CHAPTER 14

Configuring Backend Server Settings

This chapter explains how to identify and configure settings for service providers. It covers these topics:

About Backend Service Providers

In an ACE XML Gateway policy, the configuration settings for a particular service provider are contained in a server object. The server object contains settings that enable the gateway to connect to the backend server. The backend server can be an HTTP server or messaging server. The object can represent an actual server or a logical server made up of several servers (or a server farm).

An echo server is a type of server that, instead of connecting to an external server to send an outgoing request, reflects the request back to the gateway. An echo service is useful for testing a policy or performing message validation out of the traffic stream.

This chapter describes how to configure servers in the ACE XML Manager. A server definition can be defined automatically, through WSDL import, or manually. For messaging servers, the server definition must be added manually.

Working with HTTP Servers

An HTTP server definition holds settings for a particular backend server, such as its network address, the listening port on which it responds to service requests, and whether SSL is required when connecting to it.

To create, edit or remove server objects, you need to be an Administrator user in the console, or a privileged user with the routing role.

Adding an HTTP Server

To define an HTTP server backend service provider:

Step 1 In the web console, set the active subpolicy to the one in which you want to add the server.

Step 2 Click the **HTTP Servers** link from the navigation menu.

Step 3 On the **HTTP Servers** page, click **Add a New HTTP Server**.

Step 4 Type a distinguishing name for the server definition in the **Name** field. This name is used within the console only, and should identify the server object to other users of the web console. It should be unique for HTTP servers in the policy.

Note When generating server names from WSDLs, the ACE XML Manager uses the hostname appended by the port number (server.example.com:80, for example). This is intended only for identification, however, and can be any value that helps you distinguish this server from others.

Step 5 In the **Host** field, type the hostname of the backend server.

The host value must be the DNS name of the backend system (such as `swan.example.com`) or its IP address (such as `192.168.1.100`). You do not need to type the protocol prefix (http://) in the field.

Step 6 In the **Port** field, specify the port number on which the backend server listens for service requests.

If you do not specify this value correctly, the service descriptor cannot pass traffic to the service correctly. Most HTTP servers use port 80 for normal web traffic or port 443 for SSL connections.

Step 7 To configure SSL for the connection to the backend server:

a. Click the **SSL** checkbox. Only choose this option if the server supports SSL connections.

b. In bilateral SSL, the client must authenticate itself to the server, as well as vice versa. If the server requests client authentication (the ACE XML Gateway, in this case), the gateway can present the certificate you specify in the field labelled **If requested, use client public/private keypair**.

If the keypair you want to use is not listed in the menu, click **Upload** to add it as a resource to the policy.

c. Specify how the ACE XML Gateway validates certificates that the backend HTTP server presents from these options:

- To accept any certificate the server presents, leave the **Require remote server certificate signed by this CA certificate** option at its default value, **none**. This setting is the default.
- To accept any certificate authenticated by a specified Certificate Authority (CA), select the **Require remote server certificate signed by this CA certificate** option and choose the CA certificate from the menu. If the certificate does not appear in the menu, choose **Upload** and add the certificate to the ACE XML Manager's list of Trusted Certificate Authorities.
- To specify that the server must present a certificate identical to a specified certificate, click the **Require a certificate from the remote server that is identical to this certificate** button and choose the certificate from the menu. If the certificate does not appear in the menu, choose **Upload** and add the certificate to the ACE XML Manager's list of remote server certificates.

Step 8 With SSL enabled, you can specify what cipher suites the ACE XML Gateway will accept for traffic encryption on connections to this server. In the course of negotiating a connection with the server, the ACE XML Gateway and server must be able to agree on the cipher suite to use.

If the server does not support a cipher suite you specify here, the connection is not permitted. Specify a cipher suite by choosing custom from the SSL Cipher Suite menu and in the field that appears, enter the cipher suite to be accepted in OpenSSL Cipher string format, described here: http://www.openssl.org/docs/apps/ciphers.html

Note Use care when entering the cipher suite string. The ACE XML Manager interface does not verify the value you enter. If you mistype or enter a meaningless value, the ACE XML Gateway may not be able to open an SSL connection with the server.

