IBM WEBSPHERE ADAPTER FOR JDBC V6.2 – LAB EXERCISE

JDBC outbound lab – Hierarchy business objects

What this exercise is about

The objective of this lab is to provide you with an understanding of the WebSphere Adapter for JDBC and outbound request processing.

Lab requirements

- WebSphere Integration Developer V6.2 installed
- WebSphere Process Server V6.2 test environment installed
- WebSphere Adapter for JDBC V6.2 installed
- Sample code in the directory C:\Labfiles62\JDBC (Windows) or /tmp/LabFiles62/JDBC (Linux)

What you should be able to do

At the end of this lab you should be able to:

- Install and deploy the Adapter for JDBC and integrate it into an SCA application for use with outbound request processing. This exercise demonstrate one scenario which is creating a multiple cardinality business objects Hierarchy using External Service Wizard, instead of editing the parent and child Business Objects using Business Object Editor. The 'Keep Relationship', 'Ownership' and 'Required' ASIs can also be specified in Configuration Parameters Screen of External Service Wizard.

Introduction

This lab introduces you to the WebSphere Adapter for JDBC and the processing of outbound requests to a table in a database. It uses a JDBCTEST Derby database that contains a CUSTOMER table. In the lab, you will import the JDBC Adapter into WebSphere Integration Developer and run External Service to input connection information, create a service description, and discover objects existing in the specified database. You will then assemble an SCA application, wiring together a stand-alone reference and the EIS import file. To test your application, you will use the WebSphere Test Environment and Component Test, exercising various outbound requests, such as create, delete, retrieve, and retrieveAll.

Exercise instructions

Some instructions in this lab can be Windows operating-system specific. If you plan on running the lab on an operating-system other than Windows, you will need to run the appropriate commands, and use appropriate files (.sh or .bat) for your operating system. The directory locations are specified in the lab instructions using symbolic references, as follows:

Reference Variable	Windows Location	AIX®/UNIX® Location
<LAB_NAME>	JDBCOutbound	
<WID_HOME>	C:\Program Files\IBM\WebSphere\ID\6.2	
<WPS_HOME>	<WID_HOME>\runtimes\bi_v62	
<JDBCADAPTER_HOME>	<WID_HOME>\Resource Adapters\JDBC	
<LAB_FILES>	C:\Labfiles62	/tmp/Labfiles62
<TEMP>	C:\temp	/tmp

Windows users' note: When directory locations are passed as parameters to a Java™ program such as EJBdeploy or wsadmin, it is necessary to replace the backslashes with forward slashes to follow the Java convention. For example, C:\LabFiles62\ is replaced by C:/LabFiles62/

Note that the previous table is relative to where you are running WebSphere Integration Developer.

Instructions if using a remote server for testing

Note that the previous table is relative to where you are running WebSphere Integration Developer. The following table is related to where you are running the remote test environment:

Reference variable	Example: Remote Windows test server location	Example: Remote z/OS® test server location	Input your values for the remote location of the test server
<SERVER_NAME>	server1	sssr011	
<WAS_HOME>	C:\Program Files\IBM\WebSphere\AppServer	/etc/sscell/AppServer	
<HOSTNAME>	localhost	mvsxxx.rtp.raleigh.ibm.com	
<SOAP_PORT>	8880	8880	
<TELNET_PORT>	N/A	1023	
<PROFILE_NAME>	AppSrv01	default	
<USERID>	N/A	ssadmin	
<PASSWORD>	N/A	fr1day	

Instructions for using a remote testing environment, such as z/OS, AIX® or Solaris, can be found at the end of this document, in the section "**Task: Adding remote server to the WebSphere Integration Developer test environment**".

Part 1: Create the JDBCTEST database using Derby

In this part you will create the JDBCTEST database along with the CUSTOMER and ADDRESS table for user data.

If you choose to run the JDBCInbound lab first, then you only need to create additional table, ADDRESS for this exercise. If you choose to run JDBCOutbound User Defined SQL lab first, then you can skip this part and proceed to Part 2.

If you are using a remote testing environment on z/OS,it is recommended to create the Derby databases on your Windows machine and then FTP the data files to the z/OS environment. Be sure to upload the files in the /log and /seg0 directories in binary format, and the "service.properties" file in ASCII format. You used a directory on the host called /tmp/LabFiles62/DerbyDB. Also be sure to use the CHMOD command to change all the files to 777.

____ 1. Start Derby ij command

Note: The Derby embedded driver that is being used in the lab, supports a connection from only one JVM at a time. You can have either the server running and connected to the JDBCTEST database, or the vendor GUI tool connected to the JDBCTEST database; but not both at the same time.

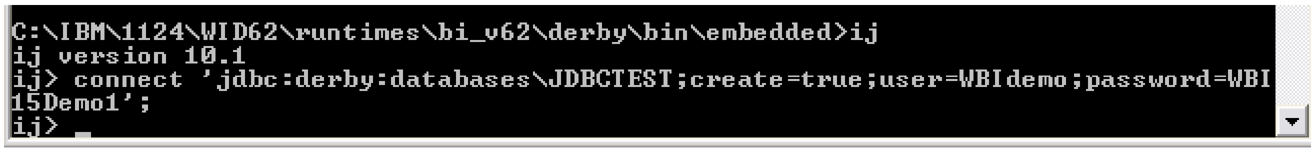
__ a. Open a command prompt window, navigate to this subdirectory, and run Derby's interactive JDBC scripting tool.

<WPS_HOME>\derby\bin\embedded>ij

____ 2. Using the command line, create the JDBCTEST database if it does not already exist. If you have already completed the JDBCOutbound lab, the database, tables, and triggers are the same.

__ a. Type in this script to create JDBC database as user '**Wbidemo'** and password '**Wbi15Demo1'** in the Console view.

connect 'jdbc:derby:databases\JDBCTEST;create=true;user=Wbidemo;password=Wbi15Demo1';



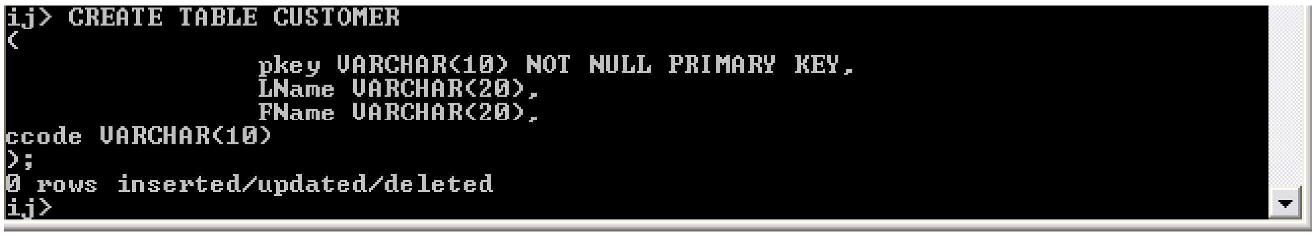
```
C:\IBM\1124\WID62\runtimes\bi_v62\derby\bin\embedded>ij
ij version 10.1
ij> connect 'jdbc:derby:databases\JDBCTEST;create=true;user=WBIdemo;password=WBI
15Demo1';
ij> _
```

____ 3. Enter SQL to create two tables: CUSTOMER and ADDRESS tables for user data.

NOTE: For your convenience, these SQL code snippets can be found in **<LAB_FILES>\JDBC\snippets\CUSTOMERSQL.txt**

__ a. Paste this script into the command window:

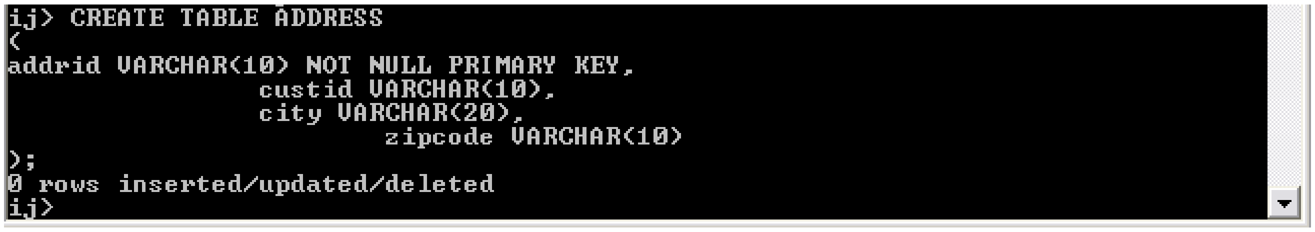
```
CREATE TABLE CUSTOMER
  (
        pkey VARCHAR(10) NOT NULL PRIMARY KEY,
        LName VARCHAR(20),
        FName VARCHAR(20),
        ccode VARCHAR(10)
  );
```



```
ij> CREATE TABLE CUSTOMER
(
                pkey VARCHAR(10) NOT NULL PRIMARY KEY,
                LName VARCHAR(20),
                FName VARCHAR(20),
ccode VARCHAR(10)
);
0 rows inserted/updated/deleted
ij>
```

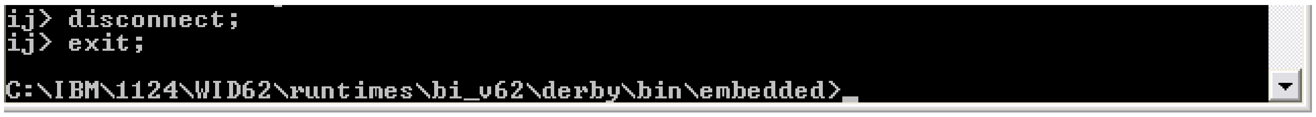
NOTE: For your convenience, these SQL code snippets can be found in **<LAB_FILES>\JDBC\snippets\ADDRESSSQL.txt**

__ b. Paste this script into the command window:



```
ij> CREATE TABLE ADDRESS
(
addrid VARCHAR(10) NOT NULL PRIMARY KEY,
                custid VARCHAR(10),
                city VARCHAR(20),
                        zipcode VARCHAR(10)
);
0 rows inserted/updated/deleted
ij>
```

____ 4. Disconnect and exit the ij command window



```
ij> disconnect;
ij> exit;

C:\IBM\1124\WID62\runtimes\bi_v62\derby\bin\embedded>_
```

Note: You must exit the ij command window, before starting the server as the Derby “embedded” database driver used in this lab allows for only one JVM connection at a time.

Remember that if you are using a remote test environment for this lab, you need to FTP the Derby database files to the remote system, as explained at the beginning of this section.

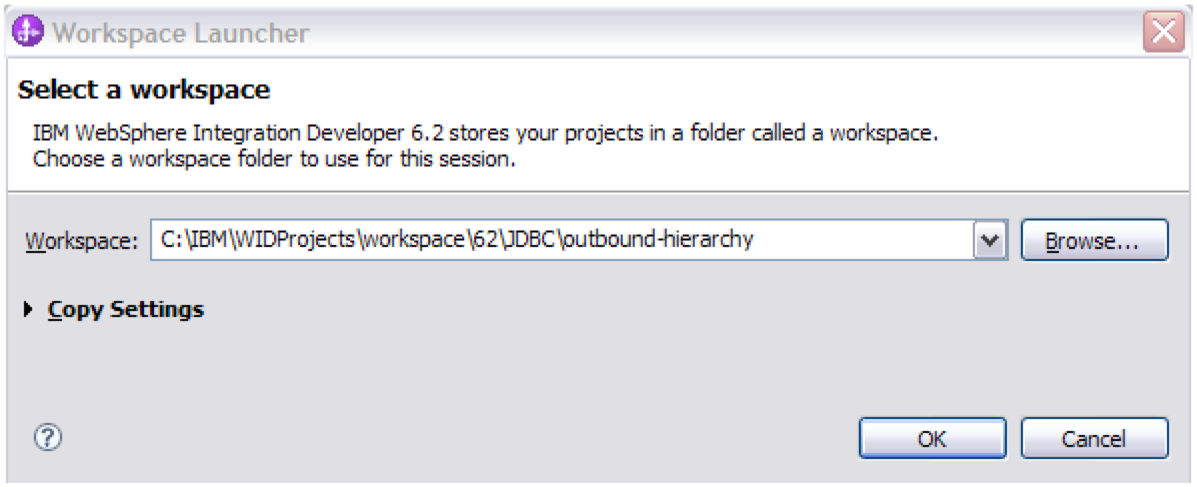
Part 2: Set up the development environment

In this part, you will start WebSphere Integration Developer, using a new workspace, import RAR file and set up the WebSphere Process Server to be used as the WebSphere Test Environment (WTE).

____ 1. Start WebSphere Integration Developer V6.2 with a new workspace.

__ a. From the start menu select **Start > Programs > IBM Software Development Platform > IBM WebSphere Integration Developer 6.2> IBM WebSphere Integration Developer V6.2**

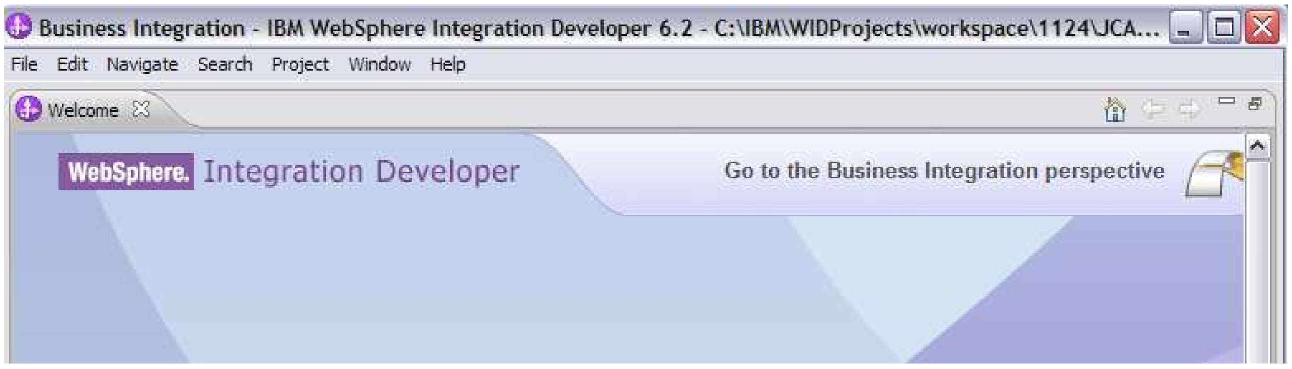
__ b. When prompted enter **<LAB_FILES>\JDBC\outbound-hierarchy\** for your workspace and click **OK**



IBM WebSphere Integration Developer 6.2 stores your projects in a folder called a workspace.
Choose a workspace folder to use for this session.

Workspace: C:\IBM\WIDProjects\workspace\62\JDBC\outbound-hierarchy

__ c. When WebSphere Integration Developer V6.2 opens, close the **Welcome page**


Business Integration - IBM WebSphere Integration Developer 6.2 - C:\IBM\WIDProjects\workspace\1124\JCA...
File Edit Navigate Search Project Window Help
Welcome
WebSphere. Integration Developer
Go to the Business Integration perspective

Part 3: Create the JDBC Outbound application

In this part you will run External Service to discover objects and create the necessary SCA artifacts, and assemble the adapter into an SCA application.

NOTE: If you are using a remote test environment on z/OS, you will first need to enable your Derby database for network access using these steps:

On the host, login to the system using telnet.

In order to allow connections to the network server, you need to edit the /<WPS_HOME>/cloudscape/db2j.properties file. Delete the '#' (pound sign) to uncomment this line:

```
db2j.drda.host=0.0.0.0
```

Navigate to /<WPS_HOME>/derby/bin/networkServer

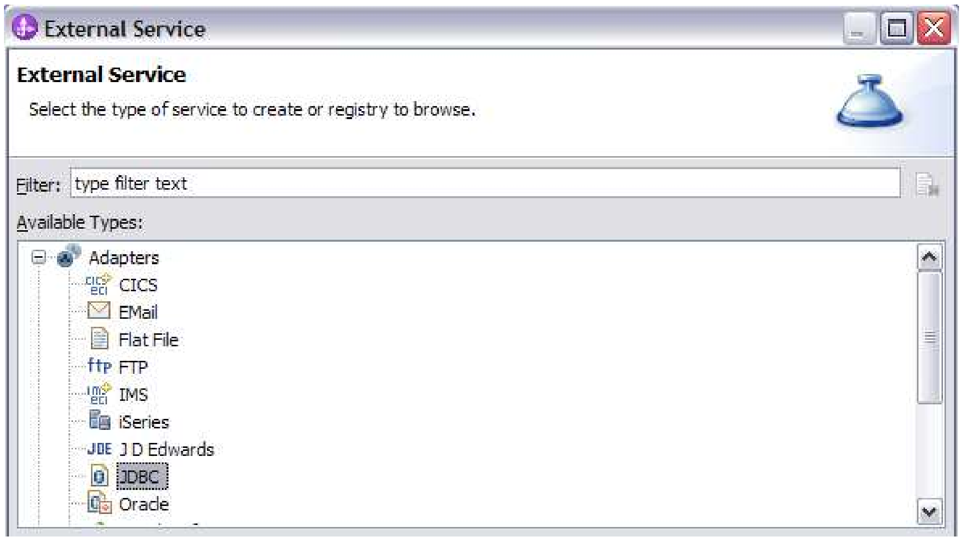
Issue the command **./startNetworkServer.sh**

Keep this telnet window open during the next few steps; this window will show when a connection has been successfully made from WebSphere Integration Developer to the database server.

