### IBM WEBSPHERE ADAPTER FOR JDBC V7.0 – LAB EXERCISE

## JDBC inbound lab – Wrapper business object

What this exercise is about	.1
Lab requirements	.1
What you should be able to do	. 1
Introduction	.2
Exercise instructions	.3
Part 1: Create the JDBCTEST database and tables using DB2	.4
Part 2: Set up the development environment	.6
Part 3: Use External Service wizard to generate business objects and other artifacts	.7
Part 4: Create the authentication alias and configure the data source	19
Part 5: Test the application using the WebSphere test environment	24
What you did in this exercise	27

## What this exercise is about

The objective of this lab is to introduce you wrapper business object for inbound processing.

## Lab requirements

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

- WebSphere Integration Developer V7.0 installed
- WebSphere Process Server V7.0 test environment installed
- WebSphere Adapter for JDBC V7.0 installed
- Sample code in the directory C:\Labfiles70\JDBC (Windows) or /tmp/LabFiles70/JDBC (Linux)

## What you should be able to do

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

• Understand how WebSphere Adapter for JDBC 7.0.0.0 retrieves customer information from an application's database. A wrapper business object is used to retrieve records from multiple tables with one event entry.

## Introduction

This lab illustrates the ability of WebSphere JDBC adapter to interact with database by polling database event from an event table. In this scenario, a legacy application makes some change of the CUSTOMER table and the GOODS table in a single operation. Then, insert an event entry record into the event table (WBIA\_EVENT\_TABLE). Then, the event is polled by JDBC adapter and sends it to one SCA component. JDBC adapter screen all the database operation details, event quality assuring details and provide a simple event interface for the application component. The following figure shows the whole scenario:



This case has three steps:

1. The legacy application will make the changes and then generate an event record. For simplify reason, you will insert records using SQL statement directly.

2. JDBC adapter will poll the event from database periodically. Thus, it will find the new events and fetch the event and corresponding business objects from database.

3. At last, JDBC adapter will convert the event to a SDO and send it to the destination SCA component.

## **Exercise instructions**

Some instructions in this lab are 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></lab_name>	JDBCInBoundWrapperBO	
<wid_home></wid_home>	C:\IBM\WID7	
<wps_home></wps_home>	C:\IBM\WID7_WTE	
<jdbcadapter_home></jdbcadapter_home>	<wid_home>\Resource Adapters\JDBC_7.0.0.0\</wid_home>	
<lab_files></lab_files>	C:\Labfiles70	/tmp/Labfiles70
<temp></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, replace C:\LabFiles70\ with C:/LabFiles70/

## Part 1: Create the JDBCTEST database and tables using DB2

In this part you will create samples tables in DB2 in preparation for this tutorial. You will create table CUSTOMER and table ADDRESS along with event store table.

\_\_\_\_1. Open IBM DB2 → Control Center

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

2. Create a new database **JDBCTEST** if it does not already exist.

Control Center - DB2COPY1 Control Center Selected Edit View Tools Hi	elp	
₽ % 8 5 2 1 1 1 2 5	• {	
C Object View		
		T60JULIELAPTOP - DB2 - JDBCTEST - Tables $J_Z^A$
		Name ⇔ Schema ⇒ Table space ⇔ Comment ⇔ Index table space
Views     Aliases     Nicknames     Cache Objects		EXPLAIN_ARGUMENT     DB2ADMIN     USERSPACE1       Image: EXPLAIN_DIAGNOSTIC     DB2ADMIN     USERSPACE1       Image: EXPLAIN_DIAGNOSTIC_DA     DB2ADMIN     USERSPACE1
Triggers		Image: State of 121 items displayed     ↓ ▲ 3 ↔ 3 ↔ 3 ↔ 3 ↔ 5 ↔ 5 ↔ 5 ↔ 5 ↔ 5 ↔ 5 ↔
Indexes Table Spaces	- 00	Tables
← C Event Monitors ← C Buffer Pools ⊕ C Application Objects ⊕ C User and Group Objects ⊕ C Federated Database Objects		Actions: Select an object from the list above to display more details.
→ C XML Schema Repository (XSR) JDEEVENT	~	

\_\_\_\_ 3. Click 🔤 to open command editor

\_\_\_\_a. Add JDBCTEST as target database

4. Paste these following scripts to create two tables and one event table for recording events

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

\_\_\_a. Create CUSTOMER table

```
CREATE TABLE CUSTOMER (

PKEY VARCHAR(10) NOT NULL PRIMARY KEY,

FNAME VARCHAR(20) ,

LNAME VARCHAR(20) ,

CCODE VARCHAR(10) ) ;
```

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

\_\_\_\_ b. Create CUSTADD table

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

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

\_\_\_ c. Create the WBIA\_JDBC\_EVENTSTORE table

```
CREATE TABLE WBIA_JDBC_EVENTSTORE (
EVENT_ID INTEGER NOT NULL GENERATED ALWAYS AS IDENTITY (START WITH 1,
INCREMENT BY 1) PRIMARY KEY,
XID VARCHAR(200),
OBJECT_KEY VARCHAR(80) NOT NULL,
OBJECT_NAME VARCHAR(40) NOT NULL,
OBJECT_FUNCTION VARCHAR(40) NOT NULL,
EVENT_PRIORITY INTEGER NOT NULL,
EVENT_TIME TIMESTAMP,
EVENT_STATUS INTEGER NOT NULL,
EVENT_COMMENT VARCHAR(100) );
```

