



IBM Connect for iSeries 1.1

JDBC Connector Sample

September, 2001

Contents

- [License and disclaimer](#)
 - [Overview](#)
 - [Set up the sample](#)
 - [Prerequisites](#)
 - [Install the sample](#)
 - [Installed directories and files](#)
 - [Configure the sample](#)
 - [Run the sample](#)
 - [Appendix: JDBC driver properties](#)
-

License and disclaimer

This material contains IBM copyrighted sample programming source code ("Sample Code"). IBM grants you a nonexclusive license to compile, link, execute, display, reproduce, distribute and prepare derivative works of this Sample Code. The Sample Code has not been thoroughly tested under all conditions. IBM, therefore, does not guarantee or imply its reliability, serviceability, or function. IBM provides no program services for the Sample Code.

All Sample Code contained herein is provided to you "AS IS" without any warranties of any kind. THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT ARE EXPRESSLY DISCLAIMED. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO THE ABOVE EXCLUSIONS MAY NOT APPLY TO YOU. IN NO EVENT WILL IBM BE LIABLE TO ANY PARTY FOR ANY DIRECT, INDIRECT, SPECIAL OR OTHER CONSEQUENTIAL DAMAGES FOR ANY USE OF THE SAMPLE CODE INCLUDING, WITHOUT LIMITATION, ANY LOST PROFITS, BUSINESS INTERRUPTION, LOSS OF PROGRAMS OR OTHER DATA ON YOUR INFORMATION HANDLING SYSTEM OR OTHERWISE, EVEN IF WE ARE EXPRESSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

(C) Copyright IBM CORP. 2001

All rights reserved.

US Government Users Restricted Rights -- Use, duplication, or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Overview

This sample documentation demonstrates how to use a JDBC connector to access a customer database in a simple multiple step flow. The objective is to access a local table to map the Buyer ID field in an incoming cXML request

(From/Credential) to the Customer ID field that is required by a back-end application. Figure 1 shows an overview of how an Order Request message flows through two-step process flow in this sample.

1. When the flow manager receives an OrderRequest cXML message, it forwards the message to the JDBC connector (1).
2. When the process flow step 1 is executed, the request field /Header/From/Credential is mapped into the BuyerID field. The JDBC connector executes an SQL SELECT statement that retrieves the corresponding CustomerID from the CUSTOMER table and stores it in the intermediate data area (2). The CUSTOMER table provides mapping between BuyerIDs (usually DUNS numbers) that flow in the cXML OrderRequest and CustomerIDs required by the Order Processing application.
3. Process flow step 2 runs the Java connector application, JavaCustId (3). The CustomerID in the intermediate data area and the original cXML request are now available to the Java connector application, which processes the request.

