
OneFabric Connect WebServices

Table of Contents

OF Connect WebServices.....1

NetSight Device WebService..... 2

NAC WebService..... 30

NAC Configuration WebService..... 169

OneFabric Connect WebServices

Sources

Name	Description
NetSight Device WebService	Web Service used to manipulate devices in the NSDEVICES table in the database. The URL is: https://<netsightserverip>:8443/axis/services/NetSightDeviceWebService
NAC WebService	This web service provides support for external integration with NAC services in NetSight. This service provides methods for obtaining information about end-systems. There are also methods for managing the appearance of end-systems in Rule Components such as End System Groups and User Groups. There are also methods for managing custom information associated with an end-system. In addition, there are some methods provided for creating Local Users in the Local User database, and for creating Registered Users and Registered Devices. The URL is: https://<netsightserverip>:8443/axis/services/NACWebService
NAC Configuration WebService	Web Service for performing NAC Configuration related operations. Web Service URL - https://<netsightserverip>:8443/axis/services/NACConfigurationWebService

NetSight Device WebService Web Service

Description

Web Service used to manipulate devices in the NSDEVICES table in the database.

The URL is: <https://<netsightserverip>:8443/axis/services/NetSightDeviceWebService>

See Also

[Methods](#) | [Complex Types](#)

Methods: NetSight Device WebService

Methods

Name	Description
addAuthCredential	Add telnet/SSH authentication credential
addAuthCredentialEx	Add telnet/SSH authentication credential, extended output
addCredentialEx	Add SNMP authentication credential, extended output.
addDeviceEx	Add Device to NetSight database, extended output
addProfileEx	Add access profile, extended output
deleteDeviceByIpEx	Delete device by IP address, extended output
exportDevicesAsNgf	Export device database in NGF format.
getAllDevices	Lists all devices in database
getDeviceByIpAddressEx	get device information by IP address, extended output
getSnmpCredentialAsNgf	get snmp credentials used to query a device defined in database in NGF output
importDevicesAsNgfEx	import devices' database in NGF format, extended output
isIpV6Enabled	True if Netsight is configured for IPv6
isNetSnmpEnabled	True if Netsight is using the NetSNMP stack
updateAuthCredential	Update existing telnet/SSH credentials
updateAuthCredentialEx	Update existing telnet/SSH credentials, extended output
updateCredential	Update existing SNMP credentials
updateCredentialEx	Update existing SNMP credentials
updateDevicesEx	Update existing device, extended output
updateProfile	Update existing access profile
updateProfileEx	Update existing access profile, extended output

Method: addAuthCredential

Description

Add telnet/SSH authentication credential

Action

urn:addAuthCredential

Style

Document

Input (Literal)

The input of this method is the argument addAuthCredential having the structure defined by the following table.

Input (Literal)

Name	Type	Occurs	Description
username	string	0..1	Username for telnet/SSH access
description	string	0..1	Textual description of the profile
loginPassword	string	0..1	Password to login a telnet/SSH session
enablePassword	string	0..1	Password to access enable mode in telnet/SSH
configurationPassword	string	0..1	Password to enable configuration mode in telnet or SSH access
type	string	0..1	Type of protocol that will use these credentials Telnet/SSH. Case sensitive. Use <i>Telnet</i> if these credentials will be used in telnet connections and <i>SSH</i> if the credentials will be used in SSH connections.

Output (Literal)

The output of this method is the argument `addAuthCredentialResponse` having the structure defined by the following table.

Name	Type	Occurs	Description
return	int	0..1	0 - This error code indicates success. 1 - This error indicates that the requested object does not exist. 2 - This error indicates that the action cannot be performed because the object already exists. 3 - This error indicates that a parameter value is invalid. 4 - This error code indicates an error parsing an input string. 5 - This error code indicates that the result would be an invalid configuration. 6 - This error code is used to report an error using a remote connection. 7 - This error code is a catch-all for an unexpected error condition. 8 - This error code is used to report the group parameter does not exist 9 - A generic CSV operation error If the credential already exists, will fail with a errorCode 2. To modify an existing credential use the <code>updateAuthCredential</code> instead

Method: addAuthCredentialEx**Description**

Add telnet/SSH authentication credential, extended output

Action

urn:addAuthCredentialEx

Style

Document

Input (Literal)

The input of this method is the argument addAuthCredentialEx having the structure defined by the following table.

Name	Type	Occurs	Description
username	string	0..1	Username for telnet/SSH access
description	string	0..1	Textual description of the profile (64char)
loginPassword	string	0..1	Password to login a telnet/SSH session (up to 32 characters)
enablePassword	string	0..1	Password to access enable mode in telnet/SSH (up to 32 characters)
configurationPassword	string	0..1	Password to enable configuration mode in telnet or SSH access (up to 32 characters)
type	string	0..1	Type of protocol to use these credentials telnet/SSH

Output (Literal)

The output of this method is the argument addAuthCredentialExResponse having the structure defined by the following table.

Name	Type	Occurs	Description
return	NsWsResult	0..1	A NsWsResult type (see complex types for structure) with the result of the operation. If the credential already exists, will fail with a errorCode 2. To modify an existing credential use the updateAuthCredentialEx instead

Method: addCredentialEx**Description**

Add SNMP authentication credential, extended output.

Action

urn:addCredentialEx

Style

Document

Input (Literal)

The input of this method is the argument addCredentialEx having the structure defined by the following table.

Name	Type	Occurs	Description
name	string	0..1	Name for the credentials set
snmpVersion	int	0..1	Choice of SNMP v1/2c/3
communityName	string	0..1	Community name if SNMPv1/2c is being used
userName	string	0..1	Username if SNMPv3 is being used
authPassword	string	0..1	Authentication password if SNMPv3 is being used
authType	string	0..1	Authentication type if SNMPv3 is being used. Empty if no snmpv3 is being used or any of MD5 or SHA1.
privPassword	string	0..1	SNMPv3 privacy password if SNMPv3 is being used.
privType	string	0..1	SNMPv3 privacy type if SNMPv3 is being used. Empty if no snmpv3 or no authType is defined or any of DES or AES.

Output (Literal)

The output of this method is the argument addCredentialExResponse having the structure defined by the following table.

Name	Type	Occurs	Description
return	NsWsResult	0..1	A NsWsResult type (see complex types for structure) with the result of the operation. If the credential already exists, will fail with a errorCode 2. To modify an existing credential use the updateCredentialEx instead

Method: addDeviceEx**Description**

Add Device to NetSight database, extended output

Action

urn:addDeviceEx

Style

Document

Input (Literal)

The input of this method is the argument addDeviceEx having the structure defined by the following table.

Input (Literal)

Name	Type	Occurs	Description
ipAddress	string	0..1	IP address of the device to add.
profileName	string	0..1	Name of the access profile that must be used to poll the device. Name as created by addProfileEx.
snmpContext	string	0..1	An SNMP context is a collection of MIB objects, often associated with a network entity. The SNMP context lets you access a subset of MIB objects related to that context. Console lets you specify a SNMP Context for both SNMPv1/v2 and SNMPv3. Or empty for no Context.
nickName	string	0..1	Common name to use for the device. Or Empty for no name.

Output (Literal)

The output of this method is the argument addDeviceExResponse having the structure defined by the following table.

Name	Type	Occurs	Description
return	NsWsResult	0..1	A NsWsResult type (see complex types for structure) with the result of the operation.

Method: addProfileEx**Description**

Add access profile, extended output

Action

urn:addProfileEx

Style

Document

Input (Literal)

The input of this method is the argument addProfileEx having the structure defined by the following table.

Name	Type	Occurs	Description
name	string	0..1	Name for the profile
snmpVersion	int	0..1	Choice of SNMP v1/2c/3
read	string	0..1	SNMP read configuration credentials name as created by addCredentialEx

Input (Literal)

Name	Type	Occurs	Description
write	string	0..1	SNMP write configuration credentials name as created by addCredentialEx
maxAccess	string	0..1	Credentials configuration to use maximum access mode to the device. Name as created by addCredentialEx
auth	string	0..1	Telnet/SSH authentication credentials used in this profile. name as created by addAuthCredentialEx or addAuthCredential

Output (Literal)

The output of this method is the argument addProfileExResponse having the structure defined by the following table.

Name	Type	Occurs	Description
return	NsWsResult	0..1	A NsWsResult type (see complex types for structure) with the result of the operation.

Remarks

The snmpVersion variable must match that of the credentials supplied.

Method: deleteDeviceByIpEx**Description**

Delete device by IP address, extended output

Action

urn:deleteDeviceByIpEx

Style

Document

Input (Literal)

The input of this method is the argument deleteDeviceByIpEx having the structure defined by the following table.

Name	Type	Occurs	Description
ipAddress	string	0..1	

Output (Literal)

The output of this method is the argument deleteDeviceByIpExResponse having the structure defined by the following table.

Name	Type	Occurs	Description
return	NsWsResult	0..1	A NsWsResult type (see complex types for structure) with the result of the operation.

Method: exportDevicesAsNgf**Description**

Export device database in NGF format.

Action

urn:exportDevicesAsNgf

Style

Document

Input (Literal)

The inputs of this method are the arguments defined by the following table.

Name	Type	Occurs	Description
------	------	--------	-------------

Output (Literal)

The outputs of this method are the arguments defined by the following table.

Name	Type	Occurs	Description
return	string	0..1	

Remarks

Information about the NGF format can be found in your Netsight installation at the URL:

https://<serverIp>:8443/Clients/help/content/console/docs/c_ht_export-import_dev_list.html

SNMPv1/v2

NGF lets you import a file that defines device name and SNMPv1/v2 security parameters for the device models being created by the import. The following parameters, separated by spaces, can be specified on each line within the import file to define a device. The minimum definition contains a device name (*dev=IP address*).

The following attributes are currently supported:

Remarks

Attribute	Description	Valid Parameters
dev	Device IP Address (mandatory)	<IP address>
ro	Read-Only Community Name (optional) (see Note)	<community name>
rw	Read-Write Community Name (optional) (see Note)	<community name>
su	Super-User Community Name (optional) (see Note)	<community name>
mt	The poll type (monitor type) defined for the device (optional)	0 (Not Polled), 1 (Ping), 2 (SNMP)
pg	The poll group defined for the device (optional)	1, 2, or 3
cliDesc	A description of the CLI credential (optional)	<description>
cliUsername	The username used for device access (optional)	<username>
cliType	The communication protocol used for the connection (optional)	Telnet or SSH
snmp	The SNMP protocol version for the credential (optional)	v1, v2, or v3

SNMPv1/v2 access information consists of the read only, read write, and super user community names for devices. The device name or IP address is the only required information. The string of information for each device must be on a separate line with no line breaks in the string. If you create a device list without community names, devices will be imported into Console using the Default SNMPv1 profile defined in the [Authorization/Device Access Window - Profiles/Credentials Tab](#).

SNMPv3

SNMPv3 access information can consist of the following settings. The device name or IP address and the user name is the only required information. Each device must be on a separate line.

Attribute	Description	Valid Parameters	Co-requisite Attributes
dev	Device IP Address (mandatory)	<IP address>	NA
user	User (optional)	<username> ⁽¹⁾	NA
		NoAuthNoPriv	NA
seclvl	Security Level (optional)	AuthNoPriv	authtype, authpwd
		AuthPriv	authtype, authpwd, privtype, privpwd
authtype	Authentication Type (optional)	MD5, SHA1	seclvl, authpwd

Remarks

Attribute	Description	Valid Parameters	Co-requisite Attributes
authpwd	Authentication Password (optional)	<password> ⁽¹⁾	seclevel, authtype
privtype	Privacy Type (optional)	DES	seclevel, privpwd, authtype, authpwd
privpwd	Privacy Password (optional)	<password> ⁽¹⁾	seclevel, privtype, authtype, authpwd

⁽¹⁾ Although SNMPv3 supports user names and passwords containing spaces, the NetSight Generated Format does not. When spaces occur in the user names and passwords in a .ngf file they are interpreted as a delimiter between parameters.

Examples:

dev=Switch1 ro=public rw=public su=public

dev=172.16.30.40 ro=public rw=public su=public

dev=10.20.77.127 mt=2 pg=1 ro=public rw=public su=public cliDesc=Default cliUsername=admin

cliType=Telnet snmp=v1

dev=172.16.17.18 ro=public rw=private

dev=172.16.17.38 user=netmgr seclevel=AuthPriv authtype=MD5 authpwd=net_mgr.pwd privtype=DES

privpwd=secret.pwd

Method: getAllDevices**Description**

Lists all devices in database

Action

urn:getAllDevices

Style

Document

Input (Literal)

The inputs of this method are the arguments defined by the following table.

Name	Type	Occurs	Description
------	------	--------	-------------

Output (Literal)

The outputs of this method are the arguments defined by the following table.

Name	Type	Occurs	Description
return	WsDeviceListResult	0..1	

Method: getDeviceByIpAddressEx

Description

get device information by IP address, extended output

Action

urn:getDeviceByIpAddressEx

Style

Document

Input (Literal)

The input of this method is the argument getDeviceByIpAddressEx having the structure defined by the following table.

Name	Type	Occurs	Description
ipAddress	string	0..1	IP address of the device to query.

Output (Literal)

The output of this method is the argument getDeviceByIpAddressExResponse having the structure defined by the following table.

Name	Type	Occurs	Description
return	WsDeviceListResult	0..1	a wsDeviceListResult (see complex values definition) with basic SNMP data of teh device.

Method: getSnpCredentialAsNgf**Description**

get snmp credentials used to query a device defined in database in NGF output

Action

urn:getSnpCredentialAsNgf

Style

Document

Input (Literal)

The input of this method is the argument getSnpCredentialAsNgf having the structure defined by the following table.

Name	Type	Occurs	Description
ipAddress	string	0..1	IP Address of the device

Output (Literal)

The output of this method is the argument `getSnmpCredentialAsNgfResponse` having the structure defined by the following table.

Name	Type	Occurs	Description
return	string	0..1	An string with the required paramenets, e.g.: ro=public rw=public su=public snmp=v1

Remarks

Information about the NGF format can be found in your Netsight installation at the URL:

https://<serverIp>:8443/Clients/help/content/console/docs/c_ht_export-import_dev_list.html

SNMPv1/v2

NGF lets you import a file that defines device name and SNMPv1/v2 security parameters for the device models being created by the import. The following parameters, separated by spaces, can be specified on each line within the import file to define a device. The minimum definition contains a device name (*dev=IP address*).

The following attributes are currently supported:

Attribute	Description	Valid Parameters
dev	Device IP Address (mandatory)	<IP address>
ro	Read-Only Community Name (optional) (see Note)	<community name>
rw	Read-Write Community Name (optional) (see Note)	<community name>
su	Super-User Community Name (optional) (see Note)	<community name>
mt	The poll type (monitor type) defined for the device (optional)	0 (Not Polled), 1 (Ping), 2 (SNMP)
pg	The poll group defined for the device (optional)	1, 2, or 3
cliDesc	A description of the CLI credential (optional)	<description>
cliUsername	The username used for device access (optional)	<username>
cliType	The communication protocol used for the connection (optional)	Telnet or SSH
snmp	The SNMP protocol version for the credential (optional)	v1, v2, or v3

SNMPv1/v2 access information consists of the read only, read write, and super user community names for devices. The device name or IP address is the only required information. The string of information for each device must be on a separate line with no line breaks in the string. If you create a device list without community names, devices will be imported into Console using the Default SNMPv1 profile defined in the Authorization/Device Access Window - Profiles/Credentials Tab.

Remarks**SNMPv3**

SNMPv3 access information can consist of the following settings. The device name or IP address and the user name is the only required information. Each device must be on a separate line.

Attribute	Description	Valid Parameters	Co-requisite Attributes
dev	Device IP Address (mandatory)	<IP address>	NA
user	User (optional)	<username> ⁽¹⁾	NA
		NoAuthNoPriv	NA
seclvl	Security Level (optional)	AuthNoPriv	authtype, authpwd
		AuthPriv	authtype, authpwd, privtype, privpwd
authtype	Authentication Type (optional)	MD5, SHA1	seclvl, authpwd
authpwd	Authentication Password (optional)	<password> ⁽¹⁾	seclvl, authtype
privtype	Privacy Type (optional)	DES	seclvl, privpwd, authtype, authpwd
privpwd	Privacy Password (optional)	<password> ⁽¹⁾	seclvl, privtype, authtype, authpwd

⁽¹⁾Although SNMPv3 supports user names and passwords containing spaces, the NetSight Generated Format does not. When spaces occur in the user names and passwords in a .ngf file they are interpreted as a delimiter between parameters.

Examples:

dev=Switch1 ro=public rw=public su=public

dev=172.16.30.40 ro=public rw=public su=public

dev=10.20.77.127 mt=2 pg=1 ro=public rw=public su=public cliDesc=Default cliUsername=admin cliType=Telnet snmp=v1

dev=172.16.17.18 ro=public rw=private

dev=172.16.17.38 user=netmgr seclvl=AuthPriv authtype=MD5 authpwd=net_mgr.pwd privtype=DES privpwd=secret.pwd

Method: importDevicesAsNgfEx**Description**

import devices' database in NGF format, extended output

Action

urn:importDevicesAsNgfEx

Style

Document

Input (Literal)

The input of this method is the argument `importDevicesAsNgfEx` having the structure defined by the following table.

Name	Type	Occurs	Description
ngfDevices	string	0..1	String in NGF format containing the devices and SNMP credentials.

Output (Literal)

The output of this method is the argument `importDevicesAsNgfExResponse` having the structure defined by the following table.

Name	Type	Occurs	Description
return	NsWsResult	0..1	A NsWsResult type (see complex types for structure) with the result of the operation.

Remarks

Information about the NGF format can be found in your Netsight installation at the URL:

https://<serverIp>:8443/Clients/help/content/console/docs/c_ht_export-import_dev_list.html

SNMPv1/v2

NGF lets you import a file that defines device name and SNMPv1/v2 security parameters for the device models being created by the import. The following parameters, separated by spaces, can be specified on each line within the import file to define a device. The minimum definition contains a device name (`dev=IP address`).

The following attributes are currently supported:

Attribute	Description	Valid Parameters
dev	Device IP Address (mandatory)	<IP address>
ro	Read-Only Community Name (optional) (see Note)	<community name>
rw	Read-Write Community Name (optional) (see Note)	<community name>
su	Super-User Community Name (optional) (see Note)	<community name>
mt	The poll type (monitor type) defined for the device (optional)	0 (Not Polled), 1 (Ping), 2 (SNMP)
pg	The poll group defined for the device (optional)	1, 2, or 3

Remarks

Attribute	Description	Valid Parameters
cliDesc	A description of the CLI credential (optional)	<description>
cliUsername	The username used for device access (optional)	<username>
cliType	The communication protocol used for the connection (optional)	Telnet or SSH
snmp	The SNMP protocol version for the credential (optional)	v1, v2, or v3

SNMPv1/v2 access information consists of the read only, read write, and super user community names for devices. The device name or IP address is the only required information. The string of information for each device must be on a separate line with no line breaks in the string. If you create a device list without community names, devices will be imported into Console using the Default SNMPv1 profile defined in the Authorization/Device Access Window - Profiles/Credentials Tab.

SNMPv3

SNMPv3 access information can consist of the following settings. The device name or IP address and the user name is the only required information. Each device must be on a separate line.

Attribute	Description	Valid Parameters	Co-requisite Attributes
dev	Device IP Address (mandatory)	<IP address>	NA
user	User (optional)	<username> ⁽¹⁾	NA
		NoAuthNoPriv	NA
seclvl	Security Level (optional)	AuthNoPriv	authtype, authpwd
		AuthPriv	authtype, authpwd, privtype, privpwd
authtype	Authentication Type (optional)	MD5, SHA1	seclvl, authpwd
authpwd	Authentication Password (optional)	<password> ⁽¹⁾	seclvl, authtype
privtype	Privacy Type (optional)	DES	seclvl, privpwd, authtype, authpwd
privpwd	Privacy Password (optional)	<password> ⁽¹⁾	seclvl, privtype, authtype, authpwd

⁽¹⁾ Although SNMPv3 supports user names and passwords containing spaces, the NetSight Generated Format does not. When spaces occur in the user names and passwords in a .ngf file they are interpreted as a delimiter between parameters.

Examples:

dev=Switch1 ro=public rw=public su=public

dev=172.16.30.40 ro=public rw=public su=public

dev=10.20.77.127 mt=2 pg=1 ro=public rw=public su=public cliDesc=Default cliUsername=admin

Remarks

cliType=Telnet snmp=v1

dev=172.16.17.18 ro=public rw=private

dev=172.16.17.38 user=netmgr seclevel=AuthPriv authtype=MD5 authpwd=net_mgr.pwd privtype=DES
privpwd=secret.pwd

Method: isIpV6Enabled**Description**

True if Netsight is configured for IPv6

Action

urn:isIpV6Enabled

Style

Document

Input (Literal)

The inputs of this method are the arguments defined by the following table.

Name	Type	Occurs	Description
------	------	--------	-------------

Output (Literal)

The outputs of this method are the arguments defined by the following table.

Name	Type	Occurs	Description
return	boolean	0..1	

Method: isNetSnmpEnabled**Description**

True if Netsight is using the NetSNMP stack

Action

urn:isNetSnmpEnabled

Style

Document

Input (Literal)

The inputs of this method are the arguments defined by the following table.

Name	Type	Occurs	Description
------	------	--------	-------------

Output (Literal)

The outputs of this method are the arguments defined by the following table.

Name	Type	Occurs	Description
return	boolean	0..1	

Method: updateAuthCredential**Description**

Update existing telnet/SSH credentials

Action

urn:updateAuthCredential

Style

Document

Input (Literal)

The input of this method is the argument updateAuthCredential having the structure defined by the following table.

Name	Type	Occurs	Description
username	string	0..1	Username for telnet/SSH access
description	string	0..1	Textual description of the profile
loginPassword	string	0..1	Password to login a telnet/SSH session
enablePassword	string	0..1	Password to access enable mode in telnet/SSH
configurationPassword	string	0..1	Password to enable configuration mode in telnet or SSH access
type	string	0..1	Type of protocol to use these credentials telnet/SSH

Output (Literal)

The output of this method is the argument updateAuthCredentialResponse having the structure defined by the following table.

Output (Literal)

Name	Type	Occurs	Description
return	int	0..1	0 - This error code indicates success. 1 - This error indicates that the requested object does not exist. 2 - This error indicates that the action cannot be performed because the object already exists. 3 - This error indicates that a parameter value is invalid. 4 - This error code indicates an error parsing an input string. 5 - This error code indicates that the result would be an invalid configuration. 6 - This error code is used to report an error using a remote connection. 7 - This error code is a catch-all for an unexpected error condition. 8 - This error code is used to report the group parameter does not exist 9 - A generic CSV operation error

Method: updateAuthCredentialEx**Description**

Update existing telnet/SSH credentials, extended output

Action

urn:updateAuthCredentialEx

Style

Document

Input (Literal)

The input of this method is the argument updateAuthCredentialEx having the structure defined by the following table.

Name	Type	Occurs	Description
username	string	0..1	Username for telnet/SSH access
description	string	0..1	Textual description of the profile
loginPassword	string	0..1	Password to login a telnet/SSH session
enablePassword	string	0..1	Password to access enable mode in telnet/SSH
configurationPassword	string	0..1	Password to enable configuration mode in telnet or SSH access

Input (Literal)

Name	Type	Occurs	Description
type	string	0..1	Type of protocol to use these credentials telnet/SSH

Output (Literal)

The output of this method is the argument updateAuthCredentialExResponse having the structure defined by the following table.

Name	Type	Occurs	Description
return	NsWsResult	0..1	

Method: updateCredential**Description**

Update existing SNMP credentials

Action

urn:updateCredential

Style

Document

Input (Literal)

The input of this method is the argument updateCredential having the structure defined by the following table.

Name	Type	Occurs	Description
name	string	0..1	Name for the credentials set
communityName	string	0..1	Community name if SNMPv1/2c is being used
userName	string	0..1	Username for telnet/SSH access
authPassword	string	0..1	Authentication password if SNMPv3 is being used
authType	string	0..1	Authentication type if SNMPv3 is being used
privPassword	string	0..1	SNMPv3 privacy password if SNMPv3 is being used.
privType	string	0..1	SNMPv3 privacy type if SNMPv3 is being used.

Output (Literal)

The output of this method is the argument updateCredentialResponse having the structure defined by the following table.

Output (Literal)

Name	Type	Occurs	Description
return	int	0..1	0 - This error code indicates success. 1 - This error indicates that the requested object does not exist. 2 - This error indicates that the action cannot be performed because the object already exists. 3 - This error indicates that a parameter value is invalid. 4 - This error code indicates an error parsing an input string. 5 - This error code indicates that the result would be an invalid configuration. 6 - This error code is used to report an error using a remote connection. 7 - This error code is a catch-all for an unexpected error condition. 8 - This error code is used to report the group parameter does not exist 9 - A generic CSV operation error

Method: updateCredentialEx**Description**

Update existing SNMP credentials

Action

urn:updateCredentialEx

Style

Document

Input (Literal)

The input of this method is the argument updateCredentialEx having the structure defined by the following table.

Name	Type	Occurs	Description
name	string	0..1	Name for the credentials set
communityName	string	0..1	Community name if SNMPv1/2c is being used
userName	string	0..1	Authentication password if SNMPv3 is being used
authPassword	string	0..1	Authentication type if SNMPv3 is being used
authType	string	0..1	Authentication type if SNMPv3 is being used

Input (Literal)

Name	Type	Occurs	Description
privPassword	string	0..1	SNMPv3 privacy password if SNMPv3 is being used.
privType	string	0..1	SNMPv3 privacy type if SNMPv3 is being used.

Output (Literal)

The output of this method is the argument updateCredentialExResponse having the structure defined by the following table.

Name	Type	Occurs	Description
return	NsWsResult	0..1	

Method: updateDevicesEx**Description**

Update existing device, extended output

Action

urn:updateDevicesEx

Style

Document

Input (Literal)

The input of this method is the argument updateDevicesEx having the structure defined by the following table.

Name	Type	Occurs	Description
devices	string	0..*	

Output (Literal)

The output of this method is the argument updateDevicesExResponse having the structure defined by the following table.

Name	Type	Occurs	Description
return	NsWsResult	0..1	

Method: updateProfile**Description**

Update existing access profile

Action

urn:updateProfile

Style

Document

Input (Literal)

The input of this method is the argument updateProfile having the structure defined by the following table.

Name	Type	Occurs	Description
name	string	0..1	Name for the profile
read	string	0..1	SNMP read configuration credentials
write	string	0..1	SNMP write configuration credentials
maxAccess	string	0..1	Credentials configuration to use maximum access mode to the device
authCred	string	0..1	Telnet/SSH authentication credentials used in this profile

Output (Literal)

The output of this method is the argument updateProfileResponse having the structure defined by the following table.

Name	Type	Occurs	Description
return	int	0..1	0 - This error code indicates success. 1 - This error indicates that the requested object does not exist. 2 - This error indicates that the action cannot be performed because the object already exists. 3 - This error indicates that a parameter value is invalid. 4 - This error code indicates an error parsing an input string. 5 - This error code indicates that the result would be an invalid configuration. 6 - This error code is used to report an error using a remote connection. 7 - This error code is a catch-all for an unexpected error condition. 8 - This error code is used to report the group parameter does not exist 9 - A generic CSV operation error

Method: updateProfileEx

Description

Update existing access profile, extended output

Action

urn:updateProfileEx

Style

Document

Input (Literal)

The input of this method is the argument updateProfileEx having the structure defined by the following table.

Name	Type	Occurs	Description
name	string	0..1	Name for the profile
read	string	0..1	SNMP read configuration credentials
write	string	0..1	SNMP write configuration credentials
maxAccess	string	0..1	Credentials configuration to use maximum access mode to the device
authCredName	string	0..1	Telnet/SSH authentication credentials used in this profile

Output (Literal)

The output of this method is the argument updateProfileExResponse having the structure defined by the following table.

Name	Type	Occurs	Description
return	NsWsResult	0..1	

Complex Types: NetSight Device WebService

Complex Types

Name	Description
NsWsResult	NsWsResult is equivalent to WsResult in other WebServices Error codes and messages should be identical.
WsDevice	
WsDeviceListResult	

Complex Type: NsWsResult

Description

NsWsResult is equivalent to WsResult in other WebServices Error codes and messages should be identical.

