Create a J2C application for an IMS phonebook transaction

Contents

Creating a J2C application for an IMS phone book transaction

This tutorial describes how to use the J2C wizard to build a simple Web application that processes an IMS™ transaction that returns a phonebook record.

Learning objectives

This tutorial enables you to:

- Use the J2C Java bean wizard to submit a transaction to IMS. Time required
- Create a Java method that accepts a customer number and returns the customer's information.

Time required

30 minutes

Related information

View thePDF version

IMS phonebook sample

Create a J2C application for an IMS phone book Phonebook transaction introduction

This tutorial illustrates how to use the J2C wizard to build a simple Web application that processes an IMS transaction that returns a phonebook record.

This tutorial might require some optionally installable components. To ensure that you have installed the appropriate optional components, see the System requirements list.

This tutorial is divided into several exercises that must be completed in sequence for the tutorial to work properly. This tutorial teaches you how to use the J2C Java™ Bean wizard to create a Java bean that runs a transaction in IMS. While completing the exercises, you will:

- Use the J2C Java bean wizard to submit a transaction to IMS.
- Create a Java method, **runPhonebook.java**, which accepts a customer number and returns the customer's information.

Time required

This tutorial should take approximately 30 minutes to finish. If you explore other concepts related to this tutorial, it could take longer to complete.

Skill level

Experienced

Audience

This tutorial is intended for users who are familiar with Enterprise Information systems (EIS) and IMS in particular.

System requirements

To complete this tutorial, you need to have the following tools and components installed:

- IBM® WebSphere® Application Server, version 6.1
- J2EE Connector (J2C) tools
- **Information about your IMS environment**: In this tutorial, your application interacts with an IMS application program in IMS. You need to obtain information such as the host name and port number of IMS Connect and the name of the IMS datastore where the transaction will run. Contact your IMS systems administrator for this information. Specifically, you need to perform some setup work in IMS if you want to run the IMS\PhoneBook IMS program.
- **A copy of the COBOL file ex01.cbl**: You may locate this file in your product installation directory: <installdir>\IBM\SDP70Shared\plugins\com.ibm.j2c.cheatsheet.content_7.0.0\samples\IMS\PhoneBook. If you wish to store it locally, you can copy the code from here: ../resources/ex01.pdf.
- **A clean workspace**.

To use this tutorial, you must have an application server installed and configured. To verify that a server runtime environment is available, click **Window → Preferences**, expand **Server**, and then click **Installed Runtimes**. You can use this pane to add, remove, or edit installed server runtime definitions. You can also download and install support for a new server.

Prerequisites

In order to complete this tutorial end to end, you should be familiar with:

- J2EE and Java programming
- Basic IMS Transaction Manager (IMS TM) concepts

Lesson 1.1: Select a resource adapter

This lesson leads you through the detailed steps to select and configure the resource adapter to connect to the IMS server.

Connecting to the IMS server

1. If the J2EE icon, , does not appear in the top right tab of the workspace, you need to switch to the J2EE perspective. From the menu bar, select **Window > Open Perspective > Other**. The Select Perspective window opens.
2. Select **J2EE**.
3. Click **OK**. The J2EE perspective opens.
4. In the J2EE perspective, select **File > New > Other**.
5. In the New page, select **J2C > J2C Java Bean**. Click **Next**
6. In the Resource Adapters Selection page, select the J2C 1.5 IMS resource adapter. For this tutorial select **IMS Connector for Java (IBM:9.1.0.2.3)**. Click **Next**.
7. In the Connection Properties page, select **Managed connection**.
8. If you already have a JNDI connection created, click **Browse** to locate your JNDI. If you do not have a JNDI connection, click **New**.
9. On the Server Selection page, select **WebSphere Application Server v6.1**.
10. Click **Next**.
11. On the New J2C Connection Factory page, accept the default Connection class name of `com.ibm.connector2.ims.ico.IMSManagedConnectionFactory`. In the blank fields, enter all the required connection information. Required fields, indicated by an asterisk (*), include the following:
    a. **For TCP/IP connection**:

        1) **Host name**: (Required) The IP address or host name of IMS Connect.
        2) **Port Number**: (Required) The number of the port used by the target IMS connect.
    b. **For local option connection**:
        1) **IMS Connect name**: (Required) The name of the target IMS connect.
    c. **For both**:
        1) **Data Store Name**: (Required) The name of the target IMS datastore.