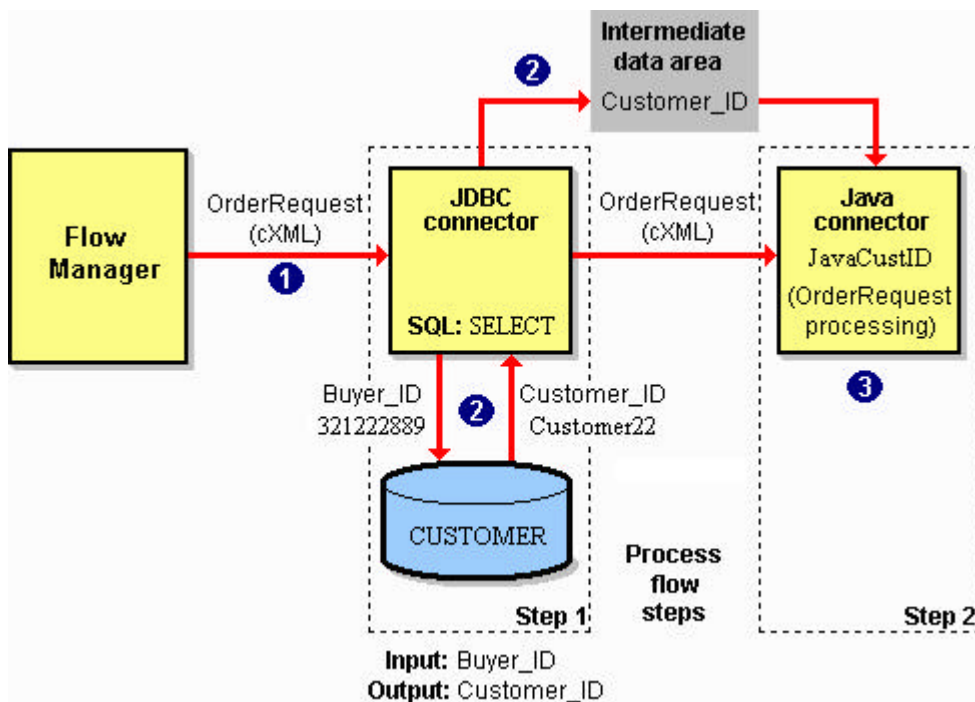


Figure 1. Overview of how a JDBC connector is used to retrieve a customer ID in this sample.

Refer to IBM Connect for iSeries 1.1 documentation in the V5R1 Information Center for a complete description of JDBC connector:

<http://publib.boulder.ibm.com/pubs/html/as400/v5r1/ic2924/index.htm?info/rzald/1.1/conjjdbc.htm>

[Top](#)

Set up the sample

- [Prerequisites](#)
- [Install the sample](#)
- [Installed objects](#)

Prerequisites for running the sample

Before running this sample, perform the following tasks:

1. Configure and run the PCML verification sample as described at:

<http://publib.boulder.ibm.com/pubs/html/as400/v5r1/ic2924/index.htm?info/rzald/1.1/pcmlver.htm>

As a result of running the sample, there is a Connect instance named samplnst on your system.

2. Add the cXML 1.2 protocol to the Ariba Network marketplace in samplnst.
3. Associate the supplier with the cXML Ariba 1.2 protocol.
4. Register three new buyers, following these generic instructions:

<http://publib.boulder.ibm.com/pubs/html/as400/v5r1/ic2924/index.htm?info/rzald/1.1/pcmlbuy.htm>

Each new buyer should have a unique Buyer ID (DUNS number). Use these DUNS numbers for your new buyers:

- 321222888
- 321222889
- 321222890

These Buyer IDs are mapped to the sample Customer IDs in the CUSTOMER table that is shipped with this sample.

[Top](#)

Install the sample

To install the sample programs follow the steps below:

1. Download the package file [JDBCSam 110.exe](#).
2. Map a network drive to the integrated file system root directory on the iSeries system on which you want to install the sample programs.
3. Double-click the package file JDBCSam_110.exe to extract its content to the mapped integrated file system drive, in the directory /CONNSAM11/JDBCSam1.
4. Create the library CONNSAM11 if it does not exist from running a previous sample. Enter this command:

```
CRTLIB LIB(CONNSAM11) TEXT('iSeries Connect 1.1 sample programs')
```

5. Copy the CUSTOMER save file (*SAVF) to the CONNSAM11 library using this command:

```
CPYFRMSTMF FROMSTMF('/CONNSAM11/JDBCSam1/Application/customersf.savf')
           TOMBR('/qsys.lib/CONNSAM11.lib/customersf.file')
```

6. Restore the contents of the save file into the CONNSAM11 library using this following command:

```
RSTOBJ OBJ(*ALL) SAVLIB(CONNSAM11) DEV(*SAVF) SAVF(CONNSAM11/CUSTOMERSF)
      MBROPT(*NEW) ALWOBJDIF(*ALL)
```

7. Copy these files from the sample directory /CONNSAM11/JDBCSam1/BPWData to your Connect instance directory /QIBM/UserData/Connect110/sampInst/Connector:
 - JavaCustID.pcmI
 - JavaCustID.AppConnector

8. Grant user *PUBLIC read, write, and execute (*RWX) data authority to the directory /CONNSAM11/JDBCSam1. Enter this command:

```
CHGAUT OBJ(' /CONNSAM11/JDBCSam1 ' ) USER( *PUBLIC ) DTAAUT( *RWX )
```

9. Grant user *PUBLIC read, and execute (*RX) data authority to the JavaCustID.class and JavaCustID.properties files. Enter these commands:

```
CHGAUT OBJ(' /CONNSAM11/JDBCSam1/Application/JavaCustID.class ' )  
      USER( *PUBLIC ) DTAAUT( *RX )
```

```
CHGAUT OBJ(' /CONNSAM11/JDBCSam1/Application/JavaCustID.properties ' )  
      USER( *PUBLIC ) DTAAUT( *RX )
```

