

Enable an IMS application as a Web service running in IMS SOAP Gateway

Skill Level: Intermediate

[Chao-Lin Liu \(cliu@us.ibm.com\)](mailto:cliu@us.ibm.com)

Software Developer, IBM®

[Evgueni Liakhovitch \(evgueni@us.ibm.com\)](mailto:evgueni@us.ibm.com)

Software Developer, IBM

[Yee-Rong Lai \(ylai@us.ibm.com\)](mailto:ylai@us.ibm.com)

Information Developer, IBM

[Shahin Mohammadi-Rashedi \(shahin@us.ibm.com\)](mailto:shahin@us.ibm.com)

IMS™ SOA Demonstration Team Lead

IBM Corporation

12 May 2009

Abstract

This tutorial takes you through a series of intensive hands-on exercises to transform an Information Management System (IMS) application to a Web service by using IBM Rational® Developer for System z® and IMS SOAP Gateway.

About this tutorial

This tutorial takes you through a series of intensive hands-on exercises designed to transform an IMS application to a Web service by using IBM Rational Developer for System z and IMS SOAP Gateway. IMS SOAP Gateway is a Web service solution that integrates IMS assets in a service-oriented architecture (SOA) environment. It enables IMS applications as Web services through easy deployment and configuration processes. The IMS SOAP Gateway solution integrates with the IMS Connect XML adapter function, which provides XML conversion support. This enables users to send and receive XML data without modification of existing IMS applications. Through the tooling in Rational Developer for System z, Web service artifacts can be easily generated. Artifacts include a Web Services Description Language (WSDL) file, the correlator file, and the XML converter. Rational Developer for System z generates the XML converter from COBOL copybooks to be used by the IMS Connect XML adapter. The IMS SOAP Gateway deployment utility completes the backend IMS connection configurations. IMS SOAP Gateway runs on multiple platforms, including z/OS® (IMS SOAP Gateway V10.1), Microsoft® Windows® XP, Microsoft Windows 2000, IBM AIX®, and zLinux.

Objectives

To help you understand and gain hands-on experience in extending IMS applications to the Web as a part of Web services. This tutorial demonstrates how using Rational Developer for System z and the SOAP Gateway utility makes the transformation processes easy. Upon completion of this tutorial, you will be able to:

*Enable an IMS application as a Web service running in IMS SOAP Gateway
© Copyright IBM Corporation 2009. All rights reserved.*

- Use Rational Developer for System z.
- Use the IMS SOAP Gateway deployment utility to deploy an IMS application as a Web service.

System requirements

Software installed on the Microsoft Windows operating system:

- IMS SOAP Gateway Version 10.1 with the latest iFix
- Rational Developer for System z Version 7.5

System software installed on IBM z/OS

- IMS Version 10
- IMS Connect Version 10 with XML Adapter configured
- IMS OTMA
- TCP/IP

Checklist for first-time implementation of a Web service

We recommend that you have the checklist in Table 1 available when you are planning for your own Web service implementation the first time. It includes a checklist for this tutorial.

Table 1. Table name

Information or resource	Your environment	For this tutorial
COBOL copybook	Get from IMS application programmers	C:\IMS PhoneBook\IMSPHBK.cpy
IMS SOAP Gateway host name (or IP address) and port number	Get from IMS application programmers	Host name: localhost Port number: 8080
IMS Connect host name (or IP address) and port number	Get from IMS application programmers	Host name: ZSERVEROS.DFW.IBM.COM Port number: 9999
IMS data source (IMS ID)	Get from IMS application programmers	IMSC
IMS SOAP Gateway installation directory	Get from the support team that installed the SOAP Gateway	C:\Program Files\IBM\IMS SOAP Gateway V10.1
Workspace directory and project name to be used by Rational Developer for System z when generating artifacts	A naming standard is recommended	Workspace directory: C:\Workspaces7.5\SANDBOX Project name: IMSPhoneBook

Overview of development tasks

Internet Management Specification (IMS) applications can be enabled as Web services running in an IMS SOAP Gateway without the need for a full-fledged application server, such as IBM WebSphere® Application Server. Different types of client applications, such as Microsoft .NET, Java™, and third-party applications, can send SOAP requests into IMS to drive the business logic of your IMS applications.

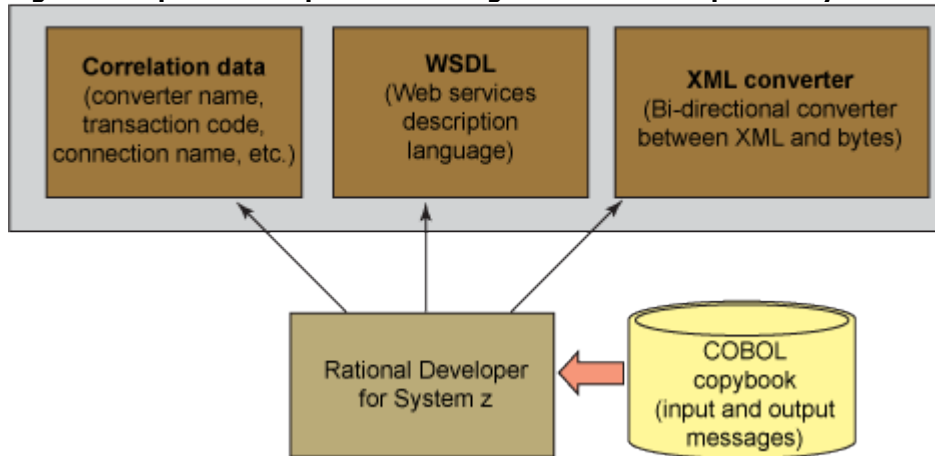
This tutorial uses the IMS phonebook application (IVTNO) as an example to demonstrate how to make it accessible as a Web service. The client application (a Java application) for accessing the Web service is provided here.

To complete this tutorial, you will go through the steps of enabling an IMS application as a Web service by using IBM Rational Developer for System z to generate the necessary artifacts and then complete the deployment processes by using the IMS SOAP Gateway utility.

To complete this tutorial, you will perform the following five tasks:

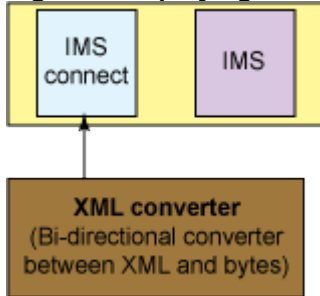
1. **Generate the artifacts by using Rational Developer for System z** (see the diagram in Figure 1).
 - A. Import the COBOL copybook.
 - B. Specify the input and output data structure.
 - C. Set the IMS SOAP Gateway host name and port number.
 - D. Set the IMS transaction code and connection information.
 - E. Specify the converter names.

Figure 1. Input and output when using Rational Developer for System z



2. **Transfer the generated XML converter to IBM z/OS.** Compile and link the converter for the IMS Connect XML adapter (Figure 2).

Figure 2. Deploying the XML converter to IMS Connect



3. **Enable the IMS application as a Web service.** Deploy the generated WSDL file and configuring the correlation properties by using the IMS SOAP Gateway V10.1 Deployment Utility (Figure 3).

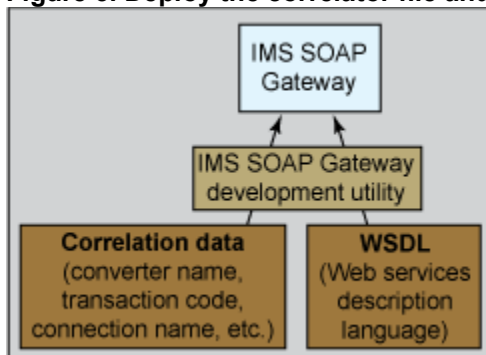
A. Provide the WSDL file that is generated by Rational Developer for System z.

