© Copyright IBM Corporation 2007 All rights reserved

IBM WEBSPHERE ADAPTER FOR JDBC 6.0.2 – 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: Using WebSphere Process Server Administrative Console to create J2C authentication alias and to configure data sources 20

Part 5: Test the application using the WebSphere test environment and component test 29

What you did in this exercise 38

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 V6.0.2 installed
- WebSphere Process Server V6.0.2 test environment installed
- WebSphere Adapter for JDBC V6.0.2 installed
- Sample code in the directory C:\Labfiles602\JDBC (Windows) or /tmp/LabFiles602/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 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 may 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 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\6.0.2	
<WPS_HOME>	<WID_HOME>\runtimes\bi_v6	
<JDBCADAPTER_HOME>	<WID_HOME>\Resource Adapters\JDBC	
<LAB_FILES>	C:\Labfiles602	/tmp/Labfiles60
<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:\LabFiles602\ would be replaced by C:/LabFiles602/

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, and 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 cview.bat.

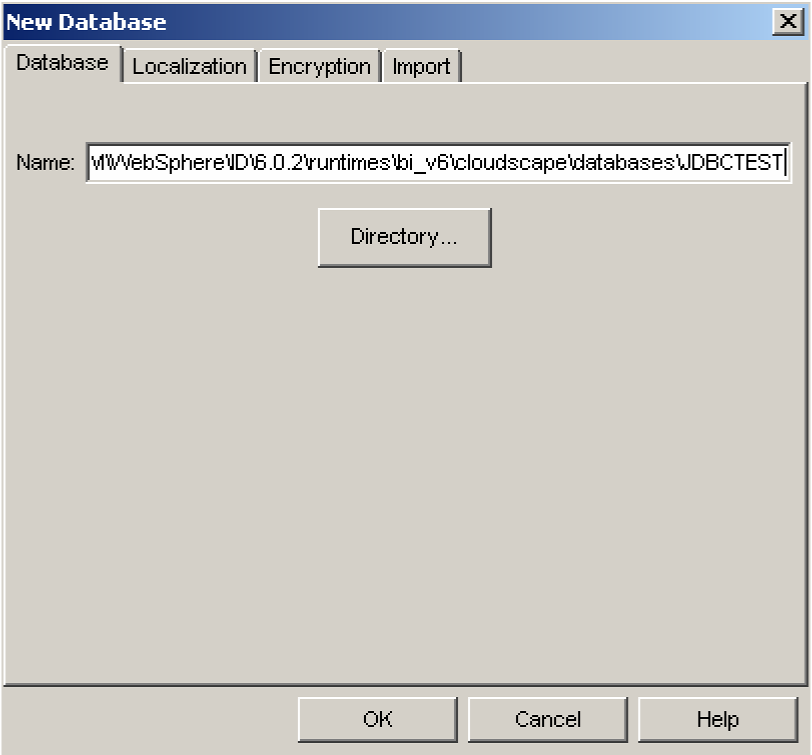
Note: The Cloudscape embedded driver that is being used in the lab, supports a connection from only one JVM at a time. You can have either the server running and connected to the JDBCTEST database, or the Cview GUI connected to the JDBCTEST database; but not both at the same time.

__ a. Open a command prompt window, navigate to the following subdirectory, and run the cview.bat program.

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

____ 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**



Name: M\WebSphere\ID\6.0.2\runtimes\bi_v6\cloudscape\databases\JDBCTEST

____ 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.

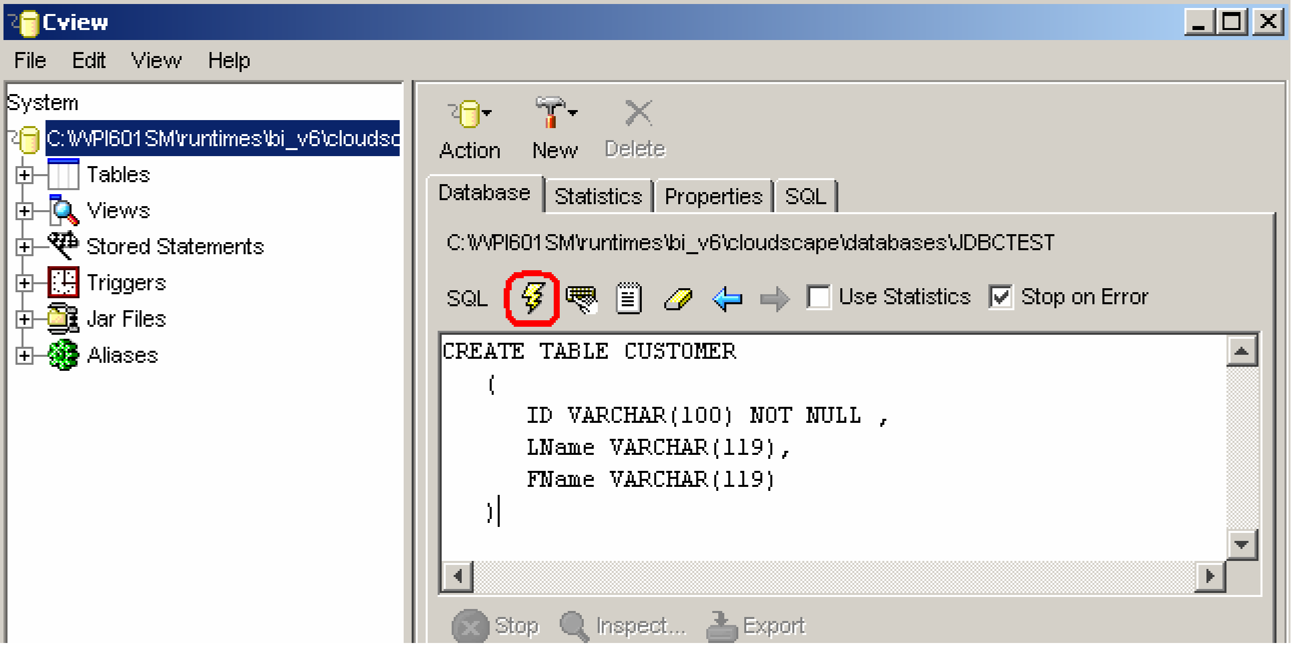
NOTE: For your convenience, the following SQL code snippets can be found in **<LAB_FILES>\JDBC\snippets\CUSTOMERSQL.txt**

__ 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 “lightening bolt” icon to run the SQL (The SQL text will appear highlighted after executing.)

© Copyright IBM Corporation 2007. All rights reserved



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

__ c. Set the key for the CUSTOMER table by pasting the following into the SQL window (you can 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 “lightening bolt” icon to run 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),
    XID VARCHAR (255)
  )
```

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

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

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

__ d. Select the “lightening bolt” icon to run 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 “lightening bolt” to run 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)
```

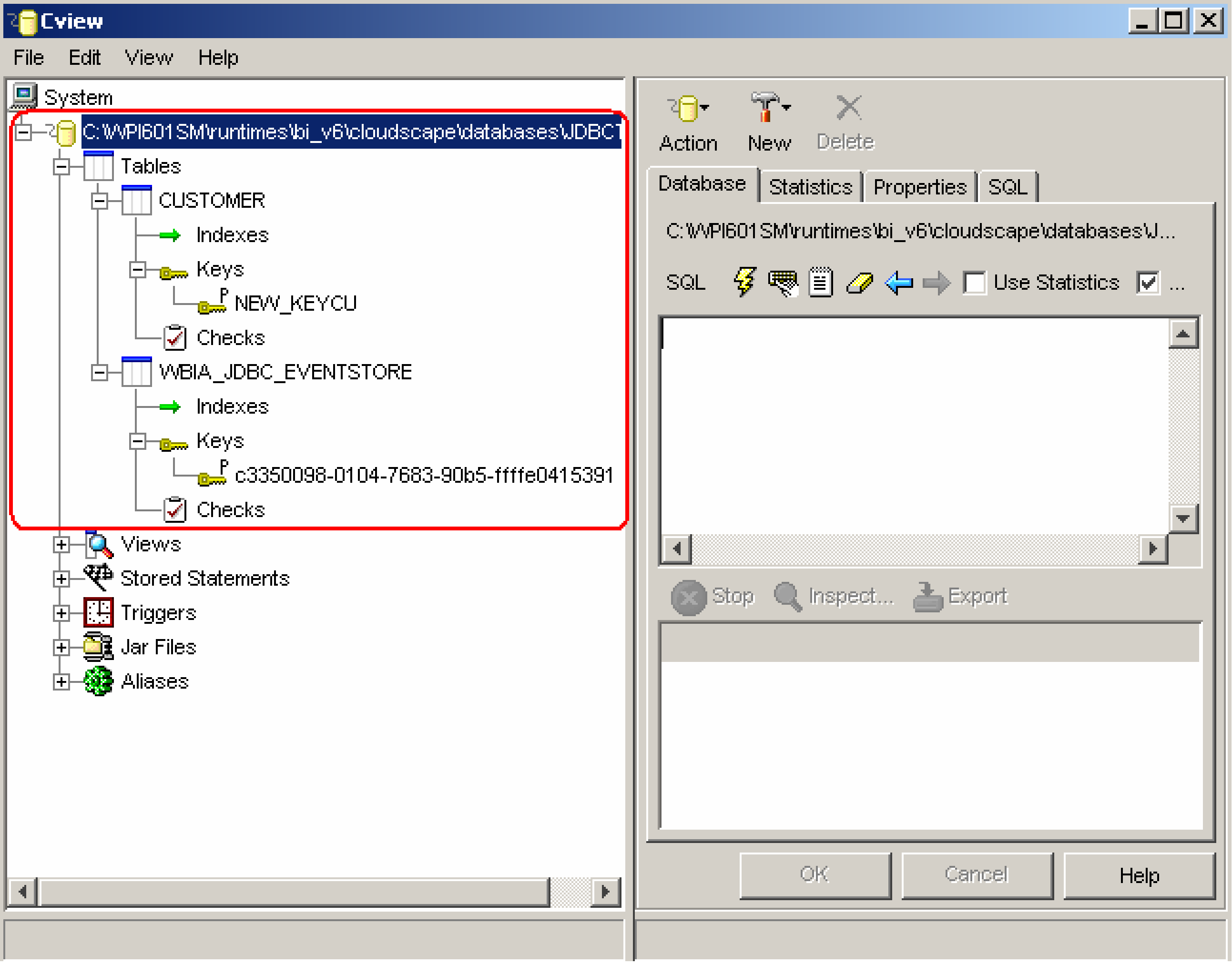
__ d. Select the “lightening bolt” to run the SQL.

__ 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 “lightening bolt” to run 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.

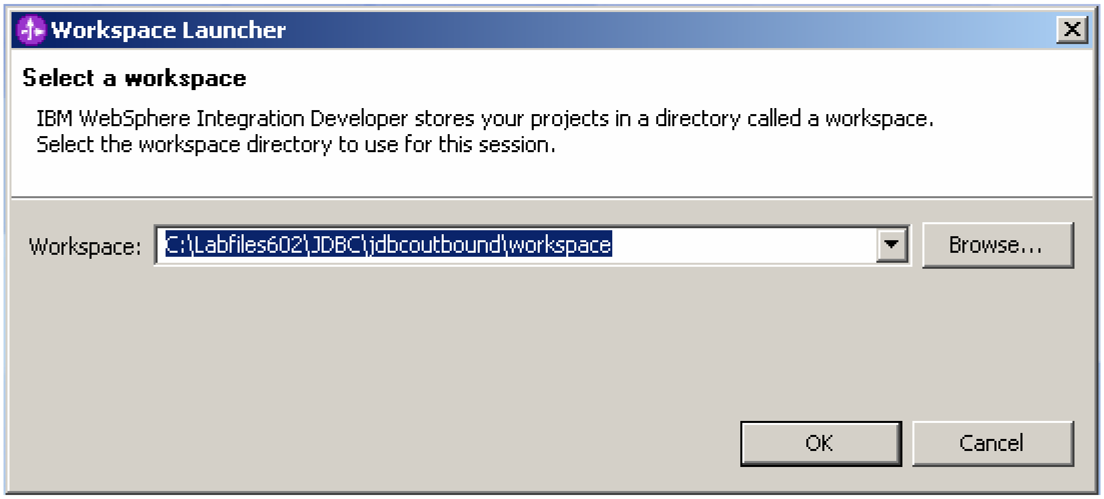
Part 2: Set up the development environment

In this part, you will start WebSphere Integration Developer and set up the WebSphere Process Server to be used as the WebSphere test environment.

____ 1. Start WebSphere Integration Developer V6.0.1 with a new workspace

__ a. From the start menu select **Start > Programs > IBM WebSphere > Integration Developer V6.0.1 > WebSphere Integration Developer V6.0.1**

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



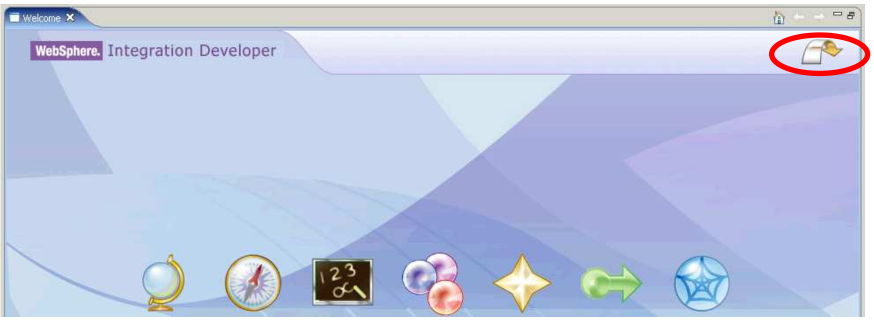
Workspace Launcher

Select a workspace

IBM WebSphere Integration Developer stores your projects in a directory called a workspace.
Select the workspace directory to use for this session.

Workspace: C:\Labfiles602\JDBC\jdbcoutbound\workspace

__ c. When WebSphere Integration Developer V6.0.1 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**

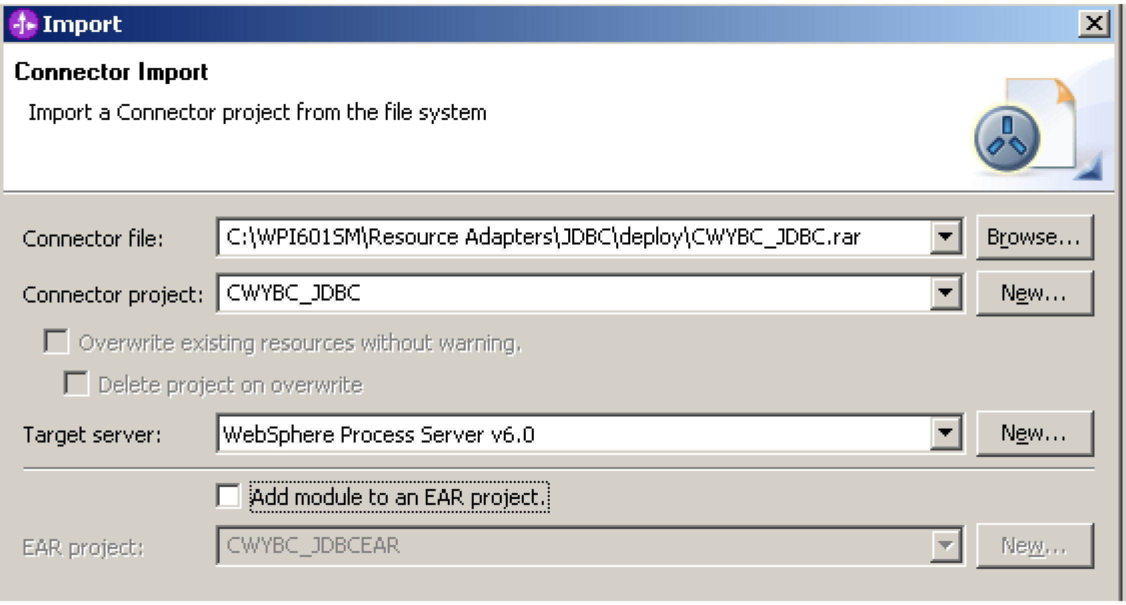
__ b. Select **RAR file,** click **next**

____ 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>WebSphere Adapters\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**.



