



APPENDIX **B**

API Support

This chapter discusses API support for the Clean Access Manager. Topics include:

- [Overview, page B-1](#)
- [Authentication Requirement, page B-1](#)
- [MAC Operations, page B-2](#)
- [Certified Devices List Operation, page B-5](#)
- [User Operations, page B-5](#)
- [Guest Access Operations, page B-8](#)
- [Report Operations, page B-10](#)

Overview

Cisco NAC Appliance provides a utility script called **cisco_api.jsp** that allows you to perform certain operations using HTTPS POST. The Clean Access API for your Clean Access Manager is accessed from a web browser as follows: **https://<ip-or-name>/admin/cisco_api.jsp**.

To use this API, note the following:

- You or someone in your organization must be competent with a scripting language, such as Perl.
- Only administrative user authentication is supported. HTTP, GET, and “No Authentication” APIs are not supported.
- You must install Perl or a similar software on the machine that runs these scripts.
- Cisco TAC does not support debugging of Perl or other scripting packages.

You can view the Cisco API page from your CAM (**https://<ccam-ip-or-name>/admin/cisco_api.jsp**) or view the information at Cisco NAC Appliance FAQ for the `cisco_api.jsp` page:

<http://www.cisco.com/warp/customer/707/ca-mgr-faq2.html#q8>

For information on exempting devices through the CAM web console interface, see [Global Device and Subnet Filtering, page 3-7](#).

Authentication Requirement

Authentication over SSL is required to access the API. Two authentication methods are supported:

- Authentication by Session

An administrator uses the *adminlogin* and *adminlogout* functions to create an authentication shell script that sets a cookie with the session ID to be accessed for the rest of the admin session. If a session ID cookie is not set, the user receives an Invalid User error. The *adminlogin* (administrator login) function returns a session ID. The *adminlogout* function terminates the session; however, if the *adminlogout* function is not used, the CAM terminates the session by the configured or default admin session timeout.

- Authentication by Function

If you do not want to create a shell script using cookies, you can instead perform authentication every time a function is used. If authenticating by function, you must add the *admin* and *password* parameters to all functions that you are using in your existing script. In this case, you do not use the *adminlogin* and *adminlogout* functions.

Administrator Operations

You can use the *adminlogin* and *adminlogout* functions to create a shell script when you use cookies to authenticate by session ID. If you choose not to use these functions, you must include the administrator's username and password parameters within each function.

adminlogin

The *adminlogin* function returns a session ID, which must be set as a cookie for use within another API.

Required In Parameters:

- op: adminlogin
- admin: Specifies an administrator's username.
- passwd: Specifies an administrator's password.

Out Parameters: <!--error=msg--> comment

- Success: msg value of 0 and a session ID appears within a second comment:
<!--session_id=SESSION_ID_STRING-->
- Failure: error string

adminlogout

The *adminlogout* function uses the session ID found in a cookie to log out an administrator.

Required In Parameter:

- op: adminlogout

Out Parameters: <!--error=msg--> comment

- Success: msg value of 0
- Failure: error string

MAC Operations

This section contains the following descriptions:

- [addmac](#), page B-3
- [removemac](#), page B-3
- [addcleanmac](#), page B-4
- [removecleanmac](#), page B-4

addmac

The *addmac* function adds one or more MAC addresses to the certified devices list.

Required In Parameters:

- op: addmac
- mac: Specifies an exact MAC address or a range.
Supported formats: 00:01:12:23:34:45 or 00:01:12:* or 00:01:12:23:34:45-11:22:33:44:55:66

Optional In Parameters:

- ip: Specifies an IPv4 address for an exact MAC address. If you use a wildcard or range to specify a MAC address range, do not use the “ip” parameter. Supported format: 192.168.0.10
- type: Specifies one of the following strings: deny (default), allow, userole, check, or ignore.
- role: Specifies a role name. The role parameter is not required for the unauthenticated role (default) but is required for “userole” or “check”.
- desc: Provides a description.
- ssip: Specifies the IP address used for configuring a Clean Access Server to Clean Access Manager. The default is global.



