IBM WebSphere Business Connection

# Using the Web Services Gateway

*Version 1.1.1*

**Note!**
Before using this information and the product it supports, be sure to read the general information under "Notices" on page 107.

**Second Edition (December 2002)**

This edition applies to Version 1, Release 1, Modification 1, of *IBM® WebSphere® Business Connection* (5724-D26) and to all subsequent releases and modifications until otherwise indicated in new editions.

IBM welcomes your comments. You can send them to the following address:

IBM Canada Ltd. Laboratory
Information Development
8200 Warden Avenue
Markham, Ontario, Canada L6G 1C7

Include the title and order number of this book, and the page number or topic related to your comment.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

**© Copyright International Business Machines Corporation 2002. All rights reserved.**
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

# Contents


**Chapter 1. Using the Web Services Gateway . . . . . 1**

**Chapter 2. Web Services Gateway - Frequently Asked Questions . . . . . 3**

**Chapter 3. Web Services Gateway - What is new in this release . . . . . 5**

**Chapter 4. Web Services Gateway - Completing the installation . . . . . 7**
Web Services Gateway - prerequisites and constraints 8
Testing the Web Services Gateway installation . . . 9
Installing other Web Services Gateway applications 10

**Chapter 5. Administering the Web Services Gateway . . . . . 13**
Setting the namespace URI and WSDL URI for the Web Services Gateway. . . . . 14
Working with channels . . . . . 16
Channels - entry points to the Web Services Gateway . . . . . 17
Listing and managing gateway-deployed channels . . . . . 17
Deploying channels to the Web Services Gateway 18
Removing channels from the Web Services Gateway . . . . . 21
Working with filters . . . . . 22
Filters - service interceptors for the Web Services Gateway . . . . . 23
The WebSphere Business Connection routing filters . . . . . 23
The Simple Log filter . . . . . 24
Listing and managing gateway-deployed filters 25
Deploying filters to the Web Services Gateway . 26
Removing filters from the Web Services Gateway 27
Writing a filter for the Web Services Gateway . . 27
Working with UDDI references . . . . . 35
Listing and managing gateway-deployed UDDI references . . . . . 36
Deploying UDDI references to the Web Services Gateway . . . . . 37
Removing UDDI references from the Web Services Gateway . . . . . 38
Working with Web services . . . . . 39
Listing and managing gateway-deployed Web services. . . . . 40
Deploying Web services to the Web Services Gateway . . . . . 41
Removing Web services from the Web Services Gateway . . . . . 44

**Chapter 6. Running the Web Services Gateway samples . . . . . 47**
Running the Stock Quote service sample . . . . 47
Running the Address Book service sample . . . . 49
Running the Calendar service sample. . . . . 50

**Chapter 7. Administering security for the Web Services Gateway . . . . . 53**
Web service security - role-based authorization . . 54

**Chapter 8. Transferring large files from gateway to gateway . . . . . 55**
Large file transfer - large files copied reliably from gateway to gateway . . . . . 55
Using large file transfer with the LFT sample . . . 56
LFT sample: checking LFT files in and out locally (server side) . . . . . 58
LFT sample: checking LFT files in and out remotely (from client to server). . . . . 59
Setting up secure large file transfer . . . . . 61
Secure LFT: Setting up HTTP servers for SSL . . 62
Secure LFT: Creating a client TrustStore and importing ServerOne's certificate . . . . . 67
Secure LFT: Creating a wsgw TrustStore and importing ServerTwo's certificate . . . . . 68
Secure LFT: Configuring the application servers 70
Secure LFT: Installing IBM Java Secure Socket Extension (JSSE) 1.0.2 . . . . . 71
Secure LFT: Creating and enabling httpr.ini. . . 72
Secure LFT: Checking files in remotely (from client to server) . . . . . 73
Large File Transfer - how the LFT sample works . . 74
Querying the LFT status file and deleting a failed entry . . . . . 77

**Chapter 9. Handling exceptions for the Web Services Gateway . . . . . 79**

**Chapter 10. Web Services Gateway troubleshooting tips . . . . . 81**
Enabling trace in WebSphere Application Server . . 82
Message reference for the Web Services Gateway . . 83

**Notices . . . . . 107**
Programming interface information . . . . . 108
Trademarks and service marks . . . . . 109


© Copyright IBM Corp. 2002

# Chapter 1. Using the Web Services Gateway

Use this topic to enable Web services through the IBM Web services Gateway

You use the IBM Web Services Gateway to handle Web Service invocations between Internet and Intranet environments. You use it to make your internal Web services available externally, and to make external Web services available to your internal systems.

With the Web Services Gateway you can administer Web services, Channels, Filters and UDDI registrations.

Detailed instructions on how to enable Web services through the IBM Web Services Gateway are given in the following tasks:

- Chapter 4, “Web Services Gateway - Completing the installation”, on page 7.
- Chapter 5, “Administering the Web Services Gateway”, on page 13.
- Chapter 6, “Running the Web Services Gateway samples”, on page 47.
- Chapter 7, “Administering security for the Web Services Gateway”, on page 53.
- Chapter 8, “Transferring large files from gateway to gateway”, on page 55.
- Chapter 9, “Handling exceptions for the Web Services Gateway”, on page 79.
- Chapter 10, “Web Services Gateway troubleshooting tips”, on page 81.

For a brief overview of what the Web Services Gateway is for, and how it works, see Chapter 2, “Web Services Gateway - Frequently Asked Questions”, on page 3.

For a list of the major changes since the AlphaWorks preview version of the Web services Gateway, see Chapter 3, “Web Services Gateway - What is new in this release”, on page 5.

For additional technical details of the Web Services Gateway, see the gateway Javadoc.

## Related tasks

Chapter 2, “Web Services Gateway - Frequently Asked Questions”, on page 3

Chapter 3, “Web Services Gateway - What is new in this release”, on page 5

Chapter 4, “Web Services Gateway - Completing the installation”, on page 7

Chapter 5, “Administering the Web Services Gateway”, on page 13

Chapter 6, “Running the Web Services Gateway samples”, on page 47

Chapter 7, “Administering security for the Web Services Gateway”, on page 53

Chapter 8, “Transferring large files from gateway to gateway”, on page 55

Chapter 9, “Handling exceptions for the Web Services Gateway”, on page 79

© Copyright IBM Corp. 2002

Chapter 10, “Web Services Gateway troubleshooting tips”, on page 81

# Chapter 2. Web Services Gateway - Frequently Asked Questions

This topic provides answers to the following set of frequently asked questions about the Web Services Gateway:

- **What are Web services?**

  Web services are modular applications that interact with one another across the Internet. Web services are based on shared, open and emerging technology standards and protocols (such as SOAP, UDDI, and WSDL) and can communicate, interact, and integrate with other applications, no matter how they are implemented.

- **What is the IBM Web Services Gateway?**

  The gateway is a middleware component that bridges the gap between Internet and Intranet environments during Web service invocations. You use it to manage

  – Web services.

  – channels that carry requests to and responses from the services.

  – filters that act upon the services.

  – references to UDDI Registries in which services can be registered.

- **How does the Web Services Gateway work?**

  The gateway builds upon the Web services Definition Language (WSDL) and the Web Services Invocation Framework (WSIF) for deployment and invocation.

  You deploy a Web service to the Web Services Gateway by deploying a WSDL file which describes how the Web Services Gateway should access it. The WSDL file can be deployed to a UDDI Registry or to a URL. You can send requests passing through the Web Services Gateway to a Java class, an enterprise bean, a SOAP server or a SOAP/JMS server (including another gateway).

  A request to the Web Services Gateway arrives through a channel, is translated into an internal form, then passed through any filters that are registered for the requested service, and finally sent on to the service implementation. Responses follow the same path in reverse.

- **What problems are solved by the Web Services Gateway?**

  – ***Securely "externalizing" Web services:*** Business applications that are exposed as Web services can be used by any Web service-enabled tool, regardless of the implementation details, to create new applications. To better integrate your business processes, you might want to expose these assets to business partners, customers and suppliers who are outside the firewall. The Web Services Gateway lets clients from outside the firewall use Web services that are buried deep inside your enterprise. The gateway also allows you to set access control on each of these deeply-buried services.

  – ***Better return on investment:*** A process that you develop as a Web service can be reused by any number of partners.

  – ***Use of existing infrastructure:*** With the Web Services Gateway, you can use your existing messaging infrastructure to make Web service requests, and use your existing Web services for external process integration.

  – ***Protocol transformation:*** You might use one particular messaging protocol to invoke Web services, while your partners use some other protocol. Using the Web Services Gateway, you can trap the request from the client and transform it to another messaging protocol.

© Copyright IBM Corp. 2002

- **Who should use the Web Services Gateway?**

  Any enterprise that chooses to share its resources selectively with its business partners and customers. IT Managers and Developers, who deploy resources, can also benefit from this technology.

## Related tasks

Chapter 1, “Using the Web Services Gateway”, on page 1

# Chapter 3. Web Services Gateway - What is new in this release

The Web Services Gateway was first made available on **AlphaWorks** http://www.alphaworks.ibm.com/tech/wsgw on 21 December 2001. The main differences between the AlphaWorks edition and the WebSphere Business Connection version are as follows:

- The gateway has been rebuilt using enterprise beans.

  **Note:** A side-effect of this is that the Web Services Gateway now only runs in an application server that has an EJB container. So it no longer runs in the Tomcat server.
- The gateway includes UDDI integration, so you can deploy and remove Web services to a UDDI registry as well as to a URL.
- In addition to the Apache SOAP channel, there is now also an Apache AXIS channel and a Large File Transfer (LFT) channel (and associated LFT Web services).
- The gateway supports bidirectional interactions (that is, both inbound and outbound requests) directly, by deploying two instances of each type of channel.

  **Note:** To achieve this configuration with the AlphaWorks version, you had to deploy two instances of the Web Services Gateway; one for inbound communication and one for outbound communication.
- ″Interceptors″ have been renamed as ″filters″.
- Channels, filters and UDDI references are deployed to the Web Services Gateway, then associated with individual Web services. So when you configure a Web service, you choose the following entities:
  - The channels on which it is available.
  - The filters (if any) which apply to it.
  - The UDDI references (if any) to which it is deployed.
- You can change the channels, filters and UDDI references that are associated with a deployed service without having to remove the service.
- You can deploy multiple targets for a single service (that is, more than one implementation of a service that has the same service interface).
- You can set security (basic authorization) on the individual methods of a Web service, as well as for the gateway as a whole.

© Copyright IBM Corp. 2002

# Chapter 4. Web Services Gateway - Completing the installation

This task assumes that you have completed the basic install and configuration of the gateway as described in the WebSphere Business Connection ″Installation and Configuration Guide″. If you have done this, then you have already completed the following steps:

1. The gateway application is installed and deployed into an individual application server instance in your network space.
2. The following channels are installed, and deployed as described in “Deploying channels to the Web Services Gateway” on page 18:
   - SOAP Channel 1
   - SOAP Channel 2
   - Axis Channel 1
   - Axis Channel 2
   - LFT Channel 1
   - LFT Channel 2
3. The WebSphere Business Connection routing filters are installed and deployed. For more general information about these filters, see “The WebSphere Business Connection routing filters” on page 23.
4. The WebSphere Business Connection exception handler is installed. For more general information see Chapter 9, “Handling exceptions for the Web Services Gateway”, on page 79.
5. The WebSphere Application Server eFix that is included in the WebSphere Business Connection package is installed.

   **Note:** Without this eFix the application server http plugin cannot handle chunked request content, and the request fails if the plugin receives a chunked request content. The un-patched plugin only functions for POST requests that contain a content length. This is acceptable for current Web browsers (which do not chunk POST content) but not for Chapter 8, “Transferring large files from gateway to gateway”, on page 55 (which does chunk POST content).
6. The LFT sample Web services are installed and deployed.

To finish the Web Services Gateway installation, complete the following steps:

1. Confirm that your system configuration complies with the “Web Services Gateway - prerequisites and constraints” on page 8.
2. “Testing the Web Services Gateway installation” on page 9 that the Web Services Gateway has been installed correctly.

**What next**

At this time you might also want to “Installing other Web Services Gateway applications” on page 10 (for example other filters or channels).

If you want to enable Large File Transfer (LFT) support, complete the steps in Chapter 8, “Transferring large files from gateway to gateway”, on page 55.

© Copyright IBM Corp. 2002

### Related tasks

Chapter 1, “Using the Web Services Gateway”, on page 1

“Testing the Web Services Gateway installation” on page 9

“Installing other Web Services Gateway applications” on page 10

### Related reference

“Web Services Gateway - prerequisites and constraints”

## Web Services Gateway - prerequisites and constraints

This topic assumes that your installation complies with the hardware and software prerequisites detailed in the WebSphere Business Connection ″Installation and Configuration Guide″.

This version of the Web Services Gateway is also subject to the following constraints:

- WSDL definitions for target services must use XML Schema version 2001. For more information, see Chapter 10, “Web Services Gateway troubleshooting tips”, on page 81.
- The gateway application (wsgw.ear) must be installed before channel and filter applications. If the gateway application needs to be reinstalled, all channels and filters must be uninstalled first, then reinstalled after the gateway application.
- The gateway does not support WSDL service definitions that contain **soap:header** elements within their **wsdl:definition** element.

You might also find it useful to enable trace for all Gateway components, and have trace, stdout and stderr for the application server written to a well-known location. For information on how to do this, see “Enabling trace in WebSphere Application Server” on page 82.

### Related tasks

Chapter 4, “Web Services Gateway - Completing the installation”, on page 7

“Testing the Web Services Gateway installation” on page 9

“Installing other Web Services Gateway applications” on page 10

### Related reference

Chapter 10, “Web Services Gateway troubleshooting tips”, on page 81

## Testing the Web Services Gateway installation

Use this task to test that the Web Services Gateway has been installed correctly.

To test the basic installation of the Web Services Gateway, complete the following steps:

1. In a Web browser, go to `http://host:port/wsgw`where *host* and *port* are the host name and port number that your HTTP server is listening on.

   The browser should display the following message:

   What do you want to do?

   - Run the admin client

2. If the previous step was successful, then complete the following steps:

   a. To test the Apache SOAP channels, use your Web browser to display the Web page at the following URL:

   `http://host:port/wsgwengine/soaprpcrouter`

   where *engine* is either `soap1` or `soap2`

   The browser should display the following message: `Sorry, I don't speak via HTTP GET - you have to use HTTP POST to talk to me.`

   b. To test the Apache Axis channels, use your Web browser to display the Web page at the following URL:

   `http://host:port/wsgwengine/axisengine`

   where *engine* is either `axis1` or `axis2`

   The browser should display the following message: `Hi there, this is an AXIS service!`

   If you don't see these messages, your server is not configured correctly - in which case, see Chapter 10, "Web Services Gateway troubleshooting tips", on page 81.

**What next**

**Note:** There is no equivalent basic test for the LFT channels. To confirm that large file transfer is working, you run the LFT sample Web services as detailed in Chapter 8, "Transferring large files from gateway to gateway", on page 55.

### Related tasks

Chapter 4, "Web Services Gateway - Completing the installation", on page 7

"Installing other Web Services Gateway applications" on page 10

### Related reference

"Web Services Gateway - prerequisites and constraints" on page 8

Chapter 10, “Web Services Gateway troubleshooting tips”, on page 81

## Installing other Web Services Gateway applications

You have already installed the WebSphere Business Conection enterprise applications, including the channels and filters listed in Chapter 4, “Web Services Gateway - Completing the installation”, on page 7

You use this topic to install any additional gateway applications - for example your own channels or filters, or “The Simple Log filter” on page 24.

To install additional EAR files into WebSphere Application Server using the Install Enterprise Application wizard of the WebSphere Application Server Administrator’s Console, complete the following steps:

1. Start the WebSphere Application Server adminstration server.
2. Start the administrative console.
3. In the navigation pane, expand **WebSphere Administrative Domain**.
4. For each EAR file that you want to install, complete the following steps:
   a. From the pop-up menu for **Enterprise Applications**, click **Install Enterprise Application**.

      The Install Enterprise Application wizard opens.
   b. In the Install Enterprise Application wizard, complete the following steps:
      1) Confirm that the **Install Application (*.ear)** radio button is enabled.
      2) Enter the path to the EAR file (or click **Browse** to locate the EAR file) on your system.
      3) Click **Next** for this panel and for each subsequent panel in the wizard, then click **Finish** on the final panel.

      The first Deploy window opens, and offers to generate code necessary for installation.
   c. On the first Deploy window, click **No**.

      The second Deploy window opens, and displays basic deployment information.
   d. On the second Deploy window, click **OK**.
   e. Application deployment begins.

      **Note:** After a time, the second Deploy window closes, but installation of the application is not yet complete.
   f. When a window opens to tell you that the application installation has completed successfully, click **OK**.
5. After you have installed all of the EAR files, complete the following steps:
   a. From the pop-up menu for the node on which the applications are installed, click **Regen the Web server plugin**.
   b. Stop then restart the application server.

### Related tasks

Chapter 4, “Web Services Gateway - Completing the installation”, on page 7

“Testing the Web Services Gateway installation” on page 9

## Related reference

"Web Services Gateway - prerequisites and constraints" on page 8

Chapter 10, "Web Services Gateway troubleshooting tips", on page 81

# Chapter 5. Administering the Web Services Gateway

To administer the Web Services Gateway, complete the following steps:

1. Start the WebSphere Application Server Administrative Server.
2. Open the following Web page: `http://host:port/wsgw/admin/index.html`
   where *host* and *port* are the host name and port number that your HTTP server is listening on. For example `localhost:8080` or `localhost:9080`.
   The main administration page is displayed:

The order of the elements on this page is significant, for the following reasons:

- If you change the namespace URI or WSDL URI (using the **Configure Gateway** option), you break the link back to the gateway for every Web service that you have already deployed. So you must set these URIs before you deploy any Web services.

© Copyright IBM Corp. 2002

- When you configure a Web service, you choose the following entities:
  - The Channels on which it is available.
  - The Filters (if any) which apply to it.
  - The UDDI References (if any) to which it is deployed.

  Each of these choices is made from a list of resources which have already been deployed to the Web Services Gateway. So you might want to deploy your channels, filters and UDDI references to the gateway before you configure the Web services that use them.

3. For more information on how to configure each element of the Web services Gateway, see the following topics:
   - "Setting the namespace URI and WSDL URI for the Web Services Gateway"
   - "Working with channels" on page 16
   - "Working with filters" on page 22
   - "Working with UDDI references" on page 35
   - "Working with Web services" on page 39

   **Note:**
   - You configure each of the above elements of the gateway by filling in fields in a panel.
   - In all of the gateway' panels, fields marks with asterisks are required.
   - After you deploy a channel, filter, or UDDI reference you should refresh all other open browser windows to ensure that up-to-date lists are displayed.

### Related tasks

Chapter 1, "Using the Web Services Gateway", on page 1

"Setting the namespace URI and WSDL URI for the Web Services Gateway"

"Working with channels" on page 16

"Working with filters" on page 22

"Working with UDDI references" on page 35

"Working with Web services" on page 39

## Setting the namespace URI and WSDL URI for the Web Services Gateway

Use this task to set the namespace URI for the Web Services Gateway.

Initial values for the namespace URI and WSDL URI are automatically configured when you Chapter 4, "Web Services Gateway - Completing the installation", on page 7.

When you deploy a Web Service to the Web Services Gateway, these two URIs are used as follows:

- The **Namespace URI for services** is used as the namespace for the gateway services in exported WSDL documents.
- The **WSDL URI for exported definitions** is used to generate the URL in import statements within exported WSDL documents.

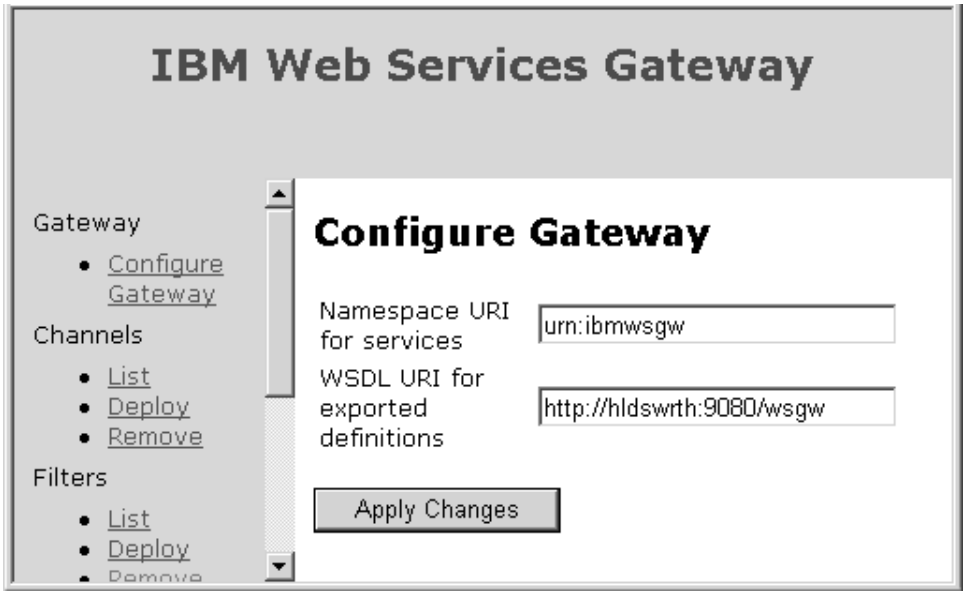
**Note:** When you change these URIs, you break the link back to the Web services Gateway for every Web service that you have already deployed. So you must set these URIs before you deploy any Web services to the Web services Gateway.

