Cisco ACE XML Gateway

Getting Started Guide

Software Version 5.1

Cisco Systems, Inc. | 170 West Tasman Drive | San Jose, CA 95134-1706 | 800 553-6387

The Cisco ACE XML Gateway is an application oriented networking product.

This product includes software developed by the Apache Software Foundation (http://www.apache.org/)

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/).

This product includes LibCURL. cURL is © 1996 - 2004, Daniel Stenberg, <daniel@haxx.se>. All rights reserved.

This product includes libxslt, the XSLT C library developed for the Gnome project, and libxml2 Libxslt is based on libxml2 the XML C library developed for the Gnome project.

This product includes OpenLDAP Copyright 1999-2003 The OpenLDAP Foundation, Redwood City, California, USA. All Rights Reserved.

This product includes software developed by Computing Services at Carnegie Mellon University (http://www.cmu.edu/computing/). [Net-SNMP]

This product includes software developed by the University of California, Berkeley and its contributors.

The regular expression support is based on Henry Spencer's POSIX 1003.2 compliant regex package that has Copyright 1992, 1993, 1994 Henry Spencer. All rights reserved.

GS5.1-070523-2303-a

OL-13875-01

CONTENTS

PART I. FIRST STEPS

This Part I introduces concepts that serve as the foundation for setting up and using the Cisco ACE XML Gateway. It describes the first steps in implementing the XML Gateway—setting it up in your network environment.

The following topics are covered:

- Introducing the Cisco XML ACE Gateway Solution
- Before Starting
- Performing the Initial Setup
- Accessing the XML Manager Web Console

Introducing the Cisco XML ACE Gateway Solution

The Cisco ACE XML Gateway, a component of the Cisco Application Control Engine (ACE) family of products, brings application intelligence to the network. It enables efficient deployment of secure, reliable, and accelerated Web service environments based on XML (Extensible Markup Language) and SOAP (Simple Object Access Protocol). It provides tools and capabilities that simplify the use of XML and SOAP-based technologies including XML Signature, XML Encryption, and SAML.

This guide introduces you to the tools and features of the ACE XML Gateway, including its security, mediation, and acceleration capabilities. This guide uses step-by-step procedures to acquaint you with how to set up and deploy a working ACE XML Gateway policy. The policy is the set of rules and behaviors that determine how the ACE XML Gateway handles traffic.

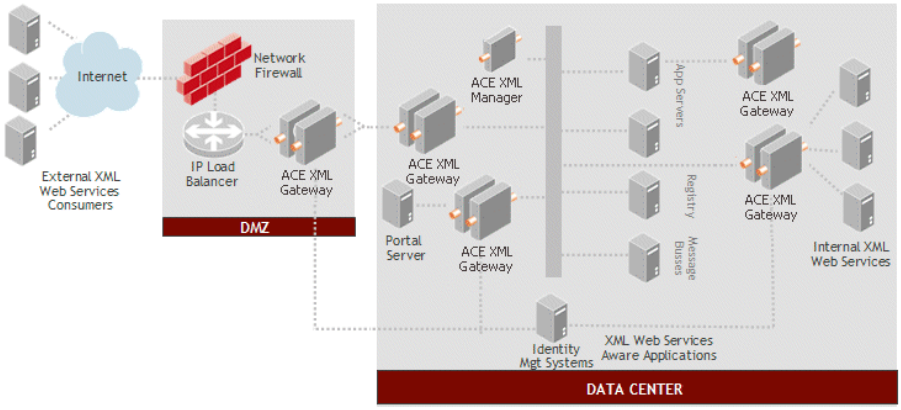
How it Works

As depicted in Figure 3-1, the ACE XML Gateway can operate in various locations in a network, depending on your requirements.

It is often deployed in the DMZ, where it can enforce access rules and other security policy requirements on traffic from external sources.

Within the protected network, it can act as a central routing and mediation point for application traffic. It can also serve to offload processor-intensive or XML processing tasks from backend application servers.

Figure 3-1: Deployment Topology



Wherever deployed, the ACE XML Gateway can provide application-level traffic management and control, ensuring uniform policies across services on your network.

Cisco ACE XML Gateway System Components

A Cisco ACE XML Gateway deployment consists of:

- ACE XML Gateway, an enterprise-class appliance for securing and managing application traffic. The ACE XML Gateway serves as the policy enforcement point on the network.
- ACE XML Manager, the administration server for the Cisco ACE XML Gateway solution. The ACE XML Manager serves the web console, the interface for developing the policy and monitoring the system.

A single appliance can operate as both manager and gateway (in what is called the standalone mode). However, this configuration is intended for policy development or evaluation only and not recommended for production deployment.

A production deployment typically consists of an ACE XML Manager and one or more ACE XML Gateways. Multiple ACE XML Gateways can be organized into clusters. While all Gateways in a single cluster should have the same policy, different clusters can apply different policies. In other words, you can have separate ACE XML Gateway clusters that each have their own policies, possibly specialized for different business needs, or to use a single ACE XML Manager to transfer policy versions between development, testing, and production environments.

The ACE XML appliances are delivered in a full-size, rack-mountable chassis or in a desktop form-factor, the Gateway-D. The Gateway-D is intended for evaluation and development settings, while the rack-mounted appliances are suitable for production environments.

Understanding Gateway Policies

The ACE XML Gateway handles traffic as specified by a *gateway policy*. A policy contains global settings for traffic at the ACE XML Gateway, such as denial-of-service (DOS) settings and content screening rules, as well as settings specific to a service.

You develop the policy in the ACE XML Manager web console. From there, the policy is deployed to the ACE XML Gateway cluster for enforcement.

In a policy, you can configure a variety of message processing features, including:

- validation of the content, arguments, or other properties of the message
- transformation of message content
- dynamic selection of service routing based on message attributes or content
- verification of consumer credentials, such as username/password, certificates, SAML assertions, and more
- encryption and decryption of content with XML Encryption
- signature verification and generation

A policy is made up of numerous policy objects and resources, which together determine the rules and behaviors of the ACE XML Gateway. An important object in a policy is the service proxy, which is discussed next.

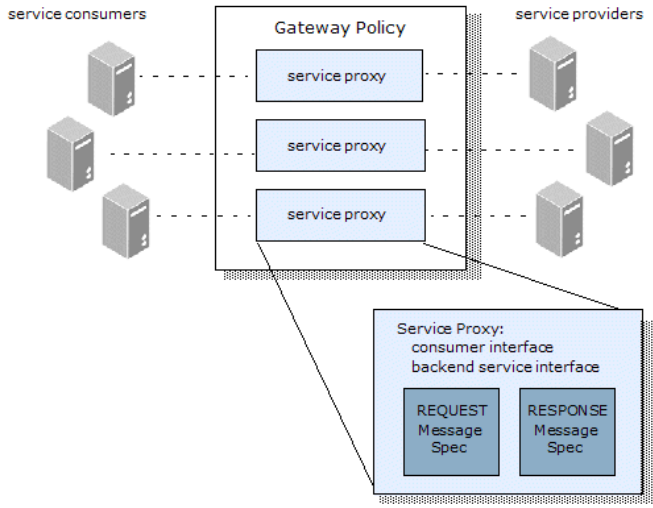
Using Service Proxies

The primary policy element for defining external services at the ACE XML Gateway is the *service proxy*. As shown in Figure 3-2, a policy can have many service proxies, each corresponding to a particular backend resource available through the ACE XML Gateway.

A service proxy identifies connection parameters for the back-end service (the service interface) and the way the service is exposed to service users (that is, the consumer interface).

For backend SOAP services, each service proxy may correspond to a particular operation or, if generated by WSDL import, all operations in the WSDL.

Figure 3-2: Components of an ACE XML Gateway Policy



A service proxy defines the requirements that messages must meet to reach the destination service. The requirements may involve consumer credentials or composition requirements, such as the presence or validity of message arguments. If the message meets the requirements, it is passed through the ACE XML Gateway. Otherwise, the Gateway blocks the message and generates an error response.

This guide gives you hands-on experience with developing service proxies in the XML Manager and testing them with live traffic at the ACE XML Gateway. The steps in this guide provide practical experience with the features and concepts introduced.

Before Starting

The following sections describe the information, tools, and system requirements needed to complete the steps in this guide.

What You Need to Know

This document assumes that you are familiar with concepts surrounding networking, security, and XML technologies. As advertised, a service-oriented architecture implementation diminishes the barriers between network and application domains, with a corresponding effect on organizational roles. Therefore, regardless of experience, implementing the ACE XML Gateway system usually involves new tasks.

This guide introduces concepts behind the ACE XML Gateway, presented through hands-on instruction. The other books in the documentation set, the *Cisco ACE XML Gateway Installation and Administration Guide* and *Cisco ACE XML Gateway User's Guide*, provide additional background material as well. For even more information on background concepts you may need to know, we recommend the following books:

- *SSL and TLS* by Eric Rescorla, for information on networking security in general, and PKI specifically
- *Securing Web Services with WS-Security* by Rosenberg and Remy, for information on WS-Security

Initial Network Configuration Requirements

The first steps described in this guide are intended to get the ACE XML appliance up and running in your network. To perform the steps, you will need:

- A static IP address for the ACE XML appliance

  Note: You only need one address if setting up the ACE XML Gateway and Manager to run as a single instance (the configuration described in this guide). Two addresses are required if deploying the ACE XML Gateway and ACE XML Manager on separate appliances.

- The IP address of the default IP gateway for the network
- The IP address of the primary DNS server for the network

- A hostname for the ACE XML Gateway registered with the local network's DNS server (recommended)
- The password for the root account on the ACE XML Gateway appliance.

  If this is a new installation and you do not know the root password, please contact your support representative.

Requirements for Generating Traffic

Once you set up the policy using the steps in this guide, you can test the policy with live traffic. To do so, you can use the test browser built into the XML Manager or an HTTP client tool, such as Microsoft WFetch or Curl.

Sample services are provided for your initial testing. If you want to follow the steps in this guide using your own services, however, you will also need:

- Web services running on a server (such as in .NET or AXIS)
- WSDL files describing the available services (although WSDL files are not required by the ACE XML Gateway, they make it easier to create a security policy)
- To test security features, you'll need resources such as a public/private keypair for the ACE XML Gateway and a client certificate.
- If testing the ACE XML Gateway with other systems (such as an LDAP server), those systems need to be available along with any associated resources (such as LDAP queries)

Performing the Initial Setup

These procedures describe how to perform the initial setup of the ACE XML appliance, from physically connecting it to the network to configuring its basic network settings.

Preparing for Installation

The basic network settings for the ACE XML Gateway or ACE XML Manager appliance are configured from the appliance operating system environment. Since the available ports on the appliance chassis vary by model, the method for accessing the operating system vary as well.

You can access the appliance operating system by connecting a monitor and USB keyboard or KVM switch to the appliance. The appliance also supports serial console connections, in which a personal computer with terminal emulation software or a dumb terminal is connected by serial cable to the appliance. The terminal emulation software or dumb terminal should be VT100-compatible.

> **Note:** While you can access the appliance by serial connection, note that boot-up messages are configured to go to KVM output rather than serial output by default. For more information on serial access, see "Connecting by Serial Cable."

The steps for performing the initial set up of the appliance are:

1. If you have not already done so, remove the appliance from its packaging.

   > **Note:** If you are evaluating the appliance, be sure to keep the packing materials for use when returning the system. Also, the packaging usually includes a return shipping label you can use when returning the appliance.

2. Connect a monitor and keyboard to the back of the appliance. For information on connecting by serial cable, see "Connecting by Serial Cable."
3. Plug a power cable into the power supply port on the appliance.
4. Attach an Ethernet cable to a network interface port on the back panel of the appliance. Rack-mounted appliances have as many as five Ethernet interface ports. Connect the cable to any of the connectors except the Integrated Lights-Out connector, which is

on the top-left side of the back panel. Make a note of which interface you are using.

5. Turn on the appliance by pressing the power button on the front panel for a Gateway-D or on the back panel for a rack-mounted appliance. (On older models, you may need to remove the faceplate to access the on/off switch.)

The system starts up. When the startup procedure finishes, a login prompt appears on the monitor.

Connecting by Serial Cable

Instead of a direct KVM connection, you can connect to the appliance using a laptop or personal computer connected by serial cable to the DB-9-type serial connector.

The laptop or personal computer needs to have VT-100-compatible terminal emulation software, such as Microsoft HyperTerminal.