Note: To verify that the objects have the proper authority use the Display Authority (DSPAUT) command. For example:

```
DSPAUT OBJ(' /connsam11/JDBCSam1 ' )
```

[Top](#)

Installed objects

The JDBC connector sample package contains these objects:

/CONNSAM11/JDBCSam1/Application directory	
File	Description
JavaCustID.java	Java connector application used in step 2 of the process flow to simulate the order request processing application.
JavaCustID.class	Compiled Java class.
JavaCustID.properties	Properties file made available to the sample JavaCustID class. This properties file is used to specify the path and name of the log file that the JavaCustID class uses to log application activity. The default log file is /CONNSAM11/JDBCSam1/Results/JavaCustID.log
/CONNSAM11/JDBCSam1/BPWDData directory	
File	Description
FindCustomer.AppConnector	Sample application connector (connector type JDBC) used in step 1 of the process flow.
JavaCustID.pcml	Sample PCML file that defines the fields in the request that the back-end Java connector application expects. Used in step 2 of the process flow.
JavaCustID.AppConnector	Sample application connector (connector type Java) used in step 2 of the process flow.
JDBCTestFlow.ProcessFlow	Sample two-step process flow. Step 1 shows JDBC connector how to access the CUSTOMER table and pass results to the intermediate data area. Step 2 runs the Java connector application to process the request.
/CONNSAM11/JDBCSam1/Results directory	
This directory includes the log file that is generated by the sample JavaCustID Java connector application.	
CONNSAM11 library	
Object	Description
CUSTOMER (table)	Sample table provides the mapping of BuyerID to CustomerID.

[Top](#)

Configure the sample

To configure the JDBC connector sample, perform these steps:

1. [Configure the JDBC application connector](#)
2. [Configure the multiple-step process flow](#)
3. [Deploy the process flow](#)
4. [Update the flow manager classpath for the sample application](#)

Configure the JDBC application connector

These instructions guide you through creating your own application connector. You may instead use the FindCustomer.AppConnector included with the sample instead of creating your own. If you prefer to use the sample JDBC application connector, copy the sample file FindCustomer.AppConnector from the /ConnSam11/JDBCSam1/BPWDData directory to the /QIBM/UserData/Connect110/SampInst/Connector directory and skip to [Configure the multiple-step process flow](#).

To configure the JDBC application connector FindCustomer.AppConnector for this sample, perform these steps:

1. With a Web browser such as Netscape 4.7 or Internet Explorer 5.0 (or later), access the iSeries Connect configuration tool. In your browser, enter the appropriate URL (based on your version of OS/400) listed below:
 - **V4R5:** `http://your_host_name:2002/BtoB/Connect`
 - **V5R1:** `http://your_host_name:2001/BtoB/Connect`