\_ 5. Click 🏴 to run the SQL scripts. Make sure all SQL commands completed successfully

CREATE TABLE WBIA\_JDBC\_EVENTSTORE ( EVENT\_ID INTEGER NOT NULL PRIMARY KEY, XID VARCHAR(200), OBJECT\_KEY VARCHAR(80) NOT NULL, OBJECT\_NAME VARCHAR(40) NOT NULL, OBJECT\_FUNCTION VARCHAR(40) NOT NULL, EVENT\_PRIORITY INTEGER NOT NULL, EVENT\_TIME TIMESTAMP, EVENT\_STATUS INTEGER NOT NULL, EVENT\_COMMENT VARCHAR(100) );

CREATE TABLE WBIA\_JDBC\_EVENTSTORE ( EVENT\_ID INTEGER NOT NULL PRIMARY KEY, XID VARCHAR(200), OBJECT\_KEY VARCHAR(80) NOT NU DB20000I The SQL command completed successfully.

\_\_6. Insert records into two tables you have just created. This will add two entries to the CUSTOMER and ADDRESS tables.

INSERT INTO CUSTOMER (PKEY, FNAME, LNAME, CCODE)
VALUES ('C1', 'JONE', 'SMITH', '1');
INSERT INTO CUSTOMER (PKEY, FNAME, LNAME, CCODE)
VALUES ('C2', 'ROTH', 'GREEN', '1');
INSERT INTO ADDRESS (ADDRID, CUSTID, CITY, ZIPCODE)
VALUES ('A1', 'C1', 'AUSTIN', '100000');
INSERT INTO ADDRESS (ADDRID, CUSTID, CITY, ZIPCODE)
VALUES ('A2', 'C2', 'HOUSTON', '200000');

\_\_\_\_\_7. Exit DB2 command editor and control center.

## Part 2: Set up the development environment

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

- 1. Start WebSphere Integration Developer V7.0 with a new workspace.
  - \_\_\_\_a. From the start menu select Start > Programs → IBM WebSphere Integration Developer → IBM WebSphere Integration Developer 7.0 → IBM WebSphere Integration Developer V7.0
  - \_\_\_\_b. When prompted enter <LAB\_FILES>\JDBC\Inbound\Wrapper for your workspace and click OK

🕒 Worksp	ace Launcher	- 🔀
Select a w	orkspace	43
IBM WebSpl Choose a w	here Integration Developer 7.0 stores your projects in a folder called a v orkspace folder to use for this session.	vorkspace.
Workspace:	C:\IBM\WIDProjects\workspace\70\JDBC\Inbound\Wraper	Browse
Copy Set	tings	
0		OK Cancel

#### \_\_\_\_ c. When WebSphere Integration Developer V7.0 opens, close the Getting Started page



# Part 3: Use External Service wizard to generate business objects and other artifacts

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

- In the Business Integration Perspective, run the External Service wizard. A Business Integration project is created for you during this process.
  - \_\_\_\_a. From the top Menu bar, select Window > Open Perspective > Other ... > Business Integration (default) click OK
- 2. From within the Business Integration Perspective, 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

3
$\bigcirc$
E.

- 3. Highlight IBM WebSphere Adapter for JDBC (IBM: 7.0.0.0) and click Next
- \_\_\_\_\_4. 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 V7.0 is selected



- 5. 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>\universalDriver\_wbi\lib and select the db2jcc.jar, db2jcc\_license\_cu.jar, and db2jcc\_license\_cisuz.jar

O New External Service	
Locate the Required Files and Libraries Connector projects require system-specific libraries and JAR files.	
When deploying the application to a stand-alone server, these files must be configured on the server. Specify the location of the JDBC driver JAR files required to access the database server. JDBC driver JA shipped with the database installation. If you do not have a local database installation, you can downlo from the database vendor's Web site. JDBC driver JAR files:	<u>More</u> AR files are ad these files
C: \IBM\70GM\WID7_WTE\runtimes\bi_v7\universalDriver_wbi\ib\db2jcc_license_cu.jar C: \IBM\70GM\WID7_WTE\runtimes\bi_v7\universalDriver_wbi\ib\db2jcc_jar C: \IBM\70GM\WID7_WTE\runtimes\bi_v7\universalDriver_wbi\ib\db2jcc_license_cisuz.jar Browse for the database-specific JDBC	Add Remove

### \_\_\_ b. Click Next

- 6. Select the type of processing the adapter will perform at runtime
  - \_\_\_\_a. Select Inbound and click Next

🚯 New External S	Service	
Select the Proce	ssing Direction	
Select the direction of	of adapter processing at run time.	
Inbound		
Inbound proces	ssing passes data from the adapter to your se	rvice export.
Outbound		
Outbound proc	essing passes data from your service import t	o the adapter.
0	< Back Next > F	inish Cancel

- 7. 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, these information are 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 **DB2 UDB** and select version of your db2
  - \_\_\_\_b. From the right panel, enter/select these following values
    - 1) Database JDBCTEST
    - 2) Hostname LOCALHOST
    - 3) Port number **50000**
  - \_\_\_\_ c. Enter valid user ID and password values to access JDBCTEST database