- Set the connection information to connect to IMS.
- Set the IMS Connect host name and port number.

B. Set the IMS data store name (IMS ID).

C. Provide the correlator file path.

Figure 3. Deploy the correlator file and WSDL file by using the IMS SOAP Gateway deployment utility



Enable an IMS application as a Web service running in IMS SOAP Gateway
© Copyright IBM Corporation 2009. All rights reserved.

4. **Deploy a client application to access the IMS PhoneBook Web service.** Write or modify an existing Java application to access the IMS application that has been deployed as a Web service.

5. **Invoke the Java application that accesses the IMS application.**

Task 1. Generate the artifacts by using IBM Rational Developer for System z

IBM Rational Developer for System z is an application development tool that helps in the development of traditional mainframe applications. You can generate the Web service artifacts shown in Table 2 by using the Enterprise Service Tools (EST) wizard provided.

Table 2. Web service artifacts

Generated file	Description
WSDL file	The WSDL file is the Web service interface for the IMS application. It describes where the Web service is located and what the input and output message format is when the IMS application is invoked. You can create a WSDL file by hand or by using an application development tool (such as IBM Rational Developer for System z) and then customize it to be used with IMS SOAP Gateway. Based on the language structures from your application, the Enterprise Service Tools wizard needs to know which data structure is the input to invoke the application.
XML converter driver file	IMS Connect has an XML adapter function that, based on this converter driver, transforms XML messages from the client into COBOL bytes for the IMS application and then from bytes back to XML for the response. This file needs to be deployed to IMS Connect.
Correlator file	This file contains information that enables the IMS SOAP Gateway to set IMS properties and call the IMS application.

Open the Enterprise Service Tools and create an IMS SOAP Gateway project

1. The IBM Rational Developer for System z V 7.5 is started and you are using the Workspaces7.5 \SANDBOX.

To start, the Enterprise Service Tools wizard needs to gather the host runtime, development scenario, application mode, and XML conversion type.

2. Click **Start** to open the dialog box for specifying the request and response message layouts.

What you have done so far

You have created an IMS SOAP Gateway project in Rational Developer for System z by using the Enterprise Service Tools wizard. You imported the COBOL copybook for the IMS phonebook application, the type of project, the host runtime, the development scenario, the application mode, and the conversion type so that the wizard can proceed to gather specific data based on your selections so far.

Select data structures for the request and response message converters

Key step:

Specify the request message and response message structure.

Based on the phonebook COBOL copybook, the Enterprise Service Tools needs to gather the information that Table 3 shows

Table 3. Information the Enterprise Service Tools wizard gathers

Information	Description
The request message layout in your application	Based on the language structures from your application, the Enterprise Service Tools wizard needs to know which data structure is the input (request message) to invoke the application.
The response message layout in your application	Based on the language structures from your application, the Enterprise Service Tools wizard needs to know which data structure is the output (response message) from the application.
The prefix for the	The Enterprise Service Tools wizard generates the converters based on the prefix. The wizard lets users choose to generate either separate converter files for inbound and outbound messages or one driver for

*Enable an IMS application as a Web service running in IMS SOAP Gateway
© Copyright IBM Corporation 2009. All rights reserved.*

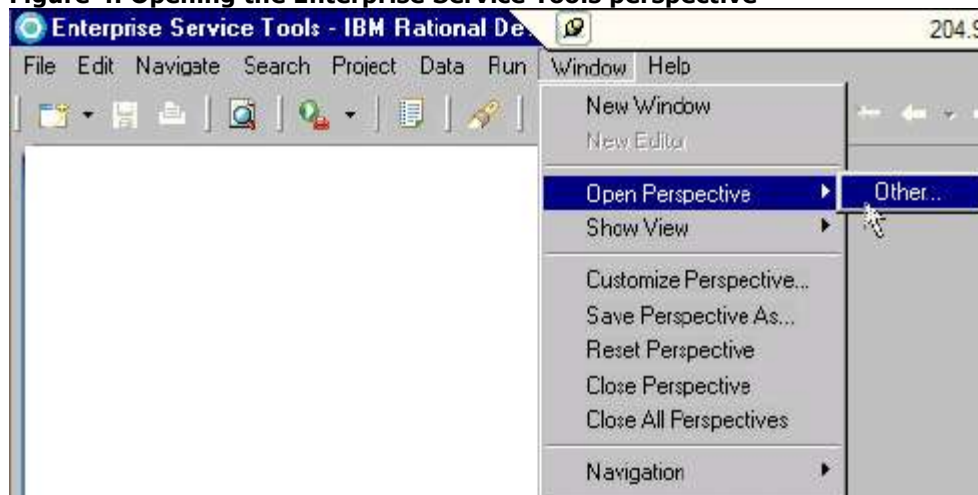
converter name	all converters. If users choose the former, the wizard appends a different character to the prefix to create different converter files. If users choose the latter, the wizard appends a D to the prefix.
Character encodings	This is the code page to use for encoding the inbound and outbound XML messages and the code page that is used by the z/OS host system.
Transaction code for your IMS application	The transaction code is required to run the IMS application.
Socket timeout and execution timeout values	The time that the IMS SOAP Gateway should wait for a response from IMS Connect and for IMS Connect to send a message to IMS and receive a response before a timeout.

Important:

For this tutorial, you are using **C:\Workspaces7.5\SANDBOX** as your work workspace.

3. Open the Enterprise Service Tools perspective (Figure 4): **Window > Open Perspective > Other.**

Figure 4. Opening the Enterprise Service Tools perspective

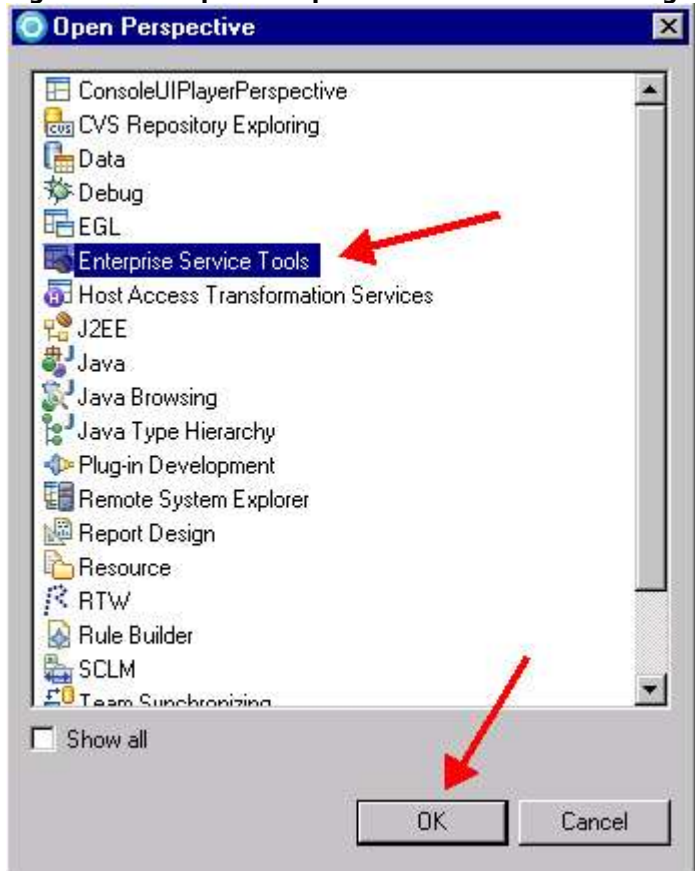


The workspace

A workspace is a directory that stores files for your projects. You can select your own directory or take the default directory. Artifacts created by Rational Developer for System z will be stored in this directory.