Step 9 Choose the **Flex Path** option if you want message handling for this server to bypass the Reactor process. Reactor is a high performance, stream-oriented XML engine in the ACE XML Gateway that dramatically speeds message processing. However, not all processing tasks performed by the gateway are supported by the Reactor. If a feature of a virtual service is not supported by the Reactor (such as protocol mediation), the Reactor automatically hands off the message for flex path processing. To ensure compatibility with external systems, you may wish to disable Reactor for a server by choosing this option. In general, it is recommended that you use Reactor in production systems only after careful testing for interoperability with clients and servers.

Step 10 Click **Save Changes** to finish the server configuration.

After creating the server definition, you can configure server pooling, failover settings, or backend request throttling, as described in the following sections.

Pooling Backend Servers

A server pool is a group of backend service providers that collaborate to provide access to a set of services. You can configure server pooling of backend service providers in the ACE XML Gateway, improving the scalability and reliability of the service access exposed through the ACE XML Gateway.

To create a server pool, you configure a primary server for the pool, and then specify the servers that compose the pool by hostname or IP address.

A server pool can operate in one of two modes:

- Failover only, in which a primary server performs all of the service provisioning functions and another server in the pool serves traffic only if the primary fails.
- Failover with load-sharing, in which the requests are distributed to servers in the pool in round-robin fashion.

A server is considered to have failed if a request to it results in an error response or if it fails to respond to a server health check. If a server fails (or for any reason becomes unreachable), the ACE XML Gateway removes the server from the active server pool for a configurable duration, called the back-off period. Once the back-off period ends, the ACE XML Gateway attempts to contact the failed server. If successful, the server is reinstated.

In failover-only mode, one server in the pool is considered the primary. The primary handles all traffic until it fails for some reason or is disabled by the administrator. In this event, another responsive server in the pool becomes the primary, and it handles traffic until a failure event or it is disabled.

Setting Up Server Pooling

To configure server pooling for a service provider:

Step 1 If the HTTP Servers page is not already open, click the **HTTP Servers** link from the navigation menu (among the **Policy** links in the **Message Routing** section of the menu).

Step 2 Click the **view** link next to the server for which you would like to configure server pooling.

By default, the server on which you configure server pooling is the primary server for the cluster, if configuring failover only. The primary server responds to requests under normal conditions.

Step 3 Click the **Edit** link next to the **Server Pooling** heading.

The **Edit Server Pooling** page appears.

Step 4 Select the **Use pooled servers for failover** check box.

Step 5 Optionally, choose **Rotate requests among pooled servers**. This option has the following effect:

- If not selected, the pool operates in failover mode only. The first server in the host list responds to all requests for the configured service, while the other servers in the pool remain inactive unless the primary fails.
- Selecting this option, on the other hand, enables round-robin request allocation among pool members.

Step 6 Specify the number of seconds the ACE XML Gateway should suspend requests to unresponsive pool members in the field next to **Suspend requests to a failed host for at least**.

This value constitutes the back-off period for a failed server. The ACE XML Gateway will not attempt to contact the failed server for the amount of time you specify. Once the time expires, it tries sending a health check message to the server and, if it receives a response, reinstates the server to the pool. Meanwhile, requests for the service are directed to remaining servers in the pool.

Note that this value applies to all servers for which you have enabled health checks, not just the servers in the current pool.

Step 7 In the **Primary Hosts** field, type the IP address or hostname and listening port of each server you want to add to the pool.

Each server address should be on its own line, as follows:

```
primary.example.com:80
pool-member1.example.com:80
pool-member2.example.com:80
192.168.10.1:8080
```

Each line must be unique. The servers do not have to be configured as HTTP servers in the policy to use them in a pool.

