

IBM WEBSHERE PROCESS INTEGRATION 6.0 – LAB EXERCISE

JDBCOutbound Lab

What this exercise is about	1
Lab Requirements	1
What you should be able to do	1
Introduction	2
Exercise Instructions	2
Part 1: Create the JDBCTEST Database using Cloudscape	3
Part 2: Set up the Development Environment	8
Part 3: Create the JDBCOutbound Application.....	9
Part 4: Test the application using the WebSphere Test Environment (WTE) and Component Test.....	21
What you did in this exercise	34

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

List of system and software required for the student to complete the lab

- WebSphere Integration Developer V6 installed
- WebSphere Process Server V6 test environment installed
- WebSphere Adapter for JDBC V6 installed
- Sample code in the directory C:\Labfiles60\JDBC (Windows) or /tmp/LabFiles60/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.

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 Cloudscape database that contains a CUSTOMER table. In the lab, you will import the JDBC Adapter into WebSphere Integration Developer and run Enterprise Discovery 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 standalone 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 may be Windows operating-system specific. If you plan on running the lab on an operating-system other than Windows, you will need to execute the appropriate commands, and use appropriate files (.sh vs. .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\6.0	
<WPS_HOME>	C:\<WID_HOME>\runtimes\bi_v6	
<JDBCADAPTER_HOME>	C:\Program Files\IBM\ResourceAdapters\JDBC	
<LAB_FILES>	C:\Labfiles60	/tmp/Labfiles60
<TEMP>	C:\temp	/tmp

Windows users please 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:\LabFiles60\ would be replaced by C:/LabFiles60/

Part 1: Create the JDBCTEST Database using Cloudscape

In this part you will create the JDBCTEST database along with the CUSTOMER table for user data. You will also create a WBIA_JDBC_EVENTSTORE table and triggers against the CUSTOMER table. The triggers will insert a record into the EVENTSTORE table for each create, update, or delete event occurring in the CUSTOMER table. This lab exercises outbound requests; and therefore, will not directly use the triggers nor the EVENTSTORE table. However, by creating records to the CUSTOMER table in this lab, you will be priming the WBIA_JDBC_EVENTSTORE table with event records in preparation for the JDBCInBound lab.

If you choose to run the JDBCInBound lab first, there are also instructions at the beginning of that lab to create the database, tables, and triggers, as well as create a new row in the CUSTOMER table. Therefore, it does not matter which JDBC lab you do first.

- ___ 1. Start Cloudscape Cview Graphical User Interface (GUI) by running the cvview.bat.

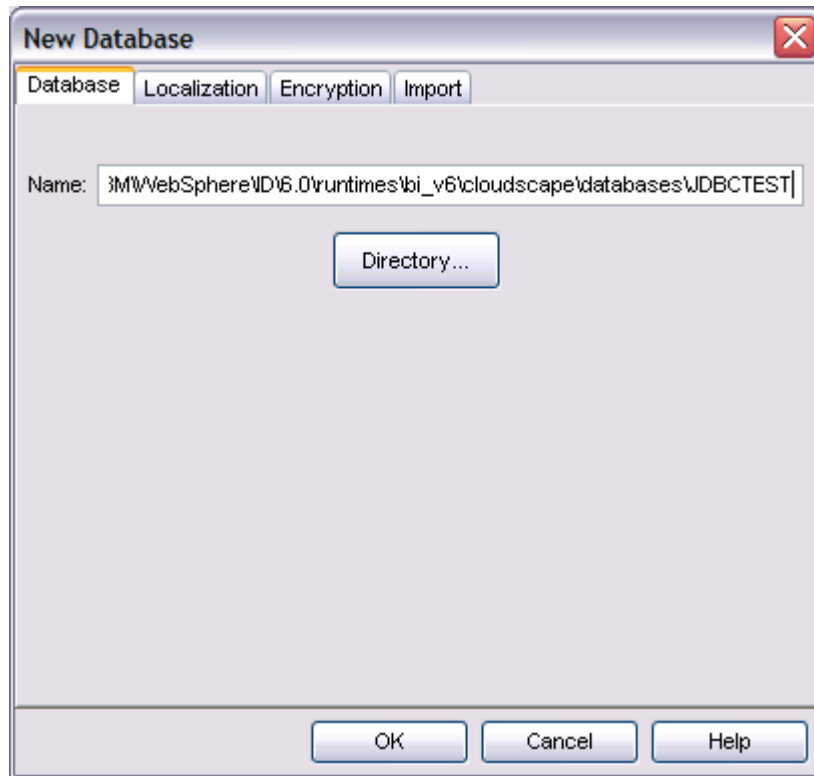
Note: The Cloudscape embedded driver that is being used in the lab, supports a connection from only one JVM at a time. You can have either the server running and connected to the JDBCTEST database, or the Cview GUI connected to the JDBCTEST database; but not both at the same time.

- ___ a. Open a command prompt window, navigate to the following subdirectory, and run the cvview.bat program.

```
<WPS_HOME>\cloudscape\bin\embedded>cvview
```

- ___ 2. Using the CView GUI, create the JDBCTEST database if it doesn't already exist. If you've already completed the JDBCInbound lab, the JDBCTEST database, tables, and triggers are the same. You can skip the remainder of Part 1: steps and continue with Part 2:

- ___ a. Select from the menu, **File -> New -> Database** Click the **Directory ...** button and browse to **<WPS_HOME>\cloudscape\databases** subdirectory , enter the name of the database to create, **JDBCTEST**, and click **Open**. You will return to the **New Database** panel as shown: Click **OK**



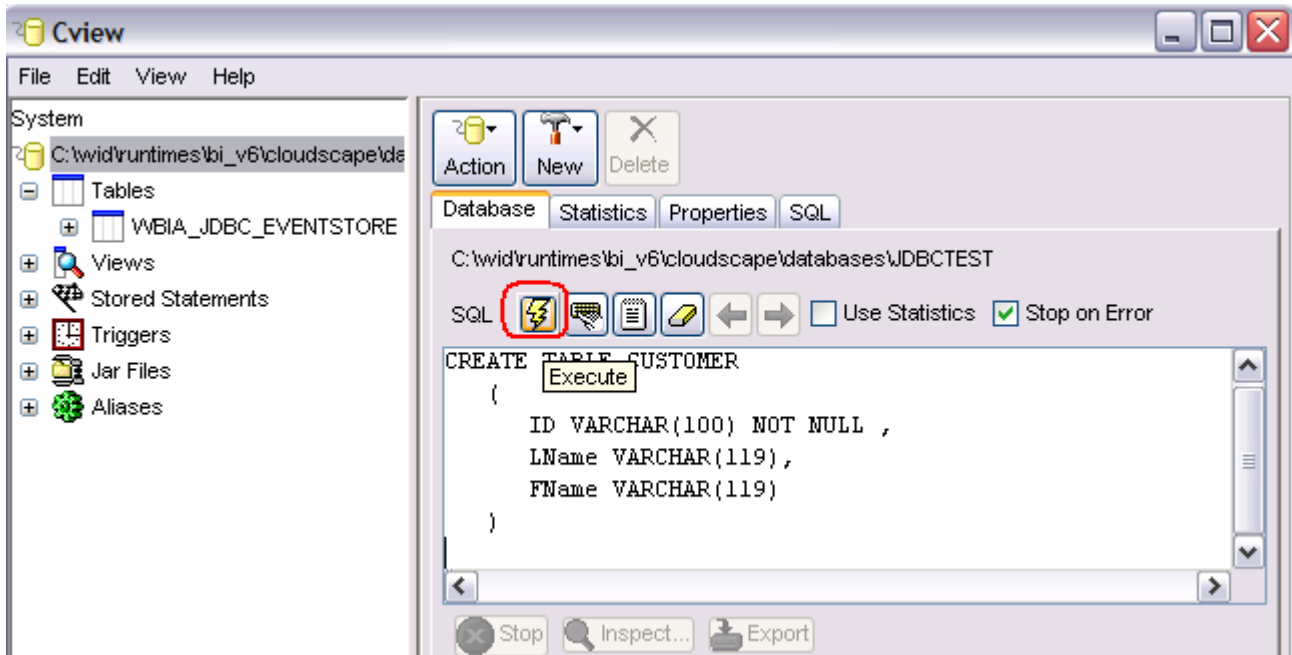
- ___ 3. Select the JDBCTEST Database, select the Database tab, and enter SQL to create two tables: CUSTOMER table for user data and WBIA_JDBC_EVENTSTORE table for recording events.

NOTE: For your convenience, the following sql code snippets can be found in <LAB_FILES>\JDBC\snippets\CUSTOMERSQL.txt

- ___ a. Select the JDBCTEST Database, select the Database tab on the right, paste the following into the SQL window:

```
CREATE TABLE CUSTOMER
(
  ID VARCHAR(100) NOT NULL ,
  LName VARCHAR(119),
  FName VARCHAR(119)
)
```

- ___ b. Select the “lightning bolt” icon to execute the SQL (The SQL text will appear highlighted after executing.)



- __ c. Set the key for the CUSTOMER table by pasting the following into the SQL window (you can simply paste over the top of and replace the previous existing SQL shown in the window)

```
ALTER TABLE CUSTOMER
ADD CONSTRAINT NEW_KEYCU Primary Key (
ID)
```

- __ d. Select the “lightning bolt” icon to execute the SQL

- ___ 4. Create the WBIA_JDBC_EVENTSTORE table and create the primary key.

NOTE: For your convenience, the following sql code snippets can be found in <LAB_FILES>JDBC/snippets/EVENTSTORESQL.txt

- __ a. Create the WBIA_JDBC_EVENTSTORE table by pasting the following into the SQL window:

```
CREATE TABLE WBIA_JDBC_EVENTSTORE
(
EVENT_ID INT DEFAULT AUTOINCREMENT INITIAL 1 INCREMENT 1 NOT NULL ,
OBJECT_KEY VARCHAR(80) NOT NULL ,
OBJECT_NAME VARCHAR(40) NOT NULL ,
OBJECT_FUNCTION VARCHAR(40) NOT NULL ,
EVENT_PRIORITY INT NOT NULL ,
EVENT_TIME TIMESTAMP DEFAULT CURRENT TIMESTAMP NOT NULL ,
EVENT_STATUS INT NOT NULL ,
EVENT_COMMENT VARCHAR(100)
)
```

- __ b. Select the “lightning bolt” icon to execute the SQL.