Import

Connector Import

Import a Connector project from the file system

Connector file: C:\WPI601SM\Resource Adapters\JDBC\deploy\CWYBC_JDBC.rar

Connector project: CWYBC_JDBC

□ Overwrite existing resources without warning.

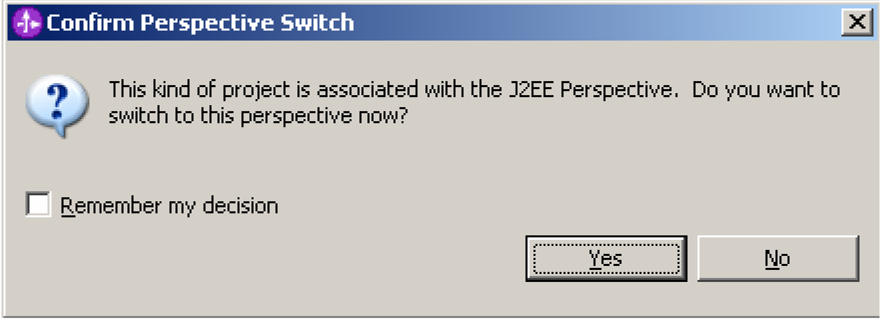
□ Delete project on overwrite

Target server: WebSphere Process Server v6.0

□ Add module to an EAR project.

EAR project: CWYBC_JDBCEAR

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



Confirm Perspective Switch

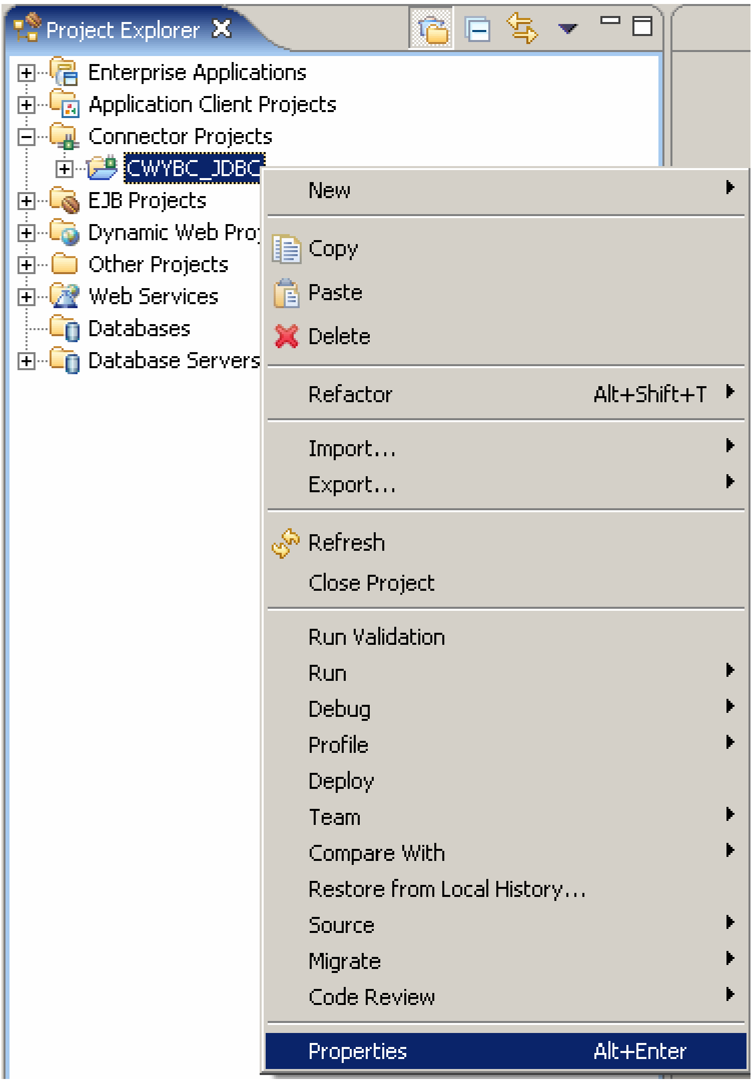
This kind of project is associated with the J2EE Perspective. Do you want to switch to this perspective now?

□ Remember my decision

© Copyright IBM Corporation 2007. All rights reserved

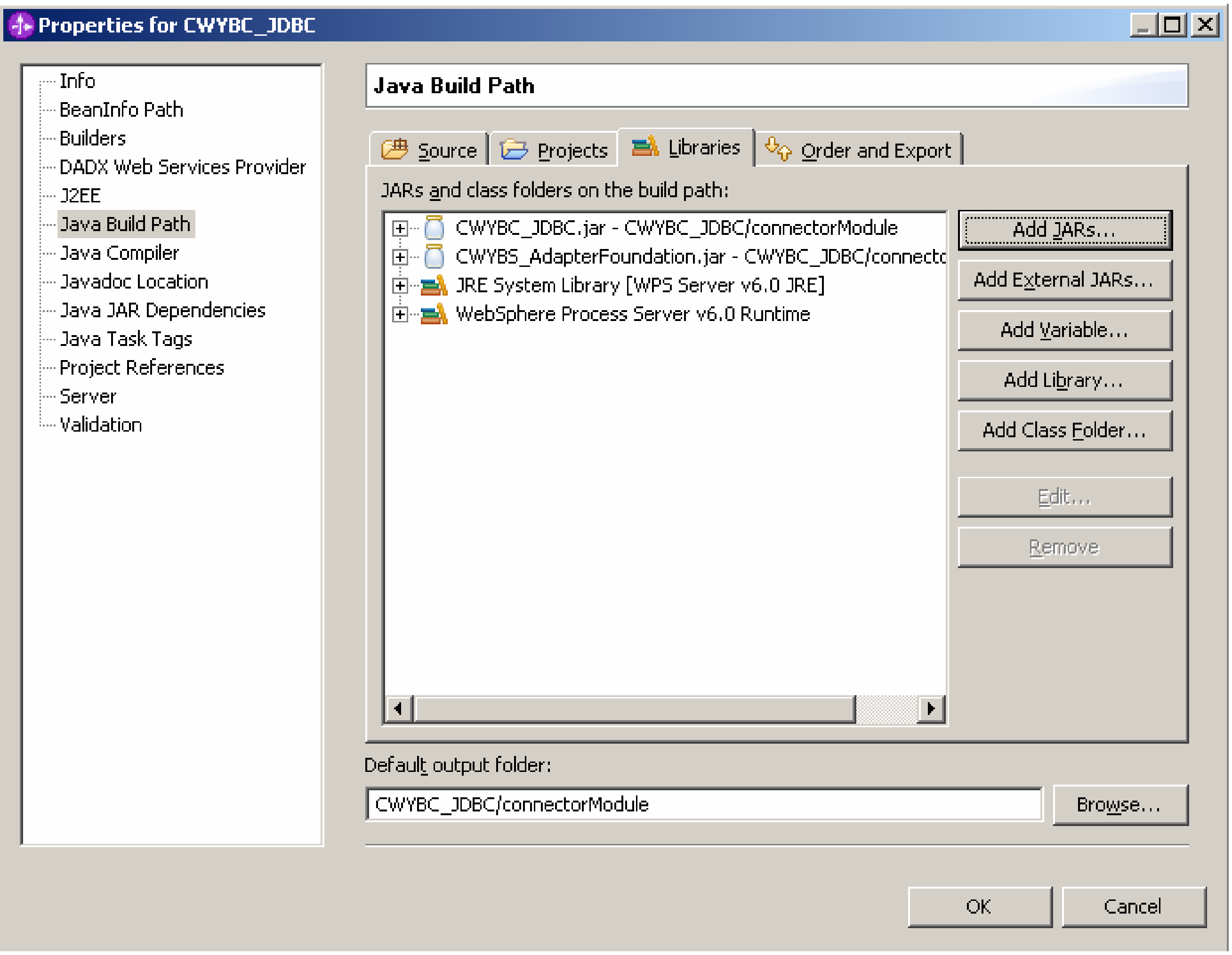
____ 3. Add any external dependencies your adapter has to the imported project. These are dependencies that the adapter may have on the JDBC applications (adapter–specific). Add the db2j.jar to the build path of the Connector project.

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



© Copyright IBM Corporation 2007. All rights reserved

__ 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



Properties for CWYBC_JDBC

- Info
- BeanInfo Path
- Builders
- DADX Web Services Provider
- J2EE
- Java Build Path
- Java Compiler
- Javadoc Location
- Java JAR Dependencies
- Java Task Tags
- Project References
- Server
- Validation

Java Build Path

Source | Projects | Libraries | Order and Export

JARs and class folders on the build path:

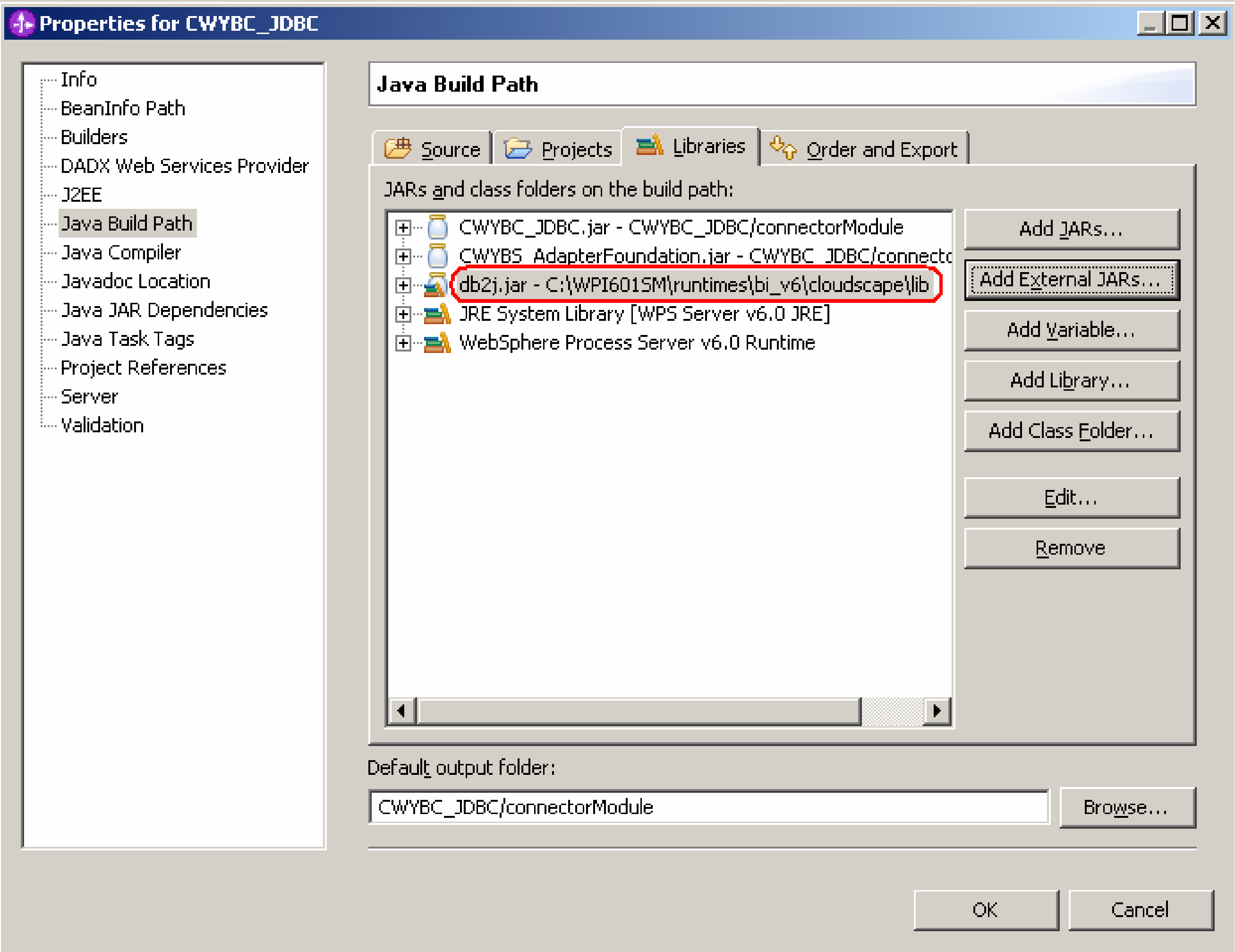
- CWYBC_JDBC.jar - CWYBC_JDBC/connectorModule
- CWYBS_AdapterFoundation.jar - CWYBC_JDBC/connecto
- JRE System Library [WPS Server v6.0 JRE]
- WebSphere Process Server v6.0 Runtime

Default output folder: CWYBC_JDBC/connectorModule

© Copyright IBM Corporation 2007. All rights reserved

__ 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**



JARs and class folders on the build path:

- CWYBC_JDBC.jar - CWYBC_JDBC/connectorModule
- CWYBS_AdapterFoundation.jar - CWYBC_JDBC/connectc
- db2j.jar - C:\WPI601SM\runtimes\bi_v6\cloudscape\lib
- JRE System Library [WPS Server v6.0 JRE]
- WebSphere Process Server v6.0 Runtime

Default output folder: CWYBC_JDBC/connectorModule

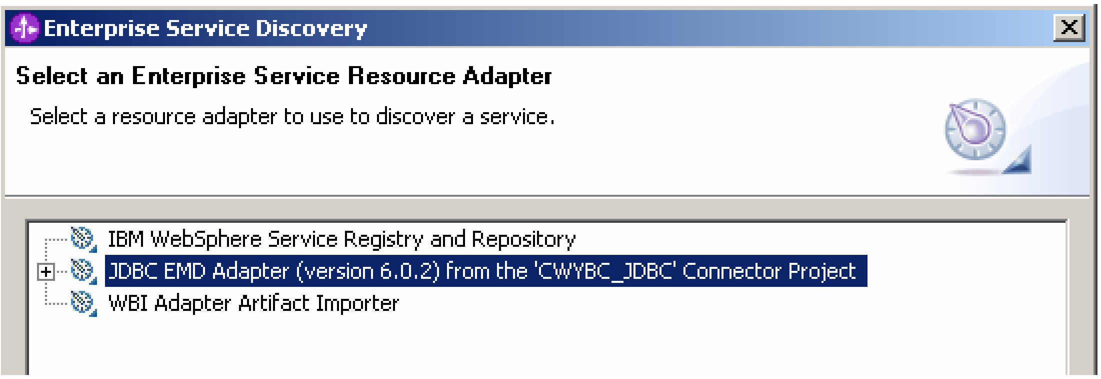
____ 4. Switch to the Business Integration Perspective and run the Enterprise Service Discovery wizard. A Business Integration project will be created for you during this process.