Note

If you do not use authentication by session ID, the *admin* and *passwd* functions are required. See [Authentication Requirement, page B-1](#).

Out Parameters: <!--error=msg--> comment

- Success: msg value of 0
- Failure: error string

removemac

The *removemac* function removes one or more MAC addresses from the device filters list.

Required In Parameters:

- op: removemac
- mac: Specifies one or more MAC addresses to delete from the device filters list. The MAC addresses must exactly match the display format including wildcards. You can specify multiple MAC addresses with a comma separated list.

Optional In Parameter:

- ssip: Specifies the IP address to use for configuring Clean Access Server to Clean Access Manager. The default is global.

**Note**

If you do not use authentication by session ID, the *admin* and *passwd* functions are required. See [Authentication Requirement, page B-1](#).

Out Parameters: <!--error=msg--> comment

- Success: msg value of 0
- Failure: error string

addcleanmac

The *addcleanmac* function adds one or more MAC addresses to the certified devices list as exempted devices.

Required In Parameters:

- op: addcleanmac
- mac: Specifies the MAC addresses to add. Supported formats 00:01:12:23:34:45 or 00-01-12-23-34-45 or 000112233445

Optional In Parameter:

- ssid: Default is global. Specifies the IP address used for configuring Clean Access Server to Clean Access Manager.

**Note**

If you do not use authentication by session ID, the *admin* and *passwd* functions are required. See [Authentication Requirement, page B-1](#).

Out Parameters: <!--error=msg--> comment

- Success: msg value of 0
- Failure: error string

removecleanmac

The *removecleanmac* function removes one or more MAC addresses from the certified devices list.

Required In Parameters:

- op: removecleanmac
- mac: Specifies one or more MAC addresses to remove. Supported formats 00:01:12:23:34:45 or 00-01-12-23-34-45 or 000112233445

Optional In Parameter:

- ssid: Default is global. Provide the IP address used for configuring Clean Access Server to Clean Access Manager.

**Note**

If you do not use authentication by session ID, the *admin* and *passwd* functions are required. See [Authentication Requirement, page B-1](#).

Out Parameters: <!--error=msg--> comment

- Success: mesg value of 0
- Failure: one or more error strings can appear if ssid is not provided and if a MAC address cannot be deleted from more than one Clean Access Server.

Certified Devices List Operation

The *clearcertified* function deletes all of the existing entries from the Clean Access certified devices list.

Required In Parameter:

- op: clearcertified



Note

If you do not use authentication by session ID, the *admin* and *passwd* functions are required. See [Authentication Requirement, page B-1](#).

Out Parameters: <!--error=msg--> comment

- Success: mesg value of 0
- Failure: error string

User Operations

This section contains the following user management functions:

- [kickuser, page B-5](#)
- [kickuserbymac, page B-6](#)
- [kickoobuser, page B-6](#)
- [queryuserstime, page B-6](#)
- [renewuserstime, page B-7](#)
- [changeuserrole, page B-7](#)
- [changeloggedinuserrole, page B-8](#)
- [getlocaluserlist, page B-8](#)
- [addlocaluser, page B-9](#)
- [deletelocaluser, page B-9](#)

kickuser

The *kickuser* function terminates the active session of one or more currently logged-in in-band users.

Required In Parameters:

- op: kickuser
- ip: Specifies one IP address or a comma separated list of IP addresses.

**Note**

If you do not use authentication by session ID, the *admin* and *passwd* functions are required. See [Authentication Requirement, page B-1](#).

Out Parameters: <!--error=mesg--> comment

- Success: mesg value of 0
- Failure: error string

kickuserbymac

The *kickuserbymac* function terminates the active session by MAC address to one or more logged-in in-band users.

Required In Parameters:

- op: kickuserbymac
- mac: Specifies one MAC address or a comma separated list of MAC addresses.

**Note**

If you do not use authentication by session ID, the *admin* and *passwd* functions are required. See [Authentication Requirement, page B-1](#).

Out Parameters: <!--error=mesg--> comment

- Success: mesg value of 0
- Failure: error string

kickoobuser

The *kickoobuser* function terminates the active session of one or more out-of-band users.