- __ c. Create the primary key by pasting the following into the SQL window:

```
ALTER TABLE WBJA_JDBC_EVENTSTORE
ADD CONSTRAINT "c3350098-0104-7683-90b5-ffffe0415391" Primary Key (
EVENT_ID)
```

__ d. Select the "lightning bolt" icon to execute the SQL.

___ 5. Create triggers on the CUSTOMER database for create, update, and delete events.

NOTE: For your convenience, the following sql code snippets can be found in
<LAB_FILES>\JDBC\snippets\CUSTOMERTRIGGERSQL.txt

__ a. Create the trigger for the create event by pasting the following into the SQL window:

```
CREATE TRIGGER event_create
AFTER INSERT ON CUSTOMER REFERENCING NEW AS N
FOR EACH ROW MODE DB2SQL
INSERT INTO wbia_jdbc_eventstore (object_key, object_name, object_function,
event_priority, event_status)
VALUES (N.id, 'AppCustomerBG', 'Create', 1, 0)
```

__ b. Select the "lightning bolt" to execute the SQL.

__ c. Create the trigger for the update event by pasting the following into the SQL window:

```
CREATE TRIGGER event_update
AFTER UPDATE ON CUSTOMER REFERENCING NEW AS N
FOR EACH ROW MODE DB2SQL
INSERT INTO wbia_jdbc_eventstore (object_key, object_name, object_function,
event_priority, event_status)
VALUES (N.id, 'AppCustomerBG', 'Create', 1, 0)
```

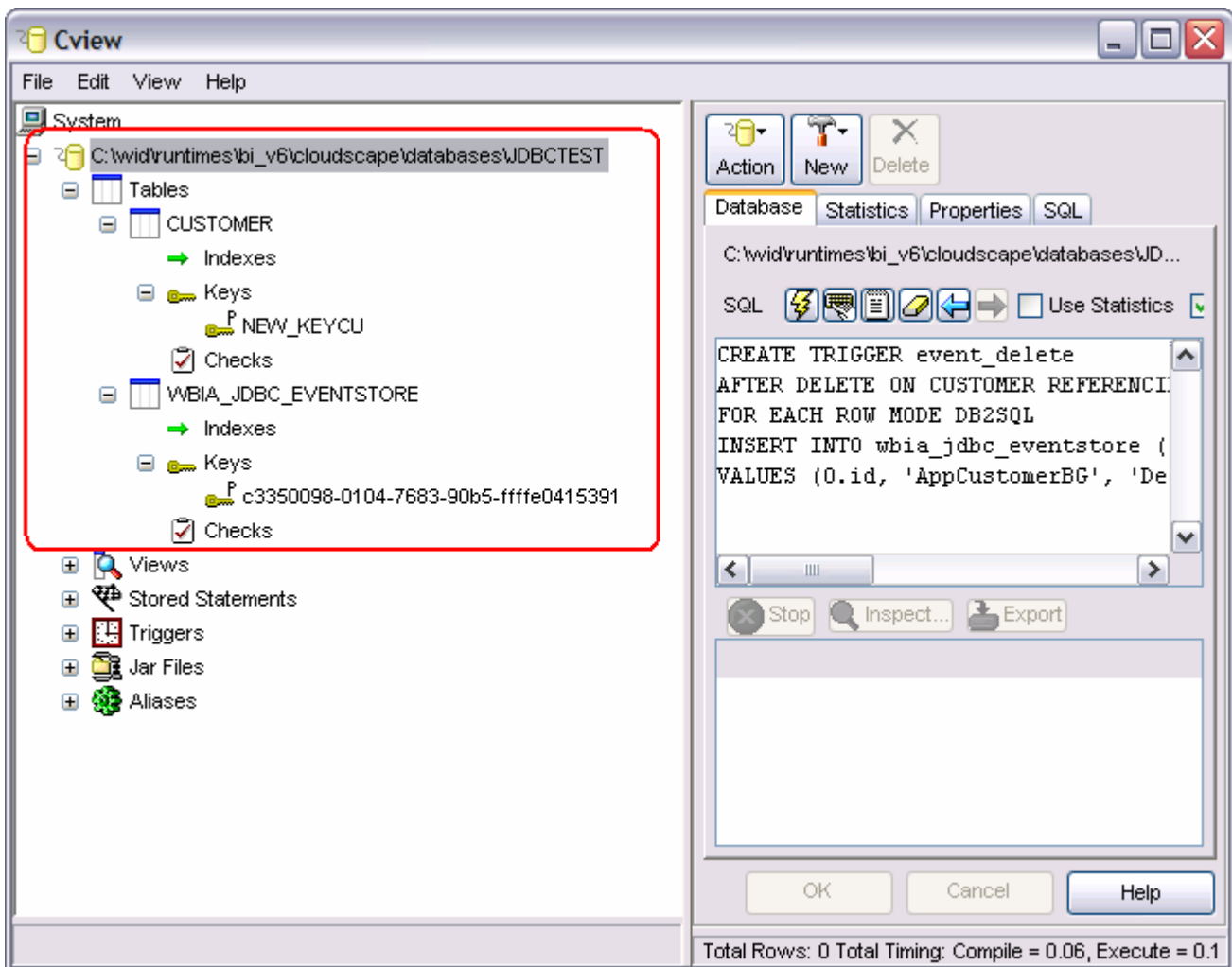
__ d. Select the "lightning bolt" to execute the SQL.

__ e. Create the trigger for the delete event by pasting the following into the SQL window:

```
CREATE TRIGGER event_delete
AFTER DELETE ON CUSTOMER REFERENCING OLD AS O
FOR EACH ROW MODE DB2SQL
INSERT INTO wbia_jdbc_eventstore (object_key, object_name, object_function,
event_priority, event_status)
VALUES (O.id, 'AppCustomerBG', 'Delete', 1, 0)
```

__ f. Select the "lightning bolt" to execute the SQL.

6. Verify that your JDBCTEST database and tables look similar to the following. Close the database and exit the Cview GUI.

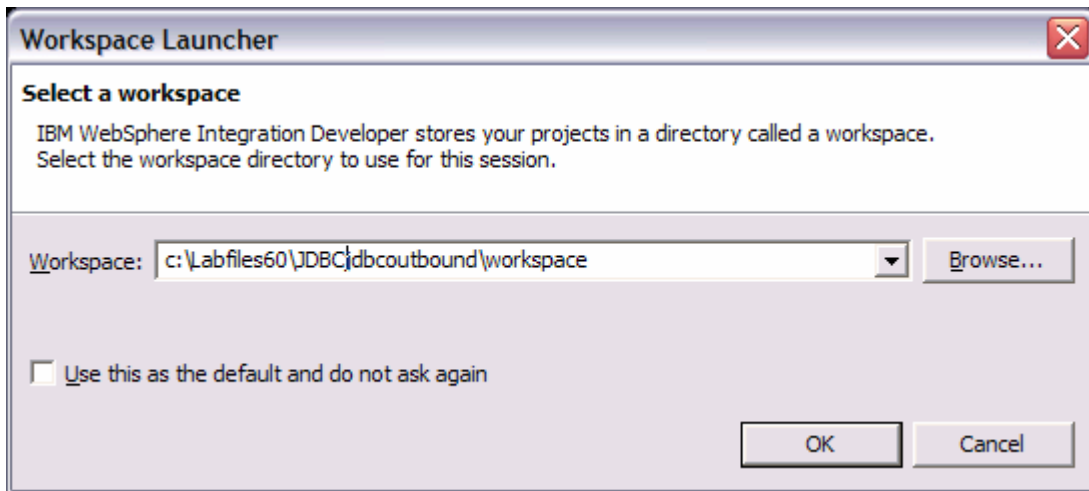


Note: You must exit the Cview GUI, before starting the server as the Cloudscape “embedded” database driver used in this lab allows for only one JVM connection at a time.

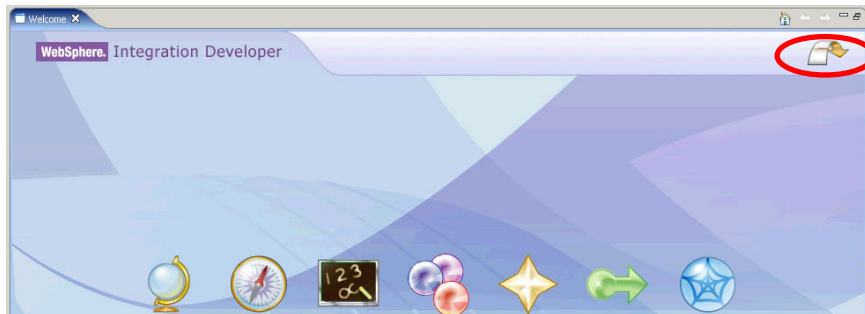
Part 2: Set up the Development Environment

In this part, you will start WebSphere Integration Developer and set up the WebSphere Process Server to be used as the WebSphere Test Environment (WTE).

- ___ 1. Start WebSphere Integration Developer v6 with a new workspace
 - ___ a. From the start menu select **Start > Programs > IBM WebSphere > Integration Developer V6.0 > WebSphere Integration Developer V6.0**
 - ___ b. When prompted enter `<LAB_FILES>\JDBC\jdbcoutbound\workspace` for your workspace and click **OK**



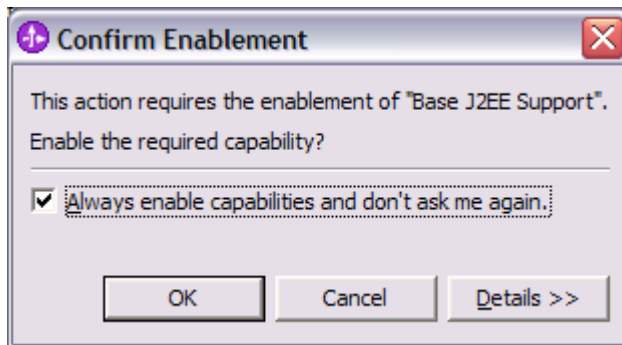
- ___ c. When WebSphere Integration Developer v6.0 opens, close the **Welcome page**



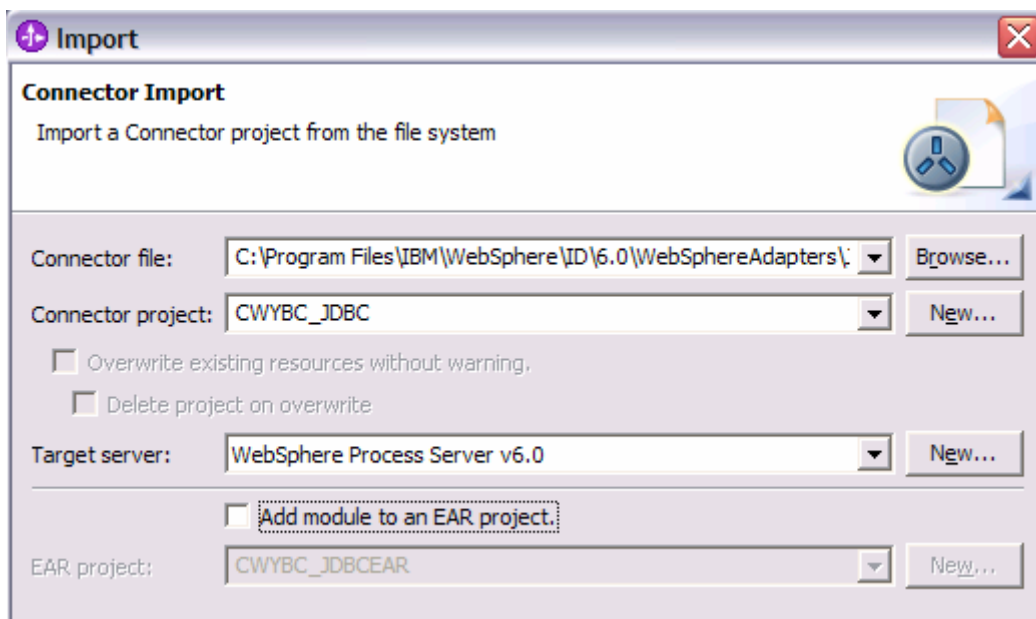
Part 3: Create the JDBCOutbound Application

In this part you will import the WebSphere Adapter for JDBC, run Enterprise Service Discovery to discover objects and create the necessary SCA artifacts, and assemble the adapter into an SCA application.