Derived By

Restricting anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
errorCode	int	0..1	0 - This error code indicates success. 1 - This error indicates that the requested object does not exist. 2 - This error indicates that the action cannot be performed because the object already exists. 3 - This error indicates that a parameter value is invalid. 4 - This error code indicates an error parsing an input string. 5 - This error code indicates that the result would be an invalid configuration. 6 - This error code is used to report an error using a remote connection. 7 - This error code is a catch-all for an unexpected error condition. 8 - This error code is used to report the group parameter does not exist 9 - A generic CSV operation error
errorMessage	string	0..1	Descriptive error message.
success	boolean	0..1	True: the operation was performed without error False: There was an error in the method

Element: errorCode [type NsWsResult]

Element: errorMessage [type NsWsResult]

Element: success [type NsWsResult]


Complex Type: WsDevice

Derived By

Restricting anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
 SEQUENCE		1..1	
baseMac	string	0..1	
bootProm	string	0..1	
chassisId	string	0..1	
chassisType	string	0..1	
deviceId	long	0..1	
firmware	string	0..1	
ip	string	0..1	
monitorType	int	0..1	
nickName	string	0..1	
note	string	0..1	
pollGroup	int	0..1	
snmpContext	string	0..1	
status	int	0..1	
sysContact	string	0..1	
sysDescriptor	string	0..1	
sysLocation	string	0..1	
sysName	string	0..1	
sysObjectId	string	0..1	
timeLastUpdated	string	0..1	
userData1	string	0..1	
userData2	string	0..1	
userData3	string	0..1	
userData4	string	0..1	

Element: baseMac [type WsDevice]

Element: bootProm [type WsDevice]

Element: chassisId [type WsDevice]

Element: chassisType [type WsDevice]

Element: deviceId [type WsDevice]

Element: firmware [type WsDevice]

Element: ip [type WsDevice]

Element: monitorType [type WsDevice]

Element: nickName [type WsDevice]

Element: note [type WsDevice]

Element: pollGroup [type WsDevice]

Element: snmpContext [type WsDevice]

Element: status [type WsDevice]

Element: sysContact [type WsDevice]

Element: sysDescriptor [type WsDevice]

Element: sysLocation [type WsDevice]

Element: sysName [type WsDevice]

Element: sysObjectId [type WsDevice]

Element: timeLastUpdated [type WsDevice]

Element: userData1 [type WsDevice]

Element: userData2 [type WsDevice]

Element: userData3 [type WsDevice]

Element: userData4 [type WsDevice]

Complex Type: WsDeviceListResult

Derived By

Restricting anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
data	WsDevice	0..*	
errorCode	int	0..1	0 - This error code indicates success. 1 - This error indicates that the requested object does not exist. 2 - This error indicates that the action cannot be performed because the object already exists. 3 - This error indicates that a parameter value is invalid. 4 - This error code indicates an error parsing an input string. 5 - This error code indicates that the result would be an invalid configuration. 6 - This error code is used to report an error using a remote connection. 7 - This error code is a catch-all for an unexpected error condition. 8 - This error code is used to report the group parameter does not exist 9 - A generic CSV operation error
errorMessage	string	0..1	
success	boolean	0..1	True: the operation was performed without error False: There was an error in the method
tableTotalRecords	int	0..1	

Element: data [type WsDeviceListResult]

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
baseMac	string	0..1	
bootProm	string	0..1	
chassisId	string	0..1	
chassisType	string	0..1	
deviceId	long	0..1	
firmware	string	0..1	
ip	string	0..1	

Content Model

Component	Type	Occurs	Description
monitorType	int	0..1	
nickName	string	0..1	
note	string	0..1	
pollGroup	int	0..1	
snmpContext	string	0..1	
status	int	0..1	
sysContact	string	0..1	
sysDescriptor	string	0..1	
sysLocation	string	0..1	
sysName	string	0..1	
sysObjectId	string	0..1	
timeLastUpdated	string	0..1	
userData1	string	0..1	
userData2	string	0..1	
userData3	string	0..1	
userData4	string	0..1	

Element: errorCode [type WsDeviceListResult]

Element: errorMessage [type WsDeviceListResult]

Element: success [type WsDeviceListResult]

Element: tableTotalRecords [type WsDeviceListResult]

Methods: NAC WebService

Description

This web service provides support for external integration with NAC services in NetSight.

This service provides methods for obtaining information about end-systems. There are also methods for managing the appearance of end-systems in Rule Components such as End System Groups and User Groups. There are also methods for managing custom information associated with an end-system.

In addition, there are some methods provided for creating Local Users in the Local User database, and for creating Registered Users and Registered Devices.

The URL is: <https://<netsightserverip>:8443/axis/services/NACWebService>

Remarks

Invocation

NAC Webservices can be called from different programming languages. We will use examples in Java, PHP and REST formats.

Java

There are multiple ways to execute the NetSight web services with Java. There are free WSDL to Java tools that will generate a client stub to perform the web service calls. Using the free WSDL to Java tools are out of the scope of this document but tutorials can be found online. For our example, we will use the `RPCServiceClient` from Apache Axis2. The Apache Axis2 framework can be downloaded at: <http://axis.apache.org/axis2/java/core/download.cgi>.

Create a Service Client

```
RPCServiceClient client = new RPCServiceClient();
Authenticator authenticator = new Authenticator();
List<String> auth = new ArrayList<String>();
auth.add(Authenticator.BASIC);
authenticator.setAuthSchemes(auth);
authenticator.setUsername("root");
authenticator.setPassword("password");
authenticator.setPreemptiveAuthentication(true);
Options options = client.getOptions();
options.setProperty(HTTPConstants.AUTHENTICATE, authenticator);
```

We first create a default `RPCServiceClient`. The NetSight web services use basic authentication so an `Authenticator` is created with the authentication set to `Authenticator.BASIC`. For the examples we will be using, the NetSight server is configured with a username `root` and the password `password`.

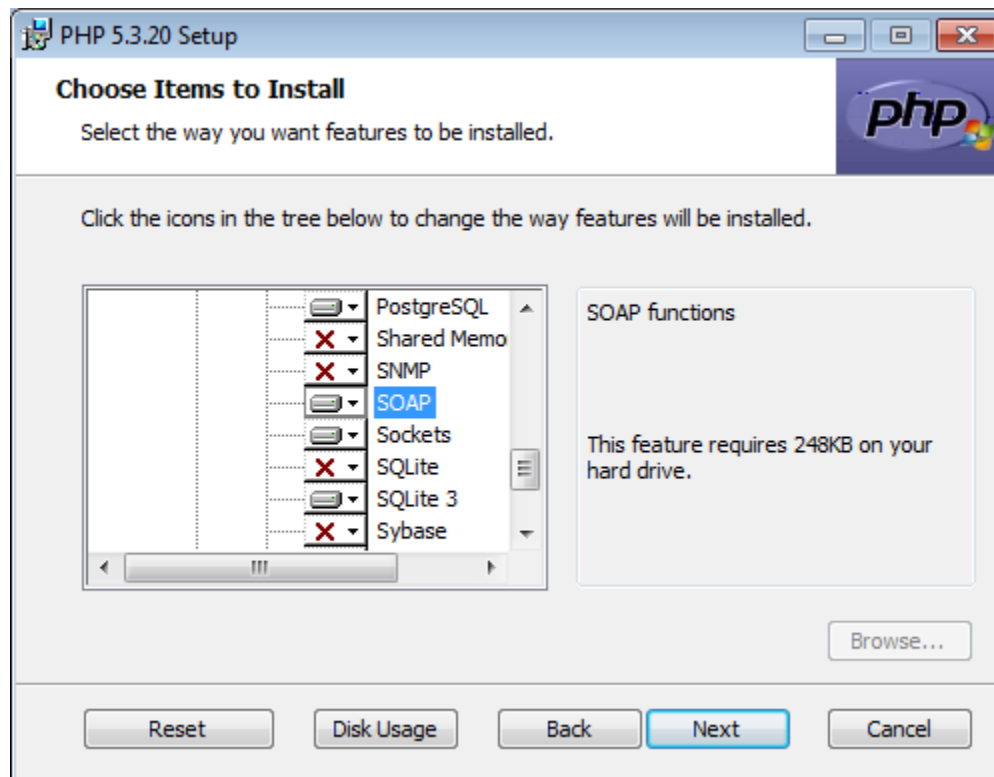
```
EndpointReference targetEPR = new
EndpointReference("https://nms.demo.com:8443/axis/services/NACWebService");
options.setTo(targetEPR);
```


Remarks

Next set the endpoint reference, in our examples we will use the NAC web service (*NACWebService*) and our NetSight server is *nms.demo.com*.

PHP

We will now cover using the NetSight web services with PHP. SOAP is required for communication to the NetSight server so be sure to install the PHP SOAP extension.



We will use the same web services that we previously covered in the Java section. The only difference is the code will be written in PHP.

Create a SOAP Client

```
$wsdl = "https://nms.demo.com:8443/axis/services/NACWebService?wsdl";
$client = new SoapClient($wsdl, array(
    "login" => "root",
    "password" => "password"
));
```

Any information on the NetSight server, web services, web service arguments, and return values were previously covered and will not be covered again. We will just cover how to execute the web services from PHP.

Web URL Call

There are 2 ways to execute a web service call through a web browser with a URL call. The call can be made

Remarks

with the username and password as part of the URL or it can be made without the username and password. Some web browsers however do not support having the username/password as part of the URL. Examples of both types of URL calls are listed below.

`https://root:password@nms.demo.com:8443/axis/services/NACWebService/getAllEndSystemMacs`

`https://nms.demo.com:8443/axis/services/NACWebService/getAllEndSystemMacs`

We will again use the same web services from our previous examples with Java and PHP. The only differences will be the web services can be executed through a web browser or with a client URL (curl). Any information on the NetSight server, web services, web service arguments, and return values were previously covered in the Java section and will not be covered again.

Method and variables are separated by the standard REST format, for example a `saveEndSystemInfoByMac` call is performed as follows:

`https://nms.demo.com:8443/axis/services/NACWebService/saveEndSystemInfoByMac?macAddress=5C:26:0A:15:52:29&custom1=myCustomOne&custom2=myCustomTwo&custom3=myCustomThree&custom4=myCustomFour`

Named lists

Named lists are containers for NAC configurations. Endsystem groups, time schedules, locations are all named lists in these webservices.

====TODO==== define format of all named lists....

See Also

[Methods](#) | [Complex Types](#)

Methods: NAC WebService

Methods

Name	Description
addHostnameToEndSystemGroup	Adds an end-system hostname to an End-System Group Optionally, the hostname is removed from other End-System Groups.
addHostnameToEndSystemGroupEx	Adds an end-system hostname to an End-System Group Optionally, the hostname is removed from other End-System Groups.
addHostnameToEndSystemGroupWithCustomDataEx	Adds an end-system hostname to an End-System Group Optionally, the hostname is removed from other End-System Groups.
addIPToEndSystemGroup	Adds an end-system IP address to an End-System Group. Optionally, the IP address is removed from other End-System Groups.
addIPToEndSystemGroupEx	Adds an end-system IP address to an End-System Group. Optionally, the IP address is removed from other End-System Groups.
addIPToEndSystemGroupWithCustomDataEx	Adds an end-system IP address to an End-System Group. Optionally, the IP address is removed from other End-System Groups.
addMACToBlacklist	Adds an end-system MAC address to the Blacklist End-System Group
addMACToBlacklistEx	Adds an end-system MAC address to the Blacklist End-System Group
addMACToBlacklistWithCustomDataEx	Adds an end-system MAC address to the Blacklist End-System Group
addMACToEndSystemGroup	Adds an end-system MAC address to an End-System Group. Optionally, the MAC address is removed from other End-System Groups.
addMACToEndSystemGroupEx	Adds an end-system MAC address to an End-System Group. Optionally, the MAC address is removed from other End-System Groups.
addMACToEndSystemGroupWithCustomDataEx	Adds an end-system MAC address to an End-System Group. Optionally, the MAC address is removed from other End-System Groups.
addUsernameToUserGroup	Adds an end-system username to a User Group Optionally, the username is removed from other User Groups.
addUsernameToUserGroupEx	Adds an end-system username to a User Group Optionally, the username is removed from other User Groups.
addValueToNamedList	Adds a value to a named list.
addValueToNamedListEx	Adds a value to a named list.
auditEnforceNacAppliances	Enforces a NAC appliance
createMacLock	Create a new Mac Lock
deleteEndSystemByMac	Deletes end-system and related configuration based on the option mask
deleteEndSystemInfoByHostname	Remove end-system information, specifying an end-system by hostname. If there is no end system data that correlates a hostname to a MAC address, then this method will fail with error code NOT_FOUND.

Methods

Name	Description
deleteEndSystemInfoByIp	Remove end-system information, specifying an end-system by IP Address. If there is no end system data that correlates an IP address to a MAC address, then this method will fail with error code NOT_FOUND.
deleteEndSystemInfoByMac	Remove end-system information, specifying an end-system by MAC Address.
deleteEndSystemInfoEx	Remove end-system information from the database.
deleteLocalUsers	Delete users from the Local User Database, specifying the users by a list of Local User IDs.
deleteLocalUsersEx	Delete users from the Local User Database, specifying the users by a list of Local User IDs.
deleteMacLock	Deletes a Mac Lock.
deleteRegisteredDevice	Removes a registered device with the matching properties from the database
deleteRegisteredDevices	Removes a set of registered devices in the database. A convenience method for
deleteRegisteredUserAndDevices	Removes a registered user with the matching properties from the database
deleteRegisteredUsers	Removes a set of registered users in the database. A convenience method for
enforceNacAppliances	Enforces a set of NAC appliances
getAllEndSystemMacs	Returns all MAC addresses for all end-systems known to NetSight/NAC. The data is returned as an array of strings.
getAllEndSystems	Returns data for all end-systems known to NetSight/NAC. The data is returned as an array of strings, where each string is a set of comma-delimited key=value pairs. If there is an error, errorCode and errorString properties will be encoded into the first element of the result.
getEndSystemAndHrByMac	Returns data for an end-system and its most recent HealthResult and Vulnerabilities, specified by MAC Address. The data is represented in XML
getEndSystemByIp	Returns data for an end-system specified by IP Address. The data is returned as a set of comma-delimited key=value pairs. If there is an error, errorCode and errorString properties will be encoded into the result.
getEndSystemByIpEx	Returns data for an end-system specified by IP Address.
getEndSystemByMac	Returns data for an end-system specified by MAC Address. The data is returned as a xml document of teh format EndSystemDTO.
getEndSystemByMacEx	Returns data for an end-system specified by MAC Address.
getEndSystemInfoArrByMac	Returns data for an end-system specified by MAC Address. The data is returned as an array of key=value pairs. If there is an error, errorCode and errorString properties will be encoded into the result.

Methods

Name	Description
getEndSystemInfoByMac	Returns data for an end-system specified by MAC Address. The data is returned as a set of comma-delimited key=value pairs. If there is an error, errorCode and errorString properties will be encoded into the result.
getEndSystemInfoByMacEx	Returns data for an end-system specified by MAC Address.
getEndSystemsByMacEx	Returns data for several end-systems specified by MAC Address. The result contains information about only those end systems that are known, and so the result list may have fewer elements than the argument list.
getExtendedEndSystemArrByMac	Returns an extended set of data for an end-system specified by MAC Address. The data includes additional information such as ELIN, portAlias, etc. The data is returned as a set of comma-delimited key=value pairs. If there is an error, errorCode and errorString properties will be encoded into the result.
getExtendedEndSystemByMac	Returns an extended set of data for an end-system specified by MAC Address. The data includes additional information such as ELIN, portAlias, etc. The data is returned as a set of comma-delimited key=value pairs. If there is an error, errorCode and errorString properties will be encoded into the result.
getNACVersion	Return the version of this interface.
getPollerStatus	Gets the success/failure of the last polling attempt
getRegisteredDevicesByMacAddress	Retrieves an array of registered devices as KEY=VALUE comma separated strings
getRegisteredUsersByUsername	Retrieves an array of registered users as KEY=VALUE comma separated strings
getRegistredDevicesByUsername	Retrieves an array of registered devices as KEY=VALUE comma separated strings
getRegistredUsersByMacAddress	Retrieves an array of registered users as KEY=VALUE comma separated strings
getUnsurfacedNamedList	Get all the Named List names and their descriptions
hashLocalUserPassword	Create a hashed password for a local user.
importEndSystemInfoEx	Save a batch of end-system information.
importEndSystemInfoFromCsv	Save a batch of end-system information provided by a CSV file. The CSV file is provided as a single string with newline characters separating lines. Each line is in the form macAddress,custom1,custom2,custom3,custom4.
processNacRequestArrFromCsv	Processes NacRequests imported from a csv file.
processNacRequestFromCsv	Processes NacRequests imported from a csv file.
reauthenticate	This method forces the selected end-system to reauthenticate. Optionally, you can also request that the end-system undergo assessment. The assessment will only take place if assessment is enabled in the configuration.

Methods

Name	Description
reauthenticateEx	This method forces the selected end-system to reauthenticate. Optionally, you can also request that the end-system undergo assessment. The assessment will only take place if assessment is enabled in the configuration.
removeHostnameFromEndSystemGroup	Removes an end-system hostname from an End-System Group
removeHostnameFromEndSystemGroupEx	Removes an end-system hostname from an End-System Group
removeIPFromEndSystemGroup	Removes an end-system IP address from an End-System Group
removeIPFromEndSystemGroupEx	Removes an end-system IP address from an End-System Group
removeMACFromBlacklist	Removes an end-system MAC address from the Blacklist End-System Group
removeMACFromBlacklistEx	Removes an end-system MAC address from the Blacklist End-System Group
removeMACFromEndSystemGroup	Removes an end-system MAC address from an End-System Group
removeMACFromEndSystemGroupEx	Removes an end-system MAC address from an End-System Group
removeUsernameFromUserGroup	Removes an end-system username from an User Group
removeUsernameFromUserGroupEx	Removes an end-system username from an User Group
removeValueFromNamedList	Removes a value from a named list.
removeValueFromNamedListEx	Removes a value from a named list.
saveEndSystemInfo	Create or update end-system information. The data is provided as a set of comma-delimited key=value pairs. The end-system can be identified by the "macAddress", "ipAddress", or "hostname" property (just one). The following properties can be specified: "custom1", "custom2", "custom3", "custom4".
saveEndSystemInfoByHostname	Create or update end-system information. The end-system is identified by hostname. If there is no end system data that correlates a hostname to a MAC address, then this method will fail with error code NOT_FOUND.
saveEndSystemInfoByIp	Create or update end-system information. The end-system is identified by IP address. If there is no end system data that correlates an IP address to a MAC address, then this method will fail with error code NOT_FOUND.
saveEndSystemInfoByMac	Create or update end-system information. The end-system is identified by MAC address.
saveEndSystemInfoEx	Create or update end-system information.
saveLocalUser	Create or update a user in the Local User Database. The local user data is provided as a set of comma-delimited key=value pairs.
saveLocalUserEx	Create or update a user in the Local User Database.

Methods

Name	Description
saveRegisteredDevice	Create a new registered device. The registered device data is provided as a as a set of comma-delimited key=value pairs.
saveRegisteredDeviceEx	Create or update a registered device for a registered user.
saveRegisteredDevices	Saves a set of devices to the database
saveRegisteredDeviceWithSponsors hip	Create a new registered device with sponsorship. The registered device data is provided as a as a set of comma-delimited key=value pairs.
saveRegisteredDeviceWithSponsors hipEx	Create or update a registered device for a registered user and then sends an email to the sponsor.
saveRegisteredUser	Create a new registered user. The registered user data is provided as a as a set of comma-delimited key=value pairs.
saveRegisteredUserEx	Create or update a registered user.
saveRegisteredUsers	Saves a set of users to the database
updateRegisteredDevice	Updates an existing registered device.The registered user data to be updated is provided as a set of comma-delimited key=value pairs.
updateRegisteredUser	Updates an existing registered user.The registered user data to be updated is provided as a set of comma-delimited key=value pairs.

Method: addHostnameToEndSystemGroup

Description

Adds an end-system hostname to an End-System Group Optionally, the hostname is removed from other End-System Groups.

Action

urn:addHostnameToEndSystemGroup

Style

Document

Input (Literal)

The input of this method is the argument addHostnameToEndSystemGroup having the structure defined by the following table.

Name	Type	Occurs	Description
endSystemGroup	string	0..1	the name of the end-system group to change
hostname	string	0..1	a hostname of an end-system

Input (Literal)

Name	Type	Occurs	Description
description	string	0..1	optional additional information stored in the end-system group with the hostname
reauthenticate	boolean	0..1	true to reauthenticate end-systems affected by this change
removeFromOtherGroups	boolean	0..1	true to remove the MAC address from other End-System Groups.

Output (Literal)

The output of this method is the argument addHostnameToEndSystemGroupResponse having the structure defined by the following table.

Name	Type	Occurs	Description
return	int	0..1	0 - This error code indicates success. 1 - This error indicates that the requested object does not exist. 2 - This error indicates that the action cannot be performed because the object already exists. 3 - This error indicates that a parameter value is invalid. 4 - This error code indicates an error parsing an input string. 5 - This error code indicates that the result would be an invalid configuration. 6 - This error code is used to report an error using a remote connection. 7 - This error code is a catch-all for an unexpected error condition. 8 - This error code is used to report the group parameter does not exist 9 - A generic CSV operation error

Method: addHostnameToEndSystemGroupEx**Description**

Adds an end-system hostname to an End-System Group Optionally, the hostname is removed from other End-System Groups.

Action

urn:addHostnameToEndSystemGroupEx

Style

Document

Input (Literal)

The input of this method is the argument addHostnameToEndSystemGroupEx having the structure defined by the following table.

Name	Type	Occurs	Description
endSystemGroup	string	0..1	the name of the end-system group to change
hostname	string	0..1	a hostname of an end-system
description	string	0..1	optional additional information stored in the end-system group with the hostname
reauthenticate	boolean	0..1	true to reauthenticate end-systems affected by this change
removeFromOtherGroups	boolean	0..1	true to remove the MAC address from other End-System Groups.

Output (Literal)

The output of this method is the argument addHostnameToEndSystemGroupExResponse having the structure defined by the following table.

Name	Type	Occurs	Description
return	WsResult	0..1	a WSResult object with extended error/success information

Method: addHostnameToEndSystemGroupWithCustomDataEx**Description**

Adds an end-system hostname to an End-System Group Optionally, the hostname is removed from other End-System Groups.

Action

urn:addHostnameToEndSystemGroupWithCustomDataEx

Style

Document

Input (Literal)

The input of this method is the argument addHostnameToEndSystemGroupWithCustomDataEx having the structure defined by the following table.

Name	Type	Occurs	Description
endSystemGroup	string	0..1	the name of the end-system group to change
hostname	string	0..1	a hostname of an end-system
description	string	0..1	optional additional information stored in the end-system group with the hostname

Input (Literal)

Name	Type	Occurs	Description
reauthenticate	boolean	0..1	true to reauthenticate end-systems affected by this change
removeFromOtherGroups	boolean	0..1	true to remove the MAC address from other End-System Groups.
custom	string	0..*	data strings to be added to endsysteminfo

Output (Literal)

The output of this method is the argument addHostnameToEndSystemGroupWithCustomDataExResponse having the structure defined by the following table.

Name	Type	Occurs	Description
return	WsResult	0..1	a WSResult object with extended error/success information

Method: addIPToEndSystemGroup**Description**

Adds an end-system IP address to an End-System Group. Optionally, the IP address is removed from other End-System Groups.

Action

urn:addIPToEndSystemGroup

Style

Document

Input (Literal)

The input of this method is the argument addIPToEndSystemGroup having the structure defined by the following table.

Name	Type	Occurs	Description
endSystemGroup	string	0..1	the name of the end-system group to change
ipAddress	string	0..1	an IP address of an end-system
description	string	0..1	optional additional information stored in the end-system group with the IP address
reauthenticate	boolean	0..1	true to reauthenticate end-systems affected by this change
removeFromOtherGroups	boolean	0..1	true to remove the MAC address from other End-System Groups.

Output (Literal)

The output of this method is the argument addIPToEndSystemGroupResponse having the structure defined by the following table.

Name	Type	Occurs	Description
return	int	0..1	0 - This error code indicates success. 1 - This error indicates that the requested object does not exist. 2 - This error indicates that the action cannot be performed because the object already exists. 3 - This error indicates that a parameter value is invalid. 4 - This error code indicates an error parsing an input string. 5 - This error code indicates that the result would be an invalid configuration. 6 - This error code is used to report an error using a remote connection. 7 - This error code is a catch-all for an unexpected error condition. 8 - This error code is used to report the group parameter does not exist 9 - A generic CSV operation error

Method: addIPToEndSystemGroupEx**Description**

Adds an end-system IP address to an End-System Group. Optionally, the IP address is removed from other End-System Groups.

Action

urn:addIPToEndSystemGroupEx

Style

Document

Input (Literal)

The input of this method is the argument addIPToEndSystemGroupEx having the structure defined by the following table.

Name	Type	Occurs	Description
endSystemGroup	string	0..1	the name of the end-system group to change
ipAddress	string	0..1	an IP address of an end-system
description	string	0..1	optional additional information stored in the end-system group with the IP address

Input (Literal)

Name	Type	Occurs	Description
reauthenticate	boolean	0..1	true to reauthenticate end-systems affected by this change
removeFromOtherGroups	boolean	0..1	true to remove the MAC address from other End-System Groups.

Output (Literal)

The output of this method is the argument addIPToEndSystemGroupExResponse having the structure defined by the following table.

Name	Type	Occurs	Description
return	WsResult	0..1	a WSResult object with extended error/success information

Method: addIPToEndSystemGroupWithCustomDataEx**Description**

Adds an end-system IP address to an End-System Group. Optionally, the IP address is removed from other End-System Groups.

Action

urn:addIPToEndSystemGroupWithCustomDataEx

Style

Document

Input (Literal)

The input of this method is the argument addIPToEndSystemGroupWithCustomDataEx having the structure defined by the following table.

Name	Type	Occurs	Description
endSystemGroup	string	0..1	the name of the end-system group to change
ipAddress	string	0..1	an IP address of an end-system
description	string	0..1	optional additional information stored in the end-system group with the IP address
reauthenticate	boolean	0..1	true to reauthenticate end-systems affected by this change
removeFromOtherGroups	boolean	0..1	true to remove the MAC address from other End-System Groups.
custom	string	0..*	data strings to be added to endsysteminfo

Output (Literal)

The output of this method is the argument addIPToEndSystemGroupWithCustomDataExResponse having the structure defined by the following table.

Name	Type	Occurs	Description
return	WsResult	0..1	a WSResult object with extended error/success information

Method: addMACToBlacklist**Description**

Adds an end-system MAC address to the Blacklist End-System Group

Action

urn:addMACToBlacklist

Style

Document

Input (Literal)

The input of this method is the argument addMACToBlacklist having the structure defined by the following table.

Name	Type	Occurs	Description
macAddress	string	0..1	a full MAC address of an end-system
description	string	0..1	optional additional information stored in the Blacklist with the MAC address
reauthenticate	boolean	0..1	true to reauthenticate end-systems affected by this change

Output (Literal)

error code

Output (Literal)

Name	Type	Occurs	Description
return	int	0..1	0 - This error code indicates success. 1 - This error indicates that the requested object does not exist. 2 - This error indicates that the action cannot be performed because the object already exists. 3 - This error indicates that a parameter value is invalid. 4 - This error code indicates an error parsing an input string. 5 - This error code indicates that the result would be an invalid configuration. 6 - This error code is used to report an error using a remote connection. 7 - This error code is a catch-all for an unexpected error condition. 8 - This error code is used to report the group parameter does not exist 9 - A generic CSV operation error

Method: addMACToBlacklistEx**Description**

Adds an end-system MAC address to the Blacklist End-System Group

Action

urn:addMACToBlacklistEx

Style

Document

Input (Literal)

The input of this method is the argument addMACToBlacklistEx having the structure defined by the following table.

Name	Type	Occurs	Description
macAddress	string	0..1	a full MAC address of an end-system
description	string	0..1	optional additional information stored in the Blacklist with the MAC address
reauthenticate	boolean	0..1	true to reauthenticate end-systems affected by this change

Output (Literal)

The output of this method is the argument addMACToBlacklistExResponse having the structure defined by the

Output (Literal)

following table.

Name	Type	Occurs	Description
return	WsResult	0..1	a WSResult object with extended error/success information

Method: addMACToBlacklistWithCustomDataEx**Description**

Adds an end-system MAC address to the Blacklist End-System Group

Action

urn:addMACToBlacklistWithCustomDataEx

Style

Document

Input (Literal)

The input of this method is the argument addMACToBlacklistWithCustomDataEx having the structure defined by the following table.

Name	Type	Occurs	Description
macAddress	string	0..1	a full MAC address of an end-system
description	string	0..1	optional additional information stored in the Blacklist with the MAC address
reauthenticate	boolean	0..1	true to reauthenticate end-systems affected by this change
custom	string	0..*	data strings to be added to endsysteminfo

Output (Literal)

The output of this method is the argument addMACToBlacklistWithCustomDataExResponse having the structure defined by the following table.

Name	Type	Occurs	Description
return	WsResult	0..1	a WSResult object with extended error/success information

Method: addMACToEndSystemGroup**Description**

Adds an end-system MAC address to an End-System Group. Optionally, the MAC address is removed from other End-System Groups.

Action

urn:addMACToEndSystemGroup

Style

Document

Input (Literal)

The input of this method is the argument addMACToEndSystemGroup having the structure defined by the following table.

Name	Type	Occurs	Description
endSystemGroup	string	0..1	the name of the end-system group to change
macAddress	string	0..1	a full MAC address of an end-system
description	string	0..1	optional additional information stored in the end-system group with the MAC address
reauthenticate	boolean	0..1	true to reauthenticate end-systems affected by this change
removeFromOtherGroups	boolean	0..1	true to remove the MAC address from other End-System Groups.

Output (Literal)

The output of this method is the argument addMACToEndSystemGroupResponse having the structure defined by the following table.

Name	Type	Occurs	Description
return	int	0..1	0 - This error code indicates success. 1 - This error indicates that the requested object does not exist. 2 - This error indicates that the action cannot be performed because the object already exists. 3 - This error indicates that a parameter value is invalid. 4 - This error code indicates an error parsing an input string. 5 - This error code indicates that the result would be an invalid configuration. 6 - This error code is used to report an error using a remote connection. 7 - This error code is a catch-all for an unexpected error condition. 8 - This error code is used to report the group parameter does not exist 9 - A generic CSV operation error

Remarks

addMACToEndSystemGroupEx Web Service

The next example will cover adding an end system to a NAC end system group.

