IBM WebSphere Adapter for IBM i 7.0.0.0

Quick Start Scenarios
Note: Before using this information and the product it supports, read the information in the “Notices” section, at the end of this document.

This edition applies to version 7, release 0, and modification 0 of IBM WebSphere Adapter for IBM i and to all subsequent releases and modifications, until otherwise indicated in new editions.

© Copyright International Business Machines Corporation 2008. US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
# Table of Contents

**Chapter 1.** Introduction ............................................................................................................. 1
   - Learning objectives .................................................................................................................. 1
   - Audience ................................................................................................................................. 1
   - Software prerequisites ............................................................................................................ 1

**Chapter 2.** Preparing to test tutorial 1 ................................................................................. 2
   - Configuration prerequisites ..................................................................................................... 2
   - Extracting the sample files ...................................................................................................... 2

**Chapter 3.** Tutorial 1: Outbound Processing Calling the RPG program .............................. 4
   - Scenario ................................................................................................................................... 4
   - Configuring the adapter for outbound processing .................................................................. 4
     - Creating the project .............................................................................................................. 5
     - Setting connection properties for the external service wizard ........................................... 8
     - Generating business object definitions and related artifacts ........................................... 9
   - Deploying the module to the test environment .................................................................... 15
   - Testing the assembled adapter application ......................................................................... 17

**Chapter 4.** Introduction to Outbound/Inbound for Data Queues ........................................ 21
   - Learning objectives ............................................................................................................... 21
   - Audience ............................................................................................................................... 21
   - Software prerequisites .......................................................................................................... 22

**Chapter 5.** Preparing to test tutorial 2 .................................................................................. 23
   - Configuration prerequisites .................................................................................................. 23
   - Extracting the sample files .................................................................................................... 23

**Chapter 6.** Tutorial 2: Outbound Processing – Writing the message to (PutQueue) and reading the message from (GetQueue) the data queue . 25
   - Configuring the adapter for outbound processing ............................................................... 26
     - Creating the project ............................................................................................................. 26
     - Setting connection properties for the external service wizard ........................................ 32
     - Generating business object definitions and related artifacts ......................................... 33
   - Deploying the module to the test environment .................................................................... 39
   - Testing the assembled adapter application ......................................................................... 40

**Chapter 7.** Tutorial 2 continued: Inbound processing on a data queue ................................. 47
Chapter 8. Introduction to Outbound/Inbound for Keyed Data Queues

Learning objectives.................................................................................................................... 69
Audience .................................................................................................................................... 69
Software Prerequisites............................................................................................................... 70

Chapter 9. Preparing to test tutorial 3 ............................................. 71

Configuration prerequisites........................................................................................................ 71
Extract the sample files.............................................................................................................. 71

Chapter 10. Tutorial 3: Outbound Processing –Writing (PutQueue) the message to and reading (GetQueue) the message from the keyed data queue

Scenario..................................................................................................................................... 73
Configuring the adapter for outbound processing ..................................................................... 74
Creating the project .................................................................................................................. 74
Setting connection properties for the external service wizard .............................................. 78
Generating business object definitions and related artifacts ............................................. 79
Deploying the module to the test environment ....................................................................... 85
Testing the assembled adapter application ........................................................................... 88

Chapter 11. Tutorial 3 continued: Inbound processing on the keyed data queue

Scenario..................................................................................................................................... 94
Configuring the adapter for inbound processing .................................................................... 94
Creating the project ................................................................................................................ 95
Setting connection properties for the external service wizard ............................................ 99
Generating business object definitions and related artifacts ............................................ 100
Deploying the module to the test environment ..................................................................... 109
Testing the assembled adapter application ......................................................................... 111

Chapter 12. Introduction to Outbound for Keyed Data Queues with support for binary content


Chapter 13. Preparing to test tutorial 4 ........................................ 116

Configuration prerequisites ................................................................. 116
Extract the sample files ................................................................. 116

Chapter 14. Tutorial 4: Outbound Processing – Binary Content Support for writing the message to (PutQueue) and reading the message from (GetQueue) the keyed data queue .............................................. 117

Scenario ................................................................................................. 117
Configuring the adapter for outbound processing ............................................ 118

Creating the project .............................................................................. 118
Setting connection properties for the external service wizard .................. 120
Generating business object definitions and related artifacts .................. 122

Deploying the module to the test environment ......................................... 128
Testing the assembled adapter application ............................................ 131

Chapter 15. Introduction to WebSphere Adapter for IBM i - Calling a Service program 135

Learning objectives .............................................................................. 135
Audience ............................................................................................... 135
Software prerequisites ........................................................................... 135

Chapter 16. Preparing to test tutorial 5 .............................................. 136

Configuration prerequisites ..................................................................... 136
Extracting the sample files .................................................................... 136

Chapter 17. Tutorial 5: Outbound Processing – Calling the Service program 138

Scenario ................................................................................................. 138
Configuring the adapter for outbound processing ............................................ 138

Creating the project .............................................................................. 138
Setting connection properties for the external service wizard .................. 142
Generating business object definitions and related artifacts .................. 143

Deploying the module to the test environment ......................................... 150
Testing the assembled adapter application ............................................ 152

Chapter 18. Introduction to WebSphere Adapter for IBM i - Calling a COBOL program 156
Learning objectives .................................................................................................................. 156
Audience .................................................................................................................................. 156
Software prerequisites ............................................................................................................. 156

Chapter 19. Preparing to test tutorial 6 .......................................................... 157
Configuration prerequisites ...................................................................................................... 157
Extracting the sample files ....................................................................................................... 157

Chapter 20. Tutorial 6: Outbound Processing – Calling the COBOL program 159
Scenario ................................................................................................................................... 159
Configuring the adapter for outbound processing ................................................................. 159
  Creating the project ............................................................................................................. 159
  Setting connection properties for the external service wizard ............................................ 163
  Generating business object definitions and related artifacts ............................................. 164
Deploying the module to the test environment ........................................................................ 171
Testing the assembled adapter application ............................................................................. 173

Notices .................................................................................................................................... 177
Chapter 1. Introduction

WebSphere® Adapter for IBM® i V7.0 exchanges business data between IBM I system and J2EE applications. The adapter retrieves from and writes to the data queue and calls RPG programs.

The document demonstrates the following scenario:

1. Calling a Report Program Generation (RPG) program with the help of Adapter for IBM i

This scenario demonstrates how WebSphere Adapter for IBM i V7.0 performs outbound operations.

Learning objectives

After completing this tutorial, you should be able to perform the following tasks:

Create an adapter project in WebSphere Integration Developer.

Discover services and associated business objects from the enterprise information system (EIS) and make them part of the adapter project.

Create a deployable module that you install on WebSphere Process Server or WebSphere Enterprise Service Bus.

Test the module and validate the results.

Audience

This tutorial is for integration developers who design, assemble, test, and deploy business integration solutions.

Software prerequisites

To use this tutorial, you must have the following applications installed:

WebSphere Integration Developer version 7.0

WebSphere Process Server version 7.0
Chapter 2. Preparing to test tutorial 1

Configuration prerequisites

Before trying any test based on this tutorial, complete the following tasks:

Create a sample RPG program on an IBM i system.

Compile the sample RPG program with the compiler parameter PGMINFO (*PCML) to generate a PCML file.

Extracting the sample files

Replicas of the artifacts that you create when using the external service wizard are provided as sample files for your reference. Use these files to verify that the files you create with the external service wizard are correct.

Note that the values for the i5/OS host name, user name, and password in the sample artifacts are from the IBM test lab. You need to change or set them appropriately for your environment.

Following table lists the artifacts that are shipped as part of samples which user can use to verify when calling an RPG program.