New External Service	ties	
,,		6
Connection properties Database system connection info	rmation	
<b>*</b>	Propercies.	
DB2 UDB	JDBC driver type:	IBM DB2 Universal
↓ √ √7.2 ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓	Database: *	JDBCTEST
- 🔷 V8.2	Host name: *	localhost
···· ♦ V9.1	Port number: *	50000
⊕ ♦ DB2 UDB iseries     ⊕ ♦ DB2 UDB iseries	JDBC driver class name: *	com.ibm.db2.jcc.DB2Driver
🕀 🛇 Orade	Database URL:	jdbc:db2://localhost:50000/JDBCTEST
⊕ SQL Server	Additional JDBC driver	2
E Seperic IDBC	[name:value;name:value]:	<u> </u>
User name:	⊧ db2admin	
Password:	r (********	
Prefix for business object names		
Advanced >>		
Change the logging properties for	the wizard	

### \_\_\_c. Click NEXT

### 8. Complete the **Discover Objects and Services** panel

\_\_\_\_a. Select the **Run Query** button. A connection is made to the DB2 JDBCTEST database and a selection of Meta data objects is presented in a tree-like structure.

Note: Select the **Edit Query** ... button to see the available options, however, do not change any of the defaults for this lab exercise.

\_\_\_\_b. Expand the JDBCTEST database, then select **Tables**, and highlight **CUSTOMER** and

ADDRESS, click button. CUSTOMER and ADDRESS now open in the Objects to be imported window. Click Next

ind Objects in the Enterprise System Use a query to find objects on the enterprise system. Click Edit Que Run Query to run it.	ery to create the query, then dick
uery: SchemaFilterLabel=null,Schema name or pattern=null,Suppo	orted database object types=Tab Edit Query.
scovered objects:	Selected objects:
款 /2 印	-
<ul> <li>DB2ADMIN</li> <li>JULIEIBM</li> <li>Tables</li> <li>CLISTOMER</li> <li>CLISTOMER</li> <li>EMPLOYEES</li> <li>Views</li> <li>Synonyms - Nick Names</li> <li>NULLID</li> <li>SQLJ</li> <li>SYSCAT</li> <li>SYSFUN</li> <li>SYSFUN</li> <li>SYSFROC</li> <li>SYSTOOLS</li> </ul>	ADDRESS     CUSTOMER
2)	Next > Finish Cancel

- 9. Complete the **Configure Objects** panel
  - \_\_\_\_a. For wrapper object names, click Add and enter Wrapper for the new wrapper business object.
  - \_\_\_\_b. In the **Table**, view, synonym, or nickname child objects for the selected wrapper area, click Add to add CUSTOMER and ADDRESS table business objects for the wrapper.
  - \_\_\_\_ c. In the **Service functions for selected wrapper object** area, click **Add** to add functions for the wrapper.
  - \_\_\_\_ d. Leave the default value for Namespace.

\_\_\_\_ e. Note that Business Graph is optional - It determines if you want to generate business graph or not. If the property is checked only then business graphs is generated. And by default the property is checked.

New External Service		
<b>pecify Composite Properties</b> Specify properties that apply to all selec	ted objects.	
Operations for selected business object	s	
Create Update Delete	e audeu to the service interface.	Add Remove
Create and configure user-defined wrap Wrapper object pames:	oper objects	
Wrapper		Add Remove
Table, view, synonym, or nickname ADDRESS CUSTOMER	child business objects for the selected wrapper obje	Add
Service functions for selected wrap Create Delete Update	per object:	Add (Remove)
Business object namespace: http://ww Specify the relative folder for generated Folder:	vw.ibm.com/xmlns/prod/websphere/j2ca/jdbc d business objects	
✓ Generate a business graph for each	business object	
0	<back next=""> Emil</back>	sh Cancel

\_\_\_f. Click Next

- \_\_\_\_ 10. Complete Publishing Object Configuration Properties
  - \_\_\_\_a. Specify the security crendentials
    - 1) Select Other to define the data source JNDI name
  - \_\_\_\_b. Leave the default "With module for use by single application" option in this exercise.
  - \_\_\_\_ c. Under Connection properties

- 1) Leave the default "Specify predefined Datasource"
- Database Vendor is pre-populated with value DB2 (If you were using Oracle, or MSSQLServer, you see those as specific adapter processing is available with those specific databases.)
- 3) Enter jdbc/JDBCTEST\_DS for Data source JNDI name