Configure the connection to use the following settings:

- Bits per second: 9600
- Data bits: 8
- Parity: None
- Stop bits: 1
- Flow Control: None

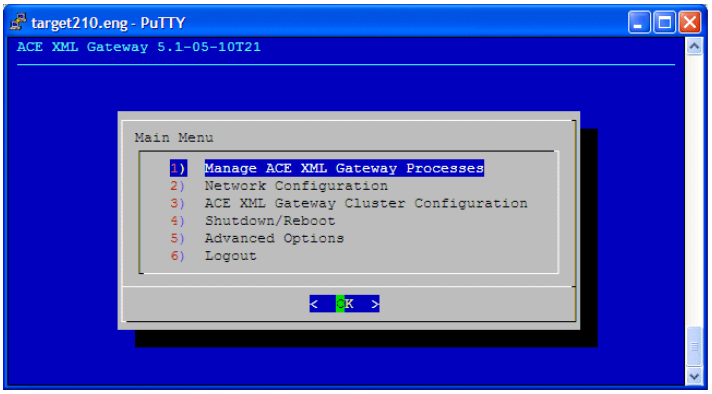
Performing the Initial Configuration

After physically connecting the appliance, configure its basic settings as follows:

1. At the appliance login prompt, log in as `root` user with the password provided to you. If you don't know the `root` password, contact your support representative.
2. When logging in with the default password, you will be prompted to change the password. To do so, at the prompt:
   a. Click **OK** or press enter.
   b. In the **Please enter the new root password** screen, type a new password and click **OK**.
   c. Confirm the password when prompted.

   The main menu of the appliance administration interface appears.

Figure 5-1: Command-line Shell Menu



4. Choose the second item, **Network Configuration**, and click **OK**.

   The **Network Configuration** menu appears.

5. Configure network settings from the menu as follows:

   a. Select the first item, **Hostname**, and click **OK**.

   b. In the hostname screen, enter the fully qualified hostname of the ACE XML Gateway (such as `xmlgate.example.com`). When finished, click **OK**.

      The **Network Configuration** menu appears again.

   c. Select **IP Gateway** and type the IP address of the default gateway of the subnetwork to which the appliance network interface is connected. When finished, click **OK**.

   d. Select **Name Servers** and enter the IP addresses of the DNS servers in your network. To specify more than one DNS server, enter their IP addresses separated by a space. The appliance will query the first server in the list, and query others in the order they appear if a name server is unresponsive. When finished, click **OK**.

   e. By default, the physical network interfaces on the appliance are disabled. In the **Network Configuration** menu, choose the interface to which you have plugged in the network cable by its identifier, such as **Interface eth0**.

   Note: The Gateway-D appliance only has one interface menu choice, **Interface eth0**, since it only has one Ethernet port. The full-sized appliances have an interface menu item for each network interface.

   f. Select the first item on the network interface menu, **enabled**.

   g. Type the IP address for the appliance and click **OK**.

h. Enter the network mask (netmask) for the IP address.

i. At the next prompt, choose an Ethernet speed for the interface and click **OK**.

j. In the **Edit Static Routes** screen, choose **Accept settings** and click **OK**, or **Cancel** to start over.

k. In the **Network Configuration** menu, select **Test Network Settings**. The ACE XML Gateway performs a few basic network tests of your settings and reports their result.

12. If prompted to configure cluster settings, choose yes to configure them at this time. (If not prompted, in the **Main Menu**, choose the third option, **ACE XML Gateway Cluster Configuration**.)

a. If working on a Gateway-D, choose the **Both Gateway and Manager** option to have the appliance operate in standalone mode. In this mode, the appliance includes the functionality of both gateway and manager (this is the mode normally used for system evaluation).

For a dedicated gateway or manager, choose an operation mode of either **ACE XML Manager** or **ACE XML Gateway Cluster Member**.

b. If you chose for the appliance to act as an ACE XML Gateway, specify the address of the ACE XML Manager that will administer this appliance.

c. When prompted, restart the appliance services for your changes to take effect. Choose **yes**.

d. If you chose for the appliance to act as an ACE XML Manager or standalone machine, click **OK** at the notice screen that you will need to add or change cluster members from the ACE XML Manager web console.

For a standalone machine, the Manager is already configured to administer its own Gateway instance. For Manager-only machines, you need to configure the Gateways to be managed from the Cluster Management pages of the web console.

Congratulations! Your ACE XML Gateway is now configured and ready to use. To exit the shell interface, from the Main Menu choose Logout.

Now that the appliance network settings are configured, you can unplug the monitor and keyboard or serial cable. The ACE XML Manager web console is now accessible from a browser on the local network. If you need to access the appliance operating system environment again, you can use a remote SSH client, such as PuTTY.

Accessing the XML Manager Web Console

The ACE XML Manager web console is the development environment for an ACE XML Gateway policy. In addition to policy development, it serves as the monitoring point for the system deployment.

Logging In to the ACE XML Manager Web Console

The ACE XML Manager web console works with recent browser versions. It is specifically supported on Mozilla Firefox 1.5.0.*x* and 2.0.0.*x* and Microsoft Internet Explorer 5.5 and 6.

JavaScript functionality must be enabled in the browser for many ACE XML Manager features to work properly.

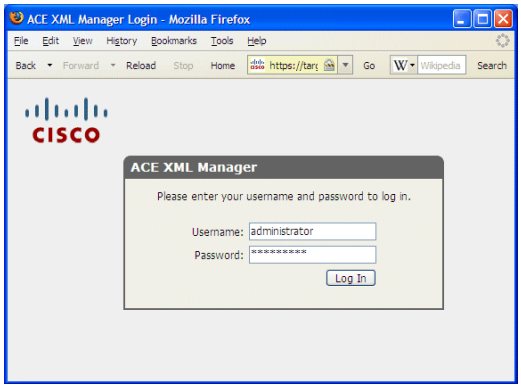
To access the ACE XML Manager from a browser:

1. In the address field of the browser, enter the URL for the console: `https://<IP_Address>:8243`

   Where *<IP_Address>* is the IP address you configured for the ACE XML Manager in the previous section. Notice that you need to connect to the ACE XML Manager web console by secure HTTP (HTTPS). Also, as shown, the default listening port for the console is 8243.

2. Accept the temporary certificate to view the login page.

Figure 6-1: Manager Login



3. In the login fields, enter `administrator` as the user name (case-sensitive) and the administrator's password. If you do not know the administrator password, contact your support representative.

Note: Keep in mind that this is a different user account (usually with a different password) from the one used to access the appliance Shell interface described in the previous chapter, "Performing the Initial Setup."

When starting a production-level project, one of the first tasks is to create user accounts in the web console for each person or group participating in the project. At evaluation time, however, you may wish to use the pre-existing `administrator` account.

4. If your system has *not* been pre-configured with a valid license key, a message appears notifying you that you need to update the product license.

   If you do not have a license file, contact your support representative. Otherwise, follow these steps to update the license:

   a. With a text editor, locate the license file provided to you and copy its contents to your system clipboard.

   b. After copying the license, click the **License Management** link at the bottom of the license error page.

   c. Paste the license text into the **License File** field and click **Save Changes**.

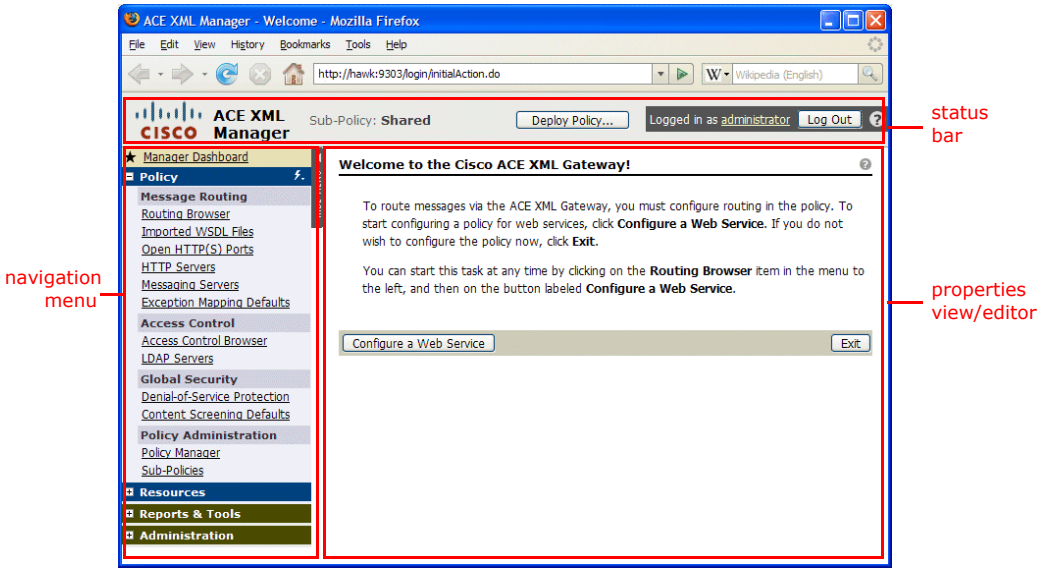
   If successful, a message appears notifying you that "Your changes have been saved". You can now access the ACE XML Manager web console.

Navigating the ACE XML Manager Web Console

When you log into an ACE XML Manager web console that has an *empty* policy (that is, a policy in its initial state), the **Welcome** page appears. You can start configuring SOAP traffic routing directly from the **Welcome** page, or exit the page to work on other parts of the policy or configure routing for other types of traffic.

Figure 6-2 shows the main elements of the ACE XML Manager web console interface.

Figure 6-2: ACE XML Manager web console



The primary components of the interface are:

- The *properties editor* shows status information and configuration settings for a particular aspect or behavior of the system.
- The *navigation menu*, which appears on the left side of the console, provides access links to the various property pages of the console.
- The *status bar* provides access to common operations, such as deploying or switching sub-policies (a policy can be organized into multiple parts, called sub-policies).

This guide contains detailed, click-by-click steps for performing various configuration tasks. Be aware that there is usually more than one way to navigate the console to accomplish a task. As you follow these steps, feel free to explore the ACE XML Manager web console interface and use the navigation path you prefer.

PART II. WORKING WITH POLICIES

Part II takes you through the steps for defining the policy objects for protecting services at the ACE XML Gateway. You start by importing a WSDL that describes the services to be proxied at the ACE XML Gateway. You then test the configuration and try out the logging tools included with the system.

The following topics are covered:

- Virtualizing Services
- Deploying the Policy to the XML Gateway
- Sending Traffic to the XML Gateway
- Using Event and Message Logs

Virtualizing Services

After installation, the XML Manager and Gateway contain an empty policy to which you can start adding your own rules and settings. You can create the basic settings for proxying services in the policy by hand (by manually adding the policy objects that represent external services) or automatically (by WSDL import).

Defining a Web Service by WSDL Import

The first step in configuring the ACE XML Gateway to handle service traffic is to create the policy objects that encapsulate settings for an external service. You can jump-start this task by importing the WSDL that describes the service. The Manager generates policy objects based on the contents of the WSDL, including service proxies, the policy object that contains settings for external web services.

A WSDL can be imported from a URL location or from a UDDI registry. In these steps, we'll import a WSDL from a URL.

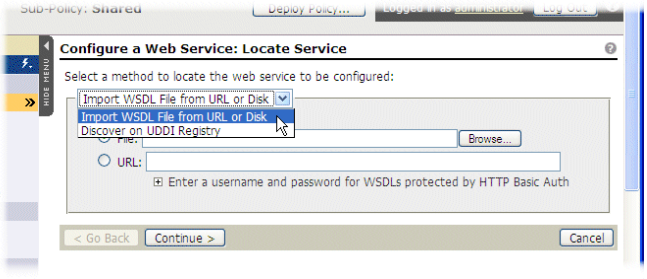
To import a WSDL:

1. On the **Welcome** page, click the **Configure a Web Service** button.

   Note: From elsewhere in the web console, you can access this button by navigating to the Routing Browser (click the **Routing Browser** link in the navigation menu).

   The first page in the **Configure a Web Service** page sequence is **Locate Service**. As shown in Figure 8-1, the menu presents several options for defining Web services in the policy.

Figure 8-1: Import options



The Manager can generate service definitions based on information acquired by a UDDI registry query or from WSDL import.

2. In this case, we'll import a WSDL from a URL location. With the **Import WSDL File from URL or Disk** option selected in the menu, choose the **URL** radio button

3. In the field next to the **URL** button, enter the URL of a service hosted on the sample resource site:

   http://example.reactivity.com/service/order.asmx?WSDL

   Note: If you are creating a policy for another service, either type the URL or identify the WSDL file you want to use on your file system using the **File** field. Also, instead of specifying a WSDL directly, you can use the controls to have the Manager discover services on a UDDI server.

4. Click **Continue**.

   In the **Consumer Interface** settings, notice that the **Exposed Port** selection is `Default HTTP port [80, Insecure]`. Later we'll configure the service to use the SSL port 443. Until then, the default HTTP port 80 is the only one available.

5. The **Exposed Path** is the invocation path that consumers use to address the service at the ACE XML Gateway. Replace the default value with `/orders`

   By default, the policy takes the invocation path from the WSDL. For security reasons, it is usually advisable to change the default invocation path. This obscures details about the backend infrastructure that may be exposed by the path and exploited by attackers (an extension of .asmx, for instance, suggests a .NET framework). It also allows you to expose a calling interface that makes more sense to consumers and it insulates consumers from possible changes to backend services.

