
H03

**Creating and Deploying an IMS Conversational
Transaction as an EJB or SOAP Service Flow**

**Shyh-Mei F. Ho
Benjamin Sheats**



St. Louis, MO

September 30 - October 3, 2002

Note:

References in this document to IBM products (including programs or services), do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product in this document is not intended to state or imply that only IBM's product may be used. Any functionally equivalent product may be used instead, although the services described as part of this offering may vary based on non-IBM Vendor agreements. Evaluation is the responsibility of the customer.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing to the IBM Director of Commercial Relations, IBM Corporation, Purchase, NY 10577.

© Copyright: International Business Machines, Inc. 2002

Workshop Overview

Objectives

This sample will show you how to:

- create two COBOL web services in WebSphere Studio Application Developer Integration Edition 4.1 (WSAD-IE 4.1)
- build an IMS conversational Service Flow composed of these two services,
- create and test a J2EE application in the WebSphere^(R) 4.1 test environment, and/or
- create and test a Web Service and deploy it using SOAP to the WebSphere test environment.

Time required

Allow the following amount of time to complete the Flow Scenario:

<u>Part I: Creating COBOL Web Services</u>	25 minutes
<u>Part II: Creating a Flow Service</u>	25 minutes
<u>Part III: Deploy the Flow Service</u>	30 minutes
<u>Part IV: Test a Flow Service using the EJB Test Client Tool</u>	20 minutes
<u>Part V: Test a Flow Service using a SOAP Proxy</u>	15 minutes
<u>Part VI: Deploy a Flow service to a production server</u>	25 minutes
Complete Scenario	140 minutes

Working with Perspectives

Most of the tasks in this sample assume that you are using the Enterprise Services perspective. To open one, click **Perspective > Open > Other > Enterprise Services** from the main menu. If you want to stay in this perspective, proceed as follows:

- Click **Window > Preferences** and click **Do not switch** under the **New Project Options** heading.
- Click **Server** in the right pane and clear **Switch to the server perspective when a server starts**.
- Click **Debug** and clear **Show Debug perspective when a program is launched in run mode**.

Creating and Deploying an IMS Conversational Service Flow

Follow these steps to create and deploy a Service flow for an IMS conversational transaction:

Part I: Creating COBOL Web Services

- [Step 1: Create a service project](#)
- [Step 2: Import the COBOL copybook](#)
- [Step 3: Create service definitions](#)

Part II: Creating a Flow Service

- [Step 4: Create a Flow service project](#)
- [Step 5: Create an abstract service definition](#)
- [Step 6: Adding services to the classpath](#)
- [Step 7: Create a Service Flow](#)
- [Step 8: Build your flow](#)

Part III: Deploy a flow service

- [Step 9: Configure the WebSphere Unit Test Environment](#)
- [Step 10: Create the Session Bean](#)
- [Step 11: Add the application to the server configuration](#)

Part IV: Test a Flow Service using the EJB Test Client Tool

- [Step 12: Testing the EJB Session bean](#)

Part V: Test a Flow Service using a SOAP Proxy

- [Step 13: Test the Web service](#)

Part VI: Deploy a Flow service to a production server

- [Step 14: Packaging a Flow service for the production server](#)
- [Step 15: Deploying a Flow service to the production server](#)
- [Step 16: Testing a Flow service on the production server using a SOAP proxy](#)

Part I

Creating COBOL Web services

Step 1: Create a service project

Proceed as follows to create a service project.

1. Launch the WSAD workbench.
2. Switch to the **Enterprise Services** perspective (to do this, click **Perspective > Open > Other > Enterprise Services**).
3. From the main menu, click **File > New > Service Project**.
4. In the Service Project window, enter the project name `Loan_CBL` and click **Finish**.

Note: If you close the **Service Provider Browser** for `Loan_CBL`, you can launch it again by right-clicking the `Loan_CBL` folder and selecting **Service Provider Browser** from the drop-down menu.

Step 2: Import the COBOL copybook

To import the COBOL copybook:

1. In the Navigator window, click the **Packages** tab
2. Right-click `Loan_CBL`, and select **New > Package** from the drop-down menu.
3. In the **Package** field, type `loan.cbl` and click **Finish**.
4. From the main menu, click **File > Import**.
5. In the Import window, select **File System** from the topic tree and click **Next**.
6. In the Import File System window, browse to `C:\$Samples\loan.cbl` <where the COBOL copybook file is located>
7. Click **Finish**.