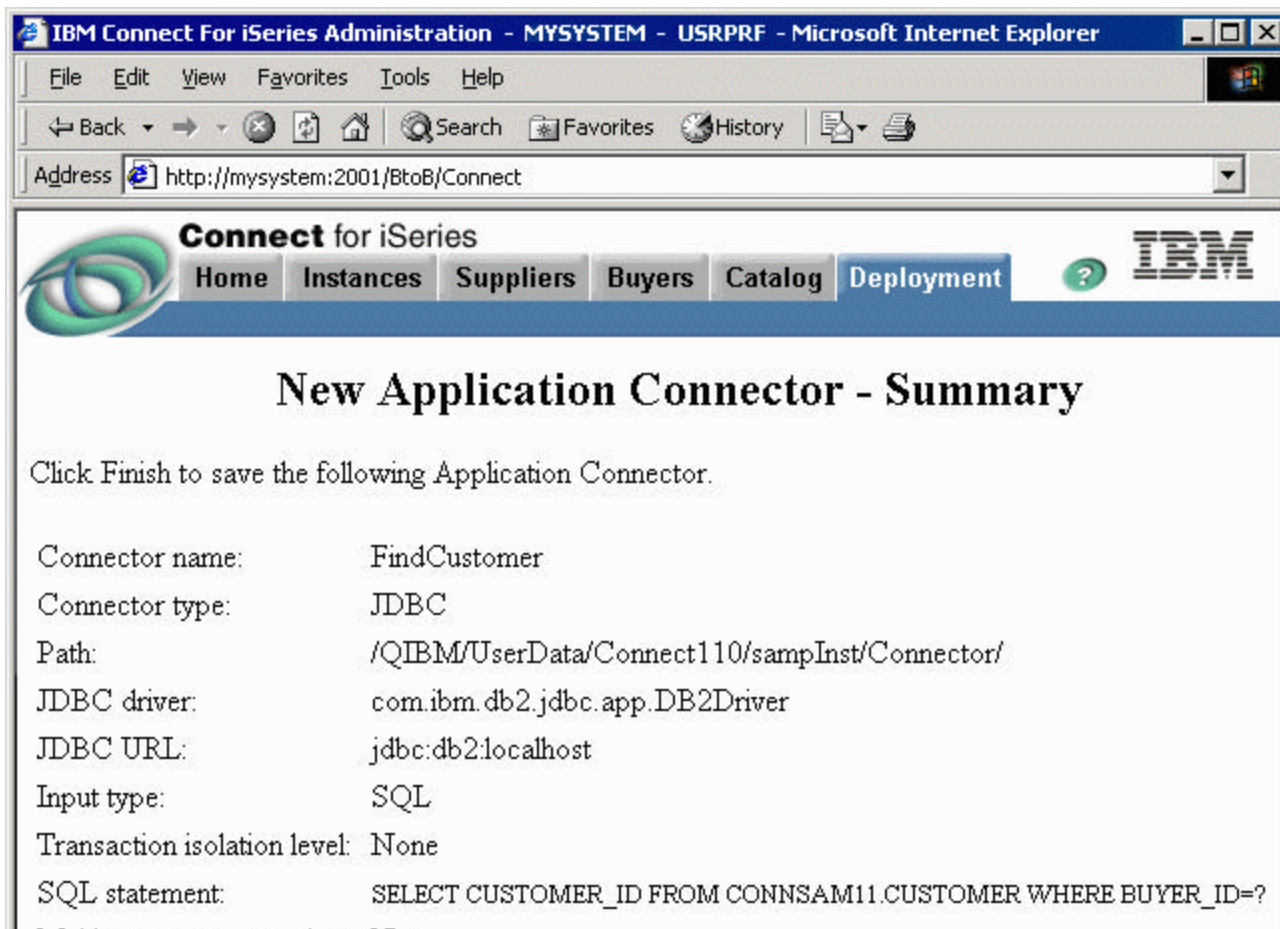
where *your_host_name* in the host name of your iSeries server.

2. Click the **Instances** tab and select the **samplnst** B2B instance. If samplnst is not listed, you have not run the [prerequisite sample](#).
3. Click the **Deployment** tab.
4. On the Deployment page, select the Marketplace (**Ariba Network** if you are using samplnst). Click **Application Connectors**.
5. On the Application Connectors page, select **Create a new application connector document** and then click **Next**.
6. To configure the new application connector, enter or select these values:

Screen / action	Field	Value
New Application Connector	<i>Connector name</i>	FindCustomer
	<i>Connector type</i>	JDBC
	<i>Path</i>	(Leave blank.) Note: This is the directory where the application connector document is saved.
Click Next .		
New JDBC Connector	<i>SQL statement</i>	SELECT CUSTOMER_ID FROM CONNSAM11.CUSTOMER WHERE BUYER_ID=? Note: Input fields represent parameter markers and are designated with ?.
	<i>Multi row result set</i>	No
Click Next .		
New JDBC Connector	<i>Database user</i>	(User ID used to connect to the database to gather the field information from the SQL statement that is provided. The specified user must have enough authority to run the SQL statement.)
	<i>Database password</i>	(Password for Database user ID.)
	<i>Admin logon</i>	(This is the user ID that is used to connect to the iSeries Connect configuration tool to gather the field information for the SQL statement that is provided.)
	<i>Admin password</i>	(Password for Admin user ID.)
	<i>Admin port</i>	Specify one of these values, based on your OS/400 version: <ul style="list-style-type: none"> • V5R1: 2001

		• V4R5: 2002
Click Next .		
New Application Connector - Input Fields	<i>InputParms</i>	InputParms (Accept default value.)
	<i>Parm1</i>	BUYER_ID Note: This input field comes from the SQL statement that is associated with the sample application connector. A more meaningful name, BUYER_ID, is assigned to it through this field.
Click Next .		
New Application Connector - Output Fields	<i>RowCount</i>	RowCount (Accept default value.)
	<i>Row</i>	Row
	<i>CUSTOMER_ID</i>	CUSTOMER_ID Note: This output field comes from the SQL statement that is associated with the FindCustomer application connector.
Click Next .		
Click Finish .		Note: The message "The file FindCustomer.AppConnector was successfully created" should be displayed.

