CHAPTER 8

Configuring the XML Interface

This chapter describes how to use Extensible Markup Language (XML) to remotely configure a Cisco Application Control Engine (ACE) module from a network management station (NMS). You can transmit, exchange, and interpret data among the applications.

This chapter contains the following major sections:

Information About XML

Web services provide network-based software applications that use XML to transmit, exchange, and interpret data among applications that would otherwise have difficulty interoperating together.

XML provides an application-independent way of sharing data between computer systems. Similar to HTML, XML consists of text delimited by tags so it is easily conveyed over the Internet. In XML, the tags define the meaning and structure of the information, enabling computer applications to use the information directly. Unlike HTML, XML tags identify the data, rather than specifying how to display it. An XML tag acts like a field name in your program; it puts a label on a piece of data that identifies it (for example: <message>...</message>).

An XML document that contains configuration commands and output results is easily transformed between the devices by using standard Internet protocols. A network management station (NMS), such as the CiscoWorks Hosting Solution Engine (HSE), can connect to the ACE and push new configurations to it over HTTP or secure HTTP (HTTPS). Any command that you can configure from the ACE CLI can be configured remotely from a NMS by exchanging XML documents over HTTP or HTTPS.

The XML application programming interface (API) allows you to automate the programmatic configuration of the ACE by using a Document Type Definition (DTD). The XML format is a translation of the CLI commands into an equivalent XML syntax. Each ACE CLI command has an equivalent XML tag, and all of the parameters of the CLI command are attributes of that element. The ACE uses an Apache HTTP server to provide the XML management interface and to provide HTTP services between the ACE and the management client. To use the ACE XML API, you must have the Admin user role.

You can use XML to do the following:

- Provide a mechanism using XML to transfer, configure, and monitor objects in the ACE. This XML capability allows you to easily shape or extend the CLI query and reply data in XML format to meet different specific business needs.
- Transfer **show** command output from the ACE CLI interface in an XML format for statistics and status monitoring. This capability allows you to query and extract data from the ACE.
- Use the ACE XML DTD schema for formatting CLI queries or parsing the XML results from the ACE to enable third-party software development through XML communications.
- Provide remote user authentication through AAA.
- Provide session and context management by the global administrator and other privileged users that have the Admin user role.

This section contains the following topics:

HTTP and HTTPS Support with the ACE

The ACE and an NMS can easily send and receive an XML document containing configuration commands or output results by using standard Internet protocols, such as HTTP or secure HTTP (HTTPS), as the transfer protocol. HTTPS uses Secure Sockets Layer (SSL) to provide encrypted communication between the management client and the ACE.

The administrator of the system designates a website as the entry point to the API, and all requests and queries are made through those URLs. This website also provides the DTDs that define the XML for requests, queries, and responses.

The XML input is submitted through the data portion of an HTTP POST request. A field named “xml” contains the XML string that defines the request or query. The response to this HTTP POST represents a pure XML response with either a success or failure indicator for a request or the response to a query.

When you use XML to transfer configuration data and results, the NMS connects to the ACE and sends a new configuration in an XML document to the ACE over HTTP or HTTPS. The ACE then applies the new configuration.

The following example shows the HTTP conversation between the client and the server, as related to the XML implementation on the ACE:

```
******** Client **************
POST /bin/xml_agent HTTP/1.1
Authorization: Basic VTpQ
Content-Length: 95
xml_cmd=<request_xml>
<interface type=”vlan” number=”80”>
<access-group access-type=”input” name=”acl1”/>
<ip_address address=”60.0.0.145” netmask=”255.255.255.0”/>
<shutdown sense=”no"/>
</interface>
<show_running-config/>
</request_xml>

******** Server **************
HTTP/1.1 200 OK
Content-Length: 21
```

```
<response_xml>
<config_command>
<command>
interface vlan 80
ip address 60.0.0.145 255.255.255.0
access-group input acl1
no shutdown
</command>
<status code="100" text="XML_CMD_SUCCESS"/>
</config_command>
</response_xml>

******** Client **************
POST /bin/xml_agent HTTP/1.1
Content-Length: 95
xml_cmd=<request_xml>
<show_running-config/>
</request_xml>

******** Server **************
HTTP/1.1 401 Unauthorized
Connection: close
WWW-Authenticate: Basic realm=/xml-config
```

HTTP Return Codes

HTTP return codes indicate the status of the request and reports errors between the server and the client. The Apache HTTP server return status codes follow the standards outlined in RFC 2616. Table 8-1 lists the supported HTTP return codes.