1) Tutorial1.zip – Project Interchange file which includes generated sample artifacts for calling an RPG program

<table>
<thead>
<tr>
<th>File/artifact name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>add2num</td>
<td></td>
</tr>
<tr>
<td>add2num/IBMImport.import</td>
<td>Contains the SCA import for the resource adapter.</td>
</tr>
<tr>
<td>add2num/ADD2NUM.xsd</td>
<td>Business object definition for the RPG program</td>
</tr>
<tr>
<td>add2num/IBMImport.wsdl</td>
<td>Contains the WSDL file configured for the resource adapter</td>
</tr>
</tbody>
</table>
2) ADD2NUM_SAVF_PCML.zip – Zip archive includes source of sample RPG program and generated PCML file

<table>
<thead>
<tr>
<th>File name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADD2NUMSAV.SAVF</td>
<td>SAVF file which has the source code for RPG sample program. Please refer your IBM i documentation for how to import this into IBM i system. Alternatively you can manually create a simple RPGLE to add two numbers like the one given below. The RPGLE source view is captured with the help of ‘IBM Rational Developer for System i’ tool.</td>
</tr>
<tr>
<td>ADD2NUM.PCML</td>
<td>PCML for sample RPG</td>
</tr>
</tbody>
</table>

Figure 1 - Sample RPG program as shown in ‘IBM Rational Developer for System i’ tool’s remote editor
Chapter 3. **Tutorial 1: Outbound Processing Calling the RPG program**

This tutorial demonstrates how WebSphere Adapter for IBM i V7.0 can be used to call an RPG program on an IBM i system.

**Scenario**

The user can call a RPG program on the IBM i system by using the outbound processing with the IBM i adapter. The scenario has the user application at one end and a RPG program on the IBM i system.

1) The user application needs to call a RPG program on the IBM i system and get the output of that program for further processing.

2) The adapter helps the user application to connect to the IBM i system.

3) The adapter calls the program without the need for the user application to know the syntax and steps to call the RPG program on the IBM i system.

4) The adapter converts the output of the program into the required format for the user application.

**Configuring the adapter for outbound processing**

Run the external service wizard to specify business objects, services, and the configuration to be used in this tutorial.
Creating the project

1. Launch WebSphere Integration Developer by clicking Start > Programs > IBM WebSphere > Integration Developer 7.0.

2. In WebSphere Integration Developer, switch to the Business Integration perspective by clicking Window > Open perspective > Other. In the Select perspective window, select Show all, then select Business Integration from the list and click OK.

3. Create a new module by clicking File > New > Module.

4. In the Module Name field, type add2num, and click Finish.

5. Open the Assembly Diagram of the module just created.
6. From the Outbound Adapters section of the Palette section, drag and drop iSeries from the list of adapters onto the Assembly Diagram editor. A window opens similar to one below.
7. Click IBM WebSphere Adapter for IBM i (IBM : 7.0.0) and click Next.

8. Select an appropriate run time from the Target runtime environment list. (WebSphere Process Server 7.0 is used as the run time in this tutorial.) Click Next.
Setting connection properties for the external service wizard

1. In the Discovery Configuration window, enter the IBM i server connection information such as the host name, user name, password, and path to the folder on the IBM i system for object discovery. From the **Object type to discover** drop-down list, select **RPG via PCML**.
2. After you have entered all properties, click **Next**.

**Generating business object definitions and related artifacts**

1. In the **Find Objects in the Enterprise System** window, in the Discovered objects pane, all objects are displayed
2. Type a query in the **Query** box and click **Run Query**. In the Discovered Objects pane, select the PCML you want to work with, click the arrow button and click **Next**.
3. In the Configuration Properties window, you can provide external commands to be executed before and after the program execution on IBM i. Click **Ok**.
4. In the **Specify Composite Properties** window, the operation CallPGM is listed in the **Operations for selected business objects** pane. Add and Remove operations are greyed out and not available to modify. Click **Next**.
5. Click Other radio button for the deployment Properties. The Host name is already populated with the value provided in the Connection Properties screen. Click Next.
6. A default name is provided for the interface. Click **Finish** to complete the configuration.
Deploying the module to the test environment

1. The result of running the external service wizard is an SCA module that contains an EIS import or export. Install this SCA module in the WebSphere Integration Developer integration test client. If WebSphere Process Server is not in the ‘Started’ state, you need to start the server.

2. Add the module you created earlier to the server using the server panel in WebSphere Integration Developer. Right-click on the server selection and then select Add and Remove Projects.
3. Add the SCA module to the server by moving them from left to right pane with the click of the button.

4. Click **Finish**.
Testing the assembled adapter application

1. Test the assembled adapter application by using the WebSphere Integration Developer integration test client.

2. Right-click inside the ‘IBMiImport’ component and select ‘Test Component’.
3. In the new editor window that opens, select `callpgmADD2NUM` from the **Operation** list, as shown in the following figure.
4. Type values for the input business objects and specify the required parameters to carry out the CallPGM operation on the RPG program. For example, you can specify 2 and 4 as values for GET1 & GET2 respectively as shown in the following figure.

6. Run the service by clicking the continue button and select the **Use this location as the default and do not ask again** check box and click **Finish**.
Figure 18

The value returned from the underlying RPG is populated in the same business object (GET3 attribute returns the value of GET1+GET2), as shown in the following figure.
The WebSphere® Adapter for IBM® i V7.0 exchanges business data between IBM i system and J2EE applications. The adapter retrieves information or request from and writes to the data queue and runs the RPG programs.

The document demonstrates two scenarios:

1. PutQueue and GetQueue outbound operations for data queues
2. Inbound operations for data queues

These scenarios demonstrate how WebSphere Adapter for IBM i V7.0 performs inbound and outbound operations. Everything you need to complete for each tutorial is contained in the tutorial. If you have performed the prerequisite tasks, you can complete each tutorial within an hour.

Learning objectives

After completing this tutorial, you should be able to perform the following tasks:

- Create an adapter project in WebSphere Integration Developer.
- Discover services and associated business objects from the enterprise information system (EIS) and make them part of the adapter project.
- Create a deployable module that you install on WebSphere Process Server or WebSphere Enterprise Service Bus.
- Test the module and validate the results.

Audience

These tutorials are for integration developers who design, assemble, test, and deploy business integration solutions.
Software prerequisites

To use these tutorials, you must have the following applications installed:

- WebSphere Integration Developer version 7.0
- WebSphere Process Server version 7.0
Chapter 5. Preparing to test tutorial 2

Configuration prerequisites

Before starting this tutorial, create a data queue on an IBM i system.

Extracting the sample files

The artifacts for Outbound, for example, DTAQOutbound.zip created using the external service wizard contain the files as listed in the following table:

Sample Artifacts here is Tutorial2.zip.

<table>
<thead>
<tr>
<th>Contents of Tutorial2.zip</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DTAQOutbound/IBMiOutboundInterface.import</td>
<td>Contains the SCA import for the resource adapter</td>
</tr>
<tr>
<td>DTAQOutbound/Fifoq.xsd</td>
<td>Business object definition for the Customer business function</td>
</tr>
<tr>
<td>DTAQOutbound/EmptyGetQueueBO.xsd</td>
<td>Business object definition for the business object container</td>
</tr>
<tr>
<td>DTAQOutbound/IBMiOutboundInterface.wsdl</td>
<td>Contains the WSDL file configured for the resource adapter</td>
</tr>
</tbody>
</table>
The artifacts for Inbound, for example, DTAQInbound.zip created using the external service wizard contain the files as listed in the following table:

<table>
<thead>
<tr>
<th>Contents of Tutorial2.zip</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DTAQInbound/IBMiInboundInterface.export</td>
<td>Contains the SCA export for the resource adapter</td>
</tr>
<tr>
<td>DTAQInbound/Fifoq.xsd</td>
<td>Business object definition for the Customer business function</td>
</tr>
<tr>
<td>DTAQInbound/IBMiInboundInterface.wsdl</td>
<td>Contains the WSDL file configured for the resource adapter</td>
</tr>
</tbody>
</table>
Chapter 6. **Tutorial 2: Outbound Processing – Writing the message to (PutQueue) and reading the message from (GetQueue) the data queue**

This tutorial demonstrates how you can use WebSphere Adapter for IBM i V7.0 to access messages using GetQueue (reading the message from data queue) and PutQueue (writing the message to data queue).