Figure 2 shows the summary page for the new JDBC application connector, with the values that are provided in the table above.



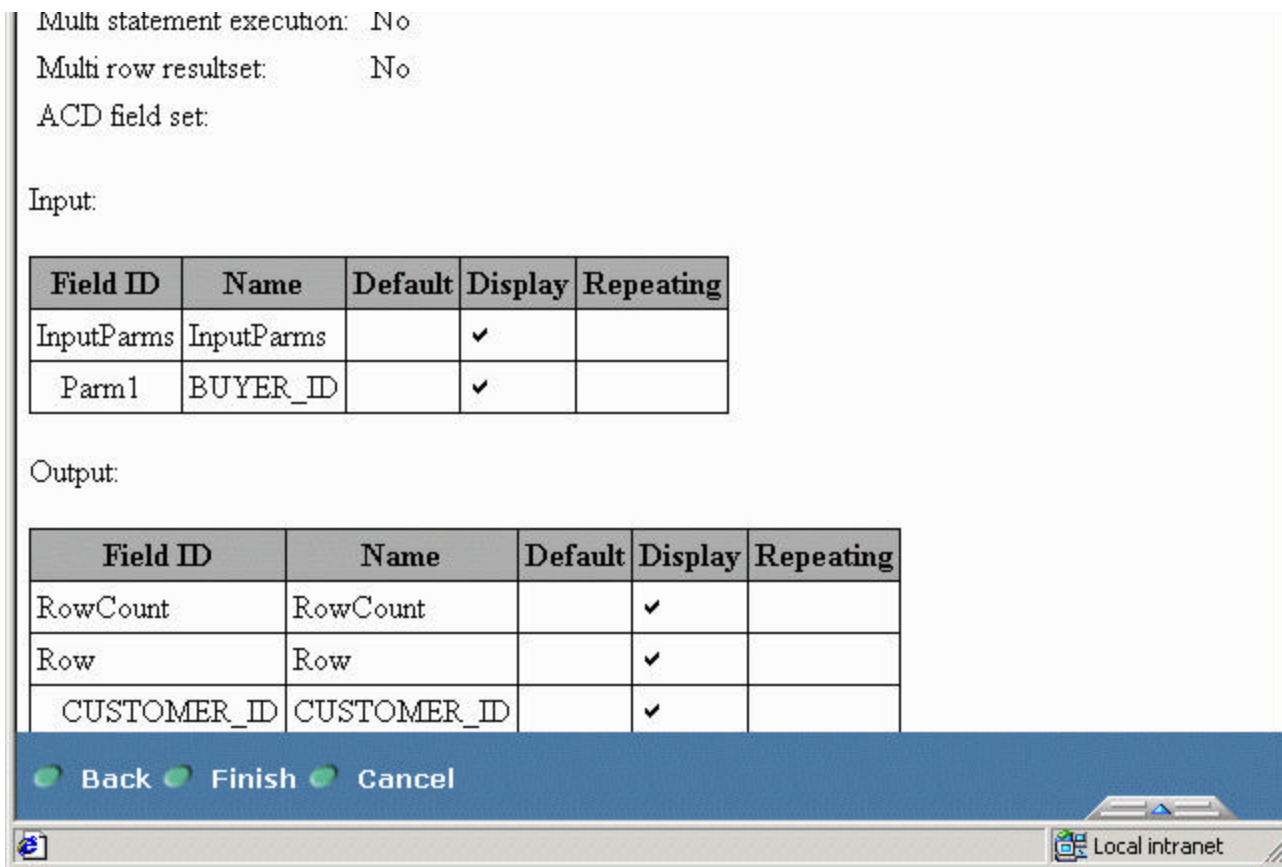


Figure 2. Sample FindCustomer JDBC application connector summary

[Top](#)

Configure the multiple-step process flow

These instructions guide you through creating a multiple-step process flow for the sample. You may instead use the JDBCTestFlow.ProcessFlow that included with the sample instead of creating your own. If you prefer to use the sample JDBC process flow, copy the sample file JDBCTestFlow.ProcessFlow from the /ConnSam11/JDBCSam1/BPWDData directory to the /QIBM/UserData/Connect110/SampInst/Connector directory and skip to [Deploy the process flow](#).