Table 8-1 Supported HTTP Return Codes for XML

Return Code	Description
200	OK
201	Created
202	Accepted
203	Non-Authoritative Information
206	Partial Content
301	Moved Permanently
302	Found
400	Bad Request
401	Unauthorized (credentials required, but not provided)
403	Forbidden (illegal credentials submitted; syslog also generated)
404	Not Found (“/xml-config” not specified)
405	Method Not Allowed
406	Not Acceptable
408	Request Time-out (more than 30 seconds has passed waiting on receive)
411	Missing Content-Length (missing or zero Content-Length field)
500	Internal Server Error

Table 8-1 Supported HTTP Return Codes for XML (continued)

Return Code	Description
501	Not Implemented (“POST” not specified)
505	HTTP Version Not Supported (“1.0” or “1.1” not specified)

The following HTTP headers are supported:

- Content-Length (nonzero value required for all POSTs)
- Connection (*close* value indicates that a request should not be persistent)
- WWW-Authenticate (sent to the client when credentials are required and missing)
- Authorization (sent from the client to specify basic credentials in base 64 encoding)

For example, when an XML error occurs, the HTTP response contains a 200 return code. The portion of the original XML document with the error is returned with an error element that contains the error type and description.

The following is a typical example of an XML error response:

```
<response_xml>
<config_command>
<command>
interface vlan 20
  no shut
  description xyz
  exit
</command>
<status code = ‘200’ text=’XML_CMD_FAILURE’>
<error_command> description xyz </error_command>
<error_message> unrecognized element - description </error_message>
</status>
</config_command>
</response_xml>
```

The returned error codes correspond to the attributes of the configuration element. The possible returned XML error can include any of the following:

```
XML_ERR_WELLFORMEDNESS   /* not a well formed xml document */
XML_ERR_ATTR_INVALID     /* found invalid value attribute */
XML_ERR_ELEM_INVALID     /* found invalid value unrecognized */
XML_ERR_CDL_NOT_FOUN     /* parser cdl file not found */
XML_ERR_INTERNAL         /* internal memory or coding error */
XML_ERR_COMM_FAILURE     /* communication failure */
XML_ERR_VSH_PARSER       /* vsh parse error on the given command */
XML_ERR_VSH_CONF_APPLY   /* vsh unable to apply the configuration */
```

Document Type Definition

A DTD is the basis for XML configuration documents that you create using the ACE. The purpose of a DTD is to define the legal building blocks of an XML document by defining the document structure with a list of legal elements.

DTD designates an XML list that specifies precisely which elements can appear in a request, query, or response document. It also specifies the contents and attributes of the elements. A DTD can be declared inline in your XML document or as an external reference.

The ACE DTD file, cisco_ace.dtd, is included as part of the software image and is accessible from a web browser using either HTTP or HTTPS. See the "Accessing the ACE DTD File" section for details. You can use a web browser to either directly access the cisco_ace.dtd file or open the cisco_ace.dtd file from the Cisco ACE Module Management page.

The following example shows the sequence of ACE CLI commands for creating a real server followed by the associated DTD XML rserver elements for the commands:

[**no**] **rserver** [**host** | **redirect**] *name*

[**no**] **conn-limit max** *maxconns* [**min** *minconns*]

[**no**] **description** *string*

[**no**] **inservice**

[**no**] **ip address** {*ip_address*}

[**no**] **probe** *name*

[**no**] **weight** *number*

```
**********************************************************************
  Elements, Attributes and Entities required for rserver
  **********************************************************************
 -->

<!--
probe-name is a string of length 1 to 32.
-->
<!ELEMENT probe_rserver EMPTY>
<!ATTLIST probe_rserver
  sense     CDATA     #FIXED    "no"
  probe-name      CDATA      #REQUIRED
>

<!--
relocation-str length is 1 to 127
-->
<!ELEMENT webhost-redirection EMPTY>
<!ATTLIST webhost-redirection
  sense      (yes | no)      #IMPLIED
  relocation-string      CDATA      #REQUIRED
  redirection-code      (301 | 302)      #IMPLIED
>

<!--
type is optional for host.
ip, probe and weight are valid only when type = host.
address-type is valid only when type=host.
name length is 1 to 32.
webhost-redirection is valid only if type=redirect.
-->
<!ELEMENT rserver (description, ip_address, conn-limit, probe_rserver,
                                        weight, inservice, webhost-redirection)*>
<!ATTLIST rserver
  sense     CDATA     #FIXED    "no"
  type      (redirect | host)      #IMPLIED
  name      CDATA      #REQUIRED
>
```

Guidelines and Limitations

To use the ACE XML interface, you must have the Admin user role.

