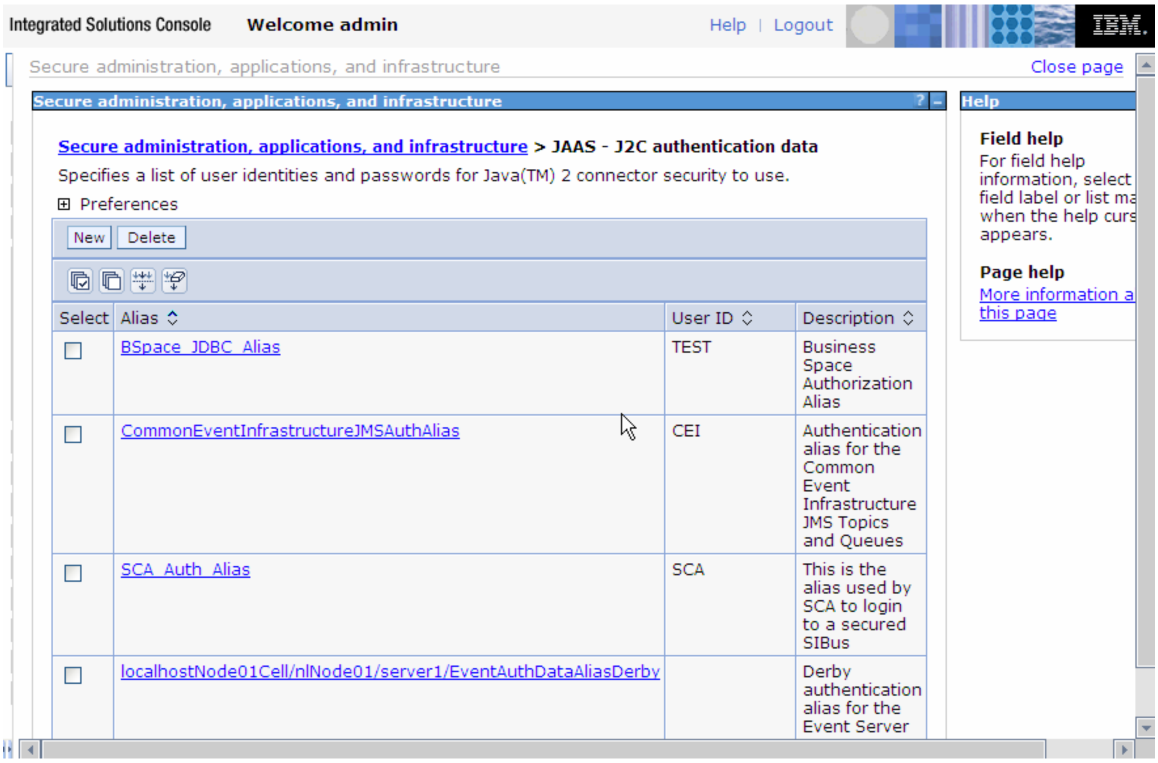
⊟ Java Authentication and Authorization Service
- Application logins
- System logins
- J2C authentication data

- Authenticati

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

Secure administration, applications, and infrastructure Close page

Secure administration, applications, and infrastructure

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.

⊞ Preferences

New Delete

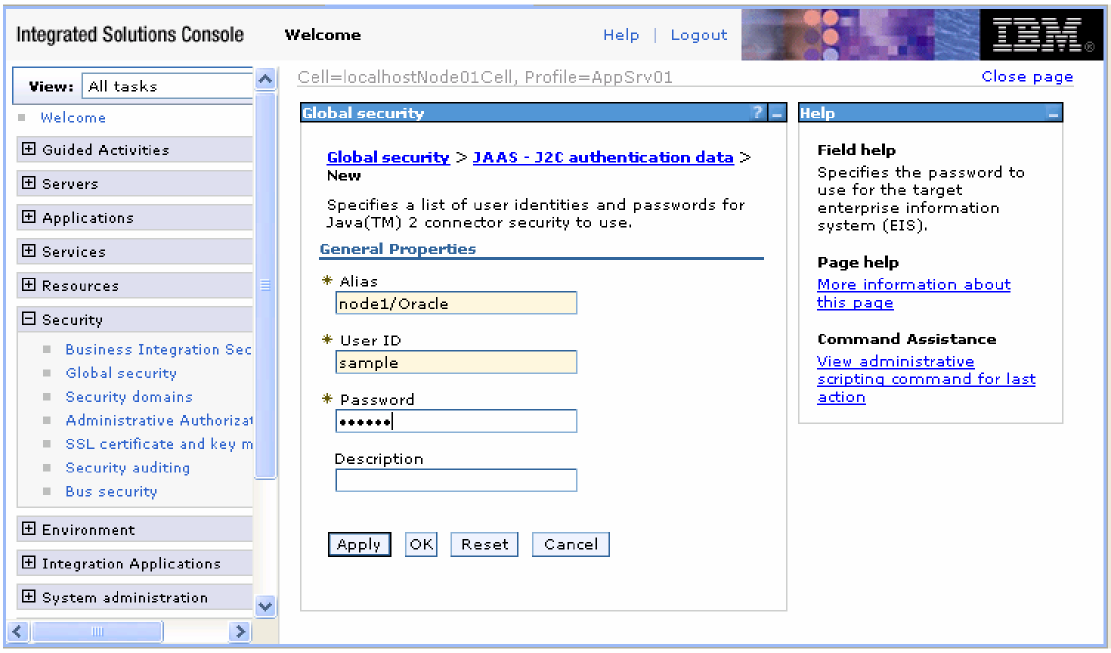
Select	Alias ⇕	User ID ⇕	Description ⇕
□	BSpace_JDBC_Alias	TEST	Business Space Authorization Alias
□	CommonEventInfrastructureJMSAuthAlias	CEI	Authentication alias for the Common Event Infrastructure JMS Topics and Queues
□	SCA_Auth_Alias	SCA	This is the alias used by SCA to login to a secured SIBus
□	localhostNode01Cell/nlNode01/server1/EventAuthDataAliasDerby		Derby authentication alias for the Event Server

Help

Field help
For field help information, select field label or list ma when the help curs appears.

Page help
More information a this page

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



Integrated Solutions Console Welcome Help | Logout

View: All tasks

Cell=localhostNode01Cell, Profile=AppSrv01 Close page

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: node1/Oracle

* User ID: sample

* Password: ••••••

Description:

Apply OK Reset Cancel

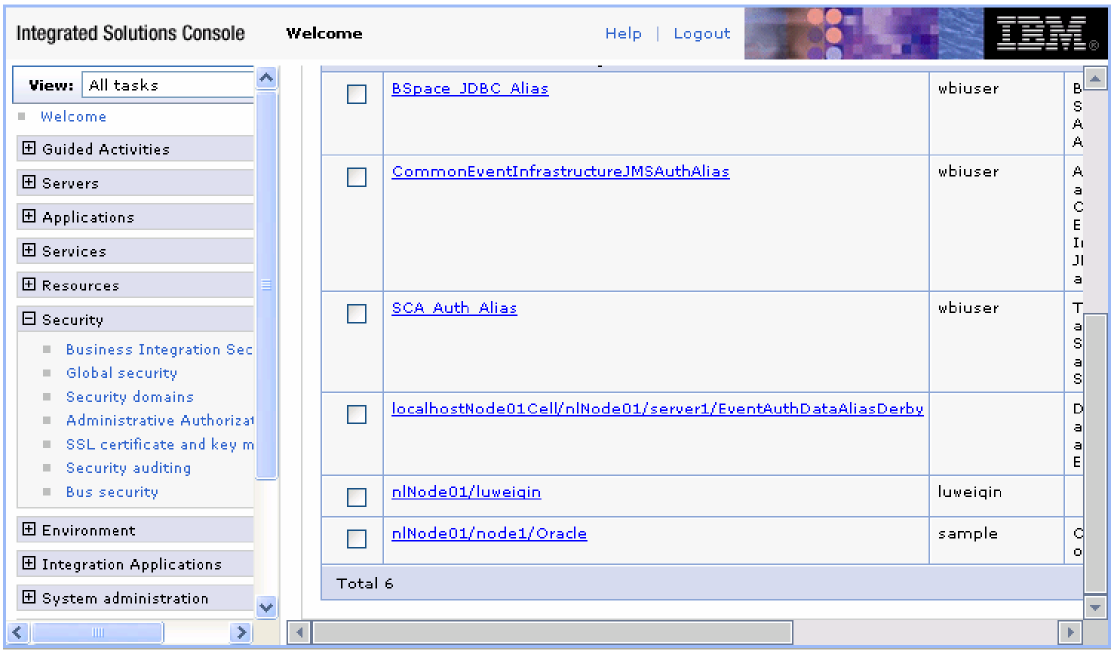
Help

Field help
Specifies the password to use for the target enterprise information system (EIS).

Page help
More information about this page

Command Assistance
View administrative scripting command for last action

You have created an authentication alias that will be used to configure the adapter properties.



□	BSpace_JDBC_Alias	wbiuser	B S A A
□	CommonEventInfrastructureJMSAuthAlias	wbiuser	A a C E I JI a
□	SCA_Auth_Alias	wbiuser	T a S a S
□	localhostNode01Cell/nlNode01/server1/EventAuthDataAliasDerby		D a a E
□	nlNode01/luweiqin	luweiqin	
□	nlNode01/node1/Oracle	sample	C o

Total 6

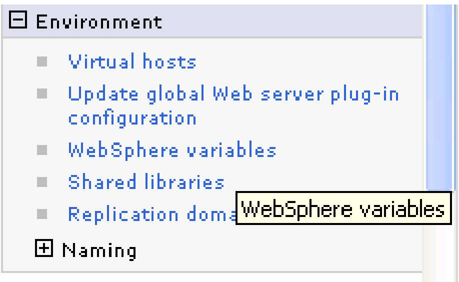
Create a data source

Create a data source in IBM 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.

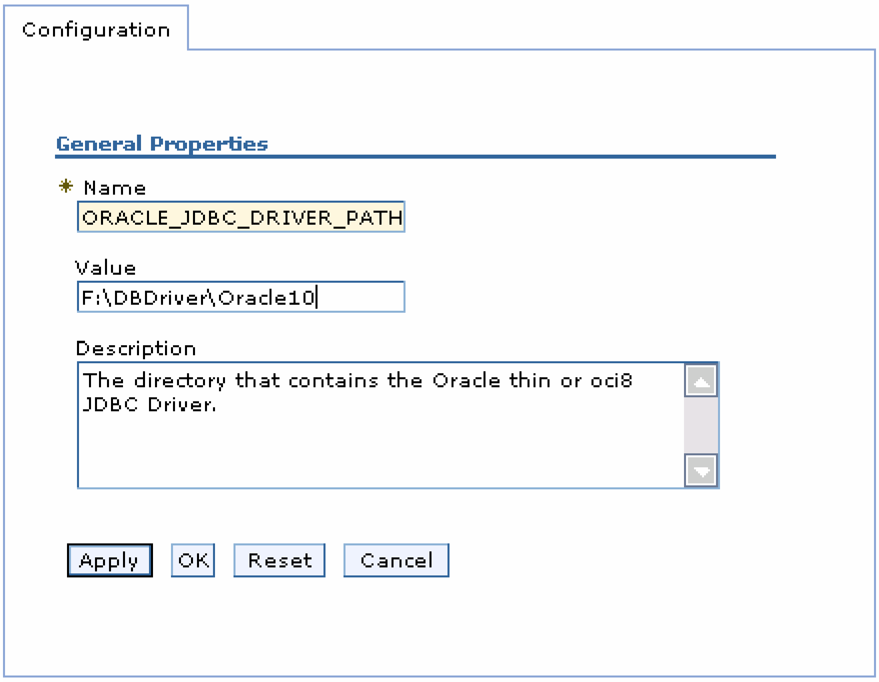
Note: This tutorial will use Oracle as the database and the Oracle thin driver, ojdbc6.jar.

Here are the steps to create the data source in the IBM 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**.

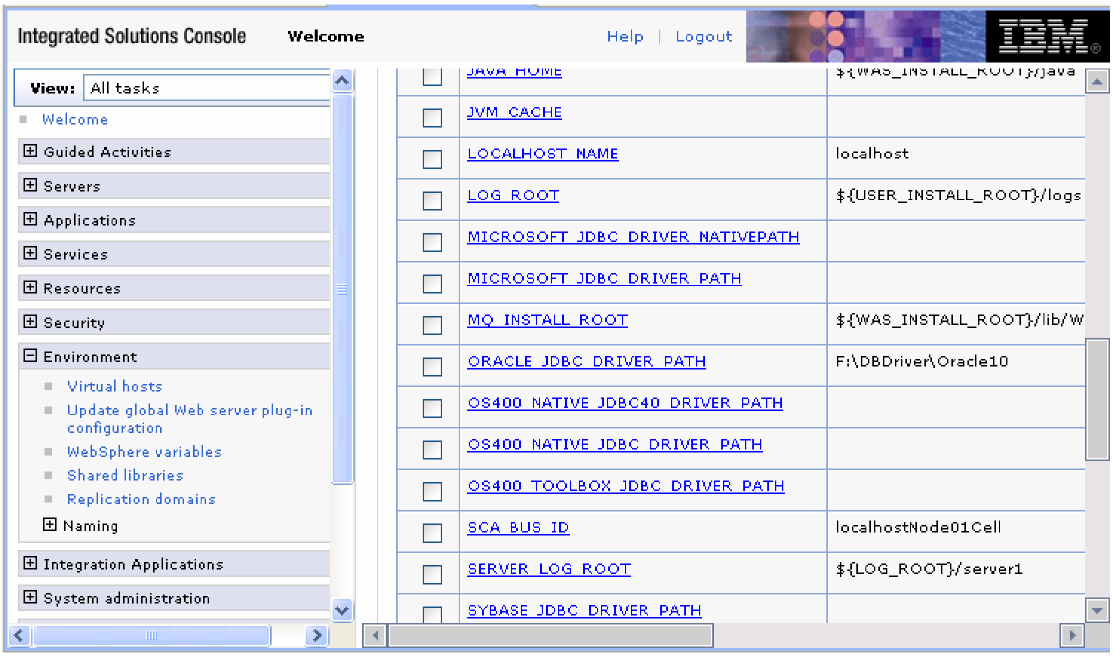


* Name: ORACLE_JDBC_DRIVER_PATH

Value: F:\DBDriver\Oracle10

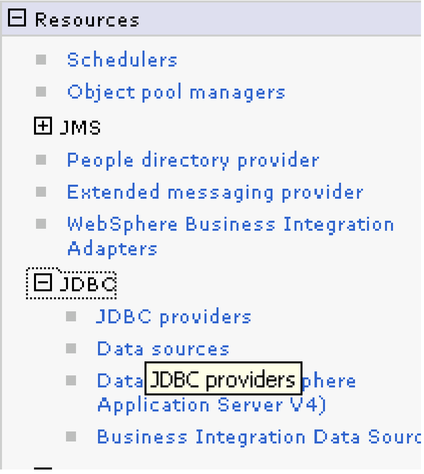
Description: The directory that contains the Oracle thin or oci8 JDBC Driver.

The variable is added and appears in the list.



□	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	

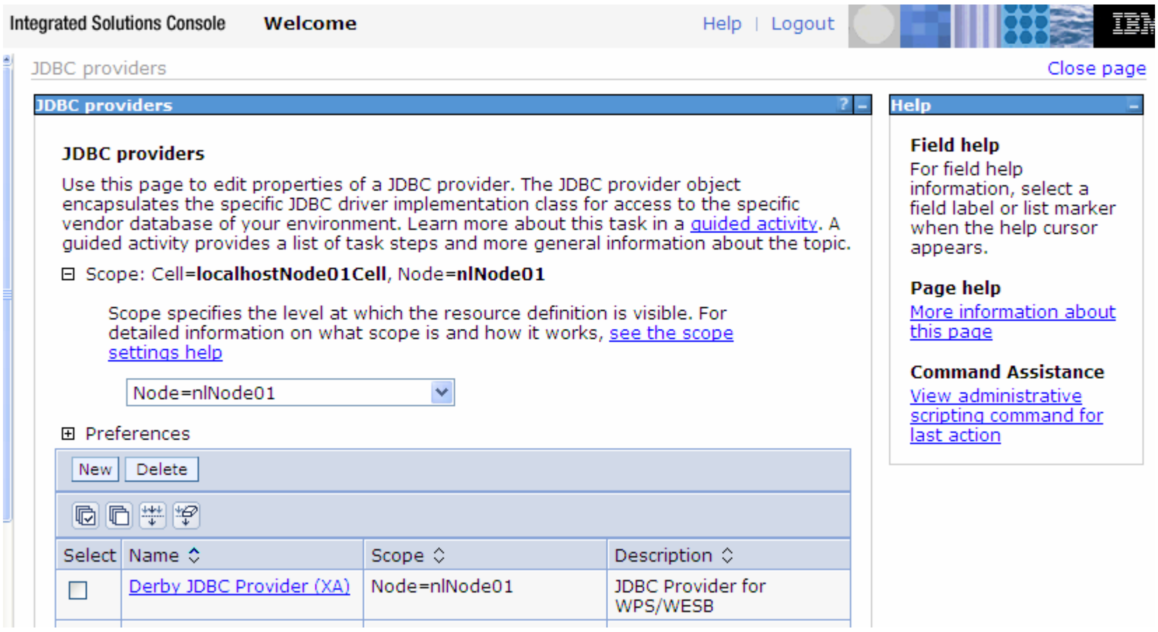
3. Select **Resources → JDBC -> JDBC Providers.**



Resources

- Schedulers
- Object pool managers
- JMS
- People directory provider
- Extended messaging provider
- WebSphere Business Integration Adapters
- JDBC
  - JDBC providers
  - Data sources
  - Data JDBC providers phere Application Server V4)
  - Business Integration Data Sourc

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



Integrated Solutions Console Welcome Help | Logout

JDBC providers Close page

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=**nlNode01**

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=nlNode01

⊞ Preferences

New Delete

Select	Name ⇕	Scope ⇕	Description ⇕
□	Derby JDBC Provider (XA)	Node=nlNode01	JDBC Provider for WPS/WESB

Help

Field help

For field help information, select a field label or list marker when the help cursor appears.

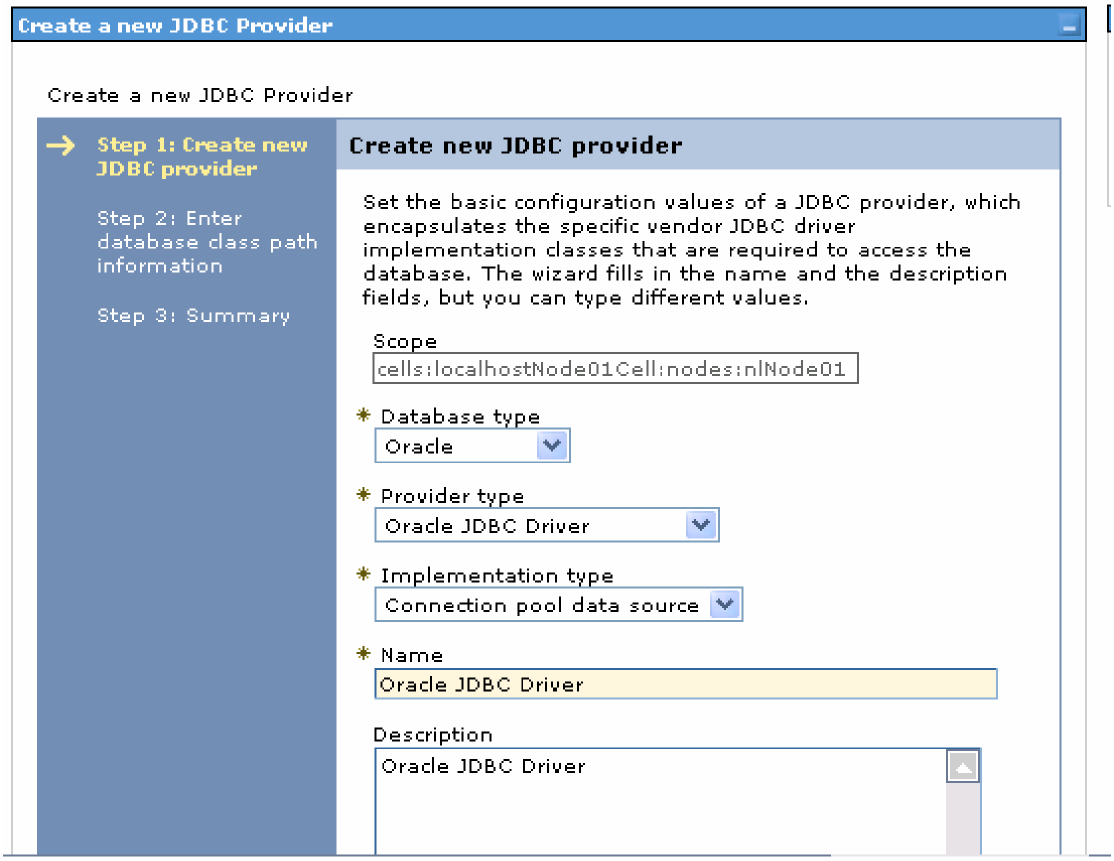
Page help

More information about this page

Command Assistance

View administrative scripting command for last action

5. Click **New**. Select the 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:nlNode01

* Database type
Oracle

* Provider type
Oracle JDBC Driver

* Implementation type
Connection pool data source

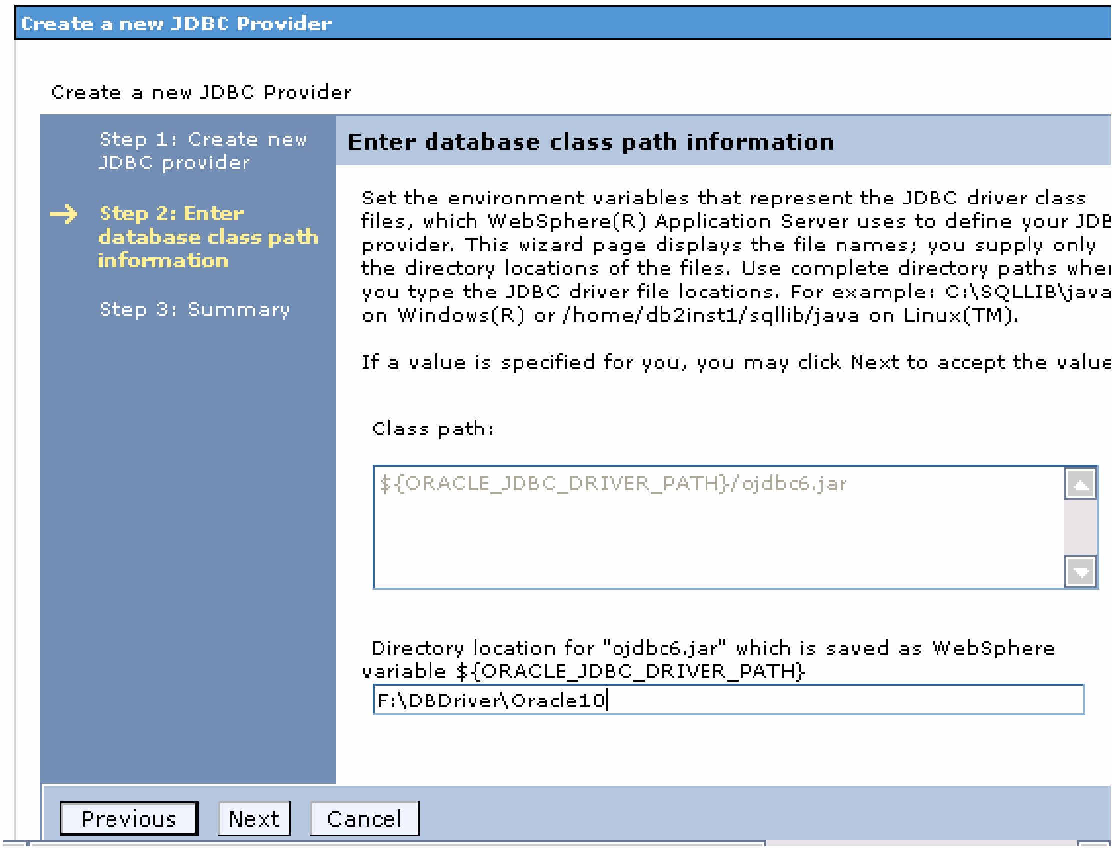
* Name
Oracle JDBC Driver

Description
Oracle JDBC Driver

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.**



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 JDB provider. This wizard page displays the file names; you supply only the directory locations of the files. Use complete directory paths whe 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

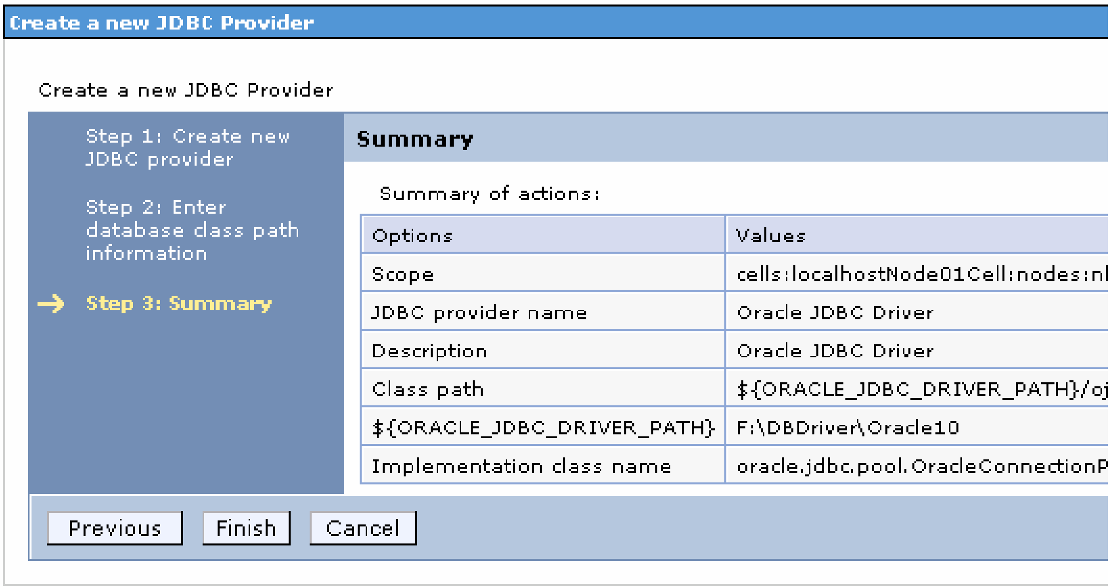
Class path:

${ORACLE_JDBC_DRIVER_PATH}/ojdbc6.jar

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

F:\DBDriver\Oracle10

8. Click **Finish**.



Create a new JDBC Provider

Create a new JDBC Provider

Step 1: Create new JDBC provider

Step 2: Enter database class path information

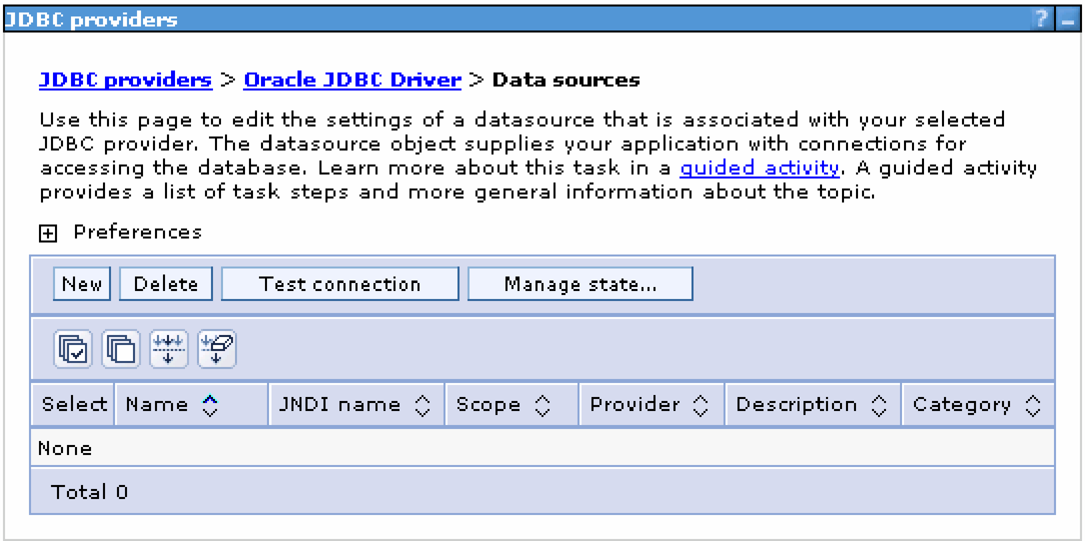
→ Step 3: Summary

Summary

Summary of actions:

Options	Values
Scope	cells:localhostNode01Cell:nodes:nl
JDBC provider name	Oracle JDBC Driver
Description	Oracle JDBC Driver
Class path	${ORACLE_JDBC_DRIVER_PATH}/oj
${ORACLE_JDBC_DRIVER_PATH}	F:\DBDriver\Oracle10
Implementation class name	oracle.jdbc.pool.OracleConnectionP

9. Under **Additional Properties**, select **Data sources**. Click **New**.



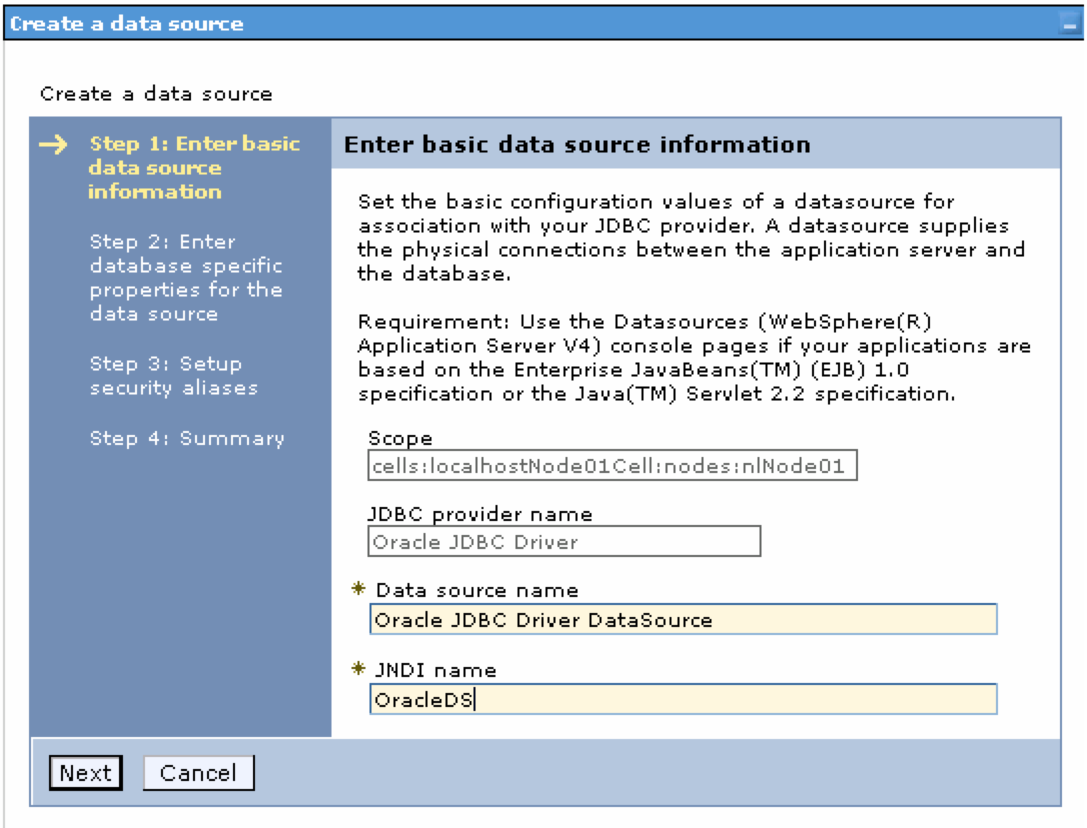
JDBC providers > Oracle JDBC Driver > Data sources

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

Select	Name	JNDI name	Scope	Provider	Description	Category
None						
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**.


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
Next
Cancel

Create a data source

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.

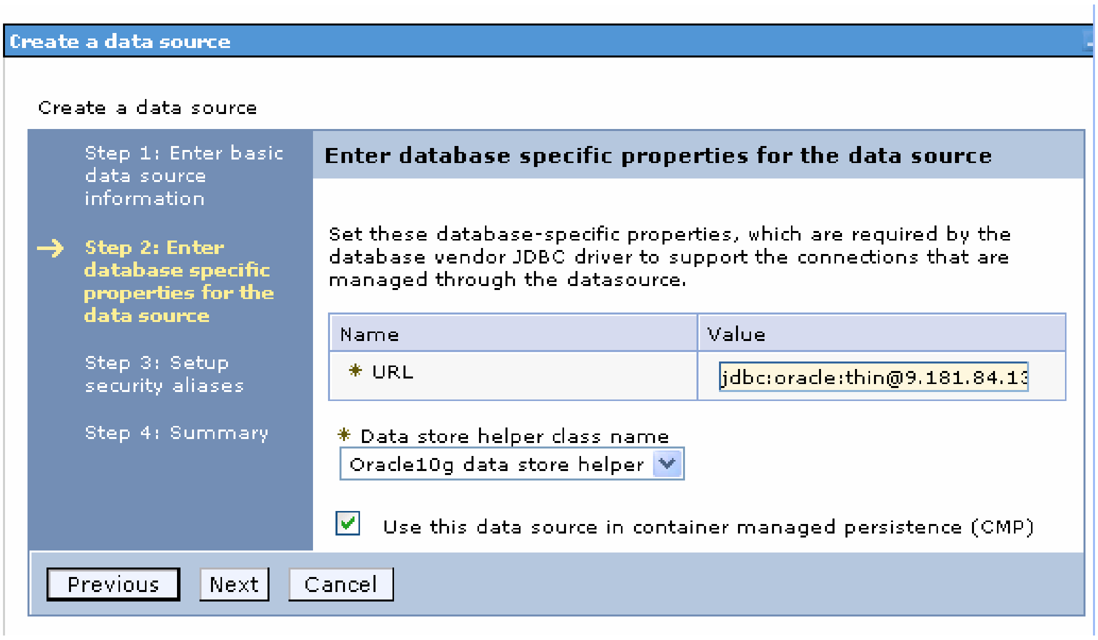
Scope: cells:localhostNode01Cell:nodes:nlNode01

JDBC provider name: Oracle JDBC Driver

* Data source name: Oracle JDBC Driver DataSource

* JNDI name: 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

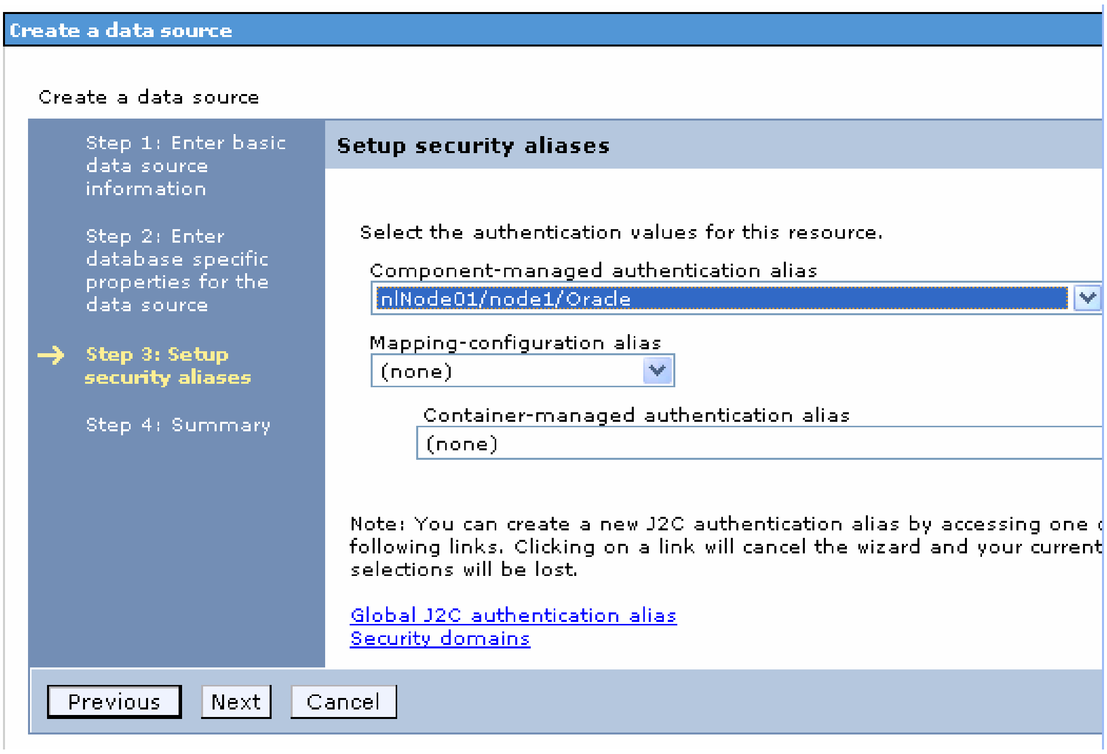
Set these database-specific properties, which are required by the database vendor JDBC driver to support the connections that are managed through the datasource.

Name	Value
* URL	jdbc:oracle:thin@9.181.84.13

* Data store helper class name: Oracle10g data store helper

☑ Use this data source in container managed persistence (CMP)

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

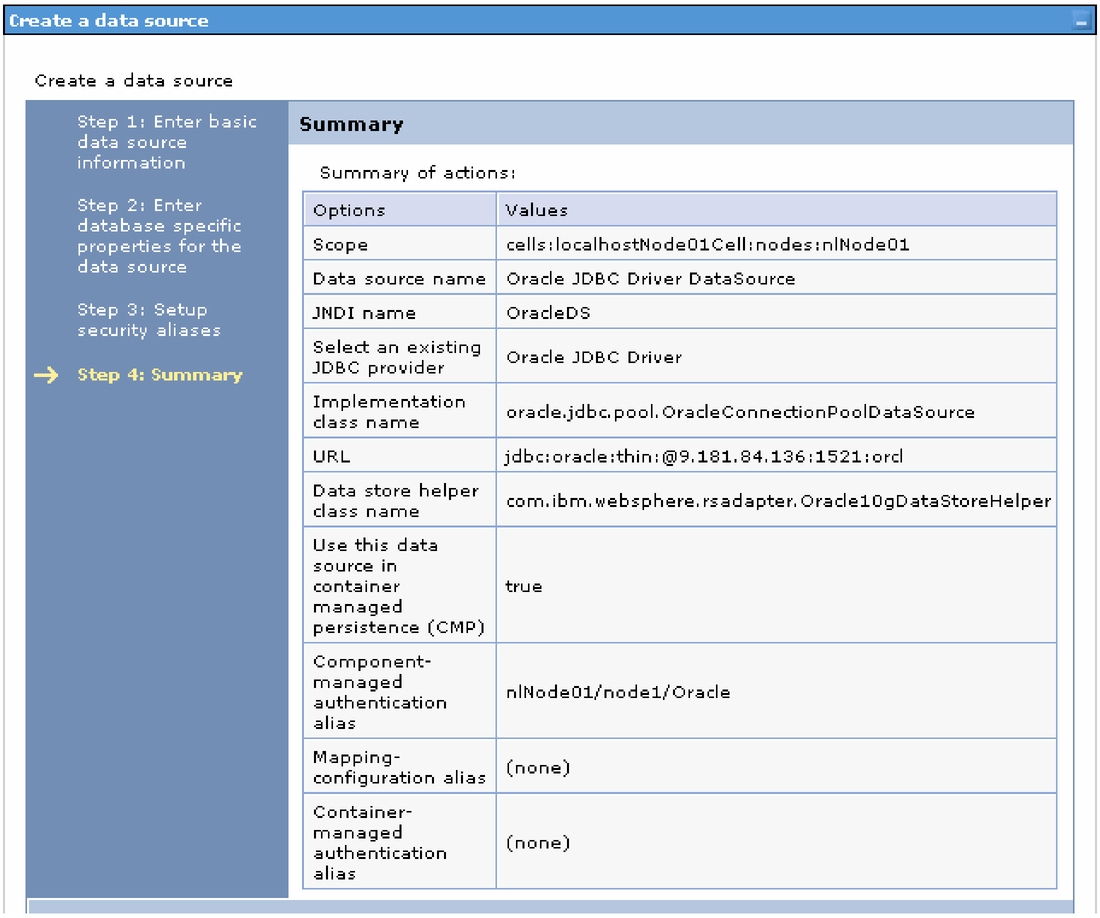
Mapping-configuration alias: (none)

Container-managed authentication alias: (none)

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

Global J2C authentication alias
Security domains

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



Create a data source

Create a data source

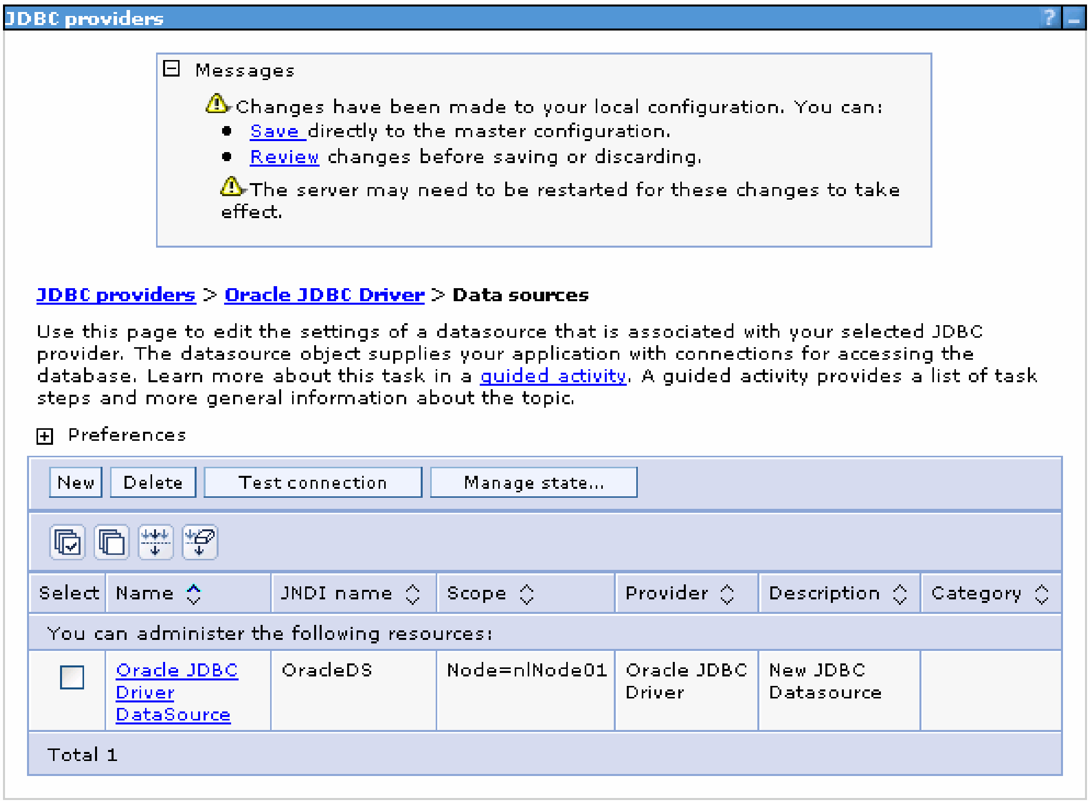
Step 1: Enter basic data source information
Step 2: Enter database specific properties for the data source
Step 3: Setup security aliases
Step 4: Summary

Summary

Summary of actions:

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:orcl
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/node1/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.

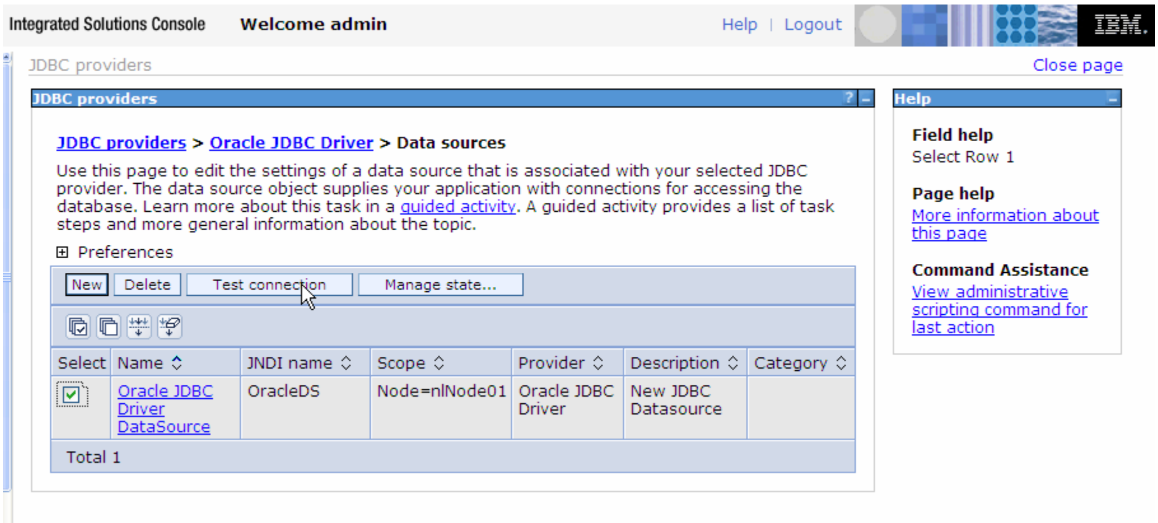
JDBC providers > Oracle JDBC Driver > Data sources

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

Select	Name	JNDI name	Scope	Provider	Description	Category
You can administer the following resources:						
□	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**.



Integrated Solutions Console Welcome admin Help | Logout IBM.

JDBC providers Close page

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

Select	Name	JNDI name	Scope	Provider	Description	Category
☑	Oracle JDBC Driver DataSource	OracleDS	Node=nlNode01	Oracle JDBC Driver	New JDBC Datasource	
Total 1						

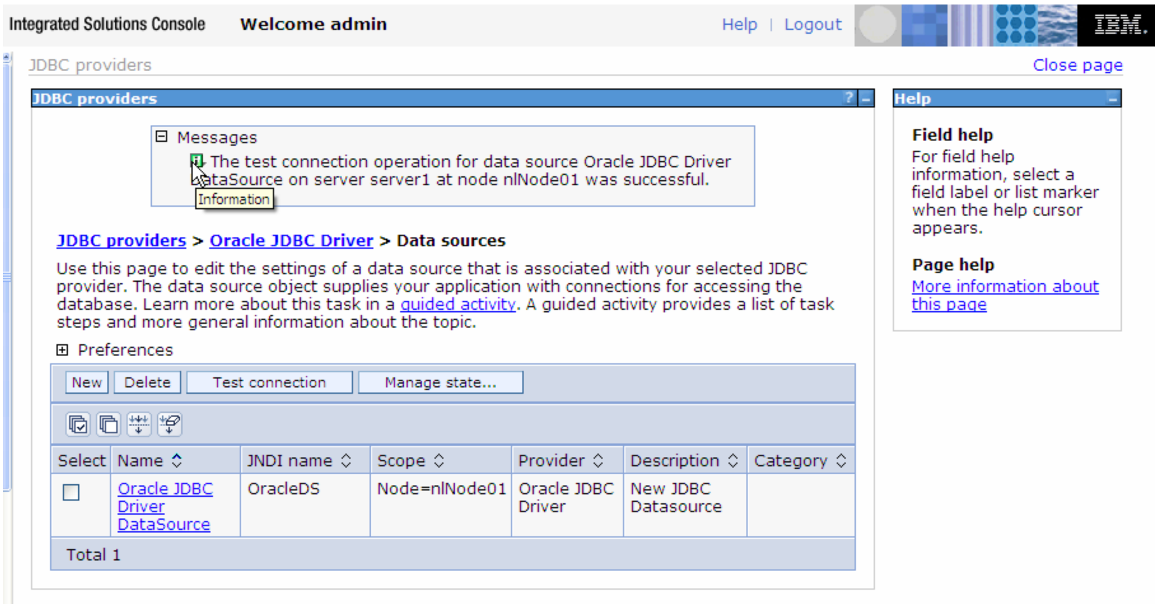
Help

Field help
Select Row 1

Page help
More information about this page

Command Assistance
View administrative scripting command for last action

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.



Integrated Solutions Console Welcome admin Help | Logout IBM.

JDBC providers Close page

JDBC providers

⊟ Messages

The test connection operation for data source Oracle JDBC Driver DataSource on server server1 at node nlNode01 was successful.

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

Select	Name	JNDI name	Scope	Provider	Description	Category
□	Oracle JDBC Driver DataSource	OracleDS	Node=nlNode01	Oracle JDBC Driver	New JDBC Datasource	
Total 1						

Help

Field help
For field help information, select a field label or list marker when the help cursor appears.

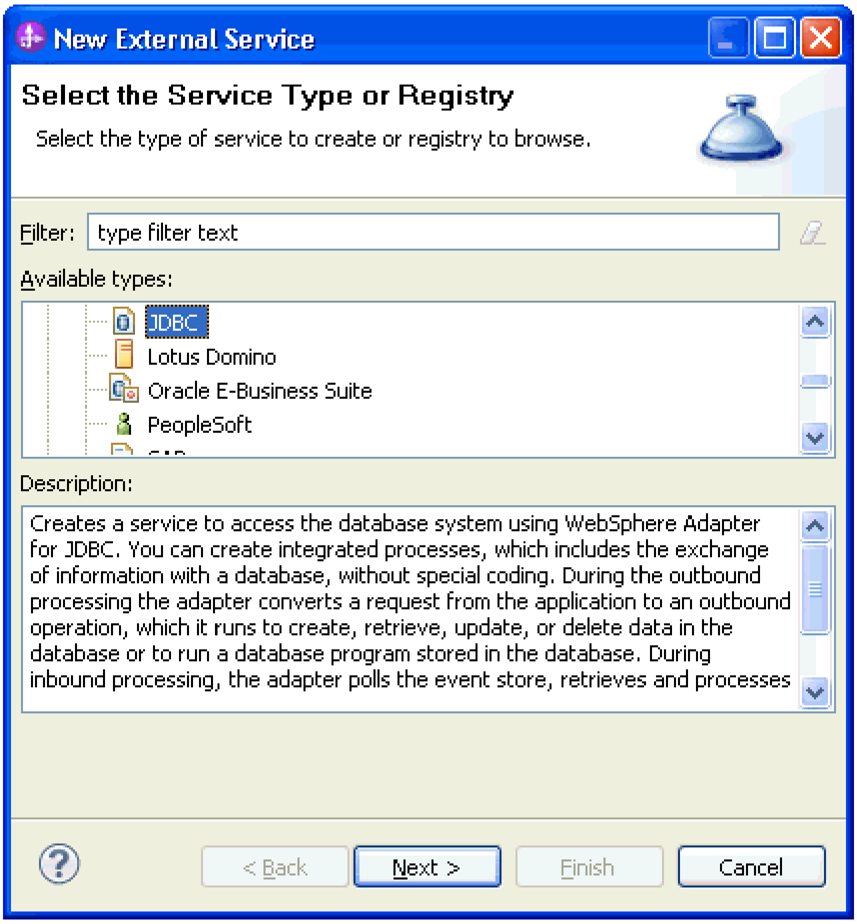
Page help
More information about this page

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 IBM Integration Designer 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**.



Select the Service Type or Registry

Select the type of service to create or registry to browse.

Filter: type filter text

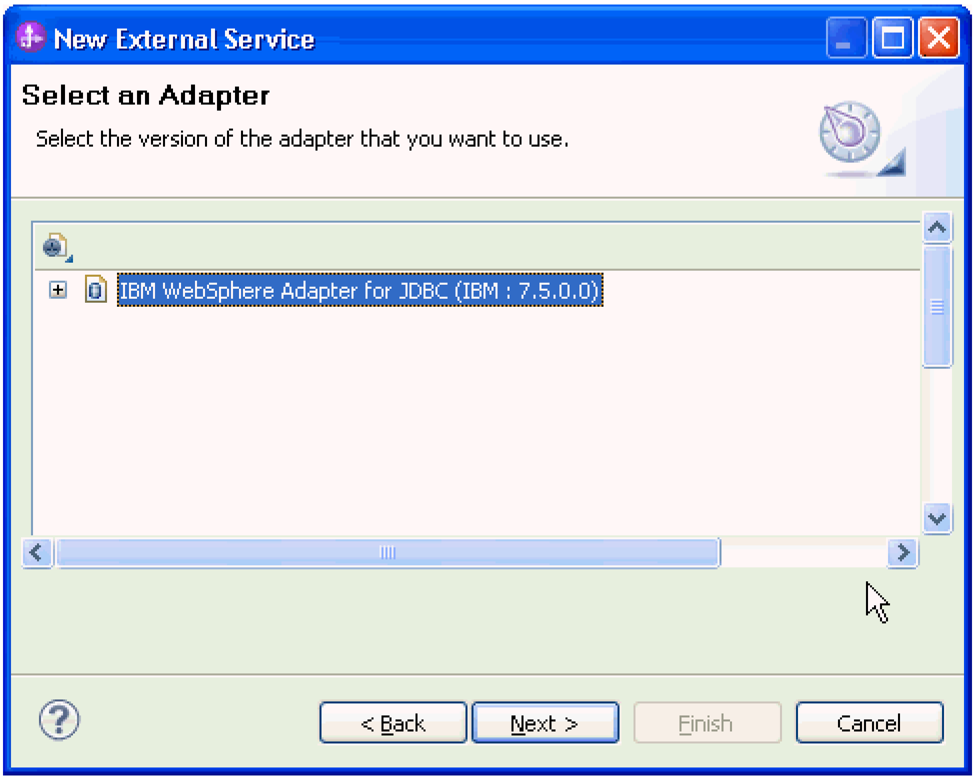
Available types:

- JDBC
- Lotus Domino
- Oracle E-Business Suite
- PeopleSoft

Description:

Creates a service to access the database system using WebSphere Adapter for JDBC. You can create integrated processes, which includes the exchange of information with a database, without special coding. During the outbound processing the adapter converts a request from the application to an outbound operation, which it runs to create, retrieve, update, or delete data in the database or to run a database program stored in the database. During inbound processing, the adapter polls the event store, retrieves and processes

4. Select **IBM WebSphere Adapter for JDBC** and click **Next**.



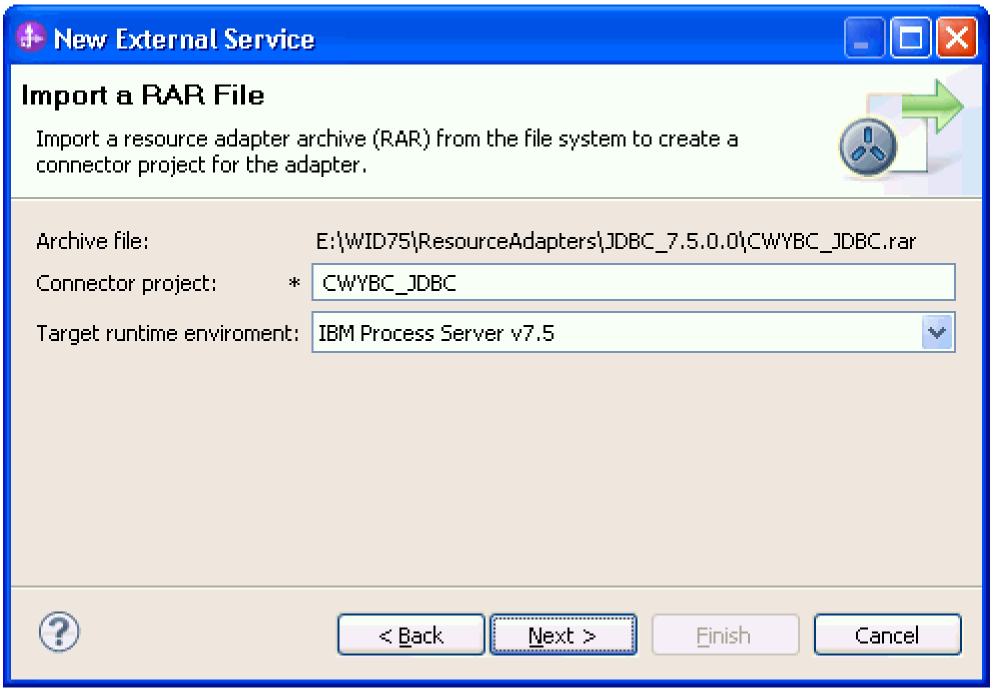
Select an Adapter

Select the version of the adapter that you want to use.

IBM WebSphere Adapter for JDBC (IBM : 7.5.0.0)

5. In the **Connector project** field enter **CWYBC_JDBC.**

6. In the **Target runtime environment** field, select the appropriate runtime and click **Next**.



Import a RAR File

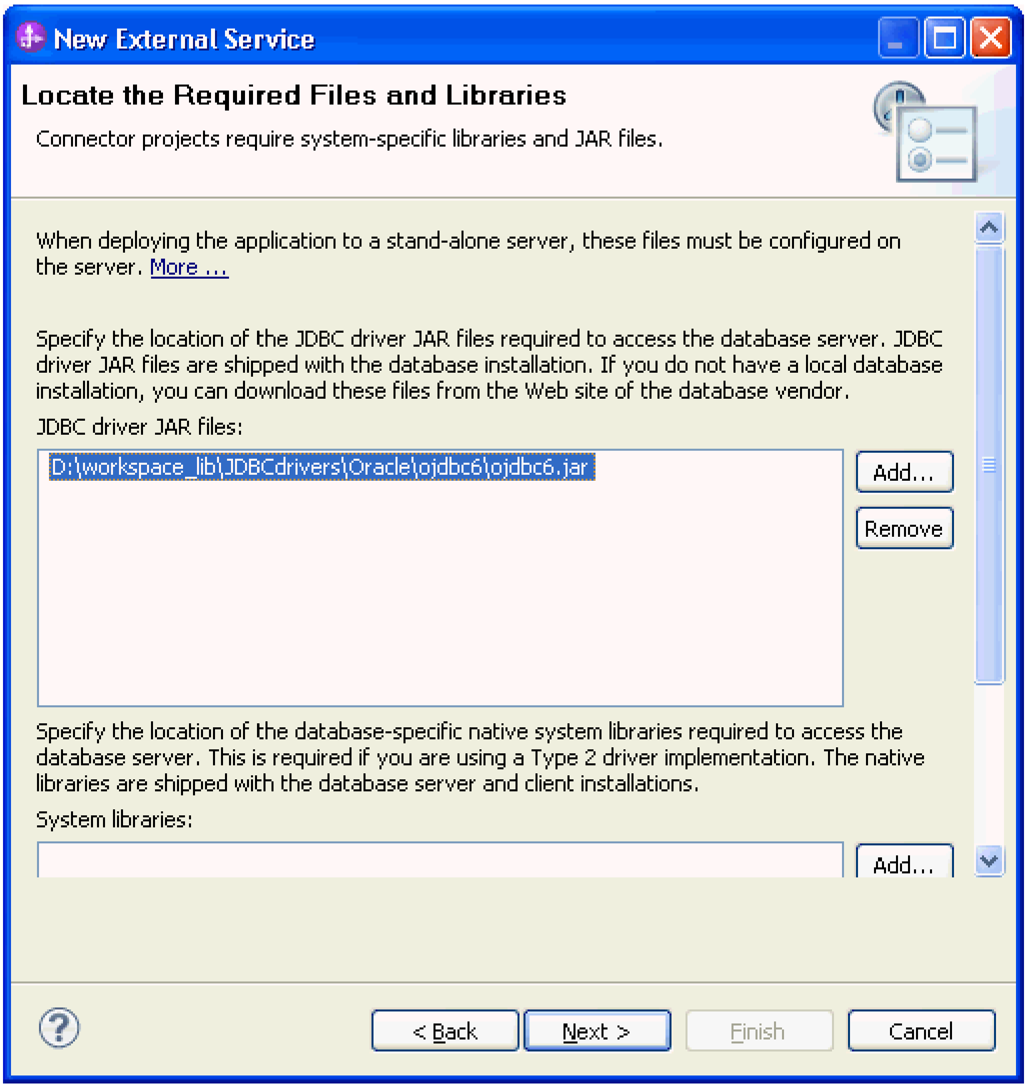
Import a resource adapter archive (RAR) from the file system to create a connector project for the adapter.

Archive file: E:\WID75\ResourceAdapters\JDBC_7.5.0.0\CWYBC_JDBC.rar

Connector project: * CWYBC_JDBC

Target runtime enviroment: IBM Process Server v7.5

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**.



New External Service

Locate the Required Files and Libraries

Connector projects require system-specific libraries and JAR files.

When deploying the application to a stand-alone server, these files must be configured on the server. More ...

Specify the location of the JDBC driver JAR files required to access the database server. JDBC driver JAR files are shipped with the database installation. If you do not have a local database installation, you can download these files from the Web site of the database vendor.

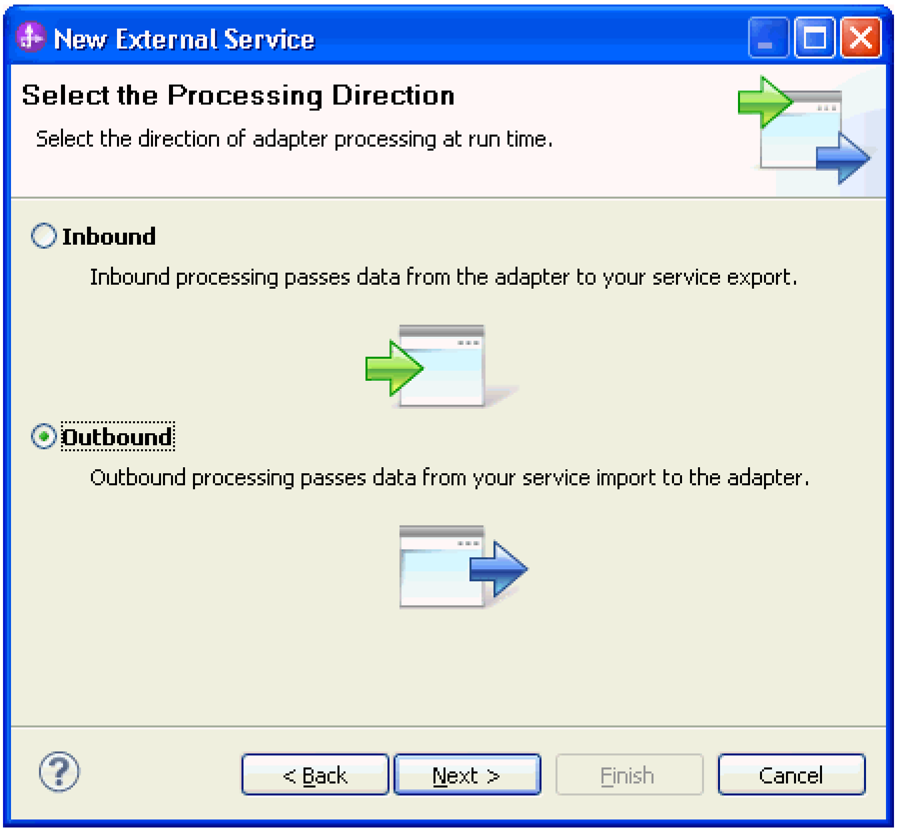
JDBC driver JAR files:

D:\workspace_lib\JDBCdrivers\Oracle\ojdbc6\ojdbc6.jar

Specify the location of the database-specific native system libraries required to access the database server. This is required if you are using a Type 2 driver implementation. The native libraries are shipped with the database server and client installations.

System libraries:

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



New External Service

Select the Processing Direction

Select the direction of adapter processing at run time.

○ Inbound

Inbound processing passes data from the adapter to your service export.

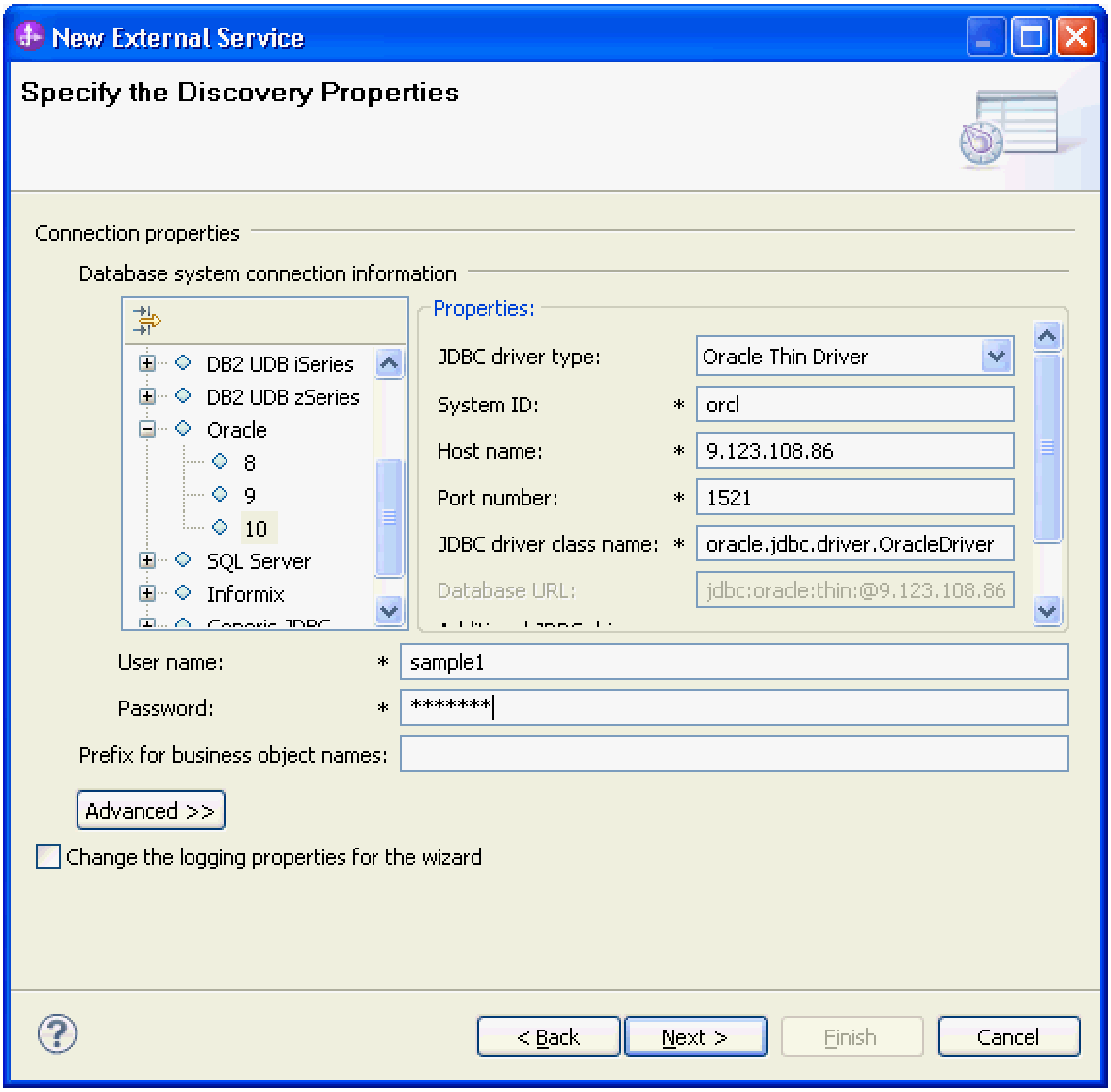
● Outbound

Outbound processing passes data from your service import to the adapter.

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**.



New External Service

Specify the Discovery Properties

Connection properties

Database system connection information

Properties:

JDBC driver type: Oracle Thin Driver

System ID: * orcl

Host name: * 9.123.108.86

Port number: * 1521

JDBC driver class name: * oracle.jdbc.driver.OracleDriver

Database URL: jdbc:oracle:thin:@9.123.108.86

User name: * sample1

Password: * *******

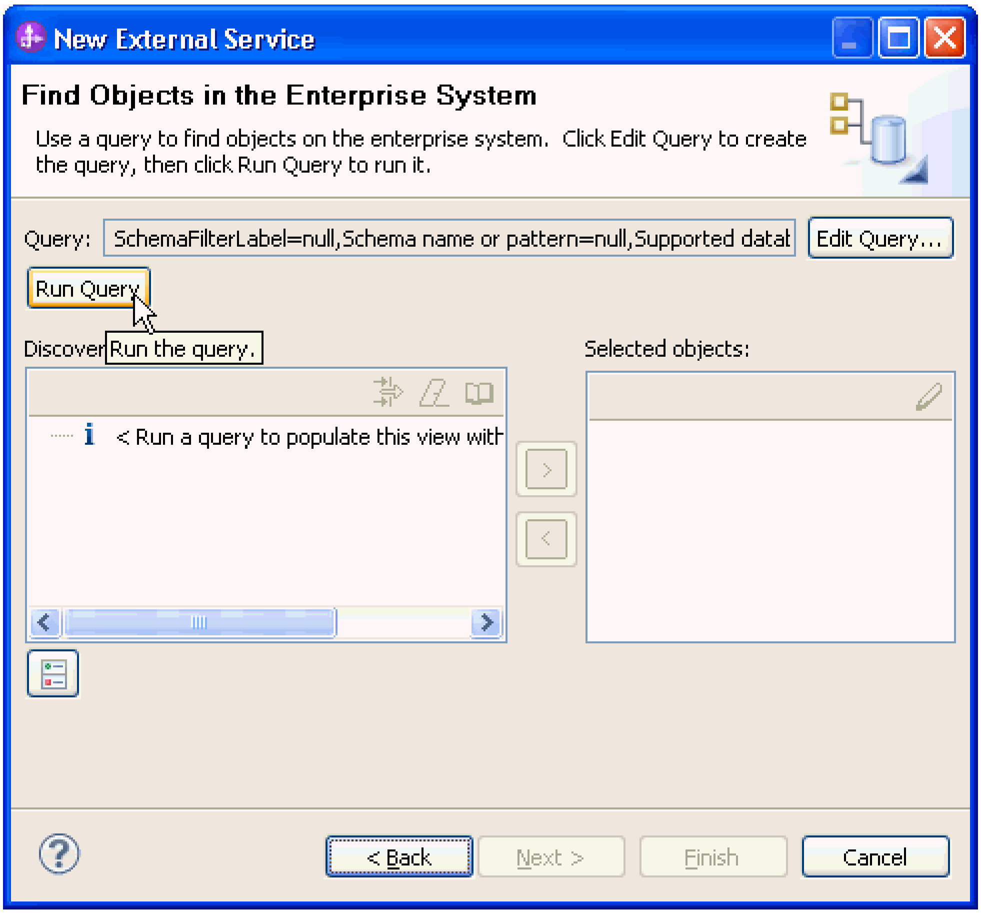
Prefix for business object names:

□ Change the logging properties for the wizard

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**.



New External Service

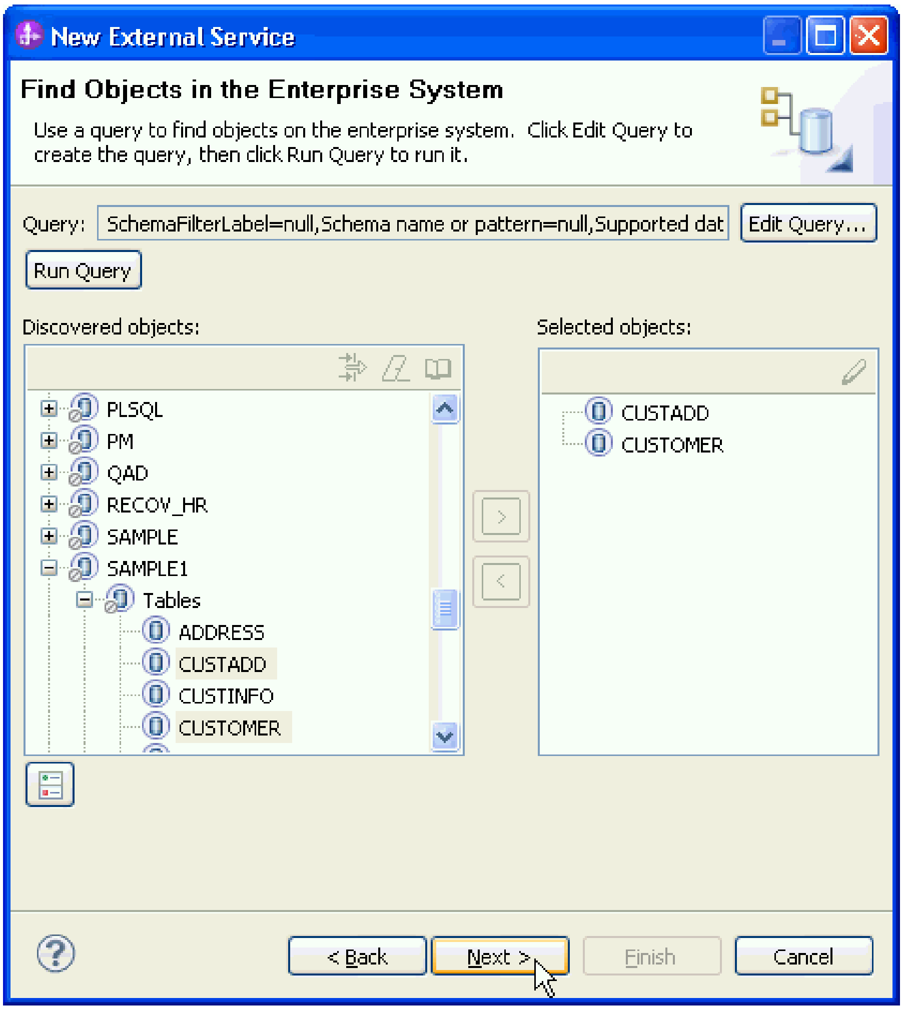
Find Objects in the Enterprise System

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 datat

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 > .



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 dat

Discovered objects:

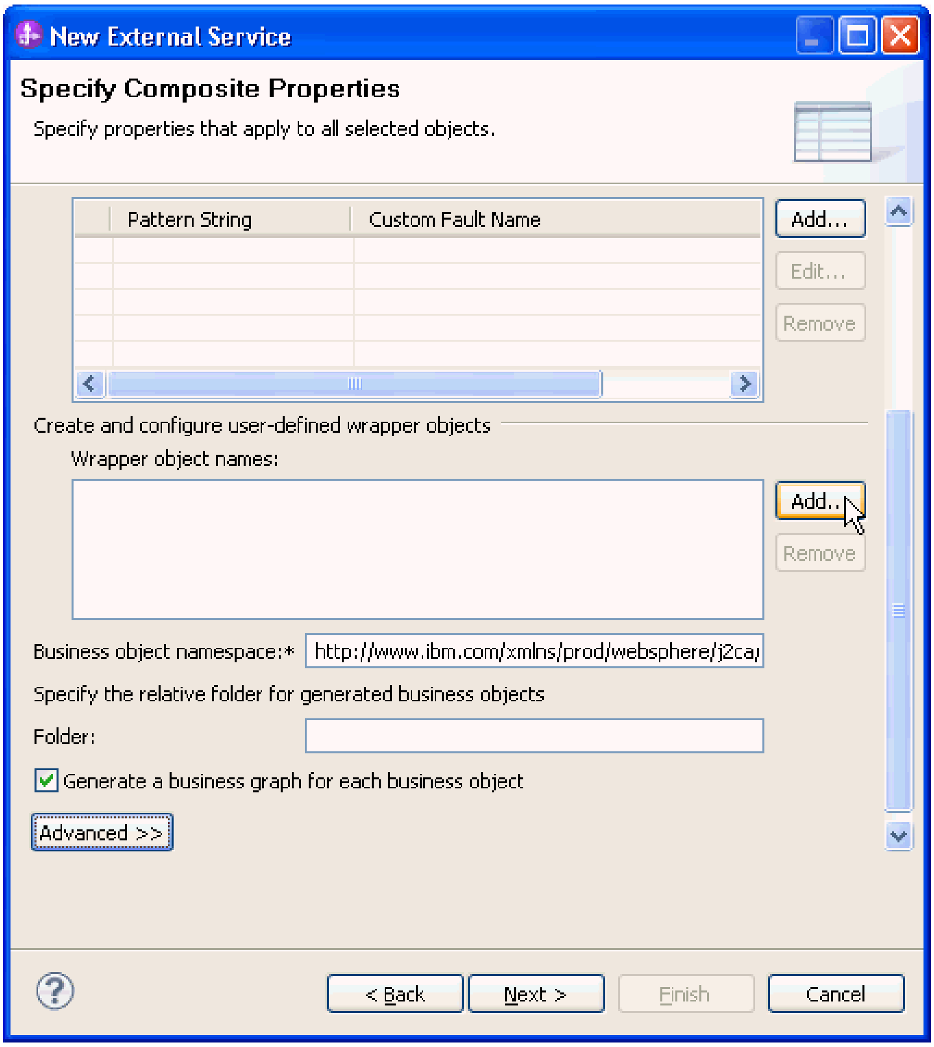
- PLSQL
- PM
- QAD
- RECOV_HR
- SAMPLE
- SAMPLE1
  - Tables
    - ADDRESS
    - CUSTADD
    - CUSTINFO
    - CUSTOMER

Selected objects:

- CUSTADD
- CUSTOMER

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.**



New External Service

Specify Composite Properties

Specify properties that apply to all selected objects.

Pattern String	Custom Fault Name

Create and configure user-defined wrapper objects

Wrapper object names:

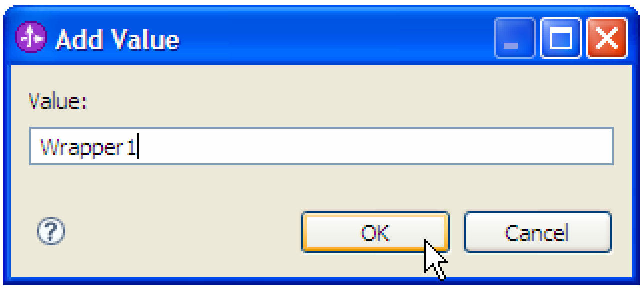
Business object namespace:* http://www.ibm.com/xmlns/prod/websphere/j2ca/

Specify the relative folder for generated business objects

Folder:

☑ Generate a business graph for each business object

6. In the Add Value window, specify the name for the wrapper. Enter **Wrapper1** and click **OK**.

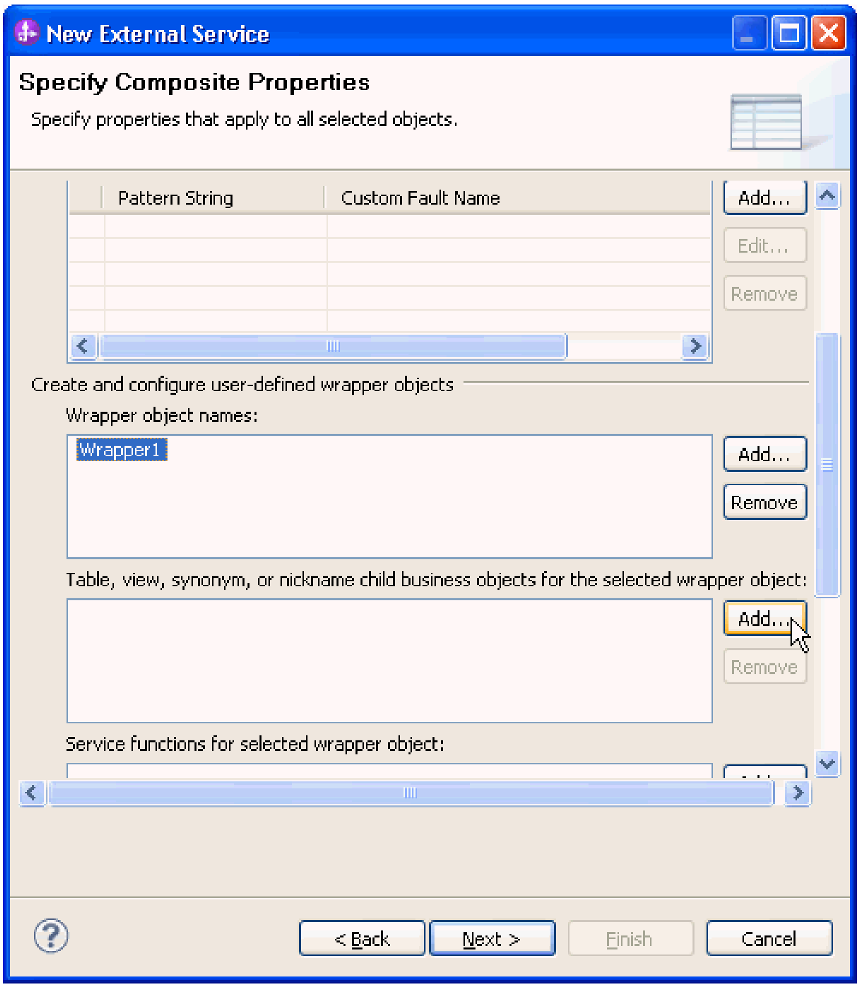


Add Value

Value: Wrapper1

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.



New External Service

Specify Composite Properties

Specify properties that apply to all selected objects.

Pattern String	Custom Fault Name

Create and configure user-defined wrapper objects

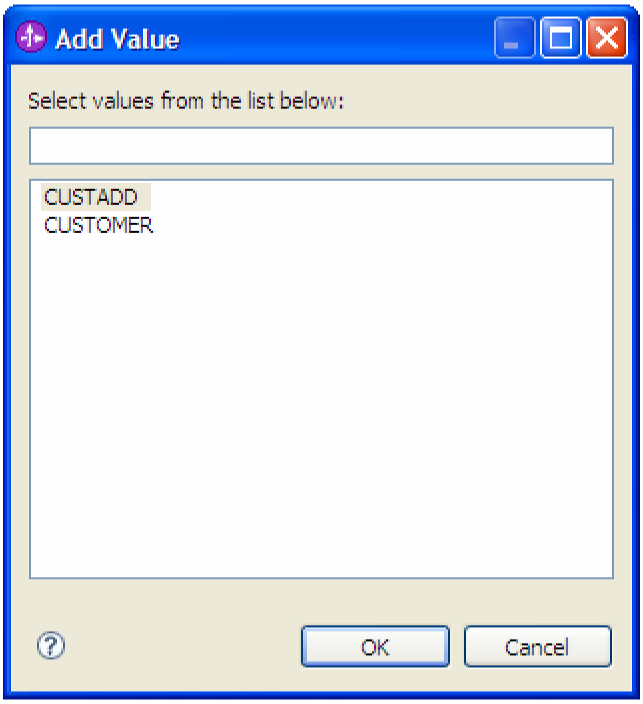
Wrapper object names:

Wrapper1

Table, view, synonym, or nickname child business objects for the selected wrapper object:

Service functions for selected wrapper object:

8. In the Add Value window, select CUSTADD and CUSTOMER tables and click **OK**.



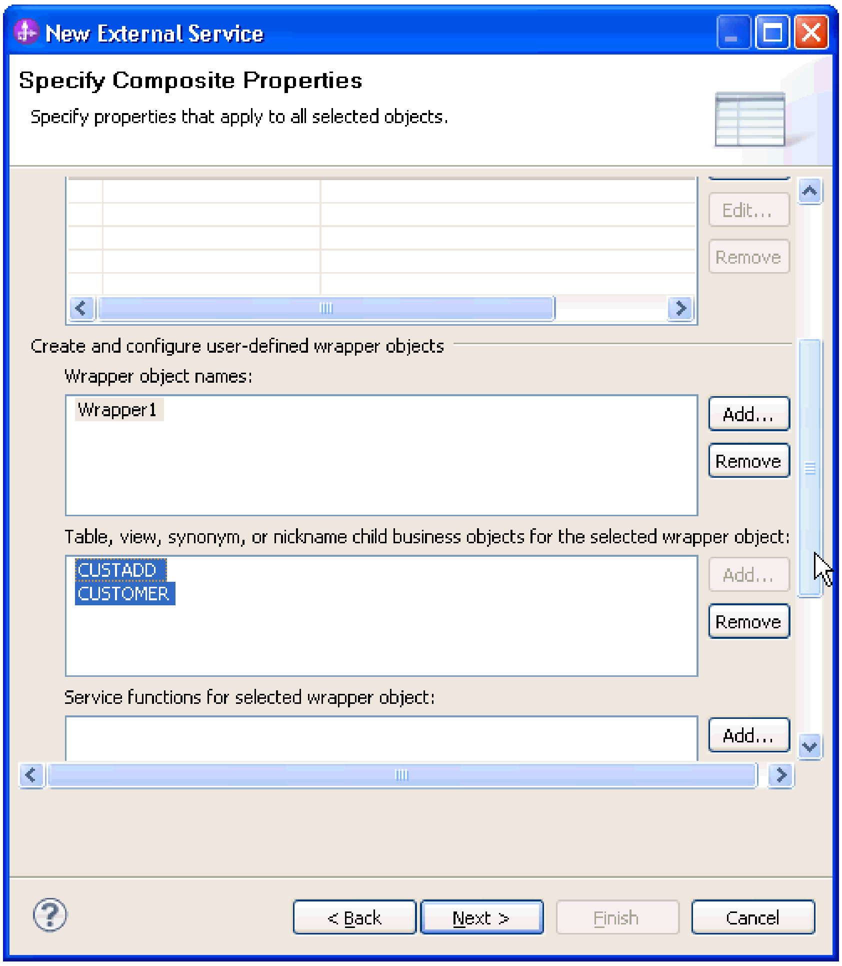
Add Value

Select values from the list below:

CUSTADD
CUSTOMER

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.



New External Service

Specify Composite Properties

Specify properties that apply to all selected objects.