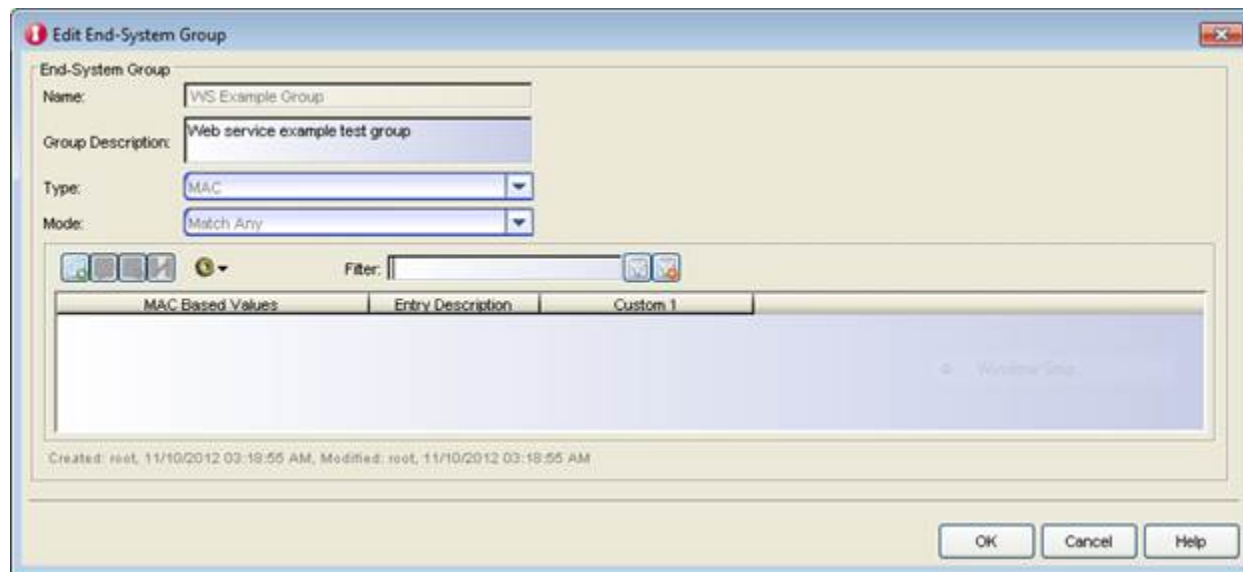
```

- <xs:element name="addMACToEndSystemGroupEx">
  - <xs:complexType>
    - <xs:sequence>
      <xs:element minOccurs="0" name="endSystemGroup" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="macAddress" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="description" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="reauthenticate" type="xs:boolean"/>
      <xs:element minOccurs="0" name="removeFromOtherGroups" type="xs:boolean"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
- <xs:element name="addMACToEndSystemGroupExResponse">
  - <xs:complexType>
    - <xs:sequence>
      <xs:element minOccurs="0" name="return" nillable="true" type="ns3:WsResult"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

From the WSDL, the *addMACToEndSystemGroupEx* web service has 5 arguments. They are the *endSystemGroup*, *macAddress*, *description*, *reauthenticate*, and *removeFromOtherGroups*. The web service returns a *WsResult* object.

For this example, we will create a MAC address based end system group and add an end system to it with the web service.



We will use 5C:26:0A:15:52:29 for this example. As you can see below, the end system is not associated to any NAC end system group.

Remarks

End-System Summary - DELL INC.:15:52:29

Details

End-System: DELL INC.:15:52:29, 192.168.10.125, ENTERAS-UD9CCDC

Activity: Last seen 11/10/2012 02:38:15 AM, First seen 11/10/2012 02:38:05 AM

Device Information: Windows (Windows Vista/ 7/ 2008)

▶ **Location** 192.168.10.10/ge.1.7 (7), Default, 192.168.30.35

▶ **Authentication Sessions** MAC

▶ **Registration** Not Registered

▶ **Miscellaneous** No MAC Lock, No IP Mapping, Not NAP Capable

▶ **Custom Information** None

▼ **Groups:**

Name	Type	Group Description	Entry Description

Add...
Remove

Close Help

Below are the complete code, output after the class has been compiled and ran, end system group configuration, and NAC end system summary.

```

RPCServiceClient client = new RPCServiceClient();
Authenticator authenticator = new Authenticator();
List<String> auth = new ArrayList<String>();
auth.add(Authenticator.BASIC);
authenticator.setAuthSchemes(auth);
authenticator.setUsername("root");
authenticator.setPassword("password");
authenticator.setPreemptiveAuthentication(true);
Options options = client.getOptions();
options.setProperty(HTTPConstants.AUTHENTICATE, authenticator);
EndpointReference targetEPR = new
EndpointReference("https://nms.demo.com:8443/axis/services/NACWebService");
options.setTo(targetEPR);
QName operation = new QName("http://ws.web.server.tam.netsight.enterasys.com",
"addMACToEndSystemGroupEx");
Object args[] = new Object[] { "WS Example Group", "5C:26:0A:15:52:29", "Test end system", true, true };
Class returnTypes[] = new Class[] { com.enterasys.netsight.tam.api.ws.WsResult.class };

```

Remarks

```
Object response[] = client.invokeBlocking(operation, args, returnTypes);
com.enterasys.netsight.tam.api.ws.WsResult value = null;
if((response != null) && (response.length > 0))
{
    value = (com.enterasys.netsight.tam.api.ws.WsResult)response[0];
    System.out.println("Success: " + Boolean.valueOf(value.isSuccess()));
}

```

Success: true

The image shows two overlapping windows from a network management application.

The top window, titled "End-System Group", contains the following fields:

- Name:** WVS Example Group
- Group Description:** Web service example test group
- Type:** MAC (selected from a dropdown)
- Mode:** Match Any (selected from a dropdown)

Below these fields is a table with the following data:

MAC Based Values	Entry Description	Custom 1
5C:26:0A:15:52:29	Test end system	

The bottom window, titled "End-System Summary - DELL INC.:15:52:29", displays details for a specific end system:

- End-System:** DELL INC.:15:52:29, 192.168.10.125, ENTERAS-UD9CCDC
- Activity:** Last seen 11/10/2012 02:38:15 AM, First seen 11/10/2012 02:38:05 AM
- Device Information:** Windows (Windows Vista/ 7/ 2008)
- Location:** 192.168.10.10/ge.1.7 (7), Default, 192.168.30.35
- Authentication Sessions:** MAC
- Registration:** Not Registered
- Miscellaneous:** No MAC Lock, No IP Mapping, Not NAP Capable
- Custom Information:** None

At the bottom of the summary window is a table showing the group assignment:

Name	Type	Group Description	Entry Description
WVS Example Group	MAC	Web service example te...	Test end system

Buttons for "Add...", "Remove", "Close", and "Help" are also visible.

Remarks**addMACToEndSystemGroupEx Web Service**

Below are the complete code and output of the PHP script.

```
$wsdl = "https://nms.demo.com:8443/axis/services/NACWebService?wsdl";
```

```
$client = new SoapClient($wsdl, array(
```

```
    "login" => "root",
```

```
    "password" => "password"
```

```
));
```

```
$response = $client->addMACToEndSystemGroupEx(array(
```

```
    "endSystemGroup" => "WS Example Group",
```

```
    "macAddress" => "5C:26:0A:15:52:29",
```

```
    "description" => "Test end system",
```

```
    "reauthenticate" => true,
```

```
    "removeFromOtherGoups" => true
```

```
));
```

```
print_r($response);
```

```
C:\Users\Administrator\Desktop>php ws.php
```

```
stdClass Object
```

```
(
```

```
    [return] => stdClass Object
```

```
    (
```

```
        [errorCode] => 0
```

```
        [errorMessage] =>
```

```
        [success] => 1
```

```
    )
```

```
)
```

Method: addMACToEndSystemGroupEx**Description**

Adds an end-system MAC address to an End-System Group. Optionally, the MAC address is removed from other End-System Groups.

Action

urn:addMACToEndSystemGroupEx

Style

Document

Input (Literal)

The input of this method is the argument addMACToEndSystemGroupEx having the structure defined by the following table.

Name	Type	Occurs	Description
endSystemGroup	string	0..1	the name of the end-system group to change
macAddress	string	0..1	a full MAC address of an end-system
description	string	0..1	optional additional information stored in the end-system group with the MAC address
reauthenticate	boolean	0..1	true to reauthenticate end-systems affected by this change
removeFromOtherGroups	boolean	0..1	true to remove the MAC address from other End-System Groups.

Output (Literal)

The output of this method is the argument addMACToEndSystemGroupExResponse having the structure defined by the following table.

Name	Type	Occurs	Description
return	WsResult	0..1	a WSResult object with extended error/success information

Method: addMACToEndSystemGroupWithCustomDataEx**Description**

Adds an end-system MAC address to an End-System Group. Optionally, the MAC address is removed from other End-System Groups.

Action

urn:addMACToEndSystemGroupWithCustomDataEx

Style

Document

Input (Literal)

The input of this method is the argument addMACToEndSystemGroupWithCustomDataEx having the structure defined by the following table.

Input (Literal)

Name	Type	Occurs	Description
endSystemGroup	string	0..1	the name of the end-system group to change
macAddress	string	0..1	a full MAC address of an end-system
description	string	0..1	optional additional information stored in the end-system group with the MAC address
reauthenticate	boolean	0..1	true to reauthenticate end-systems affected by this change
removeFromOtherGroups	boolean	0..1	true to remove the MAC address from other End-System Groups.
custom	string	0..*	data strings to be added to endsysteminfo

Output (Literal)

The output of this method is the argument addMACToEndSystemGroupWithCustomDataExResponse having the structure defined by the following table.

Name	Type	Occurs	Description
return	WsResult	0..1	a WSResult object with extended error/success information

Method: addUsernameToUserGroup**Description**

Adds an end-system username to a User Group Optionally, the username is removed from other User Groups.

Action

urn:addUsernameToUserGroup

Style

Document

Input (Literal)

The input of this method is the argument addUsernameToUserGroup having the structure defined by the following table.

Name	Type	Occurs	Description
userGroup	string	0..1	the name of the user group to change
username	string	0..1	a username of an end-system user
description	string	0..1	optional additional information stored in the end-system group with the username

Input (Literal)

Name	Type	Occurs	Description
reauthenticate	boolean	0..1	true to reauthenticate end-systems affected by this change
removeFromOtherGroups	boolean	0..1	true to remove the MAC address from other End-System Groups.

Output (Literal)

The output of this method is the argument addUsernameToUserGroupResponse having the structure defined by the following table.

Name	Type	Occurs	Description
return	int	0..1	0 - This error code indicates success. 1 - This error indicates that the requested object does not exist. 2 - This error indicates that the action cannot be performed because the object already exists. 3 - This error indicates that a parameter value is invalid. 4 - This error code indicates an error parsing an input string. 5 - This error code indicates that the result would be an invalid configuration. 6 - This error code is used to report an error using a remote connection. 7 - This error code is a catch-all for an unexpected error condition. 8 - This error code is used to report the group parameter does not exist 9 - A generic CSV operation error

Method: addUsernameToUserGroupEx**Description**

Adds an end-system username to a User Group Optionally, the username is removed from other User Groups.

Action

urn:addUsernameToUserGroupEx

Style

Document

Input (Literal)

The input of this method is the argument addUsernameToUserGroupEx having the structure defined by the following table.

Input (Literal)

Name	Type	Occurs	Description
userGroup	string	0..1	the name of the user group to change
username	string	0..1	a username of an end-system user
description	string	0..1	optional additional information stored in the end-system group with the username
reauthenticate	boolean	0..1	true to reauthenticate end-systems affected by this change
removeFromOtherGroups	boolean	0..1	true to remove the MAC address from other End-System Groups.

Output (Literal)

The output of this method is the argument addUsernameToUserGroupExResponse having the structure defined by the following table.

Name	Type	Occurs	Description
return	WsResult	0..1	a WSResult object with extended error/success information

Method: addValueToNamedList**Description**

Adds a value to a named list.

Action

urn:addValueToNamedList

Style

Document

Input (Literal)

The input of this method is the argument addValueToNamedList having the structure defined by the following table.

Name	Type	Occurs	Description
list	string	0..1	the name of the named list to change
value	string	0..1	the value to add to the named list
description	string	0..1	optional additional information stored in the named list with the value
reauthenticate	boolean	0..1	true to reauthenticate end-systems affected by this change

Output (Literal)

The output of this method is the argument `addValueToNamedListResponse` having the structure defined by the following table.

Name	Type	Occurs	Description
return	int	0..1	0 - This error code indicates success. 1 - This error indicates that the requested object does not exist. 2 - This error indicates that the action cannot be performed because the object already exists. 3 - This error indicates that a parameter value is invalid. 4 - This error code indicates an error parsing an input string. 5 - This error code indicates that the result would be an invalid configuration. 6 - This error code is used to report an error using a remote connection. 7 - This error code is a catch-all for an unexpected error condition. 8 - This error code is used to report the group parameter does not exist 9 - A generic CSV operation error

Remarks

A named list is a container for configurations in IAM. Endsystms groups, time schedules, locations, etc, are all named list.

IAM groups can be of types:

DEVICETYPE
 MAC
 TIMEOFWEEK
 USERNAME
 RADIUSUSERGROUP
 LDAPUSERGROUP
 LOCATION
 TIMEOFWEEK
 HOSTNAME
 IP
 LDAPHOSTGROUP

The Default Work Week schedule in IAM is defined as:

1/420-1080,2/420-1080,3/420-1080,4/420-1080,5/420-1080

Remarks

One entry per day with start-stop of schedule in minutes since midnight. The slash between day and time must be escaped in ws call.

Locations group entries are specified as follows:

A single source IP of the authenticator: 192.168.10.11 means all authentications from this switch or wireless controller.

Single IP and a port or port range: 192.168.10.10:ge.1.2 or 192.168.10.10:ge.1.2-24

Source Controller IP with SSID and AP specification (wildcards allowed): 192.168.30.50;*\$*, 192.168.30.50;SSID\$*, 192.168.30.50;SSID\$AP

MAC endsystem group entries are specified as follows:

MAC oui: 00:11:B2

Full MAC: 00:00:00:00:00:00

MAC and Mask: 00:09:09:09:09:09/FF:FF:FF:FF:FF:00 (MAC/MASK slash must be escaped in ws call)

IP endsystem group entries are specified as follows:

IP and mask: 192.168.0.0/255.255.0.0 (IP/MASK slash must be escaped in ws call)

User groups entries either Radius or LDAP are specified as key=value, Example:

Usergroup=302 in a radius group defines all users whose radius attribute Usegroup equals 302. IAM appliance must have the dictionaries defining the attribute Usergroup.

cn=Salem matches all users with attribute cn equal to Salem in an LDAP search.

Method: addValueToNamedListEx**Description**

Adds a value to a named list.

Action

urn:addValueToNamedListEx

Style

Document

Input (Literal)

The input of this method is the argument addValueToNamedListEx having the structure defined by the following table.

Name	Type	Occurs	Description
list	string	0..1	the name of the named list to change

Input (Literal)

Name	Type	Occurs	Description
value	string	0..1	the value to add to the named list
description	string	0..1	optional additional information stored in the named list with the value
reauthenticate	boolean	0..1	true to reauthenticate end-systems affected by this change

Output (Literal)

The output of this method is the argument addValueToNamedListExResponse having the structure defined by the following table.

Name	Type	Occurs	Description
return	WsResult	0..1	a WsResult object with extended error/success information

Remarks

A named list is a container for configurations in IAM. Endsystems groups, time schedules, locations, etc, are all named list.

IAM groups can be of types:

DEVICETYPE
 MAC
 TIMEOFWEEK
 USERNAME
 RADIUSUSERGROUP
 LDAPUSERGROUP
 LOCATION
 TIMEOFWEEK
 HOSTNAME
 IP
 LDAPHOSTGROUP

The Default Work Week schedule in IAM is defined as:

1/420-1080,2/420-1080,3/420-1080,4/420-1080,5/420-1080

One entry per day with start-stop of schedule in minutes since midnight. The slash between day and time must be escaped in ws call.

Locations group entries are specified as follows:

Remarks

A single source IP of the authenticator: 192.168.10.11 means all authentications from this switch or wireless controller.

Single IP and a port or port range: 192.168.10.10:ge.1.2 or 192.168.10.10:ge.1.2-24

Source Controller IP with SSID and AP specification (wildcards allowed): 192.168.30.50;*\$*, 192.168.30.50;SSID\$*, 192.168.30.50;SSID\$AP

MAC endsystem group entries are specified as follows:

MAC oui: 00:11:B2

Full MAC: 00:00:00:00:00:00

MAC and Mask: 00:09:09:09:09:09/FF:FF:FF:FF:FF:00 (MAC/MASK slash must be escaped in ws call)

IP endsystem group entries are specified as follows:

IP and mask: 192.168.0.0/255.255.0.0 (IP/MASK slash must be escaped in ws call)

User groups entries either Radius or LDAP are specified as key=value, Example:

Usergroup=302 in a radius group defines all users whose radius attribute Usegroup equals 302. IAM appliance must have the dictionaries defining the attribute Usergroup.

cn=Salem matches all users with attribute cn equal to Salem in an LDAP search.

Method: auditEnforceNacAppliances**Description**

Enforces a NAC appliance

Action

urn:auditEnforceNacAppliances

Style

Document

Input (Literal)

The input of this method is the argument auditEnforceNacAppliances having the structure defined by the following table.

Name	Type	Occurs	Description
nacAppliances	string	0..*	array of nac appliances to enforce

Output (Literal)

The output of this method is the argument auditEnforceNacAppliancesResponse having the structure defined

Output (Literal)

by the following table.

Name	Type	Occurs	Description
return	WsEnforceResult	0..1	a WsEnforceResult object with extended error/success information

Method: createMacLock**Description**

Create a new Mac Lock

Action

urn:createMacLock

Style

Document

Input (Literal)

The input of this method is the argument createMacLock having the structure defined by the following table.

Name	Type	Occurs	Description
mac	string	0..1	a full MAC address of an end-system
switchIp	string	0..1	
switchPort	string	0..1	
reject	boolean	0..1	a true value will reject the authentication request if the end-system tries to authenticate on a different switch/ port
policy	string	0..1	policy that should be applied if the the end-system tries to authenticate on a different switch/ port

Output (Literal)

The output of this method is the argument createMacLockResponse having the structure defined by the following table.

Output (Literal)

Name	Type	Occurs	Description
return	int	0..1	0 - This error code indicates success. 1 - This error indicates that the requested object does not exist. 2 - This error indicates that the action cannot be performed because the object already exists. 3 - This error indicates that a parameter value is invalid. 4 - This error code indicates an error parsing an input string. 5 - This error code indicates that the result would be an invalid configuration. 6 - This error code is used to report an error using a remote connection. 7 - This error code is a catch-all for an unexpected error condition. 8 - This error code is used to report the group parameter does not exist 9 - A generic CSV operation error

Method: deleteEndSystemByMac**Description**

Deletes end-system and related configuration based on the option mask

Action

urn:deleteEndSystemByMac

Style

Document

Input (Literal)

The input of this method is the argument deleteEndSystemByMac having the structure defined by the following table.

Name	Type	Occurs	Description
mac	string	0..1	a full MAC address of an end-system

Input (Literal)

Name	Type	Occurs	Description
deleteOptionsMask	int	0..1	the following set of values apply: 0x01 - delete values in named lists 0x02 - delete mac locks 0x04 - delete end-system infos 0x08 - delete registered devices 0x10 - delete force delete of end system no matter what errors occur.

Output (Literal)

The output of this method is the argument deleteEndSystemByMacResponse having the structure defined by the following table.

Name	Type	Occurs	Description
return	WsResult	0..1	a WSResult object with extended error/success information

Method: deleteEndSystemInfoByHostname**Description**

Remove end-system information, specifying an end-system by hostname.

If there is no end system data that correlates a hostname to a MAC address, then this method will fail with error code NOT_FOUND.

Action

urn:deleteEndSystemInfoByHostname

Style

Document

Input (Literal)

The input of this method is the argument deleteEndSystemInfoByHostname having the structure defined by the following table.

Name	Type	Occurs	Description
hostname	string	0..1	a hostname of an end-system

Output (Literal)

The output of this method is the argument deleteEndSystemInfoByHostnameResponse having the structure

Output (Literal)

defined by the following table.

Name	Type	Occurs	Description
return	int	0..1	0 - This error code indicates success. 1 - This error indicates that the requested object does not exist. 2 - This error indicates that the action cannot be performed because the object already exists. 3 - This error indicates that a parameter value is invalid. 4 - This error code indicates an error parsing an input string. 5 - This error code indicates that the result would be an invalid configuration. 6 - This error code is used to report an error using a remote connection. 7 - This error code is a catch-all for an unexpected error condition. 8 - This error code is used to report the group parameter does not exist 9 - A generic CSV operation error

Method: deleteEndSystemInfoByIp**Description**

Remove end-system information, specifying an end-system by IP Address.

If there is no end system data that correlates an IP address to a MAC address, then this method will fail with error code NOT_FOUND.

Action

urn:deleteEndSystemInfoByIp

Style

Document

Input (Literal)

The input of this method is the argument deleteEndSystemInfoByIp having the structure defined by the following table.

Name	Type	Occurs	Description
ipAddress	string	0..1	an IP address of an end-system

Output (Literal)

The output of this method is the argument deleteEndSystemInfoByIpResponse having the structure defined by

Output (Literal)

the following table.

Name	Type	Occurs	Description
return	int	0..1	0 - This error code indicates success. 1 - This error indicates that the requested object does not exist. 2 - This error indicates that the action cannot be performed because the object already exists. 3 - This error indicates that a parameter value is invalid. 4 - This error code indicates an error parsing an input string. 5 - This error code indicates that the result would be an invalid configuration. 6 - This error code is used to report an error using a remote connection. 7 - This error code is a catch-all for an unexpected error condition. 8 - This error code is used to report the group parameter does not exist 9 - A generic CSV operation error

Method: deleteEndSystemInfoByMac**Description**

Remove end-system information, specifying an end-system by MAC Address.

Action

urn:deleteEndSystemInfoByMac

Style

Document

Input (Literal)

The input of this method is the argument deleteEndSystemInfoByMac having the structure defined by the following table.

Name	Type	Occurs	Description
macAddress	string	0..1	a full MAC address of an end-system

Output (Literal)

The output of this method is the argument deleteEndSystemInfoByMacResponse having the structure defined by the following table.

Output (Literal)

Name	Type	Occurs	Description
return	int	0..1	0 - This error code indicates success. 1 - This error indicates that the requested object does not exist. 2 - This error indicates that the action cannot be performed because the object already exists. 3 - This error indicates that a parameter value is invalid. 4 - This error code indicates an error parsing an input string. 5 - This error code indicates that the result would be an invalid configuration. 6 - This error code is used to report an error using a remote connection. 7 - This error code is a catch-all for an unexpected error condition. 8 - This error code is used to report the group parameter does not exist 9 - A generic CSV operation error

Method: deleteEndSystemInfoEx**Description**

Remove end-system information from the database.

Action

urn:deleteEndSystemInfoEx

Style

Document

Input (Literal)

The input of this method is the argument deleteEndSystemInfoEx having the structure defined by the following table.

Name	Type	Occurs	Description
macAddress	string	0..1	a full MAC address of an end-system

Output (Literal)

The output of this method is the argument deleteEndSystemInfoExResponse having the structure defined by the following table.

Output (Literal)

Name	Type	Occurs	Description
return	WsEndSystemInfoResult	0..1	Integer returning result of operation

Method: deleteLocalUsers**Description**

Delete users from the Local User Database, specifying the users by a list of Local User IDs.

Action

urn:deleteLocalUsers

Style

Document

Input (Literal)

The input of this method is the argument deleteLocalUsers having the structure defined by the following table.

Name	Type	Occurs	Description
localUserIdsCSV	string	0..1	a list of LocalUserIds separated by commas
requestingUser	string	0..1	the name of the user requesting this operation.

Output (Literal)

The output of this method is the argument deleteLocalUsersResponse having the structure defined by the following table.

Output (Literal)

Name	Type	Occurs	Description
return	int	0..1	0 - This error code indicates success. 1 - This error indicates that the requested object does not exist. 2 - This error indicates that the action cannot be performed because the object already exists. 3 - This error indicates that a parameter value is invalid. 4 - This error code indicates an error parsing an input string. 5 - This error code indicates that the result would be an invalid configuration. 6 - This error code is used to report an error using a remote connection. 7 - This error code is a catch-all for an unexpected error condition. 8 - This error code is used to report the group parameter does not exist 9 - A generic CSV operation error

Method: deleteLocalUsersEx**Description**

Delete users from the Local User Database, specifying the users by a list of Local User IDs.

Action

urn:deleteLocalUsersEx

Style

Document

Input (Literal)

The input of this method is the argument deleteLocalUsersEx having the structure defined by the following table.

Name	Type	Occurs	Description
localUserIdsCSV	string	0..1	a list of LocalUserIds separated by commas
requestingUser	string	0..1	the name of the user requesting this operation.

Output (Literal)

The output of this method is the argument deleteLocalUsersExResponse having the structure defined by the following table.

Output (Literal)

Name	Type	Occurs	Description
return	WsResult	0..1	a WSResult object with extended error/success information

Method: deleteMacLock**Description**

Deletes a Mac Lock.

Action

urn:deleteMacLock

Style

Document

Input (Literal)

The input of this method is the argument deleteMacLock having the structure defined by the following table.

Name	Type	Occurs	Description
mac	string	0..1	a full MAC address of an end-system

Output (Literal)

The output of this method is the argument deleteMacLockResponse having the structure defined by the following table.

Name	Type	Occurs	Description
return	int	0..1	0 - This error code indicates success. 1 - This error indicates that the requested object does not exist. 2 - This error indicates that the action cannot be performed because the object already exists. 3 - This error indicates that a parameter value is invalid. 4 - This error code indicates an error parsing an input string. 5 - This error code indicates that the result would be an invalid configuration. 6 - This error code is used to report an error using a remote connection. 7 - This error code is a catch-all for an unexpected error condition. 8 - This error code is used to report the group parameter does not exist 9 - A generic CSV operation error

Method: deleteRegisteredDevice**Description**

Removes a registered device with the matching properties from the database

Action

urn:deleteRegisteredDevice

Style

Document

Input (Literal)

The input of this method is the argument deleteRegisteredDevice having the structure defined by the following table.

Name	Type	Occurs	Description
propString	string	0..1	The properties string used to delete the device
requestingUser	string	0..1	the name of the user requesting this operation.

Output (Literal)

The output of this method is the argument deleteRegisteredDeviceResponse having the structure defined by the following table.

Name	Type	Occurs	Description
return	int	0..1	0 - This error code indicates success. 1 - This error indicates that the requested object does not exist. 2 - This error indicates that the action cannot be performed because the object already exists. 3 - This error indicates that a parameter value is invalid. 4 - This error code indicates an error parsing an input string. 5 - This error code indicates that the result would be an invalid configuration. 6 - This error code is used to report an error using a remote connection. 7 - This error code is a catch-all for an unexpected error condition. 8 - This error code is used to report the group parameter does not exist 9 - A generic CSV operation error

Method: deleteRegisteredDevices

Description

Removes a set of registered devices in the database. A convenience method for

Action

urn:deleteRegisteredDevices

Style

Document

Input (Literal)

The input of this method is the argument deleteRegisteredDevices having the structure defined by the following table.

Name	Type	Occurs	Description
propStrings	string	0..*	A List of property strings of devices to be deleted from the database
requestingUser	string	0..1	the name of the user requesting this operation.

Output (Literal)

The output of this method is the argument deleteRegisteredDevicesResponse having the structure defined by the following table.

Name	Type	Occurs	Description
return	int	0..1	0 - This error code indicates success. 1 - This error indicates that the requested object does not exist. 2 - This error indicates that the action cannot be performed because the object already exists. 3 - This error indicates that a parameter value is invalid. 4 - This error code indicates an error parsing an input string. 5 - This error code indicates that the result would be an invalid configuration. 6 - This error code is used to report an error using a remote connection. 7 - This error code is a catch-all for an unexpected error condition. 8 - This error code is used to report the group parameter does not exist 9 - A generic CSV operation error

Method: deleteRegisteredUserAndDevices

Description

Removes a registered user with the matching properties from the database

Action

urn:deleteRegisteredUserAndDevices

Style

Document

Input (Literal)

The input of this method is the argument deleteRegisteredUserAndDevices having the structure defined by the following table.

Name	Type	Occurs	Description
propString	string	0..1	The properties string used to delete the device
requestingUser	string	0..1	the name of the user requesting this operation.

Output (Literal)

The output of this method is the argument deleteRegisteredUserAndDevicesResponse having the structure defined by the following table.

Name	Type	Occurs	Description
return	int	0..1	0 - This error code indicates success. 1 - This error indicates that the requested object does not exist. 2 - This error indicates that the action cannot be performed because the object already exists. 3 - This error indicates that a parameter value is invalid. 4 - This error code indicates an error parsing an input string. 5 - This error code indicates that the result would be an invalid configuration. 6 - This error code is used to report an error using a remote connection. 7 - This error code is a catch-all for an unexpected error condition. 8 - This error code is used to report the group parameter does not exist 9 - A generic CSV operation error

Method: deleteRegisteredUsers**Description**

Removes a set of registered users in the database. A convenience method for

Action

urn:deleteRegisteredUsers

Style

Document

Input (Literal)

The input of this method is the argument deleteRegisteredUsers having the structure defined by the following table.

Name	Type	Occurs	Description
propStrings	string	0..*	A List of property strings of users to be deleted from the database
requestingUser	string	0..1	the name of the user requesting this operation.

Output (Literal)

The output of this method is the argument deleteRegisteredUsersResponse having the structure defined by the following table.

Name	Type	Occurs	Description
return	int	0..1	0 - This error code indicates success. 1 - This error indicates that the requested object does not exist. 2 - This error indicates that the action cannot be performed because the object already exists. 3 - This error indicates that a parameter value is invalid. 4 - This error code indicates an error parsing an input string. 5 - This error code indicates that the result would be an invalid configuration. 6 - This error code is used to report an error using a remote connection. 7 - This error code is a catch-all for an unexpected error condition. 8 - This error code is used to report the group parameter does not exist 9 - A generic CSV operation error

Method: enforceNacAppliances**Description**

Enforces a set of NAC appliances

Action

urn:enforceNacAppliances

Style

Document

Input (Literal)

The input of this method is the argument enforceNacAppliances having the structure defined by the following table.

Name	Type	Occurs	Description
nacAppliances	string	0..*	array of nac appliances to enforce
forceMask	long	0..1	TBD
ignoreWarnings	boolean	0..1	Ignore enforce warnings and proceed.

Output (Literal)

The output of this method is the argument enforceNacAppliancesResponse having the structure defined by the following table.

Name	Type	Occurs	Description
return	WsEnforceResult	0..1	

Method: getAllEndSystemMacs**Description**

Returns all MAC addresses for all end-systems known to NetSight/NAC.

The data is returned as an array of strings.

Action

urn:getAllEndSystemMacs

Style

Document

Input (Literal)

The inputs of this method are the arguments defined by the following table.

Name	Type	Occurs	Description
------	------	--------	-------------

Output (Literal)

The outputs of this method are the arguments defined by the following table.

Name	Type	Occurs	Description
return	string	0..*	

Method: getAllEndSystems**Description**

Returns data for all end-systems known to NetSight/NAC.

The data is returned as an array of strings, where each string is a set of comma-delimited key=value pairs. If there is an error, errorCode and errorString properties will be encoded into the first element of the result.

Action

urn:getAllEndSystems

Style

Document

Input (Literal)

The inputs of this method are the arguments defined by the following table.

Name	Type	Occurs	Description
------	------	--------	-------------

Output (Literal)

The outputs of this method are the arguments defined by the following table.

Name	Type	Occurs	Description
return	string	0..*	

Method: getEndSystemAndHrByMac**Description**

Returns data for an end-system and its most recent HealthResult and Vulnerabilities, specified by MAC Address.

The data is represented in XML

Action

urn:getEndSystemAndHrByMac

Style

Document

Input (Literal)

The input of this method is the argument `getEndSystemAndHrByMac` having the structure defined by the following table.

Name	Type	Occurs	Description
macAddress	string	0..1	a full MAC address of an end-system

Output (Literal)

The output of this method is the argument `getEndSystemAndHrByMacResponse` having the structure defined by the following table.

Name	Type	Occurs	Description
return	string	0..1	

Method: getEndSystemByIp**Description**

Returns data for an end-system specified by IP Address.

The data is returned as a set of comma-delimited key=value pairs. If there is an error, `errorCode` and `errorString` properties will be encoded into the result.

Action

urn:getEndSystemByIp

Style

Document

Input (Literal)

The input of this method is the argument `getEndSystemByIp` having the structure defined by the following table.

Name	Type	Occurs	Description
ipAddress	string	0..1	an IP address of an end-system

Output (Literal)

The output of this method is the argument `getEndSystemByIpResponse` having the structure defined by the following table.

Output (Literal)

Name	Type	Occurs	Description
return	string	0..1	a text string with the database contents for the IP. Example: extendedState=NO_ERROR,nacProfileName=Student,switchIP=192.168.10.10,nacApplianceIP=192.168.30.35,switchPort=7,requestAttributes=username=student1,lastAuthEventTime=2012-11-04 07:43:09.0,locationInfo="IFNAME=ge.1.7 IFDESC=Slot: 1 1000BASE Gigabit Ethernet Frontpanel Port 7",state=ACCEPT,operatingSystemName=Windows Vista/ 7/ 2008,lastQuarantineTime=,lastAssmtHashCodeChangeTime=,lastSeenTime=2012-11-04 07:39:36.0,lastScanResultState=,ESType=Windows,lastScanTime=,regType=Authenticated Registration,macAddress=5C:26:0A:15:52:29,firstSeenTime=2012-10-22 08:19:16.0,policy="Filter-Id='Enterasys:version=1;policy=Student'",class=class com.enterasys.netsight.tam.dto.EndSystemDTO,stateDescriptor=,assmtHashCode=0,id=68,ipAddress=192.168.10.125,startAssmtWarningTime=,hostName=ENTERAS-UD9CCDC,authType=AUTH_MAC_PAP,allAuthTypes=AUTH_MAC,reason="Rule: ""Student-CP""",zone=,nacApplianceGroupName=Default,switchPortId=ge.1.7

Remarks**getEndSystemByIp Web Service**

The next Java example will call the *getEndSystemByIp* web service.