To configure the process flow for this sample, perform these steps in the iSeries Connect configuration tool (under the **Deployment** tab):

1. Click **Process Flows**
2. On the Process Flows page, select **Create a new process flow document** and then click **Next**.
3. To configure the new process flow, enter or select the following values:

Screen / action	Field	Value
New Process Flow	<i>Flow name</i>	JDBCTestFlow
	<i>Path</i>	(Leave blank.)
		Note: This is the directory where the process flow document will be saved.

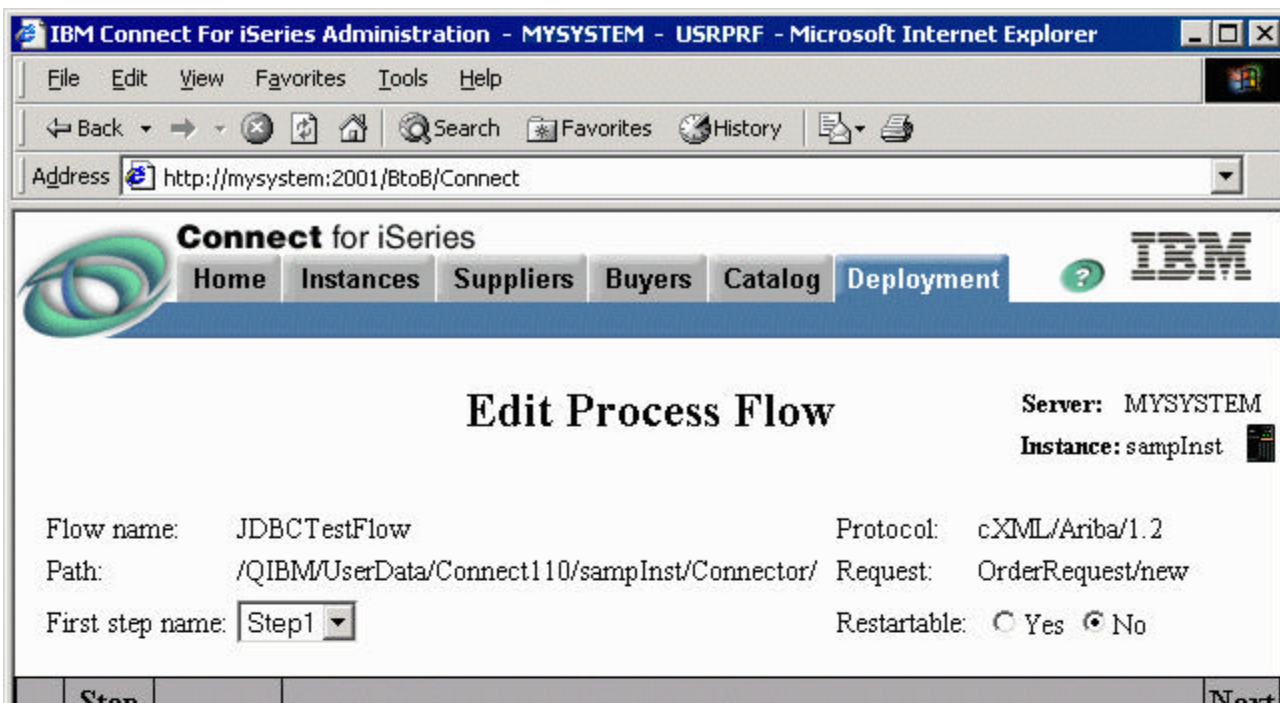
	<i>Protocol</i>	cXML/Ariba/1.2
	<i>Request</i>	OrderRequest/New
	<i>First step name</i>	Step1
	<i>Restartable</i>	No (Accept default.)
Click Next .		
Insert New Step	<i>Step name</i>	Step1
	<i>Step type</i>	Connector
	<i>Next step</i>	Step2
	<i>Error step</i>	(Leave blank.)
	<i>Connector</i>	FindCustomer.AppConnector
Click Add Mapping (Input)		
Edit Input Mapping	<i>Map Into</i>	/InputParms/BUYER_ID
	<i>Fields</i>	Request:/cXML/Header/From/Credential/Identity
Click Insert Field		
Click OK		
Click Add Mapping (Output)		
Edit Output Mapping	<i>Map From</i>	/Row/CUSTOMER_ID
	<i>Intermediate:*/</i>	tempcustomerid Note: The Intermediate field allows you to move data between steps. In this sample, the CUSTOMER_ID field that is retrieved from the CUSTOMER table by the SQL SELECT statement is stored in the intermediate field.
Click Add Field		
Click OK		
Click OK		
Click Insert		
Insert New Step	<i>Step name</i>	Step2
	<i>Step type</i>	Connector
	<i>Next step</i>	(Leave blank.)
	<i>Error step</i>	(Leave blank.)
	<i>Connector</i>	JavaCustID.AppConnector
Click Add Mapping (Input)		
Edit Input Mapping	(See table below for fields and mapping values.)	