Create and configure user-defined wrapper objects

Wrapper object names:

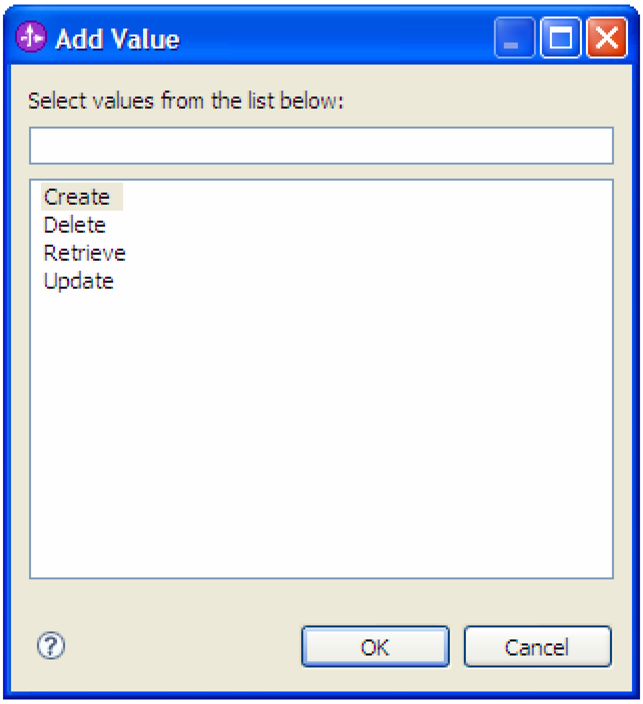
Wrapper1

Table, view, synonym, or nickname child business objects for the selected wrapper object:

CUSTADD
CUSTOMER

Service functions for selected wrapper object:

10. In the Add Value window, select the Create and click **OK**.



Add Value

Select values from the list below:

Create
Delete
Retrieve
Update

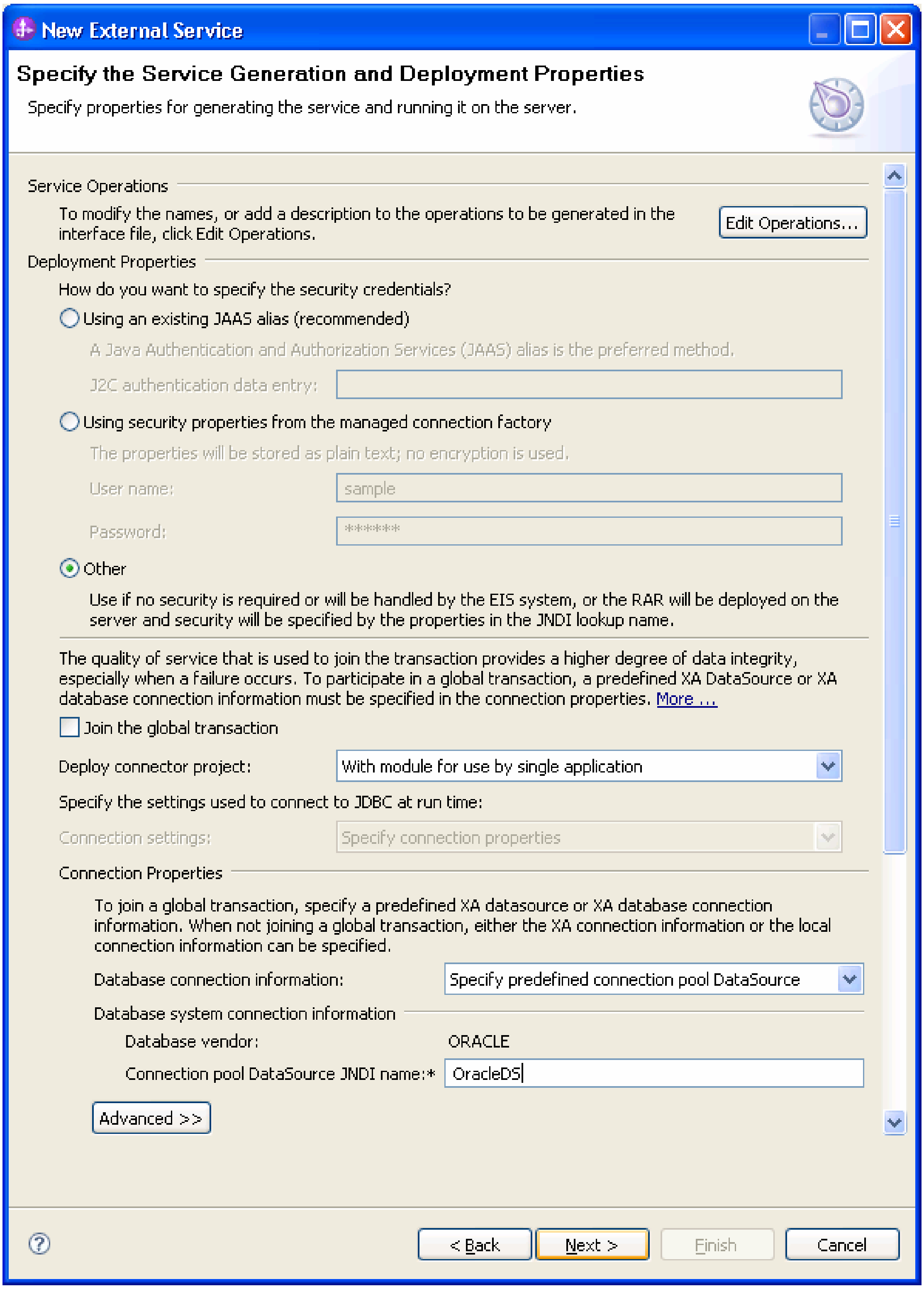
The selected service operation is added into the **Service functions for the selected wrapper object** area.

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**.



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 managed connection factory

The properties will be stored as plain text; no encryption is used.

User name: sample

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: With module for use by single application

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

Connection settings: Specify connection properties

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.

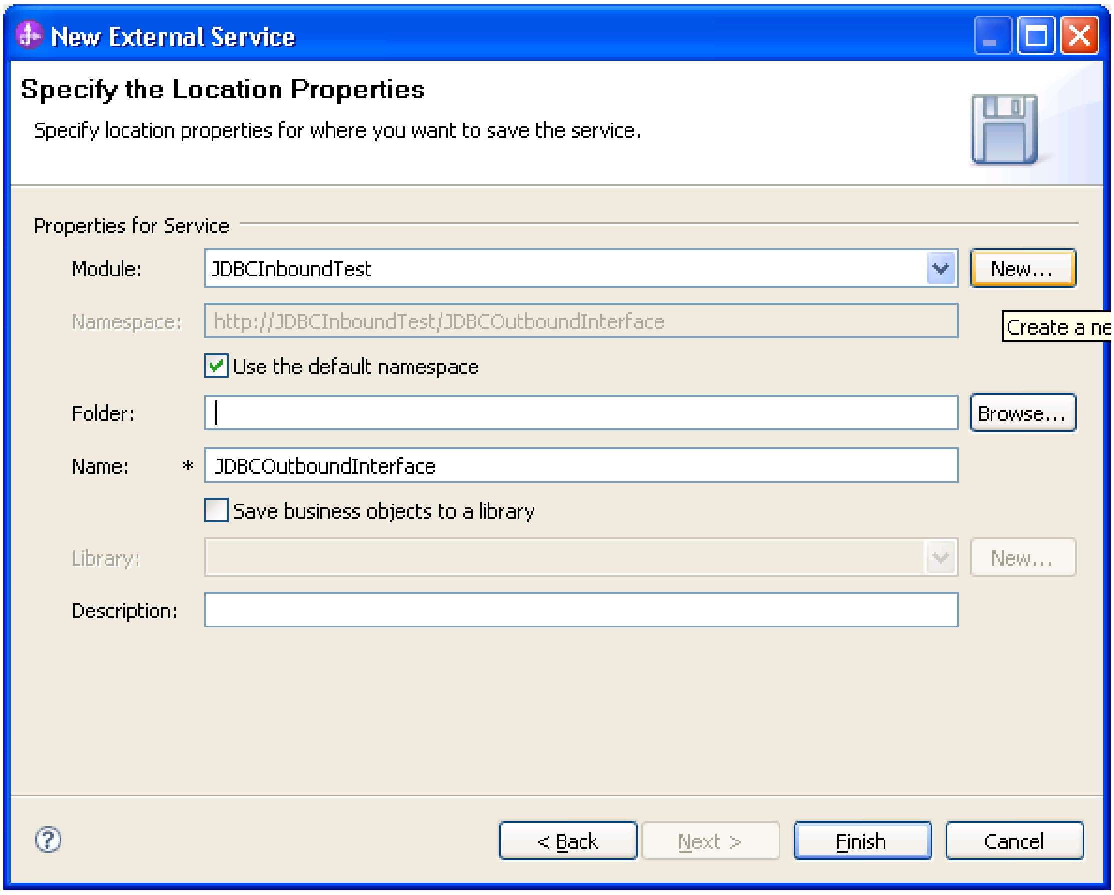
Database connection information: Specify predefined connection pool DataSource

Database system connection information

Database vendor: ORACLE

Connection pool DataSource JNDI name:* OracleDS

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



New External Service

Specify the Location Properties

Specify location properties for where you want to save the service.

Properties for Service

Module: JDBCInboundTest

Namespace: http://JDBCInboundTest/JDBCOutboundInterface

☑ Use the default namespace

Folder:

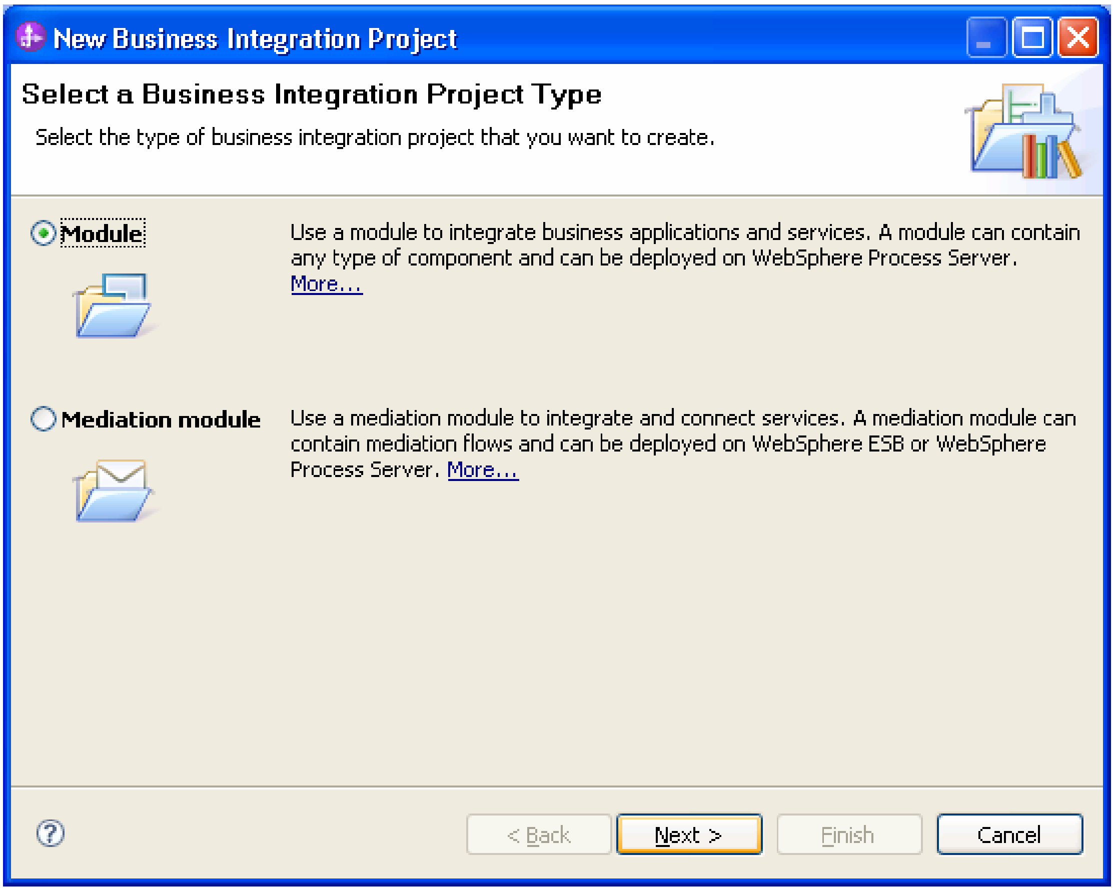
Name: * JDBCOutboundInterface

☐ Save business objects to a library

Library:

Description:

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



New Business Integration Project

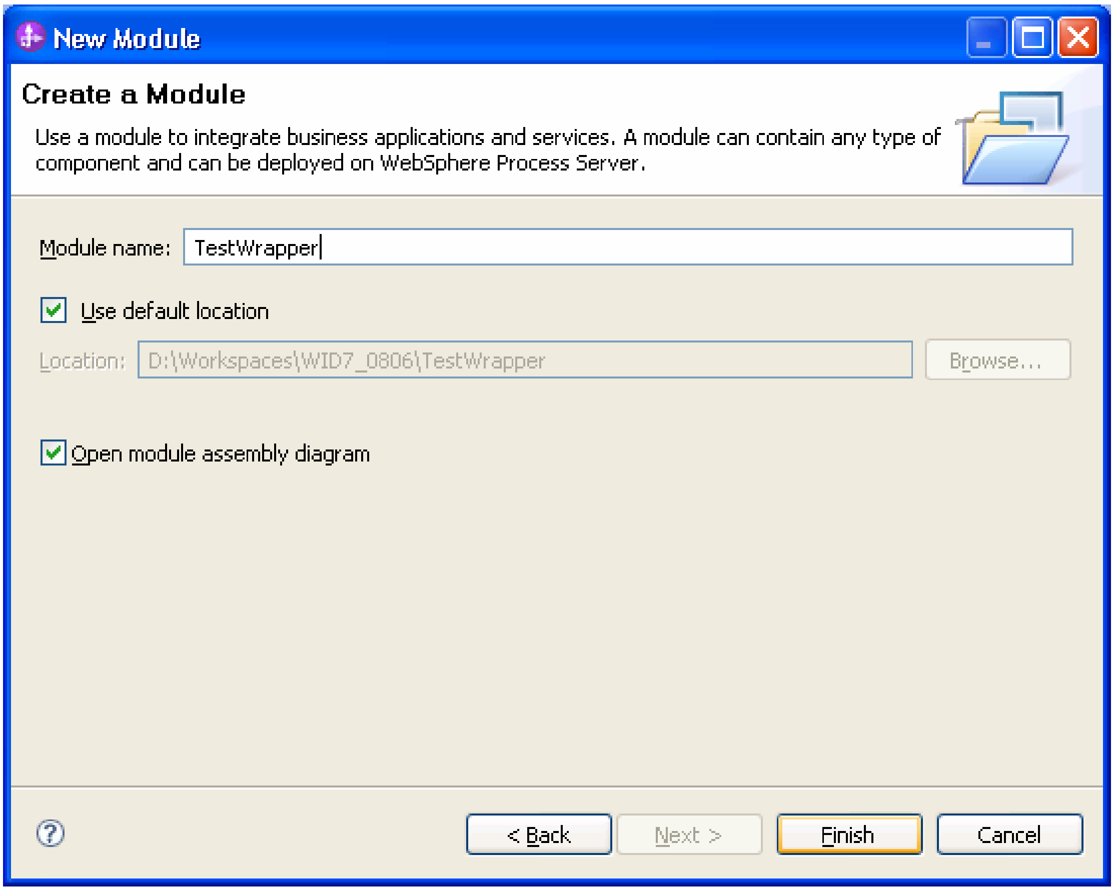
Select a Business Integration Project Type

Select the type of business integration project that you want to create.

● Module — Use a module to integrate business applications and services. A module can contain any type of component and can be deployed on WebSphere Process Server. More...

○ Mediation module — Use a mediation module to integrate and connect services. A mediation module can contain mediation flows and can be deployed on WebSphere ESB or WebSphere Process Server. More...

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



Create a Module

Use a module to integrate business applications and services. A module can contain any type of component and can be deployed on WebSphere Process Server.

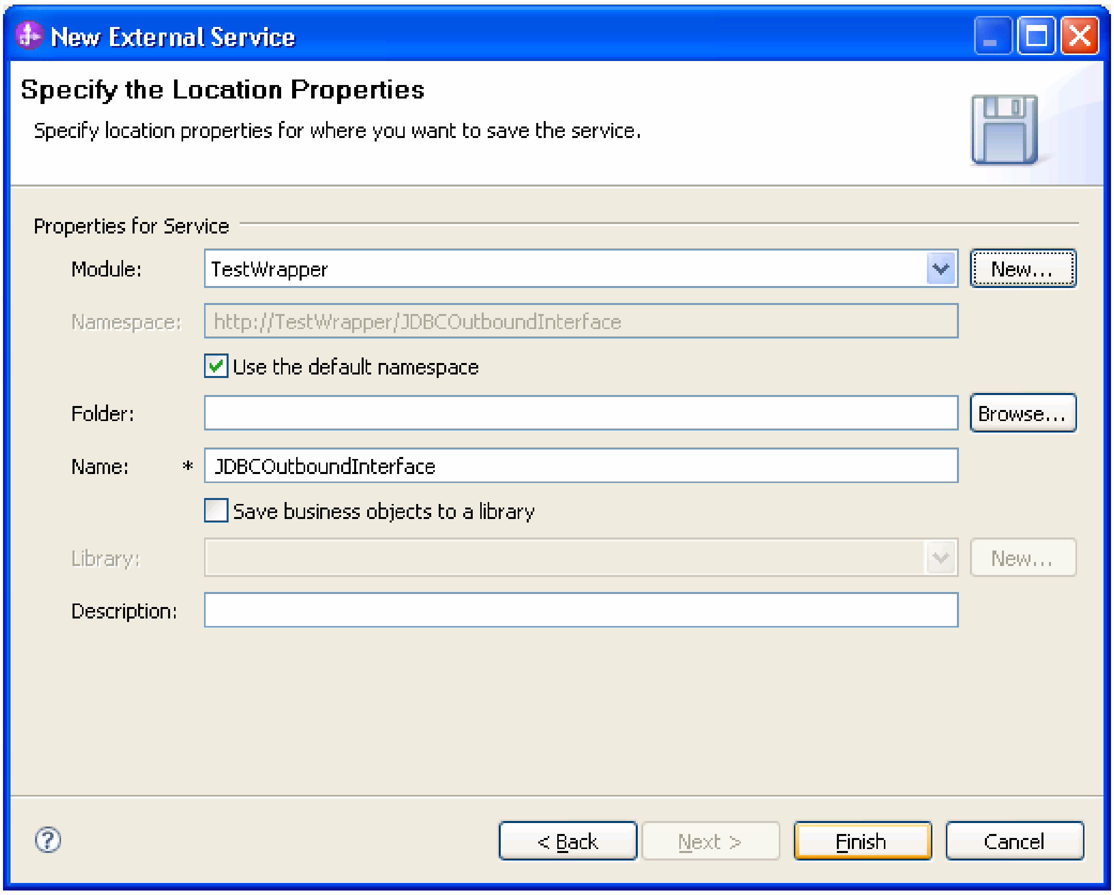
Module name: TestWrapper

☑ Use default location

Location: D:\Workspaces\WID7_0806\TestWrapper

☑ Open module assembly diagram

5. In the Specify the Location Properties window, accept the default values for all fields and click **Finish**.



New External Service

Specify the Location Properties

Specify location properties for where you want to save the service.

Properties for Service

Module: TestWrapper

Namespace: http://TestWrapper/JDBCOutboundInterface

☑ Use the default namespace

Folder:

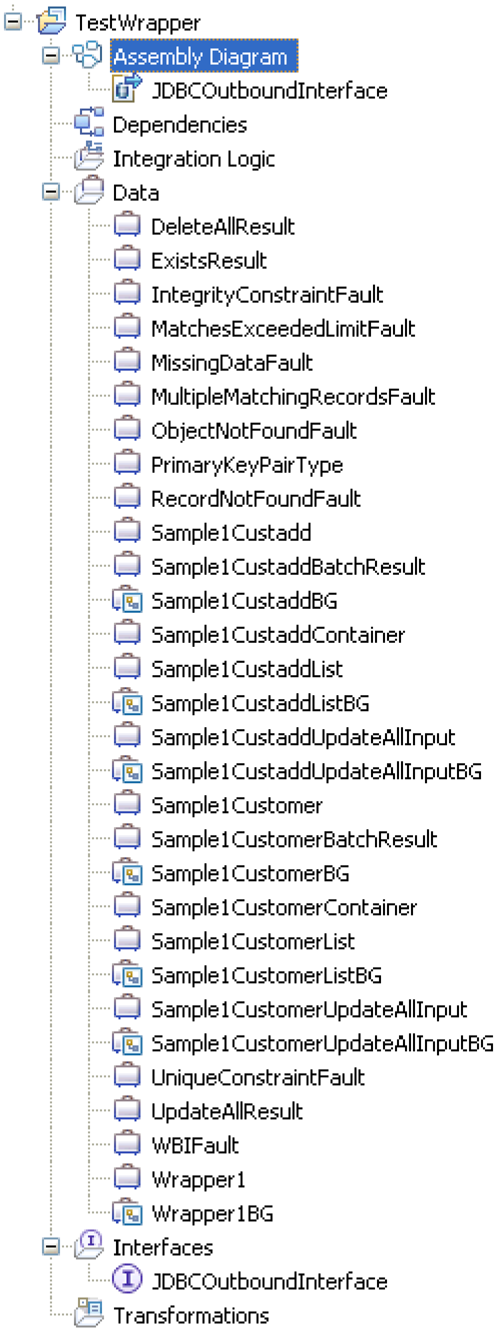
Name: * JDBCOutboundInterface

☐ Save business objects to a library

Library:

Description:

6. Open the Project Explorer and verify business objects are created correctly.



- TestWrapper
  - Assembly Diagram
    - JDBCOutboundInterface
  - Dependencies
  - Integration Logic
  - Data
    - DeleteAllResult
    - ExistsResult
    - IntegrityConstraintFault
    - MatchesExceededLimitFault
    - MissingDataFault
    - MultipleMatchingRecordsFault
    - ObjectNotFoundFault
    - PrimaryKeyPairType
    - RecordNotFoundFault
    - Sample1Custadd
    - Sample1CustaddBatchResult
    - Sample1CustaddBG
    - Sample1CustaddContainer
    - Sample1CustaddList
    - Sample1CustaddListBG
    - Sample1CustaddUpdateAllInput
    - Sample1CustaddUpdateAllInputBG
    - Sample1Customer
    - Sample1CustomerBatchResult
    - Sample1CustomerBG
    - Sample1CustomerContainer
    - Sample1CustomerList
    - Sample1CustomerListBG
    - Sample1CustomerUpdateAllInput
    - Sample1CustomerUpdateAllInputBG
    - UniqueConstraintFault
    - UpdateAllResult
    - WBIFault
    - Wrapper1
    - Wrapper1BG
  - Interfaces
    - JDBCOutboundInterface
  - Transformations

Deploy the module to the test environment

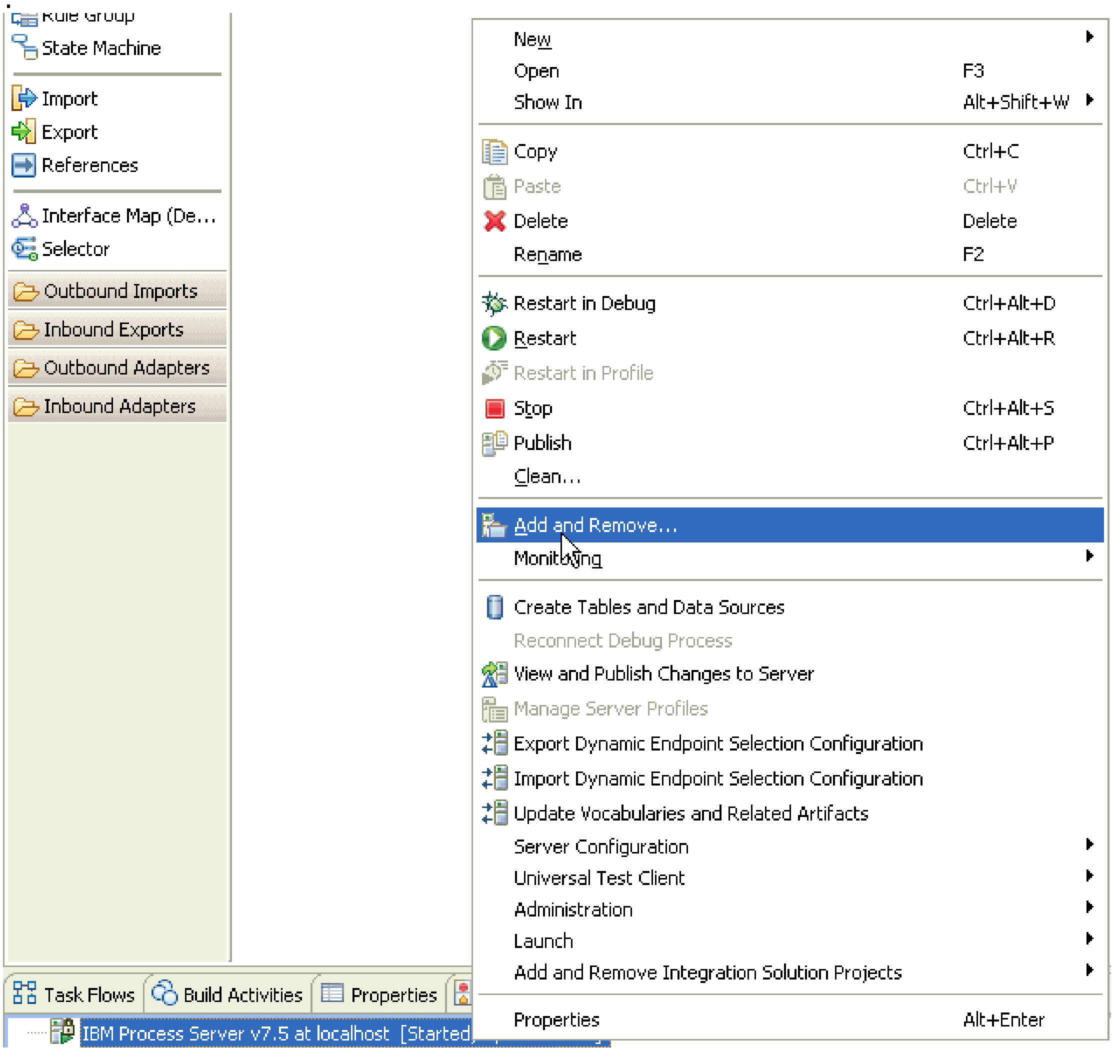
After running the external service wizard, you will have an SCA module that contains an EIS import. You must install this SCA module in the IBM Integration Designer integration test client. To do this, you must add the SCA module you created earlier to the server using the **Servers** view in IBM Integration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

Steps for adding the SCA module to the server:

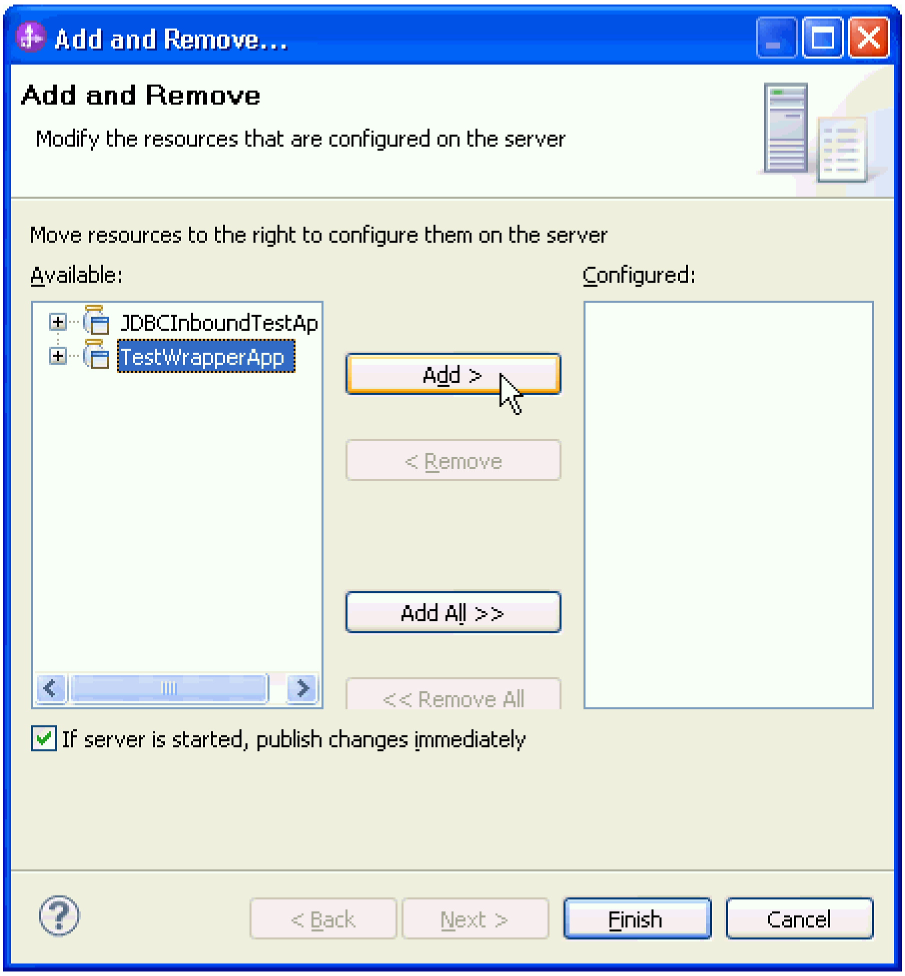
1. In IBM Integration Designer, switch to the **Servers** view by selecting from the toolbar **Window** > **Show View** > **Servers**.

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**.



The Add and Remove Projects window lists the available projects in the IBM Integration Designer workspace.

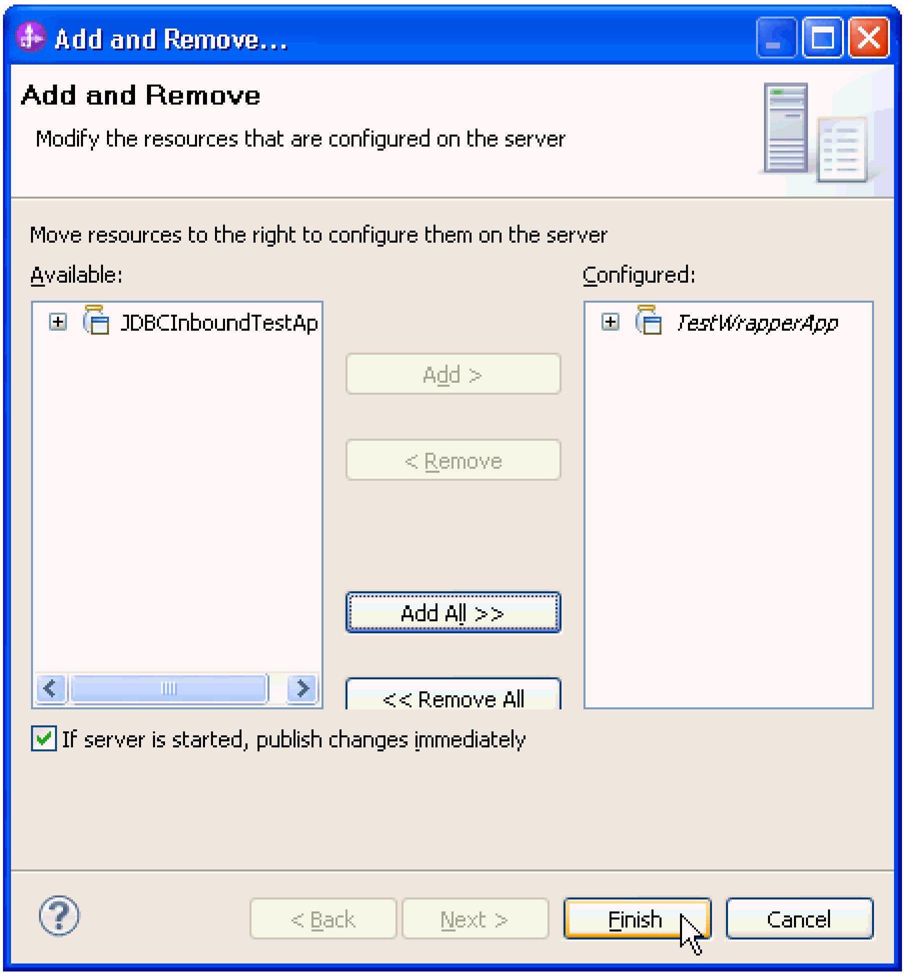


Modify the resources that are configured on the server

Move resources to the right to configure them on the server

☑ If server is started, publish changes immediately

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

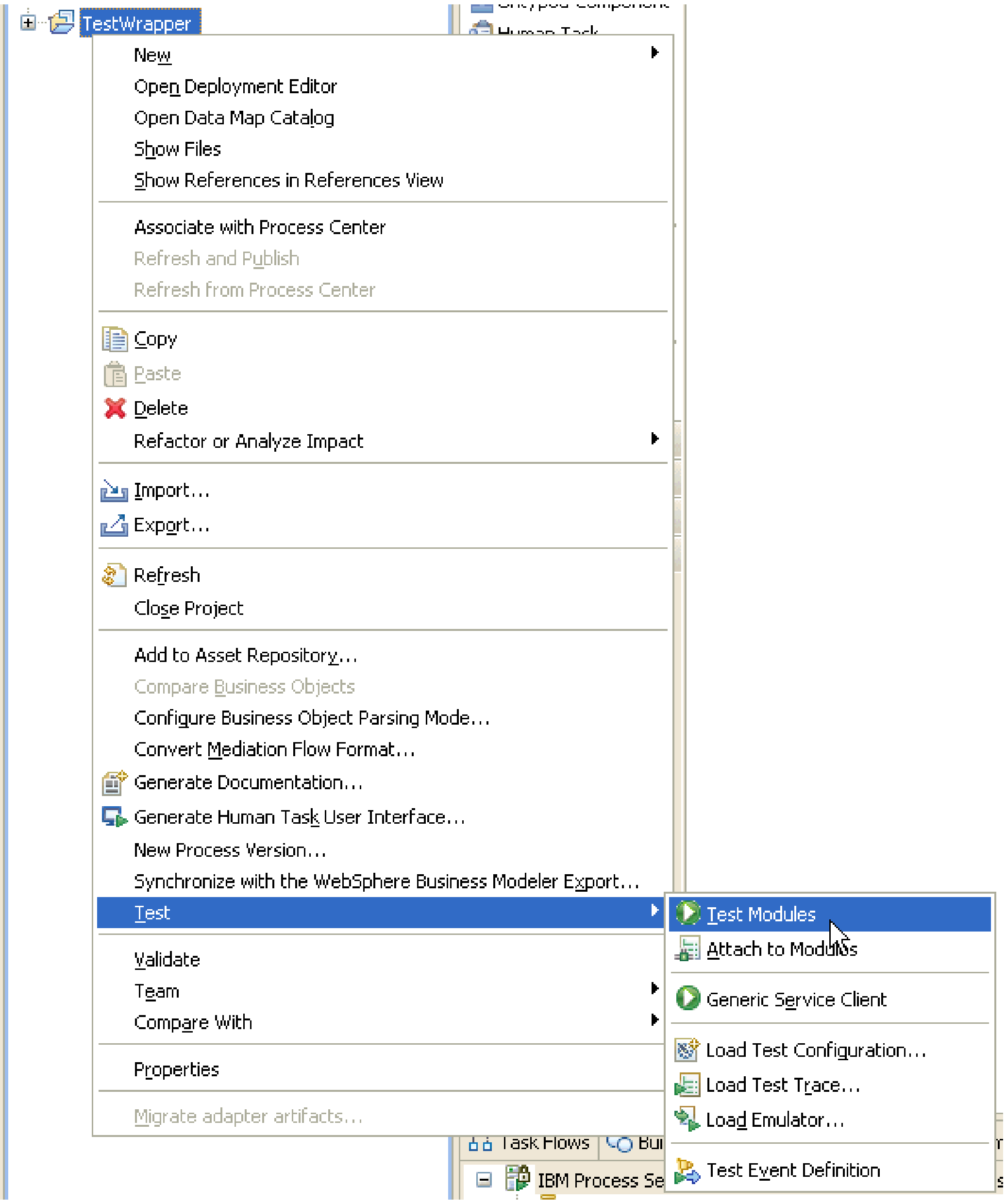


Add and Remove...
Add and Remove
Modify the resources that are configured on the server
Move resources to the right to configure them on the server
Available:
JDBCInboundTestAp
Configured:
TestWrapperApp
Add >
< Remove
Add All >>
<< Remove All
☑ If server is started, publish changes immediately
< Back
Next >
Finish
Cancel

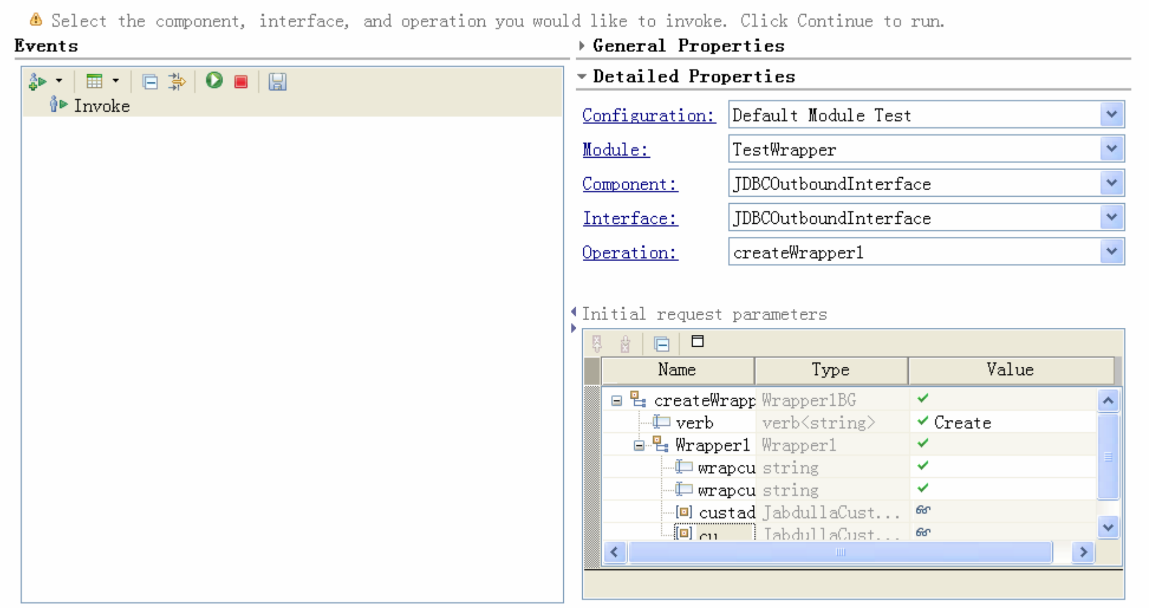
Test the assembled adapter application

Test the assembled adapter application using the IBM Integration Designer integration test client.

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


TestWrapper
New
Open Deployment Editor
Open Data Map Catalog
Show Files
Show References in References View
Associate with Process Center
Refresh and Publish
Refresh from Process Center
Copy
Paste
Delete
Refactor or Analyze Impact
Import...
Export...
Refresh
Close Project
Add to Asset Repository...
Compare Business Objects
Configure Business Object Parsing Mode...
Convert Mediation Flow Format...
Generate Documentation...
Generate Human Task User Interface...
New Process Version...
Synchronize with the WebSphere Business Modeler Export...
Test
Validate
Team
Compare With
Properties
Migrate adapter artifacts...
Test Modules
Attach to Modules
Generic Service Client
Load Test Configuration...
Load Test Trace...
Load Emulator...
Test Event Definition
Task Flows
IBM Process Se

2. Select **createWrapper1** from the **Operation** list and set "Create" as **verb**. Right-click **custaddobj** and select **Add Elements.**



Select the component, interface, and operation you would like to invoke. Click Continue to run.

Events

Invoke

General Properties

Detailed Properties

Configuration: Default Module Test

Module: TestWrapper

Component: JDBCOutboundInterface

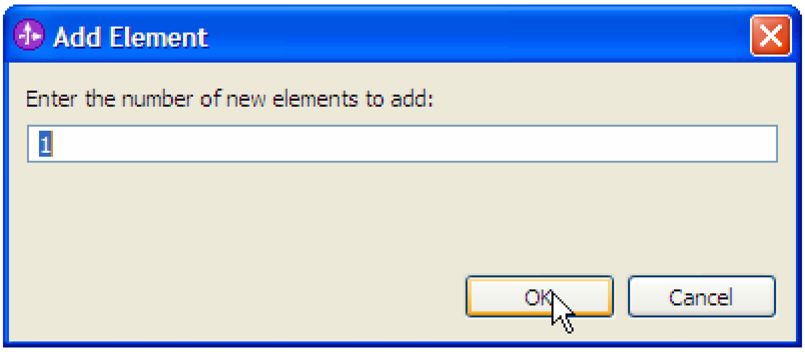
Interface: JDBCOutboundInterface

Operation: createWrapper1

Initial request parameters

Name	Type	Value
createWrapp	Wrapper1BG	✓
verb	verb<string>	✓ Create
Wrapper1	Wrapper1	✓
wrapcu	string	✓
wrapcu	string	✓
custad	JabdullaCust...	
cu	JabdullaCust...	

3. Enter 1 and click **OK.**

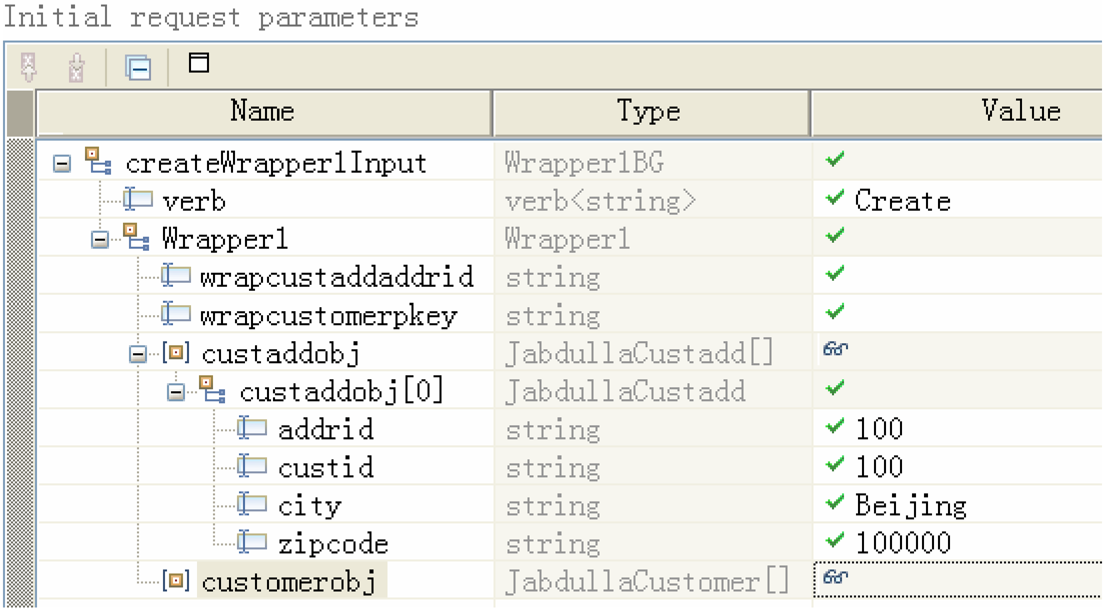


Add Element

Enter the number of new elements to add:

1

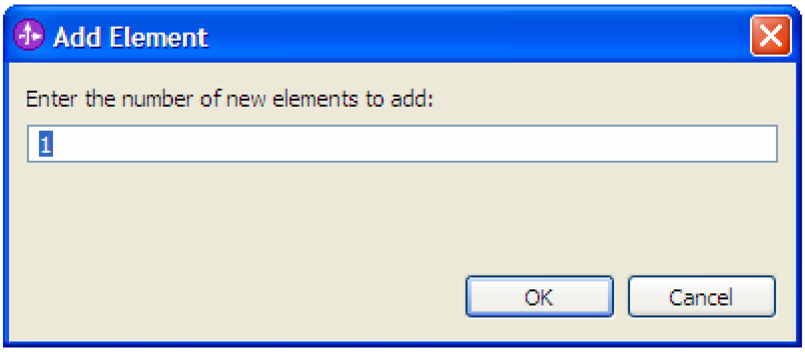
4. Enter the input values for custaddobj[0] as shown in the below figure.



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[]	

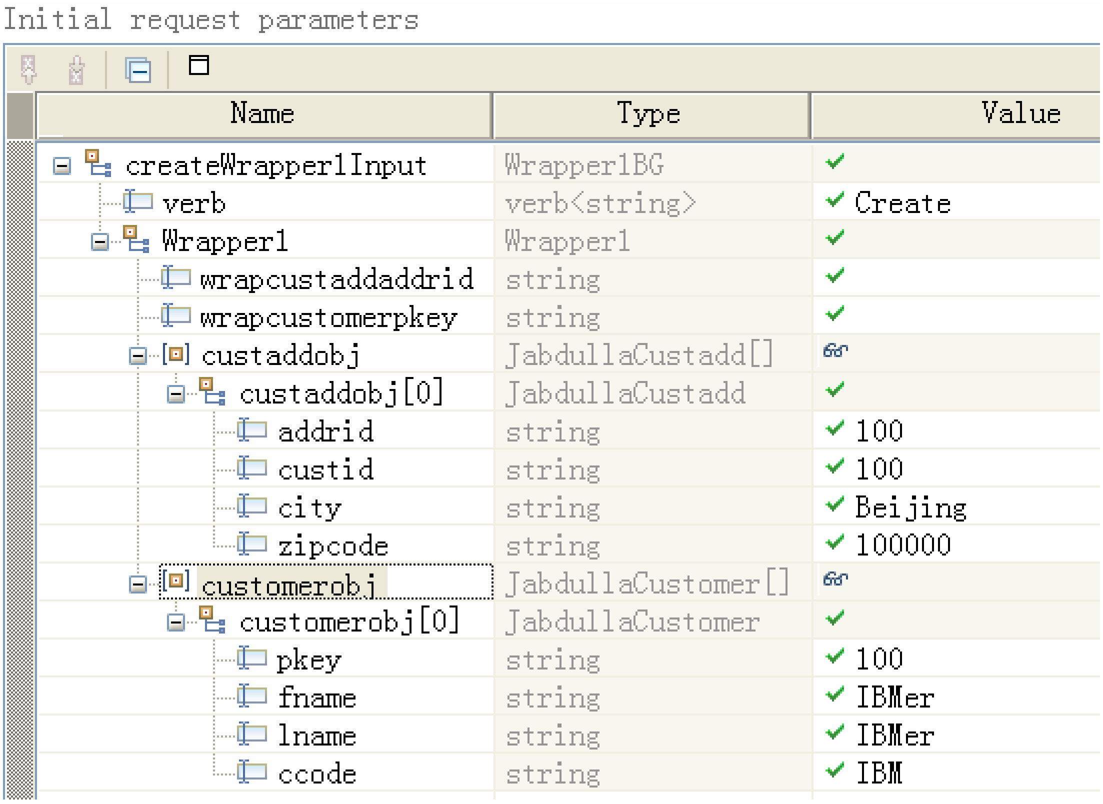
5. Now, right-click over customerobj and select **Add Elements** and enter 1 and click **OK**.



Enter the number of new elements to add:

1

6. Enter the input values for customerobj[0] as shown in the below figure.

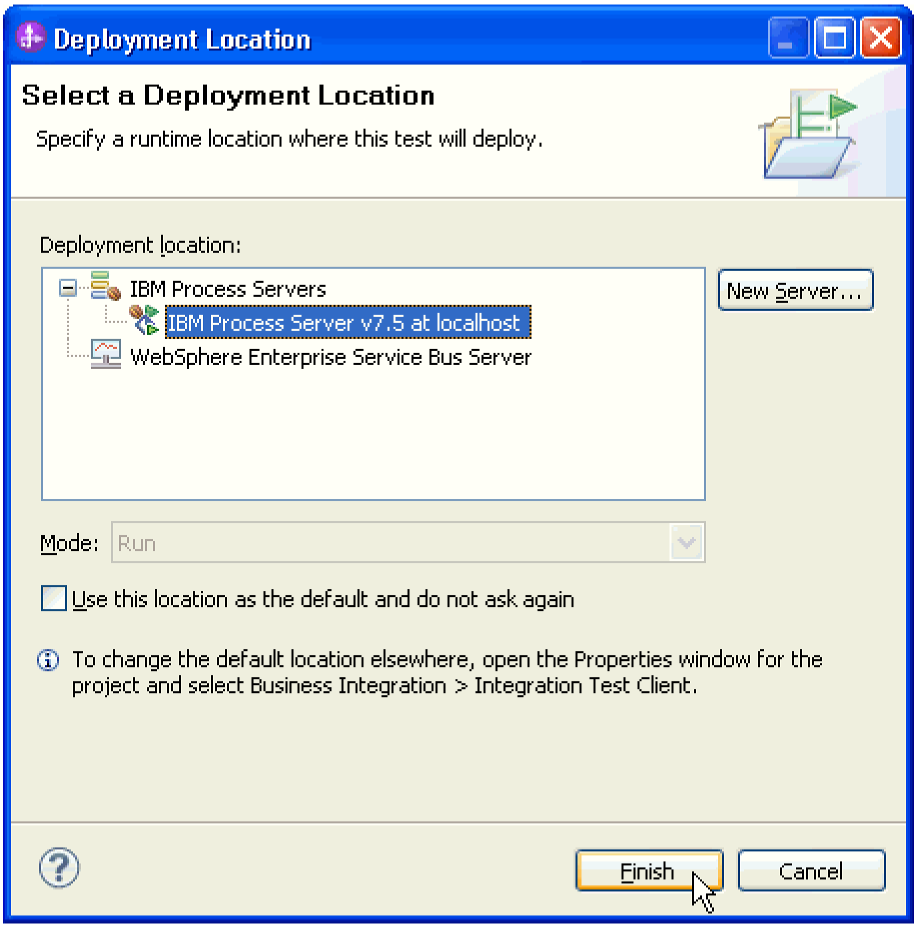


Initial request parameters

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[]	🔗
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.**



Deployment Location

Select a Deployment Location

Specify a runtime location where this test will deploy.

Deployment location:

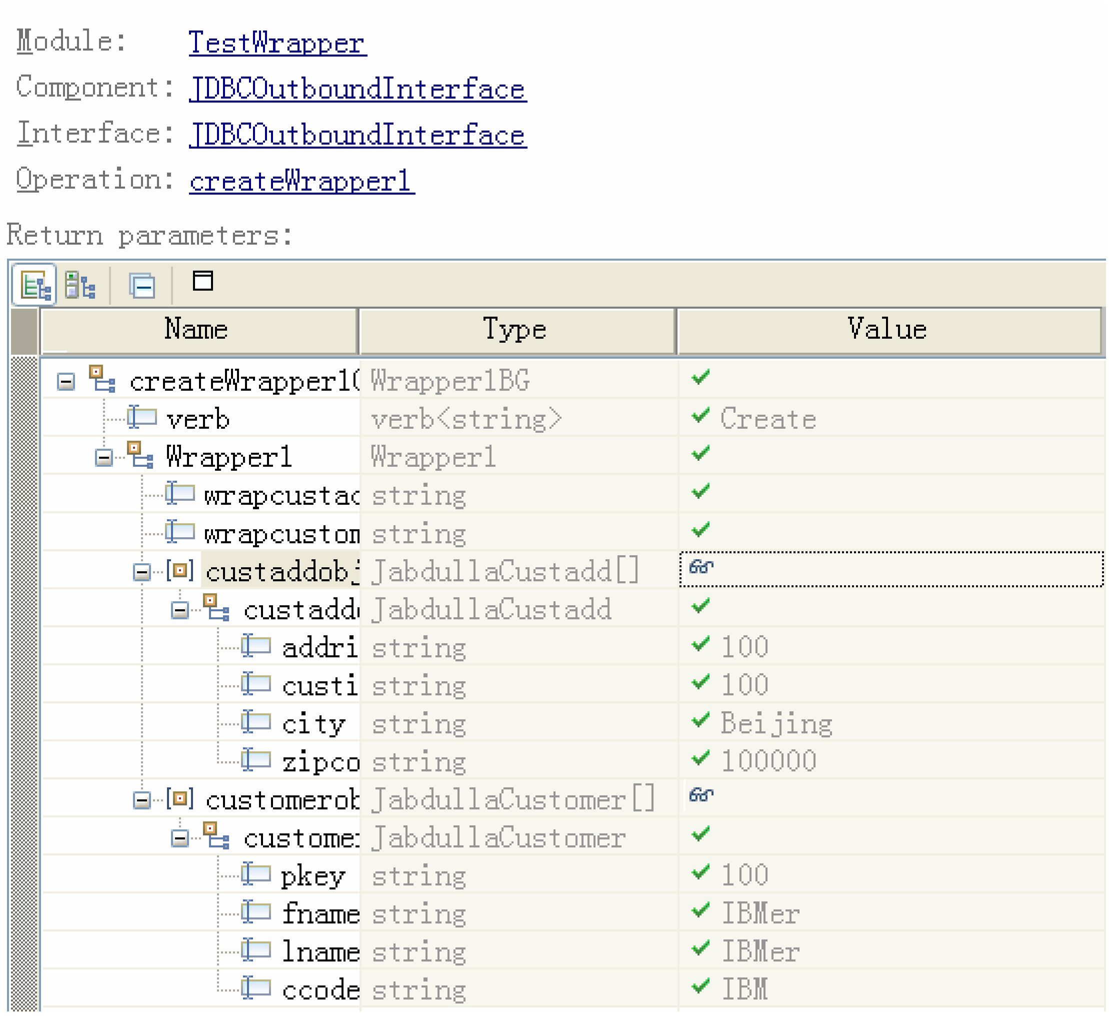
- IBM Process Servers
  - IBM Process Server v7.5 at localhost
- WebSphere Enterprise Service Bus Server

Mode: Run

□ Use this location as the default and do not ask again

To change the default location elsewhere, open the Properties window for the project and select Business Integration > Integration Test Client.

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: JDBCOutboundInterface

Interface: JDBCOutboundInterface

Operation: createWrapper1

Return parameters:

Name	Type	Value
createWrapper1(	Wrapper1BG	✓
verb	verb<string>	✓ Create
Wrapper1	Wrapper1	✓
wrapcustac	string	✓
wrapcuston	string	✓
custaddobj	JabdullaCustadd[]	6ර
custadd	JabdullaCustadd	✓
addri	string	✓ 100
custi	string	✓ 100
city	string	✓ Beijing
zipco	string	✓ 100000
customerob	JabdullaCustomer[]	6ර
custome:	JabdullaCustomer	✓
pkey	string	✓ 100
fname	string	✓ IBMer
lname	string	✓ IBMer
ccode	string	✓ IBM

Clear the sample content

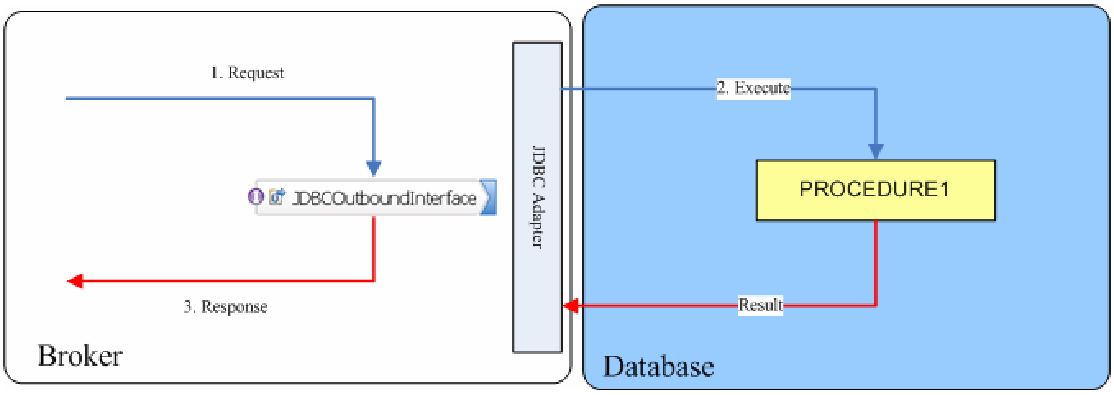
Return the data to its original state.
Nothing is required to clean up after this tutorial.

