CHAPTER 1

Introduction to the Cisco WAAS Central Manager Monitoring API

This chapter describes the Cisco WAAS Central Manager monitoring application programming interface (API), which provides a programmable interface for system developers to integrate with customized or third-party monitoring and management applications.

This chapter contains the following sections:

Monitoring API Overview

The Central Manager monitoring API communicates with the WAAS Central Manager to retrieve status information and monitoring statistics. This API does not allow device configuration.

The Central Manager monitoring API is a Web Service implementation. Web Service is defined by the W3C standard as a software system designed to support interoperable machine-to-machine (client and server) interaction over the network. The client and server communication follows the Simple Object Access Protocol or Service Oriented Architecture Protocol (SOAP) standard.

SOAP, which exhanges XML-based messages over the network using HTTP or HTTPS, is the foundation layer of the Web Service stack. It provides a basic messaging framework that allows more abstract layers to build on. SOAP encoding wraps XML headers and tags in a SOAP envelope.

To call a service, you connect to a particular Central Manager through a web browser by using a service URL that contains the IP address or hostname of the Central Manager and the name of the particular monitoring service (such as DeviceConf or TrafficStats). For example, https://<host/ip>:8443/ws/TrafficStats is the service URL for the Traffic Acceleration Service.

Next, you must post a SOAP request written in an XML format to retrieve the information. The request calls for a particular action (such as retrieveTrafficStats) and contains the WS-Security header (username and password) and the input parameter content when required. The Central Manager responds with a SOAP envelope that contains the answer in an XML format. The response contains the output values for this action.

SOAP message exchanges follow the WS-Security specification. The WS-Security specification provides a Username Token mechanism to authenticate SOAP message exchanges.

The following example shows an XML-formatted SOAP request to perform the getWANInfo action. There are no input parameters for this particular action. The example then shows the SOAP response that contains the output values for this action, such as the hostname, IP address, location, MAC address, and so forth.

Request

```
<?xml version="1.0" encoding="UTF-8" ?>
- <SOAP-ENV:Envelope SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/1999/XMLSchema">
- <SOAP-ENV:Header xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401
-wss-wssecurity-secext-1.0.xsd">
- <wsse:Security SOAP-ENC:root="1">
- <wsse:UsernameToken>
  <wsse:Username xsi:type="xsd:string">admin</wsse:Username>
  <wsse:Password xsi:type="xsd:string">default</wsse:Password>
  </wsse:UsernameToken>
  </wsse:Security>
  </SOAP-ENV:Header>
- <SOAP-ENV:Body>
  <ns1:getWANInfo xmlns:ns1="http://config.ws.waas.cisco.com" SOAP-ENC:root="1" />
  </SOAP-ENV:Body>
  </SOAP-ENV:Envelope>
```

Response

```
<?xml version="1.0" encoding="UTF-8" ?>
- <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
- <soapenv:Body>
-   <ns:getWANInfoResponse xmlns:ns="http://config.ws.waas.cisco.com"
xmlns:ax22="http://io.java/xsd" xmlns:ax23="http://config.ws.waas.cisco.com/xsd"
xmlns:ax21="http://rmi.java/xsd">
- <ns:return type="com.cisco.waas.ws.config.Device">
  <ax23:hostName xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:nil="true" />
  <ax23:id>157</ax23:id>
  <ax23:ipAddress>2.43.153.39</ax23:ipAddress>
  <ax23:location />
  <ax23:macAddress>00:14:5e:84:35:59</ax23:macAddress>
  <ax23:model>OE612</ax23:model>
  <ax23:name>ce-119-39</ax23:name>
  <ax23:role>Primary</ax23:role>
  <ax23:softwareVersion>4.1.0.b.51</ax23:softwareVersion>
  <ax23:status>Online</ax23:status>
  <ax23:type>CM</ax23:type>
  </ns:return>
</ns:getWANInfoResponse>
  </soapenv:Body>
  </soapenv:Envelope>
```

The Central Manager monitoring API consists of the following eight Web Services:

- Device Configuration
- Traffic Acceleration
- CIFS
- Video Stream
- HTTP and HTTPS

- MAPI
- NFS
- Events and Status

Administrators may control API access for a device or device group by configuring user authorization settings using the CLI or the Central Manager GUI. The authorization for a Web Service is implemented system wide as mandatory and at the service level as optional.

Web Services Description Language

In the Central Manager monitoring API, Web Services Description Language (WSDL) is used with SOAP and XML schemas to provide Web Services. WSDL is an XML-based service that describes the functionality offered by the Web Service and defines the actions, parameter names, input parameter data types, and return data types for the Web Service. When you connect to a Web Service through a web browser, you can read the WSDL file to determine which functions are available on the server. Any special data types that are used are embedded in the WSDL file in an XML schema. You can then call one of the functions listed in the WSDL file by sending a SOAP request message.

To obtain the WSDL file defined for a particular service in the Central Manager monitoring API implementation, submit a URL to the service with a **?wsdl** suffix. For example, to retrieve the WSDL for the TrafficStats service running on https://localhost:8443/ws/TrafficStats, call the WSDL file by using the URL **https://localhost:8443/ws/TrafficStats?wsdl**.

Using the Central Manager Monitoring API

This section describes how to use the Central Manager monitoring API. It contains the following topics:

Required Software, Web Standards, and Supported Hardware

The Central Manager monitoring API is supported in WAAS version 4.1.1 and later. The API requires the following development environment:

- Apache Axis2 (Version 2.1.3)
- WSDL Support: 1.1 and 2.0
- SOAP 1.1 and 1.2
- Axis Data Binding (ADB)
- WS-Security

The Central Manager monitoring API is supported on the following hardware models:

WAVE-274-K9	WAE-512-K9	WAE-612-K9	WAE-7341-K9
WAVE-474-K9	WAVE-574-K9	WAE-674-K9	WAE-7371-K9
WAE-511-K9	WAE-611-K9	WAE-7326-K9	

Generating the Client Code to Invoke a Web Service

You can use the WSDL2java utility to generate the client code which calls and implements a Web Service. The WSDL2java utility takes a WSDL document and generates fully annotated Java code from which to implement the service.

To use the WSDL2java utility, follow these steps:

Step 1 Query the Central Manager for the WDSL definitions for a particular service by using the following WSDL URL format:

```
https://<host/ip>:8443/ws/<NameOfService>?wsdl
```

where the host/ip value is the hostname or IP address of the Central Manager that has the service running, and the NameOfService value is the Web Service designation.

Step 2 Save the XML response to a file, such as NameOfService.xml.

Step 3 Call the WSDL2java script for your development environment: wsdl2java.sh or wsdl2java.bat. These scripts can be found under the bin directory of the Axis2 distribution.

Step 4 Run the following command line to generate the client code:

```
wsdl2java -uri NameOfService.xml -p com.cisco.waas.wsc -d adb -s
```

The WSDL2Java command is run against the WSDL file to create deployment descriptor templates. The utility processes the WSDL file and generates JAVA code based on the WDSL definitions for a particular service.

You may then create scripts using any general-purpose, high-level programming language, such as Python, to generate SOAP requests and parse SOAP responses.