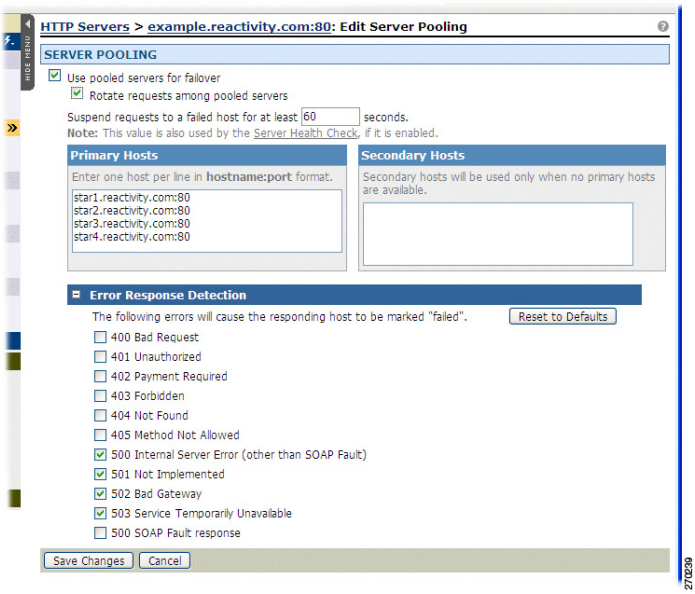
Step 8 Optionally, enter additional addresses for a secondary pool. The ACE XML Gateway only routes messages to the secondary pool if *all* servers in the primary pool fail.

Step 9 To configure how the ACE XML Gateway interprets server failure based on error responses, expand the **Error Response Detection** area of the page and modify the error response selections.

Be sure to select only error codes that actually indicate server failure. That is, if the application hosted by the server returns one of the responses to indicate application-specific status, such as a SOAP Fault response, it should not be selected.

The configuration page should look similar to Figure 14-1.

Figure 14-1 Server pooling configuration



Step 10 Click **Save Changes** when finished.

The HTTP Server information page appears again, but now with the pooled servers listed in the **Server Pooling** area.

Note The status of a newly added server is listed as `undeployed`. The ACE XML Gateway does not address the servers in the pool until you have deployed the policy. After you deploy the policy, the pool list shows the servers as enabled.

Manually Removing Pool Members

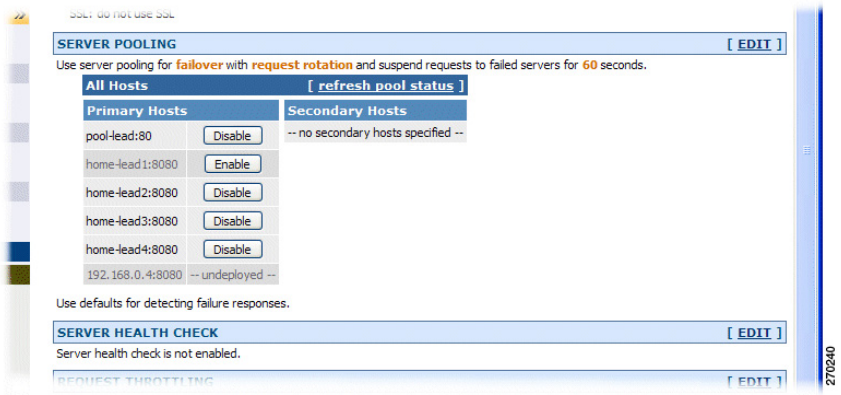
The ACE XML Gateway automatically inactivates server pool members that do not respond to message traffic or health checks for the duration of the back-off period. Additionally, you can withdraw pool members manually, for example, to perform maintenance on the server.

Enabling or disabling servers as described here takes effect immediately. You do not need to deploy the policy to have the change take effect.

You can deactivate or activate pool members as follows:

- Click **Disable** button to deactivate a server member in a pool. The ACE XML Gateway does not direct traffic to disabled pool members, nor does it attempt to send them health check messages.
- A disabled server can only be reinstated to the pool by enabling it in the console. Use the **Enable** button to reactivate the server.

Figure 14-2 Server Pool List



Service Provider Health Checks

A health check is an HTTP message that the ACE XML Gateway sends to a service provider at regular intervals to ensure that it is available. If a server does not respond to the health check, service traffic is not sent to the server for a configurable amount of time (called the back-off period).

Note Service provider health checks are not to be confused with Gateway health checks, which are used by load balancers or routers in front of the ACE XML Gateway to check its health. For information on setting up these types of health checks, see "Configuring a Static Content Response" section on page 15-163.

Health checks configured on a primary server are performed on all members of a pool. If an unavailable server is a member of a server pool, it is taken out of the server pool for the duration of the back-off period. When the back-off period expires, the ACE XML Gateway attempts to contact the server. If it is responsive, the server is reinstated to the pool.