__ 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**

© Copyright IBM Corporation 2007. All rights reserved

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



Select an Enterprise Service Resource Adapter

Select a resource adapter to use to discover a service.

- IBM WebSphere Service Registry and Repository
- JDBC EMD Adapter (version 6.0.2) from the 'CWYBC_JDBC' Connector Project
- WBI Adapter Artifact Importer

____ 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 user name and password values, for example

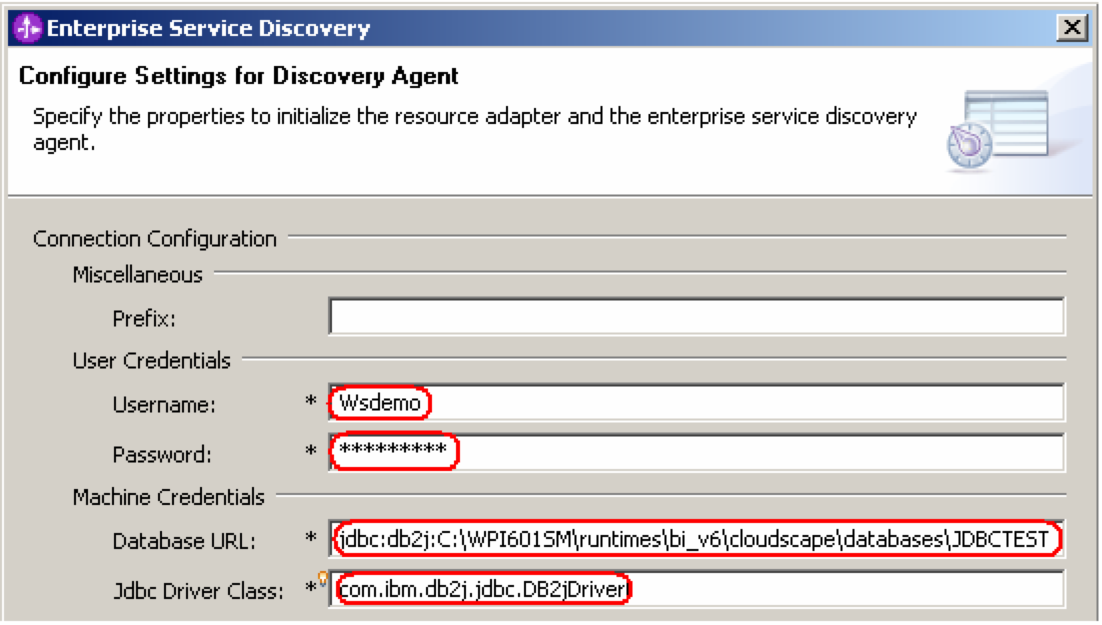
Username: **WSdemo**
Password: **WS15Demo1**

__ b. Enter the following values for DatabaseURL and JdbcDriverClass

DatabaseURL: jdbc:db2j:c:\Program Files\IBM\WebSphere\ID\6.0.1\runtimes\bi_v6\cloudscape\databases\JDBCTEST

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

__ c. Click **Next**



Configure Settings for Discovery Agent

Specify the properties to initialize the resource adapter and the enterprise service discovery agent.

Connection Configuration

Miscellaneous

Prefix:

User Credentials

Username: * Wsdemo

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

Machine Credentials

Database URL: * jdbc:db2j:C:\WPI601SM\runtimes\bi_v6\cloudscape\databases\JDBCTEST

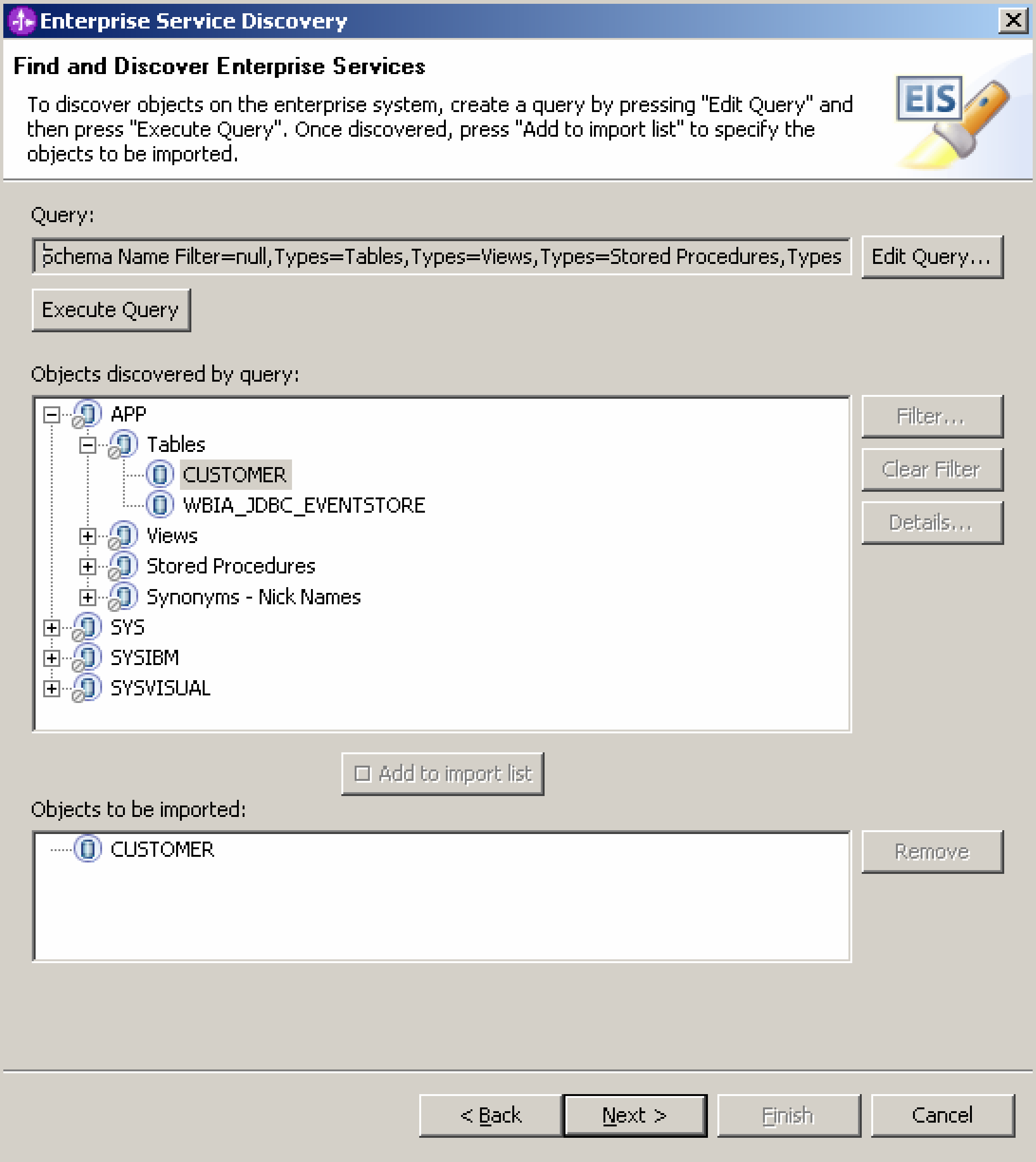
Jdbc Driver Class: * com.ibm.db2j.jdbc.DB2jDriver

____ 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 the Cloudscape tables) expand **Tables**, highlight **CUSTOMER**, click the **>> Add to import List** button. **CUSTOMER** now appears in the **Objects to be imported** window.

__ c. Click **Next**


Enterprise Service Discovery

Find and Discover Enterprise Services

To discover objects on the enterprise system, create a query by pressing "Edit Query" and then press "Execute Query". Once discovered, press "Add to import list" to specify the objects to be imported.

Query: Schema Name Filter=null,Types=Tables,Types=Views,Types=Stored Procedures,Types

Objects discovered by query:

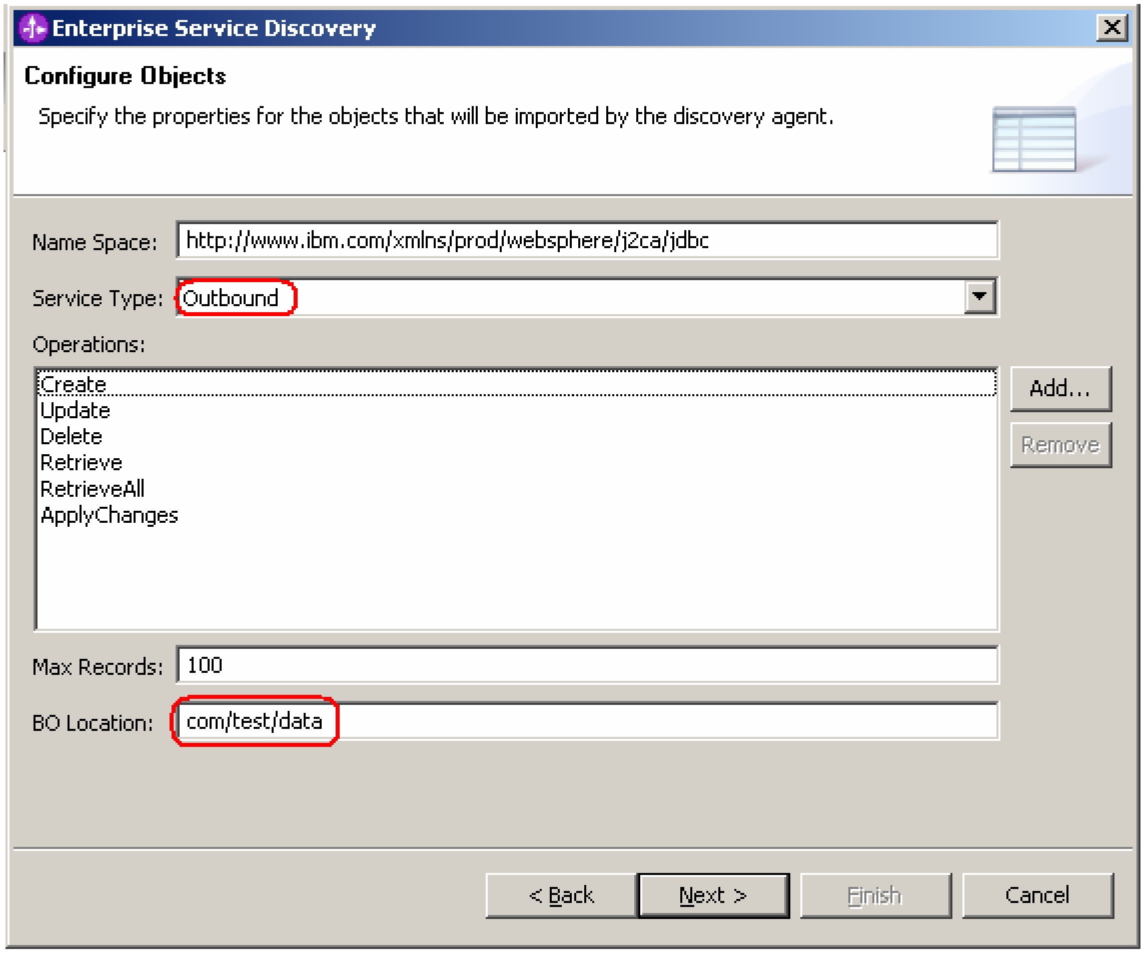
```
APP
  Tables
    CUSTOMER
    WBIA_JDBC_EVENTSTORE
  Views
  Stored Procedures
  Synonyms - Nick Names
SYS
SYSIBM
SYSVISUAL
```

Objects to be imported:

```
CUSTOMER
```

__ **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.**



Enterprise Service Discovery

Configure Objects

Specify the properties for the objects that will be imported by the discovery agent.

Name Space: http://www.ibm.com/xmlns/prod/websphere/j2ca/jdbc

Service Type: Outbound

Operations:
Create
Update
Delete
Retrieve
RetrieveAll
ApplyChanges

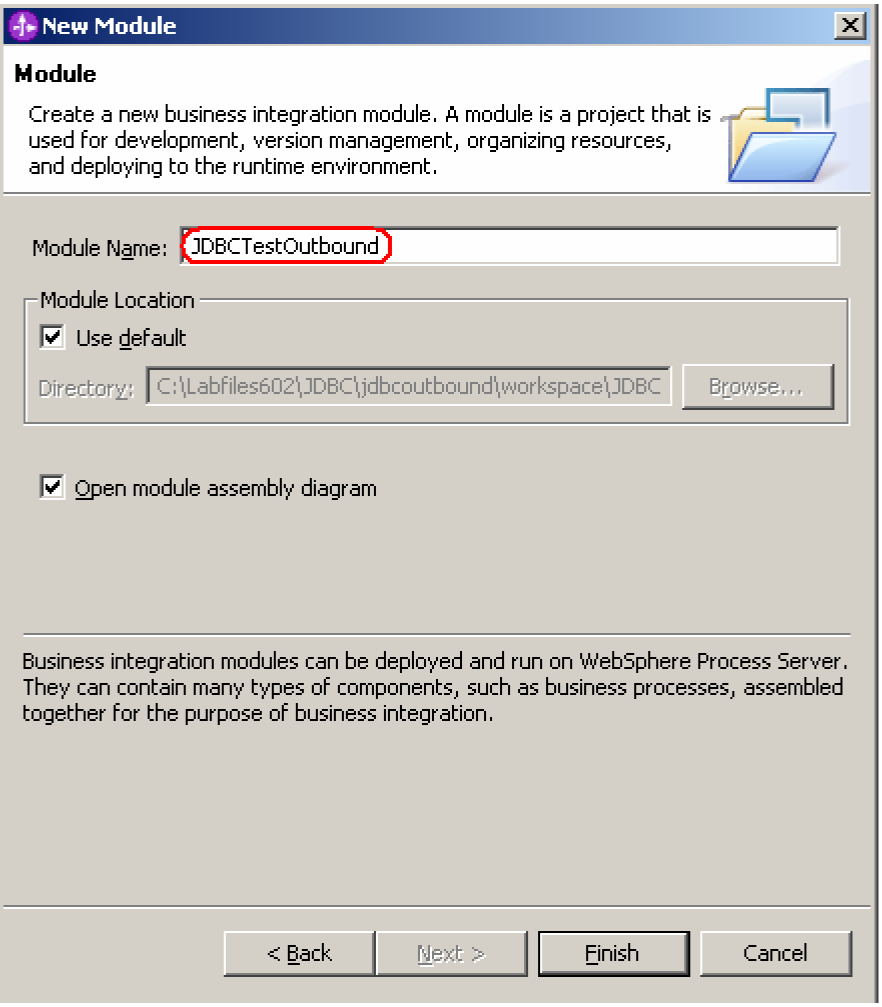
Max Records: 100

BO Location: com/test/data

____ **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.**


New Module
Module
Create a new business integration module. A module is a project that is used for development, version management, organizing resources, and deploying to the runtime environment.

