Preface

Revised: May 26, 2017, OL-25996-01

This preface explains the objectives, intended audience, and organization of the *Cisco Identity Services Engine API Reference Guide, Release 1.0.4*. The preface also describes the conventions that provide instructions and provides other types of information in the following sections:

Overview of Cisco Identity Services Engine

Cisco Identity Services Engine (ISE), as a next-generation identity and access control policy platform enables enterprises to enforce compliance, enhance infrastructure security, and streamline their service operations. The unique architecture of Cisco ISE allows enterprises to gather real-time contextual information from networks, users, and devices in order to make proactive governance decisions by tying identity to various network elements including access switches, wireless LAN controllers (WLCs), virtual private network (VPN) gateways, and data center switches.

Cisco ISE is a key component of the Cisco Security Group Access Solution. Cisco ISE is a consolidated policy-based access control solution that:

- Combines authentication, authorization, accounting (AAA), posture, profiler, and guest management services into one appliance
- Enforces endpoint compliance by checking the device posture of all endpoints accessing the network, including 802.1X environments
- Provides support for discovery, profiling, policy-based placement, and monitoring of endpoint devices on the network
- Enables consistent policy in centralized and distributed deployments allowing services to be delivered where they are needed

- Employs advanced enforcement capabilities including Security Group Access (SGA) through the use of Security Group Tags (SGTs) and Security Group (SG) Access Control Lists (ACLs)
- Supports scalability to support a number of deployment scenarios from small office to large enterprise environments

Cisco ISE comes preinstalled on a range of physical appliances to suit your network size with various performance characterizations. The inherent scalability of Cisco ISE allows you to add appliances to a deployment and increase performance and resiliency, as needed. The Cisco ISE architecture supports standalone and distributed deployments, along with high availability options. Cisco ISE allows you to configure and manage your network from a centralized portal for efficiency and ease of use.

Cisco ISE also incorporates distinct configurable roles and services, so that you can create and apply Cisco ISE services where they are needed in the network. The result is a comprehensive Cisco ISE deployment that operates as an fully functional and integrated system.

Purpose

This application programming interface (API) reference guide provides only a brief high-level overview of the capabilities afforded by the supported APIs. The purpose of this API reference guide is to provide a developer, system or network administrator, or system integrator with some basic guidelines for using the capabilities of the APIs to monitor data in selected Cisco Monitoring ISE nodes within the Cisco ISE deployment. The API calls use queries to determine the following types of data:

- Number of active sessions
- Types of active sessions
- Authentication status of active session
- MAC addresses in use
- NAS IP addresses in use
- Node versions and types
- Reasons for node session failures

Note This API reference guide is not intended to replace the *Cisco Identity Services Engine User Guide, Release 1.0.4*. For more information about the Cisco ISE network, its nodes and personas, concepts of operation or usage, or how to use the Cisco ISE user interface, see the *Cisco Identity Services Engine User Guide, Release 1.0.4* (for example, the Glossary contains a complete listing of key terms and concepts used in Cisco ISE networks).

Audience

This API reference guide is intended for experienced system administrators who administer Cisco ISE appliances within a network environment, system integrators who may want to make use of the APIs, or third-party partners who have with the responsibility for managing or troubleshooting Cisco ISE deployments. As a prerequisite to using this API reference guide, you should have a basic understanding of troubleshooting and diagnostic practices and how to make and interpret API calls.

Document Organization

This guide contains the following chapters and appendixes:

- Chapter 1, "Introduction to the Cisco ISE REST APIs"
- Chapter 2, "Using the Query APIs for Session Management"
- Chapter 4, "Using the Change of Authorization REST APIs"
- Chapter 3, "Using the Query APIs for Troubleshooting"
- Appendix A, "Using the Cisco ISE Failure Reasons Editor"

Document Conventions

Caution Means *reader be careful*. You are capable of doing something that might result in equipment damage or loss of data.

Note Means *reader take note*. Notes contain helpful suggestions or references to materials not contained in this manual.

This API reference guide uses the following conventions to convey instructions and information.

Item	Convention
Commands, keywords, special terminology, and options that should be chosen during procedures	**boldface** font
Variables for which you supply values and new or important terminology	*italic* font
Displayed session and system information, paths, and file names	`screen` font
Information you enter	**`boldface screen`** font
Variables you enter	*`italic screen`* font
Menu items and button names	**boldface** font
Indicates menu items to choose, in the order in which you choose them.	**Option > Network Preferences**

Product Documentation

Note We sometimes update the printed and electronic documentation after original publication. Therefore, you should also review the documentation on http://www.cisco.com for any updates.

Table 2 lists the related product documentation that is available for Cisco ISE Release 1.0.4 on www.cisco.com. To find end-user documentation for all products on www.cisco.com, go to:

http://www.cisco.com/go/techdocs

Documentation Updates

Table 1 lists the documentation updates for this Cisco ISE product release.

Table 1 Updates for Cisco Identity Services Engine API Reference Guide, Release 1.0.4

Date	Description
9/302011	Cisco Identity Services Engine Maintenance Release 1.0.4.573. No content updates made.
8/26/2011	Corrected schema syntax errors
8/15/2011	Added information on Removing Stale Sessions, page 24
5/17/2011	Cisco Identity Services Engine (ISE) Release 1.0

Related Documentation

This section provides information on release-specific documentation, as well as platform-specific documentation.

Release-Specific Documentation

Table 2 lists the product documentation available for the Cisco ISE Release. General product information for Cisco ISE is available at http://www.cisco.com/go/ise. End-user documentation is available on Cisco.com at http://www.cisco.com/en/US/products/ps11640/tsd_products_support_series_home.html.

Table 2 Product Documentation for Cisco Identity Services Engine

Document Title	Location
Release Notes for the Cisco Identity Services Engine, Release 1.0.4	http://www.cisco.com/en/US/products/ps11640/prod_release_notes_list.html
Cisco Identity Services Engine Network Component Compatibility, Release 1.0.4	http://www.cisco.com/en/US/products/ps11640/products_device_support_tables_list.html
Cisco Identity Services Engine User Guide, Release 1.0.4	http://www.cisco.com/en/US/products/ps11640/products_user_guide_list.html

Table 2 Product Documentation for Cisco Identity Services Engine (continued)

Document Title	Location
Cisco Identity Services Engine Hardware Installation Guide, Release 1.0.4	http://www.cisco.com/en/US/products/ps11640/prod_installation_guides_list.html
Cisco Identity Services Engine Migration Guide for Cisco Secure ACS 5.1 and 5.2, Release 1.0.4	http://www.cisco.com/en/US/products/ps11640/prod_installation_guides_list.html
Cisco Identity Services Engine Sponsor Portal User Guide, Release 1.0.4	http://www.cisco.com/en/US/products/ps11640/products_user_guide_list.html
Cisco Identity Services Engine CLI Reference Guide, Release 1.0.4	http://www.cisco.com/en/US/products/ps11640/prod_command_reference_list.html
Cisco Identity Services Engine Troubleshooting Guide, Release 1.0.4	http://www.cisco.com/en/US/products/ps11640/prod_troubleshooting_guides_list.html
Regulatory Compliance and Safety Information for Cisco Identity Services Engine, Cisco 1121 Secure Access Control System, Cisco NAC Appliance, Cisco NAC Guest Server, and Cisco NAC Profiler	http://www.cisco.com/en/US/products/ps11640/prod_installation_guides_list.html
Cisco Identity Services Engine In-Box Documentation and China RoHS Pointer Card	http://www.cisco.com/en/US/products/ps11640/products_documentation_roadmaps_list.html

Platform-Specific Documentation

Links to Policy Management Business Unit documentation are available on www.cisco.com at the following locations:

- Cisco ISE
  http://www.cisco.com/en/US/products/ps11640/prod_installation_guides_list.html
- Cisco Secure ACS
  http://www.cisco.com/en/US/products/ps9911/tsd_products_support_series_home.html
- Cisco NAC Appliance
  http://www.cisco.com/en/US/products/ps6128/tsd_products_support_series_home.html
- Cisco NAC Profiler
  http://www.cisco.com/en/US/products/ps8464/tsd_products_support_series_home.html
- Cisco NAC Guest Server
  http://www.cisco.com/en/US/products/ps10160/tsd_products_support_series_home.html

Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, submitting a service request, and gathering additional information, refer to the monthly *What's New in Cisco Product Documentation*, which also lists all new and revised Cisco technical documentation, at:

http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html

Subscribe to the *What's New in Cisco Product Documentation* as a Really Simple Syndication (RSS) feed and set content to be delivered directly to your desktop using a reader application. The RSS feeds are a free service and Cisco currently supports RSS version 2.0.

CHAPTER 1

Introduction to the Cisco ISE REST APIs

The Cisco Identity Services Engine API Reference Guide, Release 1.0.4, provides you with guidelines and examples for using the three supported categories of representational state transfer (REST) APIs and related API calls. The REST APIs and calls allow you to gather session and node-specific information by using Cisco Monitoring ISE nodes in your network. A session is defined as the duration between when you start accessing the desired node and completing the set of tasks or operations needed to gather information.

The supported categories of REST APIs that are available to users in Cisco ISE Release 1.0.4 are:

- Query
  - Session Management
  - Troubleshooting
- Change of Authorization (CoA)

Note You can use only these supported REST API categories to gather information about endpoints being monitored by the Monitoring persona. Monitoring is one of three supported personas that an ISE node type can perform in your Cisco ISE Release 1.0.4 deployment. For the remainder of this guide, "Monitoring ISE node" will be used to describe the Monitoring persona of a Cisco ISE node.

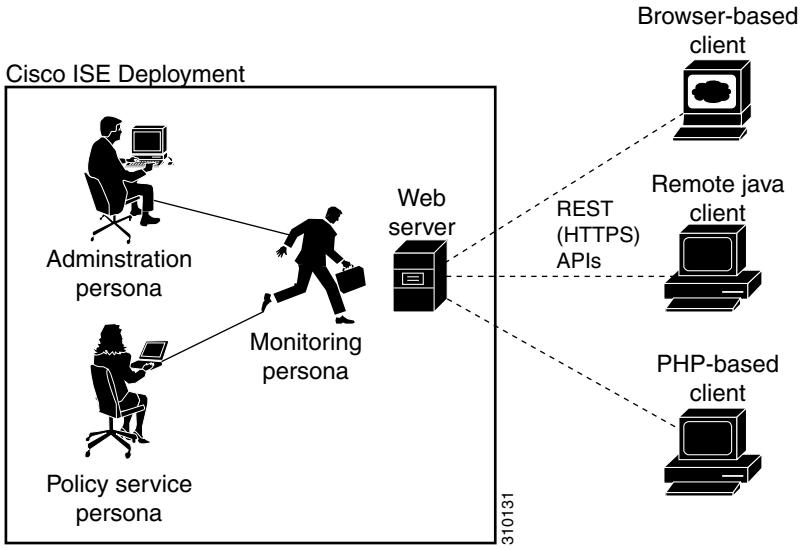
Any attempt to use these APIs to gather information about the Policy Service persona of a Cisco ISE appliance in a Cisco ISE deployment will result in an error. For more information about Cisco ISE nodes and personas, see the *Cisco Identity Services Engine User Guide, Release 1.0.4*.

The REST API calls provide the means for you to locate, monitor, and accumulate important real-time session-based information stored in individual endpoints in your network that you can access through a Cisco Monitoring ISE node.

The real-time session-based information that you gather can prove useful to understand Cisco ISE operations, assist in diagnosing conditions or issues, or be used to troubleshoot error conditions or activity or behavior that you suspect may be affecting your monitoring operations. The role that the REST APIs play in a Cisco ISE distributed deployment is shown in Figure 1-1.

As shown in Figure 1-1, the REST (HTTPS) API calls are used by supported client types: remote Java, browser-based, or PHP (hypertext preprocessor), and for the purpose of accessing the Cisco Monitoring ISE node and retrieving important session-based information that is stored in the Cisco ISE deployment endpoints.

Figure 1-1 Cisco ISE Distributed Deployment and REST APIs



Verifying a Cisco Monitoring ISE Node

Before you can successfully invoke the API calls on a Cisco Monitoring ISE node, you first need to verify that the node you want to monitor is a valid Cisco Monitoring ISE node. To verify this, you need to successfully log into and be authenticated by the Cisco ISE network.

Note To be able to use the public REST APIs, you must first authenticate with Cisco ISE using valid credentials for any of the supported Cisco ISE admin roles (Helpdesk Admin, Identity Admin, Monitoring Admin, Network Device Admin, Policy Admin, RBAC Admin, Super Admin, or System Admin).

To login and be authenticated, complete the following steps:

Step 1 Enter valid login credentials (Username and Password) in the Cisco ISE Login window, and click **Login**.

The Cisco ISE dashboard and user interface appears.

Step 2 Choose **Authorization > System > Deployment**.

The Deployment Nodes page appears, which lists all configured nodes that are deployed.

Step 3 In the Roles column of the Deployment Nodes page, verify that the role for the target node that you want to monitor shows its type as a Cisco Monitoring ISE node.

Supported API Calls

This section introduces the REST APIs, which provide an interface for programmatically issuing calls that retrieve and display the node-specific or session-specific information. The following tables list the API category, type of API call, and provide a brief description and an example of the API call format:

- Table 1-1 on page 1-3—defines the query API calls for session management.
- Table 1-2 on page 1-6—defines the query API calls for troubleshooting.
- Table 1-3 on page 1-7—defines the CoA API calls.

Note Before you can perform any of the API calls described in this guide, you first need to log into and be authenticated by the Cisco ISE network. The authentication requirement for using the public REST APIs is explained in Verifying a Cisco Monitoring ISE Node, page 1-2.

If you intend to use a generic programmatic interface to authenticate with the REST API supported by Cisco ISE, you would need to first create a REST-based client that bridges between Cisco ISE and the specific tool you use. You would then use this REST client to perform authentication with the Cisco ISE REST APIs, marshal and submit the API requests to the Monitoring ISE nodes, and unmarshal the API responses and pass these responses on to the specified tool.

Table 1-1 Cisco ISE Query API Calls - Session Management

Cisco ISE API Call Category	Cisco ISE API Call Description and Example
Session Management	
Session Counters	
• Active sessions counter	Lists the number of currently active sessions: *https://<ISEhost>/ise/mnt/api/Session/ActiveCount*
• Posture sessions counter	Lists the number of currently active Posture service sessions: *https://<ISEhost>/ise/mnt/api/Session/PostureCount* **Note** Posture is a service that aids in checking the state (or posture) for all the endpoints that connect to your Cisco ISE network.
• Profiler sessions counter	Lists the number of currently active Profiler service sessions: *https://<ISEhost>/ise/mnt/api/Session/ProfilerCount* **Note** Profiler is a service that aids in identifying, locating, and determining the capabilities of all attached endpoints on your Cisco ISE network.

Table 1-1 Cisco ISE Query API Calls - Session Management (continued)

Cisco ISE API Call Category	Cisco ISE API Call Description and Example
Simple Session List **Note** A simple session list includes the MAC address, network access switch (NAS) IP address, user name, and session ID information associated with a session.The Cisco Identity Services Engine, Release 1.0.4, is not compliant with IPv6. **Note** The level of support for IPv6 in Cisco ISE is only as it relates to the node being addressed on an IPv6 network (for example, IPv6 stateless auto-configuration and DHPv6). However, none of the Cisco ISE, Release 1.0.4, protocol stacks (such as runtime or mgmt) supports IPv6.	
• Active sessions list	Lists all currently active sessions: *https://<ISEhost>/ise/mnt/api/Session/ActiveList* **Note** In this release of Cisco ISE, the maximum number of active authenticated endpoint sessions that can be displayed is limited to 100,000.
• Authenticated sessions list	Lists all currently active authenticated sessions: *https://<ISEhost>/ise/mnt/api/Session/AuthList/<parameteroptions>* **Note** The starttime/endtime format is yyyy-mm-dd hh24:MM:ss (for example, 2010-12-10 16:30:00). **Note** You can specify the following parameter options that will return different values: • If null/null is specified, this lists all currently active authenticated sessions. • If null/endtime is specified, this list all currently active authenticated sessions after the specified endtime. • If starttime/null is specified, this lists all currently active authenticated sessions before the specified starttime. • If starttime/endtime is specified, this lists all currently active authenticated sessions between the specified starttime and endtime. See Sample Data Returned from the AuthList API Call, page 2-9, for samples that show all four parameter options. **Note** In this release of Cisco ISE, the maximum number of active authenticated endpoint sessions that can be displayed is limited to 100,000.