To set the namespace URI and WSDL URI for the Web services Gateway, complete the following steps:

1. Chapter 5, "Administering the Web Services Gateway", on page 13
2. In the navigation pane, click the following link:

   **Gateway**
   - Configure Gateway

   The gateway administration form is displayed:




3. In the **Namespace URI for services** field, type the new name.

   There is no fixed syntax for the namespace URI, but whatever name you choose is likely to be more effective if it observes the following guidelines:
   - It begins with "urn:"

     See the guidance on Internet standards for the syntax of Uniform Resource Names (URNs) at **http://www.ietf.org/rfc/rfc2141.txt** http://www.ietf.org/rfc/rfc2141.txt .
   - It is globally unique.
   - It reflects your company name.
4. In the **WSDL URI for exported definitions** field, type the new name.

   The initial value is the gateway's "best guess" at the right value, but you will probably want to overwrite it with a new value. For example it might guess a local URI such as `http://hldswrth:9080/wsgw`, and because you are giving the WSDL to people in other companies you modify this to

`http://hldswrth.your_company.com/wsgw`. Note that only the *host* and *port* parts of the initial value are modified, and that this URI should always start `http://`and end `/wsgw`.

5. Click **Apply Changes**.

### Related tasks

"Working with channels"

"Working with filters" on page 22

"Working with UDDI references" on page 35

"Working with Web services" on page 39

Chapter 5, "Administering the Web Services Gateway", on page 13

## Working with channels

You have already installed and deployed the WebSphere Business Connection channels that are listed in Chapter 4, "Web Services Gateway - Completing the installation", on page 7. Before you can work with any other channel, you must install the channel application in WebSphere Application Server as described in "Installing other Web Services Gateway applications" on page 10.

Two versions of each type of channel are supplied so that, for each channel type, you can set up separate channels for inbound and outbound requests. For more information see "Channels - entry points to the Web Services Gateway" on page 17.

From the navigation pane of the Chapter 5, "Administering the Web Services Gateway", on page 13, you can choose the following actions for **Channels**:

- **"Listing and managing gateway-deployed channels" on page 17** to list the deployed channels, and modify their deployment details.
- **"Deploying channels to the Web Services Gateway" on page 18** to deploy a channel.
- **"Removing channels from the Web Services Gateway" on page 21** to remove channels.

### Related concepts

"Channels - entry points to the Web Services Gateway" on page 17

### Related tasks

"Listing and managing gateway-deployed channels" on page 17

"Deploying channels to the Web Services Gateway" on page 18

"Removing channels from the Web Services Gateway" on page 21

"Setting the namespace URI and WSDL URI for the Web Services Gateway" on page 14

"Working with filters" on page 22

"Working with UDDI references" on page 35

"Working with Web services" on page 39

Chapter 5, "Administering the Web Services Gateway", on page 13

## Channels - entry points to the Web Services Gateway

Channels form entry points to the Web Services Gateway and carry requests and responses between Web services and the Web Services Gateway. A request to the Web Services Gateway arrives through a channel, is translated into a WSIF message, then passed through any cwsg_filter that are registered for the requested service, and finally sent on to the service implementation. Responses follow the same path in reverse.

Before you can use a channel, you must install the channel application in WebSphere Application Server then deploy the channel to the Web services Gateway.

**Note:** A deployed channel is not used until you deploy a Web Service that uses the channel.

Two versions of each type of channel are supplied with the gateway. This is so that, for each channel type, you can set up separate channels for inbound and outbound requests. This provides a simple mechanism for giving different access rights to users from outside your organisation from the rights you give to users within your organisation:

- To ensure that users outside your organisation can only access those internal services that you choose to publish externally, you deploy those services on the inbound channel.
- To give users inside your organisation access to the full range of internal and external services, you deploy those services on the outbound channel.

### Related tasks

"Working with channels" on page 16

## Listing and managing gateway-deployed channels

Use this task to list the channels that are deployed to the Web services Gateway, and modify their deployment details.

To list the channels that are currently deployed to the Web services Gateway, and view and modify their deployment details, complete the following steps:

1. Chapter 5, "Administering the Web Services Gateway", on page 13
2. In the navigation pane, click the following link:

**Channels**

- List

The main pane is updated with a list of all the channels that are deployed to the Web Services Gateway.

3. Click the name of a channel in the list. A form is displayed through which you can view and modify the current deployment details for this channel.
4. Modify the following deployment details:

   **Home Location**
   Type the name of the new home for this channel.

   **End Point Address**
   Type the new address on which the channel is to listen.
5. If this channel is intended to be used to receive asynchronous reply messages, type appropriate values in the following two fields. Otherwise leave them blank.

   **Note:** If the channel supports asynchronous messaging, then the "Web Services Gateway - Channel deployment details" on page 20 for the channel indicates what values to enter in these fields.

   **Async Reply Context Name**

   **Async Reply Context Value**
6. To start this channel, enable the **YES** radio button. To stop this channel, enable the **NO** radio button.
7. Click **Apply changes**.

**Results**

If the processing completes successfully, the list of deployed channels is redisplayed. Otherwise, an error message is displayed.

### Related tasks

"Deploying channels to the Web Services Gateway"

"Removing channels from the Web Services Gateway" on page 21

"Working with channels" on page 16

## Deploying channels to the Web Services Gateway

Use this task to deploy a channel to the Web Services Gateway.

Before you can deploy a channel, you must install the channel application in WebSphere Application Server as described in "Installing other Web Services Gateway applications" on page 10.

If you want to deploy the channels supplied with the Web Services Gateway, their deployment details are listed in "Web Services Gateway - Channel deployment details" on page 20.

To deploy a channel, complete the following steps:

1. Chapter 5, “Administering the Web Services Gateway”, on page 13
2. In the navigation pane, click the following link:

   **Channels**

   - Deploy

   A form is displayed for you to specify the deployment details.
3. Type the following channel deployment details:

   **Channel Name**
   Type the name by which the channel is known within the Web Services Gateway, and by which it is listed using the **Channels > List** option. This name must be unique within the Web Services Gateway.

   **Home Location**
   Type the name of the home for this channel.

   **End Point Address**
   Type the address on which the channel is to listen.
4. If this channel is intended to be used to receive asynchronous reply messages, type appropriate values in the following two fields. Otherwise leave them blank.

   **Note:** If the channel you are deploying supports asynchronous messaging, then the “Web Services Gateway - Channel deployment details” on page 20 for the channel indicates what values to enter in these fields.

   **Async Reply Context Name**

   **Async Reply Context Value**
5. Click **OK**.

**Results**

If the processing completes successfully, the list of deployed channels is updated to include the new channel. Otherwise, an error message is displayed.

**What next**

**Note:** The deployed channel will not be used until you deploy a Web Service that uses the channel.

## Related tasks

“Working with channels” on page 16

“Listing and managing gateway-deployed channels” on page 17

“Removing channels from the Web Services Gateway” on page 21

## Related reference

“Web Services Gateway - Channel deployment details” on page 20

## Web Services Gateway - Channel deployment details

The deployment details for the channels supplied with WebSphere Business Connection are listed below:

- Apache SOAP Channel 1
  - **Channel Name:** ApacheSOAPChannel1
  - **Home Location:** ApacheSOAPChannel1Bean
  - **End Point Address:** http://*host*:*port*/wsgwsoap1

    where *host* and *port* are the host name and port number for the application server on which the channel application is installed.
  - **Async Reply Context Name:** Leave blank. This function is not supported by this channel.
  - **Async Reply Context Value:** Leave blank. This function is not supported by this channel.
- Apache SOAP Channel 2
  - **Channel Name:** ApacheSOAPChannel2
  - **Home Location:** ApacheSOAPChannel2Bean
  - **End Point Address:** http://*host*:*port*/wsgwsoap2
  - **Async Reply Context Name:** Leave blank. This function is not supported by this channel.
  - **Async Reply Context Value:** Leave blank. This function is not supported by this channel.
- Apache Axis Channel 1
  - **Channel Name:** ApacheAxisChannel1
  - **Home Location:** ApacheAxisChannel1Bean
  - **End Point Address:** http://*host*:*port*/wsgwaxis1
  - **Async Reply Context Name:** Leave blank. This function is not supported by this channel.
  - **Async Reply Context Value:** Leave blank. This function is not supported by this channel.
- Apache Axis Channel 2
  - **Channel Name:** ApacheAxisChannel2
  - **Home Location:** ApacheAxisChannel2Bean
  - **End Point Address:** http://*host*:*port*/wsgwaxis2
  - **Async Reply Context Name:** Leave blank. This function is not supported by this channel.
  - **Async Reply Context Value:** Leave blank. This function is not supported by this channel.
- LFT Channel 1
  - **Channel Name:** LFTChannel1
  - **Home Location:** LFTChannel1Bean
  - **End Point Address:** http://*host*:*port*/wsgwlft1
  - **Async Reply Context Name:** Leave blank. This function is not supported by this channel.
  - **Async Reply Context Value:** Leave blank. This function is not supported by this channel.
- LFT Channel 2
  - **Channel Name:** LFTChannel2

- **Home Location:** LFTChannel2Bean
- **End Point Address:** http://*host*:*port*/wsgwlft2
- **Async Reply Context Name:** Leave blank. This function is not supported by this channel.
- **Async Reply Context Value:** Leave blank. This function is not supported by this channel.

### Related tasks

"Deploying channels to the Web Services Gateway" on page 18

## Removing channels from the Web Services Gateway

Use this task to remove a channel from the Web Services Gateway.

You cannot remove a channel that is being used by a Web Service. If you try to do this, you will not harm your configuration but the channel will not be removed. So before you remove a channel, use the "Listing and managing gateway-deployed Web services" on page 40 option to modify the deployment details for any deployed Web service that uses the channel.

To remove a channel, complete the following steps:

1. Chapter 5, "Administering the Web Services Gateway", on page 13
2. In the navigation pane, click the following link:

   **Channels**

   - Remove

   The main pane is updated with a list of all the channels that are deployed to the Web Services Gateway. To the left of each entry in the list is a check box.
3. Select the check box for every channel that you want to remove.
4. Click **OK**.

**Results**

If the processing completes successfully, the list of deployed channels is updated. Otherwise, an error message is displayed.

### Related tasks

"Working with channels" on page 16

"Listing and managing gateway-deployed channels" on page 17

"Deploying channels to the Web Services Gateway" on page 18

## Working with filters

You have already installed and deployed "The WebSphere Business Connection routing filters" on page 23. Before you can work with any other filter (for example "The Simple Log filter" on page 24), you must install the filter application in WebSphere Application Server as described in "Installing other Web Services Gateway applications" on page 10.

From the navigation pane of the Chapter 5, "Administering the Web Services Gateway", on page 13, you can choose the following actions for **Filters**:

- **"Listing and managing gateway-deployed filters" on page 25** to list the deployed filters, and modify their deployment details.
- **"Deploying filters to the Web Services Gateway" on page 26** to deploy a filter.
- **"Removing filters from the Web Services Gateway" on page 27** to remove filters.

You can also "Writing a filter for the Web Services Gateway" on page 27 for the Web services Gateway.

### Related concepts

"Filters - service interceptors for the Web Services Gateway" on page 23

### Related tasks

Chapter 5, "Administering the Web Services Gateway", on page 13

"The WebSphere Business Connection routing filters" on page 23

"The Simple Log filter" on page 24

"Listing and managing gateway-deployed filters" on page 25

"Deploying filters to the Web Services Gateway" on page 26

"Removing filters from the Web Services Gateway" on page 27

"Writing a filter for the Web Services Gateway" on page 27

"Setting the namespace URI and WSDL URI for the Web Services Gateway" on page 14

"Working with channels" on page 16

"Working with UDDI references" on page 35

"Working with Web services" on page 39

## Filters - service interceptors for the Web Services Gateway

Filters are used to intercept service invocations which come into the Web services Gateway, and responses which leave it. Filters can perform a wide range of tasks, from logging messages, to transforming their content, to terminating an incoming request. Filters are deployed to the Web Services Gateway as described in "Deploying filters to the Web Services Gateway" on page 26, then registered for use with individual Web services as described in "Working with Web services" on page 39.

### Related tasks

"Working with filters" on page 22

"The WebSphere Business Connection routing filters"

"The Simple Log filter" on page 24

"Listing and managing gateway-deployed filters" on page 25

"Deploying filters to the Web Services Gateway" on page 26

"Removing filters from the Web Services Gateway" on page 27

"Writing a filter for the Web Services Gateway" on page 27

## The WebSphere Business Connection routing filters

The Gateway represents each exported service as a gateway service. Each gateway service can map to one or more target services, but without a routing filter there is no point in mapping multiple targets as the gateway will always pick the first one. So if you want to map multiple targets, you also need to use the WebSphere Business Connection routing filters (configured for each gateway service) to select the target service from those available.

Two version of a routing filter are supplied with WebSphere Business Connection:

- The SOAP routing filter provides for a selection from several potential target destinations for the same Web services when using a SOAP channel.
- The SOAP/LFT routing filter provides for a selection from several potential target destinations for the same Web service when using a SOAP channel and LFT channel for the same service.

These routing filters are automatically installed and deployed as part of the initial system setup that is described in the ″Installation and Configuration Guide″.

**Note:** Each target service is identified by the target service definition location (which is unique) and target service identity information (which might not be unique). So the routing filters select the first target service they find (for the current gateway service) with identity information that matches that specified.

### Related concepts

"Filters - service interceptors for the Web Services Gateway" on page 23

### Related tasks

"Working with filters" on page 22

"The Simple Log filter"

"Listing and managing gateway-deployed filters" on page 25

"Deploying filters to the Web Services Gateway" on page 26

"Removing filters from the Web Services Gateway" on page 27

"Writing a filter for the Web Services Gateway" on page 27

## The Simple Log filter

The Simple Log filter (`simplelog.ear`) is supplied in *`WSGW_HOME`*`/lib`(where *WSGW_HOME* is the root directory for your installation of the gateway).

The Simple Log filter, when installed and associated with a service, writes to the standard error log every request or response associated with that service.

To use the Simple Log filter, complete the following steps:

1. Install the filter application `simplelog.ear`in WebSphere Application Server as described in "Installing other Web Services Gateway applications" on page 10.
2. "Deploying filters to the Web Services Gateway" on page 26.

   The deployment details for the Simple Log filter are as follows:
   - Filter Name: SimpleLogFilter
   - Home Location: SimpleLogFilterBean
3. Use the Web services "Deploying Web services to the Web Services Gateway" on page 41 or "Listing and managing gateway-deployed Web services" on page 40 option to add the deployed filter to the list of **Pre-Filters** or **Post-Filters** for one or more Web services.

### Related concepts

"Filters - service interceptors for the Web Services Gateway" on page 23

### Related tasks

"Working with filters" on page 22

"The WebSphere Business Connection routing filters" on page 23

"Listing and managing gateway-deployed filters" on page 25

"Deploying filters to the Web Services Gateway" on page 26

"Removing filters from the Web Services Gateway" on page 27

"Writing a filter for the Web Services Gateway" on page 27

## Listing and managing gateway-deployed filters

Use this task to list the filters that are deployed to the Web services Gateway, and modify their deployment details.

To list the filters that are currently deployed to the Web services Gateway, and view and modify their deployment details, complete the following steps:

1. Chapter 5, "Administering the Web Services Gateway", on page 13
2. In the navigation pane, click the following link:

   **Filters**

   - List

   The main pane is updated with a list of all the filters that are deployed to the Web Services Gateway.
3. Click the name of a filter in the list. A form is displayed through which you can view and modify the current deployment details for this filter.
4. Modify the following deployment detail:

   **Home Location**
   Type the name of the new home for this filter.
5. Click **Apply changes**.

**Results**

If the processing completes successfully, the list of deployed filters is redisplayed. Otherwise, an error message is displayed.

### Related concepts

"Filters - service interceptors for the Web Services Gateway" on page 23

### Related tasks

"Working with filters" on page 22

"The WebSphere Business Connection routing filters" on page 23

"The Simple Log filter" on page 24

"Deploying filters to the Web Services Gateway" on page 26

"Removing filters from the Web Services Gateway" on page 27

"Writing a filter for the Web Services Gateway" on page 27

## Deploying filters to the Web Services Gateway

Use this task to deploy a filter (for example "The Simple Log filter" on page 24) to the Web Services Gateway.

**Note:** The deployed filter will not be used until you deploy a Web Service that uses the filter.

Before you can deploy a filter, you must install the filter application in WebSphere Application Server as described in "Installing other Web Services Gateway applications" on page 10.

**Note:** You can deploy multiple instances of a filter by entering different filter names.

To deploy a filter, complete the following steps:

1. Chapter 5, "Administering the Web Services Gateway", on page 13
2. In the navigation pane, click the following link:

   **Filters**

   - Deploy

   A form is displayed for you to specify the deployment details.
3. Type the following filter deployment details:

   **Filter Name**
   Type the name by which the filter is known within the Web Services Gateway, and by which it is listed using the **Filters > List** option. This name must be unique within the Web Services Gateway.

   **Home Location**
   Type the name of the home for this filter.
4. Click **OK**.

**Results**

If the processing completes successfully, the list of deployed filters is updated to include the new filter. Otherwise, an error message is displayed.

### Related concepts

"Filters - service interceptors for the Web Services Gateway" on page 23

### Related tasks

"Working with filters" on page 22

"The WebSphere Business Connection routing filters" on page 23

"The Simple Log filter" on page 24

"Listing and managing gateway-deployed filters" on page 25

"Removing filters from the Web Services Gateway"

"Writing a filter for the Web Services Gateway"

## Removing filters from the Web Services Gateway

Use this task to remove a filter from the Web Services Gateway.

You cannot remove a filter that is being used by a Web Service. If you try to do this, you will not harm your configuration but the filter will not be removed. So before you remove a filter, use the "Listing and managing gateway-deployed Web services" on page 40 option to modify the deployment details for any deployed Web service that uses the filter.

To remove a filter, complete the following steps:

1. Chapter 5, "Administering the Web Services Gateway", on page 13
2. In the navigation pane, click the following link:

   **Filters**

   - Remove

   The main pane is updated with a list of all the filters that are deployed to the Web Services Gateway. To the left of each entry in the list is a check box.
3. Select the check box for every filter that you want to remove.
4. Click **OK**.

**Results**

If the processing completes successfully, the list of deployed filters is updated. Otherwise, an error message is displayed.

### Related concepts

"Filters - service interceptors for the Web Services Gateway" on page 23

### Related tasks

"Working with filters" on page 22

"The WebSphere Business Connection routing filters" on page 23

"The Simple Log filter" on page 24

"Listing and managing gateway-deployed filters" on page 25

"Deploying filters to the Web Services Gateway" on page 26

"Writing a filter for the Web Services Gateway"

## Writing a filter for the Web Services Gateway

Use this task to write a filter for the Web Services Gateway.

To use this information you should be familiar with using a J2EE session bean development environment such as IBM WebSphere Studio Application Developer.

A Web Services Gateway filter is essentially a J2EE session bean implementing specific Home and Remote interfaces.

To write a filter using IBM WebSphere Studio Application Developer, complete the following steps. For more detailed information on writing session beans, see the WebSphere Studio Application Developer documentation topic **Help -> Contents -> Application Developer Documentation -> Tasks -> Developing EJB Applications**.

1. Open the J2EE perspective.
2. To create a new EJB application project, complete the following steps:
   a. Select **File -> New -> Enterprise Application Project**.
      The Project Creation wizard opens.
   b. In the Project Creation wizard, complete the following steps:
      1) Type your project name.
      2) Clear the **Application Client Project Name** check box.
      3) Clear the **Web Project Name** check box.
      4) Click **Finish**.