Required In Parameters:

- op: kickoobuser
- mac: Specifies a MAC address or a comma separated list of MAC addresses.

**Note**

If you do not use authentication by session ID, the *admin* and *passwd* functions are required. See [Authentication Requirement, page B-1](#).

Out Parameters: <!--error=mesg--> comment

- Success: mesg value of 0
- Failure: error string

queryuserstime

The *queryuserstime* function queries the remaining session time for logged-in users. This function returns a list of logged-in users in roles with configured session timeouts.

Required In Parameters:

- op: queryuserstime



Note

If you do not use authentication by session ID, the *admin* and *passwd* functions are required. See [Authentication Requirement, page B-1](#).

Out Parameters: <!--error=msg--> comment

- Success: msg value of 0; another <!--list=iplist--> comment with an IP list and session time remaining for each IP entry
- Failure: error string

renewuserstime

Renew logged in users session timeout by a session.

Required In Parameters:

- op: renewuserstime
- list: Specifies a comma-separated list of IP addresses. Supported format: 10.1.10.10, 10.1.10.11, 10.1.10.12



Note

If you do not use authentication by session ID, the *admin* and *passwd* functions are required. See [Authentication Requirement, page B-1](#).

Out Parameters: <!--error=msg--> comment

- Success: msg value of 0
- Failure: error string

changeuserrole

The *changeuserrole* function changes in-band user access permissions prior to login by removing the user from the online users role and adding the user's MAC address to device filters list with a new role.

Required In Parameters:

- op: changeuserrole
- ip: Specifies the IP address of a user who is logged in.
- role: Specifies the role to which the user is to be moved.



Note

If you do not use authentication by session ID, the *admin* and *passwd* functions are required. See [Authentication Requirement, page B-1](#).

Out Parameters: <!--error=msg--> comment

- Success: msg value of 0
- Failure: error string

changeloggedinuserrole

The *changeloggedinuserrole* function changes access permissions for a logged-in in-band user by changing that user's current role to a new one.

Required In Parameters:

- op: changeloggedinuserrole
- ip: Specifies the IP address of a logged-in user. To specify multiple users, use a comma-separated IP list.
- role: Specifies a new role for the user.



Note

If you do not use authentication by session ID, the *admin* and *passwd* functions are required. See [Authentication Requirement, page B-1](#).

Out Parameters: <!--error=msg--> comment

- Success: msg value of 0
- Failure: error string

Guest Access Operations

This section describes the following functions:

- [getlocaluserlist, page B-8](#)
- [addlocaluser, page B-9](#)
- [deletelocaluser, page B-9](#)

These API functions allow administrators to create, delete, and view local user accounts on the CAM (local users are those internally validated by the CAM as opposed to an external authentication server). These APIs are intended to support guest access for dynamic token user access generation, providing the ability to:

- Use a webpage to access Cisco NAC Appliance API to insert a visitor username/password combination, such as *jdoh@visitor.com/jdoh112805*, and then assign a role, such as *guest1day*.
- Delete all guest users associated with the guest access role for that day.
- List all usernames associated with the guest access role.

These APIs support most implementations of guest user access dynamic token/password generation and allow the removal of those users for a guest role.

You must create the front-end generation password/token. For accounting purposes, Cisco NAC Appliance provides RADIUS accounting functionality only.

getlocaluserlist

The *getlocaluserlist* function returns a list of local user accounts with user name and role name.

Required In Parameters:

- op: getlocaluserlist

**Note**

If you do not use authentication by session ID, the *admin* and *passwd* functions are required. See [Authentication Requirement, page B-1](#).

Out Parameters: <!--error=msg--> comment

- Success: msg value of 0; <!--count=10--> shows the number of users returned and is followed by same number of comments of form <!--NAME=jdoe,ROLE=Student-->
- Failure: error string

addlocaluser

The *addlocaluser* function adds a new local user account.

Required In Parameters:

- op: addlocaluser
- username: Specifies a new local user account user name.
- userpass: Specifies the user password for the new local user account.
- userrole: Specifies the role for the new local user account.

**Note**

If you do not use authentication by session ID, the *admin* and *passwd* functions are required. See [Authentication Requirement, page B-1](#).