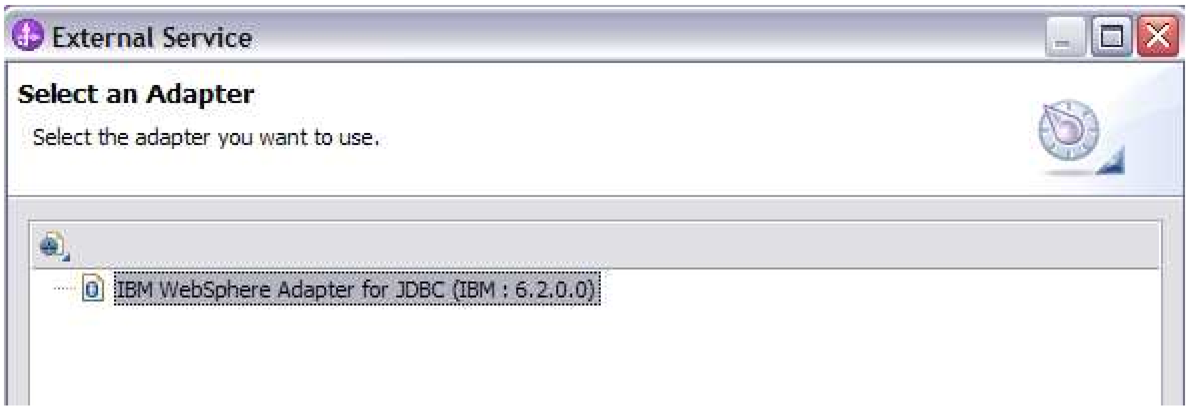
****NOTE** that there are security concerns with enabling the network server for Derby in this manner. Read the information in the db2j.properties file regarding these security concerns to determine if your installation will allow for the use of the network server for Derby. If not, you can continue doing the lab using the integrated test client in WebSphere Integration Developer instead of your remote test environment.

____ 1. From the main menu, select **File > New > External Service**. This opens an External Service wizard that helps you obtain a service which establishes connectivity with other systems. The wizard provides four connectivity options – Adapters, Java, Registers, and Messaging

__ a. Expand **Adapters,** select **JDBC,** and click **Next**



____ 2. Highlight **IBM WebSphere Adapter for JDBC (IBM: 6.2.0.0)** and click **Next**


External Service

Select an Adapter

Select the adapter you want to use.

IBM WebSphere Adapter for JDBC (IBM : 6.2.0.0)

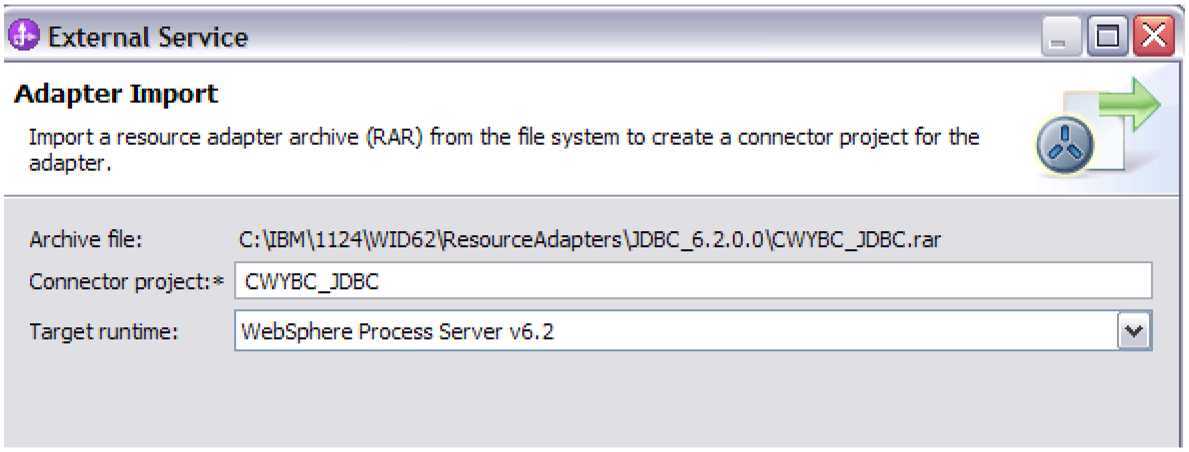
____ 3. Adapter Import screen:

In this step, you will import a connector resource adapter archive from the file system into your WebSphere Integration Developer workspace. The adapter RAR file already exists under **<JDBCADAPTER_HOME**>.

__ a. The default Connector file is selected which is shipped along with WebSphere Integration Developer

__ b. Accept the default name for Connector project, **CWYBC_JDBC**. You can change it to any other name, but for this lab, you can leave the default name.

__ c. For Target server, ensure that **WebSphere Process Server v6.2** is selected


External Service

Adapter Import

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

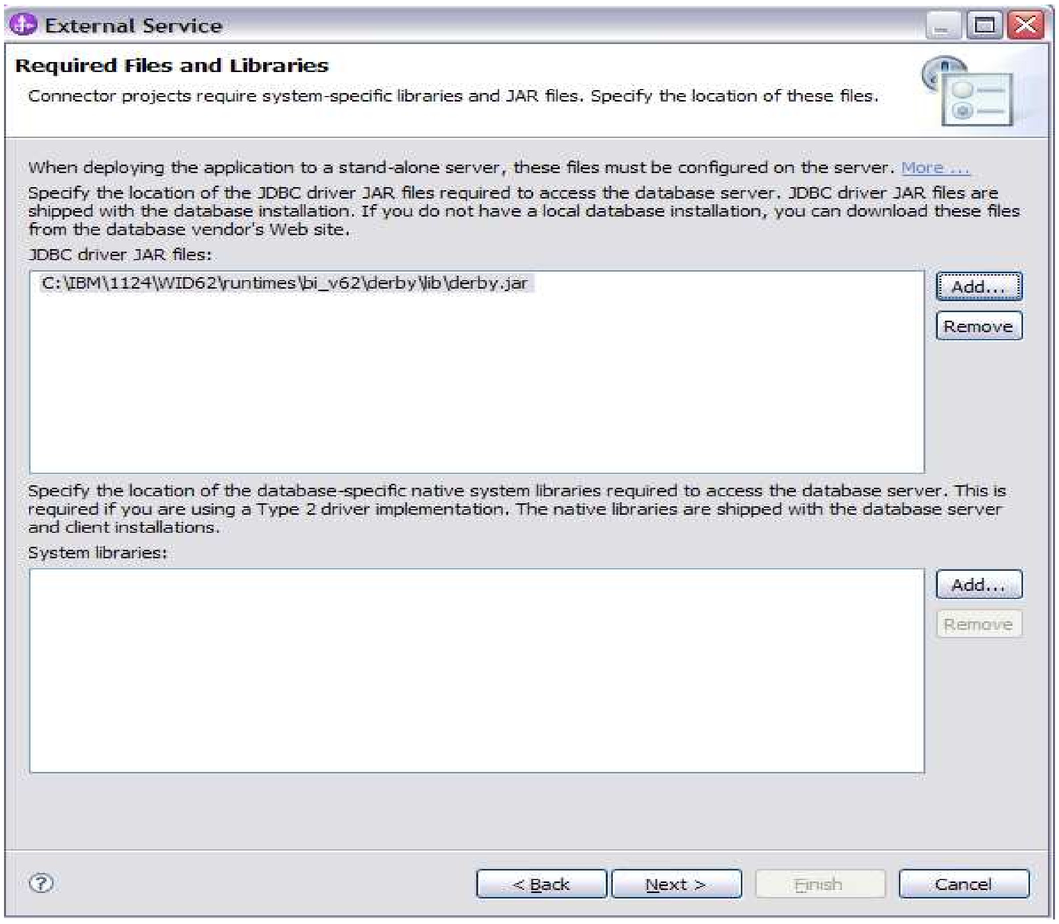
Archive file: C:\IBM\1124\WID62\ResourceAdapters\JDBC_6.2.0.0\CWYBC_JDBC.rar

Connector project:* CWYBC_JDBC

Target runtime: WebSphere Process Server v6.2

____ 4. Add any external dependencies your adapter has to the imported project. These are dependencies that the adapter can have on the JDBC applications (adapter–specific).

__ a. Click **Add** and browse to the location of c:\<WPS_HOME>\derby\lib and select the **derby.jar**. If you are using a remote test environment on z/OS, you will also need to select **db2jcc.jar** and **db2jcc.license.jar.** The db2jcc.jar file is sometimes located in directory \derby\lib\otherJars.



Required Files and Libraries

Connector projects require system-specific libraries and JAR files. Specify the location of these 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 database vendor's Web site.

JDBC driver JAR files:

C:\IBM\1124\WID62\runtimes\bi_v62\derby\lib\derby.jar

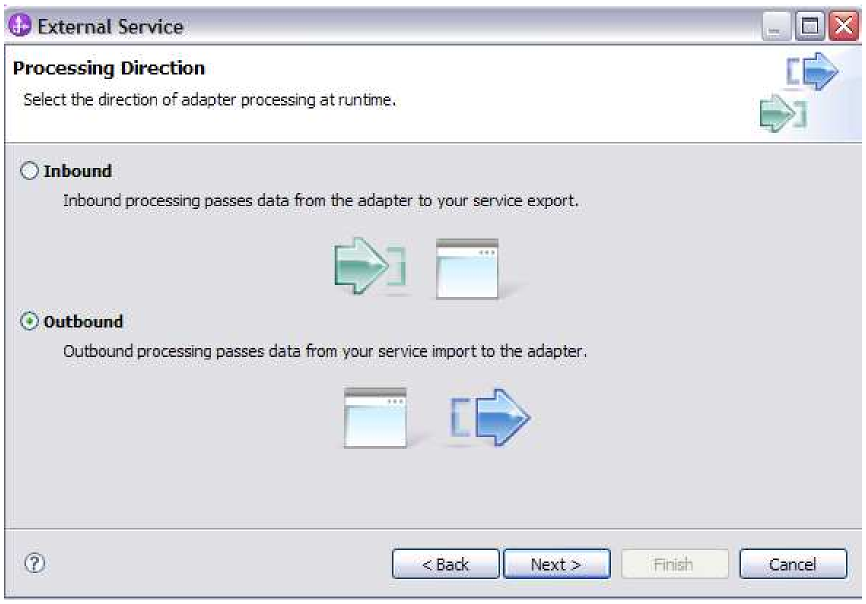
Specify the location of the database-specific native system libraries required to access the database server. This is required if you are using a Type 2 driver implementation. The native libraries are shipped with the database server and client installations.

System libraries:

__ b. Click **Next**

____ 5. Select the type of processing the adapter will perform at runtime

__ a. Select **Outbound** and click **Next**



Processing Direction

Select the direction of adapter processing at runtime.

[ ] Inbound

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

[✓] Outbound

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

____ 6. Complete the Connection Configuration for Discovery Agent Configuration panel to connect to the JDBCTEST database and discover the available services. To connect to the database, this information is necessary: username, password, database URL, and JDBC driver class. Check the driver manual for the appropriate values for driver URL and driver class.

__ a. From the left panel, expand **Generic JDBC** and select **1.0**

__ b. From the right panel, enter/select these following values

1) JDBC Driver type: **Other**

2) JDBC Driver classname: **org.apache.derby.jdbc.EmbeddedDriver**

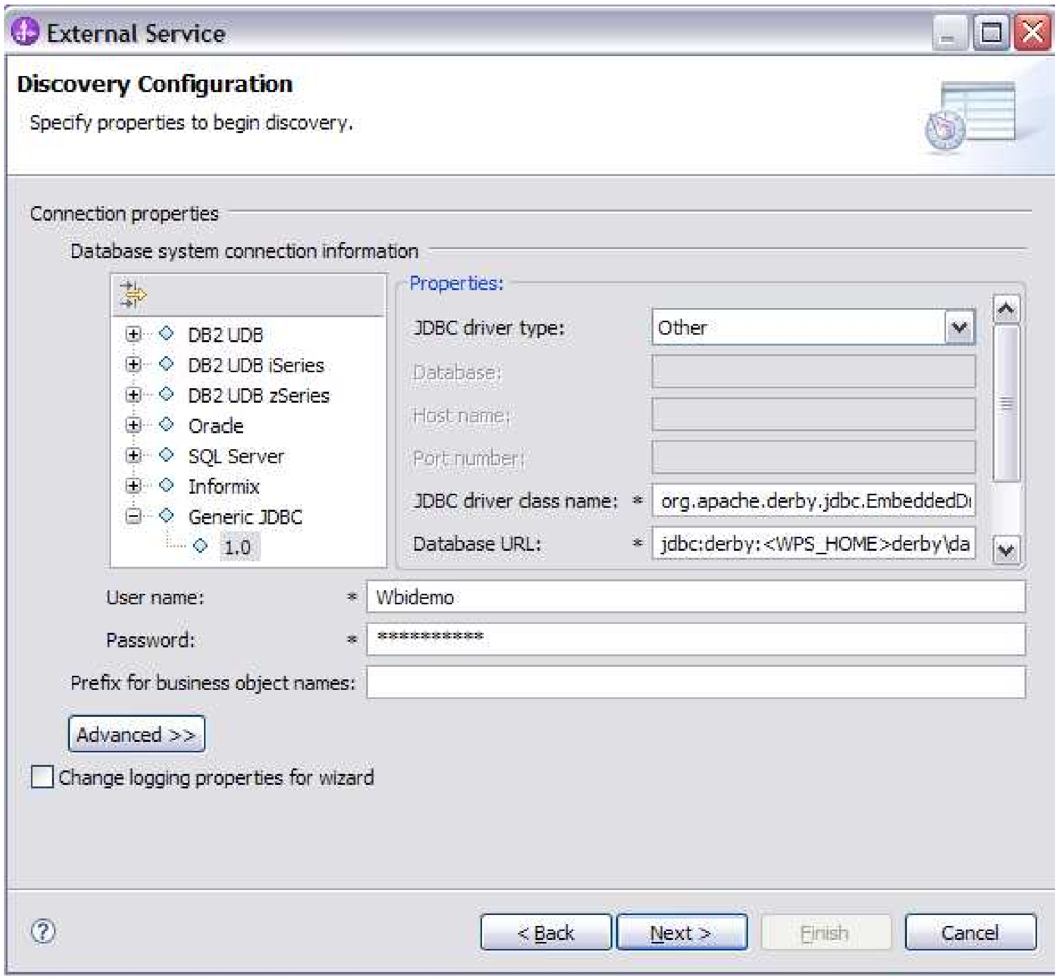
3) Database URL: **jdbc:derby:<WPS_HOME>derby\databases\JDBCTEST**

__ c. Enter valid user ID and password values, for example

Username: **Wbidemo**
Password: **Wbi15Demo1**

NOTE If you are using a remote test environment, use a valid user ID and password for the remote system and enter these values for DatabaseURL and JdbcDriverClass
DatabaseURL: jdbc:db2j:<WPS_HOME>\runtimes\bi_v62\derby\databases\JDBCTEST
JdbcDriverClass: com.ibm.db2j.jdbc.DB2jDriver

If using a remote test environment on z/OS use these values
DatabaseURL: jdbc:db2j:net://<HOST_NAME>:1527//<remote_derbydb_path>/JDBCTEST
JdbcDriverClass: COM.ibm.db2os390.sqlj.jdbc.DB2SQLJDriver



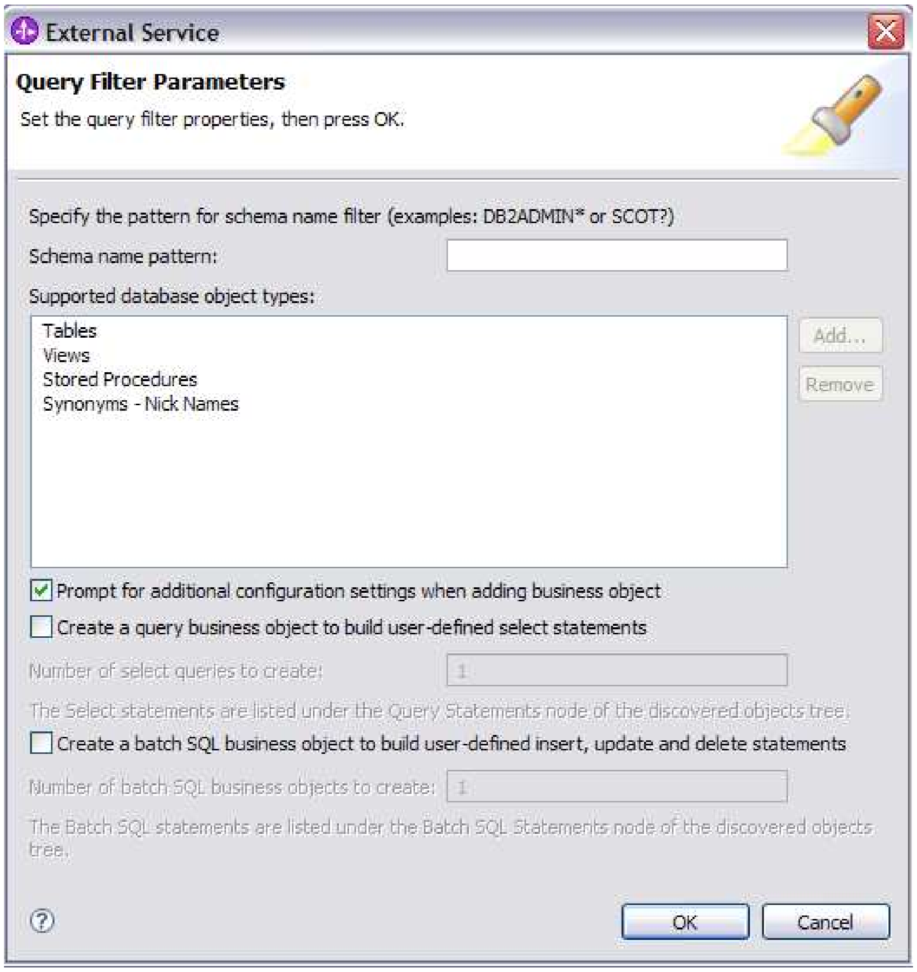
JDBC driver type: Other
Database:
Host name:
Port number:
JDBC driver class name: * org.apache.derby.jdbc.EmbeddedDr
Database URL: * jdbc:derby:<WPS_HOME>derby\da
User name: * Wbidemo
Password: * ***********
Prefix for business object names:
☐ Change logging properties for wizard

__c. Click **Next**

____ 7. Complete the **Find and Discover Enterprise Services** panel

__ a. Click the **Edit Query** button

__ b. In the Query Filter Properties window that opens up, check the **Prompt for additional configuration settings when adding business object** check box, click **OK.**



External Service

Query Filter Parameters

Set the query filter properties, then press OK.