6. For the **Access Control** option, choose **Public**. Later, we'll specify conditions that control access to the service.

7. From the **Default Message Logging** menu, choose `log bodies of inbound and outbound messages`.

   When developing a policy, it's generally a good idea to turn on message traffic logging. That way, you can view the effects of configuration changes in the actual messages passed through the ACE XML Gateway.

8. Click the expand control to display the **Advanced Options**.

   The Manager represents external services with a policy object called a service proxy. A service proxy can correspond to a single operation in the WSDL, or to each "service" element, in which case the service proxy may contain multiple operations. By default, the

Manager creates multiple operation service proxies, since they provide a single point for configuring settings for multiple operations. For now, however, we'll generate a service proxy for each operation.

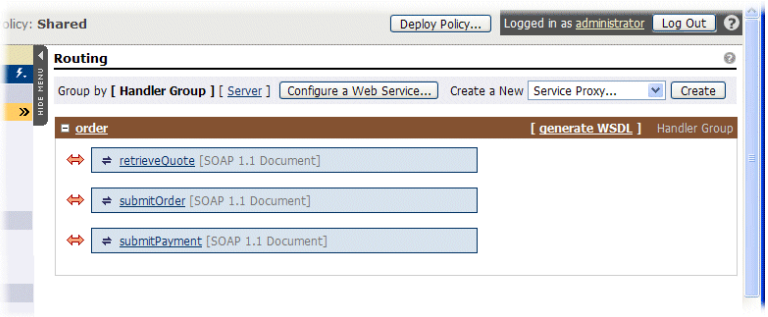
9. Click to disable the checkbox for the option labelled **Condense SOAP Document operations into one service proxy per WSDL "Service" element, if possible**, so that it is not selected.
10. Click **Continue** at the bottom of the page.

    The **Deploy Policy** page appears. As indicated by the text on the page, the changes do not take effect at the ACE XML Gateway until you deploy the policy. Until then, messages addressed to the orders services at the Gateway will be blocked. We'll deploy the policy later. First, lets look at the results of the WSDL import.
11. Click **Return to Routing Browser**.

Examining the New Policy

The Routing Browser displays the services for which the ACE XML Gateway performs message routing. As shown in Figure 8-2, several service proxies were generated based on the imported WSDL.

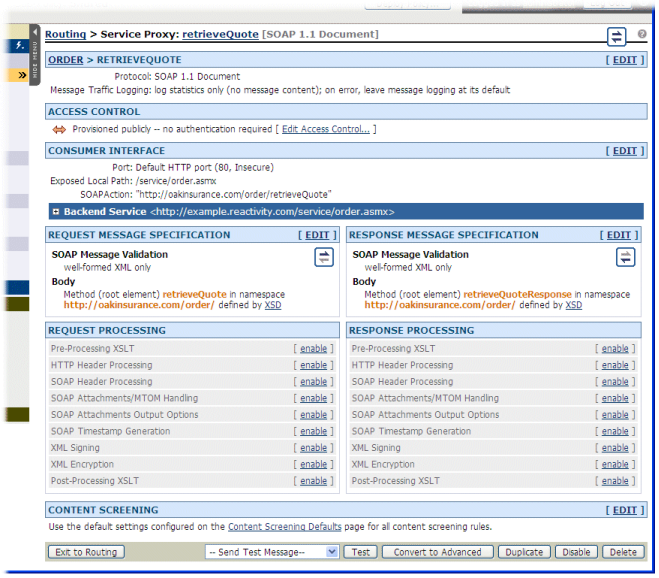
Figure 8-2: Routing Browser with New Service Proxies



The policy now contains three service proxies: `retrieveQuote`, `submitOrder`, and `submitPayment`. Clicking on the name of one of the service proxies takes you to its configuration details.

For example, click on `RetrieveQuote` to view settings for this service.

Figure 8-3: Service Proxy Information Page



This page shows settings specific for traffic handling for this service, including request and response validation and processing requirements. You will initiate most of the configuration tasks described in this document from the settings page for a service.

Deploying the Policy to the XML Gateway

So far, you've been modifying the working policy, that is, the policy under development on the ACE XML Manager. To have the policy applied to network traffic, you need to deploy it to the ACE XML Gateway.

Deployment occurs in several stages:

- The Manager compares the current and proposed policies and presents the differences for your review.
- If you accept the changes, the Manager compiles the policy.
- The compiled policy is transferred to the ACE XML Gateways.

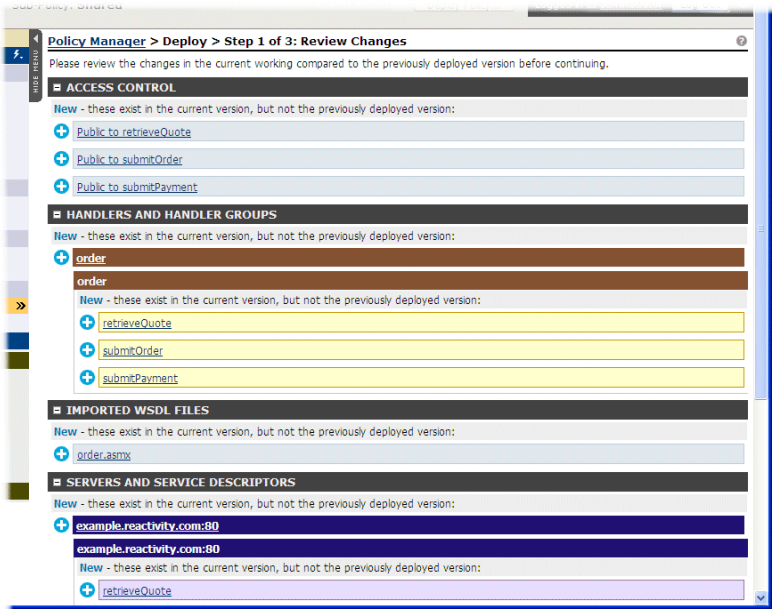
Steps for Deploying the Policy

To deploy a policy:

1. Click the **Deploy Policy** button at the top of the ACE XML Manager web console.

   The first deployment page shows changes between the new and old policy, in this case, objects added by WSDL import.

Figure 9-1: Deployment Step 1 of 3, Review Changes



2. Scroll to the bottom of the page and click **Continue to Next Step**.

The **Step 2 of 3, Basic Policy Review** page appears. It shows warnings and errors in the policy to be deployed. Notice that it presents a performance warning regarding the log setting. Since logging of message bodies affects performance, it is not recommended in production systems.

3. Click **Continue to next step**.

   The **Compile and Deploy** page appears, which shows the ACE XML Gateways in the administrative domain of the Manager. You can deploy a policy selectively to specific Gateways. Generally, however, all Gateways should have the same policy version.

   A status of **Out of date** indicates that the policy on the ACE XML Gateway differs from the one you just compiled.

4. Type a description of the policy in the **Policy Description** field, such as "Order services WSDL imported".

   By default, a policy description is an optional value. In general, it is recommended that you provide descriptions for policy versions. They help in case a rollback or policy compare needs to be performed later.

5. Click **Deploy to Selected Gateways** to have the compiled policy deployed to the ACE XML Gateway.

After a moment, the deploy page reappears. This time, the status of the ACE XML Gateway should be **Up to date**.

Managing Policies

When you deploy a policy, the deployed version is saved in the policy version history in the Policy Manager. The ACE XML Manager includes many features for administering policies, including the ability to back-up a policy (such as the one you just deployed), restore a policy to a previous version, or compare policy versions.

To access the Policy Manager, click the **Policy Manager** link in the **Policy** area of the navigation menu.

Figure 9-2: Policy Manager

Notice that the policy version history at the bottom of the page shows several versions, including one for the policy you just deployed.

From the Policy Manager, among other tasks, you can:

- *View* or *rollback* a previous version of the policy. Rolling back replaces the current working policy in the Manager with the one last deployed to the ACE XML Gateway. Once you rollback a policy, you still need to deploy the policy to propagate the version to the ACE XML Gateway. To roll back to a previous version of the policy, click the **Roll Back** button in the policy history list.
- *Export* the policy to a file. This is useful for moving policies between system environments (for example, between development and production), backing up a policy, or moving objects between sub-policies. It is also useful for troubleshooting, since it allows you to exchange policies with support.
- *Save to History* creates a copy of the current working policy, which then appears in the policy history list. Note that policy version is also saved automatically when you deploy a policy.
- *Compare policies* to view differences between the working policy and the one deployed on the ACE XML Gateway or between any two policy versions.
- *Reset policy* to its initial state. Clicking the **Reset Policy** button in the Policy Manager clears any changes to the policy since installation. This is useful when learning the system. As with roll back, you still need to deploy the policy to have the reset policy take effect at the ACE XML Gateway.

Sending Traffic to the XML Gateway

After deploying the policy, you can test the configuration by sending an Order service request to the ACE XML Gateway. The ACE XML Manager web console includes a test browser you can use to send simple requests to services at the ACE XML Gateway.

When you submit a request from the test browser, the request actually originates at the ACE XML Gateway itself. Like any client, the ACE XML Gateway must meet access requirements for the service proxy, if any. (This is important to keep in mind if the policy limits access to specific client IP addresses.)

While the test browser is primarily intended for checking connectivity between the ACE XML Gateway and backend services, it can be useful as a first test of a new service proxy.

Using the Test Browser

To try out the service proxy from the test browser:

1. In the **Routing Browser**, click the name of the `retrieveQuote` service proxy.

   The service proxy information page appears.

2. Scroll to the bottom of the page until the **Test** menu appears. (This is the menu that contains a default selection of **-- Send Test Message --**.)

3. Choose **Test Consumer Interface** from the **Test** menu, and then click the **Test** button.

   The **Send Test Message** window appears. Notice that the SOAPAction header and body appropriate for the service proxy is pre-populated in the text fields. The Manager populates the body of the request with values derived from the schema. For example:

   ```
   <ord:year>3</ord:year>
   ```

   Optionally, change the automatically generated values to more realistic ones. However, this is not necessary, since the service will work with the values generated by the test browser.

4. When ready, click **Send Test Message**.

If successful, a successful response message appears in the Response page, indicated by the 200 HTTP response code. Notice that the element named `retrieveQuoteResponse` appears in the response. If unsuccessful, an error message appears in the response window.

Client Test Tools

While the test browser provides a useful first-line testing tool for Gateway services, for a more realistic testing scenario, you should use an HTTP client test tool. A number of client test tools are available, including the freely available Microsoft WFetch, soapUI, or Curl.

Curl is command-line tool available from the following site:

`http://curl.planetmirror.com/`

Parasoft SOAtest is a SOAP client testing tool that supports automation and load testing. For more information, see:

`http://www.parasoft.com`

In the following steps, the service is tested with Microsoft's WFetch. WFetch is part of the IIS Diagnostics Toolkit, which you can get by searching on "WFetch" at the Microsoft Download Center:

`http://www.microsoft.com/downloads/search.aspx`

Testing the Policy with WFetch

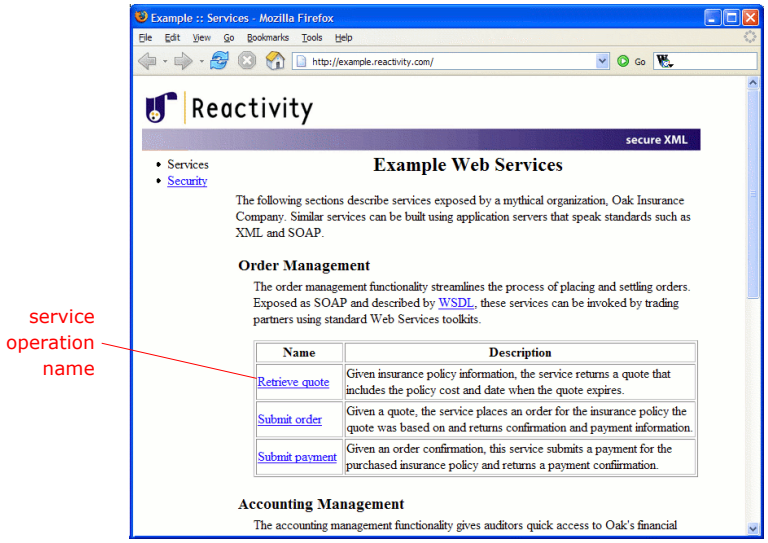
To test the service with WFetch, first download and install WFetch. For more information, see the Microsoft documentation.

The test browser in the console generated a request message for you. From another client, you will need to create the test message manually. You can do so using the message template presented on the service example site.

To view a request template, go to http://example.reactivity.com/

In the example page, click on the name of the **Retrieve quote** service operation.