Out Parameters: <!--error=msg--> comment

- Success: msg value of 0
- Failure: error string

deletelocaluser

The *deletelocaluser* function deletes one or all local user accounts.

Required In Parameters:

- op: deletelocaluser
- qtype: Specifies the data type: 'name' or 'all'
- qval: Specifies the exact username in single quotes or an empty string '' to indicate "all."

**Note**

If you do not use authentication by session ID, the *admin* and *passwd* functions are required. See [Authentication Requirement, page B-1](#).

Out Parameters: <!--error=msg--> comment

- Success: msg value of 0
- Failure: error string

Report Operations

You can create scripts to compile lists of information or reports with the following report functions:

- [getuserinfo](#), page B-10
- [getoobuserinfo](#), page B-11
- [getcleanuserinfo](#), page B-11
- [getreports](#), page B-11

getuserinfo

Given an IP address, MAC address, or username, the *getuserinfo* function retrieves the following user information:

- *IP* in IPv4 format
- MAC address
- *Name* is the username
- *Provider* can be the LDAP server
- *Role* is the current role assigned to the user
- *Origrole* is the original role assigned to the user
- *VLAN* is the original VLAN tag
- *NEWVLAN* is the current VLAN tag
- Operating system of the user's system

If multiple users match the criteria, the system returns a list of users. If you enter “all” as the *qtype* parameter, all information for all users is retrieved.

Required In Parameters:

- *op*: *getuserinfo*
- *qtype*: Specifies one of the following strings: *ip*, *mac*, *name*, or *all*.
- *qual*: Specifies an IP address, MAC address, or username depending on the *qtype* parameter; enter an empty string (‘’) to indicate “all.”



Note

If you do not use authentication by session ID, the *admin* and *passwd* functions are required. See [Authentication Requirement, page B-1](#).

Out Parameters: `<!--error=msg-->` comment

- Success: *msg* value of 0; `<!--count=10-->` shows the number of users returned and is followed by a corresponding number of comments
`<!--IP=10.1.10.12,MAC=0A:13:07:9B:82:60,NAME=jdoe,PROVIDER=LDAP
Server,ROLE=Student,ORIGROLE=Student,VLAN=1024,NEWVLAN=1024,OS=Windows XP-->`
- Failure: error string

getoobuserinfo

Given an IP address, MAC address or username, the *getoobuserinfo* function retrieves information about the logged-in out-of-band (OOB) users, or given the *qtype* “all”, the system generates a list of information about all logged-in OOB users. If multiple users match the criteria, the system generates a list of users.

Required In Parameters:

- *op*: getoobuserinfo
- *qtype*: Specifies the method of identifying one or more users: ip, mac, name, all.
- *qval*: Specifies an IP or MAC address or a username; enter an empty string (‘’) to indicate “all”.



Note

If you do not use authentication by session ID, the *admin* and *passwd* functions are required. See [Authentication Requirement, page B-1](#).

Out Parameters: <!--error=msg--> comment

- Success: msg value of 0; <!--count=10--> shows the number of users returned and is followed by a matching number of comments of form
 <!--IP=10.1.10.12,MAC=0A:13:07:9B:82:60,NAME=jdoe,PROVIDER=LDAP
 Server,ROLE=Student,AUTHVLAN=10,ACCESSVLAN=1024,OS=Windows
 XP,SWITCHIP=10.1.10.1,PORTNUM=18-->
- Failure: error string

getcleanuserinfo

Given a MAC address or username, the *getcleanuserinfo* function returns information about certified users. If there are multiple users matching the criteria, the system generates a list of certified users.

Required In Parameters:

- *op*: getcleanuserinfo
- *qtype*: Specifies the method of identifying the user: mac, name, all.
- *qval*: Specifies MAC address or username; enter an empty string (‘’) to indicate “all.”

Out Parameters: <!--error=msg--> comment

- Success: msg value of 0; <!--count=10--> shows the number of users returned and is followed by a matching number of comments of form
 <!--MAC=0A:13:07:9B:82:60,NAME=jdoe,PROVIDER=LDAP
 Server,ROLE=Student,VLAN=10-->
- Failure: error string

getreports

The *getreports* function returns a report that contains customized content. You can also use this function to compile a list of users with certain software installed.