12. Click **Finish**.
13. You may obtain the connection information from your IMS system administrator. When you have provided the required connection information, click **Next**.

EX01.cbl

Here is the code from EX01.cbl:

EX01.cbl

```
IDENTIFICATION DIVISION.
        ENVIRONMENT DIVISION.
        CONFIGURATION SECTION.
        DATA DIVISION.
       *
       *    IMS Connector for Java, COBOL Transaction Message Source
       *
       *******************************************************************/
       *                                                                 */
       * (c) Copyright IBM Corp. 2003                                   */
       * All Rights Reserved                                             */
       * Licensed Materials - Property of IBM                            */
       *                                                                 */
       * DISCLAIMER OF WARRANTIES.                                       */
       *                                                                 */
       * The following (enclosed) code is provided to you solely for the   */
       * purpose of assisting you in the development of your applications. */
       * The code is provided "AS IS." IBM MAKES NO WARRANTIES, EXPRESS OR */
       * IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF   */
       * MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, REGARDING   */
       * THE FUNCTION OR PERFORMANCE OF THIS CODE.                       */
       * IBM shall not be liable for any damages arising out of your use   */
       * of the generated code, even if they have been advised of the      */
       * possibility of such damages.                                    */
       *                                                                 */
       * DISTRIBUTION.                                                   */
       *                                                                 */
       * This generated code can be freely distributed, copied, altered,   */
       * and incorporated into other software, provided that:            */
       *   - It bears the above Copyright notice and DISCLAIMER intact     */
       *   - The software is not for resale                              */
       *                                                                 */
       *******************************************************************/
       *
        LINKAGE SECTION.

        01  INPUT-MSG.
            02  IN-LL          PICTURE S9(3) COMP.
            02  IN-ZZ          PICTURE S9(3) COMP.
            02  IN-TRCD        PICTURE X(10).
            02  IN-CMD         PICTURE X(8).
            02  IN-NAME1       PICTURE X(10).
            02  IN-NAME2       PICTURE X(10).
            02  IN-EXTN        PICTURE X(10).
            02  IN-ZIP         PICTURE X(7).

        01  OUTPUT-MSG.
        02  OUT-LL       PICTURE S9(3) COMP VALUE +0.
        02  OUT-ZZ       PICTURE S9(3) COMP VALUE +0.
        02  OUT-MSG      PICTURE X(40) VALUE SPACES.
        02  OUT-CMD      PICTURE X(8) VALUE SPACES.
        02  OUT-NAME1    PICTURE X(10) VALUE SPACES.
        02  OUT-NAME2    PICTURE X(10) VALUE SPACES.
        02  OUT-EXTN     PICTURE X(10) VALUE SPACES.
        02  OUT-ZIP      PICTURE X(7) VALUE SPACES.
        02  OUT-SEGNO    PICTURE X(4) VALUE SPACES.

    PROCEDURE DIVISION.
```

Lesson 1.2: Set up a Web project and Java interface and implementations

This lesson leads you through setting up a Web project and Java Interface and Implementations.