You can configure the message content to send, the resource on the server requested by the health check (by path), and how to interpret the server response. You can also specify the health check interval in seconds. The ACE XML Gateway only sends a health check message if there hasn't been service traffic exchanged with the server for the duration of the health check interval. As a result, the ACE XML Gateway does not check the status of servers at times of high traffic load.

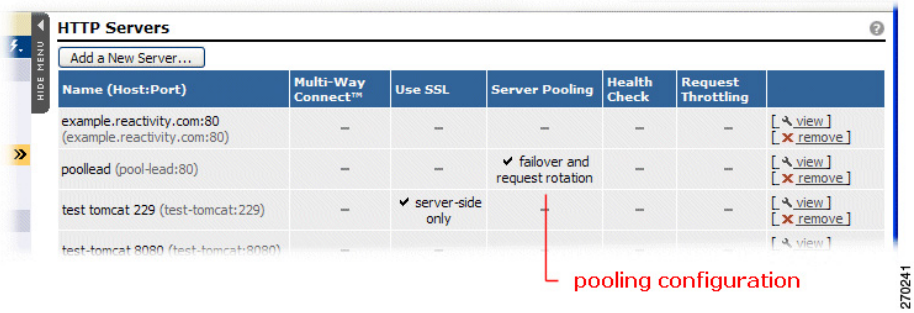
Before starting, determine the resource on the server that you want to call as part of the health check.

To configure server health checks:

Step 1 Open the configuration settings page for the server that contains the pooling configuration by clicking the **view** link next to the server in the **HTTP Servers** page.

The server that contains the pooling configuration is indicated by the **Server Pooling** information field on the servers list.

Figure 14-3 Server Pooling page



Step 2 Click the **Edit** link next to the **Server Health Check** heading.

Step 3 In the **Server Health Check** page, enable server health checking by selecting the first checkbox, labeled **Send the following HTTP request to each host (including any pooled hosts) every**, and enter the health check interval in seconds in the adjacent field.

Note that, in addition to the interval you configure here, the service provider response time can affect the actual time interval between checks. For example, say you have a health check interval of a minute and the typical response time from a server is 5 seconds. If you have 10 servers in a pool, the actual period between health checks will be 1 minute and 50 seconds.

Step 4 Choose whether you want the message to be sent as an HTTP GET or POST method.

Step 5 In the **Request Path** field, type the path to the resource on the server that is to be called as the health check.

Step 6 Specify the response that indicates that the server is available using the controls in the **Response Checking** section.

Specify the time-out period, in seconds, and either a positive test (by choosing **meets** as the response criteria) or negative test (by choosing **does not meet** as the response criteria) against one of these values or attributes of the response:

- an HTTP response code (200, by default)
- the presence of an HTTP header in the response and, optionally, the value of the header field
- a particular value in the body of the response indicated by a regular expression and identified in the response by XPath

Step 7 Optionally, modify the back-off period by typing a new value in the field labelled **If a server fails (does not send a healthy response), suspend requests for at least**.

The ACE XML Gateway does not attempt to contact a server for this amount of time after it has been determined to be unavailable.

Step 8 Click **Save Changes** when finished.

Your configuration changes should be reflected in the **Server Health Check** section of the server information page.

Throttling Backend Server Traffic

A denial-of-service (DoS) attack is a well known type of network attack intended to overwhelm a service provider so that it cannot respond to legitimate traffic. The attack can take several forms, including:

- A message can be too large for the service to handle.
- Messages may be delivered too quickly for the service to respond.
- A message may be designed to use large amounts of processing power. For example, a message may contain XML with many levels of nested entities (entities that reference other entities).

The ACE XML Gateway can prevent such attacks by limiting:

- the size of messages delivered to a service, including the size of attachments
- the rate at which messages can be delivered
- the maximum time the backend service may spend processing a message

When the ACE XML Gateway detects a pattern of traffic that violates one of these limits, it disables delivery to the backend service for an amount of time that allows further deliveries to fall within limits. Message processing resumes when enough time has passed to allow further deliveries without exceeding the rate limit.

The ACE XML Gateway rejects messages that exceed the throttling limits as invalid. It is important to note that the requests are not queued for later processing. The Gateway returns error messages with response code HTTP 500 (by default) to the client. If desired, you can have the ACE XML Gateway send a different response by modifying the Invalid Message error mapping in the exception mapping page.