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 7.5.0.0 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

```
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.

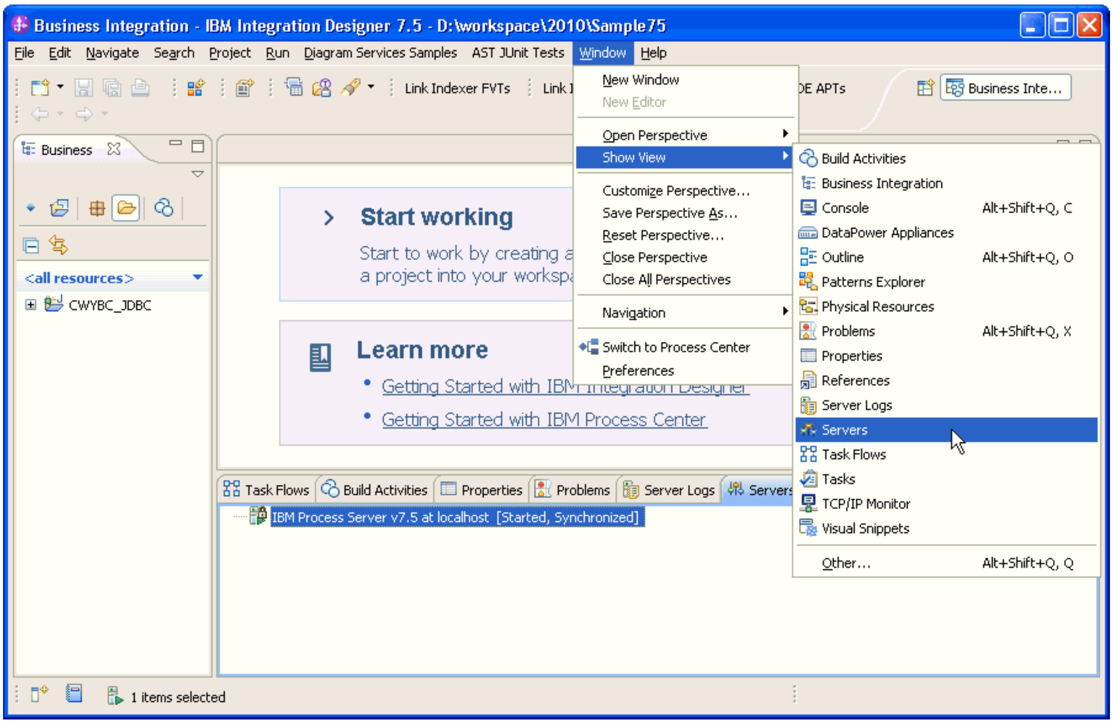
```
CREATE PROCEDURE PROCEDURE1 @var0 varchar(10),
@var1 varchar(10) OUT
AS
      SELECT PKEY,LNAME,FNAME,CCODE FROM CUSTOMER;
   SELECT ADDRID,CUSTID,CITY,ZIPCODE FROM ADDRESS;
   Set @var1= @var0;
   Return (@var1)
GO
```

Create an authentication alias

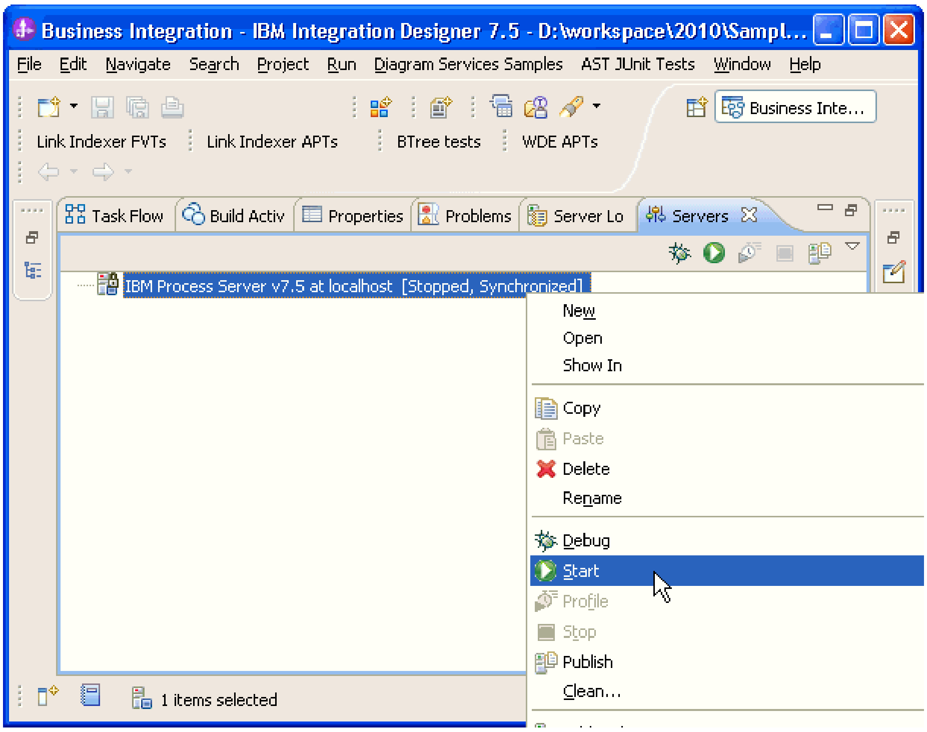
The authentication alias needs to be set because the data source created in the next section uses the username and password set in the authentication alias to connect to the database.

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

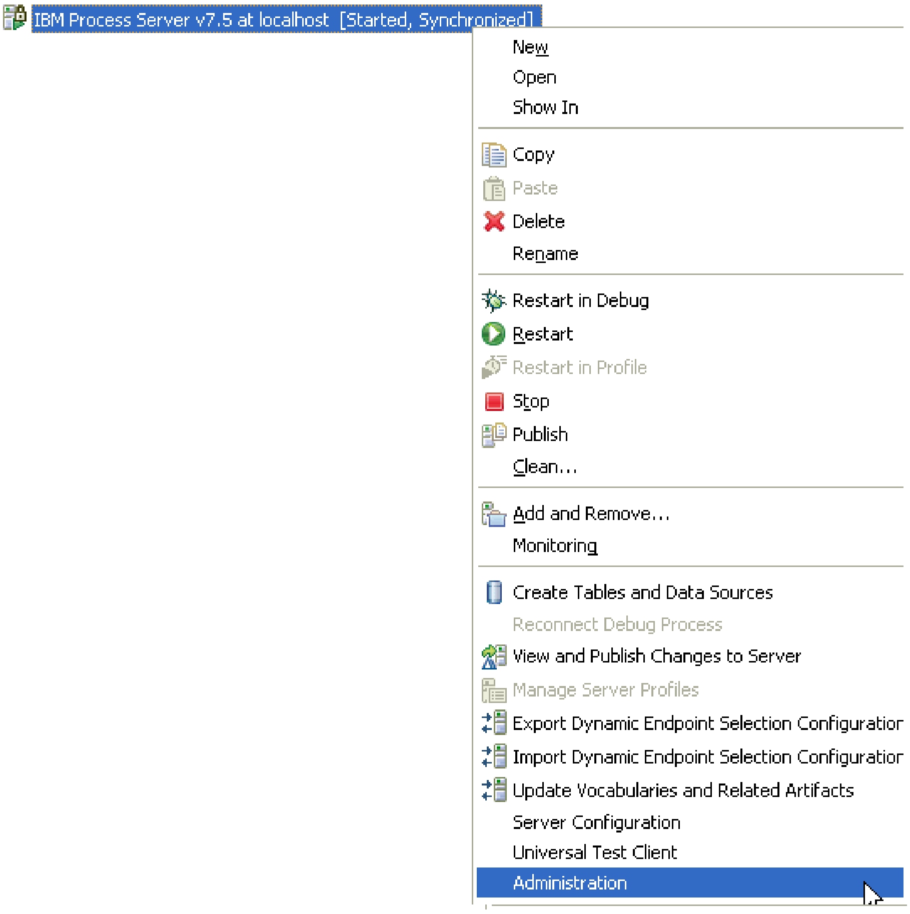
1. In IBM Integration Designer, switch to the **Servers** view by selecting **Windows > Show View > Servers**.



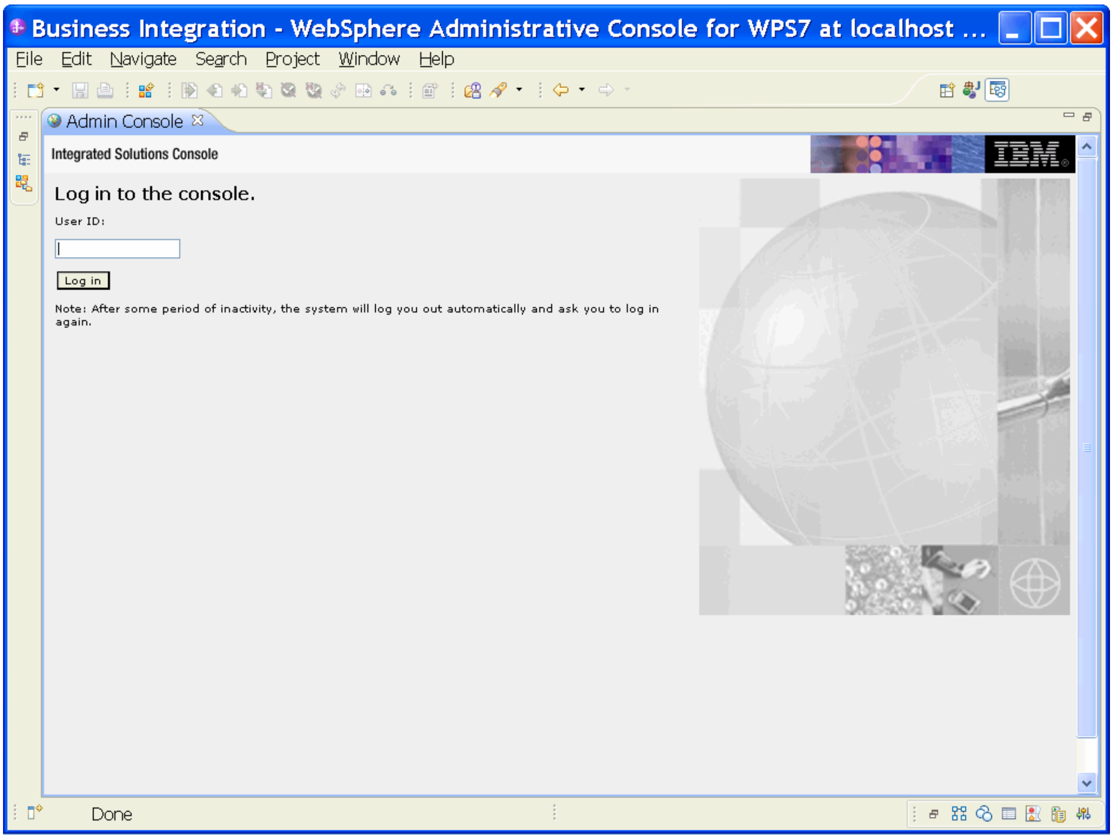
2. In the **Servers** view, right-click the server that you want to start and select **Start**.


Business Integration - IBM Integration Designer 7.5 - D:\workspace\2010\Sampl...
File Edit Navigate Search Project Run Diagram Services Samples AST JUnit Tests Window Help
Business Inte...
Link Indexer FVTs
Link Indexer APTs
BTree tests
WDE APTs
Task Flow
Build Activ
Properties
Problems
Server Lo
Servers
IBM Process Server v7.5 at localhost [Stopped, Synchronized]
New
Open
Show In
Copy
Paste
Delete
Rename
Debug
Start
Profile
Stop
Publish
Clean...
1 items selected

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


IBM Process Server v7.5 at localhost [Started, Synchronized]
New
Open
Show In
Copy
Paste
Delete
Rename
Restart in Debug
Restart
Restart in Profile
Stop
Publish
Clean...
Add and Remove...
Monitoring
Create Tables and Data Sources
Reconnect Debug Process
View and Publish Changes to Server
Manage Server Profiles
Export Dynamic Endpoint Selection Configuratior
Import Dynamic Endpoint Selection Configuratior
Update Vocabularies and Related Artifacts
Server Configuration
Universal Test Client
Administration

4. Log on to the administrative console.



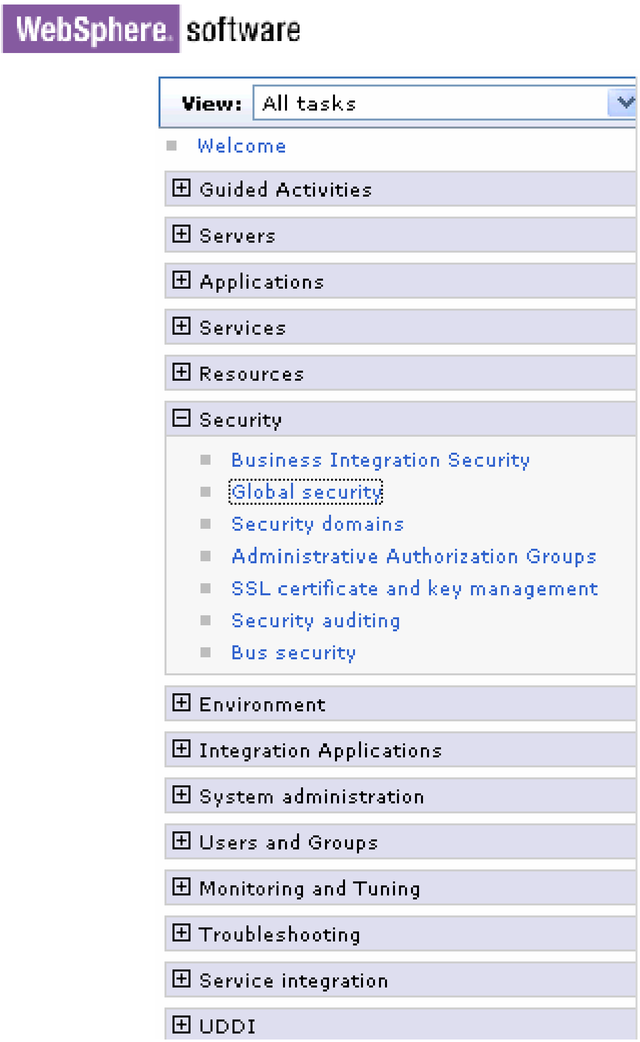
Business Integration - WebSphere Administrative Console for WPS7 at localhost ...
File Edit Navigate Search Project Window Help
Admin Console
Integrated Solutions Console
IBM

Log in to the console.

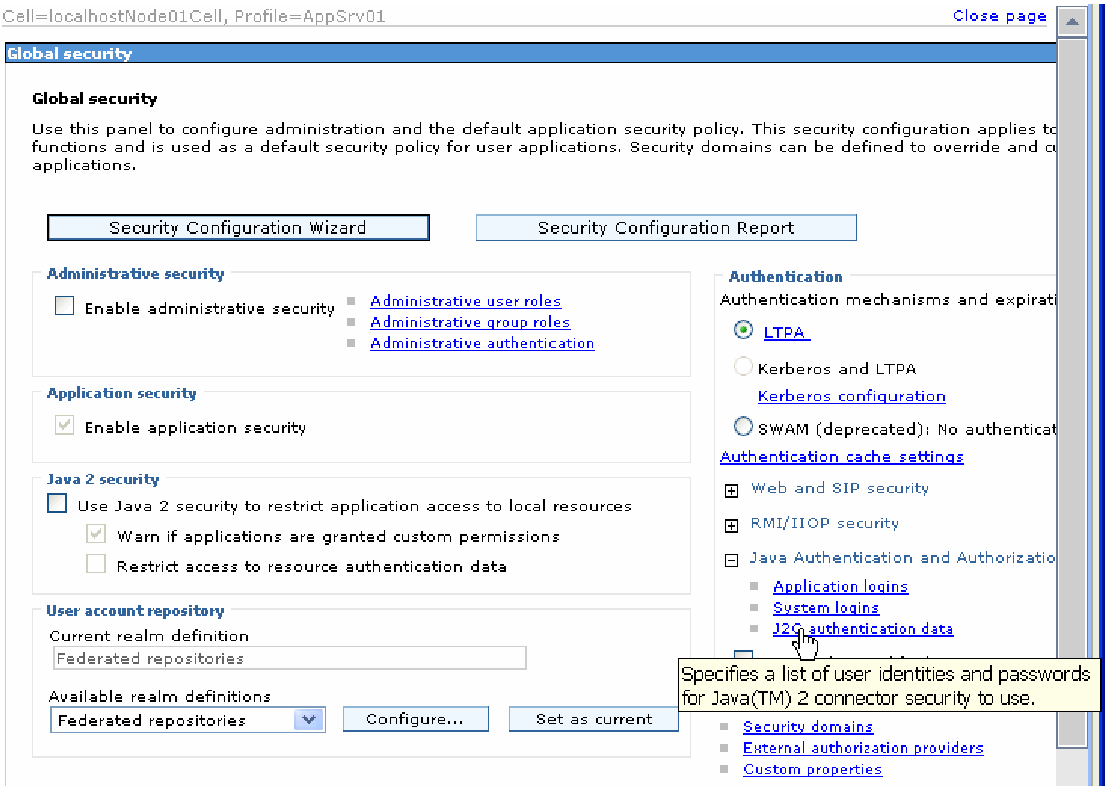
User ID:

Note: After some period of inactivity, the system will log you out automatically and ask you to log in again.

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



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



Cell=localhostNode01Cell, Profile=AppSrv01

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 cu applications.

Administrative security

□ Enable administrative security

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 expirati

● LTPA

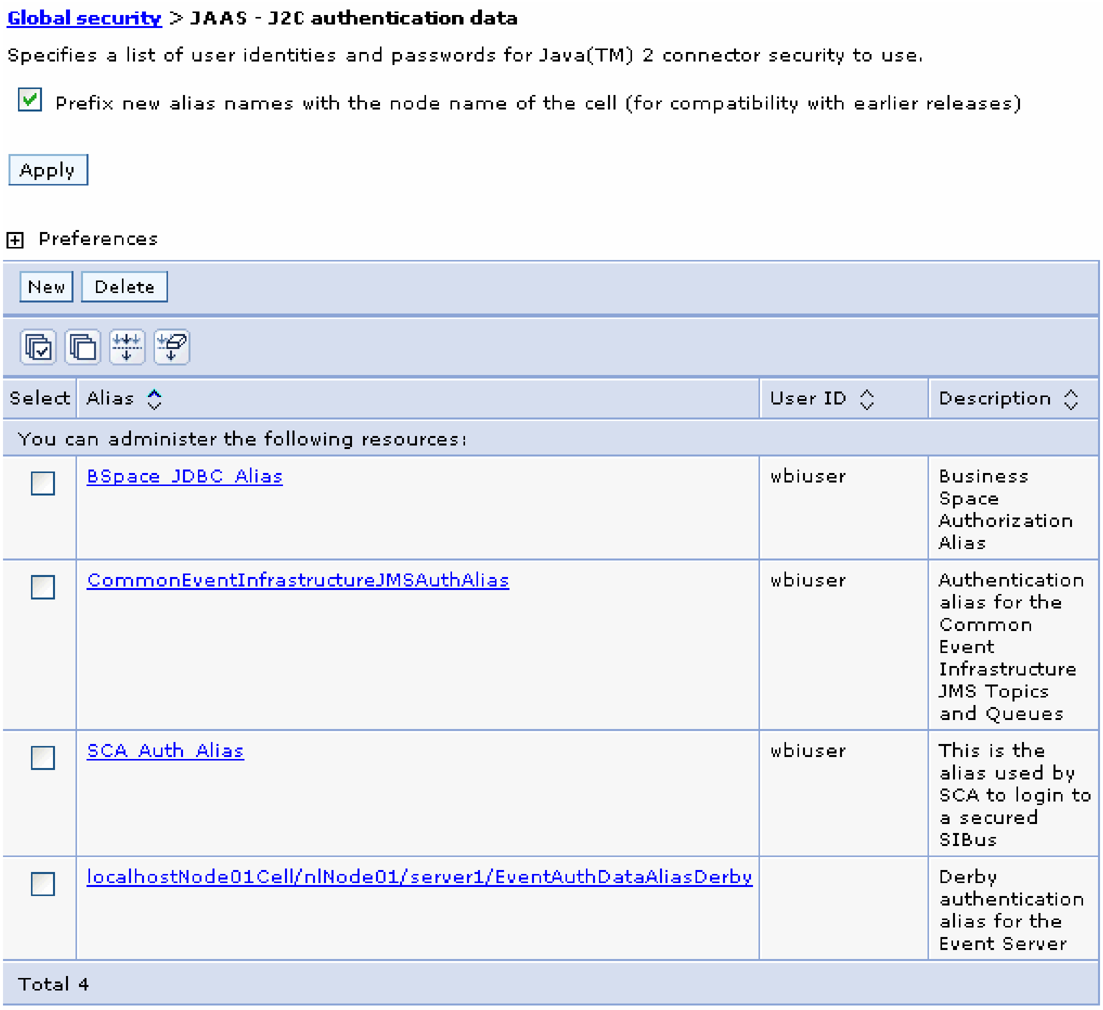
○ Kerberos and LTPA

○ SWAM (deprecated): No authenticat

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

Close page
Security Configuration Wizard
Security Configuration Report
Administrative user roles
Administrative group roles
Administrative authentication
Configure...
Set as current
Kerberos configuration
Authentication cache settings
⊞ Web and SIP security
⊞ RMI/IIOP security
⊟ Java Authentication and Authorizatio
Application logins
System logins
J2C authentication data
Security domains
External authorization providers
Custom properties

A list of existing aliases is displayed.



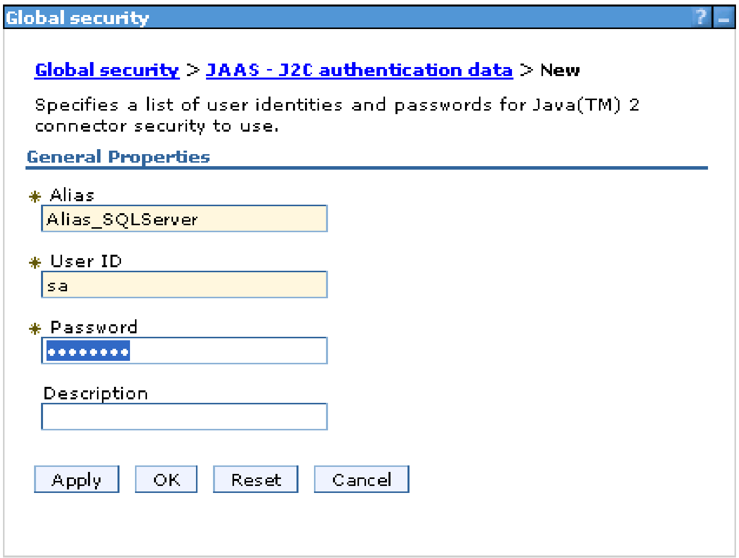
Global security > JAAS - J2C authentication data

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)

Select	Alias	User ID	Description
You can administer the following resources:			
□	BSpace_JDBC_Alias	wbiuser	Business Space Authorization Alias
□	CommonEventInfrastructureJMSAuthAlias	wbiuser	Authentication alias for the Common Event Infrastructure JMS Topics and Queues
□	SCA_Auth_Alias	wbiuser	This is the alias used by SCA to login to a secured SIBus
□	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**.



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

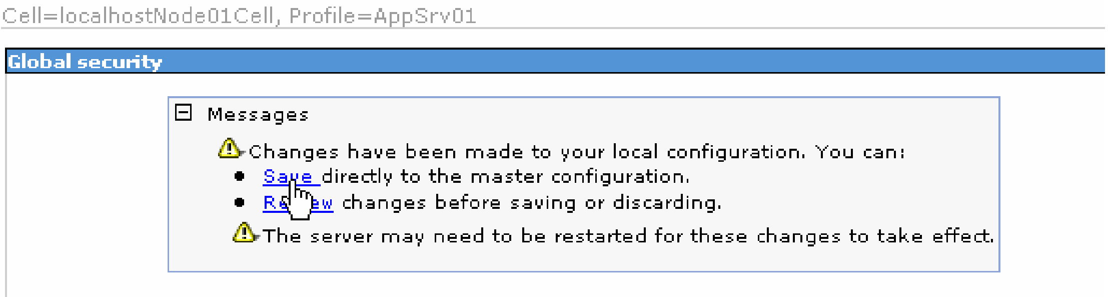
* Alias: Alias_SQLServer

* User ID: sa

* Password: ••••••••

Description:

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



Cell=localhostNode01Cell, Profile=AppSrv01

Global security

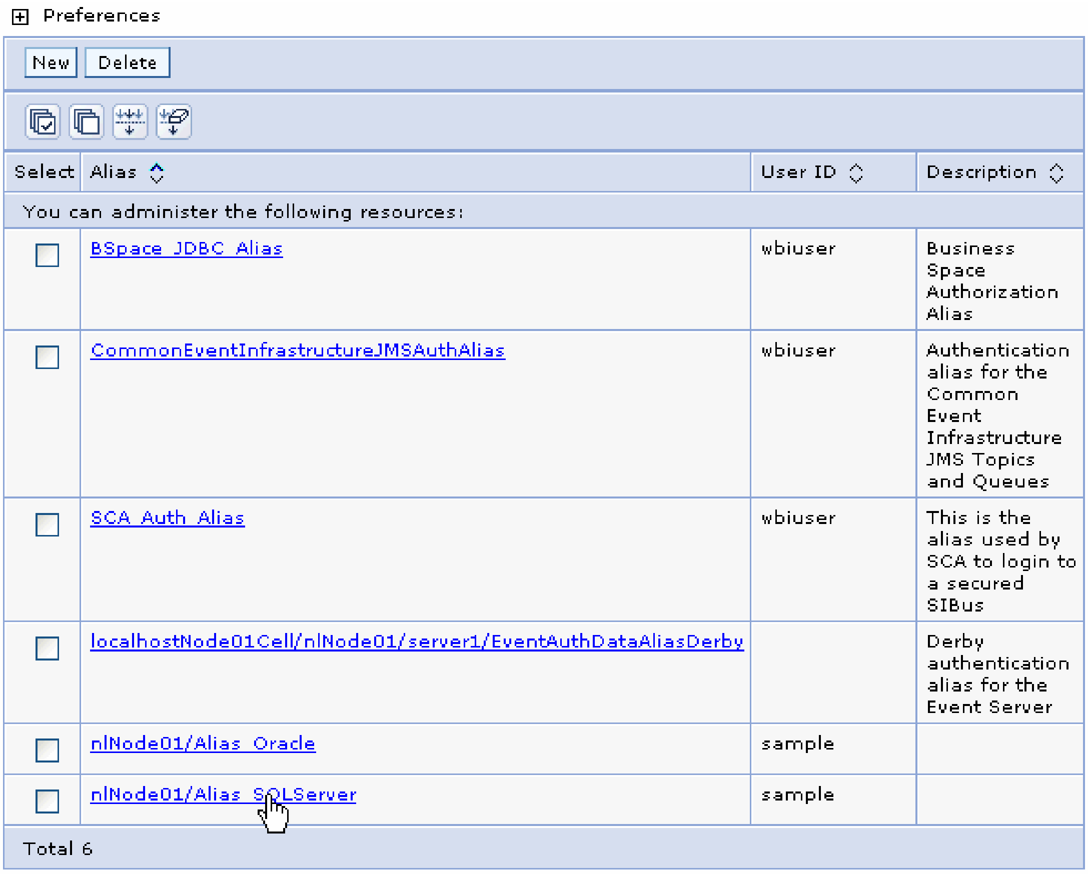
Messages

⚠ 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.

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



⊞ Preferences

Select	Alias	User ID	Description
You can administer the following resources:			
□	BSpace_JDBC_Alias	wbiuser	Business Space Authorization Alias
□	CommonEventInfrastructureJMSAuthAlias	wbiuser	Authentication alias for the Common Event Infrastructure JMS Topics and Queues
□	SCA_Auth_Alias	wbiuser	This is the alias used by SCA to login to a secured SIBus
□	localhostNode01Cell/nlNode01/server1/EventAuthDataAliasDerby		Derby authentication alias for the Event Server
□	nlNode01/Alias_Oracle	sample	
□	nlNode01/Alias_SQLServer	sample	
Total 6			

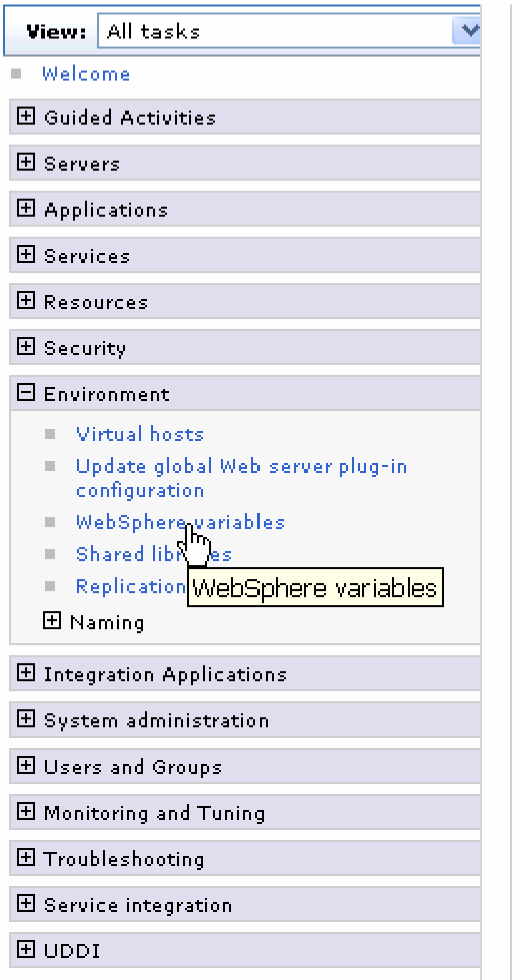
Create a data source

Create a data source in IBM 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.

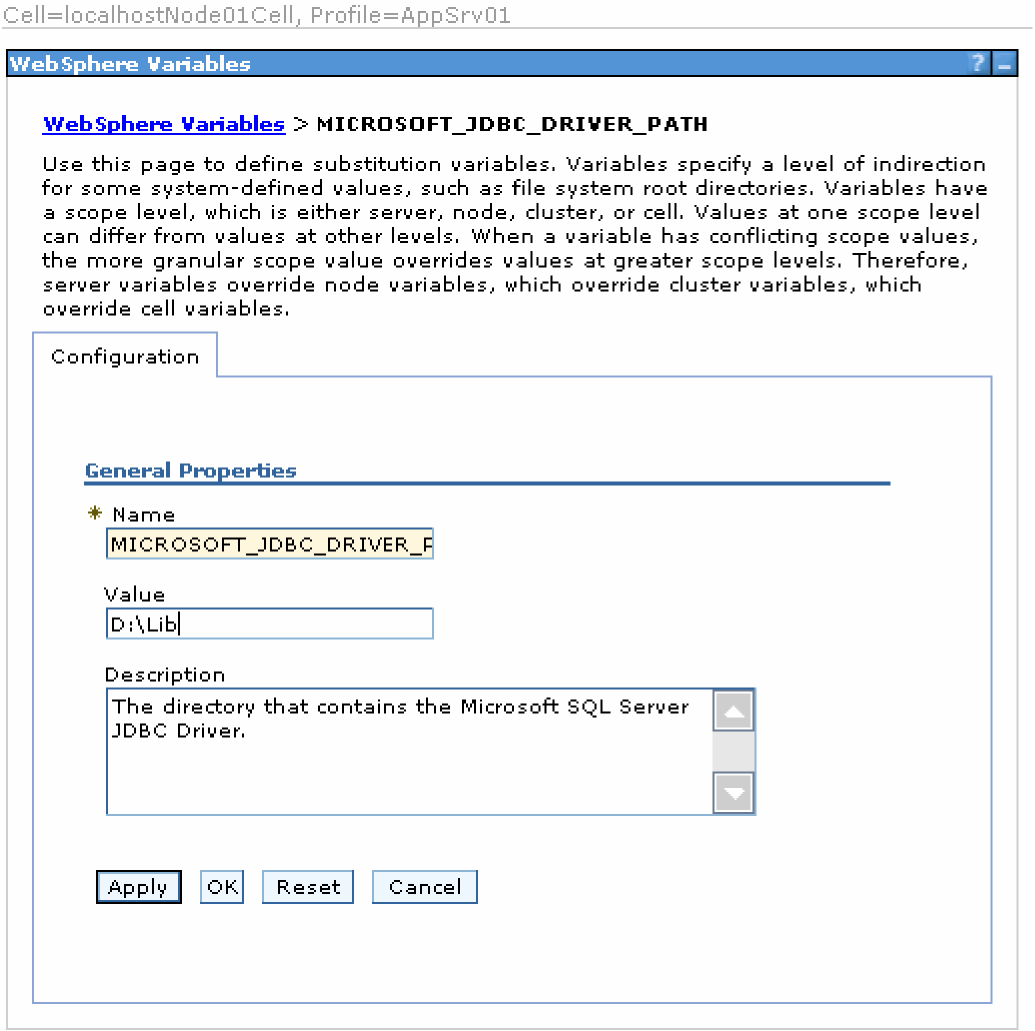
Note: This tutorial uses SQL Server as the database and the SQL Server JDBC driver sqljdbc.jar.

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

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



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



Cell=localhostNode01Cell, Profile=AppSrv01

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.

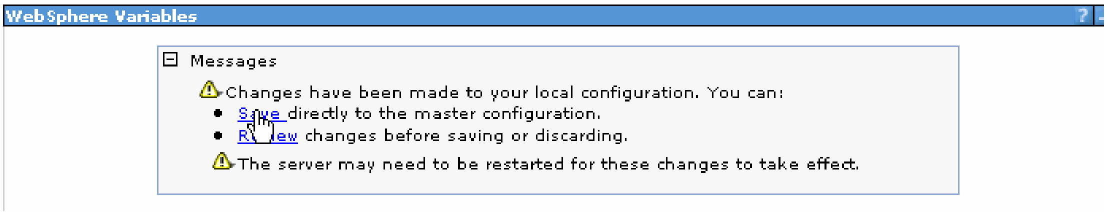
General Properties

* Name
MICROSOFT_JDBC_DRIVER_F

Value
D:\Lib

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

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



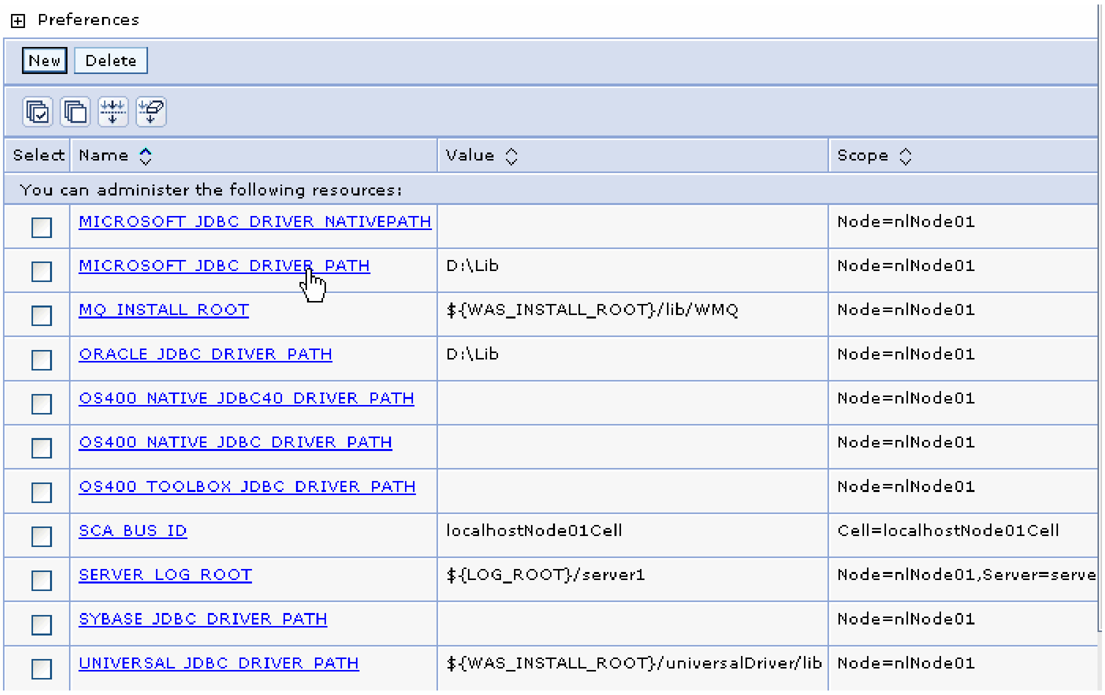
⊟ Messages

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

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

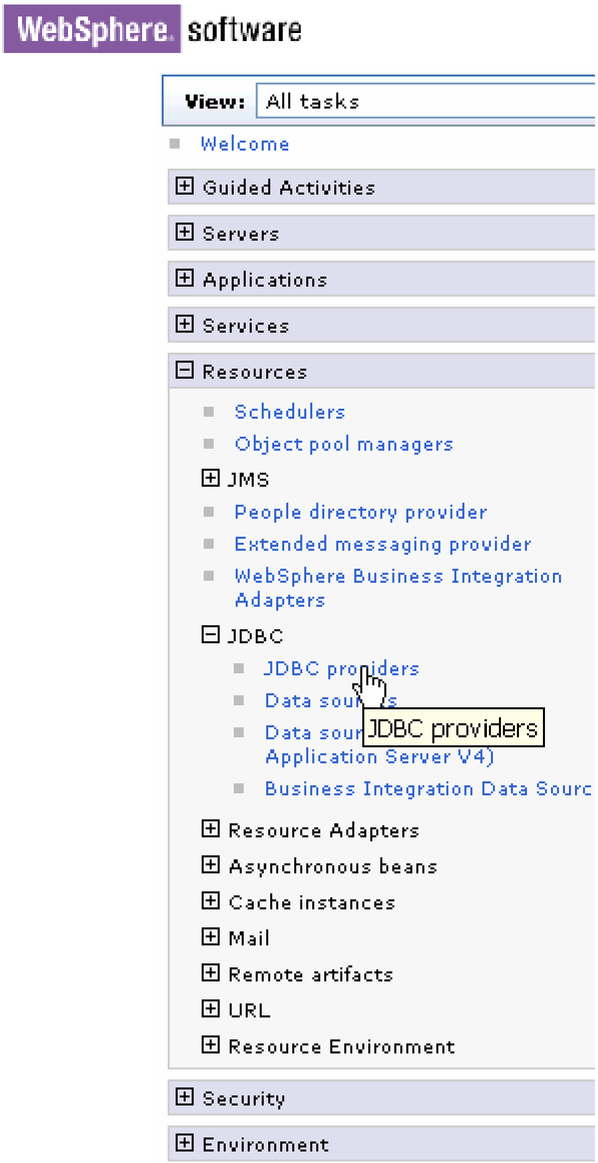
⚠ The server may need to be restarted for these changes to take effect.

The variable is added and appears in the list.

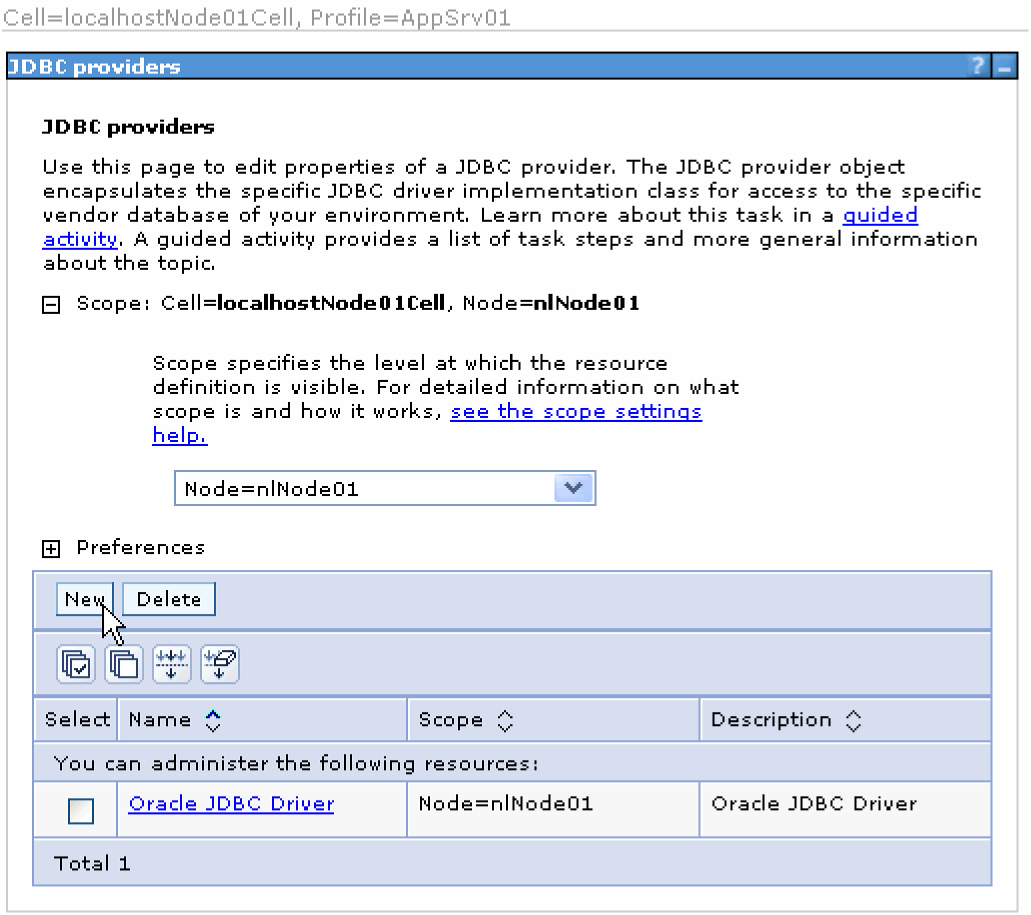


Select	Name	Value	Scope
You can administer the following resources:			
□	MICROSOFT_JDBC_DRIVER_NATIVEPATH		Node=nlNode01
□	MICROSOFT_JDBC_DRIVER_PATH	D:\Lib	Node=nlNode01
□	MQ_INSTALL_ROOT	${WAS_INSTALL_ROOT}/lib/WMQ	Node=nlNode01
□	ORACLE_JDBC_DRIVER_PATH	D:\Lib	Node=nlNode01
□	OS400_NATIVE_JDBC40_DRIVER_PATH		Node=nlNode01
□	OS400_NATIVE_JDBC_DRIVER_PATH		Node=nlNode01
□	OS400_TOOLBOX_JDBC_DRIVER_PATH		Node=nlNode01
□	SCA_BUS_ID	localhostNode01Cell	Cell=localhostNode01Cell
□	SERVER_LOG_ROOT	${LOG_ROOT}/server1	Node=nlNode01,Server=serve
□	SYBASE_JDBC_DRIVER_PATH		Node=nlNode01
□	UNIVERSAL_JDBC_DRIVER_PATH	${WAS_INSTALL_ROOT}/universalDriver/lib	Node=nlNode01

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



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



Cell=localhostNode01Cell, Profile=AppSrv01

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=**nlNode01**

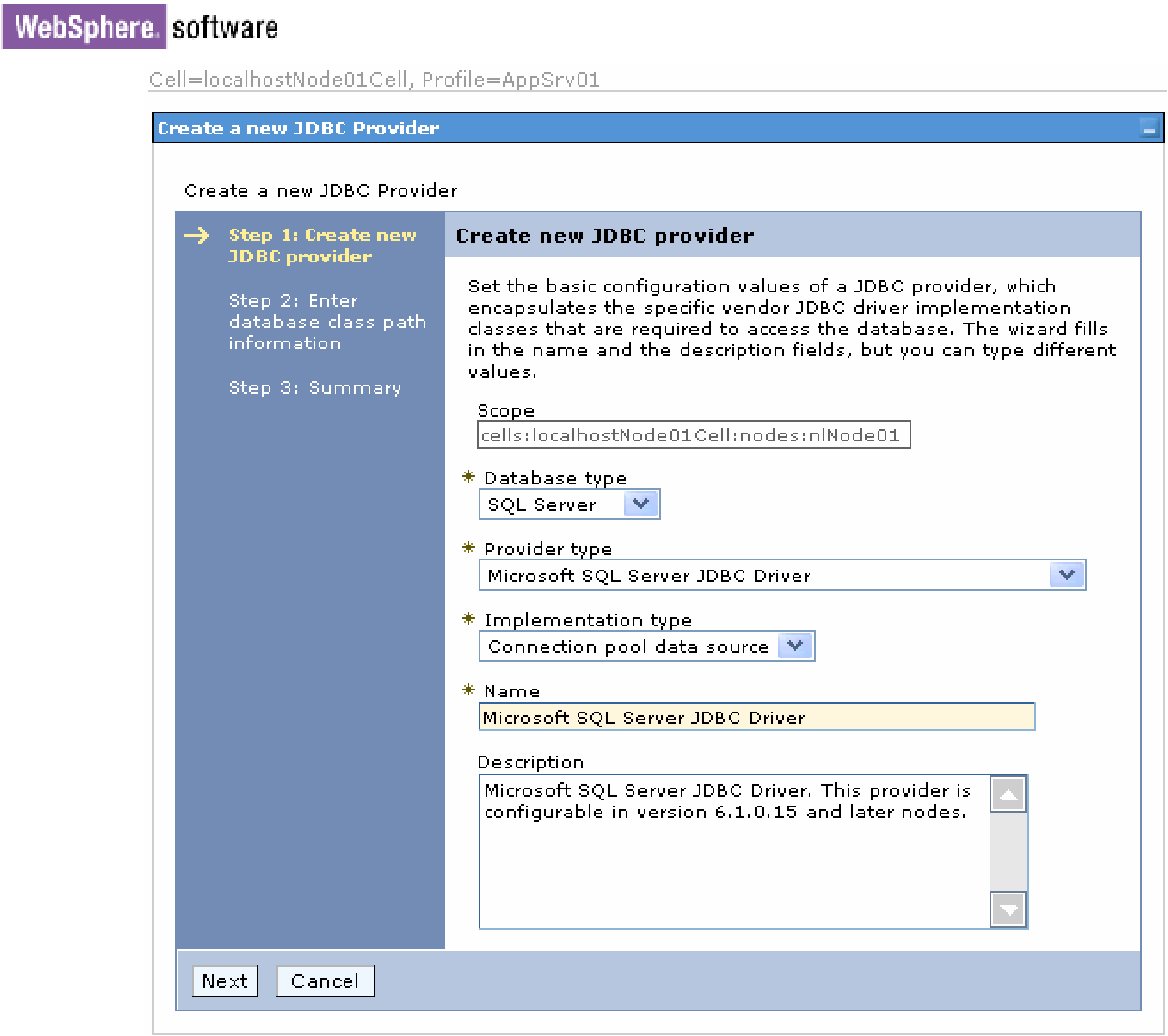
Scope specifies the level at which the resource definition is visible. For detailed information on what scope is and how it works, see the scope settings help.

Node=nlNode01

⊞ Preferences

Select	Name	Scope	Description
You can administer the following resources:			
□	Oracle JDBC Driver	Node=nlNode01	Oracle JDBC Driver
Total 1			

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



Cell=localhostNode01Cell, Profile=AppSrv01

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:nlNode01

* Database type: SQL Server

* Provider type: Microsoft SQL Server JDBC Driver

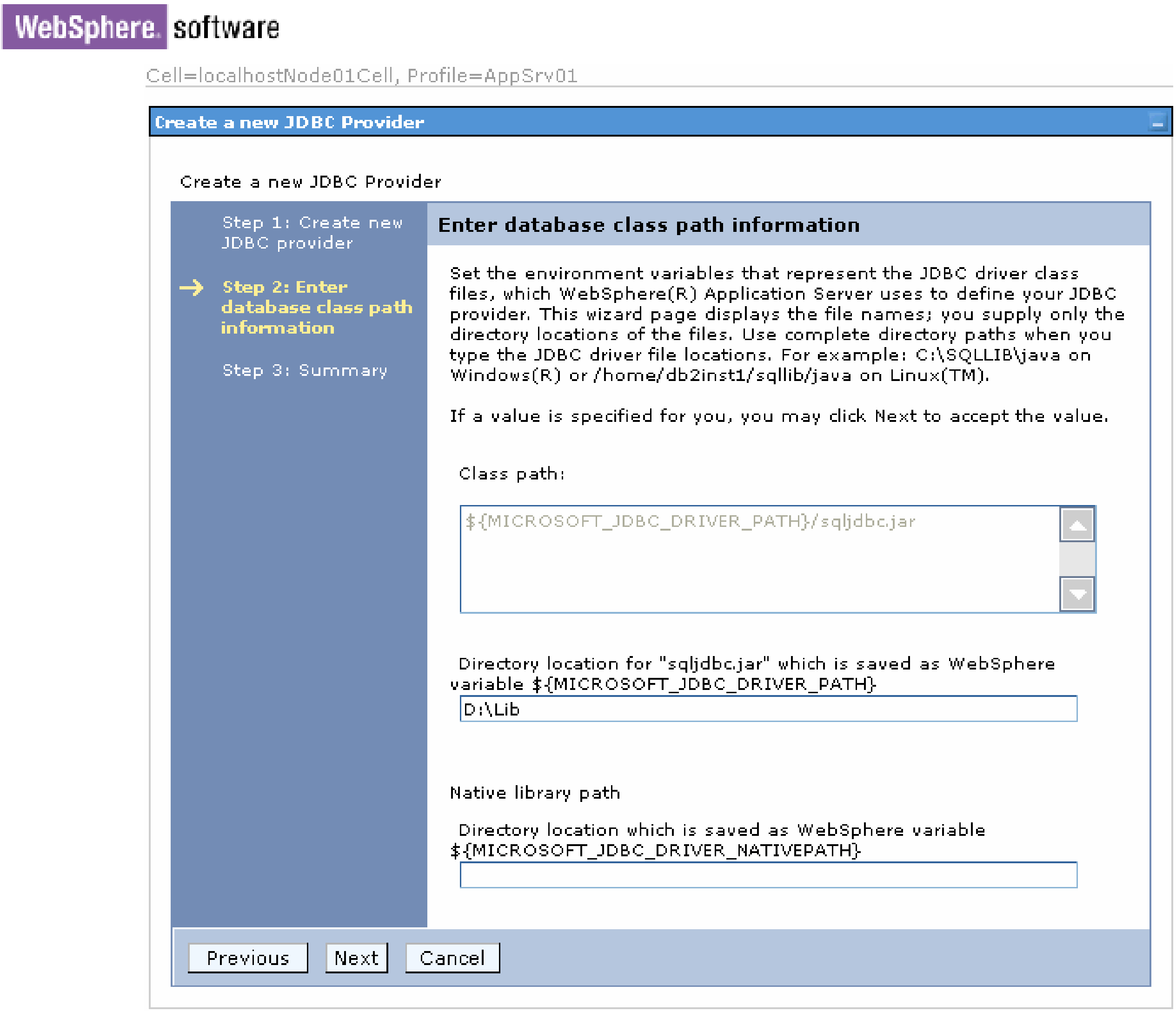
* Implementation type: Connection pool data source

* 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.

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**.



Cell=localhostNode01Cell, Profile=AppSrv01

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: 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

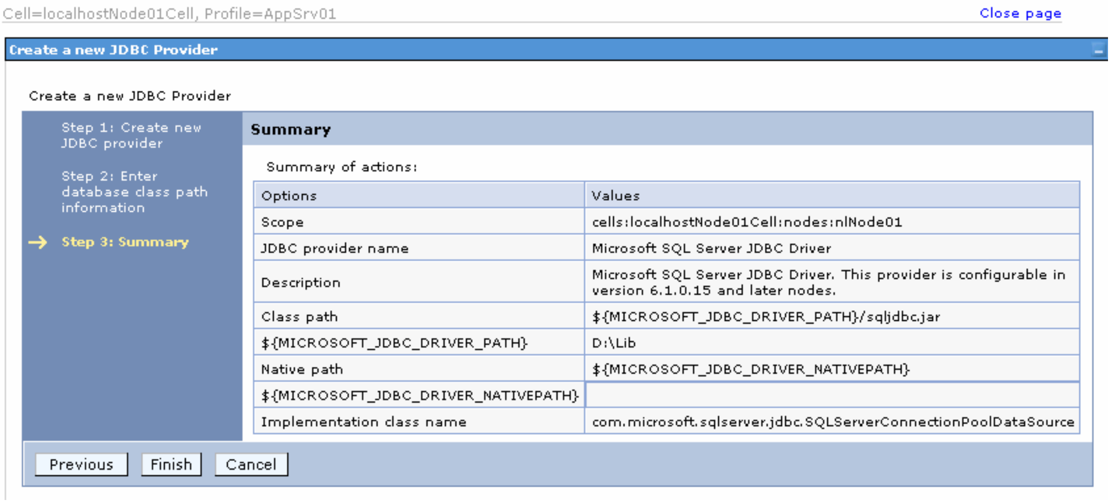
Directory location for "sqljdbc.jar" which is saved as WebSphere variable ${MICROSOFT_JDBC_DRIVER_PATH}

D:\Lib

Native library path

Directory location which is saved as WebSphere variable ${MICROSOFT_JDBC_DRIVER_NATIVEPATH}

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



Cell=localhostNode01Cell, Profile=AppSrv01

Create a new JDBC Provider

Create a new JDBC Provider

Step 1: Create new JDBC provider

Step 2: Enter database class path information

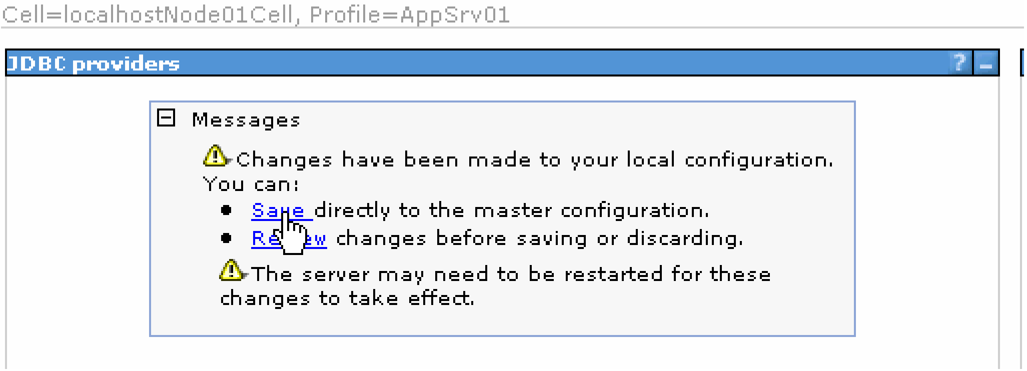
→ Step 3: Summary

Summary

Summary of actions:

Options	Values
Scope	cells:localhostNode01Cell:nodes:nlNode01
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	${MICROSOFT_JDBC_DRIVER_PATH}/sqljdbc.jar
${MICROSOFT_JDBC_DRIVER_PATH}	D:\Lib
Native path	${MICROSOFT_JDBC_DRIVER_NATIVEPATH}
${MICROSOFT_JDBC_DRIVER_NATIVEPATH}	
Implementation class name	com.microsoft.sqlserver.jdbc.SQLServerConnectionPoolDataSource

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



Cell=localhostNode01Cell, Profile=AppSrv01

JDBC providers

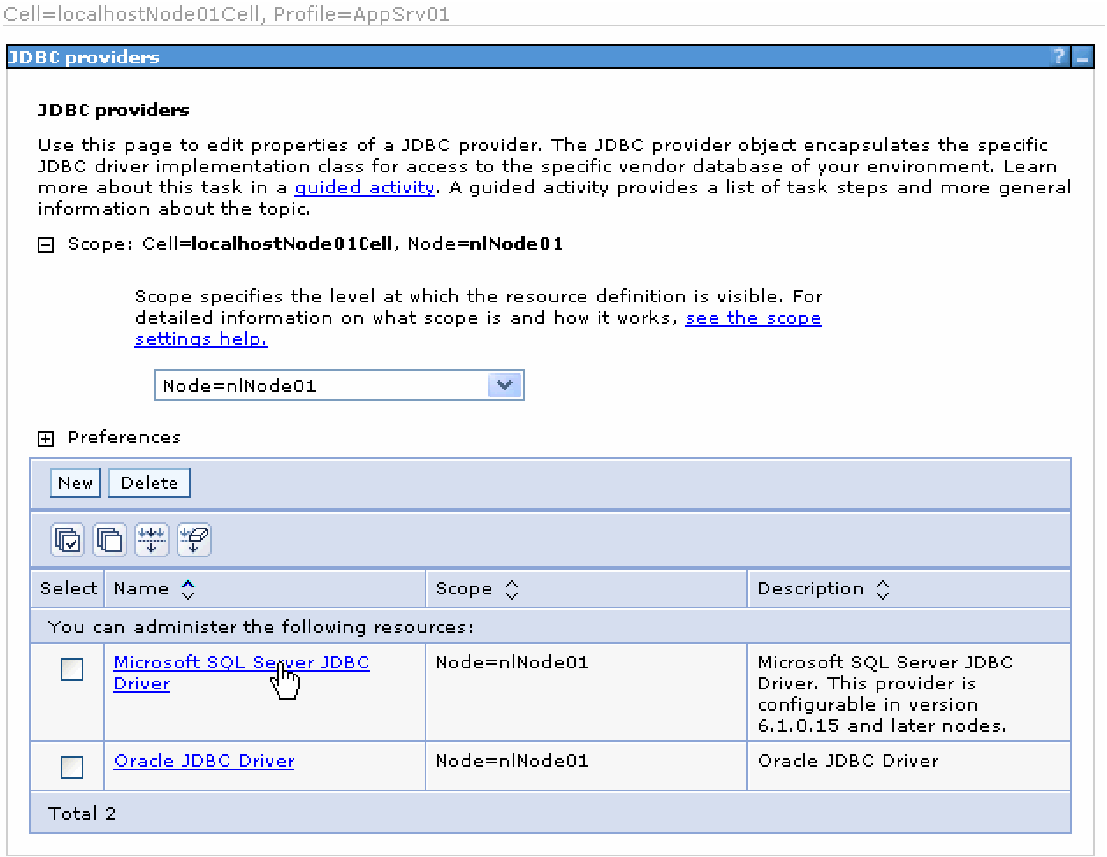
⊟ Messages

⚠ 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

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=**nlNode01**

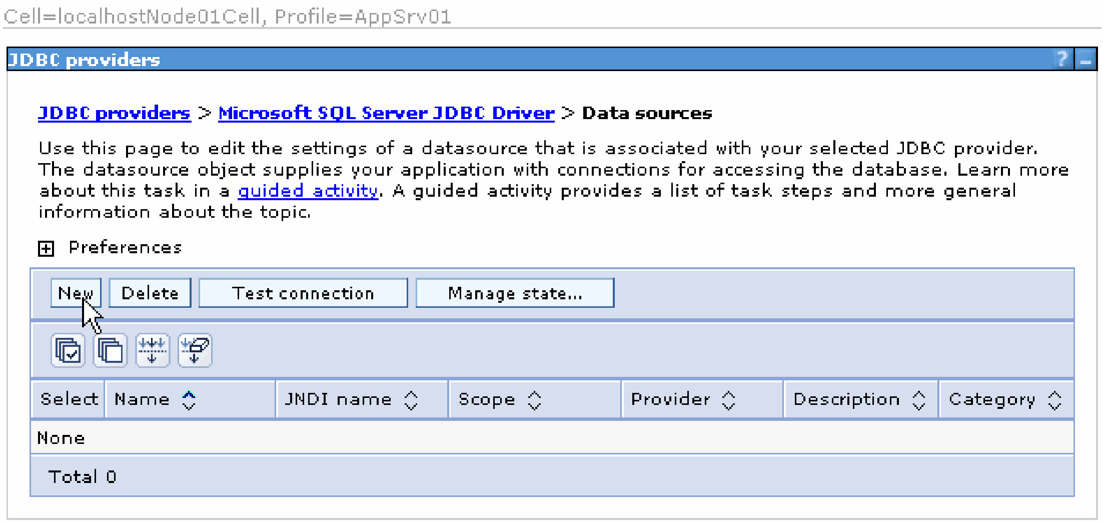
Scope specifies the level at which the resource definition is visible. For detailed information on what scope is and how it works, see the scope settings help.

Node=nlNode01

⊞ Preferences

Select	Name	Scope	Description
You can administer the following resources:			
□	Microsoft SQL Server JDBC Driver	Node=nlNode01	Microsoft SQL Server JDBC Driver. This provider is configurable in version 6.1.0.15 and later nodes.
□	Oracle JDBC Driver	Node=nlNode01	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**

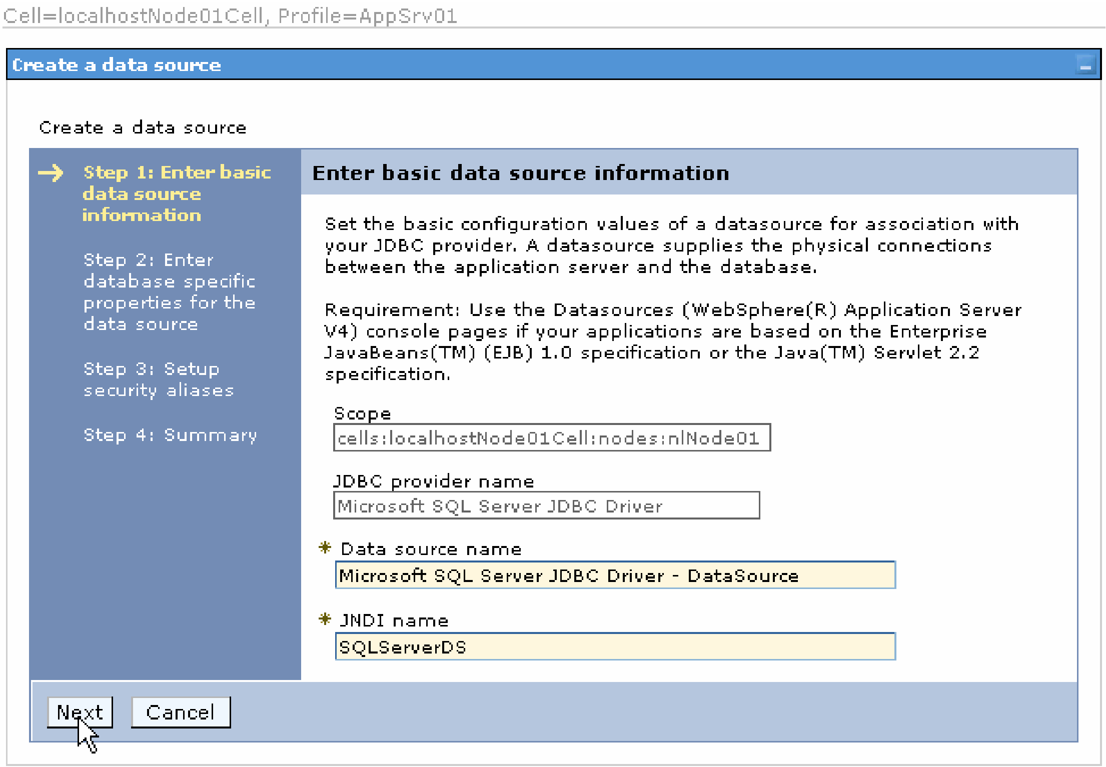
Use this page to edit the settings of a datasource that is associated with your selected JDBC provider. The datasource object supplies your application with connections for accessing the database. Learn more about this task in a guided activity. A guided activity provides a list of task steps and more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

⊞ Preferences

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.

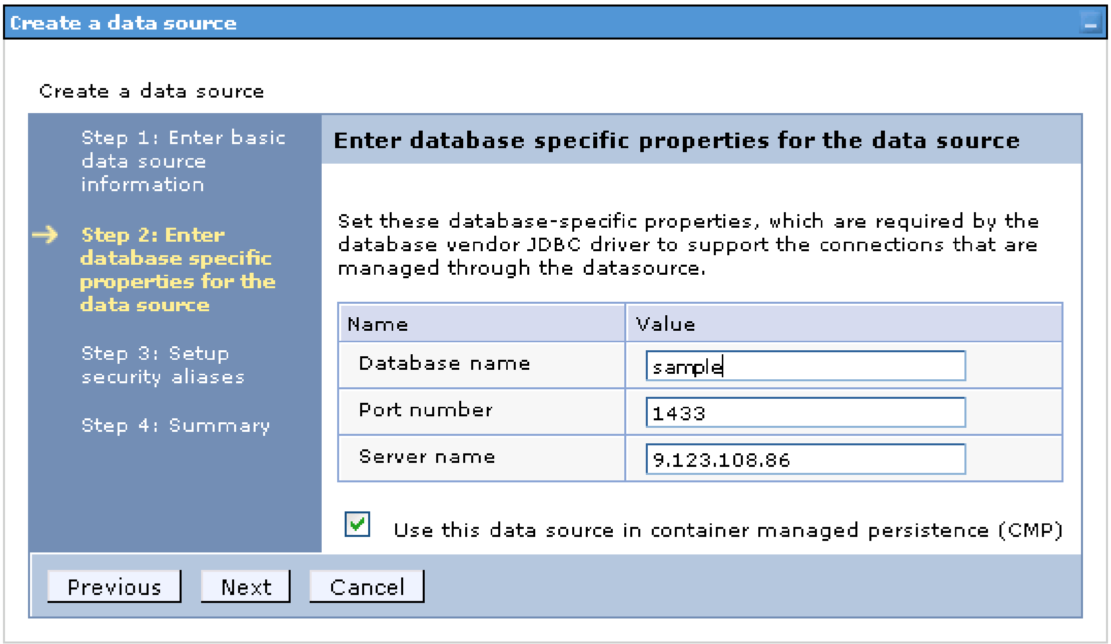
Scope
cells:localhostNode01Cell:nodes:nlNode01

JDBC provider name
Microsoft SQL Server JDBC Driver

* Data source name
Microsoft SQL Server JDBC Driver - DataSource

* JNDI name
SQLServerDS

13. Provide the appropriate **Database name**, **Port number**, **Server name** value. Click **Next**.



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

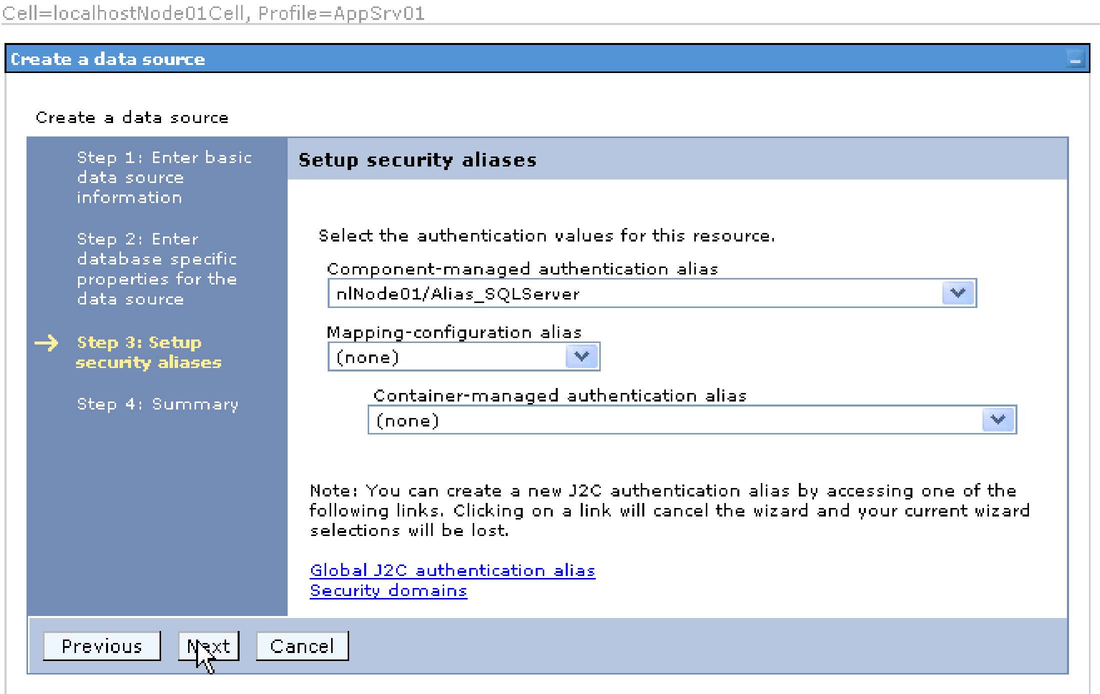
Enter database specific properties for the data source

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
Database name	sample
Port number	1433
Server name	9.123.108.86

☑ Use this data source in container managed persistence (CMP)

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

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

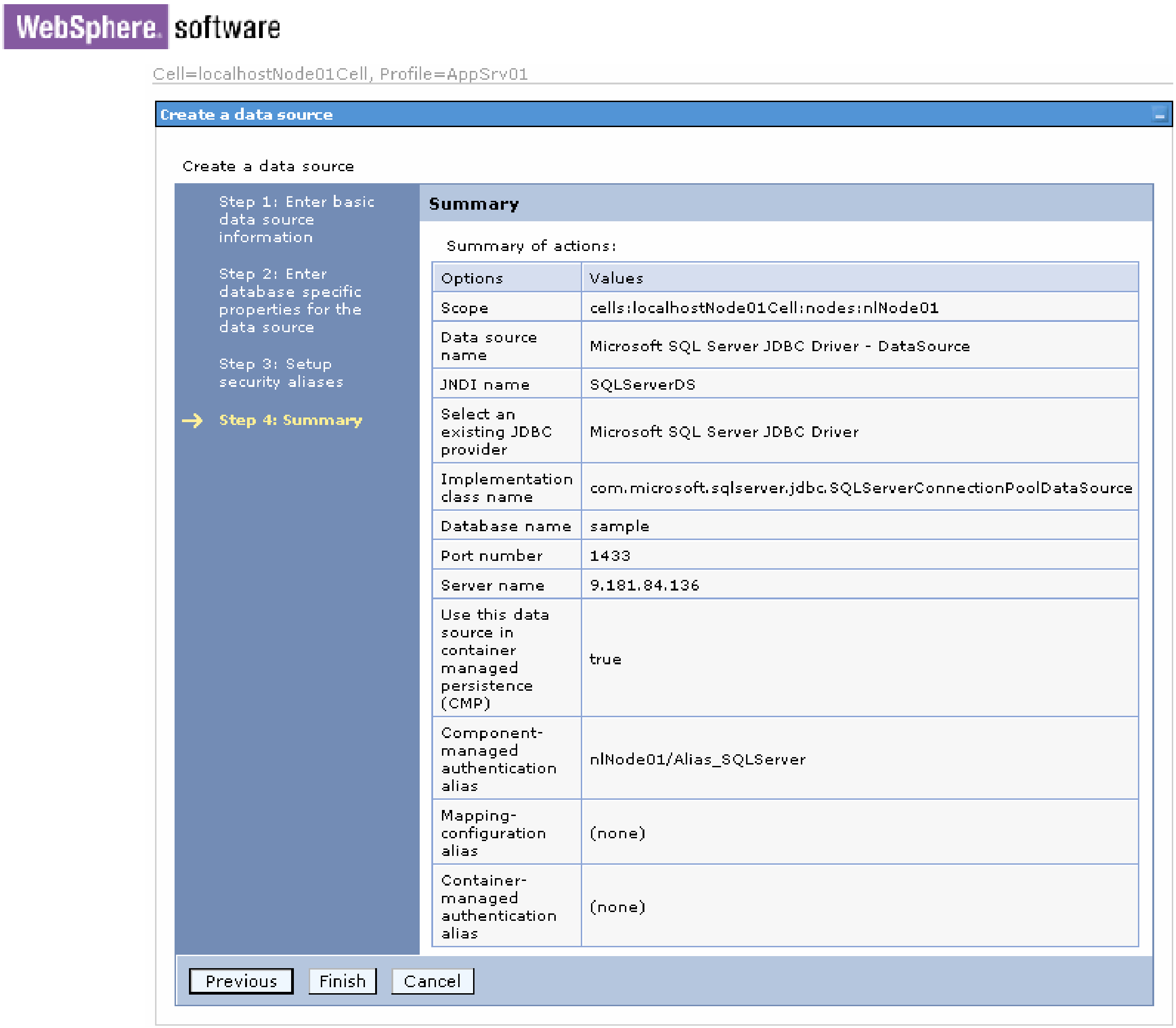
Mapping-configuration alias: (none)

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

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



Cell=localhostNode01Cell, Profile=AppSrv01

Create a data source

Create a data source

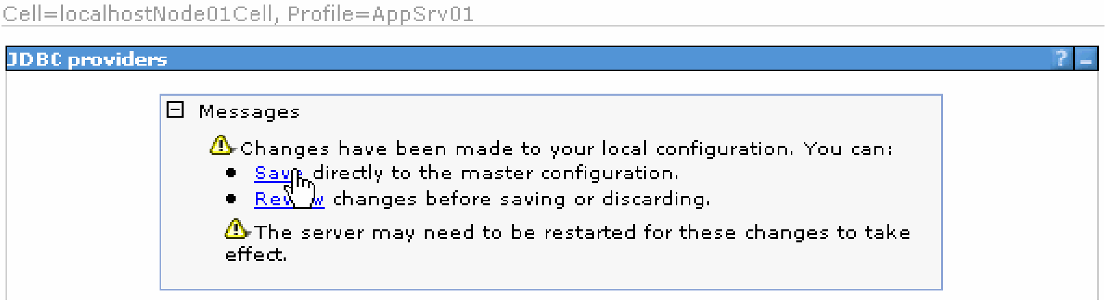
Step 1: Enter basic data source information
Step 2: Enter database specific properties for the data source
Step 3: Setup security aliases
→ Step 4: Summary

Summary

Summary of actions:

Options	Values
Scope	cells:localhostNode01Cell:nodes:nlNode01
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	nlNode01/Alias_SQLServer
Mapping-configuration alias	(none)
Container-managed authentication alias	(none)

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



Cell=localhostNode01Cell, Profile=AppSrv01

JDBC providers

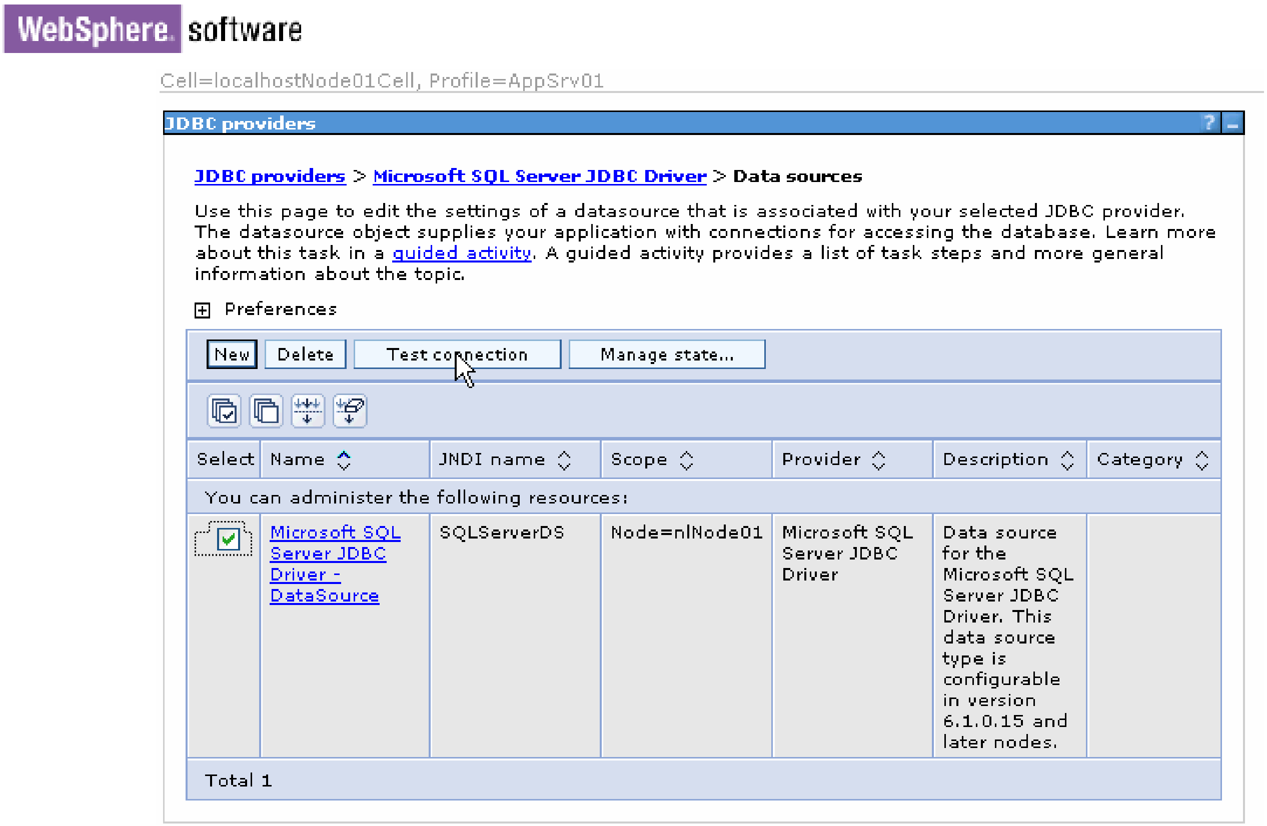
Messages

⚠ 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.

17. Select the check box for the newly created data source and click **Test connection**.



WebSphere. software

Cell=localhostNode01Cell, Profile=AppSrv01

JDBC providers

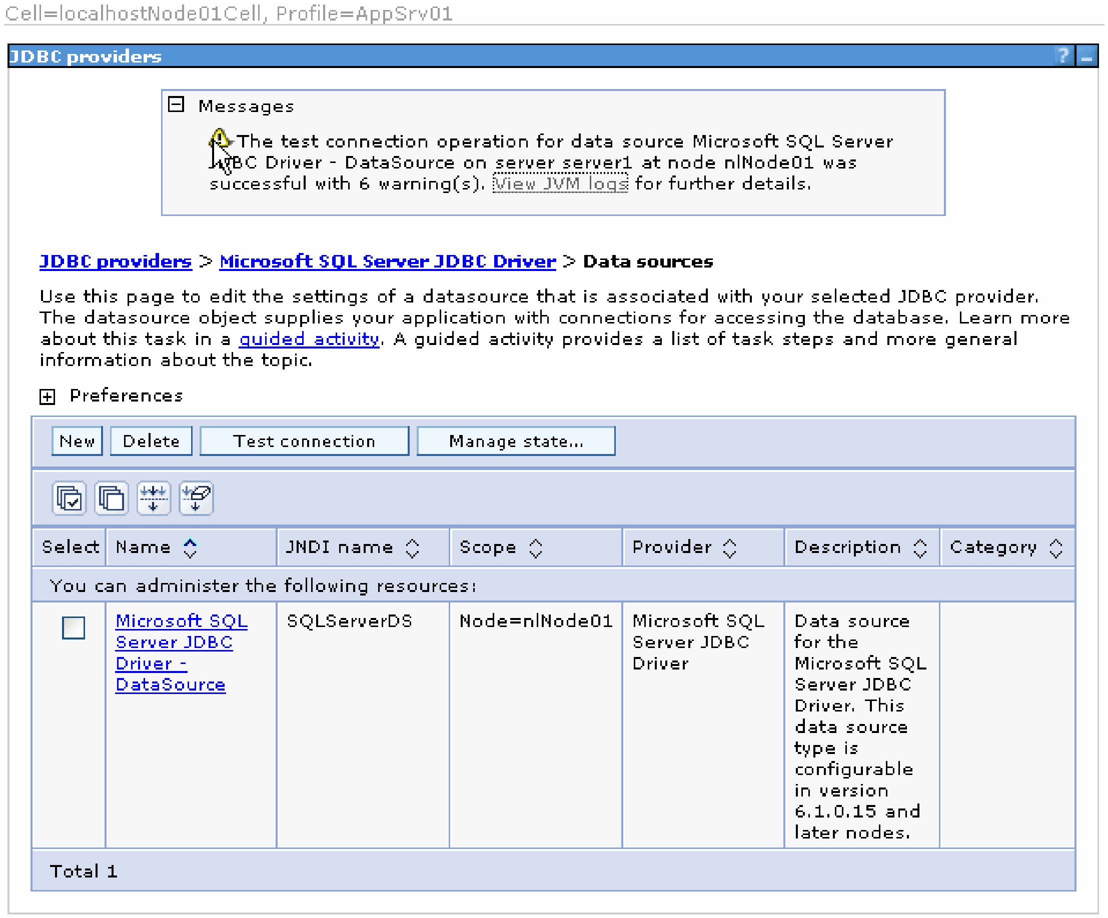
JDBC providers > Microsoft SQL Server JDBC Driver > Data sources

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

Select	Name	JNDI name	Scope	Provider	Description	Category
You can administer the following resources:						
☑	Microsoft SQL Server JDBC Driver - DataSource	SQLServerDS	Node=nlNode01	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

⊟ Messages

⚠The test connection operation for data source Microsoft SQL Server JDBC Driver - DataSource on server server1 at node nlNode01 was successful with 6 warning(s). View JVM logs for further details.

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

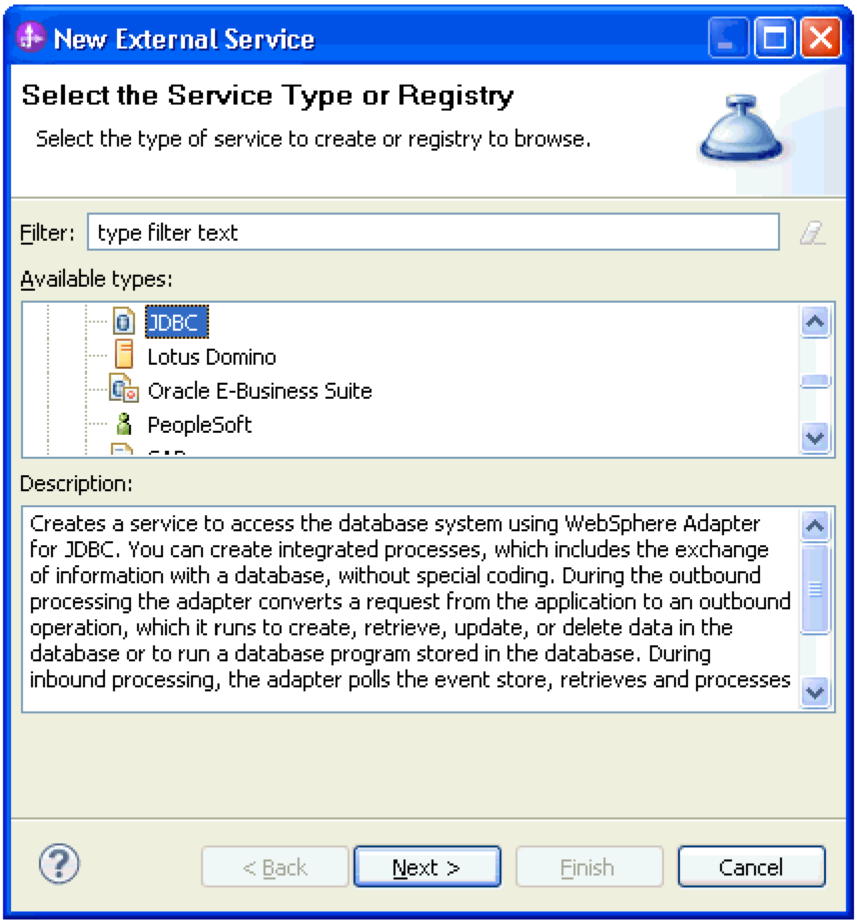
Select	Name	JNDI name	Scope	Provider	Description	Category
You can administer the following resources:						
☐	Microsoft SQL Server JDBC Driver - DataSource	SQLServerDS	Node=nlNode01	Microsoft SQL Server JDBC Driver	Data source for the Microsoft SQL Server JDBC Driver. This data source type is configurable 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

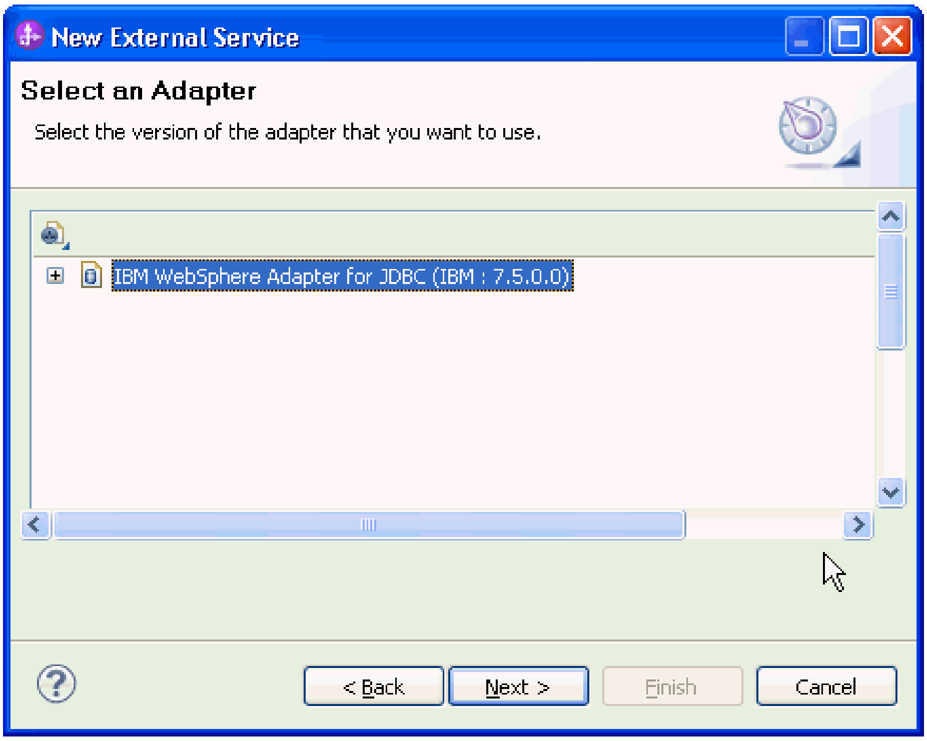
Run the external service wizard to specify business objects, services, and configuration details.

1. Switch to the Business Integration Perspective in IBM Integration Designer 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.**



Creates a service to access the database system using WebSphere Adapter for JDBC. You can create integrated processes, which includes the exchange of information with a database, without special coding. During the outbound processing the adapter converts a request from the application to an outbound operation, which it runs to create, retrieve, update, or delete data in the database or to run a database program stored in the database. During inbound processing, the adapter polls the event store, retrieves and processes

4. Select the **IBM WebSphere Adapter for JDBC (IBM: 7.5.0.0)** and click **Next.**



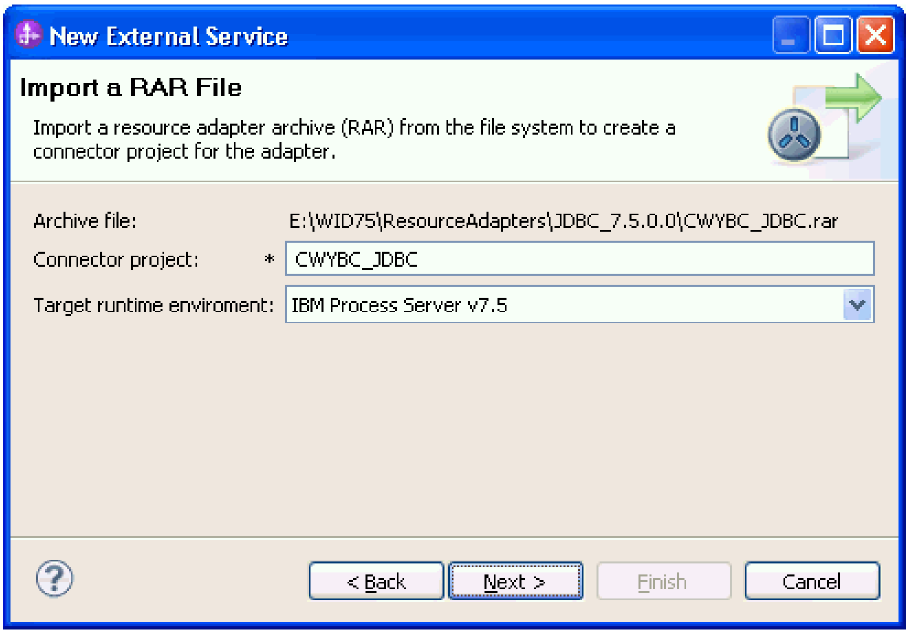
New External Service

Select an Adapter

Select the version of the adapter that you want to use.

IBM WebSphere Adapter for JDBC (IBM : 7.5.0.0)

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

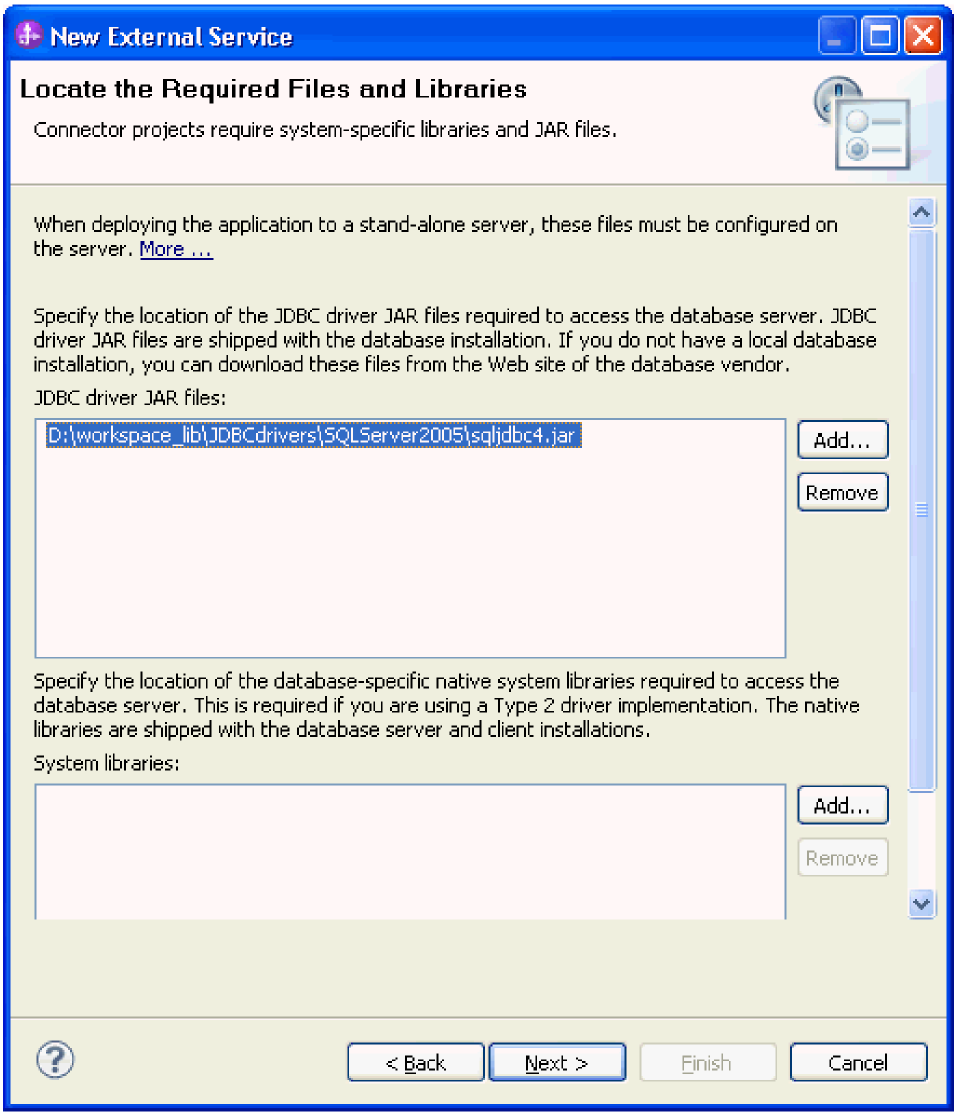
Import a resource adapter archive (RAR) from the file system to create a connector project for the adapter.

Archive file: E:\WID75\ResourceAdapters\JDBC_7.5.0.0\CWYBC_JDBC.rar

Connector project: * CWYBC_JDBC

Target runtime enviroment: IBM Process Server v7.5

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**.



New External Service

Locate the Required Files and Libraries

Connector projects require system-specific libraries and JAR files.

When deploying the application to a stand-alone server, these files must be configured on the server. More ...

Specify the location of the JDBC driver JAR files required to access the database server. JDBC driver JAR files are shipped with the database installation. If you do not have a local database installation, you can download these files from the Web site of the database vendor.

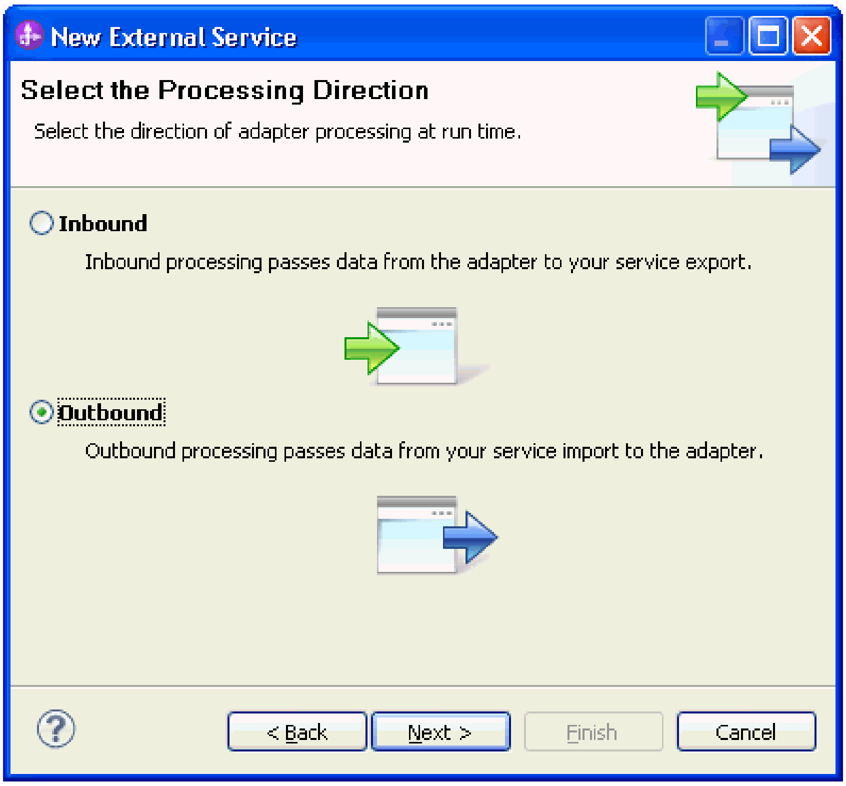
JDBC driver JAR files:

D:\workspace_lib\JDBCdrivers\SQLServer2005\sqljdbc4.jar

Specify the location of the database-specific native system libraries required to access the database server. This is required if you are using a Type 2 driver implementation. The native libraries are shipped with the database server and client installations.

System libraries:

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



Select the Processing Direction

Select the direction of adapter processing at run time.

○ Inbound

Inbound processing passes data from the adapter to your service export.

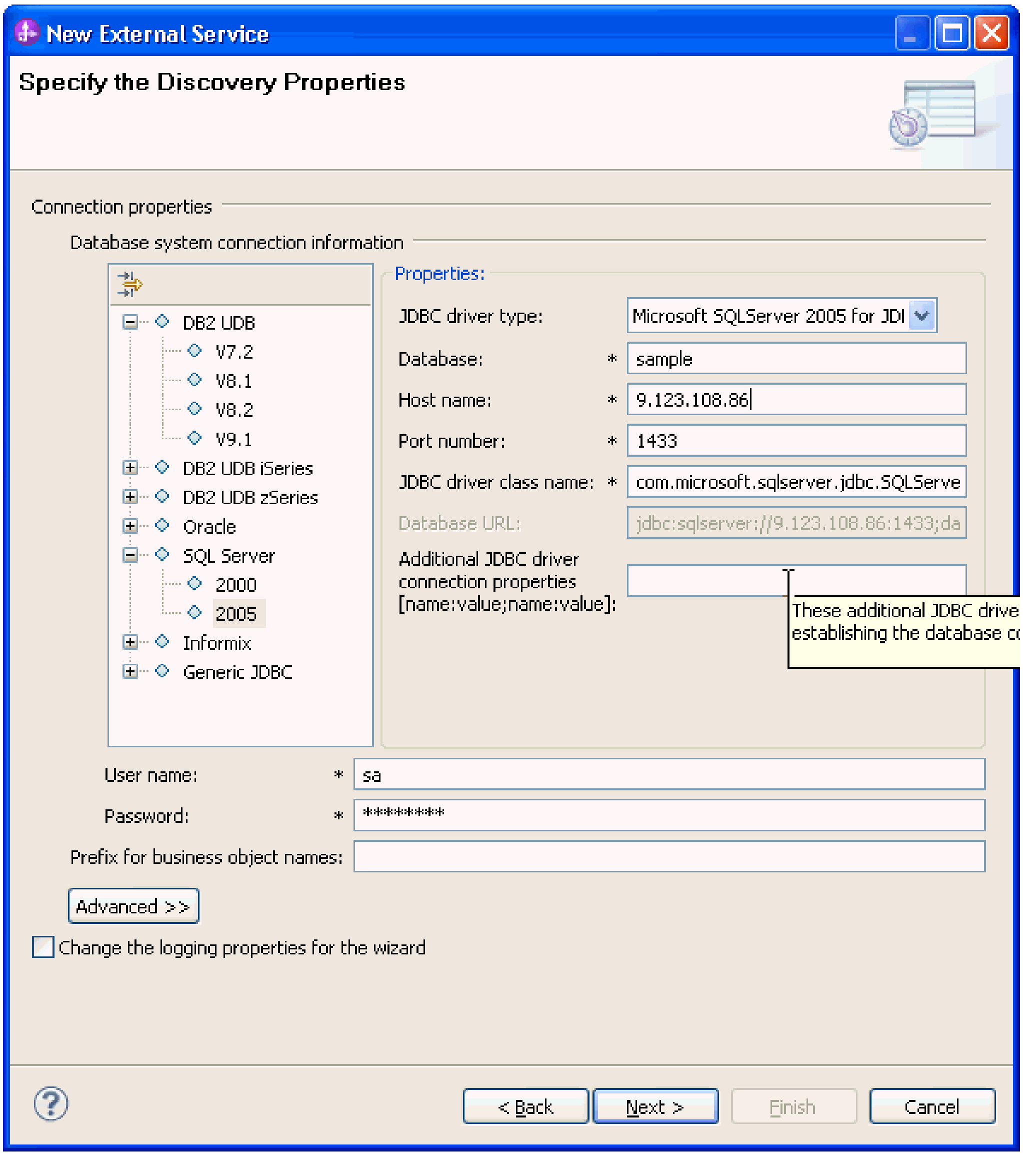
● Outbound

Outbound processing passes data from your service import to the adapter.

Set connection properties for the external service wizard

To connect to the SQL Server:

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**.



JDBC driver type: Microsoft SQLServer 2005 for JDI

Database: * sample

Host name: * 9.123.108.86

Port number: * 1433

JDBC driver class name: * com.microsoft.sqlserver.jdbc.SQLServe

Database URL: jdbc:sqlserver://9.123.108.86:1433;da

Additional JDBC driver connection properties [name:value;name:value]:

These additional JDBC drive
establishing the database c

User name: * sa

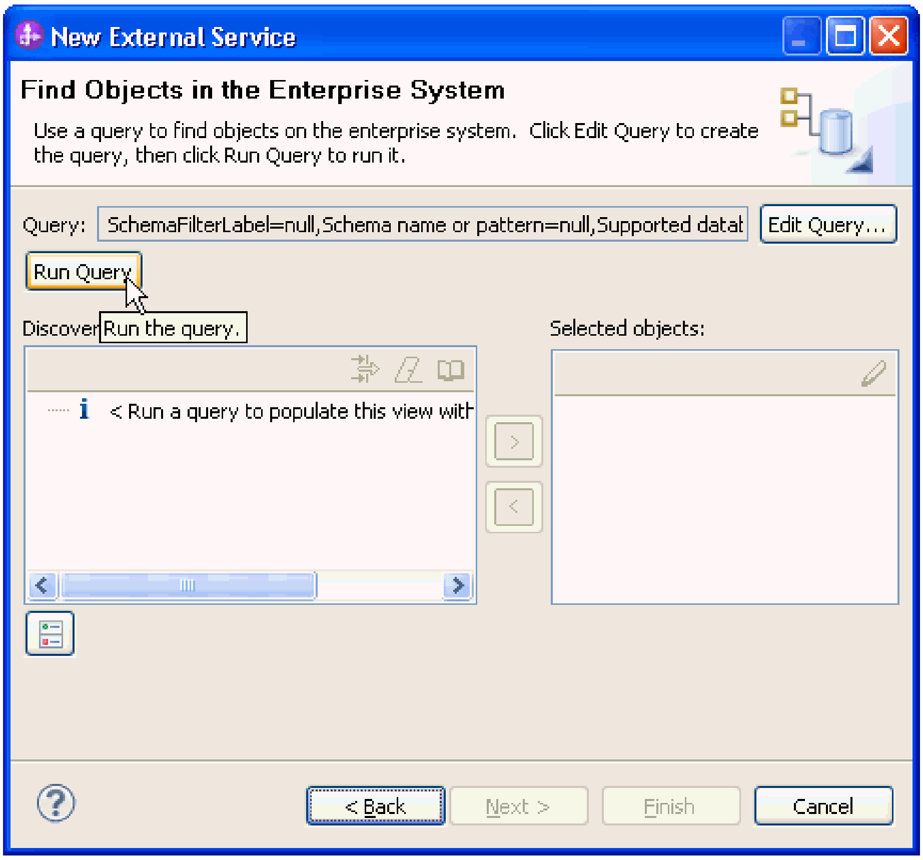
Password: * *********

Prefix for business object names:

□ Change the logging properties for the wizard

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

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



New External Service

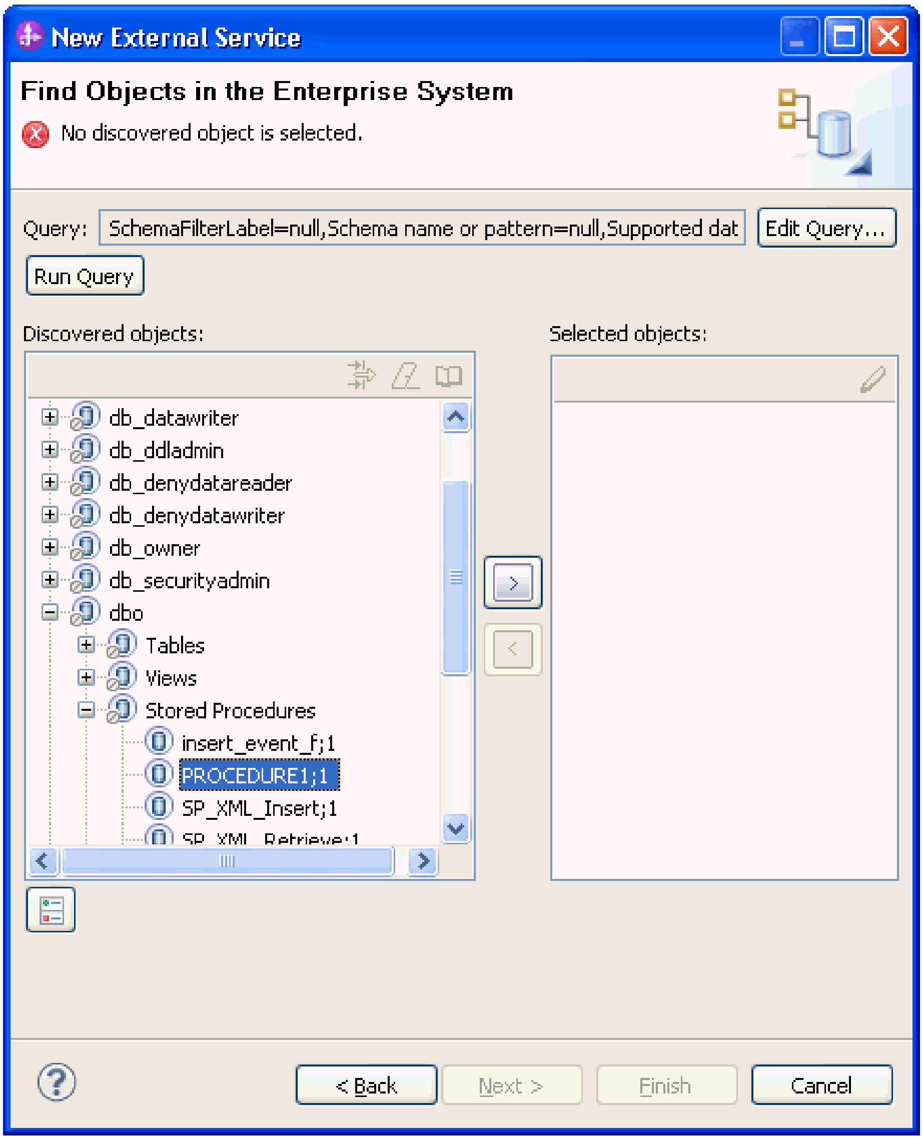
Find Objects in the Enterprise System

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 datat

2. Expand the **dbo** (for this tutorial only) node and select **Stored Procedures** and expand it.

3. Select **PROCEDURE1;1** and click >.



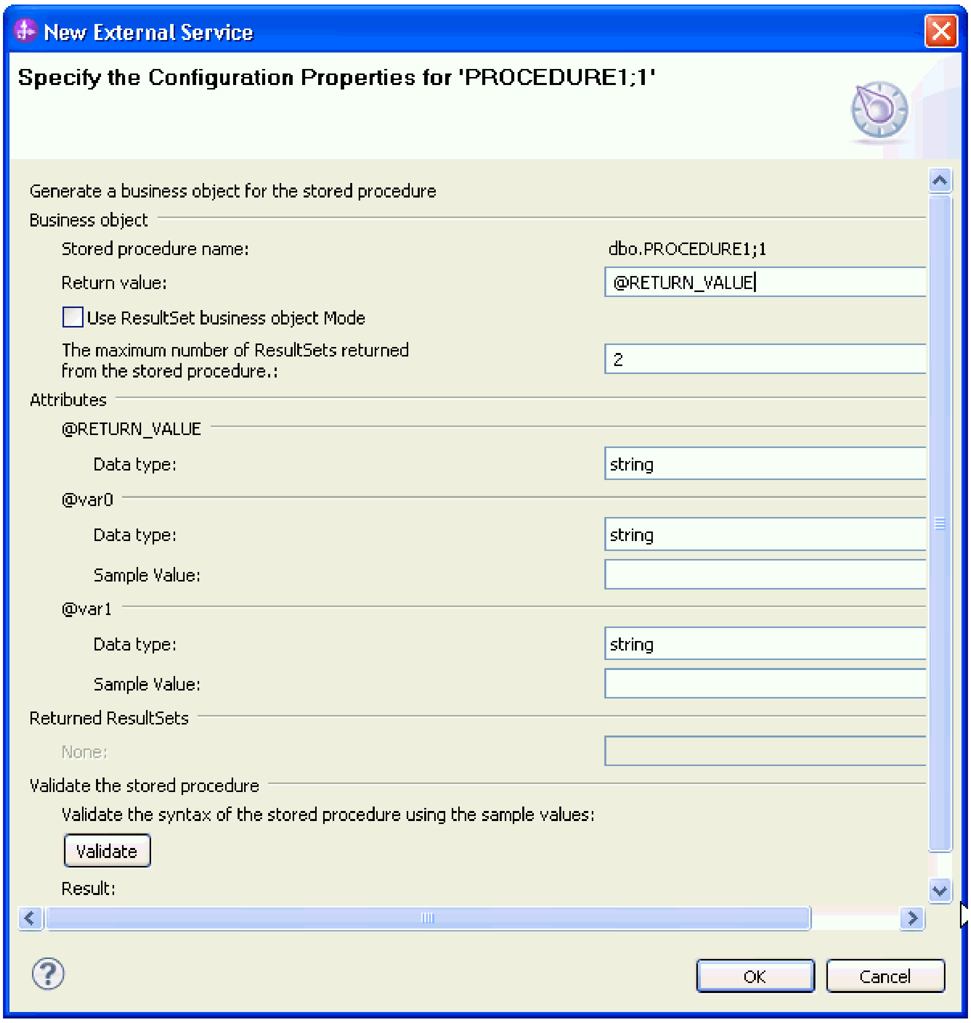
New External Service

Find Objects in the Enterprise System

No discovered object is selected.

Query: SchemaFilterLabel=null,Schema name or pattern=null,Supported dat

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: string

@var0

Data type: string

Sample Value:

@var1

Data type: string

Sample Value:

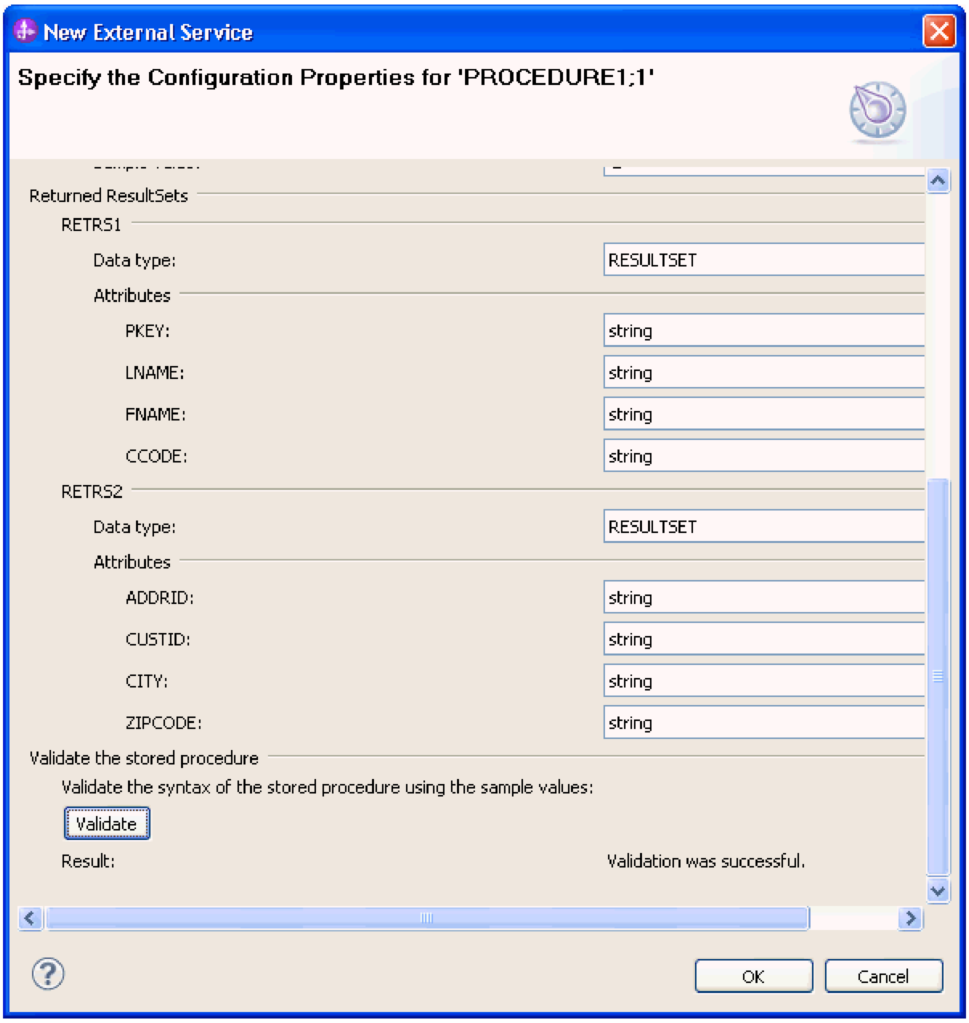
Returned ResultSets

None:

Validate the stored procedure

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

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**.



Specify the Configuration Properties for 'PROCEDURE1;1'

Returned ResultSets

RETRS1

Data type: RESULTSET

Attributes

PKEY: string

LNAME: string

FNAME: string

CCODE: string

RETRS2

Data type: RESULTSET

Attributes

ADDRID: string

CUSTID: string

CITY: string

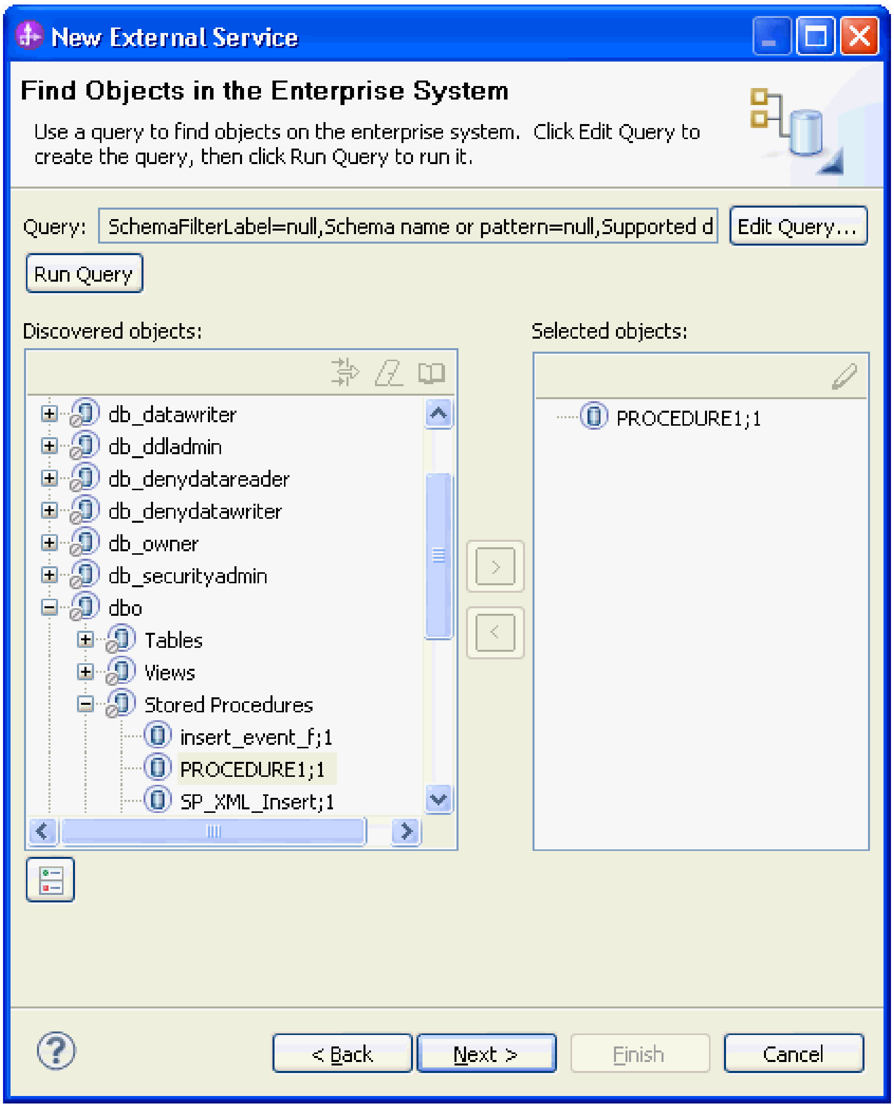
ZIPCODE: string

Validate the stored procedure

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

Result: Validation was successful.

6. Make sure procedure1 is added to the selected objects list and click **Next**.



New External Service

Find Objects in the Enterprise System

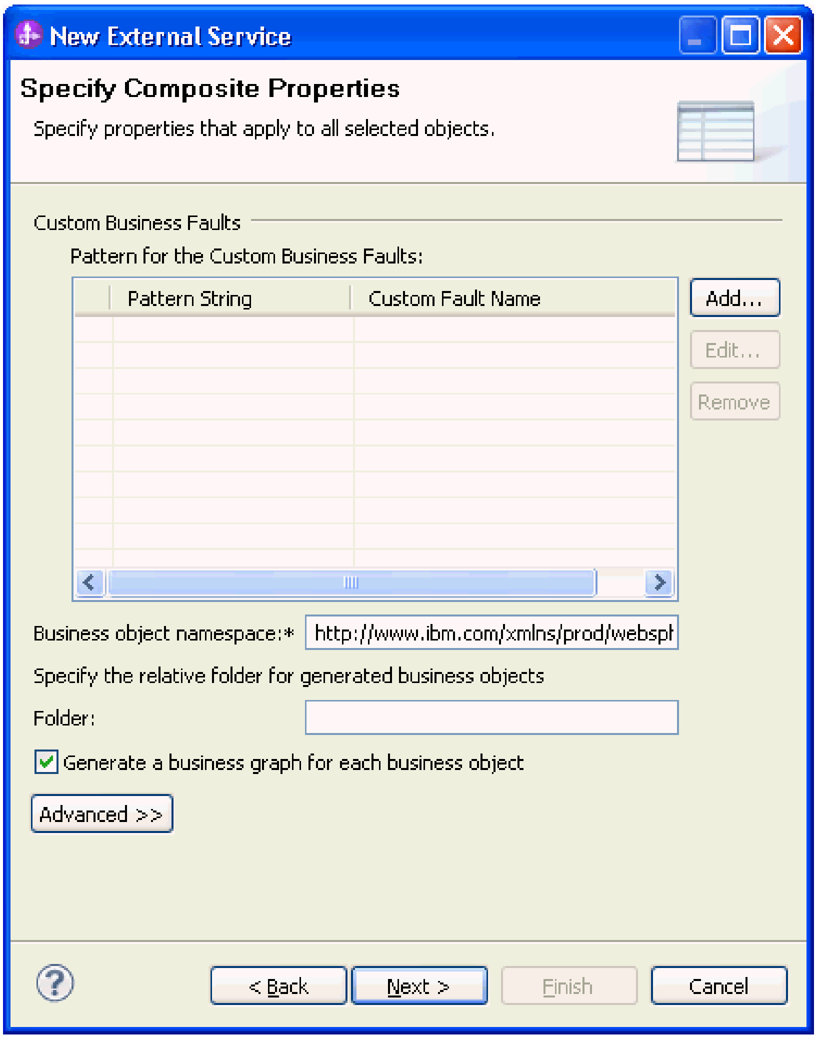
Use a query to find objects on the enterprise system. Click Edit Query to create the query, then click Run Query to run it.

Query: SchemaFilterLabel=null,Schema name or pattern=null,Supported d

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**.



New External Service

Specify Composite Properties

Specify properties that apply to all selected objects.

Custom Business Faults

Pattern for the Custom Business Faults:

Pattern String	Custom Fault Name

Business object namespace:* http://www.ibm.com/xmlns/prod/websph

Specify the relative folder for generated business objects

Folder:

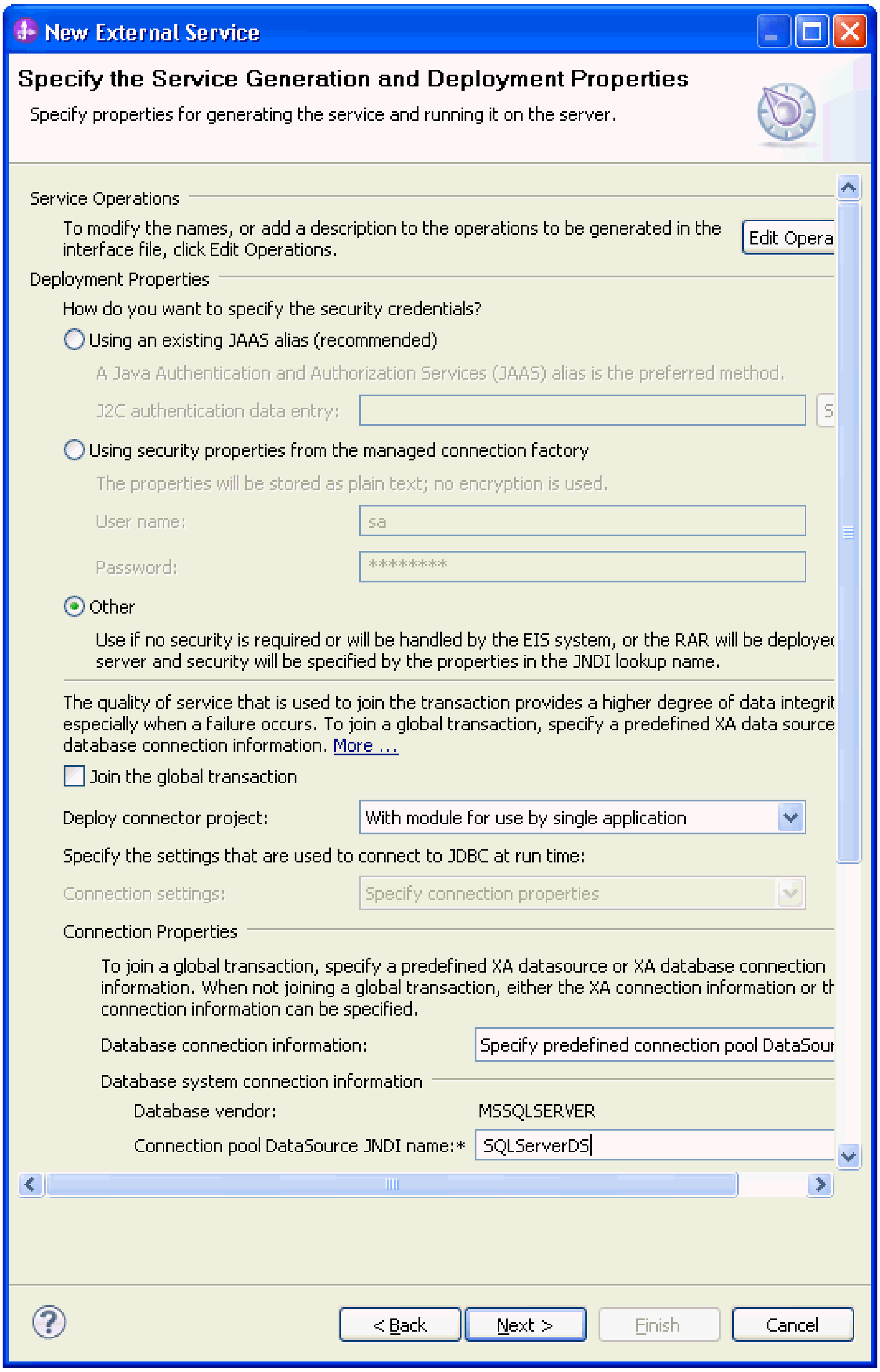
☑ Generate a business graph for each business object

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.

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: sa

Password: *********

● Other

Use if no security is required or will be handled by the EIS system, or the RAR will be deployec 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 integrit especially when a failure occurs. To join a global transaction, specify a predefined XA data source database connection information. More ...

□ Join the global transaction

Deploy connector project: With module for use by single application

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

Connection settings: Specify connection properties

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 th connection information can be specified.

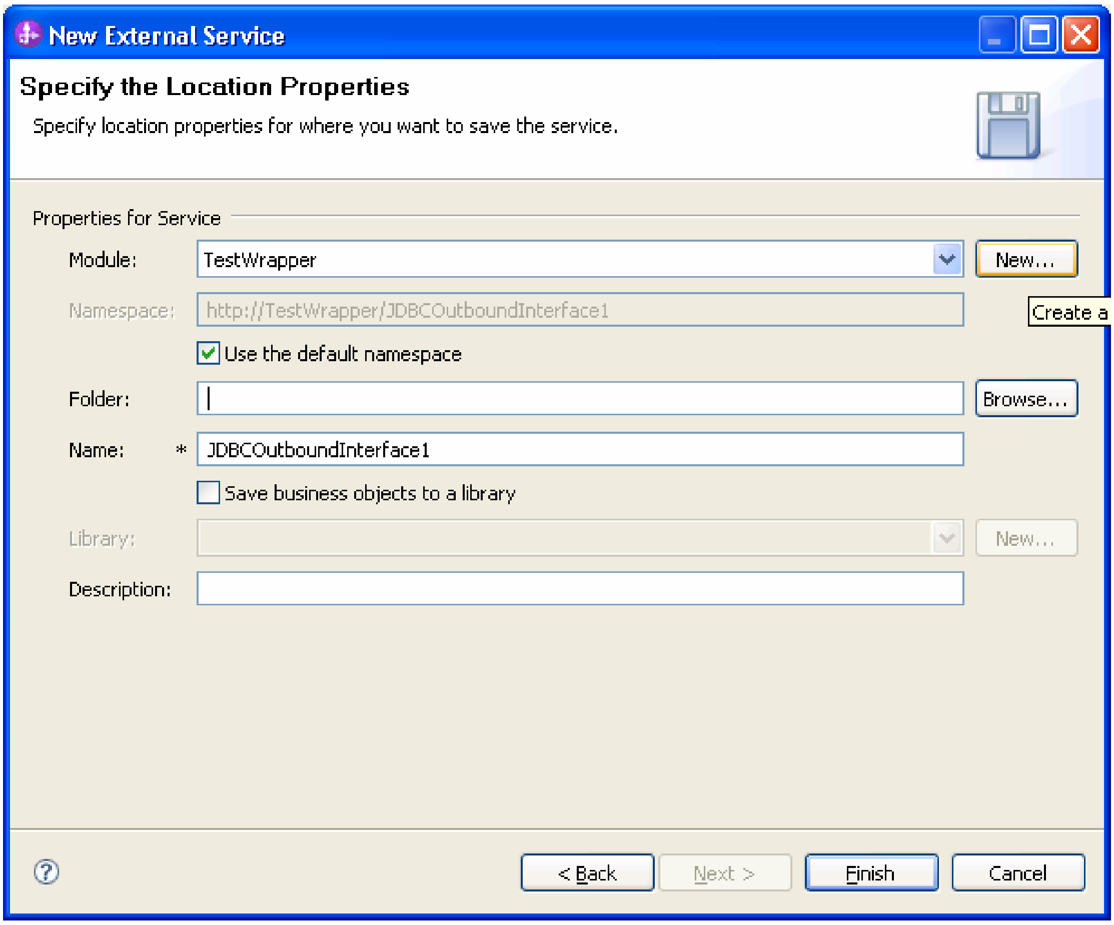
Database connection information: Specify predefined connection pool DataSour

Database system connection information

Database vendor: MSSQLSERVER

Connection pool DataSource JNDI name:* SQLServerDS

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



New External Service

Specify the Location Properties

Specify location properties for where you want to save the service.

Properties for Service

Module: TestWrapper

Namespace: http://TestWrapper/JDBCOutboundInterface1

☑ Use the default namespace

Folder:

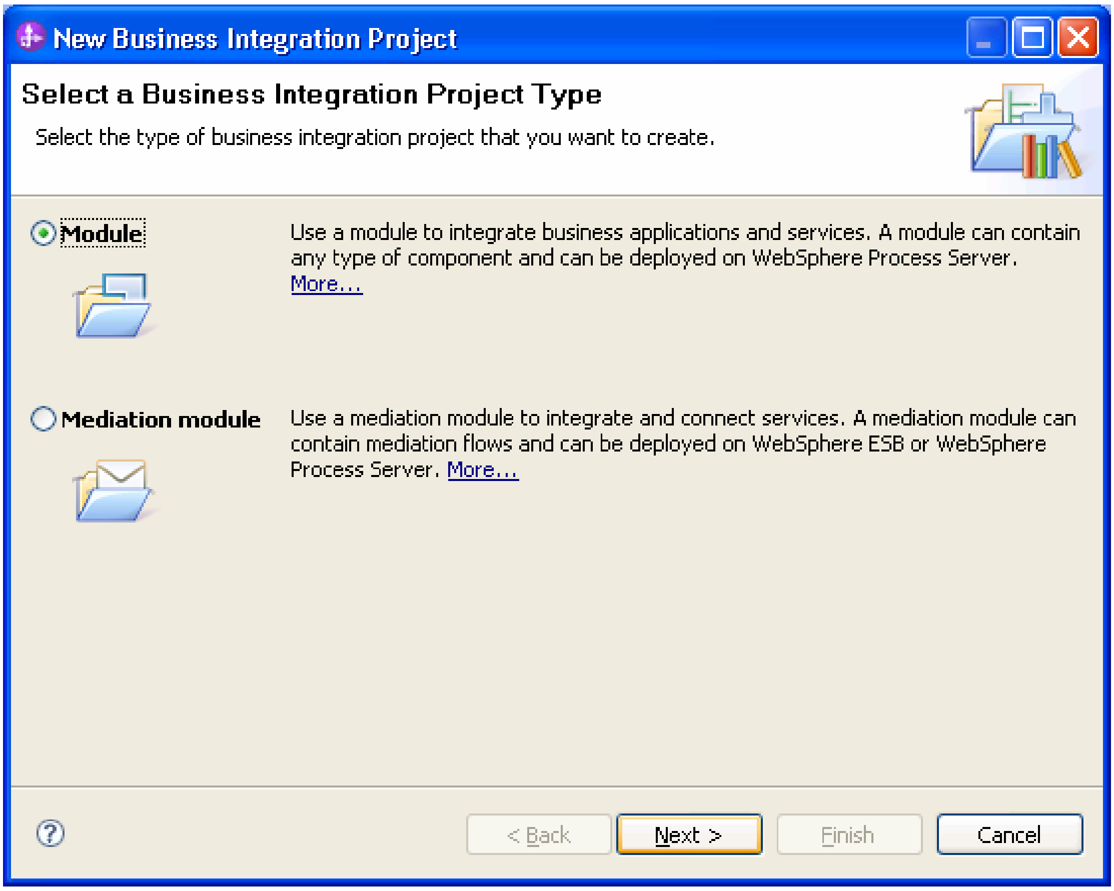
Name: * JDBCOutboundInterface1

□ Save business objects to a library

Library:

Description:

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



New Business Integration Project

Select a Business Integration Project Type

Select the type of business integration project that you want to create.

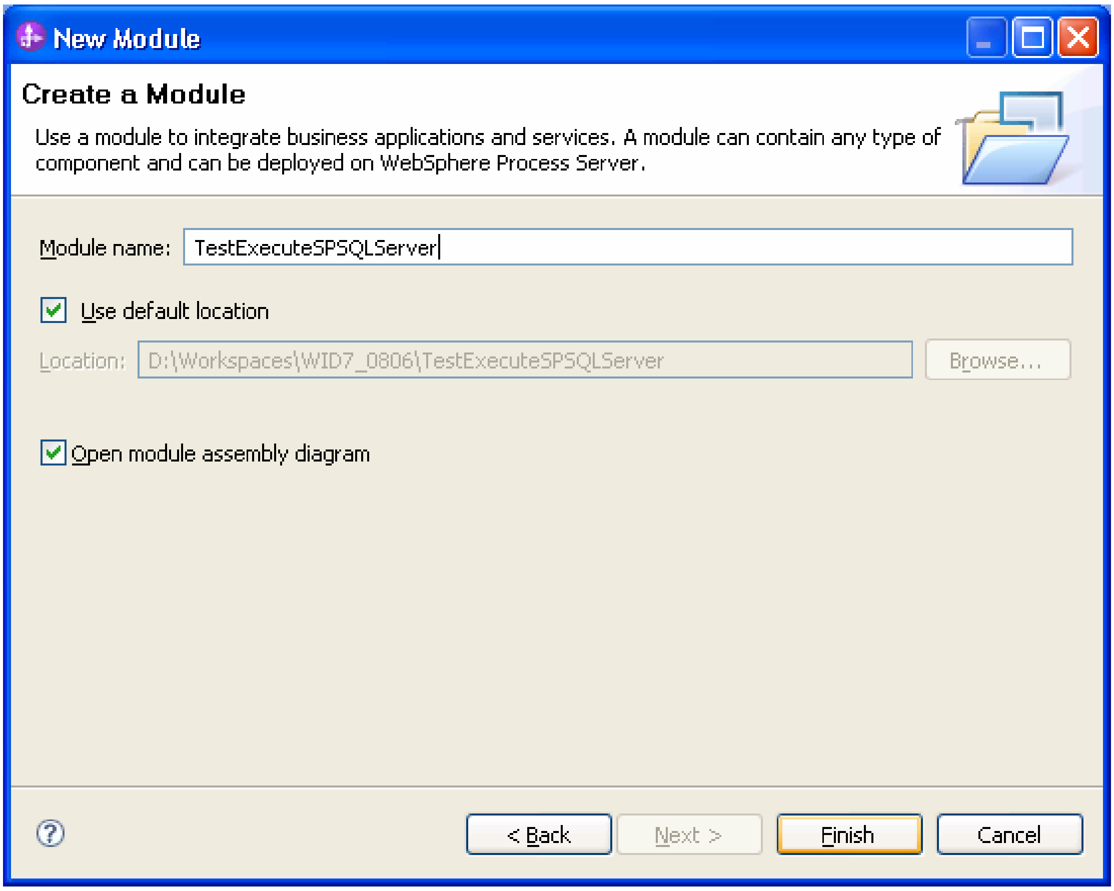
● Module

Use a module to integrate business applications and services. A module can contain any type of component and can be deployed on WebSphere Process Server. More...

○ Mediation module

Use a mediation module to integrate and connect services. A mediation module can contain mediation flows and can be deployed on WebSphere ESB or WebSphere Process Server. More...

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



Create a Module

Use a module to integrate business applications and services. A module can contain any type of component and can be deployed on WebSphere Process Server.

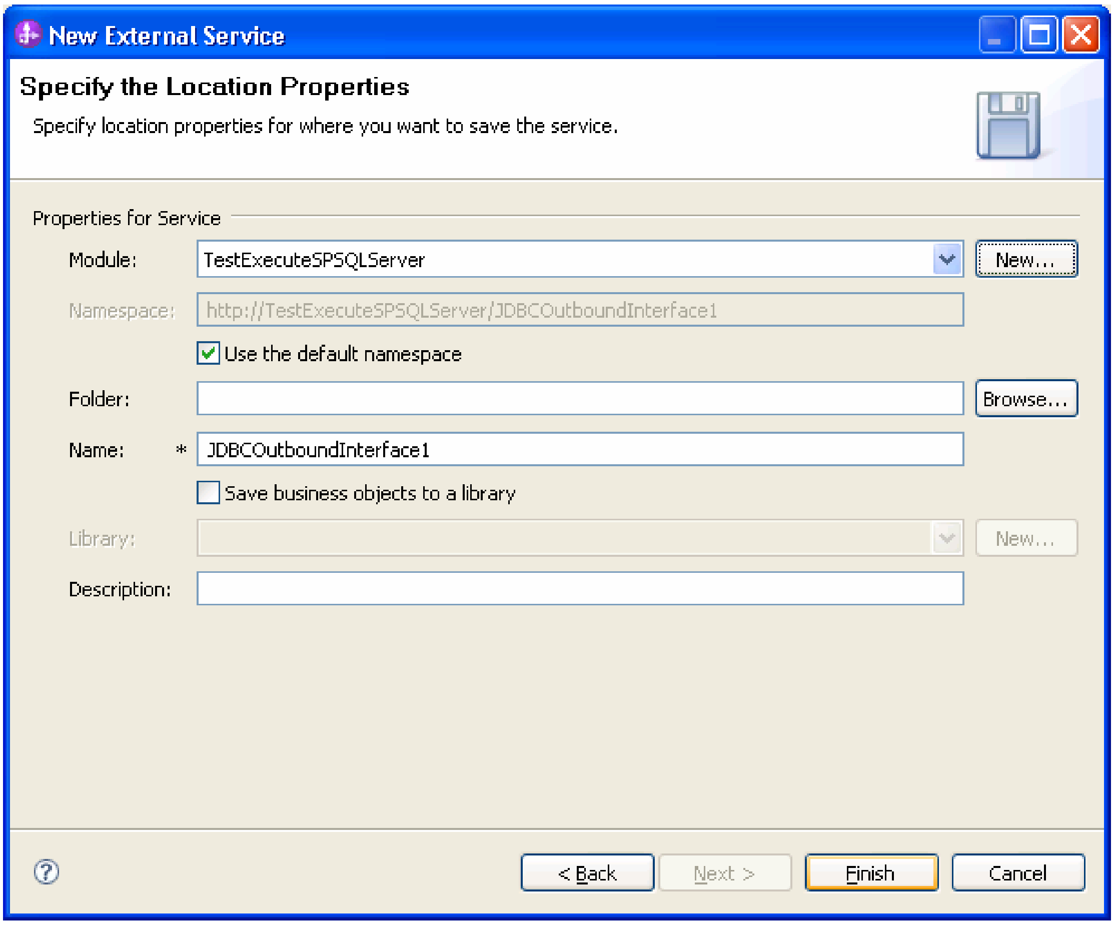
Module name: TestExecuteSPSQLServer

☑ Use default location

Location: D:\Workspaces\WID7_0806\TestExecuteSPSQLServer

☑ Open module assembly diagram

6. Accept the default values and click **Finish**.



New External Service

Specify the Location Properties

Specify location properties for where you want to save the service.

Properties for Service

Module: TestExecuteSPSQLServer

Namespace: http://TestExecuteSPSQLServer/JDBCOutboundInterface1

☑ Use the default namespace

Folder:

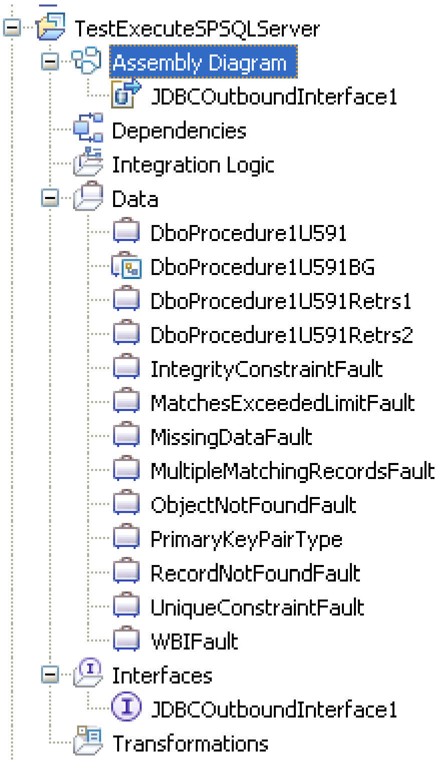
Name: * JDBCOutboundInterface1

□ Save business objects to a library

Library:

Description:

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