**Scenario**

PutQueue and GetQueue outbound operations for data queues: The user can retrieve and place messages (instructions) on queues. The queues can be first in first-out (FIFO) or last-in first-out (LIFO) data queues. The operations supported are GetQueue and PutQueue. The user can put messages to the queue using the PutQueue operation. The user can get message string from the queue using the GetQueue operation. The messages are then handled based on the type of queue.

**Steps involved for PutQueue operation**

1) The user application needs to run one or many commands (messages) on the IBM i system and get the output of these commands for further processing.

2) The user application can make a connection to the IBM i system through the adapter and send the messages (one or many) that have to be executed on the IBM i system.

3) The adapter stores these messages in a queue and sends it to the IBM i system for processing. The order in which the messages are handled is based on the type of queue.

4) The adapter stores the output of the messages (commands) from the IBM i system into the queue for the user application to pick up using the GetQueue.

**Steps involved for GetQueue operation**
1) The user has sent one or many instructions (commands) to the IBM i system for processing through the IBM i adapter.

2) The adapter makes a connection to the IBM i system and then sends these instructions (commands) stored in a queue in the order dictated by the type of queue.

3) The results of these instructions (commands) are then stored back in the queue from where the user application needs to use the GetQueue operation.

4) The user will get the results in the order based on the type of queue selected during the initial steps.

---

**Configuring the adapter for outbound processing**

Run the external service wizard to specify business objects, services, and the configuration to be used in this tutorial

**Creating the project**

1. Launch WebSphere Integration Developer by clicking **Start > Programs > IBM WebSphere > Integration Developer 7.0**.

2. Switch to the Business Integration perspective by clicking **Window > Open perspective > Other**. In the Select perspective window, select **Show all**, then select **Business Integration** from the list and click **OK**.

3. Create a new module by clicking **File > New > Module**.

4. In the **Module Name** field, type **DTAQOutbound**. Click **Finish**.
5. Accept the default settings and click **Finish**.

After the module is created, the folder structure will be as shown in the following figure.

6. Launch the external service wizard by right-clicking the module (DTAQOutbound) and selecting **New > external service**.
7. In the External Service window, click the + sign for the adapter to expand it and select iSeries. Click Next.
8. Select **IBM WebSphere Adapter for IBM i (IBM: 7.0.0.0)** and click **Next**.
Figure 6
9. Ensure that the ios RAR file is selected in the **Connector project** field, and in the **Target runtime** field select **WebSphere Process Server v7.0**. Click **Next**.

![Import a RAR File](image)

**Figure 7**

10. In the Processing Direction window, select **Outbound**. Click **Next**.
Figure 8

**Setting connection properties for the external service wizard**

1. Type the IBM i server connection information such as the host name, user name, password, and path to the folder on the IBM i system for object discovery. From the **Object type to discover** list select **Data Queue**.
2. After you have entered all properties, click **Next**.

**Generating business object definitions and related artifacts**

In the **Find Objects in the Enterprise System** window, in the Discovered objects pane, all data queues are displayed.
1. Type a query in the **Query** box and click **Run Query**. Select the data queue from the **Discovered objects** pane and move it to the **Selected objects** pane. Click **Next**.
2. In the **Specify Composite Properties** window, the operations PutQueue and GetQueue are listed in the **Operations for selected business objects** pane. Click **Next** button.
3. Click the **Other** radio button for the deployment properties. The **Host name** is already populated with the existing value. Click **Next**.
4. A default name is provided for the interface. Click **Finish** to complete the configuration.
6. Verify the results by checking the artifacts generated in business integration view for the module as shown in the following figure.
Deploying the module to the test environment

The result of running the external service wizard is an SCA module that contains an enterprise information system (EIS) import or export. Install this SCA module in the WebSphere Integration Developer integration test client.

1. Start WebSphere Process server.

2. Add the module you created earlier to the server by using the server panel in WebSphere Integration Developer. Right-click inside the server selection and then select **Add and Remove Projects**.

3. Select the DTAQOutbound module and click **Add** button, then click **Finish**.
Figure 17

Testing the assembled adapter application

1. Test the assembled adapter application by using the WebSphere Integration Developer integration test client.

2. Open the test component by right-clicking IBMiOutboundInterface then selecting Test Component.
3. Select putqueueFifoq from the operation.
4. Go to the Initial request parameters and enter the text ‘test message’ for messageContent request field.
6. Run the service by clicking the continue button and when the Select a Deployment Location window appears, select the WebSphere Process Server window to which you added the project and click the Finish button.

![Select a Deployment Location window](image)

Figure 22 Select a Deployment Location window

7. If security is set up, the User Login – Default Module Test window opens. Click OK.

![User Login – Default Module Test window](image)

Figure 23 User Login – Default Module Test window

8. The result will be displayed as show in the following screen.
9. To test the GetQueue operation on the data queue, click the button. This opens another test component. In the **Operation** field, select `getqueueFifo` and click .
The message from the data queue is displayed as return message.
WebSphere software
This tutorial demonstrates how you can use WebSphere Adapter for IBM i V7.0.0.0 to poll the data queue and send the messages to the endpoint application.

**Scenario**

The user can get messages that are polled from a queue and send them to the endpoint application. Here the data will be available from the queue based on the type of queue first-in-first-out (FIFO) or last-in-first-out (LIFO).

Steps involved in getting messages that are polled as follows:

1) The user requires to get messages from the IBM i system that have been polled by the adapter in a keyed data queue

2) The adapter will collect all the messages coming from the IBM i system and stores them in a data queue which can be a LIFO or FIFO.

3) The user application will get these messages in the order dictated by the type of queue.

**Configuring the adapter for inbound processing**

Run the external service wizard to specify business objects, services, and the configuration to be used in this tutorial.

**Creating the project**

1. Launch WebSphere Integration Developer by clicking Start > Programs > IBM WebSphere Integration Developer 7.0.
2. In WebSphere Integration Developer, switch to the Business Integration perspective by clicking Window > Open perspective > Other. In the Select perspective window, select Show all, then select Business Integration from the list and click OK.

3. Create a new module by clicking File > New > Module.

4. In the Module Name field, type DTAQInbound and click Finish.

5. After the module is created, the folder structure will look like that shown in the following figure.
6. Launch the external service wizard by right-clicking the module (DTAQOutbound) and selecting **New > external service**.

7. In the External Service window, click the + sign for the adapter to expand it and select **iSeries**. Click **Next**.
8. Click **IBM WebSphere Adapter for IBM i** and click **Next**.
9. Ensure that the ios RAR file is selected in the **Connector Project** field, and from the **Target runtime environment** list, select **WebSphere Process Server v7.0**. Click **Next**.
10. In the Select the Processing Direction window, select **Inbound** and click **Next**.
Setting connection properties for the external service wizard

1. Enter the IBM i server connection information such as the host name, user name, password and path to the folder on the IBM i for object discovery. The Object type to discover selection filed will be disabled.
2. After you enter all the properties click **Next**.

**Generating business object definitions and related artifacts**

On the metadata tree panel, all data queues will be displayed.
Figure 35

2. Type a query in the **Query** box and click **Run Query**. Select the data queue from the **Discovered objects pane**, and move it to the **Selected objects pane**. Click **Next**.
3. In the Specify Composite Properties window, the Emit operation is displayed in the **Operations for selected business objects** pane. Click **Next**.
4. Click the **Other** radio button for the Deployment properties. The **Host name** and **Poll Queue Path** fields will already be populated. Type the payload staging queue path and control language program path in the appropriate fields, and click Next.
Figure 38
Figure 39
Figure 40

5. A default name is provided for the interface. Click Finish to complete the configuration.
6. Verify the results by checking the artifacts generated in business integration view for the module as shown in following screen.

7. Drag and drop the Java Component from the Palette and draw a wire from IBMiInboundInterface to Component1.
8. Implement the java component with the logic for processing the event. This can be done by double clicking the Component1.

Figure 43

The following screen will be opened after double clicking the component1.

Deploying the module to the test environment
The result of running the external service wizard is an SCA module that contains an enterprise information system (EIS) import or export. Install this SCA module in the WebSphere Integration Developer integration test client.