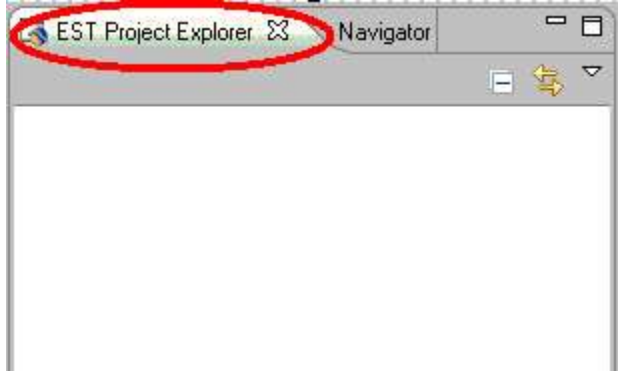
4. When the Open Perspective window opens, select **Enterprise Service Tools** and then click **OK** (Figure 5).

Figure 5. The Open Perspective window for choosing a perspective



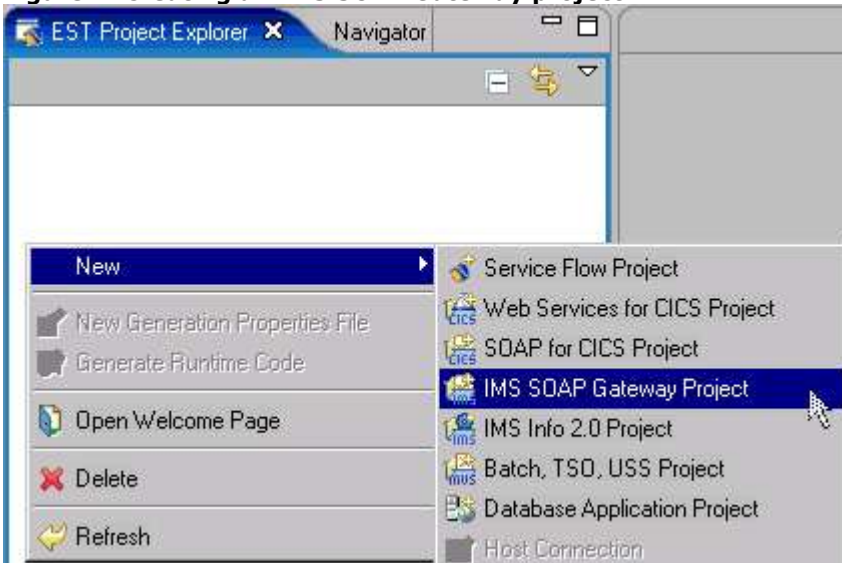
Next, you will see the EST Project Explorer tab (Figure 6).

Figure 6. The EST Project Explorer in Rational Developer for System z



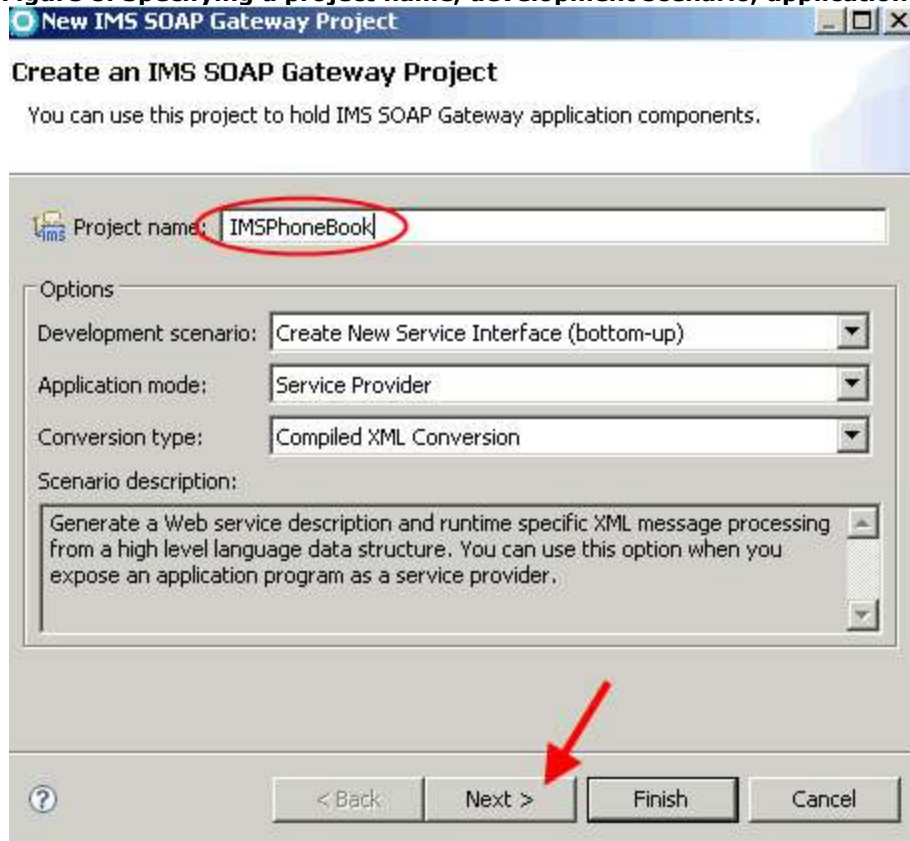
5. Right-click in the EST Project Explorer window (Figure 7), and select **New > IMS SOAP Gateway Project**.

Figure 7. Creating an IMS SOAP Gateway project



6. When the New IMS SOAP Gateway Project wizard opens (Figure 8), enter a project name (use IMSPhoneBook for this example).

Figure 8. Specifying a project name, development scenario, application mode, and conversion type



The following are the default selections. Ensure that these options are selected:

- Development scenario: **Create New Service Interface (bottom-up)**
- Application mode: **Service Provider**
- Conversion type: **Compiled XML Conversion**

Key terms

Bottom-up development scenario

The Enterprise Service Tools wizard supports the creation of service interfaces for several different types of projects, one of which is the IMS SOAP Gateway. Different projects support different options.

Create New Service Interface

This creates a bottom-up development scenario, which is for scenarios where you already have an enterprise application with existing language structures that you want to turn into a Web service.

Service Provider application mode

The Service Provider application mode means that the enterprise application serves as a Web service provider that receives requests from external clients, rather than initiating an outbound request to an external Web service.

7. Click **Next**. A window opens for you to import the application source file.

Key step:

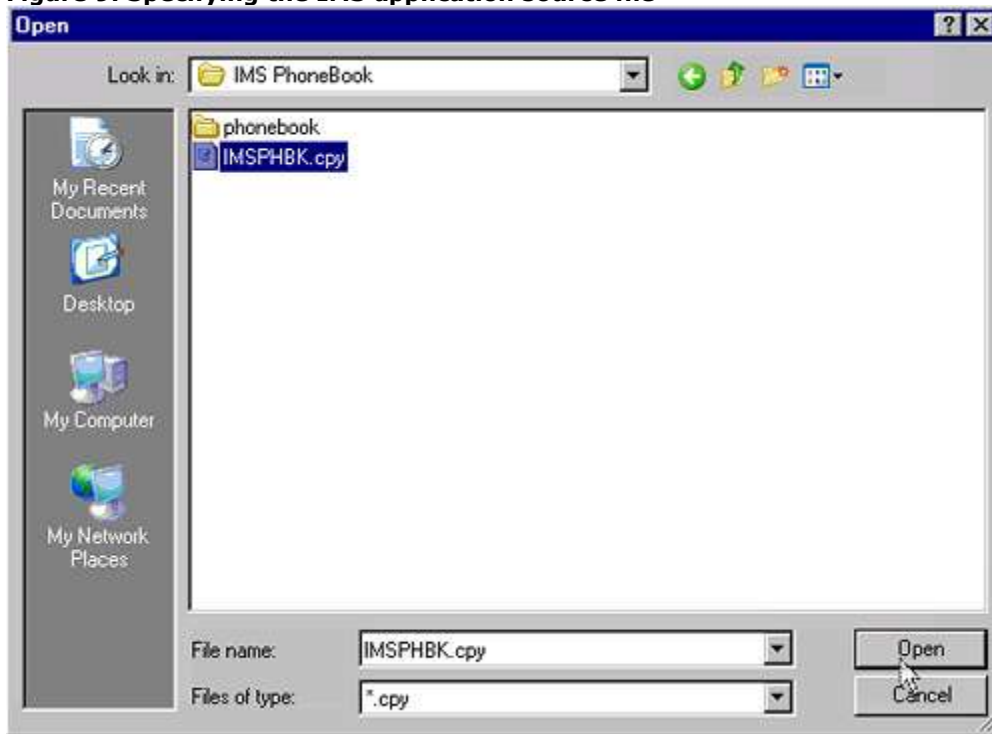
Import the COBOL copybook.