Figure 10-1: Service Operation Information



A request message template appears, along with a response example. Use the request message template to create your own message to send the ACE XML Gateway from an HTTP client tool, as follows:

1. Open WFetch and in the message configuration page, set the **Verb** option to **POST**.
2. For the **Host** field, type the hostname of the ACE XML Gateway, such as `xmlgateway.organization.com`
3. For the **Port** field, type `80`.
4. In the **Path** field, type the exposed local path of the service to test. As configured in "Defining a Web Service by WSDL Import," of "Virtualizing Services," the **Path** field should be: `/orders`

   The **Ver**, **Auth**, and **Connect** fields can remain at their default values, `1.1`, `Anonymous`, and `http`, respectively.
5. For the **Advanced Request** options, choose `Add Headers&Body`.
6. In the text field, add the header and body of the test message, as shown in Listing 10-1. Notice that a blank line should separate the HTTP headers from the body.

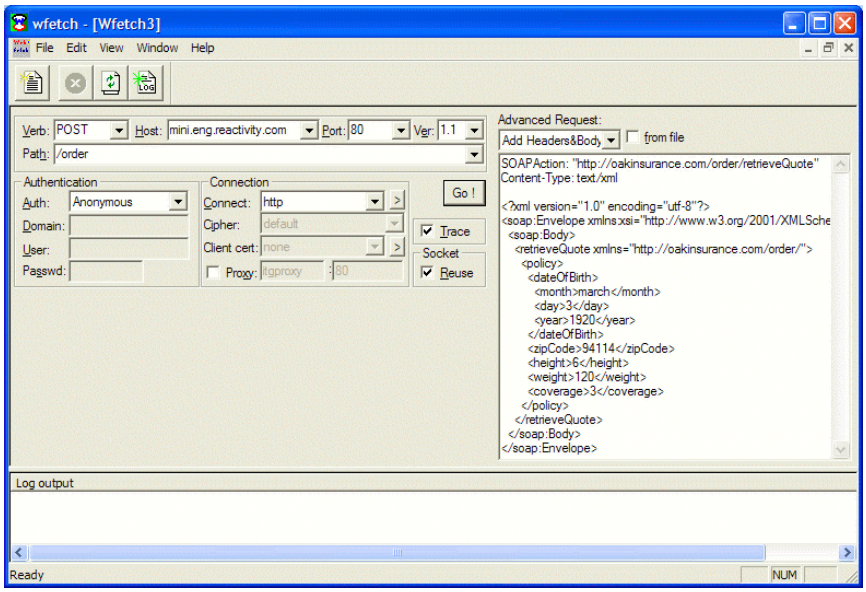
Listing 10-1: SOAP Message

```
SOAPAction:
"http://oakinsurance.com/order/retrieveQuote"
Content-Type: text/xml

<?xml version="1.0" encoding="utf-8"?>
 <soap:Envelope

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema"

xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
 <soap:Body>
  <retrieveQuote
xmlns="http://oakinsurance.com/order/">
    <policy>
      <dateOfBirth>
        <month>January</month>
        <day>1</day>
        <year>1960</year>
      </dateOfBirth>
      <zipCode>94002</zipCode>
      <height>52</height>
      <weight>150</weight>
      <coverage>402</coverage>
    </policy>
  </retrieveQuote>
 </soap:Body>
</soap:Envelope>
```

If copying the message template from example.reactivity.com, be sure to replace template text with sample values, as shown in the listing.

The WFetch message configuration page should appear as follows:

Figure 10-2: WFetch Message Configuration Page



7. Click the **Go!** button to send the request.

   The response should appear in the **Log Output** field, as shown in Figure 10-3.

Figure 10-3: WFetch Response

Notice that the response includes a `retrieveQuoteResult` element, with a sample price, expiration date, and so on.

Generating and Publishing WSDL's

The Manager can generate WSDL's for SOAP services defined in the policy. A WSDL is useful for a variety of applications—it can be used in some development environments to generate client code for services, for

example, or in testing tools to populate testing frameworks. WSDL's generated from the Manager show the ACE XML Gateway as the location of the service in its port definition.

A WSDL represents a handler group in the policy. The handler group may contain any number of service proxies, handlers, or service descriptors. Only SOAP-based services, however, will be described in a generated WSDL.

To create a WSDL for a handler group:

1. In the Routing Browser, click **generate WSDL** link next to the handler group heading for the order services.

   The WSDL appears in the browser window.

2. Click the browser's Back button to return to the routing browser.

3. After a WSDL has been generated for a handler group, you can save the WSDL file to your filesystem in the Routing Browser by right-clicking on the **generate WSDL** link and choosing **Save Target As** from the menu.

   For a new service definition, a WSDL is not available until either you deploy the policy or manually generate the WSDL as described above.

To make the WSDL available to partners or clients, you can publish it in protected fashion (using the ACE XML Gateway service directory) or publicly (at a URL reachable through the ACE XML Gateway). In the latter case, the WSDL for a given handler group is exposed at the following path:

`http://<gateway_hostname>:<port>/<service_path>?WSDL`

To publish WSDL's to a URL, configure the **WSDL Publishing** settings on the **System Management** > **Gateway Settings** page. Specifically, enable the option **Serve WSDL Files from the Gateway**.

The next time you deploy the policy, and every time thereafter, the ACE XML Gateway will automatically publish WSDL's for all handler groups in the policy to the publicly accessible URL.

Using Event and Message Logs

The ACE XML Gateway provides a range of tools and capabilities for analyzing and monitoring activities in a service-oriented environment. Not only can these tools help administer and troubleshoot the ACE XML Gateway activity, more broadly, they can help you troubleshoot activity external to the ACE XML Gateway. The ACE XML Gateway's message and event logs can significantly assist in pinpointing the source of inter-operability problems and application errors between service endpoints.

The following procedures introduce you to the logging tools provided by the ACE XML Manager.

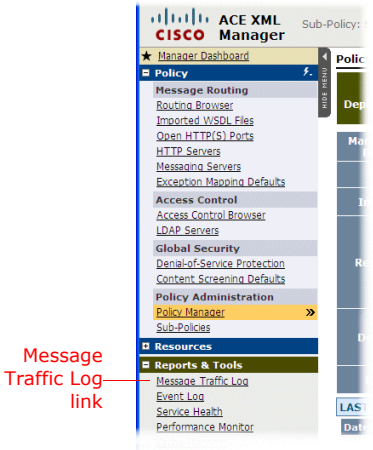
Viewing the Message Traffic Log Information

In "Defining a Web Service by WSDL Import," you set the default logging level for the `retrieveQuote` service proxy so that inbound and outbound messages are logged. After testing the service as described in "Sending Traffic to the XML Gateway," you can view the logged content.

To view the log:

1. Click the **Message Traffic Log** link in the navigation menu.

Figure 11-1: Message Traffic Log link

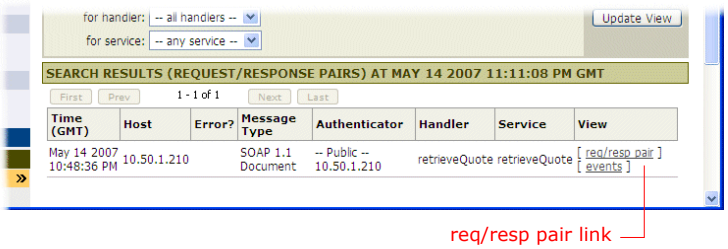


The Message traffic log appears. Note that the log records messages as request/response pairs.

Clicking the **events** link under the **View** column displays the overview information on the message event. For now though, let's take a look at the message body.

2. Click the **req/resp pair** link in the **View** column:

Figure 11-2: Accessing Events



A new window appears with information on each leg of the message interchange: incoming request, outgoing request, incoming response and outgoing response.

3. Scroll down to the **Outgoing Response Attributes** area of the page and click the link next to the **Body** label, **text/xml**.

   The content of the message appears.

Figure 11-3: Outgoing Response Message Body

The window displays the complete text of the outgoing response message. Similarly, you can view the text of the messages sent as the outgoing request, incoming response, and so on.

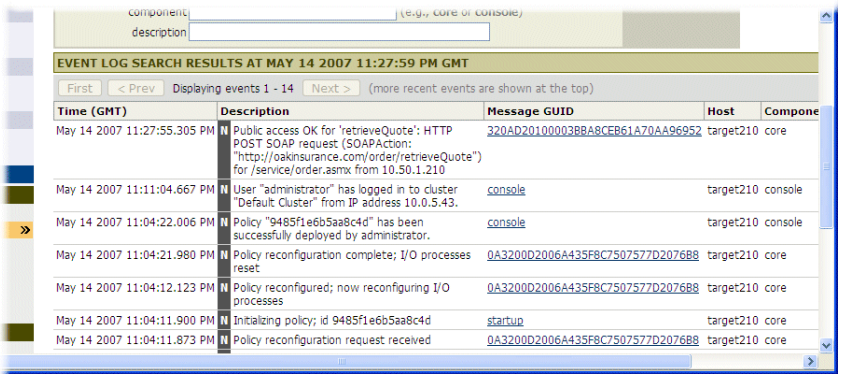
When applying complex configuration settings to a message specification, viewing the messages in each of its four forms in crossing the Gateway helps you to understand the effects of the settings on the message.

Understanding Log Levels

The previous steps describe how to view message text in the message traffic log. The message log helps you to understand and troubleshoot message handling components in a system deployment.

The event log provides additional information on message processing. It also displays information on other activities of the system, such as policy changes, state changes of the ACE XML Gateway and ACE XML Manager (such as restart events), application faults returned by the protected services, and many other types of events.

Figure 11-4: Test transaction in the event log



Every event in the log is associated with a log level. The access event resulting from the service test you just issued, for example, is logged in the event log at the notice level, indicated by the *N* description category.

There are six severity levels, as listed in Table 11-1.

Table 11-1: Log Severity Levels

Name	Description
Alert	A critical system condition requiring immediate attention to prevent system failure.
Error	An error condition that may cause incorrect system behavior.
Warn	An error condition that may cause unexpected system behavior (for example, a backend HTTP response code is 500, Manager login attempt fails, message validity check fails).

Table 11-1: Log Severity Levels

Name	Description
Notice	A normal, but significant condition (such as a request and response is allowed through the ACE XML Gateway, a request message does not match a handler, a user successfully logs in to the Manager)
Info	An informational message. Significant steps in the handling of a message.
Debug	The most verbose log level, includes reports of events that may be useful when developing or troubleshooting a system.

The system only shows those events that reflect the logging level selected in the Manager, which is at notice level or higher by default. For testing, you may wish to set the level to debug or higher. For a production deployment or for performance testing, we recommend setting the logging level to notice level or higher. As may be expected, more verbose logging results in slower system performance.

You can change the logging levels from the System Management page. Notice that the levels can be separately configured for the activities of the Manager and Gateway. The Manager events are focussed on administrative activities in the system, whereas the Gateway events provide detailed information on traffic processing. For policy development and debugging purposes, setting Gateway logging to debug is particularly helpful.

PART III. SECURITY FEATURES OF THE XML GATEWAY

Part III describes how to build upon the policy settings that resulted from WSDL import. It takes you through the steps for setting up security and message validation features of the system.

The following topics are covered:

- Controlling Access to Services
- Securing Traffic with SSL
- Validating Messages
- Using Content Screening Rules
- Preventing Attacks
- Encrypting and Decrypting XML Content
- Creating and Verifying Digital Signatures

Controlling Access to Services

The ACE XML Gateway protects services by enforcing access control restrictions on service requests. The ACE XML Gateway can evaluate a wide range of credential types in incoming requests, including:

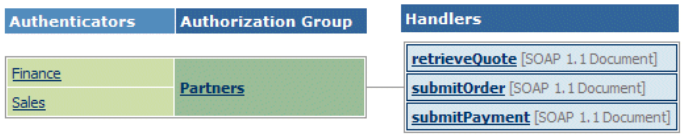
- Certificates
- HTTP Basic Authentication headers
- WSS UsernameToken
- WSS Password Digest
- Username/password combinations found in the message by XPath expression
- Netegrity token
- SAML token
- IP address
- Time of day

Password-based credentials can be verified against data in the ACE XML Gateway policy or by external mechanisms, such as Netegrity SiteMinder, LDAP, Active Directory, or a SOAP service.

You define access conditions for services using a policy object called an authenticator. All conditions in an authenticator must be satisfied by a request for it to be accepted by the authenticator.

An authenticator is associated to a service proxy through an authorization group. The authorization group ties one or more authenticators to one or more service definitions, as shown in Figure 13-1.

Figure 13-1: Access Control Policy Components



If there is more than one authenticator associated with a service, a request must meet the conditions of just one of the authenticators to access the service. The ACE XML Gateway applies optimization strategies to credential evaluation. If there are several authenticators associated with a service, it evaluates requests against the authenticator that has conditions that can be checked quickly (such as IP address restrictions) before those with more costly conditions (for example, involving cryptographic operations).