Table 1-1 Cisco ISE Query API Calls - Session Management (continued)

Cisco ISE API Call Category	Cisco ISE API Call Description and Example
Detailed Session Attributes **Note** This is a timestamp-based search for the latest session that contains the specified search attribute.	
• MAC address session search	Searches the database for the latest session that contains the specified MAC address: *https://<ISEhost>/ise/mnt/api/Session/MACAddress/<macaddress>* **Note** XX:XX:XX:XX:XX:XX is the MAC address format and is not case-sensitive (for example, 0a:0B:0c:0D:0e:0F). **Note** The MAC address serves as the only unique key to finding the correct session you want to monitor. Use the ActiveList API call to list all active sessions and their MAC addresses, from which you can base your MAC address search.
• User name session search	Searches the database for the latest session that contains the specified user name: *https://<ISEhost>/ise/mnt/api/Session/UserName/<username>* **Note** User names must conform to the same Cisco ISE password policy used for network user names. The only invalid character for REST APIs is the backslash (/) character. For details, see "User Password Policy" in the *Cisco Identity Services Engine User Guide, Release 1.0.4*.
• NAS IP address session search	Searches the database for the latest session that contains the specified NAS IP address: *https://<ISEhost>/ise/mnt/api/Session/IPAddress/<nasipaddress>* **Note** xxx.xxx.xxx.xxx is the NAS IP address format (for example, 10.10.10.10).

For specific details about the Cisco ISE query API calls for session management, see Chapter 2, "Using the Query APIs for Session Management".

Table 1-2 Cisco ISE Query API Calls - Troubleshooting

Cisco ISE API Call Category	Cisco ISE API Call Description and Example
Query - Troubleshooting	
Get Version and Type of Node	
• Node version and type	Lists the node version and type: *https://<ISEhost>/ise/mnt/api/Version* Node type can be any of the following values (0-3): STAND_ALONE_MNT_NODE = 0 ACTIVE_MNT_NODE = 1 STAND_BY_MNT_NODE = 2 NOT_AN_MNT_NODE = 3 **Note** STAND_ALONE_MNT_NODE means it is a Cisco Monitoring ISE node that functions not as part of any distributed deployment. ACTIVE_MNT_NODE means it is a primary node in a primary-secondary relationship in a distributed deployment. STAND_BY_MNT_NODE means it is a secondary node in a primary-secondary pair in this same type of deployment. NOT_AN_MNT_NODE means it is not a Cisco Monitoring ISE node. See the *Cisco Identity Services Engine User Guide, Release 1.0.4* for details about the supported ISE nodes and personas.
Get Failure Reasons Mapping	
• Failure reasons	Lists the reasons for failure: *https://<ISEhost>/ise/mnt/api/FailureReasons* Each failure reason displays an error code (failureReason id), a brief description (code), a failure reason (cause), and a possible response (resolution), as shown in the following example: <failureReason id="100009"> <code> 100009 WEBAUTH_FAIL <cause> This may or may not be indicating a violation. <resolution> Please review and resolve this issue according to your organization's policy. **Note** The use case for which the FailureReasons API call is designed addresses the need for it to be called only once to gather the information from the Monitoring ISE node. You should store the contents of any returned failure reasons into your own file system or database. The returned contents of these API calls are intended to be used for reference purposes. If you experience any issues during authentication, you should compare the failure reason code provided in the authentication response with the list of failure reasons that you have stored in you own file system or database. For a complete list of Cisco ISE failure reasons, see Appendix A, "Using the Cisco ISE Failure Reasons Editor".

Table 1-2 Cisco ISE Query API Calls - Troubleshooting (continued)

Cisco ISE API Call Category	Cisco ISE API Call Description and Example
Get Session Auth Status	
• Session authentication status	Lists the authentication status for all sessions: *https://<ISEhost>/ise/mnt/api/AuthStatus/MACAddress/<macaddress>/<numberofseconds>/<numberofrecordspermacaddress>/All* **Note** The seconds parameter <numberofseconds> is user-configurable, with the range being from a minimum of 0 to a maximum of 432000 seconds (5 days). **Note** Authentication status is defined as when all of the data fields are available in the RADIUS_AUTH table.
Get Session Accounting Status	
• Accounting session status	Lists the accounting status of all sessions within a specific period of time: *https://<ISEhost>/ise/mnt/api/Session/AcctStatusTT/MACAddress/<macaddress>/<numberof seconds>* **Note** The seconds parameter <numberofseconds> is user-configurable, with the range being from a minimum of 0 to a maximum of 432000 seconds (5 days).

For specific details about the Cisco ISE Query API calls for Troubleshooting, see Chapter 2, "Using the Query APIs for Session Management".

Table 1-3 Cisco ISE Change of Authorization API Calls

Cisco ISE API Call Category	Cisco ISE API Call Description and Example
CoA Session Management	
Session Reauth	
• Session reauthentication types	Sends a session reauthentication command and type: *https://<ISEhost>/ise/mnt/api/CoA/Reauth/<serverhostname>/<macaddress>/<reauthtype>/<nasipaddress>/<destinationipaddress>* Reauth type can be any of the following values (0-2): REAUTH_TYPE_DEFAULT = 0 REAUTH_TYPE_LAST = 1 REAUTH_TYPE_RERUN = 2 **Note** If you do not know the NAS IP address, you can enter the required values up to that point and the API will use these values in its search query. However, you must know the MAC address to perform this API call. This API call can only be executed on a Monitoring ISE node, which submits the requests to perform CoA remotely. The Administration ISE node is not involved or required to execute these CoA API calls.

Table 1-3 Cisco ISE Change of Authorization API Calls (continued)

Cisco ISE API Call Category	Cisco ISE API Call Description and Example
Session Disconnect	
• Session disconnect types	Sends a session disconnect command and port option type: *https://<ISEhost>/ise/mnt/api/CoA/Disconnect/<serverhostname>/<macaddress>/<disconnecttype>/<nasipaddress>/<destinationipaddress>* **Note** Port option type can be any of the following values (0-2): DYNAMIC_AUTHZ_PORT_DEFAULT = 0 DYNAMIC_AUTHZ_PORT_BOUNCE = 1 DYNAMIC_AUTHZ_PORT_SHUTDOWN = 2 **Note** If you do not know the NAS IP address, enter the required values up to that point and the API will use these values in its search query. However, you must know the MAC address to perform this API call.

For details about Cisco ISE Change of Authorization API calls, see Chapter 4, "Using the Change of Authorization REST APIs".

Supported API Calls using HTTP PUT

Similar to a Get Session Auth Status API call in Table 1-2, there is an HTTP PUT version of a REST API implemented that allows clients to retrieve account status. The REST APIs support both HTTP PUT and HTTP GET calls, with the examples in this guide documenting HTTP GET calls. The HTTP PUT version addresses the need for APIs that require parameter inputs. The following schema file example is a request for account status:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="acctRequest" type="mnTRESTAcctRequest"/>

  <xs:complexType name="mnTRESTAcctRequest">
    <xs:complexContent>
      <xs:extension base="mnTRESTRequest">
        <xs:sequence>
          <xs:element name="duration" type="xs:string" minOccurs="0"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
```

```
  <xs:complexType name="mnTRESTRequest" abstract="true">
    <xs:sequence>
      <xs:element name="valueList">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="value" type="xs:string" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="searchCriteria" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

CHAPTER 2

Using the Query APIs for Session Management

This chapter provides examples and describes using the following individual session management REST API calls that are supported in this release of Cisco ISE. The session management API calls provide the means for retrieving important session-related information from within the Cisco Monitoring ISE node in your Cisco ISE deployment.

The following sections provide API output schema file examples, procedures for issuing each API call, and a sample of the data returned by each API call:

Using the Session Counter API Calls

The following session counter API calls let you quickly gather a current count of session-related information on a target Cisco Monitoring ISE node in your Cisco ISE deployment:

- Active sessions (ActiveCount)
- Posture sessions (PostureCount)
- Profiler sessions (ProfilerCount)

Active Sessions Counter

You can use the ActiveCount API call to retrieve a count of all currently active sessions. This section provides a schema file output example, a procedure for counting all active sessions by invoking the ActiveCount API call, and a sample of the active sessions data returned after this API call is issued.

ActiveCount API Output Schema

This sample schema file is the output of the ActiveCount API call for retrieving a count of the active sessions on the target Monitoring persona of an ISE node:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="sessionCount" type="activeCount"/>
  <xs:complexType name="activeCount">
    <xs:sequence>
      <xs:element name="count" type="xs:int"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

Invoking the ActiveCount API Call

Note Make sure that you have verified that the target node to which you are issuing an API call is a valid Cisco Monitoring ISE node. To verify the persona of a Cisco ISE node, see Verifying a Cisco Monitoring ISE Node, page 1-2.

To issue the ActiveCount API call, complete the following steps:

Step 1 Log into the target Cisco Monitoring ISE node.

For example, when you initially log into a Cisco Monitoring ISE node with the hostname of acme123, this would display the following URL Address field for this node:

```
https://acme123/admin/LoginAction.do#pageId=com_cisco_xmp_web_page_tmpdash
```

Step 2 Enter the ActiveCount API call in the URL Address field of the target node by replacing the "/admin/" component with the API call component (/ise/mnt/api/<specific-api-call>):

```
https://acme123/ise/mnt/api/Session/ActiveCount
```

Note You must carefully enter each API call in the URL Address field of a target node because these calls are case-sensitive. The use of "mnt" in the API call convention represents the target Cisco Monitoring ISE node.

Step 3 Press **Enter** to issue the API call.

Sample Data Returned from the ActiveCount API Call

The following example illustrates the data returned (number of active sessions) when you invoke an ActiveCount API call on a target Cisco Monitoring ISE node:

```
This XML file does not appear to have any style information associated with it. The
document tree is shown below.
-
<sessionCount>
<count>5</count>
</sessionCount>
```

Posture Sessions Counter

You can use the PostureCount API call to retrieve a current count of all currently active Posture sessions. This section provides a schema file output example, a procedure for counting all currently active Posture sessions by invoking the PostureCount API call, and a sample of the Posture sessions data returned after this API call is issued.

PostureCount API Output Schema

This sample schema file is the output of the PostureCount API call for retrieving a count of the current active Posture sessions on the target Cisco Monitoring ISE node:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="sessionCount" type="postureCount"/>

  <xs:complexType name="postureCount">
    <xs:sequence>
      <xs:element name="count" type="xs:int"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

Invoking the PostureCount API Call

Note Make sure that you have verified that the target node to which you are issuing an API call is a valid Cisco Monitoring ISE node. To verify the persona of a Cisco ISE node, see Verifying a Cisco Monitoring ISE Node, page 1-2.

To issue the PostureCount API call, complete the following steps:

Step 1 Log into the target Cisco Monitoring ISE node.

For example, when you initially log into a Cisco Monitoring ISE node with the hostname of acme123, this would display the following URL Address field for this node:

```
https://acme123/admin/LoginAction.do#pageId=com_cisco_xmp_web_page_tmpdash
```

Step 2 Enter the PostureCount API call in the URL Address field of the target node by replacing the "/admin/" component with the API call component (/ise/mnt/api/Session/<specific-api-call>):

```
https://acme123/ise/mnt/api/Session/PostureCount
```

Note You must carefully enter each API call in the URL Address field of a target node because these calls are case-sensitive. The use of "mnt" in the API call convention represents the target Cisco Monitoring ISE node.

Step 3 Press **Enter** to issue the API call.

Sample Data Returned from the PostureCount API Call

The following example illustrates the data returned (number of current active Posture sessions) when you invoke a PostureCount API call on a target Cisco Monitoring ISE node:

```
This XML file does not appear to have any style information associated with it. The
document tree is shown below.

-
<sessionCount>
<count>3</count>
</sessionCount>
```

Profiler Sessions Counter

You can use the ProfilerCount API call to retrieve a count of all currently active Profiler sessions. This section provides a schema file output example, a procedure for counting all currently active Profiler sessions by invoking the ProfilerCount API call, and a sample of the Profiler sessions data returned after this API call is issued.

ProfilerCount API Output Schema

This sample schema file is the output of the ProfilerCount API call for retrieving a count of the current active Profiler sessions on the target Cisco Monitoring ISE node:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="sessionCount" type="profilerCount"/>

  <xs:complexType name="profilerCount">
    <xs:sequence>
      <xs:element name="count" type="xs:int"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

Invoking the ProfilerCount API Call

Note Make sure that you have verified that the target node to which you are issuing an API call is a valid Cisco Monitoring ISE node. To verify the persona of a Cisco ISE node, see Verifying a Cisco Monitoring ISE Node, page 1-2.

To issue the ProfilerCount API call, complete the following steps:

Step 1 Log into the target Cisco Monitoring ISE node.

For example, when you initially log into a Cisco Monitoring ISE node with the hostname of acme123, this would display the following URL Address field for this node:

```
https://acme123/admin/LoginAction.do#pageId=com_cisco_xmp_web_page_tmpdash
```

Step 2 Enter the ProfilerCount API call in the URL Address field of the target node by replacing the "/admin/" component with the API call component (/ise/mnt/api/Session/<specific-api-call>):

```
https://acme123/ise/mnt/api/Session/ProfilerCount
```

Note You must carefully enter each API call in the URL Address field of a target node because these calls are case-sensitive. The use of "mnt" in the API call convention represents a Cisco Monitoring ISE node.

Step 3 Press **Enter** to issue the API call.

Sample Data Returned from the ProfilerCount API Call

The following example illustrates the data returned (number of active Profiler sessions) when you invoke a ProfilerCount API call on a target Cisco Monitoring ISE node:

```
This XML file does not appear to have any style information associated with it. The
document tree is shown below.

-
<sessionCount>
<count>1</count>
</sessionCount>
```

Using the Simple Session List API Calls

The following simple session list API calls let you quickly gather session-related information such as the MAC address, the network access switch (NAS) IP address, user name, and session ID associated with a current active session on a target Cisco Monitoring ISE node in your Cisco ISE deployment:

- Active sessions list (ActiveList)
- Authenticated sessions list (AuthList)

Active Sessions List

You can use the ActiveList API call to list all currently active sessions. This section provides a schema file output example, a procedure for listing all the active sessions by invoking the ActiveList API call, and a sample of the active session-related data returned after this API call is issued.

Note In this release of Cisco ISE, the maximum number of active authenticated endpoint sessions that can be displayed is limited to 100,000.

ActiveList API Output Schema

This sample schema file is the output of the ActiveList API call for retrieving a list of the current active sessions (and session-related information) on the target Cisco Monitoring ISE node:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="activeSessionList" type="simpleActiveSessionList"/>

<xs:complexType name="simpleActiveSessionList">
    <xs:sequence>
      <xs:element name="activeSession" type="simpleActiveSession" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="noOfActiveSession" type="xs:int" use="required"/>
  </xs:complexType>

  <xs:complexType name="simpleActiveSession">
    <xs:sequence>
      <xs:element name="user_name" type="xs:string" minOccurs="0"/>
      <xs:element name="calling_station_id" type="xs:string" minOccurs="0"/>
      <xs:element name="nas_ip_address" type="xs:string" minOccurs="0"/>
      <xs:element name="acct_session_id" type="xs:string" minOccurs="0"/>
      <xs:element name="audit_session_id" type="xs:string" minOccurs="0"/>
      <xs:element name="server" type="xs:string" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

Invoking the ActiveList API Call

Note Make sure that you have verified that the target node to which you are issuing an API call is a valid Cisco Monitoring ISE node. To verify the persona of a Cisco ISE node, see Verifying a Cisco Monitoring ISE Node, page 1-2.

To issue the ActiveList API call, complete the following steps:

Step 1 Log into the target Cisco Monitoring ISE node.

For example, when you initially log into a Cisco Monitoring ISE node with the hostname of acme123, this would display the following URL Address field for this node:

```
https://acme123/admin/LoginAction.do#pageId=com_cisco_xmp_web_page_tmpdash
```

Step 2 Enter the ActiveList API call in the URL Address field of the target node by replacing the “/admin/” component with the API call component (/ise/mnt/api/Session/<specific-api-call>):

```
https://acme123/ise/mnt/api/Session/ActiveList
```

Note You must carefully enter each API call in the URL Address field of a target node, because these calls are case-sensitive. The use of “mnt” in the API call convention represents a Cisco Monitoring ISE node.

Step 3 Press **Enter** to issue the API call.

