CHAPTER 16

Working With JMS Traffic

This chapter discusses how to use the ACE XML Gateway with traffic in the form of Java Message Service (JMS), a Java-based technology for exchanging messages between hosts on a network. It covers these topics:

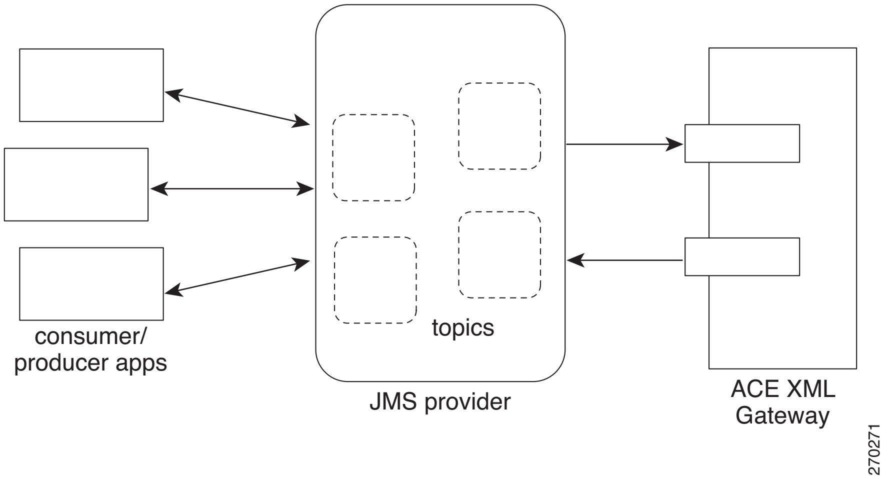
About JMS Support in the ACE XML Gateway

JMS integration capabilities are provided by the ACE XML Gateway through an optional, separately installed Gateway extension. If the extension is installed, you can use the ACE XML Gateway to manage, route, and secure JMS traffic.

Note If the JMS extension module is installed on your system, a JMS option appears in the protocol menu when creating policy handlers. If this option does not appear and you would like to use JMS routing, contact your Cisco ACE XML Gateway support representative.

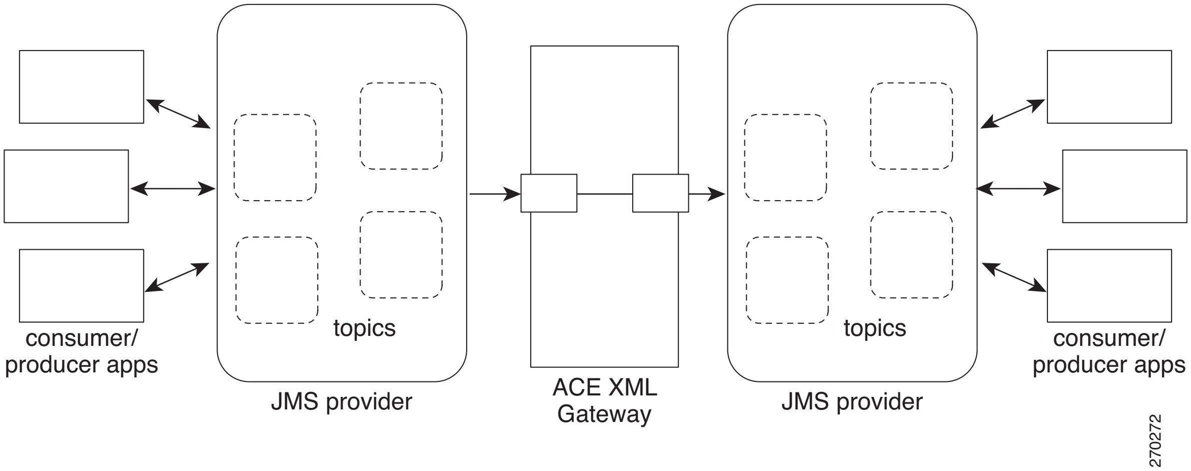
There are a number of ways in which the ACE XML Gateway can be configured to interact with JMS providers. The Gateway can receive messages by listening on a particular queue or topic, or publish messages to a JMS queue or topic. By receiving a message from one queue, validating it, and then publishing it to another, you can effect message validation.