As another optimization strategy, once a request is accepted by an authenticator, the ACE XML Gateway does not consider conditions of other authenticators. Note that, as a result, the event log shows only the authenticator that actually accepted the request, not all authenticators that could have accepted it.

Controlling Access by IP Address

If you've been following the steps in this guide closely, after creating service proxies in "Virtualizing Services" on page 23, you set the access level for the `retrieveQuote` service proxy to *public*. At this level, once the ACE XML Gateway matches an incoming request to a service proxy, it does no further credential checking. In an actual deployment, it's more likely that access to most services would be restricted in some manner.

The following steps describe how to set up an authenticator based on the consumer's IP address and associate it with the service proxy:

1. In the navigation menu, click **Access Control Browser**.

   Notice that the `retrieveQuote` service appears in the Public table.

2. Click the **Add an Authenticator** button.

3. In the **Edit General Information** page for the authenticator, type a name for the authenticator, such as `BeaglePartners`.

4. For the **Authorization Group**, type a name for a new authorization group, such as `Partners`. (**Create in new authorization group named** should be selected as the drop-down menu option.)

5. Click **Create**.

   The credentials page for the authenticator appears.

6. With `IP Address` selected in the **Add Credential** menu, click **Add**.

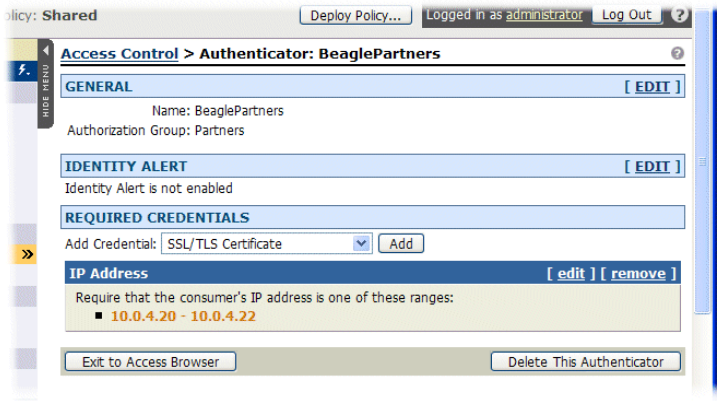
7. In the **IP Address** page, type a range of IP addresses to be permitted in the start and end range fields.

   To check that the ACE XML Gateway properly blocks unauthorized clients, first enter a range that does not include the address of the computer from which you are issuing test requests.

8. Click **Save Changes**.

The authenticator page appears again with the IP address requirement listed below the required credentials.

Figure 13-2: IP Address Requirement



Now associate a handler with the authorization group as follows:

1. Return to the Access Control Browser by clicking **Exit to Access Browser** button at the bottom of the authenticator page.

   Notice that the authenticator and authorization group you just created appear.

2. Click on the name of the `retrieveQuote` service proxy, which should appear in the Public table.

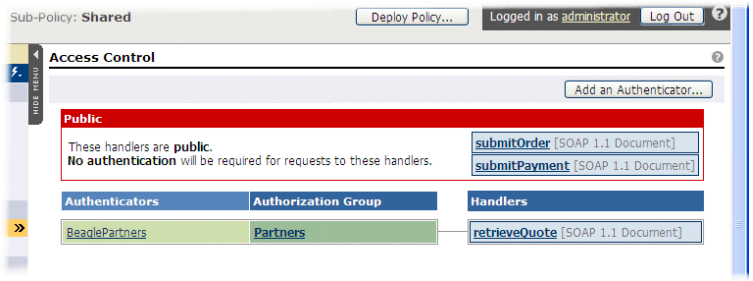
3. On the access control page for the service proxy, click the radio button labelled **Access is restricted to the following authorization groups**.

4. Click the checkbox next to the authorization group you just created, Partners.

5. Click **Save Changes**.

   The Access Control Browser appears again, this time with the authorization group you created linked to the `retrieveQuote` service proxy.

Figure 13-3: Service Proxy with an Authorization Group

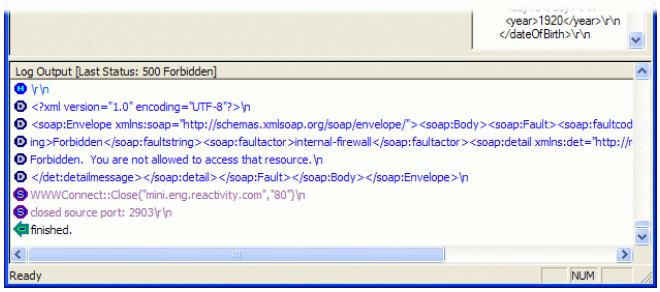


6. Deploy the policy to have your changes take effect.

After deploying the policy, issue a test request as described in "Sending Traffic to the XML Gateway."

If testing from a client with an IP address outside of the IP range you entered for the authenticator, the Gateway returns a "Forbidden" response, as shown in Figure 13-4.

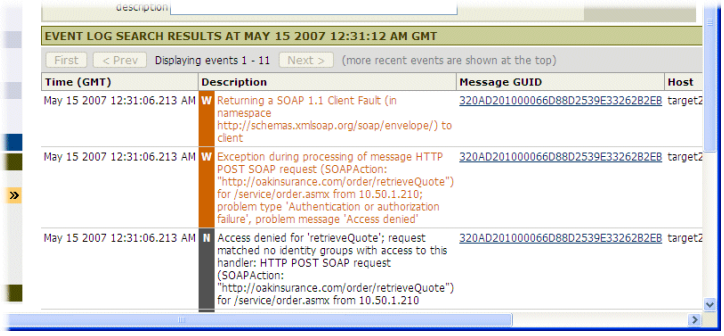
Figure 13-4: Unauthorized Access Response



To see how the event log records access block events, click the **Event Log** link in the navigation menu.

Notice that the blocked access attempt is logged at the warning level in the event log:

Figure 13-5: Access Blocked Log Event



Controlling Access by Username/Password Credentials

The ACE XML Gateway provides a great deal of flexibility for checking password-based credentials. The credentials can be validated against data kept on the Gateway itself or by an external system, such as an LDAP directory.

The following steps describe how to set up password check against an HTTP Basic Auth header in the incoming request.

1. If not already open, click **Access Control Browser** link in the navigation menu.

We'll add another authenticator to the authorization group you created in the previous steps.

2. Click the **Add an Authenticator** button.
3. Type a name for this authenticator in the **Authenticator Name** field, such as Beagle Basic Auth.
4. From the **Authorization Group** menu, choose the name of the authorization group you created earlier, Partners.
5. Click **Create**.

   The credentials page for the authenticator appears.
6. Select **HTTP(S) Basic Authentication** from the **Add Credential** menu, and click **Add**.
7. In the **HTTP(S) Basic Authentication** page, keep the default **Verify Using** menu selection of **Fixed Values**.

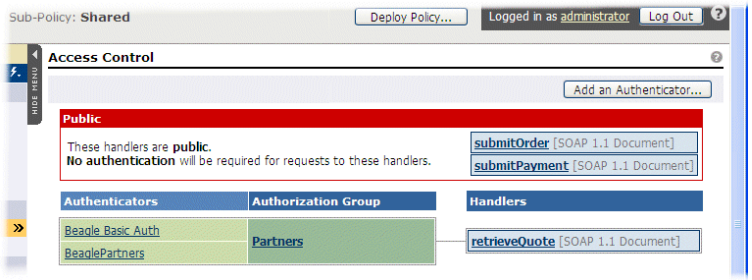
   Notice that other options in this menu include password file verification (which is a .htaccess-style file with username/password combinations) and various LDAP verification mechanisms.
8. In the **Username** field, type a username to be accepted by this authenticator.
9. In the Password fields, type the password to be accepted.
10. Click **Save Changes**.

    The authenticator page appears with the **HTTP(S) Basic Authentication** requirement listed under the required credentials.

    Identity reporting allows you to view service activity by user. We'll enable it for the new authenticator.
11. Click **Edit** next to the **Identity Reporting** header in the authenticator page.
12. Click the **Enable Identity Reporting** checkbox.
13. Enable the checkbox next to **HTTP Basic Auth Username**, the type of the credential requirement you just configured.
14. Click **Save Changes**.

The authenticator page appears again. Click **Exit to Access Browser** to see the effects of the new configuration in the access browser.

Figure 13-6: Service Proxy with an Authorization Group



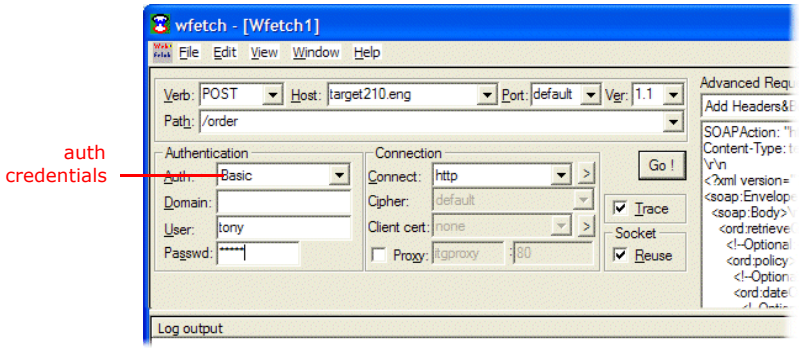
Notice that a consumer can access the retrieveQuote service by meeting *either* of the conditions defined in the two authenticators: the password requirement or the IP address requirement you set up earlier. If we had added the password requirement to the BeaglePartners authenticator instead, alongside the IP requirement, a consumer would have to meet both conditions to access the service.

To properly test the password requirement, you must set the IP address condition to block the host from which you will be testing (or delete the authenticator from the policy).

Deploy the policy to complete the configuration and test the new credential requirement.

From WFetch, prepare a request to send to the retrieveQuote service at the Gateway, as before. However, enter authentication credentials by choosing **Basic** as the **Auth** method and, in the **User** and **Passwd** fields, type the username and password you entered for the authenticator.

Figure 13-7: WFetch request with Basic Auth Credentials



Click **Go!**. Since we're going to check performance data for the service, once you get a successful response, submit the request several times again to generate additional performance data.

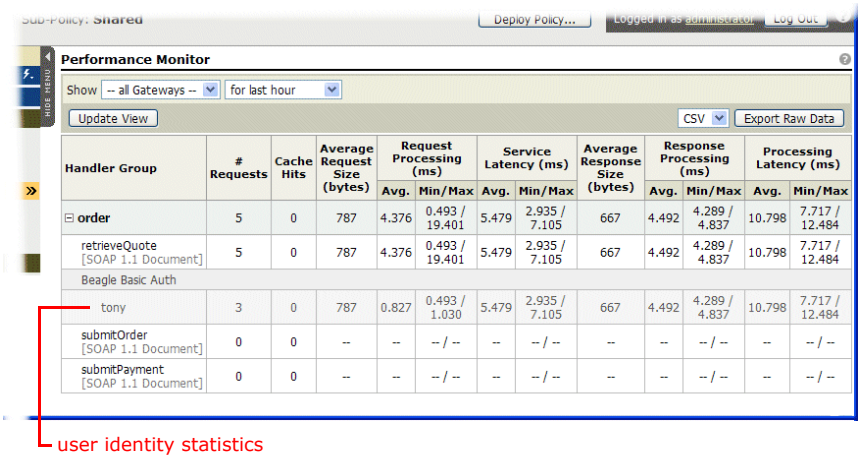
Now we can view service activity for the user identity we configured:

1. In the Manager web console, click **Performance Monitor** in the navigation menu.

2. Click the expand control next to the name of the handler group for the retrieveQuote service, order.

Notice that the monitor displays statistics for the identity of the user you created, along with totals for the service operation as a whole.

Figure 13-8: User activity in the performance monitor



In this case, the authenticator is set up to validate only one identity. If the authenticator had been set up to verify passwords against a password file or LDAP directory, there will be as many identities as there are requests from valid accounts in those credential sources in the monitor.

For details on each statistical category in the performance monitor, access the online help from the page.

Securing Traffic with SSL

The ACE XML Gateway can use SSL/TLS to secure traffic with the consumer, as well as to secure traffic between itself and the backend server. This example describes how to set up SSL/TLS between the ACE XML Gateway and the consumer.

> **Note:** For the rest of this document, SSL/TLS will be abbreviated to just SSL. Strictly speaking, versions 3.0 and below are SSL, while version 3.1 is TLS.

The SSL specification requires that the server present an X.509 certificate as part of its SSL handshake. Many clients will terminate the SSL handshake and abort the communication attempt if the certificate presented does not meet certain conditions.