Specify the pattern for schema name filter (examples: DB2ADMIN* or SCOT?)

Schema name pattern:

Supported database object types:

Tables
Views
Stored Procedures
Synonyms - Nick Names

☑ 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

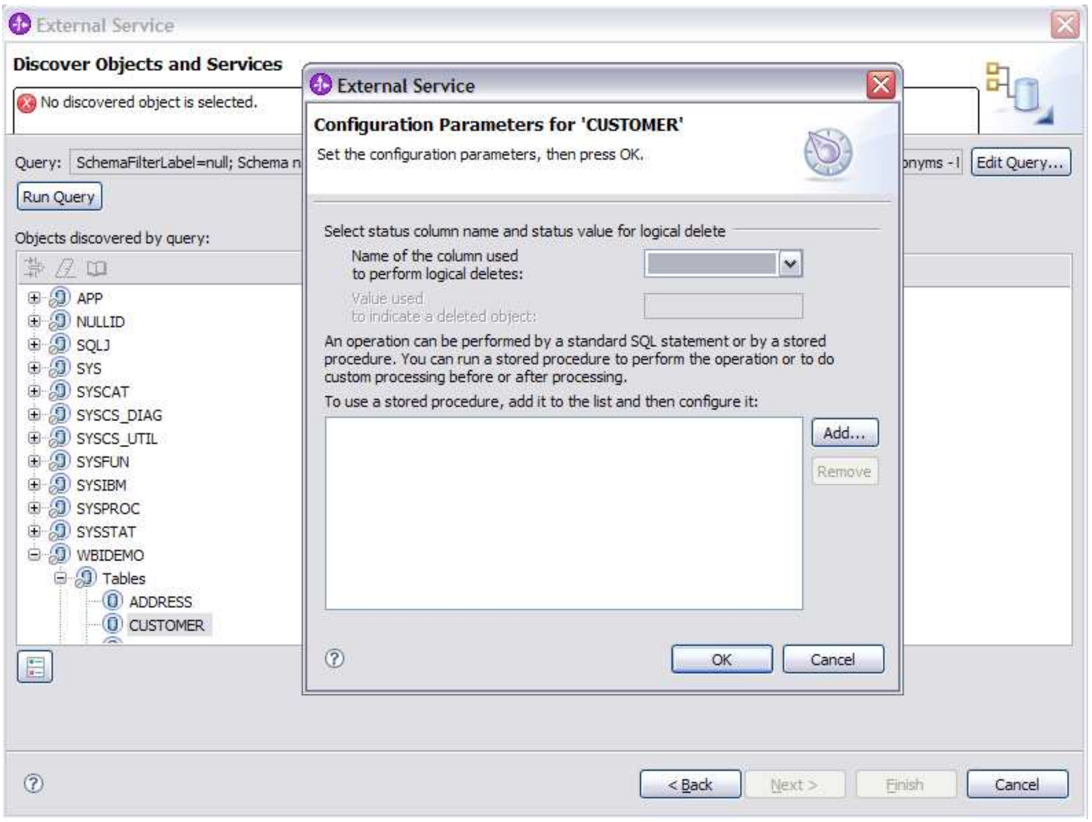
Number of batch SQL business objects to create: 1

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

__ c. Click **Run Query** button. A connection is made to the Derby JDBCTEST database and a selection of Meta data objects is presented in a tree-like structure.

__ d. Expand the schema named **WBIDEMO**, expand **Tables**, highlight **CUSTOMER**, click the button.

__ e. It opens up a window for setting the Configuration Parameters for the **CUSTOMER** table. All the controls, like Choose Parent, for Hierarchy BO Generation are not available as no object is imported yet. Click **OK** button to add **CUSTOMER** table.


External Service

Discover Objects and Services

No discovered object is selected.

Query: SchemaFilterLabel=null; Schema n ... onyms - I

Objects discovered by query:

APP
NULLID
SQLJ
SYS
SYSCAT
SYSCS_DIAG
SYSCS_UTIL
SYSFUN
SYSIBM
SYSPROC
SYSSTAT
WBIDEMO
Tables
ADDRESS
CUSTOMER

External Service

Configuration Parameters for 'CUSTOMER'

Set the configuration parameters, then press OK.

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 standard SQL statement or 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:

__ **f.** Select **Address** table and click the button.

__ **g.** It opens up a window for setting the Configuration Parameters for the **ADDRESS** table where you can select Parent table and map parent-child attributes.

__ **h.** In the Configuration Parameters screen for **Address**, select these following values

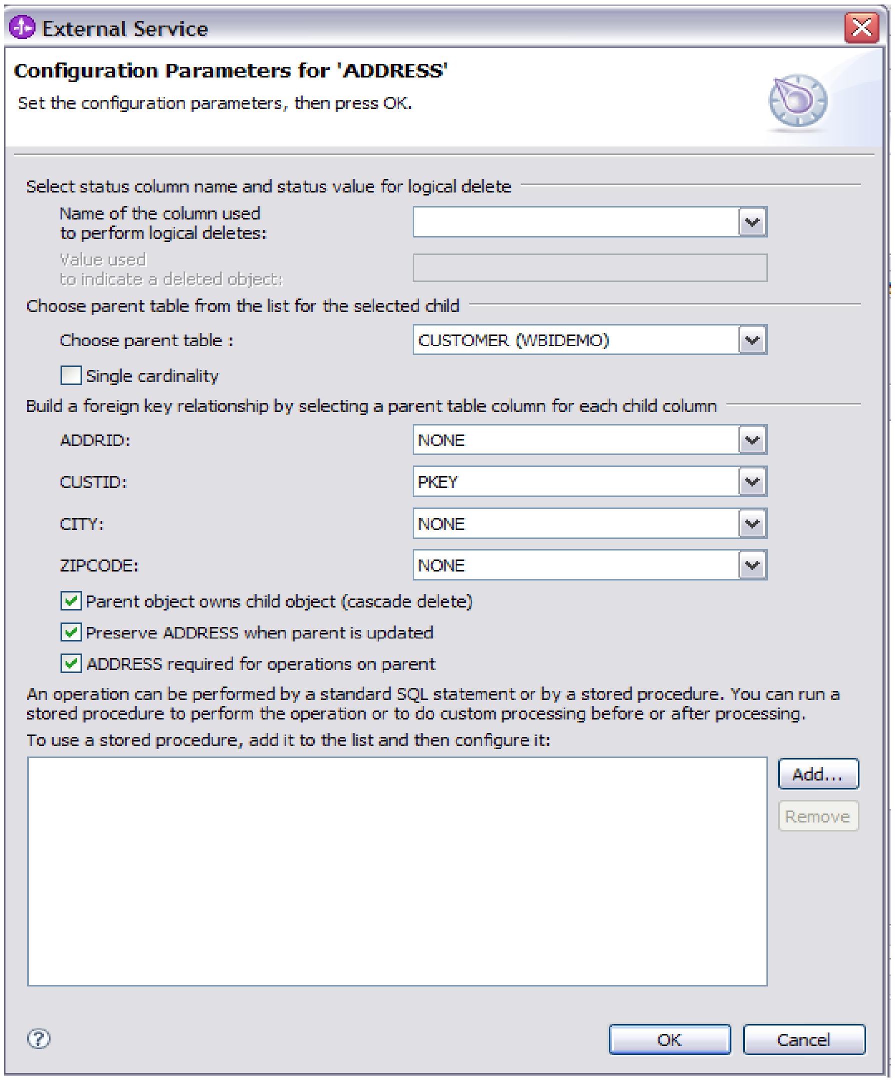
1) Choose CUSTOMER (WBIDEMO) as the Parent table for ADDRESS from the list box.

2) This particular scenario creates business object with multiple cardinality between CUSTOMER and ADDRESS, therefore do not check Single Cardinality check box.

3) To Map Parent Child Attributes, choose PKEY of CUSTOMER for CUSTID from the drop-down list.

4) Check **Parent object owns child object, Preserve ADDRESS when parent is updated, ADDRESS required for operations on parent** check boxes to specify the respective ASIs.

5) Click **Ok**


External Service

Configuration Parameters for 'ADDRESS'

Set the configuration parameters, then press OK.

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 (WBIDEMO)

☐ 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)

☑ Preserve ADDRESS when parent is updated

☑ ADDRESS required for operations on parent

An operation can be performed by a standard SQL statement or 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:

__ **i.** Click **Next**

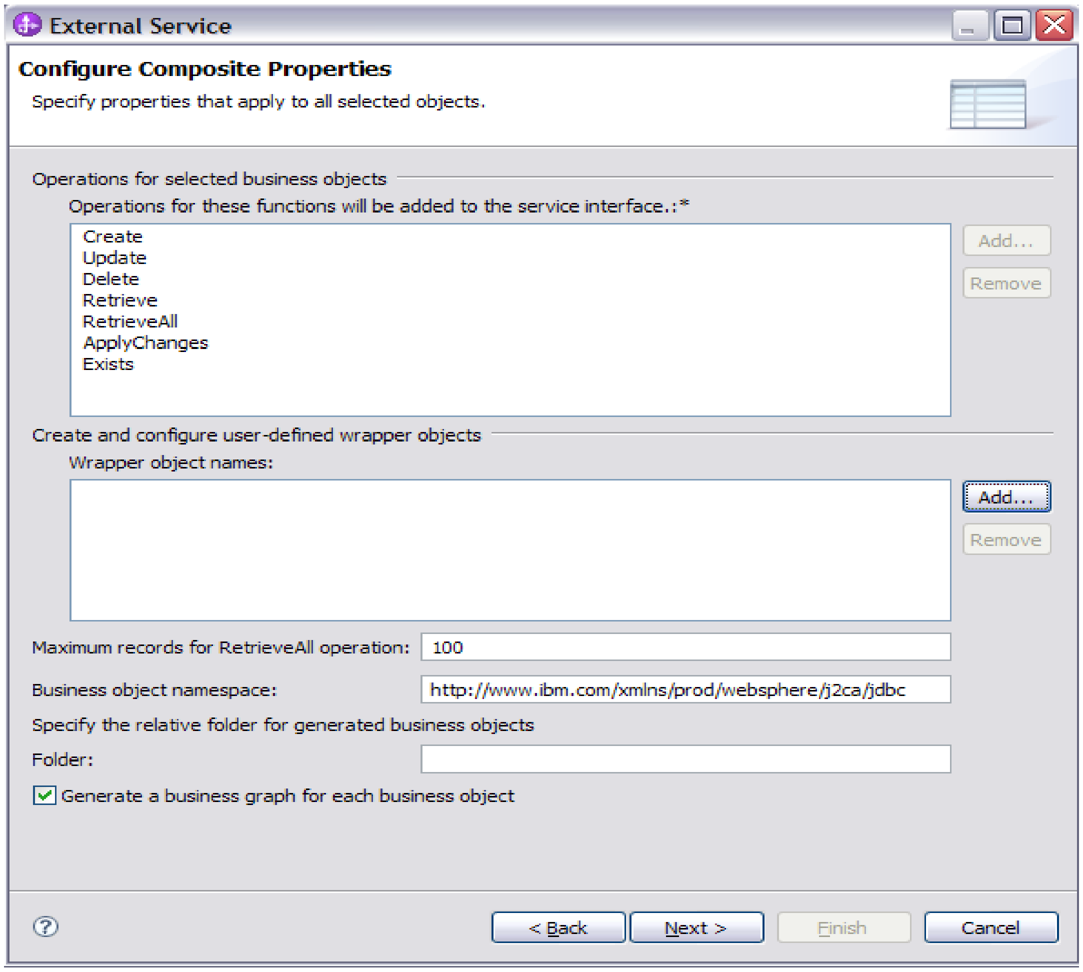
____ 8. Complete the **Configure Objects** panel

__ a. A new operation Exists has been added in addition to the current list of supported operation.

__ b. No wrapper object is required at this time.

__ c. Leave the default value for Namespace, leave blank for **BOLocation.** Note the Operations available.

__ d. Note that Business Graph is optional - It will determine if you want to generate business graph or not. If the property is checked then business graphs is generated. By default, the property is checked.



Configure Composite Properties

Specify properties that apply to all selected objects.

Operations for selected business objects

Operations for these functions will be added to the service interface.:*

Create
Update
Delete
Retrieve
RetrieveAll
ApplyChanges
Exists

Create and configure user-defined wrapper objects

Wrapper object names:

Maximum records for RetrieveAll operation: 100

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

__ e. Click **Next**

____ 9. Complete Service Generation and Deployment Configuration

__ **a.** Leave the default option "**With module for us by single application**" in this exercise.

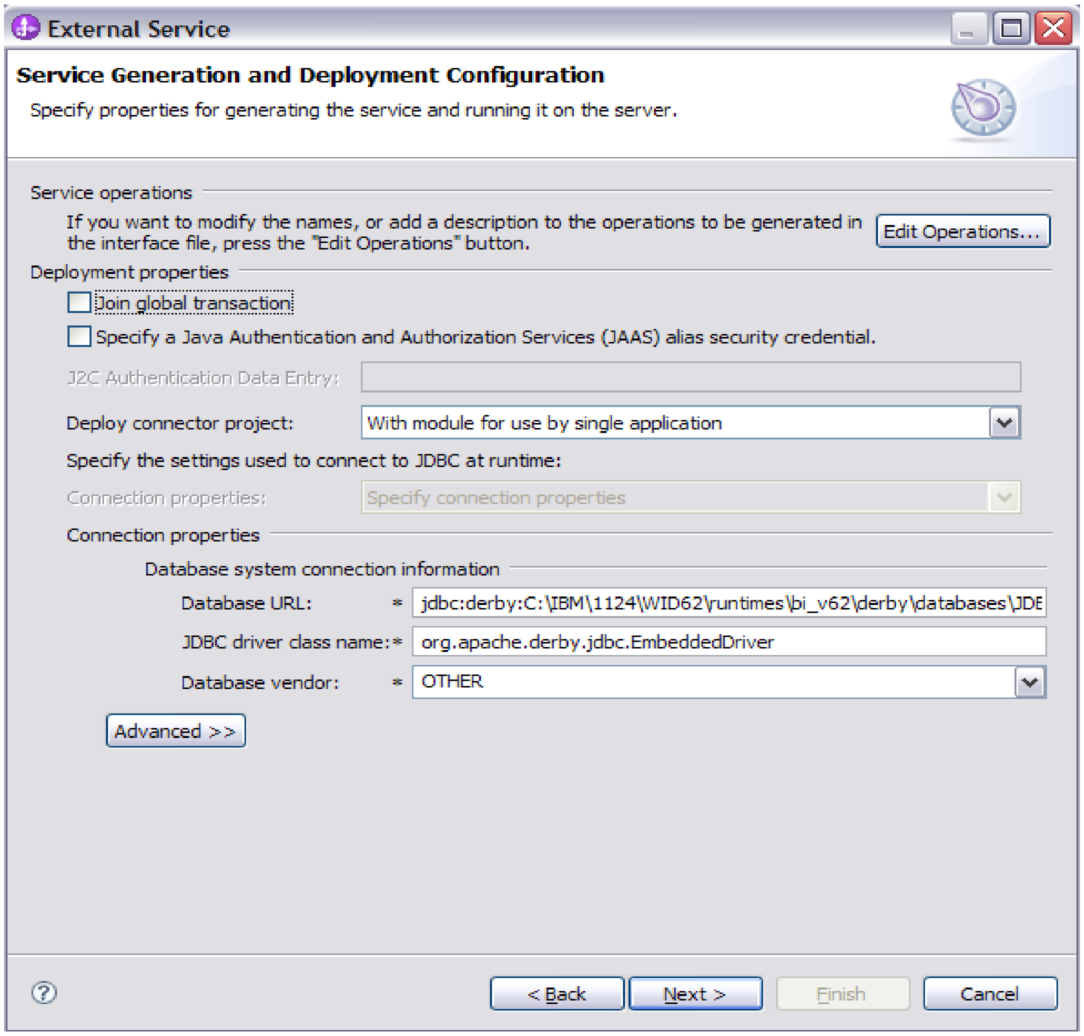
Note: All IBM WebSphere Adapters are now supported deploying RAR separately. To do so, select "With module for use by single application" from the Deploy connector project drop down list. If you want the adapter to deployed with the module which is packaged as an Enterprise Archive file (EAR file), then the "On Server for use by multiple applications" option should be selected.