Before you begin, you must complete lesson 1.1. In this lesson you will

- Create a J2C Java Bean
- Create a dynamic Web project

1. All work done in the workbench must be associated with a project. Projects provide an organized view of the work files and directories, optimized with functions based on the type of project. In the workbench, all files must reside in a project, so before you create the J2C Java bean, you need to create a project to put it in. In the New J2C Java Bean page, type the value `IMSPhoneBook` in the **Project Name** field.
2. Click the **New** button beside the **Project Name** field to create the new project
3. Now you will create a dynamic Web project. In the New Source Project Creation page, select **Web project**, and click **Next**.
4. In the New Dynamic Web Project page, click **Show Advanced**.
5. Ensure that the following values are selected:
   a. **Project name**: IMSPhoneBook
   b. **Project contents**: accept default
   c. **Target runtime** : WebSphere Application Server v6.1
   d. **Configurations**: accept default
   e. **Add project to an EAR**: checked
   f. **EAR Project name**: IMSPhoneBookEAR
6. Click **Finish**.
7. A dialog box may appear asking if you would like to switch to the Dynamic Web perspective. Click **Yes**.
8. On the J2C Java Bean Output Properties page:
   a. In the **Package Name** field, click **Browse** and select the `IMSPhoneBook` project. Click **OK**.
   b. Type `sample.ims` in the **Package Name** field.
   c. Type `PhoneBook` in the **Interface Name** field.
   d. Type `PhoneBookImpl` in the **Implementation Name** field.
9. Click **Finish**.

Lesson 1.3: Create a Java method

Lesson 1.3 leads you through the creation of a Java method.

Before you begin, you must complete Lesson 1.2: Setting up the Web project and Java Interface and Implementations. In this lesson you will:

- Create a Java method
- Create the input and output data mapping between COBOL and Java

1. **First you will create a Java method:** In the Snippets view, select **J2C**. Double click **Add Java method to J2C Java bean**.
2. In the Java Methods page, click **Add**
3. In the Java method name field, type runPhoneBook for the name of the method.
4. Click **Next**.
5. **Next you will create the input data mapping between COBOL and Java:** In this step, you will import the Ex01.cbl (COBOL) file that is needed to create your applicaion. The Ex01.cbl file is located in <installdir>\IBM\SDP70Shared\plugins\com.ibm.j2c.cheatsheet.content_6.0.0\Samples\IMS\phonebook, where <installdir> is the directory where this product is installed. The COBOL file contains the application program that runs on the IMS server. It has the definition of the structure to be passed to the IMS server via the communications area. This structure represents the customer records being returned from the IMS application program. Before you can work with a file, you must import it from the file system into the workbench. In the **Input type** field of the Java Method page, click **New**.
6. In the Data Import page, ensure that the **Choose mapping** field is **COBOL_TO_JAVA**. Click **Browse** beside the COBOL file
7. Locate the Ex01.cbl file in the file system, and click **Open**.
8. Click **Next**.
9. In the COBOL Importer page, click **Show Advanced**.
   a. Select the following options:

Table 1. COBOL Importer Parameter Settings

Parameter	Value
Platform Name	Z/OS
Codepage	IBM-037
Floating point format name	IBM Hexadecimal
External decimal sign	EBCDIC
Endian name	Big
Remote integer endian name	Big
Quote name	DOUBLE
Trunc name	STD
Nsymbol name	DBCS

   b. Click **Query** to load the data.
   c. A list of data structures from the Ex01.cbl file is shown. Select **INPUT-MSG** in the **Data structures** field.
   d. Click **Next**.
10. In the Saving properties page, select the following values for input type:
    a. Select **Default** for **Generation Style**.
    b. Click **Browse** beside the **Project Name** and choose the Web project **IMSPhoneBook** .
    c. In the **Package Name** field, type `sample.ims.data`.
    d. In the **Class Name** field, accept the default **INPUTMSG**. Click **Finish**.
11. **Next you will create the output data mapping between COBOL and Java:** In the Java method page, click **New** next to the **Output type** field.