The ACE creates two default user accounts at startup: admin and www. The admin user is the global administrator and cannot be deleted. The ACE uses the www user account for the XML interface and www cannot be deleted.

Caution When you upgrade your ACE software to version A2(1.1) or higher, you must change the default www user password if you have not already done so. Otherwise, after you upgrade the ACE software, the www user will be disabled and you will not be able to use XML to remotely configure an ACE until you change the default www user password. See Chapter 2, Configuring Virtualization, in the *Cisco Application Control Engine Module Virtualization Configuration Guide* for details on changing a user account password. In this case, the user would be www.

Default Settings

XML responses automatically appear in XML format if the corresponding CLI **show** command output supports the XML format. However, if you are running commands on the CLI console or you are running raw XML responses from NMS, the XML responses appear in regular CLI display format. See the "Enabling the Display of Raw XML Request show Command Output in XML Format" section for details. For details on the **show** command output supported in XML format, consult the cisco_ace.dtd file.

Configuring the XML Interface

This section describes how to configure the XML interface and contains the following topics:

Task Flow for Configuring XML

Follow these steps to configure XML usage with the ACE:

Step 1 If you are operating in multiple contexts, observe the CLI prompt to verify that you are operating in the desired context. If necessary, log directly in to, or change to, the correct context.

```
host1/Admin# changeto C1
host1/C1#
```

The rest of the examples in this table use the Admin context, unless otherwise specified. For details on creating contexts, see the *Cisco Application Control Engine Module Virtualization Configuration Guide*.

Step 2 Enter configuration mode.

```
host1/Admin# config
Enter configuration commands, one per line. End with CNTL/Z.
host1/Admin(config)#
```

Step 3 Create a Layer 3 and Layer 4 class map to classify the HTTP or HTTPS management traffic that can be received by the ACE.

```
host1/Admin(config)# class-map type management match-all HTTPS-ALLOW_CLASS
host1/Admin(config-cmap-mgmt)# match protocol https source-address 192.168.1.1
255.255.255.255
host1/Admin(config-cmap-mgmt)# exit
```

Step 4 Configure a Layer 3 and Layer 4 HTTP or HTTPS traffic management policy.

```
host1/Admin(config) # policy-map type management first-match MGMT_HTTPS_POLICY
host1/Admin(config-pmap-mgmt) # class HTTPS-ALLOW_CLASS
host1/Admin(config-pmap-mgmt-c) # permit
host1/Admin(config-pmap-mgmt-c) # exit
```

Step 5 Attach the traffic policy to a single interface or globally on all VLAN interfaces associated with a context, and specify the direction in which the policy should be applied. For example, to specify an interface VLAN and apply multiple service policies to the VLAN, enter:

```
host1/Admin(config)# interface vlan50
host1/Admin(config-if)# ip address 192.168.10.1 255.255.0.0
host1/Admin(config-if)# service-policy input MGMT_HTTPS_POLICY
host1/Admin(config-if)# exit
host1/Admin(config)# exit
```

Step 6 (Optional) Enable the display of raw XML request **show** command output in XML format.

Note True XML responses always automatically appear in XML format.

```
host1/Admin# xml-show on
```

Step 7 (Optional) Save your configuration changes to Flash memory.

```
host1/Admin# copy running-config startup-config
```

Configuring HTTP and HTTPS Management Traffic Services

This section describes how to configure HTTP and HTTPS remote management traffic to the ACE through class maps, policy maps, and service policies. The ACE provides support for remote management using XML over either HTTP or HTTPS to configure, monitor, and manage software objects.

The following items summarize the role of each function in configuring HTTP or HTTPS network management access to the ACE:

- Class map—Provides the remote network traffic match criteria to permit HTTP and HTTPS management traffic based on HTTP or HTTPS network management protocols or host source IP addresses.
- Policy map—Enables remote network management access for a traffic classification that matches the criteria listed the class map.
- Service policy—Activates the policy map and attaches the traffic policy to an interface or globally on all interfaces.

HTTP or HTTPS sessions are established to the ACE per context. For details on creating contexts and users, see the *Cisco Application Control Engine Module Virtualization Configuration Guide*.