      Your new EJB application project is created.
3. To add the extra JAR files that you need that are not already in the IBM WebSphere Application Server `/lib`directory, complete the following steps:
   a. Select **File -> Import**.
   b. Select the input source **File system**, then click **Next**.
   c. In the Import window, complete the following steps:
      1) Select *WSGW_HOME*`/lib`as the source directory
         where *WSGW_HOME* is the root directory for your installation of the gateway.
      2) Select the following JAR files within the source directory:
         - `commons_logging.jar`
         - `jrom.jar`
         - `qname.jar`
         - `wsdl4j.jar`
         - `wsgwejb.jar`
         - `wsif.jar`
      3) Select the root directory of your new project as the **destination for imported resources**.
      4) Click **Finish**.
   d. Repeat the previous **File -> Import** process to add any other extra JAR files that your EJB module needs.
   e. In the J2EE Hierarchy view, from the pop-up menu for your EJB module, select **Edit Module Dependencies**.
   f. In the Edit Module Dependencies window, select all the JAR files listed.
   g. Close Edit Module Dependencies window, then click **Yes** in the Save Resource window to save your changes.
4. To add extra JAR files to the Java build path for your EJB module, complete the following steps:

a. In the J2EE Hierarchy view, select your EJB module's **Properties**.
b. In the Properties window, ensure that the following JAR files are included on the **Java Build Path**:
   - *WSGW_HOME*/lib/commons_logging.jar
   - *WSGW_HOME*/lib/jrom.jar
   - *WSGW_HOME*/lib/qname.jar
   - *WSGW_HOME*/lib/wsdl4j.jar
   - *WSGW_HOME*/lib/wsgwejb.jar
   - *WSGW_HOME*/lib/wsif.jar
c. Add any other JAR files or projects that you need for compiling your filter.
d. Click **OK**.

5. To create the session bean, complete the following steps:
   a. Select **EJB Modules -> New -> Enterprise Bean**.
      The Enterprise Bean Creation wizard opens.
   b. In the Enterprise Bean Creation wizard, complete the following steps:
      1) Enter a name for the bean.
      2) Ensure that the bean type is **Session Bean**, then fill in the relevant fields and click **Next**.
      3) For the Home Interface, click **Class...**, then select the com.ibm.wsgw.beans.FilterHomeinterface.
      4) For the Remote Interface, click **Class...**, then select the com.ibm.wsgw.beans.FilterRemoteinterface.
      5) Click **Next**.
      6) Select **Add Package**, then add these packages to the import statements:
         - com.ibm.wsgw
         - com.ibm.wsgw.beans
         - org.apache.wsif
      7) Click **Finish**.

   Your new session bean is created.
6. The generated java code for your session bean does not implement the Filter. To update the code, complete the following steps:
   a. In the J2EE Hierarchy view, expand your session bean to show Java code entries for the Home interface, the Remote interface and for the session bean itself.
   b. In the J2EE Hierarchy view, double-click the entry for the session bean code. In the editor view, the generated code opens for editing.
   c. In the editor view, add ", Filter" to the end of the "implements javax.ejb.SessionBean" statement, then save the code. Ignore any errors at this stage.
7. To add the unimplemented methods of the Filter interface to your session bean, complete the following steps:
   a. Open the Outline view.
   b. In the Outline view, from the pop-up menu for your session bean, select **Add Unimplemented Methods**.
   c. Save the file. Any errors from the previous save are resolved.

   The methods of the Filter interface are added to your session bean.

8. Develop your filter.

   The exact steps that you take to develop your filter depend upon what you want it to do. However to develop any filter, you use the following resources:

   - The Filter interface.
   - The gateway Javadoc that describes the Filter interface.
   - The additional information on the Filter interface that is in "Web Services Gateway - the Filter interface" on page 31
   - The gateway message context. (This contains the context values for each message that comes into the Gateway. These are the values that your filter acts upon.)
   - The gateway Javadoc that describes the GatewayContextNames class. (To use the gateway message context values, you import the GatewayContextNames class.)
   - The additional information on the gateway message context fields that is in "Web Services Gateway - the gateway message context fields" on page 34.
   - The WSIF Javadoc for the following WSIF objects:
     - WSIFRequest
     - WSIFResponse
     - WSIFMessage
     - WSIFException

     (the methods of the Filter interface use these objects, as described in "Web Services Gateway - the Filter interface" on page 31.)
   - The example code below.

   **Note:** You must observe the J2EE programming model, and ensure that any non-gateway services you use are available on all platforms that the filter might be expected to run on. For example, you should not use static variables to store state information because on certain platforms, or in certain configurations (such as a cluster), a filter might be invoked in a different JVM for each request.

**Example**

This example shows you how to access the context and setting values in the filterRequest method of a filter.

```
import com.ibm.wsgw.GatewayContextNames;

...

    WSIFMessage contextMessage = null;

    // retrieve the context message from the WSIFRequest
    if ( request != null )
          contextMessage = request.getContextMessage();
    // set the target service and target port
    contextMessage.setObjectPart( GatewayContextNames.TARGET_SERVICE_LOCATION,
        "http://my.machine.com/MyServiceWSDL.wsdl" );
    contextMessage.setObjectPart( GatewayContextNames.TARGET_PORT_NAME,  "MyPort" );

    // set and return a FilterAction object, as the request has been updated
    FilterAction filterAction = new FilterAction();
    filterAction.setUpdatedRequest( request );

    return filterAction;

    ...
```

**What next**

After you have developed your filter, you need to generate deployment code and export the enterprise application. To do this using IBM WebSphere Studio Application Developer, complete the following steps:

1. Open the J2EE perspective.
2. In the J2EE Hierarchy view, from the pop-up menu for your EJB module, select **Generate -> Deploy and RMIC code**.
3. In the J2EE Hierarchy view, select your bean then open it with the **EJB Extension Editor**.
4. In the Editor view, on the **Bindings** tab, set the JNDI name to the Filter class name. This name will be used as the "Home Location" when the filter is deployed to the gateway, then save the file.
5. In the J2EE Hierarchy view, from the pop-up menu for your project, select **Export EAR file** to export the enterprise application.

You are now ready to "Installing other Web Services Gateway applications" on page 10 and "Deploying filters to the Web Services Gateway" on page 26 your filter.

### Related concepts

"Filters - service interceptors for the Web Services Gateway" on page 23

### Related tasks

"Working with filters" on page 22

"The WebSphere Business Connection routing filters" on page 23

"The Simple Log filter" on page 24

"Listing and managing gateway-deployed filters" on page 25

"Deploying filters to the Web Services Gateway" on page 26

"Removing filters from the Web Services Gateway" on page 27

### Related reference

"Web Services Gateway - the Filter interface"

"Web Services Gateway - the gateway message context fields" on page 34

## Web Services Gateway - the Filter interface

This topic gives more information on using each of the methods of the Filter interface. It supplements the information given in the following Javadoc:

- The gateway Javadoc for the Filter interface.
- The WSIF Javadoc for
  - WSIFRequest
  - WSIFResponse
  - WSIFMessage
  - WSIFException

The Filter interface represents an object which is called during service invocation. A bean which implements this interface can be registered to be called just before a request invocation, or just after response receipt for a particular service.

Gateway filters use the WSIFRequest context message object to get and set the "Web Services Gateway - the gateway message context fields" on page 34.

These are the two main methods you use for developing your filter:

- FilterAction filterRequest(org.apache.wsif.WSIFRequest request, org.apache.wsif.WSIFResponse response).
- FilterAction filterResponse(org.apache.wsif.WSIFRequest request, org.apache.wsif.WSIFResponse response).

Another important method that requires a specific value to be returned is getContextVersion().

If you want your filter to change the WSIFReponse and WSIFRequest messages, then note that changes to messages are only recognized if the setUpdatedRequest and setUpdatedResponse methods are called on the returned FilterAction object. The FilterAction object can also dictate whether processing of the message should continue by calling the setContinueProcess method.

**filterRequest Method**

The filterRequest method is called by the gateway Manager just before a request is sent to a target service. The return value from the method can indicate that the request should not be sent.

The request parameter contains the request WSIFMessage. This consists of a set of named parts. Each part has a value which is encoded as an instance of an appropriate Java object. Filters can change the values of the Java object instances, but should not add or remove parts, or replace the values of parts with ones of a different type.

The Filter might decide that the request should not proceed. In that case it has three options:

- Throw a FilterException. The Gateway logs the exception but continues processing filters and the request invocation.
- Throw a WSGWException. The Gateway logs and rethrows the exception, and processing of filters and the request is stopped. The exeption then goes back to the receiving channel, and the channel must determine what to do with the exception (in the case of SOAP-based channels, this results in a Fault message back to the client). This should only be done for unexpected errors in the filter.
- Return a FilterAction object with the continueProcessing flag set to false. In this case the response message in the FilterAction can also be set, and is sent to the originator of the request. No further filters are invoked.

If the request or response is modified, then it must be returned in an instance of the FilterAction class. If this is not done, any change to the response is ignored by the Web Services Gateway.

**filterResponse Method**

The filterResponse method is called by the gateway Manager just after a response has been received from a target service. The response parameter contains the response or fault WSIFMessage. This consists of a set of named parts.

Each part has a value which is encoded as an instance of an appropriate Java object. Filters can change the values of the Java object instances, but should not add or remove parts, or replace the values of parts with ones of a different type.

The Filter may decide that the response should not proceed. In that case it has three options:

- Throw a FilterException. The Gateway logs the exception but continues processing filters and the response invocation.
- Throw a WSGWException. The Gateway logs and rethrows the exception, and processing of filters and the response is stopped. The exeption then goes back to the receiving channel, and the channel must determine what to do with the exception (in the case of SOAP-based channels, this results in a Fault message back to the client). This should only be done for unexpected errors in the filter.
- Return a FilterAction object with the continueProcessing flag set to false. In this case the fault WSIF Message is set in the response message in the FilterAction, and is sent to the originator of the request. No further filters are invoked.

If the response is modified, then it must be returned in an instance of the FilterAction class. If this is not done, any change to the response is ignored by the Web Services Gateway.

If the Filter throws a FilterException, it is logged, but the gateway continues to process other filters. If it throws a WSGWException, processing of the response is stopped.

**getVersionString Method**

The getVersionString method returns a string form of the version of the filter implementation. This is used by the gateway when logging events relating to the filter so that the exact version of the filter implementation is known.

**getContextVersion Method**

The getContextVersion method indicates the approach that this filter uses to access context information. To access the message context information for IBM WebSphere Application Server Version 4, this method must be implemented to return the value: `Filter.CONTEXT_VERSION_WSIF_MESSAGE`.

**init Method**

The init method tells the filter that it has been configured with the Web Services Gateway.

This method is called by the gateway when it has been asked to add a filter.

**destroy Method**

The destroy method tells the filter that it is no longer configured with the Web Services Gateway. This method is called by the gateway when it has been asked to remove a filter.

### Related tasks

"Writing a filter for the Web Services Gateway" on page 27

## Web Services Gateway - the gateway message context fields

The gateway message context contains the context information for each incoming message.

You can use the **Context Field Constant** values if you import the class `com.ibm.wsgw.GatewayContextNames`.

For basic information on the fields that are available in the context, see the gateway Javadoc for the GatewayContextNames class. Additional information on all of these fields except `AUTH_SUBJECT` and `copyright`is given in the following table:

| Context Field | Context Field Constant | Action | Description |
|---|---|---|---|
| `WSGWAuthPassword` | `AUTH_PASSWORD` | Read and Write | Read the password from the incoming HTTP request (where available), or write the password to be set in an outbound HTTP header through WSIF |
| `WSGWAuthUserName` | `AUTH_USER_NAME` | Read and Write | Read the user name from the incoming HTTP request (where available), or write the userid to be set in an outbound HTTP header through WSIF |
| `WSGWGatewayServiceName` | `GATEWAY_SERVICE_NAME` | Read only | Name of the gateway Service for which the request was received. |
| `WSGWReceivingChannelName` | `RECEIVING_CHANNEL_NAME` | Read only | Name of the Channel on which the request was received. |
| `WSGWRetryCount` | `RETRY_COUNT` | Read and Write | Number of retries for the request (not currently used) |
| `WSGWSoapHeaders` | `SOAP_HEADERS` | Read and Write | Retrieve the SOAP headers for an inbound soap request; SOAP Headers are returned as a Vector of Nodes. |

| Context Field | Context Field Constant | Action | Description |
|---|---|---|---|
| WSGWTargetPortName | TARGET_PORT_ NAME | Read and Write | Currently selected port name. |
| WSGWTargetServic eName | TARGET_SERVICE_ LOCATION | Read and Write | Currently selected target service WSDL location. |
| WSGWTimeoutTime | TIMEOUT_TIME | Read and Write | Time-out value for the response (not currently used ) |
| WSGWMessageID | MESSAGE_ID | Read | Set by the channel and is a server-unique ID that can be used to correlate messages, for example, in trace. Can be made globally unique by prefixing with host name. |

### Related tasks

"Writing a filter for the Web Services Gateway" on page 27

## Working with UDDI references

A UDDI Reference is a pointer to a UDDI Registry. This may be a private UDDI Registry such as the , or a public UDDI Registry. For more information about UDDI and UDDI Registries, see the UDDI community at **uddi.org** http://uddi.org .

From the navigation pane of the Chapter 5, "Administering the Web Services Gateway", on page 13, you can choose the following actions for **UDDI References**:

- **"Listing and managing gateway-deployed UDDI references" on page 36** to list the deployed UDDI references, and modify their deployment details.
- **"Deploying UDDI references to the Web Services Gateway" on page 37** to deploy a UDDI reference.
- **"Removing UDDI references from the Web Services Gateway" on page 38** to remove UDDI references.

### Related tasks

"Listing and managing gateway-deployed UDDI references" on page 36

"Deploying UDDI references to the Web Services Gateway" on page 37

"Removing UDDI references from the Web Services Gateway" on page 38

"Setting the namespace URI and WSDL URI for the Web Services Gateway" on page 14

"Working with channels" on page 16

"Working with filters" on page 22

"Working with Web services" on page 39

Chapter 5, "Administering the Web Services Gateway", on page 13

## Listing and managing gateway-deployed UDDI references

Use this task to list the UDDI references that are deployed to the Web Services Gateway, and modify their deployment details.

To list the UDDI references that are currently deployed to the Web services Gateway, and view and modify their deployment details, complete the following steps:

1. Chapter 5, "Administering the Web Services Gateway", on page 13
2. In the navigation pane, click the following link:

   **UDDI References**

   - List

   The main pane is updated with a list of all the UDDI references that are deployed to the Web Services Gateway.
3. Click the name of a UDDI reference in the list. A form is displayed through which you can view and modify the current deployment details for this UDDI reference.
4. Modify the following deployment details:

   **Inquiry URL**
   Type the new URL that provides access to this registry for the SOAP inquiry API.

   **Publish URL**
   Type the new URL that provides access to this registry for the SOAP publish API.

   **User Name**
   Type the new user ID that has update access to the UDDI registry.

   **Password**
   Type the password for the new user ID specified in the **User Name** field.

   **Confirm Password**
   Type the password for the new user ID specified in the **User Name** field. This must match the password specified in the **Password** field.
5. Click **Apply changes**.

**Results**

If the processing completes successfully, the list of deployed UDDI references is redisplayed. Otherwise, an error message is displayed.

### Related tasks

"Deploying UDDI references to the Web Services Gateway"

"Removing UDDI references from the Web Services Gateway" on page 38

"Working with UDDI references" on page 35

## Deploying UDDI references to the Web Services Gateway

Use this task to deploy a UDDI reference to the Web Services Gateway.

**Note:** The deployed UDDI reference will not be used until you deploy a Web Service that uses the UDDI reference.

To deploy a UDDI reference, complete the following steps:

1. Chapter 5, "Administering the Web Services Gateway", on page 13
2. In the navigation pane, click the following link:

   **UDDI References**

   - Deploy

   A form is displayed for you to specify the deployment details.
3. Type the following UDDI Reference deployment details:

   **Reference Name**
   Type the name by which the UDDI reference is known within the Web Services Gateway, and by which it is listed using the **UDDI References > List** option. This name must be unique within the Web services Gateway.

   **Inquiry URL**
   Type the URL that provides access to this registry for the SOAP inquiry API.

   **Publish URL**
   Type the URL that provides access to this registry for the SOAP publish API.

   **User Name**
   Type the user ID that has update access to the UDDI registry.

   **Password**
   Type the password for the user ID specified in the **User Name** field.

   **Confirm Password**
   Type the password for the user ID specified in the **User Name** field. This must match the password specified in the **Password** field.

   **Note:**

   The values you enter here for **User Name** and **Password** must match those of the owner of the corresponding business in UDDI. You can see the owning user ID in UDDI by looking at the business details under the ″Authorized Name″ field.

   If the values you enter here do not match the ″Authorized Name″ values for the business that owns the service, then the service will not be published or found.

If the business that owns the service has more than one ″Authorized Name″, you might want to set up multiple UDDI references (each with a different user ID) to the same UDDI registry .

4. Click **OK**.

**Results**

If the processing completes successfully, the list of deployed UDDI references is updated to include the new UDDI reference. Otherwise, an error message is displayed.

### Related tasks

"Working with UDDI references" on page 35

"Listing and managing gateway-deployed UDDI references" on page 36

"Removing UDDI references from the Web Services Gateway"

## Removing UDDI references from the Web Services Gateway

Use this task to remove a UDDI reference from the Web Services Gateway.

You cannot remove a UDDI reference that is being used by a Web Service. If you try to do this, you will not harm your configuration but the UDDI reference will not be removed. So before you remove a UDDI reference, use the "Listing and managing gateway-deployed Web services" on page 40 option to modify the deployment details for any deployed Web service that uses the UDDI reference.

To remove a UDDI reference, complete the following steps:

1. Chapter 5, "Administering the Web Services Gateway", on page 13
2. In the navigation pane, click the following link:
   **UDDI References**
   - Remove

   The main pane is updated with a list of all the UDDI references that are deployed to the Web Services Gateway. To the left of each entry in the list is a check box.
3. Select the check box for every UDDI reference that you want to remove.
4. Click **OK**.

**Results**

If the processing completes successfully, the list of deployed UDDI references is updated. Otherwise, an error message is displayed.

### Related tasks

"Working with UDDI references" on page 35

"Listing and managing gateway-deployed UDDI references" on page 36

"Deploying UDDI references to the Web Services Gateway" on page 37

## Working with Web services

Use this task to configure a Web service.

If you change the Namespace URI, you break the link back to the Web Services Gateway for every Web service that you have already deployed. So you must "Setting the namespace URI and WSDL URI for the Web Services Gateway" on page 14 before you deploy any Web services.

When you configure a Web service, you choose the following resources:

- The channels on which the service is available.
- Any filters that apply to the service.
- Any UDDI references to UDDI registries in which the service is deployed.

Each of these choices is made from a list of resources that have already been deployed to the Web Services Gateway. So you must deploy your "Working with channels" on page 16, "Working with filters" on page 22 and "Working with UDDI references" on page 35 to the Web Services Gateway before you deploy the Web services that use those resources.

From the navigation pane of the Chapter 5, "Administering the Web Services Gateway", on page 13, you can choose the following actions for **Services**:

- **"Listing and managing gateway-deployed Web services" on page 40** to list the deployed Web services, and modify their deployment details.
- **"Deploying Web services to the Web Services Gateway" on page 41** to deploy a Web service.
- **"Removing Web services from the Web Services Gateway" on page 44** to remove Web services.

### Related tasks

"Listing and managing gateway-deployed Web services" on page 40

"Deploying Web services to the Web Services Gateway" on page 41

"Removing Web services from the Web Services Gateway" on page 44

"Setting the namespace URI and WSDL URI for the Web Services Gateway" on page 14

"Working with channels" on page 16

"Working with filters" on page 22

"Working with UDDI references" on page 35

Chapter 5, "Administering the Web Services Gateway", on page 13

## Listing and managing gateway-deployed Web services

Use this task to list the Web services that are deployed to the Web Services Gateway, and modify their deployment details.

To list the Web services that are currently deployed to the Web services Gateway, and view and modify their deployment details (including adding or removing multiple target services) complete the following steps:

1. Chapter 5, "Administering the Web Services Gateway", on page 13
2. In the navigation pane, click the following link:

   **Services**

   - List

   The main pane is updated with a list of all the Web services that are deployed to the Web Services Gateway.
3. Click the name of a Web service in the list. A form is displayed through which you can view and modify the current deployment details for this Web service, and add or remove multiple target services.
4. At the level of the gateway service itself, you can change the following settings. When you have finished making changes, click **Apply Changes**.
   a. **Authorization Policy - Control access to this service**. Use this check box to enable or disable Chapter 7, "Administering security for the Web Services Gateway", on page 53 for this gateway service.
   b. **Audit Policy - Log requests to this service**. The Audit policy indicates whether the WebSphere Business Connection Solution Manager is used to log requests and responses for this Web service.
5. In the **Target Services** section, you can add or remove single instances of multiple target services for this gateway service. To add a new target service instance, complete the following steps:
   a. **WSDL Location**. Type the location of the ″internal″ WSDL file that describes the Web service to be deployed.