Figure 16-1 Gateway to JMS



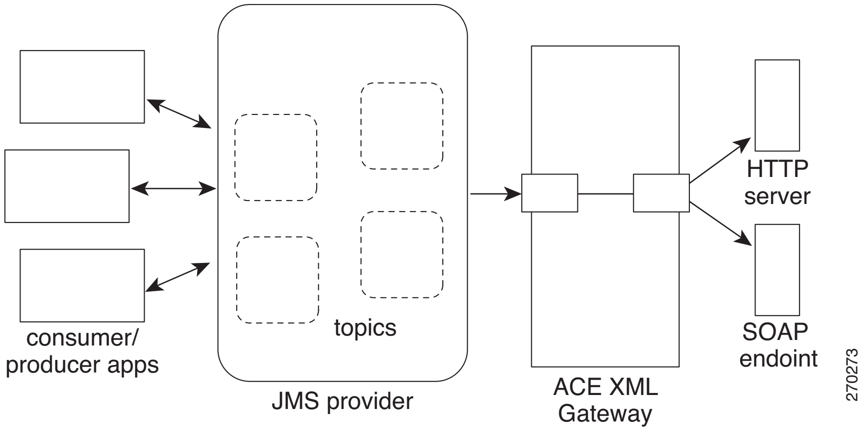
The ACE XML Gateway can also act as an integration point between separate JMS systems, as shown in the following figure.

Figure 16-2 Gateway to JMS



The ACE XML Gateway can mediate between disparate formats, for example, by converting a JMS message to a SOAP or HTTP message or vice versa. In this case, the ACE XML Gateway is situated between the disparate network entities.

Figure 16-3 JMS Mediation



A JMS message can have the following parts:

- A required header includes information on the destination, message ID, timestamp, and so on, as a set of name/value pairs.
- Optional properties are application-specific name/value pairs (supplemental to the name/value pairs of the header). You can have the ACE XML Gateway validate the properties in the message, for example, by requiring that a property is present in a message or by checking the property content or type.
- An optional payload (or body) can contain various types of data, including name/value pairs, text content (such as XML), or un-interpreted bytes. If the payload contains XML text, the ACE XML Gateway can verify or apply XML signatures on the message, encrypt or decrypt elements using XML encryption, transform the content, and more.

The ACE XML Gateway supports WebSphere MQ (MQSeries) and TIBCO Enterprise Message Service. Further, with the I/O SDK, the JMS support can be extended to work with other JMS implementations as well.

Setting Up JMS Overview

The general steps for setting up the ACE XML Gateway for JMS traffic are similar to setting up other types of protocols:

1. Configure the messaging server connection information in the policy (as a messaging server object).
2. Create a handler as a JMS topic subscriber or queue listener.
3. Create the service descriptor that designates a JMS destination using the messaging server object.
4. Connect the handler and service descriptor by a route.

This chapter describes the steps specific for JMS. For general information on creating handlers, service descriptor, and route, see Chapter 5, “Working with Handlers, Routes, and Service Descriptors.”

About JMS Mediation

The Gateway can mediate messages between endpoints of disparate protocols, for example, by transforming JMS messages into HTTP messages, and vice versa. To implement protocol mediation at the ACE XML Gateway, you set up a handler and service descriptor of the respective protocols, create a route between them, and, if needed, map properties and fields between each side.

Keep in mind that a protocol can be synchronous (in which a reply is expected per request) or asynchronous (in which no reply is expected). In general, protocol mediation is supported only between objects of protocols that are of the same mode. That is, an asynchronous object may only route to another asynchronous object, and a synchronous object only to a synchronous object. For example, asynchronous JMS messages can be routed to native Tibco/Rendezvous or asynchronous HTTP POST ebXML, but not to a SOAP Document destination, since it's request/reply-based.

This requirement is enforced in the web console, so that for a synchronous JMS handler, only valid routing options are available when setting up a route to the service descriptor.

A mapping between a JMS message and a message of another format can be as simple or as complex as you need it to be. In a simple mediation scenario, you could simply route a JMS message handler to an HTTP POST body service descriptor and specify body passthrough. In this case, when the ACE XML Gateway receives the JMS message, it creates a corresponding HTTP POST message and places the body content of the original message into the body of the HTTP message before sending it to the destination specified in the service descriptor.

In a more complex scenario, you could identify properties of the incoming JMS request and map them to properties in an outgoing message, transform an XML body content and map the output to a property in the target message, or have a portion of an XML body (indicated by XPath) mapped to a property.