Use this table to map the request fields into the back-end connector application input fields.

Fields	Map Into
MessageHeader:/com_ibm_connect_header_gatewayInstance	/b2bInstance
MessageHeader:/com_ibm_connect_header_marketplace	/marketplace
Request:/cXML/Header/From/Credential	/fromCredential
Request:/cXML/Header/From/Credential/@domain	/fromCredential/fromDomain
Request:/cXML/Header/From/Credential/Identity	/fromCredential/fromIdentity
Request:/cXML/Header/To/Credential	/toCredential
Request:/cXML/Header/To/Credential/@domain	/toCredential/toDomain
Request:/cXML/Header/To/Credential/Identity	/toCredential/toIdentity
Request:/cXML/Request/OrderRequest/OrderRequestHeader/@orderID	/orderID
Request:/cXML/Request/OrderRequest/OrderRequestHeader/@type	/requestType
Intermediate:*/tempcustomerid	/customerid
Request:/cXML/Request/OrderRequest/ItemOut	/item
Request:/cXML/Request/OrderRequest/ItemOut/@lineNumber	/item/lineNumber
Request:/cXML/Request/OrderRequest/ItemOut/ItemID/SupplierPartID	/item/identifier
Request:/cXML/Request/OrderRequest/ItemOut/@quantity	/item/quantity
Request:/cXML/Request/OrderRequest/ItemOut/ItemDetail/UnitPrice/Money	/item/price
Request:/cXML/Request/OrderRequest/ItemOut/ItemDetail/Description	/item/description

4. When you have completed mapping the fields, click **Save**.

Figure 3 shows the summary for the process flow.



Step Name	Type	Detail	Next Step
Step1	Connector	/QIBM/UserData/Connect110/sampInst/Connector/FindCustomer.AppConnector	Step2
	Input	Request:/cXML/Header/From/Credential/Identity -> /InputParms/BUYER_ID	
	Output	/Row/CUSTOMER_ID -> Intermediate:*/tempcustomerid	
Step2	Connector	/QIBM/UserData/Connect110/sampInst/Connector/JavaCustID.AppConnector	
	Input	MessageHeader:/com_ibm_connect_header_gatewayInstance -> /b2bInstance MessageHeader:/com_ibm_connect_header_marketplace -> /marketplace Request:/cXML/Header/From/Credential -> /fromCredential Request:/cXML/Header/From/Credential/@domain -> /fromCredential/fromDomain Request:/cXML/Header/From/Credential/Identity -> /fromCredential/fromIdentity Request:/cXML/Header/To/Credential -> /toCredential Request:/cXML/Header/To/Credential/@domain -> /toCredential/toDomain Request:/cXML/Header/To/Credential/Identity -> /toCredential/toIdentity Request:/cXML/Request/OrderRequest/OrderRequestHeader/@orderID -> /orderID Request:/cXML/Request/OrderRequest/OrderRequestHeader/@type -> /requestType Intermediate:*/tempcustomerid -> /customerId Request:/cXML/Request/OrderRequest/ItemOut -> /item Request:/cXML/Request/OrderRequest/ItemOut/@lineNumber -> /item/lineNumber Request:/cXML/Request/OrderRequest/ItemOut/ItemID/SupplierPartID -> /item/identifier Request:/cXML/Request/OrderRequest/ItemOut/@quantity -> /item/quantity Request:/cXML/Request/OrderRequest/ItemOut/ItemDetail/UnitPrice/Money -> /item/price Request:/cXML/Request/OrderRequest/ItemOut/ItemDetail/Description -> /item/description	
	Output		