2. Add the module you created earlier to the server using the server panel in WebSphere Integration Developer. Right-click the server and then select **Add and Remove Projects**.

![Figure 44](image)

2. Select the DTAQInbound module and click on add button, then click **Finish**.
Figure 45
Testing the assembled adapter application

Test the assembled adapter application using the WebSphere Integration Developer integration test client.

1. Right click the adapter module, **DTAQInbound** then select **Test > Attach**.
Figure 47

This will open the following screen.

Figure 48
2. Run the service by clicking the continue button which will deliver the event to the endpoint.

3. Verify this by checking for the endpoint messages in SystemOut file of WebSphere Process Server or by viewing the server console output in WebSphere Integration Developer, as shown in the following screen or WebSphere Process Server logs.

```
[9/18/08 17:14:56:130 IST] 0000005e logging I Start invoking EventSourceImpl constructor
[9/18/08 17:14:56:130 IST] 0000005e logging I Start invoking EventSourceImpl getEventPoint
[9/18/08 17:14:59:132 IST] 0000005e logging I Start invoking EventSourceImpl getEventPoint
[9/18/08 17:14:59:750 IST] 0000005e logging I Start invoking EventSourceImpl constructor
[9/18/08 17:14:59:750 IST] 0000005e logging I Start invoking EventSourceImpl getEventPoint
[9/18/08 17:35:01:625 IST] 00000061 SystemOut 0 >>> Exit Object = Test Message
```

Figure 49
Chapter 8. Introduction to Outbound/Inbound for Keyed Data Queues

WebSphere® Adapter for IBM® i V7.0 exchanges business data between system i and J2EE applications. The adapter retrieves data from and writes to the data queue and runs RPG programs.

The document demonstrates two scenarios:

1. PutQueue and GetQueue outbound operations for keyed data queues
2. Inbound operations for keyed data queues

These tutorials demonstrate how WebSphere Adapter for IBM i V7.0 performs inbound and outbound operations. To gain practical knowledge in setting up and deploying the adapter, complete one or more of the tutorials. Everything you need to complete each tutorial is contained in the tutorial. If you have performed the prerequisite tasks, you can complete each tutorial in less than an hour.

Learning objectives

After completing a tutorial, you should be able to perform the following tasks:

Create an adapter project in WebSphere Integration Developer.

Discover services and associated business objects from the enterprise information system (EIS) and make them part of the adapter project.

Create a deployable module that you install on WebSphere Process Server or WebSphere Enterprise Service Bus.

Test the module and validate the results.

Audience

These tutorials are for integration developers who design, assemble, test, and deploy business integration solutions.
Software Prerequisites

To use these tutorials, you must have the following applications installed:

WebSphere Integration Developer version 7.0

WebSphere Process Server version 7.0
Chapter 9. Preparing to test tutorial 3

Configuration prerequisites

Before doing any tutorial testing, complete the following task:

1. Create a keyed data queue on an IBM i system.

Extract the sample files

Replicas of the artifacts that you create when using the external service wizard are provided as sample files for your reference. Use these files to verify that the files you create with the external service wizard are correct.

Note that the values for the IBM i host name, user name, and password in the sample artifacts are from the IBM test lab. You need to change or set them appropriately for your environment.

Following table lists the artifacts that are shipped as part of samples which user can use to verify when handling Keyed Data Queue.

1. Tutorial3.zip - Project Interchange file which includes generated sample artifacts for handling messages on Keyed Data Queues.

<table>
<thead>
<tr>
<th>File name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>KDTAQOutbound</td>
<td>Contains the SCA import for the resource adapter</td>
</tr>
<tr>
<td>KDTAQOutbound/IBMiOutboundInterface.import</td>
<td>Business object definition for the Customer business function</td>
</tr>
<tr>
<td>KDTAQOutbound/Keyedq.xsd</td>
<td>Business object definition for the business object container</td>
</tr>
<tr>
<td>KDTAQOutbound/EmptyGetQueueBO.xsd</td>
<td>Contains the WSDL file configured for</td>
</tr>
<tr>
<td>KDTAQOutbound/IBMiOutboundInterface.wsdl</td>
<td></td>
</tr>
</tbody>
</table>
2) Tutorial3.zip - Project Interchange file which includes generated sample artifacts for polling messages on a Keyed Data Queue.

<table>
<thead>
<tr>
<th>File name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>KDTAQInbound</td>
<td>Contains the SCA export for the resource adapter</td>
</tr>
<tr>
<td>KDTAQInbound/IBMiInboundInterface.export</td>
<td>Contains the WSDL file configured for the resource adapter</td>
</tr>
<tr>
<td>KDTAQInbound/Keyedtaqkeyedq.xsd</td>
<td>Business object definition for the Customer business function</td>
</tr>
<tr>
<td>KDTAQInbound/IBMiInboundInterface.wsdl</td>
<td></td>
</tr>
</tbody>
</table>

the resource adapter
Chapter 10. Tutorial 3: Outbound Processing – Writing (PutQueue) the message to and reading (GetQueue) the message from the keyed data queue

This tutorial demonstrates how WebSphere Adapter for IBM i V7.0 can be used to put and get a string message from the keyed data queue.

Scenario

PutQueue and GetQueue outbound operations for keyed data queues: The user can retrieve and place data on the keyed data queues. The operations supported are Getqueue and Putqueue. The user can put messages to the queue using the putqueue operation. The user can get message string from the queue using the Getqueue operation.

Steps involved for PutQueue operation

1) The user application needs to run one or many commands (messages) on the IBM i system and get the output of these commands for further processing.

2) The user application can make a connection to the IBM i system through the adapter and send the messages (one or many) that have to be executed on the IBM i system.

3) The adapter stores these messages in the keyed data queue and sends it to the IBM i system for processing.

4) The adapter stores the output of the messages (commands) from the IBM i system into the queue for the user application to pick up using the Getqueue.

Steps involved for GetQueue operation
1) The user has sent one or many instructions (commands) to the IBM i system for processing through the IBM i adapter.

2) The adapter makes a connection to the IBM i system and then sends these instructions (commands) stored in the keyed data queue.

3) The results of these instructions (commands) are then stored back in the queue from where the user application needs to use the GetQueue operation.

---

**Configuring the adapter for outbound processing**

Run the external service wizard to specify business objects, services, and the configuration to be used in this tutorial

**Creating the project**

1. Launch WebSphere Integration Developer by selecting Start > Programs > IBM WebSphere > Integration Developer 7.0.

2. In WebSphere Integration Developer, switch to the Business Integration perspective. To do this, click **Window > Open perspective > Other**. In the Select perspective window, select **Show all**, then select **Business Integration** from the list and click **OK**.

3. Create a new module by selecting **File > New > Module**.

4. Type **KDTAQOutbound** in the **Module Name** field, and click **Finish**.
5. Launch the external service wizard by selecting File > New > Other > Business Integration > External Service.

6. Ensure that **Adapters** is selected and click **Next**.

7. Click IBM WebSphere Adapter for IBM i and click **Next**.
8. Ensure that **ios.rar** is selected and click **Next**.
Figure 4

9. Select **Outbound** and click **Next**.
Setting connection properties for the external service wizard

1. In the Discovery Configuration window, enter the IBM i server connection information such as the host name, user name, password, path to the folder on IBM i for object discovery and the object type to discover to keyed data queue.
2. After all properties are entered, click **Next**.

**Generating business object definitions and related artifacts**

1. On the **Find Objects in the Enterprise System** panel, click Run Query. All data queues will be displayed for the path to selected folder location.
2. Select the keyed data queue from the left Discovered object pane, click on [ ] and click Next.
3. In the Specify Composite Properties window, the operations PutQueue and GetQueue are listed in the **Operations for selected business objects** pane. Select the operations for the data queue by clicking the **Add** button. Click **Next**.
4. Select **Other** radio button for the deployment properties. The Host name is already populated with values provided by the user in the Connection Properties screen. Click Next.
5. A default name is provided for the interface. Click **Finish** to complete the configuration.
Figure 11

6. The generated module components are as shown in the following figure.
Deploying the module to the test environment

The result of running the external service wizard is an SCA module that contains an EIS import or export. Now you will install this SCA module in the WebSphere Integration Developer integration test client.