Sample Data Returned from the ActiveList API Call

The following example illustrates the session-related data returned from the list of active sessions when you invoke an ActiveList API call on a target Cisco Monitoring ISE node:

```
This XML file does not appear to have any style information associated with it. The
document tree is shown below.

-
<activeSessionList noOfActiveSession="5">
-
<activeSession>
<calling_station_id>00:0C:29:FA:EF:0A</calling_station_id>
<server>HAREESH-R6-1-PDP2</server>
</activeSession>
-
<activeSession>
<calling_station_id>70:5A:B6:68:F7:CC</calling_station_id>
<server>HAREESH-R6-1-PDP2</server>
</activeSession>
-
<activeSession>
<user_name>tom_wolfe</user_name>
<calling_station_id>00:14:BF:5A:0C:03</calling_station_id>
<nas_ip_address>10.203.107.161</nas_ip_address>
<acct_session_id>00000032</acct_session_id>
<server>HAREESH-R6-1-PDP2</server>
</activeSession>
-
<activeSession>
<user_name>graham_hancock</user_name>
<calling_station_id>00:50:56:8E:28:BD</calling_station_id>
<nas_ip_address>10.203.107.161</nas_ip_address>
<acct_session_id>0000002C</acct_session_id>
<audit_session_id>0ACB6BA10000002A165FD0C8</audit_session_id>
<server>HAREESH-R6-1-PDP2</server>
</activeSession>
-
<activeSession>
<user_name>ipepvpnuser</user_name>
<calling_station_id>172.23.130.89</calling_station_id>
<nas_ip_address>10.203.107.45</nas_ip_address>
<acct_session_id>A2000070</acct_session_id>
<server>HAREESH-R6-1-PDP2</server>
</activeSession>
</activeSessionList>
```

Authenticated Sessions List

You can use the AuthList API call to retrieve a list of all currently active authenticated sessions. This section provides a schema file output example, a procedure for listing all of the currently active authenticated sessions by invoking the AuthList API call, and a sample of the active authenticated sessions that are returned after this API call is issued.

Note In this release of Cisco ISE, the maximum number of active authenticated endpoint sessions that can be displayed is limited to 100,000.

AuthList API Output Schema

This sample schema file is the output of the AuthList API call for retrieving a list of all currently active authenticated sessions within a specified period of time (or for no specified time using the "null/null" parameter) on the target Cisco Monitoring ISE node:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="activeSessionList" type="simpleActiveSessionList"/>

<xs:complexType name="simpleActiveSessionList">
    <xs:sequence>
      <xs:element name="activeSession" type="simpleActiveSession" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="noOfActiveSession" type="xs:int" use="required"/>
  </xs:complexType>

  <xs:complexType name="simpleActiveSession">
    <xs:sequence>
      <xs:element name="user_name" type="xs:string" minOccurs="0"/>
      <xs:element name="calling_station_id" type="xs:string" minOccurs="0"/>
      <xs:element name="nas_ip_address" type="xs:string" minOccurs="0"/>
      <xs:element name="acct_session_id" type="xs:string" minOccurs="0"/>
      <xs:element name="audit_session_id" type="xs:string" minOccurs="0"/>
      <xs:element name="server" type="xs:string" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

Invoking the AuthList API Call

Note Make sure that you have verified that the target node to which you are issuing an API call is a valid Cisco Monitoring ISE node. To verify the persona of a Cisco ISE node, see Verifying a Cisco Monitoring ISE Node, page 1-2.

To issue the AuthList API call, complete the following steps:

Step 1 Log into the target Cisco Monitoring ISE node.

For example, when you initially log into a Cisco Monitoring ISE node with the hostname of acme123, this would display the following URL Address field for this node:

```
https://acme123/admin/LoginAction.do#pageId=com_cisco_xmp_web_page_tmpdash
```

Step 2 Enter the AuthList API call in the URL Address field of the target node by replacing the "/admin/" component with the API call component (/ise/mnt/api/Session/<specific-api-call>):

Note The first of the following two examples uses a defined starttime and null parameter, which displays a list of the currently active sessions that were authenticated after the specified start time. The second example uses the null/null parameter that displays a list of all currently active authenticated sessions. See Sample Data Returned from the AuthList API Call, page 2-9, which displays samples of the four parameter setting types for this API call.

```
https://acme123/ise/mnt/api/Session/AuthList/2010-12-14 15:33:15/null

https://acme123/ise/mnt/api/Session/AuthList/null/null
```

Note You must carefully enter each API call in the URL Address field of a target node because these calls are case-sensitive. The use of "mnt" in the API call convention represents a Cisco Monitoring ISE node.

Step 3 Press **Enter** to issue the API call.

Sample Data Returned from the AuthList API Call

The following examples illustrate the list of currently active authenticated sessions that is returned when you invoke an AuthList API call on a target Cisco Monitoring ISE node using one of the supported parameter options.

Using the null/null Option

```
This XML file does not appear to have any style information associated with it. The
document tree is shown below.

-
<activeSessionList noOfActiveSession="3">
-
<activeSession>
<user_name>ipepwlcuser</user_name>
<calling_station_id>00:26:82:7B:D2:51</calling_station_id>
<nas_ip_address>10.203.107.10</nas_ip_address>
<audit_session_id>0acb6b0c000000174D07F487</audit_session_id>
<server>HAREESH-R6-1-PDP2</server>
</activeSession>
-
<activeSession>
<user_name>tom_wolfe</user_name>
<calling_station_id>00:50:56:8E:28:BD</calling_station_id>
<nas_ip_address>10.203.107.161</nas_ip_address>
<acct_session_id>00000035</acct_session_id>
```

```
<server>HAREESH-R6-1-PDP2</server>
</activeSession>
-
<activeSession>
<user_name>graham_hancock</user_name>
<calling_station_id>00:14:BF:5A:0C:03</calling_station_id>
<nas_ip_address>10.203.107.161</nas_ip_address>
<acct_session_id>00000033</acct_session_id>
<server>HAREESH-R6-1-PDP2</server>
</activeSession>
</activeSessionList>
```

Using the endtime/null Option

```
This XML file does not appear to have any style information associated with it. The
document tree is shown below.

-
<activeSessionList noOfActiveSession="3">
-
<activeSession>
<user_name>ipepwlcuser</user_name>
<calling_station_id>00:26:82:7B:D2:51</calling_station_id>
<nas_ip_address>10.203.107.10</nas_ip_address>
<audit_session_id>0acb6b0c0000001F4D08085A</audit_session_id>
<server>HAREESH-R6-1-PDP2</server>
</activeSession>
-
<activeSession>
<user_name>hunter_thompson</user_name>
<calling_station_id>00:50:56:8E:28:BD</calling_station_id>
<nas_ip_address>10.203.107.161</nas_ip_address>
<acct_session_id>00000035</acct_session_id>
<server>HAREESH-R6-1-PDP2</server>
</activeSession>
-
<activeSession>
<user_name>bob_ludlum</user_name>
<calling_station_id>00:14:BF:5A:0C:03</calling_station_id>
<nas_ip_address>10.203.107.161</nas_ip_address>
<acct_session_id>00000033</acct_session_id>
<server>HAREESH-R6-1-PDP2</server>
</activeSession>
</activeSessionList>
```

Using the null/starttime Option

```
This XML file does not appear to have any style information associated with it. The
document tree is shown below.

-
<activeSessionList noOfActiveSession="3">
-
<activeSession>
<user_name>ipepwlcuser</user_name>
<calling_station_id>00:26:82:7B:D2:51</calling_station_id>
<nas_ip_address>10.203.107.10</nas_ip_address>
<audit_session_id>0acb6b0c0000001F4D08085A</audit_session_id>
<server>HAREESH-R6-1-PDP2</server>
</activeSession>
-
```

```
<activeSession>
<user_name>bob_ludlum</user_name>
<calling_station_id>00:50:56:8E:28:BD</calling_station_id>
<nas_ip_address>10.203.107.161</nas_ip_address>
<acct_session_id>00000035</acct_session_id>
<server>HAREESH-R6-1-PDP2</server>
</activeSession>
-
<activeSession>
<user_name>tom_wolfe</user_name>
<calling_station_id>00:14:BF:5A:0C:03</calling_station_id>
<nas_ip_address>10.203.107.161</nas_ip_address>
<acct_session_id>00000033</acct_session_id>
<server>HAREESH-R6-1-PDP2</server>
</activeSession>
</activeSessionList>
```

Using the starttime/endtime Option

```
This XML file does not appear to have any style information associated with it. The
document tree is shown below.
-
<activeSessionList noOfActiveSession="3">
-
<activeSession>
<user_name>ipepwlcuser</user_name>
<calling_station_id>00:26:82:7B:D2:51</calling_station_id>
<nas_ip_address>10.203.107.10</nas_ip_address>
<audit_session_id>0acb6b0c0000001F4D08085A</audit_session_id>
<server>HAREESH-R6-1-PDP2</server>
</activeSession>
-
<activeSession>
<user_name>graham_hancock</user_name>
<calling_station_id>00:50:56:8E:28:BD</calling_station_id>
<nas_ip_address>10.203.107.161</nas_ip_address>
<acct_session_id>00000035</acct_session_id>
<server>HAREESH-R6-1-PDP2</server>
</activeSession>
-
<activeSession>
<user_name>hunter_thompson</user_name>
<calling_station_id>00:14:BF:5A:0C:03</calling_station_id>
<nas_ip_address>10.203.107.161</nas_ip_address>
<acct_session_id>00000033</acct_session_id>
<server>HAREESH-R6-1-PDP2</server>
</activeSession>
</activeSessionList>
```

Using the Detailed Session Attribute API Calls

The following detailed session attribute API calls let you quickly search the latest session for key information, such as the following:

- MAC address session search (MACAddress)
- User name session search (UserName)
- NAS IP address session search (IPAddress associated with a target Monitoring ISE node)

MAC Address Session Search

You can use the MACAddress API call to retrieve a specified MAC address from a current, active session. This section provides a schema file output example, a procedure for searching the node database for the latest active session that contains the specified MAC address by invoking the MACAddress API call, and a sample of the MAC address-related data returned after this API call is issued. This API call lists a variety of session-related information drawn from node database tables.

MACAddress API Output Schema

This sample schema file is the output of the MACAddress API call for retrieving a specified MAC address from the current active sessions on the target Cisco Monitoring ISE node:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="sessionParameters" type="restsdStatus"/>

  <xs:complexType name="restsdStatus">
    <xs:sequence>
      <xs:element name="passed" type="xs:anyType" minOccurs="0"/>
      <xs:element name="failed" type="xs:anyType" minOccurs="0"/>
      <xs:element name="user_name" type="xs:string" minOccurs="0"/>
      <xs:element name="nas_ip_address" type="xs:string" minOccurs="0"/>
      <xs:element name="failure_reason" type="xs:string" minOccurs="0"/>
      <xs:element name="calling_station_id" type="xs:string" minOccurs="0"/>
      <xs:element name="nas_port" type="xs:string" minOccurs="0"/>
      <xs:element name="identity_group" type="xs:string" minOccurs="0"/>
      <xs:element name="network_device_name" type="xs:string" minOccurs="0"/>
      <xs:element name="acs_server" type="xs:string" minOccurs="0"/>
      <xs:element name="authen_protocol" type="xs:string" minOccurs="0"/>
      <xs:element name="framed_ip_address" type="xs:string" minOccurs="0"/>
      <xs:element name="network_device_groups" type="xs:string" minOccurs="0"/>
      <xs:element name="access_service" type="xs:string" minOccurs="0"/>
      <xs:element name="auth_acs_timestamp" type="xs:dateTime" minOccurs="0"/>
      <xs:element name="authentication_method" type="xs:string" minOccurs="0"/>
      <xs:element name="execution_steps" type="xs:string" minOccurs="0"/>
      <xs:element name="radius_response" type="xs:string" minOccurs="0"/>
      <xs:element name="audit_session_id" type="xs:string" minOccurs="0"/>
      <xs:element name="nas_identifier" type="xs:string" minOccurs="0"/>
      <xs:element name="nas_port_id" type="xs:string" minOccurs="0"/>
      <xs:element name="nac_policy_compliance" type="xs:string" minOccurs="0"/>
      <xs:element name="auth_id" type="xs:long" minOccurs="0"/>
      <xs:element name="auth_acsview_timestamp" type="xs:dateTime" minOccurs="0"/>
      <xs:element name="message_code" type="xs:string" minOccurs="0"/>
      <xs:element name="acs_session_id" type="xs:string" minOccurs="0"/>
      <xs:element name="service_selection_policy" type="xs:string" minOccurs="0"/>
      <xs:element name="authorization_policy" type="xs:string" minOccurs="0"/>
      <xs:element name="identity_store" type="xs:string" minOccurs="0"/>
      <xs:element name="response" type="xs:string" minOccurs="0"/>
      <xs:element name="service_type" type="xs:string" minOccurs="0"/>
      <xs:element name="cts_security_group" type="xs:string" minOccurs="0"/>
      <xs:element name="use_case" type="xs:string" minOccurs="0"/>
      <xs:element name="cisco_av_pair" type="xs:string" minOccurs="0"/>
      <xs:element name="ad_domain" type="xs:string" minOccurs="0"/>
      <xs:element name="acs_username" type="xs:string" minOccurs="0"/>
      <xs:element name="radius_username" type="xs:string" minOccurs="0"/>
      <xs:element name="nac_role" type="xs:string" minOccurs="0"/>
      <xs:element name="nac_username" type="xs:string" minOccurs="0"/>
      <xs:element name="nac_posture_token" type="xs:string" minOccurs="0"/>
      <xs:element name="nac_radius_is_user_auth" type="xs:string" minOccurs="0"/>
```

```
        <xs:element name="selected_posture_server" type="xs:string" minOccurs="0"/>
        <xs:element name="selected_identity_store" type="xs:string" minOccurs="0"/>
        <xs:element name="authentication_identity_store" type="xs:string" minOccurs="0"/>
        <xs:element name="azn_exp_pol_matched_rule" type="xs:string" minOccurs="0"/>
        <xs:element name="ext_pol_server_matched_rule" type="xs:string" minOccurs="0"/>
        <xs:element name="grp_mapping_pol_matched_rule" type="xs:string" minOccurs="0"/>
        <xs:element name="identity_policy_matched_rule" type="xs:string" minOccurs="0"/>
        <xs:element name="nas_port_type" type="xs:string" minOccurs="0"/>
        <xs:element name="query_identity_stores" type="xs:string" minOccurs="0"/>
        <xs:element name="selected_azn_profiles" type="xs:string" minOccurs="0"/>
        <xs:element name="sel_exp_azn_profiles" type="xs:string" minOccurs="0"/>
        <xs:element name="selected_query_identity_stores" type="xs:string" minOccurs="0"/>
        <xs:element name="eap_tunnel" type="xs:string" minOccurs="0"/>
        <xs:element name="tunnel_details" type="xs:string" minOccurs="0"/>
        <xs:element name="cisco_h323_attributes" type="xs:string" minOccurs="0"/>
        <xs:element name="cisco_ssg_attributes" type="xs:string" minOccurs="0"/>
        <xs:element name="other_attributes" type="xs:string" minOccurs="0"/>
        <xs:element name="response_time" type="xs:long" minOccurs="0"/>
        <xs:element name="nad_failure" type="xs:anyType" minOccurs="0"/>
        <xs:element name="destination_ip_address" type="xs:string" minOccurs="0"/>
        <xs:element name="acct_id" type="xs:long" minOccurs="0"/>
        <xs:element name="acct_acs_timestamp" type="xs:dateTime" minOccurs="0"/>
        <xs:element name="acct_acsview_timestamp" type="xs:dateTime" minOccurs="0"/>
        <xs:element name="acct_session_id" type="xs:string" minOccurs="0"/>
        <xs:element name="acct_status_type" type="xs:string" minOccurs="0"/>
        <xs:element name="acct_session_time" type="xs:long" minOccurs="0"/>
        <xs:element name="acct_input_octets" type="xs:string" minOccurs="0"/>
        <xs:element name="acct_output_octets" type="xs:string" minOccurs="0"/>
        <xs:element name="acct_input_packets" type="xs:long" minOccurs="0"/>
        <xs:element name="acct_output_packets" type="xs:long" minOccurs="0"/>
        <xs:element name="acct_class" type="xs:string" minOccurs="0"/>
        <xs:element name="acct_terminate_cause" type="xs:string" minOccurs="0"/>
        <xs:element name="acct_multi_session_id" type="xs:string" minOccurs="0"/>
        <xs:element name="acct_authentic" type="xs:string" minOccurs="0"/>
        <xs:element name="termination_action" type="xs:string" minOccurs="0"/>
        <xs:element name="session_timeout" type="xs:string" minOccurs="0"/>
        <xs:element name="idle_timeout" type="xs:string" minOccurs="0"/>
        <xs:element name="acct_interim_interval" type="xs:string" minOccurs="0"/>
        <xs:element name="acct_delay_time" type="xs:string" minOccurs="0"/>
        <xs:element name="event_timestamp" type="xs:string" minOccurs="0"/>
        <xs:element name="acct_tunnel_connection" type="xs:string" minOccurs="0"/>
        <xs:element name="acct_tunnel_packet_lost" type="xs:string" minOccurs="0"/>
        <xs:element name="security_group" type="xs:string" minOccurs="0"/>
        <xs:element name="cisco_h323_setup_time" type="xs:dateTime" minOccurs="0"/>
        <xs:element name="cisco_h323_connect_time" type="xs:dateTime" minOccurs="0"/>
        <xs:element name="cisco_h323_disconnect_time" type="xs:dateTime" minOccurs="0"/>
        <xs:element name="framed_protocol" type="xs:string" minOccurs="0"/>
        <xs:element name="started" type="xs:anyType" minOccurs="0"/>
        <xs:element name="stopped" type="xs:anyType" minOccurs="0"/>
        <xs:element name="ckpt_id" type="xs:long" minOccurs="0"/>
        <xs:element name="type" type="xs:long" minOccurs="0"/>
        <xs:element name="nad_acsview_timestamp" type="xs:dateTime" minOccurs="0"/>
        <xs:element name="vlan" type="xs:string" minOccurs="0"/>
        <xs:element name="dacl" type="xs:string" minOccurs="0"/>
        <xs:element name="authentication_type" type="xs:string" minOccurs="0"/>
        <xs:element name="interface_name" type="xs:string" minOccurs="0"/>
        <xs:element name="reason" type="xs:string" minOccurs="0"/>
        <xs:element name="endpoint_policy" type="xs:string" minOccurs="0"/>
      </xs:sequence>
    </xs:complexType>
  </xs:schema>
```

Invoking the MACAddress API Call

Note Make sure that you have verified that the target node to which you are issuing an API call is a valid Cisco Monitoring ISE node. To verify the persona of a Cisco ISE node, see Verifying a Cisco Monitoring ISE Node, page 1-2.

To issue the MACAddress API call, complete the following steps:

Step 1 Log into the target Cisco Monitoring ISE node.

For example, when you initially log into a Cisco Monitoring ISE node with the hostname of acme123, this would display the following URL Address field for this node:

```
https://acme123/admin/LoginAction.do#pageId=com_cisco_xmp_web_page_tmpdash
```

Step 2 Enter the MACAddress API call in the URL Address field of the target node by replacing the "/admin/" component with the API call component (/ise/mnt/api/<specific-api-call>/<macaddress>):

```
https://acme123/ise/mnt/api/Session/MACAddress/0A:0B:0C:0D:0E:0F
```

Note Make sure that you specify the MAC address using the XX:XX:XX:XX:XX:XX format.

Note You must carefully enter each API call in the URL Address field of a target node because these calls are case-sensitive. The use of "mnt" in the API call convention represents a Cisco Monitoring ISE node.

Step 3 Press **Enter** to issue the API call.

Sample Data Returned from the MACAddress API Call

The following example illustrates the session-related data returned from the list of active sessions when you invoke an ActiveList API call on a target Cisco Monitoring ISE node:

```
This XML file does not appear to have any style information associated with it. The
document tree is shown below.

-
<sessionParameters>
<passed xsi:type="xs:boolean">true</passed>
<failed xsi:type="xs:boolean">false</failed>
<user_name>hunter_thompson</user_name>
<nas_ip_address>10.203.107.161</nas_ip_address>
<calling_station_id>00:14:BF:5A:0C:03</calling_station_id>
<nas_port>50115</nas_port>
<identity_group>Profiled</identity_group>
<network_device_name>Core-Switch</network_device_name>
<acs_server>HAREESH-R6-1-PDP2</acs_server>
<authen_protocol>Lookup</authen_protocol>
-
```

```
<network_device_groups>
Device Type#All Device Types,Location#All Locations
</network_device_groups>
<access_service>RADIUS</access_service>
<auth_acs_timestamp>2010-12-15T02:11:12.359Z</auth_acs_timestamp>
<authentication_method>mab</authentication_method>
-
<execution_steps>
11001,11017,11027,15008,15048,15004,15041,15004,15013,24209,24211,22037,15036,15048,15048,
15004,15016,11022,11002
</execution_steps>
<audit_session_id>0ACB6BA1000000351BBFBF8B</audit_session_id>
<nas_port_id>GigabitEthernet1/0/15</nas_port_id>
<nac_policy_compliance>Pending</nac_policy_compliance>
<auth_id>1291240762077361</auth_id>
<auth_acsview_timestamp>2010-12-15T02:11:12.360Z</auth_acsview_timestamp>
<message_code>5200</message_code>
<acs_session_id>HAREESH-R6-1-PDP2/81148292/681</acs_session_id>
<service_selection_policy>MAB</service_selection_policy>
<identity_store>Internal Hosts</identity_store>
-
<response>
{UserName=00-14-BF-5A-0C-03; User-Name=00-14-BF-5A-0C-03;
State=ReauthSession:0ACB6BA1000000351BBFBF8B;
Class=CACS:0ACB6BA1000000351BBFBF8B:HAREESH-R6-1-PDP2/81148292/681;
Termination-Action=RADIUS-Request; cisco-av-pair=url-redirect-acl=ACL-WEBAUTH-REDIRECT;
cisco-av-pair=url-redirect=https://HAREESH-R6-1-PDP2.cisco.com:8443/guestportal/gateway?se
ssionId=0ACB6BA1000000351BBFBF8B&action=cwa;
cisco-av-pair=ACS:CiscoSecure-Defined-ACL=#ACSACL#-IP-ACL-DENY-4ced8390; }
</response>
<service_type>Call Check</service_type>
<use_case>Host Lookup</use_case>
<cisco_av_pair>audit-session-id=0ACB6BA1000000351BBFBF8B</cisco_av_pair>
<acs_username>00:14:BF:5A:0C:03</acs_username>
<radius_username>00:14:BF:5A:0C:03</radius_username>
<selected_identity_store>Internal Hosts</selected_identity_store>
<authentication_identity_store>Internal Hosts</authentication_identity_store>
<identity_policy_matched_rule>Default</identity_policy_matched_rule>
<nas_port_type>Ethernet</nas_port_type>
<selected_azn_profiles>CWA</selected_azn_profiles>
-
<other_attributes>
ConfigVersionId=44,DestinationIPAddress=10.203.107.162,DestinationPort=1812,Protocol=Radiu
s,Framed-MTU=1500,EAP-Key-Name=,CPMSessionID=0ACB6BA1000000351BBFBF8B,CPMSessionID=0ACB6BA
1000000351BBFBF8B,EndPointMACAddress=00-14-BF-5A-0C-03,HostIdentityGroup=Endpoint Identity
Groups:Profiled,Device Type=Device Type#All Device Types,Location=Location#All
Locations,Model Name=Unknown,Software Version=Unknown,Device IP
Address=10.203.107.161,Called-Station-ID=04:FE:7F:7F:C0:8F
</other_attributes>
<response_time>77</response_time>
<acct_id>1291240762077386</acct_id>
<acct_acs_timestamp>2010-12-15T02:12:30.779Z</acct_acs_timestamp>
<acct_acsview_timestamp>2010-12-15T02:12:30.780Z</acct_acsview_timestamp>
<acct_session_id>00000038</acct_session_id>
<acct_status_type>Interim-Update</acct_status_type>
<acct_session_time>78</acct_session_time>
<acct_input_octets>13742</acct_input_octets>
<acct_output_octets>6277</acct_output_octets>
<acct_input_packets>108</acct_input_packets>
<acct_output_packets>66</acct_output_packets>
-
```

```
<acct_class>
CACS:0ACB6BA1000000351BBFBF8B:HAREESH-R6-1-PDP2/81148292/681
</acct_class>
<acct_delay_time>0</acct_delay_time>
<started xsi:type="xs:boolean">false</started>
<stopped xsi:type="xs:boolean">false</stopped>
</sessionParameters>
```

User Name Session Search

You can use the UserName API call to retrieve a specified user name from a current, active session. This section provides a schema file output example, a procedure for searching the node database for the latest active session that contains the specified user name by invoking the UserName API call, and a sample of the user name-related data returned after this API call is issued. This API will list a variety of session-related information drawn from node database tables.

UserName API Output Schema

This sample schema file is the output of the UserName API call for retrieving a specified user name from the current active sessions on the target Cisco Monitoring ISE node:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="sessionParameters" type="restsdStatus"/>

  <xs:complexType name="restsdStatus">
    <xs:sequence>
      <xs:element name="passed" type="xs:anyType" minOccurs="0"/>
      <xs:element name="failed" type="xs:anyType" minOccurs="0"/>
      <xs:element name="user_name" type="xs:string" minOccurs="0"/>
      <xs:element name="nas_ip_address" type="xs:string" minOccurs="0"/>
      <xs:element name="failure_reason" type="xs:string" minOccurs="0"/>
      <xs:element name="calling_station_id" type="xs:string" minOccurs="0"/>
      <xs:element name="nas_port" type="xs:string" minOccurs="0"/>
      <xs:element name="identity_group" type="xs:string" minOccurs="0"/>
      <xs:element name="network_device_name" type="xs:string" minOccurs="0"/>
      <xs:element name="acs_server" type="xs:string" minOccurs="0"/>
      <xs:element name="authen_protocol" type="xs:string" minOccurs="0"/>
      <xs:element name="framed_ip_address" type="xs:string" minOccurs="0"/>
      <xs:element name="network_device_groups" type="xs:string" minOccurs="0"/>
      <xs:element name="access_service" type="xs:string" minOccurs="0"/>
      <xs:element name="auth_acs_timestamp" type="xs:dateTime" minOccurs="0"/>
      <xs:element name="authentication_method" type="xs:string" minOccurs="0"/>
      <xs:element name="execution_steps" type="xs:string" minOccurs="0"/>
      <xs:element name="radius_response" type="xs:string" minOccurs="0"/>
      <xs:element name="audit_session_id" type="xs:string" minOccurs="0"/>
      <xs:element name="nas_identifier" type="xs:string" minOccurs="0"/>
      <xs:element name="nas_port_id" type="xs:string" minOccurs="0"/>
      <xs:element name="nac_policy_compliance" type="xs:string" minOccurs="0"/>
      <xs:element name="auth_id" type="xs:long" minOccurs="0"/>
      <xs:element name="auth_acsview_timestamp" type="xs:dateTime" minOccurs="0"/>
      <xs:element name="message_code" type="xs:string" minOccurs="0"/>
      <xs:element name="acs_session_id" type="xs:string" minOccurs="0"/>
      <xs:element name="service_selection_policy" type="xs:string" minOccurs="0"/>
      <xs:element name="authorization_policy" type="xs:string" minOccurs="0"/>
      <xs:element name="identity_store" type="xs:string" minOccurs="0"/>
      <xs:element name="response" type="xs:string" minOccurs="0"/>
      <xs:element name="service_type" type="xs:string" minOccurs="0"/>
```

```
<xs:element name="cts_security_group" type="xs:string" minOccurs="0"/>
<xs:element name="use_case" type="xs:string" minOccurs="0"/>
<xs:element name="cisco_av_pair" type="xs:string" minOccurs="0"/>
<xs:element name="ad_domain" type="xs:string" minOccurs="0"/>
<xs:element name="acs_username" type="xs:string" minOccurs="0"/>
<xs:element name="radius_username" type="xs:string" minOccurs="0"/>
<xs:element name="nac_role" type="xs:string" minOccurs="0"/>
<xs:element name="nac_username" type="xs:string" minOccurs="0"/>
<xs:element name="nac_posture_token" type="xs:string" minOccurs="0"/>
<xs:element name="nac_radius_is_user_auth" type="xs:string" minOccurs="0"/>
<xs:element name="selected_posture_server" type="xs:string" minOccurs="0"/>
<xs:element name="selected_identity_store" type="xs:string" minOccurs="0"/>
<xs:element name="authentication_identity_store" type="xs:string" minOccurs="0"/>
<xs:element name="azn_exp_pol_matched_rule" type="xs:string" minOccurs="0"/>
<xs:element name="ext_pol_server_matched_rule" type="xs:string" minOccurs="0"/>
<xs:element name="grp_mapping_pol_matched_rule" type="xs:string" minOccurs="0"/>
<xs:element name="identity_policy_matched_rule" type="xs:string" minOccurs="0"/>
<xs:element name="nas_port_type" type="xs:string" minOccurs="0"/>
<xs:element name="query_identity_stores" type="xs:string" minOccurs="0"/>
<xs:element name="selected_azn_profiles" type="xs:string" minOccurs="0"/>
<xs:element name="sel_exp_azn_profiles" type="xs:string" minOccurs="0"/>
<xs:element name="selected_query_identity_stores" type="xs:string" minOccurs="0"/>
<xs:element name="eap_tunnel" type="xs:string" minOccurs="0"/>
<xs:element name="tunnel_details" type="xs:string" minOccurs="0"/>
<xs:element name="cisco_h323_attributes" type="xs:string" minOccurs="0"/>
<xs:element name="cisco_ssg_attributes" type="xs:string" minOccurs="0"/>
<xs:element name="other_attributes" type="xs:string" minOccurs="0"/>
<xs:element name="response_time" type="xs:long" minOccurs="0"/>
<xs:element name="nad_failure" type="xs:anyType" minOccurs="0"/>
<xs:element name="destination_ip_address" type="xs:string" minOccurs="0"/>
<xs:element name="acct_id" type="xs:long" minOccurs="0"/>
<xs:element name="acct_acs_timestamp" type="xs:dateTime" minOccurs="0"/>
<xs:element name="acct_acsview_timestamp" type="xs:dateTime" minOccurs="0"/>
<xs:element name="acct_session_id" type="xs:string" minOccurs="0"/>
<xs:element name="acct_status_type" type="xs:string" minOccurs="0"/>
<xs:element name="acct_session_time" type="xs:long" minOccurs="0"/>
<xs:element name="acct_input_octets" type="xs:string" minOccurs="0"/>
<xs:element name="acct_output_octets" type="xs:string" minOccurs="0"/>
<xs:element name="acct_input_packets" type="xs:long" minOccurs="0"/>
<xs:element name="acct_output_packets" type="xs:long" minOccurs="0"/>
<xs:element name="acct_class" type="xs:string" minOccurs="0"/>
<xs:element name="acct_terminate_cause" type="xs:string" minOccurs="0"/>
<xs:element name="acct_multi_session_id" type="xs:string" minOccurs="0"/>
<xs:element name="acct_authentic" type="xs:string" minOccurs="0"/>
<xs:element name="termination_action" type="xs:string" minOccurs="0"/>
<xs:element name="session_timeout" type="xs:string" minOccurs="0"/>
<xs:element name="idle_timeout" type="xs:string" minOccurs="0"/>
<xs:element name="acct_interim_interval" type="xs:string" minOccurs="0"/>
<xs:element name="acct_delay_time" type="xs:string" minOccurs="0"/>
<xs:element name="event_timestamp" type="xs:string" minOccurs="0"/>
<xs:element name="acct_tunnel_connection" type="xs:string" minOccurs="0"/>
<xs:element name="acct_tunnel_packet_lost" type="xs:string" minOccurs="0"/>
<xs:element name="security_group" type="xs:string" minOccurs="0"/>
<xs:element name="cisco_h323_setup_time" type="xs:dateTime" minOccurs="0"/>
<xs:element name="cisco_h323_connect_time" type="xs:dateTime" minOccurs="0"/>
<xs:element name="cisco_h323_disconnect_time" type="xs:dateTime" minOccurs="0"/>
<xs:element name="framed_protocol" type="xs:string" minOccurs="0"/>
<xs:element name="started" type="xs:anyType" minOccurs="0"/>
<xs:element name="stopped" type="xs:anyType" minOccurs="0"/>
<xs:element name="ckpt_id" type="xs:long" minOccurs="0"/>
<xs:element name="type" type="xs:long" minOccurs="0"/>
<xs:element name="nad_acsview_timestamp" type="xs:dateTime" minOccurs="0"/>
<xs:element name="vlan" type="xs:string" minOccurs="0"/>
<xs:element name="dacl" type="xs:string" minOccurs="0"/>
```

```
      <xs:element name="authentication_type" type="xs:string" minOccurs="0"/>
      <xs:element name="interface_name" type="xs:string" minOccurs="0"/>
      <xs:element name="reason" type="xs:string" minOccurs="0"/>
      <xs:element name="endpoint_policy" type="xs:string" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

Invoking the UserName API Call

Note Make sure that you have verified that the target node to which you are issuing an API call is a valid Cisco Monitoring ISE node. To verify the persona of a Cisco ISE node, see Verifying a Cisco Monitoring ISE Node, page 1-2.

To issue the UserName API call, complete the following steps:

Step 1 Log into the target Cisco Monitoring ISE node.

For example, when you initially log into a Cisco Monitoring ISE node with the hostname of acme123, this would display the following URL Address field for this node:

```
https://acme123/admin/LoginAction.do#pageId=com_cisco_xmp_web_page_tmpdash
```

Step 2 Enter the UserName API call in the URL Address field of the target node by replacing the "/admin/" component with the API call component (/ise/mnt/api/<specific-api-call>/<username>):

```
https://acme123/ise/mnt/api/Session/UserName/graham_hancock
```

Note You must carefully enter each API call in the URL Address field of a target node because these calls are case-sensitive. The use of "mnt" in the API call convention represents a Cisco Monitoring ISE node.

Step 3 Press **Enter** to issue the API call.

Sample Data Returned from the UserName API Call

The following example illustrates the session-related data returned from the list of active sessions when you invoke a UserName API call on a target Cisco Monitoring ISE node:

```
This XML file does not appear to have any style information associated with it. The
document tree is shown below.