      This value is either a URL or the UDDI lookup information in the form:

      *uddi_reference_name*,*tmodel_key*,*key_name*,*key_value*

      When the Web Services Gateway deploys the Web service, it generates a matching ″external″ WSDL file that it makes available to gateway users. This ″external″ WSDL file also describes the service, but is located at a new URL and is generated and maintained by the Web Services Gateway itself.
   b. **Location Type**. Select the type of location (either URL or UDDI) where the ″internal″ WSDL file is held.
   c. **Target Service Name**. If the Web service WSDL contains more than one service, type the target service's name from the target service WSDL.
   d. **Target Service Namespace**. If the Web service WSDL contains more than one service, type the namespace of the target service's name from the target service WSDL.
   e. **Target Service Identity Information**. Type the identity by which the target service is known within the Web Services Gateway. This identity need not be unique.

      **Note:** If you are mapping multiple target services, you can use "The WebSphere Business Connection routing filters" on page 23 to select a particular target service from the set.
   f. Click **add**.

6. In the **Channels** section, you can add or remove channels from the list of deployed channels through which this service is available.
7. In the **Pre-Filters** section, you can add or remove filters from the list of deployed filters that are applied to the request.

   **Note:** The filters are executed in the order shown. To add a filter into the list at a particular position, you must remove existing filters which should come after the new one, install the new one, then reinstall the old ones.
8. In the **Post-Filters** section, you can add or remove filters from the list of deployed filters that are applied to the response.

   **Note:** The filters are executed in the order shown. To add a filter into the list at a particular position, you must remove existing filters which should come after the new one, install the new one, then reinstall the old ones.
9. In the **UDDI References** section, you can add or remove UDDI references from the list of deployed UDDI references to UDDI registries in which this service is registered.

   **Note:** After you add or remove UDDI references, the gateway publishes this service to (or removes it from) the UDDI registries which correspond to the added or removed references.
10. Click **OK**.
11. To view details of the associated external WSDL for the service, click either of the two WSDL links at the botom of the form:
    a. View external WSDL implementation definition
    b. View external WSDL interface definition

    **Note:** To help your service users locate the WSDL documents for services that are deployed to the Web Services Gateway, the gateway also supports the **WS-Inspection specification** http://www.ibm.com/developerworks/webservices/library/ws-wsilspec.html . To open a WS-Inspection document which contains references to the WSDL documents of all of the gateway-deployed services, you issue an HTTP GET against

    `http://host:port/wsgw/wsinspection.wsil`

    where *host* and *port* are the host name and port number that your HTTP server is listening on.

### Related tasks

"Deploying Web services to the Web Services Gateway"

"Removing Web services from the Web Services Gateway" on page 44

"Working with Web services" on page 39

## Deploying Web services to the Web Services Gateway

Use this task to deploy a Web service to the Web Services Gateway.

Before you deploy a Web service, deploy the resources (channels, filters, and UDDI references) that the Web service uses.

To deploy a Web service, complete the following steps:

1. Chapter 5, “Administering the Web Services Gateway”, on page 13
2. In the navigation pane, click the following link:

   **Services**

   - Deploy

   A form is displayed for you to specify the deployment details.
3. Type the following Web service deployment details:
   a. **Gateway Service Name**. Type the name by which the Web service is known within the Web Services Gateway, and by which it is listed using the “Listing and managing gateway-deployed Web services” on page 40 option. This name must be unique within the Web Services Gateway and must not contain any spaces.

      **Note:** If you have several implementations of the same Web service, you can map them all to the same deployed gateway service. You use the **Deploy** option (this option) to deploy just one instance of a given gateway service. To add more target services to a deployed gateway service, you use the “Listing and managing gateway-deployed Web services” on page 40 option.
   b. Choose between the two types of **Message part representation**:

      **Generic classes**
      Select this option if there are no EJB or Java bindings, or if there are such bindings but they are not used at runtime.

      **Deployed Java classes**
      Select this option if the target services for the gateway service contain EJB bindings (as in the “Running the Calendar service sample” on page 50) or Java bindings (as in the “Running the Address Book service sample” on page 49 and the “Running the Stock Quote service sample” on page 47), or if the target services use the LFT channel.

      **Note:** When you choose this option, you must also ensure that the specific java classes that have been generated for the Web service are deployed to the application server (for example by copying the JAR file into the WebSphere Application Server’s `/AppServer/lib/app`directory, then restarting the application server).
   c. **Authorization Policy - Control access to this service**. If you want to Chapter 7, “Administering security for the Web Services Gateway”, on page 53 for this Web service, enable this check box.
   d. **Audit Policy - Log requests to this service**. If you want the WebSphere Business Connection Solution Manager to log requests and responses for this Web service, enable this check box.
   e. Select the deployed resources that the Web service is to use, from the following lists:

      **Channels**
      Select one or more deployed channels through which this service is to be available.

      **Pre-Filters**
      Select zero or more deployed filters to apply to the request.

      **Post-Filters**
      Select zero or more deployed filters to apply to the response.

**UDDI References**
Select zero or more deployed UDDI references to UDDI registries in which this service is to be registered.

**Note:**

There may be more than one UDDI Reference for the same business in the same UDDI Registry. This is because a business can have more than one entry (each with a different ″Authorized Name″) in the same UDDI registry. For more information see “Deploying UDDI references to the Web Services Gateway” on page 37.

f. Specify the **Target Service Properties** for the Web service:

**WSDL Location**
Type the location of the ″internal″ WSDL file that describes the Web service to be deployed. This value is either a URL or the UDDI lookup information in the form:

*uddi_reference_name*,*tmodel_key*,*key_name*,*key_value*

When the Web Services Gateway deploys the Web service, it generates a matching ″external″ WSDL file that it makes available to gateway users. This ″external″ WSDL file also describes the service, but is located at a new URL and is generated and maintained by the Web Services Gateway itself.

**Location Type**
Select the type of location (either URL or UDDI) where the ″internal″ WSDL file is held.

**Target Service Name**
If the Web service WSDL contains more than one service, type the target service’s name from the target service WSDL.

**Target Service Namespace**
If the Web service WSDL contains more than one service, type the namespace of the target service’s name from the target service WSDL.

**Target Service Identity Information**
Type the identity by which the Web service is known within the Web Services Gateway. This identity need not be unique.

**Note:** If you later add more target services to this gateway service, you can use “The WebSphere Business Connection routing filters” on page 23 to select a particular target service from the set.

g. If you want to publish the service to UDDI, specify the **UDDI Publication Properties**. If you have not selected any UDDI references (in step 4e above), leave these fields blank.

**Service Provider Name**
The name of the business that owns this service.

**Service Provider Description**
The description of the business that owns this service.

**TModel**
The abstract namespace reference that defines a template for the service.

**TModel Key Name**
The key name assigned to this tModel in the UDDI registry.

**TModel Key Value**
The unique key value assigned to this tModel in the UDDI registry.

4. Click **OK**.

**Results**

If the processing completes successfully, the list of deployed Web services is updated to include the new Web service. Otherwise, an error message is displayed.

**What next**

After deployment, use the "Listing and managing gateway-deployed Web services" on page 40 option to change the resources (channels, filters, and UDDI references) that the Web service uses, or to add multiple target services for this gateway service.

If you enabled the check box 'Authorization Policy - Control access to this service, you must now Chapter 7, "Administering security for the Web Services Gateway", on page 53.

### Related tasks

"Listing and managing gateway-deployed Web services" on page 40

"Removing Web services from the Web Services Gateway"

"Working with Web services" on page 39

## Removing Web services from the Web Services Gateway

Use this task to remove a Web service from the Web Services Gateway.

To remove a Web service, complete the following steps:

1. Chapter 5, "Administering the Web Services Gateway", on page 13
2. In the navigation pane, click the following link:
   **Services**
   - Remove

   The main pane is updated with a list of all the Web services that are deployed to the Web Services Gateway. To the left of each entry in the list is a check box.
3. Select the check box for every Web service that you want to remove.
4. Click **OK**.

**Results**

If the processing completes successfully, the list of deployed Web services is updated. Otherwise, an error message is displayed.

### Related tasks

"Listing and managing gateway-deployed Web services" on page 40

"Deploying Web services to the Web Services Gateway" on page 41

"Working with Web services" on page 39

# Chapter 6. Running the Web Services Gateway samples

There are three Web Services Gateway samples:

- The standard Stock Quote service sample, which requires an Internet connection.
- The Address Book service, which shows how services work that require complex types.
- The Calendar service, which is implemented by a stateless session enterprise bean.

You will use (at least) the following files:

- The source files for the samples in *WSGW_HOME*/samples.
- *WSGW_HOME*/lib/wsgw-samples.jar(used by the gateway samples clients).
- *WSGW_HOME*/lib/calendarservice-ejb.jar(used by the calendar service sample).

where *WSGW_HOME*is the root directory for your installation of the gateway.

To run the gateway samples, complete one or more of the following tasks:

1. "Running the Stock Quote service sample"
2. "Running the Address Book service sample" on page 49
3. "Running the Calendar service sample" on page 50

**What next**

In the instructions for running each of these samples, it is assumed that the WSDL file is stored locally. If you want to test the gateway taking service definitions from a private UDDI Registry such as the IBM WebSphere UDDI Registry (that is available for download **here** http://www7b.boulder.ibm.com/wsdd/downloads/UDDIregistry.html ), you should complete the following additional steps:

1. Publish the WSDL for each of these samples to UDDI. (For detailed information on how to do this, see the documentation for your private UDDI Registry).
2. Instruct the gateway to locate the service through the UDDI Registry, as described in "Deploying Web services to the Web Services Gateway" on page 41.

### Related tasks

Chapter 1, "Using the Web Services Gateway", on page 1

"Running the Stock Quote service sample"

"Running the Address Book service sample" on page 49

"Running the Calendar service sample" on page 50

## Running the Stock Quote service sample

Use this task to run the Stock Quote sample Web service

© Copyright IBM Corp. 2002

This standard Stock Quote sample Web Service requires an Internet connection, through which it gets the current stock price for a given company. In this example, you ask the Stock Quote service to return the current stock price for IBM stock. To run the Stock Quote sample, complete the following steps:

1. "Working with Web services" on page 39, specifying the following deployment details. Leave blank all the other fields in the form.

   **Gateway Service Name**
   Type `StockQuote`.

   **Message part representation**
   Select **Deployed Java classes**.

   **Location**
   Type the full path of `Stockquote.wsdl`that was supplied in the `/samples/services/stockquote/`directory.

   **Location Type**
   Select **URL**.

   **Channels**
   Select a channel.

2. Note details of the deployed Stock Quote sample:
   a. "Working with Web services" on page 39.
   b. Select the StockQuote service from the list.
   c. Click on the link to open the external WSDL file.
   d. Make a note of the SOAP address.
   e. If you have changed the Web Services Gateway "Setting the namespace URI and WSDL URI for the Web Services Gateway" on page 14, make a note of the method-namespace-uri.

      If you have not changed the Namespace URI, the method-namespace-uri is `urn:ibmwsgw#StockQuote`.

3. Make the Java bindings for the Stock Quote service available in the server by completing the following steps:
   a. Copy `wsgw-samples.jar` into the *WAS_HOME*`/lib/app`directory.
   b. Stop then restart the application server.

4. Run the sample client by completing the following steps:
   a. Make sure that your client's class path contains the following JAR files that are in the *WAS_HOME*`/lib/app`directory:
      - activation.jar
      - xerces.jar
      - mail.jar
      - soap.jar
      - wsgw-samples.jar

   b. (Unix systems) At a command line enter the following command:

      ```
      export PATH=WAS_HOME/java/bin:$PATH
      ```

      This guarantees that the IBM WebSphere Application Server JDK is being used for running the samples.

   c. At a command line enter the following command:

      ```
      java services.stockquote.GetQuote [soap-address] [method-namespace-uri]
      [action-uri] IBM
      ```

For example:

```
java services.stockquote.GetQuote http://host:port/wsgwsoap1/soaprpcrouter
urn:ibmwsgw#StockQuote "" IBM
```

where *host* and *port* are the host name and port number for the application server on which the gateway is installed.

### Related tasks

Chapter 6, "Running the Web Services Gateway samples", on page 47

"Running the Address Book service sample"

"Running the Calendar service sample" on page 50

"Setting the namespace URI and WSDL URI for the Web Services Gateway" on page 14

## Running the Address Book service sample

Use this task to run the Address Book sample Web service, which shows how services that require complex types work.

To run the Address Book sample, complete the following steps:

1. "Working with Web services" on page 39, specifying the following deployment details. Leave blank all the other fields in the form.

   **Gateway Service Name**
   Type `AddressBook`.

   **Message part representation**
   Select **Deployed Java classes**.

   **Location**
   Type the full path of `AddressBook.wsdl`that was supplied in the `/samples/services/addressbook/`directory.

   **Location Type**
   Select **URL**.

   **Channels**
   Select a channel.

2. Note details of the deployed Address Book sample:
   a. "Working with Web services" on page 39.
   b. Select the AddressBook service from the list.
   c. Click on the link to open the external WSDL file.
   d. Make a note of the SOAP address.
   e. If you have changed the Web Services Gateway "Setting the namespace URI and WSDL URI for the Web Services Gateway" on page 14, make a note of the method-namespace-uri.

      If you have not changed the Namespace URI, the method-namespace-uri is `urn:ibmwsgw#AddressBook`.

3. Make the Java bindings for the AddressBook service available in the server by completing the following steps:

a. Copy `wsgw-samples.jar` into the *WAS_HOME*`/lib/app`directory.
b. Stop then restart the application server.

4. Run the sample client by completing the following steps:
   a. Make sure that your client's class path contains the following JAR files that are in the *WAS_HOME*`/lib/app`directory:
      - activation.jar
      - xerces.jar
      - mail.jar
      - soap.jar
      - wsgw-samples.jar
   b. (Unix systems) At a command line enter the following command:

      ```
      export PATH=WAS_HOME/java/bin:$PATH
      ```

      This guarantees that the IBM WebSphere Application Server JDK is being used for running the samples.
   c. At a command line enter the following command:

      ```
      java services.addressbook.GetAddress [soap-address] [method-namespace-uri]
      [action-uri] "John B. Good"
      ```

      For example:

      ```
      java services.addressbook.GetAddress http://host:port/wsgwsoap1
      /soaprpcrouter urn:ibmwsgw#AddressBook "" "John B. Good"
      ```

      where *host* and *port* are the host name and port number for the application server on which the gateway is installed.

### Related tasks

Chapter 6, "Running the Web Services Gateway samples", on page 47

"Running the Stock Quote service sample" on page 47

"Running the Calendar service sample"

"Setting the namespace URI and WSDL URI for the Web Services Gateway" on page 14

## Running the Calendar service sample

Use this task to run the Calendar sample Web service, which is implemented by a stateless session EJB.

These instructions are specific to WebSphere Application Server. If you want to try this sample with another EJB container, the source for the enterprise bean is in the *WSGW_HOME*`/samples/services/calendar/ejb`directory. To run the Calendar sample in WebSphere Application Server, complete the following steps:

1. Deploy the Calendar service enterprise bean.

   The enterprise bean is in the JAR file `calendarservice-ejb.jar`. Use the WebSphere Application Server Administrative Console or the SEAppInstall tool to deploy the enterprise bean. Accept the default values for all of the

deployment information except for the **JNDI Name** (not the default datasource JNDI name), which you must set to `services/CalendarService`.

2. "Working with Web services" on page 39, specifying the following deployment details. Leave blank all the other fields in the form.

   **Gateway Service Name**
   Type `Calendar`

   **Message part representation**
   Select **Deployed Java classes**

   **Location**
   Type the full path of `Calendar.wsdl`that was supplied in the `/samples/services/calendar/`directory.

   **Location Type**
   Select **URL**.

   **Channels**
   Select a channel.

3. Note details of the deployed Calendar sample:
   a. "Working with Web services" on page 39.
   b. Select the Calendar service from the list.
   c. Click on the link to open the external WSDL file.
   d. Make a note of the SOAP address.
   e. If you have changed the Web Services Gateway "Setting the namespace URI and WSDL URI for the Web Services Gateway" on page 14, make a note of the method-namespace-uri.

      If you have not changed the Namespace URI, the method-namespace-uri is `urn:ibmwsgw#Calendar`.

4. Make the EJB bindings for the Calendar service available in the server by completing the following steps:
   a. Copy `wsgw-samples.jar` into the *WAS_HOME*`/lib/app`directory.
   b. Stop then restart the application server.

5. Run the sample client by completing the following steps:
   a. Make sure that your client's class path contains the following JAR files that are in the *WAS_HOME*`/lib/app`directory:
      - activation.jar
      - xerces.jar
      - mail.jar
      - soap.jar
      - wsgw-samples.jar
   b. (Unix systems) At a command line enter the following command:

      ```
      export PATH=WAS_HOME/java/bin:$PATH
      ```

      This guarantees that the IBM WebSphere Application Server JDK is being used for running the samples.
   c. At a command line enter the following command:

      ```
      java services.calendar.GetDayOfTheWeek [soap-address] [method-namespace-uri]
      [action-uri] 7 4 1776
      ```

      For example:

```
java services.calendar.GetDayOfTheWeek http://host:port/wsgwsoap1
/soaprpcrouter urn:ibmwsgw#Calendar "" 7 4 1776
```

where *host* and *port* are the host name and port number for the application server on which the gateway is installed.

## Related tasks

Chapter 6, “Running the Web Services Gateway samples”, on page 47

“Running the Stock Quote service sample” on page 47

“Running the Address Book service sample” on page 49

“Setting the namespace URI and WSDL URI for the Web Services Gateway” on page 14

# Chapter 7. Administering security for the Web Services Gateway

Use this task to administer security for the Web Services Gateway.

The Web Services Gateway provides a basic authentication and authorization mechanism based upon the security features of WebSphere Application Server. For WebSphere Business Connection, you implement gateway security as described in the ″Installation and Configuration Guide″.

Security can be applied at two levels:

1. Gateway-level authentication.
2. Operation-level authorization.

For **gateway-level authentication**, you set up a role and realm for the gateway on WebSphere Application Server's Web server and servlet container, and define the userid and password that is used by the gateway to access the role and realm. You also modify the gateway's channel applications so that they only give access to the gateway to service requestors that supply the correct userid and password for that role and realm.

**Note:** This means that gateway-level authentication must be enabled before you install any channels.

For **operation-level authorization**, you apply security to individual methods in a Web Service. To do this, you create an enterprise bean with methods matching the Web Service operations. These EJB methods perform no operation and are just entities for applying security. Existing WebSphere Application Server authentication mechanisms can be applied to the enterprise bean. Before any Web service operation is invoked, a call is made to the EJB method. If authorization is granted, the Web service is invoked. Your target Web service is protected by wrapping it in an EAR file, and applying role-based authorization to the EAR file. This process is explained in general terms in “Web service security - role-based authorization” on page 54

**Note:**

- If you want to enable operation-level authorization, you must first enable Gateway-level authentication.
- After gateway-level authentication has been enabled, filters have access to the requestor's authentication information.
- You can only apply operation-level authorization to a Web service that has already been deployed to the gateway with the check box 'Authorization Policy - Control access to this service' enabled.
- Enabling operation-level authorization involves making changes to the file /lib/wsgwauth.ear. To protect the installation version of this file, you should make a backup copy of it before you change it.
- All of the sample clients allow you to pass in two extra parameters, a userid and password, that are used in both levels of security.

© Copyright IBM Corp. 2002

The Web Services Gateway can also invoke web services that include https:// in their addresses, if the Java and WebSphere security properties have been configured to allow it as described in the WebSphere Business Connection "Installation and Configuration Guide".

**What next**

For hints on solving security-related problems, see Chapter 10, "Web Services Gateway troubleshooting tips", on page 81.

### Related tasks

Chapter 1, "Using the Web Services Gateway", on page 1

"Web service security - role-based authorization"

### Related reference

Chapter 10, "Web Services Gateway troubleshooting tips", on page 81

## Web service security - role-based authorization

During construction of an EAR file, roles can be defined and applied to methods. At deployment of the EAR file, individual users or groups can be assigned to roles. So you can use this feature of EAR files to add role-based security to your Web service.