Module Name: JDBCTestOutbound

Module Location

☑ Use default

Directory: C:\Labfiles602\JDBC\jdbcoutbound\workspace\JDBC

☑ Open module assembly diagram

Business integration modules can be deployed and run on WebSphere Process Server. They can contain many types of components, such as business processes, assembled together for the purpose of business integration.

__ **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.

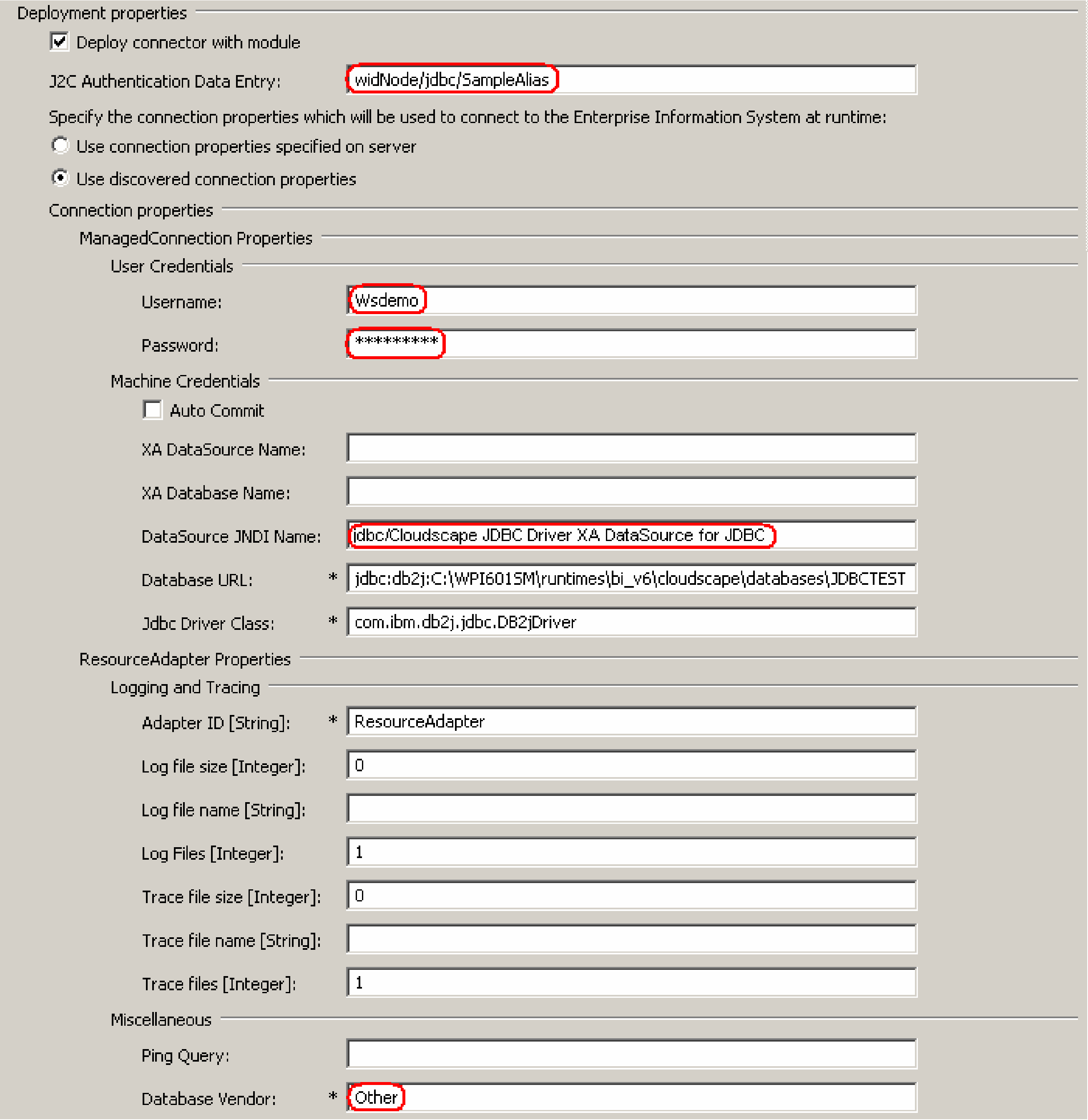
__ **e.** Select the radio button to the left of **Use discovered connection properties** Additional properties options appear to complete for ManagedConnectionProperties, ResourceAdapterProperties, and Miscellaneous properties.

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" check box should always be selected and the "Use discovered connection properties" check box should always be selected.

__ 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

DataSource JNDI Name: jdbc/Cloudscape JDBC Driver XA DataSource for JDBC (This is the datasource name you will create on the WebSphere Process Server in the next part of the lab.)
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.)



Deployment properties

☑ Deploy connector with module

J2C Authentication Data Entry: widNode/jdbc/SampleAlias

Specify the connection properties which will be used to connect to the Enterprise Information System at runtime:

○ Use connection properties specified on server

● Use discovered connection properties

Connection properties

ManagedConnection Properties

User Credentials

Username: Wsdemo

Password: *********

Machine Credentials

□ Auto Commit

XA DataSource Name:

XA Database Name:

DataSource JNDI Name: jdbc/Cloudscape JDBC Driver XA DataSource for JDBC

Database URL: * jdbc:db2j:C:\WPI601SM\runtimes\bi_v6\cloudscape\databases\JDBCTEST

Jdbc Driver Class: * com.ibm.db2j.jdbc.DB2jDriver

ResourceAdapter Properties

Logging and Tracing

Adapter ID [String]: * ResourceAdapter

Log file size [Integer]: 0

Log file name [String]:

Log Files [Integer]: 1

Trace file size [Integer]: 0

Trace file name [String]:

Trace files [Integer]: 1

Miscellaneous

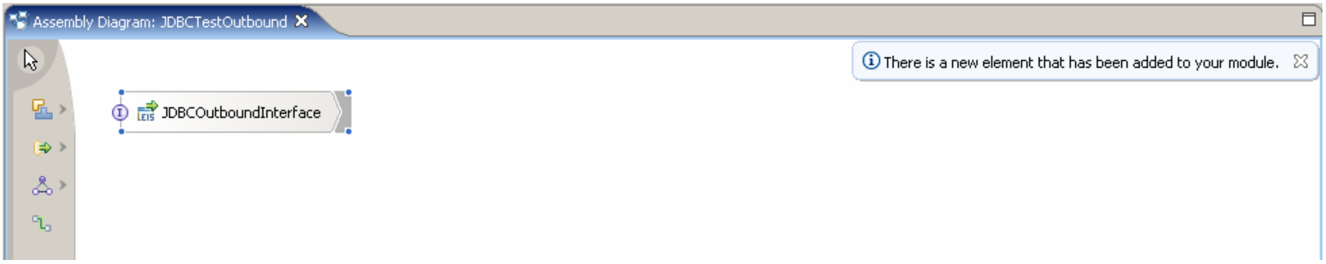
Ping Query:

Database Vendor: * Other

____ 8. Use the Assembly Diagram to wire a stand-alone reference to the com/test/data/JDBCOutboundInterface.

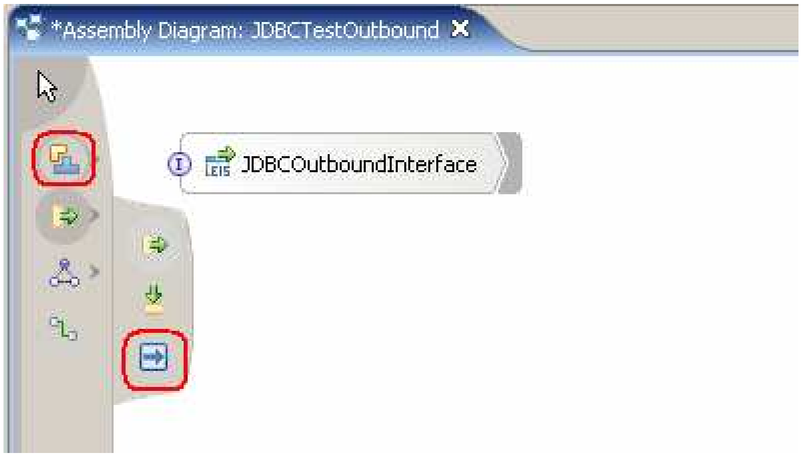
© Copyright IBM Corporation 2007. All rights reserved

__ 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.

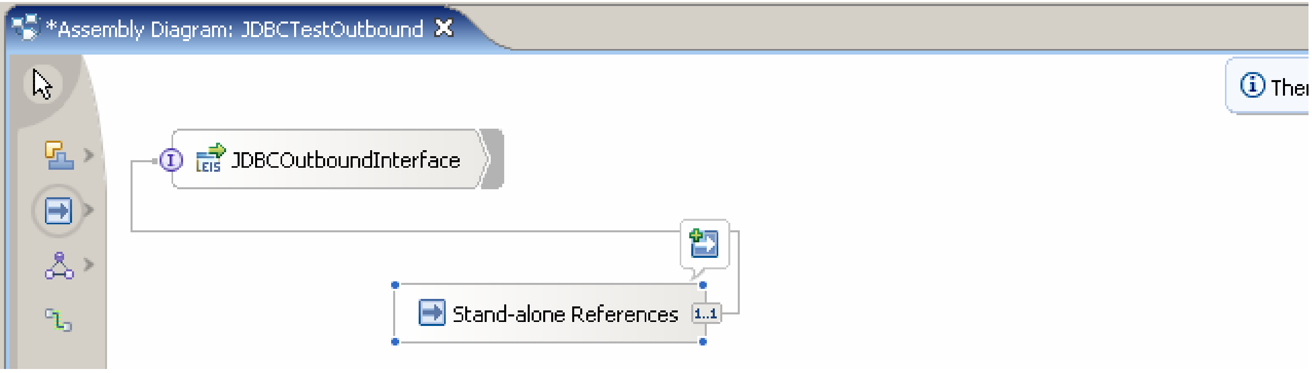


There is a new element that has been added to your module.

__ b. From the palette, select the import icon, then select the Stand-alone Reference icon and place it on the Assembly Diagram.



__ c. Wire the Stand-alone Reference to the **JDBCOutboundInterface**. At the Add Wire popup window, select **OK** in response to the “A 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:



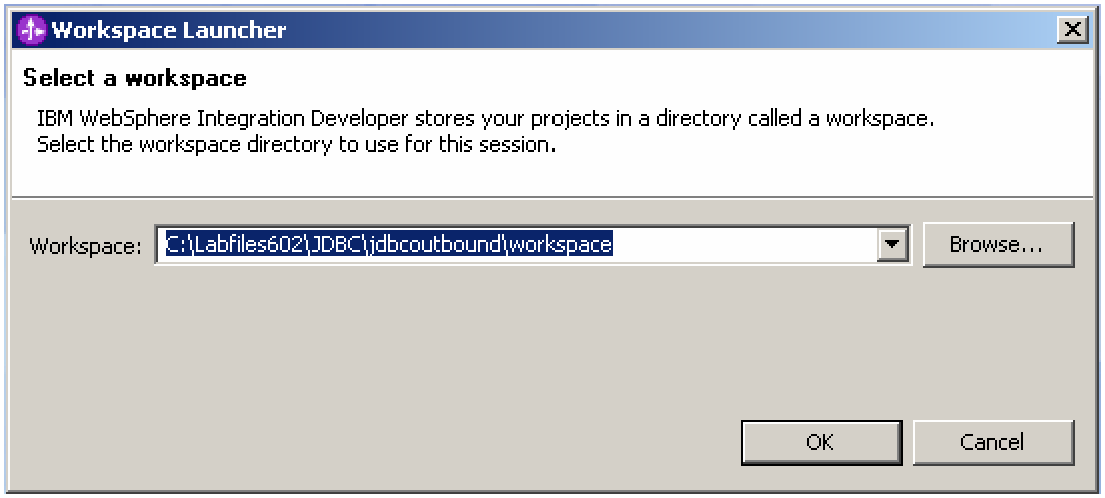
__ d. Save your work by selecting **File -> Save** from the top menu, or using the shortcut key sequence **Ctrl + S.**

__ 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 you 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.



Select a workspace

IBM WebSphere Integration Developer stores your projects in a directory called a workspace.
Select the workspace directory to use for this session.

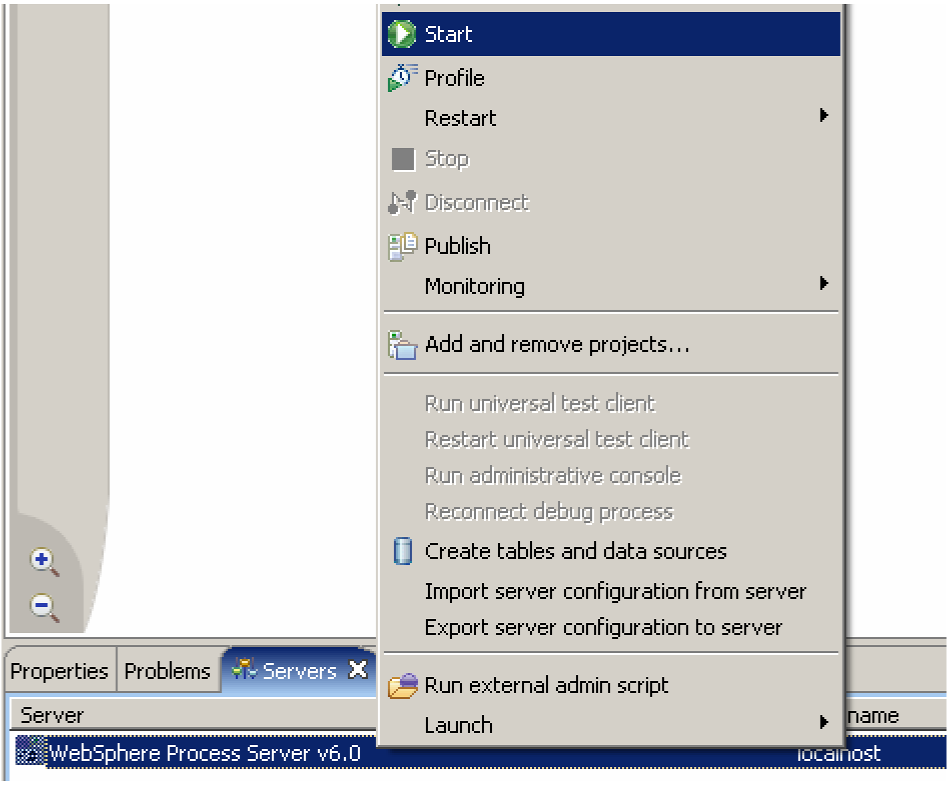
Workspace: C:\Labfiles602\JDBC\jdbcoutbound\workspace

Part 4: Using WebSphere Process Server Administrative Console to create J2C authentication alias and to 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 required DataSource JNDI Name that will be 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 creating and retrieving several records from the CUSTOMER table in the JDBCTEST database.