The exact conditions that result in connection termination vary from client to client, but generally, a certificate presented by the ACE XML Gateway for establishing SSL connections should have these properties:

- It should be valid (the current date should be between its "valid from" and "valid to" date)
- It should have the "Server Authentication" extended key usage
- Its DN should have a CN equal to a host name that resolves to the server's IP address
- The client should be configured to trust the server certificate's issuing Certificate Authority (CA), and the server certificate should not appear on the CA's certificate revocation list (CRL)

Most implementations of the system will involve acquiring a CA-signed server certificate that meets these requirements and installing it on each Gateway in your deployment.

The Manager console includes tools for generating a certificate signing request and uploading the signed certificate. For the following walkthrough, however, you will use an example certificate available on the ACE XML Gateway sample resources web site.

About SSL Certificate Authentication

The ACE XML Gateway can establish an SSL session with a client without verifying the client certificate, but in most cases, SSL will be used alongside client certificate verification.

The ACE XML Gateway can validate certificates presented to authenticate clients in several ways. The simplest method is by thumbprint match, in which the ACE XML Gateway compares the certificate presented by a consumer to the one configured as an identity requirement. If the thumbprints match, the consumer is considered authorized.

The ACE XML Gateway can also authorize a consumer based on the certificate authority that issued the client certificate.

This example shows you how to authenticate by thumbprint match. In these steps, you first create a new authenticator for the certificate credential. Then you set up a handler to listen on a secure port.

What You'll Need

If you do not have your own certificates to use, for evaluation purposes, you can use certificates from the resource sample pages at:

http://example.reactivity.com/security.html

Before starting, download a server certificate for one of the Oak servers (at the bottom of the page) in PKCS #12 format (the following steps use the first P12 certificate in the list of Oak Server certificates, `maple.p12`). Recall that PKCS #12 files are often password protected. The example certificates are protected by the password `swordfish`.

Also, you will need a PEM client certificate and P12 public/private key pair from one of the fictional trading partners of the Oak security team, such as Beagle Partners, Inc., from the examples Web page.

Opening the HTTPS Port on the XML Gateway

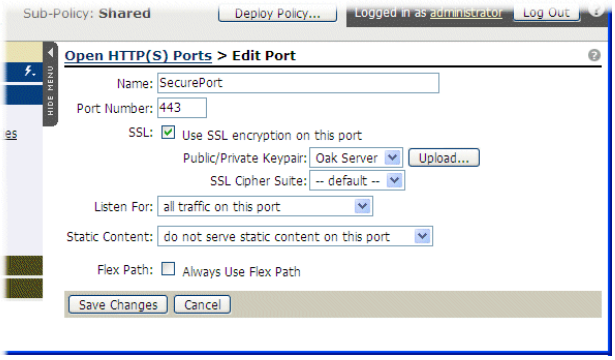
The first step is to open a listening port on the ACE XML Gateway that will be configured to use SSL:

1. Click the **Open HTTP(S) Ports** link in the navigation menu.
2. In the **Open HTTP(S) Ports** page, click the **Add a New Port** button.
3. Type a name for the port, such as `SecurePort`. This name is only used to identify the listening port within the Manager.
4. For **Port Number**, enter 443, the conventional port used for SSL.
5. Select the **SSL** check box.
6. Next to the **Public/Private Keypairs** label, a message should appear that no public/private keypairs have been uploaded. Click the **Upload** button next to the message.

7. In the upload resource window, type a name for the resource, such as `Oak Server`.
8. Click the **Browse** button and in the file chooser dialog, locate the Oak server P12 file you downloaded from the sample page, `maple.p12`.
9. In the **Password** field, type `swordfish`.
10. With the P12 file in the **Public/Private Keypairs** field, click **Upload**.

    The **Edit Port** page should appear similar to the following:

Figure 14-1: Edit Port Page



11. Click **Save Changes**.

The new port should appear in the port list.

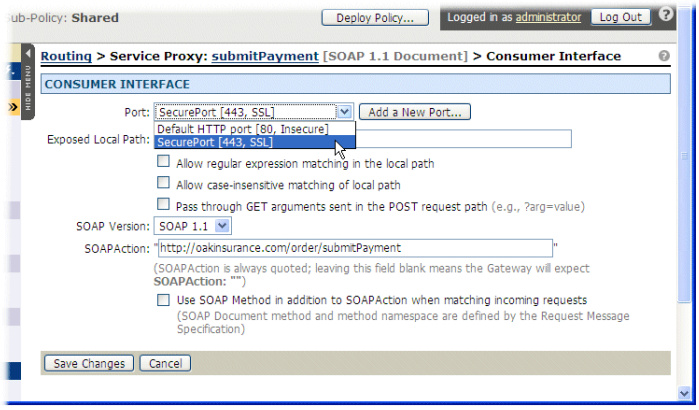
Configure the Service Proxy to Use the Secure Port

Now set up the service proxy to use the secure port as follows:

1. Click the **Routing Browser** link on the navigation menu.
2. Click on the name of the `submitPayment` service proxy.

   The information page for `submitPayment` appears.
3. Click the **Edit** link next to the **Consumer Interface** heading.
4. From the **Port** drop-down menu, choose the new port you created, SecurePort.

Figure 14-2: Changing Consumer Port



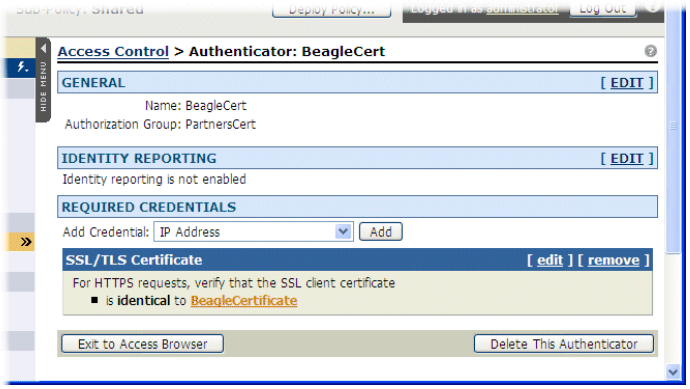
5. Click **Save Changes**.

Creating the Certificate Access Requirement

Create a new authenticator as follows:

1. Click **Access Control Browser** on the navigation menu.
2. In the Access Control Browser, click **Add an Authenticator**.
3. Provide a name for the authenticator, such as BeagleCert, and add it in a new authorization group, PartnersCert.
4. Click **Create**.
5. From the **Add Credential** menu, choose the **SSL/TLS Certificate** item and click **Add**.
6. In the **SSL/TLS Certificate** page, with the default verification method selected, **SSL Certificate Fingerprint**, click **Upload** to upload a certificate file to the ACE XML Gateway.
7. In the **Upload Consumer Certificate Resource** window, enter a name in the **Resource Name** field, such as BeagleCert. This name identifies the certificate within the policy.
8. Click the **Browse** button next to the **Local File** field and navigate to the certificate file you downloaded from the sample security resources page, `beagle.pem`.
9. With the path to the certificate file populated in the **Local File** field, click **Upload**.
10. Click **Save Changes**. The new certificate requirement appears in the credential list.

Figure 14-3: Credential List



11. Click the **Exit to Access Browser** button.
12. Click on the SubmitPayment service proxy on the right side of the Public table of the Access Control Browser.
13. In the access control for the service proxy page, choose **Access is restricted to the following authorization groups**, and select the checkbox for the group you created.
14. Click **Save Changes** and deploy the policy.

Now try issuing a request for the service proxy, passing the Beagle client certificate with the request, as described next.

Testing Certificate Access Requirement

To pass a certificate in a request with WFetch, you first need to upload the certificate into Internet Explorer (WFetch can pass certificates in IE along with requests). The following describes how to import the certificate to Internet Explorer 6.0. Alternatively, you can test certificate acceptance with the Curl command line HTTP client tool.

If needed, download the Beagle Partners P12 Certificate/keypair from the samples page, if you have not done so already:

http://example.reactivity.com/security.html

After downloading the certificate, follow these steps:

1. In Internet Explorer, click **Tools > Internet Options**.
2. Choose the **Content** tab, and click the **Certificates** button.
3. Click the **Import** button and use the Certificate Import Wizard to import the `beagle.p12` file into Internet Explorer.
4. If prompted, supply the password for the file, swordfish.

Make the following changes in WFetch before issuing a request:

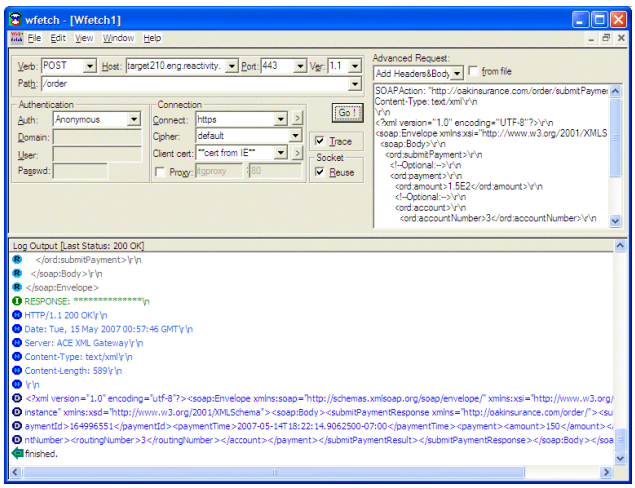
- Choose 443 as the **Port**.
- For the **Connect** field, choose `https`.
- Leave the **cipher** option at `default`.
- For the Client cert, choose `** cert from IE **` . If you have multiple certificates in Internet Explorer, choose the certificate you just uploaded.
- In the **Headers and Body** field, paste the following:

Listing 14-1: SubmitPayment Request

```
SOAPAction: "http://oakinsurance.com/order/submitPayment"

<?xml version="1.0" encoding="utf-8"?>
 <soap:Envelope
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
   <submitPayment xmlns="http://oakinsurance.com/order/">
   <payment>
    <amount>246.46</amount>
    <account>
      <accountNumber>123456</accountNumber>
      <routingNumber>789123</routingNumber>
    </account>
   </payment>
  </submitPayment>
 </soap:Body>
</soap:Envelope>
```

When you click **Go!**, the response should appear as follows:

Figure 14-4: WFetch with Certificate Configuration



Including a Certificate with a Request from Curl

With Curl, you can issue the test message with the following command (the entire command should be entered at the command line at one time):

```
curl -E beagle.pem:swordfish -k -v
  -H 'Content-Type: text/xml' -H
'SOAPAction:
"http://oakinsurance.com/order/submitPayment"'
    --data-binary @- https://10.0.101.73/order
    < payment.xml
```

Note that the password is passed in the command along with the Beagle certificate, beagle.pem in this case. Also, `payment.xml` should contain the body content passed to the service, which should have the following contents.

Listing 14-2: payment.xml Contents

```
<?xml version="1.0" encoding="utf-8"?>
 <soap:Envelope
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
   <submitPayment xmlns="http://oakinsurance.com/order/">
   <payment>
    <amount>246.46</amount>
    <account>
      <accountNumber>123456</accountNumber>
      <routingNumber>789123</routingNumber>
    </account>
   </payment>
  </submitPayment>
 </soap:Body>
</soap:Envelope>
```

Validating Messages

By validating messages, the ACE XML Gateway removes the burden from backend resources of having to contend with invalid requests.

The ACE XML Gateway can validate messages by:

- Making sure they conform to an XML schema or DTD
- Checking for XML well-formedness, without checking against a schema or DTD
- Ensuring the presence and values of parameters

When a message is found to be invalid, the ACE XML Gateway can log the event and, optionally, send a notification.

This section describes how to set up message validation against an XSD and argument validation. For complete information, see the Cisco ACE XML Gateway *User's Guide*.

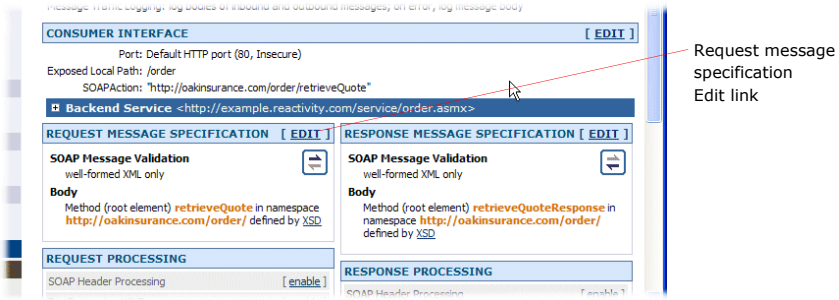
Validating Message Body

This section walks you through the steps for validating an incoming SOAP request against an XML schema. Schema validation occurs before any other processing tasks are applied to a message by the ACE XML Gateway.

To set up XML schema-based validation:

1. Access the information page for the service proxy for which you would like to configure request validation.
2. In the service proxy's information page, click the **edit** link next to **Request Message Specification**.

Figure 15-1: Accessing the Request Validation Page



For a SOAP service, **SOAP Message Validation** options appear.