Insert Remove Edit Copy Move Up Move Down Save Cancel

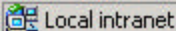
 Local intranet

Figure 3. Summary for JDBCTestFlow.ProcessFlow

[Top](#)

Deploy the process flow

To deploy the process flow, perform these steps in the iSeries Connect configuration tool (under the **Deployment** tab):

1. On the Deployment Flows page, click **Add Flow**.
2. To add a new deployment flow, enter or select these values:

Screen / action	Field	Value
Add Deployment Flow - Protocol	<i>Protocol</i>	cXML 1.2
Click Next .		
Add Deployment Flow - Process Flows	<i>Request</i>	OrderRequest
	<i>Request type</i>	New
	<i>Process Flow</i>	JDBCTestFlow
Click Next .		
Add Deployment Flow - Suppliers	<i>Select the suppliers for the specified flow.</i>	All suppliers
Click Next .		
Add Deployment Flow - Summary	<i>Deploy</i>	(Select.) Note: If there is another flow in the same instance that is deployed for the same protocol, marketplace, request, request type, and supplier and buyer combination, you must suspend the existing flow before you deploy the new one.
Click Finish .		
Connector Prompts	<i>FindCustomer JDBC userid</i>	(User ID used to connect to the database. The specified user profile must have enough authority to run the SQL statement that is provided in the application connector.)
	<i>FindCustomer JDBC password</i>	(Password for the specified user ID.)
Click OK .		

[Top](#)

Update the flow manager classpath for the sample application

You must update the flow manager classpath for the flow manager to locate the sample JavaCustId.class program in the directory /CONNSAM11/JDBCSam1/Application.

Perform these steps in the iSeries Connect configuration tool:

1. Click the **Instances** tab.
2. Select **samplnst** and click **Properties**.
3. Select the **Flow Manager** tab.
4. In the Classpaths list, add this path: /CONNSAM11/JDBCSam1/Application
5. Click **Add**.
6. Click **OK**.

[Top](#)

Run the sample

To run the JDBC connector sample, follow these steps in the iSeries Connect configuration tool:

1. Start the samplnst B2B instance.
2. Select **samplnst** and click **Test Drive Connect**.
3. Click **Next**.
4. On the Request Type page, select the second Order Request (the one for protocol cXML 1.2). Click **Next**.
5. On the Request Options page, accept the default values. Click **Next**.
6. On the Select Supplier page, select the supplier that is associated with cXML 1.2 (if you have more than one supplier). Click **Next**.
7. Enter the supplier password. Click **Next**.
8. Select any of the registered buyers who were added in the [Setup section](#) of this sample. (These buyers should have DUNS numbers of 321222888, 321222889, or 321222890.) Click **Next**.
9. On the Items to order page, select one or two items, and add price and quantity. Click **Next**.
10. On the Request to Test page, click **Next**.
11. On the Send Request page, click **Send Request**.

If the request is successful, the JavaCustID.log in /CONNSAM11/JDBCSam1/Results should look similar to this, showing the Buyer's credentials mapped to customer ID.

```
-----
In JavaCustID Java connector application:  running java connector code!
09.08.2001 at 22:27:50
-----
PropertyFileName = /CONNSAM11/JDBCSam1/Application/JavaCustID.properties
b2bInstance = samplnst
marketplace = Ariba Network
-----
orderID = D01234
requestType = new
-----
Buyer's credential
fromCredential/fromDomain = DUNS
fromCredential/fromIdentity = 321222889
customerID = Customer22
.....
Supplier's credential
toCredential/toDomain = DUNS
toCredential/toIdentity = 123222888
.....
Total number of items in the order: 2
-----
```

```
item/identifier = C1
item/lineNumber not found
item/quantity = 123
item/price = 10.0
item/description = Custom Product 1
.....
item/identifier = C2
item/lineNumber not found
item/quantity = 321
item/price = 20.0
item/description = Custom Product 2
.....
09.08.2001 at 22:27:50
The transaction took 37 milliseconds
```

[Top](#)

Appendix: JDBC driver properties

For details on IBM Toolbox for Java JDBC driver properties, see:

<http://www-1.ibm.com/servers/eserver/series/toolbox/faqjdbc.htm>

For details on IBM Developer Kit for Java JDBC driver properties, see:

<http://www.as400.ibm.com/developer/jdbc/Faqs/JDBCFAQ.html>

[Top](#)