12. In the Data Import page, ensure that the **Choose mapping** field is **COBOL_TO_JAVA**.
13. Locate the `Ex01.cbl` file in the file system, and click **Open**.
14. Click **Next**.
15. In the COBOL Importer page, click **Show Advanced**.
    a. Select the following options:

Table 2. COBOL Importer Parameter Settings

Parameter	Value
Platform Name	Z/OS
Codepage	IBM-037
Floating point format name	IBM Hexadecimal
External decimal sign	EBCDIC
Endian name	Big
Remote integer endian name	Big
Quote name	DOUBLE
Trunc name	STD
Nsymbol name	DBCS

    b. Click **Query** to load the data.
    c. A list of data structures from the Ex01.cbl file is shown. Select **OUTPUT-MSG** in the **Data structures** field.
    d. Click **Next**.
16. In the Saving Properties page,
    a. Select **Default** for **Generation Style**.
    b. Click **Browse** beside the **Project Name** and choose the Web project **IMSPhoneBook** .
    c. In the **Package Name** field, type `sample.ims.data`.
    d. In the **Class Name** field, accept the default **OUTPUTMSG** . Click**Finish**.
    e. Leave the **Save session as Ant script** unchecked.
17. On the Java Method page , click **Finish**.
18. In the Binding Details page, ensure that the **interactionVerb** is **SYNC_SEND_RECEIVE(1)** to indicate that the interaction with IMS involves a send followed by a receive interaction.
19. Click **Finish**.