-
<sessionParameters>
<passed xsi:type="xs:boolean">true</passed>
<failed xsi:type="xs:boolean">false</failed>
<user_name>graham_hancock</user_name>
<nas_ip_address>10.203.107.161</nas_ip_address>
<calling_station_id>00:14:BF:5A:0C:03</calling_station_id>
<nas_port>50115</nas_port>
<identity_group>Profiled</identity_group>
<network_device_name>Core-Switch</network_device_name>
<acs_server>HAREESH-R6-1-PDP2</acs_server>
<authen_protocol>Lookup</authen_protocol>
```

-
<network_device_groups>
Device Type#All Device Types,Location#All Locations
</network_device_groups>
<access_service>RADIUS</access_service>
<auth_acs_timestamp>2010-12-15T02:11:12.359Z</auth_acs_timestamp>
<authentication_method>mab</authentication_method>
-
<execution_steps>
11001,11017,11027,15008,15048,15004,15041,15004,15013,24209,24211,22037,15036,15048,15048,
15004,15016,11022,11002
</execution_steps>
<audit_session_id>0ACB6BA1000000351BBFBF8B</audit_session_id>
<nas_port_id>GigabitEthernet1/0/15</nas_port_id>
<nac_policy_compliance>Pending</nac_policy_compliance>
<auth_id>1291240762077361</auth_id>
<auth_acsview_timestamp>2010-12-15T02:11:12.360Z</auth_acsview_timestamp>
<message_code>5200</message_code>
<acs_session_id>HAREESH-R6-1-PDP2/81148292/681</acs_session_id>
<service_selection_policy>MAB</service_selection_policy>
<identity_store>Internal Hosts</identity_store>
-
<response>
{UserName=graham_hancock; User-Name=graham_hancock;
State=ReauthSession:0ACB6BA1000000351BBFBF8B;
Class=CACS:0ACB6BA1000000351BBFBF8B:HAREESH-R6-1-PDP2/81148292/681;
Termination-Action=RADIUS-Request; cisco-av-pair=url-redirect-acl=ACL-WEBAUTH-REDIRECT;
cisco-av-pair=url-redirect=https://HAREESH-R6-1-PDP2.cisco.com:8443/guestportal/gateway?se
ssionId=0ACB6BA1000000351BBFBF8B&action=cwa;
cisco-av-pair=ACS:CiscoSecure-Defined-ACL=#ACSACL#-IP-ACL-DENY-4ced8390; }
</response>
<service_type>Call Check</service_type>
<use_case>Host Lookup</use_case>
<cisco_av_pair>audit-session-id=0ACB6BA1000000351BBFBF8B</cisco_av_pair>
<acs_username>graham_hancock</acs_username>
<radius_username>00:14:BF:5A:0C:03</radius_username>
<selected_identity_store>Internal Hosts</selected_identity_store>
<authentication_identity_store>Internal Hosts</authentication_identity_store>
<identity_policy_matched_rule>Default</identity_policy_matched_rule>
<nas_port_type>Ethernet</nas_port_type>
<selected_azn_profiles>CWA</selected_azn_profiles>
-
<other_attributes>
ConfigVersionId=44,DestinationIPAddress=10.203.107.162,DestinationPort=1812,Protocol=Radiu
s,Framed-MTU=1500,EAP-Key-Name=,CPMSessionID=0ACB6BA1000000351BBFBF8B,CPMSessionID=0ACB6BA
1000000351BBFBF8B,EndPointMACAddress=00-14-BF-5A-0C-03,HostIdentityGroup=Endpoint Identity
Groups:Profiled,Device Type=Device Type#All Device Types,Location=Location#All
Locations,Model Name=Unknown,Software Version=Unknown,Device IP
Address=10.203.107.161,Called-Station-ID=04:FE:7F:7F:C0:8F
</other_attributes>
<response_time>77</response_time>
<acct_id>1291240762077386</acct_id>
<acct_acs_timestamp>2010-12-15T02:12:30.779Z</acct_acs_timestamp>
<acct_acsview_timestamp>2010-12-15T02:12:30.780Z</acct_acsview_timestamp>
<acct_session_id>00000038</acct_session_id>
<acct_status_type>Interim-Update</acct_status_type>
<acct_session_time>78</acct_session_time>
<acct_input_octets>13742</acct_input_octets>
<acct_output_octets>6277</acct_output_octets>
<acct_input_packets>108</acct_input_packets>
<acct_output_packets>66</acct_output_packets>
-

```
<acct_class>
CACS:0ACB6BA1000000351BBFBF8B:HAREESH-R6-1-PDP2/81148292/681
</acct_class>
<acct_delay_time>0</acct_delay_time>
<started xsi:type="xs:boolean">false</started>
<stopped xsi:type="xs:boolean">false</stopped>
</sessionParameters>
```

NAS IP Address Session Search

You can use the IPAddress API call to retrieve a specified NAS IP address from a current session. This section provides a schema file output example, a procedure for searching the node database for the latest active session that contains the specified NAS IP address by invoking the IPAddress API call, and a sample of the NAS IP address-related data returned after this API call is issued. This API will list a variety of session-related information drawn from node database tables.

IPAddress API Output Schema

This sample schema file is the output of the IPAddress API call for retrieving a specified NAS IP address from the current active sessions on the target Cisco Monitoring ISE node:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="sessionParameters" type="restsdStatus"/>

  <xs:complexType name="restsdStatus">
    <xs:sequence>
      <xs:element name="passed" type="xs:anyType" minOccurs="0"/>
      <xs:element name="failed" type="xs:anyType" minOccurs="0"/>
      <xs:element name="user_name" type="xs:string" minOccurs="0"/>
      <xs:element name="nas_ip_address" type="xs:string" minOccurs="0"/>
      <xs:element name="failure_reason" type="xs:string" minOccurs="0"/>
      <xs:element name="calling_station_id" type="xs:string" minOccurs="0"/>
      <xs:element name="nas_port" type="xs:string" minOccurs="0"/>
      <xs:element name="identity_group" type="xs:string" minOccurs="0"/>
      <xs:element name="network_device_name" type="xs:string" minOccurs="0"/>
      <xs:element name="acs_server" type="xs:string" minOccurs="0"/>
      <xs:element name="authen_protocol" type="xs:string" minOccurs="0"/>
      <xs:element name="framed_ip_address" type="xs:string" minOccurs="0"/>
      <xs:element name="network_device_groups" type="xs:string" minOccurs="0"/>
      <xs:element name="access_service" type="xs:string" minOccurs="0"/>
      <xs:element name="auth_acs_timestamp" type="xs:dateTime" minOccurs="0"/>
      <xs:element name="authentication_method" type="xs:string" minOccurs="0"/>
      <xs:element name="execution_steps" type="xs:string" minOccurs="0"/>
      <xs:element name="radius_response" type="xs:string" minOccurs="0"/>
      <xs:element name="audit_session_id" type="xs:string" minOccurs="0"/>
      <xs:element name="nas_identifier" type="xs:string" minOccurs="0"/>
      <xs:element name="nas_port_id" type="xs:string" minOccurs="0"/>
      <xs:element name="nac_policy_compliance" type="xs:string" minOccurs="0"/>
      <xs:element name="auth_id" type="xs:long" minOccurs="0"/>
      <xs:element name="auth_acsview_timestamp" type="xs:dateTime" minOccurs="0"/>
      <xs:element name="message_code" type="xs:string" minOccurs="0"/>
      <xs:element name="acs_session_id" type="xs:string" minOccurs="0"/>
      <xs:element name="service_selection_policy" type="xs:string" minOccurs="0"/>
      <xs:element name="authorization_policy" type="xs:string" minOccurs="0"/>
      <xs:element name="identity_store" type="xs:string" minOccurs="0"/>
      <xs:element name="response" type="xs:string" minOccurs="0"/>
      <xs:element name="service_type" type="xs:string" minOccurs="0"/>
```

```
<xs:element name="cts_security_group" type="xs:string" minOccurs="0"/>
<xs:element name="use_case" type="xs:string" minOccurs="0"/>
<xs:element name="cisco_av_pair" type="xs:string" minOccurs="0"/>
<xs:element name="ad_domain" type="xs:string" minOccurs="0"/>
<xs:element name="acs_username" type="xs:string" minOccurs="0"/>
<xs:element name="radius_username" type="xs:string" minOccurs="0"/>
<xs:element name="nac_role" type="xs:string" minOccurs="0"/>
<xs:element name="nac_username" type="xs:string" minOccurs="0"/>
<xs:element name="nac_posture_token" type="xs:string" minOccurs="0"/>
<xs:element name="nac_radius_is_user_auth" type="xs:string" minOccurs="0"/>
<xs:element name="selected_posture_server" type="xs:string" minOccurs="0"/>
<xs:element name="selected_identity_store" type="xs:string" minOccurs="0"/>
<xs:element name="authentication_identity_store" type="xs:string" minOccurs="0"/>
<xs:element name="azn_exp_pol_matched_rule" type="xs:string" minOccurs="0"/>
<xs:element name="ext_pol_server_matched_rule" type="xs:string" minOccurs="0"/>
<xs:element name="grp_mapping_pol_matched_rule" type="xs:string" minOccurs="0"/>
<xs:element name="identity_policy_matched_rule" type="xs:string" minOccurs="0"/>
<xs:element name="nas_port_type" type="xs:string" minOccurs="0"/>
<xs:element name="query_identity_stores" type="xs:string" minOccurs="0"/>
<xs:element name="selected_azn_profiles" type="xs:string" minOccurs="0"/>
<xs:element name="sel_exp_azn_profiles" type="xs:string" minOccurs="0"/>
<xs:element name="selected_query_identity_stores" type="xs:string" minOccurs="0"/>
<xs:element name="eap_tunnel" type="xs:string" minOccurs="0"/>
<xs:element name="tunnel_details" type="xs:string" minOccurs="0"/>
<xs:element name="cisco_h323_attributes" type="xs:string" minOccurs="0"/>
<xs:element name="cisco_ssg_attributes" type="xs:string" minOccurs="0"/>
<xs:element name="other_attributes" type="xs:string" minOccurs="0"/>
<xs:element name="response_time" type="xs:long" minOccurs="0"/>
<xs:element name="nad_failure" type="xs:anyType" minOccurs="0"/>
<xs:element name="destination_ip_address" type="xs:string" minOccurs="0"/>
<xs:element name="acct_id" type="xs:long" minOccurs="0"/>
<xs:element name="acct_acs_timestamp" type="xs:dateTime" minOccurs="0"/>
<xs:element name="acct_acsview_timestamp" type="xs:dateTime" minOccurs="0"/>
<xs:element name="acct_session_id" type="xs:string" minOccurs="0"/>
<xs:element name="acct_status_type" type="xs:string" minOccurs="0"/>
<xs:element name="acct_session_time" type="xs:long" minOccurs="0"/>
<xs:element name="acct_input_octets" type="xs:string" minOccurs="0"/>
<xs:element name="acct_output_octets" type="xs:string" minOccurs="0"/>
<xs:element name="acct_input_packets" type="xs:long" minOccurs="0"/>
<xs:element name="acct_output_packets" type="xs:long" minOccurs="0"/>
<xs:element name="acct_class" type="xs:string" minOccurs="0"/>
<xs:element name="acct_terminate_cause" type="xs:string" minOccurs="0"/>
<xs:element name="acct_multi_session_id" type="xs:string" minOccurs="0"/>
<xs:element name="acct_authentic" type="xs:string" minOccurs="0"/>
<xs:element name="termination_action" type="xs:string" minOccurs="0"/>
<xs:element name="session_timeout" type="xs:string" minOccurs="0"/>
<xs:element name="idle_timeout" type="xs:string" minOccurs="0"/>
<xs:element name="acct_interim_interval" type="xs:string" minOccurs="0"/>
<xs:element name="acct_delay_time" type="xs:string" minOccurs="0"/>
<xs:element name="event_timestamp" type="xs:string" minOccurs="0"/>
<xs:element name="acct_tunnel_connection" type="xs:string" minOccurs="0"/>
<xs:element name="acct_tunnel_packet_lost" type="xs:string" minOccurs="0"/>
<xs:element name="security_group" type="xs:string" minOccurs="0"/>
<xs:element name="cisco_h323_setup_time" type="xs:dateTime" minOccurs="0"/>
<xs:element name="cisco_h323_connect_time" type="xs:dateTime" minOccurs="0"/>
<xs:element name="cisco_h323_disconnect_time" type="xs:dateTime" minOccurs="0"/>
<xs:element name="framed_protocol" type="xs:string" minOccurs="0"/>
<xs:element name="started" type="xs:anyType" minOccurs="0"/>
<xs:element name="stopped" type="xs:anyType" minOccurs="0"/>
<xs:element name="ckpt_id" type="xs:long" minOccurs="0"/>
<xs:element name="type" type="xs:long" minOccurs="0"/>
<xs:element name="nad_acsview_timestamp" type="xs:dateTime" minOccurs="0"/>
<xs:element name="vlan" type="xs:string" minOccurs="0"/>
<xs:element name="dacl" type="xs:string" minOccurs="0"/>
```

```
      <xs:element name="authentication_type" type="xs:string" minOccurs="0"/>
      <xs:element name="interface_name" type="xs:string" minOccurs="0"/>
      <xs:element name="reason" type="xs:string" minOccurs="0"/>
      <xs:element name="endpoint_policy" type="xs:string" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

Invoking the NAS IPAddress API Call

Note Make sure that you have verified that the target node to which you are issuing an API call is a valid Cisco Monitoring ISE node. To verify the persona of a Cisco ISE node, see Verifying a Cisco Monitoring ISE Node, page 1-2.

To issue the NAS IPAddress API call, complete the following steps:

Step 1 Log into the target Cisco Monitoring ISE node.

For example, when you initially log into a Cisco Monitoring ISE node with the hostname of acme123, this would display the following URL Address field for this node:

```
https://acme123/admin/LoginAction.do#pageId=com_cisco_xmp_web_page_tmpdash
```

Step 2 Enter the IPAddress API call in the URL Address field of the target node by replacing the "/admin/" component with the API call component (/ise/mnt/api/<specific-api-call>/<nasipaddress>):

```
https://acme123/ise/mnt/api/Session/IPAddress/10.10.10.10
```

Note Make sure that you specify the NAS IP address using the xxx.xxx.xxx.xxx format.

Note You must carefully enter each API call in the URL Address field of a target node because these calls are case-sensitive. The use of "mnt" in the API call convention represents a Cisco Monitoring ISE node.

Step 3 Press **Enter** to issue the API call.

Sample Data Returned from the IPAddress API Call

The following example illustrates the session-related data returned from the list of active sessions when you invoke an IPAddress API call on a target Cisco Monitoring ISE node:

```
This XML file does not appear to have any style information associated with it. The
document tree is shown below.
-
<sessionParameters>
<passed xsi:type="xs:boolean">true</passed>
<failed xsi:type="xs:boolean">false</failed>
<user_name>ipepvpnuser</user_name>
<nas_ip_address>10.10.10.10</nas_ip_address>
<calling_station_id>172.23.130.90</calling_station_id>
```