Here is the `loan.cbl` COBOL copybook:

```
IDENTIFICATION DIVISION.  
    ENVIRONMENT DIVISION.  
    CONFIGURATION SECTION.  
    DATA DIVISION.  
    LINKAGE SECTION.  
  
    01  IN-CUSTOMER-INFO.  
        02  LL                                PICTURE S9(4) COMP.  
  
        02  ZZ                                PICTURE S9(4) COMP.  
  
        02  TRCD                             PICTURE X(8) .  
  
        02  LASTNAME                          PICTURE X(10) .  
        02  FIRSTNAME                         PICTURE X(10) .  
        02  SSN                               PICTURE X(9) .  
  
    01  OUT-PERCENT-QUALIFY.  
        02  LL                                PICTURE S9(4) COMP.  
  
        02  ZZ                                PICTURE S9(4) COMP.  
  
        02  PERCENT                           PICTURE X(3) .
```

```

01  IN-FINANCIAL-NEED.
    02  LL                                PICTURE S9(4) COMP.

    02  ZZ                                PICTURE S9(4) COMP.
    02  YEARS                            PICTURE X(3) .
    02  AMOUNT                           PICTURE X(10) .

01  OUT-LOAN-RESULT.
    02  LL                                PICTURE S9(4) COMP.

    02  ZZ                                PICTURE S9(4) COMP.
    02  RESULT                           PICTURE X(20) .

```

PROCEDURE DIVISION.

Step 3: Create service definitions

In this step, you will create WSDL service definitions for the COBOL copybook that you imported in Step 2.

To create the IMS service definition files:

1. On the Service Providers page, click **IMS Services**. If this page is not already open, select the `Loan_CBL` service project and click the **Service provider browser** Toolbar icon.
2. In the EIS Service page, type the property values appropriate for your environment. See [Providing IMS resource adapter connection properties](#) for a description of these properties. For example:
 - o In the **DataStoreName** field type `SOCKEYE`
 - o In the **HostName** field type `ECDVL91.STL.IBM.COM`
 - o In the **PortNumber** field type `9999`
 - o Select the field named **required** in order that the last entry is saved.
 - o Click **Add to Project**.
3. In the Add EIS Service page:
 - a. Type `http://cbl.loan` for the target namespace which is a unique logical namespace for your service.
 - b. Type `Loan` for the name of the port type since this field represents the interface to getting the output from the IMS transaction using the service. In this case, it would be getting the results of the request to run the Loan transaction, submitted by the application at the client. Click **Finish** to accept all other default names. An editor opens with the Binding page. This view is the Design view. To see the source view, click on the **Source** tab on the Binding page. Now you can add the operations to the service.
4. In the Design view, under **Operations**, click **New**.
5. In the IMS Operation Binding Properties page, the properties of the interaction with the EIS are shown. Type the property values appropriate for your application. See [Providing IMS resource adapter operation binding properties](#) for a description of these properties. For example:
 - o Ensure that the **imsRequestType** field is set to `1` to indicate that the interaction with IMS consists of running a transaction.
 - o Ensure that the **interactionVerb** field is set to `1` to indicate that the interaction with IMS involves a send followed by a receive interaction.
 - o Select the field named **required** in order that the last entry is saved.
 - o Click **Next**.
6. In the Operation Binding page you create new input and output messages based on the existing input and output operations. Leave **Create a new operation** selected. In the **Operation name** field type `applyLoan`. Leave the type of operation as `REQUEST_RESPONSE` because there will be two messages, one for the request to run the IMS transaction and one for the response from the IMS transaction. Click **Next**.

-
- In the Operation Binding page click **Import** next to the input message. The Type Importer wizard opens. Import the Loan.cbl file to specify the XML schema definition for the input part.
 - Click **Browse** in the Type Importer wizard.