Required In Parameters:

op: getreports

**Note**

If you do not use authentication by session ID, the *admin* and *passwd* functions are required. See [Authentication Requirement, page B-1](#).

Optional Query Parameters:

[Table B-1](#) lists the query Parameters for the *getreports* function.

Table B-1 Query Parameters for the *getreports* function

Parameter Name	Allowed Values	Description
status	One of the following values: <ul style="list-style-type: none"> any (default) success failure 	Reports only information with a specified status.
user	A string; empty single quotes is the default	Reports information about the specified user.
userKey	A string; empty single quotes is the default	Reports information containing this user key.
ip	One valid IPv4 address, such as 10.20.30.40; empty single quotes is the default	Reports information about the specified IP address.
mac	One valid MAC address, such as 00:01:12:23:34:45; empty single quotes is the default	Reports information about the specified MAC address.
os	One of the following values: <ul style="list-style-type: none"> To indicate any OS, enter empty single quotes (") (default) WINDOWS_VISTA_ALL (Windows Vista) WINDOWS_VISTA_HOME_BASIC (Windows Vista Home Basic) WINDOWS_VISTA_BUSINESS (Windows Vista Business) WINDOWS_VISTA_ULTIMATE (Windows Vista Ultimate) WINDOWS_VISTA_ENTERPRISE (Windows Vista Enterprise) WINDOWS_XP (Windows XP) WINDOWS_PRO_XP (Windows XP Pro/Home) WINDOWS_TPC_XP (Windows XP Tablet PC Edition) WINDOWS_MCE_XP (Windows XP Media Center Edition) WINDOWS_2K (Windows 2000) WINDOWS_ME (Windows ME) WINDOWS_98 (Windows 98) 	Reports information about the specified OS.

Table B-1 Query Parameters for the *getreports* function (continued)

Parameter Name	Allowed Values	Description
timeRange	timeFrom, timeTo <ul style="list-style-type: none"> • timeFrom can be one of the following values: <ul style="list-style-type: none"> – timestamp (format: yyyy-mm-dd hh:mm:ss) – negative integer representing the number of hours before now – past • timeTo can be one of the following values: <ul style="list-style-type: none"> – timestamp (format: yyyy-mm-dd hh:mm:ss) – negative integer representing the number of hours before now – now – -48, -24 (the day before last) – -24, now (within last day) – 2007-01-01 00:00:00, 2007-02-28 23:59:59 (Between Jan 1st and Feb 28th) Default: past, now (any time: all possible reports)	Reports information collected within the specified time range.
showText	One of the following values: <ul style="list-style-type: none"> • true— Returns the text. • false—Does not return the text. (default) 	Indicates whether or not to return the report text.
orderBy	One of the following values: <ul style="list-style-type: none"> • user • userKey • ip • mac • os • time (default) 	Specifies the report organization.
orderDir	One of the following values: <ul style="list-style-type: none"> • asc—Indicates ascending order. (default) • desc—Indicates descending order. 	Specifies ascending or descending order for the data.
instSoft	One of the following values: <ul style="list-style-type: none"> • Empty single quotes (‘’) indicates “any” (default) • AV—Indicates AntiVirus installed • AS—Indicates AntiSpyware installed • UNKNOWN AV/AS—Indicates an unknown AV/AS 	Restricts to reports containing this type of installed software.

Table B-1 Query Parameters for the *getreports* function (continued)

Parameter Name	Allowed Values	Description
reqName	Name of the AV or AS software requirement; empty quotes “any” (default)	Restricts to reports containing this software requirement.
reqStatus	One of the following values: <ul style="list-style-type: none"> any (default) success failure 	Restricts to reports where the software requirement is of this status (only if reqName is used).

Out Parameters: <!--error=msg--> comment

- Success: msg value of 0; <!--count=count--> shows the number of reports returned; the reports follow the count comment and are of the form:
<!--status=status,user=user,userKey=userKey,ip=ip,mac=mac,os=os,time=time,text=text-->
- Failure: error string