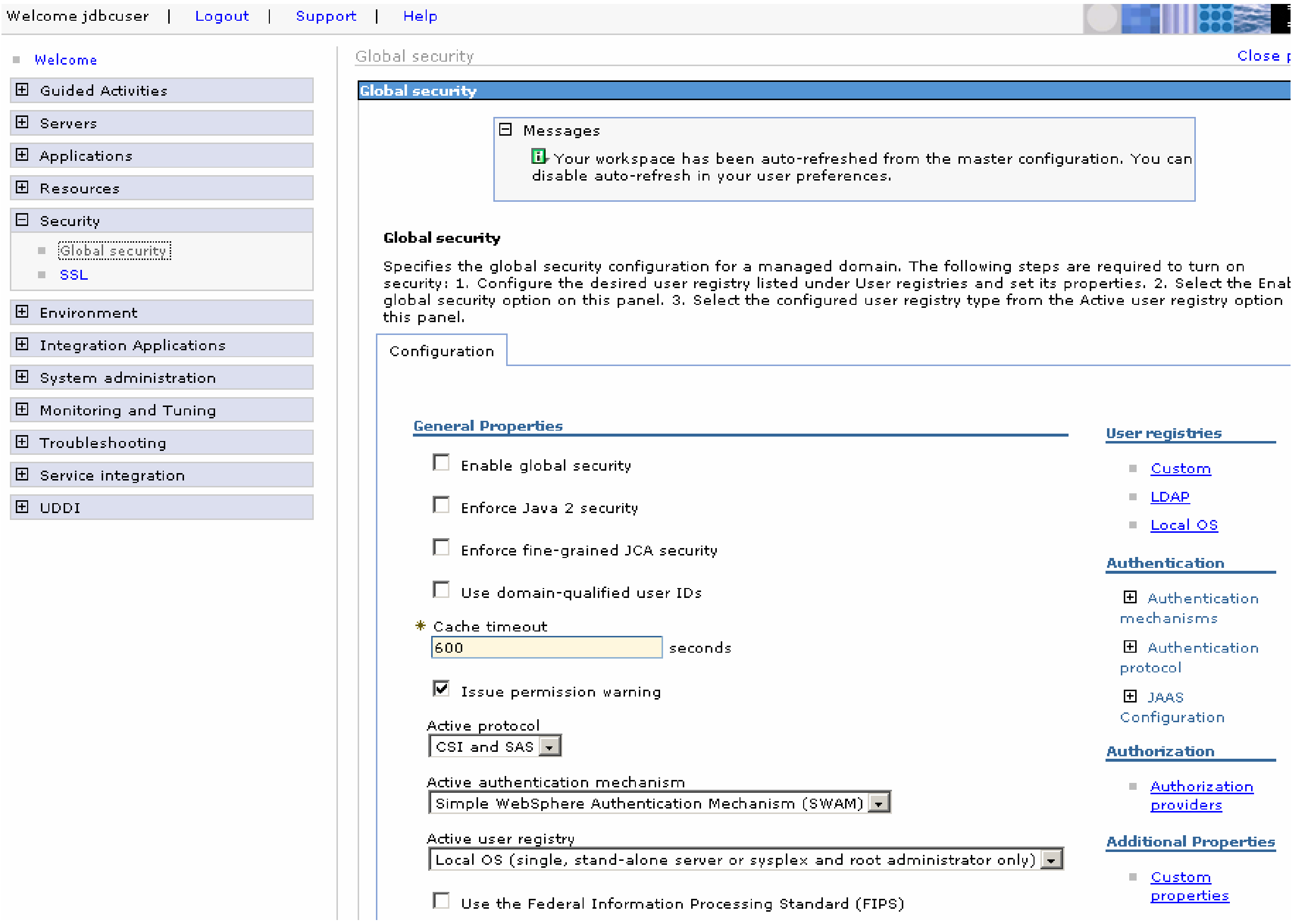
____ 1. Start the WebSphere Process Server and create an Authentication Alias named jdbc/SampleAlias.

__ a. In the Business Integration view, servers panel, right click and **Start** the WebSphere Process Server V6.0. Wait for the server to start.


Start
Profile
Restart
Stop
Disconnect
Publish
Monitoring
Add and remove projects...
Run universal test client
Restart universal test client
Run administrative console
Reconnect debug process
Create tables and data sources
Import server configuration from server
Export server configuration to server
Run external admin script
Launch
Properties
Problems
Servers
Server
name
WebSphere Process Server v6.0
localhost

__ 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



Global security

Messages

Your workspace has been auto-refreshed from the master configuration. You can disable auto-refresh in your user preferences.

Global security

Specifies the global security configuration for a managed domain. The following steps are required to turn on security: 1. Configure the desired user registry listed under User registries and set its properties. 2. Select the Enat global security option on this panel. 3. Select the configured user registry type from the Active user registry option this panel.

□ Enable global security

□ Enforce Java 2 security

□ Enforce fine-grained JCA security

□ Use domain-qualified user IDs

* Cache timeout
600 seconds

☑ Issue permission warning

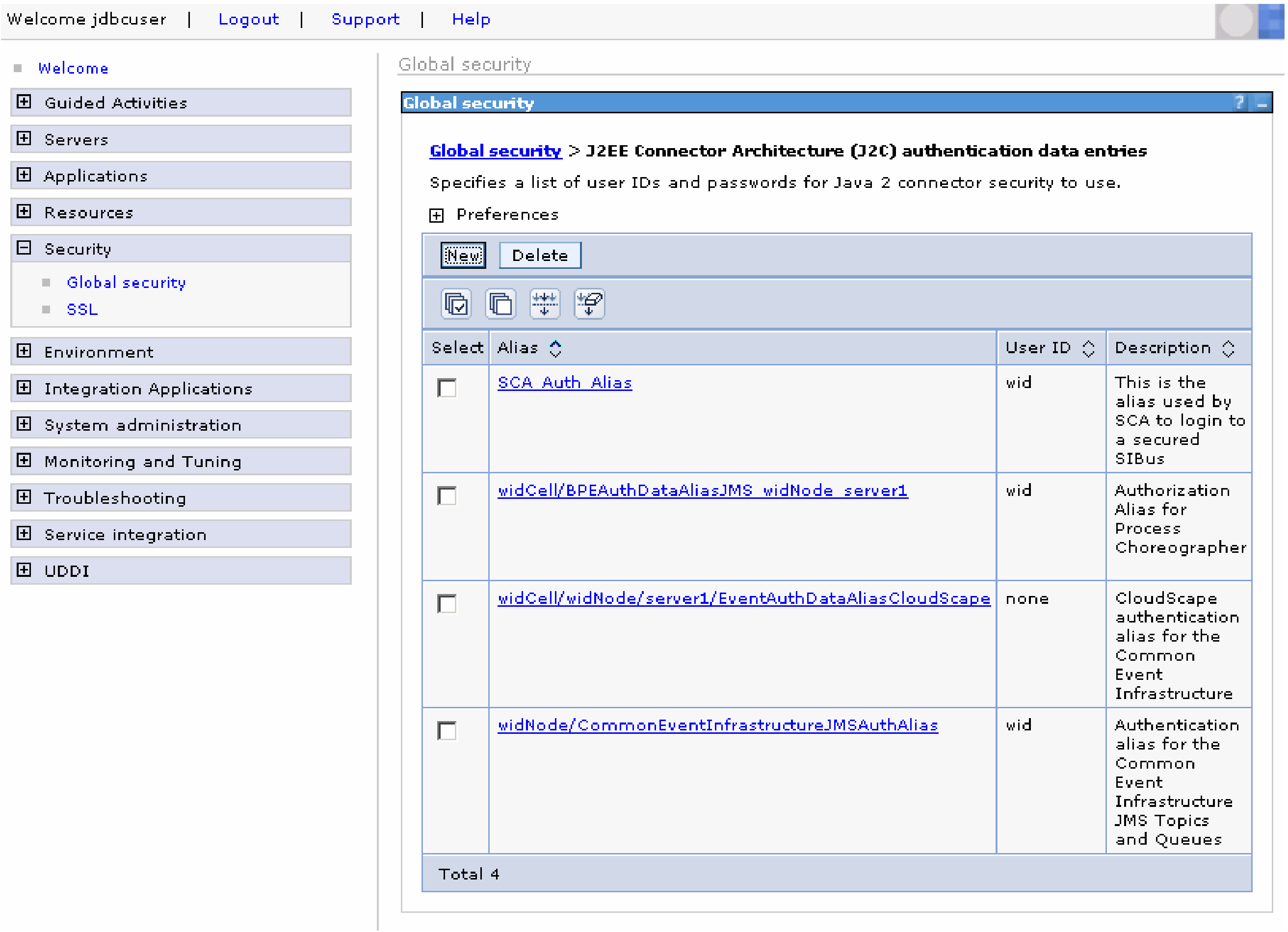
Active protocol
CSI and SAS

Active authentication mechanism
Simple WebSphere Authentication Mechanism (SWAM)

Active user registry
Local OS (single, stand-alone server or sysplex and root administrator only)

□ Use the Federal Information Processing Standard (FIPS)

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



Global security > **J2EE Connector Architecture (J2C) authentication data entries**

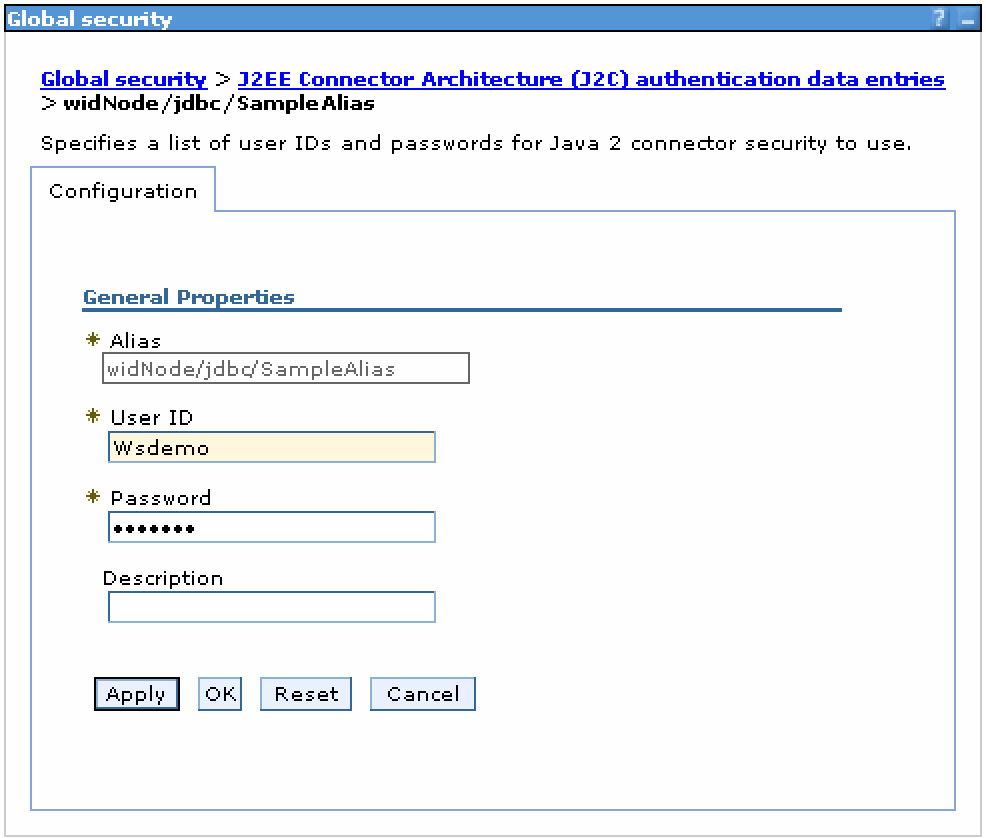
Specifies a list of user IDs and passwords for Java 2 connector security to use.

⊞ Preferences

Select	Alias	User ID	Description
□	SCA_Auth_Alias	wid	This is the alias used by SCA to login to a secured SIBus
□	widCell/BPEAuthDataAliasJMS_widNode_server1	wid	Authorization Alias for Process Choreographer
□	widCell/widNode/server1/EventAuthDataAliasCloudScape	none	CloudScape authentication alias for the Common Event Infrastructure
□	widNode/CommonEventInfrastructureJMSAuthAlias	wid	Authentication alias for the Common Event Infrastructure JMS Topics and Queues

Total 4

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



Global security

Global security > J2EE Connector Architecture (J2C) authentication data entries > widNode/jdbc/SampleAlias

Specifies a list of user IDs and passwords for Java 2 connector security to use.

Configuration

General Properties

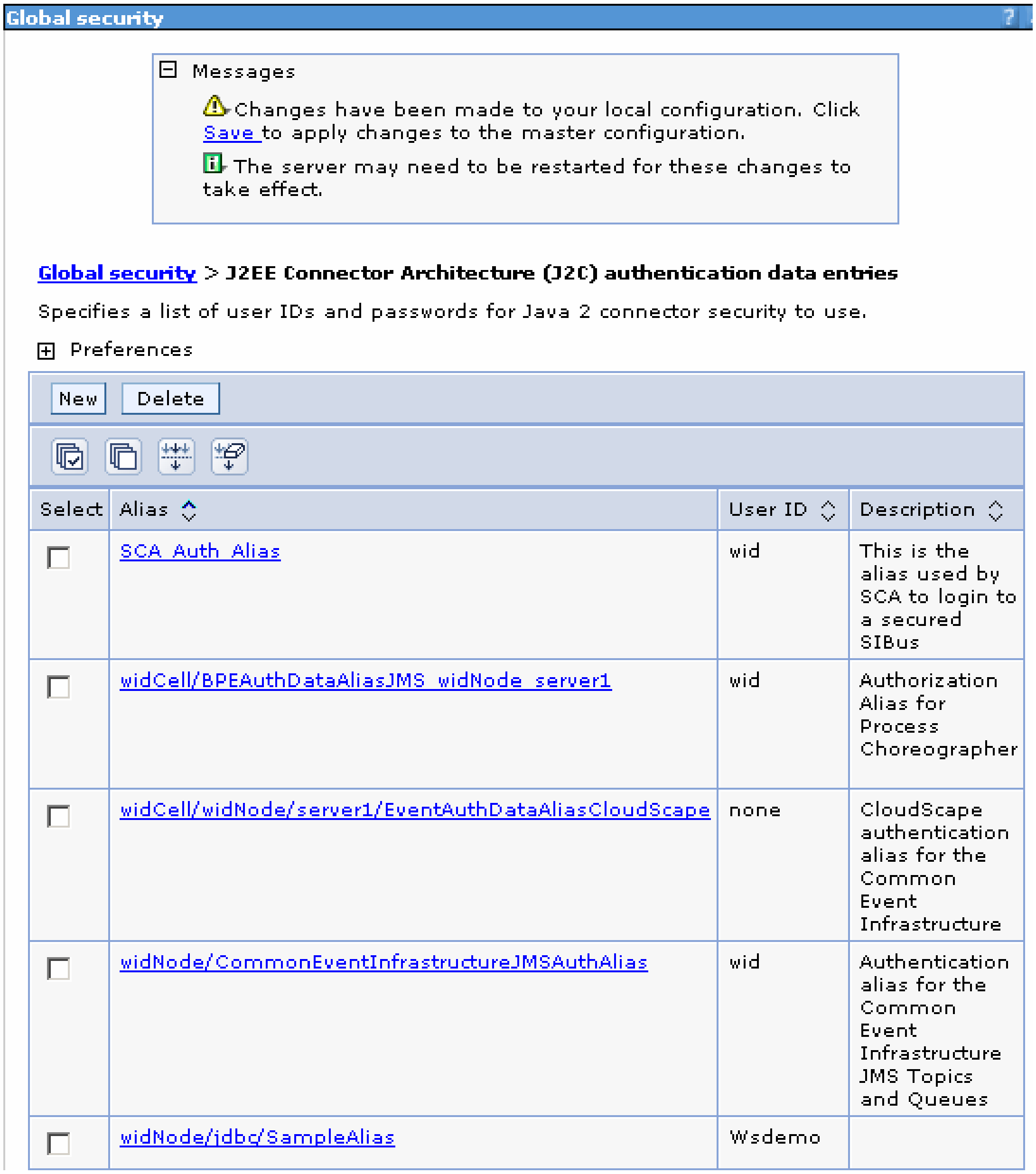
* Alias
widNode/jdbc/SampleAlias

* User ID
Wsdemo

* Password
•••••••

Description

__ f. Notice the Node name of widNode has been included to the alias name. Save the Changes. Click **Save.** Click **Save** again.