```
TestExecuteSPSQLServer
  Assembly Diagram
    JDBCOutboundInterface1
  Dependencies
  Integration Logic
  Data
    DboProcedure1U591
    DboProcedure1U591BG
    DboProcedure1U591Retrs1
    DboProcedure1U591Retrs2
    IntegrityConstraintFault
    MatchesExceededLimitFault
    MissingDataFault
    MultipleMatchingRecordsFault
    ObjectNotFoundFault
    PrimaryKeyPairType
    RecordNotFoundFault
    UniqueConstraintFault
    WBIFault
  Interfaces
    JDBCOutboundInterface1
  Transformations
```

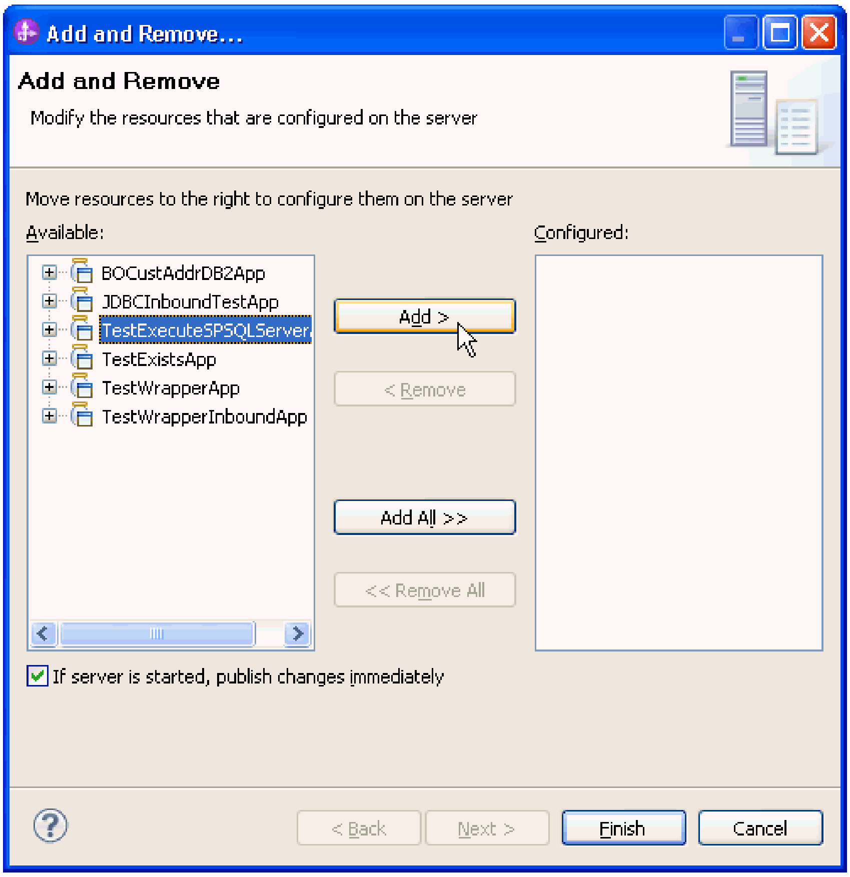
Deploy the module to the test environment

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 IBM Integration Designer integration test client. To do this, you must add the SCA module you created earlier to the server using the **Servers** view in IBM Integration Designer.

Steps for adding the SCA module to the server:

1. In IBM Integration Designer, 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**.

The Add and Remove Projects window lists the available projects in the IBM Integration Designer workspace.



Add and Remove...

Add and Remove

Modify the resources that are configured on the server

Move resources to the right to configure them on the server

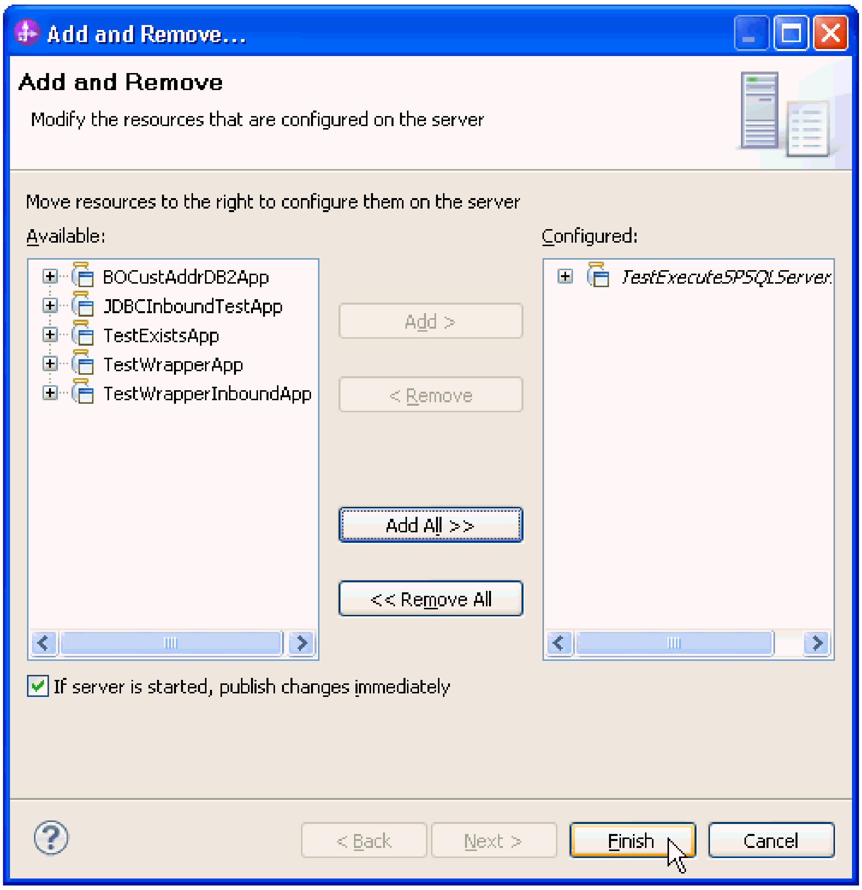
Available:

- BOCustAddrDB2App
- JDBCInboundTestApp
- TestExecuteSPSQLServer
- TestExistsApp
- TestWrapperApp
- TestWrapperInboundApp

Configured:

☑ If server is started, publish changes immediately

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



Add and Remove...

Add and Remove

Modify the resources that are configured on the server

Move resources to the right to configure them on the server

Available:

- BOCustAddrDB2App
- JDBCInboundTestApp
- TestExistsApp
- TestWrapperApp
- TestWrapperInboundApp

Configured:

- *TestExecuteSPSQLServer*

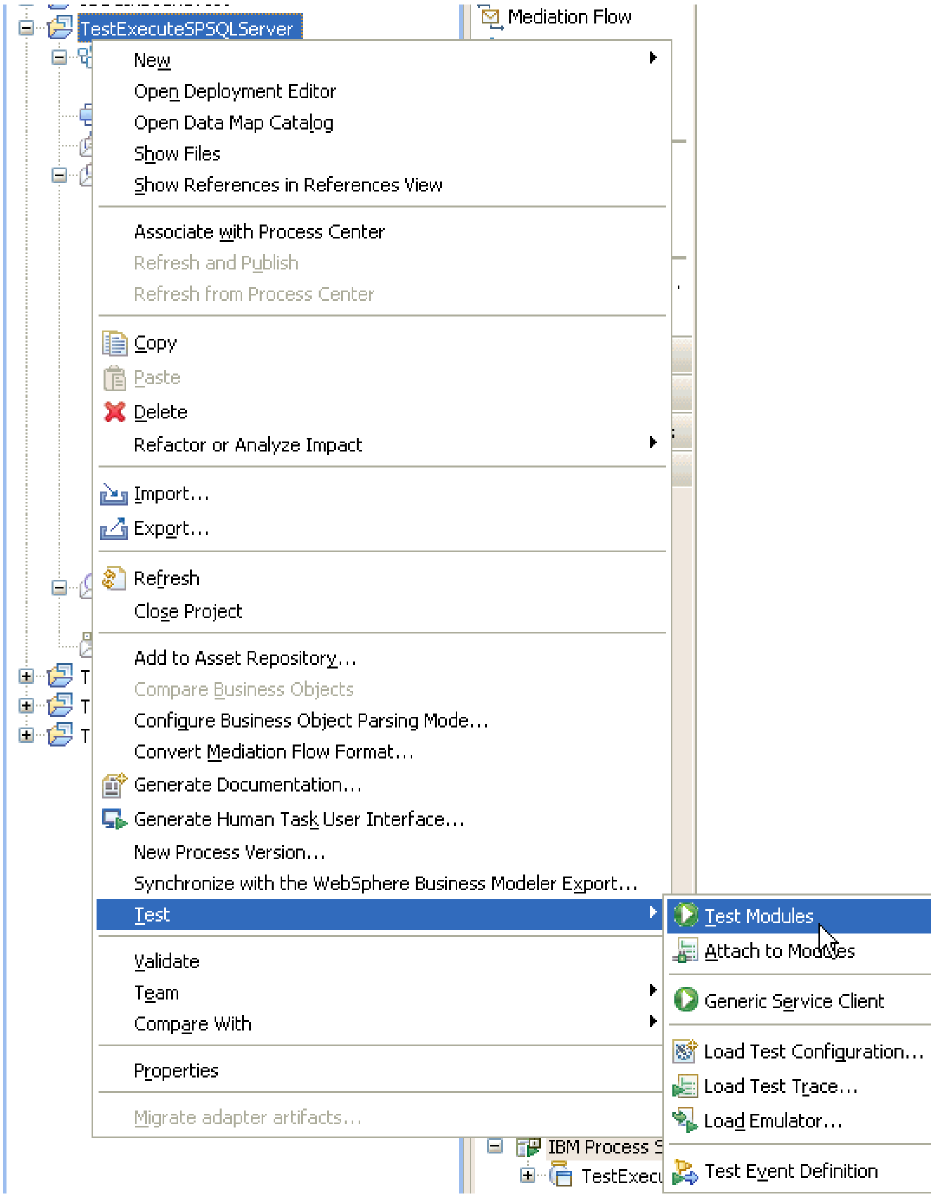
☑ If server is started, publish changes immediately

5. Click **Finish**.

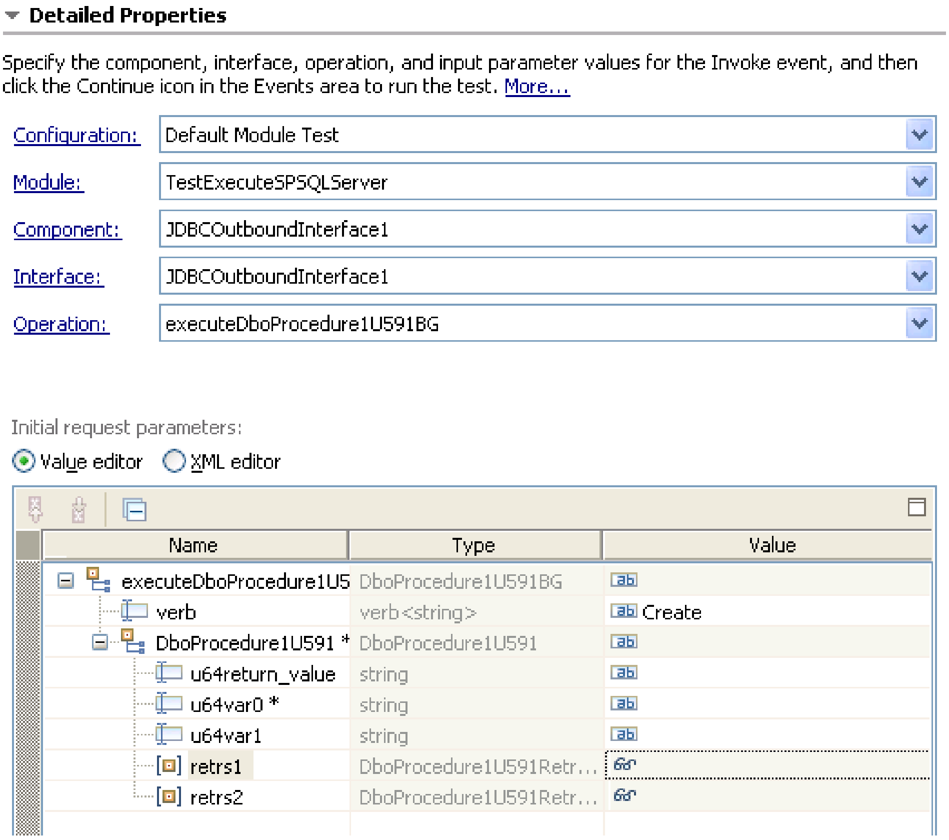
Test the assembled adapter application

Test the assembled adapter application using the IBM Integration Designer integration test client.

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


TestExecuteSPSQLServer
Mediation Flow
New
Open Deployment Editor
Open Data Map Catalog
Show Files
Show References in References View
Associate with Process Center
Refresh and Publish
Refresh from Process Center
Copy
Paste
Delete
Refactor or Analyze Impact
Import...
Export...
Refresh
Close Project
Add to Asset Repository...
Compare Business Objects
Configure Business Object Parsing Mode...
Convert Mediation Flow Format...
Generate Documentation...
Generate Human Task User Interface...
New Process Version...
Synchronize with the WebSphere Business Modeler Export...
Test
Validate
Team
Compare With
Properties
Migrate adapter artifacts...
Test Modules
Attach to Modules
Generic Service Client
Load Test Configuration...
Load Test Trace...
Load Emulator...
Test Event Definition
IBM Process S
TestExecu

2. Select **executeDboProcedure1U591BG** operation.



Detailed Properties

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

Configuration:	Default Module Test
Module:	TestExecuteSPSQLServer
Component:	JDBCOutboundInterface1
Interface:	JDBCOutboundInterface1
Operation:	executeDboProcedure1U591BG

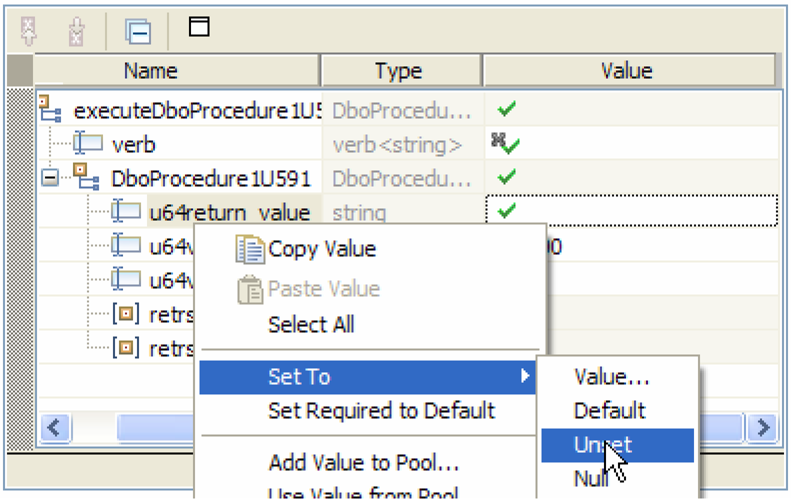
Initial request parameters:

● Value editor ○ XML editor

Name	Type	Value
executeDboProcedure1U5	DboProcedure1U591BG	ab
verb	verb<string>	ab Create
DboProcedure1U591 *	DboProcedure1U591	ab
u64return_value	string	ab
u64var0 *	string	ab
u64var1	string	ab
retrs1	DboProcedure1U591Retr...	60
retrs2	DboProcedure1U591Retr...	60

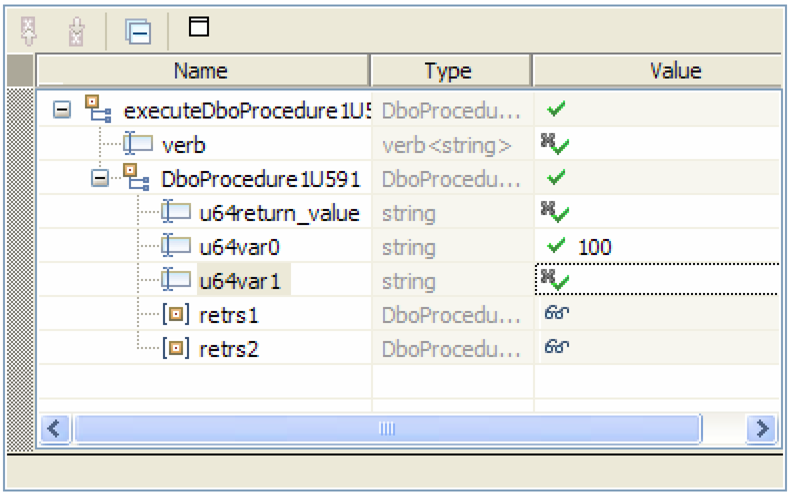
3. Enter value for the input type **var0**.

4. Unset the value for the output type **var1** and **return_value.** Right-click **u64return_value,** and select **Set To** > **Unset**.



Name	Type	Value
executeDboProcedure1U5	DboProcedu...	✓
verb	verb<string>	✓
DboProcedure1U591	DboProcedu...	✓
u64return_value	string	✓
u64v		0
u64v		
retrs		
retrs		

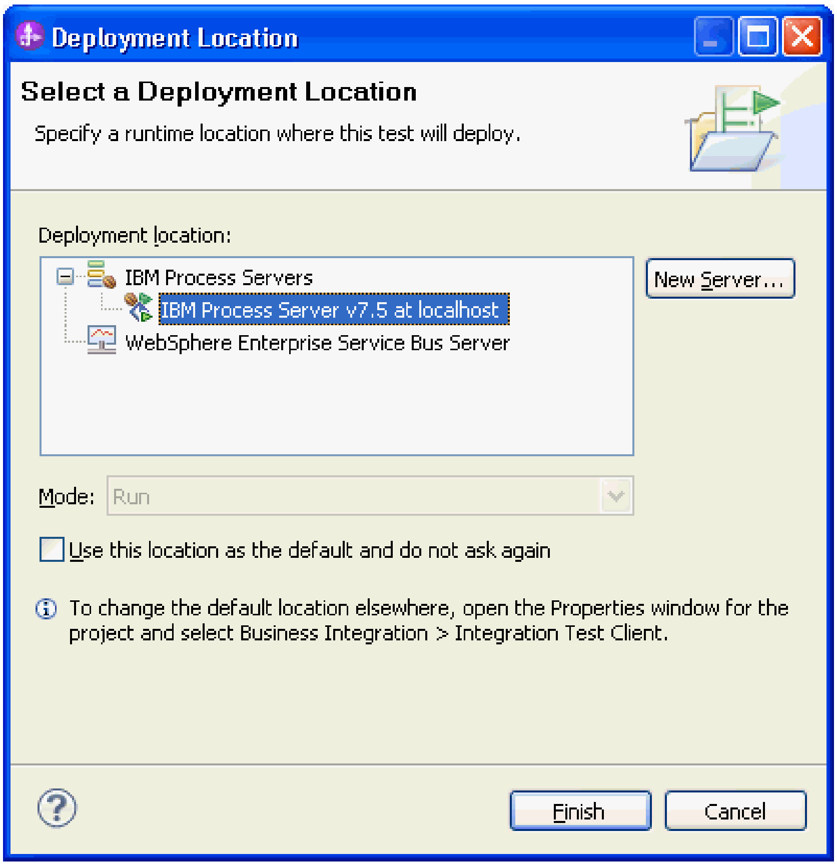
An unset field is indicated by a '**X**' mark.



Name	Type	Value
executeDboProcedure1U5	DboProcedu...	✓
verb	verb<string>	✓
DboProcedure1U591	DboProcedu...	✓
u64return_value	string	✓
u64var0	string	✓ 100
u64var1	string	✓
retrs1	DboProcedu...	
retrs2	DboProcedu...	

5. To execute the service, click ▶.

6. In the Select Deployment location window, select the server, and click **Finish**.



Select a Deployment Location

Specify a runtime location where this test will deploy.

Deployment location:

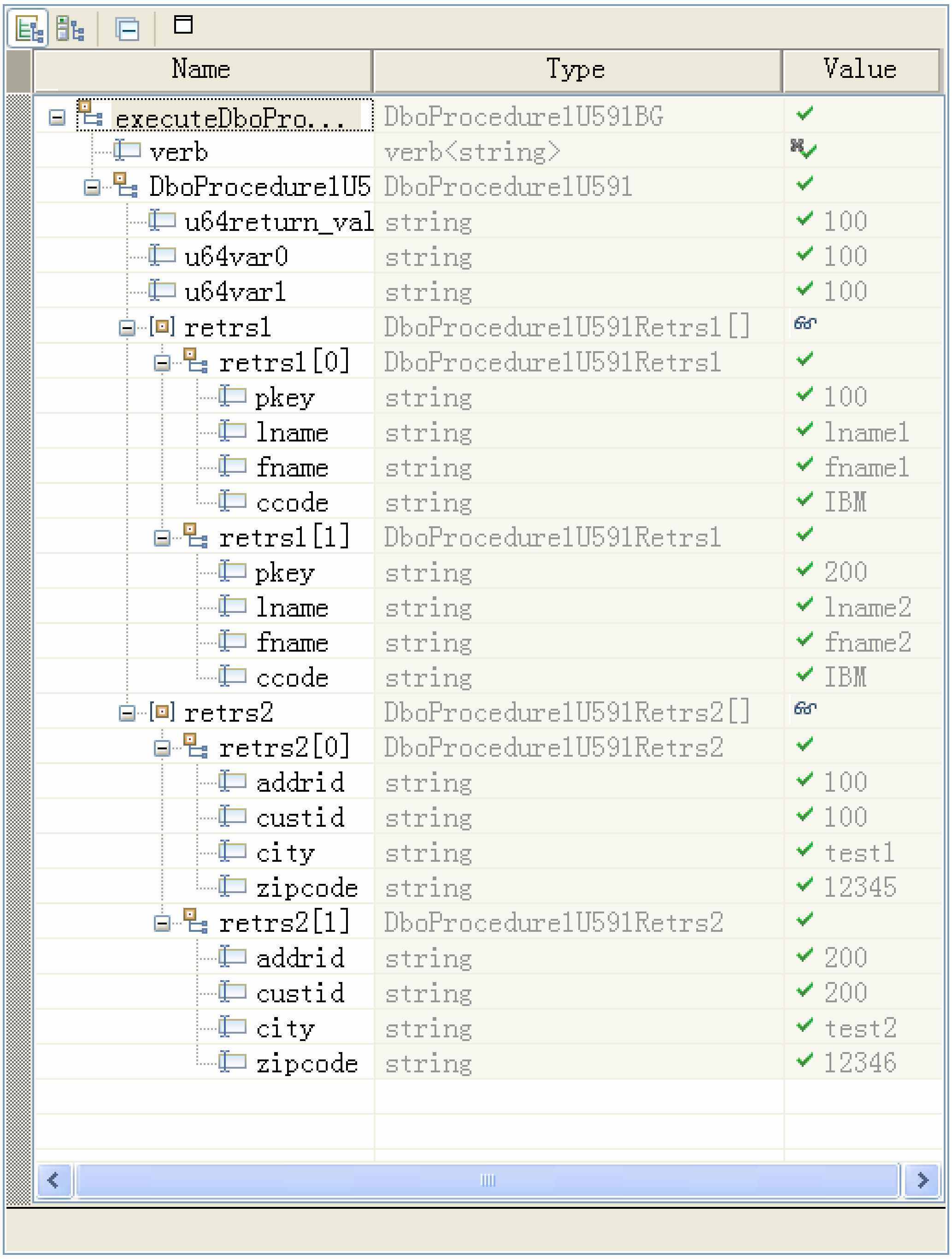
- IBM Process Servers
  - IBM Process Server v7.5 at localhost
- WebSphere Enterprise Service Bus Server

Mode: Run

□ Use this location as the default and do not ask again

To change the default location elsewhere, open the Properties window for the project and select Business Integration > Integration Test Client.

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.

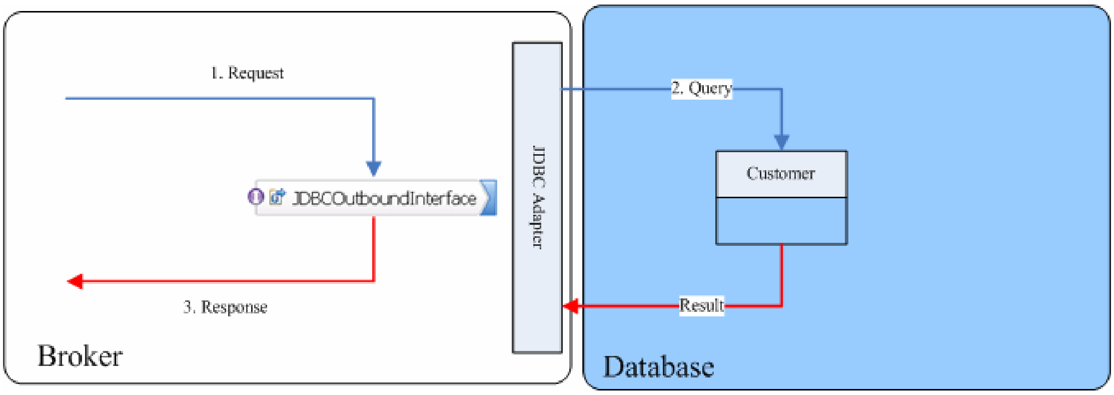
Nothing is required to clean up after this tutorial.

Chapter 13. Tutorial 12: Retrieve business object from database using user defined query (DB2)

This tutorial demonstrates how WebSphere Adapter for JDBC 7.5.0.0 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

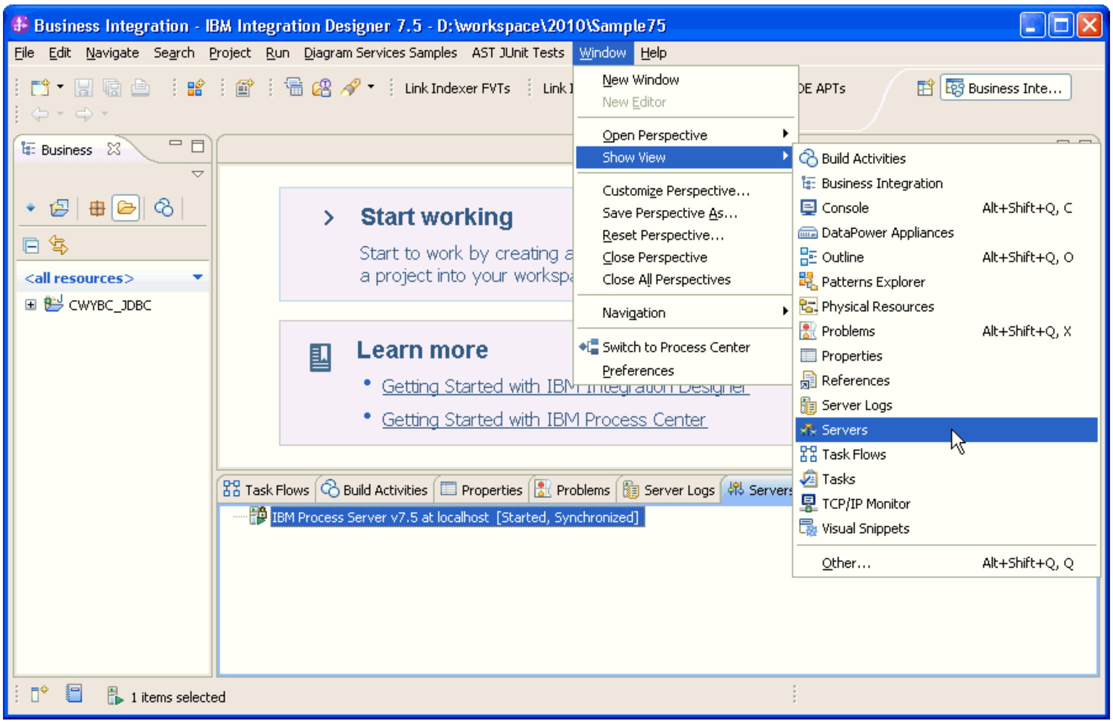
```
CREATE TABLE CUSTOMER  (
        PKEY VARCHAR(10) NOT NULL,
        FNAME VARCHAR(20) ,
        LNAME VARCHAR(20) ,
        CCODE VARCHAR(10) ,
                       PRIMARY KEY(PKEY));


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

Create an authentication alias

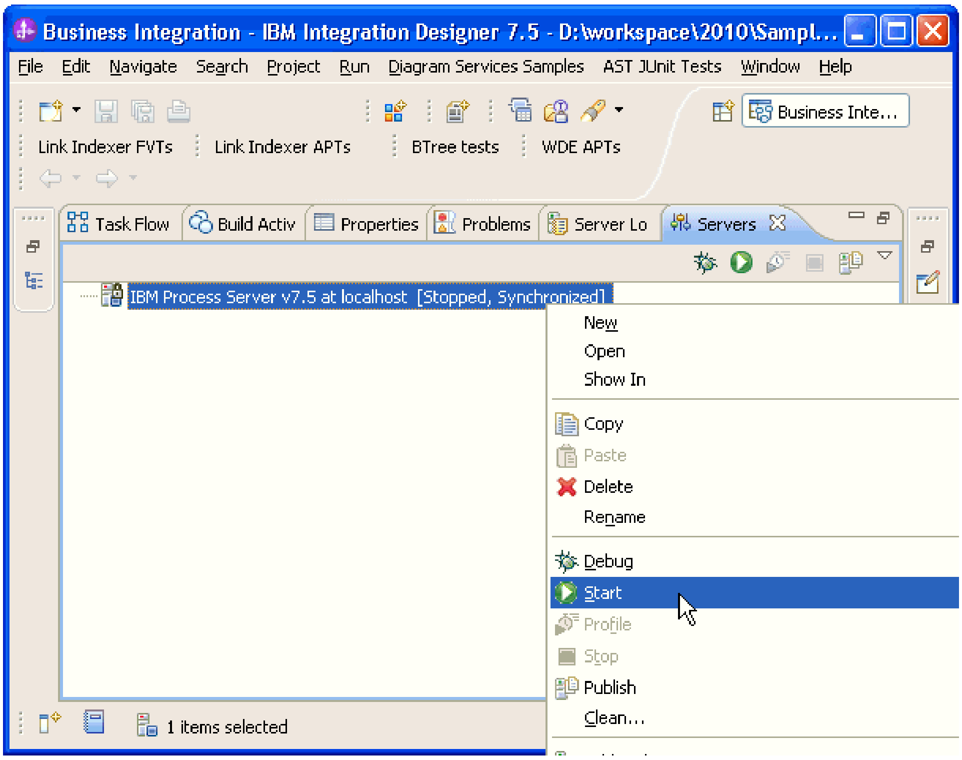
The authentication alias needs to be set because the adapter uses the username and password to connect to the database. This authentication alias will be used later when generating the artifacts for the module.

Here are the steps to set the authentication alias in IBM Process Server administration console.

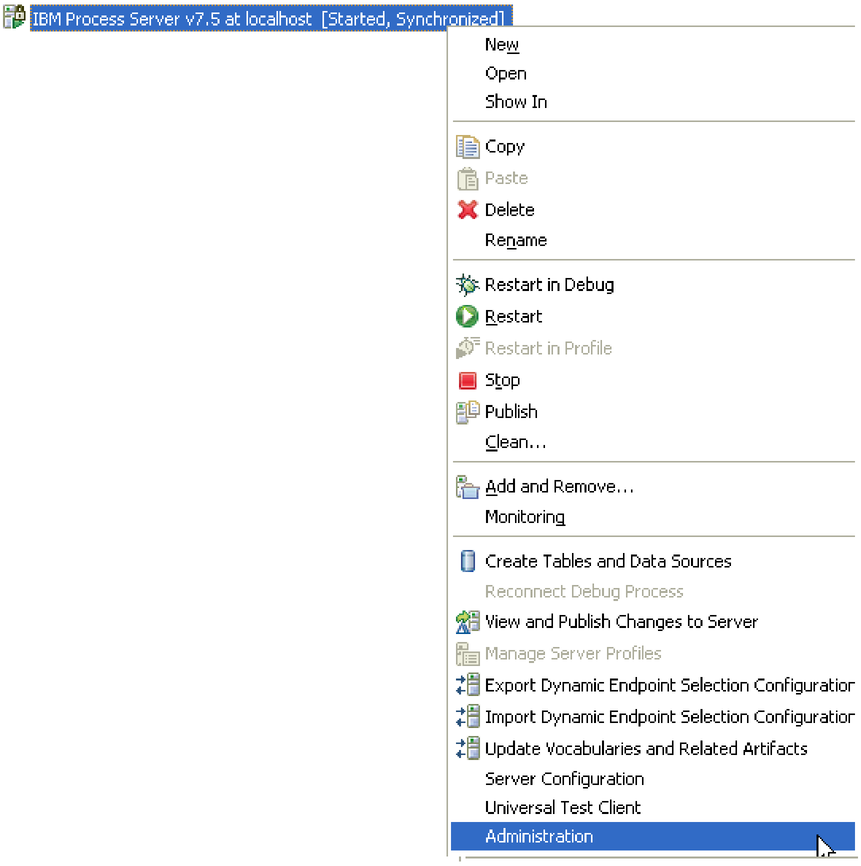
1. In IBM Integration Designer, switch to the **Servers** view by selecting **Window > Show View > Servers**.



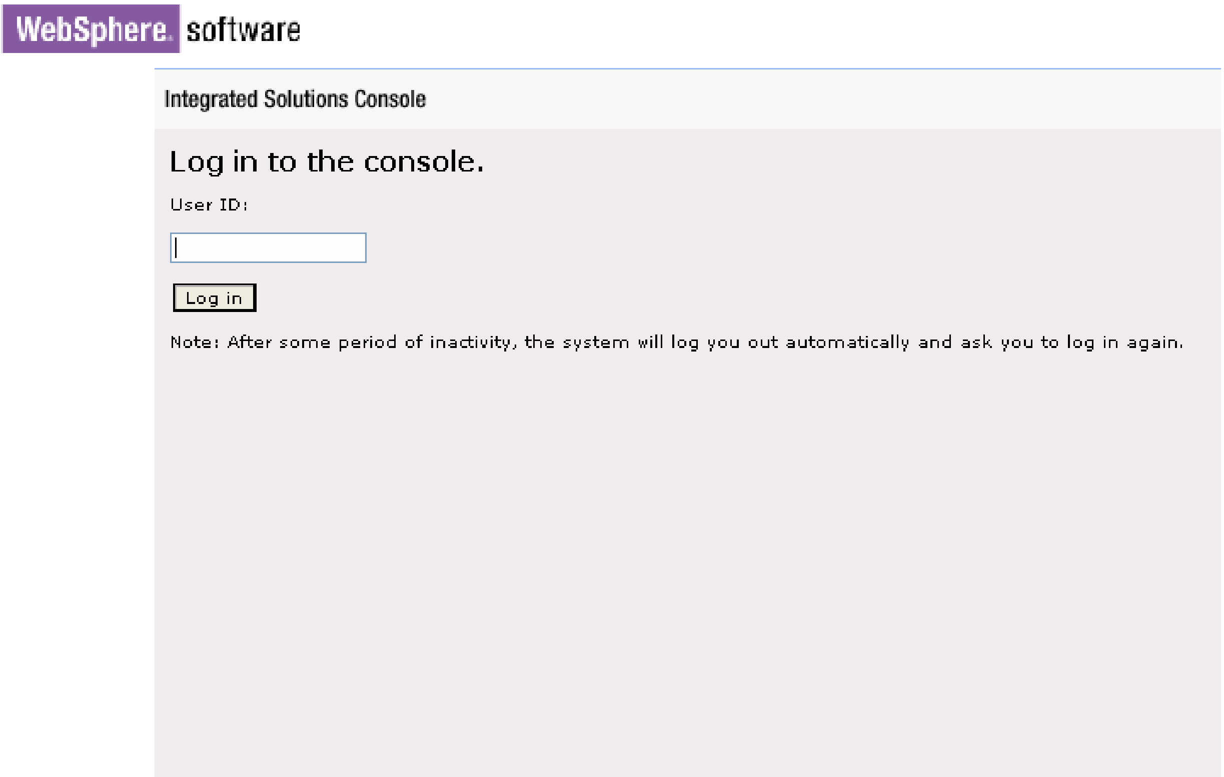
2. In the **Servers** view, right-click the server that you want to start and select **Start**.


Business Integration - IBM Integration Designer 7.5 - D:\workspace\2010\Sampl...
File Edit Navigate Search Project Run Diagram Services Samples AST JUnit Tests Window Help
Business Inte...
Link Indexer FVTs
Link Indexer APTs
BTree tests
WDE APTs
Task Flow
Build Activ
Properties
Problems
Server Lo
Servers
IBM Process Server v7.5 at localhost [Stopped, Synchronized]
New
Open
Show In
Copy
Paste
Delete
Rename
Debug
Start
Profile
Stop
Publish
Clean...
1 items selected

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


IBM Process Server v7.5 at localhost [Started, Synchronized]
New
Open
Show In
Copy
Paste
Delete
Rename
Restart in Debug
Restart
Restart in Profile
Stop
Publish
Clean...
Add and Remove...
Monitoring
Create Tables and Data Sources
Reconnect Debug Process
View and Publish Changes to Server
Manage Server Profiles
Export Dynamic Endpoint Selection Configuratior
Import Dynamic Endpoint Selection Configuratior
Update Vocabularies and Related Artifacts
Server Configuration
Universal Test Client
Administration

4. Log on to the administrative console.



WebSphere software

Integrated Solutions Console

Log in to the console.

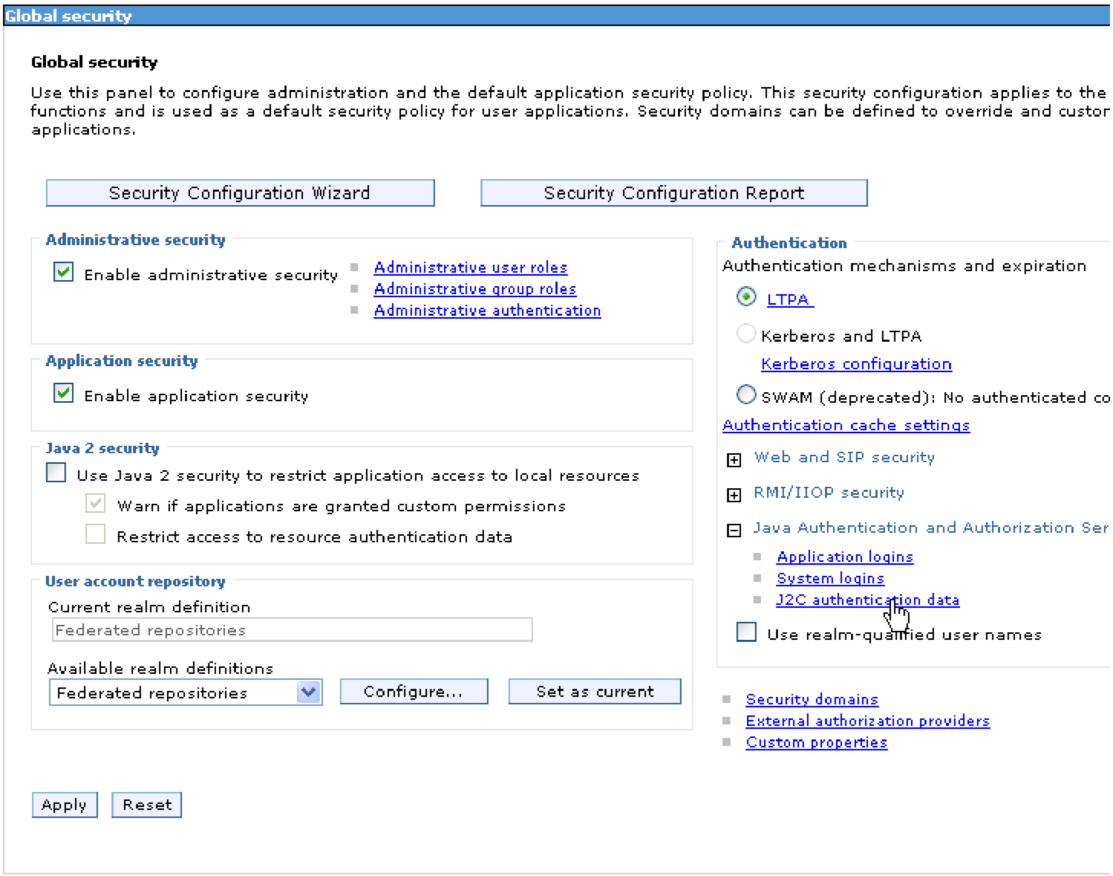
User ID:

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**.



Global security

Global security

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

Authentication cache settings

⊞ Web and SIP security

⊞ RMI/IIOP security

⊟ Java Authentication and Authorization Ser

- Application logins
- System logins
- J2C authentication data

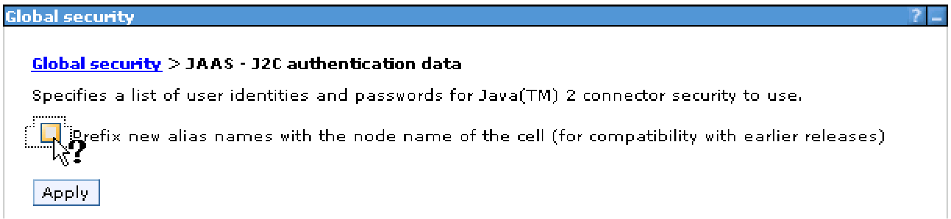
□ Use realm-qualified user names

- Security domains
- External authorization providers
- Custom properties

Apply | Reset

A list of existing aliases is displayed.

7. Uncheck **Prefix new alias names with the node name of the cell(for compatibility with earlier releases)**. And click **Apply** button.

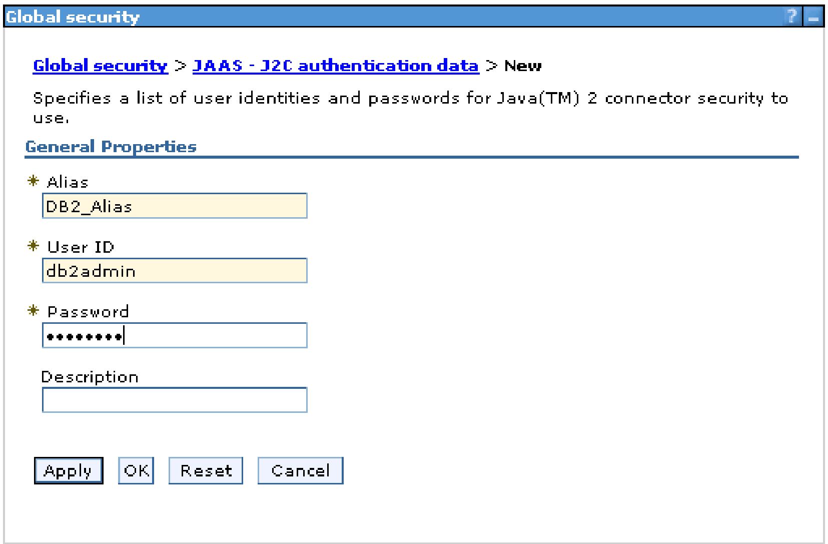


Global security > JAAS - J2C authentication data

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)

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



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

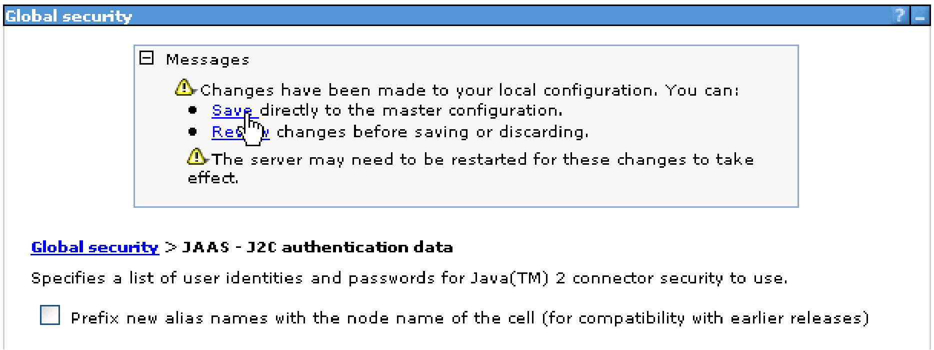
* Alias: DB2_Alias

* User ID: db2admin

* Password: ••••••••

Description:

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



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)

You have created an authentication alias that will be used to configure the adapter properties. Restart the server for the changes to take effect.

Configure the adapter for outbound processing

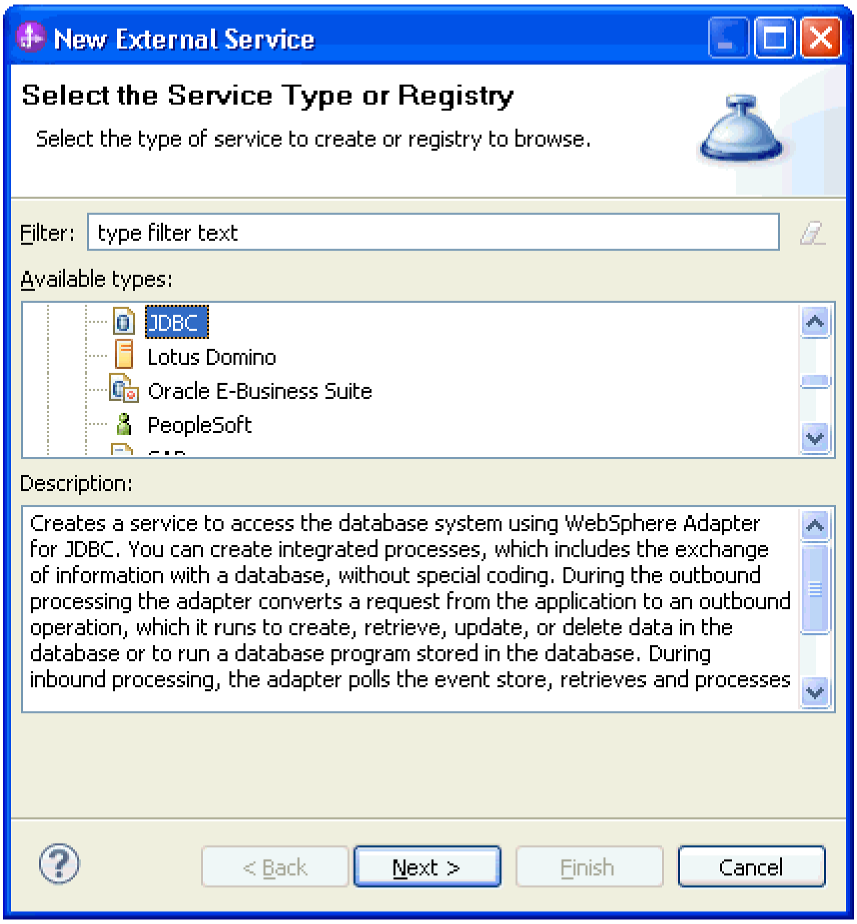
Run the external service wizard to specify business objects, services, and configuration details.

The wizard will guide you to do the following steps:

- Setting connection properties for the enterprise service discovery wizard
- Selecting the business objects and services to be used with the adapter
- Generating business object definitions and related artifacts

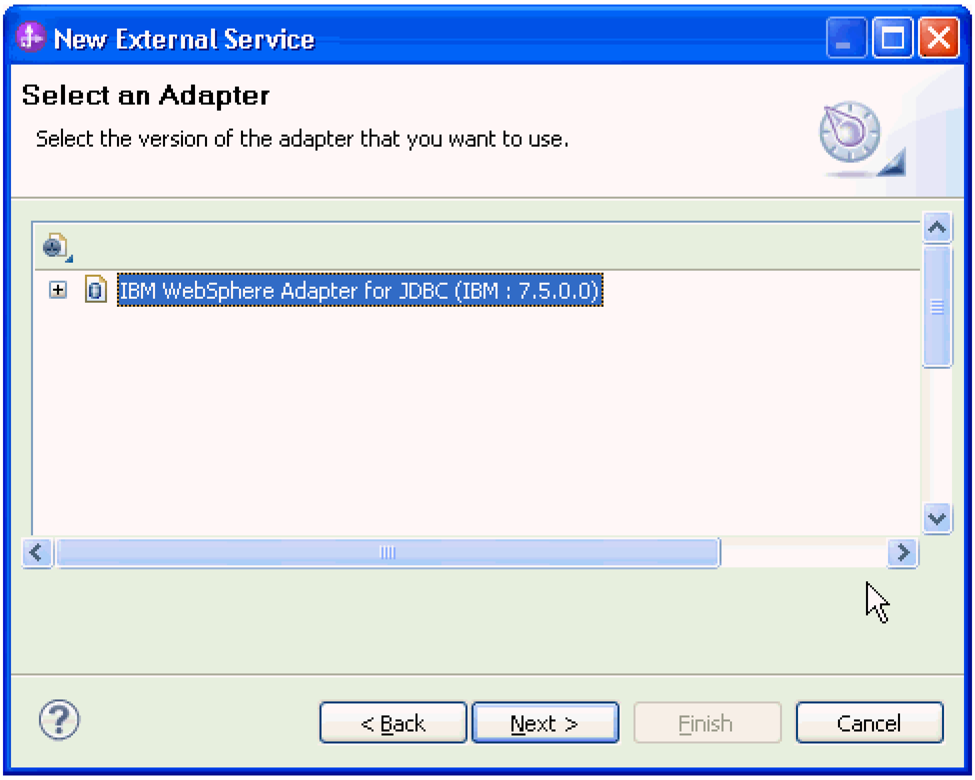
Follow these instructions to launch the Enterprise Service Discovery (ESD) wizard.

1. Switch to the Business Integration Perspective in IBM Integration Designer 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**.



Creates a service to access the database system using WebSphere Adapter for JDBC. You can create integrated processes, which includes the exchange of information with a database, without special coding. During the outbound processing the adapter converts a request from the application to an outbound operation, which it runs to create, retrieve, update, or delete data in the database or to run a database program stored in the database. During inbound processing, the adapter polls the event store, retrieves and processes

4. Select the **IBM WebSphere Adapter for JDBC (IBM: 7.5.0.0)** and click **Next.**



New External Service

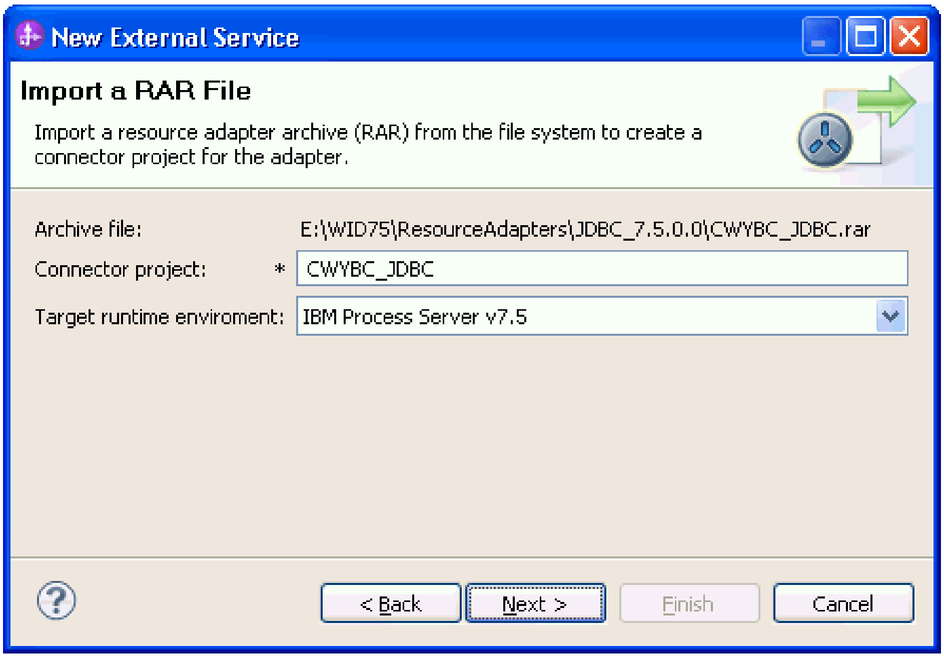
Select an Adapter

Select the version of the adapter that you want to use.

IBM WebSphere Adapter for JDBC (IBM : 7.5.0.0)

5. In the **Connector project** field enter **CWYBC_JDBC.**

6. In the **Target runtime environment** field, select appropriate runtime and click **Next**.



New External Service

Import a RAR File

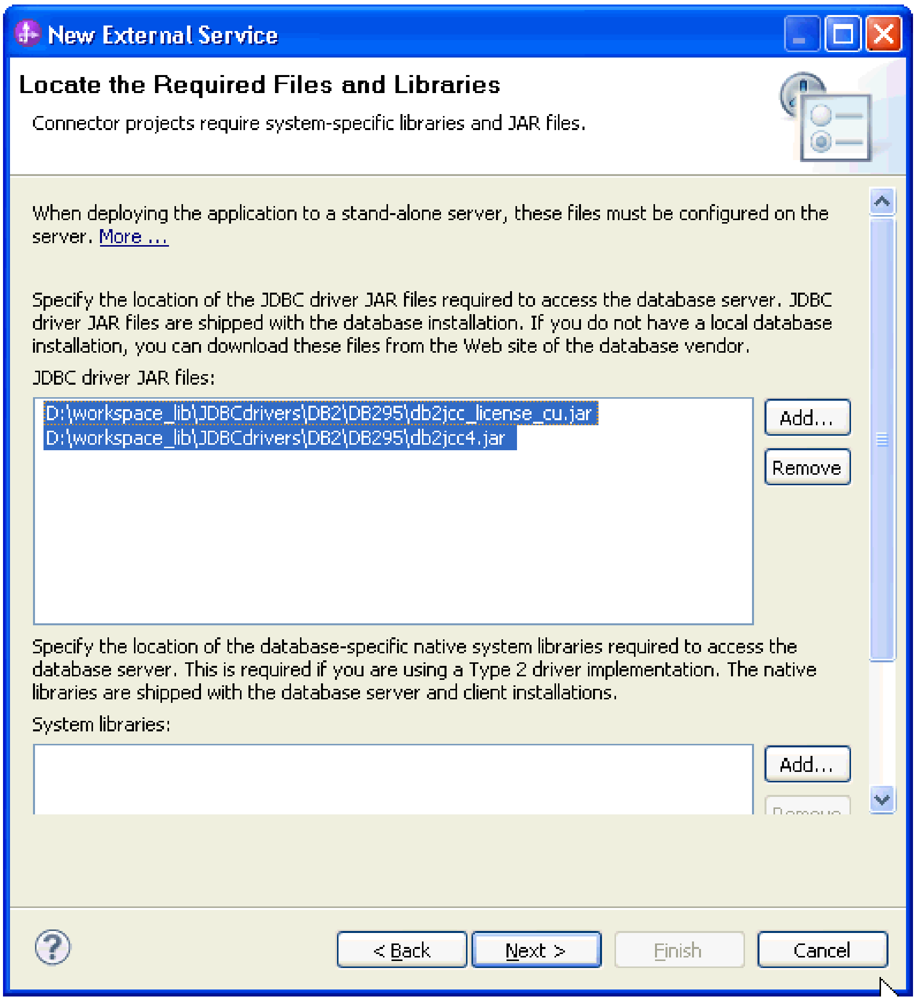
Import a resource adapter archive (RAR) from the file system to create a connector project for the adapter.

Archive file: E:\WID75\ResourceAdapters\JDBC_7.5.0.0\CWYBC_JDBC.rar

Connector project: * CWYBC_JDBC

Target runtime enviroment: IBM Process Server v7.5

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**.



New External Service

Locate the Required Files and Libraries

Connector projects require system-specific libraries and JAR files.

When deploying the application to a stand-alone server, these files must be configured on the server. More ...

Specify the location of the JDBC driver JAR files required to access the database server. JDBC driver JAR files are shipped with the database installation. If you do not have a local database installation, you can download these files from the Web site of the database vendor.

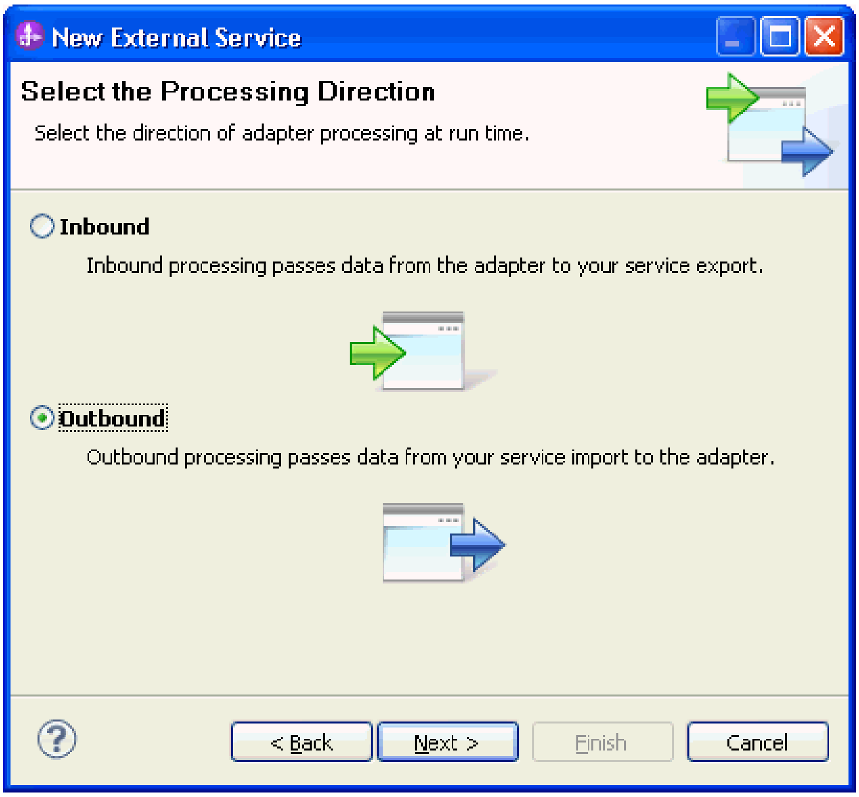
JDBC driver JAR files:

D:\workspace_lib\JDBCdrivers\DB2\DB295\db2jcc_license_cu.jar
D:\workspace_lib\JDBCdrivers\DB2\DB295\db2jcc4.jar

Specify the location of the database-specific native system libraries required to access the database server. This is required if you are using a Type 2 driver implementation. The native libraries are shipped with the database server and client installations.

System libraries:

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



Select the Processing Direction

Select the direction of adapter processing at run time.

○ Inbound

Inbound processing passes data from the adapter to your service export.

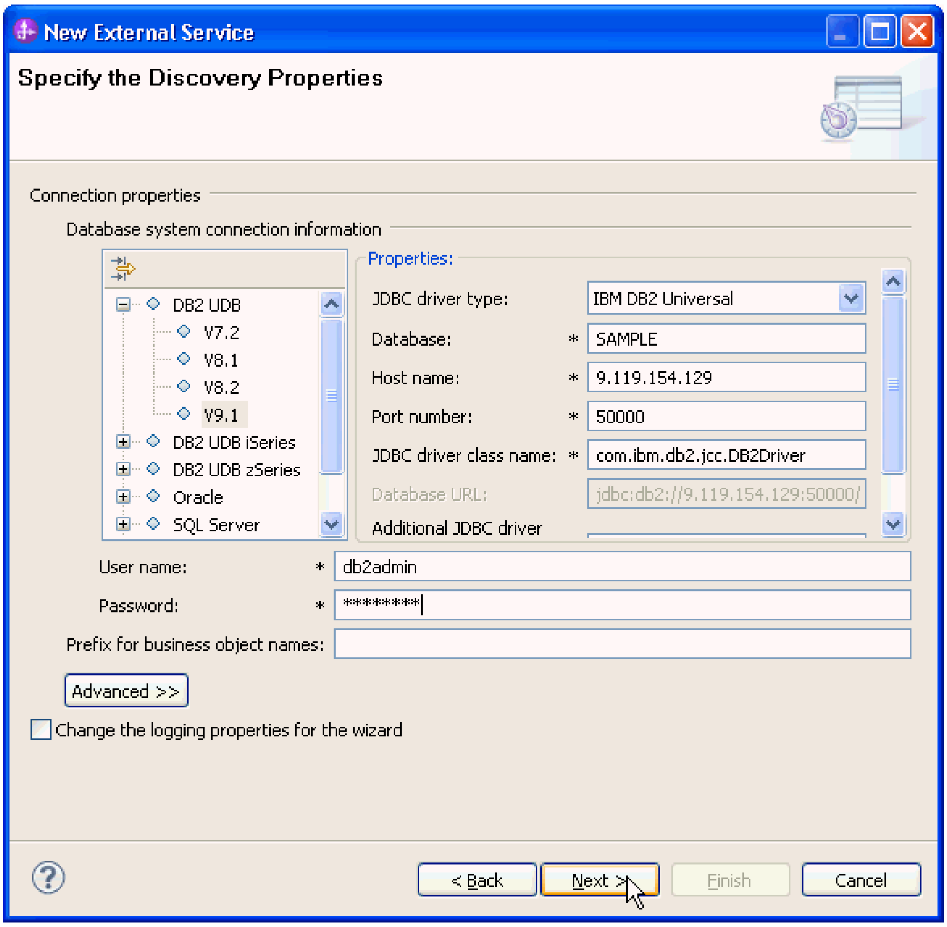
● Outbound

Outbound processing passes data from your service import to the adapter.

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**.



New External Service

Specify the Discovery Properties

Connection properties

Database system connection information

- DB2 UDB
  - V7.2
  - V8.1
  - V8.2
  - V9.1
- DB2 UDB iSeries
- DB2 UDB zSeries
- Oracle
- SQL Server

Properties:

JDBC driver type: IBM DB2 Universal

Database: * SAMPLE

Host name: * 9.119.154.129

Port number: * 50000

JDBC driver class name: * com.ibm.db2.jcc.DB2Driver

Database URL: jdbc:db2://9.119.154.129:50000/

Additional JDBC driver

User name: * db2admin

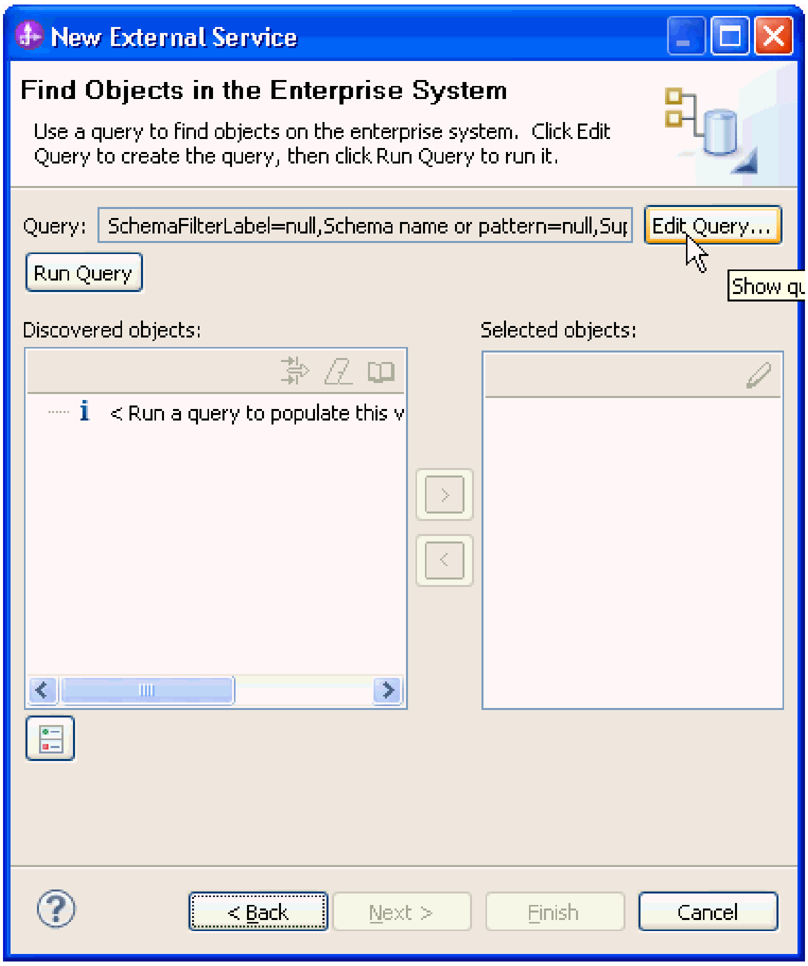
Password: * *********

Prefix for business object names:

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 **Edit Query.**



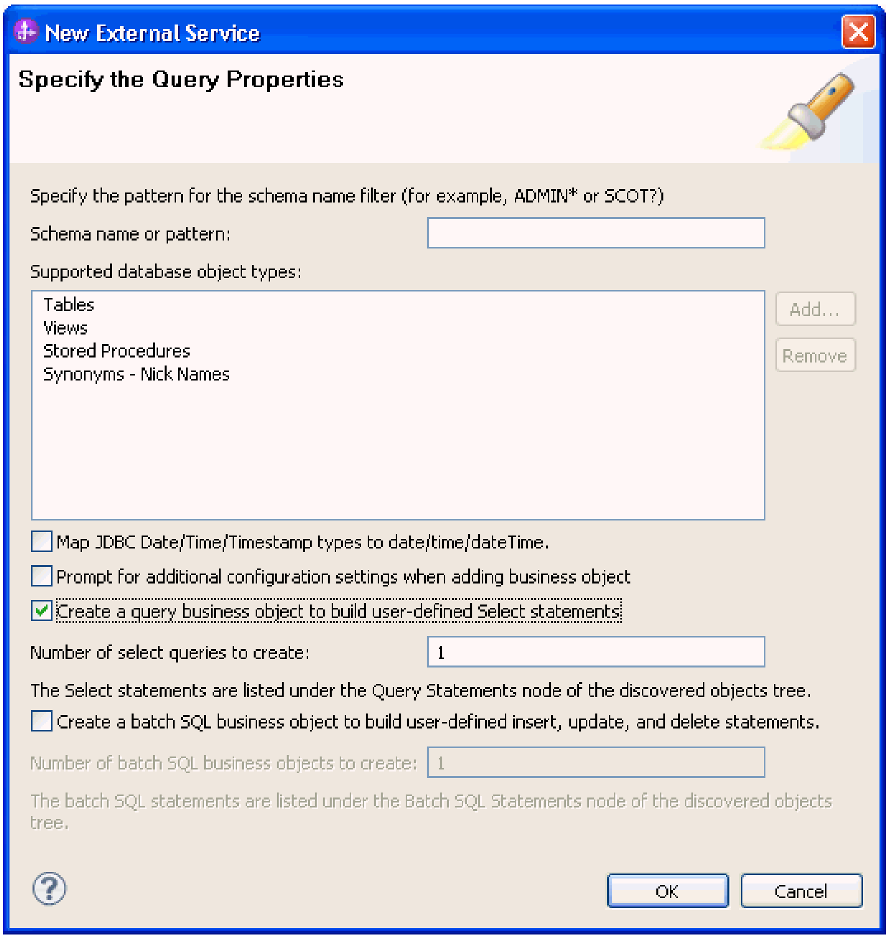
New External Service

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,Sup

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**.



New External Service

Specify the Query Properties

Specify the pattern for the schema name filter (for example, ADMIN* or SCOT?)

Schema name or pattern:

Supported database object types:

Tables
Views
Stored Procedures
Synonyms - Nick Names

☐ Map JDBC Date/Time/Timestamp types to date/time/dateTime.

☐ Prompt for additional configuration settings when adding business object

☑ Create a query business object to build user-defined Select statements

Number of select queries to create: 1

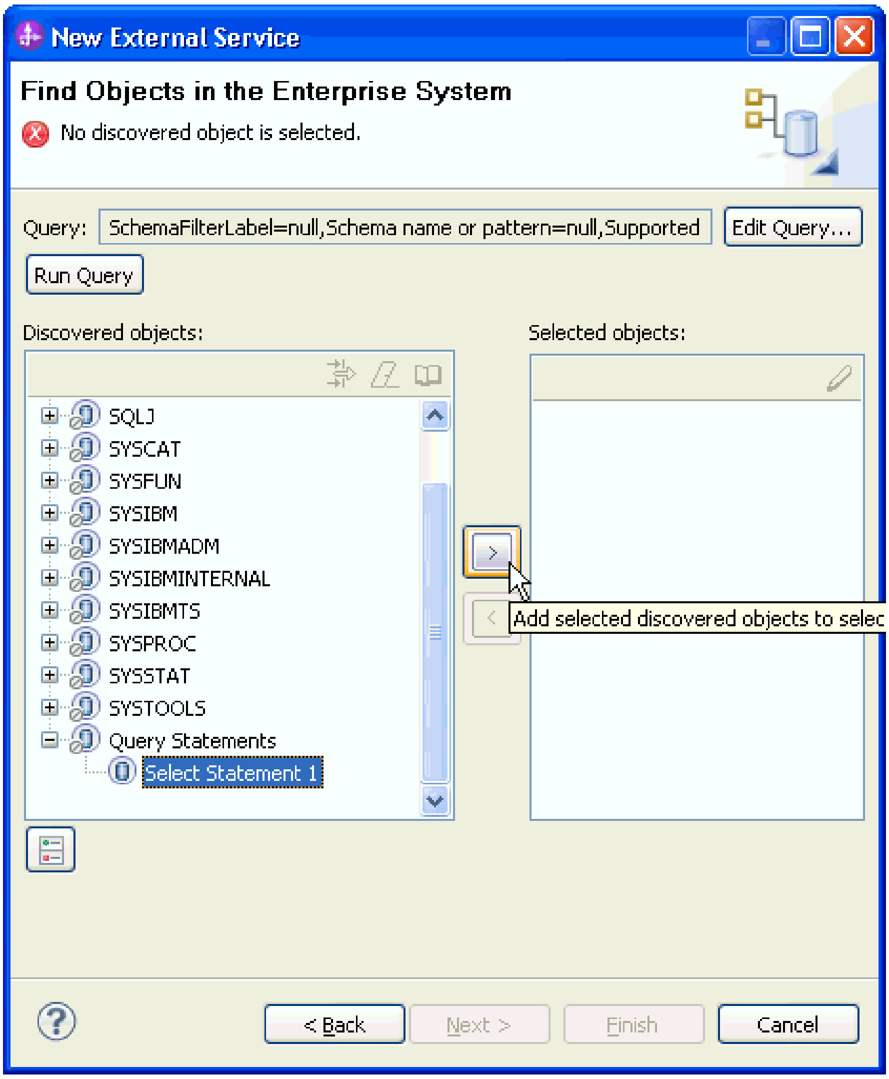
The Select statements are listed under the Query Statements node of the discovered objects tree.

☐ Create a batch SQL business object to build user-defined insert, update, and delete statements.

Number of batch SQL business objects to create: 1

The batch SQL statements are listed under the Batch SQL Statements node of the discovered objects tree.