Configuring Backend Throttling

Backend throttling settings are configurable by backend server. If the resource for which you are configuring values is actually a server pool, the settings apply to traffic directed to the entire pool, not for each individual server.

To configure traffic throttling to a backend server:

Step 1 While logged into the ACE XML Manager as an `Administrator` user or as a `Privileged` user with the `Operations` role, set the active subpolicy to the one that provides the resource to be edited.

Step 2 Click the **HTTP Servers** link in the navigation menu.

Step 3 In the **HTTP Servers** page, click the **view** link next to the server to configure.

Step 4 Click the **Edit** link in the **Request Throttling** section to limit the rate at which the ACE XML Gateway delivers messages to the server.

The **Edit Request Throttling** page appears.

Step 5 To enable request throttling, select the **Throttle the rate of requests** option.

The **Request rate**, **Request burst**, **Average request size**, and **Average latency** fields become enabled.

Step 6 Specify the upper bounds of acceptable server loading by configuring the following fields:

- **Request rate**. The acceptable number of request messages per second the ACE XML Gateway may send to the protected server. The default is 50 requests per second.
- **Request burst**. The acceptable number of request messages the ACE XML Gateway sends to the server at one time. The default is 15 requests at one time.

Note For more information on setting traffic load thresholds, see "Understanding Traffic Rate Thresholds" section on page 23-216.

- **Average request size**. The acceptable average size of requests the ACE XML Gateway sends to the server. The default value is a 1024 kilobyte (KB) average message size. An individual messages larger than this size counts as multiple messages for performance-measurement purposes.
- **Average latency**. The acceptable average response time from the backend service, expressed as a number of seconds. An individual response time that is longer than this value counts as multiple messages for performance-measurement purposes.

Step 7 To specify the maximum acceptable size of an individual message, select the **Never send any request that is larger than** checkbox, and type a whole number to express the maximum acceptable message size in kilobytes (KB) in the corresponding field.

The ACE XML Gateway rejects requests larger than the number of kilobytes specified.

Step 8 To delay subsequent requests when the service returns a code 503 (Server Busy) response, choose **If the server returns code 503 (Server Busy)**.

The **wait at least … seconds** field becomes enabled.

Step 9 To specify the number of seconds by which the ACE XML Gateway delays additional requests after the server returns a code 503 (Server Busy) response, type the number in the **wait at least… seconds** field.

When the ACE XML Gateway receives a 503 error response from the server, it sends no more requests until the specified number of seconds has elapsed.

Step 10 Click **Save Changes** to commit the new settings to the current working policy.

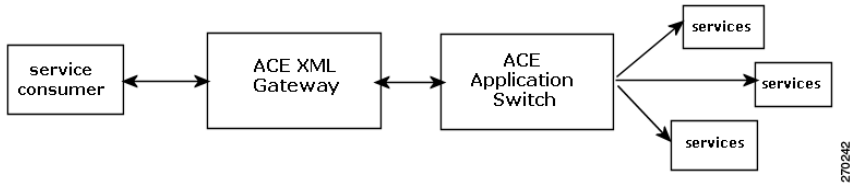
Deploy the policy to have the new throttling settings take effect at the ACE XML Gateway.

Integrating with ACE Application Switches

In production networks, application servers are rarely accessed directly by the client applications. Instead, they are usually proxied by one or more load balancers. The ACE XML Manager includes capabilities for rapidly integrating backend services that are proxied by the Cisco Application Control Engine (ACE) Application Switch. The ACE Application Switch is a high performance application switch that maximizes the availability, performance, and security of applications on the network.

By inspecting the ACE Application Switch configuration, the ACE XML Manager can quickly generate the policy configuration needed to route service traffic to application servers behind the ACE Application Switch.

Figure 14-4 Downstream ACE Appliance configuration



In ACE configuration terms, the ACE Application Switch exposes a backend server farm to the external network through a *virtual IP* (VIP). The ACE XML Manager issues a query an ACE Application Switch virtual device context to discover the VIPs it presents.

After performing VIP discovery, the ACE XML Manager presents the VIPs it found for the Cisco ACE virtual device. You can choose the VIPs for which you want to generate HTTP Server objects, which can then be used as the backend service destination for a virtual service.