⊟ 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.

Global security > J2EE Connector Architecture (J2C) authentication data entries

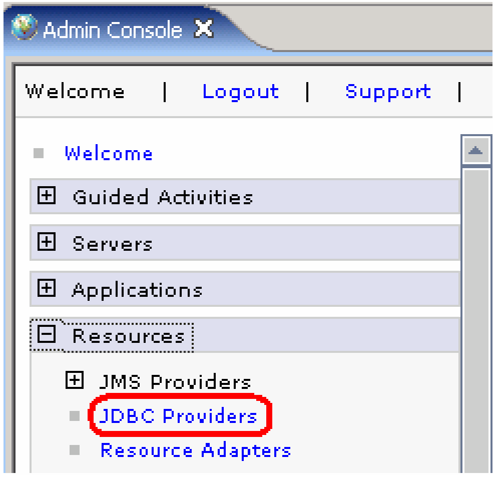
Specifies a list of user IDs and passwords for Java 2 connector security to use.

Select	Alias	User ID	Description
□	SCA Auth Alias	wid	This is the alias used by SCA to login to a secured SIBus
□	widCell/BPEAuthDataAliasJMS_widNode_server1	wid	Authorization Alias for Process Choreographer
□	widCell/widNode/server1/EventAuthDataAliasCloudScape	none	CloudScape authentication alias for the Common Event Infrastructure
□	widNode/CommonEventInfrastructureJMSAuthAlias	wid	Authentication alias for the Common Event Infrastructure JMS Topics and Queues
□	widNode/jdbc/SampleAlias	Wsdemo	

____ 2. Configure DataSource JNDI Name

__ a. Expand **Resources** on the left pane and select **JDBC Providers**

© Copyright IBM Corporation 2007. All rights reserved



__ b. Accept the default scope and click on **New**

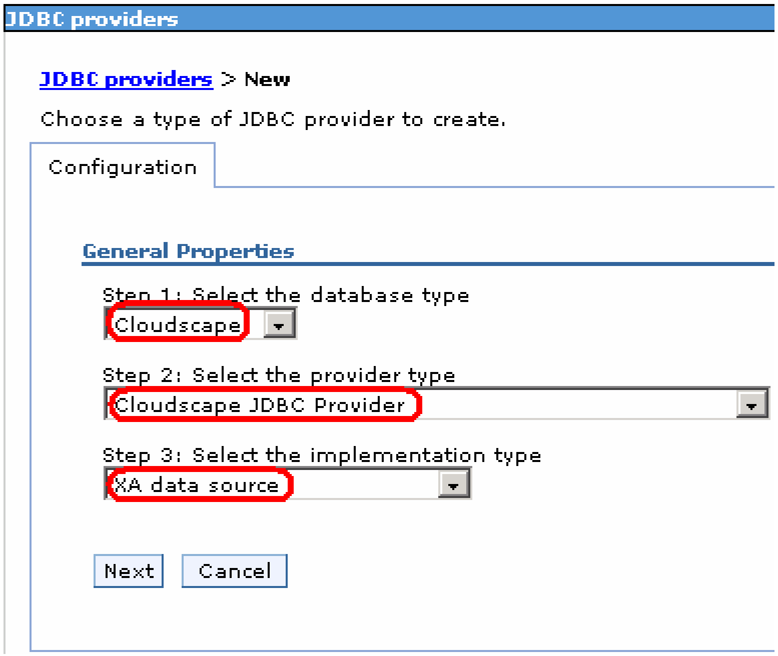
__ c. In the following screen, for General Properties, select the following from the dropdown list

1) Step 1: **Cloudscape**

2) Step 2: **Cloudscape JDBC Provider**

3) Step 3: **XA Data Source**

__ d. Click **Next**



JDBC providers

JDBC providers > New

Choose a type of JDBC provider to create.

Configuration

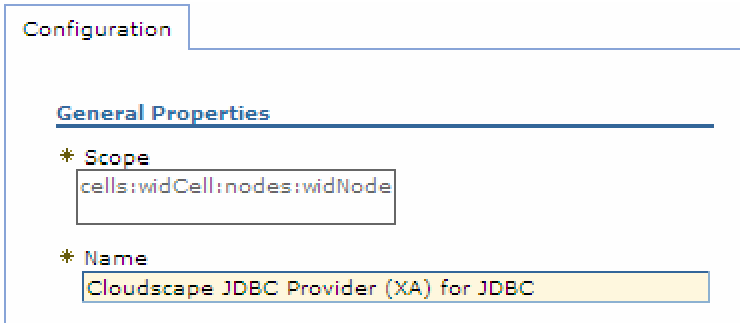
General Properties

Step 1: Select the database type: Cloudscape

Step 2: Select the provider type: Cloudscape JDBC Provider

Step 3: Select the implementation type: XA data source

__ e. Enter **Cloudscape JDBC Provider (XA) for JDBC** in the **Name** field and then click **OK**



General Properties

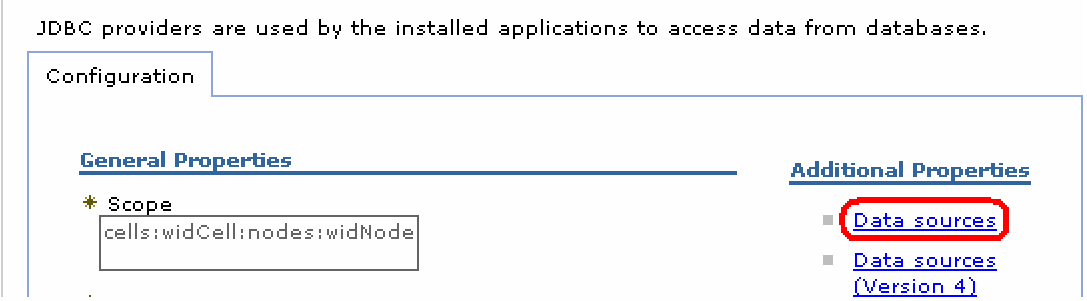
* Scope
cells:widCell:nodes:widNode

* Name
Cloudscape JDBC Provider (XA) for JDBC

__ f. Click **Save** on the top of the window, and then click for **Save** from the following screen

__ g. Click **Cloudscape JDBC Provider (XA) for JDBC** from the following screen to create a new data source

__ h. Select **Data sources** under **Additional Properties** on the right hand side



JDBC providers are used by the installed applications to access data from databases.

General Properties

* Scope
cells:widCell:nodes:widNode

Additional Properties

- Data sources
- Data sources (Version 4)

__ i. Create the required JNDI Data Source

1) Click **New**

2) Enter the following:

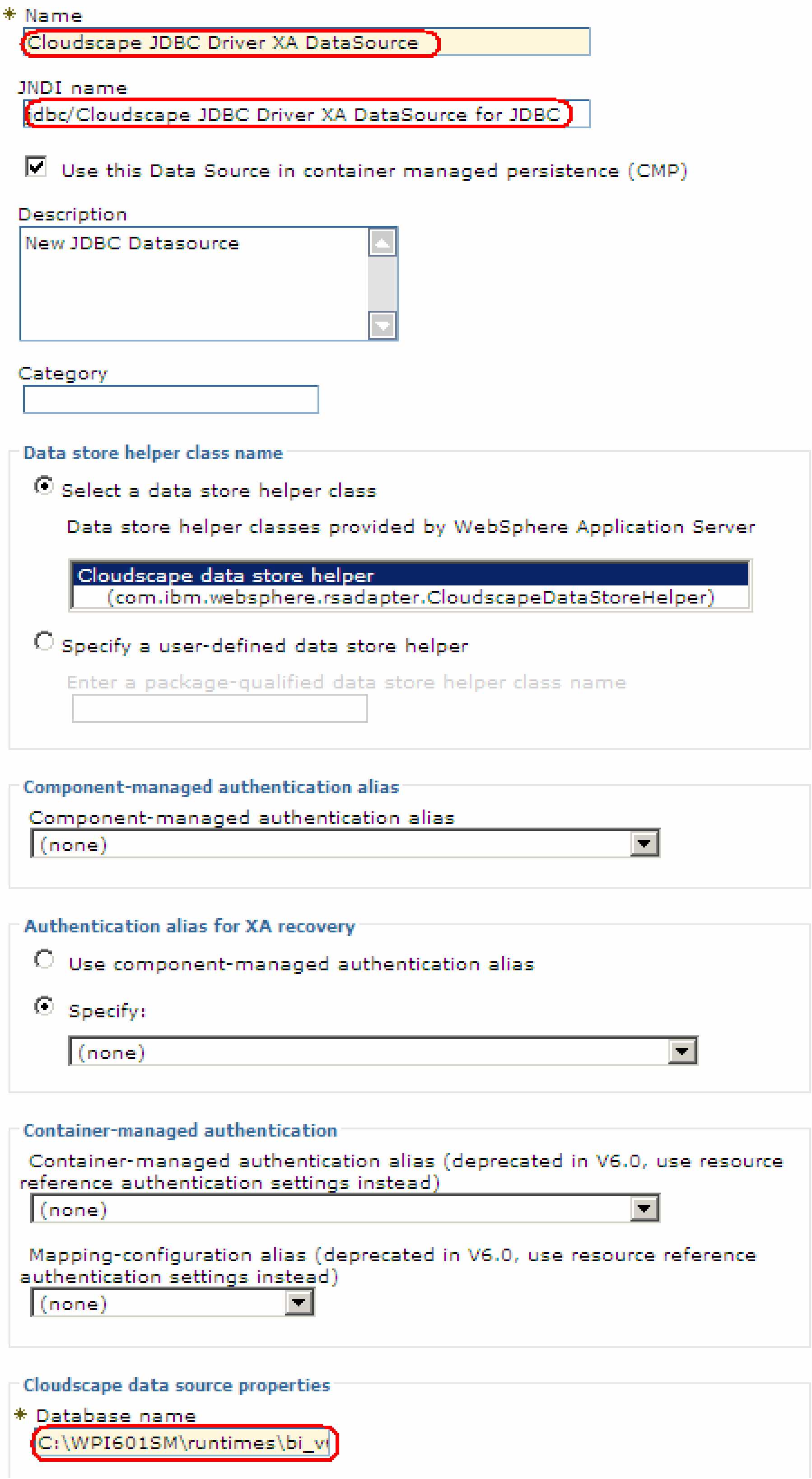
a) Name: **Cloudscape JDBC Driver XA DataSource for JDBC**

b) JNDI Name: **jdbc/Cloudscape JDBC Driver XA DataSource for JDBC**

c) Database name: **<WPS_HOME>\cloudscape\databases\JDBCTEST**

3) Click **OK**

© Copyright IBM Corporation 2007. All rights reserved



* Name
Cloudscape JDBC Driver XA DataSource

JNDI name
jdbc/Cloudscape JDBC Driver XA DataSource for JDBC

☑ Use this Data Source in container managed persistence (CMP)

Description
New JDBC Datasource

Category

Data store helper class name

● Select a data store helper class

Data store helper classes provided by WebSphere Application Server

Cloudscape data store helper
(com.ibm.websphere.rsadapter.CloudscapeDataStoreHelper)

○ Specify a user-defined data store helper

Enter a package-qualified data store helper class name

Component-managed authentication alias

Component-managed authentication alias
(none)

Authentication alias for XA recovery

○ Use component-managed authentication alias

● Specify:
(none)

Container-managed authentication

Container-managed authentication alias (deprecated in V6.0, use resource reference authentication settings instead)
(none)

Mapping-configuration alias (deprecated in V6.0, use resource reference authentication settings instead)
(none)

Cloudscape data source properties

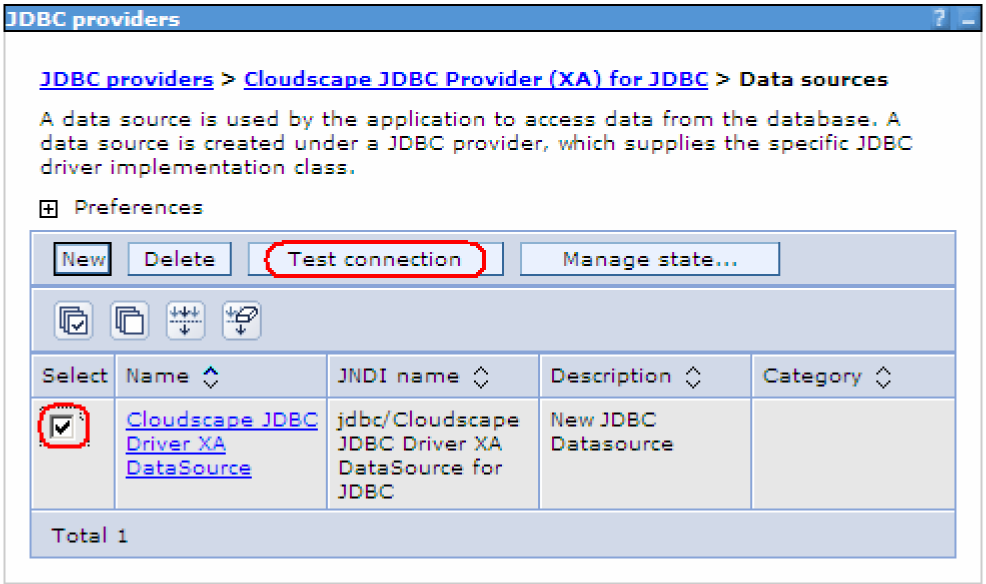
* Database name
C:\WPI601SM\runtimes\bi_v

4) Click on **Save** and then **Save** from the following screens

© Copyright IBM Corporation 2007. All rights reserved

5) Test the Data Source connection

a) Check the box next to **Cloudscape JDBC Driver XA DataSource for JDBC** and click on **Test connection** from the top of the screen



JDBC providers > Cloudscape JDBC Provider (XA) for JDBC > **Data sources**

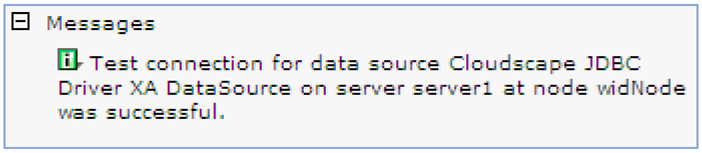
A data source is used by the application to access data from the database. A data source is created under a JDBC provider, which supplies the specific JDBC driver implementation class.