8. Click **File System** to import the COBOL copybook that describes the format of the input and output messages of your IMS application into the project that you created.

9. Navigate to **C:\IMS PhoneBook** (Figure 9). This directory contains the PhoneBook COBOL copybook and other files that you will need later.

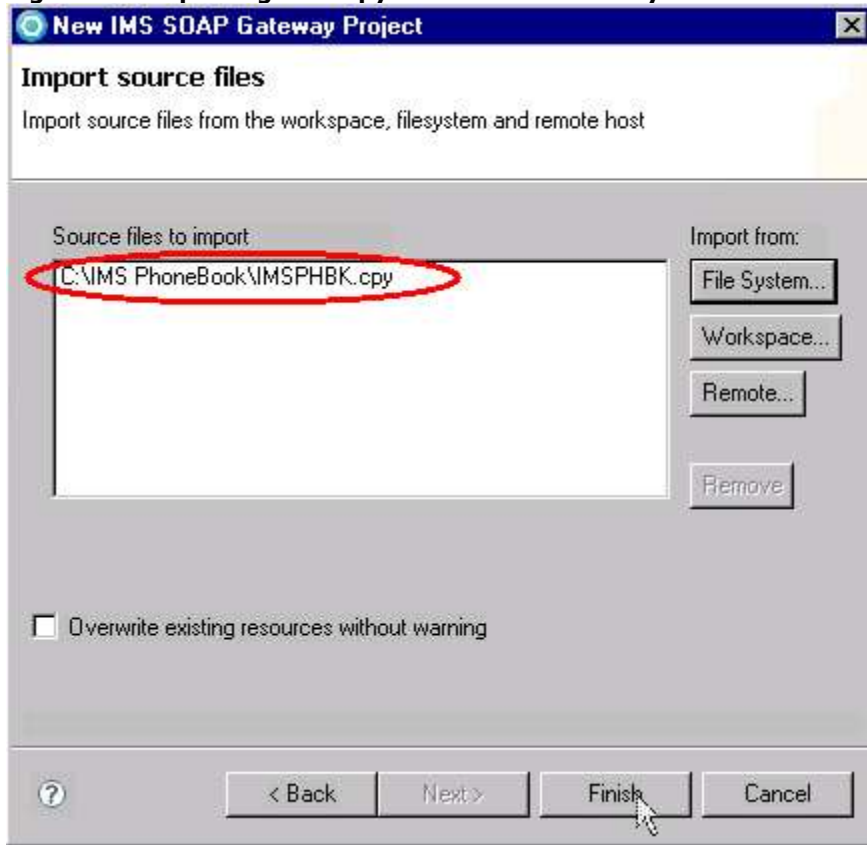
10. Select the **IMSPHBK.cpy** file, and click **Open**.

Figure 9. Specifying the IMS application source file



11. Make sure that the content of the "Source files to import" field is pointing to the correct file (Figure 10), and then click **Finish**.

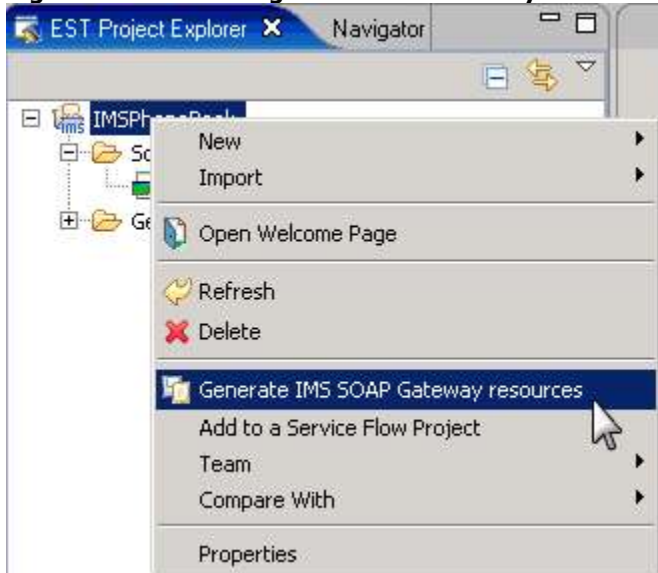
Figure 10. Importing the copybook from the File System



A new project called IMSPhoneBook is now available in your EST Project Explorer.

12. Right-click the project name, **IMSPhoneBook**, in the EST Project Explorer and then select **Generate IMS SOAP Gateway resources** (Figure 11).

Figure 11. Generating IMS SOAP Gateway resources

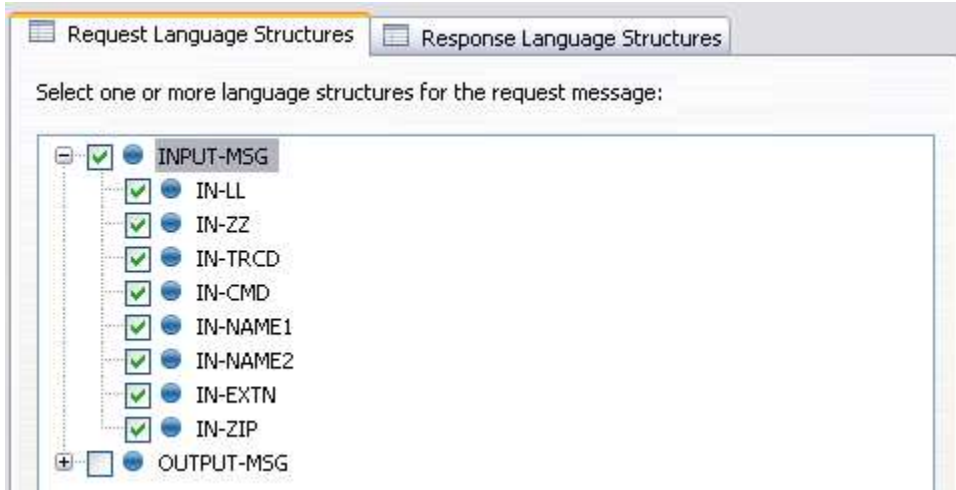


The Enterprise Service Tools Wizard launches and the Create New Service Interface page is displayed.

Enable an IMS application as a Web service running in IMS SOAP Gateway
© Copyright IBM Corporation 2009. All rights reserved.