1. Add the module you created earlier to the server by using the server panel in WebSphere Integration Developer. Right-click the server, and then select **Add and Remove Projects**.

2. Add the SCA module to the server by selecting it and clicking Add. Then click **Finish**.
Figure 14
Figure 15
Testing the assembled adapter application

1. Test the assembled adapter application using the WebSphere Integration Developer integration test client.

2. Select the service you want, and click **Test > Test Module**.

3. Select the operation as **putqueueKeyedq** from the operation list.

4. Populate values for input business objects, by selecting the operation **putQueueKeyedq**.

5. Specify the required parameters (messageKey and messageContent) to perform the PutQueue operation on the Keyedq data queue.
6. Run the service by clicking the continue icon ( ). Select the **Use this as the default and do not ask again** check box and click **Finish**.
The output is as shown in the following figure.
7. To test the GetQueue operation on data queue, click the button. Another test component is opened. Select the `putqueueKeyedq` from the Operation list and click .
The message from the data queue is displayed as return message, as shown in the following figure.
Figure 21
Chapter 11. Tutorial 3 continued: Inbound processing on the keyed data queue

This tutorial demonstrates how WebSphere Adapter for IBM i V7.0.0.0 can be used to poll the keyed data queue and send the messages to the endpoint application.

Scenario

The user can get messages that are polled from a keyed data queue and send them to the endpoint application.

Steps involved for polling operation

1) The user application can get messages that are polled from a keyed data queue through the IBM i adapter.

2) The user application can make a connection to the IBM i system through the adapter and get messages from the IBM i system that have been polled to a keyed data queue.

3) The user needs to set a staging queue path so that messages that are polled from the data queue are stored in this path for sending to the end point application.

Configuring the adapter for inbound processing

Run the external service wizard to specify business objects, services, and the configuration to be used in this tutorial.
Creating the project

1. Launch WebSphere Integration Developer by clicking Start > Programs > IBM WebSphere > Integration Developer 7.0.

2. In WebSphere Integration Developer, switch to the Business Integration perspective by clicking Window > Open perspective > Other. In the Select perspective window, select Show all, then select Business Integration from the list and click OK.

3. Create a new module by clicking File > New > Module.

4. Type KDTAQInbound in the Module Name field and click Finish.

Figure 22

The following window shows the components generated.

Figure 23
5. Launch the external service wizard by selecting File > New > External Service.

6. Ensure that **Adapters** is selected, select **iSeries** and click **Next**.

![New External Service](image)

Figure 24

7. Click IBM WebSphere Adapter for IBM i (IBM : 7.0) and click Next.
8. Ensure that the ios RAR file is selected and click **Next**.
9. Select **Inbound** and click **Next**.
Figure 27

**Setting connection properties for the external service wizard**

1. Enter the IBM i server connection information such as the host name, user name, password and path to the folder on the IBM i system for object discovery. The **Object type to discover** field will be disabled.
2. After you have entered all properties, click Next.

**Generating business object definitions and related artifacts**

1. In the Discovered objects pane, click Run Query and all data queues will be displayed so that you can select the path to the folder.
2. Select the keyed data queue from the **Discovered objects** pane and click the button to move it to the **Selected objects** pane. Click Next.
3. In the Specify Composite Properties window, the Emit operation is displayed and is disabled. Click **Next**.
Figure 30
4. Click **Other** for the deployment properties. The **Host name** and **Poll queue path** fields will be populated in this window. Type in the **Payload staging queue path** field, **Polling control language program path**, **Failed events queue path**, **Failed event archiving control language program path** and **Key for keyed data queue** fields and click **Next**.

![Service Generation and Deployment Configuration](image)

Figure 31

The advanced properties are shown below.
**Service Generation and Deployment Configuration**

Specify properties for generating the service and running it on the server.

### Event polling configuration

- **Interval between polling periods (milliseconds):** 2000
- **Maximum events in polling period:** 10
- **Time between retries in case of system connection failure (in milliseconds):** 60000
- **Maximum number of retries in case of system connection failure:** 0
- **Stop the adapter when an error is encountered while polling**
- **Retry EIS connection on startup**

### Event delivery configuration

- **Type of delivery:** ORDERED
- **Ensure assured-once event delivery (may reduce performance)**
- **Do not process events that have a time stamp in the future**
- **Event types to process:**

### Event persistence properties

- **Auto create event table**
- **Event recovery table name:** ISERIES_EVENTSTORE
- **Event recovery data source (JNDI) name:**
- **User name:**
- **Password:**
- **Database schema name:**

### Logging and tracing

- **Adapter ID:** 001
- **Disguise user data as “XXX” in log and trace files.”**

---

**Figure 32**
5. A default name is provided for the interface. Click **Finish** to complete the configuration.

![New External Service](image)

**Figure 33**

6. The generated module contains the following components in it.

![Assembly Diagram](image)

**Figure 34**
7. Generate a Java™ component and draw a wire from **IBMInboundInterface** to **Component1**.

Figure 35

8. Generate implementation for the java component interface as shown below. You can choose “default package” when asked for package for the implementation.