```
<nas_port>1015</nas_port>
<identity_group>iPEP-VPN-Group</identity_group>
<network_device_name>iPEP-HA-Routed</network_device_name>
<acs_server>HAREESH-R6-1-PDP2</acs_server>
<authen_protocol>PAP_ASCII</authen_protocol>
-
<network_device_groups>
Device Type#All Device Types,Location#All Locations
</network_device_groups>
<access_service>RADIUS</access_service>
<auth_acs_timestamp>2010-12-15T19:57:29.885Z</auth_acs_timestamp>
<authentication_method>PAP_ASCII</authentication_method>
-
<execution_steps>
11001,11017,15008,15048,15048,15004,15041,15004,15013,24210,24212,22037,15036,15048,15048,
15004,15016,11002
</execution_steps>
<audit_session_id>0acb6be4000000044D091DA9</audit_session_id>
<nac_policy_compliance>NotApplicable</nac_policy_compliance>
<auth_id>1291240762083580</auth_id>
<auth_acsview_timestamp>2010-12-15T19:57:29.887Z</auth_acsview_timestamp>
<message_code>5200</message_code>
<acs_session_id>HAREESH-R6-1-PDP2/81148292/693</acs_session_id>
<service_selection_policy>iPEP-VPN</service_selection_policy>
<identity_store>Internal Users</identity_store>
-
<response>
{User-Name=ipepvpnuser; State=ReauthSession:0acb6be4000000044D091DA9;
Class=CACS:0acb6be4000000044D091DA9:HAREESH-R6-1-PDP2/81148292/693;
Termination-Action=RADIUS-Request; }
</response>
<service_type>Framed</service_type>
-
<cisco_av_pair>
audit-session-id=0acb6be4000000044D091DA9,ipep-proxy=true
</cisco_av_pair>
<acs_username>ipepvpnuser</acs_username>
<radius_username>ipepvpnuser</radius_username>
<selected_identity_store>Internal Users</selected_identity_store>
<authentication_identity_store>Internal Users</authentication_identity_store>
<identity_policy_matched_rule>Default</identity_policy_matched_rule>
<nas_port_type>Virtual</nas_port_type>
<selected_azn_profiles>iPEP-Unknown-Auth-Profile</selected_azn_profiles>
<tunnel_details>Tunnel-Client-Endpoint=(tag=0) 172.23.130.90</tunnel_details>
-
<other_attributes>
ConfigVersionId=44,DestinationIPAddress=10.203.107.162,DestinationPort=1812,Protocol=Radiu
s,Framed-Protocol=PPP,Proxy-State=Cisco Secure
ACS9e733142-070a-11e0-c000-000000000000-2906094480-3222,CPMSessionID=0acb6be4000000044D091
DA9,CPMSessionID=0acb6be4000000044D091DA9,Device Type=Device Type#All Device
Types,Location=Location#All Locations,Model Name=Unknown,Software Version=Unknown,Device
IP Address=10.203.107.228,Called-Station-ID=172.23.130.94
</other_attributes>
<response_time>20</response_time>
<acct_id>1291240762083582</acct_id>
<acct_acs_timestamp>2010-12-15T19:57:30.281Z</acct_acs_timestamp>
<acct_acsview_timestamp>2010-12-15T19:57:30.283Z</acct_acsview_timestamp>
<acct_session_id>F1800007</acct_session_id>
<acct_status_type>Start</acct_status_type>
-
```

```
<acct_class>
CACS:0acb6be4000000044D091DA9:HAREESH-R6-1-PDP2/81148292/693
</acct_class>
<acct_delay_time>0</acct_delay_time>
<framed_protocol>PPP</framed_protocol>
<started xsi:type="xs:boolean">true</started>
<stopped xsi:type="xs:boolean">false</stopped>
</sessionParameters>
```

Removing Stale Sessions

Some devices, such as Wireless Lan Controllers (WLCs), may allow stale sessions to linger. In such cases, you can use the HTTP **DELETE** API call to manually delete the inactive sessions. To do so, use **cURL**, a free 3rd-party command line tool for transferring data with URL (HTTP, HTTPS) syntax.

Note GNU Wget, the free utility for retrieving files using HTTP and HTTPS, does not support the HTTP **DELETE** API call.

To remove a stale sessions, complete the following steps:

Step 1 Log into the target Cisco Monitoring ISE node from the command line.

Note API calls are case-sensitive, and must be entered carefully. The variable <mntnode> represents a Cisco Monitoring ISE node.

Step 2 To manually delete a stale session for a MAC address, issue the following API call on the command line:

```
curl -X DELETE https://<mntnode>/ise/mnt/api/Session/Delete/MACAddress/<madaddress>
```

Step 3 To manually delete a stale session for a session ID, issue the following API call on the command line:

```
curl -X DELETE https://<mntnode>/ise/mnt/api/Session/Delete/SessionID/<sid#>
```

Step 4 To manually delete all sessions, issue the following API call on the command line:

```
curl -X DELETE https://<mntnode>/ise/mnt/api/Session/Delete/All
```

CHAPTER 3

Using the Query APIs for Troubleshooting

This chapter provides examples and describes how to use the individual Cisco Prime Network Control System (NCS) REST API calls that are supported in this release. The Cisco Prime NCS API calls provide a mechanism for retrieving key troubleshooting information about the target Cisco Monitoring ISE node sessions that include node version and type, failure reasons, authentication status, and accounting status.

The following sections provide you with troubleshooting information obtained by using Query API calls, and this information is in the form of output schema file examples, procedures for issuing each API call, and a sample of the data returned by each API call:

Troubleshooting Cisco ISE using the Query API Calls

The following sections provide key Cisco Prime NCS troubleshooting API calls that send status requests to the target Cisco Monitoring ISE node that you designated in your Cisco ISE deployment and retrieve the following diagnostic-related information:

- Node version and type (using the Version API call)
- Failure reasons (using the FailureReasons API call)
- Authentication status (using the AuthStatus API call)
- Accounting status (using the AcctStatus API call)

Node Version and Type API Call

You can use the Version API call to test the REST programmatic interface (PI) service and the credentials of each node. This section provides a schema file output example, a procedure for requesting the version of the Cisco ISE software and the node type by invoking this API call, and a sample of the node version and type that is returned after this API call is issued.

The node types can be any of the following:

- STANDALONE_MNT_NODE = 0
- ACTIVE_MNT_NODE= 1
- BACKUP_MNT_NODE = 2
- NOT_AN_MNT_NODE = 3

Version API Output Schema

This sample schema file is the output of the Version API call after sending it to the target Cisco Monitoring ISE node:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="product" type="product"/>

  <xs:complexType name="product">
    <xs:sequence>
      <xs:element name="version" type="xs:string" minOccurs="0"/>
  <xs:element name="type_of_node" type="xs:int"/>
    </xs:sequence>
    <xs:attribute name="name" type="xs:string"/>
  </xs:complexType>
</xs:schema>
```

Invoking the Version API Call

Note Make sure that you have verified that the target node to which you are issuing an API call is a valid Cisco Monitoring ISE node. To verify the persona of a Cisco ISE node, see Verifying a Cisco Monitoring ISE Node, page 1-2.

To issue the Version API call, complete the following steps:

Step 1 Log into the target Cisco Monitoring ISE node.

For example, when you initially log into a Cisco Monitoring ISE node with the hostname of acme123, this would display the following URL Address field for this node:

```
https://acme123/admin/LoginAction.do#pageId=com_cisco_xmp_web_page_tmpdash
```

Step 2 Enter the Version API call in the URL Address field of the target node by replacing the “/admin/” component with the API call component (/ise/mnt/api/<specific-api-call>):

```
https://acme123/ise/mnt/api/Version
```

Note You must carefully enter each API call in the URL Address field of a target node because these calls are case-sensitive. The use of "mnt" in the API call convention represents a Cisco Monitoring ISE node.

Step 3 Press **Enter** to issue the API call.

Sample Data Returned from the Version API Call

The following example illustrates the data returned when you invoke a Version API call on a target Cisco Monitoring ISE node. This API call returns the following two values for the target node:

- Node version (this example displays 1.0.3.032).
- Type of Cisco Monitoring ISE node (this example displays a "1", which means an active Cisco Monitoring ISE node).

```
This XML file does not appear to have any style information associated with it. The
document tree is shown below.

-
<product name="Cisco Identity Services Engine">
<version>1.0.3.032</version>
<type_of_node>1</type_of_node>
</product>
```

Failure Reasons API Call

You can use the FailureReasons API call to return a list of failure reasons returned in the authentication status check done on the target node. This section provides a schema file output example, a procedure for requesting a list of all failure reasons logged by the Cisco Monitoring ISE node by invoking this API call, and a sample of the failure reasons returned after this API call is issued. Each failure reason that is returned consists of the following elements shown in Table 3-1.

Note For details about using the Cisco ISE Failure Reasons Editor to access the complete list of failure reasons, see Using the Cisco ISE Failure Reasons Editor, page A-1.

Table 3-1 Product Documentation for Cisco Identity Services Engine

Failure Reason Elements	Example
Failure reason ID	`<failureReason id="11011">`
Code	`<11011 RADIUS listener failed>`
Cause	`<Could not open one or more of the ports used to receive RADIUS requests>`
Resolution	`<Ensure that the ports 1812, 1813, 1645 and 1646 are not being used by another process on the system>`

Note You can also check for failure reason reports using the Cisco ISE user interface (click **Monitor > Reports > Catalog > Failure Reasons**), which will display failure reason reports.

FailureReasons API Output Schema

This sample schema file is the output of the FailureReasons API call after sending the request to a target Cisco Monitoring ISE node:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="failureReasonList" type="failureReasonList"/>

  <xs:complexType name="failureReasonList">
    <xs:sequence>
      <xs:element name="failureReason" type="failureReason" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="failureReason">
    <xs:sequence>
      <xs:element name="code" type="xs:string" minOccurs="0"/>
      <xs:element name="cause" type="xs:string" minOccurs="0"/>
      <xs:element name="resolution" type="xs:string" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="id" type="xs:string"/>
  </xs:complexType>
</xs:schema>
```

Invoking the FailureReasons API Call

Note Make sure that you have verified that the target node to which you are issuing an API call is a valid Cisco Monitoring ISE node. To verify the persona of a Cisco ISE node, see Verifying a Cisco Monitoring ISE Node, page 1-2.

To issue the FailureReasons API call, complete the following steps:

Step 1 Log into the target Cisco Monitoring ISE node.

For example, when you initially log into a Cisco Monitoring ISE node with the hostname of acme123, this would display the following URL Address field for this node:

```
https://acme123/admin/LoginAction.do#pageId=com_cisco_xmp_web_page_tmpdash
```

Step 2 Enter the FailureReasons API call in the URL Address field of the target node by replacing the "/admin/" component with the API call component (/ise/mnt/api/<specific-api-call>):

```
https://acme123/ise/mnt/api/FailureReasons
```

Note You must carefully enter each API call in the URL Address field of a target node because these calls are case-sensitive. The use of "mnt" in the API call convention represents a Cisco Monitoring ISE node.

Step 3 Press **Enter** to issue the API call.

Sample Data Returned from the FailureReasons API Call

The following example illustrates the data returned when you invoke a FailureReasons API call on a target Cisco Monitoring ISE node. This API call returns a list of failure reasons from the target node, and each failure reason is defined by a failure ID, a failure code, a cause, and a resolution (if known).

Note The following FailureReasons API call example only displays a small sample of data that can be returned.

```
This XML file does not appear to have any style information associated with it. The
document tree is shown below.

-
<failureReasonList>
-
<failureReason id="100001">
-
<code>
100001 AUTHMGR-5-FAIL Authorization failed for client
</code>
<cause>This may or may not be indicating a violation</cause>
-
<resolution>
Please review and resolve according to your organization's policy
</resolution>
</failureReason>
-
<failureReason id="100002">
-
<code>
100002 AUTHMGR-5-SECURITY_VIOLATION Security violation on the interface
</code>
<cause>This may or may not be indicating a violation</cause>
-
<resolution>
Please review and resolve according to your organization's policy
</resolution>
</failureReason>
-
<failureReason id="100003">
-
<code>
100003 AUTHMGR-5-UNAUTHORIZED Interface unauthorized
</code>
<cause>This may or may not be indicating a violation</cause>
```

```
-
<resolution>
Please review and resolve according to your organization's policy
</resolution>
</failureReason>
-
<failureReason id="100004">
-
<code>
100004 DOT1X-5-FAIL Authentication failed for client
</code>
<cause>This may or may not be indicating a violation</cause>
-
<resolution>
Please review and resolve according to your organization's policy
</resolution>
</failureReason>
-
<failureReason id="100005">
<code>100005 MAB-5-FAIL Authentication failed for client</code>
<cause>This may or may not be indicating a violation</cause>
-
<resolution>
Please review and resolve according to your organization's policy
</resolution>
</failureReason>
-
<failureReason id="100006">
-
<code>
100006 RADIUS-4-RADIUS_DEAD RADIUS server is not responding
</code>
<cause>This may or may not be indicating a violation</cause>
-
<resolution>
Please review and resolve according to your organization's policy
</resolution>
</failureReason>
-
<failureReason id="100007">
-
<code>
100007 EPM-6-POLICY_APP_FAILURE Interface ACL not configured
</code>
<cause>This may or may not be indicating a violation</cause>
-
<resolution>
Please review and resolve according to your organization's policy
</resolution>
</failureReason>
```

For more information

For more information about the Cisco ISE Failure Reasons Editor, see Appendix A, “Using the Cisco ISE Failure Reasons Editor”.

Authentication Status API Call

You can use the AuthStatus API call to check the authentication status of sessions on the target node. The query associated with this API call requires at least one MAC address to be searched for a match, with a user-configurable limit of the most recent records for the specified MAC address returned.

This section provides a schema file output example, a procedure for sending a a request to search for session authentication status on a target Monitoring mode by invoking this API call, and a sample of the data returned after this API call is issued.

The AuthStatus API call lets you configure the following search-related parameters:

- Duration—Defines the number of seconds in which an attempt is made to search and retrieve the authentication status records associated with the designated MAC address. Valid user-configurable values range from 1 to 864000 seconds (10 days). If you enter a value of 0 seconds, this specifies a default duration of 10 days.
- Records—Defines the number of session records to be searched per MAC address. Valid user-configurable values range from 1 to 500 records. If you enter 0, this specifies a default setting of 200 records.

Note If you specify the value 0 for both the duration and the records parameters, this API call returns only the very latest authentication session record associated with the designated MAC address(es).

- Attributes—Defines the number of attributes in the authentication status table that are returned from an authentication status search using the AuthStatus API call. Valid values include 0 (the default), All, or user_name+acs_timestamp (see the AuthStatus schema example, AcctStatus API Output Schema, page 3-15).
  - If you enter “0”, the attributes defined in Table 3-2 are returned. These are listed in the restAuthStatus section of the output schema.
  - If you enter “All”, a fuller set of attributes are returned. These are listed in the fullRESTAuthStatus section of the output schema.
  - If you enter the values listed in the schema for user_name+acs_timestamp, only those attributes are returned. The user_name and acs_timestamp attributes are listed in the restAuthStatus section of the output schema.

Table 3-2 Authentication Status Table Attributes

Attribute	Description
name=”passed”	One of two possible authentication status results: • Passed
name=”failed”	One of two possible authentication status results: • Failed
name=”user_name”	User name
name=”nas_ip_address”	IP address/hostname for the network access switch
name=”failure_reason”	Reason for session authentication failure
name=”calling_station_id”	Source IP address
name=”nas_port”	Network access server port

Table 3-2 Authentication Status Table Attributes (continued)

Attribute	Description
name=”identity_group”	A logical group consisting of related users and hosts
name=”network_device_name”	Name of the network device
name=”acs_server”	Name of the Cisco ISE appliance
name=”eap_authentication”	Extensible Authentication Protocol (EAP) method used for authentication request
name=”framed_ip_address”	Address configured for a specific user
network_device_groups”	A logical group consisting of related network devices
name=”access_service”	Applied access service
name=”acs_timestamp”	Time stamp that is associated with the Cisco ISE authentication request
name=”authentication_method”	Identifies the method used in authentication
name=”execution_steps”	List of message codes for each diagnostic message logged while processing the request
name=”radius_response”	Type of RADIUS response (for example, VLAN or ACL)
name=”audit_session_id”	ID of the authentication session
name=”nas_identifier”	A network access server (NAS) associated with a specific resource
name=”nas_port_id”	ID of the NAS port used
name=”nac_policy_compliance”	Reflects Posture status (compliant or non-compliant)
name=”selected_azn_profiles”	Identifies the profile used in authorization
name=”service_type”	Indicates a framed user
name=”eap_tunnel”	Tunnel or outer method used for EAP authentication
name=”message_code”	Identifier of the audit message that defines the processed request result
name=”destination_ip_address”	Identifies the destination IP address

AuthStatus API Output Schema

This sample schema file is the output of the AuthStatus API call after sending it to a specified session on the target Cisco Monitoring ISE node:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="authStatusOutputList" type="fullRESTAuthStatusOutputList"/>

  <xs:complexType name="fullRESTAuthStatusOutputList">
    <xs:sequence>
      <xs:element name="authStatusList" type="fullRESTAuthStatusList" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
```

```
  <xs:complexType name="fullRESTAuthStatusList">
    <xs:sequence>
      <xs:element name="authStatusElements" type="fullRESTAuthStatus" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="key" type="xs:string"/>
  </xs:complexType>

  <xs:complexType name="fullRESTAuthStatus">
    <xs:complexContent>
      <xs:extension base="restAuthStatus">
        <xs:sequence>
          <xs:element name="id" type="xs:long" minOccurs="0"/>
          <xs:element name="acsview_timestamp" type="xs:dateTime" minOccurs="0"/>
          <xs:element name="acs_session_id" type="xs:string" minOccurs="0"/>
          <xs:element name="service_selection_policy" type="xs:string" minOccurs="0"/>
          <xs:element name="authorization_policy" type="xs:string" minOccurs="0"/>
          <xs:element name="identity_store" type="xs:string" minOccurs="0"/>
          <xs:element name="response" type="xs:string" minOccurs="0"/>
          <xs:element name="cts_security_group" type="xs:string" minOccurs="0"/>
          <xs:element name="use_case" type="xs:string" minOccurs="0"/>
          <xs:element name="cisco_av_pair" type="xs:string" minOccurs="0"/>
          <xs:element name="ad_domain" type="xs:string" minOccurs="0"/>
          <xs:element name="acs_username" type="xs:string" minOccurs="0"/>
          <xs:element name="radius_username" type="xs:string" minOccurs="0"/>
          <xs:element name="nac_role" type="xs:string" minOccurs="0"/>
          <xs:element name="nac_username" type="xs:string" minOccurs="0"/>
          <xs:element name="nac_posture_token" type="xs:string" minOccurs="0"/>
          <xs:element name="nac_radius_is_user_auth" type="xs:string" minOccurs="0"/>
          <xs:element name="selected_posture_server" type="xs:string" minOccurs="0"/>
          <xs:element name="selected_identity_store" type="xs:string" minOccurs="0"/>
          <xs:element name="authentication_identity_store" type="xs:string"

minOccurs="0"/>
          <xs:element name="azn_exp_pol_matched_rule" type="xs:string" minOccurs="0"/>
          <xs:element name="ext_pol_server_matched_rule" type="xs:string" minOccurs="0"/>
          <xs:element name="grp_mapping_pol_matched_rule" type="xs:string" minOccurs="0"/>
          <xs:element name="identity_policy_matched_rule" type="xs:string" minOccurs="0"/>
          <xs:element name="nas_port_type" type="xs:string" minOccurs="0"/>
          <xs:element name="query_identity_stores" type="xs:string" minOccurs="0"/>
          <xs:element name="sel_exp_azn_profiles" type="xs:string" minOccurs="0"/>
          <xs:element name="selected_query_identity_stores" type="xs:string"
minOccurs="0"/>
          <xs:element name="tunnel_details" type="xs:string" minOccurs="0"/>
          <xs:element name="cisco_h323_attributes" type="xs:string" minOccurs="0"/>
          <xs:element name="cisco_ssg_attributes" type="xs:string" minOccurs="0"/>
          <xs:element name="other_attributes" type="xs:string" minOccurs="0"/>
          <xs:element name="response_time" type="xs:long" minOccurs="0"/>
          <xs:element name="nad_failure" type="xs:anyType" minOccurs="0"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="restAuthStatus">
    <xs:sequence>
      <xs:element name="passed" type="xs:anyType" minOccurs="0"/>
      <xs:element name="failed" type="xs:anyType" minOccurs="0"/>
      <xs:element name="user_name" type="xs:string" minOccurs="0"/>
      <xs:element name="nas_ip_address" type="xs:string" minOccurs="0"/>
      <xs:element name="failure_reason" type="xs:string" minOccurs="0"/>
      <xs:element name="calling_station_id" type="xs:string" minOccurs="0"/>
      <xs:element name="nas_port" type="xs:string" minOccurs="0"/>
```

```
      <xs:element name="identity_group" type="xs:string" minOccurs="0"/>
      <xs:element name="network_device_name" type="xs:string" minOccurs="0"/>
      <xs:element name="acs_server" type="xs:string" minOccurs="0"/>
      <xs:element name="eap_authentication" type="xs:string" minOccurs="0"/>
      <xs:element name="framed_ip_address" type="xs:string" minOccurs="0"/>
      <xs:element name="network_device_groups" type="xs:string" minOccurs="0"/>
      <xs:element name="access_service" type="xs:string" minOccurs="0"/>
      <xs:element name="acs_timestamp" type="xs:dateTime" minOccurs="0"/>
      <xs:element name="authentication_method" type="xs:string" minOccurs="0"/>
      <xs:element name="execution_steps" type="xs:string" minOccurs="0"/>
      <xs:element name="radius_response" type="xs:string" minOccurs="0"/>
      <xs:element name="audit_session_id" type="xs:string" minOccurs="0"/>
      <xs:element name="nas_identifier" type="xs:string" minOccurs="0"/>
      <xs:element name="nas_port_id" type="xs:string" minOccurs="0"/>
      <xs:element name="nac_policy_compliance" type="xs:string" minOccurs="0"/>
      <xs:element name="selected_azn_profiles" type="xs:string" minOccurs="0"/>
      <xs:element name="service_type" type="xs:string" minOccurs="0"/>
      <xs:element name="eap_tunnel" type="xs:string" minOccurs="0"/>
      <xs:element name="message_code" type="xs:string" minOccurs="0"/>
      <xs:element name="destination_ip_address" type="xs:string" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

Invoking the AuthStatus API Call

Note Make sure that you have verified that the target node to which you are issuing an API call is a valid Cisco Monitoring ISE node. To verify the persona of a Cisco ISE node, see Verifying a Cisco Monitoring ISE Node, page 1-2.

To issue the AuthStatus API call, complete the following steps:

Step 1 Log into the target Cisco Monitoring ISE node.

For example, when you initially log into a Cisco Monitoring ISE node with the hostname of acme123, this would display the following URL Address field for this node:

```
https://acme123/admin/LoginAction.do#pageId=com_cisco_xmp_web_page_tmpdash
```

Step 2 Enter the AuthStatus API call in the URL Address field of the target node by replacing the "/admin/" component with the API call component (/ise/mnt/api/<specific-api-call>/MACAddress/<macaddress>/<seconds>/<numberofrecordspermacaddress>/All):

```
https://acme123/ise/mnt/api/AuthStatus/MACAddress/00:50:56:10:13:02/120/100/All
```

Note You must carefully enter each API call in the URL Address field of a target node because these calls are case-sensitive. The use of "mnt" in the API call convention represents a Cisco Monitoring ISE node.

Step 3 Press **Enter** to issue the API call.

Sample Data Returned from the AuthStatus API Call

The following example illustrates the data returned when you invoke a AuthStatus API call on a target Cisco Monitoring ISE node:

```
This XML file does not appear to have any style information associated with it. The
document tree is shown below.

-
<authStatusOutputList>
-
<authStatusList key="00:25:9C:A3:7D:48">
-
<authStatusElements>
<passed xsi:type="xs:boolean">true</passed>
<failed xsi:type="xs:boolean">false</failed>
<user_name>hareesh6</user_name>
<nas_ip_address>10.203.107.10</nas_ip_address>
<calling_station_id>00:25:9C:A3:7D:48</calling_station_id>
<nas_port>1</nas_port>
<identity_group>iPEP-WLC-Group</identity_group>
<network_device_name>iPEP3</network_device_name>
<acs_server>HAREESH-R6-1-PDP1</acs_server>
<eap_authentication>EAP-MSCHAPv2</eap_authentication>
-
<network_device_groups>
Device Type#All Device Types#iPEP,Location#All Locations
</network_device_groups>
<access_service>Default Network Access</access_service>
<acs_timestamp>2010-12-20T01:38:49.566Z</acs_timestamp>
<authentication_method>MSCHAPV2</authentication_method>
-
<execution_steps>
11001,11017,15008,15048,15048,15004,11507,12500,11006,11001,11018,12101,12100,11006,11001,
11018,12102,12800,12175,12805,12806,12801,12802,12105,11006,11001,11018,12104,12804,12816,
12132,12125,11806,12105,11006,11001,11018,12104,11808,15041,15006,15013,24210,24212,22037,
11824,12105,11006,11001,11018,12104,11810,11814,11519,12128,12105,11006,11001,11018,12104,
12126,12127,15036,15048,15048,15004,15016,12171,12105,11006,11001,11018,12104,12106,11503,
15036,15048,15048,15004,15016,11002
</execution_steps>
<audit_session_id>0acb6b0b0000000D4D0EB3A9</audit_session_id>
<nas_identifier>Cisco_4d:c0:a0</nas_identifier>
<nac_policy_compliance>NotApplicable</nac_policy_compliance>
<selected_azn_profiles>iPEP-Compliant-Authz-Profile</selected_azn_profiles>
<service_type>Framed</service_type>
<eap_tunnel>EAP-FAST</eap_tunnel>
<message_code>5200</message_code>
<destination_ip_address>10.203.107.150</destination_ip_address>
<id>1292549379215912</id>
<acsview_timestamp>2010-12-20T01:38:49.567Z</acsview_timestamp>
<acs_session_id>HAREESH-R6-1-PDP1/81999140/50</acs_session_id>
<service_selection_policy>iPEP-WLC</service_selection_policy>
<authorization_policy>iPEP-WLC-Compliant-Policy</authorization_policy>
<identity_store>Internal Users</identity_store>
-
<response>
{User-Name=hareesh6; State=ReauthSession:0acb6b0b0000000D4D0EB3A9;
Class=CACS:0acb6b0b0000000D4D0EB3A9:HAREESH-R6-1-PDP1/81999140/50;
Termination-Action=RADIUS-Request;
MS-MPPE-Send-Key=04:11:2d:bf:8b:5f:c1:b0:14:b1:73:ad:48:90:65:e0:c2:a3:f7:66:2d:dc:70:f1:a
b:56:cd:09:c4:b0:b7:ae;
MS-MPPE-Recv-Key=7e:38:94:72:e2:a3:8a:e4:90:18:45:61:91:c0:44:ea:0c:21:39:14:2f:7c:9f:55:d
6:52:af:fd:55:48:3f:34; }
```

```
</response>
-
<cisco_av_pair>
audit-session-id=0acb6b0b0000000D4D0EB3A9,ipep-proxy=true
</cisco_av_pair>
<acs_username>hareesh6</acs_username>
<radius_username>anonymous</radius_username>
<selected_identity_store>Internal Users</selected_identity_store>
<authentication_identity_store>Internal Users</authentication_identity_store>
<identity_policy_matched_rule>Default</identity_policy_matched_rule>
<nas_port_type>Wireless - IEEE 802.11</nas_port_type>
-
<tunnel_details>
Tunnel-Type=(tag=0) VLAN,Tunnel-Medium-Type=(tag=0) 802,Tunnel-Private-Group-ID=(tag=0)
208
</tunnel_details>
-
<other_attributes>
ConfigVersionId=18,DestinationPort=1812,Protocol=Radius,Framed-MTU=1300,State=37CPMSession
ID=0acb6b0b0000000D4D0EB3A9;39SessionID=HAREESH-R6-1-PDP1/81999140/50;,Proxy-State=Cisco
Secure
ACS53e5cfac-0a31-11e0-c000-000000000000-2905701264-3372,Airespace-Wlan-Id=2,CPMSessionID=0
acb6b0b0000000D4D0EB3A9,IssuedPacInfo=Issued PAC type=Authorization with expiration time:
Mon Dec 20 02:38:49
2010,CPMSessionID=0acb6b0b0000000D4D0EB3A9,EndPointMACAddress=00-25-9C-A3-7D-48,Device
Type=Device Type#All Device Types#iPEP,Location=Location#All Locations,Model
Name=Unknown,Software Version=Unknown,Device IP
Address=10.203.107.11,Called-Station-ID=00-24-c4-1b-36-70:ipep3
</other_attributes>
<response_time>3</response_time>
</authStatusElements>
-
<authStatusElements>
<passed xsi:type="xs:boolean">true</passed>
<failed xsi:type="xs:boolean">false</failed>
<user_name>hareesh6</user_name>
<nas_ip_address>10.203.107.10</nas_ip_address>
<calling_station_id>00:25:9C:A3:7D:48</calling_station_id>
<nas_port>1</nas_port>
<identity_group>iPEP-WLC-Group</identity_group>
<network_device_name>iPEP3</network_device_name>
<acs_server>HAREESH-R6-1-PDP1</acs_server>
<eap_authentication>EAP-MSCHAPv2</eap_authentication>
-
<network_device_groups>
Device Type#All Device Types#iPEP,Location#All Locations
</network_device_groups>
<access_service>Default Network Access</access_service>
<acs_timestamp>2010-12-19T01:32:39.220Z</acs_timestamp>
<authentication_method>MSCHAPV2</authentication_method>
-
<execution_steps>
11001,11017,15008,15048,15048,15004,11507,12500,11006,11001,11018,12101,12100,11006,11001,
11018,12102,12800,12175,12805,12806,12801,12802,12105,11006,11001,11018,12104,12804,12816,
12132,12125,11806,12105,11006,11001,11018,12104,11808,15041,15006,15013,24210,24212,22037,
11824,12105,11006,11001,11018,12104,11810,11814,11519,12128,12105,11006,11001,11018,12104,
12126,12127,15036,15048,15048,15004,15016,12171,12105,11006,11001,11018,12104,12106,11503,
15036,15048,15048,15004,15016,11002
</execution_steps>
<audit_session_id>0acb6b0b0000000C4D0D60B6</audit_session_id>
<nas_identifier>Cisco_4d:c0:a0</nas_identifier>
<nac_policy_compliance>NotApplicable</nac_policy_compliance>
<selected_azn_profiles>iPEP-Compliant-Authz-Profile</selected_azn_profiles>
<service_type>Framed</service_type>
```

```
<eap_tunnel>EAP-FAST</eap_tunnel>
<message_code>5200</message_code>
<destination_ip_address>10.203.107.150</destination_ip_address>
<id>1292549379206881</id>
<acsview_timestamp>2010-12-19T01:32:39.218Z</acsview_timestamp>
<acs_session_id>HAREESH-R6-1-PDP1/81999140/46</acs_session_id>
<service_selection_policy>iPEP-WLC</service_selection_policy>
<authorization_policy>iPEP-WLC-Compliant-Policy</authorization_policy>
<identity_store>Internal Users</identity_store>
-
<response>
{User-Name=hareesh6; State=ReauthSession:0acb6b0b0000000C4D0D60B6;
Class=CACS:0acb6b0b0000000C4D0D60B6:HAREESH-R6-1-PDP1/81999140/46;
Termination-Action=RADIUS-Request;
MS-MPPE-Send-Key=f0:f4:5d:38:c4:5d:e8:85:51:65:ea:9e:ad:27:9f:c6:50:ae:11:ae:f8:8c:9d:c2:5
c:d3:33:06:36:be:14:79;
MS-MPPE-Recv-Key=d3:4a:2b:e6:6b:f8:31:ef:cc:84:d0:57:96:24:ab:e4:9b:45:3a:43:a7:1a:05:e7:5
d:a0:46:33:02:63:ef:39; }
</response>
-
<cisco_av_pair>
audit-session-id=0acb6b0b0000000C4D0D60B6,ipep-proxy=true
</cisco_av_pair>
<acs_username>hareesh6</acs_username>
<radius_username>anonymous</radius_username>
<selected_identity_store>Internal Users</selected_identity_store>
<authentication_identity_store>Internal Users</authentication_identity_store>
<identity_policy_matched_rule>Default</identity_policy_matched_rule>
<nas_port_type>Wireless - IEEE 802.11</nas_port_type>
-
<tunnel_details>
Tunnel-Type=(tag=0) VLAN,Tunnel-Medium-Type=(tag=0) 802,Tunnel-Private-Group-ID=(tag=0)
208
</tunnel_details>
-
<other_attributes>
ConfigVersionId=18,DestinationPort=1812,Protocol=Radius,Framed-MTU=1300,State=37CPMSession
ID=0acb6b0b0000000C4D0D60B6;39SessionID=HAREESH-R6-1-PDP1/81999140/46;,Proxy-State=Cisco
Secure
ACS53e5cfac-0a31-11e0-c000-000000000000-2905701264-3372,Airespace-Wlan-Id=2,CPMSessionID=0
acb6b0b0000000C4D0D60B6,IssuedPacInfo=Issued PAC type=Authorization with expiration time:
Sun Dec 19 02:32:39
2010,CPMSessionID=0acb6b0b0000000C4D0D60B6,EndPointMACAddress=00-25-9C-A3-7D-48,Device
Type=Device Type#All Device Types#iPEP,Location=Location#All Locations,Model
Name=Unknown,Software Version=Unknown,Device IP
Address=10.203.107.11,Called-Station-ID=00-24-c4-1b-36-70:ipep3
</other_attributes>
<response_time>3</response_time>
</authStatusElements>
-
<authStatusElements>
<passed xsi:type="xs:boolean">true</passed>
<failed xsi:type="xs:boolean">false</failed>
<user_name>hareesh6</user_name>
<nas_ip_address>10.203.107.10</nas_ip_address>
<calling_station_id>00:25:9C:A3:7D:48</calling_station_id>
<nas_port>1</nas_port>
<identity_group>iPEP-WLC-Group</identity_group>
<network_device_name>iPEP3</network_device_name>
<acs_server>HAREESH-R6-1-PDP1</acs_server>
<eap_authentication>EAP-MSCHAPv2</eap_authentication>
-
<network_device_groups>
Device Type#All Device Types#iPEP,Location#All Locations
```