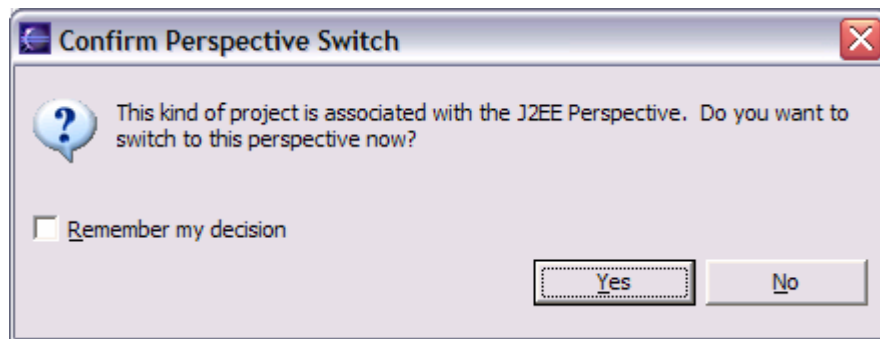
- ___ 1. Import the JDBC Adapter Resource Archive (RAR) file. This will create a J2EE Connector Project.
 - ___ a. From the top Menu bar, select **File -> Import**, Select **RAR file**, click **next**
 - ___ b. At the Confirm Enablement popup window, check the box and click **OK**



- ___ 2. Complete the Connector Import panel
 - ___ a. Connector file: **Browse...** to the location of the CWYBC_JDBC.rar adapter file and select it, for example: c:\<WID_HOME>WebSphereAdapters\JDBC\CWYBC_JDBC.rar
 - ___ b. Leave the Connector project: value as it defaults to **CWYBC_JDBC**
 - ___ c. Deselect the **“Add module to an EAR project”** and click **Finish**.

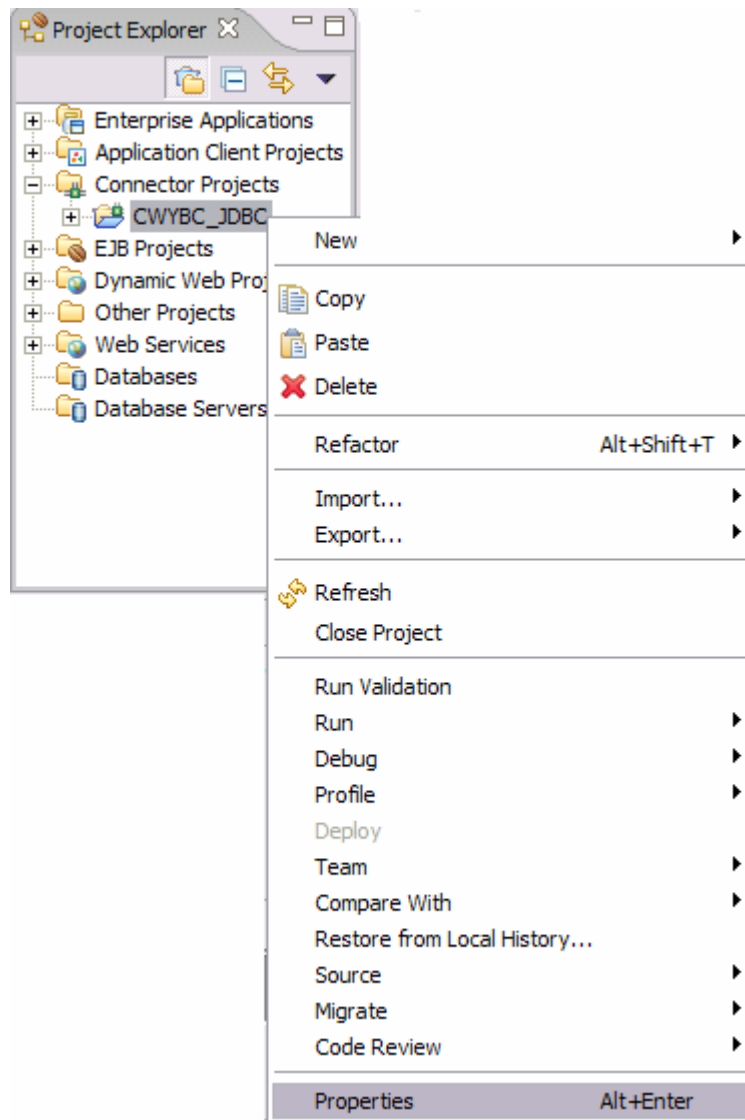


___ d. At the Confirm Perspective Switch popup window, click **Yes**

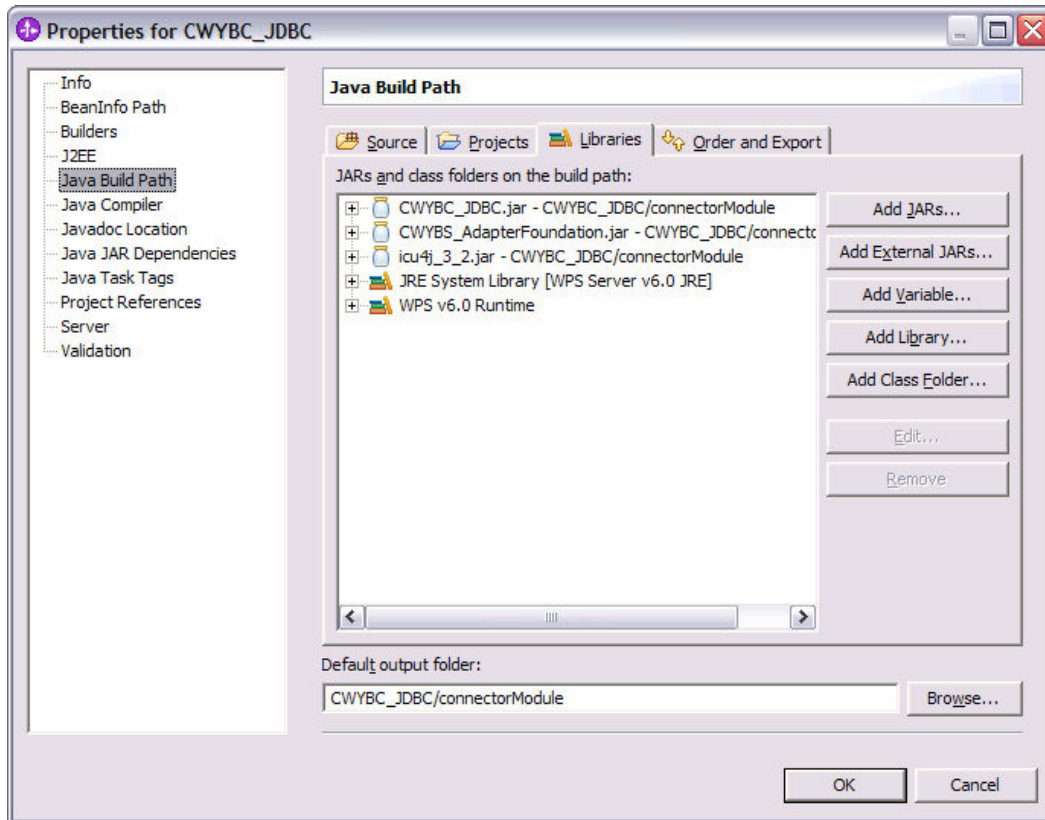


___ 3. Add any external dependencies your adapter has to the imported project. These are dependencies that the adapter may have on the JDBC applications (adapter-specific). Add the db2j.jar to the build path of the Connector project.

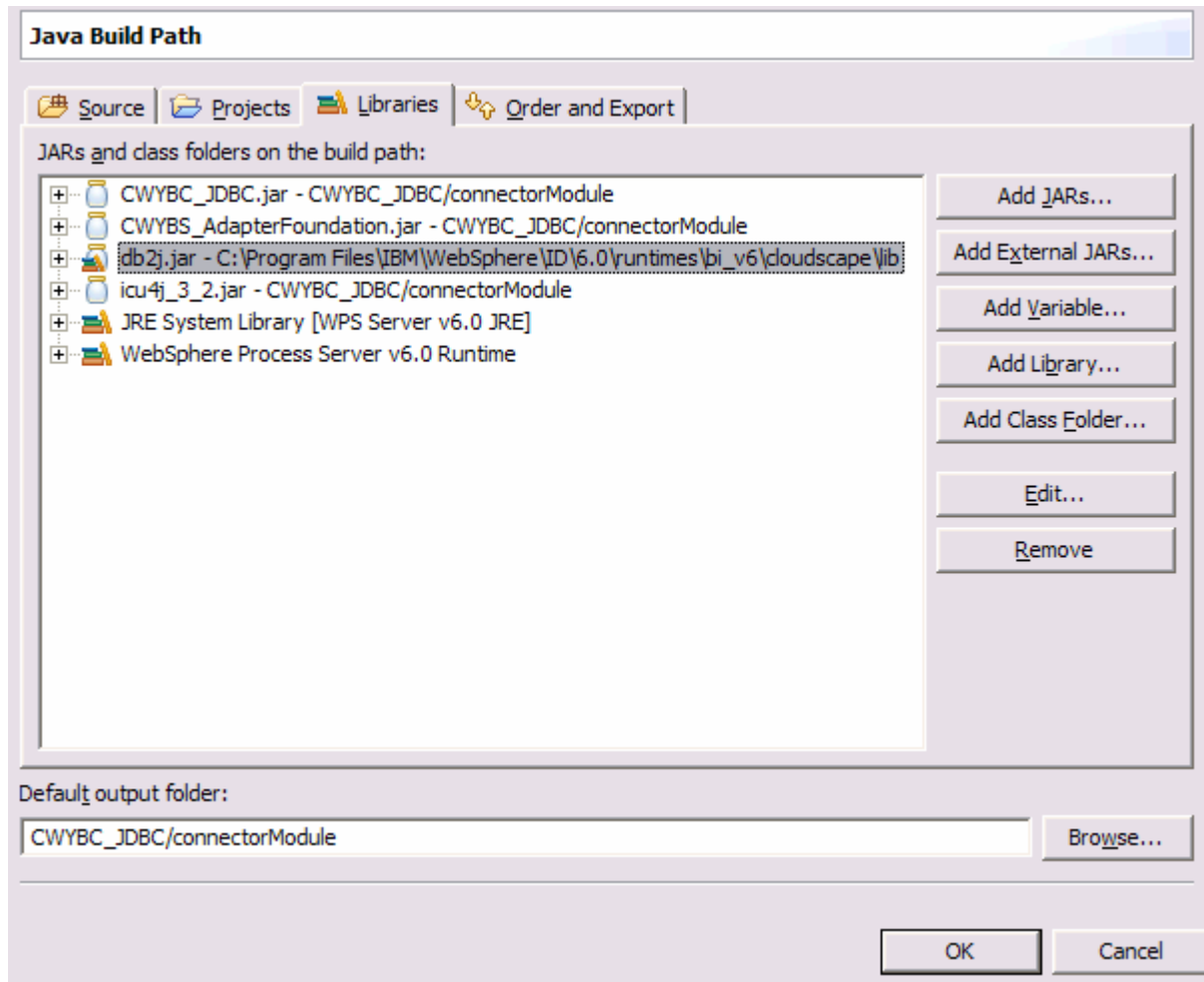
- ___ a. Expand the **Connector Projects** folder, right click the **CWYBC_JDBC** connector project, select **properties**.