____ 10. Clear the check box for **Join Global Transaction**. By clearing it, you are going to run this component in a new global transaction.

____ 11. Specify the **JAAS Alias security credential** as JNDI data source name is optional for Outbound service. For this lab, you are going to define the data source JNDI name, so clear this JAAS option.

__ a. Under **Connection Properties**

1) Database Vendor is pre-populated with value **Other** (If you were using DB2, Oracle, or MSSQLServer, or Informix you see those values instead as specific adapter processing is available with those specific databases.)



External Service

Service Generation and Deployment Configuration

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

Service operations

If you want to modify the names, or add a description to the operations to be generated in the interface file, press the "Edit Operations" button.

Deployment properties

- [ ] Join global transaction
- [ ] Specify a Java Authentication and Authorization Services (JAAS) alias security credential.

J2C Authentication Data Entry:

Deploy connector project: With module for use by single application

Specify the settings used to connect to JDBC at runtime:

Connection properties: Specify connection properties

Connection properties

Database system connection information

Database URL: * jdbc:derby:C:\IBM\1124\WID62\runtimes\bi_v62\derby\databases\JDE

JDBC driver class name: * org.apache.derby.jdbc.EmbeddedDriver

Database vendor: * OTHER

__ b. Click **Advanced** for additional properties options

__ c. **Optional -** expand **Advanced connection configuration** and enter this following value

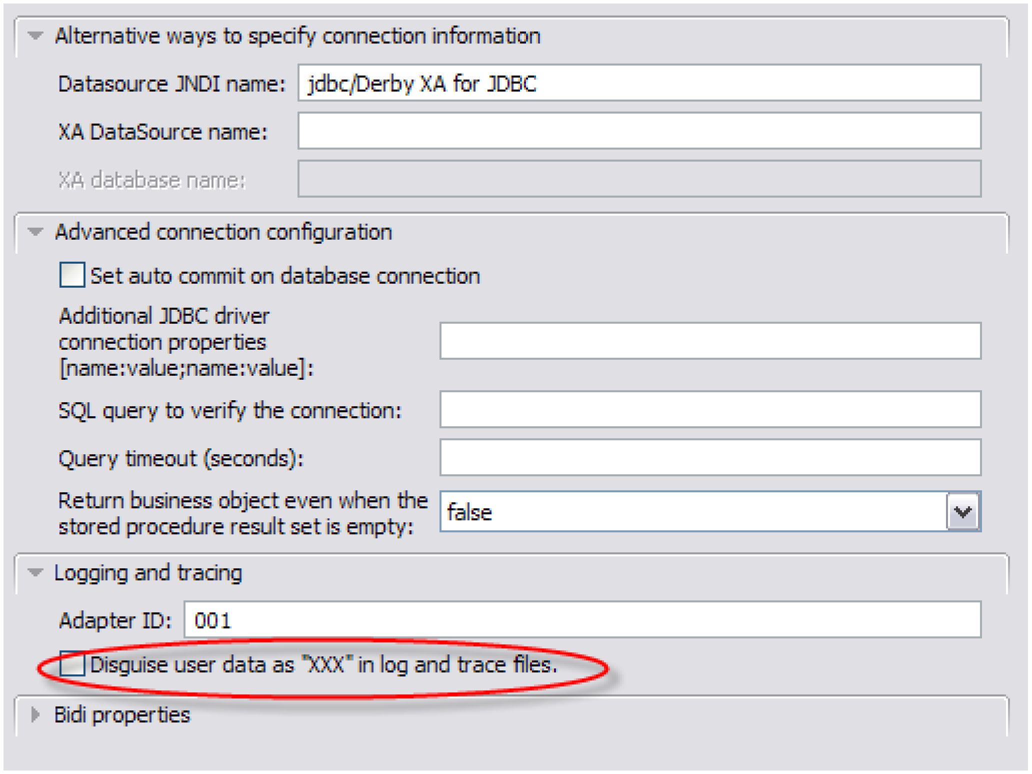
1) Data Source JNDI Name: **jdbc/Derby XA for JDBC**

Note: If you are using a remote test environment, the values for **databaseURL** and **JdbcDriverClass** should match the values provided earlier during the External Service wizard, not what is pictured below.

__ d. **Optional** for logging and tracing

1) Enter any numeric value for **Adapter ID**. This property identifies the adapter instance in log and trace files and for PMI events. The adapter ID is used with an adapter-specific identifier, MyAdapterRA, to form the component name used by Log and Trace Analyzer.

2) If you set the property to disguise user data as "XXX', the adapter will replace user data with a string of X's when writing to log and trace files. For inbound processing, this property is retrieved from the resource adapter properties. For outbound processing, it is retrieved from the managed connection factory properties.


Alternative ways to specify connection information

Datasource JNDI name: jdbc/Derby XA for JDBC

XA DataSource name:

XA database name:

Advanced connection configuration

☐ Set auto commit on database connection

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

SQL query to verify the connection:

Query timeout (seconds):

Return business object even when the stored procedure result set is empty: false

Logging and tracing

Adapter ID: 001

☐ Disguise user data as "XXX" in log and trace files.

Bidi properties

__ e. Click **Next**

____ 12. Complete Publishing Properties panel

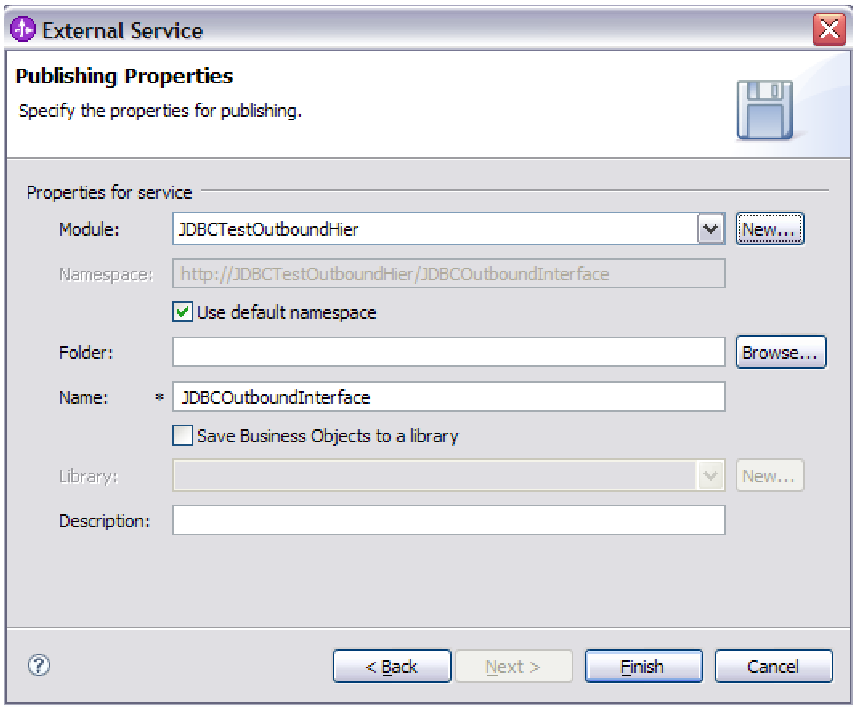
__ a. A Business Integration Module has not yet been created. You can define the module here also.

1) Select the **New** button Select the radio button "**Create a module project**" from new **Integration Project** pop up window.

2) Enter in the name **JDBCTestOutboundHier** for the Module Name

3) Accept the defaults and click **Finish**

__ b. Leave the default **JDBCOutboundInterface** for the Name


External Service

Publishing Properties

Specify the properties for publishing.

Properties for service

Module: JDBCTestOutboundHier

Namespace: http://JDBCTestOutboundHier/JDBCOutboundInterface

[✓] Use default namespace

Folder:

Name: * JDBCOutboundInterface

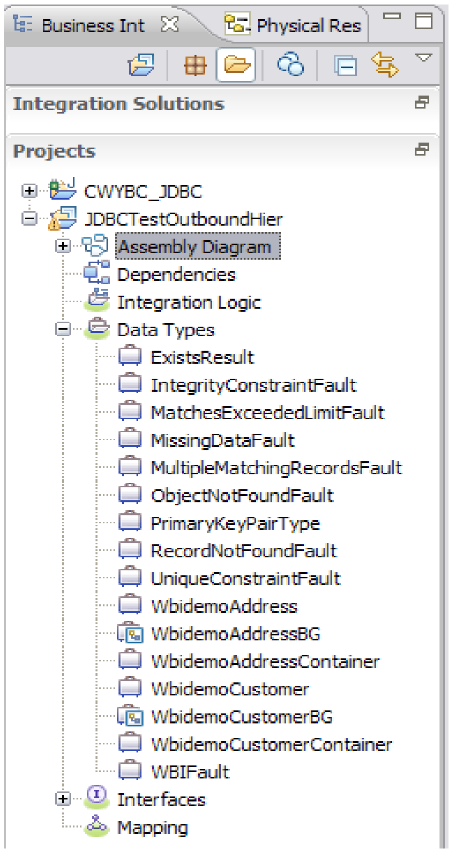
[ ] Save Business Objects to a library

Library:

Description:

__ c. Click **Finish**

____ 13. Verify the result as given below

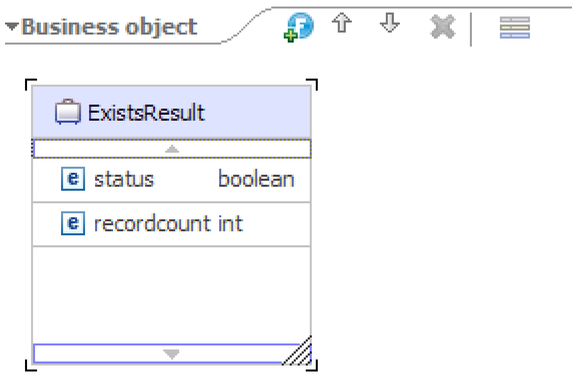


- CWYBC_JDBC
- JDBCTestOutboundHier
  - Assembly Diagram
  - Dependencies
  - Integration Logic
  - Data Types
    - ExistsResult
    - IntegrityConstraintFault
    - MatchesExceededLimitFault
    - MissingDataFault
    - MultipleMatchingRecordsFault
    - ObjectNotFoundFault
    - PrimaryKeyPairType
    - RecordNotFoundFault
    - UniqueConstraintFault
    - WbidemoAddress
    - WbidemoAddressBG
    - WbidemoAddressContainer
    - WbidemoCustomer
    - WbidemoCustomerBG
    - WbidemoCustomerContainer
    - WBIFault
  - Interfaces
  - Mapping

__ a. Double click new **ExistsResult** BO listed under Data Types

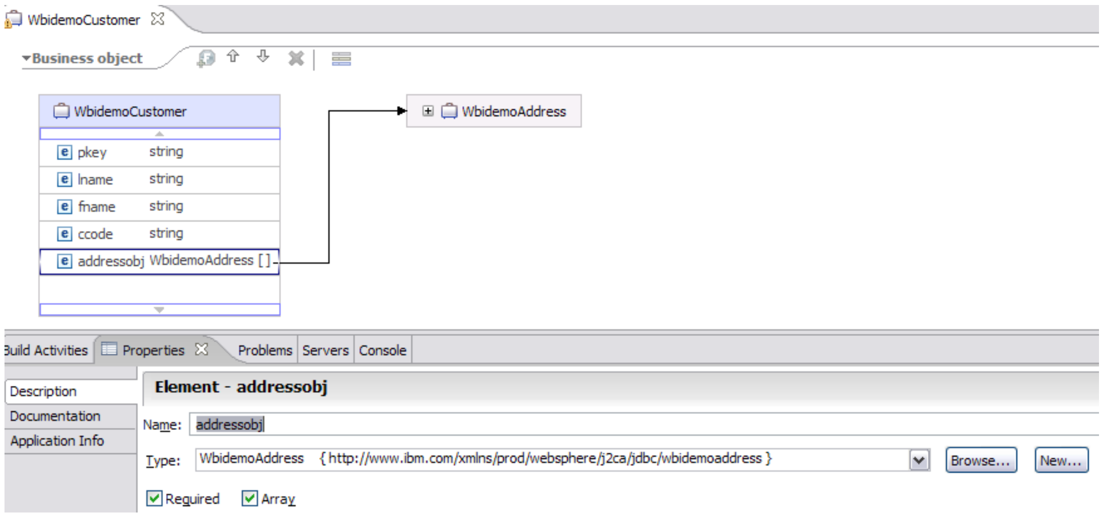
1) Status returns a Boolean value if the record is found in the database

2) Recordcount returns the records found in the database for a specified query.



____ 14. Double click the **WbidemoCustomer** business object listed under **Data Types**

__ a. Note that multiple cardinality child addressobj has been added. Select this child attribute and in the properties tab, click Description and verify that **Array** and **Required** check boxes are selected.



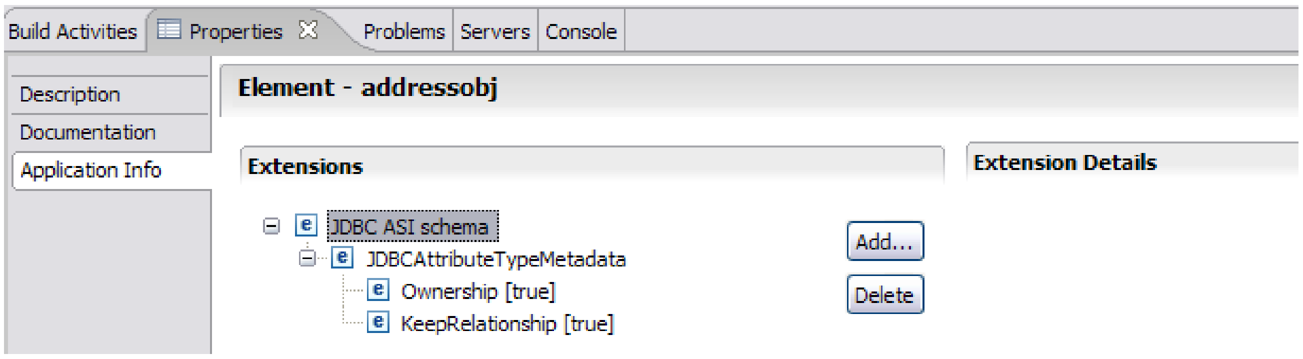
Element - addressobj

Name: addressobj

Type: WbidemoAddress {http://www.ibm.com/xmlns/prod/websphere/j2ca/jdbc/wbidemoaddress}

[✓] Required [✓] Array

__ b. Click **Application Info** in the **Properties** tab and verify that values against **jdbcasi:Ownership** and **jdbcasi:KeepRelationship** are true as you selected Ownership and Keep Relationship check boxes in External Service.

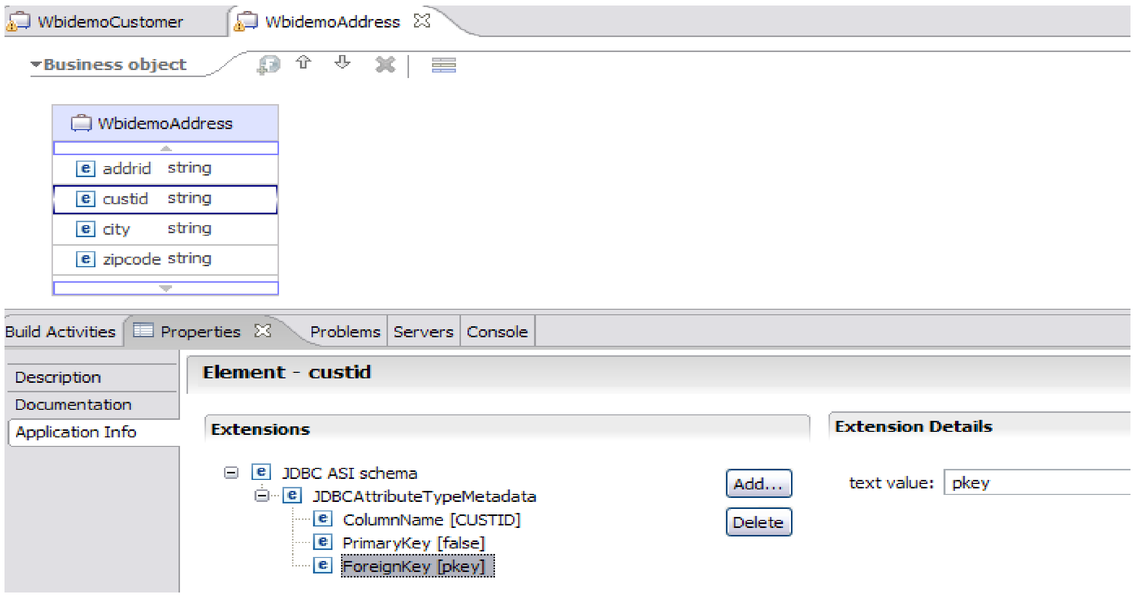


Element - addressobj

JDBC ASI schema
JDBCAttributeTypeMetadata
Ownership [true]
KeepRelationship [true]

____ 15. Double click the **WbidemoAddress** business object listed under **Data Types**

__ a. Select attribute **custid** attribute and in the **Properties** tab, click **Application Info**. Verify that **pkey** is specified against **jdbcasi:ForeignKey**.