```
</network_device_groups>
<access_service>Default Network Access</access_service>
<acs_timestamp>2010-12-18T01:26:22.089Z</acs_timestamp>
<authentication_method>MSCHAPV2</authentication_method>
-
<execution_steps>
11001,11017,15008,15048,15048,15004,11507,12500,11006,11001,11018,12101,12100,11006,11001,
11018,12102,12800,12805,12806,12807,12810,12105,11006,11001,11018,12104,12812,12804,12801,
12802,12816,12149,12105,11006,11001,11018,12104,12125,11521,12105,11006,11001,11018,12104,
11522,11806,12105,11006,11001,11018,12104,11808,15041,15006,15013,24210,24212,22037,11824,
12105,11006,11001,11018,12104,11810,11814,11519,12128,12105,11006,11001,11018,12104,12126,
12127,15036,15048,15048,15004,15016,12169,12105,11006,11001,11018,12104,12651,12107,11503,
15036,15048,15048,15004,15016,11002
</execution_steps>
<audit_session_id>0acb6b0b0000000B4D0C0DBD</audit_session_id>
<nas_identifier>Cisco_4d:c0:a0</nas_identifier>
<nac_policy_compliance>NotApplicable</nac_policy_compliance>
<selected_azn_profiles>iPEP-Compliant-Authz-Profile</selected_azn_profiles>
<service_type>Framed</service_type>
<eap_tunnel>EAP-FAST</eap_tunnel>
<message_code>5200</message_code>
<destination_ip_address>10.203.107.150</destination_ip_address>
<id>1292549379197803</id>
<acsview_timestamp>2010-12-18T01:26:22.042Z</acsview_timestamp>
<acs_session_id>HAREESH-R6-1-PDP1/81999140/30</acs_session_id>
<service_selection_policy>iPEP-WLC</service_selection_policy>
<authorization_policy>iPEP-WLC-Compliant-Policy</authorization_policy>
<identity_store>Internal Users</identity_store>
-
<response>
{User-Name=hareesh6; State=ReauthSession:0acb6b0b0000000B4D0C0DBD;
Class=CACS:0acb6b0b0000000B4D0C0DBD:HAREESH-R6-1-PDP1/81999140/30;
Termination-Action=RADIUS-Request;
MS-MPPE-Send-Key=d3:94:df:2b:fc:18:12:91:ad:4f:3b:09:d1:76:93:83:21:83:33:3a:14:b9:9b:c0:a
0:81:71:96:95:64:2c:ed;
MS-MPPE-Recv-Key=3b:c2:31:58:81:8a:34:24:d4:55:03:cd:a2:91:85:49:7f:16:36:30:d9:8d:24:a7:5
0:ec:3e:df:7a:85:ea:5c; }
</response>
-
<cisco_av_pair>
audit-session-id=0acb6b0b0000000B4D0C0DBD,ipep-proxy=true
</cisco_av_pair>
<acs_username>hareesh6</acs_username>
<radius_username>anonymous</radius_username>
<selected_identity_store>Internal Users</selected_identity_store>
<authentication_identity_store>Internal Users</authentication_identity_store>
<identity_policy_matched_rule>Default</identity_policy_matched_rule>
<nas_port_type>Wireless - IEEE 802.11</nas_port_type>
-
<tunnel_details>
Tunnel-Type=(tag=0) VLAN,Tunnel-Medium-Type=(tag=0) 802,Tunnel-Private-Group-ID=(tag=0)
208
</tunnel_details>
-
<other_attributes>
ConfigVersionId=18,DestinationPort=1812,Protocol=Radius,Framed-MTU=1300,State=37CPMSession
ID=0acb6b0b0000000B4D0C0DBD;39SessionID=HAREESH-R6-1-PDP1/81999140/30;,Proxy-State=Cisco
Secure
ACS53e5cfac-0a31-11e0-c000-000000000000-2905701264-3372,Airespace-Wlan-Id=2,CPMSessionID=0
acb6b0b0000000B4D0C0DBD,IssuedPacInfo=Issued PAC type=Tunnel V1 with expiration time: Fri
Mar 18 01:26:22
2011,CPMSessionID=0acb6b0b0000000B4D0C0DBD,EndPointMACAddress=00-25-9C-A3-7D-48,Device
```

```
Type=Device Type#All Device Types#iPEP,Location=Location#All Locations,Model
Name=Unknown,Software Version=Unknown,Device IP
Address=10.203.107.11,Called-Station-ID=00-24-c4-1b-36-70:ipep3
</other_attributes>
<response_time>3</response_time>
</authStatusElements>
</authStatusList>
</authStatusOutputList>
```

Account Status API Call

You can use the AcctStatus API call to retrieve the latest device and session account information on the target node. This section provides a schema file output example, a procedure for sending a request for the latest device and session information by invoking this API call, and a sample of the data returned after this API call is issued. The AcctStatus API call lets you configure a time-related parameter:

- Duration—Defines the number of seconds in which an attempt is made to search and retrieve the latest account device records associated with the designated MAC address. Valid user-configurable values range from 1 to 432000 seconds (5 days).
  - If you enter a value of 2400 seconds (40 minutes), this means that you want the latest account device records for the designated MAC address that are available in the past 40 minutes.
  - If you enter a value of 0 seconds, this specifies a default duration of 15 minutes (900 seconds). This means that you want the latest account device records for the designated MAC address that are available within this time period.

The AcctList API call provides the following account status data fields as API outputs (see Table 3-3):

Table 3-3 Accounting Status Data Fields

Data Field	Description
MAC address	MAC address of the client
audit-session-id	Authentication session ID
Packets in	Packets received count total
Packets out	Packets transmitted count total
Bytes in	Bytes received count total
Bytes out	Bytes transmitted count total
Session time	Duration of current sessions

AcctStatus API Output Schema

This sample schema file is the output of the AcctStatus API call after sending it to a specified session on the target Cisco Monitoring ISE node:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="acctStatusOutputList" type="restAcctStatusOutputList"/>

  <xs:complexType name="restAcctStatusOutputList">
    <xs:sequence>
      <xs:element name="acctStatusList" type="restAcctStatusList" minOccurs="0"
maxOccurs="unbounded"/>
```

```
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="restAcctStatusList">
    <xs:sequence>
      <xs:element name="acctStatusElements" type="restAcctStatus" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="macAddress" type="xs:string"/>
    <xs:attribute name="username" type="xs:string"/>
  </xs:complexType>

  <xs:complexType name="restAcctStatus">
    <xs:sequence>
      <xs:element name="calling_station_id" type="xs:string" minOccurs="0"/>
      <xs:element name="audit_session_id" type="xs:string" minOccurs="0"/>
      <xs:element name="paks_in" type="xs:long" minOccurs="0"/>
      <xs:element name="paks_out" type="xs:long" minOccurs="0"/>
      <xs:element name="bytes_in" type="xs:long" minOccurs="0"/>
      <xs:element name="bytes_out" type="xs:long" minOccurs="0"/>
      <xs:element name="session_time" type="xs:long" minOccurs="0"/>
      <xs:element name="username" type="xs:string" minOccurs="0"/>
  <xs:element name="server" type="xs:string" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

Invoking the AcctStatus API Call

Note Make sure that you have verified that the target node to which you are issuing an API call is a valid Cisco Monitoring ISE node. To verify the persona of a Cisco ISE node, see Verifying a Cisco Monitoring ISE Node, page 1-2.

To issue the AcctStatus API call, complete the following steps:

Step 1 Log into the target Cisco Monitoring ISE node.

For example, when you initially log into a Cisco Monitoring ISE node with the hostname of acme123, this would display the following URL Address field for this node:

```
https://acme123/admin/LoginAction.do#pageId=com_cisco_xmp_web_page_tmpdash
```

Step 2 Enter the AcctStatus API call in the URL Address field of the target node by replacing the "/admin/" component with the API call component (/ise/mnt/api/<specific-api-call>/MACAddress/<macaddress>/<durationofcurrenttime>):

```
https://acme123/ise/mnt/api/AcctStatus/MACAddress/00:26:82:7B:D2:51/1200
```

Note You must carefully enter each API call in the URL Address field of a target node because these calls are case-sensitive. The use of "mnt" in the API call convention represents a Cisco Monitoring ISE node.

Step 3 Press **Enter** to issue the API call.

Sample Data Returned from the AcctStatus API Call

The following example illustrates the data returned when you invoke an AcctStatus API call on a target Cisco Monitoring ISE node:

```
This XML file does not appear to have any style information associated with it. The
document tree is shown below.

-
<acctStatusOutputList>
-
<acctStatusList macAddress="00:25:9C:A3:7D:48">
-
<acctStatusElements>
<calling_station_id>00:25:9C:A3:7D:48</calling_station_id>
<audit_session_id>0acb6b0b0000000B4D0C0DBD</audit_session_id>
<paks_in>0</paks_in>
<paks_out>0</paks_out>
<bytes_in>0</bytes_in>
<bytes_out>0</bytes_out>
<session_time>240243</session_time>
<server>HAREESH-R6-1-PDP1</server>
</acctStatusElements>
</acctStatusList>
</acctStatusOutputList>
```

CHAPTER 4

Using the Change of Authorization REST APIs

This chapter provides examples and describes how to use the following individual Change of Authorization (CoA) REST API calls that are supported in this release of Cisco Identity Services Engine. The CoA API calls provide the means for sending session authentication and session disconnect commands to a specified Cisco Monitoring ISE node in your Cisco ISE deployment.

The following sections provide API output schema file examples, procedures for issuing each API call, and a sample of the data returned by each API call:

Using the CoA Session Management API Calls

The CoA session management API calls allow you to send reauthentication and disconnect commands to a specified session on a target Cisco Monitoring ISE node in your Cisco ISE deployment:

- Session reauthentication (Reauth)
- Session disconnection (Disconnect)

Session Reauthentication API Call

This section provides a schema file output example, a procedure for sending a session reauthentication command and Reauth type by invoking the Reauth API call, and a sample of the data returned after this API call is issued. The reauth types can be any of the following:

- REAUTH_TYPE_DEFAULT = 0
- REAUTH_TYPE_LAST = 1
- REAUTH_TYPE_RERUN = 2

Reauth API Output Schema

This sample schema file is the output of the Reauth API call after sending it to a specified session on the target Cisco Monitoring ISE node:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="remoteCoA" type="coAResult"/>
<xs:complexType name="coAResult">
    <xs:sequence>
      <xs:element name="results" type="xs:boolean" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="requestType" type="xs:string"/>
  </xs:complexType>
</xs:schema>
```

Invoking the Reauth API Call

Note Make sure that you have verified that the target node to which you are issuing an API call is a valid Cisco Monitoring ISE node. To verify the persona of a Cisco ISE node, see Verifying a Cisco Monitoring ISE Node, page 1-2.

To issue the Reauth API call, complete the following steps:

Step 1 Log into the target Cisco Monitoring ISE node.

For example, when you initially log into a Cisco Monitoring ISE node with the hostname of acme123, this would display the following URL Address field for this node:

```
https://acme123/admin/LoginAction.do#pageId=com_cisco_xmp_web_page_tmpdash
```

Step 2 Enter the Reauth API call in the URL Address field of the target node by replacing the "/admin/" component with the API call component (/ise/mnt/api/CoA/<specific-api-call>/<macaddress>/<reauthtype>/<nasipaddress>/<destinationipaddress>):

```
https://acme123/ise/mnt/api/CoA/Reauth/server12/00:26:82:7B:D2:51/2/10.10.10.10
```

Note You must carefully enter each API call in the URL Address field of a target node because these calls are case-sensitive. The use of "mnt" in the API call convention represents a Cisco Monitoring ISE node.

Step 3 Press **Enter** to issue the API call.

Sample Data Returned from the Reauth API Call

The following example illustrates the data returned when you invoke a Reauth API call on a target Cisco Monitoring ISE node. Two possible results can be returned from invoking this command:

- True indicates that the command was successfully executed.
- False means that the command was not executed (due to a variety of conditions).

```
This XML file does not appear to have any style information associated with it. The
document tree is shown below.

-
<remoteCoA requestType="reauth">
<results>true</results>
</remoteCoA>
```

Session Disconnect API Call

This section provides a schema file output example, a procedure for sending a session disconnect command and a port option type by invoking the Disconnect API, and a sample of the data returned after this API call is issued. The disconnect port option types can be any of the following:

- DYNAMIC_AUTHZ_PORT_DEFAULT = 0
- DYNAMIC_AUTHZ_PORT_BOUNCE = 1
- DYNAMIC_AUTHZ_PORT_SHUTDOWN = 2

Disconnect API Output Schema

This sample schema file is the output of the Disconnect API call after sending it to a specified session on the target Cisco Monitoring ISE node:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="remoteCoA" type="coAResult"/>

  <xs:complexType name="coAResult">
    <xs:sequence>
      <xs:element name="results" type="xs:boolean" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="requestType" type="xs:string"/>
  </xs:complexType>
</xs:schema>
```

Invoking the Disconnect API Call

Note Make sure that you have verified that the target node to which you are issuing an API call is a valid Cisco Monitoring ISE node. To verify the persona of a Cisco ISE node, see Verifying a Cisco Monitoring ISE Node, page 1-2.

To issue the Disconnect API call, complete the following steps:

Step 1 Log into the target Cisco Monitoring ISE node.

For example, when you initially log into a Cisco Monitoring ISE node with the hostname of acme123, this would display the following URL Address field for this node:

```
https://acme123/admin/LoginAction.do#pageId=com_cisco_xmp_web_page_tmpdash
```

Step 2 Enter the Disconnect API call in the URL Address field of the target node by replacing the "/admin/" component with the API call component (/ise/mnt/api/CoA/<Disconnect>/<serverhostname>/<macaddress>/<portoptiontype>/<nasipaddress>/<destinationipaddress>):

```
https://acme123/ise/mnt/api/CoA/Disconnect/server12/
00:26:82:7B:D2:51/2/10.10.10.10
```

Note You must carefully enter each API call in the URL Address field of a target node because these calls are case-sensitive. The use of "mnt" in the API call convention represents a Cisco Monitoring ISE node.

Step 3 Press **Enter** to issue the API call.

Sample Data Returned from the Disconnect API Call

The following example illustrates the data returned when you invoke a Disconnect API call on a target Cisco Monitoring ISE node. Two possible results can be returned from invoking this command:

- True indicates that the command was successfully executed.
- False means that the command was not executed (due to a variety of conditions).

```
This XML file does not appear to have any style information associated with it. The
document tree is shown below.

-
<remoteCoA requestType="reauth">
<results>true</results>
</remoteCoA>
```

APPENDIX A

Using the Cisco ISE Failure Reasons Editor

This appendix provides a procedure you can use to access the Cisco ISE Failure Reasons Editor. The Cisco ISE Failure Reason Editor is an option in the Cisco ISE user interface that provides information about all of the failure reasons that could be encountered. You can use this to check on those that are returned as output from a Get Failure Reason Mapping call when using the Cisco ISE Query troubleshooting API.

The Cisco ISE Failure Reasons Editor lets you access the complete list of failure reasons defined by the Cisco ISE software that apply to Cisco Monitoring ISE node operations. The following procedure lets you view or edit the list of defined failure reasons. You must log into the Cisco ISE user interface of the target Cisco Monitoring ISE node to view and access the failure reasons. For details about logging in, see Verifying a Cisco Monitoring ISE Node, page 1-2.

Note For more information about Cisco ISE failure reasons or for general troubleshooting issues, see Chapter 22, “Monitoring and Troubleshooting”, and Appendix D, “Troubleshooting Cisco ISE” in the *Cisco Identity Services Engine User Guide, Release 1.0.4*.

Viewing and Editing Failure Reasons

The Cisco ISE Failure Reason Editor allows you to view the list of failure reasons and edit the description of failure reasons. In addition, it also provides instructions on how to resolve the problem.

To view and edit failure reasons, complete the following steps:

Step 1 Choose **Administration > System > Settings**.

Step 2 In the navigation panel, expand **Monitoring** and select **Failure Reason Editor**.

A list of failure reasons appears in the right panel.

Step 3 To view a known failure reason or to search for failure reasons, complete the following tasks:

- For a known failure reason:
  - Select the radio button or name link that corresponds to the failure reason from the list in the Failure Reason Editor page.
- To search for a failure reason:
  - Enter a text string in the Filter text box and click **Filter**.
  - Select one (or more) of the matching failure reasons that are listed as search results.

Step 4 To edit a failure reason, do the following:

- Click the radio button to the left of the name (the button turns green when selected).
- Click **Edit**.
- In the appropriate field, enter or modify a description, then enter or modify the resolution steps.
- Click **Submit** to save your changes, or click **Cancel** to quit without saving any changes.