The ACE XML Manager works with the following versions of the ACE Application Switch:

- Cisco ACE appliance, software versions A1(7) and higher
- Cisco ACE module, software versions A1(0), A2(0), and higher

Integrating the ACE Application Switch and ACE XML Gateway configuration occurs in two phases:

1. Set up the connection to the ACE Application Switch virtual device from which you want to import VIPs.
2. At any time after configuring the ACE virtual device connection, direct the ACE XML Manager to inspect the Cisco ACE virtual device and generate server objects from selected VIPs.

Every time you have the ACE XML Manager performs VIP discovery on a particular ACE virtual device, the ACE XML Manager treats the discovery as a new event; that is, it will find and report VIPs for which you have already configured HTTP server objects. However, the ACE XML Manager prevents you from generating duplicate server objects from VIPs that were previously imported. It does not attempt to perform any type of configuration update for VIPs that have already been imported.

Note SSL connection requirements for backend servers as configured in the ACE Application Switch are not propagated to the HTTP server object generated by VIP import in the ACE XML Gateway policy. After importing a VIP for a backend SSL server, you will need to enable SSL connection encryption in the object settings manually.

The server management features provided by the ACE XML Gateway differ between manually created HTTP server objects and those generated by VIP import. While request throttling is configurable for ACE-based server objects, server pooling is not (since ACE Application Switches are generally relied upon to perform the task of load balancing backend servers).

Setting Up the Cisco ACE Virtual Device Connection

To import VIPs from the ACE Application Switch, first specify the connection to the virtual device on the ACE Application Switch. Keep in mind that this connection is used only for policy development on the ACE XML Manager. Service traffic will be sent to the connections that are generated from the VIPs found on the Cisco ACE virtual device. Also note that the ACE XML Manager does not modify the configuration of the ACE device specified by this connection. That is, the ACE XML Manager performs read-only operations on the Cisco ACE device only.

To configure a connection to a Cisco ACE virtual device:

Step 1 In the ACE XML Manager web console, click the **HTTP Servers** link in the navigation menu.

Step 2 Click the **Configure Integrated ACE Management** button.

Step 3 Click the **Add ACE Application Switch** button.

Step 4 Enter values in the following fields:

- **ACE VLAN Address**—The IP address for the ACE Application Switch virtual device that has the VIPs you want to import. This address should correspond to a particular VLAN address in the ACE configuration.
- **HTTP(S) Port**—The port number on which this Cisco ACE virtual device listens for HTTP requests.
- **Use HTTPS**—Whether the ACE XML Gateway should use SSL when connecting to the ACE Application Switch to inspect and import VIPs. Note that the ACE XML Gateway can use secure socket layer security to access the ACE device, but it cannot verify the certificate on the ACE Application Switch (since the certificate cannot be imported into the ACE XML Gateway policy).

  If accessing an ACE appliance using SSL, you usually need to connect to port 10443. (Port 443 is reserved for other uses on the ACE appliance.)
- **Username**—The username for an administrative account for this ACE Application Switch virtual device.
- **Password**—The password for the specified user account.
- **Connection Timeout**—The amount of time after which the Manager should abandon its attempt to validate the connection and discover VIPs from the Cisco ACE application switch. Note that this value applies to each individual request within the inspection event, not to the overall time it takes for the inspection.

Step 5 Click **Save Changes**.

The ACE XML Manager validates the connection to the Cisco ACE virtual device, and performs an initial inspection of its configuration. If successful, the newly defined connection appears in the ACE table.

You can now import VIPs from the Cisco ACE virtual device by clicking the Import VIPs link for the connection and following the instructions in the following section.

Generating Server Definitions from VIPs

After configuring the ACE Application Switch virtual device connection, you can import VIP information at any time. In this process, the Manager inspects the ACE virtual device for the VIPs it exposes and presents them in the web console. You can choose the VIPs for which you want the Manager to generate HTTP server objects, as described in the following procedures:

Step 1 If it is not already open, access the HTTP Servers page by clicking the **HTTP Servers** link in the navigation menu.

The Cisco ACE device connections that have been configured appear in the ACE portion of the HTTP Servers list. For information on configuring connections, see "Setting Up the Cisco ACE Virtual Device Connection" section on page 14-156.

Step 2 Click the **import VIPs** link next to the ACE virtual device from which you would like to import VIPs.

The ACE XML Manager takes a moment to inspect the ACE Application Switch. When it finishes, a list of IP addresses and port numbers appear for the VIPs presented in the ACE virtual device.