Specify the Service Generation and Deployment Properties         Specify properties for generating the service and running it on the server.         Service Operations         To modify the names, or add a description to the operations to be generated in the interface file, click Call Operations.         Deployment Properties         How do you want to specify the security credentials?         O Using an existing JAAS alias (recommended)         A Java Authentication and Authorization Services (JAAS) alias is the preferred method.         J2C authentication data entry:         O Using security properties from the activation specification         The properties will be stored as plain text; no encryption is used.         User name:       db2admin         Password:       *********         Deploy connector project:       With module for use by single application         Specify the settings used to connect to JDBC at run time:       Connection properties         Deploy connector project:       Specify connection properties         Deploy connector information:       Specify predefined DataSource         Database connection information:       Specify predefined DataSource         Database vendor:       DB2	Specify the Service Generation and Deploy Specify properties for generating the service and runnin Service Operations To modify the names, or add a description to the op interface file, click Edit Operations. Deployment Properties How do you want to specify the security credentials Ousing an existing JAAS alias (recommended) A Java Authentication and Authorization Service J2C authentication data entry: Ousing security properties from the activation spe The properties will be stored as plain text; no en User name: Password:	ment Properties Ig it on the server: Perations to be generated in the Edit Operations Provide the preferred method, Edit Operations Provide the preferred method, Edit Operations
Specify properties for generating the service and running it on the server.       Image: Construction of the properties of the click Call Operations.         Service Operations       Edit Operations.         To modify the names, or add a description to the operations to be generated in the interface file, click Call Operations.       Edit Operations.         Deployment Properties       Edit Operations of the security credentials?       Image: Construction of Authorization Services (JAAS) alias is the preferred method.         J2C: authentication data entry:       Image: Construction of the activation specification       The properties from the activation specification         The properties will be stored as plain text; no encryption is used.       Image: Construction of the properties of the security is required or will be handled by the EIS system, or the RAR will be deployed on the server and security will be specified by the properties in the JNDI lookup name.         Deploy connector project:       With module for use by single application         Specify the settings:       Specify connection properties         Connection settings:       Specify connection properties         Connection Properties       Specify predefined DataSource         Database connection information:       Specify predefined DataSource         Database vendor:       DBZ	Specify properties for generating the service and running Service Operations To modify the names, or add a description to the op interface file, click Edit Operations. Deployment Properties How do you want to specify the security credentials O Using an existing JAAS alias (recommended) A Java Authentication and Authorization Service J2C authentication data entry: O Using security properties from the activation spe The properties will be stored as plain text; no en User name: Password:	ig it on the server:
Service Operations To modify the names, or add a description to the operations to be generated in the Edit Operations Deployment Properties How do you want to specify the security credentials? Using an existing JAAS alias (recommended) A Java Authentication and Authorization Services (JAAS) alias is the preferred method. J2C authentication data entry: Using security properties from the activation specification The properties will be stored as plain text; no encryption is used. User name: <a href="https://www.usertext">db2admin</a> Password: <a href="https://www.usertext">www.usertext</a> Deploy connector project: <a href="https://www.usertext">With module for use by single application</a> <a href="https://www.usertext">www.usertext</a> Deploy connector project: <a href="https://with.module.com">With module for use by single application</a> <a href="https://with.module.com">w/with.module.com</a> Deploy connector project:   Deploy connection information:  Specify predefined DataSource  Database connection information:  Database vendor:  DB2	Service Operations To modify the names, or add a description to the op interface file, click Edit Operations. Deployment Properties How do you want to specify the security credentials Using an existing JAAS alias (recommended) A Java Authentication and Authorization Service J2C authentication data entry: Using security properties from the activation spe The properties will be stored as plain text; no en User name: Password:	erations to be generated in the Edit Operations
To modify the names, or add a description to the operations to be generated in the interface file, click Edit Operations. Deployment Properties How do you want to specify the security credentials? Using an existing JAAS alias (recommended) A Java Authentication and Authorization Services (JAAS) alias is the preferred method. J2C authentication data entry: Using security properties from the activation specification The properties will be stored as plain text; no encryption is used. User name: do2admin Password: ******** Use if no security is required or will be handled by the EIS system, or the RAR will be deployed on the server and security will be specified by the properties in the JNDI lookup name. Deploy connector project: With module for use by single application Specify the settings: Connection settings: Specify predefined DataSource Database connection information: Database vendor: DE2	To modify the names, or add a description to the opinterface file, click Edit Operations. Deployment Properties How do you want to specify the security credentials Using an existing JAAS alias (recommended) A Java Authentication and Authorization Service J2C authentication data entry: Using security properties from the activation spectrum user name: User name: Password:	erations to be generated in the Edit Operations
Deployment Properties How do you want to specify the security credentials? Using an existing JAAS alias (recommended) A Java Authentication and Authorization Services (JAAS) alias is the preferred method. J2C authentication data entry: Using security properties from the activation specification The properties will be stored as plain text; no encryption is used. User name: db2admin Password:  © Other Use if no security is required or will be handled by the EIS system, or the RAR will be deployed on the server and security will be specified by the properties in the JNDI lookup name. Deploy connector project: With module for use by single application Specify the settings used to connect to JDBC at run time: Connection settings: Database connection information: Database vendor: DB2	Deployment Properties         How do you want to specify the security credentials         Using an existing JAAS alias (recommended)         A Java Authentication and Authorization Service         J2C authentication data entry:         Using security properties from the activation spectrum         The properties will be stored as plain text; no end         User name:       db2admin         Password!	? es (JAAS) alias is the preferred method, cification heryption is used.