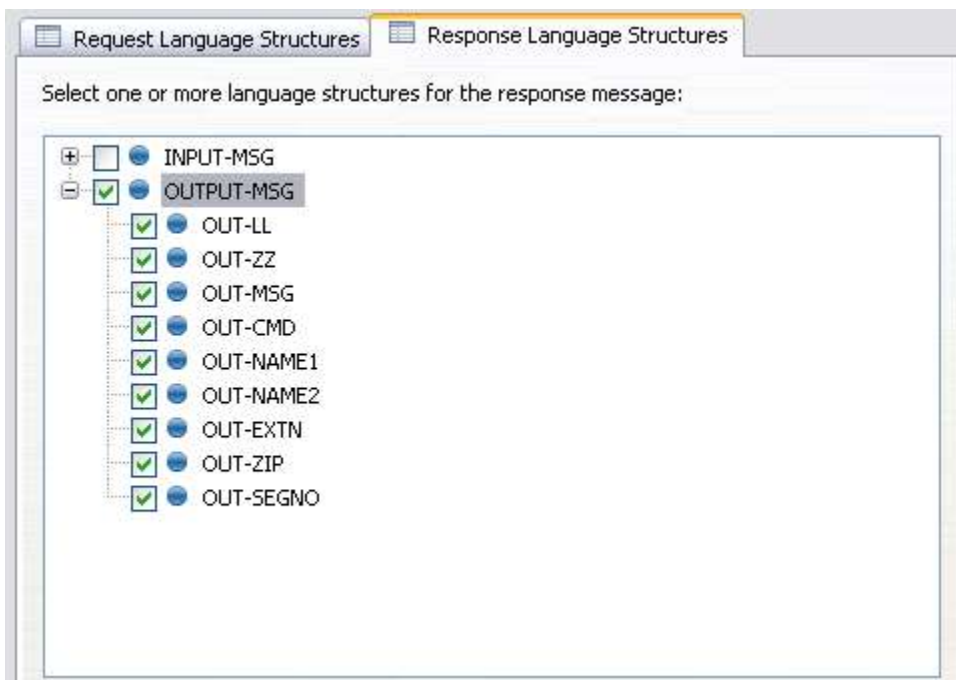
13. In the Create New Service Interface page, specify the layout of the request and response messages:
- In the **Request Language Structure** tab, ensure INPUT-MSG is listed as the language structure for the request message.

Figure 12. Specifying the request message layout



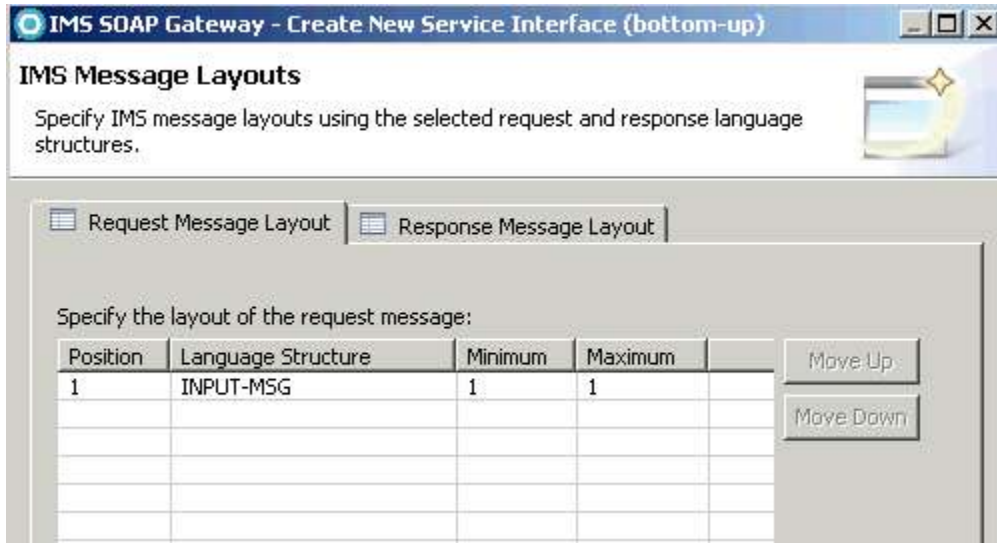
- Click the **Response Language Structure** tab, and ensure OUTPUT-MSG is listed as the language structure for the response message.

Figure 13. Specifying the outbound language structure



- Click **Next** to display IMS message layouts (Figure 14). Because there is only one language structure for the request message and one language structure for the response message. We do not need to make any adjustment or modification here. Click **Next** to continue to the *Generation options* page.

Figure 14. IMS message layouts



Specify generation options

Key step:

Set the IMS SOAP Gateway host name and port number.

15. In the Generation options page, under the XML Converters tab (for this tutorial), ensure that the converter program prefix **IMSPHBK** is unchanged. You can leave other default options as they are (see Figure 15).

Figure 15. Specifying the converter generation options

The screenshot shows a dialog box titled "Generation options" with the subtitle "Specify generation options for the Web Services enablement artifacts." It has three tabs: "XML Converters", "WSDL and XSD", and "Advanced Options". The "XML Converters" tab is active. Under "Specify identification attributes", there are three text boxes: "Converter program name prefix" (containing "IMSPHBK" and circled in red), "Author name" (containing "WD4Z"), and "Service program name" (containing "IMSPHBK"). Under "Specify Enterprise COBOL for z/OS properties", there are two dropdown menus: "Compiler level" (set to "4.1") and "XMLPARSE option" (set to "COMPAT Enterprise COBOL for z/OS XML par"). There is a checked checkbox for "Optimization". Under "Specify character encodings", there are two dropdown menus: "Request code page" (set to "1208 Unicode, UTF-8") and "Host code page" (set to "1140 USA, Canada, etc. EBCDIC with Eur"). At the bottom, there are buttons for "?", "< Back", "Next >", "Finish", and "Cancel".

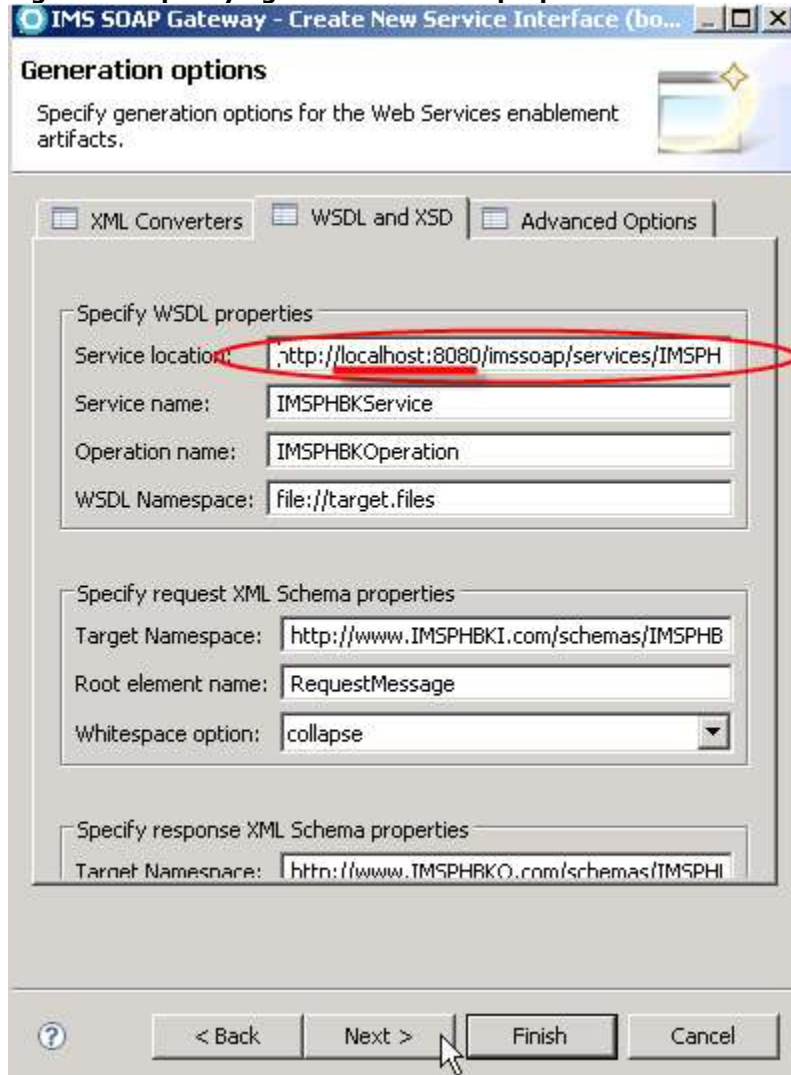
Note:

You must use this IMSPHBK prefix for this tutorial so that you do not need to manually deploy the XML converters to IMS Connect. A converter driver with that name is already deployed for this tutorial.