- In the File selection page browse to the Loan_CBL project then Java package loan.cbl and select Loan.cbl. Click **OK**. Click **Next**.
- In the Cobol Import Properties page, enter the following values:

Codepage	037
Floating Point Format	IBM 390 Hexadecimal
Endian	Big
Remote Integer Endian	Big
External Decimal Sign	EBCDIC
QUOTE	DOUBLE
TRUNC	BIN
NSYMBOL	DBCS

- Some of the values in the table above might differ for your environment. For example, you might need to specify a different value for the **Codepage** field, if your IMS data is in a codepage other than U.S. English (037). In addition, the value for the **QUOTE** field might differ, depending on your COBOL source. See [COBOL properties](#) for a complete description of the fields on this page. Click **Next**.
 - In the Cobol Importer page, the Level 01 I/O Areas from the file are displayed. Select IN-CUSTOMER-INFO which will populate the XSD type name with IN-CUSTOMER-INFO. You can accept the default to overwrite the XSD=types. Click **Finish**.
 - The invocation of the applyLoan method would result in transaction input information being passed from the application to the input message and then to the EIS. Then the transaction output would be returned from the EIS to the output message and then to the application.
- In the Operation binding page, click **Import** next to the output message. The Type Importer wizard opens. You import the Loan.cbl file to specify the XML schema definition for the output part.
 - Click **Browse** in the Type Importer wizard.
 - In the File selection page, click **Browse** and then click **OK** to select the same Loan.cbl from the loan.cbl package. In this sample, both the input and output message definitions are contained in the same COBOL source file, Loan.cbl. Click **Next**.
 - In the Cobol Import Properties page, specify the same values that you entered above for input. Click **Next**.
 - In the Cobol Import window, select OUT-PERCENT-QUALIFY in the Level 01 IOAreas list which will populate the XSD type name with OUT-PERCENT-QUALIFY. You can accept the default to overwrite the XSD types. Click **Finish** to return to the Operation binding page.
 - Click **Finish**. Save the results <Ctrl> S.
 - Repeat the above steps starting from step 4 to create a new operation to run the second iteration. Name the operation getLoan. Select IN-FINANCIAL-NEED and OUT-LOAN-RESULT as the input and output **Level 01 I/O Areas, respectively**.
 - Close the editor by clicking the **X** next to Loan.wsd1 then click **Yes** to save the changes, or <Ctrl> S. Note that you **must save the changes** in order to successfully proceed with the sample. Also close the Service Provider page in the editor. The service interface file Loan.wsd1 and the service binding file LoanIMS.wsd1 are updated in the package of the Loan_CBL project.

The binding file defines the <service> part of a WSDL description and then imports the service interface file. The interface file contains the <message>, <portType>, and <binding> parts of a

WSDL description, that describe the interface to the Web service. Messages describe input and output requests and responses to the service while port types are the collection of operations a service supports. Bindings are the implementation of the service such as an IMS binding to manage communication requests/responses between the service and IMS.

6. Right click on Loan_CBL and select **Generate Helper Classes**. Note that the following Java classes will be created:
 - Input and output beans:
 - INCUSTOMERINFO.java
 - INFINANCIALNEED.java
 - OUTPERCENTQUALIFY.java
 - OUTLOANRESULT.java
 - Input and output format handlers
 - INCUSTOMERINFOFormatHandler.java
 - INFINANCIALNEEDFormatHandler.java
 - OUTPERCENTQUALIFYFormatHandler.java
 - OUTLOANRESULTFormatHandler.java

Part II

Creating a Flow Service

Step 4: Create a Flow service project

Proceed as follows to create another service project for your flow.

1. From the main menu, click **File > New > Service Project**.
2. In the Service Project window, enter the project name `Loan_FLOW` and click **Finish**.

Step 5: Create an Abstract Service Definition

To create a WSDL file to contain your flow, follow the lettered task list below.

A. Name the Service Definition

1. Highlight the `Loan_FLOW` folder in the list of Packages.
2. From the main menu, click **File > New > Service Definition**.
3. Enter `LoanFlowDef` as the WSDL definition name, and `http://loan.flow/` as the target namespace and click **Next**
4. Enter or select "`Loan_FLOW`", with file name: `LoanFlowDef.wsdl`, and click **Finish**

B. Create a New Port Type

1. Highlight the `LoanFlowDef.wsdl`, click the **Design** tab. Then click **New** under the **Port Types** heading.
2. Enter `LoanFlowPortType` and click **Finish**.