To propagate the content of a JMS message property to a target message, you will need to identify the property (as an argument) in the message specification settings for the handler and corresponding service descriptor.

As may be expected, you will need to configure the argument in the incoming request specification of the handler for arguments mapped to the outgoing request specification of the service descriptor, and in the incoming response specification for the service descriptor to map content to outgoing responses in the handler.

Clusters and Message Buses

In general, an ACE XML Gateway performs similarly whether it is part of a cluster of gateways or acting alone. One important difference, however, is that a ACE XML Gateway cluster handles traffic for message bus servers differently from a non-clustered Gateway.

The routing model used by message-bus servers can take several forms. In JMS terms, the models are known as:

- queue-based messaging, in which a message is consumed by one and only one client (although more than one client may be listening on the queue).
- topic-based messaging, in which a message may be consumed by multiple clients, as many as are subscribed to the topic.

In topic-based distribution, because message buses simultaneously receive and send many messages with many different sources and destinations, there may be many copies of each message in flight simultaneously.

A single Gateway appliance can receive and deliver a message, but it cannot tell other Gateway appliances in the cluster to ignore that message. Because the message bus server may transmit many copies of the message, the chances are good that each Gateway appliance in a cluster will see a copy of the message, and that it will therefore be processed several times, once by each member of the cluster.

If your application logic permits messages to be processed more than once, then the fact that the ACE XML Gateway cluster may intercept and process a message multiple times is not a problem. On the other hand, if your application relies on each message being processed exactly once, then it may not be appropriate to configure a Gateway cluster to handle topic-based message traffic.

One possible solution would be to configure a single, non-clustered Gateway to process topic-based message traffic. The presence of only one Gateway appliance guarantees that each message will be processed only once. Keep in mind that queue-based message traffic, on the other hand, can be handled by either clustered or non-clustered Gateways.

For more information about configuring the Cisco ACE XML Gateway for use with message buses, contact your support representative.

Identifying the JMS Messaging Server

The first step in setting up the ACE XML Gateway for JMS traffic is creating the message server objects in the policy to identify the servers hosting the JMS daemons. The server can act as either a destination to which the ACE XML Gateway publishes messages or a source of messages.

Notice that you can set up various types of messaging servers in an ACE XML Gateway policy, including MQSeries and TIBCO Rendezvous. If attempting to implement security or mediation for JMS specifically, be sure to set up the new server as a JMS server, even if the underlying JMS implementation is TIBCO Enterprise Message Service or MQSeries.

To add a JMS messaging server representation to your policy:

Step 1 In the web console, click the **Messaging Servers** link in the **Policy** area of the navigation menu.

The **Messaging Servers** page appears. This page is organized by server type: the top of the page displays TIB/RV servers, the middle displays MQSeries servers, and JMS servers are at the bottom.

Step 2 Click the **Add a New JMS Server** button, at the top of the JMS Servers section.

If the button is not available, the JMS integration extension is not installed on your system. For more information, contact your support representative.

Step 3 In the **New Server** page, enter a name for the server that is unique for JMS servers in the policy.

Step 4 In the **Provider** field, type the connection information for the JMS service provider in *host*:*port* format, such as: example-jms.eng:7222

The host value should be the fully qualified hostname or IP address of the node on which the JMS daemon is running. The port value is the port number on which the daemon sends messages.

Step 5 In the **JMS Vendor** menu, choose the name of the service provider vendor. Identifying the vendor enables the ACE XML Gateway to manage vendor-specific variations in the JMS implementation.

Step 6 Click **Save Changes** to commit the changes to the currently active subpolicy.

The server is now available to be applied in service descriptors, as described in the following sections.

Configuring a JMS Handler and Service Descriptor

The following steps describe how to set up the service handler and descriptor for JMS messages.

A JMS handler enables the ACE XML Gateway to receive JMS messages. It specifies a JMS queue or topic on which you want the ACE XML Gateway to receive messages. The service descriptor, on the other hand, specifies a JMS destination where the ACE XML Gateway publishes messages.

The Gateway can mediate between JMS and other message formats. In this case, only one side of the handler/service descriptor pair would be a JMS protocol object.

Creating the JMS Handler

To create a JMS handler:

Step 1 Click the **Virtual Services** link in the **Policy** area of the navigation menu.

Step 2 In the **Virtual Services** browser page, choose **Handler** in the **Create a New** menu, and click the **Create** button.