- ___ b. Select **Java Build Path** from the list at the left, select the **Libraries** tab from the panel at the right, select **Add External JARs...** button

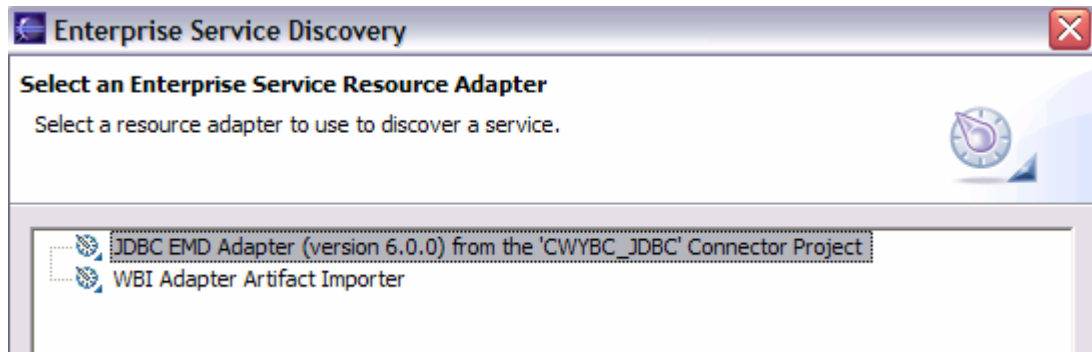


- ___ c. Browse to the location of c:\<WPS_HOME>\cloudscape\lib and select the **db2j.jar**, click **Open**.
- ___ d. You will now see the db2j.jar added in your JARS and class folders in the build path: list. Click **OK**



- ___ 4. Switch to the Business Integration Perspective and run the Enterprise Service Discovery wizard. A Business Integration project will be created for you during this process.
 - ___ a. From the top Menu bar, select **Window > Open Perspective > Other ... > Business Integration (default)** click **OK**
 - ___ b. From within the Business Integration Perspective, select **File > New > Enterprise Service Discovery**

___ c. Highlight the **JDBC EMD Adapter** and click **Next**



___ 5. Complete the Configure Settings for Discovery Agent panel to connect to the JDBCTEST database and discover the available services. Click **Next**

___ a. Enter valid userid and password values, for example

Username: **WSdemo**

Password: **WS15Demo1**

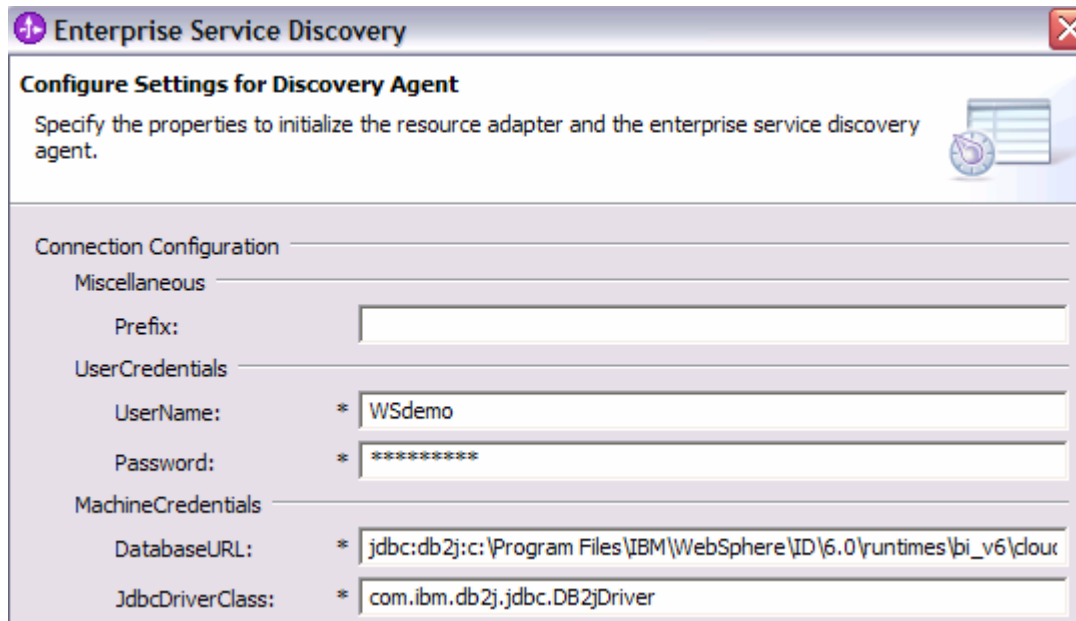
___ b. Enter the following values for DatabaseURL and JdbcDriverClass

DatabaseURL: jdbc:db2j:c:\Program

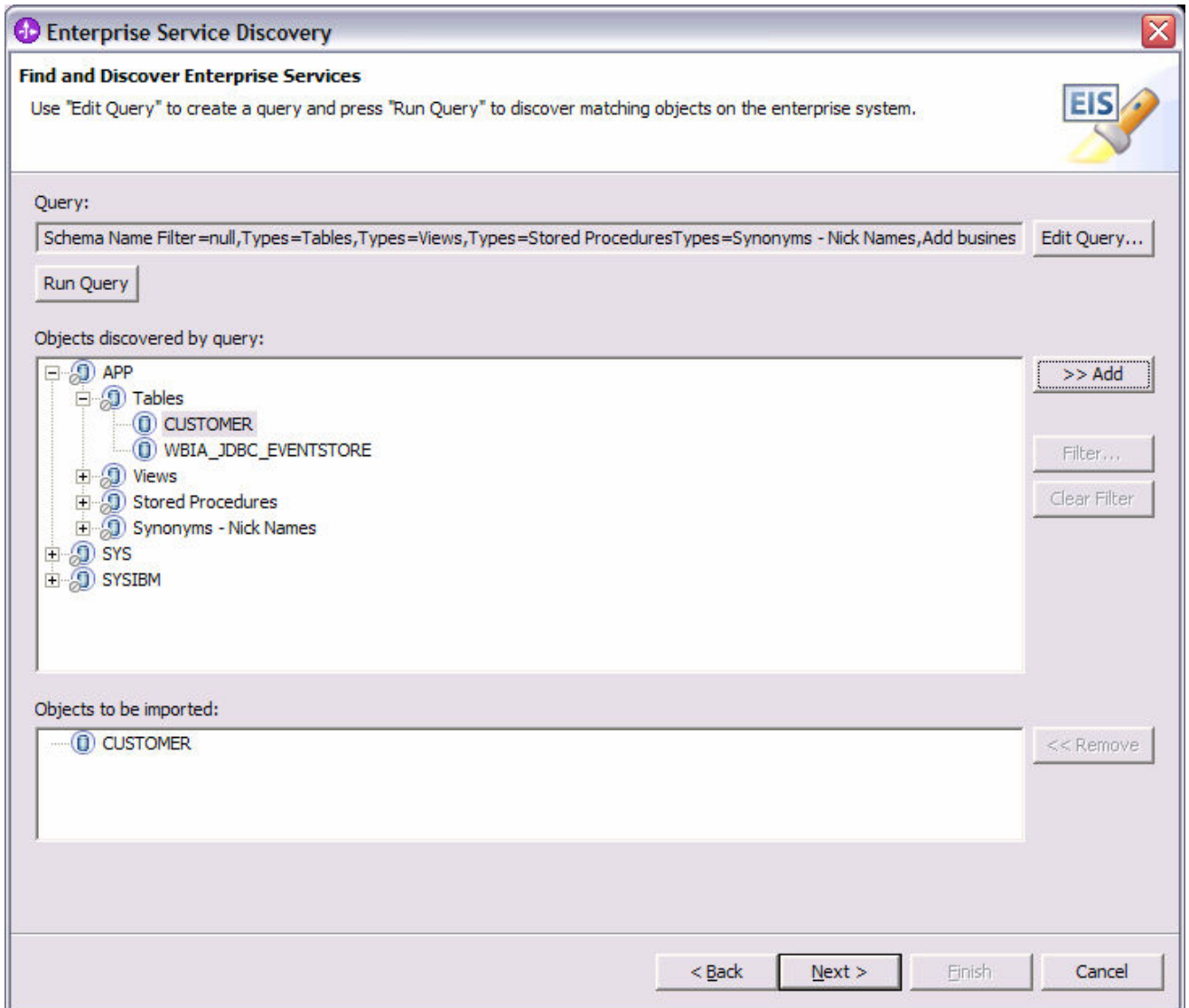
Files\IBM\WebSphere\ID\6.0\runtimes\bi_v6\cloudscape\databases\JDBCTEST