How do you want to specify the security credentials? O Using an existing JAAS alias (recommended) A Java Authentication and Authonization Services (JAAS) alias is the preferred method. J2C authentication data entry: O Using security properties from the activation specification The properties will be stored as plain text; no encryption is used. User name: Desword: ******** O Other Use if no security is required or will be handled by the EIS system, or the RAR will be deployed on the server and security will be specified by the properties in the JNDI lookup name. Deploy connector project: With module for use by single application Specify the settings used to connect to JDBC at run time: Connection settings: Database connection information: Database vendor: Database vendor: DE2	How do you want to specify the security credentials         Using an existing JAAS alias (recommended)         A Java Authentication and Authorization Service         J2C authentication data entry;         Using security properties from the activation spectrum         The properties will be stored as plain text; no end         User name:         Password:	2 es (JAAS) alias is the preferred method; cification acryption is used.
Ousing an existing JAAS alias (recommended)         A Java Authentication and Authorization Services (JAAS) alias is the preferred method.         J2C authentication data entry:         Ousing security properties from the activation specification         The properties will be stored as plain text; no encryption is used.         User name:       db2admin         Password:       *********         O Other         Use if no security is required or will be handled by the EIS system, or the RAR will be deployed on the server and security will be specified by the properties in the JNDI lookup name.         Deploy connector project:       With module for use by single application         Specify the settings used to connect to JDBC at run time:       Connection settings:         Connection Properties       Specify predefined DataSource         Database connection information:       Specify predefined DataSource         Database vendor:       DB2	<ul> <li>Using an existing JAAS alias (recommended)</li> <li>A Java Authentication and Authorization Service</li> <li>J2C authentication data entry;</li> <li>Using security properties from the activation spectrum</li> <li>The properties will be stored as plain text; no end</li> <li>User name;</li> <li>Bassword;</li> </ul>	es (JAAS) alias is the preferred method. cification acryption is used.
A Java Authentication and Authorization Services (JRAS) alias is the preferred method. J2C authentication data entry: O Using security properties from the activation specification The properties will be stored as plain text; no encryption is used. User name: db2admin Password: ******** O Other Use if no security is required or will be handled by the EIS system, or the RAR will be deployed on the server and security will be specified by the properties in the JNDI lookup name. Deploy connector project: With module for use by single application Specify the settings used to connect to JDBC at run time: Connection settings: Specify connection properties Database connection information: Database vendor: DB2	A Java Authentication and Authorization Service J2C authentication data entry: OUsing security properties from the activation spe The properties will be stored as plain text; no en User name: Password: ********	es (JAAS) alias is the preferred method. cification ncryption is used.
J2C authentication data entry:         O Using security properties from the activation specification         The properties will be stored as plain text; no encryption is used.         User name:       db2admin         Password:       ********         O Other         Use if no security is required or will be handled by the EIS system, or the RAR will be deployed on the server and security will be specified by the properties in the JNDI lookup name.         Deploy connector project:       With module for use by single application         Specify the settings used to connect to JDBC at run time:       ✓         Connection settings:       Specify connection properties         Database connection information:       Specify predefined DataSource         Database vendor:       DB2	J2C authentication data entry:          O Using security properties from the activation spe         The properties will be stored as plain text; no end         User name:       db2admin         Password:       **********	cification acryption is used.
◇ Using security properties from the activation specification         The properties will be stored as plain text; no encryption is used.         User name:       db2admin         Password:       ********         ◇ Other         Use if no security is required or will be handled by the EIS system, or the RAR will be deployed on the server and security will be specified by the properties in the JNDI lookup name.         Deploy connector project:       With module for use by single application         Specify the settings used to connect to JDBC at run time:       ✓         Connection Properties       ✓         Database connection information:       Specify predefined DataSource         Database system connection information:       DB2	O Using security properties from the activation spectrum The properties will be stored as plain text; no en User name: Password: *********	cification acryption is used.
The properties will be stored as plain text; no encryption is used. User name: db2admin Password: ******** Other Use if no security is required or will be handled by the EIS system, or the RAR will be deployed on the server and security will be specified by the properties in the JNDI lookup name. Deploy connector project: With module for use by single application  Specify the settings used to connect to JDBC at run time: Connection settings: Specify connection properties Connection Properties Database connection information: Specify predefined DataSource Database system connection information Database vendor: DB2	The properties will be stored as plain text; no er User name: db2admin Password: ********	horyption is used.
User name: db2admin Password:	User name: db2admin Password:	
Password:       ********         Other       Use if no security is required or will be handled by the EIS system, or the RAR will be deployed on the server and security will be specified by the properties in the JNDI lookup name.         Deploy connector project:       With module for use by single application         Specify the settings used to connect to JDBC at run time:         Connection settings:       Specify connection properties         Database connection information:       Specify predefined DataSource         Database vendor:       DB2	Password:	
Other   Use if no security is required or will be handled by the EIS system, or the RAR will be deployed on the server and security will be specified by the properties in the JNDI lookup name.   Deploy connector project: With module for use by single application   Specify the settings used to connect to JDBC at run time:   Connection settings:   Specify connection properties   Database connection information:   Specify predefined DataSource   Database system connection information   Database vendor:		
Deploy connector project: With module for use by single application Specify the settings used to connect to JDBC at run time: Connection settings: Specify connection properties Connection Properties Database connection information: Specify predefined DataSource Database system connection information Database vendor: DB2	server and security will be specified by the prop	erties in the JNDI lookup name.
Specify the settings used to connect to JDBC at run time: Connection settings: Connection Properties Database connection information: Database system connection information Database vendor: DB2	Deploy connector project: With module 1	for use by single application
Connection settings: Specify connection properties Connection Properties Database connection information: Specify predefined DataSource Database system connection information Database vendor: DB2	Specify the settings used to connect to JDBC at run	time:
Connection Properties Database connection information: Specify predefined DataSource Database system connection information Database vendor: DB2	Connection settings: Specify conne	ction properties 💉 👻
Database connection information: Specify predefined DataSource	Connection Properties	
Database system connection information Database vendor: DB2	Database connection information: Specify pre	defined DataSource
Database vendor: DB2	Database system connection information	
	Database vendor:	
DataSource JNDI name: * jdbc/JDBCTEST_DS	DataSource JNDI name: * jdbc/JDBC	rest_ds
	Advanced SS	
	Database connection information: Specify pre Database system connection information Database vendor: DB2	