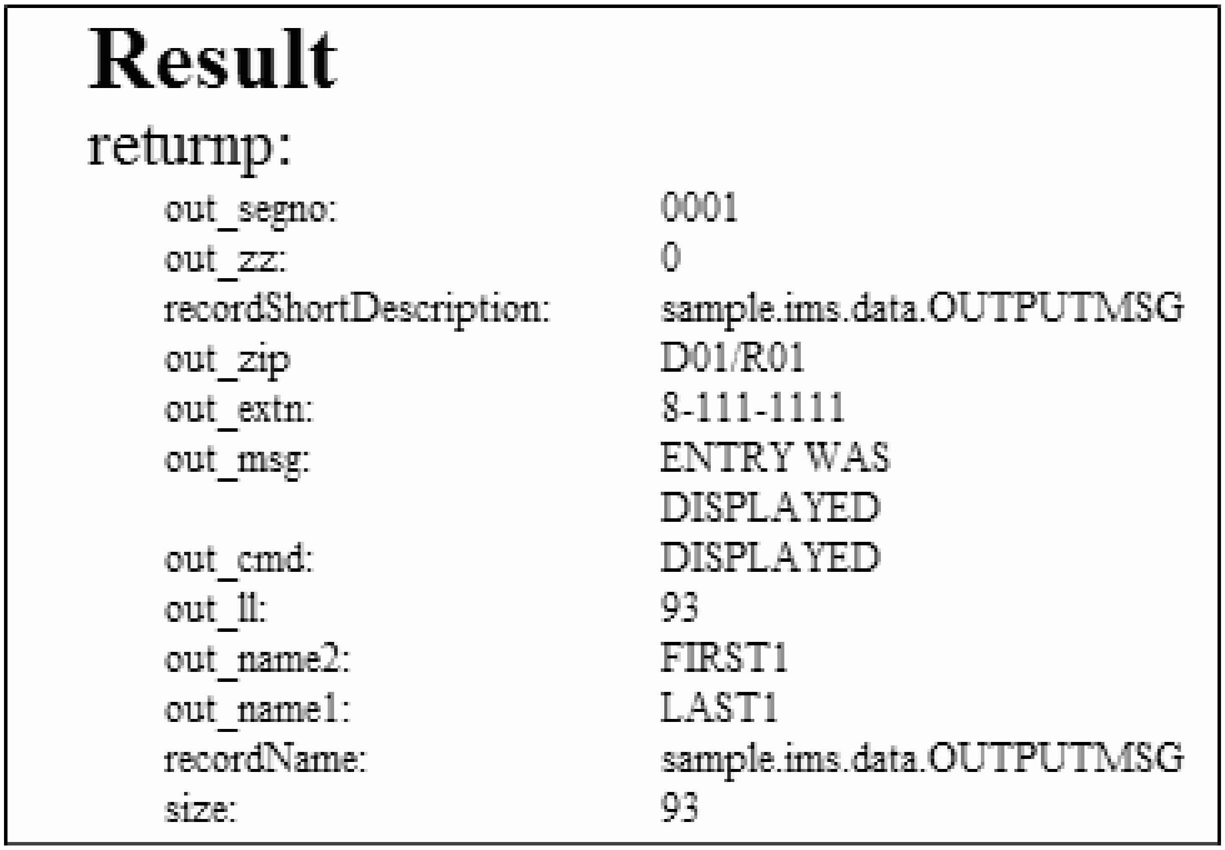
Lesson 1.4: Deploy your application

Lesson 1.4 leads you through the creation of a JSP and a Faces JSP to test your application.

Before you begin, you must complete Lesson 1.3. In this lesson you will:
- Create a JSP to test your Java application.
- Run the JSP in the WebSphere test environment.
- Create a Faces JSP to test your Java application.
- Run the Faces JSP in the WebSphere test environment.

1. **First you will create a JSP:** Click **File > New > Other > J2C**.
2. In the Select a wizard page, select **Web Page, Web Service, or EJB from J2C Java bean**.
3. Click **Next**.
4. In the J2C Java bean selection page, click **Browse**.
5. In the Find J2C bean page, type an asterisk (*) in the **Select entries** field.

6. In the **Matching types** field, select **PhoneBookImpl**.
7. Click **OK**.
8. In the J2C Java bean selection page, click **Next**.
9. In the Deployment Information page, select **Simple JSP**.
10. Click **Next**.
11. In the JSP Creation page, select . **Generate simple JSPs with default input modes**.
12. In the JSP folder field, enter a JSP Folder name, such as `SampleJSP`.
13. Click **Finish**.
14. **Next you will run your JSP:** Right-click **TestClient.jsp** and select **Run on Server**.
15. A browser window with the Test Client will launch. Click on **runPhoneBook** method.
16. Use the following values as Inputs:
    - Type `IVTNO` in the **In__trcd** field.
    - Type `0` in the **In___zz** field.
    - Type `LAST1` in the **In__name1** field.
    - Type `DISPLAY` in the **In__cmd** field.
    - Type `59` in the **In__ll** field.
    - Type `93` in the **size** field.
17. Click **Invoke**, and this output will appear in the **Result** field.