For example: You have a Web service that controls access to important information, and you want to give read-only access to some users, and write access to others. So when you build the EAR file you define two roles READ and WRITE, then you apply the READ role to the `getData` method and the WRITE role to the `writeData`method. When you deploy the EAR file in WebSphere Application Server, you assign 'All Authenticated Users' to the READ role and individual users to the WRITE role. When a user tries to access `WebService.getData`, their user name and password is checked by the operating system or by Lightweight Third Party Authentication (LTPA).

### Related tasks

Chapter 7, "Administering security for the Web Services Gateway", on page 53

# Chapter 8. Transferring large files from gateway to gateway

If you want to enable Large File Transfer (LFT) support, complete the steps in this topic.

The following topics describe large file transfer, and explain how to move a large file between two gateways using two supplied example Web services.

- For a brief overview of what large file transfer is, and how it works, see "Large file transfer - large files copied reliably from gateway to gateway".
- For detailed instructions on using the LFT sample Web services to verify the LFT installation, see "Using large file transfer with the LFT sample" on page 56.
- If you want to use LFT over a secure http connection, see "Setting up secure large file transfer" on page 61.
- If you want to use the LFT sample Web services as a starting-point for building your own LFT Web services, additional information for developers is provided in "Large File Transfer - how the LFT sample works" on page 74.
- If a large file transfer is in an uncertain state, you can use the status utility to "Querying the LFT status file and deleting a failed entry" on page 77.

### Related concepts

"Large file transfer - large files copied reliably from gateway to gateway"

"Large File Transfer - how the LFT sample works" on page 74

### Related tasks

Chapter 1, "Using the Web Services Gateway", on page 1

"Using large file transfer with the LFT sample" on page 56

"Setting up secure large file transfer" on page 61

"Querying the LFT status file and deleting a failed entry" on page 77

## Large file transfer - large files copied reliably from gateway to gateway

A familiar weakness of the present Internet is that it is a poor medium for transferring large files. Specifically, when you try and download a large file from the Web you often find that the transfer fails, or takes too long.

Large file transfer (LFT) addresses these problems in the following ways:

- Large files are transferred over reliable HTTP (HTTPR).

© Copyright IBM Corp. 2002

- Large files are chunked before sending, then reassembled on arrival at the receiving gateway. This means that, should the connection be lost during transfer, HTTPR only has to re-start the transfer from the current chunk rather than re-send the whole file.
- Large files are sent independent of any business application (Web service) that uses them.

Large file transfer is conceived as a service to be used by other Web services. For example, your business might develop a Web service to locate a large database file and report on a specific aspect of the information in that database. To enable this Web service to respond to requests quickly and easily, you might take the following approach:

- Large file transfer is used to distribute copies of the database file, as and when it is updated, to all Web Services Gateways that are deployed within your enterprise.
- When the Web service is invoked, the gateways provide access to the file from local storage rather than across the Internet.

**Note:** "Using large file transfer with the LFT sample" does no more than verify your LFT setup. For more information on how to build upon LFT to meet your particular business needs, see the gateway topic"Large File Transfer - how the LFT sample works" on page 74, and the documentation for the WBC ″Document Exchange″ feature.

### Related concepts

"Large File Transfer - how the LFT sample works" on page 74

### Related tasks

Chapter 8, "Transferring large files from gateway to gateway", on page 55

"Using large file transfer with the LFT sample"

"Setting up secure large file transfer" on page 61

"Querying the LFT status file and deleting a failed entry" on page 77

## Using large file transfer with the LFT sample

To run the LFT sample, you need two computers - one to send a file, and one to receive it.

This task assumes that you have installed and configured, on both of these computers, an application server and an instance of the gateway as described in the WebSphere Business Connection ″Installation and Configuration Guide″, and summarised in Chapter 4, "Web Services Gateway - Completing the installation", on page 7.

This task describes how you move a large file between two gateways using two supplied sample Web services, which in turn use the Axis and LFT channels.

Repeat the steps described in this task for both computers. After they have been configured, each computer can act as both LFT server and LFT client.

All test clients are run from a command prompt in the *WSGW_HOME*/samples/services/lftfileservicedirectory. On the client side, make sure there is no `httpr.jar` file under *WAS_HOME*/java/jre/lib/ext.

The initial installation and configuration of the gateway that is described in the WebSphere Business Connection ″Installation and Configuration Guide″ includes creation of the following *IBM_Lft_Client_Files_Directory* directory structure:

- (Windows) `C:\IBM\lft\client\files`
- (AIX) `/usr/IBM/lft/client/files`
- (Other Unix platforms) `/opt/IBM/lft/client/files`

To use the LFT sample Web services to move a large file between two gateways, complete the following tasks:

1. “LFT sample: checking LFT files in and out locally (server side)” on page 58.
2. LFT sample: checking LFT files in and out remotely (from client to server) (“LFT sample: checking LFT files in and out remotely (from client to server)” on page 59).

**What next**

Additional information for developers is provided in “Large File Transfer - how the LFT sample works” on page 74.

## Related concepts

“Large file transfer - large files copied reliably from gateway to gateway” on page 55

“Large File Transfer - how the LFT sample works” on page 74

## Related tasks

Chapter 1, “Using the Web Services Gateway”, on page 1

Chapter 8, “Transferring large files from gateway to gateway”, on page 55

“LFT sample: checking LFT files in and out locally (server side)” on page 58

LFT sample: checking LFT files in and out remotely (from client to server) (“LFT sample: checking LFT files in and out remotely (from client to server)” on page 59)

“Setting up secure large file transfer” on page 61

“Querying the LFT status file and deleting a failed entry” on page 77

## LFT sample: checking LFT files in and out locally (server side)

To test that the server side is working, run the following two tests (**CHECK IN** and **CHECK OUT**) on each computer:

1. **CHECK IN**. To check in locally the file demo.htm from *WSGW_HOME*/samples/services/lftfileserviceto the server's temp directory *WSGW_HOME*/serverTemp, complete the following steps:
   a. Open a command prompt.
   b. Change directory to *WSGW_HOME*/samples/services/lftfileservice.
   c. Enter the following command:

      checkin wsif *WSGW_HOME full_path_name_for_xerces.jar*

      **Note:**
      - Enter the full path for *WSGW_HOME*
      - xerces.jar is usually in *WAS_HOME*/lib

      If the run is successful you will get the following output:

      ```
      .... Registering LFT Extn .....
      coming to register here ..

      Using 'HTTPRPort' port:
      CheckIn: Message : File Checked in fine
      ```

   d. Check the *WSGW_HOME*/serverTempdirectory to check that demo.htmhas arrived. Look for a file name similar to the following:

      ```
      151743184604.1022512233929.AXIS@<hostname>%2FC%3A%2FIBM%2Flft%2Fclient
      %2Ffiles%2Fdemo.htm
      ```

2. **CHECK OUT**. To check out locally the file demo.gif from *IBM_Lft_Client_Files_Directory*to the client's temporary directory *WSGW_HOME*/clientTemp, complete the following steps:
   a. Open a command prompt.
   b. Change directory to *WSGW_HOME*/samples/services/lftfileservice.
   c. Enter the following command:

      checkout wsif *WSGW_HOME full_path_name_for_xerces.jar*

      If the run is successful you will get something like the following output:

      ```
      Using 'HTTPRPort' port:
      CheckOut : fileReference :
      151743184604.1022512233929.AXIS@<hostname>1%2FC%3A%2FIBM%2F
      lft%2Fclient%2Ffiles%2Fdemo.gif
      CheckOut : fileReferenceURL :
      file:/D:/Downloads/wsgw_was4/clientTemp/1517431
      84604.1022512233929.AXIS@<hostname>%2FC%3A%2FIBM%2Flft%2Fclient%2
      Ffiles%2Fdemo.gif
      ```

   d. Check the *WSGW_HOME*/clientTempdirectory to check that demo.gifhas arrived. Look for a file name similar to the following:

      ```
      151743184604.1022512233929.AXIS@<hostname>%2FC%3A%2FIBM%2Flft%2Fc
      lient%2Ffiles%2Fdemo.gif
      ```

**What next**

You are now ready to LFT sample: checking LFT files in and out remotely (from client to server) ("LFT sample: checking LFT files in and out remotely (from client to server)" on page 59).

### Related concepts

"Large file transfer - large files copied reliably from gateway to gateway" on page 55

"Large File Transfer - how the LFT sample works" on page 74

### Related tasks

"Using large file transfer with the LFT sample" on page 56

LFT sample: checking LFT files in and out remotely (from client to server) ("LFT sample: checking LFT files in and out remotely (from client to server)")

## LFT sample: checking LFT files in and out remotely (from client to server)

In these two tests you move files between computers through the client gateway to the server gateway. To use the samples, the server computer set up does not need to be modified, but the client computer needs a change. So before you begin, complete the following steps:

1. Decide which computer is to be the server and which is the client.
2. On the client computer, open the `LFTSample.wsdl` that is in *WSGW_HOME*`/samples/services/lftfileservice`
3. In `LFTSample.wsdl`, find the line

   ```
   <lft:address location="httpr://localhost/wsgwlft1
   /HttprServer#HTTPR.DEMO.REQUEST"/>
   ```
4. Edit the URL in the location to point to your server computer (change *localhost* to your server computer host name and port number).
5. Save and close the WSDL file.

   **Note:** By modifying this WSDL file on each computer, you can set up either computer as client or server.
6. If the application server was running when you saved `LFTSample.wsdl`, stop the application server and restart it. This is necessary because the application server has saved the original WSDL file in its cache, and the server must be reloaded so that the new (correct) URL is used.

**Note:** As you experiment with the following two gateway tests, you might encounter problems with the saved client and server status files (both called `stateFile`) being out of sync with each other. If you have followed the directions and still have problems with the samples, stop both application servers, erase all copies of `stateFile`on both computers, restart the application servers, and try again. For more information, including the default locations of the client and server state files, see "Querying the LFT status file and deleting a failed entry" on page 77.

1. **CHECK IN**. To check in the file `demo.htm` from *WSGW_HOME*`/samples/services/lftfileservice`on the client computer to directory *WSGW_HOME*`/serverTemp`on the server computer, complete the following steps:

a. Open a command prompt on the client computer.
b. Change directory to *WSGW_HOME*/samples/services/lftfileservice.
c. Enter the following command:

   checkin wsgw *WSGW_HOME full_path_name_for_xerces.jar*

   **Note:**
   - Enter the full path for *WSGW_HOME*
   - xerces.jar is usually in *WAS_HOME*/lib

   If the run is successful you will get something like the following output, which uses a *WSGW_HOME* value of wsgw_was4:

```
Starting CheckIn...
installDir is now: /wsgw_was4
wsdlURL: file:/wsgw_was4/samples/services/lftfileservice/LFTSample.wsdl
Using 'HTTPRPort' port:
FileServicePort.class: interface com.fileService.www.FileServicePort
checkIn: File Name: file:/wsgw_was4/samples/services/lftfileservice/demo.htm
CheckIn: Message : File Checked in fine
```

d. Check the *WSGW_HOME*/serverTempdirectory on the server to check that demo.htmhas arrived. Look for a file name similar to the following:

```
151743184604.1022512233929.AXIS@<hostname>%2FC%3A%2FIBM%2Flft%2Fc
lient%2Ffiles%2Fdemo.htm
```

   **Note:** This file can be deleted when tests are complete.

2. **CHECK OUT**. To check out the file demo.gif from *IBM_Lft_Client_Files_Directory*on the server computer to directory *WSGW_HOME*/serverTempon the client computer, complete the following steps:
   a. Open a command prompt on the client computer.
   b. Change directory to *WSGW_HOME*/samples/services/lftfileservice.
   c. Enter the following command:

      checkout wsgw *WSGW_HOME full_path_name_for_xerces.jar*

      If the run is successful you will get something like the following output, which uses a *WSGW_HOME* value of wsgw_was4:

```
Using 'HTTPRPort' port:
CheckOut : fileReference :
151743184604.1022512233929.AXIS@<hostname>1%2FC%3A%2FIBM%2F
lft%2Fclient%2Ffiles%2Fdemo.gif
CheckOut : fileReferenceURL :
file:/wsgw_was4/serverTemp/1517431
84604.1022512233929.AXIS@<hostname>%2FC%3A%2FIBM%2Flft%2Fclient%2
Ffiles%2Fdemo.gif
```

   d. Check the *WSGW_HOME*/serverTempdirectory on the client computer to check that demo.gifhas arrived. Look for a file name similar to the following:

```
151743184604.1022512233929.AXIS@<hostname>%2FC%3A%2FIBM%2Flft%2Fc
lient%2Ffiles%2Fdemo.gif
```

      **Note:** This file can be deleted when tests are complete.

## Related concepts

"Large file transfer - large files copied reliably from gateway to gateway" on page 55

“Large File Transfer - how the LFT sample works” on page 74

Related tasks

“Using large file transfer with the LFT sample” on page 56

“LFT sample: checking LFT files in and out locally (server side)” on page 58

## Setting up secure large file transfer

To run the LFT sample, you need two computers - one to send a file, and one to receive it.

This task assumes that you have installed and configured, on both of these computers, an application server and an instance of the gateway as described in the WebSphere Business Connection ″Installation and Configuration Guide″, and summarised in Chapter 4, “Web Services Gateway - Completing the installation”, on page 7. If you have just worked through the task “Using large file transfer with the LFT sample” on page 56, then you already have two computers that are suitably configured to begin this task.

This task describes how you can extend the LFT configuration that is described and tested in the task “Using large file transfer with the LFT sample” on page 56 so that you can move a large file between two gateways over a secure HTTP connection.

In this task you set up a one-way secure connection, from a client gateway that (in this example) is hosted on a computer called ServerOne, to a server gateway that (in this example) is hosted on a computer called ServerTwo. You then test the connection using the same CHECK IN test that is used in LFT sample: checking LFT files in and out remotely (from client to server) (“LFT sample: checking LFT files in and out remotely (from client to server)” on page 59) to check in a file from ServerOne (client computer) to ServerTwo (server computer).

Secure LFT uses a protocol stack for the client/server connection of LFT/HTTPR/HTTPS/TCP-IP.

For a more through exploration of using secure HTTP, see the topic ″Configuring SSL in WebSphere Application Server″ in the IBM WebSphere Application Server InfoCenter that is available online at http://www-3.ibm.com/software/webservers/appserv/doc/v40/aee/index.html.

To set up and test one-way secure LFT, complete the following tasks:

1. “Secure LFT: Setting up HTTP servers for SSL” on page 62.
2. Secure LFT: Creating a client TrustStore and importing ServerOne’s certificate (“Secure LFT: Creating a client TrustStore and importing ServerOne’s certificate” on page 67).
3. Secure LFT: Creating a wsgw TrustStore and importing ServerTwo’s certificate (“Secure LFT: Creating a wsgw TrustStore and importing ServerTwo’s certificate” on page 68).
4. “Secure LFT: Configuring the application servers” on page 70.

5. “Secure LFT: Installing IBM Java Secure Socket Extension (JSSE) 1.0.2” on page 71.
6. “Secure LFT: Creating and enabling httpr.ini” on page 72.
7. “Secure LFT: Checking files in remotely (from client to server)” on page 73.

**What next**

**Note:** You can use these same techniques to set up a similar secure connection in the other direction (from ServerTwo to ServerOne). You can then run, over secure HTTP, all four of the tests that are described in “Using large file transfer with the LFT sample” on page 56.

### Related concepts

“Large file transfer - large files copied reliably from gateway to gateway” on page 55

“Large File Transfer - how the LFT sample works” on page 74

### Related tasks

Chapter 1, “Using the Web Services Gateway”, on page 1

Chapter 8, “Transferring large files from gateway to gateway”, on page 55

“Using large file transfer with the LFT sample” on page 56

“Secure LFT: Setting up HTTP servers for SSL”

Secure LFT: Creating a client TrustStore and importing ServerOne’s certificate (“Secure LFT: Creating a client TrustStore and importing ServerOne’s certificate” on page 67)

Secure LFT: Creating a wsgw TrustStore and importing ServerTwo’s certificate (“Secure LFT: Creating a wsgw TrustStore and importing ServerTwo’s certificate” on page 68)

“Secure LFT: Configuring the application servers” on page 70

“Secure LFT: Installing IBM Java Secure Socket Extension (JSSE) 1.0.2” on page 71

“Secure LFT: Creating and enabling httpr.ini” on page 72

“Secure LFT: Checking files in remotely (from client to server)” on page 73

“Querying the LFT status file and deleting a failed entry” on page 77

## Secure LFT: Setting up HTTP servers for SSL

SSL (Secure Socket Layer) is used by several WebSphere Application Server components in order to provide secure communication. In particular SSL is used

by HTTPS, the application server's built-in HTTPS transport. If your HTTP server is set up to use SSL, then when the you type `https://...`instead of `http://...`the browser creates an SSL connection instead of a simple TCP connection to the HTTP server. The browser then typically either prompts the user or fails to connect if it was unable to validate the Web server or to agree upon the level of security options (the strength of the encryption algorithm to use).

To set up your HTTP servers to use SSL, complete the following tasks for both ServerOne (client computer) and ServerTwo (server computer).

1. "Secure LFT: Creating a key database and self-signed certificate".
2. Either "Secure LFT: Editing httpd.conf (Windows systems)" on page 65, or "Secure LFT: Editing httpd.conf (Unix systems)" on page 66.

**What next**

You are now ready to Secure LFT: Creating a client TrustStore and importing ServerOne's certificate ("Secure LFT: Creating a client TrustStore and importing ServerOne's certificate" on page 67).

### Related tasks

Chapter 8, "Transferring large files from gateway to gateway", on page 55

"Setting up secure large file transfer" on page 61

"Secure LFT: Setting up HTTP servers for SSL" on page 62

"Secure LFT: Creating a key database and self-signed certificate"

"Secure LFT: Editing httpd.conf (Windows systems)" on page 65

"Secure LFT: Editing httpd.conf (Unix systems)" on page 66

Secure LFT: Creating a client TrustStore and importing ServerOne's certificate ("Secure LFT: Creating a client TrustStore and importing ServerOne's certificate" on page 67)

Secure LFT: Creating a wsgw TrustStore and importing ServerTwo's certificate ("Secure LFT: Creating a wsgw TrustStore and importing ServerTwo's certificate" on page 68)

"Secure LFT: Configuring the application servers" on page 70

"Secure LFT: Installing IBM Java Secure Socket Extension (JSSE) 1.0.2" on page 71

"Secure LFT: Creating and enabling httpr.ini" on page 72

"Secure LFT: Checking files in remotely (from client to server)" on page 73

## Secure LFT: Creating a key database and self-signed certificate

When SSL is enabled, an SSL key database file is required. This file contains the CA certificates (signer certificates) as well as any personal certificates.

To create a key database file and self-signed certificate using the ″ikeyman″ key management utility, complete the following steps for both ServerOne (client computer) and ServerTwo (server computer):

1. Create a directory called myKeys in *ibm_http_server_install_dir*, where *ibm_http_server_install_dir* is the directory in which the IBM http server is installed.
2. Start the ″ikeyman″ key management utility:
   - (Windows): Select **Start -> Programs -> IBM HTTP Server -> Start Key Management Utility**
   - (Unix): Start the program *ibm_http_server_install_dir*/ssl/ikeyman(for example /usr/IBMHTTPServer/ssl/ikeyman).
3. Select **Key Database File -> New**, then enter the following values in the new database file window:
   - **Key database type**: accept the default value CMS key database file
   - **File name:** type *server_name*.kdb, where *server_name* is either ServerOne or ServerTwo.
   - **Location:** type *ibm_http_server_install_dir*/myKeys
4. Click **OK**. The Password Prompt window opens.
5. In the Password Prompt window, complete the following steps:
   a. Enter a password and confirm the password.
   b. Enable the check box **Stash the password to a file?**
   c. Click **OK**.

   This creates a file named *server_name*.sthcontaining an encoded form of the password. Note that this encoding prevents a casual viewing of the password but is not highly secure. So operating system permissions should be used to prevent unauthorized access to this file.
6. An information window opens to tell you that the password has been encrypted and saved. Click **OK** to close this window.
7. Open the list box that currently shows **Signer Certificates**, then select **Personal Certificates**.
8. Click **New Self-Signed...**. The Create New Self-Signed Certificate window opens.
9. In this window, enter the following values (and accept any defaults that are already set for other fields):
   - **Key Label**: type a name to identify the key and certificate in the database (for example HTTPSelfSigned).
   - **Common Name**: ensure that this field contains the fully-qualified host name of the server (for example www.myserver.com).
   - **Organization**: type the name of the organization that owns this certificate (for example IBM).
   - Click **OK**.
10. Click **Extract Certificate**. The Extract Certificate to a File window opens.
11. In this window, enter the following values:
    - **Data type**: accept the default value Base64-encoded ASCII data
    - **Certificate file name:** type *server_name*HTTPSelfSigned.arm
    - **Location:** type *ibm_http_server_install_dir*/myKeys
    - Click **OK**.
12. Select **Key Database File -> Close**.