This section contains the following topics:

Creating and Configuring a Class Map

This section describes how to create a Layer 3 and Layer 4 class map to classify the HTTP or HTTPS management traffic that can be received by the ACE. This process allows network management traffic by identifying the incoming IP protocols that the ACE can receive and the client source host IP address and subnet mask as the matching criteria.

A class map of type management defines the allowed network traffic as a form of management security for protocols such as HTTP or HTTPS. A class map can include multiple **match** commands. You can configure class maps to define multiple HTTP or HTTPS management protocol or source IP address **match** commands in a group that you then associate with a traffic policy. The **match-all** and **match-any** keywords determine how the ACE evaluates multiple match statements operations when multiple match criteria exist in a class map.

Detailed Steps

	Command	Purpose
Step 1	**config** **Example:** host1/Admin# config host1/Admin#(config)#	Enters global configuration mode.
Step 2	**class-map type management** [**match-all** | **match-any**] *map_name* **Example:** host1/Admin(config)# class-map type management match-all HTTPS-ALLOW_CLASS host1/Admin(config-cmap-mgmt)#	Creates a Layer 3 and Layer 4 class map to classify the HTTP or HTTPS management traffic that can be received by the ACE. The keyword options and argument are as follows: • **match-all | match-any**—(Optional) Determines how the ACE evaluates Layer 3 and Layer 4 network traffic when multiple match criteria exist in a class map. The class map is considered a match if the **match** commands meet one of the following conditions: – **match-all**—(Default) All of the match criteria listed in the class map match the network traffic class in the class map. – **match-any**—Only one of the match criteria listed in the class map matches the network traffic class in the class map. • *map_name*—Name assigned to the class map. Enter an unquoted text string with no spaces and a maximum of 64 alphanumeric characters. The class name is used for both the class map and to configure a policy for the class in the policy map. This command enters the class map management configuration mode.
	no class-map type management [**match-all** | **match-any**] *map_name* **Example:** host1/Admin(config)# no class-map type management match-all HTTPS-ALLOW_CLASS	(Optional) Removes a Layer 3 and Layer 4 network management class map from the ACE.
Step 3	**description** *text* **Example:** host1/Admin(config-cmap-mgmt)# description Allow HTTPS access to the ACE	Provides a brief summary about the Layer 3 and Layer 4 remote management class map. The *text* argument is the description that you want to provide. Enter an unquoted text string with a maximum of 240 alphanumeric characters.
	no description **Example:** host1/Admin(config-cmap-mgmt)# no description	(Optional) Remove the description from the class map.

	Command	Purpose
Step 4	[*line_number*] **match protocol** {**http** | **https**} {**any** | **source-address** *ip_address mask*} **Example:** host1/Admin(config-cmap-mgmt)# match protocol https source-address 192.168.10.1 255.255.0.0	Configures the class map to specify that the HTTP or HTTPS remote network management protocol can be received by the ACE. You configure the associated policy map to permit access to ACE for the specified management protocol. For XML support, a class map of **type management** allows IP protocols such as HTTP and HTTPS. As part of the network management access traffic classification, you also specify either a client source host IP address and subnet mask as the matching criteria or instruct the ACE to allow any client source address for the management traffic classification. You can include multiple **match protocol** commands in a class map. The keywords, arguments, and options are as follows: • *line_number*—(Optional) Line number that allows you to edit or delete individual **match** commands. Enter an integer from 2 to 255 as the line number. For example, you can enter **no** *line_number* to delete long **match** commands instead of entering the entire line. • **http**—Configures management access between the ACE HTTP server and the management client over HTTP. • **https**—Configures management access between the ACE HTTP server and the management client over secure HTTP. • **any**—Specifies any client source address for the management traffic classification. • **source-address—**Specifies a client source host IP address and subnet mask as the network traffic matching criteria. As part of the classification, the ACE implicitly obtains the destination IP address from the interface on which you apply the policy map. • *ip_address*—Source IP address of the client. • *mask*—Subnet mask of the client in dotted-decimal notation (for example, 255.255.255.0).
	no match protocol {**http** | **https**} {**any** | **source-address** *ip_address mask*} **Example:** host1/Admin(config-cmap-mgmt)# no match protocol https source-address 192.168.10.1 255.255.0.0	(Optional) Deselects the specified network management protocol match criteria from the class map.
Step 5	**do copy running-config startup-config** **Example:** host1/Admin(config-cmap-mgmt)# do copy running-config startup-config	(Optional) Copies the running configuration to the startup configuration.