JdbcDriverClass: **com.ibm.db2j.jdbc.DB2jDriver**

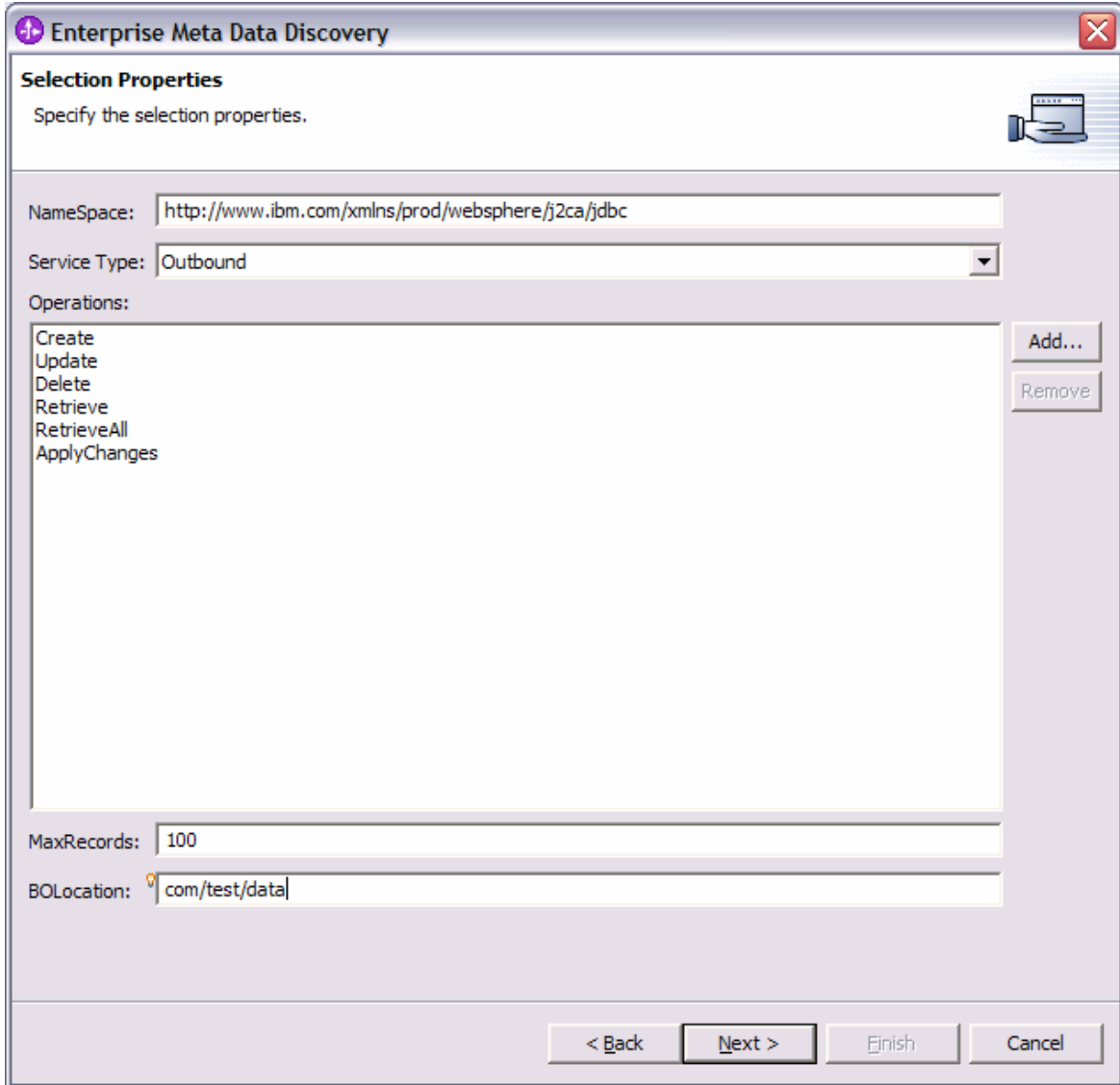
___ c. Click **Next**



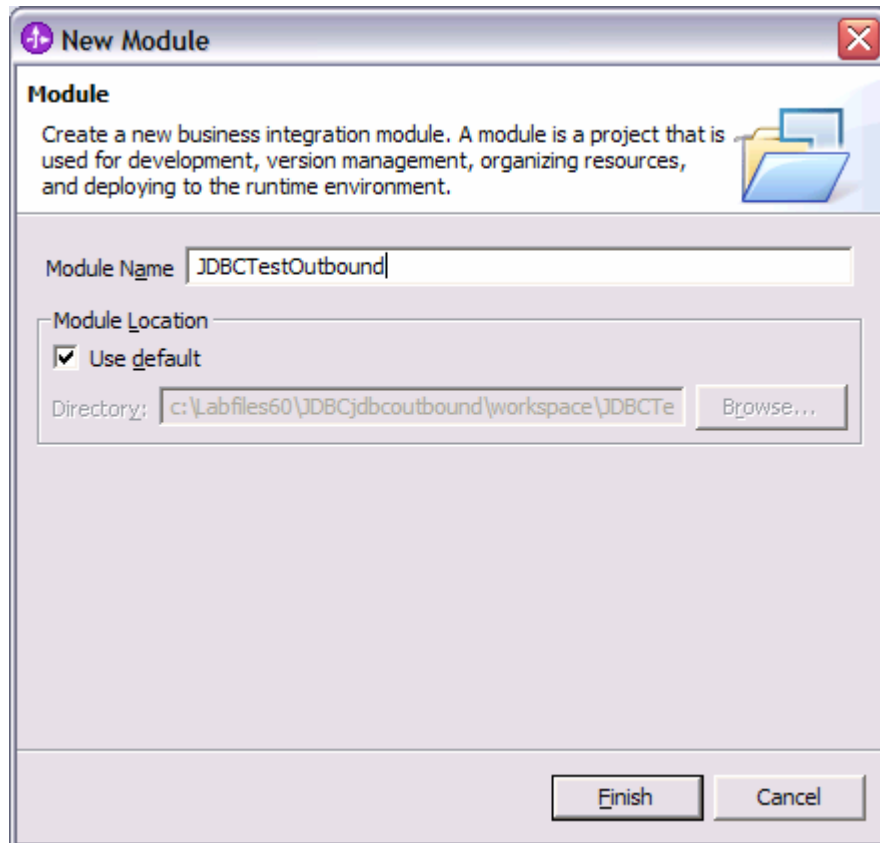
- ___ 6. Complete the **Find and Discover Enterprise Services** panel
 - ___ a. Select the **Run Query** button. A connection will be made to the Cloudscape JDBCTEST database and a selection of Meta data objects will be presented in a tree-like structure.
 - ___ b. Expand the schema named **APP**, (this is the default schema given to our Cloudscape tables) expand **Tables**, highlight **CUSTOMER**, click the >> **Add selected** button. **CUSTOMER** now appears in the **Selected meta data objects** window.
 - ___ c. Click **Next**



- ___ d. On the **Configure Objects** panel, leave the default value for Namespace, change the Service Type: to **Outbound**, and enter **com/test/data** for **BOLocation**. Note the Operations available.
- ___ e. Click **Next**.



- ___ 7. Complete the **Generate Artifacts** Panel.
 - ___ a. A Business Integration Module has not yet been created, select the **New...** button and enter in the name **JDBCTestOutbound** for the Module Name.
 - ___ b. Click **Finish**.



- ___ c. Enter `com/test/data` for the Folder value. Leave the default **JDBCOutboundInterface** for the Name value.
- ___ d. Leave the **Deploy connector with module** box checked.
- ___ e. Select the radio button to the left of **Use discovered connection properties**. Additional properties options appear to complete for ManagedConnectionProperties, ResourceAdapterProperties, and Miscellaneous properties.

Note: The IBM WebSphere Adapters (this includes the IBM WebSphere Adapter for Flat Files, IBM WebSphere Adapter for JDBC, IBM WebSphere Adapter for PeopleSoft Enterprise, IBM WebSphere Adapter for Siebel Business Applications, and IBM WebSphere Adapter for SAP Applications) are supported as “Deploy connector with module” only, meaning the adapter is deployed within the module which is packaged as an Enterprise Archive file (EAR file). Therefore, the “Deploy connector with module” checkbox should always be checked and the “Use discovered connection properties” checkbox should always be checked.

- ___ f. Enter the following values and click **Finish**. Wait for the workspace to complete building.
 J2C Authentication Data Entry: **widNode/jdbc/SampleAlias** (This is the authentication alias you will create on the WebSphere Process Server in the next part of the lab.)
 UserName: **valid value** to connect to the database
 Password: **valid value** to connect to the database
 DatabaseVendor: **Other** (Enter Other since you are using Cloudscape. If you were using DB2, Oracle, or MSSQLServer, you would enter those values instead as specific adapter processing is available with those specific databases.)

Use discovered connection properties

J2C Authentication Data Entry:

ManagedConnectionProperties

UserCredentials

UserName:

Password:

MachineCredentials

DatabaseURL: *

JdbcDriverClass: *

ResourceAdapterProperties

Logging and Tracing

Adapter ID: *

Log File Size:

Log File Name:

Number Of Log Files:

Trace File Size:

Trace File Name:

Number Of Trace Files:

Miscellaneous

AutoCommit

PingQuery:

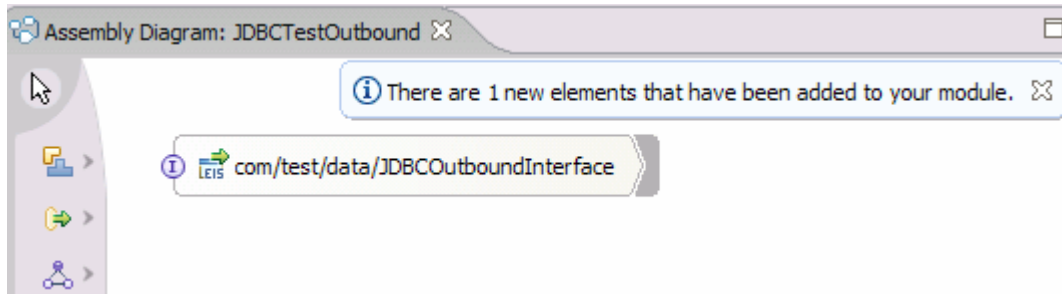
EventTableName:

DatabaseVendor: *

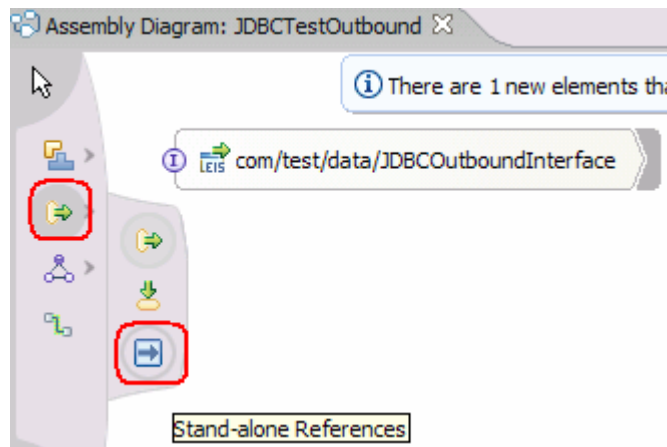
QueryTimeOut:

ReturnDummyBOForSP:

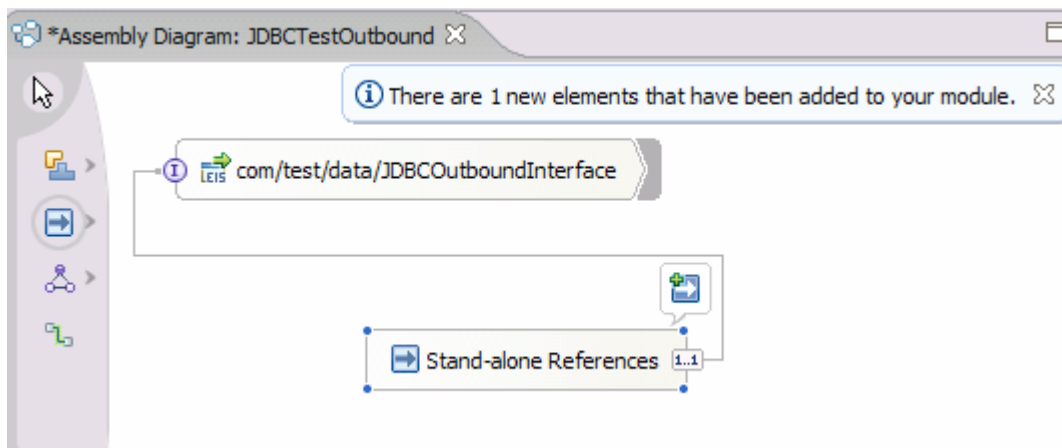
- ___ 8. Use the Assembly Diagram to wire a standalone reference to the com/test/data/JDBCOutboundInterface.
- ___ a. From the **Business Integration** view, expand the JDBCTestOutbound folder, and double click the **JDBCTestOutbound** module. This will open the module in the Assembly Diagram. You will see a message that there is one new element added to the module.



- ___ b. From the palette, select the import icon, then select the Standalone Reference icon and place it on the Assembly Diagram.



- ___ c. Wire the Standalone Reference to the **JDBCOutboundInterface**. At the Add Wire popup window, select **OK** in response to the "Matching reference will be created on the source node. Would you like to continue?" At the second Add Wire popup window, select **No** in response to "Would you like to convert the WSDL interfaces ...". The Assembly Diagram should now look as follows:

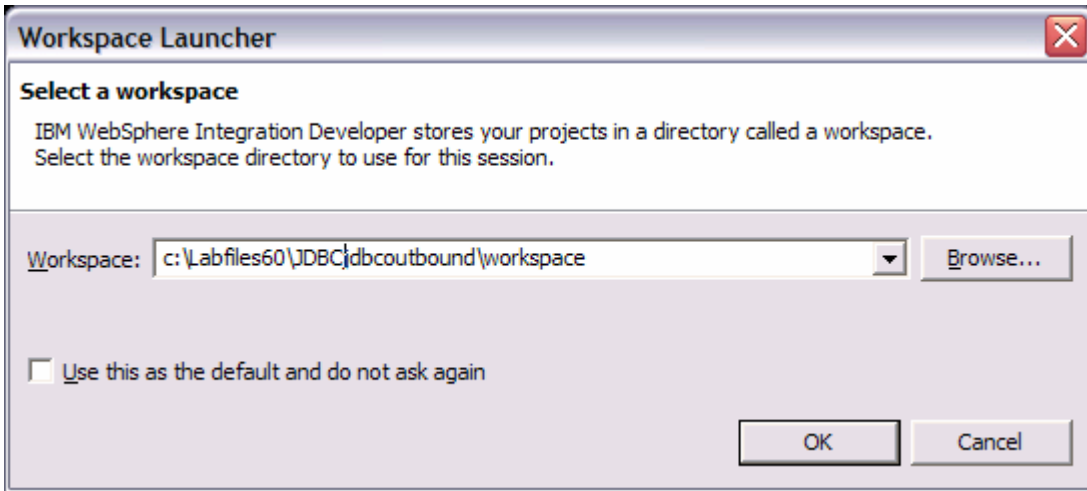


- ___ d. Save your work by selecting **File -> Save** from the top menu, or using the shortcut key sequence **Ctrl + S**.
- ___ e. Wait for the workspace build to complete.

9. Release the connection to the Cloudscape database by using **Switch Workspace**

Note: Switch Workspace is a way to release the existing connection to the Cloudscape database from the Enterprise Service Discovery 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 we are using the Cloudscape embedded driver in this exercise which supports a connection from a single JVM.

- a. From the top level menu, select **File > Switch Workspace** and select the same workspace in which you've been working.

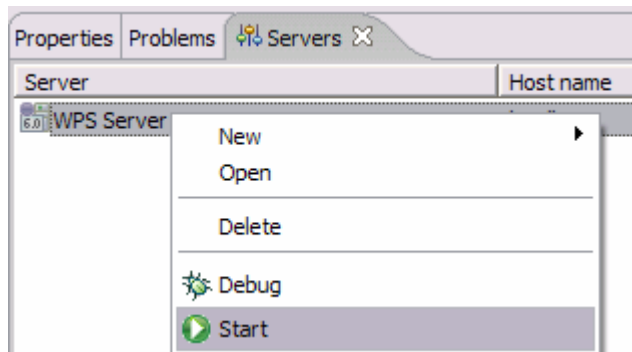


Part 4: Test the application using the WebSphere Test Environment (WTE) and Component Test

In this part you will create a J2C Authentication Alias which is required for connection to the database. You will then use the WebSphere Test Environment and Component Test to test the SCA application by creating and retrieving several records from the CUSTOMER table in the JDBCTEST database.

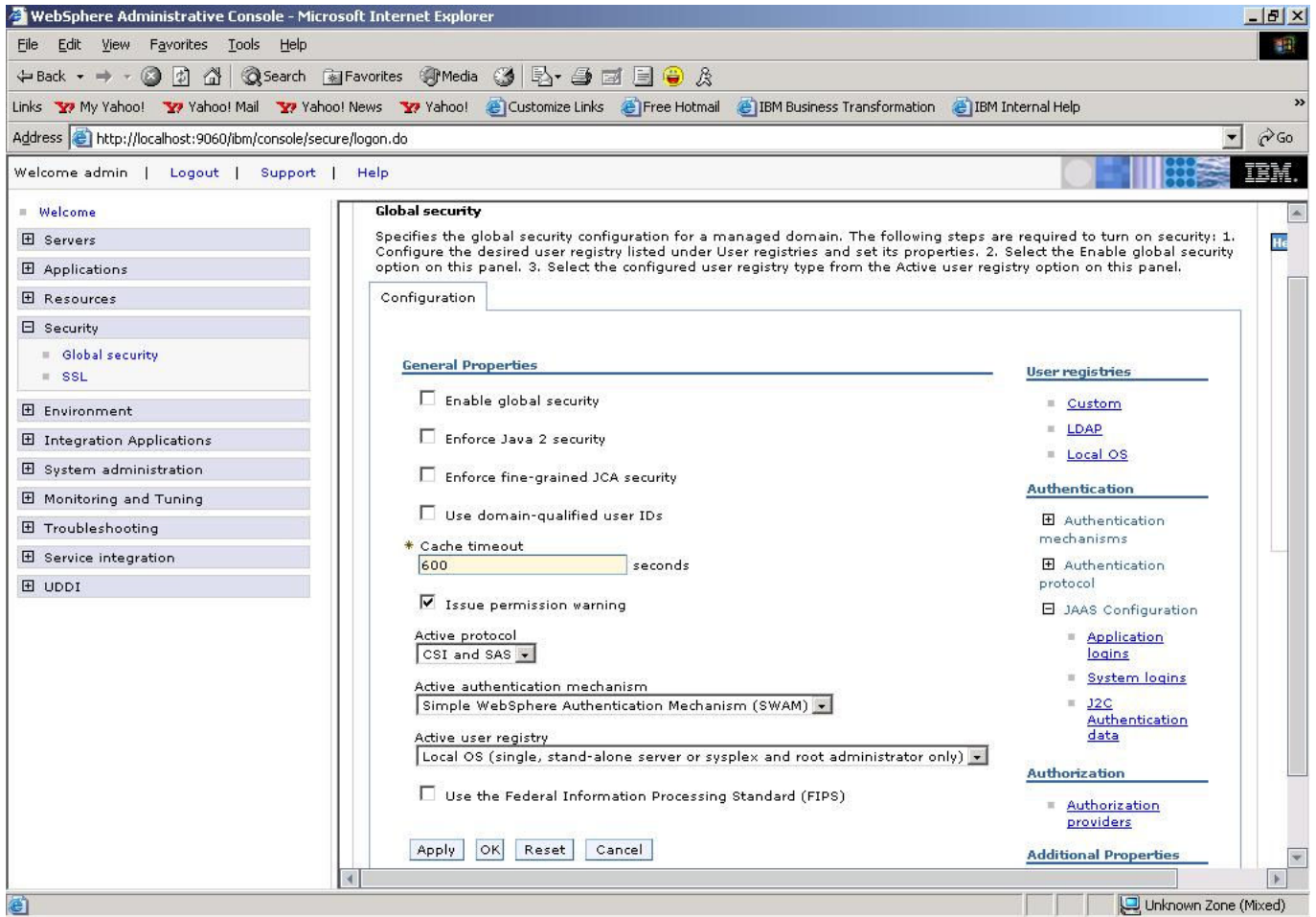
Note: This lab is using the Cloudscape embedded database driver which allows a connection from only a single JVM at a time. Cview.bat, the Enterprise Service Discovery process, and a running Server configured with an application that connects to the database, each are an active connection. This means 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 will not be able to use Cview.bat to view the database, while the server is active. You would need to first stop the server.

- ___ 1. Prior to deploying the application to the server, a J2C Authentication Alias definition must be created on the server. Start the WPS Server and create an Authentication Alias named jdbc/SampleAlias.
 - ___ a. In the Business Integration view, servers panel, right click and **Start** the WPS server v6.0. Wait for the server to start.



- ___ b. Open a browser to the url: <http://localhost:9060/ibm/console> to start the Administrative console and click the **Login** button.

- c. On the left side of the console, expand **Security** select **Global security**. On the right side of the console, expand **JAAS Configuration** under the Authentication heading



__ d. Click on the J2C Authentication data link. Click **New**.

The screenshot shows the WebSphere Administrative Console interface. The main content area displays the 'Global security' configuration page for J2C authentication data entries. The page includes a navigation pane on the left, a main content area with a table of entries, and a help pane on the right.

Global security > J2EE Connector Architecture (J2C) authentication data entries
 Specifies a list of user IDs and passwords for Java 2 connector security to use.