3. Click **Run Query**.



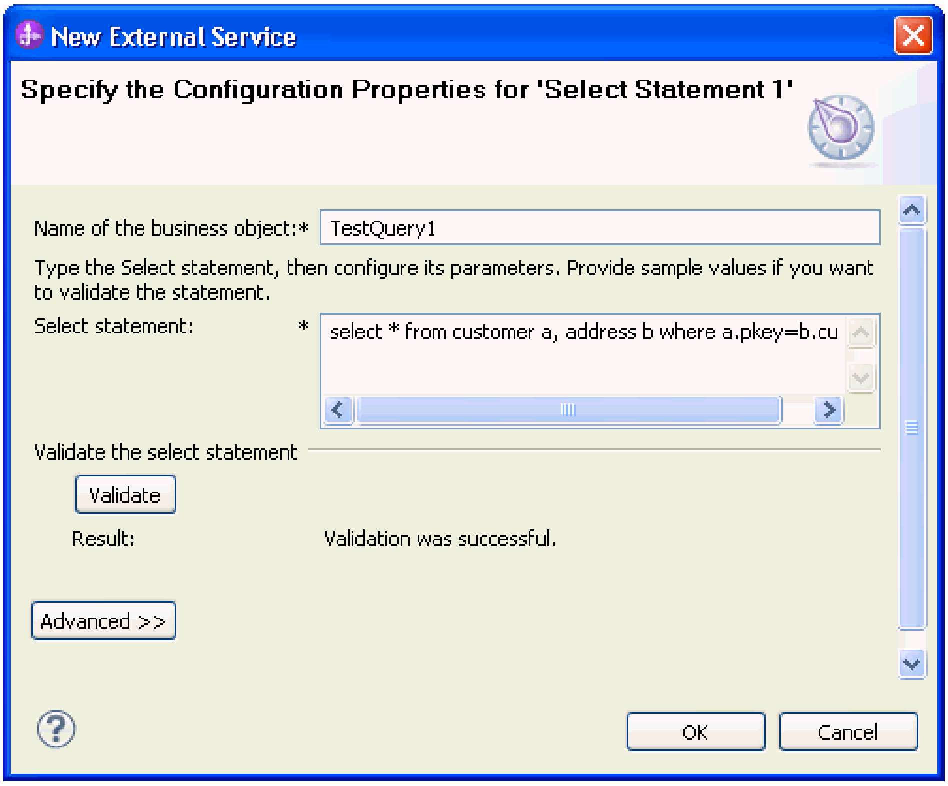
New External Service

Find Objects in the Enterprise System

No discovered object is selected.

Query: SchemaFilterLabel=null,Schema name or pattern=null,Supported

4. Span Expand the **Query Statements** node, select the **Select Statement 1** and click >. The window to configure the query object is displayed.



New External Service

Specify the Configuration Properties for 'Select Statement 1'

Name of the business object:* TestQuery1

Type the Select statement, then configure its parameters. Provide sample values if you want to validate the statement.

Select statement: * select * from customer a, address b where a.pkey=b.cu

Validate the select statement

Result: Validation was successful.

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 (?).

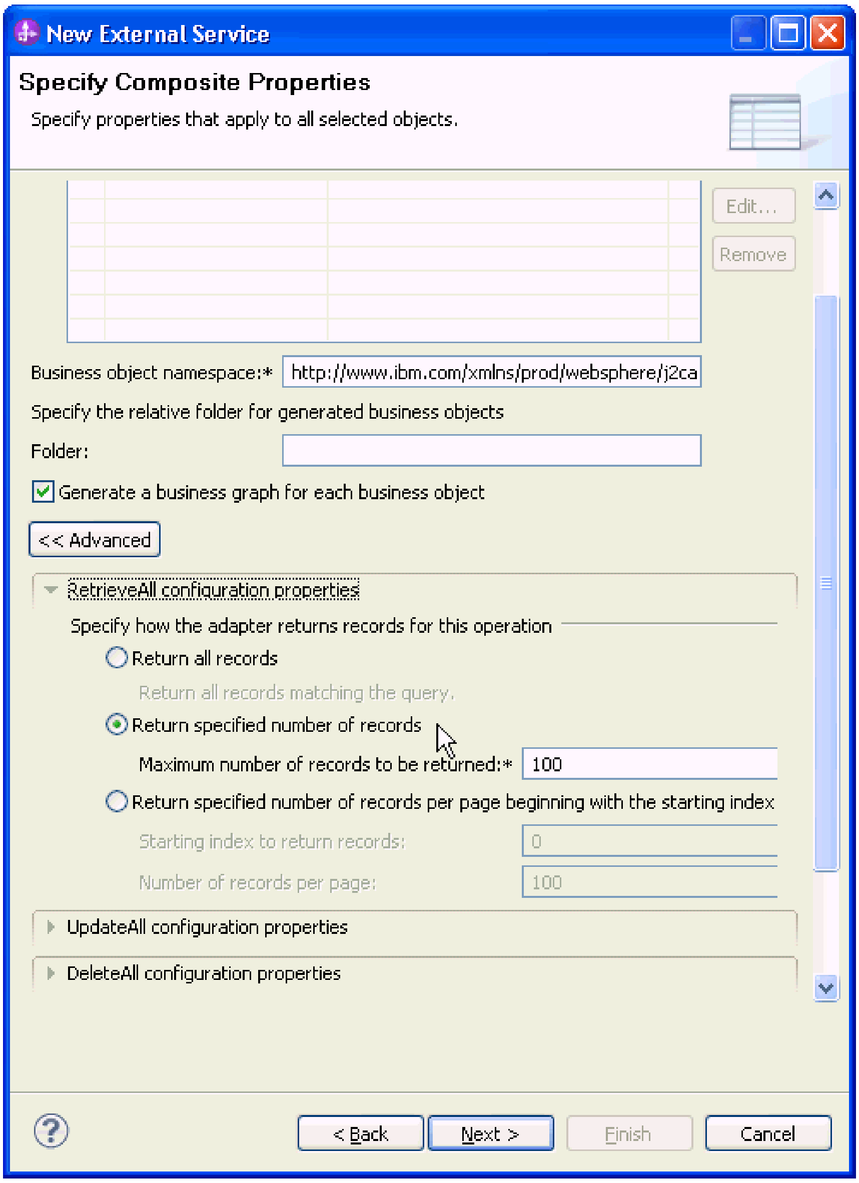
7. Click **Validate**. The Result area displays the result of the validation. Click **OK**.

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, click the **Advanced** button, collapse the **RetrieveAll configuration properties**, accept the default values for all fields in this group and click **Next**.



New External Service

Specify Composite Properties

Specify properties that apply to all selected objects.

Business object namespace:* http://www.ibm.com/xmlns/prod/websphere/j2ca

Specify the relative folder for generated business objects

Folder:

☑ Generate a business graph for each business object

RetrieveAll configuration properties

Specify how the adapter returns records for this operation

○ Return all records

Return all records matching the query.

● Return specified number of records

Maximum number of records to be returned:* 100

○ Return specified number of records per page beginning with the starting index

Starting index to return records: 0

Number of records per page: 100

UpdateAll configuration properties

DeleteAll configuration properties

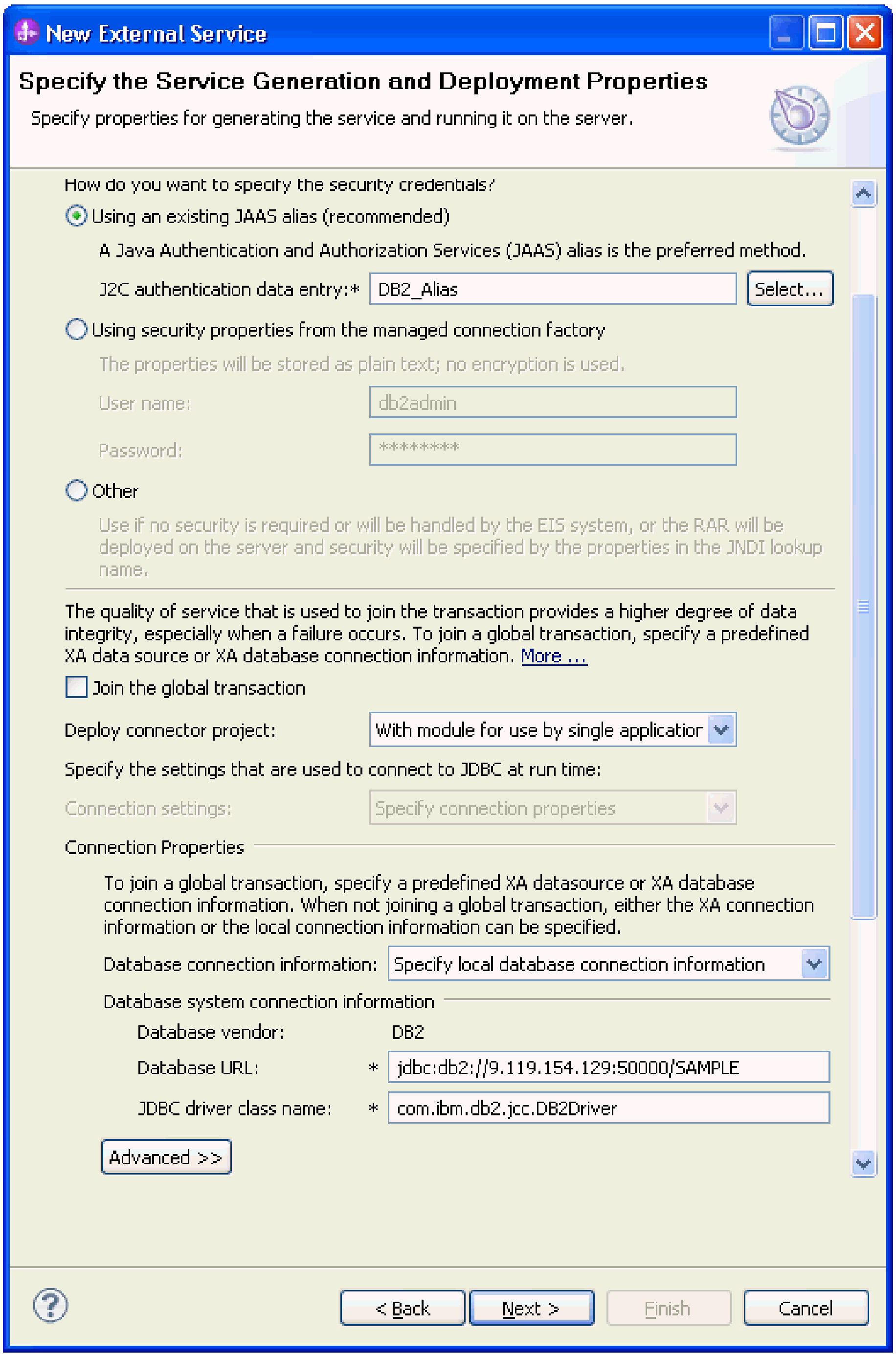
2. In the Specify the Service Generation and Deployment Properties window, perform the following steps;

   a) Select **Using an existing JAAS alias** for security options under **Deployment Properties**.

   b) Enter the authentication alias that you created in previews 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.**



Specify the Service Generation and Deployment Properties

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

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:* DB2_Alias Select...

○ Using security properties from the managed connection factory

The properties will be stored as plain text; no encryption is used.

User name: db2admin

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 join a global transaction, specify a predefined XA data source or XA database connection information. More ...

□ Join the global transaction

Deploy connector project: With module for use by single applicatior

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

Connection settings: Specify connection properties

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: Specify local database connection information

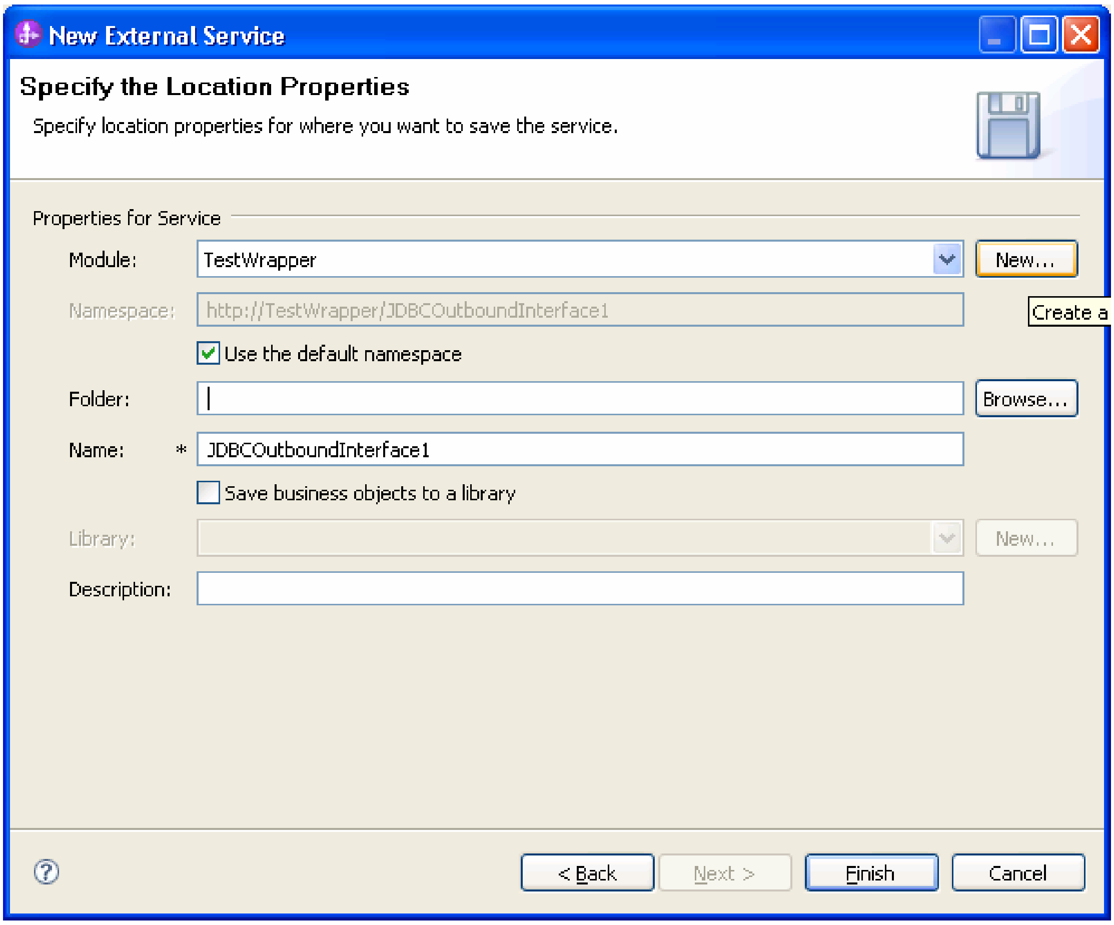
Database system connection information

Database vendor: DB2

Database URL: * jdbc:db2://9.119.154.129:50000/SAMPLE

JDBC driver class name: * com.ibm.db2.jcc.DB2Driver

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



New External Service

Specify the Location Properties

Specify location properties for where you want to save the service.

Properties for Service

Module: TestWrapper

Namespace: http://TestWrapper/JDBCOutboundInterface1

☑ Use the default namespace

Folder:

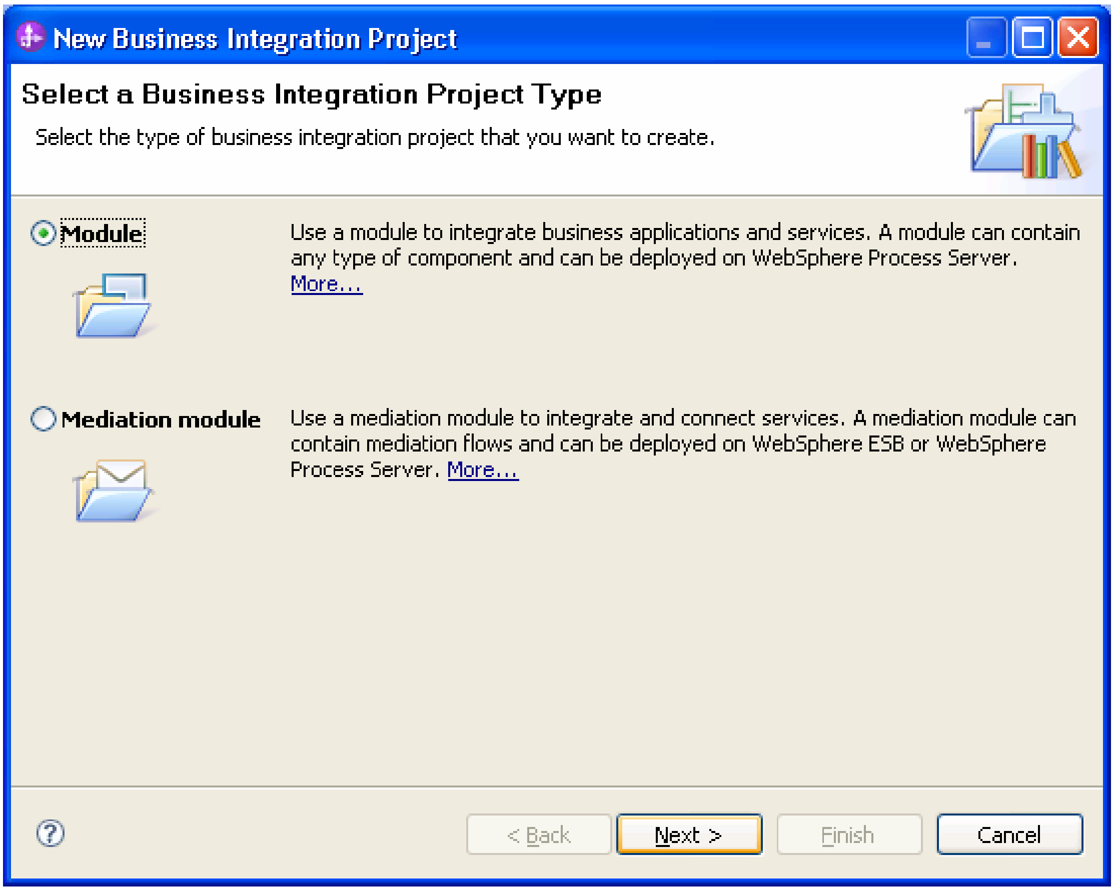
Name: * JDBCOutboundInterface1

□ Save business objects to a library

Library:

Description:

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



Select a Business Integration Project Type

Select the type of business integration project that you want to create.

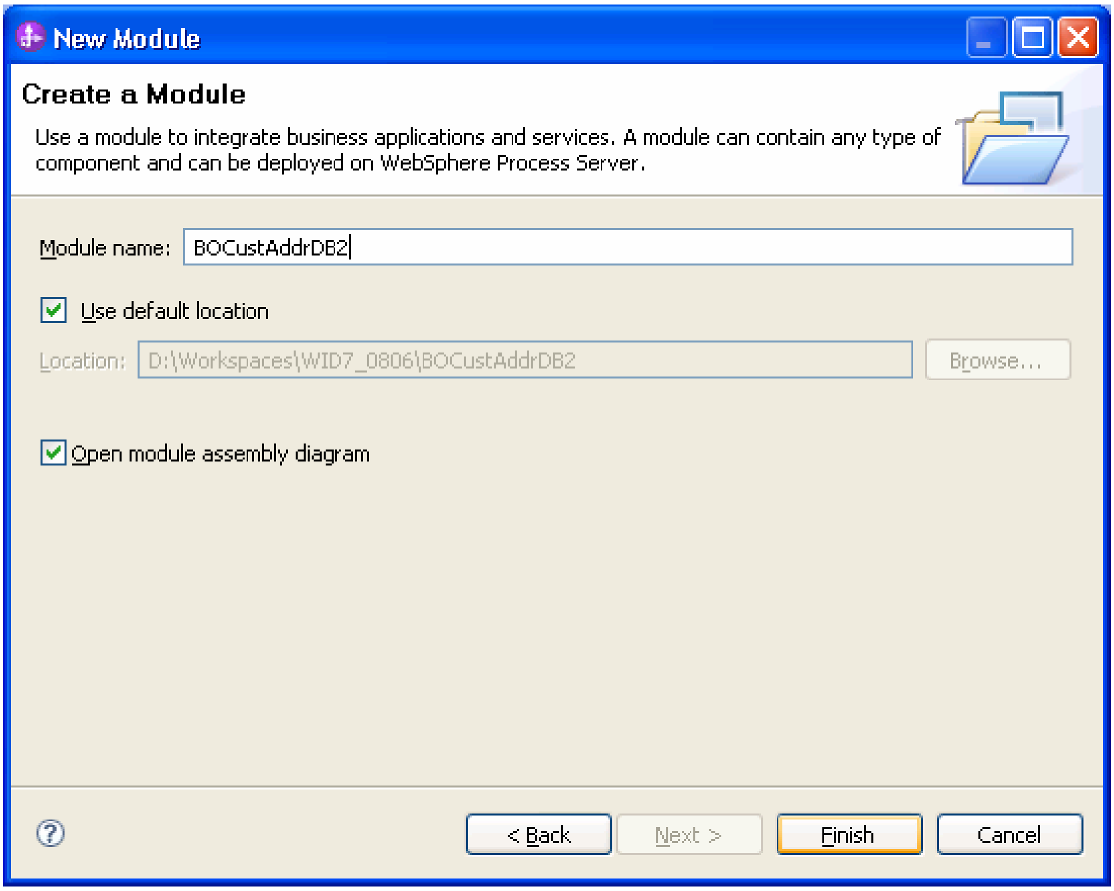
● Module

Use a module to integrate business applications and services. A module can contain any type of component and can be deployed on WebSphere Process Server. More...

○ Mediation module

Use a mediation module to integrate and connect services. A mediation module can contain mediation flows and can be deployed on WebSphere ESB or WebSphere Process Server. More...

5. In the Create a Module window, type **BOCustAddrDB2** in the **Module Name** field, and click **Finish**.



New Module

Create a Module

Use a module to integrate business applications and services. A module can contain any type of component and can be deployed on WebSphere Process Server.

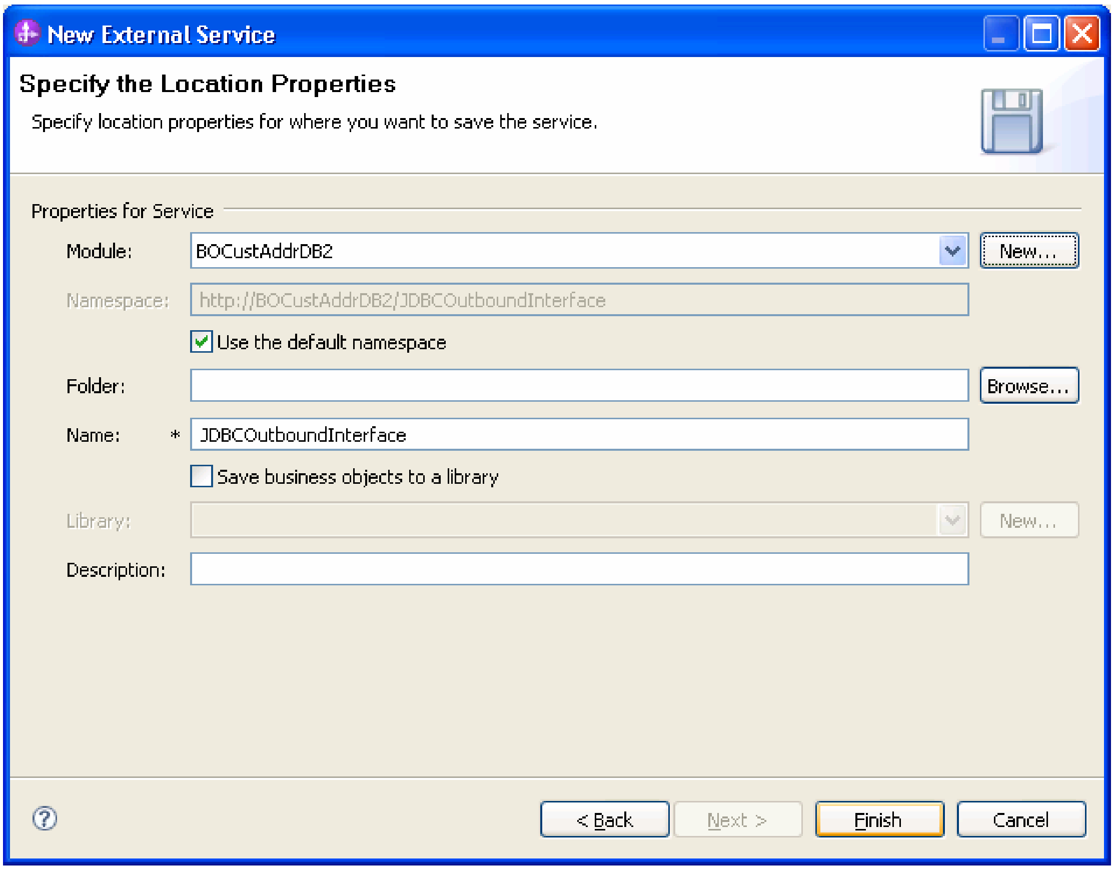
Module name: BOCustAddrDB2

☑ Use default location

Location: D:\Workspaces\WID7_0806\BOCustAddrDB2

☑ Open module assembly diagram

6. Accept the default values and click **Finish**.



New External Service

Specify the Location Properties

Specify location properties for where you want to save the service.

Properties for Service

Module: BOCustAddrDB2

Namespace: http://BOCustAddrDB2/JDBCOutboundInterface

☑ Use the default namespace

Folder:

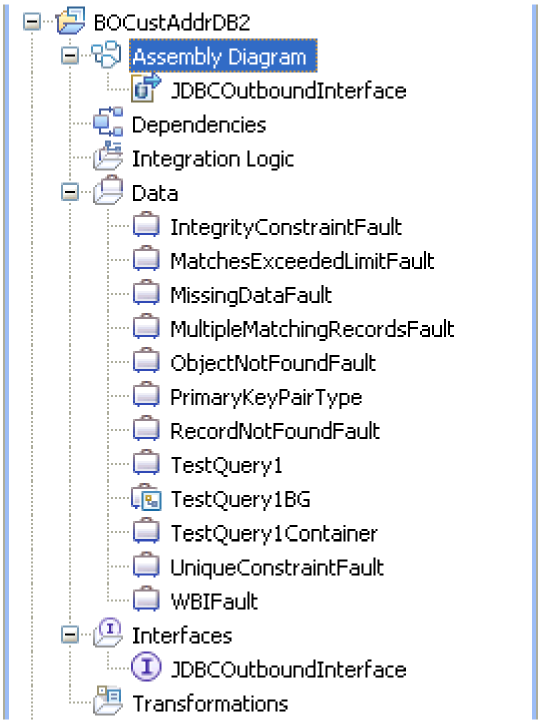
Name: * JDBCOutboundInterface

☐ Save business objects to a library

Library:

Description:

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



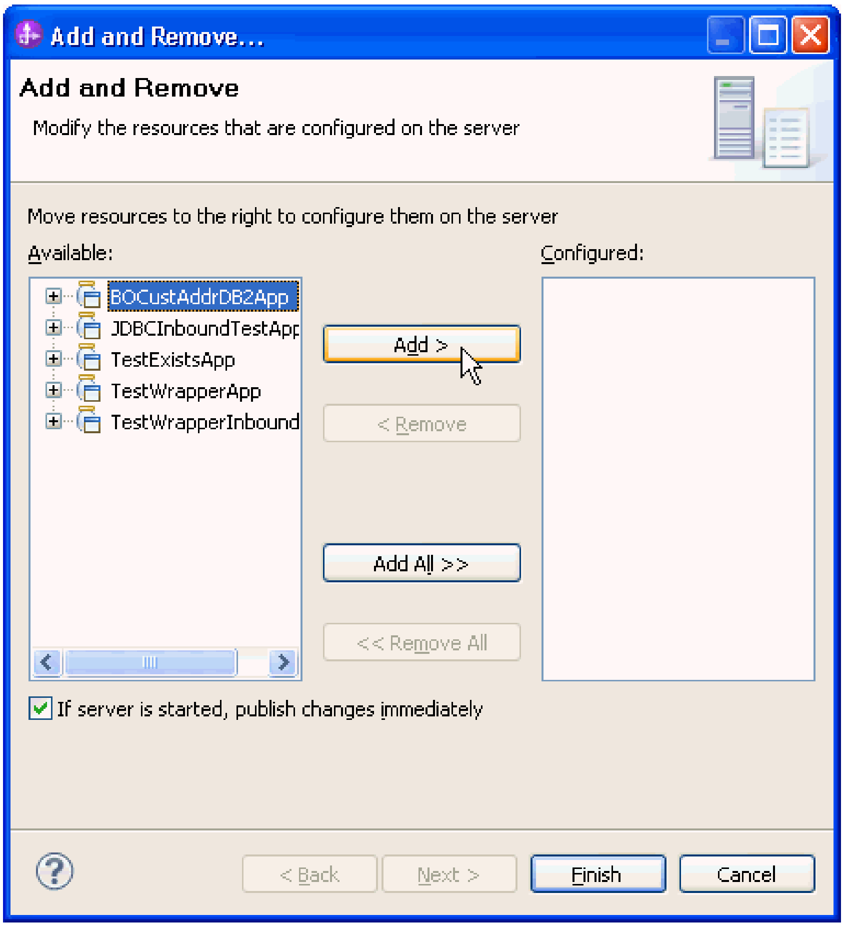
- BOCustAddrDB2
  - Assembly Diagram
    - JDBCOutboundInterface
  - Dependencies
  - Integration Logic
  - Data
    - IntegrityConstraintFault
    - MatchesExceededLimitFault
    - MissingDataFault
    - MultipleMatchingRecordsFault
    - ObjectNotFoundFault
    - PrimaryKeyPairType
    - RecordNotFoundFault
    - TestQuery1
    - TestQuery1BG
    - TestQuery1Container
    - UniqueConstraintFault
    - WBIFault
  - Interfaces
    - JDBCOutboundInterface
  - Transformations

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 IBM Integration Designer integration test client. To do this, you must add the SCA module you created earlier to the server using the **Servers** view in IBM Integration Designer.

Steps for adding the SCA module to the server:

1. In IBM Integration Designer, 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 **BOCustAddrDB2App** module to the server.



Add and Remove...

Add and Remove

Modify the resources that are configured on the server

Move resources to the right to configure them on the server

Available:

- BOCustAddrDB2App
- JDBCInboundTestApp
- TestExistsApp
- TestWrapperApp
- TestWrapperInbound

Configured:

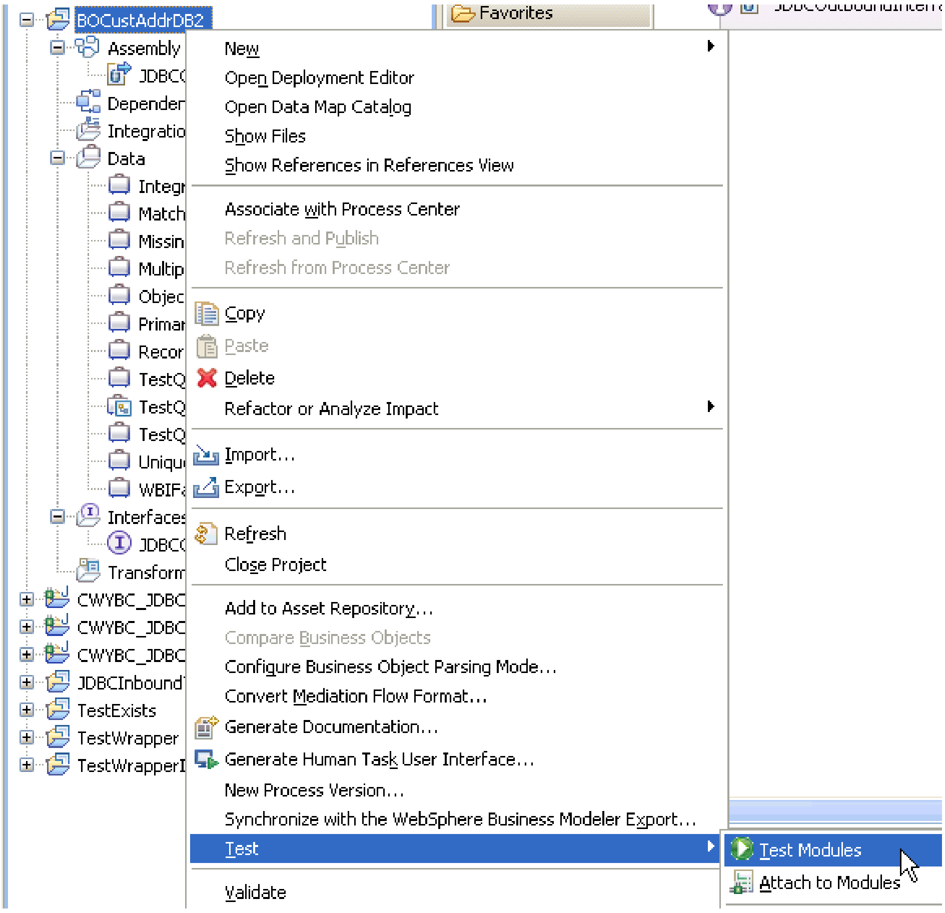
☑ If server is started, publish changes immediately

5. Click **Finish.**.

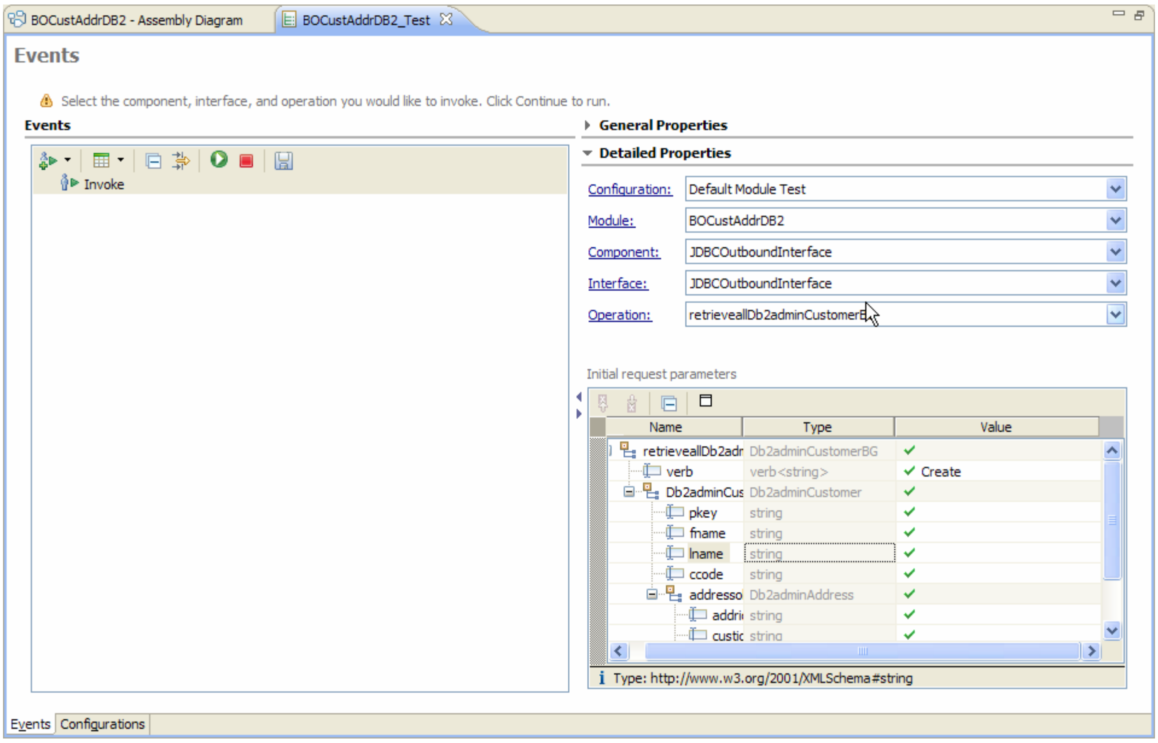
Test the assembled adapter application

Test the assembled adapter application using the IBM Integration Designer 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.



Events

Select the component, interface, and operation you would like to invoke. Click Continue to run.

Configuration: Default Module Test

Module: BOCustAddrDB2

Component: JDBCOutboundInterface

Interface: JDBCOutboundInterface

Operation: retrieveallDb2adminCustomerB

Initial request parameters

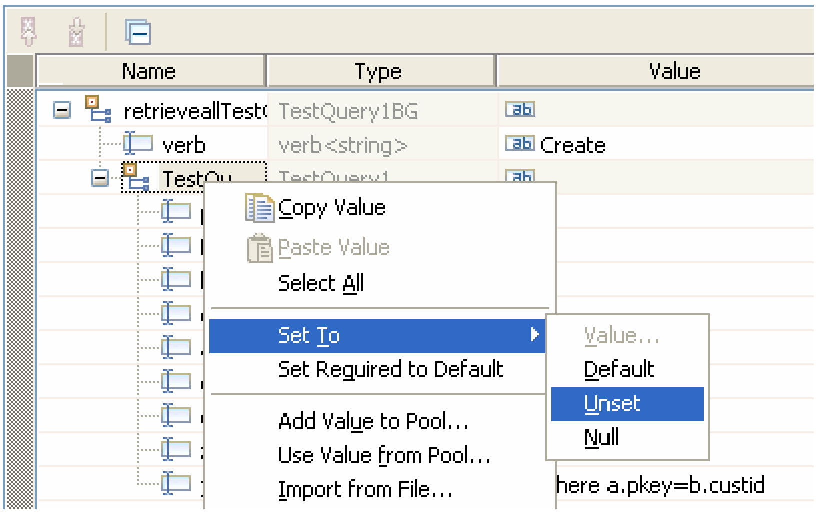
Name	Type	Value
retrieveallDb2adr	Db2adminCustomerBG	✓
verb	verb<string>	✓ Create
Db2adminCus	Db2adminCustomer	✓
pkey	string	✓
fname	string	✓
lname	string	✓
ccode	string	✓
addresso	Db2adminAddress	✓
addri	string	✓
custic	string	✓

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

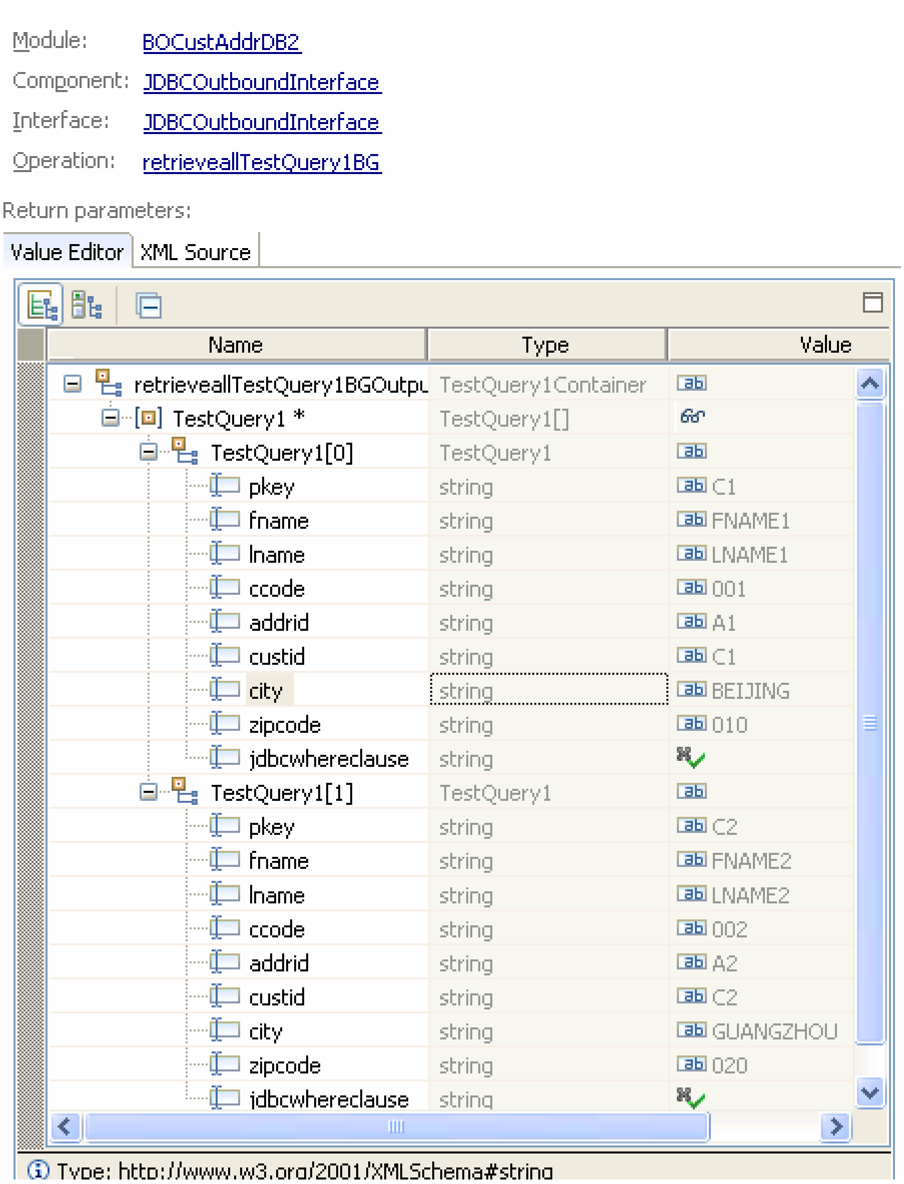
A test client page is automatically created and displayed and it is ready to accept values before the test execution. The first invocation is created by default. Subsequent invocation can be created by clicking Invoke .

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:

Name	Type	Value
retrieveallTestQuery1BGOutpu	TestQuery1Container	
TestQuery1 *	TestQuery1[]	
TestQuery1[0]	TestQuery1	
pkey	string	C1
fname	string	FNAME1
lname	string	LNAME1
ccode	string	001
addrid	string	A1
custid	string	C1
city	string	BEIJING
zipcode	string	010
jdbcwhereclause	string	
TestQuery1[1]	TestQuery1	
pkey	string	C2
fname	string	FNAME2
lname	string	LNAME2
ccode	string	002
addrid	string	A2
custid	string	C2
city	string	GUANGZHOU
zipcode	string	020
jdbcwhereclause	string	

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

Clear the sample content

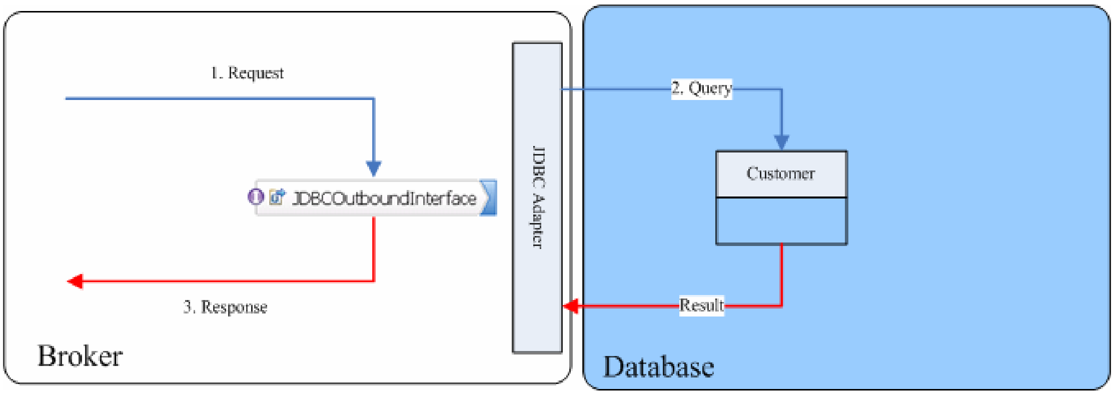
After a record has been created with the IBM Integration Designer environment, it can be removed with the Delete operation, determine the key field that uniquely represent the record just created and enter its value.

Chapter 14. Tutorial 13: Checking for the existence of a business object (Oracle)

This tutorial demonstrates how WebSphere Adapter for JDBC 7.5.0.0 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.

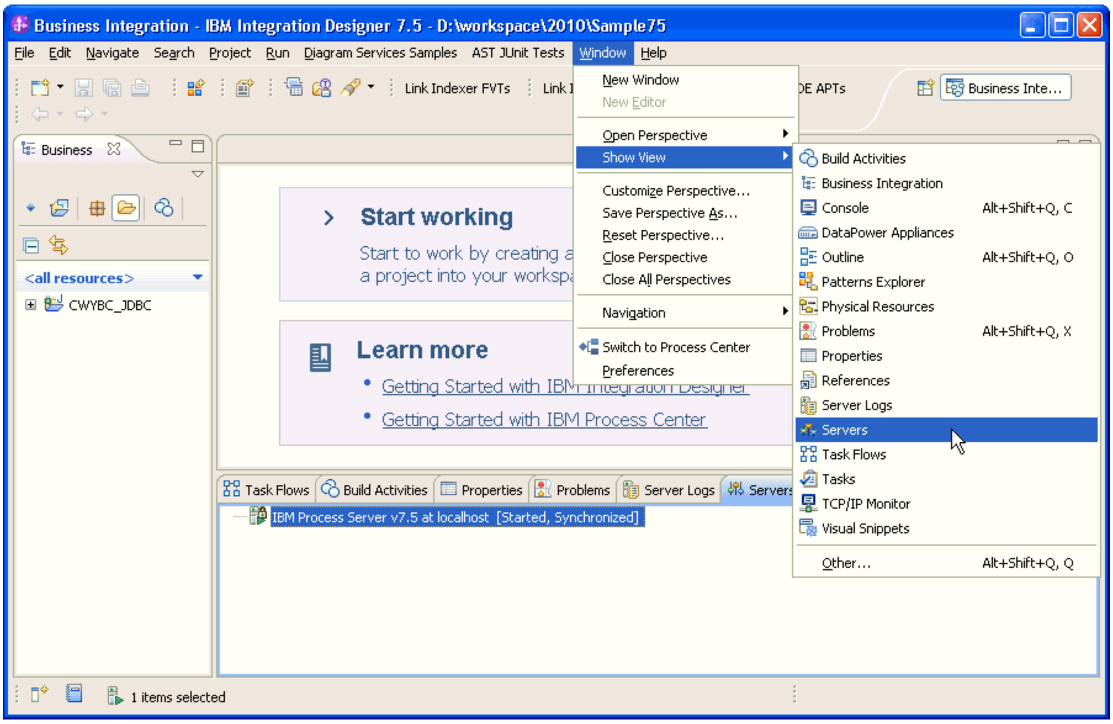
```
insert into customer values ('1000', 'testFname',
'testLname', 'testCcode')
```

Create an authentication alias

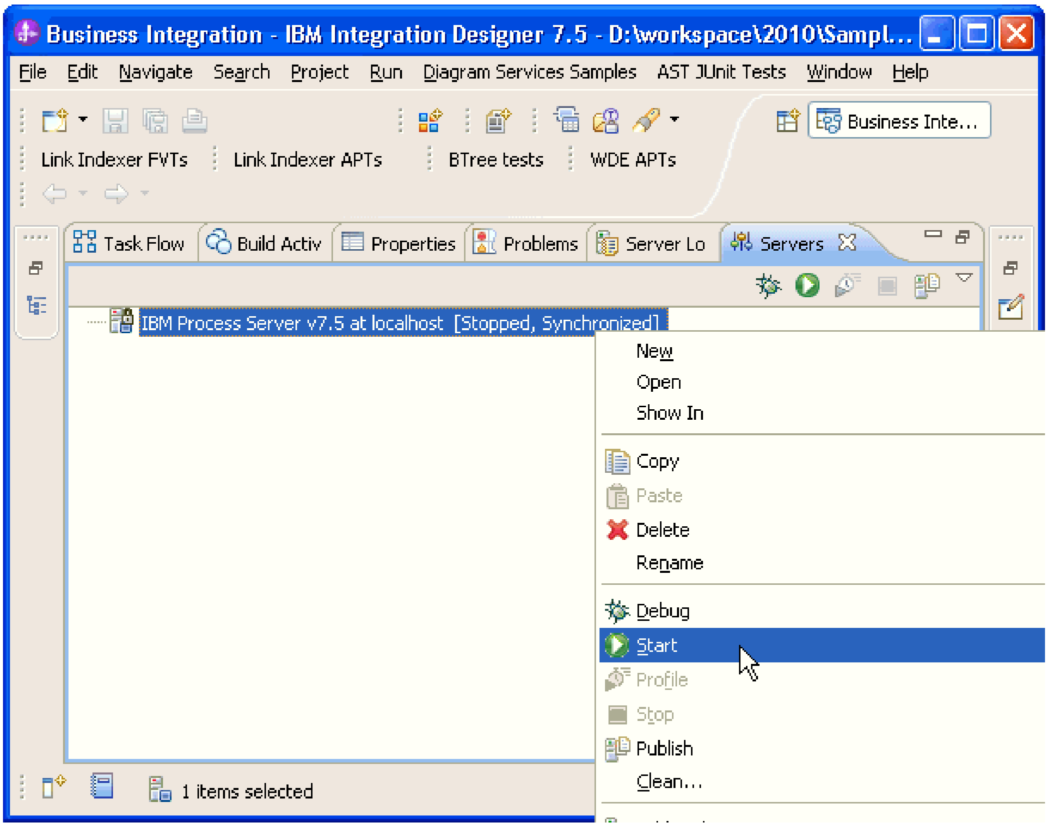
The authentication alias needs to be set because the data source created in the next section uses the username and password set in the authentication alias to connect to the database.

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

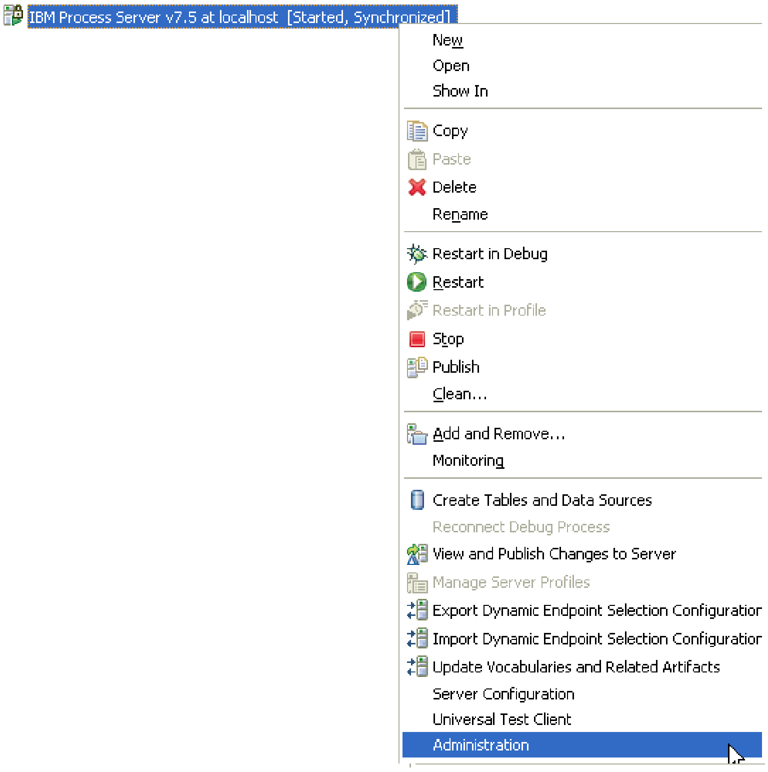
1. In IBM Integration Designer, switch to the **Servers** view by selecting **Window > Show View > Servers**.



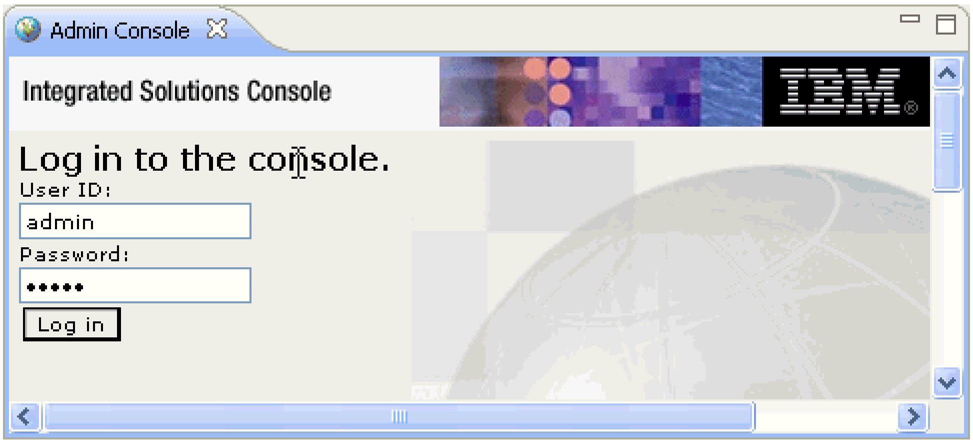
2. In the **Servers** view, right-click the server that you want to start and select **Start**.


Business Integration - IBM Integration Designer 7.5 - D:\workspace\2010\Sampl...
File Edit Navigate Search Project Run Diagram Services Samples AST JUnit Tests Window Help
Business Inte...
Link Indexer FVTs
Link Indexer APTs
BTree tests
WDE APTs
Task Flow
Build Activ
Properties
Problems
Server Lo
Servers
IBM Process Server v7.5 at localhost [Stopped, Synchronized]
New
Open
Show In
Copy
Paste
Delete
Rename
Debug
Start
Profile
Stop
Publish
Clean...
1 items selected

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


IBM Process Server v7.5 at localhost [Started, Synchronized]
New
Open
Show In
Copy
Paste
Delete
Rename
Restart in Debug
Restart
Restart in Profile
Stop
Publish
Clean...
Add and Remove...
Monitoring
Create Tables and Data Sources
Reconnect Debug Process
View and Publish Changes to Server
Manage Server Profiles
Export Dynamic Endpoint Selection Configuratior
Import Dynamic Endpoint Selection Configuratior
Update Vocabularies and Related Artifacts
Server Configuration
Universal Test Client
Administration

4. Log on to the administrative console.



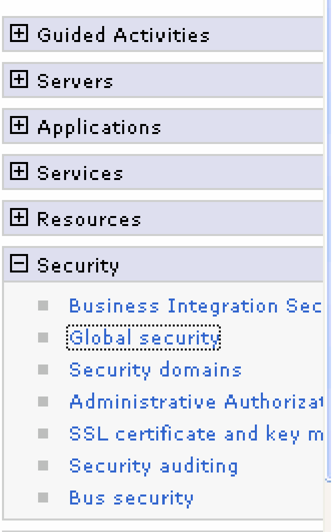
Log in to the console.

User ID: admin

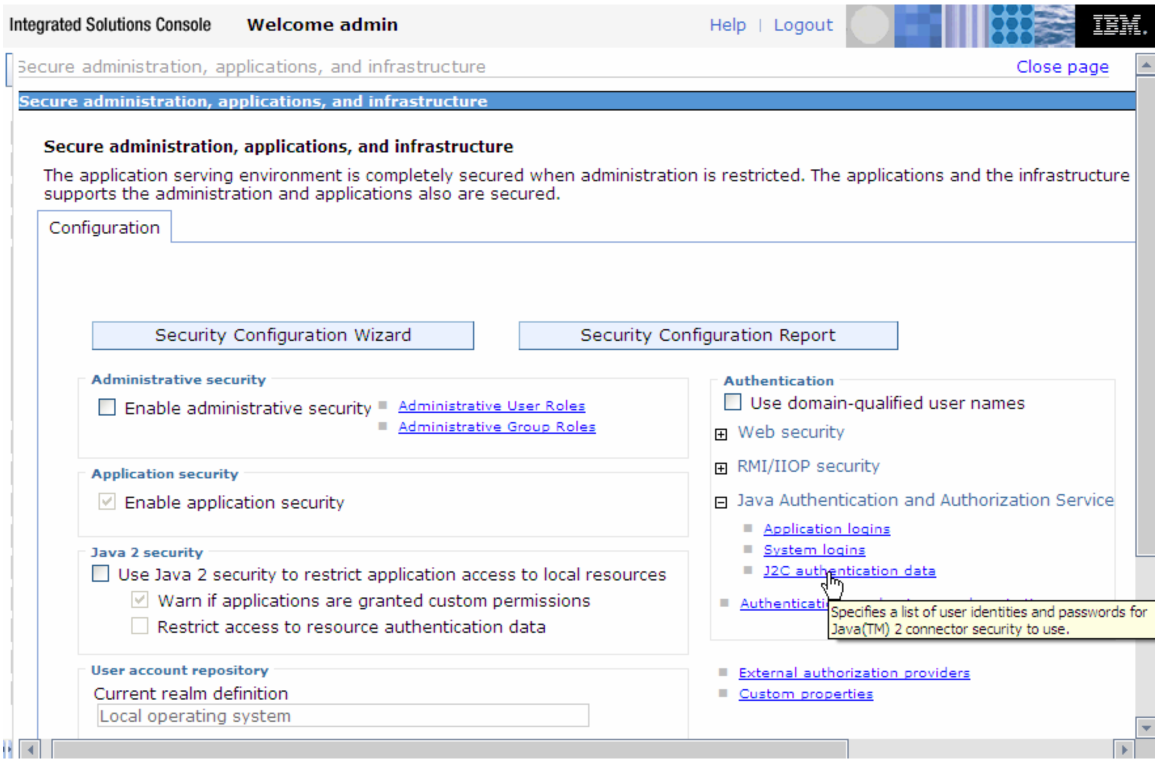
Password: •••••

Log in

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



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



Integrated Solutions Console Welcome admin Help | Logout

Secure administration, applications, and infrastructure Close page

Secure administration, applications, and infrastructure

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

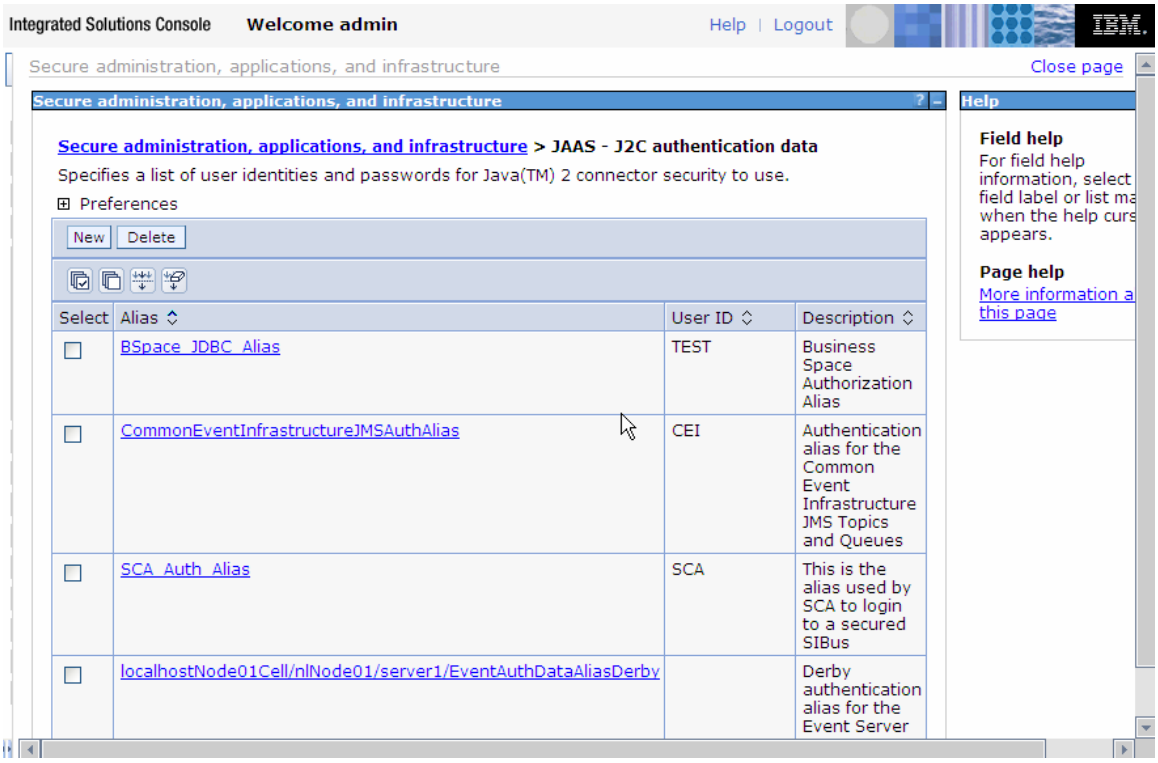
⊟ Java Authentication and Authorization Service
- Application logins
- System logins
- J2C authentication data

- Authenticati

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

Secure administration, applications, and infrastructure Close page

Secure administration, applications, and infrastructure

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.

⊞ Preferences

New Delete

Select	Alias	User ID	Description
□	BSpace_JDBC_Alias	TEST	Business Space Authorization Alias
□	CommonEventInfrastructureJMSAuthAlias	CEI	Authentication alias for the Common Event Infrastructure JMS Topics and Queues
□	SCA_Auth_Alias	SCA	This is the alias used by SCA to login to a secured SIBus
□	localhostNode01Cell/nlNode01/server1/EventAuthDataAliasDerby		Derby authentication alias for the Event Server

Help

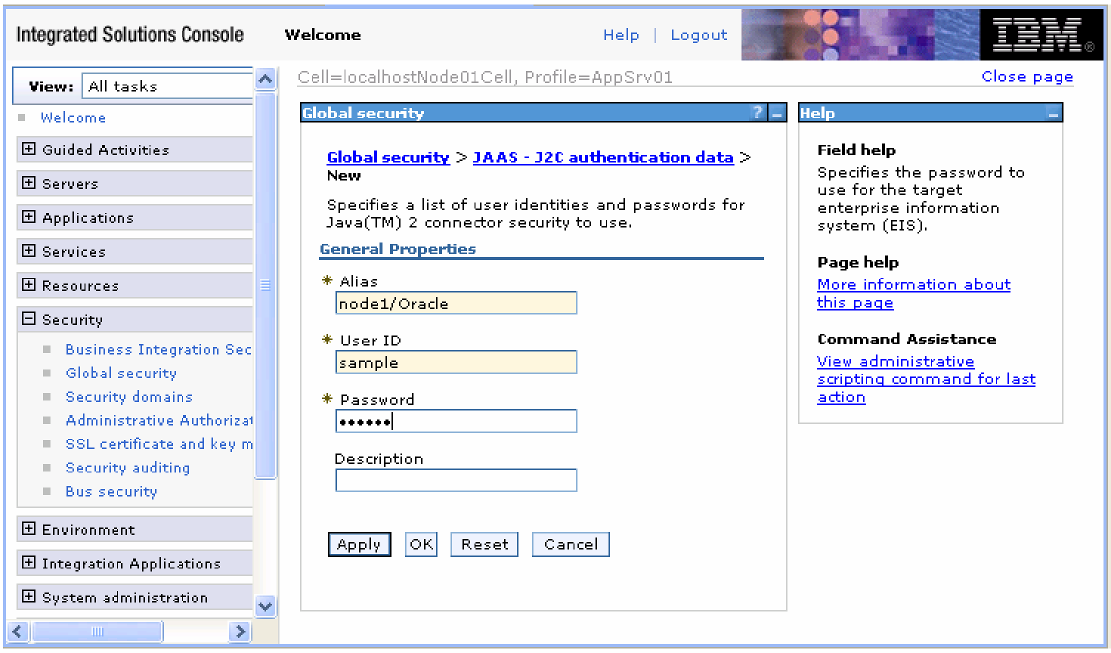
Field help

For field help information, select field label or list ma when the help curs appears.

Page help

More information a this page

7. 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 > JAAS - J2C authentication data > New

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

General Properties

* Alias: node1/Oracle

* User ID: sample

* Password: ••••••

Description:

Field help

Specifies the password to use for the target enterprise information system (EIS).

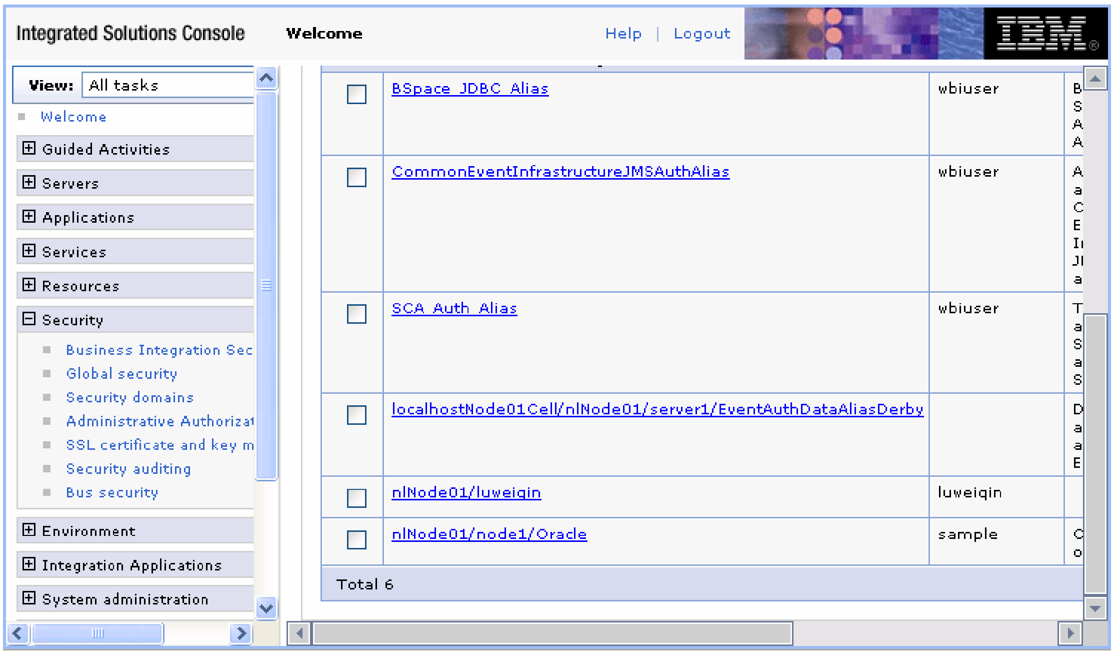
Page help

More information about this page

Command Assistance

View administrative scripting command for last action

You have created an authentication alias that will be used to configure the adapter properties.



□	BSpace_JDBC_Alias	wbiuser	B S A A
□	CommonEventInfrastructureJMSAuthAlias	wbiuser	A a C E I JI a
□	SCA_Auth_Alias	wbiuser	T a S a S
□	localhostNode01Cell/nlNode01/server1/EventAuthDataAliasDerby		D a a E
□	nlNode01/luweiqin	luweiqin	
□	nlNode01/node1/Oracle	sample	C o
Total 6			

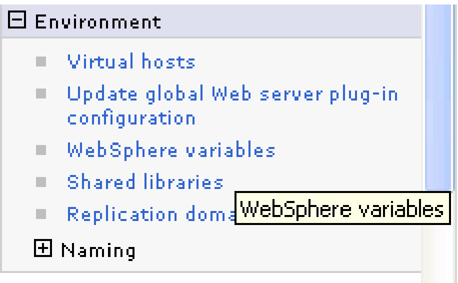
Create a data source

Create a data source in IBM 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.

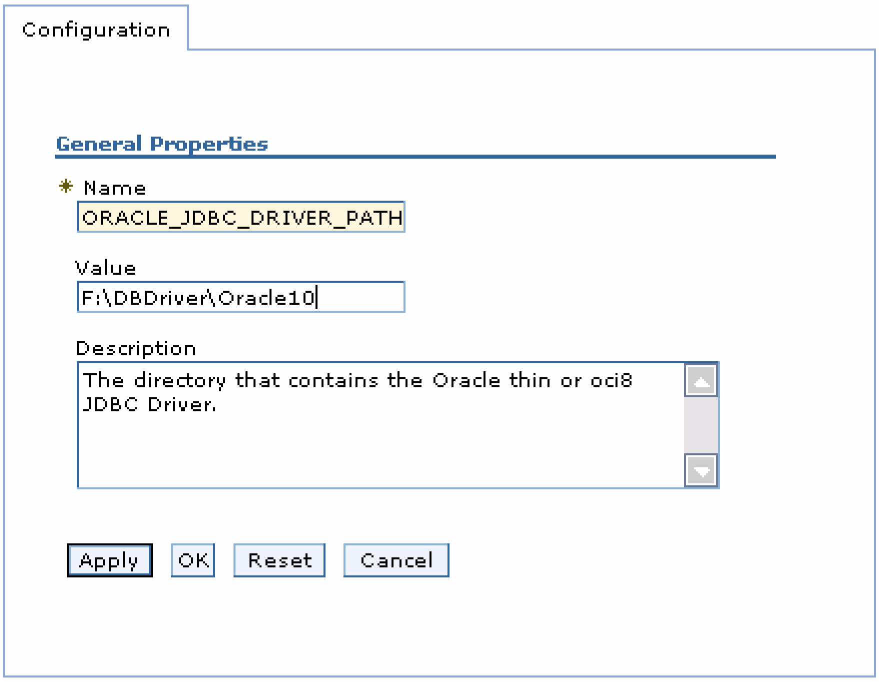
Note: This tutorial will use Oracle as the database and the Oracle thin driver, ojdbc6.jar.

Here are the steps to create the data source in the IBM 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**.

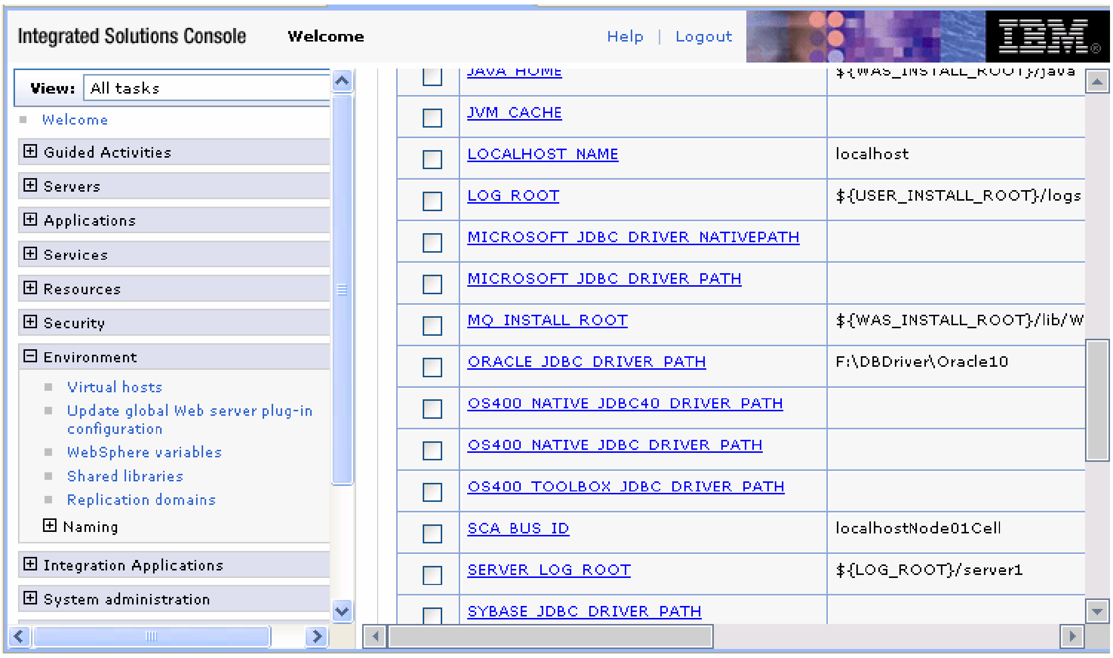


* Name
ORACLE_JDBC_DRIVER_PATH

Value
F:\DBDriver\Oracle10

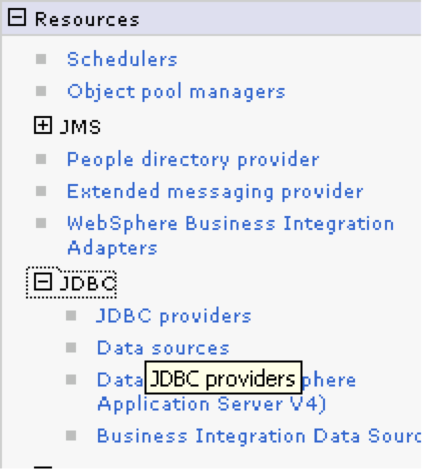
Description
The directory that contains the Oracle thin or oci8 JDBC Driver.

The variable is added and appears in the list.



☐	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.**

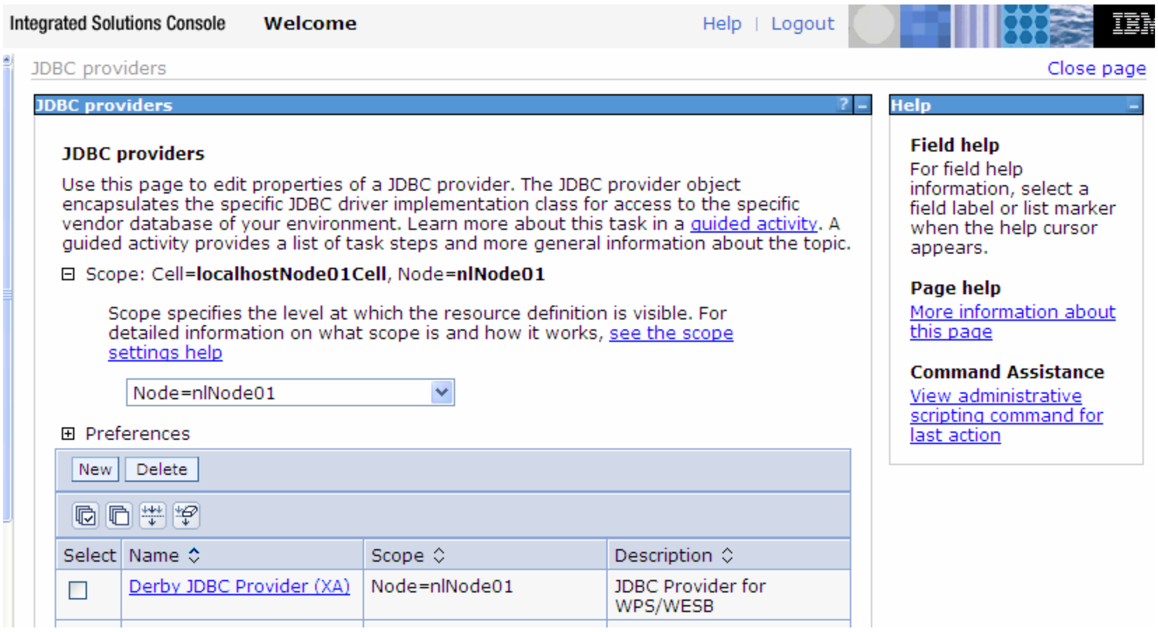


⊟ Resources
- Schedulers
- Object pool managers

⊞ JMS
- People directory provider
- Extended messaging provider
- WebSphere Business Integration Adapters

⊟ JDBC
- JDBC providers
- Data sources
- Data JDBC providers phere Application Server V4)
- Business Integration Data Sourc

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



Integrated Solutions Console   Welcome   Help | Logout

JDBC providers   Close page

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=**nlNode01**

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=nlNode01

⊞ Preferences

New   Delete

Select	Name ⇕	Scope ⇕	Description ⇕
□	Derby JDBC Provider (XA)	Node=nlNode01	JDBC Provider for WPS/WESB

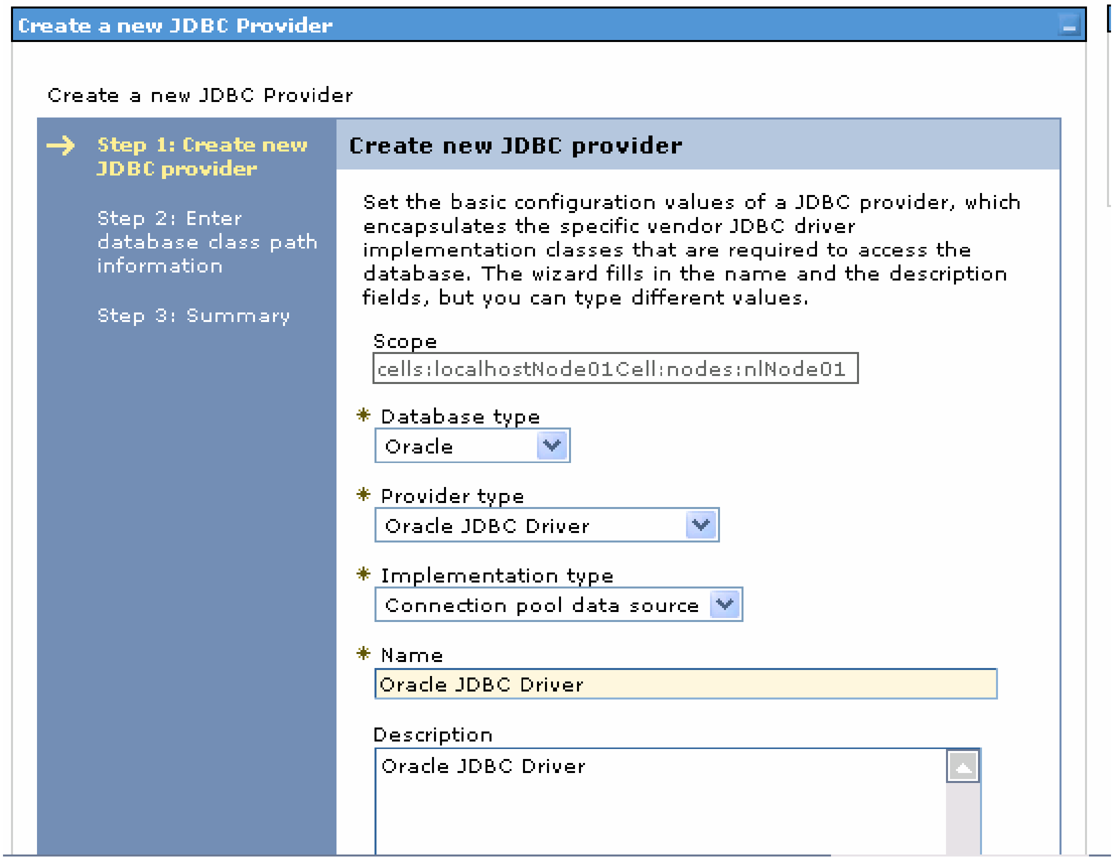
Help

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

5. Click **New**. Select the 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:nlNode01

* Database type
Oracle

* Provider type
Oracle JDBC Driver

* Implementation type
Connection pool data source

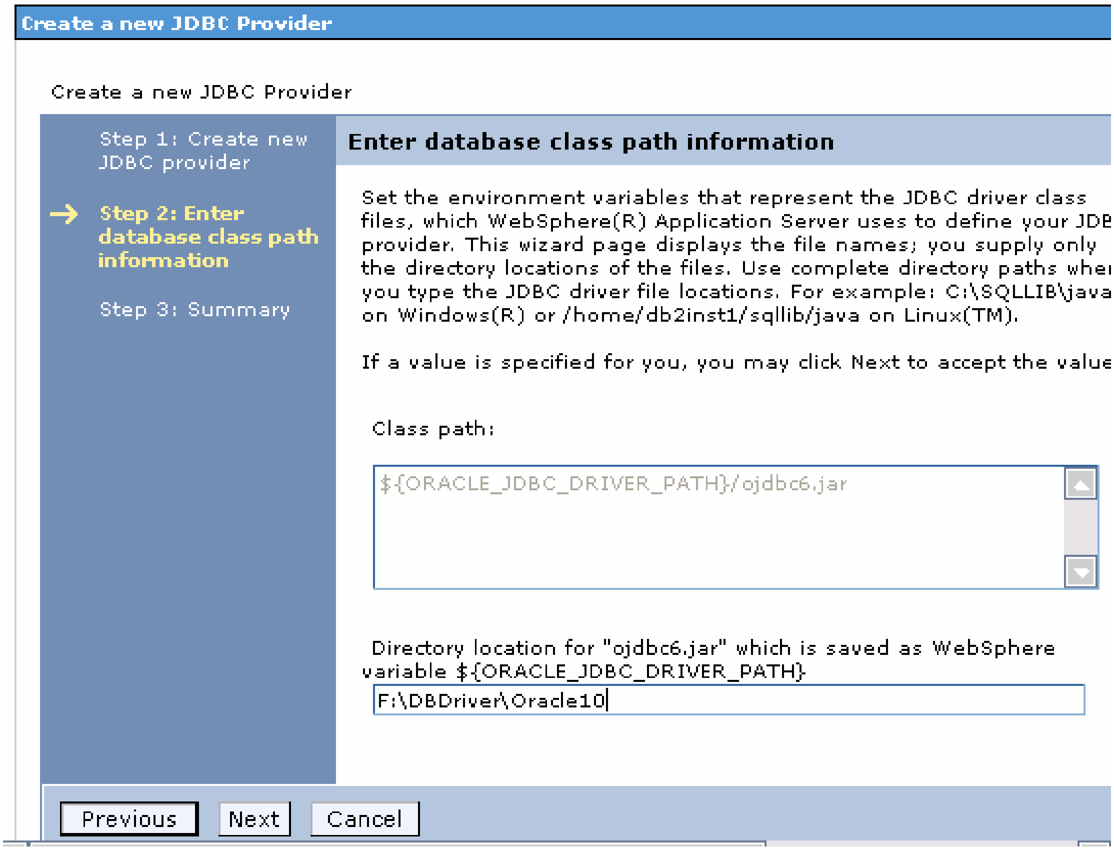
* Name
Oracle JDBC Driver

Description
Oracle JDBC Driver

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**.



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 JDE provider. This wizard page displays the file names; you supply only the directory locations of the files. Use complete directory paths wher 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

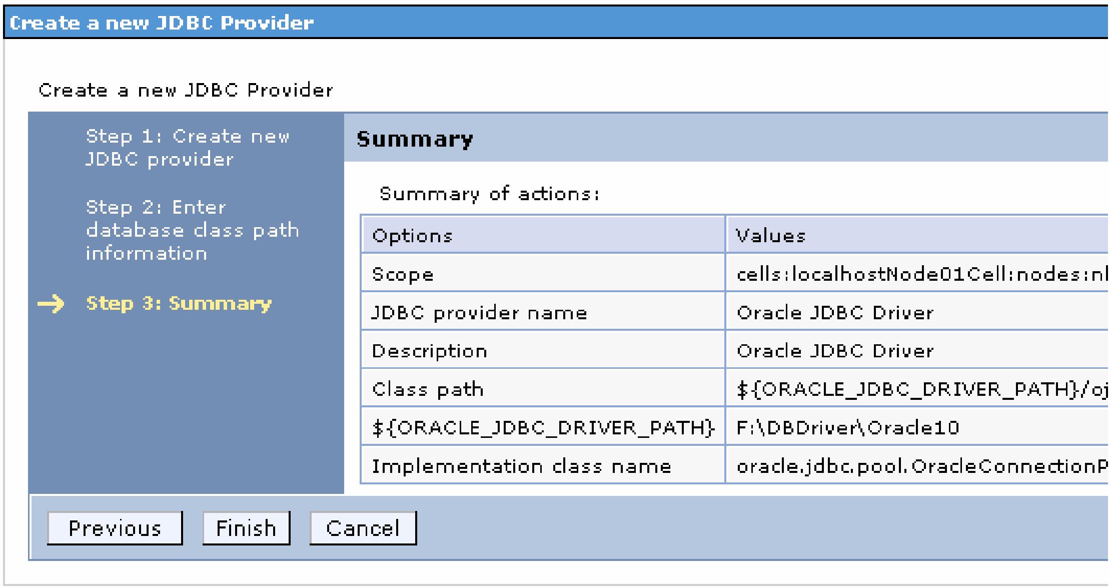
Class path:

${ORACLE_JDBC_DRIVER_PATH}/ojdbc6.jar

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

F:\DBDriver\Oracle10

8. Click **Finish**.



Create a new JDBC Provider

Create a new JDBC Provider

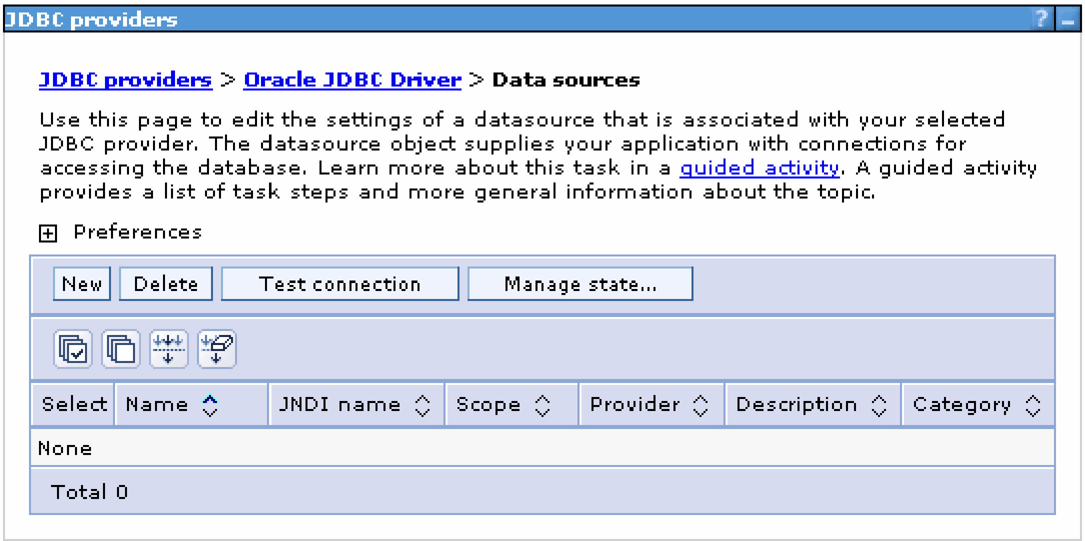
Step 1: Create new JDBC provider
Step 2: Enter database class path information
→ Step 3: Summary

Summary

Summary of actions:

Options	Values
Scope	cells:localhostNode01Cell:nodes:nl
JDBC provider name	Oracle JDBC Driver
Description	Oracle JDBC Driver
Class path	${ORACLE_JDBC_DRIVER_PATH}/oj
${ORACLE_JDBC_DRIVER_PATH}	F:\DBDriver\Oracle10
Implementation class name	oracle.jdbc.pool.OracleConnectionP

9. Under **Additional Properties**, select **Data sources**. Click **New**.



JDBC providers

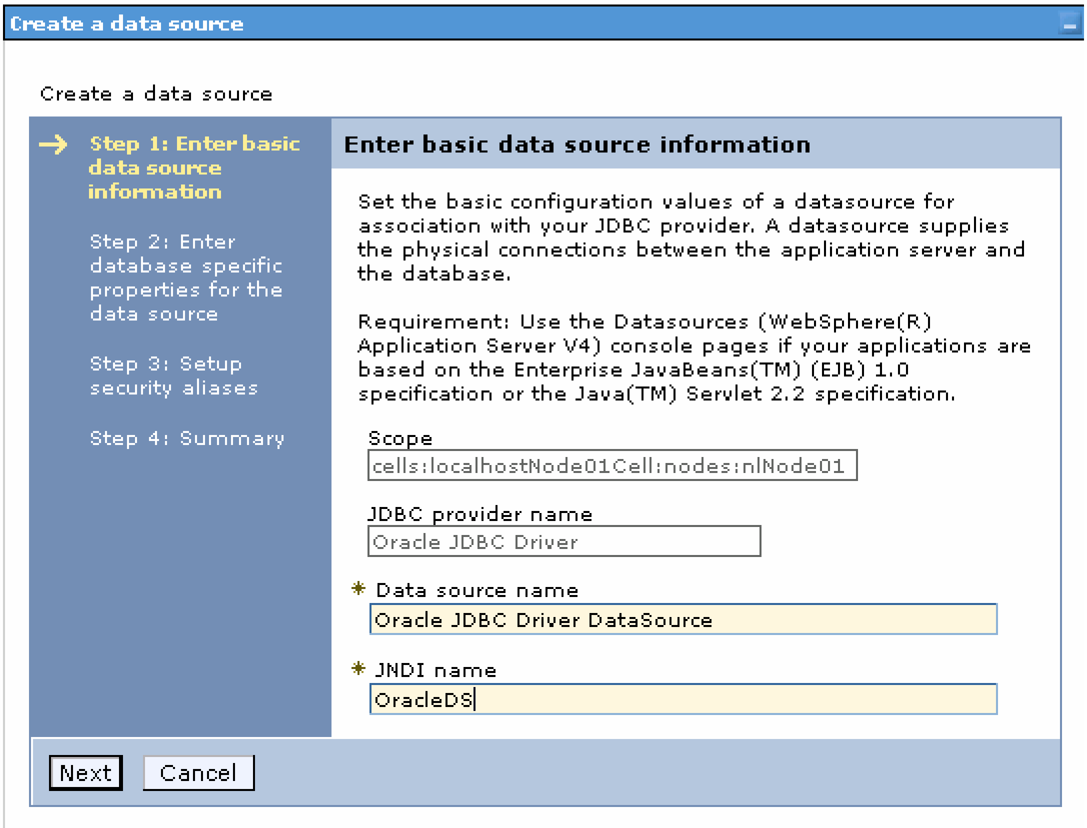
JDBC providers > Oracle JDBC Driver > Data sources

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

Select	Name	JNDI name	Scope	Provider	Description	Category
None						
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**.