- \_\_\_\_d. Expand by clicking on Advanced button for additional properties options
- \_\_\_\_e. Expand Event delivery configuration
  - 1) Verify the Ensure once-only event delivery is checked
  - 2) Enter JDBC001 for event filtering
- \_\_\_\_ f. Expand Event Configuration
  - 1) Event Order By: event\_time, event\_priority
  - 2) Event Table Name: WBIA\_JDBC\_EventStore

Event polling configuration		
Event delivery configuration		
Type of delivery:	ORDERED	
Ensure assured-once event delivery	y (may reduce performance)	
Do not process events that have a	time stamp in the future	
Event types to process:		
Adapter Instance for event filtering:	JDBC001	
Retry limit for failed events:	5	
Number of connections for event delive	ery	
Minimum:	1	
Maximum:	1	
Additional connection configuration		
· Event configuration	4 <b>1</b>	
Event order by:	event_time, event_priority	
Event table name:	* WBIA_JDBC_EventStore	
Stored procedure to run before poll:		
Stored procedure to run after poll:		
Event query type for processing event	s: Standard	
User-defined event query:		
User-defined update query:		
User-defined delete query:		_

### \_\_\_\_g. Expand Logging and Tracing

 Optional - 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 adapterspecific identifier, MyAdapterRA, to form the component name used by Log and Trace Analyzer. 2) Optional - 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.

Dogging and tra	acing
Adapter ID:*	001
Disguise us	er data as "XXX" in log and trace files.

- \_\_\_h. Click Next
- \_\_\_\_ 11. Complete **Publishing Properties** panel
  - \_\_\_\_a. A Business Integration Module has not yet been created, select the **New** button and enter in the name **JDBCTestInboundWrapper** for the Module Name. Click **Ok**.
  - \_\_\_\_ b. Leave the default JDBCInboundInterface for the Name value.

pecify location pr	operties for where you want to save the service.	H
roperties for Serv	vice	-0
Module:	JDBCTestInboundWrapper	New
Namespace;	http://JDBCTestInboundWrapper/JDBCInboundInterface	
	Use the default namespace	
Folder:		Browse,
Name: *	JDBCInboundInterface	
	Save business objects to a library	
Library:		New
Description:		1
	2.	

- \_\_\_ c. Click Finish
- 12. Use the Assembly Diagram to wire the **JDBCTestInboundWrapper** Interface to a Java Component.
  - \_\_\_\_a. Expand the JDBCTestInBoundWrapper folder.
  - \_\_\_\_b. From the **Business Integration** view, double click the **JDBCTestInBoundWrapper** module. This will open the module in the Assembly Diagram.

\_\_\_\_ c. From the palette, expand **Component**, then select the **Java Component** and drop it on the Assembly Diagram.



\_\_\_\_\_d. Wire the **JDBCInboundInterface** to the **Java Component.** At the Add Wire popup window, select **OK**.

🔂 Ado	d Wire		×
?	This action will allow the target service to be u service interface from the export will be added continue?	ised in other modules. The d to the target. Do you want	to
Alwa	ays create without prompt		
		OK Cancel	

\_\_\_\_ e. The Assembly Diagram now looks as follows:



\_\_\_\_13. Generate the implementation for the Java Component



\_\_\_\_a. Right click the Java Component1, select Generate Implementation from the pop-up menu

- \_\_\_ b. Highlight the **default** package and select **OK** (Click New Package and create new package if you want)
- \_\_\_\_ c. The Java Editor will open with the **Component1Impl.java** file.

**NOTE:** For your convenience, these code snippets can be found in <LAB\_FILES>\JDBC\snippets\Component1Impl.txt

\_\_\_\_\_ d. Scroll down and locate the createWrapperBG (Object createWrapperBGInput) method that needs to be implemented. Paste these code into the method so the complete method looks as follows:

```
public void createWrapperBG(DataObject createWrapperBGInput) {
   DataObject wrapper = createWrapperBGInput.getDataObject("Wrapper");
   System.out.println("-----");
   System.out.println("Wrapper was created.");
   DataObject customer = (DataObject) wrapper.getList("customerobj").get(0);
   DataObject addr = (DataObject) wrapper.getList("addressobj").get(0);
   System.out.println("CUSTOMER info as below:");
   System.out.println("PKEY is: "+ customer.getString("pkey"));
   System.out.println("FNAME is: "+ customer.getString("fname"));
   System.out.println("LNAME is: "+ customer.getString("lname"));
   System.out.println("CCODE is: "+ customer.getString("ccode"));
   System.out.println();
   System.out.println("CUSTOMER ADDRESS info as below:");
   System.out.println("ADDRID is: "+ addr.getString("addrid"));
   System.out.println("CUSTID is: "+ addr.getString("custid"));
   System.out.println("CITY is: "+ addr.getString("city"));
   System.out.println("ZIPCODE is: "+ addr.getString("zipcode"));
   System.out.println("-----");
   System.out.println();
```