C. Create the runLoan Operation

1. Click the newly created `LoanFlowPortType` link.
2. In the Port Type General Information window, click **New** under the Operations heading.

-
3. In the New Operation window, type `runLoan` in the **Operation Name** field, and click **Finish**.

D. Modify the `runLoanRequest` Message

1. Return to the **Overview** (click the **home** icon in the upper right corner of the editor)
2. Click **runLoanRequest** under the Messages heading.
3. Click **New** under the Parts heading, and type `lastname` into the New Part window and click **Finish**.
4. Click the newly created **lastname** link.
Below the **Type** heading, select **xsd:string** from the Built-in Type drop-down menu.
5. Return to the Message General Information view (click the **Back** arrow icon)
6. Repeat for the parts: `firstname`, `ssn`, `years`, and `amount`.

E. Modify the `runLoanResponse` Message

1. Return to the **Overview** and click **runLoanResponse** under the **Messages** heading.
2. Click **New** under the Parts heading, and type `result` into the New Part window, and click **Finish**.
3. Click the newly created **result** link.
Below the **Type** heading, select **xsd:string** from the Built-in Type drop-down menu.
4. Exit and save the WSDL file.
5. Close all active **Editor** views.

Step 6: Adding services to the Classpath

1. Right-mouse click **Loan_FLOW** in the list of Java Packages, and select **Properties** from the drop down list
2. In the topic tree, highlight **Java Build Path**, and select the **Projects** tab
3. Click the check-box beside **Loan_CBL**, and click **OK**.

Step 7: Create a Service Flow

1. Expand the `Loan_FLOW` folder, click **LoanFlowDef.wsdl**.
2. From the main menu, click **File > New > Flow** to launch a [Service Flow editor](#).
3. Accept the defaults in the **Flow** window, and click **Finish**.

Step 8: Build your Flow

In the last step, you launched a [Service Flow editor](#) in which an *Input* and *Output* node were automatically generated. This step will show you how to build this into a flow that sustains conversational transactions.

To build this flow, complete the lettered tasks that are listed below in sequence.

A. Add the Activity nodes

To add the activity nodes, proceed as follows.

1. From **Loan_CBL/cbl.loan**, click **LoanIMS.wsdl** and drag it to the editor window. A Service selection window will pop-up.
2. In the Service selection window, chose **applyLoan** from the **Operation** drop-down list and click **OK**.
3. Drag **LoanIMS.wsdl** to the editor again.
4. Select **getLoan** from the **Operation** drop-down list and click **OK**.

When you're done, your flow should look like this:

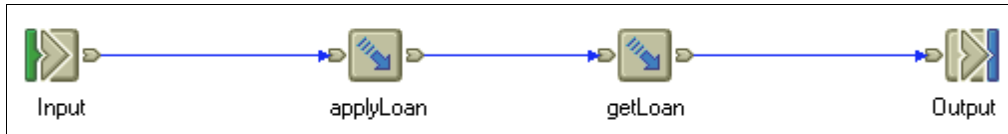


B. Add the Control links

To add the control links, proceed as follows.

1. Click the **Control Link** tool in the Palette.
2. Link the **Input** node to the input terminal of the **applyLoan** node.
3. Link the **applyLoan** node to input terminal of the **getLoan** node.
4. Link the output terminal of the **getLoan** node to the **Output** node.

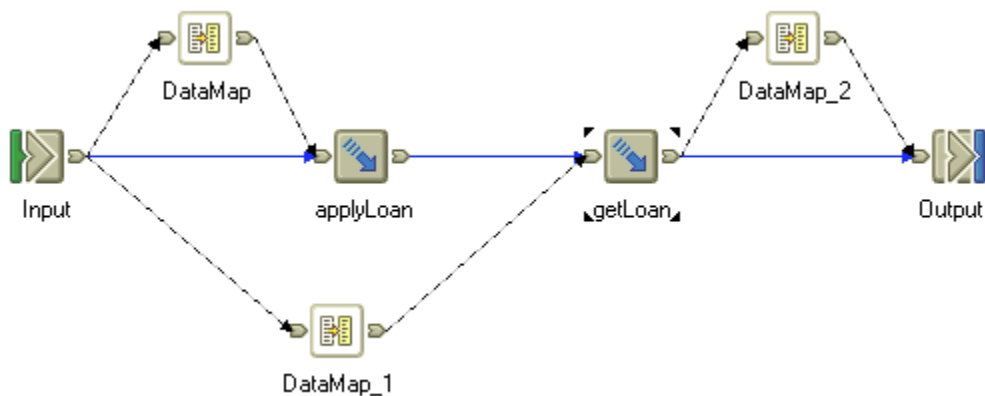
When you're done, your flow should look like this:



C. Add the Data Map nodes

To create and link the Data Map nodes, proceed as follows:

1. Create four Data Map nodes and place them in the order indicated below:
(To create a node in the Editor, click the **Data Map** tool in the palette, and click a location in the editor.)



2. Connect the Data Map nodes as shown using Data Links.

D. Mapping the Data

To instruct each individual Data Map node on how to change the message types passed between Activity nodes, you must make changes to the Java code.

1. Click the **Selection** tool in the Palette window.
2. Right-click the data map node labeled **DataMap**, and select **Open mapping** from the drop-down list.
3. Copy the wanted code from the blue comment tags into the green user code tags, and modify it accordingly.

Or copy the following code:

```
WSIFMessage inputMessage1 = (WSIFMessage)inputMessages.get("in1");
String lastname1 = (String)inputMessage1.getObjectPart("lastname");
String firstname1 = (String)inputMessage1.getObjectPart("firstname");
String ssn1 = (String)inputMessage1.getObjectPart("ssn");
loan.cbl.INCUSTOMERINFO info = new loan.cbl.INCUSTOMERINFO();
info.setLl((short)41);
info.setZz((short)0);
info.setTrcd("CDEBTRN2");
info.setLastname(lastname1);
info.setFirstname(firstname1);
info.setSsn(ssn1);
outputMessage.setObjectPart("INCUSTOMERINFOPart", info);
```

4. Scroll down to the **DataMap_1** function.
5. Copy the wanted code from the blue comment tags into the green user code tags, and modify it accordingly.

Or copy the following code:

```
WSIFMessage inputMessage1 = (WSIFMessage)inputMessages.get("in1");
String years1 = (String)inputMessage1.getObjectPart("years");
String amount1 = (String)inputMessage1.getObjectPart("amount");
loan.cbl.INFINANCIALNEED info = new loan.cbl.INFINANCIALNEED();
info.setLl((short)17);
info.setZz((short)0);
info.setYears(years1);
info.setAmount(amount1);
outputMessage.setObjectPart("INFINANCIALNEEDPart", info);
```

6. Scroll down to the **DataMap_1** function.
7. Copy the wanted code from the blue comment tags into the green user code tags, and modify it accordingly.

Or copy the following code:

```
WSIFMessage inputMessage1 = (WSIFMessage)inputMessages.get("in1");
loan.cbl.OUTLOANRESULT OUTLOANRESULTPart1 =
(loan.cbl.OUTLOANRESULT)inputMessage1.getObjectPart("OUTLOANRESULTPart");
outputMessage.setObjectPart("result", OUTLOANRESULTPart1.getResult());
```

8. Save and exit.

E. Finishing Up

1. To save your flow, type **Ctrl-S**.
The flow is saved in **LoanFlowPortTypeFlow.wsdl**, and the following files are either generated or updated:
 - o service file with flow bindings,
 - o java class with condition methods,
 - o deployed flow (*.flow)
 - o editor state file (*.serviceflow).
2. To see all the generated files, refresh the Packages view by selecting the view, and pressing **F5**.

Part III Deploy a flow service

Step 9: Configure the WebSphere Unit Test Environment

A. Creating a new server and server instance.

1. Open a Server perspective.
To do this from the main menu, click **Perspective > Open > Other** and select **Server** from the list.
2. In the Server Configuration view, right-click **Server Instances** and select **New > Server Instance and Configuration** from the drop-down list.
3. In the Create a New Server Instance and Configuration window, enter `Loan_SERVER` in both the **Server name** and **Folder** fields.
4. In the **Server Instance type** field, expand **WebSphere Servers** and select the **WebSphere Test Environment**, and click **Finish**.
5. Click **Yes**, when you are asked if you want to create a new server project.