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
Next
Cancel

Create a data source

Create a data source

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.

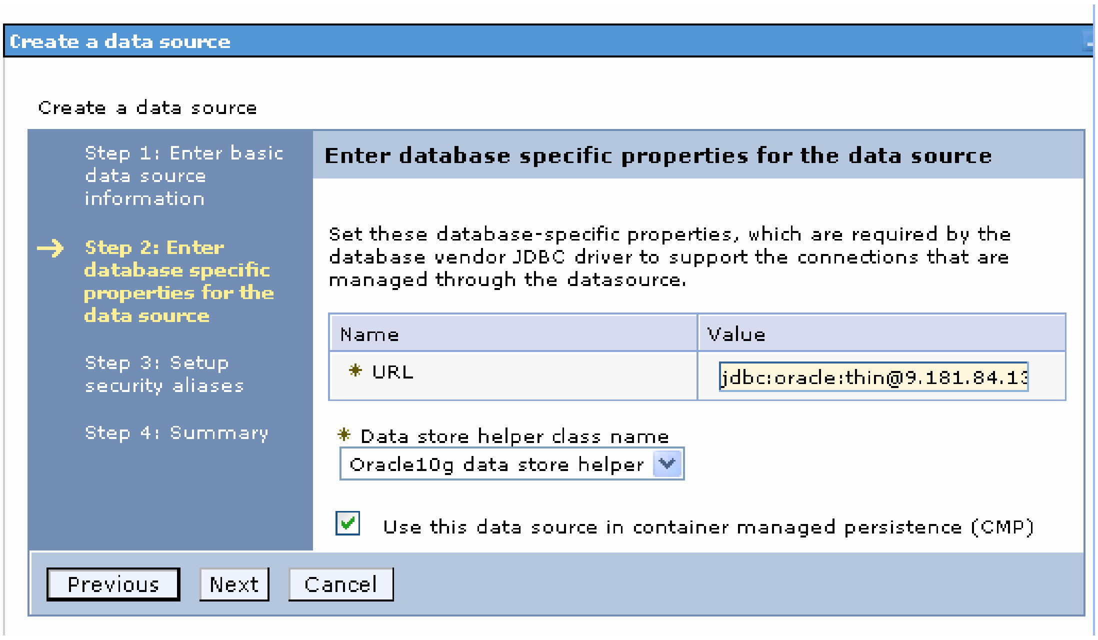
Scope
cells:localhostNode01Cell:nodes:nlNode01

JDBC provider name
Oracle JDBC Driver

* Data source name
Oracle JDBC Driver DataSource

* JNDI name
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

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.

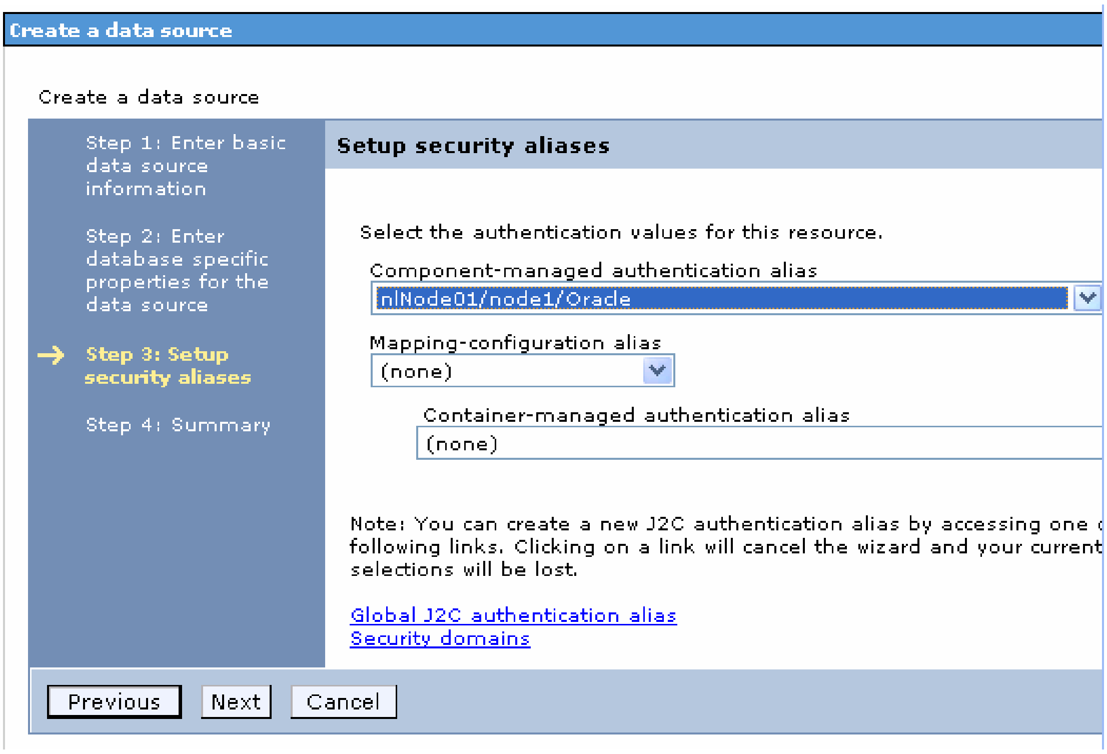
Name	Value
* URL	jdbc:oracle:thin@9.181.84.13

* Data store helper class name

Oracle10g data store helper

☑ Use this data source in container managed persistence (CMP)

12. In the Setup security aliases window, configure the aliases.



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

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

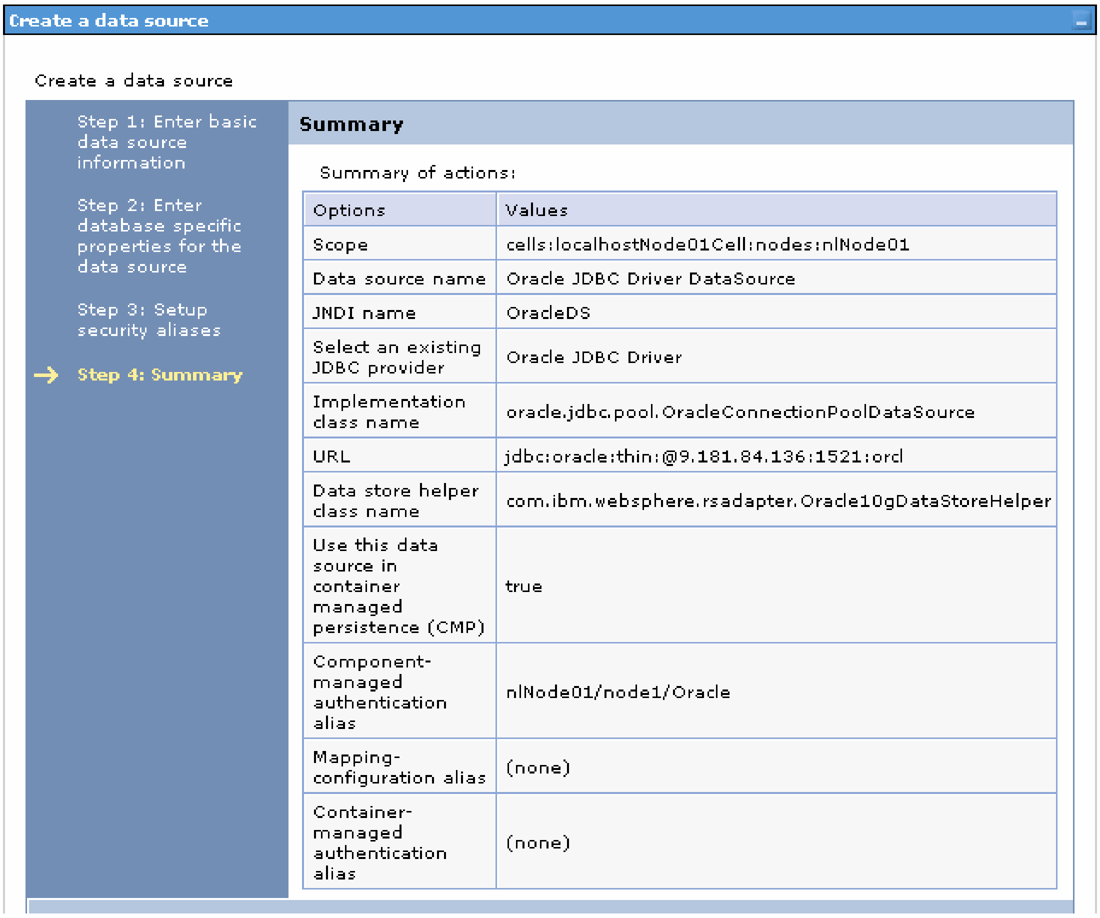
(none)

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

Global J2C authentication alias

Security domains

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



Create a data source

Create a data source

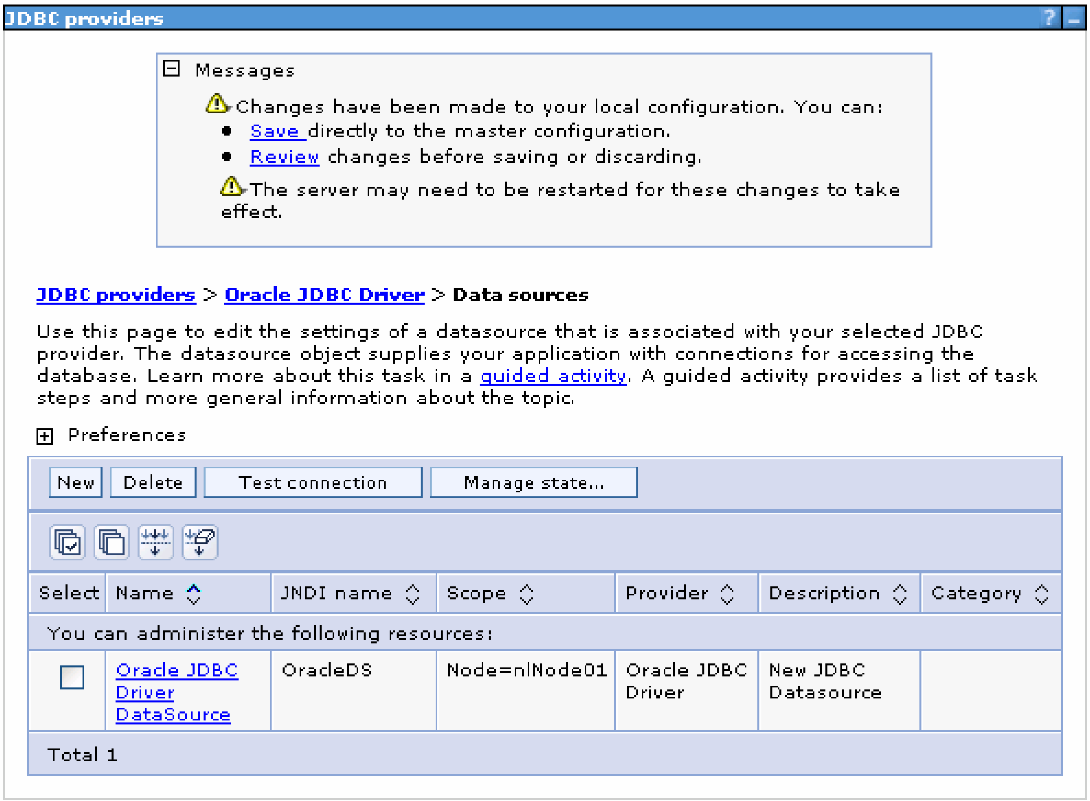
Step 1: Enter basic data source information
Step 2: Enter database specific properties for the data source
Step 3: Setup security aliases
Step 4: Summary

Summary

Summary of actions:

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:orcl
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/node1/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.

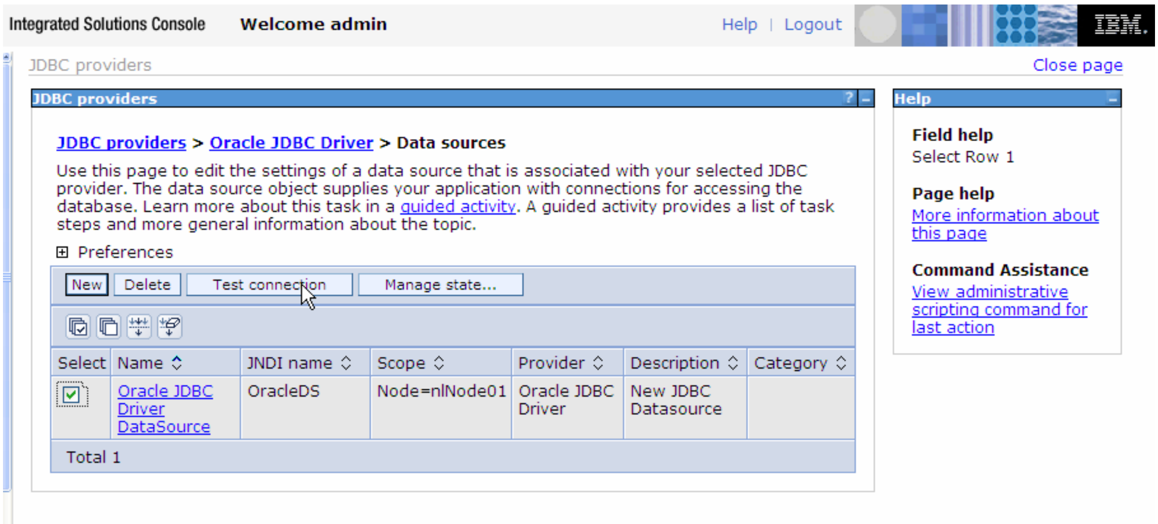
JDBC providers > Oracle JDBC Driver > **Data sources**

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

Select	Name	JNDI name	Scope	Provider	Description	Category
You can administer the following resources:						
□	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

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

Select	Name	JNDI name	Scope	Provider	Description	Category
☑	Oracle JDBC Driver DataSource	OracleDS	Node=nlNode01	Oracle JDBC Driver	New JDBC Datasource	
Total 1						

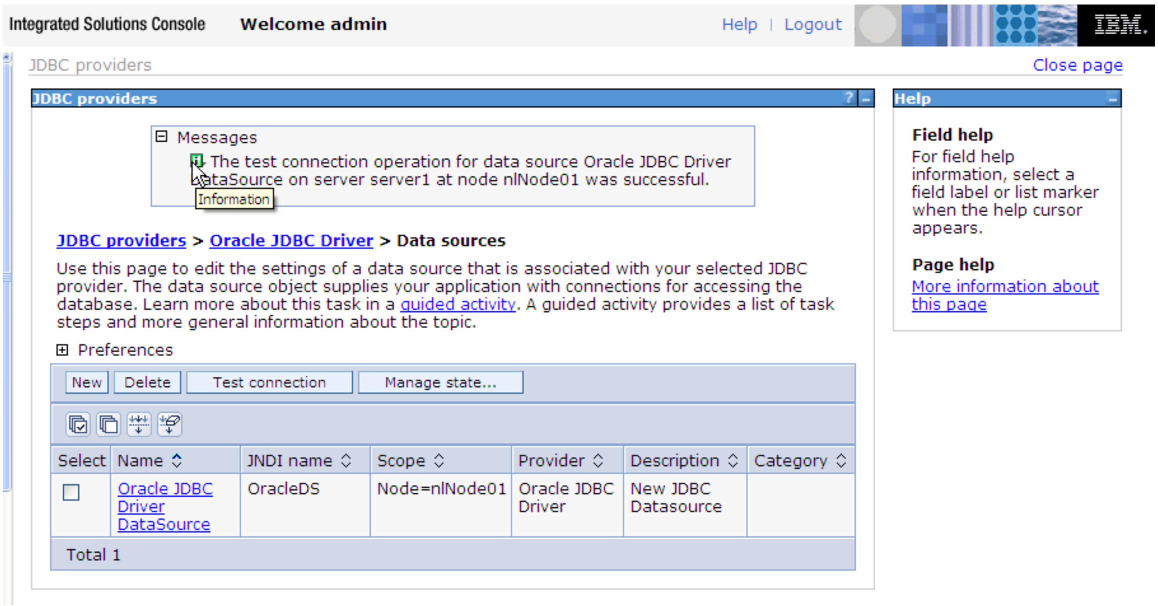
Help

Field help
Select Row 1

Page help
More information about this page

Command Assistance
View administrative scripting command for last action

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.



JDBC providers

JDBC providers

⊟ Messages

The test connection operation for data source Oracle JDBC Driver DataSource on server server1 at node nlNode01 was successful.

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

Select	Name	JNDI name	Scope	Provider	Description	Category
□	Oracle JDBC Driver DataSource	OracleDS	Node=nlNode01	Oracle JDBC Driver	New JDBC Datasource	
Total 1						

Help

Field help
For field help information, select a field label or list marker when the help cursor appears.

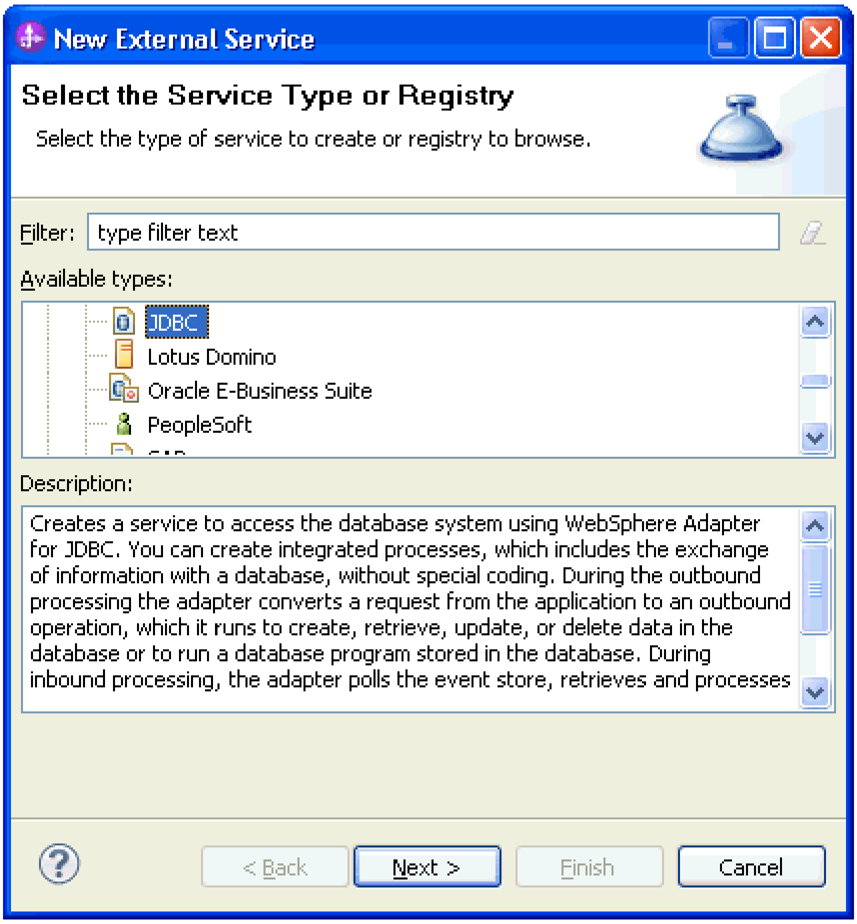
Page help
More information about this page

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 IBM Integration Designer 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**.



New External Service

Select the Service Type or Registry

Select the type of service to create or registry to browse.

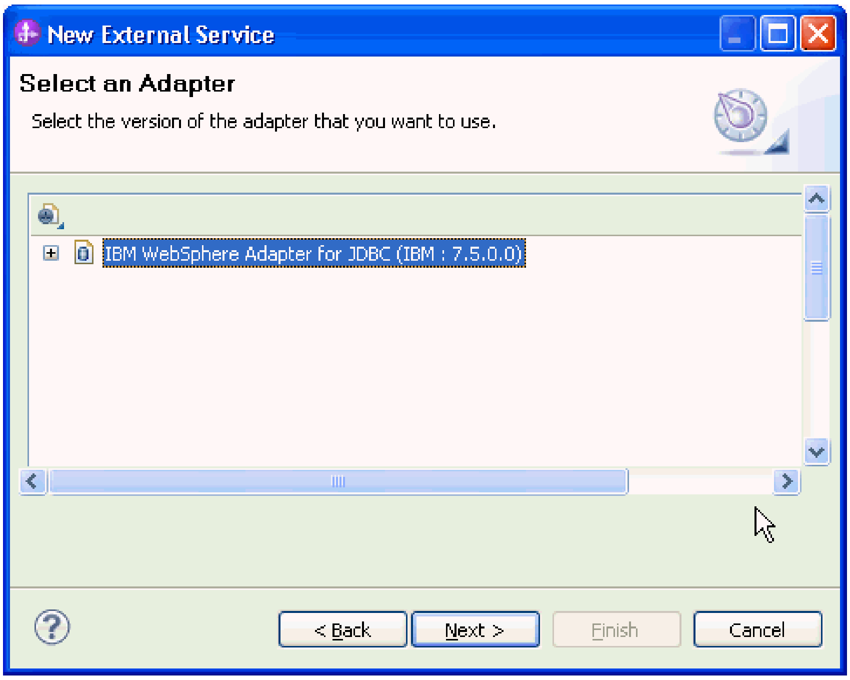
Filter: type filter text

Available types:

Description:

Creates a service to access the database system using WebSphere Adapter for JDBC. You can create integrated processes, which includes the exchange of information with a database, without special coding. During the outbound processing the adapter converts a request from the application to an outbound operation, which it runs to create, retrieve, update, or delete data in the database or to run a database program stored in the database. During inbound processing, the adapter polls the event store, retrieves and processes

4. Select the **IBM WebSphere Adapter for JDBC (IBM: 7.5.0.0)** and click **Next.**



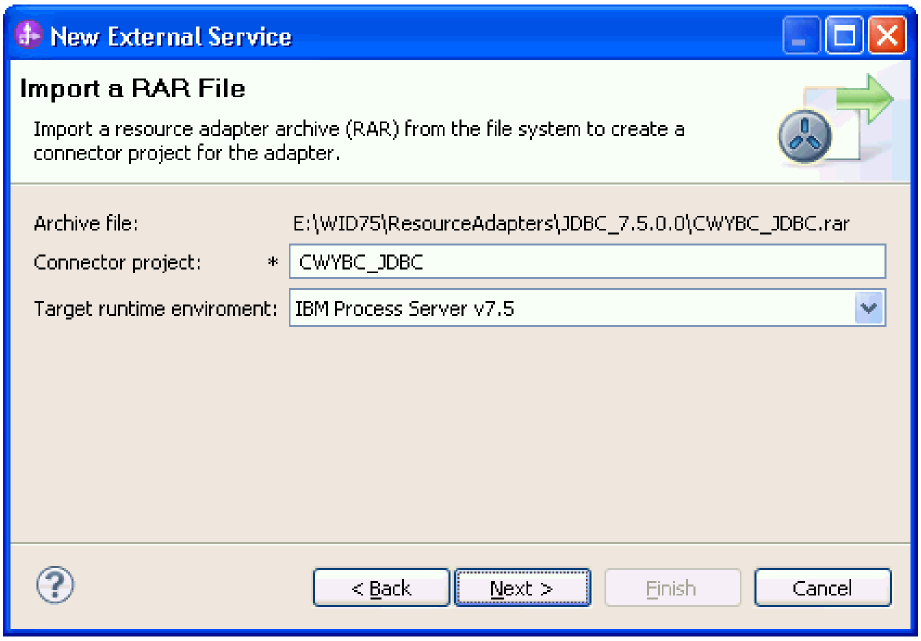
New External Service

Select an Adapter

Select the version of the adapter that you want to use.

5. In the **Connector project** field enter **CWYBC_JDBC**.

6. In the **Target runtime environment** field, select appropriate runtime and click **Next**.



New External Service

Import a RAR File

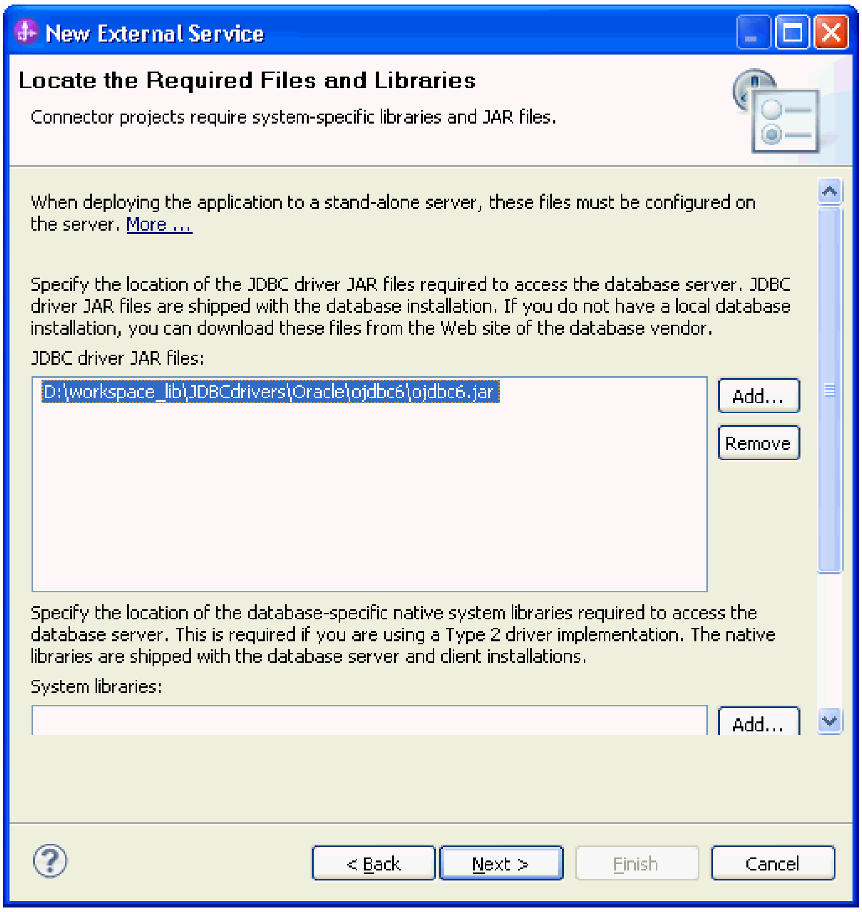
Import a resource adapter archive (RAR) from the file system to create a connector project for the adapter.

Archive file: E:\WID75\ResourceAdapters\JDBC_7.5.0.0\CWYBC_JDBC.rar

Connector project: * CWYBC_JDBC

Target runtime enviroment: IBM Process Server v7.5

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**.



New External Service

Locate the Required Files and Libraries

Connector projects require system-specific libraries and JAR files.

When deploying the application to a stand-alone server, these files must be configured on the server. More ...

Specify the location of the JDBC driver JAR files required to access the database server. JDBC driver JAR files are shipped with the database installation. If you do not have a local database installation, you can download these files from the Web site of the database vendor.

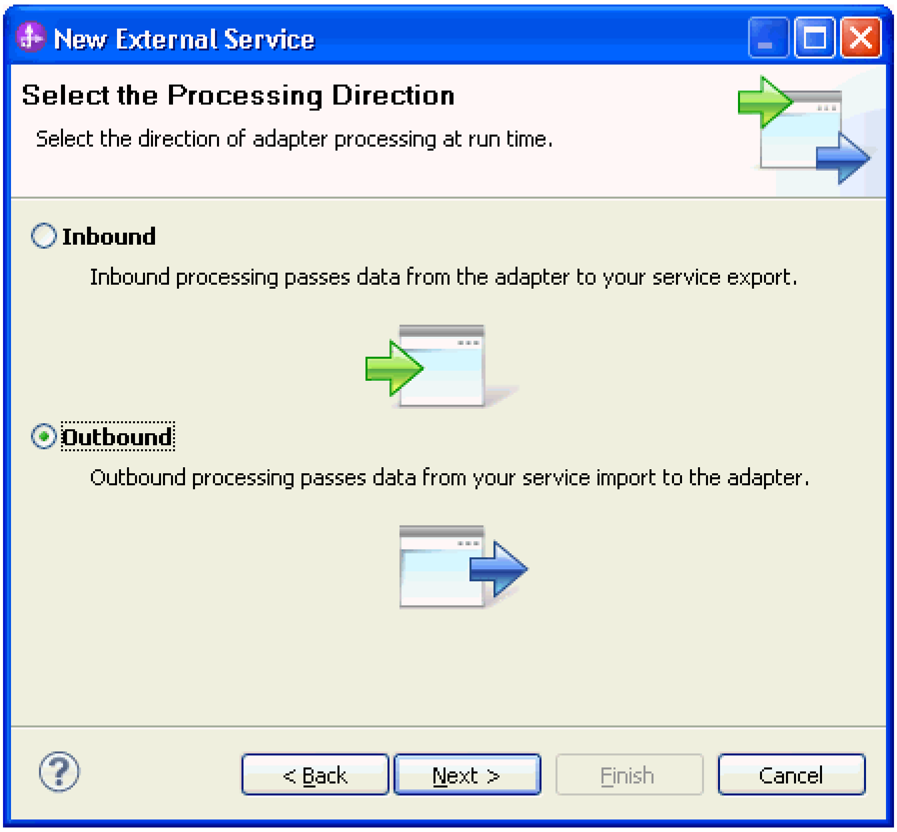
JDBC driver JAR files:

D:\workspace_lib\JDBCdrivers\Oracle\ojdbc6\ojdbc6.jar

Specify the location of the database-specific native system libraries required to access the database server. This is required if you are using a Type 2 driver implementation. The native libraries are shipped with the database server and client installations.

System libraries:

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



New External Service

Select the Processing Direction

Select the direction of adapter processing at run time.

○ Inbound

Inbound processing passes data from the adapter to your service export.

● Outbound

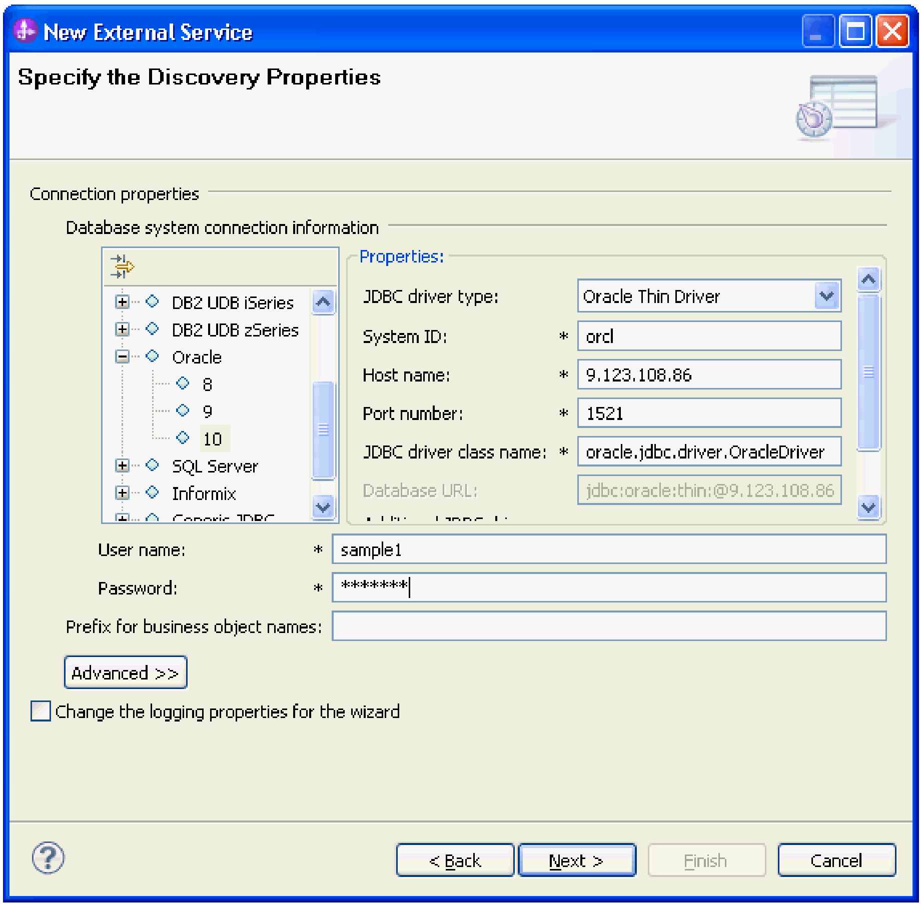
Outbound processing passes data from your service import to the adapter.

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**.



New External Service

Specify the Discovery Properties

Connection properties

Database system connection information

- DB2 UDB iSeries
- DB2 UDB zSeries
- Oracle
  - 8
  - 9
  - 10
- SQL Server
- Informix
- Generic JDBC

Properties:

JDBC driver type: Oracle Thin Driver

System ID: * orcl

Host name: * 9.123.108.86

Port number: * 1521

JDBC driver class name: * oracle.jdbc.driver.OracleDriver

Database URL: jdbc:oracle:thin:@9.123.108.86

User name: * sample1

Password: * *******

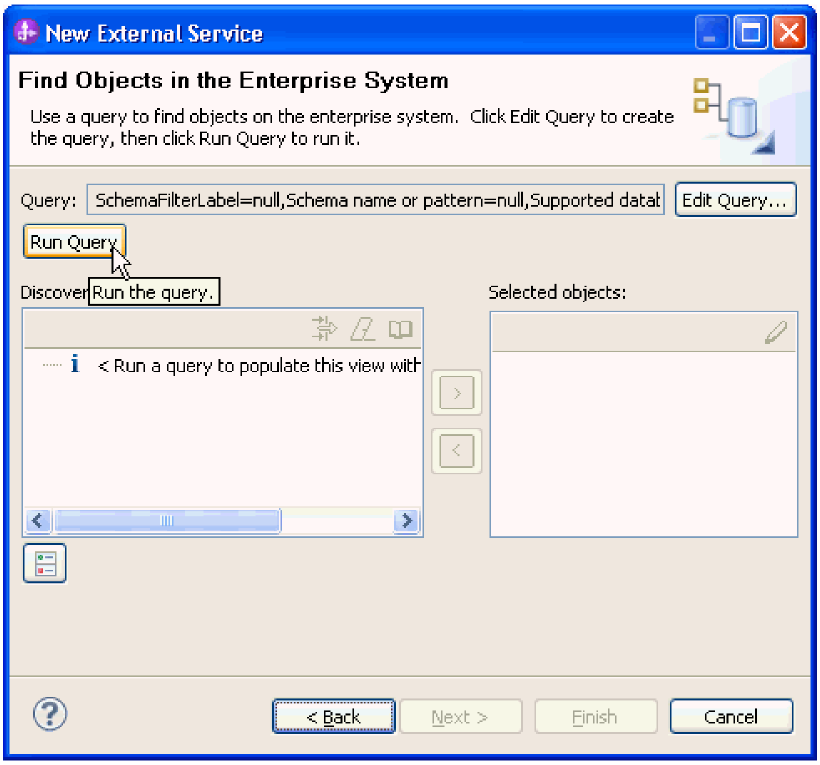
Prefix for business object names:

Advanced >>

□ Change the logging properties for the wizard

Select the business objects to be used with the adapter

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



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 datat

Edit Query...

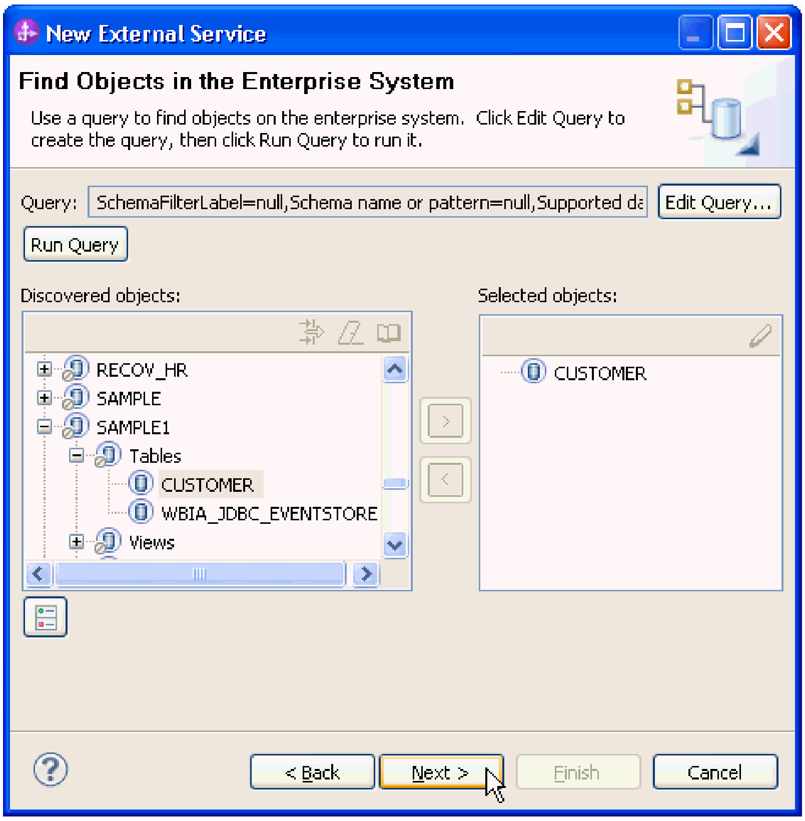
Run Query

Discover Run the query.

Selected objects:

< Run a query to populate this view with

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**.



New External Service

Find Objects in the Enterprise System

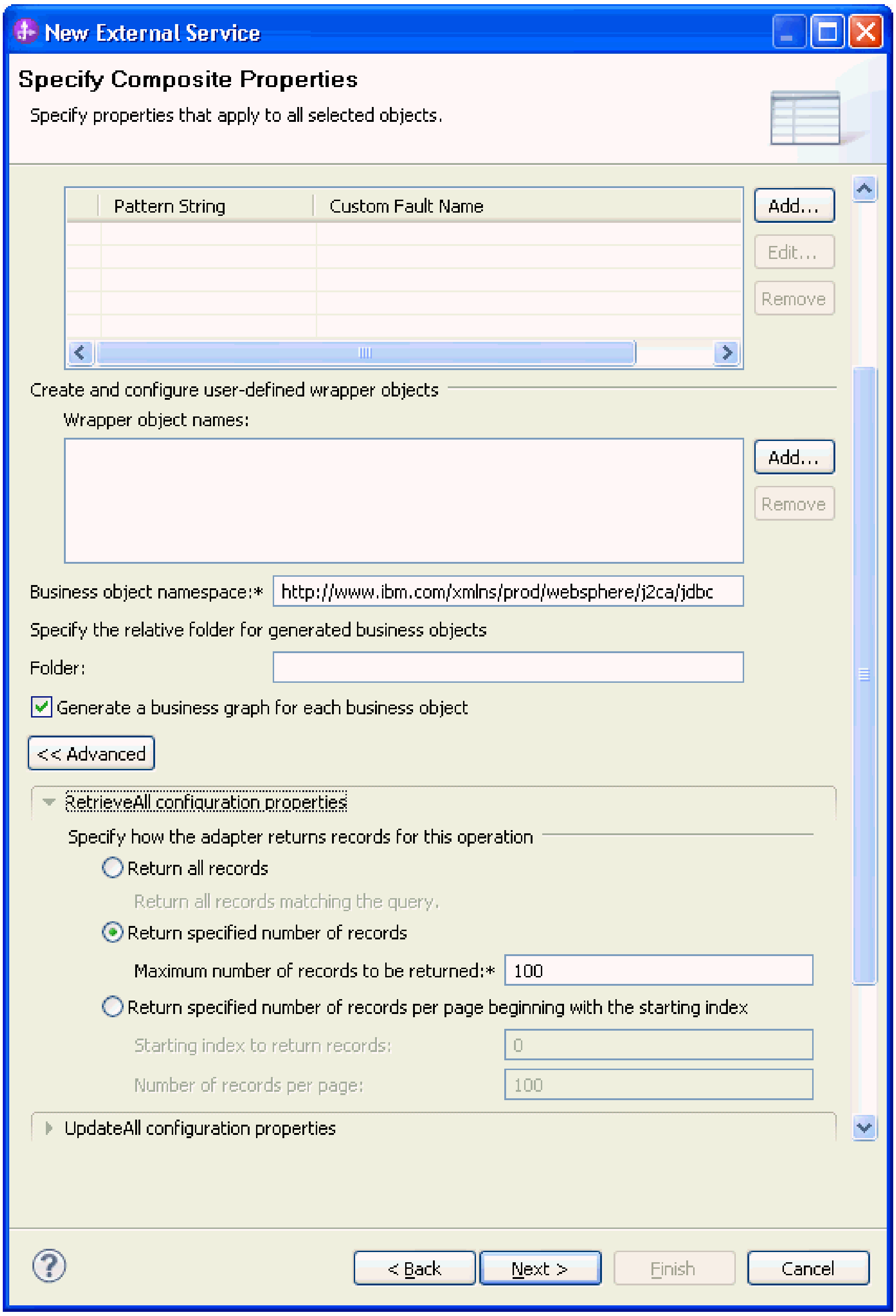
Use a query to find objects on the enterprise system. Click Edit Query to create the query, then click Run Query to run it.

Query: SchemaFilterLabel=null,Schema name or pattern=null,Supported da

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** fields check box selected and click **Next**.



New External Service

Specify Composite Properties

Specify properties that apply to all selected objects.

Pattern String	Custom Fault Name

Create and configure user-defined wrapper objects

Wrapper object names:

Business object namespace:* http://www.ibm.com/xmlns/prod/websphere/j2ca/jdbc

Specify the relative folder for generated business objects

Folder:

☑ Generate a business graph for each business object

RetrieveAll configuration properties

Specify how the adapter returns records for this operation

○ Return all records

Return all records matching the query.

● Return specified number of records

Maximum number of records to be returned:* 100

○ Return specified number of records per page beginning with the starting index

Starting index to return records: 0

Number of records per page: 100

UpdateAll configuration properties

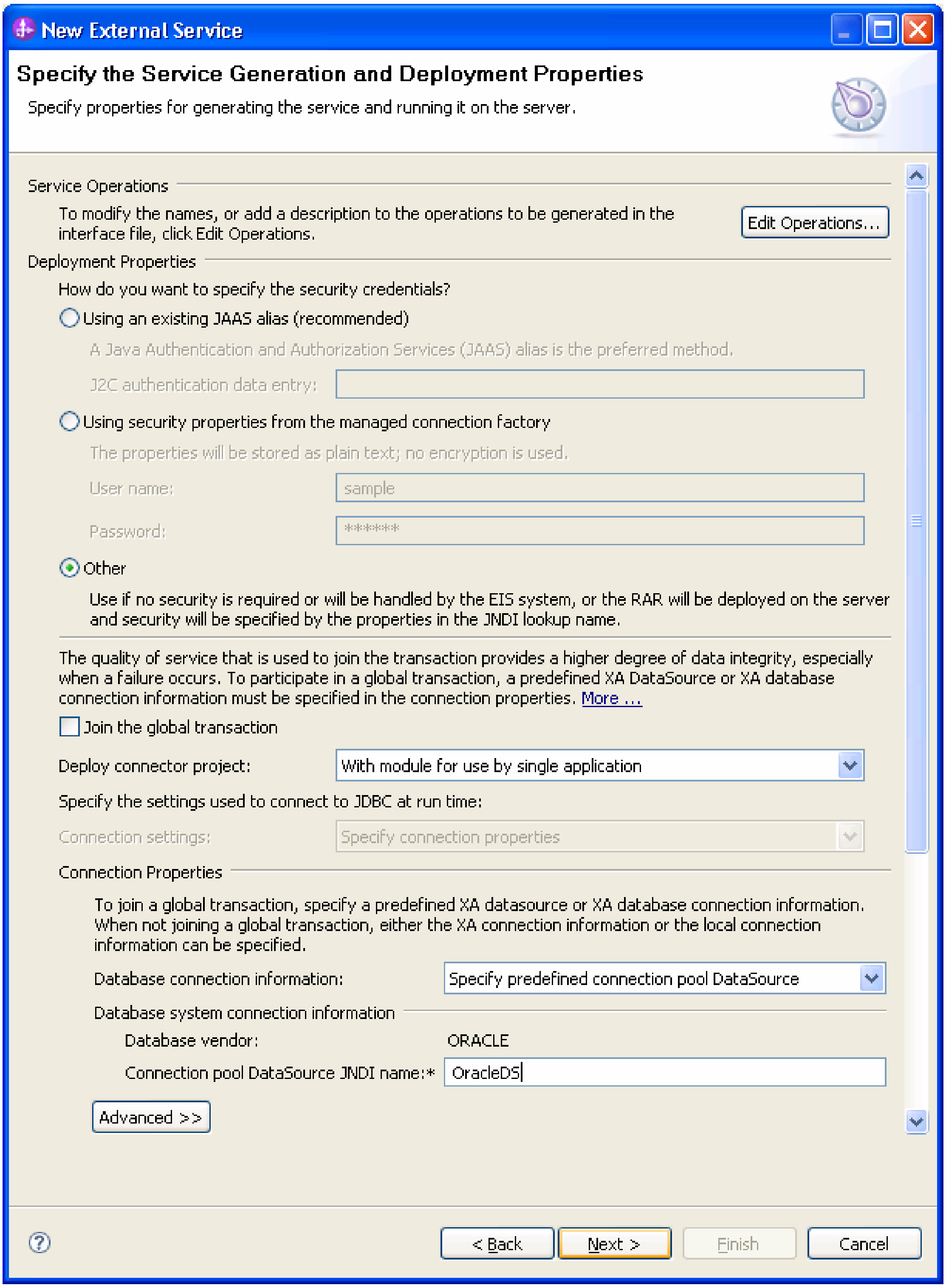
2. In the Specify the Service Generation and Deployment Properties window, perform the following steps:

   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**.



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 managed connection factory

The properties will be stored as plain text; no encryption is used.

User name: sample

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: With module for use by single application

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

Connection settings: Specify connection properties

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.

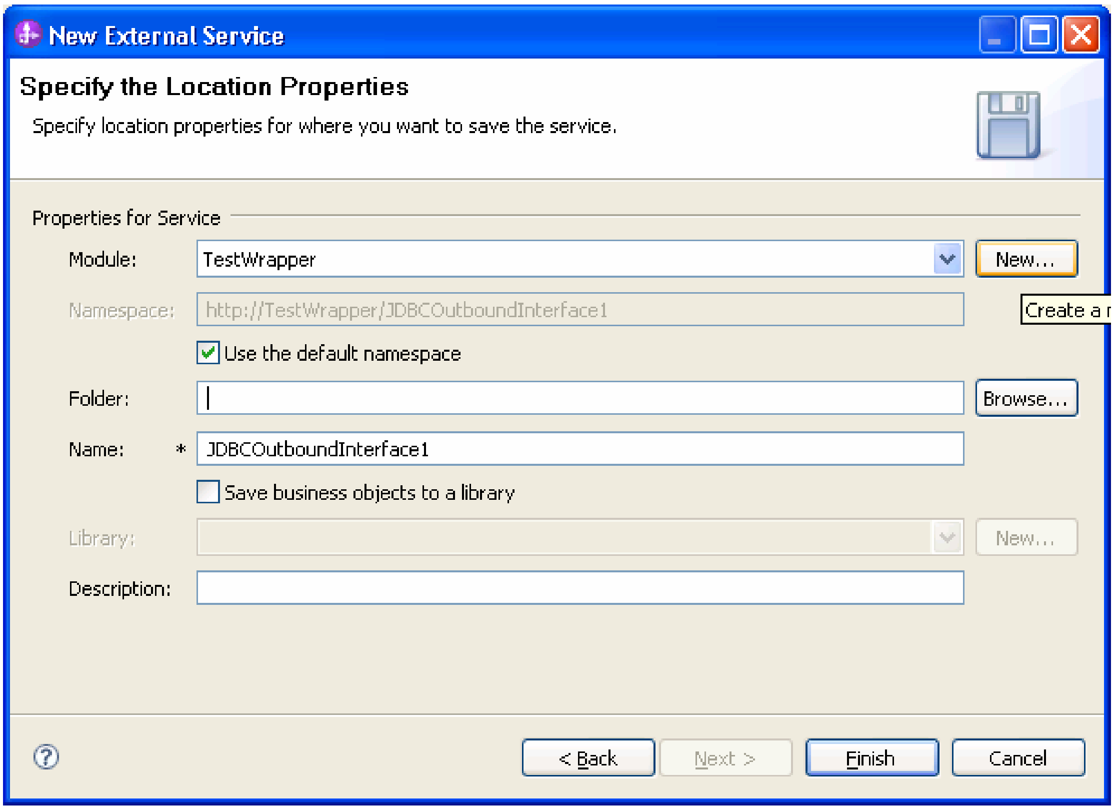
Database connection information: Specify predefined connection pool DataSource

Database system connection information

Database vendor: ORACLE

Connection pool DataSource JNDI name:* OracleDS

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



Specify the Location Properties

Specify location properties for where you want to save the service.

Properties for Service

Module: TestWrapper

Namespace: http://TestWrapper/JDBCOutboundInterface1

☑ Use the default namespace

Folder:

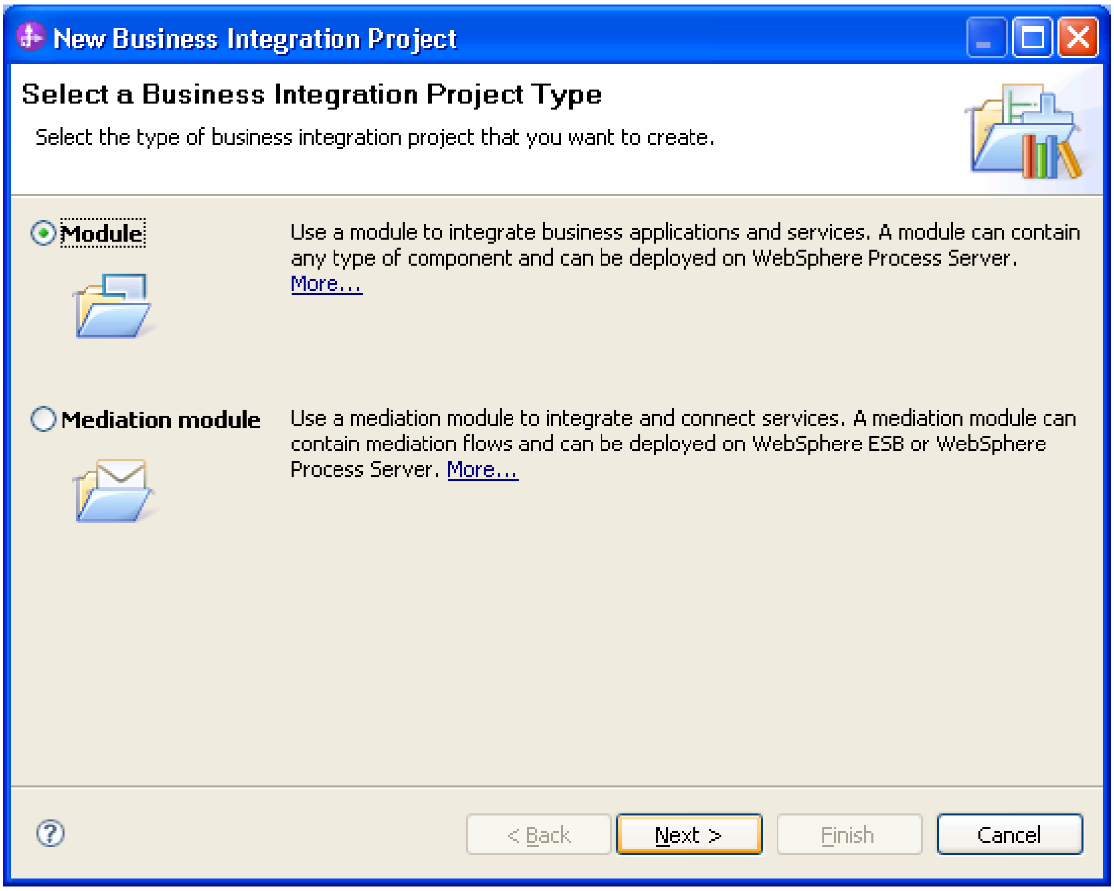
Name: * JDBCOutboundInterface1

□ Save business objects to a library

Library:

Description:

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



Select a Business Integration Project Type

Select the type of business integration project that you want to create.

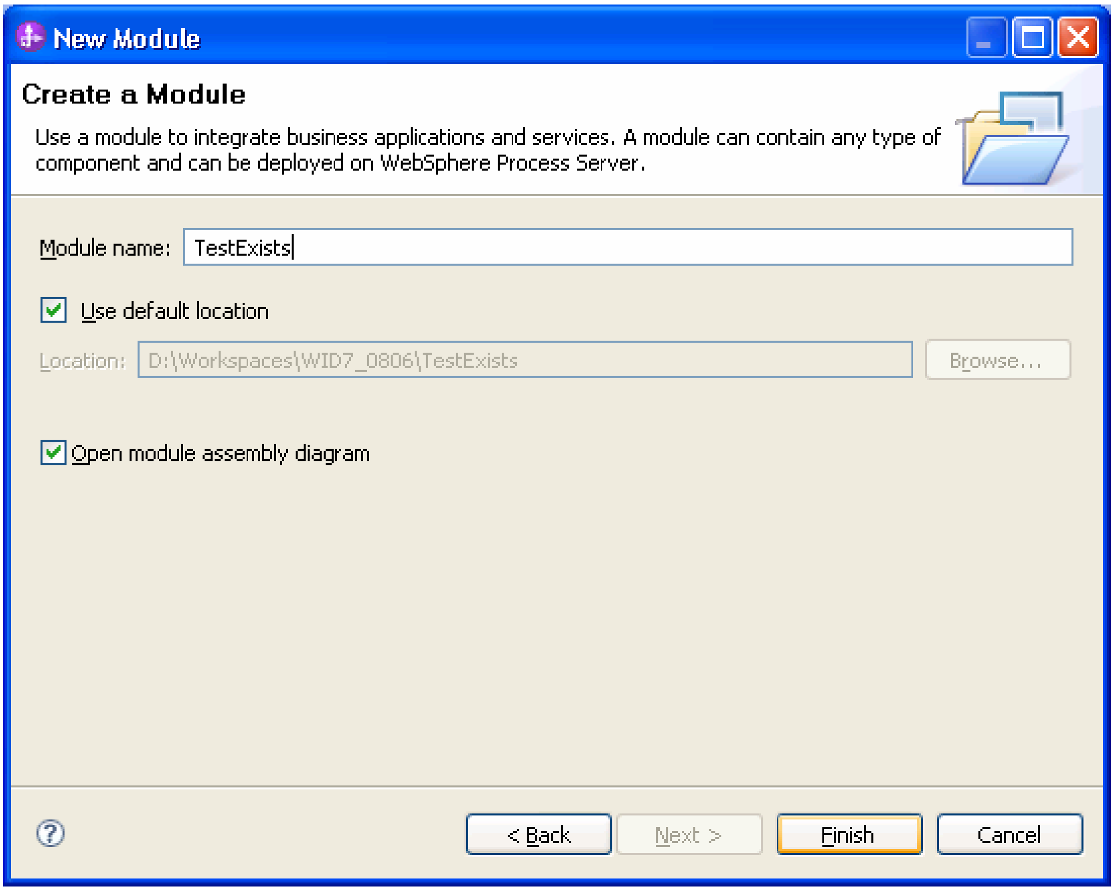
● Module

Use a module to integrate business applications and services. A module can contain any type of component and can be deployed on WebSphere Process Server. More...

○ Mediation module

Use a mediation module to integrate and connect services. A mediation module can contain mediation flows and can be deployed on WebSphere ESB or WebSphere Process Server. More...

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



New Module

Create a Module

Use a module to integrate business applications and services. A module can contain any type of component and can be deployed on WebSphere Process Server.

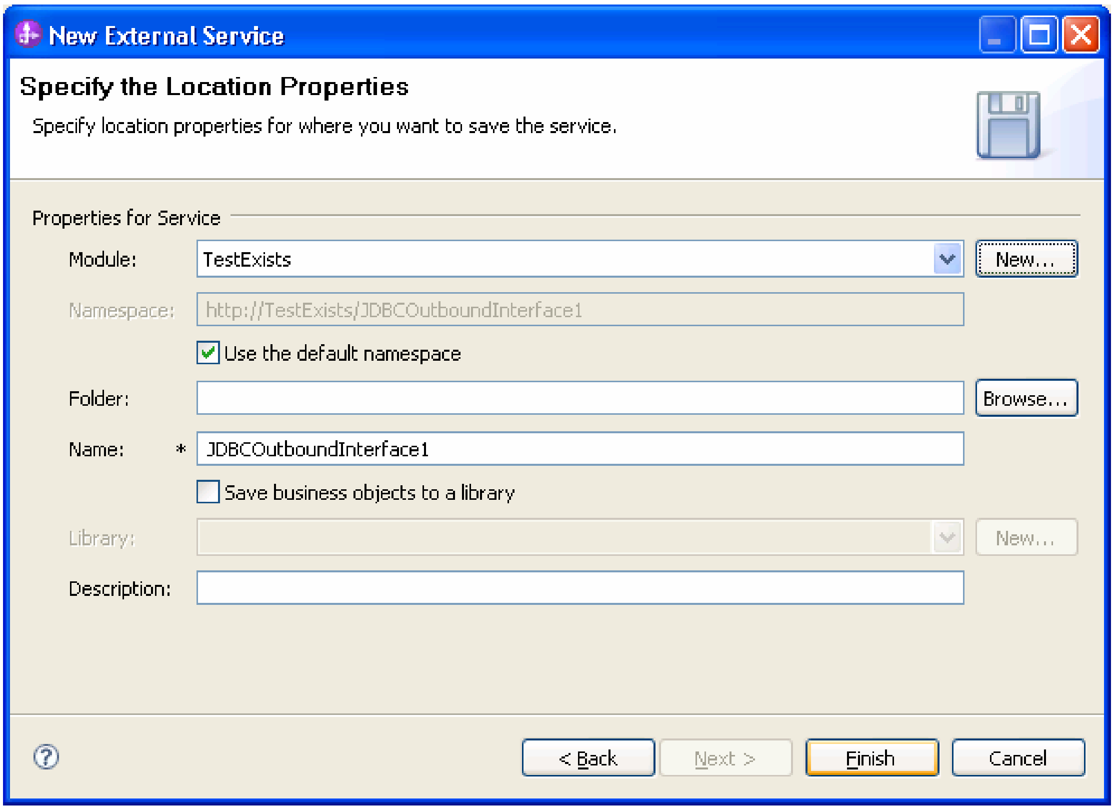
Module name: TestExists

☑ Use default location

Location: D:\Workspaces\WID7_0806\TestExists

☑ Open module assembly diagram

6. In the Specify the Location Properties window, click **Finish** to finish the service creation.



New External Service

Specify the Location Properties

Specify location properties for where you want to save the service.

Properties for Service

Module: TestExists

Namespace: http://TestExists/JDBCOutboundInterface1

☑ Use the default namespace

Folder:

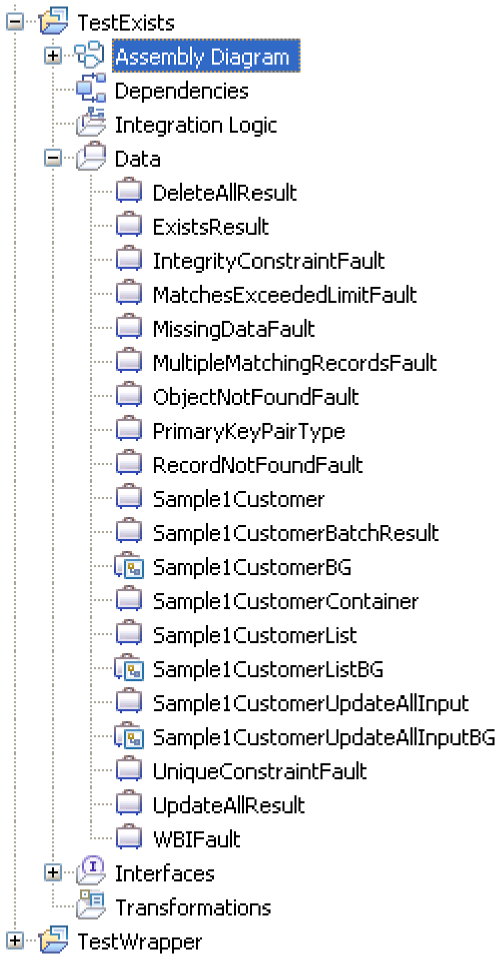
Name: * JDBCOutboundInterface1

□ Save business objects to a library

Library:

Description:

7. Open the Project Explorer and verify that the business objects are created correctly.

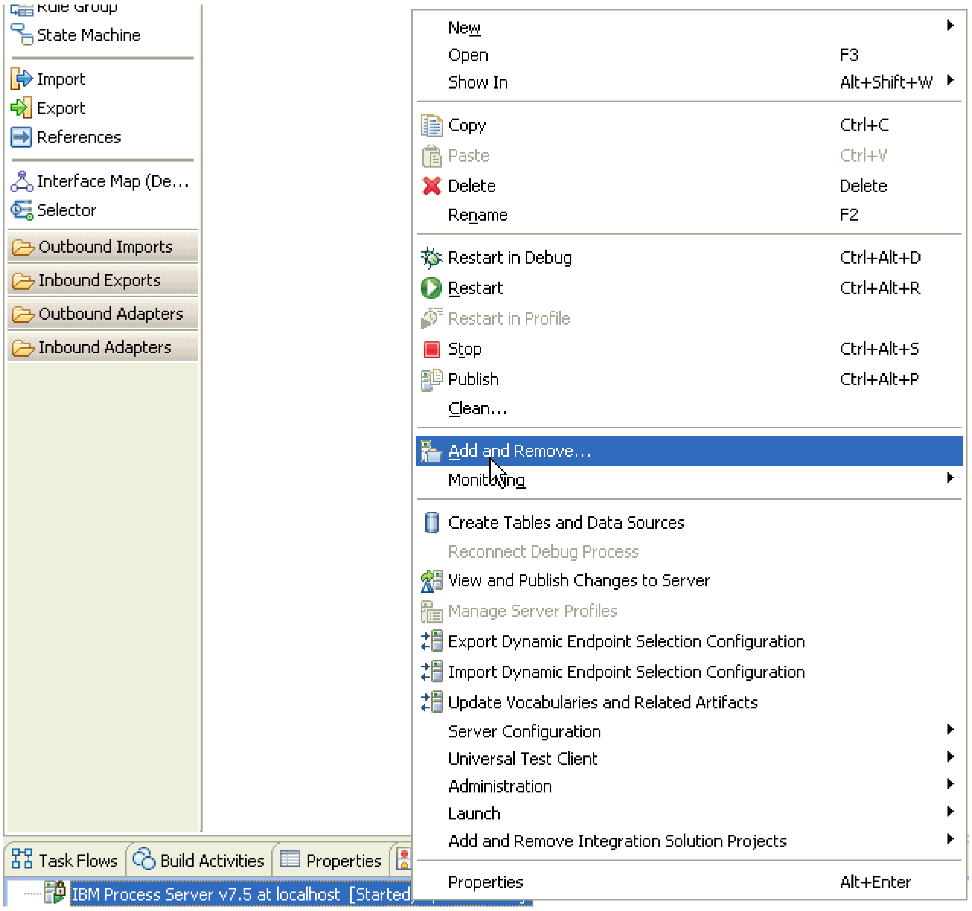


Deploy the module to the test environment

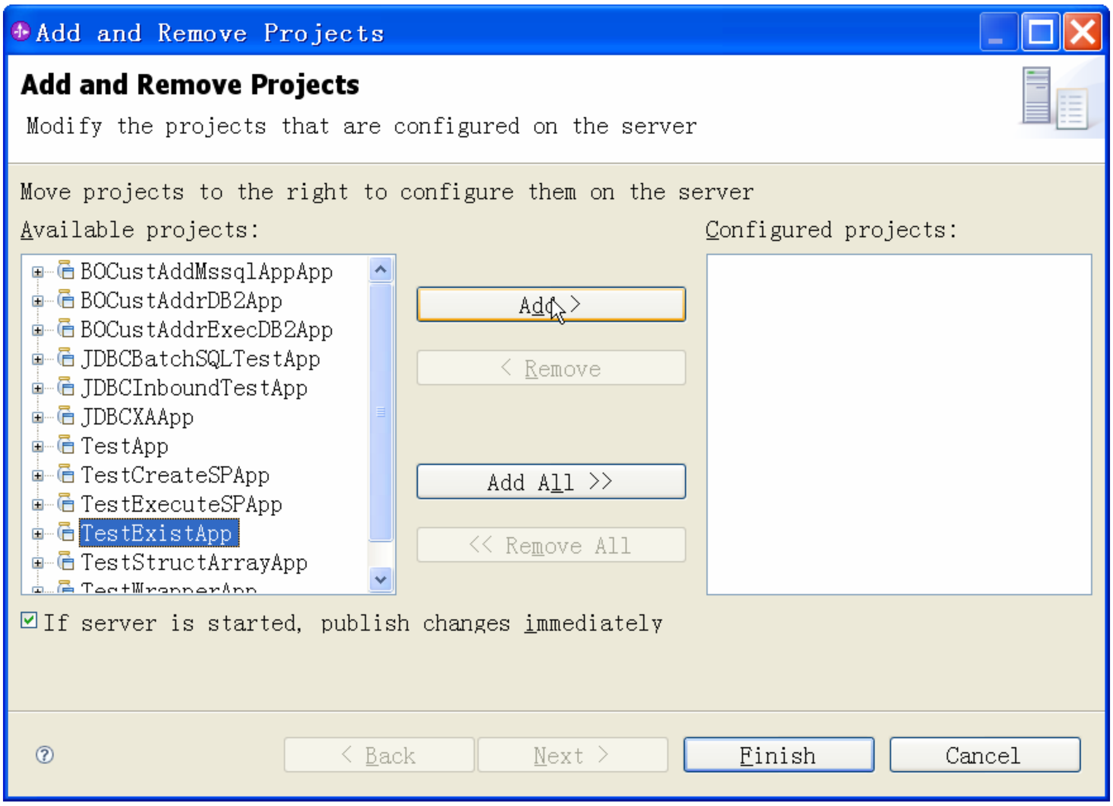
After running the external service wizard, you will have an SCA module that contains an Enterprise Information System (EIS) import. You must install this SCA module in the IBM Integration Designer integration test client to deploy it. To do this, you must add the SCA module you created earlier to the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

Steps for adding the SCA module to the server:

1. In IBM Integration Designer, 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**.



The Add and Remove Projects window lists the available projects in the IBM Integration Designer workspace.


Add and Remove Projects

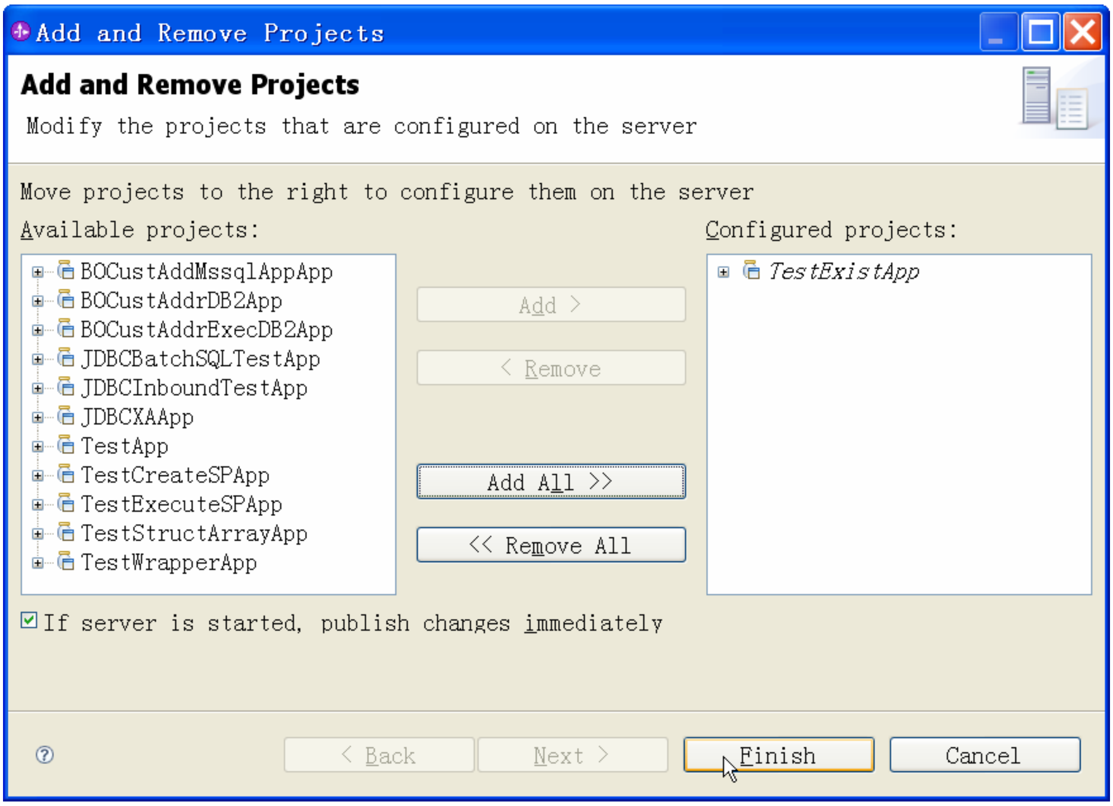
Add and Remove Projects

Modify the projects that are configured on the server

Move projects to the right to configure them on the server

☑ If server is started, publish changes immediately

4. In the Add and Remove Projects window, select your project (TestExistsApp) and click **Add** to configure the project on the server. Click **Finish.**



Add and Remove Projects

Add and Remove Projects

Modify the projects that are configured on the server

Move projects to the right to configure them on the server

Available projects:

- BOCustAddMssqlAppApp
- BOCustAddrDB2App
- BOCustAddrExecDB2App
- JDBCBatchSQLTestApp
- JDBCInboundTestApp
- JDBCXAApp
- TestApp
- TestCreateSPApp
- TestExecuteSPApp
- TestStructArrayApp
- TestWrapperApp

Configured projects:

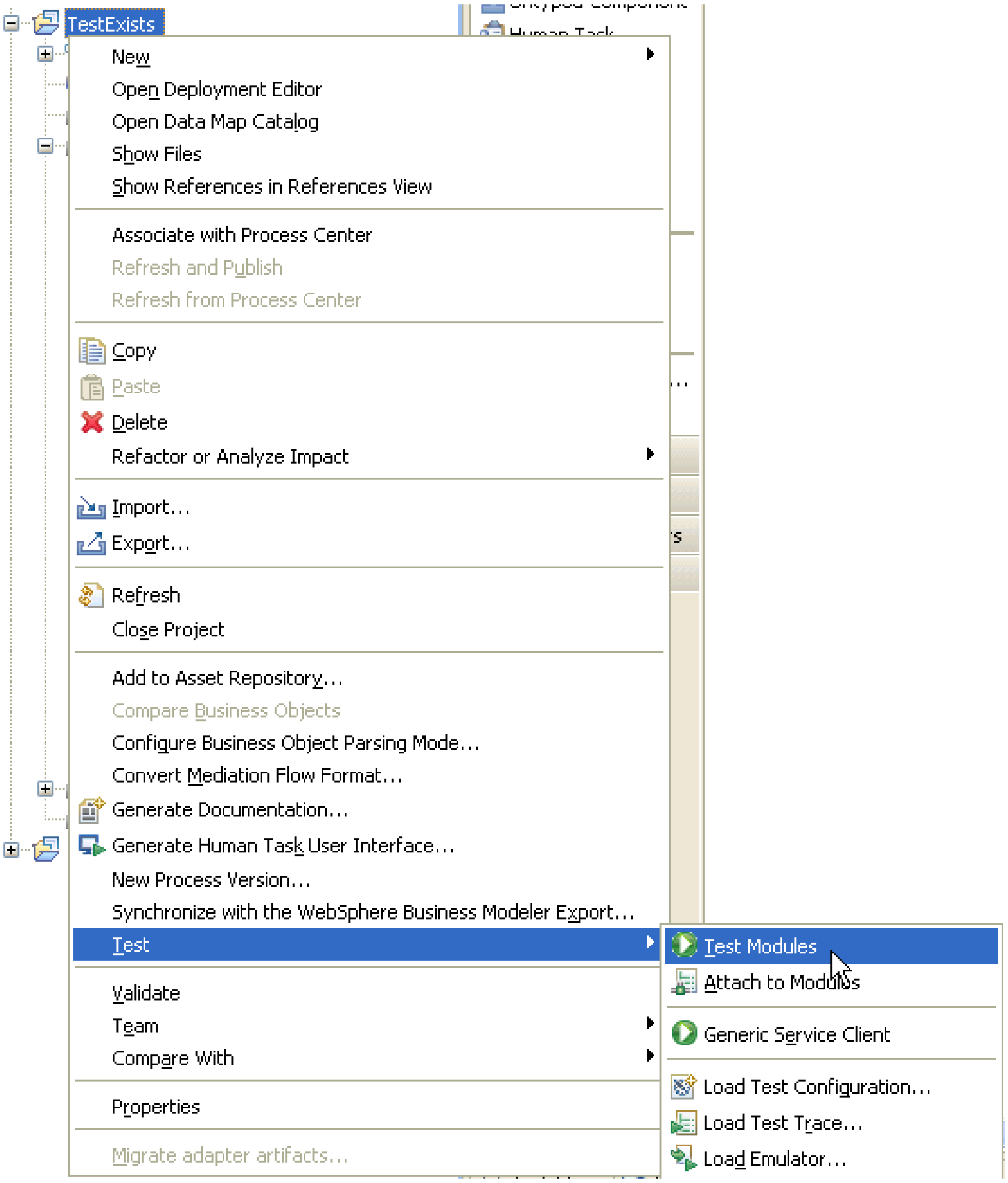
- *TestExistsApp*

☑ If server is started, publish changes immediately

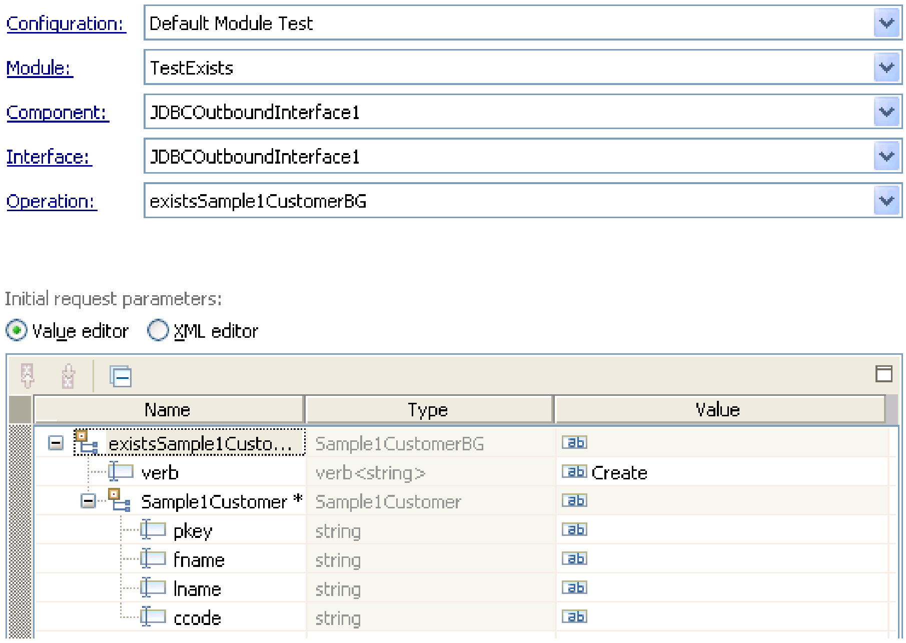
Test the assembled adapter application

Test the assembled adapter application using the IBM Integration Designer integration test client.

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


TestExists
New
Open Deployment Editor
Open Data Map Catalog
Show Files
Show References in References View
Associate with Process Center
Refresh and Publish
Refresh from Process Center
Copy
Paste
Delete
Refactor or Analyze Impact
Import...
Export...
Refresh
Close Project
Add to Asset Repository...
Compare Business Objects
Configure Business Object Parsing Mode...
Convert Mediation Flow Format...
Generate Documentation...
Generate Human Task User Interface...
New Process Version...
Synchronize with the WebSphere Business Modeler Export...
Test
Validate
Team
Compare With
Properties
Migrate adapter artifacts...
Test Modules
Attach to Modules
Generic Service Client
Load Test Configuration...
Load Test Trace...
Load Emulator...

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



Configuration: Default Module Test

Module: TestExists

Component: JDBCOutboundInterface1

Interface: JDBCOutboundInterface1

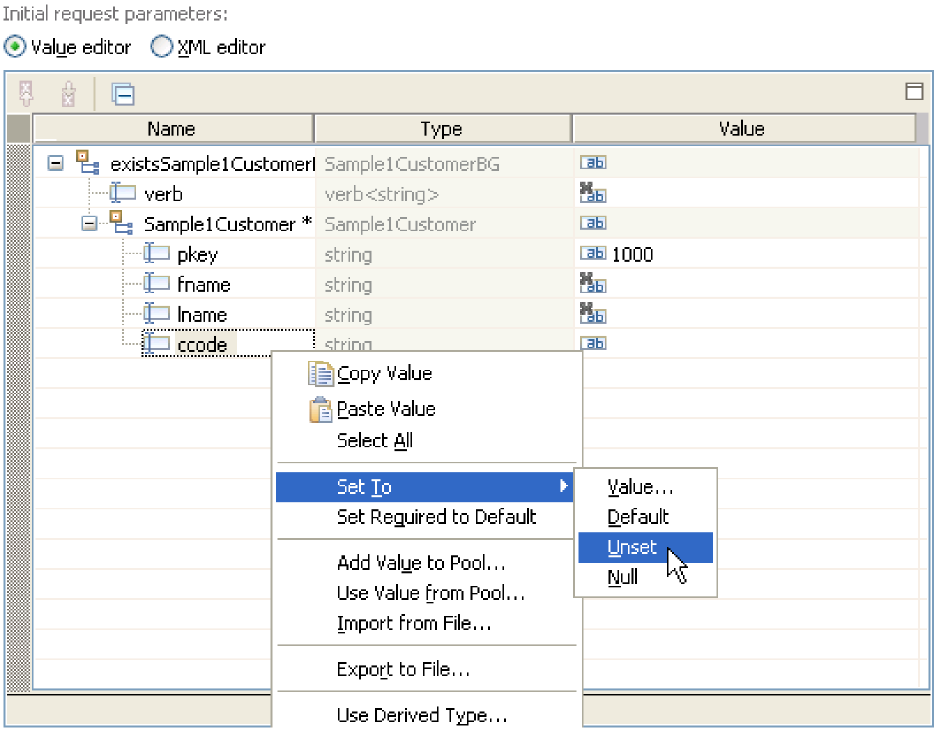
Operation: existsSample1CustomerBG

Initial request parameters:

● Value editor ○ XML editor

Name	Type	Value
existsSample1Custo...	Sample1CustomerBG	
verb	verb<string>	Create
Sample1Customer *	Sample1Customer	
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.



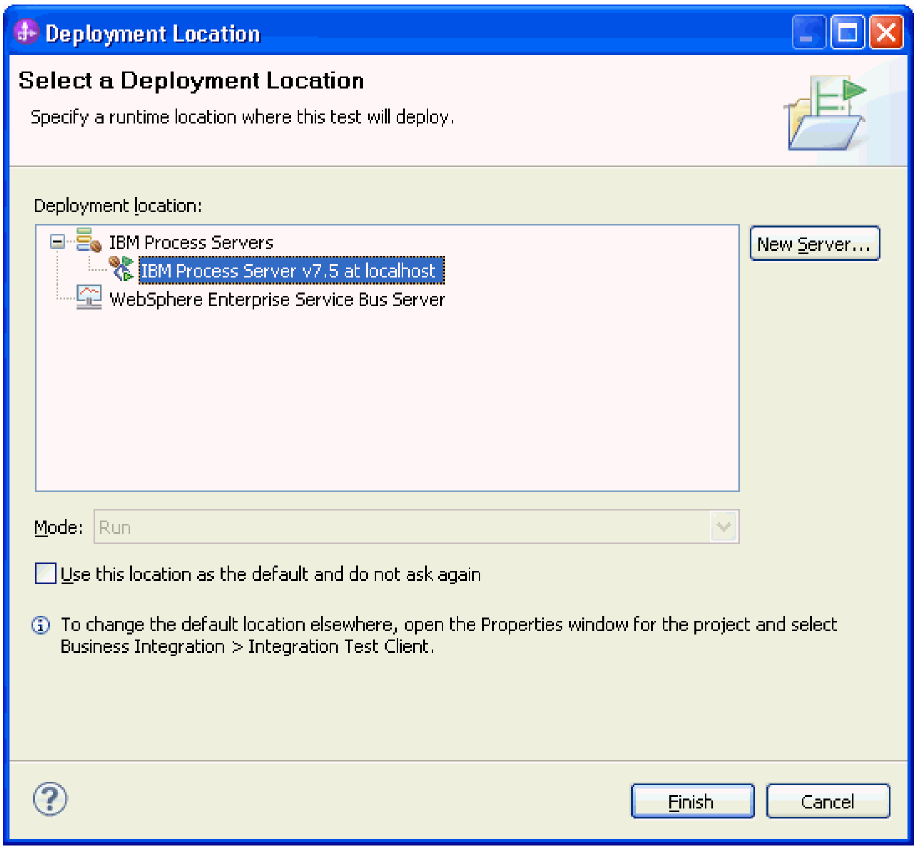
Initial request parameters:

● Value editor ○ XML editor

Name	Type	Value
existsSample1CustomerI	Sample1CustomerBG	
verb	verb<string>	
Sample1Customer *	Sample1Customer	
pkey	string	1000
fname	string	
lname	string	
ccode	string	

4. To execute the service, click Continue ▶.

5. 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:

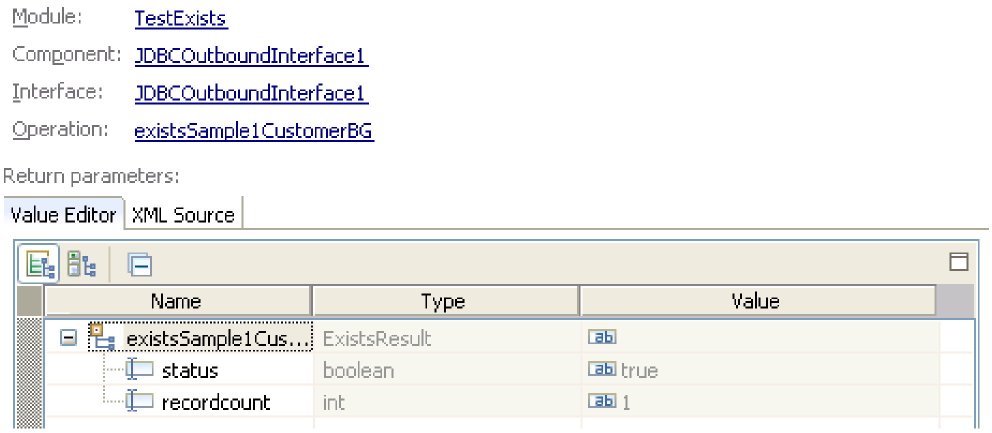
- IBM Process Servers
  - IBM Process Server v7.5 at localhost
- WebSphere Enterprise Service Bus Server

Mode: Run

□ Use this location as the default and do not ask again

To change the default location elsewhere, open the Properties window for the project and select Business Integration > Integration Test Client.

6. Check the return value to ensure it matches expected values.



Module: TestExists

Component: JDBCOutboundInterface1

Interface: JDBCOutboundInterface1

Operation: existsSample1CustomerBG

Return parameters:

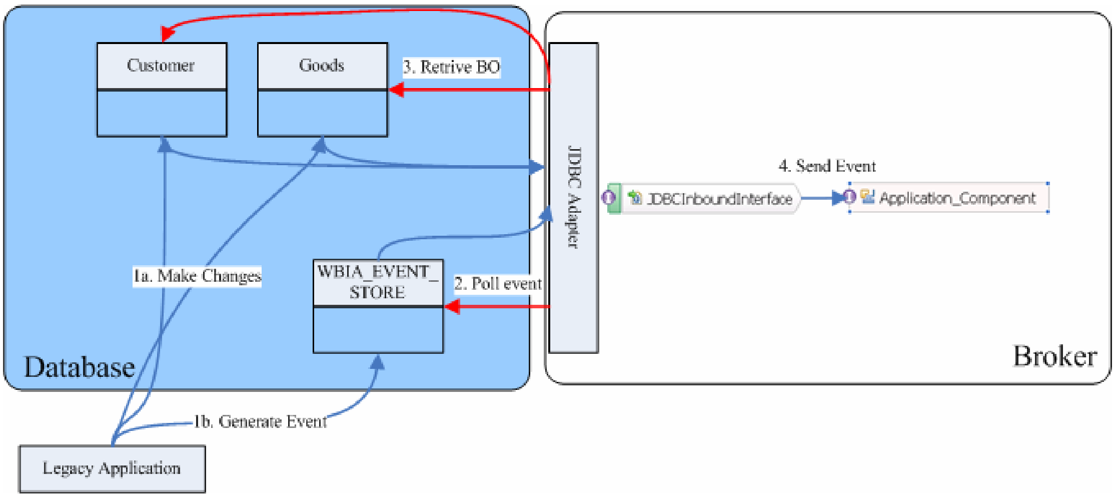
Name	Type	Value
existsSample1Cus...	ExistsResult	
status	boolean	true
recordcount	int	1

Chapter 15. Tutorial 14: Generate Wrapper business objects for Inbound (Oracle)

This tutorial demonstrates how WebSphere Adapter for JDBC 7.5.0.0 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.