16. Select the **WSDL and XSD** tab.

17. In the Service location field, change the **hostname** and **port number** to the location of IMS SOAP Gateway: `http://localhost:8080/imsssoap/services/IMSPHBKPort`. This URL specifies the address of the Web service.

Figure 16. Specifying the Web service properties



4. Click **Next** to continue.

Specify the IMS SOAP Gateway correlation properties

Key step:

Set the IMS transaction code and connection information.

18. In the IMS SOAP Gateway Correlator file tab, enter the following values (shown in Figure 17):

Transaction code: IVTNO

Inbound connection bundle: IMSPBCon

Socket timeout: 6000

19. Use the default values for the other IMS Connect interaction properties and click **Next**.

Figure 17. Specifying IMS Connect interaction properties

IMS SOAP Gateway - Create New Service Interface (bottom-up)

IMS SOAP Gateway Web Service Provider

Specify properties for defining the Web service to IMS SOAP Gateway.
The IMS Correlator file is used with the IMS SOAP Gateway deployment

IMS SOAP Gateway Correlator file

Specify service identification properties

Generate to: Same project Remote location

SOAPAction: urn:IMSPHBK

File container: /IMSPhoneBook/Generation/Targets

File name: IMSPHBK.xml

Update Overwrite

Specify IMS Connect interaction properties

Transaction code: IVTNO

Inbound connection bundle: IMSPBCon

Socket timeout: 6000 (in milliseconds)

Execution timeout: 5000 (in milliseconds)

LTERM name:

Set the socket timeout to a value larger than the execution timeout.

Notes:

- The inbound connection bundle name must match the name of the connection bundle that you will specify later in the IMS SOAP Gateway deployment utility.
- The socket timeout value should be larger than the execution timeout value. For this tutorial, 6000 is arbitrarily selected.

20. Under the XML Converters tab (Figure 18), use the default value for the converter driver file name. The inbound and outbound converter file names are automatically filled in based on the converter driver file name.

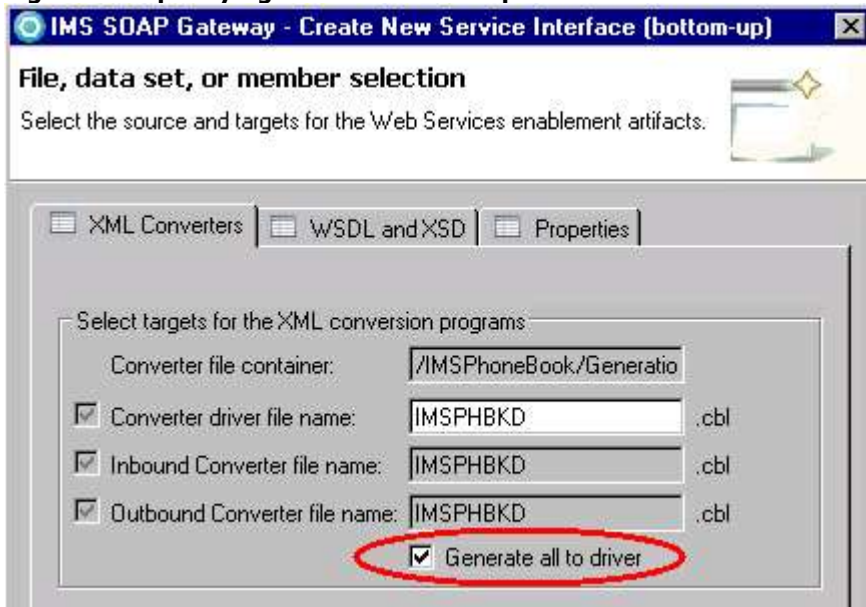
Important:

For this tutorial, you must use the default converter file names to use the XML converters that are already set up for you in IMS Connect. This setting ensures that the demo system can function properly for everyone.

21. Make sure that the "Generate all to driver" check box is checked.

22. Click **Finish**.

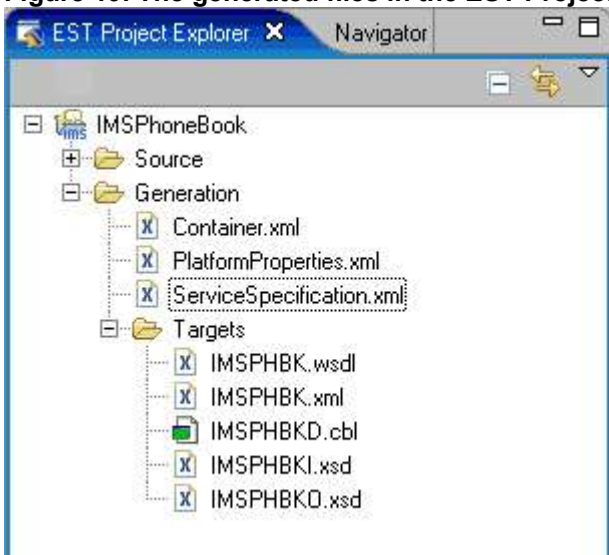
Figure 18. Specifying XML converters options



The following files are generated in your Rational Developer for System z workspace (see Figure 19):

- **IMSPHBK.wsdl:** WSDL file
- **IMSPHBK.xml:** Correlator file
- **IMSPHBKD.cbl:** COBOL converters in one driver file
- **IMSPHBKI.xsd:** Inbound XSD file (not used by the IMS SOAP Gateway)
- **IMSPHBKO.xsd:** Outbound XSD file (not used by the IMS SOAP Gateway)

Figure 19. The generated files in the EST Project Explorer view



What you have done so far

You specified the input and output data structures in the phonebook COBOL copybook, XML conversion options, and IMS Connect interaction properties in the Enterprise Service Tools wizard. The wizard generated the WSDL file, the correlator file (an XML file), and an XML converter driver. These files are located in your Rational Developer for System z workspace.

Task 2. Transfer the generated XML converter to IBM z/OS

IMS Connect has an XML adapter function that converts data between XML and bytes, based on a deployed converter driver. Typically, to deploy a converter driver, you need to complete these two tasks:

1. Use FTP to transfer the XML converter driver that is generated by Rational Developer for System z to the z/OS system.
2. Compile and link the XML converter driver into a data set that is concatenated to the STEPLIB in the IMS Connect startup JCL (job control language).

Important:

This IMSPHBKD.cbl converter driver is part of the sample that is included with IMS Connect. Therefore, no action is needed here. You can proceed directly to task 3: Deploy WSDL and configure properties with IMS SOAP Gateway.

Task 3. Enable the IMS application as a Web service

The deployment step helps you set up properties and create runtime code that will be used by the IMS SOAP Gateway to make your IMS application accessible through a Web service.

IMS SOAP Gateway provides a deployment utility that can help you perform the following tasks:

- Deploying easily end-to-end

- Setting up and modifying connection and correlation information (hostname, port number, trancode, and timeout, for example)

- Generating the runtime code from the WSDL that will be used by IMS SOAP Gateway to enable the IMS application as a Web service

The deployment utility can take you through these tasks, or you can perform each task individually, either through the interactive mode or command-line interface

Key points:

In the IMS SOAP Gateway deployment utility:

1. Provide the WSDL file that is generated by Rational Developer for System z.
2. Set the connection information to connect to IMS.
 - A. Set the IMS Connect host name and port number.

 - B. Set the IMS data store name (IMS ID).