\_\_ e. Scroll down and locate the deleteWrapperBG (Object deleteWrapperBGInput) method that needs to be implemented. Paste these code into the method so the complete method looks as follows:

```
public void deleteWrapperBG(DataObject deleteWrapperBGInput) {
   DataObject wrapper = deleteWrapperBGInput.getDataObject("Wrapper");
   System.out.println("-----");
   System.out.println("Wrapper was deleted.");
   System.out.println("PKEY is: "+ wrapper.getString("wrapcustomerpkey"));
   System.out.println("ADDRESS is: "+ wrapper.getString("wrapaddressaddrid"));
   System.out.println("-----");
   System.out.println();
   }
   ____f. Scroll down and locate the updateWrapperBG (Object updateWrapperBGInput) method that
       needs to be implemented. Paste these code into the method so the complete method looks as
       follows:
public void updateWrapperBG(DataObject updateWrapperBGInput) {
   DataObject wrapper = updateWrapperBGInput.getDataObject("Wrapper");
   System.out.println("-----");
   System.out.println("Wrapper was updated.");
   DataObject customer = (DataObject) wrapper.getList("customerobj").get(0);
   DataObject addr = (DataObject) wrapper.getList("addressobj").get(0);
   System.out.println("CUSTOMER info as below:");
   System.out.println("PKEY is: "+ customer.getString("pkey"));
   System.out.println("FNAME is: "+ customer.getString("fname"));
   System.out.println("LNAME is: "+ customer.getString("lname"));
   System.out.println("CCODE is: "+ customer.getString("ccode"));
   System.out.println();
   System.out.println("CUSTOMER ADDRESS info as below:");
   System.out.println("ADDRID is: "+ addr.getString("addrid"));
   System.out.println("CUSTID is: "+ addr.getString("custid"));
   System.out.println("CITY is: "+ addr.getString("city"));
   System.out.println("ZIPCODE is: "+ addr.getString("zipcode"));
   System.out.println("-----");
   System.out.println();
}
```

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

\_\_\_h. Wait for the workspace to complete building. Close the Assembly Diagram.

## Part 4: Create the authentication alias and configure the data source

Note: If you have already completed this section in any of JDBC Outbound labs, you can skip this and proceed to Part 5.

In this part, the authentication alias needs to be set since the data source uses the username/password set in the authentication alias to connect to the database. This data source connects to the database and is used later when generating the artifacts for the module. Here are the steps to set the authentication alias in WebSphere Process Server administrative console.

- 1. Switch to Servers view by selecting Windows  $\rightarrow$  Show View  $\rightarrow$  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 **Administration** →**Run Administrative console**

		Manage server profiles Server configuration Universal test client		•	
		Administration		•	Run administrative console
		Launch Add and Remove Integration Solutio	n Projects	;	WebSphere administration command assist Run administrative script
Task Flows Co Build Activities	🗆 Properties 🔀 Prot	WebSphere Business Monitor Event	Recording	•	
Server 🔺	State	Properties	Alt+Enter		

- \_\_\_\_ c. Log in to the administrative console by clicking the "Log in" button
- \_\_\_\_\_d. Click Security → Global Security
- \_\_\_\_e. On the right, expand Java Authentication and Authorization Service under the Authentication heading.

e this panel to configure administration and the default application and is used as a default security policy for user application objections	in security policy. This security configuration applies to the security policy for all administr is. Security domains can be defined to override and customize the security policies for us
Security Configuration Wizard Security	/ Configuration Report
Idministrative security         Administrative user roles           Image: State administrative security         Administrative croup roles           Administrative administrative authentication         Administrative authentication	Authentication Authentication mechanisms and expiration
Application security	Kerberos configuration
Enable application security	SWAM (deprecated): No authenticated communication between servers
ava 2 security	Addrend cache Security
Use Java 2 security to restrict application access to local resour	rces
☑ Warn if applications are granted custom permissions	RMI/IIOP security     RMI/IIOP security
Restrict access to resource authentication data	Java Authentication and Authorization Service
	Use realm-qualified user names
lser account repository	
Current realm definition	
recerated repositories	<ul> <li>Security domains</li> <li>External authorization providers</li> </ul>
Available realm definitions	Custom properties
Federated repositories 🛛 🖌 Configure Set as cu	irrent

- \_\_\_\_\_f. Click **J2C authentication alias**. It gives the list of existing aliases.
  - a) Click **New** and type in alias name, user ID, and password to connect to the database. Click **Ok**

eral Properties	identities and passwords for Java (149) 2 connector security to use.
Alias	
b2_Auth	
Jser ID	
lb2admin	
Password	
escription	
tin.	

\_\_\_\_ 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. Select Node
  - \_\_\_\_ c. On the right, click New and choose these following values
    - 1) Database type **DB2**
    - 2) Provider type DB2 Universal JDBC Driver Provider
    - 3) Implementation type: XA data source



\_\_\_\_\_ d. Click **Next** and enter the database class path information



\_\_\_\_e. Click Next

- \_\_\_\_ f. Click **Finish** in the summary page.
- \_\_\_\_ g. Click **Save** to 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 DB2 Universal JDBC Driver XA Data source
    - 2) JNDI name: jdbc/JDBCTEST\_DS
  - \_\_\_\_ c. Click Next to Step 2
  - \_\_\_\_ d. In Database specific properties for the data source, enter these values
    - 1) Database name JDBCTEST
    - 2) Server name LOCALHOST
    - 3) Port number accept the default (50000)
  - \_\_\_\_e. Click Next to Step 3
  - \_\_\_\_f. Set the Component-managed authentication alias to the one created in the earlier section.
  - \_\_\_\_g. Click **Next** and view the Summary