```

- <xs:element name="getEndSystemByIp">
  - <xs:complexType>
    - <xs:sequence>
      <xs:element minOccurs="0" name="ipAddress" nillable="true" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
- <xs:element name="getEndSystemByIpResponse">
  - <xs:complexType>
    - <xs:sequence>
      <xs:element minOccurs="0" name="return" nillable="true" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

From the WSDL, we can see the *getEndSystemByIp* web service has 1 argument, *ipAddress*, which is a *String*. The return value from the web service call (*getEndSystemByIpResponse*) also returns a *String*.

Remarks

For our example, NAC currently sees multiple end systems. We will retrieve the end system information for 192.168.10.125.

End-Systems							
	MAC Address	Username	IP Address	Custom 2	Switch IP	Switch Port	State
1	DELL INC. 15:52:29	student1	192.168.10.125		192.168.10.10	ge 1.7	Accept
2	APPLE, INC.:29:71:71	student1	192.168.10.116		192.168.10.10	ge 1.9	Accept
3	BC:67:78:A6:EF:54	student1	192.168.10.119		192.168.10.250	12171238235W0000 (20-...	Accept

Below are the complete code and output after the class has been compiled and ran.

```

RPCServiceClient client = new RPCServiceClient();
Authenticator authenticator = new Authenticator();
List<String> auth = new ArrayList<String>();
auth.add(Authenticator.BASIC);
authenticator.setAuthSchemes(auth);
authenticator.setUsername("root");
authenticator.setPassword("password");
authenticator.setPreemptiveAuthentication(true);
Options options = client.getOptions();
options.setProperty(HTTPConstants.AUTHENTICATE, authenticator);
EndpointReference targetEPR = new
EndpointReference("https://nms.demo.com:8443/axis/services/NACWebService");
options.setTo(targetEPR);
QName operation = new QName("http://ws.web.server.tam.netsight.enterasys.com", "getEndSystemByIp");
Object args[] = new Object[] { "192.168.10.125" };
Class returnTypes[] = new Class[] { String.class };
Object response[] = client.invokeBlocking(operation, args, returnTypes);
String es = null;
if((response != null) && (response.length > 0))
{
    es = (String)response[0];
    System.out.println("End System: " + es);
}

End System: extendedState=NO_ERROR,nacProfileName=Student,switchIP=192.168.10.10,nacApplianceIP=
92.168.30.35,switchPort=7,requestAttributes=,username=student1,lastAuthEventTime=2012-11-04
07:43:09.0,locationInfo="IFNAME=ge.1.7 IFDESC=Slot: 1 1000BASE Gigabit Ethernet Frontpanel Port
7",state=ACCEPT,operatingSystemName=Windows Vista/ 7/
2008,lastQuarantineTime=,lastAssmtHashCodeChangeTime=,lastSeenTime=2012-11-04
07:39:36.0,lastScanResultState=,ESType=Windows,lastScanTime=,regType=Authenticated
Registration,macAddress=5C:26:0A:15:52:29,firstSeenTime=2012-10-22
08:19:16.0,policy="Filter-Id='Enterasys:version=1:policy=Student'",class=class com.enterasys.netsight.tam.dto.

```

Remarks

EndSystemDTO, stateDescr=, assmtHashCode=0, id=68, ipAddress=192.168.10.125, startAssmtWarningTime=, hostName=ENTERAS-UD9CCDC, authType=AUTH_MAC_PAP, allAuthTypes=AUTH_MAC, reason="Rule: ""Student-CP""", zone=, nacApplianceGroupName=Default, switchPortId=ge.1.7

getEndSystemByIp Web Service

Below are the complete code and output of the PHP script.

```
$wsdl = "https://nms.demo.com:8443/axis/services/NACWebService?wsdl";
```

```
$client = new SoapClient($wsdl, array(
```

```
    "login" => "root",
```

```
    "password" => "password"
```

```
));
```

```
$response = $client->getEndSystemByIp(array(
```

```
    "ipAddress" => "192.168.10.125"
```

```
));
```

```
print_r($response);
```

```
C:\Users\Administrator\Desktop>php ws.php
```

```
stdClass Object
```

```
(
```

```
    [return] => extendedState=NO_ERROR, nacProfileName=Unregistered NAC Profile, switchIP=192.168.10.10,
    nacApplianceIP=192.168.30.35, switchPort=7, requestAttributes=, username=, lastAuthEventTime=2012-11-10
    02:41:11.0, locationInfo="IFNAME=ge.1.7 IFDESC=Slot: 1 1000BASE Gigabit Ethernet Frontpanel Port
    7", state=ACCEPT, operatingSystemName=Windows Vista/ 7/
    2008, lastQuarantineTime=, lastAssmtHashCodeChangeTime=, lastSeenTime=2012-11-10 02:38:15.0, lastScanR
    esultState=, EStype=Windows, lastScanTime=, regType=Transient, macAddress=5C:26:0A:15:52:29, firstSeenTi
    me=2012-11-10 02:38:05.0, policy="Filter-Id='Enterasys:version=1:policy=Unregistered'", class=class com.enter
    asys.netsight.tam.dto.EndSystemDTO, stateDescr=, assmtHashCode=0, id=69, ipAddress=192.168.10.125, startA
    ssmtWarningTime=, hostName=ENTERAS-UD9CCDC, authType=AUTH_MAC_PAP, allAuthTypes=AUTH_MAC
    , reason="Rule: ""Unregistered Guest""", zone=, nacApplianceGroupName=Default, switchPortId=ge.1.7
```

```
)
```

Method: getEndSystemByIpEx**Description**

Returns data for an end-system specified by IP Address.

Action

urn:getEndSystemByIpEx

Style

Document

Input (Literal)

The input of this method is the argument getEndSystemByIpEx having the structure defined by the following table.

Name	Type	Occurs	Description
ipAddress	string	0..1	an IP address of an end-system

Output (Literal)

The output of this method is the argument getEndSystemByIpExResponse having the structure defined by the following table.

Name	Type	Occurs	Description
return	WsEndSystemResult	0..1	

Method: getEndSystemByMac**Description**

Returns data for an end-system specified by MAC Address. The data is returned as a xml document of teh format EndSystemDTO.

Action

urn:getEndSystemByMac

Style

Document

Input (Literal)

The input of this method is the argument getEndSystemByMac having the structure defined by the following table.

Name	Type	Occurs	Description
macAddress	string	0..1	a full MAC address of an end-system

Output (Literal)

The output of this method is the argument getEndSystemByMacResponse having the structure defined by the following table.

Output (Literal)

Name	Type	Occurs	Description
return	string	0..1	<p>A comma-delimited string with the format key1=value1,key2=value2,...,keyn=valuen.</p> <p>Keys:</p> <p>extendedState=NO_ERROR</p> <p>nacProfileName=Pass Through NAC Profile</p> <p>switchIP=192.168.30.50</p> <p>nacApplianceIP=192.168.30.35</p> <p>switchPort=101</p> <p>requestAttributes=</p> <p>username=</p> <p>lastAuthEventTime=2013-03-22 17:29:47.0</p> <p>locationInfo=AP_SERIAL=11502502235K0000</p> <p>IFDESC=MDM_WLAN IFNAME=MDM_WLAN</p> <p>AP_MAC=20-B3-99-1B-D8-30</p> <p>SSID=MDM_WLAN IFALIAS=MDM_WLAN</p> <p>AP_NAME=CSA_AP_01</p> <p>state=ACCEPT</p> <p>operatingSystemName=Windows Vista/ 7/ 2008</p> <p>lastQuarantineTime=</p> <p>radiusServerIp=</p> <p>lastAssmtHashCodeChangeTime=</p> <p>lastSeenTime=2013-03-22 17:29:47.0</p> <p>lastScanResultState=</p> <p>ESType=Windows</p> <p>lastScanTime=</p> <p>regType=</p> <p>macAddress=00:1C:BF:23:61:A6</p> <p>firstSeenTime=2013-03-05 17:05:20.0</p> <p>policy=</p> <p>stateDescr=DHCP IP Rediscover</p> <p>assmtHashCode=0</p> <p>id=6</p> <p>ipAddress=192.168.31.17</p> <p>startAssmtWarningTime=</p> <p>hostName=host-WS</p> <p>authType=AUTH_MAC_PAP</p> <p>allAuthTypes=AUTH_MAC</p> <p>reason=Rule: "Bypass"</p> <p>zone=</p> <p>nacApplianceGroupName=Default</p> <p>switchPortId=CSA_AP_01</p> <p>(20-B3-99-1B-D8-30)MDM_WLAN</p>

Method: getEndSystemByMacEx**Description**

Returns data for an end-system specified by MAC Address.

Action

urn:getEndSystemByMacEx

Style

Document

Input (Literal)

The input of this method is the argument getEndSystemByMacEx having the structure defined by the following table.

Name	Type	Occurs	Description
macAddress	string	0..1	a full MAC address of an end-system

Output (Literal)

The output of this method is the argument getEndSystemByMacExResponse having the structure defined by the following table.

Name	Type	Occurs	Description
return	WsEndSystemResult	0..1	

Method: getEndSystemInfoArrByMac**Description**

Returns data for an end-system specified by MAC Address.

The data is returned as an array of key=value pairs. If there is an error, errorCode and errorString properties will be encoded into the result.

Action

urn:getEndSystemInfoArrByMac

Style

Document

Input (Literal)

The input of this method is the argument getEndSystemInfoArrByMac having the structure defined by the following table.

Name	Type	Occurs	Description
macAddress	string	0..1	Mac Address of the EndSystem

Output (Literal)

The output of this method is the argument `getEndSystemInfoArrByMacResponse` having the structure defined by the following table.

Name	Type	Occurs	Description
return	string	0..*	<p>Returns an array of key=value pairs.</p> <p>Keys:</p> <p>extendedState=NO_ERROR</p> <p>nacProfileName=Pass Through NAC Profile</p> <p>switchIP=192.168.30.50</p> <p>nacApplianceIP=192.168.30.35</p> <p>switchPort=101</p> <p>requestAttributes=</p> <p>username=</p> <p>lastAuthEventTime=2013-03-22 17:29:47.0</p> <p>locationInfo=AP_SERIAL=11502502235K0000</p> <p>IFDESC=MDM_WLAN IFNAME=MDM_WLAN</p> <p>AP_MAC=20-B3-99-1B-D8-30</p> <p>SSID=MDM_WLAN IFALIAS=MDM_WLAN</p> <p>AP_NAME=CSA_AP_01</p> <p>state=ACCEPT</p> <p>operatingSystemName=Windows Vista/ 7/ 2008</p> <p>lastQuarantineTime=</p> <p>radiusServerIp=</p> <p>lastAssmtHashCodeChangeTime=</p> <p>lastSeenTime=2013-03-22 17:29:47.0</p> <p>lastScanResultState=</p> <p>ESType=Windows</p> <p>lastScanTime=</p> <p>regType=</p> <p>macAddress=00:1C:BF:23:61:A6</p> <p>firstSeenTime=2013-03-05 17:05:20.0</p> <p>policy=</p> <p>stateDescr=DHCP IP Rediscover</p> <p>assmtHashCode=0</p> <p>id=6</p> <p>ipAddress=192.168.31.17</p> <p>startAssmtWarningTime=</p> <p>hostName=host-WS</p> <p>authType=AUTH_MAC_PAP</p> <p>allAuthTypes=AUTH_MAC</p> <p>reason=Rule: "Bypass"</p> <p>zone=</p> <p>nacApplianceGroupName=Default</p> <p>switchPortId=CSA_AP_01 (20-B3-99-1B-D8-30)MDM_WLAN</p>

Method: getEndSystemInfoByMac**Description**

Returns data for an end-system specified by MAC Address.

The data is returned as a set of comma-delimited key=value pairs. If there is an error, `errorCode` and

Description

errorString properties will be encoded into the result.

Action

urn:getEndSystemInfoByMac

Style

Document

Input (Literal)

The input of this method is the argument getEndSystemInfoByMac having the structure defined by the following table.

Name	Type	Occurs	Description
macAddress	string	0..1	a full MAC address of an end-system

Output (Literal)

The output of this method is the argument getEndSystemInfoByMacResponse having the structure defined by the following table.

Output (Literal)

Name	Type	Occurs	Description
return	string	0..1	Returns an comma-separated string of key=value pairs. Keys: extendedState=NO_ERROR nacProfileName=Pass Through NAC Profile switchIP=192.168.30.50 nacApplianceIP=192.168.30.35 switchPort=101 requestAttributes= username= lastAuthEventTime=2013-03-22 17:29:47.0 locationInfo=AP_SERIAL=11502502235K0000 IFDESC=MDM_WLAN IFNAME=MDM_WLAN AP_MAC=20-B3-99-1B-D8-30 SSID=MDM_WLAN IFALIAS=MDM_WLAN AP_NAME=CSA_AP_01 state=ACCEPT operatingSystemName=Windows Vista/ 7/ 2008 lastQuarantineTime= radiusServerIp= lastAssmtHashCodeChangeTime= lastSeenTime=2013-03-22 17:29:47.0 lastScanResultState= EType=Windows lastScanTime= regType= macAddress=00:1C:BF:23:61:A6 firstSeenTime=2013-03-05 17:05:20.0 policy= stateDescr=DHCP IP Rediscover assmtHashCode=0 id=6 ipAddress=192.168.31.17 startAssmtWarningTime= hostName=host-WS authType=AUTH_MAC_PAP allAuthTypes=AUTH_MAC reason=Rule: "Bypass" zone= nacApplianceGroupName=Default switchPortId=CSA_AP_01 (20-B3-99-1B-D8-30)MDM_WLAN

Method: getEndSystemInfoByMacEx**Description**

Returns data for an end-system specified by MAC Address.

Action

urn:getEndSystemInfoByMacEx

Style

Document

Input (Literal)

The input of this method is the argument getEndSystemInfoByMacEx having the structure defined by the following table.

Name	Type	Occurs	Description
macAddress	string	0..1	a full MAC address of an end-system

Output (Literal)

The output of this method is the argument getEndSystemInfoByMacExResponse having the structure defined by the following table.

Output (Literal)

Name	Type	Occurs	Description
return	WsEndSystemInfoResult	0..1	<p>Returns an comma-separated string of key=value pairs.</p> <p>Keys:</p> <p>extendedState=NO_ERROR</p> <p>nacProfileName=Pass Through NAC Profile</p> <p>switchIP=192.168.30.50</p> <p>nacApplianceIP=192.168.30.35</p> <p>switchPort=101</p> <p>requestAttributes=</p> <p>username=</p> <p>lastAuthEventTime=2013-03-22 17:29:47.0</p> <p>locationInfo=AP_SERIAL=11502502235K0000</p> <p>IFDESC=MDM_WLAN IFNAME=MDM_WLAN</p> <p>AP_MAC=20-B3-99-1B-D8-30</p> <p>SSID=MDM_WLAN IFALIAS=MDM_WLAN</p> <p>AP_NAME=CSA_AP_01</p> <p>state=ACCEPT</p> <p>operatingSystemName=Windows Vista/ 7/ 2008</p> <p>lastQuarantineTime=</p> <p>radiusServerIp=</p> <p>lastAssmtHashCodeChangeTime=</p> <p>lastSeenTime=2013-03-22 17:29:47.0</p> <p>lastScanResultState=</p> <p>EType=Windows</p> <p>lastScanTime=</p> <p>regType=</p> <p>macAddress=00:1C:BF:23:61:A6</p> <p>firstSeenTime=2013-03-05 17:05:20.0</p> <p>policy=</p> <p>stateDescr=DHCP IP Rediscover</p> <p>assmtHashCode=0</p> <p>id=6</p> <p>ipAddress=192.168.31.17</p> <p>startAssmtWarningTime=</p> <p>hostName=host-WS</p> <p>authType=AUTH_MAC_PAP</p> <p>allAuthTypes=AUTH_MAC</p> <p>reason=Rule: "Bypass"</p> <p>zone=</p> <p>nacApplianceGroupName=Default</p> <p>switchPortId=CSA_AP_01 (20-B3-99-1B-D8-30)MDM_WLAN</p>

Remarks

Return format is not WsEndSystemInfoResult but string

Method: getEndSystemsByMacEx

Description

Returns data for several end-systems specified by MAC Address. The result contains information about only those end systems that are known, and so the result list may have fewer elements than the argument list.

Action

urn:getEndSystemsByMacEx

Style

Document

Input (Literal)

The input of this method is the argument getEndSystemsByMacEx having the structure defined by the following table.

Name	Type	Occurs	Description
macAddresses	string	0..*	an array of full MAC address of end-systems

Output (Literal)

The output of this method is the argument getEndSystemsByMacExResponse having the structure defined by the following table.

Name	Type	Occurs	Description
return	WsEndSystemListRes ult	0..1	

Method: getExtendedEndSystemArrByMac**Description**

Returns an extended set of data for an end-system specified by MAC Address. The data includes additional information such as ELIN, portAlias, etc.

The data is returned as a set of comma-delimited key=value pairs. If there is an error, errorCode and errorString properties will be encoded into the result.

Action

urn:getExtendedEndSystemArrByMac

Style

Document

Input (Literal)

The input of this method is the argument getExtendedEndSystemArrByMac having the structure defined by the following table.

Input (Literal)

Name	Type	Occurs	Description
macAddress	string	0..1	a full MAC address of an end-system

Output (Literal)

The output of this method is the argument getExtendedEndSystemArrByMacResponse having the structure defined by the following table.

Name	Type	Occurs	Description
return	string	0..*	Returns an array of key=value pairs. Keys: extendedState=NO_ERROR nacProfileName=Pass Through NAC Profile switchIP=192.168.30.50 nacApplianceIP=192.168.30.35 switchPort=101 requestAttributes= username= lastAuthEventTime=2013-03-22 17:29:47.0 locationInfo=AP_SERIAL=11502502235K0000 IFDESC=MDM_WLAN IFNAME=MDM_WLAN AP_MAC=20-B3-99-1B-D8-30 SSID=MDM_WLAN IFALIAS=MDM_WLAN AP_NAME=CSA_AP_01 state=ACCEPT operatingSystemName=Windows Vista/ 7/ 2008 lastQuarantineTime= radiusServerIp= lastAssmtHashCodeChangeTime= lastSeenTime=2013-03-22 17:29:47.0 lastScanResultState= EType=Windows lastScanTime= regType= macAddress=00:1C:BF:23:61:A6 firstSeenTime=2013-03-05 17:05:20.0 policy= stateDescr=DHCP IP Rediscover assmtHashCode=0 id=6 ipAddress=192.168.31.17 startAssmtWarningTime= hostName=host-WS authType=AUTH_MAC_PAP allAuthTypes=AUTH_MAC reason=Rule: "Bypass" zone= nacApplianceGroupName=Default switchPortId=CSA_AP_01 (20-B3-99-1B-D8-30)MDM_WLAN

Method: getExtendedEndSystemByMac

Description

Returns an extended set of data for an end-system specified by MAC Address. The data includes additional information such as ELIN, portAlias, etc.

The data is returned as a set of comma-delimited key=value pairs. If there is an error, errorCode and errorString properties will be encoded into the result.

Action

urn:getExtendedEndSystemByMac

Style

Document

Input (Literal)

The input of this method is the argument getExtendedEndSystemByMac having the structure defined by the following table.

Name	Type	Occurs	Description
macAddress	string	0..1	a full MAC address of an end-system

Output (Literal)

The output of this method is the argument getExtendedEndSystemByMacResponse having the structure defined by the following table.

Output (Literal)

Name	Type	Occurs	Description
return	string	0..1	Returns an comma-separated string of key=value pairs. Keys: extendedState=NO_ERROR nacProfileName=Pass Through NAC Profile switchIP=192.168.30.50 nacApplianceIP=192.168.30.35 switchPort=101 requestAttributes= username= lastAuthEventTime=2013-03-22 17:29:47.0 locationInfo=AP_SERIAL=11502502235K0000 IFDESC=MDM_WLAN IFNAME=MDM_WLAN AP_MAC=20-B3-99-1B-D8-30 SSID=MDM_WLAN IFALIAS=MDM_WLAN AP_NAME=CSA_AP_01 state=ACCEPT operatingSystemName=Windows Vista/ 7/ 2008 lastQuarantineTime= radiusServerIp= lastAssmtHashCodeChangeTime= lastSeenTime=2013-03-22 17:29:47.0 lastScanResultState= EType=Windows lastScanTime= regType= macAddress=00:1C:BF:23:61:A6 firstSeenTime=2013-03-05 17:05:20.0 policy= stateDescr=DHCP IP Rediscover assmtHashCode=0 id=6 ipAddress=192.168.31.17 startAssmtWarningTime= hostName=host-WS authType=AUTH_MAC_PAP allAuthTypes=AUTH_MAC reason=Rule: "Bypass" zone= nacApplianceGroupName=Default switchPortId=CSA_AP_01 (20-B3-99-1B-D8-30)MDM_WLAN

Method: getNACVersion**Description**

Return the version of this interface.

Action

urn:getNACVersion

Style

Document

Input (Literal)

The inputs of this method are the arguments defined by the following table.

Name	Type	Occurs	Description
------	------	--------	-------------

Output (Literal)

The outputs of this method are the arguments defined by the following table.

Name	Type	Occurs	Description
return	string	0..1	The version of this interface

Remarks

This method signature never changes.

getNACVersion Web Service

QName operation = new QName("http://ws.web.server.tam.netsight.enterasys.com", "getNACVersion");

The web service we will call is *getNACVersion*, the web service's target namespace (*http://ws.web.server.tam.netsight.enterasys.com*) can be found in the WSDL, see image below.



```
Object args[] = new Object[] {};
Class returnTypes[] = new Class[] {String.class};
Object response[] = client.invokeBlocking(operation, args, returnTypes);
String version = null;
if((response != null) && (response.length > 0))
{
    version = (String)response[0];
}
```

Remarks

```
System.out.println("NAC version: " + version);
}
```

The *getNACVersion* web service requires no arguments and returns a *String*. Below are the complete code and output after the class has been compiled and ran.

```
RPCServiceClient client = new RPCServiceClient();
Authenticator authenticator = new Authenticator();
List<String> auth = new ArrayList<String>();
auth.add(Authenticator.BASIC);
authenticator.setAuthSchemes(auth);
authenticator.setUsername("root");
authenticator.setPassword("password");
authenticator.setPreemptiveAuthentication(true);
Options options = client.getOptions();
options.setProperty(HTTPConstants.AUTHENTICATE, authenticator);
EndpointReference targetEPR = new
EndpointReference("https://nms.demo.com:8443/axis/services/NACWebService");
options.setTo(targetEPR);
QName operation = new QName("http://ws.web.server.tam.netsight.enterasys.com", "getNACVersion");
Object args[] = new Object[] {};
Class returnTypes[] = new Class[] {String.class};
Object response[] = client.invokeBlocking(operation, args, returnTypes);
String version = null;
if((response != null) && (response.length > 0))
{
    version = (String)response[0];
    System.out.println("NAC version: " + version);
}
NAC version: 4.3.0.92
```

getNACVersion Web Service

Below are the complete code and output of the PHP script.

```
$wsdl = "https://nms.demo.com:8443/axis/services/NACWebService?wsdl";
$client = new SoapClient($wsdl, array(
    "login" => "root",
```

Remarks

```

        "password" => "password"
    ));
    $response = $client->getNACVersion();
    print_r($response);
    C:\Users\Administrator\Desktop>php ws.php
    stdClass Object
    (
        [return] => 4.3.0.92
    )

```

Method: getPollerStatus**Description**

Gets the success/failure of the last polling attempt

Action

urn:getPollerStatus

Style

Document

Input (Literal)

The input of this method is the argument getPollerStatus having the structure defined by the following table.

Name	Type	Occurs	Description
naclP	string	0..1	the NAC appliance to check the polling status of (IP)

Output (Literal)

The output of this method is the argument getPollerStatusResponse having the structure defined by the following table.

Name	Type	Occurs	Description
return	boolean	0..1	

Method: getRegisteredDevicesByMacAddress

Description

Retrieves an array of registered devices as KEY=VALUE comma separated strings

Action

urn:getRegisteredDevicesByMacAddress

Style

Document

Input (Literal)

The input of this method is the argument getRegisteredDevicesByMacAddress having the structure defined by the following table.

Name	Type	Occurs	Description
macAddress	string	0..1	a full MAC address of an end-system

Output (Literal)

The output of this method is the argument getRegisteredDevicesByMacAddressResponse having the structure defined by the following table.

Name	Type	Occurs	Description
return	string	0..*	Keys as defined by RegisteredDevice object

Method: getRegisteredUsersByUsername**Description**

Retrieves an array of registered users as KEY=VALUE comma separated strings

Action

urn:getRegisteredUsersByUsername

Style

Document

Input (Literal)

The input of this method is the argument getRegisteredUsersByUsername having the structure defined by the following table.

Name	Type	Occurs	Description
username	string	0..1	a username of an end-system user

Output (Literal)

The output of this method is the argument `getRegisteredUsersByUsernameResponse` having the structure defined by the following table.

Name	Type	Occurs	Description
return	string	0..*	Keys as defined by RegisteredUser Object

Method: getRegisteredDevicesByUsername**Description**

Retrieves an array of registered devices as KEY=VALUE comma separated strings

Action

urn:getRegisteredDevicesByUsername

Style

Document

Input (Literal)

The input of this method is the argument `getRegisteredDevicesByUsername` having the structure defined by the following table.