WbidemoAddress

addrid	string
custid	string
city	string
zipcode	string

Element - custid

JDBC ASI schema
JDBCAttributeTypeMetadata
ColumnName [CUSTID]
PrimaryKey [false]
ForeignKey [pkey]

text value: pkey

__ b. Save your work by selecting **File -> Save** from the top menu, or using the shortcut key sequence **Ctrl + S.** Close the file.

____ 16. Release the connection to the Derby database by using **Switch Workspace**

Note: Switch Workspace is a way to release the existing connection to the Derby database from the External Service process. In the next part, you will start the WebSphere Process Server and it will need a connection to the database to create and retrieve records. This step is necessary only because you are using the Derby embedded driver in this exercise which supports a connection from a single JVM.

__ a. From the top menu bar, select **File > Switch Workspace** and select the same workspace from which you have been working.

Part 4: Create J2C authentication alias and configure data sources

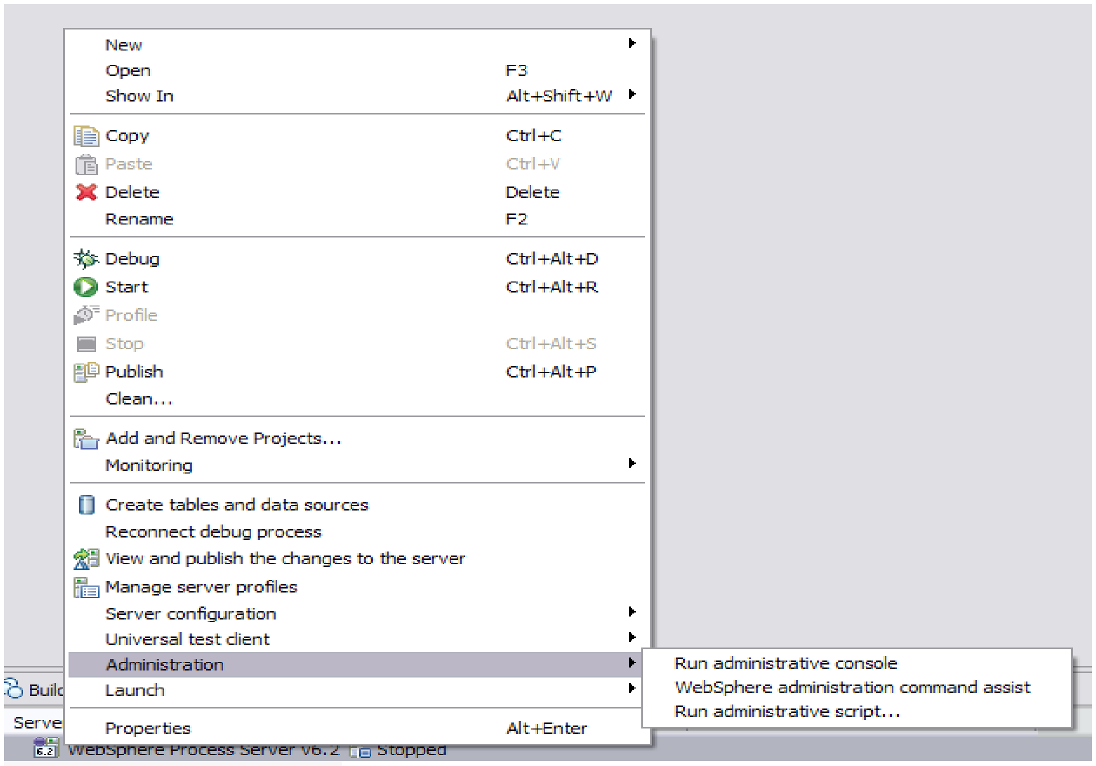
In this part you will create a J2C Authentication Alias which is required for connection to the database. You will also create the JNDI data source name that is used by the Adapter to configure itself to the endpoint. You will then use the WebSphere Test Environment and Component Test to test the SCA application by retrieving several records from the CUSTOMER table in the JDBCTEST database.

____ 1. Switch to Servers view by selecting Windows → Show View → Servers.

____ 2. Set the authentication alias from the administrative console

__ a. In the **Servers** tab in the lower-right corner pane, right click the **Server** and then select **Start**

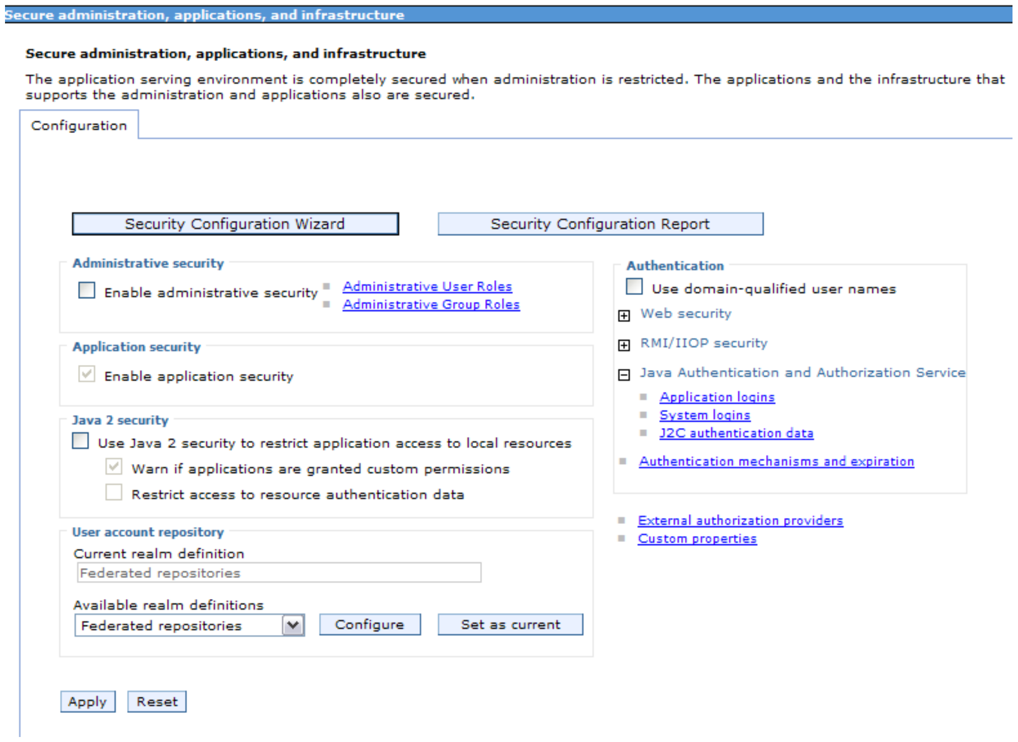
__ b. When the server status is **Started**, right click the **Servers**, and then select **Run administrative console.**



__ c. Log in to the administrative console by clicking the “Log in” button

__ d. Click **Security → Secure administration, application, and infrastructure**

__ e. On the right, expand **Java Authentication and Authorization Service** under the Authentication heading.



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 that 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: Federated repositories

Available realm definitions: Federated repositories | Configure | Set as current

Authentication

☐ Use domain-qualified user names

- Web security
- RMI/IIOP security
- Java Authentication and Authorization Service
  - Application logins
  - System logins
  - J2C authentication data
- Authentication mechanisms and expiration

- External authorization providers
- Custom properties

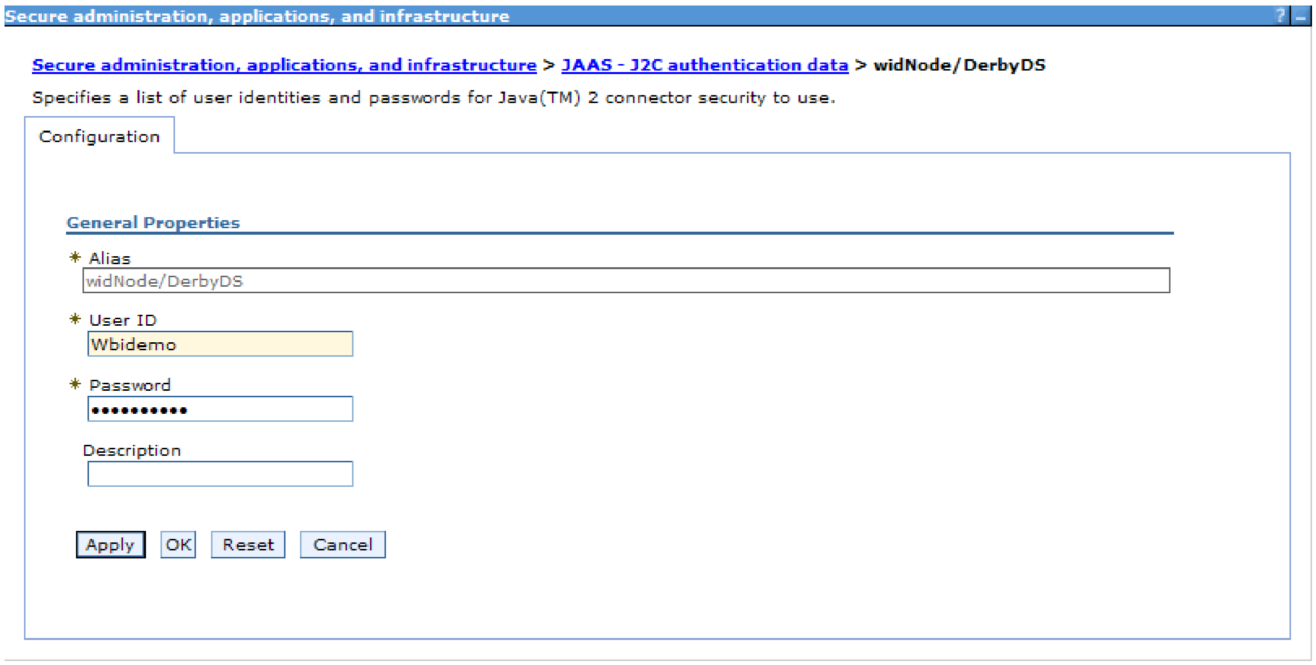
Apply | Reset

__ f. Click **J2C authentication alias**. It gives the list of existing aliases.

1) Click **New** and enter these following values

a) User ID: **Wbidemo**

b) Password: **Wbi15Demo1**



Secure administration, applications, and infrastructure

Secure administration, applications, and infrastructure > JAAS - J2C authentication data > widNode/DerbyDS

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

Configuration

General Properties

* Alias: widNode/DerbyDS

* User ID: Wbidemo

* Password: ••••••••••

Description:

Apply | OK | Reset | Cancel

__ g. Click **Ok** and save the changes

____ 3. Create a JDBC Provider to which is used to create data source

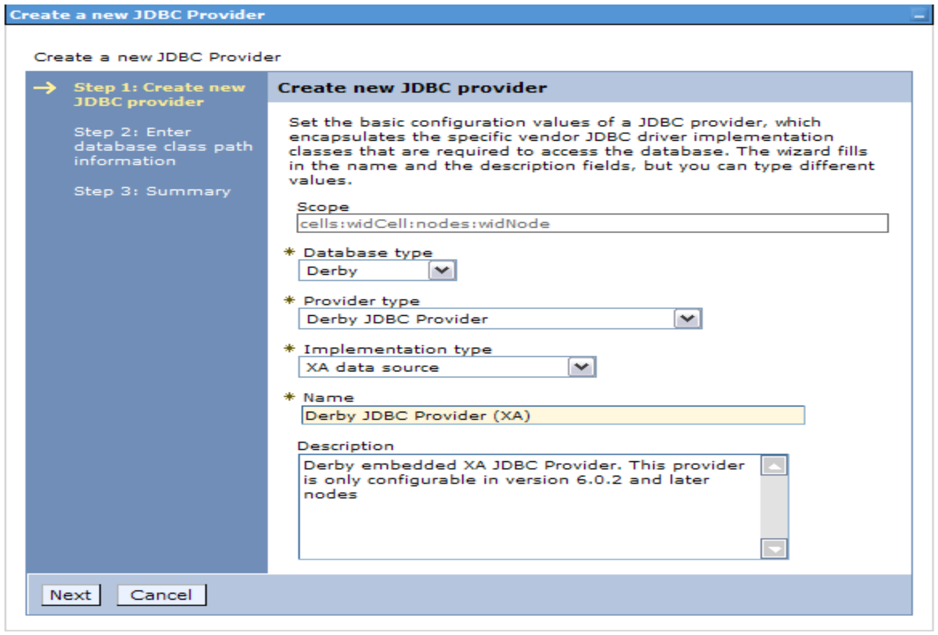
__ a. From administrative console, click **Resources → JDBC → JDBC Providers**

__ b. On the right, click **New** and choose these following values

1) Database type: **Derby**

2) Provider type: **Derby JDBC Provider**

3) Implementation type: **XA data source**


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
Next
Cancel

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:widCell:nodes:widNode

* Database type
Derby

* Provider type
Derby JDBC Provider

* Implementation type
XA data source

* Name
Derby JDBC Provider (XA)

Description
Derby embedded XA JDBC Provider. This provider is only configurable in version 6.0.2 and later nodes

__ c. Click **Next** and review the summary of changes.

__ d. Click **Finish** and save the changes.

____ 4. Create JDBC data source

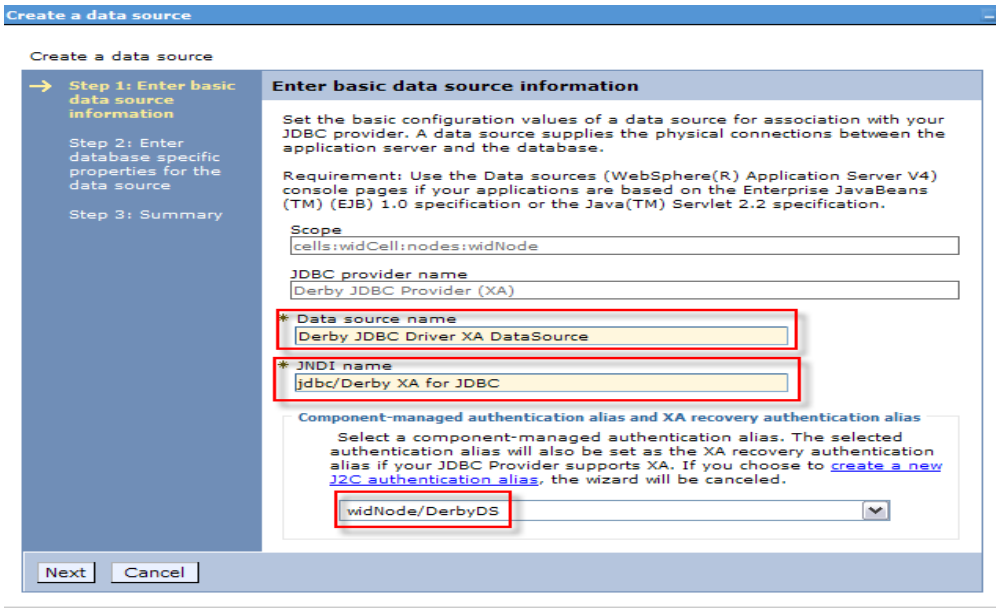
__ a. On the right, click **Data Sources** under **Additional Properties** heading

__ b. Click **New** to set a data source and enter these following values

1) Data source name: **Derby JDBC Driver XA DataSource**

2) JNDI name: **jdbc/Derby XA for JDBC**

__ c. Set the Component-managed authentication alias to the one created in the earlier section.


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 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:widCell:nodes:widNode

JDBC provider name: Derby JDBC Provider (XA)

* Data source name: Derby JDBC Driver XA DataSource

* JNDI name: jdbc/Derby XA for JDBC

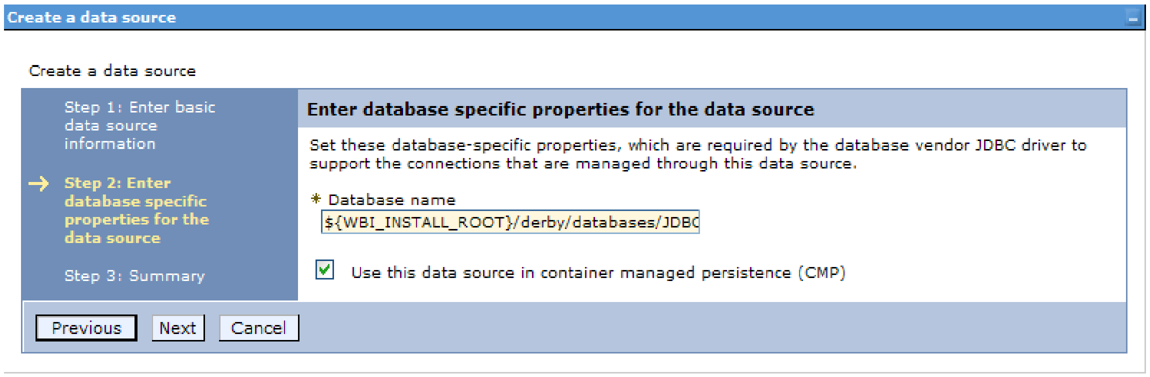
Component-managed authentication alias and XA recovery authentication alias

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.

widNode/DerbyDS

__ d. Click **Next.**

__ e. Enter the full path to the location of database name **JDBCTEST.**


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: ${WBI_INSTALL_ROOT}/derby/databases/JDBC

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

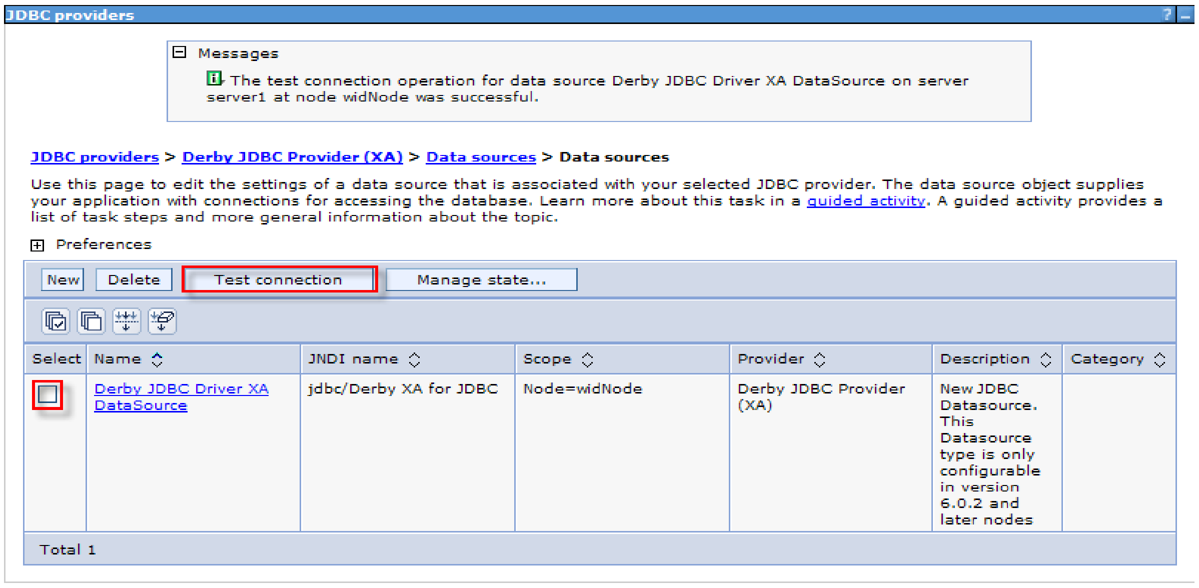
__ f. Click **Next** and view the summary

__ g. Click **Finish** and save the changes.

____ 5. Test the JDBC Data Source connection

__ a. Check the box next to **Derby JDBC Driver XA DataSource** and click on **Test connection** from the top of the screen.