Figure 36
The generated Java implementation is shown in the following screen capture.

```java
import commonj.sdo.DataObject;

public class Component1Impl {
    /**
     * Default constructor.
     */
    public Component1Impl() {
        super();
    }

    /**
     * Return a reference to the component service instance for this class. This method should be used when passing this service or if you want to invoke this component service asynchronously.
     *
     * @generated (com.ibm.wbi.java)
     */
    @SuppressWarnings("unused")
    private Object getMyService() {
        return (Object) ServiceManager.INSTANCE.locateService("sel"
Deploying the module to the test environment

The result of running the external service wizard is an SCA module that contains an EIS import or export. Install this SCA module in the WebSphere Integration Developer integration test client.


2. Add the module you created earlier to the server by using the server panel in WebSphere Integration Developer. Right-click the server, and then select Add and Remove Projects.

2. Add the SCA module from the left frame to the right frame by selecting it and clicking Add then click Finish.
Figure 40
Testing the assembled adapter application

Test the assembled adapter application by using the WebSphere Integration Developer integration test client.

1. Right-click the adapter module, **KDTAQInbound**, then select **Test > Attach**.

   The relevant business object is delivered to the endpoint.

2. Verify that the business object has been delivered by either checking for the endpoint messages in the SystemOut file of WebSphere Process Server or by viewing the server console output in WebSphere Integration Developer.
This will open the following screen.

Integration Test Client: KDTAQInbound_Test
2. Run the service by clicking the continue button which will deliver the event to the endpoint.

3. Verify this by checking for the endpoint messages in SystemOut file of WebSphere Process Server or by viewing the server console output in WebSphere Integration Developer, as shown in the following screen or WebSphere Process Server logs.

<table>
<thead>
<tr>
<th>Date/Time</th>
<th>Event Source Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>[9/18/08 17:34:56]</td>
<td>I Start invoking EventSourceImpl constructor</td>
</tr>
<tr>
<td>[9/18/08 17:34:56]</td>
<td>I Start invoking EventSourceImpl getEventPoint</td>
</tr>
<tr>
<td>[9/18/08 17:34:56]</td>
<td>I Start invoking EventSourceContextImpl.getEventSource</td>
</tr>
<tr>
<td>[9/18/08 17:34:56]</td>
<td>I Start invoking EventSourceImpl constructor</td>
</tr>
<tr>
<td>[9/18/08 17:34:56]</td>
<td>I Start invoking EventSourceImpl getEventPoint</td>
</tr>
<tr>
<td>[9/18/08 17:34:56]</td>
<td>I Start invoking EventSourceImpl constructor</td>
</tr>
<tr>
<td>[9/18/08 17:34:56]</td>
<td>I Start invoking EventSourceImpl constructor</td>
</tr>
<tr>
<td>[9/18/08 17:34:56]</td>
<td>I Start invoking EventSourceImpl getEventPoint</td>
</tr>
<tr>
<td>[9/18/08 17:34:56]</td>
<td>I Start invoking EventSourceImpl constructor</td>
</tr>
<tr>
<td>[9/18/08 17:34:56]</td>
<td>I Start invoking EventSourceImpl getEventPoint</td>
</tr>
<tr>
<td>[9/18/08 17:34:56]</td>
<td>I Start invoking EventSourceImpl constructor</td>
</tr>
<tr>
<td>[9/18/08 17:34:56]</td>
<td>I Start invoking EventSourceImpl getEventPoint</td>
</tr>
<tr>
<td>[9/18/08 17:34:56]</td>
<td>I Start invoking EventSourceImpl constructor</td>
</tr>
<tr>
<td>[9/18/08 17:34:56]</td>
<td>I Start invoking EventSourceImpl getEventPoint</td>
</tr>
<tr>
<td>[9/18/08 17:34:56]</td>
<td>I Start invoking EventSourceImpl constructor</td>
</tr>
<tr>
<td>[9/18/08 17:34:56]</td>
<td>I Start invoking EventSourceImpl getEventPoint</td>
</tr>
<tr>
<td>[9/18/08 17:34:56]</td>
<td>I Start invoking EventSourceImpl constructor</td>
</tr>
<tr>
<td>[9/18/08 17:34:56]</td>
<td>I Start invoking EventSourceImpl getEventPoint</td>
</tr>
</tbody>
</table>

Figure 44
Chapter 12. **Introduction to Outbound for Keyed Data Queues with support for binary content**

WebSphere® Adapter for IBM® i V7.0 exchanges business data between system i and J2EE applications. The adapter retrieves data from and writes to the data queue and also runs RPG, COBOL and Service programs.

The document demonstrates two scenarios:

1. PutQueue and GetQueue outbound operations for keyed data queues with binary content

These tutorials demonstrate how WebSphere Adapter for IBM i V7.0 performs inbound and outbound operations. To gain practical knowledge in setting up and deploying the adapter, complete one or more of the tutorials. Everything you need to complete each tutorial is contained in the tutorial. If you have performed the prerequisite tasks, you can complete each tutorial in less than an hour.

---

**Learning objectives**

After completing a tutorial, you should be able to perform the following tasks:

Create an adapter project in WebSphere Integration Developer.

Discover services and associated business objects from the enterprise information system (EIS) and make them part of the adapter project.

Create a deployable module that you install on WebSphere Process Server or WebSphere Enterprise Service Bus.

Test the module and validate the results.

---

**Audience**

These tutorials are for integration developers who design, assemble, test, and deploy business integration solutions.
Software Prerequisites

To use these tutorials, you must have the following applications installed:

WebSphere Integration Developer version 7.0

WebSphere Process Server version 7.0
Chapter 13. Preparing to test tutorial 4

Configuration prerequisites

Before doing any tutorial testing, complete the following task:

1. Create a keyed data queue on an IBM i system.

Extract the sample files

Replicas of the artifacts that you create when using the external service wizard are provided as sample files for your reference. Use these files to verify that the files you create with the external service wizard are correct.

Note that the values for the i5/OS host name, user name, and password in the sample artifacts are from the IBM test lab. You need to change or set them appropriately for your environment.

Following table lists the artifacts that are shipped as part of samples which user can use to verify when handling Keyed Data Queue.

1) Tutorial4.zip - Project Interchange file which includes generated sample artifacts for handling messages on Keyed Data Queues.

<table>
<thead>
<tr>
<th>File name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>KeyedDataQueueBinary</td>
<td></td>
</tr>
<tr>
<td>KeyedDataQueueBinary/IBMiOutboundInterface.import</td>
<td>Contains the SCA import for the resource adapter.</td>
</tr>
<tr>
<td>KeyedDataQueueBinary/Keyedq.xsd</td>
<td>Business object definition for the Customer business function.</td>
</tr>
<tr>
<td>KeyedDataQueueBinary/IBMiOutboundInterface.wsdl</td>
<td>Contains the WSDL file configured for the resource adapter</td>
</tr>
</tbody>
</table>
Chapter 14. Tutorial 4: Outbound Processing – Binary Content Support for writing the message to (PutQueue) and reading the message from (GetQueue) the keyed data queue

This tutorial demonstrates how WebSphere Adapter for IBM i V7.0 can be used to put and get a string message from the keyed data queue with the messages in binary format.

Scenario

PutQueue and GetQueue outbound operations for keyed data queues: The user can retrieve and place data on the keyed data queues. The operations supported are Getqueue and Putqueue. The user can put messages to the queue using the Putqueue operation. The user can get message string from the queue using the Getqueue operation. The message can be in binary format in this case.

Steps involved for PutQueue operation

1) The user application needs to run one or many commands (messages) on the IBM i system and get the output of these commands for further processing.

2) The user application can make a connection to the IBM i system through the adapter and send the messages (one or many) that have to be executed on the IBM i system.

3) The adapter stores these messages in the keyed data queue in binary format and sends it to the IBM i system for processing.

4) The adapter stores the output of the messages (commands) from the IBM i system into the queue for the user application to pick up using the Getqueue.

Steps involved for GetQueue operation
1) The user has sent one or many instructions (commands) to the IBM i system for processing through the IBM i adapter.

2) The adapter makes a connection to the IBM i system and then sends these instructions (commands) stored in the keyed data queue.

3) The results of these instructions (commands) are then stored back in the queue from where the user application needs to use the GetQueue operation.

---

**Configuring the adapter for outbound processing**

Run the external service wizard to specify business objects, services, and the configuration to be used in this tutorial

**Creating the project**

1. Launch WebSphere Integration Developer by selecting Start > Programs > IBM WebSphere > Integration Developer 7.0.

2. In WebSphere Integration Developer, switch to the Business Integration perspective. To do this, click Window > Open perspective > Other. In the Select perspective window, select Show all, then select Business Integration from the list and click OK.

3. Create a new module by selecting File > New > Module.

4. Type KeyedDataQueueBinary in the Module Name field, and click Finish.
7. Go to menu New->External Service and click **IBM WebSphere Adapter for IBM i (IBM: 7.0.0.0)** and click **Next**. Select an appropriate run time from the **Target runtime** list. (WebSphere Process Server 7.0 is used as the run time in this tutorial.) Click **Next**.
Figure 5

**Setting connection properties for the external service wizard**

1. In the Discovery Configuration window, enter the IBM i server connection information such as the host name, user name, password, and path to the folder on the IBM i system for object discovery. From the **Object type to discover** list, select Data Queue.
9. Select **Outbound** and click **Next**.
Generating business object definitions and related artifacts

1. On the Find Objects in the Enterprise System panel, click Run Query and all data queues will be displayed for the path to selected folder location.
2. Select the keyed data queue from the left Discovered object pane, click on [►] and click **Next**.
3. In the Specify Composite Properties window, the operations PutQueue and GetQueue are listed in the **Operations for selected business objects** pane. Select the operations for the data queue by clicking the **Add** button. Select the check box at the bottom of the page for setting messages in binary format. Click **Next**.
4. Select **Other** radio button for the deployment properties. The Host name is already populated. Click Next.
5. A default name is provided for the interface. Click **Finish** to complete the configuration.
6. The generated module components is as shown in the following figure.
Deploying the module to the test environment

The result of running the external service wizard is an SCA module that contains an EIS import or export. Now you will install this SCA module in the WebSphere Integration Developer integration test client.

1. Add the module you created earlier to the server by using the server panel in WebSphere Integration Developer. Right-click the server, and then select **Add and Remove Projects**.

   ![Add and Remove Projects](image)

   **Figure 13**

2. Add the SCA module to the server by selecting it and clicking Add. Click **Finish**.
Figure 14
Figure 15
Testing the assembled adapter application

1. Test the assembled adapter application using the WebSphere Integration Developer integration test client.

2. Select the service you want, and click **Test > Test Module**.

3. Select the operation as **putqueueKeyedq** from the operation list.

4. Populate values for input business objects, by selecting the operation **putqueueKeyedq**.

5. Specify the required parameters **messageKeyAsBinary** and **messageContentAsBinary** to perform the PutQueue operation on the Keyedq data queue. Note that the message is in binary format.
6. Run the service by clicking the continue icon ( ). Select the **Use this as the default and do not ask again** check box and click **Finish**.