Name	Type	Occurs	Description
username	string	0..1	a username of an end-system user

Output (Literal)

The output of this method is the argument `getRegisteredDevicesByUsernameResponse` having the structure defined by the following table.

Name	Type	Occurs	Description
return	string	0..*	Keys as defined by RegisteredDevice object

Method: getRegisteredUsersByMacAddress**Description**

Retrieves an array of registered users as KEY=VALUE comma separated strings

Action

urn:getRegisteredUsersByMacAddress

Style

Document

Input (Literal)

The input of this method is the argument `getRegisteredUsersByMacAddress` having the structure defined by the following table.

Name	Type	Occurs	Description
macAddress	string	0..1	a full MAC address of an end-system

Output (Literal)

The output of this method is the argument `getRegisteredUsersByMacAddressResponse` having the structure defined by the following table.

Name	Type	Occurs	Description
return	string	0..*	Keys as defined by RegisteredUser Object

Method: getUnsurfacedNamedList**Description**

Get all the Named List names and their descriptions

Action

urn:getUnsurfacedNamedList

Style

Document

Input (Literal)

The input of this method is the argument `getUnsurfacedNamedList` having the structure defined by the following table.

Name	Type	Occurs	Description
listName	string	0..1	Name of the named list

Output (Literal)

The output of this method is the argument `getUnsurfacedNamedListResponse` having the structure defined by the following table.

Output (Literal)

Name	Type	Occurs	Description
return	string	0..*	A string [] with each row representing Named List data in name value pairs "name=value, type=value, description=value" the commas in the description are escaped by the back slash character.

Remarks

A named list is a container for configurations in IAM. Endsystems groups, time schedules, locations, etc, are all named list.

IAM groups can be of types:

DEVICETYPE
 MAC
 TIMEOFWEEK
 USERNAME
 RADIUSUSERGROUP
 LDAPUSERGROUP
 LOCATION
 TIMEOFWEEK
 HOSTNAME
 IP
 LDAPHOSTGROUP

The Default Work Week schedule in IAM is defined as:

1/420-1080,2/420-1080,3/420-1080,4/420-1080,5/420-1080

One entry per day with start-stop of schedule in minutes since midnight. The slash between day and time must be escaped in ws call.

Locations group entries are specified as follows:

A single source IP of the authenticator: 192.168.10.11 means all authentications from this switch or wireless controller.

Single IP and a port or port range: 192.168.10.10:ge.1.2 or 192.168.10.10:ge.1.2-24

Source Controller IP with SSID and AP specification (wildcards allowed): 192.168.30.50;*\$*, 192.168.30.50;SSID\$*, 192.168.30.50;SSID\$AP

MAC endsystem group entries are specified as follows:

MAC oui: 00:11:B2

Remarks

Full MAC: 00:00:00:00:00:00

MAC and Mask: 00:09:09:09:09:09/FF:FF:FF:FF:FF:00 (MAC/MASK slash must be escaped in ws call)

IP endsystem group entries are specified as follows:

IP and mask: 192.168.0.0/255.255.0.0 (IP/MASK slash must be escaped in ws call)

User groups entries either Radius or LDAP are specified as key=value, Example:

Usergroup=302 in a radius group defines all users whose radius attribute Usegroup equals 302. IAM appliance must have the dictionaries defining the attribute Usergroup.

cn=Salem matches all users with attribute cn equal to Salem in an LDAP search.

Method: hashLocalUserPassword**Description**

Create a hashed password for a local user.

Action

urn:hashLocalUserPassword

Style

Document

Input (Literal)

The input of this method is the argument hashLocalUserPassword having the structure defined by the following table.

Name	Type	Occurs	Description
password	string	0..1	the clear-text password

Output (Literal)

The output of this method is the argument hashLocalUserPasswordResponse having the structure defined by the following table.

Name	Type	Occurs	Description
return	string	0..1	

Method: importEndSystemInfoEx

Description

Save a batch of end-system information.

Action

urn:importEndSystemInfoEx

Style

Document

Input (Literal)

The input of this method is the argument importEndSystemInfoEx having the structure defined by the following table.

Name	Type	Occurs	Description
infoList	EndSystemInfo	0..*	an array of EndSystemInfo objects
isSave	boolean	0..1	true to save end-system info, false to delete it (by MAC address)

Output (Literal)

The output of this method is the argument importEndSystemInfoExResponse having the structure defined by the following table.

Name	Type	Occurs	Description
return	WsResult	0..1	

Method: importEndSystemInfoFromCsv**Description**

Save a batch of end-system information provided by a CSV file.

The CSV file is provided as a single string with newline characters separating lines. Each line is in the form macAddress,custom1,custom2,custom3,custom4.

Action

urn:importEndSystemInfoFromCsv

Style

Document

Input (Literal)

The input of this method is the argument importEndSystemInfoFromCsv having the structure defined by the following table.

Input (Literal)

Name	Type	Occurs	Description
csvData	string	0..1	A string version of the csv file with newline delimiters.
isSave	boolean	0..1	true to save end-system info, false to delete it (by MAC address)

Output (Literal)

The output of this method is the argument `importEndSystemInfoFromCsvResponse` having the structure defined by the following table.

Name	Type	Occurs	Description
return	int	0..1	0 - This error code indicates success. 1 - This error indicates that the requested object does not exist. 2 - This error indicates that the action cannot be performed because the object already exists. 3 - This error indicates that a parameter value is invalid. 4 - This error code indicates an error parsing an input string. 5 - This error code indicates that the result would be an invalid configuration. 6 - This error code is used to report an error using a remote connection. 7 - This error code is a catch-all for an unexpected error condition. 8 - This error code is used to report the group parameter does not exist 9 - A generic CSV operation error

Method: processNacRequestArrFromCsv**Description**

Processes NacRequests imported from a csv file.

Action

urn:processNacRequestArrFromCsv

Style

Document

Input (Literal)

The input of this method is the argument `processNacRequestArrFromCsv` having the structure defined by the following table.

Input (Literal)

Name	Type	Occurs	Description
csvData	string	0..*	A string version of the csv file, one command per entity
oper	string	0..1	"reauth" , "esoverride", "useroverride"
isAdd	boolean	0..1	if its an add or delete request. Ignored for reauth
type	string	0..1	

Output (Literal)

The output of this method is the argument processNacRequestArrFromCsvResponse having the structure defined by the following table.

Name	Type	Occurs	Description
return	WsResult	0..1	

Method: processNacRequestFromCsv**Description**

Processes NacRequests imported from a csv file.

Action

urn:processNacRequestFromCsv

Style

Document

Input (Literal)

The input of this method is the argument processNacRequestFromCsv having the structure defined by the following table.

Name	Type	Occurs	Description
csvData	string	0..1	A string version of the csv file created by inserting new line delimiters
oper	string	0..1	"reauth" , "esoverride", "useroverride"
isAdd	boolean	0..1	if its an add or delete request. Ignored for reauth
type	string	0..1	

Output (Literal)

The output of this method is the argument processNacRequestFromCsvResponse having the structure defined

Output (Literal)

by the following table.

Name	Type	Occurs	Description
return	int	0..1	0 - This error code indicates success. 1 - This error indicates that the requested object does not exist. 2 - This error indicates that the action cannot be performed because the object already exists. 3 - This error indicates that a parameter value is invalid. 4 - This error code indicates an error parsing an input string. 5 - This error code indicates that the result would be an invalid configuration. 6 - This error code is used to report an error using a remote connection. 7 - This error code is a catch-all for an unexpected error condition. 8 - This error code is used to report the group parameter does not exist 9 - A generic CSV operation error

Method: reauthenticate**Description**

This method forces the selected end-system to reauthenticate. Optionally, you can also request that the end-system undergo assessment. The assessment will only take place if assessment is enabled in the configuration.

Action

urn:reauthenticate

Style

Document

Input (Literal)

The input of this method is the argument reauthenticate having the structure defined by the following table.

Name	Type	Occurs	Description
macAddress	string	0..1	a full MAC address of an end-system
assess	boolean	0..1	True to trigger assessment if configured

Output (Literal)

The output of this method is the argument reauthenticateResponse having the structure defined by the following

Output (Literal)

table.

Name	Type	Occurs	Description
return	int	0..1	0 - This error code indicates success. 1 - This error indicates that the requested object does not exist. 2 - This error indicates that the action cannot be performed because the object already exists. 3 - This error indicates that a parameter value is invalid. 4 - This error code indicates an error parsing an input string. 5 - This error code indicates that the result would be an invalid configuration. 6 - This error code is used to report an error using a remote connection. 7 - This error code is a catch-all for an unexpected error condition. 8 - This error code is used to report the group parameter does not exist 9 - A generic CSV operation error

Remarks**reauthenticate Web Service**

The next example will cover reauthenticating a single end system.

```

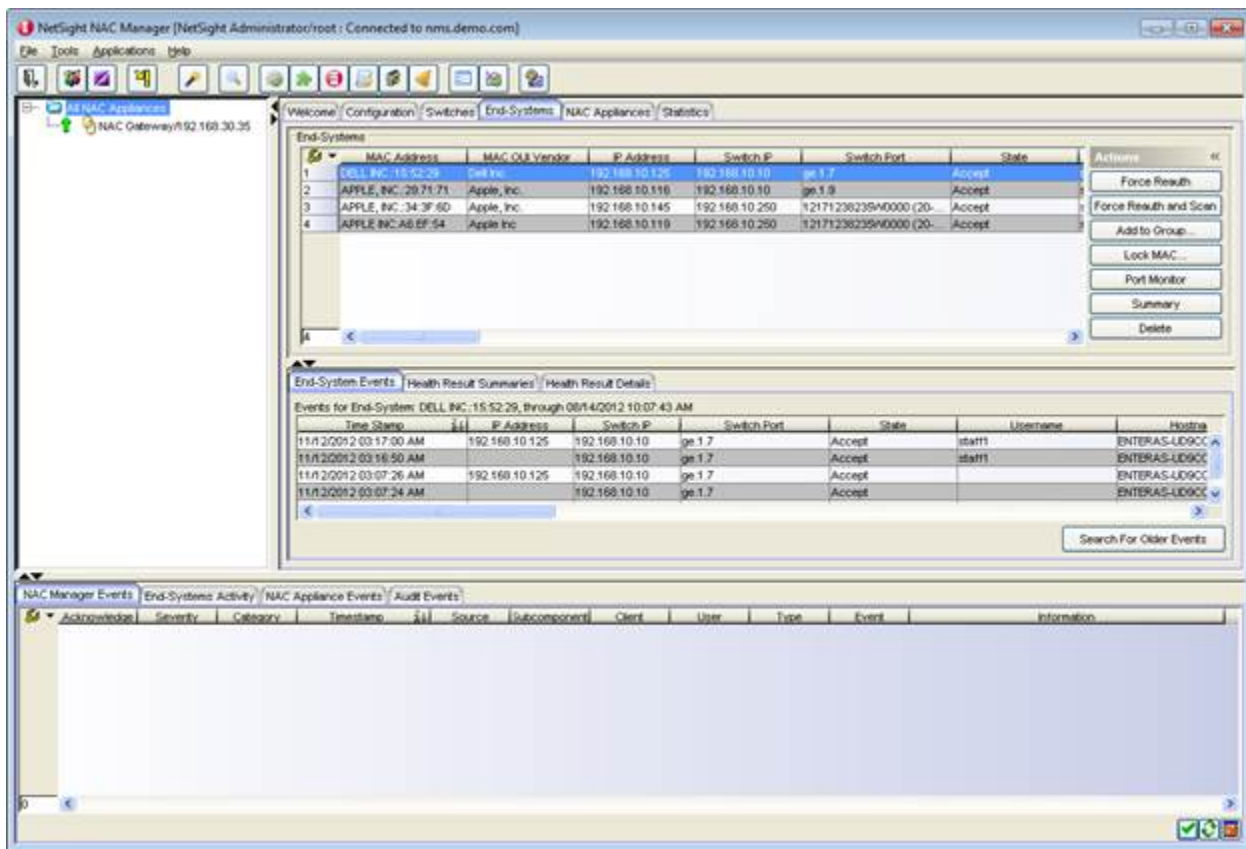
- <xs:element name="reauthenticate">
  - <xs:complexType>
    - <xs:sequence>
      <xs:element minOccurs="0" name="macAddress" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="assess" type="xs:boolean"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
- <xs:element name="reauthenticateResponse">
  - <xs:complexType>
    - <xs:sequence>
      <xs:element minOccurs="0" name="return" type="xs:int"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

From the WSDL, the *reauthenticate* web service has 2 arguments. They are the *macAddress* and *assess*. The return value for the web service is an int.

For our example, we will reauthenticate 5C:26:0A:15:52:29.

Remarks



Currently there are NAC events in NAC Manager. A NAC event will occur when an end system is reauthenticated.

Below are the complete code, output after the class has been compiled and ran, and NAC Manager screenshot.

```
RPCServiceClient client = new RPCServiceClient();
Authenticator authenticator = new Authenticator();
List<String> auth = new ArrayList<String>();
auth.add(Authenticator.BASIC);
authenticator.setAuthSchemes(auth);
authenticator.setUsername("root");
authenticator.setPassword("password");
authenticator.setPreemptiveAuthentication(true);
Options options = client.getOptions();
options.setProperty(HTTPConstants.AUTHENTICATE, authenticator);
EndpointReference targetEPR = new
EndpointReference("https://nms.demo.com:8443/axis/services/NACWebService");
options.setTo(targetEPR);
QName operation = new QName("http://ws.web.server.tam.netsight.enterasys.com", "reauthenticate");
```

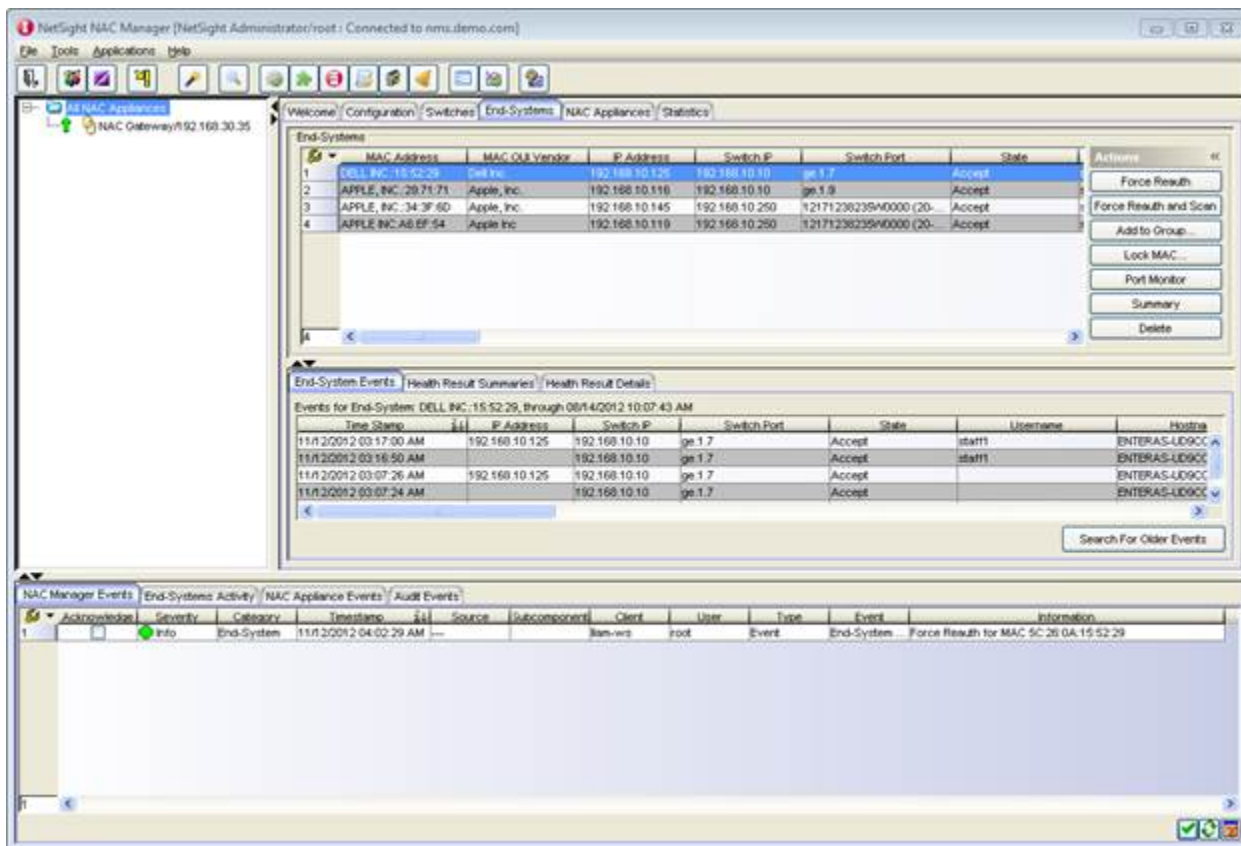
Remarks

```

Object args[] = new Object[] {"5C:26:0A:15:52:29", false};
Class returnTypes[] = new Class[] {Integer.class};
Object response[] = client.invokeBlocking(operation, args, returnTypes);
Integer value = null;
if((response != null) && (response.length > 0))
{
    value = (Integer)response[0];
    System.out.println("Return Value: " + value);
}

```

Return Value: 0

**reauthenticate Web Service**

Below are the complete code and output of the PHP script.

```
$wsdl = "https://nms.demo.com:8443/axis/services/NACWebService?wsdl";
```

```
$client = new SoapClient($wsdl, array(
```

```
    "login" => "root",
```

Remarks

```

        "password" => "password"
    ));
    $response = $client->reauthenticate(array(
        "macAddress" => "5C:26:0A:15:52:29",
        "assess" => false
    ));
    print_r($response);
C:\Users\Administrator\Desktop>php ws.php
stdClass Object
(
    [return] => 0
)

```

Method: reauthenticateEx**Description**

This method forces the selected end-system to reauthenticate. Optionally, you can also request that the end-system undergo assessment. The assessment will only take place if assessment is enabled in the configuration.

Action

urn:reauthenticateEx

Style

Document

Input (Literal)

The input of this method is the argument reauthenticateEx having the structure defined by the following table.

Name	Type	Occurs	Description
macAddress	string	0..1	a full MAC address of an end-system
assess	boolean	0..1	True to trigger assessment if configured

Output (Literal)

The output of this method is the argument reauthenticateExResponse having the structure defined by the following table.

Output (Literal)

Name	Type	Occurs	Description
return	WsResult	0..1	

Method: removeHostnameFromEndSystemGroup**Description**

Removes an end-system hostname from an End-System Group

Action

urn:removeHostnameFromEndSystemGroup

Style

Document

Input (Literal)

The input of this method is the argument removeHostnameFromEndSystemGroup having the structure defined by the following table.

Name	Type	Occurs	Description
endSystemGroup	string	0..1	the name of the end-system group to change
hostname	string	0..1	a hostname of an end-system
reauthenticate	boolean	0..1	true to reauthenticate end-systems affected by this change

Output (Literal)

The output of this method is the argument removeHostnameFromEndSystemGroupResponse having the structure defined by the following table.

Output (Literal)

Name	Type	Occurs	Description
return	int	0..1	0 - This error code indicates success. 1 - This error indicates that the requested object does not exist. 2 - This error indicates that the action cannot be performed because the object already exists. 3 - This error indicates that a parameter value is invalid. 4 - This error code indicates an error parsing an input string. 5 - This error code indicates that the result would be an invalid configuration. 6 - This error code is used to report an error using a remote connection. 7 - This error code is a catch-all for an unexpected error condition. 8 - This error code is used to report the group parameter does not exist 9 - A generic CSV operation error

Method: removeHostnameFromEndSystemGroupEx**Description**

Removes an end-system hostname from an End-System Group

Action

urn:removeHostnameFromEndSystemGroupEx

Style

Document

Input (Literal)

The input of this method is the argument removeHostnameFromEndSystemGroupEx having the structure defined by the following table.

Name	Type	Occurs	Description
endSystemGroup	string	0..1	the name of the end-system group to change
hostname	string	0..1	a hostname of an end-system
reauthenticate	boolean	0..1	true to reauthenticate end-systems affected by this change

Output (Literal)

The output of this method is the argument removeHostnameFromEndSystemGroupExResponse having the structure defined by the following table.

Output (Literal)

Name	Type	Occurs	Description
return	WsResult	0..1	

Method: removeIPFromEndSystemGroup**Description**

Removes an end-system IP address from an End-System Group

Action

urn:removeIPFromEndSystemGroup

Style

Document

Input (Literal)

The input of this method is the argument removeIPFromEndSystemGroup having the structure defined by the following table.

Name	Type	Occurs	Description
endSystemGroup	string	0..1	the name of the end-system group to change
ipAddress	string	0..1	an IP address of an end-system
reauthenticate	boolean	0..1	true to reauthenticate end-systems affected by this change

Output (Literal)

The output of this method is the argument removeIPFromEndSystemGroupResponse having the structure defined by the following table.

Output (Literal)

Name	Type	Occurs	Description
return	int	0..1	0 - This error code indicates success. 1 - This error indicates that the requested object does not exist. 2 - This error indicates that the action cannot be performed because the object already exists. 3 - This error indicates that a parameter value is invalid. 4 - This error code indicates an error parsing an input string. 5 - This error code indicates that the result would be an invalid configuration. 6 - This error code is used to report an error using a remote connection. 7 - This error code is a catch-all for an unexpected error condition. 8 - This error code is used to report the group parameter does not exist 9 - A generic CSV operation error

Method: removeIPFromEndSystemGroupEx**Description**

Removes an end-system IP address from an End-System Group

Action

urn:removeIPFromEndSystemGroupEx

Style

Document

Input (Literal)

The input of this method is the argument removeIPFromEndSystemGroupEx having the structure defined by the following table.

Name	Type	Occurs	Description
endSystemGroup	string	0..1	the name of the end-system group to change
ipAddress	string	0..1	an IP address of an end-system
reauthenticate	boolean	0..1	true to reauthenticate end-systems affected by this change

Output (Literal)

The output of this method is the argument removeIPFromEndSystemGroupExResponse having the structure defined by the following table.

Output (Literal)

Name	Type	Occurs	Description
return	WsResult	0..1	

Method: removeMACFromBlacklist**Description**

Removes an end-system MAC address from the Blacklist End-System Group

Action

urn:removeMACFromBlacklist

Style

Document

Input (Literal)

The input of this method is the argument removeMACFromBlacklist having the structure defined by the following table.

Name	Type	Occurs	Description
macAddress	string	0..1	a full MAC address of an end-system
reauthenticate	boolean	0..1	true to reauthenticate end-systems affected by this change

Output (Literal)

The output of this method is the argument removeMACFromBlacklistResponse having the structure defined by the following table.

Output (Literal)

Name	Type	Occurs	Description
return	int	0..1	0 - This error code indicates success. 1 - This error indicates that the requested object does not exist. 2 - This error indicates that the action cannot be performed because the object already exists. 3 - This error indicates that a parameter value is invalid. 4 - This error code indicates an error parsing an input string. 5 - This error code indicates that the result would be an invalid configuration. 6 - This error code is used to report an error using a remote connection. 7 - This error code is a catch-all for an unexpected error condition. 8 - This error code is used to report the group parameter does not exist 9 - A generic CSV operation error

Method: removeMACFromBlacklistEx**Description**

Removes an end-system MAC address from the Blacklist End-System Group

Action

urn:removeMACFromBlacklistEx

Style

Document

Input (Literal)

The input of this method is the argument removeMACFromBlacklistEx having the structure defined by the following table.

Name	Type	Occurs	Description
macAddress	string	0..1	a full MAC address of an end-system
reauthenticate	boolean	0..1	true to reauthenticate end-systems affected by this change

Output (Literal)

The output of this method is the argument removeMACFromBlacklistExResponse having the structure defined by the following table.

Output (Literal)

Name	Type	Occurs	Description
return	WsResult	0..1	

Method: removeMACFromEndSystemGroup**Description**

Removes an end-system MAC address from an End-System Group

Action

urn:removeMACFromEndSystemGroup

Style

Document

Input (Literal)

The input of this method is the argument removeMACFromEndSystemGroup having the structure defined by the following table.

Name	Type	Occurs	Description
endSystemGroup	string	0..1	the name of the end-system group to change
macAddress	string	0..1	a full MAC address of an end-system
reauthenticate	boolean	0..1	true to reauthenticate end-systems affected by this change

Output (Literal)

The output of this method is the argument removeMACFromEndSystemGroupResponse having the structure defined by the following table.

Output (Literal)

Name	Type	Occurs	Description
return	int	0..1	0 - This error code indicates success. 1 - This error indicates that the requested object does not exist. 2 - This error indicates that the action cannot be performed because the object already exists. 3 - This error indicates that a parameter value is invalid. 4 - This error code indicates an error parsing an input string. 5 - This error code indicates that the result would be an invalid configuration. 6 - This error code is used to report an error using a remote connection. 7 - This error code is a catch-all for an unexpected error condition. 8 - This error code is used to report the group parameter does not exist 9 - A generic CSV operation error

Method: removeMACFromEndSystemGroupEx**Description**

Removes an end-system MAC address from an End-System Group

Action

urn:removeMACFromEndSystemGroupEx

Style

Document

Input (Literal)

The input of this method is the argument removeMACFromEndSystemGroupEx having the structure defined by the following table.

Name	Type	Occurs	Description
endSystemGroup	string	0..1	the name of the end-system group to change
macAddress	string	0..1	a full MAC address of an end-system
reauthenticate	boolean	0..1	true to reauthenticate end-systems affected by this change

Output (Literal)

The output of this method is the argument removeMACFromEndSystemGroupExResponse having the structure defined by the following table.

Output (Literal)

Name	Type	Occurs	Description
return	WsResult	0..1	

Method: removeUsernameFromUserGroup**Description**

Removes an end-system username from an User Group

Action

urn:removeUsernameFromUserGroup

Style

Document

Input (Literal)

The input of this method is the argument removeUsernameFromUserGroup having the structure defined by the following table.

Name	Type	Occurs	Description
userGroup	string	0..1	the name of the user group to change
username	string	0..1	a username of an end-system user
reauthenticate	boolean	0..1	true to reauthenticate end-systems affected by this change

Output (Literal)

The output of this method is the argument removeUsernameFromUserGroupResponse having the structure defined by the following table.

Output (Literal)

Name	Type	Occurs	Description
return	int	0..1	0 - This error code indicates success. 1 - This error indicates that the requested object does not exist. 2 - This error indicates that the action cannot be performed because the object already exists. 3 - This error indicates that a parameter value is invalid. 4 - This error code indicates an error parsing an input string. 5 - This error code indicates that the result would be an invalid configuration. 6 - This error code is used to report an error using a remote connection. 7 - This error code is a catch-all for an unexpected error condition. 8 - This error code is used to report the group parameter does not exist 9 - A generic CSV operation error

Method: removeUsernameFromUserGroupEx**Description**

Removes an end-system username from an User Group

Action

urn:removeUsernameFromUserGroupEx

Style

Document

Input (Literal)

The input of this method is the argument removeUsernameFromUserGroupEx having the structure defined by the following table.

Name	Type	Occurs	Description
userGroup	string	0..1	the name of the user group to change
username	string	0..1	a username of an end-system user
reauthenticate	boolean	0..1	true to reauthenticate end-systems affected by this change

Output (Literal)

The output of this method is the argument removeUsernameFromUserGroupExResponse having the structure defined by the following table.

Output (Literal)

Name	Type	Occurs	Description
return	WsResult	0..1	

Method: removeValueFromNamedList**Description**

Removes a value from a named list.

Action

urn:removeValueFromNamedList

Style

Document

Input (Literal)

The input of this method is the argument removeValueFromNamedList having the structure defined by the following table.

Name	Type	Occurs	Description
list	string	0..1	the name of the named list to change
value	string	0..1	the value to remove from the named list
reauthenticate	boolean	0..1	true to reauthenticate end-systems affected by this change

Output (Literal)

The output of this method is the argument removeValueFromNamedListResponse having the structure defined by the following table.

Output (Literal)

Name	Type	Occurs	Description
return	int	0..1	0 - This error code indicates success. 1 - This error indicates that the requested object does not exist. 2 - This error indicates that the action cannot be performed because the object already exists. 3 - This error indicates that a parameter value is invalid. 4 - This error code indicates an error parsing an input string. 5 - This error code indicates that the result would be an invalid configuration. 6 - This error code is used to report an error using a remote connection. 7 - This error code is a catch-all for an unexpected error condition. 8 - This error code is used to report the group parameter does not exist 9 - A generic CSV operation error

Remarks

A named list is a container for configurations in IAM. Endsystems groups, time schedules, locations, etc, are all named list.

IAM groups can be of types:

DEVICETYPE
 MAC
 TIMEOFWEEK
 USERNAME
 RADIUSUSERGROUP
 LDAPUSERGROUP
 LOCATION
 TIMEOFWEEK
 HOSTNAME
 IP
 LDAPHOSTGROUP

The Default Work Week schedule in IAM is defined as:

1/420-1080,2/420-1080,3/420-1080,4/420-1080,5/420-1080

One entry per day with start-stop of schedule in minutes since midnight. The slash between day and time must be escaped in ws call.

Remarks**Locations group entries are specified as follows:**

A single source IP of the authenticator: 192.168.10.11 means all authentications from this switch or wireless controller.

Single IP and a port or port range: 192.168.10.10:ge.1.2 or 192.168.10.10:ge.1.2-24

Source Controller IP with SSID and AP specification (wildcards allowed): 192.168.30.50;*\$*, 192.168.30.50;SSID\$*, 192.168.30.50;SSID\$AP

MAC endsystem group entries are specified as follows:

MAC oui: 00:11:B2

Full MAC: 00:00:00:00:00:00

MAC and Mask: 00:09:09:09:09:09/FF:FF:FF:FF:FF:00 (MAC/MASK slash must be escaped in ws call)

IP endsystem group entries are specified as follows:

IP and mask: 192.168.0.0/255.255.0.0 (IP/MASK slash must be escaped in ws call)

User groups entries either Radius or LDAP are specified as key=value, Example:

Usergroup=302 in a radius group defines all users whose radius attribute Usegroup equals 302. IAM appliance must have the dictionaries defining the attribute Usergroup.

cn=Salem matches all users with attribute cn equal to Salem in an LDAP search.

Method: removeValueFromNamedListEx**Description**

Removes a value from a named list.

Action

urn:removeValueFromNamedListEx

Style

Document

Input (Literal)

The input of this method is the argument removeValueFromNamedListEx having the structure defined by the following table.

Name	Type	Occurs	Description
list	string	0..1	the name of the named list to change
value	string	0..1	the value to remove from the named list

Input (Literal)

Name	Type	Occurs	Description
reauthenticate	boolean	0..1	true to reauthenticate end-systems affected by this change

Output (Literal)

The output of this method is the argument `removeValueFromNamedListExResponse` having the structure defined by the following table.

Name	Type	Occurs	Description
return	WsResult	0..1	

Remarks

A named list is a container for configurations in IAM. Endsystems groups, time schedules, locations, etc, are all named list.

IAM groups can be of types:

DEVICETYPE
 MAC
 TIMEOFWEEK
 USERNAME
 RADIUSUSERGROUP
 LDAPUSERGROUP
 LOCATION
 TIMEOFWEEK
 HOSTNAME
 IP
 LDAPHOSTGROUP

The Default Work Week schedule in IAM is defined as:

1/420-1080,2/420-1080,3/420-1080,4/420-1080,5/420-1080

One entry per day with start-stop of schedule in minutes since midnight. The slash between day and time must be escaped in ws call.

Locations group entries are specified as follows:

A single source IP of the authenticator: 192.168.10.11 means all authentications from this switch or wireless controller.

Single IP and a port or port range: 192.168.10.10:ge.1.2 or 192.168.10.10:ge.1.2-24

Remarks

Source Controller IP with SSID and AP specification (wildcards allowed): 192.168.30.50;*\$*, 192.168.30.50;SSID\$*, 192.168.30.50;SSID\$AP

MAC endsystem group entries are specified as follows:

MAC oui: 00:11:B2

Full MAC: 00:00:00:00:00:00

MAC and Mask: 00:09:09:09:09:09/FF:FF:FF:FF:FF:00 (MAC/MASK slash must be escaped in ws call)

IP endsystem group entries are specified as follows:

IP and mask: 192.168.0.0/255.255.0.0 (IP/MASK slash must be escaped in ws call)

User groups entries either Radius or LDAP are specified as key=value, Example:

Usergroup=302 in a radius group defines all users whose radius attribute Usegroup equals 302. IAM appliance must have the dictionaries defining the attribute Usergroup.

cn=Salem matches all users with attribute cn equal to Salem in an LDAP search.

Method: saveEndSystemInfo**Description**

Create or update end-system information.

The data is provided as a as a set of comma-delimited key=value pairs. The end-system can be identified by the "macAddress", "ipAddress", or "hostname" property (just one). The following properties can be specified: "custom1", "custom2", "custom3", "custom4".

Action

urn:saveEndSystemInfo

Style

Document

Input (Literal)

The input of this method is the argument saveEndSystemInfo having the structure defined by the following table.

Name	Type	Occurs	Description
properties	string	0..1	a string representation of the data in key=value,key=value... format

Output (Literal)

The output of this method is the argument saveEndSystemInfoResponse having the structure defined by the

Output (Literal)

following table.

Name	Type	Occurs	Description
return	int	0..1	0 - This error code indicates success. 1 - This error indicates that the requested object does not exist. 2 - This error indicates that the action cannot be performed because the object already exists. 3 - This error indicates that a parameter value is invalid. 4 - This error code indicates an error parsing an input string. 5 - This error code indicates that the result would be an invalid configuration. 6 - This error code is used to report an error using a remote connection. 7 - This error code is a catch-all for an unexpected error condition. 8 - This error code is used to report the group parameter does not exist 9 - A generic CSV operation error

Remarks

The following properties can be specified: "custom1", "custom2", "custom3", "custom4".

When not specified or left blank the content of the existing customX field will be deleted. To leave a custom field unaffected in a update use the literal **null** as its value.

Method: saveEndSystemInfoByHostname**Description**

Create or update end-system information. The end-system is identified by hostname.

If there is no end system data that correlates a hostname to a MAC address, then this method will fail with error code NOT_FOUND.

Action

urn:saveEndSystemInfoByHostname

Style

Document

Input (Literal)

The input of this method is the argument saveEndSystemInfoByHostname having the structure defined by the following table.

Input (Literal)

Name	Type	Occurs	Description
hostname	string	0..1	the hostname of the EndSystem
custom1	string	0..1	the first custom field value
custom2	string	0..1	the second custom field value
custom3	string	0..1	the third custom field value
custom4	string	0..1	the fourth custom field value

Output (Literal)

The output of this method is the argument saveEndSystemInfoByHostnameResponse having the structure defined by the following table.

Name	Type	Occurs	Description
return	int	0..1	0 - This error code indicates success. 1 - This error indicates that the requested object does not exist. 2 - This error indicates that the action cannot be performed because the object already exists. 3 - This error indicates that a parameter value is invalid. 4 - This error code indicates an error parsing an input string. 5 - This error code indicates that the result would be an invalid configuration. 6 - This error code is used to report an error using a remote connection. 7 - This error code is a catch-all for an unexpected error condition. 8 - This error code is used to report the group parameter does not exist 9 - A generic CSV operation error

Remarks

When not specified or left blank the content of the existing customX field will be deleted. To leave a custom field unaffected in a update use the literal **null** as its value.

Method: saveEndSystemInfoByIp**Description**

Create or update end-system information. The end-system is identified by IP address.

If there is no end system data that correlates an IP address to a MAC address, then this method will fail with error code NOT_FOUND.

Action

urn:saveEndSystemInfoByIp

Style

Document

Input (Literal)

The input of this method is the argument saveEndSystemInfoByIp having the structure defined by the following table.

Name	Type	Occurs	Description
ipAddress	string	0..1	an IP address of an end-system
custom1	string	0..1	the first custom field value
custom2	string	0..1	the second custom field value
custom3	string	0..1	the third custom field value
custom4	string	0..1	the fourth custom field value

Output (Literal)

The output of this method is the argument saveEndSystemInfoByIpResponse having the structure defined by the following table.

Name	Type	Occurs	Description
return	int	0..1	0 - This error code indicates success. 1 - This error indicates that the requested object does not exist. 2 - This error indicates that the action cannot be performed because the object already exists. 3 - This error indicates that a parameter value is invalid. 4 - This error code indicates an error parsing an input string. 5 - This error code indicates that the result would be an invalid configuration. 6 - This error code is used to report an error using a remote connection. 7 - This error code is a catch-all for an unexpected error condition. 8 - This error code is used to report the group parameter does not exist 9 - A generic CSV operation error

Remarks

When not specified or left blank the content of the existing customX field will be deleted. To leave a custom field unaffected in a update use the literal **null** as its value.

Method: saveEndSystemInfoByMac**Description**

Create or update end-system information. The end-system is identified by MAC address.

Action

urn:saveEndSystemInfoByMac

Style

Document

Input (Literal)

The input of this method is the argument saveEndSystemInfoByMac having the structure defined by the following table.

Name	Type	Occurs	Description
macAddress	string	0..1	a full MAC address of an end-system
custom1	string	0..1	the first custom field value
custom2	string	0..1	the second custom field value
custom3	string	0..1	the third custom field value
custom4	string	0..1	the fourth custom field value

Output (Literal)

The output of this method is the argument saveEndSystemInfoByMacResponse having the structure defined by the following table.

Name	Type	Occurs	Description
return	int	0..1	0 - This error code indicates success. 1 - This error indicates that the requested object does not exist. 2 - This error indicates that the action cannot be performed because the object already exists. 3 - This error indicates that a parameter value is invalid. 4 - This error code indicates an error parsing an input string. 5 - This error code indicates that the result would be an invalid configuration. 6 - This error code is used to report an error using a remote connection. 7 - This error code is a catch-all for an unexpected error condition. 8 - This error code is used to report the group parameter does not exist 9 - A generic CSV operation error

Remarks

saveEndSystemInfoByMac Web Service

The next Java example will cover saving the custom fields for an end system based on a MAC address.

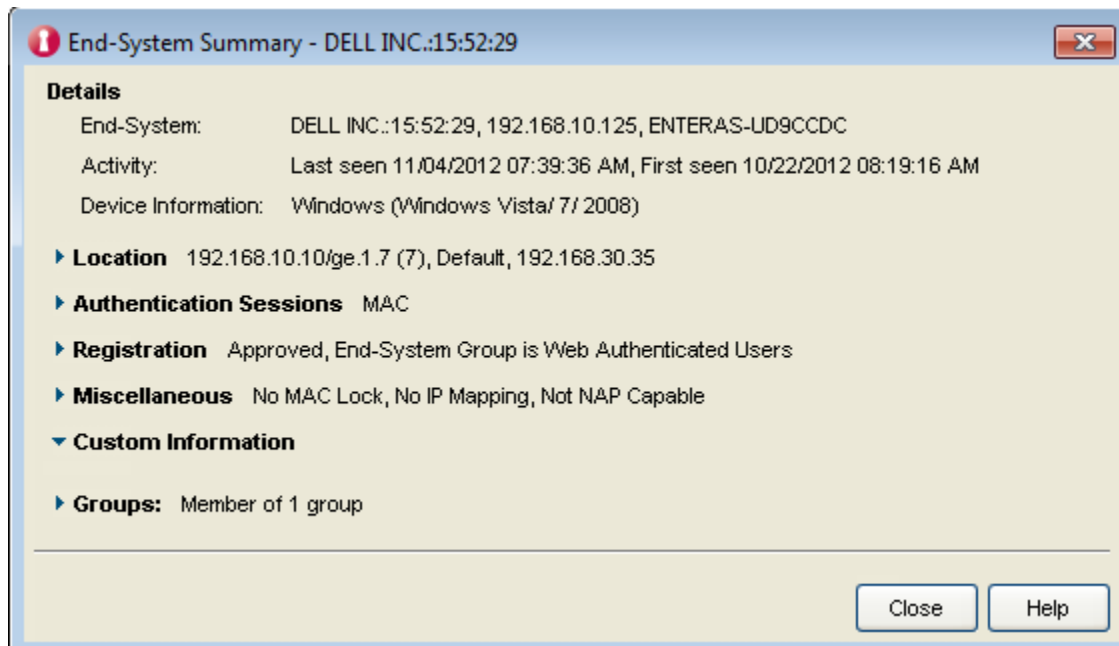
```