Creating a Layer 3 and Layer 4 Policy Map

This section describes how to create a Layer 3 and Layer 4 policy map, associate a class map with the policy map, and specify the policy map actions. A Layer 3 and Layer 4 policy map defines the actions executed on HTTP or HTTPS management traffic that matches the specified classifications.

Detailed Steps

	Command	Purpose

Step 1	**config**
	
	Example:
	`host1/Admin# config`
	`host1/Admin#(config)#`
		Enters global configuration mode.

Step 2	**policy-map type management first-match** *map_name*
	
	Example:
	`host1/Admin(config)# policy-map type management first-match MGMT_HTTPS_POLICY`
	`host1/Admin(config-pmap-mgmt)#`
		Configures a Layer 3 and Layer 4 policy map that permits the management traffic received by the ACE. The ACE executes the action for the first matching classification. The ACE does not execute any additional actions.
		
		The *map_name* argument specifies the name assigned to the Layer 3 and Layer 4 network management policy map. Enter an unquoted text string with no spaces and a maximum of 64 alphanumeric characters.
		
		This command enters the policy map management configuration mode.

	no policy-map type management first-match *map_name* **Example:** `host1/Admin(config)# no policy-map type management first-match MGMT_HTTPS_POLICY`	(Optional) Removes a network traffic management policy map from the ACE.

	Command	Purpose
Step 3	**class** {*name1* [**insert-before** *name2*] | **class-default**} **Example:** host1/Admin(config-pmap-mgmt)# class HTTPS-ALLOW_CLASS host1/Admin(config-pmap-mgmt-c)#	Associates the HTTP or HTTPS management traffic class map with the traffic policy. The arguments, keywords, and options are as follows: • *name1*—Name of a previously defined Layer 3 and Layer 4 traffic class, configured with the **class-map** command, to associate traffic to the traffic policy. Enter an unquoted text string with no spaces and a maximum of 64 alphanumeric characters. • **insert-before** *name2*—(Optional) Places the current class map ahead of an existing class map or inline match condition specified by the *name2* argument in the policy map configuration. The ACE does not save the sequence reordering as part of the configuration. Enter an unquoted text string with no spaces and a maximum of 64 alphanumeric characters. • **class-default**—Specifies the class-default class map for the Layer 3 and Layer 4 traffic policy. This class map is a reserved class map created by the ACE. You cannot delete or modify this class. All network traffic that fails to meet the other matching criteria in the named class map belongs to the default traffic class. If none of the specified classifications match, the ACE then matches the action specified under the **class class-default** command. The class-default class map has an implicit **match any** statement in it and is used to match any traffic classification. This command enters the policy map management class configuration mode.
	no class *name1* **Example:** host1/Admin(config-cmap-mgmt)# class HTTPS-ALLOW_CLASS	(Optional) Removes a class map from a Layer 3 and Layer 4 policy map.
Step 4	**permit** **Example:** host1/Admin(config-pmap-mgmt-c)# permit	Allows the HTTP or HTTPS management traffic listed in the Layer 3 and Layer 4 class map to be received by the ACE.
	no permit **Example:** host1/Admin(config-pmap-mgmt-c)# no permit	(Optional) Disallows the HTTP or HTTPS management traffic listed in the Layer 3 and Layer 4 class map to be received by the ACE.
	deny **Example:** host1/Admin(config-pmap-mgmt-c)# deny	Denies the HTTP or HTTPS management traffic listed in the Layer 3 and Layer 4 class map to be received by the ACE.

	Command	Purpose
	no deny **Example:** host1/Admin(config-pmap-mgmt-c)# no deny	Allows the HTTP or HTTPS management traffic listed in the Layer 3 and Layer 4 class map to be received by the ACE.
Step 5	**do copy running-config startup-config** **Example:** host1/Admin(config-pmap-mgmt-c)# do copy running-config startup-config	(Optional) Copies the running configuration to the startup configuration.

Examples

The following example shows how to use the **insert-before** command to define the sequential order of two class maps in the policy map:

```
host1/Admin(config-pmap-mgmt)# class HTTPS-ALLOW_CLASS insert-before
L4_REMOTE_ACCESS_CLASS
```

The following example shows how to specify the class-default class map for the Layer 3 and Layer 4 traffic policy:

```
host1/Admin(config-pmap-mgmt)# class class-default
host1/Admin(config-pmap-mgmt-c)#
```

Applying a Service Policy Globally to All VLAN Interfaces in the Same Context

This section describes how to apply an existing policy map globally to all VLAN interfaces in the same context.