13. Close ikeyman

**What next**

If you have completed the previous steps for both ServerOne (client computer) and ServerTwo (server computer), you are now ready to either "Secure LFT: Editing httpd.conf (Windows systems)", or "Secure LFT: Editing httpd.conf (Unix systems)" on page 66.

## Related tasks

"Setting up secure large file transfer" on page 61

"Secure LFT: Setting up HTTP servers for SSL" on page 62

"Secure LFT: Editing httpd.conf (Windows systems)"

"Secure LFT: Editing httpd.conf (Unix systems)" on page 66

## Secure LFT: Editing httpd.conf (Windows systems)

This task assumes that you have already "Secure LFT: Creating a key database and self-signed certificate" on page 63 for both ServerOne (client computer) and ServerTwo (server computer).

In this task you add lines to file `httpd.conf`. The extra lines cause the Web server to listen on port 443 (the default SSL port).

For Windows systems, complete the following steps for both ServerOne (client computer) and ServerTwo (server computer):

1. Open file `httpd.conf`. This file is in directory *ibm_http_server_install_dir*/conf.
2. Append the following lines to the bottom of the file. Substitute actual values for *hostname*, *ibm_http_server_install_dir* and *server_name*:

```
########################
## Start of SSL Config
########################
#
LoadModule ibm_ssl_module modules/IBMModuleSSL128.dll
Listen 443
#
<VirtualHost hostname:443>
SSLEnable
SSLClientAuth none
</VirtualHost>
#
SSLDisable
Keyfile "ibm_http_server_install_dir/myKeys/server_name.kdb"
SSLV2Timeout 100
SSLV3Timeout 1000
#
########################
## End SSL sample config
########################
```

3. Stop then restart the web server.
4. Check your configuration by issuing this URL: `https://hostname`

Because you are using a self-signed certificate, instead of a certificate issued by a CA such as Verisign, your browser should prompt you to see if you want to trust the unknown signer of the server's certificate.

**What next**

If you have "Secure LFT: Setting up HTTP servers for SSL" on page 62 for both ServerOne and ServerTwo, you are now ready to Secure LFT: Creating a client TrustStore and importing ServerOne's certificate ("Secure LFT: Creating a client TrustStore and importing ServerOne's certificate" on page 67).

## Related tasks

"Setting up secure large file transfer" on page 61

"Secure LFT: Setting up HTTP servers for SSL" on page 62

"Secure LFT: Creating a key database and self-signed certificate" on page 63

"Secure LFT: Editing httpd.conf (Unix systems)"

## Secure LFT: Editing httpd.conf (Unix systems)

This task assumes that you have already "Secure LFT: Creating a key database and self-signed certificate" on page 63 for both ServerOne (client computer) and ServerTwo (server computer).

In this task you add lines to file `httpd.conf`. The extra lines cause the Web server to listen on port 443 (the default SSL port).

For Unix systems, complete the following steps for both ServerOne (client computer) and ServerTwo (server computer):

1. Open file `httpd.conf`. This file is in directory *ibm_http_server_install_dir*/conf.
2. Append the following lines to the bottom of the file. Substitute actual values for *hostname*, *ibm_http_server_install_dir* and *server_name*:

```
########################
## Start of SSL Config
########################
#
LoadModule ibm_ssl_module      libexec/mod_ibm_ssl_128.so
AddModule mod_ibm_ssl.c
#
Listen 80
Listen 443
#
<VirtualHost hostname:443>
ServerName hostname
DocumentRoot ibm_http_server_install_dir/htdocs/en_US
SSLEnable
SSLClientAuth none
</VirtualHost>
#
SSLDisable
Keyfile "ibm_http_server_install_dir/myKeys/server_name.kdb"
SSLV2Timeout 100
SSLV3Timeout 1000
```

```
#
########################
## End SSL sample config
########################
```

3. Stop then restart the web server.
4. Check your configuration by issuing this URL: `https://hostname`

   Because you are using a self-signed certificate, instead of a certificate issued by a CA such as Verisign, your browser should prompt you to see if you want to trust the unknown signer of the server's certificate.

**What next**

If you have "Secure LFT: Setting up HTTP servers for SSL" on page 62 for both ServerOne and ServerTwo, you are now ready to Secure LFT: Creating a client TrustStore and importing ServerOne's certificate ("Secure LFT: Creating a client TrustStore and importing ServerOne's certificate").

### Related tasks

"Setting up secure large file transfer" on page 61

"Secure LFT: Setting up HTTP servers for SSL" on page 62

"Secure LFT: Creating a key database and self-signed certificate" on page 63

"Secure LFT: Editing httpd.conf (Windows systems)" on page 65

## Secure LFT: Creating a client TrustStore and importing ServerOne's certificate

This task assumes that you have already "Secure LFT: Setting up HTTP servers for SSL" on page 62. On the client machine (ServerOne) complete the following steps:

1. Create a directory called `SSLClient` in *your_sslclient_root*, where *your_sslclient_root* is any directory you choose.
2. Start the IBM WebSphere Application Server ″ikeyman″ key management utility:
   - (Windows): Select **Start -> Programs -> IBM WebSphere -> Application Server V4 AE -> IKeyman**
   - (Unix): Start the program *ibm_websphere_application_server_install_dir*`/bin/./ikeyman.sh &`(for example `usr/WebSphere/AppServer/bin/./ikeyman.sh &`).
3. Select **Key Database File -> New**, then enter the following values in the new database file window:
   - **Key database type**: `JKS`
   - **File name:** type `clientkey.jks`.
   - **Location:** type *your_sslclient_root*`/SSLClient`
4. Click **OK**. The Password Prompt window opens.
5. In the Password Prompt window, complete the following steps:
   a. Enter a password and confirm the password.
   b. Click **OK**.

6. Select and delete each Signer Certificate.
7. Click **Add** to add the HTTP Server Certificate. Enter the following values:
   - **Data type**: accept the default value `Base64-encoded ASCII data`
   - **Certificate file name:** type `ServerOneHTTPSelfSigned.arm`
   - **Location:** type *`ServerOne_ibm_http_server_install_dir`*`/myKeys`
   - Click **OK**.
8. Select **Key Database File -> Close**.
9. Close ikeyman

**What next**

You are now ready to Secure LFT: Creating a wsgw TrustStore and importing ServerTwo's certificate ("Secure LFT: Creating a wsgw TrustStore and importing ServerTwo's certificate").

### Related tasks

Chapter 8, "Transferring large files from gateway to gateway", on page 55

"Setting up secure large file transfer" on page 61

"Secure LFT: Setting up HTTP servers for SSL" on page 62

Secure LFT: Creating a wsgw TrustStore and importing ServerTwo's certificate ("Secure LFT: Creating a wsgw TrustStore and importing ServerTwo's certificate")

"Secure LFT: Configuring the application servers" on page 70

"Secure LFT: Installing IBM Java Secure Socket Extension (JSSE) 1.0.2" on page 71

"Secure LFT: Creating and enabling httpr.ini" on page 72

"Secure LFT: Checking files in remotely (from client to server)" on page 73

## Secure LFT: Creating a wsgw TrustStore and importing ServerTwo's certificate

This task assumes that you have already Secure LFT: Creating a client TrustStore and importing ServerOne's certificate ("Secure LFT: Creating a client TrustStore and importing ServerOne's certificate" on page 67). On the client machine (ServerOne) complete the following steps:

1. Create a directory called `SSLwsgw` in *`your_sslwsgw_root`*, where *your_sslwsgw_root* is any directory you choose.
2. Copy `ServerTwoHTTPSelfSigned.arm`from ServerTwo (server computer) to a temp directory on ServerOne (client computer).
3. Start the IBM WebSphere Application Server ″ikeyman″ key management utility:
   - (Windows): Select **Start -> Programs -> IBM WebSphere -> Application Server V4 AE -> IKeyman**

- (Unix): Start the program *ibm_websphere_application_server_install_dir*/bin/./ikeyman.sh &(for example usr/WebSphere/AppServer/bin/./ikeyman.sh &).

4. Select **Key Database File -> New**, then enter the following values in the new database file window:
   - **Key database type**: JKS
   - **File name:** type wsgwkey.jks.
   - **Location:** type *your_sslwsgw_root*/SSLwsgw
5. Click **OK**. The Password Prompt window opens.
6. In the Password Prompt window, complete the following steps:
   a. Enter a password and confirm the password.
   b. Click **OK**.
7. Select and delete each Signer Certificate.
8. Click **Add** to add the HTTP Server Certificate. Enter the following values:
   - **Data type**: accept the default value Base64-encoded ASCII data
   - **Certificate file name:** type ServerTwoHTTPSelfSigned.arm
   - **Location:** type *full_path_to_temp_directory_on_ServerOne*
   - Click **OK**.
9. Type *ServerTwo_server_name*Certas the certificate label.
10. Click **OK**.
11. Select **Key Database File -> Close**.
12. Close ikeyman

**What next**

You are now ready to "Secure LFT: Configuring the application servers" on page 70.

## Related tasks

Chapter 8, "Transferring large files from gateway to gateway", on page 55

"Setting up secure large file transfer" on page 61

"Secure LFT: Setting up HTTP servers for SSL" on page 62

Secure LFT: Creating a client TrustStore and importing ServerOne's certificate ("Secure LFT: Creating a client TrustStore and importing ServerOne's certificate" on page 67)

"Secure LFT: Configuring the application servers" on page 70

"Secure LFT: Installing IBM Java Secure Socket Extension (JSSE) 1.0.2" on page 71

"Secure LFT: Creating and enabling httpr.ini" on page 72

"Secure LFT: Checking files in remotely (from client to server)" on page 73

## Secure LFT: Configuring the application servers

This task assumes that you have already Secure LFT: Creating a wsgw TrustStore and importing ServerTwo's certificate ("Secure LFT: Creating a wsgw TrustStore and importing ServerTwo's certificate" on page 68). To configure the application servers to use SSL, complete the following steps:

1. To configure the host to listen on port 443 (the SSL port), complete the following steps for both ServerOne (client computer) and ServerTwo (server computer):
   a. Start the IBM WebSphere Application Server Administrative Console.
   b. In the Navigation pane, expand **Virtual Hosts** then select the virtual host you are using.
   c. In the main pane, select the **General** tab.
   d. On the **General** tab, click **Add**.
   e. Type `*:443`then click **Apply**.
   f. In the Navigation pane, from the pop-up menu for your node, click **Regen Webserver Plugin**.
   g. Stop then restart the application server.
   h. Check your configuration by issuing this URL: `https://hostname/wsgw`
2. To configure the JVM settings, complete the following steps **for ServerOne (client computer) only**:
   a. Select the JVM settings page for the application server.
   b. Add the following properties:
      - httpr.ssl=yes
      - httpr.keystore=SSLwsgw/wsgwkey.jks
      - httpr.passphrase=*your_wsgwkey_password*
      - httpr.channelStyle=process
   c. Click **Apply**.
   d. Stop then restart the application server.

**What next**

You are now ready to "Secure LFT: Installing IBM Java Secure Socket Extension (JSSE) 1.0.2" on page 71.

### Related tasks

Chapter 8, "Transferring large files from gateway to gateway", on page 55

"Setting up secure large file transfer" on page 61

"Secure LFT: Setting up HTTP servers for SSL" on page 62

Secure LFT: Creating a client TrustStore and importing ServerOne's certificate ("Secure LFT: Creating a client TrustStore and importing ServerOne's certificate" on page 67)

Secure LFT: Creating a wsgw TrustStore and importing ServerTwo's certificate ("Secure LFT: Creating a wsgw TrustStore and importing ServerTwo's certificate" on page 68)

"Secure LFT: Installing IBM Java Secure Socket Extension (JSSE) 1.0.2"

"Secure LFT: Creating and enabling httpr.ini" on page 72

"Secure LFT: Checking files in remotely (from client to server)" on page 73

## Secure LFT: Installing IBM Java Secure Socket Extension (JSSE) 1.0.2

This task assumes that you have already "Secure LFT: Configuring the application servers" on page 70. To install IBM Java Secure Socket Extension (JSSE) 1.0.2, complete the following steps **for ServerOne (client computer) only**:

1. Stop the admin server.
2. Check that there is a copy of file `ibmjsse.jar` in directory *WAS_HOME*`/java/jre/lib/ext`(where *WAS_HOME* is the root directory for your installation of IBM WebSphere Application Server).
3. If your java client uses the system JVM and not WebSphere's JVM, also copy jre/lib/ext/ibmjsse.jar to *system_JVM_home*/jre/lib/ext
4. Start the admin server.
5. Edit the security properties file *WAS_HOME*`/java/jre/lib/security/java.security`so that it includes entries for both the Sun security provider and the IBM security provider. For example:

   ```
   security.provider.1=sun.security.provider.Sun
   security.provider.2=com.ibm.jsse.IBMJSSEProvider
   ```

   **Note:** The order is significant. The Sun security provider must come before the IBM provider.
6. If your java client uses the system JVM and not WebSphere's JVM, also apply the edit described in the previous step to *system_JVM_home*`/jre/lib/security/java.security`

**What next**

You are now ready to "Secure LFT: Creating and enabling httpr.ini" on page 72.

### Related tasks

Chapter 8, "Transferring large files from gateway to gateway", on page 55

"Setting up secure large file transfer" on page 61

"Secure LFT: Setting up HTTP servers for SSL" on page 62

Secure LFT: Creating a client TrustStore and importing ServerOne's certificate ("Secure LFT: Creating a client TrustStore and importing ServerOne's certificate" on page 67)

Secure LFT: Creating a wsgw TrustStore and importing ServerTwo's certificate ("Secure LFT: Creating a wsgw TrustStore and importing ServerTwo's certificate" on page 68)

"Secure LFT: Configuring the application servers" on page 70

"Secure LFT: Creating and enabling httpr.ini"

"Secure LFT: Checking files in remotely (from client to server)" on page 73

## Secure LFT: Creating and enabling httpr.ini

This task assumes that you have already "Secure LFT: Installing IBM Java Secure Socket Extension (JSSE) 1.0.2" on page 71. To create an httpr.ini file, then make the CHECK IN test aware of it, complete the following steps **for ServerOne (client computer) only**:

1. Use a text editor to open a new text file.
2. Copy the following text into this empty file. Substitute actual values for *your_clientkey_password* and *WSGW_HOME*:

   ```
   httpr.ssl=yes
   httpr.keystore=SSLClient//clientkey.jks
   httpr.passphrase=your_clientkey_password
   httpr.channelStyle=process
   lft-directory=WSGW_HOME//serverTemp
   lft-piFactory=com.ibm.httpr.file.FileStateManagerFactory
   ```

3. Save this file with a filename of `httpr.ini`, in directory *WSGW_HOME*`/samples/services/lftfileservice`
4. Use a text editor to open file checkin.bat (Windows systems), or checkin.sh (Unix systems). This file is in directory *WSGW_HOME*`/samples/services/lftfileservice`.
5. In this file, find the `:goodparams`routine. Add the following parameter as the first parameter of the `@java`command:

   ```
   -Dhttpr.propsFile=httpr.ini
   ```

6. Save and close the file.

**What next**

You are now ready to "Secure LFT: Checking files in remotely (from client to server)" on page 73.

### Related tasks

Chapter 8, "Transferring large files from gateway to gateway", on page 55

"Setting up secure large file transfer" on page 61

"Secure LFT: Setting up HTTP servers for SSL" on page 62

Secure LFT: Creating a client TrustStore and importing ServerOne's certificate ("Secure LFT: Creating a client TrustStore and importing ServerOne's certificate" on page 67)

Secure LFT: Creating a wsgw TrustStore and importing ServerTwo's certificate ("Secure LFT: Creating a wsgw TrustStore and importing ServerTwo's certificate" on page 68)

"Secure LFT: Configuring the application servers" on page 70

"Secure LFT: Installing IBM Java Secure Socket Extension (JSSE) 1.0.2" on page 71

"Secure LFT: Checking files in remotely (from client to server)"

## Secure LFT: Checking files in remotely (from client to server)

This task assumes that you have already "Secure LFT: Creating and enabling httpr.ini" on page 72.

In the CHECK IN test you move a file between computers through the client gateway to the server gateway. To use the LFT samples over HTTPS, the ServerTwo (server computer) set up does not need to be modified, but the ServerOne (client computer) needs one more change. So before you begin, complete the following steps:

1. On ServerOne (client computer), open the `LFTSample.wsdl`that is in *WSGW_HOME*`/samples/services/lftfileservice`
2. In `LFTSample.wsdl`, find the line

   ```
   <lft:address location="httpr://localhost/wsgwlft1
   /HttprServer#HTTPR.DEMO.REQUEST"/>
   ```
3. Edit the URL in the location to point to your server computer and the SSL port (change *localhost* to *ServerTwo_server_name*`:443`).
4. Save and close the WSDL file.
5. If the application server was running when you saved `LFTSample.wsdl`, stop the application server and restart it. This is necessary because the application server has saved the original WSDL file in its cache, and the server must be reloaded so that the new (correct) URL is used.

In this task you test the HTTPS connection using the same CHECK IN test that is used in LFT sample: checking LFT files in and out remotely (from client to server) ("LFT sample: checking LFT files in and out remotely (from client to server)" on page 59) to check in a file from ServerOne (client computer) to ServerTwo (server computer).

**Note:** As you experiment with this test, you might encounter problems with the saved client and server status files (both called `stateFile`) being out of sync with each other. If you have followed the directions and still have problems with the samples, stop both application servers, erase all copies of `stateFile`on both computers, restart the application servers, and try again. For more information, including the default locations of the client and server state files, see "Querying the LFT status file and deleting a failed entry" on page 77.

To check in the file `demo.htm` from *WSGW_HOME*`/samples/services/lftfileservice`on ServerOne (client computer) to directory *WSGW_HOME*`/serverTempon` ServerTwo (server computer), complete the following steps:

1. Open a command prompt on ServerOne (client computer).
2. Change directory to *WSGW_HOME*`/samples/services/lftfileservice`.
3. Enter the following command:

   ```
   checkin wsgw WSGW_HOME full_path_name_for_xerces.jar
   ```

   **Note:**

   - Enter the full path for *WSGW_HOME*
   - xerces.jar is usually in *WAS_HOME*`/lib`

   If the run is successful you will get something like the following output, which uses a *WSGW_HOME* value of `wsgw_was4`:

```
Starting CheckIn...
installDir is now: /wsgw_was4
wsdlURL: file:/wsgw_was4/samples/services/lftfileservice/LFTSample.wsdl
Using 'HTTPRPort' port:
FileServicePort.class: interface com.fileService.www.FileServicePort
checkIn: File Name: file:/wsgw_was4/samples/services/lftfileservice/demo.htm
CheckIn: Message : File Checked in fine
```

4. Check the *WSGW_HOME*/serverTempdirectory on ServerTwo (server computer) to check that demo.htmhas arrived. Look for a file name similar to the following:

   ```
   151743184604.1022512233929.AXIS@<hostname>%2FC%3A%2FIBM%2Flft%2Fc
   lient%2Ffiles%2Fdemo.htm
   ```

   **Note:** This file can be deleted when tests are complete.

### Related tasks

Chapter 8, “Transferring large files from gateway to gateway”, on page 55

“Setting up secure large file transfer” on page 61

“Secure LFT: Setting up HTTP servers for SSL” on page 62

Secure LFT: Creating a client TrustStore and importing ServerOne’s certificate (“Secure LFT: Creating a client TrustStore and importing ServerOne’s certificate” on page 67)

Secure LFT: Creating a wsgw TrustStore and importing ServerTwo’s certificate (“Secure LFT: Creating a wsgw TrustStore and importing ServerTwo’s certificate” on page 68)

“Secure LFT: Configuring the application servers” on page 70

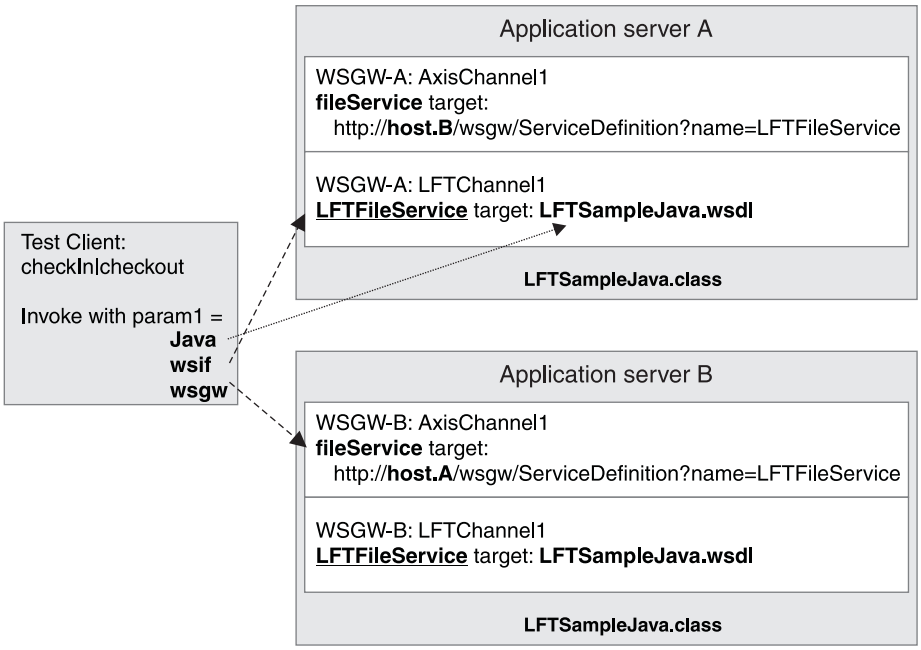
“Secure LFT: Installing IBM Java Secure Socket Extension (JSSE) 1.0.2” on page 71

“Secure LFT: Creating and enabling httpr.ini” on page 72

## Large File Transfer - how the LFT sample works

The diagram below shows how the LFT sample is deployed, configured and demonstrated.