- <xs:element name="saveEndSystemInfoByMac">
  - <xs:complexType>
    - <xs:sequence>
      <xs:element minOccurs="0" name="macAddress" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="custom1" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="custom2" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="custom3" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="custom4" nillable="true" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
- <xs:element name="saveEndSystemInfoByMacResponse">
  - <xs:complexType>
    - <xs:sequence>
      <xs:element minOccurs="0" name="return" type="xs:int"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

From the WSDL, we can see the *saveEndSystemInfoByMac* web service has 5 *String* arguments. The arguments are as follows: *macAddress*, *custom1*, *custom2*, *custom3*, and *custom4* fields. The return value from the web service call (*saveEndSystemInfoByMacResponse*) returns an int.

In this example, the MAC address we will use is 5C:26:0A:15:52:29. As you can see in the screenshot below, there are no custom fields set.



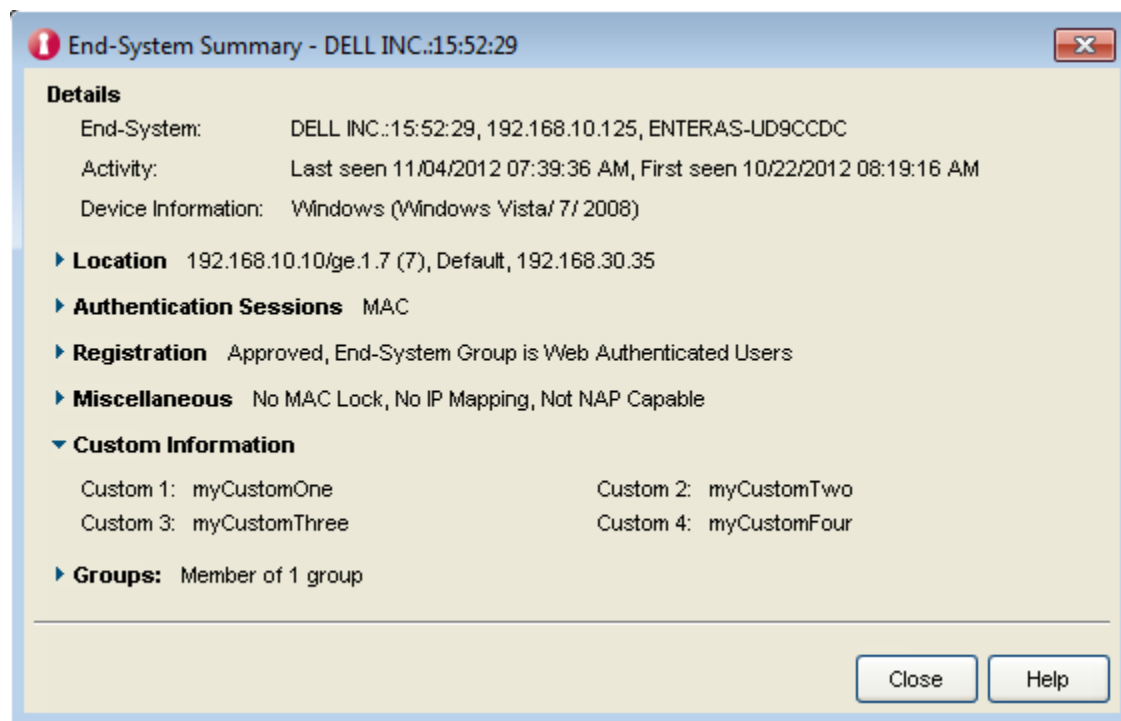
Remarks

Below are the complete code, output after the class has been compiled and ran, and NAC end system summary.

```

RPCServiceClient client = new RPCServiceClient();
Authenticator authenticator = new Authenticator();
List<String> auth = new ArrayList<String>();
auth.add(Authenticator.BASIC);
authenticator.setAuthSchemes(auth);
authenticator.setUsername("root");
authenticator.setPassword("password");
authenticator.setPreemptiveAuthentication(true);
Options options = client.getOptions();
options.setProperty(HTTPConstants.AUTHENTICATE, authenticator);
EndpointReference targetEPR = new
EndpointReference("https://nms.demo.com:8443/axis/services/NACWebService");
options.setTo(targetEPR);
QName operation = new QName("http://ws.web.server.tam.netsight.enterasys.com",
"saveEndSystemInfoByMac");
Object args[] = new Object[] {"5C:26:0A:15:52:29", "myCustomOne", "myCustomTwo", "myCustomThree",
"myCustomFour"};
Class returnTypes[] = new Class[] {Integer.class};
Object response[] = client.invokeBlocking(operation, args, returnTypes);
int value = -1;
if((response != null) && (response.length > 0))
{
    value = (Integer)response[0];
    System.out.println("Return Value: " + value);
}
Return Value: 0

```


Remarks**saveEndSystemInfoByMac Web Service**

Below are the complete code and output of the PHP script.

```
$wsdl = "https://nms.demo.com:8443/axis/services/NACWebService?wsdl";
```

```
$client = new SoapClient($wsdl, array(
```

```
    "login" => "root",
```

```
    "password" => "password"
```

```
));
```

```
$response = $client->saveEndSystemInfoByMac(array(
```

```
    "macAddress" => "5C:26:0A:15:52:29",
```

```
    "custom1" => "myCustomOne",
```

```
    "custom2" => "myCustomTwo",
```

```
    "custom3" => "myCustomThree",
```

```
    "custom4" => "myCustomFour"
```

```
));
```

```
print_r($response);
```

```
C:\Users\Administrator\Desktop>php ws.php
```

```
stdClass Object
```

```
(
```

Remarks

```
[return] => 0
)
```

When not specified or left blank the content of the existing customX field will be deleted. To leave a custom field unaffected in a update use the literal **null** as its value:

```
$response = $client->saveEndSystemInfoByMac(array(
    "macAddress" => "5C:26:0A:15:52:29",
    "custom1" => "myCustomOne",
    "custom2" => "null",
    "custom3" => "myCustomThree",
    "custom4" => "myCustomFour"
));
```

Method: saveEndSystemInfoEx**Description**

Create or update end-system information.

Action

urn:saveEndSystemInfoEx

Style

Document

Input (Literal)

The input of this method is the argument saveEndSystemInfoEx having the structure defined by the following table.

Name	Type	Occurs	Description
info	EndSystemInfo	0..1	the EndSystemInfo record to save.

Output (Literal)

The output of this method is the argument saveEndSystemInfoExResponse having the structure defined by the following table.

Name	Type	Occurs	Description
return	WsEndSystemInfoRes ult	0..1	

Method: saveLocalUser**Description**

Create or update a user in the Local User Database.

The local user data is provided as a as a set of comma-delimited key=value pairs.

Action

urn:saveLocalUser

Style

Document

Input (Literal)

The input of this method is the argument saveLocalUser having the structure defined by the following table.

Name	Type	Occurs	Description
propString	string	0..1	a string representation of the data in key=value,key=value... format
requestingUser	string	0..1	the name of the user requesting this operation.

Output (Literal)

The output of this method is the argument saveLocalUserResponse having the structure defined by the following table.

Name	Type	Occurs	Description
return	int	0..1	0 - This error code indicates success. 1 - This error indicates that the requested object does not exist. 2 - This error indicates that the action cannot be performed because the object already exists. 3 - This error indicates that a parameter value is invalid. 4 - This error code indicates an error parsing an input string. 5 - This error code indicates that the result would be an invalid configuration. 6 - This error code is used to report an error using a remote connection. 7 - This error code is a catch-all for an unexpected error condition. 8 - This error code is used to report the group parameter does not exist 9 - A generic CSV operation error

Method: saveLocalUserEx

Description

Create or update a user in the Local User Database.

Action

urn:saveLocalUserEx

Style

Document

Input (Literal)

The input of this method is the argument saveLocalUserEx having the structure defined by the following table.

Name	Type	Occurs	Description
user	LocalUser	0..1	The LocalUser to be added to the database
requestingUser	string	0..1	the name of the user requesting this operation.

Output (Literal)

The output of this method is the argument saveLocalUserExResponse having the structure defined by the following table.

Name	Type	Occurs	Description
return	WsResult	0..1	

Method: saveRegisteredDevice**Description**

Create a new registered device.

The registered device data is provided as a as a set of comma-delimited key=value pairs.

Action

urn:saveRegisteredDevice

Style

Document

Input (Literal)

The input of this method is the argument saveRegisteredDevice having the structure defined by the following table.

Input (Literal)

Name	Type	Occurs	Description
propString	string	0..1	a string representation of the data in key=value,key=value... format
requestingUser	string	0..1	the name of the user requesting this operation.

Output (Literal)

The output of this method is the argument saveRegisteredDeviceResponse having the structure defined by the following table.

Name	Type	Occurs	Description
return	int	0..1	0 - This error code indicates success. 1 - This error indicates that the requested object does not exist. 2 - This error indicates that the action cannot be performed because the object already exists. 3 - This error indicates that a parameter value is invalid. 4 - This error code indicates an error parsing an input string. 5 - This error code indicates that the result would be an invalid configuration. 6 - This error code is used to report an error using a remote connection. 7 - This error code is a catch-all for an unexpected error condition. 8 - This error code is used to report the group parameter does not exist 9 - A generic CSV operation error

Method: saveRegisteredDeviceEx**Description**

Create or update a registered device for a registered user.

Action

urn:saveRegisteredDeviceEx

Style

Document

Input (Literal)

The input of this method is the argument saveRegisteredDeviceEx having the structure defined by the following table.

Input (Literal)

Name	Type	Occurs	Description
device	RegisteredDevice	0..1	The RegisteredDevice to be saved
requestingUser	string	0..1	the name of the user requesting this operation.

Output (Literal)

The output of this method is the argument saveRegisteredDeviceExResponse having the structure defined by the following table.

Name	Type	Occurs	Description
return	WsResult	0..1	

Method: saveRegisteredDevices**Description**

Saves a set of devices to the database

Action

urn:saveRegisteredDevices

Style

Document

Input (Literal)

The input of this method is the argument saveRegisteredDevices having the structure defined by the following table.

Name	Type	Occurs	Description
propStrings	string	0..*	A list of property strings to be the devices to be added to the database
requestingUser	string	0..1	the name of the user requesting this operation.

Output (Literal)

The output of this method is the argument saveRegisteredDevicesResponse having the structure defined by the following table.

Output (Literal)

Name	Type	Occurs	Description
return	int	0..1	0 - This error code indicates success. 1 - This error indicates that the requested object does not exist. 2 - This error indicates that the action cannot be performed because the object already exists. 3 - This error indicates that a parameter value is invalid. 4 - This error code indicates an error parsing an input string. 5 - This error code indicates that the result would be an invalid configuration. 6 - This error code is used to report an error using a remote connection. 7 - This error code is a catch-all for an unexpected error condition. 8 - This error code is used to report the group parameter does not exist 9 - A generic CSV operation error

Method: saveRegisteredDeviceWithSponsorship**Description**

Create a new registered device with sponsorship.

The registered device data is provided as a as a set of comma-delimited key=value pairs.

Action

urn:saveRegisteredDeviceWithSponsorship

Style

Document

Input (Literal)

The input of this method is the argument saveRegisteredDeviceWithSponsorship having the structure defined by the following table.

Name	Type	Occurs	Description
propString	string	0..1	a string representation of the data in key=value,key=value... format
requestingUser	string	0..1	the name of the user requesting this operation.
defaultSponsorEmail	string	0..1	sponsor email address
nacApplianceIp	string	0..1	IP of the appliance where the element is being saved

Output (Literal)

The output of this method is the argument saveRegisteredDeviceWithSponsorshipResponse having the structure defined by the following table.

Name	Type	Occurs	Description
return	int	0..1	0 - This error code indicates success. 1 - This error indicates that the requested object does not exist. 2 - This error indicates that the action cannot be performed because the object already exists. 3 - This error indicates that a parameter value is invalid. 4 - This error code indicates an error parsing an input string. 5 - This error code indicates that the result would be an invalid configuration. 6 - This error code is used to report an error using a remote connection. 7 - This error code is a catch-all for an unexpected error condition. 8 - This error code is used to report the group parameter does not exist 9 - A generic CSV operation error

Method: saveRegisteredDeviceWithSponsorshipEx**Description**

Create or update a registered device for a registered user and then sends an email to the sponsor.

Action

urn:saveRegisteredDeviceWithSponsorshipEx

Style

Document

Input (Literal)

The input of this method is the argument saveRegisteredDeviceWithSponsorshipEx having the structure defined by the following table.

Name	Type	Occurs	Description
device	RegisteredDevice	0..1	RegisteredDevice to be saved
requestingUser	string	0..1	the name of the user requesting this operation.
defaultSponsorEmail	string	0..1	sponsor email address
nacApplianceIp	string	0..1	IP of the appliance where the element is being saved

Output (Literal)

The output of this method is the argument saveRegisteredDeviceWithSponsorshipExResponse having the structure defined by the following table.

Name	Type	Occurs	Description
return	WsResult	0..1	

Method: saveRegisteredUser**Description**

Create a new registered user.

The registered user data is provided as a as a set of comma-delimited key=value pairs.

Action

urn:saveRegisteredUser

Style

Document

Input (Literal)

The input of this method is the argument saveRegisteredUser having the structure defined by the following table.

Name	Type	Occurs	Description
propString	string	0..1	a string representation of the data in key=value,key=value... format
requestingUser	string	0..1	the name of the user requesting this operation.

Output (Literal)

The output of this method is the argument saveRegisteredUserResponse having the structure defined by the following table.

Output (Literal)

Name	Type	Occurs	Description
return	int	0..1	0 - This error code indicates success. 1 - This error indicates that the requested object does not exist. 2 - This error indicates that the action cannot be performed because the object already exists. 3 - This error indicates that a parameter value is invalid. 4 - This error code indicates an error parsing an input string. 5 - This error code indicates that the result would be an invalid configuration. 6 - This error code is used to report an error using a remote connection. 7 - This error code is a catch-all for an unexpected error condition. 8 - This error code is used to report the group parameter does not exist 9 - A generic CSV operation error

Method: saveRegisteredUserEx**Description**

Create or update a registered user.

Action

urn:saveRegisteredUserEx

Style

Document

Input (Literal)

The input of this method is the argument saveRegisteredUserEx having the structure defined by the following table.

Name	Type	Occurs	Description
user	RegisteredUser	0..1	The RegisteredUser to be saved
requestingUser	string	0..1	the name of the user requesting this operation.

Output (Literal)

The output of this method is the argument saveRegisteredUserExResponse having the structure defined by the following table.

Output (Literal)

Name	Type	Occurs	Description
return	WsResult	0..1	

Method: saveRegisteredUsers**Description**

Saves a set of users to the database

Action

urn:saveRegisteredUsers

Style

Document

Input (Literal)

The input of this method is the argument saveRegisteredUsers having the structure defined by the following table.

Name	Type	Occurs	Description
propStrings	string	0..*	The user property strings to be added to the database
requestingUser	string	0..1	the name of the user requesting this operation.

Output (Literal)

The output of this method is the argument saveRegisteredUsersResponse having the structure defined by the following table.

Output (Literal)

Name	Type	Occurs	Description
return	int	0..1	0 - This error code indicates success. 1 - This error indicates that the requested object does not exist. 2 - This error indicates that the action cannot be performed because the object already exists. 3 - This error indicates that a parameter value is invalid. 4 - This error code indicates an error parsing an input string. 5 - This error code indicates that the result would be an invalid configuration. 6 - This error code is used to report an error using a remote connection. 7 - This error code is a catch-all for an unexpected error condition. 8 - This error code is used to report the group parameter does not exist 9 - A generic CSV operation error

Method: updateRegisteredDevice**Description**

Updates an existing registered device. The registered user data to be updated is provided as a set of comma-delimited key=value pairs.

Action

urn:updateRegisteredDevice

Style

Document

Input (Literal)

The input of this method is the argument updateRegisteredDevice having the structure defined by the following table.

Name	Type	Occurs	Description
propString	string	0..1	a string representation of the data in key=value,key=value... format
requestingUser	string	0..1	the name of the user requesting this operation.

Output (Literal)

The output of this method is the argument updateRegisteredDeviceResponse having the structure defined by the following table.

Output (Literal)

Name	Type	Occurs	Description
return	int	0..1	0 - This error code indicates success. 1 - This error indicates that the requested object does not exist. 2 - This error indicates that the action cannot be performed because the object already exists. 3 - This error indicates that a parameter value is invalid. 4 - This error code indicates an error parsing an input string. 5 - This error code indicates that the result would be an invalid configuration. 6 - This error code is used to report an error using a remote connection. 7 - This error code is a catch-all for an unexpected error condition. 8 - This error code is used to report the group parameter does not exist 9 - A generic CSV operation error

Method: updateRegisteredUser**Description**

Updates an existing registered user. The registered user data to be updated is provided as a set of comma-delimited key=value pairs.

Action

urn:updateRegisteredUser

Style

Document

Input (Literal)

The input of this method is the argument updateRegisteredUser having the structure defined by the following table.

Name	Type	Occurs	Description
propString	string	0..1	a string representation of the data in key=value,key=value... format
requestingUser	string	0..1	the name of the user requesting this operation.

Output (Literal)

The output of this method is the argument updateRegisteredUserResponse having the structure defined by the following table.

Output (Literal)

Name	Type	Occurs	Description
return	int	0..1	0 - This error code indicates success. 1 - This error indicates that the requested object does not exist. 2 - This error indicates that the action cannot be performed because the object already exists. 3 - This error indicates that a parameter value is invalid. 4 - This error code indicates an error parsing an input string. 5 - This error code indicates that the result would be an invalid configuration. 6 - This error code is used to report an error using a remote connection. 7 - This error code is a catch-all for an unexpected error condition. 8 - This error code is used to report the group parameter does not exist 9 - A generic CSV operation error

Complex Types: NAC WebService

Complex Types

Name	Description
EndSystemDTO	Ensystem information as contained in the end system table in IAM
EndSystemInfo	An EndSystem Info is an extension of an EndSystem record containing custom fields.
LocalUser	
RegisteredDevice	
RegisteredUser	
WsEndSystemInfoResult	
WsEndSystemListResult	
WsEndSystemResult	
WsEnforceApplianceResult	
WsEnforceResult	
WsResult	

Complex Type: EndSystemDTO

Description


Ensystem information as contained in the end system table in IAM

Derived By

Restricting anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
 SEQUENCE		1..1	
allAuthTypes	string	0..1	All authentication types that provided information about the device
assmtHashCode	int	0..1	Hash code
authType	string	0..1	Authentication type used to evaluate the end system
eSType	string	0..1	Type of end system based in device fingerprint.