3. Provide the correlator file path.

Deploy the generated artifacts

1. Copy the WSDL file and the correlator file that were generated by Rational Developer for System z from the previous section into the appropriate directories. For this tutorial (see Figure 20):
 - The Rational Developer for System z workspace is this directory:C:\Workspaces7.5\SANDBOX

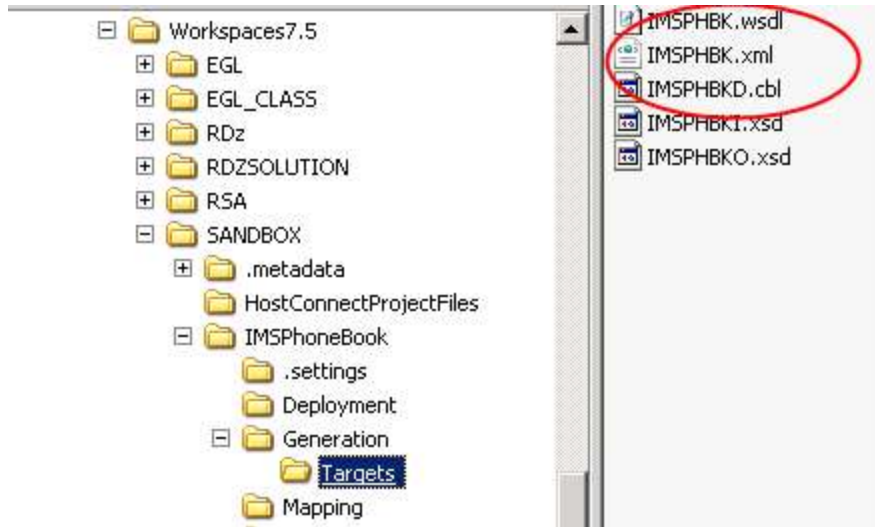
 - The WSDL file for the IMSPhoneBook project is located in C:\Workspaces7.5\SANDBOX\IMSPhoneBook\Generation\Targets

 - The IMS SOAP Gateway installation directory is C:\Program Files\IBM\IMS SOAP Gateway V10.1 (the default)
 - A. Copy the **IMSPHBK.wsdl** file from the Rational Developer for System z workspace into the IMS SOAP Gateway WSDL directory at *installation_directory\server\webapps\imsssoap\wsdl*. For this

tutorial, copy the WSDL file into C:\Program Files\IBM\IMS SOAP Gateway V10.1\server\webapps\imssoap\wsdl

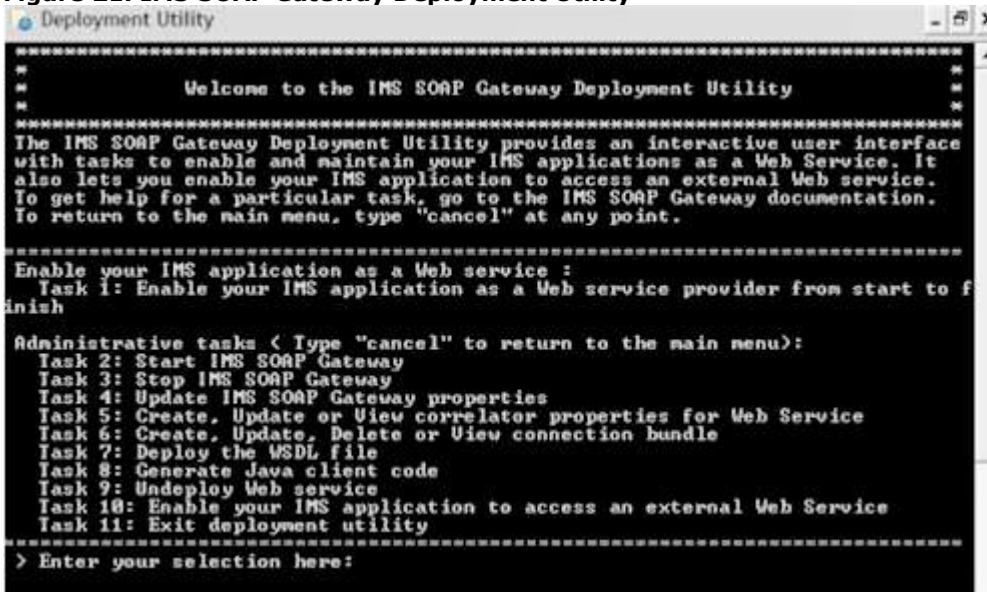
B. Copy the correlator file (**IMSPHBK.xml**) from the Rational Developer for System z workspace into the IMS SOAP Gateway XML directory at *installation_directory*\server\webapps\imssoap\xml. For this tutorial, copy it into C:\Program Files\IBM\IMS SOAP Gateway V10.1\server\webapps\imssoap\xml

Figure 20. The generated files in Windows Explorer view



2. Start the IMS SOAP Gateway. From the Windows Start menu, select **Programs > IBM IMS SOAP Gateway Version 10.1 > Start Server**. You will see a message from the console: "IMS SOAP Gateway server is now up and running."
3. Start the deployment utility in the interactive mode. From the Windows Start menu, select **Programs > IBM IMS SOAP Gateway Version 10.1 > Deployment Utility**. The deployment utility interactive mode starts and displays the main menu shown in Figure 21.

Figure 21. IMS SOAP Gateway Deployment Utility



The main menu lists the tasks that you can perform with the deployment utility.

4. Type 1 (one) and press **Enter** to select Task 1: Enable your IMS application as a Web service from start to finish.

5. The deployment utility then guides you through the following steps to complete the deployment task:

- Specify which WSDL file to use for the deployment
- Specify the connection and security information for the Web service to create a connection bundle
- Specify the interaction properties of the Web service to create a correlator file
- Deploy the Web service to IMS SOAP Gateway
- Create the Java client proxy code

6. Enter the values that are in code type (Courier New) in the steps that follow when you are prompted.

A. Provide the WSDL file for deployment. Provide the name and location of the WSDL file (for example c:\MyWSDLFile.wsdl): `IMSPHBK.wsdl`

Note: You need to specify only the file name here, because the WSDL file was copied into the IMS SOAP Gateway's WSDL directory. Otherwise, you would need to specify the full path.

B. Provide connection properties for connecting to IMS:

- Do you want to view all existing connection bundles? [Y/N]: `n`
- Do you want to create a connection bundle? [Y/N]: `y`
- Provide the name of the connection bundle. **For this sample, enter `IMSPBCon`**
- Provide IMS host name or IP address. **For this sample, enter `ZSERVEROS.DFW.IBM.COM`**
- Provide IMS port number(default: 9999). **For this sample, enter `9999`**

- Provide IMS datastore name. **For this sample, enter IMSC**
- Provide IMS userid (optional). **For this sample, press Enter.**
- Provide IMS password (optional). **For this sample, press Enter.**
- Provide IMS group name (optional). **For this sample, press Enter.**
- Do you want to set SSL properties? [Y/N]: **For this sample, enter N**
The connection bundle, `IMSPBCON`, has been created.

C. Provide interaction (correlator) properties for Web services

Note: If you use the IBM-provided Rational Developer for zSeries® solution, the correlator file is generated for you as part of that process. If you use your own solution, you must create the correlator file.

- Do you want to use an existing Correlator file? [Y/N]: `y`
- Provide the name and full path to the Correlator file: `IMSPHBK.xml`
Note:
You need to specify only the file name here, because the correlator file was copied into IMS SOAP Gateway's <installation_directory>\server\webapps\imssoap\xml directory, and we have applied IMS SOAP Gateway V10 iFix 1. Otherwise, you need to specify the full path.
- Do you want to view the correlator file? [Y/N]: `y`
Note:
The correlator file is displayed. When you are finished reviewing, press Enter to continue.

D. Deploy the WSDL to IMS SOAP Gateway. This step deploys the WSDL file and makes your IMS application accessible as a Web service. **Note:** Ensure that the IMS SOAP gateway is already started.