The scenario shown uses two computers designated as A and B. Each computer has IBM WebSphere Application Server installed on it. To demonstrate this scenario, install and configure the LFT sample as described in “Using large file transfer with the LFT sample” on page 56.




Clients use wsif to obtain their wsdl from different places depending on param1:

**java**: Uses LFTSampleJava.wsdl directly. The LFTSampleJava.class must be on the client classpath to use this invocation.

**wsif**: Uses LFTSample.wsdl from the local machine. This points by default to LFTChannel1 on localhost.

**wsgw**: Uses wsdl obtained from the AxisChannel; for example, http://host.B/wsgw/ServiceDefinition?name=fileService

The rest of this topic gives more information on how each component of the LFT sample works.

### The LFT Sample Web service

There is only one Web service used in this scenario. It is implemented with a Java class named **LFTSampleJava.class**. The source code for this service is provided. For `checkIn`, the class just records that it was called by writing to the application server `System.out`log file. It does nothing that is specific to large file transfer. For `checkOut`, the class returns (as a string) the file URL of a file that is to be transferred from the client to the computer on which the service is deployed.

There is WSDL supplied for LFTSampleJava that provides Java class bindings for the service. Edit `LFTSampleJava.wsdl`to see these bindings. WSIF is able to invoke the class using the WSDL and its Java class bindings as long as the WSIF client and the LFTSampleJava.class are in the same JVM classloader. Because LFTSampleJava.class is included in the file `WSGW-Samples.jar`, and the gateway set-up requires you to copy the samples jar file into *`WAS_HOME`*`/lib/app`, the LFTSampleJava.class is in the classpath for any application server that is running on the computer.

### The LFTFileService gateway service

When you deploy `LFTSampleJava.wsdl`into the gateway, a gateway service named LFTFileService is created with a target service described by `LFTSampleJava.wsdl`. The deployment instructions specify that this service is created on LFTChannel1.

After the service is deployed on the LFTChannel on one of the computers, the WSDL for the service is provided from URL **http://hostname/wsgw/ServiceDefinition?name=LFTFileService** http://hostname/wsgw/ServiceDefinition?name=LFTFileService .

**The fileService gateway service**

You create a gateway service named fileService available on AxisChannel1 that uses the WSDL available from the URL shown for the WSDL of the LFTFileService. The WSDL for the fileService is provided from URL **http://hostname/wsgw/ServiceDefinition?name=fileService** http://hostname/wsgw/ServiceDefinition?name=fileService .

To test lFT between two gateways, you deploy the fileService and the LFTFileService on different computers, with the fileService using the WSDL available for the LFTFileService.

**The checkIn and checkOut sample clients**

The samples include two client scenarios named checkIn and checkOut.

**checkIn**

The checkIn sample is invoked as a batch file that calls the main() method in the provided CheckIn class. This class accepts an argument from the command line that specifies how the client invokes the LFTSampleJava Web service. Study the diagram above to see the behavior when this argument is `java`, `wsif` or `wsgw`.

When checkIn runs with the `java`argument, the LFTSampleJava.wsdl is used to call the service without using the gateway at all.

When checkIn runs with the `wsif`argument, the LFTSample.wsdl is used to call the service. This WSDL can be edited to have the URL of an LFTChannel. Initially it uses LFTChannel1 on localhost. The message that is sent to the LFTChannel includes a file reference on the local file system where the client is running. The target host's LFTChannel and the client use HTTPR to transfer the file from the current directory of the client process to the directory specified in the WSGW system property named `lft-directory`. The checkIn sample transfers the file named demo.htm in the lftsamples directory.

When checkIn runs with the `wsgw`argument, WSDL obtained from the AxisChannel for the fileService is used to call the service. The message is forwarded by the receiving gateway to the LFTFileService on the other computer. The message that is sent to the LFTChannel initiates file transfer as described in the previous paragraph for the `wsif`argument.

**checkOut**

The checkOut sample is invoked as a batch file that calls the main() method in the provided CheckOut class. This class accepts an argument from the command line that specifies how the client invokes the LFTSampleJava Web service. Study the diagram above to see the behavior when this argument is `java`, `wsif` or `wsgw`.

When checkOut runs with the `java`argument, the LFTSampleJava.wsdl is used to call the service without using the gateway at all.

When checkOut runs with the `wsif`argument, the LFTSample.wsdl is used to call the service.

When checkOut runs with the `wsgw`argument, WSDL obtained from the AxisChannel for the fileService is used to call the service.

### Related concepts

"Large file transfer - large files copied reliably from gateway to gateway" on page 55

### Related tasks

Chapter 8, "Transferring large files from gateway to gateway", on page 55

"Using large file transfer with the LFT sample" on page 56

"Setting up secure large file transfer" on page 61

"Querying the LFT status file and deleting a failed entry"

## Querying the LFT status file and deleting a failed entry

If a large file transfer is in an uncertain state, you can use the **status** utility to query the status file, and to cancel file transfers.

Status information about large file transfers is written to a status file called `stateFile`. By default, the application server `stateFile` is located in *WAS_HOME*`/bin`, where *WAS_HOME* is the root directory for your installation of IBM WebSphere Application Server (you can also find out the working directory using the WebSphere Application Server Administrative Console - look on the **General** tab for the application server in which your gateway is running). The client `stateFile`is located wherever the transfer was initiated, for example `wsgw_was4/samples/services/lftfileservice`.

The utility script `status`is included in the LFT File Service sample directory.

**Note:** This utility is provided for use by system administrators only. Do not delete files with names that you do not recognise, because these files might be system-generated chunks of a large file that is being transferred (and that will be reassembled into a single file at the receiving Gateway).

To use the **status** utility, complete the following steps:

1. Open a command prompt.
2. Change to the LFT sample directory (for example `wsgw_was4/samples/services/lftfileservice`).
3. Enter the following command:
   - (Windows) `status.bat` *WSGW_HOME stateFile_dir optional_parameter*.
   - (Unix) `./status.sh` *WSGW_HOME stateFile_dir optional_parameter*.

where

- *WSGW_HOME* is the root directory for your installation of the gateway.
- *stateFile_dir* is the directory where the httpr status file `stateFile`is held.
- *optional_parameter* is `cancel`, or no parameter is typed.

**Results**

If you do not type an *optional_parameter*, then the contents of the state file is listed out in the command window. If you type `cancel`, then the contents of the state file is displayed item by item, and you choose whether or not to delete each item.

## Related concepts

"Large file transfer - large files copied reliably from gateway to gateway" on page 55

"Large File Transfer - how the LFT sample works" on page 74

## Related tasks

Chapter 8, "Transferring large files from gateway to gateway", on page 55

"Using large file transfer with the LFT sample" on page 56

"Setting up secure large file transfer" on page 61

# Chapter 9. Handling exceptions for the Web Services Gateway

During normal processing of a Web service invocation, a fault message might be generated by the target service, and is passed back to the channel to be sent to the originator. As far as the Web service Gateway is concerned there is no difference in processing a normal output message or a fault message.

But when an exception occurs during processing of a request, the channel needs some way to decide what to do with the exception.

The WebSphere Business Connection Exception Handler is called by the Web Services Gateway when an internal exception occurs. This exception handler looks at the message, exception and other information and decides whether the exception should be thrown back to the originator, or whether a fault message should be constructed. It also reports the exceptions for analysis.

## Related tasks

Chapter 1, “Using the Web Services Gateway”, on page 1

© Copyright IBM Corp. 2002

# Chapter 10. Web Services Gateway troubleshooting tips

This topic provides hints to help you resolve problems you experience when using the Web Services Gateway.

To identify and resolve gateway-related problems, you can use the standard WebSphere Application Server "Enabling trace in WebSphere Application Server" on page 82 facilities. If you encounter a problem that you think might be related to the gateway, you can check for error messages in the WebSphere Application Server Administrative Console, and in the application server's `stdout.log`file. You can also enable the application server debug trace to provide a detailed exception dump.

A list of the gateway runtime system messages, with details of what each message means, is given in "Message reference for the Web Services Gateway" on page 83.

Here is a checklist of common problems:

- You have managed to deploy your Web Service in the Web Services Gateway but you are getting a class cast exception when you invoke the operation which takes an integer parameter.

  Check that your client is using the version of `soap.jar` that is supplied in the WebSphere Application Server's `/AppServer/lib/app`directory. If you "Enabling trace in WebSphere Application Server" on page 82, you may see in the trace for the request:`<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance" xmlns:xsd="http://www.w3.org/1999/XMLSchema">`

  The gateway expects the 2001 version of the XML schema. Older versions of soap.jar (including 2.2) generate 1999 schema. If you have the soap.jar that is supplied with WebSphere Application Server in the client's classpath, you should see 2001 schema in the request, which should then work.
- The persistent state of the Web Services Gateway has become out of sync with the channel applications.

  This can happen if you remove and reinstall the Apache SOAP applications. If you need to do this, then either ensure that all corresponding channels configured with the Web Services Gateway are removed, or remove and reinstall the Web Services Gateway at the same time.

  **Note:** The Web Services Gateway application (wsgw.ear) must be installed before channel and filter applications. If the gateway application needs to be reinstalled, all channels and filters must be uninstalled first, then reinstalled after the gateway application.
- You are getting SOAP fault messages for invalid parameter types, but cannot determine the precise problem from the fault message.

  If you receive a SOAP fault message with a faultstring which is just the value of one of the parameters of the invocation, that means that that parameter's value was invalid. For example if you have a service which expects an `int`parameter and you send it a message containing the value "1.1", then the fault message you receive simply contains `1.1` as the fault string:

  ```
  <faultcode>SOAP-ENV:Client</faultcode>
  <faultstring>1.1</faultstring>
  ```

© Copyright IBM Corp. 2002

**Note:** This is Apache SOAP behavior, and not something that the gateway can do anything about.

- You are Chapter 7, "Administering security for the Web Services Gateway", on page 53, but when you install `wsgwauth.ear`, an error message appears in the WebSphere Application Server Administrative Console detailing a JNDI problem.

  Check that you entered, in the authorization session bean's 'EJB References', the correct JNDI name of the imported Web service enterprise bean. Note that this is case sensitive.
- You are trying to have a Web service Gateway send a SOAP/HTTPS message to another Web service Gateway, and you are receiving a Malformed URLException error.

  The Web Services Gateway can invoke web services that include https:// in their addresses, if the Java and WebSphere security properties have been configured to allow it as described in the WebSphere Business Connection "Installation and Configuration Guide".
- You deselect 'Authorization Policy - Control access to this service' from the deployment details for a Web service, and you find the service no longer works.

  A number of tasks are required to disable security. Clearing the check box 'Authorization Policy - Control access to this service' will still leave WebSphere Application Server security in place, so basic authentication might still be required. To disable security fully, use the WebSphere Application Server Administrative Console.
- You are Chapter 8, "Transferring large files from gateway to gateway", on page 55, and you suspect that one or more transfers might have failed to complete.

  If a large file transfer is in an uncertain state, you can use the LFT status utility to "Querying the LFT status file and deleting a failed entry" on page 77.

### Related tasks

Chapter 1, "Using the Web Services Gateway", on page 1

"Enabling trace in WebSphere Application Server"

### Related reference

"Message reference for the Web Services Gateway" on page 83

## Enabling trace in WebSphere Application Server

To enable trace in the application server, complete the following steps:

1. Start the WebSphere Application Server adminstration server.
2. Start the administrative console.
3. In the navigation pane, expand **WebSphere Administrative Domain**.
4. Expand the **Nodes** tree, then click on your application server.

   This displays the properties of your application server in the contents pane.
5. In the contents pane, select the **Services** tab.

6. In the list of services, select **Trace Service** then click **Edit Properties**.
7. This displays the Trace Service window.
8. In the Trace Service window, click **...** at the right end of the **Trace Specification** field.

   This displays the Trace window. This provides a tree of components and groups.
9. From the pop-up menu for each component or group that you want to trace, select the trace level.
10. After you have set trace for all the components or groups that you want to trace, click **OK**. The Trace window closes.
11. In the Trace Service window, select the **Trace output file: Specify** radio button, then type the filename and path that you want trace output to go to.
12. Click **OK**. The Trace Service window closes.
13. In the content pane of the administrative console, click **Apply**.
14. From the **Console** menu, click **Console -> Trace -> Enabled**.
15. Stop then restart the application server.

### Related tasks

Chapter 10, “Web Services Gateway troubleshooting tips”, on page 81

### Related reference

“Message reference for the Web Services Gateway”

## Message reference for the Web Services Gateway

WebSphere system messages are logged from a variety of sources, including application server components and applications. Messages logged by application server components and associated IBM products start with a unique message identifier that indicates the component or application that issued the message. The message identifier can be either 8 or 9 characters in length and has the form:

*CCCC1234X*

where

- *CCCC*is a 4 character alphabetic component or application identifier (WSGWfor the gateway).
- *1234*is a unique numeric identifier used to identify the specific message within that component.
- *X* is an optional alphabetic severity indicator (I=Informational, W=Warning, E=Error).

The rest of this topic contains a list of the Web Services Gateway runtime system messages, with details of what each message means.

**WSGW0001E: Channel name {0} from gateway configuration differs from that in JNDI: {1}**

**Explanation**
The name specified for the channel does not match the name of the channel as defined within the EAR file.

**User Action**
Ensure that the channel name is specified correctly

**WSGW0002E: Error storing endpoint address. Exception: {0}**

**Explanation**
An unexpected exception occurred when storing the endpoint address for a channel.

**User Action**
Contact IBM Support

**WSGW0003E: Error retrieving endpoint address. Exception: {0}**

**Explanation**
An unexpected exception occurred when retrieving the endpoint address for a channel.

**User Action**
Contact IBM Support

**WSGW0004E: Not used**

**Explanation**

**User Action**

**WSGW0005E: Error retrieving channel name. Exception: {0}**

**Explanation**
An unexpected exception occurred when retrieving the channel name from JNDI.

**User Action**
Contact IBM Support

**WSGW0006E: Error deploying service to {1}. Exception: {0}**

**Explanation**
An unexpected error occurred deploying the service to the given component.

**User Action**
This error may be caused by a previous failure. Try redeploying the service using a different gateway service name. If that fails, reinstalling the channel and gateway applications may remove the problem.

**WSGW0007E: Error getting endpoint URL from channel {0}. Exception: {1}**

**Explanation**
An unexpected error occurred generating the endpoint URL for the given channel.

**User Action**
Contact IBM Support

**WSGW0008E: Could not determine default port name for target service {0}**

**Explanation**
There are no ports in the WSDL defined for the target service that are supported by currently available WSIF providers.

**User Action**
Either ensure that a WSIF provider is correctly configured for the port in the WSDL, or ensure that the WSDL contains correctly specified port information.

**WSGW0009E: Failed to deploy service. Exception: {0}**

**Explanation**
An unexpected error occurred trying to deploy the service.

**User Action**
Contact IBM Support

**WSGW0010E: The namespaceURI attribute cannot be changed when there are active services**

**Explanation**
The namespaceURI is used to generate WSDL for Gateway services. If this global setting is changed then current WSDL becomes invalid.

**User Action**
Either remove all channels or all Gateway services from the Gateway configuration and retry the change.

**WSGW0011E: Not used**

**Explanation**

**User Action**

**WSGW0012E: Not used**

**Explanation**

**User Action**

**WSGW0013E: Could not locate home {0}. Exception: {1}**

**Explanation**
The specified home location could not be found in JNDI.

**User Action**

Ensure that the home location is specified correctly, and that it appears in JNDI.

**WSGW0014E: Not used**

**Explanation**

**User Action**

**WSGW0015E: Could not create instance of class {0}. Exception: {1}**

**Explanation**

The Gateway failed to create an instance of the specified Java class.

**User Action**

Ensure that the Java class has a public constructor with no parameters.

**WSGW0016E: Could not locate class {0}. Exception: {1}**

**Explanation**

The Gateway failed to locate the specified Java class.

**User Action**

Ensure that the Java class is visible to the Gateway application's classloader.

**WSGW0017E: Not used**

**Explanation**

**User Action**

**WSGW0018E: Not used**

**Explanation**

**User Action**

**WSGW0019E: Failed to clone definition. Exception: {1}**

**Explanation**

An unexpected error occurred cloning a WSDL definition.

**User Action**

Contact IBM Support

**WSGW0020E: Error while loading mapped type class {0}. Exception: {1}**

**Explanation**

An error occurred while trying to load the given Java class which represents a type in the deployed WSDL for a target service.

**User Action**

Ensure that the Java class is visible to the Gateway application's classloader.

**WSGW0021E: Expected WSDL definition to contain a <wsdl:type> element with a schema from one of the '{0}', '{1}', or '{2}' namespaces**

**Explanation**

Schema types in WSDL definitions must be declared using one of the specified XML Schema namespaces.

**User Action**

Update the WSDL definition to use the appropriate namespace.

**WSGW0022E: Unexpected Schema->Java problem when parsing WSDL file. Exception: {0}**

**Explanation**

An unexpected exception occurred when parsing a WSDL file. This may be due to unsupported elements in the WSDL.

**User Action**

Contact IBM Support

**WSGW0023E: Unexpected Schema->JROM problem when parsing WSDL file. Exception: {0}**

**Explanation**

An unexpected exception occurred when parsing a WSDL file. This may be due to unsupported elements in the WSDL.

**User Action**

Contact IBM Support

**WSGW0024E: Channel {0} cannot be removed because it is being used by a deployed service**

**Explanation**

Channels can only be removed when they are not in use by Gateway services.

**User Action**

Remove the channel from gateway services to which it is deployed before removing the channel.

**WSGW0025E: Target service identity cannot be specified as null**

**Explanation**

A target service can only be selected using a non-null valid for the identity.

**User Action**

Modify the calling code to ensure that the target service identity value is never null.

**WSGW0026E: Invalid gateway service name {0}. The name must be a valid XML Schema NCNAME.**

**Explanation**

The name specified for the Gateway service does not conform to the required definition.

**User Action**

Correct the Gateway service name so that it is a valid XML Schema NCNAME.

**WSGW0027E: Port {0} does not exist for target service {1}**

**Explanation**

The requested port does not exist for the target service.

**User Action**

Ensure that a valid port is requested, or update the target service WSDL to contain a port of the requested name.

**WSGW0028E: No binding for port {0} for target service {1}**

**Explanation**

The requested port for the target service does not have a binding defined in the WSDL definition of the service.

**User Action**

Ensure that the target service WSDL has a binding for the requested port, or use a different port name.

**WSGW0029E: No portType for binding {0} for port {1} for target service {2}**

**Explanation**

The requested port for the target service does not have a portType defined in the WSDL definition of the service.

**User Action**

Ensure that the target service WSDL has a portType for the requested port, or use a different port name.

**WSGW0030E: Not used**

**Explanation**

**User Action**

**WSGW0031E: Channel name {0} already exists**

**Explanation**

The name specified for the channel is the same as that of a channel that is currently deployed.

**User Action**

Choose a different name for the channel, or remove the existing channel of the given name.