Content Model

Component	Type	Occurs	Description
extendedState	string	0..1	Extended state of the device, usually contains temporary states during authentication or IP resolution and detailed information relevant to the device's state. It will be one of: Assessment Server(s) Unavailable. Assessment Server Can't Reach Host MAC to IP Resolution Failed MAC to IP Resolution Timed Out No Assessment Servers Configured Assessment Bypass Enabled No Error RADIUS Request missing Required Attributes Resolving IP Address Scan in Progress Scan Requested Scan Complete
firstSeenTime	dateTime	0..1	Date and time when the device was first detected in the network
hostName	string	0..1	Host name if detected by any of the available methods, nodealias, kerberos snooping, assessment, etc.
id	long	0..1	device id in the database.
ipAddress	string	0..1	Device IP address as detected by IP resolution methods
lastAssmtHashCode ChangeTime	dateTime	0..1	Date and time when the assessment info was updated.
lastAuthEventTime	dateTime	0..1	Date and time of the last authentication or reauthentication event.
lastQuarantineTime	dateTime	0..1	Date and time of the last quarantine event.
lastScanResultState	string	0..1	Scan result of last health assessment on the device
lastScanTime	dateTime	0..1	Date and time of the last assessment.
lastSeenTime	dateTime	0..1	Date and time the device was last seen in the network.
locationInfo	string	0..1	Location information extracted from the systemlocation mib of the authenticating device.
macAddress	string	0..1	MAC address of the end system
nacApplianceGroup Name	string	0..1	IAM appliance group name that contains the authenticating IAM.
nacApplianceIP	string	0..1	IP of the IAM that processed the authentication.
nacProfileName	string	0..1	NAC profile name that matched the last end system authentication.

Content Model

Component	Type	Occurs	Description
operatingSystemName	string	0..1	Operating system string detected by fingerprinting.
policy	string	0..1	Policy applied to the end system
radiusServerIp	string	0..1	Radius server IP that processed the last authentication request.
reason	string	0..1	String containing a descriptive reason of the device's current state and NAC profile match.
regType	string	0..1	Registration type
requestAttributes	string	0..1	Additional attributes provided during the radius exchange while authenticating the ES
startAssmtWarningTime	dateTime	0..1	Date and time when the end system was put in assessment warning.
state	string	0..1	Quarantine Scan Accept Reject Error Disconnected
stateDescr	string	0..1	Description string of the ES state
switchIP	string	0..1	
switchPort	int	0..1	
switchPortId	string	0..1	
username	string	0..1	
zone	string	0..1	

Element: allAuthTypes [type EndSystemDTO]

Element: assmtHashCode [type EndSystemDTO]

Element: authType [type EndSystemDTO]

Element: eSType [type EndSystemDTO]

Element: extendedState [type EndSystemDTO]

Element: firstSeenTime [type EndSystemDTO]

Element: hostName [type EndSystemDTO]

Element: id [type EndSystemDTO]

Element: ipAddress [type EndSystemDTO]

Element: lastAssmtHashCodeChangeTime [type EndSystemDTO]

Element: lastAuthEventTime [type EndSystemDTO]

Element: lastQuarantineTime [type EndSystemDTO]

Element: lastScanResultState [type EndSystemDTO]

Element: lastScanTime [type EndSystemDTO]

Element: lastSeenTime [type EndSystemDTO]

Element: locationInfo [type EndSystemDTO]

Element: macAddress [type EndSystemDTO]

Element: nacApplianceGroupName [type EndSystemDTO]

Element: nacApplianceIP [type EndSystemDTO]

Element: nacProfileName [type EndSystemDTO]

Element: operatingSystemName [type EndSystemDTO]

Element: policy [type EndSystemDTO]

Element: radiusServerIp [type EndSystemDTO]

Element: reason [type EndSystemDTO]

Element: regType [type EndSystemDTO]

Element: requestAttributes [type EndSystemDTO]

Element: startAssmtWarningTime [type EndSystemDTO]

Element: state [type EndSystemDTO]

Element: stateDescr [type EndSystemDTO]

Element: switchIP [type EndSystemDTO]

Element: switchPort [type EndSystemDTO]

Element: switchPortId [type EndSystemDTO]

Element: username [type EndSystemDTO]

Element: zone [type EndSystemDTO]

Complex Type: EndSystemInfo

Description


An EndSystem Info is an extension of an EndSystem record containing custom fields.

Derived By

Restricting anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
 SEQUENCE		1..1	
custom1	string	0..1	the first custom field value
custom2	string	0..1	the second custom field value
custom3	string	0..1	the third custom field value
custom4	string	0..1	the fourth custom field value
endSystemKey	string	0..1	
groupDescr1	string	0..1	
groupDescr2	string	0..1	
groupDescr3	string	0..1	
memberOfGroups	string	0..1	Groups this end system is member of.
regData1	string	0..1	first registration data field
regData2	string	0..1	second registration data field
regData3	string	0..1	third registration data field
regData4	string	0..1	fourth registration data field
regData5	string	0..1	fifth registration data field
regDeviceDescr	string	0..1	Device description from registration process
regEmail	string	0..1	registration email
regName	string	0..1	registration email
regPhone	string	0..1	registration phone
regSponsor	string	0..1	registration sponsor
registeredDeviceInfo	RegisteredDevice	0..1	
registeredUserInfo	RegisteredUser	0..1	

Element: custom1 [type EndSystemInfo]

Element: custom2 [type EndSystemInfo]

Element: custom3 [type EndSystemInfo]

Element: custom4 [type EndSystemInfo]

Element: endSystemKey [type EndSystemInfo]

Element: groupDescr1 [type EndSystemInfo]

Element: groupDescr2 [type EndSystemInfo]

Element: groupDescr3 [type EndSystemInfo]

Element: memberOfGroups [type EndSystemInfo]

Element: regData1 [type EndSystemInfo]

Element: regData2 [type EndSystemInfo]

Element: regData3 [type EndSystemInfo]

Element: regData4 [type EndSystemInfo]

Element: regData5 [type EndSystemInfo]

Element: regDeviceDescr [type EndSystemInfo]

Element: regEmail [type EndSystemInfo]

Element: regName [type EndSystemInfo]

Element: regPhone [type EndSystemInfo]

Element: regSponsor [type EndSystemInfo]

Element: registeredDeviceInfo [type EndSystemInfo]

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
applianceGroup	string	0..1	
description	string	0..1	
deviceGroup	string	0..1	
id	long	0..1	
idAsString	string	0..1	

Content Model

Component	Type	Occurs	Description
ipAddress	string	0..1	
macAddress	string	0..1	
registrationTime	dateTime	0..1	
sponsor	string	0..1	
sponsorDeviceGroup	string	0..1	
sponsored	boolean	0..1	
stateStr	string	0..1	
userName	string	0..1	

Element: registeredUserInfo [type EndSystemInfo]

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
applianceGroup	string	0..1	
attempts	int	0..1	
displayName	string	0..1	
emailAddress	string	0..1	
expiresTime	dateTime	0..1	
firstName	string	0..1	
fullName	string	0..1	
id	long	0..1	
idAsString	string	0..1	
lastName	string	0..1	
maxRegisterCount	int	0..1	
middleName	string	0..1	
phoneNumber	string	0..1	
preRegistered	boolean	0..1	
registrationTime	dateTime	0..1	
sponsor	string	0..1	
startTime	dateTime	0..1	
userData1	string	0..1	
userData2	string	0..1	

Content Model

Component	Type	Occurs	Description
userData3	string	0..1	
userData4	string	0..1	
userData5	string	0..1	
userName	string	0..1	
userTypeStr	string	0..1	
MAX_USERNAME_LENGTH	int	0..1	
AUTH_REG	string	0..1	
AUTH_REG_LONG_COMMENT	string	0..1	
AUTH_REG_SHORT_COMMENT	string	0..1	
COMMA_SEP	string	0..1	
COMMA_SUBSTITUTE	string	0..1	
COMMA_SUBSTITUTE_SEP	string	0..1	
GUEST	string	0..1	
GUEST_LONG_COMMENT	string	0..1	
GUEST_SHORT_COMMENT	string	0..1	
GUEST_WEB_ACCESS_LONG_COMMENT	string	0..1	
GUEST_WEB_ACCESS_SHORT_COMMENT	string	0..1	
PRE_REG_LONG_COMMENT	string	0..1	
PRE_REG_SHORT_COMMENT	string	0..1	
REGISTERED_DEVICE_COMMENT	string	0..1	
SECURE_GUEST_LONG_COMMENT	string	0..1	
SECURE_GUEST_SHORT_COMMENT	string	0..1	
SPONSORED_AUTH_REG_LONG_COMMENT	string	0..1	

Content Model


Component	Type	Occurs	Description
SPONSORED_AUTH_REG_SHORT_COMMENT	string	0..1	
SPONSORED_GUEST_LONG_COMMENT	string	0..1	
SPONSORED_GUEST_SHORT_COMMENT	string	0..1	
SPONSORED_PRE_REG_LONG_COMMENT	string	0..1	
SPONSORED_PRE_REG_SHORT_COMMENT	string	0..1	
SPONSORED_SECURE_GUEST_LONG_COMMENT	string	0..1	
SPONSORED_SECURE_GUEST_SHORT_COMMENT	string	0..1	
TRANSIENT	string	0..1	
USERNAME_COMMENT	string	0..1	
WEB_AUTH	string	0..1	
WEB_AUTH_COMMENT	string	0..1	
WEB_AUTH_LONG_COMMENT	string	0..1	
WEB_AUTH_SHORT_COMMENT	string	0..1	

Complex Type: LocalUser**Derived By**

Restricting anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
 SEQUENCE		1..1	

Content Model

Component	Type	Occurs	Description
dbData	string	0..1	
description	string	0..1	
displayName	string	0..1	
domainName	string	0..1	
enabled	boolean	0..1	
firstName	string	0..1	
id	long	0..1	
lastName	string	0..1	
loginId	string	0..1	
loginPassword	string	0..1	
loginPasswordHash	string	0..1	
loginPasswordHash Type	int	0..1	
preOrSelfProvisioned	boolean	0..1	
DEFAULT_PASSWORD_HASH_TYPE	int	0..1	
PKCS5_REVERSIBLE_HASH_TYPE	int	0..1	
SHA1_NON_REVERSIBLE_HASH_TYPE	int	0..1	
STR_DEFAULT_AD_MIN_NAME	string	0..1	
STR_DEFAULT_DOMAIN	string	0..1	
STR_DEFAULT_SPONSOR_NAME	string	0..1	

Element: dbData [type LocalUser]

Element: description [type LocalUser]

Element: displayName [type LocalUser]

Element: domainName [type LocalUser]

Element: enabled [type LocalUser]

Element: firstName [type LocalUser]

Element: id [type LocalUser]

Element: lastName [type LocalUser]

Element: loginId [type LocalUser]

Element: loginPassword [type LocalUser]

Element: loginPasswordHash [type LocalUser]

Element: loginPasswordHashType [type LocalUser]

Element: preOrSelfProvisioned [type LocalUser]

Element: DEFAULT_PASSWORD_HASH_TYPE [type LocalUser]

Element: PKCS5_REVERSIBLE_HASH_TYPE [type LocalUser]

Element: SHA1_NON_REVERSIBLE_HASH_TYPE [type LocalUser]

Element: STR_DEFAULT_ADMIN_NAME [type LocalUser]

Element: STR_DEFAULT_DOMAIN [type LocalUser]

Element: STR_DEFAULT_SPONSOR_NAME [type LocalUser]


Complex Type: RegisteredDevice

Derived By

Restricting anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
 SEQUENCE		1..1	
applianceGroup	string	0..1	
description	string	0..1	
deviceGroup	string	0..1	
id	long	0..1	
idAsString	string	0..1	
ipAddress	string	0..1	
macAddress	string	0..1	
registrationTime	dateTime	0..1	
sponsor	string	0..1	

Content Model

Component	Type	Occurs	Description
sponsorDeviceGroup	string	0..1	
sponsored	boolean	0..1	
stateStr	string	0..1	
userName	string	0..1	

Element: applianceGroup [type RegisteredDevice]

Element: description [type RegisteredDevice]

Element: deviceGroup [type RegisteredDevice]

Element: id [type RegisteredDevice]

Element: idAsString [type RegisteredDevice]

Element: ipAddress [type RegisteredDevice]

Element: macAddress [type RegisteredDevice]

Element: registrationTime [type RegisteredDevice]

Element: sponsor [type RegisteredDevice]

Element: sponsorDeviceGroup [type RegisteredDevice]

Element: sponsored [type RegisteredDevice]

Element: stateStr [type RegisteredDevice]

Element: userName [type RegisteredDevice]

Complex Type: RegisteredUser**Derived By**

Restricting anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	

Content Model

Component	Type	Occurs	Description
applianceGroup	string	0..1	
attempts	int	0..1	
displayName	string	0..1	
emailAddress	string	0..1	
expiresTime	dateTime	0..1	
firstName	string	0..1	
fullName	string	0..1	
id	long	0..1	
idAsString	string	0..1	
lastName	string	0..1	
maxRegisterCount	int	0..1	
middleName	string	0..1	
phoneNumber	string	0..1	
preRegistered	boolean	0..1	
registrationTime	dateTime	0..1	
sponsor	string	0..1	
startTime	dateTime	0..1	
userData1	string	0..1	
userData2	string	0..1	
userData3	string	0..1	
userData4	string	0..1	
userData5	string	0..1	
userName	string	0..1	
userTypeStr	string	0..1	
MAX_USERNAME_LENGTH	int	0..1	
AUTH_REG	string	0..1	
AUTH_REG_LONG_COMMENT	string	0..1	
AUTH_REG_SHORT_COMMENT	string	0..1	
COMMA_SEP	string	0..1	
COMMA_SUBSTITUTE	string	0..1	
COMMA_SUBSTITUTE_SEP	string	0..1	

Content Model

Component	Type	Occurs	Description
GUEST	string	0..1	
GUEST_LONG_COMMENT	string	0..1	
GUEST_SHORT_COMMENT	string	0..1	
GUEST_WEB_ACCESS_LONG_COMMENT	string	0..1	
GUEST_WEB_ACCESS_SHORT_COMMENT	string	0..1	
PRE_REG_LONG_COMMENT	string	0..1	
PRE_REG_SHORT_COMMENT	string	0..1	
REGISTERED_DEVICE_COMMENT	string	0..1	
SECURE_GUEST_LONG_COMMENT	string	0..1	
SECURE_GUEST_SHORT_COMMENT	string	0..1	
SPONSORED_AUTH_REG_LONG_COMMENT	string	0..1	
SPONSORED_AUTH_REG_SHORT_COMMENT	string	0..1	
SPONSORED_GUEST_LONG_COMMENT	string	0..1	
SPONSORED_GUEST_SHORT_COMMENT	string	0..1	
SPONSORED_PRE_REG_LONG_COMMENT	string	0..1	
SPONSORED_PRE_REG_SHORT_COMMENT	string	0..1	
SPONSORED_SECURE_GUEST_LONG_COMMENT	string	0..1	

Content Model

Component	Type	Occurs	Description
SPONSORED_SECURITY_GUEST_SHORT_COMMENT	string	0..1	
TRANSIENT	string	0..1	
USERNAME_COMMENT	string	0..1	
WEB_AUTH	string	0..1	
WEB_AUTH_COMMENT	string	0..1	
WEB_AUTH_LONG_COMMENT	string	0..1	
WEB_AUTH_SHORT_COMMENT	string	0..1	

Element: applianceGroup [type RegisteredUser]

Element: attempts [type RegisteredUser]

Element: displayName [type RegisteredUser]

Element: emailAddress [type RegisteredUser]

Element: expiresTime [type RegisteredUser]

Element: firstName [type RegisteredUser]

Element: fullName [type RegisteredUser]

Element: id [type RegisteredUser]

Element: idAsString [type RegisteredUser]

Element: lastName [type RegisteredUser]

Element: maxRegisterCount [type RegisteredUser]

Element: middleName [type RegisteredUser]

Element: phoneNumber [type RegisteredUser]

Element: preRegistered [type RegisteredUser]

Element: registrationTime [type RegisteredUser]

Element: sponsor [type RegisteredUser]

Element: startTime [type RegisteredUser]

Element: userData1 [type RegisteredUser]

Element: userData2 [type RegisteredUser]

Element: userData3 [type RegisteredUser]

Element: userData4 [type RegisteredUser]

Element: userData5 [type RegisteredUser]

Element: userName [type RegisteredUser]

Element: userTypeStr [type RegisteredUser]

Element: MAX_USERNAME_LENGTH [type RegisteredUser]

Element: AUTH_REG [type RegisteredUser]

Element: AUTH_REG_LONG_COMMENT [type RegisteredUser]

Element: AUTH_REG_SHORT_COMMENT [type RegisteredUser]

Element: COMMA_SEP [type RegisteredUser]

Element: COMMA_SUBSTITUTE [type RegisteredUser]

Element: COMMA_SUBSTITUTE_SEP [type RegisteredUser]

Element: GUEST [type RegisteredUser]

Element: GUEST_LONG_COMMENT [type RegisteredUser]

Element: GUEST_SHORT_COMMENT [type RegisteredUser]

Element: GUEST_WEB_ACCESS_LONG_COMMENT [type RegisteredUser]

Element: GUEST_WEB_ACCESS_SHORT_COMMENT [type RegisteredUser]

Element: PRE_REG_LONG_COMMENT [type RegisteredUser]

Element: PRE_REG_SHORT_COMMENT [type RegisteredUser]

Element: REGISTERED_DEVICE_COMMENT [type RegisteredUser]

Element: SECURE_GUEST_LONG_COMMENT [type RegisteredUser]

Element: SECURE_GUEST_SHORT_COMMENT [type RegisteredUser]

Element: SPONSORED_AUTH_REG_LONG_COMMENT [type RegisteredUser]

Element: SPONSORED_AUTH_REG_SHORT_COMMENT [type RegisteredUser]

Element: SPONSORED_GUEST_LONG_COMMENT [type RegisteredUser]

Element: SPONSORED_GUEST_SHORT_COMMENT [type RegisteredUser]

Element: SPONSORED_PRE_REG_LONG_COMMENT [type RegisteredUser]

Element: SPONSORED_PRE_REG_SHORT_COMMENT [type RegisteredUser]

Element: SPONSORED_SECURE_GUEST_LONG_COMMENT [type RegisteredUser]

Element: SPONSORED_SECURE_GUEST_SHORT_COMMENT [type RegisteredUser]

Element: TRANSIENT [type RegisteredUser]

Element: USERNAME_COMMENT [type RegisteredUser]

Element: WEB_AUTH [type RegisteredUser]

Element: WEB_AUTH_COMMENT [type RegisteredUser]

Element: WEB_AUTH_LONG_COMMENT [type RegisteredUser]

Element: WEB_AUTH_SHORT_COMMENT [type RegisteredUser]

Complex Type: WsEndSystemInfoResult

Derived By

Restricting anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
endSystemInfo	EndSystemInfo	0..1	

Content Model

Component	Type	Occurs	Description
errorCode	int	0..1	0 - This error code indicates success. 1 - This error indicates that the requested object does not exist. 2 - This error indicates that the action cannot be performed because the object already exists. 3 - This error indicates that a parameter value is invalid. 4 - This error code indicates an error parsing an input string. 5 - This error code indicates that the result would be an invalid configuration. 6 - This error code is used to report an error using a remote connection. 7 - This error code is a catch-all for an unexpected error condition. 8 - This error code is used to report the group parameter does not exist 9 - A generic CSV operation error
errorMessage	string	0..1	
success	boolean	0..1	

Element: endSystemInfo [type WsEndSystemInfoResult]

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
custom1	string	0..1	the first custom field value
custom2	string	0..1	the second custom field value
custom3	string	0..1	the third custom field value
custom4	string	0..1	the fourth custom field value
endSystemKey	string	0..1	
groupDescr1	string	0..1	
groupDescr2	string	0..1	
groupDescr3	string	0..1	
memberOfGroups	string	0..1	Groups this end system is member of.
regData1	string	0..1	first registration data field
regData2	string	0..1	second registration data field
regData3	string	0..1	third registration data field
regData4	string	0..1	fourth registration data field

Content Model

Component	Type	Occurs	Description
regData5	string	0..1	fifth registration data field
regDeviceDescr	string	0..1	Device description from registration process
regEmail	string	0..1	registration email
regName	string	0..1	registration email
regPhone	string	0..1	registration phone
regSponsor	string	0..1	registration sponsor
registeredDeviceInfo	RegisteredDevice	0..1	
registeredUserInfo	RegisteredUser	0..1	

Element: errorCode [type WsEndSystemInfoResult]

Element: errorMessage [type WsEndSystemInfoResult]

Element: success [type WsEndSystemInfoResult]

Complex Type: WsEndSystemListResult**Derived By**

Restricting anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
endSystems	EndSystemDTO	0..*	

Content Model

Component	Type	Occurs	Description
errorCode	int	0..1	0 - This error code indicates success. 1 - This error indicates that the requested object does not exist. 2 - This error indicates that the action cannot be performed because the object already exists. 3 - This error indicates that a parameter value is invalid. 4 - This error code indicates an error parsing an input string. 5 - This error code indicates that the result would be an invalid configuration. 6 - This error code is used to report an error using a remote connection. 7 - This error code is a catch-all for an unexpected error condition. 8 - This error code is used to report the group parameter does not exist 9 - A generic CSV operation error
errorMessage	string	0..1	
success	boolean	0..1	

Element: endSystems [type WsEndSystemListResult]

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
allAuthTypes	string	0..1	All authentication types that provided information about the device
assmtHashCode	int	0..1	Hash code
authType	string	0..1	Authentication type used to evaluate the end system
eSType	string	0..1	Type of end system based in device fingerprint.

Content Model

Component	Type	Occurs	Description
extendedState	string	0..1	Extended state of the device, usually contains temporary states during authentication or IP resolution and detailed information relevant to the device's state. It will be one of: Assessment Server(s) Unavailable. Assessment Server Can't Reach Host MAC to IP Resolution Failed MAC to IP Resolution Timed Out No Assessment Servers Configured Assessment Bypass Enabled No Error RADIUS Request missing Required Attributes Resolving IP Address Scan in Progress Scan Requested Scan Complete
firstSeenTime	dateTime	0..1	Date and time when the device was first detected in the network
hostName	string	0..1	Host name if detected by any of the available methods, nodealias, kerberos snooping, assessment, etc.
id	long	0..1	device id in the database.
ipAddress	string	0..1	Device IP address as detected by IP resolution methods
lastAssmtHashCode ChangeTime	dateTime	0..1	Date and time when the assessment info was updated.
lastAuthEventTime	dateTime	0..1	Date and time of the last authentication or reauthentication event.
lastQuarantineTime	dateTime	0..1	Date and time of the last quarantine event.
lastScanResultState	string	0..1	Scan result of last health assessment on the device
lastScanTime	dateTime	0..1	Date and time of the last assessment.
lastSeenTime	dateTime	0..1	Date and time the device was last seen in the network.
locationInfo	string	0..1	Location information extracted from the systemlocation mib of the authenticating device.
macAddress	string	0..1	MAC address of the end system
nacApplianceGroup Name	string	0..1	IAM appliance group name that contains the authenticating IAM.
nacApplianceIP	string	0..1	IP of the IAM that processed the authentication.
nacProfileName	string	0..1	NAC profile name that matched the last end system authentication.

Content Model

Component	Type	Occurs	Description
operatingSystemName	string	0..1	Operating system string detected by fingerprinting.
policy	string	0..1	Policy applied to the end system
radiusServerIp	string	0..1	Radius server IP that processed the last authentication request.
reason	string	0..1	String containing a descriptive reason of the device's current state and NAC profile match.
regType	string	0..1	Registration type
requestAttributes	string	0..1	Additional attributes provided during the radius exchange while authenticating the ES
startAssmtWarningTime	dateTime	0..1	Date and time when the end system was put in assessment warning.
state	string	0..1	Quarantine Scan Accept Reject Error Disconnected
stateDescr	string	0..1	Description string of the ES state
switchIP	string	0..1	
switchPort	int	0..1	
switchPortId	string	0..1	
username	string	0..1	
zone	string	0..1	

Element: errorCode [type WsEndSystemListResult]

Element: errorMessage [type WsEndSystemListResult]


Element: success [type WsEndSystemListResult]

Complex Type: WsEndSystemResult**Derived By**

Restricting anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
 SEQUENCE		1..1	

Content Model

Component	Type	Occurs	Description
endSystem	EndSystemDTO	0..1	
endSystemSwitchSupportsReauth	boolean	0..1	
errorCode	int	0..1	0 - This error code indicates success. 1 - This error indicates that the requested object does not exist. 2 - This error indicates that the action cannot be performed because the object already exists. 3 - This error indicates that a parameter value is invalid. 4 - This error code indicates an error parsing an input string. 5 - This error code indicates that the result would be an invalid configuration. 6 - This error code is used to report an error using a remote connection. 7 - This error code is a catch-all for an unexpected error condition. 8 - This error code is used to report the group parameter does not exist 9 - A generic CSV operation error
errorMessage	string	0..1	
success	boolean	0..1	

Element: endSystem [type WsEndSystemResult]

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
allAuthTypes	string	0..1	All authentication types that provided information about the device
assmtHashCode	int	0..1	Hash code
authType	string	0..1	Authentication type used to evaluate the end system
eSType	string	0..1	Type of end system based in device fingerprint.

Content Model

Component	Type	Occurs	Description
extendedState	string	0..1	Extended state of the device, usually contains temporary states during authentication or IP resolution and detailed information relevant to the device's state. It will be one of: Assessment Server(s) Unavailable. Assessment Server Can't Reach Host MAC to IP Resolution Failed MAC to IP Resolution Timed Out No Assessment Servers Configured Assessment Bypass Enabled No Error RADIUS Request missing Required Attributes Resolving IP Address Scan in Progress Scan Requested Scan Complete
firstSeenTime	dateTime	0..1	Date and time when the device was first detected in the network
hostName	string	0..1	Host name if detected by any of the available methods, nodealias, kerberos snooping, assessment, etc.
id	long	0..1	device id in the database.
ipAddress	string	0..1	Device IP address as detected by IP resolution methods
lastAssmtHashCode ChangeTime	dateTime	0..1	Date and time when the assessment info was updated.
lastAuthEventTime	dateTime	0..1	Date and time of the last authentication or reauthentication event.
lastQuarantineTime	dateTime	0..1	Date and time of the last quarantine event.
lastScanResultState	string	0..1	Scan result of last health assessment on the device
lastScanTime	dateTime	0..1	Date and time of the last assessment.
lastSeenTime	dateTime	0..1	Date and time the device was last seen in the network.
locationInfo	string	0..1	Location information extracted from the systemlocation mib of the authenticating device.
macAddress	string	0..1	MAC address of the end system
nacApplianceGroup Name	string	0..1	IAM appliance group name that contains the authenticating IAM.
nacApplianceIP	string	0..1	IP of the IAM that processed the authentication.
nacProfileName	string	0..1	NAC profile name that matched the last end system authentication.

Content Model

Component	Type	Occurs	Description
operatingSystemName	string	0..1	Operating system string detected by fingerprinting.
policy	string	0..1	Policy applied to the end system
radiusServerIp	string	0..1	Radius server IP that processed the last authentication request.
reason	string	0..1	String containing a descriptive reason of the device's current state and NAC profile match.
regType	string	0..1	Registration type
requestAttributes	string	0..1	Additional attributes provided during the radius exchange while authenticating the ES
startAssmtWarningTime	dateTime	0..1	Date and time when the end system was put in assessment warning.
state	string	0..1	Quarantine Scan Accept Reject Error Disconnected
stateDescr	string	0..1	Description string of the ES state
switchIP	string	0..1	
switchPort	int	0..1	
switchPortId	string	0..1	
username	string	0..1	
zone	string	0..1	

Element: endSystemSwitchSupportsReauth [type WsEndSystemResult]

Element: errorCode [type WsEndSystemResult]

Element: errorMessage [type WsEndSystemResult]

Element: success [type WsEndSystemResult]

Complex Type: WsEnforceApplianceResult

Derived By

Restricting anyType

Content Model

Contains elements as defined in the following table.

Content Model

Component	Type	Occurs	Description
SEQUENCE		1..1	
errors	string	0..*	
nacApplianceIP	string	0..1	
warnings	string	0..*	

Element: errors [type WsEnforceApplianceResult]

Element: nacApplianceIP [type WsEnforceApplianceResult]

Element: warnings [type WsEnforceApplianceResult]

Complex Type: WsEnforceResult**Derived By**

Restricting anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
applianceEnforceResults	WsEnforceApplianceResult	0..*	
errorCode	int	0..1	0 - This error code indicates success. 1 - This error indicates that the requested object does not exist. 2 - This error indicates that the action cannot be performed because the object already exists. 3 - This error indicates that a parameter value is invalid. 4 - This error code indicates an error parsing an input string. 5 - This error code indicates that the result would be an invalid configuration. 6 - This error code is used to report an error using a remote connection. 7 - This error code is a catch-all for an unexpected error condition. 8 - This error code is used to report the group parameter does not exist 9 - A generic CSV operation error
errorMessage	string	0..1	
success	boolean	0..1	

Element: applianceEnforceResults [type WsEnforceResult]

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
errors	string	0..*	
nacApplianceIP	string	0..1	
warnings	string	0..*	

Element: errorCode [type WsEnforceResult]

Element: errorMessage [type WsEnforceResult]

Element: success [type WsEnforceResult]

Complex Type: WsResult

Derived By

Restricting anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
errorCode	int	0..1	0 - This error code indicates success. 1 - This error indicates that the requested object does not exist. 2 - This error indicates that the action cannot be performed because the object already exists. 3 - This error indicates that a parameter value is invalid. 4 - This error code indicates an error parsing an input string. 5 - This error code indicates that the result would be an invalid configuration. 6 - This error code is used to report an error using a remote connection. 7 - This error code is a catch-all for an unexpected error condition. 8 - This error code is used to report the group parameter does not exist 9 - A generic CSV operation error
errorMessage	string	0..1	Descriptive error message.