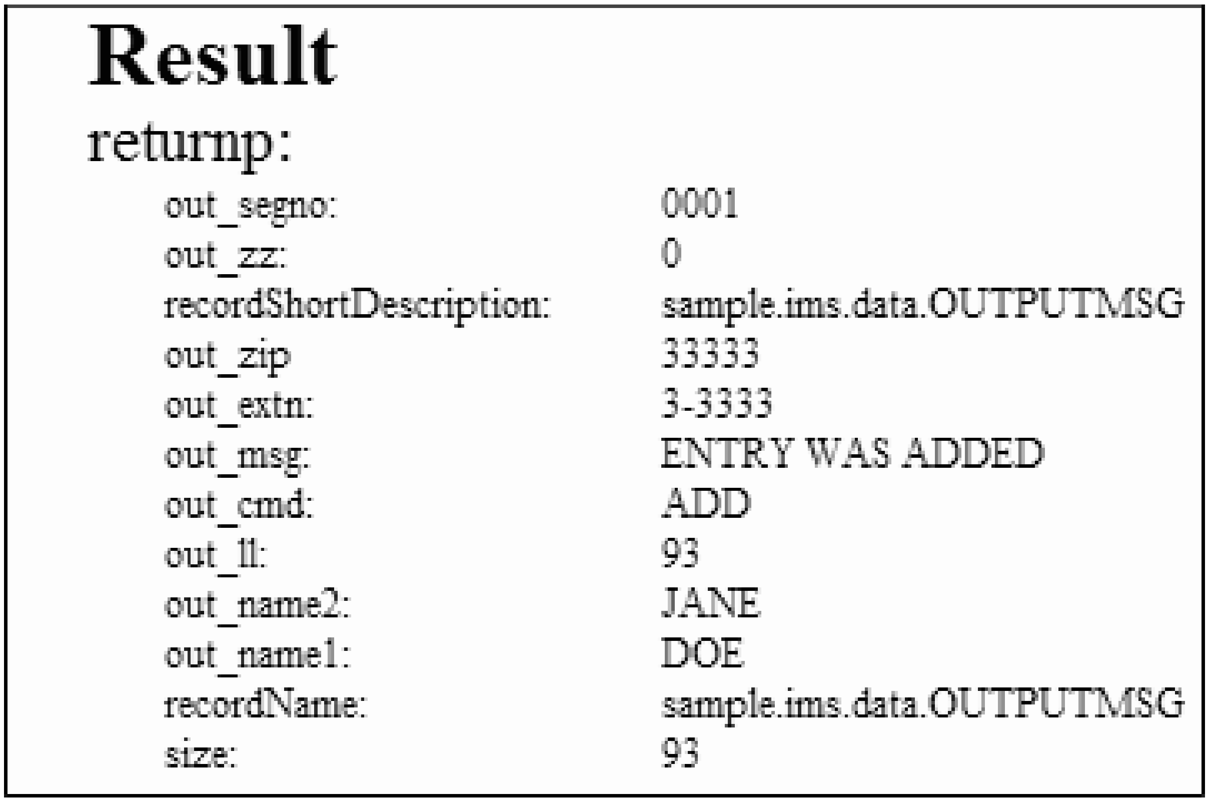


```
Result
returnp:
    out_segno:                  0001
    out_zz:                     0
    recordShortDescription:     sample.ims.data.OUTPUTMSG
    out_zip                     D01/R01
    out_extn:                   8-111-1111
    out_msg:                    ENTRY WAS
                                DISPLAYED
    out_cmd:                    DISPLAYED
    out_ll:                     93
    out_name2:                  FIRST1
    out_name1:                  LAST1
    recordName:                 sample.ims.data.OUTPUTMSG
    size:                       93
```

18. Now submit another command to add a phone book entry. Click on **runPhoneBook**method.
19. Use the following values as Inputs:
    - Type `59` in the **In__ll** field.
    - Type `0` in the **In___zz** field.
    - Type `IVTNO` in the **In__trcd** field.