Note The ACE XML Gateway does not support import of any VIP that matches a range of IP addresses in the ACE Application Switch policy.

Step 3 Choose the VIPs for which you want to create server objects by selecting their check boxes.

If a port number on the VIP is a range (as it is for the any port selection in the ACE policy), enter the specific port number on which you want requests to the backend ACE application switch to be sent in the text field next to the VIP.

Note If you want messages to be sent to one of several ports on the backend VIP, you will need to create a server object for each different port. You will need to repeat the VIP import process to create backend server objects for the additional ports.

Step 4 After choosing the VIPs for which you would like to create HTTP server objects, click the **Create HTTP Servers** button.

The VIPs appear in the HTTP Servers table. You can now use the VIPs in the backend service configuration for virtual services in the policy.

Using an HTTP Echo Server

If a virtual service uses an echo server as its backend service, the ACE XML Gateway produces the response instead of an external server. The response can be a fixed response (such as a HTML page or SOAP response configured in the policy) or the original request reflected back to the client.

An echo server is useful in several ways:

- It can be used if the backend system is under development and not ready to receive requests. The echo service lets you test client-side applications before the backend service is ready.
- The ACE XML Gateway can validate messages outside the main traffic stream.

Echo servers work only with virtual services that are HTTP-based. It does not work for messaging servers, such as MQ Series.

To set up an echo server, follow these steps:

Step 1 Click the **HTTP Servers** link in the navigation menu.

Step 2 Click the **Add a New HTTP Echo Server** button.

Step 3 In the **New Server** page, provide a distinguishing name for the echo server definition in the **Name** field. The name should be unique for echo server definitions in the policy.

Step 4 Choose how the response will be generated from these options:

- **Echo** reflects the request back to the client. Any processing or validation settings specified in the policy are applied to the message as it traverses the gateway. The echo server itself does not process the message except to adjust HTTP headers as needed; that is, it removes request-specific HTTP headers and adds response-specific HTTP headers before reflecting the message back to the gateway for response processing.
- **Fixed** returns the response you configure with the **Status Code**, **Content-Type**, **Other Headers**, and **Body** fields. (In addition to the headers configured here, the echo server adds the `Content-length` HTTP header to the returned message, with the appropriate value given the message size.)
- **Asynchronous** always returns an HTTP response with a return code of 202. This option is functionally equivalent to choosing **Fixed** response and setting the **Status Code** to 202 Accepted.

Note For more information on these options, see the discussion following these steps.

Step 5 Click **Save Changes** to save your changes to the working policy.

Step 6 In the **Backend Service** settings for a service definition for which you want messages to be echoed, choose the echo server definition as the host server. For new objects, the echo server is available in the list of servers available for the service interface.

To use the ACE XML Gateway as a processing engine outside the traffic stream, note these points:

- If you want the ACE XML Gateway to transform the message (for example, by XSLT or content replacement), use the echo mode. If the transformation fails, a HTTP 500 response is returned instead of the transformed message.
- For read-only operations, including various forms of validation (such as XSD, SOAP attachment checking, content screening) you can use either an echo response or fixed response. A fixed response may be preferable in most cases, since it will likely be more bandwidth efficient and easily parsed by the receiving application.

In either case, a status code of 200 of a response indicates to the calling application that message passed validation. Any other error code can be interpreted as validation failure.

Working with Messaging Servers

The ACE XML Gateway can route message traffic from TIBCO TIB/RV or Message Queue (MQ) Series servers. Java Messaging Service (JMS) servers are also supported through the use of an optional, SDK-based extension module.

The **Messaging Servers** page lists the messaging server profiles currently defined in the ACE XML Gateway policy. In the page you can create messaging server definitions, and edit or remove existing server definitions.

For more information on messaging servers, see Chapter 16, “Working With JMS Traffic,” and Chapter 16, “Working With JMS Traffic.”

Removing a Server Definition

You can remove server definitions by clicking the **HTTP Servers** or **Messaging Servers** link in the console navigation menu. In the servers list, remove a server by clicking the **remove** link next to the entry for the server. Confirm the operation when prompted.

You can remove a server only if there are no service definitions that use the server. In order to delete the server in these cases, either delete the object that depends on the server or change it to another server.