3. To validate the message with an XML schema, choose the second option: **Content: require SOAP message validation with the specified XML schemas; reject invalid message**.
4. If the message is Document-style SOAP, the Manager has already extracted the XML Schema file from the WSDL file. You do not need to upload the schema manually.

   If the service is not a Document-style SOAP service but is intended to handle XML body content, the required XML-Schema file is most likely not stored in the Manager yet. Click the **Upload…** button to choose and load the schema.
5. Click **Save Changes**.

In the service proxy information page, SOAP Message Validation is now indicated for the request message specification with **XML schema-based content validation**.

Once you deploy the policy, messages that fail validation are now blocked. To test schema validation add an element to the order request body that is not included in the original sample request, without breaking XML well-formedness, and try sending the request to the ACE XML Gateway. The ACE XML Gateway should return a SOAP fault response with a `faultstring` of "Validation Error."

Argument Validation

For protocols that support arguments, such as SOAP and HTTP/GET, the ACE XML Gateway can validate the arguments passed in a message. Like message body validation, argument validation ensures that messages that reach the backend make sense for the applications that have to process them.

For a given parameter you can specify:

- whether it is required
- the expected type of the value
- content requirements of the value

The parameter editor is accessible from the service proxy information page for services whose protocols support arguments.

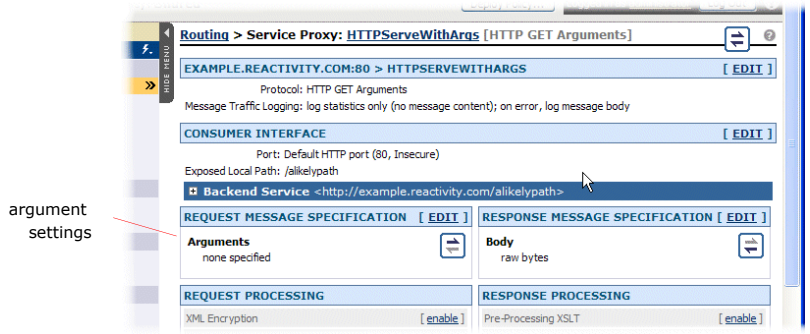
To specify argument requirements:

1. In the Routing Browser, click the name of the service proxy for which you want to configure argument requirements.

   The service proxy information page appears.

If the service supports parameters, the argument setting appears in the **Request Message Specification** and **Response Message Specification** areas of the Service Proxy information page.

Figure 15-2: Service Proxy Argument Settings



2. Click the **Edit** link next to **Request Message Specification**.
3. Click the **Add a New Row** button.
4. In the **Name** field, type the name of the parameter as identified in the message.
5. Use the other controls on the **Request Message Specification** page to specify whether it is a required argument, its type, and whether to validate its content, type, or both. For XML type parameters, you can additionally specify an XML schema or DTD against which the parameter is to be validated.

   If want to add arguments that may not be in the request, but want to verify the argument by schema validation if it is there, for example, you can leave the **Req.** check box unselected.
6. Add a new row for each parameter you want to check.
7. When finished, click **Save Changes** and deploy the policy to have your changed take effect.

The ACE XML Gateway will make sure that the arguments you specify are in the messages that reach the destination service.

Using Content Screening Rules

The ACE XML Gateway can detect and block specific threats embedded in XML content of the message. An example of such a threat is a SQL insertion attack, in which SQL commands are embedded in XML data in the attempt to get a backend server to execute those commands against a database.

You can control such content-based threats through content-screening rules. The ACE XML Gateway includes a number of pre-written screening rules that cover many known threats. In addition, you can add your own content screening rules as desired.

You can enable or disable content screening rules, as well as create new ones, from the **Content Screening Default** page. The **Content Screening Default** page displays the status of the content screening rules and the rule details for each, including:

- the expression that comprises the rule
- the log event generated when the rule is matched
- whether the rule text is case-sensitive

Enabling Content Screening Rules

To enable or disable content screening rules:

1. Click the **Content Screening Defaults** link from the navigation menu.
2. For a particular screening rule, change the applicability setting from **disable** to **enable**. This default screening rule applies to all handlers unless otherwise specified in the handler configuration.
3. Click **Save Change to Default Setting**.

Note that service proxies can override the global status setting for a given rule (enabled or disabled). You can access service proxy specific rules settings from the **Content Screening** settings area of the service proxy information page.

Testing Content Screening

You can test content screening by sending a message to the ACE XML Gateway that contains a string that matches screened content. For example, to trigger screening on the `retrieveQuote` handler created earlier, follow these steps:

1. Click the **Content Screening Defaults** item from the navigation menu.
2. Enable the content screening rules labeled SQL Commands (v. 2).
3. Click **Save Changes to Default Settings** and deploy.

Now send the following request to retrieveQuote service:

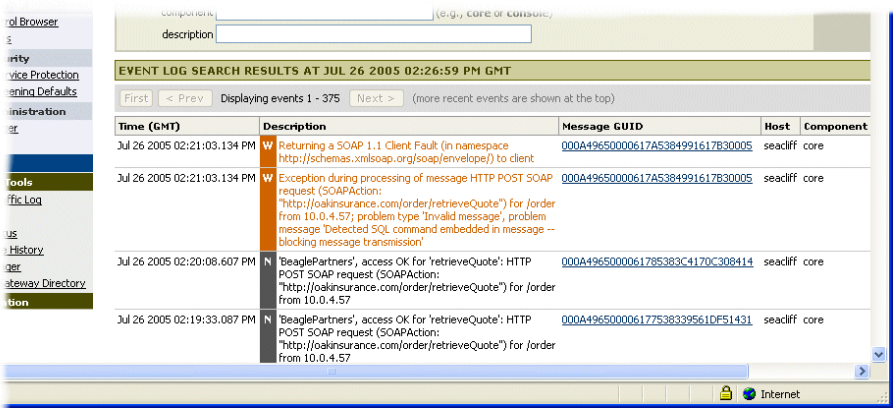
Listing 16-1: SOAP Message

```
<?xml version="1.0" encoding="utf-8"?>
 <soap:Envelope

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"

xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
 <soap:Body>
  <retrieveQuote
xmlns="http://oakinsurance.com/order/">
    <policy>
      <dateOfBirth>
        <month>January</month>
        <day>DROP TABLE</day>
        <year>1960</year>
      </dateOfBirth>
      <zipCode>94002</zipCode>
      <height>52</height>
      <weight>150</weight>
      <coverage>402</coverage>
    </policy>
  </retrieveQuote>
 </soap:Body>
</soap:Envelope>
```

The message includes content that may indicate a SQL-injection attack, DROP TABLE. When submitted to the ACE XML Gateway, the message is blocked.

Notice that the log entry generated by a block event indicates that the ACE XML Gateway detected a SQL command embedded in the message.

Figure 16-1: Content blocked log entry



Creating a Content Screening Rule

You can supplement the pre-written content screening rules with your own, custom content screening rules. The rules are specified as regular expression statements.

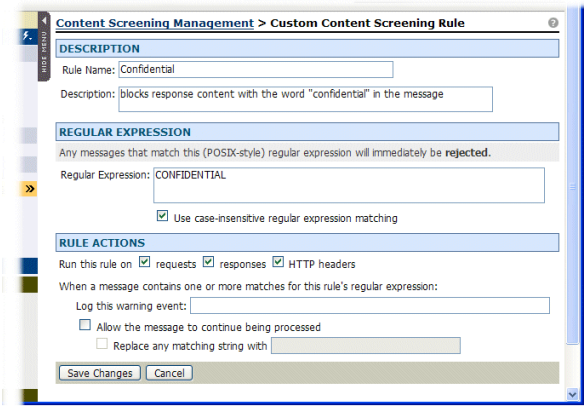
If a comparison between the message and the statement produces a match, the message is blocked.

To create a custom screening rule:

1. Click the **Content Screening Defaults** link from the navigation menu.
2. If it does not appear on your screen, scroll down until the **Custom Content Screening Rules** area is visible.
3. Click **Define a New Rule**.
4. Provide a rule name to identify the rule in the console, and a description.
5. In the **Regular Expression** field, enter the regular expressions that match the content you want to screen.
6. In the **Rule Actions** fields, you can specify how you want the rule to be applied to traffic.

   Notice that as an alternative to blocking a message, you can allow the message to continue, with the matched content replaced by a string you specify. This is useful in outgoing messages to hide sensitive or private information that may be included in the message.

Figure 16-2: Sample Content Screening Rule Configuration



7. When finished, click **Save Changes** and deploy the policy to have the rule enforced at the ACE XML Gateway.

Preventing Attacks

The ACE XML Gateway detects when suspected denial-of-service attacks are underway, particularly the types of denial-of-service attacks to which web services are susceptible, XML Denial-of-Service (XDoS).

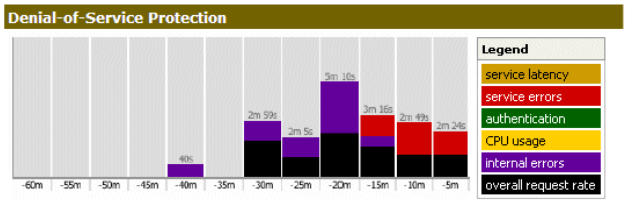
XDoS attacks attempt to take advantage of the overhead involved in processing XML messages to overwhelm service providers. The ACE XML Gateway serves as a barrier to the backend infrastructure, stopping DoS attacks before they can reach it.

Viewing Gateway Activity

When an XDoS attack is detected, the ACE XML Gateway logs the event and notifies monitoring systems. Additionally, the ACE XML Gateway can temporarily block traffic from the IP address that originated the traffic.

To see recently detected XDoS attacks, navigate to the Manager Dashboard. The **Denial-of-Service Protection** chart shows XDoS attacks detected in the previous hour.

Figure 17-1: Denial-of-Service Protection



As shown in the legend, the graph provides information on six indicators of possible XDoS attack:

- **Service Latency**. Triggered when the backend service latency repeatedly crosses the acceptable latency threshold (including timeouts).
- **Service Errors**. Triggered when the backend service returns an unusually high number of errors, such as SOAP faults or HTTP 500 errors.
- **Authentication**. Triggered when an excessive number of 401/403 errors are returned (HTTP Unauthorized/Forbidden).
- **CPU Usage**. Triggered when an inordinate number of CPU cycles on the ACE XML Gateway is required to process a message, for example if a single message were to include thousands of signatures to be verified.

- **Internal Errors**. Triggered by a large number of internal errors are detected, such as schema validation failures, malicious content, or invalid digital signatures.
- **Overall request rate**. Triggered when the number of incoming requests is excessive. This is similar to DoS attacks against web sites or other network servers.

You can view and modify the current thresholds for DoS attack detection from the **Denial-Of-Service Protection Settings** page.

Configuring Denial-of-Service Protection

To change the traffic thresholds that protect against denial-of-service attacks:

1. Under the **Policy** section of the navigation menu, click on the **Denial-of-Service Protection** link (under the **Global Security** subhead).
2. In the **Denial-Of-Service Protection Settings** page, modify the settings as appropriate for each of the categories of DoS attack.
3. If you'd like the ACE XML Gateway to block traffic from the IP address that originates a suspected DoS attack, check the checkbox **When an attack is detected, block the attacking IP address for at least 5 seconds**.
4. Click **Save Changes** and deploy the policy to have the changes take effect at the ACE XML Gateway.

Encrypting and Decrypting XML Content

In contrast to SSL/TSL encryption, which provides transport-level encryption, XML Encryption works at the message level. This imparts several benefits. For one, you can encrypt specific parts of a message, allowing the parts of the message that are not sensitive to remain in clear text. Another benefit is that the message can remain encrypted after it arrives at its destination, so that it remains protected until actually needed.

Encrypting Outgoing XML Content

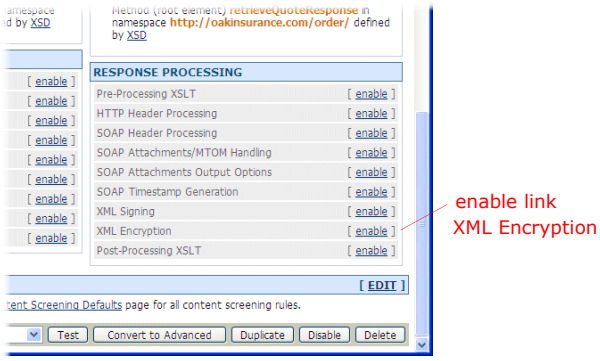
To encrypt message content, you need to use the public key of the intended recipient (who will need to use the corresponding private key to decrypt the content).

To use XML Encryption in an actual implementation, therefore, you will first need to load the consumer certificates that contain the public keys of your partners in the policy. Alternatively, the ACE XML Gateway can encrypt a message with the public key included in an XML Signature of the incoming request. For this example, we'll use a public certificate of the fictional Beagle Partners, Inc.