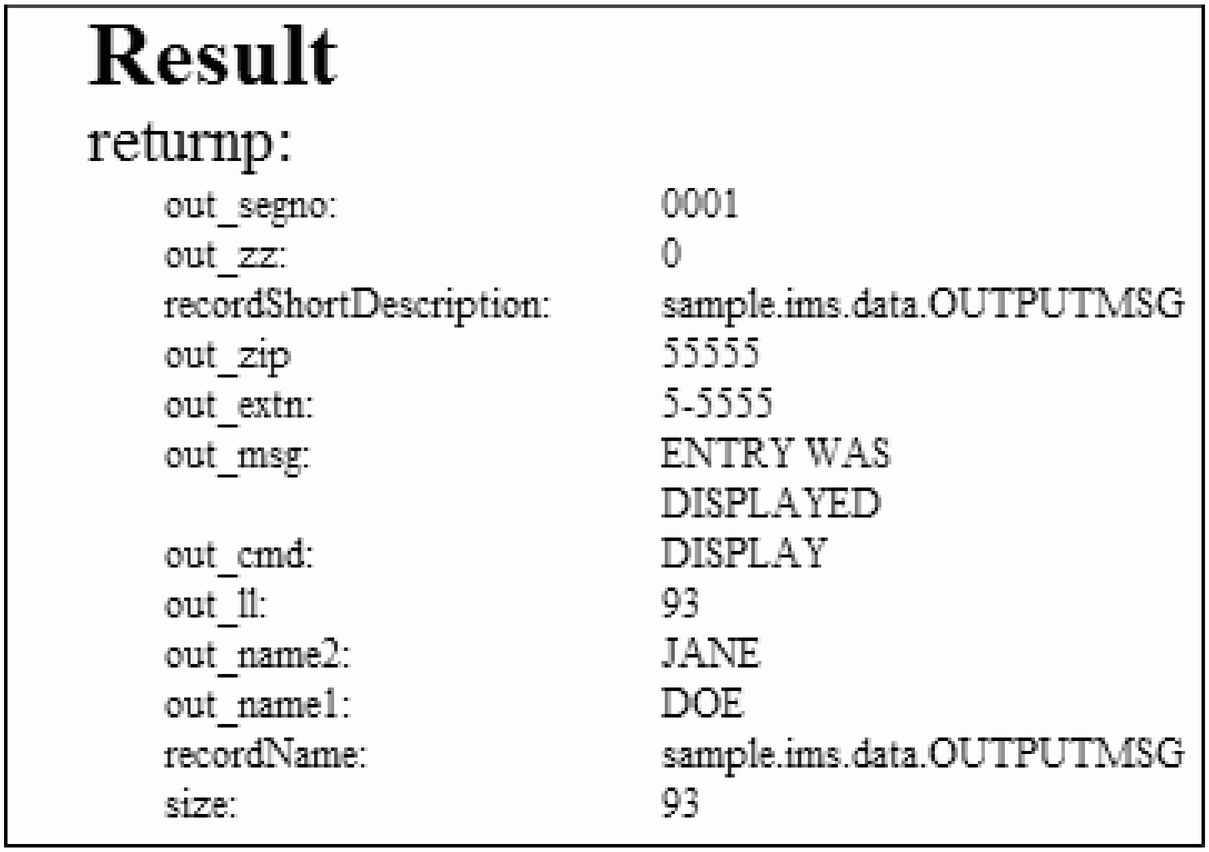
- Type Add in the **In__cmd** field.
- Type Jane in the **In__name2** field.
- Type Doe in the **In__name1** field.
- Type 55555 in the **In__zip** field.
- Type 5-5555 in the **In__extn** field.
- Type 93 in the **size** field.

20. Click **Invoke**, and this output will appear in the **Result** field.



```
Result
returnp:
    out_segno:                  0001
    out_zz:                     0
    recordShortDescription:     sample.ims.data.OUTPUTMSG
    out_zip                     33333
    out_extn:                   3-3333
    out_msg:                    ENTRY WAS ADDED
    out_cmd:                    ADD
    out_ll:                     93
    out_name2:                  JANE
    out_name1:                  DOE
    recordName:                 sample.ims.data.OUTPUTMSG
    size:                       93
```

21. Now submit another command to display the phone book entry you just added. Click on **runPhoneBook** method.
22. Use the following values as Inputs:
    - Type 59 in the **In__ll** field.
    - Type IVTNO in the **In__trd** field.
    - Type 0 in the **In___zz** field.
    - Type DISPLAY in the **In__cmd** field.
    - Type Doe in the **In__name1** field.
    - Type 93 in the **size** field.
23. Click **Invoke**, and this output will appear in the **Result** field.



```
Result
returnp:
    out_segno:                  0001
    out_zz:                     0
    recordShortDescription:     sample.ims.data.OUTPUTMSG
    out_zip                     55555
    out_extn:                   5-5555
    out_msg:                    ENTRY WAS
                                DISPLAYED
    out_cmd:                    DISPLAY
    out_ll:                     93
    out_name2:                  JANE
    out_name1:                  DOE
    recordName:                 sample.ims.data.OUTPUTMSG
    size:                       93
```