B. Setting up the connection factory.

You need to add an instance of the JCA connection factory and configure its properties. The connection factory, as its name implies, provides connections to the EIS on demand. You specify all the information needed by the resource adapter to connect to the particular instance of the EIS. For the IMS resource adapter, you must specify at least the `HostName`, `PortNumber` and `DataStoreName` properties that determine the IMS to connect to. These values determine the IMS that will be accessed through all the connections created by this instance of the connection factory. You also specify the JNDI lookup name under which the new connection factory instance would be available to components. With this lookup name, the components can quickly make a connection to the EIS. To add a connection factory, follow these steps:

1. Expand **Server Configurations** in the Server Configuration view.
2. Double-click the server configuration **Loan_SERVER**.
3. Click the **J2C** tab. In the Create Resource Adapter window click **Add** beside the J2C Resource Adaptors pane.
4. From the Resource Adapter Name drop-down list, select the resource adaptor named IMS. Click **OK**.
5. Click **Add** beside the J2C Connection Factories pane. The application client will look up this connection factory instance using the JNDI interface. The application client will then use this connection factory instance to get a connection to the underlying IMS.
6. On the Create Connection Factory window type the name `loan_cf`. You may need to click several times on this field to enter a value.
Type the JNDI name `loan_jndi`. Click **OK**. You may need to click on another field before you can click **OK**.
7. In the Resource Properties pane, type the property values appropriate for your environment. See [Providing IMS resource adapter connection properties](#) for a description of these properties. For example:
 - o In the **HostName** field type `ECDVL91.STL.IBM.COM`
 - o In the **DataStoreName** field type `SOCKEYE`
 - o In the **PortNumber** field type `9999`
8. Close the editor and click **Yes** to save the changes.

Step 10: Create the Session Bean

A. Creating the Enterprise Application Projects

1. Switch to the Enterprise Services perspective, and click the **Packages** tab.

-
2. From the main menu, click **File > New > Other**.
From the topic tree, click **J2EE**. Select **Enterprise Application Project** and click **Next**.
 3. Type `Loan_EAR_` as the name of the Enterprise application project.
 4. Clear the check-box beside **Application client project name**, and click **Finish**

B. Generate the Session Bean and SOAP deployment descriptor

1. Return to the Enterprise Services perspective.
2. Expand `Loan_FLOW\flow.loan\` and highlight `LoanFlowPortTypeFlow.wsdl`
3. From the main menu, select **File > New > Deployment**.
4. In the Deployment window, proceed as follows, and then click **Next**.
 - a. In the **Inbound Binding Type** drop-down menu, select **SOAP**.
 - b. In the **EAR Project** field, confirm `Loan_EAR_`.
 - c. Clear the **Generate helper classes** check-box.
5. Accept the default settings in the next few windows, and click **Finish**.

C. Add JNDI name in the session bean binding properties

1. Go to the Services view.
2. Expand Deployed Services.
3. Right click on `Loan_EAR_EJB`.
4. Select Open With > EJB Extension Editor.
5. Click on the Bindings tab.
6. Expand `Loan_EAR_EJB > LoanFlowPortTypeFlowService`, then select and click **ResourceRef loan...**
7. Enter `loan_jndi` for the JNDI name.
8. Save and exit the EJB Extension Editor.

D. Setting Transaction Properties

1. Go to the Services view.
2. Expand Deployed Services.
3. Right click on `Loan_EAR_EJB`.
4. Select Open With > EJB Editor.
5. Click on the Transaction tab.
6. Select `LoanFlowPortTypeFlowService`.
7. Click **Add**.
8. Select **Remote** for Method Type, `runLoan(...)` for Method, and **Required** for Transaction Type.
9. Save and exit the EJB Editor.

Step 11: Add the application to the server configuration

1. Switch to the Server perspective.
2. In the Server Configuration view, expand **Server Configurations**.
3. Right-click `Loan_SERVER`, and select **Add Project > Loan_EAR_** from the drop-down list.