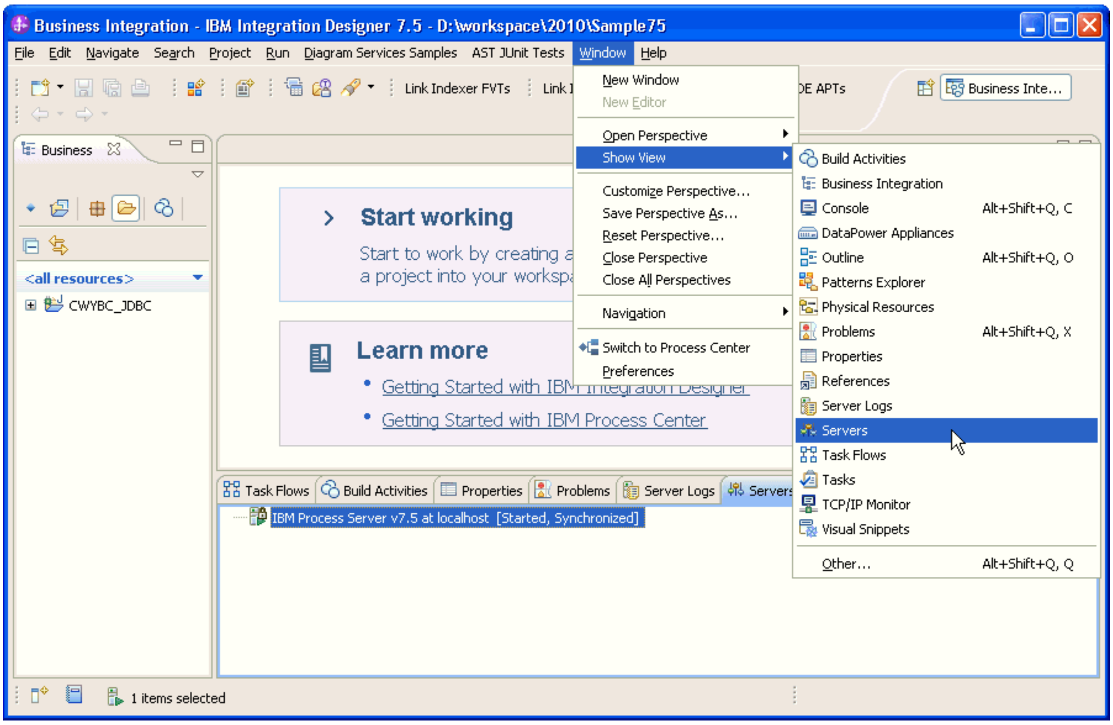
```
INSERT INTO CUSTOMER (PKEY, FNAME, LNAME, CCODE)
VALUES ('C1', 'JONE', 'TIGER', '1');
INSERT INTO CUSTOMER (PKEY, FNAME, LNAME, CCODE)
VALUES ('C2', 'ROTH', 'GREEN', '1');
INSERT INTO CUSTADD (ADDRID, CUSTID, CITY, ZIPCODE)
VALUES ('A1', 'C1', 'BEIJING', '100000');
INSERT INTO CUSTADD (ADDRID, CUSTID, CITY, ZIPCODE)
VALUES ('A2', 'C2', 'SHANGHAI', '200000');
```

Create an authentication alias

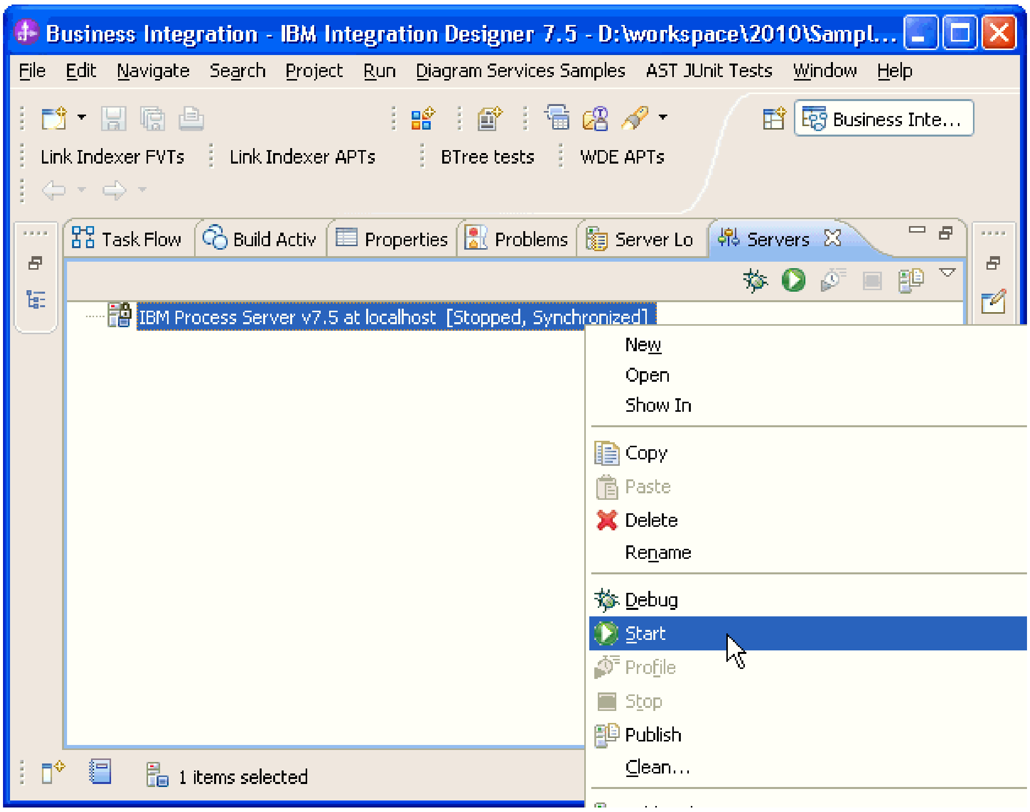
The authentication alias needs to be set because the data source created in the next section uses the username and password set in the authentication alias to connect to the database.

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

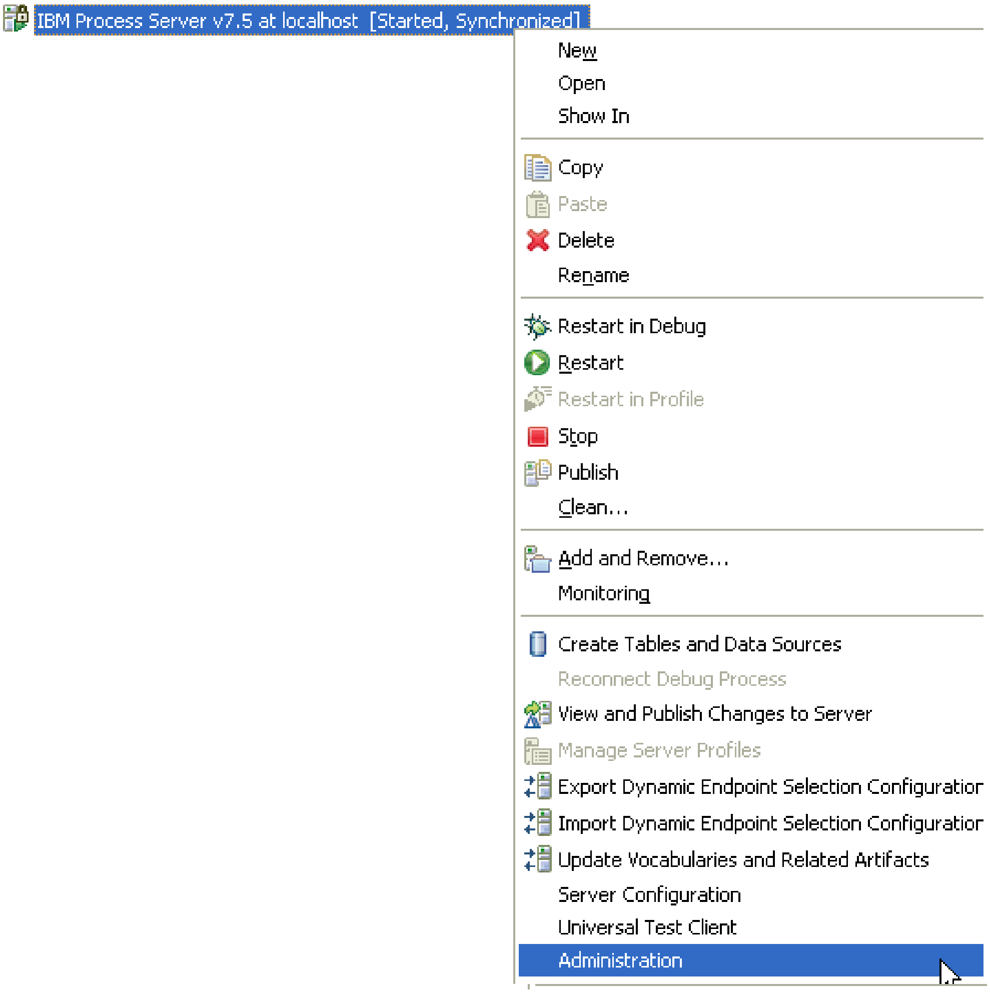
1. In IBM Integration Designer, switch to the **Servers** view by selecting **Windows > Show View > Servers**.



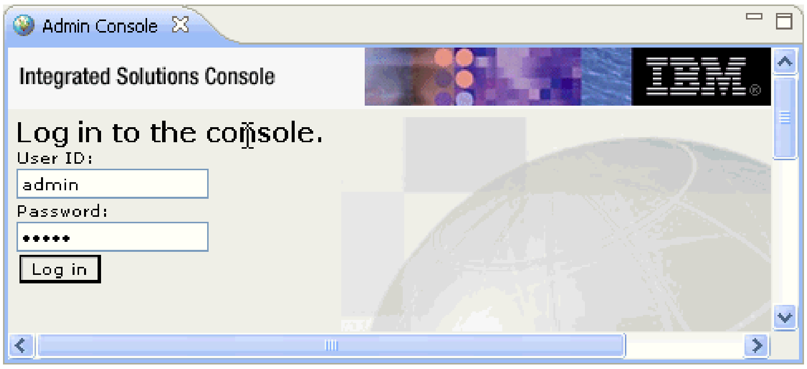
2. In the **Servers** view, right-click the server that you want to start and select **Start**.


Business Integration - IBM Integration Designer 7.5 - D:\workspace\2010\Sampl...
File Edit Navigate Search Project Run Diagram Services Samples AST JUnit Tests Window Help
Link Indexer FVTs Link Indexer APTs BTree tests WDE APTs
Business Inte...
Task Flow Build Activ Properties Problems Server Lo Servers
IBM Process Server v7.5 at localhost [Stopped, Synchronized]
New
Open
Show In
Copy
Paste
Delete
Rename
Debug
Start
Profile
Stop
Publish
Clean...
1 items selected

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


IBM Process Server v7.5 at localhost [Started, Synchronized]
New
Open
Show In
Copy
Paste
Delete
Rename
Restart in Debug
Restart
Restart in Profile
Stop
Publish
Clean...
Add and Remove...
Monitoring
Create Tables and Data Sources
Reconnect Debug Process
View and Publish Changes to Server
Manage Server Profiles
Export Dynamic Endpoint Selection Configuration
Import Dynamic Endpoint Selection Configuration
Update Vocabularies and Related Artifacts
Server Configuration
Universal Test Client
Administration

4. Log on to the administrative console.

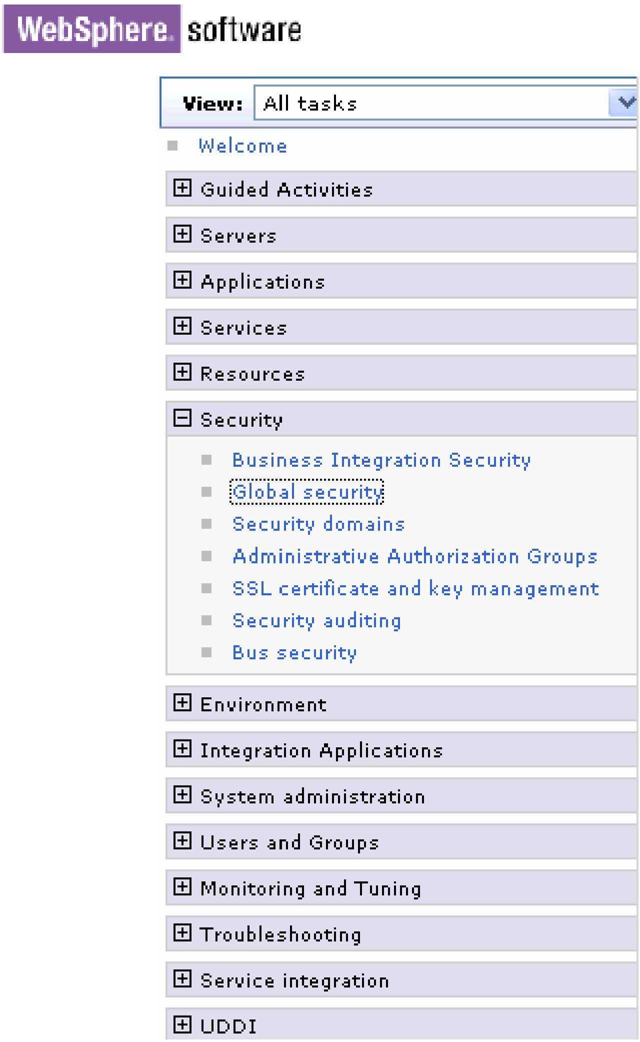


Log in to the console.

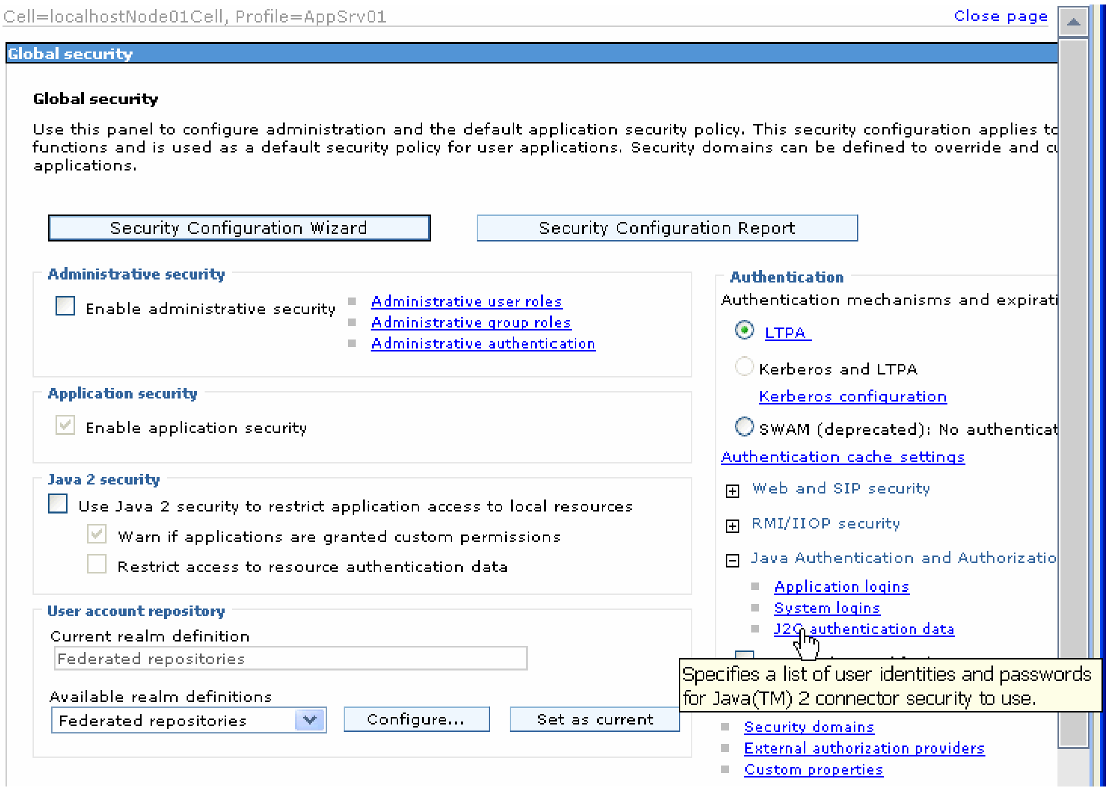
User ID: admin

Password: •••••

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



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


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 cu 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

Authentication

Authentication mechanisms and expirati

● LTPA

○ Kerberos and LTPA

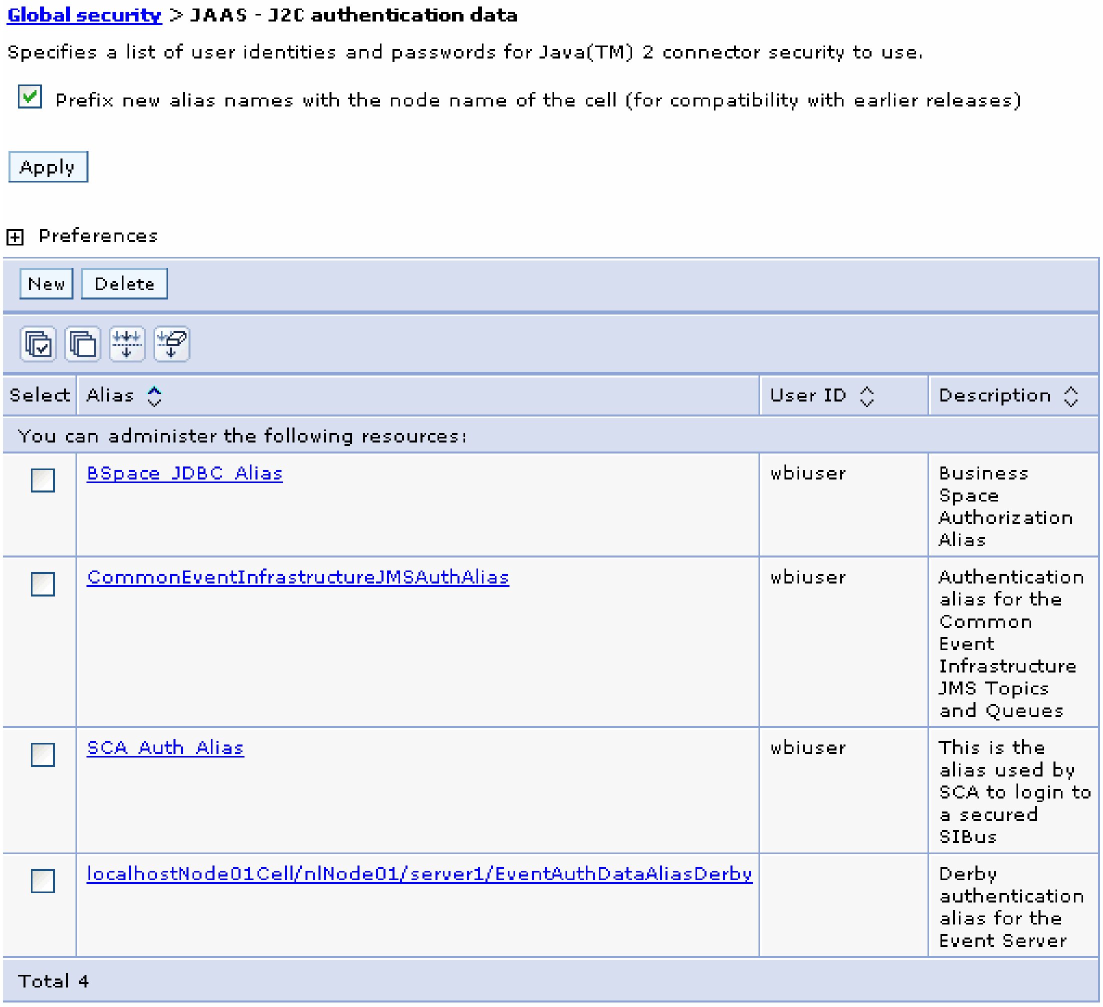
Kerberos configuration

○ SWAM (deprecated): No authenticat

Authentication cache settings

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

A list of existing aliases is displayed.



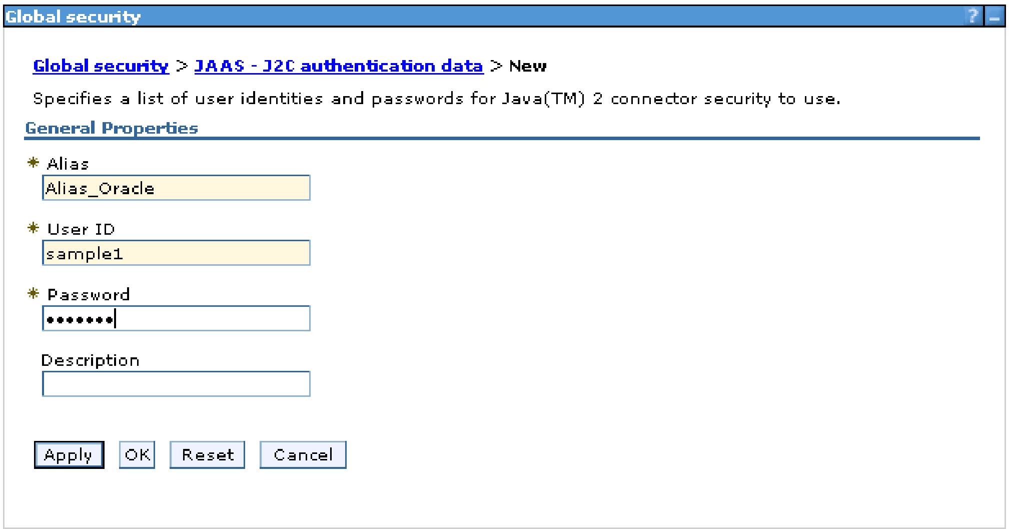
Global security > JAAS - J2C authentication data

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)

Select	Alias	User ID	Description
You can administer the following resources:			
□	BSpace_JDBC_Alias	wbiuser	Business Space Authorization Alias
□	CommonEventInfrastructureJMSAuthAlias	wbiuser	Authentication alias for the Common Event Infrastructure JMS Topics and Queues
□	SCA_Auth_Alias	wbiuser	This is the alias used by SCA to login to a secured SIBus
□	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**.



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

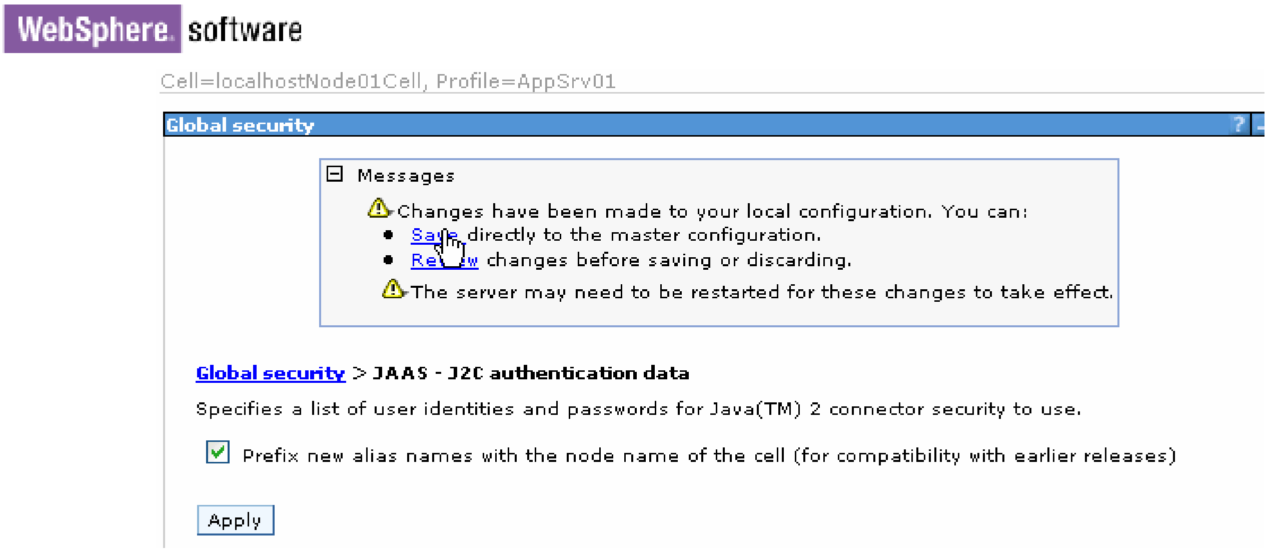
* Alias
Alias_Oracle

* User ID
sample1

* Password
•••••••

Description

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.

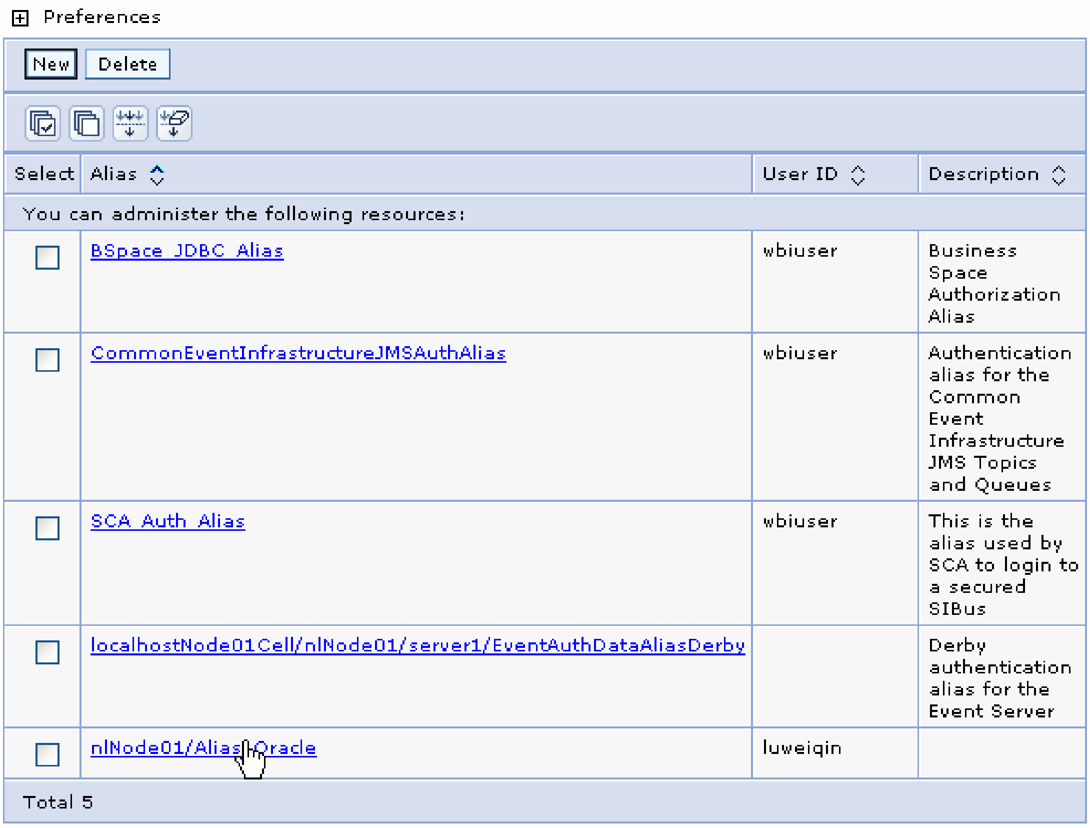
⚠ The server may need to be restarted for these changes to take effect.

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)

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



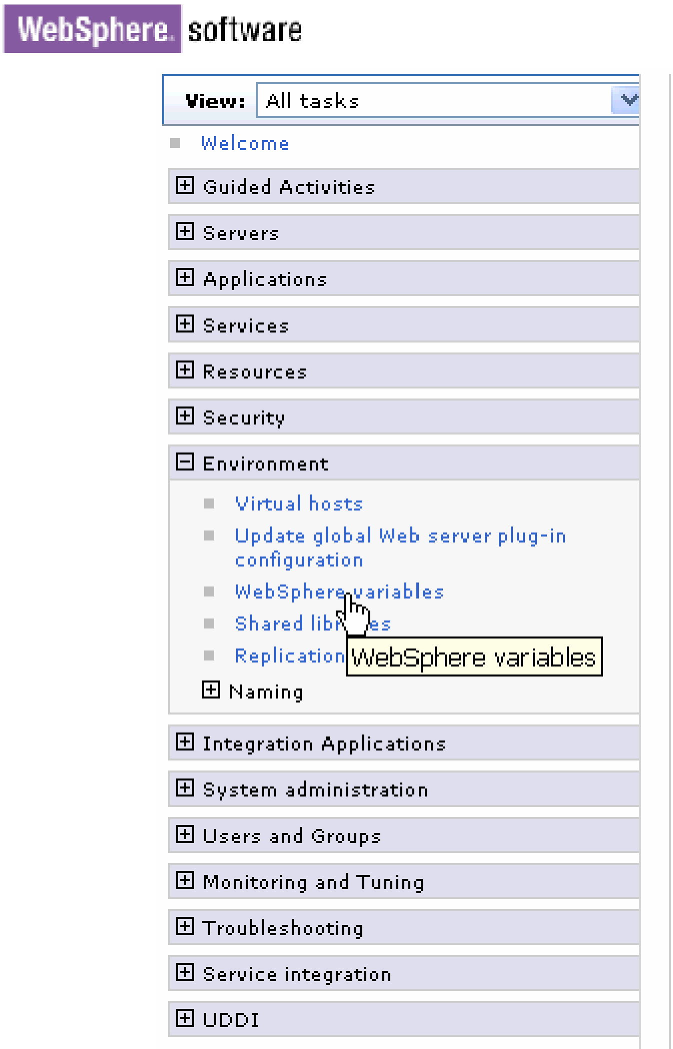
Select	Alias	User ID	Description
You can administer the following resources:			
□	BSpace_JDBC_Alias	wbiuser	Business Space Authorization Alias
□	CommonEventInfrastructureJMSAuthAlias	wbiuser	Authentication alias for the Common Event Infrastructure JMS Topics and Queues
□	SCA_Auth_Alias	wbiuser	This is the alias used by SCA to login to a secured SIBus
□	localhostNode01Cell/nlNode01/server1/EventAuthDataAliasDerby		Derby authentication alias for the Event Server
□	nlNode01/AliasOracle	luweiqin	
Total 5			

Create a data source

Create a data source in IBM Process Server, which the adapter will use to connect to the database. Here are the steps to create the data source in the IBM 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.

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

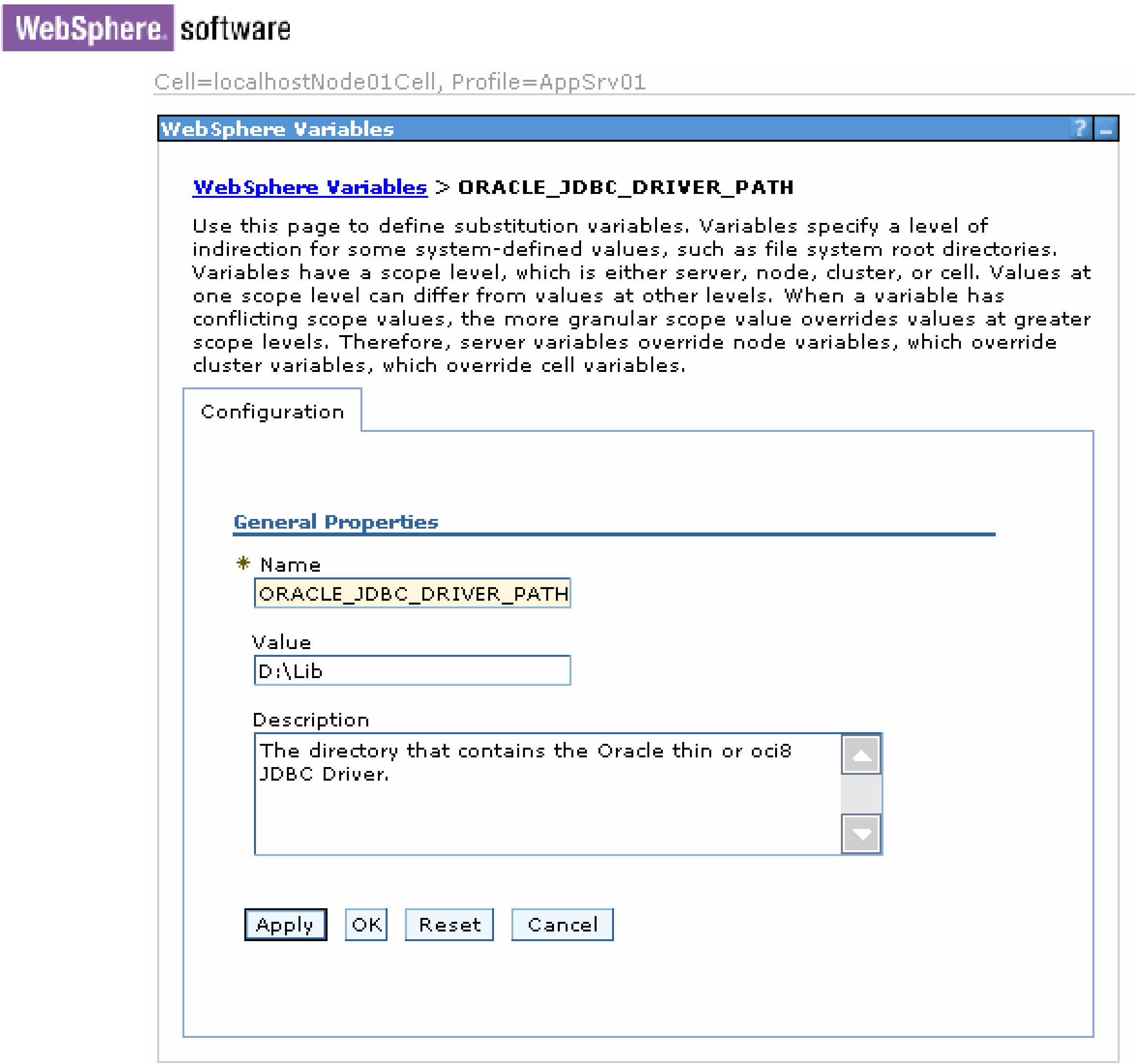


WebSphere software

View: All tasks

- Welcome
- ⊞ Guided Activities
- ⊞ Servers
- ⊞ Applications
- ⊞ Services
- ⊞ Resources
- ⊞ Security
- ⊟ Environment
  - Virtual hosts
  - Update global Web server plug-in configuration
  - WebSphere variables
  - Shared libraries
  - Replication
  - ⊞ Naming
- ⊞ Integration Applications
- ⊞ System administration
- ⊞ Users and Groups
- ⊞ Monitoring and Tuning
- ⊞ Troubleshooting
- ⊞ Service integration
- ⊞ UDDI

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

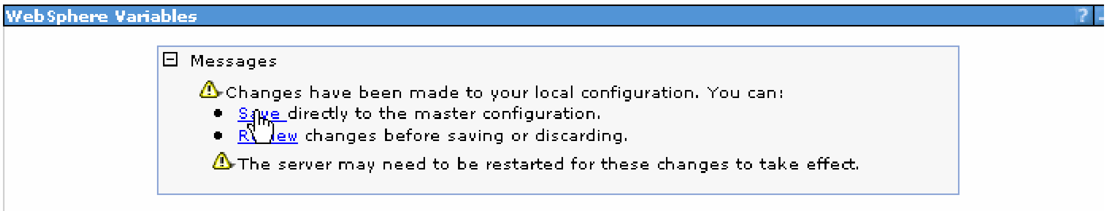
General Properties

* Name
ORACLE_JDBC_DRIVER_PATH

Value
D:\Lib

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

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



WebSphere Variables

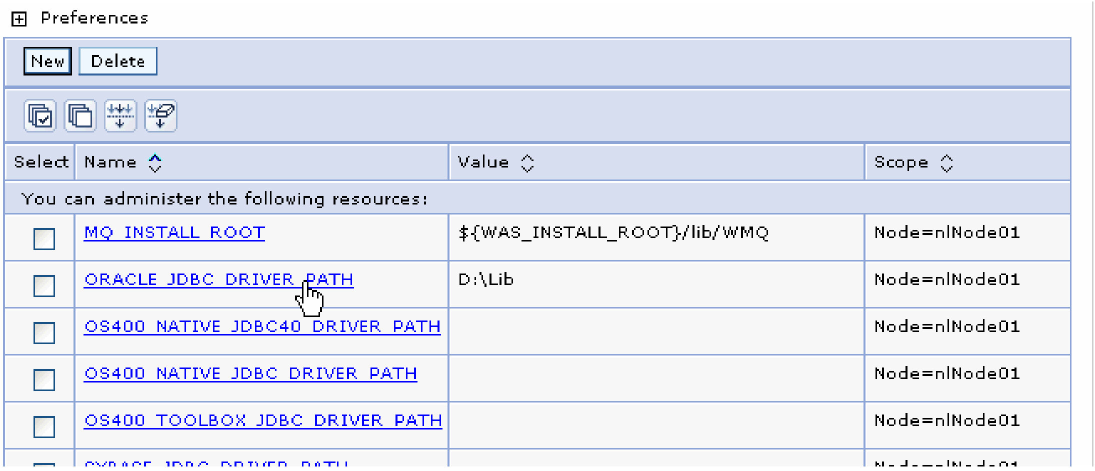
⊟ Messages

⚠ 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.

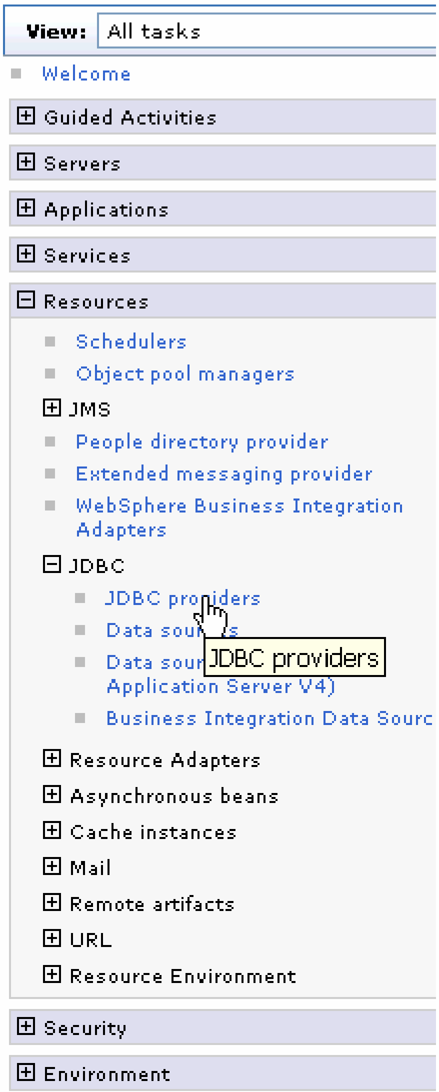
The variable is added and appears in the list.



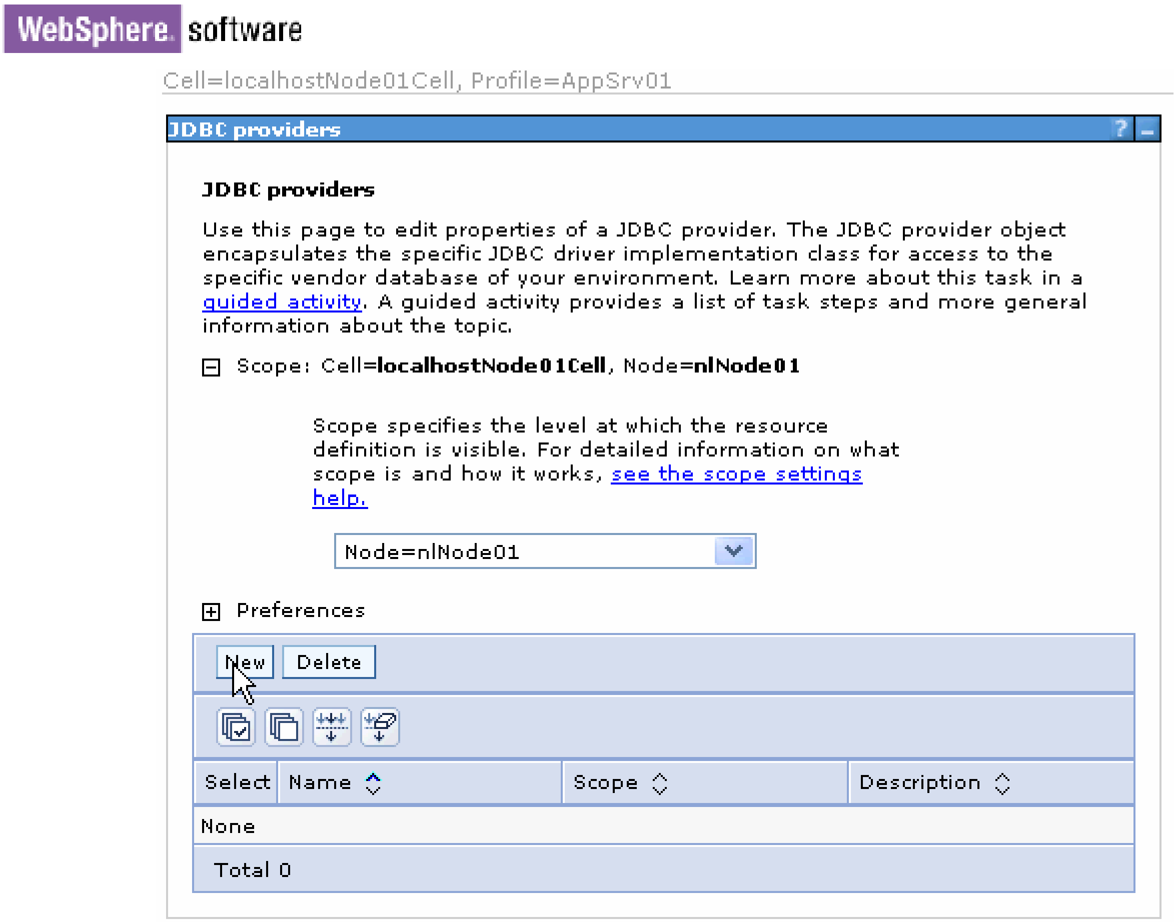
⊞ Preferences

Select	Name ◇	Value ◇	Scope ◇
You can administer the following resources:			
□	MQ_INSTALL_ROOT	${WAS_INSTALL_ROOT}/lib/WMQ	Node=nlNode01
□	ORACLE_JDBC_DRIVER_PATH	D:\Lib	Node=nlNode01
□	OS400_NATIVE_JDBC40_DRIVER_PATH		Node=nlNode01
□	OS400_NATIVE_JDBC_DRIVER_PATH		Node=nlNode01
□	OS400_TOOLBOX_JDBC_DRIVER_PATH		Node=nlNode01
□	SYBASE_JDBC_DRIVER_PATH		Node=nlNode01

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



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=**nlNode01**

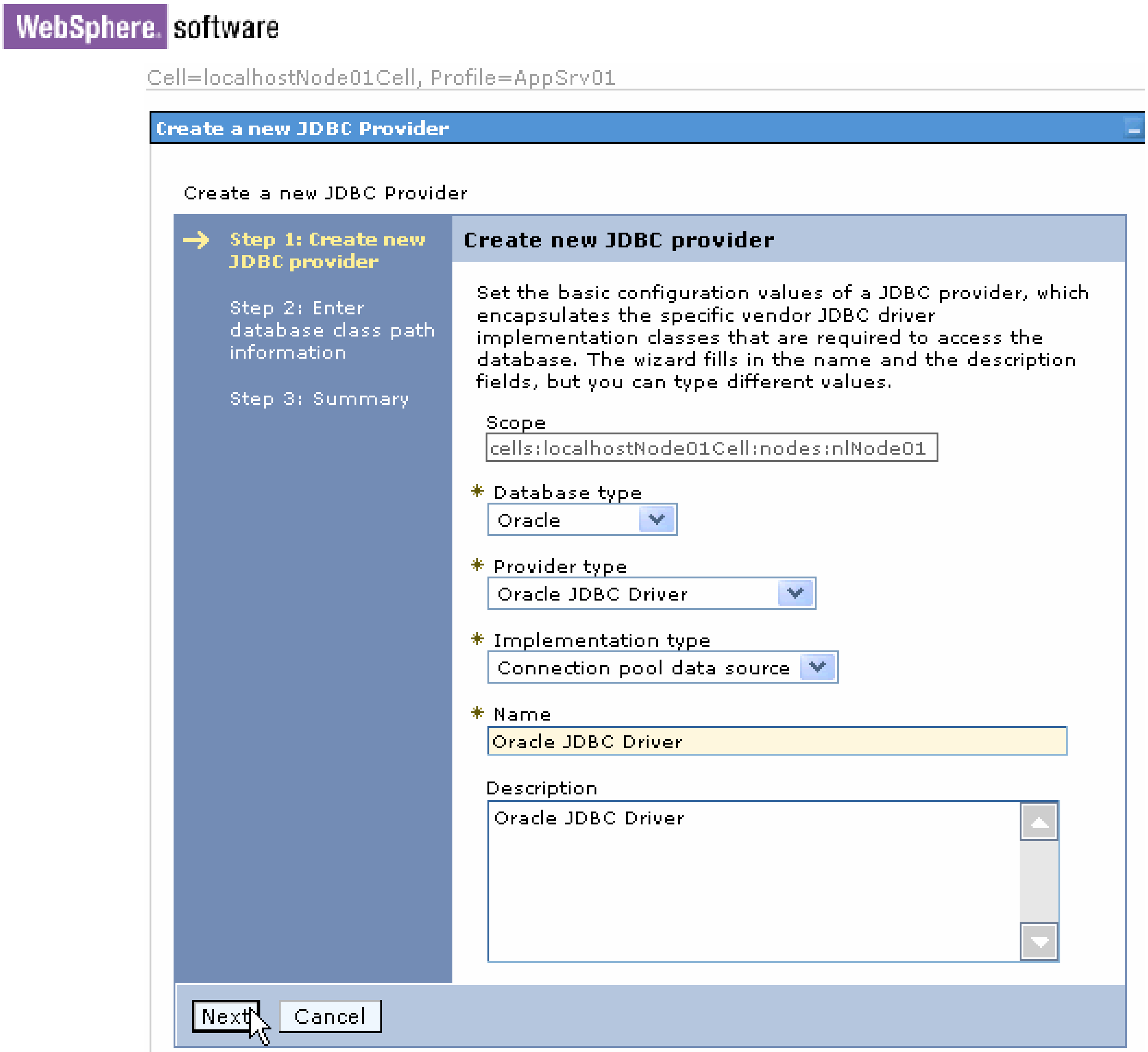
Scope specifies the level at which the resource definition is visible. For detailed information on what scope is and how it works, see the scope settings help.

Node=nlNode01

⊞ 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**.



Cell=localhostNode01Cell, Profile=AppSrv01

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:nlNode01

* Database type
Oracle

* Provider type
Oracle JDBC Driver

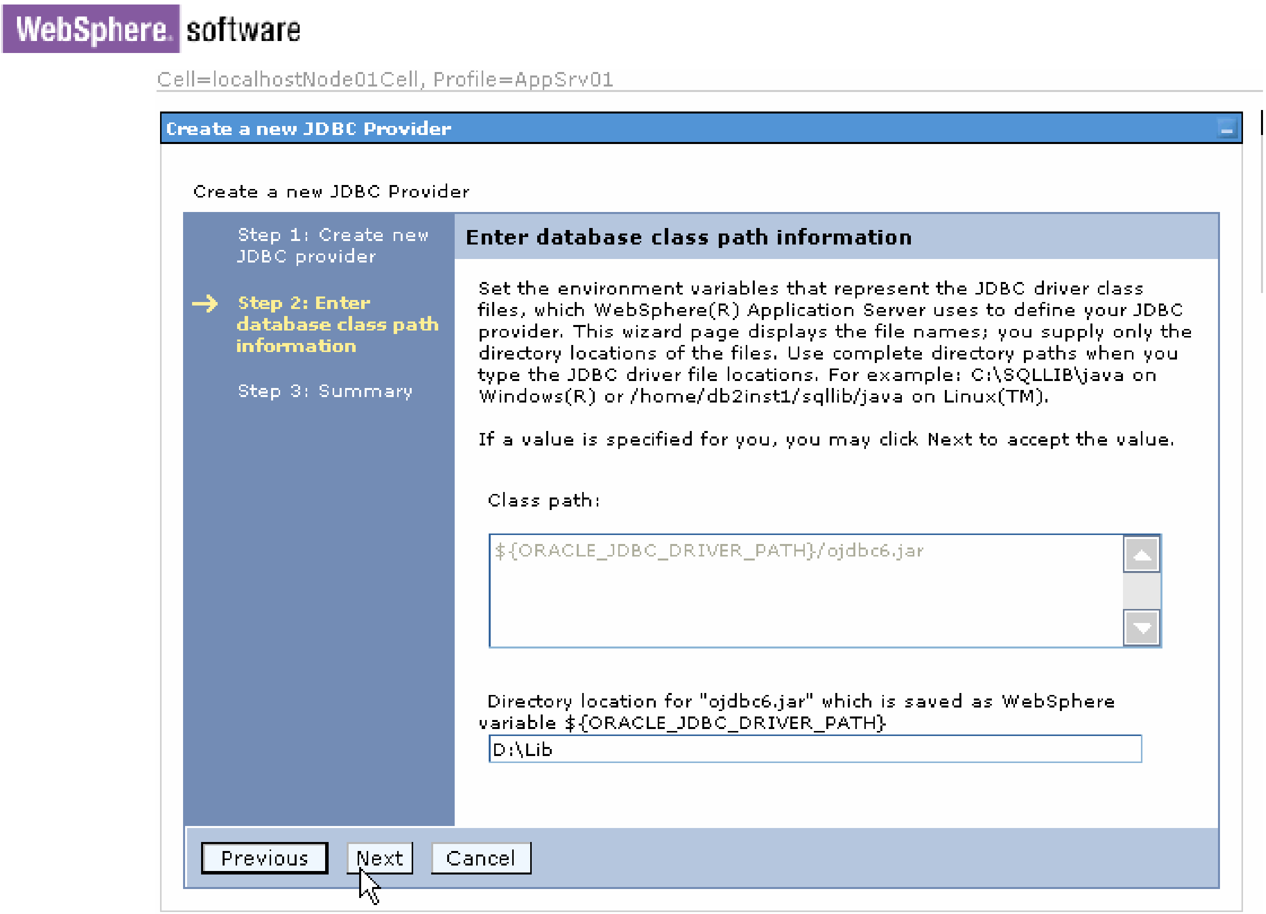
* Implementation type
Connection pool data source

* Name
Oracle JDBC Driver

Description
Oracle JDBC Driver

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**.



Cell=localhostNode01Cell, Profile=AppSrv01

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: 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.

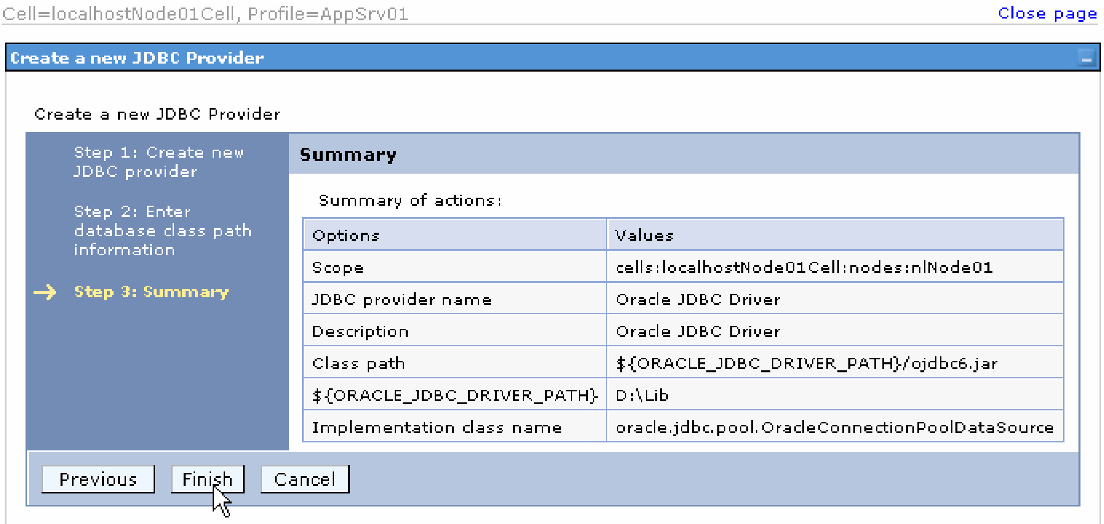
Class path:

${ORACLE_JDBC_DRIVER_PATH}/ojdbc6.jar

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

D:\Lib

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



Cell=localhostNode01Cell, Profile=AppSrv01

Create a new JDBC Provider

Create a new JDBC Provider

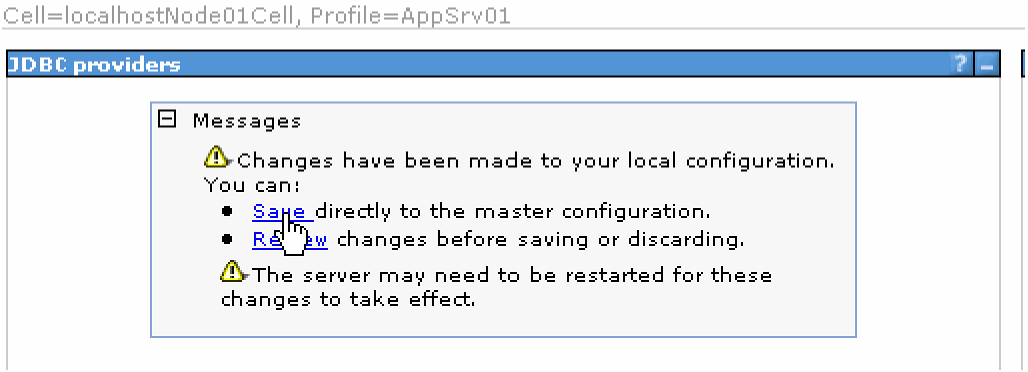
Step 1: Create new JDBC provider
Step 2: Enter database class path information
→ Step 3: Summary

Summary

Summary of actions:

Options	Values
Scope	cells:localhostNode01Cell:nodes:nlNode01
JDBC provider name	Oracle JDBC Driver
Description	Oracle JDBC Driver
Class path	${ORACLE_JDBC_DRIVER_PATH}/ojdbc6.jar
${ORACLE_JDBC_DRIVER_PATH}	D:\Lib
Implementation class name	oracle.jdbc.pool.OracleConnectionPoolDataSource

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



Cell=localhostNode01Cell, Profile=AppSrv01

JDBC providers

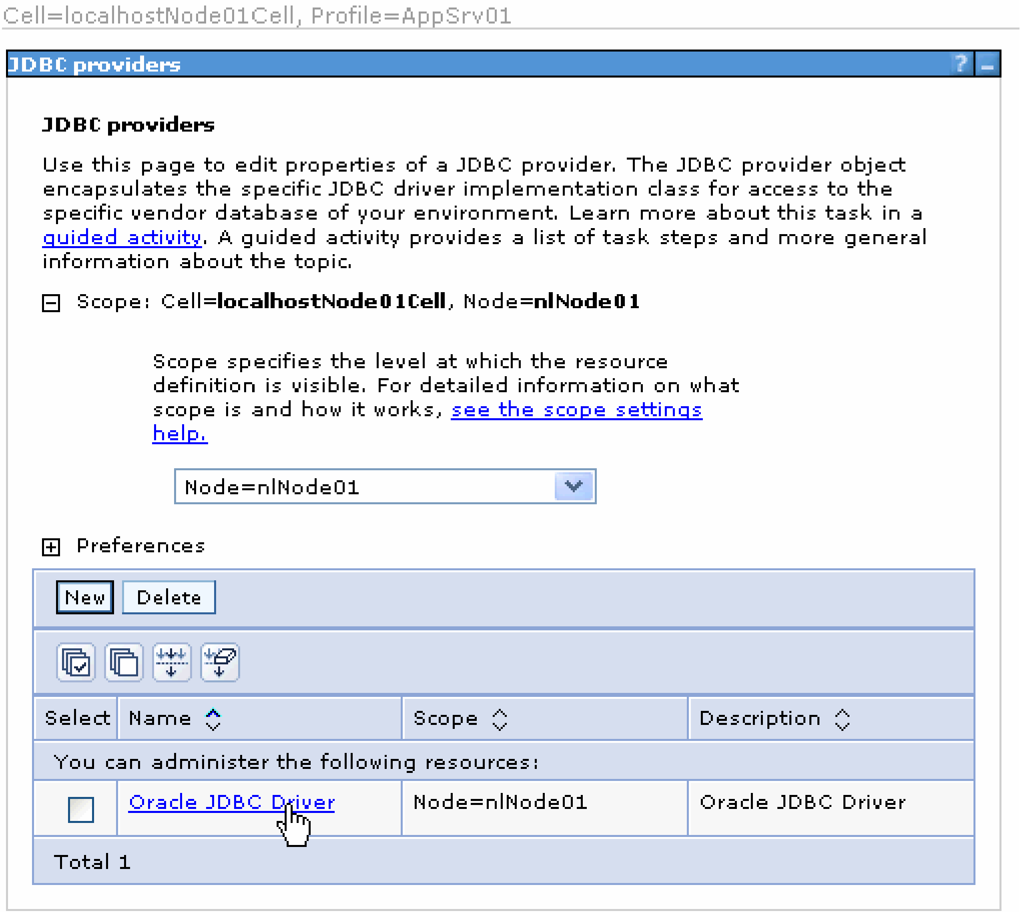
⊟ Messages

⚠ 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

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=**nlNode01**

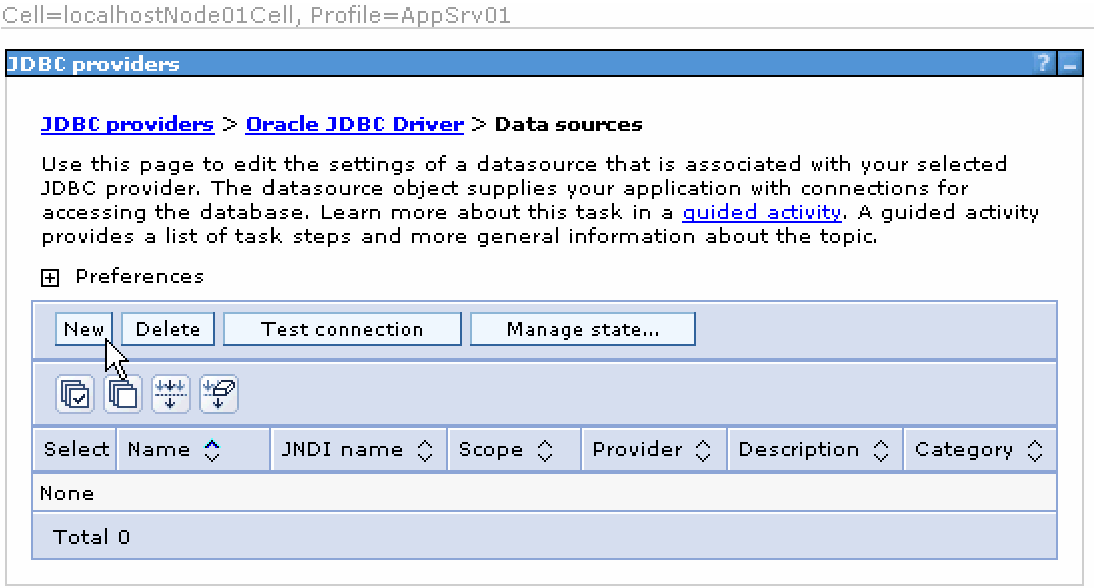
Scope specifies the level at which the resource definition is visible. For detailed information on what scope is and how it works, see the scope settings help.

Node=nlNode01

⊞ Preferences

Select	Name	Scope	Description
You can administer the following resources:			
□	Oracle JDBC Driver	Node=nlNode01	Oracle JDBC Driver
Total 1			

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



Cell=localhostNode01Cell, Profile=AppSrv01

JDBC providers

JDBC providers > Oracle JDBC Driver > Data sources

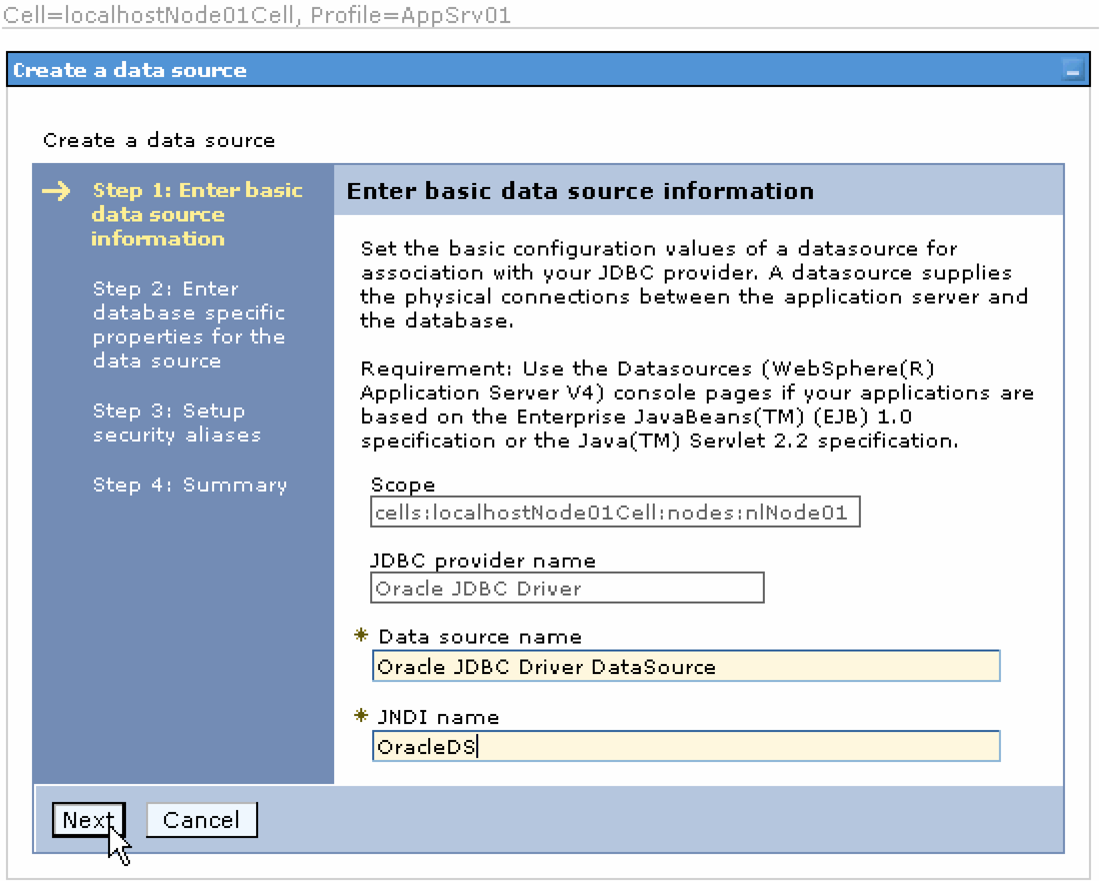
Use this page to edit the settings of a datasource that is associated with your selected JDBC provider. The datasource object supplies your application with connections for accessing the database. Learn more about this task in a guided activity. A guided activity provides a list of task steps and more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

⊞ Preferences

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.

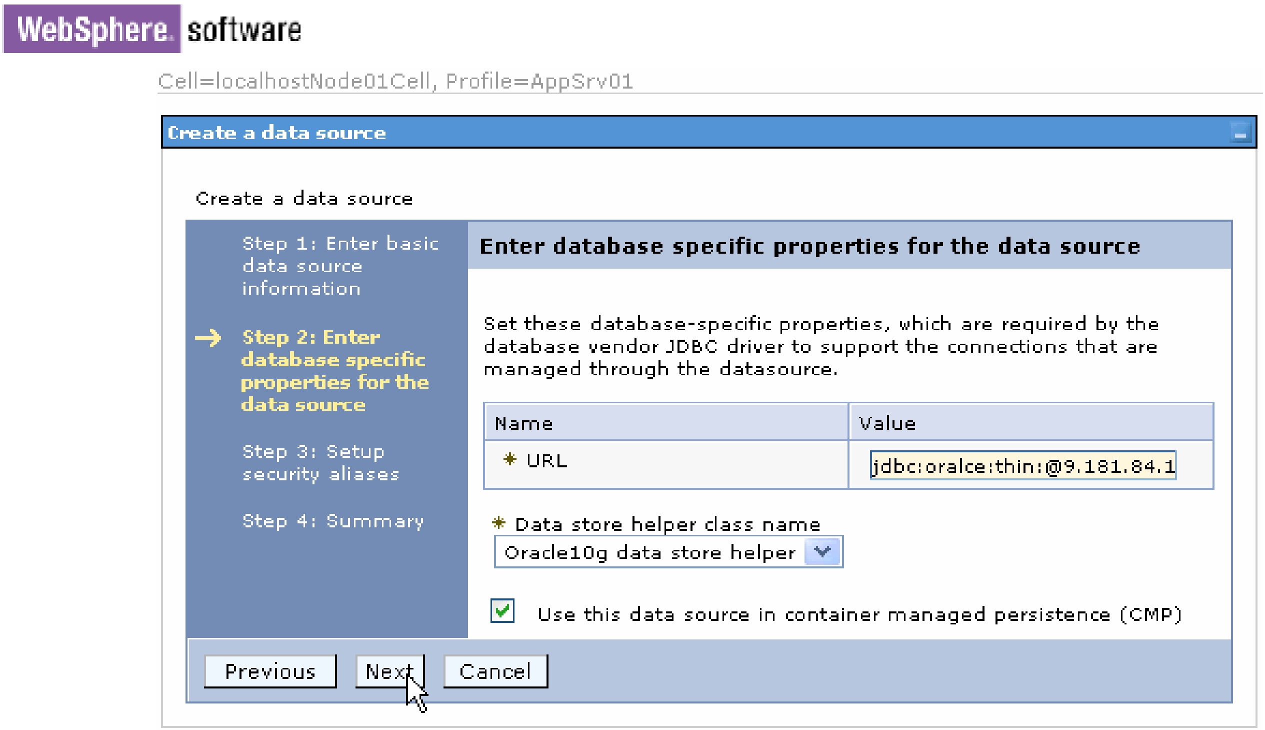
Scope
cells:localhostNode01Cell:nodes:nlNode01

JDBC provider name
Oracle JDBC Driver

* Data source name
Oracle JDBC Driver DataSource

* JNDI name
OracleDS

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

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

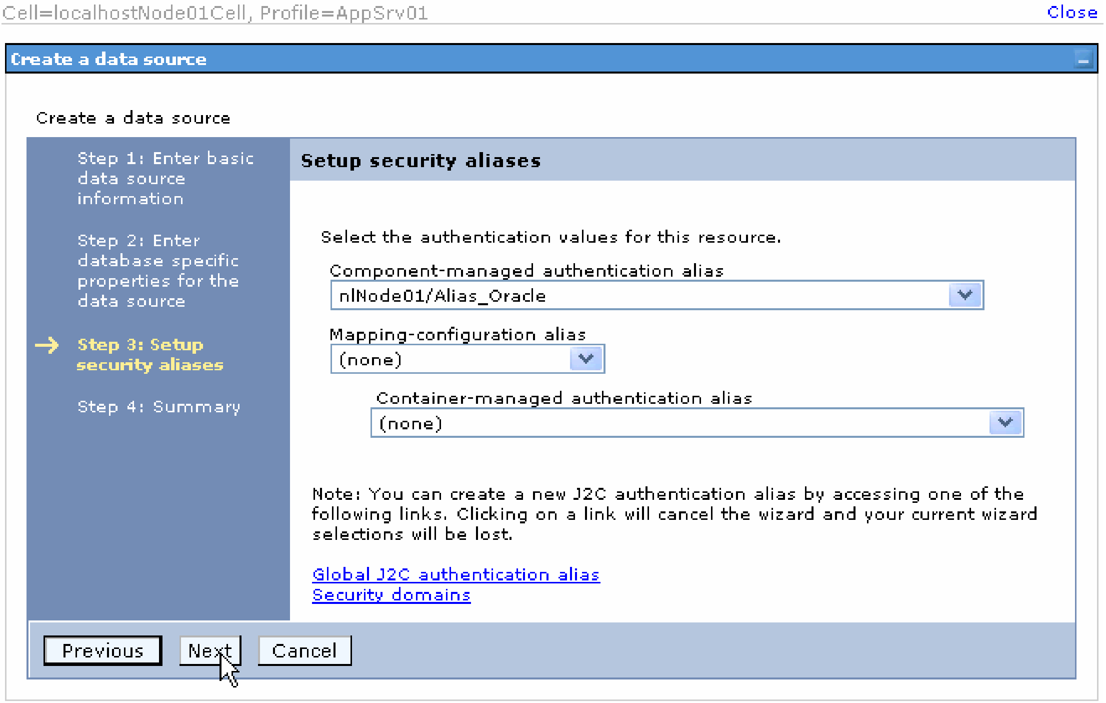
Set these database-specific properties, which are required by the database vendor JDBC driver to support the connections that are managed through the datasource.

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

* Data store helper class name: Oracle10g data store helper

☑ Use this data source in container managed persistence (CMP)

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

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

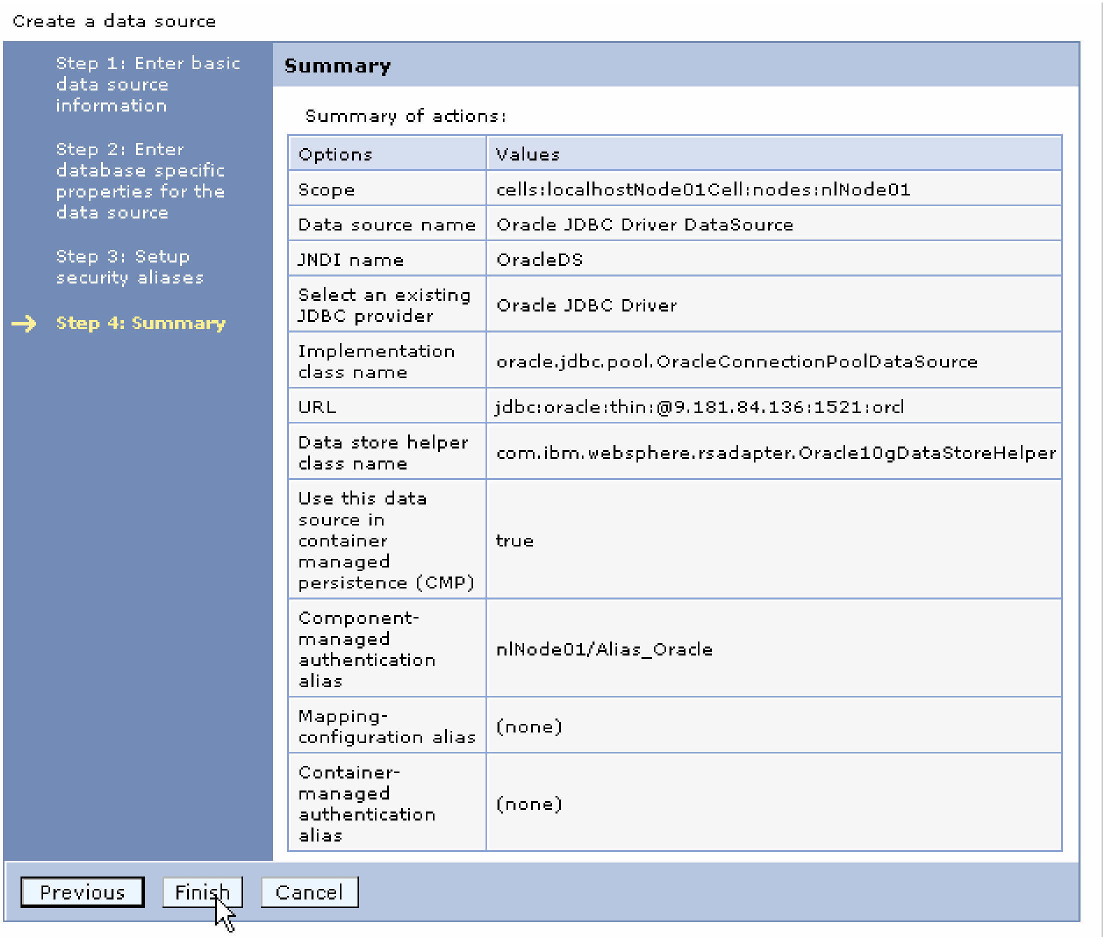
Mapping-configuration alias: (none)

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

15. 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

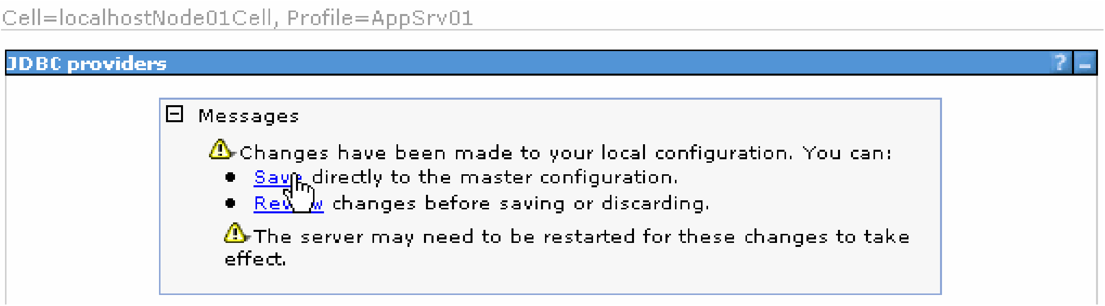
→ Step 4: Summary

Summary

Summary of actions:

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:orcl
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/Alias_Oracle
Mapping-configuration alias	(none)
Container-managed authentication alias	(none)

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



Cell=localhostNode01Cell, Profile=AppSrv01

JDBC providers

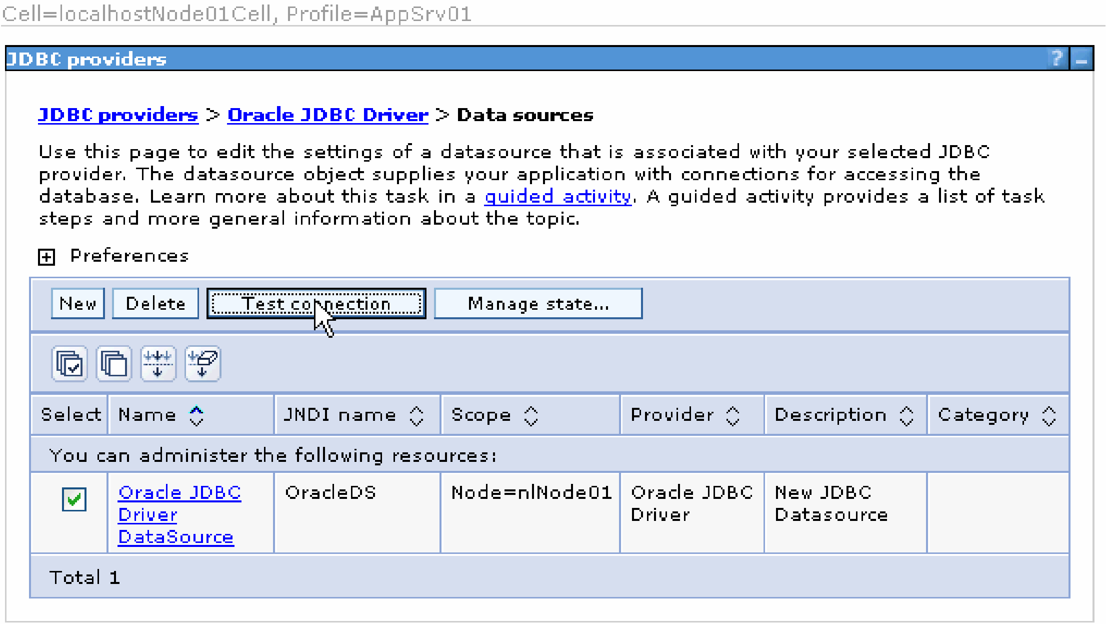
⊟ Messages

⚠ 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.

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



Cell=localhostNode01Cell, Profile=AppSrv01

JDBC providers

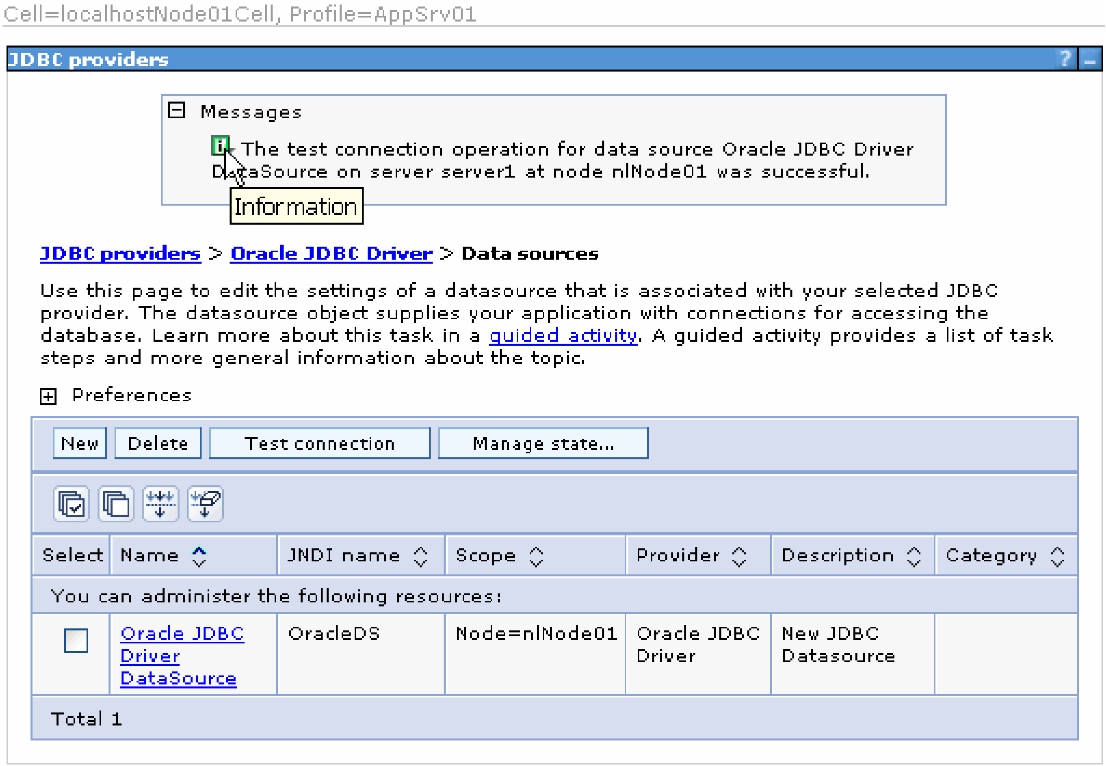
JDBC providers > Oracle JDBC Driver > Data sources

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

Select	Name	JNDI name	Scope	Provider	Description	Category
You can administer the following resources:						
☑	Oracle JDBC Driver DataSource	OracleDS	Node=nlNode01	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

⊟ Messages

The test connection operation for data source Oracle JDBC Driver DataSource on server server1 at node nlNode01 was successful.

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

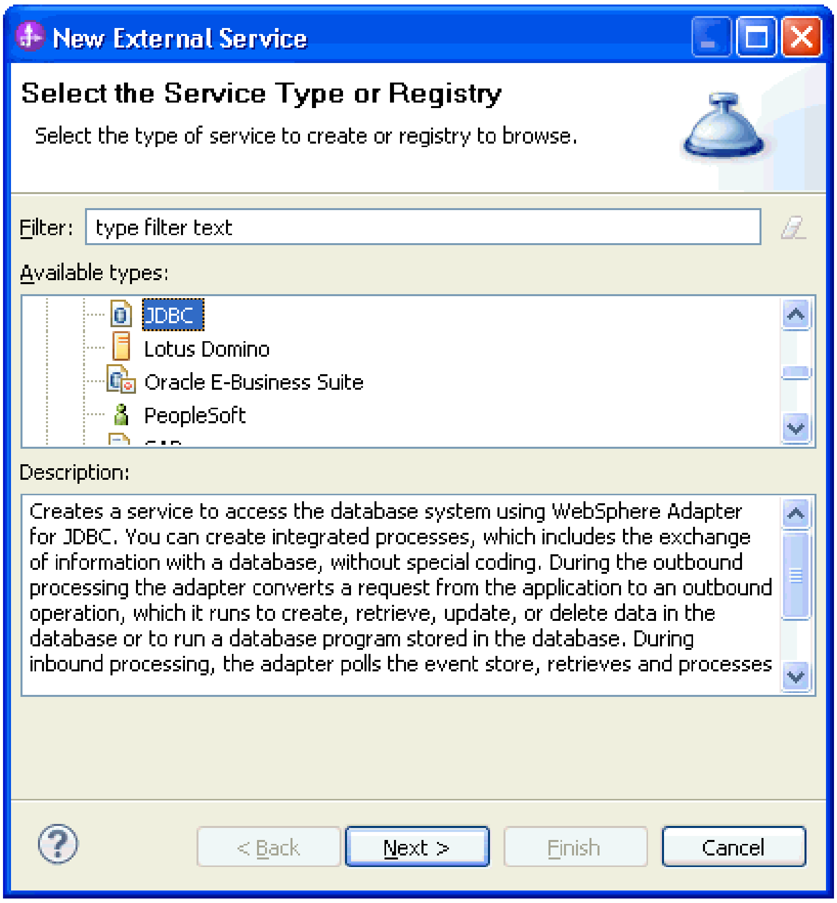
Select	Name	JNDI name	Scope	Provider	Description	Category
You can administer the following resources:						
□	Oracle JDBC Driver DataSource	OracleDS	Node=nlNode01	Oracle JDBC Driver	New JDBC Datasource	
Total 1						

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

Configure the adapter for inbound processing

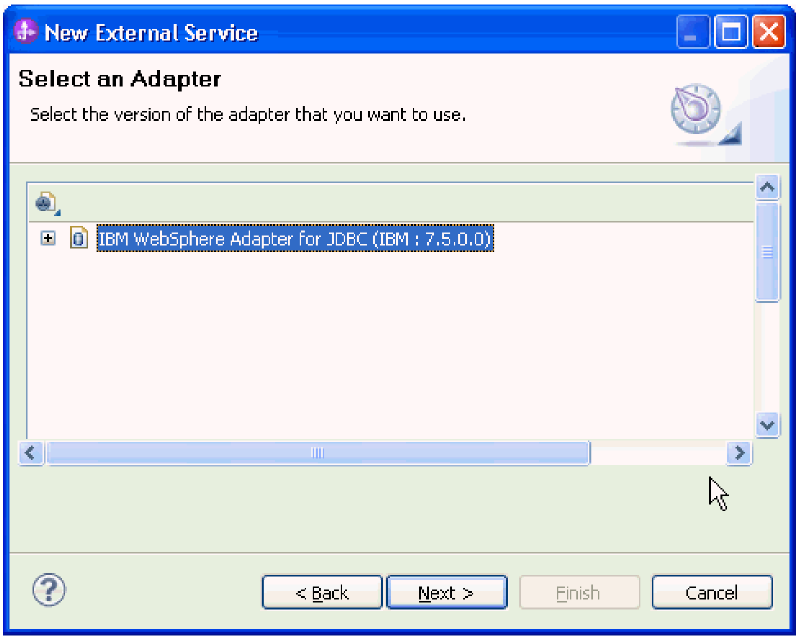
Run the external service wizard to specify business objects, services, and configuration details.

1. Switch to the Business Integration Perspective in IBM Integration Designer 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**.



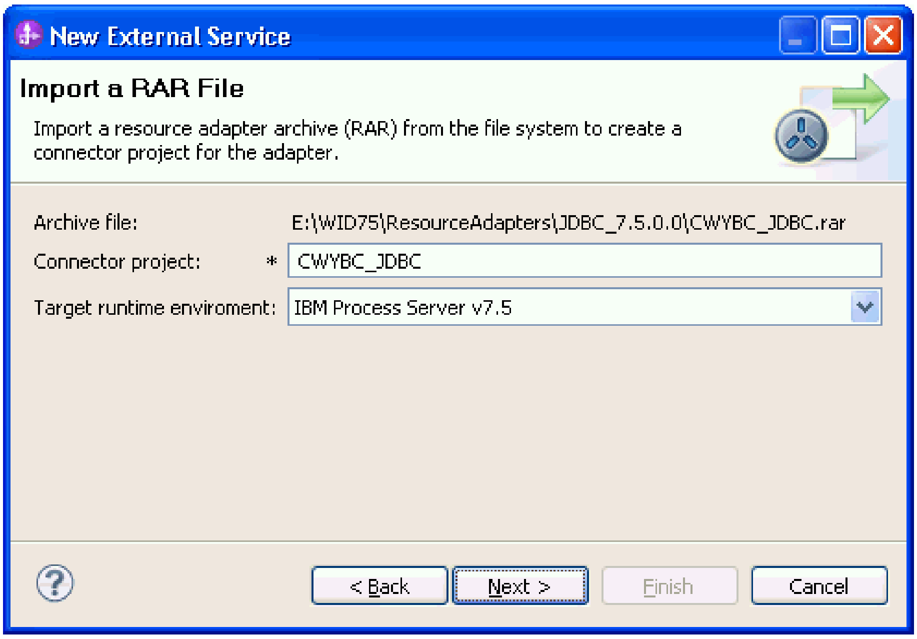
Creates a service to access the database system using WebSphere Adapter for JDBC. You can create integrated processes, which includes the exchange of information with a database, without special coding. During the outbound processing the adapter converts a request from the application to an outbound operation, which it runs to create, retrieve, update, or delete data in the database or to run a database program stored in the database. During inbound processing, the adapter polls the event store, retrieves and processes

4. Select the **IBM WebSphere Adapter for JDBC (IBM: 7.5.0.0)** and click **Next**.


New External Service
Select an Adapter
Select the version of the adapter that you want to use.
IBM WebSphere Adapter for JDBC (IBM : 7.5.0.0)
< Back
Next >
Finish
Cancel

5. In the **Connector project** field, enter **CWYBC_JDBC**.

6. In the **Target runtime environment** field, select the appropriate runtime and click **Next**.

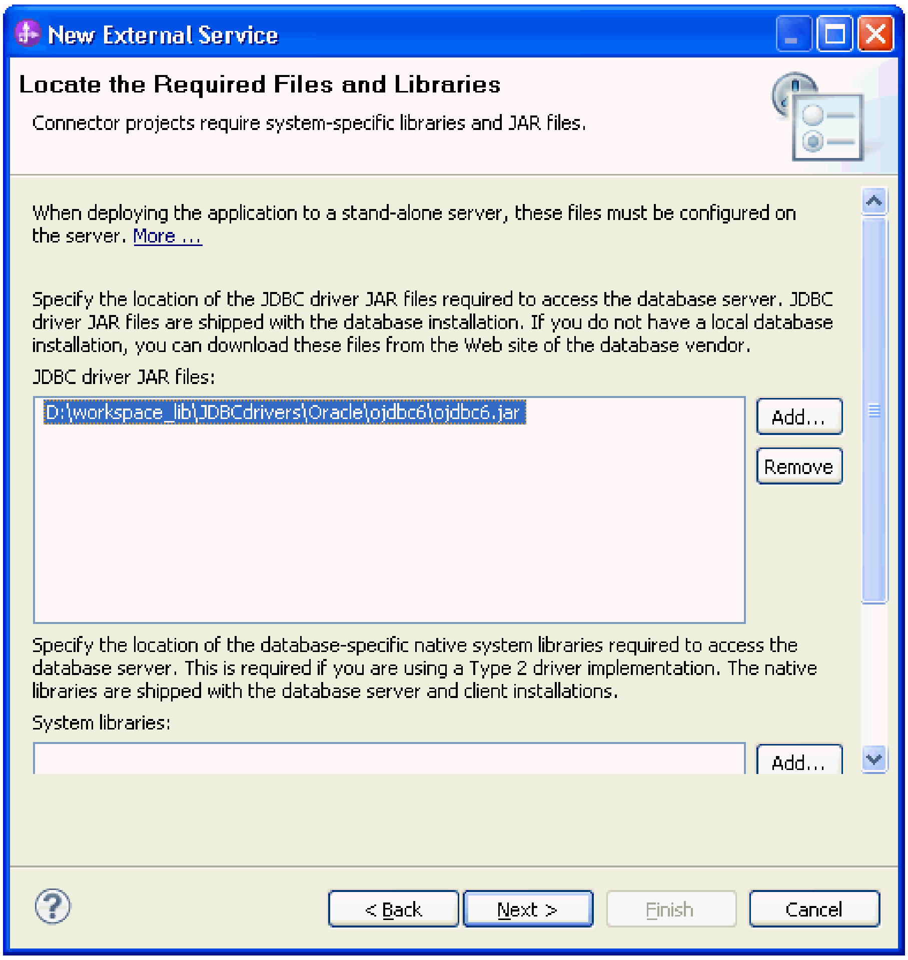

New External Service
Import a RAR File
Import a resource adapter archive (RAR) from the file system to create a connector project for the adapter.
< Back
Next >
Finish
Cancel

Archive file: E:\WID75\ResourceAdapters\JDBC_7.5.0.0\CWYBC_JDBC.rar

Connector project: * CWYBC_JDBC

Target runtime enviroment: IBM Process Server v7.5

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**.



New External Service

Locate the Required Files and Libraries

Connector projects require system-specific libraries and JAR files.

When deploying the application to a stand-alone server, these files must be configured on the server. More ...

Specify the location of the JDBC driver JAR files required to access the database server. JDBC driver JAR files are shipped with the database installation. If you do not have a local database installation, you can download these files from the Web site of the database vendor.