Step 1: Enter basic data source information Step 2: Enter	Summary	
	Summary of actions:	
	Options	Values
properties for the	Scope	cells:qcell:nodes:qnode
data source Step 3: Setup	Data source name	DB2 Universal JDBC Driver XA DataSource
security aliases	JNDI name	jdbc/JDBCTEST_DS
Step 4: Summary	Select an existing JDBC provider	DB2 Universal JDBC Driver Provider (XA)
	Implementation class name	com.ibm.db2.jcc.DB2XADataSource
	Driver type	4
	Database name	JDBCTEST
	Server name	localhost
	Port number	50000
	Use this data source in container managed persistence (CMP)	true
	Authentication alias for XA recovery	(none)
	Component-managed authentication alias	qnode/db2_Auth
	Mapping-configuration alias	(none)
	Container-managed	(none)

- \_\_\_\_h. Click **Finish** and save the changes.
- \_\_\_\_\_ 5. Test the JDBC Data Source connection
  - \_\_\_\_a. Check the box next to DB2 Universal JDBC Driver XA DataSource and click on Test connection from the top of the screen.

New	Delete Tes	t connection M	anage state			
Q	ð # \$					
Select	Name 🛟	JNDI name 🗘	Scope 🗘	Provider 🗘	Description 🗘	Category 🔿
You c	an administer th	e following resources	51		·	
	DB2 Universal JDBC Driver XA DataSource	jdbc/JDBCTEST_DS	Node=qnode	DB2 Universal JDBC Driver Provider (XA)	DB2 Universal Driver Datasource	

\_\_\_\_b. You should be these success message on the top of the screen

### 🗄 Messages

The test connection operation for data source DB2 Universal JDBC Driver XA DataSource on server server1 at node gnode was successful.

## Part 5: Test the application using the WebSphere test environment

In this part you will use the WebSphere Test Environment to test the SCA application event processing by starting the application and verifying that it polls for and picks up the Create event already waiting in the WBIA\_JDBC\_EVENTSTORE table.

- 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 click Server and select Start.
  - \_\_\_\_b. Right click the server in the server view and select Add and remove projects ...

		👆 Add and Remove Projects	
		Monitoring	() e (
		Create tables and data sources Reconnect debug process	
		😤 View and publish the changes to the server	
		Manage server profiles	
		Server configuration	2 <b>1</b> 12
		Universal test client	•
		Administration	•
		Launch	
(nn) (n)		Add and Remove Integration Solution Projects	
Task Flows Co Build	Activities 🗔 Properties 🛣	P WebSphere Business Monitor Event Recording	
Server A	State	Properties	Alt+Enter
Izen websphere Proc	ess server v7.0 De Started	aynu i unizeu	

\_\_\_\_ c. In the Add and Remove Projects dialog, select the **JDBCTestInboundApp** project from the Available projects panel.

Add and Remove Projects			
Add and Remove Projects Modify the projects that are configur	ed on the server		
Move projects to the right to configu	re them on the server		
<u>Available projects:</u>		Configured projects:	
	A <u>d</u> d > < <u>R</u> emove		oundApp
	Add All >> <		
0	< <u>B</u> ack <u>N</u> ext >	<u> </u>	Cancel

\_\_\_\_\_d. Click Add > to add it to the Configured projects panel. Click Finish

- \_\_\_\_e. In Business Integration view, right-click the JDBCTestInboundWrapper module, and select **Test** → Attach
- \_\_\_\_f. This will open Integration Test Client

Events	General Properties		
This area displays the events in a test trace. Select an event to display its properties in the			
General Properties and Detailed Properties sections, More	Select a test configuration for the Attach event, then dick the Continue icon in the Events area to start the attachmession. <u>Morecue</u> Configuration: Default Module Test Module: <u>JDBCTestInboundWrapper</u>		
g. Click b to start new service			
h. Open JDBCTEST database through DB2	command editor		

```
INSERT INTO WBIA_JDBC_EVENTSTORE
(OBJECT_KEY,OBJECT_NAME,OBJECT_FUNCTION,EVENT_PRIORITY,EVEN
T_STATUS, CONNECTOR_ID) VALUES
('A1;C1','WrapperBG','Create',1,0,'JDBC001');
INSERT INTO WBIA_JDBC_EVENTSTORE
(OBJECT_KEY,OBJECT_NAME,OBJECT_FUNCTION,EVENT_PRIORITY,EVEN
T_STATUS, CONNECTOR_ID) VALUES
('A2;C2','WrapperBG','Update',1,0,'JDBC001');
INSERT INTO WBIA_JDBC_EVENTSTORE
(OBJECT_KEY,OBJECT_NAME,OBJECT_FUNCTION,EVENT_PRIORITY,EVEN
T_STATUS, CONNECTOR_ID) VALUES
('A3;C3','WrapperBG','Delete',1,0,'JDBC001');
```

**Note:** In a real environment, a trigger or another application, which can access the database, can insert the event record.



\_\_\_\_i. Review the output of the service from test client or from server log

\_\_\_\_j. Repeat steps a through c to remove the project from the server.

## What you did in this exercise

In this exercise, you learned how to install and deploy the WebSphere Adapter for JDBC. You also were introduced to wrapper business object in inbound processing. You have learned how to retrieve records from multiple tables with one event entry using wrapper business object feature.