⊞ Preferences

Select	Name	JNDI name	Description	Category
☑	Cloudscape JDBC Driver XA DataSource	jdbc/Cloudscape JDBC Driver XA DataSource for JDBC	New JDBC Datasource	

Total 1

6) You should be the following success message on the top of the screen



⊟ Messages

Test connection for data source Cloudscape JDBC Driver XA DataSource on server server1 at node widNode was successful.

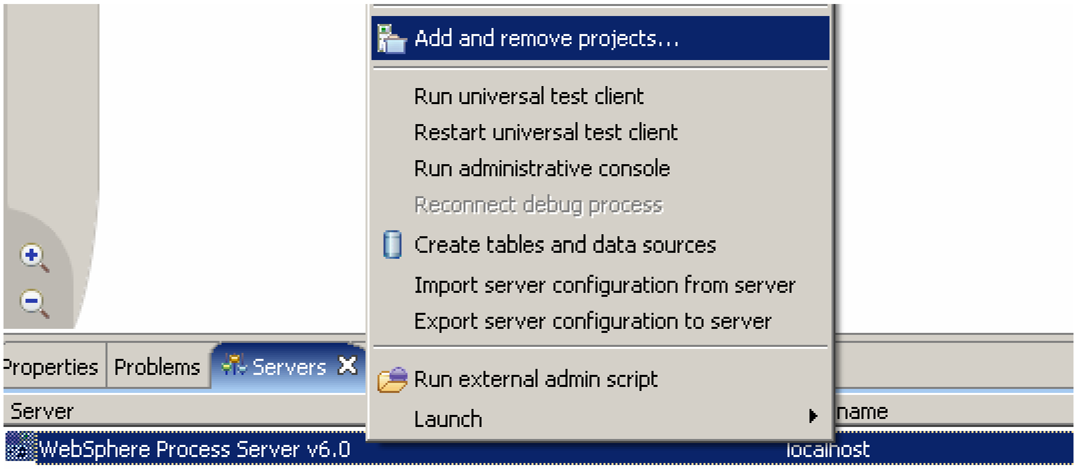
7) Log out of the Administrative Console and close it.

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

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. 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** …

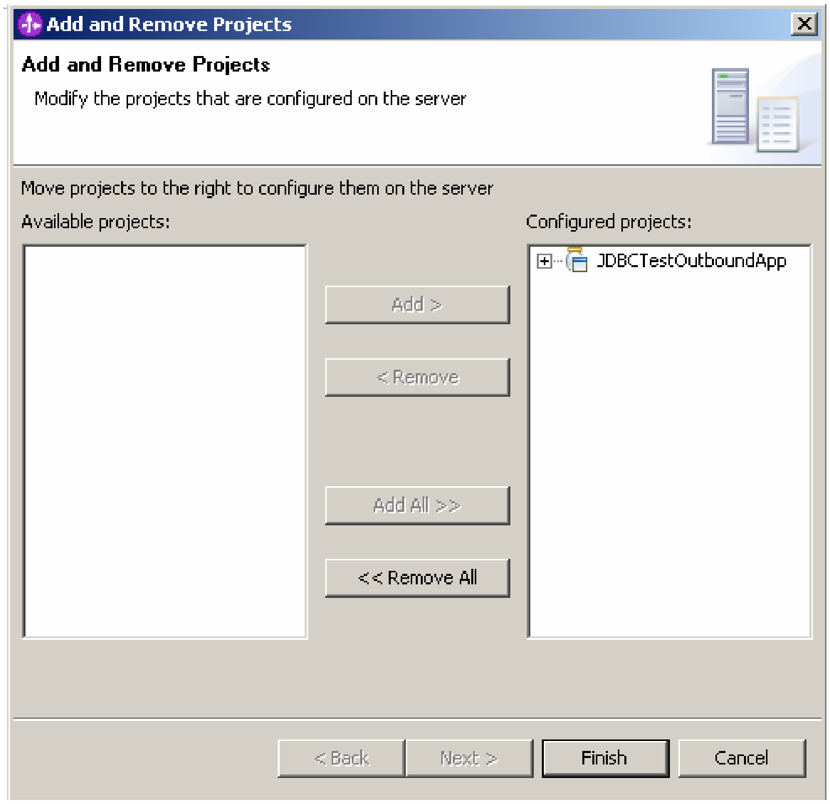

Add and remove projects...
Run universal test client
Restart universal test client
Run administrative console
Reconnect debug process
Create tables and data sources
Import server configuration from server
Export server configuration to server
Run external admin script
Launch
Properties
Problems
Servers
Server
name
WebSphere Process Server v6.0
localhost

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

© Copyright IBM Corporation 2007. All rights reserved

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

__ d. 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:

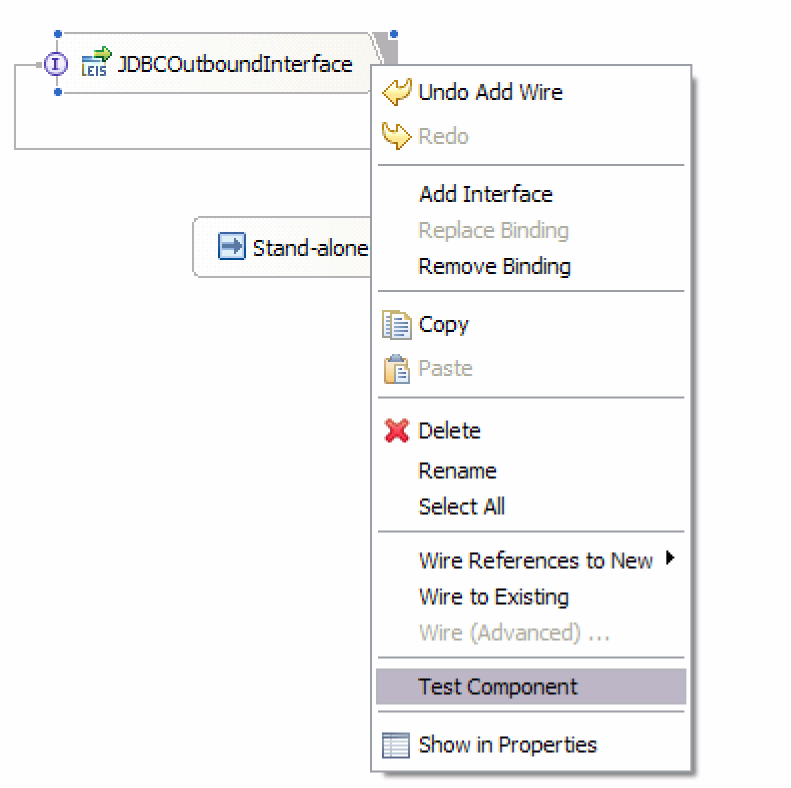
JDBCTestOutboundApp

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

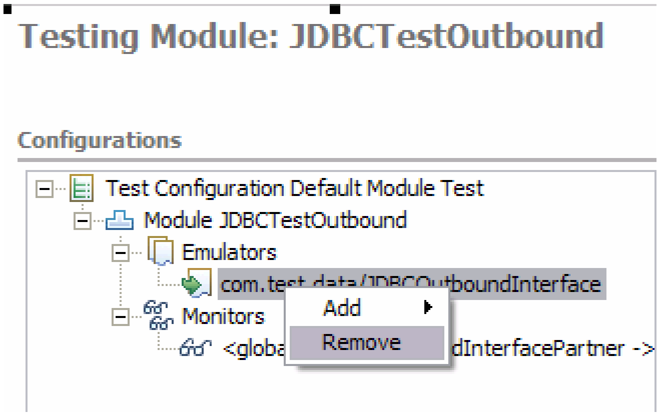
```
CWSCA3009I: The SCA module JDBCTestOutboundApp is starting.
CWSCA3010I: The SCA module started successfully.
WSVR0221I: Application started: JDBCTestOutboundApp
```

____ 2. 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**



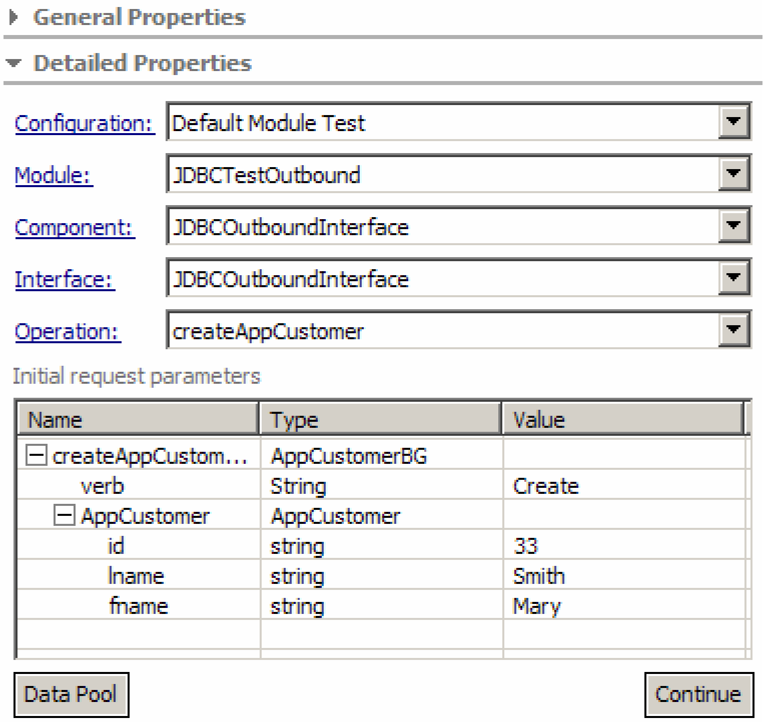
__ b. Switch to the **Configurations** tab, if you see **com/test/data/JDBCOutboundInterface** underneath **Emulators,** right click, and remove it. You may not see anything under Emulators. You want to connect to and drive the real JDBCTEST database and not have Test Component emulate for you.


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,** make sure operation is set for **createAppCustomer**

2) Under **Initial request parameters** , specify a Verb of **Create**, and specify a unique id, fname, and lname (all strings) (Do not use an id of 10 or 20 as two rows have already been created in the Customer table) Select **Continue**



Configuration: Default Module Test

Module: JDBCTestOutbound

Component: JDBCOutboundInterface

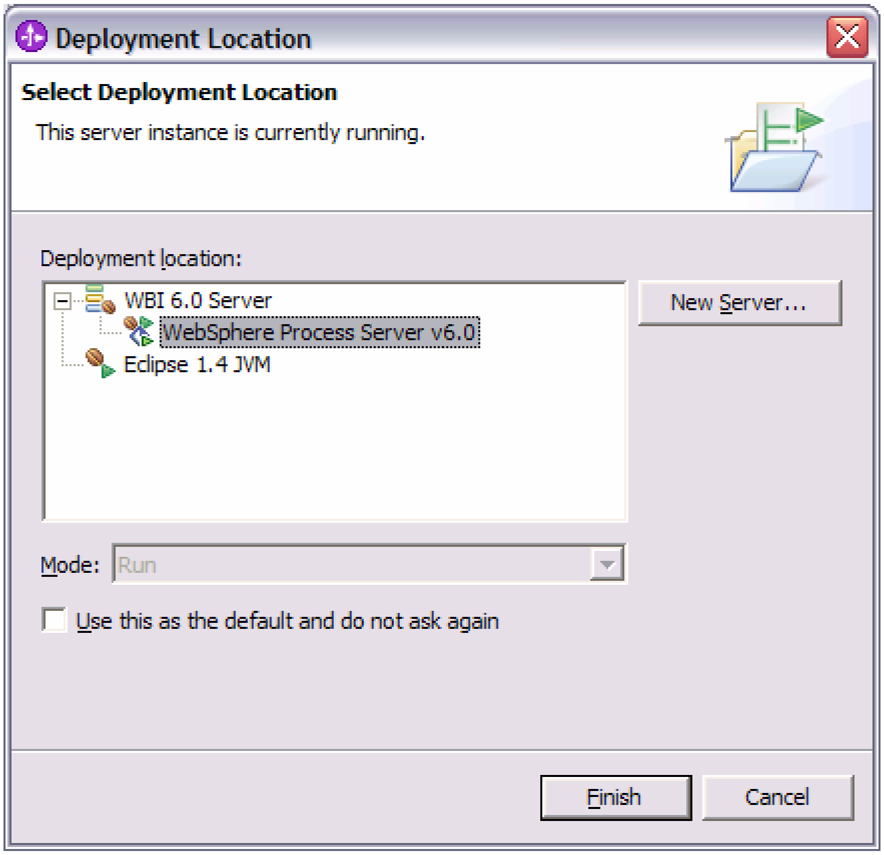
Interface: JDBCOutboundInterface

Operation: createAppCustomer

Initial request parameters

Name	Type	Value
createAppCustom...	AppCustomerBG	
verb	String	Create
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**



Deployment Location

Select Deployment Location

This server instance is currently running.