7. To test the GetQueue operation on data queue click the button. Another test component is opened. Select the putqueueKeyedq from the Operation list and click 🔄.
The message from the data queue is displayed as return message, as shown in the following figure.

Figure 21
Chapter 15. Introduction to WebSphere Adapter for IBM i - Calling a Service program

WebSphere® Adapter for IBM® i V7.0 exchanges business data between system i and J2EE applications. The adapter retrieves from and writes to the data queue and runs Service programs.

The document demonstrates the following scenario:

1. Calling a Service program with the help of Adapter for IBM i

This scenario demonstrates how WebSphere Adapter for IBM i V7.0 performs outbound operations.

Learning objectives

After completing this tutorial, you should be able to perform the following tasks:

Create an adapter project in WebSphere Integration Developer.

Discover services and associated business objects from the enterprise information system (EIS) and make them part of the adapter project.

Create a deployable module that you install on WebSphere Process Server or WebSphere Enterprise Service Bus.

Test the module and validate the results.

Audience

This tutorial is for integration developers who design, assemble, test, and deploy business integration solutions.

Software prerequisites

To use this tutorial, you must have the following applications installed:

WebSphere Integration Developer version 7.0

WebSphere Process Server version 7.0
Chapter 16. Preparing to test tutorial 5

Configuration prerequisites

Before trying any test based on this tutorial, complete the following tasks:

Create a sample Service program on an IBM i system.

Compile the sample Service program with the compiler parameter PGMINFO (*PCML) to generate a PCML file.

Extracting the sample files

Replicas of the artifacts that you create when using the external service wizard are provided as sample files for your reference. Use these files to verify that the files you create with the external service wizard are correct.

Note that the values for the i5/OS host name, user name, and password in the sample artifacts are from the IBM test lab. You need to change or set them appropriately for your environment.

Following table lists the artifacts that are shipped as part of samples which user can use to verify when running a Service program.

1) Tutorial5.zip – Project Interchange file which includes generated sample artifacts for running a Service program

<table>
<thead>
<tr>
<th>File/artifact name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ServiceProgram</td>
<td>Contains the SCA import for the resource adapter</td>
</tr>
<tr>
<td>ServiceProgram/IBMOutboundInterface.import</td>
<td>Contains the WSDL file configured for the resource adapter</td>
</tr>
<tr>
<td>ServiceProgram/ADD2NUMSRVADD2NUM.xsd</td>
<td>Business object definition for the RPG program</td>
</tr>
<tr>
<td>ServiceProgram/IBMOutboundInterface.wsdl</td>
<td>Contains the WSDL file configured for the resource adapter</td>
</tr>
</tbody>
</table>

2) ADD2NUMSRV_PCML.zip – Zip archive includes source of sample Service program and generated PCML file
ADD2NUM.RPGLE  RPGLE file which has the source code for Service Program sample program. For more details on how to import the file into the IBM i system, refer to the IBM i documentation. Alternatively you can manually create a simple RPGLE to add two numbers like the one given below. The RPGLE source view is captured with the help of ‘IBM Rational Developer for System i’ tool.

ADD2NUMSRV.PCML  PCML for sample RPG is as indicated in the following figure

Figure 1 - Sample Service program as shown in ‘IBM Rational Developer for System i’ tool's remote editor
Chapter 17. Tutorial 5: Outbound Processing – Calling the Service program

This tutorial demonstrates how WebSphere Adapter for IBM i V7.0 can be used to call a Service program on an IBM i system.

Scenario

The user can call a Service program on the IBM i system by using the outbound processing with the IBM i adapter. The scenario has the user application at one end and a Service program on the IBM i system.

1) The user application needs to run a Service program on the IBM i system and get the output of that program for further processing.

2) The adapter helps the user application to connect to the IBM i system.

3) The adapter calls the program without the need for the user application to know the syntax and steps to run the Service program on the IBM i system.

4) The adapter converts the output of the program into the required format for the user application.

Configuring the adapter for outbound processing

Run the external service wizard to specify business objects, services, and the configuration to be used in this tutorial

Creating the project
1. Launch WebSphere Integration Developer by clicking Start > Programs > IBM WebSphere > Integration Developer 7.0.

2. In WebSphere Integration Developer, switch to the Business Integration perspective by clicking Window > Open perspective > Other. In the Select perspective window, select Show all, then select Business Integration from the list and click OK.

3. Create a new module by clicking File > New > Module.

4. In the Module Name field, type ServiceProgram, and click Finish.

5. If it is not already open, open the Assembly Diagram of the module just created, by double clicking it.
6. From the Outbound Adapters section of the Palette section, drag and drop iSeries from the list of adapters onto the Assembly Diagram editor as shown in the following figure.
7. Click **IBM WebSphere Adapter for IBM i (IBM: 7.0.0.0)** and click **Next**.

8. Select an appropriate run time from the **Target runtime** list. (WebSphere Process Server 7.0 is used as the run time in this tutorial.) Click **Next**.
Figure 5

**Setting connection properties for the external service wizard**

1. In the Discovery Configuration window, enter the IBM i server connection information such as the host name, user name, password, and path to the folder on the IBM i system for object discovery. From the **Object type to discover** list, select **SRVPGM via PCML**.
2. After you have entered all properties, click Next.

**Generating business object definitions and related artifacts**

1. In the Find Objects in the Enterprise System window, in the Discovered objects pane, all objects are displayed
2. Type a query in the **Query** box and click **Run Query**. In the Discovered Objects pane, select the PCML you want to work with, select the button and click **Next**.
3. In the Configuration Properties window, we can provide external commands to be executed before and after the program execution on IBM i. Click **OK**.
Figure 9

The PCML will be selected as shown in the following figure
4. In the **Specify Composite Properties** window, the operation CallPGM is listed in the **Operations for selected business objects** pane. Because this is the only operation supported for calling Service programs, you cannot use the **Add** and **Remove** options. Click **Next**.
5. Click **Other** radio button for the deployment Properties. The **Host name** is already populated. Click **Next**.
6. A default name is provided for the interface. Click **Finish** to complete the configuration.
Deploying the module to the test environment

1. The result of running the external service wizard is an SCA module that contains an EIS import or export. Install this SCA module in the WebSphere Integration Developer integration test client. If WebSphere Process Server is not in ‘Started’ state, start the server.

2. Add the module you created earlier to the server using the server panel in WebSphere Integration Developer. Right-click the server, and then select Add and remove projects.
3. Add the SCA module to the server by moving them from left to right pane with the click of ‘Add >’ button.

4. Click **Finish**.
1. Test the assembled adapter application by using the WebSphere Integration Developer integration test client.

2. Right-click the ‘IBMiOutboundInterface’ component and select ‘Test Component’.

3. In the new editor window that opens, select `callpgmADD2NUM` from the `Operation` list, as shown in the following figure.
4. Type values for the input business objects and specify the required parameters to carry out the CallPGM operation on the Service program. For example you can specify 2 and 4 as values for GET1 & GET2 respectively as shown below.
6. Run the service by clicking the continue button and select the **Use this as the default and do not ask again** check box and click **Finish**.

![Deployment Location](image)

**Figure 17**

The value returned from the underlying Service program is populated in the same business object (GET3 attribute returns the value of GET1+GET2), as shown in the following figure.

**Figure 18**
Figure 19
Chapter 18. Introduction to WebSphere Adapter for IBM i - Calling a COBOL program

WebSphere® Adapter for IBM® i V7.0 exchanges business data between system i and J2EE applications. The adapter retrieves from and writes to the data queue and runs a COBOL program.

The document demonstrates the following scenario:

1. Calling a COBOL program with the help of Adapter for IBM i

This scenario demonstrates how WebSphere Adapter for IBM i V7.0 performs outbound operations.

Learning objectives

After completing this tutorial, you should be able to perform the following tasks:

Create an adapter project in WebSphere Integration Developer.

Discover services and associated business objects from the enterprise information system (EIS) and make them part of the adapter project.

Create a deployable module that you install on WebSphere Process Server or WebSphere Enterprise Service Bus.

Test the module and validate the results.

Audience

This tutorial is for integration developers who design, assemble, test, and deploy business integration solutions.

Software prerequisites

To use this tutorial, you must have the following applications installed:

WebSphere Integration Developer version 7.0

WebSphere Process Server version 7.0
Chapter 19. Preparing to test tutorial 6

Configuration prerequisites

Before trying any test based on this tutorial, complete the following tasks:

Create a sample COBOL program on an IBM i system.

Compile the sample COBOL program with the compiler parameter PGMINFO (*PCML) to generate a PCML file.

Extracting the sample files

Replicas of the artifacts that you create when using the external service wizard are provided as sample files for your reference. Use these files to verify that the files you create with the external service wizard are correct.

Note that the values for the i5/OS host name, user name, and password in the sample artifacts are from the IBM test lab. You need to change or set them appropriately for your environment.

Following table lists the artifacts that are shipped as part of samples which user can use to verify when running a Service program.

1) Tutorial6.zip – Project Interchange file which includes generated sample artifacts for calling a COBOL program

<table>
<thead>
<tr>
<th>File/artifact name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CobolProgram</td>
<td></td>
</tr>
<tr>
<td>CobolProgram/IBMOutboundInterface.import</td>
<td>Contains the SCA import for the resource adapter</td>
</tr>
<tr>
<td>CobolProgram/ADDDNUMBERSADDDNUMBERS.xsd</td>
<td>Business object definition for the COBOL program</td>
</tr>
<tr>
<td>CobolProgram/IBMOutboundInterface.wsdl</td>
<td>Contains the WSDL file configured for the resource adapter</td>
</tr>
</tbody>
</table>

2) ADD2NUMCobol_PCML.zip – Zip archive includes source of sample COBOL program and generated PCML file
<table>
<thead>
<tr>
<th>File name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADD2NUMSAV.SAVF</td>
<td>SAVF file which has the source code for the COBOL sample program. Please refer your IBM i documentation for how to import this into IBM i system. Alternatively, you can manually create a simple RPGLE to add two numbers like the one as shown in the following figure. The RPGLE source view is captured with the help of ‘IBM Rational Developer for System i’ tool.</td>
</tr>
<tr>
<td>ADD2NUM.PCML</td>
<td>PCML for sample COBOL</td>
</tr>
</tbody>
</table>