Preferences

Select	Alias	User ID	Description
<input type="checkbox"/>	SCA Auth Alias	administrator	This is the alias used by SCA to login to a secured SIBus
<input type="checkbox"/>	shaiiNode01/CommonEventInfrastructureJMSAuthAlias	admin	Authentication alias for the Common Event Infrastructure JMS Topics and Queues
<input type="checkbox"/>	shaiiNode01/Cell/shaiiNode01/server1/EventAuthDataAliasCloudScape	none	CloudScape authentication alias for the Common Event Infrastructure

Total 3

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](#)

- ___ e. Enter an alias name of **jdbc/SampleAlias**. Enter a user id and password that can connect to the database. Click **OK**.

Global security > J2EE Connector Architecture (J2C)
Specifies a list of user IDs and passwords for Java 2

Configuration

General Properties

* Alias
jdbc/SampleAlias

* User ID
WSDemo

* Password

Description

Apply OK Reset Cancel

- ___ f. Notice the Node name of widNode has been prepended to the alias name. Save the Changes. Click **Save**. Click **Save** again.

Global security

Global security

Messages

- ⚠ Changes have been made to your local configuration. Click [Save](#) to apply changes to the master configuration.
- ℹ The server may need to be restarted for these changes to take effect.
- ⚠ The master configuration has been updated. You currently have workspace conflicts with these modifications. To see these updates you must [Save or discard](#) your current workspace modifications

Global security > J2EE Connector Architecture (J2C) authentication data entries
 Specifies a list of user IDs and passwords for Java 2 connector security to use.

Preferences

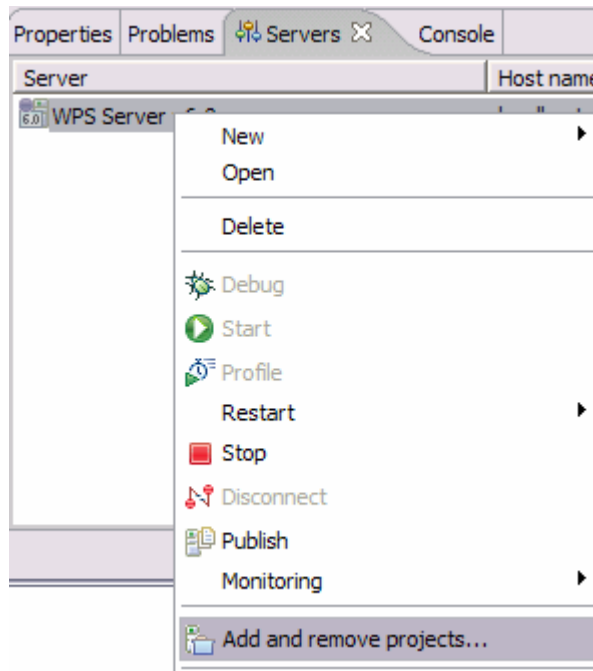
Select	Alias	User ID	Description
<input type="checkbox"/>	SCA Auth Alias	wid	This is the alias used by SCA to login to a secured SIBus
<input type="checkbox"/>	widCell/BPEAuthDataAliasJMS widNode_server1	wid	Authorization Alias for Process Choreographer
<input type="checkbox"/>	widCell/widNode/server1/EventAuthDataAliasCloudScape	none	CloudScape authentication alias for the Common Event Infrastructure
<input type="checkbox"/>	widNode/CommonEventInfrastructureJMSAuthAlias	wid	Authentication alias for the Common Event Infrastructure JMS Topics and Queues
<input type="checkbox"/>	widNode/jdbc/SampleAlias	WSDemo	

Total 5

- ___ g. Log Out of the Administrative Console.

- ___ 2. Add the project to the server for the WebSphere Test Environment.

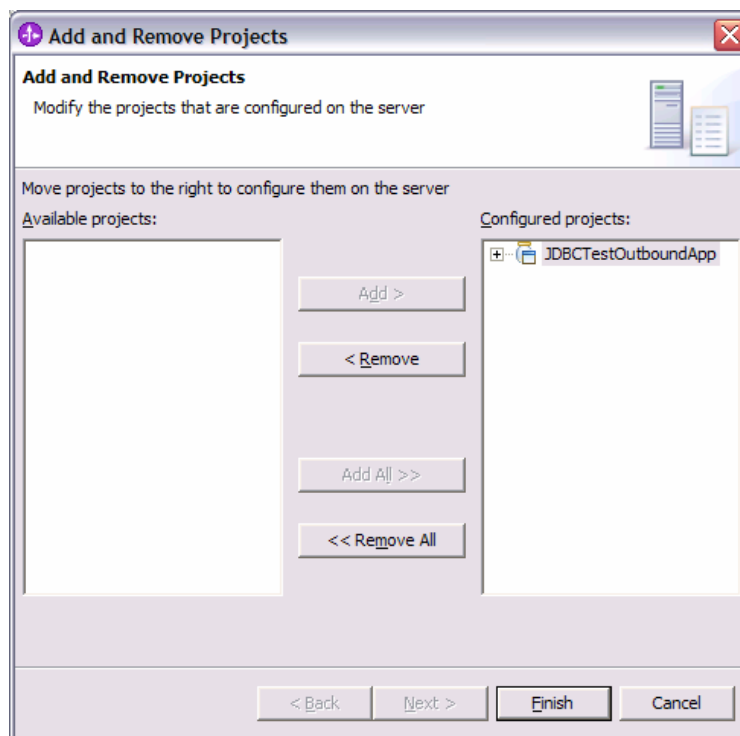
___ a. Right click on the server in the server view and select **Add and remove projects ...**



___ b. In the Add and Remove Projects dialog, select the **JDBCTestOutboundApp** project from the Available projects panel

___ c. Click **Add >** to add it to the Configured projects panel.

___ d. Click **Finish**



- ___ e. Wait for the project to be added. In the Console view, you will see a message that the application has successfully started.

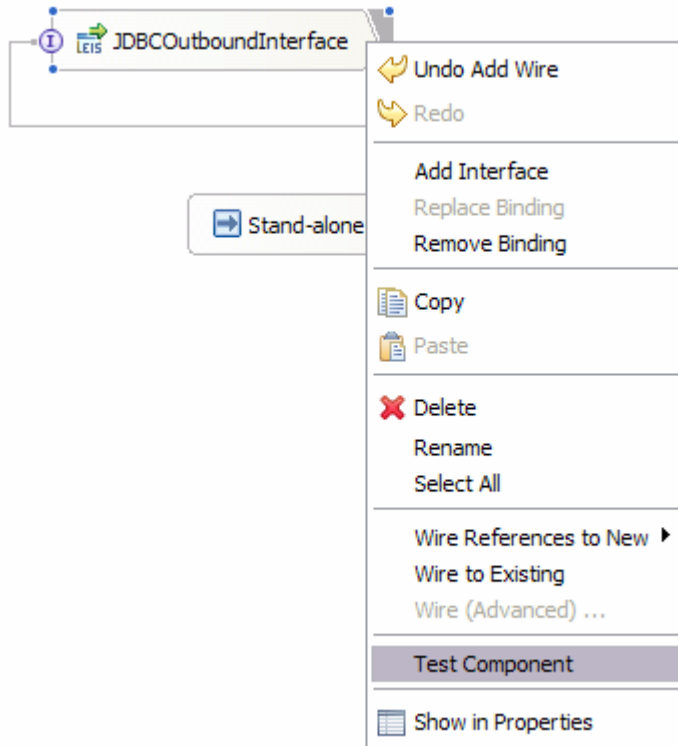
CWSCA3009I: The SCA module JDBCTestOutboundApp is starting.

CWSCA3010I: The SCA module started successfully.

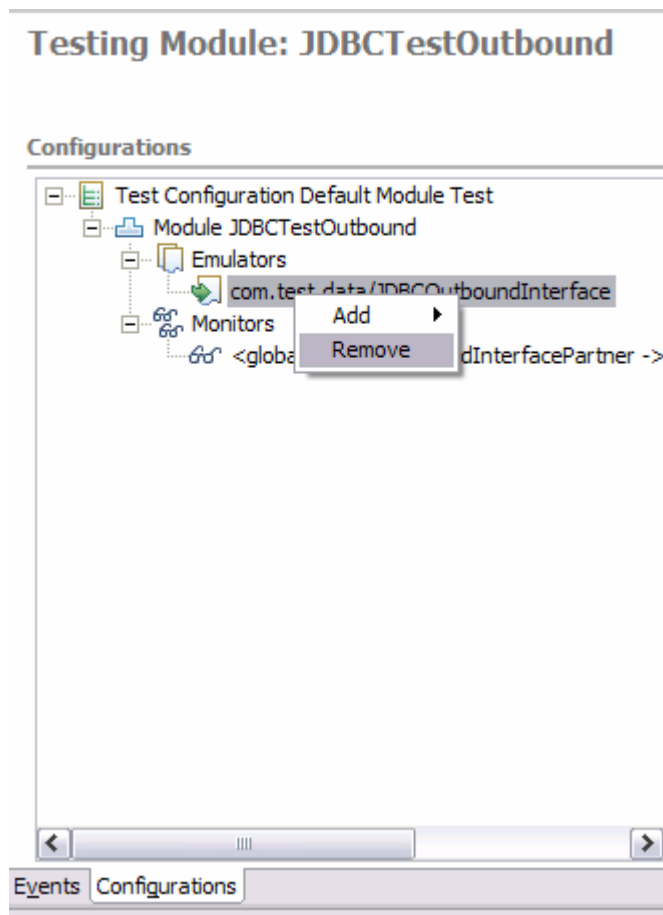
WSVR0221I: Application started: JDBCTestOutboundApp

- ___ 3. Use the Test Component to test the application.

- ___ a. In the Assembly Diagram, right click on the **com/test/data/JDBCOutboundInterface** import and select **Test Component**



- ___ b. Switch to the **Configurations** tab, if you see **com/test/data/JDBCOutboundInterface** underneath **Emulators**, right click, and remove it. You may not see anything under Emulators. We want to connect to and drive the real JDBCTEST database and not have Test Component emulate for us.