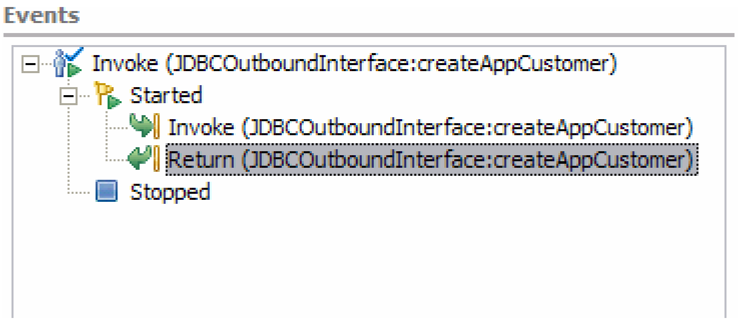
Deployment location:

- WBI 6.0 Server
  - WebSphere Process Server v6.0
- Eclipse 1.4 JVM

Mode: Run

□ Use this as the default and do not ask again

__ e. In the Events window you will see that Invoke has returned.



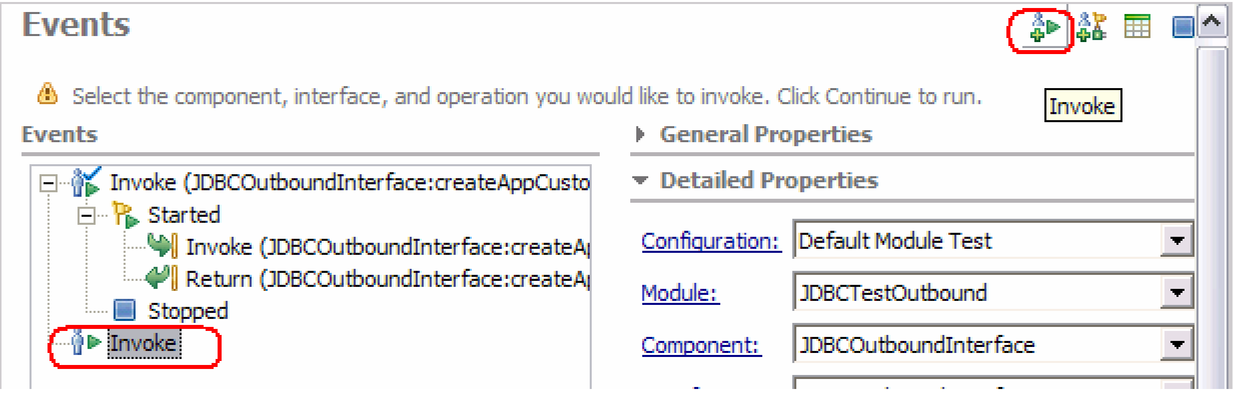
Events

- Invoke (JDBCOutboundInterface:createAppCustomer)
  - Started
    - Invoke (JDBCOutboundInterface:createAppCustomer)
    - Return (JDBCOutboundInterface:createAppCustomer)
  - Stopped

© Copyright IBM Corporation 2007. All rights reserved

____ 3. 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.



Events

⚠ Select the component, interface, and operation you would like to invoke. Click Continue to run.

Events

- Invoke (JDBCOutboundInterface:createAppCusto
  - Started
    - Invoke (JDBCOutboundInterface:createA
    - Return (JDBCOutboundInterface:createA
  - Stopped
- Invoke

▸ General Properties

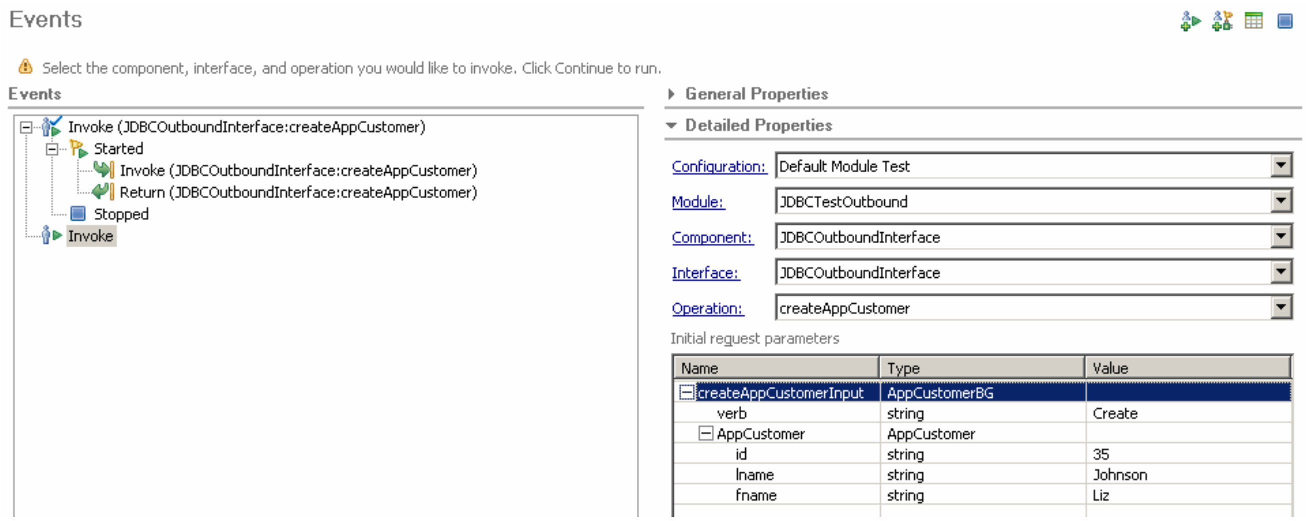
▾ Detailed Properties

Configuration: Default Module Test

Module: JDBCTestOutbound

Component: JDBCOutboundInterface

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



Events

⚠ Select the component, interface, and operation you would like to invoke. Click Continue to run.

Events

- Invoke (JDBCOutboundInterface:createAppCustomer)
  - Started
    - Invoke (JDBCOutboundInterface:createAppCustomer)
    - Return (JDBCOutboundInterface:createAppCustomer)
  - Stopped
- Invoke

▸ General Properties

▾ Detailed Properties

Configuration: Default Module Test

Module: JDBCTestOutbound

Component: JDBCOutboundInterface

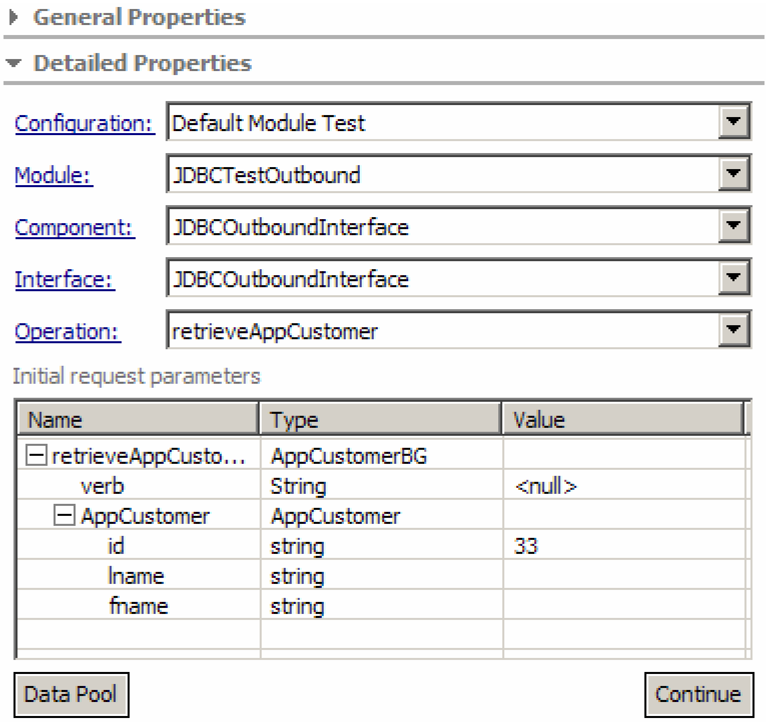
Interface: JDBCOutboundInterface

Operation: createAppCustomer

Initial request parameters

Name	Type	Value
⊟ createAppCustomerInput	AppCustomerBG	
verb	string	Create
⊟ AppCustomer	AppCustomer	
id	string	35
lname	string	Johnson
fname	string	Liz

____ 4. 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

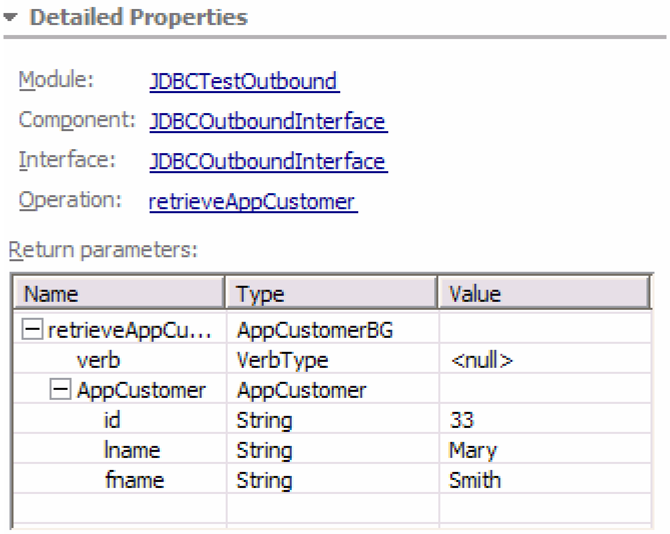
Interface: JDBCOutboundInterface

Operation: retrieveAppCustomer

Initial request parameters

Name	Type	Value
⊟ retrieveAppCusto...	AppCustomerBG	
verb	String	<null>
⊟ 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

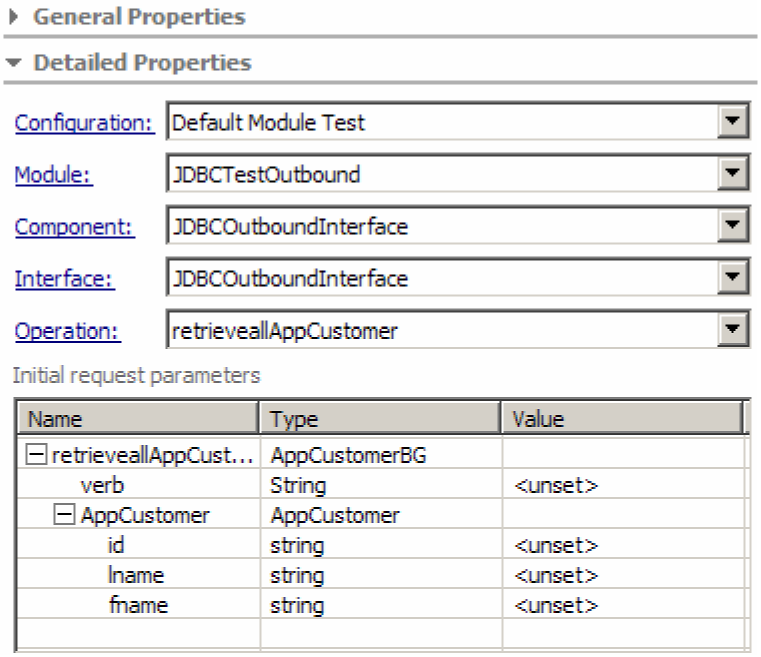
Interface: JDBCOutboundInterface

Operation: retrieveAppCustomer

Return parameters:

Name	Type	Value
⊟ retrieveAppCu...	AppCustomerBG	
verb	VerbType	<null>
⊟ 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.**



Configuration: Default Module Test

Module: JDBCTestOutbound

Component: JDBCOutboundInterface

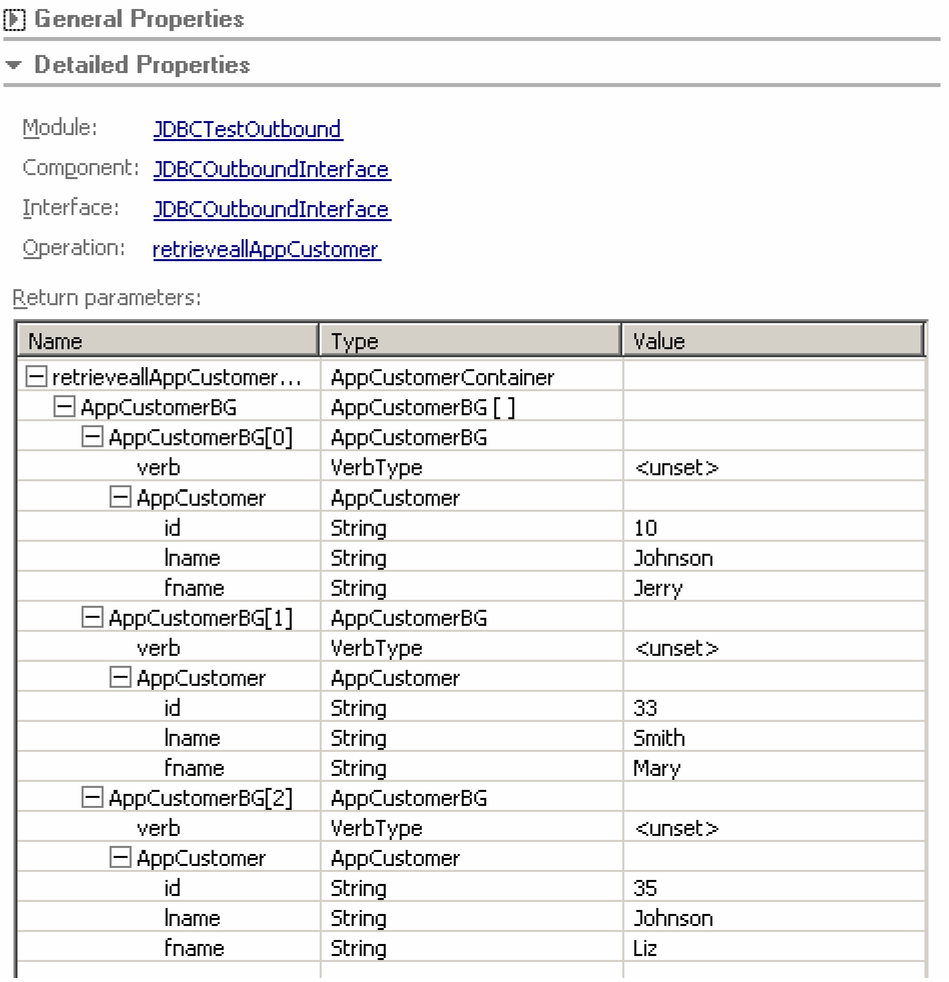
Interface: JDBCOutboundInterface

Operation: retrieveallAppCustomer

Initial request parameters

Name	Type	Value
retrieveallAppCust…	AppCustomerBG	
verb	String	<unset>
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.



General Properties

Detailed Properties

Module: JDBCTestOutbound

Component: JDBCOutboundInterface

Interface: JDBCOutboundInterface

Operation: retrieveallAppCustomer

Return parameters:

Name	Type	Value
⊟ retrieveallAppCustomer...	AppCustomerContainer	
⊟ AppCustomerBG	AppCustomerBG [ ]	
⊟ AppCustomerBG[0]	AppCustomerBG	
verb	VerbType	<unset>
⊟ AppCustomer	AppCustomer	
id	String	10
lname	String	Johnson
fname	String	Jerry
⊟ AppCustomerBG[1]	AppCustomerBG	
verb	VerbType	<unset>
⊟ AppCustomer	AppCustomer	
id	String	33
lname	String	Smith
fname	String	Mary
⊟ AppCustomerBG[2]	AppCustomerBG	
verb	VerbType	<unset>
⊟ AppCustomer	AppCustomer	
id	String	35
lname	String	Johnson
fname	String	Liz

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

____ 5. Exit the Test Component panel, remove the JDBCTestOutbound project from the server, and stop the server.

What you did in this exercise

- In this exercise, you learned how to install and deploy the Adapter for JDBC and integrate it into an SCA application for use with outbound request processing.