Part IV

Test a Flow Service using the EJB Test Client Tool

Step 12: Testing the Session bean

To test your session bean using the IBM^(R) EJB Test Client tool, proceed as follows:

-
1. In the Servers view, right-click **Loan_SERVER** and click **Start** from the drop-down menu. Wait until the Console view reports that the server is open for e-business.
 2. Click the **Servers** tab.
 3. Right-click **Loan_SERVER** and select **Test Client** from the drop-down menu.
 4. In the IBM EJB Test Client view, click **JNDI Explorer**.
 5. In the JNDI Explorer panel, expand **flow.loan** and click **LoanFlowPortTypeFlowServiceHome**.
 6. Under the EJB References heading, expand **LoanFlowPortTypeFlowService > LoanFlowPortTypeFlowServiceHome**.
 7. Click **LoanFlowPortTypeFlowService_create()**.
 8. In the Parameters pane, click **Invoke**, and then click **Work with Object**.
 9. Under the EJB References heading, expand the newly created **LoanFlowPortTypeFlowService 1** bean to see a list of methods that can be executed in the bean.
 10. Click **String runLoan(String, String, String, String, String)**.
 11. In the Parameters view, enter lastname, firstname, ssn, years, and amount. Then click <Invoke>.
 12. The result will appear beside the **Last Result** heading. In this case, you will see the message "Loan Approved."
 13. Exit the Test Client environment.

Part V

Test a Flow Service using a SOAP Proxy

Step 13: Test the Web service

A. Create a Java project

Proceed as follows to create another service project for your flow.

1. Switch to the Enterprise Services perspective, and click the **Packages** tab.
2. From the main menu, click **File > New > Service Project**.
3. In the Service Project window, enter the project name `Loan_CLIENT` and click **Finish**.

B. Create the client proxy

1. Browse to `Loan_EAR_Web/webApplication/wsd/flow.loan` and highlight **LoanFlowPortTypeSOAP.wsdl**.
2. From the main menu, click **File > New > Proxy**.
3. In the Service Proxy window's **Destination folder**, browse to and click `Loan_CLIENT`, and then click **OK**.
4. Click **Next** to continue.
5. In the Service Proxy Style window, click **Command Bean**.
6. Click the check-box beside the **runLoan** operation as shown below, and click **Finish**.

C. Write the client application

To create a client program to use your soap proxy, proceed as follows:

1. Browse to `Loan_CLIENT`, right-click **flow.loan** and click **New > Class** from the drop-down menu.
 2. In the **Name** field, enter `LoanSOAPProxyClient`.
 3. Click the check-box beside **public static void main(String[] args)** and click **Finish**.
- Replace the existing code in the Java editor with the following:

```
package loan.flow;  
public class LoanSOAPProxyClient
```

```

{
    public static void main(String[] args) throws com.ibm.wsif.WSIFException
    {
        LoanFlowPortTypeProxy proxy = new LoanFlowPortTypeProxy();
        proxy.setFirstname("lastname");
        proxy.setLastname("firstname");
        proxy.setSsn("123456789");
        proxy.setYears("004");
        proxy.setAmount("10000");

        proxy.execute();

        System.out.println("Result:\t" + proxy.getResult());
    }
}

```

4. Exit and Save.

D. Execute the application

1. Switch to the Server Perspective and start the **Loan_SERVER** server.
2. Switch to the Enterprise Services perspective and highlight **LoanSOAPPProxyClient.java** in the **Loan_CLIENT** folder.
3. From the main toolbar, click the arrow beside the **Run** icon and select **Run > Java Application** from the drop-down menu.

The Debug Perspective will launch, and execute your Java application. Verify output in the console:

```
Result: Loan Approved.
```

Part VI

Deploy a Flow service to a production server

Step 14: Packaging a Flow service for the production server

A. Edit the module dependencies

Note: if your production version of WebSphere Application Server Enterprise Edition has the module visibility set to "Application" rather than to the default value of "Module", then proceed to [Step 14B](#) below.

To edit the module dependencies, complete the following steps:

1. In the Enterprise Services perspective, click the **J2EE** tab.
2. Expand the **Web Modules** folder, then right-click **Loan_EAR_Web** and select **Edit Module Dependencies**.
3. In the Module Dependencies window, click the check-box beside **Loan_EAR_EJB.jar**, and click **Finish**.