```xml
<pcml version="4.0">
  <!-- COBOL program: ADDNUMBERS -->
  <!-- created: 07/13/09 14:50:25 -->
  <!-- source: RAJAN.LIB/CELESRC(ADDNUMBERS) -->
  <program name="ADDNUMBERS" path="/QSYS.LIB/RAJAN.LIB/ADDNUMBERS.PGM">
    <data name="INPNUN1" type="char" length="4" usage="inputoutput" />
    <data name="INPNUN2" type="char" length="4" usage="inputoutput" />
    <data name="OUTNUM" type="char" length="4" usage="inputoutput" />
  </program>
</pcml>
```

Figure 1
Chapter 20. Tutorial 6: Outbound Processing – Calling the COBOL program

This tutorial demonstrates how WebSphere Adapter for IBM i V7.0 can be used to call a COBOL program on an IBM i system.

Scenario

The user can run a COBOL program on the IBM i system by using the outbound processing with the IBM i adapter. The scenario has the user application at one end and a COBOL program on the IBM i system.

1) The user application needs to run a COBOL program on the IBM i system and get the output of that program for further processing.

2) The adapter helps the user application to connect to the IBM i system.

3) The adapter calls the program without the need for the user application to know the syntax and steps to run the Service program on the IBM i system.

4) The adapter converts the output of the program into the required format for the user application.

Configuring the adapter for outbound processing

Run the external service wizard to specify business objects, services, and the configuration to be used in this tutorial

Creating the project
1. Launch WebSphere Integration Developer by clicking Start > Programs > IBM WebSphere > Integration Developer 7.0.

2. In WebSphere Integration Developer, switch to the Business Integration perspective by clicking Window > Open perspective > Other. In the Select perspective window, select Show all, then select Business Integration from the list and click OK.

3. Create a new module by clicking File > New > Module.

4. In the Module Name field, type CobolProgram, and click Finish.

5. If it is not already open, open the Assembly Diagram of the module just created, by double clicking it.
6. From the Outbound Adapters section of the Palette section, drag and drop **iSeries** onto the Assembly Diagram editor as shown in the following figure.
7. Click **IBM WebSphere Adapter for IBM i (IBM: 7.0.0.0)** and click **Next**.

8. Select an appropriate run time from the **Target runtime** environment list. (WebSphere Process Server 7.0 is used as the run time in this tutorial). Click **Next**.

Figure 4
Figure 5

**Setting connection properties for the external service wizard**

1. In the Discovery Configuration window, enter the IBM i server connection information such as the host name, user name, password, and path to the folder on the IBM i system for object discovery. From the **Object type to discover** list, select **Cobol via PCML**.
2. After you have entered all the properties, Click Next.

**Generating business object definitions and related artifacts**

1. In the Find Objects in the Enterprise System window, in the Discovered objects pane, all objects are displayed
2. Type a query in the **Query** box and click **Run Query**. In the Discovered Objects pane, select the PCML you want to work with and select and click **Next**.
3. In the Configuration Properties window, user can provide external commands to be executed before and after the program execution on IBM i. Click **OK**.
Figure 9

The PCML is selected as shown in the figure below
4. In the Configure Composite Properties window, the operation CallPGM is listed in the **Operations for selected business objects** pane. Because this is the only operation supported for calling COBOL programs, you cannot use the **Add** and **Remove** options. Click **Next**.
5. Click **Other** radio button for the deployment Properties. The **Host name** is already populated. Click **Next**.
6. A default name is provided for the interface. Click **Finish** to complete the configuration.
Deploying the module to the test environment

1. The result of running the external service wizard is an SCA module that contains an EIS import or export. Install this SCA module in the WebSphere Integration Developer integration test client. If WebSphere Process Server is not in ‘Started’ state, start the server.

2. Add the module you created earlier to the server using the server panel in WebSphere Integration Developer. Right-click the server and then select Add and Remove Projects.
3. Add the SCA module to the server by moving them from left to right pane with the click of ‘Add >’ button.

4. Click **Finish**.
Testing the assembled adapter application

1. Test the assembled adapter application by using the WebSphere Integration Developer integration test client.

2. Right-click the ‘IBMiOutboundInterface’ component and select ‘Test Component’.

3. In the new editor window that opens, select `callpgmADD2NUM` from the Operation list, as shown in the following figure.
4. Type values for the input business objects and specify the required parameters to carry out the CallPGM operation on the COBOL program. For example, you can specify 2 and 4 as values for GET1 & GET2 respectively as shown in the following figure.

6. Run the service by clicking the continue button and select the **Use this as the default and do not ask again** check box and click **Finish**.
The value returned from the underlying COBOL is populated in the same business object (for example, GET3 attribute returns the value of GET1+GET2), as shown in the following figure.
WebSphere software
Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user’s responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication.
IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites.

The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation Department
2Z4A/SOM1 294 Route 100
Somers, NY 10589-0100 U.S.A.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

**Programming interface information**

Programming interface information, if provided, is intended to help you create application software using this program.

General-use programming interfaces allow you to write application software that obtains the services of this program's tools.
However, this information may also contain diagnosis, modification, and tuning information. Diagnosis, modification and tuning information is provided to help you debug your application software.

Warning:

Do not use this diagnosis, modification, and tuning information as a programming interface because it is subject to change.

**Trademarks**

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published.

Such trademarks may also be registered or common law trademarks in other countries. A complete and current list of IBM trademarks is available on the Web at [http://www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml)

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft and Screens are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.*

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.

This product includes software developed by the Eclipse Project ([http://www.eclipse.org](http://www.eclipse.org)).