The **Step 1 of 5: Handler Protocol** page appears.

Step 3 In the protocol menu, choose either:

- **JMS Request-Reply** for synchronous message exchange. This option is for JMS messages that are expected to produce a response. This option lets you specify requirements for the incoming message as well as for the response back to the JMS provider.
- **JMS Fire-and-Forget** for asynchronous JMS message exchange, in which a response is not specified for the message.

Step 4 Click **Continue**.

Step 5 In the **Step 2 of 5: General Information** page, enter a descriptive name for the handler in the **Name** field and choose the handler group in which the handler should be created from the **Handler Group** list. Configure other general settings as desired.

For details on configuring general settings for a handler, see "Creating Handlers" section on page 5-44.

Step 6 Click **Continue**.

The **Step 3 of 5: General Information** page appears.

Step 7 In the **JMS Server** menu, select the server you created in "Identifying the JMS Messaging Server" section on page 16-171. This server should be the host of the daemon that produces the messages that you want the ACE XML Gateway to receive and process.

Note If the server doesn't appear in the list, you can add the server to the policy by clicking the **Add New Server** button.

Step 8 From the **Subscribe to** menu, choose the JMS message distribution mechanism from the following options. Note that these correspond to the two standard delivery models defined by JMS.

- **Topic**. In this delivery mode, a message is published to as many clients as are subscribed to the specified topic.
- **Queue**. In this delivery mode, a message is delivered to a queue, where it is received by a single queue listener.

Note For more information, see JMS documentation at: http://java.sun.com/products/jms/index.jsp

Step 9 In the text field next to the subscription menu, type the identifier of the queue or topic and click the **Continue** button.

Step 10 In the **Step 4 of 5: Request Message Specification** page, specify argument settings for the incoming messages.

In the message specification you can configure what arguments are expected in incoming messages. Identifying arguments in this way serves two purpose: first, it lets you validate the incoming message (that is, you can ensure that messages have the properties that are expected by the destination); second, it enables mapping features in the route between a handler and service descriptor.

If you want to perform mediation that involves mapping of JMS properties (other than the body) to fields in other formats, you must specify the arguments in the **Request Message Specification** page.

To configure **Arguments** for the incoming request message:

a. Click the **Add a New Row** button.

   Fields for configuring argument requirements appear.

b. Specify argument validation by configuring the following fields:

   - **Name** is the name of the property as it appears in the message. This is a required field for any arguments you configure in the policy.
   - **Req** indicates whether the property is required or optional. If you choose this option and the property is not in the message, the message is blocked.
   - In the **Type** menu, choose the datatype for the property. The ACE XML Gateway can ensure that the specified argument is of the type you select, from the following types: String, XML, Integer, Float, Boolean, Time, or Binary.

     Depending on the type you select, you may additionally specify the validation mechanism (that is, an XML schema for XML arguments) or the content of the argument. For example, for an integer, you can specify that the argument must be less the 1000, for example.

Step 11 In the **Body Content Specification** area of the page, specify how the ACE XML Gateway should validate the payload of the message, from the following options:

- **Treat the body content as raw bytes (opaque data) with no validation**—Select this option if the payload contains raw bytes that should not be validated.
- **Treat the body content as XML with no validation**—Select this option to suppress validation of the message content at the ACE XML Gateway and if the content is XML. Choosing this option makes other XML options available for the service in the web console, such as transformations.
- **Validate that the body is well-formed XML only, with no content validation**—The ACE XML Gateway ensures that the body is well-formed XML, but does not verify the body against an XML schema.
- **Validate the body as XML according to**—Lets you specify an XML schema against which the ACE XML Gateway validates the message body content. You can choose the XML schema and, from within that schema, a specific namespace and element to which the message must conform.

Step 12 Click the **Continue** button.

The **Step 5 of 5: Response Message Specification** page appears. For fire-and-forget JMS message handlers, the page is empty since a response is not expected. For request-reply JMS message handlers, however, the page is the same as for the request message specification page.

Step 13 For request-reply JMS message handlers, use the controls to define requirements for the outgoing response message.

The requirements are applied to the message before it is returned to the server that produced the original request.

Step 14 Click **Continue**.

You are prompted to add a route for the handler.

Step 15 If the destination service descriptor already exists in the policy, you can add a route for it now. Otherwise, click **Finish Without Adding a Route** to complete the initial configuration of the handler.