__ b. You should see this success message on the top of the screen



JDBC providers

Messages

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

JDBC providers > Derby JDBC Provider (XA) > Data sources > 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
☐	Derby JDBC Driver XA DataSource	jdbc/Derby XA for JDBC	Node=widNode	Derby JDBC Provider (XA)	New JDBC Datasource. This Datasource type is only configurable in version 6.0.2 and later nodes	

Total 1

____ 6. Release the connection to the Derby database by using **Switch Workspace**

__ a. From the top menu bar, select **File > Switch Workspace** and select the same workspace from which you have been working.

Part 5: Test the application using the WebSphere test environment and component test

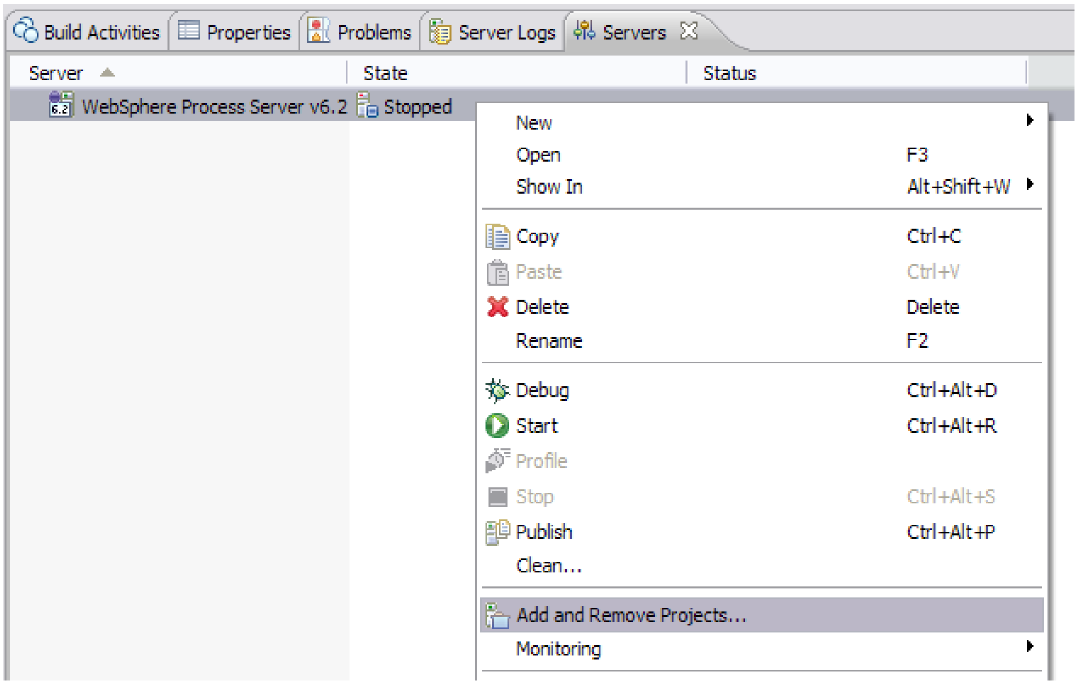
This lab is using the Derby embedded database driver which allows a connection from only a single JVM at a time. ij.bat, the External Service process, and a running Server configured with an application that connects to the database, each are an active connection. This means you can have only one of these active connections to the database at a time. For example, in this part, you will start the server and run several tests creating and retrieving records from the database. You are not able to use ij.bat to view the database, while the server is active. You need to first stop the server.

____ 1. In the **Servers** tab in the lower-right corner pane, right click the **Server** and then select **Start**

____ 2. Add the project to the server for the WebSphere Test Environment.

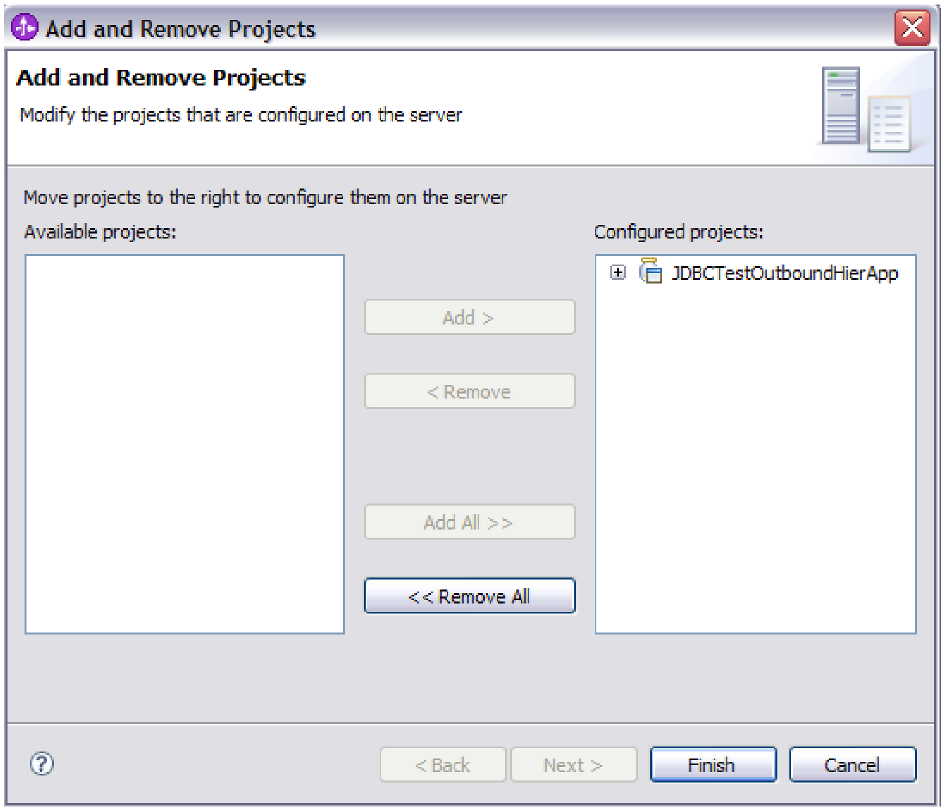
__ a. Start the server by right clicking Server and selecting **Start.**

__ b. Right click the server in the server view and select **Add and remove projects** …



__ c. In the Add and Remove Projects dialog, select the **JDBCTestOutboundHierApp** project from the Available projects panel.

__ d. Click **Add >** to add it to 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:

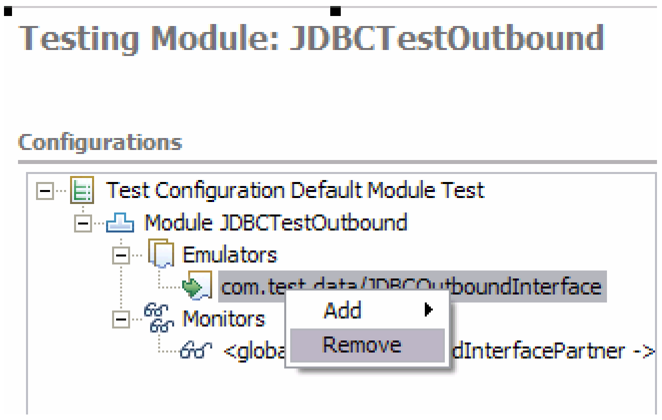
Configured projects:

JDBCTestOutboundHierApp

____ *3.* Use the Test Component to test the application. The create scenario is to create a record in the database.

__ a. Right click the **JDBCTestOutboundHier module** and select **Test → Test Component**

__ b. In JDBCTestOutboundHier_Test, switch to the **Configurations** tab, if you see **JDBCOutboundInterface** underneath **Emulators,** right click, and remove it. You might not see anything under Emulators. You want to connect to and drive the real JDBCTEST database and not have Test Component emulate for you.



Testing Module: JDBCTestOutbound
Configurations
Test Configuration Default Module Test
Module JDBCTestOutbound
Emulators
com.test.data/JDBCOutboundInterface
Monitors
<globa
dInterfacePartner ->
Add
Remove

__ c. Switch back to the **Events** tab

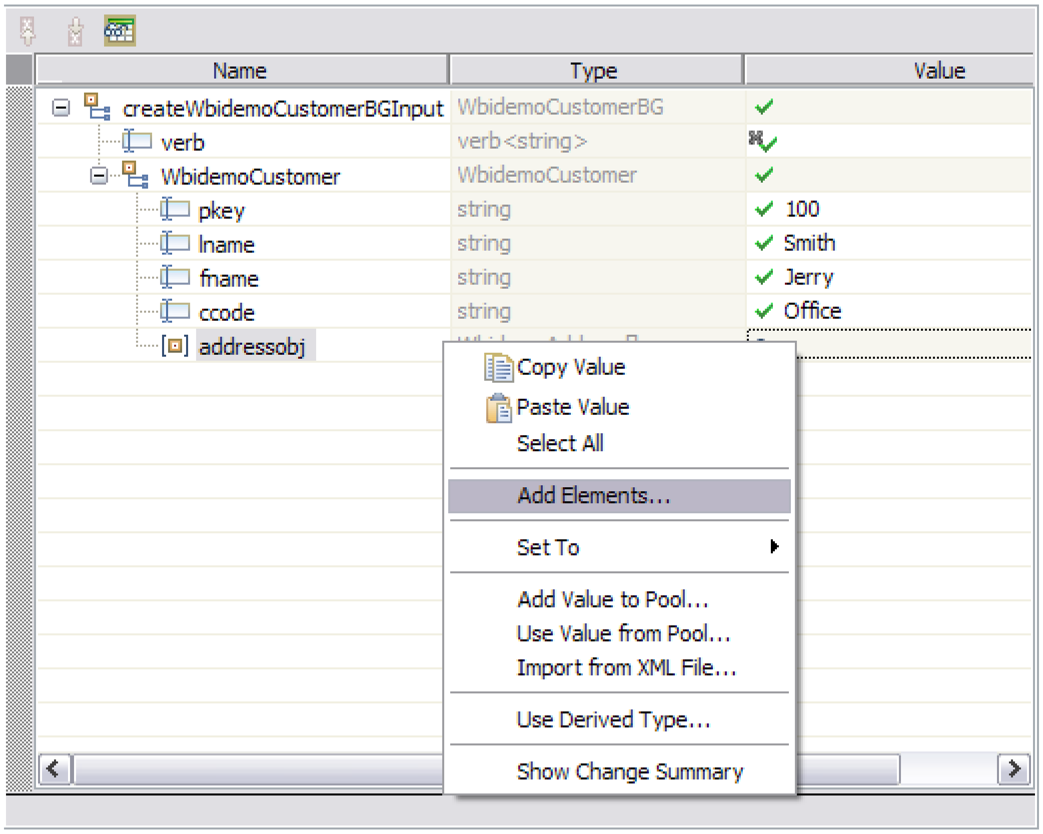
1) Under **Detailed Properties,** choose createWbidemoCustomerBG as the Operation.

2) Under **Initial request parameters**, enter these following values

a) **Verb** is depreciated so it is no longer required to enter.

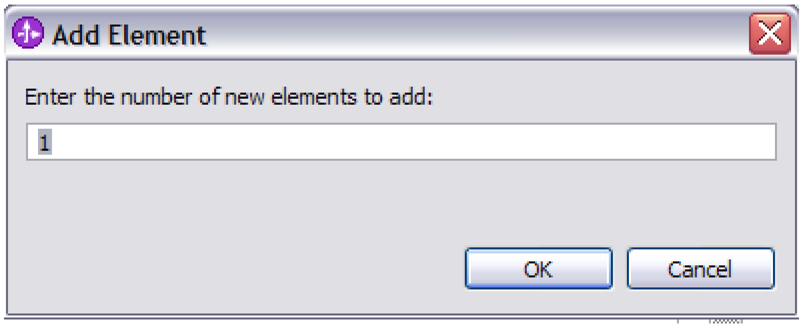
b) Populate the value for the parent business object.

c) Right click the addressobj attribute and select Add Element to create the child object.



Name	Type	Value
createWbidemoCustomerBGInput	WbidemoCustomerBG	✓
verb	verb<string>	✓
WbidemoCustomer	WbidemoCustomer	✓
pkey	string	✓ 100
lname	string	✓ Smith
fname	string	✓ Jerry
ccode	string	✓ Office
addressobj		

d) In the Add Element window, set the value as 1, since there is only one address element.

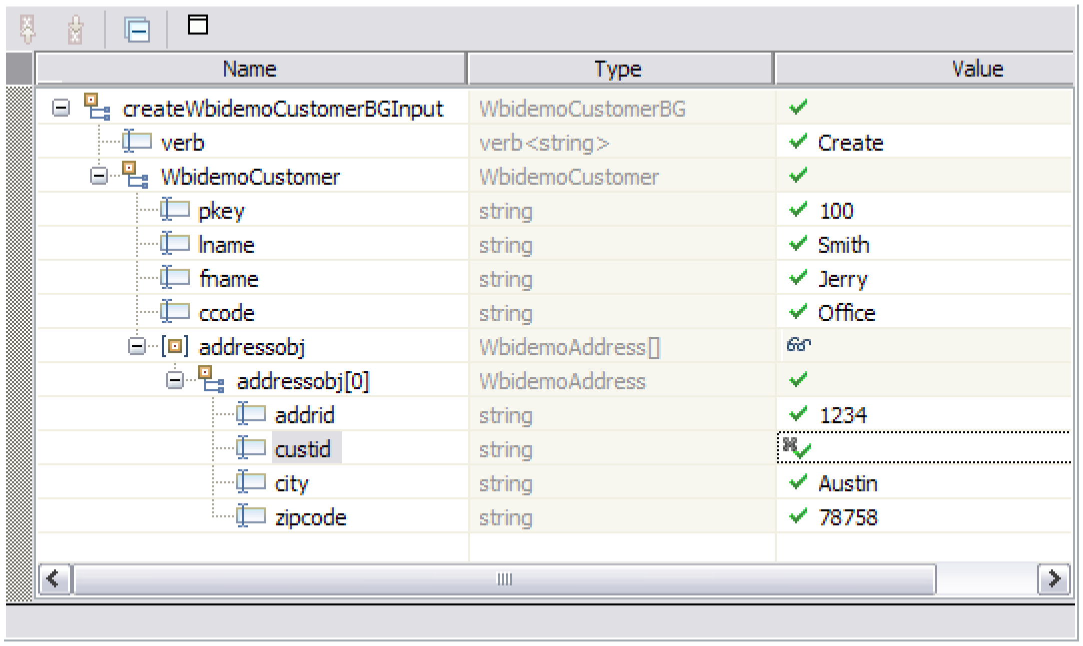


Add Element

Enter the number of new elements to add:

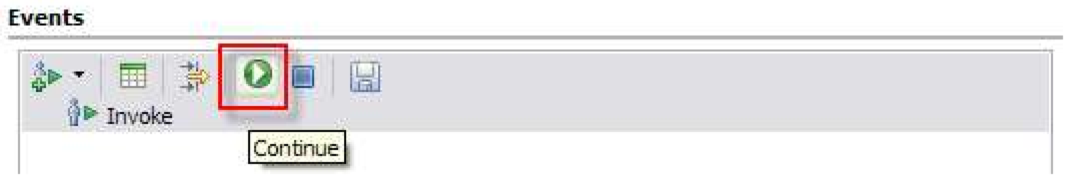
1

__ d. Populate date for the child object. Since **custid** is foreign key of **pkey** in Customer table, set **custid** to unset.