B. Export the EAR file

1. From the main menu, click **File > Export**.
2. In the Import window, select **EAR file** from the topic tree and click **Next**.
3. Select **Loan_EAR_** from the **What resources do you want to export?** drop-down list.
4. Click **Browse**, and specify an name and location for the EAR file (for example, C:\Loan.ear)
5. Click **Finish**.

Step 15: Deploying a Flow service to the production server

A. Start the Administrative console

1. From the main menu, click **Start > Programs > IBM WebSphere > Application Server V4.0 AE > Start Admin Server**.
2. From the main menu, click **Start > Programs > IBM WebSphere > Application Server V4.0 AE > Administrator's Console**.

B. Installing the resource adapter

To install the resource adapter, complete the following steps:

1. In the Administrative Console, from the **Console** menu, select **New > J2C Resource Adapter**.
2. In the **Name** field, type `loan_ra`.
3. Beside the **Archive file name** field, click the small square button to open the Install Driver page.
4. In the Install Driver page, select the node that you want to use to install the driver.
5. Beside the **Rar file** field, click **Browse** and then navigate the file system and locate and select the IMS RAR file. Click **Install** and then click **OK**.
6. From the **Console** menu, select **New > J2C Connection Factory**. The J2C Connection Factory wizard appears.
7. In the **J2C resource adapter** drop-down list, select `loan_ra`.
8. In the **Name** field, type `loan_jndi`.
9. Leave the **JNDI binding path** field blank.
10. Click the **Connections** tab and enter the property values appropriate for your environment. See [Providing IMS resource adapter connection properties](#) for a description of these properties. For example:
 - o In the **DataStoreName** field type `MYDSTOR`
 - o In the **HostName** field type `MYHOST.ABC.XYZ.COM`
 - o In the **PortNumber** field type `9999`
11. Click **OK**.

For additional information about installing resource adapters, see the WebSphere Application Server Advanced Edition documentation on how to install resource adapters and create connection factories.

C. Install the EAR file

1. From the **Console** menu, select **Wizards > Install Enterprise Application**. The Install Enterprise Application Wizard opens.
2. In the **Path** field, click **Browse** and navigate to your EAR file and select it; for example, `C:\Loan.ear`.
3. Click **Open**.
4. Click **Next** five times. In the Mapping Resource References to Resources page, click **Select Resource**. In the Select Resource dialog, select `loan_jndi` and click **OK**.
5. Click **Next** five more times again. In the Completing the Application Installation Wizard page, click **Finish**. When a message appears that asks you if you want to regenerate code, click **No**.

D. Start the application server

1. In the left frame of the Administrative console, expand **WebSphere Administrative Domain**, **Nodes**, `your_node_name`, and **Application Servers**.
2. Right-click **Default Server** and select **Start**.

Step 16: Testing a Flow service on the production server using a SOAP proxy

Prior to completing Step 16, confirm that you have properly created a SOAP proxy.

1. Launch the Integration Edition and switch to the Enterprise Services perspective.

-
2. Click the **Services** tab and expand **Service Projects/Loan_CLIENT/ flow.loan**. Double-click **LoanFlowPortTypeSOAP.wsdl**.
 3. In the WSDL editor, click the **LoanFlowPortTypeService** link under the **Services** heading.
 4. Click **LoanFlowPortTypeSOAPPort** under the **Ports** heading.
 5. On the Port page, click **Extensibility Elements**, and then proceed as follows:
 - a. Click **soap:address**
 - b. In the Properties table, change the **location** value from
`http://localhost:8080/Loan_EAR_Web/servlet/rcprouter` to
`http://host_name:9080/Loan_EAR_Web/servlet/rcprouter`.
 - c. Click **location** in the Property column to save the new value.
 6. Exit and save your changes.
 7. Switch to the Enterprise Services perspective and highlight **LoanSOAPProxyClient.java** in the **Loan_CLIENT** folder.
 8. From the main toolbar, click the arrow beside the **Run** icon and select **Run > Java Application** from the drop-down menu.

The Debug Perspective will launch, and execute your Java application. Verify output in the console:

```
Result: Loan Approved.
```