The initial set up of the handler is complete. You can then use the properties page to configure additional options, such as XML signature verification or generation, or XML encryption or decryption.

Creating the Service Descriptor

Creating a JMS service descriptor is similar to creating a JMS handler, except that you are defining settings for a JMS producer rather than a consumer. As such, you define a JMS destination, rather than a JMS subscription profile in the service descriptor interface.

To create a JMS service descriptor:

Step 1 Click the **Virtual Services** link in the **Policy** area of the navigation menu.

Step 2 In the **Virtual Services** browser page, choose **Service Descriptor** in the **Create a New** menu, and click the **Create** button.

Step 3 In the protocol selection menu, choose either:

- **JMS Request-Reply** for synchronous message exchange. If this option is selected, the ACE XML Gateway will expect a response to message.
- **JMS Fire-and-Forget** for asynchronous JMS message exchange, in which a response is not expected.

Step 4 Click the **Continue** button.

The **General Information** page appears.

Step 5 Enter a descriptive name for the service descriptor in the **Name** field.

Step 6 In the **Server** menu, choose the JMS server that hosts the provider to which you want to send messages.

If the server does not appear in the menu, click the **Add a New Server** button, and create the server as described in “Identifying the JMS Messaging Server” section on page 16-171.

Step 7 Click **Continue**.

Step 8 In the **Destination** menu, choose the type of JMS message destination from the following options. (Note that these correspond to the two standard message delivery models defined by JMS.)

- **Topic**. In this delivery mode, a message is published to as many clients as are subscribed to the specified topic.
- **Queue**. In this delivery mode, a message is delivered to a queue, where it is received by a single queue listener.

Step 9 In the text field next to the destination menu, type the identifier of the queue or topic.

Step 10 For a JMS Request/Reply service descriptor, specify the following configuration parameters as appropriate:

- **Timeout**. If a connection cannot be established within the time you specify in this field, the ACE XML Gateway logs the event and abandons the connection attempt.
- **Service Time Threshold**. If a connection is not established within the number of milliseconds you specify in this field, the ACE XML Gateway logs the event but continues the attempt to connect to the server.

Step 11 Click the **Continue** button.

Step 12 In the **Step 4 of 5: Request Message Specification** page, configure requirements for the outgoing request, and for the incoming response on the **Step 5 of 5: Response Message Specification** page.

Note that fire-and-forget does not provide settings for the response, since a response is not expected.

As for handlers, you need to specify the arguments for service descriptors if you want to implement protocol mediation in which the fields are mapped from another format to properties in the outgoing JMS request.

For more information on using the message specification controls, see the equivalent instructions in "Creating the JMS Handler" section on page 16-172.

Step 13 Click the **Continue** button to complete the initial configuration of the service descriptor. The properties page for the service descriptor appears.

This completes the initial set up of the service descriptor. You can subsequently use the properties page for it to add more settings, such as XML signature verification or generation, XML encryption or decryption, and so on.

Step 14 When finished, set up a route to this service descriptor from a handler in the **Virtual Services** page. For information on setting up routes between handlers and service descriptors of different protocols, see "Configuring JMS Mediation" section on page 16-175.

The initial set up of the service descriptor is complete. You can then use the properties page to configure additional options, such as XML signature verification or generation, or XML encryption or decryption.

Configuring JMS Mediation

This section provides an overview of the steps for setting up simple mediation. While the steps reflect a simple implementation, they will help you to understand how mediation is implemented in policies, and get you started with your own implementation.

Step 1 If you have not already, create the handler and service descriptor for the source and target messages.

Associated handlers and service descriptors can be of disparate protocol types, but handlers of asynchronous protocols can only route to service descriptors of protocols that are also asynchronous. Likewise, synchronous protocols can route only to synchronous objects.

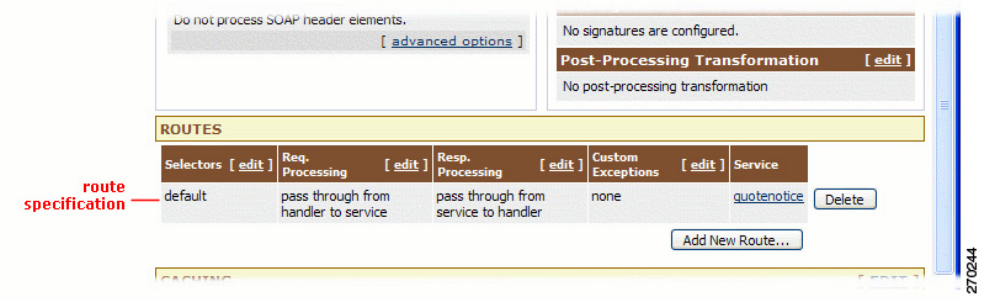
Step 2 Define the arguments for which you want to create a mapping in the handler and service descriptor.