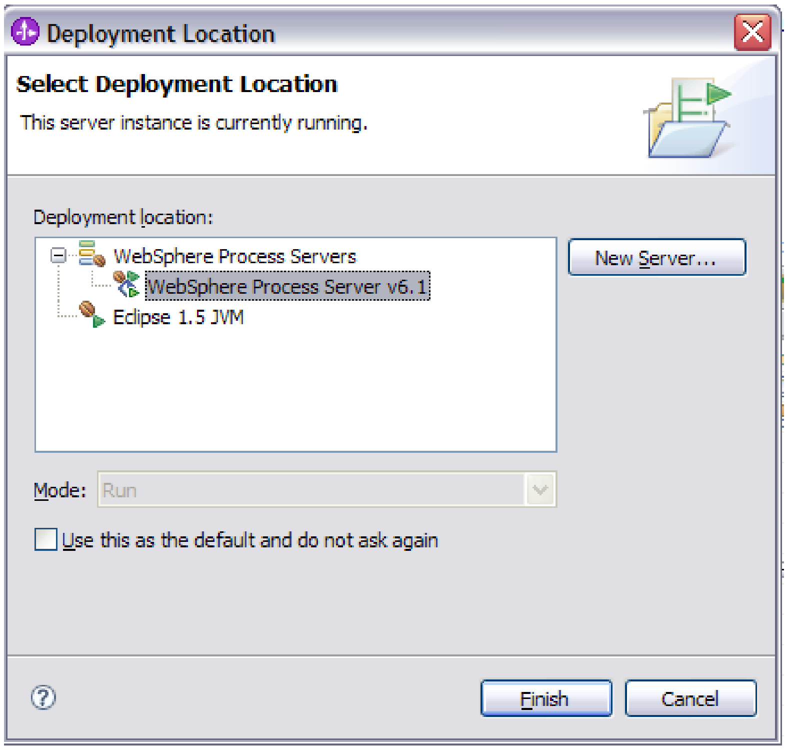


Name	Type	Value
createWbidemoCustomerBGInput	WbidemoCustomerBG	✓
verb	verb<string>	✓ Create
WbidemoCustomer	WbidemoCustomer	✓
pkey	string	✓ 100
lname	string	✓ Smith
fname	string	✓ Jerry
ccode	string	✓ Office
addressobj	WbidemoAddress[]	
addressobj[0]	WbidemoAddress	✓
addrid	string	✓ 1234
custid	string	
city	string	✓ Austin
zipcode	string	✓ 78758

__ e. Click to continue



__ f. In the **Choose a deployment location dialog**, select the **WebSphere Process Server V6.2** server. Select **Finish**



Deployment Location

Select Deployment Location

This server instance is currently running.

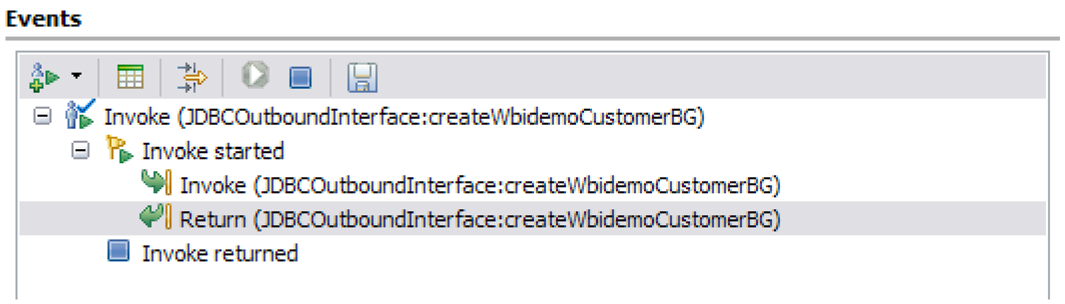
Deployment location:

- WebSphere Process Servers
  - WebSphere Process Server v6.1
- Eclipse 1.5 JVM

Mode: Run

Use this as the default and do not ask again

____ 4. In the Events window you will see that Invoke has returned. Check the data in the EIS to ensure it matches expected values.

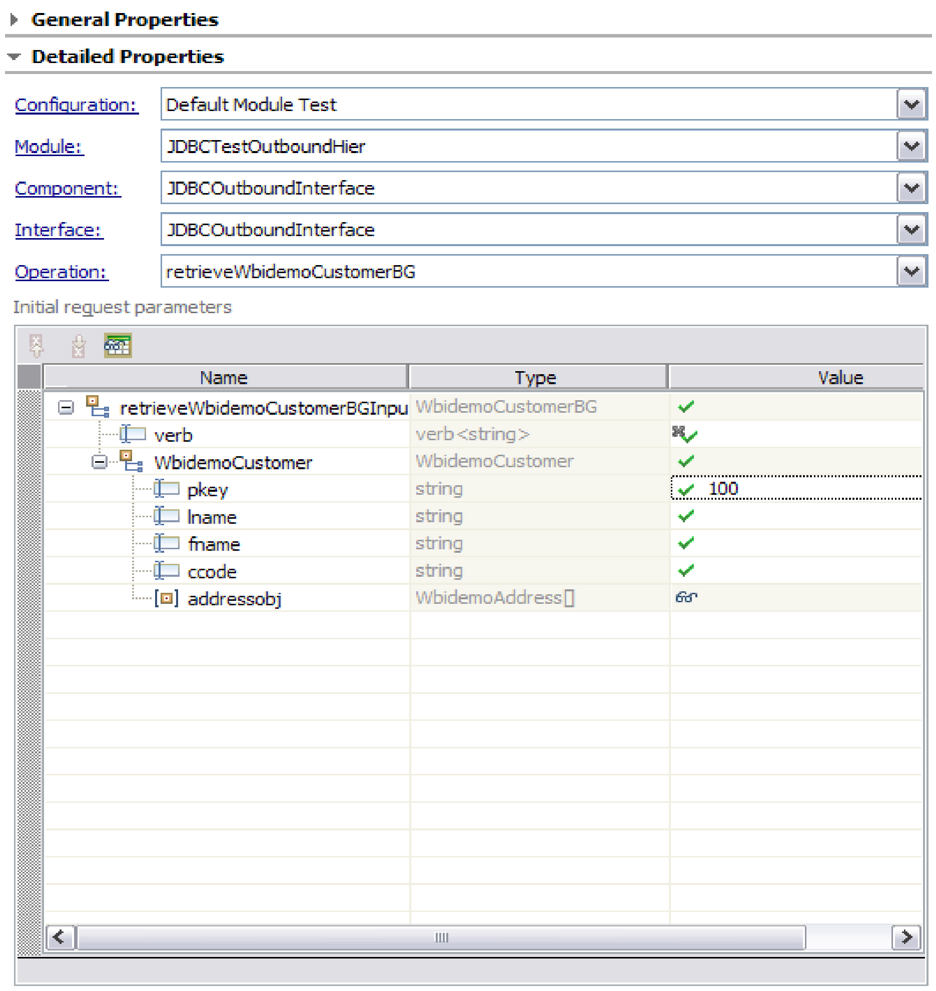


Events

- Invoke (JDBCOutboundInterface:createWbidemoCustomerBG)
  - Invoke started
    - Invoke (JDBCOutboundInterface:createWbidemoCustomerBG)
    - Return (JDBCOutboundInterface:createWbidemoCustomerBG)
  - Invoke returned

____ 5. To retrieve the record that you have just created, select the **Invoke** button in the top right corner again, under **Detailed Properties,** select the Operation **retrieveWbidemoCustomerBG.** Since Address object is a child of Customer object, you only need to enter value for Customer and that retrieves the data for child Address object as well.

__ a. Under **Initial request parameters**, enter a value for pkey of one of the previously created customers. Click **Continue**



General Properties

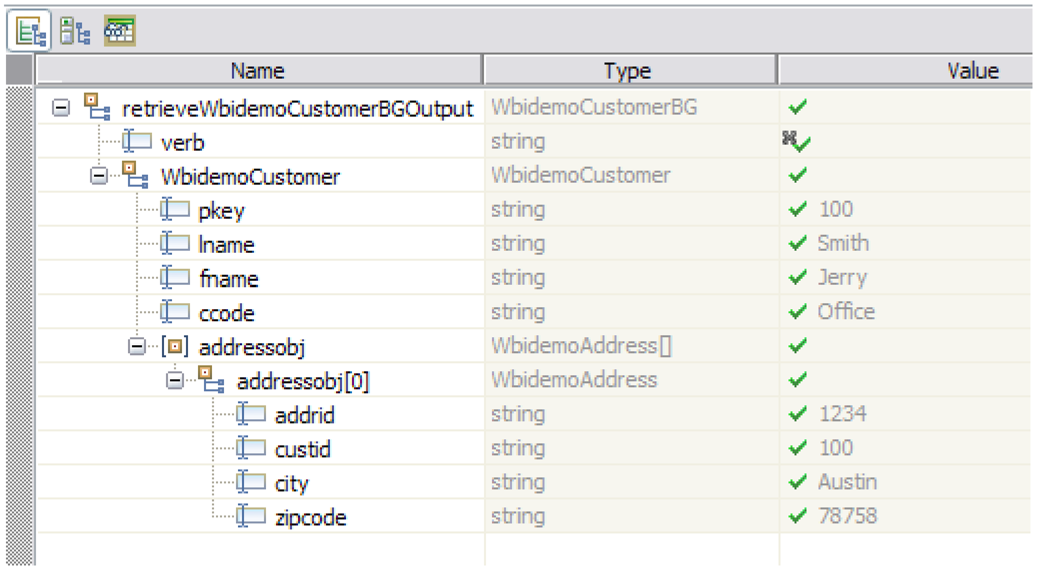
Detailed Properties

Configuration:	Default Module Test
Module:	JDBCTestOutboundHier
Component:	JDBCOutboundInterface
Interface:	JDBCOutboundInterface
Operation:	retrieveWbidemoCustomerBG

Initial request parameters

Name	Type	Value
retrieveWbidemoCustomerBGInpu	WbidemoCustomerBG	✓
verb	verb<string>	✓
WbidemoCustomer	WbidemoCustomer	✓
pkey	string	✓ 100
lname	string	✓
fname	string	✓
ccode	string	✓
addressobj	WbidemoAddress[]	

__ b. Upon return, the values matching the ID specified should be displayed in the Return parameters.

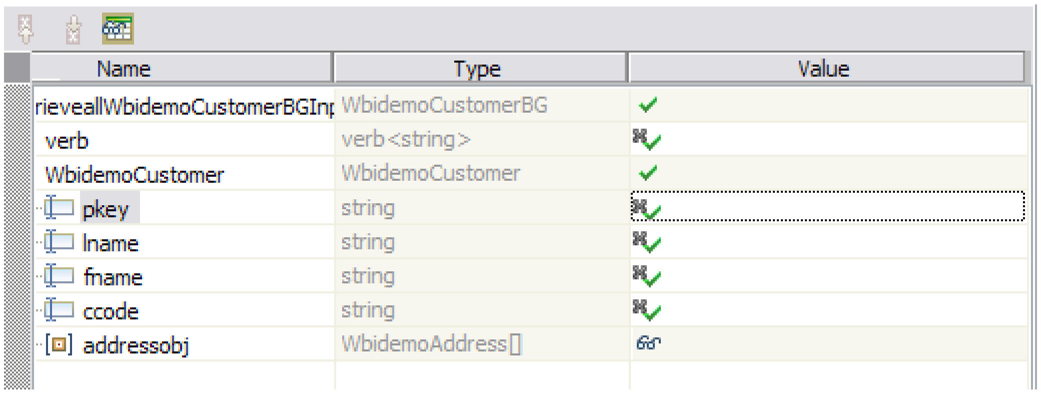


Name	Type	Value
retrieveWbidemoCustomerBGOutput	WbidemoCustomerBG	✓
verb	string	✓
WbidemoCustomer	WbidemoCustomer	✓
pkey	string	✓ 100
lname	string	✓ Smith
fname	string	✓ Jerry
ccode	string	✓ Office
addressobj	WbidemoAddress[]	✓
addressobj[0]	WbidemoAddress	✓
addrid	string	✓ 1234
custid	string	✓ 100
city	string	✓ Austin
zipcode	string	✓ 78758

____ 6. To test a retrieveAll, select the **Invoke** button, then under **Detailed Properties,** select the Operation **retrieveAllWbidemoCustomerBG**

__ a. Set the pkey, lname, fname, and ccode to <unset> by clicking on the field and selecting the <unset> from the drop down menu.

__ b. Click **Continue.**



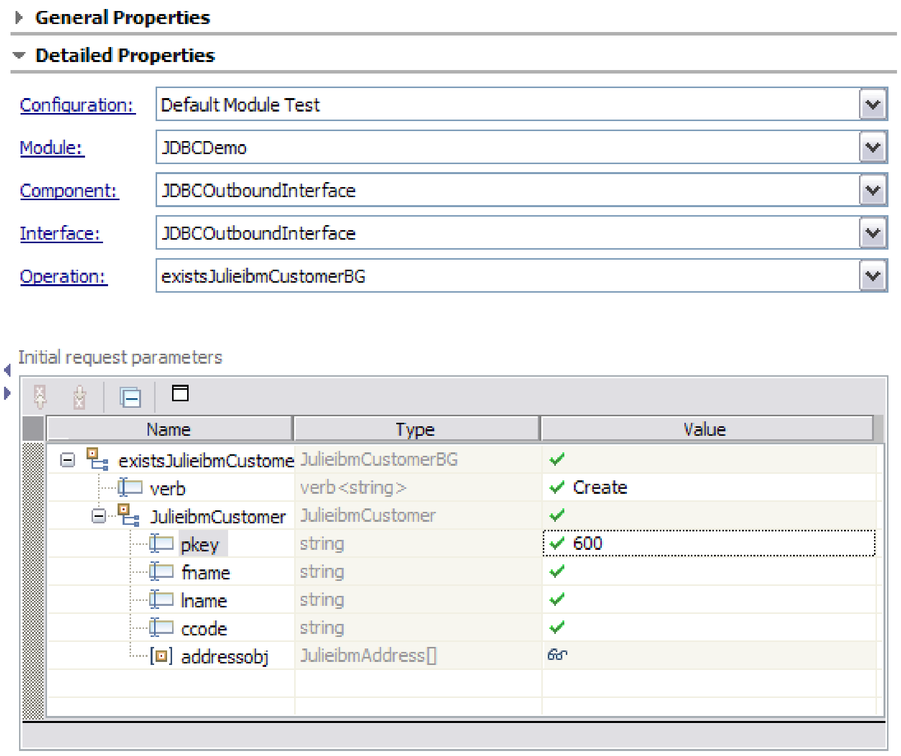
Name	Type	Value
rieveallWbidemoCustomerBGInp	WbidemoCustomerBG	✓
verb	verb<string>	✓
WbidemoCustomer	WbidemoCustomer	✓
pkey	string	✓
lname	string	✓
fname	string	✓
ccode	string	✓
addressobj	WbidemoAddress[]	

__ c. Upon return, the values for Customer table existing in the database should be displayed including values from the Address table.

__ d. View the return parameters box to check for the returned records scrolling as needed.

____ 7. To test whether records exist or not, select the **Invoke** button, then under **Detailed Properties**, select the Operation **existsWbidemoCustomerBG**. This returns the status of the query and numbers of records based on the input criteria.