**WSGW0032E: Channel name {0} not found**

**Explanation**
No channel is currently deployed with the given name.

**User Action**
Use the name of a channel that is currently deployed.

**WSGW0033E: Filter {0} cannot be removed because it is being used by a deployed service**

**Explanation**
Filters can only be removed when they are not in use by Gateway services.

**User Action**
Remove the filter from gateway services to which it is deployed before removing the filter.

**WSGW0034W: Invocation of filter {0} failed. Exception: {1}**

**Explanation**
An unexpected exception was thrown during processing of the given filter.

**User Action**
Contact IBM Support

**WSGW0035E: Filter context version {0} not supported**

**Explanation**
The context version that the filter requires is not supported by this version of the Gateway.

**User Action**
Ensure that the filter is requesting the correct context version. It may be necessary to upgrade the Gateway to support the filter.

**WSGW0036E: Target service identity information {0} not matched for gateway service {1}**

**Explanation**
A target service was requested by identity, but the identity information does not match any currently deployed target service.

**User Action**
Ensure that the identity information is correct, and that there is a target service deployed to the given gateway service with the right identity information.

**WSGW0037E: Filter name {0} already exists**

**Explanation**
The name specified for the filter is the same as that of a filter that is currently deployed.

**User Action**

Choose a different name for the filter, or remove the existing filter of the given name.

**WSGW0038E: Filter name {0} not found**

**Explanation**

No filter is currently deployed with the given name.

**User Action**

Use the name of a filter that is currently deployed.

**WSGW0039E: Error loading state from {0}. Exception {1}**

**Explanation**

An unexpected exception occurred loading the state of the Gateway from the given location.

**User Action**

Ensure that the given location is visible to the Gateway application.

**WSGW0040E: Failed to convert definition to string. Exception: {0}**

**Explanation**

An unexpected exception occurred converting a WSDL definition into a string in order to display it or return it to an application.

**User Action**

Contact IBM Support

**WSGW0041E: Failed to save state. Exception {0}**

**Explanation**

An unexpected exception occurred when saving the state of the Gateway.

**User Action**

Contact IBM Support

**2W_key=WSGW0042W: No target services available to get service definition**

**Explanation**

A request was made for the WSDL definition for the Gateway service, however no target services have been defined for the Gateway service, so it is not possible to generate a WSDL definition.

**User Action**

Deploy one or more target services to the Gateway service.

**WSGW0043E: Exception while executing operation {0} service {1}. Exception: {2}**

**Explanation**

An unexpected exception occurred when passing a request on to a target web service.

**User Action**

Ensure that the Gateway service and target service are correctly deployed (using the correct message part representation). Ensure that the target service is available and responds correctly to direct requests (i.e. not through the Gateway).

**WSGW0044E: Filter position {0} invalid**

**Explanation**

The specified position for addition or removal of the filter was not valid.

**User Action**

Ensure a valid value is specified. The value should be -1, 0 or a positive integer.

**WSGW0045E: Filter not found in list**

**Explanation**

An attempt was made to remove a filter from a Gateway service specifying -1 as the index, but the filter is not in the list at all.

**User Action**

Ensure that the correct filter is specified.

**WSGW0046E: Channel {0} already defined for gateway service {1}**

**Explanation**

The given channel has already been defined for the Gateway service.

**User Action**

Ensyre that the correct channel name is specified.

**WSGW0047E: Channel {0} not defined for gateway service {1}**

**Explanation**

The channel cannot be removed from the Gateway service as it is not currently defined for the Gateway service.

**User Action**

Ensure that the correct channel name is specified.

**WSGW0048E: UDDI reference {0} already defined for gateway service {1}**

**Explanation**

The given UDDI reference has already been defined for the Gateway service.

**User Action**

Ensure that the correct UDDI reference name is specified.

**WSGW0049E: UDDI reference {0} not defined for gateway service {1}**

**Explanation**
The UDDI reference cannot be removed from the Gateway service as it is not currently defined for the Gateway service.

**User Action**
Ensure that the correct UDDI reference name is specified.

**WSGW0050E: Target service with location {0} already defined for gateway service {1}**

**Explanation**
The given target service location has already been defined for the Gateway service.

**User Action**
Ensure that the correct target service location is specified.

**WSGW0051E: Target service with location {0} not defined for gateway service {1}**

**Explanation**
The target service location cannot be removed from the Gateway service as it is not currently defined for the Gateway service.

**User Action**
Ensure that the correct target service location is specified.

**WSGW0052E: Target service with location {0} was not found for gateway service {1}**

**Explanation**
The target service WSDL definition could not be obtained from the given location.

**User Action**
Ensure that the correct target service location is specified.

**WSGW0053E: Gateway service {0} cannot be removed as active entities and force not specified**

**Explanation**
A Gateway service with one or more target services, channels, filters or UDDI references cannot be removed.

**User Action**
Remove the target services, channels, filters and UDDI references from the gateway service.

**WSGW0054E: An exported definition for Gateway service {0} is not available as there are no defined channels for the service**

**Explanation**
A request was made for the WSDL definition for the Gateway service, however no channels have been defined for the Gateway service, so it is not possible to generate a WSDL definition.

**User Action**

Deploy one or more channels to the Gateway service.

**WSGW0055E: Not used**

**Explanation**

**User Action**

**WSGW0056E: No default target service available for {0}**

**Explanation**

The default target service location cannot be obtained for the Gateway service as no target services are defined.

**User Action**

Ensure that one or more target services are defined for the Gateway service.

**WSGW0057E: No receiving channel name in context**

**Explanation**

A request has reached the Gateway that does not contain the receiving channel name in the context.

**User Action**

Contact the supplier of the channel application.

**WSGW0058E: Channel {0} not defined for gateway service {1}**

**Explanation**

A request has reached the gateway for the given service through a channel which is not defined for that service. The request is rejected.

**User Action**

If the channel should be valid for the service, add the channel, otherwise check that the client of the request is making a valid request. This exception may be thrown when a client is making a malicious attack.

**WSGW0059E: Gateway service {0} does not exist**

**Explanation**

A request was made for a gateway service that does not exist.

**User Action**

Ensure that the correct gateway service name is specified.

**WSGW0060E: Gateway service {0} already exists**

**Explanation**

An attempt was made to create a new Gateway service using a name that is used by an existing Gateway service.

**User Action**

Use a different name for the Gateway service.

**WSGW0061E: Could not find Service in UDDI registry {0} with parameters {1}, {2}, {3}**

**Explanation**

The given parameters for UDDI lookup did not yield a match.

**User Action**

Ensure that the parameters are correct. Also ensure that the UDDI reference parameters are correct and correspond to those used to publish the service to UDDI.

**WSGW0062E: Target service WSDL contains no <service> elements**

**Explanation**

The target service WSDL could be loaded but does not contain a <service> element. This is necessary to be able to invoke the target service.

**User Action**

Ensure that the target service WSDL contains one or more <service> element.

**WSGW0063E: Target service WSDL contains more than one service, and either target service name or namespace not specified**

**Explanation**

When adding a target service to a Gateway service, you must specify both the service name and namespace values if there is more than one <service> element in the target service WSDL.

**User Action**

Specify the target service name and namespace as well as the location.

**WSGW0064E: Target service name {0} does not match service name in WSDL: {1}**

**Explanation**

A target service name was specified that is not the same as any target service name in the WSDL at the given location.

**User Action**

Ensure that a valid target service name is specified.

**WSGW0065E: Target service namespace {0} does not match service namespace in WSDL: {1}**

**Explanation**

A target service namespace was specified that is not the same as any target service namespace in the WSDL at the given location.

**User Action**

Ensure that a valid target service namespace is specified.

**WSGW0066E: Target service name {0} or namespace {1} not found in WSDL definition**

**Explanation**

A target service name and namespace were both specified, but do not match any target service name and namespace combination in the WSDL at the given location.

**User Action**

Ensure that a valid target service name and namespace combination is specified.

**WSGW0067E: UDDI reference {0} cannot be removed because it is being used by a deployed service**

**Explanation**

UDDI references can only be removed when they are not in use by Gateway services.

**User Action**

Remove the UDDI reference from gateway services to which it is deployed before removing the UDDI reference.

**WSGW0068E: UDDI reference {0} already exists**

**Explanation**

The name specified for the UDDI reference is the same as that of a UDDI reference that is currently deployed.

**User Action**

Choose a different name for the UDDI reference, or remove the existing UDDI reference of the given name.

**WSGW0069E: UDDI reference {0} not found**

**Explanation**

No UDDI reference is currently deployed with the given name.

**User Action**

Use the name of a UDDI reference that is currently deployed.

**WSGW0070E: Invalid target service location type {0}**

**Explanation**

The location type for the target service is not a valid value.

**User Action**

Ensure that a correct value is specified for the target service location type.

**WSGW0071E: Failed to load URL definition from {0}**

**Explanation**

The URL location specified was incorrect, or the WSDL it refers to cannot be loaded.

**User Action**

Ensure that the URL location is correct, and refers to a valid WSDL document.

**WSGW0072E: Failed to load UDDI definition from {0}**

**Explanation**

The UDDI location specified was incorrect, or the WSDL it refers to cannot be loaded.

**User Action**

Ensure that the UDDI location is correct, and refers to a valid WSDL document.

**WSGW0073W: Not used**

**Explanation**

**User Action**

**WSGW0074E: Not used**

**Explanation**

**User Action**

**WSGW0075E: Failed to set gateway end point address. Exception {0}**

**Explanation**

An unexpected exception occurred when automatically setting the Gateway's end point address.

**User Action**

Contact IBM Support

**WSGW0076E: Unable to access the Gateway configuration bean. Exception {0}**

**Explanation**

An unexpected exception occurred looking up the Gateway's configuration bean in JNDI.

**User Action**

Restart the application server.

**WSGW0077E: Failed to remove Gateway configuration session. Exception {0}**

**Explanation**

An unexpected exception occurred removing the session bean while access the Gateway's configuration bean.

**User Action**

Contact IBM Support

**WSGW0078E: Unable to access the Gateway EndPoint bean. Exception {0}**

**Explanation**
An unexpected exception occurred looking up the Gateway's endpoint bean in JNDI.

**User Action**
Restart the application server.

**WSGW0079E: Failed to remove endpoint session. Exception {0}**

**Explanation**
An unexpected exception occurred removing the session bean while access the Gateway's endpoint bean.

**User Action**
Contact IBM Support

**WSGW0080E: Performance monitoring error. Exception {0}**

**Explanation**
An unexpected exception occurred when recording performance monitoring information.

**User Action**
Contact IBM Support

**WSGW0081E: Unexpected error in method {0}. Exception {1}**

**Explanation**
An unexpected exception occurred in the given method.

**User Action**
Contact IBM Support

**WSGW0082E: Unable to determine WAS security setting**

**Explanation**
The WAS security setting could not be determined. It will be assumed that security is enabled.

**User Action**
No action required.

**WSGW0083W: Failed to authorise request for operation {0} on service {1}. Exception {2}**

**Explanation**
Authorization of the given request failed. The request has been rejected.

**User Action**
Ensure that the required authorization bean has been generated for the given service, and that the correct authorization policy is defined.

**WSGW0084W: Invocation of filter {0} version {1} failed. Exception {2}**

**Explanation**
An exception was thrown during processing of the given filter. Processing of the request continues.

**User Action**
Investigate the reason for the exception being thrown by the filter. Refer to the documentation for the filter on how to resolve the problem.

**WSGW0085E: Failed to publish service {0} to UDDI registry {1}. Exception: {2}**

**Explanation**
An unexpected exception occurred when publishing the given service to a UDDI registry.

**User Action**
Ensure that the properties of the Gateway service and UDDI reference are specified correctly.

**WSGW0086E: Failed to unpublish service {0} from UDDI registry {1}. Exception: {2}**

**Explanation**
An unexpected exception occurred when unpublishing the given service from a UDDI registry.

**User Action**
Ensure that the properties of the Gateway service and UDDI reference are specified correctly.

**WSGW0087I: Published service {0} to UDDI registry {1}**

**Explanation**
The service was successfully published to the UDDI registry.

**User Action**
None

**WSGW0088I: Unpublished service {0} from UDDI registry {1}**

**Explanation**
The service was successfully unpublished from the UDDI registry.

**User Action**
None

**WSGW0089I: No MessageWarehouse registered. Requests will not be logged**

**Explanation**
A MessageWarehouse implementation was not found at the expected location in JNDI, so none is being used.

**User Action**
If a MessageWarehouse has been implemented, ensure that it is bound to JNDI at the correct location.

**WSGW0090I: No ExceptionHandler registered. Exceptions will not be handled**

**Explanation**

An ExceptionHandler implementation was not found at the expected location in JNDI, so none is being used.

**User Action**

If an ExceptionHandler has been implemented, ensure that it is bound to JNDI at the correct location.

**WSGW0091I: Usage: java -jar GenAuth -DWAS_HOME=<was.install.directory> <HostName> <ServiceName>**

```
where <was.install.directory> is the location of the
WebSphere installation directory
and <HostName> is the url pointed to the installation of the gateway
and <ServiceName> is the name of the deployed gateway service.
(Please note the ServiceName is case sensitive).
For example
java -jar GenAuth.jar -DWAS_HOME=c:\\websphere\\AppServer
    http://host.machine.name.com/wsgw ServiceName

Successful execution will generate a file named <ServiceName>.ear
```

**Explanation**

Usage statement. This messsage is used by the command line utility.

**User Action**

No action required.

**WSGW0092I: Retrieving Service :**

**Explanation**

Progress message indicating that the service definition is being retrieved. This messsage is used by the command line utility.

**User Action**

No action required.

**WSGW0093I: Retrieving Port Type :**

**Explanation**

Progress message indicating that the port type information is being retrieved. This messsage is used by the command line utility.

**User Action**

No action required.

**WSGW0094I: Retrieving Methods :**

**Explanation**

Progress message indicating that method information is being retrieved. This messsage is used by the command line utility.

**User Action**

No action required.

**WSGW0095I: Making Directory :**

**Explanation**
Progress message indicating that a directory is being created. This messsage is used by the command line utility.

**User Action**
No action required.

**WSGW0096I: Using Directory :**

**Explanation**
Progress message indicating that a directory is being used. This messsage is used by the command line utility.

**User Action**
No action required.

**WSGW0097I: About to compile....**

**Explanation**
Progress message indicating that a compilation is about to start. This messsage is used by the command line utility.

**User Action**
No action required.

**WSGW0098I: Command Status :**

**Explanation**
General command status message. This messsage is used by the commandline utility.

**User Action**
No action required.

**WSGW0099I: About to create jar....**

**Explanation**
Progress message indicating that a JAR file is about to be created. This messsage is used by the command line utility.

**User Action**
No action required.

**WSGW0100I: About to create ear....**

**Explanation**
Progress message indicating that an EAR file is about to be created. This messsage is used by the command line utility.

**User Action**
No action required.

**WSGW0101E: Error retrieving port from service {1}**

**Explanation**

An error occurred retrieving the port from the service in the WSDL. This messsage is used by the command line utility.

**User Action**

Ensure that the service name is specified correctly and is deployed to the Gateway with at least one target service and one channel.

**WSGW0102E: Error retrieving service {0}**

**Explanation**

An error occurred retrieving the service. This messsage is used by the commandline utility.

**User Action**

Ensure that the service name is specified correctly and is deployed to the Gateway with at least one target service and one channel.

**WSGW0103E: Exception while retrieving service definition from URL: {0}/ServiceDefinition?name={1}. Exception: {2}**

**Explanation**

An unexpected exception occurred retrieving WSDL from the given location. This messsage is used by the command line utility.

**User Action**

Ensure that the service name is specified correctly and is deployed to the Gateway with at least one target service and one channel.

**WSGW0104E: Error retrieving methods from service {0}**

**Explanation**

An unexpected exception occurred retrieving the methods that correspond to operations on the service.

**User Action**

Contact IBM Support

**WSGW0105E: Error retrieving WAS_HOME environment variable**

**Explanation**

The value of the WAS_HOME environment variable could not be retrieved.

**User Action**

Ensure that the WAS_HOME variable is set correctly in the environment under which the command is being executed.

**WSGW0106E: Error compiling files**

**Explanation**

An unexpected error occurred compiling the generated Java files.

**User Action**

Contact IBM Support

**WSGW0107E: Error executing JAR command**

**Explanation**
An unexpected error occurred generating a JAR file.

**User Action**
Contact IBM Support

**WSGW0110E: A client attempted to load imported URL {0} for gateway service {1}. This URL is not imported by the definition for that service.**

**Explanation**
An attempt was made to use the Gateway's import mapping servlet to load information from a URL that does not correspond to one that is referenced by the WSDL definition for that service.

**User Action**
Ensure that the client is making a valid request. This may be a malicious attempt to obtain information that the client does not have access to.

**WSGW0111W: Unsupported elements within the WSDL definition for target service {0} were ignored. The functionality of this service may be compromised.**

**Explanation**
In order to be able to use the given WSDL definition within the Gateway, certain elements of the definition were ignored.

**User Action**
Refer to the service provider's documentation to determine whether this will affect the use of the service.

**WSGW0112E: Exception while sending response to queue {0}. Exception {1}, linked exception {2}**

**Explanation**
An unexpected exception occurred during execution of the JMS channel.

**User Action**
Ensure that the JMS channel and its prerequisites are correctly configured. Examine the linked exception for information on how to resolve the problem.

**WSGW0113E: Exception while creating Object Message to send response. Exception {1}, linked exception {2}**

**Explanation**
An unexpected exception occurred during execution of the JMS channel.

**User Action**
Ensure that the JMS channel and its prerequisites are correctly configured. Examine the linked exception for information on how to resolve the problem.

**WSGW0114E: Exception while creating Queue Sender to send response. Exception {1}, linked exception {2}**

**Explanation**

An unexpected exception occurred during execution of the JMS channel.

**User Action**

Ensure that the JMS channel and its prerequisites are correctly configured. Examine the linked exception for information on how to resolve the problem.

**WSGW0115E: Exception while creating Reply To Queue to send response. Exception {1}, linked exception {2}**

**Explanation**

An unexpected exception occurred during execution of the JMS channel.

**User Action**

Ensure that the JMS channel and its prerequisites are correctly configured. Examine the linked exception for information on how to resolve the problem.

**WSGW0116E: Exception while creating Queue Session to send response. Exception {1}, linked exception {2}**

**Explanation**

An unexpected exception occurred during execution of the JMS channel.

**User Action**

Ensure that the JMS channel and its prerequisites are correctly configured. Examine the linked exception for information on how to resolve the problem.

**WSGW0117E: Exception while creating Queue Connection Factory to send response. Exception {1}, linked exception {2}**

**Explanation**

An unexpected exception occurred during execution of the JMS channel.

**User Action**

Ensure that the JMS channel and its prerequisites are correctly configured. Examine the linked exception for information on how to resolve the problem.

**WSGW0118E: End point address keyword {0} is not valid for the JMS channel**

**Explanation**

The keyword specified in the end point address for the JMS channel is not valid.

**User Action**

Consult the documentation for a list of valid keywords for the JMS channel end point address.

**WSGW0119E: End point address for the JMS channel must be keyword=value pairs separated by the '&' character**

**Explanation**
The end point address specified does not have the correct syntax.

**User Action**
Consult the documentation for a specification of the syntax of the JMS channel end point address.

**WSGW0120E: Exception while removing ConversationPart {0} from Correlation Service. Exception {1}**

**Explanation**
An unexpected exception occurred when using the Correlation Service.

**User Action**
Contact IBM Support

**WSGW0121E: Exception while accessing ConversationPart {0} from Correlation Service. Exception {1}**

**Explanation**
An unexpected exception occurred when using the Correlation Service.

**User Action**
Contact IBM Support

**WSGW0122E: Exception while storing Serializable {0} at Correlation Service. Exception {1}**

**Explanation**
An unexpected exception occurred when using the Correlation Service.

**User Action**
Contact IBM Support

**WSGW0123E: Exception while storing Serializable {0} with id {1} at Correlation Service. Exception {1}**

**Explanation**
An unexpected exception occurred when using the Correlation Service.

**User Action**
Contact IBM Support

## Related tasks

Chapter 10, “Web Services Gateway troubleshooting tips”, on page 81

“Enabling trace in WebSphere Application Server” on page 82

# Notices

IBM may not offer the products, services, or features discussed in this document in all 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 give 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.

**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 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:

WebSphere Business Connection Lab Director
IBM RTP Laboratory
3039 Cornwallis Road
P.O. BOX 12195

© Copyright IBM Corp. 2002

Raleigh, NC 27709-2195
U.S.A

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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 may 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.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates 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. 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 IBM's application programming interfaces.

## Programming interface information

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

General-use programming interfaces allow you to write application software that obtain 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 and service marks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM
alphaWorks
AIX
CrossWorlds
DB2
DB2 OLAP Server
DB2 Universal Database
DeveloperWorks
MQSeries
SecureWay
WebSphere

Lotus is a trademark of International Business Machines Corporation and Lotus Development Corporation in the United States, other countries, or both.

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

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

MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States, other countries, or both.

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