Step 3 Create a route from the handler to the service descriptor. You can create a route by clicking the **Add New Route** button at the bottom of the handler properties page.

Note For details on creating a route, see "Working with Routes" section on page 5-46.

After creating a route, the route specification appears in the handler property page, as shown in Figure 16-4

Figure 16-4 Route specification


Do not process SOAP header elements.
[advanced options]
No signatures are configured.
Post-Processing Transformation [edit]
No post-processing transformation
ROUTES
Add New Route...
route specification
CACHING
270244

Selectors [edit]	Req. Processing [edit]	Resp. Processing [edit]	Custom Exceptions [edit]	Service	
default	pass through from handler to service	pass through from service to handler	none	quotenotice	Delete

Basic mediation occurs automatically. That is, by default, the ACE XML Gateway implements passthrough on the bodies of messages that are routed between service descriptors and handlers of different protocols. You can configure custom mappings, however, as follows steps.

Step 4 To modify the default message mapping behavior, in the route area of the handler page, click the **edit** link for either:

- **Req. Processing** to map properties of the request message passed from the handler to the service descriptor (that is, messages destined for the backend service).
- **Resp. Processing** to map properties of the response message as it is passed by the ACE XML Gateway from the service descriptor to the handler (that is, messages destined for the consumer).

Step 5 The controls in the **Edit Route Request Processing** page or **Edit Route Response Processing** page vary depending on the protocol of the service descriptor or handler. For instance, there are no options of configuring body content handling for HTTP GET. In general, however, you can achieve the desired mappings by choosing the type of mapping you want to be performed from the argument and body processing menu, such as **Map body to body and arguments** or **Map body and arguments to body and arguments**. When done, mapping fields appear in the page.

Step 6 Use the mapping fields to configure the source and target fields for each mapping. Any properties that were configured in the message specification form are available in the mapping menu controls.

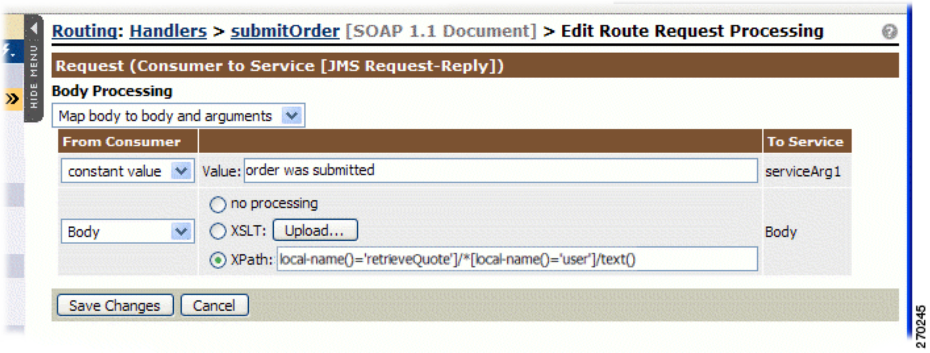
If a source field exists for which you do not specify a corresponding target field mapping, the source field is stripped from the message and its content is lost.

For XML properties or body content, you can further specify:

- An XSLT that is to be applied to the content prior to its propagation to the target field.
- An XPath that identifies a portion of the XML property or body to be propagated to the target field.

For example, the mapping in Figure 16-5 specifies a portion of an XML body to be propagated to the body of the output message, and a hard-coded value to comprise a property in the output message.

Figure 16-5 Property and Body Mappings Form



Routing: Handlers > submitOrder [SOAP 1.1 Document] > Edit Route Request Processing

Request (Consumer to Service [JMS Request-Reply])

Body Processing

Map body to body and arguments

From Consumer		To Service
constant value	Value: order was submitted	serviceArg1
Body	○ no processing ○ XSLT: Upload... ● XPath: local-name()='retrieveQuote']/*[local-name()='user']/text()	Body

Step 7 When finished, click **Save Changes** and deploy the policy to have your changes take effect at the ACE XML Gateway.