__ a. Under **Initial request parameters**, enter a value for pkey of one of the previously created customers.



General Properties

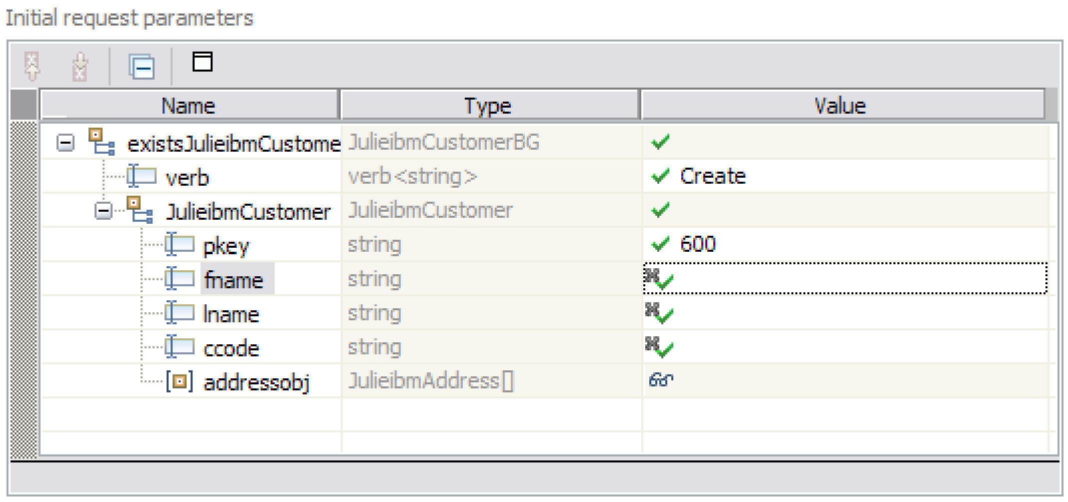
Detailed Properties

Configuration:	Default Module Test
Module:	JDBCDemo
Component:	JDBCOutboundInterface
Interface:	JDBCOutboundInterface
Operation:	existsJulieibmCustomerBG

Initial request parameters

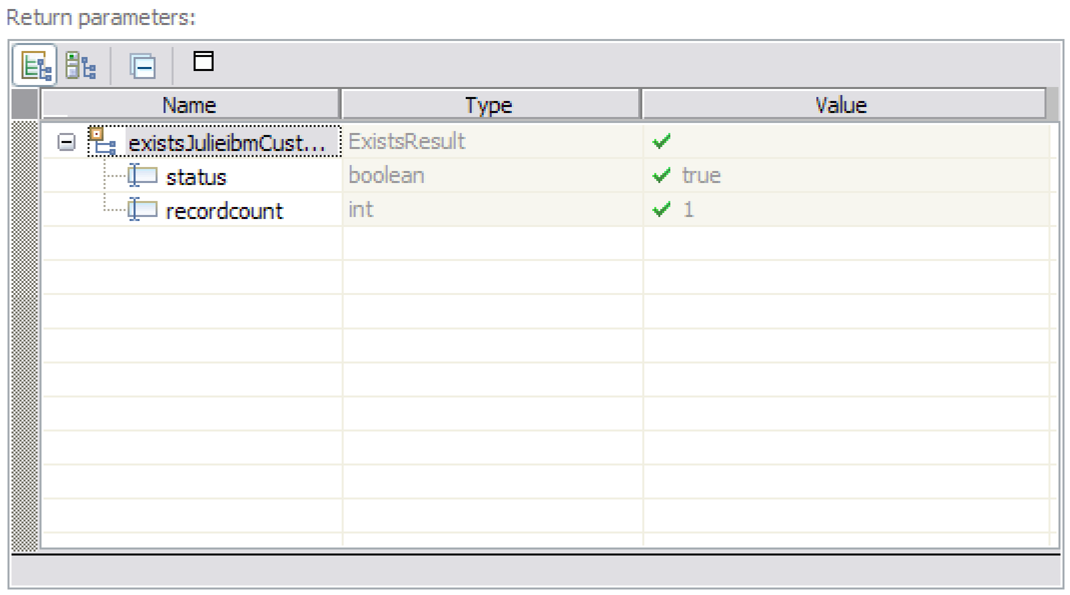
Name	Type	Value
existsJulieibmCustome	JulieibmCustomerBG	✓
verb	verb<string>	✓ Create
JulieibmCustomer	JulieibmCustomer	✓
pkey	string	✓ 600
fname	string	✓
lname	string	✓
ccode	string	✓
addressobj	JulieibmAddress[]	

__ b. Set lname, fname, and ccode to <unset> by clicking on the field and selecting the <unset> option from the drop down menu. Click **Continue.**



Name	Type	Value
existsJulieibmCustome	JulieibmCustomerBG	✓
verb	verb<string>	✓ Create
JulieibmCustomer	JulieibmCustomer	✓
pkey	string	✓ 600
fname	string	✓
lname	string	✓
ccode	string	✓
addressobj	JulieibmAddress[]	

__ c. Upon return, the result should be displayed with Boolean value whether a particular record exists in the database or not. RecordCount returns the number of records found in the database for the input criteria.

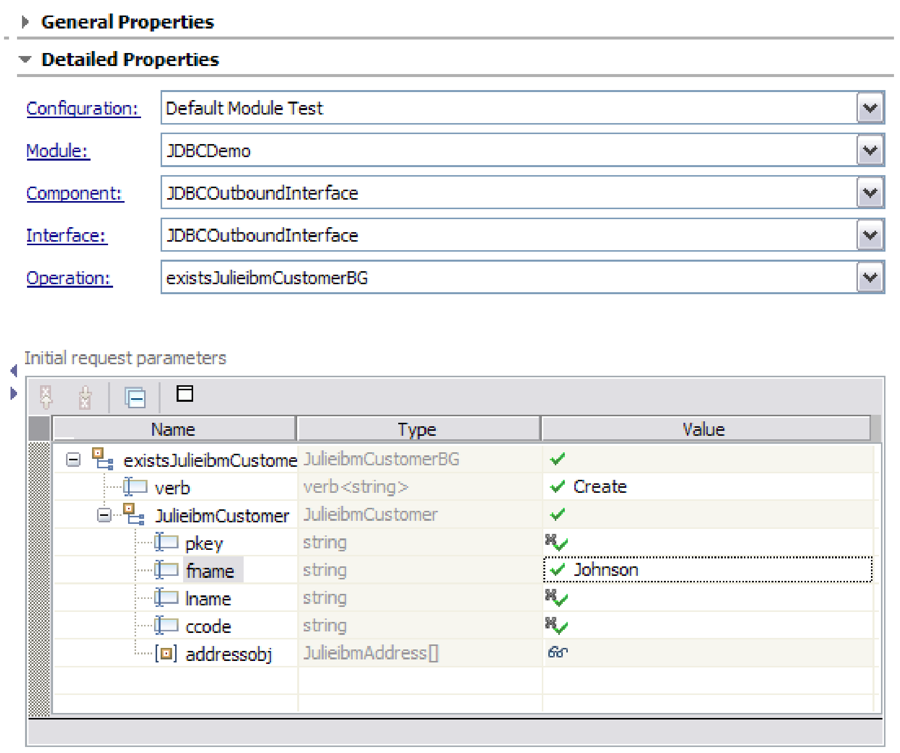


Name	Type	Value
existsJulieibmCust...	ExistsResult	✓
status	boolean	✓ true
recordcount	int	✓ 1

__ d. View the return parameters with EIS data to ensure the count is correct.

____ 8. User can also query records based on any attribute values and not just based on primary key attributes. Select the **Invoke** button, then under **Detailed Properties**, select the Operation **existsWbidemoCustomerBG**.

__ a. Under **Initial request parameters**, enter value for any attribute created customers.



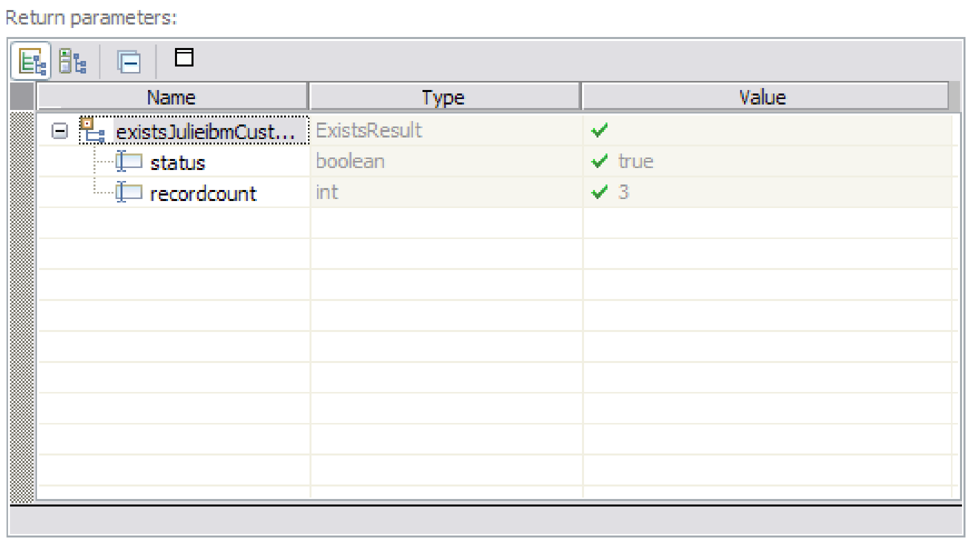
Configuration: Default Module Test
Module: JDBCDemo
Component: JDBCOutboundInterface
Interface: JDBCOutboundInterface
Operation: existsJulieibmCustomerBG

Initial request parameters

Name	Type	Value
existsJulieibmCustome	JulieibmCustomerBG	✓
verb	verb<string>	✓ Create
JulieibmCustomer	JulieibmCustomer	✓
pkey	string	
fname	string	✓ Johnson
lname	string	
ccode	string	
addressobj	JulieibmAddress[]	

__ b. Set remaining attributes to <unset> by clicking on the field and selecting the <unset> option from the drop down menu. Click **Continue.**

__ c. Upon return, the result should be displayed with Boolean value whether a particular record exists in the database or not. RecordCount returns the number of records found in the database for the input criteria.


Return parameters:

Name	Type	Value
existsJulieibmCust...	ExistsResult	✓
status	boolean	✓ true
recordcount	int	✓ 3

__ d. View the return parameters with EIS data to ensure the count is correct.

____ 9. Exit the Test Component panel, remove the JDBCTestOutboundHier project from the server, and stop the server.

What you did in this exercise

- In this exercise, you learned how to create a multiple cardinality business objects Hierarchy using External Service Wizard, instead of editing the parent and child Business Objects using Business Object Editor.

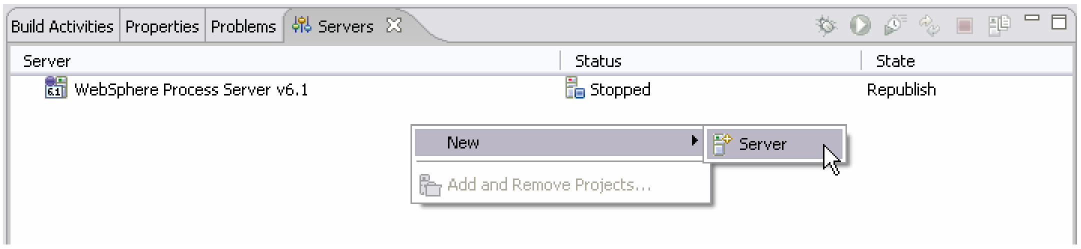
Task: Adding remote server to the WebSphere Integration Developer test environment

This task describes how to add a remote server to the WebSphere Integration Developer Test environment. This example uses a z/OS machine.

____ 1. Define a new remote server to WebSphere Integration Developer.

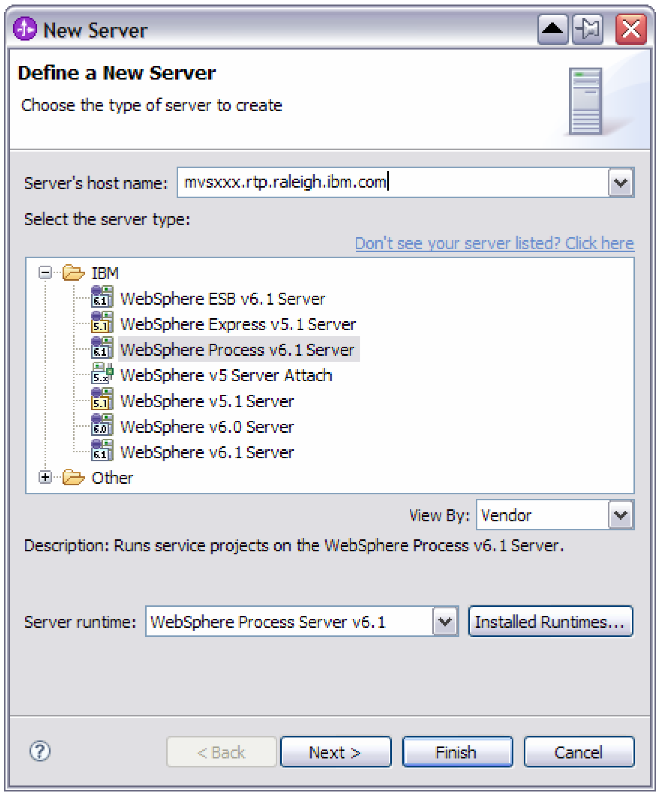
__ a. Right click the background of the **Servers** view to access the pop-up menu.

__ b. Select **New → Server**.



__ c. In the New Server dialog, specify the remote server's host name, **<HOSTNAME>**.

__ d. Ensure that the appropriate server type, '**WebSphere Process v6.2 Server**' or '**WebSphere ESB v6.2 Server**', is highlighted in the server type list


New Server
Define a New Server
Choose the type of server to create
Select the server type:
Don't see your server listed? Click here
IBM
WebSphere ESB v6.1 Server
WebSphere Express v5.1 Server
WebSphere Process v6.1 Server
WebSphere v5 Server Attach
WebSphere v5.1 Server
WebSphere v6.0 Server
WebSphere v6.1 Server
Other
Installed Runtimes...
< Back
Next >
Finish
Cancel

Server's host name: mvsxxx.rtp.raleigh.ibm.com

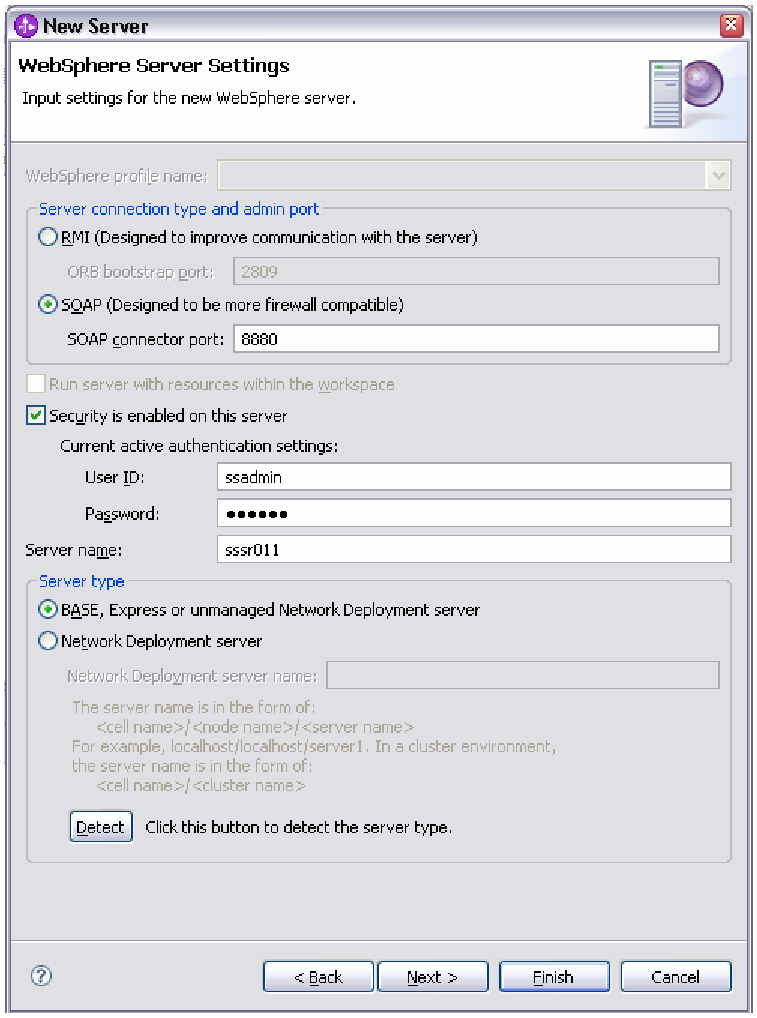
View By: Vendor

Description: Runs service projects on the WebSphere Process v6.1 Server.

Server runtime: WebSphere Process Server v6.1

__ e. Click **Next.**

__ f. On the WebSphere Server Settings page, leave the radio button for **SOAP** selected, changing the **SOAP connector port** to the correct setting (**<SOAP_PORT>**). If security is on in your server, check the box for **'Security is enabled on this server'** and input **<USERID>** for the user ID and **<PASSWORD>** for the password.


New Server
WebSphere Server Settings
Input settings for the new WebSphere server.

WebSphere profile name:

Server connection type and admin port

( ) RMI (Designed to improve communication with the server)

ORB bootstrap port: 2809

(●) SOAP (Designed to be more firewall compatible)

SOAP connector port: 8880

☐ Run server with resources within the workspace

☑ Security is enabled on this server

Current active authentication settings:

User ID: ssadmin

Password: ●●●●●●

Server name: sssr011

Server type

(●) BASE, Express or unmanaged Network Deployment server

( ) Network Deployment server

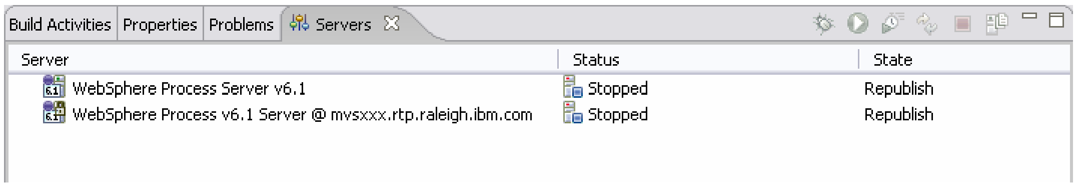
Network Deployment server name:

The server name is in the form of:
<cell name>/<node name>/<server name>
For example, localhost/localhost/server1. In a cluster environment, the server name is in the form of:
<cell name>/<cluster name>

Detect Click this button to detect the server type.

__ g. Click **Finish**.

__ h. The new server should be seen in the Server view.



Server	Status	State
WebSphere Process Server v6.1	Stopped	Republish
WebSphere Process v6.1 Server @ mvsxxx.rtp.raleigh.ibm.com	Stopped	Republish

____ 2. Start the remote server if it is not already started. WebSphere Integration Developer does not support starting remote servers from the Server View.

__ a. From a command prompt, telnet to the remote system if needed:

‘**telnet <HOSTNAME> <TELNET_PORT>**’

User ID : **<USERID>**

Password : **<PASSWORD>**

__ b. Navigate to the bin directory for the profile being used:

cd <WAS_HOME>/profiles/<PROFILE_NAME>/bin

__ c. Run the command file to start the server: **./startServer.sh <SERVER_NAME>**

__ d. Wait for status message indicating server has started:

```
ADMU3200I: Server launched. Waiting for initialization status

ADMU3000I: Server sssr01 open for e-business; process id is 0000012000000002
```