- c. Switch back to the **Events** tab. Under **Detailed Properties, Initial request parameters** , specify a Verb of **Create**, and specify a unique id, fname, and lname (all strings) (Do not use an id of 1 or 2 as two rows have already been created in the Customer table as part of the JCAAdapter.bat file that created the database and tables for you.) Select **Continue**

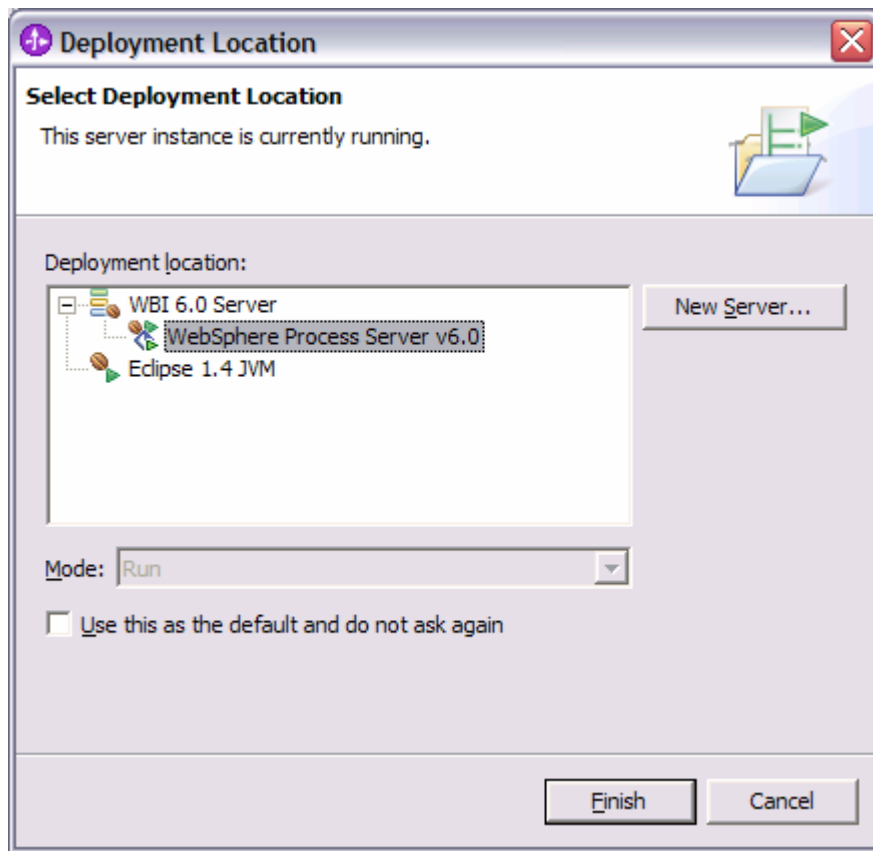
General Properties
 Detailed Properties

Configuration: Default Module Test
Module: JDBCTestOutbound
Component: JDBCOutboundInterface
Interface: JDBCOutboundInterface
Operation: createAppCustomer

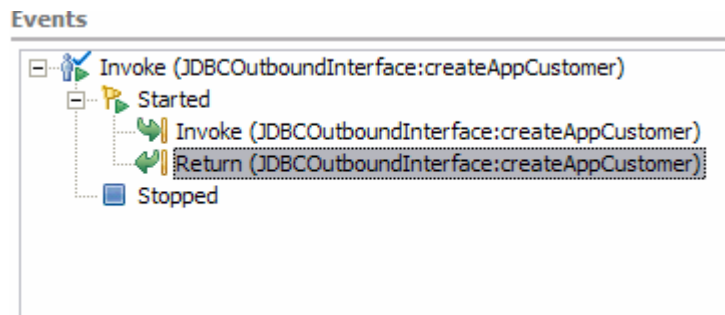
Initial request parameters

Name	Type	Value
<input type="checkbox"/> createAppCustom...	AppCustomerBG	
verb	String	Create
<input type="checkbox"/> AppCustomer	AppCustomer	
id	string	33
lname	string	Smith
fname	string	Mary

- ___ d. In the **Choose a deployment location dialog**, select the **WebSphere Process Server v6.0** server. Select **Finish**

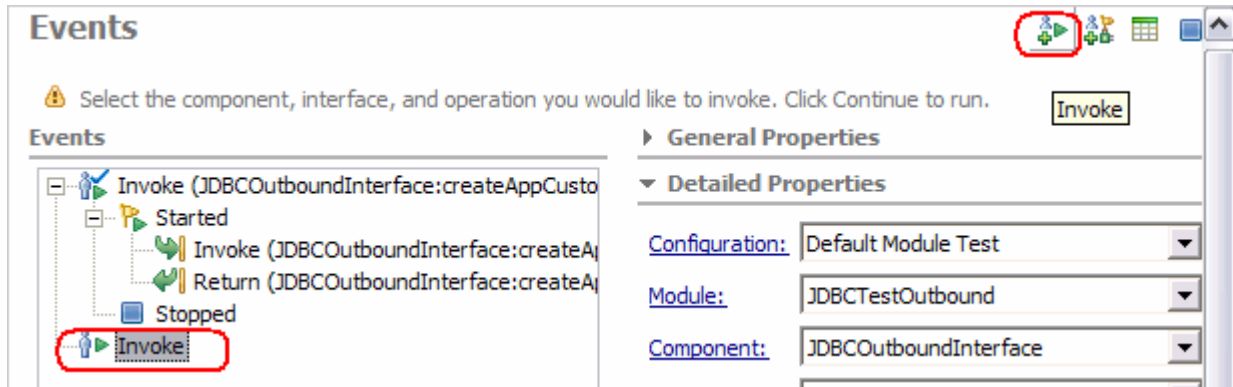


- ___ e. In the Events window you will see that the invoke has returned.

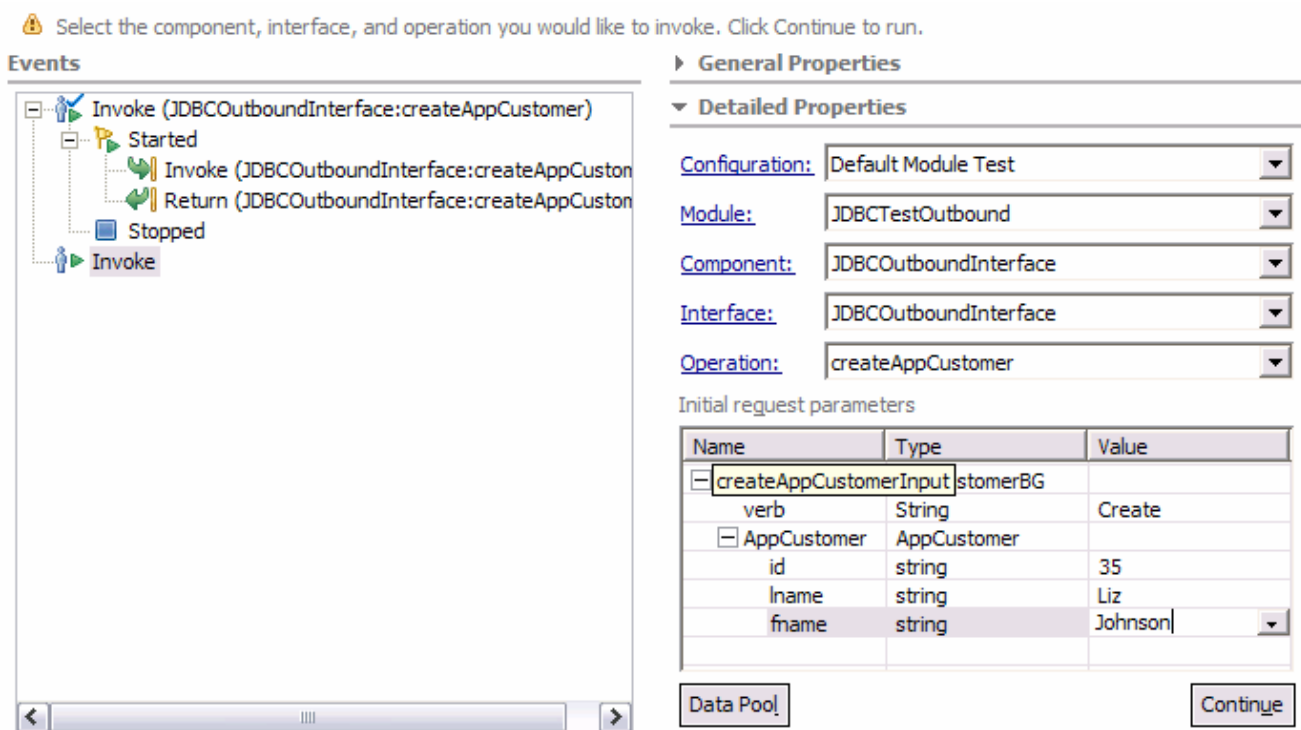


- ___ 4. Create a second entry in the CUSTOMER table

- ___ a. In the upper right corner of the **Events** panel of Test Component, click the **Invoke** button. This will cause another **Invoke** event to appear within the Events window.



- ___ b. In the Initial request parameters, specify a verb of Create, and a unique id, lname, and fname. Click **Continue**



- ___ 5. To test a retrieve, select the **Invoke** button in the top right corner again, under **Detailed Properties**, select the Operation **RetrieveAppCustomer**. Under **Initial request parameters** set the **Verb** to **<null>**, enter a value for id of one of the previously created customers. Click **Continue**

General Properties
Detailed Properties

Configuration: Default Module Test
Module: JDBCTestOutbound
Component: JDBCOutboundInterface
Interface: JDBCOutboundInterface
Operation: retrieveAppCustomer

Initial request parameters

Name	Type	Value
<input type="checkbox"/> retrieveAppCusto...	AppCustomerBG	
verb	String	<null>
<input type="checkbox"/> AppCustomer	AppCustomer	
id	string	33
lname	string	
fname	string	

- ___ a. Upon return, the values matching the id specified, should be displayed in the Return parameters.

Detailed Properties

Module: [JDBCTestOutbound](#)
Component: [JDBCOutboundInterface](#)
Interface: [JDBCOutboundInterface](#)
Operation: [retrieveAppCustomer](#)

Return parameters:

Name	Type	Value
<input type="checkbox"/> retrieveAppCu...	AppCustomerBG	
verb	VerbType	<null>
<input type="checkbox"/> AppCustomer	AppCustomer	
id	String	33
lname	String	Mary
fname	String	Smith

- ___ b. To test a retrieveAll, select the Invoke button, then under **Detailed Properties**, select the Operation **retrieveAllAppCustomer**, set the Verb, id, lname and fname to <unset> by clicking on the field and selecting the <unset> from the dropdown menu. Click **Continue**.

General Properties
 Detailed Properties

Configuration: Default Module Test
Module: JDBCTestOutbound
Component: JDBCOutboundInterface
Interface: JDBCOutboundInterface
Operation: retrieveallAppCustomer

Initial request parameters

Name	Type	Value
<input type="checkbox"/> retrieveallAppCust...	AppCustomerBG	
verb	String	<unset>
<input type="checkbox"/> AppCustomer	AppCustomer	
id	string	<unset>
lname	string	<unset>
fname	string	<unset>

- ___ c. Upon return, the values for the customers existing in the database should be displayed.

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

- ___ 6. Exit the Test Component panel, remove the JDBCTestOutbound project from the server, and stop the server.

What you did in this exercise

This lab introduced you to the WebSphere Adapter for JDBC and the processing of outbound requests to a table in a database. You used WebSphere Integration Developer to import the adapter and run the Enterprise Service Discovery wizard to input connection information, create a service description, and discover objects existing in the specified database. You then assembled an SCA application, wiring together a standalone reference and the EIS import file. You also tested your application, using the WebSphere Test Environment and Component Test, exercising various outbound requests, such as create, delete, retrieve, and retrieveAll.

This page is left intentionally blank.