Note the following guidelines when applying a service policy:

- Policy maps, applied globally in a context, are internally applied on all interfaces existing in the context.
- A policy activated on an interface overwrites any specified global policies for overlapping classification and actions.

Note To apply the policy map to a specific VLAN interface only, see the "Applying a Service Policy to a Specific VLAN Interface" section.

Restrictions

The ACE allows only one policy of a specific feature type to be activated on an interface.

Detailed Steps

	Command	Purpose
Step 1	**config** **Example:** host1/Admin# config host1/Admin#(config)#	Enters global configuration mode.
Step 2	**service-policy input** *policy_name* **Example:** host1/Admin(config)# service-policy input MGMT_HTTPS_POLICY	Globally applies the management policy map to all of the VLANs associated with a context. The keywords and arguments are as follows: • **input**—Specifies that the traffic policy is to be attached to the input direction of an interface. The traffic policy evaluates all traffic received by that interface. • *policy_name*—Name of a previously defined policy map, configured with a previously created **policy-map** command. The name can be a maximum of 40 alphanumeric characters.
	no service-policy input *policy_name* **Example:** host1/Admin(config)# no service-policy input MGMT_HTTPS_POLICY	(Optional) Removes the management policy map from all of the VLANs associated with a context. When you remove a policy, the ACE automatically resets the associated service policy statistics to provide a new starting point for the service policy statistics the next time that you attach a traffic policy to a specific VLAN interface or globally to all VLAN interfaces in the same context.
Step 3	**do copy running-config startup-config** **Example:** host1/Admin(config)# do copy running-config startup-config	(Optional) Copies the running configuration to the startup configuration.

Applying a Service Policy to a Specific VLAN Interface

This section describes how to apply an existing policy map to a specific VLN interface. A policy activated on an interface overwrites any specified global policies for overlapping classification and actions.

Note To apply the policy map globally to all VLAN interfaces in the same context, see the "Applying a Service Policy Globally to All VLAN Interfaces in the Same Context" section.

Restrictions

The ACE allows only one policy of a specific feature type to be activated on an interface.

Detailed Steps

	Command	Purpose
Step 1	**config** **Example:** host1/Admin# config host1/Admin#(config)#	Enters global configuration mode.
Step 2	**interface vlan** *number* **Example:** host1/Admin(config)# interface vlan 50 host1/Admin(config-if)#	Specifies an interface VLAN. The *number* argument is the number for a VLAN assigned to the ACE This commands enters the interface configuration mode commands for the VLAN.
Step 3	**ip address** *address* **Example:** host1/Admin(config-if)# ip address 192.168.10.1 255.255.0.0	Specifies the VLAN IP address.
Step 4	**service-policy input** *policy_name* **Example:** host1/Admin(config-if)# service-policy input MGMT_HTTPS_POLICY	Applies the management policy map to the VLAN. The keywords and arguments are as follows: • **input**—Specifies that the traffic policy is to be attached to the input direction of an interface. The traffic policy evaluates all traffic received by that interface. • *policy_name*—Name of a previously defined policy map, configured with a previously created **policy-map** command. The name can be a maximum of 40 alphanumeric characters.
	no service-policy input *policy_name* **Example:** host1/Admin(config-if)# no service-policy input MGMT_HTTPS_POLICY	(Optional) Removes the management policy from an interface VLAN. When you remove a policy, the ACE automatically resets the associated service policy statistics to provide a new starting point for the service policy statistics the next time that you attach a traffic policy to a specific VLAN interface or globally to all VLAN interfaces in the same context.
Step 5	**do copy running-config startup-config** **Example:** host1/Admin(config-if)# do copy running-config startup-config	(Optional) Copies the running configuration to the startup configuration.

Enabling the Display of Raw XML Request show Command Output in XML Format

This section describes how to enable the display of raw XML request **show** command output in XML format. By default, XML responses will automatically appear in XML format if the corresponding CLI **show** command output supports the XML format. However, if you are running commands on the CLI console or you are running raw XML responses from NMS, the XML responses appear in regular CLI display format.

You can enable the display of raw XML request **show** command output in XML format by performing one of the following actions:

- Specifying the **xml-show on** command in Exec mode from the CLI.
- Including the **xml-show on** command in the raw XML request itself (CLI commands included in an XML wrapper).

Selection of the **xml-show on** command is not required if you are running true XML (as shown in the example below).