24. **Now you will create a Faces JSP to deploy the J2C Java bean:**
25. Expand the IMSPhoneBook project, and find the WebContent folder.
26. Right click on **WebContent** folder in your **IMSPhoneBook** project and select **New > Other > Web > Faces JSP file**.
27. Type `Test` in the **Name** field.
28. Accept defaults for all other fields.
29. Click **Finish**.
30. **Next you will add the Java bean to faces JSP:** Once you have created the Faces JSP file, the page should open Test.jsp in the Design page. If the workspace does not open in the Design page of the editor, expand the **WEB-INF** folder under the **WebContent** folder. Right click on **Test.jsp**, select **Open With**, and click on **Page Designer**. Test.jsp will open in the Design page.
31. The Palette view should appear on the right panel. If it does not appear, in the top menu, click on **Window > Show view > Palette**.
32. In the **Data**folder of the Palette view, click on the **JavaBean** option of the Palette.
33. Drag and drop the JavaBean to the Test.jsp editor; the Add JavaBean wizard will open.
34. Select **Add new JavaBean**.
35. In the **Name** field, type `phonebookLookup`.
36. Click the open book icon, 📖, beside the **Class** field. The Class Selection window appears.
37. In the Class Selection page, type `PhoneBookImpl` in the **Search** field.
38. Deselect the **Add input/output controls to display the JavaBean on the Web page** check box.
39. Click **Finish**.
40. You will see **PhoneBookImpl** in the Page Data view.
41. **Adding input and output controls to the faces JSP:** Right-click **phonebookLookup Java Bean** in the Page Data view, and click **Add New JavaBean Method**.

42. From the list of available methods, click on **runPhoneBook**.
43. Click **OK**.
44. Expand **phonebookLookup Java Bean** in the Page Data view, and select the **runPhoneBook()** method.
45. Drag and drop the **runPhoneBook()** method onto the editor. The Insert JavaBean wizard appears.
46. In the **Create controls for:** field, select **Inputting data**.
47. In the **Fields to display** field, select **None**, to clear the form.
48. In the **Fields to display** field, select these input fields
    - **arg.in__trcd**
    - **arg.in__zz**
    - **arg.size**
    - **arg.in__name1**
    - **arg.in__cmd**
    - **arg.in__ll**
49. Click **Finish**.
50. Accept defaults for the other fields.
51. Click **Next**.
52. In the Configure Data Controls page, select **Create controls for displaying the results**.
53. In the **Fields to display** field, select **None**, to clear the form.
54. In the **Fields to display** field, select these output fields
    - **out__zz**
    - **out__zip**
    - **out__extn**
    - **out__msg**
    - **out__cmd**
    - **out__ll**
    - **out__name2**
    - **out__name1**
    - **size**
55. Click **Finish**.
56. Save your Faces JSP page, by pressing `Ctrl-S` or by clicking **File > Save** in the toolbar.
57. **Now you will test the Faces JSP:** Select the Servers tab. Start the test server, if it is not already running. To start the server, right click on the **WebSphere Application Server v6.0** and click **Start**.
58. Right click on Test.jsp (the faces JSP that you just created) in the Project Explorer view, and select **Run < Run on Server**.
59. Select **WebSphere Application Server v6.0** and click **Finish**.
60. The browser will open to Test.jsp.Use the following values as Inputs:
    - Type `IVTNO` in the **In__trcd** field.
    - Type `59` in the **In__ll** field.
    - Type `0` in the **In___zz** field.
    - Type `DISPLAY` in the **In__cmd** field.
    - Type `LAST1` in the **In__name1** field.
    - Type `93` in the **size** field.

    Note: Ensure that there are no extra spaces in the text box before your entry.
61. Click **Submit**.

62. You will see the output displayed in your browser.

Congratulations! You have completed the PhoneBook tutorial.

Create a J2C application for an IMS phone book transaction summary

This tutorial has taught you how to use the J2C Java bean wizard to connect to an IMS server, and return an phonebook entry.

Lessons learned

If you have completed all of the exercises, you should now be able to

- Use the J2C Java bean wizard to create a J2C application that interfaces with an IMS transaction.
- Create a Java method, **runPhoneBook.java** , which accepts an individual's name and returns the individual's contact information.
- Create a JSP to deploy the application on WebSphere Application Server.
- Create a Faces JSP to deploy the application on WebSphere Application Server.