To set up message encryption:

1. In the routing browser, click the retrieveQuote service proxy.
2. Under the **Response Message Specification**, click the enable link next to **XML Encryption**.

Figure 18-1: Encrypting outgoing response



3. In the **XML Encryption** page, choose the beagle.cer certificate resource for the **Transport with Public Key**. If you have not yet uploaded this certificate, follow these steps:
   a. Click the **Upload** button next to the option menu.

b. Enter a name for the resource in the policy.

c. In the URL field, enter the following URL:

http://example.reactivity.com/pki/client/beagle.cer

You can leave these options at their default values: **For SOAP Role**, **Encryption Algorithm**, **Transport Cipher**, and **Encryption Type**.

In an actual implementation, you may need to customize these options based on the expectations of the receiving application.

4. Select the **Element specified by these XPaths** option and in the XPath field enter the following value:

```
//*[local-name()='price']
```

Only the `price` element of the response message will be encrypted.

5. Click **Save Changes**.

6. If you configured access control for the retrieveQuote service, for simplicity you should reset it to public access at this point. Doing so will make it easier to find encryption-related events in the event log.

To set access to public, click the **Edit Access Control** link at the bottom of the settings page and choose **Public** as the access level.

Also, you may want to resume using the default, unencrypted port 80. To change the port, modify the consumer interface setting for the service proxy, as described in the previous walkthrough.

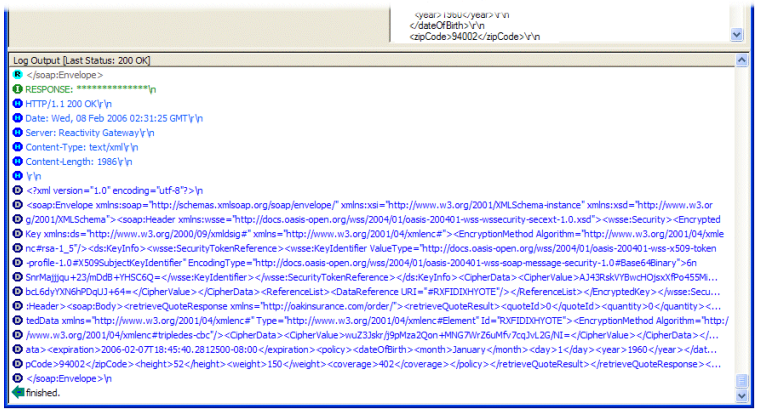
7. Be sure to save your changes and deploy the policy.

You can now test XML encryption, as described next.

Testing Encryption

After configuring the message specification for encryption, send a message to the service from WFetch. Be sure to modify WFetch settings to use the default port again and Anonymous authentication. A message similar to the following should appear as the response.

Figure 18-2: Encrypted response



There are several points to note about the response.

- A WSSE security heading has been added to the message describing the encrypted data.
- Included in the heading is a reference declaration that identifies the encrypted element: `<DataReference URI="#RXFIDIXHYOTE"/>`
- DataReference points to the encrypted `price` element, which in encrypted form appears as:

```
<EncryptedData
  Type="http://www.w3.org/2001/04/xmlenc#Element"
  Id="RXFIDIXHYOTE">
 <EncryptionMethod
    Algorithm="http://www.w3.org/2001/04/xmlenc#
      tripledes-cbc"/>
 <CipherData>
   <CipherValue>wuZ36uMfv7cqJvL2G/NI=</CipherValue>
 </CipherData>
</EncryptedData>
```

Notice the effect of configuration options on how the response was encrypted. Feel free to experiment with other options in the interface and viewing the effects on the response. For example, try using **encrypt only the contents of the specified elements** as the **Encrypt Element** choice, or specify additional elements to be encrypted by adding XPath rows to the configuration.

> **Note:** For a better formatted view of the message, open the response message in the message traffic log window instead of in WFetch.

As you experiment with the settings, notice how the ACE XML Gateway makes fairly complex technology much easier to implement. This applies not only to XML Encryption, but to other WS technologies as well, such as XML Signature, SAML, and WS UsernameToken.

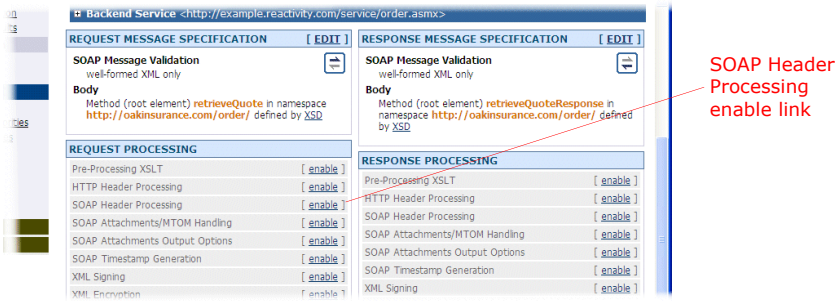
Decrypting Incoming XML Content

Encrypted content needs to be decrypted with the private key that matches the public key used to encrypt it. This implies that the partner who sent a message has the ACE XML Gateway's public key, and used it to encrypt the protected content. In the settings for decrypting the content, therefore, you would configure the ACE XML Gateway's private key as the key used for decryption.

To set up message decryption on a service proxy, follow these steps:

1. In the Routing Browser, click the service proxy that will issue the encrypted message.
2. In the service proxy information page, click the **enable** link next to **SOAP Header Processing** of the **Request Message Specification** settings.

Figure 18-3: Decrypting Incoming Request



3. Check the first checkbox, **Process header elements for SOAP Role**. Leave the drop-down menu option set to `no role`.
4. Under the `WSS:XML Decryption` heading, check **Enable XML decryption using the selected keys**.

   By enabling this option, you direct the ACE XML Gateway to decrypt encrypted SOAP elements, but only those that use the listed keys.
5. Select the private keys you would like to use for decryption. (You can use the **Upload** button to upload a certificate/key pair in PKCS#12 format, if needed).

When you **Save Changes**, a summary of the configuration should appear in the **Request Processing** area of the service proxy information page. Be sure to deploy the policy to have your changes take effect.

Creating and Verifying Digital Signatures

XML Signature allows users to ensure the authenticity and integrity of XML data, that is, that the data came from a particular source and has not been modified since the signature was generated.

The ACE XML Gateway makes it easy to apply XML Encryption to message processing. You can set up XML Signature generation on outgoing messages and signature validation of incoming messages.

To generate an XML signature, the ACE XML Gateway uses a private key to create a digest of the content you specify, either a part of a message or the entire message. Recipients will then use the ACE XML Gateway's public key to process the signature. If the public key can be used to decrypt the digest, the recipient can be sure that the message must have been generated by the holder of the equivalent (that is, the ACE XML Gateway's) private key. Similarly, to configure signature verification, you use the public key of the sender.

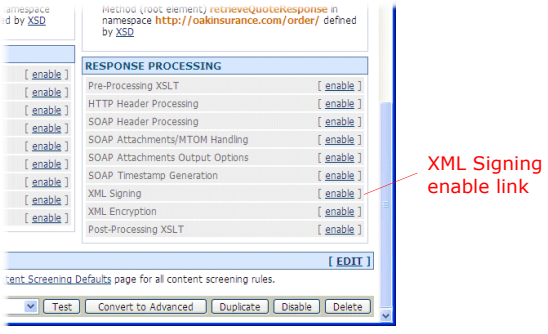
These steps describe how to sign content of outgoing responses. In them, we'll set up signing using one of the resource files on the sample resources web site, maple.p12 (the keypair for the maple.oakinsurance.com server). If you haven't do so already, get the resource from example.reactivity.com.

Signing the Response Message

To configure XML signing of an outgoing response:

1. In the routing browser, click the retrieveQuote service proxy.
2. Under **Response Message Specification**, click the enable link next to **XML Signing**.

Figure 19-1: Signing outgoing response



3. In the **XML Signature** page, click the **Upload** button next to the **Private Key** option.

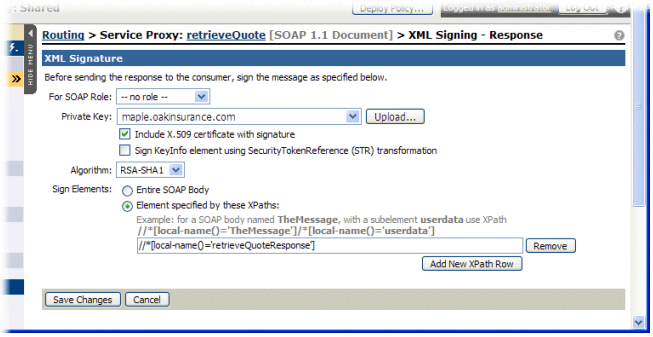
4. In the **Upload Public/Private Keypair Resource** window, type a meaningful name for the resource as it will be known in the policy, such as `maple keypair`.
5. Click **Browse** and, in the file chooser, navigate to and select the `maple.p12` file.
6. For the password, type `swordfish`.
7. Click the **Upload** button.
8. Make sure the resource is selected in the **Private Key** menu.

   You can leave these options at their default values: **For SOAP Role**, **Include X.509 certificate with signature** (which should be enabled), and **Algorithm**.
9. Select the **Element specified by these XPaths** option and in the XPath field enter the following value:

   ```
   //*[local-name()='retrieveQuoteResponse']
   ```

   This will result in the entire response element being signed.

Figure 19-2: XML Signature configuration



10. Click **Save Changes** and deploy the policy.

In these procedures, a specific element was signed. In practice, the entire SOAP message body is usually signed. Also, XML Signature and XML Encryption are often used together to ensure the safety and integrity of messages.

Testing XML Signature

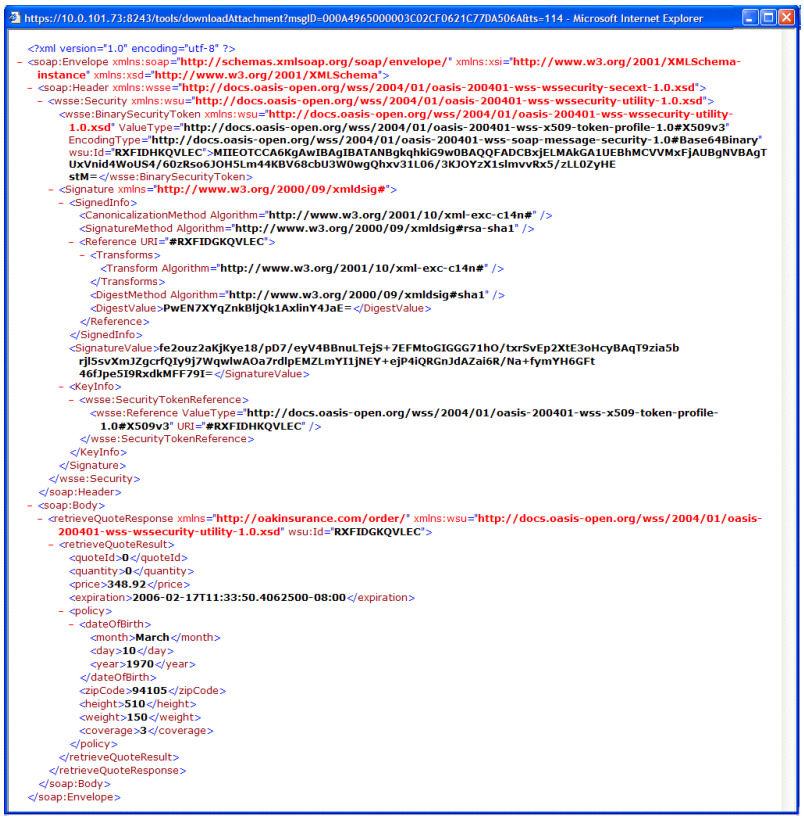
After configuring the message specification for XML signing, send a message to the service from WFetch. The results will be printed to the output pane in WFetch, but for a better view of the message, open it in the message traffic log as follows:

1. Click the **Message Traffic Log** link in the navigation menu.

2. Click on the **req/resp pair** link for the message entry.
3. In the **Logged Message Content** window, click the **[text/xml]** link in the **Outgoing Response Attributes** area.

   The window should appear similar to the following:

Figure 19-3: Signed response



Notice the following features of the response:

- A `Signature` element appears in the WSSE security header. For simplicity, the response in Figure 19-3 doesn't show encryption features configured in the previous walkthrough. If you have left encryption enabled, the `Signature` element will appear alongside the `EncryptedKey` element in the security heading.
- In the figure, the `BinarySecurityToken` element content has been trimmed. In your message, it should be considerably longer.
- The `retrieveQuoteResponse` element has several new attributes. The `wsu:Id` attribute value identifies the element as the target of the `URI` value of the `Reference` element for the signature.

These features enable a receiving application to ensure the validity and integrity of information in the signed element.