For details on the **show** command output supported in XML format, consult the ACE DTD file, cisco_ace.dtd, that is included as part of the software image (see the "Accessing the ACE DTD File" section). The ACE DTD file contains the information on the XML attributes for those **show** commands that have output that supports the XML format.

For example, if you specify the **show interface vlan 10** command, the DTD for the **show interface** command appears as follows:

```
<!--
interface-number is req for show-type vlan | bvi.
interface-number is between 1 and 4095 for vlan and 8191 for bvi.
-->
<!ENTITY % show-interface
    "interface-type      (vlan | bvi | eobc)      #IMPLIED
      interface-number      CDATA      #IMPLIED"
>
```

The XML representation of the **show interface** command appears as follows:

```
<show_interface interface-type='vlan' interface-number='10'/>
```

The following example illustrates the XML representation of the **show interface** command output:

```
<response_xml>
<exec_command>
<command>
show interface vlan 10
</command>
<status code="100" text="XML_CMD_SUCCESS"/>
<xml_show_result>
<xml_show_interface>
<xml_interface_entry>
<xml_interface>
<interface_name>vlan10</interface_name>
<interface_status>up</interface_status>
<interface_hardware>VLAN</interface_hardware>
<interface_mac>
<macaddress>00:05:9a:3b:92:b1</macaddress>
</interface_mac>
<interface_mode>routed</interface_mode>
<interface_ip>
<ipaddress>10.20.105.101</ipaddress>
<ipmask>255.255.255.0</ipmask>
</interface_ip>
<interface_ft_status>non-redundant</interface_ft_status>
<interface_description>
<interface_description>not set</interface_description>
</interface_description>
<interface_mtu>1500</interface_mtu>
<interface_last_cleared>never</interface_last_cleared>
<interface_alias>
<ipaddress>not set</ipaddress>
</interface_alias>
```

```
<interface_standby>
<ipaddress>not set</ipaddress>
</interface_standby>
<interface_sup_enabled>Assigned</interface_sup_enabled>
<interface_auto_status>up</interface_auto_status>
</xml_interface>
<interface_stats>
<ifs_input>
<ifs_unicast>50</ifs_unicast>
<ifs_bytes>8963</ifs_bytes>
<ifs_multicast>26</ifs_multicast>
<ifs_broadcast>1</ifs_broadcast>
<ifs_errors>0</ifs_errors>
<ifs_unknown>0</ifs_unknown>
<ifs_ignored>0</ifs_ignored>
<ifs_unicast_rpf>0</ifs_unicast_rpf>
</ifs_input>
<ifs_output>
<ifs_unicast>45</ifs_unicast>
<ifs_bytes>5723</ifs_bytes>
<ifs_multicast>0</ifs_multicast>
<ifs_broadcast>1</ifs_broadcast>
<ifs_errors>0</ifs_errors>
<ifs_ignored>0</ifs_ignored>
</ifs_output>
</interface_stats>
</xml_interface_entry>
</xml_show_interface>
</xml_show_result>
</exec_command>
</response_xml>
```

Details

Command	Purpose
xml-show {**off** | **on** | **status**} **Example:** host1/Admin# xml-show on	Enables the display of raw XML request **show** command output in XML format. The keywords are as follows: • **off**—Displays CLI **show** command output in regular CLI display output, not in XML format. • **on**—Displays CLI **show** command output in XML format unless a specific **show** command is not implemented to display its output in XML format. For details on the **show** command output supported in XML format, consult the the ACE DTD file, cisco_ace.dtd, that is included as part of the software image (see the "Accessing the ACE DTD File" section). • **status**—Displays the results of the **xml show** command status: on or off. The **status** keyword allows you to determine the status of the **xml show** command setting.

Accessing the ACE DTD File

This section describes how to access the ACE DTD file to perform one of the following tasks:

- Directly access the cisco_ace.dtd file.
- Open the cisco_ace.dtd file from the Cisco ACE Module Management page.

The ACE DTD file, cisco_ace.dtd, is included as part of the software image and is accessible from a web browser using either HTTP or HTTPS.

Details

Perform these steps to access and display the Cisco ACE DTD 3.0 file:

Step 1 If you have not done so, create a Layer 3 and Layer 4 class map and policy map to classify the HTTP or HTTPS management traffic that can be received by the ACE. See the "Configuring HTTP and HTTPS Management Traffic Services" section.

Step 2 Open your preferred Internet web browser application, such as Microsoft Internet Explorer or Netscape Navigator.

Step 3 Access the cisco_ace.dtd file.