- Do you want to deploy the WSDL file to IMS SOAP gateway? (y/n): `y`
- You will then see:
 - Processing file `C:\Program Files\IBM\IMS SOAP Gateway V10.1\server\webapps\imssoap\temp\server\deploy.wsdd`

- WSDL was successfully deployed.
- E. (Optional) Generate the Java client code. This step generates proxy code that you can use to build a Java application to invoke the Web service.
- Do you want to generate java client code (optional)? [Y/N]: *y*
 - Please enter the location to output client files <default: C:\Program Files\IBM\IMS SOAP Gateway V10.1\server\webapps\imssoap\temp\client>. **Press Enter to use the default.**
 - You will then see: Client code generation in directory <the SOAP Gateway install path entered above> is complete.

The Java client proxy code

The client proxy code is usually generated by a tool or utility and performs the following tasks:

1. Creates a connection to send and receive SOAP messages
2. Wraps the transaction data in a SOAP message

The IMS SOAP Gateway Deployment Utility can generate this client proxy code for you.

What you have done so far

- Generated the correlator file, the WSDL file, and the XML converter driver using Rational Developer for System z
- Deployed the XML converter driver to IMS Connect
- Deployed the phonebook application using the IMS SOAP Gateway Deployment Utility by specifying the correlator file, the WSDL file, connection information for IMS, IMS Connect host name and port number, and IMS data store name

The IMS phonebook application is now enabled as a Web service.

Verify the deployment of the IMS PhoneBook Web service

Your IMS applications are ready to be accessed by a client as a Web Service. To verify that the deployment has completed successfully, do the following:

7. Start the IMS SOAP Gateway Server Console by selecting, from the Start menu: **Programs > IBM IMS SOAP Gateway V10.1 > Administrative Console**. A Web browser opens and displays the console.

8. Click the **View deployed Web Services** link. You should see IMSPHBKPort listed in the page, which indicates that the IMS phonebook application is successfully deployed as a Web service.

Task 4. Deploy a client application to access the Web service

After the IMS application is successfully deployed in the IMS SOAP Gateway, you need to test whether you can access the Web service from a client application, such as a Java or Microsoft .NET application. The IMS SOAP Gateway deployment utility can help you generate the Java proxy code that you can use in your Java application (Step E in the previous section: [Generate the Java client code in the IMS SOAP Gateway deployment utility](#)).

For this exercise, an existing Java application is provided.

Deploy the Java application into IMS SOAP Gateway's webapps directory

1. Copy the phonebook directory from C:\IMS PhoneBook.
2. Paste the phonebook directory into the webapps file:
C:\Program Files\IBM\IMS SOAP Gateway V10.1\server\webapps

Now you should have a directory that looks like this:

C:\Program Files\IBM\IMS SOAP Gateway V10.1\server\webapps\phonebook

3. Stop IMS SOAP Gateway. From the Windows Start menu, select **Programs > IBM IMS SOAP Gateway V10.1 > Stop Server**.

Task 5. Invoke the Java application

1. Start IMS SOAP Gateway: From Windows' Start menu, select Programs > IBM IMS SOAP Gateway V10.1 > Start Server.
2. Launch Internet Explorer.
3. Go to the following URL: <http://localhost:8080/phonebook>

The Java application displays the screen that Figure 22 shows.

Figure 22. Invoking the sample Java application



4. Enter `Smith` as your inquiry and click **Look up** (see Figure 23).

Figure 23. Specifying a value to query the IMS phonebook application



The resulting display looks like Figure 24.

Figure 24. Query result from the IMS phonebook application

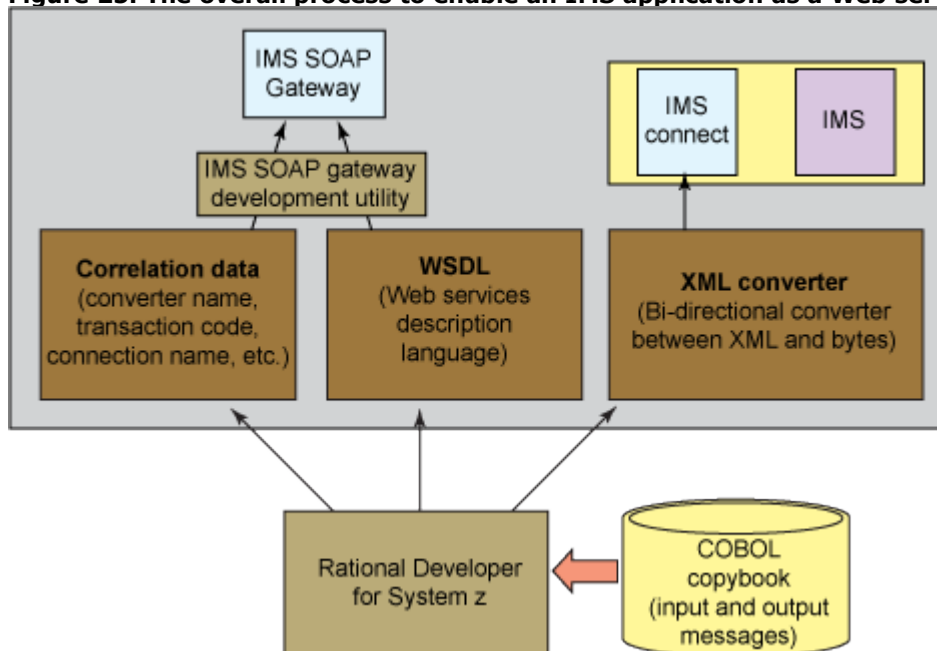


Summary of completed tasks

You have successfully completed the following tasks for this tutorial (illustrated in Figure 25):

1. Generated the WSDL file, correlator file, and converter driver for the IMS phonebook application by using Rational Developer for System z
2. Deployed the converter driver to IMS Connect (which was already done for you)
3. Enabled the IMS phonebook application as a Web service by deploying the generated WSDL and correlator files and using IMS SOAP Gateway deployment utility

Figure 25. The overall process to enable an IMS application as a Web service



*Enable an IMS application as a Web service running in IMS SOAP Gateway
© Copyright IBM Corporation 2009. All rights reserved.*

4. You then deployed a provided Java application that accessed the IMS phonebook application and invoked the IMS phonebook application from the Java application.

Congratulations! You have successfully completed this tutorial.

Resources

Learn

- Visit the [Rational software area on developerWorks™](#) for technical resources and best practices for Rational Software Delivery Platform products.
- Explore [Rational computer-based, Web-based, and instructor-led online courses](#). Hone your skills and learn more about Rational tools with these courses, which range from introductory to advanced. The courses on this catalog are available for purchase through computer-based training or Web-based training. Additionally, some "Getting Started" courses are available free of charge.
- Subscribe to the [Rational Edge e-zine](#) for articles on the concepts behind effective software development.
- Visit the [IMS SOAP Gateway product Web page](#).
- Check out the [Information Management IMS zone](#) on developerWorks.
- Subscribe to the [IBM developerWorks newsletter](#), a weekly update on the best of developerWorks tutorials, articles, downloads, community activities, webcasts and events.
- Browse the [technology bookstore](#) for books on these and other technical topics.

Get products and technologies

- Download [trial versions of IBM Rational software](#).
- Download a trial version of [IBM Rational Developer for system z](#).
- Download [IMS SOAP Gateway and two complimentary, unwarrented copies of Rational Developer for System z](#) for IMS Version 10 customers.

Discuss

- [Participate in the discussion forum](#).

- Check out [developerWorks blogs](#) and get involved in the [developerWorks community](#).

About the authors

- Chao-Lin Liu is a Software Developer at IBM
- Evgueni Liakhovitch is a Software Developer at IBM
- Yee-Rong Lai is an Information Developer at IBM
- Shahin Mohammadi-Rashedi is an IMS SOA Demonstration Team Lead at IBM

Trademark notice

IBM, the IBM logo and ibm.com are trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.