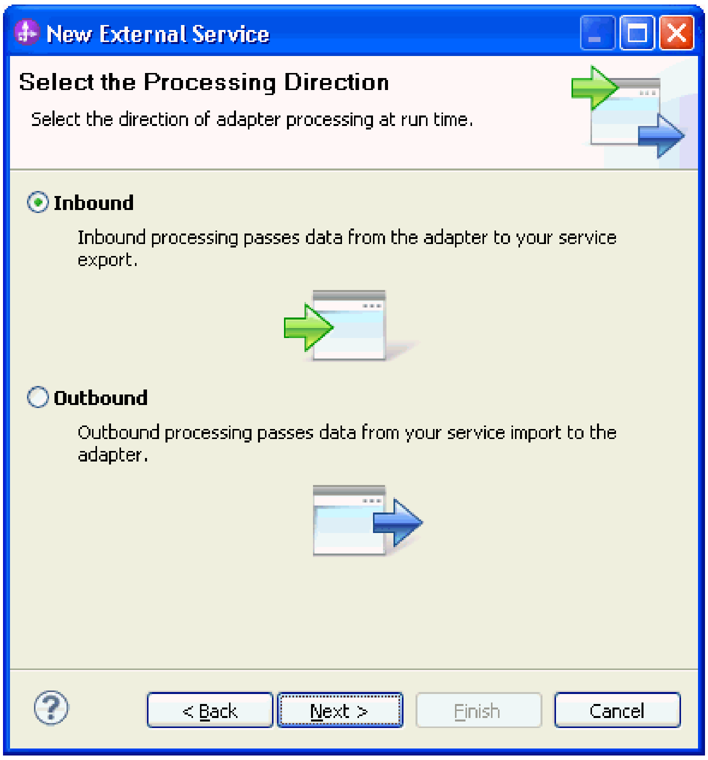
JDBC driver JAR files:

D:\workspace_lib\JDBCdrivers\Oracle\ojdbc6\ojdbc6.jar

Specify the location of the database-specific native system libraries required to access the database server. This is required if you are using a Type 2 driver implementation. The native libraries are shipped with the database server and client installations.

System libraries:

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



Select the Processing Direction

Select the direction of adapter processing at run time.

● **Inbound**

Inbound processing passes data from the adapter to your service export.

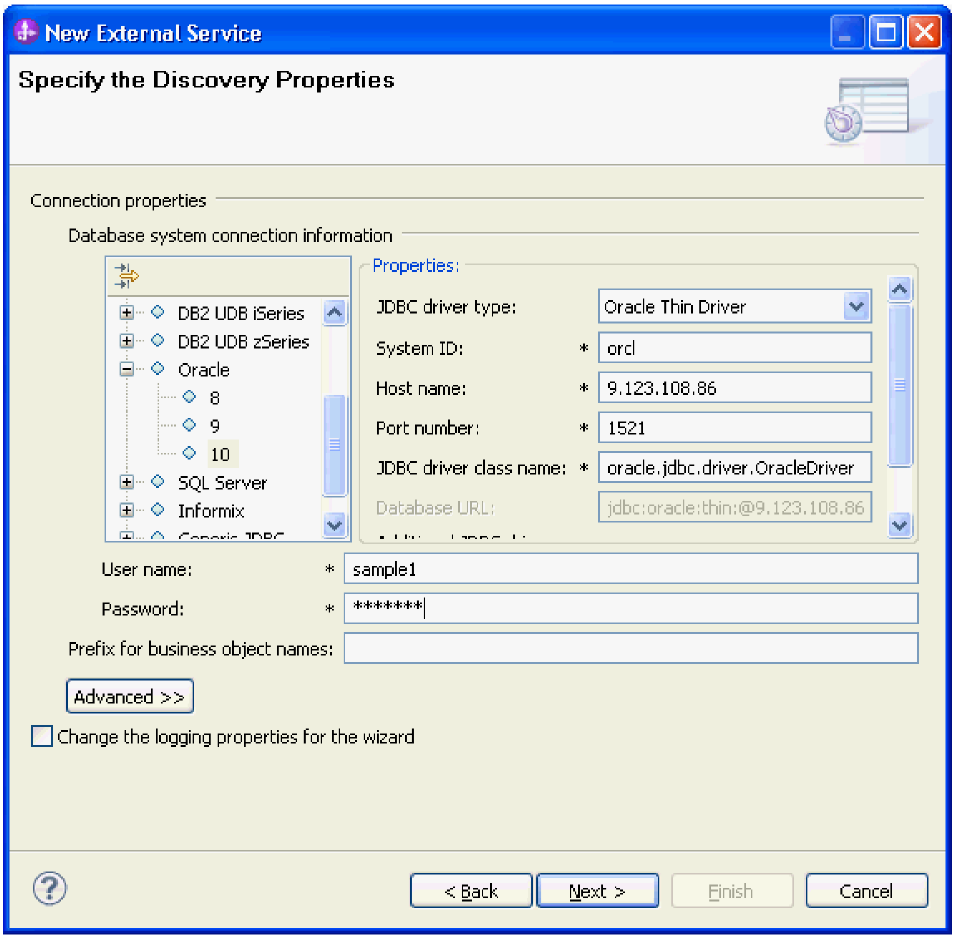
○ **Outbound**

Outbound processing passes data from your service import to the adapter.

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**.



New External Service

Specify the Discovery Properties

Connection properties

Database system connection information

Properties:

JDBC driver type: Oracle Thin Driver

System ID: * orcl

Host name: * 9.123.108.86

Port number: * 1521

JDBC driver class name: * oracle.jdbc.driver.OracleDriver

Database URL: jdbc:oracle:thin:@9.123.108.86

User name: * sample1

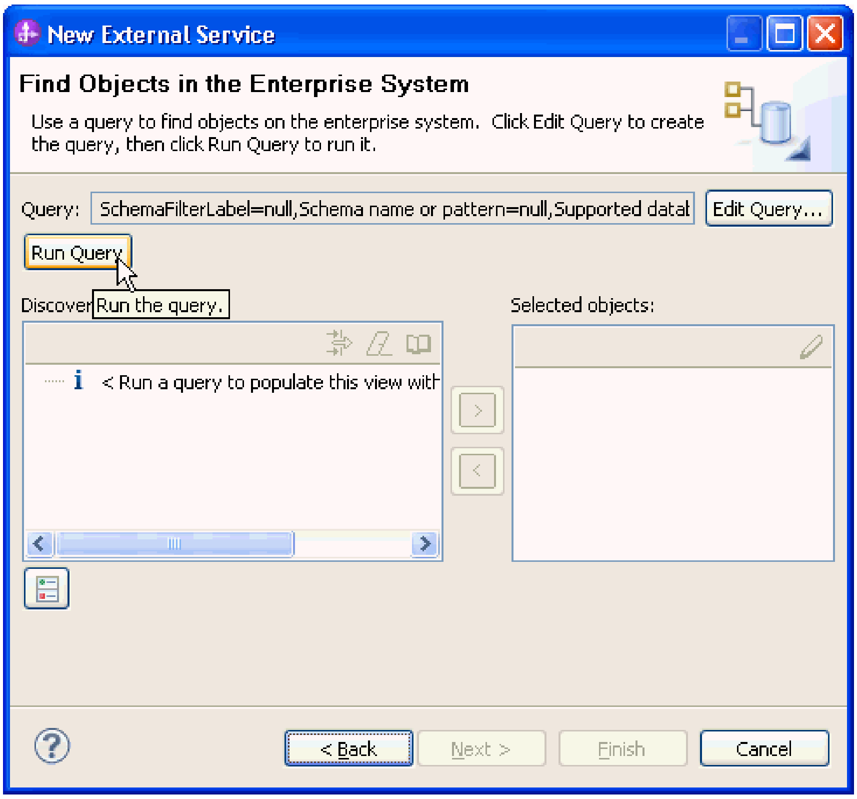
Password: * *******

Prefix for business object names:

□ Change the logging properties for the wizard

Select the business objects to be used with the adapter

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



New External Service

Find Objects in the Enterprise System

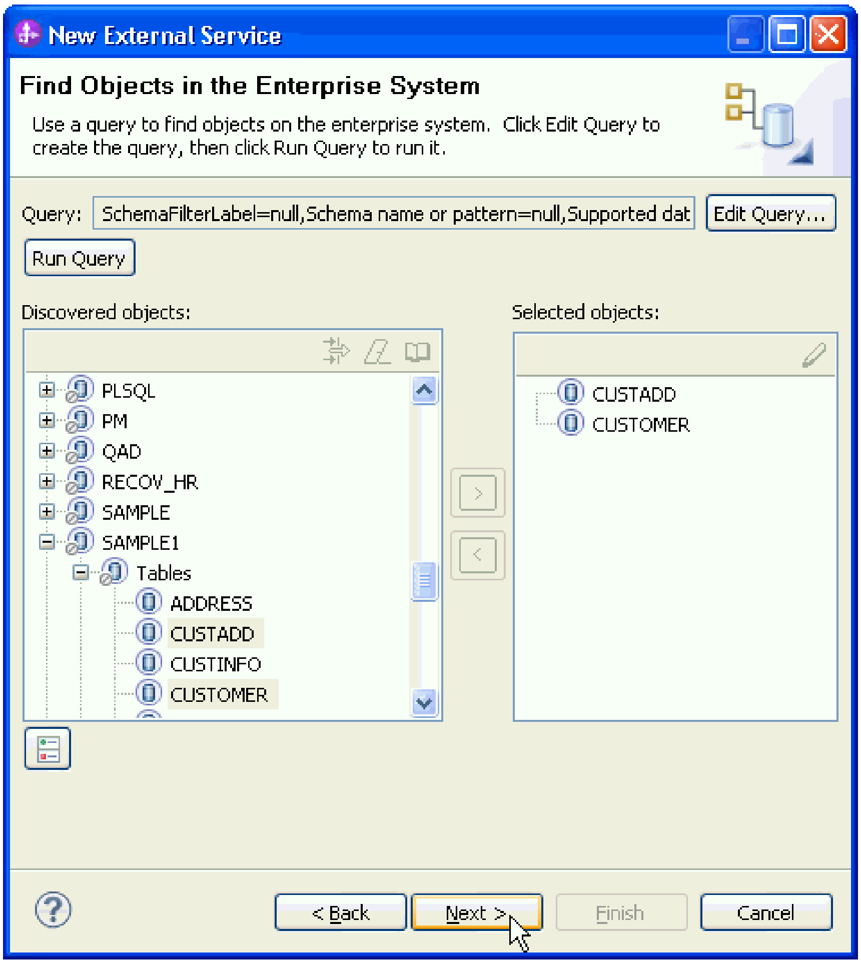
Use a query to find objects on the enterprise system. Click Edit Query to create the query, then click Run Query to run it.

Query: SchemaFilterLabel=null,Schema name or pattern=null,Supported datal

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**.



Find Objects in the Enterprise System

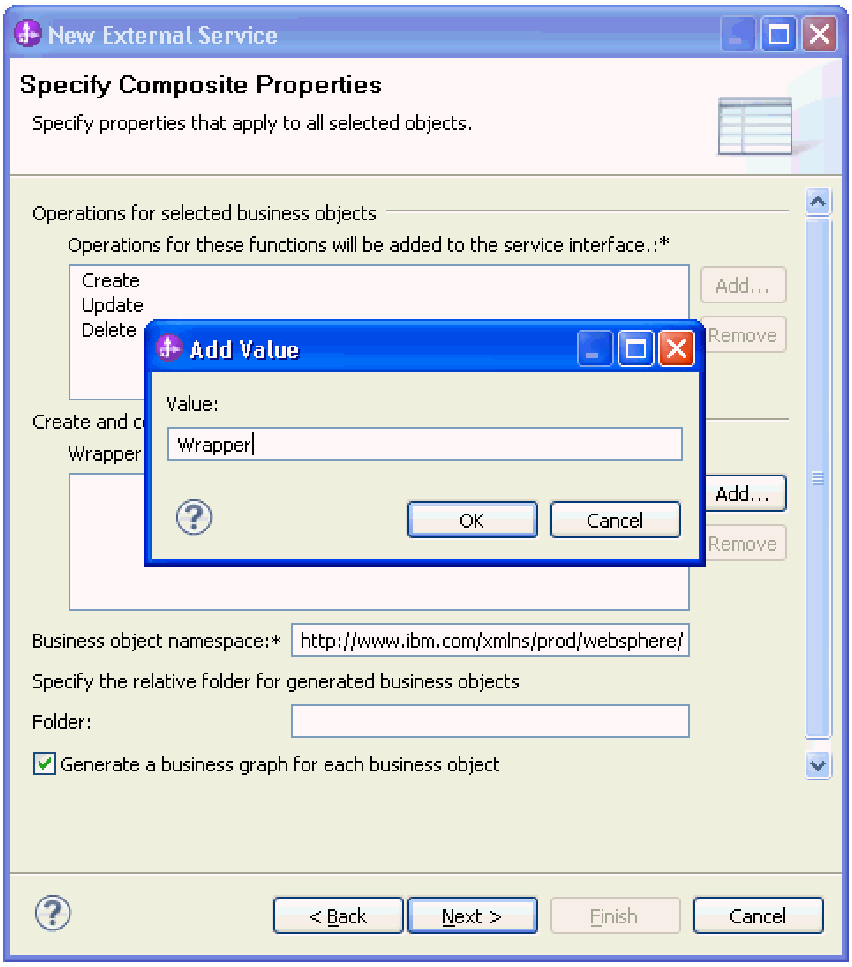
Use a query to find objects on the enterprise system. Click Edit Query to create the query, then click Run Query to run it.

Query: SchemaFilterLabel=null,Schema name or pattern=null,Supported dat

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**.



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

Create and c

Wrapper

Add Value

Value:

Wrapper

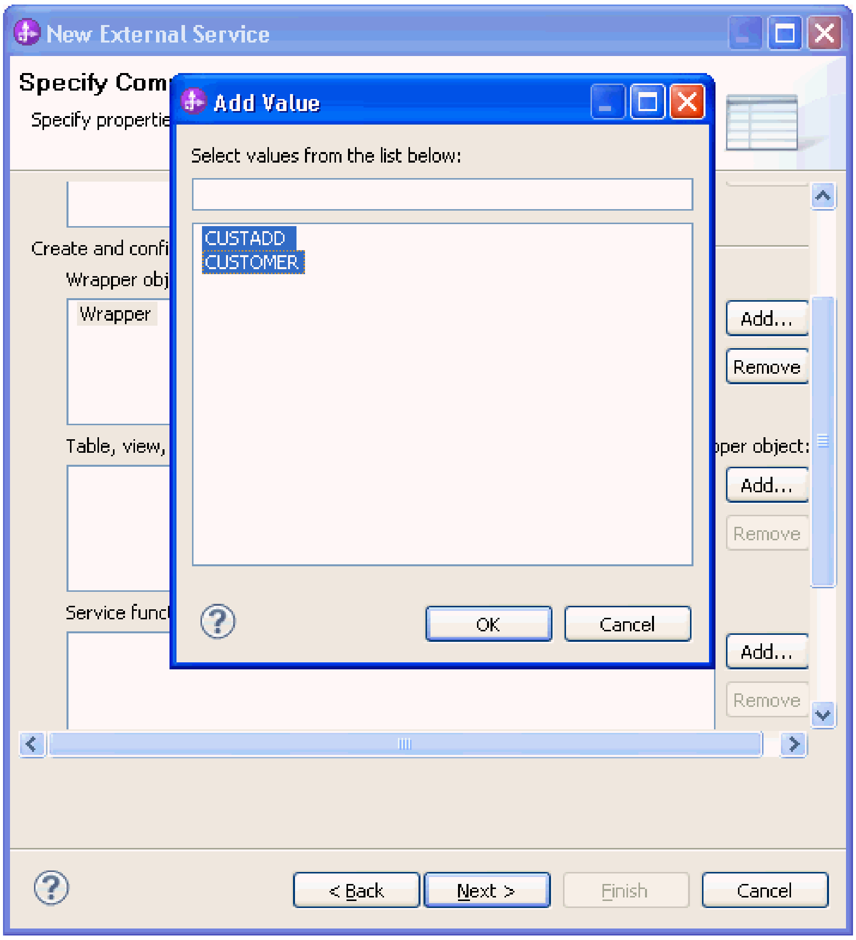
Business object namespace:* http://www.ibm.com/xmlns/prod/websphere/

Specify the relative folder for generated business objects

Folder:

☑ Generate a business graph for each business object

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.

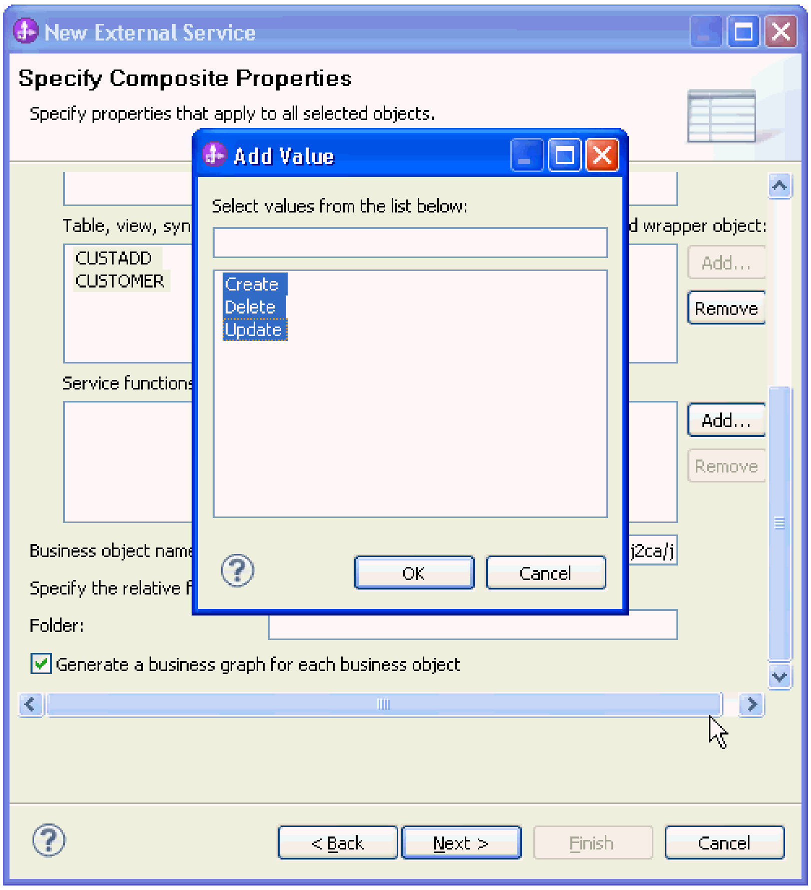


Add Value

Select values from the list below:

CUSTADD
CUSTOMER

3. In the Service **functions for selected wrapper object** area, click **Add** to add functions for the wrapper.



New External Service

Specify Composite Properties

Specify properties that apply to all selected objects.

Add Value

Select values from the list below:

Create
Delete
Update

Table, view, syn

CUSTADD
CUSTOMER

d wrapper object:

Service functions

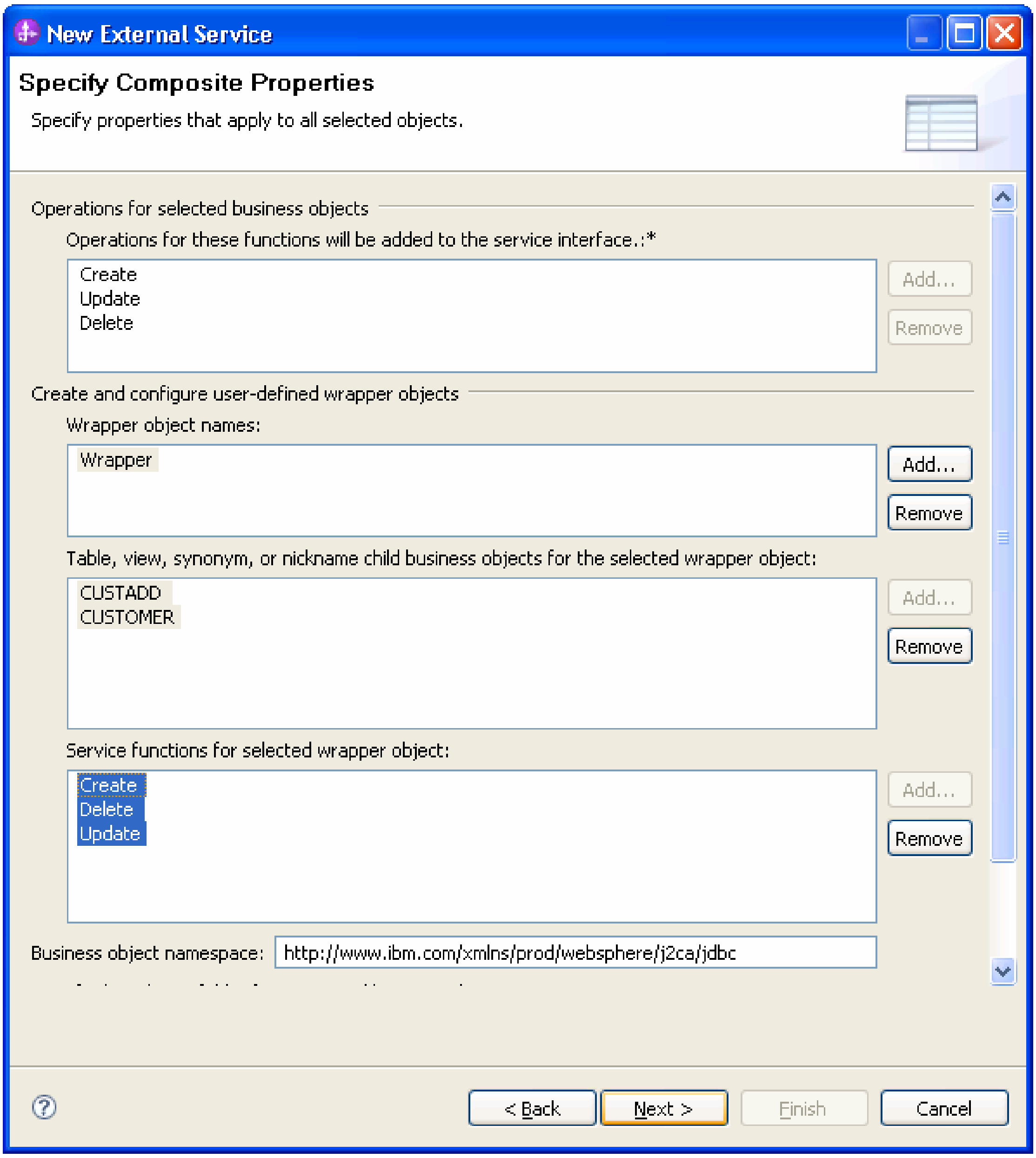
Business object name

Specify the relative f

Folder:

☑ Generate a business graph for each business object

4. Accept the default for the other fields, 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

Create and configure user-defined wrapper objects

Wrapper object names:

Wrapper

Table, view, synonym, or nickname child business objects for the selected wrapper object:

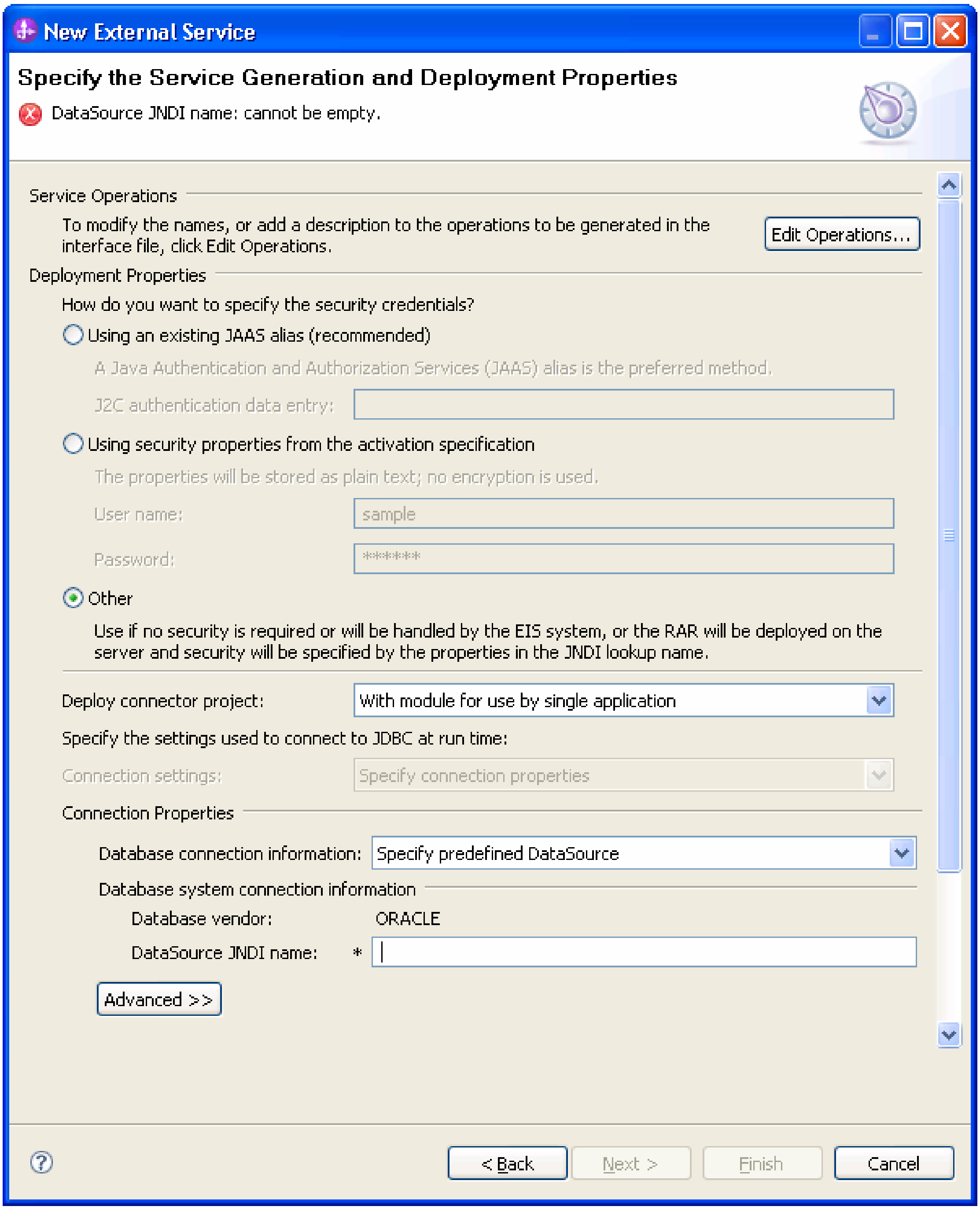
CUSTADD
CUSTOMER

Service functions for selected wrapper object:

Create
Delete
Update

Business object namespace: http://www.ibm.com/xmlns/prod/websphere/j2ca/jdbc

5. In the Specify the Service Generation and Deployment Properties window, select the security credential as **Other**. Select **Specify predefined DataSource** form the **Database connection information** list.



New External Service

Specify the Service Generation and Deployment Properties

DataSource JNDI name: cannot be empty.

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 activation specification

The properties will be stored as plain text; no encryption is used.

User name: sample

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: With module for use by single application

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

Connection settings: Specify connection properties

Connection Properties

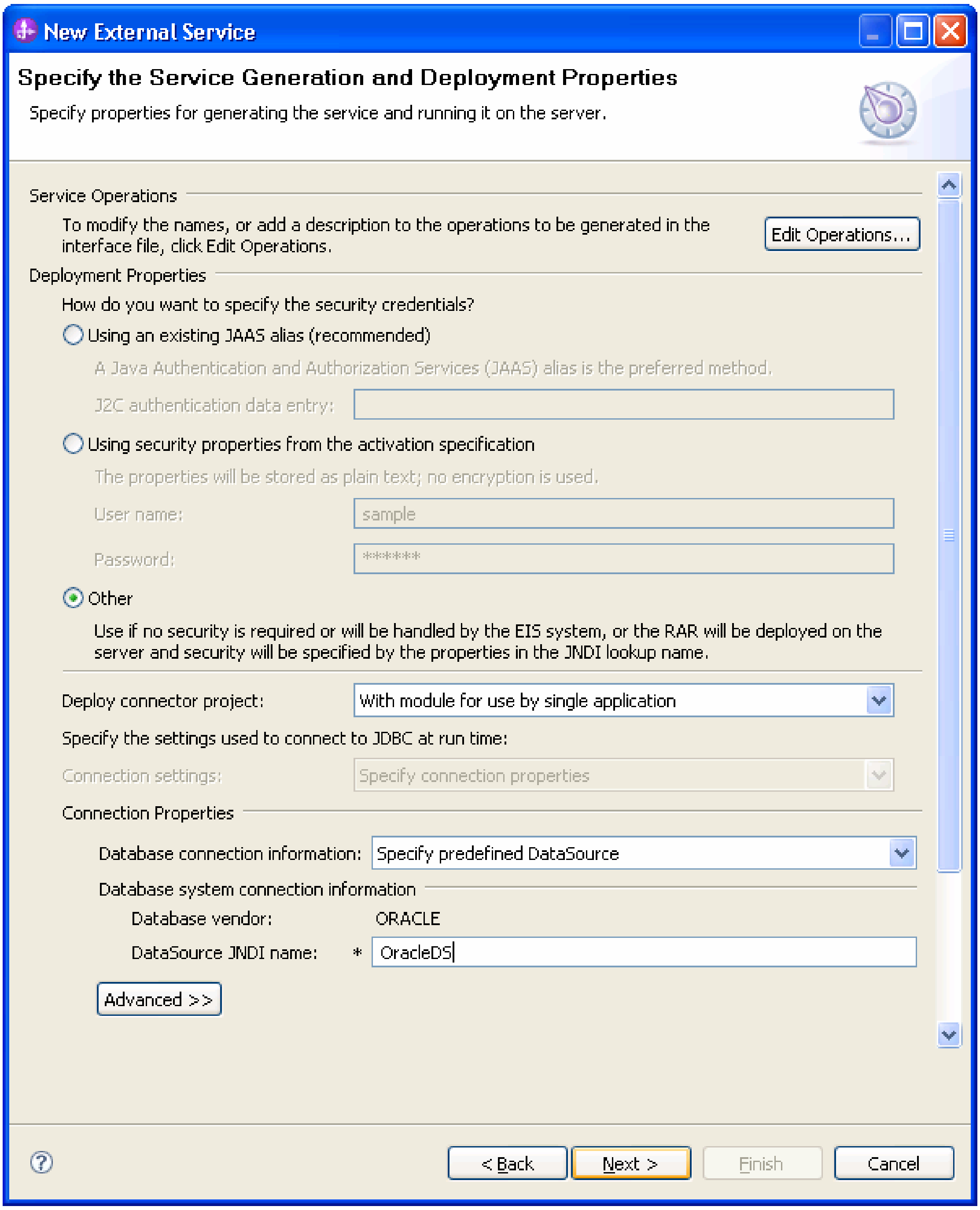
Database connection information: Specify predefined DataSource

Database system connection information

Database vendor: ORACLE

DataSource JNDI name: *

6. In the **DataSource JNDI name** field, enter the JNDI name of the data source, which you created in the previous section. Click **Next**.



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: sample

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: With module for use by single application

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

Connection settings: Specify connection properties

Connection Properties

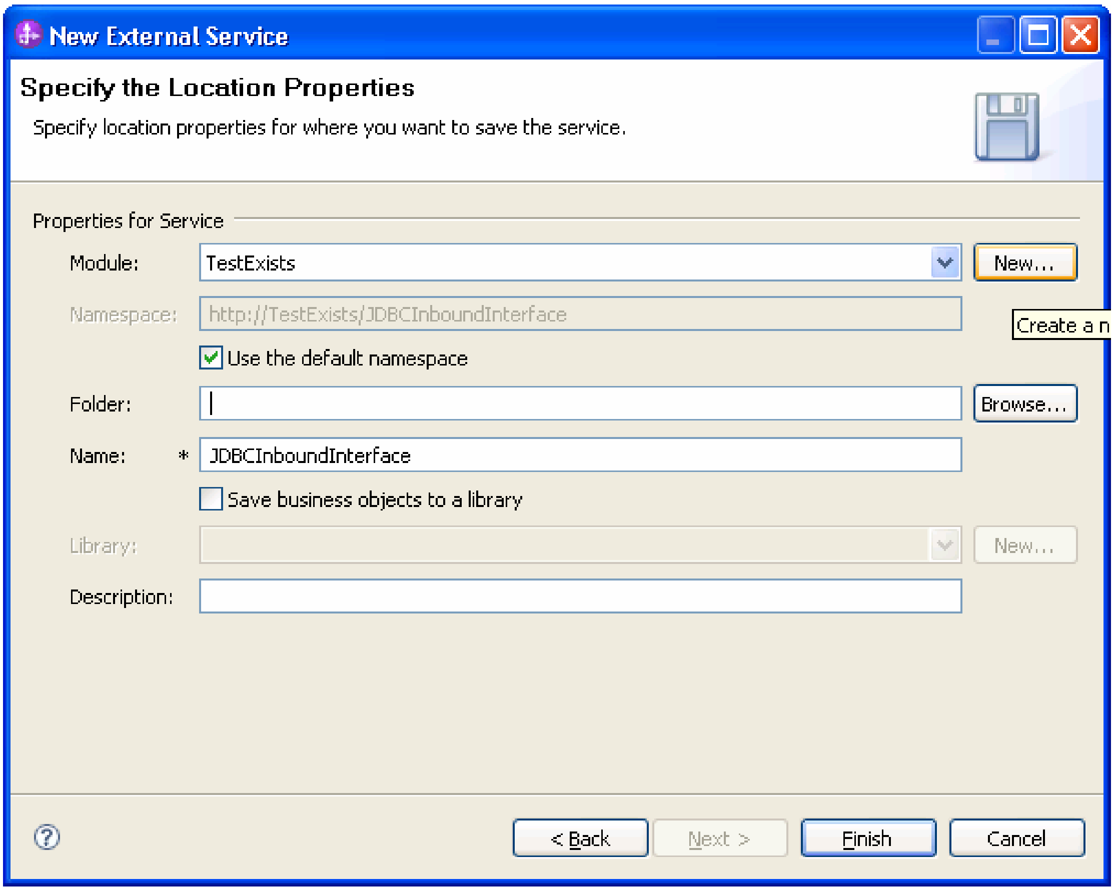
Database connection information: Specify predefined DataSource

Database system connection information

Database vendor: ORACLE

DataSource JNDI name: * OracleDS

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



New External Service

Specify the Location Properties

Specify location properties for where you want to save the service.

Properties for Service

Module: TestExists

Namespace: http://TestExists/JDBCInboundInterface

☑ Use the default namespace

Folder:

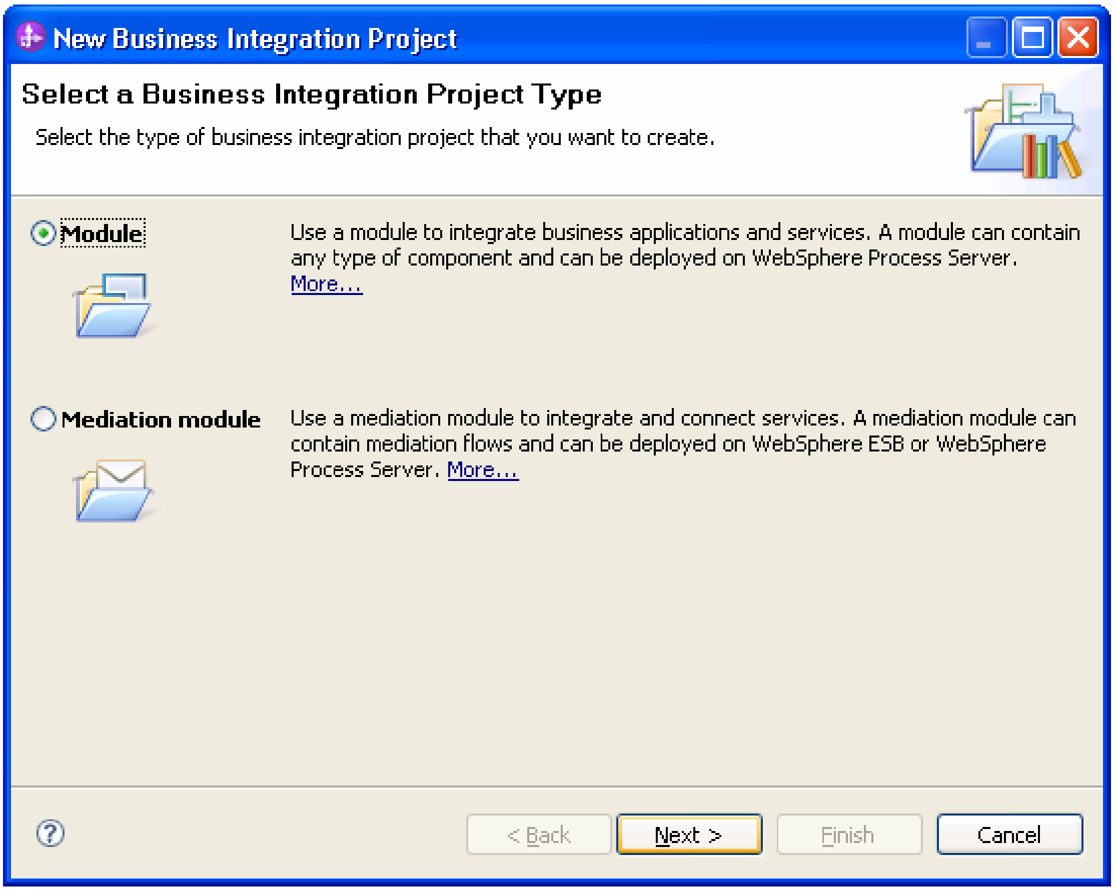
Name: * JDBCInboundInterface

☐ Save business objects to a library

Library:

Description:

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



Select a Business Integration Project Type

Select the type of business integration project that you want to create.

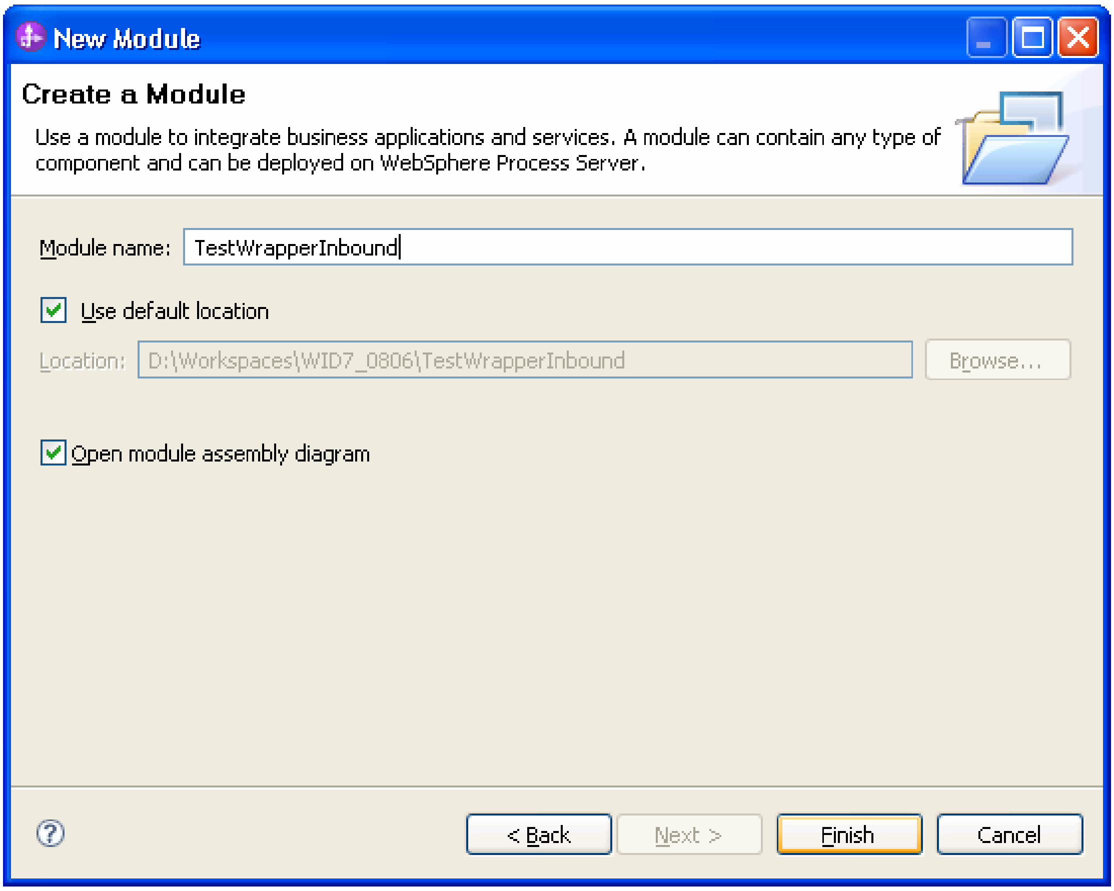
● Module

Use a module to integrate business applications and services. A module can contain any type of component and can be deployed on WebSphere Process Server. More...

○ Mediation module

Use a mediation module to integrate and connect services. A mediation module can contain mediation flows and can be deployed on WebSphere ESB or WebSphere Process Server. More...

9. In the Create a Module window, type **TestWrapperInbound** in the **Module Name** field, click **Finish.**



New Module

Create a Module

Use a module to integrate business applications and services. A module can contain any type of component and can be deployed on WebSphere Process Server.

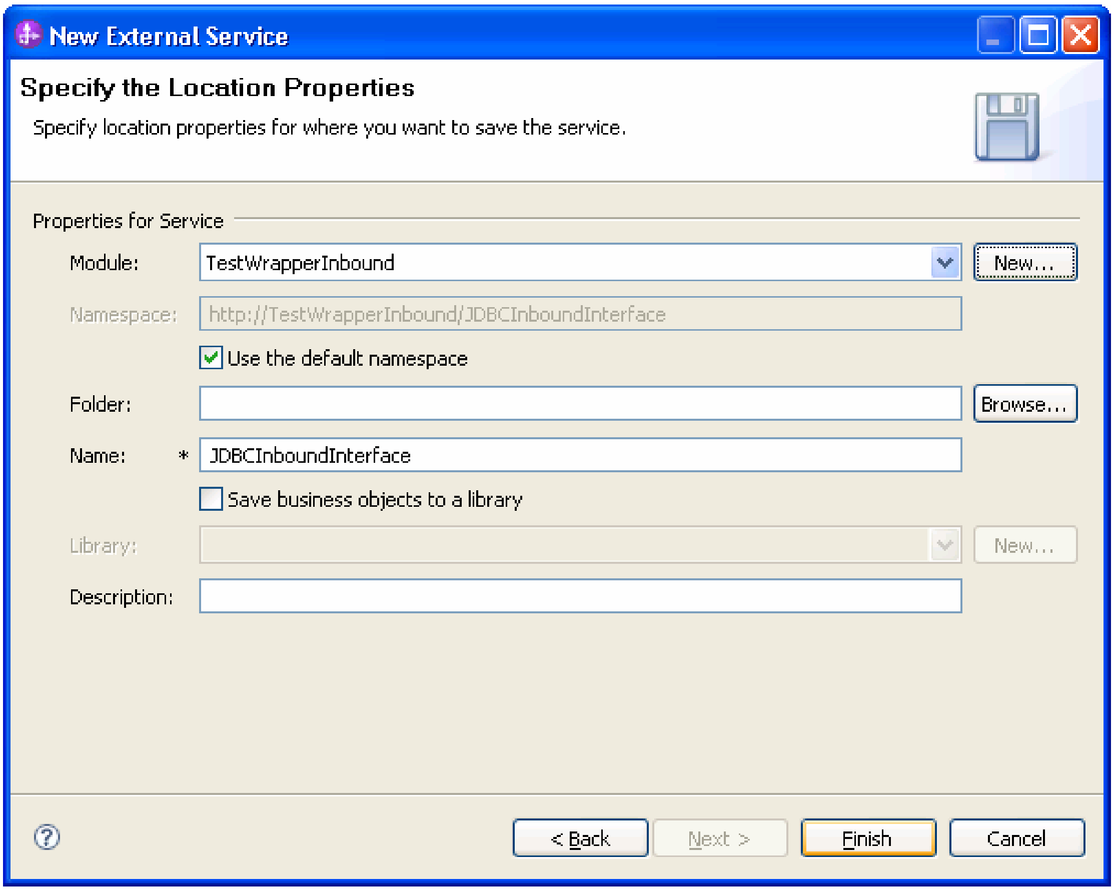
Module name: TestWrapperInbound

☑ Use default location

Location: D:\Workspaces\WID7_0806\TestWrapperInbound

☑ Open module assembly diagram

10. In the Specify the Location Properties window, click **Finish** to finish the wizard**.**



Specify the Location Properties

Specify location properties for where you want to save the service.

Properties for Service

Module: TestWrapperInbound

Namespace: http://TestWrapperInbound/JDBCInboundInterface

☑ Use the default namespace

Folder:

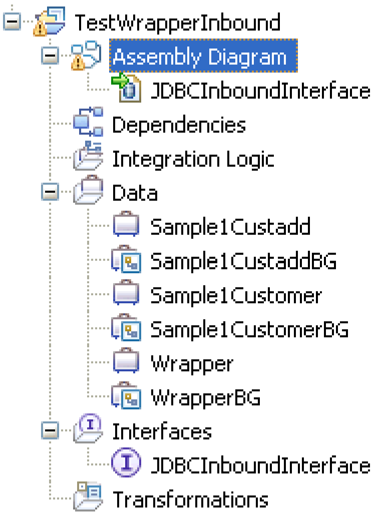
Name: * JDBCInboundInterface

☐ Save business objects to a library

Library:

Description:

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

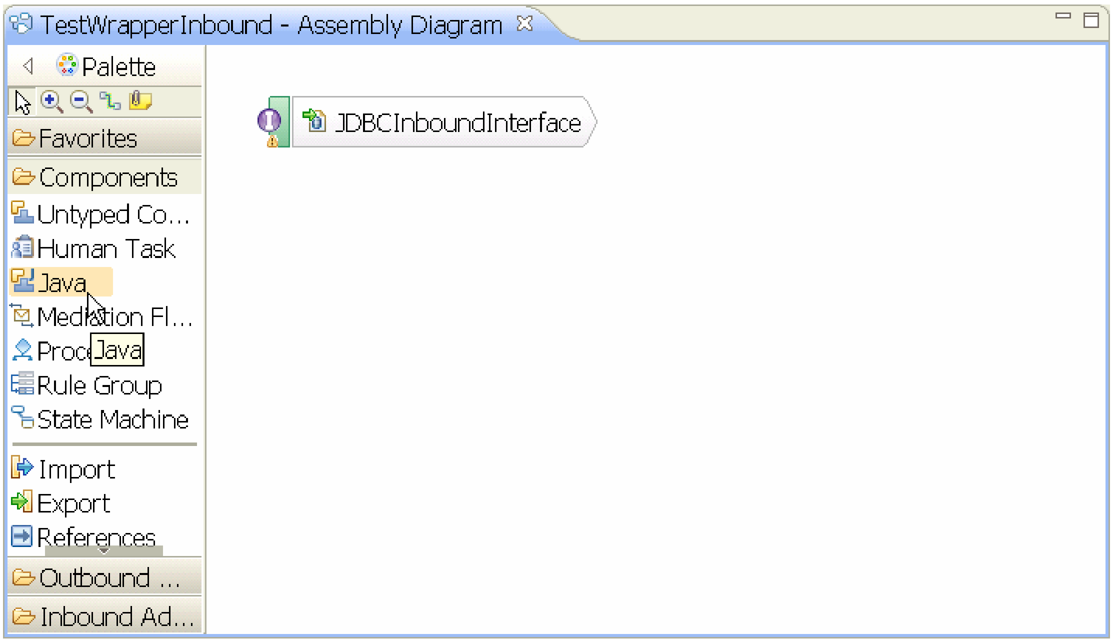


```
TestWrapperInbound
  Assembly Diagram
    JDBCInboundInterface
  Dependencies
  Integration Logic
  Data
    Sample1Custadd
    Sample1CustaddBG
    Sample1Customer
    Sample1CustomerBG
    Wrapper
    WrapperBG
  Interfaces
    JDBCInboundInterface
  Transformations
```

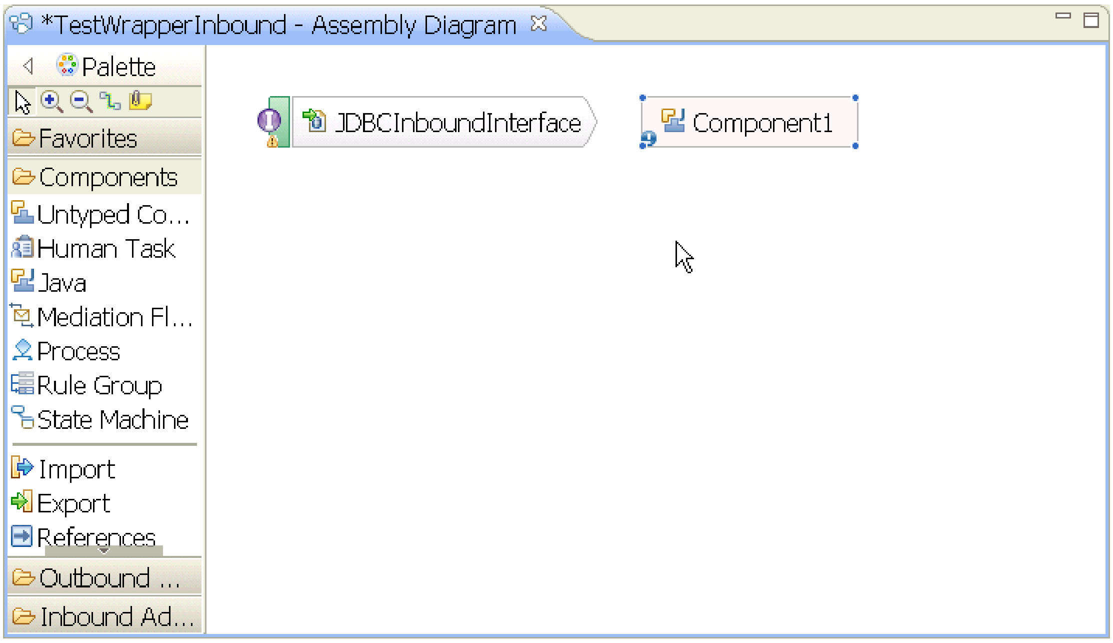
Set up the components to be part of the inbound environment

Next, we add and set up components that to be part of the inbound environment.

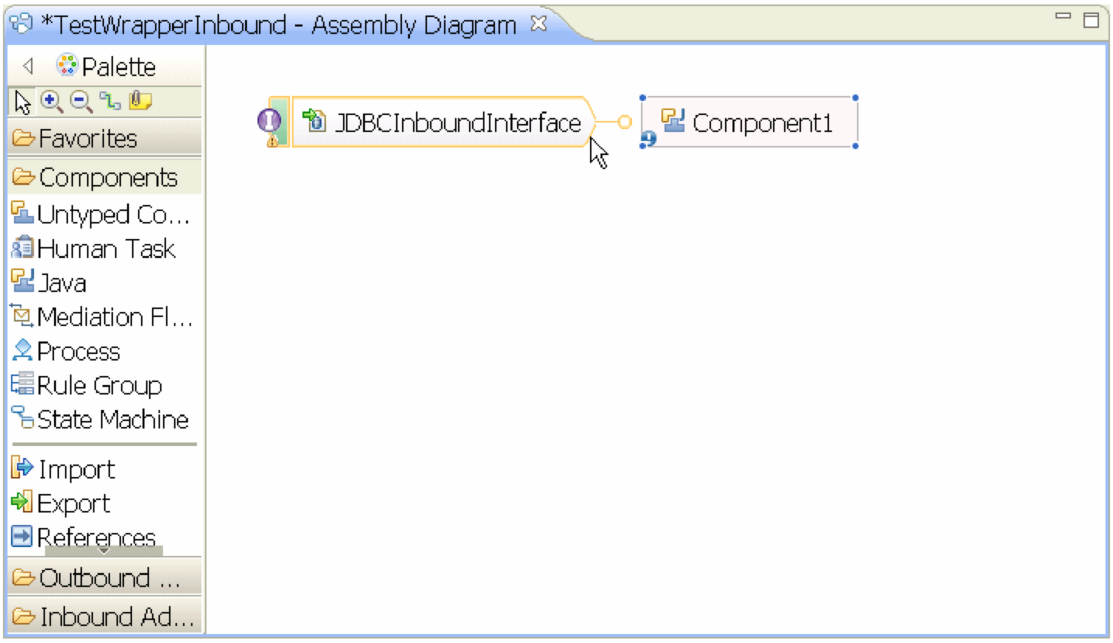
1. In the **Assembly Diagram**, in the Palette, expand **Components** and click **Java** component.



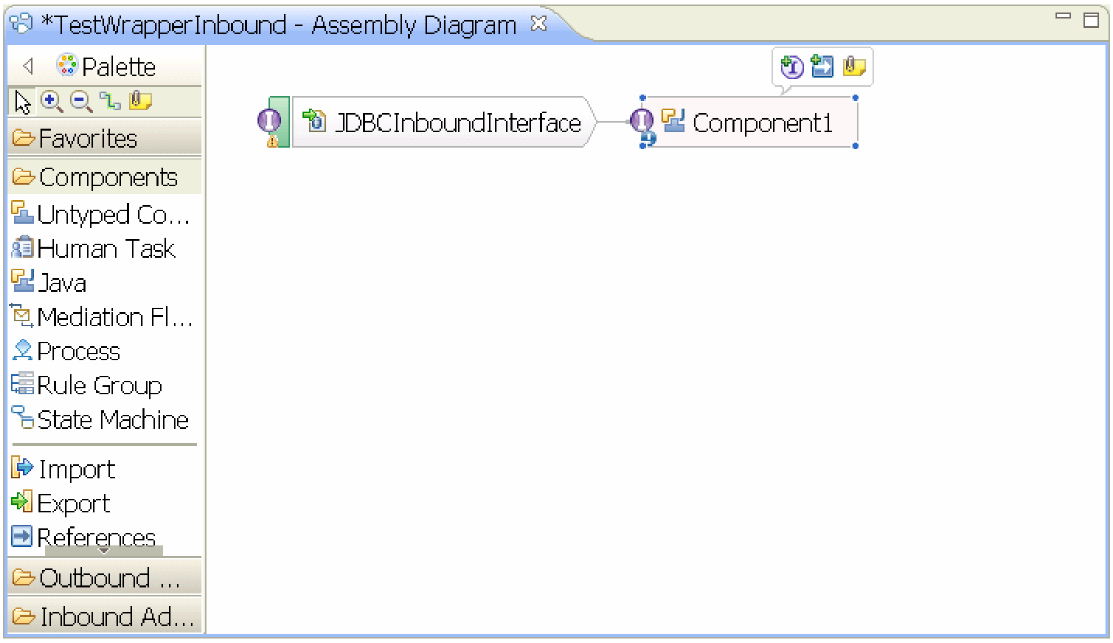
2. Click anywhere in the **TestWrapperInbound -Assembly Diagram** window (white part) to create the **Java** component, **Component1.**



3. To wire **JDBCInboundInterface** to **Component1** hover the cursor over the right end of **JDBCInboundInterface** until a yellow wire appears**.**

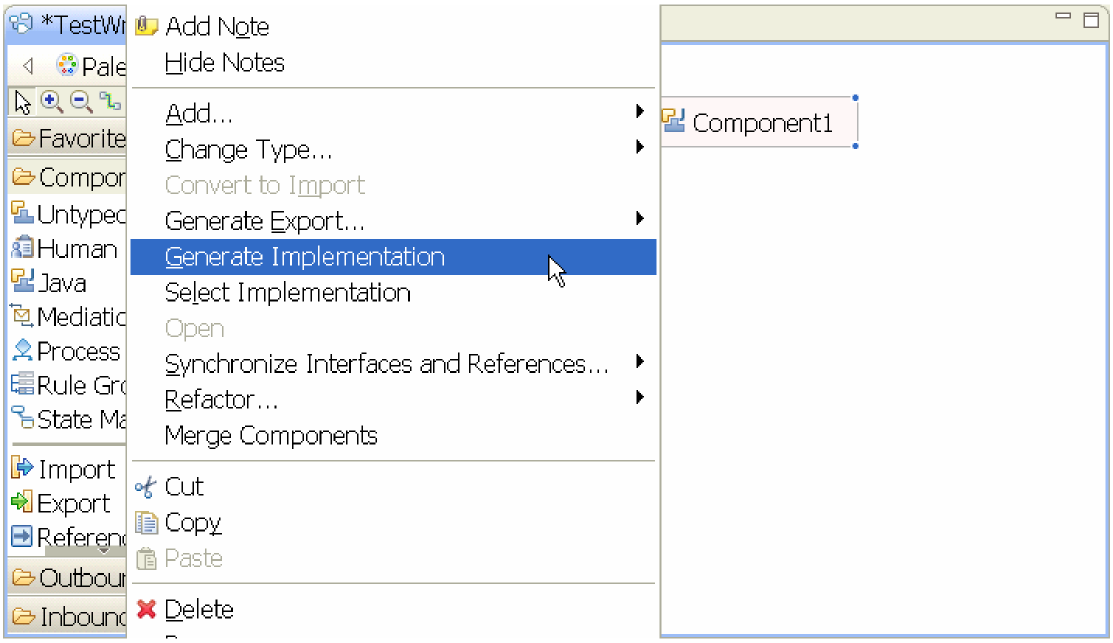


4. Click on the yellow wire and drag it to the left end of **Component1.** When the Add Wire pop-up window displays click **OK.**

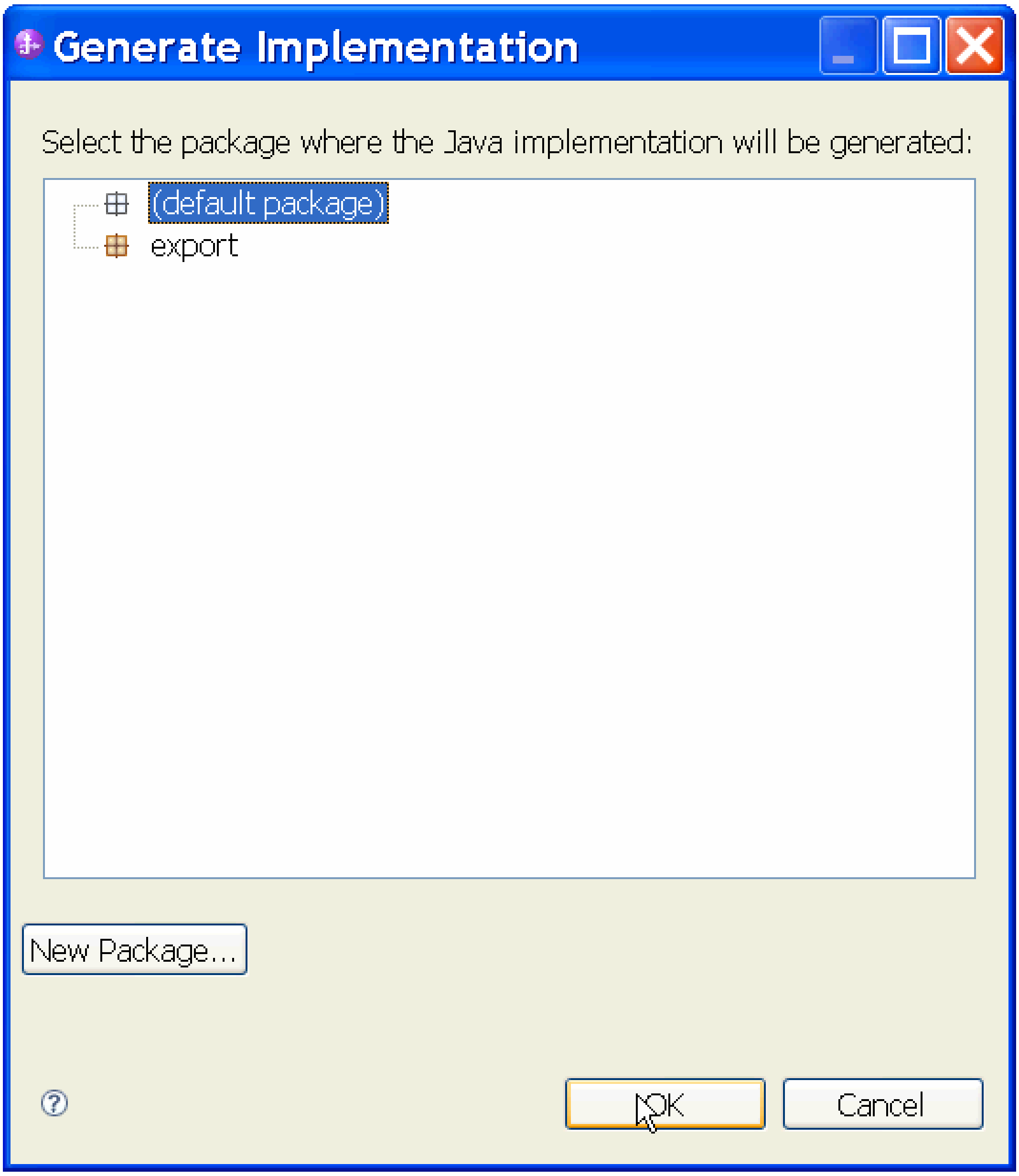


5. Click **File > Save** from the toolbar to save changes made**.**

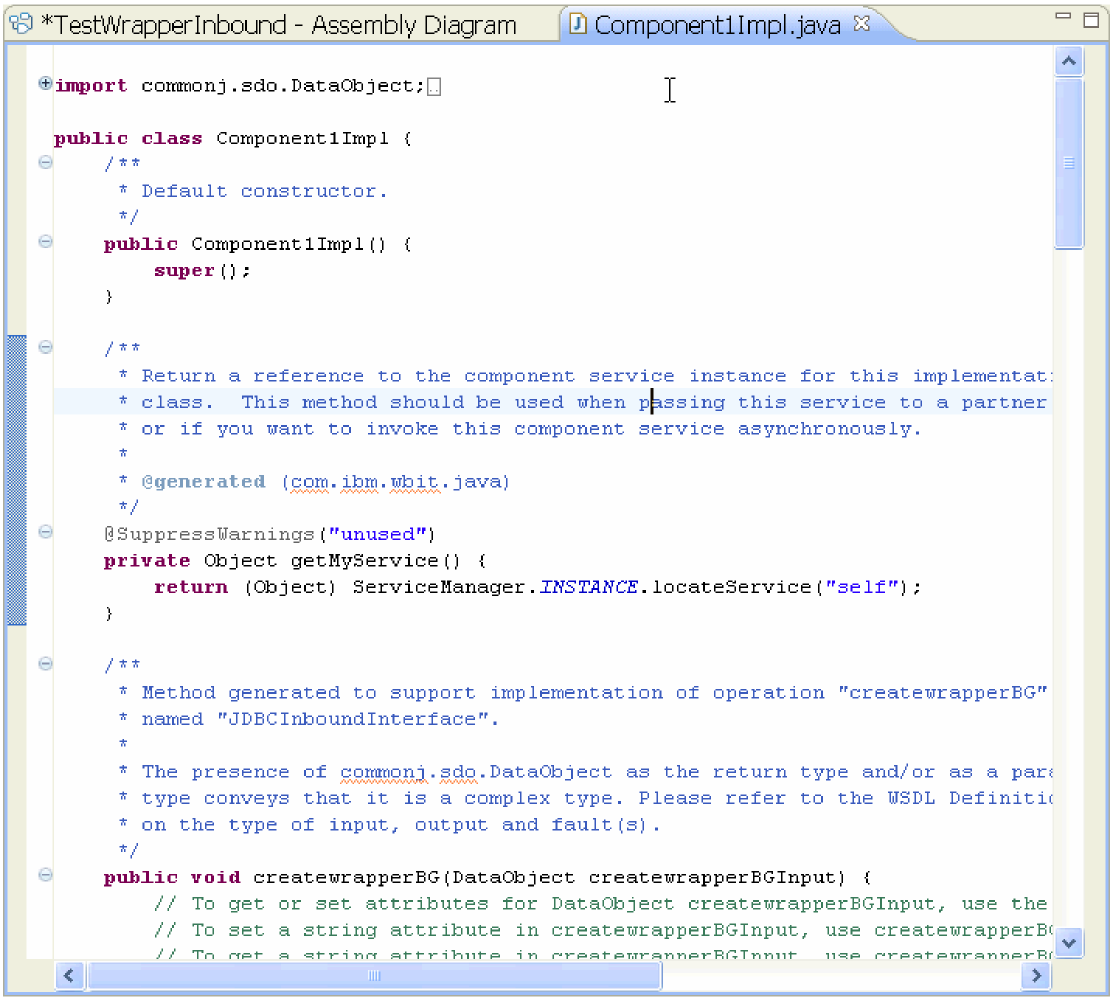
6. Right-click on **Component1** in the **Assembly Diagram** and select **Generate implementation.**



7. In the Generate Implementation window, select **default package** and click **OK.**



8. The Java Editor will open showing the **Component1Impl.java** file.



```
import commonj.sdo.DataObject;

public class Component1Impl {
    /**
     * Default constructor.
     */
    public Component1Impl() {
        super();
    }

    /**
     * Return a reference to the component service instance for this implementat
     * 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.wbit.java)
     */
    @SuppressWarnings("unused")
    private Object getMyService() {
        return (Object) ServiceManager.INSTANCE.locateService("self");
    }

    /**
     * Method generated to support implementation of operation "createwrapperBG"
     * named "JDBCInboundInterface".
     *
     * The presence of commonj.sdo.DataObject as the return type and/or as a para
     * type conveys that it is a complex type. Please refer to the WSDL Definitio
     * on the type of input, output and fault(s).
     */
    public void createwrapperBG(DataObject createwrapperBGInput) {
        // To get or set attributes for DataObject createwrapperBGInput, use the
        // To set a string attribute in createwrapperBGInput, use createwrapperB(
        // To get a string attribute in createwrapperBGInput, use createwrapperB(
```

9. Scroll down and locate the **createwrapperBG** 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();
  }
```

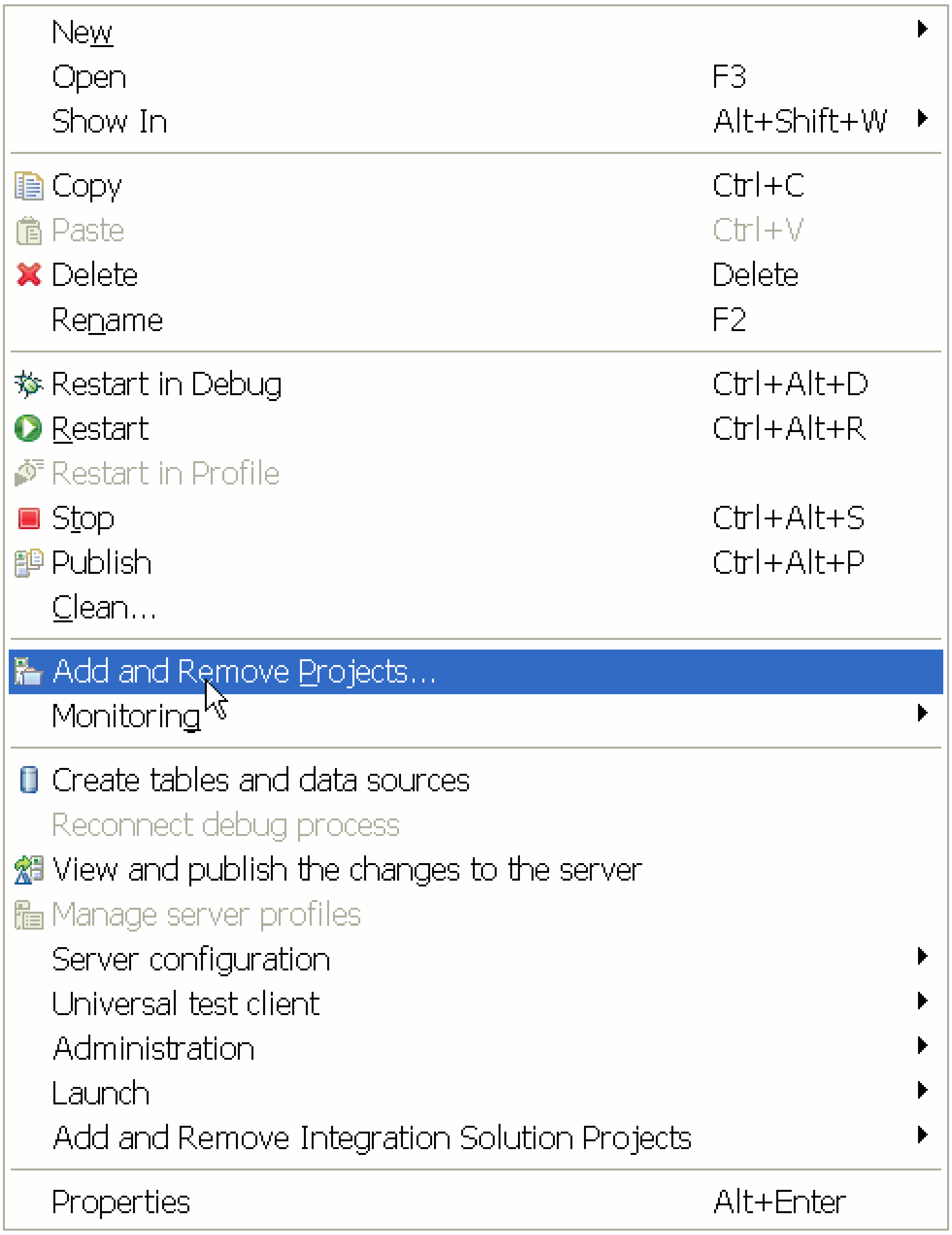
15. Click on **File > Save** from the toolbar to save the changes made.

Deploy the module to the test environment

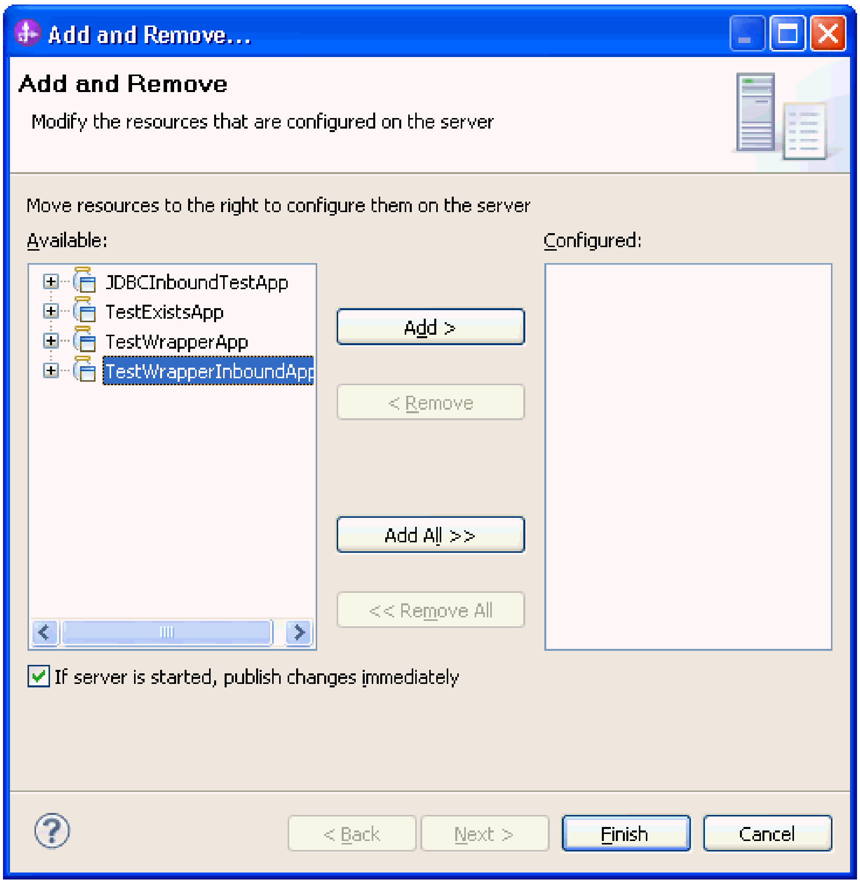
After running the external service wizard, you will have an SCA module that contains an Enterprise Information System (EIS) export. You must install this SCA module in the IBM Integration Designer integration test client to deploy it. To do this, you must add the SCA module you created earlier to the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

Steps for adding the SCA module to the server:

1. In IBM Integration Designer, 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**.



A window is displayed that shows the available projects on the IBM Integration Designer workspace.



Add and Remove...

Add and Remove

Modify the resources that are configured on the server

Move resources to the right to configure them on the server

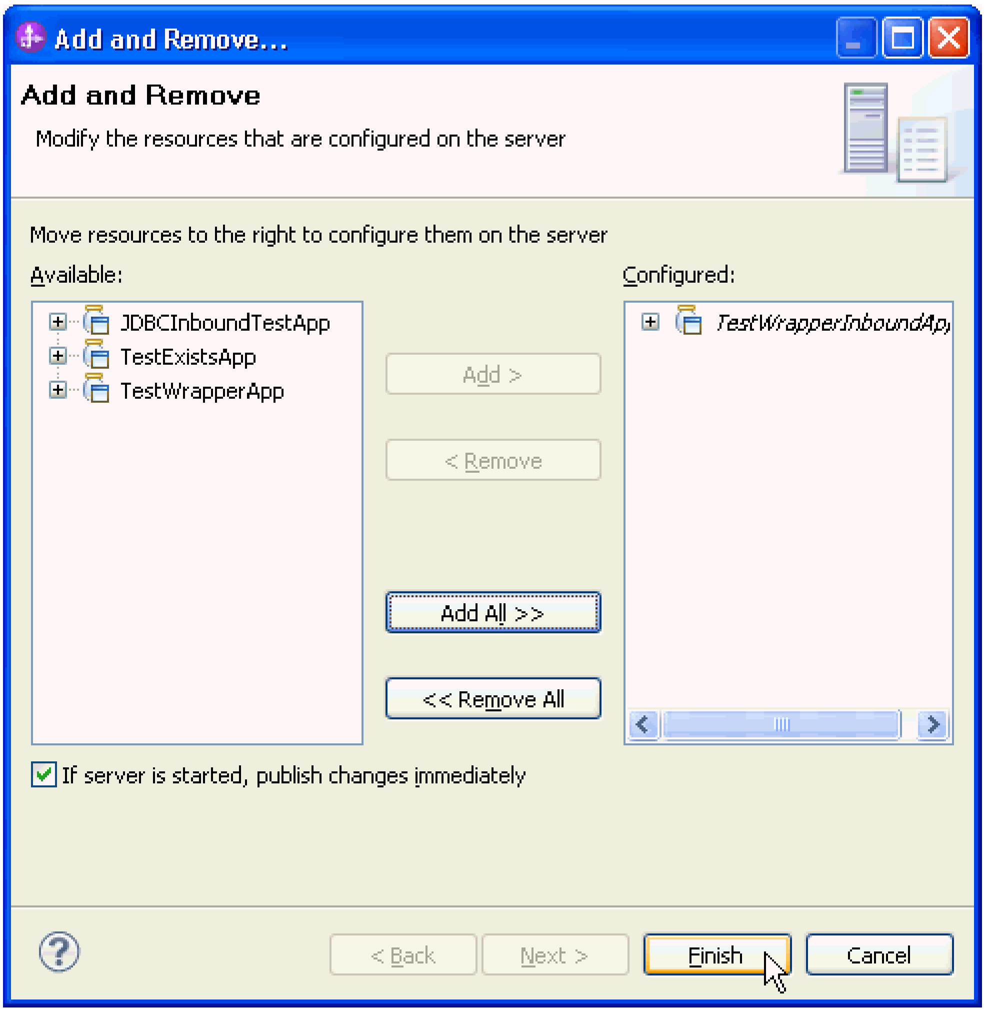
Available:

- JDBCInboundTestApp
- TestExistsApp
- TestWrapperApp
- TestWrapperInboundApp

Configured:

☑ If server is started, publish changes immediately

4. In the Add and Remove Projects window, select your project (TestWrapperInbound) and click **Add** to configure the project on the server. Click **Finish.**



Add and Remove...

Add and Remove

Modify the resources that are configured on the server

Move resources to the right to configure them on the server

Available:
- JDBCInboundTestApp
- TestExistsApp
- TestWrapperApp

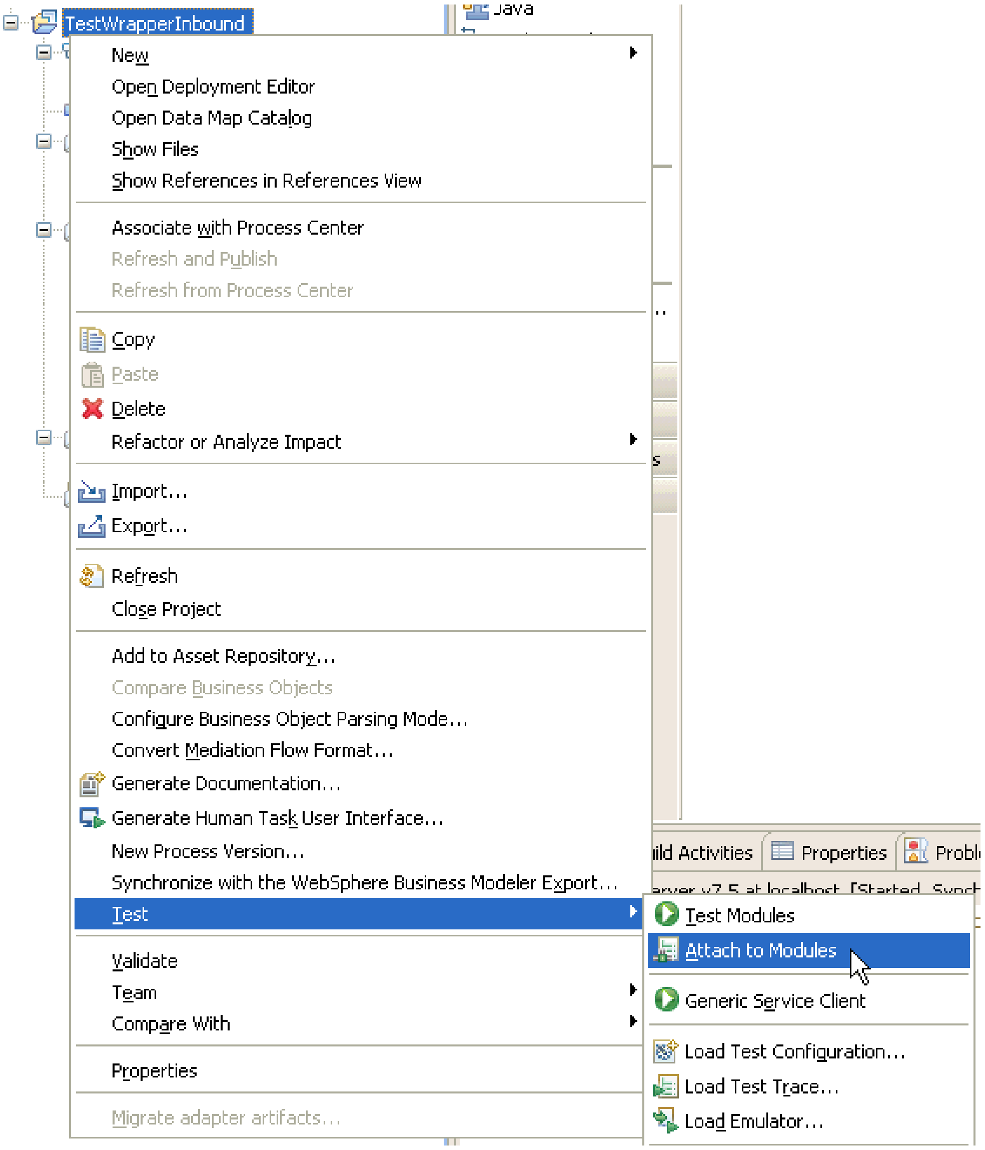
Configured:
- *TestWrapperInboundAp*

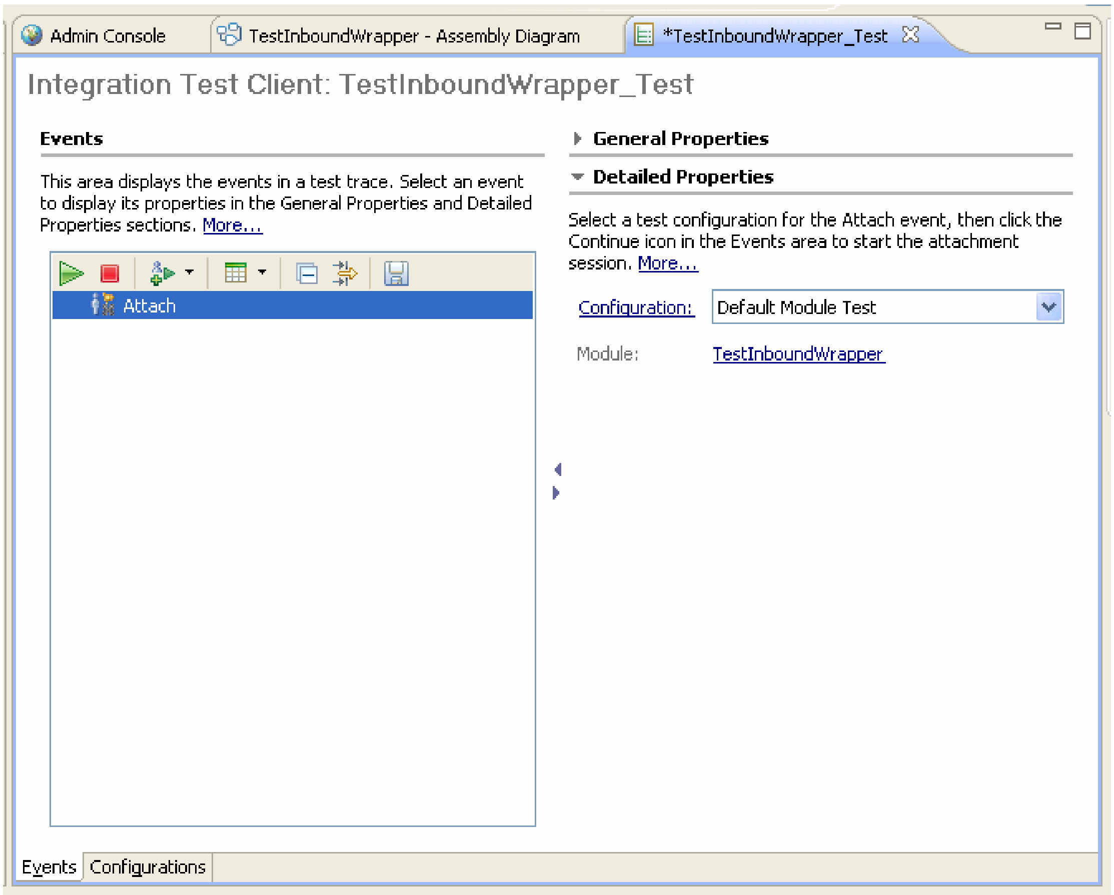
☑ If server is started, publish changes immediately

Test the assembled adapter application

Test the assembled adapter application using the IBM Integration Designer integration test client.

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


TestWrapperInbound
Java
New
Open Deployment Editor
Open Data Map Catalog
Show Files
Show References in References View
Associate with Process Center
Refresh and Publish
Refresh from Process Center
Copy
Paste
Delete
Refactor or Analyze Impact
Import...
Export...
Refresh
Close Project
Add to Asset Repository...
Compare Business Objects
Configure Business Object Parsing Mode...
Convert Mediation Flow Format...
Generate Documentation...
Generate Human Task User Interface...
New Process Version...
Synchronize with the WebSphere Business Modeler Export...
Test
Validate
Team
Compare With
Properties
Migrate adapter artifacts...
Test Modules
Attach to Modules
Generic Service Client
Load Test Configuration...
Load Test Trace...
Load Emulator...
ild Activities
Properties
Probl



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...

Attach

General Properties

Detailed Properties

Select a test configuration for the Attach event, then click the Continue icon in the Events area to start the attachment session. More...

Configuration: Default Module Test

Module: TestInboundWrapper

Events | Configurations

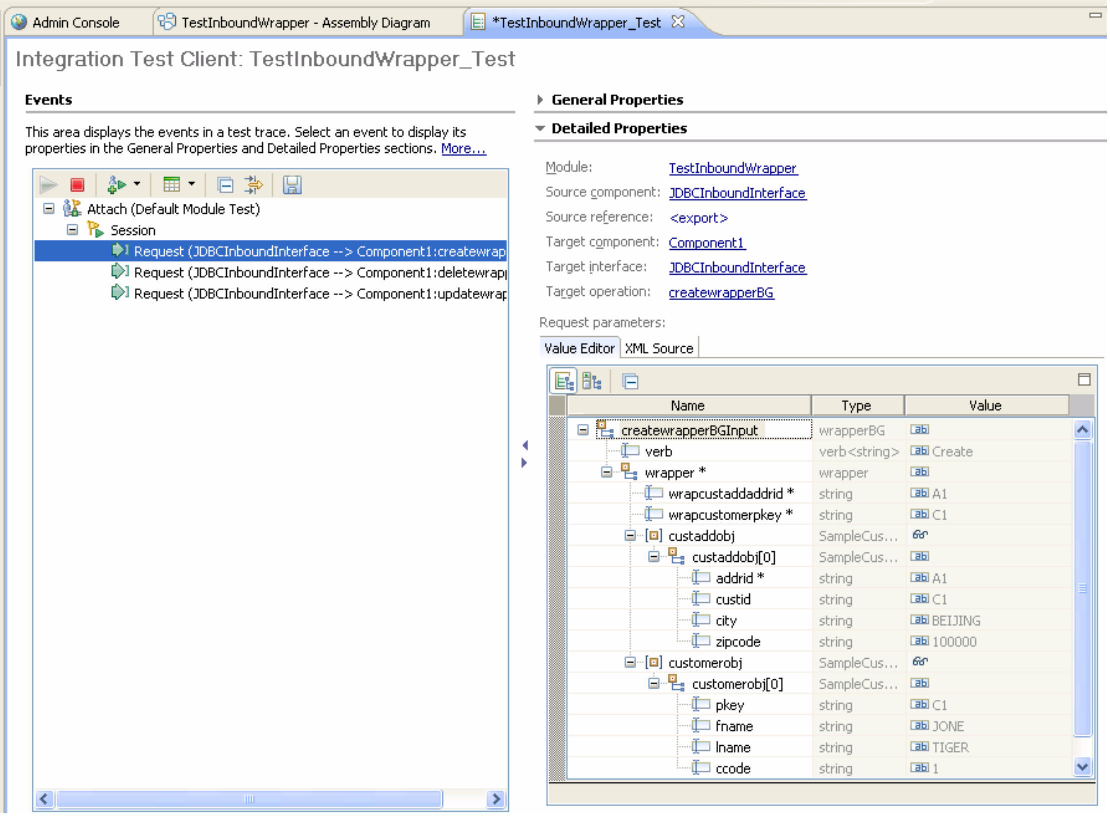
2. To execute the service, click ▷.

3. Execute below SQL scripts to generate events:

```
INSERT INTO WBIA_JDBC_EVENTSTORE
   (EVENT_ID,OBJECT_KEY,OBJECT_NAME,OBJECT_FUNCTIO
N,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_FUNCTIO
N,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_FUNCTIO
N,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...

- Attach (Default Module Test)
  - Session
    - Request (JDBCInboundInterface --> Component1:createwrap
    - Request (JDBCInboundInterface --> Component1:deletewrap
    - Request (JDBCInboundInterface --> Component1:updatewrap

General Properties

Detailed Properties

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
createwrapperBGInput	wrapperBG	
verb	verb<string>	Create
wrapper *	wrapper	
wrapcustaddaddrid *	string	A1
wrapcustomerpkey *	string	C1
custaddobj	SampleCus...	
custaddobj[0]	SampleCus...	
addrid *	string	A1
custid	string	C1
city	string	BEIJING
zipcode	string	100000
customerobj	SampleCus...	
customerobj[0]	SampleCus...	
pkey	string	C1
fname	string	JONE
lname	string	TIGER
ccode	string	1

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.

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