To directly access the cisco_ace.dtd file, specify the HTTP or secure HTTP (HTTPS) address of your ACE in the address field, followed by cisco_ace.dtd. For example, enter:

```
https://ace_ip_address/cisco_ace.dtd

http://ace_ip_address/cisco_ace.dtd
```

You can choose to either open the cisco_ace.dtd file or save it to your computer.

To access the cisco_ace.dtd file from the Cisco ACE ModuleManagement page, perform the following steps:

a. Specify the HTTP or secure HTTP (HTTPS) address of your ACE in the address field:

```
https://ace_ip_address

http://ace_ip_address
```

b. Click **Yes** at the prompt to accept (trust) and install the signed certificate from Cisco. To install the signed certificate, do one of the following:

   – If you are using Microsoft Internet Explorer, in the Security Alert dialog box, click **View Certificate**, choose the **Install Certificate** option, and follow the prompts of the Certificate Manager Import Wizard.

   – If you are using Netscape Navigator, in the New Site Certificate dialog box, click **Next** and follow the prompts of the New Site Certificate Wizard.

c. Enter your username and password in the fields provided, and then click **OK**. The Cisco ACE Module Management page appears.

d. Click the CISCO ACE DTD 3.0 link under the Resources column of the Cisco ACE Module Management page to access the cisco_ace.dtd file. You can choose to either open the cisco_ace.dtd file or save it to your computer.

Displaying or Clearing XML Service Policy Statistics

This section describes how to display or clear XML service policy statistics and contains the following topics:

Displaying XML Service Policy Statistics

To display the statistical information of the service policies associated with your XML configuration, perform the following task:

Command	Purpose
show service-policy *policy_name* [**detail**]	Displays service policy statistics for a Layer 3 and Layer 4 management policy map. The keywords, options, and arguments are as follows: • *policy_name*—Identifier of an existing policy map that is currently in service (applied to an interface) as an unquoted text string with a maximum of 64 alphanumeric characters. • **detail**—(Optional) Displays a more detailed listing of policy map statistics and status information. **Note** The ACE updates the counters that the **show service-policy** command displays after the applicable connections are closed.

Examples

The following example shows the output for the MGMT_HTTPS_POLICY policy map by using the **show service-policy** command:

```
host1/Admin# show service-policy MGMT_HTTPS_POLICY
Status      : ACTIVE
Description: Allow mgmt protocols
----------------------------------------
Context Global Policy:
  service-policy: MGMT_HTTPS_POLICY
```

Clearing XML Service Policy Statistics

To clear the statistical information of the service policies associated with your XML configuration, perform the following task:

Command	Purpose
clear service-policy *policy_name*	Clears the service policy statistics. For the *policy_name* argument, enter the identifier of an existing policy map that is currently in service (applied to an interface) as an unquoted text string with a maximum of 64 alphanumeric characters.

Example of ACE CLI Command and the XML Equivalent

The following example shows a typical VShell (VSH) CLI command configuration and its equivalent XML configuration commands:

```
##############################
## TO/FROM CP CONFIGURATION ##
##############################
conf t
access-list acl1 extended permit ip any any
int vlan 80
access-group input acl1
ip address 60.0.0.145 255.255.255.0
no shut
exit
ip route 0.0.0.0 0.0.0.0 60.0.0.1
end

<access-list id="acl1" config-type="extended" perm-value="permit"
protocol-name="ip" src- type="any" dest-type="any"/>
<interface type="vlan" number="80">
<access-group type="input" name="acl1"/>
<ip_address address="60.0.0.145" netmask="255.255.255.0"/>
<shutdown sense="no"/>
</interface>

<ip_route dest-address="0.0.0.0" dest-mask="0.0.0.0"
gateway="60.0.0.1"/>
############################
## BRIDGING CONFIGURATION ##
############################
conf t


access-list acl1 extended permit ip any any
int vlan 80
access-group input acl1
bridge-group 1
no shut
exit

int vlan 90
access-group input acl1
bridge-group 1
no shut
exit
end

<access-list id="acl1" config-type="extended" perm-value="permit"
protocol-name="ip" src-type="any" dest-type="any"/>
<interface type="vlan" number="80">
<access-group type="input" name="acl1"/>
<bridge-group value="1"/>
<shutdown sense="no"/>
</interface>
<interface type="vlan" number="90">
<access-group type="input" name="acl1"/>
<bridge-group value="1"/>
<shutdown sense="no"/>
</interface>
```