Content Model

Component	Type	Occurs	Description
success	boolean	0..1	True: the operation was performed without error False: There was an error in the method

Element: errorCode [type WsResult]

Element: errorMessage [type WsResult]

Element: success [type WsResult]

NAC Configuration WebService

Description

Web Service for performing NAC Configuration related operations.

Web Service URL - <https://<netsightserverip>:8443/axis/services/NACConfigurationWebService>

See Also

[Methods](#) | [Complex Types](#)

Methods: NAC Configuration WebService

Methods

Name	Description
createDCMVirtualAndPhysicalNetwork	Create virtual and physical network configuration for the purpose of Data Center Manager Integration. This will create NAC Rule, NAC Profile, Policy Mapping, Policy Roles and VLANs in a given NAC Configuration and Domain
createSwitch	Create a new Switch in the NAC Context. The corresponding Device should be added to NetSight Console first.
createVirtualAndPhysicalNetwork	This will create NAC Rule, NAC Profile, Policy Mapping, Policy Roles and VLANs in a given NAC Configuration and Domain
deleteSwitch	Updates an existing Switch in the NAC Context. IP Address is required. All other values can be NULL to preserve the old value.
updateSwitch	Updates an existing Switch in the NAC Context. IP Address is required. All other values can be NULL to preserve the old value.

Method: createDCMVirtualAndPhysicalNetwork

Description

Create virtual and physical network configuration for the purpose of Data Center Manager Integration. This will create NAC Rule, NAC Profile, Policy Mapping, Policy Roles and VLANs in a given NAC Configuration and Domain

Action

urn:createDCMVirtualAndPhysicalNetwork

Style

Document

Input (Literal)

The input of this method is the argument createDCMVirtualAndPhysicalNetwork having the structure defined by the following table.

Name	Type	Occurs	Description
name	string	0..1	name to be used to create NAC Rule, NAC Profile and Policy Mapping,
nacConfig	string	0..1	NAC Configuration name
domain	string	0..1	Policy Domain name
isPrivateVlan	boolean	0..1	true if a private VLAN configuration must be used (VMWare v4.0+)
primaryVlanId	int	0..1	Primary VLAN ID

Input (Literal)

Name	Type	Occurs	Description
secondaryVlanId	int	0..1	Secondary VLAN ID, only required if isPrivateVlan is set to true, otherwise it can be null or -1 (VMWare v4.0+)
mode	string	0..1	VLAN Type. Valid values are "promiscuous", "isolated" or "community". The default is promiscuous if no value or null is specified (VMWare v4.0+)
forwardAsTagged	boolean	0..1	true for Forwarding as tagged
swGroup	string	0..1	(d)vSwitch group where the configuration must be created.
nic	string	0..1	Network interface where the config must be created (Citrix)
isSync	boolean	0..1	True to sync the configuration with a supported and enabled virtualization system
isApproval	boolean	0..1	True to enable approval workflow for VMs in this configuration.

Output (Literal)

The output of this method is the argument createDCMVirtualAndPhysicalNetworkResponse having the structure defined by the following table.

Name	Type	Occurs	Description
return	string	0..*	an Array of String values that list the results for each step. If input is invalid this method will return validation errors

Remarks

This method is designed to be used in DCM environments to provide all the configurations associated with automated VM provisioning in DCM. This method will create an endsystem group into NAC with the name defined by the 'name' variable that will be exported to the virtualization platform if isSync is selected with the vlan or private vlan configuration required in the virtualization platform.

In addition to this it will create an empty policy with the name defined by the 'name' variable and a NAC configuration to apply to devices in the endsystem group the defined policy.

The policy and NAC configuration must be enforced afterwards to deploy this configuration in the network.

The policy is created empty, any rules must be added afterwards.

Note on Domain rules creation.

Policy rules will be created in the domains used by the nac configuration specified **and** the domain specified. If only a NAC configuration is specified, rules, roles, and ES groups will be created in that NAC configuration and the domains used by that configuration. If a policy domain is specified, the policy roles will be created in that domain in addition to the domains used by the NAC configuration.

Remarks

If no domain or NAC configuration are provided, NAC rules and ES groups will be created in the default NAC configuration and no policy roles will be created.

Method: createSwitch**Description**

Create a new Switch in the NAC Context. The corresponding Device should be added to NetSight Console first.

Action

urn:createSwitch

Style

Document

Input (Literal)

The input of this method is the argument createSwitch having the structure defined by the following table.

Name	Type	Occurs	Description
nacApplianceGroup	string	0..1	NAC Appliance Group for the Switch.
ipAddress	string	0..1	IP Address of the Switch
switchType	string	0..1	Type of Switch. A NULL or empty value will assume the default type - Layer 2 Out-Of-Band. Possible types: Layer 2 Out-Of-Band Layer 2 Out-Of-Band Data Center Layer 2 Out-Of-Band with PEPs Layer 2 Controller PEP Layer 2 RADIUS Only Layer 3 Out-Of-Band Layer 3 Controller PEP VPN
primaryGateway	string	0..1	IP Address of Primary NAC Gateway
secondaryGateway	string	0..1	IP Address of Secondary NAC Gateway
tertiaryGateway	string	0..1	IP Address of Tertiary NAC Gateway
quaternaryGateway	string	0..1	IP Address of Quaternary NAC Gateway

Input (Literal)

Name	Type	Occurs	Description				
authType	string	0..1	Authentication Type. A NULL or empty value will assume the default type - Network Access. Valid Values are: <table><tr><td>ALL("Any Access"),</td></tr><tr><td>MANAGEMENT("Management Access"),</td></tr><tr><td>NETWORK("Network Access"),</td></tr><tr><td>NONE("Manual RADIUS Configuration")</td></tr></table>	ALL("Any Access"),	MANAGEMENT("Management Access"),	NETWORK("Network Access"),	NONE("Manual RADIUS Configuration")
ALL("Any Access"),							
MANAGEMENT("Management Access"),							
NETWORK("Network Access"),							
NONE("Manual RADIUS Configuration")							

Input (Literal)

Name	Type	Occurs	Description
attrsToSend	string	0..1	<p>Gateway RADIUS Attributes To Send. A NULL or empty value will assume the default type - Enterasys Policy.</p> <p>Valid Values are:</p> <ul style="list-style-type: none"> • "Enterasys Policy"; • "Custom Attribute"; • "Enterasys (HiPath) Wireless 802.1X"; • "Enterasys (HiPath) Wireless MAC"; • "RFC 3580 - VLAN Name"; • "RFC 3580 - VLAN Name & Enterasys Policy"; • "RFC 3580 - VLAN Name & Enterasys (HiPath) Wireless"; • "RFC 3580 - VLAN Name & Custom Attribute"; • "RFC 3580 - VLAN ID"; • "RFC 3580 - VLAN ID & Enterasys Policy"; • "RFC 3580 - VLAN ID & Enterasys (HiPath) Wireless"; • "RFC 3580 - VLAN ID & Custom Attribute"; • "Filter-Id"; • "Filter-Id & Custom Attribute"; • "Cisco Wireless Dynamic ACL"; • "Cisco Wired Dynamic ACL"; • "No Attributes";
isRadiusAccountingEnabled	boolean	0..1	true if RADIUS Accounting should be enabled
managementRadiusServer1	string	0..1	Managment RADIUS Server 1, only applicable if Auth Type is Network Access

Input (Literal)

Name	Type	Occurs	Description
managementRadiusServer2	string	0..1	Managment RADIUS Server 2, only applicable if Auth Type is Network Access
policyDomain	string	0..1	Policy Domain
pep1	string	0..1	Policy Enforcement Point 1, only valid when Switch Type is VPN.
pep2	string	0..1	Policy Enforcement Point 2, only valid when Switch Type is VPN.

Output (Literal)

The output of this method is the argument createSwitchResponse having the structure defined by the following table.

Name	Type	Occurs	Description
return	WsResult	0..1	a WsResult with success or error information

Method: createVirtualAndPhysicalNetwork**Description**

This will create NAC Rule, NAC Profile, Policy Mapping, Policy Roles and VLANs in a given NAC Configuration and Domain

Action

urn:createVirtualAndPhysicalNetwork

Style

Document

Input (Literal)

The input of this method is the argument createVirtualAndPhysicalNetwork having the structure defined by the following table.

Name	Type	Occurs	Description
name	string	0..1	name to be used to create NAC Rule, NAC Profile and Policy Mapping
nacConfig	string	0..1	NAC Configuration name where the configuration will be created.
domain	string	0..1	Domain name, policy domain where the policies will be created.

Input (Literal)

Name	Type	Occurs	Description
isPrivateVlan	boolean	0..1	true if it is a private VLAN configuration and a secondary VlanId and vlan mode must be provided for synchronizatin with VMware dvSwitches v>4.0
primaryVlanId	int	0..1	Primary VLAN ID
secondaryVlanId	int	0..1	Secondary VLAN ID, only required if isPrivateVlan is set to true, otherwise it can be null or -1
mode	string	0..1	VLAN Type. Valid values are "promiscuous", "isolated" or "community". The default is promiscuous if no value or null is specified.
forwardAsTagged	boolean	0..1	true for Forwarding as tagged. The policy created will add the primarivlanId to teh tagged egresslist.

Output (Literal)

The output of this method is the argument createVirtualAndPhysicalNetworkResponse having the structure defined by the following table.

Name	Type	Occurs	Description
return	string	0..*	an Array of String values that list the results for each step. If input is invalid this method will return validation errors

Remarks

This method is designed to be used in standalone environments to provide all the configurations associated with automated endsystem provisioning. This method will create an endsystem group into NAC with the name defined by the 'name' variable, a NAC rule that applies to that end system group to apply a policy role by that same name to teh ES in the ES group just created.

In addition to this it will create an empty policy with the name defined by the 'name' variable and a NAC configuration to apply to devices in the endsystem group the defined policy.

The policy and NAC configuration must be enforced afterwards to deploy this configuration in the network.

The policy is created empty, any rules must be added afterwards.

Note on Domain rules creation.

Policy rules will be created in the domains used by the nac configuration specified **and** the domain specified. If only a NAC configuration is specified, rules, roles, and ES groups will be created in that NAC configuration and the domains used by that configuration. If a policy domain is specified, the policy roles will be created in that domain in addition to the domains used by the NAC configuration.

If no domain or NAC configuration are provided, NAC rules and ES groups will be created in teh default NAC configuration and no policy roles will be created.

Method: deleteSwitch**Description**

Updates an existing Switch in the NAC Context. IP Address is required. All other values can be NULL to preserve the old value.

Action

urn:deleteSwitch

Style

Document

Input (Literal)

The input of this method is the argument deleteSwitch having the structure defined by the following table.

Name	Type	Occurs	Description
ipAddress	string	0..1	

Output (Literal)

The output of this method is the argument deleteSwitchResponse having the structure defined by the following table.

Name	Type	Occurs	Description
return	WsResult	0..1	

Method: updateSwitch**Description**

Updates an existing Switch in the NAC Context. IP Address is required. All other values can be NULL to preserve the old value.

Action

urn:updateSwitch

Style

Document

Input (Literal)

The input of this method is the argument updateSwitch having the structure defined by the following table.

Name	Type	Occurs	Description
nacApplianceGroup	string	0..1	NAC Appliance Group for the Switch

Input (Literal)

Name	Type	Occurs	Description
ipAddress	string	0..1	IP Address of the Switch
switchType	string	0..1	Type of Switch.
primaryGateway	string	0..1	IP Address of Primary NAC Gateway
secondaryGateway	string	0..1	IP Address of Secondary NAC Gateway
tertiaryGateway	string	0..1	IP Address of Tertiary NAC Gateway
quaternaryGateway	string	0..1	IP Address of Quaternary NAC Gateway
authType	string	0..1	Authentication Type.
attrsToSend	string	0..1	Gateway RADIUS Attributes To Send.
isRadiusAccountingEnabled	boolean	0..1	true if RADIUS Accounting should be enabled.
managementRadiusServer1	string	0..1	Management RADIUS Server 1, only applicable if Auth Type is Network Access
managementRadiusServer2	string	0..1	Management RADIUS Server 2, only applicable if Auth Type is Network Access
policyDomain	string	0..1	Policy Domain
pep1	string	0..1	Policy Enforcement Point 1, only valid when Switch Type is VPN.
pep2	string	0..1	Policy Enforcement Point 2, only valid when Switch Type is VPN.

Output (Literal)

The output of this method is the argument `updateSwitchResponse` having the structure defined by the following table.

Name	Type	Occurs	Description
return	WsResult	0..1	WsResult with success or error information

Complex Types: NAC Configuration WebService

Complex Types

Name	Description
WsResult	Result status of NAC operations

Complex Type: WsResult

Description

Result status of NAC operations

Derived By

Restricting anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
errorCode	int	0..1	0 - This error code indicates success. 1 - This error indicates that the requested object does not exist. 2 - This error indicates that the action cannot be performed because the object already exists. 3 - This error indicates that a parameter value is invalid. 4 - This error code indicates an error parsing an input string. 5 - This error code indicates that the result would be an invalid configuration. 6 - This error code is used to report an error using a remote connection. 7 - This error code is a catch-all for an unexpected error condition. 8 - This error code is used to report the group parameter does not exist 9 - A generic CSV operation error
errorMessage	string	0..1	Descriptive error message.
success	boolean	0..1	True: the operation was performed without error False: There was an error in the method

Element: errorCode [type WsResult]

Element: errorMessage [type WsResult]

Element: success [type WsResult]

Index

A

addAuthCredential	3
addAuthCredentialEx	4
addCredentialEx	5
addDeviceEx	6
addHostnameToEndSystemGroup	37
up	
addHostnameToEndSystemGroupEx	38
upEx	
addHostnameToEndSystemGroupWithCustomDataEx	39
addIPToEndSystemGroup	40
addIPToEndSystemGroupEx	41
addIPToEndSystemGroupWithCustomDataEx	42
addMACToBlacklist	43
addMACToBlacklistEx	44
addMACToBlacklistWithCustomDataEx	45
addMACToEndSystemGroup	45
addMACToEndSystemGroupEx	50
addMACToEndSystemGroupWithCustomDataEx	51
addProfileEx	7
addUsernameToUserGroup	52
addUsernameToUserGroupEx	53
addValueToNamedList	54
addValueToNamedListEx	56
auditEnforceNacAppliances	58

C

Complex Types	25, 141, 179
createDCMVirtualAndPhysicalNetwork	170

createMacLock	59
createSwitch	172
createVirtualAndPhysicalNetwork	175

D

deleteDeviceByIpEx	8
deleteEndSystemByMac	60
deleteEndSystemInfoByHostname	61
deleteEndSystemInfoByIp	62
deleteEndSystemInfoByMac	63
deleteEndSystemInfoEx	64
deleteLocalUsers	65
deleteLocalUsersEx	66
deleteMacLock	67
deleteRegisteredDevice	68
deleteRegisteredDevices	68
deleteRegisteredUserAndDevices	69
deleteRegisteredUsers	70
deleteSwitch	177

E

enforceNacAppliances	71
exportDevicesAsNgf	9

G

getAllDevices	11
getAllEndSystemMacs	72
getAllEndSystems	73
getDeviceByIpAddressEx	11
getEndSystemAndHrByMac	73

getEndSystemByIp	74	NAC Configuration WebService	169
getEndSystemByIpEx	77	NAC WebService	30
getEndSystemByMac	78	NetSight Device WebService	2
getEndSystemByMacEx	79	ns0:baseMac [type WsDevice]	26
getEndSystemInfoArrByMac	80	ns0:bootProm [type WsDevice]	27
getEndSystemInfoByMac	81	ns0:chassisId [type WsDevice]	27
getEndSystemInfoByMacEx	83	ns0:chassisType [type WsDevice]	27
getEndSystemsByMacEx	85	ns0:data [type WsDeviceListResult]	28
getExtendedEndSystemArrByMac	86	ns0:deviceId [type WsDevice]	27
getExtendedEndSystemByMac	87	ns0:errorCode [type NsWsResult]	25
getNACVersion	89	ns0:errorCode [type WsDeviceListResult]	29
getPollerStatus	92	ns0:errorMessage [type NsWsResult]	26
getRegisteredDevicesByMacAddress	92	ns0:errorMessage [type WsDeviceListResult]	29
getRegisteredUsersByUsername	93	ns0:firmware [type WsDevice]	27
getRegisteredDevicesByUsername	94	ns0:ip [type WsDevice]	27
getRegisteredUsersByMacAddresses	94	ns0:monitorType [type WsDevice]	27
getSnmpCredentialAsNgf	12	ns0:nickName [type WsDevice]	27
getUnsurfacedNamedList	95	ns0:note [type WsDevice]	27
H		ns0:NsWsResult	25
hashLocalUserPassword	97	ns0:pollGroup [type WsDevice]	27
I		ns0:snmpContext [type WsDevice]	27
importDevicesAsNgfEx	14	ns0:status [type WsDevice]	27
importEndSystemInfoEx	97	ns0:success [type NsWsResult]	26
importEndSystemInfoFromCsv	98	ns0:success [type WsDeviceListResult]	29
isIpV6Enabled	17	ns0:sysContact [type WsDevice]	27
isNetSnmpEnabled	17	ns0:sysDescriptor [type WsDevice]	27
M		ns0:sysLocation [type WsDevice]	27
Methods	3, 33, 170	ns0:sysName [type WsDevice]	27
N		ns0:sysObjectId [type WsDevice]	27

ns0:tableTotalRecords [type WsDeviceListResult]	29	ns1:locationInfo [type EndSystemDTO]	144
ns0:timeLastUpdated [type WsDevice]	27	ns1:macAddress [type EndSystemDTO]	144
ns0:userData1 [type WsDevice]	27	ns1:nacApplianceGroupName [type EndSystemDTO]	144
ns0:userData2 [type WsDevice]	27	ns1:nacApplianceIP [type EndSystemDTO]	144
ns0:userData3 [type WsDevice]	27	ns1:nacProfileName [type EndSystemDTO]	144
ns0:userData4 [type WsDevice]	27	ns1:operatingSystemName [type EndSystemDTO]	144
ns0:WsDevice	26	ns1:policy [type EndSystemDTO]	144
ns0:WsDeviceListResult	27	ns1:radiusServerIp [type EndSystemDTO]	144
ns1:allAuthTypes [type EndSystemDTO]	143	ns1:reason [type EndSystemDTO]	144
ns1:assmtHashCode [type EndSystemDTO]	143	ns1:regType [type EndSystemDTO]	144
ns1:authType [type EndSystemDTO]	143	ns1:requestAttributes [type EndSystemDTO]	144
ns1:EndSystemDTO	141	ns1:startAssmtWarningTime [type EndSystemDTO]	144
ns1:errorCode [type WsResult]	179	ns1:state [type EndSystemDTO]	144
ns1:errorMessage [type WsResult]	179	ns1:stateDescr [type EndSystemDTO]	144
ns1:eSType [type EndSystemDTO]	143	ns1:success [type WsResult]	180
ns1:extendedState [type EndSystemDTO]	143	ns1:switchIP [type EndSystemDTO]	144
ns1:firstSeenTime [type EndSystemDTO]	143	ns1:switchPort [type EndSystemDTO]	144
ns1:hostName [type EndSystemDTO]	143	ns1:switchPortId [type EndSystemDTO]	144
ns1:id [type EndSystemDTO]	143	ns1:username [type EndSystemDTO]	144
ns1:ipAddress [type EndSystemDTO]	143	ns1:WsResult	179
ns1:lastAssmtHashCodeChange Time [type EndSystemDTO]	144	ns1:zone [type EndSystemDTO]	145
ns1:lastAuthEventTime [type EndSystemDTO]	144	ns2:applianceGroup [type RegisteredDevice]	152
ns1:lastQuarantineTime [type EndSystemDTO]	144	ns2:applianceGroup [type RegisteredUser]	155
ns1:lastScanResultState [type EndSystemDTO]	144	ns2:attempts [type RegisteredUser]	155
ns1:lastScanTime [type EndSystemDTO]	144	ns2:AUTH_REG [type RegisteredUser]	156
ns1:lastSeenTime [type EndSystemDTO]	144		

ns2:AUTH_REG_LONG_COMMENT [type RegisteredUser]	156	ns2:groupDescr1 [type EndSystemInfo]	146
ns2:AUTH_REG_SHORT_COMMENT [type RegisteredUser]	156	ns2:groupDescr2 [type EndSystemInfo]	146
ns2:COMMA_SEP [type RegisteredUser]	156	ns2:groupDescr3 [type EndSystemInfo]	146
ns2:COMMA_SUBSTITUTE [type RegisteredUser]	156	ns2:GUEST [type RegisteredUser]	156
ns2:COMMA_SUBSTITUTE_SE P [type RegisteredUser]	156	ns2:GUEST_LONG_COMMENT [type RegisteredUser]	156
ns2:custom1 [type EndSystemInfo]	145	ns2:GUEST_SHORT_COMMENT [type RegisteredUser]	156
ns2:custom2 [type EndSystemInfo]	145	ns2:GUEST_WEB_ACCESS_LONG_COMMENT [type RegisteredUser]	156
ns2:custom3 [type EndSystemInfo]	146	ns2:GUEST_WEB_ACCESS_SHORT_COMMENT [type RegisteredUser]	156
ns2:custom4 [type EndSystemInfo]	146	ns2:id [type LocalUser]	151
ns2:dbData [type LocalUser]	150	ns2:id [type RegisteredDevice]	152
ns2:DEFAULT_PASSWORD_HASH_TYPE [type LocalUser]	151	ns2:id [type RegisteredUser]	155
ns2:description [type LocalUser]	150	ns2:idAsString [type RegisteredDevice]	152
ns2:description [type RegisteredDevice]	152	ns2:idAsString [type RegisteredUser]	155
ns2:deviceGroup [type RegisteredDevice]	152	ns2:ipAddress [type RegisteredDevice]	152
ns2:displayName [type LocalUser]	150	ns2:lastName [type LocalUser]	151
ns2:displayName [type RegisteredUser]	155	ns2:lastName [type RegisteredUser]	155
ns2:domainName [type LocalUser]	150	ns2:LocalUser	149
ns2:emailAddress [type RegisteredUser]	155	ns2:loginId [type LocalUser]	151
ns2:enabled [type LocalUser]	150	ns2:loginPassword [type LocalUser]	151
ns2:EndSystemInfo	145	ns2:loginPasswordHash [type LocalUser]	151
ns2:endSystemKey [type EndSystemInfo]	146	ns2:loginPasswordHashType [type LocalUser]	151
ns2:expiresTime [type RegisteredUser]	155	ns2:macAddress [type RegisteredDevice]	152
ns2:firstName [type LocalUser]	150	ns2:MAX_USERNAME_LENGTH [type RegisteredUser]	156
ns2:firstName [type RegisteredUser]	155	ns2:maxRegisterCount [type RegisteredUser]	155
ns2:fullName [type RegisteredUser]	155	ns2:memberOfGroups [type EndSystemInfo]	146

ns2:middleName [type RegisteredUser]	155	ns2:SECURE_GUEST_LONG_ COMMENT [type RegisteredUser]	156
ns2:phoneNumber [type RegisteredUser]	155	ns2:SECURE_GUEST_SHORT _COMMENT [type RegisteredUser]	157
ns2:PKCS5_REVERSIBLE_HA SH_TYPE [type LocalUser]	151	ns2:SHA1_NON_REVERSIBLE _HASH_TYPE [type LocalUser]	151
ns2:PRE_REG_LONG_COMME NT [type RegisteredUser]	156	ns2:sponsor [type RegisteredDevice]	152
ns2:PRE_REG_SHORT_COMM ENT [type RegisteredUser]	156	ns2:sponsor [type RegisteredUser]	156
ns2:preOrSelfProvisioned [type LocalUser]	151	ns2:sponsorDeviceGroup [type RegisteredDevice]	152
ns2:preRegistered [type RegisteredUser]	155	ns2:sponsored [type RegisteredDevice]	152
ns2:regData1 [type EndSystemInfo]	146	ns2:SPONSORED_AUTH_REG _LONG_COMMENT [type RegisteredUser]	157
ns2:regData2 [type EndSystemInfo]	146	ns2:SPONSORED_AUTH_REG _SHORT_COMMENT [type RegisteredUser]	157
ns2:regData3 [type EndSystemInfo]	146	ns2:SPONSORED_GUEST_LO NG_COMMENT [type RegisteredUser]	157
ns2:regData4 [type EndSystemInfo]	146	ns2:SPONSORED_GUEST_SH ORT_COMMENT [type RegisteredUser]	157
ns2:regData5 [type EndSystemInfo]	146	ns2:SPONSORED_PRE_REG_ LONG_COMMENT [type RegisteredUser]	157
ns2:regDeviceDescr [type EndSystemInfo]	146	ns2:SPONSORED_PRE_REG_ SHORT_COMMENT [type RegisteredUser]	157
ns2:regEmail [type EndSystemInfo]	146	ns2:SPONSORED_SECURE_G UEST_LONG_COMMENT [type RegisteredUser]	157
ns2:REGISTERED_DEVICE_C OMMENT [type RegisteredUser]	156	ns2:SPONSORED_SECURE_G UEST_SHORT_COMMENT [type RegisteredUser]	157
ns2:RegisteredDevice	151	ns2:startTime [type RegisteredUser]	156
ns2:registeredDeviceInfo [type EndSystemInfo]	146	ns2:stateStr [type RegisteredDevice]	152
ns2:RegisteredUser	152	ns2:STR_DEFAULT_ADMIN_N AME [type LocalUser]	151
ns2:registeredUserInfo [type EndSystemInfo]	147	ns2:STR_DEFAULT_DOMAIN [type LocalUser]	151
ns2:registrationTime [type RegisteredDevice]	152	ns2:STR_DEFAULT_SPONSO R_NAME [type LocalUser]	151
ns2:registrationTime [type RegisteredUser]	155		
ns2:regName [type EndSystemInfo]	146		
ns2:regPhone [type EndSystemInfo]	146		
ns2:regSponsor [type EndSystemInfo]	146		

ns2:TRANSIENT [type RegisteredUser]	157	ns3:errorCode [type WsResult]	168
ns2:userData1 [type RegisteredUser]	156	ns3:errorMessage [type WsEndSystemInfoResult]	159
ns2:userData2 [type RegisteredUser]	156	ns3:errorMessage [type WsEndSystemListResult]	162
ns2:userData3 [type RegisteredUser]	156	ns3:errorMessage [type WsEndSystemResult]	165
ns2:userData4 [type RegisteredUser]	156	ns3:errorMessage [type WsEnforceResult]	167
ns2:userData5 [type RegisteredUser]	156	ns3:errorMessage [type WsResult]	168
ns2:userName [type RegisteredDevice]	152	ns3:errors [type WsEnforceApplianceResult]	166
ns2:userName [type RegisteredUser]	156	ns3:nacApplianceIP [type WsEnforceApplianceResult]	166
ns2:USERNAME_COMMENT [type RegisteredUser]	157	ns3:success [type WsEndSystemInfoResult]	159
ns2:userTypeStr [type RegisteredUser]	156	ns3:success [type WsEndSystemListResult]	162
ns2:WEB_AUTH [type RegisteredUser]	157	ns3:success [type WsEndSystemResult]	165
ns2:WEB_AUTH_COMMENT [type RegisteredUser]	157	ns3:success [type WsEnforceResult]	167
ns2:WEB_AUTH_LONG_COMMENT [type RegisteredUser]	157	ns3:success [type WsResult]	168
ns2:WEB_AUTH_SHORT_COMMENT [type RegisteredUser]	157	ns3:warnings [type WsEnforceApplianceResult]	166
ns3:applianceEnforceResults [type WsEnforceResult]	167	ns3:WsEndSystemInfoResult	157
ns3:endSystem [type WsEndSystemResult]	163	ns3:WsEndSystemListResult	159
ns3:endSystemInfo [type WsEndSystemInfoResult]	158	ns3:WsEndSystemResult	162
ns3:endSystems [type WsEndSystemListResult]	160	ns3:WsEnforceApplianceResult	165
ns3:endSystemSwitchSupports Reauth [type WsEndSystemResult]	165	ns3:WsEnforceResult	166
ns3:errorCode [type WsEndSystemInfoResult]	159	ns3:WsResult	167
ns3:errorCode [type WsEndSystemListResult]	162		
ns3:errorCode [type WsEndSystemResult]	165		
ns3:errorCode [type WsEnforceResult]	167		

O

OF Connect WebServices 1

P

processNacRequestArrFromCsv 99

processNacRequestFromCsv 100

R

reauthenticate	101
reauthenticateEx	105
removeHostnameFromEndSystemGroup	106
removeHostnameFromEndSystemGroupEx	107
removeIPFromEndSystemGroup	108
removeIPFromEndSystemGroupEx	109
removeMACFromBlacklist	110
removeMACFromBlacklistEx	111
removeMACFromEndSystemGroup	112
removeMACFromEndSystemGroupEx	113
removeUsernameFromUserGroup	114
removeUsernameFromUserGroupEx	115
removeValueFromNamedList	116
removeValueFromNamedListEx	118

S

saveEndSystemInfo	120
saveEndSystemInfoByHostname	121
saveEndSystemInfoByIp	122
saveEndSystemInfoByMac	124

saveEndSystemInfoEx	128
saveLocalUser	129
saveLocalUserEx	129
saveRegisteredDevice	130
saveRegisteredDeviceEx	131
saveRegisteredDevices	132
saveRegisteredDeviceWithSponsorship	133
saveRegisteredDeviceWithSponsorshipEx	134
saveRegisteredUser	135
saveRegisteredUserEx	136
saveRegisteredUsers	137

U

updateAuthCredential	18
updateAuthCredentialEx	19
updateCredential	20
updateCredentialEx	21
updateDevicesEx	22
updateProfile	22
updateProfileEx	23
updateRegisteredDevice	138
updateRegisteredUser	139
updateSwitch	177